IPSec with GNS3 on Linux

Miguel Frade

Department of Informatics Engineering
Polytechnic Institute of Leiria

Introduction

What is GNS3?

- Graphical Network Simulator-3 (GNS3) is a software emulator for networks;
- It allows the combination of virtual devices and real devices;
- Can be used to simulate complex networks;
- It uses Dynamips emulation software to simulate Cisco IOS;
- Launched in 2008;
- Available for Windows and Linux `http://www.gns3.com/`

Introduction

How to install the latest version of GNS3 on Linux and other important tools?

```
sudo add-apt-repository ppa:gns3/ppa                    # add a new repository
sudo apt-get update                                     # get list of SW on all repositories
sudo apt-get install gns3-gui wireshark uml-utilities   # install tools
```

Introduction

How to install the latest version of GNS3 on Linux and other important tools?

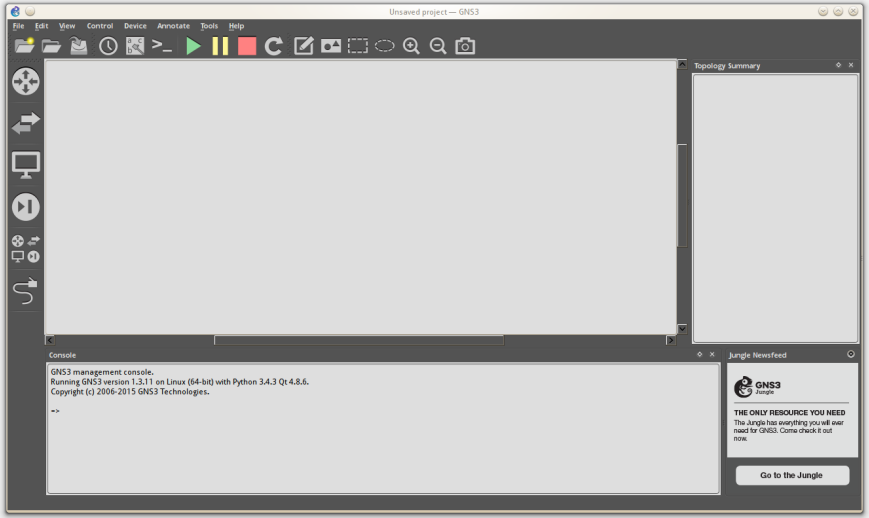
```
sudo add-apt-repository ppa:gns3/ppa                    # add a new repository
sudo apt-get update                                     # get list of SW on all repositories
sudo apt-get install gns3-gui wireshark uml-utilities   # install tools
```

Cisco's routers operating system IOS

- Download Cisco IOS from Moodle (click on this link)
- decompress it to speedup routers initialization (optional)

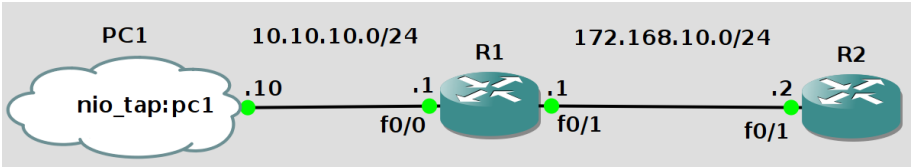
```
unzip -p c2691-adventerprisek9-mz.124-25d.bin \
> c2691-adventerprisek9-mz.124-25d_decompressed.bin
```

Introduction – GNS3 screen version 1.3.11



Introduction

Our goal is to configure this network



1. be able to ping from PC1 to R2
2. create an IPSec tunnel between R1 and R2

Configurations on the Linux command line

- add virtual network interface to enable the usage of the “cloud” option

```
sudo tunctl -t pc1 -u $USER          # create pc1 virtual network interface
sudo ip l s dev pc1 up               # activate pc1 interface
sudo ip a a 10.10.10.10/24 dev pc1   # setup IP address for pc1
```

Configurations on the Linux command line

- add virtual network interface to enable the usage of the “cloud” option

```
sudo tunctl -t pc1 -u $USER          # create pc1 virtual network interface
sudo ip l s dev pc1 up               # activate pc1 interface
sudo ip a a 10.10.10.10/24 dev pc1   # setup IP address for pc1
```

- add static route to reach R2 from your computer

```
sudo ip route add 172.168.10.0/24 dev pc1
```

Configurations on the Linux command line

- add virtual network interface to enable the usage of the "cloud" option

```
sudo tunctl -t pc1 -u $USER          # create pc1 virtual network interface
sudo ip l s dev pc1 up               # activate pc1 interface
sudo ip a a 10.10.10.10/24 dev pc1   # setup IP address for pc1
```

- add static route to reach R2 from your computer

```
sudo ip route add 172.168.10.0/24 dev pc1
```

- or use the scripts available on Moodle to do the same configurations:

```
# create virtual interface pc1, set IP address and network mask
sudo ./tap_create.sh pc1 10.10.10.10/24
# delete virtual interface pc1
sudo ./tap_delete.sh pc1
# create static route to 172.168.10.0/24 through pc1
sudo ./route_create.sh 172.168.10.0/24 pc1
# delete static route to 172.168.10.0/24
sudo ./route_delete.sh 172.168.10.0/24 pc1
```

Configurations on GNS3

Configure console application:

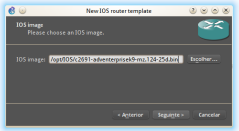
- `Edit` → `Preferences ...` → `General` → `Console applications`
- avoid the less friendly XTerm terminal option
- choose `Gnome Terminal`, for Unity, or Gnome graphical interface
- or `KDE konsole` for KDE graphical interface
- then press the `Set` button
  - for `KDE konsole` you must correct the command:

```
# remove the quotation marks, they cause an error
# the final command should look like this
konsole --new-tab -p tabtitle=%d -e telnet %h %p
```

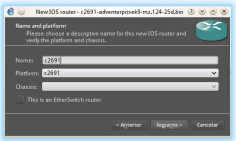
Configurations on GNS3 – add an IOS image

- `Edit` → `Preferences ...` → `IOS routers` → `New`
- choose `(1) c2691-adventerprisek9-mz.124-25d_decompressed.bin` → `(1) Next` → `(2) Next`
- `(3)` set RAM to `128 MB` → `(3) Next` → `(4) Next` → `(5) Next` → `(6) End`

1

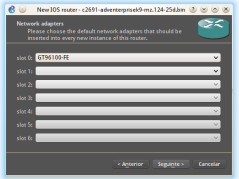


2

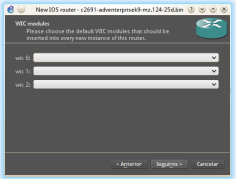


3

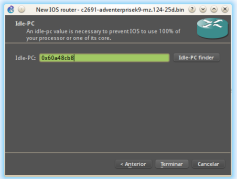
4



5

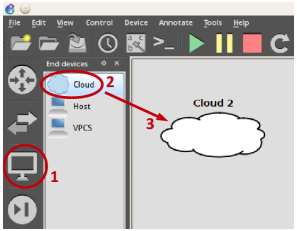


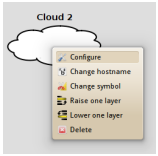
6

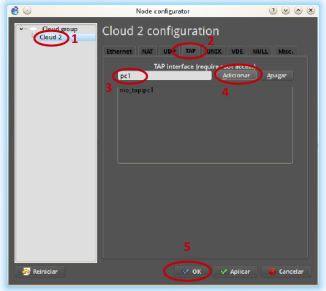


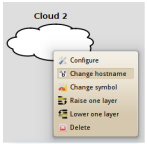
Setup network scenario

Add and configure a cloud



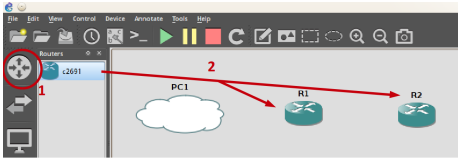


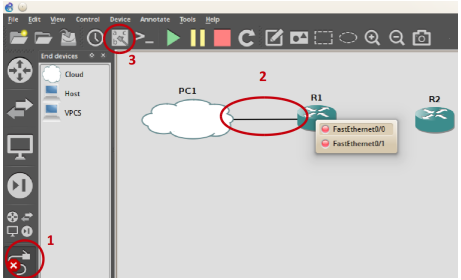




Setup network scenario

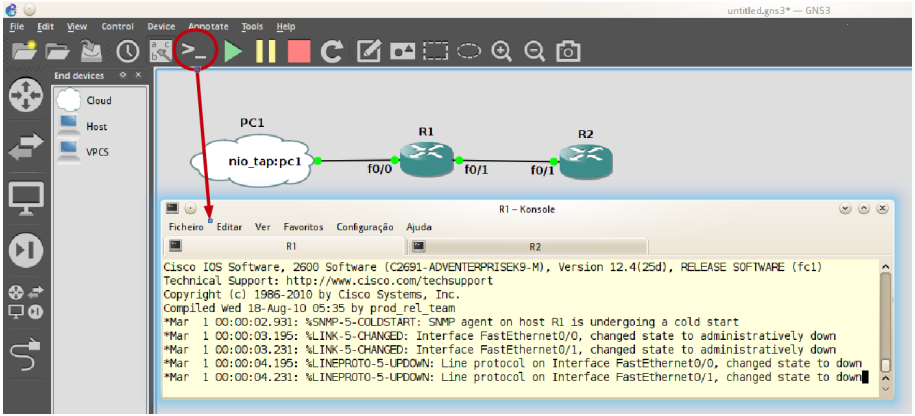
Add routers and connections





Setup network scenario

Open routers' console for configuration



Configure routers' interfaces

Configure R1 with this commands

```
R1$ conf t                                          # enter configuration mode
R1(config)$ interface f0/0                          # config. interface f0/0
R1(config-if)$ ip address 10.10.10.1 255.255.255.0
R1(config-if)$ no shutdown                          # activate interface f0/0
R1(config-if)$ exit                                 # leave interface f0/0
R1(config)$ interface f0/1
R1(config-if)$ ip address 172.168.10.1 255.255.255.0
R1(config-if)$ no shutdown
R1(config-if)$ exit
R1(config)$ router eigrp 100                        # setup routing protocol
R1(config-router)$ network 10.10.10.0 0.0.0.255
R1(config-router)$ network 172.168.10.0 0.0.0.255
R1(config-router)$ no auto-summary
R1(config-router)$ ^Z                               # ctrl+z leaves config mode
R1$ wr                                              # saves configurations
```

Configure routers' interfaces

Configure R2 with this commands

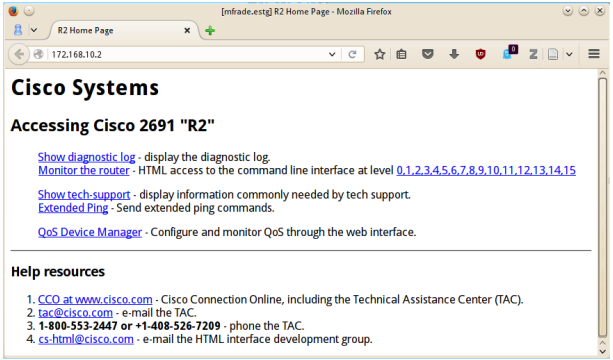
```
R2$ conf t
R2(config)$ interface f0/1
R2(config-if)$ ip address 172.168.10.2 255.255.255.0
R2(config-if)$ no shutdown
R2(config-if)$ exit
R2(config)$ router eigrp 100
R2(config-router)$ network 172.168.10.0 0.0.0.255
R2(config-router)$ no auto-summary
R2(config-router)$ exit
R2(config)$ ip http server                                  # start HTTP server on R2
R2(config)$ ip http authentication local                    # setup authentication mode
R2(config)$ username admin privilege 15 password cisco      # define username and password
R2(config)$ ^Z                                              # ctrl+z leaves config mode
R2$ wr                                                      # save configurations
```

Test the connections from PC1

- in the command line

```
ping 10.10.10.1        # ping first R1 interface
ping 172.168.10.1      # ping second R1 interface
ping 172.168.10.2      # ping R2 interface
```

- in the browser go to `http://172.168.10.2` username: `admin` password: `cisco`



Exercise 1 – Configure R1

```
R1$ conf t
R1(config)$ crypto isakmp enable
R1(config)$ crypto isakmp policy 110          # define a ISAKMP policy set with proiority 110
R1(config-isakmp)$ authentication pre-share   # pre-shared authentication
R1(config-isakmp)$ encryption des             # create IKE tunnel
R1(config-isakmp)$ group 5                    # Set the Diffie-Hellman key size group
R1(config-isakmp)$ hash sha                   #
R1(config-isakmp)$ lifetime 86400             # set SA duration
R1(config-isakmp)$ exit
R1(config)$ crypto isakmp key 0 cisco-ss address 172.168.10.2
R1(config)$ crypto ipsec transform-set TSET esp-des  # defines the IPSec protocol
R1(cfg-crypto-trans)$ mode tunnel
R1(cfg-crypto-trans)$ exit
```

Exercise 1 – Configure R1

```
# specifies what trafic must go through the IPSec tunnel, all IP packets
R1(config)$ access-list 105 permit ip 10.10.10.0 0.0.0.255 host 172.168.10.2

R1(config)$ crypto map MYMAP 10 ipsec-isakmp      # aggregates:
R1(config-crypto-map)$ match address 105          #   the ACL to define the packets
R1(config-crypto-map)$ set transform-set TSET     #   the IPSec protocol
R1(config-crypto-map)$ set peer 172.168.10.2      #   the other end of the tunnel
R1(config-crypto-map)$ exit
R1(config)$ interface f0/1
R1(config-if)$ crypto map MYMAP                   # sets interface to be used by IPSec
R1(config-if)$ ^Z
R1$ wr
```

Exercise 2 – Configure R2

1. Configure R2 symmetricaly to R1
2. Test configuration
   - with `ping`
   - then do `show crypto ipsec sa`.
     - is the packets count different from zero?
   - capture packets with Wireshark
     - can you see ESP packets?
   - Export routers configuration file → **this is important for the next test** ←
     - select a router, right click on it and select `Export config`
     - open the file and check the configurations you made
     - compare the configuration files of R1 and R2

Exercise 3 – For the same network scenario

Set up an IPSec tunnel between `R1` and `R2` to protect only the `HTTP` traffic originating from the network `10.10.10.0/24` with destination to `R2`.

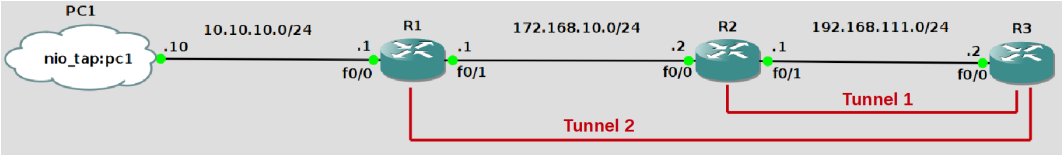
Data confidentiality of `HTTP` packets, must be ensured by `AES` and authentication ensured through the `MD5` algorithm. The Security Associations lifetime must be `3` hours and the IKE tunnel should be configured with the pre-shared key `I-love-IPSec` and the key exchange must be done by the DH algorithm group `5`. The integrity of ISAKMP communications should be guaranteed with the `SHA` and the confidentiality with `3DES`.

For paramaters not specified above, use the defaults values.

Exercise 4 – Configure 2 IPSec tunnels

Setup 2 IPSec tunnels:

1. Tunnel 1: TSET = `ESP` with `AES` and `HMAC-SHA`, only for `ICMP` packets
2. Tunnel 2: TSET = `AH` with `HMAC-MD5`, only for `HTTP` from `10.10.10.0/24` to host `192.168.111.2`



Test IPSec tunnels and verify the traffic with Wireshark