Exam questions | Week-09 SSH + Crypto-1

Made by Klasse A, Group 6

- Cph-mh748 - Malte Hviid-Magnussen
- Cph-rn118 - Rúni Vedel Niclasen
- Cph-ab363 - Asger Bjarup
- Cph-cs340 - Camilla Staunstrup

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

- SSH (secure shell) is a command line connection (server-to-server) and goes through port 22
  - The SSH protocol uses symmetric encryption, asymmetric encryption and hashing in order to secure transmission of data.
- TLS/SSL (Transport Layer Security (https)) a secure connection between browser and web server and goes through port 443
  - The connection is secure because symmetric encryption is used to encrypt the data. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret that was negotiated at the start of the session (during the handshake). The negotiation of a shared secret is secure and reliable.
  - The identity of the communicating parties can be authenticated using asymmetric encryption. This authentication can be made optional, but is generally required for at least one of the parties (typically the server).

- **Symmetric Encryption**
  - It requires just one key to help with both encryption (encoding) and decryption (decoding) of confidential data.
  - Symmetric Encryption is mostly required when dealing with the transmission of bulk data (fx. from webserver to webbrowser). This is because it’s much quicker and easy to execute.
  - The single key nature of symmetric encryption has riskful transportation. No transport is guaranteed, meaning that if the key is compromised, so is the encrypted data.

- **Asymmetric Encryption**
  - Requires a pair of matching keys (public and private) to help with encryption and decryption purposes.
  - Asymmetric Encryption is a viable option if you only wish to get a secure environment for exchanging your secret keys. This is because of the complexity it has in execution and the slow speed in using it.

    - Asymmetric encryption is safer than symmetric encryption as only the public key is provided for encrypting messages. The private key is never shared and cannot be derived from the public key. (Source)

- **Hashing**
    - Mapping data of arbitrary size to a bit-string of a fixed size (the hash value) and is a one-way function, (it is practically infeasible to invert). Ideally, the only way to find a message that produces a given hash is to attempt a brute-force search of possible inputs to see if they produce a match, or use a rainbow table of matched hashes.
    - The ideal cryptographic hash function has the following main properties:
        - it is deterministic, meaning that the same message always results in the same hash
        - it is quick to compute the hash value for any given message
        - it is infeasible to generate a message that yields a given hash value
        - it is infeasible to find two different messages with the same hash value
        - a small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value

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

- SSH key pairs are two cryptographically secure keys that can be used to authenticate a client to an SSH server. Each key pair consists of a public key and a private key. The private key is retained by the client and should be kept absolutely secret
    - SSH tests the client by encrypting some data with the recorded public key, sending it to the client, and requiring that the client decrypt and send back the same data. If the client can successfully decrypt and send back the same data, it must have the private key associated with the recorded public key. Therefore, the client is who it claims to be.
- SSH clients store host keys for hosts they have connected to. These stored host keys are called known host keys, and the collection is often called known hosts.

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

- An SSH-tunnel is a connection from a client to a server through port 22. The connection is established using the client’s key pair. The server knows the public key of the client and can therefore confirm that the client is “the real client” if the client can decode the data sent by the server.
  SSH tunnels may be used to enable secure connections to legacy systems that don’t have an implementation for secure online connections. Blocking off all incoming traffic except port 22 will allow clients or other servers etc. to connect securely to the system via SSH. Once they’re in the system, they can access different ports via localhost.

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

- See first question.

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**
    - First of all we need to set up an *ufw* (uncomplicated firewall) since we're on a linux system. Using ufw we then set the default setting to deny all incoming connections. Doing this and creating a whitelist is the easiest way to control which ips are allowed to connect.
    - If you only want to allow connection to the Mysql database through an SSH tunnel, then you need to configure the Mysql database to only accept connections from localhost. If you go for this approach then you can deny all connections except connections through port 22.

- **If set up to allow only localhost and a firewall that deny 3306, can we still connect "safely" from a remote server**
    - Yes, through an SSH-tunnel

- **how to set up an SSL connection that anyone can use,**
    - Mysql does this automatically :HideThePainHarold:
    - This question???
    - This was the "everywhere" part afaik. If you don't restrict the IP that can connect, then anyone can use the SSL connection, right?
        - https://raw.githubusercontent.com/MalteMagnussen/Security/master/week9/successfully%20made%20everywhere%20ssl%20connection.png
        - But not able to connect insecurely:
            - https://github.com/MalteMagnussen/Security/blob/master/week9/not%20possible%20to%20connect%20insecurely.png?raw=true
        - Made it here:
            - https://github.com/MalteMagnussen/Security/blob/master/week9/create%20new%20remote%20user%20to%20test%20ssl%20connection.png?raw=true
        -

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

- ~~🟥 how to set up an SSL connection that requires clients to identify themselves with a certificate.~~