Jak se přihlašovat na SSH bez zadávání hesla

15. 4. 2010

Pokud používáte Linux nebo jiný unixový systém, pravděpodobně také hojně využíváte služeb protokolu SSH. Ať už přes něj ovládáte konzolu na druhé straně nebo přenášíte soubory, zřejmě vás obtěžuje neustálé zadávání vašeho složitého a dlouhého hesla. Existuje ale jednodušší způsob: heslo prostě nepoužívat.

Doba čtení: **5 minut**

Přibližně před dvěma měsíci jsem psal o tom, jak nahradit zastaralé FTP pomocí SSH (/clanky/jak-nahradit-ftp-pomoci-sftp-a-zamknout-uzivatele/) a protokolů jako SFTP nebo SCP. Podobné téma jsem už několikrát přednášel na různých akcích a z reakcí mnoha posluchačů vyplynul překvapivý fakt – mnoho lidí netuší jak se přihlašovat k serveru pomocí klíčů, ba dokonce ani často nevědí, že je to vůbec možné.

Proč používat klíče

Autentizace obecně slouží k tomu, abychom počítači dokázali, že jsme ten uživatel, za kterého se prohlašujeme. Nejčastějším nástrojem autentizace je heslo. To má ale v praktickém životě několik nevýhod: dá se (často) uhodnout, mělo by být složité, musíte si jej pamatovat a hlavně je možné jej odposlechnout. Kompromitovaný SSH server tak může

například sledovat, co píšete na klávesnici a na který další počítač se přihlašujete. Navíc pokud používáte jedno extra složité heslo na více strojích, má útočník přístup ke všem.

Všechny výše zmíněné nevýhody řeší metoda přihlašování k SSH pomocí veřejného klíče. Princip je velmi jednoduchý: máme soukromý a veřejný klíč, ten veřejný nahrajeme na všechny naše servery a privátní si necháme. Jelikož není možné z veřejného klíče zpětně ten privátní odvodit, servery nemohou tyto klíče využít k přihlášení mezi sebou, ale jen k ověření vaší totožnosti jakožto držitele privátní části.

Přihlašování pak funguje tak, že server vygeneruje náhodná data, která vám pošle. Vy je svým privátním klíčem podepíšete a pošlete zpět. Server pomocí veřejného klíče, který jste mu předtím dodali, ověří pravost podpisu a pustí vás dovnitř. To vše samozřejmě plně automaticky a velmi rychle. Navíc nedochází vůbec k přenosu tajné informace – privátního klíče. Vlastně tak jednoduše serveru prokážete znalost „hesla“, aniž mu ho musíte ukázat. Díky tomu není server schopen klíč jakkoliv zjistit, zkopírovat a zneužít. Ať by byl server modifikován jakkoliv.

Ze stejného důvodu je pak možné jeden veřejný klíč nahrát na libovolný počet serverů nebo jej zveřejnit třeba na webu s informací „Kdo mě chce pustit na svůj server, ať si stáhne můj veřejný klíč.“ Takový postup je přitom naprosto bezpečný a nijak tím neohrožujete své další servery.

Jak na to prakticky

Použití je navíc velmi jednoduché a sestává z úvodního automatického vygenerování párů klíčů a zkopírování veřejného klíče do domovského adresáře na serveru. To je vše. Nejprve tedy spustíme příkaz pro generování klíče:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/petr/.ssh/id_rsa):
```

Program velmi rychle vygeneruje klíče a ptá se, kam mají být uloženy. Doporučuji ponechat standardní cestu. Stačí tedy

stisknout enter.

```
Enter passphrase (empty for no passphrase):
```

Aby nebylo možné klíče jednoduše z vašeho počítače zkopírovat a zneužít, jsou chráněny šifrováním. Pokud tedy někdo soubor získá, bude mu k ničemu. Přestože je možné nechat privátní klíč nezašifrovaný, rozhodně to nedoporučuji.

```
Your identification has been saved in /home/petr/.ssh/id_rsa.
Your public key has been saved in /home/petr/.ssh/id_rsa.pub.
The key fingerprint is:
5b:15:d8:dd:f1:3f:ee:ea:4b:a9:7f:99:f2:46:73:40 petr@masina
```

Tím jsme ukončili tvorbu klíčů. Oba jsou uloženy ve vašem domovském adresáři a v podadresáři `.ssh`. Mají logické názvy `id_rsa` a `id_rsa.pub`. Veřejná část je označena příponou `.pub` a ta jediná smí opustit váš počítač.

Veřejný klíč teď můžete nahrát na server, na který už máte přístup nebo jej poslat administrátorovi. V obou případech je třeba obsah souboru na serveru nahrát do souboru `.ssh/authorized_keys`. To provedete například příkazem:

```
$ cat id_rsa.pub >> .ssh/authorized_keys
```

Samotný `id_rsa.pub` pak už na serveru ležet nemusí. Pokud se vše povedlo, měli byste být příště při pokusu o přihlášení na server dotázáni jen na lokální passphrase k dešifrování privátního klíče a přihlášení by pak mělo proběhnout automaticky.

Jak se zbavit passphrase

Možná si říkáte, že jsme se vlastně toho hlavního nezbavili – jen místo hesla pro server zadáváme jiné heslo pro dešifrování klíče. Toto nové heslo se sice nikam nepřenáší a celý proces je výrazně bezpečnější, z hlediska uživatele je ale stále otravný. I tento problém je ale možné vyřešit. Řešení se nazývá **ssh-agent**.

To je program, který dokáže zjistit od uživatele heslo k dešifrování privátního klíče, tento načíst do paměti a dešifrovat a na požádání jej vydávat lokálnímu SSH klientovi. Jelikož klíč zůstává bezpečně v paměti a nikdy se v nedešifrované podobě nedostane zpět na disk, je toto řešení bezpečné a především pohodlné – program se zeptá jen jednou a pak už vás neobtěžuje. Dokud se neodhlásíte nebo agenta nedonutíte heslo zapomenout, máte od něj pokoj.

Nejprve je třeba agenta zavést do paměti, aby později očekával vaše příkazy. Existuje několik možností, jak to udělat. Pokud váš systém startuje do textového režimu, můžete příkaz vložit do `~/.bashrc`. V případě že startuje rovnou do X serveru, můžete jej zapsat do `~/.xsession` nebo jej vložit mezi spouštěné programy ve vašem oblíbeném správci oken nebo prostředí. Některé distribuce už rovnou SSH agenta zavádějí samy, takže není třeba se o jeho start starat.

Klíče pak do paměti přidáte pomocí příkazu `ssh-add`. Ten automaticky vyhledá ve vašem domovském adresáři privátní klíče, zeptá se na heslo k nim a načte je do paměti. Od této chvíle se můžete připojovat k serverům bez nutnosti zadávat heslo. Pokud budete chtít dešifrované klíče z paměti vyhodit, spusťte jednoduše `ssh-add` s parametrem `-D`.

Dodatek pro administrátory

Pokud chcete uživatelům dovolit přihlašování pomocí klíčů, musíte mít zapnutou příslušnou volbu v konfiguraci. Je pravděpodobné, že už bude zapnutá od instalace (není důvod ji vypínat), ale přesto je dobré v případě problémů ověřit, že je vše v pořádku. Volbu naleznete v souboru `/etc/ssh/sshd_config`:

```
RSAAuthentication yes
PubkeyAuthentication yes
```

Pokud jsou obě volby na `yes`, je tato možnost povolena a vše bude fungovat. Pokud své uživatele chcete donutit používat právě tuto bezpečnou formu autorizace, můžete jim heslo úplně zakázat. To provedete ve stejném konfiguračním souboru změnou následujících voleb:

```
ChallengeResponseAuthentication no
```

```
PasswordAuthentication no
UsePAM no
```

Tímto postupem jednak zajistíte bezpečnější přihlašování uživatelů, ale také zamezíte zlým hochům, aby zkoušeli hádat heslo a „loupat perníček“. V každém případě nezapomeňte po jakékoliv změně restartovat SSH server:

```
# /etc/init.d/ssh reload
```

Další materiál ke studiu

Seriál SSH intimně aneb úvod do paranoii (/serialy/ssh-intimne-aneb-uvod-do-paranoii/).

Článek Jak nahradit FTP pomocí SFTP a zamknout uživatele (/clanky/jak-nahradit-ftp-pomoci-sftp-a-zamknout-uzivatele/)

Článek Hrátky z řádky: používáme ssh (/clanky/hratky-z-radky-pouzivame-ssh/)

Článek Blokujte SSH útoky pomocí DenyHosts (/clanky/blokujte-ssh-utoky-pomoci-denyhosts/)

Článek Cacti: získávání vlastních dat pomocí SSH
Manuálová stránka programu SSH Agent
Manuálová stránka programu ssh-add
