Open SSH. Praktické použití.

Petr Zemek
xzemek02@stud.fit.vutbr.cz

30. dubna 2009

Abstrakt

Tato práce se zabývá praktickým zabezpečením Open SSH serveru a klienta pomocí konfiguračních souborů a jiných nástrojů. U nastavení jsou uvedeny důvody jeho změny a možná rizika, která mohou hrozit v případě zvolení špatné hodnoty či ponechání hodnoty standardní. Také jsou v textu uvedeny další užitečné nástroje, které dokáží úroveň zabezpečení zlepšit nad rámec standardních možností konfigurace Open SSH. Protože některá nastavení mohou znesnadnit práci s klientskými aplikacemi, tak je ukázáno několik tipů pro usnadnění práce s Open SSH.

Klíčová slova

SSH, Open SSH, šifrovaná komunikace, konfigurace, zabezpečení, usnadnění práce.

1 Úvod

Standardní konfigurace (přednastavená výrobcem) mnohých softwareových produktů bývá zaměřena na univerzální a jednoduché použití, ale nebývá vždy ta nejbezpečnější. V této práci jsou rozebrána ta nejdůležitější nastavení (z pohledu zabezpečení) otevřeného software pro šifrovanou komunikaci – Open SSH[1]. Jejich správným nastavením lze zlepšit zabezpečení serveru a snížit množství útoků. Uvedeny jsou i další užitečné nástroje, které dokáží úroveň zabezpečení zlepšit nad rámec standardních možností konfigurace Open SSH. Nastavením serveru a klienta se zabývá sekce 2. Protože některá nastavení mohou znesnadnit práci s klientskými aplikacemi, tak v sekci 3 je ukázáno několik tipů pro usnadnění práce s Open SSH.

2 Zabezpečení

Zabezpečení se týká jak strany serveru (démon `sshd`), tak strany klienta (programy `ssh`, `scp` a `sftp`). Obě strany mají své konfigurační soubory [7, 6], jejichž umístění se liší systém od systému [16], ale rozdělení podle účelu je všude stejné. V rámci tohoto článku budu uvažovat umístění, které bývá časté v systémech unixového typu:

`/etc/ssh/sshd_config` – globální konfigurace serveru

`/etc/ssh/ssh_config` – globální konfigurace klienta

`$HOME/.ssh/config` – lokální (uživatelská) konfigurace klienta

Globální konfigurace klienta se použije, pokud daný uživatel nepoužívá svou vlastní konfiguraci v domovském adresáři nebo zde nemá některá nastavení specifikované. Pro změny v konfiguračních souborech je třeba mít příslušná práva.

V rámci zajišťování bezpečnosti je kromě samotného nastavení serveru nutná pravidelná kontrola záznamů (logů), a to i tehdy, pokud máte nainstalovány nástroje, které to dělají automaticky. Dále je důležité sledování zjištěných bezpečnostních děr v používané implementaci Open SSH a následná reflexe v podobě instalace potřebných záplat či změny nastavení serveru.

Protože nejslabší článek (z hlediska bezpečnosti) bývá uživatel [12], je vhodné všechny uživatele, kteří se budou na server připojovat, instruovat o základních bezpečnostních pravidlech, především co se týče výběru a uchování hesla pro přístup na server či k soukromému klíči používaném při autentizaci.

[1] http://www.openssh.org/

2.1 Zabezpečení na straně serveru

Po změně nastavení konfiguračního nastavení serveru je třeba restartovat démona `sshd`. Způsob, jak toho docílit, je opět systémově závislý, ale na systémech unixového typu se často používá `/etc/initd/sshd restart`. Všechna nastavení jsou popsána v [7].

2.1.1 Změna portu

Open SSH server standardně běží na portu 22/tcp, který je všem známý a mnohé skripty používané crackery s tím mohou počítat (postupně prochází stroje a zkouší, zda je port 22 otevřený a v případě že ano, zahájí pokus o průnik). Změna portu (například na 1689) se provede následovně:

```
Port 1689
```

Ač je změna portu některými považována za „zabezpečení pomocí obskurnosti“ (security through obscurity) [11], tak tímto opatřením lze docílit snížení pokusů o automatizovaný útok [13]. Toto opatření nepomůže v případě, kdy cracker použije nástroj pro skenování portů.

2.1.2 Používání pouze nové verze protokolu SSH

SSH protokol existuje ve dvou verzích – 1 a 2 [4]. Nová verze je považována za bezpečnější než starší verze [14], která přestala být podporována v roce 2003. Lze nastavit, aby server používal pouze verzi 2 – takto:

```
Protocol 2
```

2.1.3 Zablokování přihlašování pod superuživatelem

Jelikož účet superuživatele (root) bývá prvním cílem crackerů při pokusu o průnik do systému (protože bývá na většině unixových systémů), tak zablokování možnosti se vzdáleně přihlásit pod účtem superuživatele zvyšuje bezpečnost systému, protože útočník musí nejdříve získat heslo někoho jiného a až z tohoto účtu se může pokusit o získání práv superuživatele [9]. Pokud budete chtít spravovat daný server vzdáleně, tak lze získat práva superuživatele pomocí programů jako `su` či `sudo` z účtu běžného uživatele, pod kterým se k serveru připojíte. Zamezení přihlašování pod superuživatelem se provede následovně:

```
PermitRootLogin no
```

2.1.4 Zamezení přihlašování s prázdným heslem

Účty s prázdným heslem by v systému z důvodu bezpečnosti neměly existovat. I přesto se může stát, že se podaří takový účet náhodou vytvořit a může se jednat o bezpečnostní díru, proto je vhodné nepovolit přihlášení s prázdným heslem.

```
PermitEmptyPasswords no
```

2.1.5 Omezení počtu pokusů o přihlášení

Pro omezení útoků hrubou silou je vhodné limitovat maximální počet pokusů o přihlášení v rámci jednoho spojení. Po překročení tohoto limitu dojde k odpojení od serveru, takže skript od crackera se musí znovu připojit k serveru, což zpomalí jeho postup. Pro trvalejší zablokování je nutno použít jiný nástroj, například `fail2ban`[2].

```
MaxAuthTries 3
```

2.1.6 Omezení doby nečinnosti při přihlašování

Standardně nastavená doba může být příliš dlouhá (například 2 minuty), což může útočník zneužít při útoku typu „odmítnutí služby“ (denial of service) [2]. Proto je vhodné nastavit hodnotu nižší, například 30 sekund.

```
LoginGraceTime 30
```

[2] http://www.fail2ban.org/

2.1.7 Omezení maximálního počtu souběžných neautorizovaných spojení

Pro omezení útoků hrubou silou je vhodné omezit maximální množství souběžných spojení, která nejsou autorizovaná. Následující hodnota znamená, že nové spojení (při dvou již otevřených a neautorizovaných spojeních) bude s pravděpodobností 75% odmítnuto a tato pravděpodobnost se lineárně zvyšuje až do hodnoty 100% při počtu 10 spojení. Protože je zde zahrnuta pravděpodobnost, tak tímto způsobem lze narušit časování v automatizovaných skriptech útočníka.

```
MaxStartups 2:75:10
```

2.1.8 Povolení (či zamezení) přihlášení jen určitých uživatelů

I přeto, že v systému na serveru existuje více uživatelů, tak z hlediska bezpečnosti je vhodné, aby se vzdáleně mohli přihlašovat jen ti uživatelé, kteří opravdu musí. Lze nastavit seznam uživatelů (a skupin), kteří mají povoleno vzdálené přihlašování přes SSH.

```
AllowUsers username1 username2

AllowGroups group1 group2
```

Stejně tak lze explicitně nastavit, kteří uživatelé (a skupiny) se přihlásit nesmí (méně užitečné než předchozí nastavení).

```
DenyUsers username3

DenyGroups group3
```

2.1.9 Naslouchat pouze na určitých rozhraních

SSH server standardně naslouchá (a přijímá spojení) na všech lokálních rozhraních (adresách). Je ovšem vhodné povolit opravdu jen ta rozhraní, která jsou potřeba (na serverech bývá několik síťových adaptérů). To se provede nastavením `ListenAddress` na pouze určité adresy. Například následující nastavení znamená, že server má naslouchat pouze na adrese 192.168.1.2.

```
ListenAddress 192.168.1.2
```

2.1.10 Povolit přihlašování pouze z určitých adres

Toto nelze nastavit u SSH serveru, ale na firewallu. Pokud víte, odkud se vaši uživatelé budou přihlašovat, tak lze na firewallu nastavit povolení příchozích spojení pouze z daných adres. Nastavení je závislé na firewallu, ale pro firewall `iptables`[3] to lze provést takto:

```
/sbin/iptables -A INPUT -m state --state NEW -p tcp -s 10.0.0.0/24 --dport 22 -j ACCEPT
```

Toto nastavení povolí přístup na port 22 (zde standardně běží SSH server) pouze ze sítě 10.0.0.0/24.

2.1.11 Zapnutí UsePrivilegeSeparation

Toto nastavení umožní serveru, aby s právy superuživatele spouštěl pouze nezbytný kód a zbytek aby prováděl proces s méně privilegiemi. Proto by toto nastavení mělo být zapnuto.

```
UsePrivilegeSeparation yes
```

2.1.12 Zapnutí StrictModes

Toto nastavení říká serveru, aby před povolením přihlášení zkontroloval oprávnění u jednotlivých souborů a domovského adresáře uživatele, který se přihlašuje. Jelikož uživatelé občas omylem nechávají své soubory a adresáře (například obsah adresáře `$HOME/.ssh/`) čitelný a zapisovatelný pro všechny [7], tak by toto nastavení mělo být zapnuto.

```
StrictModes yes
```

[3] http://www.netfilter.org/

2.1.13 Vyžadování existence reverzního záznamu u klientů

Vyžadováním existence reverzního záznamu (PTR) u klientů může přispět ke zbýšení bezpečnosti, ale může také vzniknout problém, protože někteří poskytovatelé internetového připojení nevytváří pro své klienty reverzní záznamy, takže takoví klienti se nebudou moci připojit (stejně jako někteří crackeři využívající ke svým útokům IP adresy bez existence reverzního záznamu).

```
VerifyReverseMapping yes
```

2.1.14 Používání autentizace pomocí klíčů

Standardní autentizace probíhá pouze na základě uživatelského jména a hesla. Pro zvýšení bezpečnosti lze vynutit používání autentizace pomocí soukromého a veřejného klíče [15]. Uživatel, který se chce přihlásit přes SSH na server musí mít u sebe nejen svůj soukromý klíč, ale musí k němu také znát heslo, takže tímto způsobem lze výrazně omezit možnosti útočníka, protože útok hrubou silou nelze v tomto případě použít. Nevýhodou tohoto postupu je to, že uživatel si před prvním přihlášením musí vygenerovat veřejný a soukromý klíč a umístit veřejný klíč na server.

```
PubkeyAuthentication yes

PasswordAuthentication no
```

Tímto povolíme autentizaci pomocí klíčů a zakážeme použití uživatelského jména a hesla jako autentizační metodu.

```
UsePAM no
```

Jelikož se nepoužívá autentizace pomocí uživatelského jména a hesla, tak není nutné povolovat PAM (Pluggable Authentication Modules).

2.1.15 Vypnout autentizaci založenou na důvěryhodnosti hostitelského počítače

Tyto metody předpokládají, že počítačům na síti lze věřit a povolují autentizaci na základě IP adresy či doménového jména, což je bezpečnostní riziko.

```
HostbasedAuthentication no

IgnoreRhosts yes

RhostsAuthentication no

RhostsRSAAuthentication no

IgnoreUserKnownHosts yes
```

Protože jak `HostbasedAuthentication`, tak `RhostsAuthentication` je vypnuté, tak není žádný důvod nastavovat `IgnoreUserKnownHosts` na `no`.

2.1.16 Místo TCPKeepAlive používat zabezpečenou alternativu

Místo služby TCP protokolu keep-alive [1], která je nezabezpečená (probíhá na TCP vrstvě), lze využít zabezpečenou alternativu, která funguje na aplikační vrstvě a zasílá zprávy na šifrovaném SSH kanálu [7].

```
TCPKeepAlive no

ClientAliveInterval 60

ClientAliveCountMax 3
```

`TCPKeepAlive no` vypne službu TCP protokolu keep-alive. Zbylá dvě nastavení znamenají, že po 60 sekundách (`ClientAliveCountMax`) dojde k poslání zprávy klientovi s očekáváním odpovědi a pokud odpověď nepřijde ani po zaslání třetího požadavku o odpověď (`ClientAliveInterval`), dojde k odpojení klienta.

2.1.17 Vypnout nepotřebné služby

Pokud některou službu nepotřebuji, pak není důvod ji nechávat zapnutou. V případě, že nepotřebujete připojení přes IPv6 a stačí vám IPv4, pak nastavte hodnotu `AddressFamily` na `inet`.

```
AddressFamily inet
```

Pokud nepotřebuji, aby uživatelé měli možnost přesměrovávat TCP provoz, pak tuto možnost lze vypnout[4].

```
AllowTcpForwarding no
```

Obdobně lze vypnout[4] službu přeposílání grafického (X11) provozu.

```
X11Forwarding no
```

2.1.18 Zablokování útočníků v případě útoku

Pomocí programů jako `DenyHosts`[5] či `fail2ban`[6] lze automatizovaně v případě útoků na server zablokovat útočníky.

2.2 Zabezpečení na straně klienta

Množství nastavení, která se týkají zabezpečení a dají se provést na klientovi, je vzhledem k možnostem nastavení serveru výrazně menší. I tak ale existují některá důležitá nastavení. Všechna nastavení jsou popsána v [6]. Postup konfigurace je uveden v sekci 3.1.

2.2.1 Používání přednostně nové verze protokolu SSH

Jak bylo uvedeno v sekci 2.1.2, tak protokol verze 2 je bezpečnější než starší verze, proto by se měla přednostně používat tato nová verze, což zajistíme následujícím nastavením.

```
Protocol 2,1
```

V případě, že server nebude podporovat verzi 2, tak se použije verze 1.

2.2.2 Povolit kontrolu IP adresy serveru a klíče

Z důvodů detekce možného útoku typu DNS spoofing (bez překladu) by klient měl zkontrolovat, zda IP adresy serveru a klíč sedí s údaji uvedenými v souboru `known_hosts` [17, 6].

```
CheckHostIP yes

StrictHostKeyChecking ask
```

2.2.3 Zakázat autentizaci založenou na důvěryhodnosti serveru

Ze stejného důvodu, jako byl uveden v sekci 2.1.15, by autentizace založená na důvěryhodnosti serveru neměla být používána [17].

```
HostbasedAuthentication no

RhostsRSAAuthentication no
```

2.2.4 Zakázat možnost připojení vzdálených počítačů na přesměrované porty

Z důvodů možného zneužití IP adresy klienta by měla být možnost připojení na přesměrované porty klienta vzdálenými počítači zakázana [17].

```
GatewayPorts no
```

[4] Podle [7] toto nastavení nemá vliv na zabezpečení, pokud současně nezrušíme danému uživateli přístup k shellu, protože uživatel si může po přihlášení nainstalovat své vlastní přeposílače (forwardery).

[5] http://denyhosts.sourceforge.net/

[6] http://www.fail2ban.org/

2.2.5 Nepoužívat nepotřebné služby

Pokud některou službu nevyužiju, pak není důvod ji nechávat zapnutou. Pokud nepotřebujete vzdáleně pracovat s grafickými aplikacemi, pak vypněte službu přeposílání grafického (X11) provozu.

```
ForwardX11 no

ForwardX11Trusted no
```

Volbu `ForwardX11Trusted` je doporučeno nastavit na `no`, kvůli potenciálním útokům typu zachycování stisknutých kláves útočníkem [5, 6].

3 Usnadnění práce s Open SSH

Protože některá bezpečnostní opatření snižují uživatelský komfort, existují nastavení a programy pro zjednodušení uživatelské práce s Open SSH.

3.1 Nastavení specifické konfigurace připojení k jednotlivým serverům

Ke každému serveru lze v lokálním konfiguračním souboru (na unixových systémech standardně `$HOME/.ssh/config`) specifikovat konkrétní nastavení, které se uplatní pouze při připojování na daný server [3]. Tato nastavení lze zadat i v shellu či na příkazové řádce, ale mnohem pohodlnější je využít konfigurační soubor. Všechna nastavení jsou popsána v [6].

Sekce týkající se konkrétního serveru je uvozena volbou `Host` se jménem serveru, které pak lze použít pro zkrácené přihlašování. Pokud uvedete `Host *`, budou se další nastavení týkat všech serverů. Následuje příklad nastavení parametrů pro konkrétní server:

```
Host merkur

    HostName merkur.ufg.co.uk
    Port 1493
    User jgrunel
    Protocol 2
    ForwardX11 yes
    Compression yes
```

`HostName` udává kompletní doménové jméno serveru, ke kterému se budete přihlašovat použitím aliasu `merkur`. V případě, že Open SSH server běží na jiném než standardním portu, tak pomocí volby `Port` lze specifikovat číslo portu. Volba `User` udává uživatelské jméno, pod kterým se budete k serveru přihlašovat (bez tohoto nastavení se standardně při připojování bere aktuální uživatelské jméno, pod kterým jste přihlášení v systému, odkud se vzdáleně přihlašujete na server). `Protocol 2` vynutí používání SSH protokolu verze 2. `ForwardX11` povolí zobrazování grafických aplikací přes SSH kanál a `Compression yes` zapíná kompresi při přenosu dat (může zrychlit odezvu při přenosu většího množství dat, například při používání vzdálených grafických aplikací [3]).

3.2 ssh-agent

V případě, že používáme autentizaci pomocí veřejného a soukromého klíče, tak pomocí programu `ssh-agent` lze zařídit, že při prvním připojení a zadání hesla k soukromému klíči dojde k jeho zapamatování a uložení v paměti a při příštích připojeních s tímto soukromým klíčem se použije zapamatovaná hodnota, takže nebude třeba znova zadávat heslo [3]. Popis použití programu je k dispozici v [10] a v manuálových stránkách programu [8].

4 Závěr

Vhodným nastavením Open SSH serveru lze zvýšit bezpečnost serveru a snížit množství útoků vedených na server. Kromě samotného zabezpečení serveru je nutná pravidelná kontrola záznamů a nalezených bezpečnostních chyb v Open SSH. V případě klienta je množství možných bezpečnostních nastavení výrazně menší než v případě serveru, ale i tak existují některá důležitá nastavení. Jelikož některá bezpečnostní opatření snižují uživatelský komfort, existují nastavení a programy pro zjednodušení uživatelské práce s Open SSH.

Literatura

[1] RFC 1122 - Requirements for Internet hosts - communication layers. [online], poslední aktualizace 1989-10-01. [cit. 2009-04-13]. Dostupné na URL: <http://www.faqs.org/rfcs/rfc1122.html>.

[2] Security Focus: OpenSSH LoginGraceTime remote denial of service vulnerability. [online], poslední aktualizace 2006-12-15. [cit. 2009-04-13]. Dostupné na URL: <http://www.securityfocus.com/bid/14963>.

[3] Allen, D. R.: Eleven SSH tricks. *Linux Journal*, ročník 2003, č. 112, 2003: str. 5, ISSN 1075-3583.

[4] Barret, D. J.; Silverman, R. E.: *SSH Kompletní průvodce*. Computer Press, 2003, ISBN 80-7226-852-X.

[5] Cahalan, P.: Bad security 201 - Remote X sessions over SSH. [online], poslední aktualizace 2007-07-09. [cit. 2009-04-13]. Dostupné na URL: <http://padraic2112.wordpress.com/2007/07/09/bad-security-201-remote-x-sessions-over-ssh/>.

[6] Friedl, M.: OpenSSH SSH client configuration files. [online], poslední aktualizace 2009-02-22. [cit. 2009-04-13]. Dostupné na URL: <http://www.openbsd.org/cgi-bin/man.cgi?query=ssh_config>.

[7] Friedl, M.: OpenSSH SSH daemon configuration file. [online], poslední aktualizace 2009-02-22. [cit. 2009-04-13]. Dostupné na URL: <http://www.openbsd.org/cgi-bin/man.cgi?query=sshd_config>.

[8] Friedl, M.: ssh-agent - authentication agent. [online], poslední aktualizace 2007-10-09. [cit. 2009-04-13]. Dostupné na URL: <http://www.openbsd.org/cgi-bin/man.cgi?query=ssh-agent&sektion=1>.

[9] Garfinkel, S.; Spafford, G.: *Practical UNIX and Internet security*. O'Reilly, třetí vydání, 1996, ISBN 1-56592-148-8.

[10] Hershberger, M. A.: Using ssh-agent with ssh. [online], poslední aktualizace 2002-05-18. [cit. 2009-04-13]. Dostupné na URL: <http://mah.everybody.org/docs/ssh>.

[11] Hope, C.: SSH server security through obscurity. [online], poslední aktualizace 2008-02-15. [cit. 2009-04-13]. Dostupné na URL: <http://www.electrictoolbox.com/ssh-server-security-through-obscurity/>.

[12] Lumension: Why end-users are your weakest security link. [online], poslední aktualizace 2007-11-26. [cit. 2009-04-13]. Dostupné na URL: <http://www.gss.co.uk/download/documents/lumension_why_end_users_are_your_weakest_security_link_20080310120643.pdf>.

[13] Miessler, D.: Security and obscurity: Does changing your SSH port lower your risk? [online], poslední aktualizace 2008-03-16. [cit. 2009-04-13]. Dostupné na URL: <http://dmiessler.com/blog/security-and-obscurity-does-changing-your-ssh-port-lower-your-risk>.

[14] NERSC: SSH protocol 2 required. [online], poslední aktualizace 2004-05-24. [cit. 2009-04-13]. Dostupné na URL: <http://www.nersc.gov/nusers/help/access/ssh1to2_user.php>.

[15] Suehring, S.: Using key-based authentication over SSH. [online], poslední aktualizace 2004-03-19. [cit. 2009-04-13]. Dostupné na URL: <http://www.linux.com/feature/34958>.

[16] Toxen, B.: *Bezpečnost v Linuxu: Prevence a odvracení napadení systému*. Computer Press, 2003, ISBN 80-7226-716-7.

[17] Traenkenschuh, J.: SSH security primer: Client security. [online], poslední aktualizace 2007-02-16. [cit. 2009-04-13]. Dostupné na URL: <http://www.informit.com/articles/article.aspx?p=696623>.