CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

SSH protokol

CCERT-PUBDOC-2009-08-272

+CERT.hr u suradnji s LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom CARNet CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

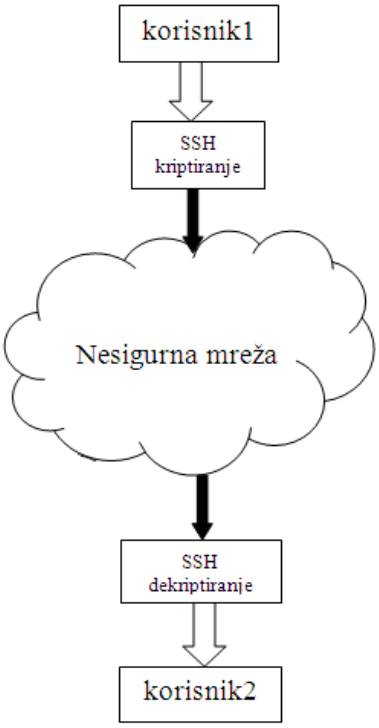
Prvi internetski protokoli nisu bili oblikovani s posebnim naglaskom na sigurnosti podataka koji putuju mrežom jer za time nije postojala ozbiljna potreba. Internet se tada koristio uglavnom u akademskim institucijama ili vojsci i sličnim zaštićenim okruženjima. S vremenom, mreža se počela široko rabiti, a danas gotovo svatko ima pristup Internetu. To dovodi do problema sigurnosti podataka koji se razmjenjuju u njemu. TELNET (eng. Telephone Network), FTP (eng. File Transfer Protocol), rsh (eng. remote shell), rcp (eng. remote copy) i drugi protokoli, kojima se razmjenjuju podaci između udaljenih računala, podatke šalju u nekriptiranom obliku, izložene napadačima. SSH protokol razvijen je kao zamjena za postojeće nesigurne protokole, a podatke štiti enkripcijom. Oblikovan je prema modelu klijent/poslužitelj. Krajnje točke komunikacije smatraju se sigurnima, dok se mreža koja ih povezuje smatra nesigurnom. Tajnost, autentičnost i integritet podataka osiguravaju se primjenom snažnih kriptografskih metoda.

Iako je dostupno više programskih izvedbi ovog protokola, daleko se najčešće koristi, posebno na operacijskim sustavima UNIX/Linux, paket OpenSSH. Riječ je o besplatnoj izvedbi SSH klijenta i poslužitelja koja omogućuje korištenje većeg broja dodatnih mogućnosti protokola kao što su: SSH tuneliranje i uspostavljanje VPN veze preko SSH kanala. Zbog raširene uporabe paketa OpenSSH, čak i u slučaju odabira neke druge izvedbe, poželjno je voditi računa o usklađenosti te izvedbe s OpenSSH ostvarenjem.

U ovom dokumentu dan je uvod u razvoj, arhitekturu i mogućnosti korištenja SSH protokola. Osim toga navedeni su neki osnovni načini uporabe protokola pomoću dostupnih programskih alata.

2. Povijest nastanka i razvoj SSH protokola

SSH (eng. Secure Shell) protokol je nastao 90-ih godina prošlog stoljeća kao zamjena za druge, nesigurne, protokole, poput rlogin, rsh, TELNET i FTP, koji putem računalne mreže razmjenjuju podatke. SSH za razliku od postojećih protokola uvodi zaštitu tajnosti podataka. Naime, kod drugih sličnih protokola podaci se kroz mrežu šalju u otvorenom (nekriptiranom) obliku i bilo koji korisnik može ih presresti, pročitati ili čak mijenjati. SSH podatke kriptira prije slanja i dekriptira nakon primitka čime se onemogućuje njihovo otkrivanje dok se kreću mrežom.



Slika 1. SSH komunikacija

2.1. Temelji sigurnosti SSH protokola

Sigurnost SSH protokola temelji se na uporabi kriptografskih metoda koje omogućuju zaštitu (tajnost) podataka koji se kreću kroz nesigurnu mrežu. Osim toga, spomenute metode mogu se iskoristiti za provjeru identiteta korisnika koji sudjeluju u komunikaciji te za zaštitu podataka od neovlaštenih izmjena, odnosno očuvanje njihova integriteta. U ovom dijelu dokumenta daje se kratak uvid u osobitosti metoda na kojima se zasniva SSH protokol.

2.1.1. Simetrična enkripcija podataka

Podaci koji se razmjenjuju između dva udaljena korisnika kroz nesigurnu mrežu moraju se zaštiti enkripcijom. U tu svrhu najčešće se koriste simetrični algoritmi kriptiranja. Riječ je o matematičkim postupcima koji mijenjaju ulazni niz po nekom ključu tako da je bez poznavanja tog ključa praktično nemoguće otkriti izvorni niz. Pritom se isti ključ koristi kod enkripcije i dekripcije podataka. Sigurnost ovakvih algoritama temelji se na tajnosti ključa, a zbog činjenice da se isti ključ koristi za kriptiranje i dekriptiranje, algoritmi se nazivaju *simetričnima*. Postoji čitav niz algoritama koji se koriste za simetričnu enkripciju, a među najčešće korištenima su:

- AES,
- DES,

- 3DES,
- Blowfish i
- Arcfour.

Svi navedeni algoritmi dostupni su za korištenje u različitim SSH izvedbama (DES nije u OpenSSH). Ovakav način enkripcije može se smatrati dovoljno sigurnim od tzv. „brute force" napada dok je tajni ključ poznat samo ovlaštenim stranama. Pritom je preporučljivo odabrati što duži tajni ključ jer se tako smanjuje vjerojatnost da će ga napadač pogoditi. Osim toga, simetrična enkripcija je zbog svoje brzine pogodna za zaštitu većih količina podataka.

Kod simetrične enkripcije podataka u nesigurnoj mreži javlja se problem dogovora oko vrijednosti tajnog ključa i njegove razmjene među korisnicima. U ovu svrhu moguće je koristiti Diffie-Hellman algoritam koji se koristi i kod SSH protokola. Primjena Diffie-Hellman algoritma u okviru SSH protokola opisana je u odgovarajućim RFC dokumentima (RFC4253 - The Secure Shell Transport Layer Protocol), a općenito se njegova sigurnost zasniva na različitim težinama računalnog izračunavanja logaritma i potencije broja. Prilikom razmjene ključa koristi se matematička operacija potenciranja prihvatljive vremenske zahtjevnosti. Za razbijanje ključa potrebno je izvesti operaciju logaritmiranja što je za dovoljno velike ključeve praktički neizvedivo u prihvatljivom vremenu.

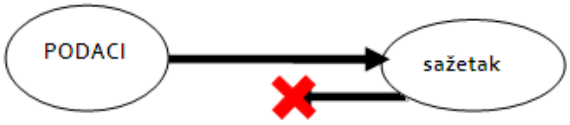
2.1.2. Asimetrična enkripcija podataka

Za razliku od simetrične enkripcije, asimetrični algoritmi enkripcije koriste dva ključa, jedan je javni i smije biti poznat svim korisnicima, a drugi je privatni i smije biti poznat samo ovlaštenim korisnicima. Bilo kakav sadržaj kriptiran javnim ključem moguće je dekriptirati jedino tajnim ključem, a sadržaj kriptiran tajnim ključem moguće je dekriptirati jedino odgovarajućim javnim ključem. Za razliku od simetrične enkripcije, asimetrična je složenija i traje bitno dulje pa ju nije preporučeno koristiti za kriptiranje većih količina podataka. Simetrični tajni ključevi imaju duljine od sto do dvjesto bitova, dok asimetrični imaju preko tisuću i dvije tisuće bitova, što utječe i na složenost kriptiranja. Ovi se algoritmi zato koriste samo za razmjenu tajnog ključa na početku komunikacije. Naime, kriptiranje vrijednosti tajnog ključa javnim ključem korisnika osigurava da nitko osim ciljanog korisnika neće biti u stanju otkriti vrijednost tog ključa (jer mu je za to potreban pripadni privatni ključ). S druge strane javni ključ korisnika može doznati bilo tko što znači da korisnik koji je primio kriptirane podatke u ovom slučaju ne može znati od koga ih je primio. Autentičnost pošiljatelja i primatelja može se osigurati korištenjem funkcija za sažimanje (eng. hash) i digitalnim potpisima. Najčešće korišteni asimetrični algoritmi, podržani i u SSH protokolu, su

- DSA (eng. Digital Signature Algorithm) i
- RSA (Rivest, Shamir i Adlerman – tvorci algoritma).

2.1.3. Digitalni potpisi i certifikati

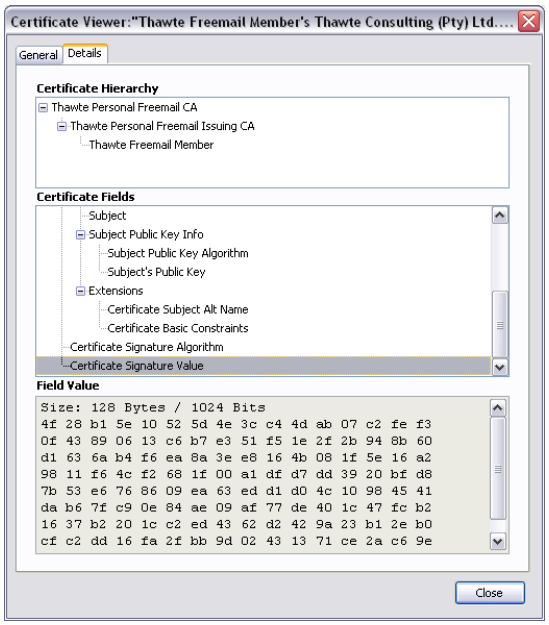
Digitalni potpisi koriste se za osiguravanje autentičnosti korisnika u komunikaciji. Digitalni potpis stvara se pomoću funkcija sažimanja (tzv. „hash" funkcija) koje nemaju povratnu funkciju, odnosno iz sažetka nije moguće izračunati izvorni tekst. Ako korisnik, koji šalje podatke, primatelju pošalje i njihov sažetak kriptiran vlastitim privatnim ključem osigurat će autentičnost i integritet podataka. Zato se takav dodatak poruci naziva i digitalni potpis. Naime, bilo koji korisnik koji presretne poslanu poruku može otkriti sažetak pomoću pošiljateljeva javnog ključa, no bilo kakva izmjena teksta zahtijevala bi izračun novog sažetka i kriptiranje sažetka privatnim ključem pošiljatelja kojeg napadač ne posjeduje. Znači ako podaci dođu do primatelja u netaknutom obliku, on može dekriptirati njihov sažetak javnim ključem pošiljatelja, izračunati sažetak pristiglih podataka i usporediti ih. Ako su jednaki, subjekt naveden kao pošiljatelj zaista jest pošiljatelj, a podaci sigurno nisu mijenjani u prometu. U ovom slučaju tajnost nije očuvana, ali se lako može očuvati dodavanjem prethodno opisanih metoda enkripcije. Autentifikacija korisnika u SSH protokolu može se obaviti algoritmima izrade digitalnog potpisa koji se temelje na RSA i DSA metodama.



Slika 2. Jednosmjernost funkcije sažimanja

2.1.4. PKI certifikati

Provjera identiteta korisnika može se obavljati pomoću digitalnih certifikata. Pouzdana neovisna tijela izdaju certifikate kojima se jamči veza između javnog ključa i njegovog korisnika. Na taj način onemogućuje se lažno podmetanje javnog ključa u ime nekog korisnika. Certifikati zapravo sadrže identifikator i javni ključ korisnika te digitalni potpis certifikacijskog centra. Budući da je certifikacijski centar pouzdana strana čiji je javni ključ dostupan na pouzdanim odredištima nema mogućnosti podmetanja lažnog javnog ključa certifikacijskog centra. Na ovaj način sasvim sigurno se može utvrditi veza između korisnika i njegovog javnog ključa. SSH protokol podržava PKI (eng. Public Key Infrastructure) autentifikaciju na temelju X.509 digitalnih certifikata[5].



Slika 3. Primjer digitalnog certifikata

2.2. Povijest razvoja SSH protokola

Prva inačica protokola SSH razvijena je 1995. godine na sveučilištu Helsinki University of Technology u Finskoj. Njezin tvorac je Tatu Ylönen, a novi protokol trebao je zamijeniti rlogin, TELNET, rsh i slične protokole koji osjetljive podatke šalju u otvorenom obliku. Osim toga, SSH je trebao jamčiti i autentičnost korisnika koji komuniciraju. Prvotno je protokol definiran i programski ostvaren za osobne potrebe, a na tržištu je objavljen kao potpuno besplatan proizvod. No budući da je program u vrlo kratkom vremenu stekao velik broj korisnika, osnovana je tvrtka SSH Communications Security čiji je cilj bio razvoj i distribucija SSH alata. Danas se prva inačica protokola naziva SSH-1, a njezina upotreba nije preporučljiva zbog sigurnosnih propusta koje sadrži. Godine 1998. otkriven je propust u inačici SSH 1.5 protokola koji napadaču omogućuje umetanje sadržaja u kriptirani tok podataka zbog nedovoljne provjere integriteta poruke (CRC-32 metoda).

Godine 1996. izašla je SSH-2 inačica protokola koja uvodi bitna sigurnosna poboljšanja u odnosu na SSH-1 inačicu. Pritom je bitno napomenuti da SSH-2 protokol nije usklađen s SSH-1 protokolom, tj. arhitektura komunikacije je drukčija pa nije moguće uspostaviti SSH kanal između SSH-1 klijenta i SSH-2 poslužitelja (ili obrnuto). Među najznačajnijim poboljšanjima su korištenje Diffie-Hellman protokola za

razmjenu simetričnog ključa i snažna provjera integriteta poruke MAC (eng. Message Authentication Codes) kodovima. Osim toga, SSH-2 inačica omogućuje i uspostavljanje proizvoljnog broja sjednica na udaljenom računalu pomoću samo jedne SSH veze.

Nakon osnivanja tvrtke SSH Communications Security, programske izvedbe protokola od slobodnih su proizvoda postale komercijalne.

Slika 4. SSH Communications Security logo

Zato je 1999. organizacija OpenBSD razvila vlastiti alat kojim se omogućuje besplatno korištenje SSH protokola, a nazvan je OpenSSH. Prvotno je razvijen samo za operacijski sustav OpenBSD, a kasnije su razvijene inačice i za ostale operacijske sustave (Linux, Solaris, FreeBSD, HP-UX, AIX, Cygwin). OpenSSH je danas najčešće korišten SSH alat.

Godine 2006. SSH je postao IETF standard, a glavni dijelovi njegove arhitekture opisani su u sljedećim RFC (eng. Request For Comments) dokumentima:

- RFC4251 - The Secure Shell (SSH) Protocol Architecture,
- RFC4252 - The Secure Shell (SSH) Authentication Protocol,
- RFC4253 - The Secure Shell (SSH) Transport Layer Protocol i
- RFC4254 - The Secure Shell (SSH) Connection Protocol.

Godine 2008. otkrivena je ranjivost u protokolu koja je napadaču u posebnim uvjetima omogućavala otkrivanje do četiri bita teksta iz SSH podatkovnog toka. Problem je ispravljen izmjenom pojedinih pretpostavljenih načina enkripcije u inačici OpenSSH5.2.

2.2.1. Ranjivost SSH-1 protokola

Kao što je spomenuto, SSH-1 inačica protokola sadrži ozbiljnu ranjivost koju napadač može iskoristiti za pokretanje proizvoljnih naredbi na SSH poslužitelju. Uvjet za izvođenje napada je otkrivanje uspostavljene SSH veze između klijenta i poslužitelja ili neovlašteno korištenje TCP sjednice. Potom je moguće podmetnuti posebno oblikovan paket kojim će se izvršiti napad. Naime, zbog slabe provjere integriteta CRC-32 metodom, moguće je podmetnuti valjani CRC-32 kod u izmijenjeni paket. Osim toga, manipulacijom pojedinim bitovima koji se zapisuju kao dopuna na kraj paketa, napadač može osigurati dekriptiranje dijela podataka u proizvoljan čisti tekst.

Scenarij uspješnog napada, nakon što je otkrivena SSH veza, uključuje sljedeće korake:

- presretanje paketa,
- izmjenu paketa na poseban način kojim će osigurati dekriptiranje dijela podataka u proizvoljnu naredbu (bez poznavanja ključa) i
- podmetanje novog CRC-32 koda kojim će se integritet poruke lažno prikazati očuvanim.

Kad paket stigne na odredište, dekriptira se i provjerava se integritet poruke. Budući da je on naizgled očuvan, pokreće se izvršavanje dobivenih naredbi, uključujući i podmetnutu napadačevu naredbu. Jedini način zaštite od ove ranjivosti je nadogradnja na SSH-2 inačicu protokola.

2.2.2. Usporedba SSH-1 i SSH-2 protokola

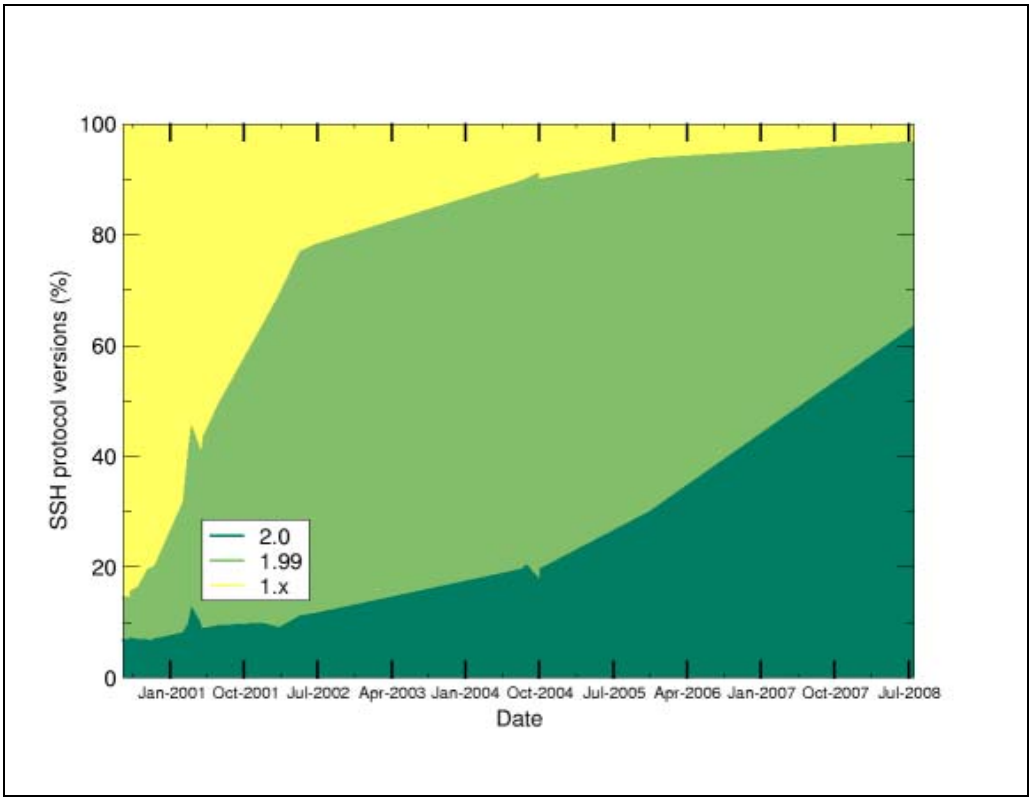
Iako danas sve više prevladava uporaba SSH-2 protokola, zbog postojanja programa koji ne podržavaju novu inačicu tog protokola ponekad nije moguće izbjeći SSH-1 protokol. Zbog toga je poželjno poznavati slabosti SSH-1 protokola u odnosu na SSH-2 inačicu.

SSH-1	SSH-2
jedinstveni protokol	troslojna arhitektura
slaba i ranjiva CRC-32 zaštita integriteta	snažna MAC zaštita integriteta
moguća jedna sjednica po vezi	moguće više sjednica po jednoj SSH vezi
ne omogućuje promjenu lozinke	omogućuje promjenu lozinke
unaprijed dogovoreni algoritmi enkripcije, autentifikacije i provjere integriteta	omogućuje dogovaranje algoritama enkripcije, autentifikacije, MAC provjere
podržava veći broj metoda autentifikacije (TIS, Kerberos)	podržava samo autentifikaciju lozinkom, javnim ključem ili preko popisa prihvatljivih poslužitelja
ključ sjednice dogovara se preko poslužiteljskog ključa	Diffie-Hellman razmjena sjedničkog ključa
ne podržava PKI certifikate	podržava PKI certifikate
moguće više zahtjeva za autentifikacijom po sjednici	samo jedna autentifikacija po sjednici
periodična zamjena sjedničkog ključa	nije moguća periodična zamjena sjedničkog ključa

Tablica 1. Usporedba SSH-1 i SSH-2 protokola

Izvor: SSH: The Secure Shell, The Definitive Guide

Na slici koja slijedi prikazan je porast uporabe SSH-2 protokola u odnosu na SSH-1 inačicu. Očito je kako od 2000. godine značajno raste uporaba SSH-2 protokola. No, budući da mnogi alati još uvijek podržavaju SSH-1 protokol, često se koriste poslužitelji s mogućnošću komunikacije i sa SSH-1 i sa SSH-2 klijentima. Oni nose oznaku inačice protokola 1.99.

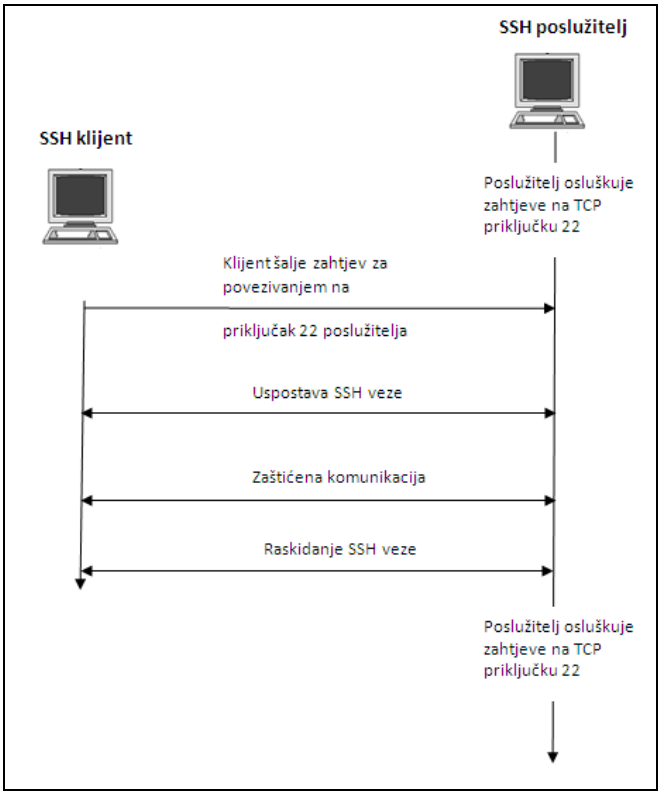


Slika 5. Kretanje udjela inačica SSH-1 i SSH-2 u uporabi

Izvor: OpenSSH

3. Unutarnja arhitektura protokola

SSH se temelji na modelu klijent/poslužitelj što znači da se komunikacija odvija između dvije različite strane. Poslužitelj s jedne strane osluškuje zahtjeve na unaprijed zadanom mrežnom priključku (eng. port), a klijent ih po potrebi šalje poslužitelju. SSH poslužitelj osluškuje zahtjeve klijenata na TCP priključku 22.



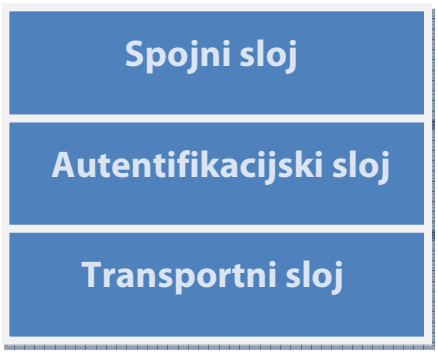
Slika 6. Shema SSH modela klijent/poslužitelj

3.1. Slojevita arhitektura SSH komunikacije

Uspostava komunikacije i sama komunikacija u SSH protokolu može se opisati troslojnom arhitekturom:

1. **Transportni sloj** (eng. Transport Layer Protocol – RFC4253),
2. **Autentifikacijski sloj** (eng. Authentication Protocol – RFC4252) i
3. **Spojni sloj** (eng. Connection Protocol – RFC4254).

Arhitektura protokola SSH i svaki sloj zasebno detaljno su opisani u odgovarajućim RFC dokumentima. U nastavku dokumenta dana su njihova osnovna obilježja. Slojevi se nadograđuju jedan na drugog kao što je prikazano na sljedećoj slici, a najniži, transportni sloj, nadograđuje se na TCP/IP mrežu.



Slika 7. Troslojna SSH arhitektura

3.1.1. Transportni sloj

Transportni sloj nadograđuje se najčešće na TCP/IP mrežu, ali ne nužno. Može se koristiti i neka druga arhitektura koja jamči pouzdan prijenos podataka na nižim slojevima mrežne arhitekture. To znači da se može pretpostaviti primitak poslanih paketa na odredištu bez uvođenja dodatnih provjera i potvrda primitka na višim slojevima mrežne arhitekture. U ovom slučaju pouzdanost ne znači tajnost ili autentičnost već jednostavno učinkovitost u prijenosu podataka. Ovaj sloj SSH protokola osigurava snažnu enkripciju i zaštitu integriteta podataka te autentifikaciju poslužitelja, a omogućeno je i sažimanje podataka radi bržeg i jednostavnijeg prijenosa. Na ovom sloju klijent i poslužitelj određuju i metode razmjene ključeva, simetrične i asimetrične algoritme koji će se koristiti te funkcije sažimanja (eng. hash) i algoritme utvrđivanja autentičnosti poruka. U okviru transportnog sloja koristi se i Diffie-Hellman metoda razmjene simetričnog ključa.

Za razmjenu podataka nakon uspostavljene SSH veze na transportnom se sloju koristi binarni paketni protokol (eng. Binary Package Protocol). Riječ je o binarnom protokolu kojim se komunikacija odvija pomoću posebno organiziranih nizova bitova koji predstavljaju pakete. Svaki paket sastoji se od određenih dijelova čija funkcionalnost je prikazanih na slici:

Slika 8. Paket binarnog paketnog protokola

Podaci se nadopunjuju do neke pogodne duljine, često povezane s algoritmom kriptiranja koji se koristi, npr. ako se kriptiraju blokovi od 512 bita, podaci se nadopunjuju na prvi višekratnik tog broja veći od duljine podataka. Na početak paketa zapisuju se duljina dopune te ukupna duljina koja uključuje zbroj duljine podataka, dopune i duljine dopune. Ovi podaci se kriptiraju prije slanja. Istovremeno se izračunava MAC vrijednost koja se ne kriptira, a osigurava integritet poslanih podataka.

3.1.2. Autentifikacijski sloj

Autentifikacijski sloj omogućuje provjeru identiteta klijenta na poslužitelju, a komunikacija na tom sloju uspostavlja se tek nakon što je komunikacija na transportnom sloju već uspostavljena. Autentifikacija klijenta može se obaviti na više načina (koje poslužitelj predlaže, a klijent odabire). Neke od tih metoda su:

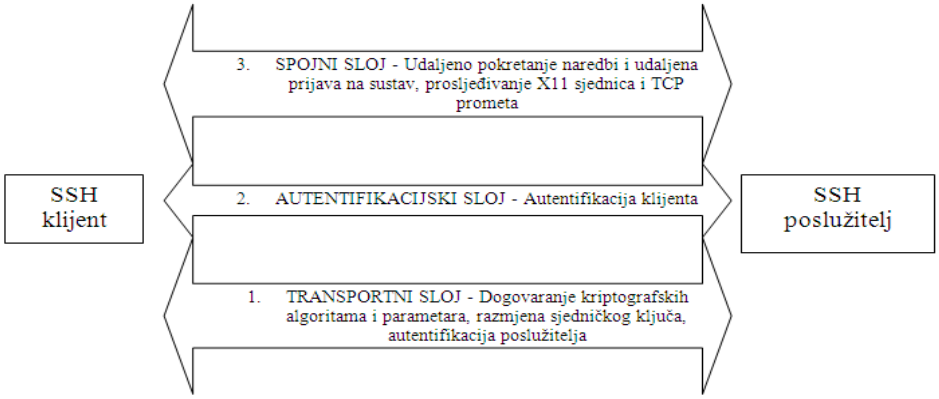
1. Putem lozinke (koja se kriptirana šalje SSH kanalom),
2. PKI (eng. Public Key Infrastructure) metode autentifikacije koje se temelje na digitalnim potpisima i asimetričnim kriptografskim algoritmima (RSA, DSA), uključujući i provjeru putem X.509 certifikata,
3. Autentifikacija zasnovana na provjeri klijenta u bazi računala kojima je dopuštena autentifikacija, a koja se nalazi na poslužitelju. Poslužitelj nakon što zaprimi zahtjev za autentifikacijom klijenta provjerava njegov FQDN (eng. Fully Qualified Domain Name) i digitalni potpis te utvrđuje radi li se o računalu kojem je dopuštena autentifikacija i valjanost primljenog digitalnog uzorka.

Ovaj sloj ostvaruje jedinstveni SSH komunikacijski kanal preko kojeg se može provesti sljedeći sloj arhitekture, tzv. „spojni sloj".

3.1.3. Spojni sloj

Na spojnom sloju ostvaruju se udaljene prijave korisnika, udaljeno izvođenje naredbi, prosljeđivanje TCP/IP i X11 veza i povezivanje svih veza u jedan kriptirani kanal. Riječ je o najvišem sloju SSH arhitekture na kojem se sva komunikacija odvija putem kanala. Ti kanali se na nižim slojevima prenose preko jedne jedine veze, no u spojnom protokolu virtualno se raspolaže proizvoljnim brojem kanala koje se međusobno razlikuju pomoću identifikatora. Također, s ovim je

slojem korisnik u najizravnijem i direktnom dodiru. Naredbe koje zadaje SSH programima (preko GUI sučelja ili naredbenog retka) prvo obrađuje spojni sloj.



Slika 9. Funkcije slojeva SSH arhitekture

3.2. Dodatne mogućnosti SSH protokola

Osim spajanja na udaljeni poslužitelj i udaljenog izvođenja naredbi, SSH protokol se može koristiti i za sigurnosno poboljšane usluge mrežne komunikacije. Među rješenjima su najčešće sigurnosno poboljšane inačice pojedinih mrežnih protokola i usluga kao na primjer:

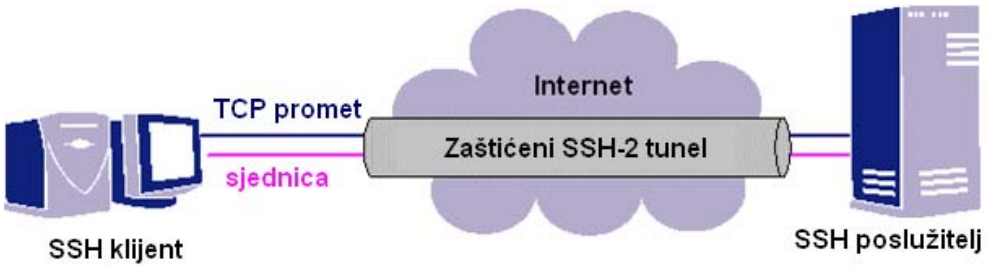
- scp – SSH inačica rcp naredbe koja kopira datoteke s lokalnog računala na udaljeno. Pritom se podaci šalju kriptirani i zaštićeni.
- sftp – SSH inačica FTP (eng. File Transfer Protocol) protokola kojim se datoteke prenose između računala,
- sshfs (eng. SSH Filesystem) – protokol za siguran rad s datotečnim sustavom udaljenog računala.

Primjerice sljedećom naredbom:

```
korisnik@posluzitelj1 ~ $ scp korisnik@posluzitelj2:/putanja_do_datoteke/datoteka2 datoteka1
```

kopira se datoteka *datoteka2* s računala *posluzitelj2* u datoteku *datoteka1* na računalu *posluzitelj1*. Podaci se pritom prenose u kriptiranom obliku te je sadržaj datoteke zaštićen od mogućih uljeza koji prisluškuju promet. Osim toga, onemogućeno je lažno predstavljanje zlonamjernog poslužitelja.

SSH protokol omogućuje i prosljeđivanje TCP/IP prometa tzv. SSH tuneliranjem.



Slika 10. SSH tuneliranje
Izvor: SSH Communications Security

Cilj ove metode je neki inače nezaštićeni promet (POP, HTTP, TELNET) preusmjeriti preko zaštićene SSH veze. Postupak tuneliranja moguće je izvesti sljedećom naredbom:

```
korisnik@posluzitelj~ ssh –L port:host:hostport
```

Pritom se priključak *port* na lokalnom računalu (klijentu) spaja s *hostport* priključkom na udaljenom poslužitelju (host). Prvo se stvori priključnica (eng. socket) koja sluša na zadanom lokalnom priključku i svaki put kad se uspostavi veza sa zadanim priključkom ona se prosljeđuje preko sigurnog SSH kanala. Odmah se uspostavlja i veza sa zadanim priključkom na udaljenom računalu.

Recimo da se želi preuzeti pošta s poslužitelja elektroničke pošte pomoću POP protokola. Spojit će se neki proizvoljan slobodan priključak, npr. 8888, na lokalnom računalu na definirani POP priključak (110). Prije izvođenja naredbe potrebno je provjeriti je li odabrani lokalni priključak slobodan, a nakon izvođenja naredbe preuzimaju se podaci s definiranog lokalnog priključka. Naredba:

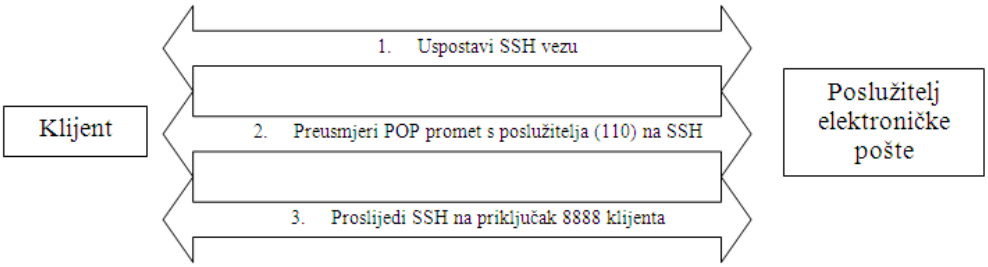
```
korisnik@posluzitelj1 ~ ssh –L 8888:mail_posluzitelj:110
```

stvara zaštićenu vezu preko koje će se preusmjeriti sav POP promet do odredišnog klijentskog računala.

Nakon toga podaci se mogu preuzeti sljedećom naredbom:

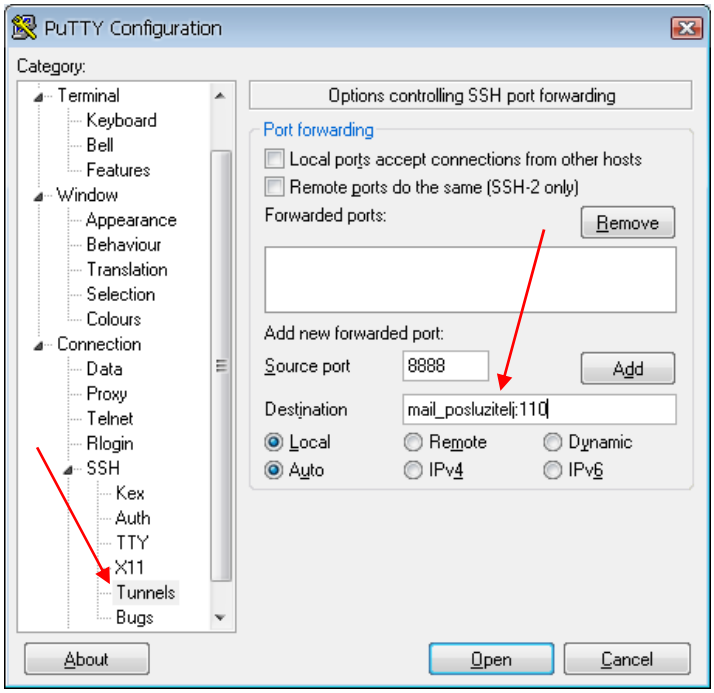
```
korisnik@posluzitelj1 ~ nc localhost 8888
```

Podaci su sa POP priključka 110 proslijeđeni na klijentski priključak 8888 te se odatle preuzimaju pomoću nc naredbe.



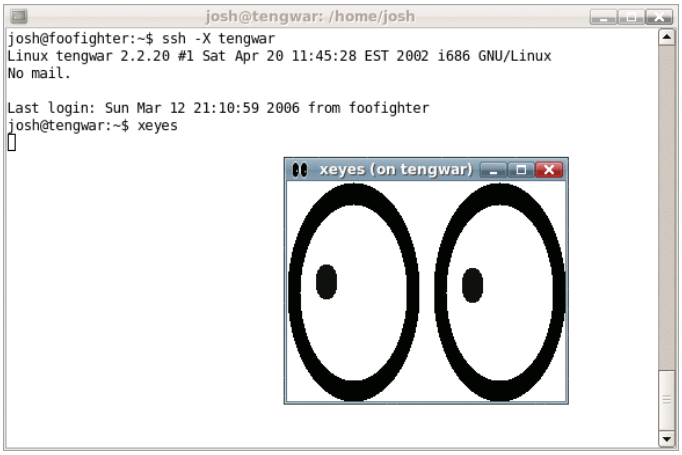
Slika 11. Primjer SSH tuneliranja

Definiranje tunela na SSH klijentu za operacijske sustave Windows (Putty) izvodi se upisom vrijednosti u odgovarajuća polja.



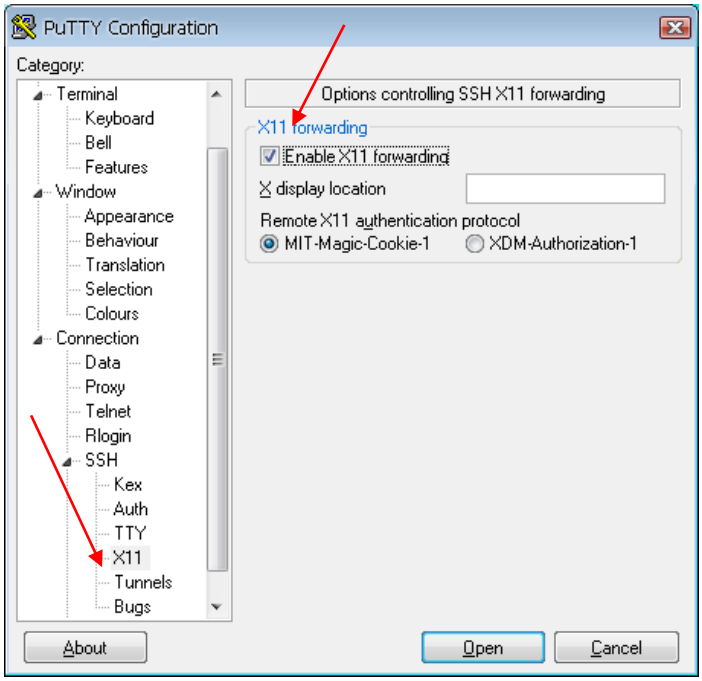
Slika 12. Dodavanje SSH tunela u Putty alatu

Također, SSH omogućuje prosljeđivanje X11 sjednica, odnosno lokalno prikazivanje grafičkog sučelja prema programu koji je pokrenut na udaljenom računalu. Na sljedećoj slici dan je primjer uspostavljanja SSH veze s omogućenim prosljeđivanjem X11 sjednica („-X" atribut). U takvoj sjednici putem terminala pokrenut je program *xeyes* (animacija očiju koji prate pokrete miša) na udaljenom računalu. Kroz SSH tunel podaci su sigurno proslijeđeni do lokalnog računala gdje je prikazano i grafičko sučelje prema programu.



Slika 13. X11 prosljeđivanje putem terminala
Izvor: Wikipedia

Preko SSH klijenta za operacijske sustave Windows omogućavanje X11 sjednica izvodi se postavljanjem kvačice u odgovarajuće polje SSH klijenta kako je prikazano na slici koja slijedi (kao primjer korišten je programski alat Putty).



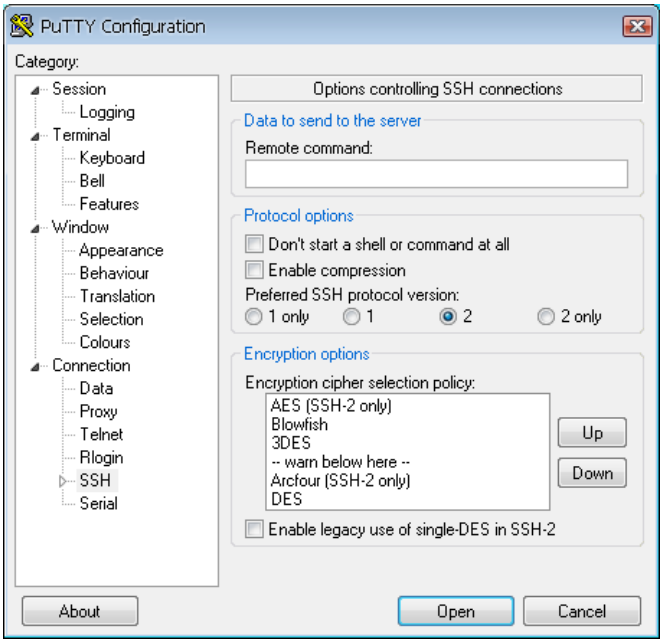
Slika 14. Omogućavanje X11 prosljeđivanja u Putty alatu

OpenSSH paket podržava i uspostavu VPN veze preko SSH protokola što će biti opisano u pripadnom poglavlju o alatu OpenSSH budući da zasad jedino on podržava ovu mogućnost.

4. Programska ostvarenja SSH protokola

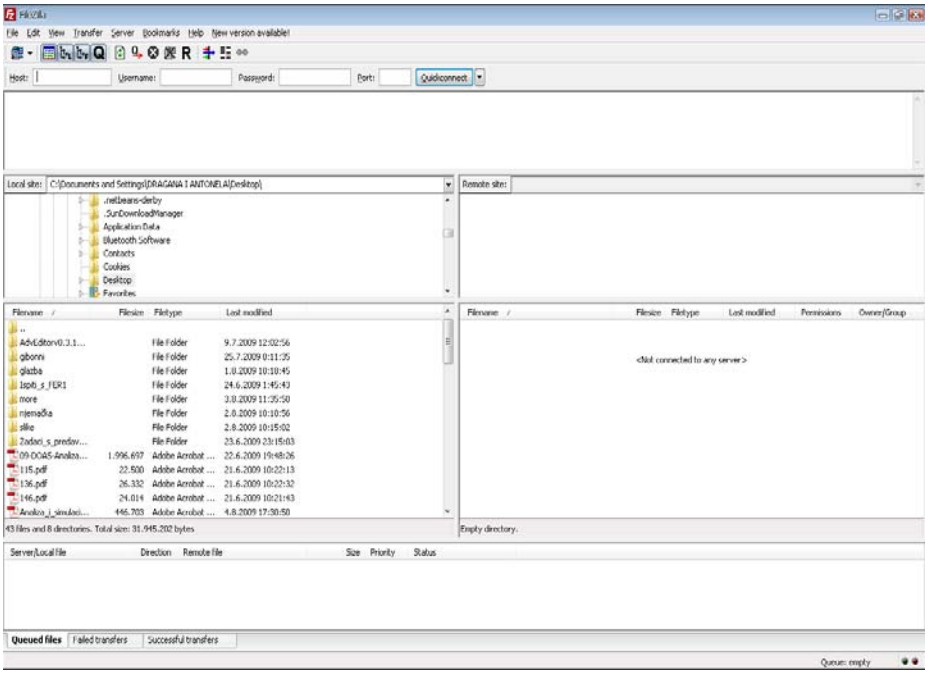
Od kad je osnovana, tvrtka SSH Communications Security bavi se razvojem komercijalnih inačica SSH programa. No, zbog dostupnosti besplatnih SSH programa, komercijalne inačice nemaju prevladavajuću ulogu na tržištu. Najrašireniju upotrebu ima paket OpenSSH koji dolazi u paketu s operacijskim sustavom OpenBSD od njegove inačice 2.6, a dostupne su i njegove inačice za brojne druge operacijske sustave. Osim paketa OpenSSH, postoje i druge besplatne izvedbe SSH protokola među kojima se kao značajnije mogu spomenuti:

- **MacSSH** – programsko ostvarenje SSH-2 protokola za operacijske sustave Mac OS. Osim toga dostupna su i Fugu grafička sučelja prema SCP i SFTP protokolima za operacijske sustave Mac OS X.
- **Dropbear** – klijent i poslužitelj za različite UNIX/Linux operacijske sustave (samo SSH-2).
- **OSSH** – zastarjeli alat koji podržava samo SSH-1 protokol. Na temelju pripadnog programskog koda izrađena je OpenSSH izvedba.
- **LSH** – klijent i poslužitelj za UNIX/Linux operacijske sustave koji podržava samo SSH-2 protokol.
- **Putty** – klijentski program za operacijski sustav Windows koji podržava SSH-1 i SSH-2 protokole



Slika 15. Grafičko sučelje klijenta Putty

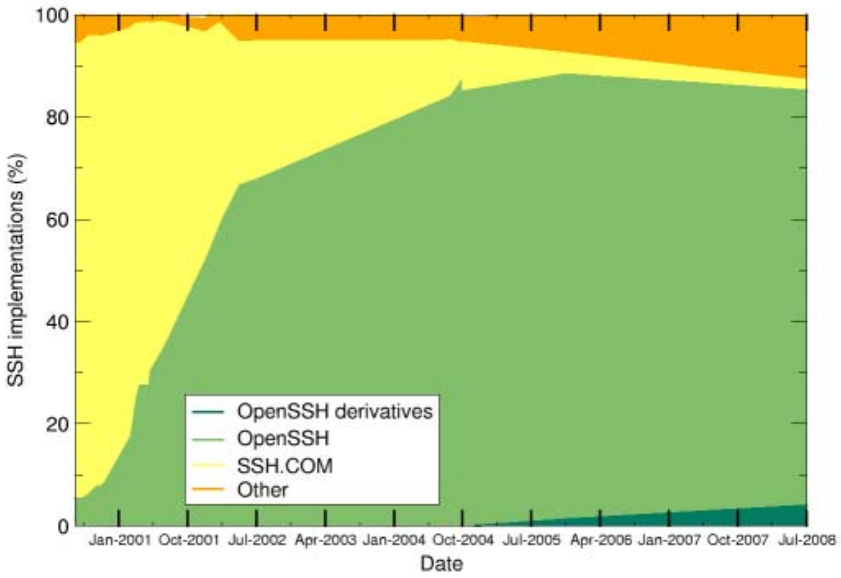
- **TeraTerm, WinSCP,PenguiNet** – SSH klijenti za operacijske sustave Windows
- **Cygwin** – OpenSSH može se ograničeno koristiti na operacijskim sustavima Windows preko alata Cygwin.
- **FileZilla** - FTP klijent za Windows OS koji podržava SFTP protokol.



Slika 16. Grafičko sučelje FileZilla klijenta

4.1. OpenSSH

OpenSSH je najpopularnija besplatno dostupna programska izvedba SSH protokola, uključujući obje njegove inačice. Budući da pruža najširi raspon SSH usluga od svih besplatnih alata, u ovom će se poglavlju dati uvid u njegove mogućnosti.



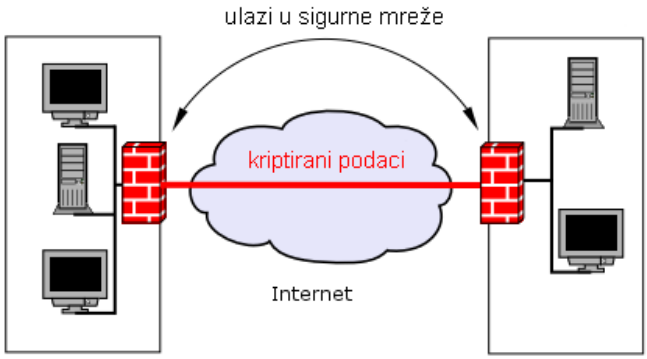
Slika 17. Udio SSH programskih izvedbi u uporabi
Izvor: OpenSSH

Sljedeće naredbe dostupne su u paketu OpenSSH:

- ssh – osnovni klijentski program za udaljeno pristupanje računalima i pokretanje naredbi (sigurna inačica naredbi rlogin i rsh)
- sshd – pozadinski poslužiteljski ssh program
- ssh_config – klijentska konfiguracijska datoteka
- sshd_config – poslužiteljska konfiguracijska datoteka
- ssh-agent – alat koji omogućuje jednostavniju autentifikaciju klijenta tako što pohranjuje i čuva privatne ključeve
- ssh-add – alat koji dodaje nove RSA ili DSA objekte u ssh-agent
- sftp – ssh izvedba ftp protokola
- scp – ssh izvedba rcp naredbe
- ssh-keygen – alat za stvaranje i rad s autentifikacijskim ključevima
- sftp-server - SFTP poslužiteljski podsustav
- ssh-keyscan – alat za skupljanje javnih ključeva većeg broja poslužitelja
- ssh-keysign – pomoćni program za autentifikaciju temeljenu na poslužiteljima

4.1.1. VPN preko SSH

Paket OpenSSH također omogućuje ostvarivanje VPN (eng. Virtual Private Network) veza baziranim na SSH protokolu. Virtualne privatne veze omogućuju spajanje udaljenih zaštićenih mreža u jedinstvenu virtualnu zaštićenu mrežu preko nesigurne računalne mreže kao što je Internet. Zaštita se ostvaruje uspostavom zaštićenog kanala između dvije mreže.

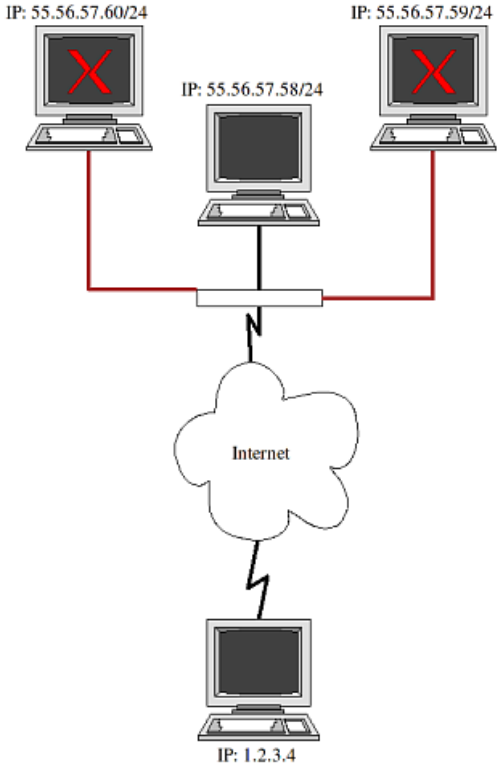


Slika 18. VPN shema
Izvor: Belnet

Virtualne mreže ostvaruju se pomoću virtualnih mrežnih uređaja. Radi se o TUN/TAP (eng. network TAP i TUNel) uređajima koji programski simuliraju rad mrežnih uređaja: usmjeritelja (TUN) i mrežnih mostova (TAP). Pokretanjem *ssh* naredbe s *„–w“* atributom uspostavlja se SSH sjednica sa sučeljem tun0 na oba kraja.

```
ssh –w 0:0 55.56.57.58
```

Pritom se za primjer koristi sljedeći model mreže:



Slika 19. Primjer mreža koje se povezuju VPN-om preko Interneta
Izvor: Perturb.org

Nakon toga potrebno je postaviti IP adrese na oba kraja veze. Uvjet uspješnog izvođenja ove naredbe je uključena mogućnost „PermitTunnel" u konfiguraciji sshd programa (sshd_config datoteka). Na lokalnom računalu izvodi se naredba

```
ifconfig tun0 10.0.2.1 netmask 255.255.255.252
```

a na udaljenom

```
ifconfig tun0 10.0.2.2 netmask 255.255.255.252
```

.

Na ovaj način stvoreno je sučelje za preusmjeravanje IP prometa. Sada je moguće uspostaviti vezu između ove dvije virtualne adrese preko virtualne mreže. Preusmjeravanje IP prometa moguće je izvesti pomoću naredbe *route* koja mijenja usmjerivačke tablice u jezgri računala s ciljem uspostavljanja stalnih mrežnih putova prema određenim poslužiteljima (atribut „-host") ili mrežama (atribut „-net"). Naredbom

```
route add -host 55.56.57.58 dev eth0
```

sav promet prema zadanom poslužitelju (ulaz u zaštićenu mrežu) preusmjeren je preko njegovog izlaznog mrežnog uređaja. Naredba:

```
route add -net 55.56.57.58/24 dev tun0
```

taj isti izlazni promet prema čitavoj zadanoj podmreži (eng. subnet) preusmjerava kroz virtualni uređaj „tun0", odnosno šalje ga kroz SSH tunel.

Uspostavom sljedećih vrijednosti na ulazu u zaštićenu mrežu, na koji se spajamo SSH VPN vezom,

```
echo 1 > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

omogućuje se obostrana komunikacija između računala u zaštićenoj mreži i udaljenog računala. Ukratko, ovim naredbama omogućuje se maskiranje adresa izlaznih IP paketa iz zaštićene mreže prema sučelju eth0. Na taj način adrese računala u zaštićenoj mreži ostaju nepoznate drugim korisnicima Interneta.

4.1.2. Razlika između uspostavljanja VPN veze i TCP/IP tunela preko SSH kanala

VPN i SSH omogućuju zaštitu i enkripciju mrežnog prometa koji kroz njih prolazi. Ipak postoji razlika između prosljeđivanja TCP prometa kroz SSH tunele i uspostavljanja VPN mreža kroz njih. VPN sav mrežni promet prema zaštićenoj mreži šalje kroz SSH kanal. Prosljeđivanje TCP prometa to čini samo s podacima koji se šalju između dva zadana priključka. SSH prosljeđivanje omogućuje dakle zaštitu specifičnog prometa između bilo koje dvije točke i ne stvara dodatna opterećenja. VPN s druge strane štiti sav promet od stranih korisnika, ali ne štiti ga specifično u zaštićenoj mreži. Ovisno o potrebama može se koristiti jedna ili druga metoda, ili čak obje.

4.2. Sigurnost SSH alata

Budući da je izuzetno teško provjeriti sigurnost programskih proizvoda u svim situacijama, praktički niti jedna njegova inačica ne može se smatrati konačnom i potpuno sigurnom. S vremena na vrijeme one se moraju nadograđivati zbog otkrića novih ranjivosti. Zato je važno, bez obzira na to koji se alat odabere, redovito pratiti njegov daljnji razvoj i primjenjivati novije, poboljšane i sigurnije inačice. Ranjivosti nisu oslobođeni niti alati koji bi trebali osiguravati sigurnost (poput SSH izvedbi).

Primarna funkcija SSH protokola je zaštita podataka u prometu enkripcijom. Osim toga, SSH omogućuje zaštitu od tzv. „man in the middle“ napada korištenjem metoda autentifikacije klijenta i poslužitelja. SSH ne osigurava komunikaciju od napada uskraćivanja usluge (eng. Denial of Service).

Ispravno korištene inačice SSH alata mogu se smatrati dovoljno sigurnima jer su temelje na snažnim kriptografskim metodama koje je danas nemoguće izravnim (eng. brute force) napadima ugroziti. Važno je pritom osigurati tajnost privatnih i tajnih kriptografskih ključeva koji se koriste u ovom protokolu jer se njihovim otkrivanjem potpuno uništava sva zaštita koju SSH nudi.

5. Alternativa i dopuna SSH sigurnosti

Više puta je navedeno kako SSH pruža zaštitu komunikacije u nesigurnoj mreži, pri čemu se oslanja na metode enkripcije i autentifikacije korisnika. No, SSH nije jedini protokol koji nudi ovu vrstu zaštite. Osim njega mogu se koristiti SSL/TLS ili IPsec protokoli. U nastavku poglavlja razmotrit će se sličnosti i razlike ovih metoda.

5.1. SSL/TLS

SSL (eng. Secure Socket Layer) i TLS (eng. Transport Layer Security) su sigurnosni protokoli kojim se omogućuje zaštita komunikacije u Internetu. Riječ je o protokolima koji se uspostavljaju na transportnom sloju mrežne komunikacije. SSH je za razliku od njih aplikacijski protokol. SSL je prethodnik TLS protokola koji je s vremenom stekao status IETF standarda, a opisan je u dokumentu RFC 2246. Kao i SSH, TLS omogućuje zaštitu komunikacije u nesigurnoj mreži, pri čemu se oslanja na enkripciju podataka.

TLS se odvija u tri osnovne faze:

1. dogovaranje algoritama,
2. razmjena ključa i autentifikacija te
3. simetrična enkripcija poruka i provjera autentičnosti i integriteta.

Za rukovanje ključevima i autentifikaciju koriste se X.509 certifikati. Očito je da se TLS koristi vrlo sličnim metodama zaštite kao i SSH protokol.

Za razliku od SSH protokola, TLS ne traži nužno autentifikaciju poslužitelja. Moguće je uspostaviti TLS vezu između potpuno anonimnih korisnika. Nadalje, TLS zahtjeva PKI autentifikaciju X.509 certifikatima. SSH podržava takvu autentifikaciju kao mogućnost, no ona se može obavljati i običnim SSH lozinkama, tj. ključevima. TLS ne podržava kompliciranije usluge poput udaljenog pokretanja programa, tuneliranja drugih protokola i veza i sl. TLS je dostupan u besplatnim OpenTLS, NSS i GnuTLS bibliotekama.

5.2. IPsec

Poput SSH i TLS protokola, i IPsec štiti komunikaciju u Internetu autentifikacijom korisnika i enkripcijom paketa. Za razliku od prethodna dva protokola, IPsec se uvodi na mrežnom sloju Internetske arhitekture,

što znači da se štite IP paketi. To također znači da programi ne moraju voditi računa o sigurnosti podataka koje šalju i primaju jer se sva zaštita odvija na nižim slojevima komunikacije. S druge strane, IPsec se najteže uvodi od sva tri navedena protokola jer zahtijeva podršku u jezgri računala, vatrozidima usmjerivačima i drugim uređajima koji rade na mrežnoj razini.

5.3. Budućnost SSH protokola

Sve tri navedene metode zaštite sigurnosni pružaju iste usluge:

- tajnost podataka,
- integritet poruke i
- autentičnost korisnika.

Razlikuju se po tome što rade na različitim slojevima arhitekture Interneta: IPsec na mrežnom, TLS na transportnom, a SSH na aplikacijskom sloju. To znači da se SSH koristi kao gotov korisnički program, TLS podrška ugrađuje se u programe, a IPsec ne zahtijeva nikakvu podršku u korisničkim programima što ga na neki način čini najprivlačnijom metodom. Ipak, IPsec se najteže uvodi u stvarne mreže jer zahtjeva prilagodbu svih mrežnih uređaja, dok je s druge strane za uvođenje SSH sigurnosti potrebno jedino instalacija programa.

Valja napomenuti kako je sva tri protokola moguće istovremeno koristiti, no to je besmisleno i nerazumno jer se jedna te ista vrsta zaštite primjenjuje više puta (što ne poboljšava bitno samu sigurnost). S obzirom na to da je SSH danas lako dostupna i jednostavna metoda zaštite podataka, može se očekivati njegov daljnji razvoj u budućnosti. Ipak, budući da se teži uvođenju zaštite podataka u Internetu na nižim razinama arhitekture, s vremenom bi IPsec mogao preuzeti dominaciju. A jednom kada se uvede takav mrežni sigurnosni protokol, potreba za korištenjem SSH protokola mogla bi iščeznuti ili se bitno smanjiti.

Kao što je rečeno u uvodu ovog poglavlja, sama enkripcija i autentifikacija nisu jedina i dovoljna zaštita podataka u Internetu. Njima se ostvaruje sigurna razmjena podataka, a pretpostavlja se pouzdanost krajnjih točaka komunikacije. Pritom nije uključena provjera podataka koji se razmjenjuju, zaštita od virusa i drugih štetnih programa. Također, iako je dokazana autentičnost klijenta i poslužitelja, sustavi na kojima se oni izvode mogu biti kompromitirani. U svakom slučaju, enkripcija i autentifikacija samo su jedan korak prema sigurnom korištenju mreže. Također, valja imati na umu da najviše štete sigurnosti nanose sami korisnici neopreznim ponašanjem, otkrivanjem lozinki, preuzimanjem datoteka s nepouzdanih izvora te nekorištenjem i ignoriranjem preporučene zaštite. Zato je važno odgovorno se ponašati i koristiti dostupne metode zaštite, uključujući enkripciju osjetljivih podataka pohranjenih na računalu, vatrozide, antivirusne alate, snažne lozinke te redovito nadograđivati postojeće programe novim i sigurnijim inačicama.

6. Zaključak

SSH je protokol aplikacijske razine koji osigurava tajnost i integritet podataka koji se razmjenjuju između dva odredišna računala te autentičnost korisnika. Osmišljen je kao nadogradnja starijih, nesigurnih protokola za razmjenu podataka putem Interneta (TELNET, FTP, rlogin, rsh, rcp). Osim uspostave sigurne veze, protokol se može napredno koristiti za SSH tuneliranje TCP priključaka i X11 sjednica ili za uspostavu VPN mreže preko SSH veze. Danas je dostupno više različitih programskih izvedbi ovog protokola od kojih neke podržavaju samo SSH klijente, dok druge imaju podršku i za SSH poslužitelje. Najčešće korišteno rješenje je besplatni program – OpenSSH, upravo zbog velikog raspona mogućnosti uporabe protokola i činjenice da ga ne treba platiti.

SSH je preporučeno koristiti u situacijama kada je potrebno obaviti sigurnu razmjenu određenih podataka između dva udaljena računala. Važno je redovito nadograđivati korišteni alat kako bi se otklonile ranjivosti koje se periodično otkrivaju. Osim toga, treba imati na umu da SSH ne štiti od DoS (eng. Denial of Service) napada. Općenito, od te je vrste napada komunikacijske protokole vrlo teško zaštiti.

Osim SSH protokola, slične usluge pružaju i protokoli koji se ostvaruju na nižim razinama mrežne arhitekture: TLS (transportni sloj) i IPsec (mrežni sloj). Budući da je mnogo jednostavniji posao uvesti sigurnost na višim nego na nižim slojevima arhitekture, SSH će se i dalje nametati kao privlačno i brzo rješenje.

Također, valja imati na umu da SSH pruža određenu razinu zaštite prilikom korištenja Interneta, no ne i potpunu zaštitu. SSH ne provjerava podatke koji se šalju i pretpostavlja pouzdanost krajnjih točaka komunikacije. Ukoliko se ne koristi primjerena zaštita, poput vatrozida i antivirusnih programa, te krajnje točke mogu biti kompromitirane. Osim programske sigurnosne podrške, izuzetno je važno naglasiti i korisničko ponašanje jer je upravo neodgovornost korisnika najčešći uzrok narušavanja sigurnosti sustava. Osim primjene programske sigurnosne zaštite (SSH, vatrozid, antivirusni alati) potrebno je i dobro zaštiti korisnička imena, lozinke, osjetljive podatke na računalu, obazirati se na automatska sigurnosna upozorenja, ne preuzimati datoteke s nepouzdanih izvora i slično.

7. Reference

1. Secure Shell, http://en.wikipedia.org/wiki/Secure_Shell, kolovoz 2009.
2. OpenSSH, http://www.openssh.com/, kolovoz 2009.
3. SSH Communications Security, http://www.ssh.com/, kolovoz 2009.
4. Daniel J. Barrett, Ph. D., Richard E. Silverman i Robert G. Byrnes, SSH: The Secure Shell *The Definitive Guide, http://www.snailbook.com/, kolovoz 2009.*
5. Nedostaci PKI infrastrukture, www.cert.hr, kolovoz 2009.