SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE VARAŽDIN

Dobrica Pavlinušić

Održavanje Internet Servera

DIPLOMSKI RAD

Varaždin, 1997.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE VARAŽDIN

Dobrica Pavlinušić
redoviti student
Broj indeksa: 29173/92-R
Smjer: Informacijski sustavi
VII/1 stupanj

Održavanje Internet Servera

DIPLOMSKI RAD

Voditelj rada:
Mr.sc. Antun Brumnić, viši predavač

Varaždin, travanj 1997.

Sadržaj:

1. Uvod

Tema ovoga rada je instalacija i održavanje Internet poslužitelja. Razmatrat će se samo UNIX poslužitelji. Postavlja se pitanje nije li to previše usko područje interesa, danas kada mnoge kompanije naglašavaju da baš njihovi poslužitelji mogu najbolje zadovoljiti zahtjeve mrežnih servisa kao što su to WWW ili FTP. Međutim, treba imati u vidu da takvi operativni sustavi nisu zapravo srasli sa TCP/IP protokolima, podrškom za mrežu i principima klijent/server arhitekture kao što je to UNIX. Ako se za Internet poslužitelj uzme neki operativni sustav koji nije prvobitno predviđen za tu funkciju, još uvijek je potrebno imati neko Unix računalo za servise koji su važni za kompletnu realizaciju povezivanja lokalne mreže na Internet, kao što je to na primjer sustav imenovanja (DNS) o kojemu će biti riječi.

Prvo će se razmatrati osnovame UNIX operativnog sustava, čija je duga prošlost prikazana u dodatku ovom radu. Nakon toga ćemo se upoznati sa mrežnim protokolima koji se koriste na Internetu, a koji su sastavni dio svakog UNIX računala. Mrežni protokoli su samo jedna strana medalje, potrebno je upoznati i upravljanje mrežom, što će biti obrađeno u petom poglavlju. Poseban naglasak ću staviti na elektroničku poštu, koja je po nekim istraživanjima, najviše primjenjivan protokol na Internetu. Sigurnosni problemi pri uvođenju novih stanica, što je tematika kojom će se svaki administrator mreže morati pozabaviti prije ili kasnije, tema su sedmog poglavlja, da bismo prešli na vatrozide koji omogućuju da se interna mreža organizacije (također poznata kao *intranet*) odijeli od javne mreže Interneta. Promotriti ćemo kako je to riješeno u sklopu Fakulteta Organizacije i Informatike u Varaždinu. Nakon toga pozabaviti ćemo se imenovanjem računala, što je također tema koja ne smije biti zaobiđena.

Ovaj rad ima za cilj prikazati mogućnosti Internet poslužitelja zasnovanog na operativnom sustavu Unix i poslužiti kao pomoć pri postavljanju i održavanju Internet poslužitelja. Pisan je dovoljno općenito da se može primijeniti na skoro svim varijantama ili verzijama UNIX-a. Međutim, dani primjeri odnose se većinom na verzije Unix-a koje se koriste na Fakultetu Organizacije i Informatike (OSF/1 na računalu barok.foi.hr i Linux na računalu firewall.foi.hr), što je i posebno naznačeno.

Nakon svakog poglavlja dan je pregled literature u kojoj je moguće naći dodatne informacije o temi obrađenoj u poglavlju. Treba međutim naglasiti da to nikako nije jedini mogući izbor, jer

je literatura iz svih područja vezanih za Internet veoma brojna, i većinom dostupna putem samog Interneta.

2. Osnovni principi UNIX operativnog sustava

Velika je prednost kada iskustvo stečeno višegodišnjim radom na nekom računalu može pomoći, ili još bolje u potpunosti se prenijeti, na neko drugo računalo na kojemu treba izvršiti neki zadatak.

2.1 Prenosivi operativni sustav

Jedan od osnovnih zadataka koji su si tvorci UNIX operativnog sustava postavili je bio stvoriti operativni sustav koji bi bio konzistentan na različitim hardverskim platformama. To znači da on mora biti prenosiv (portabilan) na različite hardverske platforme. Da bi se shvatilo kako UNIX funkcionira prvo se moraju razumjeti principi na kojima se on temelji.

Potrebno je objasniti što znači pojam prenosiv kada se on primjenjuje na operacijske sustave. Prenosiv u smislu računalskog programa znači da se program može izvršavati na različitim računalima bez promjene ili sa malom promjenom izvornog koda programa.

Međutim, kada se poveže prenosivnost i operativni sustav, stvari se donekle mijenjaju. Operativni sustav mora poznavati mnoge stvari o samoj hardverskoj strani računala kao što je to procesor, oblik sabirnice i ostale specifičnosti. Prema tome ne postoji mogućnost da se operativni sustav prenese na neku drugu mašinu "bez promjene ili sa malom promjenom izvornog koda". Prenosivnost operativnog sustava u biti znači da je on napisan na taj način da njegovo prenošenje na drugi sustav zahtijeva najmanje moguće promjene.

2.2 Pretpostavke o računalu

Da bi se smanjile promjene potrebne pri prenošenju UNIX se ograđuje od različitih specifičnih mogućnosti koje podržava neki hardver. Zapravo što je UNIX jednostavniji (engleski izraz je *generic*) i manje ovisan o nekim karakteristikama hardvera to je prenošenje jednostavnije.

Kao rezultat potrebe da bude što jednostavniji UNIX pretpostavlja da se izvršava na vrlo jednostavnom računalu koje podržava samo osnovne mogućnosti. Ideja je da svako računalo koje podržava te osnovne mogućnosti može podržavati UNIX.

Osnovne pretpostavke UNIX-a su vrlo jednostavne: računalo se sastoji od dva osnovna dijela: procesa (*processes*) i podataka (*data*). Proces je program koji se trenutno izvršava dok su podaci sve ostalo.

2.2.1 Procesi (processes)

UNIX je višeprocesni operativni sustav (*multiprocessing operating system*). To znači da se više procesa može izvršavati istovremeno. Većina računala, međutim, imaju samo jedan procesor (CPU) iako postoje neka koja ih imaju više. Jedan procesor može izvršavati samo jedan proces istovremeno. Računalo sa tri procesora bi, prema tome, moglo izvršavati tri procesa. Da bi riješio taj problem UNIX upotrebljava princip podjele vremena (*time-sharing*). Operativni sustav pamti koje sve procese treba izvršiti i svakome od njih dodjeljuje djelić vremena procesora. Ako postoji više procesora, ima i više procesorskog vremena za dijeljenje.

UNIX definira proces kao rezultat izvođenja programa. Ako se isti program pokrene dva puta, on se izvršava kao dva procesa. Sve što se događa je rezultat procesa.

Da bismo nešto nazvali računalom, ono mora imati mogućnost izvođenja procesa. Međutim, računala podržavaju i različite dodatke kao što su to diskovi (tvrdi diskovi, diskete različitih formata, compact diskovi), trake (od 1/4 inča ili u kertridžima), terminale, modeme, miševe, mreže i različite druge ulazne, izlazne i odredišne uređaje. Kako UNIX može upravljati svakim od tih specifičnih uređaja sa specifičnim svojstvima i istovremeno ostati prenostiv ? On jednostavno ignorira specifična svojstva uređaja.

2.2.2 Podaci (data)

Tvorci UNIX-a su zacrtali samo osnove što se tiče podataka koje UNIX može obrađivati. Da bi generalizirali način na koji se upravlja podacima svi podaci na UNIX-u reprezentiraju se tokom znakova (*stream of characters*). Prema tome bez obzira kako se podaci stvaraju ili gdje se spremaju u UNIX sustavu se oni obrađuju znak po znak.

Zbog toga što su svi podaci u istom obliku UNIX ne mora brinuti o detaljima pohranjivanja podataka. Nevažno je koji je uređaj stvorio podatke sve dok su oni u obliku toka znakova.

Jednako je i za izlaze na uređaje. UNIX uvijek stvara tokove znakova bez obzira na to kojem su uređaju upućeni.

2.2.3 Uređaji (devices)

UNIX također pretpostavlja da su svi izvori i odredišta podataka (npr. uređaji, *devices*) dijelovi računalskog datotečnog sustava (*file system*). Svi uređaji koji su povezani na UNIX su predstavljeni kao datoteke. Bez obzira na različite hardverske konfiguracije UNIX-u one izgledaju isto - kao datoteke koje mogu slati ili primati tokove znakova trenutno aktivnih procesa.

Zbog toga što su svi uređaji predstavljeni kao datoteke, svi programi koji mogu čitati tok znakova mogu čitati bilo koji uređaj ili bilo koji drugi tok znakova (ukoliko je to omogućeno kontrolom pristupa).

2.3 UNIX Procesi

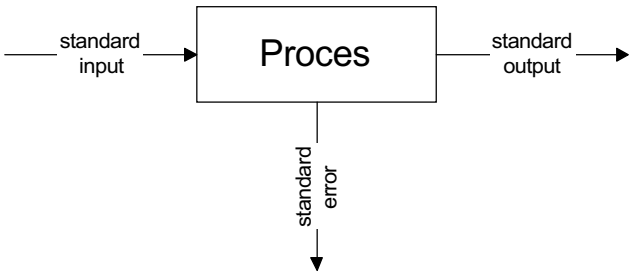
UNIX proces se stvara kada se izvršnoj datoteci dodijeli tok podataka i pokrene izvršavanje, obično kao rezultat naredbe koju je zadao korisnik, iako UNIX može i sam pokretati neke procese. Proces je neovisan o programu koji izvršava i podacima koje obrađuje. Drugim riječima, proces može izvršavati bilo koji program koji je povezan sa bilo kojim tokom podataka u sustavu.

Važno je naglasiti da proces nije program. Proces je okolina u kojoj se program izvodi. Dijelovi te okoline su i tokovi podataka sa kojima je program povezan. Proces je dinamičan i postoji samo u trenutku izvršavanja, a stvara se pokretanjem izvršnog programa kojem su dodijeljeni tokovi podataka.

Proces može na dva načina povezivati tokove i izvršne programe. Jedan je direktan pristup prema imenu navedenom unutar procesa, dok je drugi upotreba podrazumijevanih tokova (*default streams*).

2.3.1 Podrazumijevani tokovi (default streams)

Svaki UNIX proces je u trenutku kreiranja povezan sa tri osnovna toka: standardni ulaz (*standard input*), standardni izlaz (*standard output*) i standardne greške (*standard error*).



Slika 1: Podrazumijevani standardni tokovi svakog procesa

Ukoliko se to ne promijeni, sva tri standardna toka povezani su sa terminalom koji je dodijeljen procesu.

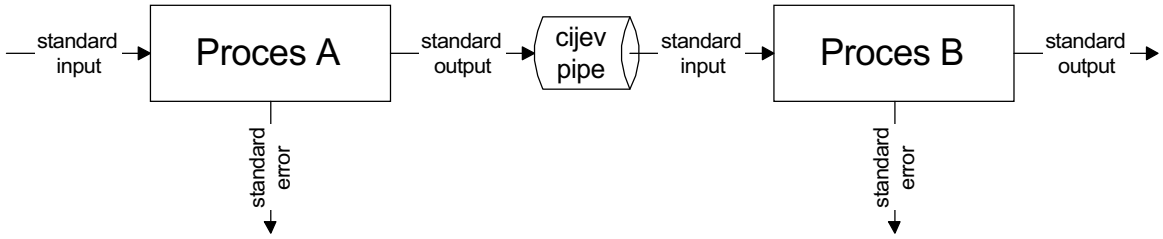
Standardni ulazni tok također poznat kao standard input ili *stdin* je tok iz kojega program čita podatke ako se drugačije ne navede. Standardni izlazni tok (standard output ili *stdout*) je tok u koji proces šalje podatke ako se ne navede druga datoteka. Standardni tok grešaka (standard error ili *stderr*) je tok u koji program upisuje poruke o greškama ukoliko program ne definira vlastitu datoteku za bilježenje grešaka. *stderr* tok se upotrebljava da bi se greške razdvojile od izlaznih podataka iz programa.

Po podrazumijevanim (*default*) vrijednostima UNIX komande čitaju sa tastature i pišu na zaslon terminala. Upotreba standardnih ulaza, izlaza i grešaka je moćan alat kojim se može posebno upravljati redirekcijom.

Kako su procesi nezavisni od načina na koji su tokovi podataka koje upotrebljavaju kreirani, UNIX može stvarati tokove podataka i na treći način: od drugih procesa.

2.4 Cjevovodi (pipelines)

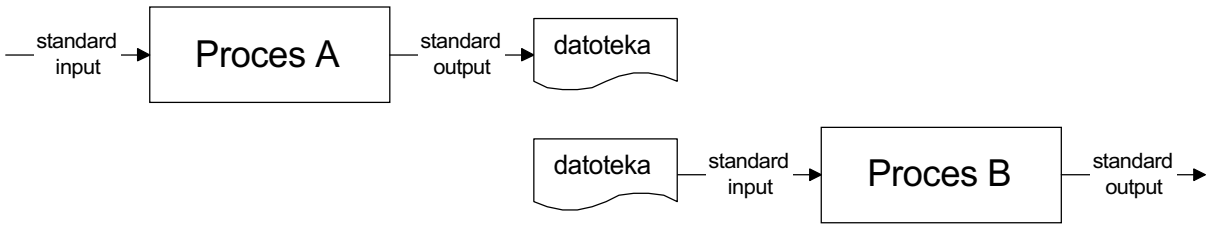
Multiprocesne mogućnosti UNIX-a se u potpunosti iskorištavaju kada se izlaz iz jednog procesa poveže sa ulazom u drugi proces. Takve se veze nazivaju cjevovodi (*pipeline*).



Slika 2: Cjevovod od procesa A do procesa B

Standardni tok grešaka (*stderr*) i dalje postoji, i može se preusmjeriti za jedan ili oba procesa.

Cjevovodi se mogu zamijeniti spremanjem izlaza procesa A u datoteku i upotrebljavanjem iste datoteke kao ulaza u proces B.



Slika 3: Zamjena cjevovoda korištenjem privremene datoteke

Međutim, upotreba cjevovoda je svrsishodnija jer ne zahtijeva spremanje u datoteku i omogućava istovremeno izvršavanje procesa A i B. Cjevovodi su važan alat kod kreiranja aplikacija za UNIX.

2.5 Literatura:

1. Ray Swartz: UNIX Applications Programming: Mastering the Shell, SAMS, Carmel, Indiana, USA, 1994.
2. Richard W. Stevens: Advanced Programming in the UNIX environment, Addison-Wesley Publishing Company, 1992.

3. Teorija TCP/IP komunikacijskih protokola

TCP/IP je skraćenica za Transmission Control Protocol/Internet Protocol skup protokola koji podržavaju Internet, najveću mrežu na svijetu. Internet je zapravo višeprotokolna mreža, jer osim TCP/IP-a podržava i različite druge protokole. Međutim, najviše zastupljen protokol na Internetu je upravo TCP/IP.

TCP/IP omogućava oblikovanje mreža kao što je to Internet, podržavajući iste servise bazirane na različitim protokolima i fizičkom hardveru koji se nalazi ispod TCP/IP-a. Glavni protokol je Internet Protocol (IP) koji podržava jedinstveni adresni prostor i usmjeravanje (*routing*) paketa po cijeloj mreži. Transimission Control Protocol (TCP) podržava pouzdanu isporuku paketa preko IP-a. Drugi protokol User Datagram Protocol (UDP) prenosi pakete bez podrške za potvrdu ispravnog prijenosa. I TCP i UDP zahtijevaju Internet Protocol (IP) i svi ostali TCP/IP protokoli kao što su telnet, ftp, gopher i se drugi baziraju na njemu.

3.1 Povijest

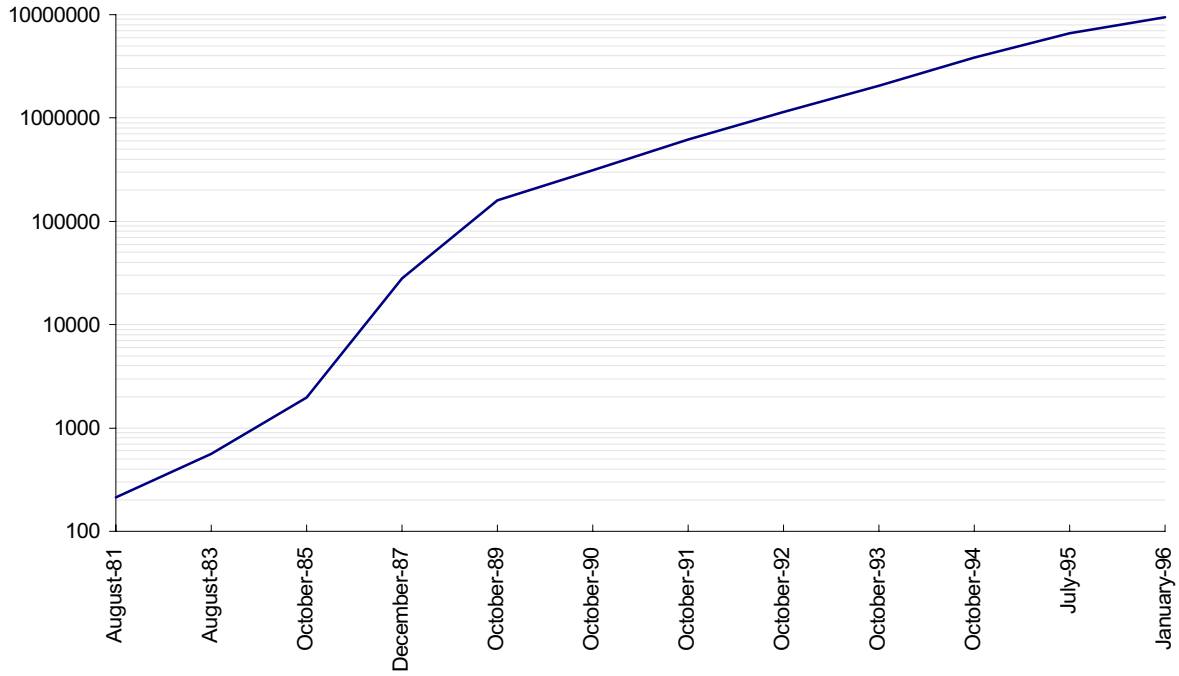
Rani počeci Interneta se nalaze u ARPANET-u koji je postojao od 1969. do 1990. ARPA je akronim od Advanced Research Project Agency Američkog ministarstva obrane (*Department od Defence - DoD*). ARPANET je kreiran kao eksperimentalna računarska mreža orjentirana na prijenos i usmjeravanje paketa. ARPANET je osnovna (*backbone*) mreža za Internet ali te dvije mreže nisu identične. Ona je prestala postojati 1990. kada su je zamijenile druge mrežne tehnologije kojima je i sama pomogla u razvoju.

Detaljnije informacije o ARPANET-u mogu se naći u mnoštvu literature koja je na raspolaganju.

Popularnost TCP/IP-a možemo zahvaliti implementaciji u 4.2BSD verziji operacijskog sustava UNIX koji je razvio Computer Systems Research Group (CSRG) Kalifornijskog sveučilišta Berkeley (University of California at Berkeley). Zbog toga što se distribuirao po cijeni same distribucije i to u izvornom kodu, ubrzo je postao vrlo popularan na raznim jeftinim mikroprocesorima kao što su to Motorolini 680x0 i Intelovi 80x86. Mnoge kompanije su

iskoristile nove i moćne procesore i operativni sustav da bi stvorili moćne radne stanice koje su doprinijele popularnosti TCP/IP-a.

Rast Interneta može se zorno prikazati sa slijedeća dva grafa. Prvi prikazuje porast broja host-ova na Internetu, dok drugi pokazuje porast broja mreža povezanih na Internet.

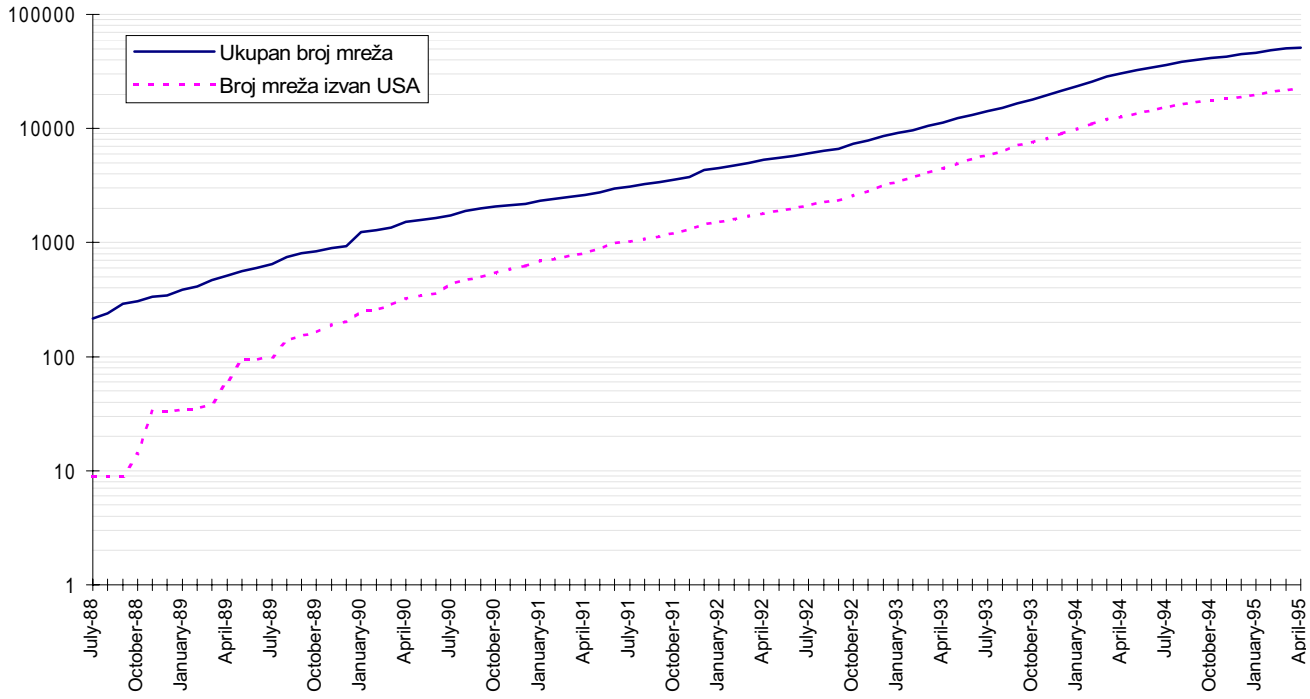


Slika 4: Porast broja hostova povezanih u Internet[1]

Treba naglasiti da su na oba grafa upotrijebljena logaritamska mjerila. Kao što je jasno vidi, razvoj Interneta je vrlo dinamičan i ne pokazuje znakove posustajanja.[2]

[1] Izvor: ftp://nic.merit.edu/nsfnet/statistics/history.hosts

[2] Treba naglasiti da je većina podataka u ovom radu preuzeta iz izvora u USA, jednostavno zato što su podaci iz USA lako dostupni. Međutim, zanteresirane za stanje u Europi mogu uputiti na RIPE (European IP Networks) http://www.ripe.net.



Slika 5: Porast broja mreža povezanih na Internet[3]

3.2 TCP/IP servisi

Za razumijevanje TCP/IP servisa prvo je potrebno razumjeti sam TCP/IP kao i klijent/server (*client/server*) model koji primjenjuju ti servisi.

3.3 Klijent/Server model

Klijent/Server model se upotrebljava pri implementaciji Internet servisa. On se može podijeliti na tri dijela:

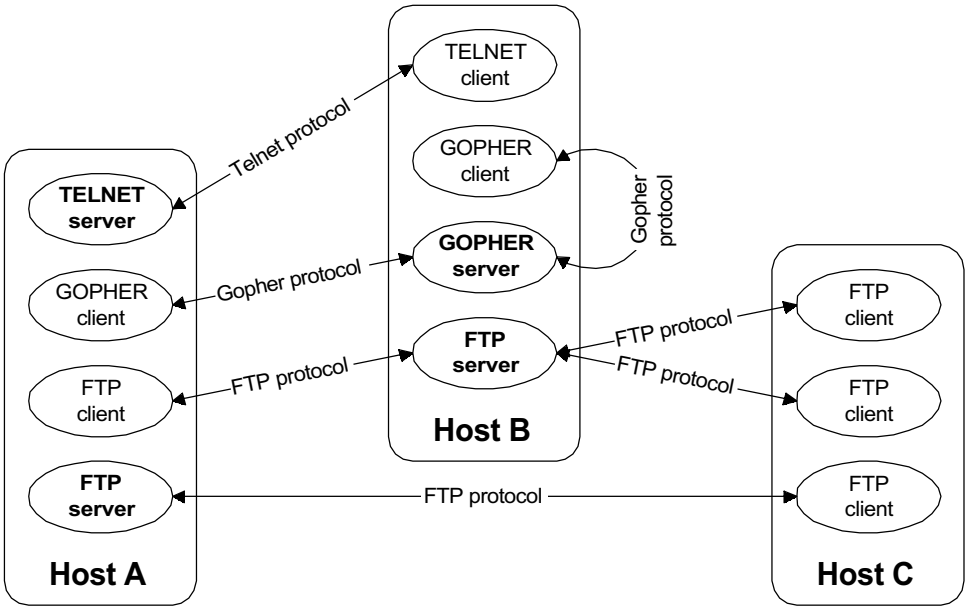
(a) proces klijent koji upotrebljava servis

(b) proces server, u većini slučajeva na drugoj mašini, koji pruža usluge servisa

(c) protokol, koji upotrebljavaju klijent i server da bi mogli komunicirati

[3] Izvor: ftp://nic.merit.edu/nsfnet/statistics/history.netcount

Primjer klijent/server modela dan je na slijedećoj slici:



Slika 6: Primjer Klijent/Server modela

Internet standard za TCP/IP servise obično specificira protokol i neke od svojstava klijenta i servera. Mnogi detalji klijenta kao što su to korisničko sučelje i detalji implementacije servera nisu navedeni. Protokol za servis i server mogu biti napravljeni tako da prihvaćaju više clienata istovremeno (za primjer pogledajte ftp i gopher server na hostu B) ili da serijaliziraju pristup serveru omogućavajući samo jednom klijentu pristup u jednom trenutku.

Klijent/Server model koji definira TCP/IP ne bazira se na pokretanju programa servera na računalima specijalno određenima samo za taj zadatak. Svako računalo može imati jedan ili više procesa koji su serveri za neki servis kao i različiti broj clienata za servise koji su na raspolaganju na tome ili bilo kojem drugom računalu u mreži.

3.3.1 Internet protokol i usmjeravanje (routing)

Usmjeravanje (*routing*) se na Internetu u osnovi obavlja korištenjem tablica koje opisuju koje spojište (*interface*) treba upotrijebiti da bi se došlo do odredišta. Postoje različite metode za

promjenu i nadopunjavanje tablica koje će biti spomenute. Međutim, da bi se omogućila promjena i nadopunjavanje tablica postoje posebni protokoli za podršku.

3.3.1.1 Protokoli za podršku

Protokoli za podršku su namijenjeni preusmjeravanju paketa, proslijeđivanju poruka o greškama i pretvaranju IP adresa u adrese pogodne za niže slojeve mreže. Oni ne upravljaju usmjeravanjem paketa, ali ih koriste drugi protokoli koji se bave usmjeravanjem paketa.

Internet Control Message Protocol (ICMP)

ICMP je protokol koji se mora implementirati zajedno sa IP-om. On obavlja osnovne mrežne usluge kao što su to slanje poruka o greškama pri slanju paketa na računala ili mreže računala koja su nedostupna kao i ostale neotklonjive greške kod usmjeravanja (*routing*-a).

Adress Resolution Protocol (ARP)

Ovaj protokol pretvara IP adresu u odgovarajuću adresu sloja veze (*datalink layer*) u sklopu lokalne mreže. Prvo se koristi emitiranje (*broadcasting*[4]) IP adresa nakon čega se čeka na odgovor od nekog računala. Nakon što se primi ARP odgovor par IP adrese i adresa sloja veze se privremeno sprema (na neki određeni vremenski interval). Na taj se način spriječava stalno emitiranje koje može dosta opteretiti mrežu. Da bi se ARP koristio sloj veze mora omogućavati emitiranje poruka kao što to omogućava npr. ethernet.

Reverse Adress Resolution Protocol (RARP)

RARP obavlja obrnuto pretvaranje: iz adrese sloja veze u IP adresu. Naročito je koristan kod stanica bez diska koje kod podizanja trebaju saznati svoju IP adresu.

3.3.1.2 Protokoli za usmjeravanje (routing)

Protokoli za usmjeravanje odlučuju o usmjeravanju paketa na razini mreže. Internet usmjerivači (*router*-i) mogu biti dio autonomnog sustava (*autonomus system*) ili AS-a. AS je skup

[4] Emitiranje ili broadcasting je slanje poruka na mrežu koje nisu namijenjene samo jednoj određenoj stanici nego svima. Na taj način svaka stanica dobiva upit "da li je tvoja IP adresa aaa.bbb.ccc.ddd", ali samo stanica čija je to stvarno adresa odgovara na upit.

usmjerivača (*router*-a) koji su pod istom administracijom. Takvi usmjerivači koriste isti protokol za usmjeravanje nazvan interior gateway protocol (IGP). Postoje različiti IGP-i, ali svi usmjerivači u AS-u koriste isti. Da bi mogli komunicirati međusobno, AS-ovi koriste *external gateway protocol* (EGP) ili *inter-AS routing protocol*. Taj protokol ne zna detalje o usmjeravanju unutar samoga AS-a. IGP bi mogao biti predstavljen u telefoniji lokalnom centralom, dok EGP odgovara međugradskoj telefonskoj centrali.

Uobičajeno usmjeravanje (default route)

Većina računala ima samo jedno mrežno spojište i većina lokalnih mreža posjeduje samo jedan usmjerivač koji ih povezuje sa vanjskim svijetom. Prema tome, sva računala na lokalnoj mreži trebaju samo jednu uobičajenu routu (*default route*) koja usmjerava pakete preko usmjerivača u vanjski svijet.[5] Ovo je standardni način usmjeravanja na većini UNIX sustava.

Routing Information Protocol (RIP)

RIP je najstariji protokol za usmjeravanje koji se uobičajeno primjenjuje na Internetu. Razvijen je za lokalne mreže i zasniva se na emitiranju (*broadcasting*-u). Demon routed[6] koji se može pronaći na većini UNIX-a zasniva se na RIP-u.

Open Shortest Path First (OSPF)

OSPF će vjerojatno uskoro postati standardni IGP protokol. Pri usmjeravanju prometa OSPF uzima u obzir i brzinu veze kao i stanje veze, a ima mogućnost dijeljenja prometa za isto odredište po različitim putevima radi boljeg iskorištenja veza.

Exterior Gateway Protocol (EGP)

EGP je prvi inter-AS protokol namijenjen povezivanju AS-a sa jednim središnjim AS-om. On pretpostavlja da središnji AS zna sam kako upućivati podatke drugim AS-ovima. EGP uzima u obzir samo dostupnost, a ne brzinu ili opterećenje veze.

---

[5] Nije potrebna routa za računala na lokalnoj mreži jer svaki paket na lokalnoj mreži i tako primaju sva računala koja su na nju povezana.

[6] demoni (ili daemoni) su procesi koji se stalno izvršavaju na Unix sustavima. Routed je daemon koji služi za usmjeravanje paketa. Daemoni za razliku od demona nemaju negativnu konotaciju.

BGP je noviji inter-AS protokol koji je nadogradnja EGP-a (trenutna verzija je BGP-3). On je također protokol dostupnosti, ali omogućava složenije topologije mreže od zvjezdaste sa centralnim AS-om koju podržava EGP.

3.3.2 Protokoli nivoa prijenosa

Postoje dva osnovna protokola koja su povezana sa IP-om: TCP i UDP.

3.3.2.1 Transmission Control Protocol (TCP)

TCP je najpopularniji dvosmjerni protokol za prijenos podataka. Upotrebljava se za poštu, prijenos datoteka, udaljeno prijavljivanje i mnoge druge mrežne servise. Omogućava sigurnu dvosmjernu (*duplex*-nu) vezu i prijenos tokova podataka, eliminiranjem duplih paketa, ponovnim prijenosom oštećenih paketa i osiguravanjem isporuke paketa u pravilnom redoslijedu.

3.3.2.2 User Datagram Protocol (UDP)

UDP je nepouzdan protokol baziran na datagramima koji je namjerno rudimentaran. Ne garantira isporuku paketa, pa samim time ni isporuku u pravilnom redoslijedu ili bez duplih paketa. On jedostavno šalje nepovezane pakete i prima pakete koji pristižu. Standardni servis za imenovanje na Internetu (DNS) upotrebljava UDP, kao i neki transparentni sustavi za pristup.

3.3.3 Upravljanje simboličkim imenima i osnovni servisi

Većina Internet servisa visoke razine zahtijeva pretvaranje tekstualnih (ili simboličkih) imena u IP adrese. Među osnovne servise spada i usklađivanje vremena.

3.3.3.1 Domain Name Service (DNS)

Internet standard DNS pretvara imena računala kao što je to barok.foi.hr u IP adrese kao što je to 161.53.120.3. To je najviše upotrebljavana usluga na internetu. DNS je hijerarhijski

organiziran kao stablo. Redundantni name serveri odgovaraju na zahtjeve korisnika za pretvaranje imena u adrese. Protokol određuje kako DNS klijenti (*DNS clients*) postavljaju upite DNS serverima kao i način na koji DNS serveri komuniciraju međusobno.

3.3.3.2 Network Information Services (NIS)

NIS firme Sun Microsystems također omogućava pretvaranje imena u adrese. Međutim, ima i mogućnosti pretvaranja korisničkih imena u oznake, kao i druge slične usluge. Iako ima prednosti, ako se koristi na lokalnoj mreži, u WAN-ovima se koristi DNS.

3.3.3.3 Network Time Protocol (NTP)

NTP omogućava održavanje istog vremena na svim računalima u nekom dijelu mreže.

3.3.4 Servisi za komuniciranje

Servisi za komuniciranje u osnovi služe za komuniciranje među ljudima koji koriste računala povezana u mrežu.

3.3.4.1 Elektronska pošta

Najčešće upotrebljavan servis za komuniciranje je upravo elektronska pošta (*e-mail*). Da bi se napisala poruka koriste se korisnički agent (*user agent*) ili UA. Postoje različiti korisnički agenti za UNIX kao što su to /bin/mail, Berkely Mail, MH, Mush, Elm i Pine. Da bi se poruka prenijela potreban je agent za prijenos poruke (*message transfer agent*) ili MTA. Postoji više različitin MTA-a za UNIX kao što su to upas, smail, zmailer i sendmail. Najpoznatiji je ipak sendmail.

Simple Mail Transfer Protocol (SMTP)

SMPT je standardni Internet protokol za poštu. MTA-ovi koji su povezani u Internet koriste SMTP bez obzira na kojem se operativnom sustavu izvršavaju.

Multipurpose Internet Mail Extensions (MIME)

MIME je proširenje standarda za elektroničku poštu koje omogućava prijenos i drugih podataka osim standardnih 7-bitnih znakova. To se postiže na taj način da se dio poruke označi kao posebno kodiran.

3.3.4.2 Mailing liste

Pošta se obično isporučuje u poštanski sandučić (*mailbox*). Poštanski sandučić je u većini slučajeva povezan sa jednom osobom koja čita njegov sadržaj. Međutim, postoji mogućnost stvaranja fiktivnih korisnika (*mail alias*-a) koji dostavljaju poštu upućenu njima u više različitih sandučića. Na taj način se omogućava skupno diskutiranje o nekoj temi.

3.3.4.3 USENET vijesti

USENET vijesti su skupne diskusije o određenim temama nazvanim grupama vijesti (*news groups*). Kada se postave vijesti na neko računalo korisnici mogu lakše raspravljati određene teme. USENET vijesti nisu ograničene samo na Internet, nego se mogu prenositi i na druge mreže kao što su to BITNET, UUCP i FidoNet. USENET nije mreža u pravom smislu te riječi, to su jednostavno računala koja izmjenjuju vijesti.

Network News Transfer Protocol (NNTP)

NNTP je uobičajen protokol za razmjenu USENET vijesti među računalima na Internetu. Klijenti čitaju i šalju vijesti korištenjem servera za vijesti (*news server*).

3.3.5 Servisi za dijeljenje resursa

Osnovna uloga prve mreže sa razmjenom paketa, ARPANET-a je bilo dijeljenje resursa i pristup udaljenim resursima kao što su to superračunala i poslužitelji datoteka preko mreže. Tri najupotrebljavanija protokola na Internetu su SMTP za komunikaciju elektronskom poštom i dva protokola za dijeljenje resursa FTP i TELNET. To su ujedno i protokoli koji se zahtijevaju od računala poslužitelja na Internetu.

Unix 4.2BSD je također dodao tim protokolima i neke specifične za UNIX kao što su: RCP (*remote copy* - udaljeno kopiranje), RLOGIN (*remote login* - udaljeno prijavljivanje) i RSH

(*remote shell* - udaljeno izvršavanje naredbi). Iako zbog svoje jednostavnosti primjenjivi samo na UNIX računalima, omogućavaju lakše dijeljenje resursa, prvenstveno zahvaljujući jednostavnoj autorizaciji koja se, sa gledišta korisnika, obavlja neprimjetno.

3.3.5.1 Prijenos datoteka

Najjednostavniji način za pristup do datoteke koja se nalazi na udaljenom poslužitelju je kopiranje njenog sadržaja na lokalni poslužitelj. U tu svrhi postoje dva protokola.

File Transfer Protocol (FTP)

FTP je standardni protokol na Internetu za prijenos datoteka. Postoji i posebna vrsta FTP-a nazvana anonimni ftp (*anonymous ftp*) koja omogućava korisnicima bez korisničkog računa pristup datotekama na računalu. Anonimni FTP ima velike mogućnosti, ali ukoliko nije pravilno konfiguriran može prouzročiti mnoge sigurnosne probleme, pa ću se njime još pozabaviti u nastavku.

Remote Copy Protocol (RCP)

RCP je napravljen kao proširenje standardne UNIX naredbe za kopiranje cp. Zamišljen je za korištenje unutar odjela ili organizacije gdje treba postojati mogućnost jednostavnog dijeljenja datoteka. Zbog toga se i ne koristi u drugim slučajevima.

3.3.5.2 Udaljeno prijavljivanje

Da bi se koristile mogućnosti udaljenog računala, potrebno se na njega prijaviti. Da bi ta prijava bila jednaka kao da je računalo lokalno, koriste se slijedeća dva protokola.

TELNET (virtual terminal protocol)

TELNET je kao i FTP standardni protokol za udaljeno prijavljivanje. On omogućava stvaranje virtualnog terminala na udaljenom računalu te omogućuje korištenje tog terminala kao da je lokalni.

Remote Login Protocol (RLOGIN)

RLOGIN koristi R* autorizaciju o kojoj će još biti riječi u nastavku i omogućava jednostavno korištenje računala unutar odjela ili organizacije.

3.3.5.3 Ostali mrežni servisi

inetd: server svih servera

Računalo koje podržava mnogo servisa trebalo bi zaseban proces za svaki od servera svakoga servisa. Čak i na UNIX sustavima, mnogi neaktivni procesi, a većina servera bi čekala na vezu, stvaraju nepotrebno opterećenje računala. Zbog toga je stvoren inetd server koji prihvaća veze i pokreće odgovarajući server za servis. To omogućava smanjenje broja procesa kao i jednostavnije kreiranje servisa jer se kompleksne operacije potrebne za uspostavljanje veze kod TCP-a ili UDP-a ostavlja inetd-u.

Remote Procedure Call (RPC)

RPC je protokol firme Sun koji upotrebljavaju NFS i NIS servisi. Također omogućava stvaranje i drugih servisa zasnovanih na pozivima funkcija.

Network File System (NFS)

NFS firme Sun omogućava neprimjetan pristup datotekama preko mreže (TFA - *transparent file access*), tako da udaljene datoteke izgledaju kao da su lokalne. Dizajniran je za upotrebu u lokalnim mrežama (LAN-ovima), a postoje klijenti i za druge operativne sustave osim UNIX-a.

Dijeljenje štampača

Mnogi korisnici PC računala i dalje misle da je dijeljenje štampača jedini način korištenja mreže. Najupotrebljavaniji protokol za dijeljenje štampača na UNIX-u je *Line Printing Protocol* (LPR) koji se pojavio sa 4.2BSD UNIX-om.

3.4 Literatura:

1. Smoot Carl-Mitchell, John S. Quarterman: Practical Internetworking with TCP/IP and UNIX, Addison-Wesley Publishing Company, 1994.

2. Kimberly C. Claffy, George C. Polyzos, Hans-Werner Braun: Traffic Characteristics of the T1 NSFNET Backbone, National Science Foundation, 1993.
3. John S. Quarterman, Smoot Carl-Mitchell: RFC 1935: What is the Internet, Anyway?, TIC, April 1996.

4. Upravljanje mrežom

Rješavanje bilo kakvih problema sa mrežom je uvijek težak i dug proces. Međutim, kod manjih mreža većina problema se može otkriti i otkloniti i korištenjem standardnih UNIX alata kao što su to ifconfig i netstat.

4.1 Alati za otkrivanje problema

4.1.1 ping

Ping je jednostavan, ali vrlo koristan program. Šalje *ICMP echo request* poruke drugom hostu koje onda vraća *ICMP echo reply*. Slanje poruka se naziva pinganje. Pinganje je dobar način za provjeru veze. Ako ping funkcionira onda je sigurno da dva hosta mogu razmjenjivati IP pakete. Ukoliko se dva hosta ne nalaze na istoj mreži, provjerava se i usmjeravanje.

Da bi se provjerilo da li je npr. host firewall dobro povezan može se sa baroka zadati slijedeća komanda:

```
$ ping firewall
PING firewall (161.53.120.20): 56 data bytes
64 bytes from 161.53.120.20: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 161.53.120.20: icmp_seq=1 ttl=255 time=1 ms
64 bytes from 161.53.120.20: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 161.53.120.20: icmp_seq=3 ttl=255 time=0 ms
64 bytes from 161.53.120.20: icmp_seq=4 ttl=255 time=0 ms
64 bytes from 161.53.120.20: icmp_seq=5 ttl=255 time=0 ms
64 bytes from 161.53.120.20: icmp_seq=6 ttl=255 time=0 ms
64 bytes from 161.53.120.20: icmp_seq=7 ttl=255 time=0 ms
64 bytes from 161.53.120.20: icmp_seq=8 ttl=255 time=1 ms
64 bytes from 161.53.120.20: icmp_seq=9 ttl=255 time=0 ms

----firewall PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/2 ms
```

Ukoliko se host odazove, kao što je to na primjeru, onda je sve u redu. Međutim, ako se host ne odazove postoje brojna objašnjenja. Možda je isključen ili možda pretvaranje simboličkog imena u numeričku adresu (DNS) ne funkcionira (može se pokušati pingati numerička adresa host-a).

Zbog toga što ping radi direktno sa IP paketima i ne oslanja se na TCP ili UDP testira se sama povezanost. Jedini uvjet koji mora biti ispunjen na lokalnoj mreži da bi ping radio je da radi ARP pretvaranje IP adresa u adrese sloja veze (tj. hardverske adrese mrežnih kartica).

Ping može korisno poslužiti i za određivanje vremena potrebnog da se poruka "probije" do nekog drugog hosta. Ukoliko je to vrijeme veće od dozvoljenog dolazi do time-out-a i veza se prekida.

4.1.2 netstat

Netstat je kompleksniji program koji omogućava prikazivanje informacije o mreži koje su u vezi sa hostom. Različite opcije omogućavaju prikazivanje podataka o operativnom sustavu koji su u vezi sa mrežom, stanje veza na mreži, stanja spojišta, ruting tabele i podatke o prijenosu mrežom.

```
$ netstat
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp        0      0  barok.1742            firewall.telnet        ESTABLISHED
tcp        0      0  barok.4491            as400.ftp              CLOSE_WAIT
udp        0      0  barok.1029            *.*
udp        0      0  localhost.domain      *.*
udp        0      0  barok.domain          *.*
```

Navođenjem -s opcije netstat-u može se dobiti pregledna statistika o mrežnim uslugama od zadnjeg podizanja računala.

```
$ netstat -s
ip:
        2431692 total packets received
        6 bad header checksums
        4 with size smaller than minimum
        1 with data size < data length
        3 with header length < data size
        0 with data length < header length
        0 fragments received
        0 fragments dropped (dup or out of space)
        0 fragments dropped after timeout
        23213 packets forwarded
        56 packets not forwardable
        443 redirects sent
icmp:
        668 calls to icmp_error
        0 errors not generated 'cuz old message was icmp
        Output histogram:
```

```
                echo reply: 2172
                destination unreachable: 668
                routing redirect: 443
        6 messages with bad code fields
        0 messages < minimum length
        0 bad checksums
        0 messages with bad length
        Input histogram:
                echo reply: 5496
                destination unreachable: 714
                source quench: 108
                routing redirect: 441
                echo: 2172
                time exceeded: 1
        2172 message responses generated
igmp:
        0 messages received
        0 messages received with too few bytes
        0 messages received with bad checksum
        0 membership queries received
        0 membership queries received with invalid field(s)
        0 membership reports received
        0 membership reports received with invalid field(s)
        0 membership reports received for groups to which we belong
        0 membership reports sent
tcp:
        1482288 packets sent
                988889 data packets (112083955 bytes)
                11380 data packets (7515666 bytes) retransmitted
                333994 ack-only packets (212073 delayed)
                0 URG only packets
                53 window probe packets
                75808 window update packets
                72165 control packets
        2341143 packets received
                926125 acks (for 112097892 bytes)
                151600 duplicate acks
                0 acks for unsent data
                930246 packets (93086717 bytes) received in-sequence
                32327 completely duplicate packets (5028182 bytes)
                309 packets with some dup. data (50822 bytes duped)
                27170 out-of-order packets (723163 bytes)
                7 packets (0 bytes) of data after window
                0 window probes
                336063 window update packets
                423 packets received after close
                0 discarded for bad checksums
                0 discarded for bad header offset fields
                0 discarded because packet too short
        24493 connection requests
        24128 connection accepts
        47955 connections established (including accepts)
        49277 connections closed (including 396 drops)
        722 embryonic connections dropped
        859945 segments updated rtt (of 867458 attempts)
        3182 retransmit timeouts
                36 connections dropped by rexmit timeout
        18 persist timeouts
        201 keepalive timeouts
                41 keepalive probes sent
                54 connections dropped by keepalive
udp:
        79586 packets sent
        58334 packets received
        0 incomplete headers
        0 bad data length fields
        0 bad checksums
```

```
0 full sockets
2137 for no port (1470 broadcasts, 0 multicasts)
```

4.1.3 ifconfig

Ifconfig se u osnovi upotrebljava za konfiguriranje mrežnog spojišta nakon podizanja mašine. Može se upotrijebiti i za provjeravanje konfiguracije spojišta za mrežu, broadcast adrese i mrežne maske (*netmask*). Krivi podaci za *broadcast* adresu ili mrežnu masku mogu dovesti do naizgled misterioznih problema.

Na slijedećem primjeru mogu se vidjeti dva različita ispisa programa ifconfig (sa OSF-a i Linux-a):

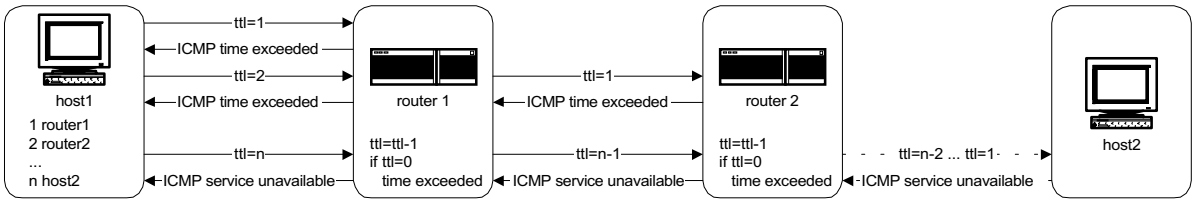
```
$ ifconfig ln0
ln0: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX>
     inet 161.53.120.3 netmask ffffff00 broadcast 161.53.120.255 ipmtu 1500

$ ifconfig eth0
eth0      Link encap:10Mbps Ethernet  HWaddr 00:80:C8:1D:66:EF
          inet addr:161.53.120.20  Bcast:161.53.120.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1879716 errors:0 dropped:0 overruns:0
          TX packets:2624095 errors:0 dropped:0 overruns:0
          Interrupt:5 Base address:0x320
```

4.1.4 traceroute

Traceroute prati kako IP paket putuje do svoga odredišta. Svaka linija označava jedan usjmerivač (*router*) preko kojega je IP paket prošao do svog odredišta. Koriste se tri UDP poruke koje se enkapsuliraju u IP paketu. Izgubljena poruka ili usmjerivač koji se ne odazove označava se sa "*".

Traceroute iskorištava "time-to-live" koji postoji u IP okviru. Ttl (*time-to-live*) služi da se paketi ne bi beskonačno zadržali u mreži i smanjuje se za jedan svaki puta kada paket prođe kroz usmjerivač. Traceroute šalje UDP paket sa ttl vrijednošću 1 za neiskorišteni port na odredišnom hostu. Međutim, kako je ttl vrijednost postavljena na 1 nakon što paket prođe prvi usmjerivač ttl vrijednost postaje 0 i usmjerivač šalje ICMP time exceeded poruku. Traceroute primi tu poruku, otkrije od kojeg je hosta došla koristeći IP adresu i ispiše rezultat. Nakon

toga se ttl vrijednost postavi na 2 i postupak se ponavlja. Takav paket generira *ICMP time exceeded* nakon što prođe kroz prva dva usmjerivača. Vrijednost ttl-a se povećava dok paket ne dođe do odredišnog hosta. Host koji je odredišni primi poruku, ali kako ne postoji port za koji je poslana poruka odredišni host vraća *ICMP service unavailable* i po tome traceroute zna da je paket došao do odredišta.



Slika 7: Način razmjene paketa kod traceroute-a

Primjer korištenja traceroute-a izgleda ovako:

```
$ traceroute pandora.dorm.umd.edu
traceroute to pandora.dorm.umd.edu (129.2.140.39), 30 hops max, 40 byte packets
 1  ltsa2 (161.53.3.19)  173 ms  187 ms  214 ms
 2  c7000.srce.hr (161.53.3.100)  191 ms  183 ms  183 ms
 3  Vienna-EBS1.Ebone.NET (192.121.159.25)  3857 ms 192.121.159.161 (192.121.159.161)  4774 ms
4548 ms
 4  STH-VIE-EBS.Ebone.NET (192.121.159.146)  4002 ms  4064 ms  4290 ms
 5  * 198.67.131.177 (198.67.131.177)  3893 ms  4104 ms
 6  icm-dc-1-F0/0.icp.net (198.67.131.36)  4245 ms  4155 ms  3938 ms
 7  icm-mae-e-H1/0-T3.icp.net (198.67.131.9)  4117 ms  4559 ms  4149 ms
 8  mae-east.digex.net (192.41.177.115)  4700 ms  3887 ms  4192 ms
 9  dca3-core1-h2-0.atlas.digex.net (206.205.246.1)  4456 ms  4857 ms  4185 ms
10  umd-gate.demarc.digex.net (206.205.243.2)  4007 ms  3657 ms  3643 ms
11  csc1gw-f0.umd.edu (128.8.1.224)  4642 ms  4318 ms  4168 ms
12  pandora.dorm.umd.edu (129.2.140.39)  4769 ms *  4801 ms
```

4.1.5 arp

Arp komanda se koristi da bi se pregledala privremena ARP-a (*ARP cache*). Može poslužiti i za ručno dodavanje podataka u privremenu memoriju ARP-a kao i njihovo brisanje.

Opcijom -a dobiva se spisak svih poznatih hardverskih adresa kartica.

```
$ arp -a
ps4 (161.53.120.14) at 00-00-c0-b3-5b-6d stale
ps2 (161.53.120.12) at (incomplete)
```

```
firewall (161.53.120.20) at 00-80-c8-1d-66-ef stale
as400 (161.53.120.10) at 42-00-00-3c-00-db
```

Kao što se na primjeru vidi, za računalo ps2 nije poznata njegova ethernet adresa. Natpis *incomplete* označava da je upravo u tijeku dodavanje njegove adrese u privremenu memoriju ARP-a (*ARP cache*). Ukoliko se natpis ne promijeni u adresu nakon nekog vremena postoji problem sa ARP upitom (*ARP request*-om) koji je poslan na mrežu.

5. Sustav elektroničke pošte

Elektronička pošta je na čvoru barok realizirana demonom sendmail koji u potpunosti ispunjava zahtjeve koji se postavljaju pred njega. On omogućava korištenje zamjenskih imena, distribuiranje pošte nekolicini korisnika istovremeno, i mnoge druge mogućnosti.

5.1 Zamjenska imena (alias-i)

Jedno od zanimljivih svojstava sendmail-a je mogućnost dodavanja zamjenskih imena (*alias*-a) za korisnike ili grupe korisnika radi lakšeg dostavljanja pošte. Zamjenska imena se nalaze u datoteci /var/adm/sendmail/aliases (također postoji link /etc/aliases koji pokazuje na istu datoteku, a služi za lakše pristupanje datoteci sa zamjenskim imenima). Na računalu barok.foi.hr se za svakog korisnika automatski, korištenjem admin korisnika i njegovih pripadajućih skripti (specijaliziranih programa pisanih u jeziku ljuske, *shell*-a), kreira zamjensko ime oblika ime.prezime@foi.hr. Međutim mogućnosti zamjenskih imena su mnogo veće. Zamjenska imena omogućavaju kreiranje *mailing list*-a. To su liste koje omogućavaju višestruko dostavljanje iste poruke.

Primjeri korištenja zamjenskih imena za sendmail su slijedeći:

```
1)  root: dpavlin, dnovak
2)  postmaster: root
3)  Dobrica.Pavlinusic: dpavlin
4)  therceg: therceg@public.srce.hr
5)  aiesec: gordana
6)  novo: dpavlin,dnovak,tvujec,poreski,dkermek,gordana
7)  archive: /var/archive_mail
8)  news: |/usr/local/uurec
```

Brojevi ispred redova su dodani samo radi lakšeg opisivanja.

Prvi red ilustrira mogućnost višestrukog slanja poruke za jednog korisnika (u ovom slučaju za korisnika root korisnicima dpavlin i dnovak).

Drugi red ilustrira rekurzivno slanje poruke. Naime, poruka poslana korisniku postmaster biti će poslana korisniku root koji će onda zahvaljujući prvome redu tu poruku dostaviti

korisnicima dpavlin i dnovak. Rekurzija je ograničena na razuman broj, u ovom slučaju 10. Ako je broj rekurzija veći od 10 javlja se poruka o grešci. Treba naglasiti da korisnik postmaster ne postoji.

Treći red je primjer punoga imena i prezimena koji se lakše pamti, a omogućava da se pošta za korisnika Dobrica.Pavlinusic@foi.hr dostavi korisniku koji ima login dpavlin. To su već prije spomenuta zamjenska imena (*alias*-i) za potpuna imena.

Četvrti red ilustrira mogućnost slanja poruke na drugi sustav koji je u drugoj domeni. Korisna primjena ovoga je mogućnost da se korisniku ostavi zamjensko ime (*mail alias*) nakon ukinuća njegovog korisničkog računa (*account*-a) da bi eventualna zalutala pošta mogla biti proslijeđena na novi korisnički račun.

Peti red je korisna primjena zamjenskih imena koja omogućavaju da jedna organizacija (u ovom slučaju aiesec) ima uvijek jednaku e-mail adresu bez obzira na to tko (i koliko korisnika) stvarno čita poštu.

Šesti red je jednostavan primjer *mailing list*-e kod koje se slanjem poruke na korisnika novo@foi.hr ona dostavlja na više odredišta.

Sedmi red predstavlja zamjensko ime koje poruku dodaje na kraj datoteke dok osmi red predstavlja preusmjeravanje u program uurec. Program uurec može nakon toga manipulirati sadržajem poruke po želji. Preusmjeravanja u programe (*pipe*-ovi) se najčešće koriste za realizaciju automatskih mail-ing listi.

5.2 Preusmjeravanje pošte

Preusmjeravanje pošte je podržano ~/.forward datotekom koja se nalazi u osnovnom korisničkom direktoriju (*home directory*) korisnika. Postoje jednake mogućnosti kao i kod zamjenskih imena. Neki od primjera su:

1. Preusmjeravanje pošte na drugi host:

```
dpavlin@jagor.srce.hr
```

2. Preusmjeravanje sa čuvanjem lokalne kopije:

```
dpavlin, dpavlin@jagor.srce.hr
```

3. Snimanje poruke u arhivu i dostavljanje poruke

```
dpavlin, /usr/users/dpavlin/mailarchive
```

4. Proslijeđivanje poruke programu

```
|"/usr/users/dpavlin/mailmanager"
```

Zbog sigurnosnih razloga ~/.forward datoteka bi trebala biti vlasništvo i čitljiva samo za korisnika. Kod primjera 3 i 4 treba napomenuti da sve putanje (*path*-ovi) moraju biti navedeni u potpunosti, jer je radni direktorij programa sendmail privremeni direktorij (*mail spool*) u kojem nije dopušteno čitanje i pisanje korisnicima.

5.3 Literatura:

1. Smoot Carl-Mitchell, John S. Quarterman: Practical Internetworking with TCP/IP and UNIX, Addison-Wesley Publishing Company, 1994.
2. Brad Knowles: comp.mail.sendmail FAQ, February 1996.

6. Uvođenje novih radnih stanica i sigurnosni problemi

Kod instaliranja novih stanica koje su povezane na Internet jedno od najvažnijih pitanja je pitanje sigurnosti. Poznati su mnogi problemi (u literaturi nazvani *security holes*) od kojih pate UNIX radne stanice. Zato je veoma važno da se sustavni administratori upoznaju sa potencijalnim problemima.

Informacije u ovom poglavlju odnose se općenito na UNIX radne stanice, a ne na Digital-ovu implementaciju OSF/1 koja se koristi na računalu barok. U tekstu će posebno biti naglašeno kako su spomenuti problemi riješeni u operacijskom sustavu OSF/1.

Sve adrese na mreži su dane u originalu, ali preporučam da se datoteke prvo potraže na jednom od servera u Hrvatskoj korištenjem Zarchie-a (hrvatske verzije archie servisa za pretraživanje anonimnih FTP servera). On se može koristiti komandom Zarchie -s ime_datoteke ili telnetom korištenjem telnet zarchie.srce.hr 5005.

6.1 Prevencija

Kao što se zna, prevencija je puno vrijednija od liječenja. Zbog toga je bitno upoznati se sa načinima prevencije neautoriziranog pristupa radnim stanicama.

6.1.1 Zaštita root korisnika

Zbog toga što root korisnik (poznat i kao superuser) ima posebna ovlaštenja njegova je zaštita vrlo bitna. Root korisnik je najčešća meta napada. U daljnjem tekstu biti će naglašeni načini zaštite root korisnika.

6.1.2 Sigurni terminali (secure terminals)

Novije verzije UNIX-a podržavaju tzv. sigurne terminale. Bez obira na njihov naziv, sigurni terminali ne moraju uopće stvarno biti sigurni. Sigurni terminali su jednostavno oni koji prihvaćaju prijavljivanje (*login*) root korisnika.

Sigurni terminali mogu biti fizički priključeni na mašinu preko konzole ili serijskog porta. Također mogu biti priključeni i logički preko telnet-a.

Definiranje jednog ili više mrežnih terminala kao sigurnih predstavlja veliki sigurnosni rizik. Bez dodatne kontrole pristupa (koju pružaju TCP wrapper ili xinetd koji su objašnjeni kasnije) napadač se može prijaviti sa bilo kojeg dijela Interneta jednako kao i sa lokalne mreže.

Podaci o sigurnim terminalima se uobičajeno nalaze u /etc/ttys ili /etc/ttytab ovisno o implementaciji UNIX-a.

Datoteka izgleda otprilike ovako:

```
# device  program                          type     status  flags
console   "/usr/etc/getty std.9600" vt100  on       local   secure
ttyp0     none                           network  off     secure
ttyp1     none                           network  off
```

Prvi stupac je ime uređaja, drugi sadrži program koji će se izvršiti pri prijavljivanju ako je četvrti stupac (status) postavljen na vrijednost on, treći označava tip terminala za taj uređaj, dok peti sadrži posebne argumente ako oni postoje. Treba primijetiti da su definirana dva sigurna terminala i to console i ttyp0.

Napadač može dobiti root privilegije i bez logiranja kao root tako da se logira kao normalan korisnik i iskoristi `su` komandu te napiše root zaporku (*password*).

OSF/1 koristi datoteku /etc/ttys za sigurne terminale.

6.1.3 Osiguravanje ostalih korisnika

Jedan od dobrih načina osiguravanja korisničkih računa (*account*-a) je ograničavanje duljine trajanja korisničke zaporke. Također, bitno je odabrati sigurnu zaporku. Najčešći uvjeti su:

- Dužina od najmanje šest slova. Preporučljivo osam ili više slova.
- Kombinacija malih i velikih slova
- Barem jedan specijalni znak ($, ~, % i slično)
- Barem jedan broj
- Zaporka koja je različita od imena računa

- Ne uzimati dijelove stvarnih imena korisnika
- Ne uzimati riječi iz rječnika ili jednostavne varijacije
- Ne uzimati riječi napisane naopako (drowssap)
- Ne uzimati strane riječi ili imena
- Ne uzimati sve brojeve

Jedan od načina primoravanja korisnika da koriste sigurnije zaporke je upotreba nekog od programa koji poštuje navedena pravila. Primjer je npasswd koji je napisao Clide Hoover sa University of Texas i može se naći na ftp://ftp.cc.utexas.edu/pub/npasswd/.

Također dobra je ideja provjeriti korisničke zaporke jednim od programa koji služe za pogađanje kao što je to crack Alleca Muffeta sa University College of Walles. Može se naći na ftp://info.cert.org/pub/tools/crack/.

Dobar način osiguravanja računa je korištenje *shadow* datoteka. U tom slučaju zaporke korisnika se ne nalaze u datoteci /etc/passwd koja je dostupna svim korisnicima i kao takva je vrlo ranjivo mjesto. OSF/1 koristi sustav *shadow* datoteka.

6.1.4 Kontrola pristupa

Kontrola pristupa se ostvaruje dodavanjam još jednog filtera ispred procesa inetd koji inače prima zahtjeve za vezu sa mreže. Jedan od takvih programa je TCP wrapper koji se može naći na ftp://ftp.win.tue.nl/pub/security/. Kontrola pristupa se ostvaruje datotekama /etc/hosts.allow i /etc/hosts.deny. Za svaku vezu prvo se ispituje datoteka /etc/hosts.allow da bi se utvrdilo da li je veza sa tim hostom dozvoljena.

Ako je datoteka prazna, dozvoljene su veze sa svim hostovima. Nakon toga se provjerava da li je veza sa tim hostom eksplicitno zabranjena u /etc/hosts.deny datoteci. Ako je veza prošla sve provjere i niti na jednoj nije onemogućen pristup, servis je na raspolaganju. Također postoji i mogućnost zabranjivanja ili dopuštanja pristupa samo određenim servisima.

Postoje i drugačije verzije inetd demona[7] koje već imaju implementiranu kontrolu pristupa. Jedan od takvih programa je i xinetd koji se može naći na ftp://ftp.irisa.fr/pub/mirrors/xinetd/. Budući da je konfiguriranje xinetd-a kompliciranije od inetd-a potrebno je proučiti dokumentaciju koja dolazi sa programom. Također ima i dodatnih mogućnosti kao što su kompletno bilježenje podataka o vezama (*logging*) i mogućnost ograničavanja servisa na određeno vrijeme dana kao i mogućnost ograničavanja broja korisnika istog servisa.

OSF/1 na računalu barok.foi.hr ima instaliran TCP wrapper.

6.1.5 Hostovi kojima se vjeruje (trusted hosts)

Da bi omogućio lakši pristup hostovima preko mreže Berkeley UNIX je uveo pojam hostova kojima se vjeruje (*trusted hosts*). Na taj način host A vjeruje hostu B i omogućava korisnicima hosta B da se prijavljuju na host A bez unošenja zaporke.

Naredbe koje koriste sustav vjerovanja su rlogin, rsh i rcp. Naredba rlogin uspostavlja terminalsku vezu, rsh pokreće bilo koju naredbu na udaljenom računalu, dok rcp omogućava kopiranje datoteka između udaljenih i lokalnih računala.

Princip povjerenja među hostovima je vrlo koristan za distribuiranje datoteka, ali ukoliko se napadač uspije probiti na jedan host ima automatski pristup i na sve ostale hostove koji vjeruju prvom hostu. Postoje dva načina provjeravanja povjerenja. Jedan se zasniva na povjerenju u host (tj. u sve korisnike nekog host-a) i koristi datoteku /etc/hosts.equiv dok se drugi zasniva na povjerenju u određene korisnike i koristi privatnu datoteku korisnika ~/.rhosts. Ako nema niti jedne od datoteka host ne vjeruje niti jednom drugom host-u, što je sigurno rješenje.

Datoteka /etc/hosts.equiv omogućava navođenje hostova u koje lokalni host ima povjerenje i omogućava svim korisnicima (osim root korisnika) korištenje host-a bez provjere zaporkom. Imena hostova se navode svako u svojem redu npr:

```
oliver.efzg.hr
jagor.srce.hr
```

[7] inetd demon - program koji radi u pozadini i posreduje u uspostavljanju većine veza između korisnika na mreži i resursa

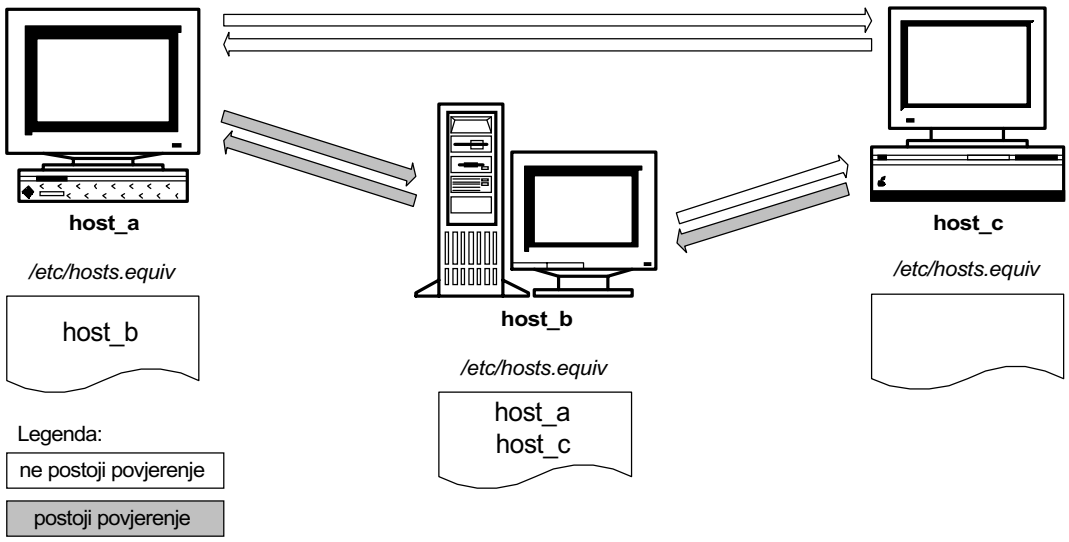
Kada korisnik želi omogućiti login sa neke druge mašine bez provjere zaporke on kreira ~/.rhosts datoteku sa imenima hostova kojima vjeruje. Također može navesti imena korisničkih računa ako su ona različita od lokalnih. Ukoliko korisnik dpavlin na host-u jagor.srce.hr želi omogućiti pristup korisnicima dpavlin i dino sa host-a barok.foi.hr napisati će slijedeću ~/.rhosts datoteku:

```
barok.foi.hr
barok.foi.hr dino
```

U prvom redu primjera se podrazumijeva ime računa dpavlin.

Treba naglasiti da se u datotekama /etc/hosts.equiv i ~/.rhosts može navesti i znak "+" što je **izrazito opasno**. Naime, znak "+" je zamjenski znak za sve mogućnosti (*wild card*) koji omogućava svim hostovima login bez autorizacije.

Princip povjerenja prikazan na slici donosi slučaj u kojem host_b vjeruje hostovima a i c i omogućava njihovim korisnicima pristup bez upisivanja zaporke. Nasuprot tome, host c ne vjeruje niti jednom hostu, dok host a vjeruje samo korisnicima sa hosta b. U primjeru je korišteno povjerenje za sve korisnike hosta (korištenjem datoteka /etc/hosts.equiv).



Slika 8: Princip razmjene povjerenja između host-ova

6.1.6 Poznati problemi sa sendmail-om

Jedan od najpoznatijih sigurnosnih problema u vezi sa Internetom je sigurno i slučaj Crva (*worm*-a), virusa koji je zarazio više od milijun računala na Internetu. Jedan od razloga za to je i sigurnosni problem sa sendmail-ovom debug funkcijom. Nikada ne biste trebali pokretati sendmail sa debug opcijom. Provjera se svodi na slijedeće:

```
$ telnet barok 25
Trying 161.53.120.3...
Connected to barok.foi.hr.
Escape character is '^]'.
220 barok.foi.hr ESMTP Sendmail 8.7.1/CI-8.7.1 ready at Tue, 30 Apr 1996 18:33:40 +0200 (MET
DST)
debug
500 Command unrecognized
```

Ukoliko se ne dobije odgovor sa kodom 500 stanica je ranjiva. Treba odmah poduzeti korake za isključivanje debug opcije, ili ako je to potrebno, za nabavljanje nove verzije izvornog koda sendmail-a i njeno prevođenje.

6.1.7 Finger

Finger servis omogućava korisnu uslugu pregledavanja podataka o udaljenom sustavu, kao i o korisnicima tog sustava bez obzira da li su oni trenutno aktivni ili nisu.

```
$ finger @barok.foi.hr
[barok.foi.hr]
Login       Name              TTY Idle    When            Office
dpavlin  Dobrica Pavlinusic    p1      Tue 17:16
info     Gopher user           p6      Tue 18:29
dpavlin  Dobrica Pavlinusic   *p5      Tue 13:59
dino     Dino Novak            pa      Tue 18:25  042 213-777 ext 116

$ finger gordana@barok.foi.hr
[barok.foi.hr]
Login name: gordana                     In real life: Gordana Cmigovic
Directory: /usr/users/stud/g/gordana    Shell: /usr/local/bin/bash
Last login Tue Apr 30 15:07 on ttyt1 from barok
No Plan.
```

Jedan od najvećih nedostataka finger-a sa stanovišta sigurnosti je to što prikazuje mnoge podatke o korisniku koji mogu biti iskorišteni pri neautoriziranom provaljivanju u sustav. Takvi podaci su datum zadnjeg korištenje korisničkog računa (obično napadač koristi korisničke račune koji se rijetko koriste) i osnovni korisnički direktorij. Ukoliko je sigurnost sustava veoma važna preporuča se instalacija verzije fingera koja odaje što manje podataka o korisniku ili potpuno uklanjanje fingerd demona iz datoteke /etc/inetd.conf.

Neke starije verzije VAX BSD 4.3 UNIX-a su također imale i bug u fingerd demonu koji je iskorištavao Internet Crv. Naime, funkcija C biblioteke gets() nije provjeravala ograničenja prilikom kopiranja stringa što je moglo prouzročiti prekoračenje stack-a (*stack overflow*) i omogućiti izvršavanje neautoriziranog programa.

Poznato je i da neke verzije fingerd demona omogućavaju korisnicima da pročitaju bilo koju datoteku u sustavu. Greška se očituje ako korisnik simbolički poveže svoju datoteku ~/.plan sa datotekom u sustavu koju želi pročitati. Nakon toga korištenjem finger-a na svoj korisnički račun (*account*) putem fingerd demona (koji se izvršava sa privilegijama root) dobiva podatke o sebi, ali i datoteku koja je simbolički povezana sa datotekom ~/.plan. U tom slučaju treba zamijeniti fingerd demon ili ga pokrenuti u /etc/inetd.conf kao neautoriziranog korisnika npr. nobody umjesto dosadašnjeg root.

6.2 Otkrivanje

Jedan od najboljih načina otkrivanja pokušaja proboja u sustav je automatizirani način, npr. korištenjem alata kao što su to COPS ili tripwire koji će biti opisani u nastavku. Također, ne treba zanemariti niti slučajno ili namjerno otkrivanje. Da bi se broj ručnih otkrivanja povećao potrebno je periodički provjeravati sustav.

6.2.1 Provjeravanje datoteka

Pri napadu na sustav mnoge se datoteke moraju promijeniti. Zbog toga je dobra praksa povremeno provjeriti slijedeće datoteke:

- /etc/passwd - potražiti nove korisnike, neobične korisničke ljuske (*shell*-ove) korisnika i korisničke račune sa korisničkim brojem (uid) 0 osim root-a
- /etc/services i /etc/inetd.conf (ili /etc/xinetd.conf) - provjeriti nove definicije servisa ili neobične putanje (*path*) do servera. Provjeriti da li je tcpd demon aktivan i da li provjerava veze i bilježi pristup (*loging*).
- skripte koje se pokreću pri podizanju računala (obično /etc/rc*) - provjeriti koje komande se pokreću i koji demoni se izvršavaju u pozadini
- crontab - provjeriti da su svi poslovi ispravni i potrebni, te provjeriti zaštitu *shell* skripti koje se pozivaju
- ~/.profile, ~/.cshrc, ~/.login - provjeriti nove ili neobične komande u skriptama za root *shell*. Provjeriti da PATH varijabla nije modificirana i da ne sadržava trenutni direktorij (označen sa ".").

6.2.2 COPS sustav za provjeravanje sigurnosti

COPS je skraćenica od Computer Oracle and Password System koji je napisao Dan Farmer, nekadašnji član CERT-a (*Computer Emergency Response Team*). Može se naći na ftp://info.cert.org/pub/tools/cops/.

COPS je modularni portabilni skup alata napisanih u awk-u, sed-u i C-u koji može pronaći mnoge sigurnosne probleme uključujući i:

- nesigurna korisnička imena

- sumnjive passwd i group redove
- programe koje pokreću skripte pri podizanju sustava ili cron, a za koje svi korisnici imaju dopuštenje za pisanje
- login i shell početne datoteke koje imaju nesigurna ovlaštenja
- opasne /etc/hosts.equiv redove
- modifikacije setuid programa

COPS ne pokušava ispraviti niti jedan od tih problema. On ih samo prijavljuje onome tko ga je pokrenuo. Treba također napomenuti da može pronaći mnoge probleme i kada je pokrenut sa bilo kojeg korisničkog računa čak i bez privilegija *superuser*-a.

6.2.3 tripwire

Alat tripwire služi za stvaranje jedinstvenih potpisa koji se sastoje od imena, zaštitne sume (*checksum*) i kontrolnih podataka, za svaku datoteku. Potpisi se zatim spremaju na medij koji je zašićen od pisanja (npr. disketa sa zaštitom ili CD-ROM) i provjeravaju periodički. Bilo kakva promjena u datoteci rezultira promijenjenim potpisom, što se prijavljuje korisniku.

Ovaj koristan alat su napisali Gene Kim i Gene Spafford sa Purdue University i može se naći na ftp://coast.cs.purdue.edu/pub/COAST/Tripwire.

6.3 Liječenje

Kada se posumnja u djelatnost napadača na jednom ili više korisničkih računa važno je provesti akciju što brže. Iako može biti izazov promatrati napadača na dijelu, to je također **veoma** opasno. Ako je to ikako moguće treba napadaču onemogućiti pristup na taj način da dokazi koje je ostavio ostanu za daljnje proučavanje.

6.3.1 Mijenjanje korisničke ljuske računa (account shell)

Jedan od najbržih načina da se nekom korisničkom računu (*account-u*) onemogući pristup je mijenjanje *shell* polja u /etc/passwd datoteci. Važno je da *shell* koji se stavi **ne** postoji u datoteci /etc/shells. Naime, na većini sustava ftpd (demon za FTP) provjerava da li korisnik ima

shell koji se nalazi u datoteci /etc/shells i ako nema odbija vezu. Prema tome, to je također način da se onemogući i FTP istovremeno.

Dobar kandidat za takav program je /bin/false koji jednostavno vraća vrijednost 1. Na taj način korisnička ljuska (*shell*) ne radi ništa osim što odjavi korisnika odmah nakon prijavljivanja.

Nakon toga je potrebno sa `ps` provjeriti da li korisnik ima aktivnih procesa i ukloniti ih.

6.3.2 Onemogućavanje FTP-om

Mijenjanje shella napadača je obično dovoljno da bi se onemogućio i FTP na taj korisnički račun (*account*), ali ne treba riskirati. Uvijek je dobro dodati korisnički račun i u /etc/ftpusers datoteku koji većina FTP servera provjerava i koja sadržava imena računa kojima je FTP onemogućen.

6.3.3 Promjena zaporke napadnutog računa

Promjena zaporke je najlogičnija mjera koja se može poduzeti. Međutim, ako je napadač već otkrio staru zaporku nitko ne garantira da neće i novu.

Također, ako se mijenja zaporka zgodno je zabilježiti njenu vrijednost u kodiranom obliku koja se nalazi u drugom polju datoteke /etc/passwd. Usporedba te vrijednosti sa drugima u datoteci može dati naslutiti koji bi još *račun*-i mogli biti ugroženi.

6.3.4 Onemogućavanje i ograničavanje povjerenja u hostove

Možda je napadač pristupio sustavu bez zaporke, korištenjem sustava povjerenja u hostove (*trusted hosts*, također poznat kao r* BSD). Ukoliko postoji datoteka /etc/hosts.equiv trebalo bi je isprazniti i time onemogućiti daljnji pristup barem privremeno. Također treba provjeriti postoji li datoteka ~/.rhosts u osnovnom korisničkom direktoriju napadnutog *račun*-a i ako postoji zapisati datum njene zadnje promjene te je preimenovati ili izbrisati. Treba i upozoriti administatora sustava koji je naveden u ~/.rhosts datoteci najbolje telefonom.

6.3.5 Uklanjanje datoteka napadnutog računa

Nakon proučavanja datoteka koje posjeduje korisnički račun (*account*) koji je napadnut uvijek je dobro provjeriti da li možda postoji još negdje datoteka kojoj je vlasnik napadnut korisnički račun naredbom:

```
$ find / -user napadnuti -exec ls -glad {} \;
```

Sve takve datoteke treba ukloniti. Datoteke napadnutog računa mogu se i arhivirati prije brisanja da bi se omogućila naknadna analiza.

6.4 Literatura:

1. Frederic J. Cooper, Chris Goggans, John K. Halvey, Larry Hughes, Lisa Morgan, Karanjit Siyan, William Stallings, Peter Stephenson: Implementing Internet Security, New Riders Publishing, Indianapolis, Indiana. USA, 1995.
2. David A. Curry: Improving the Security of Your Unix System, Final Report, April 1990.
3. Eugene H. Spafford: The Internet Worm Program: An Analysis, Purdue University, November 1988.
4. Mark W. Eichin, Jon A. Rochlis: With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988, Massachusetts Institute of Technology, February 1989.
5. Joyce K. Reynolds: RFC 1135: The Helminthiasis of the Internet, December 1989.
6. Clifford P. Stoll: The Cuckoo's Egg: Tracing a Spy Through the Maze of Computer Espionage, Doubleday, 1989.

7. Vatrozidi

Računala povezana u mrežu obično zahtijevaju identifikaciju korisnika korisničkim imenom i zaporkom. Međutim, poznato je da to ni u kom slučaju nije dovoljno. Ponekad se ne želi dopustiti pristup nekome računalu sa dijela mreže koji ne možemo direktno kontrolirati u našu privatnu mrežu. Konkretno, to znači da npr. želimo dopustiti pristup korisnika lokalne mreže na Internet, ali ne i pristup sa Interneta na lokalnu mrežu. U takvim se slučajevima koriste razne hardversko-softverske metode kontrole pristupa koje se jednim imenom nazivaju vatrozid (*firewall*).

Nakon što je Fakultet Organizacije i Informatike u Varaždinu postao CarNet čvor pojavila se potreba za izgradnjom vatrozida (*firewall*-a) koji bi omogućio komunikaciju između računala u sklopu fakulteta i Internet čvora, ali istovremeno onemogućavao neautorizirani izlazak na neki drugi čvor CarNet-a bez prethodnog prijavljivanja na naš lokalni čvor. Takvo rješenje omogućava i odterećenje samog čvora barok prebacivanjem nekih od servisa (npr. anonimni ftp - *anonymous ftp*) na računalo vatrozid, kao i povećanu sigurnost računala u sklopu fakulteta, koja se nalaze iza vatrozida.

7.1 Zaštita dijela mreže vatrozidom

7.1.1 Osnovni pojmovi

Vatrozid (*firewall*) je izraz koji se upotrebljava u automobilskoj industriji. Kod automobila, vatrozid je dio koji odvaja motor od dijela za putnike i služi za zaštitu putnika u slučaju da motor ekspodira.

Vatrozid u računalstvu je uređaj koji zaštićuje privatni dio mreže od javnog dijela i obrnuto. Primarna mu je funkcija da omogući kontrolirani pristup iz privatnog dijela mreže na Internet i obrnuto. Na vatrozid se može gledati kao na skup veoma različitih mehanizama koji se mogu podijeliti na dva osnovna principa:

- jedan koji služi spriječavanju prometa paketa u mreži i
- drugi koji služi omogućavanju prometa u mreži

Stvarni sustavi se nalaze negdje između te dvije krajnosti polažući veći značaj jednoj ili drugoj.

7.1.2 Čemu služe vatrozidi ?

Internet kao i svako drugo društvo sastoji se i od ljudi kojima je užitak stvarati probleme i neprilike, te zbog toga upotrebljavaju Internet za modernu verziju "šaranja po zidu". Drugi pak korisnici Interneta pokušavaju nešto korisno učiniti na mreži, pa ih takvi neodgovorni pojedinci smetaju. Svrha vatrozida je u tome da onemoguće "šaljivdžije", a istovremeno omoguće korištenje Interneta u za to predviđene svrhe. Mnoge korporacije koriste vatrozide da bi omogućile povezivanje na Internet i istovremeno i dalje ostale vjerne svojim principima zaštite povjerljivih računarskih podataka.

U posljednje vrijeme, vatrozidi se koriste i kao ambasadori korporacija prema Internetu. One stavljaju razne podatke koji su dostupni javnosti na vatrozide i na taj način omogućuju korisnicima Interneta pristup do njih. Neki od primjera takvih vatrozida su: UUnet.uu.net i gatekeeper.dec.com koji su pozitivno pridonijeli slici firmi koje su ih postavile.

7.1.3 Od čega štite vatrozidi ?

Neki tipovi vatrozida omogućavaju samo prolazak elektroničke pošte, i na taj način zaštićuju mrežu od svih napada osim onih elektroničkom poštom. Drugi pak, pružaju manje restriktivnu zaštitu, i onemogućavaju samo servise za koje se zna da su problematični.

U većini slučajeva, vatrozidi su konfigurirani tako da štite od neautoriziranog prijavljivanja iz "vanjskog" svijeta. To spriječava nepozvane da se prijavljuju na računala na zaštićenoj mreži. Neki tipovi vatrozida onemogućavaju promet u zaštićenu mrežu, ali omogućavaju korisnicima zaštićene mreže komuniciranje sa ostatkom Interneta. Vatrozidi pokušavaju zadovoljiti odnos sigurnosti i mogućnosti pristupa, međutim mogu zaštiti mrežu od svih mogućih napada ako se isključe iz napajanja. Takvo rješenje, međutim, također onemogućava sav pristup. Optimalna rješenja su negdje u sredini.

7.1.4 Različiti tipovi vatrozida

Da bi korisnik mogao doći do Interneta iz zaštićenog dijela mreže potrebno je prvo da prođe vatrozid, i da sa njega dođe do Internet-a. Postoji nekoliko načina realizacije vatrozida koji svi ispunjavaju postavljenu zadaću, ali na različite načine.

7.1.4.1 Selektivni usmjerivači (filtering routers)

Najjedostavniji pristup realizaciji vatrozida uključuje upotrebu programibilnih usmjerivača. Usmjerivači rade na razini IP-a (Internet Protokol-a, protokol koji se brine za dostavljanje paketa) selektivno propuštajući ili zadržavajući pakete po odredišnoj ili predajnoj adresi ili po broju porta. Ti podaci su sadržani u zaglavlju paketa. U ISO/OSI modelu selektivni usmjerivači bi odgovarali nivou mreže (3. nivou).

Ovaj način zaštite se najčešće koristi danas. Međutim, programiranje usmjerivača nije jednostavno. Isključivanje svih paketa koji nisu poželjni u zaštićenom dijelu mreže bi moglo predstavljati problem jer je sigurnost usmjerivača vrlo ovisna i o samom poretku pravila za dopuštanje ili zabranu prometa. U našem primjeru nije postojao usmjerivač predviđen za zaštitu privatnog dijela mreže, te se ovaj način nije mogao primijeniti.

7.1.4.2 Vatrozidi zasnovani na hostu

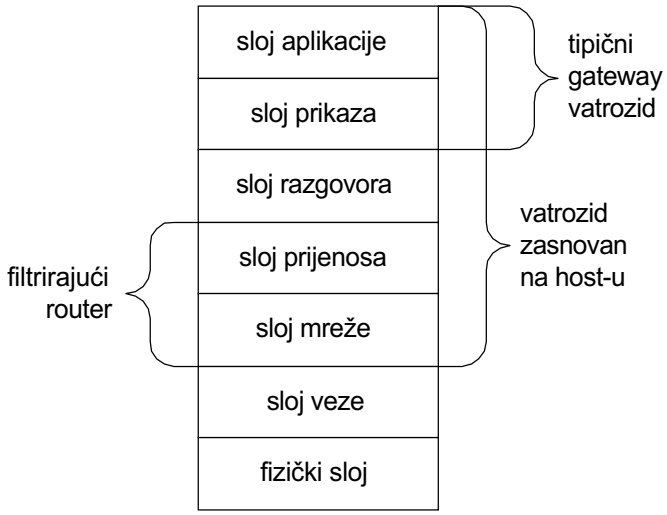
U ovom se slučaju koristi računalo umjesto usmjerivača. To nudi mnogo više mogućnosti praćenja aktivnosti koje se odvijaju preko vatrozida zasnovanog na hostu. Dok vatrozid zasnovan na usmjerivaču nadgleda pakete na IP razini, hostovi prenose kontrolu na razinu aplikacije. Da bi se osigurali od potencijalnih problema koji bi se mogli pojaviti zbog propusta u implementaciji sigurnosti u uobičajenoj programskoj podršci za mrežne usluge, vatrozidi zasnovani na hostovima obično koriste posebne verzije programa koji pružaju podršku potrebnim servisima koji su najčešće ogoljene verzije originalnih programa. Održavanje takvih kraćih programa je jednostavnije, a postoji i manja mogućnost za slučajne propuste koji narušavaju sigurnost. Osnovni nedostatak takvih vatrozida je potreba za posebnom programskom podrškom za svaki od servisa koji treba podržati za mrežu "iza" vatrozida. Zato se najčešće koristi kombinacija zaštite na razini aplikacije i selektivnog usmjeravanja (*routing filter*) kojega također obavlja sam host.

7.1.4.3 Izolacijske mreže

Izolacijske mreže su vrlo slične vatrozidima zasnovanim na hostu, osim što se između privatne mreže i Interneta ne postavlja host nego mreža. Međutim, ta se mreža može sastojati i od samo jednog čvora konfiguriranog tako da i jedna i druga mreža može pristupiti izolacijskoj mreži, ali istovremeno tako da izolacijska mreža ne propušta direktan promet između privatne mreže i Interneta. Glavna prednost izolacijske mreže je u tome što omogućava jednostavnije postavljanje i promicanje novih Internet adresa, naročito kod velikih privatnih mreža koje bi se inače morale znatno rekonfigurirati. To u osnovi znači da računala "iza" izolacijske mreže ne moraju imati adrese koje su poznate računalima na Internetu. Na taj način se može priključiti cijela mreža računala "iza" vatrozida na Internet korištenjem samo jedne Internet adrese. Ovaj je način popularan kod iznajmljivanja IP adresa od drugih davatelja Internet usluga, kao što je to u Hrvatskoj HPT.

7.1.5 Odnos selektivnih usmjerivača i vatrozida s obzirom na ISO/OSI model

Vatrozidi su skup različitih uređaja i programske podrške, pa zavisno od tipa vatrozida njegove funkcije su smještene na različitim nivoima ISO/OSI modela. Na slijedećoj slici vidi se kako bi se funkcije tri najčešća tipa vatrozida mogla smjestiti u ISO/OSI modelu:



Slika 9: Prikaz odnosa funkcija vatrozida i funkcija ISO/OSI modela

Kao što se vidi, filtriranje paketa je najjednostavnija operacija a pojavljuje se i kao dodatak vatrozidima koji kontroliraju promet na razini prikaza i aplikacije.

7.2 Primjer rješenja vatrozida

U našoj realizaciji vatrozida odabran je za vatrozid zasnovan na host-u, koji istovremeno vrši funkciju izolacijske mreže te proxy servera koji je zapravo skup zamjenskih aplikacija koje omogućavaju korištenje http (www), gopher i ftp servisa.

7.2.1 Dijelovi vatrozida

Naše rješenje vatrozida se sastoji od slijedećih dijelova:

- računala koji ima mogućnost usmjeravanja (*routing*-a). Ovdje će biti upotrijebljeno računalo sa operacijskim sustavom Linux (Unix kompatibilan) i ostala potrebna programska podrška (proxy server, anonimni ftp server i ostalo)
- dvije kartice za priključak na lokalnu mrežu (10 Mb/s ethernet sa suosnikom)
- veza sa Internetom za jednu mrežnu karticu
- veza sa privatnim dijelom mreže za drugu mrežnu karticu

Dakle, sada imate dvije odvojene mreže koji su povezane preko jednog računala. Vatrozidno računalo, koje ću nadalje zvati "firewall", može pristupiti do obje mreže, privatnog dijela i Interneta. Privatni ili zaštićeni dio mreže ne može direktno doći do Interneta kao što ni Internet ne može doći direktno do zaštićenog dijela mreže. To upravo i jest usluga koja se zahtijeva od vatrozida i zbog koje smo se i upustili u realizaciju vatrozida.

Dodatna prednost vatrozida je u tome što računala u privatnom dijelu mreže imaju vlastite adrese, neovisne o adresi na Internetu, tako da nismo ograničeni sa brojem adresa koje smo dobili (ili zakupili).

7.2.2 Realizacija vatrozida

Za realizaciju vatrozida odabrano je PC kompatibilno računalo sa Linux operativnim sustavom. Prvi korak je bio konfiguriranje mrežnih kartica. Da bi to bilo moguće bilo je potrebno izraditi specijalnu verziju *kernal*-a. Kernal je osnovni dio operacijskog sustava koji se učitava u trenutku uključenja stroja i ostaje u memoriji tijekom cijelog korištenja. Linux je u osnovi *makro-kernal* tip Unix-a jer sve upravljače vanjskih jedinica (*driver*-e) ima ugrađene u svoj *kernal*. Drugi pristup je *mikro-kernal* koji omogućava dinamičko učitavanje i uklanjanje dijelova kernal-a. U posljednjoj verziji i Linux je primijenio taj princip usvajajući module koji se po želji mogu učitati ili ukloniti, ali ipak ne za sve *driver*-e i posebne usluge kao što je to *firewalling*. Nakon što je kernal ispravno prepoznao *hardware* našeg računala i ponudio servise koji su nam bili potrebni a to je TCP/IP i IP firewalling, pristupilo se daljnjem konfiguriranju vatrozida.

Mrežne kartice su dobile logička imena /dev/eth0 i /dev/eth1. Nakon hardverskog konfiguriranja obiju mrežnih kartica trebalo je također dodati promjene u datoteku /etc/rc.d/rc.inet1. Ta datoteka je jedna od sistemskih datoteka koje se pokreću pri podizanju sustava i služi za ispravno konfiguriranje mrežnih kartica (dodjeljivanja IP adresa, mrežnih maski i routa da bi paketi znali na koju karticu moraju ići). Datoteka je nakon promjena izgledala ovako:

```
ifconfig eth0 161.53.120.20 broadcast 161.53.120.255 netmask 255.255.255.0
route add -net 161.53.120.0 netmask 255.255.255.0

ifconfig eth1 192.168.2.10
```

```
route add -net 192.168.2.0 dev eth1
route add -net 192.168.3.0 dev eth1
```

Korištenjem komande ifconfig mrežna kartica sa strane Interneta dobila adresu koja odgovara toj mreži, odnosno 161.53.120.20 i ime firewall što se vidi iz dijela /etc/hosts datoteke u kojoj su inače navedene adrese i imena računala:

```
161.53.120.20   firewall
```

Istoj mrežnoj kartici dodijeljena je i broadcast adresa[8] 161.53.120.255.

Broj iza riječi netmask (255.255.255.0) označava masku naše mreže, odnosno koliko je "velika" naša mreža. Budući da se ovaj broj spaja logičkom operacijom i (and) sa adresom u mreži jasno se vidi da naša mreža ima 254 moguće adrese (od mogućih 256 kombinacija adresa 0 je oznaka za mrežu, dok je adresa 255 rezervirana za *broadcast*).

Mrežnoj kartici sa strane mreže fakulteta dodijeljena je adresa u fiktivnoj mreži 192.168.2.0[9] i to adresa 192.168.2.10. Stanice u dvorani XII su dobile adrese 192.168.2.11 i veće.

Kako postoje dvije odvojene mreže u sklopu fakulteta, između vatrozida i Novella i iza Novella, stanice "iza" Novella su dobile adrese u mreži 192.168.3.0 i to adrese 192.168.3.51 na dalje za računala u dvorani XIII i adrese 192.168.3.100 i veće za računala koja se nalaze na ostatku fakulteta. Novell server je samo gateway (mrežni prolaz) između mreža 192.168.2.0 i 192.168.3.0. Zbog toga su njegove mrežne karte dobile adrese 192.168.2.2 sa strane vatrozida i 192.168.3.3 sa strane dvorane XIII i ostatka fakulteta.

Bilo je također potrebno dodati route do sve tri mreže koristeći naredbu route na firewall-u. Route služe da bi paketi usmjereni na adresu jedne od mreža mogli pronaći svoj put do te mreže ili odgovarajuće mrežne kartice.

Nakon toga route koje firewall podržava izgledaju ovako:

---

[8] broadcast adresa je adresa na koju se javljaju svi uređaju priključeni na taj dio mreže (mrežnom segmentu) i na taj način omogućava praćenje uređaja koji su priključeni (ili uključeni) a koji nisu

[9] mreža 192.168.0.0 je rezervirana za mreže zasnovane na IP protokolu unutar organizacija, koje se neće povezivati direktno na internet (RFC 1918)

```
firewall$ netstat -rn
Kernel routing table
Destination   Gateway        Genmask         Flags Metric Ref  Use   Iface
192.168.2.2   0.0.0.0        255.255.255.255 UH    0      0   206110 eth1
161.53.120.0  0.0.0.0        255.255.255.0   U     0      0  1748373 eth0
192.168.2.0   0.0.0.0        255.255.255.0   U     0      0  1005773 eth1
192.168.3.0   192.168.2.2    255.255.255.0   UG    0      0        0 eth1
127.0.0.0     0.0.0.0        255.0.0.0       U     0      0      846 lo
0.0.0.0       161.53.120.2   0.0.0.0         UG    0      0    98182 eth0
```

Iz ovoga se vidi da se paketi za mrežu 161.53.120.0 (Internet) usmjeravaju na prvu mrežnu karticu (eth0) i preko nje do čvora barok koji dalje takve pakete usmjerava na Internet.

Podaci za mrežu 192.168.2.0 (dvorana XII i XIV) se usmjeravaju direktno na tu mrežu preko kartice eth1 dok se podaci za mrežu 192.168.3.0 (dvorana XIII i kabineti profesora) usmjeravaju u mrežu 192.168.2.0 i to na mrežnu karticu Novella (ngw2.stud.foi.hr, 192.168.2.2) sa strane te mreže. Novell dalje takve pakete usmjerava na svoju drugu mrežnu karticu (ngw3.stud.foi.hr, 192.168.3.3) i dalje na tu mrežu (192.168.3.0).

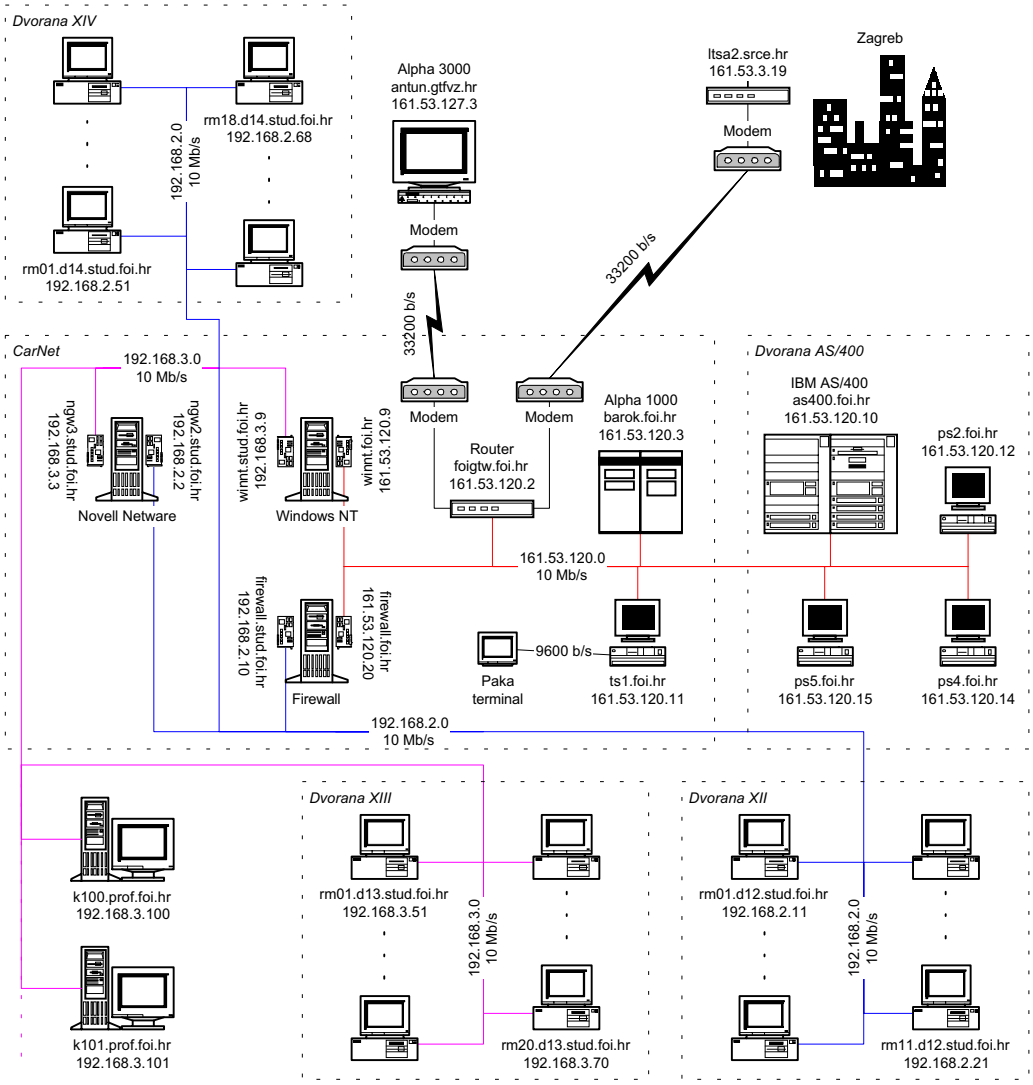
Na ispisu se vide i dvije pomoćne route, jedna do Novell gateway računala (ngw2.stud.foi.hr, 192.168.2.2) da bi routa do 192.168.3.0 znala "naći" gateway 192.168.2.2 i route 0.0.0.0 koja se naziva *default route* i služi da bi se mogao instalirati anonimni ftp server na firewall. Ona jednostavno upućuje sve pakete na foigtw.foi.hr (161.53.120.2) gateway na koji je povezan modem kojim se paketi upućuju na Internet.

Također, bilo je potrebno dodati i nove route na barok da bi se podržale nove mreže 192.168.2.0 i 192.168.3.0. Nakon toga tabele za usmjeravanje (*routing tables*) na računalu barok izgledaju ovako:

```
barok$ netstat -rn
Routing tables
Destination      Gateway            Flags     Refs     Use  Interface
Netmasks:
Inet             255.255.255.0

Route Tree for Protocol Family 2:
default          161.53.120.2       UG         38  2232426  sl0
127.0.0.1        127.0.0.1          UH         21   988452  lo0
161.53.120       161.53.120.3       U         103  8394898  ln0
192.168.2        161.53.120.20      UG         25  2489683  ln0
192.168.3        161.53.120.20      UG          2   586149  ln0
```

Time je postignuto da se sav promet za mreže 192.168.2.0 i 192.168.3.0 usmjerava preko gateway računala 161.53.120.20 (firewall.foi.hr) prema odgovarajućoj mreži.

Cijela konfiguracija Internet čvora u sklopu Fakulteta Organizacije i Informatike u Varaždinu prikazana je na slijedećoj slici.



Slika 10: Prikaz mreže Fakulteta Organizacije i Informatike

Nakon toga je konfiguriran NCSA Telnet koji se koristi za pristup Internetu iz dvorane XII. Poslije provjeravanja veze korištenjem ping-a sa strane firewall-a provjerili smo i mogućnost telnet-a direktno na firewall iz dvorane XII. To je bila osnovna postavka vatrozida.

Potom je slijedilo postavljanje proxy servera na vatrozid. Proxy server omogućava da računala "iza" vatrozida postavljaju upite drugim računalima na Internetu bez obzira na to što računala na Internetu zapravo "ne znaju" kako doći do adrese računala koje postavlja upit. Važno je naglasiti da do naših fiktivnih mreža 192.168.2.0 i 192.168.3.0 sada znaju doći samo računala firewall.foi.hr i barok.foi.hr, jer samo oni imaju route koje pokazuju na te mreže. Proxy dakle, vrši preadresiranje paketa, tako da se paketi za računala iza proxyja dostavljaju prvo proxyju (čiju adresu znaju računala na Internetu) koji ih onda prosljeđuje računalima "iza" njega. Paketi koji dolaze od računala iza vatrozida prije prosljeđivanja u Internet dobivaju adresu računala vatrozida (firewall.foi.hr) da bi se mogle vratiti potvrde ili novi podaci. Proxy se može konfigurirati da ostavlja sve prenesene podatke i na svome lokalnom disku, što omogućava ubrzavanje pristupa do često zahtijevanih odredišta. Proxy koji je instaliran na vatrozid računalu podržava http, ftp i gopher protokole.

7.2.3 Povećavanje sigurnosti vatrozida

Vatrozid ne pomaže pretjerano ako je konfiguriran kao otvoren sustav, neotporan na napade. Prvi posao kojim se trebalo pozabaviti je inetd, demon proces koji služi za pokretanje ostalih servisa o kojima ovisi Internet.

Neki od tih servisa su:

- Telnet
- Talk
- FTP
- Daytime

Da bi se ostavili samo najpotrebniji servisi bilo je potrebno editirati /etc/inetd.conf datoteku. Onemogućili smo sve servise osim telnet-a, rlogin-a i FTP-a. To omogućava mnogo veći stupanj sigurnosti.

Nakon toga datoteka /etc/inetd.conf je izgledala ovako:

```
#
# See "man 8 inetd" for more information.
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Echo, discard, daytime, and chargen are used primarily for testing.
echo    stream  tcp   nowait  root  internal
echo    dgram   udp   wait    root  internal
discard stream  tcp   nowait  root  internal
discard dgram   udp   wait    root  internal
daytime stream  tcp   nowait  root  internal
daytime dgram   udp   wait    root  internal
chargen stream  tcp   nowait  root  internal
chargen dgram   udp   wait    root  internal
time    stream  tcp   nowait  root  internal
time    dgram   udp   wait    root  internal
#
# These are standard services.
ftp     stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/wu.ftpd
telnet  stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/in.telnetd
#
# Shell, login, exec and talk are BSD protocols.
shell   stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/in.rshd
login   stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/in.rlogind
exec    stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/in.rexecd
#
# Ident service is used for net authentication
auth    stream  tcp   nowait  root  /usr/sbin/in.identd  in.identd
#
# End of inetd.conf.
```

Time je povećana sigurnost računala iza vatrozida. Isključivanjem servisa koji nisu neophodni (npr. finger), potencijalni napadači interne mreže fakulteta imaju na raspolaganju mnogo manji izbor servisa koje mogu koristiti u svojim napadima, te su na taj način i napadaji teži te manje uspješni. Vatrozidno računalo je najranjivije mjesto cijele koncepcije, naročito nakon instaliranja proxy servisa na razini aplikacije. Tada vatrozid ostaje jedino mjesto na koje potencijalni napadači mogu pokušati provaliti da bi došli do zaštićenog dijela privatne mreže. Zbog toga se preporuča što manji broj korisničnih računa na vatrozidu, po mogućnosti niti jedan. Potrebno je i redovno provjeravati sigurnost tog računala.

7.3 Zaključak

Ovako konfiguriran vatrozid u potpunosti ispunjava postavljene zahtjeve u pogledu autorizacije korisnika prije izlaska na CarNet odnosno Internet, kao i vođenja zapisa (logova) o njihovoj aktivnosti.

Povezane su sve dvorane sa računalima u sklopu fakulteta, na siguran način, a istovremeno bez žrtvovanja mogućnosti zahvaljujući proxy serveru. Računala u sklopu fakulteta ovim rješenjem ne zahtijevaju toliku pažnju pri administriranju, jer su potencijalni napadi sa Interneta onemogućeni vatrozidom.

7.4 Literatura

1. John Bryon: Izgradite firewall, Byte 30, travanj 1994, str. 50-58
2. William R. Cheswick, Steven M. Bellovin: Firewalls and Internet Security - Repelling the Wily Hacker, Addison-Wesley Publishing Company, Massachusetts, USA, 1995.
3. Frederic J. Cooper, Chris Goggans, John K. Halvey, Larry Hughes, Lisa Morgan, Karanjit Siyan, William Stallings, Peter Stephanson: Implementing Internet Security, New Raiders Publishing, Indianapolis, Indiana, USA
4. Alec Muffett: Allmost Everything You Ever Wanted to Know About Security, security FAQ, electronic document
5. Marcus Ranum, Allen Leibowitz, Brent Champman, Brian Boyle: Internet Firewalls Frequently Asked Questions, electronic document
6. David Rudder: Firewalling and Proxy HowTo, electronic document
7. Laurent Joncheray: Simple Active Attack Against TCP, Merit Network, Inc.
8. Yakov Rekhter, Robert G. Moskowitz, Daniel Karrenberg, Geert Jan de Groot, Eliot Lear: RFC 1918: Address Allocation for Private Internets, IETF Network Working Group, February 1996.

8. Imenovanje računala

Niži slojevi TCP/IP protokola uvijek koriste numeričke adrese. Međutim, sloj aplikacije omogućava korisnicima upotrebu imena umjesto brojeva. Imena su lakša za pamćenje, pa je jednostavnije napisati

```
telnet barok.foi.hr
```

nego napisati (i zapamtiti)

```
telnet 161.53.120.3
```

kada se želi korištenjem telnet-a pristupiti udaljenom računalu. Ukoliko telnet klijentu napišemo ime računala umjesto broja, on koristi servise za imenovanje da bi to ime pretvorio u adresu koju trebaju protokoli nižih razina.

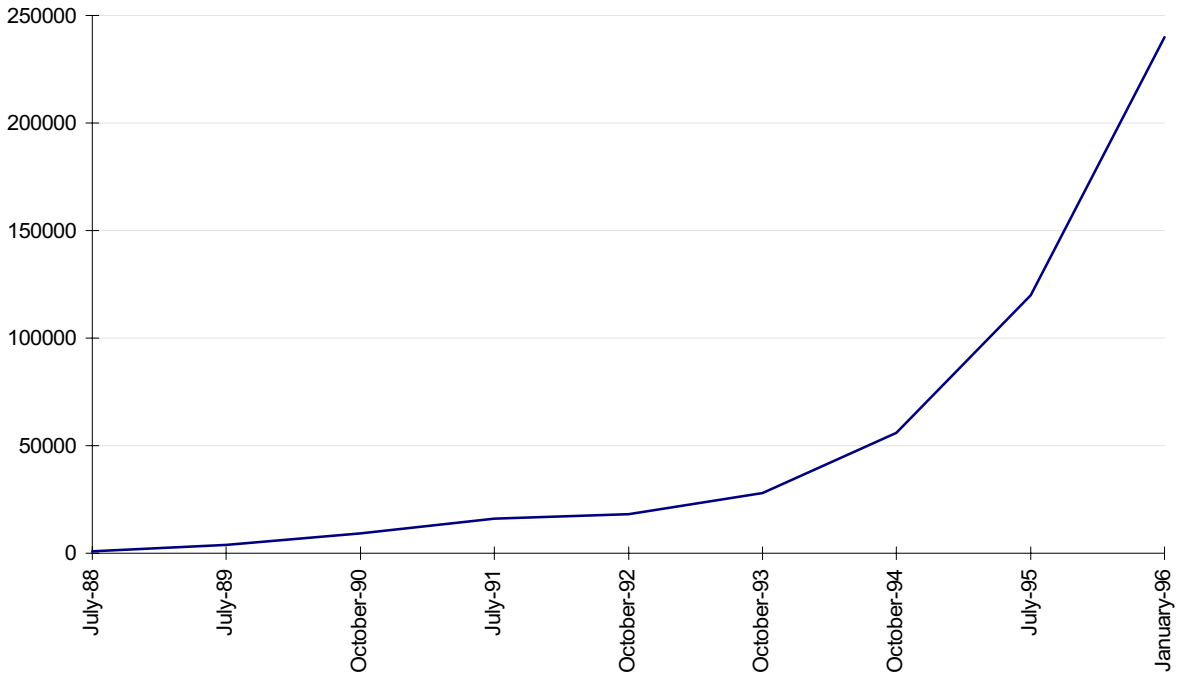
Imenovanje je evoluiralo kao i ostali protokoli, tako da mnogo naslijeđa često unosi pomutnju u imenovanje na Internetu i održavanje servera koji služe za podršku imenovanju.

8.1 Povijest

Prethodnik Interneta je bila ARPANET. Najvažniji podatak na toj mreži bila je IP adresa. ARPANET je upotrebljavala jednostavan način pretvaranja razumljivih imena u IP adrese. Dodjeljivanje imena je bilo centralizirano, a nova su se imena dodavala kada se za to ukazala potreba. Kako je većina računala na mreži u to doba imala samo jednu mrežnu karticu, ime dodijeljeno IP adresi je postalo sinonim za to računalo. Koristila su se imena kao *ut-sally* ili *sri-nic*. Uobičajena je bila konvencija da se koristi prefix koji označava ustanovu (u primjeru *ut* za University of Texas i *sri* za SRI International), te ime računala odvojeno crticom. U slučaju da je administrator zatražio ime koje se već koristi, njegov bi zahtjev bio jednostavno odbijen i morao je pronaći neko drugo ime.

Sva računala povezana na ARPANET su periodično kopirala tu centralnu bazu[10] imena i IP adresa da bi imala podatke o računalima. Rani Internet je također usvojio takav način imenovanja računala. Međutim, kada je broj računala prerastao stotine takav način je postao nepraktičan, kako zbog eksponencijalnog rasta broja računala, tako i zbog velikog zagušenja mreže kopiranjem, tada već prilično golemog, centralnog popisa.

Zbog svih tih problema, istraživači na Internetu su razvili novi sustav imenovanja koji je nazvan Domain Name System (DNS) koji će zamijeniti centralnu bazu i omogućiti automatsko distribuiranje odgovornosti o imenovanju računala. Od toga trenutka nadalje svjedoci smo eksplozivnog porasta broja domena kao što pokazuje slijedeći graf:



Slika 11: Prikaz porasta broja DNS domena[11]

Taj porast je u skladu sa porastom broja računala povezanih na Internet. DNS je službeni način imenovanja na Internetu koji primjenjuju sva računala povezana na mrežu i koji se relativno lako nosi sa tim porastom.

---

[10] Centralnu bazu imena održavao je Stanford Research Institute u Menlo Park-u, California i to njegov Network Information Center (SRI-NIC)

[11] Izvor: ftp://nic.merit.edu/nsfnet/statistics/history.hosts

8.2 Domain Name System (DNS)

Cilj uvođenja Domain Name System-a (DNS-a) može se pobrojati u nekoliko točaka:

- jedinstveno imenovanje resursa
- učinkovitost
- distribuiranost
- općenitost
- neovisnost

8.2.1 Jedinstveno imenovanje resursa

Glavni cilj je bio omogućiti jedinstvenost imenovanja, tako da bilo kojem resursu može biti dodijeljeno ime. To je značilo da prostor imena (*name space*) ne smije ovisiti o karakteristikama mreže i da mora biti neovisan o mrežnoj topologiji ili načinima usmjeravanja.

8.2.2 Učinkovitost

Rast broja računala i mreža povezanih na Internet je bio razlogom da se pri dizajniranju DNS-a posebna pažnja posveti učinkovitosti sustava. Da bi se to postiglo, sustav je građen hijararhijski i distribuiran sa upotrebom privremenog spremanja podataka (*cache-ing*).

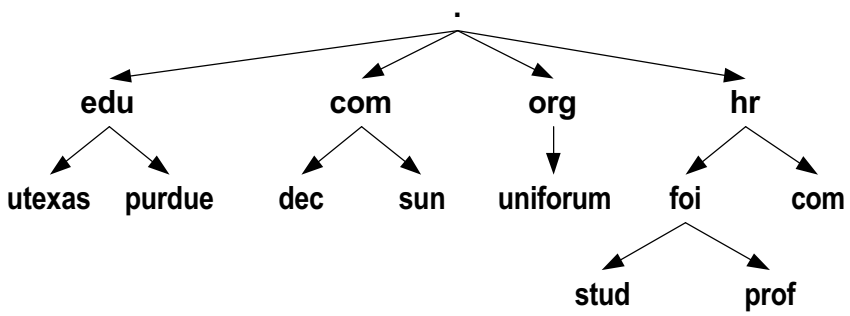
Na Internetu je vjerojatnije da računala u lokalnoj mreži komuniciraju sa drugim računalima u lokalnoj mreži, nego da komuniciraju sa računalima izvan te mreže. Prema tome, mnogo više upita je lokalne prirode. Zbog toga su podaci o lokalnim imenima računala dostupni preko lokalne baze podataka. Rjeđi upiti o udaljenim računalima rješavaju se konzultiranjem udaljene baze podataka.

Da se udaljena baza podataka ne bi svaki puta ponovo konzultirala bez potrebe, dobavljeni podaci se spremaju lokalno na neko kraće vrijeme određeno životnim vijekom podatka (*time to live - ttl*).

8.2.3 Distribuiranost

Implementacija tako velikog klijent-server modela koji je geografski raširen zahtijeva da se posebna pažnja posveti pouzdanosti. Zbog toga su uvedeni pomoćni poslužitelji (*secundary servers*) koji i u slučaju ispada glavnog računala zaduženog za neku domenu još uvijek mogu, korištenjem svoje kopije podataka, pružati usluge.

Distribuiranost se također provodi tako što su za lokalne domene zadužena lokalna računala, te na taj način organizacije, vlasnice domena, mogu same administrirati imena računala unutar sebe i dalje delegirati imena poddomena unutar svoje domene.



Slika 12: Primjer dijela stabla imenovanja

8.2.4 Općenitost

Praktični razlozi su zahtijevali da se provede općenitost, tako da bi sustav nakon uvođenja mogao poslužiti i u slučaju da se struktura podataka promijeni ili nadopuni. To se pokazalo kao veoma značajno, jer se od uvođena DNS-a 1983. godine pojavilo još mnoštvo ideja kako iskoristiti sustav, te su provedene i brojne nadopune osnovnog standarda.

8.2.5 Neovisnost

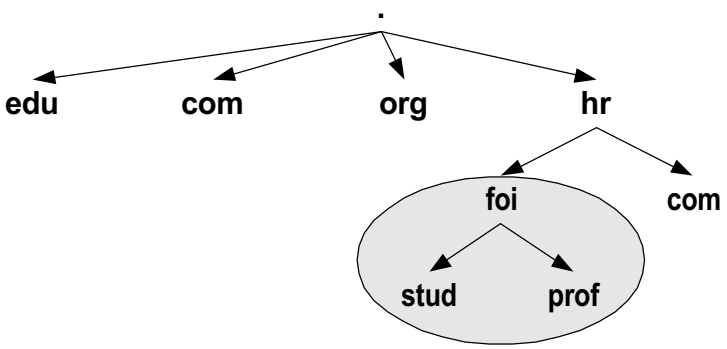
Sustav je dizajniran tako da njegove transakcije budu neovisne o tome koji ih komunikacijski sustav prenosi i na koji način.

8.3 Dijelovi DNS-a

Domain Name System se sastoji od nekoliko dijelova: servera (*name servers*), klijenata koji postavljaju upite (*resolvers*) i slogova o resursima (*resource records - RR*).

8.3.1 Prostor imena

Prostor imena (*Domain Name Space*) je skup imena u obliku stabla. Korijen toga stabla je *root* domena čija su djeca osnovne ili *top-level* domene. Na slici su prikazane edu, com, org i hr. Naravno, postoji ih mnogo više. Treba uočiti da sve zemlje imaju *top-level* domene kao dvoslovnu oznaku. Međutim, računala u Sjedinjenim Američkim Državama su dostupna preko domena com za komercijalnu, edu za akademsku, org za organizacije, mil za vojne, gov za vladine i net za službe koje pružaju podršku rada mreži, što potječe iz doba kada su *top-level* domene bile raspoređene po funkcijama i kada je većina računala na Internetu bila u Americi. Međutim, mnoge zemlje slijede takav primjer, pa tako u Hrvatskoj postoji domena com.hr za komercijalna poduzeća. Nazivi domena i nazivi računala se odvajaju točkama.



Slika 13: Domena foi.hr

Domena koja se proteže od imena računala do korijena stabla naziva se potpuno određena domena (*fully qualified domain name* ili *FQDN*). Ime koje se proteže samo dijelom stabla DNS-a naziva se relativna domena. Primjer FQDN je firewall.prof.foi.hr, dok je firewall.prof relativno ime. Potpuno određena imena (FQDN) trebala bi završavati sa točkom, da bi se lakše

razlikovala od relativnih imena. Međutim, mnoge aplikacije ne zahtijevaju, a neke čak i ne podržavaju - kao što je to elektronička pošta, završnu točku. Obično je iz konteksta jasno radi li se o relativnom ili apsolutnom imenu.

8.3.2 Imenovanje resursa

Dodjeljivanje imena resursu omogućava njegovo jednostavno referenciranje. Najvažnije je pretvaranje imena u IP adrese, i obrnuto, ali postoje i drugi tipovi resursa koji se mogu imenovati DNS-om, tako da se iskoristi slog resursa (*resource record*) koji ima slijedeći oblik:

<*domain_name ttl* IN *resource_type resource_value*>

Značenja pojedinih polja su slijedeća:

- *domain_name* - potpuno određen naziv domene (FQDN) ili relativan naziv domene koji je ključ za određivanje resursa
- *ttl* - vrijeme u sekundama koje serveri koji nisu autorizirani za neku domenu (tzv. sekundarni, redundantni serveri koji služe za povećavanje raspoloživosti) smiju čuvati informaciju u privremenoj memoriji (*cache-u*). Ova vrijednost je opcionalna.
- IN - označava da se resurs odnosi na TCP/IP odnosno Internet protokole. To je za sada jednini široko primjenjen protokol u DNS-u. Ukoliko se ne navede podrazumijeva se.
- *resource_type* - tip resursa koji omogućava da se jednom resursu dodijeli više svojstava
- *resource_value* - iznos resursa koji može biti jedna ili više vrijednosti

DNS definira slijedeće standardne tipove resursa:

- **A** - IP adrese su najčešće upotrebljavan tip resursa u DNS-u.

  domain_name A *ip_address*

  ```
  pr.    barok.foi.hr.       A      161.53.120.3
  ```

  Hostovi koji imaju više od jedne mrežne karte mogu imati više adresa dodijeljenih istom imenu.

- **HINFO** - podaci o hostu, definira hardware i operativni sustav određenog hosta.

  domain_name HINFO *hardware os*

  ```
  pr.    barok.foi.hr.       HINFO  Alpha OSF/1
  ```

- **CNAME** - zamjenska imena, povezuje ime resursa sa nekim drugim imenom.

  d*omain_name* CNAME *canonical_name*

  Najčešće se upotrebljava da bi se mogle ponuditi standardne usluge sa poznatim imenima. Tako je uobičajeno da domena koja nudi pristup preko Web-a ima host naziva www.ime.domene.

  ```
  pr.    www.foi.hr.  CNAME  barok.foi.hr.
  ```

- **MX** - *mail exchanger*, računalo zaduženo za poštu u određenoj domeni

  domain_name MX *priority server_name*

  Pošta poslana na *user_name@domain_name* biti će interpretirane kao *user_name@server_name*.

  ```
  pr.    foi.hr.             MX     barok.foi.hr.
  ```

- **WKS** - *well known services*, servisi koje nudi host

  domain_name WKS *protocol services*

  ```
  pr.    barok.foi.hr.       WKS    TCP  smtp telnet ftp
  ```

- **PTR** - pokazivači, omogućuju da jedna točka u domeni pokazuje na neku drugu domenu

  domain_name PTR *domain_name*

  Koristi se većinom da bi se omogućilo pretvaranje IP adresa u imena korištenjem specijalne *in-addr.arpa* domene o kojoj će više riječi biti kasnije.

- **NS** - *name server*, identificira računalo autorizirano za domenu.

  domain_name NS *server_name*

  Server je autoriziran za dio slogova o resursima (*resource records*) ili bazu podataka o domeni.

- **SOA -** *start of autority*, određuje početak podataka o autoriziranom serveru.

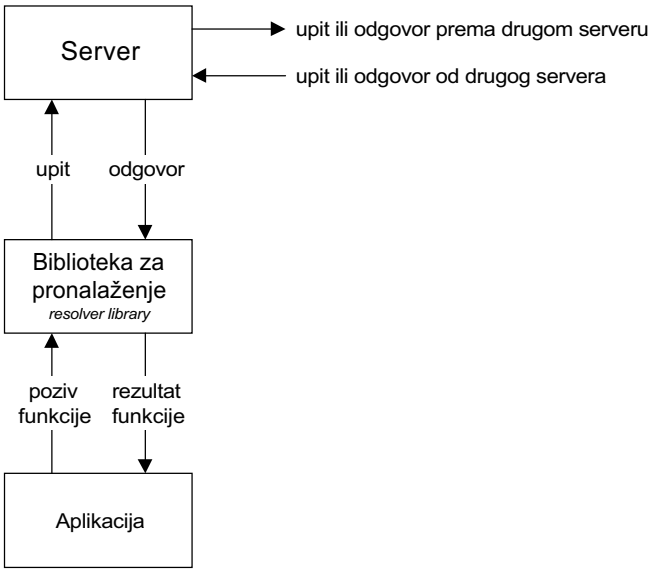
domain_name SOA *host_name mailbox serial refresh retry expire minimum*

Značenje polja je slijedeće:

- *host_name* - naziv autoriziranog servera
- *mailbox* - e-mail adresa administratora DNS-a
- *serial* - serijski broj koji označava bazu
- *refresh, retry, expire* - vrijeme u sekundama koje služi sekundarnim serverima da bi znali koliko često moraju obnavljati svoju bazu

8.4 Način funkcioniranja DNS-a

DNS se implementira upotrebom distribuiranih poslužitelja. Arhitektura DNS-a prikazana je na slijedećoj slici:

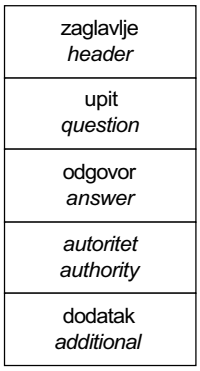


Slika 14: DNS arhitektura

Svaki server sadrži dio slogova o resursima. Serveri šalju ili primaju upite korištenjem TCP ili UDP protokola na za to rezerviranom portu 53. Upiti koriste UDP, dok se TCP koristi za prijenos cijelih zona (*zone transfer*) sekundarnim serverima. Ako server ne zna sam odgovoriti na upit, može upit proslijediti slijedećem serveru ili samo vratiti adresu servera koji bi mogao znati odgovor.

8.4.1 DNS poruke

DNS poruke se prenose između name servera i hostova koji postavljaju upite. DNS poruka se sastoji od pet dijelova kao što to pokazuje slika:



Slika 15: Format DNS poruke

Dijelovi poruke su slijedeći:

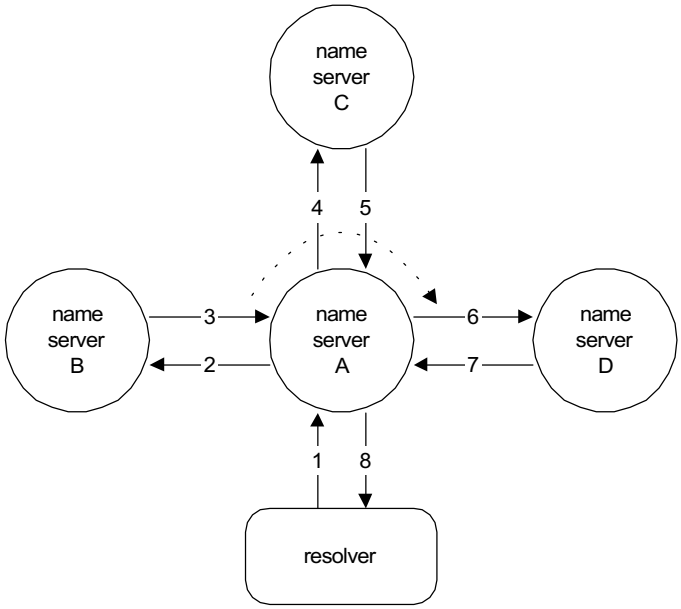
- zaglavlje - sadrži različite zastavice (*flags*) koje određuju na koji način će se vršiti obrada
- upit - ovaj dio poruke koristi klijent. Sastoji se od imena odredišne domene, tipa (*type*) i klase (*class*) upita. Tip upita može biti jedan od standardnih tipova resursa ili zamjenski znak (*wildcard*) za bilo koji podatak, dok je klasa uvijek IN za TCP/IP.
- odgovor - odgovor popunjava server koji odgovara na upit clijent-a.
- autoritet - autoritet je polje koje popunjava server, sa NS i SOA resursima za zahtijevanu domenu. Čak i ukoliko server ne zna odgovoriti na upit, može poslati NS i SOA resurse servera koji zna odgovoriti na postavljeni upit.

- dodatak - dodatni podaci za koje server smatra da bi mogli biti korisni klijentu. Npr. ako je klijent pitao za MX resurs, server mu može poslati i A resurs računala na kojega pokazuje MX resurs, jer je vjerojatno da će klijent trebati i adresu.

8.4.2 Način postavljanja upita

Aplikacije postavljaju upite DNS serverima i čekaju na odgovor. One za to upotrebljavaju standardne oblike upita koje su definirani u biblioteci za pronalaženje (*resolver library*-ju). *Resolver* se također brine o ponovnom slanju upita ukoliko odgovor nije primljen ili o slanju upita alternativnom serveru.

Unix aplikacije koje koriste DNS obično upotrebljavaju standardne funkcije *gethostbyname()* i *gethostbyaddr()* koje onda pozivaju DNS biblioteke.



Slika 16: Primjer resolving-a

Primjer resolving-a prikazan na slici ima slijedeći tok: resolver formira upit i želi dobiti odgovor od servera A. Server A može biti na istom računalu kao i resolver, na nekom računalu u lokalnoj mreži, negdje drugdje u mreži ili na jednom od korijenskih (*root*) servera.

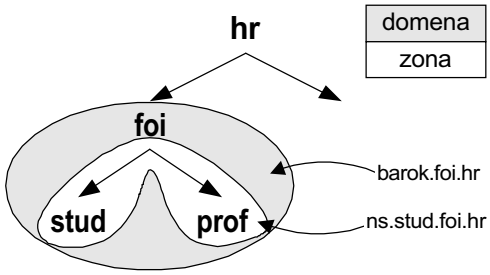
Pretpostavimo da server A ne zna odgovoriti na upit, ali smatra da bi server B mogao znati. Nakon što odgovor stigne do servera A on ga vraća resolveru koji je i postavio upit.

Svi koraci tog resolving-a prikazani su u tabeli:

Korak	Akcija
1	name server A prima upit od resolvera
2	A postavlja upit serveru B
3	B upućuje A na drugi name server, C
4	A postavlja upit serveru C
5	C upućuje A na drugi name server, D
6	A postavlja upit serveru D
7	D odgovara
8	A vraća odgovor servera D resolveru

8.4.3 Zone

Dio domene koji je u potpunosti pod kontrolom jednog *name server*-a naziva se zona. Razlika između domene i zone je u tome što zona kontrolira sva imena i podatke o domeni, osim onih koji su delegirani negdje drugdje.



Slika 17: Razlika između domene i zone

Ako pogledate sliku, vidjeti ćete da je za domenu foi.hr nadležan server barok.foi.hr. Sva računala koja se nalaze u toj domeni (npr. pandora.foi.hr, as400.foi.hr) a nisu u domenama

stud ili prof su registrirana u serveru barok.foi.hr (i odgovarajućoj zoni). Međutim, domene stud i prof su delegirani drugom serveru, ns.stud.foi.hr (i drugoj zoni).

8.5 Baza podataka DNS-a

Slogovi resursa za svaku zonu su zapisani u datotekama zone. Name server čita tu datoteku i sprema podatke u memoriju na taj način da postavljanje upita bude što lakše i brže. Format datoteke je određen i omogućava pisanje podataka o zoni bez obzira na samu implementaciju name servera. To omogućava promjenu programske podrške za name server ili premještanje baze podataka (datoteke) bez promjene samih podataka.

8.5.1 Primjer baze podataka za DNS

Slijedeći primjer je preuzet iz domene foi.hr. Linije koje počinju sa ";" su komentari, i oni se ignoriraju. Postoji nekoliko metakomandi koje počinju sa "$" u prvoj koloni. $ORIGIN je metakomanda koja definira podrazumijevanu domenu (*default domain*). Sva imena domena koja ne završavaju sa točkom uzimaju se kao relativna u odnosu na podrazumijevanu domenu. Primjer pokazuje dio zone foi.hr za pretvaranje imena u IP adrese.

```
; Data file of hostnames in this zone.
;
$ORIGIN foi.hr
;
@       IN      SOA     barok.foi.hr. postmaster.barok.foi.hr. (
                        199604301       ; Serial
                        28800           ; Refresh
                        7200            ; Retry
                        604800          ; Expire
                        86400 )         ; Minimum
        IN      NS      barok.foi.hr.
;
stud.foi.hr.    IN      NS      ns.stud.foi.hr.
prof.foi.hr.    IN      NS      ns.prof.foi.hr.
;
www             IN      CNAME           barok
ftp             IN      CNAME           pandora
gopher          IN      CNAME           barok
;
barok           IN      A               161.53.120.3
as400           IN      A               161.53.120.10
winnt           IN      A               161.53.120.9
pandora         IN      A               161.53.120.20
firewall        IN      CNAME           pandora
proxy           IN      CNAME           pandora
;
foi.hr.         IN      MX      5       barok.foi.hr.
;
```

U primjeru se vidi korištenje i relativnih i apsolutnih domena. Domena stud.foi.hr delegirana je na računalo ns.stud.foi.hr. Uvedeni su i poznata zamjenska imena (*alias*-i) tipa www.foi.hr, ftp.foi.hr i gopher.foi.hr za odgovarajuće servise. Uvođenje tih zamjenskih imena se pokazalo naročito praktično kada je anonimni ftp poslužitelj bio prebačen sa računala barok.foi.hr na računalo pandora.foi.hr. Definiran je i *mail exchanger* za domenu foi.hr, tako da se sva pošta može slati na ime.prezime@foi.hr.

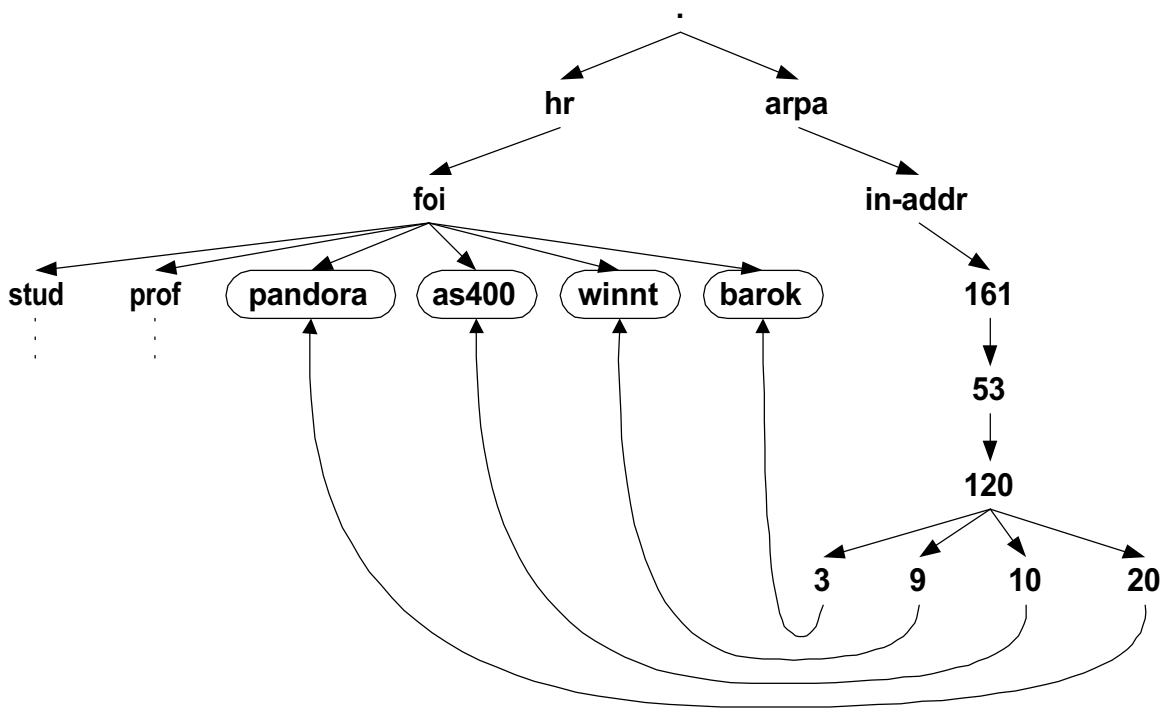
8.5.2 Pretvaranje IP adresa u simbolička imena

Iako je DNS optimiziran za pretvaranje imena u IP adrese (što se naziva *forward query*), pretvaranje IP adresa u imena je također često potrebno (to se naziva *reverse query* i po standardu nije obavezno, ali mnogi protokoli za autorizaciju, npr. BSD r* protokoli, ovise o ovim podacima). Upravo za tu svrhu postoji posebna domena *in-addr.arpa* koja upotrebljava PTR slogove da bi pretvorila IP adresa natrag u imena.

Primjer dijela zone foi.hr je slijedeći:

```
; Data file for reverse address to hostname.
;
@       IN      SOA     barok.foi.hr. postmaster.barok.foi.hr. (
                        199604301       ; Serial
                        28800   ; Refresh
                        7200    ; Retry
                        604800  ; Expire
                        86400 ) ; Minimum
        IN      NS      barok.foi.hr.
;
3       IN      PTR     barok.foi.hr.
9       IN      PTR     winnt.foi.hr.
10      IN      PTR     as400.foi.hr.
20      IN      PTR     pandora.foi.hr.
;
```

Grafički prikaz zone foi.hr možete pogledati na slijedećoj slici:

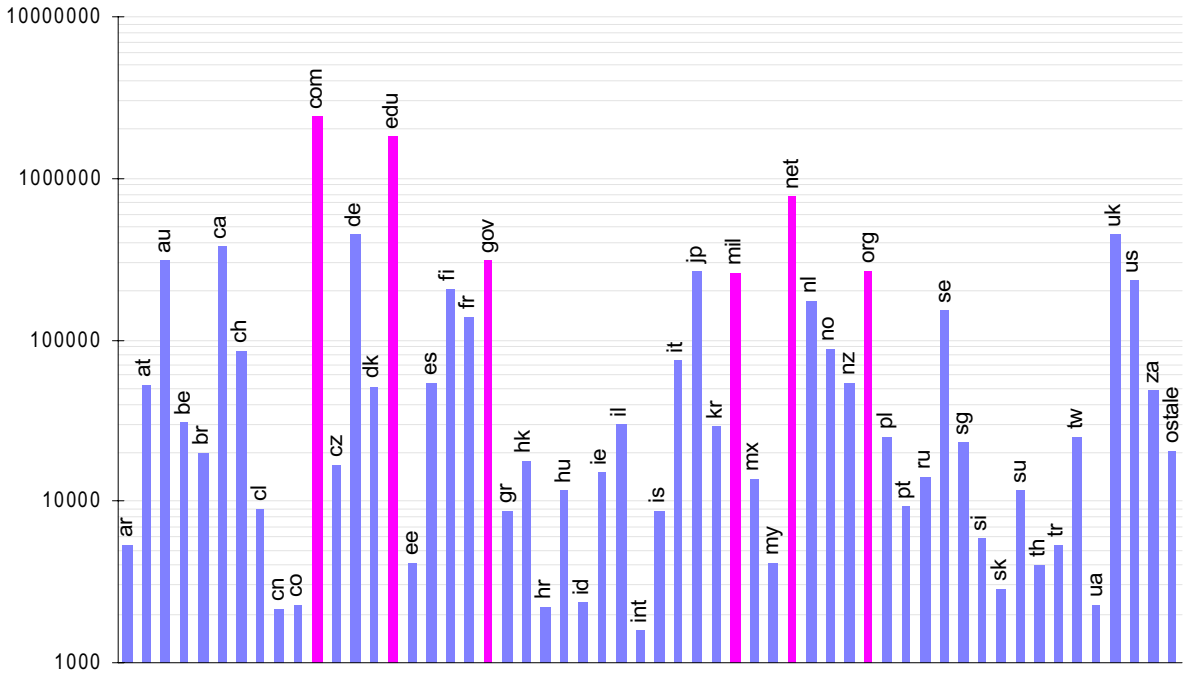


Slika 18: Prikaz zone foi.hr

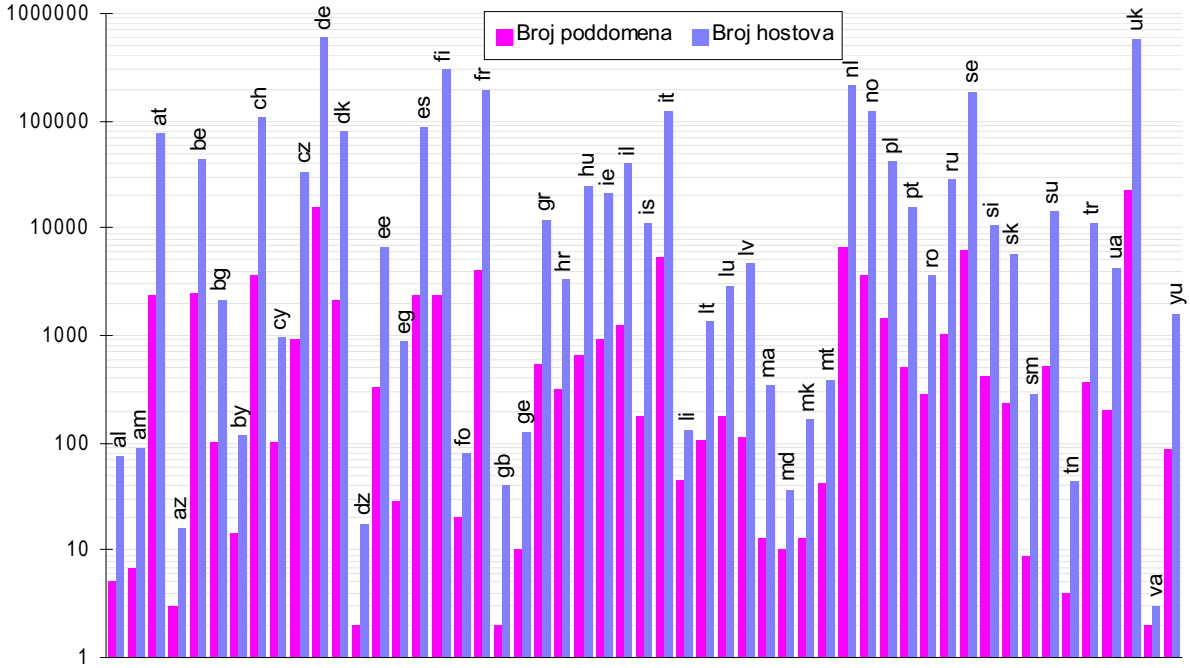
8.6 Pogled na trenutno stanje

Zanimljivo je promotriti koliko zapravo ima računala u osnovnim (*top-level*) domenama na Internetu. Prema podacima iz siječnja 1996, prikazanih na slici, ukupan broj registriranih hostova iznosi oko devet i pol milijuna razvrstanih u 136 osnovnih domena. Radi preglednije slike domene sa manje od 2000 hostova svrstane su u grupu ostale.

Domene u Americi (*com, edu, gov, mil, net i org*) posebno su označene na slici. Ova slika pokazuje da relativno mali broj domena, većinom u Americi, ima najveći dio računala na Internetu što se može objasniti povijesnim razlozima.



Slika 19: Broj hostova u domenama[12]



Slika 20: Broj poddomena i hostova po domenama u Europi[13]

[12] Izvor: ftp://nic.merit.edu/nsfnet/statistics/history.hosts

[13] Izvor: ftp://ftp.ripe.net/ripe/hostcount/RIPE-hostcount

Druga slika pokazuje osnovne (*top-level*) domene u Europi, kao i broj računala u svakoj od domena. Podaci su od Srpnja 1996. godine. Kao što je i za očekivati veći broj računala povlači i veći broj domena u kojima su ona smještena.

8.7 Literatura

1. Smoot Carl-Mitchell, John S. Quarterman: Practical Internetworking with TCP/IP and UNIX, Addison-Wesley Publishing Company, Massachusetts, USA, 1994.
2. Mark R. Horton: What is a Domain ?, Electronic document, December 1986.
3. Paul Vixie: DNS and BIND Security Issues, Internet Software Consortium, May 1995.
4. Christoph L. Schuba and Eugene H. Spafford: Countering Abuse on Name-Based Authentication, Purdue University, 1993.
5. Christoph L. Schuba: Addressing Weaknesses on the Domain Name System Protocol, Purdue University, August 1993.
6. D.W. Engebretson, Raymond Plzak: RFC 1956: Registration in the MIL Domain, DoD NIC, June 1996.
7. Paul Mockapetris: RFC 1035: Domain names: Implementation specification, November 1987.
8. Paul Mockapetris: RFC 1034: Domain names: Concepts and facilities, November 1987.
9. M. K. Lottor: RFC 1033: Domain Administrators Operations Guide, November 1987.

9. Zaključak

Kao što se iz prikazanog može zaključiti, održavanje Unix Internet poslužitelja nije jednostavano. Potrebno je solidno poznavanje mrežnih protokola i servisa, kao i mnogih prednosti i mana samog Unix operativnog sustava. Prva i osnovna prednost Unix-a je njegova otvorenost, dostupnost dokumentacije, standarda i programa (većinom u izvornom kodu).

Kompleksnost održavanja može se, međutim, mjeriti samo zadovoljstvom korisnika koje je, u posljednih 20 godina povijesti Unix-a, ostalo nepomućeno.

10. Prilog: povijest UNIX operativnog sustava

Povijest UNIX-a je povijest suradnje među korisnicima, jednako koliko i povijest tehnologije.

10.1 Uvod

Unix je rođen jednog sparnog ljeta, 1969 u New Jersey-ju kao rezultat nezadovoljstva zbog povlačenja AT&T-jevog BTL-a (*Bell Telephone Labs*) iz projekta Multics (*Multiplexed Information and Computing Service*) koji je bio združeni pothvat BTL-a, General Electric-sa i MIT-ja u kreiranju operativnog sustava za velika računala sa mogućnošću rada više tisuća korisnika istovremeno.

Priča o rastu i razvoju Unix-a je priča o jednom od najvećih napredaka u računalskoj povijesti. Kao što je to rekao David Tolbrook[14]:" Unix je omogućio korisnicima stvari koje su prije bile jednostavno nezamislive". On također naglašava da Unix nije toliko ogroman napredak u računarstvu koliko je veliko pojednostavljenje, koje je pokazalo da se relativno mali operativni sustav može izvršavati na različitim hardverskim paltformama, te da može biti portabilan, neovisan o računalu i pristupačan.

Neke od najvećih prednosti Unix-a ne potiču od njegove jednostavnosti, nego od stvarno zajedničkog načina razvoja i evolucije. Umjesto da bude proizvod jednog proizvođača sa nekim određenim hardverom, Unix je nastao kao rezultat želje pojedinaca da naprave sustav koji bi bio jednostavan, kojeg bi moglo koristiti više korisnika i koji bi mogao služiti kao ugodno okružje za programiranje.

Očuh Unix-a, AT&T, je imao vrlo sroga "kućna pravila" za Unix koja su naglašavala da nema podrške, nema popravaka pogrešaka i da nema autorstva. Stroga pravila AT&T-ja su se suprotstavljala osnovnoj ideji Unix-a, ali su također prouzročila jaku suradnju među korisnicima Unix-a. Kao rezultat toga, Unix je ne samo prvi portabilni operativni sustav, nego i prvi, ako ne i jedini, operativni sustav razvijen u međusobnoj suradnji - pravi otvoreni sustav.

[14] David Talbrook je izumitelj prvog dinamičkog pokazivača (kurzora).

10.2 Porodično stablo

Korijeni Unix-a sežu do Multics-a, koji će kasnije postići ograničeni uspjeh, ali 1969. Multics je jedva mogao posluživati tri korisnika istovremeno. Ken Thompson, iz BTL-a, je počeo razvoj igre nazvane Space Travel na Multics-u. Međutim, kako je izvođenje Space Travel-a oduzimalo mnogo procesorske snage računala GE-645, morali su naći neko drugo rješenje.

Na sreću, Thompson i Dennis Ritchie su pronašli neiskorišten DEC PDP-7 sa 340 ekranom. Operativni sustav se sastojao samo od assemblera i loader-a, i samo je jedan korisnik mogao koristiti računalo istovremeno. U takvom ograničenom okružju, dijelovi jednokorisničkog Unix sustava su ubrzo razvijeni; Space Travel je napisan ponovo za PDP-7, a napisani su i asembler i rudimentarni kernel operativnog sustava.

Tijekom travnja, svibnja i lipnja 1969. Thompson je razmišljao o pisanju višekorisničkog sustava datoteka. Iz diskusije Dennisa Ritchie, Rudd Canadeya i Kena Thompsona nastao je koncept sustava datoteka koji je implementiran za dan-dva na računalu PDP-7.

Bilo je nekoliko neuspjelih pokušaja da se BTL uvjeri da nabavi novo računalo za svoju razvojnu grupu (*computer research group*) zbog toga što je 1969. svaka nabavka računala podrazumijevala trošak od $100,000 ili više. Niti jedan od tih pokušaja nije uspio sve dok se Joe Ossanna nije sjetio da predloži kupovinu računala PDP-11 za obradu teksta. Administracija BTL-a je to smatrala vrijednom investicijom.

10.3 Unix razvojni sustav

U ljeto 1970. računalo PDP-11/20 je napokon stiglo. Ubrzo je na njega prenešen Unix. "Znali smo da se radi o prijevari. Obećali smo razviti program za obradu teksta, a ne operativni sustav" ističe Ritchie. Međutim, program za obradu teksta je bio uspješan i patentni odjel BTL-a bio je prvi komercijalni korisnik Unix-a, dijeleći PDP-11/20 sa razvojnom grupom.

Daljnji razvoj Unix-a nastavio se oko nekoliko jednostavnih principa: pisati programe koji rade jednu stvar i to dobro, pisati programe koji mogu međusobno razmjenjivati podatke i pisati programe koji rade sa tokovima znakova (*text streams*). To su bile osnove univerzalnog sučelja svih programa.

10.4 Unix postaje široko dostupan

Kako je nekoliko korisnika izvan razvojne grupe koristilo PDP-11 za obradu teksta, rasla je potreba da se operativni sustav dokumentira. Rezultat je bio prvi *Unix Programmer's Manual* koji su napisali Thompson i Ritchie u studenom 1971. U drugom izdanju, od lipnja 1972. napisano je da je broj instalacija Unix-a narastao na 10. Od toga trenutka nadalje razne verzije koje potiču iz New Jersey-ja označene su izdanjem (dokumentacije) i verzijom (diskova ili traka).

U listopadu 1973, Thompson i Ritchie su održali predavanje na Simpoziju o principima operativnih sustava (*SOSP - Symposium on Operating System Principles*) i više ništa nije moglo zaustaviti Unix. Neposredno nakon SOSP-a mnogi su centri tražili kopiju operativnog sustava. Prvi korisnik koji je dobio sustav je bio Lou Katz sa *Columbia University in Manhattan*.

Objavljivanje SOSP referata u *Communications of the ACM* izazvalo je eksplozivan porast potražnje Unix-a. U tom trenutku Unix još nije imao ni pet godina.

10.5 Suradnja među ranim korisnicima

Odluka pravnika AT&T-ja da omoguće obrazovnim institucijama korištenje Unix-a, ali bez ikakve podrške ili ispravaka grešaka imala je trenutan efekat: natjerala je korisnike da dijele međusobno. Dijelili su ideje, informacije, programe, ispravke grešaka i hardverske trikove. Prvi sastanak *Unix User Group*-a, koja će kasnije postati *Usenix Association*, održan je 15. svibnja 1974. na *Columbia's College of Physicians and Surgeons*. Pojavilo se dvadesetak ljudi.

Pravnici AT&T-ja također su uskoro odlučili omogućiti još dvjema agencijama lincenciranje Unix-a: Američkoj vladi (*U.S. goverment*) i *The Rand Corporation*, istraživačkoj grupi koja se financirala iz državnog proračuna. To je, međutim, uzrokovalo porast broja korisnika sa 33 u 1975. godini na 138 u 1976. od kojih je 37 bilo izvan Amerike. 1977. *Interactive Systems* iz Santa Monice u Kaliforniji bila je prva kompanija koja je podržavala Unix komercijalno.

10.6 Berkeley Software Distribution

Jedan od 33 korisnika iz 1975. godine je bio i *University of California-Berkeley* gdje je Ken Thompson bio student. On se 1975. odazvao na poziv da održi predavanje i donio, naravno, najnoviju verziju Unix-a.

Nekako u isto vrijeme, dva studenta Chuck Haley i Bill Joy dolaze na fakultet. Fascinirani Unix-om, oni počinju unapređivati Pascal koji je Thompson napravio, tako da je uskoro prihvaćen kao nastavno pomagalo.

Kada su fakultetski Model 33 Teletype terminali zamijenjeni sa ADM-3 ekranskim terminalima, Joy je zaključio da mu treba nešto više od običnog ed-a (editora koji je dolazio standardno sa Unix-om). Uzeli su editor em (što zapravo znači "ed za smrtnike", u originalu "*ed for mortals*") koji je napisao George Coulouris sa *Queen Mary College* u Londonu i od njega napravili linijski editor ex. Iz ex-a je Joy kasnije razvio vi ekranski editor.

Kada su se vijesti o Haleyovom i Joyevom Pascalu proširile, Joy je kreirao Berkely Software Distribution. Prva je distribucija ponuđena u ožujku 1978. Sadržavala je Pascal i editor ex, naplaćivala se 50 dolara, koliko je stajala traka na kojoj je bila snimljena.

Upravo to što je cijena distribucije bila ista kao i cijena trake na kojoj je distribuirana, te priznavanje autorstva su najvažnije karakteristika prvih 10 godina Unix-a, a ujedno i razlog zašto je postao tako popularan. Tipična situacija je slijedeća: nešto je napravljeno u BTL-u i distribuirano u izvornom kodu. Korisnik u Engleskoj je napravio nešto drugo od toga. Drugi korisnik u Kaliforniji je unaprijedio englesku verziju i to distribuirao dalje. Ta unaprijeđena verzija je uključena u slijedeću BTL distribuciju. AT&T-jeva politika patenata nije mogla nikako to kontrolirati, a sustav je postajao sve bolji i više upotrebljavan.

Bill Joy, koji se bavio distribucijom, razaslao je 30 kopija BSD-a 1978 godine. Rad na vi-ju doveo ga je do koda za optimizaciju ispisivanja na različitim terminalima. Joy je odlučio napraviti interpreter za iscrtavanje ekrana na različitim terminalima, te je tako rođen termcap.

Sredinom 1978. već je mnogo toga bilo učinjeno (Pascal je bio dovoljno robustan, a uključeni su i vi i termcap), tako da je bilo vrijeme za drugu distribuciju BSD-a. Joy je sjeo, uključio poboljšanja koje su poslali korisnici i razaslao 75 kopija 2BSD.

10.7 Unix se širi

Do tada, Unix se mogao izvoditi samo na DEC PDP računalima. Međutim, 1977. Tom Lyon je preportirao neke dijelove verzije 6 na IBM 360 u Princeton-u. Slijedeće godine Ritchie i Steve Johnson te Richard Miller su portirali Unix na računalo Interdata 8/32 i 7/32. Sustav nije bio star niti deset godina, a mogao se izvršavati na različitim mašinama DEC-a i Iterdate. Verzija 7 Unix-a bila je prvi portabilni Unix.

U trenutku kada je navršavao 10 godina, Unix se upotrebljavao svugdje po svijetu: 1976. instaliran je na Univerzitetu u Tokiju, koristio se na nekoliko univerziteta u Australiji, na mnogim mjestima u Engleskoj, u Norveškoj, Njemačkoj, Francuskoj, Danskoj, Austriji i Izraelu. Sve to je postignuto bez promocije ili tehničke podrške.

Unix verzija 7, koju je BTL izdao u lipnju 1979. sadržavala je mnoga unapređenja: veliki datotečni sustav, neograničen broj korisnika i povećanu pouzdanost. Bilo je i mnogo novih komandi kao što su awk, lint, make, uucp, find, cpio i expr. *Programmer's manual* je narastao na 400 strana, a pratila su ga dva sveska po još 400 stranica svaki. Ova je verzija uključivala i potpuni Kernighan i Ritchie C prevodilac, unapređena korisnička ljuska (*shell*, sh - Bourne shell) i mnogo *include* datoteka.

10.8 Komercijalizacija

Industrijska primjena Unix-a također je dobila zamah sa pojavljivanjem verzije 7. Pojavilo se i nekoliko 32-bitnih implementacija (Xenix2 za Intelove 8086 procesore, djelo Microsofta i Santa Cruz Operation, te portovi za Zilogove Z8000 i Motoroline 68000). Pojavio se također i 3BSD potaknut pojavom verzije 7. Verzija 8 je portirala vi i termcap (Billa Joya) te curses (Kena Arnolda) od BSD-a.

Ali, sa verzijom 7 pojavili su se i prvi problemi. Andy Tanenbaum sa Free University u Amsterdamu to objašnjava ovako: "Kada je AT&T izdao verziju 7, shvatio je da je Unix vrijedan komercijalni proizvod, pa je verziji 7 dodao licencu koja zabranjuje proučavanje izvornog koda u sklopu predavanja na univerzitetima. Mnogi univerziteti su reagirali tako što su počeli predavati samo teoriju."

Tanenbaum je odlučio napisati iz početka novi operativni sustav koji bi bio kompatibilan sa Unix-om, ali bez linije koda od AT&T-a. Nazvao ga je Minix. To je bio drugi klon Unix-a (prvi je bio Idris P. J. Plauger-a).

Slijedeća verzija 4.2BSD-a bila je veliki uspjeh. U prvih 18 mjeseci razaslano je više kopija 4.2BSD-a nego svih prijašnjih BSD-a zajedno. Mnogi komercijalni operativni sustavi su bazirani na 4.2BSD-u: DEC-ov Ultrix i SUN-ov SunOS su najpoznatiji.

10.9 Sazrijevanje Unix-a

Zanimljivo je kako su različiti proizvođači prigrlili ili odbacili Unix. IBM i BTL su razvili spoj TSS/Unix (time-shering system) relativno rano, ali to nije imalo utjecaja na IBM. DEC je ignorirao Unix, dok je Hewlett-Packard, koji je baš kao i IBM i DEC, imao svoj vlastiti operativni sustav, prigrlio Unix sa velikim entuzijazmom jednako kao i Japanci. Na kraju treba spomenuti i SUN, kod kojeg je razvoj hardware-a i Unix-a bio simbiotski, i koji je propagirao BSD.

Interno je DEC za Unix imao "NIH" (NIH je skraćenica od *not invented here* - nije izumljeno ovdje). To što je Unix bio rezultat razvoja BTL-a je bilo dovoljno da odbije DEC-ove inžinjere. Sve od 1978. za većinu svijeta Unix je značilo "AT&T-jev operativni sustav" iako je za AT&T Unix značio "alat za podršku telekomunikacijama".

Kada je sudac Green odlučio o Baby Bells-ima[15], AT&T je mogao i službeno izdati Unix, pa se pojavio System V. Sredinom i krajem 1980-tih pojavio se veliki broj komercijalih dobavljača Unix-a, aplikacija i mnoštvo korisnika. Unix se probio u velike poslove, Wall Street i pravne firme.

Uskoro su se pojavile dvije rivalske skupine, Open Software Foundation i Unix International. Ubrzo su uvučeni i sudovi. Unix System Laboratories je tužio Berkely Software Design Inc., tvrdeći da je BSDI povrijedio copyright prava i poslovne tajne USL-a. Tužba je odbijena zbog toga što je AT&T u početku distribuirao Unix u izvornom kodu bez copyright-a. Pojavile su se još neke tužbe, ali na kraju se sve riješilo, i evolucija se nastavila.

---

[15] Poznati slučaj kada se AT&T morao odvojiti od Bell-a zbog anti-trustovskih zakona koji su na snazi u USA.

Solaris, HP-UX, AIX, Ultrix i ostale verzije ujedinile su se oko OSF-a. Unix je utjecao na sve operativne sustave koji se danas prodaju. Prozori, multitasking i mreže ne bi bile takve kakve su danas da nije bilo Unix-a. Kao što je to rekao Sunil Des, sa Ciry University, London: "tehnički, Unix je jednostavan, koherentan sustav koji nekoliko dobrih ideja iskorištava do maksimuma." Međutim, neke od ideja nemaju baš ništa sa operativnim sustavima: ima veze sa dijeljenjem, suradnjom, težnjom korisnika za evolucijom tehnologije koja je bila podržana od cijelog skupa istraživača i korisnika spremnih na suradnju.

Povijest Unix-a je u početku bila jednostavna, za razliku od zbrke koja nastupa u trenutku kada se Unix komercijalizirao. Evo kako izgleda povijest BSD verzija:

Datum	Naziv	Sadržaj
kraj 1977	BSD	Unix Pascal, ex (za PDP-11)
sredina 1978	2BSD	isto kao i BSD uz dodatak vi, termcap, Mail, more i csh (za PDP-11/34)
kraj 1979	3BSD	virtualna memorija, Berkeley utilities (sukladno sa 32V za VAX)
listopad 1980	4BSD	brži file system, job control, pouzdani signali, auto-reboot, delivermail, Franz Lisp (za VAX-11/750)
lipanj 1981	4.1BSD	automatska konfiguracija, poboljšanje performansi
travanj 1982	4.1a	probna verzija za ARPANET čvorove, TCP/IP i sockets
lipanj 1982	4.1b	probna verzija korištenja za predavanja na Berkeley-ju, fast file system i novi kod za podršku mreži
kraj 1982	4.1c	skoro sve kao i u 4.2 osim nove podrške za signale, od 4.1c je nastao SunOS
rujan 1983	4.2BSD	važna verzija sa mnogim promjenama, uključuje TCP/IP, ffs, novi system interface i novu podršku za signale
lipanj 1986	4.3BSD	XNS umrežavanje, poboljšanja 4.2, cache za direktorije, Internet name server
lipanj 1988	4.3-Tahoe	interne kernel mogućnosti (alokator memorije, debugger, podrška nazivima diskova), poboljšani TCP algoritmi, podrška za CCI Power 6 (Tahoe)
studeni 1988	Net-1	dio 4.3-Tahoe, uključuje mrežu, C biblioteku, uslužne programe i login preko mreže, distribuira se anonimnim ftp-om bez zahtjeva za licencom
lipanj 1990	4.3-Reno	probna verzija nadogradnji za 4.4BSD: vnode, NFS, OSI podrška, podržava VAX, Tahoe i HP 9000/300
lipanj 1991	Net-2	dio 4.4-Reno, nova podrška za virtualnu memoriju (prema Mach-u iz Utah-a) i port na Intelove procesore 386/486
lipanj 1993	4.4BSD	potpuno novo napisan sustav da bi se izbjegao AT&T-jev kod, dodana podrška za Posix i sve iz Reno i Net-2 verzije
lipanj 1994	4.4-Lite	verzija koja se rješava svog koda na koji je USL/Novell imao primjedbe, u osnovi identična 4.4BSD

10.10 Literatura

1. Peter H. Salus: Unix at 25, BYTE, October 1994, str. 75

2. Kees J. Bot, Edvard Tuinder, et al.: Minix Programmer's Manual