SVEUČILIŠTE U ZAGREBU

FAKULTET ELKTROTEHNIKE I RAČUNARSTVA

ZAVOD ZA ELEKTRONIČKE SUSTAVE I OBRADU INFORMACIJA

SEMINARSKI RAD IZ KOLEGIJA
SUSTAVI ZA PRAĆENJE I VOĐENJE PROCESA

Protokoli za udaljeni rad na računalu (SSH, telnet, rlogin)

Niko Bako
JMBAG:0036387261
INE

Sadržaj

1. Sažetak **3**

2. Telnet **3**

2.1 Uvod

2.2 Telnet protokol **3**

2.3 Komunikacija **4**

2.4 Zaključak **5**

3. Rlogin **5**

3.1 Uvod **5**

3.2 Rlogin protokol **5**

3.3 Komunikacija **6**

3.4 Zaključak **6**

4. SSH **6**

4.1 Uvod **6**

4.2 SSH protokol **7**

4.3 Enkripcija podataka **8**

4.4 Komunikacija **9**

4.5 Zaključak **10**

5. Literatura **10**

1. Sažetak

U ovom radu opisana su tri protokola: Rlogin, Telnet te SSH protokol. Dan je povijesni pregled i razlozi nastanka pojedinih protokola tokom vremena, uz opis kako se razvijala moderna komunikacija i želja ljudi za međusobnim povezivanjem. U radu se ne ulazi duboko u samu srž protokola već samo u njihovo osnovni način rada da bi čitatelji mogli shvatiti kako funkcioniraju. Još su opisane i razlike i nedostaci među protokolima njihove prednosti i nedostaci**.**

2. Telnet

2.1 Uvod

Pri kraju 60-tih godina računala su bila jako skupa,mnogo je osoba koristilo isto računalo na koje su se spajali različiti korisnici iz npr. neke organizacije.Međutim tokom vremena potražnja za računalima brzo je rasla pa su se pojavili prvi problemi.

Jedan od problema je bio da organizacije koje imaju više računala, a za svako njihovi zaposlenici imaju dozvolu korištenja, moraju imati više terminala. Tako se poslovi nisu mogli obavljati samo sa jednog mjesta nego je bila potrebna posebna prostorija sa terminalima.

Drugi, još važniji, problem je što korisnici nisu mogli komunicirati sa udaljenim računalima direktno. Da se ovaj problem otkloni prvo su korišteni posebni sklopovi između terminala i računala i to je bilo potrebno za svaku vezu, i je jako poskupljivalo sustav.

Kao rješenje ovih problema ljudi su počeli smišljati nove načine povezivanja računala i terminala i kao rezultat među prvima nastao je Telnet protokol. Prvenstveno je zamišljen 1967g. i razvijao se do konačno 1983g. kad je objavljen dokument RFC 854 koji je opisao prvu specifikaciju. Razvoj protokola trajao je dosta dugo jer u početku oprema je bila nestandardna pa se trebao riješiti problem međusobne komunikacije.

2.2 Telnet protokol

Kako je već rečeno zbog toga što ne koriste iste načine komunikacije telnet je riješio ove probleme. Protokol je napravljen na temelju tri glavna koncepta :

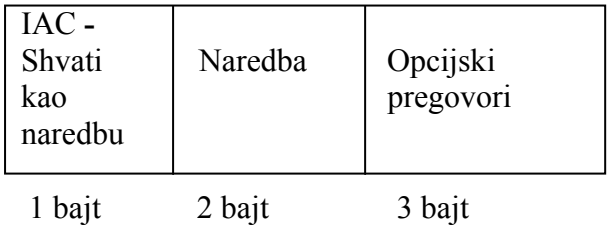
- Mrežni prividni terminal (*eng.* Network Virtual Terminal kratica NVT): Telnet definira standardizirani prividni terminal zvan NVT,kojeg koristi svaki uređaj koji komunicira. Korisnik telnet protokola uzima ulaz korisnika te ga prevodi u razumljiv format za NVT. NVT to preuzima te ga šalje serveru koji ga razumije i prevodi u sebi razumljiv jezik te može dalje s njime raditi. Proces je reverzan ali iste filozofije kada se podatci šalju sa udaljenog servera.Ovaj način omogućuje klijentu i serveru da koriste potpuno različitu opremu a da se ponovo razumiju.Posebne naredbe koje se odnose na NVT su upletene u podatke, to omogućuje klijentu i serveru da uvijek izvode najbolji način komunikacije.

- Opcije i Pregovori:Na prvi pogled se čini da su Telnet klijenti i serveri međusobno nekompatibilni, ali ovaj problem je riješen na način da klijent i server poznaju osnovnu bazu komunikacije.Međutim korištenje samo baze uvelike ograničava rad naprednih terminala i servera.Zato Telnet dopušta raznovrsni izbor mogućnosti i mehanizama međusobne komunikacije oko kojih server i klijent mogu pregovarati.

Ako se slože komuniciraju na taj način a ako ne mogu naći zajednički jezik uvijek mogu koristiti osnovnu komunikaciju koju omogućava NVT.

- Simetrične Operacije:Telnet je klijent/server protokol, posebno je napravljen da ne radi zaključke tko je server a tko klijent.Kad se veza uspostavi i server i klijent mogu slati podatke ravnopravno.Osim toga ravnopravno mogu uspostavljati zahtjeve za pregovorima. Ovo čini protokol fleksibilnim i omogućuje da se koristi na mnogo načina.

Telnet protokol je definiran u aplikacijskom sloju OSI modela,koristi TCP/IP protokol što znači da je „connected-oriented“, full-duplex. Koristi niz naredbi koje se koriste uglavnom za UNIX. Ove naredbe su zadane protokolom, sastoje se od 3 bajta od kojih je prvi bajt naredbe ima kod 255 što znači shvati kao naredbu (eng. Interpret as command -IAC). Drugi bajt je stvarna naredba dok je treći bajt „pregovaranje“. To je prikazano na slici 1.

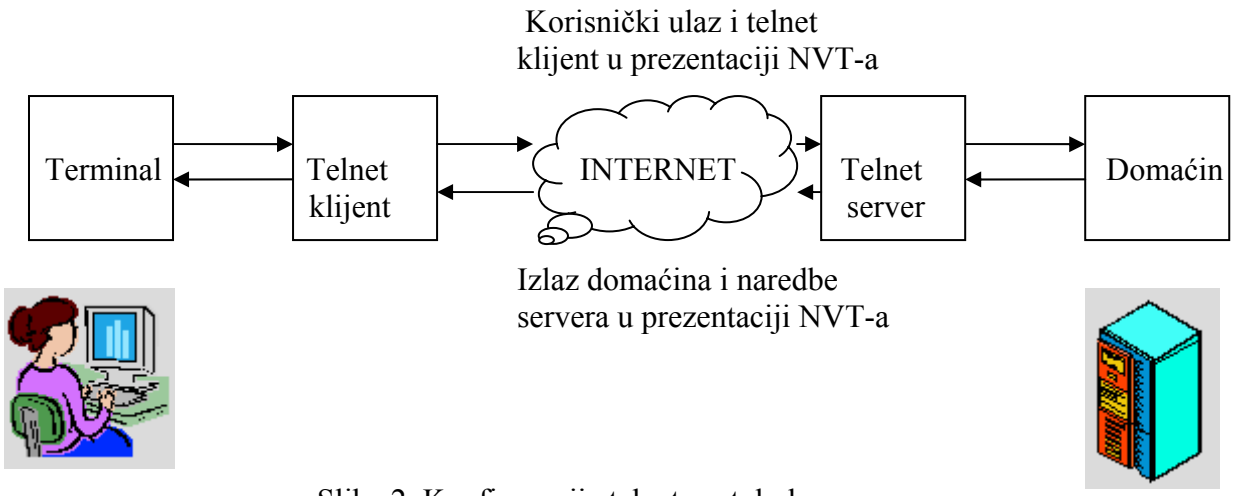


Slika 1. Struktura naredbe

Popis naredbi može se naći na popisu literature koja je korištena u radu.

2.3 Komunikacija

Na slici 2 dana je jednostavna konfiguracija telnet protokola



Slika 2. Konfiguracija telnet protokola

Klijent započinje sesiju sa upisom adrese domaćina na kojega se hoće spojiti, pri tom adresa može biti simbolička (npr. pinus.cc.fer.hr) ili fizička (upisuje se adresa u obliku brojeva). Pretpostavimo da korisnik npr. upiše adresu pinus.cc.fer.hr. Telnet server cijelo vrijeme osluškuje port 23. Kada klijent hoće pristupiti serveru na toj adresi on inicira TCP/IP protokol. Nakon što je pokrenut zahtjev za komunikacijom i nakon što server shvati da ima nešto na portu 23 nizom već prije navedenih naredbi server i klijent počinju dogovor o načinu komunikacije. Zapravo oni počinju pregovore koji će im osigurati najbolji način komuniciranja. Nakon što je dogovorena komunikacija to se javi NVT koji to omogućuje i uspostavlja jezik koji je i njemu razumljiv. Sada komunikacija može početi.

Sljedeći korak je da se korisnik registrira tj. mora upisati svoje korisničko ime i lozinku da bi server shvatio da razgovara s pravim klijentom. Recimo da korisnik hoće slati podatke. Korisnik na svojoj tipkovnici upisuje niz znakova i to on vidi na svome zaslonu, međutim sa strane servera to je samo tipkovnica koja generira znakove. Te znakove NVT sprema u svoj spremnik (eng. buffer) i čeka da se on napuni. S druge strane NVT vidi server kao obični printer na kojega mora poslati znakove. Kad je spremnik pun podatci se pošalju na server i na taj način se prenose. Isti princip vrijedi i za slanje podataka sa servera na terminal jer, kako je prije navedeno, protokol ne pretpostavlja tko je server a tko klijent. Još treba napomenuti da su mogući pregovori tokom komunikacije u svrhu ostvarivanja što bolje veze.

2.4 Zaključak

Iako je Telnet protokol vrlo star danas se ipak koristi na UNIX strojevima. Zahvaljujući činjenici da je jako fleksibilan te da koristi TCP/IP protokol danas je ideja protokola usvojena od mnogih aplikacija tipa FTP,HTTP i sličnih. Razlog što je preživio kroz godine je u tome da ne pretpostavlja tko je klijent a tko server. Međutim unatoč ovim prednostima glavni nedostatak je razina sigurnosti, kao i činjenica da NVT ne prosljeđuje podatke odmah na server nego ih čuva u međuspremniku što ga čini nepogodnim za neke aplikacije.

3. Rlogin

3.1 Uvod

Rlogin je nastao otprilike negdje u isto vrijeme kao i Telnet. Razlozi nastanka Rlogina su bili u tome jer Telnet nije omogućavao korištenje svih opcija kako se razvijao dugi niz godina. Zbog sličnosti u naredbama sa Telnetom nazivaju ga još i njegovim rođakom.

Razvijen je na Berkeleyu kao skup naredbi za spajanje na udaljena računala. Prvo je objavljen kao 4.2BSD a zatim kao RFC 1254 dokument gdje su objavljene specifikacije protokola.

3.2 Rlogin protokol

Rlogin omogućava udaljeni rad sa kontrolom toka podataka sa trenutnim pražnjenjem spremnika mrežnog terminala (ovaj terminal i spremnik su opisani u telnet protokolu). Jako je raširen među UNIX strojevima jer omogućuje bolji transport nego Telnet. Osim ove pogodnosti

posebno je popularan među UNIX strojevima jer se oni mogu konfigurirati da ne koriste lozinke.

Rlogin zahtjeva uporabu TCP/IP protokola tj. definiran je na aplikacijskom sloju OSI modela. Koristi port 513 i prenose se 8 bitni podatci.

3.3 Komunikacija

Rad je poprilično jednostavan za korisnika. Korisnik prvo pošalje null znak na server i zatim upiše svoje korisničko ime, korisničko ime servera , brzinu komunikacije te tip servera. Eksplicitno:

<null>
korisničko ime klijenta<null>
korisničko ime servera<null>
tip terminala/brzina<null>

Korisnik počinje rad u tzv. «cooked» modu. Tu šale START i STOP ASCII znakove koji govore udaljenom serveru što korisnik hoće od njega. Ovdje je bitan redoslijed znakova jer o njima ovise tražene operacije i na ovaj način ostvaruje se kontrola toka podataka. U «raw» modu START i STOP se shvaćaju kao obični znakovi i oni se tako interpretiraju.

Kad server šalje podatke šalje ih kao niz od 8 bitova. Normalni znakovi se odmah prosljeđuju na zaslon dok se posebni znakovi(npr. TAB) prvo moraju obraditi. Server također može poslati 8 bitnu kontrolnu poruku na način da je ubaci u niz podataka i da to javi TCP/IP protokolu.

3.4 Zaključak

Kako vidimo iz prije opisanog načina komunikacije rlogin je jednostavniji od Telneta. Budući da ne treba pregovore o načinu komunikacije moguće je ostvariti čistu 8 bitnu vezu. Druga važna prednost nad Telnetom je da nema spremnika, dok telnet čeka da mu se napuni spremnik i tek tada ga isprazni. Dakle svi znakovi koji se pojave na tipkovnici klijenta odmah se pojave i na strani servera. Ovo protokol čini pogodnim za programe koji trebaju tipkovnicu.

Nedostatak protokola je sigurnost jer ne zahtjeva lozinku pa je ograničen samo na povjerenje i zbog toga se koristi samo u akademskim zajednicama.I kao drugi nedostatak je da se može koristiti među opremom istog proizvođača jer nema zajedničkog jezika komunikacije među opremom raznih proizvođača.

4. SSH

4.1 Uvod

SSH je novi protokol, razvio ga je Tatu Ylonen 1995g. koji je radio na Hesinškom sveučilištu tehnologije. Razlog nastanka je krajnje zanimljiv naime protokol je nastao nakon što

je sveučilišna mreža postala žrtvom napada hakera koji su pokrali sve šifre. Ylonen je zato razvio prvu verziju SSH protokola koja je postala jako popularna i u drugim zemljama izvan Finske. Što je zapravo SSH?

SSH je mrežni servis koji korisnicima omogućuje uspostavu sigurnog komunikacijskog kanala između dva računala putem nesigurne računalne mreže. Tradicionalni servisi kao što su rlogin, telnet i drugi, iako su krajnje jednostavni imaju veliki nedostatak koji predstavlja ograničenje za njihovu uporabu. Taj nedostatak je niski nivo sigurnosti koji je ugrađen u većinu takvih servisa, budući da se u prošlosti nije pazilo na sigurnost mreža. Te slabosti ovih servisa omogućuju neovlaštenim korisnicima da na lagan način dođu do korisničkih informacija te potpuno prisluškivanje kompletnih sesija među korisnicima. Također ne postoji mehanizam za autentikaciju računala,što neovlaštenim korisnicima omogućuje lažno predstavljanje te izvođenje sesija pod drugim imenom.

Ovi navedeni problemi danas su od vrlo velikog značaja i njih u najboljoj mjeri rješava SSH protokol.Rad protokola temelji se na korištenju simetrične i asimetrične enkripcije podataka o kojima će biti više riječi u nastavku.

4.2 SSH protokol

SSH protokol sadrži tri glavne komponente:

- Transportni sloj: Ovaj sloj je sigurni transportni protokol. Omogućava jaku enkripciju i kriptografsku autentifikaciju domaćina .Također može omogućiti kompresiju ali to se zadaje opcijski.Ovaj protokol se koristi preko TCP/IP veze, ali se može koristiti na bilo kojem sigurnom nizu podataka.

- Autentifikacija korisnika:Provjerava klijenta koji se spaja na server.Nalazi se na transportnom sloju.

- Sloj za povezivanje: Multipleksira enktiptirane kanale.Nalazi se na sloju za autentifikaciju.

Klijent pošalje zahtjev za korištenje nakon što je uspostavljena veza preko transportnog sloja.Drugi zahtjev od korisnika za radom je nakon što je napravljena autentifikacija. Ovo omogućuje da se uspostave novi protokoli sa ovim gore navedenim.

Protokol za povezivanje omogućuje sigurne kanale za povezivanje kao što je TCP/IP protokol.

Najvažniji dio arhitekture protokola su ključevi domaćina(eng. Host keys) tj.servera . Ovaj ključ se koristi da se provjeri da korisnik stvarno komunicira sa pravim serverom, a da bi ovo bilo moguće korisnik mora znati javni ključ servera. Iz ovog je vidljivo da se mora imati neka razina povjerenja,pa su moguća dva pristupa:

-Klijent ima javne ključeve servera u svojoj bazi podataka.Ova metoda ne zahtjeva treću stranku koja bi vršila razmjenu među klijentom i serverom, ali nedostatak je što je jako ranjiva na napade hakera.

-Ključ se povjeri nekoj organizaciji u koju se ima povjerenja. Klijent tada zna samo ključ od ove organizacije i može koristiti ključeve koje ona čuva.

Ova druga metoda omogućava veću sigurnost ali se prvo treba prijaviti organizaciji u koju treba imati povjerenja a zatim se spojiti na server.

Duljina podataka je 28 bajta i ovisi o pregovaračkom algoritmu. Povećanje je zanemarivo za velike pakete ali je znatno osjetno na 1 bajtnim paketima.Tako za TCP/IP paket imamo povećanje sa 33 na 51 bajt, za Ethernet najveće povećanje iznosi 5 bajta. . Sve ovo rečeno o javnom ključu nas uvodi u enkripciju podataka što spada u najzanimljivi dio protokola dio protokola.

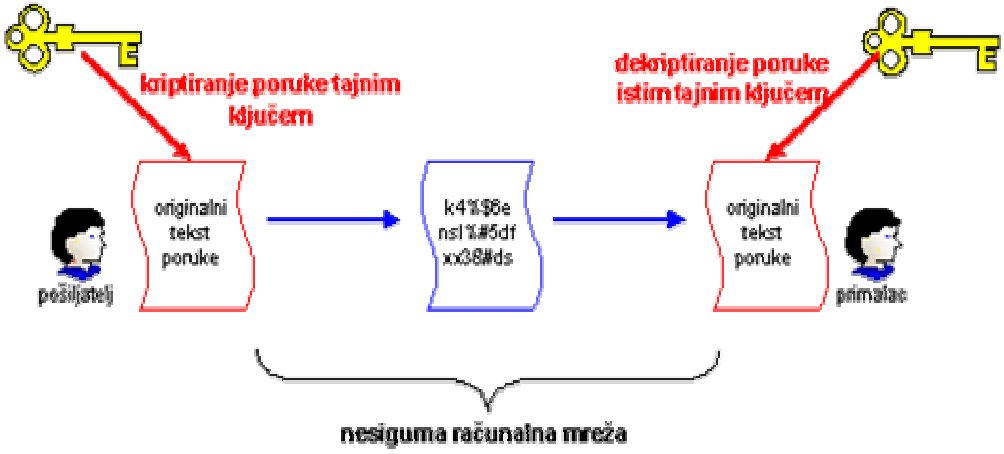
4.3 Enkripcija podataka

Za enkripciju se koriste dvije osnovne grupe algoritama:

-simetrični(bazirani na jedinom tajnom ključu)
-asimetrični(bazirani na javnom i tajnom ključu)

Obje navedene metode baziraju rad na korištenju javnog algoritma za enkripciju. Osim algoritma, za kriptiranje je potrebno znati odgovarajući ključ na temelju kojega se provodi dekriptiranje. U ovisnosti o algoritmu imamo simetričnu i asimetričnu kriptografiju.

Kod simetrične kriptografije isti ključ se koristi za kriptiranje i dekriptiranje. Pošiljalac kriptira podatke tajnim ključem i taku poruku šalje primaocu. Primalac tim ključem dekriptira poruku i dolazi do njena sadržaja.Na slici 3. je prikazana simetrična kriptografija.



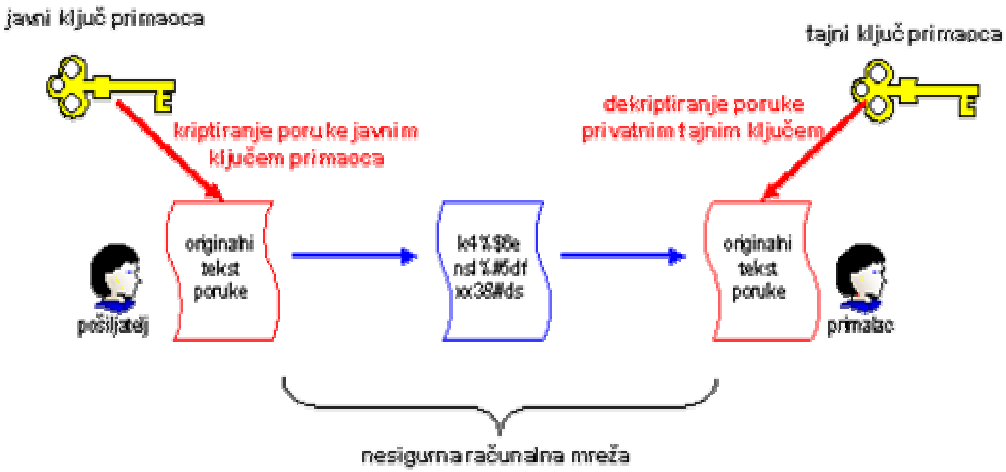
Slika 3. Simetrična kriptografija

Glavni nedostatak ovog kriptiranja je što se primalac i pošiljalac moraju dogovoriti oko ključa. Ovaj dogovor je poseban problem jer se mora obaviti preko telefona, maila i slično,što narušava privatnost ključa. Glavna prednost algoritma je što su brzi pa se često koriste. Predstavnici su DES,IDEA,3DES i drugi.

Asimetrični algoritmi svoj rad baziraju na korištenju dva ključa.Svaki korisnik ima svoj javni i tajni ključ koji su međusobno kompatibilni tako da se poruka kriptirana jednim ključem može dekriptirati samo drugim. Tajni ključ je povjerljiv i zna ga samo vlasnik dok javni pozna svatko. Postupak razmjene je sljedeći:

-poruka se kriptira javnim ključem
-primalac poruku dekriptira svojim tajnim ključem

Znači poruku može primiti svatko,uz uvjet da zna javni ključ, a dekriptirati se može jedino tajnim ključem primaoca. Javni ključevi se distribuiraju putem web-a ili nekog drugog servisa, time je uklonjen nedostatak simetrične kriptografije. Nedostatak je što treba veće snaga računala i duže trajanje dekripcije. Asimetrična kriptografija je prikazana na slici 4.

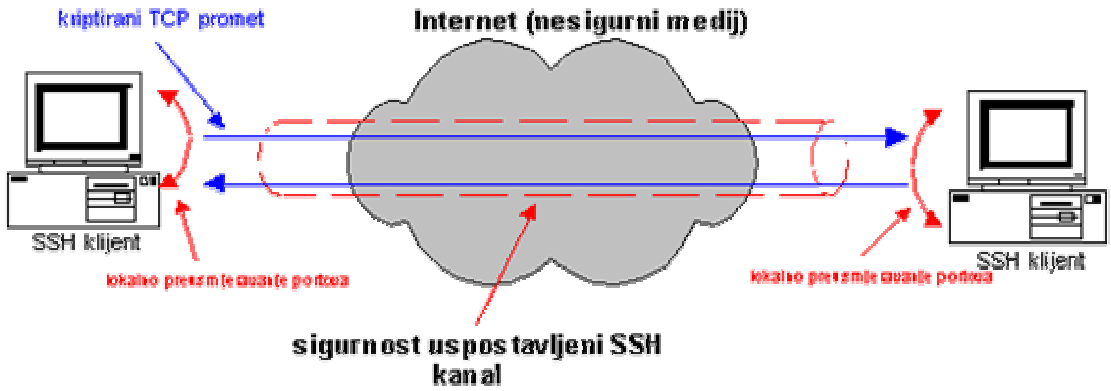


Slika 4. Asimetrična kriptografija

Današnje metode koriste prednosti obje dvije metode.

4.4 Komunikacija

Komunikacija se obavlja u nekoliko faza. Prva faza klijent i server razmjenjuju podatke o inačici protokola koji će koristiti. U idućoj fazi dogovaraju se o algoritmu i tajnom ključu koji će koristiti. Nakon ove faze dogovoren je ključ sesije i svi podatci biti će kriptirani na ovaj način. U trećoj fazi klijent se auteticira kod servera kako bi se utvrdio identitet klijenta,ukoliko je sve dobro prošlo prihvaća se sesija i razmjena podataka može početi. Komunikacija je prikazana simbolički na slici 5.



Slika 5. Uspostava komunikacije

4.5 Zaključak

Iz svega navedenog vidi se da je SSH protokol dosta kompliciran i da je još u fazi razvoja. Uz ovaj nedostatak glavna mana je da se puna verzija protokola plaća. Međutim uz sve ovo protokol ima glavnu prednost a to je velika sigurnost.Tu posebno valja spomenuti da je moguće preusmjeriti TCP/IP nesigurne portove (POP,FTP) na sigurni SSH kanal.

5. Literatura

[1] www.ssh.com
[2] www.carnet.hr
[3] History of SSH (SSH, The Secure Shell: The Definitive Guide)
[4] RFC 854 (rfc854) - Telnet Protocol Specification
[5] The TCP/IP Guide - Telnet Protocol
[6] RFC 1258 (rfc1258) - BSD Rlogin