Penetration Testing Framework Setup with DVWA & Metasploitable2

1. Project Overview

This project simulates a penetration testing environment where I exploit known vulnerabilities in vulnerable web applications and systems to demonstrate penetration testing techniques. Using **DVWA** (Damn Vulnerable Web Application) and **Metasploitable2**, I was able to create a real-world testing environment where I performed penetration testing on web application vulnerabilities like **File Upload** and **Reverse Shell Exploitation**. The main objective was to learn and showcase the use of penetration testing tools and techniques to assess and exploit vulnerabilities.

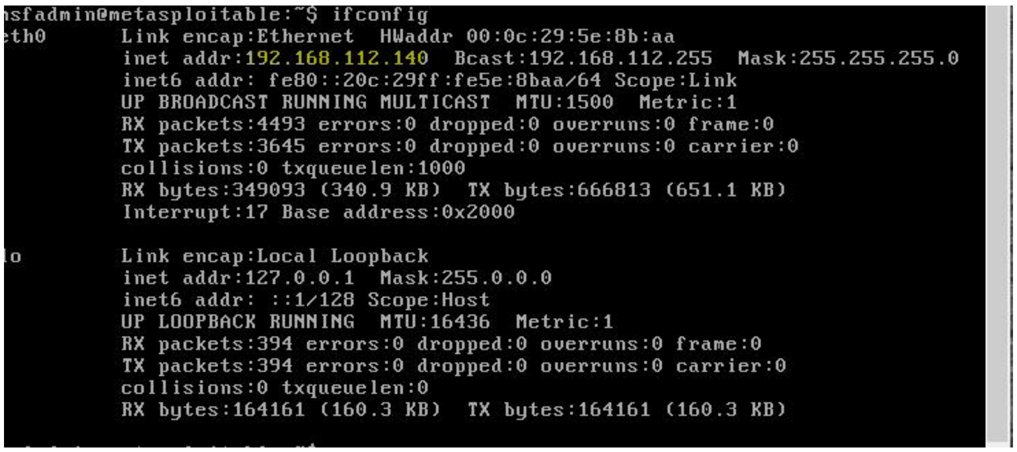
2. Tools Used

- **DVWA (Damn Vulnerable Web Application)** – A PHP/MySQL web application designed for testing vulnerabilities.
- **Metasploitable2** – A vulnerable virtual machine designed for testing and practicing penetration testing techniques.
- **Kali Linux** – A penetration testing and security auditing Linux distribution with tools like Netcat.
- **Netcat (nc)** – A simple, yet powerful tool used to listen for incoming connections.
- **PHP** – Used to craft reverse shell payloads for exploitation.

3. Project Breakdown

A. Setting Up the Lab Environment

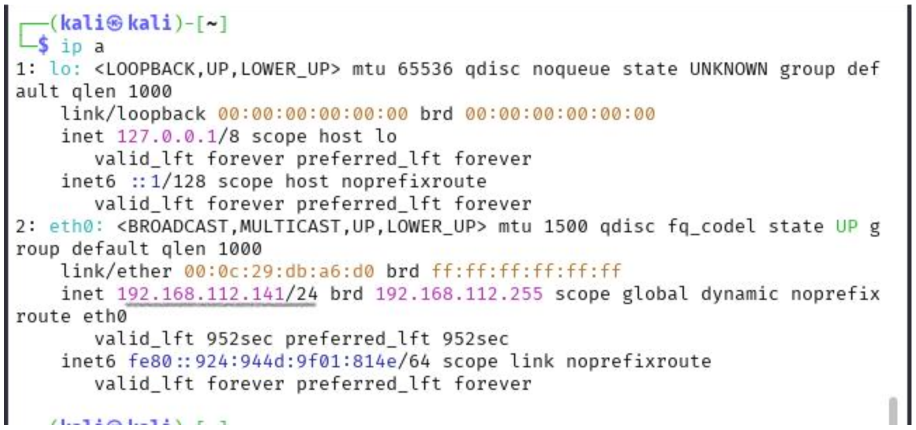
1. **VMware Configuration**:
   - I created two virtual machines using VMware Workstation:
     - **Metasploitable2** was set up as a vulnerable system designed for testing.
     - **Kali Linux** was set up as the attack machine running all necessary penetration testing tools.
   - Both VMs were placed on the **same network** to allow communication between them (host-only network).
2. **Networking Setup**:
   - Both machines were assigned IP addresses within the same subnet to ensure they could communicate with each other. For example:
     - **Metasploitable2**: 192.168.112.140



```
sfadmin@metasploitable:~$ ifconfig
th0       Link encap:Ethernet  HWaddr 00:0c:29:5e:8b:aa
          inet addr:192.168.112.140  Bcast:192.168.112.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5e:8baa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4493 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3645 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:349093 (340.9 KB)  TX bytes:666813 (651.1 KB)
          Interrupt:17 Base address:0x2000

o         Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:394 errors:0 dropped:0 overruns:0 frame:0
          TX packets:394 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:164161 (160.3 KB)  TX bytes:164161 (160.3 KB)
```

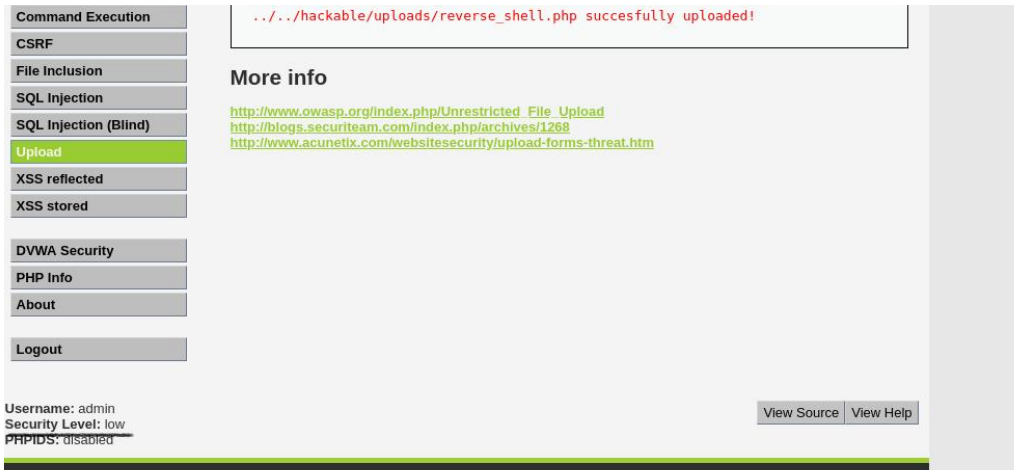
     - **Kali Linux**: 192.168.112.141



```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 00:0c:29:db:a6:d0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.112.141/24 brd 192.168.112.255 scope global dynamic noprefix
route eth0
       valid_lft 952sec preferred_lft 952sec
    inet6 fe80::924:944d:9f01:814e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

B. Penetration Testing Framework

1. **DVWA Configuration**:
   - On Metasploitable2, I installed **DVWA** and configured it to use **low security** settings to make the web application vulnerable.



```
../../hackable/uploads/reverse_shell.php succesfully uploaded!
```

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

Username: admin
Security Level: low
PHPIDS: disabled

   - I accessed the DVWA interface via the browser using the IP address of the Metasploitable2 machine, such as http://192.168.112.140/dvwa.

2. **Metasploitable2 Setup**:

   - **Metasploitable2** contains many known vulnerabilities, including a vulnerable web application that allows for easy exploitation. I used **DVWA**’s file upload vulnerability to upload a PHP reverse shell.
   - After uploading the shell, I triggered it to connect back to my Kali machine.

C. Exploit: File Upload Vulnerability

1. **Upload Attack**:
   - I exploited the **file upload vulnerability** in DVWA by uploading a **PHP shell**. The goal was to get the web server to execute the PHP code, which would allow me to establish a reverse shell connection back to my Kali Linux machine.
2. **Shell Upload**:
   - I uploaded the **reverse_shell.php** file and confirmed that it was stored in the upload directory on the Metasploitable2 web server, such as /hackable/uploads/reverse_shell.php.

D. Reverse Shell

1. **Listener Setup on Kali**:
   - On Kali Linux, I set up a **Netcat listener** on port 4444 to wait for an incoming connection from the uploaded PHP reverse shell.
2. **Triggering the Reverse Shell**:

  - I accessed the uploaded **reverse_shell.php** file by visiting http://192.168.112.140/dvwa/hackable/uploads/reverse_shell.php?cmd=whoami.
  - This triggered the reverse shell to connect back to my Kali machine, and I received a shell with control over the Metasploitable2 machine.

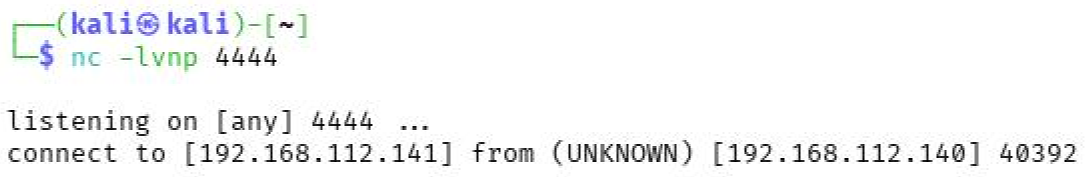
4.Challenges & Solutions

- **Firewall Issues**:
  - Initially, I faced a "connection refused" error when attempting the reverse shell due to firewall issues on the Kali machine.
  - **Solution**: I checked the firewall settings on Kali and opened port 4444.
- **PHP Syntax Error**:
  - The reverse shell code threw a **syntax error** because of the use of **incorrect PHP array syntax**.
  - **Solution**: I updated the PHP code to use the correct array syntax for compatibility.

5. Screenshots

Screenshots:

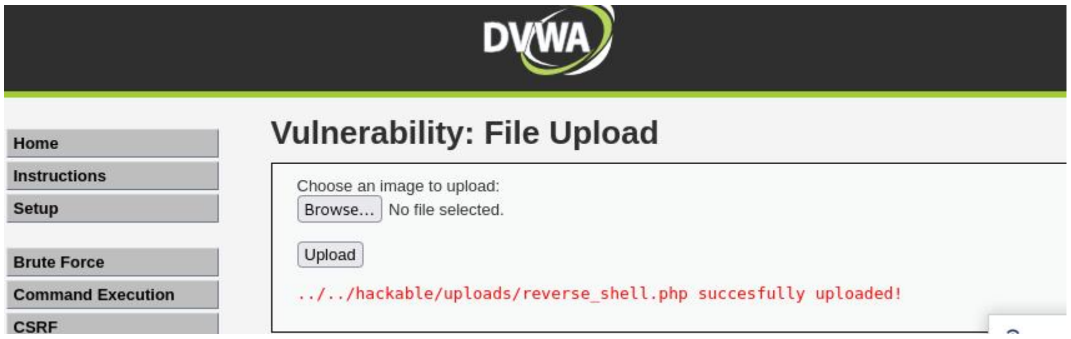
1. **Kali Netcat Listener**:



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
connect to [192.168.112.141] from (UNKNOWN) [192.168.112.140] 40392
```

2. **DVWA Upload Interface**:
   - Screenshot of the DVWA file upload page showing the upload of the reverse_shell.php file.

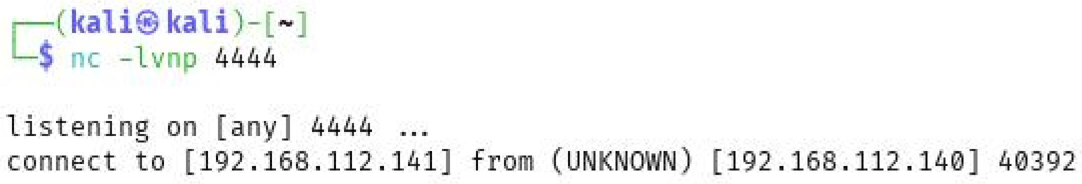


Vulnerability: File Upload

Choose an image to upload:
No file selected.

```
../../hackable/uploads/reverse_shell.php succesfully uploaded!
```

3. **Successful Reverse Shell Connection**:
   - Screenshot of Kali receiving a shell from Metasploitable2 after the reverse shell was triggered.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
connect to [192.168.112.141] from (UNKNOWN) [192.168.112.140] 40392
```

6. Learning & Outcome

- This project enhanced my understanding of web application vulnerabilities, penetration testing tools, and exploitation techniques.
- I learned about common web vulnerabilities, including **file upload vulnerabilities** and how reverse shell payloads can be used to gain unauthorized access.
- I also gained hands-on experience with configuring vulnerable web applications like DVWA, and learned to troubleshoot firewall and networking issues during penetration tests.

7. Improvements & Future Exploration

- **Privilege Escalation**: Next steps involve exploring privilege escalation techniques to gain root access on Metasploitable2 after establishing the initial reverse shell.
- **Persistence**: Implementing methods to maintain access to the target system after reboot.
- **Real-World Application**: I plan to expand this framework to simulate attacks against real-world applications using advanced vulnerability exploitation.