**Hash Funktionen**

Hash Funktionen sind unidirektionale, deterministische Funktionen zur Komprimierung von Eingangsdaten beliebiger Länge auf Ausgangsdaten fixer Länge.

**Unidirektional:** nur in eine Richtung, nahezu unmöglich die Originaldaten aus dem Hashwert zurückzurechnen

**Deterministisch:** gleiche Eingabedaten liefern denselben Hashwert

**Kollisionen:** Mehrere, unterschiedliche Eingangsdaten liefern dieselben Ausgangsdaten

**Avalanche-Effekt:** min. Änderung Eingangsdaten = max. Änderung Ausgangsdaten

**Beispiele:**

* MD5 & SHA-1: Veraltet, da kollisionsanfällig
* SHA-2 (SHA-256, SHA-512): Modern und weit verbreitet
* SHA-3: Sicherer als SHA-2, für künftige Anwendungen

**Anwendungsfälle:** Einfacher/Schneller Vergleich von Daten& Prüfwerte

**Verschlüsselung**

Verschlüsselung wandelt lesbare Daten (Plaintext) in eine verschlüsselte Form (Ciphertext) um, die nur mit einem bestimmten Schlüssel wieder lesbar gemacht werden kann.

Es gibt zwei Haupttypen von Verschlüsselung:

**Symmetrisch:**

* Selber Schlüssel für Verschlüsselung und Entschlüsselung
* **Vorteile:** Schnell, einfache Implementierung
* **Nachteile:** Wie Schlüssel austauschen, alle Beteiligten kennen Schlüssel

**Asymmetrisch:**

* Unterschiedliche Schlüssel (private & public key) für Verschlüsselung und Entschlüsselung
* **Vorteile:** Austausch sehr einfach, Authentizität kann gewährleistet werden
* **Nachteile:** Langsam, Komplexe Implementierung

**Kombination:**

* Symmetrische & Asymmetrische Verschlüsselung wird oft kombiniert, um die Vorteile von beiden zu nutzen z.B. bei TLS.

**Phishing**

Phishing ist eine Methode, bei der Angreifer gefälschte E-Mails oder Webseiten erstellen, um an vertrauliche Informationen wie Passwörter oder Bankdaten zu gelangen.

Es lässt sich an verdächtigen URLs, ungewöhnlichen Absendern oder Aufforderungen zur schnellen Aktion erkennen.

**Zwei-/Mehr-Faktor Authentifizierung** (Authentifizierung = Nachweis einer Identität)

* “Something you **know**”: Passwort / PIN
* “Something you **have**”: Smartphone / Hardware Token
* “Something you **are**”: Fingerabdruck / Iris
* “Some**where** you are”: IP Adresse / Standort
* “Some**times** you are”: Zeitfenster

**Sinn:** externer 2ter Faktor z.B. Handy

**Kein Sinn:** Zeitfenster, IP-Adresse, Standort

**Firewalls**

Firewalls schützen Netzwerke und Systeme vor unbefugtem Zugriff und Angriffen, indem sie den Datenverkehr nach festgelegten Regeln kontrollieren.

**Arten von Firewalls:**

* **Stateless Firewalls (Paketfilter):** Filtern Datenpakete nach festen Regeln.
* **Stateful Firewalls:** Berücksichtigen den Verbindungsstatus und bieten besseren Schutz als Stateless Firewalls.
* **Application-Level Firewalls:** Erkennen Protokolle und blockieren untypische Port-Nutzung.

**Firewalls-NAT:**

* NAT = Network Address Translation
* Entwickelt, um zu verhindern, dass zu wenige IPv4 Adressen verfügbar sind
* Dient dem Verstecken von IPs & Netzwerken hinter anderen IPs & Netzwerken

**Paketfilter (Stateless-Firewall):**

Bsp: Zugriff aus dem Internet auf die IP 192.168.1.3 auf Port tcp/80 und tcp/443 erlauben

Any Any 192.168.1.3/32 tcp/80,tcp/443 Allow

SRC IP | SRC Port | DST IP | DST Port | Allow / Deny

**VPN**

VPNs ermöglichen eine sichere Verbindung zu einem Netzwerk über das Internet. Sie schützen Daten, indem sie sie durch verschlüsselte Tunnel leiten.

Zugriff auf Netzwerk, ohne physikalisch vor Ort zu sein.

**VPN-Typen:**

* **Site-to-Site VPN:** Verbindet zwei Netzwerke direkt miteinander.
  + Wird durch Netzwerkhardware hergestellt (Firewall zu Firewall)
* **Remote Access VPN (Road Warrior):** Einzelne Nutzer können aus der Ferne auf ein Netzwerk zugreifen, meist auf Unternehmensnetzwerke.
  + Benötigt keine spezielle Hardware auf der Seite des Clients

**Public Key Infrastructure (PKI)**

Die PKI stellt sicher, dass öffentliche Schlüssel von Kommunikationspartnern authentisch sind, indem sie digitale Zertifikate verwendet.

Ein Zertifikat ist ein digitaler Datensatz, mit dessen Hilfe bestimmte Eigenschaften bzw. Identitäten nachgewiesen werden können.

**Typische Bestandteile eines Zertifikats**

• Subject / Common Name: Domain/Email – mittlerweile eher unwichtig

• Subject Alternative Name: Domains – sehr wichtig

• Validity: Not Before, Not After – sehr wichtig

• Issuer – informativ

• Serial Number – wichtig zur Überprüfung, ob Zertifikat gefälscht wurd

**Zwei gängige Ansätze zur Zertifikatsverwaltung sind:**

* **Web of Trust (OpenPGP):** Benutzer signieren gegenseitig Zertifikate und entscheiden selbst, welchen sie vertrauen.
* **Hierarchische Zertifizierungsstellen (X.509):** Eine Root-CA steht an der Spitze und signiert oder gibt Zertifikate an Sub-CAs weiter, die dann Zertifikate signieren. Einmal signierte Zertifikate können nicht zurückgenommen werden. Lösung dafür ist CRL, CRL gibt Auskunft über gesperrte Zertifikate. Weiterentwicklung ist OCSP Stapling, der Zertifikatstatus wird vom Server in regelmäßigen Abständen bei der VA angefragt und bei Bedarf an den TLS-Handshake angehängt.

**Let´s Encrypt**

Let’s Encrypt ist eine kostenlose Zertifizierungsstelle mit dem Ziel den gesamten Internet-Traffic HTTPS zu sichern.

* Seit 2014
* Motiviert zur Automation
* 90 Tage gültig
* Größte CA nach Domains

**Protokolle zur Ausstellung von Zertifikaten:**

* **ACME:** 
  + Version 2.0
  + Mögliche Zertifikate: Domain Validated - (Certs. / Wildcard Certs.)
  + Beschreibt die Kommunikation zwischen den Servern der CA und den Servern der Anwender
    - Basis: Austausch von JSON Nachrichten über HTTPS
    - Basis sind diese Challenges: HTTP-01 / DNS-01 / TLS-ALPN-01

**HTTP-01 Challenge:**

1. Anfrage an Let’s Encrypt ACME API
2. Let’s Encrypt ACME API liefert einen Token zurück
3. Url muss via Port 80 erreichbar sein (Datei mit Token + Fingerprint von Account-Key):
   1. http://<YOUR\_DOMAIN>/.well-known/acme-challenge/<TOKEN>
4. Rückmeldung an Let’s Encrypt API
5. Let’s Encrypt überprüft die Verfügbarkeit von mehreren Standorten weltweit
6. Zertifikat wird ausgestellt

* **Pro:** Einfach zu automatisieren/Ohne Zugriff auf DNS verwendbar
* **Con:** Keine Wildcard Zertifikate / Port 80 erforderlich / Schwieriger bei lastverteilten Webservern

**DNS-01 Challenge (WICHTIG, er fragts ab):**

1. Anfrage an Let’s Encrypt ACME API
2. Let’s Encrypt ACME API liefert einen Token zurück
3. TXT Eintrag für \_acme-challenge.<YOUR\_DOMAIN> mit dem Token als Wert muss angelegt werden
4. Rückmeldung an Let’s Encrypt API
5. Let’s Encrypt überprüft die Verfügbarkeit von mehreren Standorten weltweit
6. Zertifikat wird ausgestellt

* **Pro:** Ermöglicht Wildcard Zertifikate / Einfacher bei lastverteilten Webservern
* **Con:** Nur Sinnvoll, wenn DNS Anbieter eine API anbietet und Aktualisierungen schnell genug ausgerollt werden

**TLS-ALPN-01 Challenge:**

* Nachfolger der TLS-SNI-01 challenge
* Operiert auf Port 443 im Zuge des TLS Handhakes
* Aktuell nicht stark verbreitet
* **Pro:** Kann verwendet werden, wenn Port 80 nicht verfügbar ist
* **Con:** Aktuell nicht mit Apache/Nginx/Certbot unterstützt / Keine Wildcard Zertifikate möglich

**Certificate Transparency Logs**

* CAs müssen Info zu ausgestellten Zertifikaten in CTL ablegen
* CTLs werden von verschiedensten Anbietern bereitgestellt z.B. Google & Let´s Encrypt
* Browser prüfen, ob Zertifikat in genügend „passed“ CTLs hinterlegt, sind
  + Google Chrome
    - Gültigkeit < 180 Tage: Zertifikat muss in min. 2 CTL vorhanden sein
    - Gültigkeit > 180 Tage: Zertifikat muss in min 3 CTL vorhanden sein