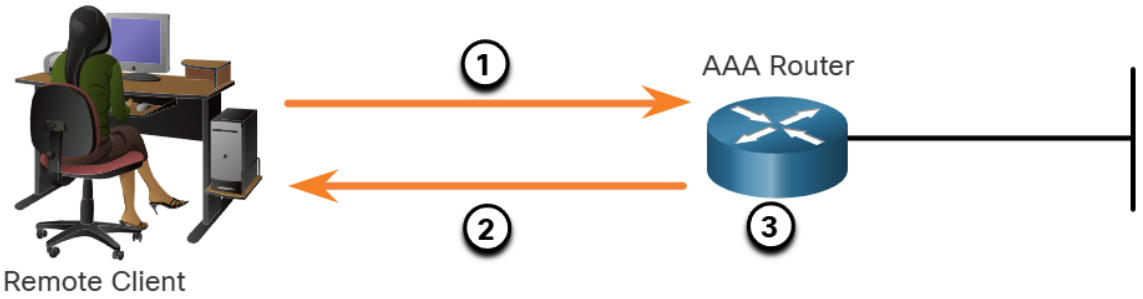
Inhaltsverzeichnis

AAA: Authentication, Authorization, Accounting (7.1.2)

- Zugangskontrolle auf Netzwerkgeräten
  - **Authentication**: Legt fest wer zugreifen darf
  - **Authorization**: Legt fest was gemacht werden darf
  - **Accounting**: Loggt was gemacht wird
- Beispiel Kreditkarte:
  - Authentication: Prüfziffer und Kartennummer
  - Authorization: Kartenlimit
  - Accounting: Kreditkartenabrechnung

Authentifizierungsmodi (7.1.3)

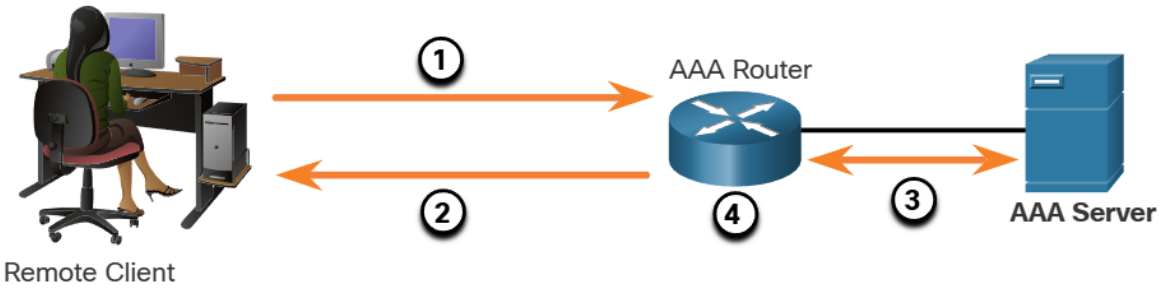
Local AAA Authentication



1. The client establishes a connection with the router.
2. The AAA router prompts the user for a username and password.
3. The router authenticates the username and password using the local database and the user is provided access to the network based on information in the local database.

- Nutzt lokale Datenbank auf Gerät
- Datenbank enthält Nutzernamen und Passwörter
- Nutzer authentifizieren sich gegen die lokale Datenbank
- Verwendung in kleinen Netzwerken

Server-Based AAA Authentication (7.3)



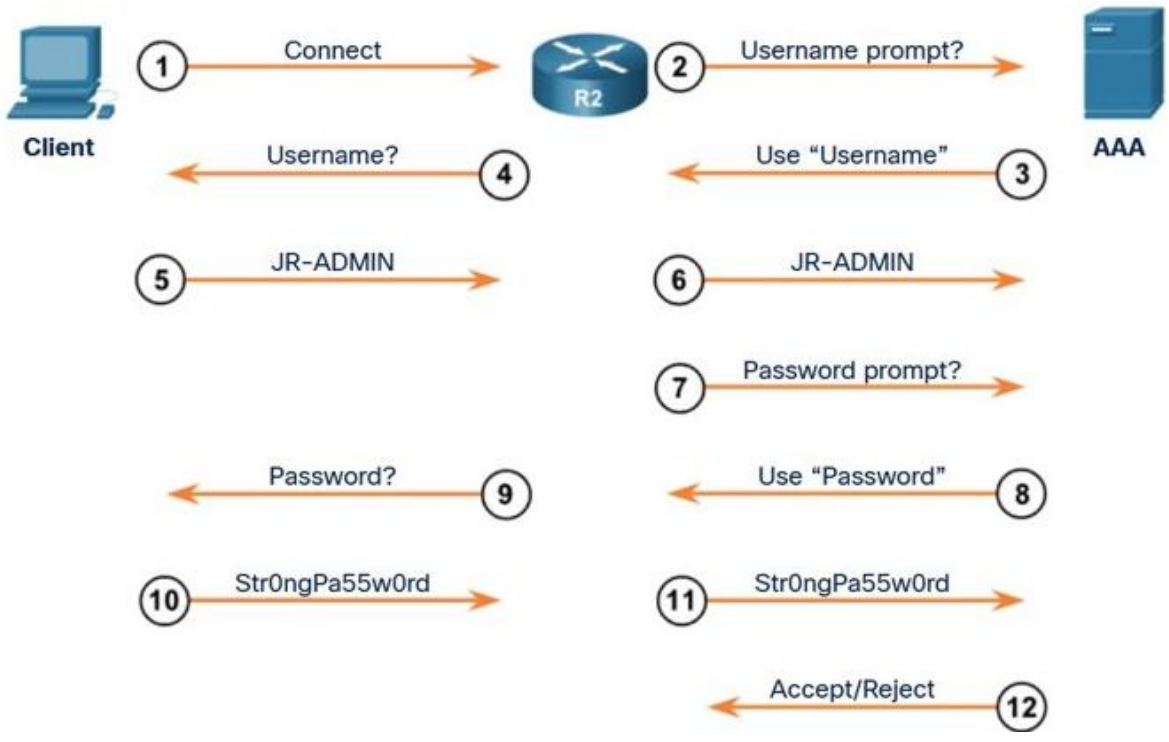
1. The client establishes a connection with the router.
2. The AAA router prompts the user for a username and password.
3. The router authenticates the username and password using a AAA server.
4. The user is provided access to the network based on information on the remote AAA server.

- Router greift auf AAA-Server zu (RADIUS / TACACS+)
- Server enthält Nutzernamen und Passwörter
- Nützlich für große Netzwerke, da Nutzer an einer zentralen Stelle verwaltet werden können

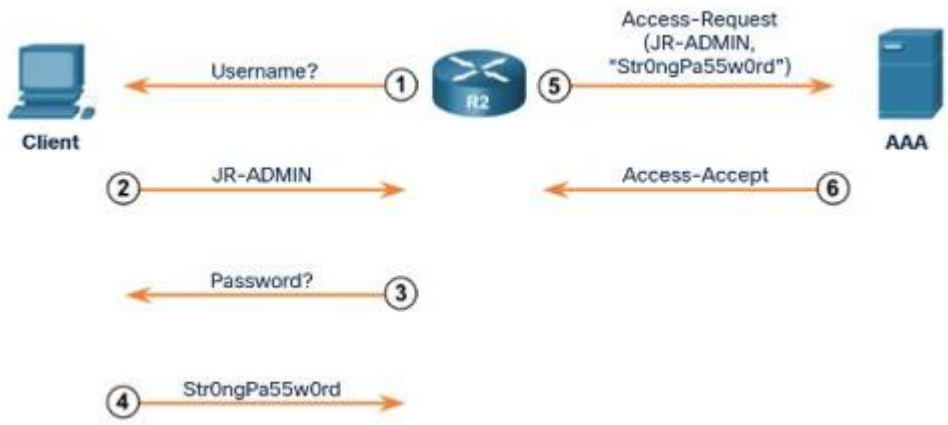
Authentifizierungsprotokolle (7.3.3)

	TACACS+	**RADIUS**
Funktionalität	- Separiert AAA - Erlaubt Modularität (Unterschiedliche Dienste für Authentication & Authorization)	- Kombiniert Authentication und Authorization - Weniger flexibel
Standard	- Cisco	- Offen/RFC
Transport	- TCP	- UDP
Challenge Handshake Authentication Protocol	- Bidirektionale Challenge und Antwort	- Unidirektionale Challenge und Antwort vom Server zum Client
Vertraulichkeit	- Jedes Paket verschlüsselt	- Nur Passwort wird verschlüsselt
Anpassbarkeit	- Befehle auf per-user / per-group Ebene	- Keine Authorization auf per-user / per-group Ebene
Accounting	- Begrenzt	- Ausführlich
Verwendung	- z.B. Organisation mit vielen Gruppen	- z.B. ISP für ausführliches Accounting

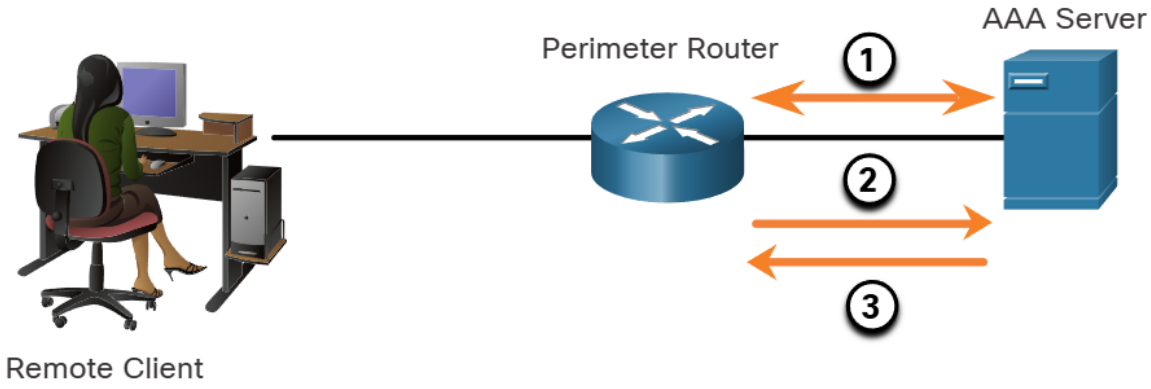
TACACS+ Authentication (7.3.4)



RADIUS Authentication (7.3.5)



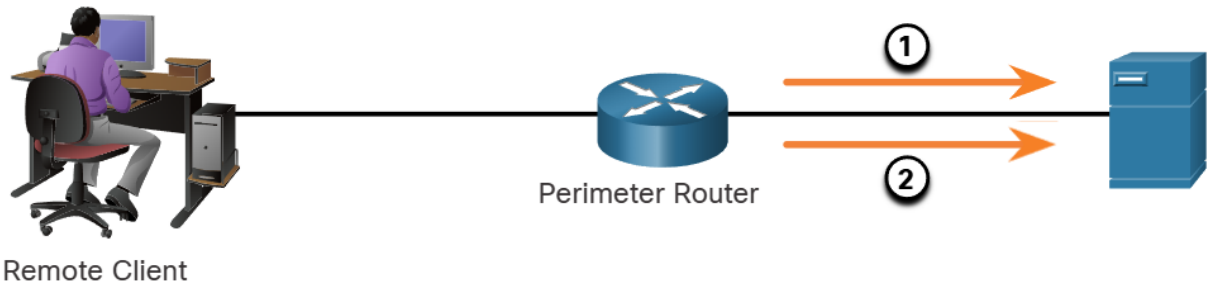
Authorization (7.1.4)



1. When a user has been authenticated, a session is established between the router and the server.
2. The router requests authorization from the AAA server for the client's requested service.
3. The AAA server returns a PASS/FAIL for authorization.

- Automatischer Schritt nach der Authentifizierung
- Regelt was Nutzer dürfen oder nicht dürfen
- Rechte werden vom Router am AAA-Server abgeglichen
- Rechteset würde vom Server an den Router übermittelt

Accounting (7.1.5)

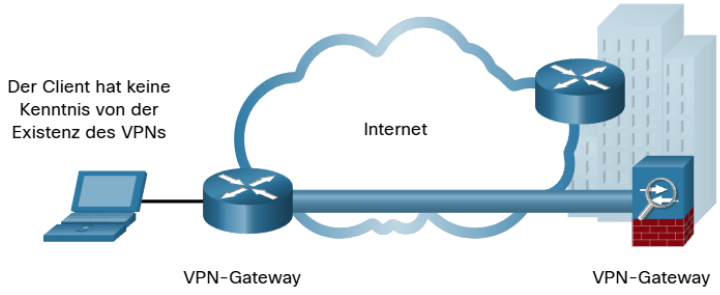


1. When a user has been authenticated, the AAA accounting process generates a start message to begin the accounting process.
2. When the user finishes, a stop message is recorded and the accounting process ends.

- Sammelt und meldet Datennutzung
- Verwendung für
  - Abrechnung
  - Audits
  - Troubleshooting
- Enthält
  - Start- & Endzeiten einer Verbindung
  - Ausgeführte Befehle
  - Paket- & Byteanzahl
- Netzwerkgerät meldet an Server

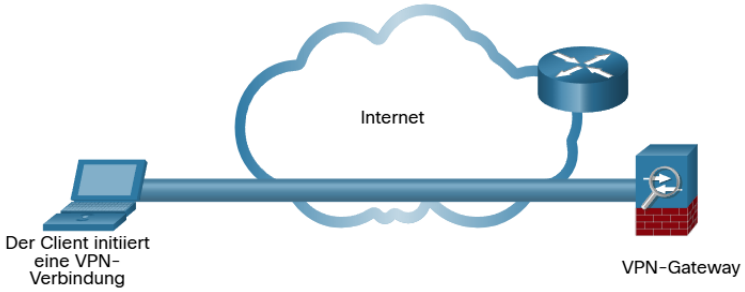
VPN: Virtual Private Network (8.1)

Site-to-Site VPN



- Zwei VPN-Gateways bauen einen Tunnel zueinander auf
- Datenverkehr wird nur zwischen den beiden Gateways verschlüsselt
- Meist durch IPsec abgesichert
- Hosts haben keine Kenntnis von der Existenz des VPNs

Remote-Access VPN (8.2.1)



- Wird bei Bedarf dynamisch aufgebaut
- Tunnel besteht zwischen Client und VPN-Gateway
- Z.B. Remoteeinwahl in die Firma

SSL-VPN (8.2.2)

- Client stellt Verbindung mit VPN-Gateway mittels SSL/TLS her
- Setzt PKI und Zertifikate voraus
- Sinnvoll für einfache Bereitstellung und breiten Support (Nicht so sicher wie IPsec)
- IPsec und SSL-VPN schließen sich **nicht** gegenseitig aus, sondern ergänzen sich

Funktion	IPsec	SSL
Anwendungsunterstützung	**Umfangreich** - Alle IP-basierten Anwendungen werden unterstützt.	**Limitiert** - Nur webbasierte Anwendungen und Dateifreigaben werden unterstützt.
Stärke der Authentifizierung	**Strong** - Verwendund einer Zwei-Wege-Authentifizierung mit gemeinsamen Schlüsseln oder digitalen Zertifikaten.	**Moderat** - Verwendung einer unidirektionalen oder bidirektionalen Authentifizierung.
Verschlüsselungsstärke	**Strong** — Schlüssellängen von 56 Bit bis 256 Bit.	**Moderat bis strong** - Schlüssellängen von 40 bis 256 Bit.
Komplexität der Verbindung	**Medium** - Erfordert einen auf einem Host vorinstallierten Client.	**Low** - Es erfordert nur einen Webbrowser auf einem Host.
Verbindungsoption	**Limited** - Nur bestimmte Geräte mit bestimmten Konfigurationen können eine Verbindung herstellen.	**Extensive** — Jedes Gerät mit einem Webbrowser kann eine Verbindung herstellen.

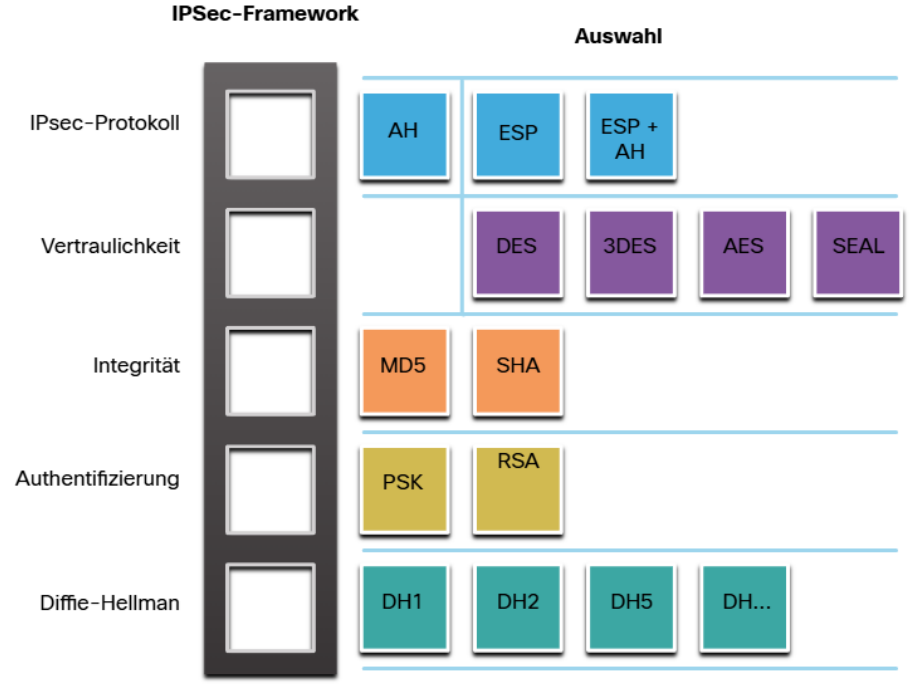
IPsec: IP Security (8.3)

- IETF-Standard, der definiert, wie ein VPN über IP-Netze gesichert werden kann
- Gibt nur ein Framework vor
- Schützt und authentifiziert zwischen Quelle & Ziel
- Kann von Layer 4 – 7 schützen

Sicherheitsfunktionen

- **Vertraulichkeit**: Verschlüsselung verhindert das Lesen des Inhalts durch Dritte
- **Integrität**: Hashing-Algorithmen stellen sicher, dass das Paket auf dem Weg nicht verändert, wurde
- **Authentifizierung des Ursprungs**: Nutzt Internet-Key-Exchange (IKE) um Quelle & Ziel zu authentifizieren (PSK oder Zertifikate)

Framework-Elemente



IPsec-Protokollkapselung

- **AH: Authentication Header**
  - Wenn keine Vertraulichkeit (Verschlüsselung) erforderlich ist
  - Unverschlüsselt
- **ESP: Encapsulation Security Protocol**
  - Verschlüsselt IP-Pakete
  - Authentifiziert Datenherkunft und Integrität

Vertraulichkeit

- Wird durch Verschlüsselung erreicht
- Je besser der Algorithmus und die Schlüssellänge, desto höher der Grad der Vertraulichkeit
- Meist symmetrische Verschlüsselung, da schneller

Integrität

- Stellt sicher, dass gesendete Daten mit empfangenen Daten übereinstimmen
- **HMAC: Hashed Message Authentication Code** – Hashcode des versendeten Pakets
- Wenn Hashcode beim Versenden und Empfangen identisch ist, gilt das Paket als Integer

Authentifizierung

- Gerät am anderen Ende des Tunnels muss authentifiziert werden damit der Tunnel als sicher gilt
- Entweder PSK oder RSA
  - **PSK**: Authentifizierungsschlüssel und ID werden gehashed
    - Schlüssel wurde zuvor manuell an den VPN-Gateways hinterlegt
    - Stimmt erhaltener und berechneter Hash überein ist die Gegenstelle authentifiziert
  - **RSA**: Authentifizierung mittels Zertifikats

Schlüsselaustausch

- Verschlüsselungsalgorithmen benötigen gemeinsamen Schlüssel
- Austausch erfolgt nach Diffie-Hellman
- Je höher die DH-Gruppe, desto sicherer

GRE over IPsec (8.2.4)

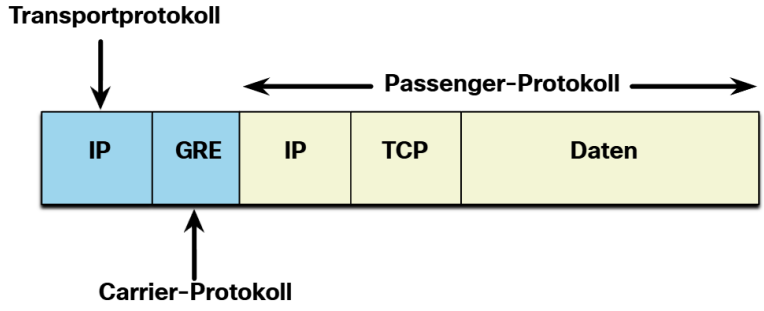
Problem

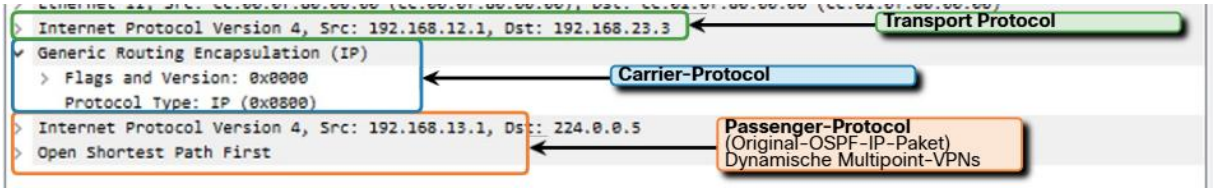
GRE: Generic Route Encapsulation

- Unsicheres Site-to-Site VPN Protokoll
- Keine Verschlüsselung
- Kann verschiedene Protokolle der Vermittlungsschicht (Network Layer [3]) verschlüsseln
- Unterstützt neben Unicast auch Multicast und Broadcast

Lösung

- Mutlicast / Broadcast wird in GRE gekapselt
- GRE-Paket wird in ein IPsec-Paket eingepackt





- **Passenger-Protokoll**: Ursprüngliches Paket – Unabhängig welches Protokoll, und ob Uni- / Mutli- / Broadcast
- **Carrier-Protokoll**: Träger-Protokoll, das das Passenger-Paket einkapselt
- **Transportprotokoll**: Protokoll, das sich um die tatsächliche Weiterleitung des Pakets handelt

Firewall (9)

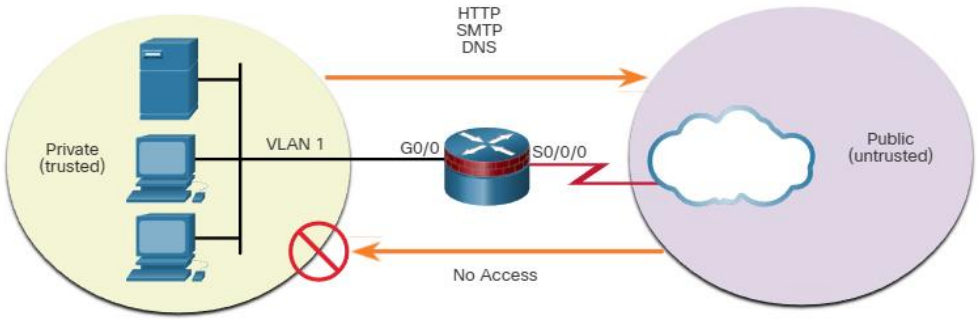
- System, das ein Regelwerk zwischen Netzwerken erzwingt
- Resistent gegen Netzwerkangriffe
- Einziger Transitpunkt zwischen internem und externem Netzwerk
- Verhindert unzulässigen Zugriff auf Systeme
- Blockiert bösartigen Netzwerktraffic
- Kann Netzwerkperformance reduzieren
- Getunnelter Traffic kann nicht verhindert werden

Arten (9.1)

Firewall	Layer	Beschreibung
Packet-Filtering (Stateless)	3 – 4	- Lediglich Analyse der Pakete und Abgleich mit Regelwerk anhand Port, Adressen und Protokoll - Bereits in vielen Routern integriert - Anfällig für IP-Spoofing
Stateful	3 – 5	- Analysiert zusätzlich Verbindungsinformationen - Bessere Performance
Application Gateway	3 – 5 & 7	- Arbeitet als proxy für den Client
Next Generation		- Intrusion-Prevention - Application awareness

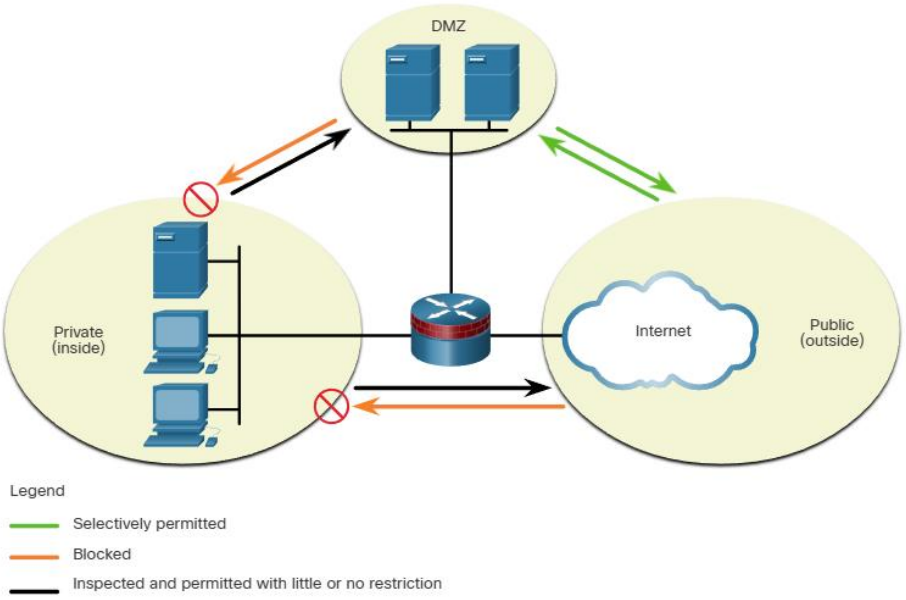
Netzwerkdesign (9.2)

Privat & Öffentlich



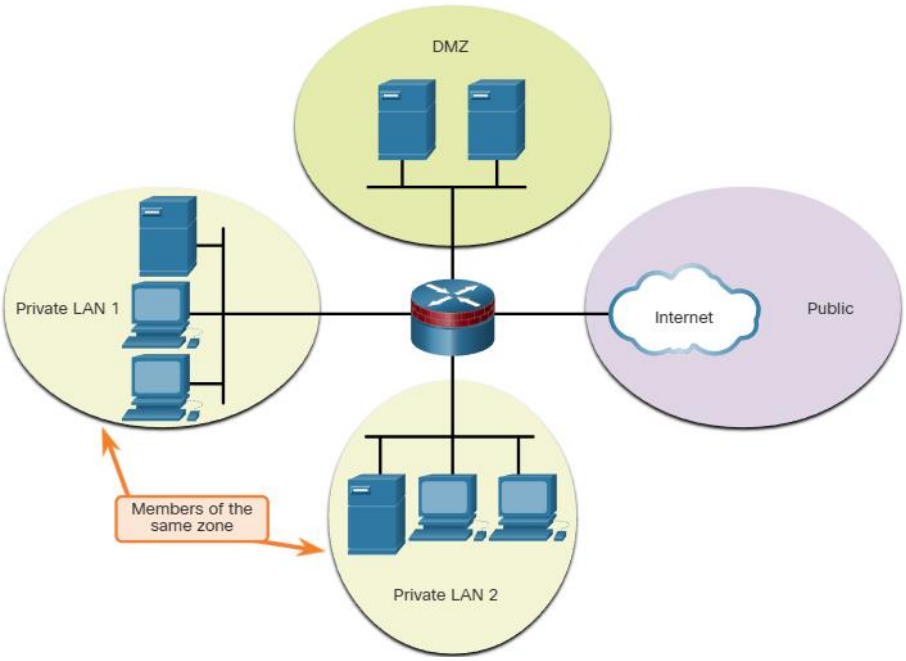
- Ausgehender Verkehr wird zugelassen
- Eingehender Verkehr wird überprüft
    - Wird zugelassen bei Verbindung mit ausgehendem Traffic
    - Wird in allen anderen Szenarien blockiert

DMZ: Demilitarisierte Zone



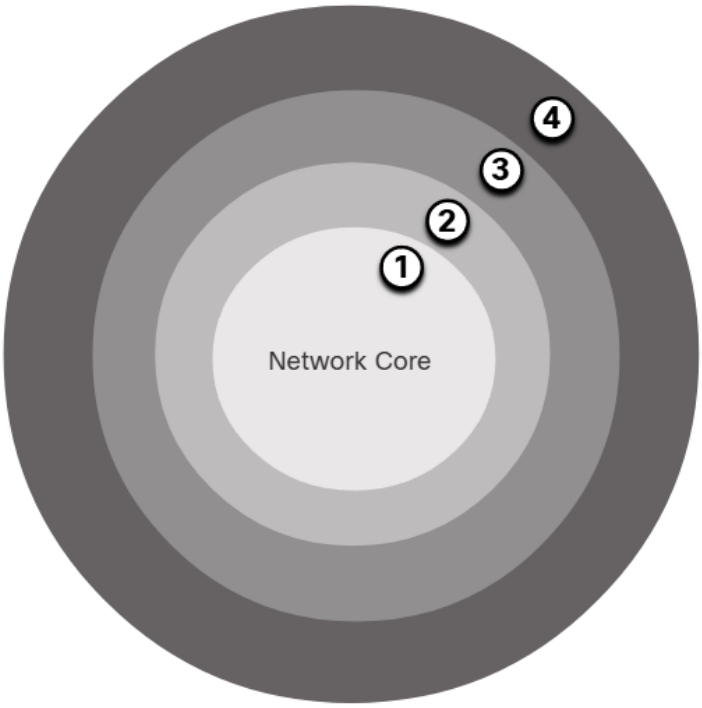
- Ausgehender Traffic Richtung DMZ/Öffentlich weitestgehend zugelassen
- Eingehender Traffic von DMZ wird blockiert
- Ausgehender Traffic von der DMZ Richtung Öffentlich entsprechend der Service-Anforderungen zugelassen
- Eingehender Traffic vom Öffentlichen Richtung DMZ wird untersucht und entsprechend der Service-Anforderung zugelassen
- Eingehender Traffic vom Öffentlichen ins private Netz wird blockiert

Zonen basiert



- Regelwerk wird auf Zonen angewandt
- Kommunikation zwischen Mitglieder derselben Zone wird nicht untersucht und immer erlaubt

Layered Defense



1. ***Network Core security*** *- Protects against malicious software and traffic anomalies, enforces network policies, and ensures survivability*
2. ***Perimeter security*** *- Secures boundaries between zones*
3. ***Communications security*** *- Provides information assurance*
4. ***Endpoint security*** *- Provides identity and device security policy compliance*

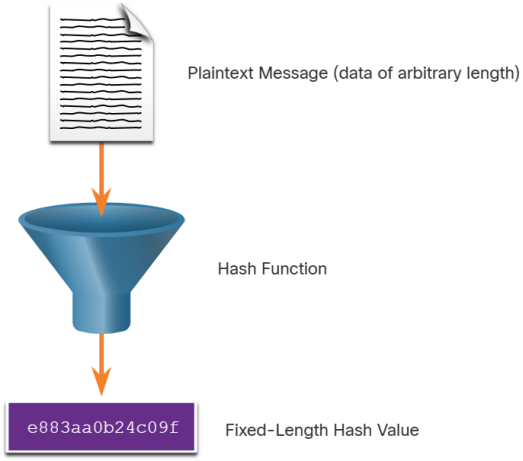
Integrität und Authentizität (16.1)

Sichere Kommunikation

- **Datenintegrität**: Garantiert, dass die Daten auf dem Weg nicht verändert wurden
- **Origin Authentifizierung**: Garantiert, dass der angegebene Absender auch der tatsächliche ist
- **Datenvertraulichkeit**: Garantiert, dass nur autorisierte Nutzer das Paket lesen können
- **Nichtabstreitbarkeit der Daten**: Garantiert, dass der Absender die gesendeten Daten nicht abstreiten kann

Hash-Funktionen

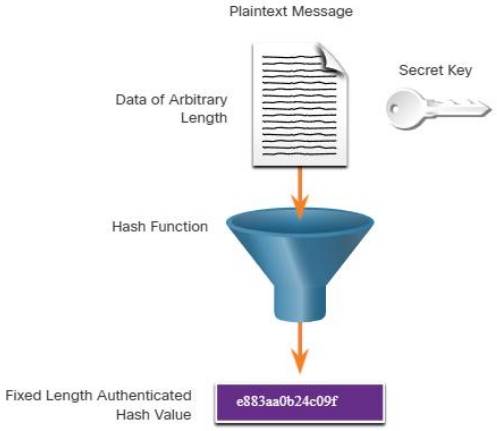
- Verwendet, um Datenintegrität zu gewährleisten
- Input: Variable Daten
- Output: Hash mit immer gleicher Länge
- Wenn die Daten beim Absenden und Empfangen den gleichen Hashwert ergeben, ist die Datenintegrität gewährleistet
- Bekannte Hashfunktionen
  - MD5 (legacy)
  - SHA



- Schützt nicht vor Man-in-the-Middle, da in diesem Fall der Angreifer einfach den mitgesendeten Hashwert, der zum Vergleich beim Empfangen dient, mit abändert.

Origin Authentication (HMAC)

- Um MITM-Attacken zu unterbinden / zu erkennen, wird dem Hashing-Prozess ein PSK hinzugefügt, den nur Absender und Empfänger kennen
    - Das gleiche Hashergebnis kann nur dann entstehen, wenn Daten und Key bei beiden Hashing-Prozessen identisch sind
    - Ein MITM-Angreifer kennt den PSK nicht und kann somit den Hash nicht anpassen
- Datenintegrität UND Authentifizierung des Origins in einem Schritt, da nur der Origin auch den PSK kennt



Digitale Signatur (17.1)

- Nutzt asymmetrische Kryptographie
- Gilt als Beweis, dass Datenaustausch stattgefunden hat
- Eigenschaften einer Signatur
    - **Authentisch**: Signatur kann nicht gefälscht werden und beweist, dass ausschließlich der Absender das Dokument signiert hat
    - **Unveränderlich**: Nach der Signierung kann das Dokument nicht mehr geändert werden
    - **Nicht transferierbar**: Die Signatur kann nicht auf ein anderes Dokument übertragen werden
    - **Nicht abstreitbar**: Gilt als Beweis, dass das Dokument von einer Person signiert wurde
- Verwendung
    - **Code-Signierung**: Für Datenintegrität und Authentifizierung bei ausführbaren Dateien
    - **Digitale Zertifikate**: Authentifiziert die Identität eines Systems und dient zum Aufbaue einer verschlüsselten Kommunikation
- Standards
    - **DSA - Digital Signature Algorithm**: Ursprünglicher Standard zur Generierung von öffentlichen, privaten Schlüsseln und Generierung und Verifizierung von digitalen Signaturen
    - **RSA – Rivest-Shamir-Adelman Algorithm**: Asymmetrischer Algorithmus zur Generierung und Verifizierung von digitalen Signaturen
    - **ECDSA – Elliptic Curve Digital Signature Algorithm**: Neuere effizientere Variante von DSA. Unterstützt Authentifizierung von digitalen Signaturen und Nichtabstreitbarkeit.

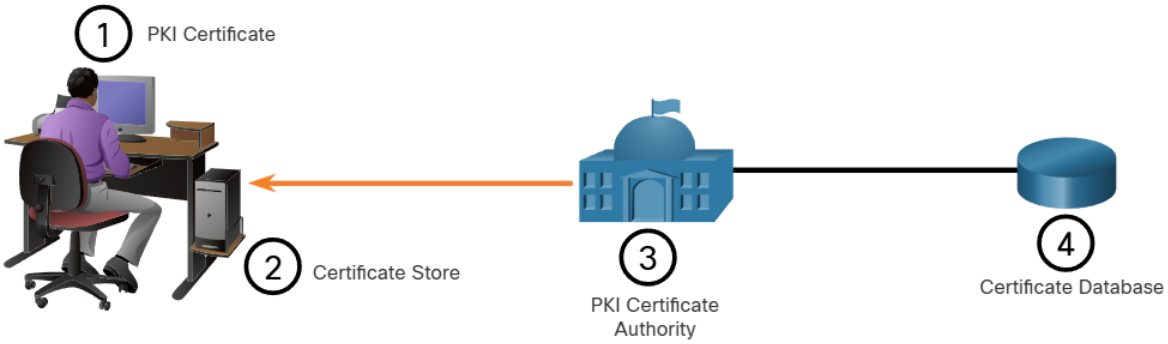
Code-Signierung

- Beweist Authentizität des Codes und, dass er tatsächlich aus der angegebenen Quelle stammt
- Beweist, dass der Code nicht auf dem Weg verändert wurde
- Beweist, dass der Code definitiv vom Entwickler veröffentlicht wurde

Digitales Zertifikat

- Authentifiziert und verifiziert die Echtheit einer Nachricht und dessen Absender
- Beispielablauf
    - Bob bestätigt eine Bestellung auf Alices Website
        - Bob erstellt einen Hash seiner Bestätigung
        - Bob verschlüsselt den Hash mit seinem **privaten Schlüssel** (Digitale Signatur)
            - **Nicht** wie bei der Asymmetrischen Verschlüsselung mit dem öffentlichen Schlüssel
            - Somit kann jeder, der Bobs öffentlichen Schlüssel hat, den Hash entschlüsseln
            - Damit beweist Bob seine Identität, da nur er im Besitz seines privaten Schlüssels sein kann
        - Bob sendet den verschlüsselten Hash mit seiner Bestätigung an Alice
        - Alice empfängt den verschlüsselten Hash und die Bestätigung
        - Alice entschlüsselt den Hash mit Bobs öffentlichen Schlüssel
        - Alice erstellt ebenfalls einen Hash der Bestätigung von Bob (ohne die Signatur von Bob!)
        - Wenn die beiden Hashes übereinstimmen, ist die Authentizität des Dokuments bewiesen
            - Die Bestätigung wurde von Bob verschickt und unterwegs nicht verändert

PKI: Public Key Infrastructure (17.2)



1. *PKI certificates contain an entity's or individual's public key, its purpose, the certificate authority (CA) that validated and issued the certificate, the date range during which the certificate is valid, and the algorithm used to create the signature.*
2. *The certificate store resides on a local computer and stores issued certificates and private keys.*
3. *The PKI Certificate of Authority (CA) is a trusted third party that issues PKI certificates to entities and individuals after verifying their identity. It signs these certificates using its private key.*
4. *The certificate database stores all certificates approved by the CA.*

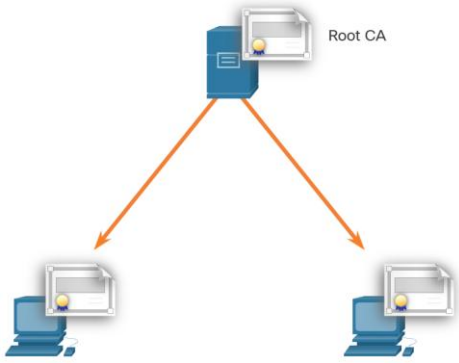
- Internet-Verkehr immer zwischen zwei Parteien
- Beim Herstellen einer Verbindung werden öffentliche Schlüsselinformationen ausgetauscht
- Beispiel Webserver
    - Webserver brauchen ein SSL-Zertifikat
    - Betreiber des Webserver kaufen ein Zertifikat für die Domain des Webservers bei einer Zertifizierungsstelle
    - Die Zertifizierungsstelle prüft die Zertifikatsanfrage und stellt ein Zertifikat aus
    - Ab diesem Zeitpunkt vertrauen automatisch alle Systeme dem ausgestellten Zertifikat, die der Zertifizierungsstelle vertrauen
- **Zertifikatsautorität (CA: Certificate Authority):** Organisation, die digitale Zertifikate erstellt, indem sie einen öffentlichen Schlüssel an eine bestätigte Identität bindet
- PKI ist notwendig, um die Verteilung und Identifizierung von öffentlichen Schlüsseln zu unterstützen
- Hoch skalierbar
- IETF X.509 v3 Standard definiert das Format eines Digitalen Zertifikats

Autoritäts-System

- CAs erstellen Zertifikate basierend auf Klassen, die angeben, wie vertrauenswürdig ein Zertifikat ist
    - **0:** Für Tests; Keine Überprüfung hat stattgefunden
    - **1:** Für Einzelpersonen; Benötigt E-Mail-Bestätigung
    - **2:** Für Organisationen; Identitätsbeweis notwendig
    - **3:** Server und Code-Signierung; Unabhängige Prüfung der Identität durch die CA
    - **4:** Für Transaktionen zwischen Firmen
    - **5:** Für private Organisationen oder Staatliche Sicherheit

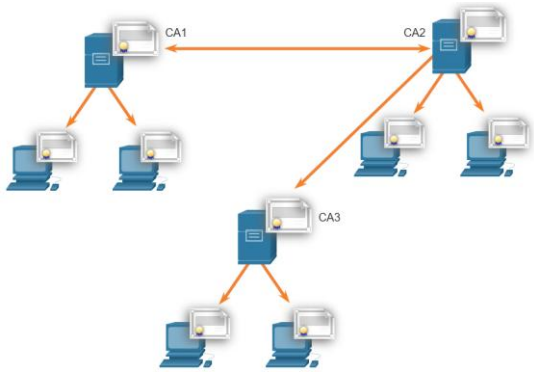
Topologien

Single-Root PKI



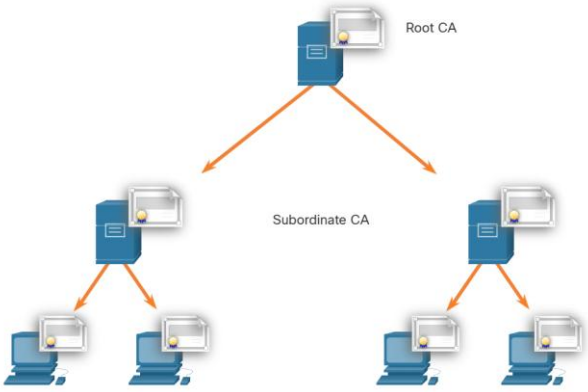
- Simpler Aufbau
- Nur eine einzige CA, die Zertifikate ausstellt
- Single-Point of Failure
- Schwer skalierbar

Cross-Certified CA



- Vertrauensbildung zwischen zwei Cas
- Stellen gegenseitig Zertifikate aus
- Nutzer jeder CA-Domain vertrauen automatisch einander
- Kein Single-Point of Failure mehr

Hierarchical CA



- Eine Root CA stellt Zertifikate für Sub-CAs aus, die Zertifikate für Endnutzer ausstellen
- Gut skalierbar
- Leicht verwaltbar
- Ggf. schwierig die Zertifikatskette zu bestimmen

Verwaltung eines Zertifikats

Ausrollen

- Kopie des öffentlichen Schlüssels der CA erhalten (Self-Signed-Certificate)
    - Verifiziert alle Zertifikate, die von der CA ausgestellt wurden
    - Nur die Root-CA kann ein Self-Signed-Certificate ausstellen.
- Verteilung der CA-Zertifikate meist automatisch

Authentifizierung

- Sobald das Zertifikat von der CA ausgerollt wurde, wird die CA zur Kommunikation zwischen zwei Parteien nicht mehr benötigt

Widerrufen

- Ggf. muss ein Zertifikat widerrufen werden, wenn der Schlüssel kompromittiert ist
    - **CRL – Certificate Revocation List**: Liste mit Seriennummern von widerrufenen Zertifikaten. PKI-Mitglieder holen sich regelmäßig diese Liste
    - **OCSP – Online Certificate Status Protocol**: Genutzt, um einen OCSP-Server über den Status eines Zertifikats abzufragen