Komposition: SSH + Crypto-1

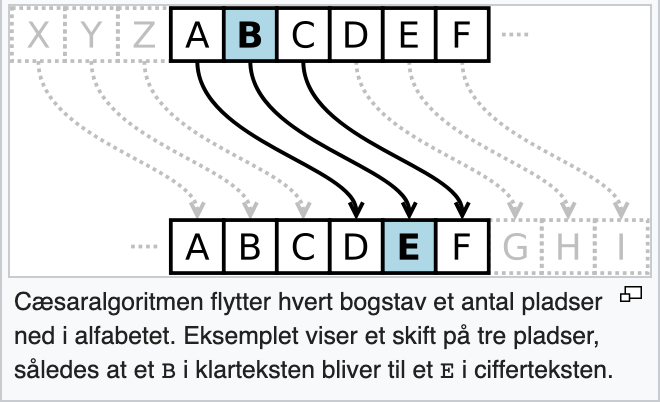
**Agenda:**

1. Forklar om symmetrisk og asymmetrisk kryptering, samt hashing.
2. Forklar om ovenstående i forbindelse med en SSH-forbindelse.
3. Forklar om SSH-tunnel og vis praktisk eksempel på dette.

**Explain conceptually all the following terms, and how/why they are needed for SSH and TLS/SSL**

**Symmetric Encryption**

* Cæsar Shift – brugt siden romertiden, og er super simpelt



* Enigma er en maskine som blev brugt af nazisterne under anden verdenskrig til symmetrisk kryptering
* Symmetrisk bruger kun 1 key til at kryptere og dekryptere
* Https bruger symmetrisk kryptering, fordi det er nemmere og derfor hurtigere
* Bruges derfor ofte til kommunikation, efter at en sikker forbindelse er etableret med asymmetrisk kryptering

**Asymmetric Encryption**

* RSA kryptering – Anvender primtal som gør det sværere at dekryptere, medmindre man bruger lave primtal
* Kom første gang på tale i 1870, men der gik ca. 100 år før det blev udviklet
* Bruger 2 keys (private og public) Public til at kryptere og private til at dekryptere
* Denne type er tung og tager lang tid, og bruges derfor ikke så tit når først en sikker forbindelse er etableret, men til at skabe en sikker forbindelse ved at udveksle public keys.
* Bruges bla. Til at etablere SSH-forbindelser
* Det kan gøres omvendt, men det bruges sjældent, da det giver mere mening at alle kan kryptere, men ikke dekryptere, frem for det modsatte
* Den omvendte metode anvendes oftest til at lave certifikater

**Hashing**

* Envejs algoritme – man kan ikke få den originale værdi ud fra en hashværdi
* Generere en værdi eller flere ud fra tekst ve brug af en matematiske funktion
* Kan bruges til sikkerhed i beskedtransmissionsprocessen, hvis beskeden kun er tiltænkt en specifik modtager
* Beskytter beskeden mod uønskede ændringer
* Hashing bruges også som metode til at sortere nøgle værdier i en database tabel
* Algoritmen skal være tidstagende for at være sikker, og modstå hackere
* Hashing bruges til at sikker at man har fået den rigtige public key ved SSH-forbindelser

**Explain what it takes to safely log in to an SSH server, without having to provide a password**

* Klientens public key skal ligges manuelt ind på serveren i filen ~/.ssh/authorized\_keys
* Første gang klienten så tilgår serveren, vil serverens public key blive sendt til klienten, sammen med den algoritme der bruges til at hashe den
* Klienten bliver bedt om manuelt at godkende den sendte public key ved at få præsenteret en hashværdi af den, genereret med den sendte algoritme
* Denne skal sammenlignes med den hashværdi som serveren har af nøglen
* Hvis disse stemmer overens, tilføjes public key til en fil der hedder ”known hosts” hos klienten
* På denne måde kender begge parter hinanden næste gang der logges in, og kan fortage log in uden adgangskode

**Explain the term SSH-tunnel, and provide a practical example for its use**

Et billede, der indeholder tegning

Automatisk genereret beskrivelse

**Generelt om SSH tunneling:**

* SSH tunneling er en måde hvorpå, datatrafikken for en given applikation ved hjælp af port forwarding. Tunneling kan etableres for enhver TCP/IP – port over SSH. Dette betyder, at applikationens datatrafik dirigeres til at strømme inde i en krypteret SSH-forbindelse. SSH-tunneling muliggør tilføjelse af netværkssikkerhed til ældre applikationer, der som udgangspunkt ikke understøtter kryptering.

**I forbindelse med en database:**

* SSH-tunnel vil sige at tilgå serveren gennem en åben SSH port (port 22) til en bruger på serveren som har adgang til de porte som er lukket for udefra kommende forbindelser f.eks. en database forbindelse (port 3306)
* SSH-tunnels kan derfor bruges til at for adgang til services på tværs af firewalls.

**Praktisk eksempel med forbindelse til databasen (se under ”Connecting to MySQL via an SSH Tunnel”):**

Vi har sat en droplet op med en Firewall, der har lukket for port 3306 (mysql).

I terminalen skriver man: sudo uwf deny mysql

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Dernæst har vi oprettet forbindelse til MySql, gennem port 22:

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

Der er oprettet forbindelse. Her er det vigtigt at bide mærke i at forbindelsen til MySql sker gennem localhost (127.0.0.1).

Et billede, der indeholder skærmbillede, computer

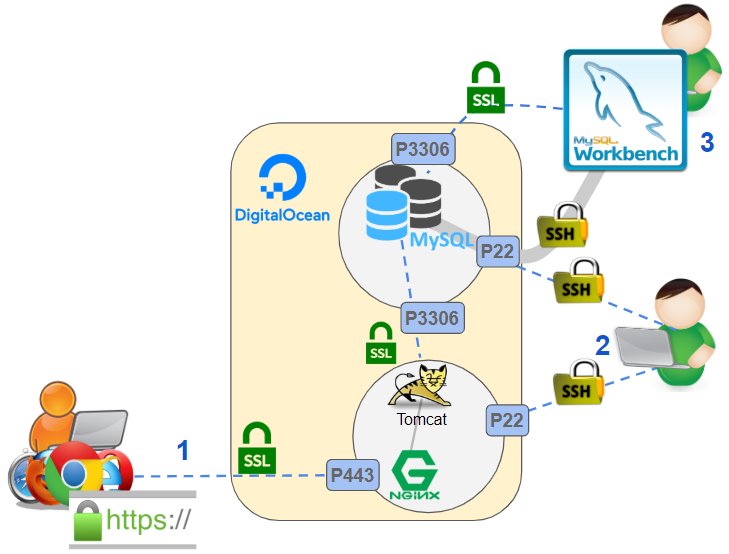
Automatisk genereret beskrivelse

Her kan man se at der ikke er adgang til MySql på serveren, gennem port 3306.

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

**Illustration af SSH-forbindelse**



**Explain conceptually the purpose of Symmetrical Encryption, Asymmetrical Encryption and hashing for an SSH-connection**

**Symmetrisk kryptering:**

Symmetrisk kryptering anvendes til data-in-motion kryptering. Data-in-motion-kryptering er en sikkerhedsfunktion, der forhindrer andre, der forsøger at lytte med, i at se data sendt via et netværk. Med andre ord holdes den transmitterede data fortrolig. Det understøttes i både SSH og SSL og fungerer ved at konvertere ren tekstdata til det, der kaldes cipher tekst.

Både SSH og SSL kryptere data-in-motion gennem symmetrisk nøglekryptering. Denne type kryptering bruger en delt nøgle, der bruges til både kryptering og dekryptering.

**Asymmetrisk kryptering:**

En krypteret forbindelse bliver ubrugelig, hvis du ubevidst har oprettet forbindelse til en falsk server eller en ondsindet klient. Mens SSH og SSL bruger symmetrisk kryptografi for at bevare fortroligheden af ​​transmitterede data, bruger de en anden form for kryptering til authentication. Autentificering tillader den ene part at kontrollere, om den anden part virkelig er den, den hævder, at den er.

For at implementere authentication, bruger SSH og SSL asymmetrisk kryptering (public key kryptering). De populære krypteringsalgoritmer til offentlig nøgle er RSA, DSA og ECDSA, som alle understøttes af både SSH og SSL.

**Hashing:**

SSH bruger hashing til at verificere integritet af ​​meddelelser. Dette gøres ved hjælp af MAC (**H**ash-based **M**essage **A**uthentication **C**odes). Dette sikrer, at den modtagne meddelse ikke er manipuleret på nogen måde.

Hver meddelelse, der transmitteres, skal indeholde en MAC, der beregnes ved hjælp af den symmetriske nøgle, pakkesekvensnummer og indholdet af meddelelsen. Det sendes uden for de symmetrisk krypterede data som den afsluttende del af kommunikationspakken. Det anbefales at data først krypteres, og derefter beregnes MAC.

**Explain the steps you have to go through to set up a server with MySQL, as secure as possible →**

* **How can we limit the client IP's that can connect**
  + Ignorer for al trafik indad: sudo ufw default deny incoming
  + Tillad adgang for en specifik IP-adresse: sudo ufw allow from <ip address>
* **If set up to allow only localhost and a firewall that deny 3306, can we still connect “safely” from a remote server**
  + Setup af mysql bruger med alle rettigheder:

mysql> CREATE USER 'no\_ssl'@'127.0.0.1' identified by 'test';

mysql> GRANT ALL ON example.\* TO 'no\_ssl'@'127.0.0.1';

mysql> GRANT SELECT ON performance\_schema.\* TO 'no\_ssl'@'127.0.0.1';  
mysql>FLUSH PRIVILEGES;

* + Setup server til kun at tillade localhost forbindelse i mysql:

 sudo nano /etc/mysql/my.cnf

Indsæt:

[mysqld]

# Require clients to connect either using SSL

# or through a local socket file

# require\_secure\_transport = ON

bind-address = 127.0.0.1

* + Luk port 3306:  sudo ufw deny mysql
  + Mysql kan sikkert tilgås gennem en SSH-tunnel via port 22 (SSH) til den oprettede Mysql bruger.

**How to set up an SSL (Secure Socket Layer (https)) connection that anyone can use**

1. Gå ind i filen: sudo nano /etc/mysql/my.cnf
2. Ændr værdien for bind-address, til: bind-address = 0.0.0.0
3. Fjern dette i filen: require\_secure\_transport = ON
4. Genstart serveren: sudo systemctl restart mysql
5. Tillad port 3306 i firewall: sudo ufw allow mysql

**Praktisk eksempel: Connecting directly to MySQL with Mandatory SSL**

Log ind på mysql: mysql -u root -p

Se status på forbindelsen til mysql, for bl.a. at se SSL ikke er i brug

mysql> \s

Et billede, der indeholder bord

Automatisk genereret beskrivelse

For at se hvilke certifikater og keys der er på mysql, skriv i terminalen:

sudo find /var/lib/mysql -name '\*.pem' -ls

Et billede, der indeholder sidder, tastatur, bord, computer

Automatisk genereret beskrivelse

Certificate Autority: **ca.pem**

MySql server process: **server-key.pem** og **server-cert.pem**

MySql clients: **client-key.pem** og **client-cert.pem**

De her certifikater er med til at lave SSL forbindelsen. MySql 8.x generer selv automatisk de nødvendige SSL/TCP-certifikater ved installation

Log ind på mysql igen og specificer host for at sikre at der laves en TCP/IP forbindelse til den lokale server: mysql -u root -p **-h 127.0.0.1**

mysql> \s

her ses den specifikke SSL cipher, som indikere at SSL er i brug:

Et billede, der indeholder bord

Automatisk genereret beskrivelse

Lav en ny bruger på mysql (vigtigt: kræver SSL-forbindelse)

mysql> CREATE USER 'everywhere'@'%' identified by 'test' REQUIRE SSL;

mysql> GRANT ALL ON example.\* TO 'everywhere'@'%';

mysql> FLUSH PRIVILEGES;

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelseVi kan forbinde til mysql på serveren, gennem port 3306 med SSL.

**Demonstrate a client application (Java or whatever you prefer) running on a separate server that access the Database using SSL**

Installer en mysql klient på serveren:

sudo apt update

sudo apt install mysql-client

Test forbindelsen og at den er sikker: **mysql -u everywhere -p -h your\_mysql\_server\_IP**

Et billede, der indeholder sort

Automatisk genereret beskrivelse

Når SSL er disabled, er det ikke længere muligt at skabe forbindelse:

mysql -u everywhere -p -h **your\_mysql\_server\_IP** --ssl-mode=disabled

Det kan ses at forbindelsen nu er lukket:



For at øge sikkerheden endnu mere, kan man sætte firewall til kun at tillade specifikke IP-adresser til at kunne forbinde til mysql gennem port 3306:

Et billede, der indeholder sort, skærm, opbevarer, sidder

Automatisk genereret beskrivelse