Network Security - Week 5

Manuel E. Correia

DCC/FCUP

2025

Web Security Considerations

The World Wide Web is fundamentally a *client/server application* running over the internet and TCP/IP intranets

Web Security Considerations

The World Wide Web is fundamentally a *client/server application* running over the internet and TCP/IP intranets

Tailored security tools are necessary

- Web servers easy to configure and manage
- Web content increasingly easy to develop
- Underlying software extraordinarily complex
- Security flaws may be hidden

Web Security Considerations

A Web server can be exploited as a **launching pad** into the corporation/agency's entier computer complex

Web Security Considerations

A Web server can be exploited as a **launching pad** into the corporation/agency's entier computer complex

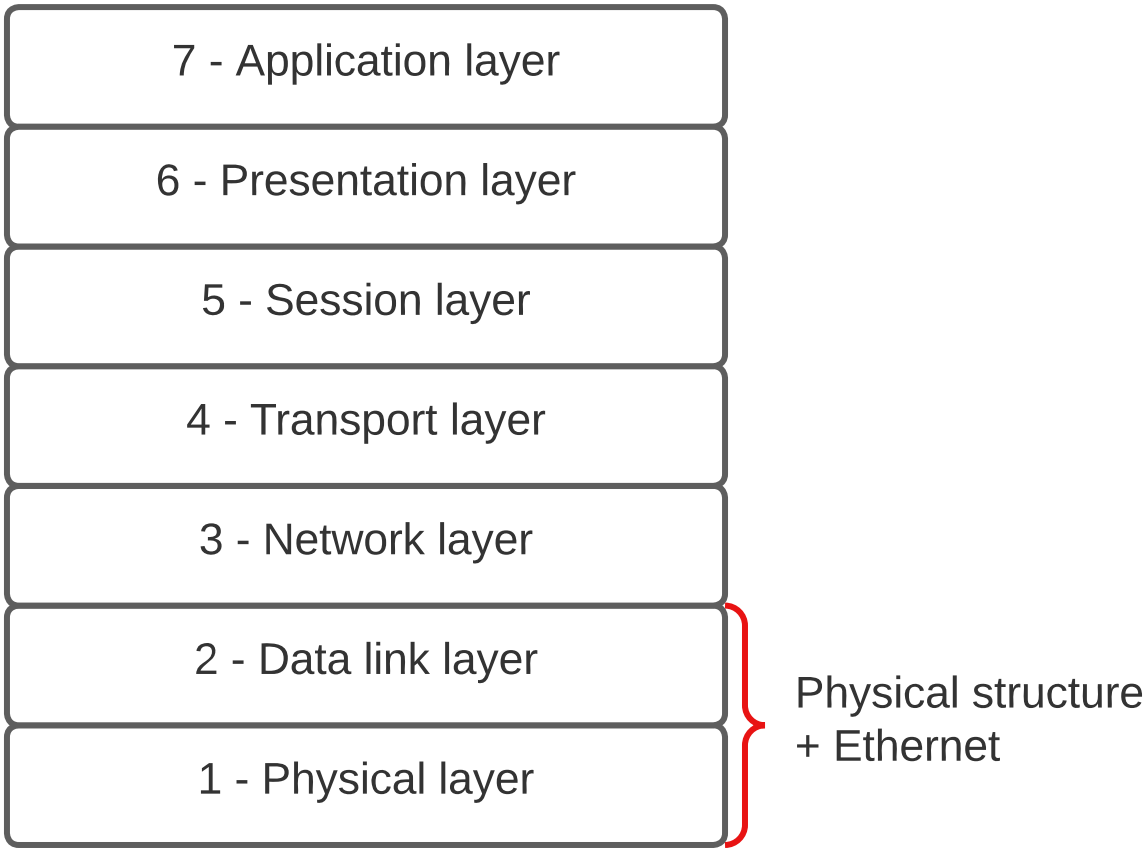
Casual/untrained users for web-based services

- Not aware of the security rists
- Don't have the tools/knowledge to take effective countermeasures...

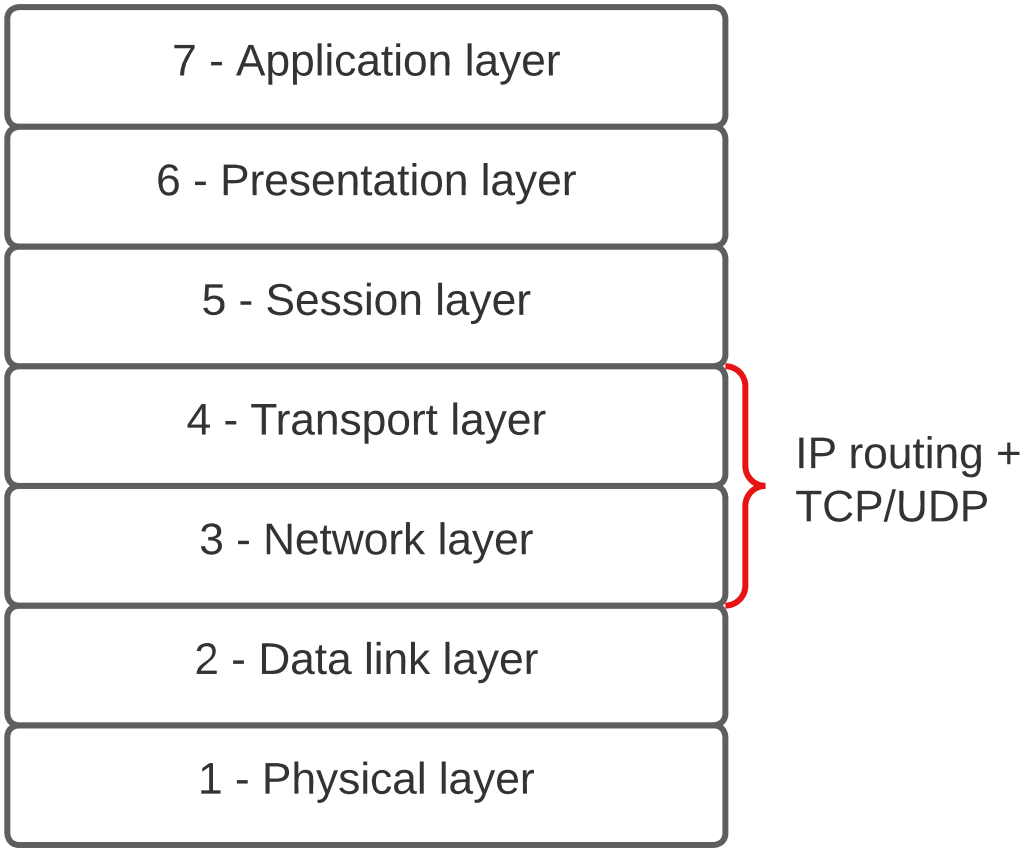
Web threats - a quick list

	Threats	Consequences	Countermeasures
Integrity	- Modification of user data - Trojan horse browser - Modification of memory - Modification of communication messages	- Loss of information - Compromise of machine - Vulnerability to all other threats	- Checksums - Erasure Codes - Message Authentication Codes
Confidentiality	- Eavesdropping on the network - Theft of information from the server - Theft of data from the client - Information about network configuration - Information about clients	- Privacy breaches - Loss of anonymity	- Encryption algorithms - Web proxies
Denial of Service	- Killing of user threads - Flooding machine with bogus requests - Filling up disk/memory - Isolating machine via DNS disruption	- Disruptive - Annoying - Preventing user from performing key tasks	**Very hard to prevent** - Traffic monitoring - Response plan
Authentication	- Impersonation of legitimate users - Man-in-the-Middle	- Misrepresentation of user - Covert eavesdrop channels - Covert message injection	- **What we learned last class!**

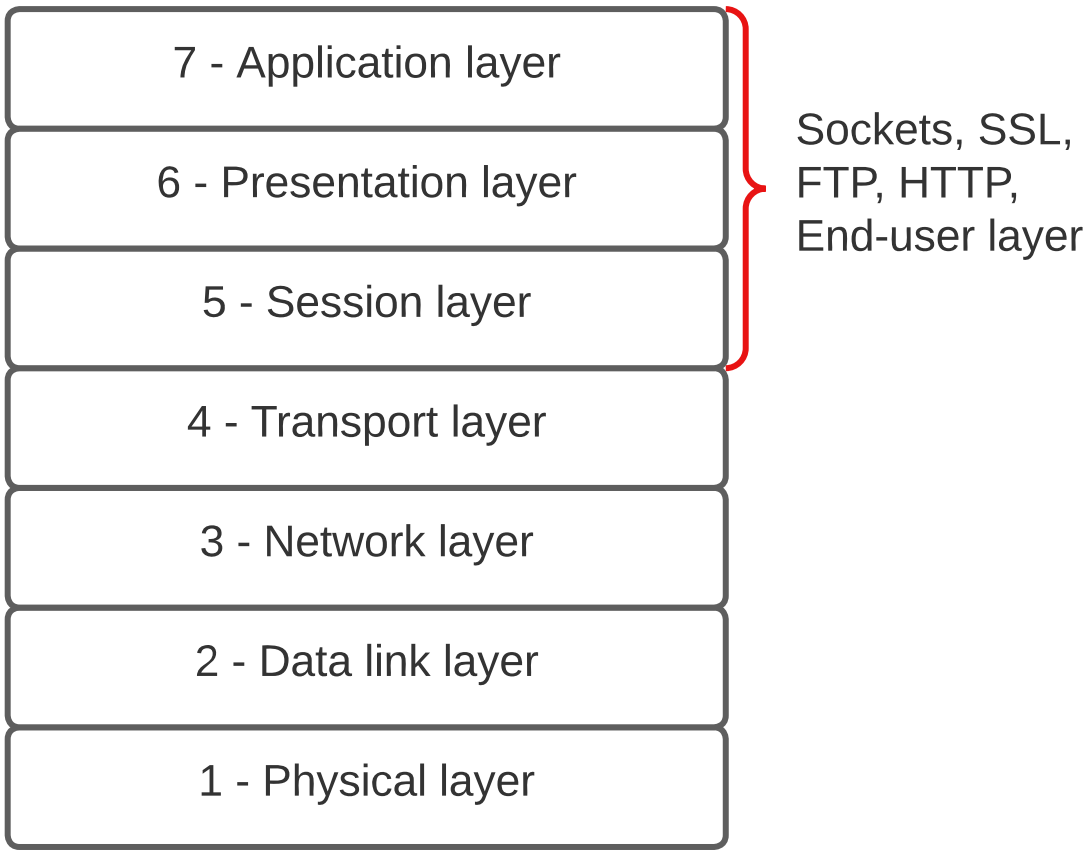
Open Systems Interconnection Layers



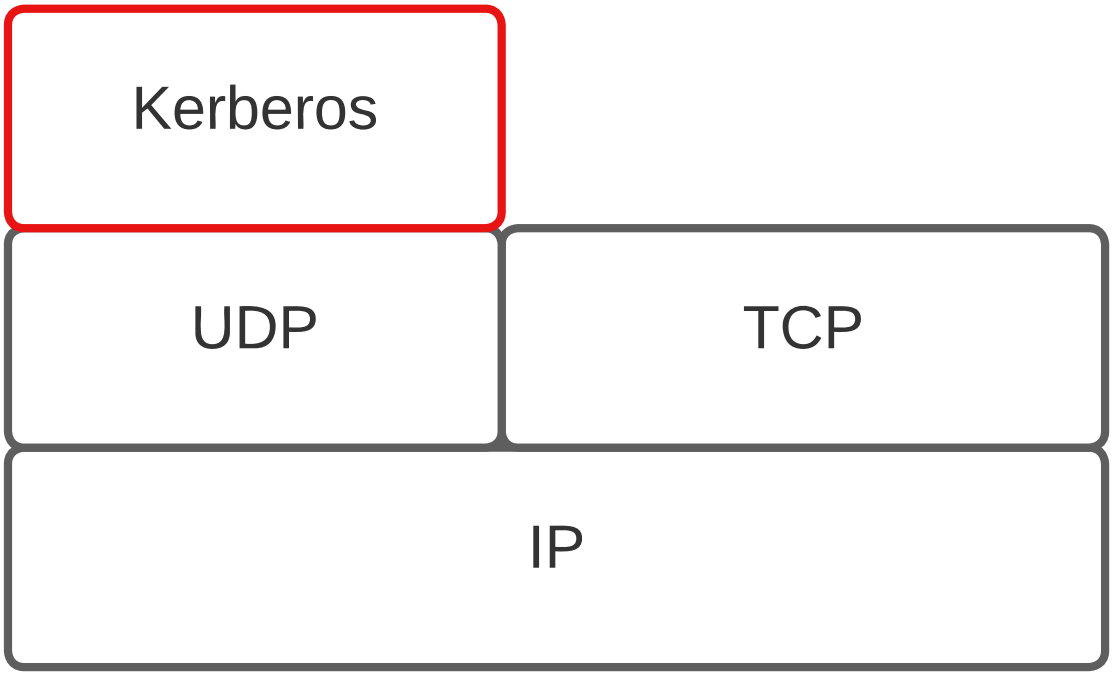
Open Systems Interconnection Layers



Open Systems Interconnection Layers

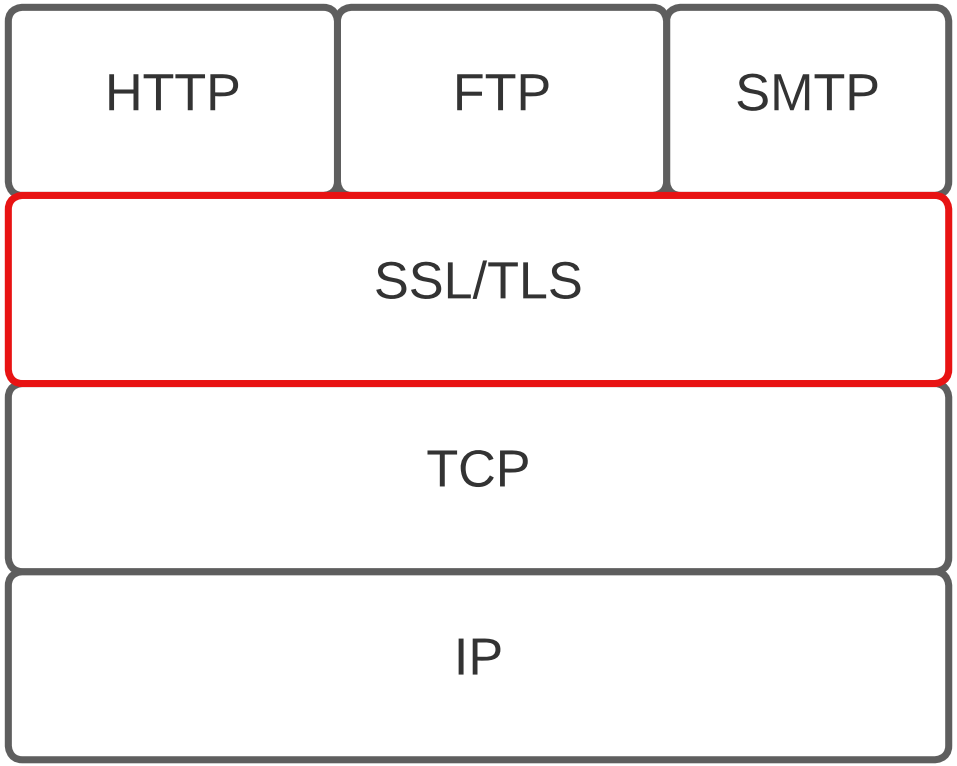


Security at the OSI Layers



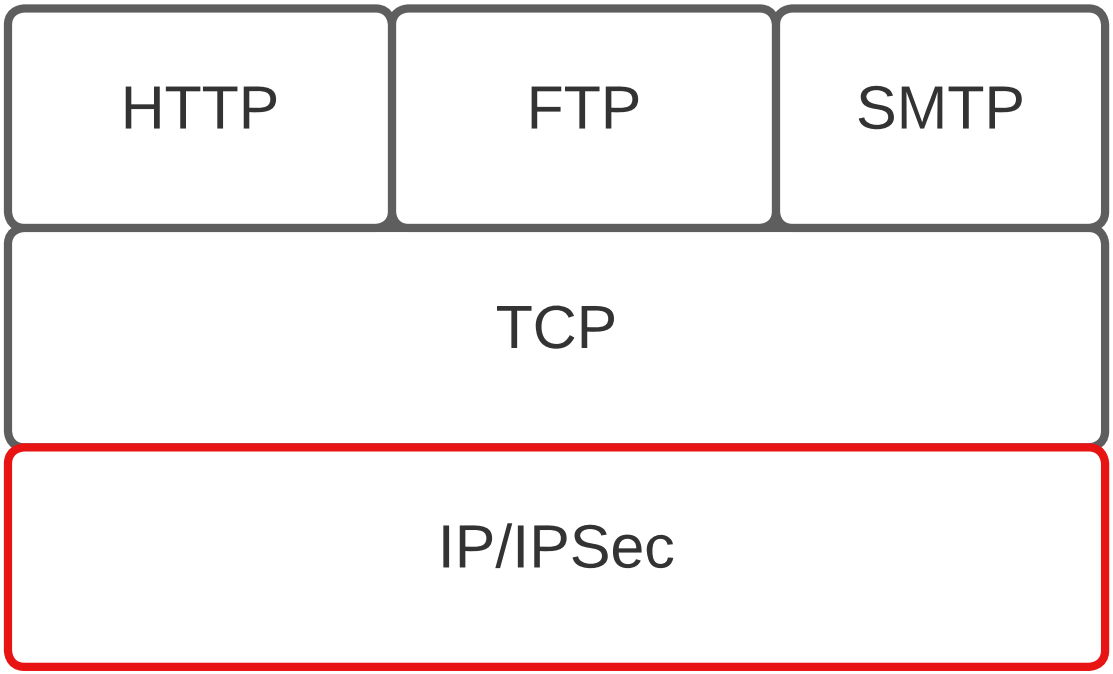
- Kerberos is at the application level - over UDP

Security at the OSI Layers



- SSL/TLS is a middleware between application and TCP

Security at the OSI Layers



- IPSec refines the IP protocol

What is SSL?

- Secure Sockets Layer (SSL) is the protocol used for the majority of secure internet transactions today

What is SSL?

- Secure Sockets Layer (SSL) is the protocol used for the majority of secure internet transactions today
- For instance, if you want to buy a book at *amazon.com* ...
  - You want to be sure you are talking with Amazon (authentication)

What is SSL?

- Secure Sockets Layer (SSL) is the protocol used for the majority of secure internet transactions today
- For instance, if you want to buy a book at *amazon.com* ...
  - You want to be sure you are talking with Amazon (authentication)
  - Credit card data must be protected (confidentiality + integrity)

What is SSL?

- Secure Sockets Layer (SSL) is the protocol used for the majority of secure internet transactions today
- For instance, if you want to buy a book at *amazon.com* ...
  - You want to be sure you are talking with Amazon (authentication)
  - Credit card data must be protected (confidentiality + integrity)
  - If payment is successful, Amazon does not care who you are
  - ... no need for mutual authentication

SSL and TLS

General-purpose system implemented as a set of protocols that **rely on TCP** to ensure message delivery guarantees

SSL and TLS

General-purpose system implemented as a set of protocols that **rely on TCP** to ensure message delivery guarantees

Implementation choices:

- Part of the underlying protocol suite
- Embedded in specific packages

SSL and TLS

General-purpose system implemented as a set of protocols that **rely on TCP** to ensure message delivery guarantees

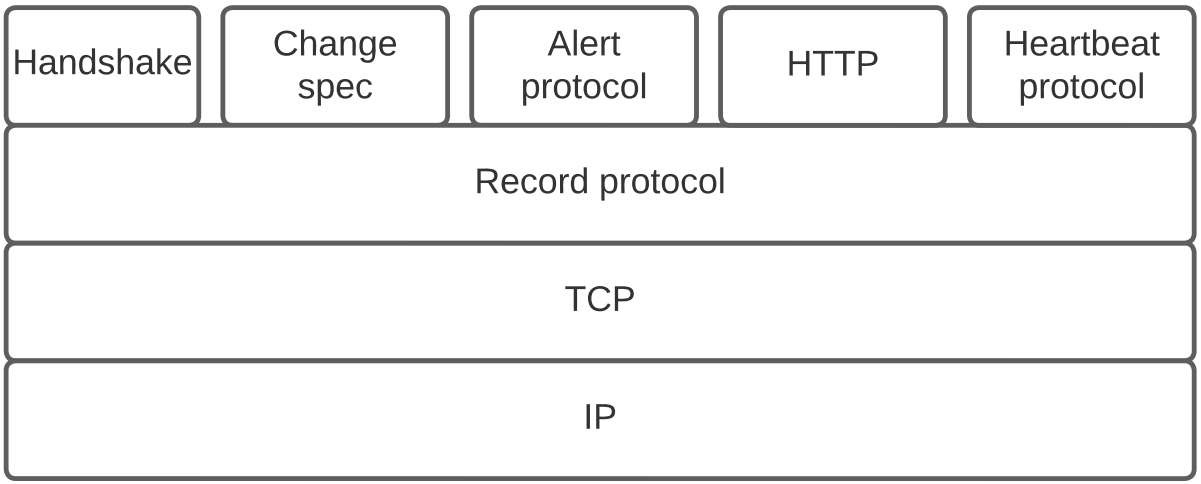
Implementation choices:

- Part of the underlying protocol suite
- Embedded in specific packages

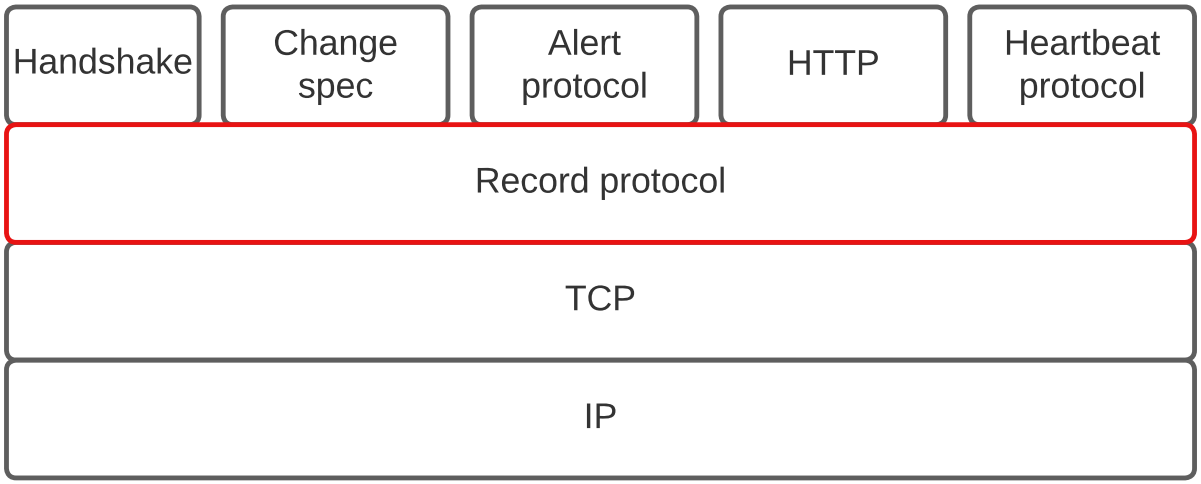
Transport Layer Security

- Evolved from the commercial protocol SSL
- Improved configurability, protocols, ...

SSL/TLS Protocol Stack



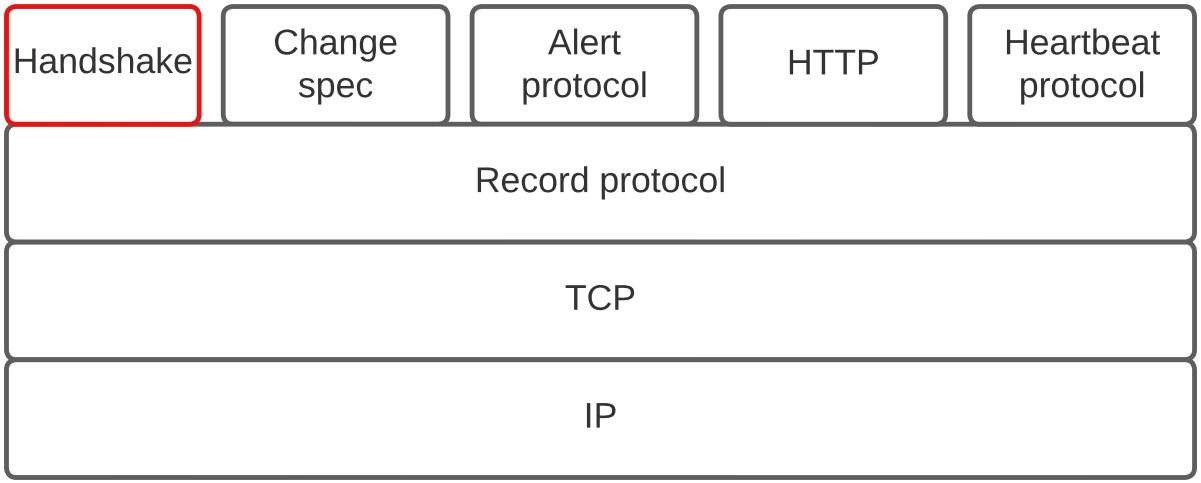
SSL/TLS Protocol Stack



Record Protocol

- Message Integrity and Confidentiality
- Uses key agreed on handshake

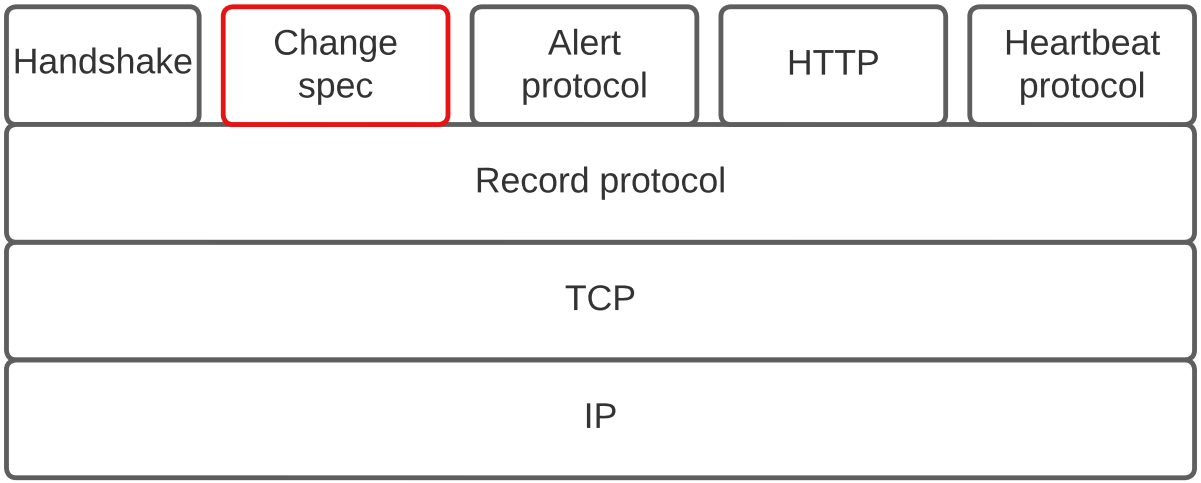
SSL/TLS Protocol Stack



Handshake

- Most complex protocol
- Crucial to establish a cryptographic key

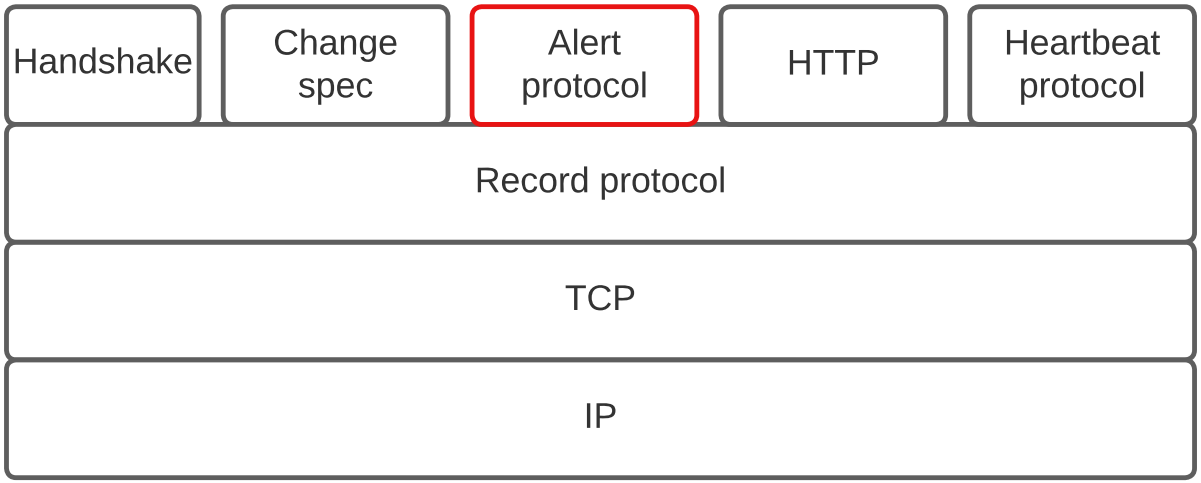
SSL/TLS Protocol Stack



Change Cipher Spec

- Single message
- Establishes agreed cipher specifications

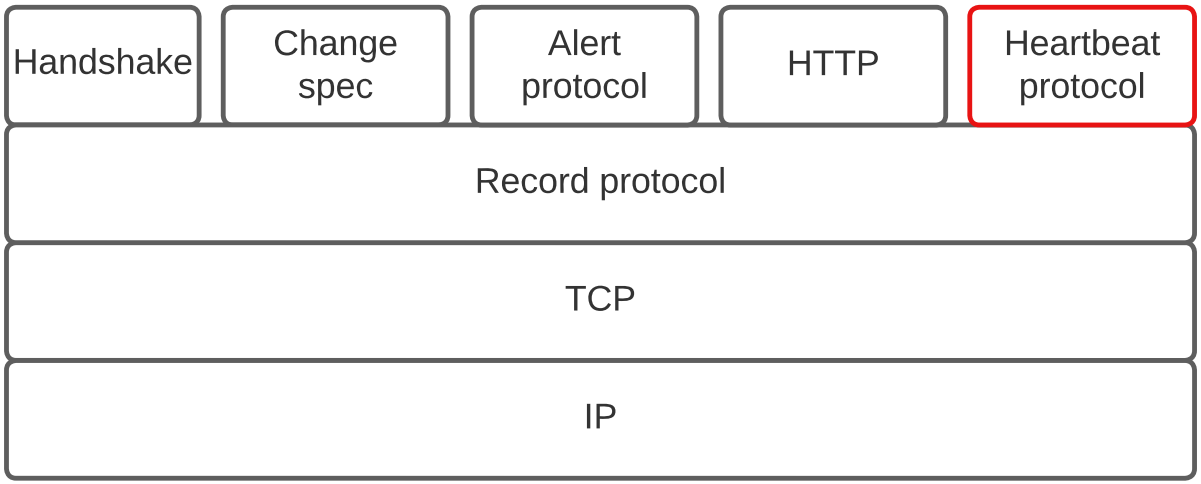
SSL/TLS Protocol Stack



Alert protocol

- TLS alerts
- Can provoke warning, or terminate connections

SSL/TLS Protocol Stack



Heartbeat protocol

- Pings regularly
- Prevents connection from shutting down

TLS Architecture

TLS connection

- A transport that provides a suitable type of service
- For TLS, such connections are peer-to-peer
- Connections are transient
- Every connection is associated with *one session*

TLS Architecture

TLS connection

- A transport that provides a suitable type of service
- For TLS, such connections are peer-to-peer
- Connections are transient
- Every connection is associated with *one session*

TLS session

- An association between a client and a server
- Created by the handshake protocol
- Defines a set of crypto security parameters, shared among multiple connections
- Used to avoid expensive negotiation stages, at the start of each connection

TLS Session State

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable

TLS Session State

- Session identifier
  - An arbitrary byte sequence chosen by the server to identify an active or resumable session state
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable

TLS Session State

- Session identifier
- Peer certificate
  - An X509.v3 certificate of the peer. Optional element of the state
- Compression method
- Cipher spec
- Master secret
- Is resumable

TLS Session State

- Session identifier
- Peer certificate
- Compression method
  - The algorithm used to compress data prior to encryption
- Cipher spec
- Master secret
- Is resumable

TLS Session State

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
  - Specified the bulk data encryption algorithm and a hash algorithm used for MAC computation; also defines cryptographic attributes, e.g. hash_size
- Master secret
- Is resumable

TLS Session State

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
  - A symmetric secret key shared between client and server
- Is resumable

TLS Session State

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable
  - A flag indicating whether the session can be used to initiate new connections

TLS Connection State

- Server and client randomness
- Server write MAC key
- Client write MAC key
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

TLS Connection State

- Server and client randomness
  - Byte sequences chosen by the server and client for each connection
- Server write MAC key
- Client write MAC key
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

TLS Connection State

- Server and client randomness
- Server write MAC key
  - Cryptographic key used to authenticate messages sent by the server
- Client write MAC key
  - Cryptographic key used to authenticate messages sent by the client
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

TLS Connection State

- Server and client randomness
- Server write MAC key
- Client write MAC key
- Server write key
  - Cryptographic key used to encrypt messages sent by the server
- Client write key
  - Cryptographic key used to encrypt messages sent by the client
- Initialization vectors
- Sequence numbers

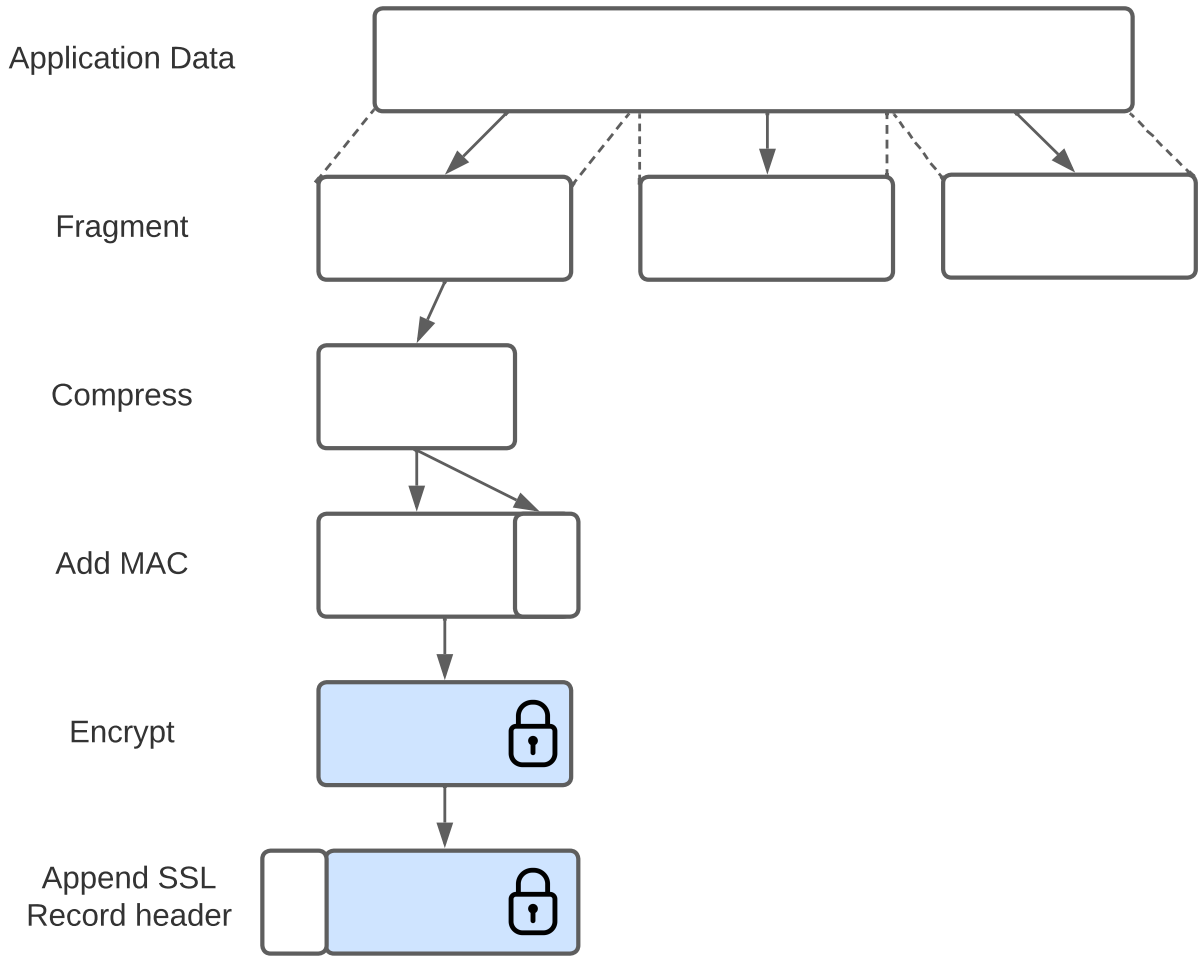
TLS Connection State

- Server and client randomness
- Server write MAC key
- Client write MAC key
- Server write key
- Client write key
- Initialization vectors
  - Values used in encryption to ensure freshness of ciphertexts, so that two encryptions of the same message do not produce the same ciphertext
  - Initialized by the handshake protocol
  - Final ciphertext of each record used as IV for the next one – chaining blocks
- Sequence numbers

TLS Connection State

- Server and client randomness
- Server write MAC key
- Client write MAC key
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers
  - Each party maintains sequence numbers for messages sent/received
  - Initialized at the cipher spec message
  - May not exceed $2^{64} - 1$

Record Protocol Operation



- Resulting unit transmitted via TCP
- Receiver decrypts, verifies, decompresses and reassembles

Handshake Protocol

- Most complex part of TLS
- Used before any application data is transmitted

Handshake Protocol

- Most complex part of TLS
- Used before any application data is transmitted
- Allows the server and client to:
  - Mutually authenticate

Handshake Protocol

- Most complex part of TLS
- Used before any application data is transmitted
- Allows the server and client to:
  - Mutually authenticate
  - Negotiate encryption and MAC algorithms

Handshake Protocol

- Most complex part of TLS
- Used before any application data is transmitted
- Allows the server and client to:
  - Mutually authenticate
  - Negotiate encryption and MAC algorithms
  - Negotiate cryptographic keys

Handshake Protocol

- Most complex part of TLS
- Used before any application data is transmitted
- Allows the server and client to:
  - Mutually authenticate
  - Negotiate encryption and MAC algorithms
  - Negotiate cryptographic keys
- Comprises a series of messages exchanged by client and server
- Exchange made on four stages

Handshake Protocol - 4 stages

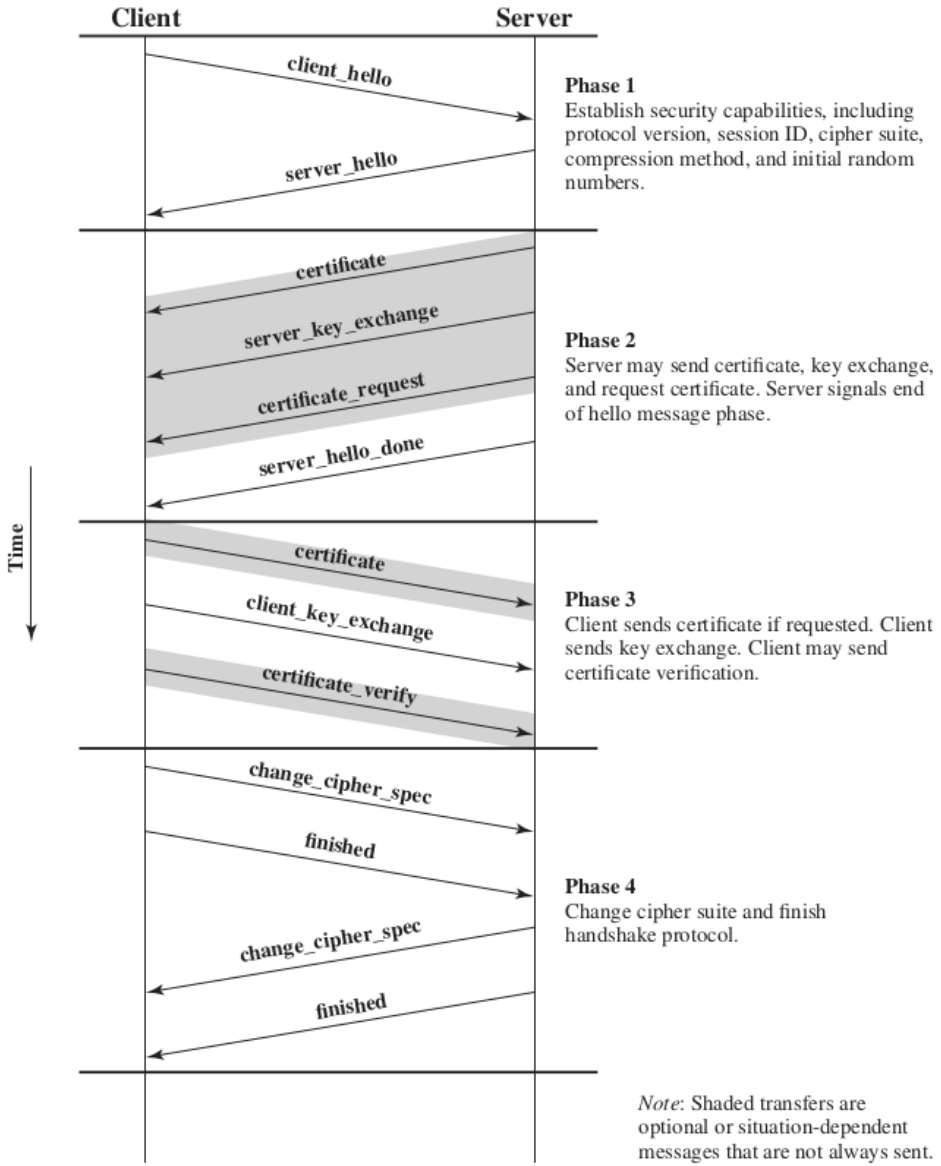


Figure 22.6 **Handshake Protocol Action**

Stage 1

- Hello!
- Here are the specs I can use
  - TLS version
  - Session ID
  - CipherSuite
  - Compression method

Handshake Protocol - 4 stages

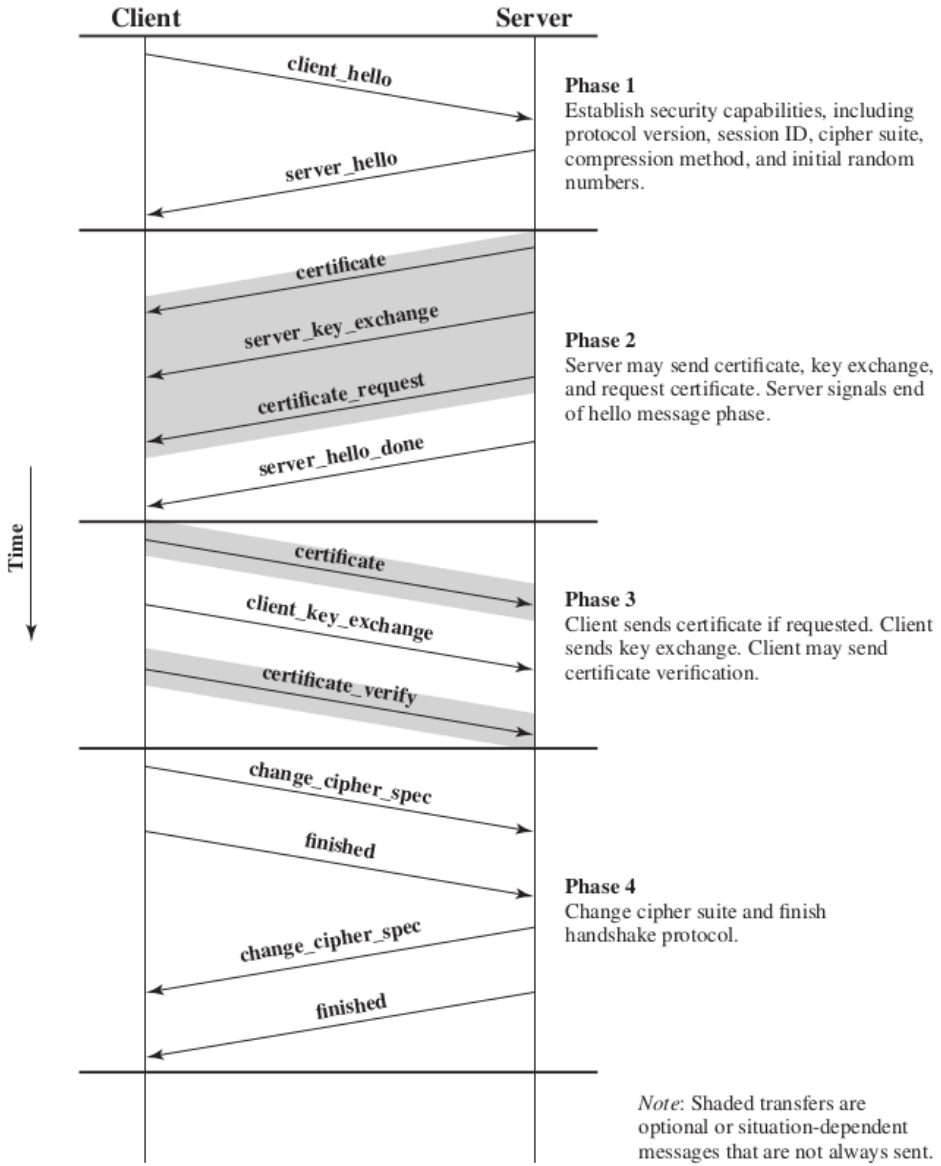


Figure 22.6 **Handshake Protocol Action**

Stage 2 and 3

- Certificate exchange
- Certificate verification
- Key agreement
  - RSA/Diffie-Hellman

Stage 4

- Client sends cipher specs
- Client sends a finished protected with authenticated encryption using new algorithms, keys and secrets
- Server verifies and does the same

Change Cipher Spec Protocol

- The simplest of the four
- A single message of a single byte. Value is either 0 or 1

Change Cipher Spec Protocol

- The simplest of the four
- A single message of a single byte. Value is either 0 or 1
- Sole purpose of this message is to cause pending state to be copied into the current state – used as confirmation message
- Hence updating the cipher suite in usage

Alert Protocol

- Conveys TLS-related alerts to peer entity
  - Alert messages are compressed and encrypted
  - Example of fatal alert: incorrect MAC
  - Example of non-fatal alert: close_notify (notifies the recipient that the sender will not send any more messages in this communication)

Alert Protocol

- Conveys TLS-related alerts to peer entity
  - Alert messages are compressed and encrypted
  - Example of fatal alert: incorrect MAC
  - Example of non-fatal alert: close_notify (notifies the recipient that the sender will not send any more messages in this communication)
- Each message consists of two bytes
- First byte refers to the severity; the second specifies

Alert Protocol

- Conveys TLS-related alerts to peer entity
  - Alert messages are compressed and encrypted
  - Example of fatal alert: incorrect MAC
  - Example of non-fatal alert: close_notify (notifies the recipient that the sender will not send any more messages in this communication)
- Each message consists of two bytes
- First byte refers to the severity; the second specifies
  - Fatal messages terminate the connection immediately
  - Other connections for that session may continue, but no additional connections are established

Heartbeat Protocol - P1

- A periodic signal is generated by hardware or software to indicate normal operation, or to synchronize with other parts of a system

Heartbeat Protocol - P1

- A periodic signal is generated by hardware or software to indicate normal operation, or to synchronize with other parts of a system
- Typically used to monitor the availability of a protocol entity – the name should be self-explanatory!

Heartbeat Protocol - P1

- A periodic signal is generated by hardware or software to indicate normal operation, or to synchronize with other parts of a system
- Typically used to monitor the availability of a protocol entity – the name should be self-explanatory!
- The heartbeat protocol runs on top of the TLS record protocol
- Relies on two message types

Heartbeat Protocol - P1

- A periodic signal is generated by hardware or software to indicate normal operation, or to synchronize with other parts of a system
- Typically used to monitor the availability of a protocol entity – the name should be self-explanatory!
- The heartbeat protocol runs on top of the TLS record protocol
- Relies on two message types
  - HEARTBEAT_REQUEST - prove you are alive
  - HEARTBEAT_RESPONSE - i am, indeed, alive

Heartbeat Protocol - P2

- Request includes payload length; payload; padding fields

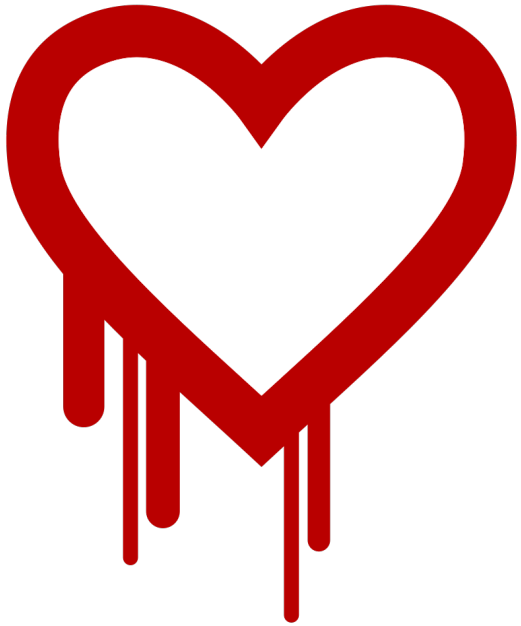
Heartbeat Protocol - P2

- Request includes payload length; payload; padding fields
- Response must include **an exact copy** of the received payload
  - Prevent replay attacks!

Heartbeat Protocol - P2

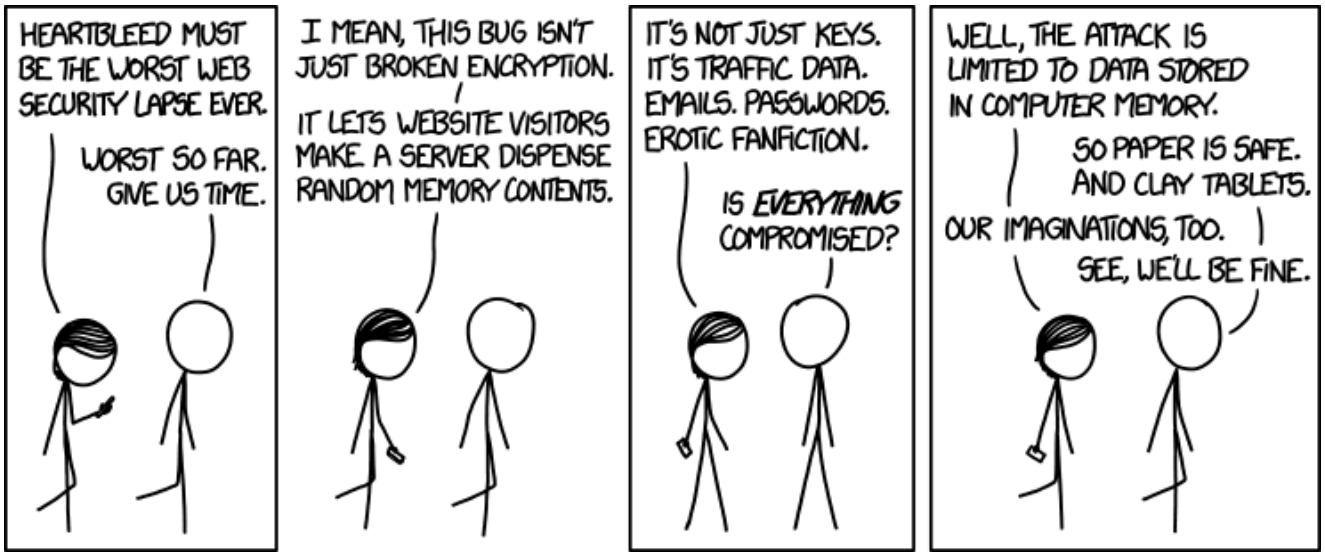
- Request includes payload length; payload; padding fields
- Response must include **an exact copy** of the received payload
  - Prevent replay attacks!
- Serves two main purposes
  - Assures the sender that the recipient is still alive, even if there is no regular activity in the underlying TCP connection
  - Generates activity across the connection during idle periods, which avoids closure by a firewall automatic mechanisms to disable idle connections

Heartbleed



- OpenSSL contains an open-source implementation of SSL/TLS
- A fatal flaw in OpenSSL, breaching privacy of log-in data
- Estimated victims: **two-thirds** of Web servers

Heartbleed - How it works



Heartbeat

- Send heartbeat message
- Extract; prep payload; send reply
- Response contains exactly the expected payload size
- Check for payload validity

Heartbleed

- Small payload disguised as big one
- Extract; prep (bad) payload; send reply
- Response contains **much** more than expected
- Gets TLS keys, cookies, passwords!

Network Security - Week 5

Manuel E. Correia

DCC/FCUP

2025