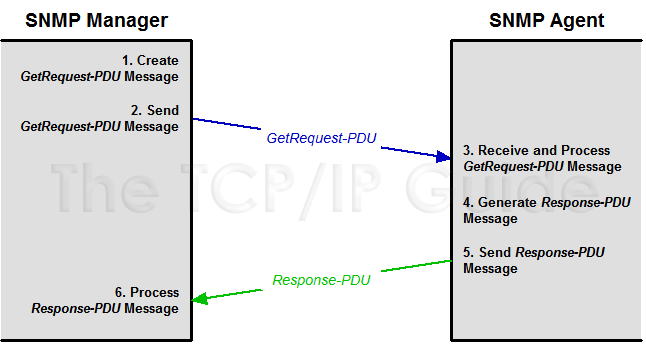
**1. naloga: Naš prijatelj Peter Zmeda je malce preslišal, da za prenos sporočil po omrežju bootp protokol uporablja UDP prenosni protokol, ter se je lotil implementacije s TCP protokolom.  
  
*VPRAŠANJA:  
1. Kje vse mora popraviti programsko opremo (na katerih računalnikih), da bo svojo novo idejo spravil v delovanje?***  
Na vseh računalnikih torej na strežniku ter klientih. Če bi samo na strežniku potem bi strežnik poslal tcp paket nazaj klient pa pozna samo udp velja tudi obratno(strežnik ne bi poznal tcp).  
***2. Ali bo njegova ideja sploh delovala? Če menite da bo, ali bo kaj izboljšala kakovost prenosa?Utemelji***  
Broadcast se naredi na 2. plasti torej bi bila implementacija možna. Na tretji plasti torej dobimo potem IP ki ga nam dodeli strežnik. Višja plast torej transportna pa je v tem primeru TCP. Tukaj na tej plasti bi potem tcp skrbel za zagotovljeno dostavo paketov, tako da samem protokolu bootp ni potrebno. Je pa težja implementacija ter integracija, UDP protokol je preprostejši. Transportna plast skrbi še za kontrolo pretoka. Kakovost pretoka bi bila boljša.

***3. Eden od razlogov, čemu se je Peter odločil implementirati svojo inačico bootp oziroma DHCP protokola, je bil ta, da je ob zagonu ˇzelel poslati odjemalcu še posebno sporošilo o vremenskih podatkih (Bog si ga vedi čemu), ki ga naj bi odjemalec shranil. Kot vemo, ta razlog ni dovoljšen, da bi se lotili svoje implemetacije protokola, saj strežniki že sedaj omogočajo pošiljanje takih podatkov. Kako?***  
Uporabimo vend polje v bootp protokolu, iz tega je nastal tudi dhcp pri čemer se je polje vend spremenilo v options polje. Torej v polju vend; možne razširitve dodamo našo informacijo o vremenu. Dodati moram še opis kako to dodamo.  
  
**2. naloga: V nekem podjetju že dlje časa uporabljajo orodja za nadzor in upravljanjez omrežji. Odgovori na naslednja vprašanja v zvezi s tem:  
  
*1. Nadzor in upravljanje sta vzpostavljena le do te mere, da lahko administratorji konfigurirajo nastavitve naprav, spremljajo varnostne mehanizme in sledijo dostopom drugih uporabnikov. Kateri aspekt upravljanja v tem sistemu ni implementirana - poimenujte in obrazložite, kaj omogoča.***  
Upravljanje z napakami. Gre zato, da tudi naprava obvesti o napaki delovanja recimo trap način ali pa da vprašamo napravo zahteva/odgovor.  
  
2. Dopolnite: Upravljalec omrežja komunicira s programsko opremo v nadzorovanih napravah imenovano **agent**z uporabo **protokola za upravljanje(lahko SNMP)**. Slednje ponujajo strukturiran dostop do svojih podatkov v obliki **MIB**, ki je definiran na osnovi jezika za zapis teh podatkov, imenovanega **SMI**. // MIB, SMI je jezik za opis objektov, parametrov MIB, vsak upravljalec pa ima svoj MDB(zbirka naprav ki jih upravlja konkretne vrednosti za MIB-e)  
  
**3. Protokol SNMP uporablja dve obliki sporočil. Za vse štiri oblike/vrste upravljanja in nadzorovanja (torej za 2×4 = 8 primerov) podaj možen primer uporabe takega sporočila.**  
SNMP uporablja dva tipa sporočil:  
1. zahteva/odgovor  
**GetRequest**[?](http://ucilnica1011.fri.uni-lj.si/mod/wiki/view.php?id=4581&page=GetRequest)-PDU, **GetNextRequest**[?](http://ucilnica1011.fri.uni-lj.si/mod/wiki/view.php?id=4581&page=GetNextRequest)-PDU, Response-PDU, **SetRequest**[?](http://ucilnica1011.fri.uni-lj.si/mod/wiki/view.php?id=4581&page=SetRequest)-PDU  
2. obvestilo  
**GetBulkRequest**[?](http://ucilnica1011.fri.uni-lj.si/mod/wiki/view.php?id=4581&page=GetBulkRequest)-PDU, **InformRequest**[?](http://ucilnica1011.fri.uni-lj.si/mod/wiki/view.php?id=4581&page=InformRequest)-PDU, Trapv2-PDU, Obsolete-snmpv1-request  
  
PDU Type value definira ali gre za sporočilo tipa 1 ali 2.  
Vrednost, ki definira tip sporočila. Vrednost 4/7 pomeni obvestilo (trap message).  
Primer:  
  
Primeri:   
Upravljanje z napakami:   
- zahteva-odgovor: Upravljalec omrežja zahteva poročilo o napaki, ki se je zgodila.

- obvestilo: Naprava obvesti upravljalca, da je linija preobremenjena.  
Upravljanje z beleženjem:  
- zahteva-odgovor: Upravljalec omrežja želi beležiti število neuspešno prijavljenih v sistem.  
- obvestilo: Naprava obvesti upravljalca omrežja, kdo izmed prijavljenih se je neuspešno prijavil.  
Upravljanje s konfiguracijami:  
- zahteva-odgovor: Upravljalec omrežja želi nastaviti novo konfiguracijo za uporabnika npr: Lojz.  
- odgovor: Naprava odgovori administratorju o uspešni spremembi konfiguracije.  
Upravljanje z varnostjo:  
- zahteva-odgovor: Upravljalec omrežja želi izvedeti kako je nastavljen požarni zid.  
- odgovor: Naprava upravljalcu omrežja odgovori s konkretno konfiguracijo požarnega zidu.  
  
**5. S principom kodiranja TLV želimo prejemniku poslati en sam celoštevilski  
podatek (tip=2) z vrednostjo 32. Zapišite zakodiran podatek.**  
TLV - type, length, value (tip, dolžina, vrednost)  
  
Recimo integer je za vrednost 259 dolg 2 bajta(za manjše vrednosti 256 je dolžina 1 bajt). Torej vrednost 259 zapišemo kot 31. 1 pomeni za prvi bajt, 3 za drugi bajt podatkov.  
1 torej 256, 3 pa 3 = 259  
  
celoštevilski podatek = tip = 2  
vrednost = 32 zapišemo z enim bajtom torej 20 šestnajstiško.  
dolžina = 1 bajt

|  |  |  |
| --- | --- | --- |
| value | length | type |
| 20 | 1 | 2 |

**3. naloga: Peter ni od muh, saj sledi tudi znanstvenim dosežkom po svetu. Tokrat je prebral, da so v pospeševalniku v CERNu uspeli izolirati za delček sekunde nekaj anti-protonov. Pri tem so uporabili napravo, ki se imenuje LHC (The Large Hadron Collider). Delovanje naprave med delovanjem opazuje vec detektorjev, ki odčitujejo podatke v stvarnem času in jih fiziki kasneje obdelujejo. Precej programske opreme za nadzor in delovanje detektorjev je ustvarilo slovensko podjetje CosyLab**[**?**](http://ucilnica1011.fri.uni-lj.si/mod/wiki/view.php?id=4581&page=CosyLab)**.**  
  
**1. Za potrebe te naloge predpostavimo, da imamo 1024 senzorjev, s katerih bi rad Peter prejemal k sebi v Spodnje Butale v stvarnem casu. Zato se je za prenos odlocil za RTP protokol. Kako naj Peter v toku podatkov razlikuje s katerega senzorja prihajajo podatki? Opišite tudi s konkretnim primerom RTP paketa.**  
SSRC – identifikator vira(Synchronization source)  
Vsi up. v RTP seji potrebujejo svoj SSRC vrednost, ker jih prejemnik potrebuje da razloči vire.  
Vrednost SSRC izbrana naključno, čeprav je še vedno potrebno rešiti če imata dva vira isti ssrc.  
CC – število mešanih virov, CC je v našem primeru 1024 **2. Zahteve za protokol stvarnega časa smo razdelili na dva dela ter za enega zadolžili sam protokol ter za drugega aplikacijo. Za kateri del zahtev smo zadolžili protokol?**  
Protokol naj poskrbi za časovno izravnano(skrbi za pravo zaporedje paketov). Za to poskrbi s časovnimi značkami in pa zaporednim številčenjem paketov.  
 **3. Kateri je drugi del in čemu smo za zanj zadolžili aplikacijo?**  
Aplikacija naj poskrbi za izgubljene pakete. Aplikacija je bolj fleksibilna, lažje nadzoruje časovni okvir, želimo da ima aplikacija nadzor(če nam nekaj paketov manjka ne opazimo skoraj nič).  
  
Če bi RTP skrbel za izgubljene pakete(na prenosni uporablja UDP) bi to zelo povečalo promet v omrežju. Na začetku smo tudi predpostavili da bi recimo namesto UDP dali TCP:  
  
TCP rešuje izgubljene pakete, samo to ni v redu, ker "cuka".  
TCP sam skrbi, da se paketi pošljejo na novo:  
- potrdi za pakete ali so uspešno prišli skozi  
- potreben je 3x čas --> 1. ko so paketi izgubljeni, 2. opozori nazaj, da je bil izgubljen, 3. ponovno pošiljanje  
- nekateri paketi bodo zamudili  
  
Aplikacija tako kot sem že povedal je fleksibilna, naj se ona odloči ali je slika, zvok, ... dovolj kvalitetna ali moramo recimo prenesti še dodatnih 50 paketov da bo slika, zvok, ... dovolj kakovosten. O obveščanju kvalitete nas obvešča RTCP(RTP Control Protocol)  
  
Opravlja štiri funkcije:  
1. sporoča o kakovosti prenašanega prometa (RR:receiver report in SR: sender report  
2. dodaten opis vira toka dogodkov (SDES: Source description items)  
3. skrbi za pravilno gostoto pošiljanja sporočil o kakovosti prenosa  
4. prenaša lahko še dodatne podatke za potrebe aplikacije (APP: Application-specific functions)  
  
**4. naloga: V omrežju uporabljamo razpošiljanje s protokoloma IGMP in PIM.  
1. Kje v omrežju se uporablja kateri izmed protokolov in kakšen je njegov namen?**  
IGMP pomeni Internet group management protocol. Uporablja se za upravljanje s skupinami. Klient se naroči, da bi tudi on dobival vse promet te skupine.  
IGMP skrbi za upravljanje s tem, kdo so prejemniki razpošiljanih sporočil. Omogoča:  
- pridružitev skupini  
- izstop iz skupine  
- zaznavanje drugih vmesnikov v skupini  
PIM (Protocol Independent Multicast) je usmerjevalni algoritem, ki skrbi za komunikacijo med usmerjevalniki za razpošiljanje. Namenjen je usmerjanju razpošiljevalnih paketov in gradnji razpošiljevalnega drevesa.

**Predavanja:**

**Kaj  je  protokol?**

Je  nek  dogovor (če gre za standard je ta dogovor formalen) med dvema strankama, na kak način se sporazumevata.

**Ponovitev osnov komunikacij**

ISO/OSI model

APLIKACIJSKA PLAST (podatki)

PREDSTAVITVENA PLAST (podatki)

SEJNA PLAST (podatki)

TRANSPORTNA PLAST (segmenti, end to end...)

OMREŽNA PLAST (paketki, IP...)

POVEZAVNA PLAST (okvirji, MAC naslovi, LLC)

FIZIČNA PLAST (biti, električni signali...)

Plast N nudi storitve plasti N+1(zgornji plasti). Plast N zahteva storitve plasti N-1(spodnje plasti). Protokol: pravila komuniciranja med istoležnima procesoma. Entitetni par predstavlja par procesov, ki komunicira na isti plasti (povezavna plast na eni in povezavna plast na drugi strani je en entitetni par).

TCP: trosmerno rokovanje, SYN, SYNACK, ACK; potrjevanje paketov, velikost okna, nadzor zamašitev- vse o je v TCP glavi.

OMREŽNA PLAST:

Na omrežni plasti je običajen tudi IPv4 protokol, ki ima v glavi cel kup podatkov, ki jih uporablja za svoje lastno delovanje. Najpomembnejša podatka sta naslov prejemnika in naslov pošijatelja. Oba naslova sta 32 bitna, naslove lahko skupaj združejemo v skupino po 8 bitov in tako dobimo zapis s pikami (xxx.xxx.xxx.xxx). Vsi elementi, ki so v omrežju ne vejo ničesar o drugih elementih omrežja. Ker, če bi morali imeti, bi si morali zapomniti zelo veliko stvari, kar bi privedlo do preobremenitve elementov in omrežja samega – velja za usmerjevalnike. Pri stikalu si pa ti elementi zapomnijo te podatke, ker je ponavadi omrežje majhno.

IPv6

Je bolj zanesljiv in uporablja varnostne elemente. Edina stvar v primerjavi z IPv4 je ta, da ima IPv6 daljše naslove --> 128 bitne, zapišemo jih v šestnajstiškem zapisu. Sočasno morata obstajati verzija 4 in verzija 6. Ne moremo ene kar ukiniti in začeti uporabljati drugo. Zaradi sočasnosti imamo dva principa. Prvi princip je v tem da imamo sklade (Dual-stack), ki hkrati govorijo oba protokola in znajo med njima preslikovati. Drugi princip pa je v tem, da v enem protokolu vstavimo drugi protocol (ga enkapsuliramo). Dejansko obstaja protokol, ki definira kako podatke iz IPv4 skozi IPv6 spravimo do naročnika, ki uporablja IPv4.

Omrežna varnost

Pomeni, da je uporabnik interneta varen. Internet ni bil snovan ozirajoč se na varnost!

Je področje ki:

Analizira možnosti vdorov v sisteme, Načrtuje tehnike in obrambe pred napadi, Snuje varne arhitekture, ki so odporne pred vdori. Veliko varnostnih groženj se širi preko interneta. Najbolj razširjene so:

Virusi, črvi, trojanski konji, Spyware in adware, Zero-day napadi (Varnostne luknje v programski opremi, za katere razvijalec še ni imel možnosti jih odpraviti), Hekerski napadi, Denial of service napadi, Prestrezanje podatkov in kraja, Kraja identitete. Elementi, ki jih moramo zagotoviti za varno povezavo:

Zaupnost-kdo sme prebrati sporočilo? (enkripcija)

Avtentikacija-dokaz, da si res ti (identifikacija–povej, kdo si, brez dokaza)

Razpoložljivost in nadzor dostopa-preprečevanje nelegitimne rabe virov (avtorizacija–ugotavljanje, ali nekaj smeš storiti)

Integriteta sporočila-je bilo sporočilo med prenosom spremenjeno?

Preprečevanje zanikanja (nonrepudiation)–res si poslal/res si prejel

Komponente omrežne varnosti so tipično:

Anti-virus anti-spyware programska oprema

Požarni zid, ki preprečuje nepooblaščen dostop do omrežja

Intrusion prevention systems (IPS), spremljajo, zaznavajo, beležijo dogajanje na omrežju.

Virtual Private Networks (VPNs), za zaščito oddaljenih povezav

**Priklop in zagon naprave**

**Zagon računalnika**

* CPE ob priklopu na napajanje nastavi vrednost programskega števca (program counter) na točno določeno vrednost nato pa začne izvajati ukaze.
* Ko CPE začne z delom se v pomnilniku nahaja inicializacija vseh naprav.

***BIOS:***

* Služi kot vmesnik med strojno opremo in operacijskim sistemom.
* Sestavljen iz:
  + kode, ki se prične izvajati ob zagonu (naloži operacijski sistem) -> koda izkorsiti gonilnike za dostop do zunanjih enot
  + gonilnikov za V/I enote
* BIOS deluje tudi pri izklopljenem računalniku (ura, datum in pomnjenje nastavitev).

***Operacijski sistem - klasično:***

* Vmesnik med uporabniškimi programi in strojno opremo.
* Skrbi za upravljanje z viri (V/I enote, ...).
* Na začetku je za V/I enote uporabljal BIOS gonilnike. Ker niso bili učinkoviti jih nadomestijo gonilniki v operacijskem sistemu.

***Nalaganje OS - sodobno:***

* BIOS naloži nek program, ki ga nato prične izvajati.
* Program najde na prvem bloku V/I enote (ponavadi disk) -> master boot record (MBR).
* Progam, ki ga naloži BIOS ni nujno operacijski sistem.

***Zagon preko omrežja -bootp******Nalaganje OS - drugače:***

* Namesto z diska BIOS naloži program s strežnika.
* Potrebujemo protokol !!!
* Rokovalnik protokola mora biti učinkovit in jedernat.

Za uspešno nalaganje operacijskega sistema mora računalnik:

1. znati poiskati strežnik, s katerega bo naložil OS
2. zna se predstaviti, kot bo svetoval/zahteval strežnik
3. prenesti OS k sebi
4. namestiti OS in ga zagnati (enak korak kot pri nalaganju z diska)

**Prednosti:**

* ne potrebujemo diska, DVD-ja, OS preprosto zamenjamo vsem računalnikom hkrati (samo na strežiniku).

**Slabosti:**

* Ranljivost,nPočasnost (če se hkrati priključi veliko uporabnikov)

***DNS:***

* Deluje na aplikacijski plasti
* Uporablja vrata 53.
* Uporablja UDP pakete (v glavi paketka je paket označen s številko 17).
* Preslikuje med črkovnim nizom in številko (www.google.com -> 74.125.39.104).
* Če želimo lahko uporabimo preslikovno tabelo (ponavadi v /etc/hosts) -> pri velikem številu računalnikov je postopek zamuden.

***IANA (The internet assigned numbers authority)****-> poznana tudi pod ICANN (Internet Corporation for Assigned Names and Numbers):*

* Inštitucija, ki nadzira globalno dodeljevanje IP številk, DNS naslovov, portov, ...
* IANA upravlja številne parametre IETF protokolov.
* Upravlja podatke na najvišjem nivoju DNS stežnikov,...

***BOOTP (RFC 951):***

* Preprost protokol, pri katerem konfiguracijski strežnik klientu pošlje IP naslov operacijskega sistema, ki je namenjen klientu.
* Koračni pogovor med odjemalcem in strežnikom (vprašanje, odgovor).
* Uporablja UDP protocol, Podpira samo IPv4.
* Hkrati je lahko prisotnih več strežnikov in več odjemalcev, ki želijo naložiti OS.

***TFTP (RFC 1350)****:*

* Poenostavljen protokol za prenos podatkov (izvira iz FTP).
* Uporablja UDP pakete.
* Ni avtentikacije ter kriptiranja.
* Uporablja zalo malo pomnilnika.
* Odjemalec na začetku pozna IP naslov strežnika (dobi ga preko bootp protokola).

**Protokol TFTP**

Zelo poenostavljena funkcionalnost ftp protokola(ohranjena predvsem možnost prenosa podatkov).

Odjemalec ve, kam mora iti po operacijski sistem(uporaba BOOTP protokola, s katerim pridobi IP naslov naprave, kjer se nahaja OS), za prenos OS na odjemalca uporabi TFTP protokol, ki na prenosni plasti uporablja nepovezavni način UDP. Odjemalec ima od programske opreme pri sebi samo še BIOS, v katerem se nahaja protokol TFTP.

Protokol(TFTP) nima avtentikacije, kriptiranja in možnosti imenika ( Linuxov ukaz **ls** ne deluje). Dobra lastnost -> maximalno veliki paketi. (Datoteke do 1TB).

**Kako načrtovati protokol?** **Kdo bo začel?** Odjemalec(pošlje zahtevo za datoteko, ki jo je dobil kot parameter v Bootp-ju). To zahtevo pošlje kot navaden UDP paket(ne vemo, ali je bil dostavljen(rešitev: določimo timer, če ni odgovora po nekem času pošljemo še enkrat, potrebno je šeoznačevati pakete, ker se izgubljajo(s strani strežnika)). Strežnik ve, da je paket dostavljen, ker odjemalec vpraša za naslednjega(ali potrditev). Strežnik mora sporočiti, kdaj je konec datoteke(prenosa).

**TFTP-PRIMER POGOVORA OB BRANJU**

-odjemalec pošlje zahtevo po branju (RRQ)  
-strežnik odgovori z DATA paketom in podatki, ki jih je zahteval odjemalec; poslani so z novih vrat in vsa komunikacija z odjemalcem mora odslej potekati preko teh vrat.  
-na vsak paket podatkov odjemalec odgovori z ACK paketom, nakar strežnik pošlje naslednji paket (prejšnja točka) – če potrditve ni v določenem času, strežnik ponovno pošlje paket  
-posebnost je zadnji paket, ki je manjši od največje dovoljene velikosti

**Programska oprema:**

Namenski software: tftpboot (ne dovoli oblik imen datotek ../, ker bi s tem lahko prišli kamorkoli, tudi do gesel).  
Ko imamo operacijski sistem v ramu(dobili smo ga od strežnika), ga rabimo samo še zagnati.

**Priklop na internet z računalnikom, ki pa ima disk**

Pridobiti moramo: IP, masko, privzet prehod, naslov dns strežnika(4. Osnovne zadeve). Tukaj pride v poštev polje VEND (omogoča opcijske nastavitve). V ta namen lahko uporabimo**bootp** s spremenjenim namenom polja VEND v bootp-u.

**RAZŠIRITVE VEND:**definirano, da lahko vsebuje poljubno (omejeno) število različnih vsebin.  
Stvar je narejena kot povezan seznam. Konec označen z magic cookie.

**PROTOKOL DHCP**

Dynamic host configuration protocol

Omogoča možnost priklopa na internet, pošljemo request paket, potuje na omrežje, odzove se DHCP strežnik, dobimo odgovor v katerem se nahaja nas IP. DHCP strežnik pošilja enake pakete kot bootp, ker je razširitev le tega. Vsebina paketa vsebuje tudi naslov prek katerega lahko dostopamo do OS(DHCP lahko pošlje naslov -> identična funkcionalnost bootp-ju). DHCP tudi definira, kako se ob priklopu na omrežje naprava obnaša.

*Izziv:  kako je z varnostjo pri DHCP protokolu? Za varnost je slabo poskrbljeno, zato se ga ne uporablja v sistemih, kjer je potreba večja varnost. Težava nastane že na začetku, ko pošljemo zahtevo za IP naslov, ne vemo če je DHCP pravi ali je ponarejen. Prav tako pa DHCP ne ve, če je client pravi ali je napadalec.*

DHCP: »Tukaj imaš vir, in ga imaš nekaj časa na voljo. Vse ostalo je namenjeno temu, da dobimo IP naslov kot vir, in ga lahko na svojem računalniku uporabljamo. Pomen med DHCP in bootp pa je pomensko malo različen.«

*Izziv: kakšno vrednost ima lahko značka DHCP message type? Ima lahko 8 različnih vrednosti:*

-          *1 = DHCP Discover message (DHCPDiscover).*

-          *2 = DHCP Offer message (DHCPOffer).*

-          *3 = DHCP Request message (DHCPRequest).*

-          *4 = DHCP Decline message (DHCPDecline).*

-          *5 = DHCP Acknowledgment message (DHCPAck).*

-          *6 = DHCP Negative Acknowledgment message (DHCPNak).*

-          *7 = DHCP Release message (DHCPRelease).*

-          *8 = DHCP Informational message (DHCPInform).*

**DHCP življenjski cikel**

Odjemalec pošlje zahtevo, če se oglasita dva DHCP strežnika se odjemalec sam odloči katerega uporablja, naredi se konfiguracija. Od DHCP-ja dobimo IP, ki je časovno omejen. IP je nek vir katerega lahko uporabljamo za dostop do omrežja. DHCP in bootp se razlikujeta v tem, da bootp nima opravka z IP naslovi.

**DHCP nevarnosti**

-          Neavtorizirani strežniki in odjemalci (ni pravih pravice za dostop, IP-fising napad),

-          Z veliko DHCP zahtevami naredi DOS napad, rezervirajo se vsi IP.

**DHCPv6**

Je bil na novo definiran, ni bil naslonjen na obliko paketov v DHCPv4(bootp).

Dve konfiguraciji računalnika:

-          Brez stanja  – se računalnik sam nastavi

-          S stanjem – se računalnik nastavi preko drugih enot

Jedro protokola ima dodatna polja, kar se tiče funkcionalnosti pa je enako kot pri IPv4. Sporočilo je nekoliko drugačno zaradi spremenjenih polj.

*Izziv: kaj je DUID? DUID (DHCP Unique Identifier) to je polje, ki se ustvari ko client dostopa do DHCPv6. DUID je podobno kot MAC naslov pri DHCPv4. Z njim se identificira.*

**Nadzor in upravljanje z omrežji**

Kaj pomeni network management? Zakaj je to potrebno?  
- Nadzorovanje omrežja kot takega da pravilno deluje.  
- Nadzorovanje prometa na omrežju.  
- Nadzorovanje naprav priključenih v omrežje.  
Skrb za učinkovito delovanje omrežja v realnem času.  
Skrb za pravilno delovanje omrežja, da ni napak, zastojev, nadzor prometa, omrežje dobi smisel ko je nanj priključeno več naprav.  
Pri velikem številu sistemov na omrežju moramo biti sistematični.   
Pogosta vprašanja:

* Kateri viri so na razpolago v omrežju?
* Koliko prometa gre skozi določeno omrežno opremo?
* Kdo uporablja povezave, zaradi katerih direktor prepočasi dobiva e-mail?
* Zakaj ne morem pošiljati podatkov določenemu računalniku?

Če ne opazujemo omrežja, ne moremo vedeti kaj se dogaja (testiranje). Ko pride do napake želimo analizirati, če je res prišlo do napake, ukrepamo tako da ponovno konfiguriramo naprave, nadzorujemo omrežje (monitor and control -> nadzor in upravljanje).  
Primeri nadzora:

* Napaka na napravi, napako sporoči nekomu - nekemu sistemu, za katerega želimo da bo napako odpravil sam po sebi, ali obvestil administratorja (lahko nas); primer: mrežna kartica je prenehala delovati, naročimo novo
* Nadzor dostopa do omrežja, prekinemo dostop do omrežja napravi, za katero ne želimo da do njega dostopa.
* Nadzor prometa na omrežju, npr. želimo priklopiti novo napravo na omrežje, zato se prej pripravimo, da pri prometu ne bo prišlo do ozkega grla...
* Proaktivno delovanje (pri vdorih)

Primeri aktivnosti:

* Nadzor delovanja računalnikov in analiza omrežja (odkrivanje topologije omrežja); omrežje doma: usmerjevalnik in npr. pet računalnikov -> natančno poznamo topoglogijo in vemo kako bomo priključili posamezen računalnik; težava se pojavi pri velikem številu računalnikov ter spreminjajočem se omrežju - potrebujemo program in primeren protokol(da vidimo sliko omrežja)
* Nadzor omrežja in prometa; zasedenost linije, pregled količin prometa, profiliranje, od protokola želimo imeti možnost, da pridobi podatke kaj se dogaja na nekem vozlišču, potem lahko primerno omejimo linije določenim uporabnikom/napravam
* <etičnost uporabe p2p; mnenja za, proti, osebno mnenje; odločitev za objavo avtorskega materiala naj bi bila v rokah avtorja; če avtor želi, naj objavi svoje delo, da bo postal bolj prepoznaven; inženirji imamo neko odgovornost in se moramo zavedati posledic in vpliva produkta, ki ga načrtujemo>
* Nadzor požarnih zidov, kaj se dogaja s prometom, vemo kdaj je prišlo do napada, kdo je prihajal čez zid; tako tudi vemo od kje je prišel napad
* Diagnostika in odkrivanje napak, kaj se dogaja z napravami, napadi, preveliko izkoriščanje virov

4 področja upravljanja:

* S konfiguracijami (upravljanje virov, npr: ip naslov, zanj skrbi DHCP, radi bi upravljali z DHCP strežnikom; potrebujemo podatke o tem koliko ipjev je razdelil, koliko jih še ima, koliko časa lahko nekdo še ima ip)
* Z varnostjo (eden od segmentov, ki postaja pomemben; kdo sme dostopati do določenega vira, kdo je izvedel napad na omrežje ali storitev)
* Beleženje (dostopov, oddajanj, DHCP naslovov)
* Z napakami (katera mrežna kartica ne deluje...)

Software za upravljanje

* CLI (command line interface) - ni več najbolj zaželjen način, upravljanje je natančno
* GUI aplikacije - vizualno lepši način

Infrastruktura za upravljanje  
komponente:

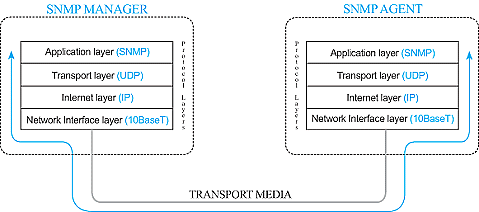
* Upravljalec (aplikacija + človek)
* Nadzorovana naprava (vsebuje agenta NMA in nadzorovane OBJEKTE, ki vsebujejo nadzorovane PARAMETRE; viri ki jih nadzorujemo so lahko požarni zidovi, različne naprave...)
* Protokol za upravljanje (SNMP na primer)
* Protokol SNMP

**Protokol SNMP**

Protokol SNMP (Simple Network Management Protocol) je protokol za upravljanje IP omrežij. Uporablja nepovezavni način UDP in skrbnikom omogoča upravljanje prikjučenih naprav brez dodatne programske opreme. S protokolom SNMP lahko tudi spremljajo delovanje omrežja, odkrivamo težave in imamo nadzor nad tem, kdo uporablja omrežje in na kakšen način.

Upravljamo lahko: routerje, tiskalnike, terminale, delovne postaje, računalnike, strežnike, programsko opremo.

SNMP deluje na aplikacijski plasti. Agent SNMP prejme prošnje preko UDP porta 161. Upravitelj lahko pošlje zahtevke iz vseh razpoložljivih izvornih portov na port agenta (port 161). Agentov odgovor bo poslan nazaj na vrata upravitelja. Upravitelj prejme obvestilo ([Traps](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#Trap) in [InformRequests](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#InformRequest)) na port 162. Agent lahko generira obvestilo iz kateregakoli prostega porta.



SNMP sporočilo prehaja skozi plasti protokola na ravni upravitelja in agenta. Vsaka plast obravnava posebne komunikacijske naloge.

2 načina delovanja:

* zahteva-odgovor (pošljemo zahtevo, dobimo odgovor)
* trap message (ob nekem dogodku naprava sama obvesti upravljalca)

Tipi sporočil SMTP:

* GetRequest            - zahteva upravljalca
* GetNextRequest     - zahteva upravljalca
* GetBulkRequest     - zahteva upravljalca
* InformRequest        - prenos podatkov MIB med upravljalci
* SetRequest            - MIB podatki od upravljalcev k agentom
* Response               - odgovor agenta
* Trap                        -obvestilo upravljalcu o dogodku

**GetRequest (Dobi zahtevo)**

Zahteva gre v smeri od upravitelja proti agentu. To je zahteva za pridobitev vrednosti ali seznama spremenljivk. Načrtovane spremenljivke so določene v spremenljivih priponah (vrednosti se ne uporabljajo). Pridobivanje določenih spremenljivih vrednosti je treba narediti kot atomsko operacijo s strani agenta. Odziv s trenutnimi vrednostmi se vrne.

**SetRequest (Nastavi zahtevo)**

Zahteva gre v smeri od upravitelja k agentu. To je zahteva za spremembo vrednosti spremenljivke ali seznama spremenljivk. Spremenljive pripone so določene v telesu zahteve. Spremembe k vsem določenim spremenljivkam so narejene kot atomske operacije s strani agentov. Odziv s trenutno novimi vrednostmi se vrne.

**GetNextRequest**

Zahteva gre v smeri od upravitelja k agentu. To je zahteva za odkrivanje razpoložljivih spremenljivk in njihovih vrednosti. Vrne odziv s spremenljivo pripono za leksikografsko naslednjo spremenljivko v bazi MIB. Po celotni agentovi bazi se sprehajamo z iterativno aplikacijo od *GetNextRequest*, začenši na OID 0. Vrstice tabele so brane po posameznem stolpcu OID (Object identifier) v spremenljivih priponah zahtevka.

**GetBulkRequest**

Optimizirana različica *GetNextRequest*. Zahteva gre v smeri od upravitelja k agentu. To je zahteva za množične iteracije *GetNextRequest*. Vrne odgovor z več spremenljivimi vezmi, ki gredo iz spremenljive vezi ali večih vezi v zahtevi. PDU (Protocol Data Unit) posebna nerepetitorska in maksimalno ponavljalna polja so uporabljena za kontrolo odziva obnašanja. *GetBulkRequest* je bil predstavljen v različici SNMPv2.

**Response (Odziv)**

Vrne spremenljive pripone in potrditve od agenta do upravitelja za GetRequest, SetRequest, GetNextRequest, GetBulkRequest in InformRequest. Poročanje o napakah je na voljo preko polj error-status in error-index. Čeprav je bil uporabljen kot odziv k obema gets in sets, je ta PDU imenovan GetResponse v verziji SNMPv1.

**Trap (Past)**

Asinhrono obvestilo v smeri od agenta do upravitelja. Vključuje trenutno vrednost *sysUpTime*, OID identifikacijo tipa pasti in izbirne spremenljive pripone. Destinacijsko naslavljanje za pasti je določeno v aplikacijsko-specificiranem načinu običajno skozi konfiguracijske spremenljivke pasti v bazi MIB. Format sporočila Trap je bil spremenjen v verziji SNMPv2 in PDU je bil preimenovan*SNMPv2-Trap*.

**InformRequest**

Potrjeno asinhrono obvestilo upravitelj-upravitelj ali agent-upravitelj. Obvestila upravitelj-upravitelj so bila že možna v verziji SNMPv1 (uporabljajoč *Trap*), ampak ker SNMP poteka preko UDP, kjer dostava ni zagotovljena in se ne poroča o izgubljenih paketih, dostava pasti ni bila zagotovljena. *InformRequest* popravlja to s pošiljanjem potrdila nazaj k prejemniku. Prejemnik odgovori z *Response*parroting vseh informacij v *InformRequest*. Ta PDU je bil uveden v verziji SNMPv2.

Obstaja več verzij SNMP:

SNMPv1 – prva verzija iz 80-tih, pokriva osnovne funkcionalnosti vendar ne omogoča želenega, ni varnosti

SNMPv2 – hitrejši, implementirana varnost, vendar prekompleksna

SNMPv3 – zadnja verzija, zagotavlja varnost na podlagi simetričnega kriptiranja in integriteto paketov z razpršilno funkcijo. Omogoča zaščito proti ponovitvi že upravljene komunikacije z žetončki – tokeni. Omogočena je tudi kontrola dostopa na podlagi uporabniških imen.

SNMP: Varnostni mehanizmi

* kriptiranje vsebine paketov (PDU) - uporaba DES ključev
* integriteta - uporaba zgoščanja sporočil s ključem, ki ga poznata pošiljatelj in prejemnik
* zaščita proti ponovitvi že opravljene komunikacije (replay attack) - uporaba enkratnih žetonov - pošiljatelj mora sporočilo kodirati glede na žeton, ki ga določa sprejemnik (to je običajno število vseh zagonov sistema pošiljatelja in čas, ki je minil od zadnjega zagona)
* kontrola dostopa - kontrola dostopa na osnovi uporabniških imen

**Protokol CMIP**

CMIP je v primerjavi s SNMP precej bolj kompleksen in se skoraj ne uporablja več.

MIB – Management Information Base

MIB je navidezna podatkovna baza za upravljanje entitet v komunikacijskih omrežjih. Največkrat jo povezujemo s protokolom SNMP. Baza je strukturirana v obliki drevesa (je hierarhična) do entitet pa dostopamo z oznako objekta.

MIB vsebuje podatke, kako konfigurirati naprave, opis upravljanja, kako do njih dostopati, definicije delovanja naprav.

Potreben je jezik (sintaksa, struktura) za opis objektov in parametrov – SMI - Structure of Management Information. Omogoča konfiguracijo naprave.

Jezik vsebuje osnovne tipe (INTEGER, OCTET STRING, OBJECT IDENTIFIED, IPaddress...), sestavljene podatkovne tipe (OBJECT-TYPE, MODULE-TYPE) ter raznorazne metode. **Enaki napravi imata enak MIB!.**

Naprav je veliko, prav tako tudi proizvajalcev, zato mora biti to na nek način usklajeno.

Objekte v SMI lahko združujemo v module (podobno kot package v Javi). Moduli so vsebinsko povezane skupine objektov.   
Dodefinirana je revizija in naziv organizacije, ki je definirala modul. Te module lahko uporabi vsak proizvajalec.

Moduli so standardizirani in lastni proizvajalcem opreme (vendor specific).

IANA shranjuje standardizirane module  
IETF je zadolžena za standardizacijo MIB modulov za usmerjevalnike, vmesnike, drugo opremo  
    - potrebno poimenovanje standardnih komponent  
    - uporabi se poimenovanje ISO ASN1 (cifre), definira IANA

MIB moduli – standardizacija (drevesna razdelitev, na vrhu ISO, ITUT, Joint)  
**MDB** – management database – tu so shranjeni podatki (MIB) o napravah, ki se nahajajo v omrežju. Pri realizaciji te baze se postavljajo sledeči problemi:

-Če se nahaja na osrednjem strežniku in če le-ta odpove, potrebujemo zasilni strežnik, ki začasno prevzame njegovo vlogo.

- Če bi vsaka naprava imela celotno bazo, bi se pojavil problem pri posodobitvi podatkov – če bi se ena vrednsot spremenila pri eni napravi, bi se moralo to sporočilo poslati vsem napravam v omrežju. Tako hitro pride do zamašitve linije.

Dejansko vsaka naprava zase hrani svoje podatke – baza je torej porazdeljena.

Kako zasnovati protokol za upravljanje?

Glavne funkcionalnosti naj bi bile:

-obvesti naj nas takoj, ko je z napravo nekaj narobe

-omogoča naj pridobitev podatkov in stanj o vseh napravah v omrežju

-omogoča naj nastavljanje vrednosti v MIB-u (npr. spremembo nastavitev požarnega zidu na routerju)

Ker si želimo preprost in hiter protokol, ki mora delovati tudi na napravah z malo pomnilniškega prostora, naj uporablja UDP. Le-ta pa ima tudi nekaj pomajkljivosti:

* ponovno pošiljanje - kako zagotoviti dostavo in potrjevanje? **Če v nekem časovnem intervalu ni odgovora, se potrditev pošlje ponovno.**
* problem z izgubljenimi obvestili - **rešen z oštevilčenjem**
* velikost paketov - UDP ima zgornjo mejo velikosti segmenta. To se lahko reši na 2 načina: Agent mora vedeti, kakšno velikost lahko sprejme prejemnik in poskuša sporočilo zgenerirati tako, da te meje ne presega. Če pa je podatkov še vedno preveč,se uporabi fragmentacija (ki pa ni ravno zaželjena, saj je zelo nezanesljiva in pogosto vodi do izgube podatkov)

**Alternativni načini nadzorovanja omrežja**

-Tehnologije na osnovi XML in SOAP: Podatki se hranijo v XML strukturah – nazorno in fleksibilno, za izmenjavo podatkov pa uporabimo SOAP. Implementacija je zelo zahtevna, lahko se uporabi tudi REST, ki je za izmenjavo podatkov enostavnejši.

-CORBA (Common Object Request Broker Architecture) – tehnologija, ki omogoča izmenjavo podatkov neodvisno od različnih računalniških arhitektur in programskih platform, ki pa ni nikoli resnično zaživela

-Dogodkovno gnano opazovanje -  Remote Network MONitoring (RMON) – Agenti RMON zbirajo in analizirajo podatke  lokalno, nato pa jih pošljejo oddaljeni nadzorni napravi. Nadzorna naprava lahko upravlja omrežje s klasičnim SNMP, vendar je manj prometa ker so podatki že obdelani.

**Promet za aplikacije v stvarnem času**

Želimo doseči, da prejemnik, ki mu preko omrežja pošljemo zvok ali sliko (podatke), dobi na svoji strani enak občutek časa kot, da bi vsebino gledal oz. poslušal na strani pošiljatelja.   
Pomembni pojmi:  
čas dospetja, čas začetka izvajanja, potreben čas za izvajanje, rok zaključka izvajanja

2 načina realnega časa:

- strog realni čas: dogodek se mora zgoditi strogo v predpisanem roku, če ne se ne zgodi

- mehki realni čas: bolj fleksibilen, sprejemljive zamude

**Zajem podatkov - zvok**

* zvok je analogen pojav spreminjanja zračnega pritiska, ki ga zaznava uho
* zvok je linearna kombinacija večih sinusnih signalov
* preddigitalno: zajem zvoka se je preko mikrofona pretvoril v analogni električni signal, ki se je uporabljal za proizvajanje zvoka preko zvočnika
* digitalno: zajamemo odmik(amplitudo, jakost, energijo). Amplitudi pretvorimo v n-bitno številko.
* .wav datoteke imajo ponavadi 44.1 kHz frekvenco vzorčenja
* Nyquist-ova frekvenca: je frekvenca, ki navadno tiči na polovici frekvence vzorčenja. Pojavi se t.i. „aliasing“ problem, ki je neke vrste statična distorzija. Npr. pri frekvenci vzorčenja 20 kHz, je Niquistova frekvenca pri 10 kHz. Signal pri 11kHz se bo mešal s signalom pri 9kHz, in obratno. Če si to predstavljamo na zajemu slike, bomo pri določeni hitrosti avtomobila, ki se vozi naprej, videli, da se platišča „vrtijo nazaj“.
* mp3 stiskanje: človeško uho ne zazna določene kombinacije signalov, zato te frekvence signalov „porežemo“ in prihranimo na prostoru

**OMREŽNI ČAS**

Včasih moramo uskladiti čas med večimi oddaljenimi sistemi. Problem je zakasnitev prenosa podatka.Za uskladitev časa uporabljamo protokol NTP (Network Time Protocol) oz. NTP strežnike.

Čas moramo uskladiti oz. spremeniti, če:

-se menja časovni pas: tega ne moremo omogočiti, saj nimamo podatkov o tem, v katerem časovnem pasu se računalnik nahaja. Tudi pri priključitvi na omrežje (protokol DHCP) tega podatka ne dobimo, saj so DHCP strežniki lahko drugje kot pa na naši lokaciji.

- pri prehodu na zimski/letni čas: tukaj ni težav pri spreminjanju časa, saj imamo točno določen datum, ko se mora ura zamakniti naprej ali nazaj.

- če je z našo lokalno uro kaj narobe: primerjamo z neko točno uro, da vidimo, če je naša lokalna ura pravilna, vendar moramo zraven vračunati še zakasnitev zaradi prenosa podatkov. To bi funkcioniralo, če omrežja ne bi bila paketna in tako imela zamud. Čas ugotovimo tako, da vprašamo "sosede". Ti nam povejo koliko je ura in koliko časa potrebuje paket za svojo pot. Tako dobimo pravilen čas.

- Marzullov algoritem

V Sloveniji se za čas uporabljajo Arnesovi NTP strežniki:

* ntp1.arnes.si. ntp2.arnes.si, ntp3.arnes.si

**OBČUTEK REALNOSTI**

**Kako prenesti podatke od A do B, tako da bo B pri branju podatkov imel občutek realnosti?**

Možne rešitve:

- A posname dogodke in časovne značke in pošlje datoteko B (brez ostalih informacij, da dogodek ni v živo, nam tudi to daje občutek realnosti)

- A, ko posname dogodek, ga opremi s časovno značko in ga takoj pošlje B (najboljša rešitev, vendar lahko pride do izgub paketov)

- nekaj vmes

Naše omrežje je paketno in zato lahko vsak paket potuje po drugi poti, različno dolgi ali pa se lahkocelo izgubi. Pojavita se 2 problema:

- Kaj narediti z izgubljenimi paketi? Če bi stvar prepustili TCP - ju, bi ta pošiljal paket, dokler ne bi prispel, vendar tukaj pride do zakasnitve, saj se naslednji paketi ne pošljejo, dokler ne pride izgubljeni paket. Zato naj se aplikacija odloči, kaj bo naredila z izgubljenimi paketi.

- Neenakomerno prihajajoči paketi: Sprijazniti se moramo s tem, da nekateri paketi preprosto zamudijo.

- Čas prihoda poslanega paketa od A-B je treba povečati toliko, da ima protokol dovolj časa na novo poslati izgubljen paket.

Občutek realnosti nam omogoča protokol **RTP (Real-Time Protocol, definiran v RFC 3550)**. Njegove osnovne funkcionalnosti so, da skrbi za pravo zaporedje paketov in da skrbi za časovne značke dogodkov.

**Ima pa tudi dodatne funkcionalnosti:**

primer da želimo združevati vire dogodkov v predavalnici (prosojnice, mikrofon, projektor, kamera..)

- ena povezava lahko prenaša več podatkovnih tokov (virov dogodkov: slika levega/desnega očesa, zvok levi/desni...)

- ima identifikator vira (identifikator seje)

- ima poseben element -mešalec(mixer), ki združuje več  RTP tokov iz različnih virov v en sam tok.

- združeni seji (komu v resnici pripada poslani paket)

**RTP**

Real Time Protocol je prenosni protokol, ki služi za prenos podatkov v realnemu času.

Največ se uporablja v sistemih za komunikcije in zabavo (razne push-to-tolk aplikacije, telefonija in video telefonija).

RTP se uporablja v povezavi z RTP Control Protocol (RTCP) ter RTSP (Real Time Streaming Protocol) .

RTP omogoča aplikacijam prenos podatkov (za predvajanje zvoka, filma...), nima pa nadzora, ni podatkov o delovanju prenosa, za to pa skrbi protokol RTCP (RTP Control Protocol).

**Nadzorni protokol RTCP (RTP Control Protocol, definiran v RFC 3550)**

Opravlja 4 funkcije:

- sporoča o kakovosti prometa

- skrbi za pravilno gostoto pošiljanja sporočil o kakovosti prenosa

- vsebuje dodaten opis vira toka dogodkov

- prenaša lahko še druge podatke za potrebe aplikacije

**RTP oblika paketa (RTP header):**

 V - trenutna verzija RTP je 22 bitov (**RTCP** velik 32 bitov)

 P - zapolnitev 1 bit

 X - nam pove ali je prisotna razširitev glave 1 bit

CC - število mešanih virov 4 biti

M - poseben bit za potrebe protokola 1 bit

PT - identifikacija protokola (format) 7 bitov

SEQUENCE NUMBER - številčenje paketov poslanih v toku, uporablja se za preverjanje če so vsi paketi prišli

(če ni 100% dostave, za to mora poskrbeti aplikacija) 16 bitov

TIMESTAMP - časovna značka dogodka 32 bitov

SSRC - identifikator vira 32 bitov

CSRC - identifikatorji mešanih virov

**VARNI RTP**

Varnost za RTP moramo dograditi sami tako da izmenjamo ključe, ampak poslani paketi se izgubljajo.

Zaradi tega uporablja drugacen nacin kriptiranja: KRIPTIRANJE S TOKOM ŠIFER

**KRIPTIRANJE S TOKOM ŠIFER**

Simetričen ključ se spreminja -> problem sinhronizacije: prejemnik mora vedeti kateri ključ jepošiljatelj uporabil (ključ lahko vežemo na številko paketa. Izboljšava številčenja paketov -> Funkcija, ki nam zgenerira ključ).

Omogoča kriptiranje samo vsebine!

 - začetna vrednost in ključ sta poznana obema stranema

 - vsak paket se ločeno zakriptira (na podlagi številke paketa se zgenerira simetričen ključ)

 - potem se naredi preprosti xor ali kakšen podoben algoritem

 - E zavrtimo v prazno tolikokrat kot se paket izgubi

**PROTOKOL SRTP (**[**http://tools.ietf.org/html/rfc3711**](http://tools.ietf.org/html/rfc3711)**)**

 ali The Secure Real-time Transport Protocol

- zasnovan na RTP

- dodana varnost s kriptiranjem s tokom šifer

**Uporaba RTP**

- IP telefonija - SIP

- Beleženje dogodkov v (oddaljenih) laboratorijih (gridcc)

- Oddaljeni VCR ali VoD (uporablja RTSP)

**PROTOKOL RTSP (**[**http://tools.ietf.org/html/rfc2326**](http://tools.ietf.org/html/rfc2326)**)**

ali Real Time Streaming Protocol (sorodnik protokola http)

RTSP = začetek seje + RTP protokol ( za prenos podatkov )

- Osnovni ukazi: Setup, Play, Record, Pause, Teardown  + dodatni ukazi za nastavljanje in branje parametrov.

Programska oprema: Strežnik- Darwin , Odjemalec- VLC

**Razpošiljanje**

ali Multicast

Pojem skupine prejemnikov: ime skupine, številka, naslov (možnosti: ali obstaja, ustvarjanje, ukinjanje, dodajanje uporabnikov...)

mrežna plast poskrbi, da se vsebina pojavi pri vseh v isti skupini

**Načini naslavljanja:**

- unicast(tradicionalno): pošiljanje enemu ciljnemu IP naslovu (unikaten v internetu/lokalnem omrežju)

- broadcast: naslavljanje "vseh prejemnikov" v podomrežju, ne dostavlja paketov izven omrežja

Kako pošiljamo samo izbrani skupini naslovov, tudi izven lokalnega omrežja:

 - multicast: naslavljanje omogoča dostavo skupinam ne glede na meje podomrežij; IP podatki na mrežni plasti,

nadzoruje ga IGMP (mora "znat pogruntat" da ima prejemnika določene skupine v IGMP-ju)

- IGMP (internet group management protocol): uporaba pri upravljanju s skupinami

**Razpošiljanje primer (iz prosojnic)**

Poslati želimo 4 od 6 računalnikom v omrežju:

- unicast: 6 kopij istega paketa -> časovno potratno

- broadcast: naslovi vse računalnike, filtriranje pravih prejemnikov prepustimo protokolom na višjih plasteh

- multicast: "posebni naslov" -> predstavlja skupino prejemnikov, ki posluša pakete naslovljene na ta naslov

paket dobijo vsi, filtriranje na omrežnem (IP) nivoju, včasih tudi povezavnem

**Razpošiljanje: usmerjanje paketov**

- broadcast paketov usmerjevalniki ne posredujejo (dobili bi jih vsi), torej ostajajo znotraj lokalnega omrežja.

- usmerjanje pri razpošiljevanju je praktično: en sam paket usmerjevalniki razmnožijo in posredujejo samo preko

tistih vmesnikov, kjer so poslušatelji paketa. Izzivi protokola: odkrivanje kje so prejemniki paketa,

varnost, posrednovanje informacije o poslušateljih.

**Aplikacije razpošiljanja**

- pošiljanje velikih datotek preko omrežja,

- data streaming (npr. pošiljanje podatkov o delnicah vsem finančnim institucijam)

- nadgradnja programske. opreme v velikem omrežju

- video na zahtevo (spremljanje TV programa)

- izvedba konferenc, audio/video streaming

- aplikacije v realnem času z RTP, ki se uporablja za zagotavljanje tekoče in kakovostne dostave v okoljih, kjer se uporablja razpošiljanje.

**1.) NASLAVLANJE SKUPIN**

Skupina se poimenuje s posebnim IP naslovom, od celega nabora je nekaj naslovov rezerviranih in ti predstavljajo skupine.

IPv4

Prvi 4-je biti so oblike 1110 (razlog zato je prvotna delitev na skupine A,B,C,D pred uvedbo maske podomrežja) ki nam povedo da gre za skupino.

Naslednji 4 biti pa nam povejo tip skupine.

Lokalna – npr: vsi z Univerze v Ljubljani

Globalna – vsi dosegljivi člani skupine v internetu

Znana – za vse usmerjevalnike in vmesnike

            - za vse usmerjevalnike

Vse to se dogaja na omrežni plasti (3.), ki tudi poskrbi za to da paket prispe na naslov, zazanesljivost prenosa pa poskrbi plast nad njo oz.

….protokol TCP.

Način po katerem deluje je 3 smerno rokovanje, za naš primer to pomeni:

1) Vzpostavitev povezave                 (vsi člani skupine se prijavijo)

2) Zanesljiv prenos podatkov           (potrditev da so vsi prejemni člani dobili pakete)

3) Prekinitev povezave                      (vsi člani se odjavijo)

...protokol UDP

-        se uporablja pri razpošiljanju na prenosni plasti

-        mreža sama, ne ve kdo je kdo. Torej kateri član je del skupine kateri ne.

IPv6

Princip je enak

Skupino določa prvih 8 bitov --> 1111 1111 torej FF

Naslednji 4 biti določajo obstoj skupine (1-temporary, 0-permanent)

Naslednji 4 biti določajo obseg razpošiljanja:

1)     NODE – znotraj enega računalnika, imamo nekaj aplikacij, ki si želijo izmenjati podatke, zato jih damo v isto skupino.

2)     LINK – imamo več računalnikov znotraj omrežja, ki želijo komunicirati in jih zato postavimo v isto skupino. Komunikacija poteka znotraj istega "vodila".

5)     SITE – enak obseg kot pri link, razlika je, da A lahko komunicira tudi z F, ampak samo čeuporabi globalni naslov.

8)   ORGANIZATION – obsega več računalnikov iz različnih SITE obsegov znotraj ene organizacije (SIOL).

E)  GLOBAL – obsega naprave kjerkoli v internetu.

Tvorjen paket se nato preda na povezavno plast (2), ki ga opremi z MAC naslovom. Tudi tukaj jepotrebna posebna definicija kakšen naslov pomeni multicast. Za to je določena oblika MAC naslova. Načina razpošiljanja paketov sta 2:

-        vsakomur svoj paket

-        vsem vse pakete (tisti ki so znotraj skupine so nastavljeni tako, da tovrstne pakete preberejo, ostali jih zavržejo. Če imamo zavarovano potem imajo prejemniki ključ s katerim dešifrirajo vsebino).

Lahko se pojavi težava, če je po prvih 4 bitih,  katerikoli od naslednjih 5 bitov ne-ničeln, ta problem mora znati reševati usmerjevalnik. Torej, da loči katere pakete bo poslal na kakšen način oz. komu.

Naslavljanje multicasta na povezavni plasti

Naslov multicasta je logični identifikator, ki določa skupino računalnikov, za katero je namenjenorazpošiljanje. Ethernet okvirji z Mac naslovi z 01 v prvih dveh številih so obravnavani kot multicast naslovi.

Poznani ethernet multicast naslovi

01-00-0C-CC-CC-CC ->  [CDP](http://en.wikipedia.org/wiki/Cisco_Discovery_Protocol" \o "Cisco Discovery Protocol) (Cisco Discovery Protocol), [VTP](http://en.wikipedia.org/wiki/VLAN_Trunking_Protocol) (VLAN Trunking Protocol)

01-00-0C-CC-CC-CD ->  Cisco Shared Spanning Tree Protocol Address

01-00-5E-xx-xx-xx ->  IPv4 Multicast

33-33-xx-xx-xx-xx ->  IPv6 Multicast

23 bitov iz multicast naslova ipv4 se preslika v MAC multicast naslov. Prejemnik ali usmerjevalnikmora iz izvlečka definirati še ostale bite IP naslova.

**2.) PRIJAVA V SKUPINO**

Pri skupini nas zanimajo možnost:

-        prijava v skupino

-        izstop iz skupine

-        prepoznava drugih članov

Vse to nam omogočata IGMP in MLD...protokol MLD(za IPv6 pakete)  
Ideja je, da vsak usmerjevalnik prepozna člane, ki želijo biti v skupini, prav tako pa naslove, ki solahko koristni za sosede.  
Usmerjevalnik se zaveda, za katere skupinske naslove obstajajo člani, ki so povezani na njega.(ne ve pa kdo so člani)  
.........protokol IGMP (na mrežni plasti in za lokalno omrežje, za IPv4 pakete)

Ideja je, da vsak usmerjevalnik poroča njegov IP skupine vsem sosednjim usmerjevalnikom, ki so posredno vezani na skupino. Zgodi se lahko, da je usmerjevalnika sam dejansko član skupine,   
takrat mora opraviti (delo usmerjevalnika, ki je posredno v skupini IN delo člana skupine)  
Delo usmerjevalnika, je da poizve informacije o članstvu v skupini, glede na zahtevo protokola  
Delo člana skupine, je da obvesti sebe in druge skupinske usmerjevalnike, da je sam postal novi član

Verzija 1 ima vse funkcionalnosti, ki jih potrebujemo za »zasilno« delovanje z skupinami, verziji 2 in 3 pa sta bili razviti za boljšo učinkovitost tako, da nam omogočata tudi informacijo o tem: kdo vse je v skupini, možnost odjave, ter možnost da imamo več vozlišč, ki so se sposobni odločiti kdo od njih postane super-vozlišče.

Ideja super-vozlišča je, da nanj usmerjevalniki pošljejo pakete, ki so namenjeni za skupino, super-vozlišče pa ve komu gre kateri paket in ga ustrezno pošlje naprej.

Naš usmerjevalnik mora biti torej sposoben storiti (preko super-vozlisca):

-        vsem dostaviti pakete

-        dostaviti pakete le tistim, ki so člani skupine in vedo kako se te pakete sprejema

to pa zato, ker se paketi za skupine pošiljajo drugače kot za posameznika.

trenutno naš paket izgleda takole:

Za prijavo v skupino se pošlje Group Membership Report najbližjemu usmerjevalniku (ki ima tabelo kdo vse je v skupini).  
Usmerjevalnik nato sporoči sosedom, da ima novega naročnika ---> problem poplavlanja  
Rešitev:  
 - RPL algoritem  
 - posebni usmerjevalni protokol (PIM-SM , PIM-MD)

**3.) RPL algoritem**  
 - je postopek za reševanje problema 2 paketov sprejetih od istega izvora  
 - X poslje nekaj za Z  
1.korak: X poslje paket na A  
2.korak: X poslje paket na B  
               A poslje paket na C  
3.korak: B poslje paket na Y  
               B poslje paket na C

4.korak: Y nima kaj s tem paketom zato se izgubi (mrezna plast --> best effort, ni zagotovil)  
5.korak: C ima sedaj 2 paketa, eden od A, eden od B.  
               C se vprasa: za paket od A, če bi jaz poslal paket na X, ali bi ga poslal na A ? ----> DA, zato ta paket sprejmem  
               in posredujem dalje na sosede  
6.korak: Z dobi paket  
7.korak: C se vprasa: za paket od B, če bi jaz poslal paket na X, ali bi ga poslal na B ? ----> NE, zato zavrzem  
8.korak: B dobi paket od C, namenjen za Z, ker nima kaj s tem paketom ga zavrze. **4.) USMERJEVALNI PROTOKOLI PIM-SM ter PIM-MD**

  - ideja je da najdemo drevo povezav, ki povezuje vse usmerjevalnike v isti skupini.  
  - povedo usmerjevalniku, da paketi namenjeni za določeno skupino, morajo biti poslani na sosede te skupine.  
  - se uporabljajo za gradnjo dreves, delujejo kot skupina.  
Delitev: 2 kriterija  
1.) Razpršeno / Gosto  
2.) Znotraj / Zunaj domene  
**1.) Razpršeno**: Potrebna je le vključitev novega člana v že obstoječe drevo (primerno za večje število vozlišč)  
     Gosto: Pakete razposljemo vsem nato se nepotrebni odjavljajo, 2 pristopa:  
                   a) Broadcast and Prune (za manjše število. vozlišč):   
                                         1.korak: iščemo drevo  
                                         2.korak: odrežemo vse povezave, katerih usmerjevalnik nima članov skupine    
                   b) Domain-Wide: prijavljamo vse člane na promet  
**2.) Znotraj domene**: se uporablja PIM-DM, torej gosto, za naprave znotraj recimo SIOL omrezja  
     Zunaj domene: uporablja se protokol PIM-SM, torej razpršeno, to pomeni, da se novega kandidata le doda v že obstoječe omrežje. Npr: če skupino ki obsega izven domene, recimo SIOL-T2, potem ne bomo vsakic gradili in podirali vse kar ne pase, ker je prevec usmerjevalnikov in naprav.

Primer: BitTorrent - Koncept Prekrivno Omrežje  
Skupnost se obnasa kot skupina naprav na internetu, in samo člani ki spadajo v to skupino so deležni paketov.  
Tisti ki spadajo v to omrežje se morajo sami dogovoriti in upravljajo z razpošiljanjem prometa.

**5.) PIM uporaba: RAZPOŠILJANJE PROMETA - DREVESA (kako se zgradi, kakšno drevo,kako deluje)**

Ideja je, da iščemo graf ki bo pokril vse naprave iste skupine in bo najcenejsi, z uporabo usmerjevalnih protokolov.   
**1.korak**: vzpostavitev arhitekture/topologije omrezja  
              Iscemo drevo, ki bo:   
                           imel čimkrajšo pot od izvora do ponorov in se bo zacel na dolocenem mestu (izvor)  
              2 resitvi iskanja drevesa:   
                          - group-shared tree (ce poznamo izvor)  
                          - source-based tree (ce ne poznamo izvora)

              group-shared tree  
                      1.moznost: isce se drevo z minimalno ceno  
                      2.moznost: za vsako skupino se definira RP, ki nato pakete razposlje vsem ostalim  
                                        RP=Rendez-Vous tocka, to je centralno vozlisce.   
                                        Usmerjevalnik se prikljuci v drevo, takoj ko sreca prvega ki je ze povezan na RP.

             source-based tree  
                         Loceno drevo za vsakega clana.  
                         1.moznost: isce se drevo najkrajsih poti  
                         2.moznost: uporaba RPL

**2.korak**: posiljanje podatkov  
           - usmerjevalniki se zaznajo in vzdrzujejo povezave s paketom PIM-SM HELLO (da se pridruzimo skupini)  
           - paket REGISTER: usmerjevalnik, ki pošilja podatke na skupinski naslov, obvesti RP s tem paketom, ki obvesti ostale  
**3.korak**: vzdrževanje naročnine  
           - paket REGISTER-STOP: ko RP zazna, da v skupini ni prejemnikov se ta paket pošlje izbranim usmerjevalnikom(odjava)  
             paket PIM-SM JOIN:  zgodi se dejanska prijava člana  
             paket PIM-SM PRUNE  zgodi se dejanska odjava člana

**6.) HRBTENIČNO OMREŽJE (MBONE – Primer: SIOL)**

 MulticastBONE je bilo poizkusno hrbtenično omrežje za IP razpošiljanje po internetu razvito v zgodnjih devetdesetih letih prejšnjega stoletja. Ker ima večina routerjev IP razpošiljanje onemogočeno zaradi zaradi sledenja in zaračunavanja povezave, je Mbone razvilo povezovanje razpošiljanja zmožnih omrežij po obstoječi internetni infrastrukturi. Komercializacija multicast usmerjevalnikov pa je težka, ker ne obstaja učinkovita kontrola dostopa do multicast dreves (multicast  usmerjevalniki in njihovi protokoli) in ker imajo ponudniki internetnih storitev težave pri zaračunavanju multicast prometa. Topologija je kombinacija mesh in topologije zvezde, ip naslovi so 224.2.0.0, registracija seje je preko IGMP protokola, zahteve prometa pa so: 32-64 kbit/s za audio in 120 kbit/s za video.

**Multicast** je pošiljanje na več naslovov, ki se zanimajo za paket. Vsa omrežja ne dopuščajo pošiljanja paketov na več naslovov. V zadnjem času je narasla potreba po tem načinu prenosa (npr. internetna televizija). Le redki protokoli poznajo pošiljanje na več naslovov. V TCP/IP obstaja "trik"z vmesnimi strežniki multicast.

**Varnostni elementi omrežij**

**IPsec**

Ime IPsec je krajšava za IP security. Gre za skupino protokolov, ki  omogočajo varen prenos podatkov na omrežni plasti. S tem, ko poskrbimo za varen prenos na omrežni plasti, pa zavarujemo tudi vse plasti višje. IPsec vsebuje tudi protokole za vzpostavitev medsebojne avtentikacije med agenti na začetku seje in pogajanjih o kriptografskih ključih, ki bodo uporabljeni med sejo. Je **end-to-end sistem varnosti**na omrežni plasti in se lahko uporablja pri varovanju toka podatkov med parom gostiteljev, varnih prehodov ali med parom gostitelj - varni prehod.

Deluje lahko na dva različna načina. To sta **AH** (Authentication Header) ali **ESP** (Encapsulating Security Payload). Pri ESP-ju zakrivamo podatke (kriptiranje). Vendar pa, če imamo veliko podatkov je zakrivanje lahko počasno, zato lahko uporabimo način AH. Tu podatkov ne zakrijemo, ampak poskrbimo za avtentifikacijo izvora, integriteto podatkov in zaščito pred ponovitvijo komunikacije. V praksi se večkrat uporablja način ESP. Pri obeh načinih pa imamo še dva načina delovanja. To sta tunelski in prenosni (neposredni). Pri prenosnem načinu  IPsec datagram potuje med končnimi odjemalci in ščiti samo zgornje plasti. Kriptira pa samo podatke v paketu. Medtem pa se pri tunelskem načinu IPsec izvaja na končnih usmerjevalnikih, tako da odjemalcem ni potrebno izvajati IPsec-a. Kriptira pa se tako glavo kot tudi podatke.

Dogovor o načinu šifriranja:

Sodelujoči strani se dogovorita o:

* algoritmu in ključu za šifriranje podatkov (za ESP),
* algoritmu in ključu za overjanje (za AH),
* času veljavnosti ključev,
* IP naslovih pošiljatelja in naslovnika,
* stopnji tajnosti (neobvezno),
* oznaki uporabnika (neobvezno).

Ta dogovor imenujemo **Security Association (SA)**. Vsaka stran določi svoj SA, v katerem je IP številka nasprotne strani, oznaka protokola (AH ali ESP) in pa 32-bitna oznaka **Security parameters index (SPI)**, ki označuje vse potrebne algoritme in ključe za šifriranje oziroma dešifriranje, za katere sta se dogovorila. Z SPI so na enostaven način ločili problem izmenjave ključev od overjanja in šifriranja paketov. Kako pride do dogovora, kakšen protokol bo tu uporabljen, še ni določeno. Vsi večji proizvajalci se trudijo, da bi bil izbran njihov predlog. Gre za postopek na višjem komunikacijskem sloju.

Uporaba tunel in ESP na paketu deluje na naslednji način:

Na originalno ip glavo in originalne ip podatke se doda ESP rep, ki je sestavljen iz zapolnitve, dolžine zapolnitve ter tipa protokola vsebovanega v naslednjih podatkih. Dobljeni rezultat kriptiramo s ključem, ki ga določa SA. Na te kriptirane podatke dodamo še ESP glavo, ki je sestavljena iz SPI indeksa, ki pove kateri SA je v uporabi ter zaporedne številke paketa, ki preprečuje napad s ponavljanjem. Dobljeni del se imenuje "enchilada". Na konec "enchilade" se doda polje ESP auth, ki vsebuje zgoščeno vrednost celotnega sporočila. To sporočilo se sedaj vstavi v podatkovno polje navadnega IP datagrama, ter se mu doda novo IP glavo, ki ima za številko protokola vrednost 50 in paket pošlje po omrežju.

Ker se uporablja tunel mode, sta prejemnik in pošiljatelj vozlišči med katerimi poteka IP sec. Prejemnik tega paketa iz SPI prebere podatke o SA, preveri integriteto sporočila (s pomočjo MAC-a "enchilade"), odkriptira "enchilado" odstrani ESP rep in podatke posreduje ciljnem računalniku.

**Protokol IKE**

Je infrastruktura, ki omogoča, da pridobimo spreminjajoče (spremembe v SPD) podatke (paramatre), katere potrebuje **SA.**Da tega ne delamo ročno, dodamo IPSec – u IKE  in takrat, ko IPSec vzpostavi povezavo mora dobiti SPI, preko katerega določi vse ostalo. Če SPI ima, in nima vsega ostalega vpraša IKE, da izve kje in kakšni podatki so v SPI.

Zato, da ni potrebno ob spremembah v SPD – ju vedno popravljati / replicirati SPD - ja na vseh sistemih, te spremembe hranimo v tej infrastrukturi.

IPSec IKE je razširjen na dve fazi:

* V prvi fazi je pridobivanje podatkov za SPI in preverjanje kdo je kdo
* V drugi fazi lahko začnemo komunicirati na ravni povezave

Prva faza je prisotna samo zato, da ni potrebno imeti SPI - ja na vsakem računalniku posebej ampak, da ga lahko pridobimo iz neke infrasrtukture  ->  uporabno pri širši uporabi.

**SSL**

SSL je varnostni mehanizem, ki deluje prek TCP protokola. Čeprav se uporablja predvsem za zaščito HTTP povezav (HTTPS), je lahko implementiran v vsaki aplikaciji, ki na transportni plasti uporablja TCP protokol. SSL deluje v treh fazah: rokovanje(*handshake*), izpeljava ključev(*key derivation*) in prenos podatkov(*data transfer*). V seji strežnik hrani par zasebnih/javnih ključev in certifikat, ki poveže stržnikov javni ključ z njegovo identiteto.

**Rokovanje**

Ob začetku rokovanja, mora klient vzpoztaviti TCP povezavo z strežnikom, mu pošlje sporočilo SSL*hello*, ta pa mu odgovori s certifikatom, ki vsebuje javni ključ. Ker je certifikat potrjen s strani CA-ja (certifikatna agencija), klient zagotovo ve, da javni ključ v certifikatu pripada strežniku, s katerim želi komunicirati. Klient nato generira *Master Secret*, ga kriptira z strežnikovim javnim ključem in ustvari*Encyrpted Master Key* in ga pošlje strežniku. Strežnik ta ključ dekriptira, da dobi *Master Secret*, ki ga tako poznata le strežnik in odjemalec, za to SSL sejo.

**Izpeljava ključev**

Odjemalec in strežnik bi lahko sedaj *Master Secret* uporabljala, kot simetrični ključ, za enkripcijo in preverjanje integritete podatkov za to sejo. Vendar, je povezava varnejša, če oba uporabljata različne ključe. Vsak izmed niju generira dva ključa:

* Eo = sejni enkripcijski ključ, poslan od odjemalca strežniku.
* Mo= sejni MAC ključ, poslan od odjemalca strežniku.
* Es = sejni enkripcijski ključ, poslan od srežnika odjemalcu.
* Ms= sejni MAC ključ, poslan od strežnika odjemalcu

Vsi štirje ključi se generirajo iz *Master Secret* sporočila. Z enkripcijskimi ključi, se bo sporočilo kriptiralo, MAC ključ pa poskrbi za integriteto podatkov.

**Prenos podatkov**

Odjemalec in strežnik sedaj imata vsak svoj MAC ključ in enkripcijski ključ, ki sta potrebna za zagotavljanje varnosti. Protokol TCP prenaša tok bajtov, z SSL-om pa ta tok "razbije" v manjše dele in jim doda MAC ključ in vsakega posebej kriptira. MAC ključ, ki se uporablja pri prenosu ni enak MAC ključu iz izpeljave ključev (Ms, Mo), saj mora biti ta za vsako poslano sporočilo drugačen. Način izračuna MAC ključa, za vsako sporočilo je sestavljen iz MAC ključa iz izpeljave ključev (Ms, Mo), vsebine sporočila, ki bo poslano in zaporedno številko poslanega sporočila. Ta novi MAC ključ je nato poslan še skozi razpršilno funkcijo, pripet sporočilu in kriptiran z enkripcijskim ključem (Es, Eo) strežnika, ali odjemalca. Zaporedna številka v MAC ključu, onemogoča *man-in-the-middle* napad, saj bi lahko napadalec poslane pakete prestrezal in jih naprej pošiljal v napačnem vrstnem redu.

**Napadi in grožnje**

**Uvod**

Napadi na omrežne sisteme so namenjenu škodovanju/obhodu računalniških in omrežnih funkcij zaradi denarne dobrobiti, škodovalnosti in poneverbe ("bad guy"). Ogrožajo integriteto, zaupnost in razpoložljivost omrežnih sistemov. Vendar sisteme prav tako ogroža malomarnost in napačna uporaba ("štorast guy"). Do napada sistema pa lahko pride tudi zaradi iskanja morebitnih lukenj in zagotavljanja varnost ("good guy").

**Pogosti napadi**

* šibki ključi
* matematični napadi na kriptografske algoritme in ključe
* ugibanje gesel (s slovarjem / brute force)
* virusi, črvi, trojanci
* izkoriščanje šibkosti v programski opremo
* socialni inženiring (izkoriščanje naivnosti uporabnikov preko emaila, socialnih omrežij)
* pregled vrat (port scan)
* brskanje po smeteh: način, s katerim napadalci pridejo do informacij o sistemu(seznami gesel, tel.št, org. dela)
* rojstnodnevni napad : napad na zgoščevalne funkcije, ki za različni sporočili zgenerira isto vrednost
* zadnja vrata (back door)
* ponarejanje IP naslovov: napadalec prepriča ciljni sistem, da je nekdo drug poznan s spreminjanjem paketov
* prestrganje komunikacije (man in the middle): napadalec prestreže komunikacijo in se obnaša, kot da je ciljni sistem (pri uporabi certifikatov lahko žrtev uporablja tudi javni ključ napdalca)
* ponovitev komunikacije (replay): prestreže stara sporočila in jih ponovno pošlje predstavljajoč se kot en izmekd udeležencev (preprečimo z uporabno časovnih značk)
* ugrabitev TCP sej
* napadi s fragmentacijo
* preprečitev delovanja sistema (dos) - prekoračitev medpomnilnika, SYN napad, napad Teardrop, napad Smurf

**Kako se obraniti?**

* Fizično varovanje (zaklepanje računalnikov, geslo za zagon (cmos/bios), onemogočiti zagon sistema s cdja/usb)
* Posodabljanje programske opreme
* Uporaba AV programa
* Uporaba požarnega zidu
* Varovanje uporabniških računov/datotečnega sistema/aplikacij (pravilne nastavitve, ne privzete vrednosti, onemogočanje priponk v email, onemogočanje izvajanja nevarnih tipov, nameščanje na nestandardna vrata/mape)

**Kaj je SSID?** Ime omrežja.

**Kakšna je vloga polja Sinhronised Source ID?** Ima vlogo identifikatorja, kdo pošilja stream paketkov.

**Iz kje lahko pravzaprav vidimo kaj ta naprava sploh je?** V RTCP protokolu so opisni podatki, kateri opisujejo vsak SSID.

**Zakaj je SSID tam noter in zakaj ni vedno enak?** Keri ma mogoče že drugi tisti ID in more biti dinamičen.

**Kaj je MSID?** Vir je lahko mikser(pri RTPju) – mikser sprejema podatke iz 3 mikrofonov in pošlje naprej kot 1 sam tok.

**Kot sprejemnik, kaj nas zanima pri prejemanju podatkov? V RTP(real time** protocol) ali je samo zvok, ali je tudi slika, RTP tudi več tokov združi in naredi manj, sicer večjih paketov.

**Kako vzpostavimo povezavo z drugim telefonom?** SIP je protocol, ki omogoča ustvarjanje povezave.

**RTSP vs. SIP?** RTSP je veliko bolj enostaven kot SIP, SIP omogoča tudi telefonske številke pošiljat ipd.

**Na kateri plasti se nahaja neighbour solification (discovery)?** Na omrežni plasti ker je to del ICMP protokola

**Kje tiči glavni razlog za opuščanje protokola ARP v omrežjih IPv6?** Problem ARP protokola je, da ga je zelo lahko napast. Govorili smo o napadu ARP poisoning ali ARP spoofing. Največja svinjarija je gracieus ARP – s tem lahko vsem računalnikom v omrežju povemo, da se je naš strojni naslov, tako da je zelo lahko izvest napad. Pri IPv6 to ni več mogoče, ker nima več gracieus. Če imamo v wiresharku gracieus paketek nastavljen na 1, pomeni da gre za ARP napad.

**ARP – neighbour discovery? Razlika podobnosti.** Gre pravzaprav za enak protokol, obadva omogočata da na podlagi IP-ja dobimo strojni naslov (MAC naslov) IPv4 uporablja ARP, IPv6 pa neighbour discovery.

**Loopback IPv4 IPv6, zakaj 127.0.0.0/8 in ::1?** Pač tako je, tudi če si nastavimo 127.1.2.3. je to še vedno pri loopback naš računalnik. Loopback je, da imamo lahko več naslovov na našem rač., kar pa pravzaprav ne potrebujemo, sploh pri IPv6.

**Kaj se uporablja pri brezžičnih kolezijah?** V wireless nimamo kolezij, ker uporabljamo CSMA CA protokol, tako da se želimo izognit temu. CTS – clear to send, RTS – ready to send … to sta dva signala. CA deluje na tak način, če imamo kej za poslat, pošljemo najprej RTS okvir acces pointu, acces point odgovori s CTS in nato lahko pošlje naš računalnik to naprej. Če ne dobimo CTS nazaj, ne smemo poslat. Tako ne pride do kolozij.

**Kdo in s kakšnim odgovorom odgovori na ARP poizvedbo, ki se sproži kot posledica iskanja fizičnega naslova k IP naslovu, ki ne sodi v lokalno omrežje?** ARP poizvebo pošljemo samo za svoj domači routerček, v kolikor ga še nimamo v svoji tabeli ARP. Od tu naprej pa router pošilja ARP poizvedbo za drug IP naslov, ki ni v našem podomrežju.

**Dolzina paketka ARP, Beacon frame, zakaj podatki niso sifrirani? Kateri kanal uporablja to omrežje?** Pogledamo v WS programu spodaj BYTES: 28 v našem primeru. Vse skupaj pa 42, celoten frame. Beacon okvirje vsaka wireless točka ves čas oddaja v omrežje, služi tem da se druge naprave znajo povezat na naše omrežje. Ker to želimo, da se naprave znajo povezat, beacon frame nikoli ni šifriran. (pogledamo tako, da ima protected flag nastavljeno na 0. Pogledamo v wireless LAN management frame, odpremo SSID parametre in vidimo kanal 6.

**Pri link state routing, ali vsak usmrejvalnik pozna zakasnitve vseh ostalih povezav v omrežju.** Ne, zanima nas samo ali imamo eno povezavo pokonci ali ne.

**Ali sta FDMA in TDMA nekolozijska protokola?** Pri tem je tako, da ima oboje za opravit s mediji, torej time division ali frequency division. FDMA pomeni da za prenos podatkov po istem kanalu uporabljamo različne frekvence (radijske postaje). TDMA pa pomeni da ima vsak kompjuter svojo časovno rezino v kateri lahko komunicira. Recimo ¼ sekunde lahko komunicira prvi računalnik in 2/4 sekunde drugi itd itd (uporabljajo ga kabelska omrežja z APS stream pasovno širino)

**Ko golob ostari, ga nadomestino z žičo popolnoma dvosmerno povezavo. Koliko mora biti najmanj kapaciteta te povezave v sako smer, da se podatki ne bojo prenašali počasneje kot z golobom. Zdej leti 106.6MB/s v eno smet, ali moramo to zdej delit z 2, ker bo dvosmerna?**

Ne, ne delimo z 2, ker potem bi imeli v eno smet 53.3MB/s, kar je počasneje, torej mora biti najmanj 106.6MB/s v eno smer.

**Na kateri plasti leži protokol ARP in zakaj?** Arp leži nad med povezavno in omrežno, ker jih pravzaprav povezuje med sabo.

**Kaj je odgovor na ARP poizvedbo?** Odgovor na ARP poizvedbo je ARP paketek v katerem je izvorni MAC in iskani MAC iz prvega paketka ARP, torej ko pošljemo poizvedbo, če ne vemo ponornega mac naslova, nazaj dobimo vse.

**Ali dobimo error, če naredimo ARP poizvedbo za naslov, ki je nekje v internetu?** Ne, ker mi naredimo ARP poizvedbo routerja na našem računalniku, šele router potem dela arp pozvedbo naprej.. lahko bo kvečjemo timeout od routerja..

**4. naloga kolokvij 24.11.2009**

a) mrežna kartica identificira naslov na povezavni plasti, prvi trije bajti, ki označujejo naslov proizvajalca in naslednji trije, ki označujejo naslov naprave 12:34:56|aa:bb:cc

b) Gre za 802.3 in Ethernet 2, razlika je v Length in Type, Length ima 802.3 Ethernet, Type pa Ethernet 2.

**Kako deluje NAT in PAT?** NAT – network address translation – spreminjanje naslova omrežne plasti (preslikava iz zasebnega v javni IP). Nat se uporablja, ko želimo recimo pri ARP pingati nek zunanji računalnik z drugim IP naslovom, nam NAT spremeni naš routerjev naslov 192.168.1.1 v nek poljuben IP naslov recimo 88.200.23.10, da tisti zunanji računalnik ne vidi da imamo mi ubistvu lokalno omrežje. Problem nastopi, ker če želi zunanji računalnik poslati podatke na naš naslov in nam z NAT prevede naslov 192.168.1.1 v nek drug naslov, ne moremo vedt koliko imamo drugih računalnikov v tem omrežju, zato pa potrebujemo PAT – port address translation, da vsakemu računalniku v omrežju določimo port. To uporabljamo, ki mamo lokalno omrežje v nekem podjetju in želimo ga povezati z zunanjim omrežjem. Tukaj uporabimo NAT+PAT.

**Kaj je dvojni sklad ipv4 in ipv6 in kako deluje?** To pomeni samo to, da imamo doma skonfiguriran ipv4 in ipv6 in oba delujeta normalno brez problema drug od drugega. Torej če pingamo ipv6 naslov, gre zadeva čez ipv6 sklad, če pa ipv4 pa čez ipv4. Privzeto pa če se le da, uporabi računalnik ipv6.

**Ali lahko nek zunanji proces naslovi nek računalnik v lokalnem omrežju, ki je za NAT routerjem?** Lahko samo v primeru, da smo forwardirali port na routerju na naš ip naslov. Torej ko pride zahteva po nekem portu na naš router, nam to zahtevo takoj preusmeri na določen IP v port forwarding opciji na routerju.

**Razlika med delovanjem datagramskega omrežja in omrežja z virtualnimi zvezami. Na kateri plasti je razlika?** Razlika je na omrežni plasti, namreč datagramsko omrežje deluje tako, da ko router dobi paketek 1, ga pošlje, dobi paketek 2 in že ne ve več in ga niti ne zanima, kam je poslal paketek 1 ampak pošlje paketek 2 čistno na novo, neglede na paketek 1. Pri virualnim zvezami, pa se vzpostavi neka virtualna povezava, po kateri se pošiljajo paketki in ni nikoli nobenega dvoma, kam je potrebno kaj poslati oz po kateri povezavi.

**Link state routing – Distance vector routing razlika v enem stavku?** Gre za isto porazdeljeno usmerjanje, razlika je samo to, da imamo samo stanje povezav ali pa metriko povezav. Se pravi, tudi odgovor.

**Glavni trije cilji za razvoj IPv6?** Večji naslovni prostor, hierarhični naslovi, torej IP-ji niso razmetani usepovprek ampak urejeni in poenostavljeno/hitrejše usmerjanje (poenostavljena glava), varnost IPsec.

**Kako izvedeš ARP napada, da vsi v omrežju (broadcast) odgovorijo samo enemu(ARP storm)?** Ponaredimo src mac naslov in naredimo več arp zahtev, ki jih pošljemo, da potem vsi odgovarjajo temu računalniku ponarejenem mac naslovu. Danes to zelo težko narediti.

**KOLOKVIJ 29.3.2011 1.) naloga**

FIZIČNA, POVEZAVNA, OMREŽNA, TRANSPORTNA, APLIKACIJSKA

Fizična ima zares fizično povezavo, recimo kakšne kable imamo vmes ali wireless.

Povezavna ima neko ethernet povezavo, kjer nas ne zanima kaj imamo spodaj. Tukaj nas zanimajo samo strojni naslovi ki se pošilja nekaj med njimi. Nas ne zanima kako se to prenaša, ali po kablu ali wifi itd.

Omrežna plast nas spet ne zanima kaj imamo spodaj, ampak tukaj uporabljamo samo IP naslove.

Na transportni povezavi spet logična povezava, komunicirata TCP in UDP med sabo.

Aplikacijska plast spet logična povezava med dvema aplikacijama, kjer komunicirata med sabo samo 2 aplikaciji. Torej nas spet ne zanima kaj je nižje.

**Denimo, da na FRI pride napadalec, ki na WC-ju skrivaj postavi lastno in povsem odprto brezžično dostopno točko, ki omogoča dostop do interneta prek napadalčeve omrežne hrbtenice. SSID te točke je LRK-FRI. Kako vam lahko škodi**

Lahko prestreže naša gesla, napad na SSL, vidi kompleten naš promet. Prestreže lahko naše komunikacije.

**Kaj je fizična plast, prenosni kanal in konektorji ali omrežna kartica? Vse skupaj?** Da, fizična plast je vse to skupaj.

**Kje tiči glavni razlog za opuščanje ARP v IPv6?** ARP ima kar nekaj varnostnih pomankljivosti. Glavna pomankljivost je ARP spoofing napad, ki ga je preenostavno izvest.

**Naloga prvih treh plasti je?** Prenos paketov po omrežju. Naloga fizične je prenos bitov po mediju, povezavne je prenos okvirjev po omrežju, naloga omrežne plasti je usmerjanje in prenos paketov po omrežju, naloga transportne pa zanesljiv prenos podatkov po omrežju od enga računalnika do drugega.

**Kam spadajo protokoli ki smo jih obravnavali na vajah?**

*POVEZAVNA:* Ethernet 2, Ethernet 802.3, Ethernet 802.11

*OMREŽNA:* ARP (med povezavno pa omrežno), IPv4, ICMP v IPv4, IPv6, ICMP v IPv6, Kot del ICMPv6 neighbour discovery (enak ARP-u)

**Kako v IEEE 802.11 preprečujemo kolizije?** Z CSMA CA. Pošiljatelj pošlje RTS okvir, dostopna točka odgovori z CTS in potem pošljemo. Torej prašamo če lahko kej povemo, in dobimo dovoljenje (kot v razredu☺). Torej kolizij NI.

**Route poisoning, kaj je to?** To se uporablja če želimo preusmeriti promet ven iz našega routerja, tako da ne bo šel promet čez naš router. Naredimo tako, da povezave ne bo možno vzpostaviti in naj bo umaknjena iz njihovih usmerjevalnih tabel.

**Za kaj je zadolžena povezavna plast?** Povezavna plast popravlja in zaznava napake.

**Podobnost med ARP STORM in SMURF?** SMURF napad imamo na omrežni plasti in izkorišča IP broadcast. Ponaredimo IP naslov pošiljatelja. Recimo imamo nekih 30 računalnikov v omrežju. En računalnik želi narediti smurf napad na drug računalnik. Kot izvorni napad pošlje na broadcast naslov njegov IP naslov. Torej mu bo 30 računalnikov odgovoril na broadcast ping, torej če napadalec uspe generirat 500mb/s, bo napaden dubu 30x500mb/s. Dejansko tok na gigabit liniji ne bo dubu, sam je pa dost za kakšen denial of service.

Pri ARP storm, pa ponaredimo MAC naslov napadenega računalnika, arp zahteva se pošlje na ethernet broadcast naslov, zato bo odgovoril samo pravi računalnik in ne vseh trideset, torej napadalec generira 1gb/s, tako da napaden dobi tudi 1gb/s odgovorov. Je manj učinkovit napad.

Glavna podobnost je da oba izkoriščata pomankljivost broadcast naslova. ARP storm na ETHERNET-u, SMURF pa na IP-ju.

**V čem je prednost uporabe stikala namesto razdelilca?** Če uporabimo stikalo, nimamo kolizij. Torej povečamo prepustnost.

**Primerjajmo glavi datagrama IPv4 in IPv6?**

IPv6: verzija IPv4: verzija

IPv6: Traffic class IPv4: differentiated services field (zadolžen za quality of service – določanje prioritet prometa ki ga pošiljaš po omrežju.

IPv6: Flowlabel IPv4: ga nimamo (edino polje)

IPv6: Payload Length IPv4: Total length

IPv6: Next header: ICMP IPv4: Protocol: ICMP

IPv6: HOP LIMIT IPv4: TTL

IPv6: Destination in source IPv4: destination in source tudi

**Razlika med usmerjevalnikom in stikalom?** Stikalo dela na povezavni plasti. Usmerjevalnik pa na omrežni plasti – zna dejansko IP-je usmerjat, stikalo pa zna samo posredovat paketke glede na strojne naslove.

**Kaj rabmo vedit o Neighbour Discovery protocol?** Da se zahtevi reče Neighbour solicitation ki je podobno kot ARP zahteva, odgovoru pa neighbour advertisement, kot pri ARP odgovoru. To je del prot. ICMPv6 in je podobna zadeva ARP-u na IPv4.

**Kakšen je dejansko TCP/IP sklad?** Lahko imaš 4 ali 5 plasti. Pri 4 imaš fizično in povezavno skupaj in jima rečemo prenosni sistem, mi pa smo se dogovorili da jih bomo ločili.

**Povezavna plast gre do routerja in ne čez router, tako da o HOPIH govorimo v lokalnem omrežju, kar pomeni od ROUTER-ja do ROUTER-ja.**

**Kaj pomeni trailer v WS?** To uporabimo takrat, to je zapolnitev, takrat ko imamo premalo podatkov za ethernet okvir. Ima omejeno minimalno dolžino (vsaj 46 B podatkov). Če jih ni, jih dopolnimo do 46 B in temu se reče trailer.

**Med ethernet 2 in 802.3 ethernet je** možno ločevat preko LENGTH in TYPE vrstice, kjer če imamo več kot 1500 podatkov, torej več kot 0x600 šestnajstiško pomeni da gre za ethernet 2, če imamo pa manj kot 1500 podatkov pa gre za 802.3.

**Traceroute mi izpiše use usmerjevalnike od mojega kompa do kompa ki gaželim doseči. V CMD napišem tracert** [**www.arnes.si**](http://www.arnes.si) **recimo. Po čem spoznam traceroute?** Po ponavljanju u troje, ECHO (PING) in TIME TO LIVE EXCEED vemo da gre za traceroute programček. Traceroute gleda tudi time-to-live, ki se povečuje čim gremo čez kakšen router več. Torej ko gre čez prvi router, je ttl 1, ko gre čez drugega je ttl 2 in tako naprej.

**Kakšen je največji MTU, da se bo 2320 B dolg datagram fragmentiral na 3 datagrame?**

Začnemo tako, da vedno naredimo za enega manj, torej

2320 B dolg datagram

3 datagrame želimo narediti

2320 = 2300 B podatkov + 20 B glave IP

2300 B podatkov / 2 = 1150 B na datagram

1150 / 8 (ker odmik je le večkratnik št. 8) = 143,75

144 \* 8 = 1152 B (vedno zaokrožimo navzgor in ponovno množimo z 8, da dobimo koliko največ število podatkov lahko damo not)

1. Paket:

Glava: 20 bajtov

* Nastavljena zastavica MF (more fragments)
* Odmik (offset) = 0

Podatki: 1152 B

1. Paket:

Glava: 20 bajtov

* Zastavica MF ni nastavljena, ker je zadnji datagram
* Odmik = 144

Podatkov: 2300 B – 1152 B = 1148 B

MTU = 20 B + 1152 B = 1172 B (torej glava + največje možno številko, koliko podatkov lah damo not)

Novi MTU = 1171 B, ker odštejemo 1B od prvega MTU-ja

1. Paket:

* Glava: 20 bajtov
* nastavljena MF
* odmik = 0

Podatkov: 143 \* 8 B = 1144 B (143 ker smo uzel 1 bajt, zadeva ni več deljiva z 8 in mormo uzet kar 8 paketkov, ne samo enega)

1. Paket:

* Glava: 20 bajtov
* MF nastavljena
* Odmik = 143

Podatkov: 2300 B – 1144 B = 1156 B 🡨 preveč podatkov za ta MTU, ker če mu prištejemo glavo pride 1176, kar je več od novega MTU-ja 1171. Zato naredimo sledeče:

143 \* 8 = 1144 B

1. Paket:

* Glava: 20 bajtov
* Nima nastavljene zastavice MF
* Odmik = 2 \* 143 = 286

Podatkov: 2300 B – 1144 B – 1144 B = 12 B

1171 je največji MTU ki ga še lahko uporabimo, da se mi bo fragmentiral v 3 datagrame.

**Aplikacijska plast: Zakaj pravimo, da je http protokol brez stanj? In Ali veste, za kakšen protokol aplikacijske plasti, ki ima stanja?** Brez stanj je zato, ker je zahteva in odgovor popolnoma ločeno eno od drugega, ne hrani nobene zgodovine sam protokol, ne zagotavlja nobene seje. Lahko to rešimo s cookiji, ki so namenjeni da se hranijo seje ali s sessni.

Protokol ki ohranja stanja je POP3, FTP… karkoli, kjer se autenticiramo.

**Zakaj je pomembna ocena rtt in zakaj ocena odmika?** Oceno odmika potrebuješ zato, da veš kdaj ti poteče časovna kontrola za nek segment.. če imaš počasen link, in pošiljaš 1mb/s ali pa 1gb/s je velika razlika, zato potrebuješ vedno rtt ki ga nujno rabiš za odmik, ki ti pa pove kakšen je ping med nami in prometom. Potrebuješ ga za izračun časovne kontrole.

**DES** je algoritem s simetrično kriptografijo. Notri se uporabljajo substitucije in permuntacije.

**Kaj je zastrupljanje DNS-ja in kako nam napadalec s tem lahko škodi?** Vedno ko hočemo na nek URL it, nam računalnik hrani zahtevo v DNS predpomnilniku, da ko enkrat iščemo to stran, bi blo neumno da bi jo zmer na novo.. Torej zastruplanje pomeni, da v nek dns predpomnilniški strežnik podamo napačen naslov, recimo namesto [www.microsoft.com](http://www.microsoft.com) IP-ja, ki je recimo 212.10.10.1. damo čisto enega drugega, torej bomo preusmerjeni namesto na microsoft, recimo na mac.com☺

**Kaj je birthsdayattack?** Gre za kolizijo zgoščevalnih funkcij. Iščemo sporočilo ki ima enak rezultat zgoščevalnih funkcij ampak drug pomen.

**Kje se prenašajo podatki pri prenosu z metodo POST?** Prenašajo se v telesu sporočilu in ne v naslovni vrstici(GET metoda)

**2. kolokvij 2.6.2010 7. Naloga**

Pošljjemo 4 http zahteve, ker imamo eno za HTML in 3 ostale objekte (JPEG sličice)

1. 4 tcp povezave, ker imamo vzporedne povezave, ki bojo trajale t-časa. Trosmerno rokovanje - En rtt rabimo za TCP SYN in SYN UP, drug rtt in potrditev nazaj, čas za prenos datoteke rtt in rušenje povezave (ni potrebno upoštevat). Vedno najprej prenesemo http, da ugotovimo da imamo notri še 3 slike, nato šele slike.
2. Torej rabimo za usak objekt rabmo svojo tcp povezavo, torej imamo zaporedne prenose. Traja za usako povezav 2xrtt za trosmerno rokovanje, čas T za prenos in opcijsko 2x rtt za zaključek, torej za rušenje povezave… na kratko, za 4x toliko.
3. Trajna povezava, torej use prenašamo po eni sami povezavi, to pomeni 4 prenose zaporedno, 2x rtt za upostavljanje, na koncu 2x rtt za rušenje povezave, vmes pa 3x t za prenos.

**Kaj je MIME in kaj se uporablja?** Multipurpose mail extension se uporablja za kodiranje priponk in mednarodnih znakov v elektronski pošti. Pravzaprav določa kako se to kodira. Uporablja se na aplikacijski plasti. Poznamo 7-bitno kodiranje besedila, quoted printable, base-64 kodiranje… Izvršljivo datoteko kodiramo z base-64 načinom kodiranja, ker email omogoča prenos samo 7-bitnih ascii znakov. Email ima tudi posebno zgradbo, recimo pika ».« v emailu pomeni konec prenosa, zato moramo priponke kodirat in pazit na piko. TXT datoteko bi kodiral s quoted printable ali base-64. Za besedila ponavadi quoted, za binarne datoteke pa base-64.

**Za kakšen namen in v kakšnih sistemih se uporablja Needham-Schroeder-jev protokol?**

Uporablja se ko imamo centralno pariteto, da se uporabniki lahko avtenticirajo z žetončki.

**TCP oddajnik s širino okna 4000 bytov odda segmente z zaporednimi številkami 3000, 4000, 5000 in 6000. Nato sprejme potrditev ACK 6000. Katere od naslednjih trditev so resnične in katere ne ? Če bo naslednja potrditev imela vrednost Receive window nastavljeno na več kot 1000, lahko oddamo naslednji segment**(Da) **b. Po prejemu naslednje potrditve (ACK 7000) se bo zamašitveno okno povečalo**(DA) **c. Max. velikost segmenta je najverjetneje 1000 bytov**. (DA) **d.Če bo naslednja potrditev ACK 6000, pomeni da prejemnik dobil segment 6000 podvojen**(NE) **e.Če bo naslednja potrditev ACK 6000, pomeni da je bilo s segmentom 6000 najbrž nekaj narobe**.(DA) **f.Če bo naslednja potrditev imela vrednost Receive window nastavljeno na več kot 1000, lahko oddamo naslednji segment.**(DA) **g.Če bo namesto naslednje potrditve zazvonil alarm časovne kontrole za segment 6000, je to lažni alarm in segmenta ni treba ponovno oddati, saj smo že 2x dobili njegovo potrditev.** Ne drži, ker ACK6000 mi segmenta 6000 še ne potrjuje, tako da ga bom moral poslati še enkrat, če se mi bo iztekla časovna kontrola za ta segment

**Kakšne težave srečamo pri aktivnem načinu prenosa podatkov pri FTP-ju?**

Odjemalec 🡺 kontrolna povezava 🡺 strežnik

🡸 podatkovna 🡸

Ker strežnik pošilja podatke nazaj, in ker smo na routerju za NAT-om, strežnik ne bo moral poslati podatkov nazaj.

Rešimo problem s pasivnim FTP-jom, kjer odjemalec pošilja podatke.

Odjemalec 🡺 kontrolna povezava 🡺 strežnik

🡺 podatkovna 🡺

**Needham – Schroeder** protokol se uporablja pri kerberosu. Deluje tako, da ga uporabimo taklrat ko imamo 2 entiteti, ko druga drugi ne zaupata, ampak zaupata obe nekem centru za distribucijo ključa. To je v npr. nek strežnik, active directory domena. Ko nekdo da nekaj v skupno rabo, prašamo active directory za nek žetonček, ga dobimo za en računalnik, s tem žetonom gremo do druzga računalnika, nato drug računalnik pri active directory preveri ali je pravi in lahko sheramo. Drugi je pa public key protocol varjanta, kjer rabimo samo skupaj zaupat nekem tretjem strežniku.

**Kako vemo koliko zahtev HTTP bo poslal naš brskalnik, da se bo stran naložila v celoti? Kaj v bistvu moramo gledati?**

ZA HTML je ena, SRC ali CSS moramo gledat. Href ni potrebno ker se naloži šele ko kliknemo, torej vse kar ni potrebno kliknit in se naloži preko SRC ponavadi.

**Ali sta ukaz MAIL FROM: in vrstica From: v glavi sporočila povezana?** Ne, to sta različni zadevi, kar lahko pridno izkoriščamo za ponarejanje sporočil. Vseeno je kaj v FROM napišemo, ni potrebno isto napisat kot v MAIL FROM, saj epoštni strežniki tega ne preverjajo. (zaenkrat☺)

**Napad z zrcaljenjem (reflection)?** Omenjamo s protokoli za avtentikacijo, challenge response. Rečemo uporabniku, daj mi geslo, preprosta stvar. Bolj zakomplicirano je pa, ko strežnik ve našo geslo in nam pošlje nek izziv. Torej moramo z geslom in izzivom nekaj poračunati, poslati nazaj strežniku, ker ima strežnik že naše geslo, on poračuna isto stvar in če je rezultat enak, je OK in se avtenticiramo. Problem je, če nam strežnik pošlje vedno isti izziv, lahko posnamemo odgovor in ga pošljemo strežniku, brez da bi vedli kaj je dejansko geslo, to je ta problem zrcaljenja.

**Napad z vrivanjem (man in the middle)** Je napad pri katerem se vrinemo med pošiljatelja in prejemnika. Najbolj znan je ARP poisoning na povezavni plasti. Lahko naredimo to tako da imamoprotokol DHCP, se mi pretvarjamo da smo DHCP strežnik, morebitnem kompu ki zahteva DHCP odgovor podtaknemo IP in naj routa promet preko mene.

**Kdaj pride v poštev receive window?** Tukaj je tako, da ko imamo nalogo, da upoštevamo tudi receive window, ga moramo upoštevati. Vedno lahko pošljemo maximalno najmanjšo vrednost med receive in conguestion window, torej če je conguestion manjši od receive, lahko pošljemo maksimalno tok, in čeje receive manjši, maksimalno kolikor je receive. Fora je da se conguestion window dinamično zmanjšuje ali povečuje, receive je pa odvisen od sprejemnika in kako ga on nastavi.

**Razloži TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA** TLS pomeni transfer layer security, pove da gre za SSL v3.1

RSA algoritem je za kriptografijo z javnim ključem in mi omgoča izmenjavo tajnega sejnega ključa znotraj SSL-a

AES256 je protokol za simetrično kriptografijo, za šifriranje podatkov med sejo oz. med računalnikoma. 256 je velikost ljuča.

CBC cyber block chain, metoda za veriženje blokov, kar pomeni da bomo pri šifriranju naslednjega bloka uporabili kot ključ rezultat prejšnjega bloka šifriranja. SHA je zgoščevalna funkcija ki jo uporabljamo za preverjanje integritete bitov.

**Denimo, da ima racunalnik C odprt UDP vtič(socket) s stevilko vrat 6565. Naj računalnik A in B posljeta C-ju vsak en UDP paket s ciljno stevilko vrat 6565. Prideta na isti vtic? C nato kot odgovor pošlje A-ju nazaj nek UDP paket: kakšno izvorno in ciljno števliko mu bo dal?** Načeloma je tako, vtič je določen z izvornimi vrati, ponornimi vrati, izvornim ipjem in ponornim ipjem. Vsak računalnik ki je paket poslal je uporabil svoj izvorni ip in vrata, torej ne moreta pridt na isti vtič.

Dal mu bo ravno obratno kot je poslal.

**Kako je določen TCP vtič in UDP vtič?** Določen je z izvornim IP-jem in vrati, ter ponornim IP in vrati. Za UDP načeloma ne rabimo izvornih vrat, vendar so ponavadi vedno napisana.

**Strežniku SMTP, POP3/IMAP**, Imenik uporabnikov in spletni strežnik. Dodali bi tudi bazo poštnih predalov. SMTP skrbi za sprejem pošte z interneta, pop3 nudita pot za dostop do poštnih predalov za uporabnike, imenik uporabnikov hrani uporabniške račune, spletni strežnik hrani spletni dostop do naših računov, epoštni predali pa so shranjeni epoštni predali.

**Kaj imata skupnega POP3 in IMAP?** Skupnega imata to da sta oba protokola za elektronsko pošto. Imap je za to, da imamo sporočila vedno shranjena na strežniku, pop3 pa vedno downloadamo sporočila.

**Navedite tri od petih metod, ki jih uporabljamo v http zahtevi po http 1.1(**GET, POST, PUT, DELETE)

**V katero od plasti sodijo:**

rušenje virtualne zveze(**omrežna***),*prenos datoteke iz omrežja torrent(**aplikacijska),** kriptiranje(**aplikacijska)***,* izogibanje kolizijam (**povezavna***),* enkapsulacija in dekapsulacija(**omrežna***),* EIGRP (**omrežna***),* prenos okvirja(**povezavna***),* avtentikacijski protokoli(**aplikacijska***),* ping(**omrežna***),* Overjanje z digitalnimi potrdili(**aplikacijska***),* UMTS(**fizična***)*

**opišite nadzor zamašitve v TCP ter izboljšave boljšav v različnih TCP izvedb?**

Slowstart, conguestion avoidence itd.

IZBOLJŠAVE:

TCP Reno, TCP Tahoe, TCP Vegas izboljšave. Kaj naredimo ko dobimo trikrat dvojno potrditev?

TCP Tahoe začnemo z začetkom s fazo slowstart.

TCP Reno začnemo od praga linearno naprej in popravimo samo prag.

TCP Vegas gleda bolj natančno RTT-je in s formulo sproti uravnava kok hitri so timeout-i.

**Zakaj pravimo, da osnovni princip nadzora zamašitev ni primerne za današnja hitra omrežja?** Zato ker SLOWSTART predolgo traja in ta nadzor zamašitev ima žagasto krivuljo (smrekcasto), tako da bi danes rabil boljši mehanizem kot trikrat dvojno potrditev. S TCP-jem danes zelo težko popolnoma izkoristmo zelo hitre povezave.

**Primerjajte sejo SMTP in FTP?** SMTP se ne avtenticiramo, FTP se avtenticiramo, FTP ma 2 tcp povezavi, ločeno za podatke in kontrolno povezavo, SMTP ima samo eno, SMTP je totalno besedilni protokol, uporabljamo BASE-64 kodiranje za binarne datoteke, pri FTP pa lahko prenašamo tudi binarne datoteke.

**Tri lastnosti dobre HASH funkcije?** Cim manj kolizij, enakomerna porazdelitev, majhna sprememba spremeni »cel« hash.

**Smo za DHCP starvation rekli da je to problem ko DHCPju zmanjka IPjev?** Drži.