L3. Virtualizare. VPN.

Conventie: Cu mici exceptii, in cadrul acestei lucrari - comenzile linux vor fi marcate cu prefixul $ sau #, ca in exemplul:

```
$ ls -l
# ifconfig eth2 down
```

Prefixul **nu** se copiaza in terminalul linux, fiind trecut doar pentru a specifica modul de lucru normal ($) sau privilegiat (#) (sudo -s) in care trebuie executate comenzile.

Etapa 1 - Configurarea mediului virtual

timp de lucru: 30-40 min

Se va realiza topologia:
Deoarece statia S1 va fi un nod central, va fi utila inspectia detaliata a fluxurilor de date. Va fi folosita o distributie Linux Ubuntu Desktop.
Nota: pentru distributiile de tip Desktop, in cadrul ecranului de autentificare va fi selectata sesiunea de tip Ubuntu Classic. Statiile C1 si C2 pot rula distributii de tip Server de mai mici dimensiuni, in linie de comanda.
Autentificarea pe ambele tipuri de distributii se va face cu username si parola ***tester***.
Se instaleaza VirtualBox:

```
# apt-get install virtualbox
```

Inaintea de inceperea lucrarii, din linie de comanda se creaza pe Desktop directorul *virtual*:

```
$ cd ~/Desktop/
$ mkdir virtual
$ cd virtual
```

Toate fisierele corespunzatoare lucrarii vor fi salvate exclusiv in acest director. Se descarca masinile virtuale:

```
$ wget http://141.85.43.141/tatcpip/Ubuntu-server.tar.gz
$ wget http://141.85.43.141/tatcpip/Ubuntu-desktop.tar.gz
$ tar xvfz Ubuntu-server.tar.gz
$ tar xvfz Ubuntu-desktop.tar.gz
$ VBoxManage clonevdi Ubuntu-server.vdi $(pwd)/C2.vdi
```

Ultima linie este folosita pentru copierea HDD-ului virtual pentru statia virtuala C2.

Crearea masinii virtuale pentru S1:
Din meniul VirtualBox se selecteaza Machine -> New;

Name:	**S1**
Operang System:	**Linux**
Version:	**Ubuntu**
Base Memory size:	**512 MB**
Virtual Hard Disk:	**Use Existing** (Si se specifica locatia fisierului Ubuntu-desktop.vdi dezarhivat).

Din setarile masinii virtuale, la sectiunea **Network**:

- Adapter 1:

Attached to: NAT,
La Avdanced:
Adapter type: Intel PRO/1000 MT Desktop (82540 OEM)
Mac address: 080027DE6439

- Adapter 2:
  Attached to: Host-only Adapter
  Name: vboxnet0
- Adapter 3:
  Attached to: Internal network
  Name: intnet1
- Adapter 4:
  Attached to: Internal network
  Name: intnet2

Notati pe hartie adresele fizice implicite ale interfetelor 2-4.
Nota: prima interfata este utilizata pentru a permite instalarea de noi pachete pe masina virtuala, iar a doua, pentru comunicatia dintre statia gazda (calculatorul fizic) si masina virtuala(si control prin ssh). Retelele de tip *intnet* sunt cele din figura de mai sus.

Dupa ce porniti masina virtuala, deschideti un terminal si dati comanda "ifconfig". Veti observa ca interfetele 1 si 2 au obtinut deja adrese prin DHCP. Mai mult, interfata corespunzatoare celui de-al doilea adaptor (se identifica prin intermediul adreselor fizice) trebuie sa aiba conectivitate cu interfata vboxnet0 de pe calculatorul gazda.
De pe calculatorul gazda, verificati cu ping conectivitatea.

Din terminalul masinii virtuale, se instaleaza server-ul ssh:
```
# apt-get install openssh-sever
```
Din acest moment, va puteti loga remote pe masina virtuala, direct de pe terminalul statiei gazda.
```
$ ssh tester@192.168.56.101
```
Se instaleaza si editorul de fisiere mcedit:
```
# apt-get install mc
```
Se shimba hostname:
```
# hostname S1
# echo S1 > /etc/hostname
```
si se modifica fisierul /etc/hosts astfel:
```
127.0.0.1       localhost
127.0.1.1       S1
```

Instalarea masinilor C1 si C2 va fi similara, cu exceptiile:

- C1 va avea ca imagine de HDD fisierul Ubuntu-server.vdi, iar C2 imaginea C2.vdi
- Pentru ambele masini virtuale se configureaza Adapter1 (NAT) - fara a se modifica adresa fizica, Adapter2 (Host-only) si Adapter3 (cu intnet-ul corespunzator - din figura)
- Din terminalele masinilor virtuale se identifica interfetele 1 si 2 si se configureaza prin DHCP (ex: $ dhclient *eth2*)

Suplimentar, pe S1 se instaleaza wireshark:
```
# apt-get install wireshark
```

Etapa 2 - Instalarea si configurarea serverului OpenVPN (Routed VPN)

Se instaleaza OpenVPN pe cele trei masini virtuale:
```
# apt-get install openvpn
```

Pe **C1**, se genereaza cheile pentr server-ul si clientii OpenVPN:

```
# cd /usr/share/doc/openvpn/examples/easy-rsa/2.0/
```

Pentru configurarea CA master, se editeaza valorile KEY_COUNTRY, KEY_PROVINCE, KEY_CITY, KEY_ORG, KEY_EMAIL din fisierul vars. Tinand cont ca acesta este un exercitiu de laborator, pot fi pastrate si valorile initiale:

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
```

Initializarea PKI:

```
. ./vars
./clean-all
./build-ca
Generating a 1024 bit RSA private key
.......++++++
.........................++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
State or Province Name (full name) [CA]:
Locality Name (eg, city) [SanFrancisco]:
Organization Name (eg, company) [Fort-Funston]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [Fort-Funston CA]:
Name []:
Email Address [me@myhost.mydomain]:
```

Generarea cheii si certicicatului pentru server:

```
./build-key-server server
```

Ca in cazul precedent, se lasa valorile implicite pentru majoritatea campurilor, dar doua intrebari vor necesita raspuns pozitiv:

```
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
```

Se genereaza cheile celor doi clienti:

```
./build-key client1
./build-key client2
```

Generarea parametrilor Diffie-Hellman:

```
./build-dh
```

Cheile si certificatele proaspat generate pot fi gasite in subdirectorul **keys**:

Filename	Needed By	Purpose	Secret
ca.crt	server + all clients	Root CA certificate	NO
ca.key	key signing machine only	Root CA key	YES
dh{n}.pem	server only	Diffie Hellman	NO

		parameters	
server.crt	server only	Server Certificate	NO
server.key	server only	Server Key	YES
client1.crt	client1 only	Client1 Certificate	NO
client1.key	client1 only	Client1 Key	YES
client2.crt	client2 only	Client2 Certificate	NO
client2.key	client2 only	Client2 Key	YES

Se copiaza cheile in directorul openvpn atat de pe server, cat si pe clienti.
Pe server:

```
# cd /usr/share/doc/openvpn/examples/easy-rsa/2.0/keys
# cp ca.crt dh1024.pem server.key server.crt /etc/openvpn/
# scp ca.crt client1.crt client1.key tester@192.168.56.102:/home/tester/
# scp ca.crt client2.crt client2.key tester@192.168.56.103:/home/tester/
```

Pe C1, C2:

```
# mv /home/tester/client* /etc/openvpn/
# mv /home/tester/ca.crt /etc/openvpn/
```

Configurarea server-ului si clientilor

Vom folosi ca punct de plecare exemplele de configuratii ale platformei, care pot fi gasite cel mai probabil in directorul /usr/share/doc/openvpn/examples/sample-config-files. Se copiaza fisierul server.conf pentru S1, respectiv client.conf pentru C1 si C2 in directorul /etc/openvpn/. Daca fisierul server.conf este comprimat .gz, se va dezarhiva cu *gunzip*.

```
S1:
# gunzip server.conf.gz
# cp server.conf /etc/openvpn/
C1, C2
# cp client.conf /etc/openvpn/
```

OpenVPN poate functiona in modul Routed sau Ethernet Bridged. Fiind mai usor de configurat, vom lucra in mod routed VPN in cadrul acestui prim exercitiu (exemplele de configuratii copiate sunt pentru modul routed VPN).

Editarea configuratiei S1

Minimul necesar pentru a avea o configuratie functionala este specificarea cheilor mai sus create, prin parametrii **ca**, **cert**, **key**, si **dh**. Daca au fost urmati toti pasii anteriori, cheile si certificatele ar trebui sa fie in acelasi director cu fisierul configuratie, iar configuratia nu va necesita modificari.
Se porneste OpenVPN:

```
# openvpn /etc/openvpn/server.conf
```

Editarea configuratiei C1

Daca au fost urmati toti pasii anteriori, cheile si certificatele ar trebui sa fie in acelasi director cu fisierul configuratie; se modifica fisierul client.conf astfel:

```
ca ca.crt
cert client1.crt
key client1.key
```

Tot in fisierul client.conf se modifica parametrul **remote** pentru a indica adresa de intnet a lui S1 vizibila dinspre C1(192.168.1.1).

```
remote 192.168.1.1 1194
```

Se porneste OpenVPN:

```
# openvpn /etc/openvpn/client.conf
```

Analog pentru C2.

Implicit, pe C1 si C2 ar trebui sa apara cate o interfata tun0, avand adresa in gama 10.8.0.0/24. Verificati cu ping conectivitatea dintre C1 si C2, respectiv C1/C2 si adresa 10.8.0.1. Functioneaza ping in ambele cazuri?

Editati configuratia server-ului decomentand linia *client-to-client*, reporniti server-ul si clientii OpenVPN si reincercati.
Inspectati traficul de pe interfetele intnet ale lui S1, folosind wireshark din interfata grafica.

Etapa 3 - Bridged VPN

Optional; timp de lucru: 10-20 min

Copiati configuratiile anterior create (in directorul /etc/openvpn/):
Pentru S1:

```
# cp server.conf serverL2.conf
```

Pentru C1/C2:

```
# cp client.conf clientL2.conf
```

In serverL2.conf:
* se comenteaza **dev tun** si se decomenteaza **dev tap**
* se comenteaza linia care incepe cu parametrul **server** (*server 10.8.0.0 255.255.255.0*)
* se adauga linia:

```
server-bridge 192.168.8.4 255.255.255.0 192.168.8.128 192.168.8.254
```

In clientL2.conf:
* se comenteaza **dev tun** si se decomenteaza **dev tap**

Pe S1:

```
# apt-get install bridge-utils
# cd /usr/share/doc/openvpn/examples/sample-scripts/
# ./bridge-start
# cd /etc/openvpn/
# openvpn /etc/openvpn/serverL2.conf
```

Porniti si clientii OpenVPN si testati conectivitatea.
Configurati dinnou interfata eth0 a lui S1

```
# dhclient eth0
```

si testati.
Dupa incheierea activitatii de laborator se vor sterge masinile si HDD-urile virtuale din VirtualBox, respectiv directorul *virtual* creat la inceputul lucarii.