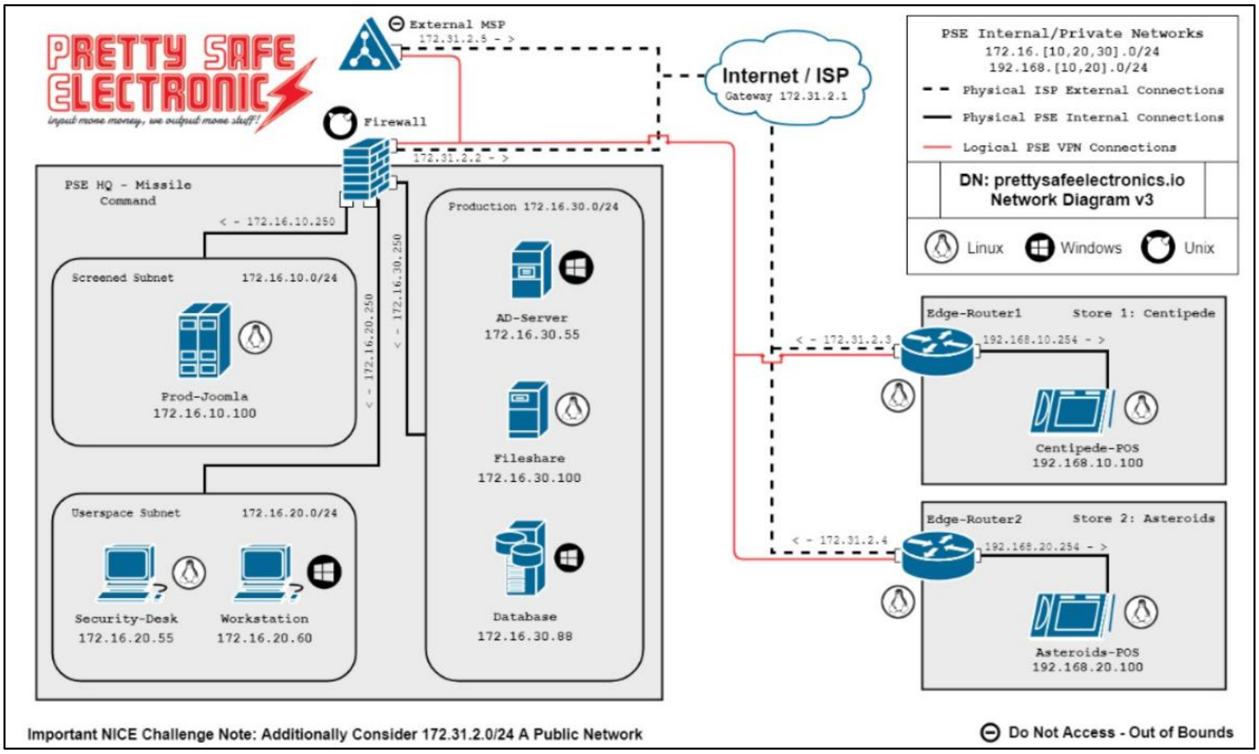
# NICE Challenge Labs

**Portfolio Samples**

Will Kittredge

## TABLE OF CONTENTS

## LAB 7: NETWORKING ANOMALIES

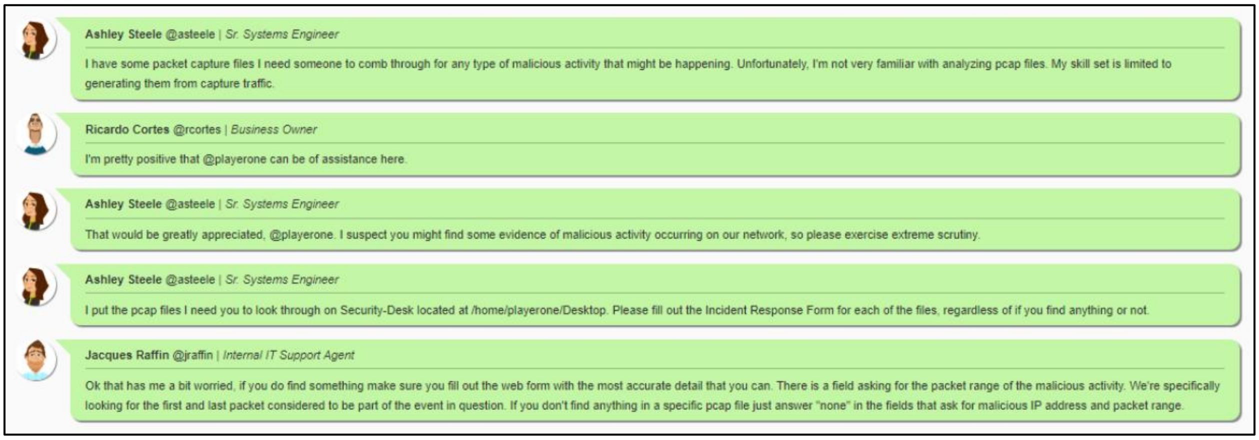


## EXECUTIVE SUMMARY

In this lab, we were presented with more packet capture files to analyze. Keeping with some of the previous labs, we were asked to identify activity within the trace that could be considered malicious by providing a suspect IP address and packet range. Unlike previous labs, however, we were not asked to identify which type of malicious activity was present in the capture. This isn't to say that we excluded the malicious activity type from consideration as part of our analysis, though. For example, one of the first things we noticed in the `workstation.pcap` file was a large number of SYN packets initiated by the same IP address. All of these packets were headed to the same destination IP address, but with different port numbers. Based on this information, we determined that a port scan was occurring. Knowing (or making an informed guess) at the malicious activity type helped us assess whether our suspect IP address/packet range made sense in the context of all available information, and it gave us an idea of what we might expect when analyzing the next trace file. Combined with high-level information (like conversations and endpoint statistics; i.e., “how many conversations are happening, what ports, protocols, and IP addresses are they associated with?”), this allowed us to identify suspicious activity with relative speed.

Our recommendations also remain in line with previous labs. The company should implement an automated monitoring system with intrusion detection/prevention capabilities. Some types of malicious activity (the suspected port scan is a prime example) would be relatively easy to detect automatically, and it’s probably not feasible in the long run to analyze every occurrence.

## CHAT LOGS



**Ashley Steele** @asteele | *Sr. Systems Engineer*

I have some packet capture files I need someone to comb through for any type of malicious activity that might be happening. Unfortunately, I'm not very familiar with analyzing pcap files. My skill set is limited to generating them from capture traffic.

**Ricardo Cortes** @rcortes | *Business Owner*

I'm pretty positive that @playerone can be of assistance here.

**Ashley Steele** @asteele | *Sr. Systems Engineer*

That would be greatly appreciated, @playerone. I suspect you might find some evidence of malicious activity occurring on our network, so please exercise extreme scrutiny.

**Ashley Steele** @asteele | *Sr. Systems Engineer*

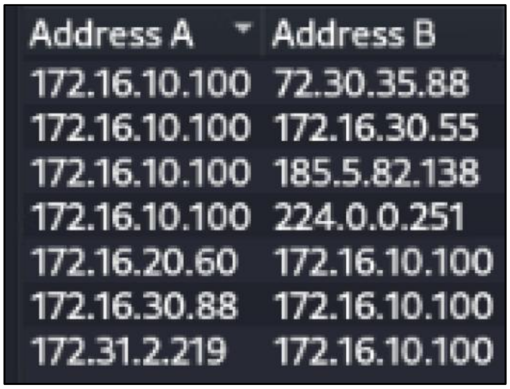
I put the pcap files I need you to look through on Security-Desk located at /home/playerone/Desktop. Please fill out the Incident Response Form for each of the files, regardless of if you find anything or not.

**Jacques Raffin** @jraffin | *Internal IT Support Agent*

Ok that has me a bit worried, if you do find something make sure you fill out the web form with the most accurate detail that you can. There is a field asking for the packet range of the malicious activity. We're specifically looking for the first and last packet considered to be part of the event in question. If you don't find anything in a specific pcap file just answer "none" in the fields that ask for malicious IP address and packet range.

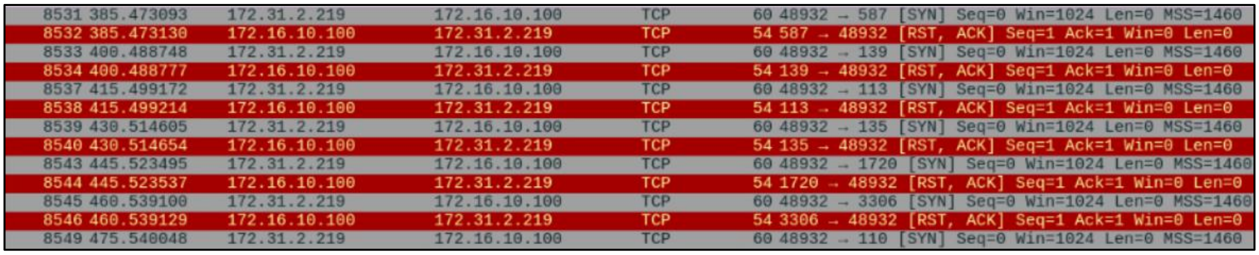
## APPROACH, STEPS, & ACTIONS TAKEN

To begin, we took a look at the `prodjoomla.pcap` file and decided to head straight to the conversations tab to see if any IPs jumped out at as. After comparing it with the network map, we noticed that one of the IPs (`172.31.2.219`) was an external IP, which raised a few red flags for us.



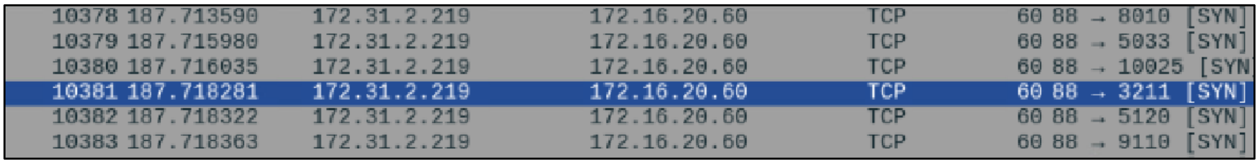
| Address A | Address B |
|---|---|
| 172.16.10.100 | 72.30.35.88 |
| 172.16.10.100 | 172.16.30.55 |
| 172.16.10.100 | 185.5.82.138 |
| 172.16.10.100 | 224.0.0.251 |
| 172.16.20.60 | 172.16.10.100 |
| 172.16.30.88 | 172.16.10.100 |
| 172.31.2.219 | 172.16.10.100 |

We decided to add this IP as a display filter so we could see what activity was associated with it in the trace file. We used the `ip.addr == 172.31.2.219` as our display filter, which led to a series of TCP SYN packets being sent from the malicious IP to `172.16.10.100`. This led us to believe that either port scanning or a SYN flood attack was occurring.



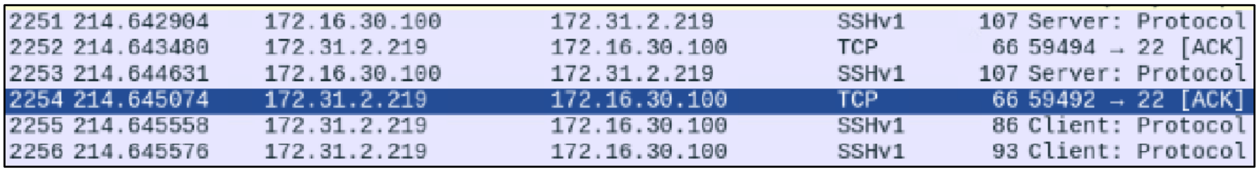
```
8531 385.473093   172.31.2.219     172.16.10.100    TCP     60 48932 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8532 385.473130   172.16.10.100    172.31.2.219     TCP     54 587 → 48932 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8533 400.488748   172.31.2.219     172.16.10.100    TCP     60 48932 → 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8534 400.488777   172.16.10.100    172.31.2.219     TCP     54 139 → 48932 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8537 415.499172   172.31.2.219     172.16.10.100    TCP     60 48932 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8538 415.499214   172.16.10.100    172.31.2.219     TCP     54 113 → 48932 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8539 430.514605   172.31.2.219     172.16.10.100    TCP     60 48932 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8540 430.514654   172.16.10.100    172.31.2.219     TCP     54 135 → 48932 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8543 445.523495   172.31.2.219     172.16.10.100    TCP     60 48932 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8544 445.523537   172.16.10.100    172.31.2.219     TCP     54 1720 → 48932 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8545 460.539100   172.31.2.219     172.16.10.100    TCP     60 48932 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8546 460.539129   172.16.10.100    172.31.2.219     TCP     54 3306 → 48932 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8549 475.540048   172.31.2.219     172.16.10.100    TCP     60 48932 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

Next, we took a look at the `workstation.pcap` file, where we employed the same strategy of going straight to the conversations screen to see if anything jumped out. We found the exact same malicious IP, so we applied the same display filter and began looking through the packets. After looking at the malicious packets, we determined that the malicious IP was most likely port scanning the target `172.16.20.60`, as a series of SYN packets were being sent to many different ports.



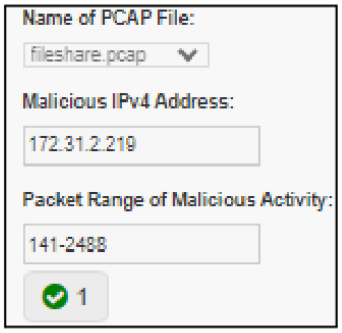
```
10378 187.713590   172.31.2.219     172.16.20.60     TCP       60 88 → 8010 [SYN]
10379 187.715980   172.31.2.219     172.16.20.60     TCP       60 88 → 5033 [SYN]
10380 187.716035   172.31.2.219     172.16.20.60     TCP       60 88 → 10025 [SYN
10381 187.718281   172.31.2.219     172.16.20.60     TCP       60 88 → 3211 [SYN]
10382 187.718322   172.31.2.219     172.16.20.60     TCP       60 88 → 5120 [SYN]
10383 187.718363   172.31.2.219     172.16.20.60     TCP       60 88 → 9110 [SYN]
```

Finally, we moved onto the `fileshare.pcap` file. We applied the same display filter as the previous two files and began digging. The beginning of the trace is filled with TCP SYN packets being sent from the malicious IP to a number of different ports on the target machine, but this changes later into the capture. It appears that an SSH connection began between the malicious IP and the target machine, however, this was all we were able to determine when looking at the trace.



```
2251 214.642904   172.16.30.100    172.31.2.219     SSHv1    107 Server: Protocol
2252 214.643480   172.31.2.219     172.16.30.100    TCP       66 59494 → 22 [ACK]
2253 214.644631   172.16.30.100    172.31.2.219     SSHv1    107 Server: Protocol
2254 214.645074   172.31.2.219     172.16.30.100    TCP       66 59492 → 22 [ACK]
2255 214.645558   172.31.2.219     172.16.30.100    SSHv1     86 Client: Protocol
2256 214.645576   172.31.2.219     172.16.30.100    SSHv1     93 Client: Protocol
```

## ANSWERS SUBMITTED



Name of PCAP File:

fileshare.pcap

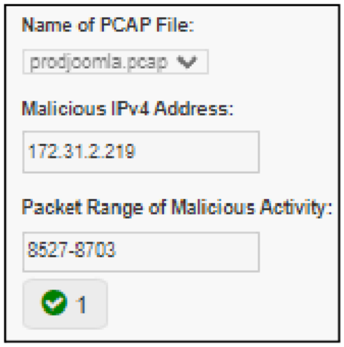
Malicious IPv4 Address:

172.31.2.219

Packet Range of Malicious Activity:

141-2488

✓ 1



Name of PCAP File:

prodjoomla.pcap

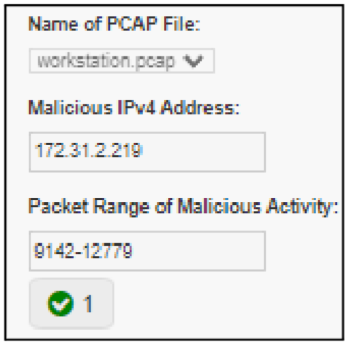
Malicious IPv4 Address:

172.31.2.219

Packet Range of Malicious Activity:

8527-8703

✓ 1



Name of PCAP File:

workstation.pcap

Malicious IPv4 Address:

172.31.2.219

Packet Range of Malicious Activity:

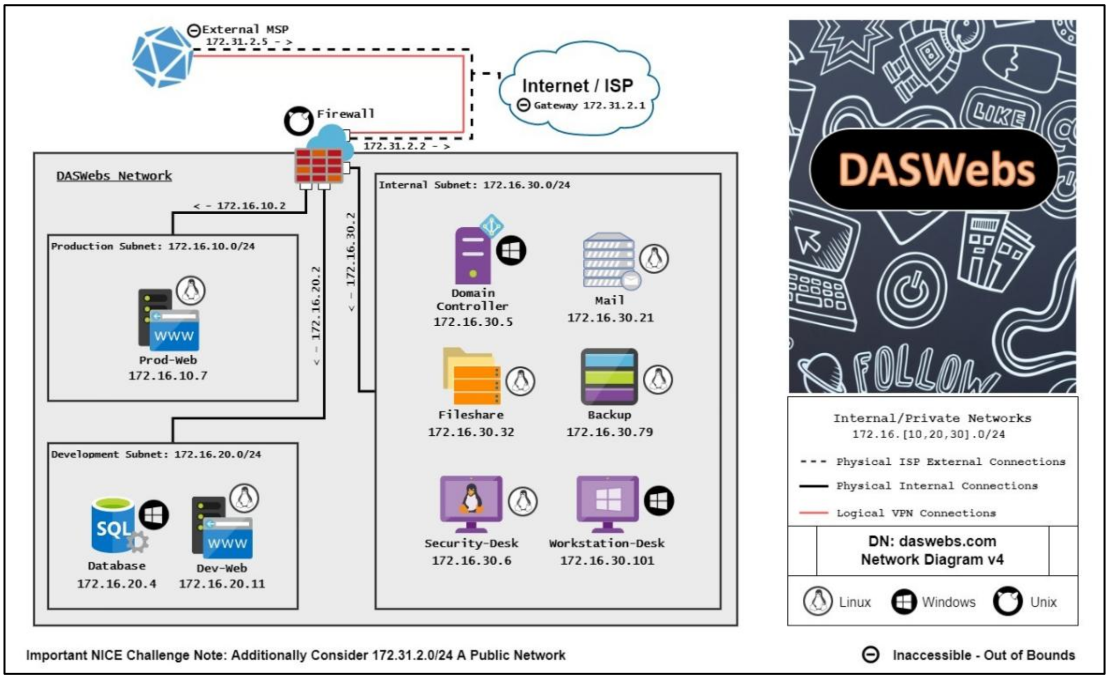
9142-12779

✓ 1

## CONCLUSIONS & RECOMMENDED ACTION(S)

We were able to quickly draw our conclusion on which IP was malicious after using the strategy that we described. Because it was the only external IP listen in the conversations screen, we were able to locate malicious activity much faster than in previous weeks. Our recommendation for the company to install an automated detection/prevention system to track and prevent attacks while they're happening rather than analyzing network traffic after the fact.

## LAB 9: LENGTHY LOGS ATTACK ANALYSIS

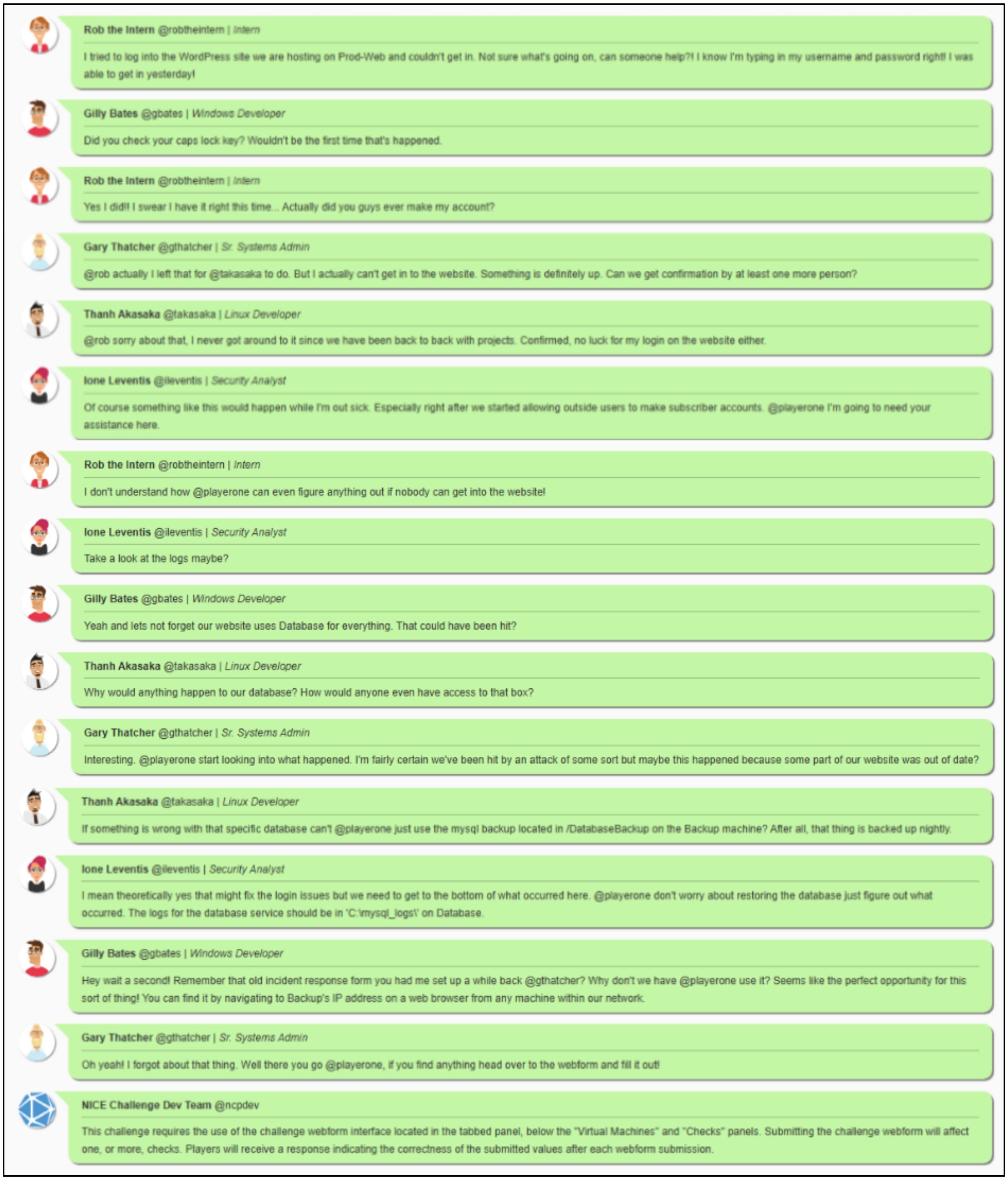


## EXECUTIVE SUMMARY

In this scenario, we are told that multiple users are experiencing an issue with logging into a WordPress site and that we need to look through the site's database logs and backup to find the root cause of these problems. Our first step is to find the most recent log file and do an initial search for anything that is obviously out-of-place. After that, the plan is to move the files to the Security-Desk system and analyze them there, hopefully finding evidence of tables being changed via commands like `INSERT`, `DROP`, `UPDATE`, or `DELETE`.

Following this procedure, we located what we believe is anomalous activity in the database log file that involves several user accounts. Based on our understanding of the SQL log and database backup files, it appears that at least four accounts were modified or deleted. We would recommend that the company take steps to harden their database and WordPress site, which ideally will mitigate or prevent future tampering.

## CHAT LOGS



**Rob the Intern** @robtheintern | *Intern*

I tried to log into the WordPress site we are hosting on Prod-Web and couldn't get in. Not sure what's going on, can someone help?! I know I'm typing in my username and password right! I was able to get in yesterday!

**Gilly Bates** @gbates | *Windows Developer*

Did you check your caps lock key? Wouldn't be the first time that's happened.

**Rob the Intern** @robtheintern | *Intern*

Yes I did!! I swear I have it right this time... Actually did you guys ever make my account?

**Gary Thatcher** @gthatcher | *Sr. Systems Admin*

@rob actually I left that for @takasaka to do. But I actually can't get in to the website. Something is definitely up. Can we get confirmation by at least one more person?

**Thanh Akasaka** @takasaka | *Linux Developer*

@rob sorry about that, I never got around to it since we have been back to back with projects. Confirmed, no luck for my login on the website either.

**Ione Leventis** @ileventis | *Security Analyst*

Of course something like this would happen while I'm out sick. Especially right after we started allowing outside users to make subscriber accounts. @playerone I'm going to need your assistance here.

**Rob the Intern** @robtheintern | *Intern*

I don't understand how @playerone can even figure anything out if nobody can get into the website!

**Ione Leventis** @ileventis | *Security Analyst*

Take a look at the logs maybe?

**Gilly Bates** @gbates | *Windows Developer*

Yeah and lets not forget our website uses Database for everything. That could have been hit?

**Thanh Akasaka** @takasaka | *Linux Developer*

Why would anything happen to our database? How would anyone even have access to that box?

**Gary Thatcher** @gthatcher | *Sr. Systems Admin*

Interesting. @playerone start looking into what happened. I'm fairly certain we've been hit by an attack of some sort but maybe this happened because some part of our website was out of date?

**Thanh Akasaka** @takasaka | *Linux Developer*

If something is wrong with that specific database can't @playerone just use the mysql backup located in /DatabaseBackup on the Backup machine? After all, that thing is backed up nightly.

**Ione Leventis** @ileventis | *Security Analyst*

I mean theoretically yes that might fix the login issues but we need to get to the bottom of what occurred here. @playerone don't worry about restoring the database just figure out what occurred. The logs for the database service should be in 'C:\mysql_logs\' on Database.

**Gilly Bates** @gbates | *Windows Developer*

Hey wait a second! Remember that old incident response form you had me set up a while back @gthatcher? Why don't we have @playerone use it? Seems like the perfect opportunity for this sort of thing! You can find it by navigating to Backup's IP address on a web browser from any machine within our network.

**Gary Thatcher** @gthatcher | *Sr. Systems Admin*

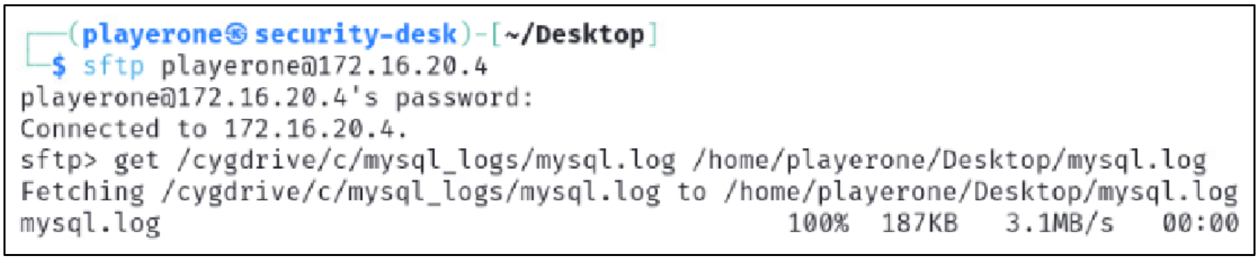
Oh yeah! I forgot about that thing. Well there you go @playerone, if you find anything head over to the webform and fill it out!

**NICE Challenge Dev Team** @ncpdev

This challenge requires the use of the challenge webform interface located in the tabbed panel, below the "Virtual Machines" and "Checks" panels. Submitting the challenge webform will affect one, or more, checks. Players will receive a response indicating the correctness of the submitted values after each webform submission.

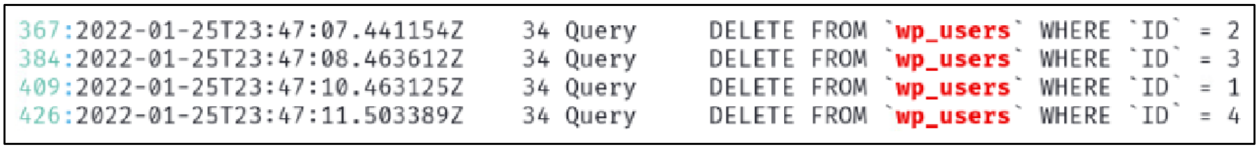
## APPROACH, STEPS, & ACTIONS TAKEN

Our first step was to orient ourselves to the situation by reviewing all of the provided information (meeting notes, documentation, network map, and incident response form). This ensured that we understood what we were being asked to find. When we felt ready, we began our analysis by finding, copying, and then opening the `mysql.log` file on the Database system. We quickly realized that the notepad would not be a good tool to use for analyzing this log, so we pulled the file to the Kali system and performed all of our analysis there.



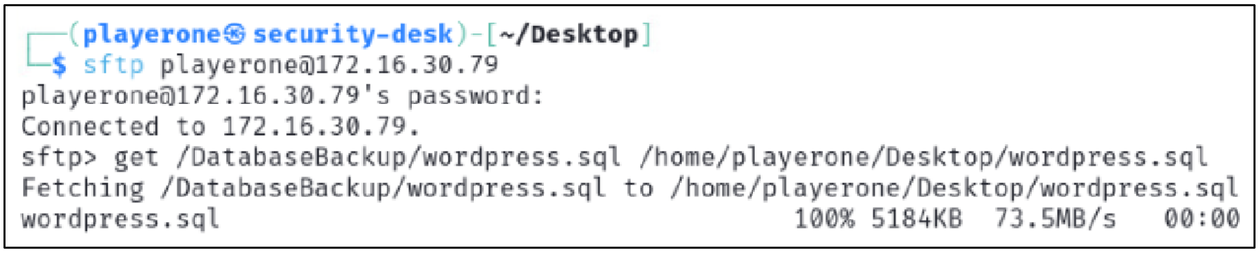
```
┌──(playerone㉿security-desk)-[~/Desktop]
└─$ sftp playerone@172.16.20.4
playerone@172.16.20.4's password:
Connected to 172.16.20.4.
sftp> get /cygdrive/c/mysql_logs/mysql.log /home/playerone/Desktop/mysql.log
Fetching /cygdrive/c/mysql_logs/mysql.log to /home/playerone/Desktop/mysql.log
mysql.log                                      100%  187KB   3.1MB/s   00:00
```

Knowing that we were looking for evidence of user account tempering in a database log file, we came up with some `grep` patterns that could help us locate the relevant information in the file. We began with a basic `grep user mysql.log` command and scrolled through the output to look for anything interesting that involved a user account being changed. It was possible to glean some information about the database structure based on the fog file. For example, a logged `SELECT * FROM wp_users WHERE user_login = ‘admin’ LIMIT 1` query told us that there is probably a table called `wp_users` that scores account information in the database. To look for evidence of user account tampering (i.e., changes to the `wp_users` table), we used `cat mysql.log | grep -v SELECT | grep -n wp_users`. Immediately, we could see that four users were deleted from the table.

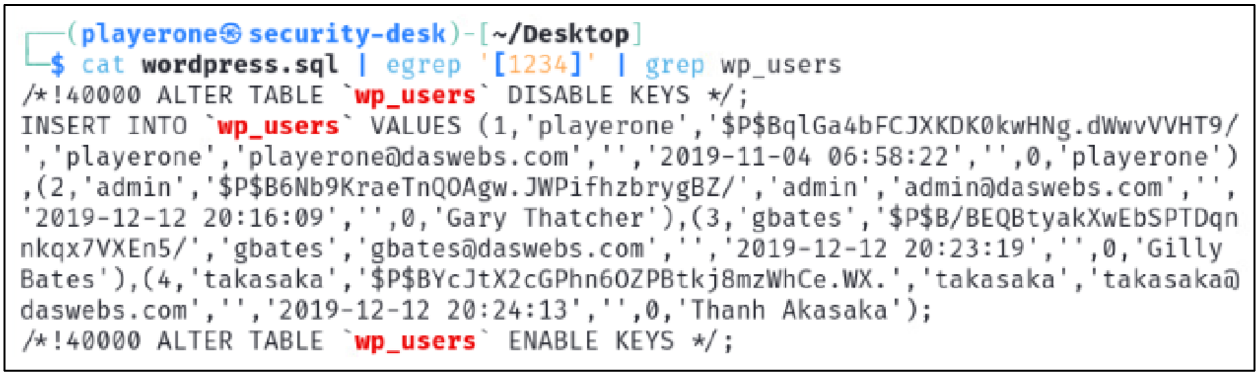


```
367:2022-01-25T23:47:07.441154Z      34 Query     DELETE FROM `wp_users` WHERE `ID` = 2
384:2022-01-25T23:47:08.463612Z      34 Query     DELETE FROM `wp_users` WHERE `ID` = 3
409:2022-01-25T23:47:10.463125Z      34 Query     DELETE FROM `wp_users` WHERE `ID` = 1
426:2022-01-25T23:47:11.503389Z      34 Query     DELETE FROM `wp_users` WHERE `ID` = 4
```

However, our searches of the log did not find the usernames that were associated with the ID numbers of the users that got deleted. Realizing that the `mysql.log` file might not have all of the information that we needed, we got a copy of the database backup file that the meeting notes mentioned. Once we had `wordpress.sql`, it was relatively easy to locate the usernames that were associated with the IDs that we found by searching for the `wp_users` table.

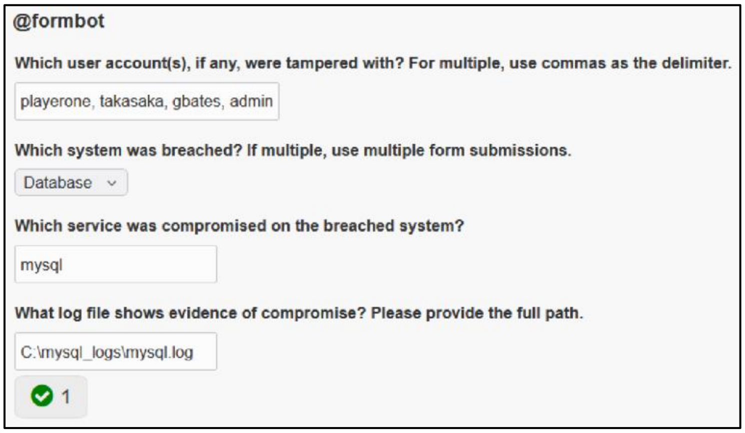


```
┌──(playerone㉿security-desk)-[~/Desktop]
└─$ sftp playerone@172.16.30.79
playerone@172.16.30.79's password:
Connected to 172.16.30.79.
sftp> get /DatabaseBackup/wordpress.sql /home/playerone/Desktop/wordpress.sql
Fetching /DatabaseBackup/wordpress.sql to /home/playerone/Desktop/wordpress.sql
wordpress.sql                                     100% 5184KB  73.5MB/s   00:00
```



```
┌──(playerone㉿security-desk)-[~/Desktop]
└─$ cat wordpress.sql | egrep '[1234]' | grep wp_users
/*!40000 ALTER TABLE `wp_users` DISABLE KEYS */;
INSERT INTO `wp_users` VALUES (1,'playerone','$P$BqlGa4bFCJXKDK0kwHNg.dWwvVVHT9/
','playerone','playerone@daswebs.com','','2019-11-04 06:58:22','',0,'playerone')
,(2,'admin','$P$B6Nb9KraeTnQOAgw.JWPifhzbrygBZ/','admin','admin@daswebs.com','',
'2019-12-12 20:16:09','',0,'Gary Thatcher'),(3,'gbates','$P$B/BEQBtyakXwEbSPTDqn
nkqx7VXEn5/','gbates','gbates@daswebs.com','','2019-12-12 20:23:19','',0,'Gilly
Bates'),(4,'takasaka','$P$BYcJtX2cGPhn6OZPBtkj8mzWhCe.WX.','takasaka','takasaka@
daswebs.com','','2019-12-12 20:24:13','',0,'Thanh Akasaka');
/*!40000 ALTER TABLE `wp_users` ENABLE KEYS */;
```

## ANSWERS SUBMITTED



**@formbot**

**Which user account(s), if any, were tampered with? For multiple, use commas as the delimiter.**

playerone, takasaka, gbates, admin

**Which system was breached? If multiple, use multiple form submissions.**

Database

**Which service was compromised on the breached system?**

mysql

**What log file shows evidence of compromise? Please provide the full path.**

C:\mysql_logs\mysql.log

✓ 1

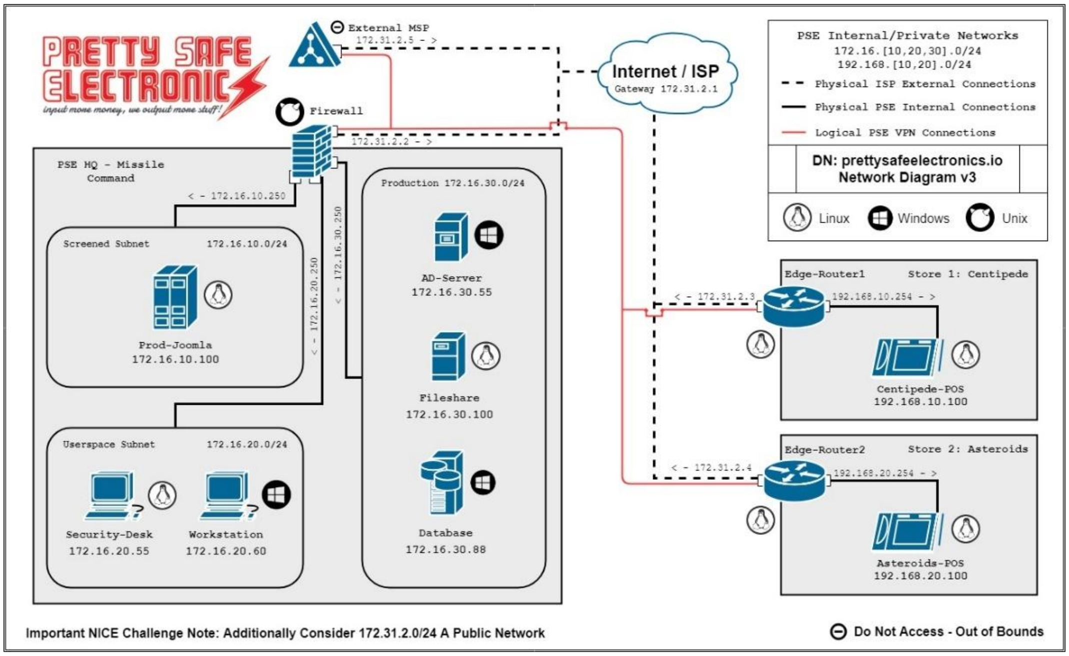
We submitted `playerone`, `takasaka`, `gbates`, and `admin` as the user accounts that were tampered with. The remaining three answers in the form were given in the provided lab information.

## CONCLUSIONS & RECOMMENDED ACTION(S)

Although we are not familiar enough with WordPress or databases to understand the activity precisely, we believe that what we found in the log is at least anomalous. Based on context provided in the meeting information, it was definitely unauthorized and unexpected. We recommend that the company harden their database and WordPress site to prevent/mitigate this kind of activity moving forward.

In the `mysql.log` file, there were multiple lines referencing a connection by the `playerone` account. If the `playerone` account is reusing `password123` as their password, then perhaps an unauthorized actor was able to leverage this to access the database. If the company chooses to restore their database from a backup (which it mentions are taken nightly), then it should ensure that the backup it restores is truly “clean” (i.e., doesn’t contain any previously-undetected anomalous activity) before bringing the system fully up again.

## LAB 11: DISASTROUS DNS DESTRUCTION

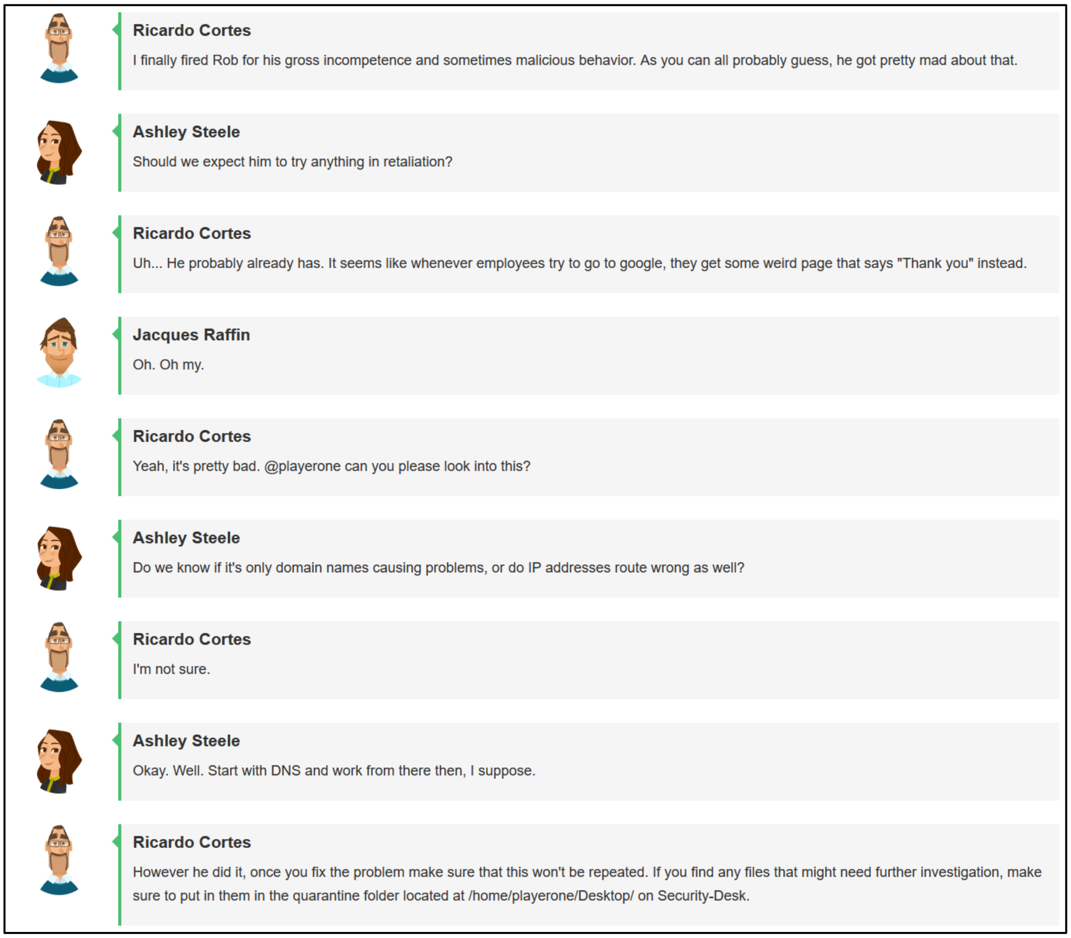


### EXECUTIVE SUMMARY

In this scenario, we were asked to resolve a DNS issue caused by an insider threat. An employee had been fired due to incompetence and malicious activity, and in retaliation, this employee caused a DNS issue. Employees headed to `google.com` were instead sent to a strange "Thank you" page that initiated the download of a suspicious executable (`FastCleanup.exe`).

We decided that a good initial approach would be to confirm the issue and then investigate further from there. After visiting the webpage to confirm the issue, we used the terminal to find the IP addresses of the webpage and our DNS server and then connected to those machines. The issue was fixed by removing `FastCleanup.exe` from the fileserver, hardening the DNS server, and clearing the DNS server's cache. We recommend that the company ensures that their DNS server remains hardened, adequate backup/recovery procedures are in place, and that automatic alerts be configured to alert analysts about similar issues in the future.

## CHAT LOGS



**Ricardo Cortes**
I finally fired Rob for his gross incompetence and sometimes malicious behavior. As you can all probably guess, he got pretty mad about that.

**Ashley Steele**
Should we expect him to try anything in retaliation?

**Ricardo Cortes**
Uh... He probably already has. It seems like whenever employees try to go to google, they get some weird page that says "Thank you" instead.

**Jacques Raffin**
Oh. Oh my.

**Ricardo Cortes**
Yeah, it's pretty bad. @playerone can you please look into this?

**Ashley Steele**
Do we know if it's only domain names causing problems, or do IP addresses route wrong as well?

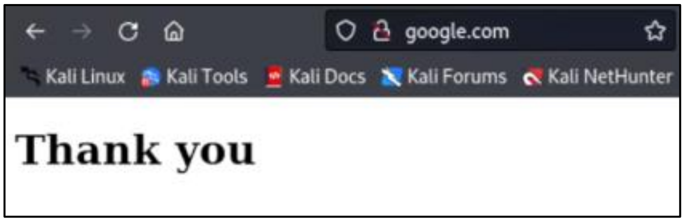
**Ricardo Cortes**
I'm not sure.

**Ashley Steele**
Okay. Well. Start with DNS and work from there then, I suppose.

**Ricardo Cortes**
However he did it, once you fix the problem make sure that this won't be repeated. If you find any files that might need further investigation, make sure to put in them in the quarantine folder located at /home/playerone/Desktop/ on Security-Desk.

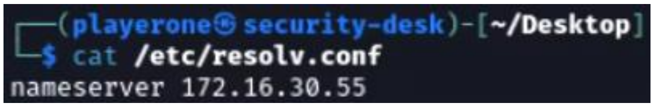
## APPROACH, STEPS, & ACTIONS TAKEN

Our first step was to read through all of the provided information and make an appropriate plan of action. Because we were uncertain about the exact nature of the problem, we decided that a good initial step would be to confirm the issue and then investigate further from there. After confirming the issue, we stored the downloaded executable in the quarantine folder.

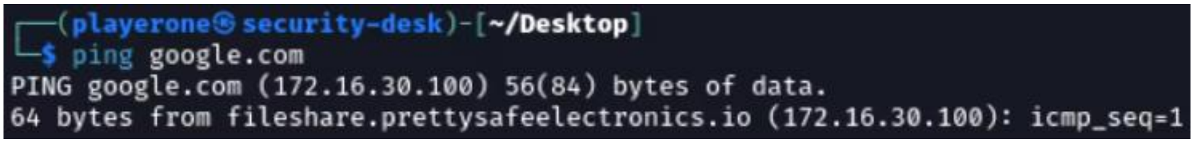


Thank you

Next, we used the terminal to find the IP addresses of the organization’s DNS server and of the webpage that it resolved for `google.com`.

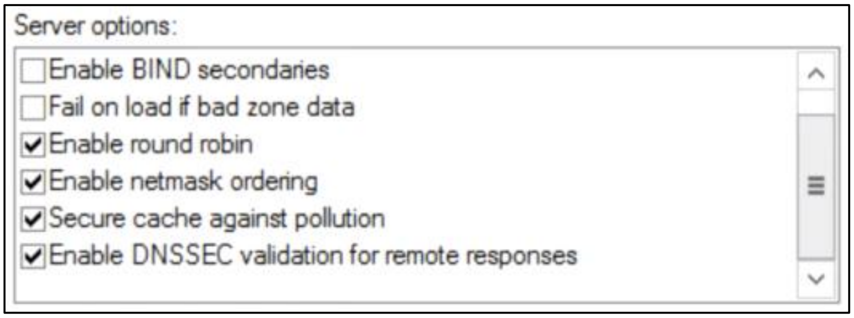


```
┌──(playerone㉿security-desk)-[~/Desktop]
└─$ cat /etc/resolv.conf
nameserver 172.16.30.55
```



```
┌──(playerone㉿security-desk)-[~/Desktop]
└─$ ping google.com
PING google.com (172.16.30.100) 56(84) bytes of data.
64 bytes from fileshare.prettysafeelectronics.io (172.16.30.100): icmp_seq=1
```

From here, we connected to both machines to make the necessary changes. On the `172.16.30.100` machine, we made a backup of the `/var/www` directory, and then deleted the entire contents of the original copy (`sudo cp -r /var/www /var/www.bak` and `sudo rm -rf /var/www/*`). On the `172.16.30.55` machine, we used the DNS tool in the Server Manager to make the necessary changes to the DNS server. We modified the server options to secure the cache against pollution and enabled DNSSEC validation (Action > Properties > Advanced tab > Check the relevant options).

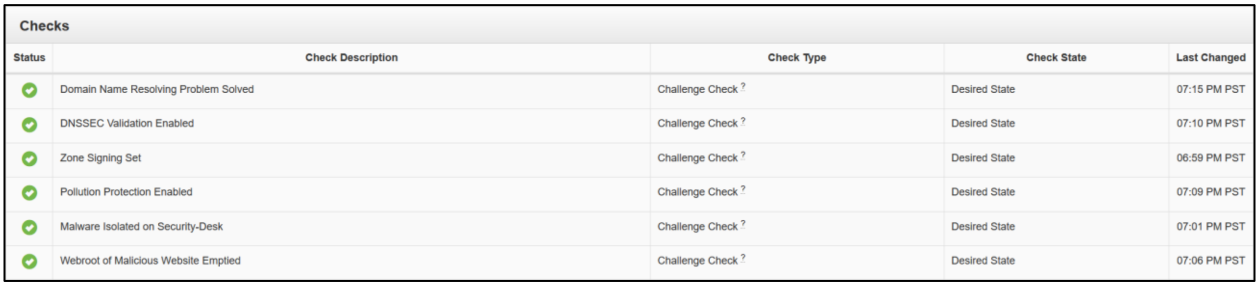


Server options:

☐ Enable BIND secondaries
☐ Fail on load if bad zone data
☑ Enable round robin
☑ Enable netmask ordering
☑ Secure cache against pollution
☑ Enable DNSSEC validation for remote responses

Next, we signed the `prettysafeelectronics.io` Forward Lookup Zone using the signing wizard with the default configuration options (expand Forward Lookup Zones > right-click `prettysafeelectronics.io` > DNSSEC > Sign the Zone). Lastly, we cleared the DNS server’s cache (Action > Clear Cache) to complete the last lab check.

## ANSWERS SUBMITTED



Checks

| Status | Check Description | Check Type | Check State | Last Changed |
|---|---|---|---|---|
| ✓ | Domain Name Resolving Problem Solved | Challenge Check ? | Desired State | 07:15 PM PST |
| ✓ | DNSSEC Validation Enabled | Challenge Check ? | Desired State | 07:10 PM PST |
| ✓ | Zone Signing Set | Challenge Check ? | Desired State | 06:59 PM PST |
| ✓ | Pollution Protection Enabled | Challenge Check ? | Desired State | 07:09 PM PST |
| ✓ | Malware Isolated on Security-Desk | Challenge Check ? | Desired State | 07:01 PM PST |
| ✓ | Webroot of Malicious Website Emptied | Challenge Check ? | Desired State | 07:06 PM PST |

## CONCLUSIONS & RECOMMENDED ACTION(S)

Although we completed the steps necessary to resolve the issue, we are still uncertain about the exact details of the problem. However, we believe that the technical reason for the DNS issue is likely related to cache pollution. For one, we did not find any explicit misconfigurations in the DNS server (as far as the records themselves go). This theory is also supported by the fact that the DNS issue was resolved after we enabled pollution protection and cleared the cache.

As part of the lab, we were asked to fix the problem in such a way that it wouldn't occur again. We have excluded those steps from this recommendations section to reduce redundancy in our report – however, for the sake of clarity, we do recommend that the DNS security settings we enabled be set. Apart from those security controls, we think it would be a good idea for the organization to have a backup and recovery plan/procedure for their DNS infrastructure. Depending on the nature of an incident, it might be easier to recover by restoring a backup or failing over to a backup DNS server. It might also be wise to have automatic checks in place for important or commonly accessed websites so that an analyst or administrator is alerted immediately when something isn't being resolved correctly.