

Fraud (Betrug) – Täuschung oder Manipulation, um finanzielle oder persönliche Vorteile zu erlangen, z. B. durch Phishing oder Identitätsdiebstahl.

Ransom Demand (Lösegeldforderung) – Angriffe (z. B. Ransomware), bei denen Daten verschlüsselt werden und erst nach Zahlung eines Lösegelds freigegeben werden.

Espionage (Spionage) – Unautorisierte Beschaffung sensibler Informationen durch Cyberangriffe, oft im wirtschaftlichen oder politischen Kontext.

Violation of Regulations (Verstoß gegen Vorschriften) – Missachtung von Datenschutz- oder Sicherheitsrichtlinien, oft durch unzureichende Sicherheitsmaßnahmen oder Datenlecks.

Misuse of Computing Resources (Missbrauch von IT-Ressourcen) – Verwendung von IT-Infrastruktur für unerlaubte Zwecke, z. B. für Krypto-Mining oder Botnets.

Reputation Loss (Reputationsverlust) – Schäden am Image eines Unternehmens durch Sicherheitsvorfälle oder Datenlecks.

System Outage (Systemausfall) – Gezielte Angriffe, die Systeme oder Netzwerke lahmlegen, z. B. durch DDoS-Attacken.

Brand Misuse (Markenmissbrauch) – Betrügerische Verwendung von Markennamen oder Logos, um Nutzer zu täuschen oder Unternehmen zu schädigen.

Sabotage – Gezielte Störung oder Zerstörung von IT-Systemen oder Daten, oft durch interne oder externe Angreifer.

Data Loss (Datenverlust) – Verlust sensibler Daten durch Angriffe, technische Fehler oder unzureichende Sicherheitsmaßnahmen.

Bekannte angriffe:

- **Malware:** Eine bösartige Software wird installiert, welche es dem Angreifer erlaubt Code auf dem Rechner des Opfers auszuführen
- **Ransomware:** Die Daten des Opfers werden verschlüsselt und eine Erpressungsnachricht wird angezeigt
- **Phishing und Social-Engineering:** Da der Mensch in der Regel sehr gutgläubig ist, gelingt es häufig mit Phishing oder anderen Social-Engineering Techniken an Benutzeraccounts oder ähnlich wichtige Informationen zu kommen
- **DDoS: Distributed Denial of Service:** Ein Angreifer benutzt ein Botnet, um sehr viele Anfragen an ein System zu schicken. Das System bricht unter der Last zusammen und ist nicht mehr für den normalen Einsatzzweck erreichbar.

Goals of Information Security

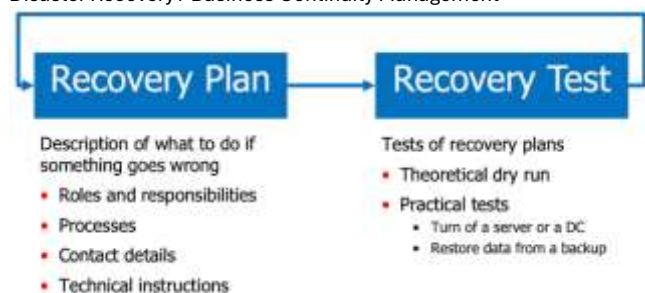


Confidentiality (Vertraulichkeit) – Schutz vor unbefugtem Zugriff (z. B. Verschlüsselung, Zugriffskontrollen).

Integrity (Integrität) – Sicherstellung der Datenkorrektheit (z. B. Hashing, digitale Signaturen).

Availability (Verfügbarkeit) – Gewährleistung der System- und Datenverfügbarkeit (z. B. Backups, Redundanz).

Disaster Recovery / Business Continuity Management



WorkFactor:

- Allgemeine Formel für den Work Factor für einen zufälligen Schlüssel der Länge n bits

$$\text{WorkFactor} = (2^n + 1)/2$$

Dies kann approximiert werden zu

- Approximierte Formel für den Work Factor für einen zufälligen Schlüssel der Länge n bits

$$\text{WorkFactor} = 2^{n-1}$$

- Approximierte Formel für den Work Factor für einen zufälligen Schlüssel der Länge n bits, wo $n > 128$ bits

$$\text{WorkFactor} = 2^n$$

Attacktypen auf Kryptosysteme:

- **Ciphertext-only attack:** Angreifer analysiert nur den Ciphertext, um Rückschlüsse auf den Plaintext oder Schlüssel zu ziehen.
- **Chosen-ciphertext attack:** Angreifer kann Ciphertexte entschlüsseln lassen und erhält den Plaintext oder Teilinformationen.
- **Known-plaintext attack:** Angreifer kennt bestimmte Plaintexte und deren Ciphertexte und versucht, daraus Muster oder den Schlüssel abzuleiten.
- **Chosen-plaintext attack:** Angreifer wählt Plaintexte zur Verschlüsselung, um Rückschlüsse auf das System oder den Schlüssel zu ziehen.
- **Brute-force attack:** Alle möglichen Schlüssel werden getestet, bis ein sinnvoller Plaintext erscheint.

- ⇒ geometrische Reihe
(zu oder abnehmend exponentiell)
- ⇒ arithmetische Reihe
(zu oder abnehmend linear)

Digital Certificates

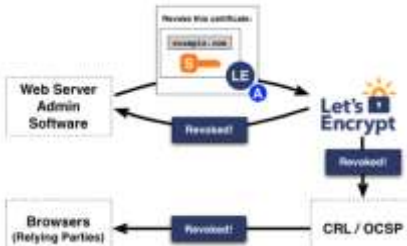
- They **bind certain things together** in a single document, e.g., server name and public key
- They are accepted as genuine because they are **issued and signed by a trusted third party**, e.g., SwissSign, Let's Encrypt, etc.
- These trusted third parties are called **Certification Authorities (CAs)**
- We trust SwissSign and Let'sEncrypt because... well, here it gets a bit tricky

TLS works on top of TCP (doesn't work on top of UDP)

Let's Encrypt: Certificate Revocation

Agent uses authorized key for certificate revocation

- Agent
 - Create revocation request
 - Sign it with authorized key A
- Let's Encrypt
 - Verify signature A
 - Confirm revocation
 - Publish revocation
 - CRL
 - OCSP
- Mechanisms
 - ISRG Root X1
 - OCSP
 - Browser-Summarized CRLs
 - ISRG Root X2
 - Browser-Summarized CRLs



Um ein Zertifikat zurückzuziehen wird ein mit dem Schlüssel A signierter revocation Request an Let's encrypt gesendet. Da A vertraut wird, wird das angegebene Zertifikat zurückgezogen. Let's encrypt verwendet aktuell sowohl OCSP als auch Browser Summarized CRLs.

Let's Encrypt ist derzeit die größte Zertifizierungsstelle (CA).

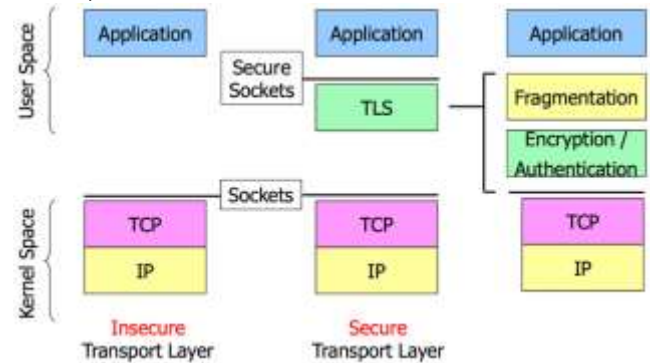
TLS 1.3 record protocol

TLS 1.3 RECORD PROTOCOL



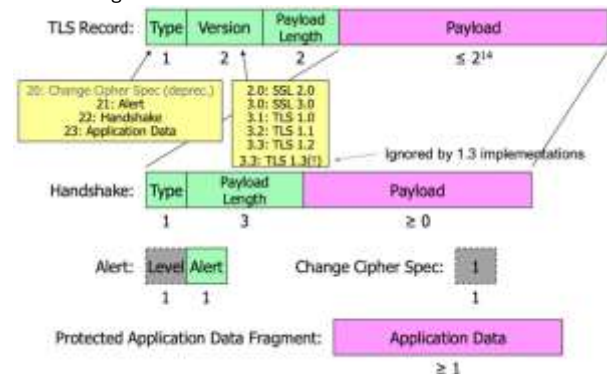
- **TLS Record Protocol:** Defines the TLS packet format; all data that are using TLS are transported within **TLS Records**
- **Handshake Protocol:** Used to establish TLS sessions

The TLS protocol stack



The TLS Layer is inserted between the Transport layer and the Application Layer. In contrast to IPSec which is a Layer 3+ protocol based directly on IPv4 or IPv6, TLS is a Layer 4+ protocol based directly on TCP

TLS message formats



TLS Phases:

TLS operates in three phases:

- **Handshake:** authentication and establishment of cryptographic algorithms and key material
 - Most complicated part of TLS, will take most time to explain
 - Doesn't consume many bytes compared to next phase
- **Data exchange:** exchange protected data
 - Usually the bulk of exchanged bytes
- **Connection teardown:** disconnect safely
 - Usually the shortest phase of all

TLS handshake high-level overview

1. Client and server Negotiate crypto algorithms
2. Client and server perform Diffie-Hellman
3. Client and server generate Handshake keys N
4. Server Authenticates to client
5. Client and server prove to one another that no one has Tampered with the previous messages
6. Client and server generate Data keys H
 - (Client and server exchange encrypted data, not part of handshake)

TLS handshake requirements

Initial situation: client and server have no previous association, no shared secrets

Attacker: Mallory: read, modify, delete, duplicate messages

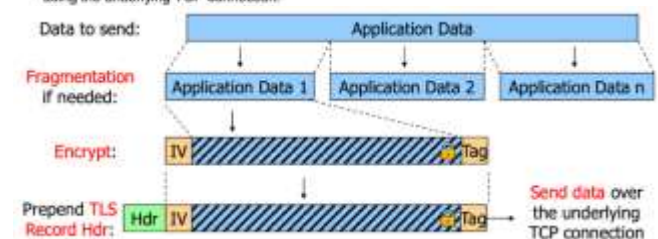
Building TLS records from application data

TLS Application Data Exchange

If the data is longer than what fits (after adding the tag, padding etc.) into the payload of an TLS record, it is split into multiple fragments.

The fragment is then encrypted. This also requires an IV, which is prepended to the encrypted data. Encryption causes an auth tag to be computed, which is added.

Finally, a TLS record header is prepended and the resulting data is sent to the communication partner using the underlying TCP connection.



Sequence number

are used to make sure that all TLS records arrive at the destination and in the correct order

- Attack scenario: The attacker (MITM) reorders / deletes / replays TLS records in a data stream
- The recipient will detect this because verifying the MAC fails
- Replay attacks are not possible!

Truncation Attack (Angriff durch Abschneiden von Daten)

Problem: Angreifer unterbricht eine TLS-Verbindung vorzeitig, indem er ein gefälschtes TCP FIN-Paket sendet.

Folge: Der Empfänger glaubt, die gesamte Nachricht erhalten zu haben, obwohl Teile fehlen → Datenintegrität wird verletzt.

Auth tag:

Computing the auth tag not only includes the application data fragment, but also a **sequence number**.

Zur Berechnung des Authentication Tags wird eine Sequence Number benötigt. Da TLS über TCP läuft, und TCP die Datenübertragung in der korrekten Reihenfolge garantiert, muss die Sequence Number nicht übertragen werden.

DTLS (Datagram Transport Layer Security, RFC 6347)

Warum DTLS? TLS funktioniert nur mit TCP, DTLS wurde für UDP entwickelt.

Hauptanpassungen an TLS:

- Explizite Sequenznummern zur Behebung von Reihenfolgenproblemen.
- Zuverlässigkeitsmechanismen für das Handshake (Timeouts, Retransmission).
- Optionale Replay-Detection, um doppelte Nachrichten zu ignorieren (Schutz gegen Replay-Angriffe).

Eigenschaft	TCP	QUIC
Transportprotokoll	TCP	UDP
Verbindungsaufbau	Langsamer (3-Wege-Handshake)	Schneller (Integrierter Handshake + TLS)
Sicherheit	TLS separat	TLS integriert
Multiplexing	Möglich, aber mit Head-of-Line Blocking	Native Unterstützung ohne Blocking
Staukontrolle	Ja	Ja (verbessert)
Hauptverwendung	Klassische Internetdienste (HTTP/2, FTP, E-Mail)	Moderne Webanwendungen (HTTP/3, Streaming, Gaming)

Verschlüsselung auf Layer 2

- **Wired Networks:**
 - Schutz gegen Abhören (physischer Zugang nötig)
 - Schutz für alte/sensible Anwendungen ohne sichere Protokolle
 - Extra-Schutz in regulierten Umgebungen
- **Wireless Networks:**
 - Abhören ohne physischen Zugang → Verschlüsselung immer nötig
 - Schutz für alte/sensible Anwendungen

Authentifizierung auf Layer 2

- Nur **authentifizierte Nutzer/Geräte** dürfen ins Netz
- Standardprotokoll: **EAP (Extensible Authentication Protocol)**

How port-based access control works



- A client connecting to a port of the switch is **first blocked**, only EAP messages are accepted
- The authenticator **relays** all EAP messages between the client and the RADIUS server
 - The **RADIUS protocol** is secured using pre-shared keys → authenticity and confidentiality between authenticator and RADIUS server
- A client gets **full network access** when the authenticator has received the «authentication successful» message from the RADIUS server
- RADIUS server can send additional data such as the VLAN ID to be used for the specific client

MACsec: MACSec ist ein Protokoll, welches zur Verschlüsselung des Netzwerkverkehrs zwischen zwei Geräten verwendet werden kann. Mit MACsec wird also aller Netzwerkverkehr verschlüsselt.

- **Confidentiality and integrity/authenticity on Layer 2 (Ethernet)**
 - secure all data including DHCP, ARP, and any higher layer protocols
 - Physical and virtual link
 - Between two devices
 - Each device on the path has access to the unencrypted data, since it decrypts the received data and re-encrypts the data to be transmitted.
- **Speed**
 - Line rate encryption with pure HW implementation
- **Cryptography**
 - Key agreement according to IEEE 802.1X (e.g., EAP-TLS) or pre shared keys
 - Cipher: GCM-AES-128 (GCM-AES-256)
- **Packet format**
 - SecTAG: pointer to key, packet number for IV
 - ICV: Integrity Check Value (MAC) based on GCM-AES



Wlan:

Ein wichtiger Unterschied von kabellosen zu kabelgebundenen Netzwerken liegt darin, dass man viel einfacher physischen Zugang zum kabellosen Netzwerk hat

WLAN "security" with wired equivalent privacy (WEP):

The AP and all clients share a **preconfigured long-term key**

- This key is used to **encrypt** individual frames
- Since all clients use the same key, every user (who knows this key) can read the traffic of every other user

The length of the key is either **40 or 104 bits**, encryption uses **RC4**

Key is often specified by entering **5 or 13 ASCII** characters or **10 or 26 HEX** symbols

- Make sure the key is **random** and not based on common words, as this may allow to get the key with a dictionary attack



A Cyclic Redundancy Check (CRC) is not a Message Authentication Code (MAC)

secure communication protocols

The basic **goals of secure communication** include the following:

- **Confidentiality** → only the communication endpoints can read the data
- **Integrity** → the endpoints can detect if data was manipulated in transit
- **Authenticity** → masquerading as an endpoint is not possible

Two **further goals** include (much less common):

- **Non-Repudiation** → an endpoint cannot deny having sent/received data
- **Anonymity** → one endpoint (or both endpoints) cannot identify the other

EAP

Ziel: Nur **authentifizierte Geräte/Nutzer** dürfen ins Netzwerk.

Ablauf:

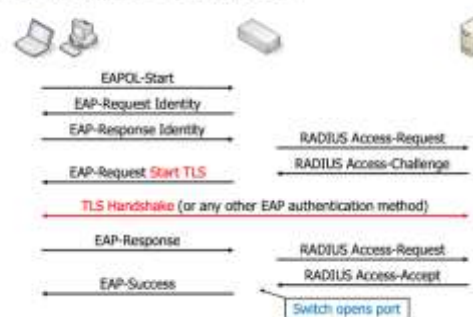
1. Switch-Port ist gesperrt.
2. Switch leitet EAP-Pakete an Auth-Server weiter.
3. Switch versteht EAP nicht, vermittelt nur.
4. Auth-Server prüft Nutzer → sagt dem Switch, ob Auth erfolgreich war.
5. Bei Erfolg: Switch öffnet den Port.

Vorteile:

- Verschiedene Auth-Methoden möglich (z.B. Passwort, 2FA)
- Switch braucht keine Nutzerdatenbank

Wichtig: EAP ist **kein Auth-Protokoll**, sondern ein **Transport-Container** für Auth-Daten.

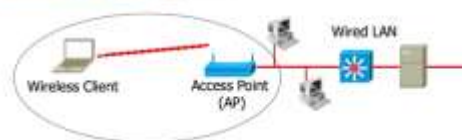
Example: IEEE 802.1X with EAP-TLS authentication



WLANs

IEEE 802.11 WLANs ("WLANs") are used everywhere today, also in companies

A typical WLAN usage scenario is as follows:



The basis is a **wired network** (e.g. Ethernet)

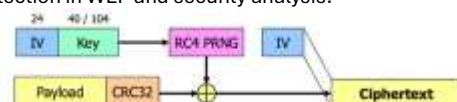
One or more **Access Points (APs)** are used to enable wireless access to the network

Wireless devices (laptops, smartphones, tablets,...) connect to the network via the access point

General security concerns with wireless networks:

- There is no cable → **sniffing packets is very easy** → data should be encrypted
- I only want **legitimate people to use my access point** (and be part of my network) → some sort of authentication needed
- This should happen at the **layer 2**, so any data exchanged between clients and AP is protected
- Even with security mechanisms available, they are **sometimes not enabled** → there are (still) a few open APs available
- IEEE 802.11 WLANs have already **quite a security history** (see following slides):
 - Original attempt: **Wired Equivalent Privacy (WEP)** → has several major design flaws
 - **Wi-Fi Protected Access (WPA)** as a «quick fix» because developing the official successor of WEP (IEEE 802.11i) took so long
 - **IEEE 802.11i** (usually named **WPA2**) as the official successor of WEP
 - **WPA3**

Frame protection in WEP and security analysis:



CRC checksum protects the payload integrity so the recipient can detect if an encrypted frame was modified in transit (does not work, see later)

40-bit keys have **too low work factor**, even when chosen randomly

104-bit keys are also on the very low end, but much better → brute force attacks very hard

Unfortunately, even with 104-bit keys, **WEP is totally insecure**

The reason is that with a 24-bit IV and a constant key, the RC4 PRNG **only generates $2^{24} = 16,777,216$ different keystreams**

→ We do **not** attack the key, but the **different keystreams!**

WiFi protected Access (WPA):

- An attacker intercepts a WEP-encrypted frame of which he knows the plaintext of the payload P_k
- The goal of the attacker is to modify the ciphertext such that the recipient gets a different P_k after decryption – without detecting the modification

- Users **cannot read** the unicast data of other users (unlike WEP)

- Based on the Advanced Encryption Standard (AES)
- This mode guarantees confidentiality and authenticity/integrity

- No weaknesses are known which make it insecure in practice

Lower layers: hop-to-hop (layer 2), difficult to deploy, more general

Main reason for using RC4 and Michael was **performance** and (to some extent) **backwards compatibility** with devices designed for WEP

Policy: Definition of who is allowed to access what
Responsible: Business Owner

Rules: Technical implementation of rules
Responsible: Firewall Administrator

Firewall: Enforcement of rules

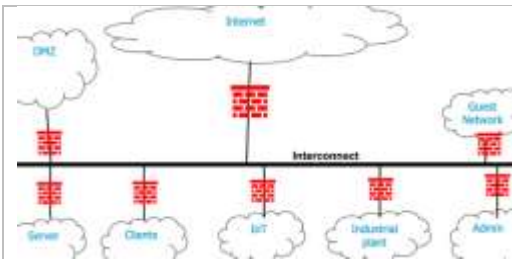
Network Segment A → [Routers] → Firewall → [allow] → Network Segment B

Firewall → [drop] → Sink

- Nur Perimeter-Schutz (nicht wirksam bei internen Angriffen)
- Umgehbar durch z. B. infizierte Geräte im Netzwerk
- Kein Schutz vor Application-Layer-Angriffen

The big picture: Adding Firewall

- NGFW offer a variety of additional features
- It is important to understand what security capabilities they offer and whether those are required



Microsegmentation: Das Ziel der Microsegmentation ist es, die einzelnen Segmente so klein wie möglich zu machen.

Grundidee:

- Innerhalb eines Segments vertrauen sich alle Maschinen gegenseitig → Risiko bei Kompromittierung.

Problem:

- Wird eine Maschine kompromittiert, sind alle anderen im Segment gefährdet.

Lösung: Microsegmentation

- Segmente so klein wie möglich machen (z. B. bis auf eine einzelne Maschine).
- Umsetzung meist durch **Netzwerkvirtualisierung** oder **lokale Firewalls**.

Tradeoff:

- **Sicherheit vs. Verwaltbarkeit (Manageability)**
→ Hohe Sicherheit = mehr Aufwand

Host Local Firewall – Überblick:

Schützt jedes Gerät individuell – unabhängig vom restlichen Netzwerk.

Firewall direkt auf jedem Gerät (Host) konfiguriert

Zusätzlicher Schutz, falls:

- Angreifer schon im Netzwerksegment ist
- Netzwerkfirewall nicht streng genug eingestellt ist

Deep Packet Inspection (inkl. TLS): Inhalte der Pakete werden geprüft

Intrusion Prevention:

- Erkennung & Blockierung von Angriffen (signature-, policy- & anomalielbasiert)

Application Awareness:

- Erkennung von Apps, nicht nur IP/Port → risikobehaftete Apps blockieren

Threat Intelligence:

- Kontinuierliche Updates mit neuen Bedrohungsdaten

Antivirus: Erkennt bekannte Viren in Datenpaketen

Sandboxing: Verdächtige Dateien werden isoliert getestet

Taking it one step further: Zero Trust

„Never trust, always verify“

→ Jeder Zugriff wird geprüft, auch im internen Netz

Nur nötige Rechte, alles wird **überwacht**

Policy Enforcement Point (PEP) = zentrale Kontrollinstanz (wie Firewall)

Seit 2020 **Standardisiert** (z. B. **NIST**)

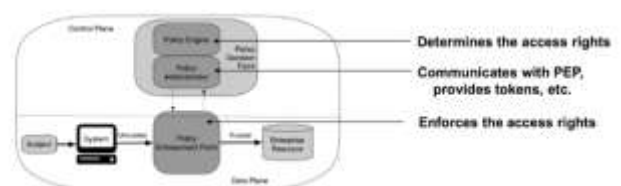
Unterschied zu Segmentierung: **kein Vertrauen innerhalb von Segmenten**

• **Concept:**

- Do not trust anything unless it is verified
- Least privilege access is enforced
- Security monitoring is implemented

• **Standardization**

- NIST Special Publication 800-207: Zero Trust Architecture, August 2020



Risiken bei Zero Trust

- **Single Point of Failure** beim PEP (Fehlkonfig. / DoS möglich)
- **Gestohlene Zugangsdaten** (z. B. durch Phishing)
- **Weniger Netzwerksichtbarkeit** wegen Verschlüsselung
- **Angreifer greift auf Monitoring-Daten** zu → sensible Infos
- **Missbrauch von Agents** (z. B. Admin-Accounts mit zu vielen Rechten)

Endpoint Detection and Response (EDR): Erweiterung von Host Local Firewalls



Vorteile:

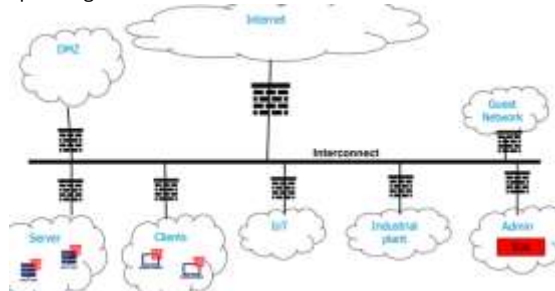
- Zentrale Verwaltung & Übersicht
- Automatisierte Reaktion auf erkannte Bedrohungen

Nachteile:

- Benötigt Agenten auf jedem Gerät → potenzielles Angriffsziel

The big picture: Adding DER

Die EDR Lösung verfügt über ein Management System, welches in der speziell geschützten Admin Zone steht.



WAF Architecture:

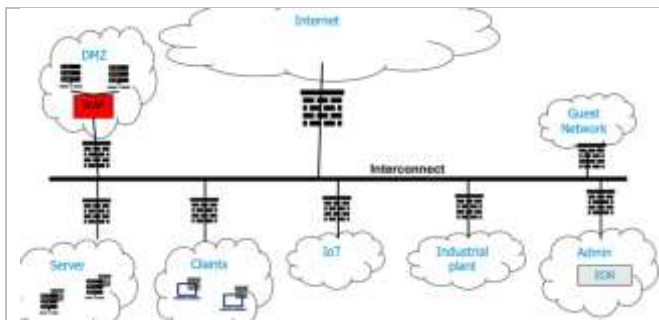
Integration einer WAF in einem Netzwerkpfad.



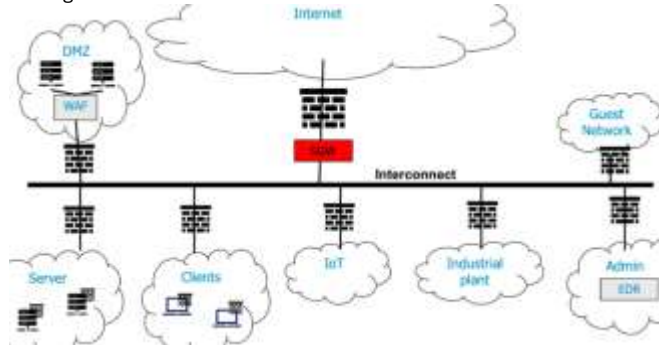
- Firewalls must deny all request directed to the servers
- DNS must be configured to point to the TLS termination
- Clients must be configured with WAF certificate to avoid certificate warnings

Adding WAF:

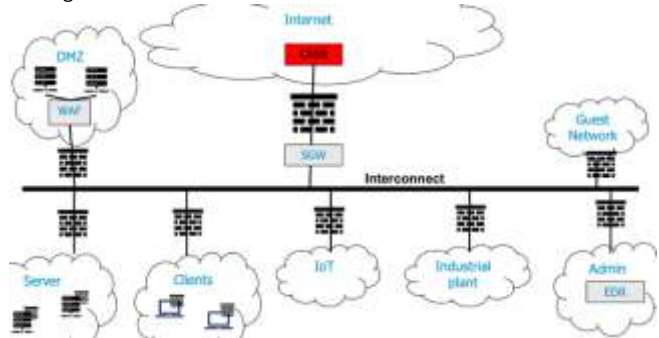
Secure Web Gateway (SWG):



Adding SGW:



Adding CASB:



Network Detection and Response

Network detection plus automatic remediation of incidents

- Changing Firewall configuration
- Isolating an infected device from the network
- Etc.

Benefits

- Speeds up reaction to incidents

Drawbacks

- Needs interfaces to other network devices to trigger response
- Wrong classifications lead to unnecessary actions



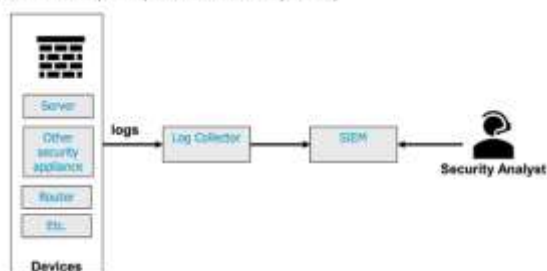
SIEM – Security Information and Event Management:

In einem SIEM werden die Logfiles von verschiedenen Servern, Security Appliances und weiteren Geräten ausgewertet.

Correlation of events provided by different log sources to find malicious activity

Consolidated view in a single dashboard

Generation of reports (also used for compliance)



Steht zwischen dem Internen Netzwerk und dem Internet. Stellt sicher, dass nur auf freigegebene Inhalte im Internet zugegriffen werden darf.

Also known as (Forward-) Proxy

Capabilities

- URL Filtering
 - Block known malicious URLs
- Data Leakage Prevention (DLP)
 - Blocking of sending sensitive information
 - Requires tagged data or some other possibility to recognize sensitive information
- TLS Inspection
- Etc.



Cloud Access Security Broker (CASB):

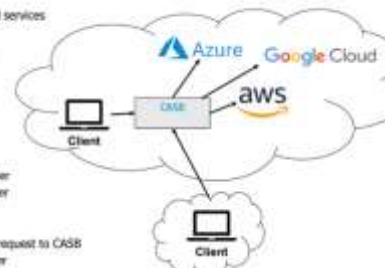
Steht zwischen einem Client und den Cloud-Diensten und überwacht die entsprechenden Verbindungen.

Capabilities

- Shadow IT discovery
 - Generation of reports of used cloud services
- Cloud usage Control
 - Set access rights to cloud services
- Data leakage prevention
 - Set policies for data sharing
- Anomaly detection
 - Alerts on unusual behavior
- Etc.

Implementation

- API Scanning: Direct API to cloud provider
 - Only Works for known cloud provider
- Forward Proxy: similar that SGW
 - Only works for managed devices
- Reverse Proxy: cloud provider redirects request to CASB
 - Only works for known cloud provider



Network Detection: Geht darum um Anomalien im Netzwerkverkehr zu erkennen.

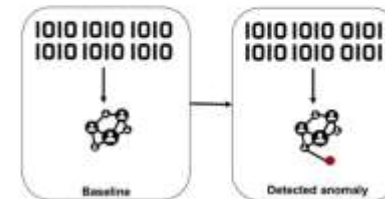
Continuous monitoring of network traffic generates baseline

Anomalies to that baseline are reported as potential incidents

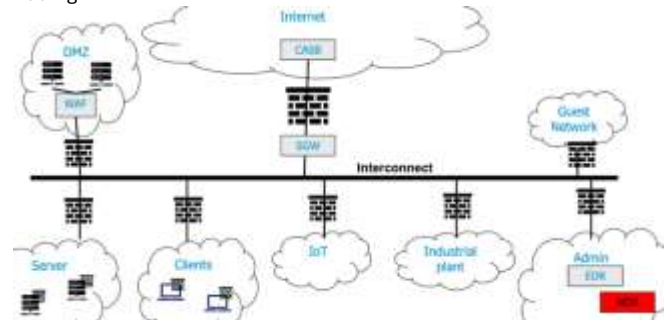
Potential incidents are analyzed by security experts and classified as actual incident (true positive) or false positive

Implementation

- Packet based
 - Analysis of all packets
- Log based
 - Analysis of network logs
 - Netflow, firewall logs, etc.



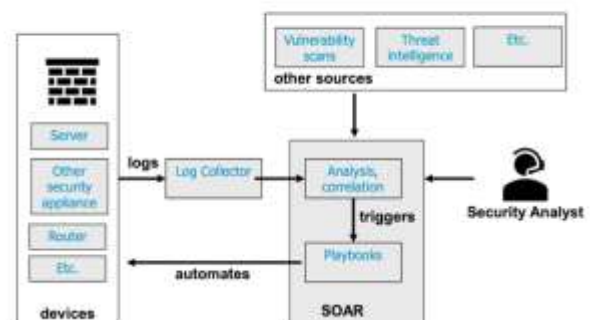
Adding NDR:



SOAR (Security Orchestration, Automation and Response):

Eine Erweiterung eines SIEMS wo man auch automatisiert Responses ausführen lassen kann. Zusätzlich zu den logs werden noch weitere Quellen analysiert

Extend capabilities of SIEM to include further sources and automate response



Soft fail:

Wenn der OSCP server nichts zurückschickt und man auf eigene gefahr und trotzdem zeigt der Browser die Website **nach einer Warnung** an

Wenn das Zertifikat z.B.:

- abgelaufen ist,
- selbst signiert ist,
- von einer unbekannten CA kommt,
- oder der Hostname nicht passt,

dann sagt der Browser:

! „Verbindung nicht sicher – Möchten Sie trotzdem fortfahren?“

Warum ist das ein Soft Fail?

Der Zugriff **wird nicht komplett blockiert**, sondern der Nutzer darf selbst entscheiden, ob er das Risiko eingehen will.

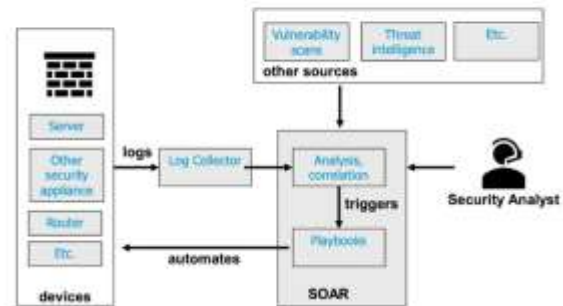
Das Zertifikat ist technisch **nicht vertrauenswürdig**, aber es **verhindert nicht den Seitenaufbau**, wenn der User zustimmt.

Dadurch können **Man-in-the-Middle-Angriffe** leichter durchrutschen, wenn der User unachtsam ist.

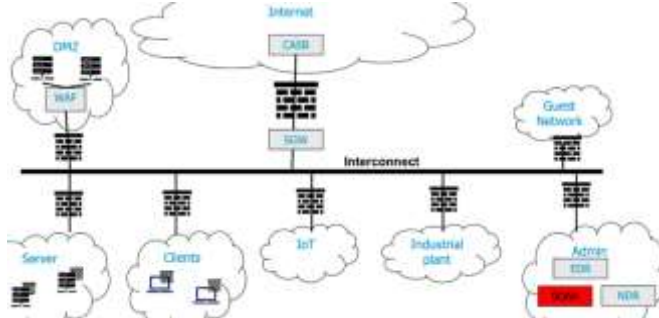
SOAR (Security Orchestration, Automation and Response)

Eine Erweiterung eines SIEM wo man auch automatisiert Responses ausführen lassen kann.

Extend capabilities of SIEM to include further sources and automate response



Adding SOAR



What is netfilter? What is nftables?

Part of Linux kernel since version 2.4

Enables packet-filtering, network address translation (NAT) and general packet mangling



netfilter is a mechanism that allows to access the packets in the network stack to analyze, modify, extract and delete them

nftables is a packet classification and mangling framework that runs on rulesets that are applied to the packets

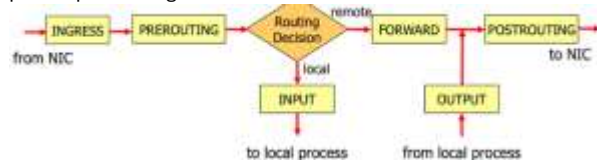
- Can be used to implement a packet-filtering firewall
- Can be used to implement IPv4 and IPv6 network address translation

nft is the name of a command line tool to configure nftables

Several commercial firewall **appliances** are based on Linux/netfilter (although probably with iptables, an earlier packet-filtering toolchain)

Netfilter and the Linux kernel:

Kernel has a number of hooks that are called at different points during packet processing



nftables rules:

Nftables rules have a **classification** part and one or more **action** parts

Classification part says to what packets this rule applies

Action says what to do with the packet. There are several actions:

- **accept**: continue to process the packet
- **drop**: stop processing the packet
- **reject**: stop processing the packet and tell the sender
- **jump**: continue processing elsewhere (see below)

Predominant recommendation is to drop unwanted packets

Other people (typically those with sysadmin experience) say that this makes it harder also for legitimate admins to administrate and debug their own networks

Some people say that drop hides the existence of a firewall, but that is **not true**, and will only be a speed bump for the attacker, nothing more

What is port scanning?

Port scanning is a technique to determine the **services that run on a host**

Often used by **attackers** to find and analyse targets, but also valuable for **system administrators** to check the own hosts and firewall configs

Corresponding software tools are called **port scanners**

For all hosts and ports they are interested in, port scanners do the following:

- Check if the host is available by **pinging** it (echo-request/echo-reply)
- Establish **TCP connections** to the ports
- If a connection can be **established** → the service is available (**open port**)
- If the server responds with a **TCP RST** → the service is not available (**closed port**)
- There are also **UDP scans** → send datagram to the ports and receive either an answer or an ICMP port unreachable message (or nothing)

Most popular port scanner: **nmap** (all platforms)

UDP Scans:

UDP scans are often not reliable because receiving no answer can mean (1) a service is listening but

does not send an answer to the UDP probe sent by the scanner or (2) a firewall silently drops the UDP probes. From the point of view of the port scanner, there's no way to distinguish these cases. With

TCP, no such problems exists because contacting an open port (with a SYN message) will always

result in a SYN/ACK message being received from the server, so open ports (that are not blocked by a

firewall) can unambiguously determined.

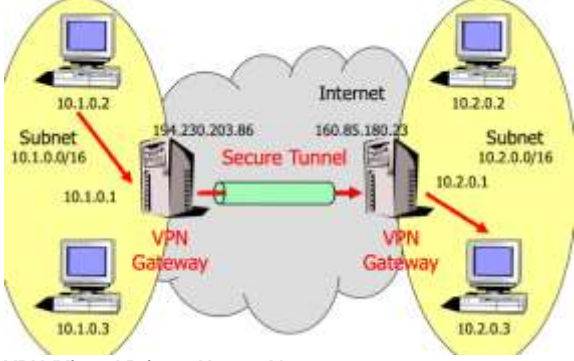



VPN

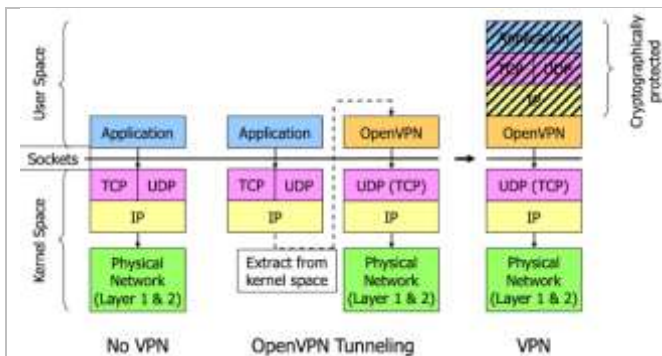
A **Virtual Private Network (VPN)** is a **private (protected) network** within a public network (e.g. Internet)

- **Private** means that outsiders can neither read nor modify the data transmitted between participants
- **Virtual** means that the privacy (protection) is not achieved by dedicated network links, but by «virtual methods» (i.e. cryptography)

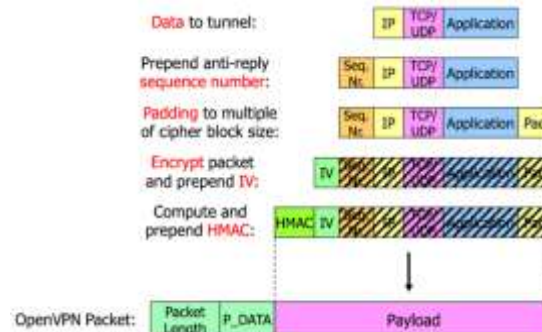
Typical usage of VPNs includes the following:

- Securely connect two (or more) **remote networks** of a company
- Allow a **partner / customer company** selectively accessing internal services in the own network
- Allow **mobile company users** access to internal services, as if they were on premise

<p>VPN Protokolle:</p> <p>Wichtige VPN-Protokolle:</p> <ul style="list-style-type: none"> • IPsec: Tunnel-Modus, lange Zeit Standard, sehr sicher. • OpenVPN: Open Source, basiert teilweise auf TLS, läuft im User Space, „leichter“ als IPsec. • WireGuard: Neues, modernes VPN-Protokoll, einfacher und schneller. <p>Nicht nur IPsec/OpenVPN:</p> <ul style="list-style-type: none"> • PPTP (veraltet) und L2TP (mit IPsec kombiniert) sind ältere Protokolle für VPNs. • L2TP/IPsec wird oft zusammen genutzt, da L2TP selbst keine Verschlüsselung bietet. <p>Hinweis: Es geht nicht um „IPsec vs. OpenVPN“ – beide haben ihre Einsatzbereiche.</p>	 <p>VPN (Virtual Private Network): Ein VPN ermöglicht sichere Kommunikation über unsichere Netzwerke (z. B. Internet) mithilfe von verschlüsselten Tunneln. Es schützt Daten durch Verschlüsselung, Authentifizierung und Integritätsprüfung.</p> <p>Secure Tunnels: VPN-Gateways verschlüsseln Daten zwischen zwei Netzen oder einem mobilen Gerät und dem Firmennetz. Die Geräte selbst merken nichts davon. Private IPs (z. B. 10.0.1.0/24) können genutzt werden.</p>
<p>IPsec – Übersicht:</p> <ul style="list-style-type: none"> • Schützt IP-Paketdaten & Teile des Headers. • Bietet Vertraulichkeit, Authentifizierung & Integrität. <p>Schlüsselaustausch:</p> <ul style="list-style-type: none"> • Verwendet IKE (Internet Key Exchange). • Heute meist IKEv2 (seit 2005), z. B. bei Windows 7+. <p>Vorteil:</p> <ul style="list-style-type: none"> • Sichert alle Protokolle oberhalb von Layer 3 (z. B. TCP, UDP, ICMP). <p>Nachteil:</p> <ul style="list-style-type: none"> • Muss im Kernel installiert werden → verändert den Netzwerk-Stack. • Kann die Systemsicherheit gefährden, da mehr Code im Kernel nötig ist. 	<p>Unterschied TLS vs. IPsec:</p> <p>TLS:</p> <ul style="list-style-type: none"> • Läuft über TCP, also zwischen zwei Anwendungen (z. B. Browser ↔ Webserver). • Schlüssel nur Anwendungen bekannt → Schutz auf Anwendungsebene. • TLS schützt die Kommunikation zwischen zwei Anwendungen. <p>IPsec:</p> <ul style="list-style-type: none"> • Schützt alle IP-Pakete zwischen zwei Hosts, unabhängig von der Anwendung. • Schlüssel nur den Hosts bekannt, Anwendungen wissen nichts vom Schutz. • IPsec schützt die Kommunikation zwischen zwei Hosts.
<p>IPsec Handshake (IKE):</p> <p>Aufgabe von IKE (ähnlich wie TLS-Handshake):</p> <ul style="list-style-type: none"> • Auswahl der Krypto-Algorithmen • Durchführung gegenseitiger Authentifizierung • Austausch von Schlüsseldaten (meist über Diffie-Hellman) <p>Zwei gängige Authentifizierungsarten:</p> <ul style="list-style-type: none"> • Digitale Signaturen & X.509-Zertifikate • Pre-shared Secrets (gemeinsames Passwort → Hash + DH-Hälfte). 	<p>Sicherheit von IKE</p> <ul style="list-style-type: none"> • Keine bekannten großen Schwachstellen. • Kritik: zu komplex – schwer zu analysieren, sollte einfacher sein. • Komplexität stammt u. a. aus Standardisierungsprozessen. <p>IPsec Modes</p> <p>Transport Mode = Rarely used</p> <p>Tunnel Mode = Used in VPNs</p>
<p>IPsec Tunnel Mode:</p> <ul style="list-style-type: none"> • Verwendet das ESP-Protokoll, um das gesamte ursprüngliche IP-Paket zu schützen. • Dieses geschützte Paket wird in ein neues IP-Paket eingebettet (für den Transport zwischen VPN-Gateways). • Ziel-Gateway entpackt das ursprüngliche Paket und leitet es weiter. • Vorteil: Versteckt interne IP-Adressen → mehr Privatsphäre + Nutzung privater Adressen möglich. 	<p>Struktur eines Pakets mit IPsec/ESP:</p> <ul style="list-style-type: none"> • Originales IP-Paket → wird komplett verschlüsselt • Zusätzliche Header sorgen für Integrität & Authentifizierung <p>IP packet without IPsec/ESP (received by the VPN gateway)</p>  <p>IP packet protected with IPsec/ESP (forwarded by the VPN gateway)</p> 
<p>Sequence Numbers bei IPsec:</p> <ul style="list-style-type: none"> • IPsec nutzt Sequenznummern, um Replay-Angriffe zu erkennen. • Jeder ESP-geschützte IPsec-Paket erhält eine eindeutige Sequenznummer. • Der Empfänger entscheidet anhand dieser Nummer: <ul style="list-style-type: none"> ○ Akzeptieren, wenn die Nummer neu und gültig ist. ○ Verwerfen, wenn die Nummer doppelt oder zu alt ist. ○ Session beenden, nur bei verdächtigen Mustern (z. B. Angriffsverdacht) 	<p>Aufgabe: Bewertung der Pakete:</p>  <p>Sequenznummern & Aktionen:</p> <p>1 ✓, 2 ✓, 4 ✓, 5 ✓, 6 ✓, 4 ✗ (Replay), 7 ✓, 3 ✗ (zu alt außerhalb Window).</p> <p>TLS erwartet strikte Reihenfolge – doppelte/alte Pakete → Verbindung kann beendet werden (Manipulationsverdacht).</p>
<p>OpenVPN:</p> <ul style="list-style-type: none"> • Nutzt TLS-Handshake für Authentifizierung & Schlüsselaustausch. • Schützt IP-Pakete zwischen Endpunkten wie IPsec. • Plattformunabhängig: Linux, Windows, macOS, xBSD. <p>Routing vs. Bridging:</p> <ul style="list-style-type: none"> • Routing: getrennte Subnetze, einfacher & gängig in der Praxis. • Bridging: simuliert eine LAN-Verbindung (inkl. Broadcasts), z. B. für Layer-2-VPNs. 	<p>Packet format:</p> <p>Ein OpenVPN-Paket besteht aus: Header (3 Byte) + Payload (n Byte)</p> <p>Bei UDP ist immer ein Paket = ein Datagramm. Opcode wird nicht verwendet bei Pre-Shared Keys.</p>



Protection of IP packets and encapsulation (assuming a block cipher in CBC mode):



IPSec vs. OpenVPN – Vergleich:

- Beide sicher & stark – kein Grund zu wechseln, wenn eines gut funktioniert.
- Sicherheitsniveau: Gleich – beide nutzen starke, bewährte Algorithmen.

Vorteile von IPSec

- Reifer & professioneller (viele kommerzielle Lösungen mit Support)

Vorteile von OpenVPN

- Weniger komplex → einfacher zu konfigurieren
- Läuft im User Space

Access Control Komponenten:

Identification Identification establishes who you claim to be: The user claims an identity, usually by supplying a user ID or a user name.

Authentication Authentication verifies that you are who you claim to be: The user supplies authentication information, which proves the binding between the user and the identity.

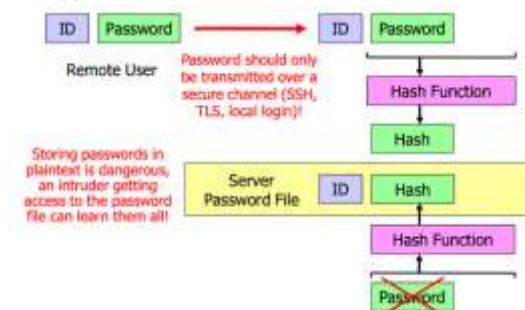
Authorization Authorization establishes what you're allowed to do e.g. which files and applications you may access: The system authorizes the (authenticated) user to do what he is allowed to do.

Audit Trail The Audit trail keeps track of what you have done.

Methoden zur Authentifizierung:



PW Hash:



WireGuard:

- Neues, leichtgewichtiges VPN-Protokoll
- Läuft auf Layer 3 (wie IPSec)
- Einfach konfigurierbar, feste Krypto & kein Protokollwechsel → kleine Angriffsfläche
- Moderne Kryptografie: ChaCha20, Curve25519
- 1-RTT Handshake, Perfect Forward Secrecy (PFS)
- Eingebaute DoS-Schutzmechanismen

Message Flow (wg0-Interface):

- Daten laufen durch ein virtuelles Interface wg0
- Kommunikation via UDP über IP
- Anwendung ↔ IP ↔ wg0 ↔ physisches Netz ↔ Peer

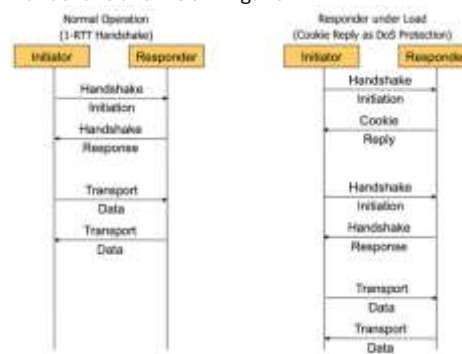
Interner Ablauf (Routing mit Public Keys):

- Ziel-IP → bestimmt öffentlichen Schlüssel
- Public Key → bestimmt Session Keys
- Paket wird über wg0 empfangen und geprüft
- Falls falsche Quell-IP nach Entschlüsselung → verworfen

Internet Endpoint-Verhalten

- Wenn gesetzt → Paket geht an festgelegte Adresse (z. B. 192.95.5.64:21841)
- Wenn nicht gesetzt → Antwort an die Absenderadresse des letzten gültigen Pakets
- Mobiler Peer? Kein Problem – Endpoint wird bei nächstem Paket automatisch aktualisiert

Handshake and DOS mitigation:



Dictionary-Angriff:

Ein Angriff, bei dem typische Passwörter aus einer Wortliste ausprobiert werden. Funktioniert, weil viele Nutzer einfache Passwörter wählen.

Precompiled Dictionary Attacks (vorkompilierte Wörterbuchangriffe)

- Angreifer berechnen im Voraus Hashes für viele häufige Passwörter → sogenannte precompiled attacks.
- Beim Angriff wird nur noch der Hash-Vergleich durchgeführt – das spart Rechenzeit.
- Solche Listen können riesig werden (z. B. 6-stellige Passwörter = über 192 Mrd. Kombinationen → mehrere TB Speicherplatz nötig).
- Rainbow Tables helfen, den Speicherplatz zu reduzieren.

Schutz vor precompiled dictionary attacks

- Ziel: Angreifer soll nicht mit vorberechneten Hash-Listen angreifen können.
- Lösung: Salting → vor dem Hashen wird ein zufälliger Wert (Salt) zum Passwort hinzugefügt.
- Salt ist z. B. 64–128 Bit lang und wird mit dem Passwort gespeichert.
- Dadurch muss der Angreifer jeden Hash einzeln berechnen, weil er das Salt nicht vorhersagen kann.

Force attacker to crack each password individually: salting

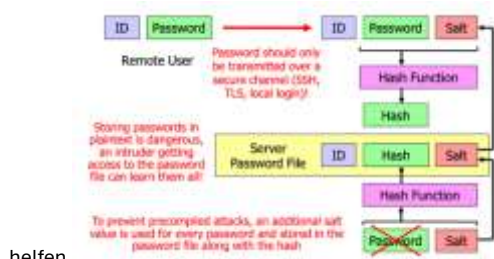
- Angreifer nutzt vorberechnete Hashes häufiger Passwörter.
- Vergleich der Hashes spart Zeit → gefährlich bei schwachen Passwörtern.

Salting als Schutz

- Zufälliger Wert (Salt) wird vor dem Hashen ans Passwort gehängt.
- Jeder Nutzer hat ein anderes Salt → Hashes sind einzigartig.
- Angreifer kann nicht vorrechnen, muss jedes Passwort einzeln knacken.

Wichtig

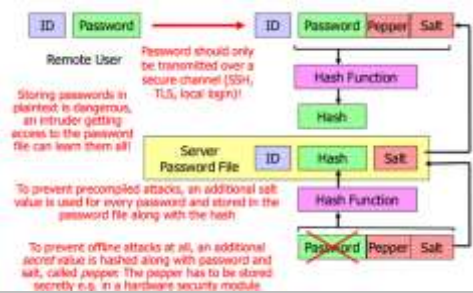
- Salt + sicheres Hash-Verfahren (z. B. SHA-2).
- Schwache Passwörter bleiben riskant → Account-Sperren



helfen.

Force attacker to crack each password individually: salting

- Peppering: zusätzlicher Schutz
- Pepper = geheimer Wert, wird wie Salt vor dem Hashen ans Passwort angehängt.
- Unterschied: Salt wird im Passwortfile gespeichert, Pepper wird geheim gehalten, z. B. in einem HSM (Hardware Security Module).
- Ziel: Selbst wenn Angreifer das Passwortfile mit Salt und Hash kauft, fehlt ihm der Pepper → kein Cracken möglich.
- HSM ist speziell geschützt – schwerer zu hacken als normale Server.
- Damit ist dein Passwort selbst bei gestohlenem Passwortfile besser geschützt.



MFA:

- Kombination aus zwei Faktoren unterschiedlicher Kategorien:
 - Wissen (z. B. Passwort)
 - Besitz (z. B. SMS-Code, Token)
 - Sein (z. B. Fingerabdruck)
- Beispiel: Passwort + SMS oder Passwort + Authenticator-App
- Erhöht die Sicherheit – Angreifer kann sich mit Passwort allein nicht einloggen

Man-in-the-Middle (Inline Attack)

- Nutzer wird auf gefälschte Login-Seite gelockt (z. B. per Phishing-Mail)
- Angreifer leitet Passwort + MFA-Code live an echte Seite weiter
- Schutz: Awareness! („Deine Bank fragt nie per E-Mail nach einem Login“)

2. MFA Fatigue

- Nutzer wird mit MFA-Anfragen bombardiert, bis er genervt auf „OK“ klickt
- Schutz: Eingabe einer zufälligen Zahl beim Login → Nutzer muss Zahl bewusst eingeben → verhindert versehentliches Bestätigen

3. Angriffe auf SMS-/Handy-basierte MFA

- Handy gestohlen → Zugriff auf SMS-Codes
- Malware liest SMS aus
- SIM Swap Fraud: Angreifer lässt sich neue SIM ausstellen und erhält deine SMS
- NIST empfiehlt mTAN nicht mehr für 2FAwh

SAML Terminology:

SAML	Description	OpenID Connect
Service Provider (SP)	Website where user wants to login	Relying Party (RP)
End-User	Person that wants access	End-User
Identity Provider (IdP)	Entity that manages the accounts	OpenID Provider
SAML assertion/token	Specifies the Identity	ID & Access Token

Direct and indirect user authentication:

Direkte Authentisierung:

Benutzer loggt sich direkt beim Dienst ein → Dienst prüft selbst.
Einfach, aber Dienst muss Passwörter kennen.

Indirekte Authentisierung:

Dienst leitet Login-Daten an externen Auth-Server weiter.

Zentrale Prüfung, mehr Sicherheit.

Beispiele: RADIUS, SAML, OpenID Connect

Abstract Protocol Flow:

Protected resource	Thing to be protected
Resource owner	Entity granting access (usually a person)
Resource server	Server hosting the protected resources
Client	Entity requiring access to protected resource
Authorization server	Server issuing access tokens

Das Protokoll funktioniert wie folgt:

1. Der Client (im Beispiel vom letzten Slide entspricht dies dem Print Server) schickt einen Authorization Request an den Resource owner (dem user aus dem letzten Slide).
2. Der Resource owner erteilt die Genehmigung
3. Der Client beantragt ein Access Token beim Authorization server (im Beispiel auf dem letzten Slide nicht gezeichnet)
4. Der Authorization server erstellt das Access Token (kryptographisch geschützt)
5. Der Client schickt das Access Token an den Resource server (die Bilddatenbank im Beispiel vom letzten Slide)
6. Der Resource server schickt die geforderte Ressource an den Client

OpenID Connect Terminology

OpenID Connect	Description	Corresponds to
Relying Party (RP)	Website where user wants to login	Client
End-User	Person that wants access	Resource Owner
OpenID Provider	Entity that manages the accounts	Authorization Server
ID & Access Token	Specifies the Identity	Access Token
claim	Specifies a property of the End-User	

Claims sind Angaben über den Benutzer (z. B. Name, E-Mail, Geburtstag).

Sie beschreiben den End-User und werden im ID Token mitgeliefert.

Standard Claims z. B.:

sub, name, email, birthdate, locale, email_verified, address usw.

Weitere eigene Claims können hinzugefügt werden.

- Der End-User will sich bei einer Website anmelden (nicht gezeichnet)
- Die Website (Relying Party) schickt eine Nachricht an den OpenID Provider mit der Bitte, den End-User zu authentisieren.
- Der OpenID Provider führt die Authentisierung durch. Dabei erteilt ihm der End-User auch die Berechtigung, die gewünschten Daten an die Website zu schicken.
- Der OpenID Provider schickt eine Nachricht über die erfolgreiche Authentisierung an die Website.
- Die Website fragt den OpenID Provider nach User Profile Information (den claims).
- Wenn der Benutzer die Erlaubnis zum Teilen der Claims erteilt hat, schickt der OpenID Provider die Claims an die Website. So können z.B. Adressdaten direkt übernommen werden und müssen nicht separat vom Benutzer ausgefüllt werden.

SAML (Security Assertion Markup Language):

- Bietet Single Sign-On (SSO) für Webanwendungen.
- Nur für Browser-Anwendungen
- Kein Support für mobile Geräte oder APIs.
- Sehr flexibel
- Nutzt XML für Assertions (kann fast alles übertragen).
- Nachteil:
- XML ist schwergewichtig → nicht für mobile Apps geeignet.

OpenID Connect vs SAML

OpenID Connect

- Nutzt leichtgewichtiges JWT und basiert auf HTTPS-Verschlüsselung.
- Unterstützt Web-, Mobile- und API-Zugriffe.
- Einfacher zu implementieren und moderner (noch in Entwicklung).
- Dient nur zur Identitätsbestätigung (keine Berechtigungen).
- Nutzer kann selbst entscheiden, welche Attribute (Claims) weitergegeben werden.

SAML

- Nutzt schwergewichtiges XML mit integrierter Verschlüsselung.
- Unterstützt nur Webanwendungen (keine mobilen Geräte oder APIs).
- Komplexer in der Implementierung, aber etablierter Standard (seit 2005).
- Überträgt auch Berechtigungen vom Identity Provider zum Service Provider.
- Die Implementierung bestimmt, welche Attribute übermittelt werden.

OpenID Connect: gut für moderne, flexible Anwendungen mit API-Zugriff.

SAML: gut für Unternehmen mit etablierten, browserbasierten Lösungen und Berechtigungsmanagement.

Kerberos:

Kerberos ist ein Authentifizierungsprotokoll, das auf symmetrischer Kryptographie basiert und mit einem zentralen Key Distribution Center (KDC) arbeitet. Es verwendet ein Ticket-basiertes Verfahren, um Benutzer sicher gegenüber Diensten zu authentifizieren. Keine Passwörter werden im Klartext übertragen.

Zeitstempel verhindern Replay-Angriffe.

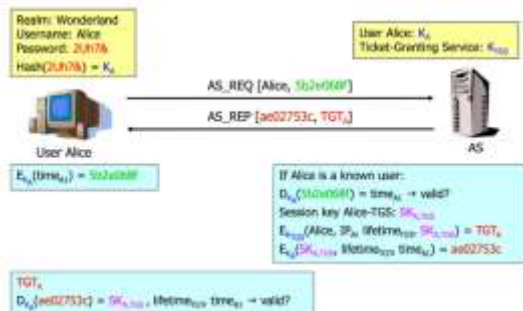
Tickets sind zeitlich begrenzt gültig (z. B. 12h).

Teilnehmer müssen Uhren synchronisiert haben.

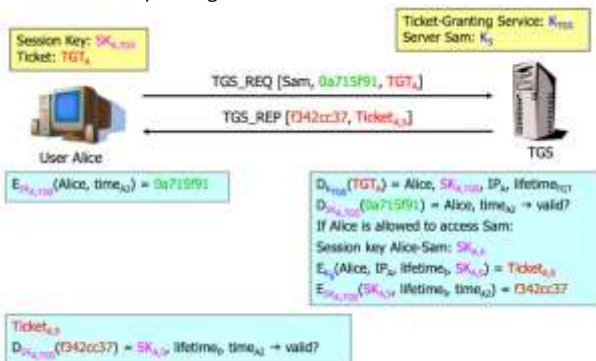
- Alice → AS: Login-Anfrage mit Benutzername → erhält TGT(spezielles Ticket für Zugriff auf TGS).
- Alice → TGS: nutzt TGT, um Ticket für z. B. Server Sam zu bekommen.
- Alice → Sam: verwendet das Ticket, um Zugriff zu bekommen.
- Sam → Alice: prüft Ticket + Zeitstempel → Zugriff gewährt.

- Der Principal schickt eine Anfrage an den Authentication Service.
- Der Authentication Service sendet ein Ticket an den Principal.
- Der Principal schickt eine Anfrage an den Ticket-Granting-Service.
- Der Ticket-Granting-Service schickt ein Ticket an den Principal.
- Der Principal schickt ein Ticket an einen anderen Principal.

Kerberos V5: getting a ticket-granting ticket (TGT)

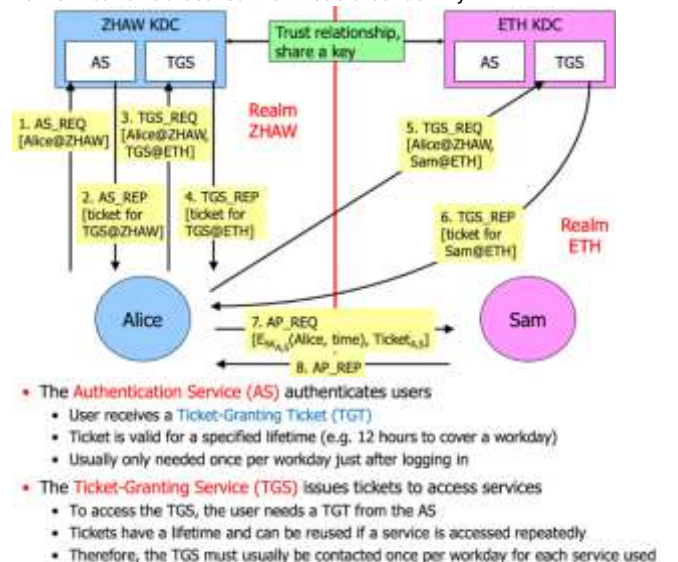


Kerberos V5: requesting a ticket to access Server Sam



Kerberos V5: accessing Server Sam:

Authentication across realms – Federated Identity:

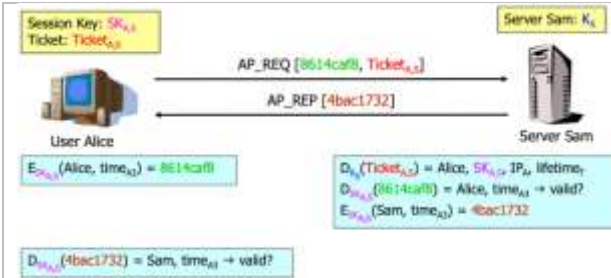


Shibboleth & Federated Identity Management:

Shibboleth ist ein System zur föderierten Identitätsverwaltung. Benutzer authentifizieren sich einmal in ihrer Heimatorganisation (z. B. ZHAW). Diese Organisation stellt ein Token aus, mit dem der Benutzer Dienste anderer Organisationen (z. B. ETH) nutzen kann. Die tatsächliche Identität muss nicht preisgegeben werden – nur Attribute wie „ist ZHAW-Student“.

- Student greift auf Ressource (SP) zu → wird umgeleitet zur IdP-Auswahl
- Discovery Service zeigt Liste der Organisationen → Benutzer wählt seine
- SP stellt Authentifizierungsanfrage an den gewählten IdP
- IdP prüft Login & sendet signierte Assertion (mit Attributen) an SP

SP prüft die Attribute → bei Erfolg erhält der Benutzer Zugriff



Authorisation

- Access Control Model: Konzeptuelles Rahmenwerk, das vorgibt, wie Zugriffskontrolle grundsätzlich möglich ist.
- Security Policy: Legt fest, wer auf welche Ressourcen wie zugreifen darf – unabhängig von der Technologie.
- Security Mechanism: Konkrete technische Umsetzung der Security Policy, z. B. durch Methoden oder Datenstrukturen.

Mandatory Access Control (MAC)

- Systemweite, zentrale Zugriffspolitik: Nur Administrator kann Rechte festlegen.
- Nutzer können Zugriff nicht selbst delegieren oder ändern (im Gegensatz zu DAC).
- Sehr hohe Sicherheit, da Umgehung fast unmöglich ist.
- Eingesetzt in sicherheitskritischen Bereichen (Militär, Behörden).
- Bekannte Modelle: Bell-LaPadula (Vertraulichkeit), Biba (Integrität).

MAC in Betriebssystemen

Wird meist nur für kritische Systemteile verwendet (z. B. Browser, Serverprozesse).

Beispiele für Mechanismen:

- Windows: Mandatory Integrity Control (MIC) seit Vista (label-basiert)
- Linux: SELinux (label-basiert), AppArmor (name-basiert)

MAC wird oft zusammen mit DAC genutzt → erst MAC, dann DAC.

Mandatory Integrity Control (MIC) – Windows

Vergibt Integrity Levels (IL) an Prozesse & Objekte:

- z. B. Installer > System > High > Medium > Low > Untrusted

Ziel: "no write-up"-Regel → niedriger IL darf nichts mit höherem IL überschreiben.

Prozesse erben IL vom Aufrufer (können aber mit reduziertem IL gestartet werden).

IL steht in der SACL des Security Descriptors (nicht vom Benutzer änderbar).

Beispiel: IL auf dem Dateisystem

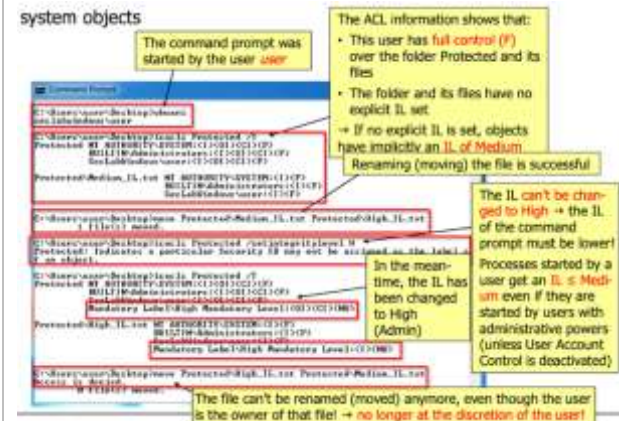
Objekte ohne expliziten IL gelten als Medium.

Benutzer mit Adminrechten erhalten trotzdem nur IL ≤ Medium (wegen UAC).

Änderung des IL nur durch Prozesse mit entsprechend hohem IL möglich.

Ergebnis: Benutzer kann Datei nicht mehr verschieben, obwohl er Owner ist → nicht mehr unter seiner Kontrolle.

system objects



Role-Based Access Control (RBAC)

Warum ein neues Modell?

- DAC/MAC sind technisch: Zugriff auf Systemobjekte, schwer auf echte Benutzer übertragbar.

Discretionary Access Control (DAC)

Besitzer entscheidet, wer Zugriff hat.

- z. B. Windows, Linux, Mac OS.
- root (Linux) bzw. Admin (Windows) kann Einschränkungen umgehen.
- Owner ist meist Ersteller des Objekts.

Access Matrix (theoretisch):

- Riesige Matrix aus Subjekten × Objekten → nicht praktikabel.

Access Control Lists (ACL)

- Pro Objekt wird festgelegt, wer was darf.
- In Metadaten gespeichert.
- In Windows, Linux, Mac OS verwendet.
- Probleme: ineffiziente Prüfung, keine flexible Delegation.

ACLs in Linux

- Meist primitive Versionen (Owner, Group, Others).
- POSIX Permissions → nicht echte ACLs.
- Echte ACLs über Linux Security Module, aber selten genutzt.

Vergleich: Windows vs. Linux (ACLs)

Merkmale	Windows	Linux
Anzahl Rechte	13 vordefinierte	3 (read, write, execute)
Subjektanzahl	Beliebig viele (Nutzer + Gruppen)	Owner, Group, Others
Vererbung	Ja	Nein
Effektive Rechte	Kombination aller Einträge (ACLs)	Rechte je nach Gruppenzugehörigkeit

Capabilities (Alternative zu ACLs)

- Token mit Rechten → im Besitz eines Subjekts.
- Speicherung spaltenweise (user → object).
- Können weitergegeben werden → DAC-ähnlich.
- Vorteile:
 - Effizient, direkt prüfbar.
- Nachteile:
 - Schwer widerrufbar, schwer audittierbar.

Beispiel:

- Token = {Port 80, {read, write}}
- Schutz durch MACs / digitale Signaturen.

Revocation mit Indirection

- Statt Token direkt: Token auf „Link“ geben.
- Zugriff wird entzogen, indem Link ins Leere zeigt (z. B. MitM → Point to nowhere).

ACL vs. Capabilities: Vergleich

ACL	Capabilities
intuitiv (Nutzer/Gruppen sichtbar)	Effizient (Tokenprüfung)
Leicht widerrufbar	Nicht widerrufbar
Delegation durch Owner/Admin	Delegation durch jeden Tokenbesitzer
Prüfung ineffizient/komplex	Zugriff = Tokenprüfung

- In der Business-Welt sind Rollen (z. B. Softwarearchitekt, Projektmitglied) sinnvoller.
- → RBAC (Role-Based Access Control): Zugriff basiert auf Rolle, nicht Benutzeridentität.

RBAC – Aufbau und Standard

Definiert in INCITS 359 (aktuell: 2017).

Komponenten:

- **Core RBAC** (Pflicht): Rollen, Benutzer-Rollen-Zuordnung, Rollen-Berechtigungen.
- **Hierarchical RBAC** (optional): Rollen können Rollen erben.
- **Constraint RBAC** (optional): z. B. Trennung von Pflichten.

RBAC – Funktionsweise

Rollen definieren erlaubte **Transaktionen** (z. B. „update savings db“).

Benutzer werden **Rollen** zugewiesen.

Transaktionen beinhalten eine **Aktion + zugehörige Objekte**.

Verwaltung zentral durch eine **Autorität**.

RBAC & Sicherheitsprinzipien

Least Privilege (POLA): Nur nötige Rechte.

Separation of Duties: Kritische Aktionen erfordern mehrere Rollen.

Data Abstraction: Rechte auf Ebene von Aktionen statt Objekten.

Einschränkungen von RBAC

Nur **eine Dimension:** Rolle → keine Attribute wie Ort, Zeit, Alter, etc.

Rollenspagat: Viele Attribute = viele Rollen (z. B. „Adult Premium“, „Child Basic“).

Betriebssysteme stellen RBAC oft **nicht bereit** → Anwendung muss es selbst umsetzen.

ABAC (Attribute-Based Access Control)

- **Erweiterung oder Ersatz** von RBAC.
- Zugriff basierend auf **Attributen**:
 - **Subjekt:** Alter, Status, Mitgliedschaft etc.
 - **Umgebung:** Uhrzeit, Ort, Bedrohungsstufe ...
- **Dynamisch, flexibel**, ermöglicht z. B. anonyme Authentifizierung über Tokens (SAML).
- Häufig kombiniert mit RBAC → RBAC definiert Basiszugriff, ABAC erweitert ihn.

Beispiel: Jakarta EE

- Benutzer-Rollen-Zuordnung z. B. über Datei oder Datenbank.
- Ressourcen-Zuordnung (z. B. URLs) deklarativ per XML oder Annotation.

Confused Deputy Problem

- Entsteht, wenn ein Programm **mehr Rechte** hat als der aufrufende Benutzer.
- Beispiel: Ein Programm läuft mit root-Rechten und erhält einen Pfad (z. B. `/etc/passwd`) von einem Benutzer, der darauf **eigentlich keinen Schreibzugriff** hat.
- Problem: Das Programm kann nicht unterscheiden, ob der Zugriff vom **berechtigten Benutzer** kommt.
- Lösung: Programm müsste den aufrufenden Benutzer **explizit überprüfen** – das ist fehleranfällig.

Beispiel: Kompilerservice

- Alice ruft Kompilerservice auf (läuft als root).
- Gibt Zielpfad `/etc/passwd` an → darf eigentlich nicht schreiben, kann es aber doch (über den Dienst).
- Lösung wäre: Das Programm verknüpft Designator mit dem aufrufenden Subjekt.

Lösung mit Capabilities

- Capability = Berechtigungstoken (z. B. für `/etc/passwd`, nur „read“).
- Designator + Berechtigung wird übergeben → kein Confused Deputy Problem.
- Zugriff scheitert, wenn Capability nicht ausreichend (z. B. nur read, kein write).
- Unterstützt Principle of Least Authority (POLA): Programme bekommen nur die minimal nötigen Rechte.

Access Control List	Capabilities
<ul style="list-style-type: none"> + Subjects (users, groups) are often associated with clearly identifiable entities and are therefore intuitive to use + Determine who is allowed to do what with a specific resource is efficient + Revocation is straight forward - Determine what a specific subject is allowed to do is inefficient - Checking of ACLs can be complex - Confused deputy problem 	<ul style="list-style-type: none"> + Authorization is efficient (check token) + No designation without authority + Principle of least authority (POLA) - Tokens can't be revoked - Determine which subjects can access a specific object is difficult - Auditing is difficult: No token-to-principal binding
Delegation: By the owner and/or administrator only	Delegation: Anyone having the token can pass it on

Kombinieren von ACLs und Capabilities

Beispiel: File Descriptors in Unix

- ACL prüft Berechtigung bei `open()`.
- Danach bekommt Prozess **File Descriptor** (= Capability).
- Bei `read()/write()` wird nur **Descriptor geprüft** – keine ACL mehr nötig.
- Vorteile:
 - Effizient, delegierbar, sicher im Kernel gespeichert.

Wichtige Begriffe

Designator: Pfad oder Objektname, z. B. `/etc/passwd`.

Capability: Unverfälschbares Token mit Berechtigung.

POLA: Principle of Least Authority → Nur minimale Rechte zuweisen.

	Based on	Models	Rules made by	Configured by	Enforced by
DAC	identity <i>e.g., computer, user, group</i>	no standard model (OS specific impl.) ACLs and capabilities are two different approaches to DAC	owner <i>typically restricted by (un)written policies/guidelines</i>	owner <i>administrator has the power to override</i>	OS
MAC	security level <i>e.g., (un)classified, restricted, secret, top secret</i>	• Windows MIC (label-based) • SELinux (label-based) • AppArmor (name-based)	security officer	admin(s) <i>labels and rules</i>	OS
RBAC	role <i>e.g., job function</i>	INCITS 359:2004: • Core RBAC • Hierarchical RBAC • Constraint RBAC Non-standard: • SELinux • Java EE	Business/security officer	application or OS admin(s)	RBAC System <i>transparent such as in SELinux or application-aware such as in RBAC-enabled applications</i>

Standard Unix Permissions

Levels:

- - No access
- x Execute
- r Read
- w Write

string representation	numerical representation	single number representation
---	000	0
--x	001	1
-w-	020	2
-wx	021	3
r--	400	4
r-x	401	5
rw-	420	6
rwX	421	7

Specifying Access Rights

When specifying access rights, this is always done in order

user, group, others

- `-rwx---` is equal to `700`
- `-rwxr--` is equal to `710`
- `-rwxr-x` is equal to `711`
- `-rwxr-xr-x` is equal to `755`

File Type Codes

- - Regular File
- b Block special File
- d Directory
- l Symbolic link
- n Network file
- p FIFO
- s socket

to delete or rename a file in a directory, the user not only needs write-rights on the directory, **but also execute-rights on the directory**

Bei der Public Key Kryptographie wird der [öffentliche] Schlüssel des [Empfängers] zur Verschlüsselung verwendet.
Bei der Entschlüsselung wird der [private] Schlüssel des [Empfängers] verwendet.
Beim Erstellen der Signatur wird der [private] Schlüssel des [Senders] verwendet.
Beim Überprüfen der Signatur wird der [öffentliche] Schlüssel des [Senders] verwendet.
Eine Hash Funktion verwandelt Input von [beliebiger] Länge in einen Output mit [fixer] Länge.
Bei Hash Funktionen kann aus dem [Output] keinen Rückschluss auf den [Input] gemacht werden.
Es ist auch nicht innert nützlicher Zeit möglich zwei [Input] mit demselben [Output] zu generieren.

1. GCM, 2.CBC, 3.ECB (Höchste Sicherheitsgarantie oben,rsp 1)

Alice und Bob wollen mittels Diffie-Hellman Key Exchange ein gemeinsames Geheimnis berechnen. Dabei gehen sie wie folgt vor:

- Sie wählen $p = 7$ und $g = 5$
- Alice wählt $a = 3$
- Bob wählt $b = 4$

$$SA = SB = g^{ab} \bmod p = 5^{12} \bmod 7 = 1$$

Sie arbeiten mit ihrem Laptop im lokalen Netz (192.168.253.0/24):

- Ihr Laptop hat die IP-Adresse 192.168.25.50 und die MAC-Adresse C3:C3:C3:C3:C3:C3.
- Die IP-Adresse des Default Gateways im Netz ist 192.168.25.1 und dessen MAC-Adresse B2:B2:B2:B2:B2:B2.
- Zudem gibt es einen DNS-Server außerhalb des lokalen Netzes mit IP-Adresse 192.168.45.101 und MAC-Adresse A1:A1:A1:A1:A1:A1.

Sie starten Wireshark um zu untersuchen, was für Pakete bei ihrem Host vorbeikommen. Plötzlich sehen Sie dabei wiederholt das folgende ARP-Reply Frame:

- Sender: ES:ES:ES:ES:ES:ES
- Empfänger: C3:C3:C3:C3:C3:C3
- Inhalt: IP-Adresse 192.168.25.1 hat MAC-Adresse ES:ES:ES:ES:ES:ES

Bearbeiten Sie dazu die folgenden Fragen.

- a) Wie nennt man die Attacke, die Sie hier beobachten und welcher Host ist der Zielfhost der Attacke? (1 Punkt)
- b) Können Sie anhand dieser Frames irgendeines Relevantes über den Angreifer aussagen? (1 Punkt)
- c) Unter der Annahme, dass die Attacke erfolgreich war, wie sieht der ARP-Cache in Ihrem eigenen Rechner aus? (1 Punkt)
- d) Das Ziel des Angreifers ist es, dass Ihre Anfragen an den oben angegebenen DNS-Server sowie dessen Antworten an Sie über seinen Host geleitet werden. Genügen dazu die in der Einleitung der Aufgabe beobachteten ARP-Reply Frames oder sind noch weitere notwendig? Falls nein, so begründen Sie dies. Falls ja, so geben Sie an, welche(n) ARP-Reply Frame(s) der Angreifer zusätzlich versenden muss, damit er die gewünschten Pakete sieht. (2 Punkte)
- e) Die Attacke heisst ARP-cache poisoning. Ziel ist der Rechner mit MAC-Adresse C3:..C3, also meinet:
- f) Der Angreifer befindet sich im selben LAN, somit könnte das Paket in nicht entstehen. (Kann nichts zum Rechner gesendet werden sein, weil der andere MAC-Adresse hat.)
- g) Der ARP-cache enthält unter der IP 192.168.25.1 die MAC-Adresse ES:..ES
- h) Da der Zielrechner 192.168.25.1 ausspricht den lokalen Hostes ist, wird ein IP-Paket mit dieser Zieladresse die MAC-Adresse des Gateways tragen. Ist durch eine ARP-Cache-Poisoning-Attacke der Verkehr zum Gateway bereits zum Angreifer umgeleitet, steht der Angreifer so bereits die DNS-Anfragen. Es sind also keine weiteren ARP-Replys nötig.

Cracking Attacke

Das Challenge Handshake Authentication Protocol (CHAP) ist ein Protokoll zur Passwort-basierten Authentisierung

eines Clients bei einem Server, das wir im Unterricht nicht besprochen haben. Es funktioniert wie folgt:

- Der Server sendet dem Client eine Challenge.
- Der Client fügt sein Passwort und die Challenge zusammen und berechnet davon den Hash (z.B. mit SHA2-256), der als Response client und zum Server gesendet wird.
- Der Server kennt das Passwort des Clients und prüft, ob die Response korrekt ist.

a) Dieses Protokoll ist veranlassend auf eine Offline Password Cracking Attacke. Erklären Sie kurz und prägnant, was eine Offline Password Cracking Attacke grundsätzlich ist. (1 Punkt)

b) Nehmen Sie an, Sie haben nun eine Liste von einer Milliarde wahrscheinlicher Passwörter. Wo würden Sie eine Offline Password Cracking Attacke in diesem Fall (CHAP) nun konkret durchführen, um alle Passwörter zu testen? Beschreiben Sie dabei alle Schritte, also auch vorbereitende Schritte bevor Sie die Passwörter selbst testen können. (2 Punkte)

c) Bei einer Offline Password Cracking Attacke ist der Angreifer im Besitz einer Liste von Passwort-Hashes. Diese Hashes kann er dann benutzt lokal (oder in einer Cloud) brechen, ohne dass er dazu das eigentliche Zielsystem (wo die Passwörter für das Login überprüft werden) benötigt.

- d)
1. Der Angreifer macht eine Man-In-The-Middle Attacke (beispielsweise via ARP-Spoofing, falls er im selben Netz sitzt)
 2. Der Angreifer sammelt alle Challenges und die dazugehörigen Responses
 3. Der Angreifer geht durch die Liste der wahrscheinlicher Passwörter, kombiniert je ein Passwort der Liste mit der abgeleiteten Challenge und berechnet daraus den Hash
 4. Der Angreifer überprüft, ob der berechnete Hash dem abgelegenen Hash entspricht. Falls ja, hat er das dazugehörige Passwort gefunden und kann sich damit einloggen.

e) Der:

Sie kontaktieren die URL <https://www.webshop.de/> und erhalten als Teil des Verbindungsaufbaus mittels TLS ein X.509 Zertifikat vom anwerbenden Server. Um festzustellen, ob der anwerbende Server ein Server des Webshops ist, müssen verschiedene Punkte geprüft werden.

Hinweis: Öffnen Sie die o.a. URL nicht, dies ist zur Bearbeitung der Aufgabe nicht erforderlich.

- a) Eine dieser Prüfungen ist das Prüfen der Signatur im Zertifikat. Beschreiben Sie kurz und prägnant die einzelnen Schritte, wie der Browser hier vorgeht. Schreiben Sie also z.B. nicht „Der Browser öffnet die Flasche“ sondern „Der Browser öffnet die Flasche mit einem Flaschenöffner aus dem Supermarkt“. (2 Punkte)
- b) Nehmen Sie an, das Format des Zertifikats sei korrekt und die Signatur wurde erfolgreich überprüft. Nennen Sie zwei weitere Punkte, die ein Browser zur Überprüfung der Gültigkeit des Zertifikats prüfen muss. (1 Punkt)
- c) Nehmen Sie nun an, das Zertifikat sei fertig überprüft und für gültig befunden worden. Das alleine reicht für die Sicherstellung der Authentizität des Servers aber noch nicht aus. Es könnte ja irgendjemand das vom Server ausgelieferte Zertifikat speichern und dann selber ausliefern. Damit dies erbracht wird, muss der Client prüfen, ob der Server den zum Public Key gehörenden Private Key kennt. Geben Sie an, wie der Client das grundsätzlich tun könnte, unabhängig des TLS Protokolls. (1 Punkt)

Sollten in Zukunft leistungsfähige Quantencomputer zur Verfügung stehen, können die bestehenden Public Key Algorithmen nach wie vor als sicher betrachtet werden.: Falsch
RSA kann sowohl für die Verschlüsselung als auch für die Signatur von Nachrichten verwendet werden.: Richtig
Nach einem Schlüsselaustausch mit dem Diffie-Hellman Key Exchange Protokoll können die Kommunikationspartner sicher sein, dass wirklich sie miteinander sprechen.: Falsch
Werden bei der Public Key Kryptographie Algorithmen mit Elliptischen Kurven verwendet, muss die Schlüssellänge gegenüber klassischen Algorithmen verlängert werden.: Falsch

Der Workfaktor von einem Passwort ist meist kleiner als die Anzahl bits vermuten lassen.: Richtig
Der Work Faktor für einen Schlüssel der Länge n ist maximal, wenn alle möglichen Schlüssel gleichverteilt sind.: Richtig
Ein Work Faktor von 64-bit wird aktuell als ausreichend eingestuft.: Falsch
Der Work Faktor hängt sowohl von der Schlüssellänge, als auch vom verwendeten Algorithmus ab.: Richtig

Nehmen Sie an, die Mitarbeiter einer Firma müssen einmalig die MAC-Adresse ihres Laptops (nach Eingabe von Firmen-Benutzername und Passwort) registrieren, damit sie Zugang zum Netz (bzw. eine IP-Adresse vom DHCP-Server) erhalten. Ein anderes LAN-Zugangsschutzverfahren ist der in der Vorlesung besprochene Standard IEEE 802.1x (Port-based Network Access Control).

Vergleichen Sie die beiden Verfahren miteinander, indem Sie 3 unterschiedliche Vorteile des einen im Vergleich zum anderen Verfahren angeben (also z.B. 2 Vorteile des IEEE-Verfahrens und 1 Vorteil des Verfahrens der Firma). Beachten Sie, dass eine plausible Aussage wie z.B. „Das Verfahren der Firma ist sicherer als IEEE 802.1x“ ohne Begründung wertlos ist. (Je 1 Punkt)

1. IEEE 802.1x kann mit den verschiedensten Authentifikationsverfahren verwendet werden. Sollte sich eines als unsicher herausstellen, kann man auf ein anderes wechseln.
 2. Das Verfahren der Firma ist einfacher als IEEE 802.1x und funktioniert auch, wenn die Endgeräte kein 802.1x können.
 3. Beim Wechsel des Firmenlogos muss keine neue Registrierung erfolgen, da bei 802.1x besser.
- Anderer Antworten möglich
- Korrigieren Sie die fünf Fehler in der nachfolgenden Aussage. Ein Fehler, der dabei mehrmals identisch vorkommt (es wird z.B. immer Begriff A statt B verwendet) gilt dabei gesamthaft nur als ein Fehler.

a) (3 Punkte)

Aufgrund des zu kurzen Schlüssels beim WEP Protokoll sowie der Verwendung eines Block Ciphers XOR mit dem Output von RC4 als Pseudo-Random Number Generator werden nur 2^8 unterschiedliche Cipherkeys erzeugt. Dadurch muss ein Angreifer nur abwarten, bis ein bereits verwendeter Keystream erneut verwendet wird. Dazu sammelt er alle übertragenen Pakete und regleicht, ob der im Klartext übermittelte Keystream in einem neu aufgeschriebenen Paket bereits früher in einem Paket übertragen wurde. Ist dies der Fall, kann er aufgrund des folgenden Zusammenhangs:

- $$C_i = K \oplus R \quad P_i \quad \text{und} \quad C_j = K \oplus R \quad P_j \quad \Leftrightarrow \quad K = P_i \oplus R \quad P_i$$
- die Plaintexts P_i und P_j rekonstruieren.
- b) (1 Punkt)
- Sowohl WPA wie auch WPA2 (IEEE 802.11i) sind für den Business-Betrieb ausgelegt und basieren auf EAP und einem RADIUS Server. Damit sind diese Modis ebenfalls ungeeignet für den Heimbetrieb, da dort kein ein RADIUS Server vorhanden ist. Stimmt diese Aussage? Begründen Sie die Antwort.

- a) Es werden 2^{24} verschiedene KEYSTREAMS erzeugt, weil der IV eben nur 24 bit hat (2 Fehler, Keystream und IV). Der Keystream wird nicht im Klartext übermittelt. Es ist $C_1 \oplus C_2 = P_1 \oplus P_2$, nicht $K = \dots$
- b) Weiter WPA noch WPA2 erfordern EAP, obwohl es es beide unterstützen

Wie das IPsec Paket mit Sequenznummer 4 nach dem Paket mit Sequenznummer 5 empfangen, so muss der Empfänger das Paket mit Sequenznummer 4 auf jeden Fall verworfen (potentielle Replay-Attacke). Falsch
Ein Nachteil von IPsec ist, dass es auf dem unzuverlässigen Internet Protokoll (IP) aufbaut. Als Folge muss die IPsec-Kommunikationsbeziehung zwischen den IPsec-Endpunkten immer dann neu aufgestellt werden, wenn entsprechende IP-Pakete verloren gehen. Falsch

IPsec kann den Kommunikationskanal sowohl zwischen Endsystemen als auch zwischen VPN-Gateway sichern. Wahr
Damit Sie mit einem Webbrowser über einen geschützten IPsec-Kanal auf einen Webserver zugreifen können, muss das vom Browser explizit unterstützt werden. Falsch
Mit einer Zero Trust Architektur kann ich meine Anwendungen auch vor Angreifer schützen, welche sich bereits in meinem Netz befinden. Richtig

- Ein guter Firewall-Rule-Manager hat einen Prozess besteht aus den folgenden Schritten:
1. Antrag stellen
 2. Antrag genehmigen
 3. Antrag umsetzen und testen
 4. Betrieb überwachen

f) Falsch
Unternehmen können den geschützten Netzwerkverkehr, welcher von ihren Systemen ausgeht oder auf ihnen terminiert, aufbrechen um ihn so besser schützen zu können. Richtig

Eine Packet Filtering Firewall ist ausrechenend um einen Webserver zu schützen. Falsch
OpenVPN verwendet sowohl für den Handshake als auch die nachfolgende Datenübertragung das TLS-Protokoll. Falsch
Ein Vorteil von OpenVPN im Vergleich zu IPsec ist, dass OpenVPN als signifikant sicherer betrachtet wird. Falsch
IPsec läuft im Kernel Space, OpenVPN im User Space. Wahr
Ein Vorteil von IPsec im Vergleich zu OpenVPN ist, dass diese nicht nur TCP-, sondern auch UDP-basierte Applikationen getunnelt werden können. Falsch

Die richtige Antwort lautet:
Sie sind ein Angreifer und haben die Kontrolle über ein kleines Bot-Netz. Sie möchten mit Hilfe eines DDOS-Angriffes etwas Geld verdienen und möchten mit möglichst wenig Aufwand das Opfer dazu bringen, dass es keine neuen legitimen Verbindungen mehr zulassen kann. Daher werden Sie einen (DDoS-Flood)-Angriff.

Leider war Ihr Opfer gut auf diesen Angriff vorbereitet und der Angriff hat nicht funktioniert. Sie überlegen sich also, dem Opfer möglichst viel Netzwerkverkehr zu senden, damit sämtliche Netzwerkverbindungen überlastet werden. Dazu wählen Sie (DNS-Angriffen) als Angriffsmethode.

Die Social Login Funktionen von Anbietern wie Facebook basiert auf Karberos. Falsch
Karberos gilt heute als veraltetes Protokoll und sollte nicht mehr verwendet werden. Falsch
Das Prinzip der indirekten Authentisierung basiert darauf, dass ein Service die Authentisierung des Benutzers an ein dediziertes System auslagert. Richtig
Karberos kann für die Implementation von Single Sign On verwendet werden. Richtig

a) Der Browser ermittelt den Aussteller des Zertifikats (issuer Name) und überprüft dessen Zertifikat, entweder eine weitere in der Zertifikatskette oder unter den installierten root-Zertifikaten. Dieses Zertifikat entnommt der Browser a) den public key des Ausstellers und b) Algorithmen für Hash und Signatur. Damit überprüft er nun die Signatur im Webseiteninhalt.

b) Der Browser muss prüfen, ob das Zertifikat wirklich gültig ist (hoffentlich, korrekter) und ob es nicht zurückgenommen wurde (revocated).

c) Der Client könnte dem Server Daten schicken, die diese akzeptieren muss. Die Signatur würde dann mit dem jetzt authentisierten öffentlichen Schlüssel geprüft und der Server hätte das Beweise das zugewordnen privaten Schlüssels nachgewiesen.

a) Auf einem neueren iPhone können Sie den Screen mittels PIN oder biometrisch mit dem Finger entsperren. Ist das eine Single-Factor oder Two-Factor Authentisierung? Begründen Sie Ihre Antwort. (1 Punkt)

b) In der Vorlesung haben wir besprochen, dass Karbenes aus dem Passwort eines Benutzers mit einer Hashfunktion den Key berechnet. In einem Buch lesen Sie dazu folgendes: Das Reihewerte dabei ist, dass damit die Stärke des Passworts deutlich verbessert wird. Hat das Passwort z.B. einen Wort Factor von 64 Bits und wir verwenden SHA-1 als Hashfunktion, dann erhalten wir einen Schlüssel mit einer Entropie von 160 Bits, weil der Hash bei SHA-1 eine Länge von 160 Bits hat. Ist diese Aussage richtig oder falsch? Begründen Sie Ihre Antwort möglichst genau. (2 Punkte)

c) Eve hat Alice überredet, ihr eine Kopie des Ticket Granting Tickets (TGT) zu übergeben, welches Alice heute Morgen beim Karbenes Authentication Server gelöst hat. Eine Analyse zeigt, dass dies nicht ausreicht, um im Namen von Alice ein gültiges Ticket für die Nutzung eines Services vom Ticket Granting Server (TGS) zu erhalten. Was müsste Eve noch zusätzlich von Alice bekommen? Begründen Sie Ihre Antwort. (2 Punkte)

a) Das ist single factor, weil nur einer der drei Faktoren (was man weiss, was man hat, was man ist) zur Authentifikation verwendet wird.

b) Das ist falsch. Um ein gefälschtes Passwort zu haben, ist man einfach das Passwort, macht das gefälschte Passwort und vergleicht mit dem gegebenen Hash. Das braucht auch nur 64 bit work factor.

c) Mindestens mit dem Session Key für die Verbindung mit dem TGS. Sonst kann sie den Session Key für den Service nicht entschlüsseln.

In ihrer Firma wird eine neue Anwendung für das Recruiting von neuen Mitarbeitenden eingeführt.

Die Anwendung unterstützt folgende Funktionen:

- Bewerber können ihre Unterlagen hochladen und den Status ihrer Bewerbung sehen, sowie nachträglich Dokumente hochladen und mit den HR Nachrichten austauschen
- Linienvorgesetzte können Bewerbungsunterlagen auf ihre Stellenausschreibungen einsehen und Bewerbungen bewerten.
- HR kann die Bewerbungsunterlagen und Bewertungen für alle Stellenausschreibungen einsehen sowie Nachrichten (Anfragen/Zusätze) an die Bewerber versenden.

a) Definieren Sie die für diese Anwendung sinnvollen Rollen (1 Punkt)

b) Definieren Sie die für diese Anwendung sinnvollen Ressourcen (1 Punkt)

c) Definieren Sie, wie welche Rechte die identifizierten Rollen auf die identifizierten Ressourcen haben sollen (1 Punkt)

a) Bewerber, Linienvorgesetzte, HR

b) Bewerbungsunterlagen, Bewerbungen, Kommunikation mit Bewerber

c) Bewerber:

	Bewerbungsunterlagen	Bewerbungen	Kommunikation
Bewerber	schreiben und lesen	keine	schreiben und lesen
Linienvorgesetzte	lesen	schreiben und lesen	keine
HR	lesen	lesen	schreiben und lesen

a) Sowohl in Linux als auch Windows gibt es quasi „administrative“ User - die Administratoren. (2 Punkte)

(1) Begründen Sie, wieso deren Existenz eigentlich dem DAC Prinzip widerspricht.

(2) Begründen Sie, wieso deren Existenz aus Sicht des Arbeitgebers notwendig ist. Geben Sie dazu ein konkretes Fallbeispiel an, das für die Firma bei „reinem“ DAC System ohne Administratoren problematisch wäre.

ii) Im Gegensatz zu Linux Systemen wie z.B. Ubuntu sind Administratoren in Windows nicht von DAC Mechanismen ausgeschlossen. D.h. der Besitzer einer Datei kann z.B. dem Administrator per ACL Eintrag den Zugriff auf eine Datei verweigern. Der betroffene Administrator kann dies bei Bedarf umgehen, indem er die Datei in seinen Besitz nimmt und anschließend den betreffenden ACL Eintrag entfernt.

Ein Kollege meint zu diesem Punkt: „Schließlich können die Admins ja in beiden Welten (Vom der Redaktion: Linux und Windows) an alle Daten ran. Es kommt aus praktischer Sicht also nicht darauf an, was ich einsetze“.

Sie wissen es besser und liefern folgendes Argument für den Windows-Ansatz... (1 Punkt)

c) Ein Verzeichnis mit dem Namen important_files auf einem Linux-System wird bei der Eingabe von ls -l wie folgt dargestellt:

```
drwxr-xr-x 2 alice rwa 4096 Jan 19 12:19 important_files
```

Beschreiben Sie möglichst genau die Rechte der Benutzer alice, bob (ist in der Gruppe rwa) und carol (ist nicht in der Gruppe rwa) bezüglich des Verzeichnisses important_files. Geben Sie insbesondere präzise an, werauf sich die Berechtigungen aus, jeweils genau bezeichnen.

alice:

bob:

carol:

a) 1. Der Administrator kann Rechte vergeben oder entziehen, was eigentlich nur dem Eigentümer der entsprechenden Ressource zusteht. 2. Ein Mitarbeiter erstellt eine wichtige Datei, die nur er lesen/modifizieren darf und verlässt. Wenn sich niemand als diese Person einloggen kann und es kein Administrator gibt, kann diese Datei niemals mehr modifiziert werden.

b) Haben wir in der Vorlesung nicht besprochen, steht in der Folie zu der Folie: Unter Windows wird die Tätigkeit des Administrators protokolliert, unter Linux nicht. Wenn also ein Administrator unter Windows auch Zugang zu einer Datei verschafft, hinterlässt das eine Spur, in Linux nicht unbedingt.

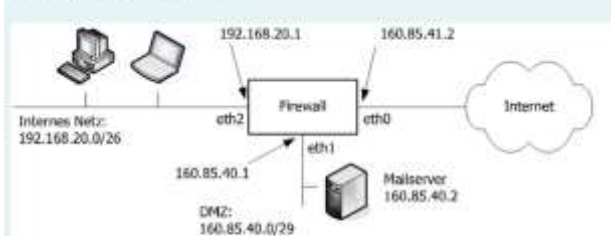
c) i)

alice: Darf das Verzeichnis lesen und modifizieren (Einträge ändern, hinzufügen, löschen), darf ins Verzeichnis wechseln

bob: Darf Verzeichniseinträge lesen und löschen wechseln

carol: Darf nur ins Verzeichnis wechseln

Die sind die Administrator des folgenden Netzes



Die Security Policy besagt, dass die Firewall wie folgt arbeiten soll:

- Die Firewall arbeitet prinzipiell statisch. Ausserdem soll alles blockiert werden, was durch die folgenden Punkte nicht explizit erlaubt wird. Wird ein Paket blockiert, so wird es stillschweigend vorgeworfen, ohne dem Absender zu benachrichtigen.
- Sowohl aus dem internen Netz als auch aus dem Internet können per SMTP-Protokoll (TCP Port 25) E-Mail Nachrichten zum Mailserver geschickt werden.
- Der Mailserver kann per SMTP-Protokoll E-Mail Nachrichten an andere SMTP-Server im Internet senden.
- Die Benutzer im internen Netz können mit dem IMAPS (IMAP over SSL, TCP Port 993) auf ihre Mailbox auf dem Mailserver zugreifen.
- Vom internen Netz darf auf identische Shares im Internet zugegriffen werden, egal welches Protokoll. Die einzige Ausnahme ist der Default-Port von „World of Warcraft“-Servern im Internet (TCP Port 3734).
- Die Firewall soll als NAT-Box (Source NAT) für das interne Netz agieren und die Source IP-Adresse in Paketen vom internen Netz ins Internet (und umgekehrt) durch die externe IP-Adresse der Firewall (160.85.41.2) ersetzen.

a) Nehmen Sie an, es sei alles korrekt gemäß der Security Policy konfiguriert worden. Sie führen nun von einem Host im Internet mit IP-Adresse 69.254.173.90 einen nmap-Scan durch:

```
nmap -PS -p21-65535 160.85.41.2
```

Welchen Output erwarten Sie? Geben Sie die „open“, „closed“ und „filtered“ Ports an. (2 Punkte)

a) open sollte sein: 25 (SMTP), rest filtered.

Datenschutz

Das Schweizer Datenschutzgesetz schützt natürliche Personen, aber nicht für juristische Personen.: Richtig

Neben der Verarbeitung von Personendaten durch Unternehmen unterliegt auch die Bearbeitung von Personendaten für den privaten Gebrauch dem Datenschutzgesetz.: Falsch

Das Datenschutzgesetz gilt nur, wenn die Daten in der Schweiz bearbeitet werden.: Falsch

Bei einer Verletzung der Datensicherheit muss der EDÖG nur dann informiert werden, wenn ein hohes Risiko für die Persönlichkeit der betroffenen Person vorliegt.: Richtig