Network Security

Objectives

Brief survey of network security challenges.

Show how network security contributes to and depends on computer security.

Introduction to the design of network security protocols, based on the Internet security protocols IPsec and SSL/TLS.

Network boundaries as security perimeters.

Principles and limitations of firewalls and Intrusion Detection Systems.

Network Attacks

Passive attacker: listens to traffic (eavesdropping, wiretapping, sniffing).

Active attacker: modifies messages, inserts new messages, corrupts network management information; active attacks are not necessarily more difficult to mount than passive attacks.

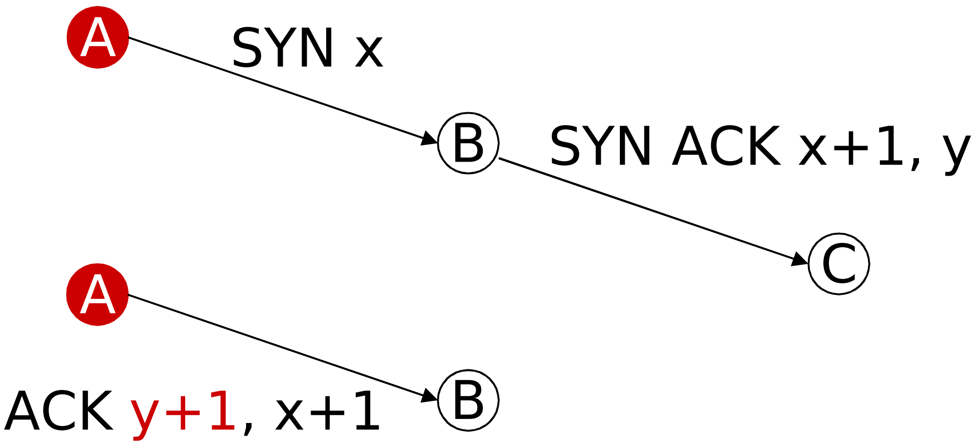
Spoofing attack: send messages with forged sender addresses.

Flooding (bombing) attack: large number of messages sent to victim.

Traffic analysis: identify communications patterns; may be possible even when the attacker cannot read individual messages.

TCP Session Hijacking

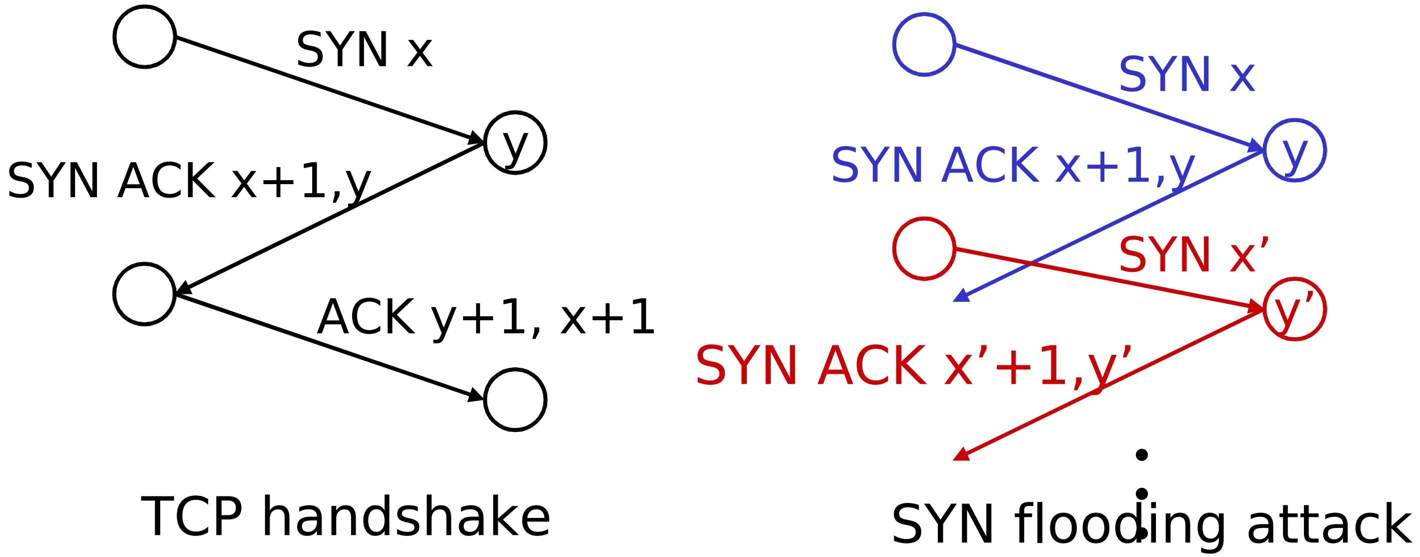
Predict challenge to send messages that appear to come from a trusted host.



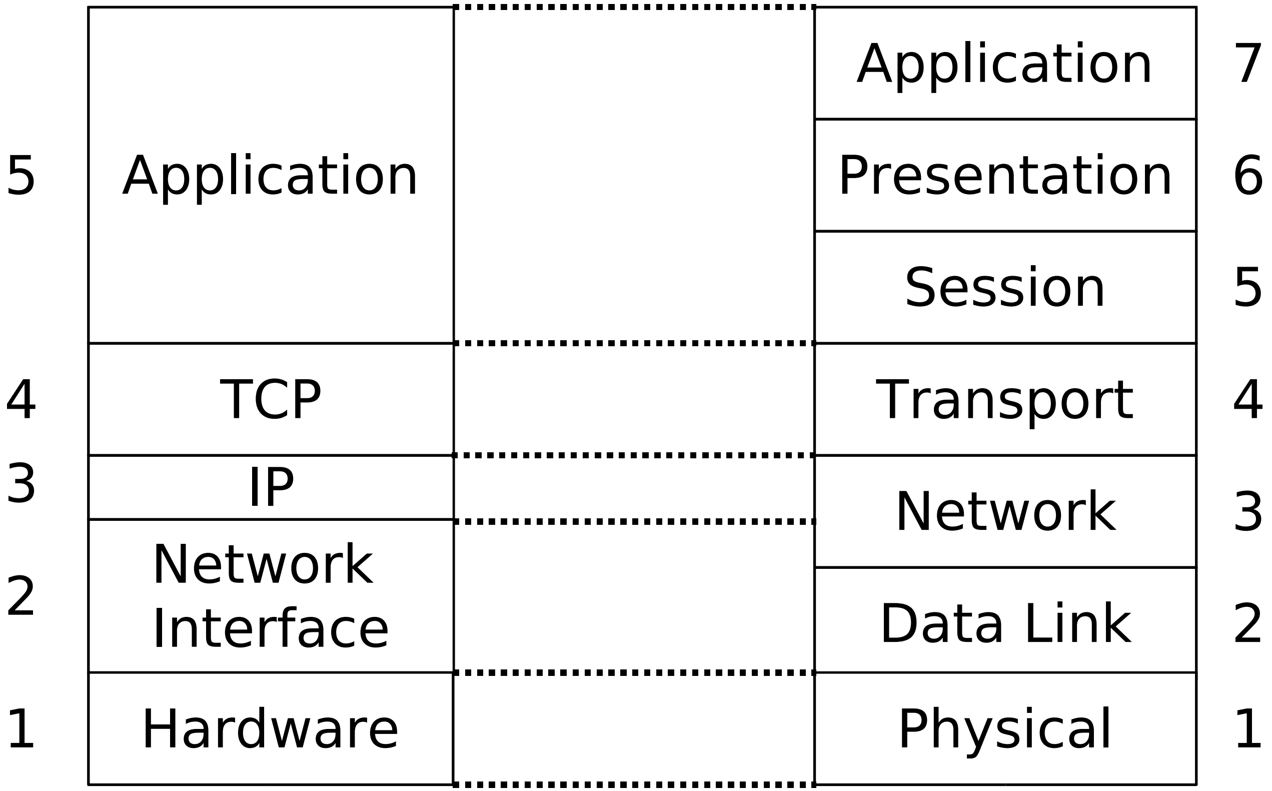
1. A spoofs SYN packet from C.
2. B sends SYN ACK to C.
3. A guesses the value y+1 to conclude the handshake.

TCP SYN Flooding Attacks

Exhaust responder’s resources by creating half-open TCP connection requests.



Protocol Layering



5 Application

4 TCP

3 IP

2 Network Interface

1 Hardware

Application 7

Presentation 6

Session 5

Transport 4

Network 3

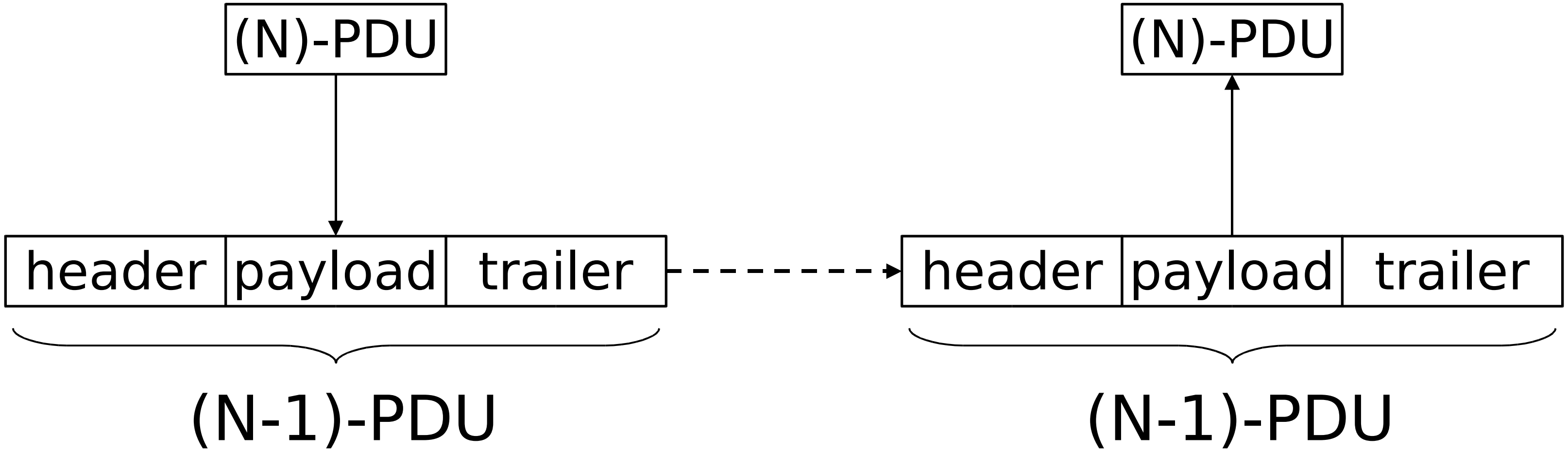
Data Link 2

Physical 1

Internet

ISO/OSI 7 layer model

Protocol Layering



PDU … Protocol Data Unit

Implementing Security Services

Header in (*N*-1)-PDU is convenient location for storing security relevant data.

Upper layer protocol can be aware of lower layer security services:

- Upper layer protocol has to change its calls so that they refer to the security facilities provided.

Lower layer security services can be transparent to upper layer protocol:

- Upper layer protocol need not be changed at all.

IPsec

Defined in IETF RFCs 2401–2412.

Provides security at network (Internet) layer.

- All IP datagrams covered.
- No re-engineering of applications.
- Transparent to upper layer.

Mandatory for next generation IPv6, optional for current generation (IPv4).

Two basic modes of use:

- Transport mode: IPsec-aware hosts as endpoints.
- Tunnel mode: for IPsec-unaware hosts, tunnel established by intermediate gateways or host OS.

IPsec

Authentication and/or confidentiality services for data:

- AH protocol [RFC 2402]
- ESP protocol [RFC 2406]

Use of AH is being deprecated in favour of ESP.

- Political reasons for introducing an authentication-only protocol in the 1990s have faded.

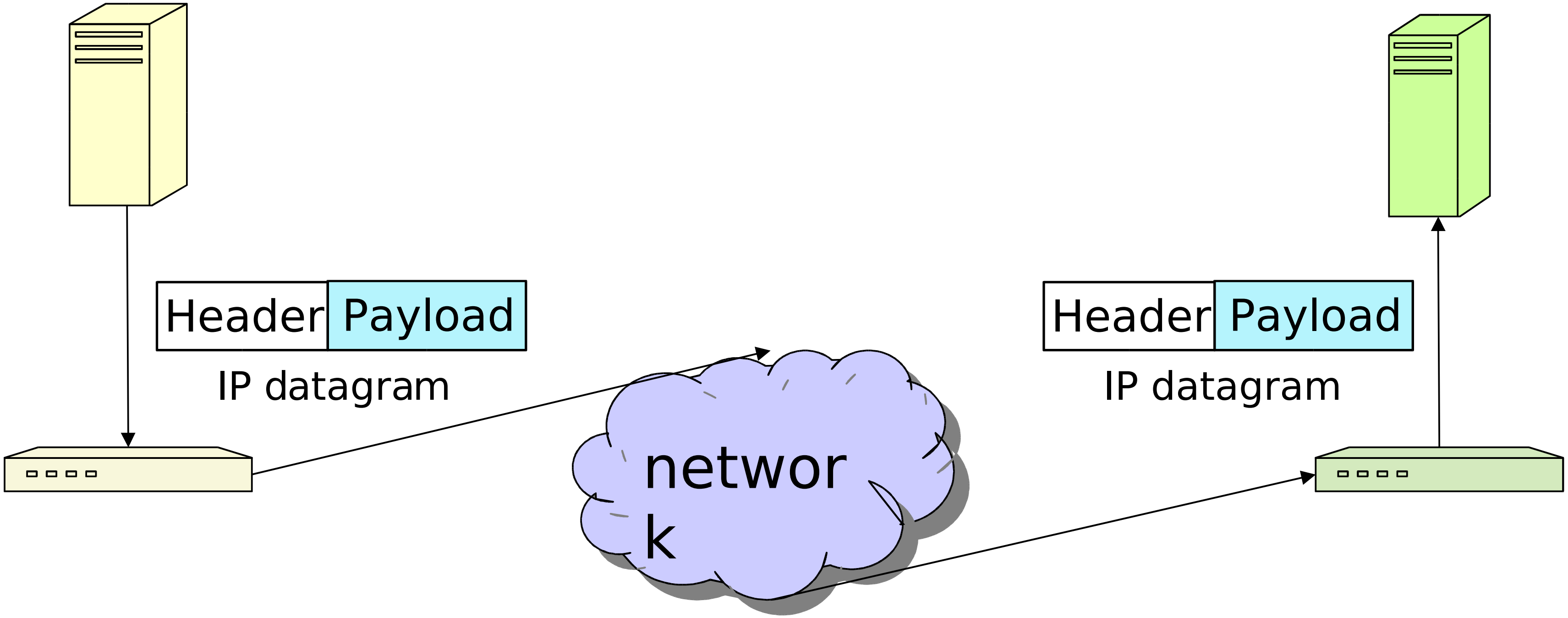
(Too?) flexible set of key establishment methods:

- IKE; IKEv2 under development.

IPsec Transport Mode

Host-to-host (end-to-end) security:

- IPsec processing performed at endpoints of secure channel.
- Endpoint hosts must be IPsec-aware.



IPsec Tunnel Mode

Entire IP datagram plus security fields treated as new payload of ‘outer’ IP datagram.

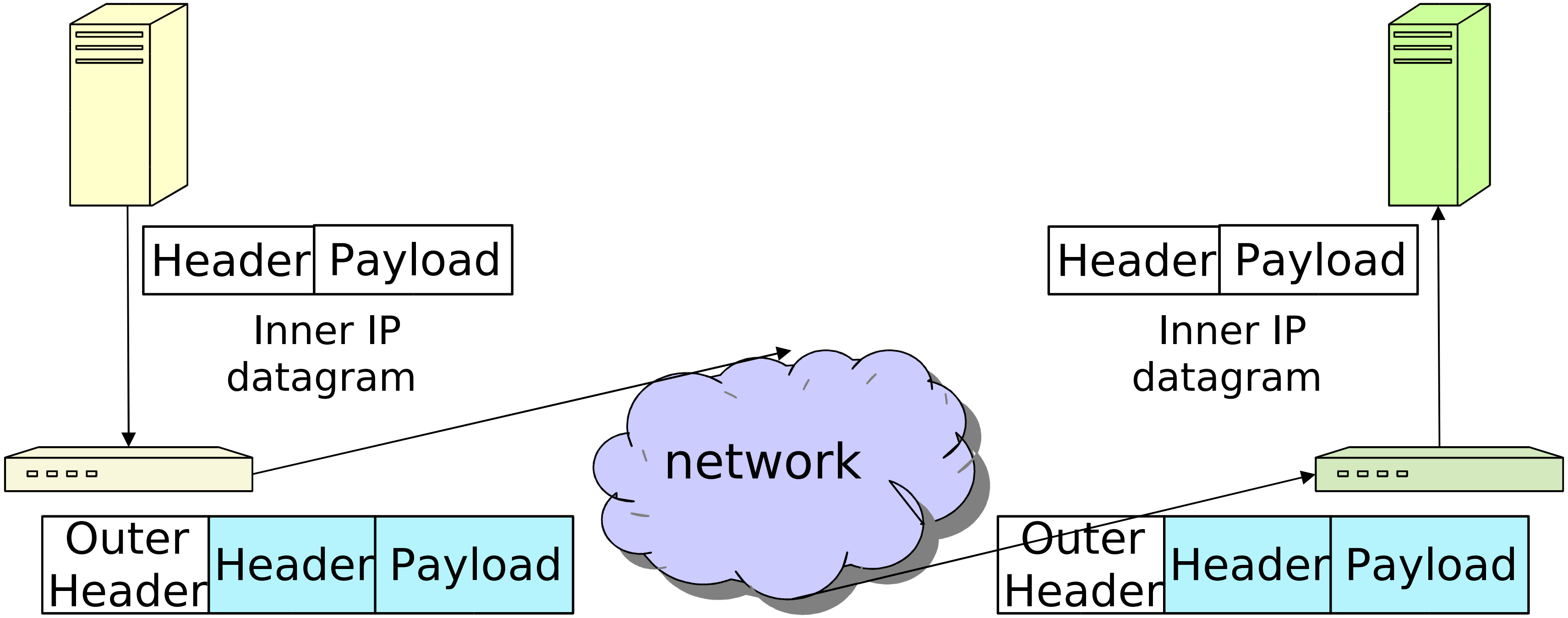
- Original ‘inner’ IP datagram encapsulated within ‘outer’ IP datagram.

IPsec processing performed at security gateways on behalf of endpoint hosts.

- Gateway could be perimeter firewall or router.
- Gateway-to-gateway but not end-to-end security.
- Hosts need not be IPsec-aware.

Encrypted inner IP datagram, including original source and destination addresses, not visible to intermediate routers.

IPSec Tunnel Mode



ESP Protocol

Encapsulating Security Payload [RFC 2406].

Provides one or both of:

- confidentiality for payload/inner datagram; sequence number not protected by encryption.
- Authentication of payload/inner datagram, but not of outer IP header.

Traffic-flow confidentiality in tunnel mode.

Symmetric encryption and MACs based on secret keys shared between endpoints.

ESP Headers

ESP specifies header and trailer to be added to IP datagrams.

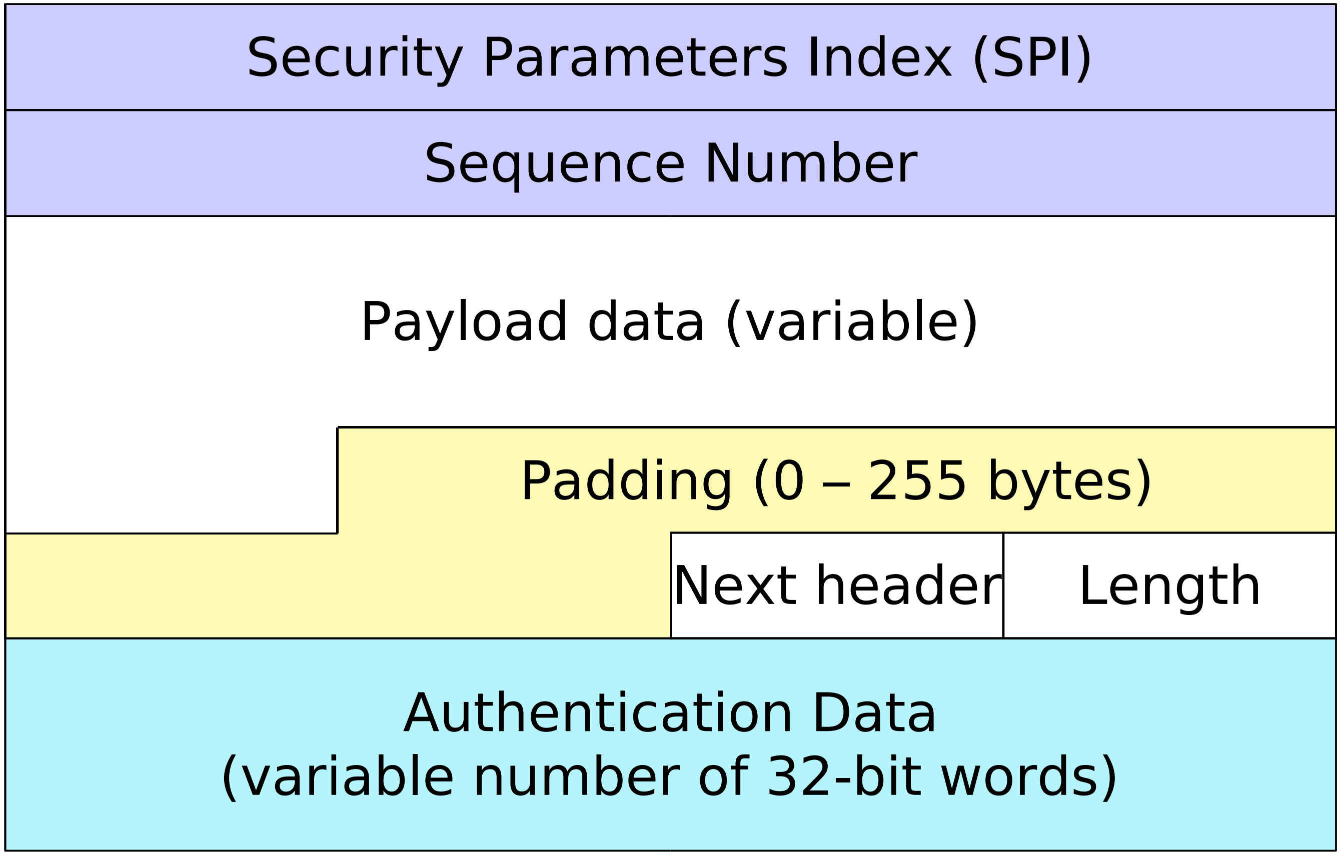
Header fields include:

- SPI (Security Parameters Index): identifies which algorithms and keys are to be used for IPsec processing (more later).
- Sequence number.

Trailer fields include:

- Any padding needed for encryption algorithm (may also help disguise payload length).
- Padding length.
- Authentication data (if any), i.e. the MAC value.

ESP Header (RFC 2406)



Security Parameters Index (SPI)

Sequence Number

Payload data (variable)

Padding (0 – 255 bytes)

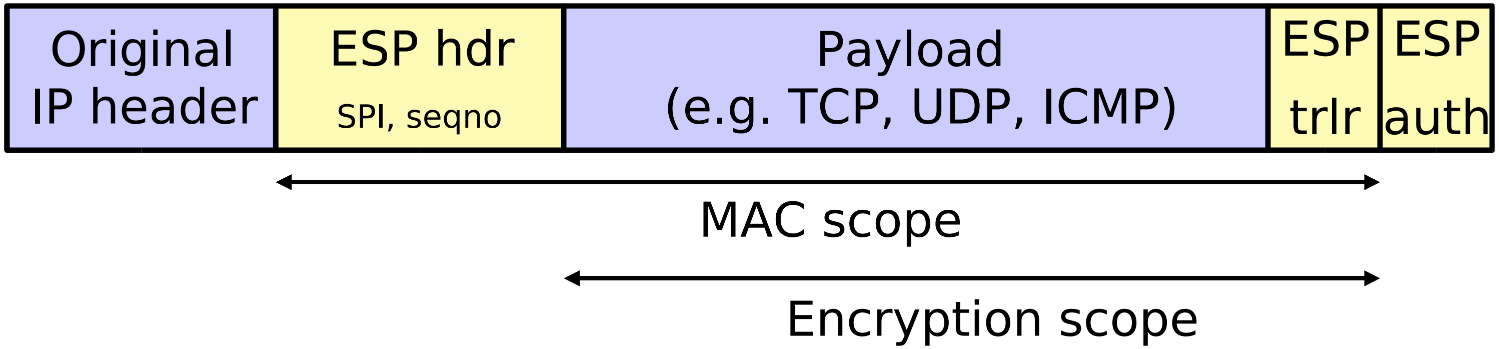
Next header

Length

Authentication Data
(variable number of 32-bit words)

ESP Protocol – Transport & Tunnel

ESP in transport mode:



Original IP header

ESP hdr SPI, seqno

Payload (e.g. TCP, UDP, ICMP)

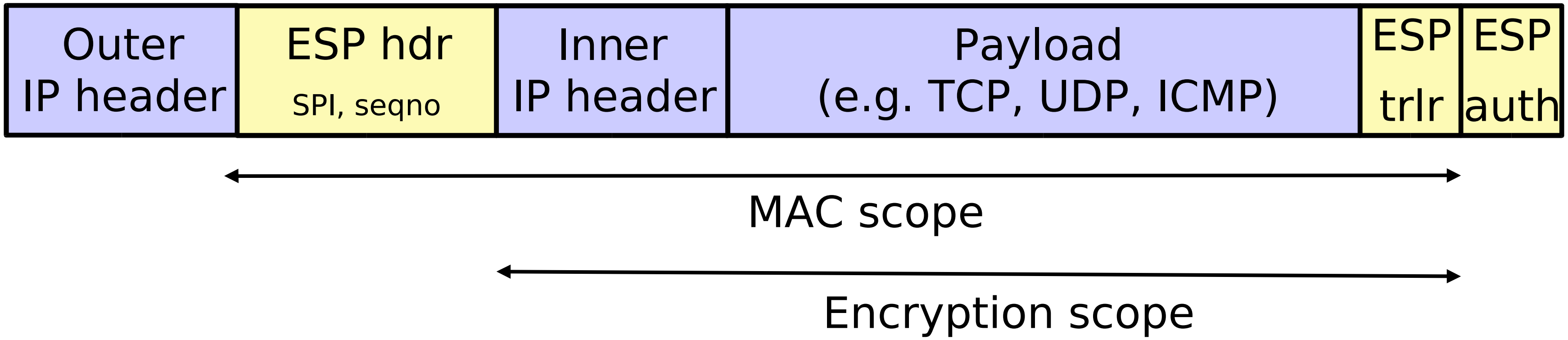
ESP trlr

ESP auth

MAC scope

Encryption scope

ESP in tunnel mode:



Outer IP header

ESP hdr SPI, seqno

Inner IP header

Payload (e.g. TCP, UDP, ICMP)

ESP trlr

ESP auth

MAC scope

Encryption scope

IPsec Security Policy

IPsec aware hosts need rules for processing packets:

- Drop, pass through, encrypt, MAC?
- Which key and algorithm to apply?

Rules stored in a Security Policy Database (SPD).

SPD consulted for each outbound and inbound packet.

Fields in packet matched against fields in SPD entries:

- Based on source and destination addresses (address ranges), transport layer protocol, transport layer port numbers, …
- Match identifies a Security Association (SA), or a group of SAs, or the need for a new SA.

IPsec Security Association (SA)

A SA is a one-way (simplex) relationship between sender and receiver.

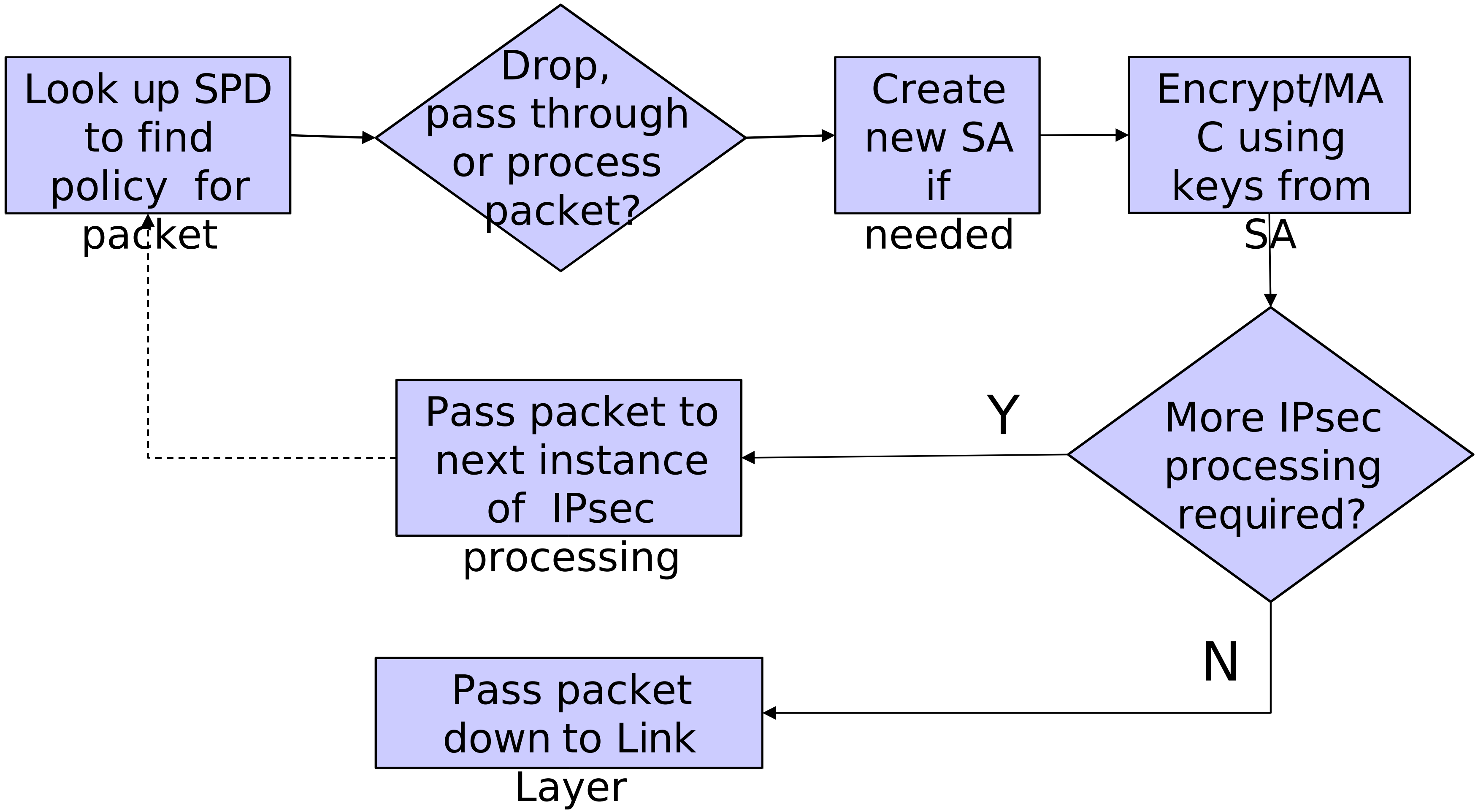
- Specifies processing to be applied to *this* datagram from *this* sender to *this* receiver.

List of active SAs held in SA database (SADB).

SA identified by SPI, source address, destination address; contains:

- Sequence number counter and anti-replay window,
- AH/ESP info: algorithms, IVs, keys, key lifetimes,
- SA lifetime,
- Protocol mode: tunnel or transport,
- …

IPsec Outbound Processing



Look up SPD to find policy for packet

Drop, pass through or process packet?

Create new SA if needed

Encrypt/MAC using keys from SA

More IPsec processing required?

Y

Pass packet to next instance of IPsec processing

N

Pass packet down to Link Layer

Combining SAs

IPsec security services may be provided at different points in network.

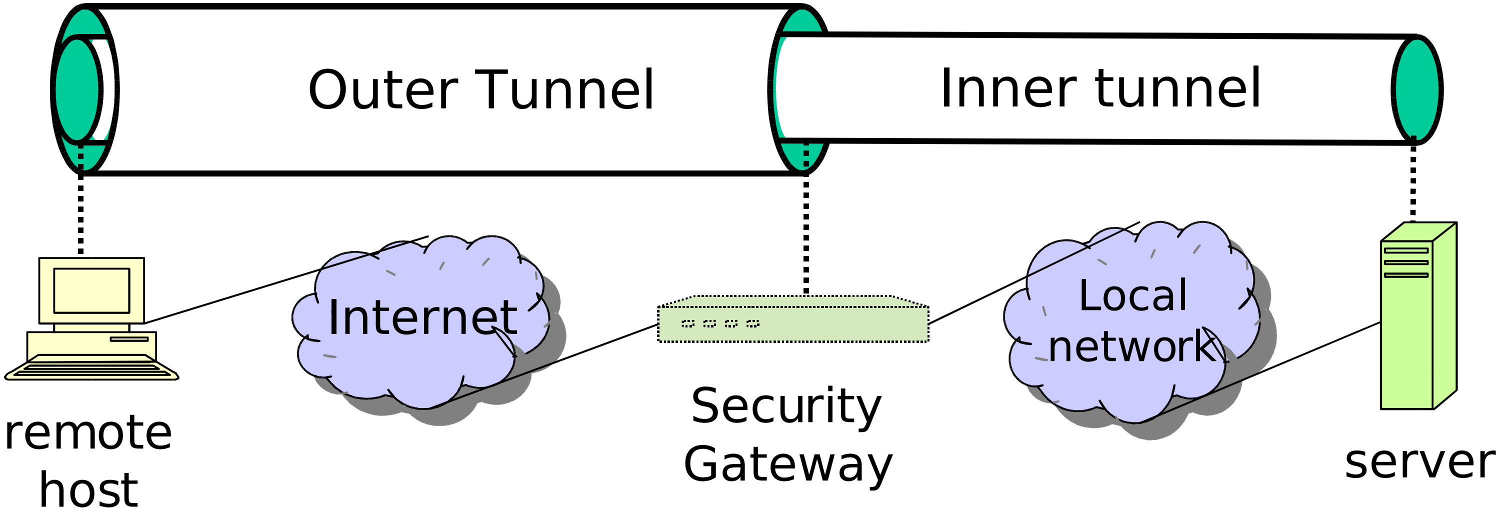
- Host-to-host.
- Gateway-to-gateway for Virtual Private Network (VPN).

SAs can be combined using:

- Transport adjacency: more than one SA applied to same IP datagram without tunnelling.
- Iterated tunnelling: multiple levels of nesting of IPsec tunnels; each level has its own SA; each tunnel can begin/end at different IPsec site along route.

Example

- Remote host has Internet access to gateway, then gains access to server behind gateway.
- Traffic to server protected in inner tunnel.
- Outer tunnel protects inner traffic over Internet.



IPsec Key Management

IPsec needs a lot of symmetric keys:

- One key for each SA.
- Different SA for each combination of {ESP,AH} × {tunnel,transport} × {sender, receiver}.

Two sources for SAs and keys:

- Manual keying: works for small number of nodes but hopeless for reasonably sized networks of IPsec-aware hosts; requires manual re-keying.
- IKE: Internet Key Exchange [RFC 2409]; many options and parameters.

IKE Security Goals

Entity authentication of participating parties.

Establish a fresh shared secret, used to derive further keys:

- for protecting IKE management channel,
- for SAs for general use.

Secure negotiation of all algorithms.

- Authentication method, key exchange method, encryption and MAC algorithms, hash algorithms.

Resistance to Denial-of-Service attacks: cookie mechanism.

Options for perfect forward secrecy, deniable authentication and identity protection.

IKE Phases

IKE operates in two phases.

Phase 1: Sets up an SA as a secure channel to carry further SA negotiation, plus error and management traffic.

- Bi-directional.
- ‘Expensive’ entity authentication and key exchange.
- Establishes a secure channel for use in Phase 2.

Phase 2: Negotiates SAs for general use.

- Fast negotiation over Phase 1 secure channel.
- Many Phase 2 runs allowed for each run of Phase 1.
- Multiple SAs can be negotiated per run.

IKE Phase 1 Main Mode: Example

Phase 1 main mode using ‘authentication with signatures’ (simplified!)

(I=Initiator, R=Responder, […]=optional)

1. I R: HDRi, SA_i
2. R I: HDRr, SA_r
3. I R: HDRi, KE_i, N_i [,Cert_Req]
4. R I: HDRr, KE_r, N_r [,Cert_Req]
5. I R: HDRi*{IDii, [Cert_i,] Sig_i}
6. R I: HDRr*{IDir, [Cert_r,] Sig_r}

Explanation

Messages 1 and 2:

- I and R exchange cookies CKY-I, CKY-R (in HDR fields) and ordered lists of preferred/accepted algorithms (in SA_i, SA_r).
- Cookies provide limited anti-DoS measure.

Messages 3 and 4:

- Exchange of Diffie-Hellman values (KE_I = g^x, KE_r = g^y), nonces (N_i, N_r), and request certificates.

Messages 5 and 6:

- Exchange of identities, certificates, and signatures on hash of (DH values, nonces, SAs,…).
- Everything inside *{…} is encrypted using key SKEYID_e derived from DH values and nonces.

SSL

SSL/TLS Overview

SSL = Secure Sockets Layer.

- unreleased v1, flawed but useful v2, good v3.

TLS = Transport Layer Security [RFC 2246]

- TLS1.0 = SSL3.0 with minor tweaks (see later)

SSL/TLS provides security ‘at TCP layer’.

- Uses TCP to provide reliable end-to-end transport.
- Usually a thin layer between TCP and HTTP.
- Applications need to be aware of SSL/TLS..

Widely used in Web browsers and servers to support ‘secure e-commerce’ over HTTP.

SSL/TLS Basic Features

SSL Record Protocol: Provides secure, reliable channel to second layer.

Second layer carries SSL Handshake Protocol, Change Cipher Spec. Protocol, Alert Protocol, HTTP, and other application protocols.

SSL Handshake Protocol establishes keys for MAC and encryption at Record Layer.

Different keys in each direction.

SSL Handshake Protocol – Goals

Entity authentication of participants.

- Participants are ‘client’ and ‘server’.
- Server nearly always authenticated, client more rarely.
- Appropriate for most e-commerce applications.

Establish a fresh, shared secret.

- Shared secret used to derive further keys.
- For confidentiality and authentication in SSL Record Protocol.

Secure ciphersuite negotiation.

- Encryption and hash algorithms
- Authentication and key establishment methods.

Sessions & Connections

Session:

- Created by handshake protocol.
- Defines set of cryptographic parameters (encryption and hash algorithm, master secret, certificates).
- Carries multiple connections to avoid repeated use of expensive handshake protocol.

Connection:

- State defined by nonces, secret keys for MAC and encryption, IVs, sequence numbers.
- Keys for many connections derived from single master secret created during handshake protocol.

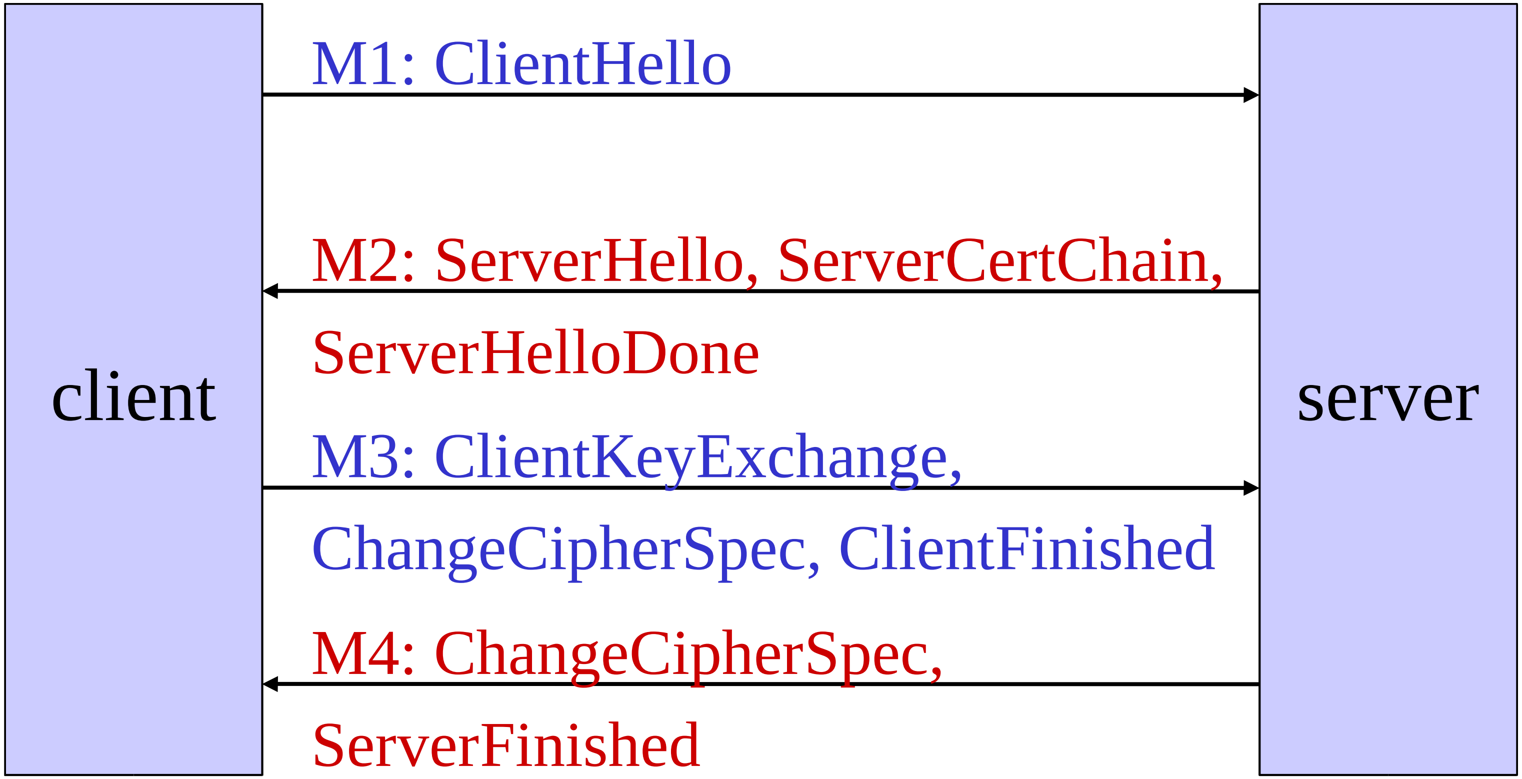
SSL Handshake Protocol: Run

We sketch the most common use of SSL:

- No client authentication.
- Client sends `pre_master_secret` using Server's public encryption key from Server certificate.
- Server authenticated by ability to decrypt to obtain `pre_master_secret`, and construct correct `finished` message.

Other protocol runs are similar.

SSL Handshake Protocol Run



M1: `ClientHello`

Client initiates connection.

Sends client version number.

- 3.1 for TLS.

Sends **`ClientNonce`**.

- 28 random bytes plus 4 bytes of time.

Offers list of ciphersuites:

- Key exchange and authentication options, encryption algorithms, hash functions.
- E.g. **`TLS_RSA_WITH_3DES_EDE_CBC_SHA`**.

M2: `ServerHello, …`

Sends server version number.

Sends **`ServerNonce`** and **`SessionID`**.

Selects single ciphersuite from list offered by client.

Sends **`ServerCertChain`** message.

- Allows client to validate server's public key back to acceptable root of trust.

(optional) **`CertRequest`** message.

- Omitted in this protocol run – no client authentication.

Finally, **`ServerHelloDone`**.

M3: `ClientKeyExchange`, …

`ClientKeyExchange` contains encryption of **`pre_master_secret`** under server’s public key.

`ChangeCipherSpec` indicates that client is updating cipher suite to be used on this session.

- Sent using SSL Change Cipher Spec. Protocol.

Optional (only when client is authenticated): **`ClientCertificate`**, **`ClientCertificateVerify`** messages.

Finally, **`ClientFinished`** message.

- MAC on all messages sent so far (both sides).
- MAC computed using **`master_secret`**.

M4: ChangeCipherSpec, …

ChangeCipherSpec indicates that server is updating cipher suite to be used on this session.

- Sent using SSL Change Cipher Spec. Protocol.

Finally, **ServerFinished** message.

- MAC on all messages sent so far (both sides).
- MAC computed using **master_secret**.
- Server can only compute MAC if it can decrypt **pre_master_secret** in M3.

SSL Handshake Protocol Run

1. Is the client authenticated to the server in this protocol run?
   - No!
2. Can an adversary learn the value of `pre_master_secret`?
   - No! Client has validated server's public key; To learn `pre_master_secret` the server's private key is needed to decrypt ClientKeyExchange
3. Is the server authenticated to the client?
   - Yes! ServerFinished includes MAC on nonces computed using key derived from **`pre_master_secret`**.

SSL/TLS Applications

Secure e-commerce using SSL/TLS.

Client authentication not needed until client decides to buy something.

SSL provides secure channel for sending credit card information.

Client authenticated using credit card information, merchant bears (most of) risk.

Widely deployed (de-facto standard).

Firewalls

Introduction

Cryptographic mechanisms protect data in transit (confidentiality, integrity).

Authentication protocols verify the source of data.

We may also control which traffic is allowed to enter our system (ingress filtering) or to leave our system (egress filtering).

Access control decisions based on information like addresses, port numbers, ...

Firewall

Firewall: a network security device controlling traffic flow between two parts of a network.

Often installed between an entire organisation's network and the Internet.

Can also be installed in an intranet to protect individual departments.

All traffic has to go through the firewall for protection to be effective.

– Dial-in lines, wireless LANs!?

Purpose

Firewalls control network traffic to and from the protected network.

Can allow or block access to services (both internal and external).

Can enforce authentication before allowing access to services.

Can monitor traffic in/out of network.

Types of Firewalls

Packet filter

Stateful packet filter

Circuit-level proxy

Application-level proxy

Packet Filter

Inspect headers of IP packets, also TCP and UDP port numbers.

Rules specify which packets are allowed through the firewall, and which are dropped.

- Actions: bypass, drop, protect (IPsec channel).

Rules may specify source / destination IP addresses, and source / destination TCP / UDP port numbers.

Rules for traffic in both directions.

Certain common protocols are difficult to support securely (e.g. FTP).

Example

TCP/IP packet filtering router.

- Router which can throw packets away.

Examines TCP/IP headers of every packet going through the Firewall, in either direction.

Packets can be allowed or blocked based on:

- IP source & destination addresses
- TCP / UDP source & destination ports

Implementation on router for high throughput.

Stateful Packet Filter

Packet filter that understands requests and replies (e.g. for TCP: SYN, SYN-ACK, ACK).

Rules need only specify packets in one direction (from client to server – the direction of the first packet in a connection).

Replies and further packets in the connection are automatically processed.

Supports wider range of protocols than simple packet filter (eg: FTP, IRC, H323).

Stateful Packet filter & FTP

Client sends ftp-request to server

Firewall stores connection state

- FTP-Server Address
- state of connection (SYN, ACK, ...)

If correct FTP-server tries to establish data connection, packets are not blocked.

Circuit-level Proxy

Similar to a packet filter, except that packets are not routed.

Similar to gateway using IPsec in tunnel mode.

Incoming TCP/IP packets accepted by proxy.

Rules determine which connections will be allowed and which blocked.

Allowed connections generate new connection from firewall to server.

Similar specification of rules as packet filter.

Application-level proxy

Layer-7 proxy server.

“Client and server in one box”.

For every supported application protocol.

SMTP, POP3, HTTP, SSH, FTP, NNTP...

Packets received and processed by server.

New packets generated by client.

Application-level proxy

Complete server & client implementation in one box for every protocol the firewall should handle.

Client connects to firewall.

Firewall validates request.

Firewall connects to server.

Response comes back through firewall and is also processed through client/server.

Large amount of processing per connection.

Can enforce application-specific policies.

Firewall Policies

Permissive: allow by default, block some.

- Easy to make mistakes.
- If you forget something you should block, it's allowed, and you might not realise for a while.
- If somebody finds find a protocol is allowed, they might not tell you ....

Restrictive: block by default, allow some.

- Much more secure.
- If you forget something, someone will complain and you can allow the protocol.

Firewall Policies – Examples

Permissive policies: Allow all traffic, but block ...

- Irc
- telnet
- snmp
- …

Restrictive policies: block all traffic, but allow ...

- http
- Pop3
- Smtp
- ssh
- …

Rule Order

A firewall policy is a collection of rules.

Packets can contain several headers (→ IPsec).

When setting a policy, you have to know in which order rules (and headers) are evaluated.

Two main options for ordering rules:

– Apply first matching entry in the list of rules.

– Apply the entry with the best match for the packet.

Typical Firewall Ruleset

Allow from internal network to Internet:

- HTTP, FTP, HTTPS, SSH, DNS

Allow reply packets

Allow from anywhere to Mail server:

- TCP port 25 (SMTP) only

Allow from Mail server to Internet:

- SMTP, DNS

Allow from inside to Mail server:

- SMTP, POP3

Block everything else

Firewall Location

A Firewall can only filter traffic which goes through it.

Where to put, for example, a mail server?

Requires external access to receive mail from the Internet.

- Should be on the inside of the firewall

Requires internal access to receive mail from the internal network.

- Should be on the outside of the firewall

Solution: “perimeter network” (aka DMZ).

Intrusion Detection Systems

Reminder: Security Strategies

Prevention: take measures that prevent your assets from being damaged.

Detection: take measures so that you can detect when, how, and by whom an asset has been damaged.

Reaction: take measures so that you can recover your assets or to recover from a damage to your assets.

Comment

Cryptographic mechanisms and protocols are fielded to prevent attacks.

Perimeter security devices (e.g. firewalls) mainly prevent attacks by outsiders.

Although it would be nice to prevent all attacks, in reality this is rarely possible.

New types of attacks occur: denial-of-service (where crypto may make the problem worse).

We will now look at ways of detecting network attacks.

Vulnerability Assessment

Examines the “security state” of a network:

- Open ports
- Software packages running (which version, patched?)
- Network topology
- Returns prioritized lists of vulnerabilities

Only as good as the knowledge base used.

- Have to be updated to handle new threats

Vulnerability Assessment Methods.

- Software solutions (ISS Scanner, Stat, Nessus etc.)
- Audit Services (manual Penetration tests etc)
- Web based commercial (Qualys, Security Point etc)

Intrusion Detection Systems

An IDS consists of a set of sensors gathering data, located on the hosts or on the network.

Sensors managed from a central console, where data is analyzed, intrusions are reported, and reactions may be triggered.

Two approaches for IDS: misuse detection and anomaly detection.

Protect communications between sensors and console, signature database and logs generated.

Needs secure scheme for getting signature updates from the IDS vendor.

Misuse Detection

Based on attack signatures:

- specific patterns of network traffic or activity in log files that indicate suspicious behaviour.

Example signatures might include:

- a number of recent failed login attempts on a sensitive host;
- a certain pattern of bits in an IP packet, indicating a buffer overflow attack;
- certain types of TCP SYN packets, indicating a SYN flood DoS attack.

Method used by all commercial IDS products.

Misuse Detection

Rules based on security policy, known vulnerabilities of particular OS and applications. known attacks.

Only as good as the information in the database of attack signatures:

- new vulnerabilities not in the database are constantly being discovered and exploited;
- vendors need to keep up to date with latest attacks and issue database updates; customers need to install these;
- large number of vulnerabilities and different exploitation methods, so effective database difficult to build;
- large database makes IDS slow to use.

Anomaly Detection

Statistical Anomaly Detection (or behaviour-based detection) uses statistical techniques to detect penetrations and attacks.

First establish base-line statistical behaviour: what is “normal” for this system?

Then gather new statistical data and measure the deviation from the base-line.

If a threshold is exceeded, issue an alarm.

Anomaly Detection

Example: monitor the number of failed login attempts at a sensitive host over a period;

- if a burst of failures occurs, an attack may be under way;
- or maybe the admin just forgot his password?

False positives (false alarm): attack is flagged when one is not taking place

False negatives: attack was missed because it fell within the bounds of normal behaviour

False negatives are also a major issue in misuse detection.

Anomaly Detection

IDS does not need to know about security vulnerabilities in a particular system; detects deviation from normal behaviour.

Problem: normal behaviour may overlap with forbidden behaviour

– Legitimate users may deviate from normality, causing false positives (e.g. user works late, forgets password, starts to use a new application).

– If the base-line is adjusted dynamically, an attacker may be able to gradually change this base-line so that the final attack does not generate an alarm.

Host-based & Network-based IDS

Network-based IDS (NIDS): looks for attack signatures in network traffic.

Host-based IDS (HIDS): looks for attack signatures in log files of hosts

- E.g. monitors system, event, and security logs on Windows and syslog in Unix environments.

The most effective IDS System will make use of both kinds of information.

There is a trend towards to host-based IDSs.

Honeypots

Technology used to track, learn and gather evidence of hacker activities

Strategically placed systems designed to mimic production systems, but not reveal “real” data

Definition:

- “… a resource whose value is being attacked or compromised”

Laurence Spitzner, “The value of honeypots”, SecurityFocus, October 2001

Honeypot Types

Level of Involvement

- Low Involvement: Port Listeners
- Mid Involvement: Fake Daemons
- High Involvement: Real Services

Risk increases with level of involvement.

Tools to detect honeypots are now available.