INTERNET SECURITY

OVERVIEW OF TECHNOLOGIES AND PROTOCOLS FOR IP BASED NETWORKS

Peter R. Egli
INDIGOO.COM

Contents

1. Network security issues

1. Secrecy (also dubbed privacy or data confidentiality, see RFC4949):

Make sure only sender and receiver of information can read it (restrict availability of information or legibility of information).

2. Privacy (see RFC4949):

Unlike secrecy privacy expresses the right of an individual to choose the degree to which it wants to share information with others.

3. Authentication / authorization:

Ascertain unambiguously the identity of someone (authentication) and what he is allowed to do an what not (authorization). Authentication uses message authentication codes (challenge response).

4. Non-repudiation:

„Nicht-Anfechtbarkeit“. Protection against false denial of involvement in a communication (e.g. denial of involvement in banking transaction).

5. Data integrity:

Assure that data is not altered (maliciously) on its transmit path. Uses one-way hash functions (MD5, SHA-1).

6. Protection agains DOS / dDOS attacks:

Protect system from attacks that render target system inaccessible by legitimate user.

7. Social engineering / social hacking:

Low tech and easiest approach. Exploit people‘s helpfulness and cooperativeness.

2. Security versus safety

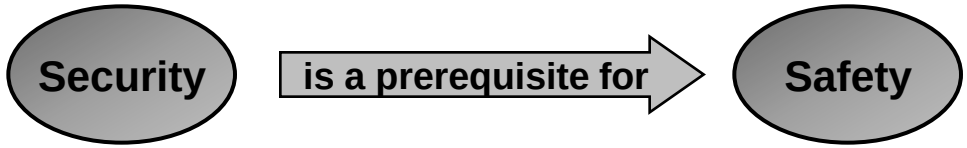
The distinction between the terms security and safety is often very subtle.
There is, however, a clear distinction between the 2 in regulated markets (medical, aerospace, industrial).

Safety is also related to security in that the lack of security may pose a safety risk (absence of IT security may lead to a system that is compromized which in turn may not be safe anymore).

Safety:
To be safe requires measures to prevent accidents (cause harm to humans or machines).
Examples: Redundant systems to guarantee availability, fire extinguisher.

Security:
Security requires measures to prevent fraud, crime, illegal activities.
Examples: Firewalling, security policy, use of encryption.

Security is a necessary but not sufficient prerequisite for achieving safety (a system without security is probably unsafe, but a system with security is not necessarily safe).



3. DOS - Denial of Service attacks (1/8)

Some common threads:

Virus:
Malicious addition (infection) of code to an existing code.

Trojan Horse:
Malicious program whose real function is camouflaged.

Worm:
Like virus, but distributes and starts on its own. Worms are best distributed in monocultures (same operating system on all machines).

Stealth virus:
Able to disguise the modifications it has done to the system (purpose: foil anti-virus programs).

Killer packet („Tschernobyl-gram“):
Causes a crash (meltdown) of the victim.

Mail bomb:
Mail with huge attachment to fill the victim‘s mailbox. Not so threatening anymore in the days of spam mail.

dDOS (distributed DOS attacks):
Attacker uses a range of intermediates (reflectors) to amplify the attack, often using IP directed network broadcast addresses. More elaborate dDoS attacks use a „Zombie Master„ program that remotely controls „Zombie“ programs planted throughout the network which carry out the attack.

3. DOS - Denial of Service attacks (2/8)

„Smurf“ (=ICMP flood, ICMP magnification attack, distributed DOS):

Procedure:

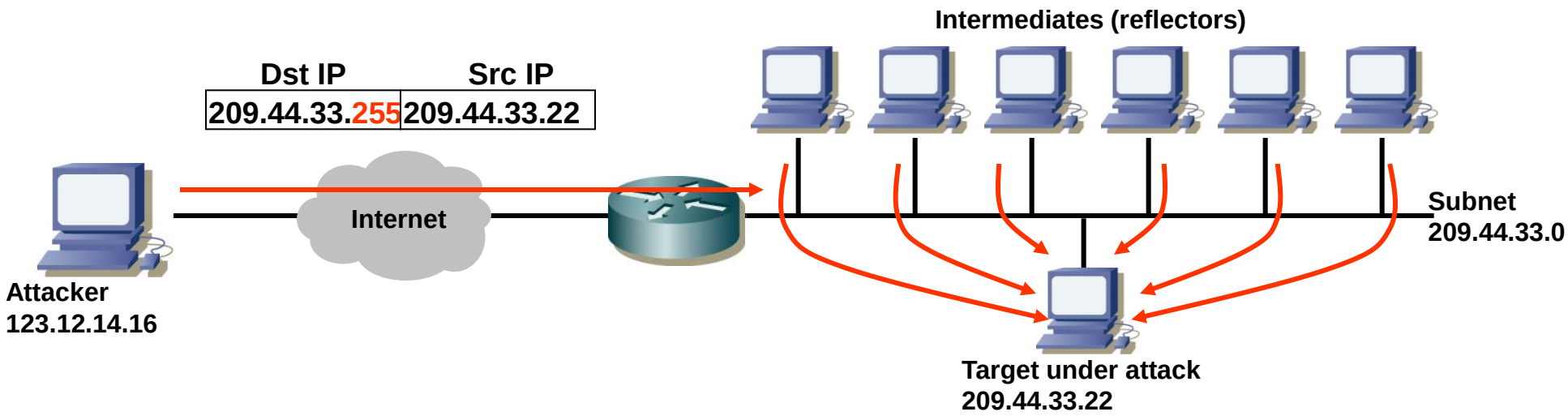
Flood network with pings (ICMP echo replies) with
IP destination address = directed network broadcast and
IP source address = target IP address (spoofed IP address).

Effect:

Consumption of target network bandwidth and target processing power.

Counter measures:

1. Configure subnet router such that directed net broadcasts are not routed.
2. Limit ICMP bandwidth (e.g. max. 2.5% of total bandwidth).



3. DOS - Denial of Service attacks (3/8)

„Fraggle“ (UDP flood):

Procedure:

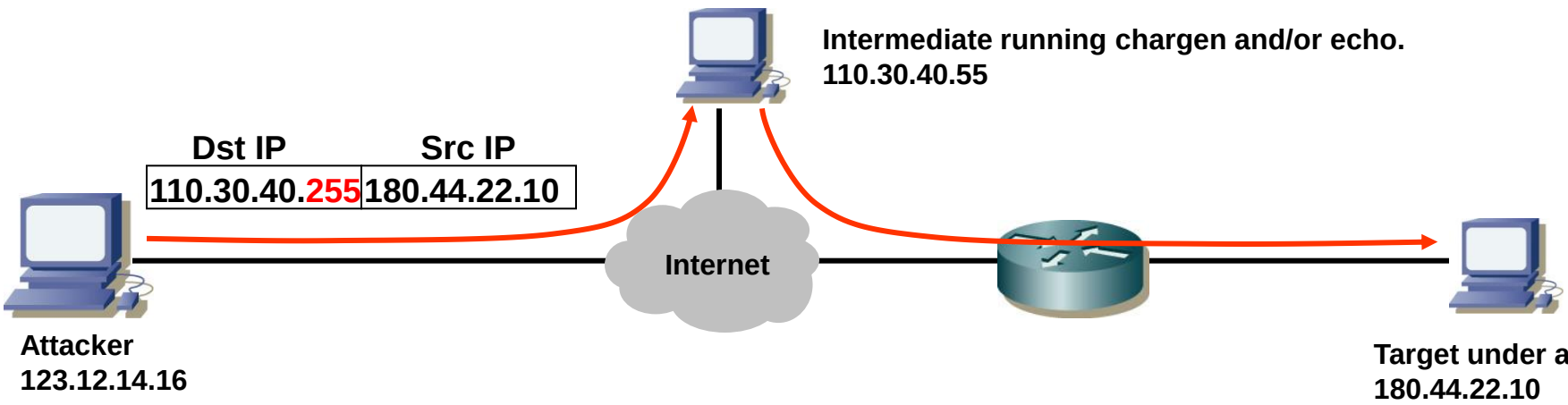
Like „smurf“, but instead ICMP use UDP packet magnification (source IP address = directed net broadcast address) using special UDP „small“ services (chargen character generation port 19, echo server port 7, time port 37 and daytime port 13).

Effect:

Network bandwidth consumption.

Counter measure:

1. Disable unnecessary and potentially harmful UDP services (chargen,echo).
2. Peer routers ingress filtering (RFC2267).
3. Restrict UDP upstream bandwidth.



3. DOS - Denial of Service attacks (4/8)

ICMP port unreachable flood:

Procedure:

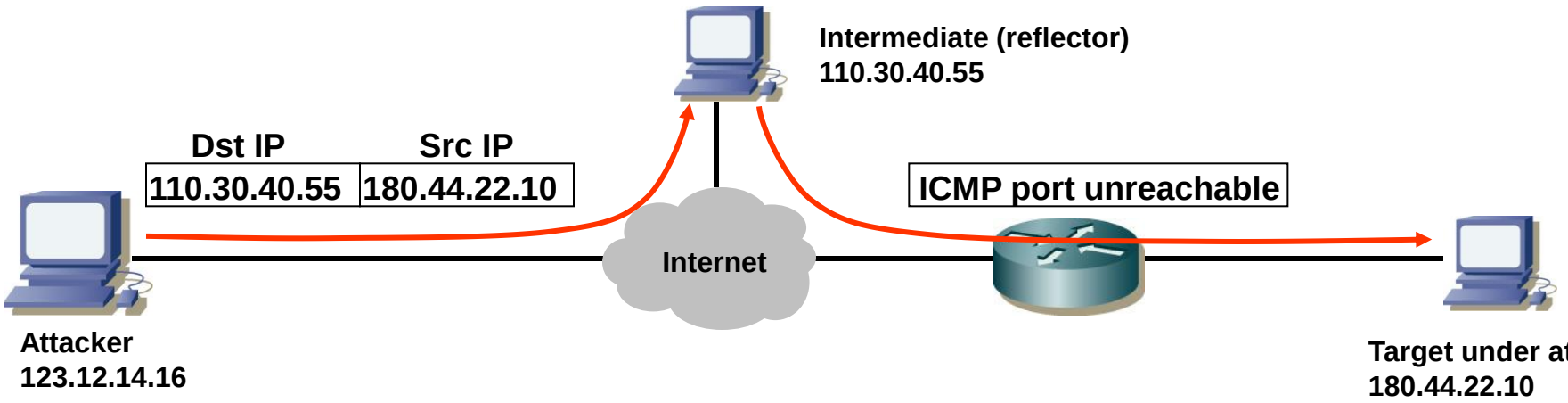
Send UDP packets to random ports on target thus generating „ICMP port port unreachable“ replies.

Effect:

Network bandwidth consumption.

Counter measure:

Disable ICMP port unreachable.



3. DOS - Denial of Service attacks (5/8)

SYN flood:

Procedure:

Flood target with faked TCP SYN packets.

Effect:

Flood will consume target resources thus making it unavailable for the intended user (SYN timeout ~= 3 min.).

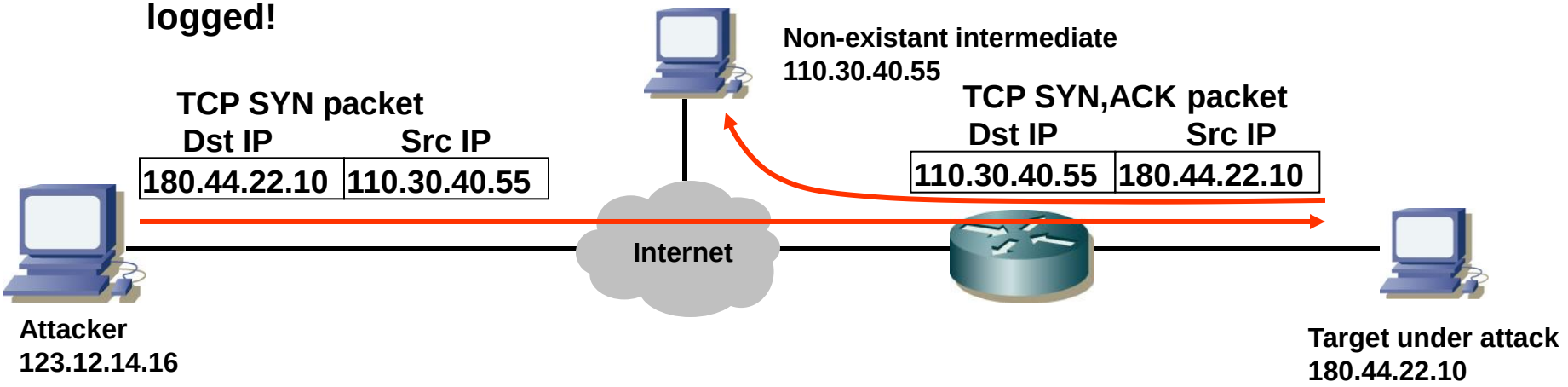
Counter measures:

Random early drop: randomly drop incomplete connections.

Stealth SYN attack:

Procedure:

Immediately after the SYN segment send a RST (reset) TCP segment (let Zombies do this job) which under UNIX does not generate log entries. Thus the attack is not logged!



3. DOS - Denial of Service attacks (6/8)

Land attack:

Procedure:

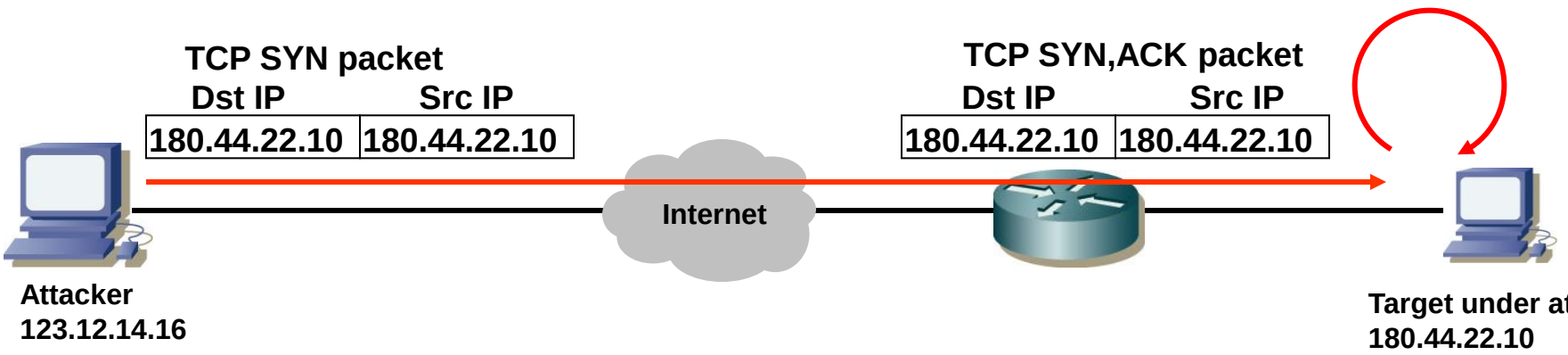
Send a TCP SYN packet with spoofed IP addresses where destination and source IP address are set to the target's IP address.

Effect:

Target sends ACK to itself creating an ACK war.

Counter measures:

OS patches.



3. DOS - Denial of Service attacks (7/8)

„Ping of death“:

Procedure:

Send ping, i.e. an ICMP echo request packet which is larger than allowed (max. 65kbytes).

Effect:

Cause the target system to crash or even meltdown („Tschernobylgram“).

Counter measure:

OS patch.

„Teardrop“:

Procedure:

Send overlapping IP fragments by purportedly adding wrong fragmentation information to the headers.

Effect:

Crash or meltdown.

Counter measure:

OS patch.

3. DOS - Denial of Service attacks (8/8)

Stack / buffer / heap overflow:

Procedure:

Cause a stack overflow and write malicious code into the stack frame and thus attain control over the target.

Effect:

Anything that malicious code can do (reproduce itself, send confidential data around, destroy data ...).

Counter measures:

OS patches.

Examples:

1998 Linux/ADM worm (buffer overflow) using a malformed DNS IQUERY

2001 CodeRed Windows IIS worm (URL stack overflow).

4. Other attacks (1/2)

TCP connection hijacking (MITM - Man In The Middle attack):

Take over a TCP connection and thus redirect traffic to an attacking host.

Web spoofing:

Redirect a web browser to another server delivering faked web pages. The user is tricked into believing that he is connected to the correct server and possibly discloses valuable information (passwords etc.).

ARP spoofing (use static ARP to prevent it):

Inject wrong ARP entries into the network.

ICMP attacks:

a. Spoofed ICMP source quench to make (legitimate) source reduce traffic to legitimate target.
b. Spoofed ICMP redirect to make (legitimiate) source use an alternate (invalid) route.
c. ICMP destination / port unreachable to ascertain existing hosts and open ports.

RIP / OSPF attack:

Generate spoofed RIP or OSPF packets to re-route traffic (man in the middle attack).

DNS poisoning:

Send phony (forged) DNS responses to the target. The target (client, server) caches the DNS record.

Exponential attacks:

All attacks that use amplification (zombies, deflectors) to amplify the effect.

4. Other attacks (2/2)

IP fragmentation:

Send IP fragmented packets with overlapping fragments etc. thus disconcerting the target („Tschernobyl-gram", system meltdown).

Firewalking:

Purpose:

1. Find open ports on firewall (which are not filtered).
2. Map (discover) hosts behind a (packet filtering) firewall.

Procedure:

Use forged traceroute program („firewalk“ utility) that uses UDP for traceroute. The port numbers are chosen such that when the traceroute packet hits the firewall they are let through (e.g. DNS port 53).

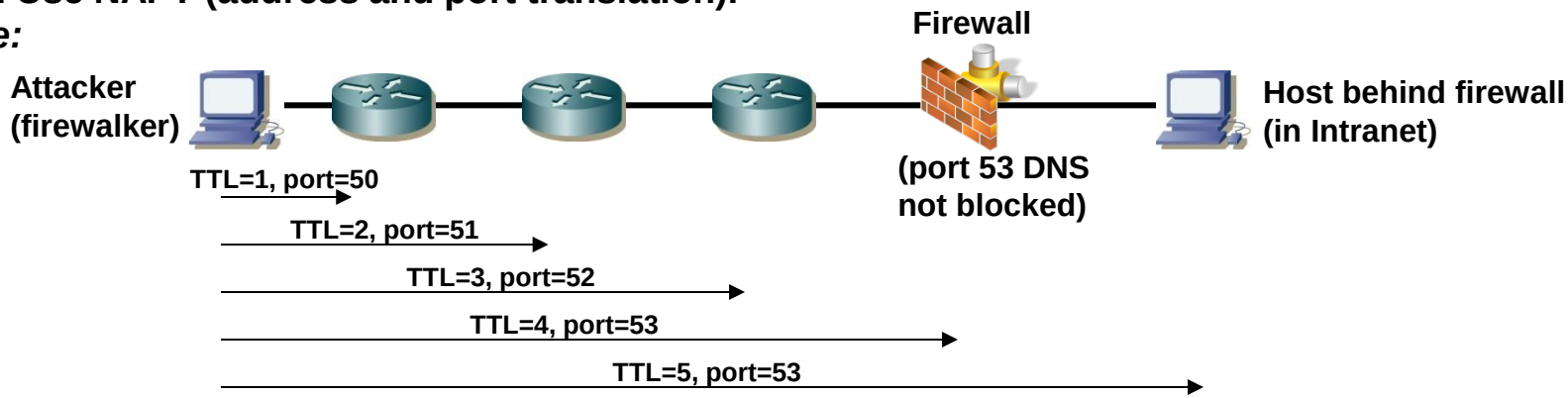
Effect:

Map (discover) hosts behind firewall that can serve as bridgehead for further attacks.

Counter measures:

1. Firewall should block outgoing ICMP TTL Exceeded messages.
2. Use NAPT (address and port translation).

Example:



5. Attacker tools

Address scanners:

Scan reachable IP addresses in the target network.

Port scanners:

Scan reachable (open) transport ports on reachable hosts.
May be used to determinet OS.
E.g. open ports 135-139 (NetBIOS) indicate a Windows systems.
E.g. ports above 512 indicate a UNIX systems.
Stealth scan (only send TCP-SYN) prevents application from logging the connection attempt (IDS is required to detect stealth scans).

Protocol analyzers:

Easy and fast extraction of useful information from sniffed traffic, e.g. for password extraction.

Banner analyzers:

Often servers disclose their version at startup of a session. Banner analyzers ascertain version of server and then exploit known security holes of this server.

Fingerprinting:

Ascertain OS and/or network stack version of target and then exploit known security holes.

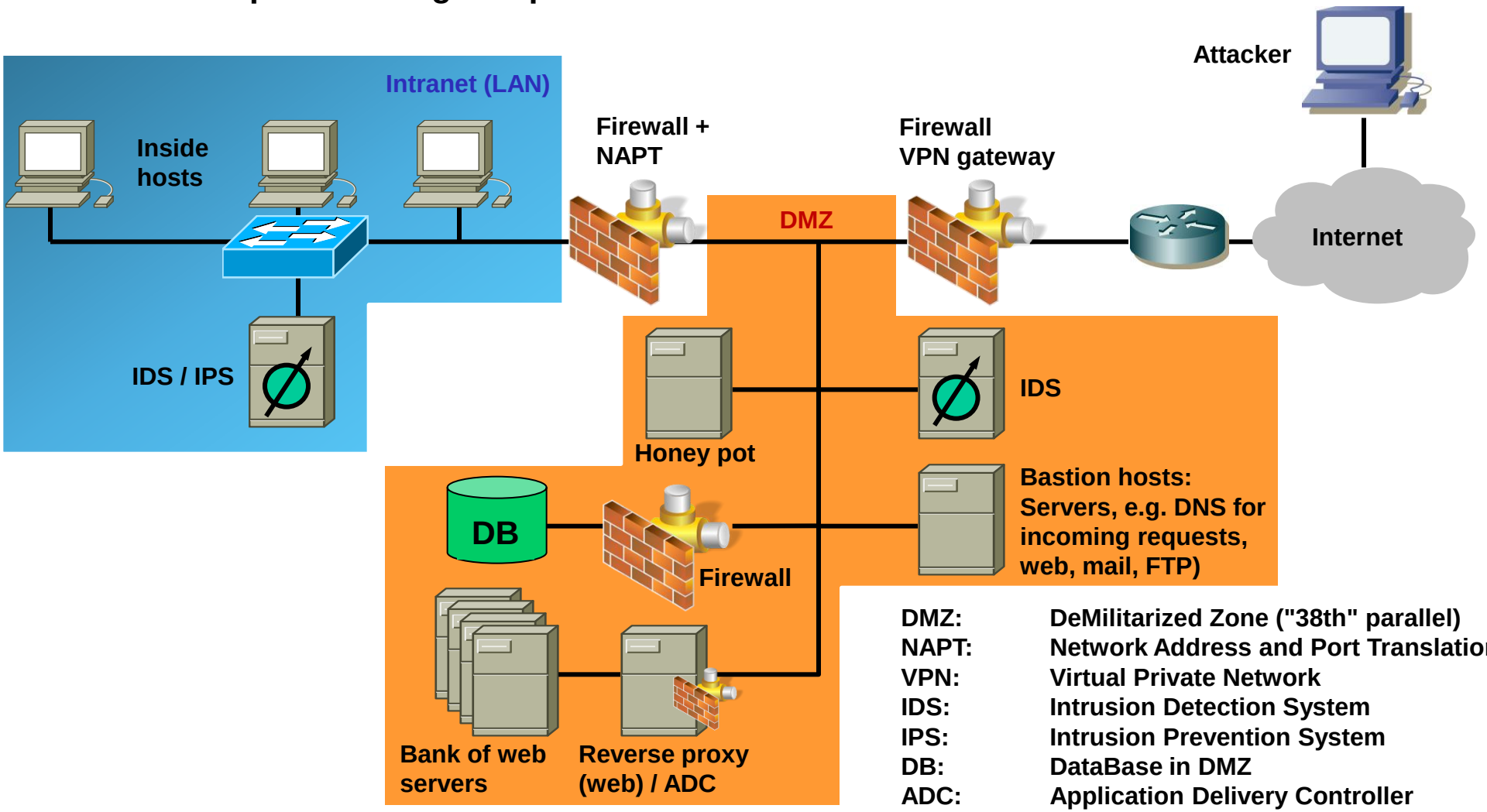
Unix Finger / who services:

Get useful information about who is logged in (and when not).

6. Network security architecture (1/7)

➔ The network is separated into an „inside“ (intranet) and „outside“ (DMZ and Internet).

➔ Firewalls are placed at ingress points to the different zones.



DMZ:	DeMilitarized Zone ("38th" parallel)
NAPT:	Network Address and Port Translation
VPN:	Virtual Private Network
IDS:	Intrusion Detection System
IPS:	Intrusion Prevention System
DB:	DataBase in DMZ
ADC:	Application Delivery Controller

6. Network security architecture (2/7)

Components of network security architecture:

Firewalls:

Firewalls are placed at ingress / egress points of networks or subnets.
Firewalls may be combined with the NAPT function. The use of private IPv4 addresses (IPv6: unique local addresses) avoids that packets leak into the Internet due to wrong routes.

DMZ (DeMilitarized Zone):

Problem / dilemma:

Servers with access from the Internet (web, mail, FTP) are complex and thus inherently insecure (provide more attack vectors).
If these servers are placed inside (Intranet), they are best protected (firewall) but if the servers are compromised (hacked), the hacker has access to the intranet.
If they are placed outside (direct access from Internet without firewall) they are least protected.
→ Thus a special zone („no man‘s land“ between „wild“ public Internet and protected intranet) is created which is carefully monitored as it contains the servers that are exposed to the outside.

6. Network security architecture (3/7)

IDS - Intrusion Detection System (1/2):

An IDS complements a firewall by monitoring the network for (suspicious) traffic (e.g. firewalking, port scans, spoofing, unusually small TTL values, IP fragments, overlapping TCP segments, network sniffers etc.).

Conceptually an IDS consists of an A-, C-, D- and E-box each with a different function.

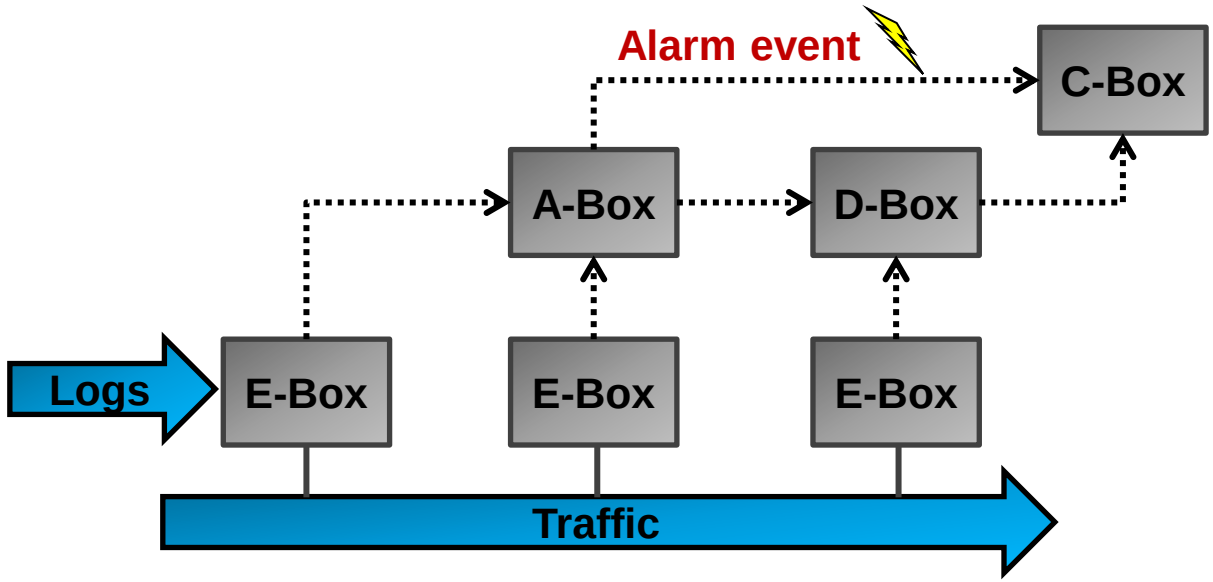
A-box: Network activity Analysis (on data gathered by E-box and stored in D-box)

C-box: Countermeasure mechanism (IPS, modify filter lists in routers etc.)

D-box: Storage mechanisms (loggers, store Data produced by E- and A-boxes)

E-box: Event generators (sensors)

IDS also protect against „enemy within“ (it is estimated that 80% of attacks come from within).



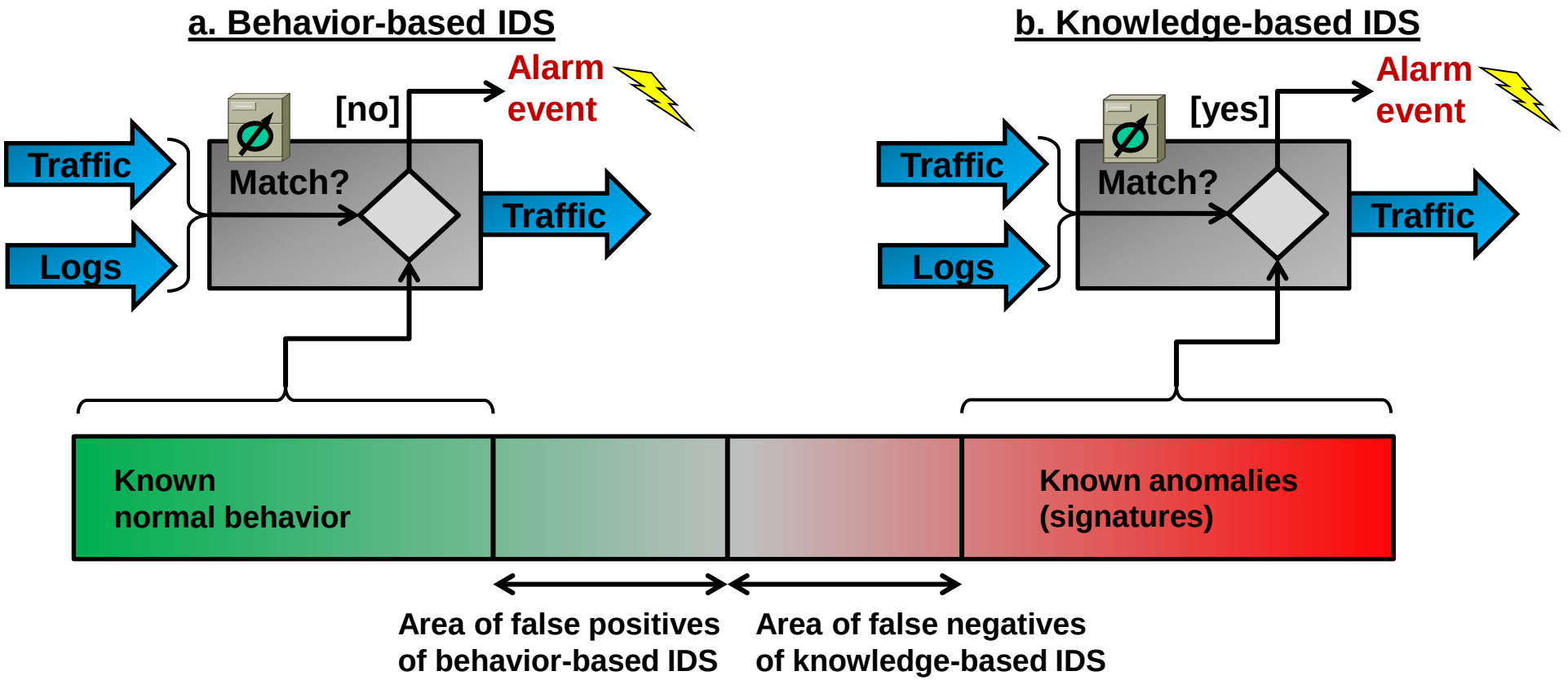
6. Network security architecture (4/7)

IDS - Intrusion Detection System (2/2):

With regard to functionality, IDS can be classified into:

a. Behavior-based IDS (=anomaly-based IDS)

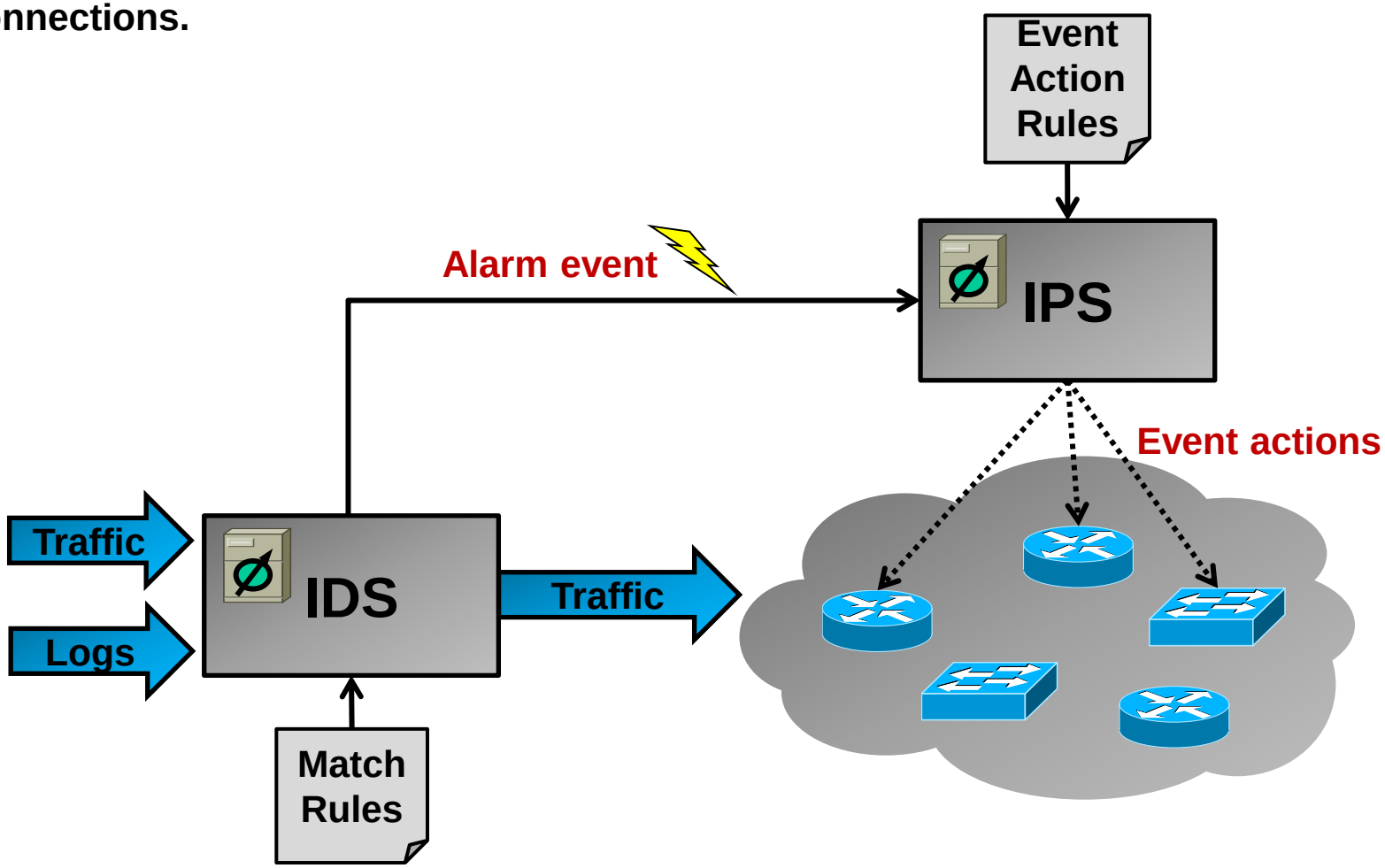
b. Knowledge-based IDS (=signature-based IDS)



6. Network security architecture (5/7)

IPS - Intrusion Prevention System:

An IPS complements an IDS (or is part of an IDS) by actively taking actions upon a detected intrusion. Example actions are changing firewall filter rules, blocking traffic and resetting TCP connections.



6. Network security architecture (6/7)

Honey pot:

A honey pot is a decoy or booby trap to tie down an attacker and gather information about the attacker (example honey pot SW: honeyd, snort).
Honey pots must be used with care and need to be well secured so they do not become security risk (present another attack vector).

Reverse proxy:

A reverse proxy is used to hide web servers from the public Internet. Only the reverse proxy is visible to the outside and relays the requests to internal server(s).
Reverse proxies control the access to web servers (e.g. block web mail access), but are also used for caching and load balancing.
An ADC (Application Delivery Controller) is a special network appliance that acts as a reverse proxy but fulfills additional functions like load balancing, web application filtering (XSS, SQL injection etc.), security through protocol termination (HTTP, TCP, FTP etc.), compression, caching, failover, SSL offloading (termination of SSL connections) and connection pooling (save connection resources).

Bastion host:

Bastion hosts are servers that are exposed to the public Internet, i.e. provide services like DNS, web, mail or FTP to the Internet. Usually they are placed into the DMZ.
Bastion host servers need to be specially secured (hardened servers).

6. Network security architecture (7/7)

DANGER! Potential backdoor # 1:

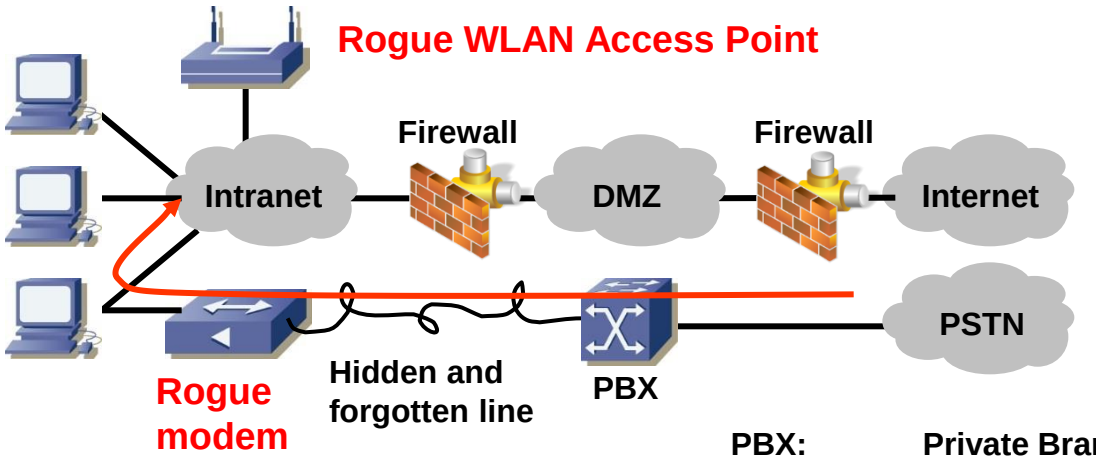
PBX (Private Branch Exchange, enterprise phone switch) or modem lines that are tied directly to a modem in the LAN are a potential backdoor. An attacker may call the modems number ("war dialing") and thus establish a direct connection into the intranet / LAN bypassing the firewalls.

Often the PBX and the LAN are administered by different departments / people so the LAN-administrators do not know about the modem lines.

DANGER! Potential backdoor # 2:

WLAN access points connected to the intranet / LAN without permission (rogue access points) present another (more recent) security problem.

Countermeasure: Rogue WLAN AP detection.



PBX: Private Branch Exchange (phone switch)
PSTN: Public Switched Telephony Network

7. Firewalls (1/14)

Purpose of a firewall:

A firewall controls and filters access to and from a protected network.
Usually a firewall is placed between the trusted internal network (LAN, intranet) and the untrusted external network (Internet).

Securing a firewall:

A firewall is a crucial component in the security perimeter. Thus a firewall should not run user programs such as web servers since these present a point of attack (attack vector).

Firewall types:

Over time different types of firewalls emerged.

1. **Packet filters (1st generation firewalls):**
   → **Filter rules applied to individual packets**
2. **Circuit level gateways (2nd generation firewalls):**
   → **Monitor connections and flows of packets**
3. **Application level gateways (3rd generation firewalls):**
   Proxy, monitor and inspect the application traffic
4. **Stateful Multi-Layer Inspection gateways SMLI:**
   → **Combination of 1., 2. and 3.**
5. **Distributed firewalls:**
   → **Host based firewall with central management**
6. **Firewalls with NAPT:**
   → **Firewalls combined with NAPT functionality**

7. Firewalls (2/14)

1. Packet filters (1/3):

Function:

Simple packet filters where the first firewall types that were deployed (1st generation firewall). Packet filters usually run on layer 3 (IP) and execute the filter rules defined in *ACLs* (Access Control List). These firewalls are *stateless* because the filtering is applied to individual packets only (filtering does not depend on the state of previous packets of the same connection).

Filter rules:

The filter rules can range from simple source / destination IP address filtering to more complex rules that filter packets based on TCP flags, protocols or combinations thereof.

- **→ Filter on source / destination IP address**
- **→ Filter on source / destination transport port**
- **→ Filter on protocol such as ICMP, UDP, TCP, SCTP, IPSec**
- **→ Filter on TCP flags (ACK, SYN, FIN, RST)**
- **→ Filter on combinations of source/destination IP, source/destination port and protocol**

Filter rule management:

Over time the table with filter rules grows. Usually more rules are added than deleted.
This may lead to large tables with rules that can not be deleted anymore because otherwise there may be the the risk that a vital service is disabled.
Therefore it is very important to document filter rules and tag them with a timestamp so it is possible in the future to decide whether a rule is still needed or not.

7. Firewalls (3/14)

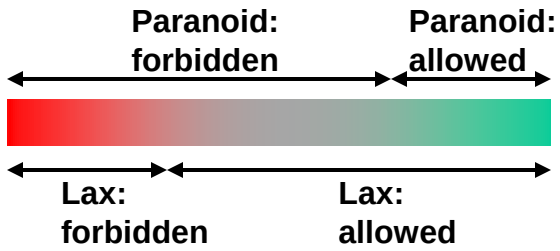
1. Packet filters (2/3):

Filter policies:

There are 2 main policies for filter rules:

Lax policy: Allow everything that is not explicitly forbidden.

Paranoid: Forbid everything that is not explicitly allowed.



The administrator has to trade-off security (paranoid policy) against usability (lax policy). If the firewall is too restrictive, there is the danger that employees establish ways around the firewall (rogue WLAN APs, SSH tunnels, HTTP tunnels).

Packet filters + NAPT:

Packet filters are often combined with address and port translation (NAPT, NPAT). NAPT is used to conserve public IP addresses. Additionally NAPT provides a minimal level of security as it hides internal IP addresses.

Where to use packet filters:

Packet filtering can be applied at many points (not only company firewalls). E.g. an ISP should install ingress filtering (perform filtering of traffic from customers into the Internet which has source addresses that do not belong to ISP, i.e. filter spoofed IP packets).

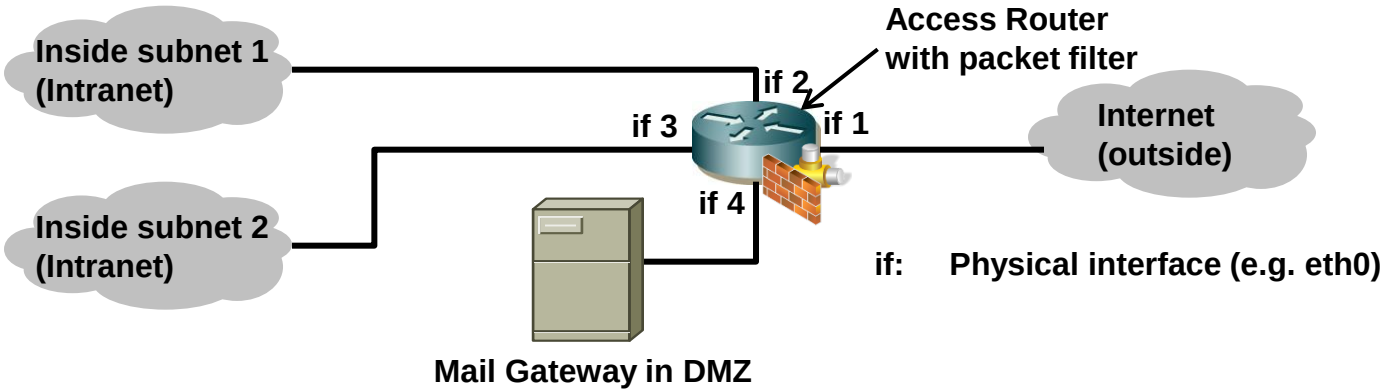
Packet filter examples:

Examples: ipchains, ipf / ipfw, iptables (Linux).

7. Firewalls (4/14)

1. Packet filters (3/3):

Filter rules consist of match elements (address, ports, flags etc.) and an action that defines what to do with a matching packet.



Example ingress filtering on if 1 (for ingress packets, i.e. outside→if1):

Action	Source IP	Source port	Dest. IP	Dest. port	Flags	Comment
block	*{subnet1}*	*	*	*		*Block spoofed packets.*
block	*{subnet2}*	*	*	*		*Block spoofed packets.*
block	*{subnet3}*	*	*	*		*Block spoofed packets.*
allow	*	*	*mail GW*	*25*		*Allow incoming SMTP to mail gateway.*
allow	*	*	*{subnet1}*	*	*ACK*	*Allow acks for outgoing TCP connections from subnet 1.*
allow	*	*	*{subnet2}*	*	*ACK*	*Allow acks for outgoing TCP connections from subnet 2.*

7. Firewalls (5/14)

2. Circuit level gateways (1/2):

Function:

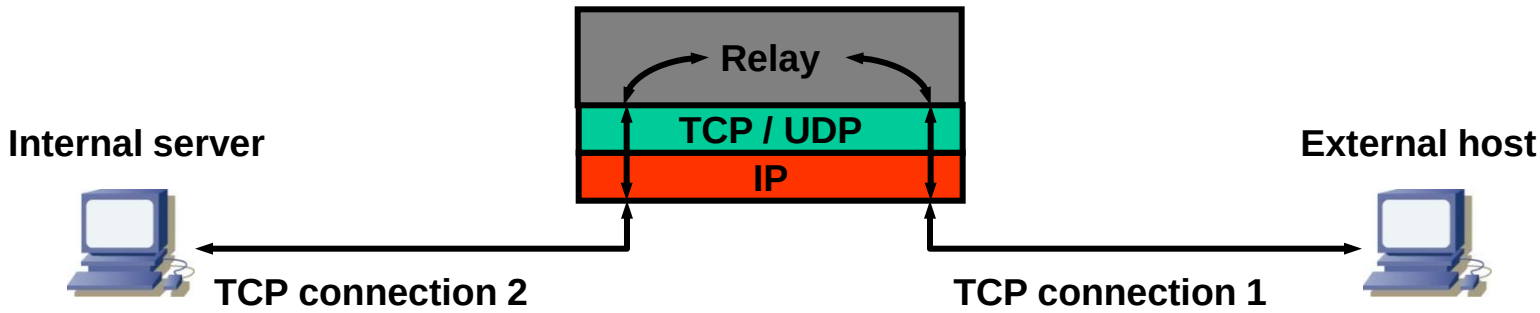
Circuit level gateways (2nd generation firewalls) relay incoming TCP connections and UDP sessions (similar to a proxy server).

The gateway terminates (is endpoint of) the external TCP connection and establishes a new TCP connection to the internal host.

Thus the internal host is not exposed and can therefore not be attacked, i.e. sending malformed packets does not harm the internal host. IP tricks like wrong fragments and firewalking probes are terminated at the gateway and the internal host is protected.

Additionally a circuit level gateway closely monitors the external TCP connection (checks if it has a correct behavior):

- → TCP flags
- → Sequence numbers
- → ACK numbers



7. Firewalls (6/14)

2. Circuit level gateways (2/2):

No routing allowed:

The circuit level gateway is not a router. Routing MUST be switched off.
If routing is switched on, the packets bypass the relay and the internal host is unprotected.
The gateway terminates TCP (and UDP) on one side and relays the payload to another TCP connection (or UDP packet) on the other side.

Examples:

SOCKS (RFC1928), tcprelay

7. Firewalls (7/14)

3. Application level gateways:

Function:

Application level gateways (3rd generation firewalls) monitor the traffic up to the application layer. Thus application level gateways are proxies.

Such gateways not only terminate transport connections, but also inspect the application traffic and only relay application payloads if these do not contain harmful data.

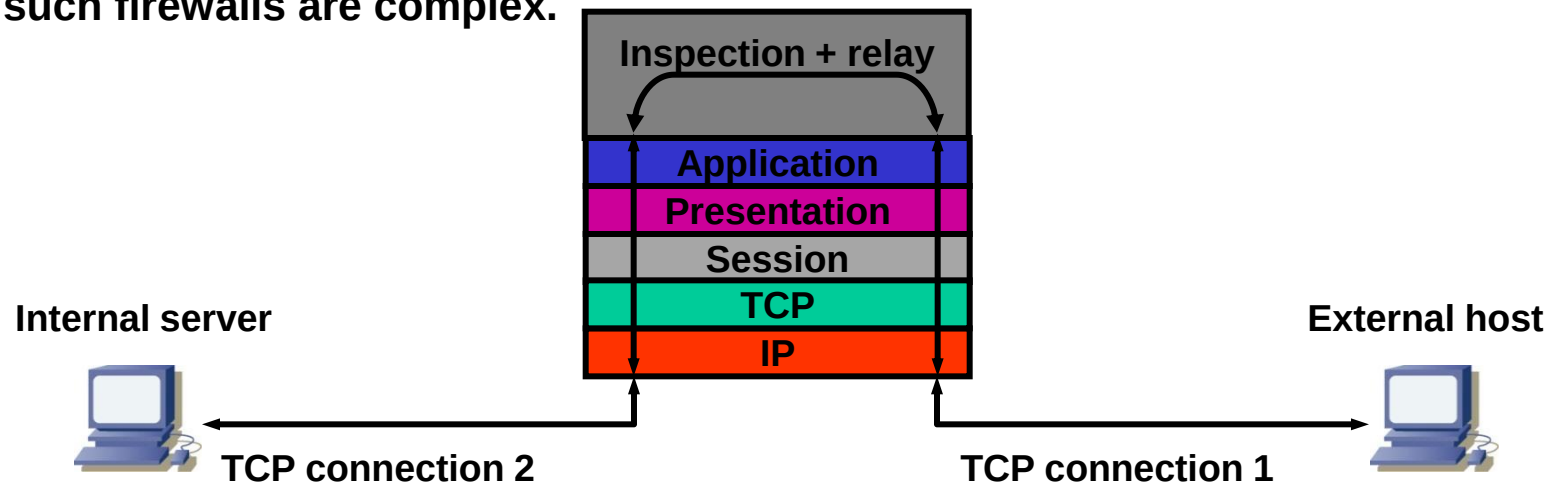
Examples of application level inspection functions:

→ Filter dangerous content such as *.exe email attachments

→ Filter banned FTP commands like STOR, e.g. when company policy disallows uploads

→ Filter active web page content like Javascript

Application firewalls must 'understand' all application protocols that pass through the firewall. Therefore such firewalls are complex.



Examples:

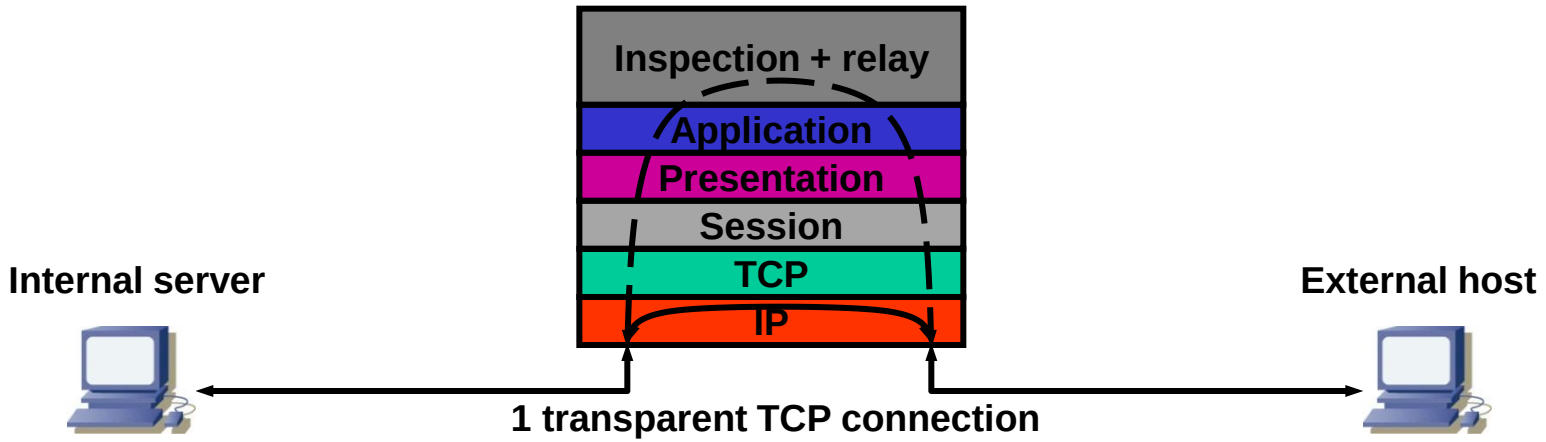
Web Application Firewall (WAF), AppArmor (Linux kernel module)

7. Firewalls (8/14)

4. Stateful Multi-Layer Inspection gateways (SMLI):

Function:

SMLI combines packet filers, circuit level gateway and application level gateway.
Such firewalls inspect the packets on OSI layer 2 through 7.
All packets are checked if they belong to a transport connection or session (state table).
SMLIs store the state of the connections and monitor it.
UDP and ICMP packets are dynamically assigned to a connection (e.g. for a DNS query (UDP) the SMLI expects a corresponding DNS respones packet (UDP).
Additionally SMLIs perform application level filtering.
Some protocols require the transport ports be dynamically opened. E.g. an FTP PORT or PASV command requires that the corresponding port be temporarily opened in the packet filter.



Example:

Checkpoint firewalls

7. Firewalls (9/14)

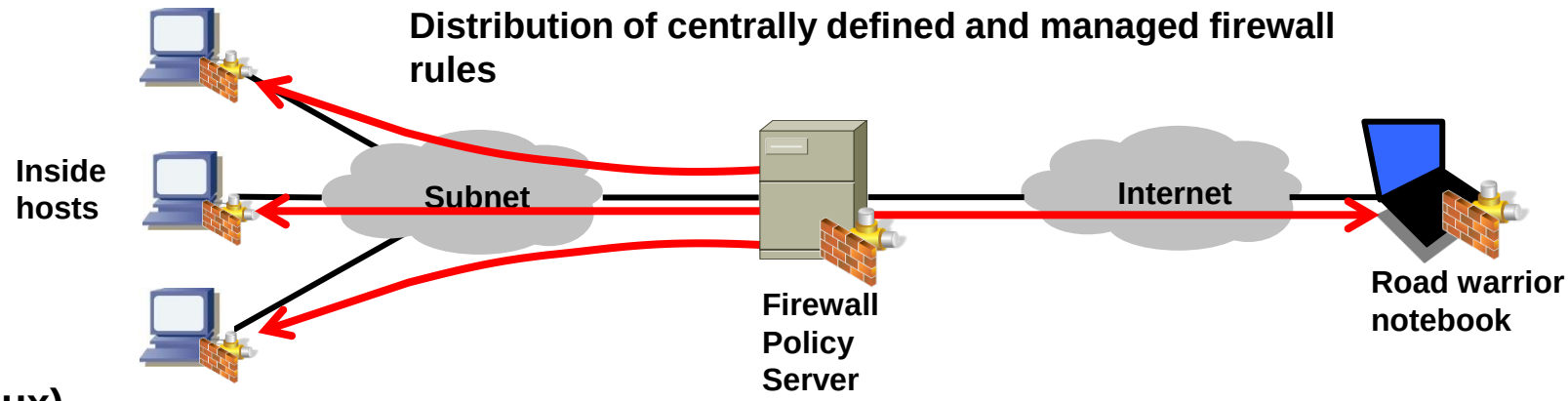
5. Distributed Firewalls (similar to personal firewall):

Function:

Distributed firewalls are host-based firewalls, i.e. every host runs its own instance of firewall. Distributed firewalls are similar to personal firewalls. The difference is that distributed firewalls are centrally managed, i.e. by centrally defining the filter rules and distributing them to all hosts.

Advantage:

Standard firewall appliances have the drawback of being a single point of failure. If the firewall breaks the entire security may be compromised.
Distributed firewalls protect each host individually.
These firewalls allow to protect also hosts that or not inside a topologically isolated space (road warrior notebooks).



Example:

IPFilter (Linux)

7. Firewalls (10/14)

6. Firewalls with NAPT (1/5):

A. Simple NAT function (without port translation):

→ *NAT for IP subnet address changes:*

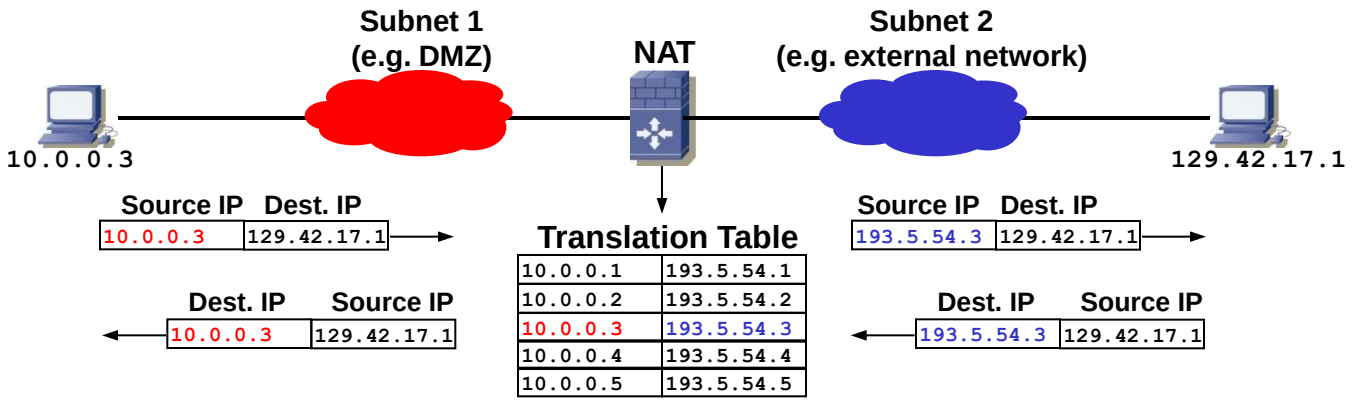
Simple NAT (Network Address Translation, i.e. replacement of an IP address by another one) may be used to change IP addresses in a subnet change without changing the routing entries in other subnets (IP address change is transparent to other subnets).

→ *NAT for mapping public IPs to private IPs in a DMZ:*

Another application of NAT is the placement of a firewall between public IPs (external network) and servers on a DMZ-network (e.g. 193.5.54.0/28 addresses (=public) are ‚NATed' to 10.0.0.0/28 addresses).

Static NAT: **The translation table entries are set up statically (by the sysadmin). Each Subnet 1 host is assigned a specific entry in the translation table.**

Dynamic NAT: **Each time a packet from Subnet 1 arrives at the NAT, the latter chooses a free entry in the translation table. The mapping exists as long as there are packets for the host in Subnet 1 flowing back and forth. The entries are timestamped and aged out and ultimately deleted once they are no longer in use.**



7. Firewalls (11/14)

6. Firewalls with NAPT (2/5):

B. NAPT = Network Address and Port Translation (1/4):

Other terms for NAPT (RFC3022):

NPAT (Network Port Address Translation).
Overloading (one public IP address is overloaded with multiple private internal IP addresses).
IP Masquerading (Linux).

Problem:

→ Imminent depletion of public IPv4 addresses.

Cause:

→ IPv4 addresses were initally organized in classes A throug E.
→ Large chunks of addresses (class B) were allocated to single organizations (class C range with 256 addresses was often too small, so organizations had to buy the next larger class B with 65'535 addresses, but did only use a small portion of that range).

Class	Range from	Range to	Mask	Hosts per network
A	0.0.0.0	127.255.255.255	255.0.0.0	16’777’216
B	128.0.0.0	191.255.255.255	255.255.0.0	65’536
C	192.0.0.0	223.255.255.255	255.255.255.0	256
D	224.0.0.0	239.255.255.255	-	(multicast addresses)
E	240.0.0.0	255.255.255.255	-	(experimental addresses)

7. Firewalls (12/14)

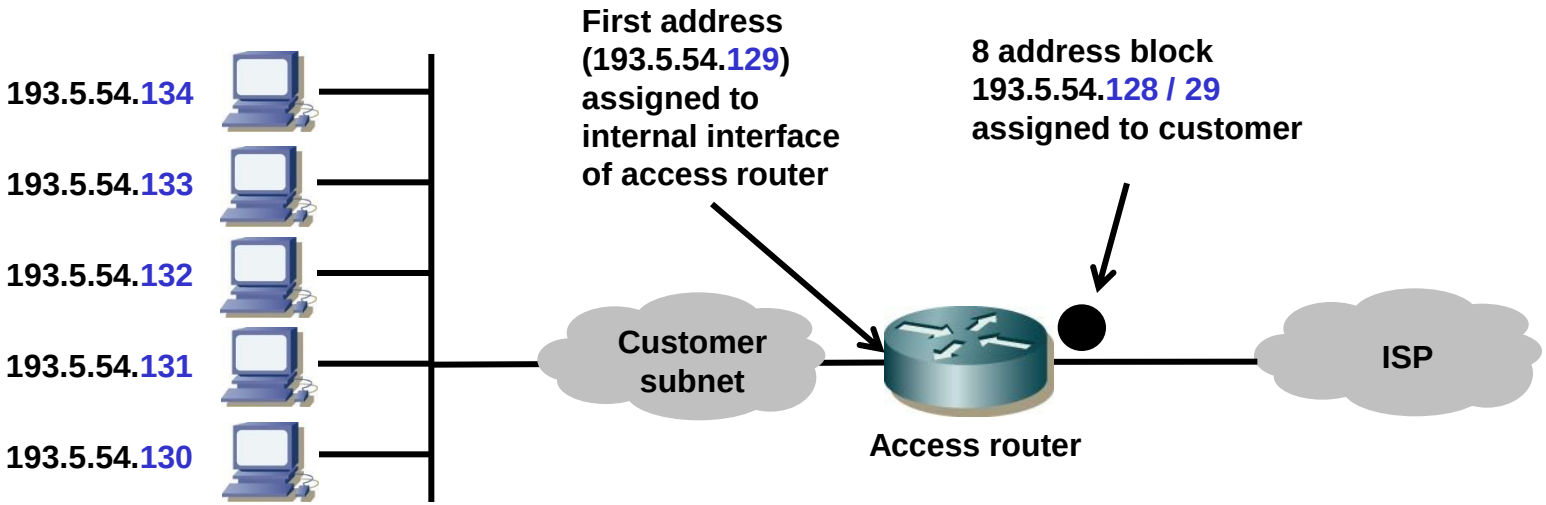
6. Firewalls with NAPT (3/5):

B. NAPT = Network Address and Port Translation (1/4):

Solution 1 for conserving IPv4 addresses:

CIDR (Classless interdomain routing, RFC1517 et.al.) was introduced where IP addresses were no longer assigned as full class A, B or C networks but in smaller chunks, e.g. 10 IP addresses out of a class A range (thus classless).

Internet routers needed to become able to route IP addresses from the same class to different locations (thus classless routing). This alleviated the problem but did not solve it.



7. Firewalls (13/14)

6. Firewalls with NAPT (4/5):

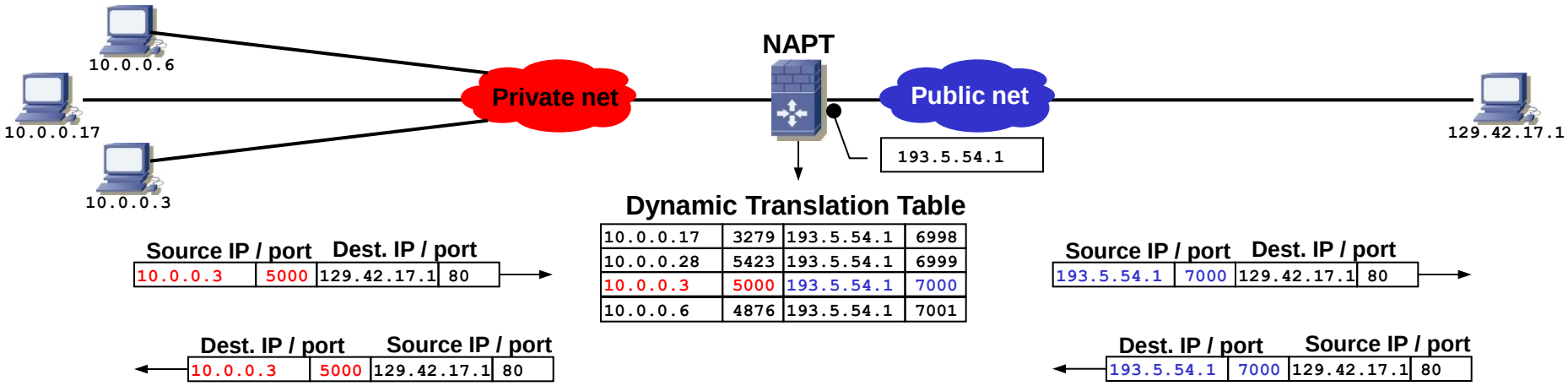
B. NAPT = Network Address and Port Translation (3/4):

Solution 2 for conserving IPv4 addresses:

The introduction of private IP addresses alleviated the problem further. RFC1918 defines 3 ranges of IP addresses that were set aside and that can be used by every organization (no need to buy these addresses):

Range from	Range to	Mask	Hosts per network
10.0.0.0	10.255.255.255	255.0.0.0 (8bit)	16'777'216
172.16.0.0	172.31.255.255	255.240.0.0 (12bit)	1'048'576
192.168.0.0	192.168.255.255	255.255.0.0 (16bit)	65'536

→ These addresses are not routed in the public Internet since they are private. Hosts with such IP addresses are either not reachable from the public Internet or need an address translation (NAPT).
→ With NAPT it is possible to hide such hosts behind one single public IP address (which is on the NAPT box):

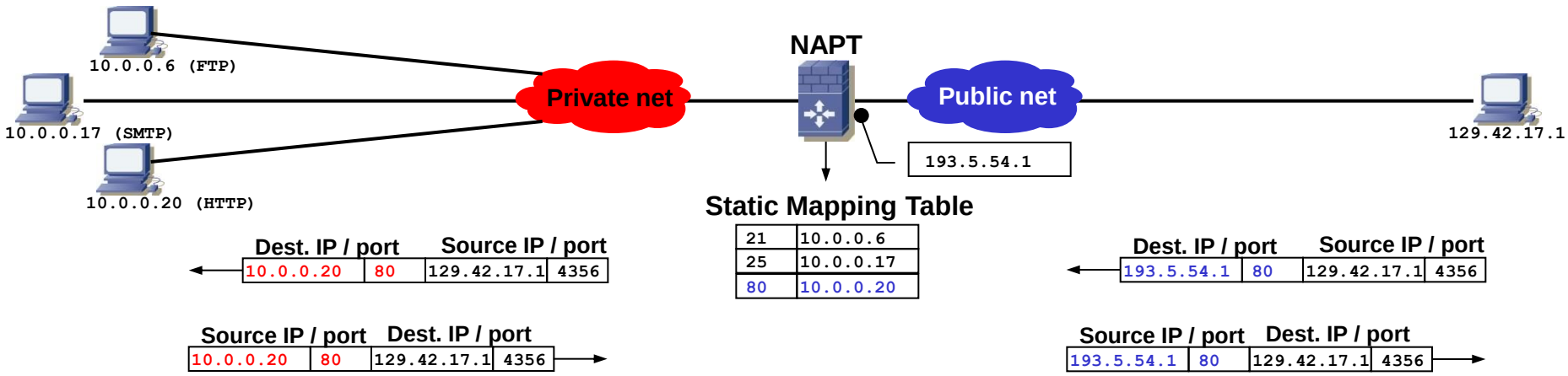


7. Firewalls (14/14)

6. Firewalls with NAPT (5/5):

B. NAPT = Network Address and Port Translation (4/4):

How to hide a server behind NAPT and still have the server accessible from the public Internet:



N.B.: It is not required to use private IP addresses when doing NAPT. Public non-published IP addresses can also be used for the private network as long as a NAPT translation is between the internal hosts and the Internet.

8. Some security guidelines (not exhaustive) (1/2)

1. Use DMZ:
Place servers (HTTP, SMTP, FTP) into a DMZ.

2. Block incoming HTTP traffic:
Block incoming HTTP and HTTPs (into LAN) unless really needed. This will foil attempts to tunnel protocols through the firewall through tunnels, e.g. using SOAP (port 80) or SSL (port 443). SOAP (Simple Object Access Protocol) = „anything over HTTP“!

3. Restrict inbound HTTP traffic to web server:
Only the web server should be allowed to receive inbound HTTP.

4. Minimize services:
Switch off any services / servers that are not needed. Every service / server represents a potential security hole (attack vector).

5. Security by obscurity:
Do not disclose information if not necessary. Hide as much information (server names, addresses, locations, operating systems, user names etc.) as much as possible, but do not rely on it. It is just another layer in the security perimeter.

6. Least priviledge:
Give services / servers and users least priviledge. If not necessary do not run servers / services as root. Otherwise if such a service / server is compromised the attacker gains root access.

8. Some security guidelines (not exhaustive) (2/2)

7. Log service:

Run logging service to log unusual activities in order to identify potential attacks. Use a separate disk partition for the log files in order to protect against DoS attacks where the log files swamp the entire machine.

8. Security vs. usability tradeoff:

Find a good tradeoff between security and usability. If users are disgruntled because they are unable to use certain services (e.g. unable to use FTP for downloading) they will find their way around the firewall and protection mechanisms (e.g. carry files on virus-infested storage devices into the company).

9. Don't be an attacker, don't be a victim:

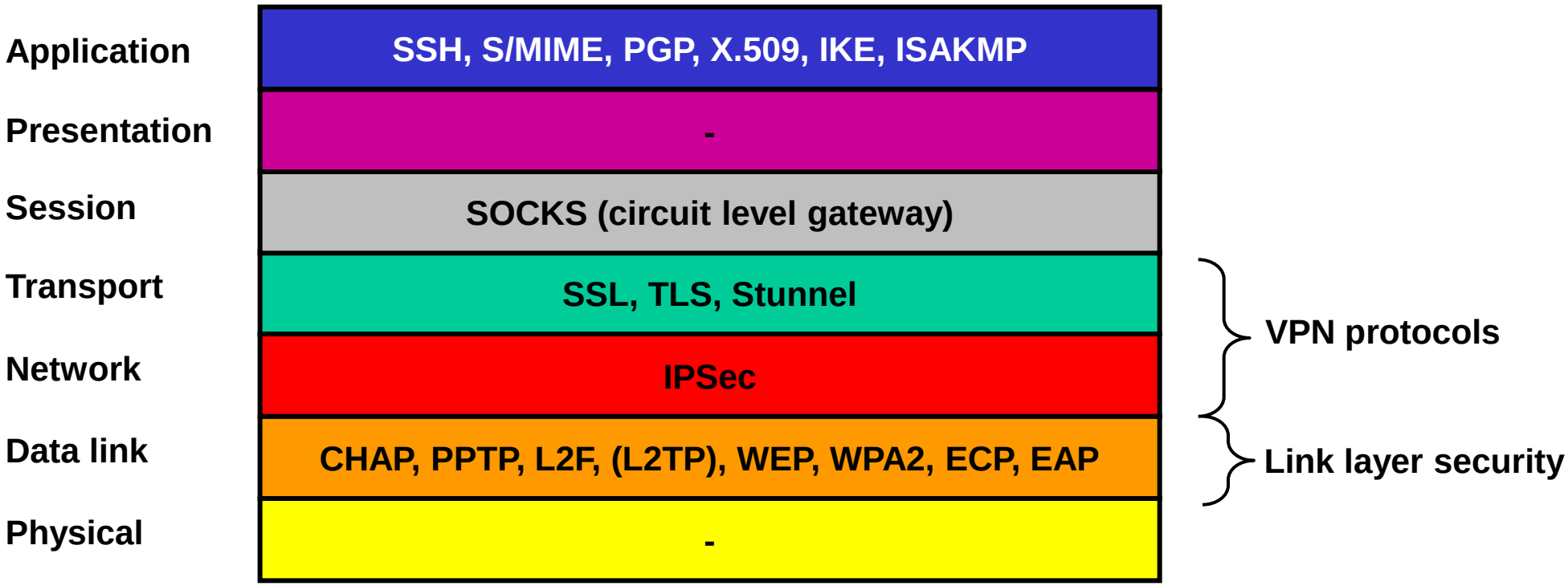
Avoid becoming a victim, but also an attacker (switch off services / servers that could become tools for the attacker for a dDoS attack). E.g. configure your mail server such that it can not become an open relay.

10. Stay informed:

Stay informed about security problems. Consult web resources such as CERT Computer Emergency Response Team http://www.cert.org/ frequently.

9. Network security protocols overview

There exist various security protocols on the different OSI layers.



CHAP:	CHallenge Authentication Protocol	IPSec:	IP Security	IKE:	Internet Key Exchange
PPTP:	Point to Point Tunneling Protocol	SSL:	Secure Socket Layer		
L2F:	Layer 2 Framing Protocol	TLS:	Transport Layer Security		
(L2TP:	Layer 2 Tunneling Protocol)	Stunnel:	Secure Tunnel		
WEP:	Wired Equivalent Privacy	SSH:	Secure SHell		
ECP:	Encryption Control Protocol	S/MIME:	Secure MIME		
EAP:	Extensible Authentication Protocol	PGP:	Pretty Good Privacy		
WPA2:	Wireless Protected Access	ISAKMP:	Internet Sec. Assoc. And Key Mgt. Prot.		

10. Secure Socket Layer SSL / TLS (1/3)

Function:

SSL / TLS provides secure communication between HTTP clients (browsers) and HTTP servers.

How it works:

SSL is a special layer between the application and transport protocol (TCP). SSL provides authentication (server and client authenticate each other or only the server authenticates towards the client), privacy (encryption) and message integrity.

→ SSL is not appl. transparent (appl. must be SSL aware and must use the SSL socket library).

→ TLS (Transport Layer Security) is the successor to SSL (TLS is very similar to SSL).

SSL / TLS enabled applications:

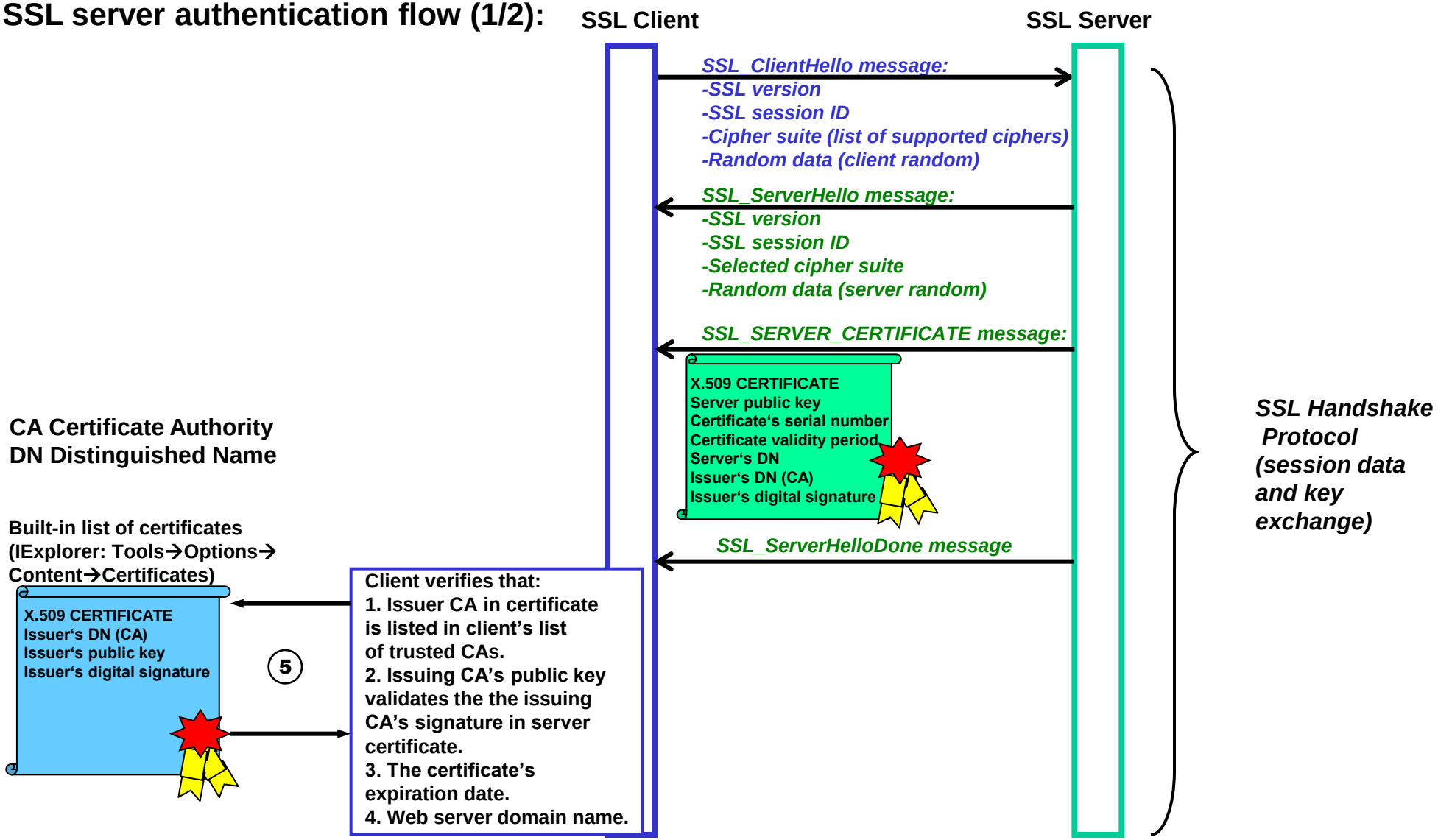
HTTPs, SMTPs, LDAPs, FTPs, TELNETs, POP3s

Transparent alternative to SSL / TLS:

Stunnel is an SSL wrapper for non-SSL aware applications (SSL tunnel proxy).

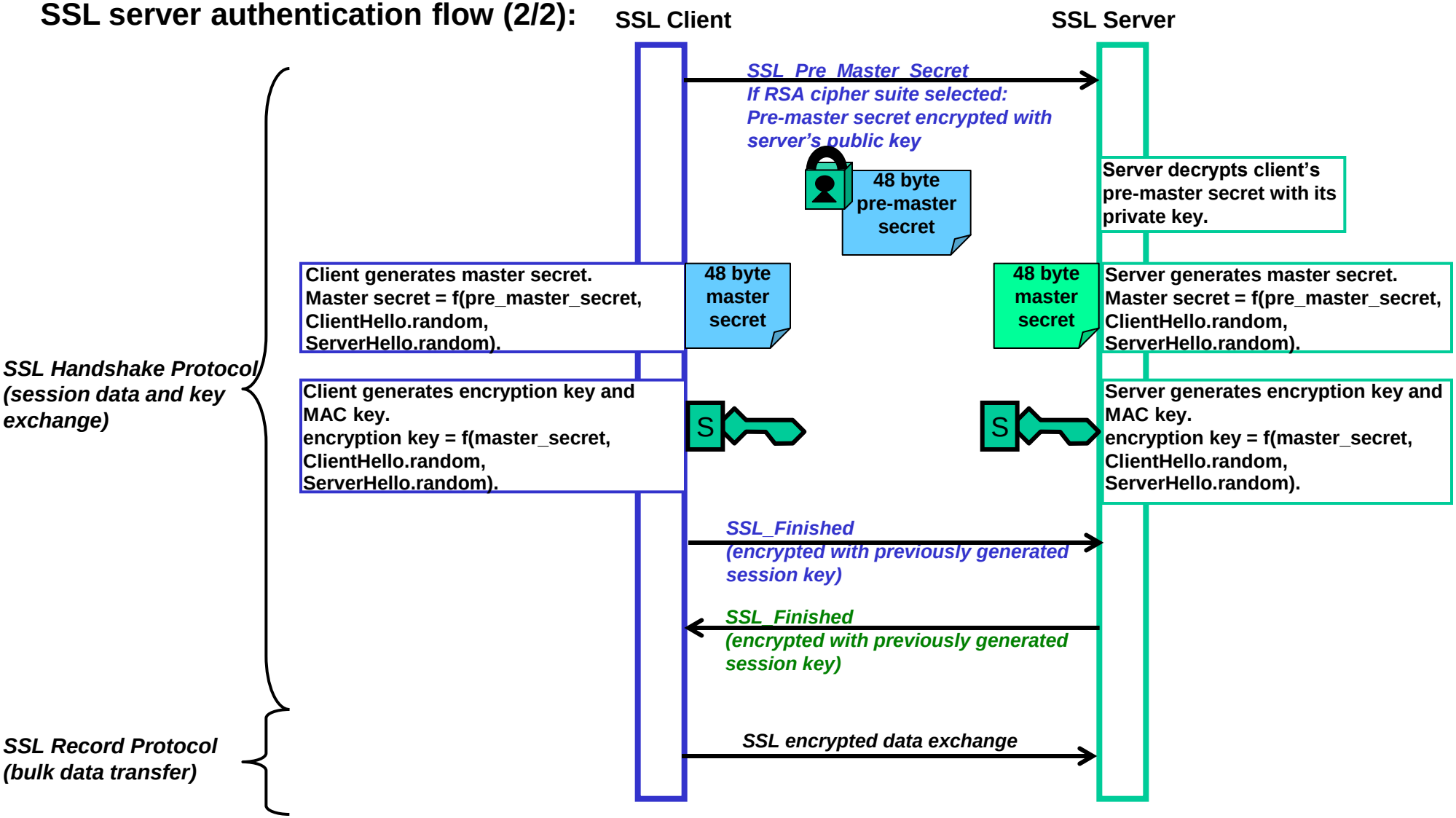
10. Secure Socket Layer SSL / TLS (2/3)

SSL server authentication flow (1/2):



10. Secure Socket Layer SSL / TLS (3/3)

SSL server authentication flow (2/2):



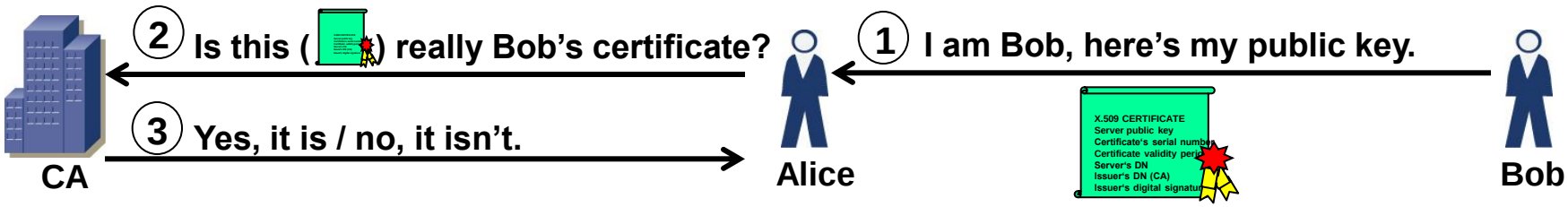
11. X.509 / Certificates / Certificate Authorities CA (1/2)

Simply getting a public key from the communication peer is not secure since an attacker could pretend to be the real communication partner. Certificates are used to ascertain the trustworthiness of a communication peer.

➔ Certificates provide a means to verify (authenticate) a public key (which is contained in a certificate). Certificates are issued by a trustworthy third party, so-called Certificate Authorities (CA).

➔ A public-key certificate is a digitally signed statement from a CA saying that the public key (and other information contained in the certificate) is trustworthy.

➔ The most common certificate protocol in use is X.509.

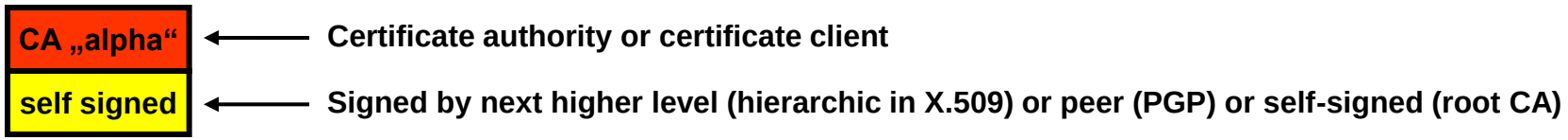


X.509 CERTIFICATE contains:

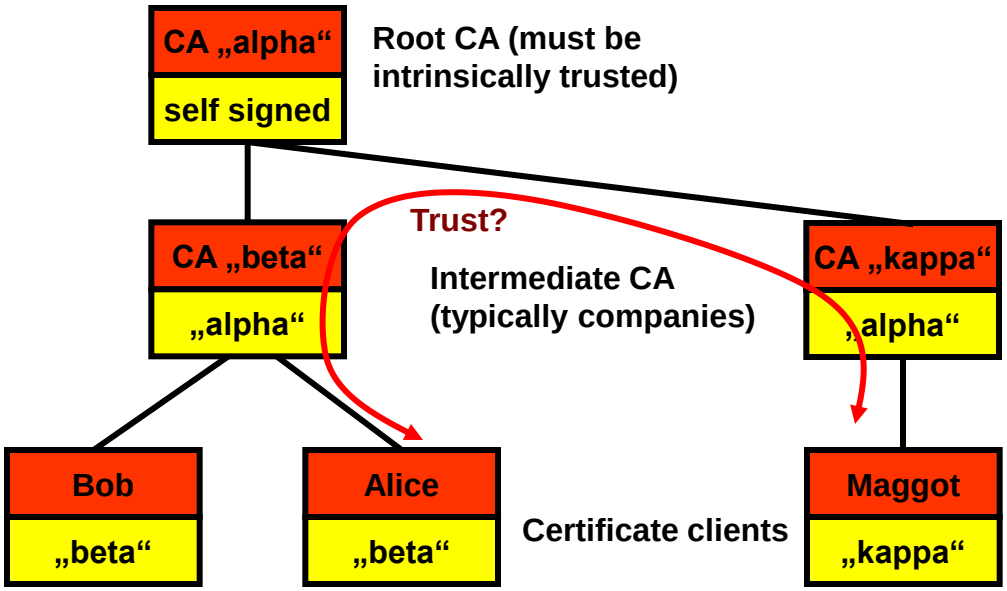
Server's public key:	**Public key of communication partner.**
Certificate's serial number:	**Unique identifier of certificate.**
Certificate validity period:	**Start and end date of validity of certificate.**
Server's DN:	**Distinguished name of server.**
Issuer's DN (CA):	**Distinguished name of certificate issuing authority.**
Issuer's digital signature:	**Certificate encoded with CA's private key. A user (Alice in the example above) uses the CA's public key (distributed through a certificate again) to decrypt the certificate and thus verify that it is sound.**

11. X.509 / Certificates / Certificate Authorities CA (2/2)

Trust is a relationship that can be established either hierarchically or without hierarchy (flat).
➔ X.509 uses a trust hierarchy (certificate chains).
➔ PGP uses a non-hierarchic web of trust (PGP Web of Trust) to verify a person‘s key (trustwortyness).



X.509 trust hierarchy



PGP web of trust