Komunikacijski protokoli in omrežna varnost

Varnostni elementi: IPsec, SSL in infrastruktura

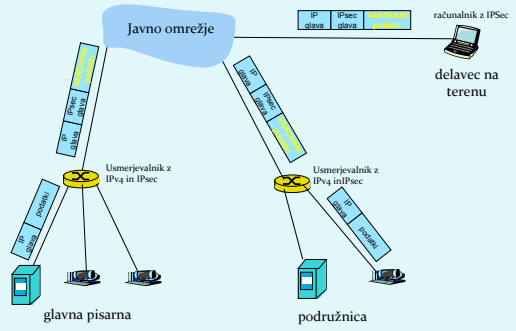
IPSec

- 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 avtentikacije 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 (VNP) 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

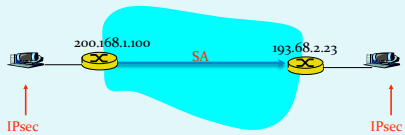


Implementacija IPsec

- mehanizem IPSec ponuja dva protokola varovanja:
  - AH - *Authentication Header*
    - zagotavlja avtentikacijo izvora in integriteto podatkov
  - *ESP - Encapsulation Security Payload*
    - zagotavlja avtentikacijo izvora, integriteto 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 enkripcije (npr. 3DES) in ključ
  - vrsta preverjanja integritete (npr. HMAC/MD5)
  - ključ za avtentikacijo

2 načina komunikacije

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

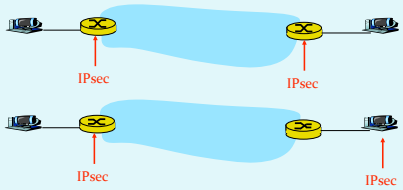
Transport mode z AH	Transport mode z ESP
Tunnel mode z AH	Tunnel mode z ESP

Najbolj pogosto!

IPsec Transport Mode

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

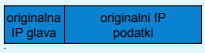
IPsec – tunneling mode



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

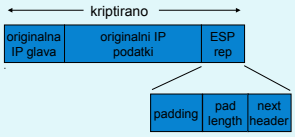
IPsec datagram: tunnel mode in ESP

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



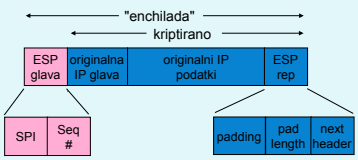
IPsec datagram: tunnel mode in ESP

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



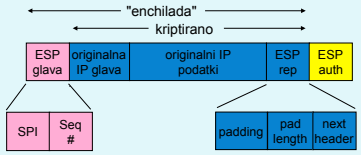
IPsec datagram: tunnel mode in ESP

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



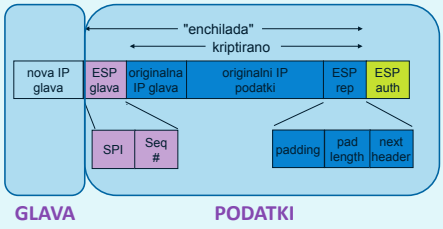
IPsec datagram: tunnel mode in ESP

- doda se polje ESP auth, ki je izračunana zgoščena vrednost cele "enchilade". Algoritem in ključ določa SA.



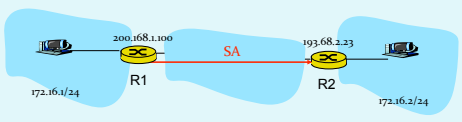
IPsec datagram: tunnel mode in ESP

- 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 enchilade, preveri Seq#, odkodira enchilado, odstrani zapolnitev, ekstrahira 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 Janez naš man-in-the-middle med R1 in R2. Janez 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
- 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: PSK and PKI

- Authentication (proof who you are) with either
  - pre-shared secret (PSK) or
  - with PKI (pubic/private keys and certificates).
- With PSK, both sides start with secret:
  - then run IKE to authenticate each other and to generate IPsec SAs (one in each direction), including encryption and authentication keys
- With PKI, both sides start with public/private key pair and certificate.
  - run IKE to authenticate each other and obtain IPsec SAs (one in each direction).
  - Similar with handshake in SSL.

IKE ima 2 fazi

- IKE uporablja PKI ali PSK (pre-shared key) za avtentikacijo odjemalcev med seboj. Ima dve fazi:
  - Faza 1: Vzpostavi dvosmeren IKE SA
    - 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 (avtentikacija 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:
    - edini način: Quick Mode

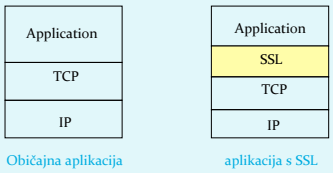
SSL

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 nakupov letno
- Razvil ga je Netscape leta 1993
- Več vrst
  - TLS: transport layer security, RFC 2246
- Zagotavlja zaupnost, integriteto, avtentikacijo
- Cilji pri razvoju:
  - uporaba pri spletnih transakcijah
  - zakrivanje podatkov (še posebej številk kreditnih kartic)
  - avtentikacija spletnih strežnikov
  - možnost avtentikacije klienta
  - čim manjši napor pri opravljanju nakupa pri drugem prodajalcu

SSL and TCP/IP

- Dostopen vsem TCP aplikacijam preko aplikacijskega vmesnika SSL



23

Zasnova SSL

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

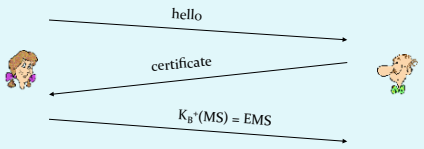
- želimo pošiljati tokove BYTEOV in interaktivne podatke, ne statična sporočila,
- za eno povezavo želimo imeti MNOŽICO ključev, ki se spreminjajo,
- kljub temu želimo uporabljati certifikate (ideja: uporabimo jih pri rokovanju)

Poenostavljeni SSL

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

- 1. ROKOVANJE: Ana in Brane uporabita certifikate, da se avtenticirata eden drugemu in izmenjata ključ
- 2. IZPELJAVA KLJUČA: Ana in Brane uporabita izmenjani 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 = kriptirani glavni ključ (encrypted master secret)
- K_B^+ - javni ključ prejemnika B

Poenostavljeni SSL: Izpeljava ključa

- Slaba praksa je uporabljati isti ključ za več kriptografskih 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 klienta strežniku
  - M_c = ključ za zgoščanje podatkov, poslanih od klienta strežniku
  - K_s = ključ za zakrivanje podatkov, poslanih od strežnika klientu
  - M_s = ključ za zgoščanje podatkov, poslanih od strežnika klientu
- Ključi se izpeljejo z uporabo posebne funkcije. Ta uporablja glavni ključ (Master Secret) in dodatne (naključne) podatke za generiranje naslednjih ključev

27

Poenostavljeni SSL: Pošiljanje podatkov

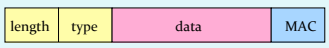
- Kako preveriti integriteto podatkov?
  - če bi pošijali po zlogih (byteih), kam bi pripeli MAC (zgoščeno vrednost sporočila)?
  - Tudi če MAC pošljemo po zaključku celega prenosa (vseh zlogov), nimamo vmesnega preverjanja integritete!
- REŠITEV: Tok podatkov razbijemo v ZAPISE
  - vsakemu zapisu pripnemo MAC
  - prejemnik lahko reagira na (ne)veljavnost integritete posameznega zapisa

Poenostavljeni SSL: Pošiljanje podatkov

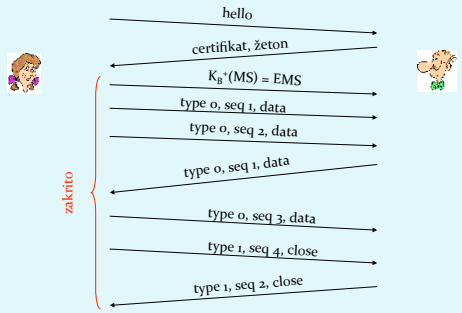
- Problem 1: številka paketa se nahaja nekriptirana v glavi TCP. Kaj lahko naredi napadalec?
  - napadalec lahko zajame in ponovi komunikacijo?
  - preštevilči vrstni red paketov?
  - prestreže in odstrani paket?
- REŠITEV: pri računanju MAC upoštevaj številko paketa
  - MAC = MAC(ključ M_x, zaporedna_številka || 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



Pravi SSL: podrobnosti

- Kakšne so dolžine polj v protokolu?
- Kateri protokoli za zakrivanje naj se uporabijo? Dogovor o uporabi protokola:
  - Želimo, da klient in strežnik lahko izbirata in se dogovarjata o kriptografskih algoritmih (angl. *negotiation*, klient 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 kriptografijo
    - RSA

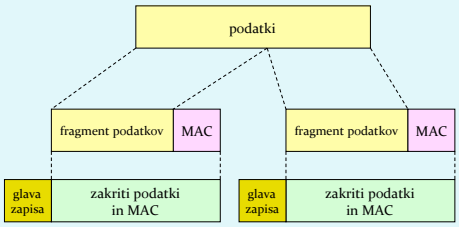
Pravi SSL: Rokovanje

- Poenostavljeni SSL: hello->, <-certifikat, kriptiran MS->
- Pravi SSL dejansko izvaja: avtentikacijo strežnika, izbiro algoritmov, določanje ključev, avtentikacijo klienta (opcijsko)
- Postopek:
  1. Klient pošlje seznam podprtih algoritmov + žeton
  2. Strežnik izbere algoritem s seznama, vrne izbiro, certifikat in svoj žeton
  3. Klient preveri certifikat, generira PMS, z javnim ključem strežnika ga kriptira in pošlje strežniku
  4. Klient in strežnik neodvisno izračunata enkripcijske in MAC ključe iz PMS in žetonov.
  5. Klient pošlje MAC od vseh sporočil v rokovanju.
  6. Strežnik pošlje MAC vseh sporočil v rokovanju.

Pravi SSL: Rokovanje

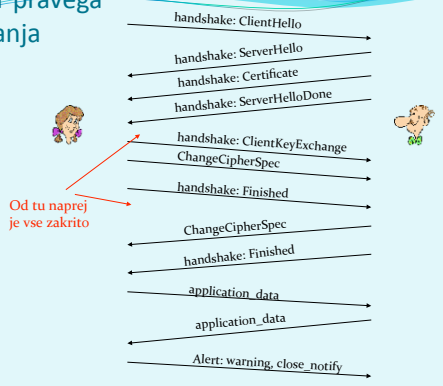
1. Zakaj izmenjava MAC v korakih 5 in 6?
   - Klient običajno ponudi več algoritmov, nekateri so šibki, drugi močnejši. Napadalec bi lahko izbrisal iz ponudbe močnejše algoritme.
   - Zadnji dve sporočilo zagotavljata integriteto vseh prenešenih sporočil in preprečita tak napad
2. Zakaj uporaba žetonov?
   - Denimo, da Zelda posluša sporočila med Ano in Branetom ter jih shrani. Naslednji dan pošlje Zelda Bobu 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 Zelda 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



SSL: izpeljava ključev

- Žetona klienta 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 klienta
  - MAC ključ strežnika
  - enkripcijski ključ klienta
  - enkripcijski ključ strežnika

  } enako kot pri poenostavljenem SSL!
  - inicializacijski vektor (IV) klienta
  - inicializacijski vektor (IV) strežnika

  } KAJ JE TOLE?

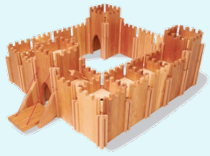
potrebna sta, kadar uporabljamo simetričen algoritem z bločno kriptografijo (3DES ali AES), ki potrebujeta inicializacijo!

Operativna varnost:

požarni zidovi in sistemi za zaznavanje vdorov

Varnost v omrežju

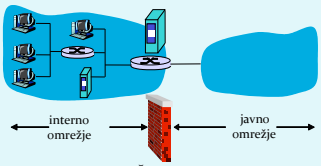
- Administrator omrežja lahko uporabnike deli na:
  - good guys: uporabniki, ki legitimno uporabljajo vire omrežja, pripadajo organizaciji,
  - bad guys: vsi ostali, njihove dostope moramo skrbno nadzorovati
- Omrežje ima običajno eno samo točko vstopa, kontroliramo dostope v njej:
  - požarni zid (firewall)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



39

Požarni zid

izolira interno omrežje od velikega javnega omrežja, določenim paketom dovoli prehod, druge blokira. Ima 3 naloge:
- filtrira VES promet,
- prepušča samo promet, ki je DOPUSTEN glede na politiko,
- je IMUN na napade



POŽARNI ZID

Požarni zid: vrste filtriranj

1. izolirano filtriranje paketov (angl. *stateless, traditional*)
2. filtriranje paketov v kontekstu (angl. *stateful filter*)
3. aplikacijski prehodi (angl. *application gateways*)

Izolirano 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)

Izolirano filtriranje paketov: primeri

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

Izolirano filtriranje paketov: primeri

Želimo doseči:	Nastavitev požarnega zidu
Onemogočen dostop navzven do poljubnega spletnega strežnika.	Zavrzi vse pakete, naslovljene na poljuben IP naslov in na vrata 80
Onemogočene 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 broadcasta za preobremenitev storitev).	Zavrzi vse ICMP pakete, naslovljene na broadcast naslov omrežja (npr. 130.207.255.255).
Preprečiti analizo omrežja s *traceroute*	Zavrzi vse odhodne pakete ICMP s sporočilom "TTL expired"

Izolirano filtriranje: Dostopovni seznami

- dostopovni seznam (angl. ACL, access control list)
- tabela pravil, upošteva se jo od zgoraj do spodaj.
- 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

Filtriranje paketov v kontekstu

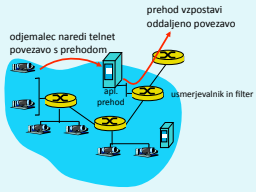
- angl. *stateful filter*, upošteva povezavo
  - izolirano filtriranje lahko dovoli vstop nesmiselnim paketom (npr. vrata = 80, ACK =1; čeprav notranji odjemalec ni vzpostavil povezave) :
- IZBOLJŠAVA: **filtriranje paketov v kontekstu** spremlja in vodi evidenco o vsaki vzpostavljeni TCP povezavi
  - 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*)

Filtriranje paketov v kontekstu

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,
- kliente je potrebno nastaviti, da se znajo povezati s prehodom (npr. IP naslov medstrežnika v brskalniku)

Naslednjič gremo naprej!

- varnost:
  - sistemi za zaznavanje vdorov (IDS)
  - varna omrežna infrastruktura
  - napadi in varnostne grožnje

50