VPN bez muke i previše troškova: OpenVPN

Vlatko Košturjak, IBM
vlatko.kosturjak at hr.ibm.com
kost at linux.hr

Agenda

- What is VPN
- OpenVPN
  - authentication
  - crypto
  - network modes
  - installation and configuration
- Example configurations
- How to sleep tight
- Q&A

What is VPN?

- Virtual Private Networking (VPN)
- Wikipedia:

"*is a communications network tunneled through another network, and dedicated for a specific network. One common application is secure communications through the public Internet, but a VPN need not have explicit security features, such as authentication or content encryption. VPNs, for example, can be used to separate the traffic of different user communities over an underlying network with strong security features.*"

Introducing OpenVPN

- VPN solution
- multiplatform
  - Linux, Windows XP/2000, OpenBSD, FreeBSD, NetBSD, Solaris, Mac OS X, ...
  - OpenVPN 2.1rc7: Windows Vista support
- utilizing OpenSSL
- user mode VPN
  - not using kernel privileges
  - minimal privileges
- free & open source

OpenVPN authentication

- preshared secret key
  - one password to authenticate
  - one password to crack/steal/... :)
  - simple and insecure
- username/password authentication (> 2.0)
  - with or without certificates
  - server still needs certificate
  - supporting PAM and Active Directory
- certificates (PKI)
  - server/client certificates
  - if implemented well – pretty secure

OpenVPN crypto

- encryption of data and control channel
- Based on OpenSSL
- You can utilize almost all cryptos in OpenSSL
  - openvpn --show-ciphers
  - openvpn --show-digests
  - DES, BlowFish, AES (128,192,256), ...
  - hardware accelerators
- Recommendations:
  - Blowfish (128): speed (by default)
  - AES (256): strength

OpenVPN modes

- layer 2
  - Ethernet traffic
  - bridging
    - needs bridging support in kernel
    - needs bridging utilities
  - utilizing TAP driver
- layer 3
  - IP traffic
  - utilizing TUN driver
  - utilize WINS server if you want to handle broadcasts on layer 2

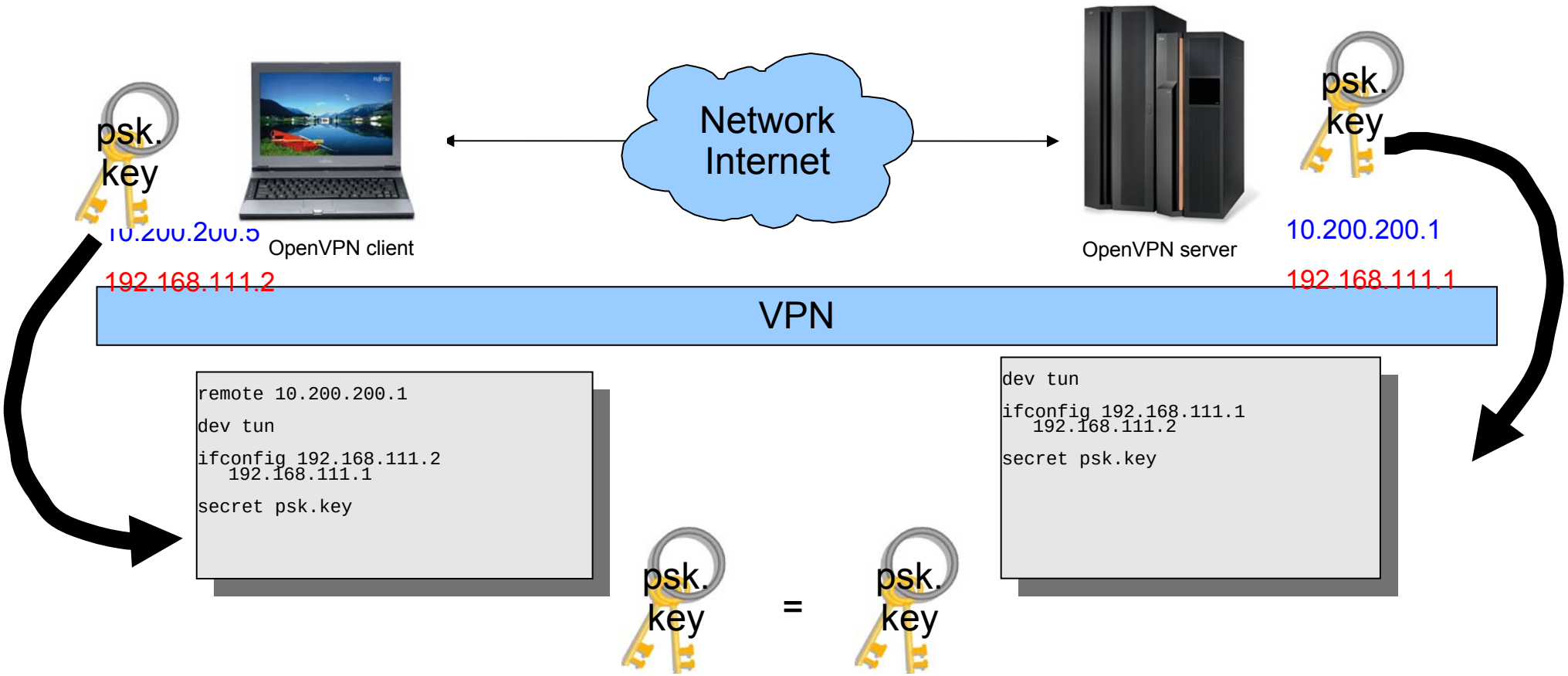
OpenVPN installation and configuration

- Single binary
  - server and client configurations
- Simple text files for configuration (.ovpn)
- There is some GUI frontends, but in early stage of development
- openvpn conf.ovpn

- apt-get install openvpn
- urpmi openvpn
- emerge openvpn
- yum install openvpn
- pkg_add openvpn
- click on that setup.exe :)

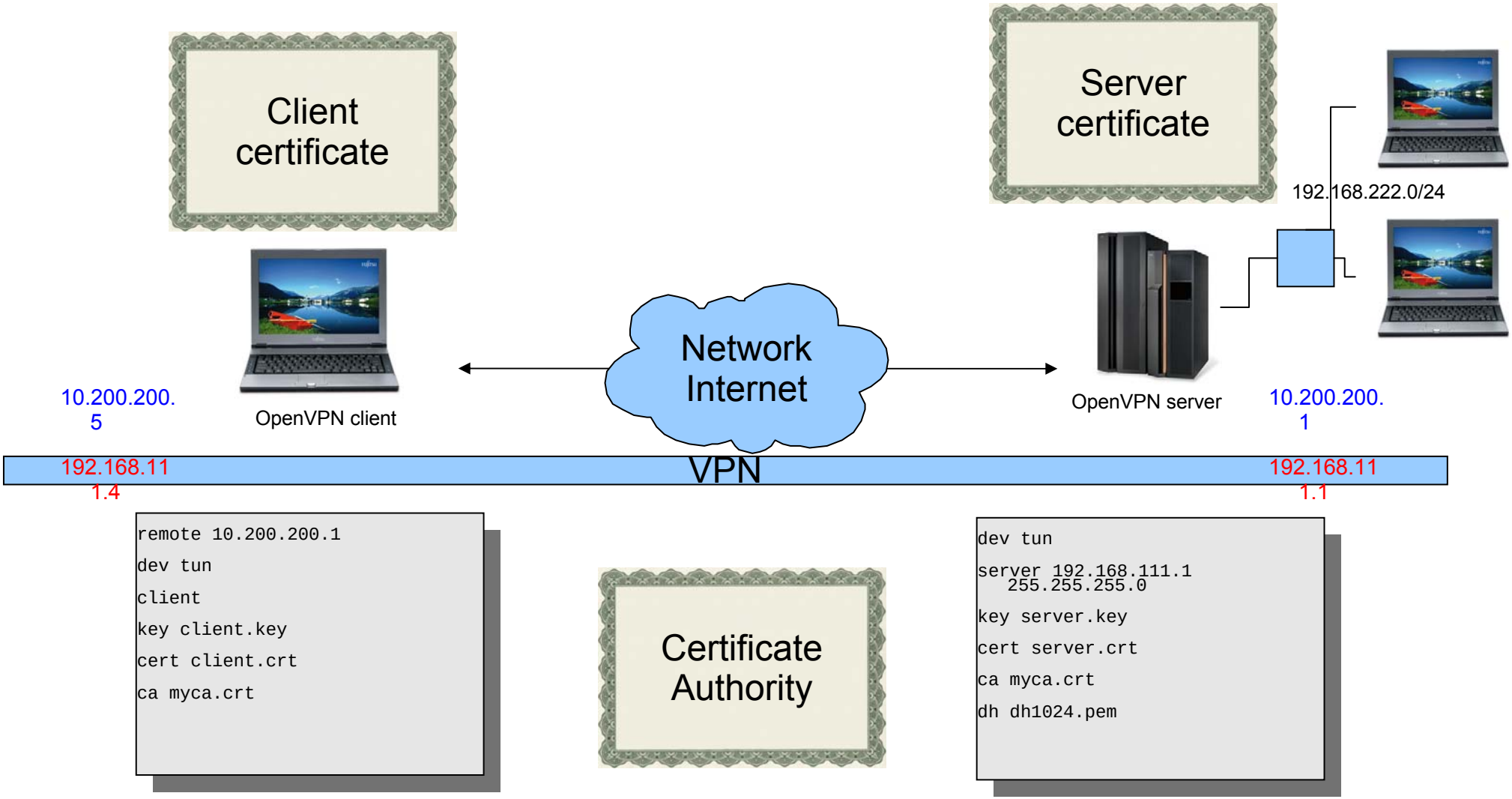
Simple example

- simple point to point example
- using pre shared keys



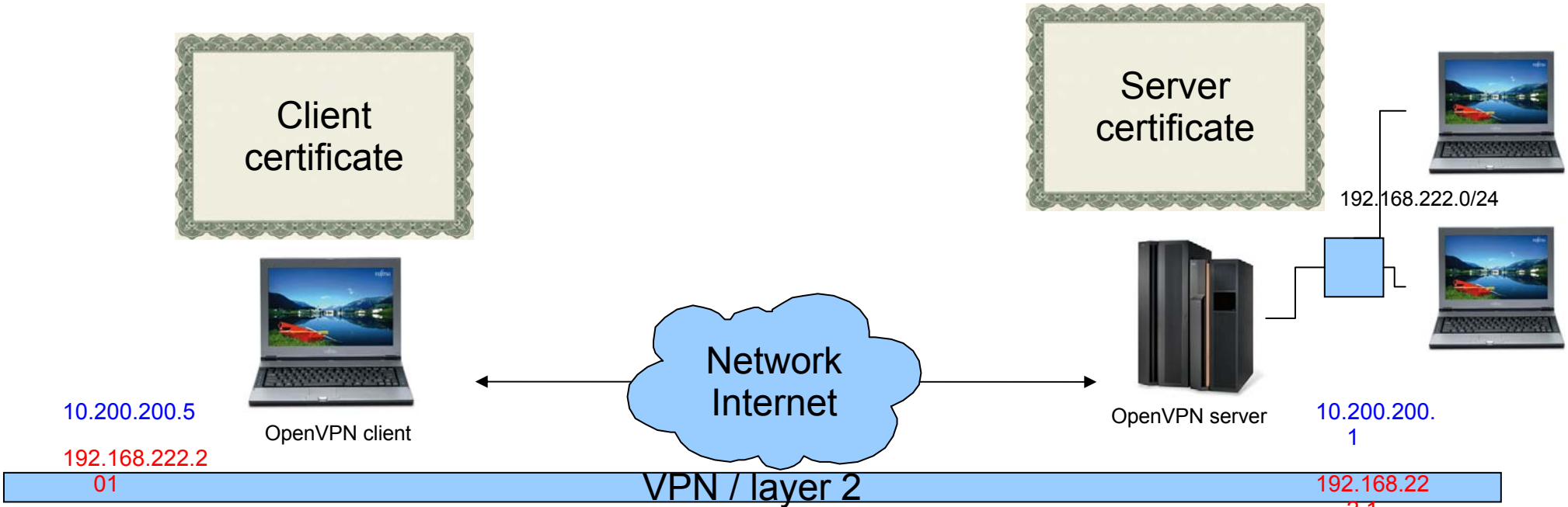
Simple routing example with certs

- utilize NAT or routing to your target network



Simple bridging example with certs

- Not enough broadcast traffic, so bridge it! :)



Other nifty stuff with OpenVPN

- Home routers
  - OpenWRT
  - dd-wrt
- Subnotebook
  - ASUS EEE
- PDA
  - Sharp Zaurus
  - PocketPC
  - ...

VPN security -

- Your VPN is secure as the weakest link in your (in)secure system
  - secure your OpenVPN server
  - secure your network
  - secure your VPN clients
  - secure your VPN keys/passwords
  - educate people
  - ...

  Yeah, go and implement that bloody ISMS!

VPN security & openSSL

- Implement CRL
  - certification revocation list
- Debian OpenSSL
  - Debian
  - Ubuntu
  - other debian based distros
- Are my keys vulnerable?
  - run openvpn-vulnkey

OpenVPN security

- OpenVPN uses mlockall
- Few suggestions:
  - Use Public Key Infrastructure (PKI)
    - utilize passwords on your private keys
    - utilize Certification Revocation Lists (CRL)
    - use ns-cert-type server
    - Use pre-shared secret to even start TLS negotiation for advanced security (if neccessary)
  - Drop your privileges (say hi to nobody)
  - Double check your file permissions
  - Implement chroot jail

OpenVPN disadvantages

- It's NOT standard IKE protocol
  - you cannot talk to CISCO or any other VPN provider
  - maybe even advantage?
    - other VPN probes won't work
      - security through obscurity
    - IKE can be complex
      - untrained personnel can open security holes
      - OpenVPN is simple and stupid (KISS principle of security)

OpenVPN

- free
- open source
- flexible
- easy to use
- multi-platform
- not standard IKE