Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

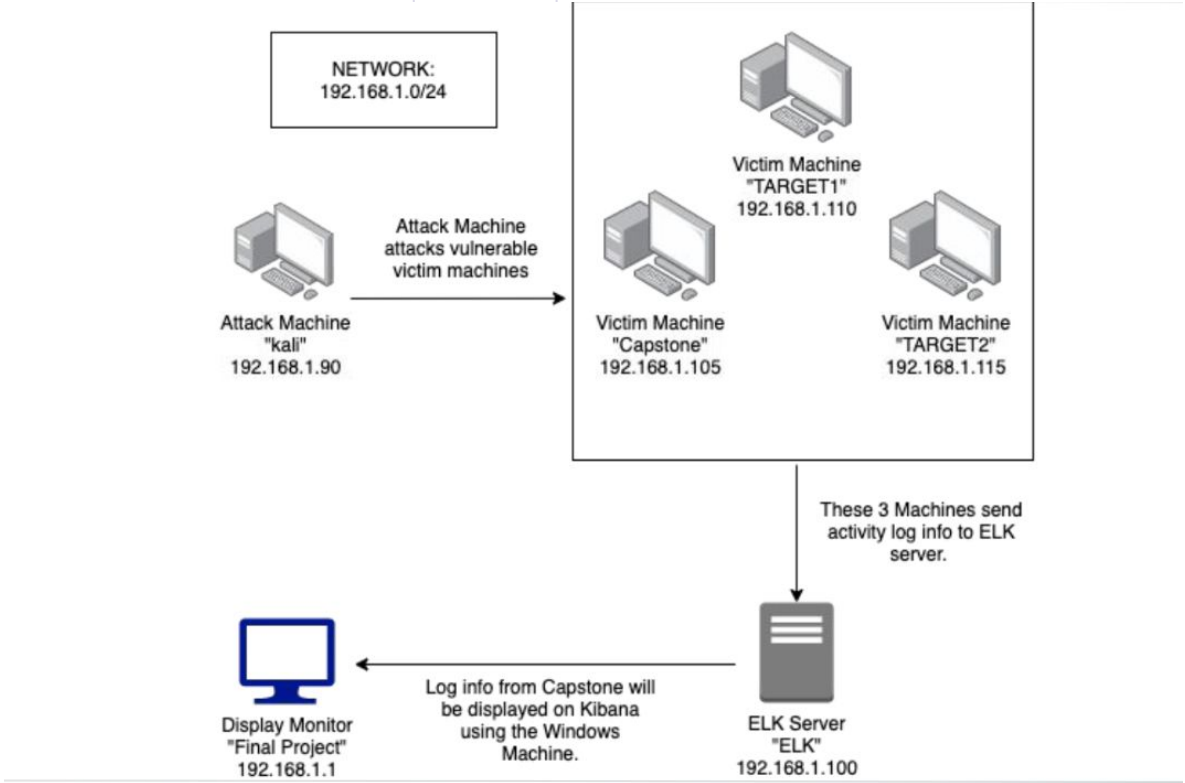
Liam, Shane, Isaac, Graham

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Attack

Target 1

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
WordPress xml rpc pingback	Can be exploited by use of POST to a specific file on the affected wordpress server	Targets internal layers, and changes configs on devices
WordPress Vulnerability scanner	Determines which hosts are vulnerable to the GHOST vulnerability through a call to the wordpress interface	This means if the target is vulnerable, the system will fault and give back a server error
WordPress DoS	WordPress parsing is vulnerable to a XML based DoS (Denial of Service)	It affects WordPress 3.5 - 3.9.2

Critical Vulnerabilities: Target 1

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

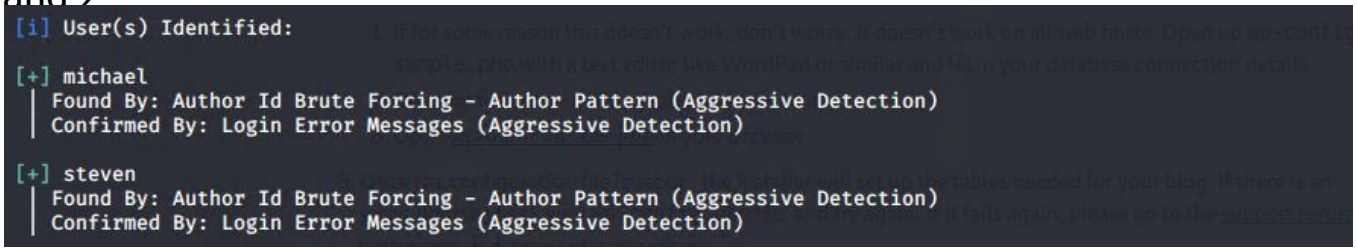
Vulnerability	Description	Impact
WordPress Username and Password Login scanner	Attempts to authenticate against a wordpress-site, using multiple usernames and passwords	This could give login access
Cron WordPress Attacks	The PingBack feature, which is enabled by default, can be used to attack other websites	It can attack multiple websites, and also potentially slow down or even crash your website if misused.

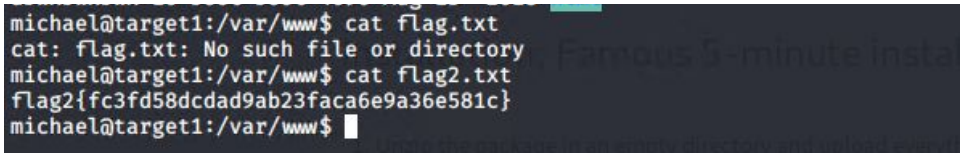
Exploits Used + Avoiding Detection

Exploitation: Open Port 22 SSH and Weak Passwords

Summarize the following:

- How did you exploit the vulnerability?
  - We used wpscan to find the users and guessed the weak password in order to SSH into the system.
- What did the exploit achieve?
  - The exploit granted us user shell access for Michael's account. We explored the files to find flags 1 and 2.





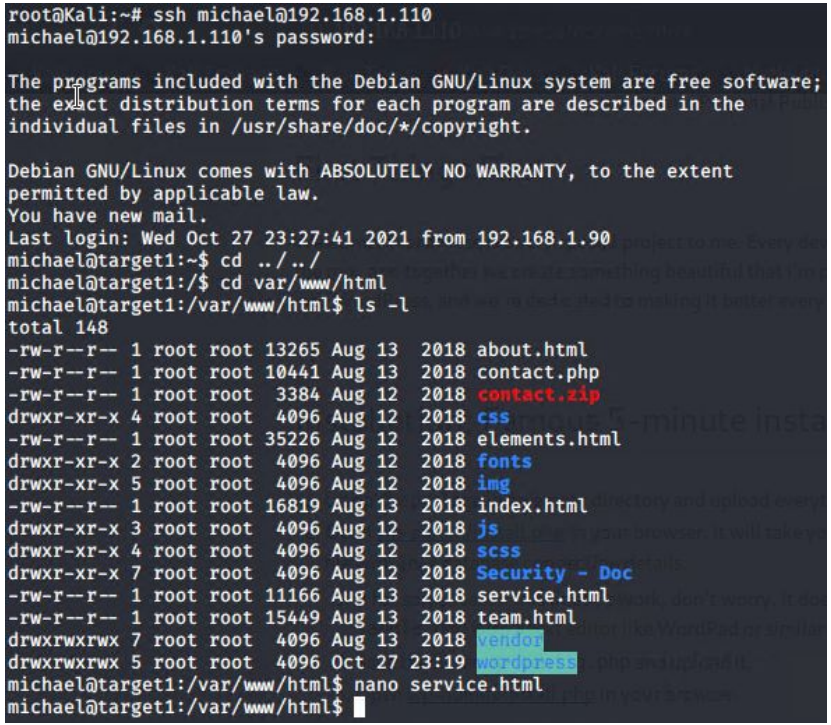
Stealth Exploitation of Open Port 22 SSH and Weak Passwords

Monitoring Overview

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

Mitigating Detection

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



Exploitation: WordPress Configuration and SQL Database

Summarize the following:

How did you exploit the vulnerability?

- The username and password to access the SQL database were in plaintext in the wp-config.php file and not hashed as is best practice, however this is a limitation of wordpress.

What did the exploit achieve?

- The exploit granted us mysql access and allowed us to find flag 3. And it also gave the password hash for steven, which meant John could be used to crack the password

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

```
As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboard</a> to delete this page and
create new pages for your content. Have fun! | Sample Page  |              | publish     | closed         | open         |              |
sample-page   |         |        | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                      |          0 | http://192.168.206.13
1/wordpress/?page_id=2                    |          0 | page      |                  |            0 |
|  4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

Stealth Exploitation: WordPress Configuration and SQL Database

Monitoring Overview & User Access

- 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 files.
  - Least privilege should be used in this case to restrict user access to the configuration file.

Mitigating Detection

- Employ IP address spoofing
- Brute-force SQL Database with Password cracking tool, Connect to the same network

Exploitation: Privilege Escalation

- We obtained Steven's password hash from the SQL database
- We cracked the password using John the Ripper and accessed his account
- We exploited Steven's python sudo privileges through a spawn shell
- The exploit achieve root access and allowed us to find flag 4

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-----------------
--------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email
ion_key | user_status | display_name  |
+----+------------+------------------------------------+---------------+-----------------
--------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org
        |           0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org
        |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-----------------
--------+-------------+---------------+
```

```
root@Kali:~/Desktop# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 30 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 26 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 25 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:20  3/3 0g/s 7961p/s 15836c/s 15836C/s ambel..111193
pink84           (steven)
```

```
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:~#
```

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

Defence

Critical Vulnerabilities: Target 1

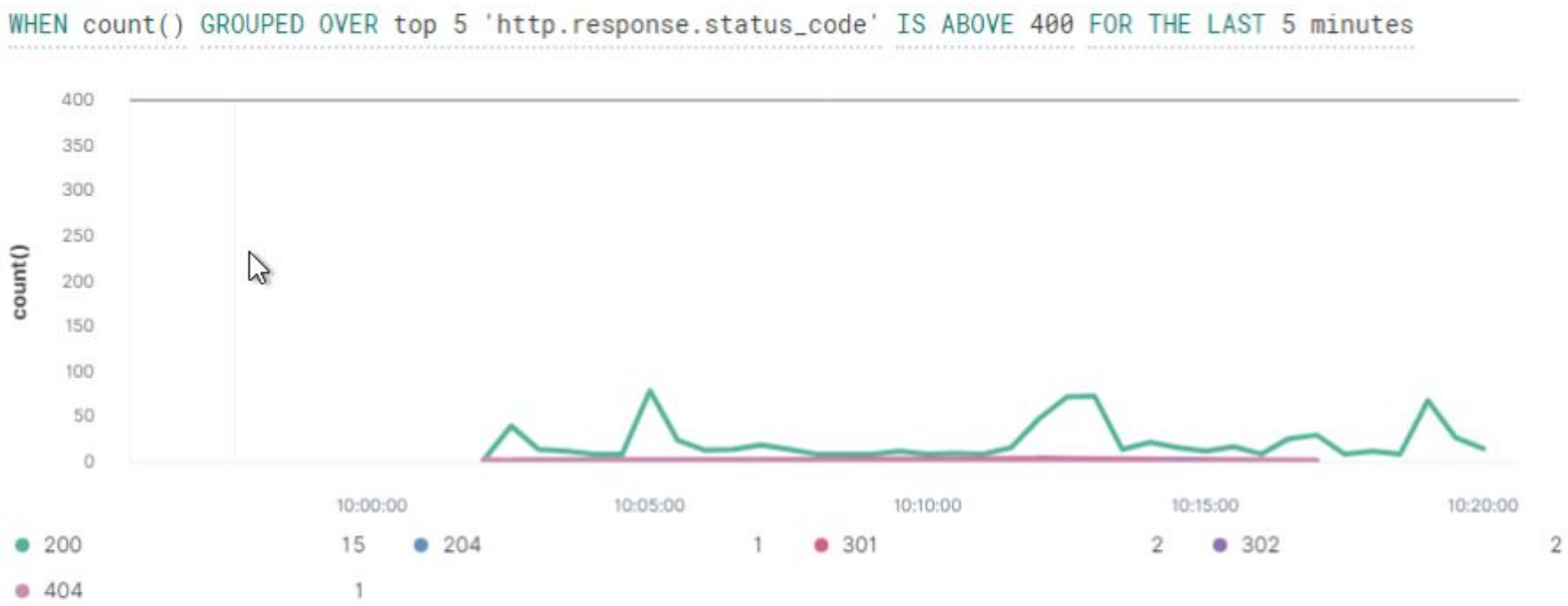
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Open access to SSH	An attacker can try SSH logins if the port is left open	Open SSH is vulnerable to brute force attacks
Brute Force Vulnerability	Attackers can perform rapid attempts to guess user names and passwords	Brute force will, given enough time, gain access to the system
Enumerate Wordpress	Unsecured Wordpress allows for information gathering and vulnerability assessment	Wordpress stored usernames, greatly reducing the time required to brute force

Alerts Implemented

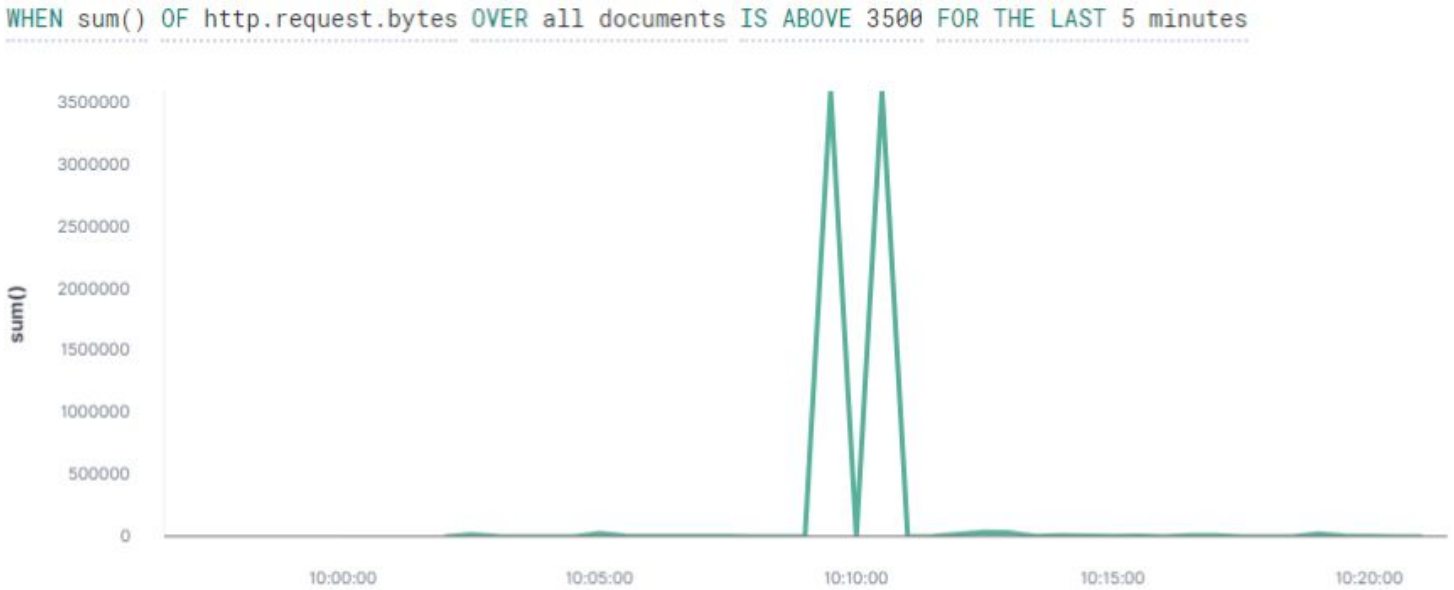
Excessive HTTP Errors

- Utilise Packetbeat to monitor *http.response.status_code*
- Implement a threshold of 400 for the last 5 minutes



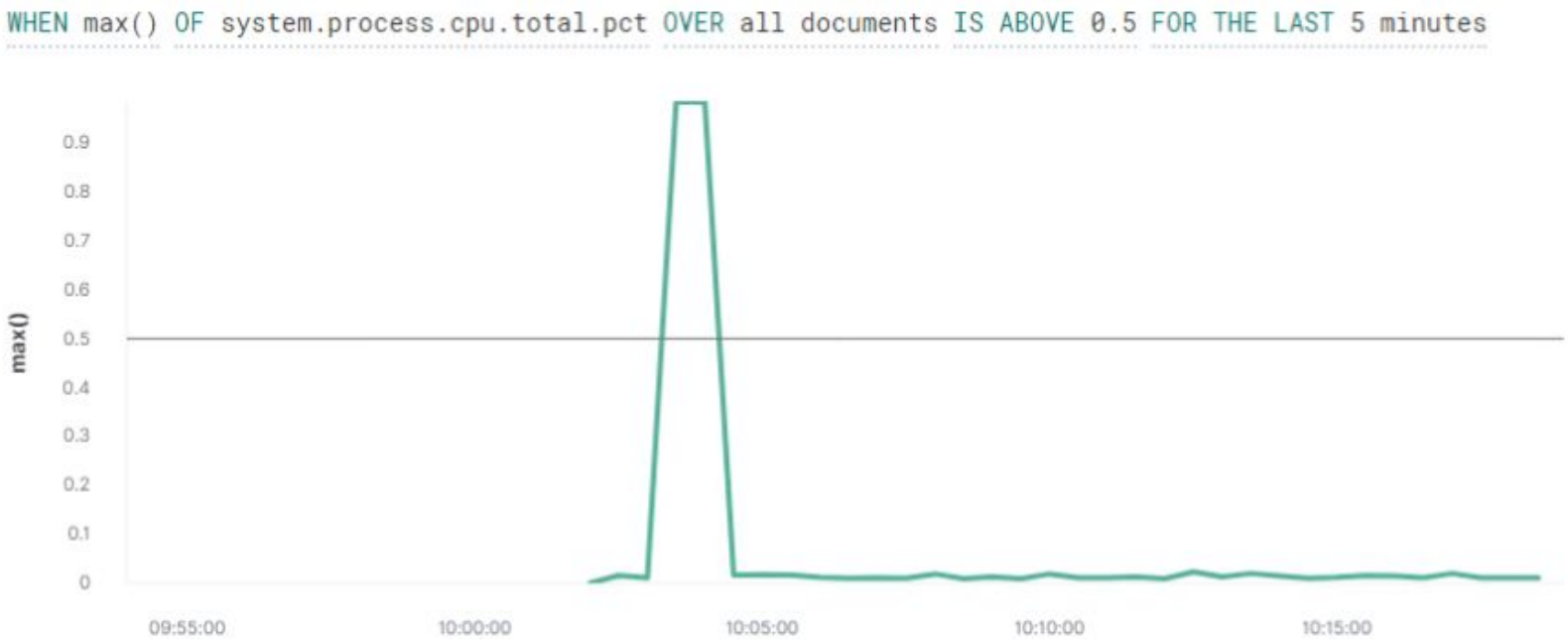
HTTP Request Size Monitor

- Utilise Packetbeat to monitor *http.request.bytes*
- Implement a threshold of 3500 for the last 5 minutes
- Provide a screenshot of the alert in action.



CPU Usage Monitor

- Utilise Metricbeat to monitor *system.process.cpu.total.pct*
- Implement a threshold of 0.5 for the last 5 minutes



Hardening

Hardening Against Open SSh on Target 1

There are a wide variety of hardening techniques for SSH. These can include:

- Set a custom TCP port by editing the */etc/ssh/sshd_config* file.
- Filter the SSH port through the firewall
- Implement SSH Passwordless Login. Uses keys to allow for login and removes the password prompt
- Disable empty passwords
- Set a custom SSH login banner. Doesn't stop logins but is a warning of active monitoring
- Keep SSH updated

Hardening Against Brute Force Vulnerability on Target 1

The simplest defense against a brute force attack is to implement an account lockout policy, however this leads to other forms of attack such as a Denial of Service or username harvesting. Other defenses include

- Implement a strong password policy
- Multi factor authentication
- Captcha
- Asking a ‘secret question’ after two failed login attempts

Hardening Against Wordpress Enumeration on Target 1

As Wordpress is a website builder with a large range there are many security aspects to consider:

- Keep Wordpress up to date.
- Disable **REST API** and **XML-RPC** if they are not being used
- Configure your web server to block requests to **/?author=<number>**
- Don’t expose **/wp-admin** and **/wp-login.php** directly to the public Internet

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

Ansible will be used to automatic the update process on each machine. This process can be scheduled to run on a regular basis through cron job. The following updates the Linux-based webserver:

name: System Update

hosts: webservers

apt: update_cache=yes force_apt_get=yes cache_valid_time=3600

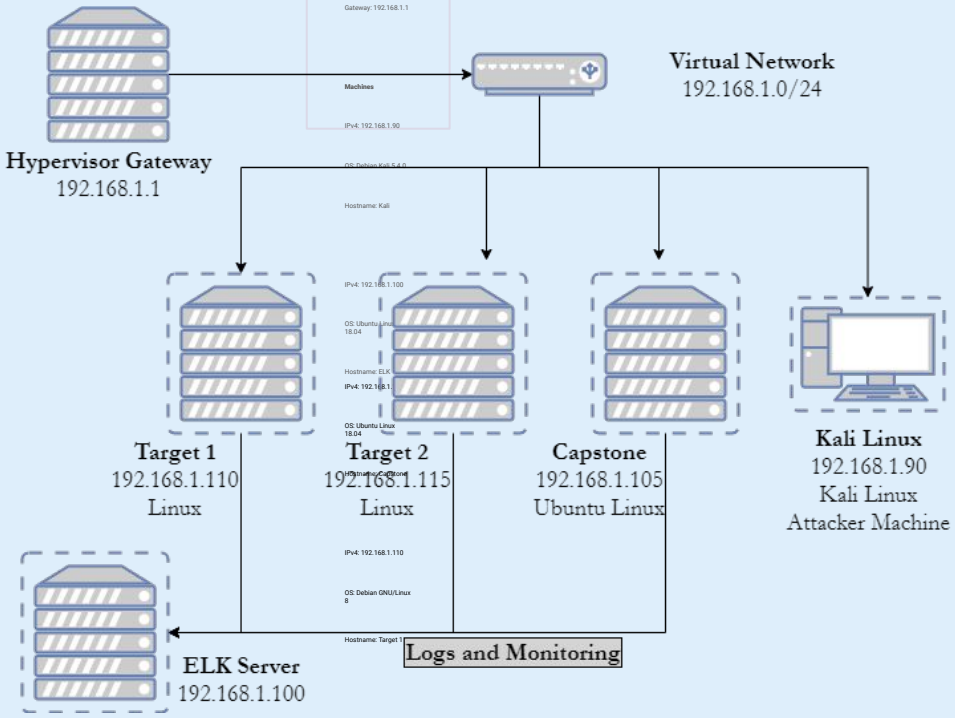
- name: reboot

- register: reboot_required_file

- stat: path=/var/run/reboot-required get_md5=no

Network Topology
& Critical Vulnerabilities

Network Topology



Traffic Profile

Traffic Profile

Your analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 166.62.111.64 10.0.0.201	Machines that sent the most traffic.
Most Common Protocols	Packets / Bytes TCP 88.5% / 89.6% UDP 11.0% / 0.1%	Three most common protocols on the network.
# of Unique IP Addresses	808 IPv4	Count of observed IP addresses.
Subnets	10.0.0.0/24 10.6.12.0/24 172.16.4.0/24	Observed subnet ranges.
# of Malware Species	1 - june11.dll (Trojan)	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

"Normal" Activity

- Browsing the web
- Watching Youtube

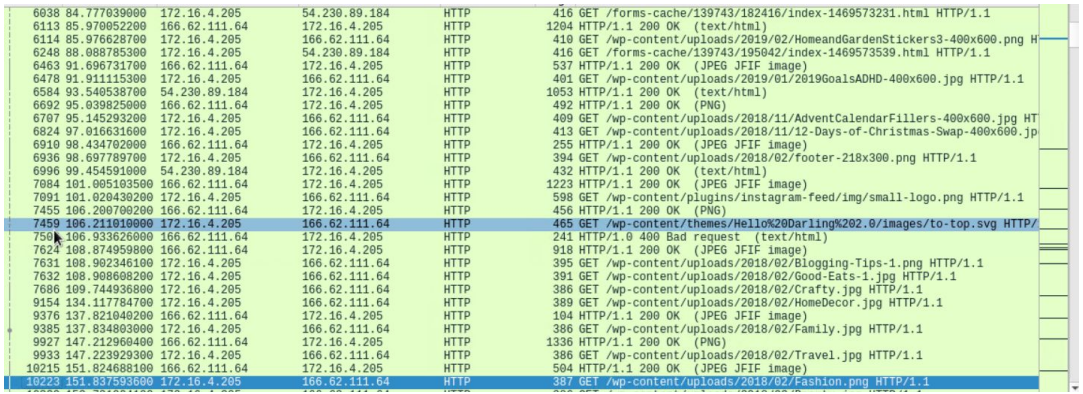
Suspicious Activity

- Malware Infection
- Illegal file downloads

Normal Activity

Browsing the Web

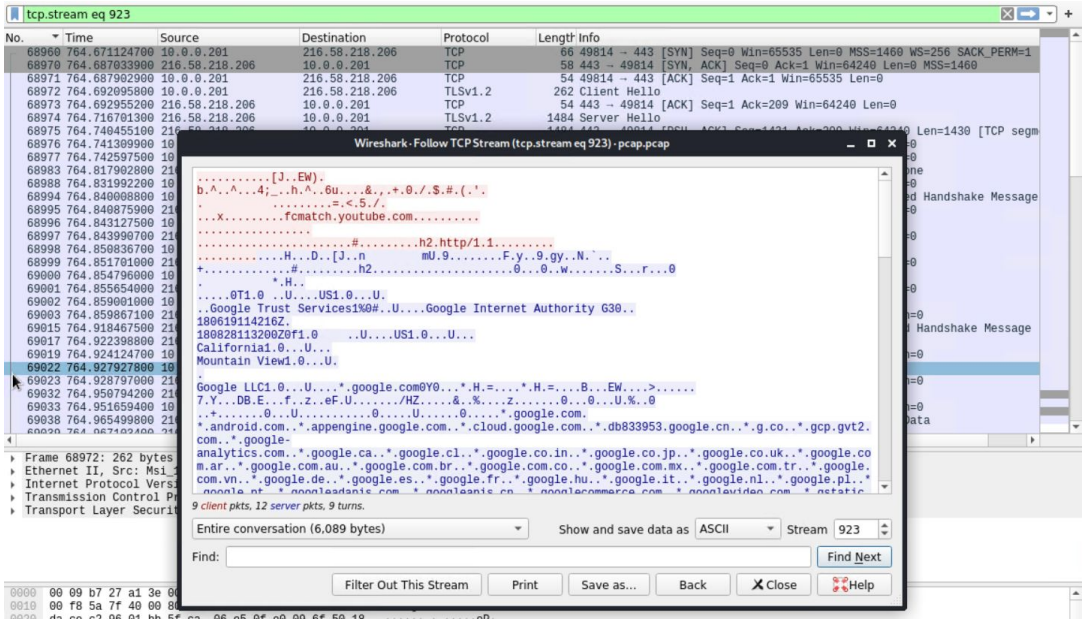
- Normal usage, browsing a wordpress framework based website.
- Normal use of HTTP



6038	84.777039000	172.16.4.205	54.230.89.184	HTTP	416	GET /forms-cache/139743/182416/index-1469573231.html HTTP/1.1
6113	85.970052200	166.62.111.64	172.16.4.205	HTTP	1204	HTTP/1.1 200 OK (text/html)
6114	85.976628700	172.16.4.205	166.62.111.64	HTTP	410	GET /wp-content/uploads/2019/02/HomeandGardenStickers3-400x600.png H
6248	88.088785300	172.16.4.205	54.230.89.184	HTTP	416	GET /forms-cache/139743/195042/index-1469573539.html HTTP/1.1
6463	91.696731700	166.62.111.64	172.16.4.205	HTTP	537	HTTP/1.1 200 OK (JPEG JFIF image)
6478	91.911115300	172.16.4.205	166.62.111.64	HTTP	401	GET /wp-content/uploads/2019/01/2019GoalsADHD-400x600.jpg HTTP/1.1
6584	93.540538700	54.230.89.184	172.16.4.205	HTTP	1053	HTTP/1.1 200 OK (text/html)
6692	95.039825000	166.62.111.64	172.16.4.205	HTTP	492	HTTP/1.1 200 OK (PNG)
6707	95.145293200	172.16.4.205	166.62.111.64	HTTP	409	GET /wp-content/uploads/2018/11/AdventCalendarFillers-400x600.jpg HT
6824	97.016631600	172.16.4.205	166.62.111.64	HTTP	413	GET /wp-content/uploads/2018/11/12-Days-of-Christmas-Swap-400x600.jp
6910	98.434702000	166.62.111.64	172.16.4.205	HTTP	255	HTTP/1.1 200 OK (JPEG JFIF image)
6936	98.697789700	172.16.4.205	166.62.111.64	HTTP	394	GET /wp-content/uploads/2018/02/footer-218x300.png HTTP/1.1
6996	99.454591000	54.230.89.184	172.16.4.205	HTTP	432	HTTP/1.1 200 OK (text/html)
7084	101.005103500	166.62.111.64	172.16.4.205	HTTP	1223	HTTP/1.1 200 OK (JPEG JFIF image)
7091	101.020430200	172.16.4.205	166.62.111.64	HTTP	598	GET /wp-content/plugins/instagram-feed/img/small-logo.png HTTP/1.1
7455	106.200700200	166.62.111.64	172.16.4.205	HTTP	456	HTTP/1.1 200 OK (PNG)
7459	106.211010000	172.16.4.205	166.62.111.64	HTTP	465	GET /wp-content/themes/Hello%20Darling%202.0/images/to-top.svg HTTP/
759	106.933626000	166.62.111.64	172.16.4.205	HTTP	241	HTTP/1.0 400 Bad request (text/html)
7624	108.874959800	166.62.111.64	172.16.4.205	HTTP	918	HTTP/1.1 200 OK (JPEG JFIF image)
7631	108.902346100	172.16.4.205	166.62.111.64	HTTP	395	GET /wp-content/uploads/2018/02/Blogging-Tips-1.png HTTP/1.1
7632	108.908608200	172.16.4.205	166.62.111.64	HTTP	391	GET /wp-content/uploads/2018/02/Good-Eats-1.jpg HTTP/1.1
7686	109.744936800	172.16.4.205	166.62.111.64	HTTP	386	GET /wp-content/uploads/2018/02/Crafty.jpg HTTP/1.1
9154	134.117784700	172.16.4.205	166.62.111.64	HTTP	389	GET /wp-content/uploads/2018/02/HomeDecor.jpg HTTP/1.1
9376	137.821040200	166.62.111.64	172.16.4.205	HTTP	104	HTTP/1.1 200 OK (JPEG JFIF image)
9385	137.834803000	172.16.4.205	166.62.111.64	HTTP	386	GET /wp-content/uploads/2018/02/Family.jpg HTTP/1.1
9927	147.212960400	166.62.111.64	172.16.4.205	HTTP	1336	HTTP/1.1 200 OK (PNG)
9933	147.223929300	172.16.4.205	166.62.111.64	HTTP	386	GET /wp-content/uploads/2018/02/Travel.jpg HTTP/1.1
10215	151.824688100	166.62.111.64	172.16.4.205	HTTP	504	HTTP/1.1 200 OK (JPEG JFIF image)
10223	151.837593600	172.16.4.205	166.62.111.64	HTTP	387	GET /wp-content/uploads/2018/02/Fashion.png HTTP/1.1

Watching Youtube

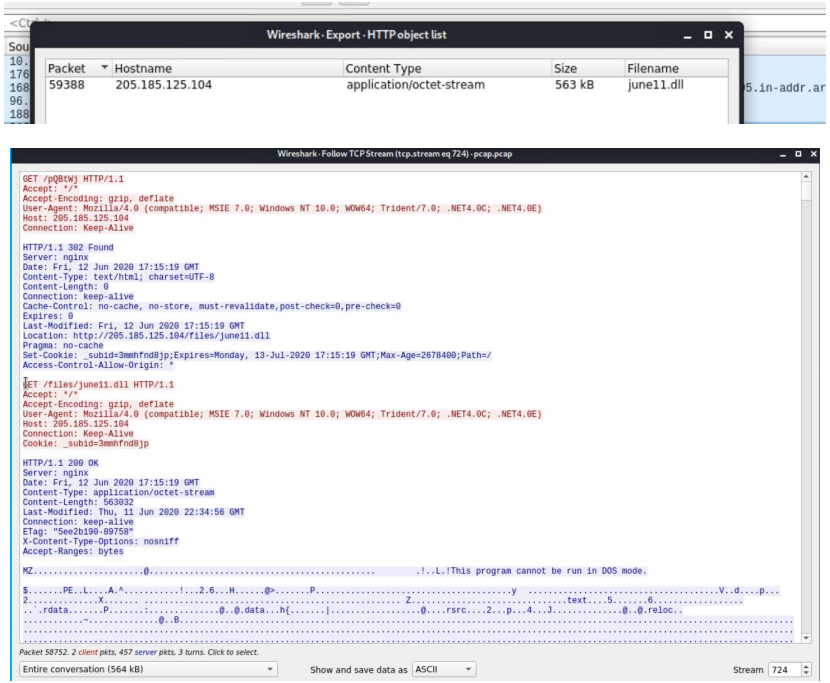
- TCP and TLS traffic to YouTube.com

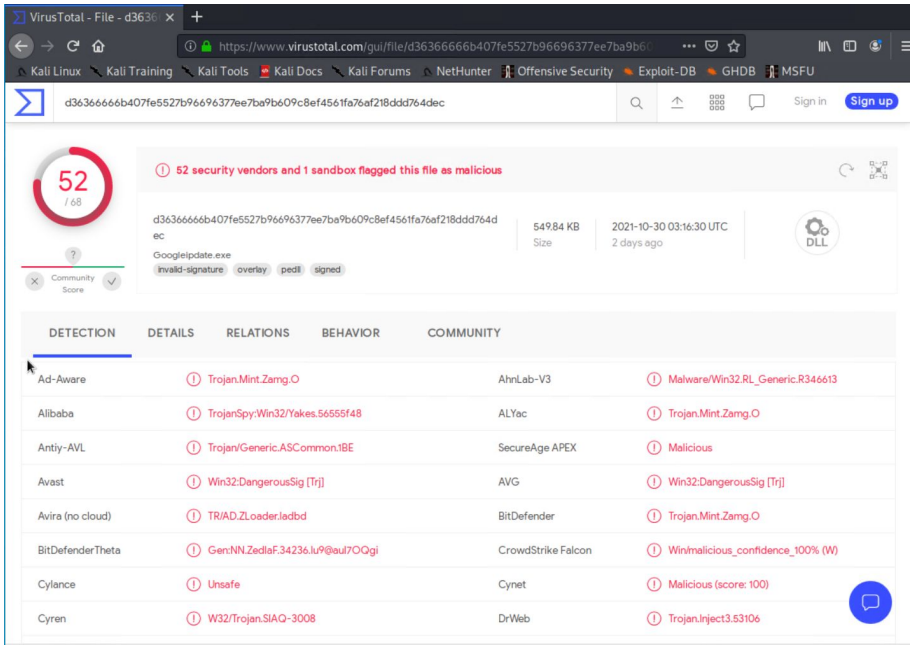


Malicious Activity

Malware Infection

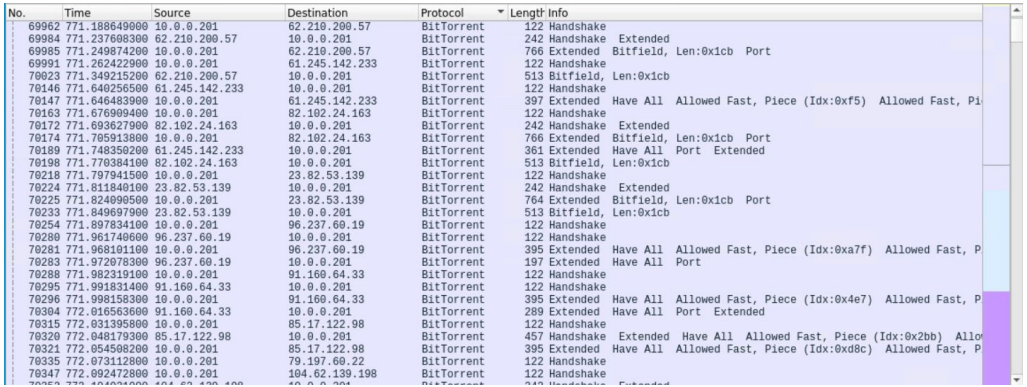
- Malicious file download “june11.dll”



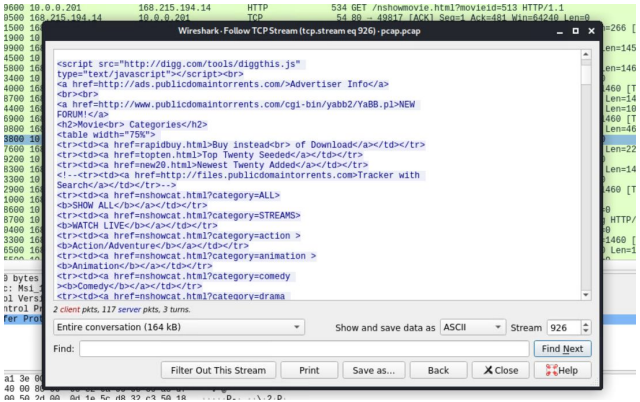


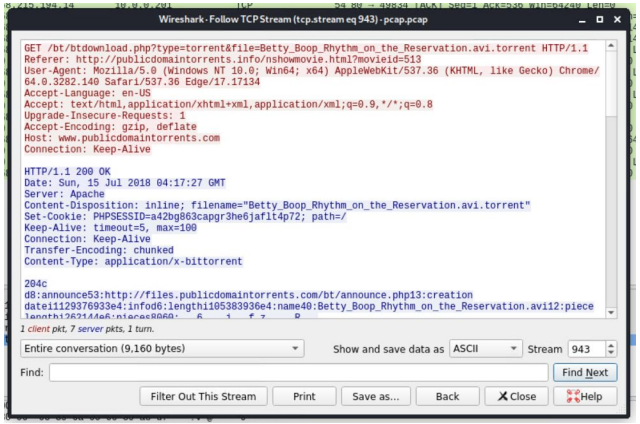
Illegal File Downloads

- Browsing for copyright material on public bittorrent tracker
- Downloading and fetching torrent data



No.	Time	Source	Destination	Protocol	Length	Info
69962	771.188649000	10.0.0.201	62.210.200.57	BitTorrent	122	Handshake
69984	771.237608300	62.210.200.57	10.0.0.201	BitTorrent	242	Handshake Extended
69985	771.249874200	10.0.0.201	62.210.200.57	BitTorrent	766	Extended Bitfield, Len:0x1cb Port
69991	771.262422900	10.0.0.201	61.245.142.233	BitTorrent	122	Handshake
70023	771.349215200	62.210.200.57	10.0.0.201	BitTorrent	513	Bitfield, Len:0x1cb
70146	771.640256500	61.245.142.233	10.0.0.201	BitTorrent	122	Handshake
70147	771.646483900	10.0.0.201	61.245.142.233	BitTorrent	397	Extended Have All Allowed Fast, Piece (Idx:0xf5) Allowed Fast, Pi
70163	771.676909400	10.0.0.201	82.102.24.163	BitTorrent	122	Handshake
70172	771.693627900	82.102.24.163	10.0.0.201	BitTorrent	242	Handshake Extended
70174	771.705913800	10.0.0.201	82.102.24.163	BitTorrent	766	Extended Bitfield, Len:0x1cb Port
70189	771.748350200	61.245.142.233	10.0.0.201	BitTorrent	361	Extended Have All Port Extended
70198	771.770384100	82.102.24.163	10.0.0.201	BitTorrent	513	Bitfield, Len:0x1cb
70218	771.797941500	10.0.0.201	23.82.53.139	BitTorrent	122	Handshake
70224	771.811840100	23.82.53.139	10.0.0.201	BitTorrent	242	Handshake Extended
70225	771.824090500	10.0.0.201	23.82.53.139	BitTorrent	764	Extended Bitfield, Len:0x1cb Port
70233	771.849697900	23.82.53.139	10.0.0.201	BitTorrent	513	Bitfield, Len:0x1cb
70254	771.897834100	10.0.0.201	96.237.60.19	BitTorrent	122	Handshake
70280	771.961740600	96.237.60.19	10.0.0.201	BitTorrent	122	Handshake
70281	771.968101100	10.0.0.201	96.237.60.19	BitTorrent	395	Extended Have All Allowed Fast, Piece (Idx:0xa7f) Allowed Fast, P
70283	771.972078300	96.237.60.19	10.0.0.201	BitTorrent	197	Extended Have All Port
70288	771.982319100	10.0.0.201	91.160.64.33	BitTorrent	122	Handshake
70295	771.991831400	91.160.64.33	10.0.0.201	BitTorrent	122	Handshake
70296	771.998158300	10.0.0.201	91.160.64.33	BitTorrent	395	Extended Have All Allowed Fast, Piece (Idx:0x4e7) Allowed Fast, P
70304	772.016563600	91.160.64.33	10.0.0.201	BitTorrent	289	Extended Have All Port Extended
70315	772.031395800	10.0.0.201	85.17.122.98	BitTorrent	122	Handshake
70320	772.048179300	85.17.122.98	10.0.0.201	BitTorrent	457	Handshake Extended Have All Allowed Fast, Piece (Idx:0x2bb) Allo
70321	772.054508200	10.0.0.201	85.17.122.98	BitTorrent	395	Extended Have All Allowed Fast, Piece (Idx:0xd8c) Allowed Fast, P
70335	772.073112800	10.0.0.201	79.197.60.22	BitTorrent	122	Handshake
70347	772.092472800	10.0.0.201	104.62.139.198	BitTorrent	122	Handshake





The End