PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Skikda — Department of Computer Science
Master's in Computer Security — Semester 3 (M2) — UEF322
Web and Mobile Application Security
Updated on 2026-10-25 at 19:40:42
© 2026 Mr Laïdi FOUGHALI — www.al-moualime.com/wmas — CC BY-NC-SA 4.0 License

---

Lab 1 — Information Gathering and Port Scanning

Background

This lab extends section 1.3 of the course ("Intrusion phases"). Before exploiting any flaw at all, an attacker – or a legitimate pentester – always starts by **mapping the target**: which services are running, on which ports, with which versions, and what information is publicly available about it. This step does not "break" anything: it only observes what is already exposed. That is exactly why it is often neglected by defensive teams, even though it gives the attacker most of their roadmap. By practicing it yourself, you will understand why section 1.4 (vulnerabilities and CVEs) stresses the direct link between an *exposed software version* and an *exploitable attack surface*.

From the defender's point of view, this lab has a second benefit: **seeing your own system the way an attacker sees it**. Administrators use exactly the same tools to check that no service is exposed by mistake.

Objective

Reproduce, in an authorized environment, the first two phases of the *Cyber Kill Chain*: information gathering and scanning (section 1.3 of the course handbook). By the end of the lab, you will be able to:

- distinguish **passive** from **active** information gathering;
- explain what a port, a service and a port state are;
- run an Nmap scan, **understand every option** and **interpret every line** of the output;
- gather the information into a **mapping sheet** of the target;
- link a software version to known vulnerabilities (CVEs);
- name the main alternatives to Nmap and know when to use them.

Prerequisites

Have studied chapter 1: types of hackers, intrusion phases, the concept of vulnerability and CVE/CVSS. Basic TCP/IP notions (IP address, TCP protocol) are useful; part 1 below reviews them.

Tools and installation

On Linux (Ubuntu/Debian or Kali, already included in most lab VMs):

```
sudo apt update
sudo apt install -y nmap whois dnsutils curl theharvester
```

On Windows, install Nmap from `https://nmap.org/download.html` (the installer also provides **Zenmap**, its graphical interface); `whois` is not bundled, so use the online lookup at `https://www.whois.com` instead; for DNS, use `nslookup` or `Resolve-DnsName` (PowerShell); `curl.exe` ships with Windows 10 and 11; `theHarvester` requires Python 3 and installs with `pip install theHarvester`.

- `nmap` (port and service scanner)
- `whois` (domain registry lookup)
- `dig` / `nslookup` (DNS queries)
- `curl` (command-line HTTP requests)
- `theHarvester` (OSINT collection)
- A Web browser

Before you start — what you need to know about Nmap and similar tools

`nmap` is not a harmless tool: it is the tool an attacker uses during the information-gathering phase. The same precautions apply to every so-called "active" tool, which itself sends requests to the target: `masscan` and `zmap` (very fast port scanners), `nikto` (Web server scanner), `OpenVAS`/`Nessus` (vulnerability scanners), `sqlmap`, the active scanner of `Burp Suite` or `OWASP ZAP`, `hydra` (password attacks), etc.

1. **No authorization, no scan.** Only scan what you own or what its owner has authorized you to test **in writing**, with a precise scope (addresses, domains, dates, tools). In Algeria, attacks on automated data processing systems are punished by the Penal Code (articles 394 bis et seq.); law 09-04 of 5 August 2009 governs the fight against offences related to information technologies. "It was for a lab" or "I didn't break anything" are not excuses.
2. **A scan is not invisible.** Firewalls, IDS and server logs record your IP address. A scan is often the first sign of an attack for a security team, and it can lead to a complaint, a block or a report to your Internet service provider. The university network is **not** a target: do not scan it.
3. **A scan can disrupt the target.** An aggressive scan can slow down or crash a fragile device (old printer, IP camera, IoT device, industrial controller). Stick to the default timing (`-T3`); avoid `-T5` and tools such as `masscan`/`zmap`, designed to sweep the entire Internet.

4. **Understand every option before using it.** Read `man nmap` or `nmap -h`. Some options go well beyond observation: `-A` and `--script` run NSE scripts, whose `intrusive`, `brute`, `dos`, `exploit` and `vuln` categories can actually attack the target. In this lab, only use the commands given in the instructions.

5. **Passive or active?** `whois` and `theHarvester` query public sources (registries, search engines, certificate logs) without contacting the target: they are "passive", but respect the terms of use of the services they query. `nmap` and the tools listed above are "active": they touch the target directly and require authorization.

6. **Privileges and network.** Some scan types (`-sS`, `-O`) require administrator rights (`sudo`); without them, `nmap` falls back to a full-connection scan (`-sT`). The terms of use of many Internet service providers, cloud hosts and VPNs forbid scanning: check before scanning from their network.

7. **Where to practice legally.** Your own virtual machines ("host-only" network), deliberately vulnerable applications installed on your own machine (OWASP Juice Shop, DVWA, Metasploitable) and hosts published for this purpose, such as `scanme.nmap.org`, in compliance with their rules.

8. **Keep a record and stop when in doubt.** Write down every command you run, its date and its result (useful for the report and to prove you stayed within scope). If you target the wrong host by mistake, stop immediately (`Ctrl+C`) and inform the teacher.

Authorized target — why `scanme.nmap.org`?

`scanme.nmap.org` is a server made available **deliberately and publicly** by the creators of Nmap, specifically so that anyone can practice port scanning **without committing an offence**. Why this matters:

- **Legality**: scanning a server without authorization can be illegal, in Algeria as elsewhere, even without malicious intent. A simple port scan can be interpreted as an intrusion attempt.
- **Explicit authorization**: the Nmap authors have published the rules of use for this host (reasonable frequency, no aggressive scanning or denial of service). You are therefore covered as long as you respect them.
- **Teaching purpose**: this is why this lab uses it as a target; you practice the real commands without any legal risk.
- **Responsibility**: each student is **solely responsible** for the use they make of the tools and techniques covered in the labs. Any use outside the authorized scope of this lab is their sole responsibility, both civil and criminal; the teacher and the University of Skikda accept no liability for any misuse or illegal use.

It is one of only two targets for this lab, together with the course server described in section 3.4. Scanning any other domain or address without prior written authorization falls outside the legal scope of the exercise (chapter 1, ethical framework).

Part 1 — Concepts to understand before scanning

1.1 Passive and active information gathering

Passive information gathering

Gathering information **without ever contacting the target**: you query third-party public sources (domain registries, public DNS, search engines, TLS certificate logs, social networks). The target sees nothing.

Active information gathering

Gathering information **by sending packets to the target yourself** (ping, port scan, HTTP requests). It is more accurate, but the target can detect and log it: hence the need for authorization.

In a professional approach, you **always** start with passive information gathering (discreet, no legal risk) before moving to active information gathering, limited to the authorized scope.

1.2 IP address, port and service

A server is identified on the Internet by its **IP address** (for example `203.0.113.10`). Several network programs run on the same machine (Web server, SSH server, database…). To know which program a packet is meant for, a **port number** from 0 to 65 535 is used.

A simple picture: the IP address is the address of a building, the port is an apartment number, and the **service** is the person who lives there and answers when you knock.

Port	Usual service	Role
`21`	FTP	file transfer (unencrypted)
`22`	SSH	encrypted remote administration
`25`	SMTP	sending e-mail
`53`	DNS	name resolution
`80`	HTTP	unencrypted website
`443`	HTTPS	website encrypted with TLS
`3306`	MySQL	database
`3389`	RDP	Windows remote desktop

Be careful: these are only **conventions**. An administrator can run SSH on port 2222; that is why you do not stop at the port number and try to **identify the actual service** (Nmap's `-sV` option, part 3).

1.3 How a scanner knows a port is open

A TCP connection starts with a three-step "handshake" (*three-way handshake*):

1. the client sends a `SYN` packet ("I would like to connect");
2. if a service is listening, the server answers `SYN/ACK` ("all right"); otherwise it answers `RST` ("nobody here");
3. the client confirms with `ACK`: the connection is established.

A port scanner simply relies on this rule: it sends a `SYN` to each port and **observes the answer**. It exploits no flaw.

Nmap state	Meaning
`open`	the target answered `SYN/ACK`: a service is listening and accepts connections.
`closed`	the target answered `RST`: the machine is reachable, but no service listens on this port.
`filtered`	no answer (or an ICMP error message): a firewall is probably blocking the packet; Nmap cannot conclude.
`open|filtered`	Nmap cannot tell "open" from "filtered" (common with UDP).

1.4 Banner, version and attack surface

Many services introduce themselves when you connect: this is the **banner** (for example `SSH-2.0-OpenSSH_8.9p1 Ubuntu`). It reveals the software and its **exact version**, and therefore the possibility of looking up the known vulnerabilities of that version. All the reachable services, with their versions, make up the target's **attack surface**.

Key takeaways

- An open port is not a flaw: it is a **door**. The risk depends on the service behind it, its version and its configuration.
- The fewer services are exposed, the smaller the attack surface.
- A scanner only observes normal answers of the TCP protocol.

Part 2 — Passive information gathering

Open a text file `lab1-log.txt`: you will write down every command, its date and what you learn from it (see precaution no. 8).

1. **Identifying the domain owner.** Run:

   ```
   whois nmap.org
   ```

   Note the owning organization, the registrar, the creation and expiry dates and the DNS servers (*Name Server*) of the domain. `whois` queries the registered domain `nmap.org`, not the `scanme` subdomain, which has no entry of its own. Many fields may be hidden ("REDACTED FOR PRIVACY"): this is normal since the GDPR.

2. **Querying DNS.** DNS translates a name into an IP address. Run:

   ```
   dig +short scanme.nmap.org A
   dig +short scanme.nmap.org AAAA
   dig +short nmap.org NS
   dig +short nmap.org MX
   ```

   (on Windows: `nslookup scanme.nmap.org` or `Resolve-DnsName scanme.nmap.org`). The record types mean:

   - `A`: IPv4 address; `AAAA`: IPv6 address;
   - `NS`: servers that are authoritative for the domain;
   - `MX`: mail servers of the domain.

Note the IPv4 and IPv6 addresses of `scanme.nmap.org`: it is one of only two hosts you will scan in part 3. Do the same for `rdv.al-moualime.com` and `www.al-moualime.com`: what do you notice?

3. **Certificate logs.** Every TLS certificate issued by a public authority is recorded in a public log (*Certificate Transparency*). Open `https://crt.sh/?q=nmap.org` in your browser: you get the list of names (subdomains) for which a certificate has been issued. It is one of the most reliable sources for discovering subdomains without touching the target.

4. **Automated OSINT collection.** `theHarvester` queries several public sources at once:

```
theHarvester -d nmap.org -b crtsh,duckduckgo
```

`-d` sets the domain, `-b` the sources queried. Note the addresses and subdomains publicly associated with the parent domain, and compare them with what `crt.sh` gave you.

Discovered subdomains — observe without scanning

The subdomains found in steps 3 and 4 belong to servers you are **not** authorized to scan. Write them down in your mapping sheet, but do not run any active tool against them: only `scanme.nmap.org` and the course server (section 3.4) are authorized.

Part 3 — Active information gathering with Nmap

3.1 Reading an Nmap command

An Nmap command always has the same form: `nmap [options] target`. The options used in this lab:

Option	Effect
`-sn`	host discovery only ("does the machine answer?"), no port scan.
(none)	scans the **1,000 most common ports**.
`-p-`	scans all **65,535** ports (`-p 22,80`: only ports 22 and 80).
`-sV`	probes each open port to identify the **service and its version**.
`--reason`	shows **why** Nmap chose this state (`syn-ack`, `reset`, `no-response`).
`-oN file`	saves the output to a text file (for the report).
`-6`	uses the target's IPv6 address.
`-sS / -sT`	SYN scan (fast, requires `sudo`) / full-connection scan (no administrator rights). Nmap picks one automatically based on your rights.

3.2 Reading Nmap output — annotated example

Here is the output of a scan of a **fictitious** machine `203.0.113.10` (an address reserved for documentation). It is only meant to teach you how to read; your results on `scanme.nmap.org` will be different.

```
Nmap scan report for 203.0.113.10
Host is up (0.045s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE    SERVICE  VERSION
22/tcp   open     ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol
    2.0)
80/tcp   open     http     nginx 1.18.0 (Ubuntu)
443/tcp  open     ssl/http nginx 1.18.0 (Ubuntu)
3306/tcp filtered mysql
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Line by line:

- `Host is up (0.045s latency)`: the machine answers, in 45 milliseconds.
- `Not shown: 996 closed tcp ports (reset)`: 996 ports answered `RST`; Nmap hides them to keep the output readable.
- `22/tcp open ssh OpenSSH 8.9p1...`: an SSH server is listening; `-sV` identified the software (OpenSSH), its version (8.9p1) and the distribution (Ubuntu). This precise version is what you then look up in CVE databases.
- `80/tcp` and `443/tcp`: the same nginx Web server, in clear text (HTTP) and encrypted (`ssl/http`).
- `3306/tcp filtered mysql`: no answer; a firewall probably protects the database. "mysql" is only a **guess** based on the port number, not an identification.
- `Service Info`: information deduced from the banners (here, a Linux system).

Summary of the fictitious mapping: 3 exposed services, a database protected by filtering, precise versions to check.

3.3 Your turn: scanning `scanme.nmap.org`

1. **Does the target answer?**

   ```
   nmap -sn scanme.nmap.org
   ```

   Check that the address shown is the one you found with `dig` in part 2.

2. **Quick scan of common ports.**

   ```
   nmap --reason scanme.nmap.org
   ```

   How many open ports? What do the reasons in the `REASON` column mean?

3. **Full mapping of ports and services.**

   ```
   nmap -sV -p- -oN scan-scanme.txt scanme.nmap.org
   ```

   This command sweeps all 65,535 ports (`-p-`), tries to identify the version of the service behind each open port (`-sV`) and saves the output (`-oN`). It may take several minutes: this is normal, do not rerun it over and over. Did the quick scan miss any ports? Why?

4. **IPv4 / IPv6 comparison.**

```
nmap -6 scanme.nmap.org
```

(only if your connection has IPv6; otherwise, simply note it). Does the same server expose the same ports over both protocols? Why might an administrator forget to protect IPv6?

5. **What the Web server shows.** If a Web port is open, look at the HTTP headers it returns:

```
curl -I http://scanme.nmap.org/
```

(on Windows: `curl.exe -I http://scanme.nmap.org/`). Does the `Server` header confirm the version found by Nmap? Also open the page in your browser and read it: it restates the host's rules of use.

3.4 Second target: your course's server

Teacher's authorization — precise scope

The names `rdv.al-moualime.com` and `www.al-moualime.com` belong to your teacher, Mr Laïdi FOUGHALI, who **authorizes you** to map them as part of this lab, **during the lab session only**, with only the commands below. Everything else (full `-p-` scan, NSE scripts, `-T4`/`-T5` timings, password attempts, other subdomains, other active tools) is **forbidden**. This is a production server: it hosts the course website and an appointment application.

1. **Two names, one machine?** Compare the addresses obtained in part 2 for `rdv.al-moualime.com` and `www.al-moualime.com`. If they are identical, both names point to the **same server**: a single port scan is enough, do not run it twice.

2. **Ports and services** (1,000 common ports, no `-p-`):

```
nmap -sV --reason -oN scan-course.txt www.al-moualime.com
```

Compare with `scanme.nmap.org`: which services are common? Which ones appear on only one of the two servers?

3. **One server, several sites.** Query each name separately:

```
curl -I https://www.al-moualime.com/
curl -I https://rdv.al-moualime.com/
```

Are the two answers identical? A single Web server can host several sites, called **virtual hosts**: it chooses which site to serve from the name requested by the client (HTTP `Host` header). Why can a port scan alone not reveal all the sites hosted on a machine?

4. **Security headers.** In the previous answers, note whether the `Strict-Transport-Security`, `Content-Security-Policy`, `X-Content-Type-Options` and `X-Frame-Options` headers are present or absent. Does the `Server` header show a version number? (These headers are studied in chapters 4 and 5.)

5. **TLS certificate.** Open `https://crt.sh/?q=al-moualime.com`: which names appear in the domain's certificates? Write them down without scanning them.

With a graphical interface — Zenmap

Zenmap, installed with Nmap on Windows, runs exactly the same commands. Enter the target and the command in the fields provided: the "Ports / Hosts" and "Topology" tabs present the same result as a table and a diagram. Handy for beginners, but the command line remains essential for automating and documenting.

Part 4 — From versions to vulnerabilities

For each service identified in part 3, check whether a published CVE matches its exact version, on `https://www.cve.org`, on the NIST database `https://nvd.nist.gov` or with the `searchsploit` tool locally. Note the CVSS score of each CVE found (see chapter 1, section 1.4).

Two reading precautions:

- **Distributions patch without changing the version.** Ubuntu or Debian often apply security fixes while keeping the original version number (the suffix, for example `3ubuntu0.6`, shows the patch level). A CVE listed for "OpenSSH 8.9" is therefore not necessarily exploitable: it is a **lead to check**, not proof.
- **Do not test any CVE.** This lab stops at identification: no exploitation, no vulnerability script against the target.

Part 5 — The mapping sheet

Gather your results in a structured sheet **per target** (`scanme.nmap.org` and the course server), following the template below (to copy and fill in your report):

Item	**To fill in**
Domain and owner	organization, registrar, dates (part 2, step 1)
Addresses	IPv4, IPv6 (part 2, step 2)
DNS and mail servers	NS and MX records
Public subdomains	sources: `crt.sh`, `theHarvester`
Ports and services	port, state, service, software and version (part 3)
Virtual hosts	names hosted on the same address, differences between the answers (section 3.4)
Security headers	present / absent (section 3.4)
IPv4 / IPv6 differences	ports present in only one of the two
Web information	`Server` header, visible technologies
Related CVEs	identifier, CVSS score, "to check" or "patched"

Part 6 — Alternatives to Nmap

Nmap remains the reference: it is the most complete and best documented tool. Other tools are nevertheless widely used by professionals, each for a specific need. For this lab, it is enough to **know their role**; they must only be used on authorized targets.

Tool	Category	Strengths	Limits
Zenmap	Nmap's graphical interface	visualization, ready-made scan profiles, comparison of two scans	same capabilities as Nmap, less convenient to automate
RustScan	fast port discovery	finds open ports very quickly, then hands the list to Nmap for identification	does not identify services itself
Naabu (Project-Discovery)	port discovery	simple, chains with the other tools of the same suite (httpx, nuclei)	fewer options and less detection than Nmap
masscan, ZMap	very large-scale scanners	designed for Internet-wide studies by researchers	dangerous outside that context: can saturate a network; unsuited to a single target
Netcat (`nc`)	multi-purpose network tool	manually test a single port, read a banner	no automation, no version detection
Angry IP Scanner	simple graphical scanner	cross-platform, very easy to use	basic results
WhatWeb, Wappalyzer, httpx	Web fingerprinting	recognize a site's server, language, CMS and libraries	limited to the Web layer
Shodan, Censys	device search engines	show ports and banners already collected, **without scanning the target** (passive)	data sometimes outdated; accounts and usage limits
Amass, Subfinder	subdomain discovery	aggregate many public sources	results to verify; some sources need an API key

Optional exercise — passive information gathering with Shodan

Open `https://www.shodan.io/host/` followed by the IPv4 address of `scanme.nmap.org`. Compare the ports and banners shown with your own scan. What advantages and drawbacks do you see in this passive approach?

Comprehension questions

1. Which phase(s) of the Cyber Kill Chain (chapter 1, section 1.3) do parts 2 and 3 of this lab respectively correspond to?
2. Explain the difference between a `closed` port and a `filtered` port. Which of the two gives an attacker more information?
3. An open port is not a vulnerability in itself. What, in the information collected, turns an exposed service into a concrete risk?
4. Why should you be wary of a CVE found only from a version number?
5. Why do automated scanners (Nessus, OpenVAS, chapter 1 section 1.4) always perform an scanning step before testing exploits?

6. Compare the two servers: which one exposes the smaller attack surface? Justify using your two sheets.
7. As the administrator of `scanme.nmap.org`, what measures would you take to reduce what an attacker can learn during this phase?

Deliverable

A report of two pages at most, containing:

- the completed mapping sheet (part 5);
- the `scan-scanme.txt` and `scan-course.txt` files as an appendix;
- for each service, a risk level assessment (low/medium/high), justified by the CVEs found in part 4;
- the answers to the comprehension questions;
- your command log (`lab1-log.txt`).