Komunikacijski protokoli in omrežna varnost

Varnostni elementi: IPsec, SSL in infrastruktura

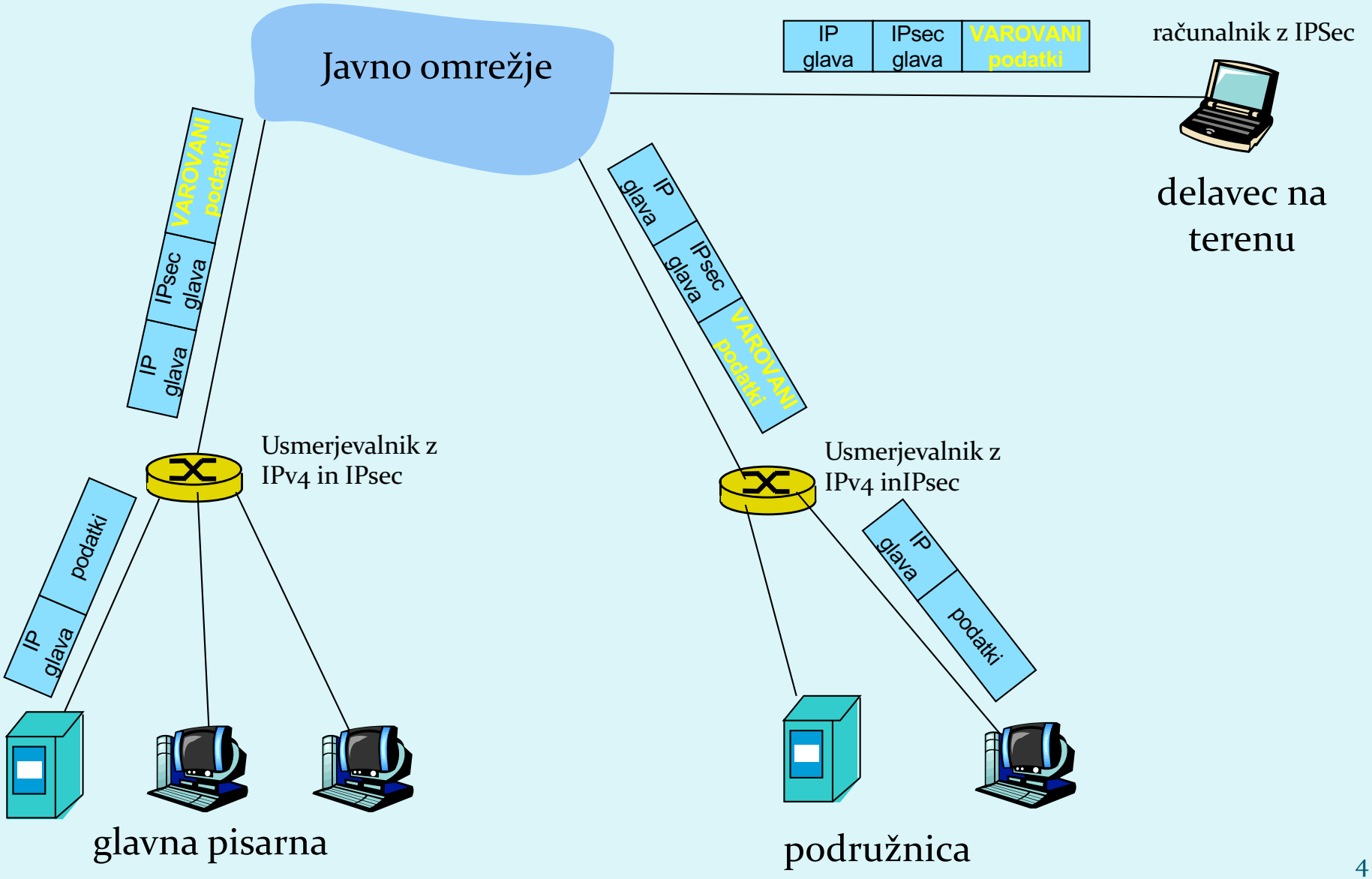
IPSec

- *IP security protocol* (varnost na omrežni plasti)
- uporaba za varovanje povezav med dvema entitetama, uporaba za VPN (navidezna zasebna omrežja)!
- varnost na omrežni plasti:
  - zakrivanje vseh vrst podatkov
    (TCP segment, UDP segment, ICMP sporočilo, OSPF sporočilo itd.)
  - zagotavljanje overovljenosti izvora
  - integriteta podatkov pred spreminjanjem
  - zaščita pred ponovitvijo komunikacije
- RFC 2411: pregled mehanizmov in delovanja IPSec

Navidezna zasebna omrežja (VPN)

- angl. *Virtual Private Network*
- podjetja, ki so na različnih geografskih lokacijah, si lahko želijo visoke varnosti pri komunikaciji. Rešitvi:
  1. gradnja ZASEBNEGA omrežja: podjetje zgradi lastno omrežje, popolnoma ločeno od preostalega Interneta (draga postavitev in vzdrževanje - potrebni usmerjevalniki, povezave, infrastruktura!)
  2. podjetje vzpostavi NAVIDEZNO ZASEBNO omrežje (VPN) z infrastrukturo javnega omrežja:
     - podatki znotraj lokalnih (zasebnih) delov omrežja se prenašajo tradicionalno (IP),
     - podatki, ki potujejo preko javnih delov omrežja se prenašajo zaščiteno (IPSec)

VPN: primer

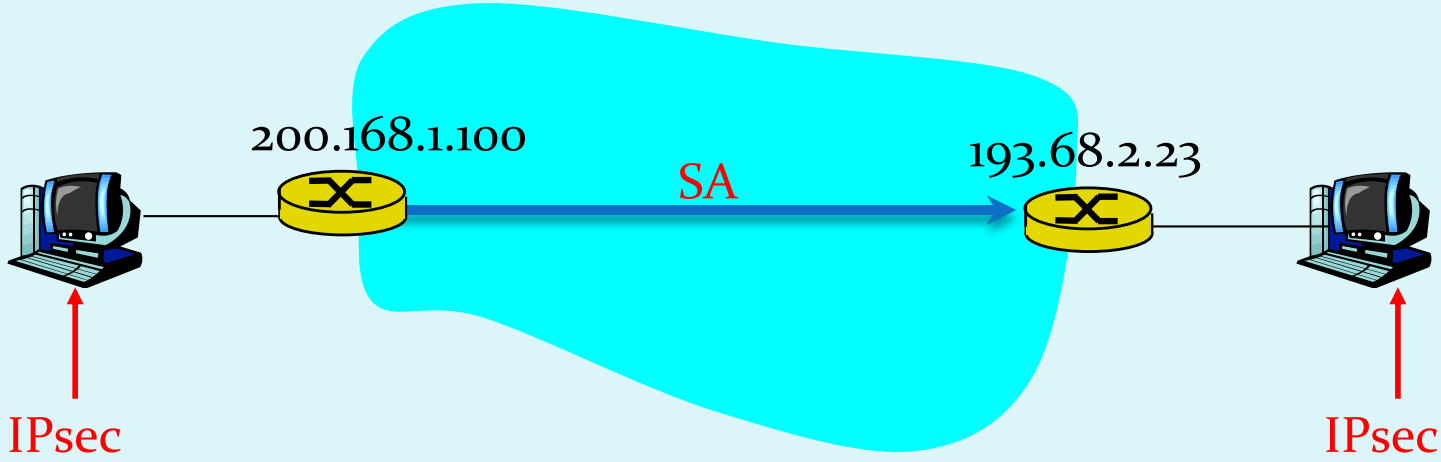

Javno omrežje
IP glava
IPsec glava
VAROVANI podatki
računalnik z IPSec
delavec na terenu
VAROVANI podatki
IPsec glava
IP glava
IP glava
IPsec glava
VAROVANI podatki
Usmerjevalnik z IPv4 in IPsec
Usmerjevalnik z IPv4 inIPsec
IP glava
podatki
IP glava
podatki
glavna pisarna
podružnica

Implementacija IPsec

- mehanizem IPSec ponuja dva protokola varovanja:
  - AH - *Authentication Header*
    - zagotavlja overovljenje izvora in celovitost podatkov
  - *ESP - Encapsulation Security Payload*
    - zagotavlja overovljenost izvora, celovitist podatkov **in** zaupnost podatkov
- za vsako smer IPSec komunikacije je potrebno vzpostaviti SA (*Security Association*)
  - primer: glavna pisarna in podružnica uporabljata dvosmerno komunikacijo. Ravno tako glavna pisarna uporablja dvosmerno komunikacijo z n delavci na terenu. Koliko SA je potrebno vzpostaviti?

$$2 + 2n$$

Vzpostavitev SA



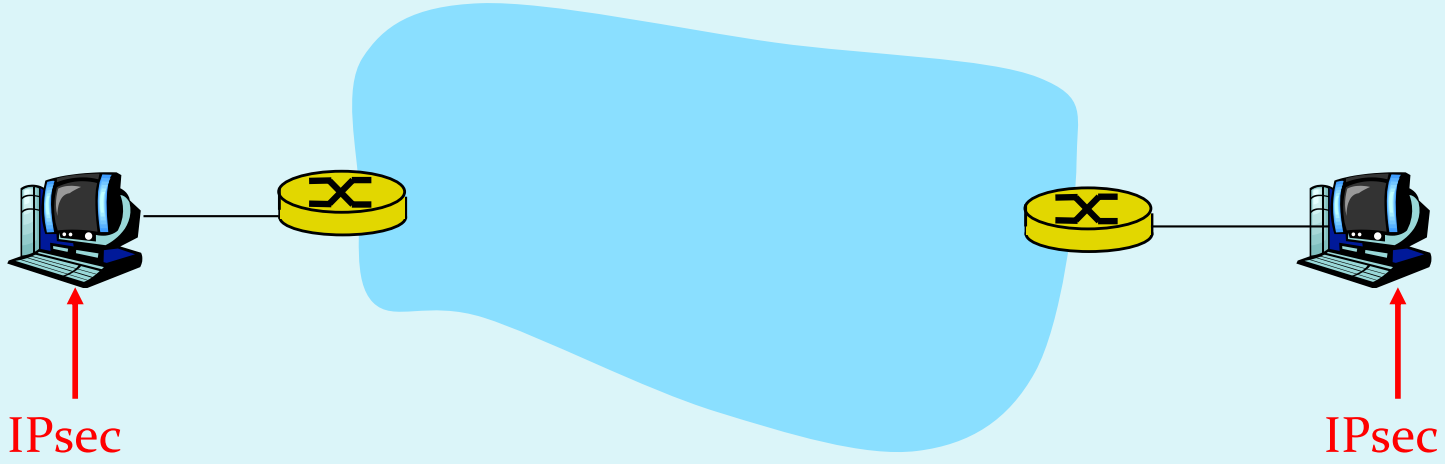
- Usmerjevalnik ima bazo SAD (*Security Association Database*), kjer hrani podatke o SA:
  - 32 bitni ID SA, imenovan SPI (*Security Parameter Index*)
  - izvorni in ponorni IP SA
  - vrsta šifriranja (npr. 3DES) in ključ
  - vrsta preverjanja celovitosti (npr. HMAC-MD5, HMAC-SHA1, ...)
  - ključ za overovitev

2 načina komunikacije

- ***transport mode*** - implementiran med končnimi odjemalci (vmesniki računalnikov), ščiti zgornje plasti protokola. Transparentno vmesnikom, šifrira samo podatke v paketu.
- ***tunnel mode*** - transparentno končnim odjemalcem, usmerjevalnik-usmerjevalnik ali usmerjevalnik-uporabnik. Šifrira podatke in glavo paketa.

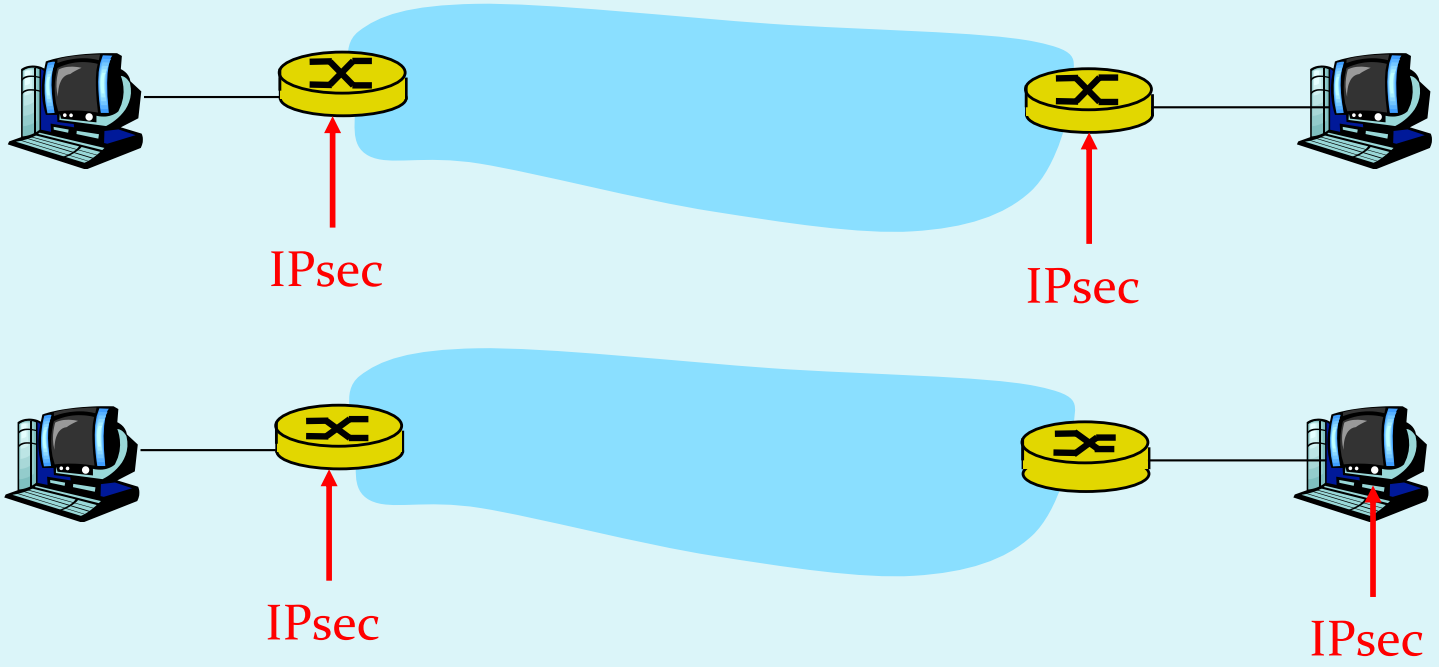
Transport mode z AH	Transport mode z ESP
Tunnel mode z AH	Tunnel mode z ESP

IPsec Transport Mode



- IPsec datagram potuje med končnima sistemoma
- ščitimo le zgornje plasti

IPsec – tunneling mode



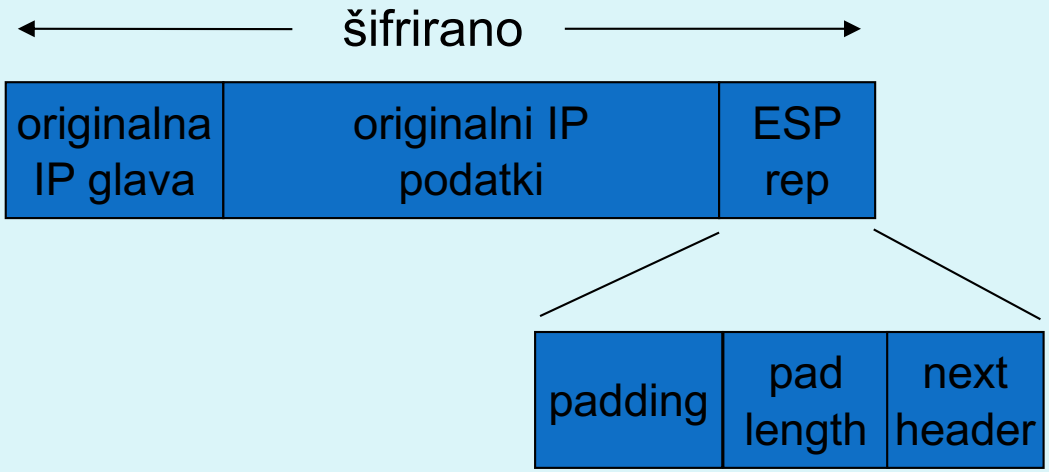
- IPsec se izvaja na končnih usmerjevalnikih
- za odjemalce ni nujno, da izvajajo IPsec

IPsec datagram: tunnel mode in ESP

- Poglejmo si, kako deluje najbolj pogosto uporabljen IPSec način
- Originalni podatki:

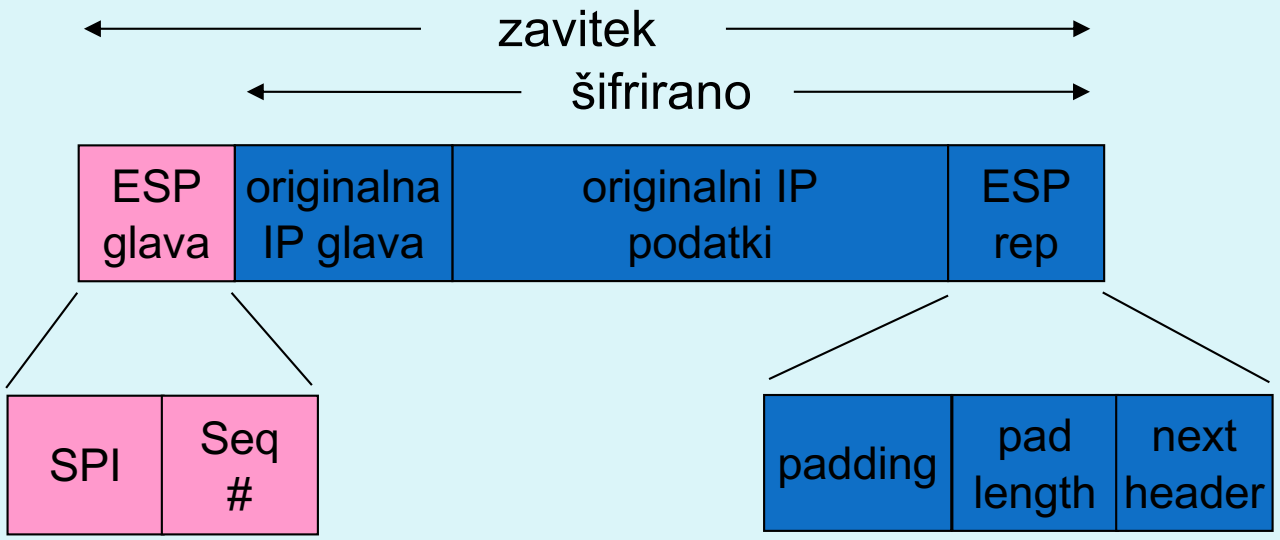
IPsec datagram: tunnel mode in ESP

- na konec datagrama se doda ESP glava (zapolnitev je potrebna za bločno šifriranje, *next header* je protokol, vsebovan v podatkih)
- rezultat se šifrira (algoritem in ključ določa SA)



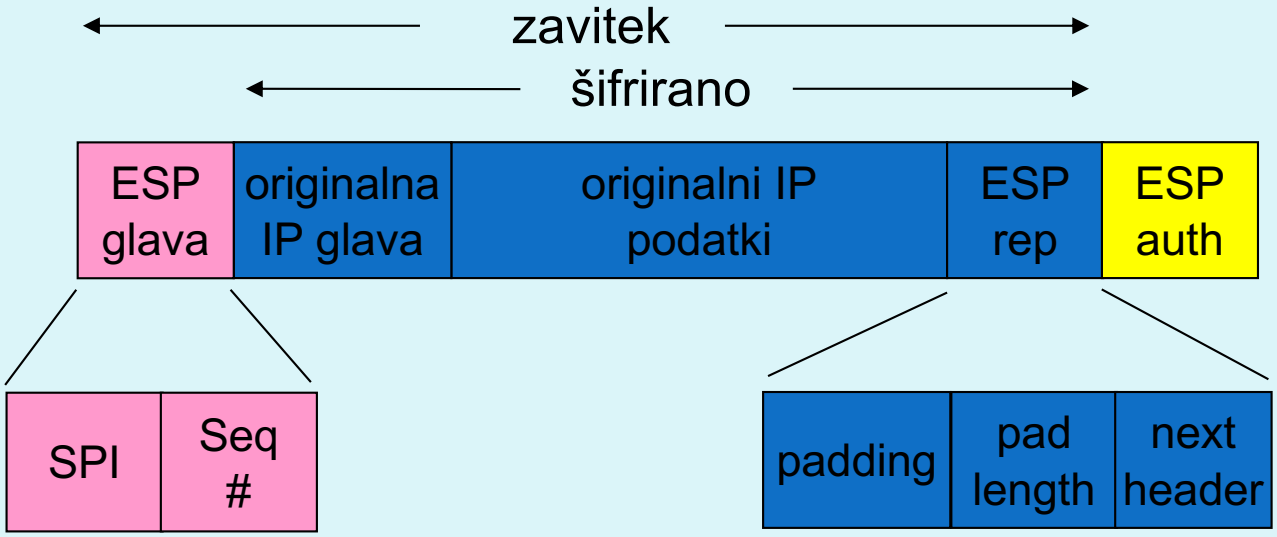
IPsec datagram: tunnel mode in ESP

- doda se ESP glava: rezultat je „*enchilada*“ (zavitek)
  (SPI - indeks SA, ki se ga uporabi za določanje nastavitev,
  Seq# - zaščita proti ponovitvi komunikacije)



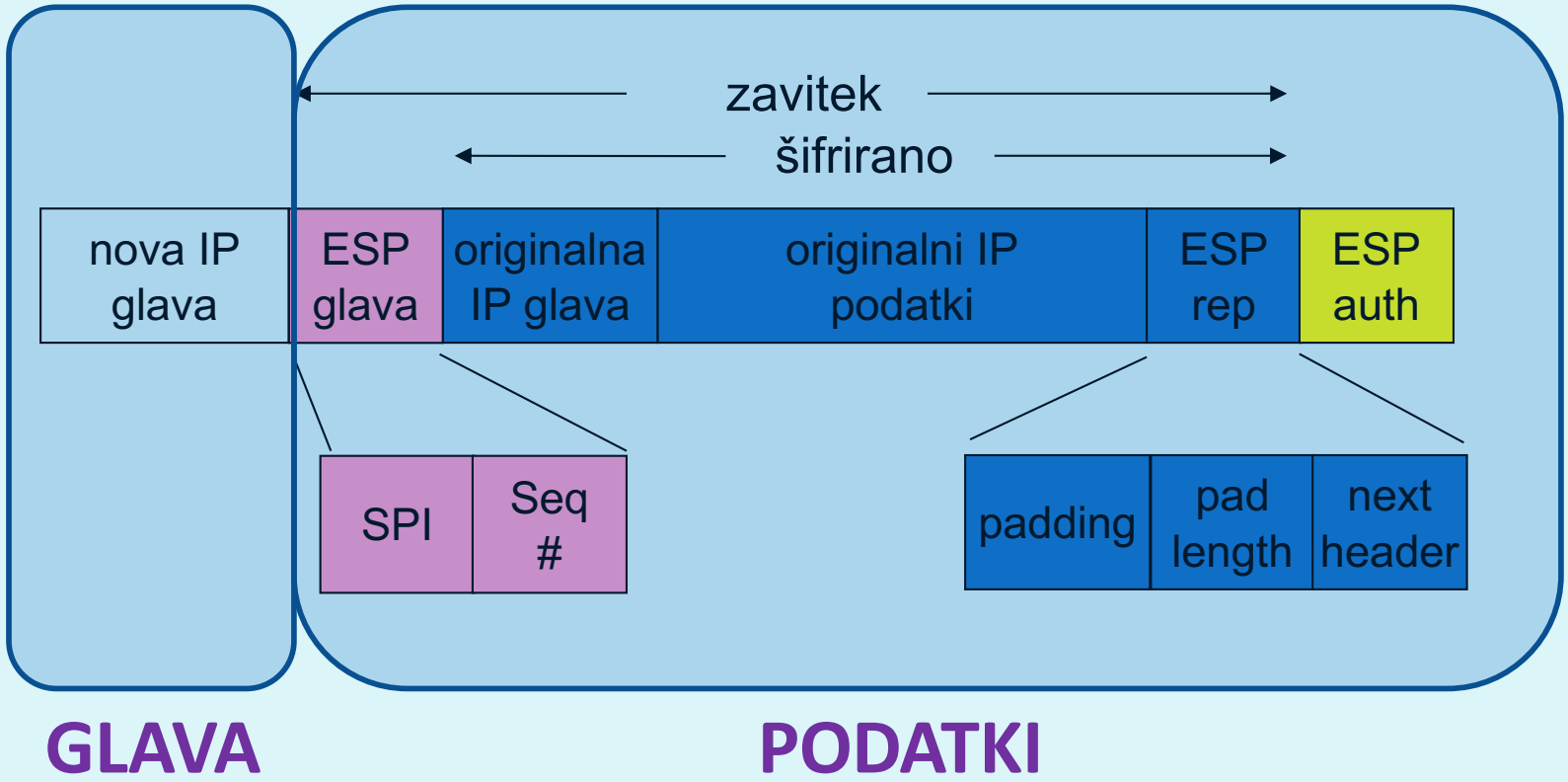
IPsec datagram: tunnel mode in ESP

- doda se polje ESP auth, ki je izračunana zgoščena vrednost cele zavitka (*enchilada*). Algoritem in ključ določa SA.



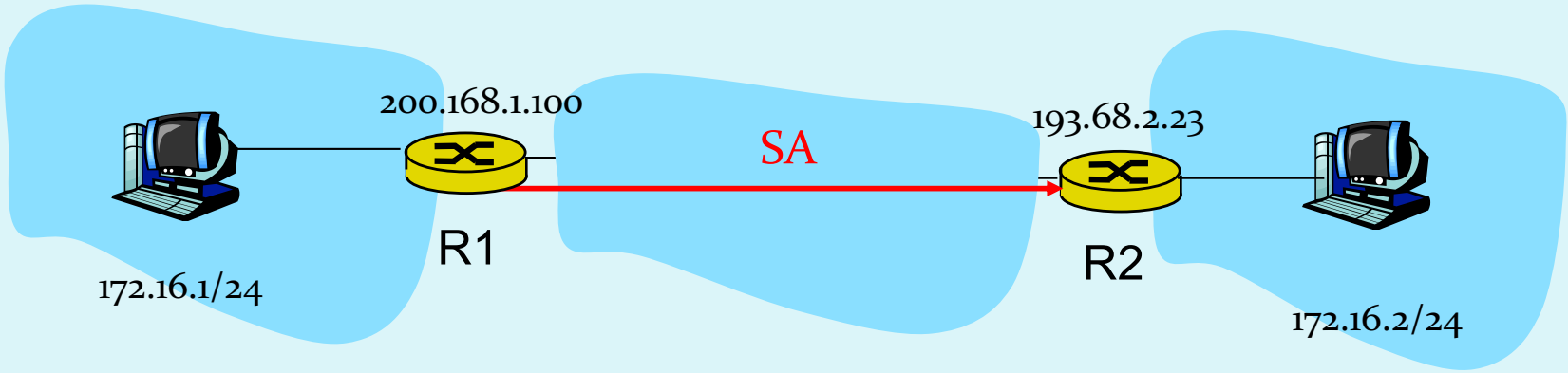
IPsec datagram: tunnel mode in ESP

- izdela se nova IP glava, ki se doda pred podatke
- oblikuje se nov IP paket, ki se klasično pošlje skozi omrežje



IPsec datagram: tunnel mode in ESP

- Kaj je v novi glavi paketa?
  - protokol = 50 (pomeni, da so podatki ESP)
  - IP pošiljatelja in prejemnika sta vozlišči, med katerima poteka IPsec (usmerjevalnika R1 in R2)
- Kaj naredi prejemnik (R2)?
  - iz SPI v glavi poišče podatke o SA, preveri MAC zavitka, preveri Seq#, odšifrira zavitek, odstrani zapolnitev, izloči podatke, posreduje ciljnemu računalniku



Kako izbrati datagrame za IPsec zaščito?

- To določa *Security Policy Database* (SPD): določa, ali naj se datagram ščiti glede na izvorni IP, ponorni IP in tip protokola
- Določa, kateri SA naj se uporabi

- SPD določa „KAJ“ narediti z datagramom
- SAD določa „KAKO“ to narediti!

Kakšno zaščito ponuja IPsec?

- Denimo, da je Cefizelj naš *man-in-the-middle* med R1 in R2. Cefizelj ne pozna ključev. Kaj lahko naredi?
  - Ali lahko vidi vsebino datagrama, izvor, ponor, protokol, port?
  - Ali lahko spremeni bite v paketu?
  - Ali lahko pošilja v imenu R1?
  - Ali lahko ponovi komunikacijo?

Protokol IKE

- IKE (angl. *Internet Key Exchange*), protokol za izmenjavo ključev preko interneta (RFC 2409, RFC 4306, RFC 5282)
- Pri IPsec je potrebno vzpostaviti SA med odjemalci, npr:

  Primer vzpostavljenega 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…

- Ročno določanje SA je nepraktično in zamudno: potrebno ga je določiti za vsako smer komunikacije in vsak par odjemalcev!
- Rešitev: uporabimo protokol *IPsec IKE*

IKE ima 2 fazi

- IKE uporablja PKI ali PSK (*pre-shared key*) za vzajemno overovljenje odjemalcev. Ima dve fazi:
  - Faza 1: Vzpostavi dvosmeren IKE SA (*INIT* in *AUTH*)
    - IKE SA je ločen SA od IPsec SA, ki se uporablja samo za izmenjavo ključev (imenuje se tudi ISAKMP SA)
    - v IKE SA se vzpostavi ključ za varovanje nadaljne komunikacije glede izmenjave ključev (overovljenje se izvede s PSK, PKI ali podpisom)
    - dva načina: *Aggressive mode* (krajši, vendar razkrije identiteto odjemalcev) in *Main mode* (daljši, skrije identiteto)
  - Faza 2: IKE generira ključe za druge storitve, kot je npr IPsec. Vzpostavi se torej IPsec SA (*CREATE_CHILD* in *INFO*)
    - edini način: *Quick Mode*

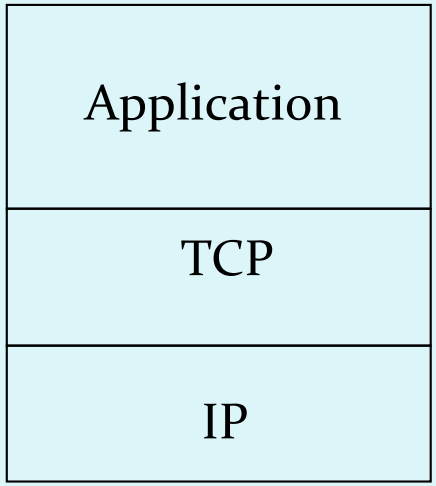
SSL

SSL: Secure Sockets Layer

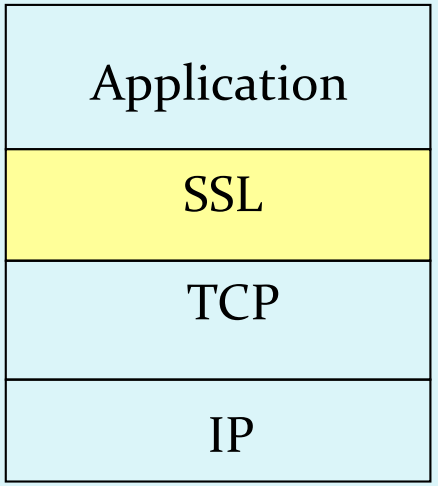
- Široko uporabljen varnosti protokol
  - podprt skoraj v vseh brskalnikih in na vseh strežnikih (https)
  - z uporabo SSL se opravi za 10 milijard dolarjev (2010) nakupov letno
- Razvil ga je Netscape leta 1993
- Več vrst
  - TLS: transport layer security, RFC 2246
- Zagotavlja zaupnost, celovitost, overovljenost
- Cilji pri razvoju:
  - uporaba pri spletnih transakcijah
  - zakrivanje podatkov (še posebej številk kreditnih kartic)
  - overovljenje spletnih strežnikov
  - možnost overovitve odjemalca
  - čim manjši napor pri opravljanju nakupa pri drugem prodajalcu

SSL and TCP/IP

- Dostopen vsem TCP aplikacijam preko aplikacijskega vmesnika SSL



Običajna aplikacija



Aplikacija s SSL

Zasnova SSL

Lahko bi ga zasnovali na osnovi kriptografije PKI (šifriranje z javnim ključem prejemnika, zasebnim ključem pošiljatelja, uporaba zgoščevalnih funkcij), vendar...

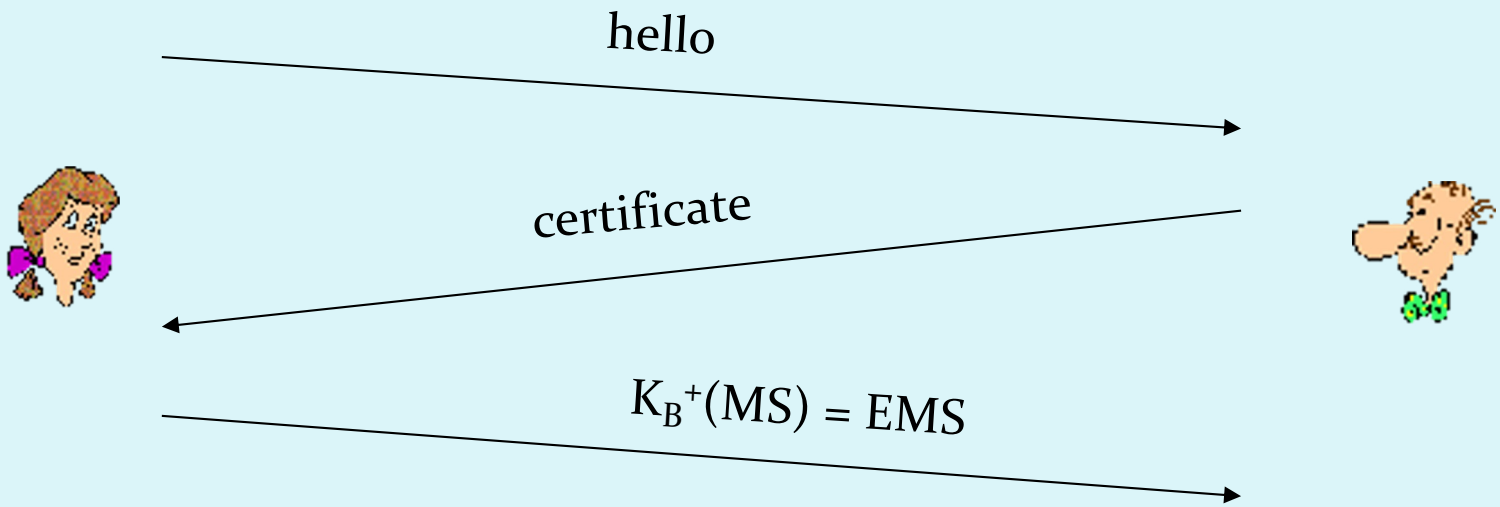
- želimo pošiljati **TOK BYTOV** in interaktivne podatke, ne sporočila – ***povezavni način prenosa***,
- za eno povezavo želimo imeti **MNOŽICO** ključev, ki se spreminjajo,
- kljub temu želimo uporabljati certifikate – overovitev
  - ideja: uporabimo jih pri rokovanju

Poenostavljeni SSL

Poglejmo najprej poenostavljeno idejo protokola SSL. Ta vsebuje naslednje 4 faze:

- 1. ROKOVANJE: Ana in Brane uporabita certifikate, da se vzajemno overovita in izmenjata glavni ključ
- 2. IZPELJAVA KLJUČA: Ana in Brane uporabita izmenjani glavni ključ, da izpeljeta množico ključev
- 3. PRENOS PODATKOV: Podatki, ki se prenašajo, so združeni v ZAPISE.
- 4. ZAKLJUČEK POVEZAVE: Za varen zaključek povezave se uporabijo posebna sporočila

Poenostavljeni SSL: Rokovanje



- MS = glavni ključ (*master secret*)
- EMS = šifrirani glavni ključ (*encrypted master secret*)
- K_B^+ - Branetov javni ključ

Poenostavljeni SSL: Izpeljava ključa

- Slaba praksa je *uporabljati isti ključ za več šifrirnih operacij*, zato: uporabimo poseben ključ za zakrivanje in posebnega za preverjanje integritete (MAC)
- Uporabljamo torej 4 ključe:
  - K_c = ključ za zakrivanje podatkov, poslanih od odjemalca strežniku
  - M_c = ključ za overjanje podatkov, poslanih od odjemalca strežniku
  - K_s = ključ za zakrivanje podatkov, poslanih od strežnika odjemalcu
  - M_s = ključ za overjanje podatkov, poslanih od strežnika odjemalcu
- Ključi se izpeljejo z uporabo posebne funkcije. Ta uporablja glavni ključ (*Master Secret*) in dodatne (naključne) podatke za generiranje naslednjih ključev

Poenostavljeni SSL: Pošiljanje podatkov

- Kako preveriti celovitost podatkov?
  - če bi pošijali po zlogih (byte-ih), kam bi pripeli MAC (podpis sporočila)?
  - Tudi če MAC pošljemo po zaključku celega prenosa (vseh zlogov), nimamo vmesnega preverjanja celovitosti!

- REŠITEV: Tok podatkov razbijemo v **ZAPISE**
  - vsakemu zapisu pripnemo podpis
  - prejemnik lahko reagira na (ne)veljavnost celovitosti posameznega zapisa

Poenostavljeni SSL: Pošiljanje podatkov

- Problem 1: številka paketa se nahaja nešifrirana v glavi TCP. Kaj lahko naredi napadalec?
  - napadalec lahko zajame in ponovi komunikacijo?
  - prešteviłči vrstni red paketov?
  - prestreže in odstrani paket?

- REŠITEV: pri računanju MAC upoštevaj številko paketa
  - MAC = MAC(ključ M_x, zaporedna_številka || podatki)
  - nimamo ločene številke paketa
  - zaščita proti ponovitvi komunikacije: uporabi enkratni žeton

Poenostavljeni SSL: Pošiljanje podatkov

- Problem 2: napadalec predčasno zaključi sejo
  - Ena ali obe strani dobita vtis, da je podatkov manj, kot jih je.

- REŠITEV: uvedimo poseben „tip zapisa“, ki nosi posebno vrednost, če gre za zaključni paket
  - npr: 0 pomeni podatke, 1 pomeni zaključek
  - uporabimo vrednost pri izračunu MAC
    MAC = MAC(ključ M_x, zaporedna_št || tip || podatki)

Poenostavljeni SSL: Primer

hello

certifikat, žeton

$K_B^+(MS) = EMS$

type 0, seq 1, data

type 0, seq 2, data

type 0, seq 1, data

zakrito

type 0, seq 3, data

type 1, seq 4, close

type 1, seq 2, close

Pravi SSL: podrobnosti

- Kakšne so dolžine polj v protokolu?
- Kateri protokoli za zakrivanje naj se uporabijo? Dogovor o uporabi protokola:
  - Želimo, da odjemalec in strežnik lahko izbirata in se dogovarjata o šifrirnih algoritmih (angl. *negotiation*, odjemalec ponudi, strežnik izbere)
  - Najpogostejši simetrični algoritmi
    - DES – Data Encryption Standard: block
    - 3DES – Triple strength: block
    - RC2 – Rivest Cipher 2: block
    - RC4 – Rivest Cipher 4: stream
  - Najpogostejši algoritem za PKI šifriranje
    - RSA

Pravi SSL: Rokovanje

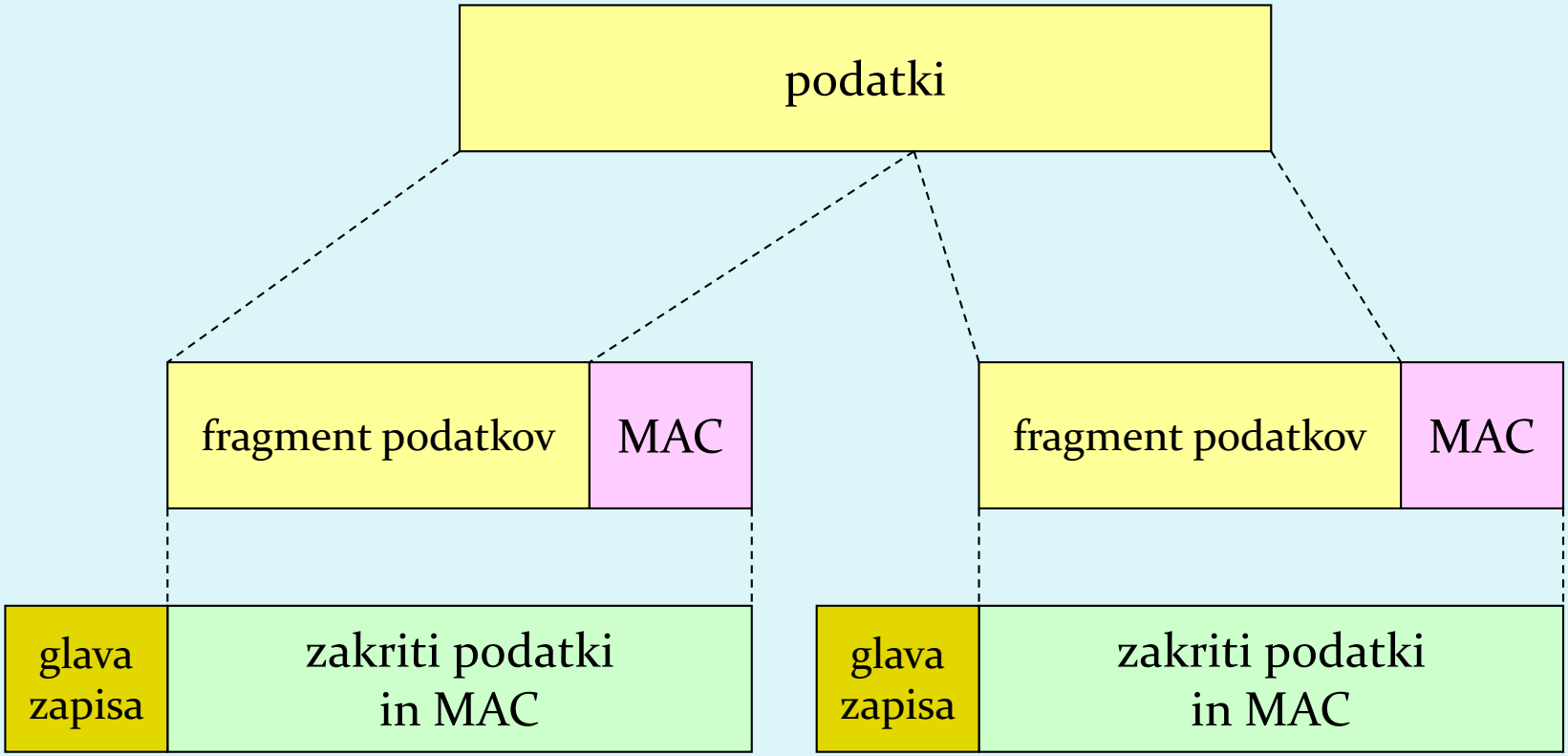
- Poenostavljeni SSL: hello->, <-certifikat, šifriran MS->
- Pravi SSL dejansko izvaja: overovljenje strežnika, izbiro algoritmov, določanje ključev, overovitev odjemalca (opcijsko)
- Postopek:

1. Odjemalec pošlje seznam podprtih algoritmov + žeton
2. Strežnik izbere algoritem s seznama, vrne izbiro, certifikat (podpisan javni ključ) in svoj žeton
3. Odjemalec preveri certifikat, tvori PMS, z javnim ključem strežnika ga šifrira in pošlje strežniku
4. Odjemalec in strežnik neodvisno izračunata šifrirne in MAC ključe iz PMS in žetonov.
5. Odjemalec pošlje MAC od vseh sporočil v rokovanju.
6. Strežnik pošlje MAC vseh sporočil v rokovanju.

Pravi SSL: Rokovanje

1. Zakaj izmenjava MAC v korakih 5 in 6?
   - odjemalec običajno ponudi več algoritmov, nekateri so šibki, drugi močnejši. Napadalec bi lahko izbrisal iz ponudbe močnejše algoritme.
   - Zadnji dve sporočilo zagotavljata integriteto vseh prenešenih sporočil in preprečita tak napad
2. Zakaj uporaba žetonov?
   - Denimo, da Cefizelj posluša sporočila med Ano in Branetom ter jih shrani. Naslednji dan pošlje Cefizelj Branetu popolnoma enaka sporočila, kot jih je prejšnji dan poslala Ana:
     - Če ima Brane trgovino, bo mislil, da Ana ponovno naroča artikle,
     - Brane za vsako komunikacijo uporabi drug žeton, tako Cefizelj ne bo mogla ponoviti iste komunikacije

SSL: pretvorba v zapise



- GLAVA ZAPISA: vrsta vsebine (1B); SSL verzija (2B); dolžina (3B)
- MAC: zaporedna_številka; MAC ključ M_x
- FRAGMENT: vsak je dolg do 2^{14} bytes (~16 Kbytes)

Primer pravega rokovanja

handshake: ClientHello

handshake: ServerHello

handshake: Certificate

handshake: ServerHelloDone

handshake: ClientKeyExchange

ChangeCipherSpec

handshake: Finished

Od tu naprej je vse zakrito

ChangeCipherSpec

handshake: Finished

application_data

application_data

Alert: warning, close_notify

SSL: izpeljava ključev

- Žetona odjemalca in strežnika ter PMS se uporabijo v funkciji, ki izračunava psevdo-naključna števila. Dobimo MS *(master secret).*
- MS in novi žetoni se vstavijo v drugi naključni generator, dobimo BLOK. BLOK se razreže na 6 delov, da se dobi:
  - MAC ključ odjemalca
  - MAC ključ strežnika
  - šifrirno ključ odjemalca
  - šifririni ključ strežnika

  } enako kot pri poenostavljenem SSL!

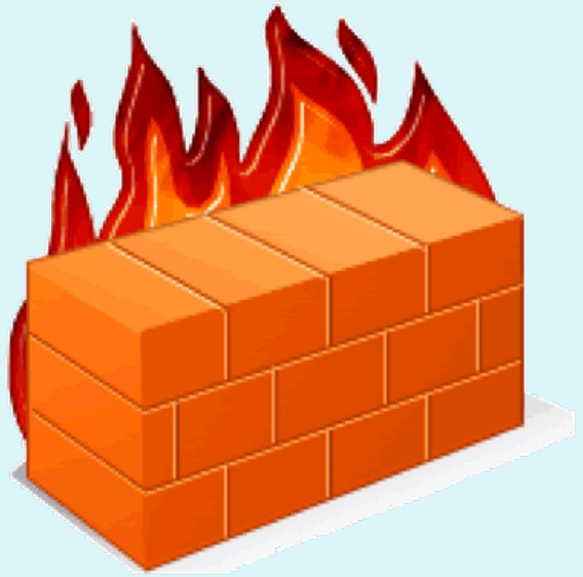
  - inicializacijski vektor (IV) odjemalca
  - inicializacijski vektor (IV) strežnika

  } KAJ JE TOLE?

potrebna sta, kadar uporabljamo simetričen algoritem z bločnim šifriranjem (3DES ali AES), ki potrebuje inicializacijo!

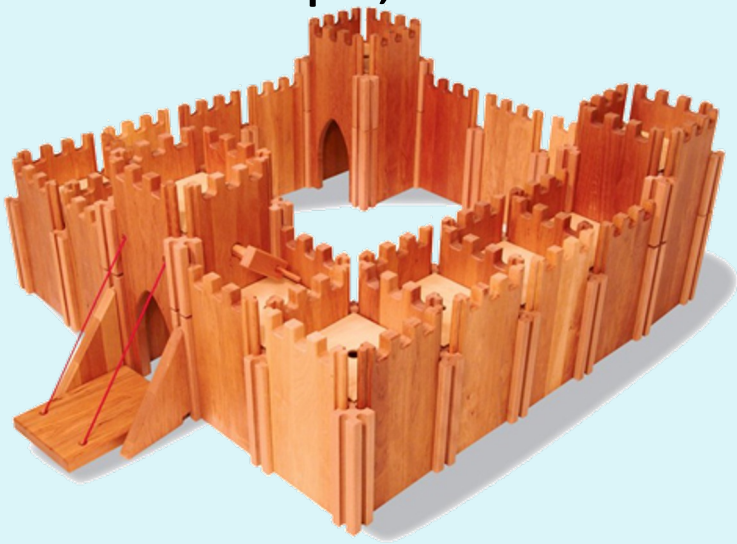
Operativna varnost:

požarne pregrade in sistemi za zaznavanje vdorov



Varnost v omrežju

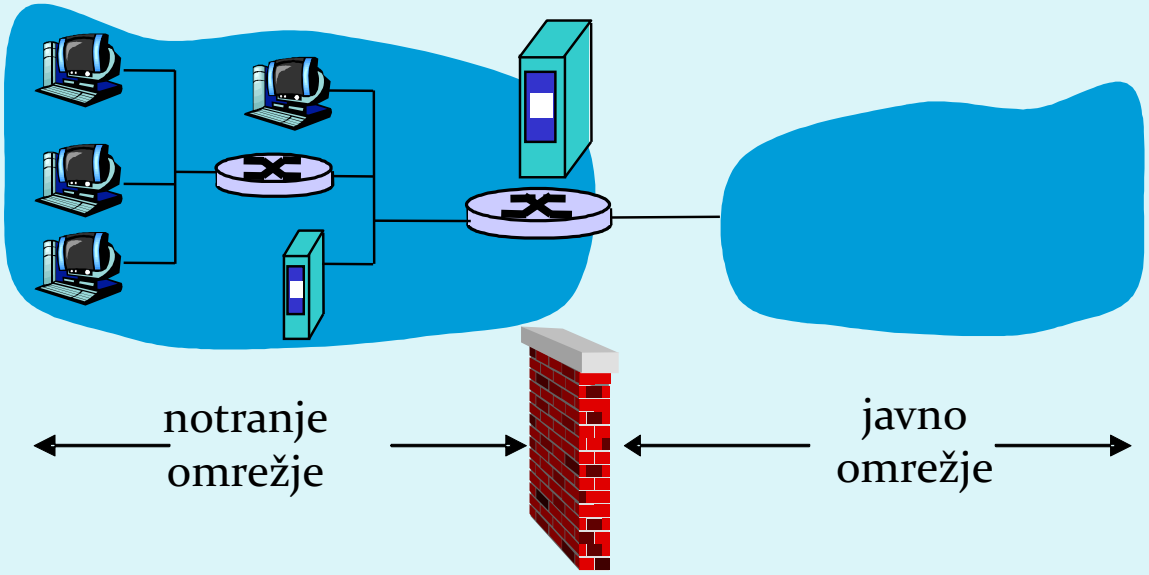
- Administrator omrežja lahko uporabnike deli na:
  - dobri (*good guys*): uporabniki, ki legitimno uporabljajo vire omrežja, pripadajo organizaciji,
  - slabi (*bad guys*): vsi ostali, njihove dostope moramo skrbno nadzorovati
- Omrežje ima običajno eno samo točko vstopa, nadzorujemo dostope v njej:
  - požarna pregrada (*firewall*)
  - sistem za zaznavanje vdorov (*IDS, intrusion detection system*)
  - sistem za preprečevanje vdorov (*IPS, intrusion prevention system*)



Požarna pregrada

osami notranje omrežje od velikega javnega omrežja, določenim paketom dovoli prehod, druge zaustavi. Ima 3 naloge:

- filtrira VES promet,
- prepušča samo promet, ki je DOPUSTEN glede na politiko,
- je IMUN na napade



POŽARNA PREGRADA

Požarna pregrada: vrste filtriranj

1. brezstanjsko filtriranje paketov (angl. *stateless, traditional*);
   „filtriranje na omrežni plasti“
2. stanjsko filtriranje paketov (angl. *stateful filter*)
   „filtriranje na prenosni plasti“
3. aplikacijski prehodi (angl. *application gateways*)
   „filtriranje na aplikacijski plasti“

Brezstanjsko filtriranje paketov

- filtriranje običajno izvaja že ,,usmerjevalnik'', ki meji na javno omrežje. Na podlagi vsebine paketov se odloča, ali bo posredoval **posamezen paket**, odločitev na podlagi:
  - IP izvornega/ponornega naslova
  - številke IP protokola: TCP, UDP, ICMP, OSPF itd.
  - TCP/UDP izvornih in ciljnih vrat
  - tip sporočila ICMP
  - TCP SYN (vzpostavitev povezave!) in ACK bits (ACK=1 velja za prvi segment pri povezovanju)

Brezstanjsko filtriranje paketov: primeri

- Primer 1: blokiraj dohodne datagrame z IP protokolom 17 (UDP) in izvornimi ali ciljnimi vrati 23 (telnet)
  - učinek: filtriramo vse (i) dohodne in odhodne UDP komunikacije in (ii) telnet povezave.
- Primer 2: Blokiraj dohodne TCP segmente z zastavico ACK=0.
  - učinek: onemogočimo zunanjim odjemalcem, da vzpostavijo povezavo z notranjimi odjemalci, dovolimo pa povezovanje v obratno smer (navzven)

Brezstanjsko filtriranje paketov: primeri

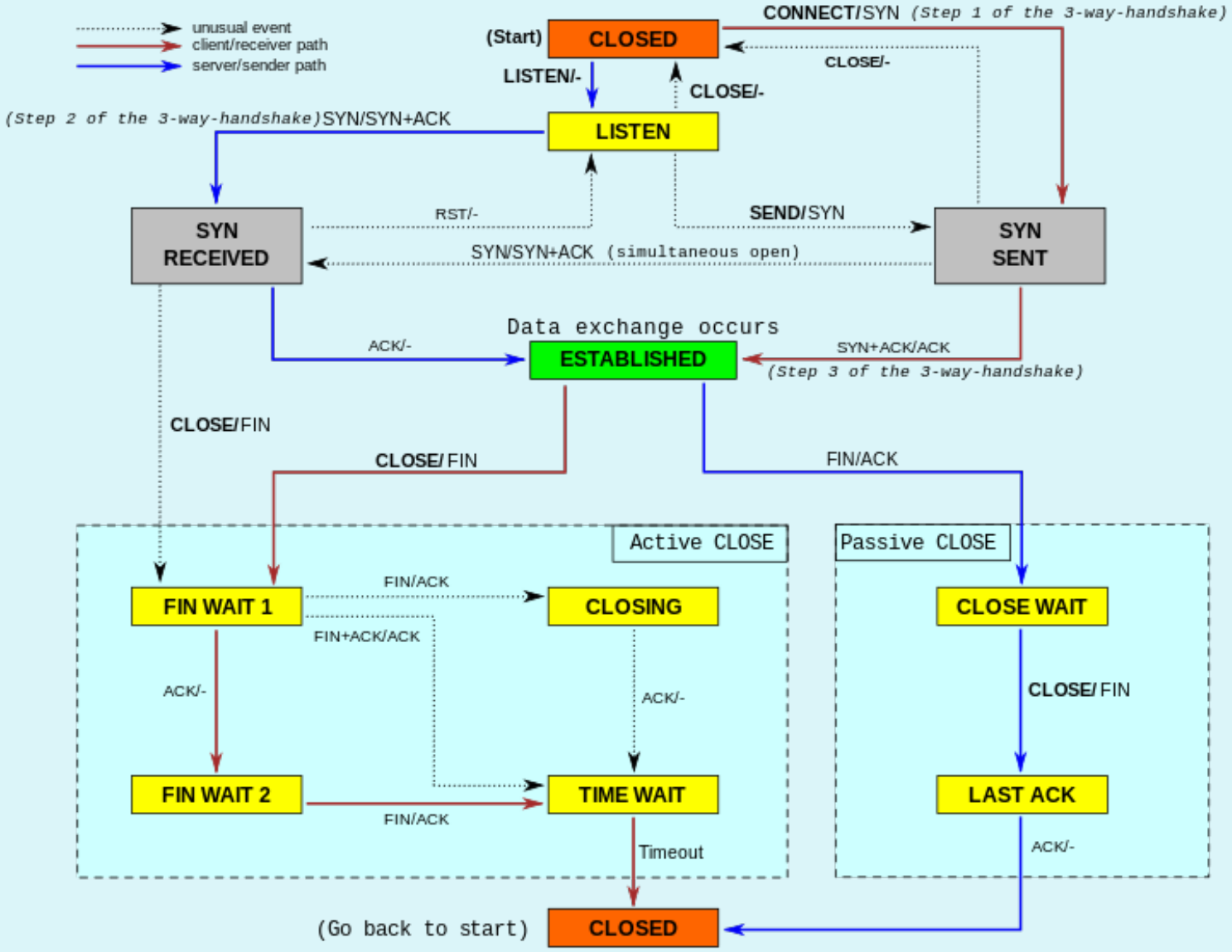
Želimo doseči:	**Nastavitev požarne pregrade**
Onemogočiti dostop navzven do poljubnega spletnega strežnika.	Zavrzi vse pakete, naslovljene na poljuben IP naslov in na vrata 80
Onemogočiti vse dohodne TCP povezave razen tistih, ki so namenjene javnemu spletnemu strežniku v podjetju (130.207.244.203).	Zavrzi vse dohodne TCP SYN pakete razen tistih, namenjenih IP naslovu 130.207.244.203, vrata 80
Preprečiti napad Smurf DoS – uporaba oddajana (*broadcast*) za preobremenitev storitev.	Zavrzi vse ICMP pakete, naslovljene na oddajni naslov omrežja (npr. 130.207.255.255).
Preprečiti analizo omrežja s *traceroute*	Zavrzi vse odhodne pakete ICMP s sporočilom "TTL expired"

Brezstanjsko filtriranje: Dostopovni seznami

- dostopovni seznam (angl. ACL, *access control list*)
- tabela pravil, upošteva se jo od zgoraj navzdol.
- zapisi so par: (**pogoj**, **akcija**)
- primer: onemogoči ves promet razen WWW navzven in DNS v obe smeri

izvorni naslov	ciljni naslov	protokol	izvorna vrata	ciljna vrata	zastavica	akcija
222.22/16	izven 222.22/16	TCP	> 1023	80	any	dovoli
izven 222.22/16	222.22/16	TCP	80	> 1023	ACK	dovoli
222.22/16	izven 222.22/16	UDP	> 1023	53	---	dovoli
izven 222.22/16	222.22/16	UDP	53	> 1023	----	dovoli
all	all	all	all	all	all	zavrzi

TCP avtomat


unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
(Step 2 of the 3-way-handshake) SYN/SYN+ACK
LISTEN
SYN RECEIVED
RST/-
SEND/SYN
SYN SENT
SYN/SYN+ACK (simultaneous open)
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3-way-handshake)
CLOSE/FIN
CLOSE/FIN
FIN/ACK
Active CLOSE
Passive CLOSE
FIN WAIT 1
FIN/ACK
CLOSING
FIN+ACK/ACK
CLOSE WAIT
ACK/-
ACK/-
CLOSE/FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
ACK/-
(Go back to start)
CLOSED

Stanjsko filtriranje paketov

- angl. *stateful filter*, upošteva povezavo in njeno trenutno stanje (TCP prenosni protokol)
  - izolirano filtriranje lahko dovoli vstop nesmiselnim paketom (npr. vrata = 80, ACK =1; čeprav notranji odjemalec ni vzpostavil povezave) :

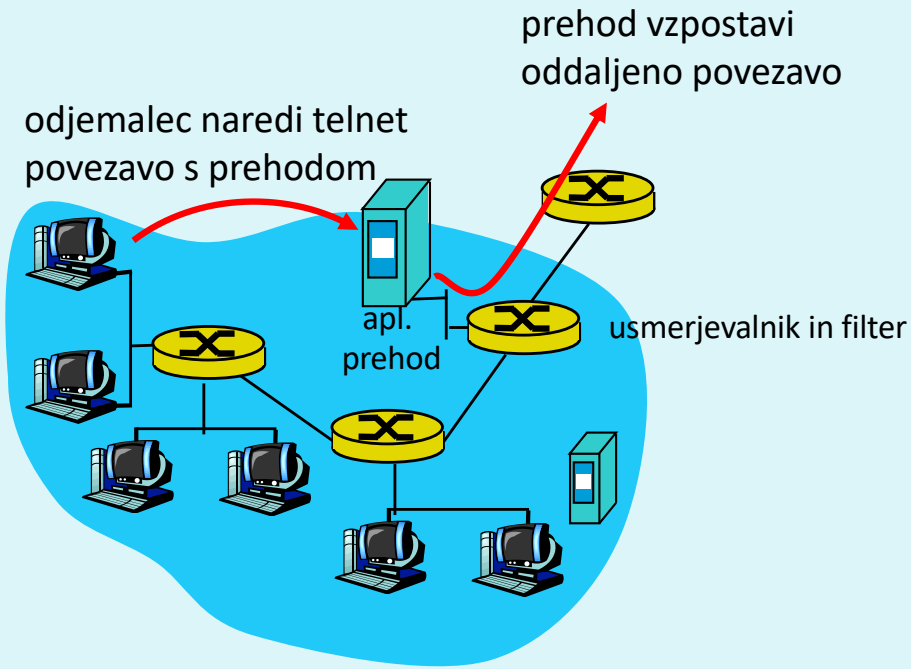
- IZBOLJŠAVA: **stanjsko filtriranje paketov** spremlja in vodi evidenco o stanju vsake vzpostavljeni TCP povezavi
  - zabeleži vzpostavitev povezave (SYN) in njen konec (FIN): na tej podlagi odloči, ali so paketi smiselni
  - po preteku določenega časa obravnavaj povezavo kot neveljavno (timeout)
  - uporabljaj podoben dostopovni seznam, ki določa, kdaj je potrebno kontrolirati veljavnost povezave (angl. *check connection*)

Stanjsko filtriranje paketov

izvorni naslov	ciljni naslov	protokol	izvorna vrata	ciljna vrata	zastavica	akcija	preveri povezavo
222.22/16	izven 222.22/16	TCP	> 1023	80	any	dovoli	
izven 222.22/16	222.22/16	TCP	80	> 1023	ACK	dovoli	X
222.22/16	izven 222.22/16	UDP	> 1023	53	---	dovoli	
izven 222.22/16	222.22/16	UDP	53	> 1023	----	dovoli	X
all	all	all	all	all	all	zavrzi	

Aplikacijski prehodi

- omogočajo dodatno filtriranje glede na izbiro uporabnikov, ki lahko uporabljajo določeno storitev
- omogočajo filtriranje na podlagi podatkov na aplikacijskem nivoju poleg polj IP/TCP/UDP.

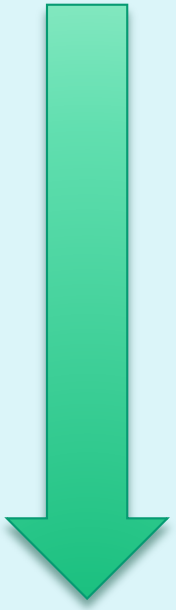


1. vsi uporabniki vzpostavljajo telnet povezavo preko prehoda,
2. samo za avtorizirane uporabnike prehod vzpostavi povezavo do ciljnega strežnika. Prehod posreduje podatke med 2 povezavama,
3. usmerjevalnik blokira vse telnet povezave razen tistih, ki izvirajo od prehoda

Aplikacijski prehodi

Tudi aplikacijski prehodi imajo omejitve:

- če uporabniki potrebujejo več aplikacij (telnet, HTTP, FTP itd.), potrebuje vsaka aplikacija svoj aplikacijski prehod,
- odjemalce je potrebno nastaviti, da se znajo povezati s prehodom (npr. IP naslov medstrežnika v brskalniku)

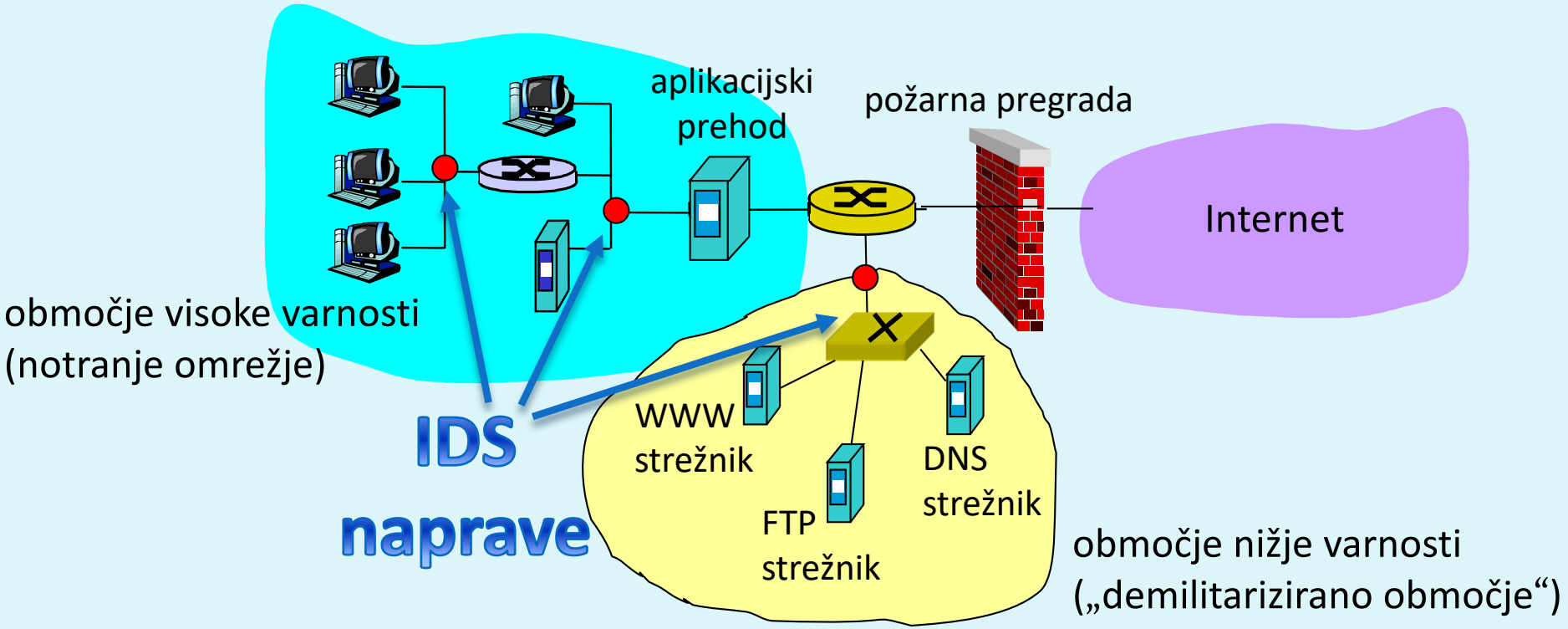


Sistemi za zaznavanje vdorov

- Požarna pregrada kot filter paketov filtrira samo na podlagi glave IP, TCP, UCP in ICMP, kar ne omogoča zaznavanja vseh napadov - za to je potrebno pogledati tudi podatke v paketu
  - primeri napadov: pregledovanje vrat (*port scan*), pregledovanje TCP vrat (*TCP stack scan*), DoS napad, črvi, virusi, napadi na OS, napadi na aplikacije
- dodatna naprava - IDS, ki izvaja **poglobljeno analizo paketov**. Na podlagi vstopa sumljivih paketov v omrežje lahko naprava prepreči njihov vstop ali razpošlje obvestila.
  - sistem za zaznavanje vdorov (IDS) pošlje sporočilo o potencialno škodljivem prometu
  - sistem za preprečevanje vdorov (IPS) filtrira sumljiv promet
  - Cisco, CheckPoint, Snort IDS

Sistemi za zaznavanje vdorov

- v omrežju imamo lahko več IDS/IPS naprav (koristno zaradi zahtevnega primerjanja vsebin paketov s shranjenimi vzorci)



Načini zaznavanja vdorov

Kako deluje IDS/IPS?

- primerjava s shranjenimi vzorci napadov (angl. ***signatures***)
- opazovanje netipičnega prometa (angl. ***anomaly-based***)

Zaznavanje z vzorci napadov

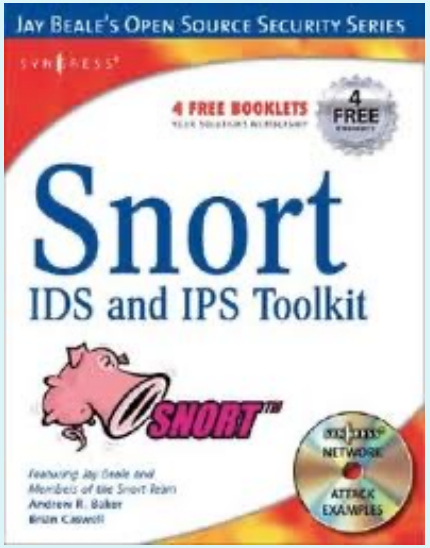
- vzorci napadov lahko hranijo izvorni IP, ponorni IP, protokol, zaporedje bitov v podatkih paketa, lahko so vezani na serijo paketov
- varnost je torej odvisna od baze znanih vzorcev; IDS/IPS slabo zaznava še nevidene napade
- možni lažni alarmi
- zahtevno procesiranje (lahko spregleda napad)

Zaznavanje z zaznavanjem netipičnega prometa

- sistem opazuje običajen promet in izračuna statistike, vezane nanj
- sistem reagira na statistično neobičajen promet (npr. nenadno velik delež ICMP paketov)

- možno zaznavanje še nevidenih napadov
- težko ločevanje med normalnim in nenavadnim prometom

Primer IDS/IPS sistema



- Snort IDS
  - public-domain, odprtokodni IDS za Linux, UNIX, Windows (uporablja isto knjižnico za branje omrežnega prometa kot Wireshark)
  - primer vzorca napada

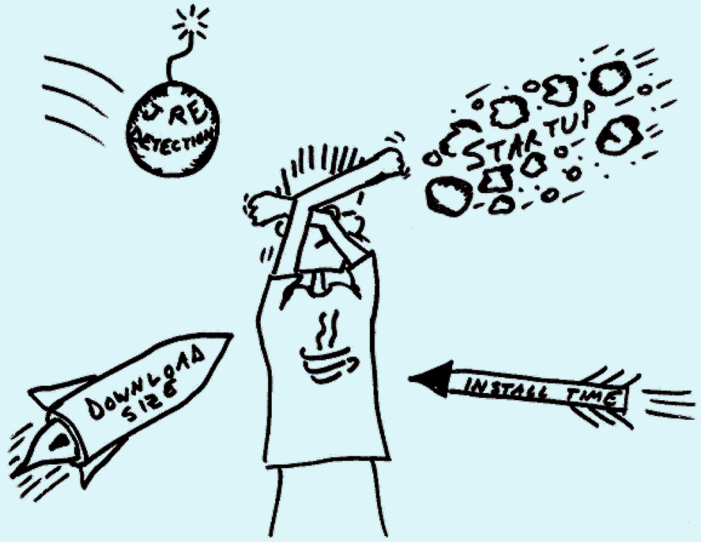
```
alert icmp $EXTERNAL_NET any -> $HOME_NET any
(msg:"ICMP PING NMAP"; dsize: 0; itype: 8;)
```

sporočilo za administratorja

prazen paket (dolžina 0) in ICMP tip 8 (=PING) sta lastnosti NMAP napada

reagiraj na VES DOHODNI ICMP promet

Napadi in grožnje



Pogosti napadi na omrežne sisteme

- **NAMEN?** Namenjeni so škodovanju ali obhodu računalniških in omrežnih funkcij.
- **ZAKAJ?** Denarna dobrobit, škodovalnost, poneverbe, ekonomske dobrobiti.
- **KAKO?** Ogrožanje zaupnosti, integritete in razpoložljivosti omrežnih sistemov
  - napadi s spreminjanjem informacij (*modification attack*)
  - zanikanje komunikacije (*repudiation attack*)
  - odpoved delovanja sistema (*denial-of-service attack*)
  - nepooblaščen dostop (*access attack*)

Pogosti napadi na omrežne sisteme

Pogosti napadi

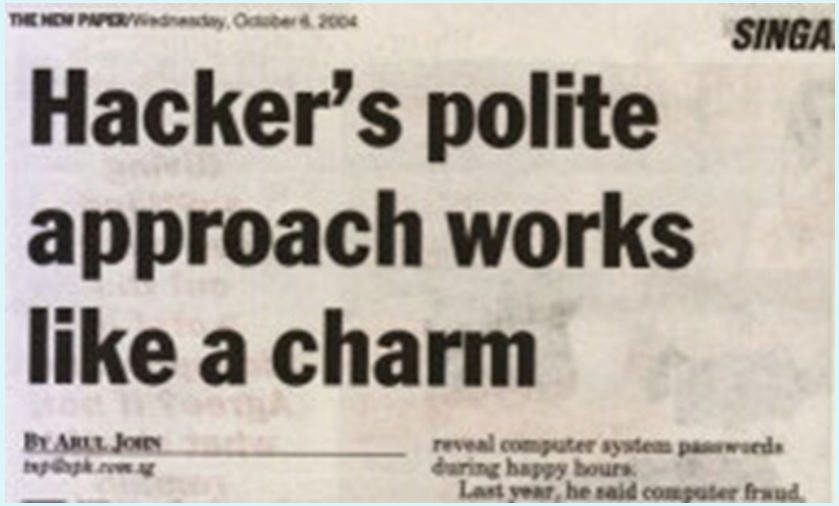
1. **pregledovanje sistema** (*reconnaissance*): napadalec z različnimi tehnikami poskuša odkriti arhitekturo sistema, storitve v njem itd.
   - pomaga pripraviti napad na sistem
   - primer (*war-dialing*) napadalec s klicanjem na naključne telefonske številke poskuša odkriti klicno številko modema za dostop do omrežja

Pogosti napadi

2. **prisluškovanje** (*eavesdropping*): prestrezanje omrežnega prometa, prisotno zlasti pri brezžičnih omrežjih (napadalec pridobi gesla, številke kreditnih kartic, ...)
   - pasivni napadalec
   - aktivni napadalec

Pogosti napadi

3. **ugibanje gesel** (groba sila (*brute force*), napad s slovarjem)
4. **virusi, črvi, trojanci**
5. **izkoriščanje šibkosti v programski opremi**
6. **socialni inženiring** (preko e-pošte, telefona, storitev)


THE NEW PAPER/Wednesday, October 6, 2004

SINGA

Hacker's polite approach works like a charm

BY ARUL JOHN

reveal computer system passwords during happy hours.

Last year, he said computer fraud.

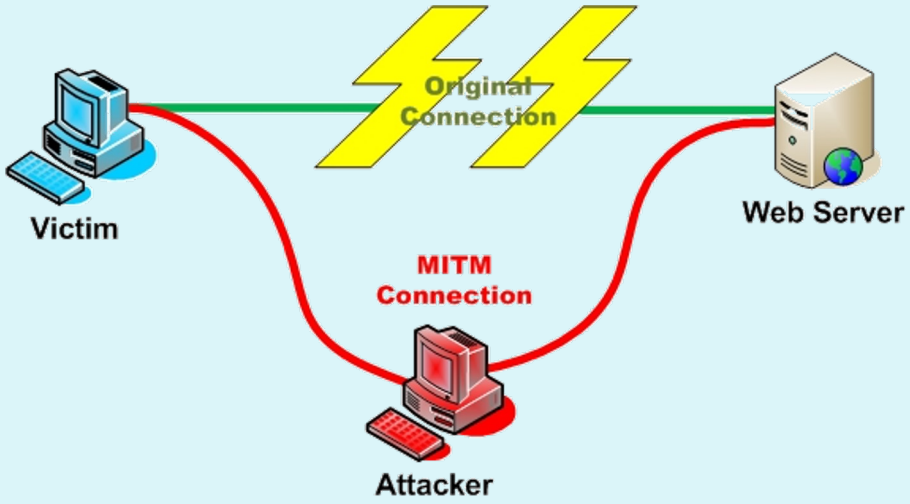
Kako se obraniti gornjih (in ostalih) napadov?

Pogosti napadi

7. **pregled vrat** *(port scan)*: napadalec testira, kateri strežniki so delujoči (npr. ping) in katere storitve ponujajo. Napadalec lahko pridobiva podatke o sistemu: DNS, storitve, operacijski sistemi)
8. **brskanje po smeteh** *(dumpster diving)*: način, s katerim lahko napadalci pridejo do informacij o sistemu (navodila za uporabo, seznami gesel, telefonskih številk, organizacija dela)
9. **matematični napadi** na šifrirne algoritme in ključe
10. **rojstnodnevni napad** *(birthday attack)*: je napad na zgoščevalne funkcije, za katere zahtevamo, da nobeni dve sporočili ne generirata iste zgoščene vrednosti. Pri slabših funkcijah napadalec išče sporočilo, ki bo dalo isto zgoščeno vrednost.

Pogosti napadi

11. **zadnja vrata** (*back door*): napadalec zaobide varnostne kontrole in dostopi do sistema preko druge poti
12. **ponarejanje IP naslovov** (*IP spoofing*): napadalec prepriča ciljni sistem, da je nekdo drug (poznan) s spreminjanjem paketov,
13. **prestrezanje komunikacije** (*man-in-the-middle*): napadalec prestreže komunikacijo in se obnaša, kot da je ciljni sistem (pri uporabi certifikatov lahko žrtvi napadalec podtakne svoj javni ključ)



Pogosti napadi

14. **ponovitev komunikacije** (*replay*): napadalec prestreže in shrani stara sporočila ter jih ponovno pošlje kasneje, predstavljajoč se kot eden izmed udeležencev
    - kako preprečimo napade s ponovitvijo komunikacije?
15. **ugrabitev TCP sej** (*TCP hijacking*): napadalec prekine komunikacjo med uporabnikoma in se vrine v mesto enega od njiju; drugi verjame, da še vedno komunicira s prvim
    - kaj napadalec pridobi s tem?
16. **napadi s fragmentacijo** (*fragmentation attack*): z razbijanjem paketa na fragmente razdelimo glavo paketa med fragmente tako, da jih požarna pregrada ne more filtrirati
    - tiny fragment attack: deli glavo prvega paketa
    - overlapping fragment attack: napačen offset prepiše prejšnje pakete

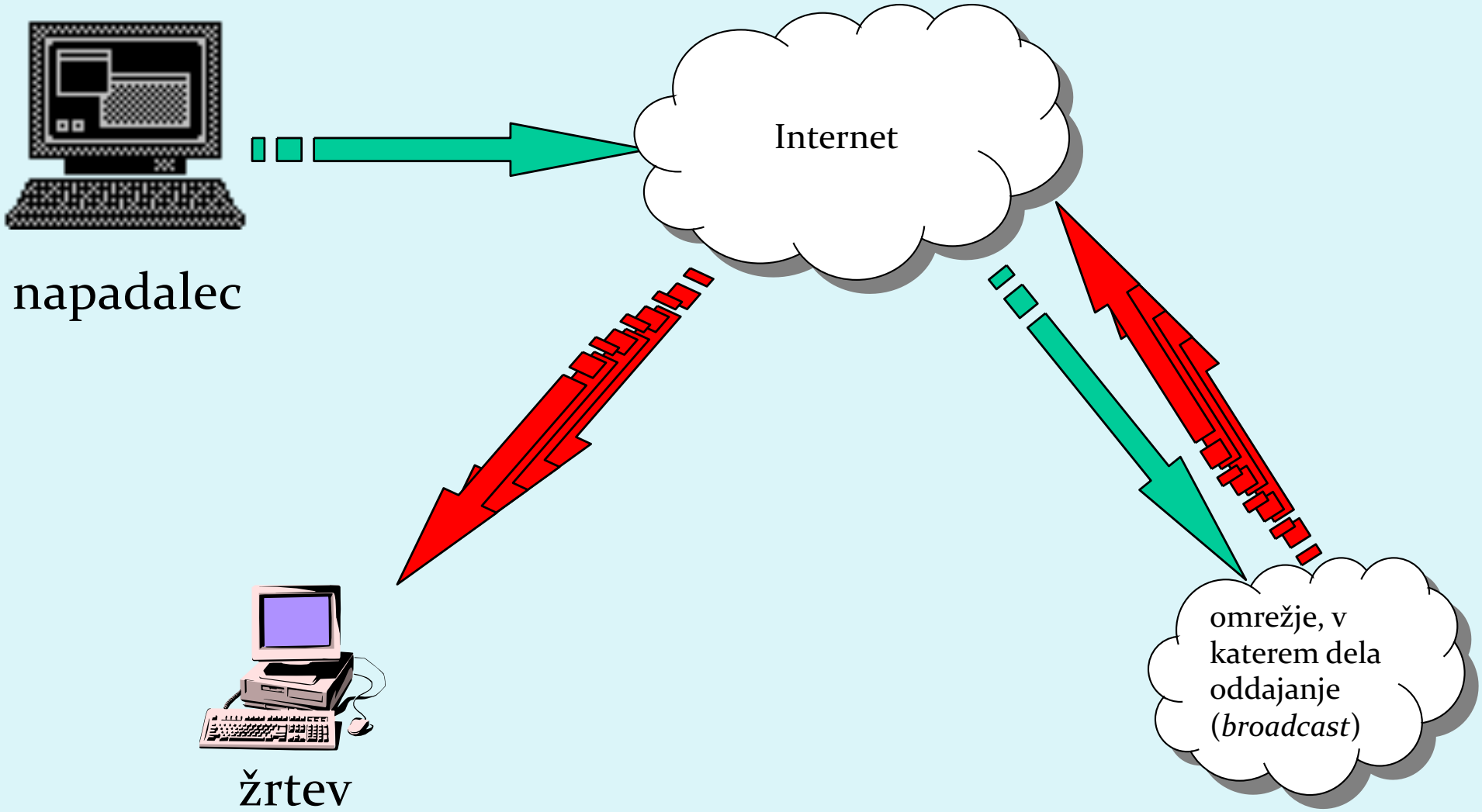
Pogosti napadi - DoS (1/5)

17. **preprečitev delovanja sistema** (*Denial-of-Service)*
    - Cilj napadalca: obremeni omrežne vire tako, da se nehajo odzivati zahtevam regularnih uporabnikov (npr. vzpostavitev velikega števila povezav, zasedanje diskovnih kapacitet, ...).
    - DDoS (*distributed*): DoS napad, ki ga povzroči napadalec z več omrežnih sistemov naenkrat.
    - Uporabniki porazdeljenih omrežnih sistemov lahko da ne vedo, da je napadalna oprema nameščena pri njih.

Pogosti napadi - DoS (2/5)

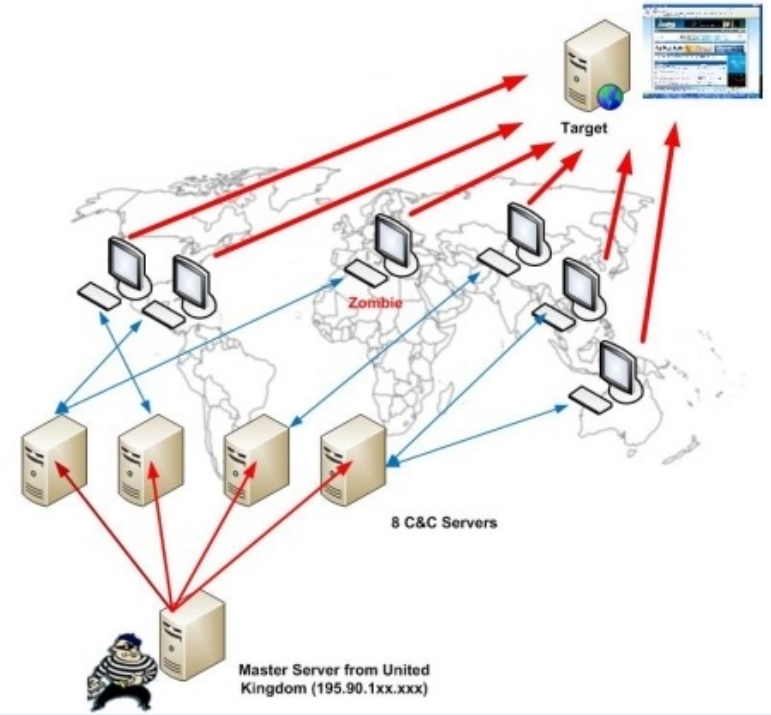
- Primeri:
  - **prekoračitev medpomnilnika** (*buffer overflow*): procesu pošljemo več podatkov, kot lahko sprejme (*Ping of death*: ICMP z več kot 65K podatkov je povzročil sesutje sistema);
  - **SYN napad**: napadalec pošlje veliko število zahtev za vzpostavitev povezave in se na odgovor sistema ne odzove; pride do preobremenitve vrste zahtev v sistemu
    - rešitev: omejitev števila odprtih povezav, timeout
  - **napad Teardrop**: napadalec spremeni podatke o številu in dolžini fragmentov v IP paketu, kar zmede prejemnika;
  - **napad Smurf** (naslednja prosojnica): uporaba posrednega oddajanja za preobremenitev sistema;

Pogosti napadi - napad DoS Smurf (3/5)



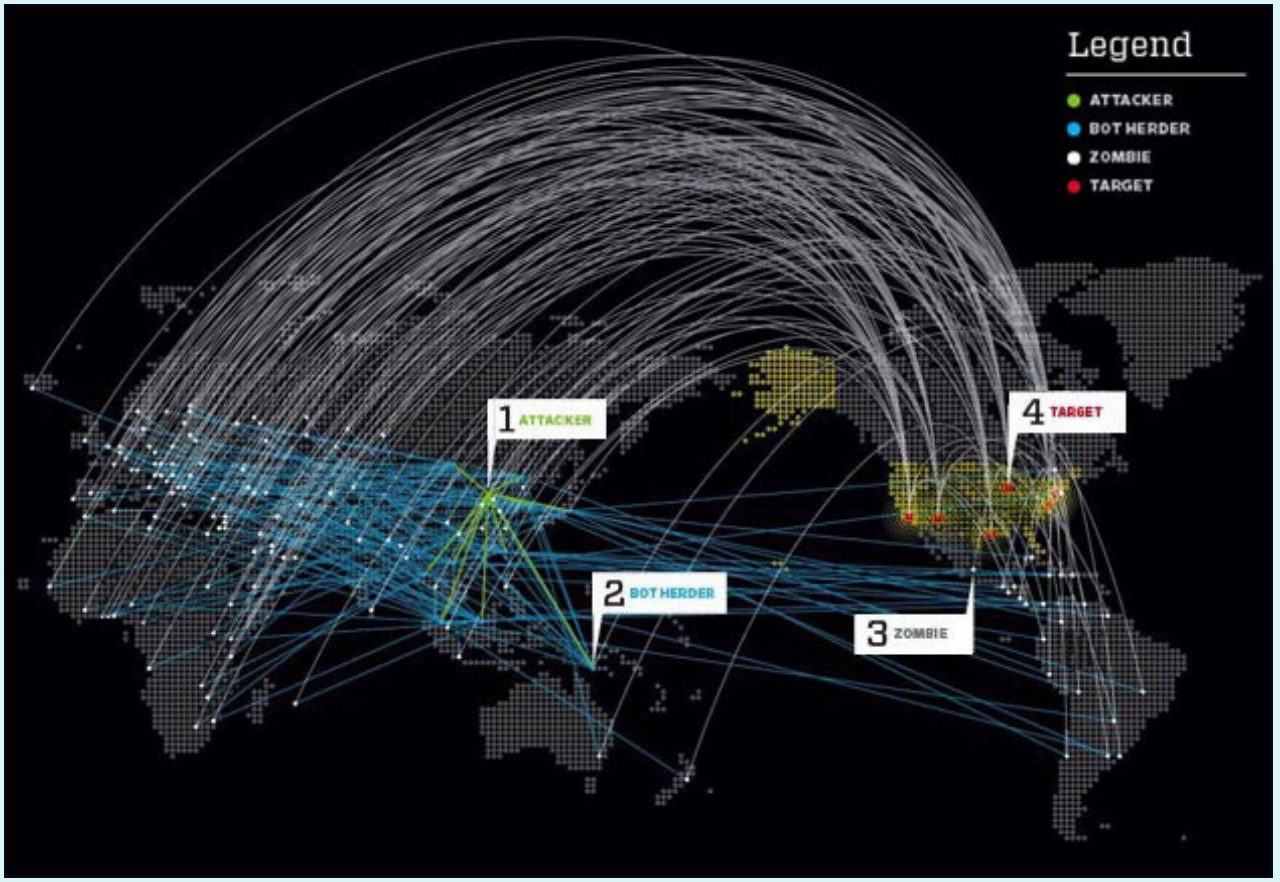
Pogosti napadi - DoS (4/5)

- Uporaba *bot*-ov (*web roBOT*) za organizacijo napadov na ciljni sistem
  - boti so lahko računalniki, okuženi s trojanskimi konji
  - njihovi uporabniki običajno ne vejo, da sodelujejo v napadu



Pogosti napadi - DoS (5/5)

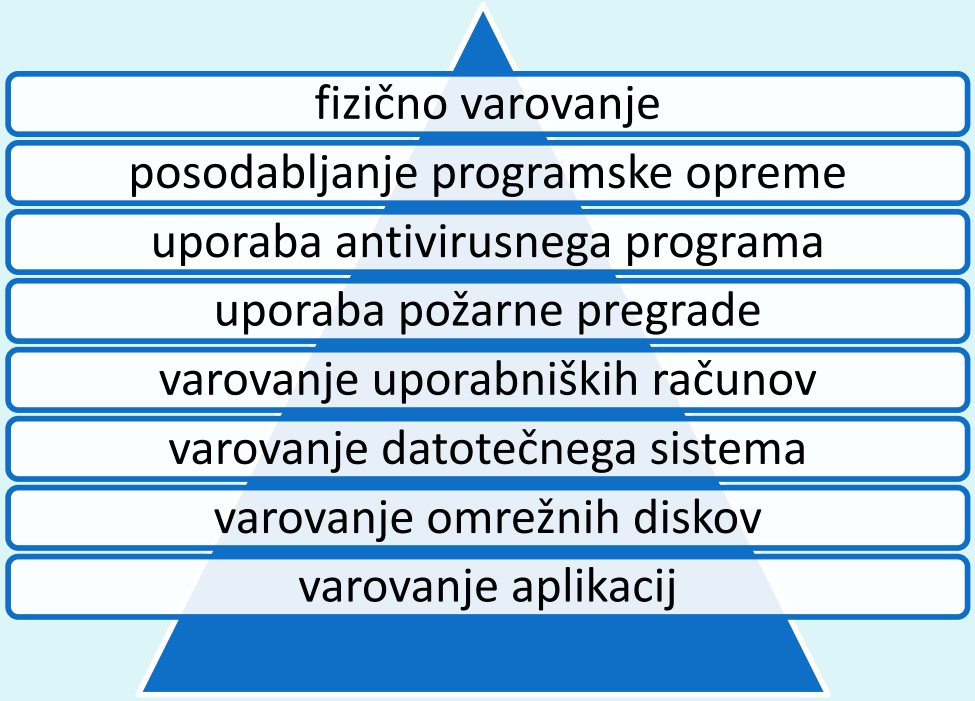
- odeleženci napada: **napadalec**, osrednji računalnik za **krmiljenje botov** (*herder*), **boti** (zombie), **cilj**



Obramba pred napadi

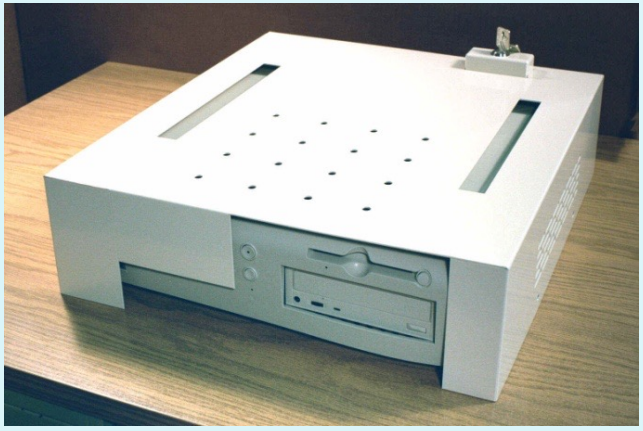
Tehnike obrambe

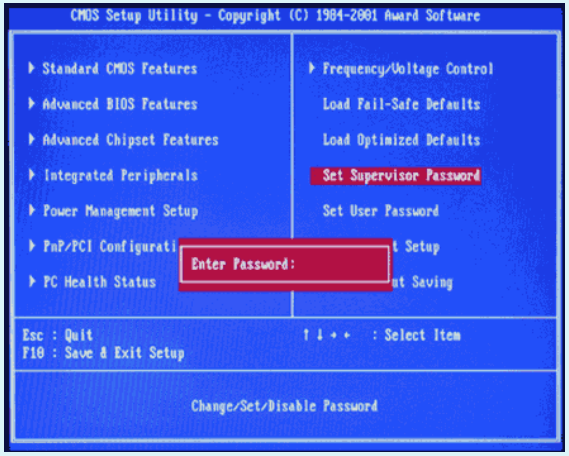
- V omrežju zadošča le en šibki člen - najšibkejši uporabnik, ki ogrozi omrežje. Administrator mora preprečiti prenos škodljivih programov na delovne postaje uporabnikov in zapreti varnostne luknje v infrastrukturi (konfiguracija):



Fizično varovanje sistema

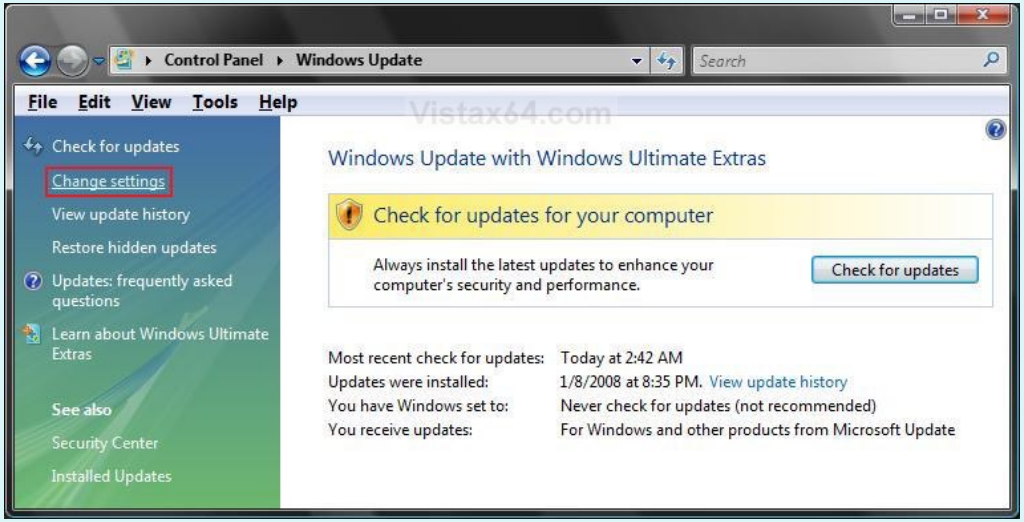
- Omejimo fizičen dostop do strežnikov in računalnikov
  - zaklepanje računalnikov
  - nastavi geslo za zagon (CMOS/BIOS)
  - nastavi geslo za dostop do BIOS nastavitev (varnost, zagon, ipd.)
  - onemogoči zagon sistema s pomnilniške palčke (ključka), CD – zunanjih medijev





Posodabljanje aplikacij

- Posodabljamo programsko opremo (krpanje, *patching*), s čimer proizvjalec omogoči popravljanje varnostnih lukenj
  - administrator potrebuje načrt testiranja, uvajanja in namestitve popravkov



Uporaba AV / požarne pregrade

- Uporaba antivirusnih programov
  - več možnosti: namestitev na odjemalcu/strežniku, avtomatsko posodabljanje, zaščita v realnem času.
    - Priporočeno: namestitev na odjemalcu, ker škodljiva oprema začenja delovati tam. AV na aplikacijskih prehodih ponavadi skrbijo za podmnožico protokolov na tisti lokaciji
  - posodabljanje (posamezno ali centralizirano)

- Uporaba požarne pregrade
  - v omrežju / osebna požarna pregrada

Varovanje uporabniških računov

- Napadalci iščejo neuporabljane, neaktivne, nezaščitene račune za dostop do sistema:
  - preimenovanje uporabniških imena administratorja (*superuser*, *root*, *administrator*),
  - omejitev števila računov z visokimi pravicami (ločeni admin računi, pogoste menjave gesel),
  - onemogočenje uporabe starih računov,
  - uporaba zahtevnih gesla

Varovanje datotečnega/omrežnega sistema

- Zaščita datotečni sistem
  - za dostop do datotečnega sistema dodeli uporabnikom najmanjše potrebne pravice
  - odstranitev nepotrebne aplikacije
  - zaščita zagonska področja. Primer - 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

Varovanje aplikacij

- pravilna nastavitev aplikacij (privzete vrednosti niso vedno najbolj varne!)
- odstranitev odvečnih aplikacij
- onemogočanje priponk v e-pošti
- onemogočanje izvajanje nevarnih tipov datotek
- nameščanje aplikacij na nestandarna vrata in v nestandardne mape
- ...

Naslednjič gremo naprej!

- varnost:
  - varna omrežna infrastruktura
  - podatki za delovanje omrežja