

# Short brief of SSH

Vložil/a [il\\_man](#) [1], 6 Listopad, 2005 - 14:25

- [Encryption](#) [2]
- [Security](#) [3]

K čemu SSH je? Mno používá se na připojení ke vzdálenému serveru... kde nám povolí spouštět příkazy (vzdálený terminal), popřípadě kopírovat různé soubory, vlastně jako by jsme byli u daného počítače. Kdykoliv se ssh z nějaký masiny připojí na jinou tak vytvoří šifrovaný kanál tím se stává mnohem výhodnější oproti např. telnetu, rsh apod.

Problém s těmito "starsími" protokoly (telnet atd..) je hlavně ten, že posílají naprosto všechno nezasifrované cíli včetně vašich přihlašovacích údajů potažmo příkazů... pro člověka který si odchytává vámi poslané pakety není problém je číst či hur je ménit, pak místo třeba ls / může poslat rm -r / atd. Díky tomu, že tyto protokoly nemají "ponětí" s kým to hovoří tak je útoku celá rada. Při připojení na neznámý stroj se nám zobrazí něco jako:

```
$ ssh ssh-server.cz
The authenticity of host 'ssh-server.cz' (xxx.xxx.xxx.xxx) can't be established.
RSA key fingerprint is 98:e4:54:d8:f7:c2: .....
Are u sure u want to continue connecting (yes/no) ?
```

mno a teď pokud budeme souhlasit tak se nám klic toho počítače přidá permanentně k našim známým klicům (/etc/ssh/ssh\_known\_hosts) poté pokážete, když se budeme chtít přihlásit znova k tehle masině tak, už se nás to nebude ptát, jestli chceme verit-ulozit klic, ale rovnou nás požádá o zadání jména... pokud jsme paranoidní tak se dá nastavit aby nás klient pokážete kontoloval fingerprinty...

Pokud se na stroj takto přihlásíme, nebo i předtím poprvé, tak je dobré ocheckovat jestli klic-fingerprint, který jsme přijmuly do známých souhlasí s klicem stroje kam se chceme přihlásit, protože se nám mohlo stát, že nám někdo tento klic podvrhnuł ...

Overení můžeme provést několika způsoby, jeden trošku krkolomný, prostě zavolat adminovi, aby nám ho řekl, nebo vhodnější způsob pokud je klic zveřejněn někde na stránkách, tak přes ssl nebo jinou zabezpečenou cestu ho zkontrolovat ... pokud není nějaká jiná cesta "venkem" můžeme udělat jinou věc a to podívat se přímo na stroji kam jsme se přihlásili ten se dá zkontrolovat ve slozce /etc/ssh/ příkazem:

```
ls *.pub | xargs -nl ssh-keygen -lf
```

ten vám vypíše všechny fingerprinty jak rsa tak dsa pro všechny veřejné klíče... pokud chcete vypsat jen jeden klic:

```
ssh-keygen -lf /etc/ssh/ssh_host_typklice_key.pub
```

Tímto jsme odbyli přihlášení k masině, ale co když nás nebaví neustále zadávat přihlašovací údaje nebo nám tato metoda připadá málo bezpečná? Co pak? Tady na scéně přichází metoda zvaná Identity/PublicKey, zadávání jména a hesla je sice zasifrováno ale existují i různé jiné metody jak tyto údaje zjistit třeba keylogger atd.

Co my vlastně můžeme udělat je že, počítači ke kterému se chceme připojit, poskytneme náš veřejný klic a tím mu vlastně dáme pro přísti přihlášení vědet, že jsme to my.

## Short brief of SSH

Publikováno na serveru Security-Portal.cz (<https://security-portal.cz>)

---

Mno a jak to vlastne probiha ? Pocitaci poskytneme nas public (verejny) klic on vezme nejaky data a podle tohoto klice je zasifruje a posle nam je (rika se tomu challenge) nas pocitac je rozsifruje a posle mu je zpátky... zde je jasne, ze jediny kdo to muze rozsifrovat jsme my, protoze jsme jedini kdo ma privatni-soukromy klic... pokud se tento zpusob vlastne nepovede tak se nas pocitac klasicky zepta na jmeno a heslo...

Tak ted z praxe nejdriv jak na to ve widlich :) budete potrebovat prog. jmenem puttygen nebo neco podobneho... mno ten pustite vyberete jaky z klicu chcete generovat a kliknete generate :) je to u nej myslim spojeny s pohybem mysi po obrazovce takze s ni budete muset parkrat pohnout, aby byl ten klic opravdu nahodny ... po vygenerovani se nam vlastne cely ukaze i s fingerprintem... zde si pote muzete urcit jakesi heslo (key passphrase), kterym se to jeste jednou zasifruje .. duvod? kdyby se nekdo dostal na vase konto a chtel by se pak prihlasit pomoci vaseho klice na jiny pocitac, tak by mu nekladl zadny odpor, kdeztu ted bude muset vedet i toto "heslo", kterym se pri generovani klicu sifruje tajny klic ... mno a ted kdyz jste ho zadali, uz si staci jen ulozit jak privatni tak verejny klic.

Jakmile jsme tyto dva klice vygenerovali, tak nas verejny musime prenest na pocitac kam se bude chtit prihlasovat ... preneseme ho do ~/.ssh nebo neco podneho ... mno a ted ho pridame do souboru authorized\_keys, pokud jsme ho pojmenovali treba "key" tak ho tam prekopcime jednoduchym prikazem cat key > ~/.ssh/authorized\_keys ...

Takze ted kdyz se budeme chtit pripojit, tak nejdriv budeme muset nasemu klientovi rict, aby pouzival nas vygenerovany privatni klic, v pripade "meho" putty kliknete na auth/browse a najdete vas privat. klic date ok:) kdyz se budeme nyne pripojovat tak se siti preda akorat informace o tom jaky klic se ma pouzít, na odsifrovani kontrolnich dat a putty se nas zepta na keyphrase pokud jsme tedy nejakou zvolili, aby se nas privat klic mohl pouzít ... cili siti se neprenesly zadne citlive udaje jako heslo a jmeno...

Kdyz budeme tyto klice generovat pod nejakym linuxem tak vam pomuze snad tahle mala tabulka:

```
*****
|SSH protokol | generovani          | jmena                |
|-----|-----|-----|
|1           | ssh-keygen -t rsa1  | identity,identity.pub |
|2           | ssh-keygen -t rsa   | id_rsa,id_rsa.pub     |
|2           | ssh-keygen -t dsa   | id_dsa,id_dsa.pub     |
*****
```

jinak v nastaveni serveru ssh v configu pak mate radky HostKey cesta/ke/klici, pokud se rozhodnete ty klice menit nebo budete pridavat tak je samo musite zmenit/pridat i v tomto konfiguracnim souboru.

Sifra RSA a DSA se pouziva jen pri autentizaci a ne pri samotne praci, protoze je asymetricka (pouziva sifrovaci klice) = pomalejsi. Na prenos dat se pouzivaji jine napr Blowfish, Cast128 atd.. ktere sou symetricke. Muzete si vybrat da se to nastavit ve vesem klientovi teda aspon vetsinou.

## SSH tunneling

O co jde? Asi hodne z vas uz tohle sluvko nekde zaslechlo... mno jde o vytvoreni kanalu/spojeni s nejakym jinym pocitacem, ktere bude sifrovano a muze byt pripojeno na jakykoliv jeho port a vas port... prakticke vyuziti je napr. kdyz mam poskytovatele pripojeni, ktery nedovoluje odesilat postu, cili 25 port je blokovan a co mi tedy muzeme udelat? vytvorit si prave tunel z mail.nekde.cz primo k nam na masinu treba na port 666 :) co se ted vlastne deje? na firewallu u meho poskytovatele to vypada jako bych mel pusteny ssh ale ja si spokojene posilam postu jedine co ted staci udelat je posilat postu na localhost (na 127.0.0.1) na port 666 a muzu posilat posty kolik je mi libo.

Jak na to? jednoduse v linu staci napsat:

## Short brief of SSH

Publikováno na serveru Security-Portal.cz (<https://security-portal.cz>)

---

```
ssh -L 666:mail.nekde.cz:25 mail.nekde.cz
```

ted sem prave vytvoril tunel ze stroje mail.nekde.cz z 25 portu na muj port 666. Lze se pak jednoduse overit jestli to funguje. Staci napsat napr. nestat -ln --tcp a vypiseme si vsechny otevrene porty na localu ... a jeste overit funkcnost staci telnet 127.0.0.1 666 a objevi se na nem uvodni obrazovka na stroji mail.nekde.cz :) Jinak pro windows existuje prog. zvaný stun(n)el a ten vlastne tyto uplne nejjednodussi tunely vytvari, ale moc sem s nim nedelal... jen pro pripojeni na jisty irc server pac muj klient nepodporoval kryptovani...

Dalsi moznosti vyuziti kterou uvitaji hlavne lidi jako ja, kteri jsou za pocitacem s NATem a potrebují aby se nekdo z venci mohl pripojit na jejich komp. Jak to vlastne vypada? no prikaz vypada asi nejak takhle

```
ssh -R 5555:localhost:80 firewall.nekde.cz
```

5555 - na ktery port...

localhost:80 - co a kam budeme tunelovat...

firewall.nekde.cz - kam se budeme hlasit = pocitac s natem...

Ted uz je to mozna trosku jasnejsi prepínac R, znamena remote (vzdaleny) L je local (mistni). Vlastne staci nejakemu cloveku zvenci se pripojit na firewall.nekde.cz na port 5555 a uvidi nase stranky :) takhle se daji mimochodem i spoustet ftpka a hrát hry... vyuziti je urcite hodne... pozor ale u ftp musite nastavit pasiv jinak vam to nepujde, protoze pocitace by komunikovali na dvou portech na jednom by si vymenovali ridici data a na druhem normalni data nebo proste to co chcete prenaset ...

## Vyhody SSH

- Zbezpecena komunikace (end to end)
- Hesla neputuji siti - pri pouziti te pokrocilejsi metody pouziti klicu
- Umi prenaset soubory scp, sftp atd...

## Nevyhody

mno asi oproti nekryptovanym je jasne ze je trosku pomalejsi.. nejde udp cili vsechno spojeni musi byt pres tcp... a to bude asi vse :)

**URL článku:** <https://security-portal.cz/clanky/short-brief-ssh>

### Odkazy:

[1] <https://security-portal.cz/users/ilman>

[2] <https://security-portal.cz/category/tagy/encryption>

[3] <https://security-portal.cz/category/tagy/security>