Computer Security Foundations
Week 12: Network Security Protocols

Bernardo Portela

L.EIC - 24

Web Security Considerations

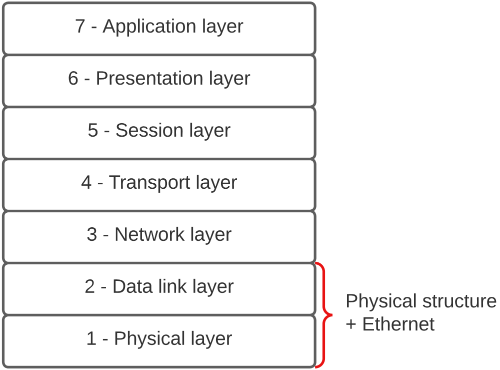
The World Wide Web is fundamentally a client/server application running over the internet and TCP/IP intranets

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

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

Open Systems Interconnection Layers



7 - Application layer

6 - Presentation layer

5 - Session layer

4 - Transport layer

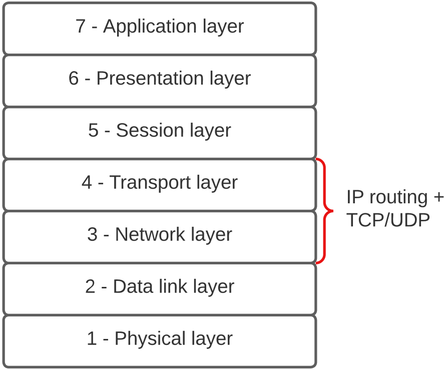
3 - Network layer

2 - Data link layer

1 - Physical layer

Physical structure
+ Ethernet

Open Systems Interconnection Layers



7 - Application layer

6 - Presentation layer

5 - Session layer

4 - Transport layer

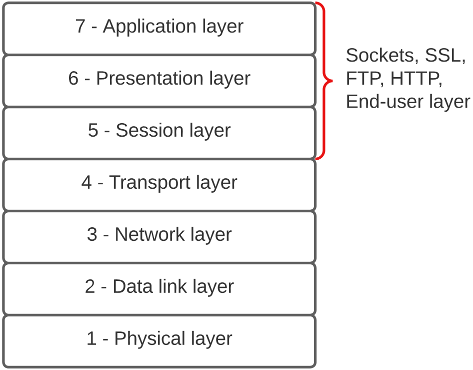
3 - Network layer

2 - Data link layer

1 - Physical layer

IP routing +
TCP/UDP

Open Systems Interconnection Layers



7 - Application layer

6 - Presentation layer

5 - Session layer

4 - Transport layer

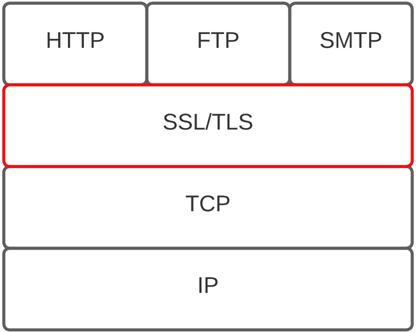
3 - Network layer

2 - Data link layer

1 - Physical layer

Sockets, SSL,
FTP, HTTP,
End-user layer

Security at the OSI Layers



HTTP FTP SMTP

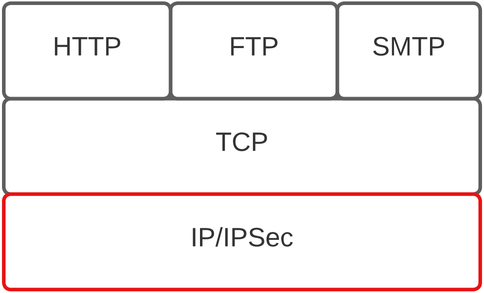
SSL/TLS

TCP

IP

- 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)
  - 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

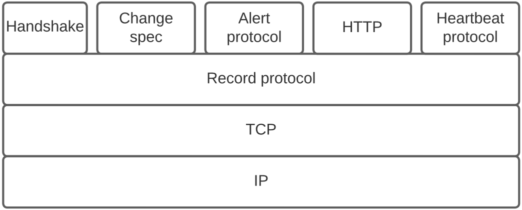
Implementation choices:

- 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



Handshake

Change spec

Alert protocol

HTTP

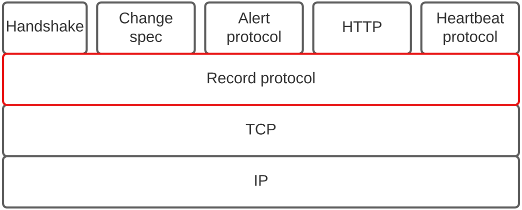
Heartbeat protocol

Record protocol

TCP

IP

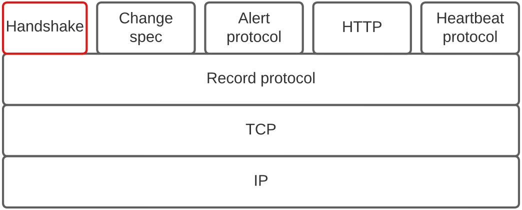
SSL/TLS Protocol Stack



Record Protocol

- Message Integrity and Confidentiality
- Uses key agreed on handshake

SSL/TLS Protocol Stack



Handshake

Change spec

Alert protocol

HTTP

Heartbeat protocol

Record protocol

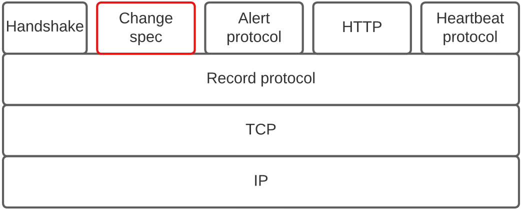
TCP

IP

Handshake

- Most complex protocol
- Crucial to establish a cryptographic key

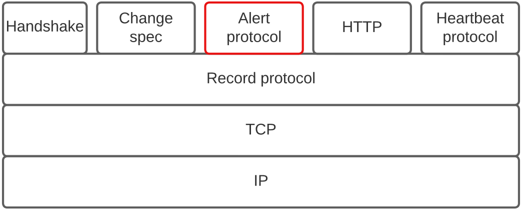
SSL/TLS Protocol Stack



Change Cipher Spec

- Single message
- Establishes agreed cipher specifications

SSL/TLS Protocol Stack



Handshake | Change spec | Alert protocol | HTTP | Heartbeat protocol

Record protocol

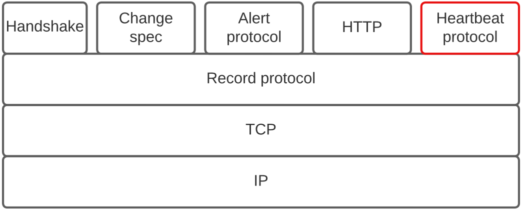
TCP

IP

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

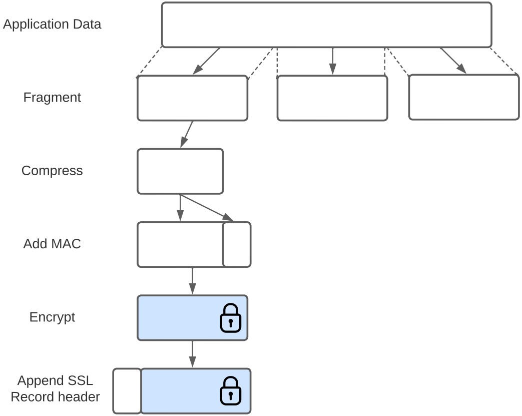
TLS connection

- 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

Record Protocol Operation

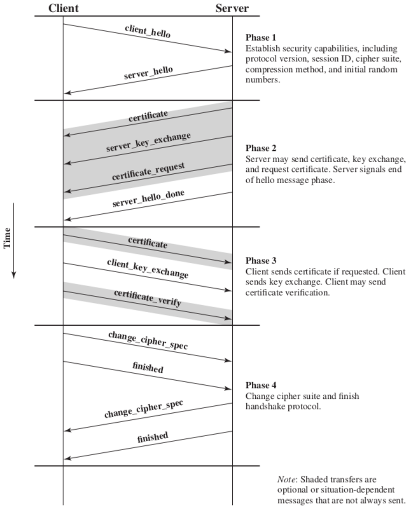


- Resulting unit transmitted via TCP
- Receiver decrypts, verifies, decompresses and reassembles

Handshake Protocol

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



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

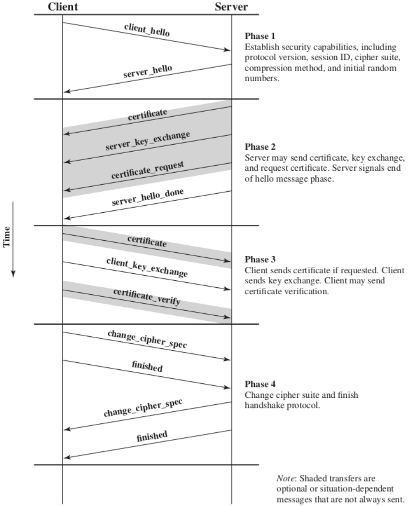
Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Figure 22.6 **Handshake Protocol Action**

Stage 1

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

Handshake Protocol - 4 stages



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

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
- 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)

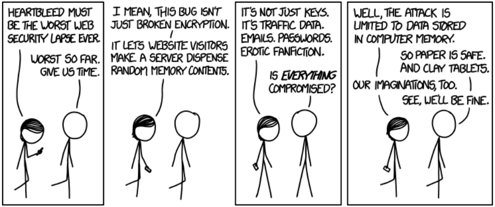
Alert Protocol

- 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
  - Fatal messages terminate the connection immediately
  - Other connections for that session may continue, but no additional connections are established

Heartbeat Protocol

- 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

Heartbleed


HEARTBLEED MUST BE THE WORST WEB SECURITY LAPSE EVER.
WORST SO FAR. GIVE US TIME.
I MEAN, THIS BUG ISN'T JUST BROKEN ENCRYPTION.
IT LETS WEBSITE VISITORS MAKE A SERVER DISPENSE RANDOM MEMORY CONTENTS.
IT'S NOT JUST KEYS. IT'S TRAFFIC DATA. EMAILS. PASSWORDS. EROTIC FANFICTION.
IS EVERYTHING COMPROMISED?
WELL, THE ATTACK IS LIMITED TO DATA STORED IN COMPUTER MEMORY.
SO PAPER IS SAFE. AND CLAY TABLETS.
OUR IMAGINATIONS, TOO.
SEE, WE'LL BE FINE.

- A fatal flaw in OpenSSL, breaching privacy of log-in data
- Estimated victims: **two-thirds** of Web servers

Heartbeat

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

Heartbleed

- Small payload disguised as big one
- Extract; prep (bad); send reply
- Response contains **much** more than expected

HTTPS

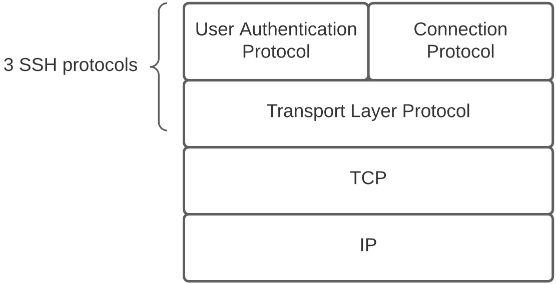
HTTP over SSL

- Combination of HTTP and SSL to implement secure communication between a Web browser and a Web server
- Build into all modern Web browsers
  - URL addresses begin with HTTPS://
- Agent acting as HTTP client also acts as the TLS client
- When HTTPS is used, the following elements are protected:
  - URL of requested document
  - Document contents
  - Contents of browser forms
  - Cookies sent from browser to server and vice-versa
  - Contents of HTTP header

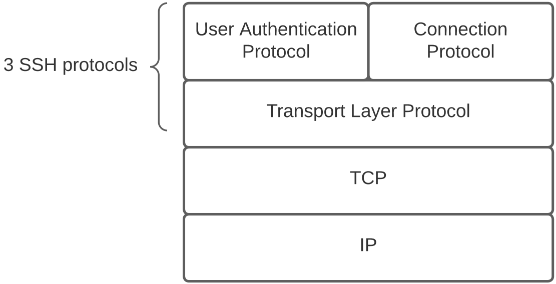
Secure Shell Protocol

- Originally developed for UNIX, now available on most OSs
- Provides an authenticated, encrypted path to the OS command line over the network
- Replacement for insecure utilities such as Telnet, rlogin, rsh
- Protects against spoofing attacks and modification of data
- The *de facto* method to access remote resources

SSH Protocol(s)

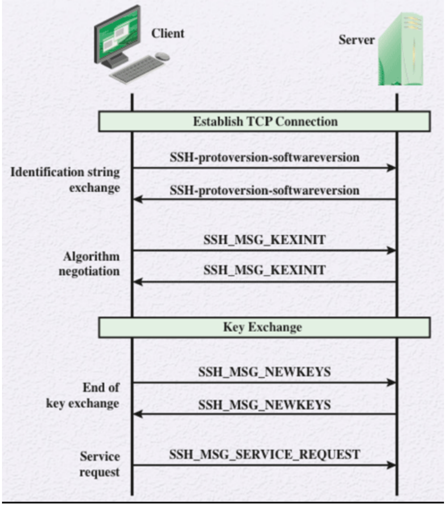


SSH Protocol(s)



- **Transport Layer Portocol** provides server authentication, confidentiality, and integrity.
- **User Authentication Protocol** authenticates the client-side user to the server
- **Connection Protocol** multiplexes the encrypted tunnel into several logical channels

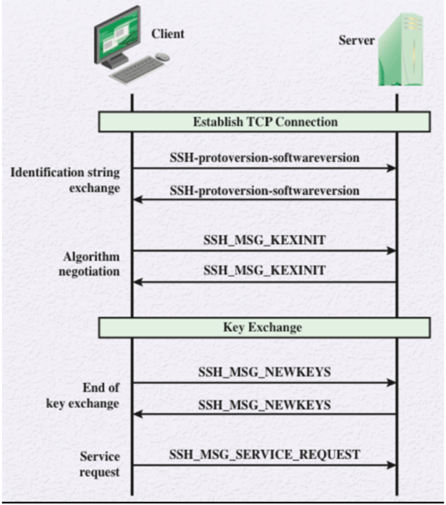
SSH Transport Layer Protocol



Multiple stages

1. Protocol and SW versions agreement
2. Supported algorithms exchanged
3. Key exchange finishes
4. Service ready to execute

SSH Transport Layer Protocol



Algorithm Agreement

- One (or more) algorithms must be listed
- Encryption algorithm used for confidentiality
- MAC algorithm used for data authentication
- Compression algorithm optional

SSH Authentication Methods

Public Key

- The client sends a message to the server that has the client's public key. Signed with the private key
- Upon receiving the message, the server check if the key is acceptable for authentication, and if the signature is correct

Password

Hostbased

SSH Authentication Methods

Public Key

Password

- The client sends a message containing a plaintext password, encrypted via the Transport Layer Protocol

Hostbased

SSH Authentication Methods

Public Key

Password

Hostbased

- Authentication is performed on the client's host rather than the client itself
- This method works by having the client send a signature created with the private key of its host
- Instead of verifying the client identity, the host identity is checked
- Provides group anonymity

SSH Connection Protocol

- SSH Connection Protocol runs on top of the Transport Layer Protocol
    - The secure authenticated connection, referred to as *tunnel*, is used by the Connection Protocol to multiplex a number of logical channels

SSH Connection Protocol

- SSH Connection Protocol runs on top of the Transport Layer Protocol
  - The secure authenticated connection, referred to as *tunnel*, is used by the Connection Protocol to multiplex a number of logical channels
- Channel mechanism
  - All types of communications using SSH supported via separate channels
  - Either side can open a channel
  - Channel type identifies the application/purpose of the channel

Channel Types

- **Session**
    - The remote execution of a program
    - Program may be a shell, an application such as file transfer, a system command, or a built-in subsystem

Channel Types

- **Session**
  - The remote execution of a program
  - Program may be a shell, an application such as file transfer, a system command, or a built-in subsystem
- **X11**
  - Refers to the X Window System, a computer software system and network protocol that provide a GUI for networked computers

Channel Types

- **Session**
  - The remote execution of a program
  - Program may be a shell, an application such as file transfer, a system command, or a built-in subsystem
- **X11**
  - Refers to the X Window System, a computer software system and network protocol that provide a GUI for networked computers
- **Forwarded-tcpip**
  - Remote port forwarding (from a remote computer to the local computer)
- **Direct-tcpip**
  - Local port forwarding (insecure TCP connection → SSH tunnel)

Key Takeaways

- TLS exists just over TCP
  - Under the application layer
  - Above the IP logistics

Key Takeaways

- TLS exists just over TCP
  - Under the application layer
  - Above the IP logistics

- TLS architecture
  - Record protocol takes care of encryption/authentication
  - Handshake establishes cryptographic material
  - Change cipher spec sets fixes the agreed specifications
  - Alert manages issues
  - Heartbeat maintains connection

Key Takeaways

- TLS exists just over TCP
  - Under the application layer
  - Above the IP logistics
- TLS architecture
  - Record protocol takes care of encryption/authentication
  - Handshake establishes cryptographic material
  - Change cipher spec sets fixes the agreed specifications
  - Alert manages issues
  - Heartbeat maintains connection
- HTTPS is simply the HTTP protocol layered over TLS connection

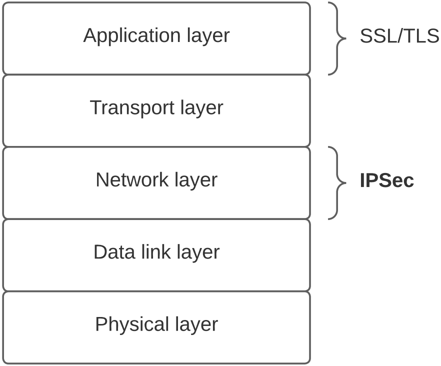
Key Takeaways

- TLS exists just over TCP
  - Under the application layer
  - Above the IP logistics
- TLS architecture
  - Record protocol takes care of encryption/authentication
  - Handshake establishes cryptographic material
  - Change cipher spec sets fixes the agreed specifications
  - Alert manages issues
  - Heartbeat maintains connection
- HTTPS is simply the HTTP protocol layered over TLS connection
- SSH follows a similar structure
  - Agreement + Key exchange
  - Multiple authentication methods
  - Different channel types

IP Security (IPSec)

- Various application security mechanisms exist
  - S/MIME, Kerberos, SSL/HTTPS
- Security (is often) a concern cross protocol layers
- One would like security implemented at the network layer
  - All applications can benefit from it, transparently!
- Authentication and encryption security features included in next-generation IPv6
- Also usable for good old IPv4

Network vs Application layer



- IPSec lives at the network layer
- It is transparent to applications

IPSec Applications

- Secure branch office connectivity over the internet
- Secure remote access over the internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security

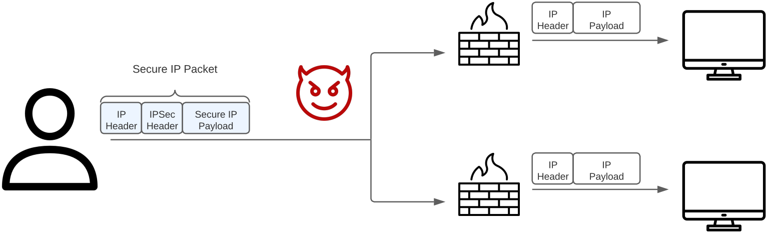
IPSec Applications

- Secure branch office connectivity over the internet
- Secure remote access over the internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security

Bottom line: IPSec thrives in applications where the same security is *always* necessary, and *the same security techniques* can be applied for all applications.

A Typical IPSec use case

VPN Security



- IPSec exists at the network layer
- From IP onward, everything is the same

Benefits of IPSec

- When implemented in a firewall or router, it provides strong security to all traffic crossing the perimeter
    - Clear context in which security is provided
    - See previous slide!

Benefits of IPSec

- When implemented in a firewall or router, it provides strong security to all traffic crossing the perimeter
  - Clear context in which security is provided
  - See previous slide!
- Transparent to applications and end users
  - Applications can be designed assuming secure channels
  - But that restricts flexibility...
  - What if the application wants to store encrypted messages?
  - Redundant security mechanisms.

Benefits of IPSec

- When implemented in a firewall or router, it provides strong security to all traffic crossing the perimeter
  - Clear context in which security is provided
  - See previous slide!
- Transparent to applications and end users
  - Applications can be designed assuming secure channels
  - But that restricts flexibility...
  - What if the application wants to store encrypted messages?
  - Redundant security mechanisms.
- Secures routing architecture
  - Authentication and integrity for all routing messages
  - Protects against attacks such as IP spoofing!

Scope - Two main functions

ESP

- Encapsulated Security Payload
- A combined function for authentication/encryption
- Key exchange function

AH

- Authentication Header
- An authentication-only function
- AH included in IPSecv3 for backward compatibility

Scope - Two main functions

ESP

- Encapsulated Security Payload
- A combined function for authentication/encryption
- Key exchange function

AH

- Authentication Header
- An authentication-only function
- AH included in IPSecv3 for backward compatibility

- VPNs want both authentication and encryption
- Specification is quite complex
- Numerous Request for Comments (RFCs)
    - 2401/4302/4303/4306

IPSec Architecture

1. Key Exchange Management
    - Internet Key Exchange (IKE) protocol

IPSec Architecture

1. Key Exchange Management
   - Internet Key Exchange (IKE) protocol
2. Two security header extensions
   - Authentication Header (AH)
   - Encapsulating Security Payload (ESP)

IPSec Architecture

1. Key Exchange Management
   - Internet Key Exchange (IKE) protocol
2. Two security header extensions
   - Authentication Header (AH)
   - Encapsulating Security Payload (ESP)
3. Two modes of operation
   - Transport mode - add information/security to the original packet
   - Tunnel mode - protect the original packet by encapsulating it into a new IP packet

Internet Key Exchange (IKE)

IKE has 2 stages:

- Phase 1 - IKE security association (SA)
- Phase 2 - IPSec security association

Internet Key Exchange (IKE)

IKE has 2 stages:

- Phase 1 - IKE security association (SA)
- Phase 2 - IPSec security association

- Phase 1 is comparable to SSL/TLS **session** - handshake; select cryptographic parameters; choose a master secret
- Phase 2 is comparable to SSL/TLS **connection** - ephemeral, uses Phase 1 to select encryption/MAC keys

Unlike SSL, necessity of two phases is not as obvious. If multiple Phase 2s do not occur, then it is **more** costly to have two phases!

Features of IKE Key Agreement

Algorithm used is (quite) a bit more complex than the Diffie-Hellman previously presented

1. Cookies thwart clogging attacks
    - Not the same as HTTP cookies!
2. Specifies the global parameters used by Diffie Hellman
3. Uses nonce to prevent against replay attacks
4. Allows Diffie-Hellman to exchange public key values
5. Authenticates Diffie-Hellman against man-in-the-middle attacks

IKE Phase 2

- Phase 1 establishes IKE Security Association
  - Defines parameters for authentication and key exchange
  - SSL Session

- **Phase 2** establishes IPSec Security Association
  - Services for secure communications
  - SSL Connection

IPSec Security Association (SA)

- A one-way connection between a sender and a receiver that affords security services to the traffic carried on it.
- In any IP packet, the SA is uniquely identified by the Destination Address in the IPv4 and IPv6 header, and the Security Parameters Index in the extension header (AH/ESP)

IPSec Security Association (SA)

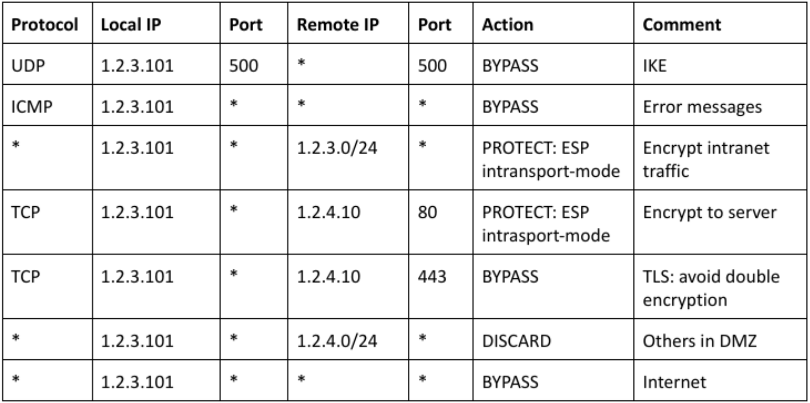
- A one-way connection between a sender and a receiver that affords security services to the traffic carried on it.
- In any IP packet, the SA is uniquely identified by the Destination Address in the IPv4 and IPv6 header, and the Security Parameters Index in the extension header (AH/ESP)

SA defined by three parameters

- Security Parameters Index (SPI) - A 32-bit unsigned integer assigned, only with local significance
- Security protocol identifier - Indicating whether it is an AH or ESP security association
- IP Destination Address - Address of the destination endpoint of the SA
  - May be an end-user system, or a network (firewall / router)

Security Policy Database (SPD)

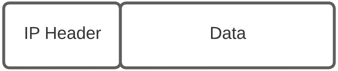
- Means by which IP traffic relates to SAs
- Entries define subset of IP traffic and point to SAs
- Allows for complex system configurations



Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intrasport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

Review - Internet Protocol

An IP datagram is something of the form



IP Header

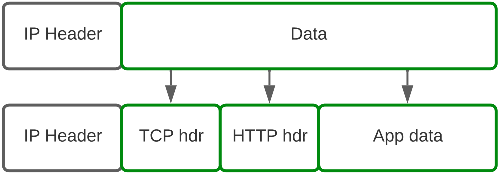
Data

- Routers *must see the destination address in the IP header*
  - They have to route the packet
- Some of its fields change as the packet is forwarded
- Routers don't have access to the session key...
- ... So we *can't encrypt* the IP header

Upper Layers

Remember that Web traffic is iteratively encapsulating data

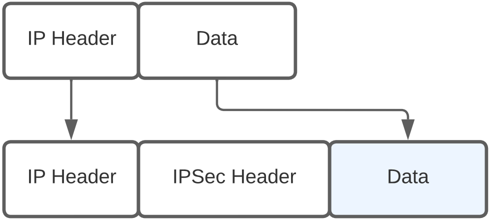
- IP encapsulates TCP
- TCP encapsulates HTTP



- IP data includes TCP header, HTTP header, ...

Two Execution Modes

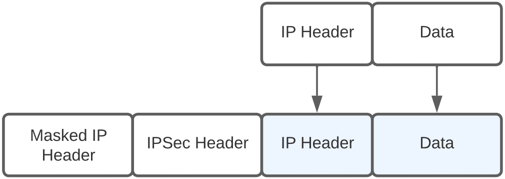
Transport Mode



- Designed for *host-to-host* communication
- Very efficient
  - Minimal extra header
- Original header remains
  - An attacker can see who is communicating

Two Execution Modes

Tunnel Mode



- Designed for *firewall-to-firewall* traffic
- Original IP packet encapsulated in IPSec
- Original IP header not visible to attacker
  - IP header now refers to the firewall
  - Attacker *can see* which firewalls are communicating
  - Attacker *cannot know* which hosts within that domain are talking

Going back to the IPSec Algorithms

A quick recap from a couple of slides ago...

- AH - Authentication header
  - Integrity only (**no confidentiality**)
  - Protect everything beyond IP header and some header fields

- ESP - Encapsulating Security Payload
  - Integrity and Confidentiality **both required**
  - Protects everything beyond IP header

The purpose of an Authenticated Header

- AH protects *immutable* fields in the IP header
  - Cannot protect all header fields
  - e.g. TTL changes

[1] C. Kaufman, R. Perlman, and M. Speciner, Network Security, second edition, Prentice Hall, 2002.

The purpose of an Authenticated Header

- AH protects *immutable* fields in the IP header
  - Cannot protect all header fields
  - e.g. TTL changes

Why does AH exist, then?

- ESP does not protect the integrity of the IP header
- Encrypting data prevents the firewall from inspecting its contents

[1] C. Kaufman, R. Perlman, and M. Speciner, Network Security, second edition, Prentice Hall, 2002.

The purpose of an Authenticated Header

- AH protects *immutable* fields in the IP header
  - Cannot protect all header fields
  - e.g. TTL changes

Why does AH exist, then?

- ESP does not protect the integrity of the IP header
- Encrypting data prevents the firewall from inspecting its contents
- The story goes that *"someone from Microsoft gave an impassioned speech about how AH was useless ..." and "... everyone in the room looked around and said, Hmm. He's right, and we hate AH also, but if it annoys Microsoft let's leave it in since we hate Microsoft more than we hate AH"*[1]

[1] C. Kaufman, R. Perlman, and M. Speciner, Network Security, second edition, Prentice Hall, 2002.

SSL/TLS vs IPSec - P1

SSL/TLS

- Lives at the socket layer (user space)
- Encryption, integrity, authentication, etc.
- Relatively simple
- Elegant(-ish) specification

SSL/TLS vs IPSec - P1

SSL/TLS

- Lives at the socket layer (user space)
- Encryption, integrity, authentication, etc.
- Relatively simple
- Elegant(-ish) specification

IPSec

- Lives at the network layer (OS space)
- Encryption, integrity, authentication, etc.
- Very complex!

SSL/TLS vs IPSec - P2

- IPSec: OS must be aware, but not the applications
- SSL/TLS: Applications must be aware, but not the OS
- TLS designed for application-level security
  - Easier to adapt to individual application needs
  - Does not protect from IP spoofing (lower layer)!
- IPSec often used in VPNs
  - Secure tunnel
  - All communications must be confidential and authenticated!
- Reluctance to retrofit applications for SSL
- IPSec not widely deployed (complexity is a major factor)

SSL/TLS vs IPSec - P2

- IPSec: OS must be aware, but not the applications
- SSL/TLS: Applications must be aware, but not the OS
- TLS designed for application-level security
  - Easier to adapt to individual application needs
  - Does not protect from IP spoofing (lower layer)!
- IPSec often used in VPNs
  - Secure tunnel
  - All communications must be confidential and authenticated!
- Reluctance to retrofit applications for SSL
- IPSec not widely deployed (complexity is a major factor)

Internet is less secure than it could be!

Computer Security Foundations
Week 12: Network Security Protocols

Bernardo Portela

L.EIC - 24