OpenSSH

Ondřej Caletka

1. června 2013

OpenSSH

- Svobodná implementace protokolu SSH od tvůrců OpenBSD.
- Používaná snad ve všech unixových OS.
- Nejde zdaleka jen o „šifrovaný telnet“.
- Standardizováno v IETF.
- Přednáška bude zaměřená na ukázky.

Běžné přihlášení

- Server drží privátní klíče od každého algoritmu.
- Klient při prvním přihlášení potvrdí pravost otisku veřejného klíče. Vazbu adresy a klíče si uloží klient v `~/.ssh/known_hosts`
- Uživatel se představí heslem.
- Spustí se shell.

Jednoduchá vylepšení

- Vylepšete svého klienta v souboru `~/.ssh/config`
- RandomArt obrázek `VisualHostKey yes`
- Udržení spojení `ServerAliveInterval 10`
- Ukončení mrtvého spojení pomocí `Enter ~ .`
- Napovídání jmen z `known_hosts` pomocí `bash-completion` vyžaduje nehashovaná jména serverů: `HashKnownHosts no`

Použití uživatelského klíče

- Vygenerujeme klíč pomocí `ssh-keygen`
- Dobrá praxe vyplnit smysluplný komentář `-C O.Caletka`
- Klíč chráníme silným heslem
- V zájmu maximální kompatibility nepoužíváme ECDSA klíče (Podporováno od OpenSSH 5.7)
- Kopírování na server ručně nebo pomocí `ssh-copy-id`

Sdílené spojení

- Jedním SSHv2 spojením může být multiplexováno více nezávislých relací.
- Realizováno pomocí řídicího soketu. *Master* jej vytvoří a vede síťové spojení, *slave* jej otevře a komunikuje prostřednictvím *mastera*.
- Autentizaci provádí pouze *master*.

Příklad konfigurace sdílení

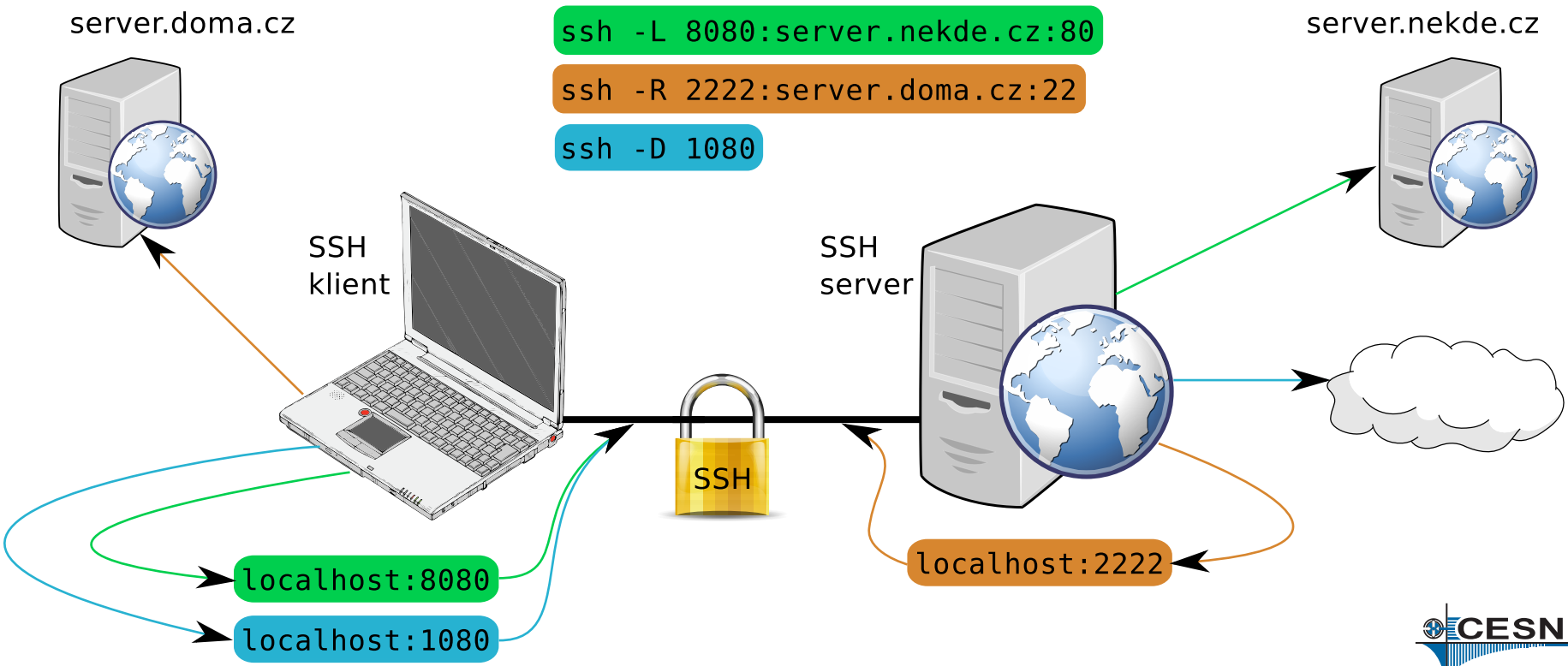
```
ControlMaster auto
ControlPath ~/.ssh/controlsock-%h-%p-%r
ControlPersist  30
```

SSH agent

- Klíčenka s privátními klíči uživatele.
- Nepustí privátní klíč.
- Komunikace unix domain socketem.
- Přidání klíče pomocí `ssh-add -c` vynutí vyžádání potvrzení před každým vystavením podpisu.
- Agenta je možné tunelovat pomocí `ssh -A` nebo volby `ForwardAgent yes`

Tunelování

- Statické i dynamické (SOCKS v4 / SOCKS v5).
- Nejde o TCP-in-TCP.
- Na poslouchací straně omezeno na localhost.



Netcat mode

- Přepínač `-W <host>:<port>` spojí *stdio* klienta k TCP spojení na straně serveru
- Hodí se na tunelování SSH spojení SSH spojením
- Výborně se kombinuje s volbou `ProxyCommand`

~/.ssh/config

```
Host server-behind-firewall
    ProxyCommand ssh -W 10.0.0.1:22 firewall.nekde.cz
```

Ověření serveru pomocí DNSSEC

- Vygenerujeme otisk klíče serveru pro DNS pomocí `ssh-keygen -r <owner>`
- Klientovi nastavíme volbu
  `VerifyHostKeyDNS <yes|ask>`
- Výhoda – klíč je možné bezešvě rolovat

Příklad

```
The authenticity of host 'server.example.com (1.2.3.4)'
can't be established. RSA key fingerprint is
aa:55:cc:9c:a5:c6:1b:f1:a5:d2:be:eb:7e:1c:53:05.
Matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)?
```

OpenSSH certifikáty

- V SSH je možné používat certifikáty jako v PKI
- Certifikační autoritou je libovolný SSH klíč
- Certifikát je možné vystavit jak k serverovému, tak k uživatelskému klíči
- Použití například jako:
  - Osobní CA – pro všechny mé klíče – můžu je průběžně měnit
  - Serverová CA – podepíšu veřejné klíče serverů a nemusím se bát MitM
  - Centrální uživatelská CA – budu vystavovat certifikáty různým uživatelům, nemusím složitě distribuovat `authorized_keys`

Mosh – Mobile Shell

- SSH je přenášeno TCP protokolem
- Na nekvalitní lince je práce nepříjemná
- Mosh používá UDP zprávy a lokální odezvu
- Autentizace pomocí SSH, žádný nový démon
- Usnutí, či změna připojení za běhu
- Ale…
  - neumí tunely
  - neumí IPv6
  - nemá klienta pro mobilní platformy
- Další alternativou je SSH přes Multipath TCP

Závěr

Děkuji za pozornost