Firewalls and Security Protocols

Today's Plan

- 9:00 – 9:15: Review Python activities.
- 9:15 – 10:00: **Mark Stamp, *Information Security Principles and Practices* [Chapters 8 and 10], Second Edition, Wiley, USA.**
  - Firewalls
  - Security Protocols
    - SSH
    - SSL
- 10:00 – 11:00: Demo.
- 11:00 – 12:00: Lab 6 (Firewalls and SSH in Linux).

Firewalls

Firewalls

- The secretary of the network.
- Examines requests for access to your network.
- Positioned between your network and the internet.

Types of Firewalls

- Packet Filter: Operates at the network layer.
- Stateful Packet Filter: Operates at the transport layer.
- Application proxy: Operates at the application layer.

Packet Filter

- Filters based on:
  - Source IP address.
  - Destination IP address.
  - Source port.
  - Destination port.
  - TCP flags (SYN, ACK, RST, …) **?**
- Filters based on ingress and egress.
- Configured using ACLs.
- Pros:
  - Efficiency.
- Cons:
  - Each packet is treated independently.
  - Can’t examine a TCP connection.
  - Blind to viruses and malware.

ACLs for Packet Filtering

	Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
1	Allow	Inside	Outside	Any	80	HTTP	Any
2	Allow	Outside	Inside	80	> 1023	HTTP	ACK
3	Deny	All	All	All	All	All	All

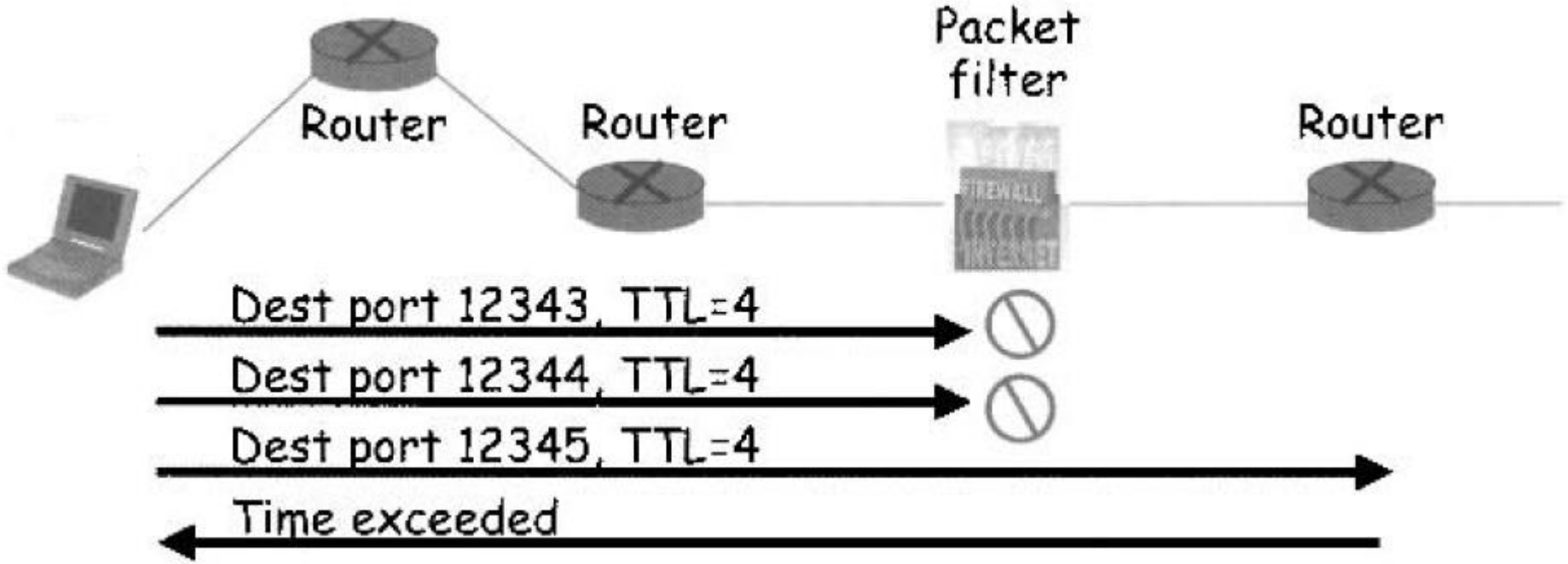
Attack on a Packet Filtering (TCP ACK)

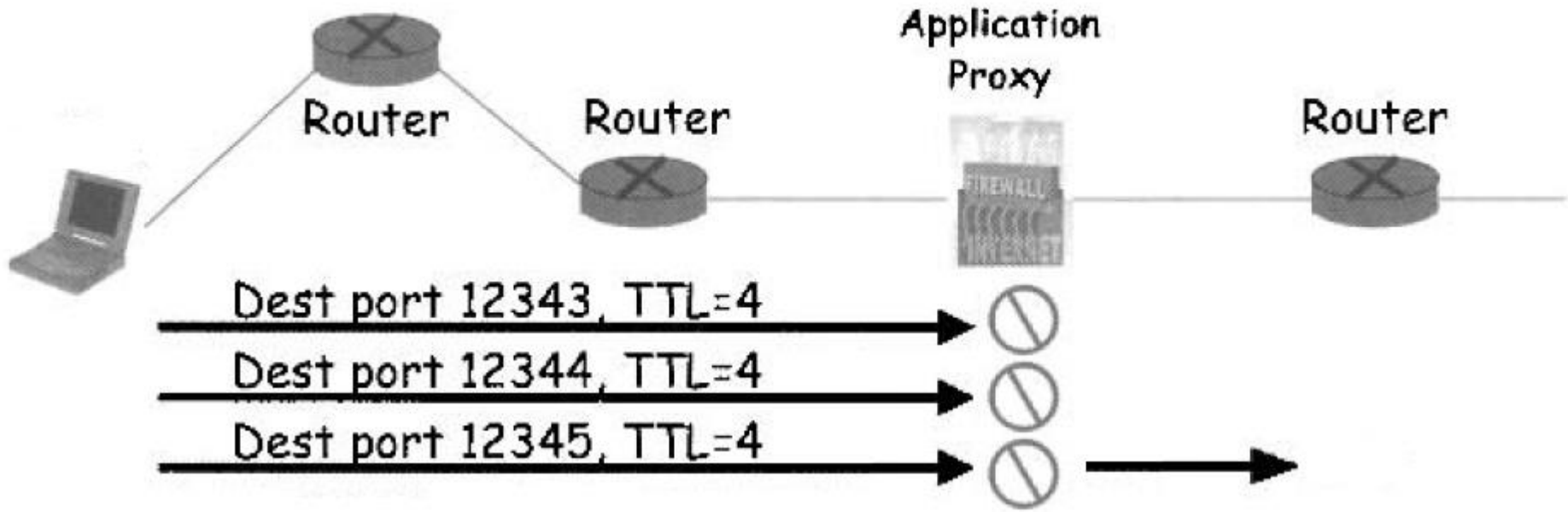
Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

- Port scan:
  - Send a packet with ACK bit set (even if such thing violates the TCP 3-way handshake protocol).
  - A host receiving this packet will respond with a RST packet to terminate the connection.
  - If the attacker receives a RST packet, then such port is open.
- Prevented by using stateful packet filters:
  - Remembers TCP and UDP connections.
  - Slower processing, still not aware of viruses/malware.

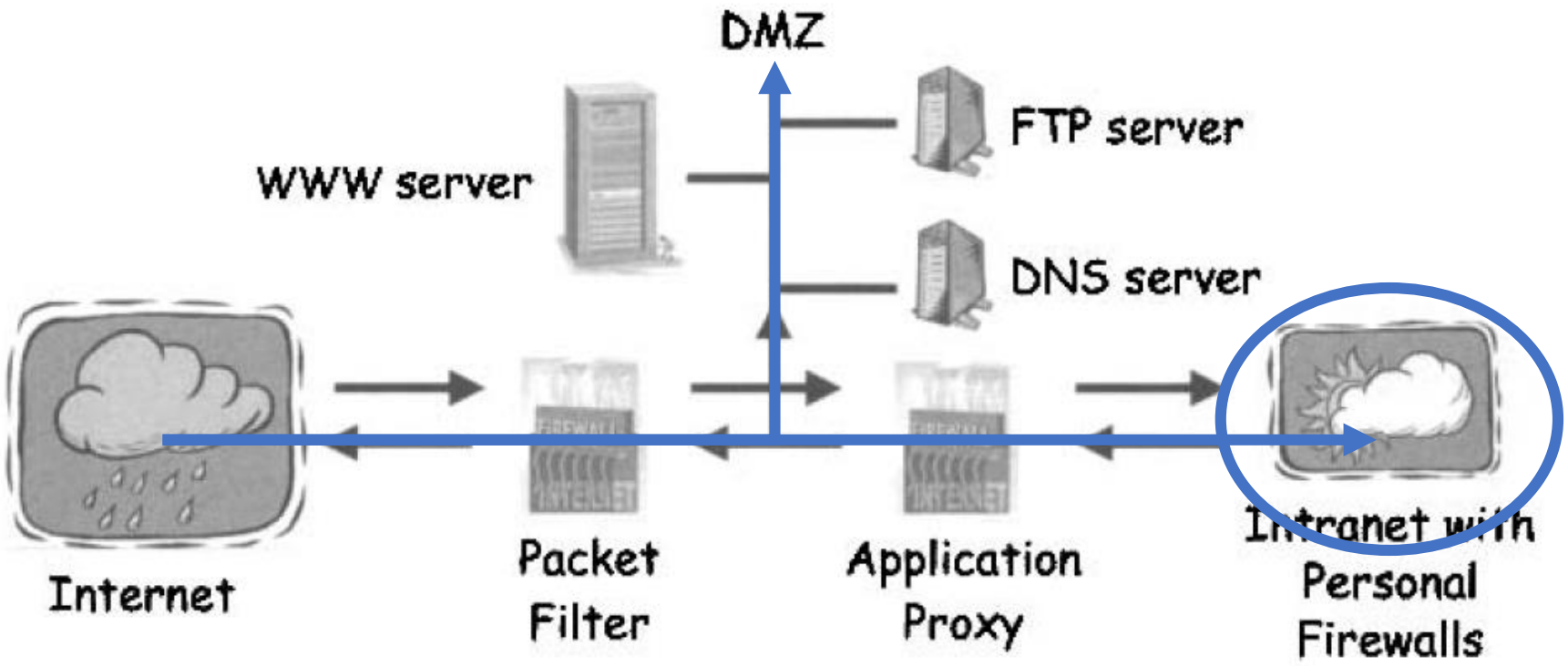
Application Proxy

- Proxy: Something that acts on your behalf.
- The firewall, acting on the user's behalf, is able to verify that the packet is legitimate and that the content is safe.
- When a packet passes through the firewall, the incoming one is destroyed and a new one is created in place.
  - This avoids TCP ACK attacks since the attacker will not receive time exceeded responses.


Router
Router
Packet filter
Router
Dest port 12343, TTL=4
Dest port 12344, TTL=4
Dest port 12345, TTL=4
Time exceeded


Router
Router
Application Proxy
Router
Dest port 12343, TTL=4
Dest port 12344, TTL=4
Dest port 12345, TTL=4

Defence-in-Depth Using Firewalls



Secure Shell (SSH)

Secure Shell (SSH)

- Creates a secure tunnel used to transmit insecure commands:
  - UNIX *rlogin* to log into a remote machine.
  - Ubuntu *ssh* command.
  - Once the SSH connection is established, such login typically only requires a password sent in the clear.
  - Sending this password through SSH will secure that password from sniffers.

- Based on:
  - Public key cryptography (Diffie-Hellman, RSA, …).
  - Digital certificates.
  - Hashing functions.

  or
  - ID/Passwords.

Diffie-Hellmann (DH) algorithm

- “Invented” by Withfield Diffie and Martin Hellman.
- Public-key cryptography protocol.
- Key exchanging: Allows users to establish a shared symmetric key.
- Relies on the computational complexity of calculating a discrete logarithm.

The Discrete Logarithm Problem

- Given $x = g^k$, to know the value of k you compute $log_g(x)$.
- This is known as the **logarithm problem**.
- Given $x = g^k \ mod \ p$, to determine k you compute the logarithm problem in a discrete setting.
- This is known as the **discrete logarithm problem** (NP-complete).

DH

- Let p be a prime number and g a *generator* (both public).
- For every $x \in \{1,2, \dots, p-1\}$ there exists an exponent n such that $x = g^n$.
- A first user (Alice) randomly selects a secret exponent a as well as a second user (Bob) selects a secret exponent b.
- Alice computes $g^a \ mod \ p$ and sends result to Bob. Bob computes $g^b \ mod \ p$ and sends result to Alice.
- Alice does $g^{b^a} \ mod \ p = g^{ab} \ mod \ p$ as Bob does $g^{a^b} \ mod \ p = g^{ab} \ mod \ p$.

DH

- As a result, $g^{ab}\ mod\ p$ becomes the shared secret (never transmitted).

Alice

Bob

- Any attacker can see $g^{a}\ mod\ p$ or $g^{b}\ mod\ p$.

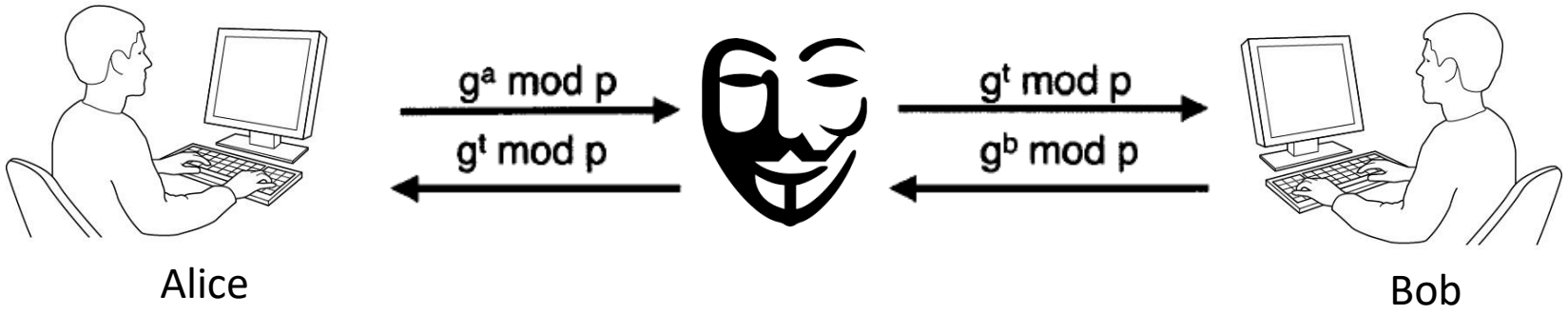
- However, it will be very hard for the attacker to know $g^{ab}\ mod\ p$ since:

$$g^{a} * g^{b} = g^{a+b} \neq g^{ab}\ mod\ p$$

and the attacker needs to find *a* or *b*.

Fundamental Problem

- Man-in-the-middle attack.



- Solution:
  - Encrypt DH with shared symmetric or public key techniques.
  - Sign DH values.

Then why using DH?

SSH (simplified)

$$\begin{aligned}
\text{certificate}_A &= \text{Alice's certificate} \\
\text{certificate}_B &= \text{Bob's certificate} \\
\text{CP} &= \text{crypto proposed} \\
\text{CS} &= \text{crypto selected} \\
H &= h(\text{Alice}, \text{Bob}, \text{CP}, \text{CS}, R_A, R_B, g^a \bmod p, g^b \bmod p, g^{ab} \bmod p) \\
S_B &= [H]_{\text{Bob}} \\
K &= g^{ab} \bmod p \\
S_A &= [H, \text{Alice}, \text{certificate}_A]_{\text{Alice}}
\end{aligned}$$

h is a cryptographic hash function.

Alice

Bob

Security of SSH

- Nonce R_A is Alice's challenge to Bob, and S_B is Bob's response (and vice versa).
- Both Alice and Bob get authenticated through their signatures.

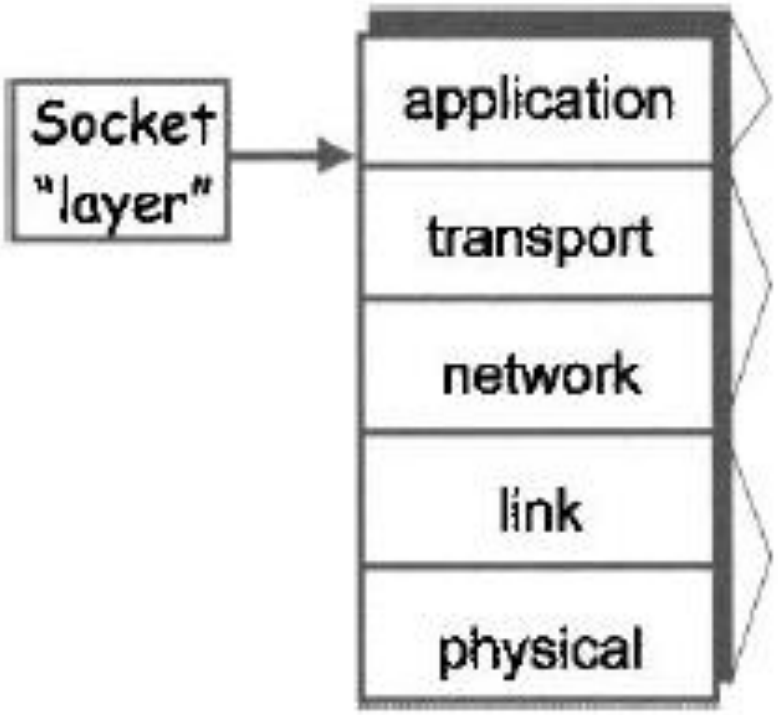
Will a man-in-the middle attack on the DH values compromise SSH?

Then what could?

Secure Socket Layer (SSL)

Secure Socket Layer

- Socket layer between the transport and application layers.
- Often deals with web browsing.
  - Application protocol is HTTP.
  - Transport protocol is TCP.
- When a website is secured by SSL, it displays **https**.
- The most current version of SSL is called Transport Layer Security (**TLS**).



Secure Socket Layer

- Preferred choice for internet transactions:
  - Non-mutual operation.
  - Security for the buyer's detail is held by the seller once the secure connection is established.

Alice

Bob

SSL (Simplified)

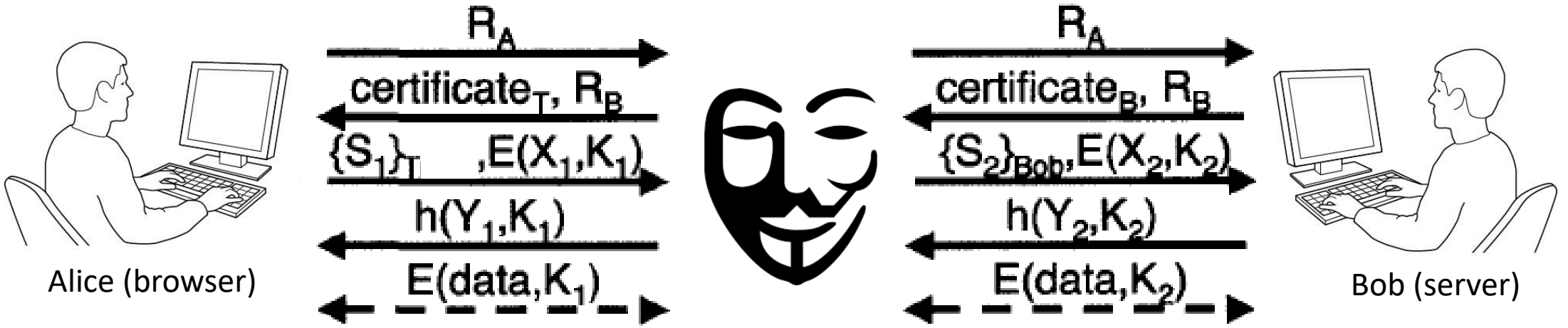
S = the pre-master secret
$K = h(S, R_A, R_B)$
msgs = shorthand for "all previous messages"
CLNT = literal string
SRVR = literal string

Alice (browser)

Bob (server)

Is Alice being authenticated by Bob?

Man-in-the-Middle attack on SSL



Why will this attack on SSL fail?

What if the attacker makes a false $certificate_{B'}$?

What if the attacker uses true $certificate_B$?

Who allows SSL attacks to happen?

How does SSL look like?

- Standard SSL Certificates

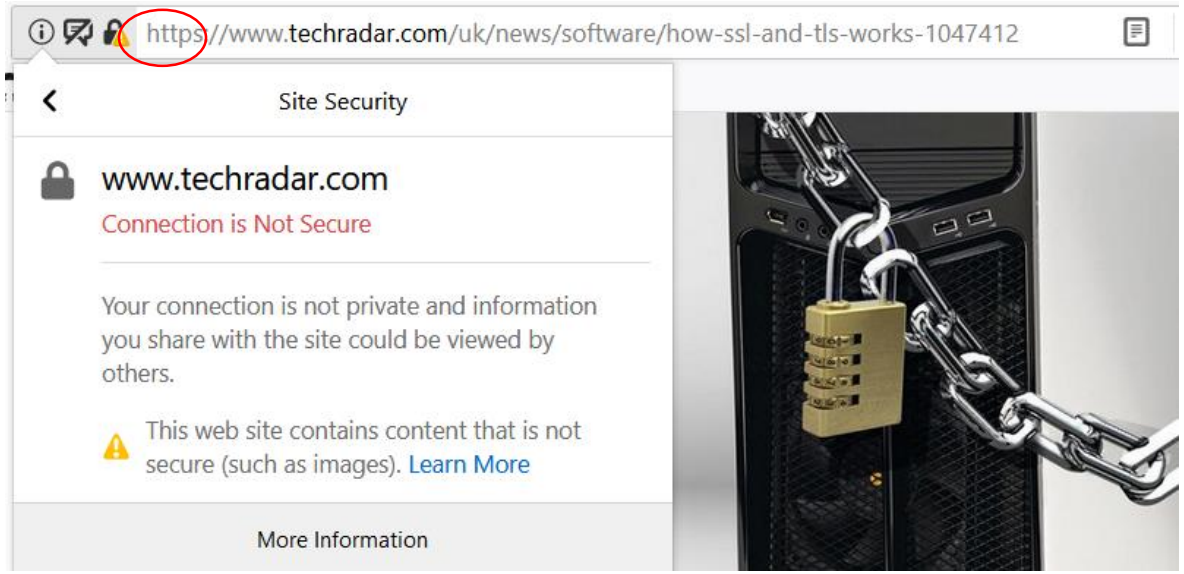
The standard HTTP is changed to HTTPS, automatically telling the browser that the connection between the server and browser must be secured using SSL.

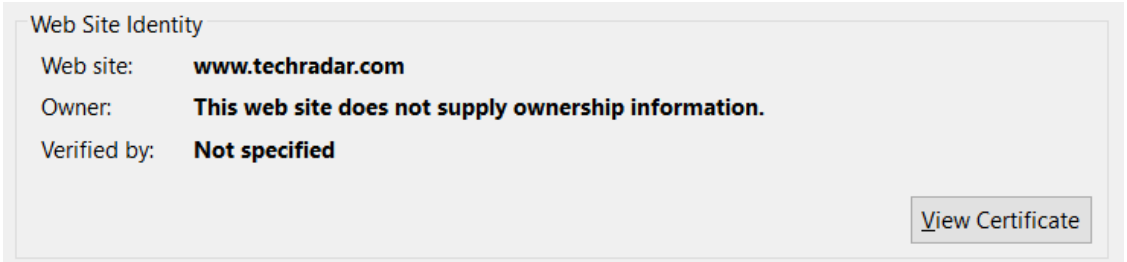
The padlock is activated, showing you that the browser connection to the server is now secure. If there is no padlock or the padlock shows a broken symbol, the page does not use SSL.

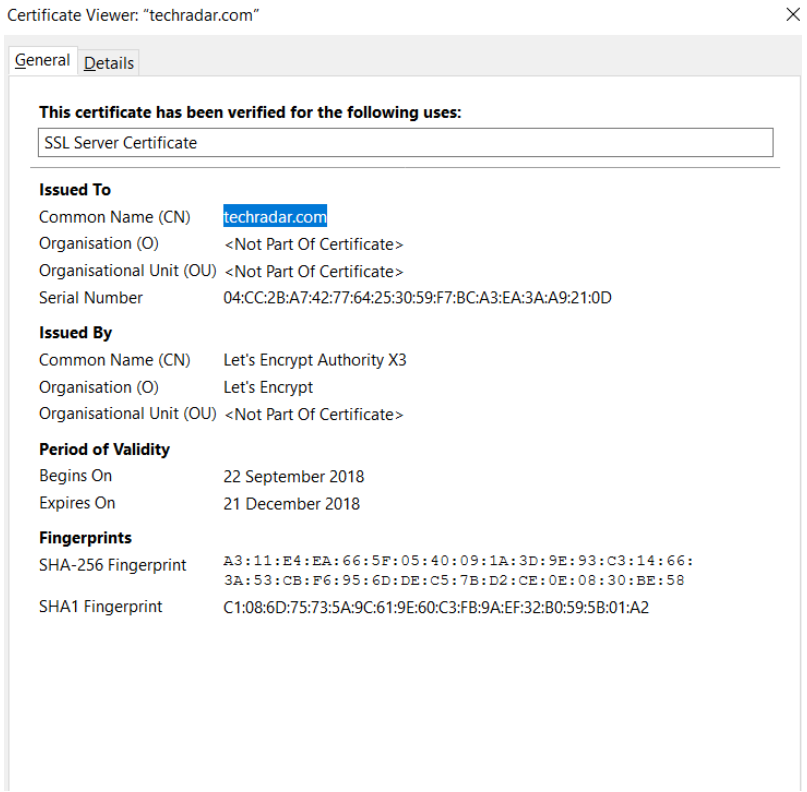
https://www.globalsign.com/en/ssl-information-center/what-is-an-ssl-certificate/

How does SSL look like?

- Issues with Standard SSL Certificates

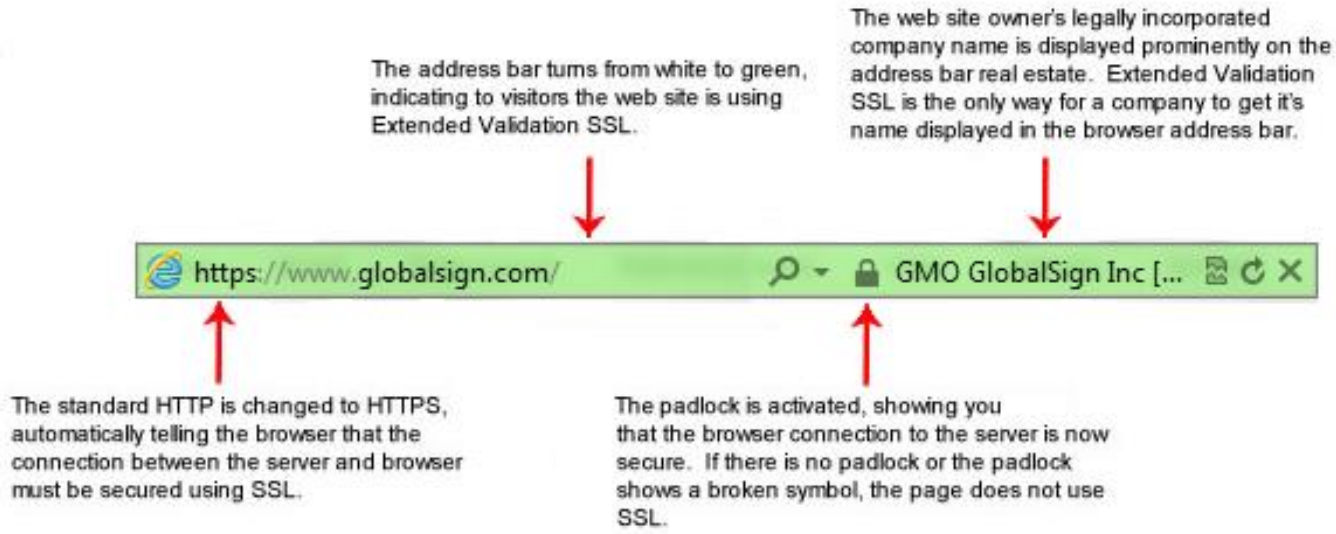




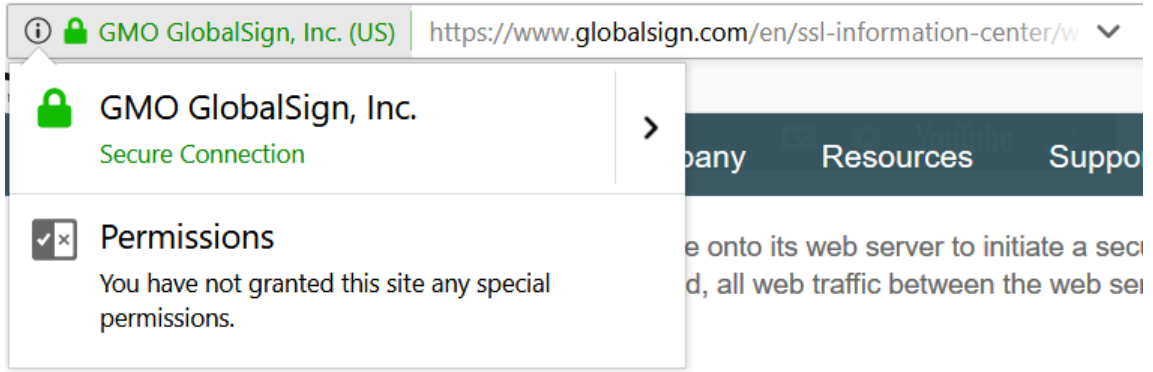


How does SSL look like?

- Extended Validation (EV)



https://www.globalsign.com/en/ssl-information-center/what-is-an-ssl-certificate/



SSH vs SSL

	SSH	SSL
Abbreviation	Secure Shell	Secure Socket Layer
Port	22	443
Application	Encrypting communications between computers.	Encrypting communications between browser and server.
Adopted by industry	Networking, Internal IT.	E-commerce, Banking, Social Media, Government, Healthcare.
Authenticated entities	Both entities are authenticated.	Browser authenticates server, but NOT vice versa.
Authentication type	Asymmetric key or ID/Password.	Asymmetric key.

Demo

Lab 6: Firewalls and SSH in Linux