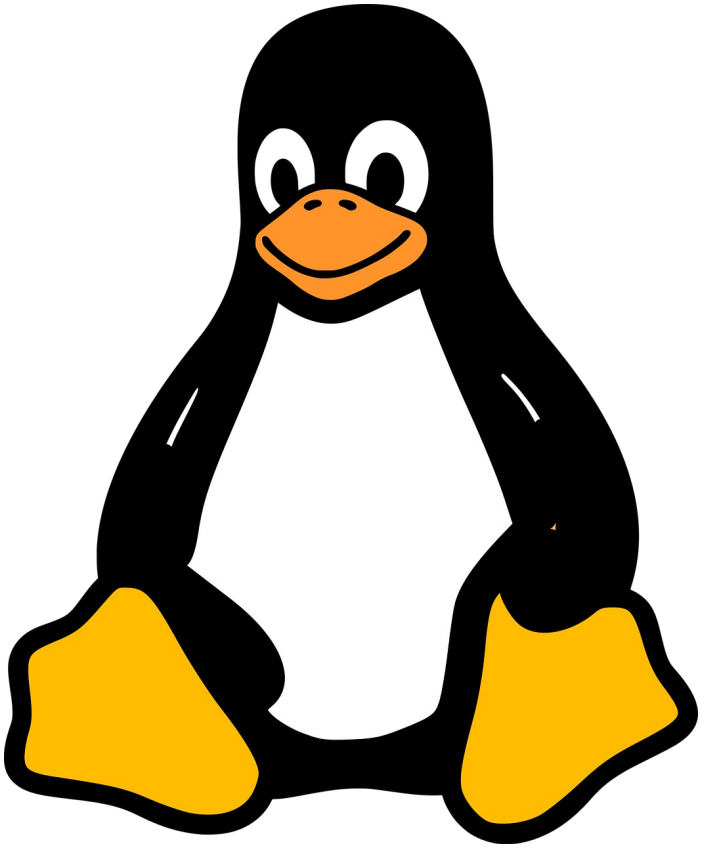
HackerFrogs Afterschool
Linux 1: Basic Navigation

Class:
Linux OS Operations

Workshop Number:
AS-LIN-01

Document Version:
1.75

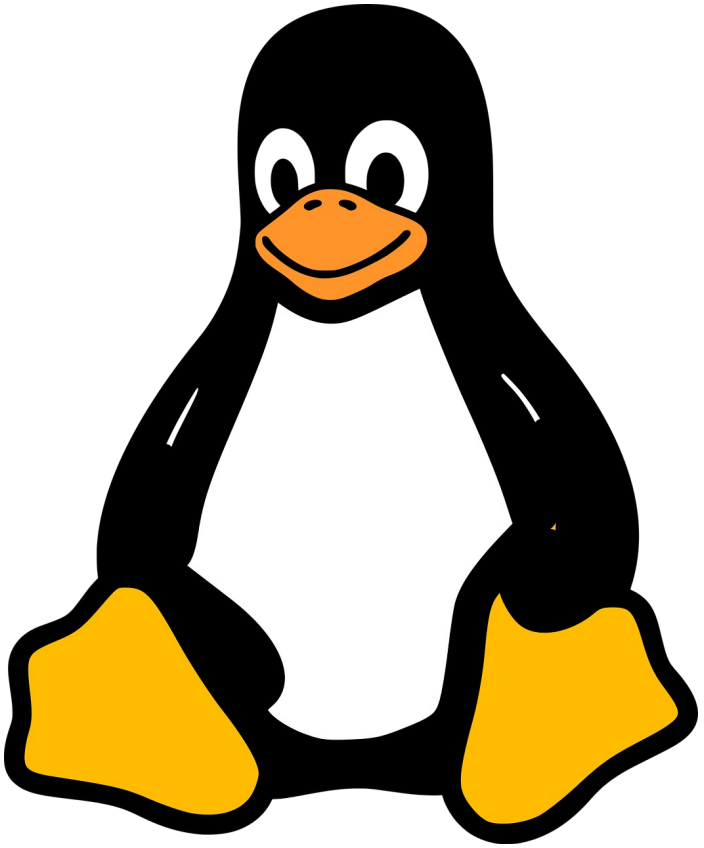
Special Requirements:
None



Linux OS Operations

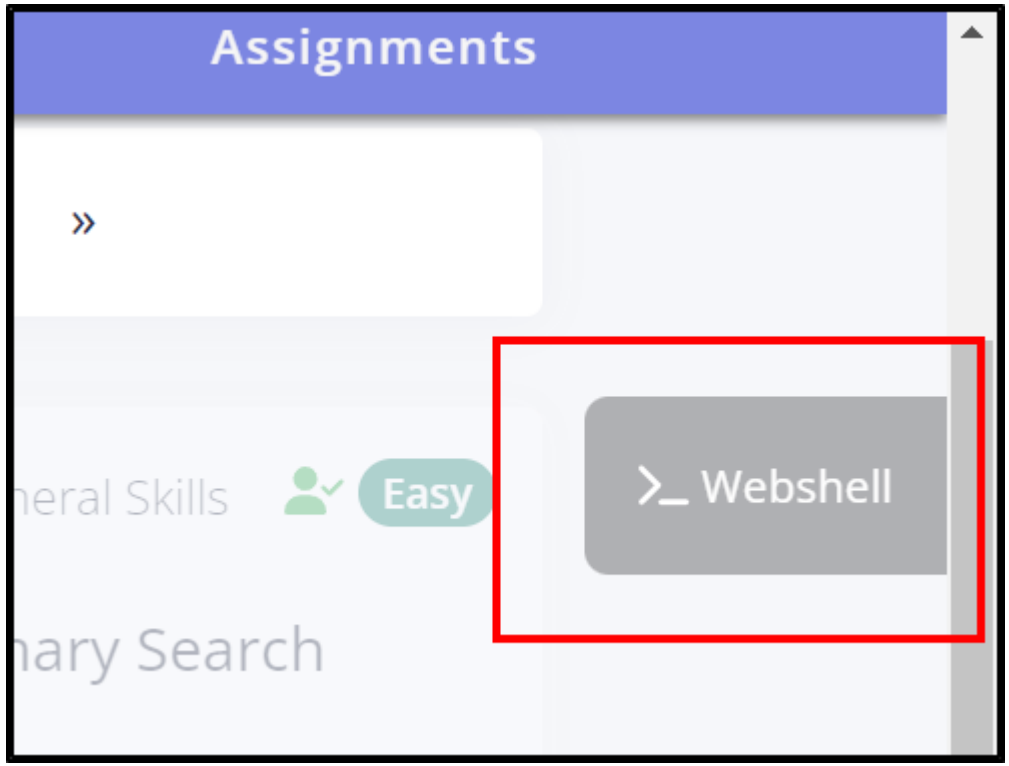
This is the first workshop for intro Linux OS Operations.

Let us begin!



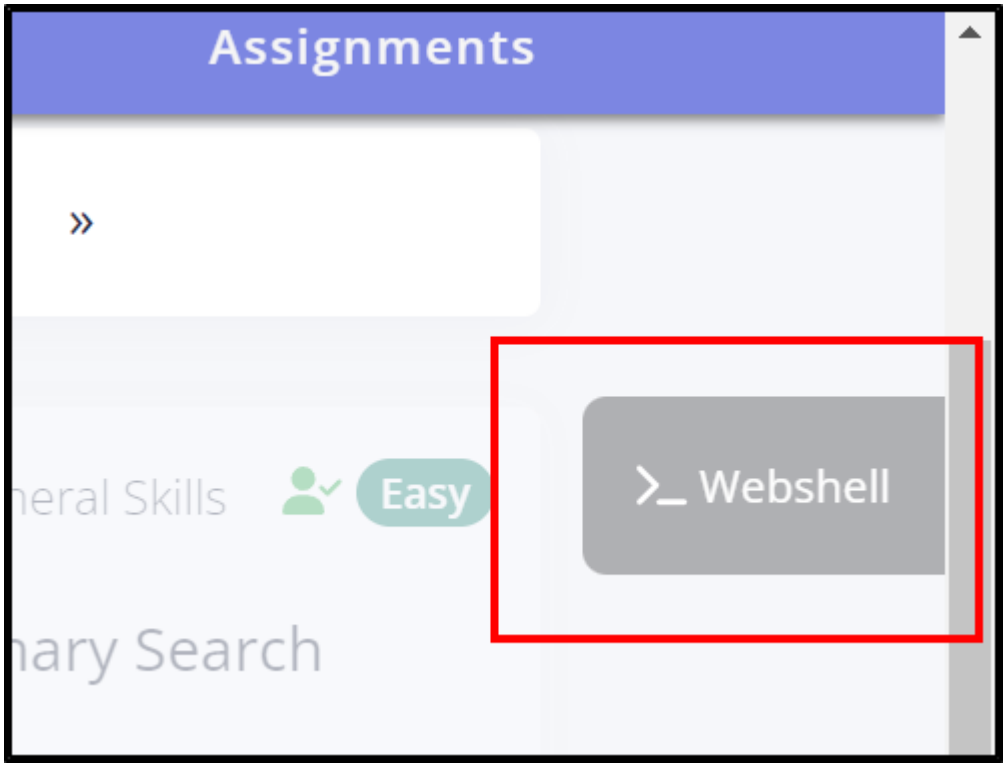
Accessing a Terminal

The first thing we need to do is open our command-line interface (CLI) terminal. After logging into PicoCTF, we can access the web-shell.

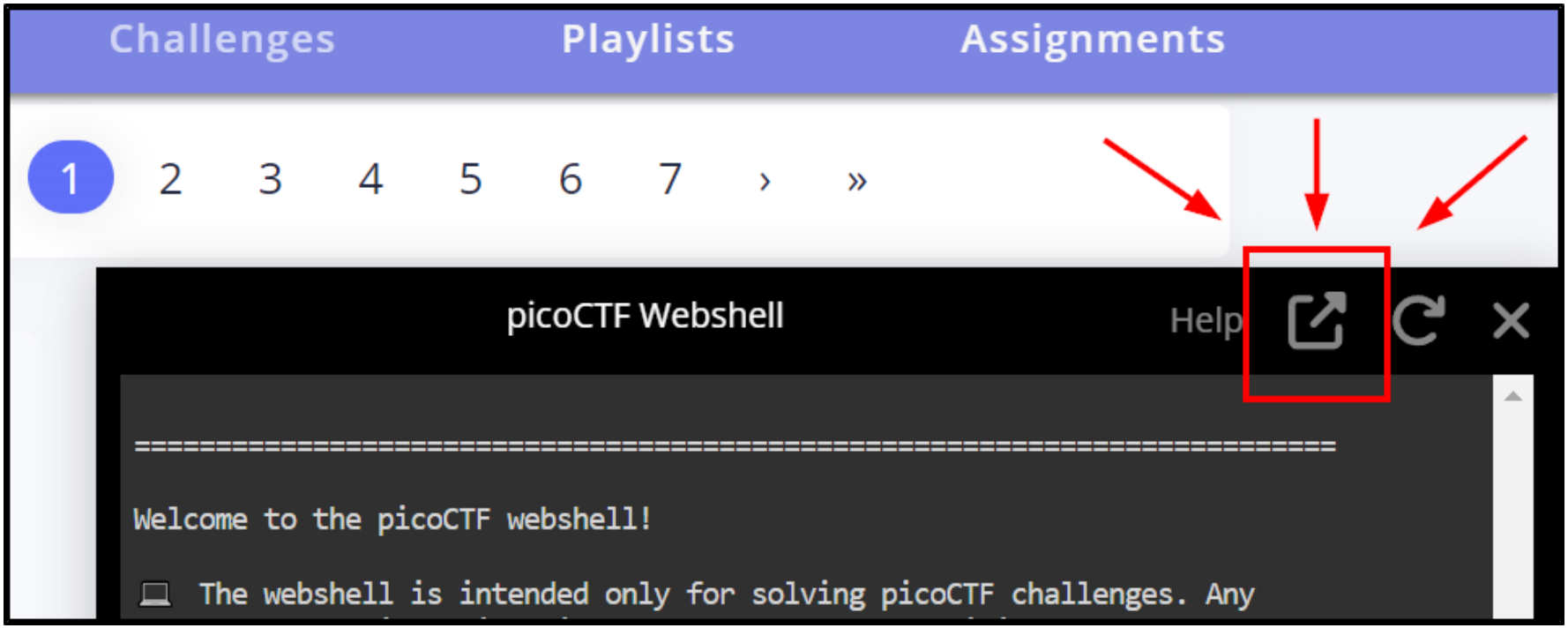


Accessing a Terminal

The button is found on the Practice webpage, and it's located at the upper-right corner of the webpage. If we can't see it, we should scroll down until it appears



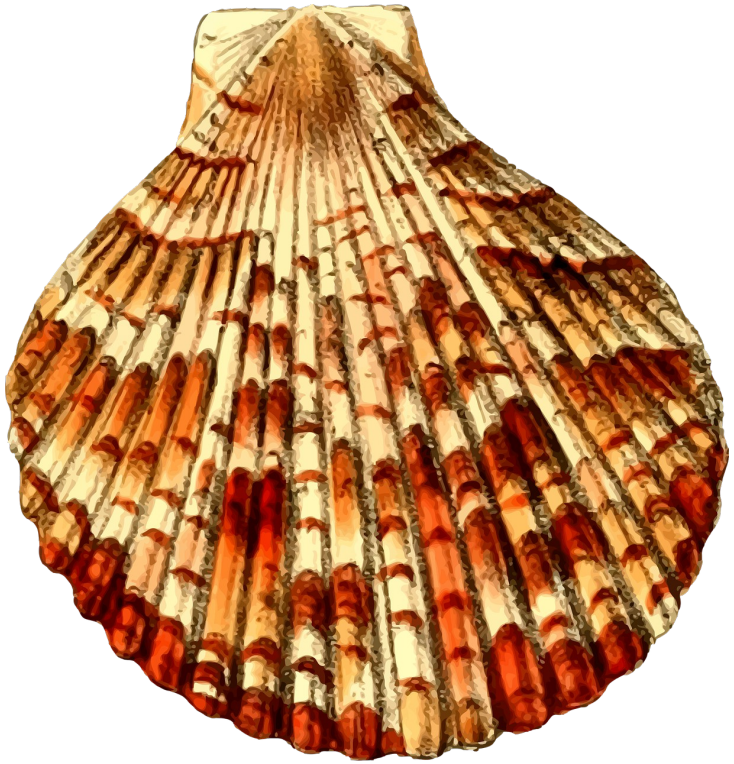
Accessing a Terminal



After opening the webshell, we should click on the **Popout Webshell Terminal** button, outlined in the image above

Connecting to a Server Using SSH

Let's first learn about how to connect to remote servers using SSH.



Super SSH Challenge

After logging into PicoCTF, opening the webshell, and putting it in a separate browser tab, navigate to the following webpage to access the Super SSH challenge:

https://play.picoctf.org/practice/challenge/424?page=1&search=super

SSH (Secure Shell)

To use SSH we will need username and server information. Let's get that information from the Bandit CTF homepage:

https://overthewire.org/wargames/bandit/bandit0.html

SSH (Secure Shell)

After clicking on the blue **Start Instance** button we are given information we can use to login to the remote server using SSH:

- a username: **ctf-player**
- a server name: **titan.picoctf.net**
- a port number to connect on: **<port_num>**
- a password: **<password>**

SSH (Secure Shell)

We can combine this information to form a command using SSH like this:

```
ssh ctf-player@titan.picoctf.net -p <port_num>
```

Let's breakdown what this command does:

SSH (Secure Shell)

```
ssh ctf-player@titan.picoctf.net -p <port_num>

ssh              ← the command itself

ctf-player       ← the user account on the server

titan.           ← the server URL
picoctf.net

-p <port_num> ← the networking port where the
                service is located
```

SSH (Secure Shell)

```
ssh ctf-player@titan.picoctf.net -p <port_num>

ssh                 ← the command itself

ctf-player          ← the user account on the server

titan.              ← the server URL
picoctf.net

-p <port_num>       ← the networking port where the
                      service is located
```

SSH (Secure Shell)

```
ssh ctf-player@titan.picoctf.net -p <port_num>

ssh              ← the command itself

ctf-player       ← the user account on the server

titan.           ← the server URL
picoctf.net

-p <port_num> ← the networking port where the
                service is located
```

SSH (Secure Shell)

```
ssh ctf-player@titan.picoctf.net -p <port_num>

ssh                 ← the command itself

ctf-player          ← the user account on the server

titan.              ← the server URL
picoctf.net

-p <port_num>       ← the networking port where the
                      service is located
```

SSH (Secure Shell)

```
ssh ctf-player@titan.picoctf.net -p <port_num>

ssh                 ← the command itself

ctf-player          ← the user account on the server

titan.              ← the server URL
picoctf.net

-p <port_num>       ← the networking port where the
                      service is located
```

SSH (Secure Shell)

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[titan.picoctf.net]:50505' (ED25519) to the list of known host
s.
ctf-player@titan.picoctf.net's password:
Welcome ctf-player, here's your flag: picoCTF{s3cur3_c0nn3ct10n_[illegible]}
Connection to titan.picoctf.net closed.
```

After hitting enter to execute the command, it will ask if you want to continue connecting:
Type **yes** and hit enter

SSH (Secure Shell)

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[titan.picoctf.net]:50505' (ED25519) to the list of known host
s.
ctf-player@titan.picoctf.net's password:
Welcome ctf-player, here's your flag: picoCTF{s3cur3_c0nn3ct10n_[illegible]}
Connection to titan.picoctf.net closed.
```

Then the system will prompt us for a password. Go back to the challenge webpage and copy the password, then go back to the webshell and paste in the password and hit enter to login

SSH (Secure Shell)

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[titan.picoctf.net]:50505' (ED25519) to the list of known host
s.
ctf-player@titan.picoctf.net's password:
Welcome ctf-player, here's your flag: picoCTF{s3cur3_c0nn3ct10n_[redacted]}
Connection to titan.picoctf.net closed.
```

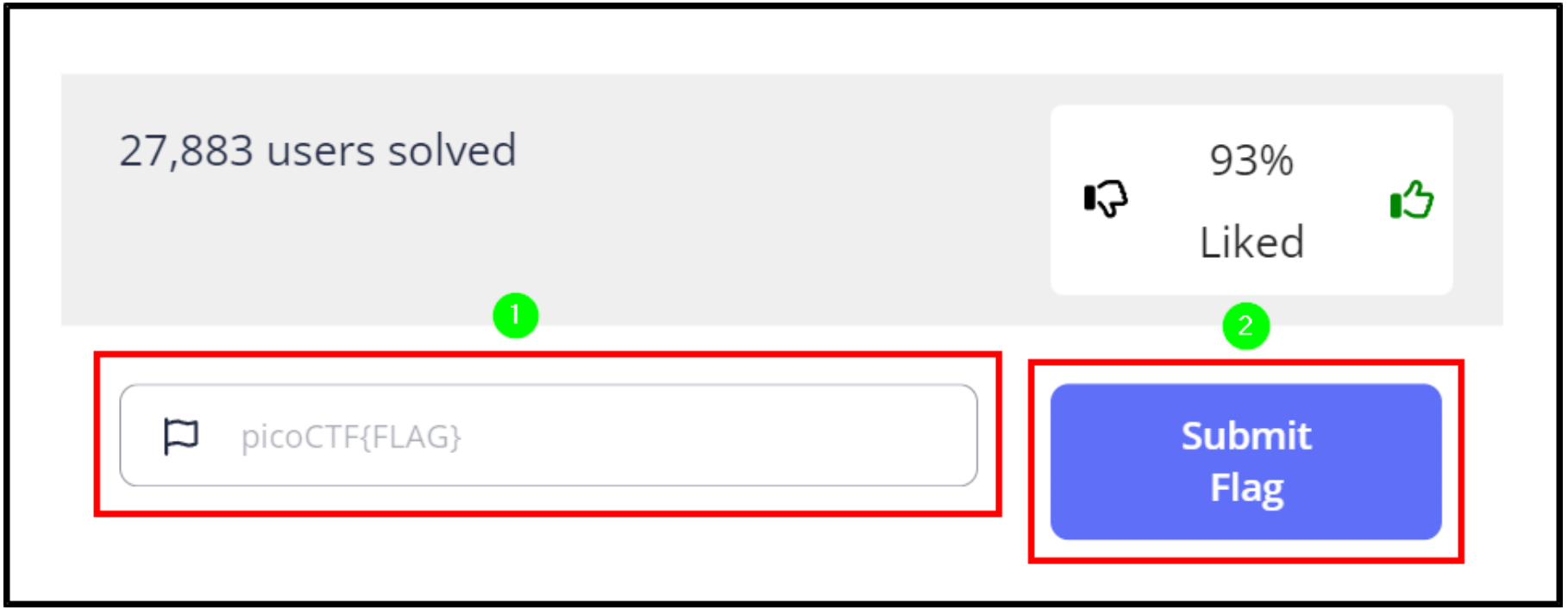
Note that we won't see any feedback from the system when we enter our password. This is normal when typing a password into Linux, just paste in the password and hit enter.

Capture the Flag

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[titan.picoctf.net]:50505' (ED25519) to the list of known host
s.
ctf-player@titan.picoctf.net's password:
Welcome ctf-player, here's your flag: picoCTF{s3cur3_c0nn3ct10n_[illegible]}
Connection to titan.picoctf.net closed.
```

The goal of the challenge was to login using SSH so when we login, we are give the flag for the challenge. To finish the challenge, we need to copy the flag from the webshell...

Capture the Flag



Then go back to the challenge page and paste the flag into the flag submission field, then click on the blue Submit Flag button.

Next Challenge: Obedient Cat

Our next challenge will help us understand downloading and reading files in the Linux terminal. Access the Obedient Cat challenge here:

https://play.picoctf.org/practice/challenge/147?category=5&page=1&search=cat

Obedient Cat

```
theshyhat-picoctf@webshell:~$ wget https://mercury.picoctf.net/static/217686fc11d733b80be6
2dcfcfca6c75/flag
--2024-09-30 19:42:38--  https://mercury.picoctf.net/static/217686fc11d733b80be62dcfcfca6c
75/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: 'flag'

flag                  100%[===========================>]      34  --.-KB/s    in 0s

2024-09-30 19:42:38 (22.6 MB/s) - 'flag' saved [34/34]
```

Then back at the webshell, type **wget**, then space, then paste in the address we copied from the challenge page.

Obedient Cat

```
theshyhat-picoctf@webshell:~$ wget https://mercury.picoctf.net/static/217686fc11d733b80be6
2dcfcfca6c75/flag
--2024-09-30 19:42:38--  https://mercury.picoctf.net/static/217686fc11d733b80be62dcfcfca6c
75/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: 'flag'

flag                  100%[===========================>]      34  --.-KB/s    in 0s

2024-09-30 19:42:38 (22.6 MB/s) - 'flag' saved [34/34]
```

Press the Enter key, then the challenge file will be downloaded to our webshell terminal.

Obedient Cat

```
theshyhat-picoctf@webshell:~$ ls
README.txt  flag
theshyhat-picoctf@webshell:~$
```

We can use the **ls** command to list all of the files in our current directory.

Obedient Cat

```
theshyhat-picoctf@webshell:~$ cat flag
picoCTF{s4n1ty_v3r1f13d_[redacted]}
theshyhat-picoctf@webshell:~$
```

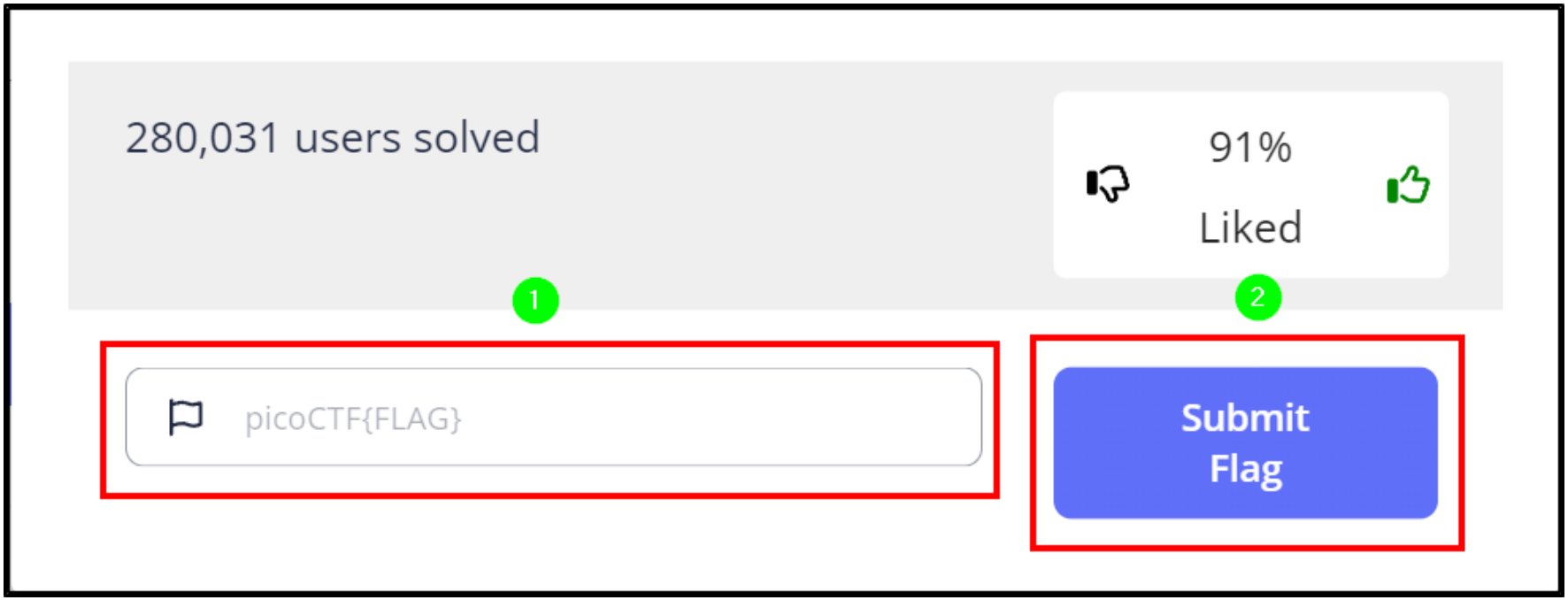
To finish the challenge, we need to read the **flag** file using the **cat** command. Type **cat flag**, then press the enter key, and the challenge's flag string will appear

Obedient Cat

```
theshyhat-picoctf@webshell:~$ cat flag
picoCTF{s4n1ty_v3r1f13d_[redacted]}
theshyhat-picoctf@webshell:~$
```

To finish the challenge, we copy the flag output...

Obedient Cat



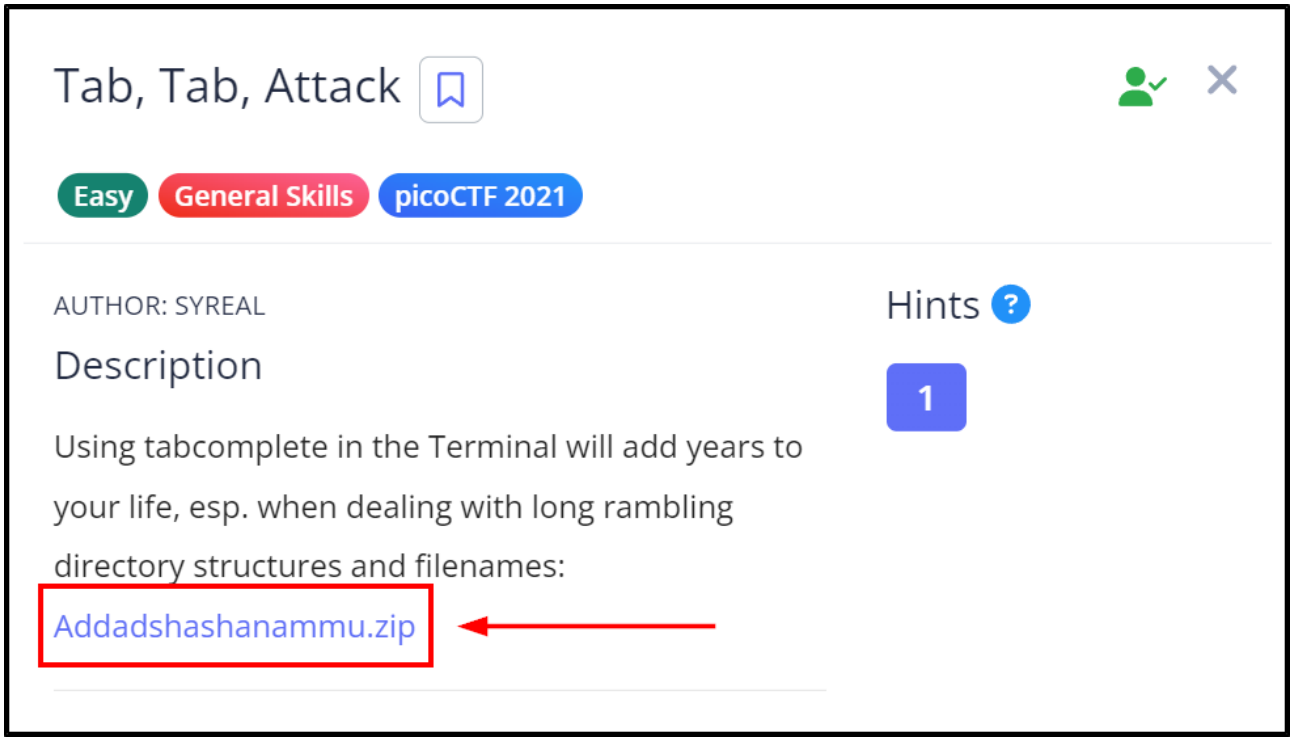
Then go back to the challenge page, and paste the flag value into the field submission field, then click on the blue **Submit flag** button

Final Challenge: Tab Tab Attack

The last challenge we'll cover this session is called **Tab, Tab, Attack**, which can be accessed from the following URL:

https://play.picoctf.org/practice/challenge/176?category=5&page=1&search=tab

Tab Tab Attack



Once more, we have to copy the link to the challenge file, then download it to our webshell using the **wget** command

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ ls
Addadshashanammu.zip  README.txt  flag
theshyhat-picoctf@webshell:~$
```

After the file is downloaded, we can use **ls** to make sure it's downloaded to our directory.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ unzip Addadshashanammu.zip
Archive:  Addadshashanammu.zip
   creating: Addadshashanammu/
   creating: Addadshashanammu/Almurbalarammi/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/Assurnabitashpi/
```

Since this file is a zip file, we need to extract the files from it using the unzip command. Type **unzip**, then space, then the letters **Ad**

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ unzip Addadshashanammu.zip
Archive:  Addadshashanammu.zip
   creating: Addadshashanammu/
   creating: Addadshashanammu/Almurbalarammi/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/
   creating: Addadshashanammu/Almurbalarammi/Ashalmimilkala/Assurnabitashpi/
```

Then press the **Tab** key on the keyboard. This will auto-complete the name of the file. Tab auto-complete is a very useful method of inputting the names of files with long names.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ ls
Addadshashanammu  Addadshashanammu.zip  README.txt  flag
theshyhat-picoctf@webshell:~$
```

If we use the **ls** command to see the contents of the directory, we see that there is a directory in here that named **Addadshashanammu**.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ ls
Addadshashanammu  Addadshashanammu.zip  README.txt  flag
theshyhat-picoctf@webshell:~$
```

The point of this challenge is to use the tab auto-complete to enter directories with long, complicated names to obtain the flag.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ cd Addadshashanammu/
theshyhat-picoctf@webshell:~/Addadshashanammu$ ls
Almurbalarammi
theshyhat-picoctf@webshell:~/Addadshashanammu$
```

To enter the directory, type **cd** (change directory), then space, then the first two letters of the directory, **Ad**, then press the **Tab** key, and press enter.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~$ cd Addadshashanammu/
theshyhat-picoctf@webshell:~/Addadshashanammu$ ls
Almurbalarammi
theshyhat-picoctf@webshell:~/Addadshashanammu$
```

If we use the **ls** command, we see that there is another directory in here named **Almurbalarammi**.

Tab Tab Attack

```
theshyhat-picoctf@webshell:~/Addadshashanammu$ cd Almurbalarammi/
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi$ ls
Ashalmimilkala
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi$
```

So we need to use the **cd** command and tab auto-complete to enter this new directory.

Tab Tab Attack

```
i/Maelkashishi/Onnissiralis/Ularradallaku$ ls
fang-of-haynekhtnamet
theshyhat-picoctf@webshell:~/Addadshashanammu/
i/Maelkashishi/Onnissiralis/Ularradallaku$
```

After doing this process 5 more times, we find this file in the directory, named **fang-of-haynekhtnamet**.

Tab Tab Attack

```
i/Maelkashishi/Onnissiralis/Ularradallaku$ ./fang-of-haynekhtnamet
*ZAP!* picoCTF{l3v3l_up!_t4k3_4_r35t!_[illegible]}
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi/Ashal
i/Maelkashishi/Onnissiralis/Ularradallaku$
```

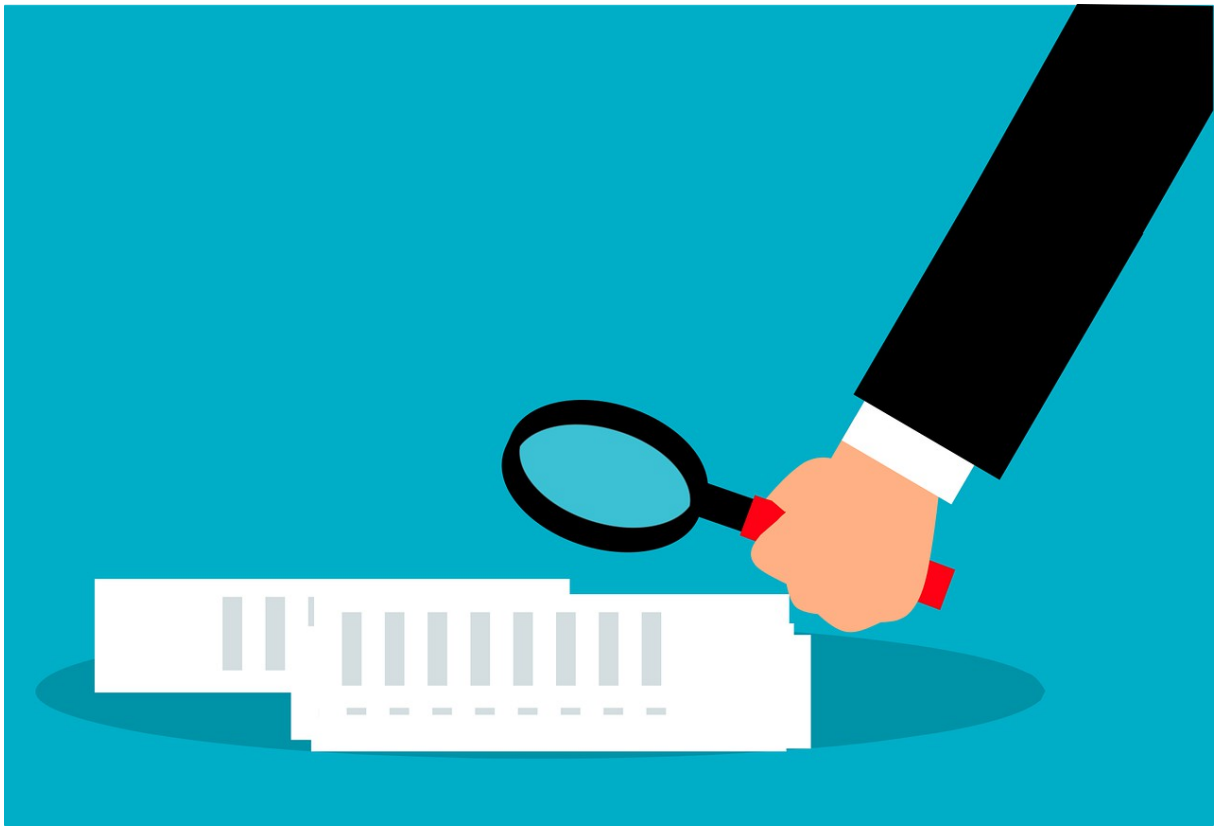
To run an executable file in our directory, we have to type **./** then type the first couple of letters from the file name, **fa**, then tab auto-complete.

Tab Tab Attack

```
i/Maelkashishi/Onnissiralis/Ularradallaku$ ./fang-of-haynekhtnamet
*ZAP!* picoCTF{l3v3l_up!_t4k3_4_r35t!_[illegible]}
theshyhat-picoctf@webshell:~/Addadshashanammu/Almurbalarammi/Ashal
i/Maelkashishi/Onnissiralis/Ularradallaku$
```

To complete the challenge, we repeat the process we did for the previous challenges, copying the flag, then pasting it into the flag submission field.

Summary



Let's review the Linux commands we learned in today's workshop:

Ls Command

The Ls command lists the files and directories in the current directory.

It can be used with the `-l` argument to output in a list format, and with the `-a` argument to include hidden files and directories in the output. These two arguments can be combined to produce both ouputs, e.g., `-la`

Ls Command

```
└─$ ls -la
total 12
drwxr-xr-x  2 shyhat shyhat 4096 May 30 09:28 .
drwxr-xr-x 42 shyhat shyhat 4096 May 30 09:21 ..
-rw-r--r--  1 shyhat shyhat   12 May 30 09:28 example.txt
```

Here we see the output of the **ls** command with the **l** and **a** flags combined

Cat Command

The Cat command reads the contents of a file. The name of the file to be read must be supplied as an argument to the command.

E.g., `cat example.txt`

Cat Command

```
└─$ cat example.txt
sample text
```

Here the contents of the **example.txt** file is read using the **cat** command

Cd Command

The Cd command changes the current directory to the one specified. The new directory must be supplied as an argument to the command.

E.g., `cd downloads`

Cd Command

```
┌──(shyhat㉿hackerfrog)-[~]
└─$ cd example

┌──(shyhat㉿hackerfrog)-[~/example]
└─$
```

File Command

The File command identifies the type of contents for a specified file. The file name must be supplied as an argument to the File command.

E.g., `file picture.jpg`

File Command

```
└─$ file example.txt
example.txt: ASCII text
```

Unzip Command

The **unzip** command allows files to be extracted from zip files. We the syntax is like this:

```
unzip <file_name>.zip
```

For example:

```
unzip zipfile.zip
```

What's Next?

In the next HackerFrogs Afterschool Linux OS workshop, we'll continue learning Linux commands with PicoCTF.

Extra Credit

Looking for more study material on this workshop's topics?

See this video's description for links to supplemental documents and exercises!

Until Next Time, HackerFrogs!