LAB4: Virtual Private Networks (VPN)

Tomáš Rebok

Brief VPN introduction

What is VPN?

The goal of a **Virtual Private Network (VPN)** is to provide private communications within the public Internet infrastructure

- they employ various networking technologies to achieve the goal
  - can occur at any layer of the OSI protocol stack
  - theoretical background provided by the lecture
- basic VPN idea:
  - build a virtual overlay network that is run on top of the Internet infrastructure
    - *"virtual"* . . . means that there is not a new infrastructure necessary
  - connect private networks by the overlay networks
    - can be built between two end systems, an end system and a network, or among two or more networks

VPNs Basic Functions

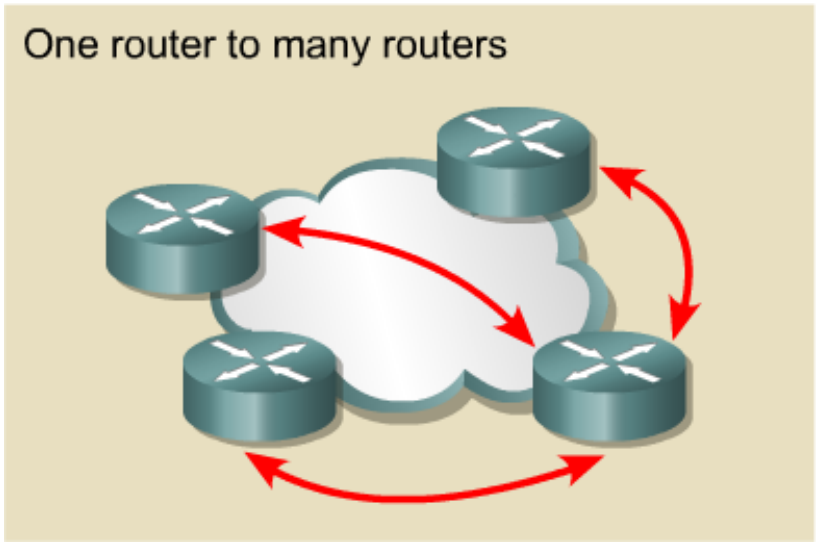
VPNs provide four critical functions:

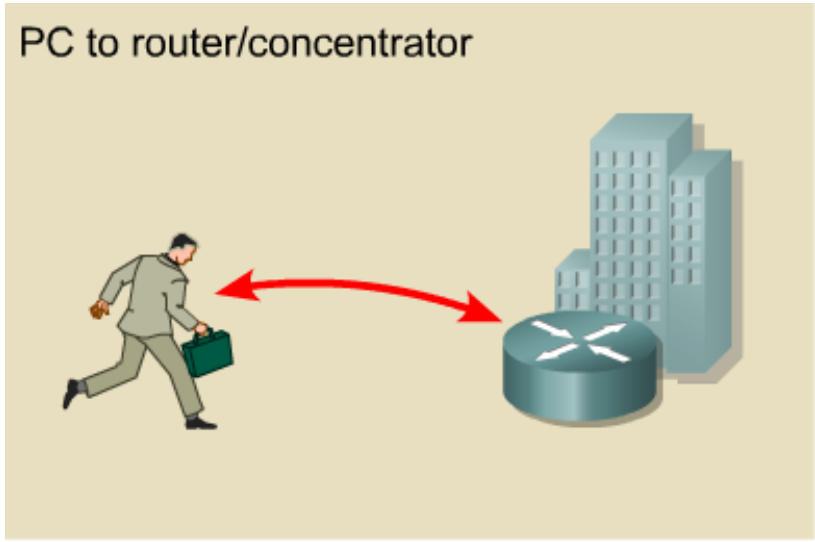
- **Confidentiality** – the sender can encrypt the packets before transmitting them across a public network
  - by doing so, no one can access the communication without permission
  - if intercepted, the communications cannot be read
- **Data integrity** – the receiver can verify that data was transmitted through the Internet without being altered
- **Origin authentication** – the receiver can authenticate the packet sender, guaranteeing and certifying the source of the information
- **User authorization** – limits unauthorized users from accessing the network

VPN Deployment Scenarios

There are **two basic VPN deployment scenarios:**

- **Site-to-Site Intranet VPN**
  - interconnects multiple network sites at different locations within the same organization
    - forms a larger corporate network
- **Remote Access VPN**
  - connect a single remote device to a corporate intranetwork
    - enable flexible access to corporate network





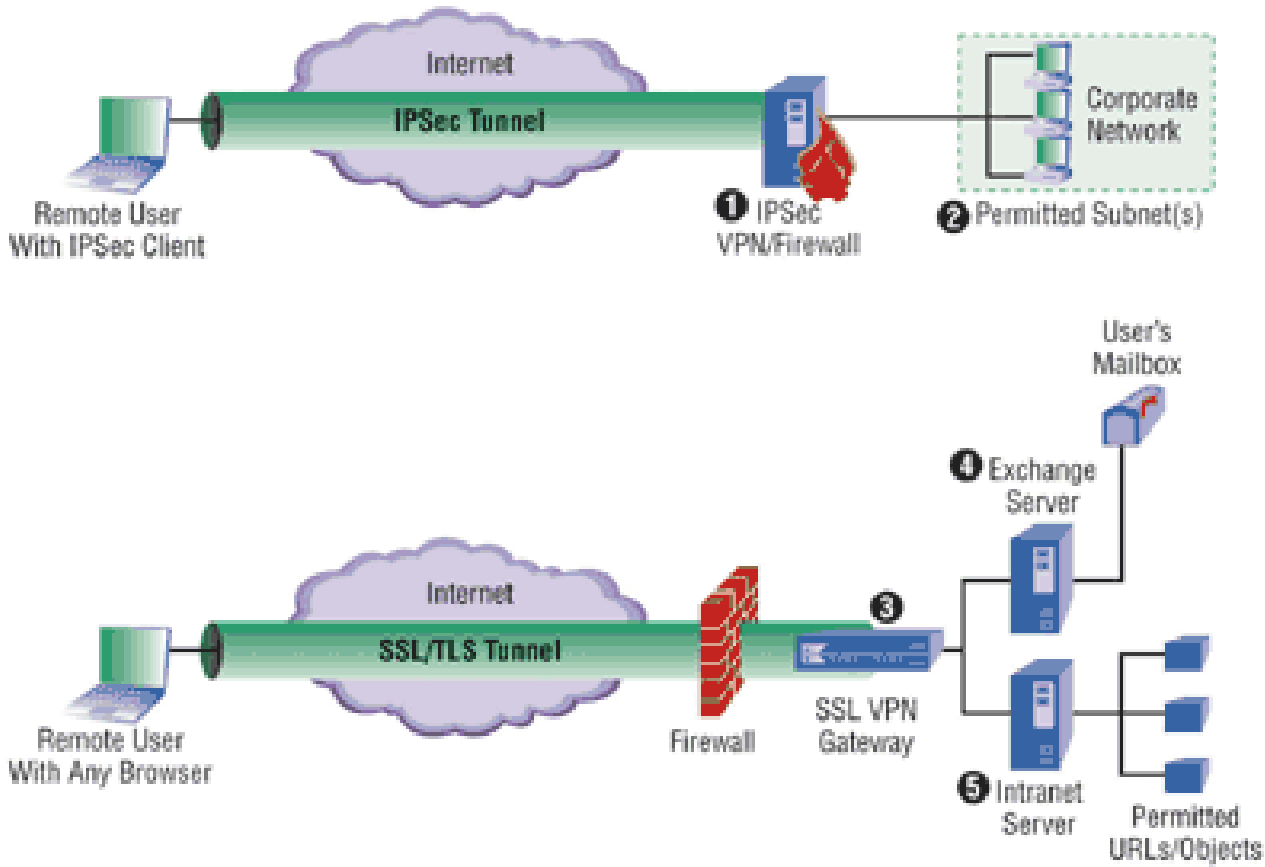
VPN Approaches

Taxonomy of **VPN approaches** based on the ISO/OSI layer:

- **Layer 2 VPN**
  - *MPLS* – Multiprotocol Label Switching
    - analogy of a virtual wire
- **Layer 3 VPN**
  - *IPSec, PPTP, L2TP*
  - usually implemented on the perimeter firewall (network border)
    - *Point-to-Point Tunneling Protocol* (obsolete) and *Layer 2 Tunneling Protocol*
    - *IPSec* – see animation at `https://frakira.fi.muni.cz/~jeronimo/vyuka/IPSec` *(part of IPv6 animation at* `https://frakira.fi.muni.cz/~jeronimo/vyuka/IPv6`*)*
- **Layer 4 VPN**
  - *SSL/TLS* VPNs
  - usually allow to access specific applications rather than entire subnets

VPN Approaches – IPSec VPN vs. SSL VPN

FIGURE 1



IPSec VPN gateways ❶ are usually implemented on the perimeter firewall, and permit or deny remote host access to ❷ entire private subnets. SSL VPN gateways ❸ are usually deployed behind the perimeter firewall, with rules that permit or deny access to application services or data. In this example, SSL users have access to their own mailboxes on ❹ an Exchange Server and to a subset of URLs hosted on ❺ an intranet Web server.

Warming QUIZ!

Warming QUIZ!

Q1: VPN stands for:

a) Virtual Public Network
b) Virtual Private Network
c) Virtual Protocol Network
d) Virtual Perimeter Network

Warming QUIZ!

Q1: VPN stands for:

a) Virtual Public Network
b) Virtual Private Network
c) Virtual Protocol Network
d) Virtual Perimeter Network

b) Virtual Private Network
(or *Virtual Private Networking*)

A VPN is a private network in the sense that it carries controlled information, protected by various security mechanisms, between known parties. VPNs are only "virtually" private, however, because this data actually travels over shared public networks instead of fully dedicated private connections.

Warming QUIZ!

Q2: What are the acronyms for the most common VPN protocols?

- identify their ISO/OSI layer as well

Warming QUIZ!

Q2: What are the acronyms for the most common VPN protocols?

- identify their ISO/OSI layer as well

Most common VPN protocols (and approaches) taxonomied by layers:

- *Layer 2* – (VPN over) MPLS
- *Layer 3* – PPTP, L2TP, IPSec
- *Layer 4* – (VPN over) SSL/TLS

Warming QUIZ!

Q3: What are the basic VPN deplyment scenarios?

Warming QUIZ!

Q3: What are the basic VPN deplyment scenarios?

There are two basic deployment scenarios:

- *Site-to-Site VPNs*
- *Remote Access VPNs*

Warming QUIZ!

Q4: What is the main benefit of VPNs compared to dedicated networks utilizing frame relay, leased lines, and traditional dial-up?

a) better network performance
b) less downtime on average
c) flexibility and reduced cost
d) improved security

Warming QUIZ!

Q4: What is the main benefit of VPNs compared to dedicated networks utilizing frame relay, leased lines, and traditional dial-up?

a) better network performance
b) less downtime on average
c) flexibility and reduced cost
d) improved security

c) flexibility and reduced cost

The main benefit of a VPN is the potential for significant cost savings compared to traditional leased lines or dial-up networking. These savings come with a certain amount of risk, however, particularly when using the public Internet as the delivery mechanism for VPN data.

Warming QUIZ!

Q5: In VPNs, the term "tunneling" refers to ...

a) an optional feature, that increases network performance if it is turned on

b) the encapsulation of packets inside packets of a different protocol to create and maintain a virtual circuit

c) the method a system administrator uses to detect hackers on the network

d) a marketing strategy that involves selling VPN products for very low prices in return for expensive service contracts

Warming QUIZ!

Q5: In VPNs, the term "tunneling" refers to . . .

a) an optional feature, that increases network performance if it is turned on

b) the encapsulation of packets inside packets of a different protocol to create and maintain a virtual circuit

c) the method a system administrator uses to detect hackers on the network

d) a marketing strategy that involves selling VPN products for very low prices in return for expensive service contracts

b) the encapsulation of packets inside packets of a different protocol to create and maintain a virtual circuit

Several computer network protocols have been implemented specifically for use with VPN tunnels – *Point-to-Point Tunneling Protocol (PPTP)*, *Layer Two Tunneling Protocol (L2TP)*, and *Internet Protocol Security (IPsec)*.

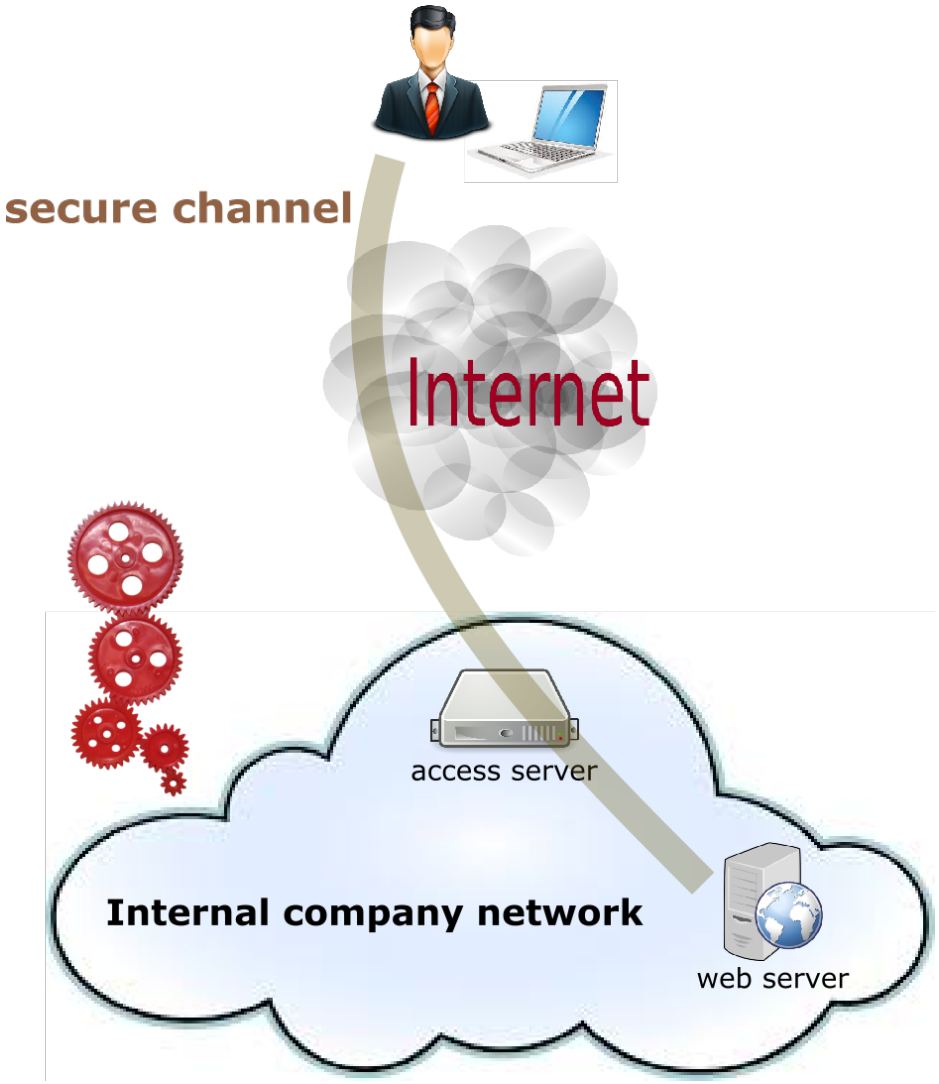
OpenVPN & practical example

OpenVPN Introduction

- VPNs can be realized both using specialized HW devices and SW tools
  - SW tools may require specific OS functionality (L2 + L3 VPNs) or not (L4 VPNs)
  - the most known and widely-used open-source SW tool is **OpenVPN**

- **OpenVPN** (`http://openvpn.net`)
  - open-source VPN solution
  - uses SSL certificates (X.509)
  - clients available for most OSes (Linux, OSX, Windows, DD-WRT, Tomato)
  - simple setup for small networks
  - user-mode, not kernel-mode
  - **the tool we will use during this practical lab**

Lab Scenario and Infrastructure



A small company called RedGears Ltd. (producing red wheels) requires you – as a network administrator – to configure the network so that their Sales Representatives can access internal network resources (webserver with internal pricelist) during travelling. All the communication has to be sufficiently secured.

Goal: Establish a VPN server (VPN gateway) and configure clients to establish a secured VPN connection.

Lab Tasks

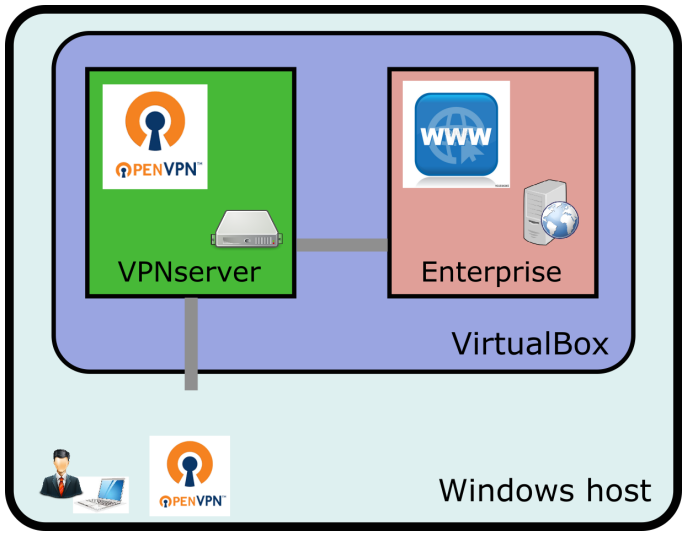
1. build the basic infrastructure
   - and test its functionality...
2. configure the OpenVPN server
   A. create server certificates
   B. create server configuration file
   C. adjust server networking configuration
   D. start and check the server
3. configure the OpenVPN client
4. connect the client to the server and observe behavior
   - both Windows and Linux clients
5. questions and another possible scenarios
6. homework assignment

1. Building the Lab Infrastructure

- **start your VirtualBox**
- import VPN server and Enterprise server VMs
  - **VirtualBox:** File → Import Appliance
    - O:\PA197\Lab 4\PA197-L4-VPNserver.ova
    - O:\PA197\Lab 4\PA197-L4-Enterprise.ova
  - do not start the VMs yet

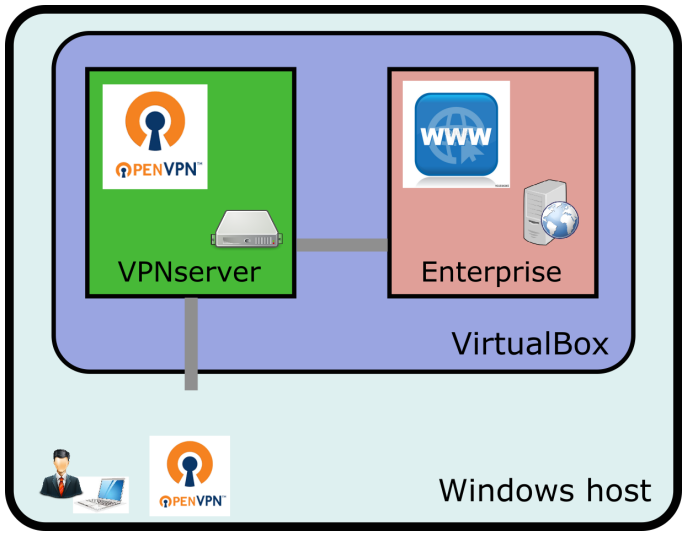
1. Building the Lab Infrastructure

- **start your VirtualBox**
- import VPN server and Enterprise server VMs
  - **VirtualBox:** File → Import Appliance
    - O:\PA197\Lab 4\PA197-L4-VPNserver.ova
    - O:\PA197\Lab 4\PA197-L4-Enterprise.ova
  - do not start the VMs yet
- observe the VMs configuration
  - network settings & port forwarding
    - note internal & external network config.



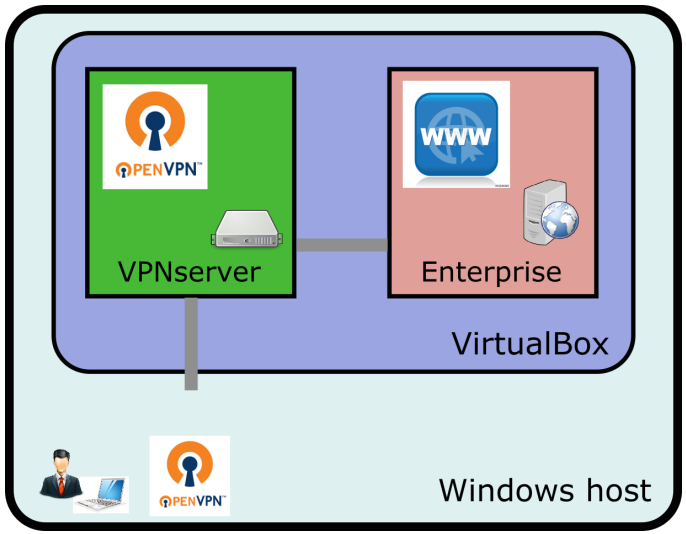
1. Building the Lab Infrastructure

- **start your VirtualBox**
- import VPN server and Enterprise server VMs
  - **VirtualBox:** File → Import Appliance
    - O:\PA197\Lab 4\PA197-L4-VPNserver.ova
    - O:\PA197\Lab 4\PA197-L4-Enterprise.ova
  - do not start the VMs yet
- observe the VMs configuration
  - network settings & port forwarding
    - note internal & external network config.
- start the VMs
  - *users:* root & pa197
    - *passwords:* pa197
  - observe the internal configuration (networking, tools, . . .)
    - make yourself root (`sudo su`)



1. Building the Lab Infrastructure

- **start your VirtualBox**
- import VPN server and Enterprise server VMs
  - **VirtualBox:** File → Import Appliance
    - O:\PA197\Lab 4\PA197-L4-VPNserver.ova
    - O:\PA197\Lab 4\PA197-L4-Enterprise.ova
  - do not start the VMs yet
- observe the VMs configuration
  - network settings & port forwarding
    - note internal & external network config.
- start the VMs
  - *users:* root & pa197
    - *passwords:* pa197
  - observe the internal configuration (networking, tools, . . .)
    - make yourself root (`sudo su`)
- test the communication
  - from *VPNserver* to *Enterprise*
    - ping, SSH, WWW browser



2. OpenVPN Server Configuration

A. Generate Certificates

- necessary for VPN server authentication
  - usable for client authentication too
- **PKI: Public Key Infrastructure** – the tools, procedures and people used to manage the creation, management and revocation of digital certificates
- **X.509** – standardized format for certificates, cert revocation and path verification Standardized by the ITU Telecommunication Standardization Sector
- **Certificate Authority** – entity that creates & signs digital certificates
- **EasyRSA SW tool** – a set of scripts allowing for the easy creation, signing and revocation of X.509 certificates used by OpenVPN
  - abstracts the use of OpenSSL (run in background)
  - distributed with OpenVPN

2. OpenVPN Server Configuration

A. Generate Certificates

- become `root`
  - `pa197@VPNserver$ sudo su -`
- **EasyRSA Setup**
  - create a CA directory with basic CA content
    - `# make-cadir /root/openvpn-ca`
  - move into that directory
    - `# cd /root/openvpn-ca`
  - configure the CA variables
    - `# mcedit vars`
      experienced users: `# vim vars`
    - see `export KEY_*` variables (not necessary to change)
    - change `KEY_NAME` to `server`
    - *variables will be used as defaults for all the generated certificates*

2. OpenVPN Server Configuration

A. Generate Certificates

- build the CA
  - source the variables into environment
    `# source vars`
    - the same as `# . vars`
  - clean previously generated keys (if any)
    `# ./clean-all`
  - build the root CA `# ./build-ca` (press ENTERs through the prompts)
    (The CA key can be password protected by using the "`--pass`" option. This password will be required to sign any certificates using the key.)
- **EasyRSA Setup** contn'd.
  - create the OpenVPN server certificate
    - `# ./build-key-server server`
      (press ENTERs & answer 'y')
  - generate strong Diffie-Hellman keys to use during key exchange
    - `# ./build-dh`

2. OpenVPN Server Configuration

B. Configure the OpenVPN service

- copy-out the CA cert and key, our server cert and key, and the Diffie-Hellman keys to OpenVPN server directory
  - `# cd /root/openvpn-ca/keys`
  - `# cp ca.crt ca.key server.crt server.key dh2048.pem /etc/openvpn`
- copy and unzip sample OpenVPN configuration file
  - `# gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz >/etc/openvpn/server.conf`
- make yourself familiar with the OpenVPN configuration
  - `# mcedit /etc/openvpn/server.conf`

2. OpenVPN Server Configuration

B. Configure the OpenVPN service

- personalize the OpenVPN server configuration
  - edit `/etc/openvpn/server.conf`
  - at least, see the options:

```
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh2048.pem                                   <-- CHANGE
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"        <-- CHANGE
cipher AES-128-CBC                              <-- CHANGE
comp-lzo
user nobody                                     <-- CHANGE
group nogroup                                   <-- CHANGE
persist-key
persist-tun
log /var/log/openvpn.log                        <-- CHANGE
```

2. OpenVPN Server Configuration

B. Configure the OpenVPN service

- set client authentication method
  - various methods available, see `https://openvpn.net/index.php/open-source/documentation/howto.html#auth`
    - authentication via a script/command (any script/command could be called, username/password passed via a file or environment variables)
    - various plugins (PAM, LDAP, htpasswd, RADIUS, etc.)
  - we will use PAM plugin (authentication against system users)
    $\rightarrow$ add the folowing options at the end of the `server.conf` file:
    - 2 lines:

      ```
      plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so login
      client-cert-not-required
      ```
- This should finalize the OpenVPN server configuration.

2. OpenVPN Server Configuration

C. Adjust server networking configuration

- allow IP forwarding
  - `# mcedit /etc/sysctl.conf`
  - remove '#' before `net.ipv4.ip_forward=1`
  - run `# sysctl --load`

D. Start and test the OpenVPN server

- reboot the server and examine log file(s) for errors
  - `# reboot`
  - once booted, run `# cat /var/log/openvpn.log`
    - has to be run with root privileges
- later, you will use common services to start/stop the OpenVPN server
  - `# service openvpn stop` (if running)
  - `# service openvpn start`

3. Configure the OpenVPN client

- prepare the client configuration file (`PA197-L4.ovpn`)
  - again, by adapting sample config file
    - `# cd /root`
    - `# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf PA197-L4.ovpn`
- and adapt it to server configuration
  - at least, see the options:

```
client
dev tun
proto udp
remote localhost 1194                <-- CHANGE
user nobody                          <-- CHANGE
group nogroup                        <-- CHANGE
persist-key
persist-tun
;ca ...                              <-- CHANGE
;cert ...                            <-- CHANGE
;key ...                             <-- CHANGE
cipher AES-128-CBC                   <-- CHANGE
comp-lzo                             <-- CHANGE
auth-user-pass                       <-- CHANGE
```

3. Configure the OpenVPN client

- include CA certificate into the client configuration file
  - attach the content of `ca.crt` file between options "<ca>" and "</ca>"

    ```
    <ca>
    ... include content of ca.crt
    </ca>
    ```
  - *Hint:* `# cat FILE1 >>FILE2`
    - `# cat /etc/openvpn/ca.crt >>PA197-L4.ovpn`
    - add <ca> and </ca> marks using an editor (just after `auth-user-pass` option)
- transfer the configuration file to the client (Windows host)
  - WinSCP from Windows host to `localhost`, port 2222
    - use `pa197` user credentials
  - and save to `C:\Program Files\OpenVPN\config\`
- finally, **try to connect to the OpenVPN server**
  - using `pa197` username and `pa197` password
  - examine the OpenVPN log files
  - if you are successfull, you should be able to access `http://10.10.10.10` from the Windows host's WWW browser

Questions & Tasks

Open network sniffer/analyzer application (Wireshark) and examine the content of the captured packets (on both VPN ends)

1. **Are the passing packets encrypted?**
2. **Are all the packets (even external) passing the OpenVPN server?**
   - if YES, how would you change the configuration so that just packets destined to the internal network(s) will go through the VPN?
   - if NO, could you capture and identify the ones not going through the VPN tunnel?

Finally, **connect to the VPN server from your Linux host** (Enterprise VM for current testing purposes).

Homework

Homeworks. . .

Your homework tasks:

1. make the example (basic) configuration **more secure**
   - *hint:* inspire at OpenVPN webpage (`https://openvpn.net`) or other pages providing tips to secure VPN tunnels (e.g. `https://blog.g3rt.nl/openvpn-security-tips.html`)
2. adapt the configurations to authenticate clients using **personal certificates** (not username & password)
3. our configuration has used so-called *routing mode* (L3-mode); try to adapt it to so-called **bridged mode** (L2-mode)
4. **optional challenge:**
   - between two Linux hosts, establish a site-to-site bridged VPN (interconnecting both networks into a single large network)

All the reports should contain **all the configuration files** (server, client) and support files (e.g. certificates), including **textual description** of all the changes performed on the server/client side (including their explanation and rationale). If you success with the bridge configuration, include small packet captures (PCAP format) as well.