Communication protocols and network security

Security elements: IPsec, SSL and infrastructure

IPSec

- IP security protocol (security on the network layer)
- used to secure the link between two entities, used for VPN (virtual private network)!
- Security on network layer:
  - Hide all types of data
    (TCP segment, UDP segment, ICMP message, OSPF message etc.)
  - Ensuring source authentication
  - Integrity of data before the change
  - Protection from re-establishing communication
- RFC 2411: review of mechanisms and IPSec operation

Virtual Private Network(VPN)

- Companies on different geographic locations want high communication security . Solutions:

  1. Establishing a PRIVATE network: the company builds its own network completely separate from the Internet (expensive establishment and management – routers, links, infrastructure needed !)

  2. The company establish a VIRTUAL PRIVATE network (VNP) with the infrastructure of the public network:

     - Data on local (private) parts of the network is transmitted normally (IP),
     - Data sent on public parts of the network is protected (IPSec)

VPN: example
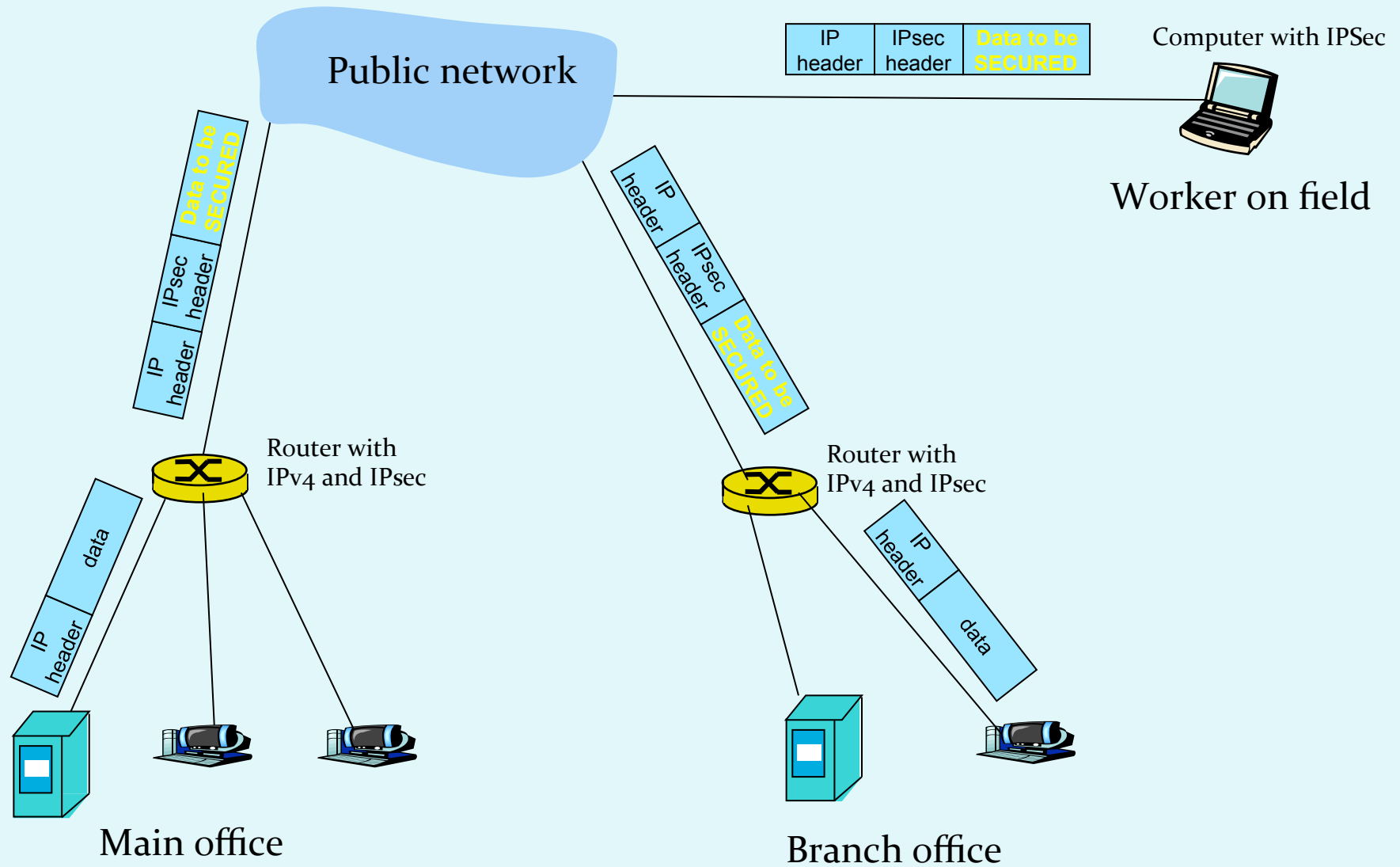
Pisec implementation

- IPSec mechanism offers two protocols for protection:
  - AH - *Authentication Header*
    - ensures source authentication and data integrity
  - *ESP - Encapsulation Security Payload*
    - ensures source authentication, data integrity AND confidentiality
- For each way of the IPSec communication it needs to be established a SA (Security Association)
  - example: the main and branch offices are using two-way communication. The main office also uses two-way communication with n workers on the field. How many SA they need to establish?

$$2 + 2n$$

Establishing SA

200.168.1.100

SA

193.68.2.23

IPsec

IPsec
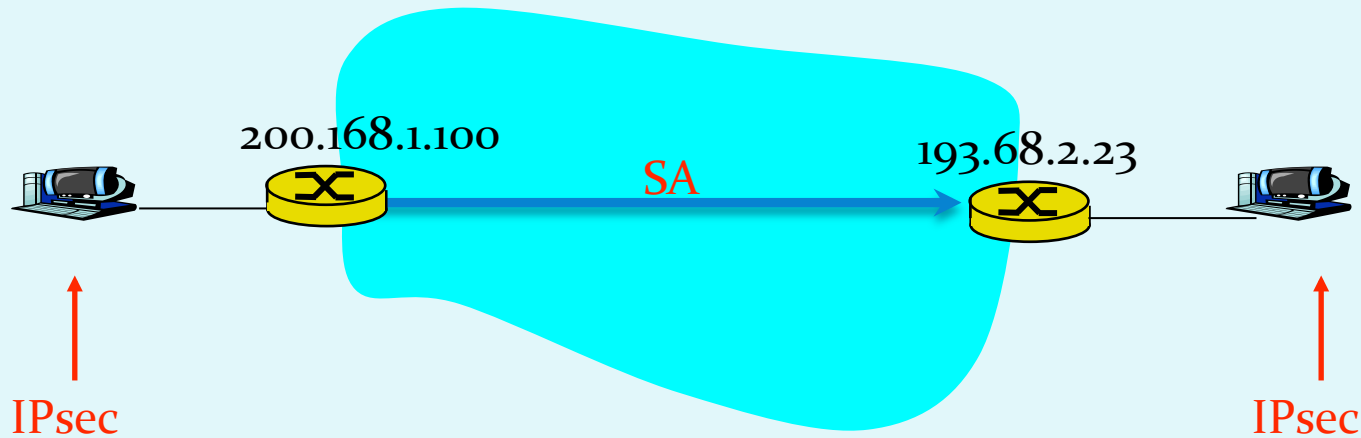
- Router has the SAD database(*Security Association Database*) where it keeps data about SA:
  - 32 bit ID SA, called SPI (Security Parameter Index)
  - Source and destination IP SA
  - Type of encryption(e.g .3DES) and key
  - Type of integrity test(e.g. HMAC/MD5)
  - Authentication key

2 ways of communication

- **transport mode** – implemented between the end-users (computer interfaces), protects protocol's upper layers. Transparently to the interface, it encrypts only data in the package.

- **tunnel mode** – Transparent to the end-user, router-router or router-user. It encrypts data and header.

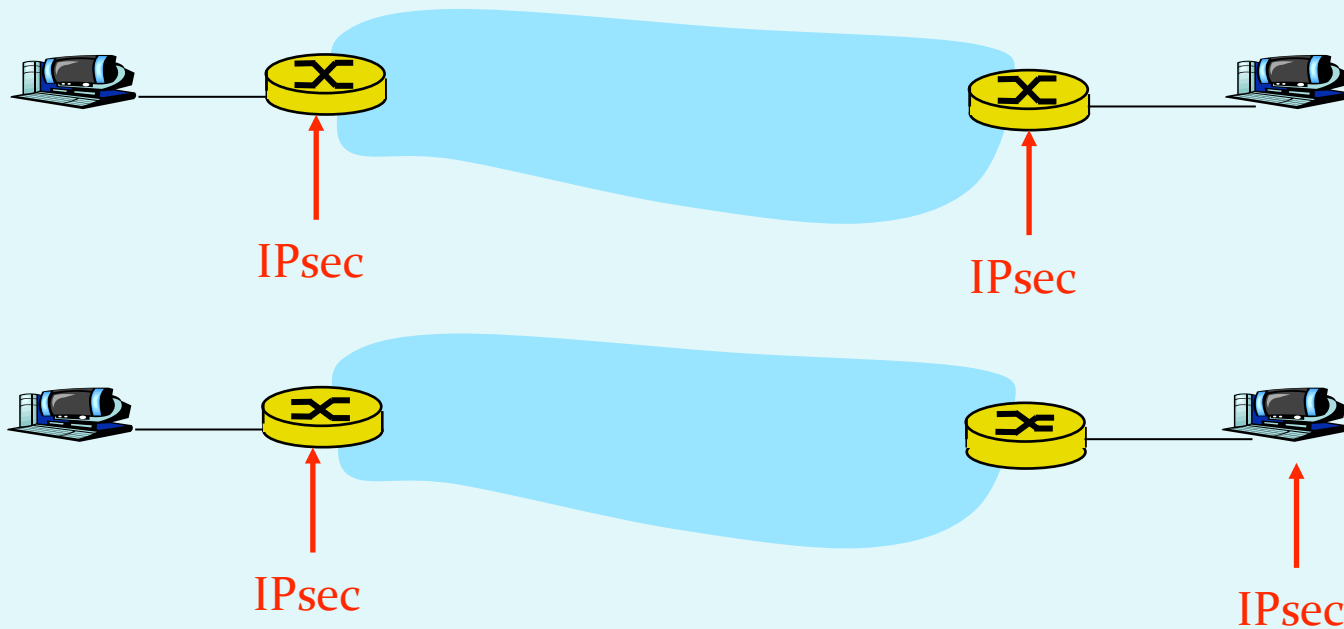| Transport mode with AH	Transport mode with ESP
Tunnel mode with AH	Tunnel mode with ESP

Most common!

IPsec Transport Mode

IPsec ... IPsec

- IPsec travels between end systems
- We protect only the upper layers

IPsec – tunneling mode

- IPsec is used at the end routers
- for customers is not necessarily to implement IPsec

IPsec datagram: tunnel mode and ESP

- Let's look at how the most common IPsec usage works
- Original data:

| original IP header	original IP data

IPsec datagram: tunnel mode and ESP

- The ESP header is added to the end of the datagram (the fill is needed for block coding, next header is a protocol, contained in the data)

- Result is encrypted (algorithm and key define SA!)
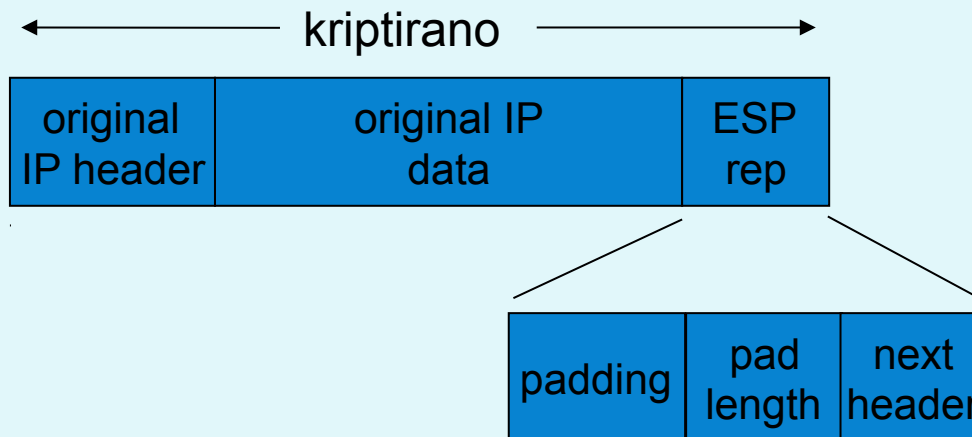
IPsec datagram: tunnel mode in ESP

- ESP header is added: result is "enchilada"
  (SPI - index SA, which is used to determine the setting,
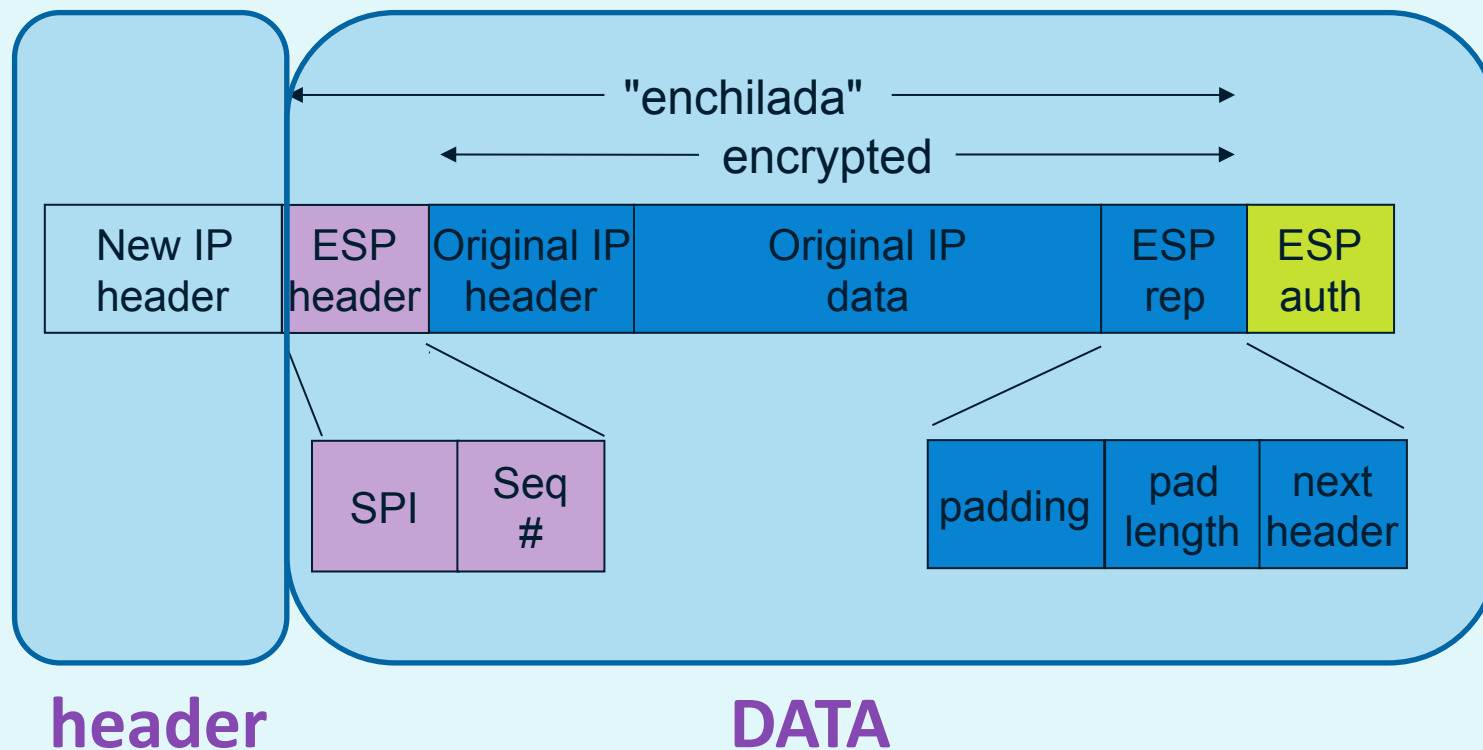 Seq # - protection against recurrence of communication)
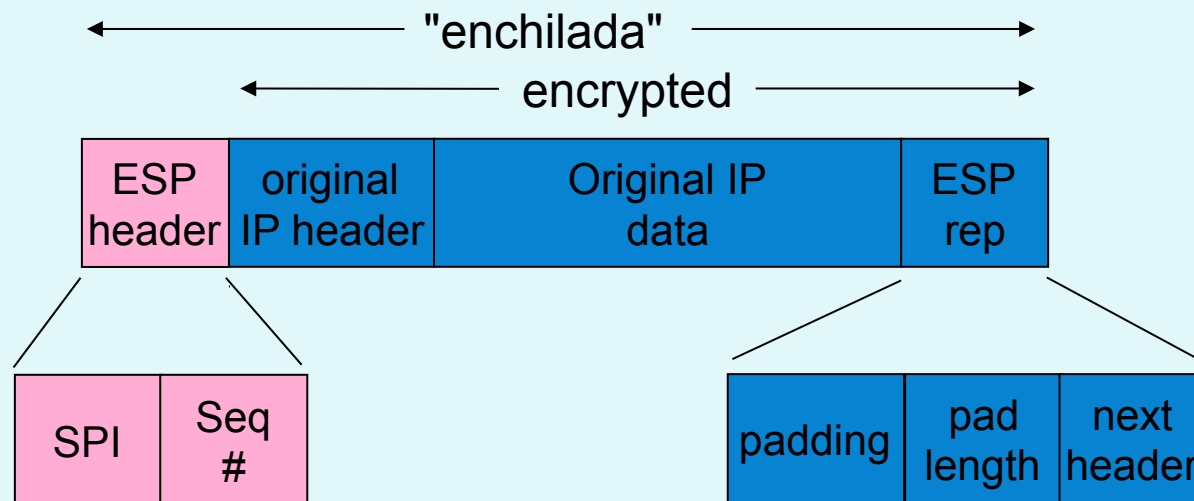
IPsec datagram: tunnel mode in ESP

- ESP auth field is added, which is the calculated hash value of the whole "enchilada". Algorithm and key set the SA.
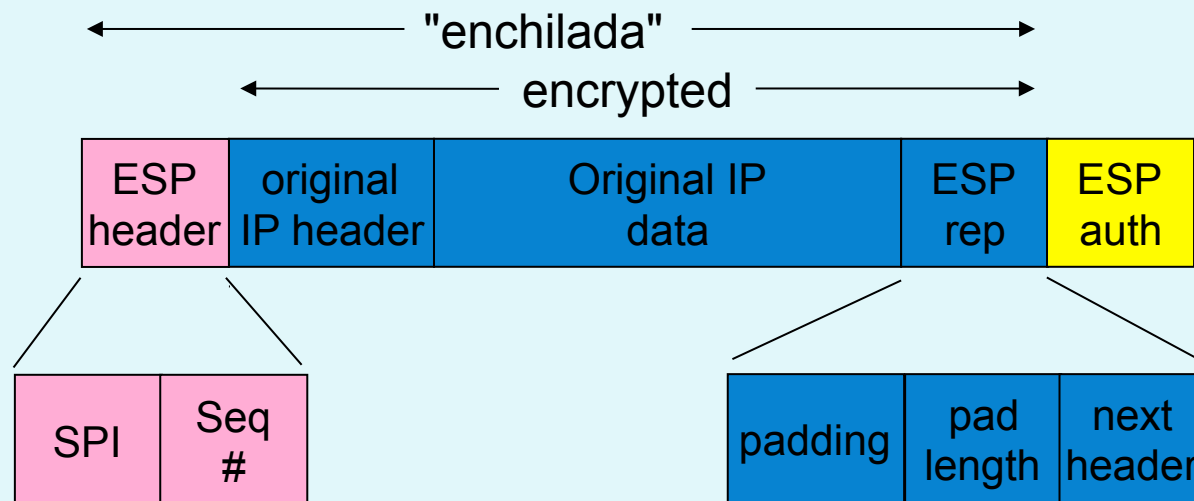
IPsec datagram: tunnel mode in ESP

- New IP header is built, which is added befor the data
- A new IP packet is created, which is sent normally over the network

IPsec datagram: tunnel mode in ESP

- What is in the new packet header?
  - protocol = 50 (means, that the data is ESP)
  - IPsec takes place between the source and destination IP nodes (routers R1 and R2)
- What does the receiver do(R2)?
  - from SPI in the packet header takes the data about SA, checks MAC enchilada, checks Seq#, decrypts enchilada, removes the fill, extracts the data, sends to the target computer

200.168.1.100

193.68.2.23

SA

R1

R2

172.16.1/24

172.16.2/24
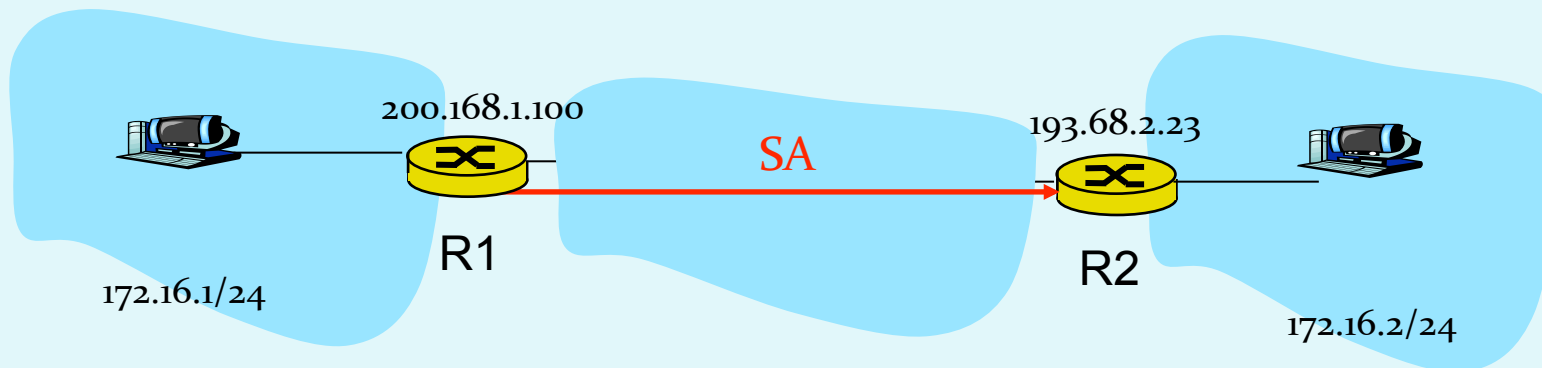
How to choose the datagrams for IPsec protection?

- This is defined by the Security Policy Database (SPD): it definesif the datagram should be protected based on the source IP, destination IP and type of protocol
- Defines which SA should be used

- SPD defines "WHAT" to do with the datagram
- SAD defines „HOW" to do it!

What level of protection does the Ipsec offer?

- Let`s say that Janez is our man-in-the-middle between R1 and R2. Janez doesn`t know the keys. What can he do?

  - Can he see the datagram content, source, destination, protocol, port?
  - Can he change bits in the packet?
  - Can he send in the name of R1?
  - Can he repeat the communication?

Protocol IKE

- IKE (*Internet Key Exchange*), protocol for key exchange over the internet
- With IPsec we need to establish the SA between clients, for example:

  Example of an established SA:

  SPI: 12345
  Source IP: 200.168.1.100
  Dest IP: 193.68.2.23
  Protocol: ESP
  Encryption algorithm: 3DES-cbc
  HMAC algorithm: MD5
  Encryption key: 0x7aeaca...
  HMAC key:0xc0291f...

- Specifying the SA by hand is impractical and time-consuming: it needs to be set for every way of communication and for every client pair!
- Solution: *IPsec IKE protocol*

IKE has 2 fases

- IKE uses PKI or PSK (pre-shared key) for client authenthication. It has two fases:
  - Fase 1: Establish a two-way IKE SA
    - IKE SA is a separated SA from IPsec SA, which is used only for key exchange (it is also called ISAKMP SA)
    - in IKE SA the key is established to protect further communications of key exchange(authenthication is performed with PSK, PKI or signature)
    - Two ways: Aggressive mode (shorter, but it reveals the identity of the client) and Main mode (longer, hide identity)
  - Fase 2: IKE generates keys for other services like Ipsec for example. Therefore IPsec SA is established:
   - Only way: Quick Mode

SSL

SSL: Secure Sockets Layer

- Widely used security protocol
  - supported in almost all browsers and on all servers (https)
  - Using SSL over 10 billion dollars of purchases are made annually
- Developed by Netscape in 1993
- Several types
  - TLS: transport layer security, RFC 2246
- Ensures <u>confidentiality, integrity, authentication</u>
- Developing objectives:
  - use in online transactions
  - concealment of information (especially credit card numbers)
  - web server authentication
  - client authentication
  - minimize the efforts in carrying out the purchase of other vendor

SSL and TCP/IP

- Accessible to all TCP applications over SSL API
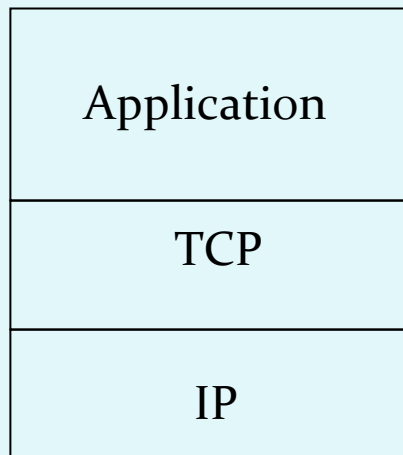
Application
TCP
IP

Common application

Application
SSL
TCP
IP
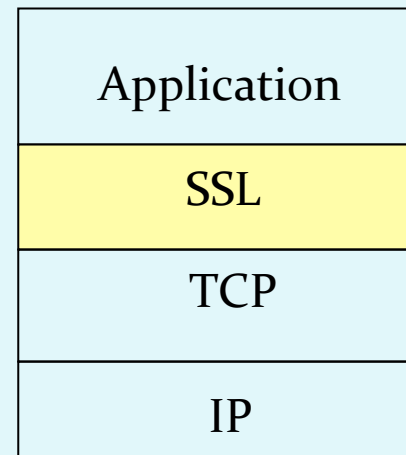
Applocation withSSL

SSL design

We could design it based on PKI encryption (encryption with the public key of the recipient, sender's private key, use of hash functions), but...

- We want to send streams of BYTES and interactive data, not static messages,
- For one link we want to have a MULTITUDE of keys, which changes,
- Despite that we want to use certificates (idea: we use theme at handshake)

Simplified SSL

Let's first look at a simplified idea of an SSL protocol. This has 4 phases:

- 1. HANDSHAKE: Ana and Brane use certificates to authenticate to one another and exchange keys

- 2. KEY DERIVATION: Ana and Brane use the exchanged key to make a multitude of keys

- 3. DATA TRANSMISSION: The data to be transferred is merged into RECORDS.

- 4. END OF TRANSMISSION: To ensure a safe end of transmission, special messages are sent

Simplified SSL: Handshake

hello

certificate

$K_B^+(MS) = EMS$
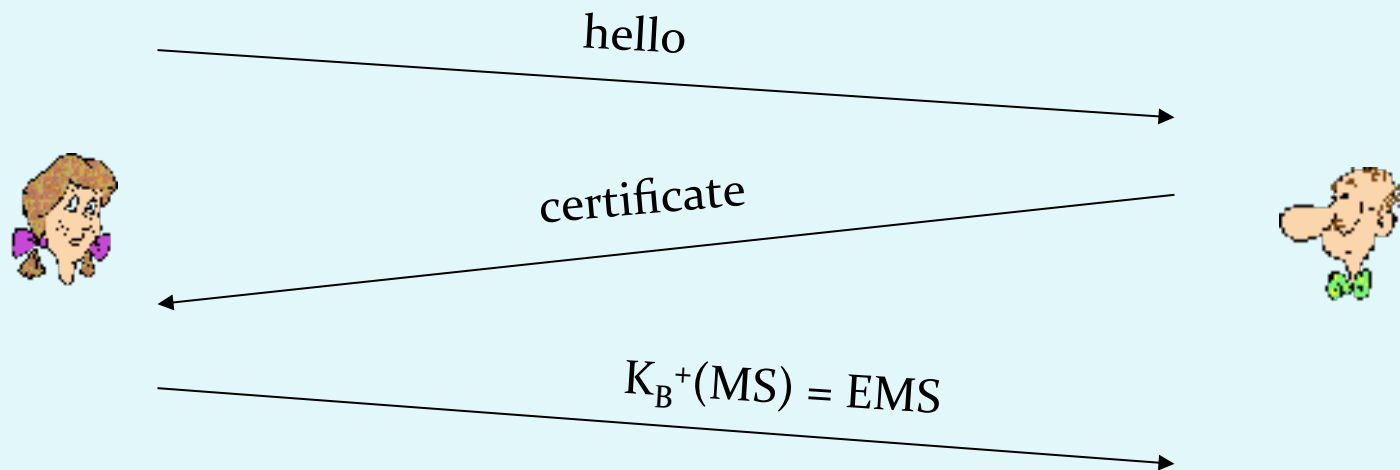
- MS = master secret
- EMS = encrypted master secret
- K_B^+ - public key of the receiver B

Simplified SSL: key derivation

- It is a bad practice to use the same key for several cryptographic operations, so : we use a special key to hide and a special key for integrity check(MAC)

- So we use 4 keys:

  - K_c = key to <u>hide</u> data sent from client to server

  - M_c = key for data <u>hashing</u>, sent from client to server

  - K_s = key to <u>hide</u> data sent from server to client

  - M_s = key for data <u>hashing</u>, sent from servert to client

- Keys are made using a special function. This uses the Master Secret and additional (random) data to generate the other keys

Simplified SSL: Data sending
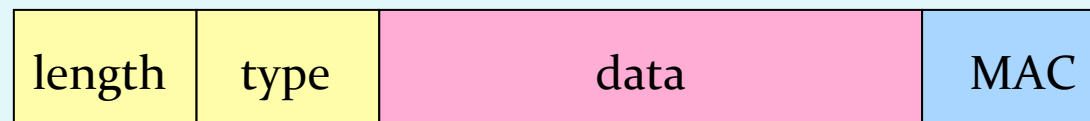
- How to chech for data integrity?
  - If we send in bytes, where do we attach the MAC (hash value of the message)?
  - Even if we send the mac MAC at the end of the transmissin (all bytes), we do not have the mid-term integrity tests!

- SOLUTION: Break the data stream in RECORDS
  - We attach MAC to every record
  - The receiver can act to integritete (in)validity of any record

Simplified SSL: Data sending

- Problem 1: packet number is unencrypted in the TCP packet header. What can an attacker do?
  - Can the attacker intercept and repeat the communication?
  - Can he change the packet numbers?
  - Can he intercept and remove the packet?

- SOLUTION: account feo the packet number when calculationg MAC
  - MAC = MAC(key M_x, serial_number || data)
  - we do not have separate packet number
  - protection against recurrence of communication : a one-time use token

Simplified SSL: Data sending

- Problem 2: attacker prematurely terminates session
  - One or both parties may feel that the data is missing.

- SOLUTION: introducing a special "type of record", which has a particular value in case of the final message
  - example: 0 means data, 1 means end
  - We use this value when calculation MAC
    MAC = MAC(key M_x, serial_number||type||data)

length	type	data	MAC

Simplified SSL: Example

hello →

← certificate, token

$K_B^+(MS) = EMS$ →

type 0, seq 1, data →

type 0, seq 2, data →

← type 0, seq 1, data

hidden

type 0, seq 3, data →

type 1, seq 4, close →

← type 1, seq 2, close
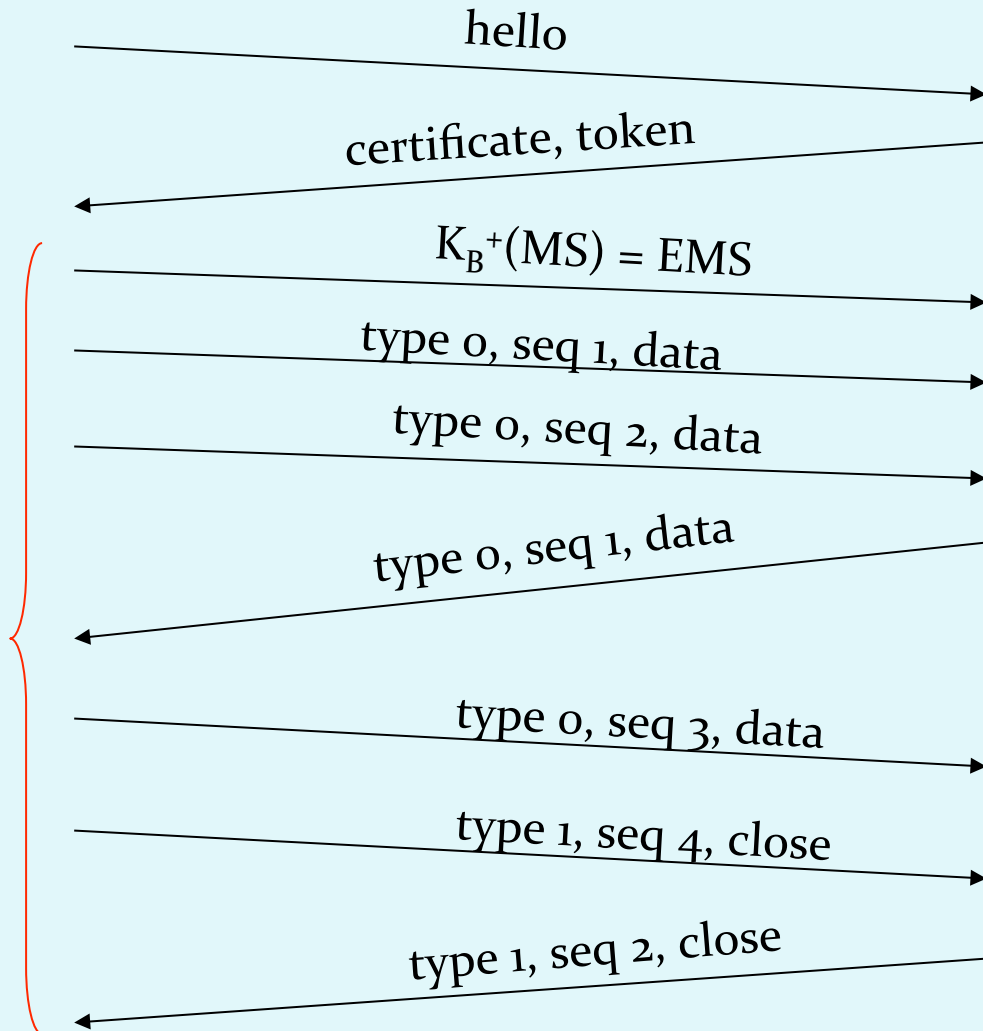
Real SSL: details

- What are the lenghts of the protocol fields?
- Which protocol should be used for hiding? Agreement for using the protocol:
  - We want to allow the client and server to choose about cryptographic algorithms(*negotiation*, client offers, server choose)
  - Most common simetric algorithms
    - DES – Data Encryption Standard: block
    - 3DES – Triple strength: block
    - RC2 – Rivest Cipher 2: block
    - RC4 – Rivest Cipher 4: stream
  - Most common algorithms for PKI criptography
    - RSA

Real SSL: Handshake

- Simplified SSL: hello->, <-certificat, encrypted MS->

- Real SSL actually do: server authentication, algorithm selection, key determination, client authentication (optional)

- Process:

1. • Client send a list of supported algorithms + token

2. • Server chooses the algorithm from the list, return the selection, certificate and his tokn

3. • Client checks the certificate, generate PMS, and with server`s public key it cryptates it, then it sends it back to the server

4. • Client and server independently calculate encryption and MAC leys from PMS and tokens.

5. • Odjemalec pošlje MAC od vseh sporočil v rokovanju.

6. • Server sends MAC of all messages in the handshake.

Pravi SSL: Rokovanje

1. ## Why MAC exchange in steps 5 and 6?

   - Client usually offer mor than one algorithm, some of them are weaker, other are stronger. An attacker could delete from the offer the stronger ones.
   - The last two messages ensure the integrity of all the other messages that have been sent so they prevent an attack like that

2. ## Why the use of tokens?

   - Let`s say, that Zelda is listening to the messages between Ana and Brane and saving them. The next day Zelda sends to Brane exactly the same messages Ana sent to him the day before:
     - If Brane has a shop, he will think that Ana is buying again
     - Brane is using a different token for every communication, so Zelda can`t replicate the same conversation
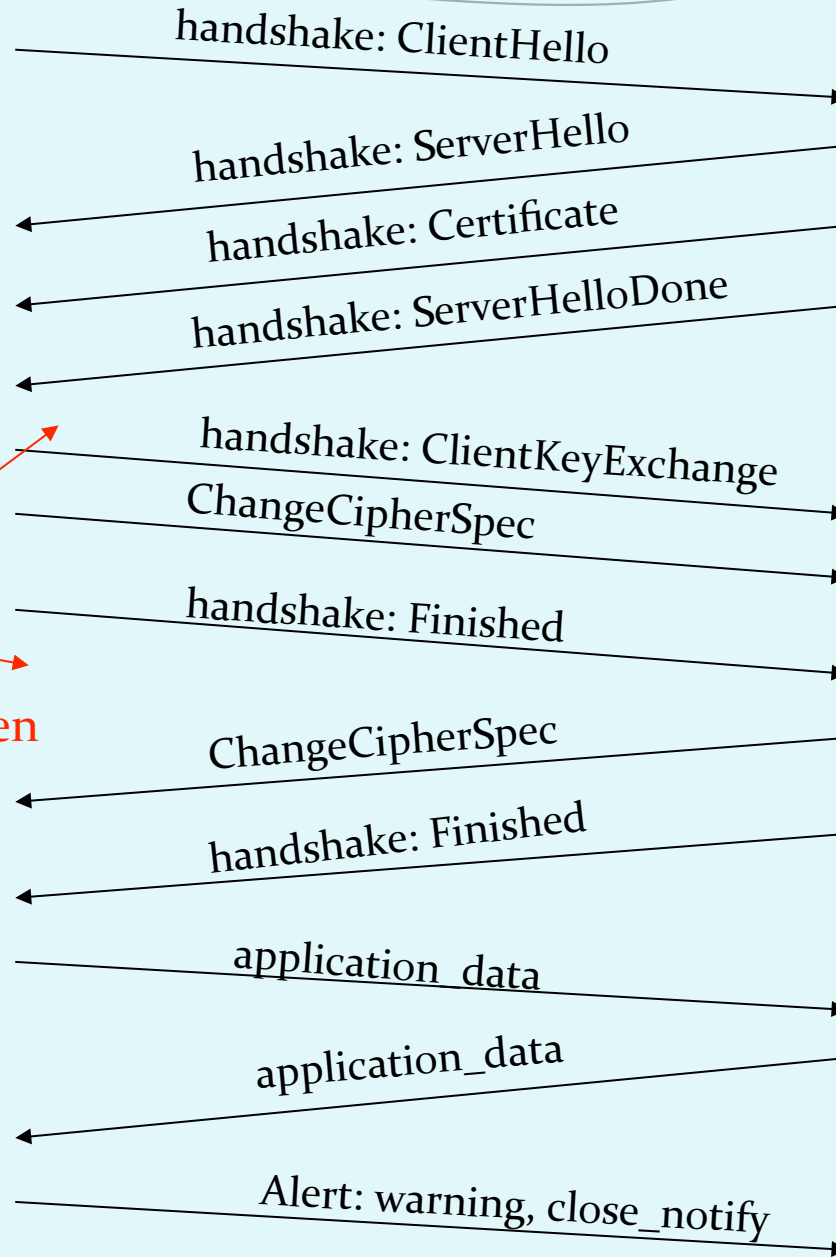
SSL: conversion to records
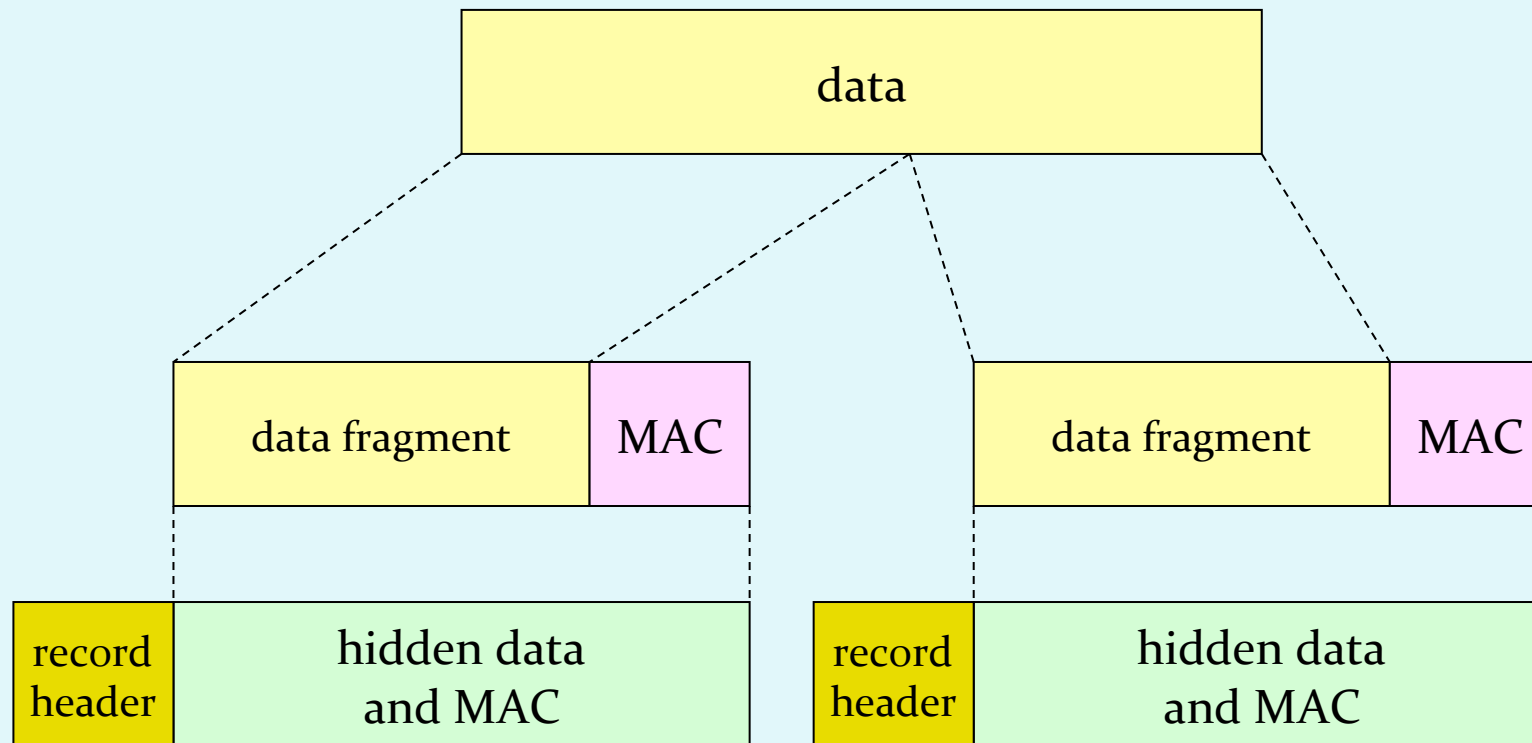
- RECORD header: type of content(1B); SSL version (2B); length (3B)
- MAC: serial_number; MAC key M_x
- FRAGMENT: max length is 2^{14} bytes (~16 Kbytes)

Example of a real handshake

handshake: ClientHello →

handshake: ServerHello ←

handshake: Certificate ←

handshake: ServerHelloDone ←

handshake: ClientKeyExchange →

ChangeCipherSpec →

handshake: Finished →

From here on everything is hidden

ChangeCipherSpec ←

handshake: Finished ←

application_data →

application_data ←

Alert: warning, close_notify →

SSL: key derivation

- Client and server token and the PMS is used in the function, which calculates the pseudo-random numbers. We get MS *(master secret).*

- MS and new tokens areused in a second random generator, we get a BLOCK. BLOCK is cut in 6 pieces, so we get:
  - MAC client key
  - MAC server key
  - Client encryption key
  - Server encryption key
  - Client initialisation vector (IV)
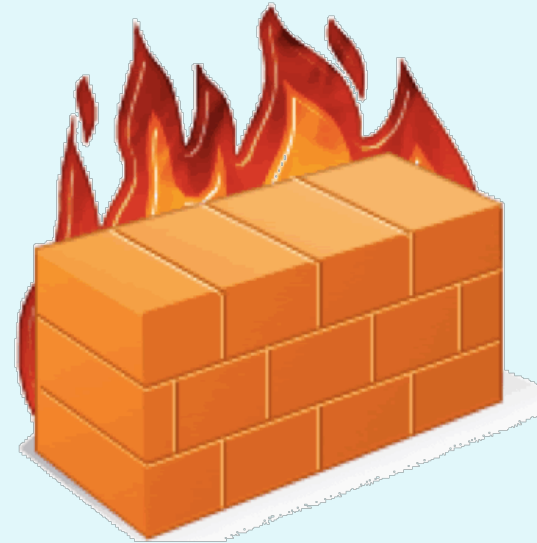 - Server initialisation vector (IV)

Like with the simplified SSL!

WHAT IS THIS?

They are needed, when we use a symmetric algorithm z block cypher criptography(3DES or AES), which needs initialisation!
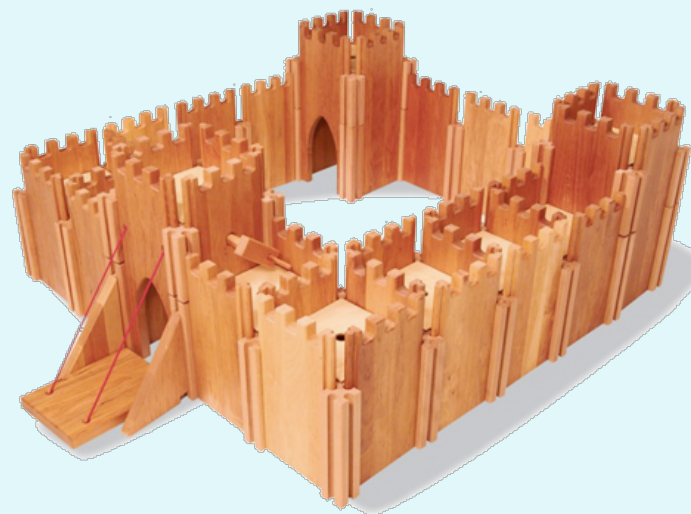
Network security

- An administrator can divide users into:
  - Good guys: users who legitimately use network resources, belong to the organization
  - Bad guys: everyone else, their access must be closely monitored
- The network has normally only one access point, there we control the accesses :
  - firewall
  - IDS, intrusion detection system
  - IPS, intrusion prevention system

Požarni zid

An isolated network allow some packets to pass, others it blocks. It has
3 tasks:
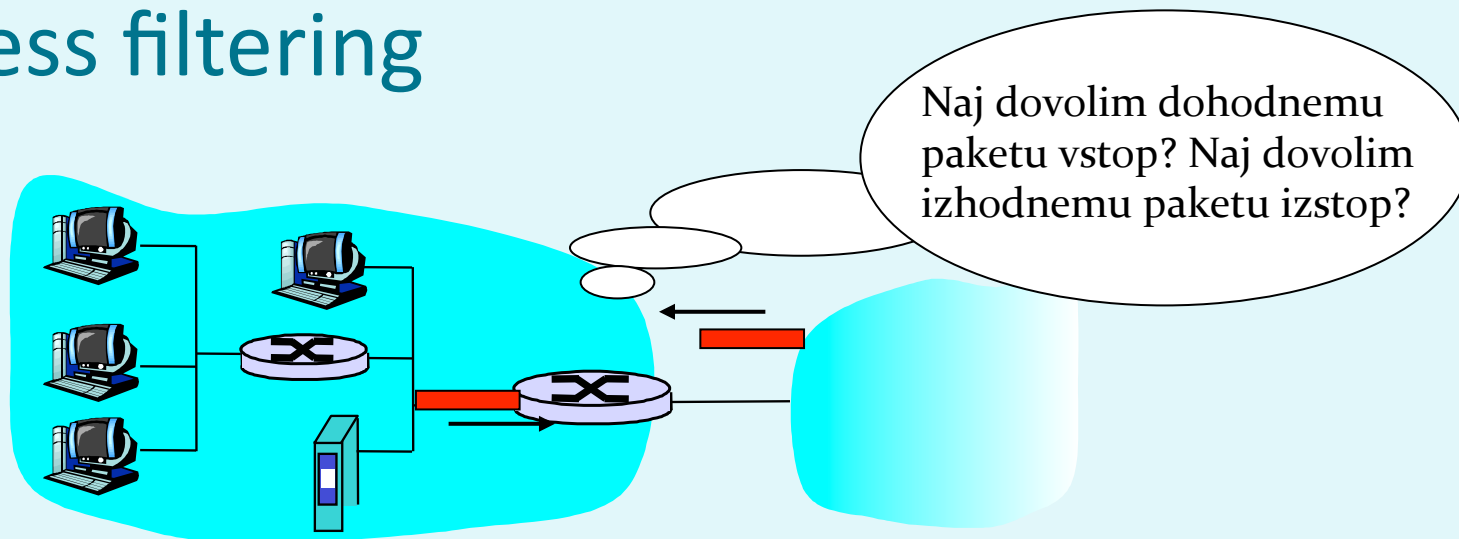- Filter ALL traffic,
- leaves only traffic that is ADMISSABLE according to policy,
- Is IMMUNE to attacks

internal
network

public
network
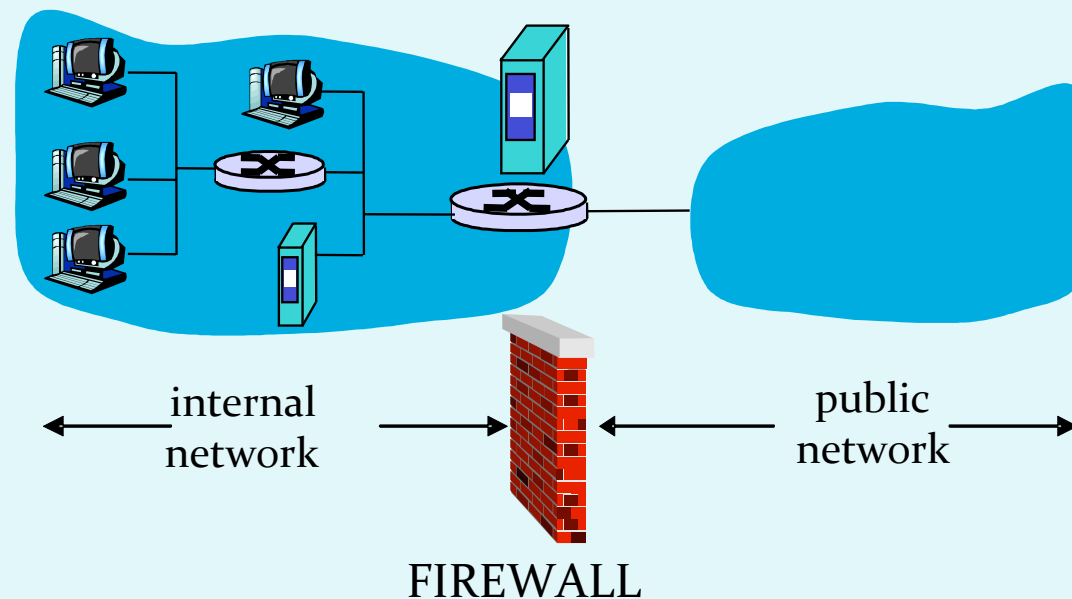
FIREWALL

Firewal: filtering options

1. *stateless, traditional*
2. *stateful filter*
3. *application gateways*

Stateless filtering

- Usually it`s allready done by the router, which is adjacent to a public network. Based on the contents of the packets, it decides whether to pass any **single package**. Decision is based on:
  - Source/destination IP
  - IP protocol number: TCP, UDP, ICMP, OSPF etc.
  - TCP/UDP source and destination ports
  - Type of ICMP
  - TCP SYN (connection establishment!) and ACK bits (ACK=1 stands for the first segment when connecting)

Stateless filtering: examples

- Example 1: block ingoing datagrams with IP protocol 17 (UDP) and source or destination port 23 (telnet)
  - result: we filter all ingoing and outgoing UDP connections and telnet connections.

- Example 2: Blokiraj ingoing TCP segments with flag ACK=0.
  - result: block external clients from connecting with internal clients and allow in the opposite direction (outward)

Stateless filtering: example

We want to achieve:	Firewall settings
Deny access to any external web server.	Reject all packets with any IP address on port 80
Deny all TCP connections except the ones which are intended for the public web server on 130.207.244.203.	Reject all incomming TCP SYN packets, except the ones with the IP 130.207.244.203, port 80
Prevent Smurf DoS attack (using broadcast to overload the service).	Reject all ICMP pakete with a broadcast network adress(eg. 130.207.255.255).
Deny network analysis with traceroute	Reject all outgoing ICMP packets with the message "TTL expired"

Stateless filtering: access lists

- ACL, access control list
- Table of rules
- Records in pairs: (condition, action)
- Example: deny all traffic except outgoing WWW and DNS in both ways

Source adress	Destination adress	Protocol	Source port	Destination port	flag	action
222.22/16	From outside 222.22/16	TCP	> 1023	80	any	allow
From outside 222.22/16	222.22/16	TCP	80	> 1023	ACK	allow
222.22/16	From outside 222.22/16	UDP	> 1023	53	---	allow
From outside 222.22/16	222.22/16	UDP	53	> 1023	----	allow
all	all	all	all	all	all	deny

Statefull filtering

- It takes into account the connection and its current state
  - Isolated filtering can allow to pass pointless packets (e.g.. port = 80, ACK =1; although internal client has not established a connection) :

- IMPROVEMENT: **Stateful packet filtering** monitor and keep a record of the status of each TCP connection established
  - record the start of a connection (SYN) and it's end (FIN): based on this it determines if the package makes sense
  - after a certain time treat the connection as invalid (timeout)
  - Use a similar access list that determines when it is necessary to control the validity of links (check connection)

Context packet filtering

Source adress	Destination adress	protocol	Source port	Destination port	flag	action	Check connection
222.22/16	From outside 222.22/16	TCP	> 1023	80	any	allow	
From outside 222.22/16	222.22/16	TCP	80	> 1023	ACK	allow	X
222.22/16	From outside 222.22/16	UDP	> 1023	53	---	allow	
From outside 222.22/16	222.22/16	UDP	53	> 1023	----	allow	X
all	all	all	all	all	all	deny	

Application gateways

- allow further filtering by selecting users that can use a particular service
- Allow filtering based on data on the application layer rather only on fields IP/TCP/UDP.

Client establish a telnet connection with the gateway

Gateway establish The remote connection

app. gateway

router and filter
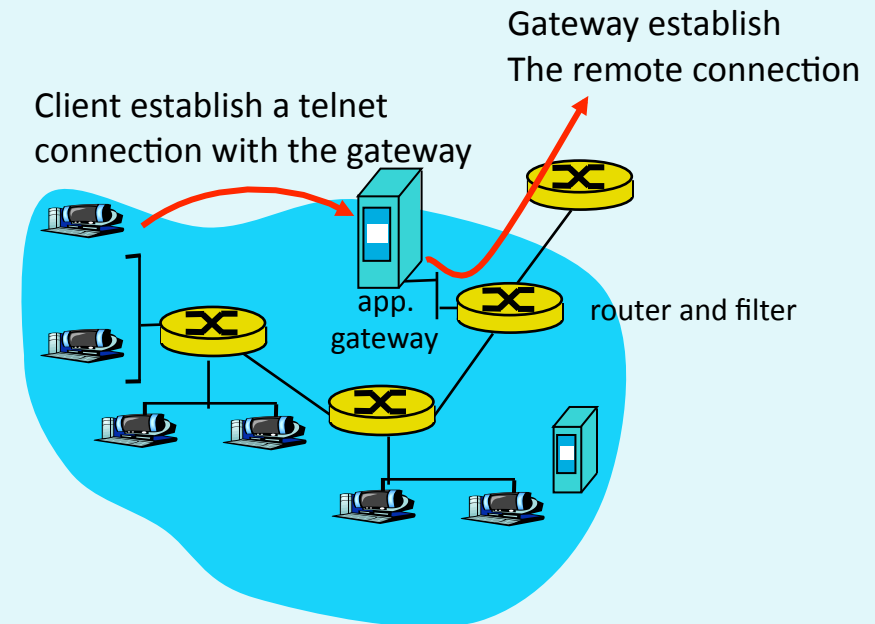
1. All clients establish a connection over the gateway,
2. The gateway establish the remote connection with the destination server only for authorised clients. The gateway forwaeds data between 2 connections,
3. Router block all telnet connections except the ones that originate from the gateway

Application gateways

Even application gateways have limitations:

- If users need more applications(telnet, HTTP, FTP etc.), every application needs its own application gateway,

- Clients need to be configured in order to be able to connect with the gateway (e.g.. IP address of the browser server)
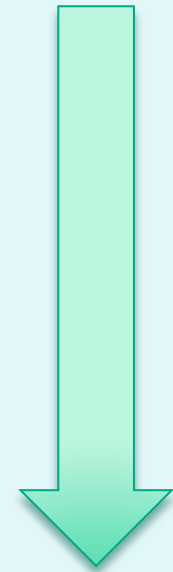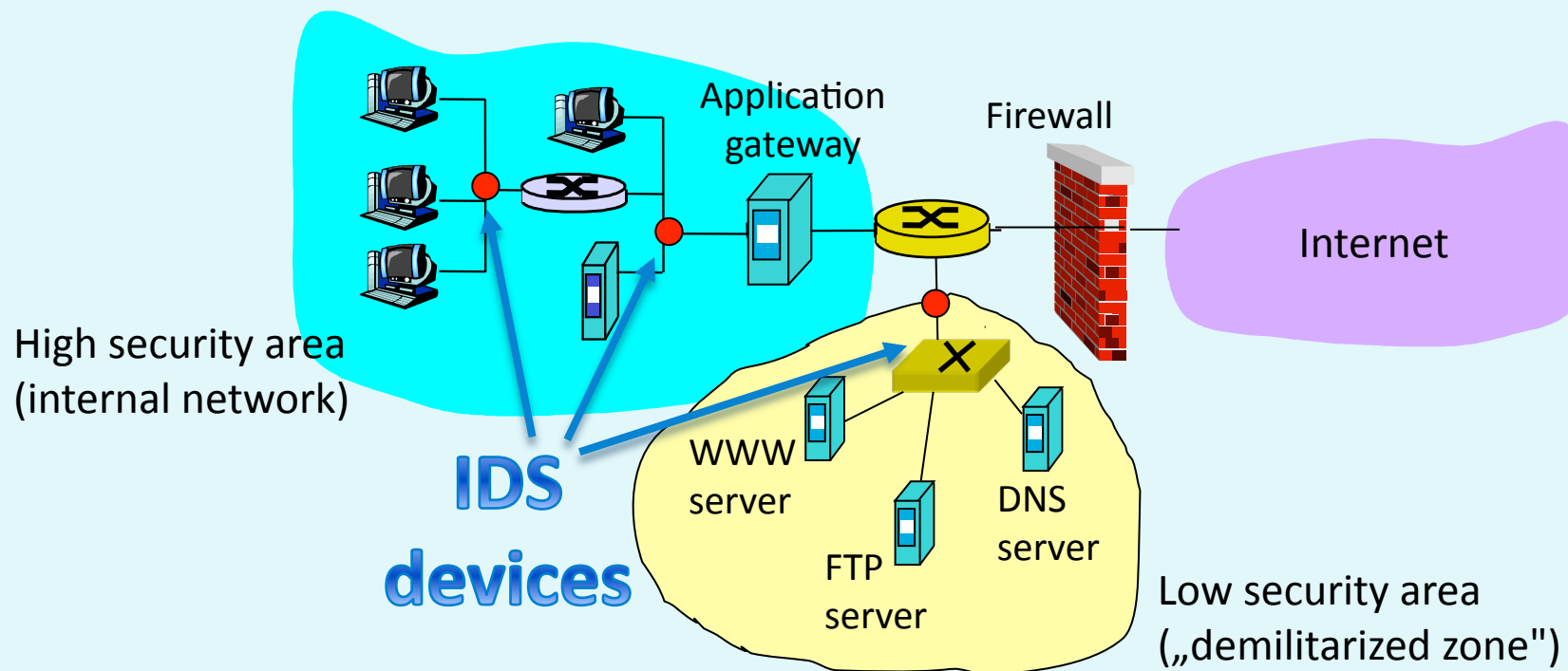
Intrusion detection system

- Firewall as a packet filter filters only based on IP, TCP, UCP and ICMP heads, which does not provide detection for all attacks – for this, the data in the packet also needs to be checked
  - Attack examples: port scan, TCP stack scan, DoS attack, worms, viruses, attacks on the OS, attacks on applications
- Additional device - IDS, which does **in-depth package analysis**. For suspicious packages entering the network, the device can prevent their entry or send warning messages.
  - Intrusion detection system(IDS) sends a message about potentially malicious traffic
  - Intrusion prevention system(IPS) filters suspicious traffic
  - Cisco, CheckPoint, Snort IDS

Intrusion detection system

- We can have more IDS/IPS devices in a network (useful for comparing complex content packages with stored patterns)

Application gateway

Firewall

Internet

High security area (internal network)

IDS devices

WWW server

FTP server

DNS server

Low security area („demilitarized zone")

Methods of intrusion detection

How IDS/IPS works?

- comparison with stored samples of attacks(*signatures*)
- observation of atypical traffic (*anomaly-based*)

Detection with signatures

- Signatures can store source IP, destination IP, protocol, sequence of bits in a data packet, can be linked to a series of packets

- Safety therefore depends on the database of known samples; IDS/IPS poorly detect yet unseen attacks

- Possible false alarms

- Demanding processing(may overlook the attack)

Anomaly-based intrusion detection

- The system observes the normal traffic and calculates statistics related to it

- It reacts to statistically unusual traffic neobičajen promet (e.g.. sudden large number of ICMP packets)

- Can detect yet unseen attacks

- Hard to distinguish between normal and unusual traffic

Example of an IDS/IPS system

- Snort IDS
  - public-domain, open source IDS for Linux, UNIX, Windows (for network reading it uses the same library as Wireshark)
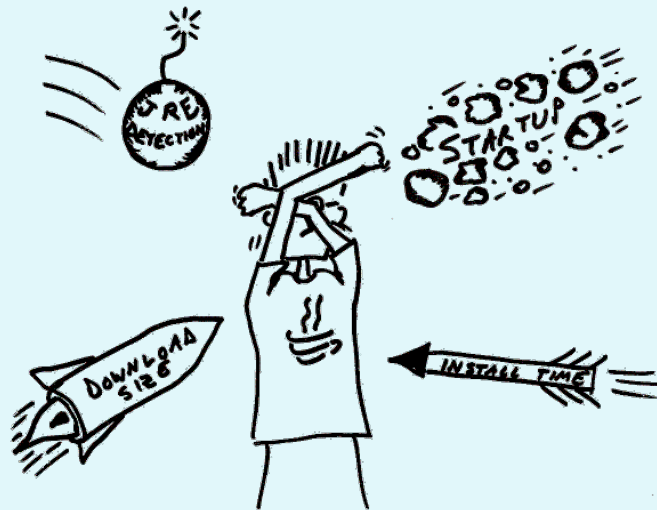 - Example of an attack signature
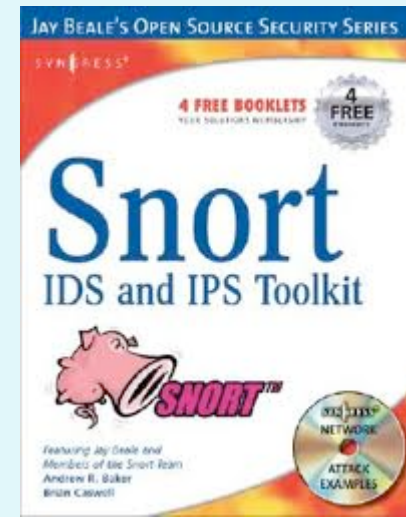
```
alert icmp $EXTERNAL NET any -> $HOME NET any
(msg:"ICMP PING NMAP"; dsize: 0; itype: 8;)
```

Message for administrator

Empty packet(length 0) and ICMP type 8 (=PING) are properties of an NMAP attack
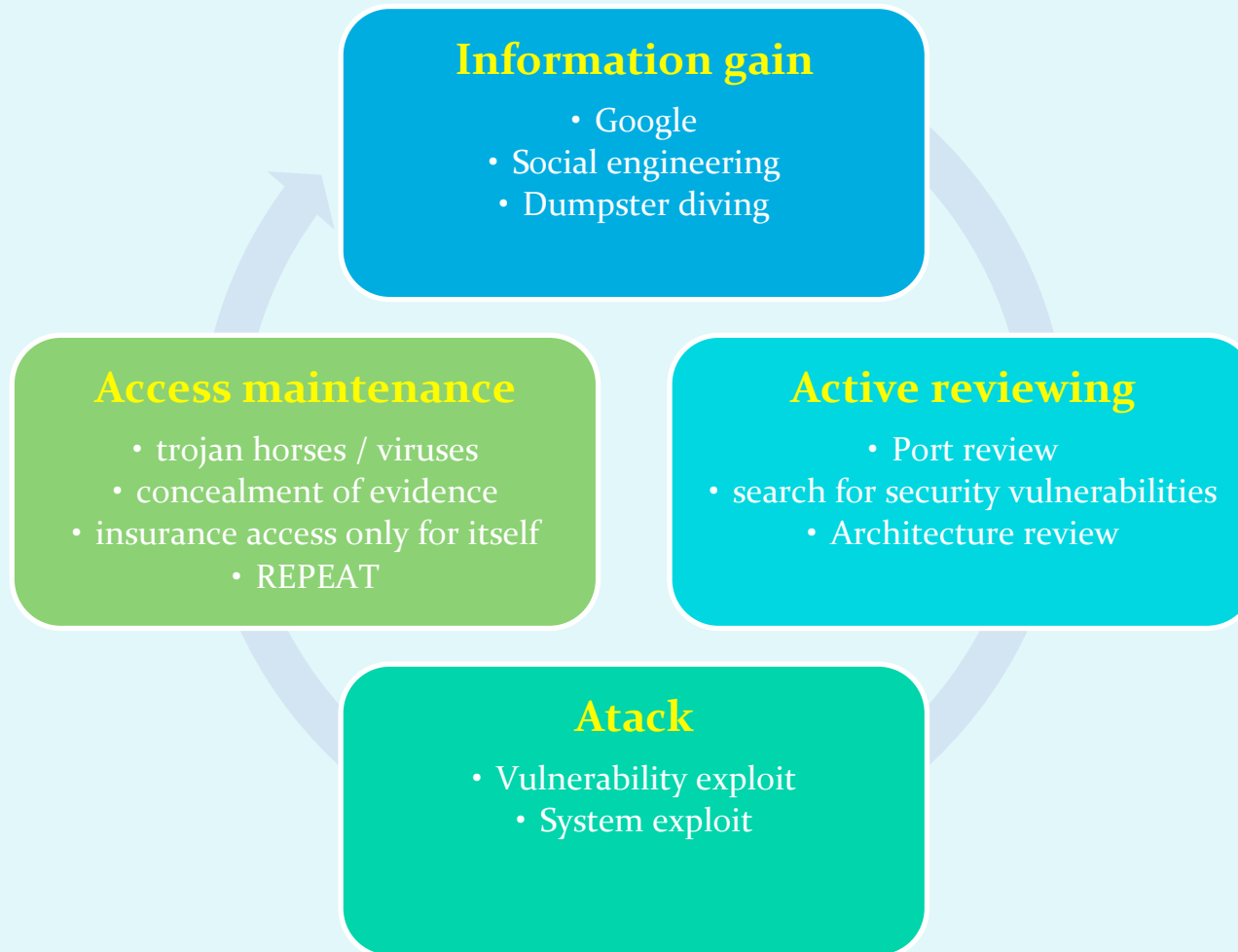
React to ALL INCOMING ICMP traffic

Attacks and threats

Frequent attacks on network systems

- **PURPOSE?** They are designed to harm or bypass computer and network functions.

- **WHY?** Finantial benefits, harmness, misappropriation, economic benefits.

- **HOW?** Threats to confidentiality, integrity and availability of network systems
  - attacks by changing the information (*modification attack*)
  - denial of communication (*repudiation attack*)
  - System failure (*denial-of-service attack*)
  - unauthorized access(*access attack*)

Frequent attacks on network systems

Information gain

- Google
- Social engineering
- Dumpster diving

Active reviewing

- Port review
- search for security vulnerabilities
- Architecture review

Atack

- Vulnerability exploit
- System exploit

Access maintenance

- trojan horses / viruses
- concealment of evidence
- insurance access only for itself
- REPEAT

Common attacks

- Reconnaissance: the attacker try with a variety of techniques to identify the system architecture, services, etc.
  - It helps to prepare the attack on the system
  - example (*war-dialing*): attacker by calling random phone numbers try to identify the number the modem uses to connect to the network

Common attacks

- Eavesdropping: intercept network traffic, present especially in wireless networks (attacker obtains passwords, credit card numbers, …)
  - Passive attacker
  - Activ attacker
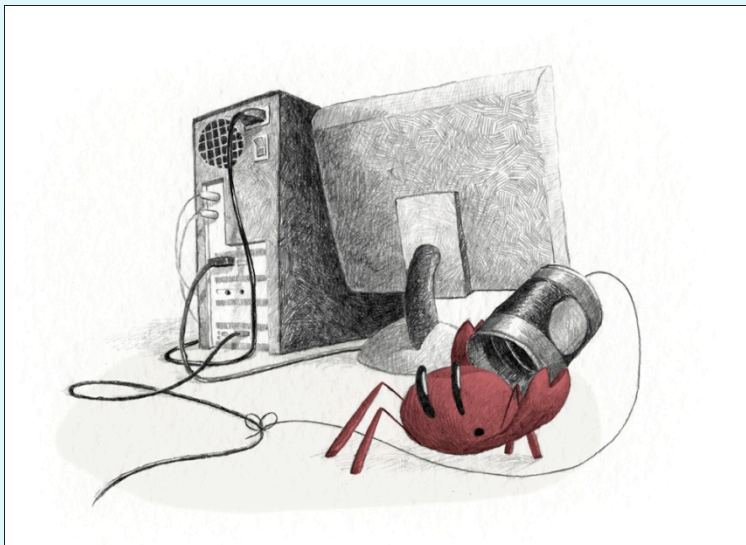
Common attacks

1. **Weak keys**
2. **mathematical attacks** on cryptographic algorithms and keys
3. **Password guessing** (*brute force*, the dictionary attack)
4. **viruses, worms, tojan horses**
5. **exploit weaknesses in the software**
6. **Social engineering** (over e-maila, telephone, services)

SOCIAL ENGINEERING SPECIALIST
Because there is no patch for human stupidity

THE NEW PAPER/Wednesday, October 6, 2004   SINGA

Hacker's polite approach works like a charm

BY ARUL JOHN
tsp@sph.com.sg

reveal computer system passwords
during happy hours.
Last year, he said computer fraud.

How do you defend on the risks above?

Common attacks

7. **port scan**: intruder test, which servers are functioning (e.g. ping) and what services they offer. An attacker can acquire information about the system: DNS, services, operating systems)

8. **Dumpster diving**: a method by which attackers can access information about the system (instructions, lists of passwords, phone numbers, work organization)

9. **Mathematical attacks** on the cryptographic algorithms and keys (brute force)

10. **Birthday attack**: is an attack on hash functions, which require that two messages will not generate the same compressed value. For weaker functions an attacker is looking for a message that will give the same hash value.

Common attacks

11. **Back door**: the attacker bypass security checks and access the system via another way

12. **IP spoofing**: the attacker tricks the target system to be someone else (someone known) by changing packets,

13. **Man-in-the-middle**: the attacker intercepts communication and behaves as if he is the target system (when using certificates the victim may use the public key of the attacker)
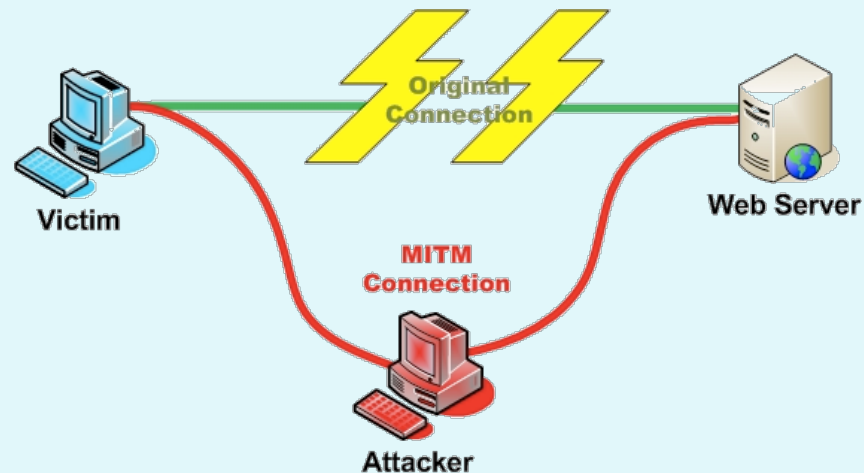
Common attacks

14. **Replay**: the attacker intercepts and saves old messages and send them back after some time, posing as one of the participants
    - How do we prevent replay attacks?

15. **TCP hijacking**: the attacker interrupts communication between the users and insert himself in place of one of them, the other believes that he is still communicating with the first
    - What can the attacker gain with this?

16. **Fragmentation attack**: packets are divided into fragments. The header is divided into different fragments in a way that the firewall can not filter
    - tiny fragment attack: divide the header of the first packet
    - overlapping fragment attack: a wrong offset overwrites previous packets
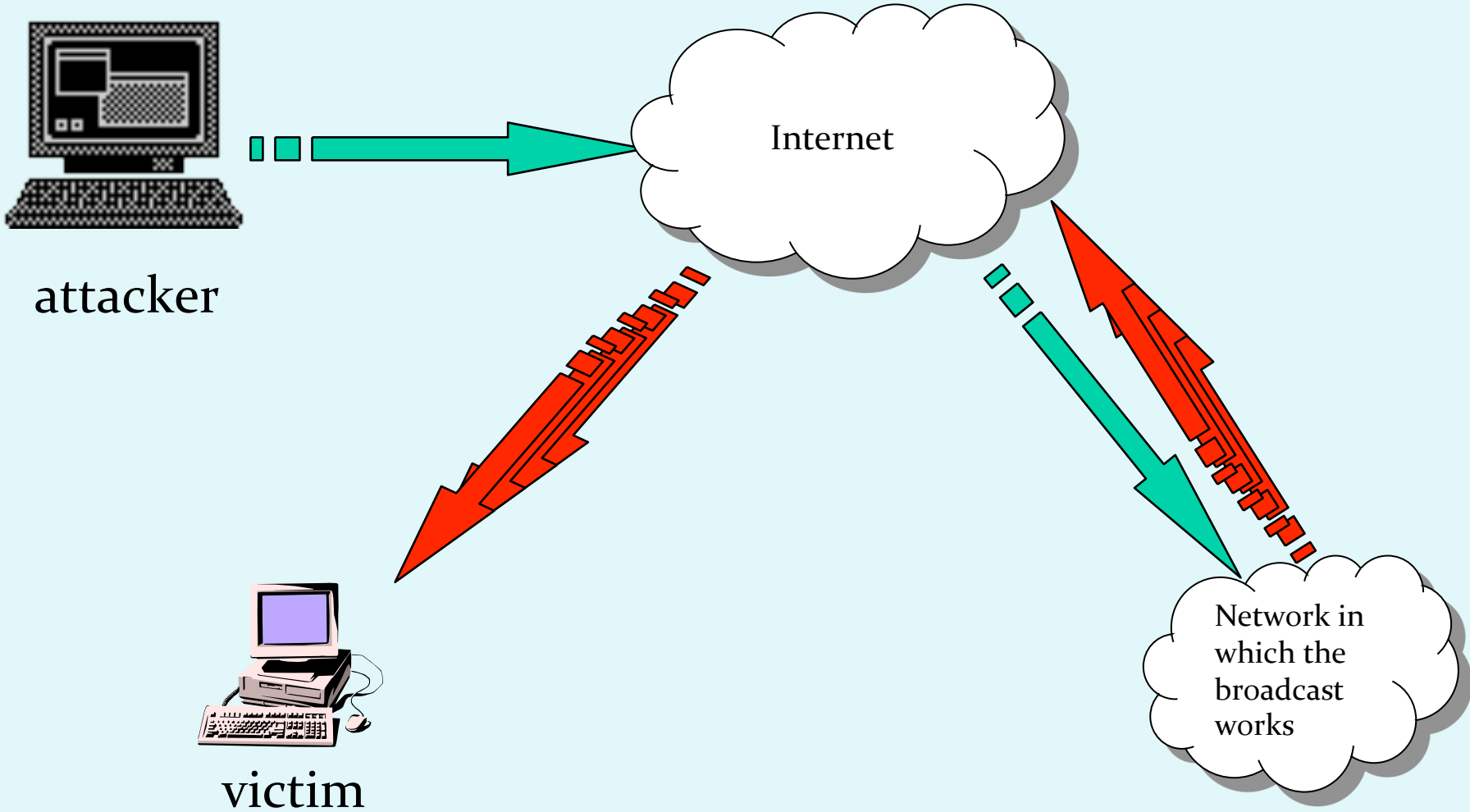
Common attacks - DoS (1/5)

17. Denial-of-Service

- The aim of the attacker: overload network resources so they stop responding to the requirements of regular users (e.g.. setting up a large number of connections, consume storage capacity, …)

- DDoS (*distributed*): DoS attack, caused by an attacker using multiple network systems at once

- users of distributed network systems may not know that the equipment that is attacking is installed where they are
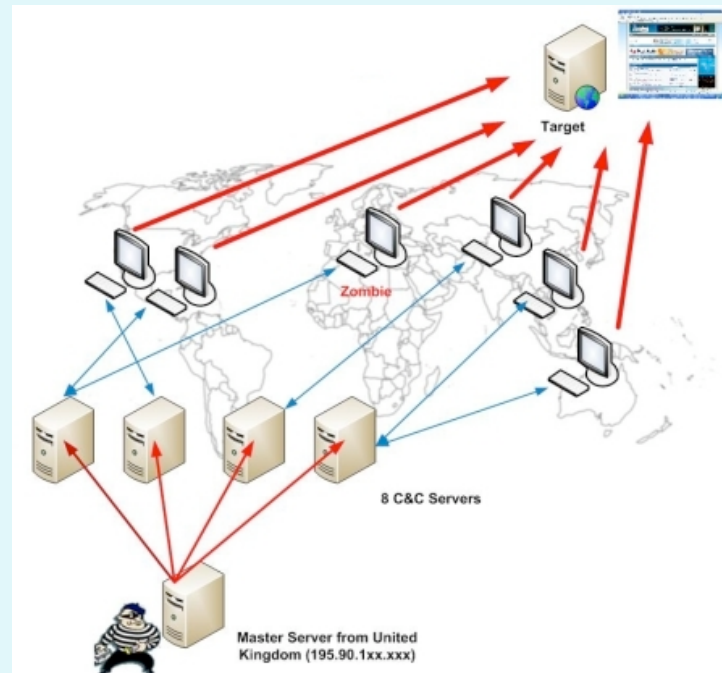
Common attacks - DoS (2/5)

- Examples:
  - **Buffer overflow**: the attacker sends more data to a process than it can take(Ping of death: ICMP with more than 65K of data has caused a system crash)
  - **SYN attack**: the attacker sends a large number of connection requests and then he ignores the system response so the system connection queue gets overloaded
    - solution: limit the number of open connections, timeout
  - **Teardrop attack**: the attacker changes the number and length of the fragments in the IP packet. In that way the recipient gets confused
  - **Smurf attack**(on the following slide): using indirect broadcast to overload the system

Common attacks - DoS Smurf attack(3/5)

attacker

Internet

victim

Network in which the broadcast works

Common attacks - DoS (4/5)

- The use of bots(*web roBOT*) for organizing attacks against the target system
  - Bots can be computers, infected with trojan horses
  - Their owners may not know that they are attacking the target system

Common attacks - DoS (5/5)

- subjects in the attack: **the attacker**, the central computer to **control the bots** (Herder), **bots** (zombie), **the goal**
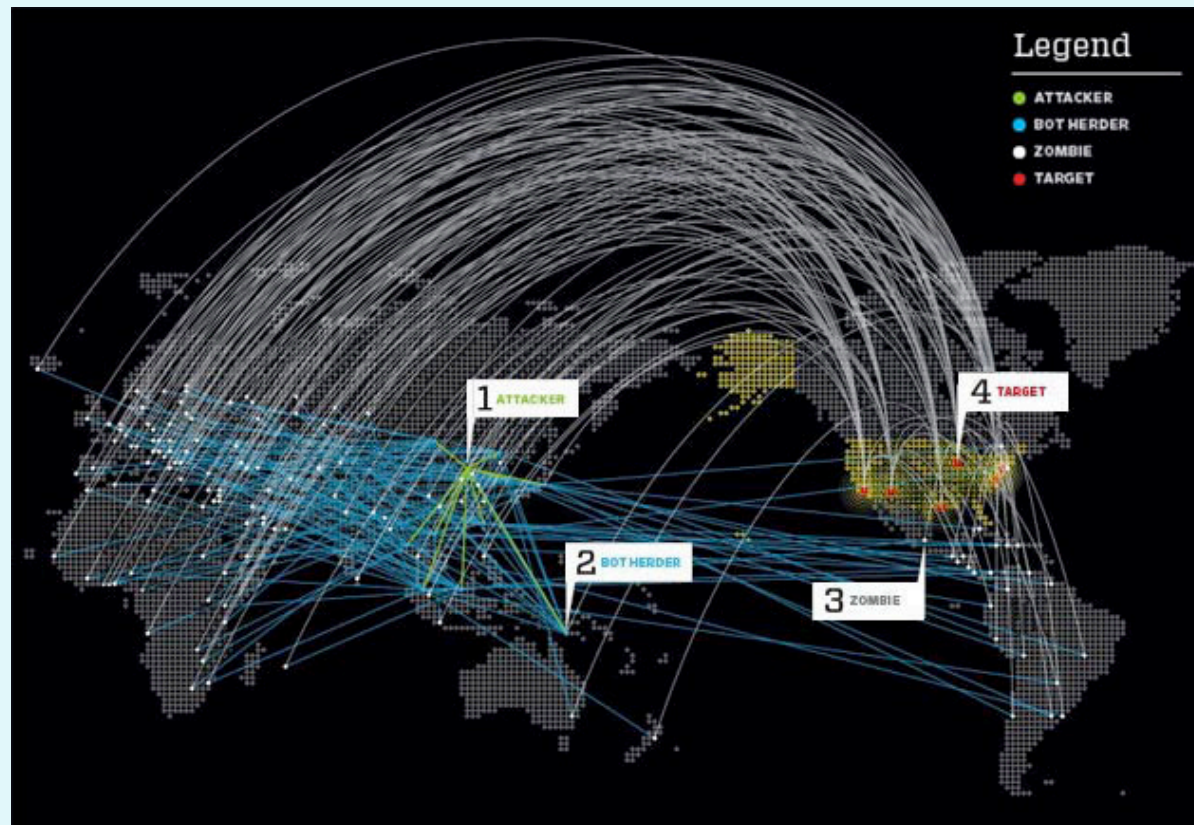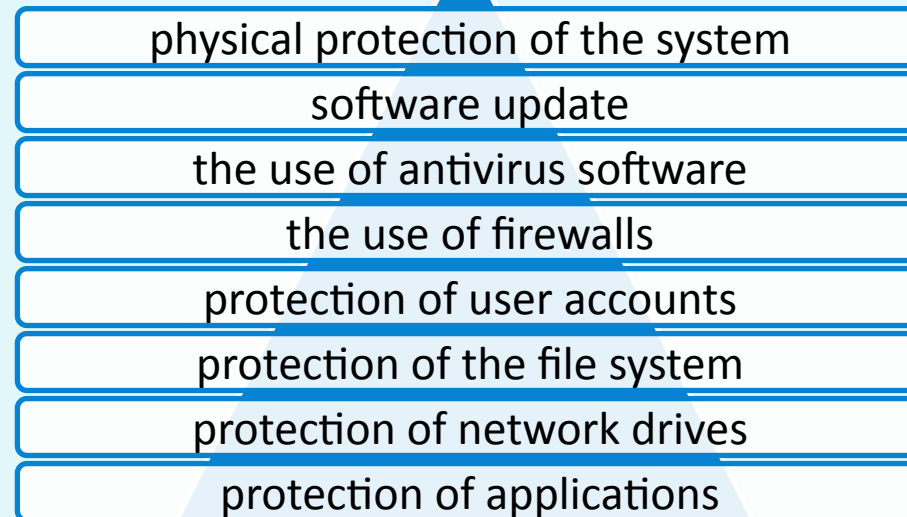
Defense against attacks

Defense techniques

- The network needs only one weak link - the weakest user to compromise the network. The administrator must prevent the transfer of harmful programs on the user's workstations and close security holes in the infrastructure (configuration):

physical protection of the system

software update

the use of antivirus software

the use of firewalls

protection of user accounts

protection of the file system

protection of network drives
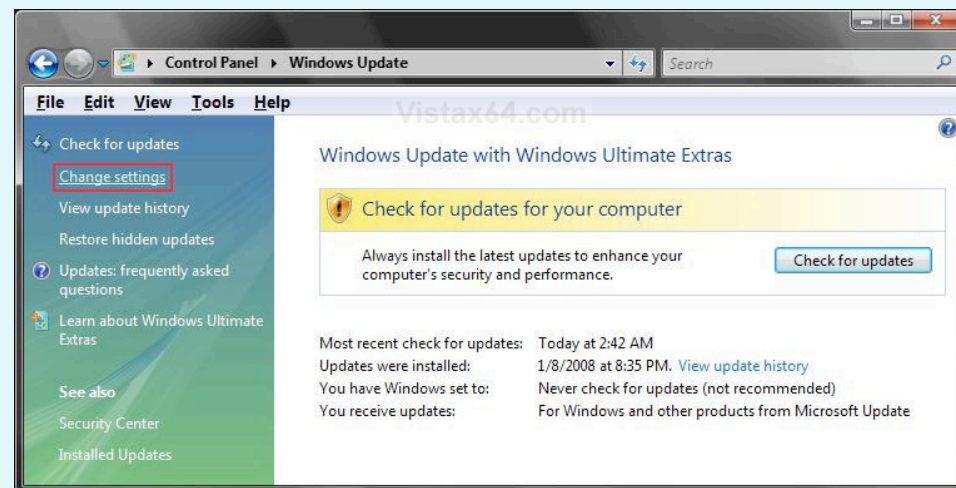
protection of applications

Physical protection of the system

- Restrict physical access to servers and computers
  - Computer locking
  - Boot password(CMOS/BIOS)
  - Password for accessing the BIOS(security, boot, etc.)
  - Disable boot from floppy or cd
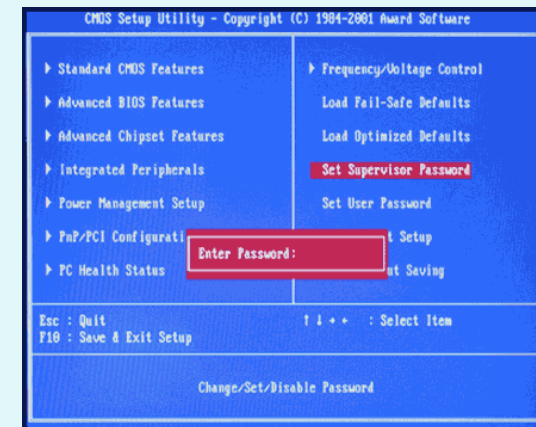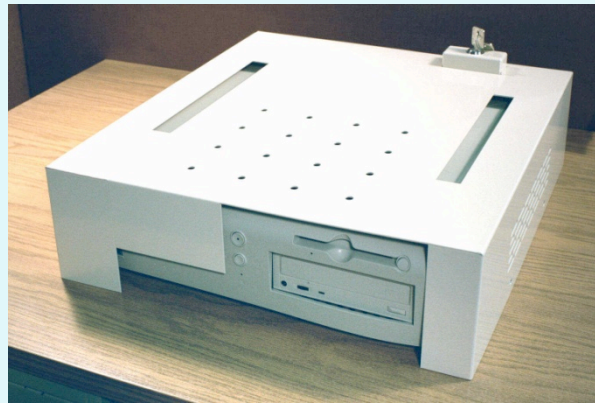
Software update

- Updating the software(*patching*), by which the developer enables us to repair security holes
  - The administrator needs a plan for test, introduction and installation of patches

Use of AV / firewall

- The use of antivirus software
  - Multiple options: installation on the client / server, automatic updates, real-time protection.
    - Recommended: install on the client, because malicious software begins to operate there. AV on application gateways tend to look for a subset of protocols on that location
  - update
    (individual or centralized)

- The use of firewall
  - On a network / personal firewalls

User accounts protection

- Attackers are looking fo unused, inactive, unprotected accounts to access the system:
    - Rename the administrator user name(superuser, root, administrator),
    - limit the number of accounts with high privileges (separate admin accounts, frequent changes of passwords),
    - disable the use of old accounts,
    - use complex passwords

Protection of file/network system

- Protect the file system
  - Assign the minimum rights required to users to acces the file system
  - uninstall unnecessary applications
  - Protect areas with boot management. Example - Windows:

1. c:\autoexec.bat
2. c:\config.sys
3. windir\wininit.ini - Usually used by setup programs to have a file run once and then get deleted.
4. windir\winstart.bat
5. windir\win.ini - [windows] "load"
6. windir\win.ini - [windows] "run"
7. windir\system.ini - [boot] "shell"
8. windir\system.ini - [boot] "scrnsave.exe"
9. windir\dosstart.bat - Used in Win95 or 98 when you select the "Restart in MS-DOS mode" in the shutdown menu.
10. windir\system\autoexec.nt
11. windir\system\config.nt
12. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
13. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
14. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
15. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
16. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
17. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
18. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run registry key
19. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run registry key
20. C:\Documents and Settings\All Users\Start Menu\Programs\Startup
21. C:\wont\Profiles\All Users\Start Menu\Programs\Startup
22. C:\Documents and Settings\All Users\Start Menu\Programs\Startup
23. c:\windows\start menu\programs\startup
24. C:\Documents and Settings\LoginName\Start Menu\Programs\Startup
25. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
26. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
27. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
28. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
29. HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\load
30. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify
31. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows
32. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion \ShellServiceObjectDelayLoad
33. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer \SharedTaskScheduler

Application protection

- correct application settings (default values are not always the safest!)
- removing of unnecessary applications
- disabling attachments in e-mails
- disabling execution of hazardous types of files
- installing applications on non-standard ports and non-standard directories
- ...

Next time we go on!

- Security:
  - Secure network infrastructute
  - information for network operation