yoTech24 · Akademie

IT-Grundlagen

Netzwerk, Betriebssysteme, Hardware, Server, Virtualisierung, Storage, Security, Cloud, Monitoring, Scripting sowie Ausbildung & Soft Skills – verständlich erklärt.

89 Kapitel · Lesefassung zum Nachschlagen und Ausdrucken · Stand Oktober 2026 · yotech24.com/akademie

Inhalt

LB-01 · Netzwerk

1. Netzwerk-Grundlagen & OSI-Modell
2. IP-Adressierung & Subnetting
3. Switching & VLANs
4. STP, RSTP & Link Aggregation
5. Routing-Grundlagen
6. Dynamisches Routing
7. NAT & PAT
8. WLAN-Technologien
9. VPN-Technologien
10. Netzwerk-Diagnose & Tools
11. Netzwerk-Sicherheit
12. SD-WAN & moderne Netzarchitekturen

LB-02 · Betriebssysteme

1. OS-Geschichte & Hersteller
2. OS-Kernkomponenten & Architektur
3. Windows Desktop – Grundlagen & Administration
4. Windows Server – Grundlagen
5. DOS, CMD & klassische Befehle
6. PowerShell – Grundlagen & Praxis
7. Linux-Administration
8. Software-Deployment & Paketsysteme
9. Update- & Patch-Management
10. CVE, Zero-Day & Schwachstellen-Management
11. macOS-Administration
12. Dateisysteme

LB-03 · Hardware

1. CPU – Geschichte, Architektur & Markt
2. RAM – Typen, Specs & ECC
3. Storage – HDD, SSD, NVMe
4. Mainboard & Formfaktoren
5. Server-Hardware
6. Netzwerkkarten & Verkabelung
7. Hardware-Diagnose & Lifecycle

LB-04 · Server & Dienste

1. Active Directory – Grundlagen
2. Active Directory – Administration
3. DNS – Deep Dive
4. DHCP – Deep Dive
5. Fileserver & DFS
6. Printserver & Druckverwaltung
7. Remote Desktop Services (RDS)
8. IIS & Webserver-Grundlagen
9. Datenbankserver

LB-05 · Virtualisierung & Container

1. Virtualisierung – Grundlagen
2. VMware & ESXi
3. Hyper-V
4. Proxmox VE
5. Docker & Container-Grundlagen
6. Kubernetes-Einstieg

LB-06 · Storage & Backup

1. RAID-Systeme
2. SAN & NAS
3. Backup-Grundlagen & Strategien
4. VSS, Snapshots & agentenloses Backup
5. Datenbank-Backup
6. Backup-Software & Hersteller

LB-07 · IT-Security

1. Schutzziele & Grundlagen
2. Malware & Angriffstechniken
3. Antivirus & NGAV
4. EDR & XDR
5. Firewall & Netzwerksicherheit
6. Mail-Security & Anti-Spam
7. SSL/TLS & PKI
8. Identity & Access Management
9. SIEM & SOC-Grundlagen
10. BSI IT-Grundschutz & Compliance

LB-08 · Cloud

1. Cloud-Grundlagen & Modelle
2. Microsoft 365 – Administration
3. Microsoft Azure – Kernprodukte
4. AWS – Kernprodukte
5. Google Cloud & Workspace
6. Cloud-Migration
7. Infrastructure as Code
8. Cloud-Security & FinOps

LB-09 · Monitoring & Management

1. Monitoring-Grundlagen
2. Monitoring-Tools Open Source
3. Monitoring-Tools Commercial
4. RMM-Systeme
5. ITSM & Ticketsysteme
6. IT-Dokumentation

LB-10 · Scripting & Automatisierung

1. PowerShell – Grundlagen
2. PowerShell – Praxis für Admins
3. Bash & Linux-Scripting
4. Cron & Task-Scheduling
5. Python für IT-Admins
6. Git & Versionskontrolle
7. CI/CD-Grundlagen

LB-13 · Ausbildung & Soft Skills

1. Kundenkommunikation
2. Projektarbeit & Zeitmanagement
3. Teamarbeit & Konfliktmanagement
4. Präsentation & Fachgespräch
5. Lernstrategien & Prüfungsvorbereitung
6. Ausbildungsrecht & Berufseinstieg

LB-01

Netzwerk

1. Netzwerk-Grundlagen & OSI-Modell
2. IP-Adressierung & Subnetting
3. Switching & VLANs
4. STP, RSTP & Link Aggregation
5. Routing-Grundlagen
6. Dynamisches Routing
7. NAT & PAT
8. WLAN-Technologien
9. VPN-Technologien
10. Netzwerk-Diagnose & Tools
11. Netzwerk-Sicherheit
12. SD-WAN & moderne Netzarchitekturen

LB-01 · NETZWERK · KAPITEL 1

Netzwerk-Grundlagen & OSI-Modell

Die sieben Schichten als roter Faden jeder Fehlersuche

Nach diesem Kapitel können Sie …

- die sieben Schichten des OSI-Modells benennen und ihre Aufgaben beschreiben,
- Switch, Router und Firewall der passenden Schicht zuordnen,
- Kollisions- und Broadcast-Domäne unterscheiden,
- TCP und UDP vergleichen und die Rolle von Ports erklären,
- eine Fehlersuche systematisch von unten nach oben aufbauen.

Warum ein Schichtenmodell?

Wenn zwei Computer miteinander kommunizieren, passiert eine ganze Menge gleichzeitig: Elektrische Signale oder Lichtimpulse laufen durch ein Kabel, Geräte im lokalen Netz müssen einander finden, Pakete müssen über mehrere Netze hinweg ihren Weg finden, und am Ende soll eine Anwendung – etwa ein Browser oder ein E-Mail-Programm – genau die Daten erhalten, die sie erwartet. Um diese Vielfalt beherrschbar zu machen, teilt man die Kommunikation in Schichten auf. Jede Schicht hat eine klar umrissene Aufgabe und arbeitet nur mit den Schichten direkt über und unter ihr zusammen.

Das bekannteste Modell dafür ist das **OSI-Modell** (Open Systems Interconnection). Es wirkt auf den ersten Blick wie trockene Theorie, ist im Alltag aber eines der wichtigsten Diagnosewerkzeuge. Wer weiß, auf welcher Schicht ein Problem liegt, weiß auch, welches Gerät oder welchen Dienst er prüfen muss. Die Fehlersuche wird dadurch systematisch statt zum Ratespiel. Zudem sorgt das Modell für eine herstellerunabhängige Standardisierung: Geräte unterschiedlicher Hersteller können zusammenarbeiten, weil sie sich an dieselbe Aufgabenteilung halten. Und nicht zuletzt gehört das OSI-Modell zum festen Bestandteil jeder IT-Prüfung.

Die Schichten 1 bis 4

Die unteren vier Schichten kümmern sich um den eigentlichen Transport der Daten. Ganz unten liegt **Schicht 1, die Bitübertragung**. Hier geht es um das physische Medium: Kupferkabel, Glasfaser oder Funk. Diese Schicht legt fest, wie Nullen und Einsen als Signale übertragen werden. Ein defektes Kabel, ein lockerer Stecker oder eine gestörte Funkverbindung sind typische Probleme auf dieser Ebene.

Schicht 2, die Sicherung, organisiert die Kommunikation innerhalb eines lokalen Netzes. Hier arbeiten Switches, und hier spielen *MAC-Adressen* die Hauptrolle – fest vergebene Hardware-Adressen jeder Netzwerkkarte. Das bekannteste Protokoll dieser Schicht ist Ethernet. **Schicht 3, die Vermittlung**, verbindet verschiedene Netze miteinander. Hier arbeiten Router, die Pakete anhand von *IP-Adressen* weiterleiten; auch das Subnetting gehört auf diese Ebene.

Schicht 4, der Transport, sorgt dafür, dass Daten zwischen zwei Anwendungen ankommen. Hier sind die beiden Protokolle *TCP* und *UDP* zu Hause, die weiter unten genauer beschrieben werden.

Die Schichten 5 bis 7

Die oberen drei Schichten sind näher an der Anwendung. **Schicht 5, die Sitzung**, regelt den Auf- und Abbau von Verbindungen. **Schicht 6, die Darstellung**, kümmert sich um die Kodierung der Daten, also darum, dass Sender und Empfänger Zeichen und Formate gleich interpretieren; auch die Verschlüsselung wird hier eingeordnet. **Schicht 7, die Anwendung**, ist die Ebene der Protokolle, mit denen Programme direkt arbeiten: HTTP für Webseiten, DNS für die Namensauflösung, SMTP für E-Mail oder SSH für den sicheren Fernzugriff.

In der Praxis werden die Schichten 5 bis 7 häufig zu den „Anwendungsschichten“ zusammengefasst, weil sich ihre Aufgaben in modernen Protokollen kaum noch sauber trennen lassen. Für die Fehlersuche genügt es meist zu wissen, dass ein Problem „oberhalb des Transports“ liegt, also bei der Anwendung oder einem Dienst wie DNS.

Schicht	Name	Typische Beispiele
7–5	Anwendung, Darstellung, Sitzung	HTTP, DNS, SMTP, SSH, Verschlüsselung
4	Transport	TCP, UDP, Ports
3	Vermittlung	Router, IP-Adressen, Subnetting
2	Sicherung	Switch, MAC-Adressen, Ethernet
1	Bitübertragung	Kabel, Glasfaser, Funk

Geräte und ihre Schichten

Viele Netzwerkgeräte lassen sich eindeutig einer Schicht zuordnen. Ein **Switch** arbeitet auf Schicht 2: Er leitet Daten anhand der MAC-Adresse gezielt an den richtigen Anschluss im lokalen Netz weiter. Ein **Router** arbeitet auf Schicht 3 und verbindet unterschiedliche Netze anhand der IP-Adresse – etwa das Firmennetz mit dem Internet. Der früher verbreitete **Hub** arbeitete auf Schicht 1: Er gab jedes eingehende Signal einfach

an alle Anschlüsse weiter. Hubs sind heute vollständig durch Switches abgelöst. Moderne **Firewalls**, insbesondere sogenannte Next-Generation-Firewalls, prüfen den Verkehr dagegen bis in die oberen Schichten und erkennen zum Beispiel, welche Anwendung hinter einer Verbindung steckt.

Kollisions- und Broadcast-Domäne

Zwei Begriffe tauchen in der Netzwerktechnik immer wieder auf. Eine **Kollisionsdomäne** ist der Bereich, in dem sich gleichzeitig gesendete Signale gegenseitig stören können. Bei Hubs teilten sich alle Geräte eine einzige Kollisionsdomäne. Bei einem Switch ist jeder Port eine eigene Kollisionsdomäne, sodass Kollisionen heute praktisch keine Rolle mehr spielen.

Eine **Broadcast-Domäne** ist dagegen der Bereich, den ein Rundruf – ein Broadcast – erreicht. Ohne weitere Unterteilung bildet ein ganzer Switch, oft sogar ein ganzes Netz aus mehreren Switches, eine gemeinsame Broadcast-Domäne. Jeder Broadcast erreicht dann alle Geräte und muss von ihnen verarbeitet werden. Je größer die Domäne, desto mehr Last entsteht. Deshalb unterteilt man Netze mit *VLANs*: Jedes VLAN bildet eine eigene Broadcast-Domäne. Kleinere Domänen bedeuten weniger Last und mehr Übersicht.

> **Merke**
>
> Switch-Ports trennen Kollisionsdomänen, VLANs und Router trennen Broadcast-Domänen. Ein VLAN entspricht genau einer Broadcast-Domäne.

TCP, UDP und Ports

Auf Schicht 4 gibt es zwei grundlegend verschiedene Arbeitsweisen. **TCP** ist verbindungsorientiert und zuverlässig: Vor dem Datenaustausch wird eine Verbindung aufgebaut, jede Lieferung wird bestätigt, und verlorene Daten werden erneut gesendet. Das kostet etwas Zeit, garantiert aber, dass alles vollständig und in der richtigen Reihenfolge ankommt – wichtig etwa beim Laden einer Webseite oder beim Versand einer E-Mail. **UDP** ist dagegen verbindungslos und schnell. Es schickt Daten ohne Bestätigung los. Gehen einzelne Pakete verloren, wird nicht nachgeliefert. Das ist ideal für Streaming, Internettelefonie (VoIP) oder DNS-Anfragen, wo Tempo wichtiger ist als Vollständigkeit. Als Faustregel gilt: TCP für Korrektheit, UDP für Tempo.

Damit ein Rechner weiß, für welche Anwendung ein Paket bestimmt ist, gibt es **Ports**. Man kann sie sich als Türnummern vorstellen: Die IP-Adresse findet das Gerät, der Port findet den Dienst darauf. Einige Ports sind fest vergeben und sollten bekannt sein: 80 für HTTP, 443 für HTTPS, 22 für SSH, 53 für DNS, 25 für SMTP und 3389 für den Win-

dows-Remotedesktop (RDP). Die Ports 0 bis 1023 heißen „Well-Known Ports“ und sind meist Systemdiensten vorbehalten. Wichtig ist auch: Ein offener Port allein bringt nichts – nur wenn dahinter ein Dienst läuft, ist er erreichbar.

Aus der Praxis

Funktionieren plötzlich keine verschlüsselten Webseiten mehr, unverschlüsselte aber schon, lohnt ein Blick auf die Firewall: Wird Port 443 blockiert, scheitern alle HTTPS-Verbindungen. Mit dem Wissen um Ports lässt sich so ein Fehler gezielt eingrenzen, statt zu raten.

Fehlersuche von unten nach oben

Der größte praktische Nutzen des OSI-Modells liegt in einer klaren Reihenfolge für die Fehlersuche. Beginnen Sie immer unten: Ist das Kabel eingesteckt, leuchtet die Link-LED, besteht eine WLAN-Verbindung? Danach folgt die IP-Konfiguration auf Schicht 3: Hat das Gerät eine gültige Adresse, stimmt das Gateway? Erst dann prüfen Sie die Namensauflösung per DNS und zuletzt die Anwendung selbst.

Diese Reihenfolge verhindert, dass Sie stundenlang eine Software-Einstellung untersuchen, obwohl in Wahrheit nur ein Kabel locker ist. Jede Schicht baut auf der darunterliegenden auf – funktioniert eine untere Schicht nicht, kann keine darüber zuverlässig arbeiten. Die konkreten Werkzeuge für jeden Schritt werden im Kapitel zur Netzwerk-Diagnose ausführlich behandelt.

Das Wichtigste in Kürze

- Das OSI-Modell teilt Kommunikation in sieben Schichten mit klaren Aufgaben.
- Switches arbeiten auf Schicht 2 (MAC), Router auf Schicht 3 (IP).
- Ein VLAN bildet eine eigene Broadcast-Domäne.
- TCP steht für Zuverlässigkeit, UDP für Tempo; Ports adressieren den Dienst.
- Fehlersuche immer von unten nach oben: Kabel, IP, DNS, Anwendung.

IP-Adressierung & Subnetting

IPv4, CIDR, private Bereiche, Subnetzrechnung und IPv6

Nach diesem Kapitel können Sie …

- den Aufbau einer IPv4-Adresse und die Rolle der Subnetzmaske erklären,
- CIDR-Schreibweise lesen und die Größe eines Netzes bestimmen,
- private Adressbereiche nach RFC 1918 erkennen,
- ein Netz mit der Blockgrößen-Methode in Subnetze aufteilen,
- die Grundzüge von IPv6 einordnen.

Wozu IP-Adressen?

Jedes Gerät in einem Netzwerk braucht eine eindeutige Adresse, damit Daten den richtigen Empfänger erreichen. Auf Schicht 2 gibt es dafür bereits die fest eingebaute MAC-Adresse. Sie reicht aber nur innerhalb eines lokalen Netzes. Sobald Daten über Netzgrenzen hinweg transportiert werden sollen – etwa vom Büro-PC zu einem Server im Internet –, braucht es eine **logische Adresse** auf Schicht 3: die **IP-Adresse**.

Eine IP-Adresse besteht immer aus zwei Teilen: dem *Netz-Anteil*, der angibt, zu welchem Netz ein Gerät gehört, und dem *Host-Anteil*, der das einzelne Gerät innerhalb dieses Netzes bezeichnet. Router benötigen nur den Netz-Anteil, um zu entscheiden, wohin sie ein Paket weiterleiten. Diese Zweiteilung macht Routing über viele Netze hinweg überhaupt erst möglich.

Aufbau einer IPv4-Adresse

Eine IPv4-Adresse ist 32 Bit lang. Damit Menschen sie lesen können, schreibt man sie als vier Zahlenblöcke, sogenannte *Oktette*, die durch Punkte getrennt sind – zum Beispiel `192.168.10.25`. Jedes Oktett umfasst 8 Bit und kann daher Werte von 0 bis 255 annehmen.

Welcher Teil der Adresse zum Netz und welcher zum Host gehört, legt die **Subnetzmaske** fest. Die Maske `255.255.255.0` bedeutet: Die ersten 24 Bit bilden das Netz, die letzten 8 Bit stehen für die Geräte. Zwei Adressen in jedem Netz sind reserviert und können keinem Gerät zugewiesen werden: die *Netzadresse*, bei der alle Host-Bits 0 sind, und die *Broadcast-Adresse*, bei der alle Host-Bits 1 sind. Im Netz `192.168.10.0` mit der genannten Maske ist also `192.168.10.0` die Netzadresse und `192.168.10.255` der Broadcast.

CIDR-Schreibweise

Lange Subnetzmasken sind unhandlich. Deshalb hat sich die **CIDR-Notation** durchgesetzt (Classless Inter-Domain Routing): Hinter die Netzadresse schreibt man nach einem Schrägstrich einfach die Anzahl der Netz-Bits. `192.168.10.0/24` ist gleichbedeutend mit der Maske `255.255.255.0`.

Aus der Präfixlänge lässt sich die Größe eines Netzes direkt ablesen. Ein /24-Netz hat 8 Host-Bits und damit 256 Adressen, von denen 254 nutzbar sind. Ein /25 umfasst 128 Adressen, ein /26 noch 64 und ein /27 nur 32. Jedes zusätzliche Netz-Bit halbiert also die Größe. Merken Sie sich: Je größer die Zahl hinter dem Schrägstrich, desto kleiner das Netz.

Präfix	Maske (letztes Oktett)	Adressen	Nutzbare Hosts
/24	.0	256	254
/25	.128	128	126
/26	.192	64	62
/27	.224	32	30
/30	.252	4	2

Private Adressbereiche

Öffentliche IPv4-Adressen sind knapp. Deshalb wurden im Standard **RFC 1918** drei Bereiche festgelegt, die jede Organisation intern frei verwenden darf: `10.0.0.0/8` für sehr große Netze, `172.16.0.0/12` für mittlere und `192.168.0.0/16`, das man aus nahezu jedem Heim- und KMU-Netz kennt. Diese Bereiche werden häufig noch mit den alten Adressklassen A, B und C in Verbindung gebracht, die heute aber durch CIDR abgelöst sind.

Private Adressen werden im Internet nicht geroutet. Damit Geräte mit solchen Adressen trotzdem ins Internet kommen, übersetzt der Router sie beim Übergang in eine öffentliche Adresse. Dieses Verfahren heißt *NAT* und wird in einem eigenen Kapitel behandelt.

Subnetting Schritt für Schritt

Beim **Subnetting** teilt man ein Netz in mehrere kleinere Teilnetze auf, zum Beispiel um Abteilungen voneinander zu trennen. Dazu „leiht“ man sich Bits aus dem Host-Anteil und nutzt sie als zusätzliche Netz-Bits. Ein Beispiel: Das Netz `192.168.1.0/24` soll in vier gleich große Subnetze geteilt werden. Für vier Netze braucht man zwei zusätzliche Bits, denn $2^2 = 4$. Aus /24 wird also /26. Jedes Subnetz hat dann 64 Adressen, davon 62 nutzbar.

Die vier Subnetze lauten `.0–.63`, `.64–.127`, `.128–.191` und `.192–.255`. In jedem Block ist die erste Adresse die Netzadresse und die letzte der Broadcast. Im zweiten Subnetz ist also `.64` das Netz, `.65` die erste nutzbare Adresse, `.126` die letzte nutzbare und `.127` der Broadcast.

> **Merke**
>
> Blockgröße = 256 − Maskenwert im letzten relevanten Oktett. Bei /27 (Maske .224) ergibt das 32: Die Subnetze beginnen bei .0, .32, .64 und so weiter. Nutzbare Hosts = Adressen im Block minus 2.

Mit dieser Methode lassen sich auch Prüfungsaufgaben schnell lösen: Sie bestimmen die Blockgröße, suchen den Block, in dem die gegebene Adresse liegt, und lesen Netzadresse, erste und letzte Host-Adresse sowie Broadcast direkt ab.

In der Praxis sind nicht alle Teilnetze gleich groß. Mit **VLSM** (Variable Length Subnet Mask) können Sie Subnetze unterschiedlicher Größe bilden: große für Abteilungen mit vielen Geräten, sehr kleine /30-Netze mit genau zwei nutzbaren Adressen für Punkt-zu-Punkt-Verbindungen zwischen zwei Routern. Das spart Adressen gegenüber lauter gleich großen Netzen. Damit sich Bereiche nicht überschneiden, plant man von groß nach klein: Zuerst werden die größten Subnetze vergeben, danach die kleineren in den verbleibenden Lücken.

IPv6 in Kürze

Der Nachfolger **IPv6** löst die Adressknappheit grundsätzlich. Eine IPv6-Adresse ist 128 Bit lang und wird hexadezimal in acht Blöcken geschrieben, etwa `2001:db8::1`. Zur Verkürzung dürfen führende Nullen in einem Block weggelassen werden, und eine einzige zusammenhängende Folge von Null-Blöcken darf durch `::` ersetzt werden. Der Adressraum ist so groß, dass praktisch jedes Gerät eine weltweit eindeutige Adresse erhalten kann – NAT ist dafür nicht mehr nötig.

Im Alltag sollten Sie einige Grundregeln kennen. Ein Subnetz hat bei IPv6 standardmäßig das Präfix /64. Geräte beziehen ihre Adresse entweder selbstständig per *SLAAC* (Stateless Address Autoconfiguration) oder von einem DHCPv6-Server. Zusätzlich besitzt jede IPv6-Schnittstelle immer eine *Link-Local-Adresse* aus dem Bereich `fe80::/10`, die nur im lokalen Netzsegment gilt. Die meisten Netze arbeiten heute im **Dual-Stack**-Betrieb, in dem IPv4 und IPv6 parallel laufen.

Aus der Praxis

Sieht ein Rechner bei IPv6 nur eine Adresse mit `fe80::`, ist das kein Fehler – diese Adresse ist immer vorhanden. Fehlt aber eine globale Adresse, erhält das Gerät aus dem Netz keine IPv6-Konfiguration.

Das Wichtigste in Kürze

- Eine IP-Adresse besteht aus Netz- und Host-Anteil, getrennt durch die Subnetzmaske.
- CIDR /n gibt die Zahl der Netz-Bits an; größerer Präfix bedeutet kleineres Netz.
- RFC-1918-Adressen sind privat und gelangen nur per NAT ins Internet.
- Subnetting: Blockgröße = 256 − Maskenwert im letzten Oktett.
- IPv6 nutzt 128 Bit, /64 pro Subnetz und kommt ohne NAT aus.

Switching & VLANs

MAC-Tabelle, Broadcast-Domänen, 802.1Q und Inter-VLAN-Routing

Nach diesem Kapitel können Sie …

- beschreiben, wie ein Switch MAC-Adressen lernt und Frames weiterleitet,
- erklären, wozu VLANs dienen und warum sie Broadcast-Domänen trennen,
- Access- und Trunk-Ports sowie das 802.1Q-Tagging unterscheiden,
- die Varianten des Inter-VLAN-Routings einordnen,
- ein einfaches VLAN-Design planen und typische Fehler erkennen.

Was ein Switch tut

Ein **Switch** ist das zentrale Gerät im lokalen Netz. Er arbeitet auf Schicht 2 des OSI-Modells und verbindet Geräte, die zum selben Netz gehören – PCs, Drucker, Server, Access Points oder Telefone. Die Daten, die er transportiert, heißen auf dieser Ebene *Frames* (Datenrahmen). Jeder Frame enthält die MAC-Adresse des Absenders und des Empfängers.

Anders als der früher übliche Hub, der jedes Signal stumpf an alle Anschlüsse weitergab, leitet ein Switch einen Frame gezielt nur an den Port weiter, an dem das Ziel tatsächlich angeschlossen ist. Das entlastet das Netz erheblich und erhöht die Sicherheit, weil andere Geräte den Verkehr gar nicht erst zu sehen bekommen.

Die MAC-Adresstabelle

Woher weiß der Switch, welches Gerät an welchem Port hängt? Er lernt es selbstständig. Jedes Mal, wenn ein Frame eintrifft, schaut der Switch auf die *Absender*-MAC-Adresse und merkt sich in seiner **MAC-Adresstabelle**, an welchem Port diese Adresse zu finden ist. Will später ein anderes Gerät genau diese Adresse erreichen, schlägt der Switch in der Tabelle nach und schickt den Frame direkt an den richtigen Port.

Kennt der Switch eine Ziel-Adresse noch nicht, *flutet* er den Frame: Er sendet ihn an alle Ports außer dem, über den er hereinkam. Sobald das Ziel antwortet, kennt der Switch auch dessen Port. Die Einträge bleiben nicht ewig bestehen, sondern altern nach einer gewissen Zeit ohne Verkehr aus (*Aging Timer*). So bleibt die Tabelle aktuell, auch wenn Geräte umgesteckt werden.

Broadcast-Domänen und VLANs

Weil jeder Switch-Port eine eigene Kollisionsdomäne bildet und im Vollduplex-Betrieb gleichzeitig senden und empfangen kann, gibt es auf modernen Switches keine Kollisionen mehr. Anders sieht es bei der **Broadcast-Domäne** aus: Ohne weitere Maßnahmen erstreckt sie sich über den gesamten Switch und alle damit verbundenen Switches. Jeder Broadcast erreicht dann sämtliche Geräte. Große Broadcast-Domänen bedeuten mehr Last – und mehr Risiko, weil jedes Gerät potenziell jedes andere erreichen kann.

Hier kommen **VLANs** (Virtual LANs) ins Spiel. Sie unterteilen einen physischen Switch in mehrere logische Netze, die sich gegenseitig nicht sehen – als hätte man mehrere separate Switches. Jedes VLAN ist eine eigene Broadcast-Domäne, und zwar unabhängig davon, wo die Geräte physisch stehen. So lassen sich Bereiche nach Abteilung, Funktion oder Sicherheitsbedarf trennen, etwa Verwaltung, Produktion und Gäste. Das reduziert die Broadcast-Last und isoliert die Bereiche sauber voneinander.

> **Merke**
>
> Ein VLAN ist eine eigene Broadcast-Domäne. Geräte in unterschiedlichen VLANs können nur über einen Router oder Layer-3-Switch miteinander kommunizieren.

Access-Port, Trunk-Port und 802.1Q

Damit der Switch weiß, welcher Anschluss zu welchem VLAN gehört, unterscheidet man zwei Port-Arten. Ein **Access-Port** gehört zu genau einem VLAN. An ihm hängt ein Endgerät, das von seinem VLAN nichts bemerkt: Es sendet und empfängt ganz normale Frames ohne zusätzliche Kennzeichnung. Ein **Trunk-Port** transportiert dagegen mehrere VLANs gleichzeitig über eine einzige Leitung – typischerweise zwischen zwei Switches oder vom Switch zu einem Router.

Auf dem Trunk muss jeder Frame markiert werden, damit die Gegenseite weiß, zu welchem VLAN er gehört. Dafür sorgt der Standard **IEEE 802.1Q**: Er fügt in den Frame ein 4 Byte großes *Tag* ein, das unter anderem die VLAN-ID enthält. Gültig sind die IDs 1 bis 4094. Getaggt wird nur auf Trunks; Access-Ports senden immer ungetaggt.

Eine Besonderheit ist das **Native VLAN**. Frames dieses VLANs werden auch auf dem Trunk ungetaggt übertragen; standardmäßig ist das VLAN 1. Wichtig ist, dass auf beiden Enden eines Trunks dasselbe Native VLAN eingestellt ist. Weichen die Einstellungen voneinander ab, landen Frames im falschen VLAN – ein schwer zu findender Fehler.

Port-Art	VLANs	Tagging	Typischer Einsatz
Access-Port	genau eines	ungetaggt	PC, Drucker, Telefon
Trunk-Port	mehrere	802.1Q-Tag (außer Native VLAN)	Switch zu Switch, Switch zu Router

Inter-VLAN-Routing

VLANs trennen Netze konsequent – ohne Routing bleiben sie vollständig voneinander isoliert. Soll zum Beispiel die Buchhaltung auf einen Server im Server-VLAN zugreifen, braucht es eine Verbindung auf Schicht 3. Dafür gibt es zwei verbreitete Lösungen. Beim **Router-on-a-Stick** wird ein Router über einen einzigen Trunk an den Switch angeschlossen. Auf dem Router legt man für jedes VLAN ein logisches Sub-Interface an, das als Gateway dieses VLANs dient. Die Lösung ist günstig, aber die eine Leitung kann zum Engpass werden.

Leistungsfähiger ist ein **Layer-3-Switch**. Er beherrscht neben dem Switching auch Routing und leitet Verkehr zwischen VLANs intern weiter, deutlich schneller als ein externer Router. In Unternehmensnetzen ist das heute die übliche Lösung, oft ergänzt durch eine Firewall für die Übergänge, die besonders kontrolliert werden sollen.

Managed Switches in der Praxis

Ein verwalteter (managed) Layer-3-Switch kann weit mehr als VLANs und Routing. Über *SNMP* lässt er sich ins Monitoring einbinden. Mit *PoE* (Power over Ethernet) versorgt er Access Points oder IP-Telefone direkt über das Netzwerkkabel mit Strom. *QoS* (Quality of Service) priorisiert zeitkritischen Sprach- und Videoverkehr. Mit *ACLs* (Access Control Lists) filtert er Verkehr direkt auf dem Switch. Außerdem lassen sich per *LACP* mehrere Uplinks bündeln, und beim *Stacking* werden mehrere Switches wie ein einziges logisches Gerät verwaltet.

Für die zentrale Verwaltung mehrerer Switches gibt es zwei Wege: klassisch vor Ort (on-premises) über einen eigenen Controller oder einen Stack – oder cloudbasiert über die Verwaltungsplattform des Herstellers, wie sie etwa Cisco Meraki, Aruba, Ubiquiti oder TP-Link anbieten. Cloud-Lösungen sind bequem, bringen aber meist laufende Lizenzkosten und eine feste Bindung an einen Hersteller mit sich.

Aus der Praxis

Bewährt hat sich ein VLAN je Funktion – etwa für Nutzer, Telefonie, Management und Gäste. Trennen Sie das Management-VLAN strikt vom Nutzerverkehr, nutzen Sie das Default-VLAN 1 möglichst nicht produktiv und verwenden Sie im ganzen Netz einheitliche VLAN-IDs. Die häufigsten Fehler sind verwechselte Access- und Trunk-Ports, unterschiedliche Native VLANs an beiden Trunk-Enden und ein VLAN, das in der Liste der erlaubten VLANs auf dem Trunk fehlt.

Das Wichtigste in Kürze

- Ein Switch lernt MAC-Adressen und leitet Frames gezielt weiter.

- Jedes VLAN ist eine eigene Broadcast-Domäne.
- Access-Ports gehören zu einem VLAN, Trunks tragen viele mit 802.1Q-Tag.
- Verkehr zwischen VLANs braucht Routing auf Schicht 3.

STP, RSTP & Link Aggregation

Schleifen verhindern und Leitungen bündeln

Nach diesem Kapitel können Sie …

- erklären, warum redundante Switch-Verbindungen ohne Schutz gefährlich sind,
- die Arbeitsweise von STP und die Wahl der Root-Bridge beschreiben,
- die Vorteile von RSTP gegenüber klassischem STP benennen,
- Link Aggregation mit LACP einordnen und richtig konfigurieren,
- Schutzfunktionen wie PortFast und BPDU Guard gezielt einsetzen.

Das Problem: Schleifen auf Layer 2

Ausfallsicherheit ist im Netzwerk ein zentrales Ziel. Naheliegend ist deshalb, Switches nicht nur über eine, sondern über zwei oder mehr Leitungen miteinander zu verbinden: Fällt eine aus, übernimmt die andere. Genau diese Redundanz erzeugt auf Schicht 2 aber ein ernstes Problem – eine **Schleife**.

Ein Broadcast wird von jedem Switch an alle Ports weitergeleitet. Gibt es zwei Wege zwischen zwei Switches, wandert der Broadcast über den einen Weg hin und über den anderen zurück, wird erneut weitergeleitet und so fort. Anders als IP-Pakete auf Schicht 3, die mit dem Feld TTL (Time to Live) eine Art Ablaufdatum haben, besitzen Ethernet-Frames keine solche Begrenzung. Sie kreisen endlos und vervielfachen sich dabei. Das Ergebnis ist ein **Broadcast-Storm**, der ein ganzes Netz innerhalb von Sekunden lahmlegen kann. Zusätzlich werden die MAC-Tabellen instabil, weil dieselbe Absenderadresse ständig an wechselnden Ports auftaucht.

Spanning Tree: der schleifenfreie Baum

Die Lösung für dieses Problem ist das **Spanning Tree Protocol (STP)**, standardisiert als IEEE 802.1D. Es baut aus einem vermaschten Netz mit mehreren Wegen einen logisch schleifenfreien Baum. Dazu werden überzählige, redundante Pfade nicht entfernt, sondern *logisch blockiert*: Die Kabel bleiben gesteckt, aber über die blockierten Ports fließen keine Nutzdaten. Fällt ein aktiver Pfad aus, aktiviert STP automatisch einen der Reservepfade.

Damit das funktioniert, tauschen die Switches untereinander laufend kleine Steuernachrichten aus, die **BPDUs** (Bridge Protocol Data Units). Aus diesen Informationen ermitteln sie gemeinsam die Struktur des Baums und erkennen Änderungen in der Topologie.

Root-Bridge, Port-Rollen und Zustände

Jeder Baum hat eine Wurzel, die **Root-Bridge**. Zur Root-Bridge wird der Switch mit der niedrigsten *Bridge-ID*. Diese setzt sich aus einer einstellbaren Priorität und der MAC-Adresse des Switches zusammen. Bei gleicher Priorität entscheidet also die niedrigste MAC-Adresse – und das ist oft ausgerechnet das älteste Gerät im Netz. Deshalb sollten Administratoren die Priorität gezielt setzen, damit ein zentraler, leistungsfähiger Switch die Root-Rolle übernimmt. Alle Pfade werden relativ zur Root-Bridge berechnet.

Jeder Port erhält eine Rolle. Der *Root-Port* ist bei jedem Switch der Port mit dem kürzesten Weg zur Root-Bridge. Ein *Designated-Port* ist auf jedem Netzsegment derjenige Port, der dieses Segment mit dem kürzesten Weg an die Root-Bridge anbindet und den Verkehr für das Segment weiterleitet. Alle übrigen Ports werden zu *Blocking-Ports* und verhindern die Schleife. Beim klassischen STP durchläuft ein Port beim Aktivieren mehrere Zustände: *Blocking*, *Listening*, *Learning* und schließlich *Forwarding*. Diese Abfolge ist bewusst vorsichtig, dauert aber lange: Klassisches STP benötigt etwa 30 bis 50 Sekunden, bis nach einer Änderung wieder alles läuft.

RSTP: der schnelle Nachfolger

Eine Unterbrechung von fast einer Minute ist für moderne Netze mit Telefonie und Videokonferenzen nicht akzeptabel. Deshalb wurde **Rapid Spanning Tree (RSTP)** entwickelt, standardisiert als IEEE 802.1w. RSTP konvergiert in wenigen Sekunden. Dazu führt es zusätzliche Port-Rollen ein: Ein *Alternate-Port* ist ein bereits vorberechneter Ersatzweg zur Root, ein *Backup-Port* eine Reserve für ein Segment. Statt Zeitgeber abzuwarten, verständigen sich benachbarte Switches über einen schnellen Handshake. RSTP ist heute der Standard und abwärtskompatibel zum klassischen STP.

STP absichern

An Ports, an denen nur Endgeräte hängen, gehören **PortFast** und **BPDU Guard**. PortFast schaltet den Port sofort in den Weiterleitungszustand, sodass ein PC nicht lange auf seine Verbindung warten muss. BPDU Guard sperrt den Port, sobald dort BPDUs ankommen – also jemand einen Switch angeschlossen hat. **Root Guard** verhindert zusätzlich, dass ein fremder Switch die gewählte Root-Bridge verdrängt.

Link Aggregation: Leitungen bündeln

STP sorgt für Sicherheit, verschenkt aber Kapazität: Blockierte Leitungen liegen brach. Einen anderen Weg geht die **Link Aggregation (LAG)**. Dabei werden mehrere physische Verbindungen zwischen zwei Geräten zu einer einzigen logischen Verbindung gebündelt. Für STP sieht diese Bündelung wie eine einzige Leitung aus, es muss also

nichts blockiert werden – alle Links sind aktiv. Das bringt gleichzeitig mehr Bandbreite und Ausfallsicherheit: Fällt eine Leitung aus, laufen die Daten über die verbleibenden weiter.

Ausgehandelt wird die Bündelung in der Regel mit dem **Link Aggregation Control Protocol (LACP)**, ursprünglich standardisiert als IEEE 802.3ad. LACP prüft, ob beide Seiten zueinander passen, und nimmt fehlerhafte Leitungen automatisch aus dem Bündel.

> **Merke**
>
> STP verhindert Schleifen, indem es Wege blockiert. LAG bündelt Wege, sodass alle gleichzeitig genutzt werden. Beide Verfahren ergänzen sich.

LAG richtig einrichten

Damit ein Bündel zuverlässig funktioniert, müssen beide Seiten identisch konfiguriert sein – gleiche Geschwindigkeit, gleicher Duplex-Modus, gleiche VLAN-Einstellungen. Bündeln Sie nur gleichartige Links. Bei LACP gibt es die Modi *active* (sendet aktiv Aushandlungspakete) und *passive* (antwortet nur). Mindestens eine Seite muss active sein; zwei passive Seiten finden nie zueinander.

Ein verbreitetes Missverständnis betrifft die Geschwindigkeit: Ein Bündel aus vier 1-Gbit/s-Links ergibt nicht automatisch 4 Gbit/s für eine einzelne Datei-Übertragung. Die Lastverteilung erfolgt *pro Datenfluss* anhand eines Hash-Werts, etwa aus Quell- und Zieladresse, nicht pro Paket. Eine einzelne Verbindung nutzt daher immer nur einen Link; erst viele parallele Verbindungen verteilen sich auf das ganze Bündel.

Das Wichtigste in Kürze

- Redundante Switch-Verbindungen erzeugen ohne Schutz Broadcast-Storms.
- STP verhindert Layer-2-Schleifen durch logisches Blockieren.
- Root-Bridge wird der Switch mit der niedrigsten Bridge-ID – Priorität bewusst setzen.
- RSTP konvergiert in Sekunden und ist heute Standard.
- LAG mit LACP bündelt Links für mehr Bandbreite und Redundanz.

Routing-Grundlagen

Wie Pakete den Weg zwischen Netzen finden

Nach diesem Kapitel können Sie …

- erklären, was ein Router tut und wie er sich vom Switch unterscheidet,
- eine Routing-Tabelle lesen und ihre Einträge deuten,
- statische Routen und die Default-Route sinnvoll einsetzen,
- mit Longest-Prefix-Match und administrativer Distanz die gewählte Route bestimmen,
- typische Routing-Fehler wie eine fehlende Rückroute erkennen.

Was Routing bedeutet

Ein Switch verbindet Geräte innerhalb eines Netzes auf Schicht 2. Sobald Daten jedoch ein anderes Netz erreichen sollen – den Server in der Zentrale, die Filiale oder eine Webseite im Internet –, braucht es einen **Router**. Er arbeitet auf Schicht 3, verbindet verschiedene Netze miteinander und entscheidet für jedes einzelne Paket anhand der *Ziel-IP-Adresse*, wohin es als Nächstes geschickt wird.

Auf dem Weg vom Sender zum Empfänger durchläuft ein Paket oft viele Router. Jede dieser Zwischenstationen nennt man einen **Hop**. Kein Router kennt dabei den kompletten Weg; jeder trifft nur die Entscheidung, an welchen nächsten Router er das Paket übergibt. Aus dieser Kette einzelner Entscheidungen setzt sich die gesamte Route zusammen. Nach diesem Prinzip funktioniert das gesamte Internet.

Die Routing-Tabelle

Grundlage jeder Entscheidung ist die **Routing-Tabelle**. Für jedes ankommende Paket sucht der Router darin den Eintrag, der am besten zum Ziel passt. Ein Eintrag enthält im Wesentlichen das *Zielnetz*, den *nächsten Hop* (also die Adresse des Routers, an den das Paket weitergegeben wird), die *Ausgangs-Schnittstelle* und eine *Metrik*, die beschreibt, wie „teuer“ der Weg ist.

Die Einträge stammen aus verschiedenen Quellen. Netze, die direkt an einer Schnittstelle des Routers angeschlossen sind, kennt er automatisch; sie erscheinen in der Tabelle als *connected*. Alle anderen Netze muss der Router erst kennenlernen – entweder über *statische Routen*, die ein Administrator von Hand einträgt, oder über *dynamische Routen*, die Router mithilfe von Routing-Protokollen untereinander austauschen. Wie das funktioniert, behandelt das folgende Kapitel.

```
Ziel              Nächster Hop    Interface  Herkunft
192.168.1.0/24    –               eth0       connected
10.20.0.0/16      192.168.1.254   eth0       statisch
0.0.0.0/0         203.0.113.1     eth1       statisch (Default)
```

Statische Routen und die Default-Route

Bei **statischem Routing** legt der Administrator jede Route selbst fest. Das hat Vorteile: Die Wege sind vollständig kontrollierbar und vorhersehbar, und es entsteht kein zusätzlicher Verkehr durch Routing-Protokolle. Bei vielen Netzen wird diese Methode jedoch aufwendig und fehleranfällig, weil jede Änderung an allen beteiligten Routern nachgetragen werden muss. Statische Routen eignen sich daher vor allem für kleine, stabile Umgebungen.

Eine ganz besondere statische Route ist die **Default-Route**, geschrieben `0.0.0.0/0`. Sie bedeutet sinngemäß: „Alles, wofür ich keine bessere Route kenne, schicke ich hierhin." Meist zeigt sie auf das Gateway des Internetanbieters. Ohne Default-Route müsste ein Router jedes einzelne Netz im Internet kennen – mit ihr genügt ein einziger Eintrag. In einem typischen Firmennetz kennt der Router daher nur die eigenen internen Netze und schickt alles andere über die Default-Route hinaus.

Longest-Prefix-Match

Häufig passen mehrere Einträge der Routing-Tabelle auf dieselbe Zieladresse. Dann gilt eine feste Regel, der **Longest-Prefix-Match**: Gewählt wird immer die Route mit dem längsten Präfix, also die spezifischste. Ein Beispiel: Für das Ziel `10.1.1.5` existieren die Routen `10.0.0.0/8` und `10.1.1.0/24`. Beide passen, aber die /24-Route beschreibt das Ziel genauer und gewinnt deshalb.

Die Default-Route hat die Präfixlänge 0 und ist damit die unspezifischste Route überhaupt. Sie passt zwar auf jede Adresse, wird aber nur genutzt, wenn keine andere Route zutrifft. Genau deshalb funktioniert sie als Auffangnetz für alles Unbekannte.

> **Merke**
>
> Spezifischer schlägt allgemeiner: Die Route mit dem längsten passenden Präfix gewinnt immer. Erst wenn zwei Routen gleich spezifisch sind, entscheidet die administrative Distanz.

Administrative Distanz

Ein Router kann dasselbe Zielnetz gleichzeitig aus mehreren Quellen kennen – zum Beispiel als statische Route und über ein Routing-Protokoll. Für diesen Fall bewertet er die Vertrauenswürdigkeit jeder Quelle mit der **administrativen Distanz (AD)**. Je niedriger

der Wert, desto vertrauenswürdiger die Quelle. Direkt verbundene Netze haben den Wert 0, statische Routen typischerweise 1. Routen aus dynamischen Protokollen liegen höher, etwa OSPF bei 110 und RIP bei 120. Die genauen Werte sind herstellerabhängig; die genannten sind weit verbreitet.

Die Reihenfolge der Prüfung ist dabei wichtig: Zuerst entscheidet der Longest-Prefix-Match. Nur bei gleich langen Präfixen aus unterschiedlichen Quellen kommt die administrative Distanz zum Zug. Innerhalb derselben Quelle entscheidet schließlich die Metrik.

Quelle	Typische AD	Bewertung
Direkt verbunden	0	am vertrauenswürdigsten
Statische Route	1	sehr vertrauenswürdig
OSPF	110	nachrangig
RIP	120	nachrangig

Statisch, dynamisch und die Fehlersuche

Statisches Routing ist einfach und kontrolliert, aber wartungsintensiv. Dynamisches Routing passt sich dagegen automatisch an, wenn Leitungen ausfallen oder neue Netze hinzukommen. Größere Netze kommen deshalb ohne dynamische Protokolle kaum aus, während kleine Umgebungen mit wenigen statischen Routen und einer Default-Route oft bestens bedient sind.

Bei Routing-Problemen hilft ein schrittweises Vorgehen. Prüfen Sie zunächst, ob das Zielnetz in der Routing-Tabelle überhaupt bekannt ist, und kontrollieren Sie Default-Route und Gateway. Mit `ping` testen Sie die Erreichbarkeit, mit `traceroute` (unter Windows `tracert`) sehen Sie, an welchem Hop der Weg endet.

Aus der Praxis

Ein Klassiker: Die Route zum Zielnetz ist eingetragen, trotzdem kommt keine Antwort. Häufig fehlt dann die *Rückroute* – der Router auf der Gegenseite weiß nicht, wie er die Antwort zurückschicken soll. Routing muss immer in beide Richtungen funktionieren.

Das Wichtigste in Kürze

- Router leiten Pakete auf Schicht 3 anhand der Ziel-IP-Adresse weiter.
- Die Routing-Tabelle enthält Zielnetz, nächsten Hop, Interface und Metrik.

- Der Longest-Prefix-Match bestimmt die gewählte Route.
- Die Default-Route 0.0.0.0/0 fängt alles Unbekannte ab.
- Die administrative Distanz entscheidet zwischen Quellen – und Routing braucht immer beide Richtungen.

Dynamisches Routing

OSPF, BGP und EIGRP – Routen automatisch lernen

Nach diesem Kapitel können Sie …

- erklären, wann dynamisches Routing sinnvoll ist und was es kostet,
- Distance-Vector- und Link-State-Verfahren unterscheiden,
- OSPF, BGP und EIGRP einordnen und ihren Einsatzbereich benennen,
- zentrale OSPF-Begriffe wie Area 0, Router-ID und Kosten erklären,
- Maßnahmen für ein stabiles, sicheres Routing nennen.

Warum dynamisch routen?

In einem kleinen Netz mit wenigen Standorten lassen sich alle Routen problemlos von Hand eintragen. Sobald jedoch viele Netze und mehrere mögliche Wege hinzukommen, wird statisches Routing schnell unübersichtlich und fehleranfällig. Fällt eine Leitung aus, merkt eine statische Route davon nichts – der Router schickt die Pakete weiterhin ins Leere, bis jemand die Konfiguration ändert.

Beim **dynamischen Routing** tauschen Router ihre Routing-Informationen automatisch über ein *Routing-Protokoll* aus. Sie lernen neue Netze selbstständig, erkennen Ausfälle und berechnen bei Bedarf Ersatzwege. Damit skaliert dynamisches Routing dort, wo statische Routen unpraktikabel werden. Dieser Komfort hat einen Preis: Die Protokolle benötigen etwas Rechenleistung, Arbeitsspeicher und Bandbreite für den Informationsaustausch. In modernen Netzen fällt das aber kaum ins Gewicht.

Ein wichtiger Begriff in diesem Zusammenhang ist die **Konvergenz**. Ein Netz ist konvergiert, wenn nach einer Änderung alle Router wieder ein übereinstimmendes Bild der Topologie haben. Je schneller die Konvergenz, desto kürzer sind Unterbrechungen nach einem Ausfall.

Zwei Grundprinzipien

Routing-Protokolle lassen sich nach ihrer Arbeitsweise in zwei Familien einteilen. Beim **Distance-Vector-Verfahren** kennt ein Router nur die Richtung, in die er ein Paket schicken muss, und die Entfernung zum Ziel. Diese Informationen erhält er von seinen direkten Nachbarn – gewissermaßen vom Hörensagen. Ein bekanntes Beispiel ist RIP, das die Zahl der Hops als Entfernung verwendet. Das Verfahren ist einfach, verbreitet Änderungen aber nur schrittweise von Nachbar zu Nachbar und konvergiert deshalb langsam.

Beim **Link-State-Verfahren** kennt dagegen jeder Router die komplette Topologie des Netzes: alle Router, alle Verbindungen und ihre Kosten. Aus dieser Landkarte berechnet jeder Router den besten Weg selbst. Ändert sich etwas, wird die Information sofort im Netz verteilt, und alle Router rechnen neu. Link-State-Protokolle konvergieren dadurch schneller und stabiler, benötigen aber mehr Speicher und Rechenleistung.

OSPF: der Standard im Unternehmensnetz

OSPF (Open Shortest Path First) ist das wichtigste Routing-Protokoll innerhalb von Unternehmensnetzen. Es ist ein Link-State-Protokoll und ein offener Standard, den praktisch alle Hersteller unterstützen. OSPF berechnet den kürzesten Pfad nicht nach der Zahl der Hops, sondern nach *Kosten*, die sich aus der Bandbreite der Verbindungen ergeben: Eine schnelle Leitung ist „billiger“ als eine langsame.

Damit große Netze beherrschbar bleiben, gliedert OSPF sie in **Areas**. Die zentrale Area heißt *Area 0* und bildet das Backbone; alle anderen Areas müssen daran angebunden sein. Innerhalb einer Area kennen die Router die vollständige Topologie, zwischen den Areas werden nur zusammengefasste Informationen ausgetauscht. Jeder OSPF-Router wird durch eine eindeutige *Router-ID* identifiziert. Damit Router überhaupt Informationen austauschen, müssen sie mit ihren Nachbarn eine *Nachbarschaft* (Adjacency) aufbauen. Kommt diese nicht zustande – etwa wegen unterschiedlicher Einstellungen auf beiden Seiten –, ist das eine der häufigsten Fehlerquellen.

> **Merke**
>
> OSPF ist ein Link-State-Protokoll: Jeder Router kennt die Topologie seiner Area und berechnet die Wege nach Kosten, die von der Bandbreite abhängen. Area 0 ist das Backbone, an das alle anderen Areas angeschlossen sind.

BGP: das Protokoll des Internets

Das Internet besteht aus Zehntausenden von Netzen, die von unterschiedlichen Organisationen betrieben werden – Providern, Rechenzentren, großen Unternehmen. Jedes dieser Netze bildet ein **autonomes System (AS)** mit einer eigenen Nummer. Verbunden werden diese autonomen Systeme durch das **Border Gateway Protocol (BGP)**.

BGP ist ein *Pfadvektor-Protokoll*. Es kennt für jedes Ziel die Folge der autonomen Systeme, die auf dem Weg liegen, und trifft seine Entscheidungen vor allem nach *Richtlinien* (Policies) – also nach geschäftlichen und technischen Vorgaben, etwa welcher Provider bevorzugt wird. Die reine Entfernung spielt nur eine Nebenrolle. BGP ist extrem skalierbar und trägt die Routing-Tabelle des gesamten Internets, konvergiert dafür aber vergleichsweise langsam. Im Mittelstand begegnet BGP vor allem dann, wenn ein Unternehmen über mehrere Provider gleichzeitig ans Internet angebunden ist.

EIGRP und der Vergleich

EIGRP (Enhanced Interior Gateway Routing Protocol) stammt ursprünglich vom Hersteller Cisco. Es ist eine Mischform, oft als *Advanced Distance-Vector* bezeichnet, und erreicht dank des Verfahrens *DUAL* eine sehr schnelle Konvergenz, weil es Ersatzwege bereits im Voraus berechnet. EIGRP war lange proprietär und ist inzwischen teilweise offengelegt. Verbreitet ist es vor allem in reinen Cisco-Umgebungen.

Man unterscheidet außerdem nach dem Einsatzbereich. *Interior Gateway Protocols* (IGP) wie OSPF und EIGRP arbeiten innerhalb einer Organisation. *Exterior Gateway Protocols* (EGP) wie BGP verbinden verschiedene Organisationen und Provider miteinander.

Protokoll	Verfahren	Einsatz	Besonderheit
RIP	Distance-Vector	IGP, kleine Netze	einfach, langsam
OSPF	Link-State	IGP, Unternehmensnetz	offener Standard, Areas
EIGRP	Advanced Distance-Vector	IGP, Cisco-Umgebungen	schnelle Konvergenz (DUAL)
BGP	Pfadvektor	EGP, zwischen AS	richtlinienbasiert, sehr skalierbar

Konvergenz und Stabilität im Betrieb

Ein dynamisch geroutetes Netz soll bei Änderungen schnell neu rechnen, im Normalbetrieb aber ruhig und stabil laufen. Dazu tragen einige bewährte Maßnahmen bei. Mit **Route Summarization** fasst man mehrere benachbarte Netze zu einer einzigen, kürzeren Route zusammen. Das verkleinert die Routing-Tabellen und verhindert, dass jede kleine Änderung im ganzen Netz Neuberechnungen auslöst.

An Schnittstellen, an denen keine weiteren Router hängen – etwa in Richtung der Arbeitsplätze –, sollten Sie **Passive Interfaces** konfigurieren. Das Netz wird dann zwar weiterhin bekannt gemacht, aber der Router sendet dort keine Routing-Nachrichten. Das spart Last und verhindert, dass ein unbefugtes Gerät sich als Routing-Nachbar ausgibt. Aus demselben Grund gehört die **Authentifizierung der Routing-Updates** zu einer sauberen Konfiguration: Router akzeptieren dann nur Informationen von Nachbarn, die einen gemeinsamen Schlüssel kennen.

Faustregel

OSPF im eigenen Haus, BGP nach draußen. Wer innerhalb des Unternehmens mehrere Standorte oder Wege dynamisch verbinden will, ist mit OSPF gut beraten; BGP kommt ins Spiel, sobald mehrere Provider oder fremde Netze angebunden werden.

Das Wichtigste in Kürze

- Beim dynamischen Routing lernen Router Routen selbst und reagieren auf Ausfälle.
- Link-State konvergiert schneller und stabiler als Distance-Vector.
- OSPF ist der Link-State-Standard im LAN und WAN, gegliedert in Areas.
- BGP verbindet autonome Systeme und hält das Internet zusammen.
- Summarization, Passive Interfaces und Authentifizierung sorgen für Stabilität.

NAT & PAT

Adressumsetzung zwischen privaten und öffentlichen Netzen

Nach diesem Kapitel können Sie …

- erklären, warum NAT notwendig ist und wo es stattfindet,
- Static NAT, Dynamic NAT und PAT unterscheiden,
- Port-Forwarding als Destination-NAT einrichten und absichern,
- die Grenzen von NAT benennen,
- typische NAT-Probleme gezielt eingrenzen.

Warum es NAT gibt

Als IPv4 entworfen wurde, schienen gut vier Milliarden Adressen mehr als genug. Heute ist längst klar: Öffentliche IPv4-Adressen sind knapp. Jedes Smartphone, jeder Laptop, jeder Drucker und jede Kamera bräuchte eigentlich eine eigene. Die Lösung, die sich durchgesetzt hat, heißt **Network Address Translation (NAT)**. Dabei verwenden Geräte im lokalen Netz private Adressen nach RFC 1918, und erst beim Übergang ins Internet übersetzt der Router sie in eine oder wenige öffentliche Adressen.

NAT findet meist auf dem Router oder der Firewall am Internetanschluss statt. Ein Nebeneffekt ist, dass die interne Struktur des Netzes nach außen verborgen bleibt: Von außen sieht man nur die öffentliche Adresse, nicht die einzelnen Geräte dahinter. Da von außen zunächst niemand direkt ein internes Gerät ansprechen kann, bietet NAT auch einen gewissen Schutz – eine Firewall ersetzt es aber nicht, wie weiter unten deutlich wird.

Static NAT und Dynamic NAT

Die einfachste Form ist das **Static NAT**. Dabei wird eine private Adresse fest und dauerhaft einer bestimmten öffentlichen Adresse zugeordnet – eine 1:1-Beziehung. Diese Zuordnung funktioniert in beide Richtungen: Das interne Gerät erscheint nach außen immer mit derselben öffentlichen Adresse und ist unter dieser auch von außen erreichbar. Static NAT ist vorhersehbar und eignet sich für Server, die dauerhaft erreichbar sein sollen. Der Nachteil: Für jedes Gerät wird eine eigene öffentliche Adresse verbraucht.

Beim **Dynamic NAT** verfügt der Router über einen Pool öffentlicher Adressen. Möchte ein internes Gerät ins Internet, erhält es nach dem Prinzip „wer zuerst kommt“ eine freie Adresse aus diesem Pool. Wird sie nicht mehr gebraucht, geht sie zurück in den

Pool. Auch hier gilt aber pro aktivem Gerät eine 1:1-Zuordnung. Sind alle Adressen vergeben, muss ein weiteres Gerät warten, bis wieder eine frei wird. In der Praxis spielt Dynamic NAT deshalb eine untergeordnete Rolle.

PAT: viele Geräte, eine Adresse

Der mit Abstand häufigste Fall zu Hause und in kleinen und mittleren Unternehmen ist **PAT** (Port Address Translation), auch *NAT-Overload* genannt; unter Linux spricht man von *Masquerading*. Hier teilen sich beliebig viele interne Geräte eine einzige öffentliche IP-Adresse. Unterschieden werden die einzelnen Verbindungen über die Portnummern.

Das funktioniert so: Ruft ein PC mit der Adresse `192.168.1.20` eine Webseite auf, ersetzt der Router die private Absenderadresse durch seine öffentliche Adresse und vergibt bei Bedarf einen eigenen Quellport. Diese Zuordnung aus interner Adresse und Port sowie externer Adresse und Port speichert er in seiner **Übersetzungstabelle**. Kommt die Antwort aus dem Internet zurück, schlägt der Router in der Tabelle nach und leitet sie an den richtigen internen Rechner weiter. Weil es Zehntausende Ports gibt, können sehr viele Verbindungen gleichzeitig über eine einzige Adresse laufen.

> **Merke**
>
> NAT übersetzt Adressen, PAT übersetzt Adressen und Ports. PAT erlaubt es, dass ein ganzes Firmennetz mit einer einzigen öffentlichen IPv4-Adresse auskommt.

Port-Forwarding: von außen nach innen

PAT funktioniert nur, wenn die Verbindung von innen aufgebaut wird. Soll ein interner Dienst von außen erreichbar sein – etwa ein Webserver oder ein Kamerasystem –, kennt der Router zunächst keinen passenden Eintrag und verwirft die Anfrage. Abhilfe schafft das **Port-Forwarding**, technisch ein *Destination-NAT*. Dabei legen Sie fest, dass Anfragen auf einen bestimmten öffentlichen Port an ein bestimmtes internes Gerät weitergeleitet werden, zum Beispiel: öffentlicher Port 443 an den internen Webserver `192.168.1.10:443`.

Port-Forwarding ist sicherheitsrelevant. Jeder weitergeleitete Port öffnet einen Weg aus dem Internet ins interne Netz und ist ein potenzielles Angriffsziel. Öffnen Sie daher nur die Ports, die wirklich benötigt werden, und setzen Sie die Funktion sparsam ein. Für Fernzugriffe auf interne Systeme ist ein VPN in aller Regel die sicherere Wahl.

NAT-Varianten und Grenzen

Grundsätzlich unterscheidet man zwei Richtungen. **Source-NAT** ändert die Quelladresse und ist der typische Weg von innen nach außen – PAT ist eine Form davon. **Destination-NAT** ändert die Zieladresse und ist der Weg von außen nach innen, wie beim

Port-Forwarding. Ein Sonderfall ist das *Hairpinning*: Ein internes Gerät greift über die öffentliche Adresse auf einen Dienst im eigenen Netz zu, und der Router muss den Verkehr gewissermaßen umdrehen. Nicht jeder Router beherrscht das. Manche Internetanbieter setzen zudem *Carrier-Grade-NAT* ein: Dann teilen sich mehrere Kundenanschlüsse eine öffentliche Adresse, und eigene Port-Weiterleitungen funktionieren nicht mehr ohne Weiteres.

NAT hat auch Grenzen. Es erschwert Ende-zu-Ende-Verbindungen, die manche Anwendungen benötigen, etwa Internettelefonie (VoIP) oder Peer-to-Peer-Dienste. Und vor allem ist NAT **keine vollwertige Firewall**: Es filtert keine Inhalte und setzt keine Sicherheitsregeln durch. Mit IPv6 wird NAT weitgehend überflüssig, weil genügend Adressen vorhanden sind, um jedem Gerät eine eigene öffentliche Adresse zu geben. Umso wichtiger ist dort eine sauber konfigurierte Firewall.

Aus der Praxis

Funktioniert eine Weiterleitung nicht, prüfen Sie zuerst die Übersetzungstabelle des Routers. Testen Sie Port-Forwardings immer von außen, etwa über eine Mobilfunkverbindung – aus dem internen Netz klappt der Test ohne Hairpinning oft nicht. Achten Sie darauf, dass derselbe öffentliche Port nicht doppelt vergeben ist. Protokolle wie VoIP benötigen unter Umständen ein *ALG* (Application Layer Gateway) oder einen Helper, der die Adressen auch im Inhalt der Pakete anpasst.

Das Wichtigste in Kürze

- NAT übersetzt private in öffentliche Adressen und verbirgt die interne Struktur.
- PAT teilt eine öffentliche IP über Portnummern – der Normalfall im Alltag.
- Port-Forwarding macht interne Dienste von außen erreichbar und muss sparsam eingesetzt werden.
- NAT ist keine Firewall; mit IPv6 wird es weitgehend überflüssig.

WLAN-Technologien

Standards, Frequenzen, Sicherheit und Architektur

Nach diesem Kapitel können Sie …

- die Besonderheiten von WLAN als geteiltes Medium erklären,
- die wichtigsten 802.11-Standards und Frequenzbänder einordnen,
- Kanäle und Sendeleistung sinnvoll planen,
- sichere und unsichere Verschlüsselungsverfahren unterscheiden,
- die passende Architektur für kleine und größere Umgebungen wählen.

Grundlagen drahtloser Netze

Drahtlose lokale Netze, kurz **WLAN**, basieren auf der Standardfamilie **IEEE 802.11**. Sie verbinden Endgeräte per Funk mit einem *Access Point* (AP), der wiederum am kabelgebundenen Netz hängt. Im Alltag ist WLAN so selbstverständlich geworden, dass man leicht vergisst, wie sehr es sich vom Kabelnetz unterscheidet.

Der wichtigste Unterschied: WLAN ist ein *geteiltes Medium*. Alle Geräte, die in einer Funkzelle auf demselben Kanal arbeiten, teilen sich die verfügbare Kapazität. Zudem arbeitet WLAN im *Half-Duplex*-Verfahren – ein Gerät kann nicht gleichzeitig senden und empfangen. Reichweite und Geschwindigkeit hängen außerdem stark von der Umgebung ab: Wände, Decken, Metallregale, Glasflächen und andere Funknetze dämpfen oder stören das Signal. Die auf der Verpackung angegebenen Höchstwerte werden in der Praxis daher selten erreicht.

Standards und Frequenzbänder

Die 802.11-Familie ist über die Jahre stetig gewachsen. Die klassischen Standards 802.11a, b, g und n reichen bis Wi-Fi 4 (802.11n). Heute relevant sind vor allem **Wi-Fi 5** (802.11ac), **Wi-Fi 6** und **Wi-Fi 6E** (802.11ax) sowie zunehmend **Wi-Fi 7** (802.11be). Jede Generation bringt mehr Durchsatz und vor allem eine effizientere Nutzung des Funkspektrums, was bei vielen gleichzeitig verbundenen Geräten wichtig ist. Die Standards sind abwärtskompatibel, ältere Geräte können sich also weiterhin verbinden. Allerdings bremsen langsame Clients die Funkzelle, weil sie für ihre Übertragungen mehr Sendezeit benötigen.

Ebenso wichtig wie der Standard ist das **Frequenzband**. Das *2,4-GHz-Band* hat eine größere Reichweite und durchdringt Wände besser, ist aber stark belegt – auch von Bluetooth, Mikrowellen und Nachbarnetzen – und bietet nur drei überlappungsfreie Ka-

näle: 1, 6 und 11. Das *5-GHz-Band* ist schneller, bietet deutlich mehr Kanäle und weniger Störungen, hat aber eine kürzere Reichweite. Mit Wi-Fi 6E kommt das *6-GHz-Band* hinzu, das zusätzlichen, bislang wenig genutzten Funkraum bereitstellt.

Band	Stärke	Schwäche
2,4 GHz	große Reichweite	viele Störungen, nur 3 überlappungsfreie Kanäle
5 GHz	schnell, viele Kanäle	kürzere Reichweite
6 GHz	viel freier Funkraum	nur mit Wi-Fi 6E/7, geringere Reichweite

SSID, Kanäle und Sendeleistung

Die **SSID** (Service Set Identifier) ist schlicht der Name eines Funknetzes, den Nutzer in der Liste der verfügbaren Netze sehen. Mehrere Access Points können dieselbe SSID ausstrahlen, damit Geräte zwischen ihnen wechseln können.

Entscheidend für ein stabiles WLAN ist die Kanalplanung. Benachbarte Access Points sollten auf überlappungsfreien Kanälen arbeiten, im 2,4-GHz-Band also abwechselnd auf 1, 6 und 11. Funken zu viele APs auf demselben Kanal, stören sie sich gegenseitig und müssen ständig aufeinander warten. Auch die *Kanalbreite* sollte bewusst gewählt werden: Breitere Kanäle ermöglichen mehr Durchsatz, belegen aber mehr Spektrum und sind anfälliger für Störungen. Die Sendeleistung sollte nicht maximal, sondern passend eingestellt werden. Ein zu starker AP lockt Geräte an, die eigentlich einen näheren AP nutzen sollten, und stört weiter entfernte Zellen.

WLAN-Sicherheit

Weil Funkwellen nicht an der Bürowand haltmachen, ist Verschlüsselung im WLAN unverzichtbar. Die alten Verfahren **WEP** und das erste **WPA** mit TKIP gelten als unsicher und lassen sich mit frei verfügbaren Werkzeugen brechen – sie dürfen nicht mehr eingesetzt werden. **WPA2** mit AES-Verschlüsselung war lange der Standard und ist noch weit verbreitet. Aktuell ist **WPA3**, das mit dem Verfahren *SAE* (Simultaneous Authentication of Equals) einen deutlich besseren Schutz gegen das Durchprobieren von Passwörtern bietet.

Im Unternehmen sollte zudem kein gemeinsames Passwort für alle verwendet werden. Im **Enterprise-Modus** meldet sich jeder Nutzer oder jedes Gerät nach **802.1X** individuell an, und ein *RADIUS-Server* prüft die Zugangsdaten. Verlässt jemand das Unternehmen, wird nur sein Zugang gesperrt, statt das Passwort auf allen Geräten ändern zu müssen.

> **Merke**
>
> Mindestens WPA2, besser WPA3 – niemals WEP oder WPA mit TKIP. Im Unternehmen gehört 802.1X mit RADIUS zum Standard, statt eines gemeinsamen Passworts für alle.

Architektur und Roaming

Für kleine Umgebungen genügen oft einzelne **Standalone-Access-Points**, die jeweils manuell konfiguriert werden. Sobald mehrere APs im Einsatz sind, wird das unübersichtlich: Jede Änderung muss auf jedem Gerät einzeln vorgenommen werden. Ab mehreren APs empfiehlt sich deshalb eine **zentrale Verwaltung** über einen WLAN-Controller oder eine Cloud-Plattform. Sie sorgt für einheitliche Einstellungen, erleichtert die Kanal- und Leistungsplanung und skaliert problemlos mit.

Eine zentrale Verwaltung ist auch die Voraussetzung für sauberes **Roaming**. Darunter versteht man den Wechsel eines Geräts von einem AP zum nächsten, während sich der Nutzer durch das Gebäude bewegt. Damit etwa ein Telefonat dabei nicht abbricht, muss der Wechsel schnell gehen. Standards wie *802.11r* (Fast Roaming) beschleunigen ihn, indem die Anmeldung beim neuen AP verkürzt wird.

Im Enterprise-WLAN werden außerdem häufig mehrere SSIDs angeboten, etwa für Mitarbeitende, Gäste und IoT-Geräte. Jede SSID wird auf ein eigenes VLAN abgebildet, sodass die Bereiche im kabelgebundenen Netz getrennt bleiben. Das Gäste-Netz wird zusätzlich isoliert: Mit *Client-Isolation* können sich Gäste-Geräte nicht gegenseitig erreichen, und ein *Captive Portal* zeigt vor der Nutzung eine Anmelde- oder Hinweisseite.

> **Aus der Praxis**
>
> Planen Sie vor dem Rollout eine Ausleuchtung (Site Survey). Dabei wird gemessen, wo welche Signalstärke ankommt und wo Störquellen liegen. Für reibungsloses Roaming sollten sich benachbarte Funkzellen um etwa 15 bis 20 Prozent überlappen. Ein Site Survey vor der Installation spart später viel Ärger mit Funklöchern und Verbindungsabbrüchen.

Das Wichtigste in Kürze

- WLAN basiert auf IEEE 802.11; heute relevant sind vor allem Wi-Fi 5, 6 und 6E.
- 2,4 GHz bietet Reichweite, 5 GHz Tempo, 6 GHz zusätzlichen Funkraum.
- Mindestens WPA2, besser WPA3, niemals WEP; im Unternehmen 802.1X.
- Ab mehreren Access Points zentral über Controller oder Cloud verwalten.
- Ein Site Survey vor dem Rollout verhindert Funklöcher und Störungen.

VPN-Technologien

Sichere Tunnel über unsichere Netze

Nach diesem Kapitel können Sie …

- erklären, was ein VPN leistet und wofür Unternehmen es einsetzen,
- Site-to-Site- und Remote-Access-VPNs unterscheiden,
- IPsec, OpenVPN und WireGuard einordnen und die passende Technik wählen,
- typische Fehlerursachen gezielt eingrenzen.

Was ein VPN eigentlich tut

Das Internet ist ein offenes Netz: Daten laufen über Leitungen und Router, die weder Ihnen noch Ihrem Kommunikationspartner gehören. Ein **Virtual Private Network** löst dieses Problem, indem es zwischen zwei Endpunkten einen verschlüsselten Tunnel aufbaut. Alles, was durch diesen Tunnel läuft, ist für Dritte unlesbar und lässt sich unterwegs nicht unbemerkt verändern.

Ein VPN sichert damit drei Dinge gleichzeitig ab: die *Vertraulichkeit* (niemand liest mit), die *Integrität* (niemand verändert die Daten) und die *Authentizität* (die Gegenstelle ist wirklich die, für die sie sich ausgibt). Für die Anwender fühlt sich das an, als stünden entfernte Rechner im selben lokalen Netz – der Drucker in der Filiale ist genauso erreichbar wie der Dateiserver in der Zentrale.

Zwei Einsatzformen: Standorte oder Menschen verbinden

In der Praxis begegnen Ihnen VPNs in zwei Grundformen. Beim **Site-to-Site-VPN** werden ganze Standorte dauerhaft miteinander verbunden, typischerweise von Firewall zu Firewall. Die Mitarbeitenden merken davon nichts: Ihr Rechner schickt Daten wie gewohnt ins Netz, und erst die Firewall leitet sie verschlüsselt zum anderen Standort weiter.

Beim **Remote-Access-VPN** verbindet sich dagegen ein einzelnes Gerät – etwa das Notebook im Homeoffice – mit dem Firmennetz. Dafür läuft auf dem Endgerät eine Client-Software, die den Tunnel aufbaut, sobald die Person sich anmeldet. Viele Unternehmen nutzen beide Formen parallel: Site-to-Site für die Filialen, Remote-Access für alle, die mobil arbeiten.

Die drei gängigen Techniken

IPsec ist der klassische, herstellerübergreifende Standard und die erste Wahl für Site-to-Site-Verbindungen zwischen Firewalls unterschiedlicher Hersteller. Es arbeitet auf Schicht 3 des OSI-Modells: Über das Protokoll *IKE* handeln die Gegenstellen zuerst Schlüssel aus, danach verschlüsselt *ESP* die eigentlichen Datenpakete. IPsec gilt als sehr robust, ist aber auch umfangreich in der Konfiguration – schon kleine Abweichungen in den Einstellungen beider Seiten verhindern den Verbindungsaufbau.

OpenVPN ist eine weit verbreitete Open-Source-Lösung und basiert auf TLS, also derselben Technik, die auch Webseiten per HTTPS absichert. Sein großer Vorteil ist die Flexibilität bei den Ports: OpenVPN kann zum Beispiel über TCP-Port 443 laufen und kommt damit auch durch restriktive Hotel- oder Gäste-WLANs, in denen andere VPNs blockiert werden. Auch NAT-Router auf dem Weg bereiten ihm in der Regel keine Probleme. Die Anmeldung erfolgt über Zertifikate, Benutzername und Passwort oder eine Kombination aus beidem.

WireGuard ist der jüngste der drei Ansätze und bewusst schlank gehalten. Der Code ist so kompakt, dass er sich gut prüfen lässt, und die Geschwindigkeit ist auch auf schwacher Hardware hoch. Konfiguriert wird über einfache Schlüsselpaare, ähnlich wie bei SSH. Für neue Installationen entwickelt sich WireGuard zunehmend zum Standard.

Welche Technik die richtige ist, ergibt sich meist aus dem Einsatzzweck. Sollen Firewalls verschiedener Hersteller oder ein externer Partner angebunden werden, führt an IPsec selten ein Weg vorbei, weil es praktisch jedes Gerät beherrscht. Müssen sich Mitarbeitende aus wechselnden, fremden Netzen einwählen, spielt OpenVPN seine Stärke aus. Wird eine Umgebung neu aufgebaut und unterstützen alle beteiligten Geräte WireGuard, ist es wegen seiner Einfachheit und Geschwindigkeit oft die naheliegende Wahl.

Technik	Stärke	Typischer Einsatz
IPsec	Herstellerübergreifend, robust	Standortvernetzung Firewall zu Firewall
OpenVPN	Flexibel bei Ports, kommt durch Firewalls	Remote-Access aus fremden Netzen
WireGuard	Schnell, einfach, wenig Code	Neue Installationen, mobile Geräte

Split-Tunnel oder Full-Tunnel?

Eine wichtige Entscheidung beim Remote-Access ist, welcher Datenverkehr überhaupt durch den Tunnel läuft. Beim **Split-Tunnel** geht nur der Verkehr ins Firmennetz durch das VPN; Videokonferenzen oder das allgemeine Surfen laufen direkt ins Internet. Das entlastet die Firmenleitung und ist schnell, die IT verliert aber die Kontrolle über diesen Teil des Verkehrs.

Beim **Full-Tunnel** läuft der gesamte Verkehr über das Firmennetz und damit auch durch die zentrale Firewall mit ihren Filtern. Das bringt mehr Sicherheit und einheitliche Regeln, belastet aber Internetanschluss und Firewall der Zentrale deutlich stärker.

> **Merke**
>
> Full-Tunnel bedeutet Kontrolle, Split-Tunnel bedeutet Tempo. Welche Variante passt, hängt davon ab, wie viel Bandbreite die Zentrale hat und wie streng der Datenverkehr gefiltert werden muss.

Zertifikate und Schlüssel

Wie sicher ein VPN ist, hängt stark davon ab, wie sich die Gegenstellen ausweisen. Der einfachste Weg ist ein *Pre-Shared Key* – ein gemeinsames Passwort, das auf beiden Seiten hinterlegt ist. Das ist schnell eingerichtet, aber schwer sauber zu verwalten: Verlässt jemand das Unternehmen, müsste der Schlüssel überall getauscht werden. Sicherer sind *Client-Zertifikate* aus einer internen Zertifizierungsstelle (PKI), die sich einzeln sperren lassen.

Achten Sie außerdem auf *Perfect Forward Secrecy*: Damit werden für jede Sitzung eigene Schlüssel erzeugt, sodass ein später gestohlener Schlüssel keine früher aufgezeichneten Verbindungen entschlüsselt. Und weil Zertifikate ablaufen, gehört ihr Ablaufdatum ins Monitoring – ein abgelaufenes Zertifikat ist eine der häufigsten Ursachen für plötzlich ausfallende Verbindungen.

> **Aus der Praxis**
>
> Kombinieren Sie jedes Remote-Access-VPN mit einer Multi-Faktor-Authentifizierung. Ohne zweiten Faktor genügt ein einziges erbeutetes Passwort, um aus dem Internet direkt ins Firmennetz zu gelangen.

Wenn der Tunnel nicht aufgeht

Bei IPsec hilft eine einfache Frage weiter: Scheitert die Verbindung in *Phase 1*, also beim Aufbau des gesicherten Kanals, oder erst in *Phase 2*, wenn der eigentliche Datenkanal ausgehandelt wird? In Phase 1 passen meist Schlüssel, Identitäten oder Verschlüsselungsverfahren nicht zusammen, in Phase 2 die freigegebenen Netze.

Steht der Tunnel, aber einzelne Anwendungen hängen, ist häufig die *MTU* schuld: Durch die zusätzlichen VPN-Header werden Pakete zu groß und müssen zerteilt werden. Prüfen Sie außerdem, ob die Uhrzeit auf beiden Seiten stimmt (wichtig für Zertifikate) und ob auf dem Weg ein NAT-Router sitzt – dann muss *NAT-Traversal* aktiviert sein.

Das Wichtigste in Kürze

- Ein VPN ist ein verschlüsselter Tunnel durch das Internet.
- Site-to-Site verbindet Standorte, Remote-Access verbindet einzelne Geräte.
- IPsec ist der klassische Standard, OpenVPN der flexible, WireGuard der moderne.
- Full-Tunnel für Kontrolle, Split-Tunnel für Tempo – und immer mit MFA.

Netzwerk-Diagnose & Tools

Systematische Fehlersuche mit den richtigen Werkzeugen

Nach diesem Kapitel können Sie …

- eine Netzwerkstörung systematisch von unten nach oben eingrenzen,
- die IP-Konfiguration eines Geräts prüfen und deuten,
- ping, traceroute und nslookup gezielt einsetzen,
- offene Ports und Nachbarn mit netstat, ss und arp untersuchen,
- einordnen, wann Wireshark und nmap sinnvoll sind.

Systematisch statt auf gut Glück

„Das Internet geht nicht“ – kaum eine Meldung ist im IT-Support häufiger, und kaum eine sagt weniger über die tatsächliche Ursache aus. Das Problem kann ein loses Kabel sein, ein ausgefallener DHCP-Server, ein falsch eingetragener DNS-Server oder eine Störung beim Provider. Wer hier wahllos Einstellungen ändert, verliert Zeit und schafft womöglich neue Fehler.

Bewährt hat sich ein Vorgehen entlang des OSI-Modells, **von unten nach oben**: Zuerst prüfen Sie Kabel und Verbindung, dann die IP-Konfiguration, danach die Erreichbarkeit, anschließend die Namensauflösung und erst zuletzt die Anwendung. Gehen Sie Schritt für Schritt vor und prüfen Sie zuerst das, was am wahrscheinlichsten ist. Dokumentieren Sie Ihre Ergebnisse – das hilft bei der Übergabe an Kollegen und bei wiederkehrenden Störungen.

Die IP-Konfiguration prüfen

Ist die physische Verbindung in Ordnung – leuchtet also die Link-LED am Port oder besteht eine WLAN-Verbindung –, folgt der erste Blick auf die IP-Konfiguration. Unter Windows nutzen Sie dafür `ipconfig /all`, unter Linux `ip a`. Die zentrale Frage lautet: Hat das Gerät überhaupt eine gültige IP-Adresse?

Eine Adresse aus dem Bereich `169.254.x.x` ist ein deutliches Warnsignal. Sie wird vom Betriebssystem selbst vergeben (APIPA, Automatic Private IP Addressing), wenn kein DHCP-Server erreicht wurde. Die Ursache liegt dann meist beim DHCP-Server, beim Switch-Port oder im falschen VLAN. Hat das Gerät eine plausible Adresse, prüfen Sie als Nächstes, ob Standard-Gateway und DNS-Server korrekt eingetragen sind.

Erreichbarkeit und Pfad: ping und traceroute

Mit `ping` prüfen Sie, ob ein Ziel antwortet. Das Werkzeug sendet kleine ICMP-Pakete und zeigt an, ob und wie schnell Antworten zurückkommen. Pingen Sie zuerst das eigene Gateway: Antwortet es, funktioniert das lokale Netz. Danach testen Sie ein externes Ziel, etwa einen bekannten öffentlichen DNS-Server wie `1.1.1.1`. Neben der reinen Erreichbarkeit zeigt ping auch Paketverlust und Laufzeit – beides gute Hinweise auf eine überlastete oder gestörte Verbindung.

Beachten Sie aber: Manche Geräte und Firewalls blockieren ICMP bewusst. Bleibt eine Antwort aus, heißt das also nicht zwingend, dass das Ziel offline ist.

Wo genau der Weg abbricht, zeigt `tracert` unter Windows beziehungsweise `traceroute` unter Linux. Das Werkzeug listet jeden Hop – also jeden Router – auf dem Weg zum Ziel einschließlich der Laufzeit bis dorthin. Endet die Liste schon im eigenen Netz, liegt das Problem intern; bricht sie erst hinter dem Router des Providers ab, liegt es eher außerhalb. So lassen sich Provider- und interne Probleme gut voneinander trennen.

```
ipconfig /all          (Windows)   ip a              (Linux)
ping 192.168.1.1                    ping 1.1.1.1
tracert example.com    (Windows)   traceroute example.com (Linux)
nslookup example.com               dig example.com
```

Namensauflösung: nslookup und dig

Ein besonders typisches Fehlerbild sieht so aus: Ein Ping auf eine IP-Adresse funktioniert, ein Ping auf einen Namen wie `example.com` dagegen nicht. Dann ist mit hoher Wahrscheinlichkeit die **Namensauflösung per DNS** gestört. Prüfen Sie das mit `nslookup name` oder unter Linux mit `dig name`. Beide Werkzeuge zeigen, welcher DNS-Server geantwortet hat und welche Adresse er liefert. Zum Eingrenzen hilft es, gezielt einen anderen DNS-Server abzufragen: Liefert dieser eine Antwort, liegt der Fehler beim ursprünglich eingetragenen Server.

> **Merke**
>
> Ping auf die IP-Adresse klappt, Ping auf den Namen nicht – das ist fast immer ein DNS-Problem. Die bewährte Reihenfolge lautet: ipconfig/ip, ping, tracert/traceroute, nslookup/dig.

Verbindungen, Ports und Nachbarn

Funktioniert das Netz grundsätzlich, aber ein bestimmter Dienst nicht, lohnt ein Blick auf die offenen Ports. `netstat -ano` unter Windows und `ss -tulpn` unter Linux zeigen, welche Ports geöffnet sind, welche Verbindungen bestehen und welcher Prozess dahin-

tersteckt. Das hilft etwa bei der Fehlermeldung „Port bereits belegt“, wenn ein Dienst nicht starten kann, weil ein anderes Programm seinen Port verwendet.

Für Fragen auf Schicht 2 ist `arp -a` nützlich. Es zeigt die Zuordnung von IP- zu MAC-Adressen im lokalen Netz. So lässt sich etwa erkennen, ob ein Gerät das Gateway überhaupt auf Schicht 2 erreicht oder ob zwei Geräte dieselbe IP-Adresse verwenden.

Schicht	Prüfen	Werkzeug
1	Kabel, Link-LED, Port	Sichtprüfung
2/3	IP-Konfiguration, Nachbarn, richtiges VLAN	`ipconfig` / `ip` , `arp`
3	Gateway, dann externes Ziel	`ping` , `tracert` / `traceroute`
4–7	Port offen, Namensauflösung	`ss` / `netstat` , `nslookup` / `dig`

Tiefer analysieren: Wireshark und nmap

Reichen die Standardwerkzeuge nicht aus, kommen zwei Spezialisten zum Einsatz. **Wireshark** schneidet den Netzwerkverkehr paketgenau mit und zeigt exakt, was über die Leitung geht. Es ist das Mittel der Wahl für schwierige Protokollfehler. Wichtig ist dabei der Unterschied zwischen zwei Filterarten: *Capture-Filter* begrenzen bereits die Aufzeichnung, *Display-Filter* nur die Anzeige bereits aufgezeichneter Pakete. Filtern Sie zunächst nach IP-Adresse oder Port, um die Datenmenge überschaubar zu halten. Lernen Sie, den TCP-Verbindungsaufbau zu erkennen – den Drei-Wege-Handshake aus SYN, SYN-ACK und ACK. Häufen sich *Retransmissions*, also erneut gesendete Pakete, deutet das auf Paketverluste hin.

nmap ist ein Port- und Netzwerkscanner. Es ermittelt, welche Hosts in einem Netz erreichbar sind und welche Dienste auf ihnen laufen. Das ist hilfreich für Inventarisierung und Sicherheitsprüfungen.

> **Rechtlicher Hinweis**
>
> Setzen Sie nmap nur in eigenen oder ausdrücklich freigegebenen Netzen ein. Das Scannen fremder Systeme ohne Erlaubnis kann als Angriffsvorbereitung gewertet werden. Auch Mitschnitte mit Wireshark können personenbezogene Daten enthalten und sollten nur mit klarem Auftrag erstellt werden.

Das Wichtigste in Kürze

- Fehler schichtweise von unten nach oben eingrenzen statt raten.
- Kette: ipconfig/ip, dann ping, dann tracert/traceroute, dann nslookup/dig.

- 169.254.x.x bedeutet: kein DHCP-Server erreicht.
- netstat/ss und arp klären Ports und Nachbarn im lokalen Netz.
- Wireshark und nmap dienen der Tiefenanalyse – nmap nur mit Erlaubnis.

Netzwerk-Sicherheit

Angriffe auf Layer 2 erkennen und abwehren

Nach diesem Kapitel können Sie …

- begründen, warum auch das interne Netz auf Schicht 2 geschützt werden muss,
- Zugangskontrolle mit 802.1X und NAC erklären,
- DHCP-Snooping, Dynamic ARP Inspection und Port-Security einordnen,
- typische Layer-2-Angriffe und ihre Gegenmittel benennen,
- Segmentierung und Härtung als Sicherheitsstrategie anwenden.

Der oft vergessene Layer 2

Wenn von Netzwerksicherheit die Rede ist, denken viele zuerst an die Firewall am Internetanschluss. Sie schützt den *Perimeter*, also die Grenze zwischen Firmennetz und Außenwelt. Viele Angriffe setzen aber gar nicht dort an, sondern im lokalen Netz selbst. Wer physischen Zugang zu einer Netzwerkdose hat oder sich ins WLAN einwählen kann, befindet sich bereits hinter der Firewall und kann die Switching-Ebene, also Schicht 2, direkt angreifen.

Ein solcher interner Angreifer ist kein theoretisches Szenario: Es kann ein Besucher sein, ein manipuliertes Gerät, ein Dienstleister oder ein bereits mit Schadsoftware infizierter Rechner. Deshalb braucht auch die Switching-Ebene eigene Schutzmaßnahmen. Die Grundlage bildet die konsequente Härtung der **Access-Switches**, an denen die Endgeräte angeschlossen sind.

Zugangskontrolle mit 802.1X und NAC

Die erste Verteidigungslinie ist die Frage, wer überhaupt ins Netz darf. Der Standard **IEEE 802.1X** sorgt dafür, dass ein Switch-Port gesperrt bleibt, bis sich das angeschlossene Gerät erfolgreich authentifiziert hat. Die Prüfung übernimmt ein zentraler *RADIUS-Server*, der Benutzer oder Geräte anhand von Zertifikaten oder Zugangsdaten identifiziert. Ein fremder Laptop, der einfach an eine freie Dose gesteckt wird, erhält so keinen Zugang.

Oft ist 802.1X Teil einer umfassenderen **Network Access Control (NAC)**. Eine NAC-Lösung kann zusätzlich den Zustand des Geräts bewerten – etwa ob Virenschutz und Updates aktuell sind – und das Gerät je nach Ergebnis in ein passendes VLAN schicken oder in eine Quarantäne-Zone verschieben.

DHCP-Snooping und Dynamic ARP Inspection

Ein gefährlicher, aber einfacher Angriff ist ein *Rogue-DHCP-Server*: Ein Angreifer betreibt im Netz einen eigenen DHCP-Server und verteilt an die Clients falsche Einstellungen, etwa ein manipuliertes Gateway oder einen eigenen DNS-Server. Der gesamte Verkehr läuft dann über das Gerät des Angreifers. Dagegen hilft **DHCP-Snooping**. Der Switch erlaubt DHCP-Antworten nur von Ports, die als vertrauenswürdig markiert sind – typischerweise dem Uplink zum echten DHCP-Server. Antworten von allen anderen Ports werden verworfen.

Als Nebenprodukt baut DHCP-Snooping eine *Binding-Tabelle* auf, in der festgehalten wird, welche IP-Adresse welcher MAC-Adresse an welchem Port zugewiesen wurde. Diese Tabelle ist die Grundlage für weitere Schutzfunktionen.

Eine davon ist **Dynamic ARP Inspection (DAI)**. Das Address Resolution Protocol (ARP) ordnet IP-Adressen MAC-Adressen zu und prüft dabei nicht, ob eine Antwort echt ist. Beim *ARP-Spoofing* nutzt ein Angreifer das aus und behauptet etwa, die MAC-Adresse des Gateways zu besitzen. So leitet er den Verkehr anderer Geräte über sich um – ein klassischer *Man-in-the-Middle*-Angriff. DAI prüft jedes ARP-Paket gegen die Binding-Tabelle aus dem DHCP-Snooping und verwirft gefälschte Antworten. DAI setzt deshalb voraus, dass DHCP-Snooping aktiviert ist.

> **Merke**
>
> DHCP-Snooping und Dynamic ARP Inspection gehören zusammen: Snooping blockt falsche DHCP-Server und baut die Binding-Tabelle auf, DAI nutzt diese Tabelle, um ARP-Spoofing zu verhindern.

Port-Security und typische Angriffe

Mit **Port-Security** legen Sie fest, welche und wie viele MAC-Adressen an einem Switch-Port erlaubt sind. Schließt jemand einen fremden Laptop, einen kleinen Switch oder einen Hub an, erkennt der Switch die unbekannten Adressen. Je nach Einstellung sperrt er dann den Port oder löst einen Alarm aus. Port-Security ist besonders an Ports sinnvoll, an denen Endgeräte hängen.

Daneben sollten Sie einige typische Angriffe kennen. Beim **VLAN-Hopping** versucht ein Angreifer, Zugriff auf fremde VLANs zu erlangen – etwa indem er seinem Port vorgaukelt, ein Trunk zu sein, oder Frames mit doppeltem VLAN-Tag sendet. Gegenmittel sind ein gehärtetes Native VLAN und das Abschalten der automatischen Trunk-Aushandlung an Endgeräte-Ports. Beim **MAC-Flooding** überschwemmt ein Angreifer den Switch mit gefälschten MAC-Adressen, bis dessen Tabelle voll ist; der Switch flutet dann Frames an alle Ports, und der Angreifer kann mitlesen. Bei der **DHCP-Starvation** fordert er so

viele Adressen an, bis der Pool des DHCP-Servers erschöpft ist. Mit einer **STP-Manipulation** versucht er, sich selbst zur Root-Bridge zu machen und so Verkehr über sein Gerät zu lenken.

Angriff	Wirkung	Gegenmittel
VLAN-Hopping	Zugriff auf fremde VLANs	Native VLAN härten, kein Auto-Trunk
ARP-Spoofing	Verkehr wird umgeleitet	DAI und DHCP-Snooping
MAC-Flooding	Switch flutet Frames an alle Ports	Port-Security
DHCP-Starvation / Rogue-DHCP	Adresspool leer, falsche Einstellungen	DHCP-Snooping, Port-Security
STP-Manipulation	Angreifer wird Root-Bridge	Root Guard, BPDU Guard

Segmentierung als Strategie

Selbst die beste Zugangskontrolle kann nicht verhindern, dass irgendwann ein Gerät kompromittiert wird. Entscheidend ist dann, wie weit sich der Schaden ausbreiten kann. Hier setzt die **Segmentierung** an: Ein kompromittiertes Gerät darf nicht automatisch das ganze Netz gefährden. Legen Sie IoT-Geräte wie Kameras oder Gebäudetechnik sowie Gäste-Geräte in eigene, isolierte VLANs. Trennen Sie Produktions- und Büronetz strikt voneinander. Zwischen den Segmenten sollte kein freier Durchgang bestehen, sondern eine Firewall mit klaren Regeln, die nur den wirklich benötigten Verkehr zulässt.

Netzwerk-Härtung im Alltag

Zur Grundhärtung gehören einige einfache, aber wirkungsvolle Maßnahmen. Schalten Sie unbenutzte Switch-Ports ab, damit niemand eine freie Dose nutzen kann, und deaktivieren Sie nicht benötigte Dienste auf den Netzwerkgeräten. Die Verwaltungszugänge der Switches, Router und Access Points gehören in ein eigenes Management-Netz, das vom Nutzerverkehr getrennt ist. Sichern Sie die Konfigurationen regelmäßig und versionieren Sie sie, damit sich Änderungen nachvollziehen und im Fehlerfall zurücknehmen lassen. Änderungen am Netz sollten zudem über eine geregelte Change-Kontrolle laufen.

Aus der Praxis

Ein guter erster Schritt ohne großen Aufwand: Alle nicht belegten Ports deaktivieren, PortFast mit BPDU Guard an Endgeräte-Ports setzen und DHCP-Snooping einschalten. Damit sind mehrere der häufigsten Layer-2-Angriffe bereits deutlich erschwert.

Das Wichtigste in Kürze

- Layer 2 aktiv absichern – nicht nur den Perimeter.
- 802.1X und NAC kontrollieren, wer ins Netz darf.
- DHCP-Snooping und DAI schützen vor Rogue-DHCP und ARP-Spoofing.
- Port-Security nutzen und unbenutzte Ports deaktivieren.
- Segmentierung begrenzt den Schaden, wenn doch ein Gerät kompromittiert wird.

SD-WAN & moderne Netzarchitekturen

Software-definierte Netze, Zero Trust und SASE

Nach diesem Kapitel können Sie …

- die Schwächen des klassischen WAN benennen,
- das Konzept von SD-WAN mit Underlay, Overlay und zentralem Controller erklären,
- das Prinzip Zero Trust und seine Bausteine beschreiben,
- SASE und Netzwerkvirtualisierung einordnen,
- eine schrittweise Migration zu SD-WAN skizzieren.

Das klassische WAN und seine Grenzen

Unternehmen mit mehreren Standorten müssen diese miteinander verbinden – zu einem *Wide Area Network* (WAN). Traditionell geschah das über angemietete Standleitungen, häufig auf Basis von **MPLS**, einer vom Provider betriebenen, privaten Netztechnik. Solche Leitungen sind zuverlässig, aber teuer und starr: Jeder Standort muss einzeln konfiguriert werden, und Änderungen dauern oft Wochen.

Mit dem Siegeszug von Cloud-Diensten zeigte sich eine weitere Schwäche. Im klassischen Aufbau läuft der gesamte Verkehr der Filialen über die Zentrale, wo die Firewall und der Internetanschluss sitzen. Möchte eine Mitarbeiterin in der Filiale ein Cloud-Programm nutzen, reisen die Daten erst zur Zentrale und von dort ins Internet – ein unnötiger Umweg, der Leitungen belastet und Anwendungen verlangsamt.

Das Konzept SD-WAN

SD-WAN (Software-Defined WAN) geht einen anderen Weg. Die Grundidee ist die Trennung von *Steuerung* und *Weiterleitung*: Die Steuerung – also die Entscheidung, welcher Verkehr welchen Weg nimmt – erfolgt zentral. Die Geräte an den Standorten setzen diese Vorgaben nur um. Alle Standorte werden über ein zentrales Management verwaltet, statt Gerät für Gerät konfiguriert zu werden.

Technisch unterscheidet SD-WAN zwei Ebenen. Das **Underlay** sind die physischen Leitungen: normale Internetanschlüsse, Mobilfunk wie LTE oder weiterhin vorhandene MPLS-Leitungen. Darüber legt SD-WAN ein **Overlay** aus verschlüsselten, logischen Tunneln, die alle Standorte verbinden. Ein zentraler *Controller* orchestriert dieses Overlay.

Besonders wichtig ist das **Application-aware Routing**: SD-WAN erkennt, welche Anwendung hinter einem Datenstrom steckt, und wählt für jede Anwendung automatisch den besten Pfad. Videokonferenzen laufen dann etwa über die Leitung mit der geringsten Verzögerung, große Datensicherungen über die günstigste. Fällt eine Leitung aus oder wird sie schlechter, schwenkt der Verkehr selbstständig um.

> **Merke**
>
> Underlay = physische Leitungen, Overlay = logische Tunnel darüber. SD-WAN steuert zentral, welche Anwendung welchen Weg nimmt.

Die Vorteile im Überblick

SD-WAN bringt mehrere handfeste Vorteile. Es ist meist *günstiger*, weil normale Internetleitungen für viele Zwecke ausreichen und teure Standleitungen reduziert werden können. Es ist *schneller in Richtung Cloud*, weil jeder Standort per **Local Breakout** direkt ins Internet gehen kann, statt den Umweg über die Zentrale zu nehmen. Richtlinien werden *zentral* festgelegt und gelten sofort für alle Standorte. Und neue Standorte sind *schnell in Betrieb*: Beim **Zero-Touch-Provisioning** wird das Gerät vor Ort nur angeschlossen und bezieht seine Konfiguration automatisch vom zentralen Controller – ohne dass eine IT-Fachkraft anreisen muss.

Merkmal	Klassisches WAN	SD-WAN
Leitungen	meist teure MPLS-Standleitungen	beliebige Leitungen: Internet, LTE, MPLS
Verwaltung	Gerät für Gerät	zentral über Controller
Cloud-Zugang	Umweg über die Zentrale	direkt per Local Breakout
Neue Standorte	aufwendige Einrichtung	Zero-Touch-Provisioning

Zero Trust: prüfen statt vertrauen

Klassische Netze folgen oft dem Burg-und-Graben-Prinzip: Außen steht eine starke Firewall, innen vertraut man jedem Gerät. Genau das wird zum Problem, sobald ein Angreifer einmal im Netz ist – oder wenn Mitarbeitende ohnehin von überall arbeiten. **Zero Trust** setzt dem den Grundsatz „Never trust, always verify“ entgegen: Kein Gerät und kein Nutzer gilt allein deshalb als vertrauenswürdig, weil er sich im internen Netz befindet.

Stattdessen wird *jeder Zugriff* geprüft, und zwar anhand von Identität und Kontext – also wer zugreift, von welchem Gerät, von wo und auf welche Ressource. Statt eines großen, flachen Netzes arbeitet Zero Trust mit **Mikrosegmentierung**: Systeme werden in sehr

kleine Bereiche aufgeteilt, zwischen denen nur ausdrücklich erlaubter Verkehr fließt. Grundprinzip ist **Least Privilege**: Jeder erhält nur genau die Rechte, die er für seine Aufgabe braucht.

SASE: Netz und Sicherheit aus der Cloud

Ein eng verwandtes Konzept ist **SASE** (Secure Access Service Edge). Es kombiniert SD-WAN mit Sicherheitsfunktionen, die als Cloud-Dienst bereitgestellt werden. Dazu gehören typischerweise ein *Secure Web Gateway* (SWG) zum Filtern von Webzugriffen, ein *Cloud Access Security Broker* (CASB) zur Kontrolle von Cloud-Anwendungen, *Zero Trust Network Access* (ZTNA) als Alternative zum klassischen VPN und eine *Firewall as a Service* (FWaaS).

Der Gedanke dahinter: Der Schutz soll nah am Nutzer stattfinden, egal ob dieser im Büro, im Homeoffice oder unterwegs arbeitet. Statt allen Verkehr zur zentralen Firewall zu schicken, wird er am nächstgelegenen Cloud-Knoten geprüft. Damit passt SASE gut zur heutigen Realität aus Cloud-Anwendungen und verteiltem Arbeiten.

Netzwerkvirtualisierung

Auch innerhalb von Rechenzentren werden Netze zunehmend in Software abgebildet – unabhängig von der darunterliegenden Hardware. Mit **Overlay-Netzen**, etwa auf Basis von *VXLAN*, lassen sich logische Netze über eine physische Infrastruktur legen, ähnlich wie beim SD-WAN. Lösungen wie VMware NSX nutzen das unter anderem für Mikrosegmentierung zwischen virtuellen Maschinen. Weil die Netzkonfiguration in Software vorliegt, lässt sie sich automatisieren und wie Programmcode verwalten – man spricht von *Infrastructure as Code*.

Migration in der Praxis

SD-WAN wird meist schrittweise eingeführt. Häufig betreibt man zunächst einen hybriden Aufbau aus vorhandenen MPLS-Leitungen und neuen Internetanschlüssen und baut MPLS erst ab, wenn sich das neue Overlay bewährt hat. Neue Standorte werden per Zero-Touch-Provisioning angebunden, und die zentrale Sichtbarkeit erleichtert das Monitoring. Planen Sie die Sicherheit von Anfang an mit – der Weg führt dabei oft in Richtung SASE.

Das Wichtigste in Kürze

- SD-WAN trennt zentrale Steuerung von der Weiterleitung und nutzt flexible Leitungen.
- Local Breakout bringt einen direkten, schnellen Weg in die Cloud.

- Zero Trust heißt: jeden Zugriff prüfen, segmentieren, minimale Rechte vergeben.
- SASE bündelt Netz und Sicherheit als Cloud-Dienst.
- Netzwerkvirtualisierung macht Netze software-steuerbar und automatisierbar.

LB-02

Betriebssysteme

1. OS-Geschichte & Hersteller
2. OS-Kernkomponenten & Architektur
3. Windows Desktop – Grundlagen & Administration
4. Windows Server – Grundlagen
5. DOS, CMD & klassische Befehle
6. PowerShell – Grundlagen & Praxis
7. Linux-Administration
8. Software-Deployment & Paketsysteme
9. Update- & Patch-Management
10. CVE, Zero-Day & Schwachstellen-Management
11. macOS-Administration
12. Dateisysteme

OS-Geschichte & Hersteller

Von DOS und Unix bis Windows 11, Linux und macOS

Nach diesem Kapitel können Sie …

- die DOS-basierte Windows-Linie von der NT-Linie unterscheiden,
- erklären, wie Unix, Linux und die großen Distributionsfamilien zusammenhängen,
- macOS, Android und iOS in die Unix-Familie einordnen,
- die Verbreitung der Systeme auf Desktop, Server und Mobilgeräten einschätzen,
- proprietäre und quelloffene Lizenzmodelle gegeneinander abgrenzen.

Warum sich der Blick zurück lohnt

Betriebssystemgeschichte klingt zunächst nach trockenem Stoff. Tatsächlich stammen aber viele Konzepte, mit denen Sie heute täglich arbeiten, aus den 1970er- und 1980er-Jahren: die Kommandozeile, hierarchische Dateisysteme, Benutzerkonten und Dateirechte. Sie sind alt, aber äußerst lebendig.

Wer weiß, woher ein System kommt, versteht seine Design-Entscheidungen besser – und tut sich bei der Fehlersuche leichter. Warum Windows Laufwerksbuchstaben verwendet und Linux nicht, warum unter macOS die gleichen Terminal-Befehle funktionieren wie auf einem Linux-Server: Das alles erklärt sich aus der Abstammung. Zudem sind die Generationen der Betriebssysteme und ihre Verbreitung ein klassisches Prüfungsthema.

Die DOS-Ära

Ab 1981 lieferte IBM seinen Personal Computer mit **PC-DOS** aus, Microsoft vertrieb das nahezu identische **MS-DOS**. DOS war ein reines Textsystem: Nach dem Start erschien die Eingabeaufforderung `C:\>`, und alles Weitere geschah über getippte Befehle. Es gab nur einen Benutzer und nur eine Aufgabe zur gleichen Zeit – kein Multitasking und keinen Speicherschutz. Stürzte ein Programm ab, stand meist der ganze Rechner.

Prägend war auch die berühmte Grenze von 640 KB Arbeitsspeicher, mit der Anwender und Programmierer eine ganze Generation lang kämpften. Trotz dieser Einschränkungen war DOS die Grundlage für den Siegeszug des PCs in Büros und Haushalten. Und es lebt bis heute weiter: Die Windows-Eingabeaufforderung `cmd.exe` ist ein direkter Nachfahre, viele Befehle wie `dir`, `copy` oder `cd` funktionieren dort noch genauso wie vor über vierzig Jahren.

Zwei Stammlinien bei Windows

Bei Windows lassen sich zwei Entwicklungslinien unterscheiden. Die erste begann mit **Windows 1.0** und führte über Windows 3.11 zu **Windows 95, 98 und ME**. Diese Versionen boten eine grafische Oberfläche, waren aber im Inneren noch auf DOS aufgesetzt. Ohne sauberen Speicherschutz konnte ein fehlerhaftes Programm das ganze System mitreißen – entsprechend berüchtigt war ihre Instabilität.

Parallel dazu entwickelte Microsoft die **NT-Linie**: Windows NT, dann Windows 2000, XP, 7, 10 und 11. NT brachte einen eigenen, von DOS unabhängigen Kernel mit, dazu Speicherschutz zwischen Prozessen und echten Mehrbenutzerbetrieb mit Rechten. Mit Windows XP wurde diese Linie auch für Privatanwender zum Standard; seitdem basieren alle Windows-Versionen auf NT. Windows 11 verlangt zusätzlich ein 64-Bit-System und einen TPM-2.0-Sicherheitschip.

Linie	Versionen	Merkmale
DOS-basiert	Windows 1.0 bis 3.11, 95, 98, ME	Grafisch, aber auf DOS aufgesetzt, kein sauberer Speicherschutz, instabil
NT-Linie	NT, 2000, XP, 7, 10, 11	Eigener Kernel, Speicherschutz, Mehrbenutzerbetrieb

Unix, Linux und die Distributionen

Der zweite große Stammbaum beginnt 1969 in den Bell Labs mit **Unix**. Unix führte Ideen ein, die fast alle späteren Systeme geprägt haben: ein hierarchisches Dateisystem mit einem einzigen Wurzelverzeichnis, die Darstellung von Geräten als Dateien, kleine Werkzeuge, die sich über Pipes verketten lassen, und ein durchdachtes Benutzer- und Rechtekonzept. Aus Unix gingen mehrere Zweige hervor, darunter **BSD** als zweiter großer Unix-Zweig neben den kommerziellen Varianten. Damit Programme zwischen den Unix-Varianten portierbar bleiben, definiert der **POSIX-Standard** eine gemeinsame Schnittstelle.

1991 veröffentlichte ein finnischer Informatikstudent den ersten **Linux-Kernel** – frei und quelloffen. Ein Kernel allein ist jedoch noch kein benutzbares System. Erst in Kombination mit den Werkzeugen des **GNU-Projekts** – Shell, Compiler, Dienstprogramme – entstand ein vollständiges freies Betriebssystem. Deshalb spricht man genau genommen auch von GNU/Linux.

Linux wird in Form von **Distributionen** ausgeliefert. Eine Distribution bündelt den Kernel mit Paketverwaltung, Werkzeugen und Programmen und trifft dabei wichtige Entscheidungen: welches Init-System startet, welches Paketformat verwendet wird und wie lange es Sicherheitsupdates gibt. In der Praxis begegnen Ihnen drei große Familien.

Der **Debian-Zweig** mit Debian, Ubuntu und Linux Mint verwendet den Paketmanager `apt` und ist auf Servern wie auf Desktops weit verbreitet. Der **RHEL-Zweig** rund um Red Hat Enterprise Linux, die kompatiblen Ableger Rocky Linux und AlmaLinux sowie Fedora nutzt `dnf` und ist vor allem im Enterprise-Umfeld zu Hause. Den dritten großen Zweig bilden **SUSE und openSUSE** mit dem Paketmanager `zypper`.

> **Merke**
>
> „Linux“ bezeichnet streng genommen nur den Kernel. Was Sie installieren, ist immer eine Distribution – und die bestimmt Paketmanager, Init-System und Support-Zeitraum.

macOS, Mobilgeräte und eingebettete Systeme

Auch Apples Desktop-Betriebssystem **macOS** gehört zur Unix-Familie. Seine Wurzeln liegen in NeXTSTEP, einem 1989 erschienenen System der Firma NeXT, die Apple Ende der 1990er-Jahre übernahm. Unter der Oberfläche arbeitet **Darwin**, eine Kombination aus BSD-Unix und dem Mach-Kernel. Seit 2020 setzt Apple in seinen Rechnern eigene Prozessoren mit ARM-Architektur ein, die sogenannten M-Chips (Apple Silicon).

Auf Smartphones dominieren zwei Systeme, die ebenfalls in diese Tradition gehören: **Android** basiert auf einem Linux-Kernel, **iOS** stammt wie macOS aus der Unix-Familie. In eingebetteten Geräten – Steuerungen, Messgeräten, Fahrzeugelektronik – kommen dagegen oft **Echtzeitbetriebssysteme** (RTOS) zum Einsatz, die garantieren, dass eine Reaktion innerhalb einer festen Zeitspanne erfolgt.

Wer wo dominiert

Die Verbreitung hängt stark vom Einsatzgebiet ab. Auf dem **Desktop** liegt Windows mit deutlichem Abstand vorn, macOS folgt mit einem spürbaren Anteil, Linux bewegt sich im einstelligen Prozentbereich, wächst aber langsam. Bei **Servern und in der Cloud** ist das Bild umgekehrt: Linux ist die Basis fast aller Webserver, Container und Cloud-VMs. Auf **Mobilgeräten** liegt Android weltweit vor iOS. Die genauen Marktanteile schwanken je nach Region und Erhebung; die Grundtendenz ist jedoch seit Jahren stabil.

Für Administratorinnen und Administratoren folgt daraus eine klare Konsequenz: Sie sollten sich in beiden Welten sicher bewegen. Der Arbeitsplatz läuft meist unter Windows, der Webserver, die Firewall oder der Container-Host dagegen sehr oft unter Linux.

Aus der Praxis

Viele Linux-Kenntnisse lassen sich direkt auf macOS übertragen und umgekehrt, weil beide aus der Unix-Familie stammen. Wer die Grundbefehle einer Unix-Shell beherrscht, ist damit auf Servern, Macs und Netzwerkgeräten gleichermaßen handlungsfähig.

Lizenzmodelle: proprietär und Open Source

Windows und macOS sind **proprietäre** Systeme: Der Quellcode ist nicht öffentlich, und die Nutzung ist an eine Lizenz des Herstellers gebunden. Linux und viele zugehörige Programme sind dagegen **Open Source** – der Quellcode ist frei einsehbar, darf verändert und weitergegeben werden. Innerhalb der Open-Source-Welt unterscheidet man zwei Lizenzarten: *Copyleft*-Lizenzen wie die GPL verlangen, dass veränderte Versionen wieder unter derselben Lizenz stehen; *permissive* Lizenzen wie BSD oder MIT erlauben auch die Einbindung in geschlossene Produkte.

Open Source bedeutet allerdings nicht automatisch „kostenlos im Betrieb“. Unternehmen zahlen häufig für Support, garantierte Updates und Zertifizierungen – typischerweise als jährliches Abonnement (Subscription). Bei Enterprise-Distributionen ist genau das das Geschäftsmodell.

Das Wichtigste in Kürze

- Die Windows-NT-Linie und Unix/Linux sind die zwei prägenden Stammbäume; macOS gehört zur Unix-Familie.
- DOS lebt in der Windows-Eingabeaufforderung weiter.
- Linux ist ein Kernel mit vielen Distributionen: Debian (apt), RHEL (dnf), SUSE (zypper).
- Desktop: überwiegend Windows; Server und Cloud: überwiegend Linux; mobil: Android vor iOS.
- Open Source ist quelloffen, aber nicht zwingend kostenlos im Betrieb.

OS-Kernkomponenten & Architektur

Kernel, Prozesse, Scheduling und Speicherverwaltung

Nach diesem Kapitel können Sie …

- die Aufgaben eines Betriebssystems benennen und Kernel und Userspace unterscheiden,
- monolithische, Mikro- und Hybrid-Kernel einordnen,
- Prozesse und Threads auseinanderhalten und erklären, wie der Scheduler arbeitet,
- virtuellen Speicher, Paging und Auslagerung beschreiben,
- die Stationen des Bootvorgangs in der richtigen Reihenfolge nennen.

Was ein Betriebssystem leistet

Ein Betriebssystem vermittelt zwischen der Hardware und den Anwendungen. Diese Rolle lässt sich in drei Kernaufgaben zerlegen. Die erste ist **Abstraktion**: Ein Textverarbeitungsprogramm muss nicht wissen, welche Festplatte, welche Grafikkarte oder welche Netzwerkkarte verbaut ist. Es ruft einfach „Datei speichern“ auf, und das Betriebssystem kümmert sich um den Rest.

Die zweite Aufgabe ist die **Ressourcenverwaltung**. Prozessor, Arbeitsspeicher, Datenträger und Geräte sind begrenzt und werden von vielen Programmen gleichzeitig beansprucht. Das Betriebssystem verteilt sie möglichst fair und effizient. Die dritte Aufgabe ist **Schutz und Isolation**: Programme dürfen sich nicht gegenseitig in den Speicher schreiben, und Benutzer dürfen nicht ohne Weiteres auf die Daten anderer zugreifen.

Kernel und Userspace

Das Herzstück jedes Betriebssystems ist der **Kernel**. Er läuft im privilegierten Modus des Prozessors, bei x86-Prozessoren als „Ring 0“ bezeichnet, und hat vollen Zugriff auf die Hardware. Anwendungen laufen dagegen abgeschottet im **Userspace** („Ring 3“). Dort dürfen sie weder direkt auf Geräte zugreifen noch fremden Speicher lesen.

Braucht ein Programm etwas vom System – etwa eine Datei öffnen oder ein Netzwerkpaket senden –, nutzt es einen **System Call**, einen Systemaufruf. Das ist eine fest definierte, kontrollierte Schnittstelle: Der Kernel prüft die Anfrage, führt sie aus und gibt das Ergebnis zurück. Diese Trennung ist ein zentrales Sicherheits- und Stabilitätskonzept. Stürzt eine Anwendung ab, betrifft das nur sie selbst, nicht das ganze System.

Eine wichtige Ausnahme bilden **Treiber**. Sie laufen meist im Kernel, weil sie die Hardware direkt ansprechen müssen. Ein fehlerhafter Treiber kann deshalb das gesamte System zum Absturz bringen – bei Windows etwa in Form eines Bluescreens.

Kernel-Typen

Wie viel Funktionalität im privilegierten Kern steckt, unterscheidet sich je nach Architektur. Beim **monolithischen Kernel**, wie ihn Linux verwendet, laufen Dateisysteme, Netzwerkstack und Treiber gemeinsam im Kern. Das ist schnell, weil keine aufwendigen Wechsel zwischen Kernel und Userspace nötig sind. Der Nachteil: Ein Fehler in einem Treiber trifft den Kern direkt.

Beim **Mikrokernel** steckt nur das Nötigste im Kern – etwa Speicherverwaltung, Scheduling und die Kommunikation zwischen Prozessen. Treiber und Dienste laufen als eigene Prozesse im Userspace. Das ist robuster, denn ein abgestürzter Treiber lässt sich neu starten, kostet aber durch die zusätzliche Kommunikation Leistung. Windows NT und macOS mit seinem XNU-Kernel verwenden **Hybrid-Kernel**, also Mischformen, die beide Ansätze kombinieren.

Kernel-Typ	Beispiel	Vorteil	Nachteil
Monolithisch	Linux	Schnell	Treiberfehler treffen den Kern
Mikrokernel	Forschungs- und Spezialsysteme	Robust, gute Isolation	Mehr Overhead
Hybrid	Windows NT, macOS (XNU)	Kompromiss aus beidem	Komplexe Architektur

Prozesse, Threads und Scheduling

Ein laufendes Programm wird zum **Prozess**. Jeder Prozess hat seinen eigenen, abgeschotteten Speicherbereich und eigene Ressourcen wie geöffnete Dateien. Innerhalb eines Prozesses können mehrere **Threads** laufen, also Ausführungsfäden. Threads teilen sich den Speicher ihres Prozesses und können dadurch effizient zusammenarbeiten. Ein Browser kann so etwa gleichzeitig eine Seite laden und auf Eingaben reagieren. Merksatz: Prozesse sorgen für Isolation, Threads für Parallelität innerhalb eines Programms.

Welcher Prozess wann rechnen darf, entscheidet der **Scheduler**. Er teilt die Rechenzeit in kurze *Zeitscheiben* (Time Slices) und vergibt sie nach Prioritäten an die lauffähigen Prozesse. Bei **präemptivem Multitasking** kann das Betriebssystem einen Prozess jederzeit unterbrechen, um einem anderen die CPU zu geben – kein Programm kann den Rechner blockieren. Ziel ist eine Balance aus Fairness und Reaktionsfähigkeit.

Jeder Wechsel zwischen Prozessen, ein sogenannter *Kontextwechsel*, kostet allerdings selbst etwas CPU-Zeit, weil Registerinhalte gesichert und wiederhergestellt werden müssen.

> **Merke**
>
> Auf einem einzelnen Prozessorkern läuft nie wirklich alles gleichzeitig. Der Scheduler wechselt nur so schnell zwischen den Aufgaben, dass es für Menschen gleichzeitig wirkt.

Speicherverwaltung

Jeder Prozess sieht einen eigenen **virtuellen Adressraum** – so, als hätte er den Arbeitsspeicher für sich allein. Das Betriebssystem bildet diese virtuellen Adressen mithilfe der Hardware auf den tatsächlichen physischen Arbeitsspeicher ab. Dadurch sind Programme vom echten RAM entkoppelt, und kein Prozess kann versehentlich in den Speicher eines anderen schreiben.

Verwaltet wird der Speicher in gleich großen Blöcken, den **Pages** (Seiten); das Verfahren heißt **Paging**. Reicht der physische Arbeitsspeicher nicht aus, lagert das System selten genutzte Seiten auf den Datenträger aus – unter Linux in den *Swap*-Bereich, unter Windows in die *Auslagerungsdatei*. So entsteht der Eindruck von mehr Speicher, als physisch vorhanden ist.

> **Aus der Praxis**
>
> Lagert ein Server ständig aus, weil zu wenig Arbeitsspeicher vorhanden ist, wird er quälend langsam – Datenträger sind um ein Vielfaches langsamer als RAM. Im Monitoring zeigt sich das an hoher Swap- oder Auslagerungsaktivität. Die Lösung ist dann meist schlicht mehr Arbeitsspeicher.

Dateisysteme, Treiber und Ein-/Ausgabe

Das Dateisystem legt fest, wie Daten auf einem Datenträger organisiert werden. Unter Windows ist **NTFS** Standard, für sehr große Volumes gibt es zusätzlich **ReFS**. Linux nutzt vor allem **ext4**, **XFS** und **Btrfs**, macOS verwendet **APFS**. Moderne Dateisysteme arbeiten mit *Journaling*: Änderungen werden zuerst protokolliert, sodass das Dateisystem nach einem Absturz schnell wieder in einen konsistenten Zustand kommt. Auch die Berechtigungen sind Teil des Dateisystems. Linux kann NTFS-Datenträger in der Regel problemlos lesen; Windows benötigt für ext4 dagegen Zusatzsoftware. Details dazu folgen im Kapitel über Dateisysteme.

Die Hardware spricht der Kernel über **Treiber** an. Geräte melden Ereignisse – etwa „Taste gedrückt“ oder „Daten von der Netzwerkkarte angekommen“ – über **Interrupts** an die CPU, die daraufhin die passende Routine ausführt. Unter Linux erscheinen Geräte zudem als Dateien unter `/dev`, etwa die erste Festplatte als `/dev/sda`.

Der Bootvorgang

Beim Einschalten durchläuft jeder Rechner eine feste Kette: Zuerst startet die **Firmware** – heute UEFI, bei älteren Systemen das BIOS. Sie prüft die Hardware und lädt den **Bootloader**, bei Linux meist GRUB, bei Windows den Windows Boot Manager. Der Bootloader lädt den **Kernel** in den Speicher und übergibt ihm die Kontrolle. Der Kernel initialisiert die Hardware und startet schließlich das **Init-System** – unter Linux meist systemd –, das alle weiteren Dienste startet.

Diese Kette sollten Sie kennen, denn bei Startproblemen hilft sie, den Fehler einzugrenzen: Erscheint nicht einmal das Bootmenü, liegt das Problem bei Firmware oder Bootloader; bleibt das System später hängen, eher bei Treibern oder Diensten.

Das Wichtigste in Kürze

- Der Kernel läuft privilegiert, Anwendungen isoliert im Userspace; der Übergang erfolgt über System Calls.
- Linux ist monolithisch, Windows und macOS nutzen Hybrid-Kernel.
- Prozess bedeutet Isolation, Thread bedeutet Parallelität; der Scheduler verteilt die CPU-Zeit.
- Virtueller Speicher und Paging sind das Fundament der Speicherverwaltung.
- Bootkette: Firmware → Bootloader → Kernel → Init-System.

Windows Desktop – Grundlagen & Administration

Editionen, Registry, Dienste, Ereignisprotokoll & Benutzer

Nach diesem Kapitel können Sie …

- die für Unternehmen geeigneten Windows-Editionen benennen,
- mit Registry, Diensten, Ereignisprotokoll und Task-Manager sicher umgehen,
- das Prinzip der minimalen Rechte erklären und anwenden,
- Autostart, abgesicherten Modus und Systemwiederherstellung zur Fehlerbehebung nutzen.

Die passende Edition

Windows auf dem Arbeitsplatzrechner ist das Betriebssystem, das Ihnen im Support am häufigsten begegnet. Microsoft bietet es in mehreren Editionen an, doch für Unternehmen sind vor allem zwei relevant: **Pro** und **Enterprise**. Nur sie lassen sich in eine Active-Directory-Domäne aufnehmen und zentral per Gruppenrichtlinie verwalten. Enterprise bringt darüber hinaus zusätzliche Sicherheits- und Verwaltungsfunktionen mit. Die **Home**-Edition ist nicht domänenfähig und daher eher für den privaten Einsatz gedacht.

Im Unternehmen ist heute **Windows 11** der Standard. Es setzt einen TPM-2.0-Chip und aktiviertes Secure Boot voraus – ältere Hardware erfüllt diese Anforderungen nicht immer. Der reguläre Support für Windows 10 endete am 14. Oktober 2025; Geräte, die noch damit laufen, erhalten Sicherheitsupdates nur noch über kostenpflichtige Zusatzprogramme und sollten zügig migriert werden.

> **Merke**
>
> Für Firmenrechner kommen nur Pro oder Enterprise in Frage. Ein mit Home-Edition gekauftes Gerät lässt sich nicht in die Domäne aufnehmen und muss erst auf Pro aktualisiert werden.

Die Registry

Die **Registry**, auf Deutsch Registrierungsdatenbank, ist der zentrale Speicher für fast alle Einstellungen von Windows und vielen Programmen. Sie ist als Baumstruktur aufgebaut: Oberste Ebene sind die Hauptschlüssel, etwa `HKEY_LOCAL_MACHINE` für rechnerweite

und `HKEY_CURRENT_USER` für benutzerbezogene Einstellungen. Darunter folgen Schlüssel, Unterschlüssel und schließlich die einzelnen Werte.

Bearbeitet wird die Registry mit dem Programm `regedit`. Dabei ist Vorsicht geboten: Ein falscher Eingriff kann Programme oder das ganze System lahmlegen, und es gibt keine Rückgängig-Funktion. Die goldene Regel lautet deshalb: Die Registry nur anfassen, wenn es keinen anderen Weg gibt – etwa über die Einstellungen oder eine Gruppenrichtlinie. Und vor jeder Änderung den betroffenen Zweig über „Datei → Exportieren“ sichern. Die entstehende `.reg`-Datei lässt sich bei Problemen per Doppelklick wieder einspielen.

Dienste und Autostart

Dienste (Services) sind Programme, die im Hintergrund laufen – oft bereits, bevor sich jemand anmeldet. Druckwarteschlange, Windows Update, Virenschutz oder die Netzwerkanmeldung arbeiten als Dienste. In der Dienste-Verwaltung (`services.msc`) sehen Sie, welche Dienste laufen, und legen ihren **Starttyp** fest: *automatisch* (startet mit dem System), *manuell* (startet bei Bedarf) oder *deaktiviert*.

Beachten Sie dabei die **Abhängigkeiten**: Manche Dienste benötigen andere, um zu funktionieren. Wer einen Basisdienst deaktiviert, legt unter Umständen gleich mehrere Funktionen lahm. Im Eigenschaftenfenster eines Dienstes zeigt der Reiter „Abhängigkeiten“, wovon er abhängt und welche Dienste auf ihn angewiesen sind.

Verwandt damit ist der **Autostart** von Programmen, die bei der Anmeldung starten. Zu viele Einträge verlängern den Systemstart und belegen dauerhaft Speicher und Rechenzeit. Der Task-Manager zeigt im Reiter „Autostart“ die Einträge samt ihrer Startauswirkung und erlaubt, sie zu deaktivieren. Gründlicher ist das kostenlose Microsoft-Werkzeug **Autoruns** aus der Sysinternals-Sammlung: Es findet auch versteckte Startpunkte wie geplante Aufgaben, Treiber oder Browser-Erweiterungen. Die Faustregel ist einfach: weniger Autostart, schnellerer und stabilerer Rechner.

Ereignisprotokoll und Task-Manager

Wenn etwas nicht funktioniert, ist das **Ereignisprotokoll** (Event Log, aufrufbar über `eventvwr.msc`) oft der schnellste Weg zur Ursache. Es ist gewissermaßen das Tagebuch von Windows: Fehler, Warnungen, Dienststarts und Anmeldungen werden dort mit Zeitstempel festgehalten. Die wichtigsten Protokolle sind „Anwendung“, „System“ und „Sicherheit“.

Da schnell Tausende Einträge zusammenkommen, sollten Sie gezielt filtern – nach Quelle, Ebene (etwa nur Fehler) und Zeitraum. Grenzen Sie den Zeitpunkt des Problems ein und prüfen Sie, was kurz davor protokolliert wurde. Jedes Ereignis trägt zudem eine

Ereignis-ID, die sich gezielt nachschlagen lässt. So steht etwa die ID 4625 im Sicherheitsprotokoll für eine fehlgeschlagene Anmeldung.

Für den schnellen Überblick über die aktuelle Last eignet sich der **Task-Manager** (`Strg+Umschalt+Esc`). Er zeigt, wie stark Prozessor, Arbeitsspeicher, Datenträger und Netzwerk ausgelastet sind, und welcher Prozess die Ressourcen verbraucht. Hängende Programme lassen sich dort beenden, Autostart-Einträge verwalten.

Benutzer und das Prinzip der minimalen Rechte

Auf jedem Windows-Rechner gibt es **lokale Konten und Gruppen**, etwa die Gruppen „Administratoren“ und „Benutzer“. Lokal lassen sich Konten anlegen, Kennwörter zurücksetzen und Gruppenmitgliedschaften ändern. Im Unternehmen geschieht die Benutzerverwaltung allerdings meist zentral über das **Active Directory**: Ein Konto gilt dann auf allen Rechnern der Domäne, und Rechte werden über Gruppen gesteuert. Dieses Thema wird im Lernbereich zu Serverdiensten vertieft.

Die wirksamste einzelne Schutzmaßnahme am Arbeitsplatz ist das Prinzip der **minimalen Rechte** (Least Privilege): nicht dauerhaft als Administrator arbeiten. Melden Sie sich für die tägliche Arbeit als normaler Benutzer an und heben Sie Rechte nur dann an, wenn es nötig ist – etwa über die Benutzerkontensteuerung (UAC) bei einer Installation. Schadsoftware, die unter einem Standardkonto startet, kann so deutlich weniger Schaden anrichten.

> **Aus der Praxis**
>
> Auch Administratorinnen und Administratoren sollten zwei Konten haben: ein normales für E-Mail, Internet und Büroarbeit und ein separates Admin-Konto, das nur für Verwaltungsaufgaben verwendet wird.

Abgesicherter Modus und Systemwiederherstellung

Startet Windows nicht mehr sauber oder stürzt es immer wieder ab, helfen zwei eingebaute Rettungswege. Der **abgesicherte Modus** startet Windows nur mit den nötigsten Treibern und Diensten. Läuft das System darin stabil, liegt die Ursache sehr wahrscheinlich bei einem Treiber, einem Dienst oder einem Autostart-Programm – ideal, um den Fehler einzugrenzen und das störende Element zu entfernen.

Die **Systemwiederherstellung** setzt Systemdateien, Registry und installierte Programme auf einen früheren Wiederherstellungspunkt zurück. Persönliche Dateien bleiben dabei unberührt; ein Ersatz für ein Backup ist sie deshalb nicht. Setzen Sie vor größeren Änderungen – Treiberupdates, neuer Software, Registry-Eingriffen – bewusst einen Wiederherstellungspunkt. Das ist der schnellste Rückweg, wenn etwas schiefgeht.

Das Wichtigste in Kürze

- Im Unternehmen Pro oder Enterprise einsetzen, Home ist nicht domänenfähig.
- Registry nur im Notfall bearbeiten und vorher den Zweig exportieren.
- Dienste, Ereignisprotokoll und Task-Manager sind die täglichen Werkzeuge der Fehlersuche.
- Benutzer werden im Unternehmen zentral über Active Directory verwaltet.
- Immer mit minimalen Rechten arbeiten und vor Änderungen einen Wiederherstellungspunkt setzen.

Windows Server – Grundlagen

Editionen, Rollen, Installationsvarianten und Verwaltung

Nach diesem Kapitel können Sie …

- die Editionen Standard und Datacenter unterscheiden und die Grundzüge der Lizenzierung erklären,
- einschätzen, wann Device-CALs und wann User-CALs sinnvoll sind,
- Rollen und Features unterscheiden und Server Core als Installationsvariante begründen,
- die wichtigsten Verwaltungswerkzeuge und das LTSC-Release-Modell einordnen,
- grundlegende Betriebsregeln für Server benennen.

Windows Server im Überblick

Windows Server ist das Server-Betriebssystem von Microsoft und stellt in vielen Unternehmen die zentralen IT-Dienste bereit: Active Directory für Benutzer und Rechner, DNS für die Namensauflösung, DHCP für die automatische Adressvergabe sowie Datei- und Druckdienste. Technisch teilt es sich viel Code mit dem Desktop-Windows, ist aber auf Dauerbetrieb, viele gleichzeitige Zugriffe und Stabilität über viele Jahre ausgelegt.

Ein Desktop-Windows könnte einige Serveraufgaben technisch zwar übernehmen, ist dafür aber weder gebaut noch lizenziert. Windows Server bringt die nötigen Serverrollen mit, ist für den Hintergrundbetrieb abgestimmt und wird nach einem eigenen Modell lizenziert. Neue Hauptversionen erscheinen etwa alle drei Jahre, zuletzt Windows Server 2019, 2022 und 2025.

Editionen und Lizenzierung

Im Kern gibt es zwei Editionen. **Standard** eignet sich für physische Server und Umgebungen mit wenigen virtuellen Maschinen: Eine vollständig lizenzierte Standard-Edition berechtigt zum Betrieb von zwei VMs auf dem lizenzierten Host. **Datacenter** ist für hochvirtualisierte Umgebungen gedacht und erlaubt unbegrenzt viele VMs. Zusätzlich enthält sie Funktionen wie *Storage Spaces Direct*, mit dem sich aus den lokalen Datenträgern mehrerer Server ein gemeinsamer, ausfallsicherer Speicher bilden lässt.

Lizenziert wird nach **physischen CPU-Kernen**. Dabei gilt eine Mindestmenge: Pro Server sind mindestens 16 Kern-Lizenzen zu erwerben, auch wenn weniger Kerne verbaut sind. Hinzu kommen **CALs** (Client Access Licenses), die jeden Zugriff auf den Server abdecken – unabhängig davon, wie viele Server es gibt. Über eine optionale *Software As-*

surance sichern sich Unternehmen zusätzlich das Recht auf neue Versionen und weitere Nutzungsrechte. Weil sich die Lizenzbedingungen im Detail ändern können, sollten Sie bei konkreten Projekten stets die aktuellen Bestimmungen des Herstellers oder eines Lizenzpartners heranziehen.

Device-CAL oder User-CAL?

CALs gibt es in zwei Varianten, und die Wahl hat spürbare Auswirkungen auf die Kosten. Eine **Device-CAL** lizenziert ein Gerät – egal, wie viele Personen es nutzen. Sie lohnt sich bei Schichtbetrieb: Ein Kiosk-PC im Lager, an dem sich mehrere Mitarbeitende pro Tag abwechseln, braucht nur eine einzige Lizenz statt einer pro Person.

Eine **User-CAL** lizenziert dagegen eine Person – egal, mit wie vielen Geräten sie zugreift. Sie lohnt sich zum Beispiel im Homeoffice: Wer mit Laptop, Tablet und Diensthandy auf den Server zugreift, benötigt eine Lizenz statt drei. Beide Varianten dürfen gemischt werden, etwa User-CALs für das Außendienstteam und Device-CALs für gemeinsam genutzte PCs. Wichtig: Für den Zugriff über Remote Desktop Services ist eine **RDS-CAL** erforderlich – zusätzlich zur normalen CAL. Dieses Thema wird im Lernbereich zu Serverdiensten vertieft.

Variante	Lizenziert	Lohnt sich bei
Device-CAL	Ein Gerät, beliebig viele Personen	Schichtbetrieb, gemeinsam genutzten PCs
User-CAL	Eine Person, beliebig viele Geräte	Homeoffice, Außendienst, mehreren Geräten pro Person

Ein ähnliches Prinzip gilt beim Datenbankserver **SQL Server**. Hier gibt es zwei Modelle: Beim *Per-Core*-Modell wird nach CPU-Kernen lizenziert, und beliebig viele Nutzer und Geräte dürfen zugreifen – das ist gut planbar bei großer oder schwankender Nutzerzahl. Beim Modell *Server + CAL* kauft man eine Server-Lizenz und zusätzlich eine CAL pro Nutzer oder Gerät; diese Variante gibt es nur für die Standard Edition. Bei wenigen Nutzern ist Server + CAL meist günstiger, ab einer gewissen Nutzerzahl kippt die Rechnung zugunsten von Per-Core. Wo genau diese Schwelle liegt, hängt vom Szenario und den aktuellen Preisen ab und sollte jeweils neu durchgerechnet werden.

Rollen und Features

Windows Server installiert nicht automatisch alle Funktionen, sondern nur das, was Sie ausdrücklich auswählen. Die Hauptaufgaben heißen **Rollen**, zum Beispiel AD DS (Active Directory Domain Services), DNS, DHCP, Datei- und Druckdienste, der Webserver IIS oder die Virtualisierungsplattform Hyper-V. **Features** sind ergänzende Bausteine wie das .NET Framework oder das Failover-Clustering. Installiert wird beides über den Server-Manager oder per PowerShell, etwa mit `Install-WindowsFeature DHCP -IncludeManagementTools`.

AD, DNS und DHCP bilden zusammen das Fundament einer Windows-Umgebung; Datei- und Druckdienste bauen darauf auf. Trotzdem gilt: Jede zusätzliche Rolle vergrößert die Angriffsfläche. Ein Server sollte daher möglichst wenige Rollen tragen. Das sorgt für klarere Zuständigkeiten, und ein Problem mit einem Dienst reißt nicht gleich alle anderen mit. Wo es die Ressourcen erlauben, werden die Rollen auf mehrere Server – oft virtuelle Maschinen – verteilt.

> **Merke**
>
> Ein Server macht nur, was er soll: so wenige Rollen wie möglich pro Server. Das verkleinert die Angriffsfläche und vereinfacht Wartung und Fehlersuche.

Installationsvarianten und Verwaltung

Bei der Installation wählen Sie zwischen zwei Varianten. Mit **Desktop Experience** erhalten Sie die gewohnte grafische Oberfläche. Das ist einfacher zu bedienen, bringt aber mehr installierte Komponenten mit – und damit mehr Angriffsfläche und mehr Updates. **Server Core** verzichtet auf die grafische Oberfläche; es gibt nur eine Konsole, verwaltet wird überwiegend aus der Ferne. Server Core ist schlanker, sicherer und braucht seltener einen Neustart. Für Infrastrukturrollen wie Domänencontroller, DNS oder DHCP wird es ausdrücklich empfohlen.

Für die Verwaltung stehen mehrere Werkzeuge bereit. Der **Server-Manager** zeigt lokale und entfernte Server in einer Übersicht. Das **Windows Admin Center** ist eine moderne, browserbasierte Oberfläche, die sich besonders für Server Core eignet. Für skalierbare Automatisierung ist **PowerShell mit Remoting** das Mittel der Wahl: Ein Befehl kann damit auf Dutzenden Servern gleichzeitig laufen. Die Konfiguration vieler Rechner und Server im Verbund erfolgt schließlich über **Gruppenrichtlinien** (GPOs).

Release-Modelle

Für klassische Server ist der **Long-Term Servicing Channel (LTSC)** der Standard. Jede LTSC-Version erhält fünf Jahre allgemeinen und fünf weitere Jahre erweiterten Support, also insgesamt rund zehn Jahre Sicherheitsupdates. Das ermöglicht einen planbaren Langzeitbetrieb ohne häufige Versionswechsel. Daneben bietet Microsoft Varianten mit engerer Cloud-Anbindung an, etwa eine Azure Edition, sowie **Hotpatching**, bei dem sich viele Sicherheitsupdates ohne Neustart einspielen lassen. Diese Optionen setzen jedoch bestimmte Voraussetzungen und Lizenzen voraus.

Best Practices im Betrieb

Einige Regeln gelten für jeden Server, unabhängig von seiner Rolle. Arbeiten Sie mit **dedizierten Admin-Konten** und nutzen Sie kein Domänen-Admin-Konto für den Alltag. Spielen Sie **Updates regelmäßig** in festgelegten Wartungsfenstern ein. Sorgen Sie für **getestete Backups** – ein Backup, dessen Wiederherstellung nie geprüft wurde, ist im Ernstfall wertlos – und dokumentieren Sie die Konfiguration. Ein **Monitoring** meldet Probleme, bevor die Anwender sie bemerken.

Aus der Praxis

Kein Server ohne Wiederherstellungsplan: Halten Sie schriftlich fest, wie ein ausgefallener Server neu aufgebaut oder aus dem Backup zurückgeholt wird, und spielen Sie den Ablauf mindestens einmal tatsächlich durch.

Das Wichtigste in Kürze

- Standard oder Datacenter je nach Virtualisierungsgrad; lizenziert wird nach Kernen plus CALs.
- Device-CAL für geteilte Geräte, User-CAL für Personen mit mehreren Geräten.
- Nur benötigte Rollen installieren und auf mehrere Server verteilen.
- Server Core mit Remote-Verwaltung als sichere Basis.
- LTSC für planbaren Langzeitbetrieb, dazu Updates, getestete Backups und Monitoring.

Hinweis: Lizenzbedingungen und Preise ändern sich. Die beschriebenen Modelle geben das Grundprinzip wieder; für konkrete Beschaffungen gelten die aktuellen Herstellerbedingungen.

DOS, CMD & klassische Befehle

Die Kommandozeile für schnelle Diagnose und Wartung

Nach diesem Kapitel können Sie …

- sich in der Eingabeaufforderung bewegen und Dateien verwalten,
- Netzwerkprobleme mit der Befehlskette ipconfig, ping, tracert und nslookup eingrenzen,
- Prozesse, Dienste und Benutzer per Kommandozeile steuern,
- beschädigte Systeme mit sfc, DISM und chkdsk reparieren,
- einfache Batch-Dateien mit Variablen, Pipes und Umleitungen schreiben.

Von DOS zur cmd.exe

Die Windows-Eingabeaufforderung `cmd.exe` ist der direkte Nachfahre von MS-DOS. Viele halten sie für ein Relikt – zu Unrecht. Für zahlreiche Prüfungen ist sie auch heute noch der schnellste Weg: Statt sich durch mehrere Fenster zu klicken, genügen wenige getippte Befehle, um etwa die Netzwerkkonfiguration zu prüfen. Sie läuft überall, auch auf einem Windows Server Core ohne grafische Oberfläche, und sie ist die Grundlage der Batch-Automatisierung mit `.bat`- und `.cmd`-Dateien.

Alle Befehle bieten mit dem Schalter `/?` eine eingebaute Hilfe, etwa `robocopy /?`. Schalter werden in der CMD traditionell mit einem Schrägstrich eingeleitet; manche Netzwerkbefehle wie `netstat` verwenden stattdessen einen Bindestrich.

Navigation und Dateiverwaltung

Mit `dir` listen Sie den Inhalt des aktuellen Verzeichnisses auf; `dir /a` zeigt auch versteckte und Systemdateien. Mit `cd` (change directory) wechseln Sie in einen Ordner, mit `cd ..` eine Ebene nach oben. Der Befehl `tree` stellt die gesamte Ordnerstruktur als Baum dar, und `cls` leert den Bildschirm.

Für die Arbeit mit Dateien gibt es die Grundbefehle `copy` (kopieren), `move` (verschieben), `del` (löschen) und `ren` (umbenennen). Verzeichnisse legen Sie mit `mkdir` an und entfernen sie mit `rmdir`. Für ganze Ordnerbäume gibt es `xcopy`.

Für ernsthafte Kopierjobs ist jedoch **robocopy** („robust copy“) das Werkzeug der Wahl. Es setzt abgebrochene Übertragungen fort, wiederholt fehlgeschlagene Dateien automatisch und kann Berechtigungen mitkopieren. Mit dem Schalter `/MIR` spiegelt es ein Verzeichnis komplett – das Ziel ist danach eine exakte Kopie der Quelle. Für Datenumzüge und einfache Sicherungen ist das deutlich zuverlässiger als ein simples `copy`.

```
robocopy D:\Daten \\nas01\backup\Daten /MIR /R:3 /W:5 /LOG:C:\Logs\backup.log
```

> **Vorsicht bei /MIR**
>
> Spiegeln bedeutet auch löschen: Dateien, die in der Quelle fehlen, werden im Ziel entfernt. Wer Quelle und Ziel verwechselt, überschreibt seine Daten. Prüfen Sie die Reihenfolge vor dem Start genau.

Netzwerk-Diagnose

Bei Netzwerkproblemen hat sich eine feste Reihenfolge bewährt, die fast jeden Fehler eingrenzt. Zuerst zeigt `ipconfig /all` die eigene Konfiguration: IP-Adresse, Subnetzmaske, Gateway, DNS-Server und ob die Adresse per DHCP bezogen wurde. Anschließend prüft `ping`, ob ein Ziel überhaupt antwortet – zuerst das Gateway, dann ein Ziel im Internet. Antwortet ein entferntes Ziel nicht, verfolgt `tracert` den Weg dorthin Station für Station und zeigt, an welchem Router die Verbindung abbricht. Schließlich testet `nslookup` die Namensauflösung über DNS.

Ein typisches Muster: Funktioniert `ping` auf eine IP-Adresse, aber nicht auf einen Namen, liegt das Problem sehr wahrscheinlich beim DNS und nicht an der Verbindung selbst. Bei Problemen mit DHCP geben `ipconfig /release` und `ipconfig /renew` die Adresse frei und fordern eine neue an.

Ergänzend zeigt `netstat -ano` alle aktiven Verbindungen und lauschenden Ports zusammen mit der Prozess-ID (PID). Das ist wertvoll, wenn eine Anwendung meldet, ein Port sei bereits belegt. Mit `arp -a` lässt sich außerdem die ARP-Tabelle anzeigen, also die Zuordnung von IP- zu MAC-Adressen im lokalen Netz.

Schritt	Befehl	Klärt die Frage
1	`ipconfig /all`	Ist die eigene Konfiguration korrekt?
2	`ping`	Antwortet das Ziel?
3	`tracert`	Wo auf dem Weg bricht es ab?
4	`nslookup`	Funktioniert die Namensauflösung?

System, Dienste und Benutzer

`tasklist` zeigt alle laufenden Prozesse mit ihrer PID, `taskkill /PID 1234` beendet einen bestimmten Prozess; mit `/F` wird das Beenden erzwungen. Dienste fragen Sie mit `sc query` ab und starten oder stoppen sie mit `sc start` bzw. `sc stop` und dem Dienstnamen. Der vielseitige Befehl `net` verwaltet mit `net user` lokale Benutzerkonten und ver-

bindet mit `net use` Netzlaufwerke. Einen schnellen Systemüberblick mit Betriebssystemversion, Installationsdatum, Arbeitsspeicher und installierten Updates liefert `systeminfo`.

Wartung und Reparatur

Wenn Windows selbst Probleme macht, gibt es ein Reparatur-Trio, das Sie kennen sollten. `sfc /scannow` (System File Checker) prüft alle geschützten Systemdateien und ersetzt beschädigte durch intakte Kopien. Kann sfc nicht reparieren, weil auch der lokale Vorrat an Ersatzdateien beschädigt ist, hilft `DISM /Online /Cleanup-Image /RestoreHealth`: Es repariert das Windows-Abbild selbst, bei Bedarf mit Dateien aus Windows Update. Danach lohnt ein erneuter Lauf von sfc. `chkdsk /f` schließlich prüft das Dateisystem auf Fehler und korrigiert sie; beim Systemlaufwerk geschieht das beim nächsten Neustart.

Ein weiterer Klassiker im Unternehmensnetz ist `gpupdate /force`. Er wendet alle Gruppenrichtlinien sofort neu an, statt auf das nächste automatische Intervall zu warten – praktisch, wenn eine neue Richtlinie gleich getestet werden soll.

> **Aus der Praxis**
>
> Alle drei Reparaturbefehle benötigen eine Eingabeaufforderung mit Administratorrechten („Als Administrator ausführen“). Ohne erhöhte Rechte brechen sie mit einer Fehlermeldung ab.

Batch-Grundlagen, Variablen und Pfade

Mehrere Befehle lassen sich in einer Textdatei mit der Endung `.bat` oder `.cmd` bündeln und als Skript ausführen. Die erste Zeile lautet meist `@echo off`; sie verhindert, dass jeder Befehl vor seiner Ausführung noch einmal angezeigt wird. Variablen werden in Prozentzeichen geschrieben, etwa `%name%`; an das Skript übergebene Argumente stehen in `%1`, `%2` und so weiter. Für einfache Logik gibt es `if`, Schleifen mit `for` und Sprünge mit `goto`.

Mit `set` zeigen und setzen Sie Umgebungsvariablen. Besonders wichtig ist `%PATH%`: Diese Variable enthält die Liste der Ordner, in denen Windows nach ausführbaren Programmen sucht. Ein Befehl wird nur gefunden, wenn er in einem dieser Ordner liegt oder mit vollem Pfad aufgerufen wird. Wo ein Befehl tatsächlich liegt, verrät `where`, etwa `where ping`.

Für alles, was über einfache Abläufe hinausgeht, ist PowerShell die bessere Wahl. Sie wird im folgenden Kapitel behandelt.

Pipes und Umleitung

Die Ausgabe eines Befehls lässt sich weiterverarbeiten. Mit dem senkrechten Strich `|` (Pipe) wird sie an den nächsten Befehl weitergegeben, mit `>` in eine Datei geschrieben – eine vorhandene Datei wird dabei überschrieben. Mit `>>` wird die Ausgabe an das Ende einer Datei angehängt. Mit `findstr` durchsuchen Sie Ausgaben nach einem Begriff:

```
netstat -ano | findstr :443
ipconfig /all > C:\Temp\netz.txt
systeminfo >> C:\Temp\netz.txt
```

Das Wichtigste in Kürze

- Die CMD ist nicht tot – für schnelle Diagnose und einfache Skripte oft der kürzeste Weg.
- Netzwerk-Kette: `ipconfig` → `ping` → `tracert` → `nslookup`.
- `robocopy` statt `copy` für ernsthafte Kopierjobs, `/MIR` mit Bedacht.
- `sfc`, `DISM` und `chkdsk` bilden das Reparatur-Trio.
- Mit `|`, `>` und `>>` werden Ausgaben verkettet und gespeichert.

PowerShell – Grundlagen & Praxis

Objektorientierte Automatisierung für Windows und mehr

Nach diesem Kapitel können Sie …

- erklären, warum PowerShell mit Objekten statt mit Text arbeitet,
- Cmdlets nach dem Verb-Substantiv-Schema finden und die eingebaute Hilfe nutzen,
- Daten in der Pipeline filtern, sortieren, formatieren und exportieren,
- Variablen, Vergleichsoperatoren und Fehlerbehandlung in Skripten einsetzen,
- Skripte sicher testen und sauber strukturieren.

Was PowerShell ausmacht

PowerShell ist Shell und Skriptsprache zugleich und das Standardwerkzeug für die Automatisierung von Verwaltungsaufgaben unter Windows. Sie ist tief in Windows und in Microsoft-Produkte wie Active Directory, Exchange oder Microsoft 365 integriert. Seit Version 7 ist PowerShell zudem plattformübergreifend und läuft auch unter Linux und macOS.

Der entscheidende Unterschied zur klassischen Eingabeaufforderung liegt darin, was zwischen den Befehlen weitergereicht wird. In der CMD fließt reiner Text: Wer etwa nur die Prozess-ID aus einer Ausgabe braucht, muss sie mühsam aus den Textzeilen herausschneiden. In PowerShell fließen dagegen echte **Objekte** durch die Pipeline. Ein Dienst ist ein Objekt mit Eigenschaften wie `Name`, `Status` und `StartType`, auf die Sie direkt zugreifen können. Das macht Filtern, Sortieren und Weiterverarbeiten zuverlässig und gut lesbar.

Cmdlets und das Verb-Substantiv-Schema

Die Befehle in PowerShell heißen **Cmdlets** (gesprochen „Commandlets“) und folgen einem festen Muster: *Verb-Substantiv*. Das Verb beschreibt die Aktion, das Substantiv das Objekt. `Get` fragt ab, `Set` ändert, `New` erstellt, `Remove` entfernt, `Start` und `Stop` starten und beenden. So ergeben sich Befehle wie `Get-Service`, `Get-Process`, `New-Item` oder `Set-Content`. Dieses Schema ist sehr vorhersehbar: Wer `Get-Service` kennt, kann `Stop-Service` und `Start-Service` erraten.

Drei Cmdlets sind am Anfang besonders hilfreich. `Get-Command` findet Befehle, etwa alle, die mit Diensten zu tun haben: `Get-Command *Service*`. `Get-Help` liefert die Hilfe zu einem Befehl, mit `-Examples` sogar direkt einsetzbare Beispiele. Und `Get-Member` zeigt,

welche Eigenschaften und Methoden ein Objekt besitzt – unverzichtbar, um zu wissen, wonach Sie filtern können.

> **Merke**
>
> `Get-Command`, `Get-Help` und `Get-Member` sind die drei wichtigsten Werkzeuge zum Lernen: Befehl finden, Hilfe lesen, Objekt untersuchen.

Die Pipeline: Filtern, Auswählen, Ausgeben

Mit dem senkrechten Strich `|` geben Sie Objekte an das nächste Cmdlet weiter. Ihre Eigenschaften bleiben dabei vollständig erhalten. Das folgende Beispiel holt alle Dienste, behält nur die laufenden und sortiert sie nach Namen:

```
Get-Service | Where-Object Status -eq 'Running' | Sort-Object Name
```

Für die Arbeit in der Pipeline gibt es eine Reihe von Standard-Cmdlets. `Where-Object` filtert Objekte nach einer Bedingung. `ForEach-Object` führt eine Aktion für jedes einzelne Objekt aus. `Select-Object` wählt bestimmte Eigenschaften aus oder nur die ersten Objekte, etwa mit `-First 5`. `Sort-Object` sortiert, `Group-Object` gruppiert nach einer Eigenschaft, und `Measure-Object` zählt oder berechnet Summen und Durchschnitte.

Für die Anzeige formatieren `Format-Table` und `Format-List` die Ausgabe als Tabelle oder Liste. Zum Speichern schreibt `Out-File` die Ausgabe in eine Textdatei, `Export-Csv` erzeugt eine CSV-Datei, die sich etwa in einer Tabellenkalkulation weiterverarbeiten lässt. Wichtig: Format-Cmdlets gehören immer ans Ende einer Pipeline, denn danach liegen nur noch Anzeigedaten vor, keine verwertbaren Objekte mehr.

Variablen und Operatoren

Variablen beginnen in PowerShell mit einem Dollarzeichen, zum Beispiel `$name = 'Server01'`. Arrays, also Listen mehrerer Werte, entstehen einfach durch Kommas: `$liste = 1,2,3`. In Skriptblöcken steht `$_` für das jeweils aktuelle Objekt in der Pipeline.

Für Vergleiche verwendet PowerShell keine mathematischen Zeichen wie `=` oder `>`, sondern Kürzel mit Bindestrich: `-eq` (gleich), `-ne` (ungleich), `-gt` (größer als), `-lt` (kleiner als), `-like` für Mustervergleiche mit Platzhaltern wie `*` und `-match` für reguläre Ausdrücke. Logische Verknüpfungen erfolgen mit `-and`, `-or` und `-not`. Das Zeichen `>` ist dagegen – wie in der CMD – für die Umleitung in Dateien reserviert.

Fehlerbehandlung

Robuste Skripte fangen Fehler kontrolliert ab, statt einfach weiterzulaufen. Dafür gibt es die Struktur `try`, `catch` und `finally`: Im `try`-Block steht der eigentliche Code, der `catch`-Block reagiert auf einen Fehler, und `finally` wird in jedem Fall ausgeführt, etwa um Verbindungen zu schließen.

```
try {
    Get-Item 'C:\Daten\bericht.xlsx' -ErrorAction Stop
} catch {
    Write-Warning "Datei nicht gefunden: $($_.Exception.Message)"
}
```

Wichtig ist der Parameter `-ErrorAction Stop`: Viele Fehler sind in PowerShell standardmäßig „nicht abbrechend“ und würden vom `catch`-Block nicht erfasst. Erst mit `Stop` werden sie abfangbar. Die automatische Variable `$Error` enthält die zuletzt aufgetretenen Fehler. Für Statusmeldungen eignen sich `Write-Verbose` (nur sichtbar mit dem Schalter `-Verbose`) und `Write-Warning`.

Wichtige Bausteine für Administratoren

Viele Funktionen werden über **Module** nachgerüstet. Das Modul *ActiveDirectory* verwaltet Benutzer und Gruppen, *ImportExcel* erstellt Excel-Dateien ohne installiertes Office, *PSWindowsUpdate* steuert Updates. Über **PowerShell-Remoting**, das auf WinRM basiert, verwalten Sie entfernte Rechner: `Enter-PSSession` öffnet eine interaktive Sitzung auf einem anderen Rechner, `Invoke-Command` führt Befehle auf einem oder vielen Rechnern gleichzeitig aus.

Die **Ausführungsrichtlinie** legt fest, welche Skripte überhaupt starten dürfen; mit `Get-ExecutionPolicy` und `Set-ExecutionPolicy` wird sie abgefragt und geändert. Sie ist eher ein Schutz vor versehentlichem Ausführen als eine echte Sicherheitsgrenze. Skripte werden als `.ps1`-Dateien gespeichert.

Ein Praxisbeispiel und saubere Skripte

Ein typischer Anwendungsfall: Nach einem Neustart sollen alle Dienste, die automatisch starten müssten, aber gestoppt sind, wieder gestartet werden. In PowerShell gelingt das in einer Zeile:

```
Get-Service | Where-Object {$_.StartType -eq 'Automatic' -and $_.Status -eq 'Stopped'}
| Start-Service -WhatIf
```

Der Schalter `-WhatIf` zeigt nur an, was passieren würde, ohne es tatsächlich auszuführen. Erst wenn das Ergebnis passt, entfernen Sie den Schalter. So werden aus zwanzig Klicks eine geprüfte Zeile.

Aus der Praxis

Testen Sie jedes Skript, das etwas verändert oder löscht, zuerst mit `-WhatIf` und möglichst in einer Testumgebung. Ein falscher Filter wirkt sonst auf Hunderte Objekte gleichzeitig.

Damit Skripte auch nach Monaten noch verständlich sind, sollten Sie sie strukturieren: sprechende Namen für Variablen und Funktionen, Kommentare an den wichtigen Stellen und typisierte Parameter in einem `param()`-Block am Anfang, etwa `[string]$ComputerName`. Wiederkehrende Abläufe gehören in Funktionen, häufig genutzte Funktionen in ein eigenes Modul.

Das Wichtigste in Kürze

- PowerShell denkt in Objekten – das macht Filtern und Weiterverarbeiten zuverlässig.
- Verb-Substantiv-Cmdlets und die Pipeline bilden den Kern.
- `Get-Help` und `Get-Member` sind die wichtigsten Helfer.
- Mit `try/catch` und `-ErrorAction Stop` werden Skripte robust.
- `-WhatIf` schützt vor Unfällen.

Linux-Administration

Dateisystem, Rechte, Pakete, systemd und Automatisierung

Nach diesem Kapitel können Sie …

- sich in der Linux-Verzeichnisstruktur orientieren,
- Dateirechte lesen und mit chmod und chown setzen,
- Software mit apt oder dnf installieren und aktualisieren,
- Dienste mit systemctl steuern und Logs mit journalctl auswerten,
- Aufgaben per cron automatisieren und Server sicher per SSH verwalten.

Die Verzeichnisstruktur

Die allermeisten Server und ein Großteil der Cloud-Infrastruktur laufen unter Linux. Wer von Windows kommt, bemerkt zuerst: Es gibt keine Laufwerksbuchstaben. Alles hängt unter einem einzigen Wurzelverzeichnis, geschrieben als Schrägstrich `/`. Weitere Datenträger werden an beliebiger Stelle in diesen Baum „eingehängt“. Die Ordnung folgt dem **Filesystem Hierarchy Standard** (FHS).

Einige Verzeichnisse sollten Sie kennen: `/etc` enthält die Konfigurationsdateien des Systems und der Dienste. Unter `/var` liegen variable Daten wie Logs, Warteschlangen und Datenbanken. `/home` nimmt die persönlichen Verzeichnisse der Benutzer auf, `/usr` die installierten Programme und Bibliotheken. Eine Besonderheit sind `/proc` und `/sys`: Sie enthalten keine echten Dateien auf der Festplatte, sondern virtuelle Schnittstellen zum Kernel. So liefert etwa `/proc/cpuinfo` Informationen über den Prozessor.

Berechtigungen und Benutzer

Jede Datei und jedes Verzeichnis hat Rechte für drei Klassen: den Besitzer (*user*), die Gruppe (*group*) und alle anderen (*others*). Für jede Klasse gibt es drei Rechte: Lesen (*r*, read), Schreiben (*w*, write) und Ausführen (*x*, execute). Der Befehl `ls -l` zeigt sie etwa als `rwxr-x---` an.

Mit `chmod` setzen Sie die Rechte, besonders kompakt in der Zahlenschreibweise. Dabei steht 4 für Lesen, 2 für Schreiben und 1 für Ausführen; die Werte werden pro Klasse addiert. `chmod 750 datei` bedeutet also: Besitzer darf alles (4+2+1=7), die Gruppe darf lesen und ausführen (4+1=5), alle anderen dürfen nichts (0). Mit `chown user:group datei` ändern Sie Besitzer und Gruppe.

> **Merke**
>
> r = 4, w = 2, x = 1. Drei Ziffern stehen für Besitzer, Gruppe und andere. `chmod 750` heißt: Besitzer voll, Gruppe lesen und ausführen, andere nichts.

Benutzerkonten sind in der Datei `/etc/passwd` hinterlegt, Gruppen in `/etc/group`; die Passwort-Hashes liegen geschützt in `/etc/shadow`. Administrative Aufgaben erledigen Sie nicht als dauerhaft angemeldeter Benutzer `root`, sondern indem Sie einem einzelnen Befehl `sudo` voranstellen. Wer das darf, legt die Datei `sudoers` fest, die mit `visudo` bearbeitet wird. Der Wert `umask` bestimmt, welche Standardrechte neu angelegte Dateien erhalten. Mit Vorsicht zu behandeln ist das **SUID-Bit**: Ein so markiertes Programm läuft mit den Rechten seines Besitzers, oft also mit Root-Rechten – eine Sicherheitslücke darin kann daher das ganze System gefährden.

Paketverwaltung

Software wird unter Linux in aller Regel über einen **Paketmanager** installiert, und welcher das ist, hängt von der Distribution ab. Auf Debian und Ubuntu ist es `apt`, das auf dem Basiswerkzeug `dpkg` und dem .deb-Format aufbaut. Auf Red Hat Enterprise Linux, Rocky Linux und Fedora heißt das Werkzeug `dnf`, die Basis bilden das Werkzeug `rpm` und das gleichnamige Paketformat.

Aufgabe	Debian / Ubuntu	RHEL / Rocky / Fedora
Paketlisten aktualisieren	`apt update`	(automatisch bei `dnf`)
Paket installieren	`apt install nginx`	`dnf install nginx`
System aktualisieren	`apt upgrade`	`dnf update`
Paketformat	.deb (dpkg)	.rpm (rpm)

Die Pakete stammen aus **Repositories**, also Paketquellen der Distribution, und sind digital signiert. Der Paketmanager prüft diese Signatur vor der Installation und löst Abhängigkeiten automatisch auf: Benötigt ein Programm weitere Bibliotheken, werden sie gleich mitinstalliert.

Dienste mit systemd

Auf den meisten aktuellen Distributionen ist **systemd** das Init-System, also der erste Prozess nach dem Kernelstart. Es startet und überwacht alle Dienste. Jeder Dienst ist als *Unit* beschrieben – einer Konfigurationsdatei, die festlegt, welches Programm gestartet wird und wovon es abhängt. Das zentrale Werkzeug ist `systemctl`:

```
systemctl status nginx      # Zustand anzeigen
systemctl restart nginx     # neu starten
systemctl enable nginx      # beim Booten automatisch starten
systemctl list-units --type=service
```

Beachten Sie den Unterschied: `start` startet einen Dienst sofort, `enable` sorgt dafür, dass er beim nächsten Booten automatisch mitstartet. Für beides zusammen gibt es `systemctl enable --now`.

Logs analysieren

Läuft ein Dienst nicht, verrät meist das Log den Grund. systemd sammelt Meldungen zentral im *Journal*, das Sie mit `journalctl` auslesen. `journalctl -u nginx` zeigt gezielt das Protokoll eines Dienstes, `journalctl -f` liest neue Meldungen live mit, und `journalctl -p err -b` listet alle Fehler seit dem letzten Systemstart. Daneben gibt es unter `/var/log` weiterhin klassische Logdateien, die viele Programme zusätzlich oder ausschließlich verwenden.

Automatisierung mit cron

Wiederkehrende Aufgaben wie Backups oder Aufräumskripte lassen sich mit **cron** zeitgesteuert ausführen. Mit `crontab -e` bearbeiten Sie die eigene Aufgabenliste. Jede Zeile beginnt mit fünf Zeitfeldern – Minute, Stunde, Tag des Monats, Monat und Wochentag –, gefolgt vom Befehl. Ein Stern bedeutet „jeder Wert“:

```
0 2 * * * /opt/skripte/backup.sh    # täglich um 02:00 Uhr
```

Als modernere Alternative bietet systemd sogenannte *Timer*, die sich besser überwachen lassen und ihre Ausgaben im Journal protokollieren.

Netzwerk, SSH und Ressourcen

Für die Netzwerkdiagnose zeigt `ip a` die IP-Adressen aller Schnittstellen, `ip r` die Routing-Tabelle. `ss -tulpn` listet alle offenen Ports zusammen mit dem Prozess, der sie belegt. Auf entfernte Server greifen Sie mit `ssh user@host` über eine verschlüsselte Verbindung zu.

Aus der Praxis

Melden Sie sich per SSH mit Schlüsseln statt mit Passwörtern an. Ein Schlüsselpaar wird mit `ssh-keygen` erzeugt und mit `ssh-copy-id` auf den Server übertragen. Danach kann die Passwortanmeldung in der SSH-Konfiguration deaktiviert werden – automatisierte Passwort-Rateangriffe laufen dann ins Leere.

Laufende Prozesse zeigen `ps` und interaktiv `top` oder das komfortablere `htop`. Mit `kill` und der Prozess-ID beenden Sie einen Prozess. Freien Speicherplatz pro Dateisystem zeigt `df -h`, den Platzbedarf eines Verzeichnisses `du -sh`. Die Auslastung des Arbeitsspeichers liefert `free -h`.

Das Wichtigste in Kürze

- Alles hängt unter `/`; Konfiguration in `/etc`, Logs unter `/var/log`.
- Rechte für Besitzer, Gruppe und andere: r = 4, w = 2, x = 1.
- `apt` für Debian/Ubuntu, `dnf` für RHEL/Rocky/Fedora.
- `systemctl` steuert Dienste, `journalctl` liest Logs, cron automatisiert.
- SSH immer mit Schlüsseln absichern.

Software-Deployment & Paketsysteme

Software zuverlässig verteilen – Windows und Linux

Nach diesem Kapitel können Sie …

- begründen, warum Softwareverteilung standardisiert werden sollte,
- die Windows-Paketformate MSI und MSIX sowie die Paketmanager winget und Chocolatey einordnen,
- die Verteilung über Intune, MECM und RMM-Werkzeuge vergleichen,
- klassische Linux-Repositories von universellen Paketformaten unterscheiden,
- einen Rollout in Ringen mit Rückweg planen.

Warum Softwareverteilung standardisiert wird

Bei drei Rechnern ist es kein Problem, Software von Hand zu installieren. Bei dreihundert wird es zum Albtraum: Auf manchen Geräten laufen unterschiedliche Versionen, einige werden vergessen, und niemand kann sicher sagen, was wo installiert ist. Manuelle Installation skaliert nicht und ist fehleranfällig.

Standardisiertes Deployment löst diese Probleme. Auf allen Geräten landet dieselbe Version mit denselben Einstellungen. Der Vorgang ist nachvollziehbar, wiederholbar und lässt sich automatisieren. Und weil die Verteilung zentral gesteuert wird, lassen sich neue Versionen schnell ausrollen – und fehlerhafte ebenso schnell wieder zurücknehmen.

Paketformate unter Windows

Der Klassiker unter den Windows-Installationsformaten ist **MSI** (Microsoft Installer). Ein MSI-Paket beschreibt die Installation in einer Datenbank und unterstützt von Haus aus eine *unbeaufsichtigte Installation* (Silent-Install) ohne Rückfragen an den Benutzer sowie ein Rollback, falls die Installation fehlschlägt. Eine Silent-Installation sieht etwa so aus:

```
msiexec /i app.msi /qn /l*v C:\Logs\app-install.log
```

Der Schalter `/qn` unterdrückt jede Oberfläche, `/l*v` schreibt ein ausführliches Protokoll. Das modernere Format **MSIX** installiert Anwendungen in einer abgeschotteten Umgebung (Container). Dadurch hinterlassen sie keine Spuren im System und lassen sich besonders sauber wieder entfernen. Viele Hersteller liefern aber weiterhin klassische

EXE-Setups; auch diese bieten meist Silent-Schalter wie `/S`, `/silent` oder `/quiet`, die sich allerdings je nach Installationsprogramm unterscheiden und in der Herstellerdokumentation nachgeschlagen werden müssen.

Paketmanager: winget und Chocolatey

Ähnlich wie unter Linux gibt es auch für Windows Paketmanager, die Software per Kommandozeile aus einer zentralen Quelle installieren. **winget** ist der in aktuelle Windows-Versionen eingebaute Paketmanager von Microsoft. Mit `winget search` finden Sie ein Programm, mit `winget install` und der Paket-ID installieren Sie es, `winget upgrade --all` aktualisiert alle bekannten Programme auf einmal. **Chocolatey** ist eine von der Community gepflegte Paketquelle, die sich in vielen Unternehmen etabliert hat, auch mit kommerziellen Varianten für den Firmeneinsatz.

```
winget install --id 7zip.7zip --silent
```

Beide Werkzeuge eignen sich hervorragend für Skripte und für die Erstinstallation neuer Geräte: Statt jedes Programm einzeln herunterzuladen, läuft ein Skript mit einer Liste von Paketen durch.

Zentrale Verteilung: Intune, MECM und RMM

Für die zentrale, cloudbasierte Verwaltung setzt Microsoft auf **Intune**, eine Lösung für Mobile Device Management (MDM). Der große Vorteil: Die Geräte müssen nicht im lokalen Netz sein, die Verteilung erfolgt über das Internet – ideal für Homeoffice und Außendienst. Über Intune werden Apps, Konfigurationen und Sicherheitsrichtlinien zentral verwaltet. Eigene Win32-Programme werden dafür in das Format `.intunewin` verpackt und anschließend Benutzer- oder Gerätegruppen zugewiesen.

Intune ist allerdings kein Automatismus für jedes Unternehmen. Voraussetzung sind eine Registrierung der Geräte (Enrollment), eine passende Microsoft-Lizenz, die Intune enthält, und eine laufende Internetverbindung der Geräte. Im klassischen, lokalen Umfeld ist zudem **MECM** (Microsoft Endpoint Configuration Manager, früher SCCM) verbreitet. Viele Unternehmen kombinieren beide Ansätze; man spricht dann von *Co-Management*.

In kleinen und mittleren Unternehmen und bei IT-Dienstleistern (Managed Service Providern) ist in der Praxis häufig ein pragmatischerer Weg zu finden: **RMM-Werkzeuge** (Remote Monitoring and Management). Sie bündeln Softwareverteilung, Patching, Überwachung und Fernzugriff in einer Konsole, ohne dass ein vollständiges Intune- und Entra-ID-Setup nötig ist. Diese Werkzeuge werden im Lernbereich zu Monitoring und Management ausführlich vorgestellt. Für die serverseitige Softwareverteilung kommen außerdem Automatisierungswerkzeuge wie **Ansible** oder eigene Skripte zum Einsatz.

Werkzeug	Ansatz	Typischer Einsatz
Intune	Cloudbasiert (MDM)	Moderne Verwaltung, mobile Geräte, Homeoffice
MECM	Lokal (On-Premises)	Große, klassische Unternehmensnetze
RMM-Werkzeug	Cloud-Konsole mit Agent	KMU und IT-Dienstleister, gemischte Umgebungen
Ansible / Skripte	Automatisierung	Server, vor allem unter Linux

Paketverteilung unter Linux

Unter Linux gibt es zwei Wege. Der klassische führt über die **Repositories** der Distribution mit `apt` oder `dnf`. Die Pakete sind auf die jeweilige Distribution abgestimmt, nutzen gemeinsame Systembibliotheken und sind dadurch schlank und gut integriert. Für System- und Serverpakete ist das der Standard.

Daneben gibt es **universelle Formate** wie Snap, Flatpak und AppImage. Sie bringen ihre Abhängigkeiten selbst mit und laufen deshalb distributionsunabhängig. Der Preis dafür ist ein höherer Speicherbedarf, weil Bibliotheken mehrfach vorhanden sein können. Snap und Flatpak werden vor allem für Desktop-Anwendungen genutzt, bei denen Anwender eine aktuelle Version unabhängig vom Versionsstand der Distribution wünschen.

Paketquellen und Signaturen

Software sollte nur aus vertrauenswürdigen Quellen stammen. Repositories signieren ihre Pakete digital, unter Linux typischerweise mit **GPG**. Der Paketmanager prüft vor der Installation, ob die Signatur zum hinterlegten Schlüssel passt. So wird erkannt, ob ein Paket unterwegs manipuliert oder von einer gefälschten Quelle untergeschoben wurde. Viele Unternehmen betreiben zusätzlich **interne Spiegel** der Paketquellen: Das spart Bandbreite und erlaubt es, nur geprüfte Versionen freizugeben.

Rollout-Strategien und Best Practices

Die wichtigste Regel beim Deployment lautet: nie sofort an alle. Rollen Sie neue Software oder neue Versionen **in Ringen** aus. Zuerst erhält eine kleine Pilotgruppe – etwa die IT-Abteilung und einige erfahrene Anwender – das Paket. Laufen dort keine Probleme auf, folgt ein breiterer Ring, und erst zum Schluss werden alle Geräte versorgt. Größere Rollouts gehören in ein angekündigtes Wartungsfenster.

> **Merke**
>
> Erst im Testring ausrollen, dann breit – und immer einen Rückweg haben. Wie sich die Software sauber deinstallieren oder auf die alte Version zurücksetzen lässt, wird vor dem Rollout getestet, nicht erst im Ernstfall.

Dokumentieren Sie für jedes Paket die Version und die verwendeten Silent-Schalter, damit die Installation jederzeit reproduzierbar bleibt. Führen Sie außerdem ein **Software-Inventar**: Nur wer weiß, welche Programme in welcher Version auf welchen Geräten installiert sind, kann Lizenzen nachweisen und bei einer Sicherheitslücke schnell reagieren.

Das Wichtigste in Kürze

- Standardisierte Verteilung sorgt für gleiche Versionen, Nachvollziehbarkeit und schnelles Zurückrollen.
- MSI/MSIX sowie winget und Chocolatey sind die Werkzeuge unter Windows.
- Intune verteilt über die Cloud, MECM lokal, RMM-Werkzeuge pragmatisch für KMU.
- Unter Linux: Repositories für Systempakete, Snap/Flatpak/AppImage für universelle Pakete.
- Testring, Rollback und Inventar sind das Sicherheitsnetz.

Update- & Patch-Management

Systeme sicher und stabil aktuell halten

Nach diesem Kapitel können Sie …

- erklären, warum regelmäßiges Patchen eine der wirksamsten Schutzmaßnahmen ist,
- Windows Update, WSUS, Intune und RMM-Systeme als Update-Wege einordnen,
- den monatlichen Patch-Zyklus und Update-Ringe beschreiben,
- Updates unter Linux planen und automatisieren,
- Patches priorisieren, testen, zurückrollen und den Patch-Stand nachweisen.

Warum patchen?

„Läuft doch alles“ – genau das ist der Trugschluss. Die meisten erfolgreichen Angriffe nutzen keine geheimnisvollen neuen Tricks, sondern längst bekannte Schwachstellen, für die es bereits ein Update gibt, das aber nie eingespielt wurde. Ungepatchte Systeme sind damit das häufigste Einfallstor für Angreifer, und regelmäßiges Patchen ist eine der wirksamsten Schutzmaßnahmen überhaupt.

Neben der **Sicherheit** sprechen zwei weitere Gründe für ein geordnetes Update-Management. Updates beheben Fehler und verbessern damit die **Stabilität**. Und in vielen Branchen ist ein nachweisbar aktueller Patch-Stand Teil der **Compliance**: Wer ihn bei einem Audit oder nach einem Sicherheitsvorfall nicht belegen kann, hat ein Problem.

Windows Update und WSUS

Der einfachste Weg ist **Windows Update**: Jeder Rechner holt seine Updates direkt bei Microsoft. Das ist unkompliziert, bietet der IT aber wenig Kontrolle darüber, wann welches Update auf welchem Gerät landet.

Der klassische Unternehmensweg war über viele Jahre **WSUS** (Windows Server Update Services), ein lokaler Update-Server. WSUS lädt die Updates einmal herunter, die IT gibt sie frei, und die Clients beziehen sie aus dem lokalen Netz. Microsoft hat WSUS jedoch 2024 als veraltet eingestuft: Der Dienst läuft technisch weiter, wird aber nicht mehr weiterentwickelt. Hinzu kommen grundsätzliche Grenzen. WSUS verteilt nur Updates für Microsoft-Produkte, keine Drittanbieter-Software. Und es setzt voraus, dass die Geräte den Server im lokalen Netz erreichen. Bei verteilten Standorten und viel Homeoffice ohne dauerhafte VPN-Verbindung ist das in der Praxis kaum noch zeitgemäß.

Moderne Update-Verwaltung

Moderner arbeitet **Windows Update for Business**, das meist über **Intune** gesteuert wird. Die Geräte laden ihre Updates weiterhin direkt aus der Cloud, die IT legt aber zentral fest, wann sie installiert werden. Dazu definiert sie **Update-Ringe** mit unterschiedlichen Verzögerungen (Deferral): Die Pilotgruppe erhält Updates sofort, der breite Ring einige Tage später, die restlichen Geräte zum Schluss. So erfolgen Freigaben weitgehend automatisch, aber gestaffelt.

Für detaillierte Steuerung in lokalen Netzen gibt es **MECM**. Sein Einrichtungs- und Lizenzaufwand ist für kleine Umgebungen allerdings oft zu hoch. In kleinen und mittleren Unternehmen sowie bei IT-Dienstleistern übernehmen deshalb häufig **RMM-Systeme** das Patchen. Ihr Vorteil: Ein Werkzeug deckt Windows, macOS, Linux und auch Drittanbieter-Software wie Browser, PDF-Reader oder Office-Pakete ab. Gerade Drittanbieter-Programme sind ein häufig übersehenes Risiko.

Weg	Stärke	Grenze
Windows Update	Einfach, keine Infrastruktur	Kaum zentrale Kontrolle
WSUS	Lokale Freigabe	Auslaufmodell, nur Microsoft, nur im LAN
Intune / WUfB	Cloudbasierte Ringe, ortsunabhängig	Lizenz und Enrollment nötig
RMM-System	Alle Betriebssysteme und Drittanbieter	Zusätzliches Werkzeug mit Agent

Der Patch-Zyklus

Microsoft veröffentlicht seine Sicherheitsupdates gebündelt einmal im Monat, am sogenannten **Patch Tuesday** – dem zweiten Dienstag im Monat. Das macht die Planung berechenbar: Die IT weiß, wann Updates kommen, und kann Tests und Wartungsfenster danach ausrichten. Bei besonders kritischen, bereits ausgenutzten Lücken erscheinen zusätzlich außerplanmäßige **Out-of-Band-Updates**, die nicht bis zum nächsten Termin warten.

Der Ablauf folgt dem gleichen Ringprinzip wie bei der Softwareverteilung: Pilot, dann breit, dann alle. Für Server werden feste Wartungsfenster eingeplant, in denen Neustarts möglich sind, ohne den Betrieb zu stören. Die Anwender sollten rechtzeitig informiert werden.

Updates unter Linux

Unter Linux gilt dasselbe Prinzip. Regelmäßige Updates mit `apt upgrade` beziehungsweise `dnf update` gehören fest in den Betriebsplan. Sicherheitsupdates lassen sich automatisieren: Auf Debian und Ubuntu übernimmt das `unattended-upgrades`, auf RHEL-ba-

sierten Systemen `dnf-automatic`. Beachten Sie, dass Kernel-Updates in der Regel erst nach einem Neustart wirksam werden. Für Systeme, die nicht neu gestartet werden können, gibt es Livepatch-Verfahren, mit denen sich bestimmte Kernel-Korrekturen im laufenden Betrieb einspielen lassen.

Priorisieren, testen und zurückrollen

Nicht jeder Patch ist gleich dringend, und selten lässt sich alles gleichzeitig einspielen. Priorisieren Sie deshalb: **Kritische, aktiv ausgenutzte Lücken** kommen zuerst. Systeme, die aus dem Internet erreichbar sind – Webserver, VPN-Gateways, Firewalls –, werden bevorzugt behandelt. Und am Ende steht immer eine Abwägung zwischen dem Risiko der Lücke und dem Risiko, dass das Update selbst Probleme verursacht.

Vor der breiten Freigabe werden Updates im Testring geprüft. Lesen Sie die *Known Issues* der Hersteller, also die bekannten Probleme eines Updates, und stellen Sie sicher, dass geschäftskritische Anwendungen weiterhin funktionieren. Freigaben erfolgen gestaffelt, nicht alles auf einmal.

> **Merke**
>
> Kein Patch ohne Rückweg. Vor größeren Updates wird ein Snapshot der virtuellen Maschine oder ein Backup erstellt. Geht etwas schief, lässt sich das Update gezielt deinstallieren oder der vorherige Stand wiederherstellen.

Aus der Praxis

Ein VM-Snapshot ist ein kurzfristiges Sicherheitsnetz, kein Backup. Löschen Sie ihn, sobald das Update erfolgreich getestet ist – alte Snapshots wachsen stetig und können Speicher und Leistung des Hosts belasten.

Compliance und Reporting

Ein nachweisbarer Patch-Stand ist oft eine ausdrückliche Anforderung von Kunden, Versicherungen oder Prüfern. Messen Sie deshalb regelmäßig, wie viele Geräte auf dem aktuellen Stand sind und wo Updates fehlen. Moderne Verwaltungswerkzeuge liefern dafür fertige Berichte. Geräte, die bewusst nicht aktualisiert werden – etwa eine Maschinensteuerung, deren Hersteller ein Update nicht freigibt –, sind als **Ausnahme** zu dokumentieren und zu begründen, idealerweise mit einer ausgleichenden Maßnahme wie einer Netztrennung. So halten Sie für Audits jederzeit belastbare Nachweise bereit.

Das Wichtigste in Kürze

- Ungepatchte Systeme sind das Hauptrisiko – die meisten Angriffe nutzen bekannte Lücken.
- WSUS ist ein Auslaufmodell; Intune/Windows Update for Business oder RMM-Systeme sind der praktische Standard.
- Patch Tuesday ist der zweite Dienstag im Monat, ausgerollt wird in Ringen.
- Vor dem breiten Rollout testen und immer einen Rückweg bereithalten.
- Kritische und exponierte Systeme zuerst; Ausnahmen dokumentieren.

CVE, Zero-Day & Schwachstellen-Management

Schwachstellen erkennen, bewerten und priorisieren

Nach diesem Kapitel können Sie …

- erklären, was eine Schwachstelle und ein Exploit sind,
- CVE-Kennungen lesen und CVSS-Werte den Schweregraden zuordnen,
- die wichtigsten Informationsquellen zu Schwachstellen nennen,
- beschreiben, was einen Zero-Day so gefährlich macht und wie man reagiert,
- Schwachstellen-Management als Kreislauf aufsetzen und sinnvoll priorisieren.

Was eine Schwachstelle ist

Eine **Schwachstelle** (englisch *vulnerability*) ist ein Fehler oder eine Schwäche in einem System, die ein Angreifer ausnutzen kann. Das kann ein Programmierfehler sein, etwa eine fehlende Prüfung von Eingaben, eine *Fehlkonfiguration* wie ein Standardpasswort oder ein offener Port, oder ein grundlegender *Designfehler*, etwa ein Protokoll ohne Verschlüsselung. Ein Programm oder Verfahren, das eine bestimmte Lücke gezielt ausnutzt, heißt **Exploit**.

Das Ziel des Schwachstellen-Managements lässt sich in einem Satz zusammenfassen: Lücken schneller schließen, als Angreifer sie ausnutzen können. Weil täglich neue Schwachstellen veröffentlicht werden, braucht es dafür eine gemeinsame Sprache, eine Bewertung und einen geregelten Prozess.

CVE: eindeutige Namen für Schwachstellen

Damit Hersteller, Sicherheitsforscher und Administratoren weltweit über dieselbe Lücke sprechen, gibt es **CVE** – Common Vulnerabilities and Exposures. Das ist ein öffentlicher Katalog, in dem jede bekannte Schwachstelle eine eindeutige Kennung erhält. Das Format besteht aus dem Präfix „CVE“, dem Jahr und einer laufenden Nummer, zum Beispiel `CVE-2024-12345`.

Die Nummern werden vom CVE-Programm vergeben, das die Organisation MITRE betreibt, sowie von zahlreichen **CNAs** (CVE Numbering Authorities). Dazu gehören große Softwarehersteller, Sicherheitsfirmen und nationale Stellen, die für ihre eigenen Produkte oder Bereiche Nummern zuteilen dürfen. Jede Schwachstelle erhält genau eine ID, die herstellerübergreifend eindeutig ist. Sicherheitsmeldungen (Advisories), Patches und Schwachstellen-Scanner beziehen sich auf diese Kennungen.

CVSS: den Schweregrad bewerten

Wie gefährlich eine Lücke ist, beschreibt das **CVSS** – Common Vulnerability Scoring System. Es bewertet den Schweregrad auf einer Skala von 0 bis 10 und ordnet den Wert einer von vier Stufen zu:

CVSS-Wert	Schweregrad	Bedeutung
0,1 – 3,9	Niedrig	Geringeres Risiko
4,0 – 6,9	Mittel	Einplanen
7,0 – 8,9	Hoch	Dringend handeln
9,0 – 10,0	Kritisch	Sofort handeln

In den Wert fließen mehrere Metrik-Gruppen ein. Die *Base-Metriken* beschreiben die Lücke selbst und ändern sich nicht. Zeitabhängige Metriken – in CVSS 3.x „Temporal“, in Version 4.0 „Threat“ genannt – berücksichtigen etwa, ob bereits ein funktionierender Exploit existiert. Die *Environmental-Metriken* passen die Bewertung an die eigene Umgebung an: Eine Lücke auf einem isolierten Testsystem wiegt weniger schwer als auf einem zentralen Server.

Zu jedem CVSS-Wert gehört ein **Vektor**, der die Bewertung nachvollziehbar macht. Er beschreibt unter anderem den *Angriffsvektor* – über das Netzwerk, aus einem angrenzenden Netz, lokal oder nur mit physischem Zugriff –, die *Komplexität* des Angriffs, die *benötigten Rechte* und ob eine *Nutzer-Interaktion* nötig ist, etwa das Öffnen einer Datei. Eine Lücke, die sich ohne Anmeldung und ohne Mitwirkung eines Nutzers über das Netzwerk ausnutzen lässt, ist naturgemäß besonders gefährlich.

> **Merke**
>
> CVE benennt eine Schwachstelle, CVSS bewertet ihren Schweregrad. Der Score hilft beim Priorisieren, ersetzt aber nie den Blick auf die eigene Umgebung.

Wichtige Informationsquellen

Die zentrale öffentliche Datenbank ist die **NVD** (National Vulnerability Database) des US-amerikanischen NIST. Sie ergänzt die CVE-Einträge um CVSS-Bewertungen, betroffene Produkte und Verweise. Die **Exploit-DB** sammelt veröffentlichte Exploit-Codes; findet sich dort ein Eintrag zu einer Lücke, ist ihre Ausnutzung deutlich wahrscheinlicher geworden. Unverzichtbar sind außerdem die **Advisories der Hersteller**, etwa von Microsoft oder Netzwerkausrüstern, denn sie nennen betroffene Versionen, Patches und Workarounds.

Besonders wertvoll für die Priorisierung ist der **KEV-Katalog** (Known Exploited Vulnerabilities) der US-Behörde CISA. Er listet Lücken, die nachweislich bereits aktiv ausgenutzt werden. Steht eine Schwachstelle dort, ist sie keine theoretische Gefahr mehr, sondern ein reales Angriffswerkzeug.

Zero-Day: die Lücke ohne Patch

Ein **Zero-Day** ist eine Schwachstelle, für die es noch keinen Patch gibt – der Hersteller hatte gewissermaßen „null Tage“ Zeit, sie zu schließen. Oft wird sie erst entdeckt, weil Angreifer sie bereits ausnutzen. Zero-Days sind besonders gefährlich, weil die Standardabwehr per Update fehlt. Auf dem Schwarzmarkt werden sie teuer gehandelt und meist gezielt gegen ausgewählte Opfer eingesetzt.

Bis ein Patch verfügbar ist, bleiben nur Zwischenlösungen: **Workarounds** aus dem Hersteller-Advisory, etwa das Abschalten einer betroffenen Funktion, eine zusätzliche **Härtung** wie das Einschränken des Zugriffs über die Firewall und ein verschärftes **Monitoring**, um Angriffsversuche früh zu erkennen. Sobald der Patch erscheint, hat er höchste Priorität.

Schwachstellen-Management als Kreislauf

Schwachstellen-Management ist kein einmaliges Projekt, sondern ein dauerhafter Kreislauf aus vier Schritten: **Scannen → Priorisieren → Beheben → Verifizieren** – und dann von vorn. Für das regelmäßige Scannen gibt es spezialisierte Werkzeuge wie Nessus, OpenVAS oder Qualys, die Systeme im Netz auf bekannte Schwachstellen prüfen. Priorisiert wird nach CVSS-Wert, Erreichbarkeit und tatsächlicher Ausnutzung. Nach dem Patchen wird erneut gescannt, um zu bestätigen, dass die Lücke wirklich geschlossen ist.

Für die Priorisierung hat sich eine einfache Staffelung bewährt. Lücken, die *aktiv ausgenutzt* werden, also etwa im KEV-Katalog stehen, werden sofort behoben. *Kritische* Lücken auf Systemen, die aus dem Internet erreichbar sind, haben sehr hohe Priorität. *Hohe* Lücken auf internen oder isolierten Systemen werden geplant behoben. *Niedrige* Lücken laufen im normalen Patch-Zyklus mit.

Vom Advisory zur Aktion

Erscheint eine neue Sicherheitsmeldung, entscheiden zwei Fragen über die Dringlichkeit: *Sind eigene Systeme betroffen?* und *Sind sie erreichbar?* Gleichen Sie dazu die im Advisory genannten Versionen mit Ihrem Software-Inventar ab. Sind Systeme betroffen, spielen Sie den Patch ein oder setzen zunächst den empfohlenen Workaround um. Verfolgen Sie den Vorgang anschließend bis zur Verifizierung nach – erst wenn ein erneuter Scan die Lücke nicht mehr findet, ist der Fall abgeschlossen.

Aus der Praxis

Ein vollständiges und aktuelles Software-Inventar ist die Voraussetzung für schnelles Handeln. Ohne Inventar lässt sich die Frage „Sind wir betroffen?“ nicht in Minuten beantworten, sondern erst nach tagelanger Suche.

Das Wichtigste in Kürze

- CVE ist die eindeutige ID einer Schwachstelle, CVSS bewertet den Schweregrad von 0 bis 10.
- NVD, Hersteller-Advisories und der CISA-KEV-Katalog sind die wichtigsten Quellen.
- Ein Zero-Day ist eine Lücke ohne Patch – Workarounds, Härtung und Monitoring überbrücken.
- Management ist ein Kreislauf: Scannen, Priorisieren, Beheben, Verifizieren.
- Entscheidend ist, ob eine Lücke aktiv ausgenutzt wird und das System erreichbar ist.

macOS-Administration

Das dritte große Betriebssystem im Unternehmen – Unix-Basis, MDM und Praxis

Nach diesem Kapitel können Sie …

- einschätzen, wo macOS im Unternehmen typischerweise eingesetzt wird,
- die Bedeutung von Apple Silicon und der Unix-Basis für die Administration erklären,
- beschreiben, wie Macs per MDM verwaltet werden und welche Rolle der Apple Business Manager spielt,
- Macs ins Firmennetz einbinden und die Grenzen von Time Machine benennen.

Wo macOS im Unternehmen begegnet

Neben Windows und Linux ist **macOS** das dritte Betriebssystem, das im IT-Alltag eine Rolle spielt. Vor allem in drei Bereichen ist es verbreitet: in **Kreativabteilungen**, die mit Design, Video oder Audio arbeiten, in der **Software-Entwicklung** und in Unternehmen, die sich bewusst für Apple-Geräte entschieden haben. Hinzu kommt, dass Geschäftsführung oder einzelne Fachabteilungen häufig Macs nutzen, auch wenn der Rest des Hauses auf Windows setzt.

Für den Support bedeutet das: Sobald Entwicklerinnen, Designer oder die Geschäftsleitung im Haus sind, taucht früher oder später ein Mac im Ticketsystem auf. Grundkenntnisse in macOS gehören deshalb zum Handwerkszeug, auch in überwiegend Windows-geprägten Umgebungen.

Apple Silicon

Einen großen Einschnitt brachte der Wechsel von Intel-Prozessoren zu Apples eigenen **M-Chips**, die ab 2020 schrittweise eingeführt wurden und in mehreren Generationen weiterentwickelt werden. Sie basieren auf der **ARM-Architektur**, die ursprünglich aus der Welt der Smartphones stammt, statt auf der klassischen x86-Architektur der meisten PCs.

Für Anwender zeigt sich das vor allem in einer sehr hohen Effizienz pro Watt: lange Akkulaufzeiten bei hoher Leistung und kaum Lüftergeräusch, auch unter Last. Für die Administration ist die Architektur vor allem dann relevant, wenn ältere Software nur für

x86 vorliegt. Solche Programme laufen über die Übersetzungsschicht Rosetta 2, was in der Regel gut funktioniert. Treiber und systemnahe Software müssen dagegen ausdrücklich für Apple Silicon verfügbar sein.

Unix-Basis und Terminal

Unter der grafischen Oberfläche läuft macOS auf einer **Unix-Basis**. Für alle, die bereits mit Linux gearbeitet haben, fühlt sich das Terminal deshalb sofort vertraut an. Die Standard-Shell ist `zsh`, und viele Befehle wie `ls`, `grep`, `ssh` oder `ping` funktionieren genauso wie unter Linux. Einige Werkzeuge verhalten sich im Detail etwas anders, weil sie aus der BSD-Tradition stammen.

Ein deutlicher Unterschied liegt in der Softwareinstallation: macOS bringt ab Werk keinen Paketmanager für die Kommandozeile mit. Diese Lücke füllt in der Praxis **Homebrew**, ein quelloffener Paketmanager, den viele Administratoren und Entwickler als Erstes installieren. Damit lassen sich Programme ähnlich wie mit `apt` per Befehl installieren und aktualisieren:

```
brew install wget
brew upgrade
```

Aus der Praxis

Wer die Unix-Grundlagen aus der Linux-Administration beherrscht – Dateirechte, Shell, SSH, Pipes –, findet sich auf dem Mac überraschend schnell zurecht. Das Wissen lässt sich weitgehend direkt übertragen.

Geräteverwaltung per MDM

Ein Mac lässt sich nicht wie ein Windows-Client in die Domäne aufnehmen und per Gruppenrichtlinie steuern: **Active Directory und klassische GPOs greifen bei macOS nicht direkt.** Die Verwaltung im Unternehmen läuft stattdessen über **MDM** (Mobile Device Management). Verbreitete Lösungen sind etwa Jamf, eine auf Apple-Geräte spezialisierte Plattform, und Microsoft Intune, das neben Windows auch Macs verwalten kann.

Über das MDM werden sogenannte **Konfigurationsprofile** verteilt. Sie enthalten zum Beispiel WLAN-Zugangsdaten, VPN-Einstellungen und Sicherheitsrichtlinien wie Bildschirmsperre, Passwortregeln oder die Festplattenverschlüsselung FileVault. Auch die **Softwareverteilung** erfolgt zentral, ohne dass der Benutzer selbst etwas einrichten muss.

Für die volle Kontrolle über ein Gerät gibt es allerdings einen wichtigen Haken. Viele weitreichende Verwaltungsfunktionen stehen nur im sogenannten **Supervised Mode** zur Verfügung. Den erhält ein Mac in der Regel nur, wenn er über den **Apple Business Manager** (ABM) der Organisation zugeordnet ist und sich bei der Einrichtung automatisch beim MDM registriert. Das klappt problemlos bei Geräten, die über einen autorisierten Händler oder direkt bei Apple für das Unternehmen gekauft wurden. Bei privat oder gebraucht gekauften Geräten ohne diese Zuordnung ist die volle Kontrolle nicht ohne Weiteres möglich.

> **Merke**
>
> Macs werden über MDM verwaltet, nicht über AD und GPO. Volle Kontrolle gibt es nur für Geräte, die über den Apple Business Manager dem Unternehmen zugeordnet sind – deshalb sollten Firmen-Macs über passende Bezugswege gekauft werden.

Diese Registrierungsanforderung ist ein Grund, warum neben Intune häufig spezialisierte Lösungen wie Jamf oder RMM-Werkzeuge mit Mac-Unterstützung eingesetzt werden. Kenntnisse in MDM-Werkzeugen sind bei IT-Administratoren entsprechend zunehmend gefragt.

Netzwerkeinbindung

Im Netzwerk verhält sich ein Mac weitgehend unauffällig. Standard-WLAN mit Unternehmensauthentifizierung, kabelgebundene Anschlüsse und auch **VLAN-getaggte Netze** lassen sich ganz normal über die Systemeinstellungen konfigurieren oder per Konfigurationsprofil verteilen. Für den Zugriff auf Dateifreigaben von Windows-Servern spricht macOS das Protokoll **SMB**; Freigaben werden im Finder über „Mit Server verbinden“ und eine Adresse wie `smb://fileserver/daten` eingebunden.

Daneben bringt macOS eigene Mechanismen mit, die es unter Windows so nicht gibt: **AirDrop** für den drahtlosen Dateiaustausch zwischen Apple-Geräten in der Nähe und **Time Machine** als eingebaute Backup-Lösung. Im Unternehmen sollte per MDM festgelegt werden, ob und wie AirDrop genutzt werden darf, damit vertrauliche Dateien nicht unkontrolliert weitergegeben werden.

Backup mit Time Machine

Time Machine ist komfortabel: Einmal eine externe Festplatte oder einen geeigneten Netzwerkspeicher als Ziel eingerichtet, sichert macOS automatisch stündlich und im Hintergrund. Ältere Stände bleiben erhalten, solange Platz vorhanden ist, und einzelne Dateien lassen sich bequem wiederherstellen. Für den Einzelplatz ist das eine gute Lösung.

Im Unternehmen ersetzt Time Machine jedoch **keine zentrale Backup-Strategie**. Es fehlen eine zentrale Überwachung, ob alle Sicherungen erfolgreich laufen, eine geregelte Versionierung und Aufbewahrung sowie eine Kopie an einem getrennten Ort. Firmendaten gehören deshalb zusätzlich in die zentrale Datensicherung oder sollten von vornherein auf zentral gesicherten Speicherorten liegen.

Das Wichtigste in Kürze

- macOS ist das dritte relevante Betriebssystem, vor allem in Kreativbereichen, Entwicklung und Geschäftsleitung.
- Unix-Basis mit zsh; Software per Homebrew, Linux-Wissen ist direkt übertragbar.
- Verwaltung über MDM (z. B. Jamf, Intune) statt AD/GPO – volle Kontrolle nur mit Apple Business Manager.
- Netzwerkeinbindung mit Standardtechnik: WLAN, VLAN, SMB.
- Time Machine und AirDrop sind Bordmittel, ersetzen aber keine zentrale Backup-Strategie.

Dateisysteme

NTFS, ext4, FAT32 und exFAT – was Sie im Alltag wirklich wissen müssen

Nach diesem Kapitel können Sie …

- erklären, warum NTFS-Datenträger unter Linux beim Schreiben Probleme machen können,
- NTFS und ext4 hinsichtlich Rechten, Verschlüsselung und Journaling vergleichen,
- die 4-GiB-Grenze von FAT32 erkennen und begründen,
- für Server, Clients und Wechselmedien das passende Dateisystem wählen.

Die Ausgangssituation

Ein Dateisystem legt fest, wie Dateien und Ordner auf einem Datenträger organisiert, benannt und mit Rechten versehen werden. Solange ein Datenträger nur in einem einzigen Betriebssystem verwendet wird, merkt man davon wenig. Spannend wird es, sobald Datenträger zwischen verschiedenen Systemen wandern.

Ein klassischer Fall aus dem Support: Ein Kunde bringt eine externe Festplatte mit Windows-Daten, und sie wird an einen Linux-Server angeschlossen, um Daten zu übernehmen. Das Lesen funktioniert fast immer. Das Schreiben ist komplizierter – und genau dort entstehen die häufigsten Support-Fälle.

NTFS unter Linux

NTFS ist das Standard-Dateisystem von Windows. Linux kann NTFS-Datenträger in aller Regel ohne zusätzliche Einrichtung lesen. Für zuverlässiges **Schreiben** wird jedoch ein passender Treiber benötigt. Klassisch ist das das Paket `ntfs-3g`; neuere Linux-Kernel bringen zusätzlich den eingebauten Treiber `ntfs3` mit. Welcher Weg verfügbar ist, hängt von Distribution und Kernelversion ab. Fehlt ein schreibfähiger Treiber, wird der Datenträger nur lesend eingebunden – und man wundert sich, warum sich keine Dateien speichern lassen.

```
sudo apt install ntfs-3g
sudo mount -t ntfs-3g /dev/sdb1 /mnt/usb
```

Aus der Praxis

Lässt sich ein NTFS-Datenträger trotz installiertem Treiber nur lesend einbinden, wurde er unter Windows oft nicht sauber getrennt – etwa wegen des Windows-Schnellstarts oder des Ruhezustands. Den Datenträger dann unter Windows ordnungsgemäß auswerfen bzw. Windows vollständig herunterfahren und erneut anschließen.

NTFS und ext4 im Vergleich

NTFS ist der Standard unter Windows. Es verwaltet Zugriffsrechte über **ACLs** (Access Control Lists), mit denen sich für einzelne Benutzer und Gruppen sehr fein abgestufte Berechtigungen vergeben lassen – etwa „Lesen und Ausführen“, aber nicht „Löschen“. Außerdem unterstützt NTFS mit **EFS** (Encrypting File System) die Verschlüsselung einzelner Dateien direkt im Dateisystem.

ext4 ist der verbreitete Standard unter Linux. Es nutzt das klassische Unix-Rechtesystem mit den drei Klassen Benutzer, Gruppe und andere, jeweils mit Lese-, Schreib- und Ausführungsrecht. Das ist einfacher als ACLs, reicht für die meisten Serveraufgaben aber völlig aus.

Gemeinsam ist beiden das **Journaling**: Änderungen werden zuerst in einem Protokoll vermerkt und erst dann ausgeführt. Stürzt das System mitten in einem Schreibvorgang ab, lässt sich das Dateisystem anhand des Journals schnell wieder in einen konsistenten Zustand bringen, statt beschädigt zurückzubleiben. Beide Dateisysteme können zudem sehr große Dateien im Terabyte-Bereich speichern – ext4 bei der üblichen Blockgröße bis 16 TiB pro Datei, NTFS je nach Clustergröße noch deutlich mehr. Im Alltag spielt diese Grenze praktisch keine Rolle.

Merkmal	NTFS	ext4
Standard für	Windows	Linux
Rechte	Feingranulare ACLs	Benutzer, Gruppe, andere (rwx)
Verschlüsselung	EFS eingebaut	Über zusätzliche Mechanismen
Journaling	Ja	Ja

Merke

Für Windows-Server ist NTFS gesetzt. Für Linux-Server ist ext4 der Standard.

Wechselmedien: FAT32

Bei USB-Sticks, Speicherkarten und externen Festplatten zählt vor allem eines: Kompatibilität. Hier ist **FAT32** der Klassiker. Es wird von nahezu allem gelesen und beschrieben – von Windows, macOS und Linux ebenso wie von Druckern, Kameras, Fernsehern oder Autoradios.

Der große Haken ist die **Dateigrößenbegrenzung**: Eine einzelne Datei darf höchstens 4 GiB minus ein Byte groß sein, also exakt 4.294.967.295 Byte. Der Grund liegt darin, dass FAT32 die Dateigröße in einem 32-Bit-Feld speichert. Wer eine größere Datei kopieren will – etwa ein Video, ein ISO-Abbild oder ein Backup-Archiv –, erhält eine wenig aussagekräftige Fehlermeldung, oft sinngemäß „Datei zu groß für das Zieldateisystem“, obwohl auf dem Stick noch reichlich Platz frei ist.

Der Nachfolger: exFAT

exFAT wurde als Nachfolger für Wechselmedien entwickelt und löst die Größenbegrenzung von FAT32 auf. Es kennt keine praktisch relevante Grenze für die Dateigröße und wird von allen modernen Betriebssystemen unterstützt. Damit ist exFAT die erste Wahl für USB-Sticks und externe Platten, auf denen große Dateien zwischen verschiedenen Systemen ausgetauscht werden. Nur sehr alte oder einfache Geräte, etwa manche Drucker oder Autoradios, kommen mitunter nur mit FAT32 zurecht. Anders als NTFS und ext4 bietet exFAT allerdings weder Journaling noch Rechteverwaltung – für fest eingebaute Systemlaufwerke ist es daher nicht gedacht.

Für die Praxis ergibt sich daraus eine einfache Regel: Wer auf einem FAT32-Stick eine große Filmdatei speichern will, sichert vorhandene Daten und formatiert den Stick mit exFAT neu.

Das Wichtigste in Kürze

- Windows-Systeme → NTFS, Linux-Systeme → ext4.
- NTFS unter Linux: Lesen geht fast immer, für zuverlässiges Schreiben ist ein passender Treiber wie ntfs-3g nötig.
- FAT32 ist maximal kompatibel, aber auf knapp 4 GiB pro Datei begrenzt.
- Für Wechselmedien mit großen Dateien ist exFAT die erste Wahl.

LB-03

Hardware

1. CPU – Geschichte, Architektur & Markt
2. RAM – Typen, Specs & ECC
3. Storage – HDD, SSD, NVMe
4. Mainboard & Formfaktoren
5. Server-Hardware
6. Netzwerkkarten & Verkabelung
7. Hardware-Diagnose & Lifecycle

CPU – Geschichte, Architektur & Markt

x86 und ARM, Kerne, Takt, Cache, TDP & Sockel

Nach diesem Kapitel können Sie …

- die Architekturen x86 und ARM unterscheiden und die wichtigsten Produktreihen zuordnen,
- die Kennzahlen Kerne, Threads, Takt, Cache und TDP erklären,
- nachvollziehen, warum Modellbezeichnungen allein wenig über die echte Leistung aussagen,
- prüfen, ob eine CPU zu Sockel, Chipsatz und Einsatzzweck passt.

Das Herzstück des Rechners

Die **CPU** (Central Processing Unit), also der Prozessor, führt die Befehle aus, aus denen jedes Programm besteht. Sie rechnet, vergleicht, verschiebt Daten und steuert, was als Nächstes passiert. Alle anderen Komponenten – Arbeitsspeicher, Datenträger, Netzwerkkarte – liefern ihr Daten zu oder nehmen Ergebnisse entgegen. Wie schnell ein System insgesamt reagiert, hängt deshalb stark davon ab, wie leistungsfähig der Prozessor ist und wie gut er zur jeweiligen Aufgabe passt.

Für die Praxis müssen Sie nicht wissen, wie ein Prozessor auf Transistorebene aufgebaut ist. Wichtig sind vielmehr zwei Dinge: die grundlegende Architektur, auf der eine CPU beruht, und eine Handvoll Kennzahlen, mit denen sich Modelle vergleichen lassen.

Zwei große Architekturen: x86 und ARM

Eine Prozessorarchitektur legt fest, welche Befehle eine CPU versteht. Software muss für eine bestimmte Architektur übersetzt sein, damit sie darauf läuft. Historisch dominiert im PC- und Servermarkt die **x86-Architektur**, die vor allem von Intel und AMD gebaut wird. Nahezu alle klassischen Windows-Rechner und ein Großteil der Server in Rechenzentren arbeiten mit x86-Prozessoren.

Daneben gewinnt die **ARM-Architektur** stark an Bedeutung. Sie stammt ursprünglich aus der Welt der Mobilgeräte und ist auf Energieeffizienz ausgelegt: ARM-Prozessoren liefern viel Rechenleistung pro Watt. Bekannte Beispiele sind die Chips der M-Reihe in aktuellen Apple-Rechnern. Zunehmend finden sich ARM-Prozessoren auch im Server, weil dort Stromverbrauch und Kühlung erhebliche Kostenfaktoren sind.

Innerhalb der Hersteller gibt es getrennte Produktreihen für unterschiedliche Einsatzzwecke. Bei Intel steht *Core* für Desktop- und Notebook-Prozessoren, *Xeon* für Server und Workstations. Bei AMD heißen die entsprechenden Reihen *Ryzen* und *EPYC*. Server-Prozessoren unterscheiden sich nicht nur durch mehr Kerne, sondern auch durch Funktionen wie die Unterstützung von ECC-Arbeitsspeicher, mehr Speicherkanäle und mehr PCIe-Anschlüsse.

Die wichtigsten Kennzahlen

Vier Werte sollten Sie bei jeder CPU kennen. Die Anzahl der **Kerne** gibt an, wie viele Aufgaben der Prozessor wirklich gleichzeitig bearbeiten kann – jeder Kern ist im Grunde ein eigenständiges Rechenwerk. Der **Takt**, angegeben in Gigahertz, beschreibt, wie schnell ein einzelner Kern arbeitet. Moderne Prozessoren passen ihren Takt laufend an: Bei geringer Last sinkt er, um Strom zu sparen, unter Last steigt er kurzzeitig auf einen höheren Boost-Takt.

Der **Cache** ist ein kleiner, sehr schneller Zwischenspeicher direkt im Prozessor. Und die **TDP** (Thermal Design Power) gibt in Watt an, wie viel Wärme die Kühlung ungefähr abführen muss. Sie ist ein Richtwert für die Auslegung des Kühlers, nicht unbedingt der tatsächliche Höchstverbrauch.

Ein weiterer Begriff, der Ihnen auf Datenblättern begegnet, sind **Threads**. Mit einer Technik namens *Hyper-Threading* (Intel) oder allgemein *SMT* (Simultaneous Multithreading) kann ein Kern zwei Befehlsströme abwechselnd bearbeiten und Leerlaufzeiten besser nutzen. Ein Prozessor mit acht Kernen erscheint im Betriebssystem dann als sechzehn logische Prozessoren. Das hilft bei paralleler Last spürbar, ist aber nicht gleichwertig mit echten Kernen: Zwei Threads teilen sich die Rechenwerke desselben Kerns.

Kennzahl	Bedeutung	Wichtig vor allem für
Kerne / Threads	Parallele Bearbeitung von Aufgaben	Virtualisierung, Datenbanken, Rendering
Takt (GHz)	Geschwindigkeit je Kern	Anwendungen, die nur wenige Kerne nutzen
Cache	Schneller Zwischenspeicher in der CPU	Datenbanken, Server-Last
TDP (Watt)	Abzuführende Verlustleistung	Kühlung, Stromkosten

Cache-Stufen: Warum der Zwischenspeicher so wichtig ist

Der Arbeitsspeicher ist aus Sicht der CPU vergleichsweise langsam. Müsste der Prozessor für jeden Befehl auf den RAM warten, stünde er die meiste Zeit still. Der Cache überbrückt diese Lücke, indem er häufig benötigte Daten direkt im Chip bereithält. Er

ist in mehrere Stufen gegliedert: Der **L1-Cache** ist winzig, aber extrem schnell und gehört jeweils zu einem Kern. Der **L2-Cache** ist größer und meist ebenfalls pro Kern vorhanden. Der **L3-Cache** ist der größte und wird von allen Kernen gemeinsam genutzt.

Mehr Cache zahlt sich besonders bei Datenbanken und Serveranwendungen aus, die mit großen Datenmengen arbeiten und immer wieder auf dieselben Bereiche zugreifen.

Herstellertricks: Binning und Kernsegmentierung

Nicht jede Modellbezeichnung steht für einen eigenen Chip-Entwurf. Prozessoren werden auf großen Siliziumscheiben, den *Wafern*, gefertigt, und dabei gelingen nicht alle Chips gleich gut. Beim sogenannten **Binning** werden die einzelnen Chips (Dies) nach Qualität sortiert. Ein Die mit defekten oder schwächeren Kernen wird über eingebrannte Sicherungen, sogenannte *eFuses*, dauerhaft beschnitten und als günstigeres Modell verkauft. So kann aus einem Acht-Kern-Die mit zwei fehlerhaften Kernen beispielsweise ein Sechs-Kern-Modell der Mittelklasse werden statt eines Oberklasse-Modells. Besonders gelungene Exemplare landen dagegen in den übertaktbaren Modellen – bei Intel sind das die mit „K“ gekennzeichneten Prozessoren.

Seit einigen Generationen kombiniert Intel zudem leistungsstarke *P-Cores* (Performance) mit sparsameren *E-Cores* (Efficiency) in einem Hybriddesign. Das hat technische Vorteile, weil Hintergrundaufgaben auf den sparsamen Kernen laufen können. Es dient aber auch der Produktabgrenzung, denn die Gesamtzahl der Kerne wirkt auf dem Datenblatt eindrucksvoll. AMD setzt bei seinen Desktop-Ryzen-Prozessoren bisher überwiegend auf gleichartige Kerne. Die Lehre daraus: Die Kernzahl allein sagt wenig über die echte Mehrleistung aus. Vergleichen Sie Prozessoren lieber anhand unabhängiger Tests für Ihren konkreten Einsatzzweck.

> **Merke**
>
> Zahlen auf dem Datenblatt sind nur ein Anhaltspunkt. Kerne sind nicht gleich Kerne, und Threads sind keine echten Kerne. Entscheidend ist, wie sich ein Prozessor bei Ihrer tatsächlichen Arbeitslast verhält.

Sockel, Chipsatz und Kompatibilität

Eine CPU sitzt in einem **Sockel** auf dem Mainboard, und dieser muss exakt zum Prozessor passen. Ein AMD-Ryzen passt nicht in einen Intel-Sockel und umgekehrt; auch zwischen den Generationen eines Herstellers wechseln die Sockel regelmäßig. Der **Chipsatz** des Mainboards bestimmt außerdem den Funktionsumfang, etwa die Zahl der USB- und SATA-Anschlüsse oder ob Übertakten möglich ist.

Selbst wenn der Sockel passt, erkennt ein älteres Mainboard einen neueren Prozessor häufig erst nach einem **UEFI-Update**, also einer Aktualisierung der Firmware. Prüfen Sie vor einem Kauf oder Wechsel deshalb immer die Kompatibilitätsliste des Mainboard-Herstellers.

Kühlung und die Auswahl für Server

Jeder Prozessor erzeugt Wärme, die abgeführt werden muss. Gängig sind Luftkühler mit Kühlkörper und Lüfter sowie Wasserkühlungen. Die Kühlung muss zur TDP des Prozessors und zur Befestigung des Sockels passen. Ist sie zu schwach, drosselt die CPU ihren Takt, um sich zu schützen – das System wird spürbar langsamer.

Im Server gelten eigene Maßstäbe. Für Virtualisierung und Datenbanken zählen viele Kerne oft mehr als ein besonders hoher Takt, denn dort laufen zahlreiche Aufgaben gleichzeitig. Die Kernzahl sollte zur erwarteten Last passen – zu viele Kerne treiben unnötig die Kosten, und bei manchen Softwareprodukten richtet sich die Lizenzierung nach der Kernzahl. Außerdem muss die CPU ECC-Arbeitsspeicher unterstützen, damit Speicherfehler erkannt und korrigiert werden. Und weil Server rund um die Uhr laufen, zählt die *Leistung pro Watt*: Effiziente Prozessoren senken die Strom- und Klimakosten über die Jahre spürbar.

Aus der Praxis

Bevor Sie einen Server-Prozessor auswählen, klären Sie, welche Software darauf laufen soll und wie diese lizenziert wird. Eine CPU mit wenigen, schnellen Kernen kann bei kernbasierter Lizenzierung günstiger sein als ein Modell mit sehr vielen Kernen, die gar nicht ausgelastet werden.

Das Wichtigste in Kürze

- x86 (Intel, AMD) und ARM sind die beiden großen Prozessorarchitekturen; ARM punktet mit Effizienz.
- Kerne, Takt, Cache und TDP sind die zentralen Kennzahlen; Threads ersetzen keine echten Kerne.
- Modellnamen und Kernzahlen allein sagen wenig über die tatsächliche Leistung aus.
- Die CPU muss zu Sockel und Chipsatz passen; oft ist vorher ein UEFI-Update nötig.
- Im Server zählen viele Kerne, ECC-Unterstützung und Leistung pro Watt.

RAM – Typen, Specs & ECC

Arbeitsspeicher richtig verstehen und auswählen

Nach diesem Kapitel können Sie …

- erklären, welche Aufgabe der Arbeitsspeicher im System hat,
- DDR4, DDR5, LPDDR und Registered-Module unterscheiden,
- die Kennzahlen Kapazität, Takt, Timings und Kanäle einordnen,
- begründen, warum ECC-Speicher im Server unverzichtbar ist,
- Arbeitsspeicher richtig bestücken und Speicherfehler eingrenzen.

Was der Arbeitsspeicher leistet

Der Arbeitsspeicher, kurz **RAM** (Random Access Memory), hält die Daten und Programme bereit, mit denen der Prozessor gerade arbeitet. Wenn Sie ein Programm starten oder eine Datei öffnen, wird sie vom Datenträger in den RAM geladen, weil die CPU dort um ein Vielfaches schneller darauf zugreifen kann als auf jede SSD oder Festplatte.

RAM ist allerdings *flüchtig*: Sobald der Rechner ausgeschaltet wird, ist sein Inhalt verloren. Deshalb müssen Daten dauerhaft auf einem Datenträger gespeichert werden. Ein weiterer wichtiger Punkt: Reicht der Arbeitsspeicher nicht aus, lagert das Betriebssystem Teile davon auf den Datenträger aus. Weil dieser viel langsamer ist, wird das System dann spürbar träge. Zu wenig RAM ist einer der häufigsten Gründe für langsame Arbeitsplätze und Server.

Die Generationen: DDR4 und DDR5

Heute verbreiteter Arbeitsspeicher arbeitet nach dem DDR-Verfahren (Double Data Rate), das Daten pro Taktzyklus zweimal überträgt. Im Einsatz sind vor allem zwei Generationen. **DDR4** ist etabliert, günstig und in sehr vielen bestehenden Systemen verbaut. **DDR5** bietet eine höhere Bandbreite und ermöglicht größere Module, setzt aber eine neuere Plattform voraus.

Die Generationen sind nicht untereinander austauschbar. DDR4- und DDR5-Module haben eine unterschiedliche Kerbe und passen physisch nicht in denselben Steckplatz. Welchen Typ Sie verwenden können, bestimmen Mainboard und Prozessor gemeinsam – ein Blick ins Handbuch klärt das vor jedem Kauf.

Die wichtigsten Kennzahlen

Für die meisten Aufgaben ist die **Kapazität**, angegeben in Gigabyte, der wichtigste Faktor. Ein System mit ausreichend Speicher, der etwas langsamer ist, arbeitet in der Regel flüssiger als eines mit sehr schnellem, aber knappem Speicher. Erst wenn genug Kapazität vorhanden ist, lohnt der Blick auf die Geschwindigkeit.

Der **Takt** wird in Megahertz oder genauer in MT/s (Megatransfers pro Sekunde) angegeben und beschreibt, wie viele Daten pro Sekunde übertragen werden können. Die **Timings** – allen voran die *CAS-Latenz* (CL) – beschreiben die Verzögerung, bis angeforderte Daten bereitstehen. Ein niedrigerer CL-Wert bedeutet bei gleichem Takt eine geringere Latenz.

Großen Einfluss auf die Leistung haben die **Speicherkanäle**. Der Prozessor kann über mehrere Kanäle gleichzeitig auf den Speicher zugreifen. Werden Module paarweise im *Dual-Channel*-Betrieb eingesetzt, verdoppelt sich die verfügbare Bandbreite grob. Server und Workstations arbeiten häufig mit vier oder mehr Kanälen (*Quad-Channel* und darüber).

ECC: Fehlerkorrektur für Server

Arbeitsspeicher ist nicht unfehlbar. Durch elektrische Störungen, kosmische Strahlung oder alternde Bauteile kann gelegentlich ein einzelnes Bit „kippen“ – aus einer Null wird eine Eins. Auf einem Arbeitsplatzrechner bleibt das meist folgenlos oder führt zu einem einmaligen Absturz. Auf einem Server, der rund um die Uhr Datenbanken, Dateien und virtuelle Maschinen verarbeitet, kann ein solcher Fehler jedoch unbemerkt Daten verfälschen, die anschließend dauerhaft gespeichert werden.

Hier setzt **ECC-RAM** an (Error-Correcting Code). Solche Module speichern zusätzliche Prüfinformationen, mit denen einzelne Bitfehler automatisch erkannt und korrigiert werden. Größere Fehler werden zumindest erkannt und gemeldet, statt still Daten zu beschädigen. Voraussetzung ist, dass Prozessor und Mainboard ECC unterstützen. Im typischen Desktop-Rechner ist das meist nicht der Fall; Server-Plattformen bringen die Unterstützung dagegen standardmäßig mit.

Ein häufiges Missverständnis betrifft DDR5: Diese Generation bringt eine interne Fehlerkorrektur auf dem Speicherchip mit (*On-Die-ECC*). Sie dient vor allem der Stabilität der Chips selbst und ersetzt kein vollwertiges ECC, das auch die Übertragung zwischen Modul und Prozessor absichert.

> **Merke**
>
> ECC ist im Server Pflicht. Es verhindert stille Datenverfälschung, funktioniert aber nur, wenn CPU, Mainboard und Module es gemeinsam unterstützen.

Weitere Bauformen

Neben dem Speichertyp unterscheidet sich RAM auch in der Bauform. Desktop-Rechner und Server verwenden lange Module, sogenannte **DIMMs**. In Notebooks kommen die kürzeren **SO-DIMMs** zum Einsatz. Besonders flache und sparsame Geräte nutzen **LPDDR** (Low Power DDR), das deutlich weniger Strom verbraucht, aber oft fest auf die Hauptplatine gelötet ist – ein späteres Aufrüsten ist dann nicht möglich. Achten Sie beim Kauf solcher Geräte deshalb darauf, von Anfang an genügend Speicher zu wählen.

Für Server mit sehr viel Arbeitsspeicher gibt es **Registered DIMMs** (RDIMMs). Sie enthalten einen zusätzlichen Baustein, der die Signale zwischenspeichert und den Speichercontroller entlastet. Dadurch bleiben auch voll bestückte Server mit vielen großen Modulen stabil.

Bauform	Einsatz	Besonderheit
DIMM	Desktop, Server	Standardmodul, steckbar
SO-DIMM	Notebooks, Mini-PCs	Kompakte Bauform
LPDDR	Ultrabooks, Mobilgeräte	Stromsparend, meist fest verlötet
RDIMM	Server mit viel RAM	Stabil bei hoher Bestückung, meist mit ECC

RAM richtig bestücken

Damit der Mehrkanalbetrieb funktioniert, müssen die Module in den richtigen Steckplätzen sitzen. Welche das sind, steht im Handbuch des Mainboards – die Reihenfolge ist nicht immer intuitiv. Kaufen Sie Module möglichst als **Kit**, also als zusammengehörigen Satz gleichen Typs mit identischen Timings. Module unterschiedlicher Hersteller oder Serien lassen sich zwar oft kombinieren, führen aber häufiger zu Instabilitäten oder zwingen das System auf die niedrigsten gemeinsamen Werte.

Viele Desktop-Module erreichen ihren beworbenen Takt erst, wenn ein entsprechendes Profil im UEFI aktiviert wird – bei Intel-Plattformen heißt es *XMP*, bei AMD *EXPO*. Ohne diese Einstellung laufen die Module mit einem langsameren Standardtakt. Und im Server gilt: ECC-Module nicht aus Kostengründen weglassen.

Speicherfehler erkennen

Defekter Arbeitsspeicher äußert sich oft durch zufällige Abstürze und Bluescreens ohne erkennbares Muster, beschädigte Dateien oder Programme, die sich unerklärlich beenden. Zur gründlichen Prüfung eignet sich ein Speichertest wie **Memtest86**, der von einem USB-Stick startet und den gesamten RAM über mehrere Durchläufe testet. Lassen Sie einen solchen Test auch nach jeder Aufrüstung laufen.

Findet der Test Fehler, prüfen Sie die Module einzeln, um den defekten Riegel zu identifizieren. Bei Servern mit ECC lohnt der Blick in das Protokoll des Management-Controllers (BMC): Dort werden korrigierte Speicherfehler protokolliert. Häufen sie sich bei einem Modul, sollte es getauscht werden, bevor aus korrigierbaren Fehlern ernsthafte werden.

Aus der Praxis

Treten Abstürze erst nach einer Aufrüstung oder nach dem Aktivieren von XMP/EXPO auf, testen Sie zuerst mit den Standardeinstellungen. Läuft das System dann stabil, liegt das Problem meist am zu hohen Takt oder an nicht zusammenpassenden Modulen.

Das Wichtigste in Kürze

- RAM ist schnell, aber flüchtig; zu wenig Speicher führt zu Auslagerung und Trägheit.
- Für die meisten Aufgaben geht Kapazität vor Takt.
- DDR4 und DDR5 sind nicht mischbar; Board und CPU bestimmen den Typ.
- ECC ist im Server Pflicht und braucht Unterstützung von CPU und Board.
- Im passenden Kanal-Layout als Kit bestücken und nach Aufrüstung testen.

Storage – HDD, SSD, NVMe

Datenträgertechnik von der Festplatte bis NVMe

Nach diesem Kapitel können Sie …

- HDD, SATA-SSD und NVMe-SSD unterscheiden und ihre Stärken benennen,
- Anschlüsse und Formfaktoren wie SATA, M.2 und U.2 zuordnen,
- Haltbarkeits- und Leistungskennzahlen wie TBW, DWPD und IOPS lesen,
- den passenden Datenträger für Arbeitsplatz, Server und Archiv auswählen.

Die Speicherhierarchie

Während der Arbeitsspeicher Daten nur vorübergehend hält, bewahren **Massenspeicher** sie dauerhaft auf – auch ohne Strom. Dafür sind sie langsamer als RAM. Innerhalb der Massenspeicher gibt es jedoch große Unterschiede: Die klassische **Festplatte (HDD)** arbeitet mechanisch und ist günstig pro Gigabyte. Die **SSD** speichert elektronisch in Flash-Zellen, ist schnell und robust. Und **NVMe** bezeichnet SSDs, die direkt über den PCIe-Bus angebunden sind – sie sind die schnellste Variante.

Welcher Datenträger der richtige ist, hängt davon ab, wie viel Kapazität benötigt wird, wie schnell die Zugriffe sein müssen und wie viel geschrieben wird. In der Praxis werden die Typen häufig kombiniert.

Die Festplatte: viel Platz für wenig Geld

Eine HDD besteht aus rotierenden Magnetscheiben, über die ein Schreib-Lese-Kopf fährt. Ihre große Stärke ist die Kapazität: Nirgends bekommen Sie so viel Speicherplatz für so wenig Geld. Deshalb sind Festplatten nach wie vor erste Wahl für Archive, Backups und große Datenablagen.

Die Mechanik bringt aber Nachteile mit sich. Bis der Kopf die richtige Stelle erreicht, vergeht Zeit, besonders bei vielen kleinen, verstreuten Zugriffen. Festplatten sind zudem empfindlich gegenüber Stößen und unterliegen mechanischem Verschleiß. Die **Drehzahl** beeinflusst das Tempo: Gängig sind 5.400 und 7.200 Umdrehungen pro Minute, im Server auch mehr. Für Netzwerkspeicher (NAS) gibt es spezielle Modelle, die auf Dauerbetrieb und den Einsatz in Gehäusen mit vielen Platten ausgelegt sind.

SSD und NVMe: Tempo ohne Mechanik

SSDs haben keine beweglichen Teile. Sie greifen nahezu verzögerungsfrei auf beliebige Stellen zu und sind unempfindlich gegen Erschütterungen. Der wichtigste Unterschied zwischen SSDs liegt in der Anbindung. Eine **SATA-SSD** nutzt dieselbe Schnittstelle wie Festplatten und ist damit überall einsetzbar, wird aber durch die Bandbreite von SATA begrenzt. Eine **NVMe-SSD** nutzt stattdessen PCIe und ist dadurch um ein Vielfaches schneller, sowohl beim Datendurchsatz als auch bei der Zahl gleichzeitiger Zugriffe.

Ein Hinweis zur Pflege: SSDs müssen nicht defragmentiert werden. Weil der Zugriff auf jede Speicherzelle gleich schnell ist, bringt das Umsortieren keinen Vorteil, kostet aber unnötig Schreibvorgänge. Moderne Betriebssysteme erkennen SSDs und führen stattdessen den *TRIM*-Befehl aus, der freie Bereiche an die SSD meldet.

Anschlüsse und Formfaktoren

Datenträger gibt es in verschiedenen Bauformen. Klassische Laufwerke mit SATA-Anschluss sind 3,5 Zoll (Festplatten in Desktop und Server) oder 2,5 Zoll (SSDs und Notebook-Festplatten) breit. **M.2** ist ein kompaktes Steckmodul, das direkt auf dem Mainboard sitzt und meist NVMe nutzt – es gibt allerdings auch M.2-Module mit SATA-Anbindung, was beim Kauf beachtet werden sollte. **U.2** ist eine im Server verbreitete Bauform für NVMe-Laufwerke im 2,5-Zoll-Format, die sich im laufenden Betrieb tauschen lassen.

Weil NVMe-SSDs PCIe-Lanes belegen, sollten Sie im Handbuch des Mainboards prüfen, wie viele Lanes zur Verfügung stehen. Häufig teilen sich M.2-Steckplätze Bandbreite mit anderen Anschlüssen, sodass ein belegter M.2-Slot einen PCIe-Steckplatz oder SATA-Ports deaktiviert.

Haltbarkeit und Leistung messen

Flash-Zellen vertragen nur eine begrenzte Zahl von Schreib- und Löschvorgängen, die sogenannten **P/E-Zyklen** (Program/Erase). Die SSD verteilt die Schreibzugriffe deshalb gleichmäßig über alle Zellen – das nennt man *Wear-Leveling*. Hersteller geben die Haltbarkeit meist als **TBW** (Terabytes Written) an, also die Datenmenge, die über die gesamte Lebensdauer geschrieben werden darf. Im Enterprise-Bereich ist auch **DWPD** üblich (Drive Writes Per Day): Dieser Wert sagt aus, wie oft die gesamte Kapazität während der Garantiezeit pro Tag überschrieben werden darf.

Für die Leistung ist neben dem Durchsatz in MB/s vor allem **IOPS** wichtig – die Zahl der Ein- und Ausgabeoperationen pro Sekunde. Datenbanken und Virtualisierung erzeugen viele kleine Zugriffe, und hier spielen NVMe-SSDs ihre Stärke aus.

Auch die Art der Flash-Zellen macht einen Unterschied. **TLC**-Zellen speichern drei Bit pro Zelle, **QLC**-Zellen vier. QLC erlaubt günstigere und größere SSDs, verträgt aber weniger Schreibvorgänge. Als grobe Größenordnung hält eine TLC-SSD mit einem Terabyte

oft etwa 600 TBW aus, eine vergleichbare QLC-SSD häufig nur 200 bis 400 TBW. Für Datenbanken oder virtuelle Desktops gehören deshalb TLC-SSDs ins System; QLC eignet sich für große Ablagen, die überwiegend gelesen werden.

Typ	Stärke	Typischer Einsatz
HDD	Viel Kapazität, günstig pro GB	Archiv, Backup, große Ablagen
SATA-SSD	Schnell, universell einsetzbar	Arbeitsplätze, Aufrüstung älterer Systeme
NVMe-SSD	Höchster Durchsatz und viele IOPS	System, Datenbanken, Virtualisierung

Auswahl nach Einsatz und Pflege im Betrieb

Consumer-SSDs sind für Desktop und Notebook mit moderater Schreiblast gedacht. **Enterprise- und NAS-Laufwerke** sind dagegen auf Dauerbetrieb ausgelegt, bieten eine deutlich höhere TBW und verfügen oft über eine *Power-Loss-Protection*: Kondensatoren liefern bei einem Stromausfall genug Energie, um Daten aus dem Zwischenspeicher noch sicher zu schreiben. Für Server sollten Sie solche Laufwerke wählen; insbesondere Datenbanken und Virtualisierung profitieren stark von NVMe.

Eine bewährte Aufteilung lautet: Betriebssystem und aktive Daten auf SSD oder NVMe, große Archive auf HDD. Behalten Sie außerdem die **SMART-Werte** im Blick. Diese Selbstdiagnose der Laufwerke meldet unter anderem Temperatur, Betriebsstunden, Verschleiß und die Zahl der Ersatzsektoren. Steigt die Zahl umgelagerter Sektoren oder sinkt die Reserve einer SSD deutlich, ist das eine Frühwarnung – tauschen Sie den Datenträger rechtzeitig. Halten Sie zudem die Firmware der Laufwerke aktuell, denn Updates beheben gelegentlich Fehler, die zu Datenverlust führen können.

> **Merke**
>
> Ein RAID schützt vor dem Ausfall einzelner Datenträger, ersetzt aber kein Backup. Gelöschte, verschlüsselte oder beschädigte Daten sind auf allen Platten eines RAID gleichzeitig betroffen. Mehr dazu in Lernbereich 6.

Das Wichtigste in Kürze

- HDD ist günstig und groß, SSD schnell und robust, NVMe am schnellsten.
- NVMe nutzt PCIe statt SATA; M.2 und U.2 sind die typischen Bauformen.
- TBW, DWPD, P/E-Zyklen und IOPS beschreiben Haltbarkeit und Leistung.
- Für Dauerlast Enterprise-SSDs mit TLC und Power-Loss-Protection wählen.
- System auf SSD, Archiv auf HDD – und SMART-Werte regelmäßig prüfen.

Mainboard & Formfaktoren

Die Platine, die alles verbindet

Nach diesem Kapitel können Sie …

- die Rolle von Mainboard, Sockel und Chipsatz beschreiben,
- die gängigen Formfaktoren ATX, microATX und Mini-ITX unterscheiden,
- PCIe-Generationen und -Lanes richtig einordnen,
- BIOS und UEFI vergleichen und die Bedeutung von Secure Boot und TPM erklären,
- ein Mainboard passend zu CPU, Ausbau und Zukunftsplanung auswählen.

Die Rolle des Mainboards

Das **Mainboard**, auch Hauptplatine genannt, ist die zentrale Leiterplatte eines Rechners. Es verbindet alle Komponenten miteinander: den Prozessor im Sockel, den Arbeitsspeicher in seinen Steckplätzen, die Datenträger und sämtliche Erweiterungskarten. Über die Leiterbahnen des Boards laufen die Daten zwischen diesen Bauteilen, und es verteilt die Stromversorgung an alle Komponenten.

Zwei Eigenschaften des Mainboards sind besonders wichtig. Der **Sockel** legt fest, welche Prozessoren überhaupt verwendet werden können. Der **Chipsatz** bestimmt den Funktionsumfang: wie viele USB- und SATA-Anschlüsse zur Verfügung stehen, wie viele zusätzliche PCIe-Lanes es gibt und ob Funktionen wie das Übertakten möglich sind. Zusammen entscheiden Sockel und Chipsatz damit auch darüber, wie gut sich ein System später aufrüsten lässt.

Formfaktoren: Größe bestimmt Ausbau

Der **Formfaktor** beschreibt Größe, Form und die Lage der Befestigungspunkte eines Mainboards. Er entscheidet, welches Gehäuse passt und wie viele Steckplätze Platz finden. Der verbreitete Standard für Desktops und Workstations ist **ATX**. Die etwas kleinere Variante **microATX** bietet weniger Erweiterungssteckplätze, passt dafür in kompaktere Gehäuse. **Mini-ITX** ist sehr klein und kommt in Mini-PCs, kompakten Arbeitsplätzen und Homelabs zum Einsatz, hat aber meist nur einen einzigen Erweiterungssteckplatz und zwei RAM-Slots.

Die Grundregel lautet: Je kleiner das Board, desto weniger Steckplätze und Anschlüsse. Server-Mainboards folgen häufig eigenen Standards oder herstellerspezifischen Formaten, weil sie auf bestimmte Gehäuse, Kühlkonzepte und Einbautiefen im Rack zuge-

schnitten sind.

Formfaktor	Größe	Typischer Einsatz
ATX	Standard, viele Steckplätze	Desktops, Workstations
microATX	Kompakter, weniger Slots	Büro-PCs, kleinere Gehäuse
Mini-ITX	Sehr klein, meist ein Slot	Mini-PCs, Homelabs
Server-Formate	Herstellerspezifisch	Rack- und Tower-Server

PCI Express: die Datenautobahn für Erweiterungen

PCI Express (PCIe) ist das Bussystem, über das Grafikkarten, NVMe-SSDs, Netzwerkkarten und viele weitere Erweiterungen angebunden werden. Die Bandbreite einer Verbindung hängt von zwei Faktoren ab: von der *Generation* und der Zahl der *Lanes*. Jede neue PCIe-Generation verdoppelt ungefähr die Übertragungsrate pro Lane. Die Lanes sind einzelne Datenleitungen, die gebündelt werden – üblich sind x1, x4, x8 und x16. Eine Grafikkarte nutzt typischerweise x16, eine NVMe-SSD x4.

PCIe ist abwärtskompatibel: Eine Karte der neueren Generation funktioniert in einem älteren Steckplatz, dann allerdings nur mit dessen Geschwindigkeit. Wichtig ist außerdem, dass die verfügbaren Lanes begrenzt sind. Prozessor und Chipsatz stellen nur eine bestimmte Anzahl bereit, und viele Boards teilen sie zwischen Steckplätzen auf. Ein mechanischer x16-Slot ist deshalb nicht immer auch elektrisch mit 16 Lanes angebunden, und ein belegter M.2-Steckplatz kann einen anderen Slot ausbremsen oder abschalten. Die genaue Aufteilung steht im Handbuch.

Anschlüsse und Header auf der Platine

Neben den großen Steckplätzen bietet ein Mainboard zahlreiche kleinere Anschlüsse. Für Datenträger stehen SATA-Ports und M.2-Steckplätze zur Verfügung. Sogenannte **Header** sind Stiftleisten, an die Gehäuseanschlüsse angeschlossen werden: zusätzliche USB-Buchsen, die Front-Panel-Kontakte für Ein-Schalter und Status-LEDs sowie die Lüfter. Lüfter-Header mit *PWM-Steuerung* erlauben es, die Drehzahl je nach Temperatur zu regeln – das sorgt für einen leisen Betrieb bei geringer Last. Planen Sie bei der Auswahl genug Anschlüsse für künftige Erweiterungen ein, denn nachträglich lässt sich nur wenig ergänzen.

Firmware: vom BIOS zum UEFI

Beim Einschalten startet zuerst die **Firmware** des Mainboards. Sie prüft die Hardware, initialisiert die Komponenten und übergibt dann an den Bootloader des Betriebssystems. Früher übernahm das das **BIOS**: eine schlichte Textoberfläche, die mit der älteren MBR-

Partitionierung arbeitete. MBR ist auf Datenträger von gut zwei Terabyte begrenzt.

Heute ist **UEFI** Standard. Es bietet häufig eine grafische Oberfläche mit Mausbedienung, unterstützt die modernere **GPT**-Partitionierung für sehr große Datenträger, startet schneller und bringt Sicherheitsfunktionen mit. Firmware-Updates sind nicht nur für die Unterstützung neuer Prozessoren wichtig, sondern schließen auch Sicherheitslücken. Behandeln Sie die Firmware deshalb genauso wie das Betriebssystem: als etwas, das regelmäßig aktualisiert werden muss.

Secure Boot und TPM

Secure Boot ist eine Funktion des UEFI, die beim Start nur Bootloader mit gültiger digitaler Signatur zulässt. Schadsoftware, die sich vor dem Betriebssystem in den Startvorgang einnistet – sogenannte Bootkits –, wird dadurch deutlich erschwert. Ergänzt wird Secure Boot durch das **TPM** (Trusted Platform Module), einen Sicherheitschip oder eine in den Prozessor integrierte Firmware-Lösung, die Schlüssel sicher speichert und den Startvorgang vermessen kann. Funktionen wie die Laufwerksverschlüsselung mit BitLocker nutzen das TPM.

Windows 11 setzt ein UEFI-System mit Secure-Boot-Fähigkeit und TPM 2.0 voraus. Bei manchen Linux-Installationen muss Secure Boot angepasst werden, etwa wenn eigene oder nicht signierte Treiber geladen werden sollen. Beide Funktionen sind Teil der UEFI-Konfiguration.

> **Merke**
>
> UEFI mit GPT hat BIOS mit MBR abgelöst. Secure Boot und TPM 2.0 bilden die Grundlage für einen vertrauenswürdigen Systemstart und sind für aktuelle Windows-Versionen Voraussetzung.

Aufrüsten und Kompatibilität

Wählen Sie ein Board immer passend zu Sockel und Chipsatz des gewünschten Prozessors. Achten Sie außerdem auf ausreichend PCIe- und M.2-Steckplätze für die Zukunft und prüfen Sie die **Kompatibilitätsliste** des Herstellers (QVL, Qualified Vendor List). Sie zeigt, welche Prozessoren, Speichermodule und teilweise auch Datenträger getestet wurden.

Ein klassischer Stolperstein: Neue CPU-Generationen werden von einem Board oft erst nach einem UEFI-Update erkannt. Das Update muss dann *vor* dem Prozessorwechsel eingespielt werden – mit dem alten Prozessor oder über eine Notfallfunktion, die manche Boards für Updates ohne eingesetzte CPU anbieten.

Aus der Praxis

Aktualisieren Sie die Firmware neuer Systeme vor der Inbetriebnahme. Unterbrechen Sie ein laufendes Firmware-Update niemals – ein Stromausfall in diesem Moment kann das Board unbrauchbar machen. Bei Servern empfiehlt sich dafür ein Zeitpunkt mit gesicherter Stromversorgung.

Das Wichtigste in Kürze

- Sockel und Chipsatz bestimmen, welche CPU passt und was das Board kann.
- Der Formfaktor legt Größe, Steckplätze und passende Gehäuse fest.
- PCIe-Lanes sind begrenzt und werden oft zwischen Slots geteilt.
- UEFI mit GPT statt BIOS mit MBR; Secure Boot und TPM für sicheren Start.
- Vor CPU-Wechsel und Inbetriebnahme die Firmware aktualisieren.

Server-Hardware

Was Server von Desktops unterscheidet

Nach diesem Kapitel können Sie …

- erklären, worin sich Server grundlegend von Arbeitsplatzrechnern unterscheiden,
- die Bauformen Rack, Tower und Blade einordnen und passend auswählen,
- Redundanz und Hot-Swap als Grundlage der Ausfallsicherheit beschreiben,
- Out-of-Band-Management mit iLO, iDRAC und IPMI erklären und absichern,
- die wichtigsten Anforderungen an Kühlung und Stromversorgung benennen.

Ein Server ist kein großer PC

Auf den ersten Blick enthält ein Server dieselben Bauteile wie ein Desktop-Rechner: Prozessor, Arbeitsspeicher, Datenträger und Netzwerkkarte. Der entscheidende Unterschied liegt jedoch nicht in der reinen Rechenleistung, sondern im Anspruch an Zuverlässigkeit. Ein Server ist auf **Dauerbetrieb** rund um die Uhr über viele Jahre ausgelegt. Von ihm hängen oft ganze Abteilungen oder das gesamte Unternehmen ab – fällt er aus, steht die Arbeit still.

Daraus ergeben sich drei Kernmerkmale. Server sind *redundant* aufgebaut, damit der Ausfall eines Bauteils nicht zum Stillstand führt. Sie lassen sich *aus der Ferne warten*, auch wenn das Betriebssystem nicht mehr reagiert. Und sie sind dafür gemacht, in großer Zahl *zentral verwaltet* zu werden. Hinzu kommen serverspezifische Komponenten wie ECC-Arbeitsspeicher, Enterprise-Datenträger und Prozessoren mit vielen Kernen.

Bauformen: Rack, Tower und Blade

Der **Rack-Server** ist der Standard im Rechenzentrum. Er ist flach gebaut und wird in einen genormten 19-Zoll-Schrank geschraubt. Seine Höhe wird in *Höheneinheiten* angegeben: Eine Einheit (1U) entspricht 44,45 Millimetern. Gängig sind Server mit 1U oder 2U; höhere Modelle bieten Platz für mehr Datenträger oder Grafikkarten. Racks sparen Platz, erleichtern eine saubere Verkabelung und bündeln Strom, Netzwerk und Kühlung an einem Ort.

Der **Tower-Server** sieht aus wie ein großer Desktop-PC und steht auf dem Boden. Er eignet sich für kleine Standorte, an denen kein Serverschrank vorhanden ist, und ist meist leiser als ein Rack-Server. **Blade-Server** schließlich sind schmale Einschübe, die

gemeinsam in einem Gehäuse (Chassis) stecken und sich Stromversorgung, Kühlung und Netzwerkanbindung teilen. Sie erreichen eine sehr hohe Dichte und lohnen sich dort, wo viele Server auf engem Raum betrieben werden.

Bauform	Stärke	Typischer Einsatz
Rack (1U/2U)	Platzsparend, standardisiert	Rechenzentrum, Serverraum mit Schrank
Tower	Kein Rack nötig, leiser	Kleine Standorte, Filialen
Blade	Sehr hohe Dichte, gemeinsame Infrastruktur	Große Umgebungen mit vielen Servern

Die Bauform sollte sich nach Platz, benötigter Dichte und Budget richten – nicht nach dem Prospekt. Für die meisten mittelständischen Umgebungen ist der Rack-Server die naheliegende Wahl, sobald ein Serverschrank vorhanden ist.

Redundanz und Wartung im laufenden Betrieb

Redundanz bedeutet, dass kritische Komponenten doppelt oder mehrfach vorhanden sind. Fällt eine aus, übernimmt die andere, und der Server läuft weiter. Typisch sind zwei **Netzteile**, die idealerweise an getrennte Stromkreise angeschlossen werden, ein **RAID**-Verbund für die Datenträger und mehrere **Netzwerkpfade**, sodass auch ein defektes Kabel oder ein ausgefallener Switch-Port die Verbindung nicht unterbricht.

Viele dieser Komponenten sind **Hot-Swap**-fähig: Festplatten und Netzteile lassen sich im laufenden Betrieb tauschen, ohne den Server herunterzufahren. Werkzeuglose Gehäuse erleichtern den Austausch zusätzlich, und zahlreiche Sensoren überwachen Temperaturen und Lüfter. Damit die Redundanz im Ernstfall auch funktioniert, sollte sie regelmäßig getestet werden – etwa indem bewusst ein Netzteil gezogen wird. Kritische Ersatzteile wie Festplatten oder Netzteile vorrätig zu halten, verkürzt die Zeit, in der ein Server ohne Reserve läuft.

Out-of-Band-Management: Fernwartung ohne Betriebssystem

Eine der wichtigsten Server-Funktionen ist das **Out-of-Band-Management**. Dafür besitzt der Server einen eigenen kleinen Steuerrechner, den *Baseboard Management Controller* (BMC), der unabhängig vom Hauptsystem arbeitet und meist über einen eigenen Netzwerkanschluss erreichbar ist. Bei HPE heißt diese Lösung **iLO**, bei Dell **iDRAC**; der herstellerübergreifende Standard dafür ist **IPMI**.

Über den Management-Controller können Sie den Server aus der Ferne ein- und ausschalten, auf die Bildschirmkonsole zugreifen, Installationsmedien einbinden und Sensorwerte sowie Fehlerprotokolle auslesen – selbst dann, wenn das Betriebssystem hängt oder noch gar nicht installiert ist. Das spart Fahrten zum Standort und beschleunigt die Fehlersuche erheblich.

> **Merke**
>
> Wer Zugriff auf den Management-Controller hat, hat die volle Kontrolle über den Server. Das Management-Netz gehört deshalb strikt vom Produktivnetz getrennt, die Zugangsdaten müssen geändert und die Firmware des BMC aktuell gehalten werden.

Kühlung und Stromversorgung

Server erzeugen viel Wärme und sind auf eine durchdachte Kühlung angewiesen. Im Rechenzentrum hat sich die Trennung von **Kalt- und Warmgang** bewährt: Alle Server saugen kühle Luft von der Vorderseite an und blasen die warme Luft nach hinten aus. Die Racks werden so aufgestellt, dass sich kalte und warme Luft nicht vermischen. Ein freier Luftweg und funktionierende Lüfter sind die Voraussetzung dafür, dass Komponenten nicht überhitzen.

Bei der Stromversorgung sind neben redundanten, Hot-Swap-fähigen Netzteilen zwei weitere Bausteine wichtig. Eine **USV** (unterbrechungsfreie Stromversorgung) überbrückt kurze Stromausfälle und ermöglicht bei längeren Ausfällen ein geordnetes Herunterfahren. Und eine **PDU** (Power Distribution Unit) verteilt den Strom im Rack; Modelle mit Monitoring zeigen den Verbrauch je Anschluss an und warnen vor Überlast.

Der Markt für Server-Hardware

Im Geschäft mit Systemhäusern und mittelständischen Kunden gehören **Dell** und **HPE** traditionell zu den führenden Anbietern. **Supermicro** ist vor allem durch die starke Nachfrage nach GPU- und KI-Servern gewachsen. Weitere bekannte Größen sind **Lenovo** mit einem breiten Portfolio, **Cisco** mit seiner UCS-Serie sowie **Fujitsu**, das vor allem in Europa verbreitet ist. Marktanteile verschieben sich allerdings laufend.

Ein Detail am Rande: Die großen Cloud-Anbieter, die sogenannten Hyperscaler, kaufen ihre Server häufig nicht bei Markenherstellern, sondern direkt bei Auftragsfertigern, den *ODMs* (Original Design Manufacturers), und lassen nach eigenen Vorgaben bauen. Für kleine und mittlere Unternehmen ist dieses Modell nicht relevant – hier zählen Markenhersteller mit verlässlichem Support, Ersatzteilversorgung und Management-Werkzeugen.

> **Aus der Praxis**
>
> Achten Sie beim Serverkauf nicht nur auf die Hardware, sondern auch auf den Support-Vertrag: Reaktionszeit, Vor-Ort-Service und Laufzeit entscheiden im Störungsfall darüber, wie lange ein System ausfällt.

Das Wichtigste in Kürze

- Server sind auf Dauerbetrieb, Redundanz und Fernwartung ausgelegt.
- Rack ist Standard im Rechenzentrum, Tower für kleine Standorte, Blade für hohe Dichte.
- Redundante Netzteile, RAID und mehrere Netzwerkpfade halten den Betrieb aufrecht.
- iLO, iDRAC und IPMI ermöglichen Out-of-Band-Management – streng abgesichert.
- Kalt-/Warmgang, USV und überwachte PDUs sichern Kühlung und Strom.

Netzwerkkarten & Verkabelung

Vom NIC bis zum Patchpanel

Nach diesem Kapitel können Sie …

- Netzwerkkarten nach Geschwindigkeit und Anschluss auswählen,
- Kupfer- und Glasfaserverbindungen sowie Single-Mode und Multi-Mode unterscheiden,
- die passende Kabelkategorie für die gewünschte Geschwindigkeit bestimmen,
- die Grundsätze strukturierter Verkabelung anwenden und typische Fehler vermeiden.

Die Netzwerkkarte als Schnittstelle

Die **Netzwerkkarte**, häufig mit NIC (Network Interface Card) abgekürzt, verbindet ein Gerät mit dem Netzwerk. In Arbeitsplatzrechnern und Notebooks ist sie meist fest auf dem Mainboard verbaut, in Servern kommen zusätzlich steckbare Karten über PCIe zum Einsatz. Wichtigstes Merkmal ist die Geschwindigkeit: Am Arbeitsplatz ist 1 Gigabit pro Sekunde (1G) verbreitet, im Server sind 10G und 25G üblich, im Rechenzentrum auch 40G, 100G und mehr.

Server besitzen häufig mehrere Netzwerkanschlüsse. Diese lassen sich zu einer logischen Verbindung bündeln – unter Windows spricht man von **Teaming**, unter Linux von **Bonding**. Die Bündelung erfüllt zwei Zwecke: Sie erhöht die verfügbare Bandbreite, und sie sorgt für Ausfallsicherheit, weil die Verbindung bestehen bleibt, wenn ein Kabel oder Port ausfällt. Damit das funktioniert, muss die Gegenseite – meist ein Switch – passend konfiguriert sein.

Kupfer oder Glasfaser?

Für die physische Verbindung gibt es zwei grundlegende Medien. **Kupferkabel** mit dem bekannten RJ45-Stecker sind günstig, einfach zu verlegen und überall verbreitet. Ihre Reichweite ist allerdings begrenzt: Eine Kupferstrecke darf in der Regel höchstens 100 Meter lang sein. Außerdem sind sie anfälliger für elektrische Störungen.

Glasfaser überträgt Daten als Lichtsignale. Sie überbrückt große Distanzen, erreicht sehr hohe Geschwindigkeiten und ist völlig unempfindlich gegenüber elektromagnetischen Störungen. Angeschlossen wird Glasfaser über steckbare Module, die **SFP**- bzw. **SFP+**-Transceiver, die in entsprechende Ports an Switches und Netzwerkkarten einge-

setzt werden. Die Module müssen zur gewünschten Geschwindigkeit, Reichweite und zum Fasertyp passen. Im Rechenzentrum und für die Verbindungen zwischen Etagen und Gebäuden – das *Backbone* – dominiert Glasfaser.

Glasfaser-Typen: Single-Mode und Multi-Mode

Bei Glasfasern werden zwei Typen unterschieden. **Multi-Mode**-Fasern haben einen dickeren Kern, in dem sich das Licht auf mehreren Wegen ausbreitet. Sie eignen sich für kürzere Strecken innerhalb eines Gebäudes oder Rechenzentrums und sind meist günstiger in der Anschlusstechnik. **Single-Mode**-Fasern haben einen sehr dünnen Kern, in dem das Licht nur auf einem Weg läuft. Dadurch erreichen sie wesentlich größere Distanzen und werden für Backbones und Standortverbindungen eingesetzt.

Fasertyp und SFP-Modul müssen zusammenpassen; ein Single-Mode-Modul arbeitet nicht zuverlässig an einer Multi-Mode-Faser. Verbreitete Steckertypen sind *LC* (klein, heute Standard im Rechenzentrum) und *SC* (größer, oft in älteren Installationen).

Medium	Stärke	Typischer Einsatz
Kupfer (RJ45)	Günstig, einfach, bis ca. 100 m	Arbeitsplätze, Verbindungen im Rack
Multi-Mode-Glasfaser	Schnell, störungsfrei, kürzere Strecken	Innerhalb von Gebäude und Rechenzentrum
Single-Mode-Glasfaser	Sehr große Reichweite	Backbone, Standortverbindungen

Kabelkategorien bei Kupfer

Kupfer-Netzwerkkabel werden in **Kategorien** (Cat) eingeteilt, die angeben, welche Frequenzen und damit welche Geschwindigkeiten sie zuverlässig übertragen. **Cat5e** reicht für 1 Gigabit. Für 10 Gigabit sollten Sie eher **Cat6** oder besser **Cat6a** verwenden – Cat6 schafft 10G nur auf kürzeren Strecken, Cat6a auf den vollen 100 Metern. **Cat7** und **Cat8** sind für höchste Ansprüche gedacht; Cat8 ist für sehr kurze, schnelle Verbindungen etwa im Rechenzentrum ausgelegt.

Grundsätzlich gilt: Je höher die Kategorie, desto mehr Bandbreite und meist auch desto bessere Schirmung gegen Störungen. Die Kategorie allein garantiert aber keine Leistung. Qualität der Kabel und Stecker sowie eine fachgerechte Verlegung haben ebenso großen Einfluss. Eine Strecke ist immer nur so gut wie ihr schwächstes Glied – ein einzelnes minderwertiges Patchkabel kann eine hochwertige Installation ausbremsen.

Strukturierte Verkabelung

In professionellen Umgebungen werden Netzwerkkabel nicht direkt vom Arbeitsplatz zum Switch gezogen. Stattdessen setzt man auf eine **strukturierte Verkabelung**: Fest verlegte Leitungen führen von den Netzwerkdosen in den Räumen zu einem **Patchpanel** im Serverschrank. Das Patchpanel ist der zentrale Rangierpunkt. Von dort aus werden kurze *Patchkabel* zum Switch gesteckt. Soll ein Arbeitsplatz einem anderen Port zugeordnet werden, genügt es, ein Patchkabel umzustecken – die fest verlegte Leitung bleibt unangetastet.

Damit das funktioniert, ist eine konsequente **Beschriftung** Pflicht: Jede Dose, jeder Patchpanel-Port und jedes Kabel braucht eine eindeutige Bezeichnung, die im gesamten Rack einheitlich ist und in einer Dokumentation festgehalten wird. Ohne Beschriftung wird jede Fehlersuche zur zeitraubenden Suche nach dem richtigen Kabel. Eine saubere Kabelführung im Rack mit Kabelmanagement-Schienen sorgt zudem für Übersicht und freie Luftwege.

> **Merke**
>
> Struktur und Beschriftung sparen später viel Zeit. Ein Rack, in dem jedes Kabel eindeutig bezeichnet und dokumentiert ist, lässt sich auch von Kolleginnen und Kollegen warten, die es nicht aufgebaut haben.

Häufige Fehler und gute Praxis

Viele Netzwerkprobleme haben eine physische Ursache. Zu den häufigsten Fehlern gehören zu lange oder minderwertige Kabel, Knicke und zu enge Biegeradien, die vor allem Glasfasern beschädigen, sowie fehlende Dokumentation. Ein weiterer Klassiker ist ein **Duplex- oder Speed-Mismatch**: Sind beide Seiten einer Verbindung unterschiedlich eingestellt – etwa eine fest auf eine Geschwindigkeit, die andere auf automatische Aushandlung –, entstehen Übertragungsfehler und die Verbindung wird langsam oder instabil.

Nach dem Verlegen sollten neue Leitungen mit einem Messgerät **zertifiziert** werden. Die Messung belegt, dass die Strecke die Anforderungen ihrer Kategorie tatsächlich erfüllt. Planen Sie außerdem Reserve-Adern und freie Ports ein, damit spätere Erweiterungen ohne neue Verlegung möglich sind.

Aus der Praxis

Glasfaserstecker reagieren empfindlich auf Staub und Fingerabdrücke. Verschließen Sie ungenutzte Stecker und Ports mit Schutzkappen und reinigen Sie Stecker vor dem Einstecken. Verschmutzte Stirnflächen sind eine häufige Ursache für hohe Fehlerraten auf Glasfaserstrecken.

Das Wichtigste in Kürze

- Die NIC-Geschwindigkeit passend zur Last wählen; Teaming/Bonding für Bandbreite und Redundanz.
- Kupfer ist kurz und günstig, Glasfaser weit, schnell und störungsfrei.
- Multi-Mode für kurze Strecken, Single-Mode für große Distanzen.
- Die Kabelkategorie bestimmt die mögliche Bandbreite, die Verlegung die tatsächliche.
- Patchpanel, Beschriftung und Zertifizierung sparen später viel Zeit.

Hardware-Diagnose & Lifecycle

Fehler finden und Hardware planvoll ablösen

Nach diesem Kapitel können Sie …

- Hinweise aus POST, Piepcodes, LEDs und Protokollen auswerten,
- passende Diagnosewerkzeuge für Arbeitsspeicher und Datenträger einsetzen,
- typische Symptome wahrscheinlichen Ursachen zuordnen und Fehler systematisch eingrenzen,
- Hardware über ihren gesamten Lebenszyklus planen und sicher außer Betrieb nehmen.

Der erste Hinweis: POST und Boot

Jedes Mal, wenn ein Rechner eingeschaltet wird, führt die Firmware einen **Power-On Self-Test** (POST) durch. Dabei prüft sie, ob Prozessor, Arbeitsspeicher, Grafik und weitere grundlegende Komponenten vorhanden sind und reagieren. Erst wenn dieser Test erfolgreich war, wird das Betriebssystem gestartet.

Schlägt der POST fehl, meldet das System den Fehler auf einfachem Weg: über *Piepcodes*, also eine Folge kurzer und langer Töne, über Status-LEDs auf dem Mainboard oder über Fehlercodes in einem kleinen Display. Die Bedeutung steht im Handbuch des Herstellers. Bleibt der Bildschirm komplett schwarz, sind Prozessor, Arbeitsspeicher und Grafikkarte die ersten Verdächtigen. Bei Servern liefert zusätzlich das Protokoll des Management-Controllers (BMC) wertvolle Informationen, denn dieser zeichnet Hardwarefehler auch dann auf, wenn das System nicht startet.

Bewährte Diagnose-Werkzeuge

Für gezielte Tests gibt es eine Reihe bewährter Werkzeuge. **Memtest86** prüft den Arbeitsspeicher gründlich auf Fehler. Es startet von einem USB-Stick, also unabhängig vom installierten Betriebssystem, und sollte mehrere Durchläufe absolvieren, da manche Fehler nur sporadisch auftreten.

Für Datenträger lesen **CrystalDiskInfo** unter Windows oder `smartctl` aus dem Paket smartmontools unter Linux die SMART-Werte aus. Diese Selbstdiagnose zeigt unter anderem umgelagerte Sektoren, Lesefehler, Temperatur und bei SSDs den Verschleißgrad. Ein kurzer Aufruf genügt:

```
smartctl -a /dev/sda
```

Server- und Business-Hersteller wie Dell, HPE oder Lenovo bringen zudem eigene **Diagnosewerkzeuge** mit, die oft direkt aus der Firmware oder dem Management-Controller gestartet werden und alle Komponenten systematisch prüfen. Ein bewährter Trick ist außerdem der Einsatz eines *Live-Systems* vom USB-Stick: Läuft der Rechner damit stabil, liegt das Problem eher am installierten Betriebssystem oder an Treibern als an der Hardware.

Typische Symptome richtig deuten

Viele Hardwareprobleme zeigen wiederkehrende Muster. **Zufällige Abstürze** ohne erkennbaren Zusammenhang deuten häufig auf fehlerhaften Arbeitsspeicher, ein schwächelndes Netzteil oder Überhitzung hin. **Langsame Zugriffe** und hängende Programme können auf einen sterbenden Datenträger hinweisen – hier helfen die SMART-Werte weiter. **Bluescreens** können sowohl von Treibern als auch von Hardware verursacht werden; der angezeigte Fehlercode und die Ereignisanzeige geben Hinweise auf die Ursache.

Bricht ein System unter Last plötzlich in der Leistung ein, steckt oft **Thermal Throttling** dahinter: Der Prozessor drosselt seinen Takt, um sich vor Überhitzung zu schützen. Ursache sind meist verstaubte Kühler, ausgefallene Lüfter oder eingetrocknete Wärmeleitpaste.

Symptom	Häufige Ursachen	Erster Prüfschritt
Zufällige Abstürze	RAM, Netzteil, Temperatur	Memtest86, Sensorwerte
Langsame Zugriffe	Alternder Datenträger	SMART-Werte auslesen
Bluescreens	Treiber oder Hardware	Fehlercode, Ereignisanzeige
Leistungseinbruch unter Last	Thermal Throttling durch Staub, Kühlung	Temperaturen prüfen, reinigen

Fehler systematisch eingrenzen

Bei unklaren Fehlern hilft nur ein systematisches Vorgehen. Lesen Sie zuerst die vorhandenen **Protokolle** – das BMC-Log beim Server, die Ereignisanzeige unter Windows oder das Systemjournal unter Linux. Oft steht dort bereits, welche Komponente auffällig ist.

Hilft das nicht weiter, wird getauscht und reduziert. Bewährt hat sich, das System mit *minimaler Bestückung* zu starten – also nur mit Prozessor, einem Speichermodul und der nötigsten Hardware – und dann Komponente für Komponente zu ergänzen, bis der Fehler wieder auftritt. Entscheidend ist dabei, immer nur **eine Variable** gleichzeitig zu ändern. Wer drei Dinge auf einmal tauscht, weiß hinterher nicht, welche Maßnahme geholfen hat. Ist die verdächtige Komponente gefunden, wird sie isoliert getestet, etwa in einem anderen System.

> **Merke**
>
> Erst Logs lesen, dann systematisch eingrenzen – und immer nur eine Sache auf einmal ändern. So finden Sie die Ursache, statt nur das Symptom zu beseitigen.

Monitoring und vorbeugende Wartung

Viele Ausfälle kündigen sich an. Systeme, die ungewöhnlich warm werden oder deren Lüfter lauter laufen als sonst, haben oft ein beginnendes Problem. Deshalb sollten Temperaturen, Lüfterdrehzahlen und Spannungen dauerhaft überwacht werden. Bei Datenträgern lohnt es sich, die SMART-Werte im *Trend* zu beobachten statt nur als Momentaufnahme: Ein einzelner umgelagerter Sektor ist kein Grund zur Sorge, eine stetig steigende Zahl dagegen schon. Für kritische Werte sollten automatische Alarme eingerichtet werden; mehr dazu in Lernbereich 9.

Ergänzend hilft vorbeugende Wartung: Staub regelmäßig entfernen und Luftwege freihalten, bei älteren Systemen die Wärmeleitpaste erneuern, Firmware und BIOS aktuell halten und kritische Ersatzteile vorrätig haben.

Lifecycle-Management und Außerbetriebnahme

Hardware hat eine begrenzte, planbare Lebensdauer. Hersteller kündigen dafür feste Termine an: **EOL** (End of Life) bezeichnet das Ende des Verkaufs, **EOSL** (End of Service Life) das Ende von Support, Ersatzteilen und Firmware-Updates. Wer diese Termine kennt, kann Ersatz rechtzeitig planen und budgetieren. Server werden typischerweise nach etwa fünf Jahren erneuert; dieser Zeitraum orientiert sich häufig an der Laufzeit von Garantie- und Supportverträgen, die deshalb ebenfalls im Blick behalten werden sollten.

Auch das Ende eines Geräts muss geordnet ablaufen. Datenträger werden **sicher gelöscht oder physisch vernichtet**, damit keine Unternehmens- oder personenbezogenen Daten in fremde Hände geraten. Die Geräte werden in der Inventarliste als ausgemustert dokumentiert, aus Verwaltungssystemen und Lizenzen abgemeldet und anschließend umweltgerecht entsorgt oder dem Recycling zugeführt.

> **Aus der Praxis**
>
> Lassen Sie sich die Vernichtung von Datenträgern durch einen Entsorgungsdienstleister schriftlich bestätigen. Ein solches Vernichtungsprotokoll ist ein wichtiger Nachweis für den Datenschutz.

Das Wichtigste in Kürze

- POST, Piepcodes, LEDs und Logs sind die erste Anlaufstelle bei Hardwarefehlern.
- Memtest86 prüft den RAM, SMART-Werte zeigen den Zustand der Datenträger.
- Fehler systematisch eingrenzen: minimale Bestückung, eine Änderung nach der anderen.
- Sensoren und SMART-Trends proaktiv überwachen, Systeme regelmäßig warten.
- Lifecycle mit EOL/EOSL planen und Datenträger sicher löschen oder vernichten.

LB-04

Server & Dienste

1. Active Directory – Grundlagen
2. Active Directory – Administration
3. DNS – Deep Dive
4. DHCP – Deep Dive
5. Fileserver & DFS
6. Printserver & Druckverwaltung
7. Remote Desktop Services (RDS)
8. IIS & Webserver-Grundlagen
9. Datenbankserver

Active Directory – Grundlagen

Domäne, Forest, Trust, DC, Kerberos & LDAP

Nach diesem Kapitel können Sie …

- erklären, wofür ein Verzeichnisdienst wie Active Directory eingesetzt wird,
- die Grundbegriffe Domäne, Forest, Trust und Domänencontroller einordnen,
- die Aufgaben von Kerberos und LDAP unterscheiden,
- die Bedeutung von DNS, FSMO-Rollen und Gruppenrichtlinien für den Betrieb beschreiben.

Was Active Directory leistet

In einem kleinen Büro mit drei Rechnern lässt sich jedes Benutzerkonto noch von Hand an jedem Gerät anlegen. Sobald jedoch Dutzende oder Hunderte Menschen an wechselnden Arbeitsplätzen arbeiten, wird dieser Ansatz unübersichtlich und fehleranfällig. Genau hier setzt **Active Directory** (kurz AD) an, der zentrale Verzeichnisdienst von Microsoft.

Ein Verzeichnisdienst ist im Kern eine gut organisierte Datenbank, in der Benutzer, Computer und Gruppen mit ihren Eigenschaften gespeichert sind. Wer sich an einem Firmenrechner anmeldet, wird nicht mehr vom einzelnen Gerät geprüft, sondern zentral vom Verzeichnis. Dadurch gilt ein Konto im ganzen Netz: Dasselbe Passwort funktioniert am Arbeitsplatz, im Besprechungsraum und am Notebook. Für die IT bedeutet das vor allem eines – alles wird an einer Stelle gepflegt statt an jedem Rechner einzeln. Wird ein Konto gesperrt, ist es sofort überall gesperrt.

Die Grundbegriffe

Die wichtigste Verwaltungseinheit im Active Directory ist die **Domäne**. Sie fasst alle Objekte zusammen, die gemeinsam verwaltet werden, und trägt einen DNS-Namen wie `firma.local` oder `ad.firma.de`. Viele kleine und mittlere Unternehmen kommen mit genau einer Domäne aus.

Mehrere Domänen können zu einem **Forest** (Gesamtstruktur) zusammengefasst werden. Alle Domänen eines Forests teilen sich ein gemeinsames Schema, also die Definition, welche Objektarten und Eigenschaften es überhaupt gibt. Der Forest bildet damit die äußere Sicherheitsgrenze einer AD-Umgebung.

Damit Benutzer einer Domäne auf Ressourcen einer anderen zugreifen können, braucht es eine **Vertrauensstellung**, auf Englisch *Trust*. Innerhalb eines Forests entstehen diese Trusts automatisch; zu fremden Forests – etwa nach einer Firmenübernahme – werden sie bewusst eingerichtet.

Bereitgestellt wird das Verzeichnis von **Domänencontrollern** (DCs). Das sind Server, auf denen die AD-Datenbank liegt und die Anmeldungen prüfen. Jeder DC besitzt eine vollständige Kopie der Domänendaten, sodass Clients sich an jedem beliebigen DC anmelden können.

Anmelden und Abfragen: Kerberos und LDAP

Für die Anmeldung nutzt Active Directory das Protokoll **Kerberos**. Vereinfacht funktioniert es wie ein Ticketsystem: Beim Anmelden weist sich der Benutzer einmal gegenüber dem Domänencontroller aus und erhält ein Ticket. Möchte er anschließend auf einen Dateiserver oder Drucker zugreifen, legt er dort ein passendes Dienstticket vor. Das Passwort muss dadurch nicht ständig durch das Netz geschickt werden, was die Sicherheit deutlich erhöht.

Tickets haben eine begrenzte Gültigkeit und enthalten Zeitstempel. Deshalb ist für Kerberos eine korrekte **Zeitsynchronisation** entscheidend: Weichen die Uhren von Client und Server zu stark voneinander ab – standardmäßig sind fünf Minuten toleriert –, schlägt die Anmeldung fehl. Fehlerbilder, die nach „Passwort falsch“ aussehen, haben deshalb manchmal eine schlichte Ursache: eine falsch gehende Uhr.

LDAP (Lightweight Directory Access Protocol) ist dagegen das Protokoll, mit dem das Verzeichnis abgefragt und durchsucht wird. Wenn eine Anwendung etwa wissen möchte, in welchen Gruppen ein Benutzer Mitglied ist, oder ein Mailprogramm das Adressbuch lädt, geschieht das über LDAP.

> **Merke**
>
> Kerberos ist für die *Anmeldung* zuständig, LDAP für die *Abfrage* des Verzeichnisses. Kerberos funktioniert nur mit synchronen Uhren.

Objekte, OUs und Gruppenrichtlinien

Alles, was im Verzeichnis gespeichert ist, heißt **Objekt**: Benutzer, Computer und Gruppen sind die häufigsten. Damit diese Objekte nicht in einer einzigen langen Liste liegen, werden sie in **Organisationseinheiten** (OUs) einsortiert. OUs funktionieren ähnlich wie Ordner und lassen sich verschachteln.

Gegliedert wird sinnvollerweise nach Verwaltung oder Standort – etwa „Hamburg/Vertrieb“ oder „Clients/Notebooks“. Die OU-Struktur ist nämlich mehr als Ordnung: Sie ist die Grundlage für die Delegation von Rechten (ein Standort-Admin darf nur „seine“ OU verwalten) und für die Zuweisung von Gruppenrichtlinien.

Gruppenrichtlinien (Group Policy Objects, GPOs) sind zentrale Regeln für Benutzer und Computer. Sie steuern Sicherheitseinstellungen, Softwareverteilung, Netzlaufwerke oder das Aussehen des Desktops. Eine GPO wird mit einem Standort, der Domäne oder einer OU verknüpft und wirkt von oben nach unten. Widersprechen sich zwei Richtlinien, gewinnt die weiter unten verknüpfte, also näher am Objekt liegende. Mit dem Befehl `gpupdate` lässt sich die Aktualisierung am Client sofort anstoßen, statt auf das nächste automatische Intervall zu warten.

Eine Einschränkung sollten Sie kennen: Gruppenrichtlinien wirken nur, wenn ein Gerät regelmäßig Kontakt zu einem Domänencontroller hat. Notebooks im Homeoffice ohne VPN oder ohne hybride Anbindung an einen Cloud-Verzeichnisdienst bleiben dadurch oft tagelang auf altem Stand. In der Praxis schließen häufig Fernwartungssysteme (RMM) oder cloudbasierte Richtlinienwerkzeuge diese Lücke.

Ohne DNS geht nichts

Active Directory ist eng mit **DNS** verzahnt. Ein Client findet seinen Domänencontroller nicht über eine fest eingetragene Adresse, sondern fragt den DNS-Server nach speziellen Diensteinträgen (SRV-Records). Liefert DNS keine oder falsche Antworten, findet der Client keinen DC – und dann funktionieren weder Anmeldung noch Gruppenrichtlinien noch Dateifreigaben zuverlässig.

> **Aus der Praxis**
>
> Bei nahezu jedem AD-Problem lohnt es sich, zuerst DNS zu prüfen. Ein typischer Fehler: Ein Client verwendet als DNS-Server den Router oder einen öffentlichen Resolver statt des Domänencontrollers – dann kann er die Domäne nicht finden.

FSMO-Rollen, Standorte und Replikation

Grundsätzlich sind alle Domänencontroller gleichberechtigt: Änderungen können auf jedem DC vorgenommen werden. Einige Aufgaben vertragen jedoch keine gleichzeitigen Änderungen an mehreren Stellen. Diese fünf Sonderaufgaben heißen **FSMO-Rollen** (Flexible Single Master Operations) und werden jeweils nur von einem DC übernommen.

Geltungsbereich	Rolle	Aufgabe (vereinfacht)
Forest-weit	Schema-Master	Änderungen am Schema
Forest-weit	Domänennamen-Master	Hinzufügen und Entfernen von Domänen
Je Domäne	RID-Master	Vergabe eindeutiger Kennungsbereiche an DCs
Je Domäne	PDC-Emulator	Zeitquelle, Passwortänderungen, Kontosperren
Je Domäne	Infrastruktur-Master	Verweise auf Objekte anderer Domänen aktuell halten

Besonders wichtig im Alltag ist der **PDC-Emulator**: Er ist die maßgebliche Zeitquelle der Domäne und wird bei Passwortänderungen bevorzugt informiert. Fällt ein DC mit FSMO-Rollen aus, lassen sich die Rollen auf einen anderen DC verschieben – geordnet per *Transfer*, solange der alte DC noch erreichbar ist, oder erzwungen per *Seize*, wenn er endgültig verloren ist.

Damit alle DCs denselben Stand haben, gleichen sie ihre Daten per **Replikation** ab. Über **Sites** (Standorte) wird im AD die physische Netzstruktur abgebildet: Innerhalb eines Standorts wird schnell repliziert, zwischen Standorten gebündelt und zeitgesteuert, um langsame Leitungen zu schonen. Der **Global Catalog** schließlich ist eine Funktion bestimmter DCs, die eine Teilkopie aller Objekte des gesamten Forests vorhält und damit Forest-weite Suchen beantwortet.

> **Faustregel**
>
> Betreiben Sie mindestens zwei Domänencontroller je Domäne. Fällt einer aus, laufen Anmeldung und Namensauflösung über den zweiten weiter.

Das Wichtigste in Kürze

- Active Directory verwaltet Benutzer, Computer und Gruppen zentral.
- Domäne, Forest, Trust und Domänencontroller sind die Grundbegriffe.
- Anmeldung über Kerberos (Zeit beachten), Abfragen über LDAP.
- Objekte werden in OUs strukturiert, GPOs werden daran verknüpft.
- Ohne funktionierendes DNS kein Active Directory.

Active Directory – Administration

OUs, Gruppenrichtlinien und FSMO im Griff

Nach diesem Kapitel können Sie …

- eine OU-Struktur nach Verwaltungsgesichtspunkten planen,
- die Verarbeitungsreihenfolge von Gruppenrichtlinien (LSDOU) anwenden,
- Replikation und Gesundheit von Domänencontrollern mit Bordwerkzeugen prüfen,
- Sicherung, Wiederherstellung und Härtung des AD planen.

OU-Design: nach Verwaltung, nicht nach Optik

Eine gut durchdachte Struktur aus **Organisationseinheiten** ist das Fundament jeder AD-Verwaltung. Häufig wird das Organigramm des Unternehmens eins zu eins nachgebaut – das sieht übersichtlich aus, hilft aber selten. Entscheidend ist vielmehr, wie verwaltet wird: Welche Objekte bekommen dieselben Richtlinien? Wer soll welche Konten bearbeiten dürfen?

Bewährt hat sich eine Gliederung nach Objekttyp und Standort, etwa eine OU „Clients“ mit Unter-OUs für Notebooks und Desktops, getrennt von „Server“ und „Benutzer“. Für jede OU lassen sich Rechte gezielt **delegieren**: Der Helpdesk darf in der Benutzer-OU Passwörter zurücksetzen, ohne Domänen-Admin zu sein. Gleichzeitig ist die OU der Ort, an dem Gruppenrichtlinien gezielt zugewiesen werden.

Gruppenrichtlinien und ihre Reihenfolge

Mit **Gruppenrichtlinien** (GPOs) verteilen Sie Einstellungen zentral an Benutzer und Computer – von Passwortregeln über Netzlaufwerke bis zur Bildschirmsperre. Eine GPO wirkt erst, wenn sie mit einem Standort, der Domäne oder einer OU verknüpft ist.

Wirken mehrere Richtlinien auf dasselbe Objekt, werden sie in einer festen Reihenfolge verarbeitet: zuerst die *lokale* Richtlinie des Rechners, dann der *Standort* (Site), dann die *Domäne* und zuletzt die *OU*, bei verschachtelten OUs von außen nach innen. Merken lässt sich das mit der Abkürzung **LSDOU** (Local, Site, Domain, OU). Weil später verarbeitete Richtlinien frühere überschreiben, setzt sich bei einem Konflikt in der Regel die Einstellung durch, die am nächsten am Objekt verknüpft ist.

Nach einer Änderung müssen Sie nicht auf das automatische Aktualisierungsintervall warten. Der folgende Befehl erzwingt die sofortige Neuanwendung aller Richtlinien am Client:

```
gpupdate /force
```

> **Merke**
>
> GPO-Reihenfolge: **L**okal → **S**ite → **D**omäne → **OU**. Was zuletzt angewendet wird, gewinnt.

FSMO-Rollen planvoll betreuen

Fünf Aufgaben im AD dürfen jeweils nur von einem Domänencontroller ausgeführt werden – die **FSMO-Rollen**. Zwei davon gelten für den gesamten Forest (Schema-Master und Domänennamen-Master), drei je Domäne (PDC-Emulator, RID-Master und Infrastruktur-Master). Als Administrator sollten Sie jederzeit wissen, welcher DC welche Rolle hält; an der Kommandozeile zeigt das etwa `netdom query fsmo`.

Wird ein DC geplant außer Betrieb genommen, übertragen Sie seine Rollen vorher geordnet auf einen anderen DC (*Transfer*). Ist ein DC mit FSMO-Rollen dagegen unwiederbringlich ausgefallen, werden die Rollen auf einem verbleibenden DC übernommen (*Seize*). Wichtig: Ein DC, dessen Rollen per Seize übernommen wurden, darf danach nicht wieder ans Netz.

Replikation und Diagnose

Mehrere Domänencontroller gleichen Änderungen laufend untereinander ab. Über **Sites** steuern Sie die Replikationswege: Innerhalb eines Standorts geht es schnell, zwischen Standorten wird gebündelt repliziert. Auch hier spielt die Zeitsynchronisation eine kritische Rolle – laufen die Uhren auseinander, scheitern Kerberos-Anmeldungen und damit auch die Replikation.

Für die Fehlersuche bringt Windows zwei zentrale Werkzeuge mit. `dcdiag` prüft die Gesundheit eines Domänencontrollers: DNS-Registrierung, Dienste, Erreichbarkeit und vieles mehr. `repadmin` zeigt, ob und wann die Replikation zwischen den DCs zuletzt erfolgreich war.

```
dcdiag /v
repadmin /replsummary
repadmin /showrepl
```

Eine kurze, regelmäßige Kontrolle dieser Ausgaben – etwa wöchentlich – deckt Probleme auf, bevor Benutzer sie bemerken.

Automatisieren mit PowerShell

Vieles, was in der grafischen Oberfläche mühsam Klick für Klick erledigt wird, lässt sich mit **PowerShell** in Sekunden umsetzen. Das AD-Modul stellt Befehle bereit, um Konten in großer Zahl anzulegen oder zu ändern – etwa aus einer CSV-Datei der Personalabteilung. Ebenso lassen sich Berichte über Gruppenmitgliedschaften und Rechte erzeugen oder verwaiste Konten aufspüren, die seit Monaten nicht mehr benutzt wurden:

```
Search-ADAccount -AccountInactive -TimeSpan 90.00:00:00 -UsersOnly
```

Wiederkehrende Aufgaben, die als Skript vorliegen, laufen jedes Mal gleich ab und sind nachvollziehbar dokumentiert. Das reduziert Fehler und spart Zeit.

Sichern und Wiederherstellen

Das Active Directory wird über das **System-State-Backup** eines Domänencontrollers gesichert. Bei der Wiederherstellung unterscheidet man zwei Varianten. Bei der *nicht-autoritativen* Wiederherstellung wird ein DC aus der Sicherung zurückgeholt und holt sich anschließend alle neueren Änderungen von den übrigen DCs – der Normalfall nach einem Hardwaredefekt. Bei der *autoritativen* Wiederherstellung werden gezielt Objekte als maßgeblich markiert, etwa eine versehentlich gelöschte OU, damit sie per Replikation auf alle DCs zurückkehren.

Für einzelne gelöschte Objekte ist der **AD-Papierkorb** (Recycle Bin) deutlich einfacher: Ist er aktiviert, lassen sich gelöschte Benutzer oder Gruppen samt Attributen und Mitgliedschaften wiederherstellen, ohne ein Backup anzufassen. Der Papierkorb muss allerdings einmalig eingeschaltet werden, bevor etwas gelöscht wird.

> **Aus der Praxis**
>
> Ein AD-Backup, dessen Wiederherstellung nie geübt wurde, ist im Ernstfall wenig wert. Testen Sie die Wiederherstellung regelmäßig in einer isolierten Umgebung und betreiben Sie immer mindestens zwei Domänencontroller.

Active Directory härten

Wer die Domäne kontrolliert, kontrolliert das gesamte Unternehmensnetz. Entsprechend beliebt ist das AD bei Angreifern. Ein wirksames Konzept ist das **Tiering**: Administrative Konten werden in Ebenen getrennt, sodass sich ein Domänen-Admin nie an einem normalen Arbeitsplatz anmeldet. So kann ein kompromittierter Client nicht zur Übernahme der ganzen Domäne führen, weil dort keine hochprivilegierten Anmeldedaten zu finden sind.

Dazu gehört, Domänen-Admin-Konten nicht dauerhaft für die tägliche Arbeit zu nutzen, sondern getrennte Konten für Alltag und Administration zu führen. Mit **LAPS** (Local Administrator Password Solution) erhält jeder Rechner ein eigenes, automatisch wechselndes Passwort für das lokale Administratorkonto – ein einmal erbeutetes Passwort öffnet so nicht alle anderen Geräte. Privilegierte Konten verdienen zusätzlichen Schutz, etwa durch Mehr-Faktor-Authentifizierung, starke Passwörter und eine Überwachung ihrer Anmeldungen.

Das Wichtigste in Kürze

- OUs nach Verwaltung strukturieren – sie tragen Delegation und GPOs.
- GPO-Reihenfolge LSDOU: zuletzt angewendet gewinnt.
- FSMO-Rollen kennen, Ausfälle per Transfer oder Seize planen.
- `dcdiag` und `repadmin` regelmäßig nutzen, AD-Backup testen.
- Admin-Ebenen trennen und lokale Admin-Passwörter mit LAPS absichern.

DNS – Deep Dive

Namensauflösung im Detail

Nach diesem Kapitel können Sie …

- die wichtigsten DNS-Record-Typen und ihren Zweck benennen,
- Zonen, autoritative Server und AD-integrierte Zonen erklären,
- den Ablauf einer Namensauflösung mit Forwardern und Caching beschreiben,
- DNS-Probleme mit `nslookup` und `dig` eingrenzen und DNS absichern.

Das Telefonbuch des Netzes

Menschen merken sich Namen, Computer arbeiten mit Zahlen. Das **Domain Name System** (DNS) übersetzt deshalb Namen wie `fileserver.firma.local` oder `www.beispiel.de` in IP-Adressen. Ohne diese Übersetzung läuft im Unternehmen fast nichts: Webseiten, E-Mail, Dateifreigaben und vor allem das Active Directory sind auf DNS angewiesen. Domänencontroller werden ausschließlich über DNS gefunden.

Das Tückische: DNS-Fehler äußern sich selten mit einer klaren Meldung. Stattdessen „geht nichts“ – an vielen Stellen gleichzeitig und scheinbar ohne Zusammenhang. Wer DNS versteht, findet solche Fehler schneller.

Die wichtigsten Record-Typen

In einer DNS-Zone stehen Einträge, sogenannte *Resource Records*. Jeder Typ erfüllt einen eigenen Zweck:

Typ	Bedeutung	Beispiel
A / AAAA	Name → IPv4- bzw. IPv6-Adresse	Webserver, Fileserver
CNAME	Alias, verweist auf einen anderen Namen	`intranet` → `web01`
MX	Zuständiger Mailserver einer Domäne	Eingehende E-Mail
PTR	Reverse-Lookup: Adresse → Name	Prüfung durch Mailserver
SRV	Ort eines Dienstes inkl. Port	Domänencontroller im AD
TXT	Freitext, meist für Prüfverfahren	SPF, DKIM, DMARC

Besonders für Administratoren wichtig sind die **SRV-Records**: Über sie findet ein Windows-Client die Domänencontroller seiner Domäne. Die **TXT-Records** spielen beim E-Mail-Schutz eine große Rolle, weil dort die Regeln für SPF, DKIM und DMARC hinterlegt werden, mit denen empfangende Server die Echtheit von Absendern prüfen.

Zonen und autoritative Server

Eine **Zone** ist ein verwalteter Ausschnitt des Namensraums, etwa `firma.local`. Der Server, der diese Zone verwaltet, ist für sie **autoritativ** – er kennt die „Wahrheit" und beantwortet Anfragen verbindlich. Man unterscheidet *Forward-Zonen*, die Namen in Adressen auflösen, und *Reverse-Zonen*, die den umgekehrten Weg gehen.

Für Ausfallsicherheit gibt es klassisch einen *primären* Server, auf dem Änderungen vorgenommen werden, und einen oder mehrere *sekundäre* Server, die per Zonentransfer eine Kopie erhalten. Im Active-Directory-Umfeld nutzt man dagegen meist **AD-integrierte Zonen**: Die DNS-Daten liegen dann im Verzeichnis selbst und werden automatisch mit der AD-Replikation auf alle Domänencontroller verteilt. Jeder DC mit DNS-Rolle kann Änderungen annehmen.

Wie eine Anfrage aufgelöst wird

Ein Client stellt seine Frage in der Regel **rekursiv**: Er erwartet von seinem DNS-Server eine fertige Antwort, egal wie viel Arbeit dahintersteckt. Kennt der Server die Antwort nicht, fragt er sich selbst **iterativ** durch die Hierarchie – von den Root-Servern über die Top-Level-Domain bis zum zuständigen autoritativen Server – und erhält bei jedem Schritt nur einen Verweis auf die nächste Stelle.

In Unternehmen übernehmen das oft **Forwarder**: Der interne DNS-Server leitet alle Anfragen, die er nicht selbst beantworten kann, an einen festgelegten Server weiter, etwa den des Providers. **Conditional Forwarder** leiten nur Anfragen für bestimmte Domänen weiter, zum Beispiel an die DNS-Server eines Partnerunternehmens. Die **Root Hints**, eine Liste der Root-Server, dienen als Rückfallebene, wenn keine Weiterleitung greift.

Damit nicht jede Anfrage neu aufgelöst werden muss, speichern Server und Clients Antworten zwischen. Wie lange, bestimmt die **TTL** (Time To Live) des jeweiligen Eintrags. Eine kurze TTL sorgt dafür, dass Änderungen schnell überall ankommen, erzeugt aber mehr Anfragen. Bei hartnäckigen Fehlern nach einer Änderung hilft es oft, den Cache zu leeren – auf dem Client mit `ipconfig /flushdns`, auf dem Server über die DNS-Verwaltung.

> **Merke**
>
> Senken Sie die TTL eines Eintrags einige Zeit *vor* einer geplanten Änderung, etwa einem Serverumzug. Dann verbreitet sich die neue Adresse schnell, statt dass Clients stundenlang die alte aus dem Cache verwenden.

Fehlersuche mit nslookup und dig

Die wichtigsten Werkzeuge sind `nslookup` (unter Windows und Linux verfügbar) und `dig` (vor allem unter Linux). Testen Sie dabei möglichst gegen einen *konkreten* Server, statt sich auf die Standardeinstellung zu verlassen – so sehen Sie, ob ein bestimmter Server falsch antwortet:

```
nslookup fileserver.firma.local 192.168.10.5
nslookup -type=SRV _ldap._tcp.dc._msdcs.firma.local
dig @192.168.10.5 www.beispiel.de A
```

Prüfen Sie auch Reverse-Lookups: Fehlende oder falsche PTR-Einträge führen zu Problemen, etwa wenn Mailserver eingehende Verbindungen ablehnen. Ein Sonderfall ist **Split-Brain-DNS**: Derselbe Name liefert intern eine andere Antwort als von außen, etwa die interne Adresse des Webservers im Firmennetz und die öffentliche aus dem Internet. Das ist oft gewollt, kann aber verwirren, wenn ein Test „von außen“ und „von innen“ unterschiedliche Ergebnisse bringt.

Wartung und Sicherheit

Mit der Zeit sammeln sich in dynamischen Zonen veraltete Einträge von Geräten, die längst nicht mehr existieren. Das **Scavenging** entfernt solche Einträge automatisch nach einer festgelegten Frist und hält die Zone sauber.

Auf der Sicherheitsseite sollten **Zonentransfers** nur an bekannte sekundäre Server erlaubt sein – sonst kann jeder die komplette Liste Ihrer internen Systeme abrufen. Interne Namen haben in öffentlichem DNS nichts zu suchen, weil sie Angreifern einen Überblick über die Infrastruktur verschaffen. **DNSSEC** signiert DNS-Antworten kryptografisch, sodass ein Client erkennen kann, ob eine Antwort unterwegs manipuliert wurde. Schließlich lohnt ein Blick in die DNS-Protokolle: Ungewöhnlich viele Anfragen an unbekannte Domänen können auf Schadsoftware hinweisen.

> **Aus der Praxis**
>
> Melden mehrere Benutzer gleichzeitig ganz unterschiedliche Störungen, prüfen Sie zuerst die Namensauflösung. Ein schneller `nslookup` gegen den eigenen DNS-Server klärt in Sekunden, ob DNS die Ursache ist.

Das Wichtigste in Kürze

- DNS ist das Fundament – bei Störungen zuerst hier suchen.
- Record-Typen (A, AAAA, CNAME, MX, PTR, SRV, TXT) und Zonen verstehen.
- Forwarder, Root Hints und TTL bestimmen Auflösung und Caching.
- `nslookup` und `dig` gegen einen konkreten Server einsetzen.
- Zonentransfers einschränken, Scavenging nutzen, DNSSEC kennen.

DHCP – Deep Dive

Automatische IP-Vergabe zuverlässig betreiben

Nach diesem Kapitel können Sie …

- den DORA-Ablauf einer DHCP-Adressvergabe erklären,
- Scopes, Ausschlüsse, Reservierungen und Lease-Dauer sinnvoll planen,
- die wichtigsten DHCP-Optionen zuordnen,
- DHCP ausfallsicher und sicher betreiben und typische Fehler erkennen.

Wozu DHCP?

Jedes Gerät im Netz braucht eine IP-Adresse, eine Subnetzmaske, ein Standard-Gateway und mindestens einen DNS-Server. Diese Werte bei jedem Rechner, Telefon und Drucker von Hand einzutragen, wäre aufwendig und fehleranfällig – doppelt vergebene Adressen und Tippfehler wären an der Tagesordnung. Das **Dynamic Host Configuration Protocol** (DHCP) übernimmt diese Aufgabe automatisch: Ein Gerät wird eingeschaltet, fragt nach einer Konfiguration und ist wenige Sekunden später einsatzbereit.

Der DORA-Ablauf

Die Adressvergabe folgt immer denselben vier Schritten, deren Anfangsbuchstaben das Kürzel **DORA** ergeben. Zuerst sendet der Client ein *Discover* als Broadcast ins Netz, weil er noch keine Adresse hat und keinen Server kennt. Jeder erreichbare DHCP-Server antwortet mit einem *Offer*, also einem Angebot für eine Adresse samt Parametern. Der Client wählt ein Angebot aus und bestätigt es mit einem *Request*. Abschließend bestätigt der Server die Vergabe mit einem *Acknowledge* (ACK) – ab jetzt darf der Client die Adresse nutzen.

Die Adresse gehört dem Client nicht dauerhaft, sondern ist für eine begrenzte Zeit „geliehen“ – die **Lease**. Etwa zur Hälfte der Laufzeit versucht der Client, die Lease beim Server zu verlängern. Gelingt das, behält er seine Adresse; bleibt er länger weg, wird sie nach Ablauf wieder frei.

Scopes, Ausschlüsse und Reservierungen

Ein **Scope** (Bereich) ist der Adressbereich, den ein DHCP-Server für ein bestimmtes Subnetz verwaltet, etwa `192.168.10.100` bis `192.168.10.200`. Adressen, die Sie statisch vergeben haben – etwa für Server, Router oder Switches –, nehmen Sie über **Ausschlüsse** aus der Vergabe heraus, damit es nicht zu Konflikten kommt.

Eine **Reservierung** sorgt dafür, dass ein bestimmtes Gerät anhand seiner MAC-Adresse immer dieselbe IP-Adresse bekommt. Das ist praktisch für Drucker, Zeiterfassungsterminals oder andere Geräte, die unter einer festen Adresse erreichbar sein sollen, aber trotzdem zentral verwaltet werden. Reservierungen sollten sauber dokumentiert sein, damit später nachvollziehbar ist, welches Gerät welche Adresse hat.

Die **Lease-Dauer** wählen Sie passend zur Umgebung. In einem Büro mit festen Arbeitsplätzen sind mehrere Tage üblich. Im Gäste-WLAN, in dem ständig neue Geräte auftauchen und wieder verschwinden, ist eine kurze Lease von wenigen Stunden sinnvoll, damit der Bereich nicht vollläuft.

DHCP-Optionen

Neben der Adresse liefert DHCP über nummerierte **Optionen** weitere Einstellungen mit. Einige davon begegnen Ihnen in nahezu jeder Umgebung:

Option	Inhalt
3	Standard-Gateway (Router)
6	DNS-Server
15	DNS-Suffix (Domänenname)
Weitere	z. B. Angaben für VoIP-Telefone oder den PXE-Netzwerkstart

Optionen lassen sich auf Serverebene für alle Scopes oder gezielt pro Scope festlegen; die Scope-Einstellung hat dabei Vorrang. Mit **Vendor- und User-Classes** können Sie noch feiner unterscheiden: Telefone eines bestimmten Herstellers erhalten dann zum Beispiel die Adresse ihres Provisionierungsservers, während normale PCs diese Option gar nicht sehen. Über **PXE-Boot-Optionen** starten Rechner direkt aus dem Netz, was für die automatische Installation von Betriebssystemen genutzt wird.

Hochverfügbarkeit und Relay

Fällt der einzige DHCP-Server aus, merken das zunächst nur wenige – bestehende Leases laufen ja weiter. Doch neue Geräte und Geräte mit abgelaufener Lease bekommen keine Adresse mehr. Planen Sie deshalb Redundanz ein. Windows Server bietet dafür **DHCP-Failover** mit zwei Betriebsarten: Im *Lastausgleich* teilen sich zwei Server die Anfragen, im *Hot-Standby* springt der zweite Server nur ein, wenn der erste ausfällt. Beide Server kennen den Stand der Leases des jeweils anderen.

Weil das Discover ein Broadcast ist, endet es am Router. Für Subnetze ohne eigenen DHCP-Server wird deshalb auf dem Router oder Layer-3-Switch ein **DHCP-Relay** eingerichtet, häufig als *IP-Helper* bezeichnet. Er nimmt die Broadcasts entgegen und leitet sie

gezielt an den zentralen DHCP-Server weiter.

> **Merke**
>
> Ohne DHCP bekommen neue Geräte keine Adresse. Failover sorgt für Ausfallsicherheit, das Relay (IP-Helper) bringt DHCP über Subnetzgrenzen.

Sicherheit

Ein gefährliches Szenario ist ein **Rogue-DHCP-Server**: ein nicht autorisierter Server, der falsche Adressen verteilt. Das kann ein versehentlich angeschlossener Heimrouter sein – oder ein Angreifer, der sich selbst als Gateway oder DNS-Server einträgt und so den Verkehr mitlesen kann. Dagegen hilft **DHCP-Snooping** auf den Switches: Nur an festgelegten, vertrauenswürdigen Ports dürfen DHCP-Antworten ins Netz gelangen. Im Active Directory müssen Windows-DHCP-Server zudem **autorisiert** werden, bevor sie Adressen vergeben. Ungewöhnliche Vergaben – etwa plötzlich sehr viele neue Leases – sollten Sie im Blick behalten.

Fehlersuche, Überwachung und Wartung

Ein Klassiker bei der Fehlersuche ist eine Adresse aus dem Bereich `169.254.x.x`. Sie stammt nicht vom DHCP-Server, sondern hat sich der Client selbst gegeben, weil keine Antwort kam – das nennt man **APIPA** (Automatic Private IP Addressing). Mit einer solchen Adresse ist keine normale Kommunikation möglich. Ursachen sind ein nicht erreichbarer oder ausgefallener Server, ein fehlendes Relay im Subnetz oder ein erschöpfter Scope.

```
ipconfig /release
ipconfig /renew
ipconfig /all
```

Ein voller Scope bedeutet, dass neue Geräte keine Adresse mehr bekommen. Überwachen Sie deshalb die **Auslastung** der Bereiche und reagieren Sie rechtzeitig, indem Sie den Bereich vergrößern oder die Lease-Dauer verkürzen. Achten Sie außerdem auf Adresskonflikte, die entstehen, wenn ein Gerät eine Adresse statisch nutzt, die auch per DHCP vergeben wird. Und vergessen Sie nicht, die DHCP-Datenbank samt Konfiguration regelmäßig zu sichern.

> **Aus der Praxis**
>
> Zeigt ein Client eine Adresse mit `169.254`, liegt das Problem fast nie am Client selbst. Prüfen Sie Server, Relay und freie Adressen im Scope.

Das Wichtigste in Kürze

- DORA: Discover, Offer, Request, Acknowledge.
- Scopes, Ausschlüsse und Reservierungen sauber planen und dokumentieren.
- Optionen 3, 6 und 15 liefern Gateway, DNS-Server und DNS-Suffix.
- Failover und Relay für Ausfallsicherheit, Snooping gegen Rogue-DHCP.
- Eine APIPA-Adresse (169.254.x.x) bedeutet: DHCP-Problem.

Fileserver & DFS

Dateidienste, Berechtigungen und verteilte Namespaces

Nach diesem Kapitel können Sie …

- den Nutzen eines zentralen Fileservers begründen,
- das Zusammenspiel von Share- und NTFS-Rechten erklären,
- Rechte nach dem Prinzip der minimalen Rechte über Gruppen vergeben,
- Quotas, Dateiscreening, DFS-Namespaces und DFS-Replikation einordnen.

Warum ein zentraler Fileserver?

Liegen wichtige Dateien verstreut auf einzelnen Arbeitsplatzrechnern, entstehen „Dateninseln“: Niemand weiß genau, wo die aktuelle Version liegt, Kolleginnen und Kollegen können nicht darauf zugreifen, und bei einem Festplattendefekt ist die Arbeit von Monaten verloren. Ein **Fileserver** schafft eine zentrale Ablage mit geregeltem Zugriff. Er lässt sich an einer Stelle sichern, Berechtigungen werden zentral gepflegt, und Teams können gemeinsam an Dokumenten arbeiten.

Bereitgestellt werden die Ordner als **Freigaben** (Shares) über das Protokoll **SMB** (Server Message Block). Die Benutzer sehen eine Freigabe als Netzlaufwerk oder rufen sie über einen Pfad wie `\\fileserver\projekte` auf.

Zwei Rechte-Ebenen: Share und NTFS

Ein prüfungsrelevanter Klassiker ist das Zusammenspiel zweier Berechtigungsebenen. Die **Share-Rechte** hängen an der Freigabe selbst und gelten nur, wenn über das Netz zugegriffen wird. Die **NTFS-Rechte** hängen an Dateien und Ordnern im Dateisystem und gelten immer – auch wenn sich jemand direkt am Server anmeldet.

Greift ein Benutzer über das Netz zu, werden beide Ebenen geprüft, und es gilt die **restriktivere** der beiden. Hat jemand auf der Freigabe „Ändern“, im NTFS aber nur „Lesen“, darf er nur lesen. Umgekehrt gilt dasselbe.

Share-Recht	NTFS-Recht	Effektiv über das Netz
Vollzugriff	Lesen	Lesen
Lesen	Ändern	Lesen
Ändern	Ändern	Ändern

Damit Sie nicht an zwei Stellen gleichzeitig Rechte pflegen müssen, hat sich eine einfache Regel bewährt: Setzen Sie die Share-Rechte großzügig, etwa „Ändern“ für authentifizierte Benutzer, und steuern Sie den eigentlichen Zugriff präzise über NTFS.

> **Merke**
>
> Über das Netz gewinnt immer die restriktivere Berechtigung. Best Practice: Share großzügig, NTFS präzise.

Rechte sauber vergeben

Vergeben Sie Rechte grundsätzlich über **Gruppen**, nie direkt an einzelne Benutzer. Wechselt jemand die Abteilung, ändern Sie nur die Gruppenmitgliedschaft statt Dutzender Ordnerberechtigungen. Nutzen Sie die **Vererbung** bewusst: Rechte, die auf einem übergeordneten Ordner gesetzt sind, gelten automatisch für die Unterordner. Das ist praktisch, sollte aber nur an gut überlegten Stellen unterbrochen werden, weil die Struktur sonst schnell unübersichtlich wird.

Leitlinie ist das **Prinzip der minimalen Rechte** (Least Privilege): Jede Person erhält genau die Rechte, die sie für ihre Arbeit braucht. Prüfen Sie regelmäßig, ob noch Berechtigungen für ausgeschiedene Mitarbeitende oder aufgelöste Gruppen existieren, und räumen Sie solche verwaisten Rechte auf. Mit **Access-Based Enumeration** lässt sich zudem erreichen, dass Benutzer in einer Freigabe nur die Ordner sehen, auf die sie auch zugreifen dürfen.

Speicherplatz steuern mit FSRM

Ohne Grenzen füllen sich Fileserver erstaunlich schnell – oft mit privaten Fotos, Videos oder doppelt abgelegten Dateien. Der **File Server Resource Manager** (FSRM) unter Windows Server bietet dafür drei Werkzeuge. Mit **Quotas** legen Sie Kontingente pro Ordner oder Benutzer fest; bei Erreichen einer Schwelle gibt es eine Warnung oder es wird gesperrt. Das **Dateiscreening** verhindert das Ablegen bestimmter Dateitypen, zum Beispiel ausführbarer Programme. Und **Speicherberichte** zeigen, welche Ordner wachsen, welche Dateien doppelt vorhanden sind und was seit Jahren nicht mehr geöffnet wurde.

DFS: ein Pfad für viele Server

Wächst ein Unternehmen, verteilen sich Freigaben über mehrere Server und Standorte. Für die Benutzer wird es dann unübersichtlich, und jeder Serverumzug bedeutet, Laufwerkszuordnungen anzupassen. Das **Distributed File System** (DFS) löst das mit zwei Bausteinen.

Der **DFS-Namespace** fasst Freigaben verschiedener Server unter einem einheitlichen Pfad zusammen, etwa `\\firma.local\daten\vertrieb`. Wo die Daten tatsächlich liegen, sieht der Benutzer nicht. Ziehen die Daten auf einen neuen Server um, wird nur das Ziel im Namespace geändert – der Pfad bleibt gleich.

Die **DFS-Replikation** (DFS-R) hält Ordner zwischen mehreren Servern synchron, auch standortübergreifend. Benutzer greifen dann auf die Kopie am eigenen Standort zu, und fällt ein Server aus, steht die Kopie am anderen Standort bereit. DFS-R hat allerdings Grenzen: Dateien, die gerade geöffnet sind, werden nicht sofort repliziert, und wenn an zwei Standorten gleichzeitig dieselbe Datei bearbeitet wird, gewinnt eine Version. Für die Replikation braucht DFS-R außerdem einen ausreichend großen **Staging-Speicher** als Zwischenablage.

Praxis: Schattenkopien, Auditing und Migration

Aktivieren Sie auf Datenlaufwerken die **Schattenkopien** über den Volume Shadow Copy Service (VSS). Benutzer können dann über „Vorgängerversionen“ selbst eine ältere Fassung einer Datei wiederherstellen, ohne den Helpdesk zu bemühen. Das **Auditing** protokolliert Zugriffe und Änderungen und hilft bei der Frage, wer eine Datei gelöscht oder verschoben hat. Regelmäßige Berechtigungsberichte zeigen, wer worauf zugreifen darf.

Für den Umzug von Daten auf einen neuen Server ist **Robocopy** das Werkzeug der Wahl, weil es Dateien samt NTFS-Rechten, Zeitstempeln und Besitzern überträgt und abgebrochene Kopien fortsetzen kann:

```
robocopy D:\Daten \\neuserver\D$\Daten /MIR /COPYALL /R:1 /W:1 /LOG:C:\robocopy.log
```

Vorsicht mit `/MIR`: Der Schalter spiegelt das Ziel exakt auf die Quelle und löscht dort Dateien, die in der Quelle fehlen.

Aus der Praxis

RAID und Schattenkopien ersetzen kein Backup. Sie schützen vor Plattenausfall oder versehentlichem Überschreiben, nicht aber vor Ransomware, Brand oder dem Ausfall des ganzen Servers. Geben Sie außerdem eine klare Ordnerstruktur vor – sie erleichtert Rechtevergabe und Sicherung gleichermaßen.

Das Wichtigste in Kürze

- Über das Netz gilt die restriktivere von Share- und NTFS-Recht.
- Rechte über Gruppen und nach dem Prinzip der minimalen Rechte vergeben.
- FSRM steuert Speicherplatz mit Quotas, Screening und Berichten.

- DFS-Namespace vereinheitlicht Pfade, DFS-R repliziert zwischen Servern.
- Schattenkopien helfen im Alltag, ersetzen aber kein Backup.

Printserver & Druckverwaltung

Drucken zentral bereitstellen

Nach diesem Kapitel können Sie …

- die Vorteile eines zentralen Printservers erklären,
- Treiberstrategien und die Verteilung von Druckern per GPO beschreiben,
- Warteschlangendruck und Direktdruck gegeneinander abwägen,
- typische Druckstörungen beheben und Druckdienste absichern.

Warum ein Printserver?

Drucken wirkt unspektakulär, verursacht im IT-Alltag aber erstaunlich viele Tickets. Wird jeder Drucker an jedem Rechner einzeln eingerichtet, muss bei einem Treiberwechsel oder einem neuen Gerät jeder Arbeitsplatz angefasst werden. Ein **Printserver** zentralisiert diese Arbeit: Er stellt die Drucker als Freigaben bereit, verwaltet die Treiber und führt die Druckwarteschlangen.

Für die IT bedeutet das: Es gibt einen Ort für Treiber und Konfiguration. Warteschlangen lassen sich zentral einsehen, anhalten oder leeren, und Einstellungen wie Standardpapierformat oder Duplexdruck werden einmal festgelegt und gelten für alle. Der Aufwand pro Client sinkt deutlich, weil die Arbeitsplätze die Drucker nur noch verbinden müssen.

Treiber: weniger ist mehr

Die meisten Druckprobleme haben ihre Ursache in Treibern. Jedes Druckermodell bringt eigene Treiber mit, oft in mehreren Versionen und für verschiedene Betriebssysteme. Je mehr unterschiedliche Treiber im Umlauf sind, desto häufiger kommt es zu Konflikten und Abstürzen des Druckdienstes.

Eine bewährte Strategie ist deshalb, die Vielfalt zu reduzieren. **Universaltreiber** der Hersteller unterstützen viele Modelle mit einem einzigen Treiber, und die Treiber werden zentral auf dem Printserver bereitgestellt statt auf jedem Client einzeln installiert. Microsoft hat zudem mit den sogenannten **Typ-4-Treibern** ein Treibermodell eingeführt, das die Verteilung vereinfacht: Der Client muss dabei keinen herstellerspezifischen Treiber mehr vom Server laden, sondern nutzt einen allgemeinen Klassentreiber. In der Praxis ist es sinnvoll, Treiberkonflikte gar nicht erst entstehen zu lassen, indem Sie möglichst wenige, getestete Treiber einsetzen.

Drucker automatisch verteilen

Damit die Benutzer ihre Drucker nicht selbst suchen müssen, werden sie in der Regel per **Gruppenrichtlinie** zugewiesen. Über die OU-Struktur oder Sicherheitsgruppen erhalten etwa alle Mitarbeitenden eines Standorts oder einer Abteilung automatisch die passenden Drucker. Auch der **Standarddrucker** lässt sich so vorgeben – zum Beispiel immer das Gerät auf dem eigenen Stockwerk.

Ergänzend erlauben viele Umgebungen eine **Selbstbedienung**: Über ein Portal oder die Druckersuche im Verzeichnis können Benutzer zusätzliche Drucker selbst verbinden, etwa wenn sie ausnahmsweise in einem anderen Gebäude arbeiten.

Wie gedruckt wird: Warteschlange oder Direktdruck

Technisch gibt es zwei Wege. Beim **Warteschlangendruck** schickt der Client den Auftrag an den Printserver, der ihn zwischenspeichert und an das Gerät weitergibt. Das bietet Kontrolle: Aufträge lassen sich zentral überwachen, protokollieren und bei Bedarf umleiten. Der Nachteil ist die Abhängigkeit vom Server – fällt er aus, druckt niemand.

Beim **Direktdruck** spricht der Client den Netzwerkdrucker unmittelbar an, meist über einen TCP-Port (häufig Port 9100) oder über **IPP**, das Internet Printing Protocol. IPP ist ein modernes, netzwerkfähiges Protokoll, das auch Statusinformationen vom Drucker zurückliefert und verschlüsselt über HTTPS betrieben werden kann. Direktdruck reduziert die Abhängigkeit vom Server, macht aber die zentrale Steuerung schwieriger.

Verfahren	Vorteil	Nachteil
Warteschlange über Printserver	Zentrale Kontrolle, Protokollierung, einheitliche Treiber	Server ist zentrale Abhängigkeit
Direktdruck (TCP/IPP)	Keine Serverabhängigkeit, einfache Struktur	Weniger Überblick, Verwaltung pro Client

Wenn nichts mehr druckt

Für die Fehlersuche hat sich eine feste Reihenfolge bewährt. Häufigste Ursache ist ein hängender **Druckspooler**, also der Windows-Dienst, der die Aufträge verwaltet. Ein Neustart des Dienstes und das Leeren der Warteschlange lösen viele Probleme sofort:

```
net stop spooler
del /q %systemroot%\System32\spool\PRINTERS\*
net start spooler
```

Hilft das nicht, prüfen Sie die Netzwerkseite: Ist der Drucker eingeschaltet und unter seiner IP-Adresse erreichbar, etwa per `ping`? Hat er nach einem Neustart vielleicht eine neue Adresse bekommen? Bleiben Probleme nur bei bestimmten Druckern oder Anwendungen bestehen, ist häufig der Treiber die Ursache. Testen Sie dann mit einem anderen oder neueren Treiber, um das Problem einzugrenzen.

> **Merke**
>
> Erste Anlaufstelle bei Druckstörungen: Spooler-Dienst neu starten und Warteschlange leeren. Danach Netzwerk, dann Treiber prüfen.

Sicherheit nach PrintNightmare

Der Druckspooler läuft mit hohen Systemrechten und war deshalb wiederholt Ziel von Angriffen. Besonders bekannt wurde die Schwachstellenreihe **PrintNightmare** aus dem Jahr 2021, bei der Angreifer über die Treiberinstallation Schadcode mit Systemrechten ausführen konnten. Seitdem gelten einige Grundregeln: Härten Sie **Point-and-Print** so, dass normale Benutzer keine Treiber mehr selbst installieren dürfen. Aktivieren Sie die **Treiberisolierung**, damit ein fehlerhafter Treiber nicht den gesamten Spooler mit sich reißt, und lassen Sie nur **signierte Treiber** zu. Spielen Sie Sicherheitsupdates für Printserver zeitnah ein, und deaktivieren Sie den Spooler-Dienst auf Servern, die gar nicht drucken müssen – insbesondere auf Domänencontrollern.

Trends und moderne Alternativen

Klassische Printserver im eigenen Haus werden seltener. **Cloud-Druckdienste**, etwa Microsofts Universal Print, verlagern die Warteschlangen in die Cloud, sodass auch Geräte ohne Verbindung zum Firmennetz drucken können. Gleichzeitig reduziert der Direktdruck die Abhängigkeit von einem zentralen Server.

Wichtiger wird das **sichere Drucken**: Der Auftrag wird erst ausgegeben, wenn sich die Person am Gerät mit PIN oder Mitarbeiterausweis anmeldet. So bleiben vertrauliche Ausdrucke nicht im Ausgabefach liegen, und nicht abgeholte Aufträge verursachen keine Papierkosten. Über ein **Print-Management** lassen sich Druckvolumen und Kosten zudem pro Abteilung auswerten und steuern.

> **Aus der Praxis**
>
> Bevor ein neuer Printserver aufgebaut wird, lohnt die Frage, ob Direktdruck oder ein Cloud-Druckdienst die Anforderungen nicht einfacher erfüllt – vor allem bei vielen mobilen Mitarbeitenden.

Das Wichtigste in Kürze

- Ein Printserver zentralisiert Treiber, Freigaben und Warteschlangen.
- Universaltreiber und wenige getestete Treiber reduzieren Konflikte.
- Gruppenrichtlinien verteilen Drucker automatisch nach Standort oder Abteilung.
- Spooler und Warteschlange sind die erste Fehlerquelle.
- Point-and-Print härten, nur signierte Treiber, Server zeitnah patchen.

Remote Desktop Services (RDS)

Zentrale Desktops und Anwendungen bereitstellen

Nach diesem Kapitel können Sie …

- das Grundprinzip von Remote Desktop Services erklären,
- die RDS-Rollen Session Host, Gateway, Web Access und Connection Broker zuordnen,
- RemoteApp und vollständige Remote-Desktops unterscheiden,
- Lizenzierung, sicheren Zugang und Skalierung einer RDS-Umgebung planen.

Die Grundidee

Bei den **Remote Desktop Services** (RDS) arbeiten die Benutzer nicht mehr auf der Rechenleistung ihres eigenen Geräts, sondern auf zentralen Servern im Rechenzentrum. Programme laufen auf dem Server; zum Arbeitsplatz werden nur Bildschirminhalte übertragen, und in die Gegenrichtung gehen Tastatur- und Mauseingaben. Als Protokoll dient dabei RDP, das Remote Desktop Protocol.

Dieses Prinzip hat mehrere Vorteile. Am Arbeitsplatz genügt ein einfacher **Thin Client** oder ein beliebiges Gerät, auch ein privates Notebook oder ein Tablet. Software wird nur einmal auf dem Server installiert und aktualisiert statt auf jedem Rechner. Und weil die Daten das Rechenzentrum nicht verlassen, sind sie besser geschützt – geht ein Gerät verloren, liegt darauf nichts Wertvolles.

Die Rollen im Überblick

Eine RDS-Umgebung besteht aus mehreren Rollen, die auf einem oder mehreren Servern installiert werden können:

Rolle	Aufgabe
RD Session Host	Führt die Benutzersitzungen und Anwendungen aus
RD Connection Broker	Verteilt Sitzungen auf die Session Hosts und verbindet getrennte Sitzungen wieder
RD Web Access	Bietet Zugriff auf Desktops und Anwendungen über den Browser
RD Gateway	Ermöglicht sicheren Zugang von außen über HTTPS
RD-Lizenzserver	Verwaltet die Zugriffslizenzen (CALs)

Der **Session Host** ist das Arbeitspferd: Auf ihm laufen die Sitzungen vieler Benutzer gleichzeitig. Der **Connection Broker** sorgt dafür, dass sich die Last gleichmäßig auf mehrere Session Hosts verteilt, und bringt einen Benutzer nach einem Verbindungsabbruch wieder in seine bestehende Sitzung zurück. Über **Web Access** finden die Benutzer ihre freigegebenen Programme im Browser, und das **Gateway** ist der kontrollierte Eingang aus dem Internet.

RemoteApp: nur die Anwendung

Nicht immer braucht es einen kompletten entfernten Desktop. Mit **RemoteApp** wird nur eine einzelne Anwendung bereitgestellt, die sich für den Benutzer wie ein lokal installiertes Programm anfühlt: Sie erscheint im Startmenü, hat ein eigenes Fenster in der Taskleiste und lässt sich neben lokalen Programmen nutzen. Tatsächlich läuft sie aber auf dem Session Host.

Das ist ideal für einzelne Fachanwendungen, etwa eine Branchensoftware, die eine schnelle Verbindung zur Datenbank im Rechenzentrum braucht, oder ein Programm, das nur für wenige Benutzer lizenziert ist. Die Benutzer arbeiten ansonsten ganz normal an ihrem eigenen Rechner.

Lizenzierung mit CALs

Für die Nutzung von RDS sind neben den Serverlizenzen spezielle **Client Access Licenses** (RDS-CALs) erforderlich. Es gibt sie in zwei Varianten. Die **User-CAL** wird pro benanntem Benutzer vergeben, unabhängig davon, von wie vielen Geräten diese Person zugreift. Die **Device-CAL** gilt pro Gerät, egal wie viele Personen es nutzen.

Welches Modell passt, hängt von der Nutzung ab: Arbeiten Mitarbeitende mit Notebook, Tablet und Homeoffice-Rechner, sind User-CALs meist günstiger. Teilen sich dagegen mehrere Schichten wenige Geräte, etwa in der Produktion, lohnen sich Device-CALs. Verwaltet werden die Lizenzen über einen **RD-Lizenzserver**. Ohne ihn läuft eine RDS-Umgebung nur während einer begrenzten Übergangsfrist; danach sind keine Verbindungen mehr möglich.

> **Merke**
>
> User-CAL pro Person, Device-CAL pro Gerät. Ein RD-Lizenzserver ist Pflicht – ohne ihn endet der Betrieb nach der Übergangsfrist.

Zugang absichern

RDP direkt aus dem Internet erreichbar zu machen, gehört zu den häufigsten und gefährlichsten Fehlern. Offene RDP-Ports werden automatisiert gesucht und mit Passwortangriffen überzogen; viele Ransomware-Vorfälle beginnen genau so. Der Zugang von au-

ßen gehört deshalb über das **RD Gateway**, das die Verbindung in HTTPS verpackt, und zwar in Kombination mit **Mehr-Faktor-Authentifizierung**. Alternativ wird RDS nur über ein VPN erreichbar gemacht.

Zusätzlich sollte die **Authentifizierung auf Netzwerkebene** (Network Level Authentication, NLA) aktiv sein. Dabei muss sich ein Benutzer anmelden, bevor überhaupt eine vollständige Sitzung aufgebaut wird – das schont Ressourcen und erschwert Angriffe. Auf dem Session Host gilt das Prinzip der minimalen Rechte: Benutzer erhalten keine Administratorrechte, damit sie nicht die Sitzungen anderer beeinträchtigen können. Und weil viele Personen auf demselben Server arbeiten, sind zeitnahe Sicherheitsupdates besonders wichtig.

Aus der Praxis

Prüfen Sie regelmäßig von außen, ob Port 3389 Ihrer öffentlichen Adressen erreichbar ist. Ist er offen, sollte das sofort geändert werden – Zugang nur über Gateway mit MFA oder über VPN.

Skalierung, Profile und Betrieb

Mehrere Session Hosts werden zu einer **Farm** zusammengefasst, auf die der Connection Broker die Benutzer verteilt. Damit jemand auf jedem Host seine gewohnten Einstellungen vorfindet, müssen die **Benutzerprofile** zentral liegen. Verbreitet sind dafür Profilcontainer wie FSLogix, die das Profil als virtuelle Festplatte auf einem Fileserver ablegen und bei der Anmeldung einbinden, sowie die Ordnerumleitung für Dokumente und Desktop.

Planen Sie die **Ressourcen pro Benutzer** sorgfältig: Arbeitsspeicher und Prozessorleistung teilen sich alle Sitzungen eines Hosts. Eine einzige ressourcenhungrige Anwendung kann die Arbeit vieler bremsen. Überwachen Sie deshalb die Leistung laufend und erweitern Sie die Farm rechtzeitig.

Alternativen aus der Cloud

Neben der eigenen RDS-Installation gibt es heute Cloud-Angebote, die dasselbe Prinzip ohne eigene Serverhardware umsetzen, etwa Azure Virtual Desktop oder Windows 365. Sie sparen eigene Infrastruktur und lassen sich flexibel an die Benutzerzahl anpassen, verursachen aber laufende Kosten und erfordern eine gute Internetanbindung. Ob die eigene RDS-Farm oder ein Cloud-Desktop sinnvoller ist, hängt von Anwendungen, Datenschutzanforderungen und Budget ab.

Das Wichtigste in Kürze

- RDS zentralisiert Desktops und Anwendungen auf Servern im Rechenzentrum.
- Rollen: Session Host, Connection Broker, Web Access, Gateway, Lizenzserver.
- RemoteApp stellt einzelne Programme bereit, als wären sie lokal installiert.
- User- oder Device-CALs passend zur Nutzung wählen.
- Nie RDP offen ins Internet – Gateway mit MFA, NLA und zeitnahe Patches.

IIS & Webserver-Grundlagen

Webinhalte und -anwendungen bereitstellen

Nach diesem Kapitel können Sie …

- IIS, nginx und Apache einordnen,
- Sites, Bindings und Host-Header erklären,
- den Zweck von Application Pools beschreiben,
- einen Webserver mit HTTPS, Reverse Proxy und Basis-Hardening absichern und Fehler anhand der Logs eingrenzen.

Webserver im Überblick

Ein Webserver nimmt Anfragen von Browsern oder anderen Programmen entgegen und liefert Inhalte zurück – statische Dateien wie Bilder und HTML-Seiten oder die Ergebnisse von Webanwendungen. Die Kommunikation läuft über HTTP beziehungsweise dessen verschlüsselte Variante HTTPS. Auch viele interne Anwendungen, vom Intranet bis zur Zeiterfassung, werden heute über einen Webserver bereitgestellt.

Drei Webserver begegnen Ihnen besonders häufig. **IIS** (Internet Information Services) ist der Webserver von Microsoft und tief in Windows Server integriert; er ist die naheliegende Wahl für ASP.NET-Anwendungen und Windows-Umgebungen. **Apache** und **nginx** sind vor allem unter Linux verbreitet. nginx wird zudem sehr häufig als *Reverse Proxy* eingesetzt, also als vorgeschalteter Server, der Anfragen an andere Server weiterleitet. Welcher Webserver passt, richtet sich nach Plattform und Anwendung.

Sites und Bindings

Ein Webserver kann viele Websites gleichzeitig betreiben. Damit er weiß, welche Anfrage zu welcher Site gehört, gibt es **Bindings** (Bindungen). Ein Binding verknüpft eine Site mit einer IP-Adresse, einem Port und einem Hostnamen – bei HTTPS zusätzlich mit einem Zertifikat. Die Standardports sind 80 für HTTP und 443 für HTTPS.

Teilen sich mehrere Sites dieselbe IP-Adresse und denselben Port, unterscheidet der Server sie über den **Host-Header**: Der Browser schickt bei jeder Anfrage mit, welchen Namen er aufgerufen hat, etwa `intranet.firma.de` oder `shop.firma.de`. So können auf einem einzigen Server und unter einer einzigen Adresse beliebig viele Websites laufen. Für HTTPS übernimmt diese Unterscheidung die TLS-Erweiterung SNI, damit der Server das passende Zertifikat ausliefern kann.

Application Pools: Anwendungen isolieren

Eine besondere Stärke von IIS sind die **Application Pools**. Jeder Pool ist ein eigener Arbeitsprozess, in dem eine oder mehrere Webanwendungen laufen. Stürzt eine Anwendung ab oder hängt sie, betrifft das nur ihren Pool – die Anwendungen in anderen Pools laufen ungestört weiter.

Jeder Pool hat zudem eine eigene **Identität**, also ein eigenes Konto, mit dem er auf Dateien und Datenbanken zugreift. Dadurch lassen sich Rechte pro Anwendung genau begrenzen: Wird eine Anwendung kompromittiert, hat der Angreifer nur die Rechte dieses einen Pools. Über das **Recycling** startet IIS einen Pool in festgelegten Abständen oder bei hohem Speicherverbrauch kontrolliert neu. Das wirkt Speicherlecks entgegen, die sonst den Server über Tage hinweg verlangsamen würden. Außerdem können Sie die Ressourcen je Pool begrenzen, damit eine einzelne Anwendung nicht den ganzen Server auslastet.

> **Merke**
>
> Bindings verknüpfen Site, IP, Port und Hostname. Application Pools isolieren Anwendungen voneinander – jede wichtige Anwendung bekommt ihren eigenen Pool.

HTTPS und Zertifikate

Webseiten ohne Verschlüsselung sind heute nicht mehr zeitgemäß, auch im internen Netz nicht. Binden Sie für jede Site ein **TLS-Zertifikat**, dessen Name zur Adresse passt, und richten Sie eine automatische **Weiterleitung von HTTP auf HTTPS** ein, damit niemand versehentlich unverschlüsselt zugreift. Zertifikate haben eine begrenzte Laufzeit; ein abgelaufenes Zertifikat führt zu Warnmeldungen im Browser oder macht Anwendungen unbenutzbar. Überwachen Sie deshalb die Ablaufdaten oder automatisieren Sie die Erneuerung. Veraltete und unsichere Protokollversionen wie SSL 3.0, TLS 1.0 und TLS 1.1 sollten deaktiviert sein.

Reverse Proxy und URL-Rewrite

Mit **URL-Rewrite** schreibt der Webserver Adressen um oder leitet sie weiter, bevor die Anwendung die Anfrage sieht. So werden alte Adressen nach einem Relaunch auf neue umgeleitet oder lange technische Pfade in lesbare Adressen übersetzt.

Ein **Reverse Proxy** nimmt Anfragen aus dem Netz entgegen und leitet sie an einen oder mehrere dahinterliegende Server (Backends) weiter. Bei IIS übernimmt diese Aufgabe die Erweiterung ARR (Application Request Routing), unter Linux häufig nginx. Der Reverse Proxy kann die Last auf mehrere Backends verteilen, die TLS-Verschlüsselung zentral übernehmen und die eigentlichen Anwendungsserver vor direktem Zugriff aus dem Internet abschirmen.

Basis-Hardening

Jede aktivierte Funktion ist eine mögliche Angriffsfläche. Deaktivieren Sie deshalb **nicht benötigte Module**, etwa ungenutzte Authentifizierungsverfahren oder Skriptsprachen. Das **Verzeichnis-Browsing** gehört abgeschaltet, damit Besucher nicht einfach die Dateilisten eines Ordners einsehen können. Fehlermeldungen sollten für Außenstehende **nicht zu gesprächig** sein: Detaillierte Meldungen mit Pfaden, Versionsnummern oder Datenbankfehlern helfen Angreifern und gehören nur in die Logs. Schließlich setzen Sie **Security-Header**, die dem Browser zusätzliche Schutzregeln mitgeben – etwa HSTS, das den Browser anweist, die Site künftig nur noch per HTTPS aufzurufen.

Logs, Fehlersuche und Performance

Die **IIS-Logs** verzeichnen jede Anfrage mit Zeitpunkt, Adresse, aufgerufenem Pfad und Statuscode. Die Statuscodes sind der schnellste Hinweis auf die Ursache eines Problems:

Code	Bedeutung	Typische Ursache
401 / 403	Nicht angemeldet / verboten	Authentifizierung oder Dateirechte
404	Nicht gefunden	Falscher Pfad, fehlende Datei, falsches Binding
500	Interner Serverfehler	Fehler in der Anwendung oder Konfiguration
503	Dienst nicht verfügbar	Application Pool gestoppt oder überlastet

Für schwierige Fälle bietet IIS das **Failed Request Tracing**, das den Weg einer fehlschlagenden Anfrage durch alle Module detailliert aufzeichnet. Prüfen Sie außerdem immer den Zustand des Application Pools – ein gestoppter Pool ist eine häufige Ursache für Fehler 503.

Für gute **Performance** aktivieren Sie Output-Caching und Kompression, damit häufig abgerufene Inhalte nicht jedes Mal neu erzeugt und übertragen werden. Statische Inhalte wie Bilder und Skripte sollten mit langen Cache-Zeiten ausgeliefert werden. Reicht ein Server nicht mehr aus, wird eine **Web-Farm** aus mehreren gleichartigen Servern hinter einem Load Balancer aufgebaut.

Aus der Praxis

Meldet eine Website plötzlich Fehler 503, prüfen Sie zuerst im IIS-Manager, ob der zugehörige Application Pool läuft. Häufig wurde er nach wiederholten Abstürzen automatisch angehalten – das Ereignisprotokoll verrät dann den eigentlichen Grund.

Das Wichtigste in Kürze

- IIS für Windows-Umgebungen, nginx und Apache vor allem unter Linux.
- Bindings verknüpfen Site, IP, Port, Hostname und Zertifikat.
- Application Pools isolieren Anwendungen und begrenzen deren Rechte.
- HTTPS mit Weiterleitung und Basis-Hardening sind Pflicht.
- Logs und Statuscodes sind der Einstieg in jede Fehlersuche.

Datenbankserver

Grundlagen für IT-Admins

Nach diesem Kapitel können Sie …

- die Rolle eines Datenbankservers im Unternehmen beschreiben,
- die gängigen Systeme SQL Server, MySQL/MariaDB und PostgreSQL einordnen,
- grundlegende Administrationsaufgaben wie Rechte, Wartung und Backup planen,
- Ansätze für Hochverfügbarkeit, Performance und Sicherheit benennen.

Das Herzstück vieler Anwendungen

Hinter fast jeder Geschäftsanwendung steht eine Datenbank: Warenwirtschaft, Buchhaltung, CRM, Ticketsystem oder Webshop speichern ihre Daten strukturiert in Tabellen. Ein **Datenbankserver** verwaltet diese Daten, beantwortet die Abfragen der Anwendungen und sorgt dafür, dass gleichzeitige Änderungen vieler Benutzer sich nicht gegenseitig stören.

Für die IT-Administration bedeutet das eine besondere Verantwortung. Ist die Datenbank langsam, ist die Anwendung langsam; fällt sie aus, steht oft ein ganzer Geschäftsbereich still. Und weil hier häufig die wertvollsten Daten des Unternehmens liegen, sind Backups unverzichtbar. Als Administrator müssen Sie kein Datenbankentwickler sein, sollten aber die Grundlagen von Betrieb, Sicherung und Absicherung beherrschen.

Die gängigen Systeme

Microsoft SQL Server ist im Windows-Umfeld weit verbreitet und tief in Windows und Active Directory integriert – Benutzer können sich etwa mit ihrem Domänenkonto anmelden. **MySQL** und der daraus hervorgegangene Ableger **MariaDB** sind Open-Source-Systeme, die besonders häufig unter Linux und für Webanwendungen eingesetzt werden. **PostgreSQL** ist ebenfalls quelloffen und gilt als besonders funktionsreich und standardtreu.

System	Typische Plattform	Besonderheit
SQL Server	Windows (auch Linux)	Enge Integration in Windows und AD, verschiedene Editionen
MySQL / MariaDB	Linux, Webhosting	Weit verbreitet bei Webanwendungen
PostgreSQL	Linux, plattformübergreifend	Sehr funktionsreich, quelloffen

Die Editionen der Systeme unterscheiden sich teils erheblich in Preis und Funktionsumfang – bei SQL Server etwa reicht die Spanne von einer kostenlosen Einstiegsedition mit Größenbeschränkungen bis zu Editionen für große Rechenzentren. In der Praxis wählen Sie das System selten frei: Meist gibt die Anwendung vor, welche Datenbank sie unterstützt.

Admin-Grundlagen

Zunächst lohnt ein Blick auf den Aufbau. Eine **Instanz** ist ein laufender Datenbankdienst auf einem Server; auf einem Server können mehrere Instanzen nebeneinander laufen. Jede Instanz verwaltet eine oder mehrere **Datenbanken**, und innerhalb einer Datenbank gliedern **Schemata** die Tabellen und weiteren Objekte in logische Bereiche.

Benutzer und Rechte vergeben Sie nach dem Prinzip der minimalen Rechte: Eine Anwendung erhält ein eigenes Konto, das nur auf ihre Datenbank zugreifen darf – und nur in dem Umfang, den sie braucht. Administratorrechte für Anwendungskonten sind ein häufiger, aber vermeidbarer Fehler.

Behalten Sie außerdem **Dienste und Ports** im Blick. Jedes System lauscht auf einem Standardport, bei SQL Server ist das TCP 1433, bei MySQL 3306, bei PostgreSQL 5432. Diese Ports sollten nur von den Anwendungsservern aus erreichbar sein, die sie wirklich benötigen, nie aus dem Internet. Für wiederkehrende Aufgaben richten Sie **Wartungspläne** ein, die Backups, Index-Pflege und Integritätsprüfungen automatisch und regelmäßig ausführen.

Backup im Überblick

Datenbanken werden anders gesichert als gewöhnliche Dateien. Eine Datenbank ist ständig in Bewegung; eine einfache Dateikopie im laufenden Betrieb kann halb geschriebene Transaktionen enthalten und damit unbrauchbar sein. Deshalb nutzt man die eigenen Sicherungsmechanismen der Datenbanksysteme. Typisch ist eine Kombination aus **Vollsicherung**, **Differenzsicherung** und **Transaktionslog-Sicherung**. Letztere er-

möglicht die **Point-in-Time-Wiederherstellung**, also das Zurücksetzen auf einen beliebigen Zeitpunkt, etwa auf den Moment kurz vor einem versehentlichen Löschvorgang. Die Einzelheiten behandelt der Lernbereich Storage & Backup.

> **Merke**
>
> Ein Datenbank-Backup zählt erst, wenn der Restore funktioniert. Testen Sie die Wiederherstellung regelmäßig in einer Testumgebung.

Hochverfügbarkeit

Für kritische Datenbanken reicht ein einzelner Server oft nicht aus. Bei der **Replikation** werden Änderungen laufend auf einen oder mehrere weitere Server übertragen. Diese Kopien können lesende Abfragen übernehmen, etwa für Auswertungen, und bei einem Ausfall einspringen. Ein **Failover-Cluster** verbindet mehrere Server so, dass ein anderer Knoten den Dienst automatisch übernimmt, wenn einer ausfällt. Bei SQL Server sind dafür die **Always-On-Verfügbarkeitsgruppen** verbreitet, die Datenbanken auf mehrere Server spiegeln.

Auch bei hochverfügbaren Systemen brauchen Updates und Wartungsarbeiten feste **Wartungsfenster**, die mit den Fachabteilungen abgestimmt sind. Ein laufendes **Monitoring** von Last, Speicherplatz und Sperren (Locks) zeigt, wenn sich Abfragen gegenseitig blockieren oder der Server an seine Grenzen kommt.

Performance und Wartung

Der größte Hebel für schnelle Abfragen sind **Indizes** und aktuelle **Statistiken**. Ein Index funktioniert wie das Stichwortverzeichnis eines Buches: Statt die ganze Tabelle zu durchsuchen, springt die Datenbank direkt an die richtige Stelle. Durch viele Änderungen werden Indizes mit der Zeit fragmentiert; regelmäßige Index-Pflege und Reorganisation halten sie effizient. Die Statistiken helfen der Datenbank, den schnellsten Weg für eine Abfrage zu wählen.

Auf der Hardwareseite profitieren Datenbankserver vor allem von ausreichend **Arbeitsspeicher**, weil häufig genutzte Daten dann im Cache liegen und nicht von der Platte gelesen werden müssen, sowie von schnellen **Datenträgern** wie NVMe-SSDs. Bleibt ein System trotzdem langsam, gilt es, die **langsamen Abfragen** zu identifizieren – alle gängigen Systeme bieten dafür Protokolle oder Auswertungen, die zeigen, welche Abfragen am meisten Zeit verbrauchen.

Datenbank-Sicherheit

Neben sparsamen Rechten für Datenbankkonten gehören einige weitere Maßnahmen zum Grundschutz. **Verbindungen** zwischen Anwendung und Datenbank sollten per TLS verschlüsselt sein, damit niemand Daten oder Zugangsdaten im Netz mitlesen kann. Die **Verschlüsselung im Ruhezustand**, bei SQL Server etwa über Transparent Data Encryption (TDE), schützt die Datenbankdateien und Backups, falls Datenträger gestohlen werden. Schließlich sollten Zugriffe **auditiert** werden: Wer hat wann welche Daten gelesen oder geändert? Gerade bei personenbezogenen Daten ist das auch für den Datenschutz wichtig.

Aus der Praxis

Klagen Benutzer über eine plötzlich langsame Anwendung, lohnt ein Blick auf den Datenbankserver: Sind Arbeitsspeicher oder Datenträger ausgelastet, blockieren Sperren andere Abfragen oder ist die Index-Pflege seit Wochen nicht gelaufen?

Das Wichtigste in Kürze

- SQL Server, MySQL/MariaDB und PostgreSQL sind die verbreitetsten Systeme.
- Rechte sparsam vergeben, Ports abschotten, Wartung automatisch planen.
- Datenbanken mit eigenen Mechanismen sichern und den Restore testen.
- Replikation und Failover-Cluster sichern kritische Datenbanken ab.
- Indizes, Statistiken und genug RAM sind der größte Hebel für Performance.

LB-05

Virtualisierung & Container

1. Virtualisierung – Grundlagen
2. VMware & ESXi
3. Hyper-V
4. Proxmox VE
5. Docker & Container-Grundlagen
6. Kubernetes-Einstieg

Virtualisierung – Grundlagen

Mehrere Systeme auf einer physischen Maschine

Nach diesem Kapitel können Sie …

- erklären, was ein Hypervisor tut und woraus eine virtuelle Maschine besteht,
- Typ-1- und Typ-2-Hypervisoren unterscheiden und Beispiele zuordnen,
- Ressourcen wie vCPU, RAM und Speicher sinnvoll zuweisen,
- Vorteile, typische Einsatzfelder und Risiken der Virtualisierung einschätzen.

Was Virtualisierung bedeutet

Ein moderner Server ist leistungsfähig – so leistungsfähig, dass ein einzelner Dienst wie ein Druckserver oder ein kleiner Webserver ihn kaum auslastet. Früher bekam trotzdem oft jede Anwendung ihren eigenen physischen Server. Die Folge waren volle Serverräume, hohe Stromkosten und Maschinen, die die meiste Zeit fast nichts taten.

Virtualisierung löst dieses Problem. Eine spezielle Software, der **Hypervisor**, teilt die physische Hardware in mehrere **virtuelle Maschinen** (VMs) auf. Jede VM verhält sich wie ein eigenständiger Computer mit eigenem Betriebssystem, eigenen Programmen und eigener Netzwerkadresse. Die VMs sind voneinander isoliert: Stürzt eine ab oder wird sie kompromittiert, laufen die anderen unbeeinträchtigt weiter. Der physische Rechner, auf dem der Hypervisor läuft, wird als *Host* bezeichnet, die VMs als *Gäste*.

Typ-1 und Typ-2: zwei Arten von Hypervisoren

Ein **Typ-1-Hypervisor**, auch *Bare-Metal-Hypervisor* genannt, läuft direkt auf der Hardware, ohne ein darunterliegendes Betriebssystem. Er ist schlank, effizient und stabil und deshalb die Wahl für Server. Bekannte Vertreter sind VMware ESXi, Microsoft Hyper-V und Proxmox VE, die in den folgenden Kapiteln vorgestellt werden.

Ein **Typ-2-Hypervisor**, auch *Hosted Hypervisor*, wird dagegen wie ein normales Programm auf einem bestehenden Betriebssystem installiert. Beispiele sind VirtualBox oder VMware Workstation. Weil zwischen Hardware und VM ein komplettes Betriebssystem liegt, sind Typ-2-Lösungen weniger effizient. Dafür sind sie schnell eingerichtet und ideal, um auf dem eigenen Arbeitsplatzrechner etwas auszuprobieren oder eine Testumgebung zu betreiben.

Merkmal	Typ 1 (Bare-Metal)	Typ 2 (Hosted)
Läuft auf	Direkt auf der Hardware	Einem vorhandenen Betriebssystem
Beispiele	ESXi, Hyper-V, Proxmox VE	VirtualBox, VMware Workstation
Einsatz	Server, Produktivbetrieb	Desktop, Tests, Schulung
Effizienz	Hoch	Geringer

Woraus eine VM besteht

Eine virtuelle Maschine bekommt vom Hypervisor **virtuelle Hardware** zugewiesen: eine oder mehrere virtuelle CPUs, Arbeitsspeicher, eine virtuelle Festplatte und eine virtuelle Netzwerkkarte. Das Gastbetriebssystem sieht diese Komponenten wie echte Hardware. Die virtuelle Festplatte ist dabei nichts anderes als eine Datei – oder ein Satz von Dateien – auf dem Speicher des Hosts. Letztlich ist eine ganze VM also eine Sammlung von Dateien, und genau das macht sie so flexibel.

Mit einem **Snapshot** lässt sich der Zustand einer VM zu einem bestimmten Zeitpunkt festhalten. Geht anschließend etwas schief, etwa bei einem Update, kehren Sie mit wenigen Klicks zum gespeicherten Stand zurück. Damit die VM optimal mit dem Hypervisor zusammenarbeitet, werden im Gast zusätzlich **Gasterweiterungen** installiert. Sie liefern optimierte Treiber, synchronisieren die Uhrzeit und ermöglichen ein sauberes Herunterfahren vom Host aus.

Ressourcen zuweisen

Alle VMs teilen sich die physischen Ressourcen des Hosts. Jeder VM werden virtuelle CPUs (**vCPUs**) und eine bestimmte Menge Arbeitsspeicher zugewiesen. Dabei ist **Überbuchung** (Overcommitment) möglich: Die Summe der zugewiesenen vCPUs oder des RAM darf größer sein als das, was physisch vorhanden ist, weil selten alle VMs gleichzeitig unter Volllast stehen. Das erhöht die Auslastung, muss aber mit Bedacht geschehen. Wird zu stark überbucht, konkurrieren die VMs um Ressourcen und alle werden langsam.

Bei virtuellen Festplatten gibt es zwei Varianten. Beim **Thin Provisioning** wächst die Datei erst mit den tatsächlich geschriebenen Daten – das spart Platz, birgt aber das Risiko, dass der physische Speicher unbemerkt vollläuft. Beim **Thick Provisioning** wird der gesamte Platz von Anfang an reserviert. Für den Arbeitsspeicher kennen viele Hypervisoren das **Ballooning**: Ein Treiber im Gast gibt ungenutzten Speicher an den Host zurück, damit er anderen VMs zur Verfügung steht.

Vorteile und typische Einsatzfelder

Der größte Gewinn ist die **Konsolidierung**: Viele Dienste laufen auf wenigen leistungsstarken Hosts, was Hardware, Platz und Strom spart. Hinzu kommt **Flexibilität** – ein neuer Server ist nicht mehr eine Frage von Wochen, sondern von Minuten. Aus einer vorbereiteten Vorlage, einem *Template*, lassen sich neue VMs schnell und einheitlich ausrollen. Mit der **Live-Migration** lässt sich eine laufende VM ohne Unterbrechung auf einen anderen Host verschieben, etwa um diesen zu warten.

Auch das Sichern wird einfacher. Weil eine VM aus Dateien besteht, können Backup-Programme sie *agentenlos* direkt über den Hypervisor sichern, ohne in jeder VM eine eigene Software zu installieren. Ganze Systeme lassen sich so schnell wiederherstellen.

Typische Einsatzfelder sind neben der Serverkonsolidierung **Test- und Entwicklungsumgebungen**, die schnell aufgebaut und per Snapshot zurückgesetzt werden können. Ältere **Legacy-Systeme**, für die es keine passende Hardware mehr gibt, lassen sich gekapselt in einer VM weiterbetreiben. Und bei der **Desktop-Virtualisierung** (VDI) laufen ganze Arbeitsplätze zentral im Rechenzentrum.

Grenzen und Risiken

Virtualisierung bündelt viele Systeme auf wenig Hardware – und damit auch das Risiko. Fällt ein Host aus, sind alle seine VMs gleichzeitig betroffen. Deshalb ist **Host-Redundanz** Pflicht: mindestens zwei Hosts, gemeinsamer Speicher und Funktionen, die VMs bei einem Ausfall automatisch auf einem anderen Host neu starten. Auch die **Lizenzierung** kann komplex werden, weil viele Softwarehersteller nach physischen Kernen, Hosts oder VMs abrechnen und die Regeln je nach Produkt unterschiedlich sind.

> **Merke**
>
> Ein Snapshot ist kein Backup. Er liegt auf demselben Speicher wie die VM selbst – fällt dieser aus, sind VM und Snapshot gleichermaßen verloren. Zudem verlangsamen lange bestehende Snapshots die VM und wachsen stetig an.

Aus der Praxis

Nutzen Sie Snapshots gezielt und kurzfristig, etwa unmittelbar vor einem Update. Ist die Änderung erfolgreich getestet, löschen Sie den Snapshot wieder. Für die dauerhafte Sicherung sorgt ein echtes Backup auf einem getrennten Speicher.

Das Wichtigste in Kürze

- Ein Hypervisor teilt physische Hardware in isolierte virtuelle Maschinen auf.

- Typ 1 läuft direkt auf der Hardware und ist für Server gedacht, Typ 2 für Desktop und Tests.
- Eine VM besteht aus Dateien – das erleichtert Klonen, Vorlagen und Backup.
- Überbuchung und Thin Provisioning sparen Ressourcen, brauchen aber Überwachung.
- Host-Redundanz einplanen; Snapshots ersetzen kein Backup.

VMware & ESXi

Der etablierte Enterprise-Hypervisor

Nach diesem Kapitel können Sie …

- ESXi, vSphere und vCenter voneinander abgrenzen,
- die Kernfunktionen vMotion, Storage vMotion, HA und DRS erklären,
- die Rolle von Datastores, Shared Storage und virtuellen Switches beschreiben,
- Vorlagen und Klone für das schnelle Ausrollen von VMs nutzen,
- die veränderte Lizenzsituation einordnen.

ESXi, vSphere und vCenter

VMware gilt seit vielen Jahren als Marktführer für Server-Virtualisierung im Unternehmensumfeld. Um die Produktwelt zu verstehen, sollten Sie drei Begriffe auseinanderhalten. **ESXi** ist der eigentliche Typ-1-Hypervisor, der direkt auf dem Server installiert wird. Er ist bewusst schlank gehalten und bringt nur eine einfache Weboberfläche für die Verwaltung des einzelnen Hosts mit.

vSphere ist der Name der gesamten Produktsuite rund um ESXi. Und **vCenter** ist die zentrale Verwaltung, die viele ESXi-Hosts zusammenfasst. Ohne vCenter müsste jeder Host einzeln verwaltet werden – das funktioniert bei einem oder zwei Servern, skaliert aber nicht. Mit vCenter laufen alle Fäden an einem Ort zusammen: die Konsole für sämtliche Hosts und VMs, die Rollen- und Rechteverwaltung, Vorlagen für das schnelle Ausrollen neuer VMs und Automatisierungsfunktionen. Viele der wichtigsten Funktionen von VMware stehen überhaupt erst mit vCenter zur Verfügung.

Die Kernfunktionen: vMotion, HA und DRS

Mehrere ESXi-Hosts werden in vCenter zu einem **Cluster** zusammengefasst. Innerhalb dieses Verbunds entfalten die zentralen Funktionen ihre Wirkung. **vMotion** verschiebt eine laufende VM ohne Unterbrechung von einem Host auf einen anderen. Anwenderinnen und Anwender bemerken davon nichts; Verbindungen bleiben bestehen. So lassen sich Hosts warten, ohne dass VMs ausfallen.

HA (High Availability) überwacht die Hosts im Cluster. Fällt einer aus, startet HA die betroffenen VMs automatisch auf den verbleibenden Hosts neu. Das ist kein unterbrechungsfreier Betrieb – die VMs starten neu –, verkürzt die Ausfallzeit aber auf wenige

Minuten, ohne dass jemand eingreifen muss. **DRS** (Distributed Resource Scheduler) verteilt die Last automatisch: Ist ein Host stark ausgelastet, verschiebt DRS per vMotion VMs auf weniger belastete Hosts.

Eine Ergänzung ist **Storage vMotion**. Während das normale vMotion nur die Ausführung der VM, also CPU und Arbeitsspeicher, auf einen anderen Host verlegt, verschiebt Storage vMotion die VM-Dateien im laufenden Betrieb auf einen anderen Datastore. So lässt sich ein Speichersystem warten oder ablösen, ohne VMs abzuschalten. Auf Wunsch kann dabei auch das Festplattenformat geändert werden, etwa auf Thin Provisioning.

Funktion	Was sie tut	Nutzen
vMotion	Verschiebt laufende VMs zwischen Hosts	Wartung ohne Ausfall
Storage vMotion	Verschiebt VM-Dateien zwischen Datastores	Speicherumzug im Betrieb
HA	Startet VMs nach Host-Ausfall neu	Kurze Ausfallzeit, automatisch
DRS	Verteilt Last über die Hosts	Gleichmäßige Auslastung

Storage und Netzwerk

Die Dateien der VMs liegen in sogenannten **Datastores**. Das sind Speicherbereiche, die entweder mit dem VMware-eigenen Clusterdateisystem **VMFS** formatiert sind – etwa auf einem SAN über iSCSI oder Fibre Channel – oder per **NFS** von einem Netzwerkspeicher eingebunden werden.

Damit vMotion und HA funktionieren, brauchen die Hosts **Shared Storage**, also gemeinsamen Speicher, auf den alle Hosts gleichzeitig zugreifen können. Nur dann kann ein anderer Host eine VM übernehmen, ohne dass ihre Dateien erst kopiert werden müssen. Für die Ausfallsicherheit wird der Speicher über mehrere unabhängige Wege angebunden; dieses **Multipathing** sorgt dafür, dass ein defektes Kabel oder ein ausgefallener Switch die Verbindung nicht unterbricht.

Im Netzwerk verbinden **virtuelle Switches** die VMs untereinander und mit dem physischen Netz. Ein Standard-vSwitch wird auf jedem Host einzeln konfiguriert. Ein **Distributed Switch** wird dagegen zentral im vCenter angelegt und gilt für alle Hosts gleichzeitig – das vermeidet Konfigurationsunterschiede, die sonst bei einer Migration zu Problemen führen können.

> **Merke**
>
> ESXi ist der Hypervisor, vCenter die zentrale Verwaltung. vMotion, HA und DRS sind die Kernvorteile – und sie alle setzen gemeinsamen Speicher voraus.

Vorlagen und Klone

In größeren Umgebungen werden neue VMs selten von Hand installiert. Stattdessen dient ein **Template** als „goldenes Abbild“: eine fertig eingerichtete, aktualisierte VM, aus der beliebig viele neue VMs erzeugt werden. Ein **Full Clone** ist eine vollständige, eigenständige Kopie. Ein **Linked Clone** teilt sich dagegen die Basisdatei mit dem Original und speichert nur die Unterschiede – das spart viel Speicherplatz, macht den Klon aber von der Basis abhängig.

Damit nicht jede neue VM denselben Namen und dieselbe IP-Adresse trägt, passt eine **Customization Specification** beim Ausrollen automatisch Rechnername, Netzwerkeinstellungen und Domänenbeitritt an.

Praxis und Wartung

In jeder VM sollten die **VMware Tools** installiert sein. Sie liefern optimierte Treiber, synchronisieren die Uhrzeit und ermöglichen ein sauberes Herunterfahren der VM vom Host aus. Steht eine Wartung an einem Host an, versetzen Sie ihn in den **Maintenance Mode**: DRS verschiebt dann alle laufenden VMs auf andere Hosts, und der Host kann gefahrlos aktualisiert oder neu gestartet werden. Patches und Updates werden zentral über den **vSphere Lifecycle Manager** verteilt. Behalten Sie außerdem Warnungen und Gesundheitsmeldungen im vCenter im Blick – volle Datastores oder alte Snapshots fallen dort frühzeitig auf.

Die veränderte Lizenzsituation

Nach der Übernahme von VMware durch Broadcom, die Ende 2023 abgeschlossen wurde, haben sich die Lizenzmodelle grundlegend geändert. Unbefristete Lizenzen wurden durch **Abonnements** ersetzt, und Einzelprodukte wurden zu größeren **Bundles** zusammengefasst. Für viele Unternehmen, insbesondere kleinere, sind die Kosten dadurch gestiegen.

Als Folge prüfen viele Organisationen Alternativen wie Proxmox VE oder Hyper-V. Ob sich ein Wechsel lohnt, hängt vom Einzelfall ab: Funktionsumfang, vorhandenes Know-how, Backup-Lösungen und der Aufwand der Migration müssen den möglichen Einsparungen gegenübergestellt werden. Die Lizenzbedingungen ändern sich zudem weiterhin – prüfen Sie den aktuellen Stand deshalb immer direkt beim Hersteller oder Ihrem Lizenzpartner.

Aus der Praxis

Steht eine Vertragsverlängerung an, planen Sie die Bewertung frühzeitig. Eine Migration auf eine andere Plattform braucht in der Regel mehrere Monate Vorlauf für Tests, Schulung und die schrittweise Umstellung.

Das Wichtigste in Kürze

- ESXi ist der Typ-1-Hypervisor, vSphere die Suite, vCenter die zentrale Verwaltung.
- vMotion, HA und DRS ermöglichen Wartung ohne Ausfall und automatische Lastverteilung.
- Shared Storage ist Voraussetzung für Live-Migration und HA.
- Templates, Klone und Customization Specifications beschleunigen das Ausrollen.
- Die Lizenzlage seit der Broadcom-Übernahme regelmäßig neu bewerten.

Hyper-V

Microsofts Virtualisierung in Windows Server

Nach diesem Kapitel können Sie …

- Hyper-V als Typ-1-Hypervisor einordnen und bereitstellen,
- die drei Arten virtueller Switches und die VM-Generationen unterscheiden,
- Live Migration, Replica und Failover-Cluster für Verfügbarkeit nutzen,
- Checkpoints und Dynamic Memory richtig einsetzen,
- die passenden Verwaltungswerkzeuge auswählen.

Hyper-V im Überblick

Hyper-V ist die Virtualisierungslösung von Microsoft. Sie ist in Windows Server integriert und wird dort als *Server-Rolle* aktiviert. Obwohl die Installation über ein Windows-Betriebssystem erfolgt, ist Hyper-V ein Typ-1-Hypervisor: Nach der Aktivierung läuft der Hypervisor direkt auf der Hardware, und das ursprüngliche Windows wird selbst zu einer besonderen Partition, über die Hyper-V verwaltet wird.

Auch die Pro- und Enterprise-Editionen von Windows für Arbeitsplätze enthalten Hyper-V. Dort eignet es sich gut, um Testumgebungen auf dem eigenen Rechner zu betreiben. Ein großer Vorteil ist die enge Integration in die Microsoft-Welt: Active Directory, PowerShell, Failover-Clustering und Backup-Schnittstellen greifen nahtlos ineinander. Wer ohnehin eine Windows-Infrastruktur betreibt, findet sich schnell zurecht.

Virtuelle Switches

Damit VMs kommunizieren können, werden sie an **virtuelle Switches** angeschlossen. Hyper-V kennt drei Typen. Ein **externer Switch** ist mit einer physischen Netzwerkkarte des Hosts verbunden; die VMs erreichen darüber das reale Netzwerk, als wären sie eigene Rechner im LAN. Ein **interner Switch** verbindet nur den Host mit den VMs, nicht aber mit dem physischen Netz. Ein **privater Switch** erlaubt ausschließlich die Kommunikation der VMs untereinander – nicht einmal der Host ist beteiligt.

Den Switch-Typ wählen Sie je nach Zweck: Produktive Server hängen am externen Switch, isolierte Testumgebungen eher an einem privaten. Zusätzlich lassen sich VLAN-Kennungen direkt an der virtuellen Netzwerkkarte einer VM setzen, sodass VMs verschiedener Netzsegmente denselben physischen Anschluss nutzen können.

Switch-Typ	Verbindung	Typischer Einsatz
Extern	VMs ↔ physisches Netz (und Host)	Produktive Server
Intern	Host ↔ VMs	Verwaltung, Datenaustausch mit dem Host
Privat	Nur VM ↔ VM	Isolierte Labor- und Testnetze

Generation 1 und Generation 2

Beim Anlegen einer VM legen Sie fest, welcher **Generation** sie angehört – und diese Entscheidung lässt sich später nicht mehr ändern. Eine VM der **Generation 2** nutzt UEFI statt des klassischen BIOS, unterstützt Secure Boot und kann über Standard-Netzwerkkarten per PXE aus dem Netz starten. Sie ist die richtige Wahl für alle modernen 64-Bit-Betriebssysteme.

Die **Generation 1** bildet ältere Hardware nach und ist für Gastsysteme gedacht, die mit UEFI nicht zurechtkommen, etwa sehr alte Windows- oder Linux-Versionen sowie 32-Bit-Systeme. Als Faustregel gilt: Generation 2 verwenden, außer ein altes Gastsystem erzwingt Generation 1.

Live Migration, Replica und Hochverfügbarkeit

Wie andere Hypervisoren kann auch Hyper-V laufende VMs ohne Unterbrechung auf einen anderen Host verschieben. Diese **Live Migration** funktioniert am komfortabelsten innerhalb eines **Failover-Clusters**: Mehrere Hyper-V-Hosts greifen auf gemeinsamen Speicher zu, und fällt ein Host aus, startet der Cluster die betroffenen VMs automatisch auf einem anderen neu. Der Failover-Cluster ist damit die Grundlage für Hochverfügbarkeit. Live Migration ist zwar auch zwischen einzelnen Hosts ohne Cluster möglich, dann aber ohne den automatischen Neustart im Fehlerfall.

Für den Katastrophenfall bietet Hyper-V eine eingebaute Lösung: **Hyper-V Replica** überträgt den Zustand einer VM in regelmäßigen Abständen *asynchron* an einen zweiten Host, idealerweise an einem anderen Standort. Fällt der Hauptstandort aus, kann die Kopie dort gestartet werden. Weil die Übertragung zeitversetzt erfolgt, gehen dabei die Änderungen der letzten Minuten verloren. Replica ist deshalb eine einfache, kostengünstige Option für die Notfallwiederherstellung (Disaster Recovery), aber kein Ersatz für einen Cluster oder ein Backup.

Checkpoints und Dynamic Memory

Was bei anderen Hypervisoren Snapshot heißt, nennt Hyper-V **Checkpoint**. Es gibt zwei Varianten. Ein **Standard-Checkpoint** erfasst auch den Inhalt des Arbeitsspeichers und hält den Zustand damit exakt fest. Laufende Anwendungen wie Datenbanken bemerken

davon aber nichts, sodass sie nach einem Zurücksetzen in einem nicht sauberen Zustand sein können. Ein **Production-Checkpoint** nutzt dagegen innerhalb der VM den Volumeschattenkopie-Dienst (VSS) bzw. unter Linux einen vergleichbaren Mechanismus. Er ist *anwendungskonsistent* und wird für Produktivsysteme empfohlen.

Checkpoints sind praktisch vor Updates, sollten aber nicht dauerhaft bestehen bleiben: Lange offene Checkpoints kosten Leistung und Speicherplatz, und sie ersetzen kein Backup.

Dynamic Memory weist einer VM Arbeitsspeicher nach tatsächlichem Bedarf zu. Sie legen einen Startwert, einen Minimal- und einen Maximalwert fest, und Hyper-V passt die Zuteilung im laufenden Betrieb an. Das erhöht die Zahl der VMs, die ein Host tragen kann. Für manche Arbeitslasten, etwa bestimmte Datenbankserver, die ihren Speicher selbst verwalten, ist ein fest zugewiesener Arbeitsspeicher jedoch die bessere Wahl.

> **Merke**
>
> Für Produktivsysteme Production-Checkpoints verwenden und sie nach erfolgreichem Test wieder entfernen. Checkpoints sind Momentaufnahmen für kurze Zeit, keine Datensicherung.

Verwaltung, Integration und Verschachtelung

Für die Verwaltung stehen mehrere Werkzeuge bereit. Der **Hyper-V-Manager** genügt für einzelne Hosts. Große Umgebungen mit vielen Hosts und Clustern werden mit dem **System Center Virtual Machine Manager** (SCVMM) verwaltet. Das **PowerShell-Modul** für Hyper-V erlaubt es, nahezu jede Aufgabe zu automatisieren, und das **Windows Admin Center** bietet eine moderne Weboberfläche im Browser. Ein einfaches Beispiel für die PowerShell:

```
Get-VM | Where-Object State -eq 'Running'
```

In den VMs sorgen die **Integration Services** für optimierte Treiber und Dienste wie Zeitsynchronisierung und sauberes Herunterfahren; in aktuellen Windows-Versionen sind sie bereits enthalten. Mit **Nested Virtualization** lässt sich Hyper-V innerhalb einer VM betreiben – ideal für Schulungs- und Laborumgebungen. Und das **Resource Metering** misst, wie viel CPU, Arbeitsspeicher, Speicher und Netzwerk eine VM verbraucht, etwa für eine interne Verrechnung.

Das Wichtigste in Kürze

- Hyper-V ist ein in Windows Server integrierter Typ-1-Hypervisor.
- Externer, interner und privater Switch decken unterschiedliche Netzwerkzwecke ab.

- Generation 2 mit UEFI und Secure Boot bevorzugen; die Wahl ist endgültig.
- Failover-Cluster für Hochverfügbarkeit, Replica als einfache DR-Option.
- Hyper-V-Manager, SCVMM, PowerShell und Windows Admin Center zur Verwaltung.

Proxmox VE

Open-Source-Virtualisierung mit VMs und Containern

Nach diesem Kapitel können Sie …

- erklären, wie Proxmox VE KVM-VMs und LXC-Container vereint,
- Cluster, Quorum und Hochverfügbarkeit in Proxmox beschreiben,
- die Speicheroptionen einschließlich ZFS und Ceph einordnen,
- Backups mit vzdump und Proxmox Backup Server planen,
- Chancen und Aufwand einer Migration von VMware abwägen.

Was Proxmox VE ist

Proxmox Virtual Environment, kurz Proxmox VE, ist eine offene Virtualisierungsplattform auf Basis von Debian Linux. Sie vereint zwei Technologien unter einer gemeinsamen Oberfläche: vollwertige virtuelle Maschinen und leichtgewichtige Container. Proxmox VE ist als Open-Source-Software kostenlos nutzbar. Wer professionelle Unterstützung und besonders gründlich getestete Updates benötigt, kann ein optionales Support-Abonnement abschließen.

Verwaltet wird Proxmox über eine **Weboberfläche**, die im Browser läuft und keine zusätzliche Verwaltungssoftware erfordert. Dort lassen sich VMs und Container anlegen, Speicher einbinden, Backups planen und Cluster verwalten. Für die Automatisierung stehen außerdem eine Kommandozeile und eine Programmierschnittstelle zur Verfügung. Diese Kombination aus Funktionsumfang und niedrigen Einstiegskosten macht Proxmox in kleinen und mittleren Unternehmen sowie in Homelabs sehr beliebt.

Zwei Welten in einer Oberfläche: KVM und LXC

Für virtuelle Maschinen nutzt Proxmox **KVM** (Kernel-based Virtual Machine), die im Linux-Kernel integrierte Virtualisierungstechnik. KVM-VMs sind vollwertige virtuelle Rechner mit eigenem Kernel. Darin lässt sich nahezu jedes Betriebssystem betreiben, von Windows Server über Linux bis zu Firewall-Distributionen.

Daneben bietet Proxmox **LXC**-Container (Linux Containers). Sie teilen sich den Kernel des Hosts und sind dadurch deutlich schlanker: Sie starten in Sekunden, benötigen wenig Arbeitsspeicher und kaum Speicherplatz. Weil sie den Linux-Kernel des Hosts mitnutzen, können darin jedoch nur Linux-Systeme laufen, und die Isolation ist schwächer als bei einer VM. LXC-Container eignen sich deshalb besonders für einzelne, schlanke Linux-Dienste wie einen Webserver, einen DNS-Dienst oder ein Monitoring-Werkzeug.

Merkmal	KVM-VM	LXC-Container
Kernel	Eigener Kernel	Teilt den Kernel des Hosts
Betriebssysteme	Beliebig (Windows, Linux, …)	Nur Linux
Ressourcenbedarf	Höher	Sehr gering
Isolation	Stark	Geringer

Clustering und Hochverfügbarkeit

Mehrere Proxmox-Server, sogenannte **Knoten**, lassen sich zu einem **Cluster** zusammenschließen. Danach werden alle Knoten zentral über dieselbe Weboberfläche verwaltet, unabhängig davon, mit welchem Knoten man sich verbunden hat. VMs und Container können zwischen den Knoten migriert werden, bei gemeinsamem Speicher auch im laufenden Betrieb.

Damit ein Cluster konsistent bleibt, arbeitet er mit einem **Quorum**: Entscheidungen dürfen nur getroffen werden, wenn mehr als die Hälfte der Knoten erreichbar ist. So wird verhindert, dass zwei voneinander getrennte Teile des Clusters gleichzeitig dieselbe VM starten und Daten beschädigen. Für ein zuverlässiges Quorum sind mindestens drei Stimmen nötig – bei nur zwei Servern kann ein zusätzliches, kleines Gerät als dritte Stimme dienen. Auf dieser Grundlage bietet Proxmox **HA** (High Availability): Fällt ein Knoten aus, werden die als hochverfügbar markierten VMs automatisch auf einem anderen Knoten neu gestartet.

Speicheroptionen und Ceph

Proxmox unterstützt viele Speicherarten. Neben lokalem Speicher auf den Knoten selbst können Netzwerkspeicher per **NFS** oder **iSCSI** eingebunden werden. Sehr beliebt ist **ZFS**, ein Dateisystem mit integriertem Software-RAID, Prüfsummen gegen schleichende Datenfehler, Komprimierung und effizienten Snapshots.

Besonders interessant für Cluster ist **Ceph**, ein verteiltes Speichersystem, das direkt in Proxmox integriert ist. Ceph verteilt die Daten redundant über die lokalen Datenträger mehrerer Knoten. Fällt ein Knoten oder eine Festplatte aus, bleiben die Daten auf den anderen Knoten verfügbar. So entsteht gemeinsamer Speicher ohne separates Speichersystem – eine Voraussetzung für Live-Migration und HA. Ceph braucht allerdings mindestens drei Knoten und eine schnelle Netzwerkverbindung zwischen ihnen. Snapshots und Backups sind für alle Speicherarten in Proxmox integriert.

Backup, Vorlagen und Klone

Proxmox bringt die Datensicherung direkt mit. Das Werkzeug **vzdump** sichert komplette VMs und Container, wahlweise im laufenden Betrieb. Zeitpläne lassen sich pro VM oder Container in der Weboberfläche festlegen. Für größere Umgebungen gibt es den **Proxmox Backup Server** (PBS), eine eigenständige Lösung, die Sicherungen inkrementell überträgt und *dedupliziert* – identische Datenblöcke werden nur einmal gespeichert. Das spart Speicherplatz und verkürzt die Sicherungszeiten erheblich. Wie bei jeder Backup-Lösung gilt: Die Wiederherstellung muss regelmäßig getestet werden.

Für das schnelle Ausrollen lassen sich fertig eingerichtete VMs oder Container in eine **Vorlage** (Template) umwandeln. Daraus entstehen neue Systeme als **Full Clone** – eine eigenständige Kopie – oder als **Linked Clone**, der sich die Basis mit der Vorlage teilt und dadurch viel Speicher spart.

> **Merke**
>
> Proxmox VE kombiniert VMs (KVM) und Container (LXC), Clustering mit HA, verteilten Speicher mit Ceph und integriertes Backup – ohne Lizenzkosten für die Software selbst.

Sicherheit und Migration von VMware

Proxmox enthält eine integrierte **Firewall**, die sich auf Cluster-, Host- und VM-Ebene konfigurieren lässt. Über eine **rollenbasierte Rechteverwaltung** erhalten Nutzerinnen und Nutzer nur die Berechtigungen, die sie benötigen. Für die Weboberfläche sollte unbedingt eine **Zwei-Faktor-Authentifizierung** aktiviert werden. Updates kommen aus Paketquellen: Das *Enterprise-Repository* setzt ein Abonnement voraus und liefert besonders stabile Pakete, das *No-Subscription-Repository* ist frei zugänglich und eher für Test- und kleinere Umgebungen gedacht. Achten Sie darauf, dass das passende Repository eingetragen ist.

Nach den Lizenzänderungen bei VMware gilt Proxmox als eine der meistgenannten Alternativen. Bestehende VMs lassen sich importieren, und die Kosten sinken häufig deutlich. Unterschätzen Sie aber nicht den Aufwand: Bedienung, Begriffe und Abläufe unterscheiden sich, das Team muss geschult werden, und auch Backup-Software und Überwachung müssen zur neuen Plattform passen.

> **Aus der Praxis**
>
> Planen Sie eine Migration schrittweise: Bauen Sie zuerst einen Test-Cluster auf, migrieren Sie unkritische VMs und sammeln Sie Erfahrungen mit Backup, Wiederherstellung und Updates, bevor die wichtigen Systeme folgen.

Das Wichtigste in Kürze

- Proxmox VE vereint KVM-VMs und LXC-Container in einer Weboberfläche.
- Cluster mit Quorum und HA starten VMs bei Ausfall automatisch neu.
- ZFS und Ceph bieten leistungsfähigen, redundanten Speicher.
- vzdump und Proxmox Backup Server sichern dedupliziert und inkrementell.
- Eine ernsthafte, günstige VMware-Alternative – mit Schulungsaufwand.

Docker & Container-Grundlagen

Anwendungen leichtgewichtig verpacken

Nach diesem Kapitel können Sie …

- Container und virtuelle Maschinen voneinander abgrenzen,
- die Begriffe Image, Container, Dockerfile und Registry erklären,
- Daten mit Volumes dauerhaft speichern und Dienste über Ports erreichbar machen,
- mehrteilige Anwendungen mit Docker Compose beschreiben,
- grundlegende Sicherheitsregeln für Container anwenden.

Container und VMs: der entscheidende Unterschied

Eine virtuelle Maschine bildet einen kompletten Computer nach – mit eigener virtueller Hardware und eigenem Betriebssystem-Kernel. Das ist sehr flexibel, aber auch aufwendig: Jede VM braucht einige Gigabyte Speicher und startet so lange wie ein echter Rechner. **Container** gehen einen anderen Weg. Sie teilen sich den **Kernel des Hosts** und verpacken nur die Anwendung selbst mit allen Bibliotheken und Einstellungen, die sie benötigt.

Dadurch sind Container deutlich leichter. Sie starten in Sekunden, verbrauchen wenig Ressourcen und lassen sich in großer Zahl auf einem Host betreiben. Die Isolation erfolgt auf *Prozessebene*: Das Betriebssystem grenzt Container mit eigenen Mechanismen voneinander ab, sodass jeder seine eigene Sicht auf Dateisystem, Prozesse und Netzwerk hat. Diese Trennung ist allerdings schwächer als bei einer VM, die auf Hardware-Ebene isoliert ist. In der Praxis werden beide Technologien oft kombiniert: Container laufen innerhalb von VMs.

Ein weiterer großer Vorteil ist die Reproduzierbarkeit. Ein Container enthält alles, was die Anwendung braucht, und verhält sich deshalb auf dem Notebook einer Entwicklerin genauso wie auf dem Produktivserver. Das bekannte Problem „Auf meinem Rechner lief es noch“ tritt damit deutlich seltener auf.

Die Docker-Architektur

Docker ist die bekannteste Plattform für Container. Ihr Kern ist die **Docker Engine**, ein Hintergrunddienst (Daemon), der Container erstellt, startet und überwacht. Gesteuert wird sie über die Kommandozeile `docker` oder über eine Programmierschnittstelle. Un-

ter Linux läuft Docker nativ, weil Container den Linux-Kernel nutzen. Unter Windows und macOS arbeitet Docker mit einer kleinen virtuellen Maschine im Hintergrund, die diesen Kernel bereitstellt.

Die Vorlagen für Container stammen aus einer **Registry**. Die bekannteste öffentliche Registry ist Docker Hub; Unternehmen betreiben häufig zusätzlich eigene, private Registries für ihre selbst gebauten Anwendungen.

Images, Container und das Dockerfile

Zwei Begriffe sind zentral. Ein **Image** ist eine unveränderliche Vorlage, die Betriebssystem-Grundlage, Anwendung und Abhängigkeiten enthält. Ein **Container** ist eine laufende Instanz dieses Images. Aus einem Image können beliebig viele Container gestartet werden – ähnlich wie aus einer Kopiervorlage beliebig viele Kopien entstehen. Container sind bewusst *kurzlebig und ersetzbar*: Statt einen laufenden Container zu verändern, baut man ein neues Image und ersetzt den alten Container durch einen neuen.

Gebaut wird ein Image aus einem **Dockerfile**, einer Textdatei, die Schritt für Schritt beschreibt, wie das Image entsteht. Die wichtigsten Anweisungen sind `FROM` (das Basis-Image), `RUN` (Befehle beim Bauen ausführen), `COPY` (Dateien hineinkopieren) und `CMD` (der Befehl, der beim Start des Containers ausgeführt wird):

```
FROM python:3-alpine
COPY app.py /app/
RUN pip install flask
CMD ["python", "/app/app.py"]
```

Jede Anweisung erzeugt eine eigene **Schicht** (Layer). Unveränderte Schichten werden beim nächsten Bauen aus dem Zwischenspeicher übernommen, was den Vorgang stark beschleunigt. Wählen Sie möglichst kleine Basis-Images, etwa auf Grundlage von Alpine Linux. Mit *Multi-Stage-Builds* lässt sich das Endergebnis zusätzlich schlank halten: Werkzeuge, die nur zum Bauen nötig sind, gelangen gar nicht erst ins fertige Image.

Versionen werden über **Tags** gekennzeichnet, zum Beispiel `app:1.2`. Ein festes Tag sorgt dafür, dass immer genau dieselbe Version eingesetzt wird.

Daten, Netzwerk und Konfiguration

Weil Container ersetzbar sind, geht alles verloren, was nur im Container selbst gespeichert ist. Dauerhafte Daten – etwa die Dateien einer Datenbank – gehören deshalb in ein **Volume**, einen Speicherbereich außerhalb des Containers, der auch nach dem Ersetzen erhalten bleibt. Damit ein Dienst von außen erreichbar ist, wird ein **Port** des Hosts auf einen Port des Containers abgebildet. Container, die miteinander sprechen sollen, ver-

bindet man über eigene **Docker-Netzwerke**. Die Konfiguration, etwa Datenbankadressen, wird meist über **Umgebungsvariablen** übergeben, damit dasselbe Image in verschiedenen Umgebungen funktioniert.

```
docker run -d -p 8080:80 -v webdaten:/usr/share/nginx/html nginx
```

Dieser Befehl startet einen Webserver im Hintergrund, macht ihn über Port 8080 des Hosts erreichbar und speichert seine Inhalte in einem Volume.

> **Merke**
>
> Image = Vorlage, Container = laufende Instanz. Container sind ersetzbar – alles, was dauerhaft bleiben soll, gehört in ein Volume.

Docker Compose: mehrere Container als Einheit

Viele Anwendungen bestehen aus mehreren Teilen, etwa aus der eigentlichen Anwendung, einer Datenbank und einem Cache. Mit **Docker Compose** werden alle diese Container gemeinsam in einer Datei `docker-compose.yml` beschrieben – inklusive Volumes, Netzwerken und Umgebungsvariablen. Ein einziger Befehl wie `docker compose up -d` startet dann den gesamten Stack. Weil die Datei als Text vorliegt, lässt sie sich versionieren und jederzeit reproduzierbar auf einem anderen System einsetzen.

Sicherheit und Betrieb

Container sind nur so sicher wie die Images, aus denen sie entstehen. Verwenden Sie ausschließlich **vertrauenswürdige Images** aus offiziellen oder geprüften Quellen und halten Sie sie aktuell, denn auch Images enthalten Software mit bekannten Schwachstellen (CVEs). Scannen Sie Images regelmäßig auf Sicherheitslücken; mehr dazu in Lernbereich 7. Anwendungen sollten im Container **nicht als root** laufen, damit ein Angreifer bei einer Lücke nicht sofort volle Rechte hat. Und Passwörter oder Schlüssel gehören nie fest ins Image, sondern werden zur Laufzeit übergeben.

Für die Beobachtung im Betrieb zeigen `docker logs` die Ausgaben eines Containers und `docker stats` den aktuellen Ressourcenverbrauch. Setzen Sie **Ressourcenlimits** für CPU und Arbeitsspeicher, damit ein einzelner Container nicht den ganzen Host ausbremst, und legen Sie **Restart-Policies** fest, damit abgestürzte Container automatisch neu starten. Eine Grenze von Docker ist jedoch: Es verwaltet Container auf einem einzelnen Host. Sollen viele Container über mehrere Server hinweg betrieben, verteilt und automatisch skaliert werden, kommt Kubernetes ins Spiel – das Thema des nächsten Kapitels.

Das Wichtigste in Kürze

- Container teilen den Kernel des Hosts und sind dadurch leicht und schnell.
- Ein Image ist die unveränderliche Vorlage, gebaut aus einem Dockerfile; ein Container ist die laufende Instanz.
- Volumes speichern Daten dauerhaft, Ports machen Dienste erreichbar.
- Docker Compose bündelt mehrteilige Anwendungen in einer Datei.
- Vertrauenswürdige, aktuelle Images verwenden, nicht als root laufen, keine Secrets im Image.

Kubernetes-Einstieg

Container im großen Maßstab orchestrieren

Nach diesem Kapitel können Sie …

- erklären, wann Kubernetes gegenüber Docker allein sinnvoll ist,
- die Grundbegriffe Pod, Deployment, Service und Namespace zuordnen,
- das Prinzip des gewünschten Zustands mit Selbstheilung und Rolling Updates beschreiben,
- Konfiguration, Secrets und externen Zugriff per Ingress einordnen,
- mit kubectl erste Diagnoseschritte durchführen.

Warum Kubernetes?

Docker eignet sich hervorragend, um Container auf einem einzelnen Server zu betreiben. Sobald jedoch viele Container über mehrere Server verteilt werden sollen, stößt dieser Ansatz an Grenzen. Wer entscheidet, auf welchem Server ein Container läuft? Was passiert, wenn ein Server ausfällt? Wie wird eine Anwendung bei steigender Last automatisch vervielfältigt, und wie spielt man ein Update ein, ohne dass die Anwendung kurz nicht erreichbar ist?

Diese Aufgaben übernimmt **Kubernetes**, oft kurz *K8s* genannt. Es ist eine Plattform zur **Orchestrierung** von Containern: Kubernetes verteilt Container automatisch auf eine Gruppe von Servern, den *Cluster*, überwacht sie, ersetzt ausgefallene Instanzen und skaliert nach Bedarf. Kubernetes hat sich als De-facto-Standard für die Container-Orchestrierung durchgesetzt und wird sowohl im eigenen Rechenzentrum als auch bei allen großen Cloud-Anbietern eingesetzt.

Die Grundbegriffe

Kubernetes arbeitet mit einer Reihe von Objekten, die Sie kennen sollten. Ein **Pod** ist die kleinste Einheit, die Kubernetes verwaltet. Er enthält einen oder mehrere Container, die eng zusammengehören und sich Netzwerkadresse und Speicher teilen. In den meisten Fällen enthält ein Pod genau einen Container.

Pods werden selten direkt angelegt. Stattdessen beschreibt ein **Deployment**, welches Image verwendet werden soll und wie viele Kopien (Replikate) eines Pods laufen sollen. Das Deployment sorgt dafür, dass diese Zahl eingehalten wird, und steuert Updates. Weil Pods kommen und gehen und dabei jedes Mal eine neue Adresse erhalten, gibt es den **Service**: eine stabile Adresse, über die alle zusammengehörenden Pods erreichbar

sind und die Anfragen auf sie verteilt. Ein **Namespace** schließlich trennt Bereiche logisch voneinander, etwa Test und Produktion oder die Anwendungen verschiedener Teams.

Begriff	Aufgabe
Pod	Kleinste Einheit, enthält einen oder mehrere Container
Deployment	Verwaltet Pods, deren Anzahl und Rollouts
Service	Stabile Adresse und Lastverteilung für Pods
Namespace	Logische Trennung von Bereichen
Ingress	Externer Zugriff und Routing zu Services

Gewünschter Zustand, Selbstheilung und Skalierung

Das Kernprinzip von Kubernetes ist der **gewünschte Zustand**. Sie beschreiben – meist in einer YAML-Datei –, wie die Anwendung aussehen soll: welches Image, wie viele Replikate, welche Ressourcen. Kubernetes vergleicht diesen Soll-Zustand laufend mit dem tatsächlichen Zustand und gleicht Abweichungen automatisch aus.

Daraus ergibt sich die **Selbstheilung**: Stürzt ein Pod ab oder fällt ein ganzer Server aus, startet Kubernetes die fehlenden Pods selbstständig neu, gegebenenfalls auf einem anderen Server. Updates laufen als **Rolling Update**: Kubernetes ersetzt die alten Pods schrittweise durch neue, sodass die Anwendung währenddessen erreichbar bleibt. Treten dabei Fehler auf, lässt sich das Update stoppen und auf die vorherige Version zurückrollen.

Bei steigender Last skaliert Kubernetes *horizontal*, also durch zusätzliche Pods statt größerer Server. Der **Horizontal Pod Autoscaler** erhöht oder verringert die Zahl der Replikate automatisch anhand von Messwerten wie der CPU-Auslastung. Damit Kubernetes die Pods sinnvoll verteilen kann, sollten für jeden Pod **Requests** (garantierte Mindestressourcen) und **Limits** (Obergrenzen) für CPU und Arbeitsspeicher festgelegt werden.

> **Merke**
>
> In Kubernetes beschreiben Sie, *was* laufen soll – nicht, *wie* es gestartet wird. Kubernetes setzt den gewünschten Zustand selbstständig durch und stellt ihn nach Störungen wieder her.

Konfiguration, Secrets und Ingress

Wie bei Docker gilt auch hier: Konfiguration gehört nicht ins Image. Nicht geheime Einstellungen wie Adressen oder Schalter werden in **ConfigMaps** abgelegt, Passwörter, Zugangsschlüssel und Zertifikate in **Secrets**. Beide werden den Pods zur Laufzeit als Umgebungsvariablen oder Dateien bereitgestellt. So lässt sich dasselbe Image unverändert in Test und Produktion verwenden. Wichtig: Secrets sind in der Grundeinstellung nur kodiert, nicht verschlüsselt. Sie müssen deshalb zusätzlich abgesichert werden – durch eine rollenbasierte Zugriffskontrolle (RBAC), die festlegt, wer sie lesen darf, und durch Verschlüsselung im Cluster-Datenspeicher.

Für den Zugriff von außen sorgt ein **Ingress**. Er legt fest, welche Anfragen – etwa anhand des Hostnamens oder des Pfads einer Webadresse – an welchen Service weitergeleitet werden. Umgesetzt wird das von einem *Ingress-Controller*, der als zentraler Eingang in den Cluster dient und in der Regel auch die TLS-Verschlüsselung übernimmt.

Steuerung mit kubectl und Betrieb

Das zentrale Werkzeug für die Arbeit mit Kubernetes ist die Kommandozeile **kubectl**. Mit ihr werden Zustände abgefragt, Konfigurationen ausgerollt und Fehler untersucht:

```
kubectl get pods
kubectl apply -f deployment.yaml
kubectl logs mein-pod
kubectl describe pod mein-pod
kubectl get events
```

`kubectl get pods` zeigt alle Pods und ihren Status, `kubectl apply -f` rollt eine YAML-Datei aus. Bei Problemen liefern `kubectl logs` die Ausgaben eines Containers, `kubectl describe` Details und Ereignisse zu einem Objekt und `kubectl get events` einen Überblick über das Geschehen im Cluster – etwa, wenn ein Image nicht geladen werden kann oder zu wenig Ressourcen frei sind.

Damit Kubernetes erkennt, ob eine Anwendung wirklich funktioniert, werden **Health-Probes** definiert. Die *Liveness-Probe* prüft, ob der Container noch lebt; schlägt sie fehl, wird er neu gestartet. Die *Readiness-Probe* prüft, ob er bereit ist, Anfragen anzunehmen; erst dann leitet der Service Verkehr an ihn weiter. Weil Pods kurzlebig sind, sollten ihre Protokolle zudem zentral eingesammelt werden; mehr dazu in Lernbereich 9.

Der Einstieg: Helm und Übungsumgebungen

Komplexe Anwendungen bestehen in Kubernetes aus vielen YAML-Dateien. **Helm** fasst sie zu sogenannten *Charts* zusammen – Paketen, die sich mit einem Befehl installieren, konfigurieren und aktualisieren lassen. Für viele gängige Anwendungen gibt es fertige Charts.

Für erste Schritte eignen sich schlanke Distributionen wie **K3s** oder **Minikube**, die auf einem einzelnen Rechner oder in einer VM laufen. Wer Kubernetes produktiv einsetzt, ohne den Cluster selbst betreiben zu wollen, kann auf verwaltete Angebote der Cloud-Anbieter zurückgreifen, etwa AKS (Microsoft Azure), EKS (Amazon Web Services) oder GKE (Google Cloud). Sie übernehmen den Betrieb der Steuerungsebene und nehmen viel Arbeit ab.

Aus der Praxis

Kubernetes ist mächtig, aber auch komplex. Fangen Sie klein an: eine Übungsumgebung mit K3s, eine einfache Anwendung, ein Deployment und ein Service. Erst wenn diese Grundkonzepte sitzen, lohnen sich Ingress, Autoscaling und Helm. Für wenige Container auf einem Server ist Docker Compose oft die einfachere Lösung.

Das Wichtigste in Kürze

- Kubernetes orchestriert Container automatisch über viele Hosts.
- Pod, Deployment, Service und Namespace sind die zentralen Bausteine.
- Der gewünschte Zustand ermöglicht Selbstheilung, Rolling Updates und Autoscaling.
- ConfigMaps und Secrets trennen Konfiguration vom Image; Secrets zusätzlich absichern.
- kubectl und Helm sind die wichtigsten Werkzeuge; K3s oder Minikube zum Lernen.

LB-06

Storage & Backup

1. RAID-Systeme
2. SAN & NAS
3. Backup-Grundlagen & Strategien
4. VSS, Snapshots & agentenloses Backup
5. Datenbank-Backup
6. Backup-Software & Hersteller

RAID-Systeme

Datenträger zu Verbünden zusammenschalten

Nach diesem Kapitel können Sie …

- erklären, was ein RAID leistet und wovor es nicht schützt,
- die RAID-Level 0, 1, 5, 6 und 10 unterscheiden und ihre nutzbare Kapazität berechnen,
- Hardware- und Software-RAID gegeneinander abwägen,
- das Rebuild-Risiko einschätzen und ein RAID sicher betreiben.

Was ist ein RAID?

Eine einzelne Festplatte oder SSD hat zwei Schwächen: Sie kann ausfallen, und ihre Geschwindigkeit ist begrenzt. Ein **RAID** (Redundant Array of Independent Disks) schaltet deshalb mehrere Datenträger zu einem Verbund zusammen, der sich für das Betriebssystem wie ein einziges Laufwerk verhält. Je nach Bauart, dem sogenannten *RAID-Level*, gewinnt man dadurch Tempo, Ausfallsicherheit oder beides.

Die meisten RAID-Level schützen davor, dass der Ausfall einer Platte zu Datenverlust und Stillstand führt: Der Server läuft weiter, die defekte Platte wird getauscht, und der Verbund baut sich wieder auf. Wovor ein RAID dagegen nicht schützt, ist ebenso wichtig – dazu später mehr.

Die wichtigsten RAID-Level

Bei **RAID 0** (Striping) werden die Daten in Blöcken abwechselnd auf mehrere Platten verteilt. Weil alle Platten gleichzeitig lesen und schreiben, ist das sehr schnell. Allerdings gibt es keinerlei Redundanz: Fällt eine Platte aus, sind die Daten des gesamten Verbunds verloren. Das „R“ für „redundant“ trägt RAID 0 also eigentlich zu Unrecht.

RAID 1 (Spiegelung) schreibt dieselben Daten gleichzeitig auf zwei Platten. Fällt eine aus, liegen alle Daten vollständig auf der anderen. Das ist einfach und sicher, halbiert aber die nutzbare Kapazität.

RAID 5 verteilt Daten und zusätzliche *Paritätsinformationen* über mindestens drei Platten. Aus der Parität lassen sich die Daten einer ausgefallenen Platte rechnerisch rekonstruieren. RAID 5 verträgt den Ausfall genau einer Platte. **RAID 6** arbeitet ähnlich, speichert aber doppelte Parität über mindestens vier Platten und verträgt deshalb zwei gleichzeitige Ausfälle.

RAID 10 kombiniert beide Prinzipien: Je zwei Platten werden gespiegelt, und über diese Spiegelpaare wird gestriped. Das Ergebnis ist schnell und sicher zugleich und wird gern für Datenbanken und Virtualisierung eingesetzt – kostet aber wie RAID 1 die Hälfte der Rohkapazität.

Level	Mindestanzahl Platten	Verträgt Ausfälle	Nutzbare Kapazität
RAID 0	2	keinen	alle Platten (n)
RAID 1	2	1	die Hälfte
RAID 5	3	1	n − 1 Platten
RAID 6	4	2	n − 2 Platten
RAID 10	4	mind. 1 (eine pro Spiegelpaar)	die Hälfte

Ein Rechenbeispiel: Sechs Platten mit je 4 TB ergeben 24 TB Rohkapazität. Als RAID 5 sind davon 20 TB nutzbar, als RAID 6 noch 16 TB, als RAID 10 nur 12 TB. Die Faustregel lautet: Mehr Sicherheit bedeutet weniger nutzbaren Platz. Welches Level passt, ergibt sich aus der Abwägung von Kapazität, Tempo und Sicherheit.

Hardware- oder Software-RAID?

Ein **Hardware-RAID** wird von einem eigenen Controller verwaltet, einer Steckkarte oder einem Baustein auf der Hauptplatine. Das Betriebssystem sieht nur das fertige Laufwerk. Gute Controller besitzen einen eigenen Schreib-Cache, der durch eine Batterie oder einen Kondensator (BBU) gegen Stromausfall geschützt ist. Ein Nachteil: Fällt der Controller selbst aus, braucht man oft ein baugleiches Ersatzgerät, um an die Daten zu kommen.

Beim **Software-RAID** übernimmt das Betriebssystem die Arbeit, zum Beispiel mit *mdadm* unter Linux, *Storage Spaces* unter Windows oder dem Dateisystem *ZFS*. Moderne Prozessoren haben genug Leistung dafür, und die Platten lassen sich bei Bedarf auch in einem anderen Server wieder einbinden. ZFS bietet zusätzlich **Prüfsummen** für alle Daten und erkennt dadurch auch stille Datenfehler, die ein klassisches RAID nicht bemerkt.

Das Rebuild-Risiko

Fällt eine Platte aus, läuft der Verbund im *degradierten* Zustand weiter, also ohne volle Redundanz. Nach dem Tausch beginnt der **Rebuild**: Die neue Platte wird aus den Daten und der Parität der übrigen Platten neu beschrieben. Dabei werden alle verbleibenden Platten stark belastet – und genau in dieser Phase droht ein zweiter Ausfall, zumal Platten aus derselben Lieferung oft ähnlich gealtert sind.

Bei heutigen großen Platten kann ein Rebuild viele Stunden bis Tage dauern. Bei einem RAID 5 bedeutet ein zweiter Ausfall oder ein nicht lesbarer Sektor während dieser Zeit den Verlust des Verbunds. **RAID 6** puffert diesen Fall ab, weil es einen zweiten Ausfall verkraftet. Deshalb wird für große Platten und wichtige Daten RAID 6 oder RAID 10 empfohlen.

> **Merke**
>
> Je größer die Platten, desto länger der Rebuild und desto höher das Risiko eines zweiten Ausfalls. Für wichtige Systeme mit großen Platten sind RAID 6 oder RAID 10 die bessere Wahl als RAID 5.

Hot-Spare, Überwachung und Prüfläufe

Ein **Hot-Spare** ist eine eingebaute Reserveplatte, die im Normalbetrieb ungenutzt bleibt. Fällt eine Platte aus, springt der Hot-Spare automatisch ein und der Rebuild beginnt sofort – ohne dass erst jemand eine Ersatzplatte besorgen und einbauen muss. Das verkürzt die Zeit im ungeschützten Zustand erheblich.

Ein RAID nützt wenig, wenn niemand bemerkt, dass eine Platte ausgefallen ist. Aktivieren Sie deshalb die **Alarme** des Controllers oder Betriebssystems und werten Sie die **SMART-Werte** der Platten aus, die frühe Anzeichen eines Defekts liefern. Regelmäßige Prüfläufe – je nach System *Scrubbing* oder *Patrol Read* genannt – lesen alle Daten und finden so stille Fehler, bevor sie bei einem Rebuild zum Problem werden. Ausgefallene Platten sollten Sie zeitnah tauschen.

Weitere Praxistipps: Verwenden Sie möglichst **identische Platten** in Größe, Typ und Firmware, halten Sie die Controller-Firmware aktuell und prüfen Sie bei Hardware-RAID regelmäßig den Zustand der Cache-Batterie. Planen Sie die Rebuild-Dauer bei großen Platten von vornherein ein.

RAID ist kein Backup

Der wichtigste Satz dieses Kapitels: Ein RAID ersetzt niemals ein Backup. Es schützt ausschließlich vor dem Ausfall einzelner Datenträger. Löscht jemand versehentlich einen Ordner, wird die Löschung sofort auf allen Platten ausgeführt – gelöschte Daten bleiben gelöscht. Verschlüsselt Ransomware die Daten, verschlüsselt sie den gesamten Verbund mit. Auch gegen Brand, Diebstahl, Überspannung oder einen defekten Controller, der Daten falsch schreibt, hilft ein RAID nicht.

Aus der Praxis

Betrachten Sie ein RAID als Maßnahme für die Verfügbarkeit, nicht für die Datensicherung. Zusätzlich ist immer ein echtes Backup auf einem getrennten System zwingend.

Das Wichtigste in Kürze

- Ein RAID verbindet Platten für Tempo und/oder Ausfallsicherheit.
- RAID-Level nach Tempo, Sicherheit und Kapazität wählen.
- RAID 10 oder RAID 6 für wichtige Systeme.
- Rebuild-Risiko bei großen Platten beachten, Hot-Spare und Alarme nutzen.
- RAID ersetzt niemals ein Backup.

SAN & NAS

Zentraler Speicher im Netzwerk

Nach diesem Kapitel können Sie …

- den Unterschied zwischen datei- und blockbasiertem Speicher erklären,
- typische Einsatzgebiete von NAS und SAN benennen,
- die Protokolle SMB, NFS, iSCSI und Fibre Channel zuordnen,
- Speichersysteme nach Leistung, Verfügbarkeit und Kapazität dimensionieren.

Warum zentraler Speicher?

Solange jeder Server seine Daten auf eigenen, eingebauten Platten speichert, ist der Speicher auf viele Geräte verteilt. Auf dem einen Server wird es eng, auf dem anderen liegt ungenutzter Platz, und jedes System muss einzeln gesichert und überwacht werden. Zentraler Speicher im Netzwerk bündelt die Kapazität an einer Stelle und stellt sie den Servern oder Benutzern bereit. Dafür gibt es zwei grundlegend verschiedene Ansätze: NAS und SAN.

NAS und SAN: Dateien oder Blöcke

Der zentrale Unterschied liegt in der Ebene, auf der zugegriffen wird. Ein **NAS** (Network Attached Storage) arbeitet *dateibasiert*. Es verwaltet selbst ein Dateisystem und stellt Ordner als Freigaben bereit, auf die Clients über das Netzwerk zugreifen – ganz so, wie man es von einem Netzlaufwerk kennt. Das NAS weiß, welche Dateien es speichert, und kümmert sich um Rechte und Sperren.

Ein **SAN** (Storage Area Network) arbeitet dagegen *blockbasiert*. Es stellt Servern rohe Speicherbereiche bereit, sogenannte **LUNs** (Logical Unit Numbers). Für den Server sieht eine LUN aus wie eine lokal eingebaute Festplatte: Er formatiert sie mit seinem eigenen Dateisystem und verwaltet sie selbst. Das Speichersystem kennt keine Dateien, sondern nur Datenblöcke.

Merkmal	NAS	SAN
Zugriff	Dateibasiert	Blockbasiert (LUNs)
Protokolle	SMB, NFS	iSCSI, Fibre Channel
Dateisystem	Auf dem NAS	Auf dem Server
Typischer Einsatz	Freigaben, Backups, Medien	VM-Datastores, Datenbanken
Betrieb	Einfach	Anspruchsvoller

> **Merke**
>
> NAS = Dateien teilen (SMB/NFS). SAN = Blöcke bereitstellen, die der Server wie eine eigene Platte nutzt (iSCSI/Fibre Channel).

Das NAS im Einsatz

Ein NAS ist im Grunde ein spezialisierter Dateiserver in einem kompakten Gehäuse, meist mit mehreren Platten im RAID-Verbund und einer einfachen Weboberfläche. Windows-Clients greifen über **SMB** zu, Linux- und Unix-Systeme häufig über **NFS** (Network File System). Viele Geräte beherrschen beide Protokolle gleichzeitig.

Typische Aufgaben sind Dateifreigaben für Abteilungen, Ablage für Backups und die Speicherung großer Mediendateien. Bekannte Hersteller im kleinen und mittleren Segment sind etwa QNAP und Synology; im Unternehmensumfeld gibt es zudem größere Systeme vieler weiterer Anbieter. Für die meisten kleinen und mittleren Umgebungen ist ein NAS die praktische Wahl, weil es sich ohne Spezialwissen einrichten und betreiben lässt.

Das SAN-Konzept

Ein SAN stellt Servern Blockspeicher über ein **eigenes, schnelles Speichernetz** bereit, das vom normalen Datennetz getrennt ist. So stört der hohe Speicherverkehr nicht die Kommunikation der Benutzer und umgekehrt. Seine Stärken spielt ein SAN überall dort aus, wo viele Server gemeinsam auf schnellen Speicher zugreifen müssen – vor allem bei **Virtualisierung**, wo mehrere Hypervisor-Hosts auf dieselben Datastores zugreifen, damit virtuelle Maschinen im laufenden Betrieb zwischen den Hosts wandern können, und bei **Datenbanken** mit hoher Last.

Für hohe Verfügbarkeit ist ein SAN über **mehrere Pfade** angebunden: Jeder Server hat mindestens zwei Verbindungen über getrennte Switches zum Speichersystem. Die Technik dafür heißt **Multipathing**. Fällt ein Kabel, ein Switch oder ein Netzwerkanschluss

aus, läuft der Verkehr unterbrechungsfrei über den anderen Pfad weiter, und im Normalbetrieb kann die Last auf beide Pfade verteilt werden.

Anbindung: iSCSI und Fibre Channel

iSCSI transportiert Blockspeicher-Befehle über gewöhnliches Ethernet und TCP/IP. Dabei verbindet sich der *Initiator* – der Server – mit dem *Target* – dem Speichersystem. Weil normale Netzwerkkarten und Switches genutzt werden können, ist iSCSI vergleichsweise günstig und flexibel. Für gute Leistung sollte es dennoch über ein eigenes Netz oder zumindest ein eigenes VLAN laufen. **Jumbo Frames**, also größere Ethernet-Pakete, können den Durchsatz dabei erhöhen, müssen aber auf allen beteiligten Geräten einheitlich eingestellt sein.

Fibre Channel (FC) ist ein eigenständiges, dediziertes Speichernetz mit eigenen Switches und Adapterkarten (HBAs). Es ist auf Speicherverkehr spezialisiert, sehr schnell und latenzarm, aber teurer und erfordert spezielles Know-how. Fibre Channel findet sich vor allem in größeren Rechenzentren. In beiden Fällen gehört Multipathing zum Standard.

Aus der Praxis

Für kleinere Virtualisierungsumgebungen ist iSCSI über ein getrenntes, redundant aufgebautes Netz meist ausreichend und deutlich günstiger als Fibre Channel. Entscheidend ist, dass Multipathing auf beiden Seiten korrekt eingerichtet ist.

Verfügbarkeit des Speichersystems

Zentraler Speicher ist ein zentraler Ausfallpunkt: Fällt er aus, stehen alle angeschlossenen Server still. Professionelle Speichersysteme haben deshalb **redundante Controller** – fällt einer aus, übernimmt der andere. Netzteile und Lüfter sind ebenfalls doppelt vorhanden. **Snapshots** auf Ebene des Speichersystems ermöglichen eine schnelle Rückkehr zu einem früheren Zustand, etwa nach einem fehlgeschlagenen Update. Für Katastrophenfälle wie Brand oder Wasserschaden lassen sich Daten per **Replikation** auf ein zweites System an einem anderen Standort spiegeln. Ein Backup ersetzen diese Funktionen jedoch nicht, da Snapshots auf demselben System liegen und Replikation auch Fehler sofort mitkopiert.

Auswahl und Dimensionierung

Am Anfang jeder Auswahl steht die Frage nach dem Bedarf: Sollen Dateien geteilt werden, oder brauchen Server schnellen Blockspeicher? Danach zählt nicht nur die **Kapazität**, sondern ebenso die **Leistung**. Sie wird in **IOPS** (Ein- und Ausgabeoperationen pro

Sekunde) und **Durchsatz** (Datenmenge pro Sekunde) gemessen. Datenbanken und Virtualisierung erzeugen viele kleine Zugriffe und brauchen hohe IOPS, Backups und Videodateien eher hohen Durchsatz.

Planen Sie zur aktuell benötigten Kapazität eine **Reserve für Wachstum** ein und prüfen Sie, wie sich das System später erweitern lässt. Weitere Kriterien sind Redundanz und Skalierbarkeit, das Budget, der Betriebsaufwand für das eigene Team sowie der **Herstellersupport**: Wie schnell kommt ein Ersatzteil, und wie lange wird das Modell noch mit Updates versorgt?

Das Wichtigste in Kürze

- NAS arbeitet dateibasiert, SAN blockbasiert.
- SMB und NFS für NAS, iSCSI und Fibre Channel für SAN.
- Ein SAN eignet sich für VMs und Datenbanken mit hoher Last.
- Multipathing und redundante Controller sichern die Verfügbarkeit.
- Dimensionieren nach Kapazität, IOPS und Durchsatz – mit Reserve.

Backup-Grundlagen & Strategien

Daten zuverlässig sichern und wiederherstellen

Nach diesem Kapitel können Sie …

- die 3-2-1-1-0-Regel erklären und anwenden,
- Voll-, differenzielle, inkrementelle und synthetische Sicherungen unterscheiden,
- RPO und RTO definieren und ihre Auswirkungen auf die Strategie einschätzen,
- Aufbewahrung, Sicherungsziele und einen Notfallplan sinnvoll planen.

Warum Backups unverzichtbar sind

Daten gehen auf viele Arten verloren: Hardware fällt aus, Menschen löschen versehentlich Dateien oder überschreiben sie, Software-Updates schlagen fehl, und Ransomware verschlüsselt ganze Netzwerke. Hinzu kommen Brand, Wasserschaden oder Diebstahl. Gegen all diese Szenarien gibt es letztlich nur eine verlässliche Absicherung: eine funktionierende Datensicherung. Das Backup ist die **letzte Verteidigungslinie** – wenn alle anderen Schutzmaßnahmen versagt haben, entscheidet es darüber, ob ein Unternehmen weiterarbeiten kann. Ein vollständiger Datenverlust kann Existenzen kosten.

Dabei zählt nicht das Backup selbst, sondern die Wiederherstellung. Eine Sicherung, die nie testweise zurückgespielt wurde, ist nur eine Hoffnung. Erst ein erfolgreich getesteter **Restore** beweist, dass die Daten im Ernstfall wirklich zur Verfügung stehen.

Die 3-2-1-1-0-Regel

Als Leitlinie für eine robuste Sicherungsstrategie hat sich die **3-2-1-1-0-Regel** etabliert, eine Erweiterung der klassischen 3-2-1-Regel:

Ziffer	Bedeutung	Schützt vor
3	Drei Kopien der Daten (Original plus zwei Sicherungen)	Ausfall einer einzelnen Kopie
2	Auf zwei verschiedenen Medientypen	Medienspezifische Fehler
1	Eine Kopie außer Haus	Brand, Wasser, Diebstahl am Standort
1	Eine Kopie offline oder unveränderlich	Ransomware, Sabotage
0	Null Fehler beim Restore-Test	Unbemerkt defekte Sicherungen

Besonders wichtig ist heute die zweite Eins. Ransomware sucht gezielt nach erreichbaren Backups und verschlüsselt oder löscht sie mit. Eine Kopie, die physisch vom Netz getrennt ist (*Air-Gap*) oder technisch nicht verändert werden kann (*immutable*), bleibt davon unberührt. Die Null erinnert daran, die Sicherungen regelmäßig zu verifizieren.

> **Merke**
>
> 3 Kopien · 2 Medien · 1 extern · 1 offline oder unveränderlich · 0 Fehler beim Restore-Test.

Backup-Arten

Die **Vollsicherung** sichert bei jedem Lauf alle ausgewählten Daten. Sie ist am einfachsten wiederherzustellen, weil alles in einer Sicherung steckt, benötigt aber viel Zeit und Speicherplatz.

Die **differenzielle Sicherung** erfasst alle Änderungen seit der letzten Vollsicherung. Sie wächst von Tag zu Tag, bis die nächste Vollsicherung läuft. Für die Wiederherstellung brauchen Sie nur zwei Teile: die letzte Vollsicherung und die letzte differenzielle Sicherung.

Die **inkrementelle Sicherung** erfasst nur die Änderungen seit dem jeweils letzten Backup – egal welcher Art. Sie ist besonders schnell und sparsam. Dafür entsteht eine längere Kette: Für eine Wiederherstellung werden die Vollsicherung und alle folgenden Inkremente benötigt, und ist ein Glied defekt, sind die späteren Stände nicht mehr verwendbar.

Die **synthetische Vollsicherung** löst dieses Dilemma: Die Backup-Software setzt aus einer vorhandenen Vollsicherung und den Inkrementen nachträglich eine neue Vollsicherung zusammen – ohne die Daten erneut von den Quellsystemen zu lesen. In der Praxis werden die Arten bewusst kombiniert, etwa eine wöchentliche (synthetische) Vollsicherung mit täglichen Inkrementen.

RPO und RTO: Wie viel Verlust, wie viel Ausfall?

Zwei Kennzahlen bestimmen, wie eine Sicherungsstrategie aussehen muss. Das **RPO** (Recovery Point Objective) beantwortet die Frage: Wie viel Datenverlust ist akzeptabel? Wird nur einmal pro Nacht gesichert, können im schlimmsten Fall die Daten eines ganzen Arbeitstages verloren gehen. Ist das nicht tragbar, muss häufiger gesichert werden. Das RPO bestimmt damit das Backup-Intervall.

Das **RTO** (Recovery Time Objective) beantwortet die Frage: Wie schnell muss ein System wieder laufen? Eine Wiederherstellung aus der Cloud kann bei großen Datenmengen Tage dauern, eine lokale Sicherung auf Festplatte nur Stunden. Wer ein kurzes RTO

braucht, muss in schnelle Wiederherstellungswege investieren.

Beide Werte bestimmen Strategie und Kosten – je kürzer, desto teurer. Sie sind keine rein technische Entscheidung und sollten deshalb mit der Geschäftsführung und den Fachabteilungen abgestimmt werden: Die Buchhaltung kann vielleicht einen Tag warten, der Webshop nicht.

Wohin und was sichern?

Für die Sicherungsziele gibt es verschiedene Medien mit eigenen Stärken. **Disk-to-Disk**, also die Sicherung auf Festplatten in einem Backup-Server oder NAS, ermöglicht eine schnelle Wiederherstellung. **Bänder** sind günstig für große Datenmengen und lange Archivierung und lassen sich nach dem Beschreiben physisch vom System trennen. Die **Cloud** eignet sich gut als externes Exemplar außer Haus. Für den Schutz vor Ransomware sollte mindestens ein Ziel per Air-Gap getrennt oder unveränderlich sein.

Nicht alle Daten sind gleich wichtig, deshalb lohnt eine **Priorisierung**. Neben den Nutzdaten gehören System-State und Konfigurationen von Servern, Firewalls und Switches ins Backup. Ganze virtuelle Maschinen und Datenbanken werden mit eigenen, dafür geeigneten Methoden gesichert. Kritische Systeme sichern Sie häufiger und stellen sie im Notfall zuerst wieder her. Und vergessen Sie die **Dokumentation** nicht – Passwörter, Netzpläne und Wiederherstellungsanleitungen müssen auch dann verfügbar sein, wenn die Server ausgefallen sind.

Aufbewahrung

Wie lange werden Sicherungen aufbewahrt? Das legen **Aufbewahrungsfristen** fest. Ein verbreitetes Schema ist die **Großvater-Vater-Sohn-Rotation**: Tägliche Sicherungen („Söhne“) werden etwa eine Woche aufbewahrt, wöchentliche („Väter“) einige Wochen und monatliche („Großväter“) über Monate oder Jahre. So stehen sowohl aktuelle als auch ältere Stände zur Verfügung, ohne dass jede einzelne Sicherung ewig gespeichert werden muss.

Beachten Sie zudem **gesetzliche Vorgaben**: Handels- und steuerrechtlich relevante Unterlagen müssen je nach Art mehrere Jahre aufbewahrt werden, während der Datenschutz verlangt, personenbezogene Daten nicht länger als nötig zu speichern. Ältere Sicherungen werden üblicherweise auf günstigere Medien ausgelagert.

Der Notfallplan

Backups sind nur die halbe Miete. Im Ernstfall muss klar sein, wer was in welcher Reihenfolge wiederherstellt – erst Netzwerk und Active Directory, dann die Datenbanken, dann die Anwendungen. Diese Abläufe gehören in **Runbooks**, also schriftliche Schritt-

für-Schritt-Anleitungen. Bereiten Sie einen **Ausweichstandort** oder eine Wiederherstellung in der Cloud vor, falls die eigene Hardware nicht mehr verfügbar ist.

Aus der Praxis

Üben Sie den Ernstfall regelmäßig. Eine Notfallübung zeigt, ob die Runbooks stimmen, ob RTO und RPO eingehalten werden und ob alle Beteiligten ihre Aufgaben kennen – Erkenntnisse, die man lieber in der Übung als im echten Ausfall gewinnt.

Das Wichtigste in Kürze

- Die 3-2-1-1-0-Regel ist die Leitlinie für robuste Sicherungen.
- Voll, differenziell, inkrementell und synthetisch bewusst kombinieren.
- RPO und RTO gemeinsam mit dem Business festlegen.
- Eine Kopie offline oder unveränderlich gegen Ransomware.
- Restore und Notfallablauf regelmäßig testen – sonst ist es kein Backup.

VSS, Snapshots & agentenloses Backup

Konsistente Sicherungen im laufenden Betrieb

Nach diesem Kapitel können Sie …

- das Konsistenzproblem bei Sicherungen im laufenden Betrieb erklären,
- die Funktionsweise von VSS mit Writern und Providern beschreiben,
- crash-konsistente und app-konsistente Sicherungen unterscheiden,
- VM-Snapshots, Changed Block Tracking und agentenloses Backup einordnen und deren Grenzen benennen.

Das Konsistenzproblem

Server laufen rund um die Uhr. Während eine Sicherung läuft, werden Dateien geöffnet, geändert und gespeichert, und Datenbanken schreiben ständig Transaktionen. Kopiert man einfach Datei für Datei, entsteht ein Abbild, das an verschiedenen Stellen unterschiedliche Zeitpunkte erfasst: Die erste Datei stammt von 22:00 Uhr, die letzte von 22:40 Uhr, und eine Datenbankdatei wurde vielleicht mitten in einem Schreibvorgang kopiert.

Für einfache Dokumente mag das noch gut gehen. Anwendungen wie Datenbanken oder Mailserver brauchen jedoch einen **sauberen, in sich stimmigen Stand**, bei dem alle zusammengehörigen Daten vom selben Moment stammen und keine Transaktion halb geschrieben ist. Fehlt diese Konsistenz, droht im schlimmsten Fall ein Backup, das sich zwar wiederherstellen, aber nicht mehr benutzen lässt.

VSS unter Windows

Windows löst dieses Problem mit dem **Volume Shadow Copy Service** (VSS), dem Volumeschattenkopie-Dienst. Er erzeugt im laufenden Betrieb eine konsistente Momentaufnahme eines Laufwerks, aus der die Backup-Software anschließend in Ruhe liest, während die Anwendungen normal weiterarbeiten.

Dabei arbeiten drei Beteiligte zusammen. Der **Requester** – meist die Backup-Software – fordert die Schattenkopie an. Die **VSS-Writer** sind Komponenten der Anwendungen, etwa von SQL Server, Exchange oder dem Active Directory. Sie bringen ihre Anwendung kurz in einen sicheren Zustand, schreiben offene Daten auf die Platte und halten Schreibvorgänge für einen Moment an. Dann erstellt der **VSS-Provider** – das Betriebssystem selbst oder ein Speichersystem – blitzschnell den Snapshot, und die Anwendungen laufen weiter. Der ganze Vorgang dauert meist nur Sekunden.

VSS ist die Grundlage für praktisch alle Windows-Backups und auch für die „Vorgängerversionen“, mit denen Benutzer im Explorer ältere Stände einer Datei wiederherstellen können. Den Zustand der Writer prüfen Sie bei Problemen an der Kommandozeile:

```
vssadmin list writers
vssadmin list shadows
```

Steht ein Writer dort nicht auf „Stabil“ beziehungsweise „Stable“ oder meldet einen Fehler, ist das oft die Ursache für fehlschlagende Sicherungen.

Crash-konsistent oder app-konsistent?

Bei Sicherungen unterscheidet man zwei Qualitätsstufen. Eine **crash-konsistente** Sicherung entspricht dem Zustand nach einem plötzlichen Stromausfall: Alle Daten auf der Platte stammen vom selben Moment, aber was eine Anwendung noch im Arbeitsspeicher hatte, fehlt. Moderne Datenbanken können sich davon meist selbst erholen, garantiert ist das aber nicht.

Eine **app-konsistente** Sicherung geht einen Schritt weiter: Die Anwendungen werden vor dem Snapshot gezielt in einen sauberen Zustand gebracht – man spricht vom *Quiescing*. Datenbanken sind dann ordnungsgemäß abgeschlossen, und die Wiederherstellung ist ohne Reparaturlauf möglich. Unter Windows sorgen die VSS-Writer dafür, unter Linux oder bei Anwendungen ohne Writer übernehmen das häufig **Pre- und Post-Skripte**, die eine Anwendung vor dem Snapshot anhalten und danach wieder freigeben.

> **Merke**
>
> Crash-konsistent = wie nach einem Stromausfall. App-konsistent = Anwendungen wurden vorher sauber in einen sicheren Zustand gebracht. Für Datenbanken und Mailserver ist app-konsistent der Standard.

Sicherung virtueller Maschinen

Bei virtuellen Maschinen wird meist nicht innerhalb der VM, sondern auf Ebene des **Hypervisors** gesichert. Der Hypervisor erstellt einen Snapshot der virtuellen Festplatte, die Backup-Software liest die Daten aus diesem eingefrorenen Stand, und der Snapshot wird anschließend wieder entfernt.

Bei VMware stellt dafür die Schnittstelle **VADP** (vSphere APIs for Data Protection) die nötigen Funktionen bereit. Besonders wichtig ist dabei das **Changed Block Tracking** (CBT): Der Hypervisor merkt sich, welche Blöcke einer virtuellen Festplatte sich seit der letzten Sicherung geändert haben. Statt die ganze Platte erneut zu lesen, holt die

Backup-Software nur diese geänderten Blöcke. Hyper-V bietet mit seiner eigenen Änderungsverfolgung (Resilient Change Tracking) ein vergleichbares Verfahren. Das Ergebnis sind sehr schnelle inkrementelle Sicherungen, selbst bei großen VMs.

Agentenloses Backup

Klassisch wurde auf jedem Server ein **Backup-Agent** installiert, der die Daten von innen sichert. Bei Dutzenden oder Hunderten VMs bedeutet das viel Pflegeaufwand: Jeder Agent muss installiert, aktualisiert und überwacht werden. Beim **agentenlosen Backup** greift die Backup-Software stattdessen über den Hypervisor auf die VMs zu und sichert jede VM als Ganzes – mit Betriebssystem, Anwendungen und Daten.

Das spart Aufwand und erfasst automatisch auch neu angelegte VMs. Damit die Sicherung trotzdem app-konsistent ist, arbeiten viele Lösungen **application-aware**: Sie lösen über die Gastintegration des Hypervisors die VSS-Writer innerhalb der VM aus, bevor der Snapshot erstellt wird. So sind auch Datenbanken in der VM sauber gesichert.

Der Lebenszyklus eines Snapshots

Ein VM-Snapshot ist kein Abbild, das einfach zur Seite gelegt wird. Ab dem Moment der Erstellung werden alle Änderungen in einer separaten **Delta-Datei** gespeichert, während die ursprüngliche Platte eingefroren bleibt. Diese Delta-Datei wächst mit jeder Änderung, und jeder Lesezugriff muss beide Dateien berücksichtigen – das kostet Leistung. Beim Entfernen des Snapshots müssen die Änderungen zurück in die Originalplatte geschrieben werden, was bei großen Delta-Dateien lange dauern kann.

Snapshots sollten deshalb nach dem Backup **automatisch wieder entfernt** und nie als dauerhafter Zustand offen gelassen werden. Überwachen Sie den Speicherplatz für Snapshots und prüfen Sie regelmäßig, ob verwaiste Snapshots vergessen wurden – etwa nach einem abgebrochenen Backup-Lauf.

Wichtige Warnung

Ein Snapshot fühlt sich an wie ein Backup, ist aber keines. Er liegt auf demselben Speicher wie das Original und ist ohne dieses wertlos. Fällt das Speichersystem aus oder verschlüsselt Ransomware die Datastores, sind Original und Snapshot gleichermaßen verloren. Snapshots sind nur kurzfristige Zwischenstände und müssen immer in ein echtes, ausgelagertes Backup überführt werden.

Das Wichtigste in Kürze

- Sicherungen im laufenden Betrieb brauchen einen konsistenten Stand.
- VSS sorgt mit Writern und Providern für konsistente Windows-Sicherungen.

- App-konsistent ist für Datenbanken der Standard, crash-konsistent nur die Mindestqualität.
- Changed Block Tracking sichert nur geänderte Blöcke, agentenlos spart Aufwand.
- Ein Snapshot ist kein Backup und darf nicht dauerhaft offen bleiben.

Datenbank-Backup

Datenbanken korrekt sichern und wiederherstellen

Nach diesem Kapitel können Sie …

- begründen, warum Datenbanken eigene Sicherungsverfahren brauchen,
- Voll-, Differenz- und Transaktionslog-Sicherung bei SQL Server sowie die Recovery-Modelle erklären,
- die Sicherungswerkzeuge von MySQL/MariaDB und PostgreSQL einordnen,
- Point-in-Time-Recovery, heiße Backups und Restore-Tests planen.

Warum Datenbanken besonders sind

Eine Datenbank ist ständig in Bewegung. Während Sie sie sichern, werden laufend Datensätze eingefügt, geändert und gelöscht, oft als Teil von **Transaktionen**: zusammengehörigen Änderungen, die entweder vollständig oder gar nicht ausgeführt werden müssen. Eine Überweisung etwa besteht aus einer Abbuchung und einer Gutschrift – nur eines von beiden darf niemals gespeichert sein.

Eine einfache Kopie der Datenbankdateien im laufenden Betrieb kann Transaktionen mittendrin erwischen und ist dann inkonsistent. Deshalb nutzt man die **eigenen Backup-Mechanismen** der Datenbanksysteme oder Backup-Software, die diese Mechanismen gezielt anspricht. Ziel ist dabei mehr als nur eine Kopie: Man möchte eine **Point-in-Time-Wiederherstellung** ermöglichen, also die Datenbank auf einen beliebigen Zeitpunkt zurücksetzen können – zum Beispiel auf 14:31 Uhr, eine Minute bevor jemand versehentlich eine Tabelle gelöscht hat.

SQL Server: drei Bausteine

Bei Microsoft SQL Server setzt sich eine Sicherungsstrategie aus drei Arten zusammen. Die **Vollsicherung** (Full) sichert die gesamte Datenbank. Die **differenzielle Sicherung** (Differential) enthält alle Änderungen seit der letzten Vollsicherung. Die **Transaktionslog-Sicherung** schließlich sichert das Protokoll aller Transaktionen seit der letzten Log-Sicherung.

Erst die Log-Sicherung ermöglicht die Wiederherstellung auf einen beliebigen Zeitpunkt – im Prinzip auf die Sekunde genau. Sie hat außerdem einen wichtigen Nebeneffekt: Nach einer erfolgreichen Log-Sicherung kann SQL Server den gesicherten Teil des Logs wiederverwenden, und die Logdatei bleibt klein. Eine typische Kombination ist eine wöchentliche Vollsicherung, tägliche Differenzsicherungen und Log-Sicherungen alle 15 bis 60 Minuten.

```
BACKUP DATABASE Warenwirtschaft TO DISK = 'E:\Backup\ww_full.bak' WITH CHECKSUM;
BACKUP LOG Warenwirtschaft TO DISK = 'E:\Backup\ww_log.trn' WITH CHECKSUM;
```

Recovery-Modelle

Welche Sicherungen überhaupt möglich sind, bestimmt das **Recovery-Modell** einer Datenbank. Im Modell **Simple** wird das Transaktionslog automatisch geleert; Log-Sicherungen sind nicht möglich. Es gibt nur Voll- und Differenzsicherungen, und wiederhergestellt werden kann nur der Stand einer dieser Sicherungen – alles danach ist verloren. Im Modell **Full** bleibt das Log erhalten, bis es gesichert wird. Damit sind Log-Sicherungen und Point-in-Time-Wiederherstellung möglich.

Recovery-Modell	Log-Sicherung	Point-in-Time	Typischer Einsatz
Simple	nicht möglich	nein	Test- und Entwicklungsdatenbanken, unkritische Daten
Full	möglich und nötig	ja	Produktive Datenbanken mit wichtigen Daten

> **Merke**
>
> Im Recovery-Modell Full muss das Transaktionslog regelmäßig gesichert werden. Sonst wächst die Logdatei immer weiter, bis im schlimmsten Fall das Laufwerk vollläuft und die Datenbank stehen bleibt.

MySQL und MariaDB

Für MySQL und MariaDB gibt es zwei grundlegende Ansätze. Ein **logisches Backup** mit `mysqldump` exportiert die Datenbank als Folge von SQL-Befehlen, mit denen sich Tabellen und Daten wieder aufbauen lassen. Das ist einfach, gut lesbar und unabhängig von der Version, bei großen Datenbanken aber langsam – vor allem bei der Wiederherstellung.

```
mysqldump --single-transaction --routines --all-databases > komplett.sql
```

Die Option `--single-transaction` sorgt bei Tabellen mit der Speicher-Engine InnoDB dafür, dass ein konsistenter Stand gesichert wird, ohne die Datenbank zu sperren. Für große Datenbanken eignen sich **physische Backups**, etwa mit **Percona XtraBackup** (bzw. dem verwandten Mariabackup bei MariaDB). Diese Werkzeuge kopieren die Daten-

dateien im laufenden Betrieb – man spricht vom *heißen Backup* – und sorgen selbst für die Konsistenz. Für die Point-in-Time-Wiederherstellung werden zusätzlich die **Binärlogs** (Binlogs) gesichert, in denen alle Änderungen protokolliert sind.

PostgreSQL

Auch PostgreSQL kennt beide Wege. Mit `pg_dump` wird eine einzelne Datenbank logisch gesichert, `pg_dumpall` sichert alle Datenbanken eines Servers samt Benutzern und Rollen. Für die physische Sicherung erstellt man ein **Basis-Backup**, zum Beispiel mit `pg_basebackup`, und archiviert zusätzlich fortlaufend das **WAL** (Write-Ahead Log). Das WAL enthält alle Änderungen in der Reihenfolge, in der sie geschehen sind.

Bei einer Wiederherstellung wird zuerst das Basis-Backup zurückgespielt, dann spielt PostgreSQL die archivierten WAL-Dateien bis zu einem gewünschten Zeitpunkt nach. So entsteht die **Point-in-Time-Recovery** – das Gegenstück zur Log-Sicherung bei SQL Server und den Binlogs bei MySQL.

Heiße Backups und Betrieb

Produktive Datenbanken lassen sich selten für eine Sicherung anhalten. Alle genannten Systeme unterstützen deshalb **heiße Backups** im laufenden Betrieb ohne langes Sperren. Die Konsistenz wird dabei über die datenbankeigenen Mechanismen sichergestellt, nicht über ein bloßes Kopieren von Dateien. Trotzdem erzeugen Sicherungen Last auf Prozessor, Datenträgern und Netzwerk. Behalten Sie die **Auswirkungen auf die Leistung** im Blick und legen Sie aufwendige Operationen – etwa eine Vollsicherung großer Datenbanken oder Integritätsprüfungen – in Wartungsfenster außerhalb der Hauptarbeitszeit.

Aufbewahrung und Restore-Test

Definieren Sie **Aufbewahrungsketten**, zum Beispiel tägliche Sicherungen für zwei Wochen, wöchentliche für drei Monate und monatliche für ein Jahr, und beachten Sie dabei Compliance-Vorgaben. Bewahren Sie die Backups **außerhalb** des Datenbankservers auf, ausgelagert und vor Manipulation geschützt – ein Backup auf derselben Platte wie die Datenbank geht mit dieser verloren. Überwachen Sie, ob jede Sicherung erfolgreich gelaufen ist, und lassen Sie sich bei Fehlern alarmieren.

Die goldene Regel gilt bei Datenbanken ganz besonders: Ohne getesteten Restore ist ein Backup nur eine Hoffnung. Ein Datenbank-Backup zählt erst, wenn es nachweislich in einer **Testinstanz** wiederhergestellt wurde und die Anwendung mit den Daten arbeiten kann.

Aus der Praxis

Üben Sie die Wiederherstellung regelmäßig und dokumentieren Sie dabei, wie lange sie gedauert hat. So wissen Sie im Ernstfall, ob die vereinbarte Wiederherstellungszeit realistisch ist, und haben eine erprobte Anleitung zur Hand.

Das Wichtigste in Kürze

- Datenbankeigene Mechanismen statt einfacher Dateikopie nutzen.
- Transaktionslog, Binlog bzw. WAL ermöglichen Point-in-Time-Recovery.
- Bei SQL Server bestimmt das Recovery-Modell, was möglich ist.
- Je System eigene Werkzeuge: Backup-Befehle, mysqldump/XtraBackup, pg_dump/WAL.
- Restore in einer Testinstanz regelmäßig testen ist Pflicht.

Backup-Software & Hersteller

Werkzeuge für die Datensicherung

Nach diesem Kapitel können Sie …

- begründen, warum Unternehmen dedizierte Backup-Software statt reiner Bordmittel einsetzen,
- die wichtigsten Funktionen einer Backup-Lösung benennen und erklären,
- Cloud-Backup sinnvoll in ein Sicherungskonzept einordnen,
- Auswahlkriterien für eine Backup-Lösung anwenden,
- Maßnahmen zum Schutz der Backups vor Ransomware beschreiben.

Warum dedizierte Backup-Software?

Betriebssysteme bringen eigene Werkzeuge zur Datensicherung mit, und für einen einzelnen Rechner reichen sie oft aus. In einem Unternehmen mit mehreren Servern, virtuellen Maschinen, Datenbanken und Cloud-Diensten stoßen diese Bordmittel jedoch schnell an Grenzen. Hier kommt **dedizierte Backup-Software** ins Spiel: Sie steuert alle Sicherungen von einer zentralen Konsole aus, führt sie nach festen Zeitplänen automatisch durch und überwacht, ob jeder Job erfolgreich war.

Ein entscheidender Vorteil sind *application-aware Backups*. Die Software erkennt, dass auf einem Server etwa eine Datenbank oder ein Mailserver läuft, und sorgt dafür, dass die Anwendung vor der Sicherung in einen konsistenten Zustand gebracht wird – zum Beispiel über die Schattenkopie-Mechanismen, die Sie in den vorherigen Kapiteln kennengelernt haben. Dazu kommen **Berichte und Alarme**: Schlägt eine Sicherung fehl, erfährt die IT davon sofort und nicht erst, wenn eine Wiederherstellung nötig wird. Verschlüsselung und Deduplizierung gehören ebenfalls zum Standardumfang. Und nicht zuletzt beschleunigt eine gute Software den **Restore** erheblich, weil sie gezielt einzelne Objekte oder ganze Systeme zurückholen kann.

Hersteller im Überblick

Der Markt für Backup-Software ist groß, und Sie sollten die bekannten Anbieter zumindest grob einordnen können. Als Beispiele begegnen Ihnen in der Praxis etwa **Veeam** und **Nakivo**, deren Schwerpunkt traditionell auf virtualisierten und modernen Umgebungen liegt. Anbieter wie **Acronis** oder **Barracuda** verfolgen häufig einen breiteren Ansatz und kombinieren die Datensicherung mit weiteren Sicherheitsfunktionen, oft unter dem Begriff „Cyber-Protection“ zusammengefasst. Daneben gibt es zahlreiche weitere Lösungen, auch aus dem Open-Source-Bereich.

Eine pauschal „beste“ Lösung gibt es nicht. Welches Produkt passt, hängt von der vorhandenen Umgebung, den Anforderungen und dem verfügbaren Budget ab. Wichtig ist, dass Sie die Produkte anhand nachvollziehbarer Kriterien vergleichen, statt sich nur nach Bekanntheit zu richten. Prüfen Sie dabei besonders, ob unveränderliche Sicherungen und Kopien an einen externen Standort unterstützt werden.

Cloud-Backup als externe Kopie

Neben klassischer Backup-Software spielen Cloud-Dienste eine wachsende Rolle. Die großen Cloud-Plattformen bieten eigene Sicherungsdienste an, zum Beispiel **Azure Backup** oder **AWS Backup**. Viele Backup-Programme können ihre Daten zudem direkt in einen Cloud-Objektspeicher schreiben.

Im Sicherungskonzept eignet sich die Cloud vor allem als *externe Kopie*. In der 3-2-1-1-0-Regel steht die erste „1“ für mindestens eine Kopie außerhalb des eigenen Standorts – genau diese Rolle kann ein Cloud-Speicher übernehmen. Brennt der Serverraum ab oder wird das lokale Netz verschlüsselt, liegen die Daten noch an einem anderen Ort.

Zwei Punkte sollten Sie dabei bedenken. Erstens fallen laufende Kosten an, und zwar nicht nur für den belegten Speicherplatz, sondern je nach Anbieter auch für Zugriffe und das Herunterladen der Daten. Zweitens dauert eine Wiederherstellung aus der Cloud spürbar länger als von einem lokalen Speicher, weil alle Daten erst über die Internetleitung zurückkommen müssen. Diese Wiederherstellzeiten gehören deshalb in die Notfallplanung.

> **Merke**
>
> Cloud-Backup ist ideal als externe Kopie im 3-2-1-1-0-Konzept. Planen Sie aber die Kosten für Speicher und Zugriff sowie die längeren Wiederherstellzeiten von Anfang an ein.

Wichtige Funktionen

Unabhängig vom Hersteller gibt es einige Funktionen, die eine professionelle Backup-Lösung beherrschen sollte. **Deduplizierung und Kompression** sparen Speicherplatz: Bei der Deduplizierung werden identische Datenblöcke nur einmal gespeichert, auch wenn sie in vielen Sicherungen vorkommen – etwa die Systemdateien zahlreicher gleichartiger virtueller Maschinen. Die Kompression verkleinert die Daten zusätzlich.

Unveränderliche (immutable) Backups sind Sicherungen, die für einen festgelegten Zeitraum weder gelöscht noch verändert werden können – auch nicht von einem Administrator. Sie sind einer der wirksamsten Schutzmechanismen gegen Ransomware.

Ebenso unverzichtbar ist die **Verschlüsselung**, und zwar sowohl bei der Übertragung als auch im gespeicherten Zustand („in Ruhe“). So bleiben die Daten geschützt, selbst wenn ein Sicherungsmedium verloren geht.

Im Alltag besonders wertvoll ist der **granulare Restore**. Damit lassen sich einzelne Dateien, Ordner oder E-Mails gezielt aus einer Sicherung zurückholen, ohne gleich einen kompletten Server wiederherzustellen. Löscht jemand versehentlich eine wichtige Präsentation, ist sie so in wenigen Minuten wieder da.

Auswahlkriterien

Wenn Sie eine Backup-Lösung auswählen oder bewerten, hilft ein strukturiertes Vorgehen. An erster Stelle steht die Frage, welche **Workloads** gesichert werden müssen: virtuelle Maschinen unter VMware oder Hyper-V, physische Server, Microsoft-365-Daten, Datenbanken oder Cloud-Dienste. Eine Lösung, die einen wichtigen Teil der Umgebung nicht abdeckt, scheidet aus oder erzwingt ein zweites Produkt.

Danach folgen **Lizenzmodell und Skalierbarkeit**: Wird pro Server, pro Workload oder nach Datenmenge lizenziert, und wächst die Lösung mit, wenn das Unternehmen größer wird? Ebenso wichtig sind die bereits genannten **Immutable- und Offsite-Optionen**. Schließlich zählen der **Herstellersupport und der Reifegrad** des Produkts – im Ernstfall brauchen Sie schnelle und kompetente Hilfe.

Kriterium	Leitfrage
Unterstützte Workloads	Werden alle Systeme der Umgebung abgedeckt?
Lizenzmodell und Skalierbarkeit	Passt die Abrechnung zur Umgebung, wächst die Lösung mit?
Immutable und Offsite	Lassen sich unveränderliche und externe Kopien erstellen?
Support und Reifegrad	Ist das Produkt erprobt, gibt es verlässliche Hilfe?

Ransomware-Resilienz

Moderne Ransomware-Angriffe verschlüsseln nicht nur die produktiven Daten. Angreifer suchen gezielt nach den Backups und versuchen, diese zu löschen oder ebenfalls zu verschlüsseln – denn ohne funktionierende Sicherung steigt der Druck, Lösegeld zu zahlen. Backups sind damit eines der Hauptziele eines Angriffs und müssen besonders geschützt werden.

Die wichtigsten Bausteine sind **unveränderliche Repositories**, in denen Sicherungen nicht vorzeitig gelöscht werden können, sowie ein **Air-Gap** oder zumindest getrennte Netzsegmente, damit die Backup-Infrastruktur vom übrigen Netz aus nicht direkt erreichbar ist. Ebenso wichtig sind **eigene Zugangsdaten** für das Backup-System: Ver-

wenden Sie dort nicht die Konten der Domänen-Administratoren. Wird die Domäne kompromittiert, bleibt das Backup so trotzdem geschützt. Die Backup-Konsole sollte zusätzlich mit einer **Multi-Faktor-Authentifizierung** abgesichert sein.

Aus der Praxis

Binden Sie den Backup-Server nicht in die Windows-Domäne ein oder nutzen Sie dafür zumindest getrennte, ausschließlich für das Backup vorgesehene Konten. Erbeuten Angreifer ein Domänen-Admin-Konto, haben sie sonst gleichzeitig Zugriff auf Produktivsysteme und Sicherungen.

Restore-Tests und laufender Betrieb

Ein Backup ist erst dann eines, wenn der Restore nachweislich funktioniert. Dieser Grundsatz ist der vielleicht wichtigste des gesamten Lernbereichs. Viele Unternehmen merken erst im Ernstfall, dass Sicherungen unvollständig, beschädigt oder nicht wiederherstellbar sind. Planen Sie deshalb **regelmäßige, dokumentierte Restore-Tests** ein – nicht nur für einzelne Dateien, sondern auch für die komplette Notfall-Wiederherstellung ganzer Systeme. Messen Sie dabei die tatsächliche Wiederanlaufzeit und vergleichen Sie sie mit dem vereinbarten Ziel, der *Recovery Time Objective* (RTO).

Auch der laufende Betrieb braucht feste Abläufe. Backup-Jobs müssen überwacht werden, und bei Fehlern muss ein Alarm ausgelöst werden, auf den jemand reagiert. Die Speicherkapazität sollte vorausschauend geplant werden, wobei Deduplizierung und Kompression den tatsächlichen Bedarf beeinflussen. Und ein oft vergessener Punkt: Sichern Sie auch den Backup-Server selbst und seine Konfiguration. Fällt er aus, wollen Sie ihn schnell wieder aufbauen können, um an die gesicherten Daten zu gelangen.

> **Merke**
>
> Restore-Tests sind nicht verhandelbar. Nur eine regelmäßig geprüfte und dokumentierte Wiederherstellung beweist, dass das Backup im Ernstfall wirklich hilft.

Das Wichtigste in Kürze

- Backup-Software zentralisiert, automatisiert und überwacht die Datensicherung.
- Hersteller wie Veeam, Nakivo, Acronis oder Barracuda werden nach Umgebung, Anforderungen und Budget ausgewählt.
- Cloud-Backup eignet sich als externe Kopie – Kosten und Wiederherstellzeiten einplanen.
- Unveränderliche und externe Kopien, getrennte Zugangsdaten und MFA schützen vor Ransomware.

- Regelmäßige, dokumentierte Restore-Tests sind Pflicht.

LB-07

IT-Security

1. Schutzziele & Grundlagen
2. Malware & Angriffstechniken
3. Antivirus & NGAV
4. EDR & XDR
5. Firewall & Netzwerksicherheit
6. Mail-Security & Anti-Spam
7. SSL/TLS & PKI
8. Identity & Access Management
9. SIEM & SOC-Grundlagen
10. BSI IT-Grundschutz & Compliance

Schutzziele & Grundlagen

CIA-Triade, Risiko und Defense in Depth

Nach diesem Kapitel können Sie …

- die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit erklären und um Authentizität und Nichtabstreitbarkeit ergänzen,
- beschreiben, wie ein Risiko entsteht und welche Möglichkeiten es gibt, damit umzugehen,
- die Prinzipien „minimale Rechte“ und „minimale Angriffsfläche“ anwenden,
- begründen, warum Sicherheit immer aus mehreren Schichten bestehen muss.

Warum es Schutzziele braucht

IT-Sicherheit wirkt auf den ersten Blick wie eine lose Sammlung von Werkzeugen: Firewall, Virenschutz, Passwortregeln, Backups. Damit daraus ein schlüssiges Gesamtbild wird, braucht es einen gemeinsamen Maßstab. Genau diesen Maßstab liefern die **Schutzziele**. Sie beschreiben, *was* eigentlich geschützt werden soll – und damit auch, woran sich jede einzelne Maßnahme messen lassen muss.

Wer die Schutzziele verstanden hat, besitzt den roten Faden für alle weiteren Sicherheitsthemen. Ob es um E-Mail-Filter, Verschlüsselung oder Zugriffsrechte geht: Am Ende dient jede Technik dazu, eines oder mehrere dieser Ziele zu erreichen.

Die CIA-Triade

Das klassische Modell ist die sogenannte **CIA-Triade**, benannt nach den englischen Anfangsbuchstaben der drei Grundziele. *Confidentiality*, auf Deutsch **Vertraulichkeit**, bedeutet: Nur Befugte dürfen Daten sehen. Eine Gehaltsliste soll nur die Personalabteilung lesen können, nicht jede Person im Unternehmen. Typische Maßnahmen sind Zugriffsrechte und Verschlüsselung.

Integrity, also **Integrität**, heißt: Daten werden nicht unbemerkt verändert. Es geht dabei nicht nur um absichtliche Manipulation, etwa eine geänderte Kontonummer in einer Rechnung, sondern auch um technische Fehler. Prüfsummen, digitale Signaturen und eine saubere Rechtevergabe helfen, Veränderungen zu verhindern oder zumindest zu erkennen.

Availability, die **Verfügbarkeit**, bedeutet: Systeme und Daten stehen zur Verfügung, wenn man sie braucht. Ein perfekt geschützter Server nützt wenig, wenn er nach einem Hardwaredefekt tagelang ausfällt. Redundanz, Backups und Notfallpläne sichern dieses

Ziel ab.

Diese drei Ziele bilden das Grundgerüst jeder Sicherheitsbetrachtung. Wenn Sie ein System bewerten, lohnt sich immer die Frage: Was passiert, wenn hier die Vertraulichkeit, die Integrität oder die Verfügbarkeit verletzt wird?

Ergänzende Ziele: Authentizität und Nichtabstreitbarkeit

Häufig wird die CIA-Triade um die **Authentizität** ergänzt. Sie beschreibt die Sicherheit, dass jemand oder etwas wirklich das ist, was es vorgibt zu sein. Eine E-Mail, die angeblich von der Geschäftsführung stammt, ist nur dann vertrauenswürdig, wenn ihre Herkunft belegt ist. Ebenso muss sich ein Server gegenüber dem Browser ausweisen können. Authentizität ist damit die Basis für Vertrauen und Nachweisbarkeit.

Eng damit verwandt ist die **Nichtabstreitbarkeit**. Sie stellt sicher, dass sich eine Handlung nachweislich einer bestimmten Person zuordnen lässt und diese sie später nicht leugnen kann. Das ist besonders wichtig bei Verträgen, Überweisungen oder administrativen Eingriffen in Systeme. Umgesetzt wird Nichtabstreitbarkeit vor allem über manipulationssichere Protokolle (Logs) und digitale Signaturen. Sie ergänzt die drei klassischen Ziele, ersetzt sie aber nicht.

> **Merke**
>
> Die drei Grundziele sind Vertraulichkeit, Integrität und Verfügbarkeit (CIA). Authentizität und Nichtabstreitbarkeit ergänzen sie, wenn es darum geht, Herkunft und Urheberschaft eindeutig nachzuweisen.

Zielkonflikte und Schutzbedarf

Die Schutzziele stehen nicht immer im Einklang. Sicherheit ist deshalb stets ein Abwägen. Maximale Vertraulichkeit kann etwa die Verfügbarkeit einschränken: Werden Daten so stark verschlüsselt und abgeschottet, dass nur eine einzige Person den Schlüssel hat, sind sie bei deren Abwesenheit nicht mehr nutzbar. Umgekehrt erhöht jede zusätzliche Kopie für eine bessere Verfügbarkeit das Risiko, dass Unbefugte an die Daten gelangen. Kein Ziel „gewinnt“ absolut – die richtige Balance hängt vom Schutzbedarf ab.

Nicht alles ist gleich schützenswert. Der Aufwand für Sicherheitsmaßnahmen richtet sich nach dem Schaden, der bei einer Verletzung der Schutzziele entstehen würde. In der Praxis haben sich drei Stufen bewährt: Beim Schutzbedarf **normal** ist der mögliche Schaden begrenzt und überschaubar, Standardmaßnahmen genügen. Beim Schutzbedarf **hoch** kann der Schaden beträchtlich sein, bei **sehr hoch** sogar existenzbedrohend. In diesen Fällen sind zusätzliche, strengere Maßnahmen nötig. Diese Einteilung ist auch die Grundlage für die Priorisierung im BSI IT-Grundschutz, der im letzten Kapitel dieses Lernbereichs ausführlich behandelt wird.

Schutzbedarf	Mögliche Auswirkung	Konsequenz
normal	Begrenzter, überschaubarer Schaden	Standardmaßnahmen genügen
hoch	Beträchtlicher Schaden	Zusätzliche Maßnahmen prüfen
sehr hoch	Existenzbedrohender Schaden	Strenge, oft individuelle Maßnahmen

Risiko und Risikomanagement

Wovor schützt man sich eigentlich? Die einfache Antwort lautet: vor Bedrohungen, die auf Schwachstellen treffen. Eine *Bedrohung* ist zum Beispiel ein Angreifer, ein Brand oder ein Stromausfall. Eine *Schwachstelle* ist eine Lücke, die diese Bedrohung ausnutzen kann – etwa eine fehlende Sicherheitsaktualisierung oder ein Serverraum ohne Brandschutz. Erst wenn beides zusammenkommt, entsteht ein **Risiko**. Bewertet wird es in der Regel danach, wie wahrscheinlich ein Schadensfall ist und wie groß der Schaden wäre.

Das **Risikomanagement** sorgt dafür, dass Risiken systematisch erkannt, bewertet und behandelt werden. Für den Umgang mit einem Risiko gibt es vier grundsätzliche Optionen. Man kann es *vermeiden*, indem man auf eine riskante Tätigkeit ganz verzichtet. Man kann es *vermindern*, indem man Schutzmaßnahmen umsetzt. Man kann es *übertragen*, beispielsweise durch eine Versicherung oder die Auslagerung an einen Dienstleister. Oder man *akzeptiert* es bewusst, wenn der Aufwand für weitere Maßnahmen in keinem Verhältnis zum möglichen Schaden steht. Wichtig ist dabei: Nicht jedes Risiko lässt sich wirtschaftlich sinnvoll ausschließen. Entscheidend ist, dass verbleibende Risiken bewusst getragen und nicht einfach übersehen werden.

Angriffsfläche und minimale Rechte

Ein zentraler Begriff der IT-Sicherheit ist die **Angriffsfläche**. Gemeint ist die Summe aller Stellen, an denen ein Angreifer ansetzen könnte. Jeder offene Port, jeder unnötig laufende Dienst und jedes zu weit gefasste Zugriffsrecht vergrößert sie. Ein Server, auf dem nur die wirklich benötigten Dienste laufen, bietet schlicht weniger Ansatzpunkte als einer mit vielen zusätzlichen, vergessenen Programmen.

Daraus ergeben sich zwei Grundprinzipien, die sich durch alle weiteren Kapitel ziehen. Das Prinzip der **minimalen Angriffsfläche** verlangt, nur das zu betreiben und zu öffnen, was tatsächlich gebraucht wird. Das Prinzip der **minimalen Rechte** (Least Privilege) verlangt, dass Personen und Programme nur genau die Berechtigungen erhalten, die sie für ihre Aufgabe benötigen. Wird ein Konto mit wenigen Rechten kompromittiert, bleibt der Schaden begrenzt.

Aus der Praxis

Arbeiten Sie im Alltag nicht mit einem Administratorkonto. Für E-Mail, Internet und Office genügt ein normales Benutzerkonto; administrative Rechte werden nur für die jeweilige Aufgabe genutzt. Schon diese einfache Trennung verhindert, dass ein versehentlich geöffneter Schadanhang sofort das ganze System übernehmen kann.

Defense in Depth: Sicherheit in Schichten

Keine einzelne Sicherheitsmaßnahme ist perfekt. Ein Virenschutz übersieht neue Schadsoftware, eine Firewall lässt erlaubten, aber manipulierten Verkehr durch, und auch gut geschulte Mitarbeitende klicken gelegentlich auf einen gefälschten Link. Deshalb setzt man auf **Defense in Depth**, die gestaffelte Verteidigung: Mehrere Schutzschichten liegen hintereinander, sodass ein Angreifer, der eine Schicht überwindet, an der nächsten scheitert.

Diese Schichten sind nicht nur technischer Natur. Echter Schutz entsteht erst aus dem Zusammenspiel von *Technik* (etwa Firewall, Verschlüsselung, Backups), *Prozessen* (etwa Rechtevergabe, Notfallplänen, regelmäßigen Updates) und *geschulten Menschen*, die Angriffe erkennen und richtig reagieren. Fällt eine Ebene aus, fangen die anderen den Fehler idealerweise auf. Sie werden in den folgenden Kapiteln die einzelnen Schichten genauer kennenlernen.

Das Wichtigste in Kürze

- CIA: Vertraulichkeit, Integrität, Verfügbarkeit – ergänzt um Authentizität und Nichtabstreitbarkeit.
- Schutzziele können in Konflikt stehen; die Balance richtet sich nach dem Schutzbedarf.
- Risiko entsteht, wenn eine Bedrohung auf eine Schwachstelle trifft – es wird vermieden, vermindert, übertragen oder akzeptiert.
- Minimale Rechte und minimale Angriffsfläche sind Grundprinzipien.
- Sicherheit immer mehrschichtig denken: Technik, Prozesse und Menschen.

Malware & Angriffstechniken

Von Viren und Ransomware bis Social Engineering

Nach diesem Kapitel können Sie …

- die wichtigsten Arten von Schadsoftware unterscheiden und ihre Ziele benennen,
- erklären, warum Ransomware für Unternehmen besonders gefährlich ist,
- typische Formen von Social Engineering wie Phishing und CEO-Fraud erkennen,
- die Phasen eines Angriffs und sinnvolle Erstmaßnahmen im Vorfall beschreiben.

Was Malware ist

Malware ist ein Sammelbegriff für schädliche Software – abgeleitet aus dem Englischen *malicious software*. Hinter diesem Begriff verbergen sich sehr unterschiedliche Programme mit ebenso unterschiedlichen Zielen: Datendiebstahl, Erpressung, Sabotage oder Spionage. Manche Schadsoftware will möglichst schnell möglichst viel Schaden anrichten, andere bleibt bewusst unauffällig, um über lange Zeit Informationen abzuziehen.

Auf die Rechner gelangt Malware über verschiedene Wege: als Anhang oder Link in einer E-Mail, über manipulierte Webseiten, über USB-Sticks oder durch das Ausnutzen von Sicherheitslücken in Software, die nicht rechtzeitig aktualisiert wurde. Besonders häufig ist dabei nicht die Technik der Einstiegspunkt, sondern der Mensch. Ein überzeugend gestalteter Anhang, der geöffnet wird, umgeht viele technische Hürden auf einmal.

Die klassischen Typen

Ein **Virus** hängt sich an vorhandene Dateien oder Programme an und wird aktiv, sobald diese ausgeführt werden. Er braucht also einen „Wirt“ und in der Regel eine Aktion des Nutzers. Ein **Wurm** dagegen verbreitet sich selbstständig über das Netz, indem er Schwachstellen auf anderen Rechnern ausnutzt. Er benötigt keinen Nutzer, um sich weiterzuverbreiten, und kann dadurch in kurzer Zeit ganze Netzwerke befallen.

Ein **Trojaner** tarnt sich – wie das Trojanische Pferd der Sage – als nützliches Programm, etwa als vermeintliches Werkzeug oder als Rechnung im Anhang. Im Hintergrund öffnet er jedoch eine Hintertür, über die Angreifer Zugriff auf das System erhalten. Häufig lädt ein Trojaner anschließend weitere Schadsoftware nach, er dient also als Türöffner für den eigentlichen Angriff.

Spyware sammelt heimlich Daten, zum Beispiel Tastatureingaben, gespeicherte Zugangsdaten oder Bildschirminhalte, und schickt sie an die Angreifer. Ein **Rootkit** nistet sich tief im Betriebssystem ein und verschleiert seine eigene Präsenz, sodass es von gängigen Werkzeugen kaum entdeckt wird. Rootkits sind schwer zu finden und noch schwerer zuverlässig zu entfernen. Oft ist es deshalb die sicherste Lösung, ein befallenes System vollständig neu aufzusetzen.

Typ	Merkmal	Typisches Ziel
Virus	Hängt sich an Dateien, braucht Ausführung	Verbreitung, Schaden
Wurm	Verbreitet sich selbst über das Netz	Schnelle, breite Infektion
Trojaner	Tarnt sich als nützliches Programm	Hintertür, Nachladen weiterer Malware
Ransomware	Verschlüsselt Daten	Erpressung von Lösegeld
Spyware / Rootkit	Arbeitet heimlich bzw. versteckt sich	Spionage, dauerhafter Zugriff

Ransomware: die größte Bedrohung für Unternehmen

Ransomware verschlüsselt Daten auf Rechnern und Servern und fordert anschließend ein Lösegeld für die Entschlüsselung. Sie gilt seit Jahren als eine der größten Bedrohungen für Unternehmen jeder Größe, weil ein erfolgreicher Angriff den gesamten Betrieb lahmlegen kann. Häufig setzen die Täter auf *Double Extortion*, die doppelte Erpressung: Sie verschlüsseln die Daten nicht nur, sondern kopieren sie vorher heimlich und drohen zusätzlich mit ihrer Veröffentlichung. Selbst wer die Daten aus einer Sicherung wiederherstellen kann, steht dann weiter unter Druck.

Der wirksamste Schutz sind **getestete, offline gehaltene Backups**. „Offline“ bedeutet, dass die Sicherung vom Netz getrennt oder unveränderbar gespeichert ist, sodass die Ransomware sie nicht mitverschlüsseln kann. „Getestet“ bedeutet, dass die Wiederherstellung tatsächlich geprobt wurde – eine Sicherung, die sich im Ernstfall nicht zurückspielen lässt, ist wertlos. Ebenso wichtig ist die **Segmentierung** des Netzes: In einem flachen Netz, in dem jeder Rechner jeden erreicht, breitet sich Ransomware rasend schnell aus. Getrennte Netzbereiche bremsen sie aus.

> **Merke**
>
> Gegen Ransomware helfen vor allem Backups, die vom Netz getrennt und regelmäßig auf ihre Wiederherstellbarkeit geprüft sind – kombiniert mit einer Segmentierung, die die Ausbreitung begrenzt.

Gezielte Angriffe: APT

Neben breit gestreuten Angriffen gibt es **Advanced Persistent Threats**, kurz APT. Dahinter stehen meist gut ausgestattete Gruppen mit einem klaren Ziel, etwa Industriespionage. Sie gehen gezielt und geduldig vor, kombinieren mehrere Techniken und nutzen auch sogenannte *Zero-Days* – Sicherheitslücken, für die es noch keine Aktualisierung gibt. Charakteristisch ist, dass solche Angreifer oft über Monate unbemerkt im Netz bleiben und sich Schritt für Schritt Zugang zu den gewünschten Informationen verschaffen.

Social Engineering: der Angriff auf den Menschen

Viele Angriffe zielen nicht auf technische Lücken, sondern auf menschliche Eigenschaften wie Hilfsbereitschaft, Neugier, Respekt vor Autoritäten oder Zeitdruck. Man spricht von **Social Engineering**. Der häufigste Einstieg ist das **Phishing**: gefälschte E-Mails, die zum Klick auf einen Link, zum Öffnen eines Anhangs oder zur Eingabe von Zugangsdaten verleiten. Typisch sind angebliche Paketbenachrichtigungen, Sicherheitswarnungen oder Rechnungen.

Beim **Spear-Phishing** wird gezielt eine einzelne Person angegriffen. Die Täter recherchieren vorab, etwa in sozialen Netzwerken, und formulieren die Nachricht so passend, dass sie kaum von einer echten zu unterscheiden ist. Eine besonders teure Variante ist der **Business E-Mail Compromise** (BEC), auch *CEO-Fraud* genannt: Jemand gibt sich als Geschäftsführung aus und weist die Buchhaltung an, dringend und vertraulich eine Überweisung auszuführen.

Aus der Praxis

Ungewöhnliche Zahlungsanweisungen per E-Mail – vor allem mit Zeitdruck und der Bitte um Verschwiegenheit – sollten immer über einen zweiten, bekannten Kanal bestätigt werden, etwa durch einen Rückruf unter der bekannten Telefonnummer. Diese einfache Regel verhindert die meisten CEO-Fraud-Schäden.

Die Angriffskette und Erstmaßnahmen

Angriffe laufen meist in typischen Phasen ab, die man als **Kill Chain** oder Angriffskette beschreibt. Am Anfang steht der *Erstzugang*, etwa über Phishing oder eine Schwachstelle. Es folgen die *Ausführung* des Schadcodes und die *Persistenz*, also das dauerhafte Festsetzen im System, damit der Zugang auch nach einem Neustart erhalten bleibt. Danach bewegen sich die Angreifer *lateral* durch das Netz, von Rechner zu Rechner, bis sie wertvolle Systeme erreichen. Am Ende stehen der *Datenabfluss* (Exfiltration) oder die Verschlüsselung. Wer diese Phasen kennt, versteht, dass sich ein Angriff an jeder Stelle der Kette noch unterbrechen lässt.

Besteht der Verdacht auf einen Ransomware-Befall, zählt jede Minute. Betroffene Systeme sollten vom Netz getrennt werden, damit sich die Schadsoftware nicht weiter ausbreitet – aber nicht einfach ausgeschaltet, denn im Arbeitsspeicher befinden sich wertvolle Spuren für die forensische Analyse. Anschließend werden Zugangsdaten zurückgesetzt, die IT-Leitung informiert und gegebenenfalls auch Behörden eingeschaltet.

Schutz auf mehreren Ebenen

Ein einzelnes Werkzeug reicht gegen diese Vielfalt nicht aus. Wirksamer Schutz entsteht erst aus mehreren Schichten: *Awareness-Schulungen* sensibilisieren Mitarbeitende für Phishing. Regelmäßiges *Patchen* schließt Sicherheitslücken, *Multi-Faktor-Authentifizierung* und *minimale Rechte* begrenzen den Schaden gestohlener Zugangsdaten. *Backups*, *Segmentierung* und *Monitoring* sorgen dafür, dass ein Angriff früh erkannt wird und sich die Folgen beheben lassen. Mail- und Endpoint-Schutz ergänzen sich dabei – sie werden in den folgenden Kapiteln genauer vorgestellt.

Das Wichtigste in Kürze

- Malware umfasst Viren, Würmer, Trojaner, Ransomware, Spyware und Rootkits.
- Ransomware ist eine Top-Bedrohung – getestete Offline-Backups sind Pflicht.
- Der Mensch ist ein Hauptangriffsziel, vor allem über Phishing und BEC.
- Angriffe verlaufen in Phasen; im Vorfall isolieren statt ausschalten.
- Schutz immer mehrschichtig denken.

Antivirus & NGAV

Von Signaturen zu verhaltensbasierter Erkennung

Nach diesem Kapitel können Sie …

- erklären, wie signaturbasierter Virenschutz funktioniert und wo seine Grenzen liegen,
- heuristische und verhaltensbasierte Erkennung sowie NGAV einordnen,
- mit Fehlalarmen und Ausnahmen verantwortungsvoll umgehen,
- typische Umgehungstechniken benennen und begründen, warum Antivirus allein nicht mehr genügt.

Klassisches Antivirus: der Abgleich mit Fingerabdrücken

Der klassische Virenschutz arbeitet mit **Signaturen**. Eine Signatur ist eine Art digitaler Fingerabdruck einer bekannten Schadsoftware – zum Beispiel ein charakteristisches Byte-Muster oder ein Hashwert der Datei. Der Virenscanner vergleicht jede geprüfte Datei mit seiner Datenbank. Gibt es eine Übereinstimmung, schlägt er Alarm und verschiebt die Datei in die Quarantäne oder löscht sie.

Dieses Verfahren ist bei bekannter Malware sehr zuverlässig und erzeugt kaum Fehlalarme, denn eine Signatur passt nur auf genau das, wofür sie geschrieben wurde. Genau darin liegt aber auch seine Schwäche: Gegenüber brandneuen Varianten ist ein rein signaturbasierter Scanner blind. Schon kleine Veränderungen am Schadcode können dazu führen, dass der Fingerabdruck nicht mehr passt. Außerdem braucht dieser Ansatz ständige Signatur-Updates – ein Virenschutz mit veralteter Datenbank schützt nur noch gegen die Bedrohungen von gestern.

Heuristik und Verhaltenserkennung

Um auch unbekannte Schadsoftware zu erkennen, kamen weitere Verfahren hinzu. Statt nur zu fragen „Kenne ich das?“, fragt die moderne Erkennung: „Verhält sich das verdächtig?“ Die **Heuristik** untersucht eine Datei auf typische Merkmale von Schadcode, etwa ungewöhnliche Befehlsfolgen oder Tarnmechanismen. Die **verhaltensbasierte Erkennung** beobachtet Programme während der Ausführung und greift ein, wenn sie auffällige Aktionen durchführen.

Ein klassisches Beispiel ist die Massen-Verschlüsselung: Beginnt ein Programm plötzlich, in kurzer Zeit Hunderte von Dokumenten zu öffnen, zu verschlüsseln und zu überschreiben, deutet das stark auf Ransomware hin – ganz gleich, ob die konkrete Variante schon bekannt ist. Der Vorteil liegt auf der Hand: Auch neue Schädlinge werden erkannt. Der

Preis dafür ist ein höheres Risiko für Fehlalarme, denn auch legitime Programme verhalten sich manchmal ungewöhnlich, etwa ein Backup-Werkzeug oder eine Verschlüsselungssoftware.

Next-Generation Antivirus (NGAV)

Die moderne Stufe des Virenschutzes heißt **Next-Generation Antivirus**, kurz NGAV. Sie kombiniert mehrere Techniken miteinander. Statt sich nur auf Signaturen zu verlassen, setzt NGAV auf *maschinelles Lernen*: Modelle wurden mit großen Mengen gutartiger und schädlicher Dateien trainiert und bewerten anhand vieler Merkmale, wie wahrscheinlich eine Datei schädlich ist. Hinzu kommt die Verhaltensanalyse während der Ausführung.

Ein weiterer Baustein sind *Cloud-Lookups*: Der Client fragt bei unbekannten Dateien oder Adressen in Echtzeit eine Cloud-Datenbank des Herstellers ab, die ständig mit aktuellen Bedrohungsdaten aus vielen Quellen gefüttert wird. So profitiert jedes Gerät von Erkenntnissen, die an anderer Stelle gerade erst gewonnen wurden. Insgesamt verschiebt sich der Fokus weg von einzelnen Dateien hin zu *Techniken* und Verhaltensmustern, die Angreifer typischerweise einsetzen.

Verfahren	Stärke	Schwäche
Signaturen	Sehr zuverlässig bei Bekanntem, wenige Fehlalarme	Blind bei neuen Varianten, braucht Updates
Heuristik / Verhalten	Erkennt auch Unbekanntes	Mehr Fehlalarme
NGAV	Kombiniert ML, Verhalten und Cloud-Daten	Sieht trotzdem nicht die ganze Angriffskette

Fehlalarme bewusst managen

False Positives, also Fehlalarme, sind die Kehrseite einer aggressiven Erkennung. Dabei wird legitime Software fälschlich als schädlich eingestuft und blockiert – im schlimmsten Fall eine geschäftskritische Fachanwendung. Die naheliegende Reaktion ist eine Ausnahme im Virenschutz. Solche Ausnahmen sollten jedoch mit Bedacht gesetzt werden: so eng wie möglich (etwa für eine bestimmte Datei statt für einen ganzen Ordner), begründet und dokumentiert. Jede Ausnahme ist schließlich auch ein mögliches Schlupfloch.

Ein zweites Problem ist die *Abstumpfung*. Wer täglich Dutzende unbegründete Warnungen sieht, nimmt sie irgendwann nicht mehr ernst – und übersieht dann auch die echte. Ziel ist deshalb nicht die maximale Zahl an Alarmen, sondern eine gute Balance zwischen Erkennungsrate und Fehlalarmen.

> **Merke**
>
> Jede Ausnahme im Virenschutz ist eine bewusst geöffnete Lücke. Setzen Sie Ausnahmen so eng wie möglich, begründen Sie sie und halten Sie fest, wer sie wann und warum eingerichtet hat.

Wie Angreifer die Erkennung umgehen

Angreifer kennen die Funktionsweise von Virenscannern und versuchen gezielt, sie zu umgehen. Man spricht von **AV-Evasion**. Eine verbreitete Technik ist das *Packen oder Verschlüsseln* des Schadcodes: Die eigentliche Schadfunktion wird erst zur Laufzeit entpackt, sodass die Datei auf der Festplatte harmlos aussieht und keine Signatur greift.

Noch schwerer zu fassen sind *dateilose Angriffe*. Sie legen gar keine verdächtige Datei auf der Festplatte ab, sondern laufen direkt im Arbeitsspeicher. Eng verwandt ist das Prinzip *„Living off the Land“* (LotL): Statt eigene Werkzeuge mitzubringen, missbrauchen Angreifer legitime Bordmittel des Betriebssystems, etwa die PowerShell oder Verwaltungswerkzeuge. Weil diese Programme vertrauenswürdig und für den Betrieb notwendig sind, lassen sie sich nicht einfach blockieren.

Betrieb im Unternehmen und Grenzen

Im Unternehmen wird der Virenschutz **zentral verwaltet**. Eine Management-Konsole zeigt, ob auf allen Endpunkten der Schutz aktiv ist, ob Definitionen und Scan-Engine aktuell sind und wo Funde aufgetreten sind. Neben dem Echtzeitschutz, der Dateien beim Zugriff prüft, gehören regelmäßige vollständige Scans zum Standard. Ausnahmen werden zentral, begründet und dokumentiert gepflegt.

Trotzdem hat reines Antivirus klare Grenzen. Es sieht meist nur die einzelne Datei, nicht die gesamte Angriffskette. Dateilose und mehrstufige Angriffe entgehen ihm häufig, und wenn doch einmal etwas durchrutscht, fehlt die Möglichkeit zur Rückschau: Was ist auf dem Gerät passiert, welche Prozesse wurden gestartet, wohin wurde kommuniziert? Deshalb wird der Virenschutz heute ergänzt – durch **EDR/XDR**, das im nächsten Kapitel behandelt wird, durch **Application Control**, die nur freigegebene Programme zulässt, und durch Awareness-Maßnahmen gegen Social Engineering.

> **Aus der Praxis**
>
> Prüfen Sie in der zentralen Konsole regelmäßig, ob wirklich alle Geräte gemeldet und aktuell sind. Ein einzelner Rechner mit abgeschaltetem oder veraltetem Schutz – etwa ein selten genutztes Notebook – ist oft genau die Lücke, die ein Angreifer findet.

Das Wichtigste in Kürze

- Signaturen sind stark bei Bekanntem, aber blind bei Neuem.
- Verhaltensbasierte Erkennung und NGAV erkennen auch Unbekanntes.
- Fehlalarme und Ausnahmen bewusst und dokumentiert managen.
- Dateilose Angriffe und „Living off the Land“ umgehen klassischen Schutz.
- Antivirus allein reicht heute nicht mehr – EDR ergänzt ihn.

EDR & XDR

Endpunkte überwachen, Angriffe erkennen und reagieren

Nach diesem Kapitel können Sie …

- erklären, wie sich EDR von klassischem Virenschutz unterscheidet,
- beschreiben, welche Daten EDR sammelt und welche Reaktionen es ermöglicht,
- EDR, XDR und SOAR gegeneinander abgrenzen,
- MITRE ATT&CK als gemeinsame Sprache für Angriffstechniken einordnen.

Warum EDR: Erkennen statt nur Blockieren

Klassischer Virenschutz verfolgt vor allem ein Ziel: Schadsoftware am Eintritt zu hindern. Das ist wichtig, aber nicht ausreichend. Wie das vorige Kapitel gezeigt hat, gelingt es Angreifern immer wieder, an dieser Hürde vorbeizukommen – etwa mit dateilosen Angriffen oder dem Missbrauch legitimer Werkzeuge. **Endpoint Detection and Response**, kurz EDR, geht deshalb von einer anderen Grundannahme aus: Ein Angreifer kommt irgendwann durch. Entscheidend ist, ihn dann schnell zu erkennen und zu stoppen.

Dazu zeichnet EDR kontinuierlich auf, was auf einem Endpunkt – also einem Notebook, Arbeitsplatzrechner oder Server – geschieht. Es bewertet nicht nur einzelne Dateien, sondern erkennt verdächtige *Ketten* von Ereignissen. Ein Office-Dokument, das eine PowerShell startet, die wiederum eine Verbindung ins Internet aufbaut und anschließend Dateien nachlädt, ist in jedem Einzelschritt unauffällig, als Abfolge aber hochverdächtig. Zugleich ermöglicht EDR eine gezielte Reaktion direkt auf dem betroffenen Gerät.

Wie EDR arbeitet

Auf jedem Endpunkt läuft ein *Sensor* (auch Agent genannt), der laufend **Telemetrie** sammelt: welche Prozesse gestartet werden und von wem, welche Netzwerkverbindungen entstehen, welche Änderungen an der Registry vorgenommen werden und welche Dateien erstellt, verändert oder gelöscht werden. Diese Daten werden an eine zentrale Plattform übertragen und dort ausgewertet.

Die Plattform **korreliert** die einzelnen Ereignisse zu einer zusammenhängenden Angriffsgeschichte. Statt einer isolierten Meldung erhält das Security-Team einen Alarm mit Kontext: Auf welchem Gerät hat es begonnen, welcher Prozess war der Auslöser, was ist danach passiert? Das spart bei der Bewertung viel Zeit.

Für die **Reaktion** stellt EDR wirksame Werkzeuge bereit. Ein verdächtiger Host lässt sich mit einem Klick vom Netz *isolieren*, während die Verbindung zur EDR-Plattform bestehen bleibt – so kann der Angriff sich nicht weiter ausbreiten, die Analyse aber weiterlaufen. Schädliche Prozesse können beendet und, je nach Produkt, vorgenommene Änderungen zurückgerollt werden.

> **Merke**
>
> EDR folgt dem Dreiklang aufzeichnen, erkennen, reagieren. Die Grundhaltung lautet: nicht nur verhindern, sondern davon ausgehen, dass ein Angriff gelingen kann – und ihn dann schnell eindämmen.

Forensik und Threat Hunting

Weil EDR die Aktivitäten auf den Endpunkten aufzeichnet, liefert es zugleich die Datenbasis für die **Forensik**. Nach einem Vorfall lässt sich nachvollziehen, wie der Angriff abgelaufen ist: über welchen Weg die Angreifer eingedrungen sind, welche Konten sie benutzt und welche Systeme sie erreicht haben. Diese Rückschau fehlt beim klassischen Virenschutz.

Darüber hinaus ermöglicht EDR **Threat Hunting** – die proaktive Suche nach Spuren von Angreifern, bevor ein Alarm ausgelöst wurde. Ein typischer Anlass sind *Indicators of Compromise* (IOCs), also Merkmale eines bekannten Angriffs wie bestimmte Dateihashes, Domains oder IP-Adressen. Taucht ein solcher Indikator irgendwo auf, kann das Team gezielt im gesamten Gerätebestand danach suchen und sofort sehen, ob weitere Systeme betroffen sind.

Von EDR zu XDR

EDR konzentriert sich auf den Endpunkt. Angriffe spielen sich jedoch selten nur dort ab: Sie beginnen mit einer Phishing-Mail, nutzen gestohlene Identitäten, bewegen sich durch das Netzwerk und greifen auf Cloud-Dienste zu. **Extended Detection and Response**, kurz XDR, erweitert deshalb die Sicht über den Endpunkt hinaus auf Netzwerk, E-Mail, Identität und Cloud und korreliert die Signale all dieser Ebenen miteinander.

Der Gewinn ist mehr Kontext und damit weniger blinde Flecken. Eine verdächtige Anmeldung allein mag harmlos wirken, ebenso ein ungewöhnlicher Prozess auf einem Rechner. Erst die Verknüpfung – dieselbe Person hat kurz zuvor eine Phishing-Mail geöffnet – zeigt den Zusammenhang, der auf einer einzelnen Ebene unsichtbar bliebe.

Ansatz	Blickwinkel	Kernnutzen
EDR	Endpunkt	Aufzeichnen, erkennen, auf dem Gerät reagieren
XDR	Endpunkt, Netzwerk, Mail, Identität, Cloud	Zusammenhänge über Ebenen hinweg
SOAR	Prozesse und Abläufe	Reaktion automatisieren

SOAR: Reaktion automatisieren

Ein verwandter Begriff ist **SOAR** – Security Orchestration, Automation and Response. SOAR automatisiert die Reaktion auf Sicherheitsvorfälle mithilfe von *Playbooks*. Ein Playbook beschreibt Schritt für Schritt, was bei einem bestimmten Vorfalltyp geschehen soll, und führt wiederkehrende Aufgaben selbstständig aus. Wird etwa ein Konto bei einer verdächtigen Anmeldung erkannt, kann SOAR es automatisch sperren, den betroffenen Rechner isolieren und ein Ticket mit allen Informationen anlegen. Das entlastet das Team von Routinearbeit und verkürzt die Zeit bis zur Eindämmung deutlich.

MITRE ATT&CK als gemeinsame Sprache

Um Erkennungsfähigkeiten einzuordnen, hat sich **MITRE ATT&CK** etabliert – ein frei verfügbarer Katalog realer Angriffstechniken, gegliedert nach den Zielen der Angreifer, etwa Erstzugang, Persistenz oder laterale Bewegung. ATT&CK lenkt den Blick auf *Techniken* statt auf einzelne Signaturen. Unternehmen können ihre eigene Erkennung gegen den Katalog abbilden und so sehen, welche Techniken sie bereits erkennen und wo Lücken bestehen. Zugleich schafft ATT&CK eine gemeinsame Sprache im Team und mit Dienstleistern.

Auswahl und Betrieb

Am Markt gibt es zahlreiche Lösungen; bekannte Beispiele sind CrowdStrike Falcon, SentinelOne und Microsoft Defender for Endpoint. Welche passt, hängt von der vorhandenen Umgebung, der Integration in bestehende Werkzeuge und dem Budget ab. Wichtiger als die Produktwahl ist der Betrieb: Die Sensoren müssen möglichst *lückenlos* auf allen Endpunkten laufen, denn jedes ungeschützte Gerät ist ein blinder Fleck. Die Erkennungsregeln müssen getunt werden, um Fehlalarme zu senken, und Reaktions-Playbooks sollten vorbereitet sein, bevor der Ernstfall eintritt.

Aus der Praxis

EDR erzeugt Alarme rund um die Uhr – auch nachts und am Wochenende. Kleinere Unternehmen ohne eigenes Security-Team greifen deshalb häufig auf *Managed EDR* oder ein externes SOC zurück, bei dem Fachleute die Alarme sichten und im Notfall sofort reagieren.

Das Wichtigste in Kürze

- EDR zeichnet Endpunkte auf, erkennt verdächtige Ketten und ermöglicht gezielte Reaktion.
- XDR erweitert die Sicht auf Netzwerk, Mail, Identität und Cloud.
- SOAR automatisiert wiederkehrende Reaktionen über Playbooks.
- MITRE ATT&CK hilft, Erkennung und Lücken einzuordnen.
- Grundhaltung: Erkennung statt reiner Prävention.

Firewall & Netzwerksicherheit

Vom Paketfilter zur Next-Generation Firewall

Nach diesem Kapitel können Sie …

- die Aufgabe einer Firewall und das Prinzip „Default Deny“ erklären,
- Paketfilter, Stateful Firewall und Next-Generation Firewall unterscheiden,
- Proxy, SSL-Inspection sowie IDS und IPS einordnen,
- Segmentierung, DMZ und eine saubere Regelpflege begründen.

Die Firewall als Torwächter

Eine **Firewall** kontrolliert den Datenverkehr zwischen Netzen anhand von Regeln. Sie steht an den Übergängen, etwa zwischen dem Firmennetz und dem Internet, und entscheidet für jede Verbindung, ob sie durchgelassen wird oder nicht. Damit trennt sie vertrauenswürdige von unsicheren Zonen und ist ein zentraler Baustein der Netzwerksicherheit.

Das wichtigste Grundprinzip lautet **Default Deny**: Zunächst ist alles verboten, und nur das, was ausdrücklich gebraucht wird, wird gezielt erlaubt. Der umgekehrte Ansatz – alles erlauben und nur Bekanntes verbieten – ist zu riskant, weil jede vergessene Lücke offen bliebe. Eine Firewall nach Default Deny schützt dagegen auch vor Verbindungen, an die beim Einrichten niemand gedacht hat.

Entwicklungsstufen: vom Paketfilter zur NGFW

Der einfachste Typ ist der **Paketfilter**. Er entscheidet für jedes einzelne Paket nur anhand von Absender- und Zieladresse sowie Port, ohne einen Zusammenhang zu kennen – er arbeitet also zustandslos. Das ist simpel und schnell, aber zu grob: Für jede Verbindung müssen Hin- und Rückweg einzeln freigegeben werden, und gefälschte Pakete lassen sich kaum erkennen.

Die **Stateful Firewall** verfolgt dagegen den Zustand jeder Verbindung. Baut ein Rechner im Firmennetz eine Verbindung zu einem Webserver auf, merkt sie sich das und lässt die passenden Antworten automatisch zurück. Unaufgeforderte Pakete von außen, die zu keiner bestehenden Verbindung gehören, werden verworfen. Stateful Inspection ist heute der Mindeststandard.

Der aktuelle Standard ist die **Next-Generation Firewall** (NGFW). Sie erkennt nicht nur Ports, sondern konkrete *Anwendungen* und Inhalte. Das ist wichtig, weil viele Anwendungen heute über dieselben Ports laufen, vor allem über den Web-Port 443. Mit *Appli-*

cation Control lassen sich gezielt einzelne Anwendungen erlauben oder sperren. Möglich macht das die *Deep Packet Inspection* (DPI), bei der die Firewall auch in die Nutzdaten der Pakete hineinsieht. Hinzu kommen meist ein integriertes Intrusion-Prevention-System und Malware-Schutz.

Typ	Entscheidet anhand von	Bewertung
Paketfilter	IP-Adresse und Port, zustandslos	Zu grob für heutige Anforderungen
Stateful Firewall	Verbindungszustand	Mindeststandard
NGFW	Anwendung und Inhalt (DPI), mit IPS	Aktueller Standard

Proxy und SSL-Inspection

Firewall und **Proxy** lösen unterschiedliche Aufgaben, werden aber oft im selben Gerät kombiniert. Ein *Forward-Proxy* vermittelt den ausgehenden Verkehr der Mitarbeitenden ins Internet. Er kann Webadressen nach Kategorien filtern, häufig abgerufene Inhalte zwischenspeichern (Caching) und den Zugriff zentral protokollieren. Ein *Reverse-Proxy* steht dagegen vor einem Server: Er nimmt Anfragen aus dem Internet entgegen, verbirgt die echte Adresse des Servers, verteilt die Last auf mehrere Systeme und übernimmt häufig die TLS-Verschlüsselung.

Eine besondere Herausforderung ist verschlüsselter Verkehr. Da heute fast alle Verbindungen per TLS verschlüsselt sind, kann eine Firewall ohne weitere Maßnahmen nicht hineinsehen – Schadsoftware käme unerkannt durch. Bei der **SSL-Inspection** entschlüsselt die Firewall den Verkehr deshalb, prüft ihn und verschlüsselt ihn anschließend neu. Damit das funktioniert, müssen die Geräte im Unternehmen einem Zertifikat der Firewall vertrauen. Dabei sind Datenschutz und Mitbestimmung zu beachten, und für sensible Bereiche wie Online-Banking oder Gesundheitsportale werden üblicherweise Ausnahmen eingerichtet.

IDS und IPS

Ein **Intrusion Detection System** (IDS) erkennt Angriffe im Netzverkehr und meldet sie – es arbeitet passiv und greift selbst nicht ein. Ein **Intrusion Prevention System** (IPS) sitzt dagegen direkt im Datenstrom (inline) und kann erkannte Angriffe aktiv blockieren. Beide arbeiten signaturbasiert, also mit Mustern bekannter Angriffe, und anomaliebasiert, indem sie Abweichungen vom normalen Verkehr aufspüren. Moderne NGFWs bringen ein IPS meist gleich mit.

> **Merke**
>
> Ein IDS erkennt und meldet, ein IPS erkennt und blockiert. Default Deny ist das Grundprinzip jeder Firewall; Stateful ist das Minimum, die NGFW der heutige Standard.

Segmentierung, Zonen und DMZ

Nicht alles gehört in dasselbe Netz. Bei der **Segmentierung** wird das Firmennetz in mehrere Bereiche aufgeteilt, zum Beispiel für Clients, Server, Drucker, Produktion und Gäste. Zwischen diesen Segmenten regelt die Firewall, wer mit wem sprechen darf. Gelingt einem Angreifer der Einstieg in ein Segment, kann er sich nicht ungehindert weiter ausbreiten. Man spricht von *Trust-Zonen* mit klar definierten Übergängen.

Dabei ist nicht nur der Verkehr zwischen innen und außen wichtig (Nord-Süd), sondern auch der interne Verkehr zwischen Systemen im Firmennetz (Ost-West). Gerade über diesen Weg bewegen sich Angreifer nach dem Erstzugang lateral weiter. Für besonders kritische Systeme geht man noch einen Schritt weiter und setzt auf *Mikrosegmentierung*, bei der jede einzelne Anwendung oder jeder Server eigene, eng gefasste Regeln erhält.

Dienste, die aus dem Internet erreichbar sein müssen – etwa ein Webserver oder ein Mail-Gateway –, kommen in eine **DMZ** (demilitarisierte Zone). Dieses eigene Segment liegt zwischen Internet und internem Netz. Wird ein Server in der DMZ kompromittiert, steht dem Angreifer nicht automatisch das gesamte Firmennetz offen.

Das Regelwerk pflegen

Eine Firewall ist nur so gut wie ihr Regelwerk. Regeln nach dem Muster „Any-Any“, die beliebigen Verkehr von überall nach überall erlauben, haben darin nichts zu suchen. Jede Regel sollte so eng wie möglich gefasst sein: nur die nötigen Quellen, Ziele und Dienste. Außerdem gehört zu jeder Regel ein Kommentar mit Zweck, Ersteller und Datum. So lässt sich auch Jahre später nachvollziehen, warum sie existiert.

Mit der Zeit sammeln sich Regeln an, die niemand mehr braucht – etwa für einen längst abgeschalteten Server. Solche Altlasten vergrößern die Angriffsfläche und machen das Regelwerk unübersichtlich. Ein regelmäßiges Review, bei dem nicht mehr benötigte Regeln entfernt werden, gehört deshalb zum festen Betrieb.

> **Aus der Praxis**
>
> Legen Sie für jede neue Firewall-Regel ein Ablauf- oder Prüfdatum fest. Temporäre Freigaben, etwa für einen Dienstleister, geraten sonst schnell in Vergessenheit und bleiben jahrelang offen.

Das Wichtigste in Kürze

- Default Deny – alles verbieten, nur gezielt erlauben.
- Stateful ist Minimum, die NGFW mit Application Control und DPI ist Standard.
- SSL-Inspection, IPS und Proxy ergänzen die Firewall.
- Segmentierung begrenzt Angriffe, die DMZ trennt öffentliche Dienste vom internen Netz.
- Regeln eng fassen, dokumentieren und regelmäßig prüfen.

Mail-Security & Anti-Spam

SPF, DKIM, DMARC und Schutz vor Phishing

Nach diesem Kapitel können Sie …

- begründen, warum E-Mail der wichtigste Angriffsweg ist,
- SPF, DKIM und DMARC erklären und ihr Zusammenspiel beschreiben,
- Schutzmaßnahmen auf Inhaltsebene wie Sandbox und Link-Prüfung einordnen,
- die Rolle der Mitarbeitenden im Mailschutz einschätzen.

E-Mail als Hauptangriffsweg

Die meisten Angriffe auf Unternehmen beginnen mit einer E-Mail. Über diesen Weg kommen Phishing-Nachrichten, die zur Eingabe von Zugangsdaten verleiten, Anhänge mit Schadsoftware und gefälschte Zahlungsanweisungen im Sinne des Business E-Mail Compromise (BEC). E-Mail ist für Angreifer so attraktiv, weil jede Person im Unternehmen ein Postfach hat und Nachrichten von außen ganz selbstverständlich empfangen werden.

Hinzu kommt ein grundsätzliches Problem: Das E-Mail-Protokoll SMTP stammt aus einer Zeit, in der Sicherheit kaum eine Rolle spielte. Der Absender einer Mail lässt sich deshalb technisch erschreckend leicht fälschen – ähnlich wie bei einem Brief, auf dessen Umschlag man jeden beliebigen Absender schreiben kann. Wirksamer Schutz muss daher auf zwei Ebenen ansetzen: bei der *Echtheit des Absenders* auf Protokollebene und bei der Prüfung der *Inhalte*.

SPF: Wer darf für eine Domain senden?

Das **Sender Policy Framework** (SPF) legt per DNS fest, welche Mailserver berechtigt sind, im Namen einer Domain E-Mails zu versenden. Der Domaininhaber hinterlegt dazu einen DNS-TXT-Eintrag mit einer Liste erlaubter Server oder Dienste. Erhält ein Mailserver eine Nachricht, fragt er diesen Eintrag ab und prüft, ob der einliefernde Server darin aufgeführt ist.

SPF schützt damit vor der Fälschung der sogenannten *Envelope-Domain*, also der Absenderadresse, die beim Transport zwischen den Servern verwendet wird. Die Adresse, die Empfänger im Mailprogramm als „Von“ sehen, wird von SPF allein jedoch nicht geprüft – dafür braucht es das Zusammenspiel mit DMARC.

DKIM: die digitale Unterschrift

DomainKeys Identified Mail (DKIM) signiert ausgehende E-Mails kryptografisch. Der versendende Mailserver berechnet aus wichtigen Kopfzeilen und dem Inhalt der Nachricht eine Signatur und fügt sie als zusätzliche Kopfzeile hinzu. Dafür verwendet er einen privaten Schlüssel, den nur er kennt. Der passende öffentliche Schlüssel liegt im DNS der Absenderdomain.

Der Empfänger ruft diesen öffentlichen Schlüssel ab und verifiziert damit die Signatur. Stimmt sie, ist belegt, dass die Mail tatsächlich von einem berechtigten Server der Domain stammt und unterwegs nicht verändert wurde. DKIM sichert also Echtheit und Unveränderheit – zwei der Schutzziele aus dem ersten Kapitel.

DMARC: die Richtlinie darüber

DMARC (Domain-based Message Authentication, Reporting and Conformance) verknüpft SPF und DKIM mit einer Richtlinie. Es beantwortet die Frage: Was soll der Empfänger tun, wenn die Prüfung scheitert? Zugleich verlangt DMARC, dass die geprüfte Domain zur sichtbaren Absenderadresse passt – erst dadurch wird auch das „Von“-Feld wirksam geschützt.

Die Richtlinie kennt drei Stufen. Bei `none` wird nur beobachtet, die Mail wird normal zugestellt. Bei `quarantine` landen fehlgeschlagene Mails im Spam- oder Quarantäneordner, bei `reject` werden sie abgewiesen. Ein großer Vorteil von DMARC sind die *Reports*: Empfangende Server schicken regelmäßig Berichte darüber, welche Server Mails im Namen der Domain versendet haben. So wird sichtbar, ob die eigene Domain missbraucht wird – und ob alle legitimen Versender, etwa ein Newsletter-Dienst, korrekt eingetragen sind.

```
beispiel.de.          TXT  "v=spf1 mx include:_spf.mailanbieter.example -all"
_dmarc.beispiel.de.   TXT  "v=DMARC1; p=none; rua=mailto:dmarc@beispiel.de"
```

Das Beispiel zeigt einen SPF-Eintrag, der die eigenen Mailserver und einen externen Anbieter erlaubt und alle anderen ablehnt, sowie einen DMARC-Eintrag in der Beobachtungsstufe, dessen Berichte an eine eigene Adresse gehen.

> **Merke**
>
> SPF prüft den sendenden Server, DKIM die Signatur der Mail, DMARC legt fest, was bei einem Fehlschlag passiert. Starten Sie DMARC mit `none`, werten Sie die Reports aus und verschärfen Sie dann schrittweise über `quarantine` bis `reject`.

Inhaltsprüfung: Filter, Sandbox und Link-Schutz

Auch eine Mail mit korrektem SPF, DKIM und DMARC kann schädlich sein – etwa wenn Angreifer eine eigene, ähnlich klingende Domain registriert oder ein echtes Postfach übernommen haben. Deshalb braucht es zusätzlich eine Prüfung der Inhalte. **Anti-Spam- und Anti-Malware-Filter** bewerten eingehende Nachrichten nach Reputation des Absenders, typischen Spam-Merkmalen und bekannten Schadsignaturen.

Verdächtige Anhänge werden in einer **Sandbox** geöffnet, einer abgeschotteten Testumgebung. Dort wird beobachtet, was die Datei tut – man sagt, sie wird „detoniert". Verhält sie sich schädlich, erreicht sie das Postfach gar nicht erst. Zusätzlich lassen sich Anhänge nach Typ filtern: Ausführbare Dateien oder Office-Dokumente mit Makros haben im normalen Mailverkehr selten etwas zu suchen.

Links in E-Mails werden häufig *umgeschrieben* und erst im Moment des Klicks geprüft (*Time-of-Click*). Das ist wichtig, weil Angreifer eine harmlose Webseite oft erst nach der Zustellung in eine schädliche verwandeln. Ein *Impersonation-Schutz* erkennt Nachrichten, die sich als bekannte Personen ausgeben, etwa als Geschäftsführung mit leicht abgewandelter Adresse. Und ein *Warnbanner* bei externen Absendern macht auf einen Blick sichtbar, dass eine Mail nicht aus dem eigenen Haus stammt.

Lösungen am Markt

Mailschutz wird entweder als vorgeschaltetes *Gateway* umgesetzt, über das alle Nachrichten laufen, bevor sie den eigenen Mailserver erreichen, oder direkt in den Cloud-Maildienst integriert. Bekannte Anbieter sind zum Beispiel Hornetsecurity, Proofpoint und Mimecast; für Microsoft-365-Umgebungen gibt es Microsoft Defender for Office 365. Welche Lösung passt, hängt von der vorhandenen Mailplattform, dem gewünschten Funktionsumfang und dem Budget ab.

Die Mitarbeitenden einbeziehen

Selbst die beste Technik fängt nicht jede gefälschte Nachricht ab. Aufmerksame Nutzerinnen und Nutzer sind deshalb die letzte und oft entscheidende Schutzschicht. Regelmäßige **Awareness-Schulungen** vermitteln, woran man Phishing erkennt: ungewöhnlicher Zeitdruck, abweichende Absenderadressen, unerwartete Anhänge oder Links, die auf fremde Domains führen. **Phishing-Simulationen** trainieren dieses Wissen im Alltag, ohne echten Schaden anzurichten.

Aus der Praxis

Ein einfacher *Melde-Button* im Mailprogramm senkt die Hürde, Verdachtsfälle weiterzugeben. Wichtig ist eine positive Fehlerkultur: Wer versehentlich geklickt hat, sollte das sofort melden können, ohne Ärger befürchten zu müssen – denn jede Minute zählt.

Das Wichtigste in Kürze

- E-Mail ist der Angriffsweg Nummer eins, Absender lassen sich leicht fälschen.
- SPF, DKIM und DMARC schützen gemeinsam vor Absender-Fälschung.
- DMARC schrittweise von `none` bis `reject` scharf schalten.
- Sandbox, Link-Schutz und Warnbanner sichern die Inhaltsebene ab.
- Geschulte Mitarbeitende und ein einfacher Meldeweg ergänzen die Technik.

SSL/TLS & PKI

Verschlüsselte Verbindungen und Zertifikate verstehen

Nach diesem Kapitel können Sie …

- symmetrische und asymmetrische Verschlüsselung sowie Hashing unterscheiden,
- erklären, was TLS leistet und wie der Handshake abläuft,
- die Zertifikatskette bis zur Root-CA nachvollziehen,
- öffentliche Zertifikate, interne PKI und typische Fehlerquellen im Betrieb einordnen.

Die Grundlagen der Verschlüsselung

Fast alle sicheren Verbindungen im Internet beruhen auf zwei grundsätzlichen Verfahren. Bei der **symmetrischen Verschlüsselung** verwenden beide Seiten denselben Schlüssel zum Ver- und Entschlüsseln. Ein bekanntes Verfahren ist AES. Symmetrische Verschlüsselung ist sehr schnell und eignet sich daher für große Datenmengen. Ihr Problem ist der sichere Schlüsselaustausch: Wie bekommen zwei Parteien, die sich noch nie begegnet sind, denselben geheimen Schlüssel, ohne dass ihn jemand unterwegs abfängt?

Hier hilft die **asymmetrische Verschlüsselung**. Jede Partei besitzt ein Schlüsselpaar aus einem *öffentlichen* und einem *privaten* Schlüssel. Was mit dem öffentlichen Schlüssel verschlüsselt wird, kann nur der zugehörige private Schlüssel entschlüsseln. Der öffentliche Schlüssel darf also jedem bekannt sein, der private bleibt geheim. Bekannte Verfahren sind RSA und Verfahren auf Basis elliptischer Kurven. Asymmetrische Verschlüsselung löst das Austauschproblem, ist aber deutlich rechenintensiver.

TLS kombiniert deshalb beide Ansätze: Asymmetrische Verfahren kommen im Handshake zum Einsatz, um sich sicher auf einen gemeinsamen Sitzungsschlüssel zu einigen; die eigentlichen Nutzdaten werden danach schnell symmetrisch verschlüsselt. Ein verwandtes Werkzeug ist das **Hashing**, etwa mit SHA-256. Ein Hash ist ein eindeutiger Prüfwert fester Länge, der sich nicht zurückrechnen lässt. Er ist keine Verschlüsselung, sichert aber die Integrität: Schon die kleinste Änderung an den Daten ergibt einen völlig anderen Hashwert.

Verfahren	Prinzip	Stärke / Schwäche
Symmetrisch (z. B. AES)	Ein gemeinsamer Schlüssel	Schnell, aber Schlüsselaustausch schwierig
Asymmetrisch (z. B. RSA)	Öffentlicher und privater Schlüssel	Löst den Austausch, aber rechenintensiv
Hashing (z. B. SHA-256)	Eindeutiger Prüfwert	Sichert Integrität, keine Verschlüsselung

Wozu TLS dient

Transport Layer Security (TLS) sichert Verbindungen zwischen zwei Endpunkten ab – am bekanntesten beim Aufruf von Webseiten über HTTPS, aber ebenso bei E-Mail, VPNs oder Programmierschnittstellen. TLS sorgt dabei für drei Dinge gleichzeitig: *Vertraulichkeit*, damit niemand mitliest, *Integrität*, damit Daten unterwegs nicht verändert werden, und *Authentizität*, damit sichergestellt ist, dass der Server wirklich der ist, für den er sich ausgibt. Für Letzteres sorgt das Zertifikat. Der Begriff SSL ist übrigens der Name des Vorgängerprotokolls; im Alltag werden beide Begriffe oft synonym verwendet, gemeint ist heute aber immer TLS.

Der TLS-Handshake

Bevor die eigentlichen Daten fließen, führen Client und Server einen **Handshake** durch. Zunächst einigen sie sich auf die TLS-Version und die Verschlüsselungsverfahren, die beide unterstützen. Der Server schickt sein Zertifikat, das der Client prüft: Ist es für diesen Namen ausgestellt, noch gültig und von einer vertrauenswürdigen Stelle signiert? Anschließend etablieren beide über einen Schlüsselaustausch einen gemeinsamen *Sitzungsschlüssel*. Erst danach beginnt die eigentliche Übertragung, symmetrisch verschlüsselt mit diesem Schlüssel.

Die Zertifikatskette

Woher weiß ein Browser, dass er einem Zertifikat trauen kann? Die Antwort ist die **Zertifikatskette**. Das *Server-Zertifikat* ist für eine konkrete Domain ausgestellt und von einer Zertifizierungsstelle, einer *CA* (Certificate Authority), digital signiert. Diese Signatur stammt in der Regel nicht direkt von der obersten Stelle, sondern von einem *Zwischenzertifikat*. Das Zwischenzertifikat wiederum ist von einer *Root-CA* signiert. Die Zertifikate der bekannten Root-CAs sind in Betriebssystemen und Browsern bereits hinterlegt; ihnen wird von Haus aus vertraut.

Der Browser prüft die Kette Glied für Glied vom Server-Zertifikat bis zur Root. Reißt die Kette an einer Stelle – etwa weil ein Zwischenzertifikat fehlt oder ein Zertifikat abgelaufen ist –, erscheint eine Sicherheitswarnung.

> **Merke**
>
> Vertrauen entsteht über eine lückenlose Kette: Server-Zertifikat, Zwischenzertifikat(e) und Root-CA. Liefern Sie auf dem Server immer die vollständige Kette einschließlich der Zwischenzertifikate aus.

Öffentliche Zertifikate und interne PKI

Für öffentlich erreichbare Webserver hat sich **Let's Encrypt** weit verbreitet: eine Zertifizierungsstelle, die kostenlose, domain-validierte Zertifikate vollautomatisch ausstellt. Ausstellung und Erneuerung laufen über das Protokoll *ACME*. Die Laufzeit ist bewusst kurz gehalten – bislang 90 Tage, mit der branchenweiten Tendenz zu noch kürzeren Laufzeiten. Eine Automatisierung der Erneuerung ist deshalb Pflicht.

Zertifikate unterscheiden sich nach Prüftiefe: Bei *DV* (Domain Validation) wird nur die Kontrolle über die Domain geprüft, bei *OV* (Organization Validation) und *EV* (Extended Validation) zusätzlich die Organisation dahinter. Ein *Wildcard-Zertifikat* gilt für alle Subdomänen einer Ebene, etwa `*.beispiel.de`; ein *SAN-Zertifikat* enthält mehrere konkrete Namen.

Für interne Dienste, die nicht aus dem Internet erreichbar sind, betreiben viele Unternehmen eine eigene **Public Key Infrastructure** (PKI) mit eigener Zertifizierungsstelle. Unter Windows geschieht das typischerweise mit den *Active Directory Certificate Services* (AD CS). Damit die intern ausgestellten Zertifikate akzeptiert werden, muss die interne Root auf allen Clients als vertrauenswürdig hinterlegt sein. Die Root-CA ist das Herzstück dieser Vertrauenskette und sollte besonders geschützt werden – am besten offline, also nur für das Signieren von Zwischenzertifikaten eingeschaltet.

Betrieb und typische Fehlerquellen

Die häufigsten Probleme im Alltag sind **abgelaufene Zertifikate** und **fehlende Zwischenzertifikate**. Ein abgelaufenes Zertifikat gehört zu den häufigsten Ausfallursachen überhaupt: Von einem Moment auf den anderen sind Webseite, Mailserver oder VPN nicht mehr erreichbar. Ablaufdaten sollten daher zentral überwacht und Zertifikate rechtzeitig, idealerweise automatisch, erneuert werden.

Wird ein privater Schlüssel kompromittiert, muss das zugehörige Zertifikat vor Ablauf *widerrufen* werden. Dafür gibt es Sperrlisten (CRL) und die Online-Abfrage OCSP; einige Zertifizierungsstellen setzen inzwischen vor allem auf Sperrlisten und kurze Laufzeiten. Private Schlüssel müssen in jedem Fall streng geschützt werden. Und schließlich gehören veraltete TLS-Versionen und schwache Verfahren abgeschaltet.

```
openssl s_client -connect www.beispiel.de:443 -showcerts
```

Aus der Praxis

Mit dem obigen Befehl zeigen Sie die Zertifikatskette an, die ein Server tatsächlich ausliefert, und erkennen fehlende Zwischenzertifikate sofort. Nehmen Sie außerdem alle Zertifikate mit ihrem Ablaufdatum in das Monitoring auf und lassen Sie sich mehrere Wochen vorher warnen.

Das Wichtigste in Kürze

- TLS kombiniert asymmetrische (Handshake) und symmetrische Verschlüsselung (Nutzdaten).
- TLS sichert Vertraulichkeit, Integrität und Authentizität.
- Vertrauen entsteht über die Zertifikatskette bis zur Root-CA.
- Let's Encrypt automatisiert öffentliche Zertifikate, AD CS die interne PKI.
- Ablauf-Monitoring und vollständige Ketten verhindern die häufigsten Ausfälle.

Hinweis: Maximale Zertifikatslaufzeiten und Sperrverfahren werden von Browserherstellern und Zertifizierungsstellen laufend angepasst. Die Angaben beschreiben deshalb die Grundprinzipien.

Identity & Access Management

MFA, Passkeys, SSO und Zero-Trust-Zugriff

Nach diesem Kapitel können Sie …

- Authentifizierung und Autorisierung sicher unterscheiden,
- erklären, warum MFA und Passkeys den Schutz von Konten entscheidend verbessern,
- SSO und die Standards OAuth 2.0, OIDC, SAML und SCIM einordnen,
- Privileged Access Management, Zero Trust und den Konten-Lebenszyklus beschreiben.

Identität als neue Sicherheitsgrenze

Früher galt das Firmennetz als sicherer Bereich: Wer drinnen war, dem wurde vertraut, und die Firewall schützte die Grenze nach außen. Heute arbeiten Menschen im Homeoffice und unterwegs, Daten liegen in Cloud-Diensten, und viele Anwendungen sind direkt aus dem Internet erreichbar. Die klassische Netzwerkgrenze verliert damit an Bedeutung. An ihre Stelle tritt die **Identität**: Wer sich erfolgreich als eine bestimmte Person anmeldet, erhält Zugriff – ganz gleich, von wo aus. **Identity and Access Management** (IAM) umfasst alle Prozesse und Werkzeuge, mit denen Identitäten verwaltet und Zugriffe gesteuert werden.

Authentifizierung und Autorisierung

Zwei Begriffe werden dabei häufig verwechselt. Die **Authentifizierung** (kurz AuthN) beantwortet die Frage „Wer sind Sie?“ – eine Person weist ihre Identität nach, etwa mit Passwort und zweitem Faktor. Die **Autorisierung** (kurz AuthZ) beantwortet die Frage „Was dürfen Sie?“ – sie legt fest, welche Rechte und Rollen die nachgewiesene Identität hat. Die Reihenfolge ist immer dieselbe: erst die Identität nachweisen, dann die Rechte anwenden.

Bei der Autorisierung gilt das Prinzip der *minimalen Rechte* (Least Privilege): Jede Person erhält nur so viele Rechte, wie sie für ihre Aufgabe tatsächlich benötigt. Rechte werden üblicherweise über Rollen oder Gruppen vergeben, nicht einzeln pro Person – das macht die Verwaltung übersichtlicher und nachvollziehbarer.

Multi-Faktor-Authentifizierung und Passkeys

Ein Passwort allein ist ein schwacher Schutz: Es kann erraten, durch Phishing abgegriffen oder bei einem Datenleck eines anderen Dienstes erbeutet werden. Die **Multi-Faktor-Authentifizierung** (MFA) kombiniert deshalb mehrere Faktoren aus unterschiedlichen Kategorien: etwas, das man *weiß* (Passwort, PIN), etwas, das man *besitzt* (Smartphone, Sicherheitsschlüssel), und etwas, das man *ist* (Fingerabdruck, Gesichtserkennung). Ein erbeutetes Passwort genügt dann nicht mehr für die Anmeldung. MFA gilt als der größte einzelne Hebel gegen die Übernahme von Konten.

Nicht jeder zweite Faktor ist gleich sicher. Einmalcodes per SMS sind angreifbar, etwa durch das Umleiten der Rufnummer auf eine andere SIM-Karte. App-basierte Verfahren mit Bestätigung oder zeitbasierten Codes sind deutlich robuster. Gegen raffinierte Phishing-Seiten, die auch Einmalcodes in Echtzeit abgreifen, schützen sie aber nicht vollständig.

Hier setzen **Passkeys** auf Basis des FIDO2-Standards an. Statt eines Passworts wird ein kryptografisches Schlüsselpaar erzeugt: Der private Schlüssel bleibt auf dem Gerät oder einem Sicherheitsschlüssel, der Dienst kennt nur den öffentlichen. Die Anmeldung wird lokal per Biometrie oder PIN freigegeben. Entscheidend ist, dass der Passkey an die echte Domain gebunden ist. Auf einer gefälschten Seite funktioniert er schlicht nicht – ein Phishing-Versuch läuft ins Leere, weil es kein abgreifbares Passwort mehr gibt. Passkeys gelten deshalb als *phishing-resistent*.

> **Merke**
>
> MFA ist Pflicht für jedes Konto mit Zugang zu Unternehmensdaten. App-basierte Verfahren sind besser als SMS, Passkeys und FIDO2-Sicherheitsschlüssel sind darüber hinaus phishing-resistent.

Single Sign-On, Föderation und Standards

Mit **Single Sign-On** (SSO) melden sich Mitarbeitende einmal an und können danach viele Dienste nutzen, ohne sich jeweils neu anzumelden. Möglich macht das ein zentraler *Identity Provider* (IdP), der die Anmeldung für alle angeschlossenen Dienste übernimmt – man spricht von *Föderation*. Weniger Passwörter bedeuten weniger Angriffsfläche, und Sicherheitsregeln wie MFA lassen sich an einer zentralen Stelle durchsetzen.

Technisch beruht SSO auf offenen Standards. **OAuth 2.0** regelt die Autorisierung, also welche Anwendung in wessen Namen auf welche Daten zugreifen darf. **OpenID Connect** (OIDC) setzt darauf auf und ergänzt die Anmeldung, also den Nachweis der Identität. Im klassischen Unternehmensumfeld ist zudem **SAML** verbreitet, ein älterer, XML-basierter Standard für die Föderation. Für die automatische Bereitstellung von Benutzerkonten in angeschlossenen Diensten sorgt **SCIM**.

Standard	Zweck
OAuth 2.0	Autorisierung: Zugriff von Anwendungen auf Daten
OIDC	Anmeldung (Authentifizierung) auf Basis von OAuth 2.0
SAML	Föderierte Anmeldung im klassischen Enterprise-Umfeld
SCIM	Automatisches Anlegen und Entfernen von Konten

Privileged Access Management

Administrative Konten sind das bevorzugte Ziel von Angreifern, denn wer sie kontrolliert, kontrolliert die IT. **Privileged Access Management** (PAM) sorgt für ihren besonderen Schutz. Dazu gehören getrennte Admin-Konten, die nicht für den Alltag mit E-Mail und Internet genutzt werden. Statt dauerhafter Berechtigungen werden Rechte *Just-in-Time* vergeben – nur für die Dauer einer konkreten Aufgabe. Administrative Sitzungen werden protokolliert, und Passwörter für privilegierte Konten liegen in einem gesicherten Passworttresor, aus dem sie bei Bedarf ausgecheckt werden.

Zero Trust und der Konten-Lebenszyklus

Zero Trust ist eine Grundhaltung: Jeder Zugriff wird geprüft, unabhängig davon, ob er aus dem Firmennetz oder von außen kommt. Über den Zugriff entscheiden die Identität, der Zustand des Geräts – etwa ob es verwaltet und aktuell ist – und der Kontext wie Ort, Uhrzeit oder Risiko der Anmeldung. Durchgesetzt werden solche Richtlinien über *Conditional Access*. Zugrunde liegt die Haltung „Assume Breach“: Man geht davon aus, dass ein Angreifer bereits im Netz sein könnte.

Ebenso wichtig ist der saubere **Lebenszyklus** von Konten nach dem Prinzip *Joiner – Mover – Leaver*. Beim Eintritt (Joiner) werden Konten und Rechte möglichst automatisch bereitgestellt. Beim Wechsel der Rolle (Mover) werden Rechte angepasst – und alte Rechte entzogen, damit sich nicht über die Jahre immer mehr ansammeln. Beim Austritt (Leaver) wird der Zugang sofort entzogen. Regelmäßige *Rezertifizierungen*, bei denen Vorgesetzte die Rechte ihrer Teams bestätigen, decken verbliebene Altlasten auf.

Aus der Praxis

Verknüpfen Sie das Deaktivieren von Konten direkt mit dem Austrittsprozess der Personalabteilung. Verwaiste Konten ehemaliger Mitarbeitender oder Dienstleister sind ein häufiger und leicht vermeidbarer Einstiegspunkt für Angreifer.

Das Wichtigste in Kürze

- Authentifizierung klärt „wer“, Autorisierung „was“ – immer nach Least Privilege.
- MFA ist Pflicht, Passkeys und FIDO2 sind phishing-resistent.
- SSO über einen zentralen IdP mit OAuth 2.0/OIDC oder SAML reduziert Passwörter.
- PAM schützt privilegierte Konten, Zero Trust prüft jeden Zugriff.
- Joiner-Mover-Leaver-Prozesse und Rezertifizierung verhindern Rechte-Wildwuchs.

SIEM & SOC-Grundlagen

Logs zentral auswerten und auf Vorfälle reagieren

Nach diesem Kapitel können Sie …

- begründen, warum Logs zentral gesammelt und ausgewertet werden müssen,
- die Aufgaben eines SIEM beschreiben und gängige Werkzeuge einordnen,
- Aufbau und Arbeitsweise eines SOC erklären,
- die Phasen der Incident Response benennen und Alert-Fatigue vermeiden.

Warum zentrale Logs?

Fast jedes IT-System schreibt Protokolle, sogenannte **Logs**: Der Server vermerkt Anmeldungen, die Firewall blockierte Verbindungen, das Active Directory geänderte Gruppenmitgliedschaften. Einzeln betrachtet sagen diese Einträge meist wenig. Ein Angriff hinterlässt jedoch Spuren über viele Systeme hinweg – ein bisschen auf der Firewall, ein bisschen auf dem Server, ein bisschen im Verzeichnisdienst. Erst die Zusammenschau macht ihn sichtbar.

Solange die Logs verstreut auf Dutzenden Systemen liegen, sind sie praktisch nicht auswertbar. Niemand kann manuell die Protokolle aller Server, Firewalls und Endpunkte vergleichen, schon gar nicht in Echtzeit. Zudem löschen Angreifer gern lokale Logs, um ihre Spuren zu verwischen. Die zentrale Sammlung an einem geschützten Ort ist deshalb die Voraussetzung für jede wirksame Erkennung.

Was ein SIEM leistet

Ein **SIEM** – Security Information and Event Management – erfüllt drei Kernaufgaben: sammeln, korrelieren, alarmieren. Bei der *Aggregation* werden Logs aus vielen Quellen zusammengeführt und in ein einheitliches Format gebracht. Bei der *Korrelation* verknüpft das SIEM einzelne Ereignisse anhand von Regeln zu aussagekräftigen Mustern. Und bei verdächtigen Mustern löst es eine *Alarmierung* aus.

Ein einfaches Beispiel: Fünfzig fehlgeschlagene Anmeldungen an einem Konto innerhalb weniger Minuten, gefolgt von einer erfolgreichen Anmeldung aus einem ungewöhnlichen Land, und kurz danach wird dasselbe Konto in die Gruppe der Administratoren aufgenommen. Jedes dieser Ereignisse stammt aus einer anderen Quelle und wäre für sich allein unauffällig. Das SIEM erkennt den Zusammenhang und schlägt Alarm.

Solche Erkennungsregeln heißen **Use Cases**: Detektionsregeln für konkrete Szenarien wie Brute-Force-Angriffe, verdächtige Rechteausweitung oder Datenabfluss. Sie werden fortlaufend nachgeschärft, um False Positives gezielt zu reduzieren. Zusätzlich lassen sich Informationen über aktuelle Bedrohungen einbinden (*Threat Intelligence*), etwa Listen bekannter schädlicher IP-Adressen. Für jeden Alarmtyp sollte ein *Playbook* festlegen, wie zu reagieren ist.

Gängige Werkzeuge

Am Markt gibt es eine Reihe etablierter SIEM-Lösungen. Bekannte Beispiele sind Splunk, das cloud-native Microsoft Sentinel, Graylog und der Elastic-Stack. Als Open-Source-Option ist Wazuh verbreitet, das SIEM-Funktionen mit Endpunkt-Überwachung verbindet. Welches Werkzeug passt, hängt von der vorhandenen Umgebung, der zu verarbeitenden Datenmenge und dem Budget ab – bei vielen Lösungen richten sich die Kosten nach dem Datenvolumen.

Log-Quellen und Aufbewahrung

Ein SIEM ist nur so gut wie seine Quellen. Zu den Kernquellen gehören das *Active Directory* (Anmeldungen, Konten- und Gruppenänderungen), die *Firewall* (Verbindungen und blockierter Verkehr), das *EDR* (Vorgänge auf den Endpunkten) und der *Proxy* (aufgerufene Webadressen). Nach und nach kommen weitere Quellen wie Mailsysteme, Cloud-Dienste und wichtige Anwendungen hinzu.

Damit Ereignisse aus verschiedenen Systemen richtig zueinander in Beziehung gesetzt werden können, müssen alle Uhren synchron laufen. Eine einheitliche **Zeitsynchronisation per NTP** auf allen Systemen ist deshalb Pflicht – schon wenige Minuten Abweichung machen eine Korrelation unzuverlässig. Bei der Aufbewahrung der Logs sind gesetzliche Fristen und der Datenschutz zu beachten, denn Logs enthalten häufig personenbezogene Daten.

> **Merke**
>
> Ohne zentrale Logs keine Erkennung, ohne synchrone Uhren keine Korrelation. Binden Sie zuerst die Kernquellen Active Directory, Firewall, EDR und Proxy an und sorgen Sie überall für korrekte Zeit per NTP.

Das SOC: das Team hinter den Werkzeugen

Ein SIEM erzeugt Alarme – bewerten und darauf reagieren müssen Menschen. Diese Aufgabe übernimmt das **Security Operations Center** (SOC). Kurz gesagt: Das SOC ist das Team, das SIEM ist das Werkzeug. Ein SOC ist meist in Stufen organisiert. Im *Tier 1* werden eingehende Alarme gesichtet und vorsortiert (Triage): Ist der Alarm echt, wie

dringend ist er? Bestätigte oder komplexe Fälle gehen an *Tier 2 und 3*. Dort wird tiefer analysiert, proaktiv nach Angreifern gesucht (Threat Hunting) und die Incident Response gesteuert.

Ein SOC kann rund um die Uhr überwachen. Für viele kleinere und mittlere Unternehmen ist ein eigenes 24/7-Team jedoch personell und finanziell nicht darstellbar. Sie nutzen deshalb häufig einen *Managed Security Service Provider* (MSSP), der den SOC-Betrieb als Dienstleistung übernimmt.

Incident Response und Alarmqualität

Kommt es zu einem Sicherheitsvorfall, folgt die **Incident Response** einem festen Ablauf: *vorbereiten*, *erkennen*, *eindämmen*, *beseitigen*, *wiederherstellen* und *lernen*. Die Vorbereitung entscheidet dabei über den Erfolg: Klare Rollen, Erreichbarkeiten und ein Notfallplan müssen vorab feststehen und geübt sein. Im Ernstfall gilt: erst eindämmen, dann beseitigen. Zuerst wird also die Ausbreitung gestoppt, etwa durch das Isolieren betroffener Systeme, bevor die Ursache entfernt wird. In der Nachbereitung wird geklärt, wie es zum Vorfall kam und was sich verbessern lässt.

Gerade die letzte Phase wird in der Praxis gern vernachlässigt, wenn der Betrieb wieder läuft. Dabei liefert sie die wertvollsten Erkenntnisse: Welche Erkennungsregel hat gefehlt, welche Log-Quelle war nicht angebunden, an welcher Stelle hat die Abstimmung zu lange gedauert? Die Antworten fließen zurück in die Vorbereitung und machen das Team beim nächsten Vorfall schneller.

Eine typische Gefahr im SOC-Alltag ist die **Alert-Fatigue**, die Alarmmüdigkeit. Zu viele Alarme sind so gefährlich wie zu wenige, denn wenn täglich Hunderte Meldungen eingehen, gehen die wirklich wichtigen im Rauschen unter. Regeln müssen deshalb gepflegt, Fehlalarme konsequent reduziert und Alarme nach Risiko priorisiert werden. Das Ziel ist nicht die maximale Menge an Alarmen, sondern ihre Qualität.

Aus der Praxis

Üben Sie den Notfallplan regelmäßig, zum Beispiel in einer Planspiel-Übung am Tisch: Ein angenommener Ransomware-Fall wird Schritt für Schritt durchgespielt. Dabei fällt schnell auf, wo Telefonnummern fehlen, Zuständigkeiten unklar sind oder Entscheidungswege zu lang dauern.

Das Wichtigste in Kürze

- Ein SIEM sammelt, korreliert und alarmiert.
- Das SOC ist das Team, das SIEM das Werkzeug – intern oder als Managed Service.
- Gute Log-Quellen und synchrone Uhren (NTP) sind die Grundlage.

- Incident Response ist ein geübter Prozess: erst eindämmen, dann beseitigen.
- Alarmqualität schlägt Alarmmenge.

BSI IT-Grundschutz & Compliance

Methodik, Schutzbedarf und der Bezug zu ISO 27001 & DSGVO

Nach diesem Kapitel können Sie …

- erklären, was der BSI IT-Grundschutz ist und wozu ein ISMS dient,
- die Methodik aus Strukturanalyse, Modellierung und Soll-Ist-Abgleich beschreiben,
- eine Schutzbedarfsfeststellung nach den Stufen normal, hoch und sehr hoch durchführen,
- das Zusammenspiel von IT-Grundschutz, ISO 27001 und DSGVO einordnen.

Was der BSI IT-Grundschutz ist

Die vorangegangenen Kapitel haben viele einzelne Schutzmaßnahmen vorgestellt – von der Firewall über die Mail-Security bis zum SIEM. Offen bleibt die Frage: Wie entscheidet ein Unternehmen systematisch, welche Maßnahmen es braucht, und wie stellt es sicher, dass nichts Wichtiges vergessen wird? Eine Antwort darauf liefert der **IT-Grundschutz**, ein bewährtes deutsches Rahmenwerk für Informationssicherheit. Herausgegeben wird es vom **BSI**, dem Bundesamt für Sicherheit in der Informationstechnik.

Der IT-Grundschutz besteht im Kern aus zwei Teilen: den BSI-Standards, die das Vorgehen beschreiben, und dem IT-Grundschutz-Kompendium mit praxisnahen **Bausteinen**, die konkrete Anforderungen enthalten. Gemeinsam bilden sie die Basis für ein **ISMS** – ein Informationssicherheits-Managementsystem. Ein ISMS ist kein Programm, sondern ein dauerhafter Rahmen aus Regeln, Verantwortlichkeiten und Abläufen, mit dem ein Unternehmen seine Informationssicherheit plant, umsetzt, überprüft und verbessert.

Die Methodik: von der Strukturanalyse zum Soll-Ist-Abgleich

Ein *Baustein* beschreibt ein typisches Objekt der IT-Landschaft – etwa einen Server, ein Netz, eine Anwendung, aber auch organisatorische Themen wie Personal oder Notfallmanagement – samt der Anforderungen, die für dieses Objekt gelten. Daraus ergibt sich die Kette *Bausteine → Anforderungen → Maßnahmen*.

Am Anfang steht die **Strukturanalyse**: Was ist überhaupt vorhanden? Erfasst werden Geschäftsprozesse, Anwendungen, IT-Systeme, Räume und Netzverbindungen sowie ihre Abhängigkeiten. Viele Unternehmen stellen dabei erstmals fest, wie viele Systeme sie tatsächlich betreiben. Anschließend folgt die **Modellierung**: Jedem erfassten Objekt

werden die passenden Bausteine zugeordnet. Im **Soll-Ist-Abgleich** wird schließlich geprüft, welche Anforderungen bereits erfüllt sind und wo Lücken bestehen. Diese Lücken bilden die Grundlage für einen Umsetzungsplan.

Damit der Einstieg nicht zu groß wird, kennt der IT-Grundschutz drei Vorgehensweisen. Die *Basis-Absicherung* setzt mit überschaubarem Aufwand die wichtigsten Anforderungen in der Breite um und eignet sich gut als Einstieg. Die *Standard-Absicherung* ist das empfohlene, umfassende Vorgehen für die gesamte Organisation. Die *Kern-Absicherung* konzentriert sich zunächst auf besonders kritische Bereiche, die sogenannten Kronjuwelen, und schützt diese in der Tiefe.

Die Schutzbedarfsfeststellung

Ein zentraler Schritt ist die **Schutzbedarfsfeststellung**. Für jedes Objekt wird gefragt: Wie schützenswert ist es hinsichtlich *Vertraulichkeit*, *Integrität* und *Verfügbarkeit* – also der CIA-Triade aus dem ersten Kapitel? Die Bewertung erfolgt getrennt für jedes dieser drei Ziele, denn ein öffentlicher Webauftritt kann etwa bei der Vertraulichkeit unkritisch, bei der Verfügbarkeit aber sehr wichtig sein.

Der IT-Grundschutz unterscheidet drei Stufen. Beim Schutzbedarf **normal** sind die Schadensauswirkungen begrenzt und überschaubar. Bei **hoch** können sie beträchtlich sein. Bei **sehr hoch** können sie ein existenziell bedrohliches, katastrophales Ausmaß erreichen. Je höher der Schutzbedarf, desto strengere Maßnahmen sind erforderlich. Für Objekte mit hohem oder sehr hohem Schutzbedarf reichen die Standardanforderungen oft nicht aus, sodass eine zusätzliche Risikoanalyse durchgeführt wird.

Schutzbedarf	Schadensauswirkung	Folge
normal	Begrenzt und überschaubar	Standardanforderungen der Bausteine
hoch	Beträchtlich	Zusätzliche Anforderungen, ggf. Risikoanalyse
sehr hoch	Existenzbedrohend	Strenge Maßnahmen und Risikoanalyse

> **Merke**
>
> Der Schutzbedarf wird für jedes Objekt getrennt nach Vertraulichkeit, Integrität und Verfügbarkeit in den Stufen normal, hoch und sehr hoch bewertet. Er bestimmt, wie streng die Maßnahmen ausfallen müssen.

Der Bezug zu ISO 27001

Die **ISO/IEC 27001** ist der international anerkannte Standard für Informationssicherheits-Managementsysteme. Sie beschreibt, welche Anforderungen ein ISMS erfüllen muss, lässt aber weitgehend offen, wie die konkreten Maßnahmen aussehen. Die ISO-Norm ist damit generischer, der IT-Grundschutz deutlich konkreter: Er liefert für viele typische Objekte bereits ausformulierte Anforderungen.

Beide Ansätze passen gut zusammen. Wer den IT-Grundschutz umsetzt, kann darauf aufbauend eine *ISO-27001-Zertifizierung auf der Basis von IT-Grundschutz* erlangen, die vom BSI vergeben wird. Gemeinsam ist beiden Ansätzen das Ziel eines **kontinuierlichen Verbesserungsprozesses**: Sicherheit ist kein einmaliges Projekt, sondern wird regelmäßig überprüft und an neue Bedrohungen und Veränderungen angepasst.

Dieser Kreislauf wird häufig mit dem PDCA-Zyklus beschrieben: *Plan* (Ziele und Maßnahmen planen), *Do* (umsetzen), *Check* (Wirksamkeit prüfen, etwa durch interne Audits) und *Act* (Verbesserungen ableiten). Danach beginnt der Zyklus von vorn – zum Beispiel, wenn neue Systeme hinzukommen, sich Geschäftsprozesse ändern oder neue Angriffstechniken bekannt werden.

Der Bezug zur DSGVO

Auch der Datenschutz stellt Anforderungen an die IT-Sicherheit. Die **Datenschutz-Grundverordnung** (DSGVO) verlangt in Artikel 32 „geeignete technische und organisatorische Maßnahmen“, kurz **TOM**, um personenbezogene Daten angemessen zu schützen. Dazu gehören ausdrücklich die Vertraulichkeit, Integrität und Verfügbarkeit der Systeme und Daten.

Hier zeigt sich ein praktischer Vorteil: Viele Sicherheitsmaßnahmen, die ohnehin im Rahmen des IT-Grundschutzes umgesetzt werden – Zugriffsrechte, Verschlüsselung, Backups, Protokollierung –, dienen zugleich als Nachweis für die TOM-Pflicht. Entscheidend ist im Ernstfall die *Dokumentation*. Kommt es zu einer Datenschutzverletzung, muss das Unternehmen belegen können, welche Maßnahmen getroffen waren.

Rollen, Nachweise und Prüfungsrelevanz

Die Verantwortung für die Informationssicherheit trägt die **Leitung** des Unternehmens. Sie kann Aufgaben delegieren, nicht aber die Verantwortung. Restrisiken, die nicht weiter reduziert werden, müssen von der Leitung bewusst akzeptiert und dokumentiert werden. Für die Koordination im Alltag wird üblicherweise ein **Informationssicherheitsbeauftragter** (ISB) benannt, der den Sicherheitsprozess steuert, berät und an die Leitung berichtet. Die Dokumentation dient als Nachweis bei Audits und als Grundlage für den kontinuierlichen Verbesserungsprozess.

Prüfungsrelevant

Für Prüfungen sollten Sie die CIA-Triade sicher erklären können, die Grundbegriffe ISMS, Schutzbedarf und Baustein beherrschen und das Zusammenspiel von BSI IT-Grundschutz, ISO 27001 und DSGVO beschreiben können.

Das Wichtigste in Kürze

- Der IT-Grundschutz ist ein praxisnaher Weg zu einem ISMS.
- Methodik: Strukturanalyse, Modellierung, Soll-Ist-Abgleich.
- Schutzbedarf normal, hoch oder sehr hoch – getrennt nach Vertraulichkeit, Integrität und Verfügbarkeit.
- IT-Grundschutz kann zur ISO-27001-Zertifizierung führen.
- Sicherheitsmaßnahmen erfüllen zugleich die TOM-Pflicht der DSGVO.

LB-08

Cloud

1. Cloud-Grundlagen & Modelle
2. Microsoft 365 – Administration
3. Microsoft Azure – Kernprodukte
4. AWS – Kernprodukte
5. Google Cloud & Workspace
6. Cloud-Migration
7. Infrastructure as Code
8. Cloud-Security & FinOps

Cloud-Grundlagen & Modelle

IaaS/PaaS/SaaS, Bereitstellung & geteilte Verantwortung

Nach diesem Kapitel können Sie …

- erklären, was Cloud Computing ist und welche Vorteile und Grenzen es hat,
- die Service-Modelle IaaS, PaaS und SaaS unterscheiden,
- Public, Private und Hybrid Cloud als Bereitstellungsmodelle einordnen,
- das Modell der geteilten Verantwortung und die Grundzüge des Datenschutzes anwenden,
- das Kostenmodell der Cloud und typische Kostenfallen beschreiben.

Was „Cloud“ eigentlich bedeutet

Unter **Cloud Computing** versteht man die Nutzung von IT-Ressourcen als Dienst über das Internet. Statt eigene Server, Speichersysteme und Netzwerkkomponenten zu kaufen, aufzustellen und zu warten, mietet ein Unternehmen diese Ressourcen bei einem Anbieter. Rechenleistung, Speicherplatz, Datenbanken oder komplette Anwendungen stehen dann auf Abruf bereit – meist innerhalb von Minuten statt Wochen.

Zwei Eigenschaften machen die Cloud besonders attraktiv. Erstens wird *nach Verbrauch* bezahlt: Wer eine virtuelle Maschine nur drei Tage braucht, bezahlt auch nur drei Tage. Zweitens lassen sich Ressourcen schnell *hoch- und herunterskalieren*. Ein Onlineshop, der im Weihnachtsgeschäft ein Vielfaches an Besuchern bewältigen muss, kann kurzfristig zusätzliche Kapazität dazunehmen und sie danach wieder abgeben.

Trotzdem gilt: Nicht alles gehört in die Cloud. Anwendungen mit sehr gleichmäßiger Dauerlast, strengen Latenzanforderungen oder besonderen rechtlichen Vorgaben können im eigenen Rechenzentrum günstiger oder einfacher zu betreiben sein. Die Entscheidung sollte deshalb immer pro Anwendung getroffen werden.

Die Service-Modelle: IaaS, PaaS und SaaS

Cloud-Dienste werden nach der Frage eingeteilt, wie viel der Anbieter übernimmt und wie viel beim Kunden bleibt. Bei **IaaS** (Infrastructure as a Service) stellt der Anbieter virtuelle Maschinen, Speicher und Netzwerke bereit. Das Betriebssystem, seine Updates, die installierte Software und die Daten verwalten Sie selbst. IaaS fühlt sich deshalb ähnlich an wie ein eigener Server – nur dass die Hardware woanders steht.

Bei **PaaS** (Platform as a Service) geht der Anbieter einen Schritt weiter und liefert eine fertige Plattform, etwa eine verwaltete Datenbank oder eine Umgebung für Webanwendungen. Um Betriebssystem und Laufzeitumgebung müssen Sie sich nicht mehr kümmern; Sie bringen nur noch Ihre Anwendung und Ihre Daten mit.

Bei **SaaS** (Software as a Service) nutzen Sie schließlich eine fertige Anwendung über den Browser oder eine App, zum Beispiel Microsoft 365 für E-Mail und Office. Technik, Betrieb und Updates liegen vollständig beim Anbieter; Ihre Aufgabe ist vor allem die Verwaltung von Benutzern, Rechten und Einstellungen.

Modell	Der Anbieter stellt bereit	Sie verwalten
IaaS	Hardware, Virtualisierung, Netzwerk	Betriebssystem, Software, Daten
PaaS	Zusätzlich Betriebssystem und Plattform	Anwendung und Daten
SaaS	Die komplette Anwendung	Benutzer, Rechte, Konfiguration, Daten

Bereitstellungsmodelle: Public, Private, Hybrid

Unabhängig vom Service-Modell stellt sich die Frage, *wo* und *für wen* die Cloud betrieben wird. Die **Public Cloud** läuft in den Rechenzentren eines Anbieters; die Hardware wird von vielen Kunden gemeinsam genutzt, die Kunden sind aber logisch streng voneinander getrennt. Eine **Private Cloud** steht dagegen exklusiv einer einzigen Organisation zur Verfügung – im eigenen Rechenzentrum oder bei einem Dienstleister. Sie bietet mehr Kontrolle, verlangt aber auch mehr eigenen Aufwand.

Die **Hybrid Cloud** kombiniert beide Ansätze. Typisch ist, dass sensible oder schwer migrierbare Systeme lokal bleiben, während etwa E-Mail, Backups oder neue Anwendungen in der Public Cloud laufen. In der Praxis arbeiten viele Unternehmen, oft ohne es so zu nennen, bereits hybrid.

Geteilte Verantwortung und Datenschutz

Ein zentrales Prinzip der Cloud ist das **Modell der geteilten Verantwortung** (Shared Responsibility Model). Der Anbieter sichert die Cloud selbst ab: Rechenzentrum, Stromversorgung, Hardware und die grundlegende Virtualisierung. Sie sichern ab, was Sie in der Cloud betreiben – also Konfiguration, Zugänge, Daten und, je nach Service-Modell, auch Betriebssystem und Anwendungen.

> **Merke**
>
> Der Anbieter sichert die Cloud – Sie sichern, was Sie darin betreiben. Die häufigste Ursache für Sicherheitsvorfälle in der Cloud ist nicht ein Fehler des Anbieters, sondern eine Fehlkonfiguration auf Kundenseite, etwa ein öffentlich zugänglicher Speicher.

Eng damit verbunden ist der **Datenschutz**. Sobald personenbezogene Daten in der Cloud verarbeitet werden, gilt die DSGVO. Prüfen Sie, in welchem Land die Daten gespeichert werden, und schließen Sie mit dem Anbieter einen Vertrag zur Auftragsverarbeitung ab. Technisch gehören Verschlüsselung und eine saubere Zugriffskontrolle dazu: Nur wer die Daten wirklich braucht, sollte sie auch sehen können.

Skalierung und Elastizität

Ein großer Vorteil der Cloud ist ihre **Elastizität**: Ressourcen wachsen und schrumpfen mit dem tatsächlichen Bedarf. Dabei unterscheidet man zwei Richtungen. Bei der *vertikalen Skalierung* (scale up) bekommt eine einzelne Maschine mehr CPU oder Arbeitsspeicher. Das ist einfach, stößt aber irgendwann an Grenzen und erfordert oft einen Neustart. Bei der *horizontalen Skalierung* (scale out) werden zusätzliche Maschinen parallel betrieben, die sich die Last teilen. Diese Variante eignet sich besonders für Lastspitzen.

Mit **Auto-Scaling** lässt sich die horizontale Skalierung automatisieren: Steigt die Last über einen festgelegten Wert, startet die Plattform weitere Instanzen; sinkt sie wieder, werden diese heruntergefahren. Gerade dieses Herunterfahren spart Geld, denn bezahlt wird nur, was läuft.

Das Kostenmodell verstehen

Die Cloud arbeitet nach dem Prinzip **Pay-as-you-go**: Abgerechnet werden nur die tatsächlich genutzten Ressourcen. Für die Buchhaltung bedeutet das einen Wechsel von *CapEx* (Investitionen in eigene Hardware) zu *OpEx* (laufende Betriebskosten). Für planbare Dauerlast bieten die Anbieter *Reservierungen* an, bei denen Sie sich für einen längeren Zeitraum binden und dafür einen deutlichen Rabatt erhalten. *Spot-Instanzen* nutzen freie Kapazitäten des Anbieters besonders günstig, können aber kurzfristig entzogen werden und eignen sich daher nur für unterbrechbare Aufgaben.

Die Flexibilität hat eine Kehrseite: Kosten entstehen leicht unbemerkt. Häufige Ursachen sind vergessene Testmaschinen, nicht mehr benötigte Speicher-Snapshots und der *Datenabfluss* (Egress) – also Daten, die aus der Cloud heraus übertragen werden und oft separat berechnet werden.

Aus der Praxis

Richten Sie Budgets, Kostenalarme und ein Tagging-Konzept gleich zu Beginn ein. Mit Tags (Schlagworten) wie Kostenstelle, Projekt oder Umgebung lässt sich später jede Ressource einem Verantwortlichen zuordnen – nachträglich ist das deutlich mühsamer.

Die großen Anbieter

Den Markt prägen die sogenannten **Hyperscaler**: Amazon Web Services (AWS), Microsoft Azure und Google Cloud. Sie betreiben weltweit Rechenzentren und bieten einen riesigen Katalog an Diensten – von einfachen virtuellen Maschinen bis zu KI-Plattformen. Daneben gibt es regionale und europäische Anbieter mit Rechenzentren in der EU, die für Unternehmen mit strengen Datenschutzanforderungen interessant sein können.

Manche Unternehmen verteilen ihre Dienste bewusst auf mehrere Anbieter (**Multi-Cloud**). Das erhöht die Flexibilität und verringert die Abhängigkeit, macht Betrieb, Sicherheit und Abrechnung aber komplexer. Umgekehrt führt die starke Nutzung anbieterspezifischer Dienste zum **Vendor-Lock-in**: Ein späterer Wechsel wird teuer und aufwendig. Dieses Risiko lässt sich nicht immer vermeiden, sollte aber bewusst abgewogen werden.

Das Wichtigste in Kürze

- Cloud bedeutet IT-Ressourcen als Dienst, bezahlt nach Verbrauch.
- IaaS, PaaS und SaaS unterscheiden sich darin, wie viel Sie selbst verwalten.
- Public, Private und Hybrid beschreiben, wo und für wen die Cloud betrieben wird.
- Geteilte Verantwortung: Fehlkonfiguration ist das Hauptrisiko, die DSGVO gilt auch in der Cloud.
- Budgets, Alarme und Tags von Anfang an verhindern Kostenfallen.

Microsoft 365 – Administration

Tenant, Dienste und Sicherheit verwalten

Nach diesem Kapitel können Sie …

- erklären, was ein Microsoft-365-Tenant ist und wie er verwaltet wird,
- die Kerndienste Exchange Online, SharePoint, OneDrive und Teams einordnen,
- einen Tenant mit MFA, Conditional Access und durchdachten Admin-Rollen absichern,
- die wichtigsten Betriebsaufgaben von Lizenzen bis Backup und Compliance benennen.

Was Microsoft 365 ist

Microsoft 365 ist eine Cloud-Suite, die die bekannten Office-Anwendungen wie Word, Excel und Outlook mit Cloud-Diensten für E-Mail, Dateiablage und Zusammenarbeit verbindet. Aus Sicht des vorherigen Kapitels handelt es sich um ein klassisches SaaS-Angebot: Microsoft betreibt Server, Software und Updates, Ihr Unternehmen nutzt die Dienste und verwaltet Benutzer, Rechte und Einstellungen.

Jede Organisation erhält dafür einen eigenen **Tenant**, auf Deutsch *Mandant*. Der Tenant ist gewissermaßen die Organisation in der Cloud: Er enthält alle Benutzerkonten, Gruppen, Postfächer, Dateien und Einstellungen und ist strikt von anderen Kunden getrennt. Die Identitäten – also Benutzerkonten und Anmeldungen – werden in **Entra ID** verwaltet, dem Verzeichnisdienst, der früher Azure Active Directory hieß. Lizenziert wird pro Benutzer.

Tenant und Admin Center

Verwaltet wird ein Tenant über die **Admin Center** im Browser. Das zentrale Microsoft-365-Admin-Center dient für alltägliche Aufgaben wie das Anlegen von Benutzern oder das Zuweisen von Lizenzen. Für einzelne Dienste gibt es spezialisierte Admin Center, etwa für Exchange, Teams, SharePoint, Entra ID oder die Compliance-Funktionen. Viele Aufgaben lassen sich außerdem per PowerShell automatisieren, was bei größeren Umgebungen viel Zeit spart.

Wichtig sind die **Mandanteneinstellungen**: Sie legen zentral fest, was in der gesamten Organisation erlaubt ist – zum Beispiel, ob Dateien mit externen Personen geteilt werden dürfen oder ob Gäste in Teams eingeladen werden können. Diese Einstellungen sollten bewusst gesetzt und dokumentiert werden, statt die Voreinstellungen ungeprüft zu übernehmen.

Die Kerndienste im Überblick

Exchange Online ist der E-Mail- und Kalenderdienst. Er stellt Postfächer, gemeinsame Kalender, Verteilerlisten und freigegebene Postfächer bereit und enthält Schutzfunktionen gegen Spam und Schadsoftware. **SharePoint** ist die Plattform für Dokumente und Zusammenarbeit im Team: Abteilungen oder Projekte erhalten eigene Websites mit Dokumentbibliotheken, Versionsverlauf und Berechtigungen. **OneDrive** ist dagegen der persönliche Dateispeicher jedes Benutzers, vergleichbar mit einem Home-Laufwerk, das auf allen Geräten verfügbar ist.

Teams schließlich bündelt Chat, Besprechungen, Telefonie und Zusammenarbeit. Technisch ist Teams kein ganz eigenständiger Dienst, sondern sitzt auf SharePoint und Exchange auf: Dateien eines Teams liegen in einer SharePoint-Website, Kalender und manche Daten in Exchange. Dieses Zusammenspiel zu kennen, hilft bei der Fehlersuche und bei Fragen zu Berechtigungen und Aufbewahrung.

Identität und Sicherheit

Da Microsoft 365 aus dem Internet erreichbar ist, entscheidet die Anmeldung über die Sicherheit des gesamten Tenants. Die beiden wichtigsten Schutzmaßnahmen sind **Multi-Faktor-Authentifizierung (MFA)** und **Conditional Access**. MFA verlangt neben dem Passwort einen zweiten Nachweis, etwa eine Bestätigung in einer Authenticator-App. Sie sollte möglichst für alle Konten erzwungen werden, nicht nur für Administratoren.

Conditional Access – bedingter Zugriff – geht noch weiter: Über Richtlinien wird festgelegt, unter welchen Bedingungen ein Zugriff erlaubt ist. So kann der Zugriff etwa nur von verwalteten Firmengeräten, aus bestimmten Ländern oder nur mit MFA gestattet werden. Veraltete Anmeldeverfahren, die keine MFA unterstützen, lassen sich damit ebenfalls blockieren.

> **Merke**
>
> MFA und Conditional Access sind die wirksamsten Hebel zur Absicherung eines Microsoft-365-Tenants. Ein Tenant ohne erzwungene MFA ist mit einem einzigen erbeuteten Passwort angreifbar.

Rollen und Delegation

Nicht jede Person in der IT braucht die Rolle des *globalen Administrators*, die praktisch alles darf. Microsoft 365 bietet zahlreiche spezifische Admin-Rollen, etwa für Exchange, Teams oder die Benutzerverwaltung. Vergeben Sie Rechte nach dem Prinzip der **minimalen Rechte**: Jede Person erhält genau die Rolle, die sie für ihre Aufgaben benötigt.

Noch besser ist es, hohe Rechte nur zeitlich begrenzt zu vergeben. Mit **Privileged Identity Management** (PIM) werden Admin-Rollen nicht dauerhaft zugewiesen, sondern bei Bedarf für eine begrenzte Zeit aktiviert – *Just-in-Time*. Die Aktivierung kann eine Begründung oder eine Freigabe erfordern und wird protokolliert. Zusätzlich sollten Sie mindestens zwei **Notfallkonten** (Break-Glass-Konten) anlegen, die nicht an einzelne Personen gebunden sind, besonders stark gesichert sind und nur genutzt werden, wenn die reguläre Anmeldung nicht mehr funktioniert.

Betrieb, Datensicherheit und Compliance

Zum laufenden Betrieb gehört zunächst die **Lizenzverwaltung**. Microsoft bietet verschiedene Pläne an – grob gesagt Business-Pläne für kleinere Organisationen und Enterprise-Pläne mit erweiterten Sicherheits- und Compliance-Funktionen. Welche Pläne aktuell angeboten werden und was sie enthalten, ändert sich regelmäßig, daher lohnt ein Blick in die aktuelle Übersicht des Herstellers. Praktisch bewährt hat sich die *gruppenbasierte Lizenzierung*: Wer Mitglied einer Gruppe wird, erhält automatisch die passende Lizenz. Ungenutzte Lizenzen, etwa von ausgeschiedenen Mitarbeitenden, sollten Sie regelmäßig erkennen und freigeben, und Zusatzpakete gezielt statt pauschal für alle kaufen.

Ebenfalls zum Betrieb gehört der Blick auf die **Service Health**, also den Dienststatus. Bei Störungen zeigt Microsoft dort an, welche Dienste betroffen sind – das erspart unnötige Fehlersuche im eigenen Netz.

Aus der Praxis

Denken Sie an ein Backup Ihrer Microsoft-365-Daten. Der Anbieter garantiert die Verfügbarkeit seiner Dienste, nicht aber die Wiederherstellung von Daten, die Benutzer versehentlich gelöscht oder Angreifer verschlüsselt haben. Papierkorb und Versionsverlauf ersetzen kein eigenständiges Backup.

Für die **Datensicherheit** stehen weitere Werkzeuge bereit. *Data Loss Prevention* (DLP) erkennt sensible Inhalte wie Kreditkarten- oder Personalnummern und verhindert, dass sie ungewollt per E-Mail oder Freigabe nach außen gelangen. *Vertraulichkeitskennzeichnungen* (Sensitivity Labels) markieren Dokumente etwa als „intern“ oder „vertraulich“ und können sie automatisch verschlüsseln. *Aufbewahrungsrichtlinien* sorgen dafür, dass Daten so lange aufbewahrt werden, wie es gesetzliche oder interne Vorgaben verlangen, und *eDiscovery* ermöglicht es, im Rechtsfall gezielt nach Inhalten zu suchen. Alarme im Compliance-Portal weisen auf auffällige Aktivitäten hin und sollten regelmäßig gesichtet werden.

Das Wichtigste in Kürze

- Microsoft 365 besteht aus einem Tenant und Cloud-Diensten, Identitäten liegen in Entra ID.
- Kerndienste sind Exchange Online, SharePoint, OneDrive und Teams.
- MFA und Conditional Access sind die wichtigsten Schutzmaßnahmen.
- Admin-Rechte minimal und möglichst zeitlich begrenzt vergeben, Notfallkonten absichern.
- Backup, Lizenzpflege und Compliance-Richtlinien gehören fest zum Betrieb.

Microsoft Azure – Kernprodukte

Die wichtigsten Bausteine der Azure-Cloud

Nach diesem Kapitel können Sie …

- die Grunddienste von Azure für Rechenleistung, Netzwerk und Speicher einordnen,
- Ressourcen sinnvoll in Management Groups, Subscriptions und Resource Groups organisieren,
- Werkzeuge für Sicherheit, Governance und Betrieb benennen,
- die Kosten einer Azure-Umgebung aktiv steuern.

Azure im Überblick

Microsoft Azure ist die Public Cloud von Microsoft. Sie bietet Dienste aus allen Service-Modellen – von virtuellen Maschinen (IaaS) über verwaltete Datenbanken und Webplattformen (PaaS) bis zu speziellen Diensten für Analyse und künstliche Intelligenz. Die Ressourcen laufen in *Regionen*, also Rechenzentrumsstandorten auf der ganzen Welt, darunter auch mehrere in Europa. Abgerechnet wird nach Verbrauch.

Ein besonderer Vorteil von Azure ist die enge Verzahnung mit Microsoft 365 und Entra ID. Unternehmen, die ihre Benutzerkonten bereits in Entra ID verwalten, können dieselben Identitäten für den Zugriff auf Azure-Ressourcen nutzen. Für viele Organisationen mit Windows-Servern und Microsoft-Lizenzen ist Azure deshalb der naheliegende Einstieg in die Cloud.

Compute und Netzwerk

Der Grundbaustein für Rechenleistung sind **Azure Virtual Machines** (VMs). Sie funktionieren wie virtuelle Server im eigenen Rechenzentrum: Sie wählen Größe, Betriebssystem und Speicher, und das Betriebssystem verwalten Sie selbst. Jede VM benötigt ein Netzwerk, und dieses heißt in Azure **Virtual Network** (VNet). Ein VNet ist ein privates, logisch abgetrenntes Netz mit eigenem Adressbereich, das in Subnetze aufgeteilt wird – ganz ähnlich wie ein klassisches Firmennetz mit VLANs.

Den Datenverkehr regeln **Network Security Groups** (NSGs). Sie wirken wie virtuelle Firewalls und legen über Regeln fest, welche Verbindungen zu einem Subnetz oder einer VM erlaubt sind. Für die Anbindung an das eigene Rechenzentrum stehen zwei Wege zur Verfügung: ein *VPN-Gateway*, das einen verschlüsselten Tunnel über das Internet aufbaut, oder *ExpressRoute*, eine dedizierte Leitung über einen Netzbetreiber, die nicht über das öffentliche Internet läuft.

Speicher in Azure

Für unstrukturierte Daten wie Dateien, Bilder, Backups oder Protokolle ist **Blob Storage** zentral. „Blob“ steht für *Binary Large Object*; gemeint ist ein Objektspeicher, der praktisch beliebig große Datenmengen aufnimmt. Die Festplatten der VMs werden dagegen als **Managed Disks** bereitgestellt, deren Verwaltung Azure übernimmt.

Über *Zugriffsebenen* (Tiers) steuern Sie die Kosten: *Hot* für häufig genutzte Daten, *Cool* für selten genutzte und *Archive* für Daten, die kaum abgerufen werden. Je „kälter“ die Ebene, desto günstiger der Speicher, desto teurer und langsamer aber der Abruf. Mit den *Redundanzoptionen* legen Sie die Ausfallsicherheit fest: *LRS* (lokal redundant) hält mehrere Kopien innerhalb eines Rechenzentrums, *GRS* (georedundant) repliziert die Daten zusätzlich in eine weit entfernte zweite Region.

Ressourcen organisieren und regeln

In einer wachsenden Azure-Umgebung entsteht schnell Unübersichtlichkeit. Azure gibt deshalb eine Hierarchie vor, die von oben nach unten aufgebaut ist: **Management Groups** fassen mehrere Abonnements zusammen, **Subscriptions** (Abonnements) bilden die Abrechnungs- und Verwaltungseinheit, und **Resource Groups** bündeln zusammengehörige Ressourcen, etwa alle Komponenten einer Anwendung. Ergänzt wird diese Struktur durch *Tags* für Kostenstellen und Zuständigkeiten sowie durch einheitliche Namenskonventionen, an denen man Zweck, Umgebung und Region einer Ressource auf einen Blick erkennt.

Auf dieser Hierarchie baut die **Governance** auf. Mit **Azure Policy** werden Regeln automatisch durchgesetzt – zum Beispiel, dass Ressourcen nur in bestimmten europäischen Regionen angelegt werden dürfen oder dass jede Ressource ein Kostenstellen-Tag tragen muss. Mehrere Richtlinien lassen sich zu Initiativen bündeln und gemeinsam zuweisen. *Resource Locks* schützen wichtige Ressourcen vor versehentlichem Löschen oder Ändern.

> **Merke**
>
> Rechte (RBAC) und Richtlinien (Azure Policy) vererben sich entlang der Hierarchie nach unten. Was auf einer Management Group festgelegt wird, gilt für alle darunterliegenden Subscriptions und Ressourcen. Eine durchdachte Struktur spart deshalb später viel Einzelarbeit.

Sicherheit und Betrieb

Für Zugriffsrechte nutzt Azure **RBAC** (Role-Based Access Control). Statt Personen einzeln Rechte zu geben, weisen Sie Rollen wie „Leser“, „Mitwirkender“ oder „Besitzer“ auf einer bestimmten Ebene zu. So lässt sich sehr fein steuern, wer welche Ressourcen

sehen oder verändern darf. Passwörter, API-Schlüssel und Zertifikate gehören in den **Key Vault**, einen geschützten Tresor, aus dem Anwendungen ihre Geheimnisse abrufen, statt sie im Code oder in Konfigurationsdateien zu speichern.

Für den Betrieb sind drei weitere Dienste wichtig. **Azure Backup** sichert VMs, Datenbanken und Dateien, **Azure Site Recovery** repliziert ganze Systeme in eine andere Region und ermöglicht im Katastrophenfall (Disaster Recovery) den schnellen Wiederanlauf. **Azure Monitor** sammelt Metriken und Logs aller Ressourcen und löst bei Auffälligkeiten Alarme aus.

Beim Netzwerk gelten zusätzliche Regeln: NSGs schützen einzelne Subnetze, die *Azure Firewall* sorgt für eine zentrale Kontrolle des Datenverkehrs. Dienste wie Datenbanken oder Speicher sollten über *Private Endpoints* nur aus dem eigenen VNet erreichbar sein statt über eine öffentliche Adresse. Für Wartungszugriffe auf VMs bietet sich *Azure Bastion* an, damit RDP- und SSH-Ports nicht offen ins Internet zeigen. Öffentlich erreichbare Dienste lassen sich zusätzlich mit DDoS-Schutz absichern.

Aus der Praxis

Ein offener RDP-Port im Internet wird innerhalb kurzer Zeit von automatisierten Angriffswerkzeugen gefunden und mit Anmeldeversuchen überzogen. Nutzen Sie für Administrationszugriffe immer Bastion, ein VPN oder vergleichbare Lösungen – niemals eine öffentliche IP mit offenem Verwaltungsport.

Kosten im Griff behalten

Ohne Kontrolle laufen die Kosten in der Cloud schnell davon. Azure bietet dafür **Cost Management**, mit dem Sie Ausgaben nach Subscription, Resource Group oder Tag auswerten. Legen Sie *Budgets* mit Alarmen fest, damit Sie bei Überschreitungen frühzeitig gewarnt werden. Schalten Sie ungenutzte Ressourcen ab oder löschen Sie sie – eine gestoppte, aber nicht freigegebene VM kann weiterhin Kosten verursachen, etwa für Datenträger und öffentliche IP-Adressen. Für Systeme mit gleichmäßiger Dauerlast lohnen sich *Reservierungen*, bei denen Sie sich über einen längeren Zeitraum binden und dafür deutlich günstigere Preise erhalten.

Das Wichtigste in Kürze

- VMs und VNet mit NSGs bilden die Basis, VPN oder ExpressRoute binden das eigene Netz an.
- Blob Storage und Managed Disks speichern Daten, Tiers und Redundanz steuern Kosten und Sicherheit.
- Management Groups, Subscriptions und Resource Groups ordnen die Umgebung, Azure Policy regelt sie.

- Key Vault, Backup, Site Recovery, Monitor und RBAC sind Kernwerkzeuge für den Betrieb.
- Kosten aktiv steuern: Budgets, Alarme, Abschalten und Reservierungen.

AWS – Kernprodukte

Die zentralen Dienste von Amazon Web Services

Nach diesem Kapitel können Sie …

- den Aufbau von AWS mit Regionen und Availability Zones erklären,
- die Kerndienste EC2, VPC, S3, RDS, IAM, Lambda, CloudWatch und Route 53 einordnen,
- mehrere AWS-Konten mit AWS Organizations sinnvoll strukturieren,
- Grundregeln für Netzwerk, Berechtigungen, Kosten und Protokollierung anwenden.

AWS im Überblick

Amazon Web Services (AWS) gilt seit Jahren als größter Anbieter im Public-Cloud-Markt und bietet ein außergewöhnlich breites Portfolio – von einfachen virtuellen Servern bis zu Diensten für Datenanalyse, Internet der Dinge oder maschinelles Lernen. Abgerechnet wird nach dem Prinzip *Pay-as-you-go*, also nach tatsächlicher Nutzung.

Die Infrastruktur ist in **Regionen** gegliedert, etwa eine Region in Frankfurt. Jede Region besteht aus mehreren **Availability Zones**: voneinander getrennten Rechenzentren mit eigener Strom- und Netzanbindung, die nur wenige Kilometer auseinanderliegen. Wer eine Anwendung auf mehrere Availability Zones verteilt, bleibt erreichbar, auch wenn ein ganzes Rechenzentrum ausfällt. Diese Unterscheidung ist grundlegend für jede hochverfügbare Architektur in AWS.

Compute und Netzwerk

Virtuelle Server heißen in AWS **EC2-Instanzen** (Elastic Compute Cloud). Sie wählen einen Instanztyp, der CPU, Arbeitsspeicher und Netzwerkleistung festlegt, sowie ein Betriebssystem-Abbild. Das private Netzwerk, in dem die Instanzen laufen, ist die **VPC** (Virtual Private Cloud). Sie wird in Subnetze aufgeteilt, die jeweils einer Availability Zone zugeordnet sind.

Man unterscheidet *öffentliche Subnetze*, die über ein Internet-Gateway direkt erreichbar sind – etwa für einen Load Balancer –, und *private Subnetze* für Anwendungsserver und Datenbanken. Damit Systeme in privaten Subnetzen trotzdem Updates aus dem Internet laden können, nutzt man ein *NAT Gateway*, das nur ausgehende Verbindungen zulässt. Für die Elastizität sorgt **Auto Scaling**, das je nach Last Instanzen hinzufügt oder entfernt, und ein **Load Balancer** verteilt die eingehenden Anfragen auf die verfügbaren Instanzen.

Den Datenverkehr steuern zwei Mechanismen. *Security Groups* sind an Instanzen gebunden und *zustandsbehaftet*: Ist eine eingehende Verbindung erlaubt, wird die Antwort automatisch zugelassen. *Network ACLs* (NACLs) wirken dagegen auf ganze Subnetze und sind zustandslos – Hin- und Rückweg müssen jeweils eigens erlaubt werden.

Merkmal	Security Group	Network ACL
Wirkt auf	Einzelne Instanzen	Ganze Subnetze
Zustand	Zustandsbehaftet	Zustandslos
Typischer Einsatz	Feine Regeln pro Dienst	Grobe zusätzliche Schutzschicht

Speicher und Datenbanken

S3 (Simple Storage Service) ist der Objektspeicher von AWS und einer der am weitesten verbreiteten Cloud-Dienste überhaupt. Daten werden als Objekte in sogenannten *Buckets* abgelegt, von Website-Dateien über Backups bis zu großen Datenbeständen für Analysen. Über verschiedene *Storage-Klassen* passen Sie die Kosten an die Nutzung an: Häufig genutzte Daten liegen in der Standardklasse, selten genutzte oder archivierte Daten in günstigeren Klassen mit längeren Abrufzeiten.

RDS (Relational Database Service) stellt verwaltete relationale Datenbanken bereit, etwa mit MySQL, PostgreSQL oder Microsoft SQL Server. AWS übernimmt dabei Installation, Updates, automatische Backups und auf Wunsch die Replikation in eine zweite Availability Zone. Nutzen Sie Backups und Snapshots konsequent – auch in der Cloud können Daten durch Fehlbedienung oder Angriffe verloren gehen.

Weitere Kerndienste

Vier weitere Dienste begegnen Ihnen in fast jeder AWS-Umgebung. **IAM** (Identity and Access Management) verwaltet Benutzer, Gruppen, Rollen und Berechtigungen. **Lambda** ermöglicht *Serverless*-Funktionen: Sie laden nur Code hoch, der bei einem Ereignis ausgeführt wird, ohne dass Sie einen Server betreiben müssen; abgerechnet wird pro Ausführung. **CloudWatch** sammelt Metriken und Logs und löst Alarme aus. **Route 53** ist der DNS-Dienst von AWS, der Domains verwaltet und Anfragen auf Wunsch je nach Verfügbarkeit oder Standort an unterschiedliche Ziele leitet.

> **Merke**
>
> Nutzen Sie IAM-Rollen statt langlebiger Zugriffsschlüssel. Rollen vergeben Rechte temporär: Eine EC2-Instanz oder eine Lambda-Funktion erhält über eine Rolle automatisch kurzlebige Zugangsdaten. Fest hinterlegte Access Keys in Skripten oder Konfigurationsdateien sind dagegen eine häufige Ursache für Sicherheitsvorfälle.

Konten sauber organisieren

In AWS ist das *Konto* (Account) die wichtigste Trenngrenze. Bewährt hat sich, nicht alles in einem einzigen Konto zu betreiben, sondern getrennte Konten für Produktion, Test und Experimente (Sandbox) anzulegen. Ein Fehler oder ein kompromittierter Zugang in der Testumgebung kann dann nicht die Produktion beeinträchtigen.

AWS Organizations bündelt diese Konten unter einer gemeinsamen Verwaltung und Abrechnung. Mit *Service Control Policies* (SCPs) setzen Sie Leitplanken für ganze Kontogruppen – etwa, dass bestimmte Regionen nicht genutzt oder die Protokollierung nicht abgeschaltet werden darf. Diese Regeln gelten selbst für Administratoren der einzelnen Konten. Das gesamte Fundament aus Kontenstruktur, Identitäten, Netzwerk und Leitplanken bezeichnet man als **Landing Zone**; sie sollte stehen, bevor die ersten produktiven Anwendungen einziehen.

Kosten, Audit und Einstieg

Für die Kostenkontrolle stehen der **Cost Explorer** zur Auswertung und **AWS Budgets** mit Alarmen zur Verfügung. Ein durchgängiges *Tagging* sorgt dafür, dass sich Kosten Projekten und Teams zuordnen lassen. Für Nachvollziehbarkeit sorgt **CloudTrail**: Der Dienst protokolliert alle API-Aufrufe, also jede Aktion in der Konsole, per Kommandozeile oder per Skript. Ergänzend erkennt **GuardDuty** verdächtige Aktivitäten, etwa ungewöhnliche Anmeldungen oder Verbindungen zu bekannten Schadservern.

Für den Einstieg empfiehlt es sich, diese Grundlagen von Beginn an mitzudenken, statt sie später nachzurüsten: CloudTrail in allen Regionen aktivieren, Budgets mit Alarmen einrichten, das Root-Konto mit MFA absichern und danach nur noch für Ausnahmefälle verwenden. Diese wenigen Schritte kosten kaum Zeit und verhindern viele der typischen Probleme neuer AWS-Umgebungen.

Aus der Praxis

Als Leitlinie für gute Architekturen dient das *Well-Architected Framework* von AWS. Es beschreibt bewährte Vorgehensweisen zu Sicherheit, Zuverlässigkeit, Leistung, Kosten und Betrieb. Wer sich beruflich mit AWS beschäftigt, findet außerdem einen klaren Zertifizierungspfad – vom Einstiegsniveau „Cloud Practitioner“ bis zu den „Associate“-Zertifizierungen und darüber hinaus.

Das Wichtigste in Kürze

- EC2 und VPC bilden das Fundament, verteilt über Regionen und Availability Zones.
- S3 speichert Objekte, RDS stellt verwaltete Datenbanken bereit.
- IAM, Lambda, CloudWatch und Route 53 gehören zum Grundwissen.

- Getrennte Konten, AWS Organizations und SCPs schaffen eine sichere Landing Zone.
- Rechte mit Rollen und Least Privilege, Kosten mit Budgets und Tags diszipliniert managen.

Google Cloud & Workspace

Googles Cloud-Angebote im Überblick

Nach diesem Kapitel können Sie …

- Google Workspace und Google Cloud (GCP) klar voneinander unterscheiden,
- Workspace und Microsoft 365 sachlich vergleichen,
- die Kerndienste und die Ressourcen-Hierarchie von GCP beschreiben,
- einschätzen, in welchen Szenarien Google seine besonderen Stärken ausspielt.

Zwei Welten unter einem Namen

Wenn von „Google in der Cloud“ die Rede ist, sind meist zwei sehr unterschiedliche Angebote gemeint. **Google Workspace** ist eine Produktivitäts-Suite für Büroarbeit: E-Mail mit Gmail, Textverarbeitung und Tabellen mit Docs und Sheets, Dateiablage mit Drive und Videokonferenzen mit Meet. Es handelt sich um ein SaaS-Angebot und ist damit das direkte Gegenstück zu Microsoft 365.

Google Cloud, oft auch **GCP** (Google Cloud Platform) genannt, ist dagegen eine Infrastruktur- und Plattform-Cloud. Hier mieten Unternehmen virtuelle Maschinen, Speicher, Datenbanken und Analysewerkzeuge – vergleichbar mit Microsoft Azure oder AWS. Beide Angebote können unabhängig voneinander genutzt werden: Ein Unternehmen kann Workspace für die Büroarbeit einsetzen und seine Server bei einem anderen Anbieter betreiben, oder umgekehrt.

Workspace und Microsoft 365 im Vergleich

Funktional decken Workspace und Microsoft 365 weitgehend dasselbe ab: E-Mail und Kalender, Dateiablage, gemeinsames Arbeiten an Dokumenten und Videokonferenzen. Die Unterschiede liegen eher in der Philosophie. Workspace ist konsequent für den Browser entwickelt und besonders stark beim gleichzeitigen Bearbeiten von Dokumenten in Echtzeit. Microsoft 365 hat seine Wurzeln in den Desktop-Anwendungen von Office und punktet häufig dort, wo komplexe Excel-Dateien, umfangreiche Word-Dokumente oder eine enge Anbindung an Windows-Umgebungen und Active Directory gefragt sind.

Die Wahl ist deshalb oft weniger eine technische als eine strategische Entscheidung: Welches Ökosystem nutzt das Unternehmen bereits, welche Fachanwendungen hängen daran, und womit sind Mitarbeitende und Geschäftspartner vertraut? Ein späterer Wechsel zwischen beiden Welten ist möglich, aber echter Aufwand – Postfächer, Dateien, Freigaben, Kalender und Gewohnheiten müssen umziehen.

Aufgabe	Google Workspace	Microsoft 365
E-Mail und Kalender	Gmail, Kalender	Exchange Online, Outlook
Dateien	Drive, geteilte Ablagen	OneDrive, SharePoint
Dokumente	Docs, Sheets, Slides	Word, Excel, PowerPoint
Kommunikation	Meet, Chat	Teams

Die Kerndienste von GCP

Die Grundbausteine von Google Cloud ähneln denen der anderen großen Anbieter. Die **Compute Engine** stellt virtuelle Maschinen bereit. **Cloud Storage** ist der Objektspeicher für Dateien, Backups und große Datenmengen, ebenfalls mit Speicherklassen für unterschiedlich häufig genutzte Daten. **VPC-Netzwerke** bilden das private Netz, in dem die Ressourcen laufen; eine Besonderheit bei Google ist, dass ein VPC-Netz global angelegt wird und Subnetze in verschiedenen Regionen enthalten kann. Über **IAM** werden schließlich Berechtigungen vergeben.

Wer die Konzepte aus Azure oder AWS kennt, findet sich in GCP daher schnell zurecht. Die Namen sind andere, die Prinzipien – virtuelle Maschinen, Objektspeicher, private Netze, rollenbasierte Rechte – bleiben gleich.

Ressourcen-Hierarchie und Betrieb

Google Cloud ordnet alle Ressourcen in einer klaren Hierarchie: Ganz oben steht die **Organization**, die in der Regel der Firmendomain entspricht. Darunter lassen sich **Folders** (Ordner) anlegen, etwa für Abteilungen oder Umgebungen, und darin liegen die **Projects**. Projekte sind die zentrale Einheit: Jede Ressource gehört zu genau einem Projekt, und Projekte trennen Umgebungen wie Produktion und Test sowie deren Kosten voneinander.

IAM-Rechte vererben sich in dieser Hierarchie von oben nach unten. Wer auf Ebene eines Ordners eine Rolle erhält, hat diese Rechte in allen Projekten darunter. Zur weiteren Zuordnung, etwa nach Kostenstelle oder Team, dienen *Labels*. Für den Betrieb gelten dieselben Grundregeln wie bei den anderen Anbietern: Budgets und Alarme setzen, Rechte nach dem Prinzip der minimalen Rechte vergeben und bei der Wahl der Region die Anforderungen an Compliance und Datenschutz beachten.

Sicherheit in Google Cloud

Ein besonderes Augenmerk verdienen **Service Accounts**. Das sind technische Konten, mit denen Anwendungen und Dienste auf andere Ressourcen zugreifen. Sie sind sehr mächtig, denn sie arbeiten automatisiert und ohne menschliche Kontrolle bei jedem Zu-

griff. Setzen Sie Service Accounts deshalb sparsam ein und geben Sie ihnen nur die Rechte, die sie für ihre konkrete Aufgabe benötigen.

> **Merke**
>
> Ein Service Account mit zu weitreichenden Rechten ist ein offenes Tor: Wer seine Zugangsdaten erbeutet, kann alles tun, was das Konto darf. Minimale Rechte sind hier besonders wichtig.

Für besonders schützenswerte Daten bietet Google weitere Werkzeuge. *VPC Service Controls* ziehen einen Sicherheitsperimeter um ausgewählte Dienste und verhindern, dass Daten aus diesem Bereich in fremde Projekte abfließen – selbst wenn gültige Zugangsdaten verwendet werden. Mit *kundenverwalteten Schlüsseln* behalten Sie die Kontrolle über die Verschlüsselung sensibler Daten selbst, statt sie vollständig dem Anbieter zu überlassen.

Wann Google die richtige Wahl ist

Seine größten Stärken spielt Google bei **Daten, Analyse und künstlicher Intelligenz** aus. **BigQuery** ist ein vollständig verwaltetes Data Warehouse, mit dem sich auch sehr große Datenmengen per SQL in kurzer Zeit auswerten lassen, ohne dass eine eigene Datenbankinfrastruktur betrieben werden muss. Mit **Vertex AI** stellt Google eine Plattform für maschinelles Lernen und KI-Anwendungen bereit, und viele fertige KI-Dienste lassen sich direkt nutzen. Google hat außerdem Kubernetes ursprünglich entwickelt; mit **GKE** (Google Kubernetes Engine) bietet es entsprechend ein ausgereiftes verwaltetes Kubernetes an.

Für hybride und Multi-Cloud-Szenarien bietet Google zudem Werkzeuge, mit denen sich Container-Umgebungen in mehreren Clouds und im eigenen Rechenzentrum einheitlich verwalten lassen. Und wer bereits Workspace nutzt, profitiert von einer nahtlosen Zusammenarbeit zwischen beiden Welten.

> **Aus der Praxis**
>
> Die Entscheidung für einen Cloud-Anbieter hat langfristige Folgen. Bewerten Sie nicht nur einzelne Dienste, sondern das gesamte Ökosystem: vorhandenes Know-how im Team, bestehende Lizenzen, Integration in die Büro-Software und die Verfügbarkeit von Dienstleistern in Ihrer Region.

Das Wichtigste in Kürze

- Workspace ist die Produktivitäts-Suite, GCP die Infrastruktur- und Plattform-Cloud.

- Workspace ist die browserorientierte Alternative zu Microsoft 365, die Wahl ist oft strategisch.
- GCP gliedert Ressourcen in Organization, Folders und Projects, Rechte vererben sich nach unten.
- Service Accounts sparsam und mit minimalen Rechten einsetzen.
- Google glänzt bei Daten und KI, etwa mit BigQuery, Vertex AI und GKE.

Cloud-Migration

Von On-Premises in die Cloud

Nach diesem Kapitel können Sie …

- Gründe für und gegen eine Cloud-Migration abwägen,
- die gängigen Migrationsstrategien („6 R“) unterscheiden und pro Anwendung auswählen,
- eine Migration in Phasen planen und dabei eine Landing Zone voranstellen,
- typische Fehler vermeiden und die Zeit nach der Migration gestalten.

Warum Unternehmen migrieren

Unter einer **Cloud-Migration** versteht man den Umzug von Anwendungen, Daten und Infrastruktur aus dem eigenen Rechenzentrum – *On-Premises* – in die Cloud. Die Beweggründe sind vielfältig: Ressourcen lassen sich flexibel skalieren, neue Systeme stehen in Minuten bereit, und das Unternehmen muss weniger eigene Hardware anschaffen, betreiben und alle paar Jahre erneuern. Finanziell verschiebt sich der Schwerpunkt von großen Einmalinvestitionen hin zu laufenden Betriebskosten.

Ebenso wichtig ist aber die Einsicht, dass nicht alles in die Cloud gehört. Anwendungen mit gleichmäßig hoher Dauerlast, Spezialhardware, sehr geringen Latenzanforderungen oder strengen rechtlichen Auflagen können lokal sinnvoller aufgehoben sein. Und ohne sorgfältige Planung wird eine Migration schnell teurer als erwartet. Eine gute Migration beginnt deshalb nicht mit Technik, sondern mit der Frage, welches Ziel erreicht werden soll.

Die Migrationsstrategien

Für jede Anwendung gibt es mehrere Möglichkeiten, sie in die Cloud zu bringen – oder eben nicht. Diese Strategien sind als die **„6 R“** bekannt; manche Anbieter zählen inzwischen ein siebtes hinzu. Die beiden wichtigsten Pole bilden Lift & Shift auf der einen und Refactoring auf der anderen Seite.

Beim **Rehost**, auch *Lift & Shift* genannt, wird eine Anwendung eins zu eins verschoben: Aus dem lokalen virtuellen Server wird eine virtuelle Maschine in der Cloud. Das geht schnell und erfordert kaum Änderungen, nutzt aber die Vorteile der Cloud kaum aus. Beim **Replatform** wird die Anwendung leicht angepasst, etwa indem die selbst betriebene Datenbank durch einen verwalteten Datenbankdienst ersetzt wird. Beim **Refactor** wird die Anwendung grundlegend umgebaut oder neu entwickelt, um cloud-native Dienste voll auszuschöpfen – mit dem größten Nutzen, aber auch dem größten Aufwand.

Daneben gibt es Optionen, die oft übersehen werden: **Repurchase** ersetzt eine Anwendung durch ein fertiges SaaS-Produkt, **Retire** schaltet nicht mehr benötigte Systeme ab, und **Retain** belässt eine Anwendung bewusst lokal. Gerade die Bestandsaufnahme bringt häufig Systeme zum Vorschein, die niemand mehr braucht.

Strategie	Vorgehen	Aufwand / Nutzen
Rehost (Lift & Shift)	1:1 verschieben	Gering / gering
Replatform	Leicht anpassen, z. B. verwaltete Datenbank	Mittel / mittel
Refactor	Neu bauen, cloud-nativ	Hoch / hoch
Repurchase	Durch SaaS ersetzen	Abhängig vom Produkt
Retire / Retain	Abschalten bzw. bewusst behalten	Spart Migrationsaufwand

> **Merke**
>
> Die Migrationsstrategie wird pro Anwendung entschieden, nicht pauschal für das ganze Unternehmen. Eine Mischung aus Rehost, Replatform, Retire und Retain ist der Normalfall.

Die Phasen einer Migration

Erfolgreiche Migrationen folgen einem klaren Ablauf in vier Phasen. Im **Assessment** wird der Bestand erfasst: Welche Server und Anwendungen gibt es, wie stark sind sie ausgelastet, und – besonders wichtig – welche Abhängigkeiten bestehen zwischen ihnen? Eine Anwendung, die täglich Daten mit einem lokalen Buchhaltungssystem austauscht, kann nicht isoliert umziehen.

In der **Planung** werden die Zielarchitektur und die Reihenfolge festgelegt. Bewährt hat sich, mit weniger kritischen Systemen zu beginnen, um Erfahrung zu sammeln. Die eigentliche **Migration** erfolgt dann schrittweise, Anwendung für Anwendung oder in zusammengehörigen Gruppen. In der abschließenden **Optimierung** werden Kosten und Betrieb feinjustiert – ein Schritt, der gern vergessen wird, aber über den wirtschaftlichen Erfolg entscheidet.

Die großen Anbieter stellen dafür eigene Werkzeuge bereit, etwa *Azure Migrate* oder die Migrationsdienste von AWS. Sie erfassen den Bestand, erkennen Abhängigkeiten automatisiert anhand der Netzwerkverbindungen und unterstützen beim eigentlichen Umzug. Führen Sie vor jedem produktiven Umzug eine Testmigration durch und halten Sie einen Rückfallplan bereit, mit dem Sie im Problemfall zum alten System zurückkehren können.

Landing Zone zuerst

Bevor die erste Anwendung umzieht, muss das Fundament stehen. Diese Grundstruktur heißt **Landing Zone** und umfasst Konten bzw. Subscriptions, die Netzwerk-Grundstruktur, die Anbindung an das lokale Netz, die Identitäts- und Zugriffsverwaltung mit Single Sign-on sowie Leitplanken für Sicherheit, Governance und Kostenkontrolle. Wird dieses Fundament übersprungen, entstehen schnell unkontrollierte Umgebungen, die später mühsam aufgeräumt werden müssen.

Daten sicher übertragen

Eine eigene Herausforderung ist die **Datenmigration**. Kleinere und mittlere Datenmengen werden online über die Internetleitung oder eine dedizierte Verbindung übertragen. Bei sehr großen Datenbeständen kann das Wochen dauern; dann bieten die Anbieter physische Transfer-Geräte an, die im eigenen Rechenzentrum befüllt und per Kurier zurückgeschickt werden. Planen Sie Bandbreite und Zeitfenster realistisch ein.

Um die Ausfallzeit gering zu halten, werden die Daten meist zunächst im laufenden Betrieb synchronisiert. Zum Umschaltzeitpunkt werden dann nur noch die letzten Änderungen übertragen, und die Anwendung ist nur kurz nicht verfügbar. Nach der Übertragung sollte die Integrität der Daten geprüft werden, etwa über Prüfsummen oder Stichproben.

Aus der Praxis

Die häufigsten Fehler bei Migrationen sind unterschätzte Kosten, übersehene Abhängigkeiten, zu spät bedachte Sicherheits- und Compliance-Anforderungen und ein fehlender Rückweg. Besonders reines Lift & Shift ohne anschließende Optimierung führt oft zu höheren Kosten als im eigenen Rechenzentrum, weil Server in der Cloud häufig überdimensioniert weiterlaufen.

Nach der Migration

Mit dem Umzug ist die Arbeit nicht beendet. Im **Rightsizing** werden überdimensionierte Ressourcen auf die tatsächlich benötigte Größe verkleinert – lokal wurden Server oft großzügig ausgelegt, in der Cloud kostet jede ungenutzte CPU Geld. Monitoring und Alarme müssen für die neue Umgebung eingerichtet werden. Die Kosten werden fortlaufend beobachtet und optimiert, ein Ansatz, der unter dem Begriff *FinOps* bekannt ist. Und schließlich sollte die alte lokale Umgebung geordnet abgebaut werden, inklusive sicherer Datenlöschung und Kündigung nicht mehr benötigter Verträge.

Das Wichtigste in Kürze

- Migrieren bringt Flexibilität, aber nicht alles gehört in die Cloud.
- Die Strategie („6 R“) wird pro Anwendung gewählt, von Rehost bis Retire.
- Ablauf: Assessment, Planung, Migration, Optimierung – mit einer Landing Zone als Fundament.
- Testmigration und Rückfallplan sind Pflicht, Kosten von Anfang an mitdenken.
- Nach dem Umzug: Rightsizing, Monitoring, FinOps und geordneter Rückbau.

Infrastructure as Code

Infrastruktur automatisiert und reproduzierbar

Nach diesem Kapitel können Sie …

- erklären, was Infrastructure as Code ist und welche Vorteile es bringt,
- deklarative und imperative Ansätze unterscheiden,
- Terraform, Bicep, CloudFormation und Ansible einordnen,
- die Rolle von State, Modulen, GitOps und Pipelines beschreiben,
- gute Praxis wie Drift-Vermeidung und sicheren Umgang mit Secrets anwenden.

Infrastruktur als Code statt Klickarbeit

Wer eine Cloud-Umgebung über die Weboberfläche aufbaut, klickt sich durch zahlreiche Masken: Netzwerk anlegen, Subnetze definieren, Firewall-Regeln setzen, virtuelle Maschinen erstellen. Das funktioniert für einzelne Versuche, wird aber schnell unübersichtlich. Niemand weiß nach einigen Monaten mehr genau, welche Einstellung warum gesetzt wurde, und eine zweite, identische Umgebung für Tests lässt sich kaum zuverlässig nachbauen.

Infrastructure as Code (IaC) löst dieses Problem, indem die gesamte Infrastruktur in Textdateien beschrieben wird. Ein Werkzeug liest diese Dateien und erzeugt daraus die tatsächlichen Ressourcen. Der Code wird so zur einzigen verbindlichen Quelle der Wahrheit: Er ist *reproduzierbar*, weil sich dieselbe Umgebung beliebig oft identisch aufbauen lässt, *versionierbar*, weil jede Änderung in einem Versionsverwaltungssystem wie Git nachvollziehbar gespeichert wird, und *überprüfbar*, weil Kolleginnen und Kollegen Änderungen vor der Umsetzung begutachten können. Ganz nebenbei sinkt die Zahl manueller Fehler deutlich.

Deklarativ oder imperativ

Grundsätzlich gibt es zwei Arten, Infrastruktur in Code zu beschreiben. Beim **imperativen** Ansatz schreiben Sie die einzelnen Schritte auf: „Erstelle ein Netzwerk, dann ein Subnetz, dann eine VM.“ So arbeiten klassische Skripte. Das Problem: Läuft das Skript ein zweites Mal, versucht es alles erneut anzulegen – Sie müssen selbst prüfen, was bereits existiert.

Beim **deklarativen** Ansatz beschreiben Sie nur den gewünschten *Zielzustand*: „Es soll ein Netzwerk mit zwei Subnetzen und drei VMs geben.“ Das Werkzeug vergleicht diesen Sollzustand mit dem Istzustand und führt nur die Änderungen aus, die nötig sind. Fehlt

eine VM, wird sie angelegt; ist alles vorhanden, passiert nichts. Bei IaC ist der deklarative Ansatz der Regelfall.

> **Merke**
>
> Deklarativ heißt: Sie beschreiben, *was* am Ende da sein soll, und das Werkzeug ermittelt selbst, *wie* es dorthin kommt. Imperativ heißt: Sie beschreiben jeden einzelnen Schritt.

Die wichtigsten Werkzeuge

Terraform ist das bekannteste cloud-übergreifende Werkzeug. Über sogenannte *Provider* unterstützt es Azure, AWS, Google Cloud und viele weitere Plattformen mit einer einheitlichen Sprache. Daneben haben die großen Anbieter eigene, native Werkzeuge: **Bicep** und die darunterliegenden ARM-Vorlagen für Azure sowie **CloudFormation** für AWS. Native Werkzeuge unterstützen neue Dienste des jeweiligen Anbieters oft besonders schnell, binden aber an diesen einen Anbieter.

Ein kurzes Beispiel zeigt, wie deklarativer Terraform-Code aussieht. Er beschreibt eine Ressourcengruppe in Azure, ohne einen einzigen Arbeitsschritt zu nennen:

```
resource "azurerm_resource_group" "web" {
  name     = "rg-web-prod"
  location = "westeurope"
  tags = {
    kostenstelle = "IT-100"
  }
}
```

Ansible und das Konfigurationsmanagement

Ansible liegt etwas quer zu den bisher genannten Werkzeugen. Sein Schwerpunkt ist das *Konfigurationsmanagement*: Es richtet Betriebssysteme und Software auf vorhandenen Servern ein, installiert Pakete, verteilt Konfigurationsdateien und startet Dienste. Ansible arbeitet *agentenlos* – auf den Zielsystemen muss keine Zusatzsoftware installiert sein; die Verbindung erfolgt über SSH bei Linux oder WinRM bei Windows. Die Aufgaben werden in *Playbooks* im leicht lesbaren YAML-Format beschrieben.

In der Praxis ergänzen sich die Werkzeuge häufig: Terraform legt Netzwerk und virtuelle Maschinen an, Ansible konfiguriert anschließend die Server und die darauf laufende Software.

State und Module

Damit ein deklaratives Werkzeug wie Terraform weiß, welche Ressourcen es bereits verwaltet, führt es einen **State** – eine Zustandsdatei, die abbildet, was tatsächlich existiert. Der State ist kritisch: Geht er verloren oder wird er beschädigt, verliert das Werkzeug den Überblick über die Infrastruktur. Außerdem kann er sensible Informationen enthalten. Speichern Sie den State deshalb nicht lokal auf einem Laptop, sondern *remote* in einem geschützten Speicher, und aktivieren Sie *Locking*, damit nicht zwei Personen gleichzeitig Änderungen ausführen.

Für wiederkehrende Bausteine – etwa ein standardisiertes Netzwerk oder eine Webanwendung mit Datenbank – nutzen Sie **Module**. Sie werden einmal geschrieben und mehrfach mit unterschiedlichen Parametern verwendet, statt Code per Copy-and-paste zu vervielfältigen. Variablen und Umgebungen wie Test und Produktion sollten sauber voneinander getrennt sein.

GitOps und Pipelines

Ein modernes Betriebsmodell für IaC ist **GitOps**: Der gewünschte Zustand der Infrastruktur liegt in einem Git-Repository, und Änderungen erfolgen ausschließlich über Commits. Eine automatisierte Pipeline übernimmt die Umsetzung. So ist jede Änderung nachvollziehbar, geprüft und bei Bedarf wieder rückgängig zu machen.

Dabei gilt ein klarer Ablauf: Erst *planen*, dann *anwenden*. Der Befehl `terraform plan` zeigt in einer Vorschau, welche Ressourcen angelegt, geändert oder gelöscht würden. Diese Vorschau wird im Pull-Request angezeigt und begutachtet. Erst nach der Freigabe führt die Pipeline `terraform apply` aus und verändert die echte Infrastruktur. Ergänzend kann *Policy-as-Code* automatisch prüfen, ob Regeln eingehalten werden – etwa, dass keine Speicher öffentlich erreichbar sind. So verändert nur geprüfter Code die Infrastruktur.

```
terraform init    # Provider und Remote-State einrichten
terraform plan    # Änderungen als Vorschau anzeigen
terraform apply   # Änderungen nach Freigabe umsetzen
```

Aus der Praxis

Klicken Sie in einer per IaC verwalteten Umgebung nichts von Hand nach. Jede manuelle Änderung erzeugt *Drift* – eine Abweichung zwischen Code und Wirklichkeit, die beim nächsten Durchlauf überschrieben wird oder zu unerwarteten Fehlern führt. Halten Sie Änderungen klein und nachvollziehbar, beziehen Sie Secrets aus einem Tresor und legen Sie State-Dateien und Passwörter niemals in ein Repository.

Das Wichtigste in Kürze

- IaC beschreibt Infrastruktur als versionierten, reproduzierbaren Code.
- Deklarative Werkzeuge wie Terraform, Bicep und CloudFormation sind der Regelfall.
- Ansible konfiguriert Server agentenlos über SSH oder WinRM mit YAML-Playbooks.
- Der State ist kritisch: remote speichern, sperren und nie ins Repository legen.
- GitOps: Git als Quelle der Wahrheit, erst plan, dann apply nach Freigabe.

Cloud-Security & FinOps

Sicher und kosteneffizient in der Cloud

Nach diesem Kapitel können Sie …

- die geteilte Verantwortung als Grundlage der Cloud-Sicherheit anwenden,
- bewährte Regeln für Identitäten und Berechtigungen (IAM) umsetzen,
- Werkzeuge wie CSPM und CASB sowie zentrale Protokollierung einordnen,
- erklären, was FinOps bedeutet und wie sich Cloud-Kosten gezielt senken lassen.

Geteilte Verantwortung als Ausgangspunkt

Jede Betrachtung von Cloud-Sicherheit beginnt mit dem **Modell der geteilten Verantwortung**. Der Anbieter sichert die Cloud selbst: Rechenzentren, Hardware, Netz und die Virtualisierungsschicht. Sie sichern, was Sie in der Cloud betreiben: Konfigurationen, Zugänge, Daten und – je nach Service-Modell – auch Betriebssysteme und Anwendungen. Wo genau die Grenze verläuft, hängt davon ab, ob Sie IaaS, PaaS oder SaaS nutzen, und sollte für jeden Dienst klar abgegrenzt und dokumentiert sein.

Genau an dieser Grenze liegt das größte Risiko. Die häufigste Ursache für Sicherheitsvorfälle in der Cloud sind **Fehlkonfigurationen** auf Kundenseite: ein Speicher, der versehentlich öffentlich lesbar ist, eine Datenbank mit offener Internetadresse oder ein Administratorkonto ohne zweiten Faktor. Weil sich Ressourcen in der Cloud mit wenigen Klicks anlegen lassen, entstehen solche Fehler schnell – und werden von automatisierten Suchwerkzeugen der Angreifer ebenso schnell gefunden.

Identitäten und Berechtigungen im Griff

In der Cloud gibt es keinen Netzwerkrand mehr, hinter dem man sich verstecken kann – die Identität ist der neue Perimeter. Saubere **IAM-Praktiken** sind deshalb die wichtigste Grundlage. Vergeben Sie Rechte konsequent nach dem Prinzip der *minimalen Rechte* (Least Privilege): Jedes Konto erhält nur die Berechtigungen, die es tatsächlich benötigt. Erzwingen Sie *MFA* für alle Konten, insbesondere für Administratoren.

Die besonders mächtigen Hauptkonten – das *Root-Konto* bei AWS oder die Rolle des *Besitzers* (Owner) in Azure – sollten nicht für die tägliche Arbeit genutzt werden. Sichern Sie sie stark ab und verwenden Sie sie nur für die wenigen Aufgaben, die wirklich nur mit ihnen möglich sind. Überprüfen Sie Berechtigungen außerdem regelmäßig: Mit der Zeit sammeln sich Rechte an, die niemand mehr braucht, etwa nach Projektwechseln oder bei ausgeschiedenen Mitarbeitenden.

Werkzeuge für Cloud-Sicherheit

Weil Fehlkonfiguration das Hauptrisiko ist, gibt es Werkzeuge, die gezielt danach suchen. **CSPM** (Cloud Security Posture Management) prüft Ihre Cloud-Umgebung fortlaufend gegen bewährte Sicherheitsregeln und meldet Abweichungen – etwa unverschlüsselte Speicher, offene Ports oder fehlende Protokollierung. Die großen Anbieter haben solche Funktionen in ihre eigenen Security-Center integriert; daneben gibt es herstellerübergreifende Lösungen.

Ein **CASB** (Cloud Access Security Broker) setzt an anderer Stelle an: Er kontrolliert, welche Cloud-Dienste die Mitarbeitenden nutzen, und erkennt sogenannte *Schatten-IT*, also Dienste, die ohne Wissen der IT eingesetzt werden. Zusätzlich kann ein CASB Richtlinien durchsetzen, etwa das Hochladen vertraulicher Dateien in private Cloud-Speicher verhindern.

Unverzichtbar sind außerdem *Verschlüsselung* – sowohl für gespeicherte Daten (in Ruhe) als auch für Daten bei der Übertragung – und eine aktivierte *Protokollierung*. Sammeln Sie Logs zentral über alle Konten und Subscriptions hinweg, nutzen Sie die cloud-nativen Security-Center, priorisieren Sie die Befunde der CSPM-Werkzeuge nach Risiko und leiten Sie Alarme direkt in Ticketsystem oder Team-Chat, damit sie nicht ungelesen liegen bleiben.

> **Merke**
>
> Verschlüsselung ist in der Cloud heute Standard. Entscheidend ist, wer die Schlüssel kontrolliert. Für besonders sensible Daten lohnt sich eine eigene Schlüsselverwaltung über KMS- oder HSM-Dienste, bei der Zugriffe auf die Schlüssel protokolliert und ausgewertet werden.

Datenschutz in der Cloud

Neben der technischen Sicherheit spielt der rechtliche Rahmen eine wichtige Rolle. Achten Sie darauf, in welcher Region und damit in welchem Rechtsraum Ihre Daten gespeichert werden, und ob der Anbieter Zugriffe aus anderen Rechtsräumen ermöglichen könnte. Für personenbezogene Daten gilt die DSGVO; ein Vertrag zur Auftragsverarbeitung ist Pflicht. Ein *Key Management Service* (KMS) oder ein *Hardware-Sicherheitsmodul* (HSM) hilft dabei, die Kontrolle über die Schlüssel zu behalten. Und wie überall gilt: Zugriffe auf sensible Daten sollten protokolliert und regelmäßig ausgewertet werden.

Was FinOps bedeutet

Der zweite große Schwerpunkt dieses Kapitels sind die Kosten. **FinOps** – ein Kunstwort aus „Finance“ und „Operations“ – beschreibt eine Kostenkultur für die Cloud. Es geht nicht um ein einzelnes Werkzeug, sondern um Zusammenarbeit zwischen Technik, Fi-

nanzen und Fachbereichen. Die Grundidee: Wer Ressourcen anlegt, soll auch sehen, was sie kosten, und dafür Verantwortung übernehmen.

Dazu werden die Kosten zunächst *sichtbar* gemacht, etwa über Tags und regelmäßige Berichte. Anschließend wird die *Verantwortung in die Teams* gegeben: Beim *Showback* werden den Teams ihre Kosten nur angezeigt, beim *Chargeback* werden sie ihnen tatsächlich intern verrechnet. Und schließlich wird *kontinuierlich optimiert* – in regelmäßigen Reviews statt einmal im Jahr beim Blick auf die Gesamtrechnung. Anomalie-Alarme melden zusätzlich ungewöhnliche Kostensprünge, etwa wenn über Nacht versehentlich große Instanzen gestartet wurden.

Kosten gezielt senken

Die wirksamsten Hebel sind oft überraschend einfach. Schalten Sie *ungenutzte Ressourcen* konsequent ab: vergessene Testmaschinen, verwaiste Datenträger, alte Snapshots und nicht zugeordnete öffentliche IP-Adressen. Beim **Rightsizing** wählen Sie für jede Ressource die passende Größe – viele Systeme sind überdimensioniert, weil sie aus dem eigenen Rechenzentrum mit großzügigen Reserven übernommen wurden.

Für Systeme mit gleichmäßiger Dauerlast lohnen sich **Reserved Instances** oder **Savings Plans**: Sie verpflichten sich für einen längeren Zeitraum zu einer bestimmten Nutzung und erhalten dafür deutliche Rabatte gegenüber dem regulären Preis. Weitere Einsparungen bringen passende *Storage-Tiers*, die selten genutzte Daten automatisch in günstigere Speicherklassen verschieben, sowie *Auto Scaling*, das Kapazität nur dann bereitstellt, wenn sie gebraucht wird.

Aus der Praxis

Zeitpläne für Test- und Entwicklungsumgebungen sind ein schneller Gewinn: Werden diese Systeme abends und am Wochenende automatisch heruntergefahren, laufen sie nur noch einen Bruchteil der Woche – und verursachen entsprechend geringere Rechenkosten.

Das Wichtigste in Kürze

- Geteilte Verantwortung verstehen: Fehlkonfiguration ist das Hauptrisiko.
- Least Privilege, MFA für alle und keine Alltagsnutzung von Root- oder Owner-Konten.
- CSPM findet Fehlkonfigurationen, CASB kontrolliert die Nutzung von Cloud-Diensten.
- FinOps macht Kosten sichtbar und gibt die Verantwortung in die Teams.
- Abschalten, Rightsizing und Reservierungen sparen am meisten.

LB-09

Monitoring & Management

1. Monitoring-Grundlagen
2. Monitoring-Tools Open Source
3. Monitoring-Tools Commercial
4. RMM-Systeme
5. ITSM & Ticketsysteme
6. IT-Dokumentation

Monitoring-Grundlagen

Systeme im Blick behalten, bevor Nutzer klagen

Nach diesem Kapitel können Sie …

- erklären, warum Monitoring für einen stabilen IT-Betrieb unverzichtbar ist,
- Monitoring und Observability sowie die drei Säulen Metriken, Logs und Traces unterscheiden,
- festlegen, was überwacht werden sollte und wie sinnvolle Schwellwerte aussehen,
- Alarm-Müdigkeit vermeiden und Verfügbarkeit mit SLA, SLO und Fehlerbudget messen.

Warum Monitoring?

Ohne Monitoring erfährt die IT von Problemen meist auf die unangenehmste Weise: durch einen Anruf aus der Fachabteilung, dass „nichts mehr geht“. Dann ist der Schaden bereits eingetreten, und die Fehlersuche beginnt unter Zeitdruck. **Monitoring** – die laufende, automatisierte Überwachung von Systemen – dreht dieses Verhältnis um. Es erkennt Probleme, bevor sie zum Ausfall werden: Eine Festplatte, die sich langsam füllt, ein Dienst, der immer häufiger neu startet, oder ein Zertifikat, das in zwei Wochen abläuft.

Gutes Monitoring erfüllt dabei mehrere Aufgaben. Es sichert *Verfügbarkeit und Leistung*, weil Störungen früh auffallen. Es liefert *Trends* für die Kapazitätsplanung, etwa wann der Speicher erweitert werden muss. Und es hilft, Fehler schneller *einzugrenzen*, weil der Verlauf der Messwerte zeigt, wann und wo eine Veränderung begonnen hat. Kurz gesagt: Monitoring macht den IT-Betrieb proaktiv statt reaktiv.

Monitoring und Observability

In den letzten Jahren hat sich neben dem Monitoring der Begriff **Observability** etabliert. Beide hängen eng zusammen, setzen aber unterschiedliche Schwerpunkte. Klassisches Monitoring überwacht *bekannte* Kennzahlen und beantwortet erwartete Fragen: Ist der Server erreichbar? Liegt die CPU-Last unter 90 Prozent? Läuft der Dienst?

Observability – frei übersetzt „Beobachtbarkeit“ – geht einen Schritt weiter. Ein beobachtbares System liefert so viele und so gut verknüpfte Daten, dass sich daraus auch *unerwartete* Probleme verstehen lassen. Die Frage lautet dann nicht mehr nur „Ist X in Ordnung?“, sondern „Warum verhält sich das System gerade so?“. Gerade in verteilten Anwendungen, in denen eine Anfrage viele Dienste durchläuft, ist das entscheidend.

Die drei Säulen: Metriken, Logs, Traces

Die Daten, auf denen Monitoring und Observability aufbauen, werden häufig in drei Säulen eingeteilt. **Metriken** sind Zahlenwerte über die Zeit, etwa CPU-Auslastung, freier Speicher oder Antwortzeiten. Sie sind kompakt, lassen sich gut speichern und eignen sich ideal für Diagramme und Alarme. **Logs** sind einzelne Ereignisse im Detail, zum Beispiel ein fehlgeschlagener Anmeldeversuch oder eine Fehlermeldung einer Anwendung. Sie liefern den Kontext, warum etwas passiert ist. **Traces** schließlich verfolgen den Weg einer einzelnen Anfrage durch mehrere Systeme – vom Webserver über verschiedene Dienste bis zur Datenbank – und zeigen, an welcher Stelle Zeit verloren geht.

> **Merke**
>
> Metriken zeigen, *dass* etwas nicht stimmt, Logs erklären, *was* passiert ist, und Traces zeigen, *wo* im Zusammenspiel der Systeme das Problem liegt. Erst zusammen ergeben sie das Gesamtbild.

Was überwacht werden sollte

Die Basis bilden die klassischen Ressourcen: *CPU, Arbeitsspeicher, Datenträger und Netzwerk*. Darauf aufbauend werden die *Dienste und Anwendungen* überwacht, die eigentlich für die Benutzer wichtig sind – ein Server kann einwandfrei laufen, während der Datenbankdienst darauf gestoppt ist. Oft vergessen, aber häufige Ursache für Ausfälle sind *Zertifikate und Ablaufdaten*, etwa von Webserver-Zertifikaten, Lizenzen oder Domains.

Ein weiterer wichtiger Baustein ist das *Endnutzer-Erlebnis*. Mit **synthetischem Monitoring** werden typische Benutzeraktionen automatisch und regelmäßig nachgestellt, etwa das Aufrufen einer Webseite oder eine Anmeldung. So fällt auf, wenn eine Anwendung zwar technisch läuft, für Benutzer aber unerträglich langsam ist. Der Grundsatz lautet dabei: das Richtige überwachen, nicht alles.

Schwellwerte, Baselines und Alarm-Müdigkeit

Ein Alarm wird ausgelöst, wenn ein Messwert eine Grenze überschreitet. **Statische Schwellwerte** eignen sich für klare Grenzen, etwa „Datenträger zu mehr als 90 Prozent gefüllt“. Viele Werte schwanken jedoch natürlich: Nachts ist die Last gering, am Monatsende laufen Abrechnungsjobs. **Dynamische Baselines** lernen deshalb das normale Verhalten eines Systems und melden Abweichungen davon. Sie berücksichtigen saisonale Muster wie Tag und Nacht oder Monatswechsel.

Zu eng gesetzte Schwellen erzeugen dagegen Rauschen – und damit eine der größten Gefahren im Monitoring: die **Alarm-Müdigkeit** (Alert Fatigue). Wer täglich Hunderte Meldungen erhält, die meist harmlos sind, übersieht irgendwann den einen wichtigen

Alarm. Zu viele Alarme sind deshalb ebenso schädlich wie zu wenige.

Aus der Praxis

Prüfen Sie bei jedem Alarm: Muss hier jemand handeln? Wenn nicht, gehört die Meldung in einen Bericht, nicht in die Alarmierung. Priorisieren und gruppieren Sie Alarme, definieren Sie klare Eskalationswege und räumen Sie die Alarmregeln regelmäßig auf. Alarmqualität schlägt Alarmmenge.

SLA, SLO und Fehlerbudget

Um Verfügbarkeit messbar zu machen, werden Ziele festgelegt. Ein **SLA** (Service Level Agreement) ist eine verbindliche Zusage gegenüber Kunden oder Fachbereichen, etwa eine Verfügbarkeit von 99,9 Prozent im Monat. Ein **SLO** (Service Level Objective) ist das interne Ziel, das meist etwas strenger angesetzt wird, damit das SLA sicher eingehalten wird. Damit beides überprüfbar ist, muss die Verfügbarkeit sauber gemessen und berichtet werden.

Aus dem SLO ergibt sich das **Fehlerbudget**: die erlaubte Ausfallzeit, bevor Konsequenzen folgen. Bei 99,9 Prozent Verfügbarkeit sind das rund 43 Minuten im Monat. Ist das Budget noch weitgehend vorhanden, können Teams mutiger Änderungen ausrollen; ist es aufgebraucht, haben Stabilität und Ursachenbehebung Vorrang. Das Fehlerbudget steuert so das Tempo von Änderungen.

Dashboards und Berichte

Die gesammelten Daten entfalten ihren Nutzen erst, wenn sie verständlich aufbereitet werden. **Dashboards** sollten rollengerecht gestaltet sein: Das Technikteam braucht detaillierte Messwerte einzelner Systeme, die Geschäftsführung eher eine Übersicht über Verfügbarkeit und Trends. Langfristige Auswertungen liefern die Grundlage für die Kapazitätsplanung. Grundsätzlich gilt: klar statt überladen – die wichtigsten Signale gehören an die prominenteste Stelle.

Das Wichtigste in Kürze

- Monitoring macht den Betrieb proaktiv statt reaktiv.
- Metriken, Logs und Traces ergeben zusammen das Gesamtbild.
- Das Richtige überwachen, nicht alles – inklusive Zertifikaten und Endnutzer-Erlebnis.
- Alarmqualität schlägt Alarmmenge: sinnvolle Schwellwerte, Baselines und Eskalationswege.
- SLA, SLO und Fehlerbudget machen Verfügbarkeit messbar und steuerbar.

Monitoring-Tools Open Source

Bewährte freie Werkzeuge

Nach diesem Kapitel können Sie …

- Vor- und Nachteile von Open-Source-Monitoring abwägen,
- die Arbeitsweise von Prometheus und Grafana einschließlich Pull-Modell und Alertmanager erklären,
- klassische Allrounder wie Zabbix, Checkmk und Icinga einordnen,
- Werkzeuge für die zentrale Logsammlung benennen und eine passende Auswahl treffen.

Warum Open Source?

Im Bereich Monitoring gibt es eine Reihe ausgereifter Open-Source-Werkzeuge, die in Unternehmen jeder Größe eingesetzt werden. Ihre Vorteile liegen auf der Hand: Es fallen *keine Lizenzgebühren* an, die Werkzeuge sind sehr *flexibel* und lassen sich an fast jede Umgebung anpassen, und hinter den bekannten Projekten stehen *große Communities*, die Erweiterungen, Vorlagen und Hilfe bei Problemen bereitstellen. Für viele Projekte gibt es zusätzlich kommerziellen Support oder Enterprise-Versionen.

Der Preis für diese Freiheit ist der **Betriebsaufwand**. Installation, Updates, Skalierung, Datensicherung und Hochverfügbarkeit liegen in Ihrer eigenen Verantwortung. Dieser Aufwand sollte von Anfang an realistisch eingeplant werden – ein Monitoring-System, das selbst nicht überwacht und gepflegt wird, fällt im ungünstigsten Moment aus.

Prometheus und Grafana

Im Umfeld moderner, cloud-nativer Anwendungen hat sich das Duo **Prometheus** und **Grafana** als Standard etabliert. Die Aufgaben sind klar verteilt: Prometheus sammelt und speichert Metriken, Grafana visualisiert sie in übersichtlichen Dashboards. Grafana kann dabei nicht nur Prometheus, sondern auch zahlreiche andere Datenquellen anzeigen und wird deshalb oft als zentrale Oberfläche für mehrere Systeme genutzt.

Prometheus arbeitet nach dem **Pull-Modell**. Die zu überwachenden Systeme stellen ihre Messwerte über sogenannte *Exporter* auf einer Webadresse bereit, und Prometheus holt sie in festen Abständen ab. Für Linux-Server gibt es etwa den Node Exporter, für Datenbanken, Webserver und viele andere Dienste eigene Exporter. Für kurzlebige Aufgaben wie nächtliche Batch-Jobs, die zum Abholzeitpunkt vielleicht schon beendet sind, steht das *Pushgateway* bereit, an das solche Jobs ihre Werte aktiv senden.

Gespeichert werden die Daten als *Zeitreihen*. Jede Metrik trägt dabei **Labels** – Schlüssel-Wert-Paare wie Server, Umgebung oder Dienst –, über die sich die Daten flexibel filtern und zusammenfassen lassen. Ein kurzer Auszug zeigt, wie ein Exporter seine Werte bereitstellt:

```
node_filesystem_avail_bytes{mountpoint="/",instance="web01"} 1.2e+10
node_load1{instance="web01"} 0.42
```

Alerting mit dem Alertmanager

Prometheus prüft regelmäßig festgelegte Alarmregeln und gibt ausgelöste Alarme an den **Alertmanager** weiter. Dieser entscheidet, was mit den Alarmen geschieht. Er *gruppiert* zusammengehörige Meldungen, sodass bei einem Netzwerkausfall nicht fünfzig Einzelalarme eintreffen, sondern eine zusammenfassende Nachricht. Über *Silences* lassen sich Alarme für geplante Wartungsarbeiten gezielt stummschalten. Und über das *Routing* gelangen Alarme an das richtige Ziel – in den Team-Chat, als Ticket ins Ticketsystem oder an die Rufbereitschaft.

> **Merke**
>
> Prometheus sammelt Metriken im Pull-Modell über Exporter, Grafana visualisiert sie, und der Alertmanager gruppiert, unterdrückt und verteilt die Alarme.

Klassische Allrounder

Neben Prometheus gibt es bewährte Allround-Lösungen, die vor allem für klassische Infrastruktur aus Servern, Netzwerkgeräten und Diensten beliebt sind. **Zabbix** und **Checkmk** bieten umfassendes Infrastruktur-Monitoring mit einer integrierten Weboberfläche, automatischer Erkennung von Systemen und vielen fertigen Vorlagen für gängige Hardware und Software. **Nagios** ist der langjährige Klassiker für die Verfügbarkeitsüberwachung; **Icinga** ist aus Nagios hervorgegangen und wurde seitdem eigenständig weiterentwickelt.

Diese Werkzeuge arbeiten wahlweise mit *Agenten*, die auf den überwachten Systemen installiert werden und tiefe Einblicke liefern, oder *agentenlos*, etwa über SNMP, Ping oder Abfragen von Diensten. Für Einsteiger und kleinere Umgebungen sind sie oft schneller produktiv als ein Prometheus-Setup, weil vieles bereits fertig mitgeliefert wird.

Werkzeug	Schwerpunkt	Typischer Einsatz
Prometheus + Grafana	Metriken als Zeitreihen, Dashboards	Cloud-native Umgebungen, Container
Zabbix / Checkmk	Umfassendes Infrastruktur-Monitoring	Server, Netzwerk, klassische IT
Nagios / Icinga	Verfügbarkeits-Checks	Bewährte Dienstüberwachung
ELK-Stack / Graylog	Zentrale Logsammlung und Suche	Fehleranalyse, Sicherheitsauswertung

Logs zentral sammeln

Für die zweite Säule der Observability, die Logs, gibt es eigene Werkzeuge. Weit verbreitet ist der **ELK-Stack** aus *Elasticsearch* (Speicherung und Suche), *Logstash* (Einsammeln und Aufbereiten der Logs) und *Kibana* (Oberfläche für Suche und Visualisierung). Eine häufig als schlanker empfundene Alternative ist **Graylog**. Beide ermöglichen es, die Logs aller Server und Anwendungen an einer Stelle zu durchsuchen, statt sich einzeln auf jedem System anzumelden.

Wichtig ist dabei die Planung von Speicher und *Aufbewahrungsdauer* (Retention): Logs wachsen schnell, und nicht jede Meldung muss über Jahre aufbewahrt werden. Gleichzeitig können gesetzliche oder interne Vorgaben verlangen, bestimmte Protokolle eine Mindestzeit lang aufzubewahren.

Skalierung und Speicher

Mit wachsender Umgebung wächst auch der Speicherbedarf des Monitorings – oft schneller als erwartet. Legen Sie die Aufbewahrung bewusst fest und unterscheiden Sie zwischen detaillierten Kurzzeitdaten und verdichteten Langzeitdaten. Prometheus selbst ist eher für kürzere Zeiträume ausgelegt; für Langzeitspeicherung und sehr große Umgebungen gibt es Erweiterungen wie *Thanos* oder *Mimir*. Hochverfügbarkeit erreichen Sie durch Redundanz, etwa durch zwei parallel laufende Prometheus-Instanzen.

Aus der Praxis

Klären Sie vor der Auswahl, ob Sie vor allem Metriken, Logs oder beides brauchen, bewerten Sie den Betriebsaufwand realistisch und prüfen Sie die Skalierbarkeit. Starten Sie mit einem kleinen Setup für die wichtigsten Systeme und erweitern Sie es schrittweise – statt gleich alles auf einmal überwachen zu wollen.

Das Wichtigste in Kürze

- Open Source ist mächtig und lizenzkostenfrei, der Betrieb liegt aber bei Ihnen.

- Prometheus und Grafana sind der Standard für Metriken im cloud-nativen Umfeld.
- Zabbix, Checkmk und Icinga sind bewährte Allrounder für klassische Infrastruktur.
- ELK-Stack oder Graylog sammeln und durchsuchen Logs zentral.
- Speicher und Aufbewahrung früh planen, klein anfangen und schrittweise ausbauen.

Monitoring-Tools Commercial

Kommerzielle Lösungen mit Support

Nach diesem Kapitel können Sie …

- abwägen, wann sich eine kommerzielle Monitoring-Lösung lohnt,
- bekannte Produkte und die cloud-nativen Dienste der großen Anbieter einordnen,
- erklären, was APM leistet und worin sich agentenbasiertes und agentenloses Monitoring unterscheiden,
- eine Lösung anhand von Kosten, Integrationen und Datenschutz auswählen.

Warum eine kommerzielle Lösung?

Open-Source-Werkzeuge sind leistungsfähig, verlangen aber eigenen Betriebsaufwand und Know-how. Kommerzielle Monitoring-Lösungen setzen genau hier an: Sie bieten *fertige Funktionen*, vorkonfigurierte Vorlagen und eine einheitliche Oberfläche, sodass die Einführung meist deutlich schneller gelingt. Bei SaaS-Lösungen entfällt zudem der Betrieb der Monitoring-Server selbst.

Ein weiterer Vorteil ist der **Herstellersupport**. Bei Problemen gibt es einen Ansprechpartner mit vertraglich zugesicherten Reaktionszeiten (SLAs) – für viele Unternehmen, insbesondere mit kleinen IT-Teams, ein wichtiges Argument. Der Preis dafür sind *laufende Lizenzkosten*, die mit der Größe der überwachten Umgebung wachsen. Kommerziell heißt also: weniger Eigenbetrieb, dafür höhere laufende Kosten.

Bekannte Produkte

PRTG ist eine verbreitete Lösung mit einem einfachen, *sensorbasierten* Ansatz: Jeder überwachte Wert – etwa die CPU-Last eines Servers, der Datendurchsatz eines Switch-Ports oder die Erreichbarkeit einer Webseite – ist ein Sensor. Die Einrichtung ist vergleichsweise einfach, weshalb PRTG vor allem in kleinen und mittleren Unternehmen beliebt ist.

Datadog und **New Relic** sind Cloud-SaaS-Plattformen, die ihre Stärken bei *Observability* und *Application Performance Monitoring* (APM) haben. Sie bündeln Metriken, Logs und Traces in einer Oberfläche und eignen sich besonders für moderne, verteilte Anwendungen. **SolarWinds** bietet eine breite Suite mit Modulen für Netzwerk, Server, Anwendungen und Datenbanken.

Die Preismodelle unterscheiden sich stark: Abgerechnet wird je nach Produkt nach Anzahl der Sensoren, der überwachten Hosts oder nach dem verarbeiteten Datenvolumen. Ein genauer Vergleich anhand der eigenen Umgebung ist deshalb unerlässlich.

Cloud-native Werkzeuge

Wer Ressourcen in der Cloud betreibt, findet bei den großen Anbietern eigene Monitoring-Dienste. **Azure Monitor** überwacht Azure-Ressourcen, **Amazon CloudWatch** die Dienste in AWS. Ihr großer Vorteil ist die tiefe Integration in die jeweilige Cloud: Viele Metriken stehen ohne zusätzliche Einrichtung sofort zur Verfügung, und Alarme können direkt Aktionen wie das Hinzufügen weiterer Instanzen auslösen.

Auch hier sollten Sie die Kosten im Blick behalten. Grundlegende Metriken sind oft enthalten, zusätzliche Metriken, detaillierte Logs und lange Aufbewahrungszeiten werden aber nach Volumen abgerechnet. In Multi-Cloud- oder Hybrid-Umgebungen stoßen die nativen Werkzeuge außerdem an Grenzen, weil sie vor allem die eigene Plattform gut abdecken.

APM und Observability

APM (Application Performance Monitoring) beantwortet eine Frage, die klassisches Infrastruktur-Monitoring nicht beantworten kann: Wo genau innerhalb einer Anwendung geht Zeit verloren? Ein Server kann normal ausgelastet sein, während Benutzer trotzdem lange auf eine Antwort warten – etwa weil eine einzelne Datenbankabfrage langsam ist oder ein externer Dienst verzögert antwortet.

APM-Werkzeuge verfolgen dazu einzelne Anfragen mit *Traces* über verteilte Systeme hinweg, messen die Dauer jedes Schritts und zeigen Fehler mit dem dazugehörigen Kontext an. Für komplexe Anwendungen mit vielen Diensten ist das oft der schnellste Weg zur eigentlichen Ursache eines Performance-Problems.

> **Merke**
>
> Infrastruktur-Monitoring zeigt, *ob* ein Server läuft. APM zeigt, *wo* in einer Anwendung Zeit verloren geht – nicht nur, ob sie erreichbar ist.

Agent oder agentenlos

Die meisten Lösungen unterstützen zwei Arten der Datenerfassung. Ein **Agent** ist eine kleine Software, die auf dem überwachten System installiert wird. Er liefert besonders tiefe Einblicke: laufende Prozesse, Logdateien, Anwendungsmetriken und Details, die von außen nicht sichtbar sind. Dafür müssen Agenten ausgerollt, aktualisiert und gepflegt werden.

Agentenloses Monitoring fragt Systeme von außen über Standardschnittstellen ab, etwa über SNMP bei Netzwerkgeräten, WMI bei Windows oder die APIs von Cloud-Diensten und Anwendungen. Das ermöglicht eine breite Abdeckung ohne Installationsaufwand, liefert aber weniger Tiefe. In der Praxis wird meist eine Mischung aus beidem eingesetzt: Agenten auf wichtigen Servern, agentenlose Abfragen für Netzwerkgeräte, Drucker und Appliances.

Integrationen, Auswahl und Kosten

Ein Monitoring-System entfaltet seinen vollen Nutzen erst im Zusammenspiel mit anderen Werkzeugen. Wichtig sind die Anbindung an *Ticketsysteme* (ITSM), damit aus Alarmen automatisch Tickets entstehen, und *ChatOps*, bei dem Alarme direkt im Team-Chat erscheinen. Für Rufbereitschaft und Eskalation gibt es spezialisierte On-Call-Dienste, die dafür sorgen, dass nachts die richtige Person geweckt wird. Fortgeschrittene Lösungen können zudem *automatische Reaktionen* auslösen, etwa einen Dienst nach einem festgelegten Runbook neu starten.

Für die Auswahl sollten Sie mehrere Kriterien prüfen: Soll die Lösung im eigenen Rechenzentrum (On-Premises) oder als SaaS betrieben werden? Wie ist das Preismodell aufgebaut, und wie entwickeln sich die Kosten, wenn die Umgebung wächst? Welche Integrationen in Cloud, Ticketsystem und Kommunikationswerkzeuge gibt es? Und wo werden die Daten gespeichert – gerade bei SaaS-Lösungen spielen Datenstandort und Datenschutz eine wichtige Rolle, denn Logs können personenbezogene Daten enthalten.

Aus der Praxis

Die Aufbewahrungsdauer der Daten beeinflusst die Kosten oft stärker als erwartet. Legen Sie bewusst fest, wie lange detaillierte Logs und Metriken benötigt werden. Bedenken Sie außerdem den Vendor-Lock-in: Dashboards, Alarmregeln und Agenten lassen sich bei einem späteren Wechsel nicht einfach mitnehmen.

Das Wichtigste in Kürze

- Kommerziell bedeutet weniger Eigenbetrieb und Herstellersupport, aber laufende Kosten.
- PRTG ist im KMU-Umfeld verbreitet, Datadog und New Relic sind stark bei Observability.
- Azure Monitor und CloudWatch sind tief in die jeweilige Cloud integriert.
- APM zeigt, wo in einer Anwendung Zeit verloren geht.
- Preismodell, Integrationen, Datenstandort und Aufbewahrung vor der Auswahl prüfen.

RMM-Systeme

Remote Monitoring & Management

Nach diesem Kapitel können Sie …

- erklären, was ein RMM-System ist und welche Kernfunktionen es bietet,
- begründen, wann ein RMM gegenüber WSUS, Gruppenrichtlinien, Intune oder MDM die praktischere Lösung ist,
- den Einsatz bei MSPs und in der internen IT unterscheiden,
- ein RMM sicher betreiben und Patches kontrolliert ausrollen.

Was ein RMM-System ist

RMM steht für *Remote Monitoring and Management*, also Fernüberwachung und Fernverwaltung. Ein RMM-System ermöglicht es, eine große Zahl von Endpunkten – Arbeitsplatzrechner, Notebooks und Server – zentral von einer Konsole aus zu betreuen. Dazu wird auf jedem verwalteten Gerät ein kleiner **Agent** installiert. Er meldet regelmäßig den Zustand des Geräts an die zentrale Plattform und führt Aufgaben aus, die von dort angestoßen werden.

Besonders verbreitet sind RMM-Systeme bei **Managed Service Providern** (MSPs), also IT-Dienstleistern, die die IT vieler Kundenunternehmen betreuen. Für sie ist es schlicht nicht machbar, jeden Rechner einzeln vor Ort zu warten. Aber auch interne IT-Abteilungen setzen RMM zunehmend ein.

Die Kernfunktionen

Ein RMM bündelt mehrere Aufgaben, die sonst auf verschiedene Werkzeuge verteilt wären. Das *Monitoring* überwacht Zustand und Leistung der Geräte: Ist die Festplatte fast voll, läuft der Virenschutz, ist ein wichtiger Dienst ausgefallen? Das *Patch-Management* verteilt Updates zentral auf alle Geräte. Über den *Fernzugriff* können Technikerinnen und Techniker sich bei Problemen direkt auf ein Gerät schalten, oft auch im Hintergrund, ohne die Arbeit des Benutzers zu unterbrechen. Und über *Automatisierungen und Skripte* lassen sich wiederkehrende Aufgaben auf Hunderte Geräte gleichzeitig ausrollen.

Der Kern des Nutzens liegt in der Effizienz: Viele Geräte werden von einer Stelle aus betreut, Probleme fallen auf, bevor sich Benutzer melden, und Routinearbeiten laufen automatisch.

Bekannte Anbieter

Der Markt für RMM-Lösungen ist groß. Zu den modernen, weit verbreiteten Plattformen zählen **NinjaOne** und **Datto RMM**. Weitere gängige Lösungen sind **N-able**, **Atera** und **Syncro**. **ConnectWise** ist im MSP-Umfeld stark vertreten und bietet neben RMM auch weitere Werkzeuge für Dienstleister. **ManageEngine** ist mit Produkten wie Endpoint Central und RMM Central vor allem bei Unternehmen verbreitet, die bereits andere Werkzeuge dieses Herstellers einsetzen. Welche Lösung passt, hängt von der Zahl der Geräte, den benötigten Integrationen – etwa in Ticketsystem und Dokumentation – und dem Preismodell ab.

Warum RMM statt WSUS, Intune oder MDM?

Eine naheliegende Frage lautet: Mit WSUS, Gruppenrichtlinien (GPOs) und Intune gibt es doch bereits Werkzeuge von Microsoft – wozu dann ein RMM? Die Antwort: Ein RMM ist kein grundsätzlicher Ersatz, aber in vielen Situationen die praktischere Ergänzung oder Alternative.

WSUS und **Gruppenrichtlinien** setzen voraus, dass die Geräte regelmäßig Kontakt zur Domäne bzw. zum lokalen Netz haben. Bei verteilten Standorten und viel Homeoffice ohne dauerhafte VPN-Verbindung greift das oft nicht zuverlässig – Geräte bleiben wochenlang ohne Updates oder neue Richtlinien. **Intune** und Apple-MDM-Lösungen funktionieren zwar über das Internet, erfordern aber eine vollständige Registrierung der Geräte (Enrollment) samt passender Lizenzen. Eine komplette Umgebung mit Intune, Autopilot und Entra ID aufzusetzen, ist für viele kleine und mittlere Unternehmen aufwendig.

Ein RMM-Agent kommuniziert dagegen direkt über das Internet, unabhängig von Domänenmitgliedschaft oder Standort. Er patcht nicht nur Windows, sondern auch macOS, Linux und *Drittanbieter-Software* wie Browser, PDF-Reader oder Java – WSUS beschränkt sich auf Microsoft-Produkte. Für KMU ist ein RMM daher oft günstiger und schneller eingerichtet. Und statt getrennter Werkzeuge für Monitoring, Patching und Fernwartung gibt es eine einzige Oberfläche.

Kriterium	WSUS / GPO	Intune / MDM	RMM
Benötigt Domäne oder LAN	Ja	Nein	Nein
Drittanbieter-Updates	Nein	Eingeschränkt	Ja
Plattformen	Windows	Windows, macOS, Mobilgeräte	Windows, macOS, Linux
Monitoring und Fernzugriff	Nein	Eingeschränkt	Integriert

MSP oder interne IT

Ein **MSP** betreut mit einem RMM viele Kunden gleichzeitig. Dafür ist die *Mandantenfähigkeit* zentral: Die Geräte, Richtlinien und Berichte jedes Kunden müssen sauber voneinander getrennt sein, und Techniker sollen nur die Kunden sehen, für die sie zuständig sind. Eine **interne IT** verwaltet dagegen die eigene Geräteflotte. Auch hier ist der Effizienzgewinn groß, insbesondere bei mehreren Standorten oder vielen mobilen Mitarbeitenden.

Sicherheit und Grenzen

Gerade die Stärke eines RMM ist zugleich sein größtes Risiko. Über die zentrale Konsole lassen sich Skripte mit höchsten Rechten auf allen angeschlossenen Geräten ausführen. Wird das RMM kompromittiert, sind auf einen Schlag sämtliche Endpunkte gefährdet. RMM-Systeme sind deshalb ein bekanntes und begehrtes Angriffsziel, und es hat bereits Angriffe gegeben, bei denen über kompromittierte RMM-Zugänge Schadsoftware bei vielen Unternehmen gleichzeitig verteilt wurde.

> **Merke**
>
> Ein RMM ist so mächtig wie ein Generalschlüssel für alle Geräte. MFA und starke Zugänge sind zwingend, Techniker erhalten nur die Rechte, die sie benötigen, alle Aktivitäten werden protokolliert und es werden nur geprüfte Skripte ausgerollt.

Onboarding, Automatisierung und Patching

Im Alltag beginnt die Arbeit mit dem *Onboarding*: Der Agent wird per Massenverteilung auf die Geräte gebracht, etwa über ein Installationspaket, eine bestehende Softwareverteilung oder beim Einrichten neuer Rechner. Danach werden die Geräte in *Gruppen* eingeteilt – nach Kunde, Standort, Abteilung oder Gerätetyp –, um Aktionen gezielt auszuführen. Eine gepflegte *Skript-Bibliothek* deckt wiederkehrende Aufgaben ab, und *Automatisierungs-Richtlinien* sorgen für Selbstheilung: Fällt ein wichtiger Dienst aus, startet das RMM ihn automatisch neu und erstellt nur dann ein Ticket, wenn das nicht hilft.

Beim **Patch-Management** gilt: Updates nicht auf alle Geräte gleichzeitig ausrollen. Bewährt hat sich ein Vorgehen in *Ringen* mit festen Zeitfenstern. Zunächst erhält ein kleiner Testring die Updates, etwa die Geräte der IT selbst. Treten keine Probleme auf, folgt der breite Produktivring. So bleibt ein fehlerhaftes Update auf wenige Geräte beschränkt. Das gilt für Betriebssystem- ebenso wie für Drittanbieter-Updates. Regelmäßige Berichte zum Patch-Stand zeigen, welche Geräte noch Nachholbedarf haben.

Aus der Praxis

Prüfen Sie den Patch-Bericht regelmäßig auf Geräte, die sich länger nicht gemeldet haben. Oft sind das Notebooks, die selten eingeschaltet werden – und gerade sie sind bei der nächsten Nutzung besonders verwundbar.

Das Wichtigste in Kürze

- RMM bedeutet Fernüberwachung und Fernverwaltung vieler Endpunkte per Agent.
- Kernfunktionen sind Monitoring, Patching, Fernzugriff und Automatisierung.
- Bekannte Anbieter sind u. a. NinjaOne, Datto, N-able, ConnectWise und ManageEngine.
- Bei verteilten Standorten und Homeoffice ist RMM oft praktikabler als WSUS, GPO oder MDM allein.
- Streng absichern mit MFA, Least Privilege, Protokollierung und Patch-Ringen.

ITSM & Ticketsysteme

IT-Service-Management mit System

Nach diesem Kapitel können Sie …

- erklären, was IT-Service-Management ist und welche Rolle ITIL dabei spielt,
- Incident, Problem, Service Request und Change sicher unterscheiden,
- Prioritäten aus Auswirkung und Dringlichkeit ableiten und realistische SLAs festlegen,
- die Aufgaben eines Ticketsystems und wichtige Kennzahlen beschreiben.

Was ITSM bedeutet

IT-Service-Management (ITSM) beschreibt den geordneten Betrieb von IT-Diensten. Statt dass Anfragen per Zuruf im Flur, per Anruf auf dem Handy oder per formloser E-Mail an einzelne Personen gehen, werden sie über definierte Prozesse erfasst, bearbeitet und abgeschlossen. Das hat zwei große Vorteile: Die Bearbeitung wird *nachvollziehbar* – jeder kann sehen, wer was wann getan hat –, und die Servicequalität wird *messbar*.

Das bekannteste Rahmenwerk für ITSM ist **ITIL** (Information Technology Infrastructure Library). Es sammelt bewährte Vorgehensweisen für den IT-Betrieb und definiert eine gemeinsame Sprache. ITIL ist kein starres Regelwerk, das vollständig umgesetzt werden muss; vielmehr übernehmen Unternehmen die Bausteine, die zu ihrer Größe und ihren Anforderungen passen.

Incident, Problem und Service Request

Eine der wichtigsten Unterscheidungen im ITSM – und ein häufiges Prüfungsthema – ist die zwischen Incident und Problem. Ein **Incident** ist eine konkrete Störung oder Qualitätsminderung eines Dienstes: Der Drucker in der Buchhaltung druckt nicht, eine Anwendung lässt sich nicht starten. Ziel des Incident-Managements ist es, den Dienst *so schnell wie möglich wiederherzustellen* – notfalls mit einer Umgehungslösung.

Ein **Problem** ist dagegen die dahinterliegende Ursache eines oder mehrerer Incidents. Fällt etwa jeden Montag der VPN-Zugang aus, sind das mehrere Incidents, aber womöglich ein einziges Problem. Das Problem-Management sucht die Ursache und beseitigt sie dauerhaft. Bis dahin werden bekannte Fehler mit ihren Umgehungslösungen als *Known Errors* dokumentiert, damit der Service Desk bei der nächsten Meldung sofort helfen kann.

Davon zu trennen ist der **Service Request**, eine Serviceanfrage. Dabei ist nichts kaputt, sondern jemand wünscht eine Standardleistung: einen neuen Benutzerzugang, eine Softwareinstallation oder einen zusätzlichen Monitor. Service Requests folgen meist festen, oft automatisierbaren Abläufen und sollten nicht mit Störungen vermischt werden, weil sie sonst die Kennzahlen verfälschen und echte Incidents verdrängen.

> **Merke**
>
> Incident heißt: den Betrieb schnell wiederherstellen. Problem heißt: die Ursache dauerhaft beseitigen. Erst wird der Betrieb gesichert, dann die Ursache gelöst.

Change Management

Viele Störungen entstehen nicht von selbst, sondern durch Änderungen: ein Update, eine neue Firewall-Regel, eine umgestellte Konfiguration. Das **Change Management** sorgt deshalb dafür, dass Änderungen an produktiven Systemen *geplant, geprüft und dokumentiert* erfolgen – nicht spontan.

Vor jeder Änderung werden Risiko und Auswirkung bewertet: Welche Systeme und Benutzer sind betroffen, was kann schiefgehen? Je nach Risiko ist eine Freigabe erforderlich, etwa durch die Teamleitung oder ein Change Advisory Board. Die Umsetzung erfolgt in einem abgestimmten *Wartungsfenster*, und es gibt immer einen *Rückfallplan* (Rollback), mit dem sich der vorherige Zustand wiederherstellen lässt. Abschließend wird die Änderung dokumentiert. Für häufige Änderungen mit geringem Risiko werden in der Regel *Standard-Changes* mit vereinfachter Freigabe definiert, damit der Prozess nicht zum Hemmschuh wird.

Ticketsysteme in der Praxis

Das zentrale Werkzeug des ITSM ist das **Ticketsystem**. Jede Anfrage – ob per E-Mail, Telefon, Portal oder automatisch aus dem Monitoring – wird zu einem Ticket mit Kategorie, Priorität, Zuständigkeit und Verlauf. Im Systemhaus-Umfeld verbreitet sind beispielsweise **TANSS** und **Halo**, daneben **Freshdesk** und **Jira Service Management**. Gemeinsam ist ihnen, dass sie SLAs und Prioritäten abbilden, Eskalationen auslösen und eine *Wissensdatenbank* integrieren, in der bekannte Lösungen gesammelt werden.

Weitere ITIL-Bausteine ergänzen das Ticketsystem. Ein *Service-Katalog* beschreibt, welche Leistungen die IT anbietet, und ein *Self-Service-Portal* ermöglicht es Benutzern, Standardanfragen selbst zu stellen und den Bearbeitungsstand einzusehen – oder Lösungen in der Wissensdatenbank direkt selbst zu finden. Eine **CMDB** (Configuration Management Database) bietet den Überblick über Assets und ihre Beziehungen: Welcher Server gehört zu welcher Anwendung, welche Benutzer sind betroffen, wenn ein bestimmtes System ausfällt?

Prioritäten und SLAs

Nicht jedes Ticket ist gleich wichtig. Die **Priorität** ergibt sich aus zwei Faktoren: der *Auswirkung* (Wie viele Benutzer oder wie wichtige Prozesse sind betroffen?) und der *Dringlichkeit* (Wie schnell entsteht Schaden?). Ein ausgefallener Arbeitsplatz ist dringend für die betroffene Person, hat aber geringe Auswirkung; ein Ausfall des Warenwirtschaftssystems betrifft das ganze Unternehmen und wird höchste Priorität erhalten.

Auswirkung / Dringlichkeit	Hoch	Mittel	Niedrig
Hoch (viele Benutzer)	Priorität 1	Priorität 2	Priorität 3
Mittel (Abteilung)	Priorität 2	Priorität 3	Priorität 4
Niedrig (Einzelperson)	Priorität 3	Priorität 4	Priorität 5

Jeder Priorität werden in den **SLAs** (Service Level Agreements) Reaktions- und Lösungszeiten zugeordnet, etwa eine Reaktion innerhalb von 30 Minuten bei Priorität 1. Diese Zusagen müssen realistisch und messbar sein – ein SLA, das regelmäßig gerissen wird, schadet dem Vertrauen mehr als eine ehrliche, längere Frist. Ebenso wichtig sind klar definierte *Eskalationswege*: Wer wird informiert, wenn ein Ticket seine Frist zu überschreiten droht?

Kennzahlen und gute Praxis

Ein Ticketsystem liefert wertvolle **Kennzahlen**. Die *Erstlösungsquote* zeigt, wie viele Anfragen schon beim ersten Kontakt gelöst werden. Die *MTTR* (Mean Time to Resolve bzw. Repair) gibt an, wie lange die Lösung im Durchschnitt dauert. Der *Backlog* zeigt die Menge offener Tickets, die *Wiedereröffnungsrate* deckt vorschnell geschlossene Tickets auf, und die *Kundenzufriedenheit* (CSAT) wird über kurze Umfragen nach Abschluss erfasst.

Aus der Praxis

Kennzahlen sind kein Selbstzweck. Nutzen Sie sie für kontinuierliche Verbesserung: Tauchen dieselben Incidents immer wieder auf, ist das ein Fall für das Problem-Management. Werden bestimmte Standardanfragen häufig gestellt, lohnt sich ein Eintrag im Self-Service-Portal. Klare Kategorien und Prioritäten sind die Voraussetzung, damit die Auswertungen aussagekräftig sind.

Das Wichtigste in Kürze

- ITSM ordnet den IT-Betrieb in Prozesse, ITIL ist das bekannteste Rahmenwerk.

- Incident heißt beheben, Problem heißt Ursache beseitigen, Service Requests werden getrennt behandelt.
- Changes werden geplant, freigegeben und immer mit Rückfallplan umgesetzt.
- Priorität ergibt sich aus Auswirkung und Dringlichkeit, SLAs müssen realistisch sein.
- Ticketsystem, Wissensdatenbank und Kennzahlen bilden die Grundlage für Verbesserung.

IT-Dokumentation

Was, womit und warum dokumentieren

Nach diesem Kapitel können Sie …

- begründen, warum eine gute IT-Dokumentation im Ernstfall unverzichtbar ist,
- festlegen, welche Inhalte in eine IT-Dokumentation gehören,
- geeignete Werkzeuge für Dokumentation und Zugangsdaten auswählen,
- die Dokumentation aktuell halten und angemessen absichern,
- Runbooks und ein Notfallhandbuch sinnvoll aufbauen.

Warum Dokumentation so wichtig ist

Dokumentation gehört zu den unbeliebtesten Aufgaben in der IT – und zu den wichtigsten. Ihr Wert zeigt sich meist erst dann, wenn etwas schiefgeht: Ein Server fällt aus, der zuständige Kollege ist im Urlaub, und niemand weiß genau, wie das System eingerichtet war, welche Dienste davon abhängen oder wo die Zugangsdaten für den Hersteller-Support liegen. Im Ernstfall fehlt die Zeit, alles neu herauszufinden. Ob ein Ausfall Minuten oder Stunden dauert, hängt dann oft allein davon ab, ob eine brauchbare Dokumentation existiert.

Gute Dokumentation macht das Unternehmen außerdem *unabhängig von einzelnen Köpfen*. Wissen, das nur in den Köpfen weniger Personen existiert, geht bei Krankheit, Kündigung oder Ruhestand verloren. Und für die *Einarbeitung* neuer Mitarbeitender ist eine strukturierte Dokumentation unbezahlbar: Sie verkürzt die Zeit, bis jemand selbstständig arbeiten kann, erheblich.

Was in die Dokumentation gehört

Eine vollständige IT-Dokumentation umfasst mehrere Bereiche. Dazu gehören *Netzwerkpläne und IP-Konzepte*: Welche Netze und VLANs gibt es, welche Adressbereiche werden wofür genutzt? Ebenso wichtig ist die *Serverlandschaft* mit ihren Rollen, wichtigen Konfigurationen und vor allem den *Abhängigkeiten* zwischen Systemen. Für den Notfall werden *Notfallkontakte* – intern, bei Dienstleistern und Herstellern – sowie *Wiederanlaufpläne* benötigt, die festlegen, in welcher Reihenfolge Systeme nach einem Ausfall wieder hochgefahren werden.

Auch *Zugänge* gehören dazu, allerdings nicht als Klartext in der Dokumentation selbst. Passwörter in einer Excel-Tabelle oder einem Wiki sind ein erhebliches Sicherheitsrisiko. Die Dokumentation verweist stattdessen auf den Eintrag in einem geschützten Passwort-

Manager.

Netzplan und Inventar

Zwei Bestandteile verdienen besondere Aufmerksamkeit. Der **Netzplan** zeigt alle wichtigen Geräte, ihre Verbindungen und die IP-Bereiche. Er beantwortet im Störfall sofort die Frage, was mit was zusammenhängt: Welcher Switch hängt an welchem Firewall-Port, über welche Leitung ist die Außenstelle angebunden?

Das **Inventar** listet Hard- und Software mit Standort, Seriennummer und verantwortlicher Person. In größeren Umgebungen wird es als *CMDB* (Configuration Management Database) geführt, die zusätzlich die Beziehungen zwischen den Komponenten abbildet. Ins Inventar gehören auch *Lizenzen und Wartungsverträge* mit ihren Laufzeiten – so fällt rechtzeitig auf, wenn eine Garantie abläuft oder ein Supportvertrag verlängert werden muss.

> **Merke**
>
> Ein aktueller Netzplan spart im Störfall die meiste Zeit, weil er sofort zeigt, was mit was zusammenhängt. Eine veraltete Dokumentation ist dagegen schlimmer als gar keine – weil man sich auf sie verlässt.

Werkzeuge für die Dokumentation

Für die IT-Dokumentation gibt es spezialisierte Werkzeuge. **IT Glue** und **Hudu** sind gezielt auf IT-Dokumentation zugeschnitten und vor allem im MSP-Umfeld verbreitet. Sie bieten vorgefertigte Strukturen für Geräte, Netzwerke, Zugänge und Abläufe und lassen sich mit RMM- und Ticketsystemen verbinden, sodass etwa Geräteinformationen automatisch übernommen werden. **Confluence** ist allgemeiner angelegt und eignet sich als Wiki für Wissen, Anleitungen und Projektdokumentation.

Für **Passwörter und Geheimnisse** gehört ein eigenes Werkzeug dazu: ein Passwort-Manager wie Bitwarden oder 1Password oder, für technische Geheimnisse wie API-Schlüssel, ein Vault. Diese Werkzeuge verschlüsseln die Daten, vergeben Zugriffsrechte fein abgestuft und protokollieren, wer wann auf welchen Eintrag zugegriffen hat. Für besonders kritische Konten, etwa globale Administratoren, empfiehlt sich das *Vier-Augen-Prinzip*. Notfall-Zugänge sollten zusätzlich versiegelt und getrennt aufbewahrt werden, etwa in einem Tresor, damit sie auch dann verfügbar sind, wenn der Passwort-Manager selbst nicht erreichbar ist.

Aktualität und Sicherheit

Die größte Gefahr für jede Dokumentation ist, dass sie veraltet. Ein Netzplan, der einen inzwischen ausgetauschten Switch zeigt, oder eine Anleitung mit falschen IP-Adressen führt im Ernstfall in die Irre. Die wirksamste Regel lautet deshalb: *Beim Ändern direkt mitpflegen.* Wer eine Konfiguration ändert, passt im selben Arbeitsschritt die Dokumentation an. Ergänzend sollte die Dokumentation regelmäßig überprüft werden, etwa einmal im Quartal. Und sie sollte so knapp wie möglich und so ausführlich wie nötig sein – ausufernde Texte werden weder gelesen noch gepflegt.

Gleichzeitig enthält die Dokumentation sensible Informationen über die gesamte Infrastruktur und ist damit für Angreifer äußerst wertvoll. Der Zugriff sollte nur für Befugte möglich sein, die Daten sollten verschlüsselt gespeichert und selbstverständlich ins Backup einbezogen werden.

Runbooks und Notfallhandbuch

Ein **Runbook** ist eine Schritt-für-Schritt-Anleitung für eine wiederkehrende Aufgabe – etwa das Anlegen eines neuen Benutzers, den Neustart eines bestimmten Dienstes oder die monatliche Kontrolle der Backups. Gute Runbooks sind so geschrieben, dass auch eine Vertretung die Aufgabe zuverlässig erledigen kann.

Das **Notfallhandbuch** regelt dagegen, wer bei einem größeren Ausfall was tut: Wer wird informiert, wer entscheidet, in welcher Reihenfolge werden Systeme wiederhergestellt? Es muss auch *offline* verfügbar sein, etwa ausgedruckt oder auf einem separaten Gerät – denn bei einem Ausfall oder einem Ransomware-Angriff ist das Wiki womöglich selbst nicht erreichbar. Kontaktlisten und Eskalationswege sind aktuell zu halten.

Aus der Praxis

Spielen Sie die Wiederherstellungsschritte aus dem Notfallhandbuch regelmäßig durch. Erst bei einer Übung zeigt sich, ob eine Anleitung vollständig ist, ob die hinterlegten Zugänge funktionieren und ob die angegebenen Ansprechpartner noch erreichbar sind.

Das Wichtigste in Kürze

- Dokumentation spart im Ernstfall enorm Zeit und macht unabhängig von einzelnen Köpfen.
- Hinein gehören Netzpläne, Server, Abhängigkeiten, Inventar, Verträge und Notfallpläne.
- Passwörter gehören in einen Passwort-Manager, nicht in Excel oder ein Wiki.
- Aktuell halten ist entscheidend: beim Ändern direkt mitpflegen.

- Dokumentation absichern und das Notfallhandbuch offline verfügbar halten.

LB-10

Scripting & Automatisierung

1. PowerShell – Grundlagen
2. PowerShell – Praxis für Admins
3. Bash & Linux-Scripting
4. Cron & Task-Scheduling
5. Python für IT-Admins
6. Git & Versionskontrolle
7. CI/CD-Grundlagen

PowerShell – Grundlagen

Sprachbausteine: Variablen, Logik, Schleifen, Funktionen

Nach diesem Kapitel können Sie …

- erklären, was PowerShell von einer klassischen Kommandozeile unterscheidet,
- Variablen, Bedingungen und Schleifen in eigenen Skripten einsetzen,
- einfache Funktionen mit Parametern schreiben,
- Fehler gezielt abfangen und Skripte sicher ausführen.

Mehr als eine Eingabeaufforderung

Auf den ersten Blick sieht **PowerShell** aus wie eine gewöhnliche Kommandozeile: Man tippt einen Befehl ein und erhält eine Ausgabe. Tatsächlich steckt dahinter eine vollwertige Skriptsprache mit Variablen, Verzweigungen, Schleifen und Funktionen. Befehlsfolgen lassen sich als Skriptdateien mit der Endung `.ps1` speichern und jederzeit erneut ausführen – die Grundlage jeder Automatisierung.

Der entscheidende Unterschied zu älteren Shells liegt in der **Objekt-Pipeline**. Klassische Kommandozeilen reichen zwischen zwei Befehlen nur Text weiter, den der nächste Befehl mühsam zerlegen muss. PowerShell übergibt dagegen strukturierte Objekte mit Eigenschaften. Ruft man etwa `Get-Process` auf, erhält man keine Textzeilen, sondern Prozessobjekte, deren Name, Speicherverbrauch oder Startzeit man direkt filtern und sortieren kann. Das macht Skripte kürzer und weniger fehleranfällig.

Unter Windows ist PowerShell fest eingebaut. Daneben gibt es die quelloffene Variante *PowerShell 7*, die auch unter Linux und macOS läuft. Die Grundlagen in diesem Kapitel gelten für beide Varianten gleichermaßen.

Variablen und Datentypen

Variablen erkennen Sie in PowerShell am vorangestellten Dollarzeichen. Eine Zuweisung wie `$name = 'Server01'` legt die Variable an und speichert den Text darin. Neben einfachen Zeichenketten (Strings) und Zahlen kennt PowerShell *Arrays*, also Listen mehrerer Werte, und *Hashtables*, in denen Werte unter einem Schlüssel abgelegt werden – vergleichbar mit einem kleinen Nachschlagewerk.

Eine Typangabe ist nicht zwingend nötig, PowerShell erkennt den passenden Datentyp in der Regel selbst. Wer sichergehen will, kann den Typ aber festlegen: `[int]$anzahl = 5` erzwingt eine Ganzzahl, und der Versuch, dort später einen Text abzulegen, führt zu einem Fehler. Gerade in längeren Skripten hilft das, Verwechslungen früh zu bemerken.

Daneben gibt es **automatische Variablen**, die PowerShell selbst befüllt. Die wichtigste ist `$_`: Sie steht in der Pipeline für das Objekt, das gerade verarbeitet wird.

```
$server  = @('Server01', 'Server02')        # Array
$config  = @{ Name = 'Web'; Port = 443 }    # Hashtable
[int]$anzahl = $server.Count
```

Bedingungen: den Ablauf steuern

Mit `if`, `elseif` und `else` trifft ein Skript Entscheidungen. Ungewohnt für Einsteiger sind die Vergleichsoperatoren: PowerShell nutzt nicht die Zeichen `=`, `>` oder `<`, sondern Kürzel mit Bindestrich. `-eq` steht für „gleich“, `-ne` für „ungleich“, `-gt` und `-lt` für „größer“ und „kleiner“. Mit `-like` lassen sich Muster mit Platzhaltern vergleichen, etwa alle Servernamen, die mit „SRV“ beginnen.

Mehrere Bedingungen verknüpfen Sie mit `-and` und `-or`, eine Bedingung kehren Sie mit `-not` um. Müssen viele mögliche Werte unterschieden werden, ist `switch` übersichtlicher als eine lange Kette von `elseif`-Zweigen.

```
if ($frei -lt 10) { 'Kritisch' }
elseif ($frei -eq 10) { 'Grenzwert' }
else { 'In Ordnung' }
```

> **Merke**
>
> Das Gleichheitszeichen ist in PowerShell ausschließlich eine Zuweisung. Für Vergleiche nutzen Sie immer die Operatoren mit Bindestrich wie `-eq`, `-gt` oder `-like`.

Schleifen und Funktionen

Schleifen wiederholen Anweisungen. Die `foreach`-Schleife durchläuft eine Liste Element für Element und ist im Admin-Alltag die häufigste Form, etwa um jeden Server einer Liste abzufragen. Die klassische `for`-Schleife zählt einen Zähler hoch und eignet sich, wenn eine feste Anzahl von Durchläufen gebraucht wird. `while` und `do … while` laufen dagegen so lange, wie eine Bedingung erfüllt ist. Direkt in der Pipeline übernimmt `ForEach-Object` diese Aufgabe und verarbeitet jedes ankommende Objekt einzeln.

Sobald ein Codeabschnitt mehrfach gebraucht wird, lohnt sich eine **Funktion**. Sie wird mit dem Schlüsselwort `function` eingeleitet, ihre Parameter stehen im `param(…)`-Block – idealerweise mit Typangabe. Eine Besonderheit: Ein `return` ist nicht zwingend nötig. Al-

les, was eine Funktion ausgibt, wird automatisch zu ihrem Rückgabewert. Für den Namen empfiehlt sich das Muster *Verb-Substantiv*, das auch alle eingebauten Befehle verwenden, also etwa `Get-Status` oder `Set-Kennwort`.

```
function Get-Status {
    param([string]$Name)
    Test-Connection -ComputerName $Name -Count 1 -Quiet
}
foreach ($s in $server) { "$s : $(Get-Status -Name $s)" }
```

Fehler abfangen

Ein Skript, das bei jedem kleinen Problem abbricht oder Fehler stillschweigend übergeht, ist im Betrieb wenig hilfreich. PowerShell bietet dafür den Block `try { … } catch { … } finally { … }`. Im `try`-Teil steht der eigentliche Code, im `catch`-Teil die Reaktion auf einen Fehler, und `finally` wird in jedem Fall ausgeführt – etwa um eine Verbindung sauber zu schließen.

Wichtig zu wissen: Viele Cmdlets erzeugen standardmäßig nur „nicht abbrechende“ Fehler, die ein `catch` gar nicht erreichen. Erst der Parameter `-ErrorAction Stop` macht daraus einen Fehler, der abgefangen werden kann. Die automatische Variable `$Error` enthält zudem die zuletzt aufgetretenen Fehler und hilft bei der Analyse. Geben Sie im `catch`-Block immer eine aussagekräftige Meldung aus, aus der hervorgeht, was genau nicht funktioniert hat.

Skripte sicher ausführen

Damit nicht jede heruntergeladene Datei einfach gestartet werden kann, gibt es die **Ausführungsrichtlinie** (Execution Policy). Mit `Get-ExecutionPolicy` sehen Sie die aktuelle Einstellung, mit `Set-ExecutionPolicy` ändern Sie sie. In sensiblen Umgebungen ist es üblich, nur *signierte* Skripte zuzulassen, deren Herkunft per Zertifikat nachgewiesen ist. Die Richtlinie ist allerdings eher ein Schutz vor versehentlichem Ausführen als eine echte Sicherheitsgrenze.

Ein zweiter wichtiger Schutzmechanismus ist der Parameter `-WhatIf`. Viele verändernde Cmdlets unterstützen ihn: Statt etwas zu löschen oder umzubenennen, zeigen sie nur an, was passieren würde. So testen Sie ein Skript gefahrlos, bevor es echte Änderungen vornimmt.

> **Aus der Praxis**
>
> Testen Sie jedes Skript, das Dateien, Konten oder Einstellungen verändert, zuerst mit `-WhatIf` und an einer kleinen Testmenge. Ein Fehler in einer Schleife wirkt sonst sofort auf hunderte Objekte.

Nützliche Cmdlets und gute Skript-Bausteine

Niemand muss alle Befehle auswendig kennen. `Get-Help` zeigt die Hilfe zu einem Cmdlet samt Beispielen, `Get-Command` findet Befehle anhand eines Stichworts. Besonders wertvoll ist `Get-Member`: Es zeigt, welche Eigenschaften und Methoden ein Objekt besitzt, und beantwortet damit die häufige Frage, worauf man eigentlich filtern kann. Für Dateien gibt es `Get-Content` und `Set-Content`, und `Test-Path` prüft, ob eine Datei oder ein Ordner existiert, bevor man darauf zugreift.

Gut lesbare Skripte enthalten Kommentare – einzeilig mit `#`, mehrzeilig im Block `<# … #>`. Parameter stehen typisiert im `param()`-Block am Anfang. Bei Befehlen mit vielen Parametern hilft **Splatting**: Die Parameter werden in einer Hashtable gesammelt und mit `@` statt `$` übergeben. Das hält lange Befehlszeilen übersichtlich.

```
$parameter = @{ Path = 'C:\Logs'; Filter = '*.log'; Recurse = $true }
Get-ChildItem @parameter
```

Das Wichtigste in Kürze

- PowerShell ist eine objektorientierte Skriptsprache, keine reine Text-Shell.
- Variablen, Bedingungen, Schleifen und Funktionen bilden die Basis jedes Skripts.
- Vergleiche laufen über Operatoren wie `-eq` und `-gt`.
- `try` / `catch` mit `-ErrorAction Stop` macht Skripte robust.
- Ausführungsrichtlinie und `-WhatIf` sorgen für Sicherheit.

PowerShell – Praxis für Admins

Automatisierung im echten Admin-Alltag

Nach diesem Kapitel können Sie …

- Benutzer und Gruppen im Active Directory per Skript verwalten,
- CSV-Dateien als Ein- und Ausgabe für Massenoperationen nutzen,
- Befehle per Remoting auf vielen Servern gleichzeitig ausführen,
- Skripte als geplante Aufgabe einrichten, protokollieren und absichern.

Vom Skript zur Automatisierung

Im vorigen Kapitel ging es um die Sprachbausteine von PowerShell. Ihren eigentlichen Wert zeigt die Sprache aber erst im Alltag der Administration: überall dort, wo dieselbe Aufgabe immer wieder anfällt. Fünfzig neue Benutzerkonten zum Ausbildungsbeginn, ein Bericht über alle Konten mit abgelaufenem Kennwort oder ein Update auf zwanzig Servern – per Mausklick dauert das Stunden und ist fehleranfällig. Ein Skript erledigt dieselbe Arbeit in Sekunden.

Dazu kommen zwei Vorteile, die oft unterschätzt werden. Ein Skript ist **reproduzierbar**: Es macht jedes Mal exakt dasselbe, ohne dass ein Häkchen vergessen wird. Und es ist **dokumentierbar**: Wer später wissen will, wie die Konten angelegt wurden, liest einfach das Skript. Im letzten Schritt wird aus dem Skript eine echte Automatisierung, indem es als geplante Aufgabe ohne menschliches Zutun läuft.

Active Directory per Skript verwalten

Das bekannteste Einsatzfeld ist das **Active Directory**, der Verzeichnisdienst, in dem Windows-Netzwerke Benutzer, Gruppen und Computer verwalten. Das Modul *ActiveDirectory*, das mit den Verwaltungswerkzeugen für Windows Server installiert wird, bringt passende Cmdlets mit. `Get-ADUser` fragt Konten ab, `New-ADUser` legt neue an, `Set-ADUser` ändert bestehende. Für Gruppen gibt es entsprechend `Add-ADGroupMember` und `Remove-ADGroupMember`, mit denen sich Mitgliedschaften pflegen lassen.

Weil die Cmdlets Objekte liefern, lassen sie sich gut kombinieren. Ein typisches Beispiel ist die Suche nach Konten, die sich lange nicht mehr angemeldet haben, und deren anschließende Deaktivierung. Gerade bei solchen verändernden Aktionen gilt die Regel aus dem Grundlagenkapitel: erst mit `-WhatIf` prüfen, dann ausführen.

CSV: Tabellendaten ein- und ausgeben

Die Daten für Massenoperationen kommen in der Praxis meist als Tabelle, zum Beispiel aus der Personalabteilung. Das Standardformat dafür ist **CSV** (Comma-Separated Values), eine einfache Textdatei mit einer Zeile pro Datensatz. `Import-Csv` liest eine solche Datei ein und macht aus jeder Zeile ein Objekt, dessen Eigenschaften den Spaltenüberschriften entsprechen. Mit `ForEach-Object` wird anschließend jede Zeile verarbeitet.

```
Import-Csv .\nutzer.csv -Delimiter ';' | ForEach-Object {
    New-ADUser -Name $_.Name -SamAccountName $_.Login -Department $_.Abteilung
}
```

Der umgekehrte Weg ist ebenso wichtig: `Export-Csv` schreibt Ergebnisse in eine Datei, die sich in jeder Tabellenkalkulation öffnen lässt. Mit `Select-Object` legen Sie vorher fest, welche Spalten der Bericht enthalten soll. Achten Sie bei deutschsprachigen Dateien auf das Trennzeichen – Excel erwartet hierzulande meist ein Semikolon. Wer echte Excel-Dateien statt CSV braucht, greift zum Modul *ImportExcel*, das dazu nicht einmal ein installiertes Excel voraussetzt.

> **Merke**
>
> Active-Directory-Cmdlets und CSV-Dateien sind eine mächtige Kombination: Die Tabelle liefert die Daten, das Skript setzt sie für jede Zeile gleichartig und fehlerfrei um.

Remoting: viele Server auf einmal

Wer mehrere Server betreut, möchte sich nicht auf jedem einzeln anmelden. PowerShell-Remoting nutzt dafür den Windows-Dienst **WinRM** (Windows Remote Management). Mit `Enter-PSSession` öffnen Sie eine interaktive Sitzung auf einem entfernten Rechner und arbeiten dort, als säßen Sie davor. `Invoke-Command` geht einen Schritt weiter: Es führt einen Befehl oder ein ganzes Skript auf vielen Hosts gleichzeitig aus und sammelt die Ergebnisse ein.

```
Invoke-Command -ComputerName SRV01, SRV02, SRV03 -ScriptBlock {
    Get-Service -Name Spooler
}
```

Damit das funktioniert, muss WinRM auf den Zielsystemen aktiviert sein. In Domänen geschieht das meist zentral per Gruppenrichtlinie. Weil Remoting einen sehr mächtigen Zugang darstellt, sollte er abgesichert werden: Zugriff nur für berechtigte Administra-

torgruppen, Firewall-Regeln auf die Verwaltungsnetze beschränken und außerhalb einer Domäne verschlüsselte Verbindungen per HTTPS nutzen. Richtig eingerichtet ist Remoting ideal für die Pflege ganzer Server-Flotten.

Geplante Aufgaben und nützliche Module

Echte Automatisierung entsteht, wenn ein Skript regelmäßig von selbst läuft – etwa jede Nacht ein Bericht oder jeden Montag eine Bereinigung. Unter Windows übernimmt das die Aufgabenplanung (Task Scheduler), die sich vollständig per PowerShell konfigurieren lässt. Mit `New-ScheduledTaskAction` und `New-ScheduledTaskTrigger` beschreiben Sie, was wann passieren soll, und `Register-ScheduledTask` legt die Aufgabe an. Das nächste Kapitel geht genauer auf die Zeitsteuerung ein.

Viele Aufgaben müssen Sie nicht selbst programmieren, weil es dafür fertige **Module** gibt. Neben *ImportExcel* ist *PSWindowsUpdate* ein Klassiker, mit dem sich Windows-Updates per Skript suchen, installieren und protokollieren lassen. Solche Module werden aus der *PowerShell Gallery* installiert, dem zentralen Verzeichnis für PowerShell-Erweiterungen, etwa mit `Install-Module ImportExcel`. Da jeder dort Module veröffentlichen kann, sollten Sie nur verbreitete, gepflegte und vertrauenswürdige Module verwenden und im Zweifel einen Blick in den Code werfen.

Fehlerbehandlung und Protokollierung

Ein Skript, das nachts ohne Aufsicht läuft, muss seine Fehler selbst melden können. Die Grundlage kennen Sie bereits: `try` / `catch` zusammen mit `-ErrorAction Stop`. Statt den Parameter bei jedem einzelnen Befehl anzugeben, können Sie das Verhalten über die Variable `$ErrorActionPreference` für das ganze Skript festlegen. Im `catch`-Block sollte eine aussagekräftige Meldung stehen, die auch Wochen später noch verständlich ist.

Für die Nachvollziehbarkeit eignet sich `Start-Transcript`: Es zeichnet einen kompletten Lauf mit allen Ausgaben und Fehlern in einer Textdatei auf, bis `Stop-Transcript` die Aufzeichnung beendet. So lässt sich am nächsten Morgen genau nachlesen, was in der Nacht passiert ist.

Geplant und signiert: Skripte absichern

Ein automatisiertes Skript läuft mit den Rechten des Kontos, unter dem es gestartet wird. Verwenden Sie deshalb nie Ihr persönliches Administratorkonto, sondern ein eigenes **Dienstkonto mit minimalen Rechten**, das nur genau das darf, was das Skript braucht. Setzen Sie außerdem die Ausführungsrichtlinie passend zur Umgebung, und signieren Sie Skripte in sensiblen Bereichen mit einem Zertifikat. Dann fällt sofort auf, wenn jemand den Inhalt nachträglich verändert hat.

Aus der Praxis

Legen Sie geplante Skripte in einem Ordner ab, in den nur Administratoren schreiben dürfen. Kann ein normaler Benutzer das Skript ändern, kann er über die geplante Aufgabe Befehle mit den Rechten des Dienstkontos ausführen lassen.

Das Wichtigste in Kürze

- Active-Directory-Cmdlets und CSV-Dateien ersetzen hunderte Einzelklicks.
- Remoting über WinRM skaliert Befehle auf viele Server gleichzeitig.
- Geplante Aufgaben machen aus Skripten echte Automatisierung.
- Module wie ImportExcel und PSWindowsUpdate gezielt und nur aus vertrauenswürdiger Quelle einsetzen.
- Protokollieren, mit Dienstkonto und minimalen Rechten laufen lassen, sensibel signieren.

Bash & Linux-Scripting

Robuste Shell-Skripte für Linux-Systeme

Nach diesem Kapitel können Sie …

- ein Bash-Skript anlegen, ausführbar machen und starten,
- Variablen, Bedingungen, Schleifen und Funktionen korrekt einsetzen,
- Textausgaben mit grep, sed und awk gezielt verarbeiten,
- Skripte so schreiben, dass sie bei Fehlern früh und sicher abbrechen.

Das erste Skript

Die **Bash** ist auf den meisten Linux-Systemen die Standard-Shell. Alles, was Sie interaktiv in die Kommandozeile eintippen, lässt sich auch in eine Datei schreiben und als Skript ausführen. Damit das System weiß, welcher Interpreter die Datei verarbeiten soll, beginnt jedes Skript mit einer *Shebang-Zeile*. Die Schreibweise `#!/usr/bin/env bash` sucht die Bash im aktuellen Suchpfad und funktioniert daher auch auf Systemen, auf denen sie an einem ungewöhnlichen Ort liegt.

Anschließend muss die Datei ausführbar gemacht werden, denn Linux startet nur Dateien mit gesetztem Ausführungsrecht. Danach wird das Skript mit vorangestelltem `./` gestartet – so weiß die Shell, dass die Datei im aktuellen Verzeichnis gemeint ist. Kommentare beginnen mit einer Raute und gehören in jedes Skript, das länger als ein paar Zeilen ist.

```
#!/usr/bin/env bash
# Zeigt den freien Speicherplatz an
df -h /

chmod +x skript.sh
./skript.sh
```

Variablen, Arrays und Anführungszeichen

Variablen werden in der Bash ohne Leerzeichen um das Gleichheitszeichen zugewiesen: `name="Server01"`. Schon ein einziges Leerzeichen führt dazu, dass die Bash `name` als Befehl interpretiert. Genutzt wird die Variable mit vorangestelltem Dollarzeichen als `$name` oder, wenn direkt Text folgt, in der eindeutigen Form `${name}`. Arrays werden in runden Klammern angelegt, etwa `liste=(a b c)`; auf das erste Element greifen Sie mit `${liste[0]}` zu.

Die wichtigste Regel im Umgang mit Variablen lautet: **immer in doppelte Anführungszeichen setzen**. Ohne Anführungszeichen zerlegt die Bash den Inhalt an Leerzeichen in einzelne Wörter. Enthält eine Variable etwa den Dateinamen „Bericht 2024.txt“, würde ein Befehl ohne Anführungszeichen zwei getrennte Dateien suchen – im schlimmsten Fall löscht ein `rm` dann die falschen Dateien.

> **Merke**
>
> Schreiben Sie Variablen grundsätzlich als `"$var"`. Das verhindert, dass Leerzeichen oder Sonderzeichen im Inhalt das Verhalten des Skripts unvorhersehbar verändern.

Bedingungen, Schleifen und Funktionen

Verzweigungen werden mit `if … then … fi` gebildet. Die eigentliche Prüfung steht in eckigen Klammern: `[ … ]` ist die klassische, überall verfügbare Form, `[[ … ]]` die erweiterte Bash-Variante, die robuster mit leeren Variablen umgeht und Mustervergleiche erlaubt. Zahlen vergleicht man mit Operatoren wie `-gt` (größer) oder `-eq` (gleich), Texte mit `=` und `!=`. Für viele mögliche Werte gibt es `case`, das deutlich übersichtlicher ist als verschachtelte Bedingungen.

Die `for`-Schleife durchläuft Listen und ist besonders praktisch in Verbindung mit Dateimustern, etwa um alle Logdateien eines Ordners zu verarbeiten. `while` läuft dagegen, solange eine Bedingung erfüllt ist, und eignet sich zum Beispiel zum zeilenweisen Lesen einer Datei.

```
for f in /var/log/*.log; do
  if [ "$(stat -c %s "$f")" -gt 10485760 ]; then
    echo "Groß: $f"
  fi
done
```

Funktionen fassen wiederverwendbare Abläufe zusammen, etwa `backup() { tar -czf "$1" "$2"; }`. Die übergebenen Argumente stehen innerhalb der Funktion als `$1`, `$2` und so weiter zur Verfügung. Variablen, die nur innerhalb der Funktion gelten sollen, deklarieren Sie mit `local` – sonst überschreiben sie womöglich gleichnamige Variablen im übrigen Skript. Anders als in vielen Programmiersprachen liefert `return` keinen beliebigen Wert, sondern nur einen Exit-Code zwischen 0 und 255.

Textwerkzeuge: die eigentliche Stärke

Unter Linux ist fast alles Text: Logdateien, Konfigurationen, Befehlsausgaben. Die Bash entfaltet ihre Stärke daher vor allem im Zusammenspiel mit klassischen Textwerkzeugen, die über die Pipe `|` verkettet werden. `grep` filtert Zeilen, die ein Muster enthalten.

`sed` ersetzt oder bearbeitet Text, etwa um in vielen Konfigurationsdateien einen Servernamen auszutauschen. `awk` verarbeitet Daten spaltenweise und kann rechnen. Ergänzt werden sie durch kleine Helfer: `cut` schneidet Spalten aus, `sort` sortiert, `uniq` entfernt Doppelte und zählt sie auf Wunsch, `wc` zählt Zeilen oder Wörter.

```
grep "Failed password" /var/log/auth.log | awk '{print $(NF-3)}' | sort | uniq -c |
sort -rn
```

Diese Zeile zeigt, welche IP-Adressen am häufigsten fehlgeschlagene SSH-Anmeldungen verursacht haben – eine typische Aufgabe, die ohne Skript mühsam wäre. Für Massenaktionen auf Dateien ist `find` in Verbindung mit `xargs` das passende Werkzeug. `tee` schreibt eine Ausgabe gleichzeitig auf den Bildschirm und in eine Datei, und `watch` führt einen Befehl wiederholt aus, um etwa eine Warteschlange zu beobachten.

Robuste Skripte

Standardmäßig arbeitet die Bash nach einem Fehler einfach weiter. Scheitert etwa der Wechsel in ein Verzeichnis, läuft der folgende Löschbefehl womöglich im falschen Ordner. Abhilfe schafft eine Zeile direkt nach der Shebang: `set -euo pipefail`. Die Option `-e` bricht das Skript beim ersten fehlgeschlagenen Befehl ab, `-u` meldet die Verwendung nicht gesetzter Variablen als Fehler, und `pipefail` sorgt dafür, dass auch ein Fehler mitten in einer Pipe erkannt wird. Das Skript scheitert damit früh und deutlich, statt still Schaden anzurichten.

Jeder Befehl liefert einen **Exit-Code**, den die Variable `$?` enthält: 0 bedeutet Erfolg, jeder andere Wert einen Fehler. Kurze Prüfungen lassen sich mit `befehl && echo ok || echo fehler` formulieren. Ergänzen Sie das um ein Logging mit Zeitstempel, damit sich später nachvollziehen lässt, wann was passiert ist.

Argumente und Sicherheit

Skripte werden flexibler, wenn sie Argumente annehmen. Die einzelnen Werte stehen in `$1`, `$2` usw., alle zusammen in `"$@"`. Für Optionen wie `-v` oder `-f datei` gibt es den eingebauten Befehl `getopts`. Mit `read` fragen Sie interaktiv Eingaben ab, und mit `${VAR:-default}` legen Sie einen Standardwert fest, falls eine Variable leer ist.

Aus Sicherheitssicht gilt neben dem konsequenten Quoten eine zweite Regel: Verwenden Sie `eval` niemals mit ungeprüften Eingaben. Der Befehl führt beliebigen Text als Code aus, und eine manipulierte Eingabe wird so zu einem eingeschleusten Befehl.

Aus der Praxis

Lassen Sie neue Skripte durch ein Prüfwerkzeug wie ShellCheck laufen. Es findet fehlende Anführungszeichen und typische Fallstricke, bevor sie im Betrieb Probleme verursachen.

Das Wichtigste in Kürze

- Jedes Skript beginnt mit einer Shebang-Zeile und wird mit `chmod +x` ausführbar.
- Variablen ohne Leerzeichen zuweisen und immer als `"$var"` verwenden.
- `grep`, `sed` und `awk` sind das Werkzeug für Textverarbeitung.
- `set -euo pipefail` und geprüfte Exit-Codes machen Skripte robust.

Cron & Task-Scheduling

Aufgaben zeitgesteuert und zuverlässig ausführen

Nach diesem Kapitel können Sie …

- Cron-Zeitpläne lesen und eigene Einträge in der Crontab anlegen,
- die häufigste Fehlerquelle – die minimale Cron-Umgebung – vermeiden,
- geplante Aufgaben unter Windows per PowerShell einrichten,
- systemd-Timer als moderne Alternative einordnen und Jobs zuverlässig gestalten.

Warum Aufgaben planen?

Viele Arbeiten in der IT wiederholen sich in festen Abständen: Backups laufen jede Nacht, alte Logdateien sollen wöchentlich gelöscht werden, ein Bericht wird jeden Montag verschickt, Updates werden am Wochenende eingespielt. Solche Aufgaben von Hand zu erledigen ist nicht nur lästig, sondern auch unzuverlässig – irgendwann wird es vergessen. Mit einem **Scheduler**, also einem Zeitplandienst, laufen sie automatisch.

Ein solcher Dienst arbeitet unabhängig davon, ob gerade jemand am System angemeldet ist. Er startet die Aufgabe zum festgelegten Zeitpunkt, und bei sauberer Einrichtung lässt sich anhand von Logdateien jederzeit nachvollziehen, wann was gelaufen ist und ob es erfolgreich war. Unter Linux ist *Cron* der Klassiker, unter Windows die *Aufgabenplanung*, und auf modernen Linux-Systemen kommen zunehmend *systemd-Timer* zum Einsatz.

Die Cron-Syntax verstehen

Ein Cron-Eintrag besteht aus fünf Zeitfeldern, gefolgt vom auszuführenden Befehl. Die Felder stehen der Reihe nach für **Minute**, **Stunde**, **Tag des Monats**, **Monat** und **Wochentag**. Ein Stern bedeutet „jeder Wert“. Mit einem Schrägstrich geben Sie Intervalle an, mit Komma Aufzählungen und mit Bindestrich Bereiche. Beim Wochentag steht 0 (oder 7) für Sonntag, 1 für Montag.

Eintrag	Bedeutung
`0 2 * * *`	Täglich um 02:00 Uhr
`*/15 * * * *`	Alle 15 Minuten
`0 3 * * 1`	Montags um 03:00 Uhr
`30 7 1 * *`	Am Ersten jedes Monats um 07:30 Uhr

Für häufige Fälle gibt es außerdem Kurzformen. Besonders nützlich ist `@reboot`: Ein so markierter Befehl wird einmal beim Systemstart ausgeführt. Daneben existieren Kürzel wie `@daily` oder `@weekly`, die dasselbe bedeuten wie die entsprechenden Feldangaben.

Die Crontab verwalten

Jeder Benutzer hat eine eigene Zeitplantabelle, die **Crontab**. Mit `crontab -e` öffnen Sie sie im Editor, mit `crontab -l` lassen Sie sich die vorhandenen Einträge anzeigen. Die Befehle laufen dann mit den Rechten dieses Benutzers. Systemweite Aufgaben liegen dagegen in der Datei `/etc/crontab` und in den Verzeichnissen unter `/etc/cron.*`. In `/etc/cron.d` legen Softwarepakete und Administratoren eigene Dateien ab; dort steht zwischen Zeitangabe und Befehl zusätzlich der Benutzer, unter dem der Job laufen soll. Skripte in Ordnern wie `/etc/cron.daily` werden ohne eigene Zeitangabe einmal täglich ausgeführt.

Ein Cron-Job läuft unsichtbar im Hintergrund. Leiten Sie deshalb seine Ausgaben immer in eine Logdatei um, damit Sie bei Problemen etwas nachlesen können. Die Angabe `2>&1` sorgt dafür, dass auch Fehlermeldungen in der Datei landen. Alternativ verschickt Cron Ausgaben per E-Mail an die Adresse, die in der Variable `MAILTO` am Anfang der Crontab steht – vorausgesetzt, auf dem System ist ein Mailversand eingerichtet.

```
MAILTO=admin@example.com
0 2 * * * /usr/local/bin/backup.sh >> /var/log/backup.log 2>&1
```

Die häufigste Fehlerquelle: die Cron-Umgebung

Ein Satz fällt in der Administration immer wieder: „Das Skript läuft von Hand, aber nicht per Cron.“ Die Ursache ist fast immer dieselbe. Cron startet Befehle mit einer **minimalen Umgebung**. Die Variable `$PATH`, die festlegt, wo nach Programmen gesucht wird, enthält dort oft nur wenige Standardverzeichnisse. Befehle, die in der eigenen Shell problemlos gefunden werden, sind für Cron dann schlicht unbekannt. Auch weitere Variablen aus den persönlichen Profildateien fehlen, und das Arbeitsverzeichnis ist nicht das, aus dem Sie das Skript sonst starten.

Die Lösung ist einfach: Verwenden Sie in Cron-Jobs und den aufgerufenen Skripten immer **absolute Pfade**, also `/usr/bin/rsync` statt nur `rsync`. Benötigte Variablen setzen Sie entweder im Skript selbst oder am Anfang der Crontab.

> **Merke**
>
> Cron kennt Ihre gewohnte Shell-Umgebung nicht. Absolute Pfade und selbst gesetzte Variablen verhindern den Klassiker „läuft manuell, aber nicht per Cron“.

Windows: die Aufgabenplanung

Das Gegenstück unter Windows ist die **Aufgabenplanung** (Task Scheduler). Sie lässt sich grafisch bedienen, wird aber besser per PowerShell eingerichtet, weil sich die Konfiguration dann dokumentieren und auf andere Server übertragen lässt. Eine Aufgabe besteht aus einer Aktion (welches Programm oder Skript), einem oder mehreren Auslösern (Trigger) und dem Konto, unter dem sie läuft. Als Auslöser kommen nicht nur Uhrzeiten infrage, sondern auch die Anmeldung eines Benutzers oder ein bestimmtes Ereignis im Ereignisprotokoll.

```
$aktion  = New-ScheduledTaskAction -Execute 'powershell.exe' -Argument '-File
C:\Skripte\bericht.ps1'
$trigger = New-ScheduledTaskTrigger -Daily -At 2am
Register-ScheduledTask -TaskName 'Nachtbericht' -Action $aktion -Trigger $trigger
```

Wie bei Cron gilt: Die Aufgabe sollte unter einem eigenen **Dienstkonto mit minimalen Rechten** laufen, nicht unter einem persönlichen Administratorkonto.

systemd-Timer: die moderne Alternative

Auf aktuellen Linux-Distributionen bietet systemd eine eigene Zeitsteuerung. Ein **systemd-Timer** besteht aus zwei Dateien: einer *Service-Unit*, die beschreibt, was ausgeführt wird, und einer *Timer-Unit*, die festlegt, wann das geschieht. Der Zeitplan steht in der Zeile `OnCalendar=`, etwa `OnCalendar=*-*-* 02:00:00` für täglich um zwei Uhr. Mit `systemctl list-timers` sehen Sie alle Timer samt nächstem Ausführungszeitpunkt.

Der größte Vorteil gegenüber Cron sind die Logs: Jede Ausführung landet automatisch im Journal und lässt sich mit `journalctl -u name.service` nachlesen. Zudem kann ein Timer mit `Persistent=true` einen verpassten Lauf nachholen, wenn der Rechner zum geplanten Zeitpunkt ausgeschaltet war.

Jobs zuverlässig machen

Ein häufiges Problem sind sich **überlappende Läufe**: Ein Backup alle 15 Minuten dauert an einem Tag länger als 15 Minuten, und schon startet der nächste Lauf, während der erste noch arbeitet. Das Werkzeug `flock` verhindert das, indem es eine Sperrdatei setzt und einen zweiten Start ablehnt, solange der erste läuft.

Ebenso wichtig ist eine **Benachrichtigung bei Fehlern**. Ein still gescheitertes Backup fällt sonst erst auf, wenn man es braucht. Beachten Sie schließlich die Zeitzone des Systems: Server laufen oft auf UTC, sodass „zwei Uhr“ im Zeitplan nicht der deutschen Ortszeit entspricht. Auch Umstellungen auf Sommer- und Winterzeit können dazu führen, dass Jobs in der Nacht doppelt oder gar nicht laufen.

Aus der Praxis

Starten Sie lange Cron-Jobs mit `flock -n /run/backup.lock /usr/local/bin/backup.sh`. Läuft bereits eine Instanz, beendet sich der neue Aufruf sofort, statt das System doppelt zu belasten.

Das Wichtigste in Kürze

- Cron nutzt fünf Zeitfelder: Minute, Stunde, Tag, Monat, Wochentag.
- Die Cron-Umgebung ist minimal – absolute Pfade verwenden und Ausgaben protokollieren.
- Unter Windows richten Sie die Aufgabenplanung am besten per PowerShell ein.
- systemd-Timer bieten bessere Logs und holen verpasste Läufe nach.
- Überlappungen mit `flock` verhindern und bei Fehlern benachrichtigen.

Python für IT-Admins

Wenn Shell-Skripte an ihre Grenzen stoßen

Nach diesem Kapitel können Sie …

- einschätzen, wann Python die bessere Wahl ist als Bash oder PowerShell,
- die Grundsyntax von Python lesen und einfache Skripte schreiben,
- Projekte mit virtuellen Umgebungen und `requirements.txt` sauber aufsetzen,
- typische Bibliotheken für Admin-Aufgaben benennen und Skripte sicher gestalten.

Warum Python?

Bash und PowerShell sind hervorragend, um Befehle zu verketten und das eigene Betriebssystem zu steuern. Sobald die Logik aber komplexer wird – Daten aus mehreren Quellen zusammenführen, eine Web-Schnittstelle abfragen, verschachtelte Strukturen auswerten, Fehler differenziert behandeln –, werden Shell-Skripte schnell unübersichtlich. Hier spielt **Python** seine Stärken aus.

Python gilt als besonders gut lesbar: Der Code sieht fast aus wie strukturierter Pseudocode, was die Pflege durch Kolleginnen und Kollegen erleichtert. Die Sprache läuft gleichermaßen unter Windows, Linux und macOS, sodass ein Skript ohne Anpassung auf verschiedenen Systemen funktionieren kann. Der vielleicht größte Vorteil ist das **Ökosystem**: Für fast jede Aufgabe gibt es bereits eine fertige, gut dokumentierte Bibliothek, von der Excel-Datei bis zur Netzwerkverbindung.

Die Grundsyntax

Die auffälligste Besonderheit von Python ist, dass **Einrückung Teil der Syntax** ist. Während andere Sprachen Codeblöcke mit geschweiften Klammern abgrenzen, erkennt Python sie allein an der Einrückung. Alles, was nach einem `if` oder in einer Schleife um vier Leerzeichen eingerückt ist, gehört zu diesem Block. Das zwingt zu sauber formatiertem Code, sorgt bei Einsteigern aber anfangs für Fehlermeldungen, wenn Tabulatoren und Leerzeichen gemischt werden.

Variablen werden ohne Typdeklaration und ohne vorangestelltes Sonderzeichen angelegt: `server = "web01"`. Für Verzweigungen gibt es `if`, `elif` und `else`, für Wiederholungen `for` und `while`. Funktionen werden mit dem Schlüsselwort `def` definiert.

```
def pruefe_platte(prozent):
    if prozent > 90:
        return "kritisch"
    elif prozent > 75:
        return "Warnung"
    return "ok"

for wert in [40, 80, 95]:
    print(wert, pruefe_platte(wert))
```

Virtuelle Umgebungen und Pakete

Python-Projekte nutzen häufig zusätzliche Pakete. Würden alle Projekte ihre Pakete in dieselbe, systemweite Python-Installation laden, käme es früher oder später zu Konflikten – ein Skript braucht Version 1 einer Bibliothek, ein anderes Version 2. Die Lösung ist die **virtuelle Umgebung**: ein eigener, isolierter Ordner pro Projekt, in dem genau die Pakete liegen, die dieses Projekt braucht. Auf vielen aktuellen Linux-Systemen ist das sogar Pflicht, weil die Distribution die Installation in das System-Python blockiert.

Angelegt wird sie mit `python -m venv .venv`, danach wird sie aktiviert. Pakete installieren Sie mit `pip install`. Damit ein Skript später auf einem anderen Rechner genauso läuft, halten Sie die benötigten Pakete samt Versionen in der Datei `requirements.txt` fest. Diese Datei gehört ins Projekt, der Ordner `.venv` selbst dagegen nicht – er lässt sich jederzeit neu erzeugen.

```
python -m venv .venv
source .venv/bin/activate        # Windows: .venv\Scripts\activate
pip install requests psutil
pip freeze > requirements.txt
pip install -r requirements.txt  # auf dem Zielsystem
```

Der Werkzeugkasten für Admins

Für die IT-Administration haben sich einige Bibliotheken als feste Größen etabliert. Mit **requests** sprechen Sie Web-Schnittstellen (REST-APIs) an – etwa um Daten aus einem Monitoring- oder Ticketsystem abzurufen oder dort automatisch einen Eintrag anzulegen. **paramiko** stellt SSH-Verbindungen her und ermöglicht es, Befehle auf entfernten Servern auszuführen oder Dateien zu übertragen.

pandas ist das Standardwerkzeug für die Auswertung tabellarischer Daten: Logauszüge, Inventarlisten oder Messwerte lassen sich damit filtern, gruppieren und zusammenfassen. **psutil** liefert System- und Prozessinformationen wie CPU-Last, Speicherbelegung oder laufende Prozesse – plattformübergreifend über eine einheitliche Schnittstelle. Für Excel-Dateien gibt es **openpyxl**, und das bereits eingebaute Modul **subprocess** ruft vorhandene Systembefehle auf und wertet deren Ausgabe aus.

Typische Praxisbeispiele

Im Admin-Alltag tauchen immer wieder ähnliche Aufgaben auf, für die Python sich anbietet. Ein Klassiker ist das *Auswerten von Logdateien*: Ein Skript liest die Logs mehrerer Server, zählt Fehler nach Typ und erzeugt daraus einen übersichtlichen Bericht. Ebenso häufig ist das *Abfragen von REST-APIs*, etwa um alle offenen Tickets einer Kategorie abzurufen oder Warnungen aus dem Monitoring weiterzuverarbeiten.

Mit paramiko lassen sich *Server per SSH konfigurieren*, zum Beispiel um auf zwanzig Maschinen denselben Konfigurationsschritt auszuführen und das Ergebnis zu prüfen. Und schließlich entstehen viele *wiederkehrende Reports* als CSV- oder Excel-Datei, die per geplanter Aufgabe jede Woche automatisch erzeugt und verschickt werden.

Aus der Praxis

Fangen Sie klein an: Ein Skript, das mit psutil die Festplattenbelegung prüft und bei Überschreitung eines Schwellwerts eine Meldung schreibt, ist in wenigen Zeilen erledigt und lässt sich später schrittweise erweitern.

Struktur und gute Praxis

Die wichtigste Sicherheitsregel lautet: **keine Zugangsdaten im Code**. Passwörter, API-Schlüssel und Tokens werden aus Umgebungsvariablen oder einem Passwort-Tresor gelesen, etwa mit `os.environ["API_TOKEN"]`. Ein im Skript hinterlegtes Kennwort landet sonst früher oder später in einer Versionsverwaltung, einer E-Mail oder einem Backup – und ist damit für viele Personen lesbar.

Fehler fangen Sie mit `try` / `except` ab, und zwar möglichst gezielt für die erwarteten Fehlerarten, etwa eine fehlgeschlagene Netzwerkverbindung. Statt Meldungen mit `print` auszugeben, nutzen Sie das eingebaute Modul `logging`, das Zeitstempel und Schweregrade mitbringt und sich in Dateien umleiten lässt. Für Kommandozeilen-Argumente gibt es `argparse`, das automatisch eine Hilfeseite erzeugt. *Type Hints* wie `def pruefe(prozent: int) -> str:` verändern das Verhalten nicht, machen den Code aber deutlich lesbarer.

Halten Sie Skripte klein und testbar, trennen Sie Module und Importe sauber, und verwalten Sie alles mit Git. Als Einstiegspunkt eines Skripts hat sich der Block `if __name__ == "__main__":` bewährt: Der Code darin wird nur ausgeführt, wenn die Datei direkt gestartet wird – nicht, wenn ein anderes Skript ihre Funktionen importiert.

> **Merke**
>
> Geheimnisse wie Passwörter und API-Schlüssel gehören nie in den Quellcode. Lesen Sie sie aus Umgebungsvariablen oder einem Tresor.

Das Wichtigste in Kürze

- Python eignet sich für komplexere Logik, Datenverarbeitung und API-Anbindungen.
- Einrückung ist Teil der Syntax; Funktionen werden mit `def` definiert.
- Jedes Projekt bekommt eine eigene `venv`, Pakete stehen in `requirements.txt`.
- requests, paramiko, pandas und psutil bilden den Kern des Admin-Werkzeugkastens.
- Secrets nie hart codieren, Fehler abfangen, mit `logging` protokollieren.

Git & Versionskontrolle

Änderungen nachvollziehbar verwalten – auch für Configs

Nach diesem Kapitel können Sie …

- erklären, warum Versionskontrolle auch für Skripte und Konfigurationen wichtig ist,
- den Grund-Workflow aus `add`, `commit` und `push` anwenden,
- mit Branches arbeiten und Merge von Rebase unterscheiden,
- Änderungen sicher rückgängig machen und Geheimnisse aus dem Repository fernhalten.

Warum Versionskontrolle?

Wer Skripte oder Konfigurationsdateien ohne Versionskontrolle pflegt, kennt das Ergebnis: Dateien wie `backup_alt.sh`, `backup_neu.sh` und `backup_final2.sh`, und niemand weiß mehr genau, welche Version gerade produktiv läuft. **Git** löst dieses Problem. Es protokolliert jede Änderung mit Zeitpunkt, Urheber und einer kurzen Beschreibung, und es macht jede dieser Änderungen bei Bedarf wieder rückgängig.

Das lohnt sich auch dann, wenn Sie allein arbeiten: Die Historie beantwortet zuverlässig die Frage, was sich seit gestern geändert hat. Im Team kommt der zweite große Vorteil hinzu – mehrere Personen können an denselben Dateien arbeiten, ohne sich gegenseitig Änderungen zu überschreiben oder Kopien per E-Mail hin- und herzuschicken. Versionskontrolle ist daher längst nicht mehr nur ein Thema der Softwareentwicklung, sondern ebenso wertvoll für Admin-Skripte und Konfigurationen.

Der Grund-Workflow

Ein Git-Projekt heißt **Repository**. Ein neues legen Sie mit `git init` an, ein bestehendes von einem Server holen Sie mit `git clone`. Danach besteht die tägliche Arbeit im Kern aus drei Schritten. Mit `git add` wählen Sie aus, welche Änderungen in die nächste Version aufgenommen werden sollen. `git commit` speichert diese Auswahl als neuen Versionsstand, einen sogenannten *Commit*, zusammen mit einer Nachricht. Mit `git push` laden Sie Ihre Commits auf den gemeinsamen Server hoch.

In die andere Richtung holt `git pull` die Änderungen anderer auf Ihren Rechner. Den aktuellen Zustand zeigt jederzeit `git status`: welche Dateien verändert, welche bereits vorgemerkt und welche Git noch unbekannt sind.

```
git status
git add backup.sh
git commit -m "Backup: Aufbewahrung von 7 auf 14 Tage erhöht"
git push
```

Besonders wichtig sind **aussagekräftige Commit-Nachrichten**. „Änderungen“ oder „Fix“ helfen in einem halben Jahr niemandem weiter. Eine gute Nachricht beschreibt kurz, was geändert wurde und warum. Bewährt haben sich zudem kleine, häufige Commits: Jeder Commit enthält genau eine zusammengehörige Änderung, was die Historie lesbar macht und gezieltes Zurückgehen erleichtert.

Branches: parallel arbeiten

Ein **Branch** (Zweig) ist eine eigene Entwicklungslinie innerhalb des Repositorys. Der Hauptzweig – meist `main` genannt – enthält den stabilen Stand. Für eine neue Funktion oder ein größeres Umbauvorhaben legen Sie mit `git branch` einen eigenen Zweig an und wechseln mit `git switch` hinein. Dort können Sie experimentieren, ohne den Hauptzweig zu stören. Ist die Arbeit fertig und getestet, wird der Zweig in den Hauptzweig zusammengeführt.

Für das Zusammenführen gibt es zwei Wege. Ein **Merge** verbindet die beiden Zweige und behält die Historie beider vollständig bei; dabei entsteht in der Regel ein zusätzlicher Merge-Commit. Ein **Rebase** setzt die Commits des eigenen Zweigs dagegen neu auf den aktuellen Stand des Hauptzweigs auf. Das Ergebnis ist eine lineare, aufgeräumte Historie – allerdings werden dabei Commits neu geschrieben.

Verfahren	Ergebnis	Empfehlung
Merge	Historie beider Zweige bleibt erhalten	Sicherer Standard, gut für Einsteiger
Rebase	Lineare, aufgeräumte Historie	Nur auf eigenen, noch nicht geteilten Zweigen

> **Merke**
>
> Verwenden Sie Rebase nie auf Zweigen, die andere bereits heruntergeladen haben. Weil Rebase Commits neu schreibt, passt deren Stand danach nicht mehr zu Ihrem, und es entsteht Chaos.

Konflikte und Rückgängigmachen

Haben zwei Personen dieselbe Zeile einer Datei unterschiedlich geändert, kann Git nicht selbst entscheiden, welche Version richtig ist. Es meldet einen **Merge-Konflikt** und markiert die betroffenen Stellen in der Datei. Sie lösen den Konflikt manuell, indem Sie

die gewünschte Fassung herstellen, die Markierungen entfernen und das Ergebnis committen.

Um eine fehlerhafte Änderung zurückzunehmen, ist `git revert` das Mittel der Wahl. Es erzeugt einen neuen Commit, der die Wirkung eines früheren genau umkehrt – die Historie bleibt dabei vollständig und ehrlich erhalten. `git reset` setzt den Stand dagegen auf einen früheren Commit zurück und kann Historie verwerfen. Es sollte daher nur lokal und mit Bedacht eingesetzt werden, niemals auf bereits geteilten Commits. Wer mitten in einer Arbeit kurz etwas anderes erledigen muss, parkt seine noch nicht fertigen Änderungen mit `git stash` und holt sie später mit `git stash pop` zurück.

Plattformen, Zusammenarbeit und .gitignore

Für die gemeinsame Arbeit wird ein zentrales Repository auf einer Plattform wie **GitHub**, **GitLab** oder **Azure DevOps** gehostet; GitLab lässt sich auch im eigenen Rechenzentrum betreiben. Diese Plattformen bieten mit *Pull Requests* (bei GitLab *Merge Requests*) einen geordneten Weg zur Prüfung: Bevor ein Zweig in den Hauptzweig einfließt, sehen sich Kolleginnen oder Kollegen die Änderungen an und geben sie frei. Mit *Tags* markieren Sie wichtige Stände dauerhaft, etwa eine freigegebene Version.

Eine Datei verdient besondere Aufmerksamkeit: die `.gitignore`. Darin stehen Dateien und Ordner, die Git ignorieren soll – temporäre Dateien, Logs, virtuelle Umgebungen und vor allem Dateien mit Zugangsdaten. Denn was einmal committet wurde, bleibt in der Historie erhalten, selbst wenn die Datei später gelöscht wird.

```
# .gitignore
*.log
.venv/
.env
secrets/
```

Aus der Praxis

Ist doch einmal ein Passwort oder API-Schlüssel in ein Repository gelangt, genügt Löschen nicht. Betrachten Sie das Geheimnis als kompromittiert und tauschen Sie es sofort aus.

Git für Konfigurationen

Für Administratoren ist Git besonders bei **Konfigurationsdateien** wertvoll. Liegen etwa nginx-Konfigurationen, systemd-Units oder Ansible-Playbooks unter Versionskontrolle, ist jede Änderung dokumentiert und lässt sich zurückrollen. Machen Sie es sich zur Gewohnheit, vor jeder riskanten Änderung einen Commit des funktionierenden Zu-

stands anzulegen. Geht danach etwas schief, kehren Sie gezielt zu diesem Stand zurück, statt unter Zeitdruck aus dem Gedächtnis zu rekonstruieren, wie die Datei vorher aussah.

Strukturieren Sie Repositories dabei sinnvoll – etwa ein Repository pro Dienst oder pro Serverrolle – und dokumentieren Sie in einer kurzen Beschreibungsdatei, wofür das Repository dient.

Das Wichtigste in Kürze

- Git dokumentiert, wer wann was geändert hat, und macht Änderungen umkehrbar.
- Der Kern-Workflow lautet `add` → `commit` → `push`.
- Branches ermöglichen paralleles Arbeiten; Merge ist sicher, Rebase räumt auf.
- `git revert` macht Fehler rückgängig, ohne die Historie umzuschreiben.
- Secrets per `.gitignore` fernhalten – niemals Zugangsdaten committen.

CI/CD-Grundlagen

Automatisiert testen und ausrollen

Nach diesem Kapitel können Sie …

- die Begriffe Continuous Integration, Continuous Delivery und Continuous Deployment unterscheiden,
- den Aufbau einer Pipeline mit Build, Test und Deploy beschreiben,
- gängige Werkzeuge wie GitHub Actions, GitLab CI/CD und Jenkins einordnen,
- Grundregeln für sichere und verlässliche Deployments anwenden.

Was CI/CD bedeutet

In vielen Unternehmen lief das Ausrollen neuer Software lange so ab: Jemand kopierte Dateien von Hand auf den Server, startete Dienste neu und hoffte, dass nichts vergessen wurde. Je seltener und größer solche Updates waren, desto höher war das Risiko. **CI/CD** ersetzt diese Handarbeit durch einen automatisierten, immer gleichen Ablauf.

Continuous Integration (CI) bedeutet, dass Änderungen laufend in den gemeinsamen Stand zusammengeführt und dabei automatisch gebaut und getestet werden. Fehler fallen so kurz nach ihrer Entstehung auf, solange noch klar ist, welche Änderung sie verursacht hat. Das Kürzel **CD** steht für zwei verwandte Begriffe: Bei *Continuous Delivery* wird jede erfolgreich getestete Version automatisch so vorbereitet, dass sie jederzeit ausgerollt werden kann – den letzten Schritt löst ein Mensch aus. Bei *Continuous Deployment* geht auch dieser Schritt automatisch, und jede Änderung, die alle Prüfungen besteht, landet direkt in der Produktion.

Das Ziel ist in beiden Fällen dasselbe: weniger manuelle Fehler und ein schnelleres, verlässlicheres Ausrollen. Für die Administration ist das nicht nur bei Programmcode interessant, sondern ebenso bei Konfigurationen, Skripten und Infrastrukturbeschreibungen.

Die Pipeline

Das Herzstück ist die **Pipeline**: ein automatisierter Ablauf, der in der Regel durch einen Git-Push ausgelöst wird. Sie gliedert sich in aufeinanderfolgende Phasen, sogenannte *Stages*. Typisch ist die Folge **Build → Test → (Staging) → Deploy**. Im Build wird aus dem Quellcode ein lauffähiges Ergebnis erzeugt, etwa ein Programm oder ein Container-Image. Im Test wird dieses Ergebnis automatisch geprüft. Optional folgt das Ausrollen auf eine Staging-Umgebung, die der Produktion möglichst genau entspricht. Erst danach geht es in den Produktivbetrieb.

Entscheidend ist, dass **jede Phase die Pipeline stoppen kann**. Schlägt ein Test fehl, wird nichts ausgerollt. Nur Code, der alle Prüfungen bestanden hat, gelangt weiter. Alle Läufe werden protokolliert, sodass sich jederzeit nachvollziehen lässt, welche Version wann mit welchem Ergebnis gebaut und ausgerollt wurde.

Zwischen den Stages werden *Artefakte* weitergegeben – also die Ergebnisse einer Phase, etwa das gebaute Programm, das in der Deploy-Phase ausgerollt wird. So wird genau das ausgeliefert, was zuvor getestet wurde. *Caching* beschleunigt die Läufe, indem zum Beispiel heruntergeladene Abhängigkeiten zwischengespeichert und beim nächsten Durchlauf wiederverwendet werden.

> **Merke**
>
> CI heißt automatisch zusammenführen und testen, CD heißt automatisch bereitstellen oder ausrollen. Die Pipeline verbindet beides: Build → Test → Deploy, und jede Stufe kann den Ablauf stoppen.

GitHub Actions als Einstieg

Ein verbreiteter Einstieg ist **GitHub Actions**, das direkt in die Plattform GitHub integriert ist. Eine Pipeline – dort *Workflow* genannt – wird als YAML-Datei im Verzeichnis `.github/workflows/` des Repositorys beschrieben und damit selbst versioniert. Ausgelöst wird sie durch Ereignisse wie einen Push, einen Pull Request oder einen Zeitplan. Für viele Standardaufgaben gibt es fertige Bausteine, sogenannte *Actions*, etwa zum Auschecken des Codes oder zum Einrichten einer Programmiersprache.

```
name: Test
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - run: ./tests/run.sh
```

Weitere Werkzeuge

GitHub Actions ist nicht die einzige Lösung. **GitLab CI/CD** beschreibt Pipelines in der Datei `.gitlab-ci.yml` und ist besonders beliebt, wenn GitLab im eigenen Rechenzentrum betrieben wird. **Azure DevOps Pipelines** ist häufig in Umgebungen anzutreffen, die stark auf Microsoft-Produkte setzen. **Jenkins** ist ein langjähriger, sehr flexibler Klassiker, der vor allem im klassischen On-Premises-Umfeld verbreitet ist und sich über zahlreiche Plugins erweitern lässt.

So unterschiedlich die Oberflächen und Dateiformate sind – alle Werkzeuge folgen demselben Grundprinzip: Ein Ereignis löst eine Pipeline aus, die in Stufen baut, prüft und ausrollt. Wer das Prinzip an einem Werkzeug verstanden hat, findet sich in den anderen schnell zurecht.

Ein einfaches Deployment

Ein typisches Beispiel aus der Administration: Die Konfiguration oder Anwendung eines Webservers liegt in einem Git-Repository. Ein Push in den Hauptzweig löst die Pipeline aus. Diese baut das Projekt, führt automatische Tests aus – etwa eine Syntaxprüfung der Konfiguration – und rollt das Ergebnis bei Erfolg auf den Zielserver aus.

Für das Ausrollen braucht die Pipeline Zugangsdaten, zum Beispiel einen SSH-Schlüssel für den Zielserver. Diese gehören **niemals ins Repository**, sondern werden im geschützten Bereich der CI-Plattform hinterlegt, oft als *Secrets* bezeichnet. Dort sind sie verschlüsselt gespeichert, werden der Pipeline nur zur Laufzeit bereitgestellt und erscheinen nicht im Klartext in den Protokollen.

Aus der Praxis

Geben Sie dem Zugang, den die Pipeline zum Ausrollen nutzt, nur die Rechte, die sie wirklich braucht – etwa ein eigenes Deploy-Konto, das nur in das Zielverzeichnis schreiben darf. Wird das Secret kompromittiert, bleibt der Schaden begrenzt.

Qualität, Sicherheit und gute Praxis

Eine Pipeline ist nur so gut wie ihre Prüfungen. Automatische Tests und *Linting* – die Prüfung des Codes auf Stil- und typische Programmierfehler – laufen bei jedem Push und wirken als **Gate**: Schlägt etwas fehl, geht der Code nicht weiter. Ergänzend prüfen Abhängigkeits-Scans, ob verwendete Bibliotheken bekannte Sicherheitslücken enthalten, und statische Sicherheitsanalysen (SAST) durchsuchen den eigenen Code nach riskanten Mustern.

Für das Ausrollen selbst gilt eine goldene Regel: **erst nach Staging in die Produktion** – und immer mit einem Rückweg. Ein *Rollback* stellt bei Problemen schnell die vorherige Version wieder her. Beim *Blue-Green-Deployment* laufen zwei gleichwertige Umgebungen parallel; die neue Version wird auf der inaktiven Seite bereitgestellt, und erst nach erfolgreicher Prüfung wird der Verkehr umgeschaltet. Bei Problemen genügt das Zurückschalten.

Bewährt haben sich zudem kleine, häufige Änderungen statt großer Sprünge, weil sich Fehler dann leichter eingrenzen lassen. Halten Sie Pipelines schnell und ihre Meldungen aussagekräftig – eine Pipeline, die eine halbe Stunde braucht und nur „fehlgeschlagen“ meldet, wird im Alltag gern umgangen.

Das Wichtigste in Kürze

- CI bedeutet laufend zusammenführen und testen, CD bedeutet automatisch bereitstellen oder ausrollen.
- Die Pipeline läuft in Stufen: Build → Test → (Staging) → Deploy.
- GitHub Actions, GitLab CI/CD, Azure DevOps und Jenkins folgen demselben Prinzip.
- Tests und Scans sind ein Gate; Secrets liegen nur im Tresor der CI-Plattform.
- Erst Staging, dann Produktion – und immer mit Rollback.

LB-13

Ausbildung & Soft Skills

1. Kundenkommunikation
2. Projektarbeit & Zeitmanagement
3. Teamarbeit & Konfliktmanagement
4. Präsentation & Fachgespräch
5. Lernstrategien & Prüfungsvorbereitung
6. Ausbildungsrecht & Berufseinstieg

Kundenkommunikation

Übersetzer zwischen Technik und Mensch

Nach diesem Kapitel können Sie …

- erklären, warum gute Kommunikation für die Zufriedenheit genauso wichtig ist wie die technische Lösung,
- Supportgespräche kundengerecht führen und Fachbegriffe verständlich übersetzen,
- aktives Zuhören und Empathie gezielt einsetzen,
- realistische Zeitangaben machen und Tickets vollständig dokumentieren.

Technik allein reicht nicht

Wer eine Ausbildung in der IT beginnt, denkt zunächst an Server, Netzwerke und Programmcode. Im Berufsalltag stellt sich aber schnell heraus, dass ein großer Teil der Arbeit aus Gesprächen besteht: mit Kolleginnen und Kollegen aus anderen Abteilungen, mit Kunden am Telefon, mit Vorgesetzten, die eine Entscheidung treffen müssen. Fachkräfte in der IT sind damit immer auch **Übersetzer zwischen Technik und Mensch**.

Wie wichtig diese Rolle ist, zeigt eine einfache Beobachtung aus dem Support: Ein technisch perfekt gelöster Fehler kann einen unzufriedenen Kunden hinterlassen, wenn die Kommunikation schlecht war – etwa weil sich die Person nicht ernst genommen, überfordert oder im Unklaren gelassen fühlte. Umgekehrt führt eine Lösung, die technisch nicht ideal ist, bei exzellenter Kommunikation oft zu mehr Zufriedenheit. Kundinnen und Kunden bewerten nicht nur das Ergebnis, sondern vor allem, wie sie den Weg dorthin erlebt haben.

Ein Beispiel aus dem Support

Eine Mitarbeiterin aus der Buchhaltung ruft an: Sie kann nicht drucken. Die Ursache ist schnell gefunden – der Druckwarteschlangendienst unter Windows hängt. Eine mögliche Antwort lautet:

„Ihr Spool-Service hat sich aufgehängt, führen Sie `net stop spooler` aus, dann `net start spooler`.“

Technisch ist das korrekt, kommunikativ aber ein Fehlschlag. Die Anruferin weiß nicht, was ein Spool-Service ist, und sie weiß auch nicht, wie man einen Dienst stoppt oder wo man solche Befehle eingibt. Sie fühlt sich überfordert und womöglich sogar dumm. Ihr

eigentliches Problem – sie möchte drucken – ist dadurch kein Stück gelöst, und zusätzlich hat sie nun das Gefühl, etwas falsch gemacht zu haben.

Eine bessere Antwort lautet:

„Ich schaue mir das sofort an. Ich sehe das Problem auf Ihrer Seite und löse es kurz per Fernwartung – das dauert etwa zwei Minuten. Sie müssen nichts tun, ich gebe Ihnen Bescheid, sobald alles wieder läuft.“

Diese Antwort enthält mehrere wichtige Elemente. Das Problem wird **übernommen**: Die Anruferin muss sich nicht mehr darum kümmern. Es fallen **keine Fachbegriffe**, die sie nicht versteht. Es gibt eine **realistische Zeitangabe**, sodass sie weiß, woran sie ist. Und sie wird informiert, wenn die Arbeit abgeschlossen ist. Die technische Lösung ist in beiden Fällen dieselbe – das Erlebnis für die Kundin ist völlig verschieden.

> **Merke**
>
> Die Kundin oder der Kunde möchte in der Regel nicht wissen, was technisch kaputt ist, sondern wann das Problem gelöst ist und ob er oder sie selbst etwas tun muss.

Aktives Zuhören und Empathie

Wer viel Erfahrung im Support hat, erkennt typische Probleme oft schon nach dem ersten Satz. Genau darin liegt eine Gefahr: Man denkt bereits an die Lösung und hört nicht mehr richtig zu. Dabei stecken die entscheidenden Hinweise häufig im zweiten oder dritten Satz – etwa dass das Problem erst seit dem letzten Update auftritt oder nur bei einem bestimmten Programm. **Aktives Zuhören** bedeutet, das Gegenüber vollständig ausreden zu lassen, gezielt nachzufragen und das Verstandene kurz mit eigenen Worten zusammenzufassen: „Habe ich richtig verstanden, dass …?“ So stellen Sie sicher, dass Sie das richtige Problem lösen, und Ihr Gegenüber merkt, dass es ernst genommen wird.

Ebenso wichtig ist **Empathie**. Menschen, die beim IT-Support anrufen, stehen oft unter Druck: Eine Frist läuft ab, eine Präsentation beginnt gleich, die Arbeit stockt. Frust oder Ungeduld richten sich dann manchmal an die Person am Telefon, obwohl sie gar nicht gemeint ist. Eine hilfreiche Grundhaltung lautet: *Der Kunde hat ein Problem – er ist nicht das Problem.* Wer das verinnerlicht, bleibt auch in schwierigen Gesprächen ruhig und lösungsorientiert.

Technisch übersetzen

Fachsprache ist unter Fachleuten effizient, für Laien aber eine Hürde. Gute Kommunikation übersetzt technische Sachverhalte in Alltagssprache, ohne dabei herablassend zu wirken. Statt „Out of Memory Exception“ sagen Sie „Der Arbeitsspeicher ist voll“. Statt

„Der DNS-Server löst nicht auf“ heißt es „Der Rechner findet die Adresse der Webseite gerade nicht“. Hilfreich sind auch Vergleiche aus dem Alltag, solange sie den Kern treffen.

Fachjargon	Kundengerecht
„Out of Memory Exception“	„Der Arbeitsspeicher ist voll.“
„Der Spooler hängt.“	„Die Druckerwarteschlange ist blockiert.“
„Ihr Profil ist korrupt.“	„Ihre persönlichen Einstellungen sind beschädigt, ich richte sie neu ein.“

Das Ziel ist nicht, Ihr Gegenüber zur IT-Fachkraft auszubilden, sondern ihm so viel Verständnis zu geben, wie es für seine Situation braucht. Achten Sie dabei auf Rückmeldungen: Fragt jemand nach oder wird still, war die Erklärung vermutlich noch zu technisch.

Realistische Zeitangaben

Zeitangaben gehören zu den heikelsten Aussagen im Kundengespräch. Wer „fünf Minuten“ verspricht und nach einer Stunde noch nicht fertig ist, verliert Vertrauen – selbst wenn die Lösung am Ende gut ist. Eine einfache Regel hilft: **Lieber 30 Minuten ansagen und in 15 fertig sein als umgekehrt.** Eine vorsichtige Schätzung, die unterboten wird, wirkt professionell; eine zu optimistische, die überschritten wird, wirkt unzuverlässig.

Zeichnet sich ab, dass es länger dauert, melden Sie sich von sich aus, bevor die genannte Zeit abläuft. Ein kurzes „Es dauert leider etwas länger, ich rechne mit einer weiteren halben Stunde“ ist deutlich besser als Schweigen.

Aus der Praxis

Ein Satz, den Sie im Kundengespräch nie sagen sollten: „Das ist eigentlich ganz einfach.“ Für die Person am anderen Ende ist es nicht einfach – sonst hätte sie nicht angerufen.

Tickets vollständig dokumentieren

Zur Kommunikation gehört auch die schriftliche Seite. Jede Supportanfrage sollte im Ticketsystem so dokumentiert werden, dass eine andere Person sie ohne Rückfrage nachvollziehen kann: Was wurde *gemeldet*? Was wurde *geprüft*? Wie wurde das Problem *gelöst*? Und wie *lange* hat es gedauert? Eine solche Dokumentation hilft, wenn derselbe Fehler erneut auftritt, sie macht Arbeitsaufwand gegenüber Vorgesetzten und Kunden sichtbar und ist oft die Grundlage für die Abrechnung. Für Auszubildende ist sie zudem eine gute Übung für den Ausbildungsnachweis und die Projektdokumentation.

Das Wichtigste in Kürze

- Wie eine Lösung kommuniziert wird, prägt die Zufriedenheit oft stärker als die Technik selbst.
- Probleme übernehmen, ohne Fachbegriffe erklären und klar sagen, ob der Kunde etwas tun muss.
- Vollständig zuhören und daran denken: Der Kunde hat ein Problem, er ist nicht das Problem.
- Zeitangaben vorsichtig schätzen und Tickets vollständig dokumentieren.

Projektarbeit & Zeitmanagement

Das IHK-Abschlussprojekt planen, dokumentieren und im Zeitrahmen abschließen

Nach diesem Kapitel können Sie …

- Umfang und Bedeutung des betrieblichen Abschlussprojekts einschätzen,
- die drei häufigsten Planungsfehler erkennen und vermeiden,
- ein Projekt in die klassischen Phasen nach dem Wasserfallmodell gliedern,
- Aufgaben mit der Eisenhower-Matrix priorisieren und den Ausbildungsnachweis sinnvoll führen.

Das betriebliche Abschlussprojekt

Für angehende Fachinformatikerinnen und Fachinformatiker ist das betriebliche Abschlussprojekt der wichtigste Einzelbaustein der Abschlussprüfung. Zusammen mit Präsentation und Fachgespräch macht die Projektarbeit nach der aktuellen Prüfungsordnung **50 Prozent der Gesamtnote** aus. Entsprechend umfangreich sind die Anforderungen: Das Projekt wird im Ausbildungsbetrieb geplant und umgesetzt, vollständig dokumentiert, im Betrieb abgenommen und anschließend vor dem Prüfungsausschuss der IHK präsentiert.

Für die Durchführung steht eine fest vorgegebene Stundenzahl zur Verfügung, die vom Ausbildungsberuf und der Fachrichtung abhängt. Nach der seit 2020 geltenden Ausbildungsordnung sind es zum Beispiel in der Fachrichtung Systemintegration 40 Stunden, in der Anwendungsentwicklung 80 Stunden. Maßgeblich sind immer die Vorgaben Ihrer zuständigen IHK. Wer ohne ein Grundverständnis von Projektmanagement in diese Zeit startet, gerät erfahrungsgemäß schnell in Zeitnot.

Fehler 1: Ein zu breites Thema

Einige Dutzend Stunden klingen zunächst nach viel Zeit. Tatsächlich sind sie schnell verbraucht, wenn das Thema schlecht abgegrenzt ist. „Einführung eines neuen Monitoringsystems für das gesamte Unternehmen“ ist kein Abschlussprojekt, sondern ein Vorhaben für Monate. Ein gutes Projektthema beschreibt eine klar umrissene Aufgabe mit einem überprüfbaren Ergebnis – etwa die Auswahl, Einrichtung und Übergabe einer Monitoringlösung für einen bestimmten Serverbereich.

Entscheidend ist, dass das Projekt in der vorgegebenen Zeit **tatsächlich** abgeschlossen werden kann – nicht nur theoretisch, wenn alles reibungslos läuft. Stimmen Sie Thema und Abgrenzung frühzeitig mit Ihrer Ausbilderin oder Ihrem Ausbilder ab, und formulieren Sie im Projektantrag ausdrücklich, was nicht zum Projekt gehört.

Fehler 2 und 3: Dokumentation am Ende, kein Puffer

Ein zweiter klassischer Fehler ist, die **Dokumentation erst am Ende** zu schreiben. Das ist zeitlich kaum zu schaffen und führt fast immer zu einem schlechteren Ergebnis: Viele Details, Zwischenschritte und Begründungen sind nach einigen Tagen bereits vergessen. Wer dagegen parallel dokumentiert – also nach jedem Arbeitsschritt kurz festhält, was getan wurde und warum –, hat am Ende nur noch zu überarbeiten statt neu zu schreiben.

Der dritte Fehler ist ein Zeitplan **ohne Pufferzeit**. Irgendetwas dauert immer länger als gedacht: Eine Lieferung verzögert sich, ein Test schlägt fehl, eine Ansprechperson ist krank. Ohne Puffer gerät dann der gesamte Zeitplan ins Rutschen, und am Ende fehlt die Zeit für Test oder Dokumentation. Planen Sie deshalb bewusst Reserven ein und behandeln Sie sie nicht als frei verfügbare Zeit.

> **Merke**
>
> Ein gutes Abschlussprojekt ist klar abgegrenzt, wird während der Arbeit dokumentiert und hat einen Zeitplan mit echtem Puffer.

Projektphasen nach dem Wasserfallmodell

Für die Gliederung eines Abschlussprojekts eignet sich das klassische **Wasserfallmodell**, bei dem die Phasen nacheinander durchlaufen werden. Es bildet zugleich eine bewährte Struktur für die Projektdokumentation.

In der *Analyse* wird der Ist-Zustand erfasst und beschrieben, welches Problem gelöst werden soll und welche Anforderungen bestehen. Die *Planung* legt fest, wie die Lösung aussehen soll, welche Alternativen es gibt und wie viel Zeit die einzelnen Schritte benötigen. Zur Planung gehört ausdrücklich auch eine **Wirtschaftlichkeitsbetrachtung**: Was kostet das Projekt an Material und Arbeitszeit, und welcher Nutzen steht dem gegenüber? Das klingt nach Betriebswirtschaft, ist aber fester Prüfungsbestandteil.

In der *Durchführung* wird die Lösung umgesetzt. Der *Test* prüft, ob sie die Anforderungen erfüllt – idealerweise anhand vorher festgelegter Kriterien. Abschließend folgen *Übergabe und Dokumentation*: Die Lösung wird an die Nutzenden oder den Auftraggeber übergeben und abgenommen, die Dokumentation wird fertiggestellt.

Phase	Leitfrage
Analyse	Was ist das Problem, was wird gebraucht?
Planung	Wie wird es gelöst, was kostet es, was bringt es?
Durchführung	Umsetzung der geplanten Lösung
Test	Erfüllt die Lösung die Anforderungen?
Übergabe & Dokumentation	Abnahme, Einweisung, vollständige Dokumentation

Prioritäten setzen mit der Eisenhower-Matrix

Zeitmanagement betrifft nicht nur das Projekt, sondern den gesamten Ausbildungsalltag. Ein einfaches Werkzeug zur Priorisierung ist die **Eisenhower-Matrix**. Sie ordnet Aufgaben nach zwei Fragen: Ist die Aufgabe *wichtig*? Und ist sie *dringend*? Daraus ergeben sich vier Felder mit jeweils eigener Handlungsempfehlung.

Was *dringend und wichtig* ist, wird sofort erledigt – etwa ein Serverausfall. Was *wichtig, aber nicht dringend* ist, wird terminiert und fest eingeplant. Genau hier liegt die Prüfungsvorbereitung: Sie ist heute selten dringend, entscheidet aber über das Ergebnis. Aufgaben, die *dringend, aber nicht wichtig* sind, werden nach Möglichkeit delegiert. Was *weder dringend noch wichtig* ist, wird weggelassen.

Der Nutzen der Matrix liegt vor allem im zweiten Feld. Wer nur auf Dringendes reagiert, schiebt wichtige, aber nicht drängende Aufgaben immer weiter vor sich her – bis sie plötzlich doch dringend werden.

Der Ausbildungsnachweis

Ein fester Bestandteil der Ausbildung ist der **Ausbildungsnachweis**, oft noch Berichtsheft genannt. Seine Führung ist gesetzlich vorgeschrieben, und ein ordnungsgemäß geführter Nachweis ist Voraussetzung für die Zulassung zur Abschlussprüfung. Je nach Vorgabe des Betriebs und der IHK wird er täglich oder wöchentlich, schriftlich oder elektronisch geführt und regelmäßig von der ausbildenden Person abgezeichnet.

Auch wenn die Vorgabe wöchentlich lautet, lohnt es sich, jeden Tag ein paar Minuten zu investieren. Fünf Minuten am Ende des Arbeitstags sind kaum Aufwand; am Freitagabend die ganze Woche aus dem Gedächtnis nachzuschreiben ist mühsam und meist unvollständig.

Aus der Praxis

Wer täglich schreibt, ärgert sich einmal. Wer es vergisst, ärgert sich jeden Freitag. Ein gut geführter Ausbildungsnachweis ist außerdem eine wertvolle Gedächtnisstütze bei der Prüfungsvorbereitung.

Mehr als Technik

Das Abschlussprojekt zeigt besonders deutlich, dass der Beruf weit über Technik hinausgeht. Angehende Fachkräfte sind darin Technikerin oder Techniker, Kommunikator, Dokumentator und Projektmanager zugleich. Die technischen Fähigkeiten öffnen die Tür in den Beruf; die sogenannten Soft Skills – Planung, Organisation, Kommunikation – sorgen dafür, dass man dort erfolgreich bleibt und sich weiterentwickelt.

Das Wichtigste in Kürze

- Die betriebliche Projektarbeit samt Präsentation und Fachgespräch macht 50 Prozent der Abschlussnote aus.
- Das Thema muss so abgegrenzt sein, dass es in der vorgegebenen Zeit tatsächlich fertig wird.
- Parallel dokumentieren und Puffer einplanen statt alles ans Ende zu schieben.
- Phasen: Analyse, Planung (mit Wirtschaftlichkeitsbetrachtung), Durchführung, Test, Übergabe.
- Wichtiges, aber nicht Dringendes – wie Prüfungsvorbereitung – fest terminieren.

Hinweis: Projektumfang, Stundenzahl und Gewichtung hängen vom Ausbildungsberuf und der jeweils geltenden Prüfungsordnung ab. Verbindlich sind die Angaben Ihrer zuständigen IHK.

Teamarbeit & Konfliktmanagement

Wie Teams gut zusammenarbeiten und Konflikte lösen

Nach diesem Kapitel können Sie …

- Sachkonflikte von persönlichen Konflikten unterscheiden,
- Meinungsverschiedenheiten mit klaren Kriterien und Zeitrahmen auflösen,
- Feedback mit Ich-Botschaften konstruktiv und zeitnah geben,
- Standups und Retrospektiven kurz, konkret und ergebnisorientiert gestalten.

IT-Arbeit ist Teamarbeit

Das Bild der einzelnen IT-Fachkraft, die allein im Serverraum Probleme löst, hat mit dem Berufsalltag wenig zu tun. Tickets werden im Team verteilt und übergeben, Projekte entstehen in Zusammenarbeit mehrerer Fachrichtungen, und bei einem Deployment müssen Entwicklung, Betrieb und oft auch die Fachabteilung Hand in Hand arbeiten. Wie gut ein Team zusammenarbeitet, entscheidet daher wesentlich darüber, wie zuverlässig die IT funktioniert.

Wo Menschen zusammenarbeiten, gibt es unterschiedliche Sichtweisen, Arbeitsstile und Vorlieben. Das ist nicht nur unvermeidlich, sondern auch wertvoll: Verschiedene Perspektiven decken Schwachstellen auf, die einer einzelnen Person entgangen wären. Entscheidend ist, wie ein Team mit diesen Unterschieden umgeht.

Konflikte sind normal

Unterschiedliche Meinungen sind gesund. Sie zeigen, dass beide Seiten mitdenken und sich für das Ergebnis interessieren. Ein Team, in dem nie widersprochen wird, trifft nicht unbedingt bessere Entscheidungen – oft traut sich dort nur niemand, Bedenken zu äußern.

Problematisch wird ein Konflikt erst dann, wenn er von der **Sachebene** auf die **persönliche Ebene** wechselt. Auf der Sachebene geht es um die Frage, welche Lösung die beste ist. Auf der persönlichen Ebene geht es plötzlich darum, wer recht behält, wer sich durchsetzt oder wer sich übergangen fühlt. Dann werden Argumente nicht mehr nach ihrem Inhalt bewertet, sondern danach, von wem sie kommen – und eine sachliche Lösung rückt in weite Ferne.

> **Merke**
>
> Ein Sachkonflikt ist eine Chance auf eine bessere Lösung. Gefährlich wird es, wenn aus „Welche Lösung ist besser?“ die Frage „Wer hat recht?“ wird.

Beispiel: Streit um das richtige Werkzeug

Ein typisches Beispiel aus dem IT-Alltag: Für ein neues Monitoring soll ein Werkzeug eingeführt werden. Zwei Kollegen sind sich uneinig, welches es sein soll. Jeder hat gute Erfahrungen mit „seinem“ Tool gemacht, die Diskussion wird mit jeder Runde lauter, und das Team wartet auf eine Entscheidung.

Der erste Schritt ist, die Ebenen zu **trennen**: Geht es wirklich um das Werkzeug – oder darum, wer am Ende recht behält? Diese Frage offen zu stellen, entschärft die Situation oft schon spürbar, weil sie den Blick zurück auf die Sache lenkt.

Im zweiten Schritt werden **Kriterien vorher festgelegt**, bevor die Werkzeuge verglichen werden. Sinnvolle Kriterien wären hier zum Beispiel die Kosten, der Lernaufwand für das Team und die Integration in die bestehende Umgebung. Anschließend werden beide Kandidaten anhand dieser Kriterien bewertet – nicht anhand von Bauchgefühl oder persönlicher Sympathie für ein Produkt. Oft zeigt sich dabei, dass beide Vorschläge Stärken haben und die Entscheidung weniger eindeutig ist, als die Diskussion vermuten ließ.

Kommt das Team trotzdem nicht weiter, hilft ein **fester Zeitrahmen**: Bis zu einem bestimmten Termin, etwa bis Freitag, wird eine gemeinsame Entscheidung gesucht. Gelingt das nicht, entscheidet die Teamleitung. Diese Regel verhindert, dass ein Streit das Team wochenlang blockiert, und sie sorgt dafür, dass am Ende überhaupt eine Entscheidung steht.

> **Aus der Praxis**
>
> Schreiben Sie die Entscheidungskriterien und ihre Gewichtung sichtbar auf, bevor die Diskussion über konkrete Produkte beginnt. Wer die Kriterien mitbestimmt hat, kann eine Entscheidung auf dieser Grundlage deutlich leichter mittragen – auch wenn sein Favorit nicht gewinnt.

Feedback richtig geben

Viele Konflikte im Team entstehen nicht durch große Streitfragen, sondern durch kleine Ärgernisse, die nie angesprochen werden – oder auf eine Weise, die das Gegenüber verletzt. Gutes Feedback hilft, solche Situationen früh und ohne Eskalation zu klären.

Die wichtigste Regel lautet: **Ich-Botschaften statt Vorwürfe**. Ein Satz wie „Du machst immer …“ greift die Person an, verallgemeinert und lädt zur Verteidigung ein. Eine Ich-Botschaft beschreibt dagegen die eigene Wahrnehmung und ihre Wirkung: „Mir ist aufgefallen, dass die Tickets in den letzten Tagen ohne Lösungsbeschreibung geschlossen wurden. Dadurch fehlen mir bei Rückfragen die Informationen.“ Die Aussage bleibt konkret, ist nicht verletzend und lässt Raum für die Sicht des anderen.

Ebenso wichtig ist der Zeitpunkt. Feedback sollte **zeitnah** gegeben werden, solange die Situation noch frisch ist und beide Seiten sich erinnern. Wer Kritik wochenlang sammelt und sie dann erst in der nächsten Retrospektive auf den Tisch legt, überrumpelt das Gegenüber und wärmt Dinge auf, die längst hätten geklärt sein können. Am besten geschieht das unter vier Augen und nicht vor dem ganzen Team.

Standups und Retrospektiven

Viele IT-Teams arbeiten mit festen Besprechungsformaten aus der agilen Arbeitswelt. Im **Daily Standup** stimmt sich das Team kurz über den aktuellen Stand ab: Woran wird gearbeitet, was steht als Nächstes an, wo gibt es Hindernisse? Der Name verrät das Ziel – die Runde ist so kurz, dass man sie im Stehen abhalten kann. Ausführliche Diskussionen werden in ein separates Gespräch mit den Beteiligten verlagert.

Die **Retrospektive** blickt in regelmäßigen Abständen auf die Zusammenarbeit zurück. Drei Fragen bilden den Kern: Was lief gut? Was lief schlecht? Und was ändern wir konkret? Gerade die dritte Frage ist entscheidend. Eine Retrospektive, in der nur Frust abgeladen wird, ohne dass daraus Konsequenzen folgen, frustriert auf Dauer mehr, als sie hilft. Vereinbaren Sie deshalb wenige, klare Maßnahmen und prüfen Sie beim nächsten Mal, ob sie umgesetzt wurden. Für beide Formate gilt: kurz, konkret, ergebnisorientiert.

Spannungen früh ansprechen

Nicht jeder Konflikt ist ein Sachkonflikt. Manchmal entstehen echte Spannungen zwischen Menschen, etwa weil sich jemand dauerhaft übergangen fühlt oder die Arbeitsweisen nicht zusammenpassen. Die Versuchung ist groß, solche Situationen auszusitzen und auf Besserung zu hoffen. Die Erfahrung zeigt jedoch: Ausgesessene Konflikte werden fast nie kleiner. Sie verfestigen sich, beeinflussen die Stimmung im ganzen Team und brechen später umso heftiger auf. Sprechen Sie Spannungen deshalb früh, ruhig und unter vier Augen an. Gelingt die Klärung nicht, ist es kein Zeichen von Schwäche, die Teamleitung oder – in der Ausbildung – die Ausbilderin oder den Ausbilder einzubeziehen.

Das Wichtigste in Kürze

- Meinungsverschiedenheiten sind normal; kritisch wird es beim Wechsel auf die persönliche Ebene.
- Sachkonflikte mit vorher festgelegten Kriterien statt Bauchgefühl lösen, bei Stillstand mit Zeitrahmen.
- Feedback als Ich-Botschaft und zeitnah geben, nicht erst Wochen später.
- Standups und Retrospektiven kurz, konkret und mit echten Konsequenzen führen.
- Echte Spannungen früh ansprechen – ausgesessene Konflikte wachsen meist.

Präsentation & Fachgespräch

Das Abschlussprojekt vorstellen und Rückfragen souverän beantworten

Nach diesem Kapitel können Sie …

- den Ablauf von Präsentation und Fachgespräch in der Abschlussprüfung beschreiben,
- eine Präsentation entlang einer klaren Storyline aufbauen,
- Folien übersichtlich und vortragsunterstützend gestalten,
- sich gezielt auf typische Fragen im Fachgespräch vorbereiten und mit Nervosität umgehen.

Der Ablauf

Mit der Abgabe der Projektdokumentation ist das Abschlussprojekt noch nicht ganz abgeschlossen. Im zweiten Teil der IHK-Abschlussprüfung stellen Sie Ihr Projekt dem **Prüfungsausschuss** vor und beantworten anschließend dessen Fragen. Dieser Prüfungsteil fließt zusammen mit Projekt und Dokumentation in die Bewertung der Projektarbeit ein und hat damit erhebliches Gewicht für die Abschlussnote.

Der Ablauf ist zweigeteilt. Zunächst halten Sie eine **Präsentation** von rund 15 Minuten, in der Sie Ihr Projekt vorstellen. Daran schließt sich das **Fachgespräch** an, in dem die Prüferinnen und Prüfer Rückfragen stellen – zum Projekt selbst, zu Ihren Entscheidungen und auch zu Alternativen, die Sie nicht gewählt haben. Für Fachinformatikerinnen und Fachinformatiker dauern beide Teile zusammen etwa eine halbe Stunde; die genauen Zeiten legt die jeweilige Prüfungsordnung fest. Halten Sie die vorgegebene Präsentationszeit ein: Ein deutliches Über- oder Unterschreiten fällt auf und kann sich in der Bewertung niederschlagen.

Die Storyline der Präsentation

Eine gute Präsentation erzählt eine Geschichte mit einem klaren roten Faden. Für ein Abschlussprojekt hat sich eine einfache Dreiteilung bewährt, die sich an den Fragen orientiert, die sich jede Zuhörerin und jeder Zuhörer stellt.

Am Anfang steht die **Ausgangslage**: Was war das Problem? Beschreiben Sie kurz das Unternehmen bzw. die Abteilung, den Ist-Zustand und warum Handlungsbedarf bestand. Die Prüfenden kennen Ihren Betrieb nicht, deshalb braucht es hier genug Kontext, um den Rest zu verstehen – aber nicht mehr.

Im Mittelteil folgt die **Lösung**: Was wurde umgesetzt, und warum genau so? Dieser Teil ist das Herzstück. Zeigen Sie nicht nur, was Sie getan haben, sondern begründen Sie Ihre Entscheidungen. Welche Alternativen gab es, nach welchen Kriterien haben Sie ausgewählt, und welche Rolle spielte die Wirtschaftlichkeit?

Den Abschluss bildet das **Ergebnis**: Hat es funktioniert? Wurden die Ziele erreicht, und wie wurde das überprüft? Ein ehrlicher Rückblick gehört dazu – was würden Sie im Nachhinein anders machen? Ein solches Fazit zeigt Reflexionsvermögen und wirkt deutlich überzeugender als die Behauptung, alles sei perfekt gelaufen.

> **Merke**
>
> Ausgangslage, Lösung, Ergebnis: Diese drei Teile tragen jede Projektpräsentation. Das Gewicht liegt auf der Lösung und vor allem auf der Begründung Ihrer Entscheidungen.

Folien gestalten

Ein häufiger Fehler sind Folien, die mit Text überladen sind. Eine solche „Textwüste“ führt dazu, dass das Publikum liest statt zuhört – und meist schneller liest, als Sie sprechen. Halten Sie sich deshalb an eine einfache Regel: **wenige Stichpunkte pro Folie**, kurze Formulierungen und lieber eine Folie mehr als eine überfüllte.

Wo es passt, sagen Diagramme, Netzwerkpläne oder Screenshots mehr als Text. Ein übersichtlicher Plan der neuen Infrastruktur oder eine einfache Grafik zum Projektablauf lässt sich schneller erfassen als jeder Absatz. Achten Sie auf ausreichend große Schrift, guten Kontrast und ein einheitliches Erscheinungsbild.

Vor allem aber: Der Vortrag lebt von Ihnen, nicht vom Text auf der Leinwand. Die Folien unterstützen, was Sie sagen; sie ersetzen es nicht. Wer seine Folien nur vorliest, verschenkt den wichtigsten Teil der Präsentation – den Eindruck, das Thema wirklich zu beherrschen.

Das Fachgespräch

Vor dem Fachgespräch haben viele Prüflinge den größten Respekt. Dabei hilft ein Perspektivwechsel: Die Prüferinnen und Prüfer wollen Ihre Lösung nicht zerreißen. Sie wollen verstehen, ob die Lösung wirklich durchdacht ist und ob Sie die Zusammenhänge hinter Ihrem Projekt verstanden haben. Das Gespräch ist eine Gelegenheit, genau das zu zeigen.

Eine typische Frage lautet: *„Warum haben Sie sich für X entschieden und nicht für Y?“* Hier überzeugt eine **ehrliche Abwägung** weit mehr als eine auswendig gelernte Antwort, die bei der ersten Nachfrage in sich zusammenfällt. Nennen Sie die Kriterien,

nach denen Sie entschieden haben, und benennen Sie offen, welche Nachteile Ihre Lösung hat. Zum Fachgespräch gehören außerdem Fragen zu den technischen Grundlagen Ihres Projekts – wer etwa einen VPN-Zugang eingerichtet hat, sollte auch erklären können, wie die Verschlüsselung grundsätzlich funktioniert.

Wenn Sie eine Antwort nicht wissen, sagen Sie es ehrlich und versuchen Sie, laut nachzudenken: „Das habe ich im Projekt nicht untersucht, ich würde aber vermuten, dass …“ Das ist deutlich besser als Raten oder Ausweichen.

Aus der Praxis

Gehen Sie Ihre Dokumentation Abschnitt für Abschnitt durch und notieren Sie zu jeder Entscheidung die Frage „Warum so und nicht anders?“. Wer diese Fragen vorab beantwortet hat, geht deutlich ruhiger ins Fachgespräch.

Üben und mit Nervosität umgehen

Die wirksamste Vorbereitung ist einfach: **laut üben**. Halten Sie die Präsentation mehrmals vollständig, am besten vor Publikum – vor Kolleginnen und Kollegen, der Familie oder notfalls vor dem Spiegel. Wer den eigenen Vortrag noch nie laut gesprochen hat, merkt erst live, an welchen Stellen die Formulierungen holpern, welche Übergänge fehlen und wie lange die Präsentation tatsächlich dauert. Lassen Sie sich beim Üben ruhig auch Fragen stellen, um das Fachgespräch zu simulieren.

Nervosität vor einer Prüfung ist völlig normal, und die Prüfenden wissen das. Kommt sie während des Vortrags auf, hilft ein einfaches Vorgehen: kurz innehalten, tief durchatmen und dann weitersprechen. Eine Pause von wenigen Sekunden wirkt auf das Publikum viel kürzer als auf Sie selbst, und den Faden nach einer kurzen Unterbrechung wieder aufzunehmen, ist ausdrücklich erlaubt. Eine kleine Karteikarte mit den wichtigsten Stichpunkten gibt zusätzlich Sicherheit.

Planen Sie auch das Drumherum: Klären Sie vorab, welche Technik vor Ort zur Verfügung steht, halten Sie die Präsentation in einem gängigen Format und auf einem zweiten Speichermedium bereit, und seien Sie am Prüfungstag rechtzeitig vor Ort.

Das Wichtigste in Kürze

- Auf eine Präsentation von rund 15 Minuten folgt das Fachgespräch mit dem Prüfungsausschuss.
- Storyline: Ausgangslage – Lösung mit Begründung – Ergebnis mit ehrlichem Rückblick.
- Wenige Stichpunkte pro Folie; der Vortrag lebt von Ihnen, nicht vom Text.

- Im Fachgespräch überzeugt eine ehrliche Abwägung mehr als eine auswendig gelernte Antwort.
- Laut üben und bei Nervosität kurz durchatmen – weiterreden ist erlaubt.

Lernstrategien & Prüfungsvorbereitung

Für Teil 1 und Teil 2 der Abschlussprüfung – was wirklich hilft

Nach diesem Kapitel können Sie …

- erklären, warum das Üben alter Prüfungen der wirksamste Hebel ist,
- Spaced Repetition für das langfristige Behalten von Wissen einsetzen,
- einen realistischen Lernplan rückwärts vom Prüfungstermin aufstellen,
- Einzel- und Gruppenlernen sinnvoll kombinieren.

Zwei Prüfungen, ein Ziel

Die Abschlussprüfung in den IT-Berufen ist als **gestreckte Abschlussprüfung** aufgebaut. Teil 1 findet etwa zur Mitte der Ausbildung statt und ersetzt die frühere Zwischenprüfung – mit dem wichtigen Unterschied, dass das Ergebnis bereits in die Abschlussnote einfließt. Teil 2 folgt am Ende der Ausbildung und umfasst neben dem betrieblichen Projekt auch schriftliche Prüfungsbereiche. Beide Teile verlangen also eine gezielte Vorbereitung.

Viele Auszubildende investieren viel Zeit ins Lernen und sind trotzdem enttäuscht vom Ergebnis. Das liegt selten an fehlendem Fleiß, sondern meist an der Methode. Stundenlanges Lesen von Lehrbüchern und Mitschriften fühlt sich produktiv an, bringt aber vergleichsweise wenig. Die folgenden Strategien sind deutlich wirksamer.

Der größte Hebel: alte Prüfungen üben

Die schriftlichen IHK-Prüfungen werden bundesweit einheitlich erstellt und ähneln sich in Struktur und Aufgabentypen von Jahr zu Jahr stark. Bestimmte Themen und Fragestellungen tauchen in leicht veränderter Form immer wieder auf: Subnetting, Kostenberechnungen, Datenbankabfragen, Datenschutzfragen oder das Lesen von Programmcode und Diagrammen.

Daraus folgt die wichtigste Empfehlung dieses Kapitels: **Rechnen Sie alte Prüfungen durch**, statt nur Theorie zu lesen. Wer zehn alte Prüfungen vollständig bearbeitet hat, kennt die typischen Fallen, bevor sie ihm in der echten Prüfung begegnen. Er weiß, wie Aufgaben formuliert sind, wie viel Zeit für welchen Aufgabentyp nötig ist und wo besonders genaues Lesen gefragt ist. Alte Prüfungssätze erhalten Sie in der Regel über die IHK, den Ausbildungsbetrieb oder die Berufsschule.

Wirksam ist das allerdings nur, wenn Sie die Aufgaben wirklich selbst lösen – möglichst unter Prüfungsbedingungen mit Zeitlimit – und erst danach die Lösung ansehen. Wer gleich zur Musterlösung blättert, hat das Gefühl, alles verstanden zu haben, kann es aber in der Prüfung nicht abrufen.

> **Merke**
>
> Alte Prüfungen selbst durchzurechnen schlägt das Lesen von Theorie. Erst bearbeiten, dann die Lösung vergleichen – und die Fehler gezielt nacharbeiten.

Spaced Repetition: verteilt wiederholen

Für Wissen, das man schlicht auswendig können muss – Portnummern, Fachbegriffe, Fristen, Abkürzungen –, eignet sich **Spaced Repetition**, also das verteilte Wiederholen. Dabei werden Lerninhalte auf Karteikarten festgehalten und in wachsenden Abständen wiederholt: zunächst am nächsten Tag, dann nach einigen Tagen, dann nach einer Woche und so weiter. Karten, die sicher sitzen, kommen seltener an die Reihe, schwierige Karten häufiger. Das funktioniert mit Papierkarten in einem Karteikasten ebenso wie mit einer Lernkarten-App, die die Abstände automatisch berechnet.

Hinter der Methode steht eine gut belegte Erkenntnis der Lernforschung: Das Gehirn merkt sich Dinge besser, die es mehrmals über Tage verteilt aktiv abrufen muss. Inhalte, die einmal am Stück durchgelesen wurden, bleiben dagegen kaum hängen. Entscheidend ist der aktive Abruf – also die Antwort selbst zu formulieren, bevor man die Karte umdreht.

Ein realistischer Lernplan

Ein guter Lernplan wird **rückwärts vom Prüfungstermin** geplant. Ausgehend vom Datum der Prüfung legen Sie fest, bis wann welche Themen bearbeitet sein sollen und wann Probeprüfungen stattfinden. Planen Sie dabei einen echten **Puffer** ein – für Krankheit, einen stressigen Arbeitsabschnitt im Betrieb oder ein Thema, das länger dauert als erwartet. Die letzte Woche vor der Prüfung sollte für Wiederholung reserviert sein, nicht für neuen Stoff.

Priorisieren Sie die Themen nach ihrem **Prüfungsgewicht**. Nicht jedes Thema zählt gleich viel, und nicht jedes ist für Sie gleich schwierig. Wer viel Zeit in ein Randthema steckt, das er ohnehin beherrscht, dem fehlt diese Zeit bei den großen Brocken. Ein Blick in alte Prüfungen zeigt, welche Themen regelmäßig vorkommen und viele Punkte bringen.

Für die einzelnen Lerneinheiten gilt: **Kurze, fokussierte Blöcke** von 25 bis 30 Minuten mit einer kurzen Pause dazwischen sind wirksamer als stundenlanges, unkonzentriertes Lesen. Legen Sie das Smartphone während eines Blocks außer Reichweite, und nehmen

Sie sich für jeden Block ein konkretes Ziel vor, etwa „zwei Subnetting-Aufgaben lösen“ statt „Netzwerke lernen“.

Aus der Praxis

Tragen Sie Lernblöcke wie Termine fest in Ihren Kalender ein. Prüfungsvorbereitung ist wichtig, aber selten dringend – ohne festen Termin wird sie leicht von Alltagsaufgaben verdrängt.

Allein oder in der Gruppe?

Beides hat seinen Platz, und am wirksamsten ist die Kombination. **Allein** lassen sich Grundlagen in Ruhe erarbeiten: Stoff durchgehen, Aufgaben lösen, Karteikarten wiederholen – jeweils im eigenen Tempo und mit Fokus auf die eigenen Schwächen.

Die **Gruppe** hat eine andere Stärke: Sie deckt Verständnislücken auf. Wer einen Sachverhalt jemand anderem erklären kann, hat ihn wirklich verstanden. Wer ihn dagegen nur wiedererkennt, wenn er die Antwort liest, hat ihn noch nicht verinnerlicht – und merkt das oft erst, wenn er ihn in eigenen Worten erklären soll. In einer Lerngruppe können Sie sich gegenseitig abfragen, schwierige Aufgaben gemeinsam lösen und unterschiedliche Lösungswege vergleichen. Damit die Gruppe produktiv bleibt, hilft eine klare Verabredung, welche Themen beim nächsten Treffen bearbeitet werden.

Lernform	Stärke	Typischer Einsatz
Allein	Eigenes Tempo, gezielt an Schwächen arbeiten	Grundlagen erarbeiten, Karteikarten, Probeprüfungen
In der Gruppe	Verständnislücken werden sichtbar	Erklären, gegenseitig abfragen, Lösungswege vergleichen

Kurz vor der Prüfung

In den letzten Tagen vor der Prüfung bringt neuer Stoff kaum noch etwas. Wiederholen Sie stattdessen Ihre Karteikarten, sehen Sie sich Ihre typischen Fehler aus den Probeprüfungen noch einmal an und sorgen Sie für ausreichend Schlaf – im Schlaf festigt das Gehirn Gelerntes. Klären Sie rechtzeitig, welche Hilfsmittel zugelassen sind, etwa ein Taschenrechner, und wann und wo die Prüfung stattfindet. In der Prüfung selbst lohnt es sich, zunächst alle Aufgaben zu überfliegen und mit denen zu beginnen, die Ihnen sicher liegen.

Das Wichtigste in Kürze

- Alte Prüfungen selbst durchrechnen ist der wirksamste Hebel der Vorbereitung.
- Spaced Repetition: in wachsenden Abständen aktiv abrufen statt einmal pauken.
- Rückwärts vom Prüfungstermin planen, mit Puffer und nach Prüfungsgewicht priorisiert.
- Kurze Lernblöcke von 25 bis 30 Minuten schlagen stundenlanges Lesen.
- Allein Wissen aufbauen, in der Gruppe Lücken aufdecken – wer erklären kann, hat verstanden.

Ausbildungsrecht & Berufseinstieg

Das juristische Grundgerüst rund um die Ausbildung – kurz und praxisnah

Nach diesem Kapitel können Sie …

- die Regeln zur Probezeit nach dem Berufsbildungsgesetz erklären,
- unterscheiden, wann und wie ein Ausbildungsverhältnis gekündigt werden kann,
- einordnen, wie Berufsschulzeit angerechnet wird und was der Jugendarbeitsschutz regelt,
- Ihre Situation rund um Übernahme und Ausbildungszeugnis realistisch einschätzen.

Die rechtliche Grundlage

Eine duale Ausbildung ist mehr als ein gewöhnliches Arbeitsverhältnis. Der Betrieb verpflichtet sich, Ihnen die Inhalte Ihres Ausbildungsberufs zu vermitteln, und Sie verpflichten sich, diese Inhalte zu lernen. Die wichtigsten Regeln dafür stehen im **Berufsbildungsgesetz** (BBiG). Es legt unter anderem fest, wie lange die Probezeit dauert, unter welchen Bedingungen gekündigt werden kann, wie die Berufsschule angerechnet wird und welches Zeugnis Ihnen am Ende zusteht. Für Auszubildende unter 18 Jahren gilt zusätzlich das **Jugendarbeitsschutzgesetz** (JArbSchG).

Ergänzend können ein *Tarifvertrag* oder eine *Betriebsvereinbarung* weitere Regeln enthalten, die für Sie günstiger sind als das Gesetz. Es lohnt sich daher, nicht nur den eigenen Ausbildungsvertrag genau zu lesen, sondern auch nachzufragen, ob im Betrieb solche Vereinbarungen gelten. Dieses Kapitel gibt einen Überblick über die Punkte, die im Ausbildungsalltag am häufigsten eine Rolle spielen. Es ersetzt keine Rechtsberatung – bei konkreten Problemen sind die zuständige Kammer, die Berufsschule oder eine Gewerkschaft gute Anlaufstellen.

Die Probezeit

Jedes Ausbildungsverhältnis beginnt mit einer **Probezeit**. Laut Berufsbildungsgesetz dauert sie **mindestens einen und höchstens vier Monate**. Die genaue Länge steht im Ausbildungsvertrag. Die Probezeit dient beiden Seiten dazu, sich kennenzulernen: Der Betrieb kann prüfen, ob Sie für den Beruf geeignet sind, und Sie können herausfinden, ob Beruf und Betrieb wirklich zu Ihnen passen.

Deshalb ist die Kündigung in dieser Phase besonders einfach. **Beide Seiten** können das Ausbildungsverhältnis während der Probezeit **jederzeit ohne Kündigungsfrist und ohne Angabe von Gründen** beenden. Wichtig ist nur, dass die Kündigung schriftlich

erfolgt – eine mündliche Kündigung oder eine Nachricht per Messenger reicht nicht aus.

> **Merke**
>
> Die Probezeit dauert ein bis vier Monate. In dieser Zeit können Betrieb und Auszubildende jederzeit fristlos und ohne Begründung kündigen – allerdings immer schriftlich.

Kündigung nach der Probezeit

Ist die Probezeit vorbei, ändert sich die Lage deutlich. Das Gesetz schützt nun das Ausbildungsverhältnis, damit die Ausbildung möglichst ungestört zu Ende geführt werden kann. Für Sie als Auszubildende oder Auszubildender gilt: Sie können nur noch mit einer **Frist von vier Wochen** kündigen, und zwar nur dann, wenn Sie die **Ausbildung ganz aufgeben** oder sich für einen **anderen Beruf** ausbilden lassen wollen. Den Betrieb zu wechseln, um denselben Beruf woanders zu lernen, ist auf diesem Weg nicht vorgesehen – dafür braucht es in der Regel eine einvernehmliche Lösung mit dem bisherigen Betrieb.

Der **Ausbildungsbetrieb** kann nach der Probezeit nur noch **aus einem wichtigen Grund fristlos** kündigen. Ein solcher Grund liegt nur bei schwerwiegenden Pflichtverletzungen vor, etwa bei Diebstahl oder wiederholtem unentschuldigtem Fehlen trotz Abmahnung. Eine ordentliche Kündigung, wie sie bei normalen Arbeitsverhältnissen möglich ist, gibt es für den Betrieb nicht. Auch nach der Probezeit muss jede Kündigung schriftlich erfolgen und die Gründe nennen.

Zeitpunkt	Auszubildende	Ausbildungsbetrieb
In der Probezeit	Jederzeit, ohne Frist und Begründung	Jederzeit, ohne Frist und Begründung
Nach der Probezeit	Mit vier Wochen Frist, nur bei Aufgabe der Ausbildung oder Berufswechsel	Nur aus wichtigem Grund, fristlos

Berufsschule und Jugendarbeitsschutz

Die Berufsschule ist fester Bestandteil der dualen Ausbildung. Daher gilt: **Berufsschulzeit ist Ausbildungszeit**. Der Betrieb muss Sie für den Unterricht freistellen und die Ausbildungsvergütung für diese Zeit weiterzahlen. Die Berufsschule ist also keine Freizeit und kein Urlaub, sondern Teil Ihrer Arbeitszeit. Umgekehrt sind Sie verpflichtet, am Unterricht teilzunehmen.

Sind Sie noch **unter 18 Jahre** alt, greift zusätzlich das Jugendarbeitsschutzgesetz. Es begrenzt die Arbeitszeit auf **höchstens acht Stunden pro Tag** und schreibt feste **Pausenregelungen** vor, die je nach Länge der Arbeitszeit gestaffelt sind. Außerdem zählt ein **Berufsschultag mit viel Unterricht** – gemeint sind mehr als fünf Unterrichtsstunden – einmal pro Woche bereits als vollständiger Arbeitstag. Nach einem solchen Tag müssen Jugendliche also nicht mehr in den Betrieb zurückkehren. Auch für volljährige Auszubildende wird ein solcher langer Berufsschultag nach dem Berufsbildungsgesetz einmal pro Woche mit der durchschnittlichen täglichen Ausbildungszeit angerechnet.

Aus der Praxis

Führen Sie Ihren Ausbildungsnachweis sorgfältig und tragen Sie die Berufsschultage korrekt ein. So lassen sich Fragen zur Anrechnung von Arbeitszeit später leicht klären, und der Nachweis ist ohnehin Voraussetzung für die Zulassung zur Abschlussprüfung.

Übernahme nach der Ausbildung

Gegen Ende der Ausbildung stellt sich die Frage, wie es weitergeht. Viele Auszubildende gehen davon aus, dass der Betrieb sie automatisch übernimmt. Tatsächlich gibt es aber **normalerweise keinen generellen gesetzlichen Anspruch auf Übernahme**. Eine Ausnahme besteht, wenn ein Tarifvertrag oder eine Betriebsvereinbarung eine Übernahme ausdrücklich vorsieht. Das Ausbildungsverhältnis endet in der Regel mit dem Bestehen der Abschlussprüfung.

Daraus folgt eine klare Empfehlung: Warten Sie nicht einfach ab. Sprechen Sie **einige Monate vor dem Prüfungsende aktiv** mit Ihrem Betrieb über eine mögliche Übernahme. So wissen Sie rechtzeitig, woran Sie sind, und können Ihre nächsten Schritte planen. Parallel lohnt es sich, den Arbeitsmarkt im Blick zu behalten, falls eine Übernahme doch nicht zustande kommt. Gerade in der IT gibt es viele Einstiegsmöglichkeiten, und eine frühe Orientierung verschafft Ihnen Spielraum.

Das Ausbildungszeugnis

Am Ende der Ausbildung steht Ihnen ein **Ausbildungszeugnis** zu. Das *einfache Zeugnis* enthält Angaben zu Art, Dauer und Ziel der Ausbildung sowie zu den erworbenen Fertigkeiten und Kenntnissen. Auf Ihren Wunsch muss der Betrieb ein *qualifiziertes Zeugnis* ausstellen, das zusätzlich **Leistung und Verhalten** bewertet.

Dieses Zeugnis ist die Grundlage für jede spätere Bewerbung. Künftige Arbeitgeber lesen daraus ab, wie Sie gearbeitet haben und wie Sie im Team aufgetreten sind. Fordern Sie deshalb ausdrücklich ein qualifiziertes Zeugnis an und prüfen Sie es sorgfältig, be-

vor Sie den Betrieb verlassen. Unklare oder fehlerhafte Formulierungen lassen sich am einfachsten klären, solange der Kontakt zu den Ausbildungsverantwortlichen noch besteht.

> **Merke**
>
> Ein Anspruch auf Übernahme besteht nur, wenn ein Tarifvertrag oder eine Betriebsvereinbarung ihn vorsieht. Ein qualifiziertes Ausbildungszeugnis erhalten Sie dagegen immer – Sie müssen es nur verlangen.

Das Wichtigste in Kürze

- Die Probezeit dauert ein bis vier Monate; beide Seiten können in dieser Zeit ohne Frist und Begründung kündigen.
- Nach der Probezeit kündigen Auszubildende nur mit vier Wochen Frist bei Ausbildungsaufgabe oder Berufswechsel, der Betrieb nur aus wichtigem Grund.
- Berufsschulzeit ist bezahlte Ausbildungszeit; für unter 18-Jährige gilt zusätzlich das Jugendarbeitsschutzgesetz.
- Ein Übernahmeanspruch besteht meist nicht – rechtzeitig selbst nachfragen.
- Das qualifizierte Ausbildungszeugnis ist die Grundlage für jede weitere Bewerbung.