

Powerpoints

Hashfunktionen

- Unidirektional (nur in eine Richtung)
- Deterministisch (selbe Eingabe => selber Hashwert)
- Eingabe => Beliebige Länge / Ausgabe => Fixe Länge
- Bsp: MD5, SHA-1, SHA-2 (256, 512), SHA-3
- Kollision: andere Eingang => gleiche Ausgabe
- Avalanche effect: kleine Änderung der Eingabe => große Änderung der Ausgabe
- Anwendung: Vergleich von Daten (Passwort), Prüfwert (downloadend Software),
- Problem => selber Hash = selbes Passwort
- Lösung: Salt **#Mitschrift**

Verschlüsselung

- Bidirektionale Funktion (verschlüsseln + entschlüsseln)
- nur mit Schlüssel entschlüsseln
- Symmetrisch / Asymmetrisch
- Symmetrisch: selber Schlüssel zum verschlüsseln und Entschlüsseln
- Pos: Schnell, Einfach zu Programmieren

- Neg: Austausch?, alle kennen Schlüssel (ist Nachricht wirklich von mir?)
- Asymmetrisch: privat und public key
- Pos: Public Key einfach austauschen, Autenzität gewährleistet (wirklich von mir?)
- Neg: langsam, schwer zu programmieren
- IRL: kobination
- TLS: Asymmetrisch um Symmetrischen Schlüssel zu tauschen => mit Symmetrischen entschlüsselt. Symmetrischer Schlüssel oft getauscht.

Phishing

- Email oder Website sieht offiziell aus um an Informationen zu kommen
- Wie erkennen **#Mitschrift**

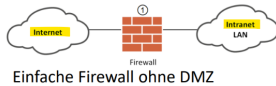
Mehr Faktor Authentifizierung

Something you:

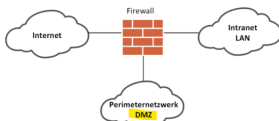
- know (Pin, Passwort)
- have (Handy, Token)
- are (Fingerabdruck, Gesicht, Iris)
- where (IP, GPS)
- times (Zeitraum)
- Sinn und kein Sinn **#Mitschrift**

Firewalls

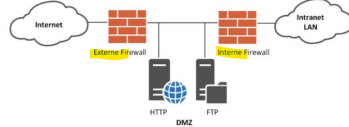
- Zugriff auf Netzwerk/System => Regeln (Policies)
- Hardware und Software mäßig
- gehärtetes Betriebssystem (keine Unnötigen Services)
- Platzierung: Gateway (Knoten)



Einfache Firewall ohne DMZ



Firewall mit mehreren Interfaces



Mehrstufige Firewall mit jeweils einem internen und einem externen Interface

00 Wiederholung

IT Sicherheit 23/24

12

DMZ => Services für Intern + Extern (Web, Mail)

Mehrstufige => von 2 Unterschiedlichen Herstellern

- Arten: Stateless, Statefull, Application Layer
- Layer 3(Network, IP), 4(Transport, TCP/UDP), 7
- Stateless: statischer Paketfilter (Regeln)
- Layer: 3 + 4
- Regeln siehe **#Übungen**
- Statefull: Kontext berücksichtigt
- State Tabelle (z.B. TCP Handshake)
- Layer 3 + 4
- Schutz vor Komplexen Attacks
- Application Level (Layer7): Webfilter + Untypische Protokolle am Falschen Port erkennen

NAT - Firewalls

- Network Address Translation
- ursprünglich: weil zu wenig IPv4 Adressen
- verstecken von IP/Netzwerken hinter anderen IP/Netzwerken
- Ablauf siehe **#Mitschrift**

VPN

- Zugriff auf Netzwerk obwohl nicht vor Ort
- Site 2 Site
- Remote Access Tunnel (Road Warrior)
- Welche verwenden? siehe **#Übungen**

Public Key Infrastruktur

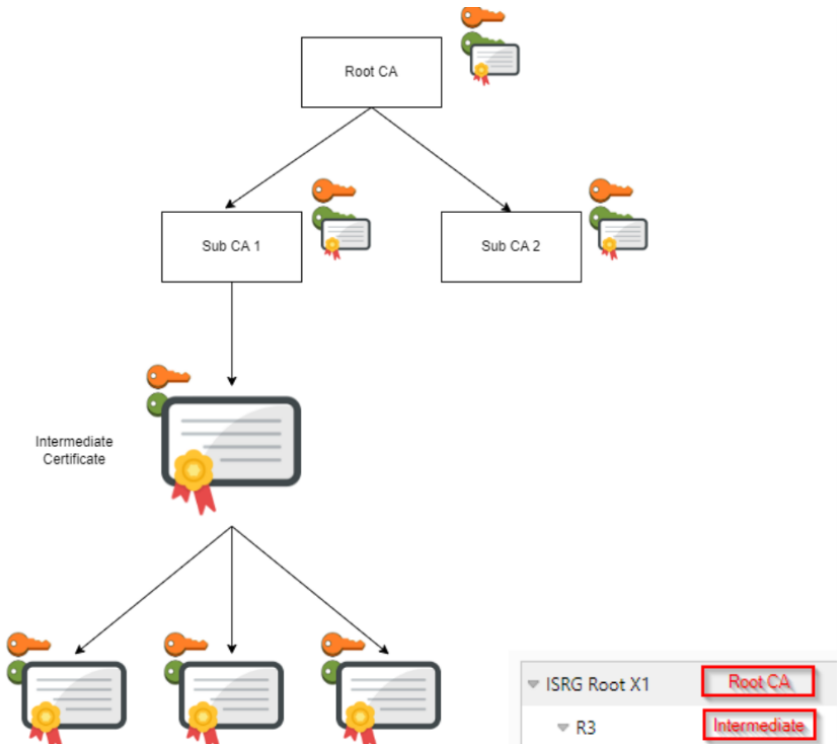
- Problem wie bekommt B den Public Key von A (könnte Ausgetauscht werden)
- Lösung: Zertifikate => Digitaler Datensatz mit dem ich Eigenschaften und Identitäten Nachweisen kann
- PKI: Struktur zur Ausstellung, Verteilung und Prüfung von Zertifikaten
- Varianten: OpenPGP-Web of Trust / X.509-Hierarchische Zertifizierungsstellen

Web of Trust

- Netz aus Teilnehmern + Teilnehmer erstellen selbst
- Teilen Selber (Keyserver, Email, USB)
- von Anderen Teilnehmern signiert

- selber entscheiden ob man Vertraut => mix x müssen signiert haben, bestimmter muss signiert haben

Hirachische Stellen



- Spitze => Root CA
- nur CA signiert
- mit Root CA => Sub Ca signieren => wieder signieren (jeder vertraut der der Root vertraut)
- Root CA: Private Key (offline only)
- Public Key und Zertifikate (von Browsern, OS vertraut)
- Sub CA: Private Key (offline only)
- Public Key und Zertifikate (von Browsern, OS vertraut)

- von Root CA signiert
- Intermediate Certificate:
 - Private Key: für Signierung von anderen Zertifikaten
 - Zertifikat: Private Key (von Admin erzeugt)
 - Public Key und Zertifikate (von Intermediate Public signiert)
- Vertrauen => Intermediate Key oder Sub CA mitgeben
- Problem: 1 mal signiert = nicht mehr zurück nehmen
- Lösung: CRL (Certificat Revocation List)
- gibt Auskunft über gesperrt Zertifikate
- Weiterentwicklung: OCSP (Online Certificate Service Protokoll) => Auskunft über Status
- Weiterentwicklung: OCSP Stapling => Status wird vom Server immer wieder von der CA angefragt, man kann ihn auch an den TLS Handshake anhängen
- Bestandteile:
 - Common Name => Domain = eher unwichtig
 - Subjekt Alternative Name: Domains = sehr wichtig
 - Validity: Not before, Not after => sehr wichtig
 - Issuer => informativ
 - Serial Number => überprüfen, ob es gefälscht wurde

Lets Encrypt

- Gratis
- seit 2014
- Ziel: HTTPS Standard
- Gültigkeit: 90 Tage

- Reduzieren des Schadens bei Kompromittierung
- Motiviert zur Automatisierung
- Größte CA (Domains, ausgestellte Zertifikate)
- Protokoll zur Ausstellung => ACME
- Mögliche Zertifikate: Domain Validate, Domain Validated Wildcards
- Kommunikation zwischen Server der CA und User Server
- JSON über HTTPS
- Challenges: HTTP, DNS, TLS-ALPIN

Challenges

- HTTP:
 - Anfrage an Lets Encrypt ACME
- API liefert Token
- URL muss über port 80 erreichbar sein
- <http://yourdomain/.well-known/acme-challenge/token>
- Rückmeldung an die API
- überprüft von mehreren Standorten
- bekommt Zertifikat
- Pro: automatisieren, ohne zugriff auf DNS
- Neg: keine Wildcards, braucht port 80, probleme bei lastenverteilten Webservern
- DNS:
 - Anfrage an ACME API
 - Liefert Token zurück

- TXT eintrag mit Token als Wert muss dort sein:
_acme-challenge.yourDomain
- Rückmeldung an die API
- von mehreren Standorten Prüfen
- Bekommt Zertifikat
- Pos: Wildcards Möglich, einfacher bei Lastenverteilten Servern
- Neg: nur sinnvoll wenn DNS Anbieter eine API anbietet und Aktualisierungen schnell genug gemacht werden
- TLS-Alpin:
 - Nachfolger von: TLS-SNI challenge
 - auf Port 443 mit TLS-Handshake
 - Pro: wenn Port 80 nicht verfügbar
 - Neg: nicht mit Apache/NGinx/Certbot unterstützt, keine Wildcards

Certificat Transparency Logs

- Cas müssen Infos zu ausgestellten Zertifikaten in CTL ablegen
- von verschiedenen Anbietern (Google, Letsencrypt)
- Browser prüfen ob sie bei genügend hinterlegt wurden
- Gültigkeit < 180 days => bei min 2 hinterlegt
- Gültigkeit > 180 days => min 3 hinterlegt

Quality SSL Test

- Bekommt Note

Übungen

- Firewall:

1. Zugriffe 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
```

2. Zugriffe aus dem Internet auf die IP 192.168.1.10 auf Port tcp/22 erlauben

```
Any Any 192.168.1.10(/32) tcp/22 Allow
```

3. Zugriffe von der IP-Range 192.168.3.0/24 auf IP 192.168.1.10 tcp/22 erlauben

```
192.168.3.0/24 Any 192.168.1.10(/32) tcp/22 Allow
```

Eigentlich Irrelevant da die 2. Regel einen SuperSet hat

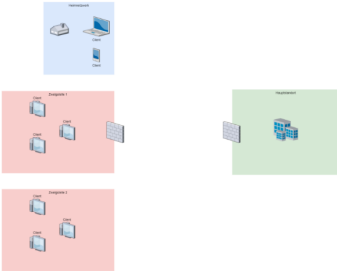
4. Alle anderen werden blockiert

```
Any Any Any Any Deny
```

- Hash:
- MD5 kürzer als SHA-256
- Symmetrisch Schneller

- BASE 64 = art der Darstellung keine Verschlüsselung

VPN



- Mit welcher Art von Tunnel werden die Standorte an den Hauptstandort angebunden (und weshalb):
 - Heimnetzwerk
 - Zweigstelle 1
 - Zweigstelle 2

00 Übungen
IT Sicherheit 23/24
5

- Heimnetz => Road Worrier
- Zweigstelle 1 => Site 2 Site (Gerät zur Verbindung)
- Zweigstelle2 => Road Worrier

Mitschrift

- Salted Hash:

1 als Datenbank User: Ich sehe die haben offenbar dasselbe Passwort (wohmöglich auch ein leichtes)

Lösung: Fixed Salted Hash: SALT = ABCD1234

User	MD5 (Passwort, Salt)
User 1	\$01\$ABCD1234\$edf\$
User 2	\$01\$ABCD1234\$edf\$

Endlösung: Random Salted Hash

User	MD5 (Passwort, Random Salt)
User 1	\$01\$xyzabc\$5f0\$
User 2	\$01\$12345a\$0db\$

- Phishing:
- Sendername bei Email sehr einfach setzen
- tiny url

- g00gle
- http statt https
- Datei Endungen
- Mehrfaktor athentifizierung
- Sinn: von Extern 2ter Faktor, aus der Firma nicht
- kein Sinn: IP + Zeit Fenster
- NAT:
- src: 172.18.8.96 : 4789 -> 142.251.36.68. : 80
(google)
- Public der HTL: src: 193.170.206.242 : 48222
(Übersetzung in der Nat-Tabelle)
- src: 142.251.36.68. : 80 (google) dest:
193.170.206.242 : 48222
- In der Tabelle schauen
- Mehrere Leute (Port)