Reconnaissance and Scanning Lab

6COSC019W - Cyber Security

Dr. Ayman El Hajjar

Two weeks labs: Week 3 and 4

STOP and READ

Before you start your work in the lab you need to setup the lab environment, if you are using a university machine.

Below is a summary of the steps you need to do. However you should read the Lab Environment Setup if you are stuck on any of those steps.

1. Delete all Virtual machines that already exist inside VirtualBox to ensure that you are starting your labs from a clean state.
2. Browse to the C:\:VirtualMachines folder.
3. You should have two files .7z, one for Kali and another for OWASP.
4. It is safe to delete folders but NOT the 7z files.
5. Extract the files by right clicking on each and select 7-zip, then selecting **Extract here**.
6. Add the VMs to VirtualBox.
7. Double check if the Network settings for the Lab environment are set correctly.

- In this lab, **my OWASP machine IP address is 192.168.56.111**. Instead- Use this IP address as a reference but **substitute my IP address with your OWASP IP address** .

Figure 1: **"The quieter you become, the more you are able to hear."**

Lab objectives

This activity assumes that you now are familiar with how to setup the lab environment and you have completed the **Introduction to Linux Lab.**

After completion of this you should be able to:

* Understand why a hacker or an ethical hacker would conduct Reconnaissance and Information gathering activities.
– Understand Passive and Active reconnaissance
* Understand Scanning and Enumeration
* Complete the Reconnaissance and Scanning stages on the OWASP machine
* Understand why a hacker or an ethical hacker would conduct Reconnaissance and Information gathering activities.

Getting to know web applications on a vulnerable VM

OWASP-BWA (Broken Web Applications) contains many web applications, intentionally made vulnerable to the most common attacks. Some of them are focused on the practice of some specific technique while others try to replicate real-world applications that happen to have vulnerabilities.

1. With OWASP VM running, open your Kali Linux host's web browser and navigate

to the OWASP IP address. In my lab environment, OWASP is allocated the IP http://192.168.56.111. Check the IP address of your OWASP machine.

2. Once you are there, You will see a list of all applications the server contains.

3. For **Damn vulnerable Web Application**. Use username **admin** and password **admin**. We can see a menu on the left; this menu contains links to all the vulnerabilities that we can practice in this application: Brute Force, Command Execution, SQL Injection, and so on.

4. **OWASP WebGoat.NET**. This is a .NET application where we will be able to practice file and code injection attacks, cross-site scripting, and encryption vulnerabilities. It also has a WebGoat Coins Customer Portal that simulates a shopping application and can be used to practice not only the exploitation of vulnerabilities but also their identification.

The applications in the home page are organized in the following six groups:

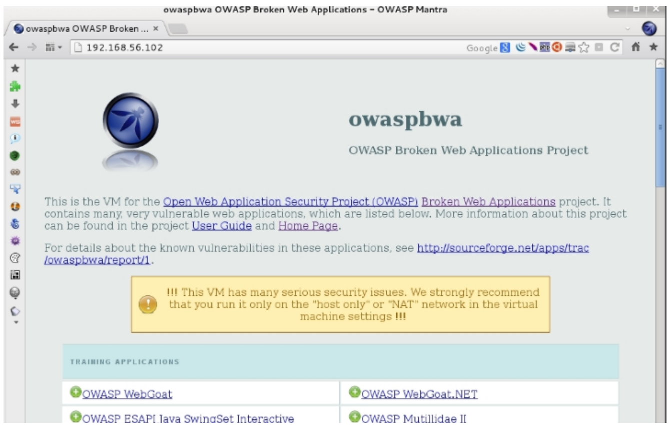
- **Training applications:** These are the ones that have sections dedicated to practice-specific vulnerabilities or attack techniques; some of them include tutorials, explanations, or other kind of guidance.

- **Realistic, intentionally vulnerable applications:** Applications that act as real-world applications (stores, blogs, and social networks) and are intentionally left vulnerable by their developers for the sake of training.

- **Old (vulnerable) versions of real applications:** Old versions of real applications, such as WordPress and Joomla are known to have exploitable vulnerabilities; these are useful to test our vulnerability identification skills.

- **Applications for testing tools:** The applications in this group can be used as a benchmark for automated vulnerability scanners.

- **Demonstration pages / small applications:** These are small applications that have only one or a few vulnerabilities, for demonstration purposes only. OWASP demonstration application: OWASP AppSensor is an interesting application, it simulates a social network and could have some vulnerabilities in it. But it will log any attack attempts, which is useful when trying to learn; for example, how to bypass some security devices such as a web application firewall.


owaspbwa OWASP Broken Web Applications - OWASP Mantra
owaspbwa OWASP Broken ...
192.168.56.102
Google

owaspbwa

OWASP Broken Web Applications Project

This is the VM for the Open Web Application Security Project (OWASP) Broken Web Applications project. It contains many, very vulnerable web applications, which are listed below. More information about this project can be found in the project User Guide and Home Page.

For details about the known vulnerabilities in these applications, see http://sourceforge.net/apps/trac/owaspbwa/report/1.

!!! This VM has many serious security issues. We strongly recommend that you run it only on the "host only" or "NAT" network in the virtual machine settings !!!

TRAINING APPLICATIONS

OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II

Figure 2: OWASP Web Interface

1 Reconnaissance: Open Source Intelligence (OSINT)

PLEASE READ- Requirements and Notes

* For this section of the lab, you need to make sure that your Kali Linux machine is connected to the Internet (NAT).

* For any other sections, if you need to be connected to the Internet, to install an application or for any other reasons, this will be made clear to you.

* OSINT can be carried out by conducting different activities on the internet to search for information. Tools are there to put all of those activities in one place and to automate them.

* There are many tools to use to conduct your OSINT. We will be using some, but feel free to explore others.

* **You are conducting this assessment on Live websites.** This is called passive reconnaissance.

* You can see what type of sources can be used in OSINT investigation using the OSINT Framework

* **cwscenario.site** is a domain we use for OSINT activities. In your assessment, when you are conducting the OSINT investigation **ONLY**, you are assuming this is your public website.

1.1 DNS reconnaissance

In this section, we will gather information about the DNS record for the DNS public records.

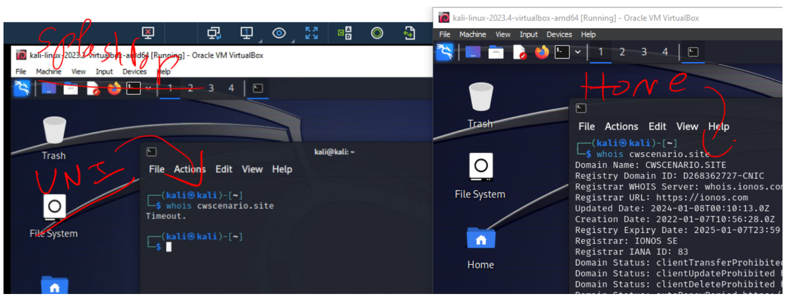
1.1.1 WHOIS

- WHOIS, which stands for "Who is," is a protocol used to query databases that store information about domain registrations and IP address allocations.
- It provides valuable details about the owner, registrar, and contact information for a given domain or IP address.
- Ethical hackers often leverage WHOIS information during reconnaissance to gather intelligence on potential targets.
- Analysing WHOIS data helps ethical and malicious hackers understand the digital footprint of a target, aiding in the early stages of information gathering and vulnerability assessment.
- You can use WHOIS in two different ways:
  - Via the online WHOIS databases site https://who.is/
  - Using the command **WHOIS**- For example we can use to check cwscenario or the university domain

    whois cwscenario.site

    - For some reasons from inside the university many DNS enumeration and reconnaissance commands are being flagged as malicious and are not being completed (Timeout) See figure 3. However you can conduct the Whois on the website. you can use it however on your device at home. The next command is working as normal from within the university.

    whois westminster.ac.uk



Splashtop
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Trash
UNI
File System
kali@kali: ~
File Actions Edit View Help
kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Home
Trash
File System
Home
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ whois cwscenario.site
Timeout.

┌──(kali㉿kali)-[~]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ whois cwscenario.site
Domain Name: CWSCENARIO.SITE
Registry Domain ID: D268362727-CNIC
Registrar WHOIS Server: whois.ionos.co
Registrar URL: https://ionos.com
Updated Date: 2024-01-08T00:10:13.0Z
Creation Date: 2022-01-07T10:56:28.0Z
Registry Expiry Date: 2025-01-07T23:59
Registrar: IONOS SE
Registrar IANA ID: 83
Domain Status: clientTransferProhibite
Domain Status: clientUpdateProhibited
Domain Status: clientDeleteProhibited
```

Figure 3: who.is response for cwscenario

1.1.2 DNS Enumeration

- Now we will dig deeper and try to extract more information about the DNS records. This activity is considered a DNS enumeration technique. DNS enumeration is the process of locating all DNS servers and DNS entries for an organization. DNS enumeration will allow us to gather critical information about the organization such as usernames, computer names, IP addresses, and so on.

- **Note:** Although the commands below are still OSINT commands and are exploiting information that are already public, you should not conduct them on any website without permission.

- **dnsenum** is a tool that is pre installed on Kali Linux.

- Kali has three tools pre installed. Using the Kali Linux menu, you can find them in the Information gathering folder, DNS analysis section. The tools are **dnsenum**, **dnsrecon** and **fierce**

  1. Using **dnsenum**

     dnsenum --enum scanme.nmap.org

     dnsenum --enum cwscenario.site

  2. Using **dnsrecon**

     dnsrecon -d scanme.nmap.org

     dnsrecon -d cwscenario.site

  3. Using **fierce**

     fierce --domain scanme.nmap.org

     fierce --domain cwscenario.site

- We should get an output with information like host, name server(s), mail server(s), and if we are lucky, a zone transfer.

- How do the results obtained from the three tools compare?

1.2 TheHarvester

- theHarvester is an Information gathering tool that i used to conduct passive reconnaissance on domain names.

- Since theHarvester searches for information already are online, it means your search will go unnoticed and will not raise any alarm.

- If you type **theHarvester -h** on an open terminal, you will see a collection of different options you can use.

- We will focus on **-d** option to specify the domain, **-b** option to specify the data source (google, linkedin, bing, twitter, etc..) . We also need to use **-l** option for certain sources as they block your request if the number of requests is not set (**Note this is an L in small letters**) .
- Lets see some examples:

  theHarvester -d www.westminster.ac.uk -b all -l 100

  theHarvester -d westminster.ac.uk -b bing -l 500

- you can also specify all sources

  theHarvester -d cwscenario.site -b all

- You can use the **-f** option to specify the filename to save the output to. I will save them in an html format so that I can open them in a browser.

  theHarvester -d cwscenario.site -b all -f myresults.html

- When using theHarvester, you might have noticed that it was returning some information about the ASN records.

1.3 ASN

- ASN stands for Autonomous System Number. It is basically a method to tell us how data travels between routers on the Internet to reach the Server where the site resides.
- It is a good idea to dig deeper in information related to the Domain name Service as many times , you end up with information related to the organisation owning the domain. Sometimes you might also find a range of IP addresses that this organisation uses.
- Let's see how we can make use of ASN we found.
- **For security reasons, the university network blocks all requests for ASN records, you can do the activity below when you are at home. I will explain how it works, and if you cannot do it, you can skip it and move to section 1.2**
- I am going to make use of a database called **radb.net** This database supports requests using the whois protocol.

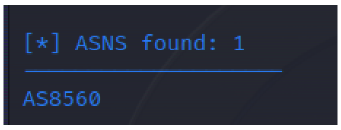
  whois -h whois.radb.net AS8560

  * Now find the ASN numbers for westminster.ac.uk and try to request information about those ASNs from the radb.net database.

- This lookup for using the AS number has given us some information regarding the hosting company.

- In fact, the **whois** command is quite powerful. We can use it in many different ways and it can give many information.
- for example, we can use it to look for routing data.

  whois -T route 185.117.153.79



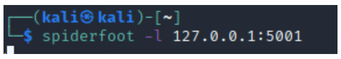
```
[*] ASNS found: 1
AS8560
```

Figure 4: the AS number for cwscenario.site

1.4 SpiderFoot

- Another OSINT you should explore is SpiderFoot.
- SpiderFoot is pre-installed on Kali.
- To start you need to open a terminal and run it. You need to specify the server address and the port number to use for its application.

  spiderfoot -l 127.0.0.1:5001

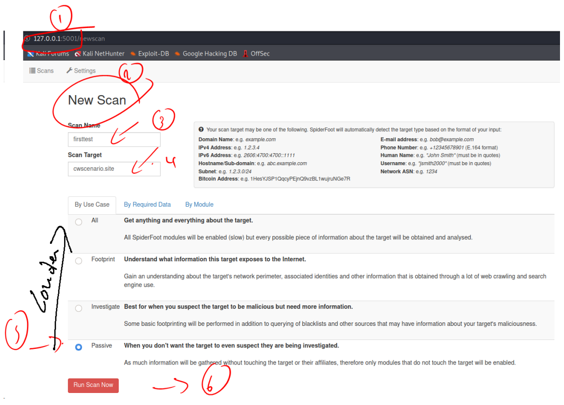


```
┌──(kali㉿kali)-[~]
└─$ spiderfoot -l 127.0.0.1:5001
```

Figure 5: Starting spiderfoot on localhost port 5001

- While Spiderfoot is running on the terminal, open a new browser and point it to 127.0.0.1:5001 (you can specify any other port number you want. I suggest you use a number larger than 5000)
- Spiderfoot is module based, some of the modules requires the use of API keys. In some cases, these modules will not function without properly setting up the API keys first. In other modules, the API keys are only to speed up the scanning. Check this website for more information on the modules. Ignore the installation part.
- At the beginning, and since we are conducting an OSINT investigation, at least at this stage.
- Click on "New Scan"
- Choose "Passive scan" and give a name for the scan and for IP address you can choose either an IP address or a domain name to investigate.
- Give it some time until it finishes running. Results will come gradually until the scan finishes.

- Once it is done, explore the obtained results.



127.0.0.1:5001/newscan

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Scans Settings

New Scan

Scan Name: firsttest

Scan Target: cwscenario.site

Your scan target may be one of the following. SpiderFoot will automatically detect the target type based on the format of your input:

Domain Name: e.g. *example.com*
IPv4 Address: e.g. *1.2.3.4*
IPv6 Address: e.g. *2606:4700:4700::1111*
Hostname/Sub-domain: e.g. *abc.example.com*
Subnet: e.g. *1.2.3.0/24*
Bitcoin Address: e.g. 1HesYJSP1QqcyPEjnQ9vzBL1wujruNGe7R
E-mail address: e.g. *bob@example.com*
Phone Number: e.g. *+12345678901* (E.164 format)
Human Name: e.g. *"John Smith"* (must be in quotes)
Username: e.g. *"jsmith2000"* (must be in quotes)
Network ASN: e.g. *1234*

By Use Case | By Required Data | By Module

○ All — **Get anything and everything about the target.**
All SpiderFoot modules will be enabled (slow) but every possible piece of information about the target will be obtained and analysed.

○ Footprint — **Understand what information this target exposes to the Internet.**
Gain an understanding about the target's network perimeter, associated identities and other information that is obtained through a lot of web crawling and search engine use.

○ Investigate — **Best for when you suspect the target to be malicious but need more information.**
Some basic footprinting will be performed in addition to querying of blacklists and other sources that may have information about your target's maliciousness.

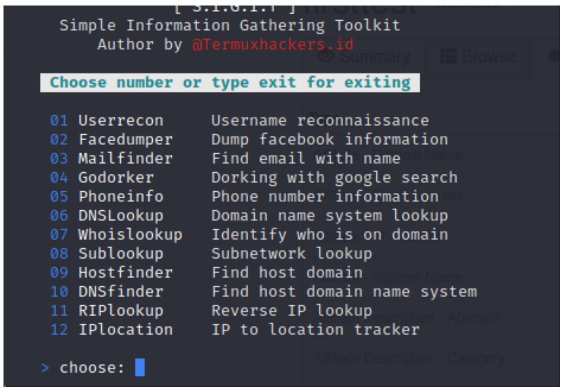
● Passive — **When you don't want the target to even suspect they are being investigated.**
As much information will be gathered without touching the target or their affiliates, therefore only modules that do not touch the target will be enabled.

Run Scan Now

Figure 6: Creating new scan on my testing domain name. Feel free to use this domain for testing.

1.5 SIGIT - Simple Information Gathering Toolkit

- There are many tools to conduct your Open Source Intelligence. A simple search on Github will brings hundreds of tools, however most will do very similar jobs, the difference is usually in the output and how it is presented.
- One tool that I personally find useful, is SIGIT- Simple Information Gathering Toolkit shown in Fig.7.
- As you can see in Fig.7 SIGIT has several features, such as user reconnaissance, mail finder, dnslookup and others.



```
      Simple Information Gathering Toolkit
          Author by @Termuxhackers.id

    Choose number or type exit for exiting

    01 Userrecon      Username reconnaissance
    02 Facedumper     Dump facebook information
    03 Mailfinder     Find email with name
    04 Godorker       Dorking with google search
    05 Phoneinfo      Phone number information
    06 DNSLookup      Domain name system lookup
    07 Whoislookup    Identify who is on domain
    08 Sublookup      Subnetwork lookup
    09 Hostfinder     Find host domain
    10 DNSfinder      Find host domain name system
    11 RIPlookup      Reverse IP lookup
    12 IPlocation     IP to location tracker

  > choose:
```

Figure 7: SIGIT - Simple Information Gathering Toolkit

- Let us now download SIGIT and install it at the same time.

wget https://raw.githubusercontent.com/termuxhackers-id/SIGIT/main/installkali.sh && sudo bash installkali.sh

- Let's run it now and see what information we can obtain. To run it open a terminal and type:

  sigit

- SIGIT brings an interface with a number of options you can use.
- Choose the one you need and put its number.
- Try for example the username you usually use for many services online and see if it can find it.
- You can also try to find a specific email with specific names and many other information.

2 Active Reconnaissance

Your wordlists

- Ask any good Pen tester, what is the most important toolkit in your arsenal and he/she will say my **wordlist/s**.
- Before you start this section, we are going to create two files in the home folder, one for users called users.txt and the other for passwords called pass.txt .

  cd ~

  touch users.txt

  touch pass.txt

- Now each time you see a word that you feel might be used as a username or a password, you add it to the file. For example:

  echo 'root' » users.txt – this will add the word **root** to users.txt

  echo 'owaspbwa' » pass.txt – this will add the word **owaspbwa** to pass.txt

2.1 Information gathering from simply looking at code

- Looking into a web page's source code allows us to understand some of the programming logic, detect the obvious vulnerabilities, and also have a reference when testing, as we will be able to compare the code before and after a test and use that comparison to modify our next attempt.

- Let's go to the OWASP server home page. On the browser, browse to the homepage of the server (192.168.56.111)
  - Right click anywhere on the page and select "View page source"
  - You will find that for each application, the username and password are hardcoded
- **By simply looking at the code of the website, you can learn a lot about it.**
- **Update both wordlists you created before with usernames and passwords you found!**

  To add to the users file: **echo 'theusername' » users.txt**

  To add to the passwords file: **echo 'thepassword' » users.txt**

2.2 Taking advantage of robots.txt

- One step further into reconnaissance, we need to figure out if there is any page or directory in the site that is not linked to what is shown to the common user. For example, a login page to the intranet or to the **content management systems (CMS)** administration. Finding a site similar to this will expand our testing surface considerably and can give us some important clues about the application and its infrastructure.
- We will use the **robots.txt** file to discover some files and directories that may not be linked to anywhere in the main application.
- Browse to http://192.168.56.111/vicnum/
- Now we add robots.txt to the URL.
- This file tells search engines that the indexing of the directories jotto and cgi-bin is not allowed for every browser (user agent). However, this doesn't mean that we cannot browse them.
- Let's browse to http://192.168.56.111/vicnum/cgi-bin/
- We can click and navigate directly to any of the Perl scripts in this directory.
- Let's browse to http://192.168.56.111/vicnum/jotto/
- Click on the file named jotto. Jotto is a game about guessing five-character words; could this be the list of possible answers? Check it by playing the game; if it is, we have already hacked the game!

How these tools works

robots.txt is a file used by web servers to tell search engines about the directories or files that they should index and what they are not allowed to look into. Taking the perspective of an attacker, this tells us if there is a directory in the server that is accessible but hidden to the public using what is called "security through obscurity" (that is, assuming that users won't discover the existence of something, if they are not told about it).

2.2.1 Finding Files and folders

DirBuster is a tool created to discover, by brute force, the existing files and directories in a web server. We will use it in this activity to search for a specific list of files and directories. We will use a text file that contains the list of words that we will ask DirBuster to look for.

- Create a text file and call it folderbuster.txt.
  - **touch folderbuster.txt**
- Now edit the file
  - you can use the terminal based editor **nano** or the graphical based editor **mousepad**.
    - **nano folderbuster.txt** or **mousepad folderbuster.txt**
- Now add the following keywords:
  - info
  - server-status
  - server-info
  - cgi-bin
  - robots.txt
  - phpmyadmin
  - admin
  - login
- save the file
- Now open a terminal and type **dirbuster**
  - This will open dirbuster in graphical mode.
  - On the DirBuster's window, set the target URL to **http://192.168.56.111:80**
  - Set the number of threads to 20.

- Select **List based brute force** and click on **Browse**.
- In the browsing window, select the file we just created (**folderbuster.txt**).
- Uncheck the **Be Recursive** option.
- Click on **Start**.
- If we go to the **Results** tab, we will see that DirBuster has found at least two of the files in our dictionary: **cgi-bin** and **phpmyadmin**.

- Now open a terminal and type **dirbuster**
  - This will open dirbuster in graphical mode.

How this tool works

- You can also use dirb tool. It works in a similar way to DirBuster but has no GUI.
  - **dirb http://192.168.56.111**
- Have a look at the manual, by typing **dirb -h**
  - For example I found that I can use the **-o** option to have the output stored in a file.
    - **dirb http://192.168.56.111 -o ~/Desktop/owaspfolders**
- Note: **The tools we have used are not comprehensive and there are many others either pre - installed with Kali or available as open source. You are encouraged to explore those tools.**

3 Scanning and Enumeration

One of the most important stages of an attack is information gathering. To be able to launch an attack, we need to gather basic information about our target. So, the more information we get, the higher the probability of a successful attack.

PLEASE READ BEFORE YOU START THIS SECTION

- In this section, both the OWASP VM and Kali VM should be running.
- Both VMs should be connected to the Host-Only adapter **unless** the command you are using is querying a live website domain name.
- You should take note for the OWASP machine IP address in your lab environment. To find your IP address, you can use the **ifconfig** command. **In my lab environment the IP address for OWASP is 192.168.56.111**
- Both VMs should be set on host only adapter. You can check the IP of Kali using **ifconfig**. It should be allocated an IP address on the same network **192.168.56.XXX**
- We will be mainly using nmap however there are many other tools we can be used. For other tools pre-installed on Kali about OSINT, Information Gathering, and tools to enumerate networks ports and addresses, services and protocols, you click on Kali home button, and select → **01 - Information Gathering**.

3.1 Setup DNS server

- Before we start with the Information Gathering activity, we need to setup the DNS server, otherwise nmap will keep on complaining that it cannot find the DNS server for our host only connection.
- Type:

```
sudo nano /etc/resolv.conf
```

- add

```
nameserver 6.7.8.9
```

3.2 Information gathering with Nmap

Nmap is probably the most used port scanner in the world. It can be used to identify live hosts, scan TCP and UDP open ports, detect firewalls, get versions of services running in remote hosts, and even, with the use of scripts, find and exploit vulnerabilities.

- We first need to see if the server is answering to a ping or if the host is up. We will use nmap with the **-sn** parameter. We do this to instruct Nmap to only check if the server was responding to the ICMP requests (or pings).
- According to the nmap help page, the **sn flag** disable ports scan while pinging the IP address. This is because the
- Is the server Responding?
  - nmap -sn 192.168.56.111
- Our server responded that it is up. If you don't get this, make sure you have OWASP is up and is on the host only network.
- According to the nmap help page, the **sn flag** disable ports scan while pinging the IP address.
- Now, we need to scan our IPs further, and see what ports are open on the OWASP server and what services are running on it.
- In the first command, with the **-sn** parameter, we instructed Nmap to only check if the server was responding to the ICMP requests (or pings). According to the nmap help page, the sn flag disable ports scan while pinging the IP address.
- Can you find any ports open on the server?

  nmap 192.168.56.111
  - The command below will return the same results

    sudo nmap -sS 192.168.56.111
- Rather than disabling port scans using the **sn** flag as we previously did, we can conduct an ARP host discovery on a network without scanning ports using the **PR** flag. We will be simply sending an ARP broadcast request to all hosts and see who responds.
  - For one specific host:

    nmap -PR 192.168.56.111
  - Or even a whole local network on a subnet looking for hosts who respond.

    nmap -PR 192.168.56.0/24

3.2.1 Identifying active machines with Nmap

- Before attempting a target to conduct penetration testing on, we first need to identify the active machines that are on the target network range.
- A simple way would be by performing a **ping** on the target network. Of course, this can be rejected or known by a host, and we don't want that.

- Let's begin the process of locating active machines by opening a terminal window.
  - Using Nmap we can find if a host is up or not

    nmap -sP 192.168.56.111
  - You can also use Nmap to find a collection of hosts in the same network if they are active or no.

    nmap -sP 192.168.56.*

3.2.2 Finding open ports

- With the knowledge of the victim's network range and the active machines, we'll proceed with the port scanning process to retrieve the open TCP and UDP ports and access points.
  - To find open ports on a single IP

    nmap 192.168.56.111
  - We can also explicitly specify the ports to scan (in this case, we are specifying only the first 1000 ports)

    nmap -p 1-1000 192.168.56.111
- We can also explicitly specify which port to scan on:
  - a specific host

    nmap -p 22 192.168.56.111
  - the whole network

    nmap -p 22 192.168.56.*
- You can output the result to a specific file in a specified format

  nmap -p 22 192.168.56.* -oG ∼/Desktop/myresults.txt
  - Note:The symbol ∼ means the home directory. This is the same as navigating to /home/kali. Kali is the username in this case.

3.3 Other scans

3.3.1 UDP scan

Up until this point, all of our scans have been for TCP ports. Some services and ports use UDP to communicate to the outside world. Our previous scan types (-sS and -sT) will not find UDP ports as they are only looking for TCP ports. Some services only run on UDP, such NTP (port 123) and SNMP (port 161). To find these ports and services, we need to do a UDP scan. We can do this with the -sU switch:

- To find these ports and services, we need to do a UDP scan. We can do this with the -sU switch

sudo nmap -sU 192.168.56.111

* **Note**- A UDP scan will take very long time and longer than TCP scans. This is because UDP is a connectionless protocol. It is advisable not to conduct it when you have limited time (such as in class)
* Scanning all UDP ports will take longer and is resource intensive due to the lack of the three way handshake so all will go through and no need for acknowledgment regardless whether it is used or not.

3.3.2 Reason

- Note in the output from the UDP scan above that some ports are reported as open/filtered. This indicates that nmap cannot determine whether the port is open or it is filtered by a device such as a firewall.
- Unlike TCP ports that respond with a RST packet when they are closed, UDP ports respond with an ICMP packet when they are closed. This can make scans far less reliable, as often the ICMP response is blocked or dropped by intermediate devices (firewalls or routers).
- Nmap has a switch that will return the reason why it has placed a particular port in a particular state. For instance, we can run the same UDP scan as above with the --reason switch and nmap will return the same results, but this time will give us the reason it has determined the particular state of the port.

  sudo nmap -sU --reason 192.168.56.111

3.3.3 List of targets

- Many times we want to scan a list of IP addresses and not an entire subnet. We can use any text editor and create a list of IP addresses and "feed" it to nmap. This is the list of IP addresses I want to scan.
  - **mousepad scanlist.txt**
  - **Inside this file put several IP**. For example you can put your host IP, the OWASP IP and the KALI Linux IP. Each on it's own line.
  - To scan all IP addresses in the list type:

    nmap -iL scanlist.txt

    * Note: You need to be in the same location as your scanlist file for the command to run.

3.3.4 Spoof or Decoy Scan!

- When we are scanning machines that are not ours, we often want to hide our IP (our identity).

- Obviously, every packet must contain our source address or else the response from the target system will not know where to return to.
- The same applies to spoofing our IP when using nmap. We CAN spoof our IP address in nmap, but as a result, any response and any info we are trying to gather will return to the spoofed IP. Not very useful, if we are scanning for info gathering.
- A better solution is to obfuscate our IP address. In other words, bury our IP address among many IP addresses so that the network/security admin can't pinpoint the source of the scan. Nmap allows us to use decoy IP addresses so that it looks like many IP addresses are scanning the target.
  - We can do this by using the -D switch. the D switch will simply give several IP addresses. In another word, different IP addresses will show up on the victim machine including yours but it will be difficult to pinpoint which one is yours.

    ```
    sudo nmap -sS 192.168.56.111 -D 10.0.0.1,10.0.0.2,10.0.0.4
    ```

3.4 Enumeration

- Enumeration is a phase in the information-gathering process during cybersecurity assessments.
- It involves extracting detailed information about a target system or network to understand its configuration, services, and potential vulnerabilities.
- The goal of enumeration is to create a comprehensive profile of the target, aiding in the identification of security weaknesses.
- This includes identifying version of servers, services and and operating systems the target uses.

3.4.1 Enumeration for Operating systems

- At this point of the information gathering process, we should now have documented a list of IP addresses, active machines, and open ports identified from the target organization. The next step in the process is determining the running operating system of the active machines in order to know the type of systems we're pentesting.
- Using **Nmap**, we issue the following command with the **-O** option to enable the OS detection feature.

  ```
  nmap -O 192.168.56.111
  ```

  - ∗ The terminal should warm you that this command requires root privilege! This is a very noisy command and nmap is warning you that an Intrusion detection system will pick it up.

  * Root privilege means to use **sudo** before the command to get Super User privilege.

3.4.2 Enumeration for services

Determining the services running on specific ports will ensure a successful pentest on the target network. It will also remove any doubts left resulting from the OS fingerprinting process.

- Let's begin the process of service fingerprinting by opening a terminal window

  nmap -sV 192.168.56.111

- We can merge options in Nmap, for example we can conduct enumeration for both services and the operating system at the same time

  nmap -sV -O 192.168.56.111

  * -sV asks for the banner—header or self identification—of each open port found, which is what it uses as the version
  * -O tells Nmap to try to guess the operating system running on the target using the information collected from open ports and versions

3.5 Using Nmap scripting Engine (NSE) for enumeration

- The Nmap scripting engine is one of Nmap's most powerful and, at the same time, most flexible features. It allows users to write their own scripts and share these scripts with other users for the purposes of networking, reconnaissance, etc.[1] These scripts can be used for:
  - Network discovery
  - More sophisticated and accurate OS version detection
  - Vulnerability detection
  - Backdoor detection
  - Vulnerability exploitation

- To locate Nmap scripting Engine, you need to move to the script where they are installed. in Kali, their location is:

  cd /usr/share/nmap/scripts

  * Can you categorise the various scripts ? which script you think will be useful when gathering information about our OWASP Vulnerable machine?

[1]https://null-byte.wonderhowto.com/how-to/hack-like-pro-using-nmap-scripting-engine-nse-for-reconnaissance-0158681/

3.5.1 Apache server enumeration

- We have identified from previous activities that Apache server is running. Let us try to get some more information about it.
- First, although we have identified previously which version of Apache is running, we can check if the **Banner script** can reveal more information. Which web server software is running on the target server and also find out the version using nmap.

  nmap -sV -script banner 192.168.56.111

- An Apache server usually has some common folders. We will use Dirbuster to see if those folders exist in our server.

3.5.2 Detecting & Identifying a web application Firewall

- Nmap includes a couple of scripts to test for the presence of a WAF. Let's try some on our vulnerable-VM.
  - Detecting WAF

    nmap -p 80,443 --script=http-waf-detect 192.168.56.111

    · the OWASP VM does not have a WA- Switch Kali to NAT network. and try the same command on a server that actually has a firewall protecting it.

    nmap -p 80,443 --script=http-waf-detect www.cwscenario.site
  - fingerprinting WAF
    - You can also check if WAF is installed on other websites when you are connected to the Internet.

      nmap -p 80,443 --script=http-waf-fingerprint www.cwscenario.site
  - We can do both as well.

    nmap -p 80,443 --script http-waf-detect,http-waf-fingerprint www.tryhackme.com

- **A web application firewall (WAF)** is a device or a piece of software that checks packages sent to a web server in order to identify and block those that might be malicious, usually based on signatures or regular expressions.
- We can end up dealing with a lot of problems in our penetration test if an undetected WAF blocks our requests or bans our IP address. When performing a penetration test, the reconnaissance phase must include the detection and identification of a WAF, **intrusion detection system (IDS)**, or **intrusion prevention system (IPS)**. This is required in order to take the necessary measures to prevent being **blocked** or **banned**.

- we will use different methods, along with the tools included in Kali Linux, to detect and identify the presence of a web application firewall between our target and us.

detect Vs. fingerprint

- waf-detect
  - The primary purpose of the waf-detect script is to detect the presence of a WAF by sending specific requests and analysing the responses.
  - you are simply gathering information on the server, part of it is to see if a WAF is present.
- waf-fingerprint
  - The waf-fingerprint script is designed to gather detailed information about a detected WAF. It goes beyond simple detection and attempts to fingerprint and identify the specific WAF product in use.
  - In this case, we are not only identifying the presence of WAF but also enumerating the WAF firewall if detected.

4 Other scanning and enumeration tools

- Although Nmap is the most popular tool and is the most widely used, it is not the only port scanner available and, depending on varying tastes, maybe not the best either.
- There are some other alternatives included in Kali Linux. We present some examples below. There are many others you can explore.
  - One in particular that is interesting in how it represents the ARP table and discover devices is Netdiscover but there are others pre-installed on Kali
  - Netdiscover uses the ARP table to discover devices by sniffing ARP requests and Responses.
  - You can try it by yourself. Type
    - **sudo netdiscover -i eth0**
  - the -i eth0 means that we are simply sniffing on the network interface eth0.
  - If will take some time as it will check all possible local IP address. By defaults it starts with 192 prefix
  - You can specify the address or range to speed it up
    - **sudo netdiscover -i eth0 -r 192.168.56.0/24**
- Example of other pre-installed tools for information gathering on Kali Linux are:
  - hping3

sudo hping3 --traceroute -V -1 cwscenario.site

- masscan
  - Go to the manual of the tool and check its SYNOPSIS.

4.1 Mapping the network

- Another network scanner and mapper tool that some administrator use to keep track of their network is **legion**. Legion is pre installed on Kali and you can run by typing

  sudo legion

- Try Legion and note that the settings you change align with the Nmap options you used in previous activities
- Legion is basically an Nmap graphical representation. It employs the Nmap commands in the background and presents them in GUI mode.

5 Checklist

- **Lab objectives**

  Understand Passive and Active reconnaissance ☐

  Understand Scanning and Enumeration ☐

  Conduct Reconnaissance and Scanning stages on the OWASP machine ☐

- You can now complete the following tasks from the assessment:

  A- Information Gathering ☐

  (1) OSINT Activities ☐

  (2) Website Reconnaissance ☐

  (3) Port Scanning and Enumeration ☐

More tools

There are many more tools and methods you can use to conduct OSINT and Information gathering.

- A simple search on Github for OSINT returns many tools OSINT search
- A simple search on Github for OSINT returns many tools Scanning and Discovery search
- You are encouraged to explore some external tools in your own time.

Nmap summary

- Nmap is a port scanner, this means that it sends packets to a number of TCP or UDP ports on the indicated IP address and checks if there is a response. If there is, it means the port is open; hence, a service is running on that port.
- In the first command, with the -sn parameter, we instructed Nmap to only check if the server was responding to the ICMP requests (or pings). Our server responded, so it is alive.
- The second command is the simplest way to call Nmap; it only specifies the target IP address. What this does is ping the server; if it responds then Nmap sends probes to a list of 1,000 TCP ports to see which one responds and then reports the results with the ones that responded.
- The third command adds the following two tasks to the second one:
- -sV asks for the banner—header or self identification—of each open port found, which is what it uses as the version
- -O tells Nmap to try to guess the operating system running on the target using the information collected from open ports and versions
- In each command we used, you can use **man** command to look at the various options that it can be used with.
- Other useful parameters when using Nmap are:
  - -sT: By default, when it is run as a root user, Nmap uses a type of scan known as the SYN scan. Using this parameter we force the scanner to perform a full connect scan. It is slower and will leave a record in the server's logs but it is less likely to be detected by an intrusion detection system.
  - -Pn: If we already know that the host is alive or is not responding to pings, we can use this parameter to tell Nmap to skip the ping test and scan all the specified targets, assuming they are up.
  - -v: This is the verbose mode. Nmap will show more information about what it is doing and the responses it gets. This parameter can be used multiple times in the same command: the more it's used, the more verbose it gets (that is, -vv or -v -v -v -v).
  - -p N1,N2,...,Nn: We might want to use this parameter if we want to test specific ports or some non-standard ports, where N1 to Nn are the port numbers that we want Nmap to scan. For example, to scan ports 21, 80 to 90, and 137, the parameters will be: -p 21,80-90,137.
  - --script=script_name: Nmap includes a lot of useful scripts for vulnerability checking, scanning or identification, login test, command execution, user enumeration, and so on. Check the use of some Nmap scripts at: https://nmap.org/nsedoc/scripts.