Assignment 1

Requirements

- This is a Group of 2 Assignment.
- It is highly recommended that you use Notion.so to create your assignments.
- Each task must be solved in order.
- At the top of each file, following **MUST** be specified in the following manner:

```
GroupID: <GroupID>
Member1: <Roll-Number>:<Name>
Member2: <Roll-Number>:<Name> [None:None if None]
Course: <Course-Name>
Date: <Date-of-Submission> (DD/MM/YYYY)
FlagsSubmitted: <Number-of-Flags submitted>
---
```

Example:

```
GroupID: 5
Member1: 190792:Ali Taqi Wajid
Member2: None:None
Course: CY243L - Penetration Testing - Lab
Date: 05/11/2023
FlagsSubmitted: 5
---
```

Submission Requirements

- You are required to submit a single PDF file with the following naming convention:

```
<course-code>-<batch|section>-<groupid>-<member1-rollnumber>_<member2-rollnumber>.pdf
## Example:
CY102L-F23-A-5-190792_190764.pdf
CY102L-F23-A-5-190792_None.pdf
```

⇒ *Any other file name will not be considered.*

Follow the Guidelines to ensure maximum marks.

Details

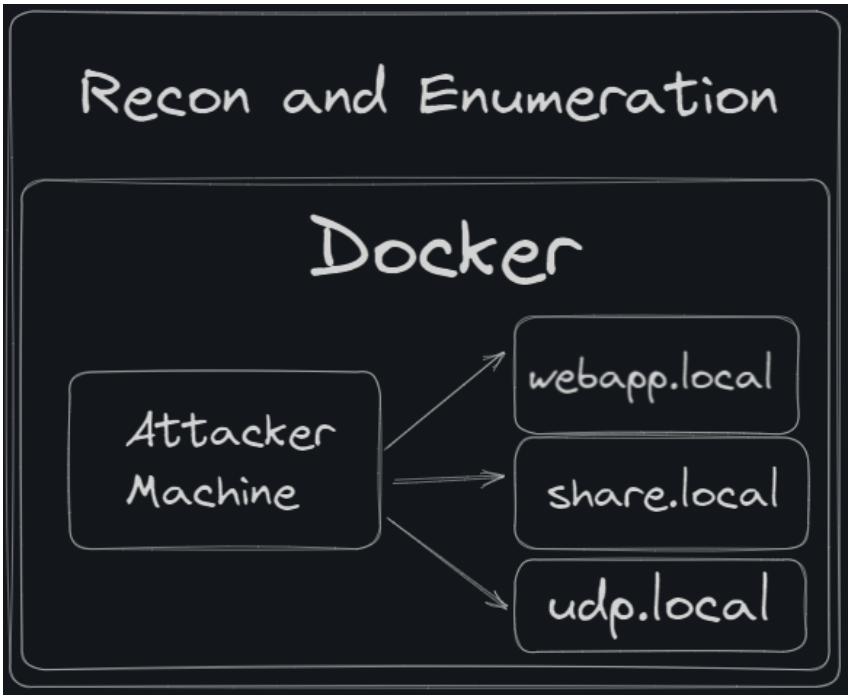
This assignment involves the following topics

- Scripting
- Recon and Enumeration

Following tools will be used:

- nmap
- gobuster
- smbclient/enum4linux

Overview of Lab Environment



For this lab, you will need to install the following on your `Kali Linux VMs`. Make Sure you have these setup for all the future exams:

```
sudo apt install docker.io docker-compose -y
```

👉 In case of errors, try to resolve them, most common error that you might will be related to repositories, you can fix that by simply googling and changing your repos.

Explanation of Structure

> For the setup of this entire Lab, you need atleast 2 **GB Disk Space free** on your Kali Linux VM and atleast **2 GB RAM** must be alloted to the VM.

The lab structure will firstly be deploying an attacker machine. We will be using this particular `Attacker Machine` in order to interact with the rest of the machines.

1. `webapp.local` : This is the web-app that's hosting a few webpages and sub-directories. We will performing the use of `Nikto`, `gobuster`, `ffuf` etc.
2. `share.local`: This container hosts a `samba` share, that we can enumerate using `SMBCLIENT` and `enum4Linux`.
3. `udp.local`: This container has a UDP Port open and a service running on it. You will connect to it using `nc` and then somehow get the flag from it.

> **NOTE:** You won't be able to access any of these hostnames from your own `Kali`. Only from the `AttackerMachine`. You can google about ***Docker***, how it works.

Lab Setup

Please download the attached zip file and copy it to your Virtual Machines. Once done, open a terminal (in the folder where you have the `Assignment-1.zip` file.)

```
unzip Assignment-1.zip
cd Assignment-1
chmod +x *.sh
./start.sh
```

⚠️ This process may take some time depending on your internet speed as all the files will be pulled from the internet and then come in a running condition.

```
{11:17}~ ➲ ls -l Assignment-1.zip
-rw-r--r-- 1 root root 2864 Oct 11 11:17 Assignment-1.zip
{11:17}~ ➲ unzip Assignment-1.zip
Archive:  Assignment-1.zip
   creating: Assignment-1/
  inflating: Assignment-1/.common
  inflating: Assignment-1/cleanup.sh
  inflating: Assignment-1/docker-compose.yml
  inflating: Assignment-1/start.sh
 extracting: Assignment-1/stop.sh
{11:17}~ ➲
```

```
{11:18}~/Assignment-1 ➲ ./start.sh
[*] Found: Docker version 20.10.21, build 20.10.21-0ubuntu1~20.04.2
/usr/lib/python3/dist-packages/paramiko/transport.py:219: CryptographyDeprecationWarning: Blowfish has been deprecated
  "class": algorithms.Blowfish,
[*] Found: docker-compose version 1.25.0, build unknown
Checking if container attacker is running : No.
Checking if container webapp is running : No.
Checking if container share is running : No.
Checking if container udp is running : No.
Setting up the containers...
/usr/lib/python3/dist-packages/paramiko/transport.py:219: CryptographyDeprecationWarning: Blowfish has been deprecated
  "class": algorithms.Blowfish,
Creating network "assignment-1_base" with driver "bridge"
Pulling attacker (theflash2k/cy243l-attacker:recon)...
recon: Pulling from theflash2k/cy243l-attacker
3f94e4e483ea: Pulling fs layer
a1b773f82077: Pulling fs layer
2920a386126f: Pulling fs layer
e2ab75b99ef9: Pulling fs layer
69e64152c578: Pulling fs layer
05538eb786f7: Pulling fs layer
4d3cb25acfd9: Pulling fs layer
361342d1c7d6: Pulling fs layer
4c0ad7aa6bc5: Pulling fs layer
```

⛔ **NOTE:** If you face any error, try and resolve them using Google first. The most common error may involve not granting execution permission to the script, not having enough space on disk or not having docker/docker-compose installed.

Once everything's done, you'll get a screen like this:

```
Status: Downloaded newer image for theflash2k/cy243l-recon-udp.local:latest
Creating assignment-1_udp-local_1    ... done
Creating assignment-1_attacker_1     ... done
Creating assignment-1_share-local_1  ... done
Creating assignment-1_webapp-local_1 ... done
Attacker container ID: 2cd80ce91965
Webapp container ID: f9de5e44f9ea
Share container ID: addec97899bf
UDP container ID: addec97899bf 05e59299dede
Attaching to the attacker container...
Use the following command if you ever want to open another shell on the attacker machine:
docker exec -it 2cd80ce91965 bash


Spawning a shell on attacker...
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kali@kali:~$
```

```
kali@kali:~$ ls -l
total 12
-rw-r--r-- 1 root root 345 Oct 11 10:56 README.txt
-rw-r--r-- 1 kali kali 109 Oct 11 10:56 flag.txt
-rw-r--r-- 1 kali kali 169 Oct 11 10:56 hosts.txt
kali@kali:~$ cat flag.txt
Well done! You've done the setup correctly. Here's your flag: CY243L{SETUP_4c96ff56a94a680d0e5ab9d144907a53}
kali@kali:~$
```

This means that everything has deployed successfully and you can start your assignment.

⛔ Once you have completed the assignment, you may run:

```
./stop.sh
./cleanup.sh # This will delete all the downloaded images. So make sure you run this when you've completed your assignment.
```

Flags:

> **NOTE:** Once again, all flags will be dynamically generated, all service ports will be dynamically alloted and **NO** two students will get the same flag/same port. If found, the assignment will be cancelled for all members of both groups.

There are total of 5 Flags in this assignment

1. 1x `attacker` (As shown in the above screenshot)
2. 1x `webapp.local` [Considering this requires some scripting, it carries 2 points]
3. 2x `share.local` [Enumerate all services for hint to the second flag.]
4. 1x `udp.local`

Reporting

The final attached report must contain (in detail the following) for each endpoint

- Enumeration (How did you find the open ports)
- Exploitation (Which tool did you use to exploit the service)
- Flag(s) found.

A seperate heading that will contain only the flags found and their screenshots.

Tools and Wordlists:

All the required tools have been installed and provided. If any other tool is required, it can be installed using `apt` inside the container.

Along with the tools, following two wordlists have also been provided:

- `/opt/common.txt`
- `/opt/wordlist.txt` → This is the `medium` wordlist from Dirbuster.

Additional Questions

Following questions need to be answered as well

- What service is running on `webapp.local`?
- What user exists on `share.local`?
- What is the password for the user on `share.local`? [Add a screenshot of how you found it]
- How many tcp ports are open on `udp.local`?
- How many udp ports are open on `udp.local`?

Deliverables

- A report

At the of your report, add a heading of `Appendix` and add the script/command you used to find the flag in `webapp.local`.

Good Luck!