OpenVPN a dynamické směrování

Ondřej Caletka

3. března 2013

1 Cíl workshopu

1. Vyzkoušet si instalaci a konfiguraci OpenVPN.
2. Použít dynamické směrování pomocí protokolu OSPF a démona BIRD.

2 Příprava

Pro práci na počítači budeme potřebovat vypnutý firewall, rootovská oprávnění a několik nainstalovaných balíků:

```
# apt-get install openvpn bird
```

Počítače budou fungovat jako *směrovače* (routery), je tedy potřeba zapnout přesílání paketů:

```
# echo 1 >/proc/sys/net/ipv4/ip_forward
```

Také bude zapotřebí hodně terminálových oken, případně terminálový multiplexor (`screen`, `tmux`), protože budeme většinu démonů spouštět na popředí.

2.1 Adresace point-to-point spojů

Pro VPN linky mezi počítači využijeme privátní IPv4 adresy z rozsahu `10.0.0.0/8`. Počítače v učebně očíslujeme čísly 1 – N. Dvoubodouvou linku mezi počítači X a Y, kde $X < Y$ oadresujeme takto:

Počítač X
: 10.X.Y.1/30

Počítač Y
: 10.X.Y.2/30

2.2 Unikátní (loopbacková) adresa počítače

Při stavění dynamicky směrovaných sítí se občas hodí mít k dispozici pevnou adresu pro dané zařízení bez ohledu stav linek, které k zažízení vedou. Použijeme k tomu fiktivní rozhraní dummy0. Také bychom adresu mohli dát přímo na loopbackové rozhraní, ale na samostatném rozhraní se nám k ní nebude plést skutečná loopbacková adresa `127.0.0.1`, kterou protokolem OSPF šířit nechceme.

```
# modprobe dummy
# ip addr add 10.X.X.1 dev dummy0
# ip link set dev dummy0 up
```

3 Směrovací démon BIRD

Směrovací démon BIRD zajistí šíření směrovacích informací mezi jednotlivými routery. Dokáže si téměř vše zjistit sám. Nastavíme pouze id routeru shodně s loopbackovou adresou a vyjmenujeme rozhraní, na kterých má OSPF běžet. Není chybou, když rozhraní v době startu démona neexistuje, BIRD si vzniku všimne i za běhu a okamžitě s ním začne pracovat.

V souboru `bird.conf` je ukázková konfigurace, tu je lepší pro přehlednost vymazat a nahradit touto jednoduchou konfigurací:

```
router id 10.X.X.1;

protocol device {
    scan time 10;
}

protocol kernel {
    export all;
}

protocol ospf {
    area 0.0.0.0 {
        interface "dummy0";
        interface "tap*", "tun*";
    };
}
```

3.1 Interaktivní ovládání BIRDa

Není-li BIRD dosud spuštěn, spustíme ho. K ovládání běžícího BIRDa slouží příkaz `birdc`. V ovládací konzoli můžeme sledovat stav směrovacích procesů. Nejdůležitější příkazy jsou:

`configure`
: Načte znovu konfigurační soubor a aplikuje změny.

`show protocols`
: Vypíše stručný přehled nakonfigurovaných směrovacích protokolů a jejich stavu.

`disable ospf1 / enable ospf1`
: Zakáže nebo povolí určený protokol (změna ale nevydrží přes příkaz `configure`).

`show route`
: Ukáže interní směrovací tabulku, která není totožná se směrovací tabulkou jádra.

`show ospf neighbors`
: Vypíše přehled o OSPF sousedech.

`show ospf state`
: Vypíše detailní přehled stavu OSPF sítě.

4 Konfigurace OpenVPN

Nyní již máme vše připraveno k vytvoření první VPN. OpenVPN nepotřebuje konfigurarční soubor, všechny volby z konfiguračního souboru je možné zadat také přímo na příkazový řádek (s prefixem `--`).

4.1 Konfigurační volby OpenVPN

Nejprve shrňme nejdůležitější volby, které budeme používat. Pro další nahlížejte do manulálové stránky `openvpn(8)`.

`dev tap`
: Typ tunelového rozhraní (tap = virtuální ethernet, tun = L3 rozhraní).

`remote <host> [port] [proto]`
: Spojit se se vzdálenou adresou a portem.

`proto udp`
: Použitý protokol.

`nobind`
: Neposlouchat na pevném čísle portu (režim klienta – navazuje spojení z náhodného čísla portu).

`port 1194`
: Číslo portu, na kterém démon poslouchá i na který se spojuje.

`ifconfig <local IP> <netmask/remote>`
: Nakonfiguruje VPN rozhraní danou adresu (totéž je možné udělat i standardními systémovými nástroji). Druhý parametr je v případě tap rozhraní síťová maska (`255.255.255.252`), v případě tun rozhraní adresa protistrany.

4.2 Nejjednodušší VPN spoj

Nyní již máme vše připraveno k postavení jednoduchého dvoubodového VPN spoje. Zjistíme skutečnou adresu svého souseda zkusíme navázat spojení:

```
# openvpn --dev tap --remote pc-ASDF.sh.nat \
          --ifconfig 10.X.Y.<1|2> 255.255.255.252
```

Pakliže se povede navázat spojení, podívejte se, co se změnilo v BIRDovi a pokud všechno funguje, navažte stejným způsobem VPN s dalším sousedem. Jen u další instance OpenVPN bude potřeba změnit výchozí port (protože ten bude obsazený běžícím procesem). Také můžete zkusit server/klient variantu, kdy jedna strana (server) vypustí volbu `--remote`, zatímco druhá (klient) použije navíc `--nobind`.

Až se spojení naváže, prozkoumejte BIRDa a příkazy ping a traceroute vyzkoušejte, jak až daleko se dostanete. Také můžete použít Wireshark a podívat se, jak vypadá VPN provoz, teď ještě nešifrovaný.

Pokud si chcete vyzkoušet stejnou variantu s rozhraním TUN, neadresujte spojovací síť zvláštními adresami, ale použijte přímo loopbackové adresy strojů. Sdílení adres není chyba, dáme tím operačnímu systému najevo, že tyto dvě adresy jsou propojeny uvedeným dvoubodovým spojem. Pokud bychom použili vyhrazené adresy podobně jako v předchozím případě, narazili bychom na problém s tím, že tato síť by nebyla v OSPF šířena a vzdálenější počítače by k těmto adresám neznaly cestu. Nastavení bude vypadat tedy takto:

```
# openvpn --dev tun --remote pc-ASDF.sh.nat \
          --ifconfig 10.X.X.1 10.Y.Y.1
```

Volby také můžete vložit do konfiguračního souboru. Odstraňte `--` a napište každou na samostatný řádek. Na debianských OS pak stačí takový soubor s příponou `.conf` umístit do `/etc/openvpn` a restartovat službu.

4.3 Šifrováním předsdíleným klíčem

Nejprve vygenerujeme klíč pomocí:

```
# openvpn --genkey --secret /root/vpn.key
```

Tento klíč nějakým způsobem dopravíme protistraně. Obě strany pak ke své konfiguraci přidají `--secret /root/vpn.key`.

4.4 Generování X.509 certifikátů

Balík `easy-rsa` býval dříve součástí OpenVPN, dnes existuje samostatně na adrese `https://github.com/OpenVPN/easy-rsa`. Po stažení a rozbalení můžeme hned začít generovat certifikáty:

```
$ cd easy-rsa/2.0
$ source vars
$ ./clean-all
$ ./build-ca
$ ./build-dh
$ ./build-key-server mujserver
$ ./build-key mujklient
```

V podadresáři `keys` se objeví vygenerované certifikáty (`.crt`), klíče (`.key`) a žádosti o certifikáty (`.csr`).

4.5 Vícebodová VPN s certifikáty

Pro VPN typu point-to-multipoint použijeme adresy z rozsahu `10.X.99.0/24`, kde X je číslo počítače, který plní roli serveru. Jeho konfigurace bude vypadat takto:

```
dev tap
port 1194
server 10.X.99.0 255.255.255.0
dh   dh2048.pem
ca   ca.crt
cert mujserver.crt
key  mujserver.key
```

Klientský konfigurační soubor bude vypadat takto:

```
dev tap
client
remote pc-ASDF.sh.nat 1194
ca   ca.crt
cert mujklient.crt
key  mujklient.key
ns-cert-type server
```

Poslední volba zajistí, že klient naváže TLS spojení pouze s držitelem serverového certifikátu (vygenerovaného pomocí `build-key-server`). Tím je zabráněno tomu, aby kterýkoli jiný klient předstíral identitu serveru svým certifikátem.

Při používání multibodových VPN s dynamickým směrováním je nezbytné používat rozhraní typu tap. S rozhraním typu tun je uvnitř VPN další směrovač, který ale BIRD nedokáže ovládat.