Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Flagggggs!

```
ebian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
ermitted by applicable law.
oot@target1:~# ls
lag4.txt
oot@target1:~# _
```

```
<div class="footer-social d-flex align-items-cent
        <a href="#"><i class="fa fa-facebook"></i
        <a href="#"><i class="fa fa-twitter"></i>
        <a href="#"><i class="fa fa-dribbble"></i
        <a href="#"><i class="fa fa-behance"></i>
    </div>
</div>
</div>
</div>
</div>
</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/popper.js/1.12.9/umd/popper.min.js" i
Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
<script src="js/vendor/bootstrap.min.js"></script>
<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBhO
ihAA"></script>
<script src="js/easing.min.js"></script>
<script src="js/hoverIntent.js"></script>
<script src="js/superfish.min.js"></script>
<script src="js/jquery.ajaxchimp.min.js"></script>
<script src="js/jquery.magnific-popup.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.sticky.js"></script>
<script src="js/jquery.nice-select.min.js"></script>
<script src="js/waypoints.min.js"></script>
<script src="js/jquery.counterup.min.js"></script>
<script src="js/parallax.min.js"></script>
<script src="js/mail-script.js"></script>
<script src="js/main.js"></script>
</body>
</html>
```

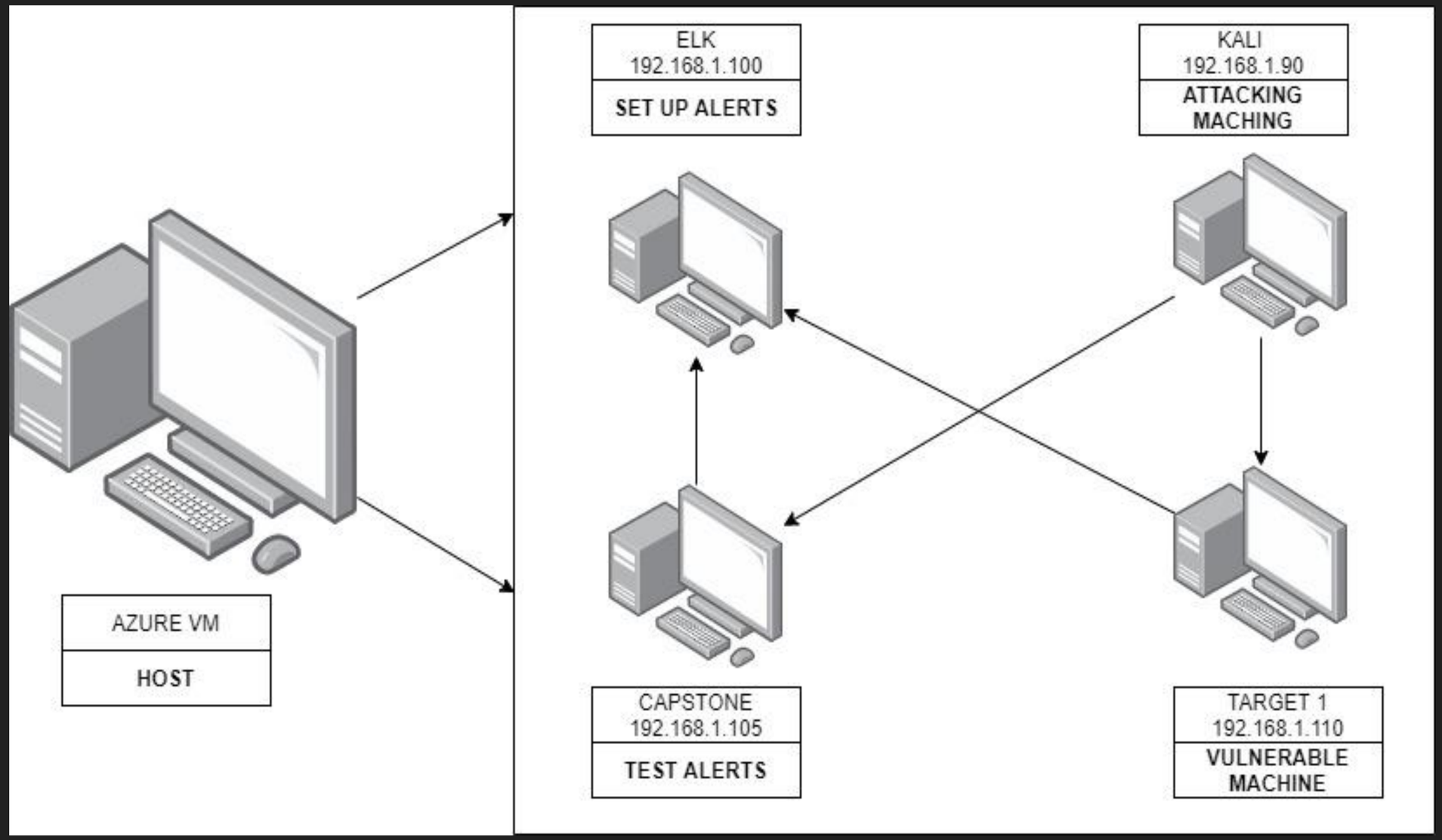
```
michael@target1:/var/www/html$
michael@target1:/var$ cd /www
-bash: cd: /www: No such file or directory
michael@target1:/var$ cd /wwww
-bash: cd: /wwww: No such file or directory
michael@target1:/var$ cd /var/www/html
michael@target1:/var/www/html$ ls
about.html   contact.zip  elements.html  img        js    Security - Doc  team.html  wordpr
contact.php  css          fonts          index.html scss  service.html    vendor
michael@target1:/var/www/html$ cd /var/www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

```
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:~#
```

```
rdpress/?page_id=2                                  0 | page      |                  0 |
  4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}

                     | flag3       |              | draft       | open          | open        |              |
      |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                                  0 | http://raven.local/wordpress/?p=4
                                 0 | post        |                  0 |
  5 |          1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}

                     | flag4       |              | inherit     | closed        | closed      |              | 4-revision-v1 |
      |        | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                                  4 | http://raven.local/wordpress/index.php/2
18/08/12/4-revision-v1/ |          0 | revision    |                  0 |
  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}

                     | flag3       |              | inherit     | closed        | closed      |              | 4-revision-v1 |
      |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                                  4 | http://raven.local/wordpress/index.php/2
18/08/13/4-revision-v1/ |          0 | revision    |                  0 |
----+-------------+---------------------+---------------------+-------------+-------------+-------------+---------------
-----+--------+---------------------+---------------------+-------------+-------------+-------------+---------------
----------------------+------------+-----------+---------------+---------------+
rows in set (0.00 sec)
```

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range:192.168.1.0/24
-192.168.1.1
Gateway:192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Debian 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux 8
Hostname: Target 1

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu
Hostname: ELK

Critical Vulnerabilities: Target 1

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

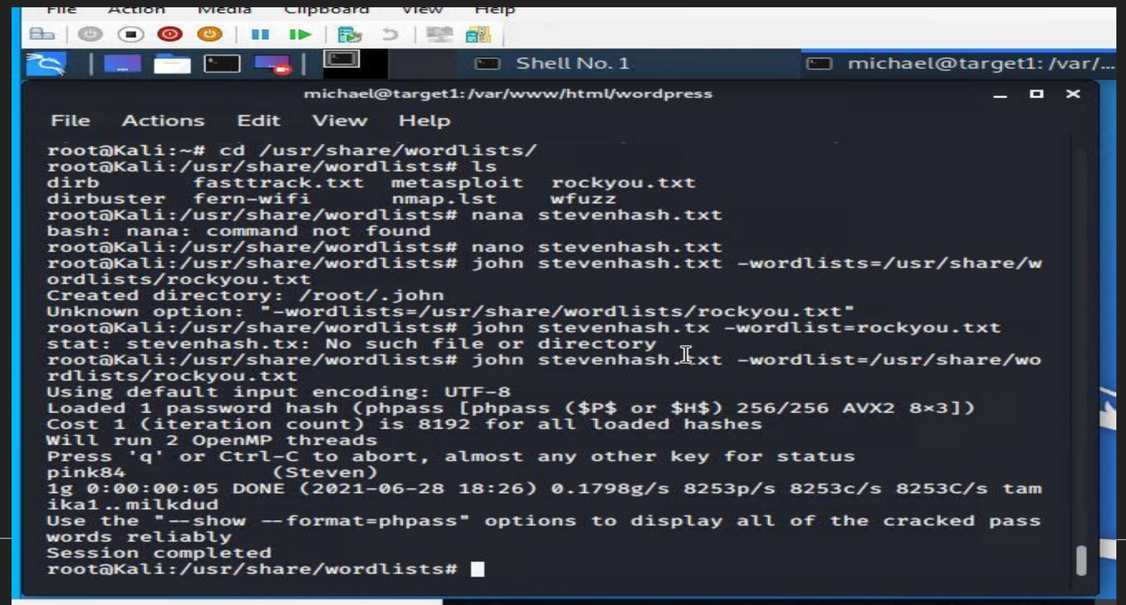
Vulnerability	Description	Impact
Brute force/ vulnerability	Having weak passwords that is easily guessable.no lockout policy	we were able to access michael's password and steven password by using john the ripper
wpscan/ vulnerability	enumerate wordpress users in the system	scanning the wordpress, gave us two users, michael and steven account.
sudo python access/escalated privilege	vulnerable to python script/escalated privileges	escalate to root privileges to gain full access to user account
MYSQL/ login	database stored sensitive information, including hashes. users should not be able to access this database	we were able to login with root and R@v3nSecurity


Exploits Used

Exploitation: Brute force/weak password

Summarize the following:

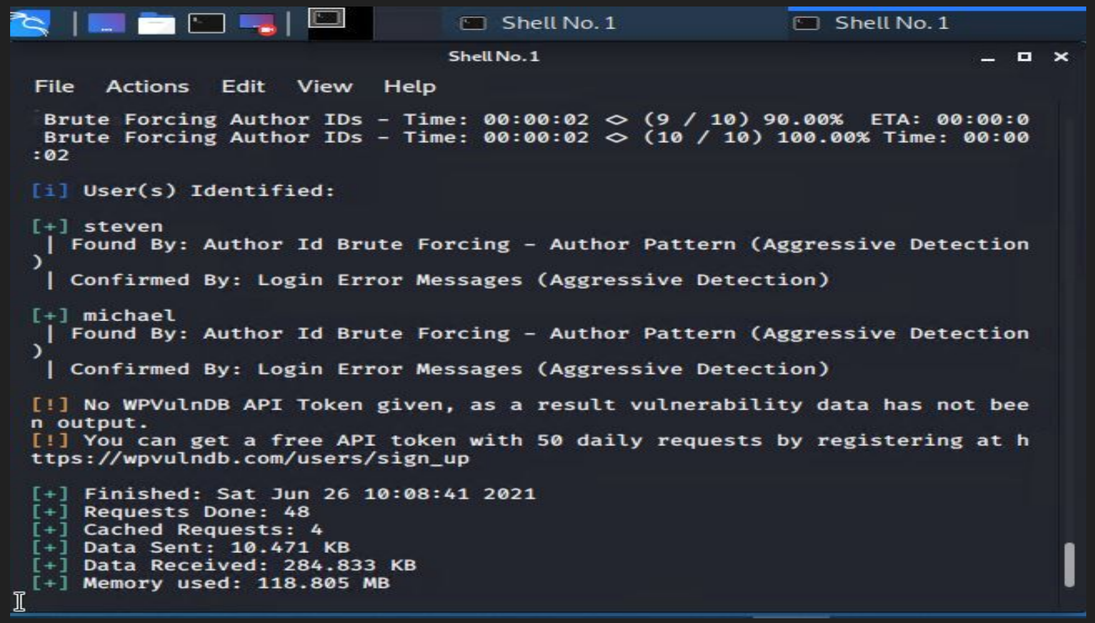
- How did you exploit the vulnerability? SSH'd into Steven and Michael's. account. Initiated Brute force and John the Ripper to retrieve passwords.
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.? We were granted root access into Steven and Michael's account. We brute forced Michael's password, and initiated John the Ripper for Steven's password.



Exploitation: Wpscan /Vulnerability

Summarize the following:

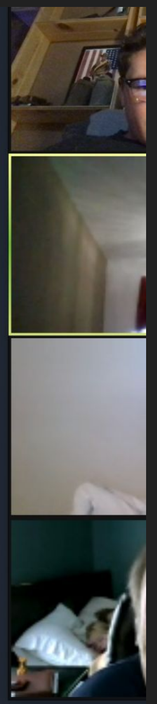
- How did you exploit the vulnerability? **Wpscan --url http://192.168.1.110/wordpress -eu**
- What did the exploit achieve? Wpscan informed us that the users are Steven and Michael.



```
Shell No. 1
File  Actions  Edit  View  Help
 Brute Forcing Author IDs - Time: 00:00:02 <> (9 / 10) 90.00%  ETA: 00:00:0
 Brute Forcing Author IDs - Time: 00:00:02 <> (10 / 10) 100.00% Time: 00:00
:02

[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 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: Sat Jun 26 10:08:41 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.833 KB
[+] Memory used: 118.805 MB
```

Exploitation: Mysql/ sensitive information

```
readme.html  wp-blog-header.php  wp-config-sample.php  wp-includes  wp-login.php      wp-signup.php
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * 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', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
```



Summarize the following:

- How did you exploit the vulnerability? We SSH'd into michaels account by guessing his simple password and located the wordpress config file path
- This exploit granted us easy access to the MySQL server by displaying the username and password

Avoiding Detection

Stealth Exploitation of Wordpress User Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors
- Which metrics do they measure?
  - When count () grouped over top 5 http.response.status_code
- Which thresholds do they fire at?
  - Above 400 for the last 5 minutes

Mitigating Detection

- How can you execute the same exploit without triggering the alert? Sending an obscene amount of HTTP requests. However, the alert is highly reliable so it would be hard not to trigger the alert.

- Are there alternative exploits that may perform better?
  - You could use a tool called Gobuster, which is used to Brute Force directories and files, the downside is that it could get flagged by a SIEM.

Stealth Exploitation of Local File Intrusion

Monitoring Overview

- Which alerts detect this exploit?
  - HTTP Request size
- Which metrics do they measure?
  - WHEN sum () OF http response bytes over bytes over all Documents
- Which thresholds do they fire at?
  - Above 3500 for last 1 minute

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Limit the size of files to less than 3500 btes.
- Are there alternative exploits that may perform better?
  - DDOS attacks

Stealth Exploitation of Directory Exploration

Monitoring Overview

- Which alerts detect this exploit? CPU Use Monitor
- Which metrics do they measure? When max() of system.process. cup.total.pct
- Which thresholds do they fire at? Above 0.5 for the last 5 minutes

Mitigating Detection

- How can you execute the same exploit without triggering the alert? Use an extremely high amount of CPU usage.
- Are there alternative exploits that may perform better? You can run a NMAP using

  nmap -sV -sS 192.168.1.110 where -sV attempts to determine the service running on the port and

  -sS is a TCP SYN port scan.