OpenVPN

Ondřej Caletka

O.Caletka@sh.cvut.cz
http://www.pslib.cz/caletka

VPN – Co je to?

- VPN = Virtuální Privátní Síť (Virtual Private Network)
- Vytváří soukromou sít prostřednictvím veřejné, například Internetu
- Obecně se tedy jedná o (šifrovaný *) tunel pro přenos TCP/IP protokolů (L3), nebo Ethernet rámců (L2) přes TCP/IP spojení

*) Obecně není šifrování podmínkou, ale pak se těžko jedná o soukromou síť.

VPN – K čemu je to dobré?

- Bezpečné rozšíření služeb privátní sítě na mobilní klienty mimo síť – notebooky
- Propojení dvou strojů za NATem, aby bylo možné provozovat služby s NATem nekompatibilní (H323, SIP, IAX, FTP, X11, SSH, VNC)
- Šifrovaný kanál pro zvýšení zabezpečení komunikace, např. ve Wi-Fi sítích

Přehled VPN implementací

- IPSEC + L2TP
  - Secured IP + Layer 2 Tunneling Protocol
  - Nativně podporováno Windows
  - Není plně kompatibilní s NAT
  - Pod UN*Xy obtížné rozjet L2TP jak ve funkci serveru, tak ve funkci klienta
- IPSEC
  - Secured IP bez tunelovacího protokolu
  - Lepší podpora v UN*Xech
  - Je možně přemluvit Windows pomocí „Registry hacku“

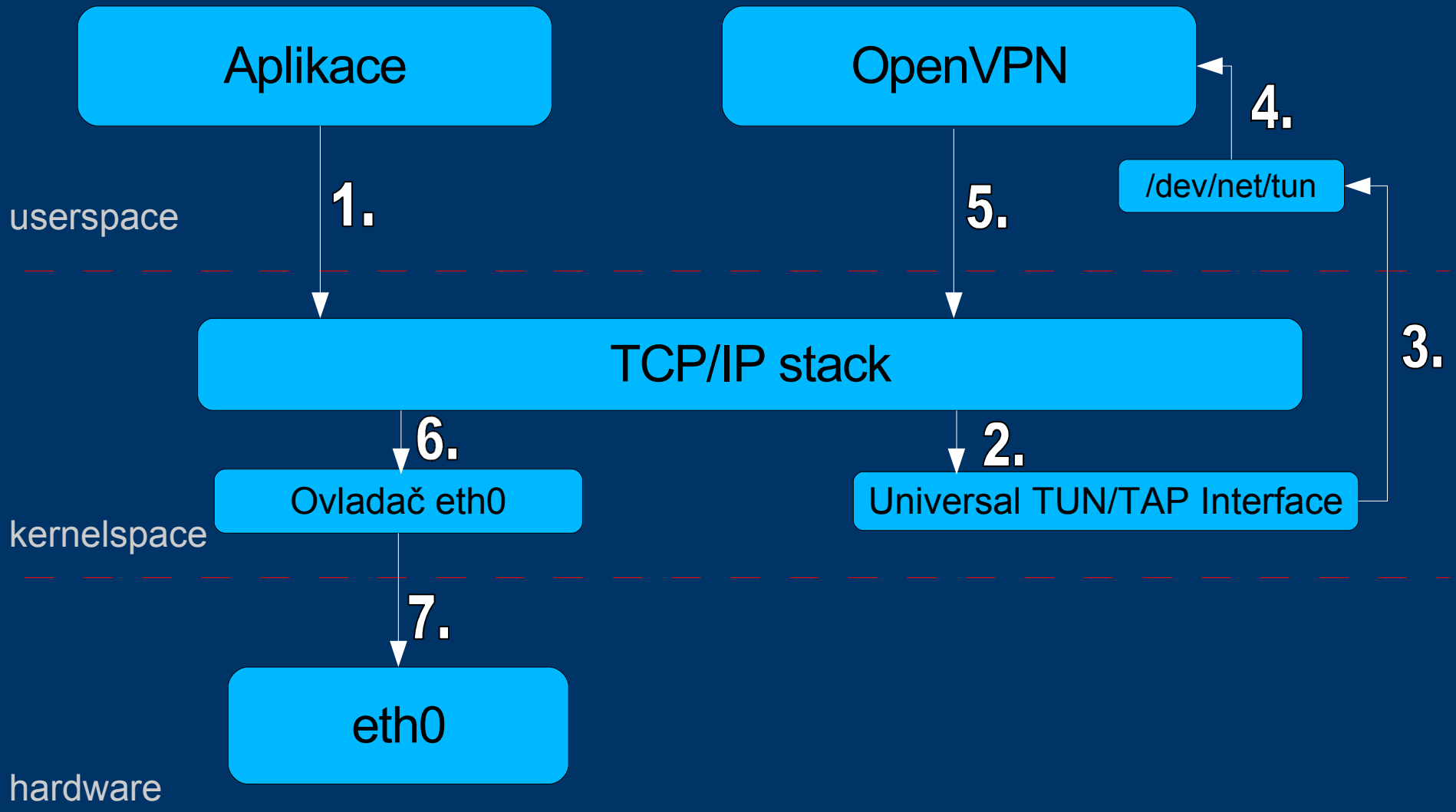
Přehled VPN implementací

- Hamachi
  - proprietální klient, freeware pro Win a Linux
  - víceméně otevřený protokol
  - snadná konfigrace i pro nesíťaře
  - průchod i dvěma NATy (iniciaci asistují servery výrobce)
  - podpora i jiných protokolů, např IPX
- OpenVPN
  - free software GNU GPL
  - spojení přes jedno UDP (nebo i TCP) spojení
  - průchod NATem
  - používá SSL – běží v userspace

OpenVPN – Co je to?

- openvpn.net
- Svobodná implementace VPN
- Tunel je realizován jedním UDP (TCP) spojením
- Kompletní userspace řešení – používá Universal TUN/TAP Interface v kernelu
- Kromě UN*Xů je k dispozici i pro nové Windows

OpenVPN – princip činnosti



OpenVPN – jednoduchá konfigurace

- Typ point-to-point tunelu, šifrovaného předsdíleným klíčem (PSK, Pre-Shared Key)
- Vygenerování sdíleného klíče:

```
openvpn --genkey --secret tajny.key
```

- Konfigurační soubor serveru (L3 verze)

```
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret tajny.key
```

OpenVPN – jednoduchá konfigurace

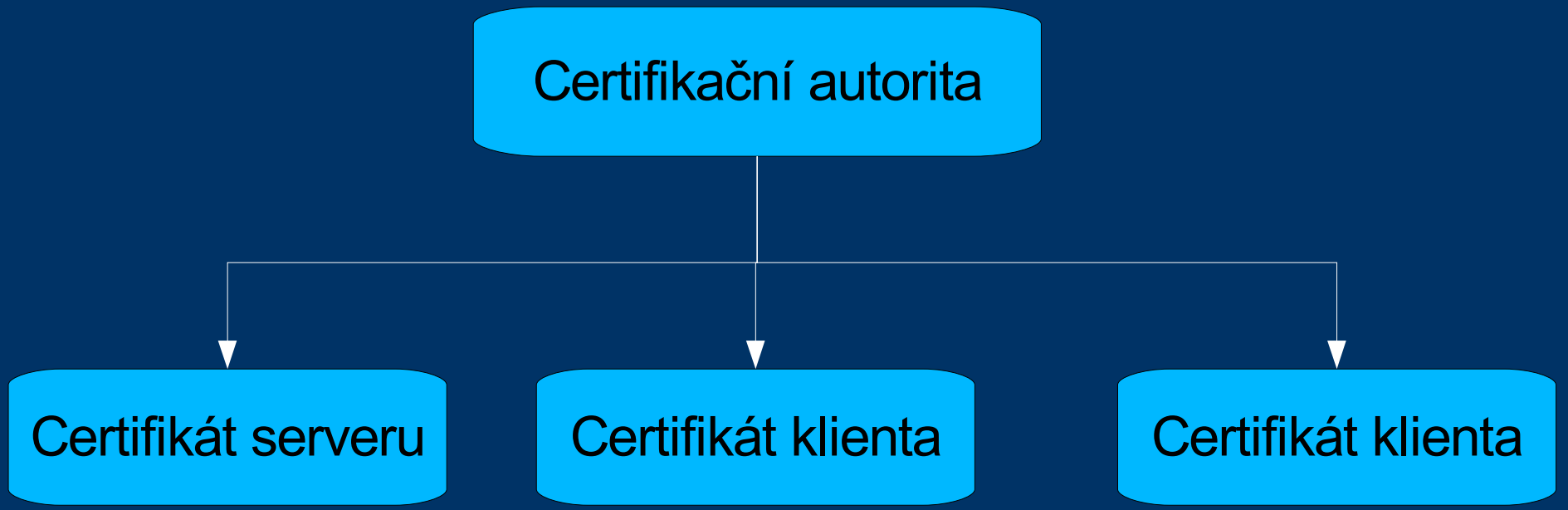
- Konfigurační soubor klienta (L3 verze):

```
remote vzdaleny.stroj.cz
dev tun
ifconfig 10.8.0.2 10.8.0.1
secret tajny.key
```

- Pro L2 verzi se nahradí „`dev tun`“ za „`dev tap`“ a vypustí se „`ifconfig`“
- V této jednoduché verzi se vždy využívá UDP spojení přes port 1194

OpenVPN – PKI varianta

- PKI = Public Key Infrastructure



OpenVPN – PKI varianta

- Každý uživatel OpenVPN obdrží certifikát (veřejný klíč) a klíč (soukromý)
- Při navazování spojení si klient se serverem vymění certifikáty, komunikace je zašifrována certifikátem protistrany
- Certifkát musí být podepsaný certifikační autoritou (CA), která ověřuje, že certifikát patří tomu, za koho se vydává

Vytvoření vlastní PKI

- K tomuto účelu je součástí balíku OpenVPN adresář easy-rsa se skripty pro jednoduché vytvoření vlastní PKI
- Začneme editací souboru vars. Zde nastavíme popisová pole certifikátů. Pokračujeme:

```
$ . ./vars
$ ./clean-all
$ ./build-ca
```

Vytvoření vlastní PKI

- Vytvoření cerifikátu a klíče pro server:
  ```
  $ ./build-key-server jmenoserveru
  ```
- Vytvoření Diffie Hellman pro server:
  ```
  $ ./build-dh
  ```
- Vytvoření cerifikátu a klíče pro klienta:
  ```
  $ ./build-key jmenoklienta
  ```

Vytvoření vlastní PKI

- Uvedený postup vyžaduje přenesení soukromého klíče z místa vytvoření na počítač klienta. Pokud k tomu není k dispozici bezpečný kanál (scp), jedná se o nejrizikovější část postupu. Proto je možné nejprve na klientském systému vytvořit klíč a k němu nepodepsaný certifikát:

```
$ . ./vars
$ ./build-request jmenoklienta
```

  a ten pak odeslat CA k podepsání

Vytvoření vlastní PKI

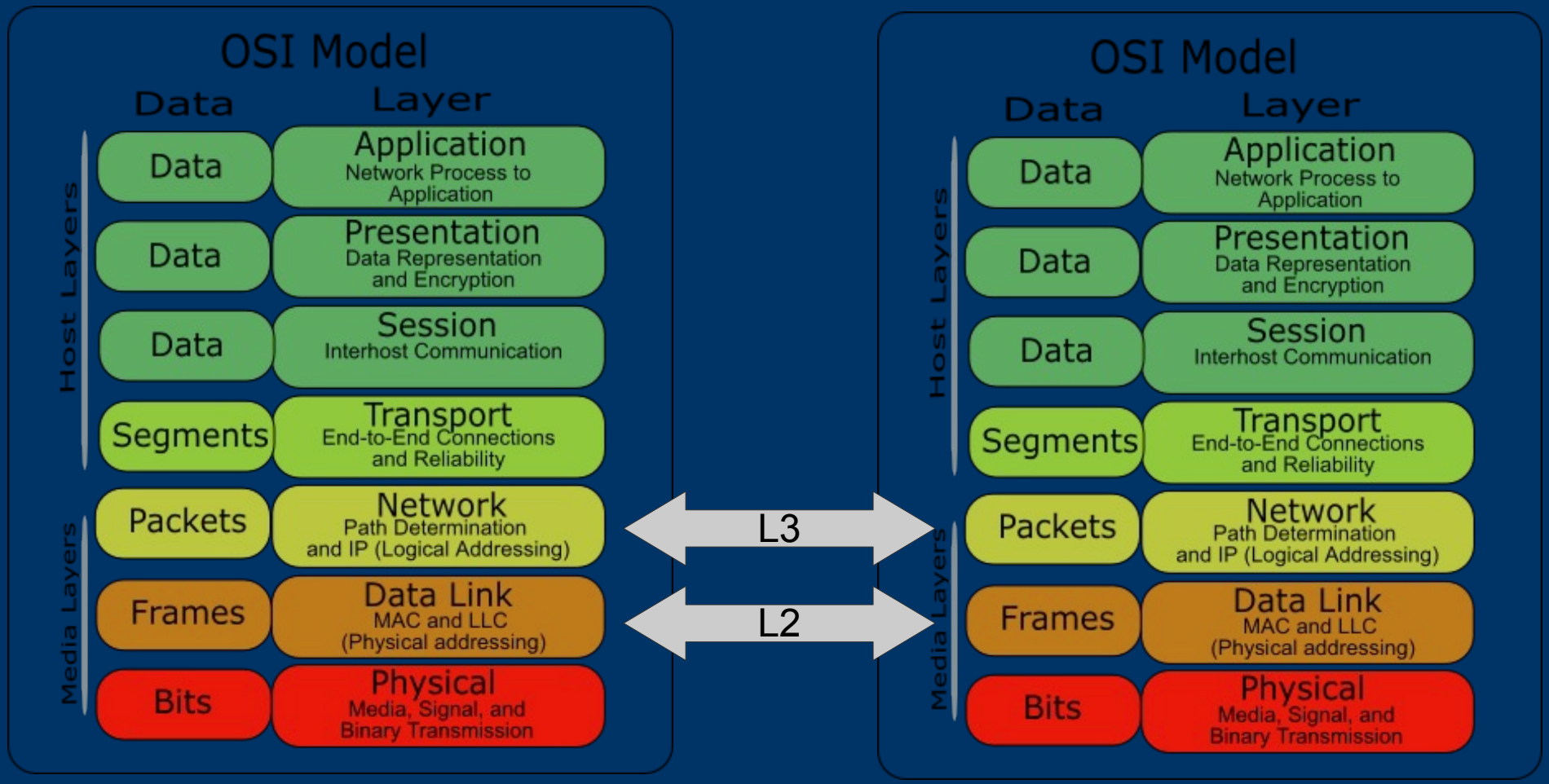
- Certifikační autorita soubor .csr podepíše:

  ```
  $ ./sign-req jmenoklienta
  ```

  a výstupní soubor stačí poslat zpět klientovi.

- Pro ještě vyšší zabezpečení je možné, aby klíč CA sídlil na dedikovaném počítači odpojeném od všech sítí.

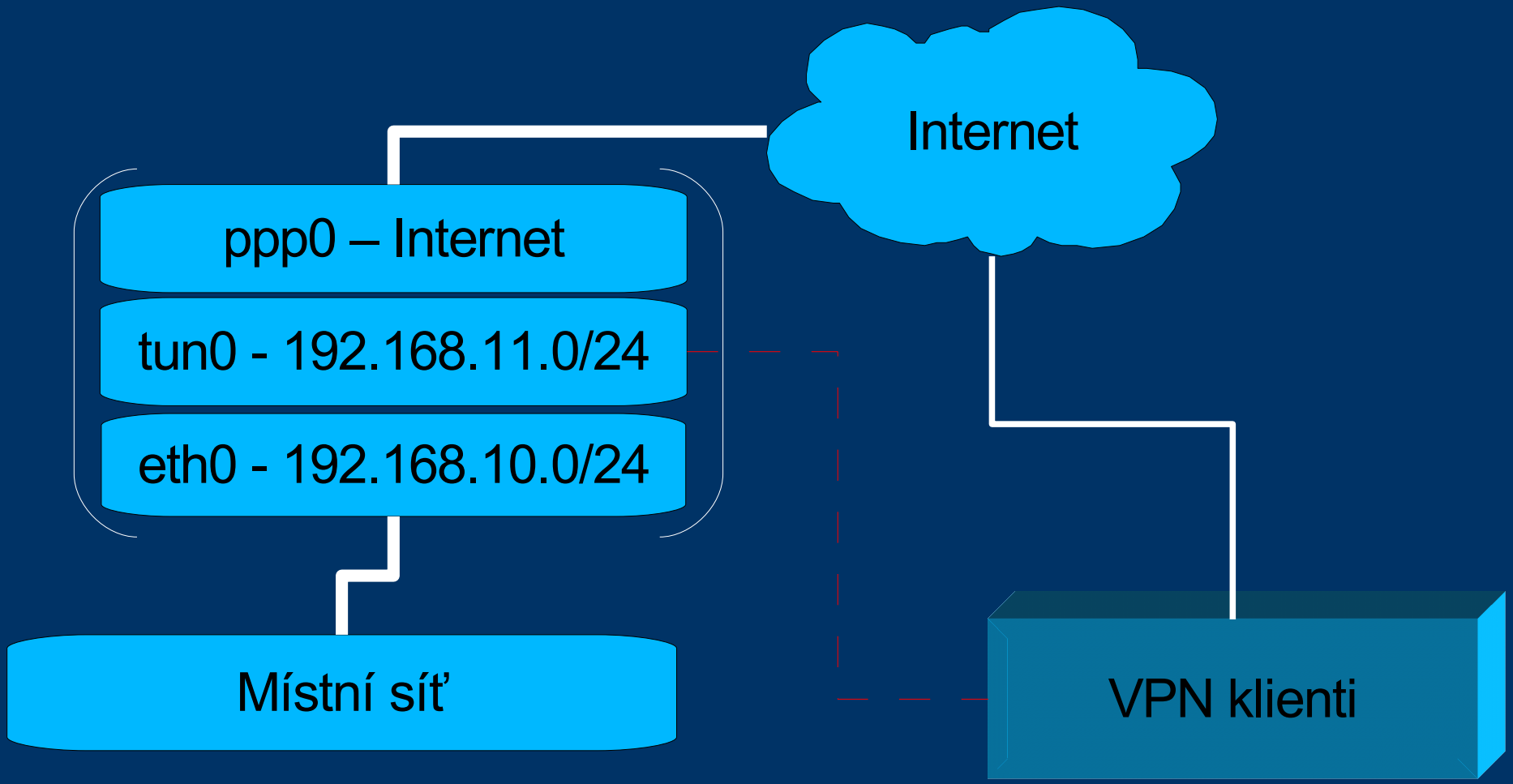
Rozdíl mezi L2 a L3 tunelem



Rozdíl mezi L2 a L3 tunelem

- L3 tunel (routed, tun)
  - PointToPoint i PointToMultipoint TCP/IP spojení
  - Nutnost nastavit routování => problémy pokud VPN koncentrátor není zároveň výchozí branou místní sítě
  - Neprojdou přes něj broadcasty vnitřní sítě
  - Oddělené rozsahy místní sítě a VPN klientů
- L2 tunel (bridged, tap)
  - Virtuální Ethernet adaptér
  - Nutnost vytvořit bridge
  - Adresy místní sítě a VPN klientů se překrývají
  - Projdou i broadcasty a jiné protokoly
  - Problém s DHCP serverem (default gateway)

Modelová situace



Konfigurace serveru

```
port 1194
proto udp
dev tun                 # Režim L3
ca ca.crt               # Certifikát CA
cert server.crt         # Certifikát serveru
key server.key          # Klíč serveru
dh dh1024.pem
server 192.168.11.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push “route 192.168.10.0 255.255.255.0“
client-to-client
```

Konfigurace klienta

```
dev tun                 # Režim L3
ca ca.crt               # Certifikát CA
cert klient.crt         # Certifikát klienta
key klient.key          # Klíč klienta
ns-cert-type server # Ochrana před MitM

client
proto udp
remote vpn.koncentrator.cz 1194
```

Poznámky ke konfiguraci

- Klienti dostanou vždy IP adresy z rozsahu /30. Nicméně adresou serveru pro všechny klienty je první adresa z rozsahu
- Až na výjimečné případy rozhodně není dobrý nápad posílat VPN klientům výchozí bránu.
- MitM útok spočívá ve zneužití klientského certifikátu (který je podepsán CA) pro vytvoření falešného serveru – proto kontrolujeme u serverového certifikátu přitomnost nsCertType server

Konfigurace pro L3 verzi

- Povolení IPv4 forwardování:

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

- Konfigurace FORWARD tabulky:

```
$ iptables -A FORWARD -s 192.168.10.0/24 -d 192.168.11.0/24 -j ACCEPT
$ iptables -A FORWARD -s 192.168.11.0/24 -d 192.168.10.0/24 -j ACCEPT
```

- Protože je server zároveň výchozí branou vnitřní sítě, není problém. Jinak by bylo potřeba na výchozí bráně nastavit:

```
$ ip route add 192.168.11.0/24 via 192.168.10.1
```

Interaktivní povolování firewallu

Pokud jsou klienti jednoho VPN koncentrátoru různě důvěryhodní, nelze se spoléhat na konstantní IP adresu. Místo toho je možné přidávat do firewallu pravidla přes příkaz v konfiguračním souboru:

```
learn-address /usr/local/sbin/fwscript
```

fwscript bude spuštěn jako:

```
fwscript <add|update|delete> <IP adresa> <jméno certifikátu>
```

a vrátí-li nenulovou hodnotu, OpenVPN odmítne klienta připojit.

Podobně se dá vyřešit i interaktivní přidávání záznamů do DNS zóny.

Konfigurace pro L2 verzi

- Vytvoření síťového mostu:

```
$ brctl addbr br0
$ brctl addif br0 tap0
$ brctl addif br0 eth0
```

- Nebo je možno použít skript bridge-start z balíku openvpn
- Po nahození bridge se změní název rozhraní, kterým je PC, které bridgeuje – pozor na konfigraci firewallu.

Konfigurace pro L2 verzi

Server je možno zkonfigurovat dvěma způsoby:

- Buď server bude suplovat dhcp – toho se dosáhne příkazem:

```
server-bridge 192.168.10.1 255.255.255.0 192.168.10.100 192.168.10.150
server-bridge <IP brány>   <maska>       <první IP>     <poslední IP>
```

Tento rozsah musí ležet mimo rozsah lokálního DHCP serveru.

- Nebo lze použít lokální DHCP server pro přidělování adres jak místním, tak VPN klientům (nedoporučeno – totální promíchání adres, přidělení default brány – možným řešením je detekce VPN klienta podle MAC adresy 00:FF:...)

Závěr

- Praktické ukázky:
  - Jednoduchá konfigurace
  - Pokročilá konfigurace
  - Dynamická synchronizace DNS

- Další zdroje informací:
  - man openvpn
  - openvpn.net
  - Google: openvpn
  - root.cz/clanky/openvpn-vpn-jednoduse

VLAK PŘIJÍŽDÍ
DO STANICE

TRAIN
IS APROACHING
THE STATION

No Show