SSL - Socket Socket Layer

Uses asymmetric and symmetric encryption.

asymmetric:

① Authentication with server

② Exchange of symmetric key

symmetric:

① For transmission

client

① Request SSL Connection → Server

② Sends certificate ←

③ Client verifies certificate

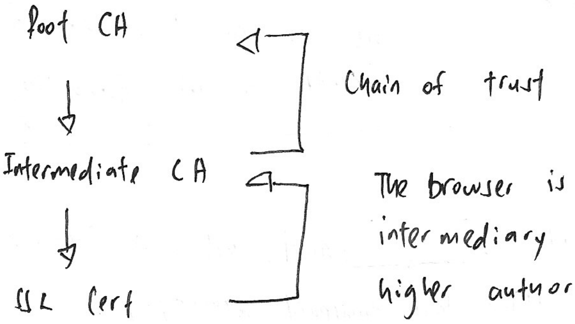
④ Client send encrypted session key →

⑤ Server begins encrypted connection ←

Contains public key of server which is used to encrypt session key that is sent back to server. Server can use its private key to decrypt session key. The session key is then used as a symmetric key for encryption between client and server.

This is only possible due to PKI (Public key infrastructure). PKI is a method for distributing a symmetric key.

How does browser authenticate servers? Through certificate chains of trust.

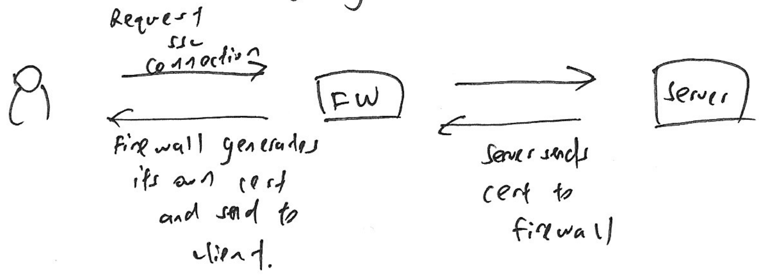


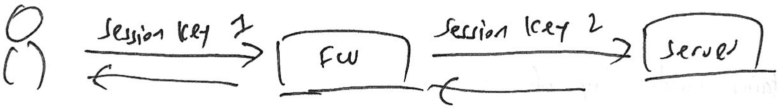
The browser is able to check which authority issued the intermediary certificate, retrive the key from that higher authority and verify the intermediary certificate. This process would contiune until a root CA is encountered. Root certificate is self-signed certificate, since the issuing CA is itself.

More details of certificate verification and digital signature is in INS (Information Security).

Outbound SSL Inspection by Foward Proxy

→ FW inspects outgoing SSL traffic to allow or block based on policy.


Request SSL connection
FW
Server
Firewall generates its own cert and send to client.
Server sends cert to firewall



Two session keys are used to maintain connection with external server. FW functions as a forward proxy or a mitm. (Man-in the middle)

How it works

① Firewall intercepts client SSL cert request

② Firewall forwards the request to the server, but generates a certificate on-demand in response to client's request.

③ This results in secure connection between FW and client.

④ Firewall acts as forward proxy and initiates a secure connection to actual server using server's cert.

⑤ Firewall mitm succeeded. Traffic flowing through the client and server can be read by firewall.

Minor Issues

① Firewall issued a self-signed cert to client. This cert is not valid and will incur wrath on client's browser (especially chrome's HSTS (HTTP strict transport security))

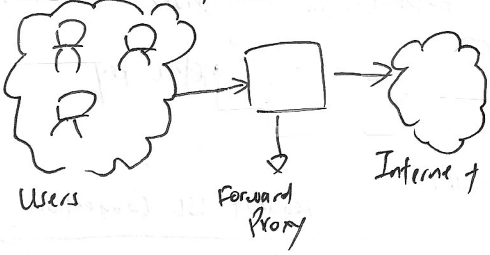
Solution: 1) If using a self-signed CA, export public CA cert from firewall and install the cert as a trusted Root CA (chain of trust) on each client's machine's browser to avoid untrusted certificate error messages.

2) Network admin use GPO (Group policy object → windows server thing) to push this certificate to each workstation. (Mass deployment)

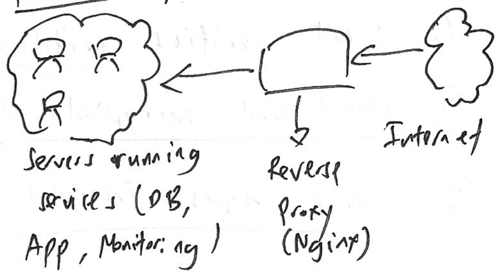
② What if certificate from server is untrusted by FW (FW also validates server's cert)?

Solution: An "untrusted" cert will be generated by the FW and sent to client. This will inform user of an actual untrusted cert error and possibility of mitm.

Forward Proxy

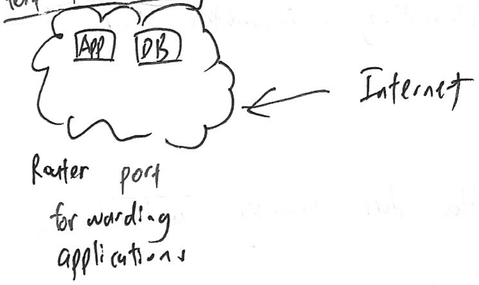


Reverse Proxy



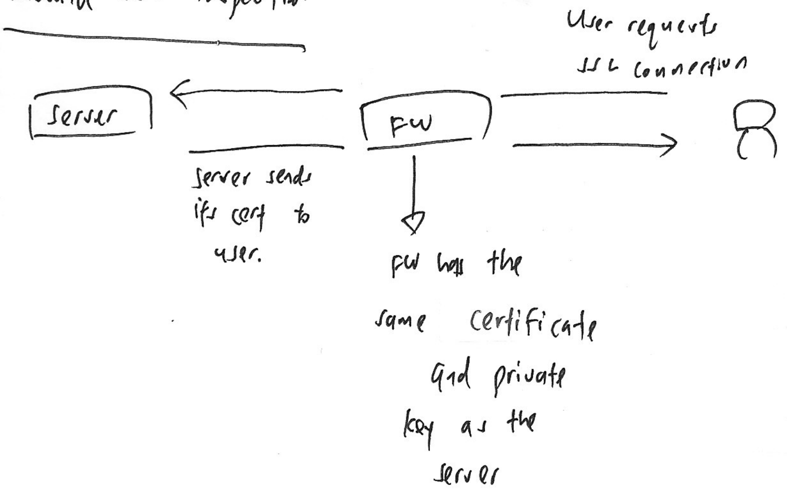
Reverse Proxy is different from port forwarding.

Port forwarding



There is no extra layer of security for port forwarding

Inbound SSL Inspection


Server
FW
User requests SSL connection
Server sends its cert to user.
FW has the same certificate and private key as the server

Packet Data remains unchanged and the connection is secure from client system to internal SSL server.

How it works

1. SSL Decryption Policy needs to be set on firewall to inspect incoming traffic.
2. Once this is done, FW will be able to decrypt and read the traffic prior to forwarding traffic to server.
3. Data packet is re-encrypted and no changes is made to the data.
4. Secure channel is built from external client to internal SSL server.

Unsupported Apps

1. Apps that use client side certificates
2. Non-rfc compliant applications
3. Servers using unsupported cryptographic settings

} Set to "no-decrypt" for affected applications.

"When first implementing SSL decryption, an approach will be to avoid breaking applications that cannot be decrypted"