Network security – basic attacks

Laboratory for the class "Information Systems Security" (01TYM, 02KRQ)
Politecnico di Torino – AA 2023/24
Prof. Diana Berbecaru

prepared by:
Diana Berbecaru (diana.berbecaru@polito.it)
Enrico Bravi (enrico.bravi@polito.it)

v. 1.0 (03/10/2022)

Contents

Purpose of this laboratory

The widespread use of networks and especially of Internet has created on one hand a virtual workspace enabling people to collaborate and communicate in real time, but at the same time it opened up the door to cyberspace attackers. Even if the most sophisticated network security attacks can only be performed by very experienced people with strong motivations and budget, the exercises proposed in this laboratory will prove that also people with basic security notions and only using freely available software can be very dangerous in some contexts.

ATTENTION

Some of the operations described in this document are illegal and are liable to prosecution. The purpose of this document is to present the actions (and tools) just for educational purposes. Students are required to try the attacks only towards the computers located in the laboratory dedicated to the course "Information Systems Security", or towards the computers they own. The authors of this document decline any responsibility for actions performed by the participants of this laboratory in violation of this policy.

Software tools

The tools listed below will be used throughout this (practical) laboratory:

Nmap - network and port scanner (open source). It is designed to perform quick scanning of large networks. For example, it allows to determine which type of operating system a particular network host is running and which (application) services are active at one time. Available for Linux and Windows.
Home page = https://nmap.org

Ettercap - open source tool, which allows to perform man-in-the-middle (MITM) attacks and sniffing attacks in a Local Area Network (LAN). Available for Linux e Windows.
Home page = http://ettercap.github.io/ettercap/

Wireshark - open source tool (having a user-friendly graphical interface) which allows to capture network traffic. Available for Linux and Windows.
Home page = https://www.wireshark.org/

GVM - open source tool that perform vulnerability scanning. It allows to detect nodes that are subject to known software vulnerabilities, as well as insecure configurations of the service(s) running on the network hosts. Disponibile per Linux e Windows.
Home page = https://openvas.org/

Additional tools

When executing the exercises, you will need also to start, stop or reconfigure some services:

Apache2 - Web server
To start, stop and restart the web server, use the command:
`sudo systemctl { start | stop | restart } apache2`

The configuration of server ports (e.g. the listening port) can be found in the file
`/etc/apache2/ports.conf`

VSFTP - FTP server
To start, stop and restart the FTP server, use the command:
`systemctl { start | stop | restart } vsftpd`

The configuration of server ports can be found in the file
`/etc/vsftpd.conf`

For example, to change the `listen_port` to 201, you need to add/modify the line
`listen_port=201`

SSH2 - SSH server
To start, stop and restart the SSH server, use the command:
`sudo systemctl { start | stop | restart } ssh`

The configuration of the listening port on the server is found in `/etc/ssh/sshd_config`.

Before starting the server for the first time, you must generate the keys of the host with the command:

```
ssh-keygen -A
```

Exim - Mail server
To start, stop and restart the mail server, use the command:
`sudo systemctl { start | stop | restart } exim4` The configuration of the ports can be modified in the file `/etc/default/exim4`.

Using the arp commands

You can delete an IP address from the ARP table by using the `arp` command with the `-d` option followed by the address to remove. For example, to delete the IP address 192.168.1.44:

```
arp -d 192.168.1.44
```

If you are not sure which IP address you are looking for, then you can look at the ARP table to check the IP against the MAC address by using the `-e` option. For example:

```
arp -e
```

Use of IP

The command `ip` is a powerful command that can be used to analyse and manipulate the routing of IP packets on a Linux machine.

It allows also to remove all the entries of an ARP table (this operation cannot be done in a single step with the `arp` command) with the following command:

```
ip neigh flush all
```

It is useful to add the flag `-s` (repeated twice) in order to print information about the result of the command:

```
ip -s -s neigh flush all
```

During the laboratory, you will need to execute commands that require `root` privileges. For this reason, we suggest you to switch to the super user with `sudo su` from the beginning of the practical exercises.

1 Information gathering

1.1 Network Scanning

The first phase of the preparation of an attack consists in detecting the target(s) of the attacks. For this purpose, a technique called *Network Scanning* is typically used. The objective of this technique is to obtain information about which hosts in a particular network are active and which ones are not active.

How can this be done in practice? Which is the most common method used to determine whether a certain host is reachable?

→

Why (in general) it is not recommended to perform network scanning inside a monitored network?

→

Once you have identified the active network hosts that are actually running, you can proceed to gain more information about them in order to detect their security weaknesses. In practice, a series of techniques known as *Network Fingerprinting* allow to obtain information on the remote host's operating system. The *Network Fingerprinting* technique is based on the fact that various types of operating systems (e.g. Windows and Linux) implement differently the TCP/IP stack.

The program named `nmap` is a very easy-to-use and powerful network scanner, which allows to detect the operating system running on a remote host, by means of network fingerprinting. For a description of the overall program, use the following command:

```
man nmap
```

Exercise 1. Two students form a group of two actors named Alice and Bob (one for the victim and one for the attacker), then proceed as follows:

1. Alice (victim): starts the Apache web server.
2. Bob (attacker): tries to establish a TCP connection (`-sT`) on the port 80 (`-p 80`) of the target host Alice (*IPaddress_Alice*), in order to obtain information about the operating system (`-O`) running on the victim's machine:

   ```
   nmap -sT -p 80 -O -v IPaddress_Alice
   ```

1.2 Port Scanning

After discovering the target host, the attacker typically tries to find out which (application) services are actually running on that host. The technique known as *Port Scanning* is used for this purpose. To obtain information

about which ports of a particular host are open (waiting for incoming connections) and which ones are closed, a *Port Scanning* tool can be used.

By using such a tool, it is possible also to determine for each port:

- the (default) name of the known service (if one exists, like for example `http` service or `ftp` service), item the port number,
- the port's state (such as `open`, `filtered` by a firewall or by a packet filter, `unfiltered`)
- the corresponding protocol.

For a description of the various types of port scanning currently supported by Nmap, you can run the corresponding `man` command, i.e. `man nmap`.

Exercise 2. Two students form a group of two, named Alice and Bob (for the victim and the attacker), and proceed as follows:

1. Alice (victim): chooses two services among `http`, `ftp`, `ssh` and activate them.
2. Alice checks that only the services she has selected are actually active (check out by using `lsof -i` or `netstat -ltu`)
3. Bob (attacker): makes a TCP port scan (`-sT`) on the first 1024 ports of the target host Alice (*IPaddress_Alice*):

   ```
   nmap -sT -Pn -p 1-1024 -v IPaddress_Alice
   ```

Do you manage to discover which services have been activated by the victim (Alice)?

→

Why is it useful to avoid pinging the host (the option `-Pn`)?

→

Exercise 3. Now execute the following operations:

1. Alice (victim) chooses again two services among `http` (i.e. Apache2), `ftp` (i.e. VSFTP), `ssh` (i.e. SSH2)
2. Next, she starts the chosen services on ports different from the default ones (see the introductory notes for instructions on changing the ports).
3. Bob (attacker) performs a new TCP port scanning on the host of her colleague (Alice) with the same command as above:

   ```
   nmap -sT -Pn -p 1-1024 -v IPaddress_Alice
   ```

Have you encountered problems? How do you think they can be solved?

→

1.3 Identification of (application) services

Exercise 4. To identify the (application) services running on the open ports on your colleague's machine, try to use the option (`-sV`):

```
nmap -sV -Pn -p 1-1024 -v IPaddress_Alice
```

Nmap can exploit also the flag `-A` that activates a more aggressive scanning. By using this flag, more tests are performed and more information is provided on the target machine. Run the following command to note the differences:

```
nmap -A -Pn -p 1-1024 -v IPaddress_Alice
```

For a detailed description of the techniques used by Nmap to identify the services, you can use `man nmap`.

NOTE

To perform an in-depth scanning of the services and to discover their potential vulnerabilities you can use a vulnerability scanner like GVM (as further described in the optional exercises).

Why is it recommended to use the flag `-sV` rather than the option `-A` ?

→

In your opinion, which techniques and security tools can be used to defend against the security attacks presented above?

→

NOTE

Nmap has also the flag `-T` that accepts an integer parameter (from 0 to 6) used to set internal timers for the scanning operations. Lower is this value higher is the time required to perform scanning, because more time passes between successive requests. However, the benefit of using small values is that the scanning will result more "invisible" (difficult to detect) because less it will generate less "anomalous" traffic that could be detected by monitoring devices.
The default value is 3, while the values 0 and 1 are used typically to perform an *IDS evasion*. For example, to perform a more "invisible" scanning of Alice you can use the command:

```
nmap -T0 -sV -Pn -p 1-1024 -v IPaddress_Alice
```

2 Capture and manipulation of network traffic

2.1 Man-in-the-middle

Ettercap is a versatile tool that can be used to execute (besides sniffing, filtering etc) man-in-the-middle attacks in a LAN. In particular, in this exercise, we will concentrate on the "ARP poisoning" attack.

For a description of the parameters of Ettercap tool and of its configuration, you can run the following commands:

```
man ettercap
```

```
man etter.conf
```

Given the type of attack, you can use the command `arp` to view the content of the ARP cache:

```
man arp
```

Exercise 5. Three students form a group of three VMs (let's say Alice, Bob and Chuck) and proceed as follows:

1. Examine the operation of the network in normal conditions: Alice, Bob e Chuck exchange ping messages and note down the ARP cache of each host in Table **??**.
2. Chuck tries to sniff the network traffic with `wireshark`
3. Chuck launches the attack, by starting ettercap in the following way (pay attention to the "/" characters):

   ```
   ettercap -Tq -M arp /IPaddr_host_Alice// /IPaddr_host_Bob//
   ```

   The program starts in interactive mode. Chuck can view a brief help, by pressing the button `h`. While executing the command, check out the network traffic on the three hosts by capturing the packets with `wireshark`.
4. Alice and Bob exchange again ping messages.

What happened to the ARP caches of Alice, Bob and Chuck? Verify your hypothesis by printing out the content of the ARP caches with the `arp` command in Table **??**.

Role	IP Address	MAC Address
Alice		
Bob		
Chuck		

Table 1: *ARP cache(s) content before the attack*

Role	IP Address	MAC Address
Alice		
Bob		
Chuck		

Table 2: *ARP cache(s) content after the attack*

What kind of network traffic was able to capture Chuck before the attack?

→

What type of network traffic is able to sniff Chuck after the attack?

→

Can you figure out from the captured network traffic how the attack works?

→

What happened to the ARP tables of Alice, Bob and Chuck? Check out.

→

Take a look at the Figure **??** and try to respond to this question (where `?` can be a *hub*, a *switch*, a *router* or even *Internet*): when is it possible to make a man-in-the-middle attack with ARP poisoning and which are its effects?

→

Optionally, try to identify a possible countermeasure to this security attack:

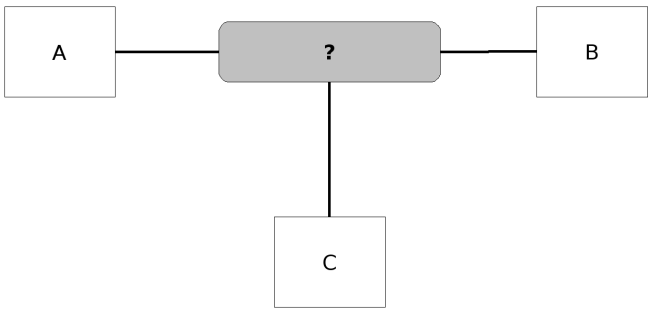

Figure 1: man-in-the-middle

1. Chuck modifies the line `arp_poison_delay` in the file `/etc/ettercap/etter.conf`, by setting it to `1000`. Next he restarts `ettercap`.
2. Bob deletes from the ARP cache the line corresponding to the host of Alice. After that, he tries to execute a `ping` to Alice's host.

What happened?

What Chuck did at point 1?

What is the meaning of the variable `arp_poison_delay`? Prove again the exercise by setting the variable to 1 or to the default value, 10.

→

2.2 Sniffing attacks

In this section, we'll experiment with the passive network security attacks. In practice, *sniffing* is a passive attack that consists in capturing the packets passing through our Ethernet network card set in promiscuous mode. In this way all the packets that should be ignored, i.e. the ones that don't correspond to the MAC address of the network card, are instead copied in a buffer.

Actually, you have already tried the sniffing technique in the exercises proposed above, when you have captured the network packets with `wireshark`. In this exercise, we will concentrate on capturing sensitive data (e.g. passwords) exchanged inside application protocols, instead of capturing generic data traffic.

To experiment the sniffing attack, we will use again `ettercap`, which extracts username and password from network traffic exchanged in common services (including FTP). Another useful tool is the `Ngrep` tool (`http://ngrep.sourceforge.net`), which allows to search for pre-defined strings in the captured traffic. In practice, `Ngrep` provides functionality similar to the classical `grep` command, which is applied to the (captured) network traffic.

Exercise 6. Three students form a group of three VMs (Alice, Chuck and Bob):

1. Alice starts her FTP server and creates the users `alice` and `bob`, and sets accordingly the passwords:

```
adduser alice
```

```
adduser bob
```

2. Chuck launches the MITM attack with `ettercap` as in the Exercise 5.
3. Bob uses the `ftp` command to connect to the FTP server running on the machine of Alice, he executes login (with his username and the password set above) and then he disconnects

NOTE

In some cases, the following exercise (option `-e` of `ettercap`) does not function correctly with the virtual machines. The ARP spoofing is performed correctly, but the identification module of regexp does not manage to recognize the strings.

Exercise 7. Try now to extract sensitive information from the mail traffic, by executing the following steps (Alice must have already created users `alice` and `bob` through the `adduser` command as previously explained):

1. Alice configures the `exim` mail server with the command:

   ```
   dpkg-reconfigure exim4-config
   ```

   and by selecting the parameters in the following manner:

   (a) General type of mail configuration: Internet site; mail is sent and received directly using SMTP.
   (b) System mail name: kali
   (c) IP-addresses to listen on for incoming SMTP connections: *// leave blank (delete data if present)*
   (d) Other destinations for which mail is accepted: kali
   (e) Domains to relay mail for: *// leave blank (delete data if present)*
   (f) Machines to relay mail for: *// leave blank (delete data if present)*
   (g) Keep number of DNS-queries minimal (Dial-on-Demand) ?: No
   (h) Delivery method for local mail: mbox format in /var/mail
   (i) Split configuration into small files ? : No
   (j) Root and postmaster mail recipient: *// leave blank (delete data if present)*

   Subsequently starts the server with the command:

   ```
   systemctl start exim4
   ```

2. Chuck executes the command:

   ```
   ettercap -T -M arp /IP_host_Alice//25 /IP_host_Bob// -e ¨Credit Card¨
   ```

3. Bob sends an e-mail to `alice@kali` with the following message “Don’t do this in practice: do not send any Credit Card number (like 7865-8993-6282-8282) by mail!” by using the e-mail client named `s-nail`

   ```
   s-nail -S mta=smtp://IP_host_Alice -S 'from=bob@kali' -s ¨Exam Security¨
   alice@kali
   ```

   once you have finished editing the e-mail text message press Enter then `Ctrl-D`

4. Alice can verify that the user `alice` has received the message with the following command (don’t waste time configuring an e-mail client for this exercise)

   ```
   cat /var/mail/alice
   ```

   Take a look on the output shown on Chuck machine.

Exercise 8. Try now to spy the web navigation, by executing the following steps:

1. Alice activates the service `http`
2. Chuck enables the access to the display for the user `root`, by opening a terminal and executing the command:

   ```
   xhost +
   ```

3. Chuck, in the Kali menu, opens the `Default Applications` and sets Firefox as predefined Web Browser (if Firefox is not shown in the Kali menu then select `Other...` and set `/root/firefox/firefox "%s"`).
4. Chuck, launches `ettercap` with the following command to activate the `remote_browser` plug-in:

   ```
   ettercap -T -M arp /IPaddr_Alice//80 /IPaddr_Bob// -P remote_browser
   ```

   Note: the configuration of the plug-in can be found in `/etc/ettercap/etter.conf`.
5. Now Bob can connect with a browser to the HTTP server running on Alice's machine:

   ```
   http:// IPaddr_Alice/
   ```

   Chuck can now verify the web page loaded by Bob in his browser.

How can Chuck intercept all the web traffic of Alice?

→

How can you defend against these types of sniffing attacks?

→

Try to execute a login by using SSH protocol and to capture the username and password. Are you able to do so (why) ?

→

3 Additional exercises

3.1 DNS spoofing

In this exercise, we will see how to apply spoofing technique against DNS.

> **NOTE**
>
> In a virtual environment or when the DNS server is in the same network of the attacker and the victim, these tests might not work because the success of DNS spoofing is based on the speed of the response. It depends on whether the attacker manages to send a response corresponding to a DNS request faster than the legitimate server. In virtual environments, we deal with virtual switches (that is software processes that simulate the behaviour of a real switch) which behave both as DHCP and DNS server, so the DNS server of the virtual switch responds faster than the attacker. So, in this case, it might not be possible to obtain a "quicker" response from the attacker.

As said, in this exercise we use `ettercap` for this attack. The attacker will behave as a man in the middle between the victim and the DNS server. Next, `ettercap` will be in charge of filtering the DNS requests originating from the victim and to respond before the DNS server will do so.

Exercise 9. Form a group of two VMs: Alice (the attacker) and Bob (the victim). Next proceed as follows:

1. Alice adds in the file `/etc/ettercap/etter.dns` the command

   ```
   www.polito.it A 130.192.39.120
   ```
2. Bob controls the content of the file `/etc/resolv.conf`

3. Now, try to understand and check out the functionality of the network under normal conditions: Bob tries to connect to a web site in Internet, for example to the web site www.polito.it. Check out directly the DNS query sent by using the command:

   ```
   host -v www.polito.it
   ```

   You should see information provided by DNS.

   →

4. Alice identifies the IP address of the gateway (you can find it with the command `route`). Next, she starts the DNS spoofing attack with the following command:

   ```
   ettercap -T -M arp:remote /IPaddr_host_Bob// /IPaddr_GW// -P dns_spoof
   ```

5. Bob tries to connect to the web site www.polito.it.

What does Bob see in his web browser?

→

What could have happened if the attack didn't work? Make your assumptions and confront them with what we have indicated in the initial note.

→

What does Alice see in the `ettercap` window?

→

Check out the configuration of the DNS plug-in, in the file `/etc/ettercap/etter.dns`.

How can an attacker obtain similar results (as the ones described in this section), even though the attacker is not close to the victim?

→

3.2 Vulnerability assessment with GVM

Let's try to identify the vulnerabilities of a "victim" node, by using the GVM (Greenbone Vulnerability Management).

GVM is the new name of a famous vulnerability scanner named OpenVAS, which, at its turn was derived from another vulnerability scanner, named Nessus, which is no longer available nowadays as an open-source software.

GVM has been initially the open source alternative to Nessus, with similar functionalities. Nowadays, GVM has a free version and a commercial one, which includes attacks against enterprise systems.

Plug-ins and definitions. To perform this exercise, you need all the plug-ins and the definitions required by GVM. Thus, you should download them. However, to allow you to save time and bandwidth for this download, we have inserted the last archive (available) in the Kali image (ISO file). Thus, you can install them starting from this archive, reducing thus considerably the time required to download them from Internet. The specific

commands you need to use this archive are given below. In this way, you may install these plug-ins from this local archive.

GVM is a sophisticated program, composed by a server to which several clients can connect to perform vulnerability analysis. An overview of its functionality and its architecture is presented at the link:

https://securitytrails.com/blog/openvas-vulnerability-scanner

To perform vulnerability scanning and to view the result it is possible to connect via a web browser to GVM web interface.

> **NOTE**
>
> If you created ISO Live virtual machines with less than 8 GB RAM some operations may fail because the RAMdisk on which Kali saves the temporary data for installation is not sufficiently big. We have tested the exercise with 8 GB and everything works correctly.

To correctly configure and update GVM, you need to execute several steps. It is possible to run them automatically by using a dedicated script, provided by GVM. If it is launched in default mode, this script uses `rsync` to update the plug-ins from the remote repository (in Internet).

> **NOTE**
>
> The initial configuration of GVM can take minutes due to the amount of data that has to be downloaded in the setup phase.

1. Run the following command to update GVM and to setup the work environment:

   ```
   gvm-setup
   ```

   Now, try to guess what the setup command is doing.

   ```
   →
   ```

   > **ATTENTION**
   >
   > Write down the password displayed on the screen at the end of the operation, because it will be necessary later for connecting to the GVM server.

   You can verify that everything is correctly configured with the command:

   ```
   gvm-check-setup
   ```

After installing GVM, you can execute now the exercise.

Exercise 10. Form groups of two hosts (named Alice and Bob). Let's assume Bob has installed GVM.

1. Alice comments the line `Require local` in the file `/etc/apache2/mods-available/status.conf` and uncomments `Require ip 192.0.2.0/24` (note: you need to check the address of the host and of the subnet in which the web server is deployed. If it is different from the one indicated above, you have to change the `Require ip` accordingly.)
2. Alice starts Apache
3. Bob proceeds as follow:

(a) Bob makes the login in GVM by connecting with his browser to `https://localhost:9392`, and by using the user admin that has been created before and the admin password that you have written before. Note that if the browser signals certificate problems, it is due to the fact the certificate is self-signed (so you have to accept the certificate).

(b) Creates a new target, by choosing `targets` from the menu "Configuration"), then clicking on the icon "New Target" (star placed upleft), and assigning the name "alice" (to the target). Next, he chooses as IP address of the target the IP address of Alice. You can define a port range, e.g. the one containing only the port on which it runs the Apache server of Alice (e.g. the port 80). Alternatively, if you can wait for a few minutes, you can scan the whole host.

(c) Creates a new task, by choosing `tasks` from the menu "Scans"), then clicking on the icon "New Task" (star placed in upleft position), and choosing as target the one that has been created previously (placed in upleft position)

(d) Selects the type of analysis "Full and Fast" and closes the window.

(e) Starts the scanning by clicking on the triangle close to the task that has been previously created.

Alternatively, Bob can prepare a scanning with the guided procedure, by pressing the Wizard (violet icon). The information to be inserted is basically the same, although it is provided in a slightly different order.

What is the result obtained by Bob in performing the vulnerability scanning ?

→

Have you identified any issue? If yes, of what type and how Alice can solve it ?

→

NOTE

Sometimes the solutions proposed by GVM are not updated or correctly formatted (e.g. spaces between the dash - and the options), or the solutions do not apply to the specific host configuration. We suggest to always read the official documentation for verification and avoid plain copy-and-paste from the GVM window. Most important, try always to understand the proposed solution, rather than performing a blind copy.