Network Security - Week 6

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

SSL/TLS Applications

- Secure e-commerce using SSL/TLS
- Client authentication not needed until client decides to buy something, which triggers the authentication stage

SSL/TLS Applications

- Secure e-commerce using SSL/TLS
- Client authentication not needed until client decides to buy something, which triggers the authentication stage
- SSL provides secure channel for sending credit card information

SSL/TLS Applications

- Secure e-commerce using SSL/TLS
- Client authentication not needed until client decides to buy something, which triggers the authentication stage
- SSL provides secure channel for sending credit card information
- Client authenticated using credit card information, merchant bears (most of) the risk
- Widely deployed

SSL/TLS Applications

- Secure e-commerce using SSL/TLS
- Client authentication not needed until client decides to buy something, which triggers the authentication stage
- SSL provides secure channel for sending credit card information
- Client authenticated using credit card information, merchant bears (most of) the risk
- Widely deployed
- Secure remote login / **Secure Shell**
- Authenticated, encrypted path to the OS over the network

HTTPS

HTTP over SSL

- Combination of HTTP and SSL to implement secure communication between a Web browser and a Web server

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://

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

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

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

HTTPS

HTTP over SSL

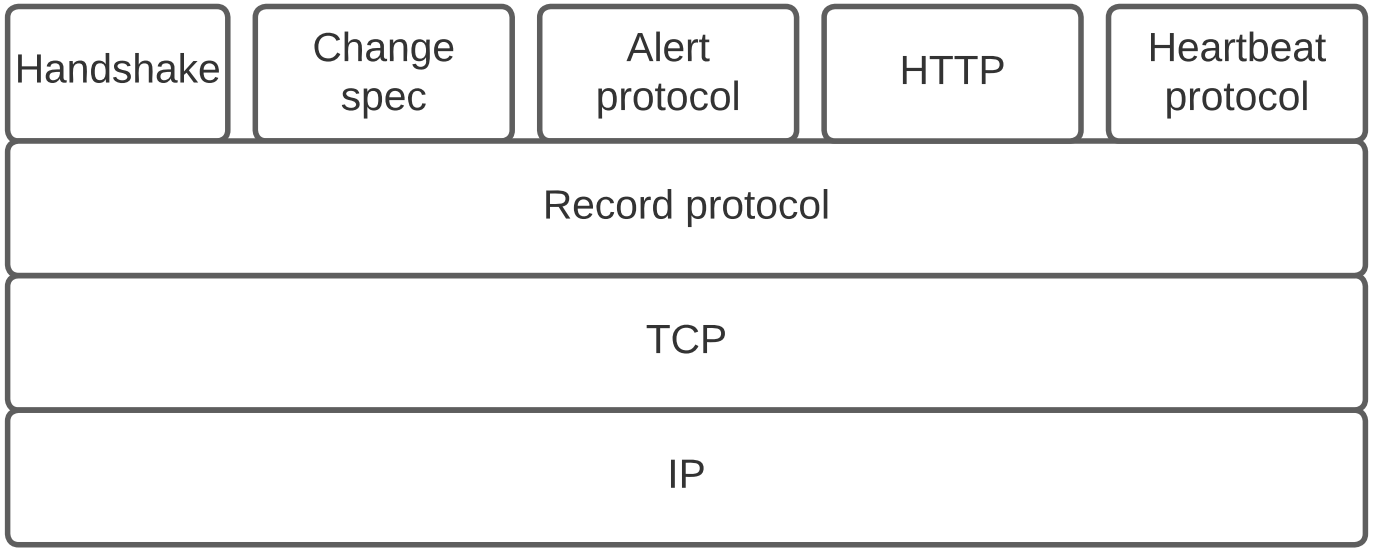
- Combination of HTTP and SSL to implement secure communication between a Web browser and a Web server
- Build into all modern Web browsers
  - URL addresses begin with HTTPS://
- Agent acting as HTTP client also acts as the TLS client
- When HTTPS is used, the following elements are protected:
  - URL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

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

First handshake, then HTTP through TLS



Layered connection

- Connection begins with a TLS ClientHello, which triggers the TLS handshake
- When it finishes, the client sends the first HTTP request
- All data sent as TLS application data

Closing Connections

- An HTTP client or server can indicate the closing of a connection by including the line CONNECTION: CLOSE in an HTTP record

Closing Connections

- An HTTP client or server can indicate the closing of a connection by including the line CONNECTION: CLOSE in an HTTP record
- The closure of an HTTPS connection requires TLS to close the connection with the peer TLS entity on the remote side – closing the underlying TCP connection

Closing Connections

- An HTTP client or server can indicate the closing of a connection by including the line CONNECTION: CLOSE in an HTTP record
- The closure of an HTTPS connection requires TLS to close the connection with the peer TLS entity on the remote side – closing the underlying TCP connection
- TLS implementation must initiate an exchange of closure alerts before closing a connection

Closing Connections

- An HTTP client or server can indicate the closing of a connection by including the line CONNECTION: CLOSE in an HTTP record
- The closure of an HTTPS connection requires TLS to close the connection with the peer TLS entity on the remote side – closing the underlying TCP connection
- TLS implementation must initiate an exchange of closure alerts before closing a connection
- TLS session ensures that cryptographic parameters are kept (avoiding expensive negotiations)

Wrap up

SSL and TLS

- Architecture over classical network layers

Wrap up

SSL and TLS

- Architecture over classical network layers
- TLS connections are ephemeral, sessions allow for multiple connections

Wrap up

SSL and TLS

- Architecture over classical network layers
- TLS connections are ephemeral, sessions allow for multiple connections
- Protocols rely on TLS for secure communication

Wrap up

SSL and TLS

- Architecture over classical network layers
- TLS connections are ephemeral, sessions allow for multiple connections
- Protocols rely on TLS for secure communication
- HTTPS uses TLS over HTTP

Wrap up

SSL and TLS

- Architecture over classical network layers
- TLS connections are ephemeral, sessions allow for multiple connections
- Protocols rely on TLS for secure communication
- HTTPS uses TLS over HTTP
- Attacks at the TLS layer - Heartbleed

Wrap up

SSL and TLS

- Architecture over classical network layers
- TLS connections are ephemeral, sessions allow for multiple connections
- Protocols rely on TLS for secure communication
- HTTPS uses TLS over HTTP
- Attacks at the TLS layer - Heartbleed

Secure Shell Protocol

- Originally developed for UNIX, now available on most OSs

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

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

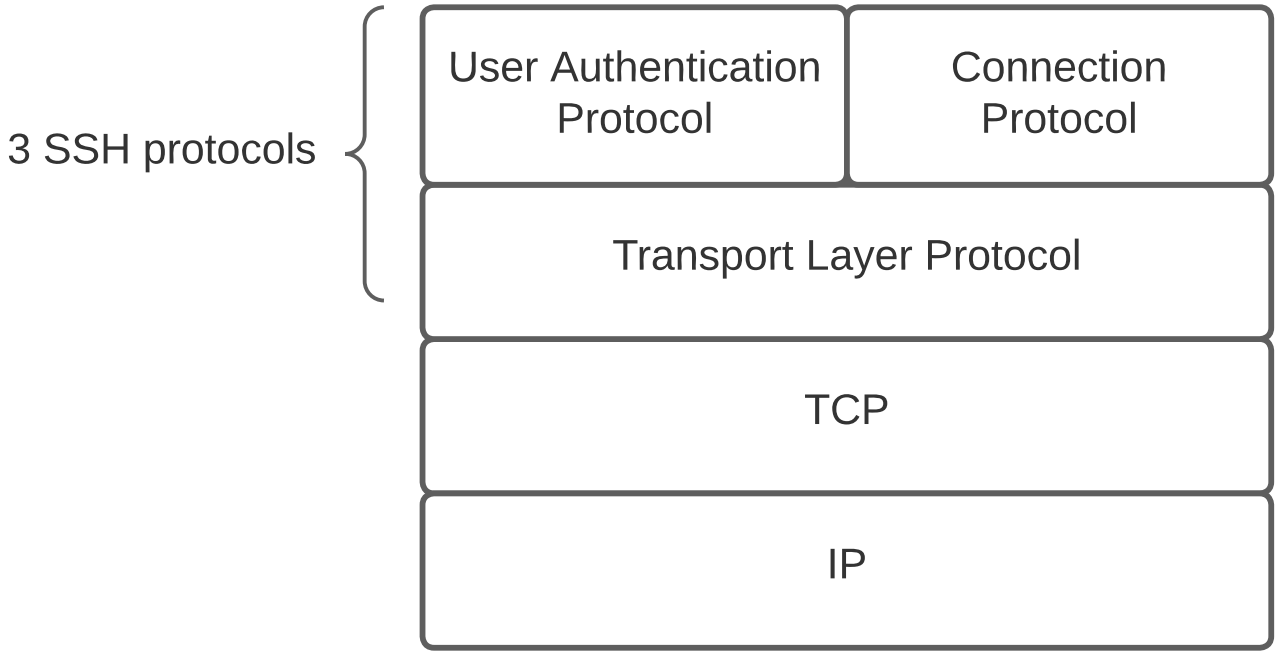
Secure Shell Protocol

- 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

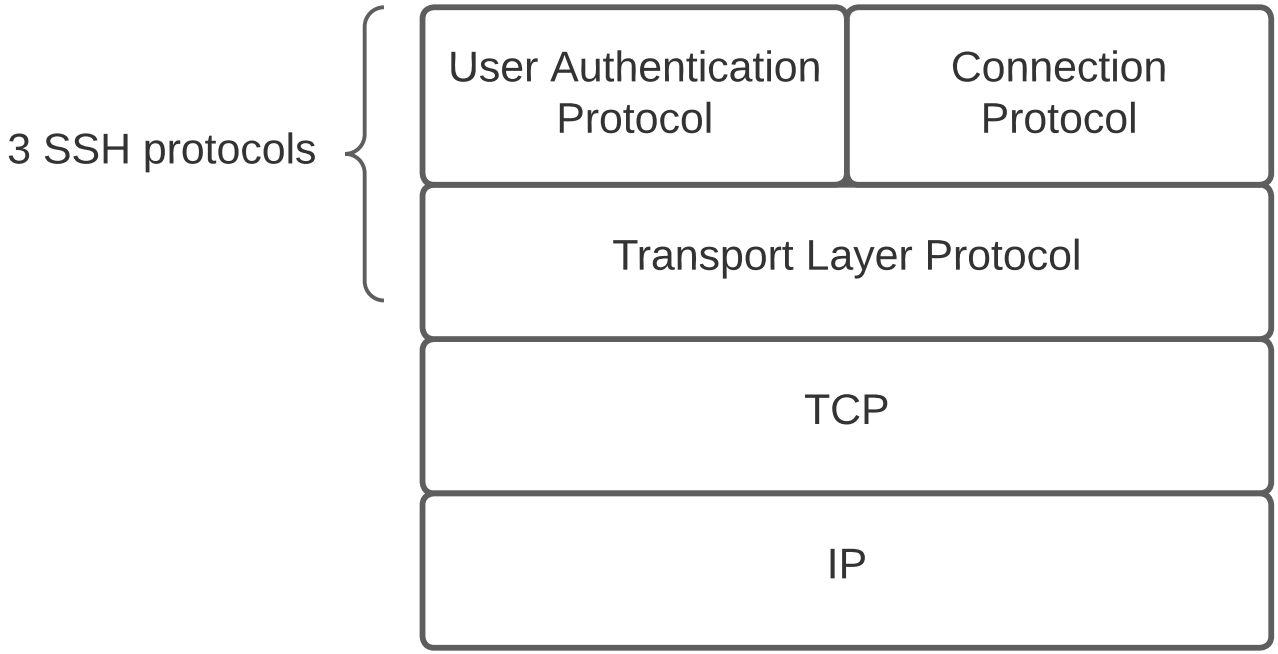
Secure Shell Protocol

- 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)

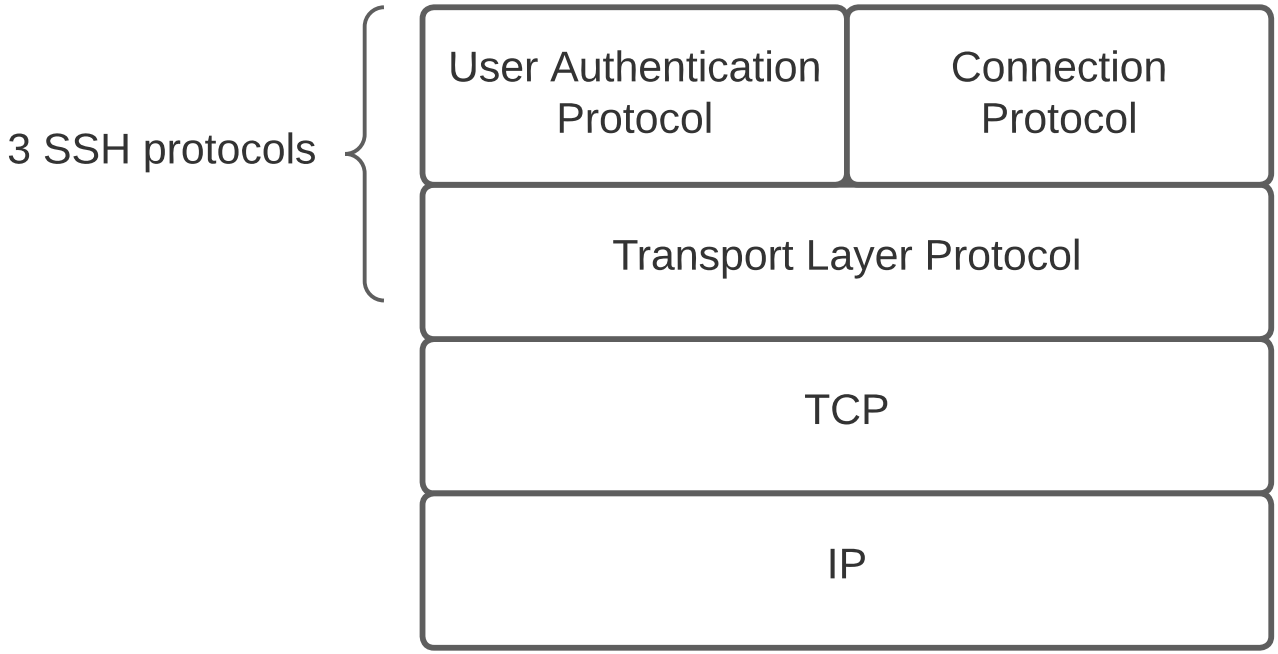


SSH Protocol(s)



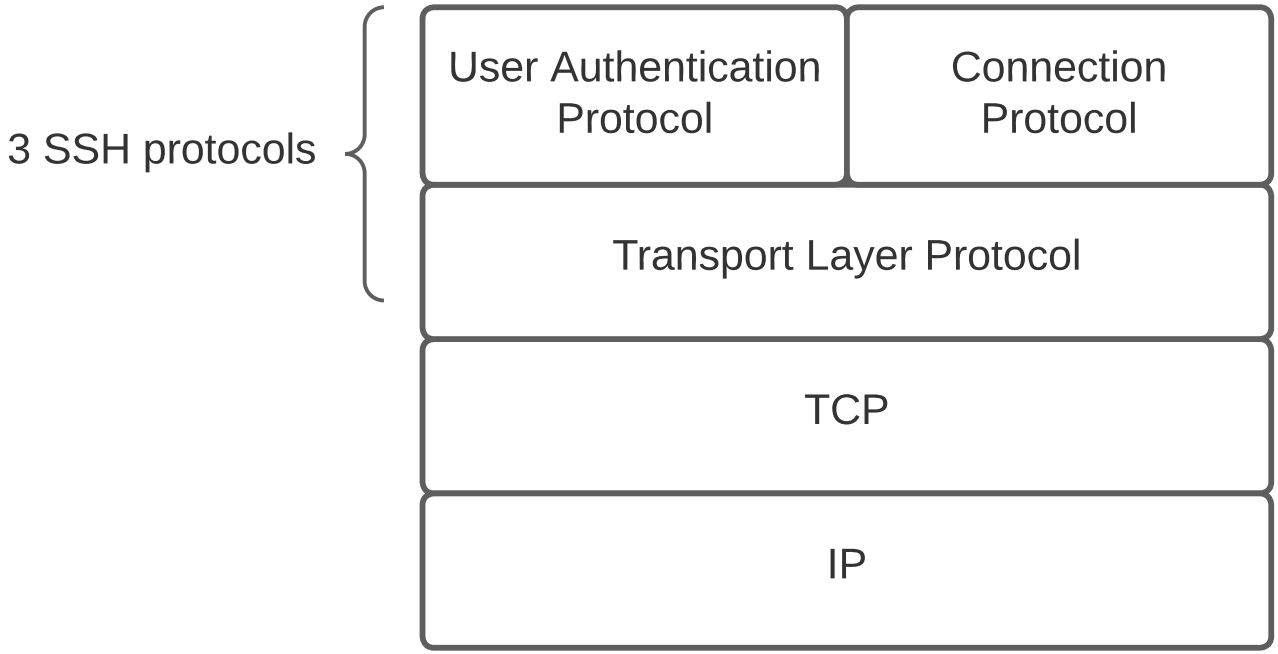
- **Transport Layer Protocol** provides server authentication, confidentiality, and integrity.

SSH Protocol(s)



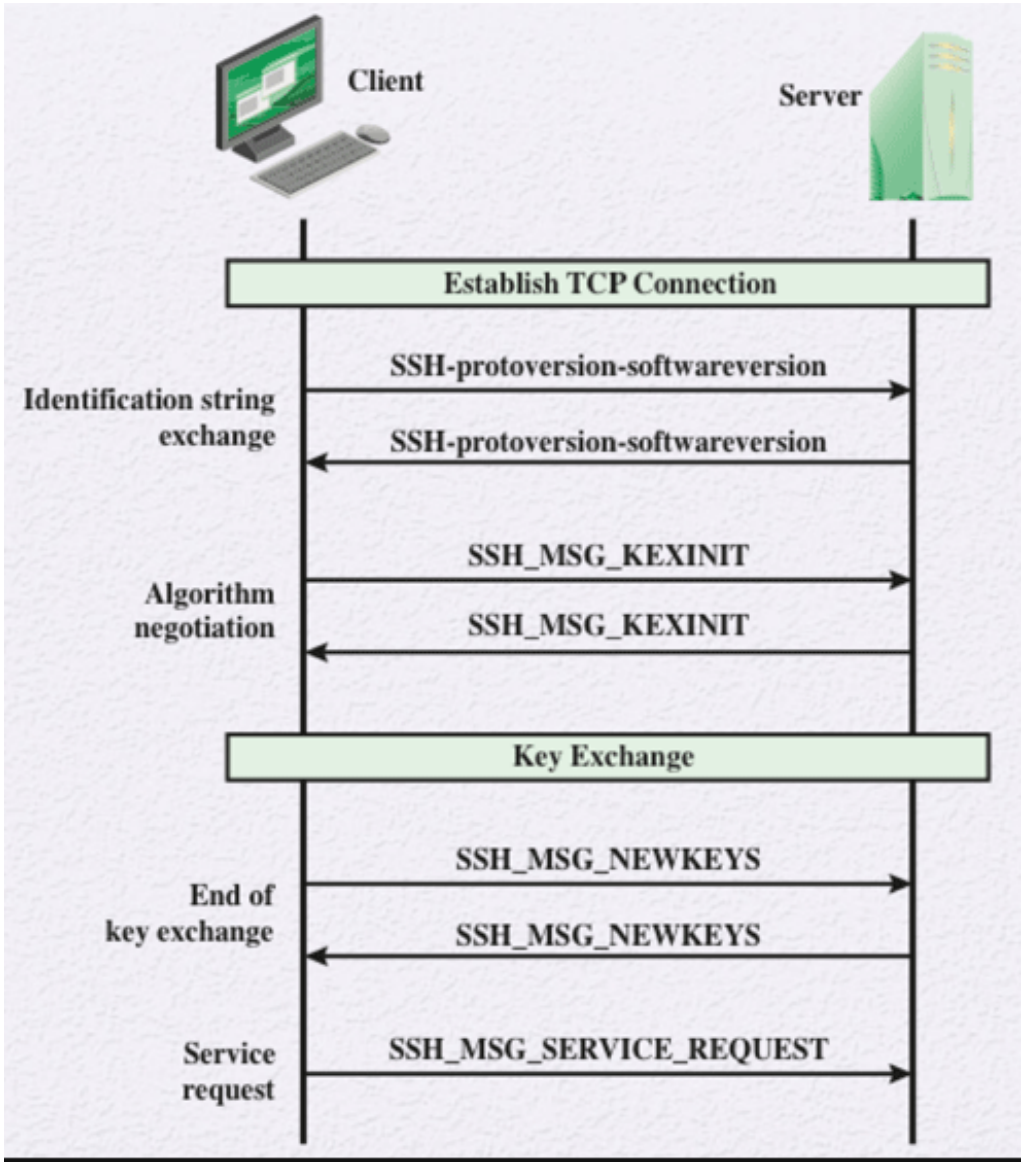
- **Transport Layer Protocol** provides server authentication, confidentiality, and integrity.
- **User Authentication Protocol** authenticates the client-side user to the server

SSH Protocol(s)



- **Transport Layer Protocol** 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

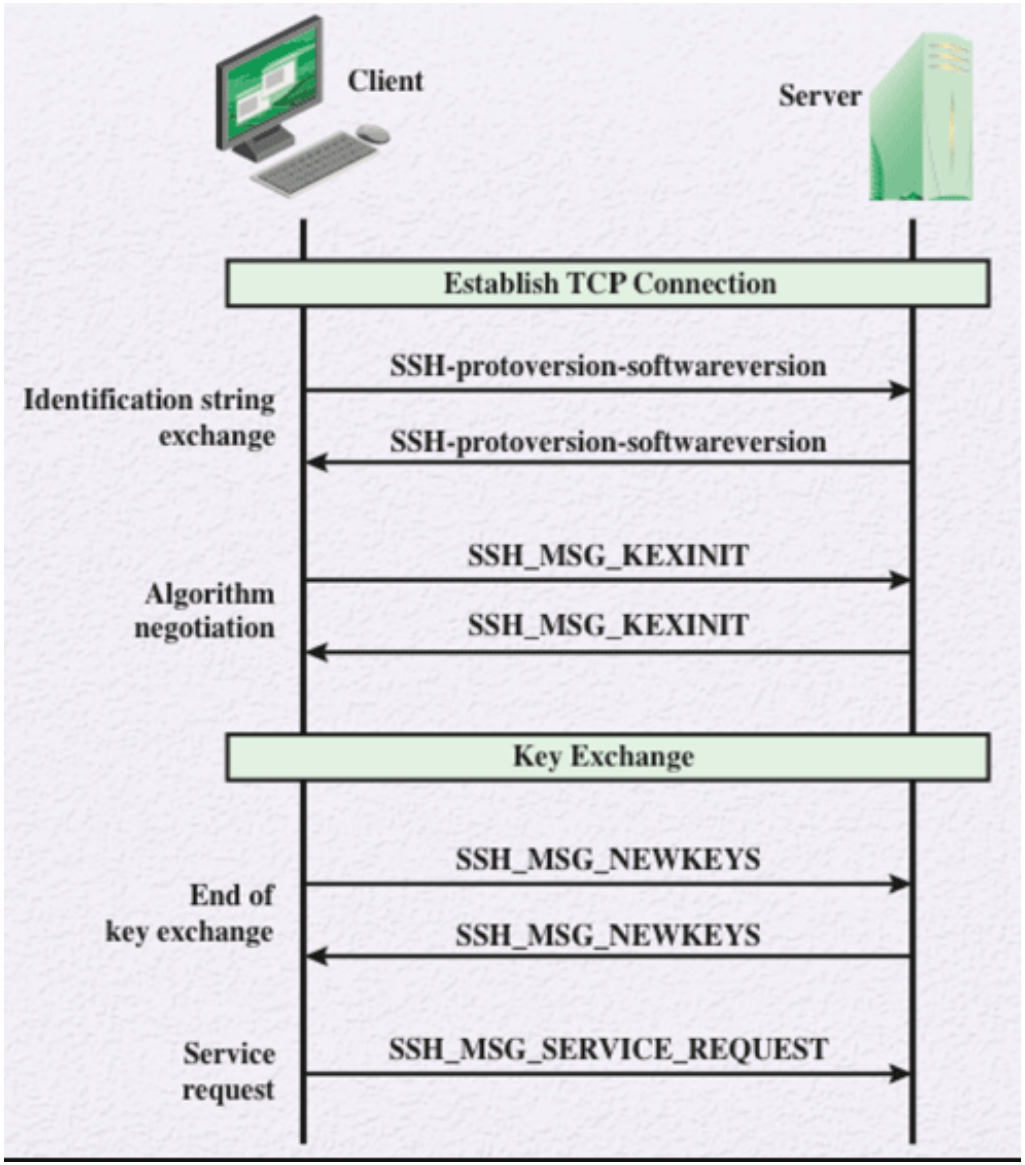
SSH Transport Layer Protocol



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 $\rightarrow$ SSH tunnel)

Port Forwarding

- Provides the ability to convert any insecure TCP connection into a secure SSH connection – a.k.a. SSH tunneling

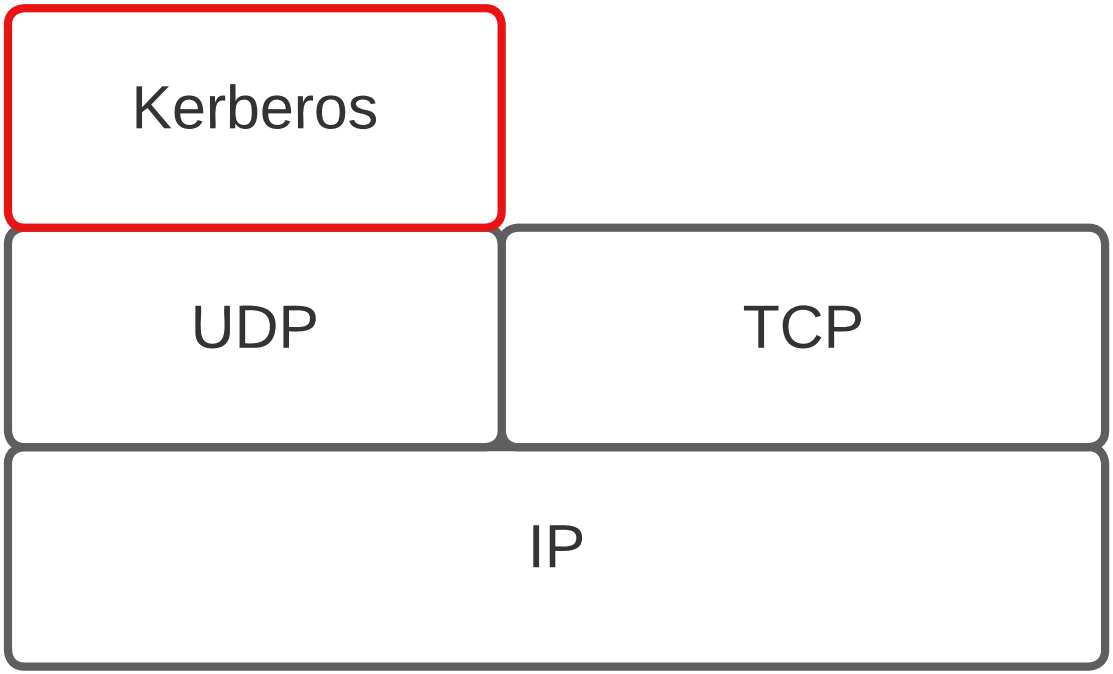
Port Forwarding

- Provides the ability to convert any insecure TCP connection into a secure SSH connection – a.k.a. SSH tunneling
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number (an identifier of a user in TCP)

Port Forwarding

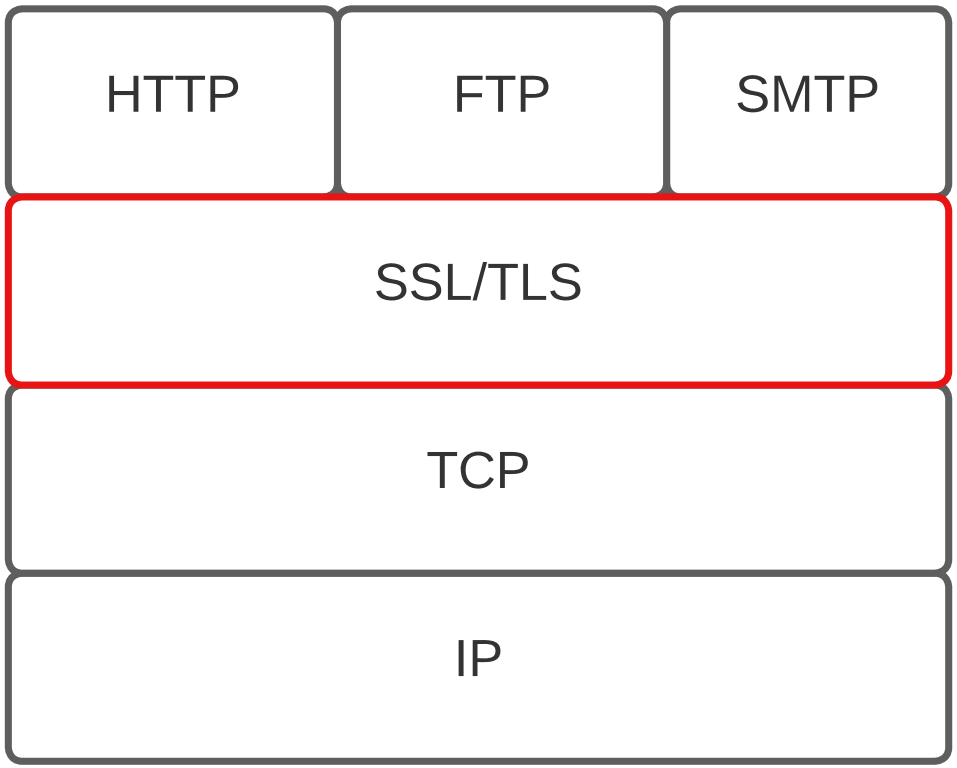
- Provides the ability to convert any insecure TCP connection into a secure SSH connection – a.k.a. SSH tunneling
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number (an identifier of a user in TCP)
- An application may employ multiple port numbers
  - HTTP servers usually listen on port 80 (443 for HTTPS)

Previously, on Network Security



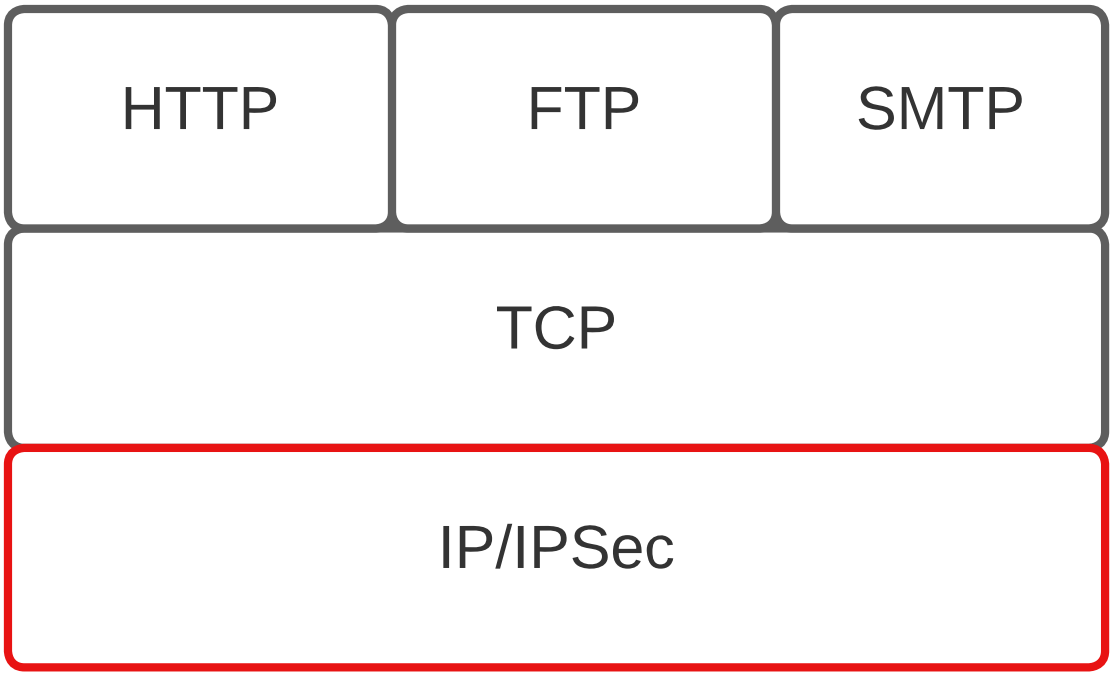
- Kerberos is at the application level - over UDP
- Security over insecure communication

Previously, on Network Security



- SSL/TLS is a middleware between application and TCP
- Security over reliable communication

Previously, on Network Security

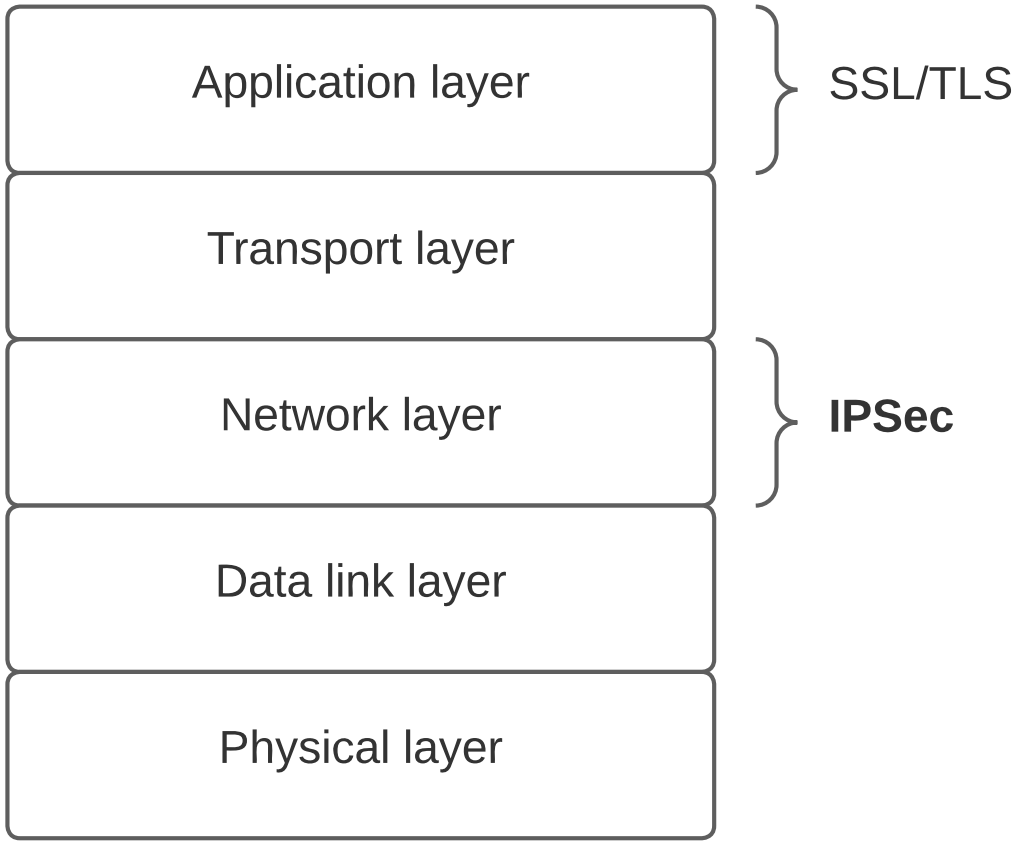


- IPSec refines the IP protocol
- Enhanced security **for all applications**

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 to be 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

SSL/TLS vs IPSec - P1

SSL/TLS

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

SSL/TLS vs IPSec - P1

SSL/TLS

- Lives at the socker 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
- SSL build into the Web early-on (Netscape)
- 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
- SSL build into the Web early-on (Netscape)
- 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!

Goals of IPSec

Authentication

Assures that a received packet was transmitted as the source in the *packet header*, and that the packet has *not been altered* in transit.

Goals of IPSec

Authentication

Assures that a received packet was transmitted as the source in the *packet header*, and that the packet has *not been altered* in transit.

Confidentiality

Enables communicating nodes to *encrypt* messages to prevent *eavesdropping* by third parties

Goals of IPSec

Authentication

Assures that a received packet was transmitted as the source in the *packet header*, and that the packet has *not been altered* in transit.

Confidentiality

Enables communicating nodes to *encrypt* messages to prevent *eavesdropping* by third parties

Key management

Ensures that communicating nodes can *securely exchange* cryptographic material (keys). Provided by the internet key exchange standard IKEv2.

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

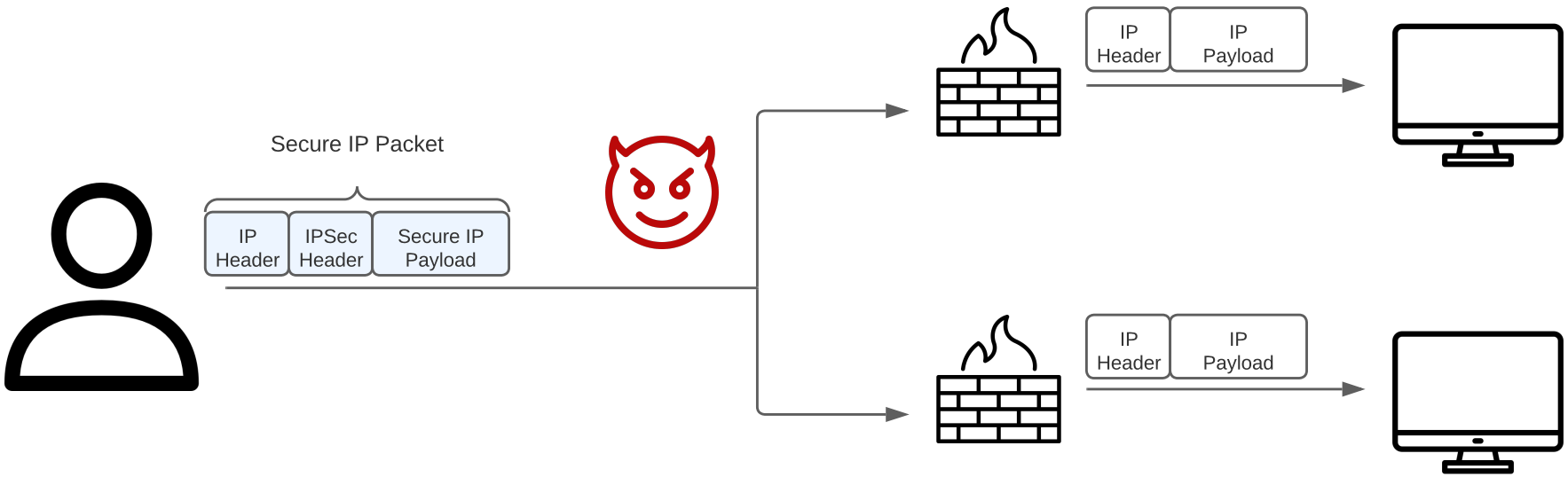
IPSec Applications

- 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

Request for Comments (RFCs)

Security Architecture for the Internet Protocol

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Table of Contents

The following SAD fields are used in doing IPsec processing:

- o Sequence Number Counter: a 32-bit value used to generate the Sequence Number field in AH or ESP headers. [REQUIRED for all implementations, but used only for outbound traffic.]
- o Sequence Counter Overflow: a flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent transmission of additional packets on the SA. [REQUIRED for all implementations, but used only for outbound traffic.]
- o Anti-Replay Window: a 32-bit counter and a bit-map (or equivalent) used to determine whether an inbound AH or ESP packet is a replay. [REQUIRED for all implementations but used only for inbound traffic. NOTE: If anti-replay has been disabled by the receiver, e.g., in the case of a manually keyed SA, then the Anti-Replay Window is not used.]
- o AH Authentication algorithm, keys, etc. [REQUIRED for AH implementations]
- o ESP Encryption algorithm, keys, IV mode, IV, etc. [REQUIRED for ESP implementations]
- o ESP authentication algorithm, keys, etc. If the authentication service is not selected, this field will be null. [REQUIRED for ESP implementations]
- o Lifetime of this Security Association: a time interval after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur. This may be expressed as a time or byte count, or a simultaneous use of both, the first lifetime to expire taking precedence. A compliant implementation MUST support both types of lifetimes, and must support a simultaneous use of both. If time is employed, and if IKE employs X.509 certificates for SA establishment, the SA lifetime must be constrained by the validity intervals of the certificates, and the NextIssueDate of the CRLs used in the IKE exchange

- Managed by IETF: Internet Engineering Task Force
- Open international community
  - Network designers; industry; academia

IPSec Documents

- Architecture
- Internet Key Exchange
- Encapsulating Security Payload
- Authentication Header
- Cryptographic algorithms
- Other

IPSec Documents

- Architecture
  - Covers the general concepts, security requirements, definitions and mechanisms defining IPSec technology
  - Current specification is RFC4301, *Security Architecture for the Internet Protocol*
- Internet Key Exchange
- Encapsulating Security Payload
- Authentication Header
- Cryptographic algorithms
- Other

IPSec Documents

- Architecture
- Internet Key Exchange
  - A collection of documents describing the key management schemes to use with IPSec
  - Main specification is RFC 729, *Internet Key Exchange (IKEv2) Protocol*, but there are many related RFCs
- Encapsulating Security Payload
- Authentication Header
- Cryptographic algorithms
- Other

IPSec Documents

- Architecture
- Internet Key Exchange
- Encapsulating Security Payload
  - An encapsulating header and trailer used to provide encryption or combined encryption/authentication
  - Current specification if RFC 4303, *IP Encapsulating Security Protocol (ESP)*
- Authentication Header
- Cryptographic algorithms
- Other

IPSec Documents

- Architecture
- Internet Key Exchange
- Encapsulating Security Payload
- Authentication Header
  - An extension header to provide message authentication
  - Current specification is RFC 4302, *IP Authentication Header*
  - Deprecated: guarantees already provided by ESP
- Cryptographic algorithms
- Other

IPSec Documents

- Architecture
- Internet Key Exchange
- Encapsulating Security Payload
- Authentication Header
- Cryptographic algorithms
  - A large set of documents defining and describing cryptographic algorithms for encryption, message authentication, psudorandom functions, and cryptographic key exchange.
- Other

IPSec Documents

- Architecture
- Internet Key Exchange
- Encapsulating Security Payload
- Authentication Header
- Cryptographic algorithms
- Other
  - There are also many other IPSec-related RFCs
  - Especially related with security policy management

IPSec Services

IPSec provides security services at the IP Layer. Enables a system to:

- Select the required security protocols
- Determine the algorithms for each service
- Input any cryptographic material required for said services

IPSec Services

IPSec provides security services at the IP Layer. Enables a system to:

- Select the required security protocols
- Determine the algorithms for each service
- Input any cryptographic material required for said services

According to RFC4301...

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
  - Partial sequence integrity
- Confidentiality

Architecture

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

Architecture

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

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

IPSec Key Management

- Handles key generation and distribution
- Often requires two pairs of keys
  - One for each direction
- Two types of key management

Manual

- A system administrator manually configures each system with its own keys as well as keys of other communicating systems
- Useful for small, static environments

Automated

- On-demand creation of keys
- Useful for large distributed systems, with evolving configuration

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

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!

IKE Phase 1

4 different “key options"

IKE Phase 1

4 different “key options"

- Public key encryption (original version)
- Public key encryption (improved version)

IKE Phase 1

4 different “key options"

- Public key encryption (original version)
- Public key encryption (improved version)
- Public key signature

IKE Phase 1

4 different “key options"

- Public key encryption (original version)
- Public key encryption (improved version)
- Public key signature
- Symmetric key

IKE Phase 1

4 different “key options"

- Public key encryption (original version)
- Public key encryption (improved version)
- Public key signature
- Symmetric key

For each of these, we have 2 different “modes":

- Main mode - thorough handshake with encrypted authentication
- Aggressive mode - faster method, e.g. both peers have external dynamic IP addresses

IKE Phase 1

4 different “key options"

- Public key encryption (original version)
- Public key encryption (improved version)
- Public key signature
- Symmetric key

For each of these, we have 2 different “modes":

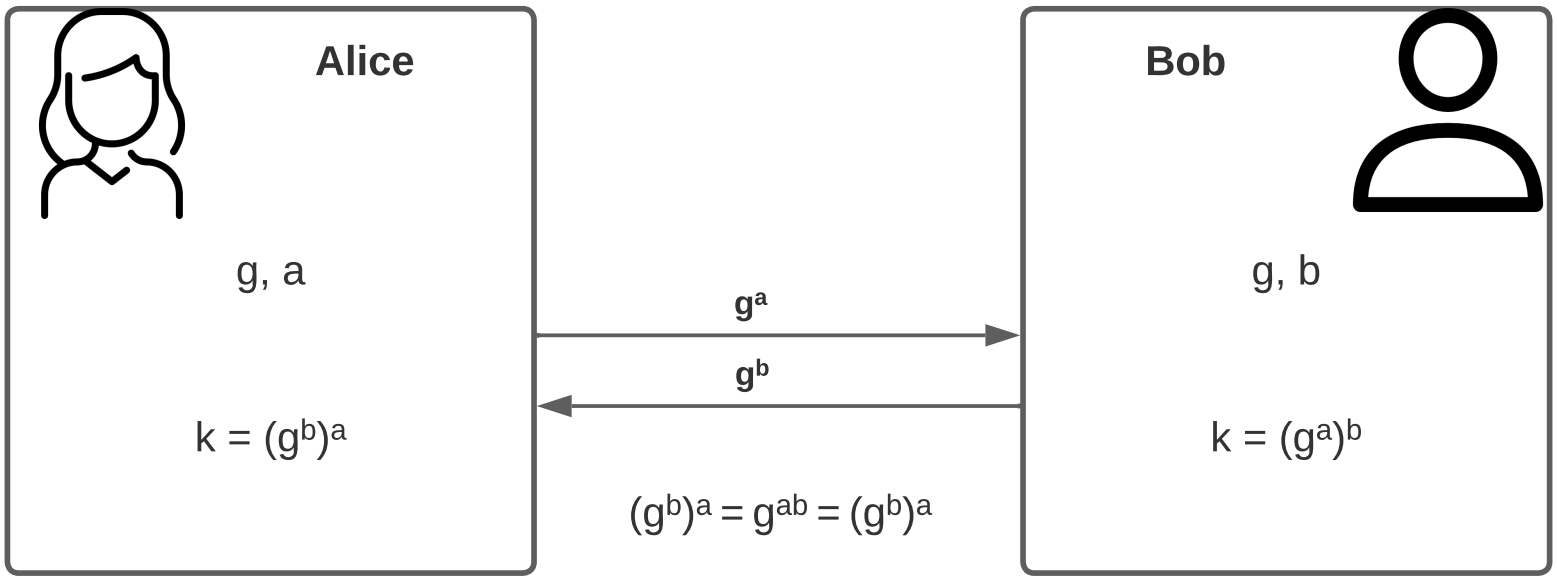
- Main mode - thorough handshake with encrypted authentication
- Aggressive mode - faster method, e.g. both peers have external dynamic IP addresses

8 versions of IKE Phase 1!

- Common downside: over-engineered

IKE Phase 1

Diffie-Hellman



- Many commercial products employ this key exchange techniques
- The algorithm itself is limited to the exchange of secret values
- Security relies on the difficulty of computing discrete logarithms

Features of IKE Key Agreement

Algorithm used is (quite) a bit more complex than what was 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

1 - Thwart clogging attacks

Clogging attack:

- An adversary forges the source address of a legitimate user and sends a public DH key to the victim
- The victim performs a modular exponentiation and computes the key
- Repeated message clog the system with useless work
  - Modular exponentiations aren't as free as you might think!

Cookie exchange:

1 - Thwart clogging attacks

Clogging attack:

Cookie exchange:

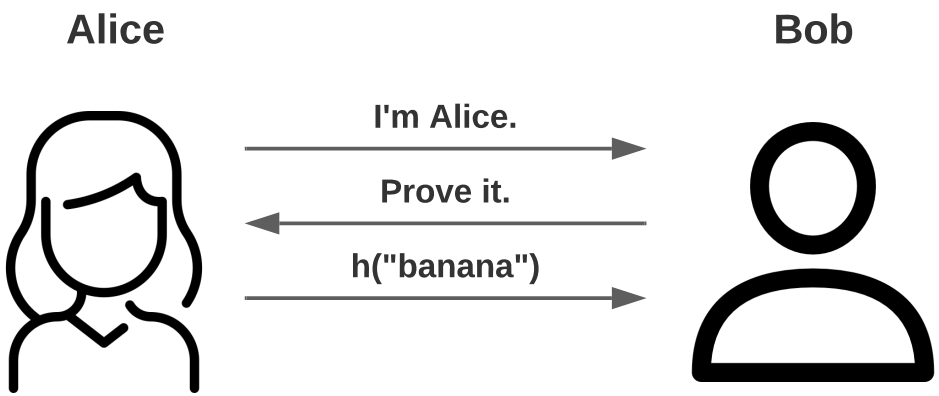
- Each side sends a pseudorandom number (cookie) in the initial message. The other side acknowledges (ack)
- Ack must be repeated in the first message of the DH key exchange
- If the source address was forged, the adversary gets no answer!
  - Adversary can still force users to generate acks
  - But users never have to generate useless exponentiations!

3 - Nonces to prevent replay

Replay attack:

- Authentication protocol
- Valid data transmission from legitimate user to receiver
- Captured by adversary and resent to authenticate him!

Nonces:

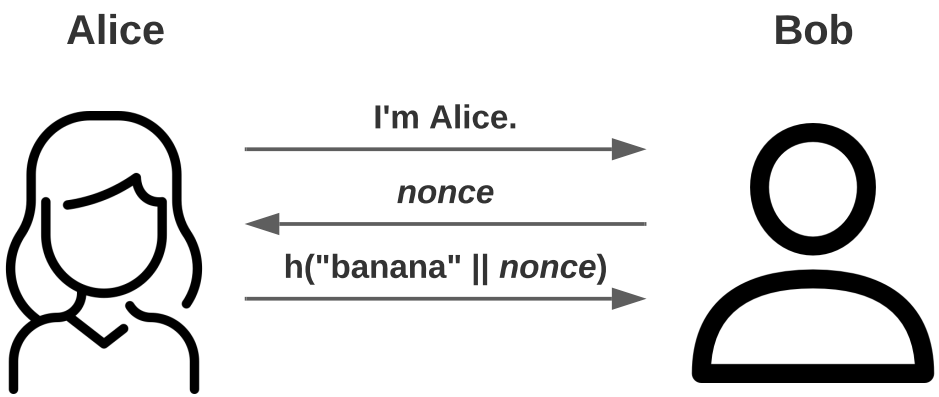


3 - Nonces to prevent replay

Replay attack:

Nonces:

- Locally generated pseudorandom number
- Challenge-response mechanism
  - Sent as challenge
  - Response must depend on it in some way
- Encrypted during certain portions of the exchange



Summary of IKE Phase 1

Output of Phase 1

- Mutual authentication
- Shared symmetric key
- IKE Security Association (SA)

Summary of IKE Phase 1

Output of Phase 1

- Mutual authentication
- Shared symmetric key
- IKE Security Association (SA)

Similar to TLS/SSL, Phase 1 is **expensive**

IKE was developed to use in many other applications other than IPSec

- Hence the adaptability/complexity

IKE Phase 2

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

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 Association Database (SAD)

A Security Association Database is used to store long-term parameters associated with each SA.

Security Association Database (SAD)

A Security Association Database is used to store long-term parameters associated with each SA.

Normally defined by:

- *Security parameter index*
- Sequence number counter
- Sequence number overflow
- Anti-replay window
- *AH information*
- *ESP information*
- Lifetime of Security Association
- *IPSec protocol mode*
- ...

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

Summary of IKE Phase 2

Outputs

- Phase 1 gives us an IKE SA
- Phase 2 gives us an IPSec SA
- We now have a symmetric session key

Summary of IKE Phase 2

Outputs

- Phase 1 gives us an IKE SA
- Phase 2 gives us an IPSec SA
- We now have a symmetric session key

Now what?

- We want to protect IP datagrams
- What is an IP datagram?
- ... and how can IPSec help?

Review - Internet Protocol

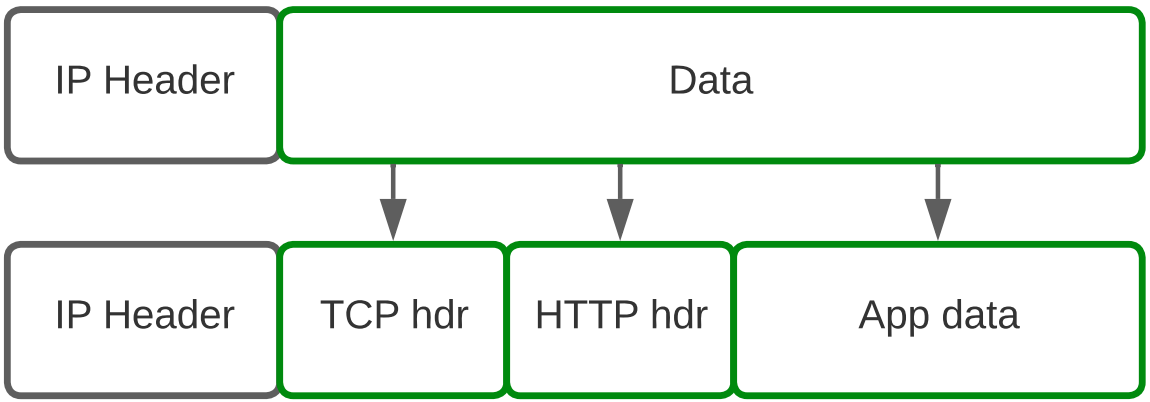
An IP datagram is something of the form

- 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

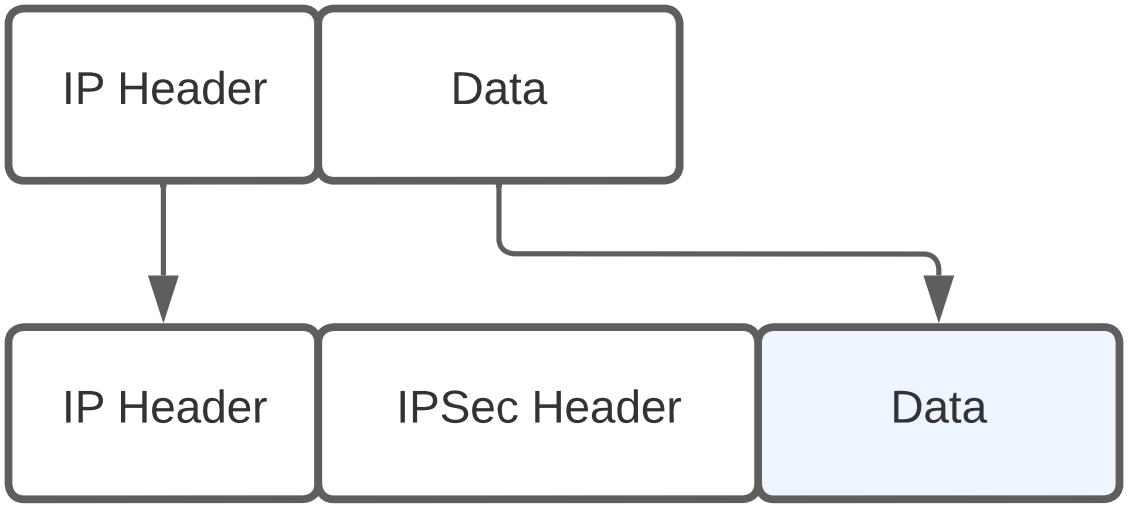
- IP encapsulates TCP
- TCP encapsulates HTTP



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

Two Execution Modes

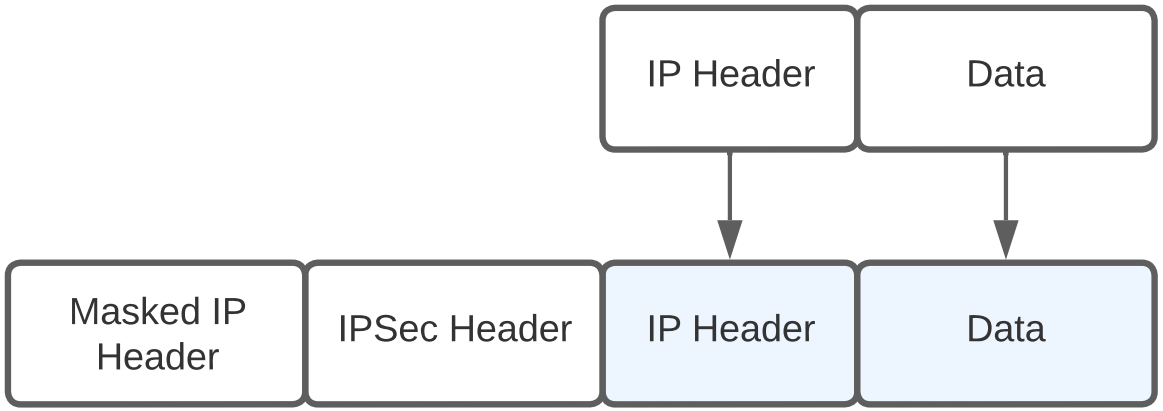
Transport Mode



- 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

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

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 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"*[a]

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

IPSec Modes

Summary

	Transport Mode SA	**Tunnel Mode SA**
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header + payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts the entire inner IP payload.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload.	Encrypts the entire inner IP packet. Authenticates the inner IP packet.

Wrap up

SSH

- Relies on an handshake similar to TLS...

Wrap up

SSH

- Relies on an handshake similar to TLS...
- ... but authentication is not certificate-based

Wrap up

SSH

- Relies on an handshake similar to TLS...
- ... but authentication is not certificate-based
- Allows for different channels with different purposes

IPSEC

- Key establishment protocol (IKE) is done over UDP. This makes it unreliable, and thus is blocked by some firewalls.

Wrap up

SSH

- Relies on an handshake similar to TLS...
- ... but authentication is not certificate-based
- Allows for different channels with different purposes

IPSEC

- Key establishment protocol (IKE) is done over UDP. This makes it unreliable, and thus is blocked by some firewalls.
- IPSec and firewalls don't always mix well.
  - Some firewalls change authenticated addresses, subverting the datagram structure

Wrap up

SSH

- Relies on an handshake similar to TLS...
- ... but authentication is not certificate-based
- Allows for different channels with different purposes

IPSEC

- Key establishment protocol (IKE) is done over UDP. This makes it unreliable, and thus is blocked by some firewalls.
- IPSec and firewalls don't always mix well.
  - Some firewalls change authenticated addresses, subverting the datagram structure
- Managing IPSec policy is quite complex
  - Mistakes lead to loss of connectivity
  - Mistakes lead to loss of security
  - Many options to keep track of.

Wrap up

SSH

- Relies on an handshake similar to TLS...
- ... but authentication is not certificate-based
- Allows for different channels with different purposes

IPSEC

- Key establishment protocol (IKE) is done over UDP. This makes it unreliable, and thus is blocked by some firewalls.
- IPSec and firewalls don't always mix well.
  - Some firewalls change authenticated addresses, subverting the datagram structure
- Managing IPSec policy is quite complex
  - Mistakes lead to loss of connectivity
  - Mistakes lead to loss of security
  - Many options to keep track of.
- Still, many applications benefit from IPSec!

Wrap up

IPSEC

IPSec assures that:

- A router advertisement comes from an authorized router
- A router seeking to establish/maintain neighbour relationship with a router in another domain is authorized
- A redirect message comes back to its authentic original source
- A routing update is not forged

Network Security - Week 6

Manuel E. Correia

DCC/FCUP

2025