INCS 3320 – Project
Ethical Hacking and Penetration Testing

Pentest Report
Reporting Vulnerabilities found in the System

Name: SAIF CHHIPA
Username: DarkSpirit
Instructor: Hamidreza Talebi
Date: 28th October 2024

Table of Contents

Objective

Enumeration (Finding Vulnerabilities)

- **Scanning each IP address for vulnerabilities**
- **Exploring open ports**
- **Determining running services**
- **Finding operating system running and versions of the services**
- **Mapping the network**

Exploitation (Exploiting Vulnerabilities)

- **Exploiting running services**
- **Gaining remote access**
- **Escalating privilege**

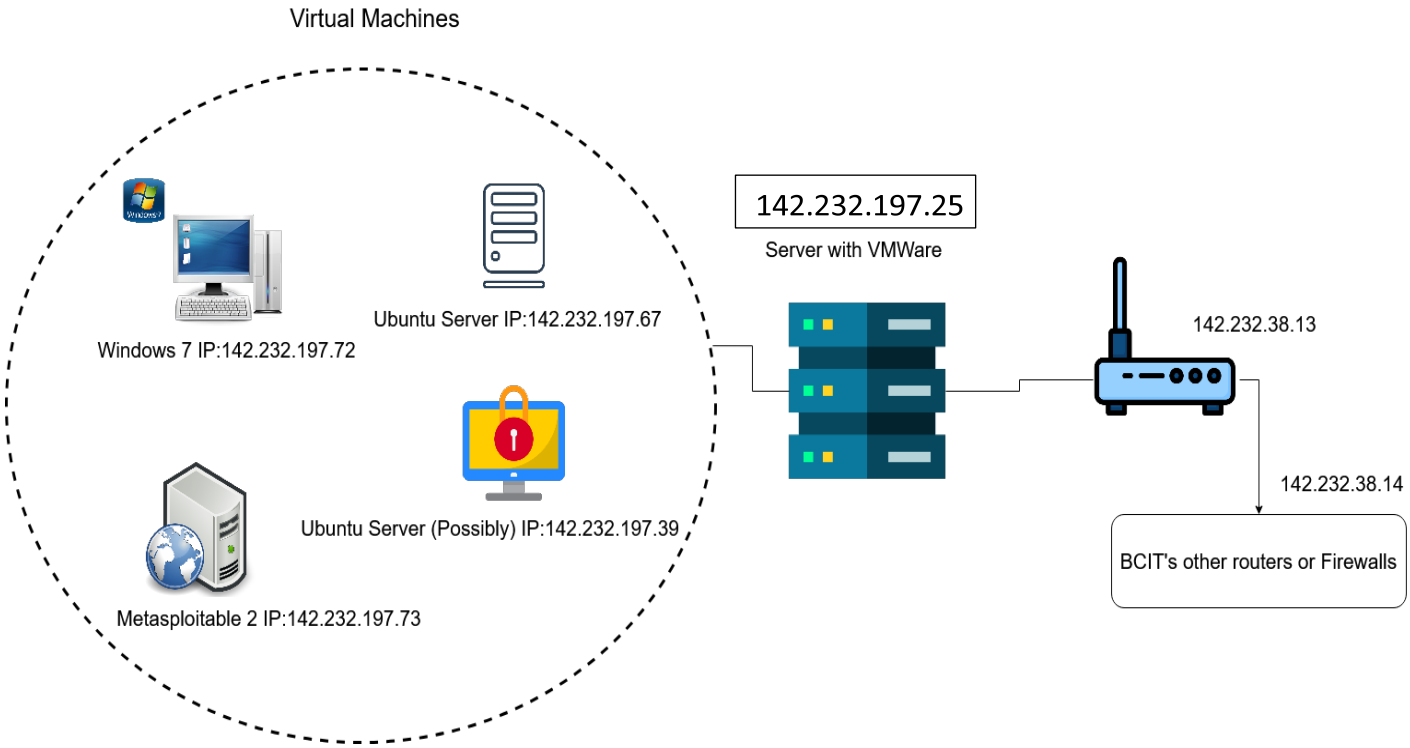
Security Suggestion

- **Possible ways to avoid or mitigate the threat**
- **Possible solutions to existing vulnerabilities**

Target IP Addresses

1. **142.232.197.73**
2. **142.232.197.72**
3. **142.232.197.67**
4. **142.232.197.39**

Network Topology



The specified IP addresses are Virtual Machines operating on a server with VMWare ESXi, accessible at the IP address 142.232.197.246. These VMs have a default gateway set to 142.232.197.254, which connects to a router.

These VMs are connected in the same subnet, functioning like a LAN environment with a switch.

Host 1 – Metasploitable

IP Address: 142.232.197.73

Objective: To enumerate active services and exploit potential vulnerabilities.

Overview

Host 142.232.197.73 is a Linux Metasploitable machine. It is designed to be vulnerable for learning purposes. It has numerous vulnerabilities and open ports, almost every port can be exploited. It has all the vulnerabilities mentioned in OWAS top 10, and many others.

Enumeration & Methodology

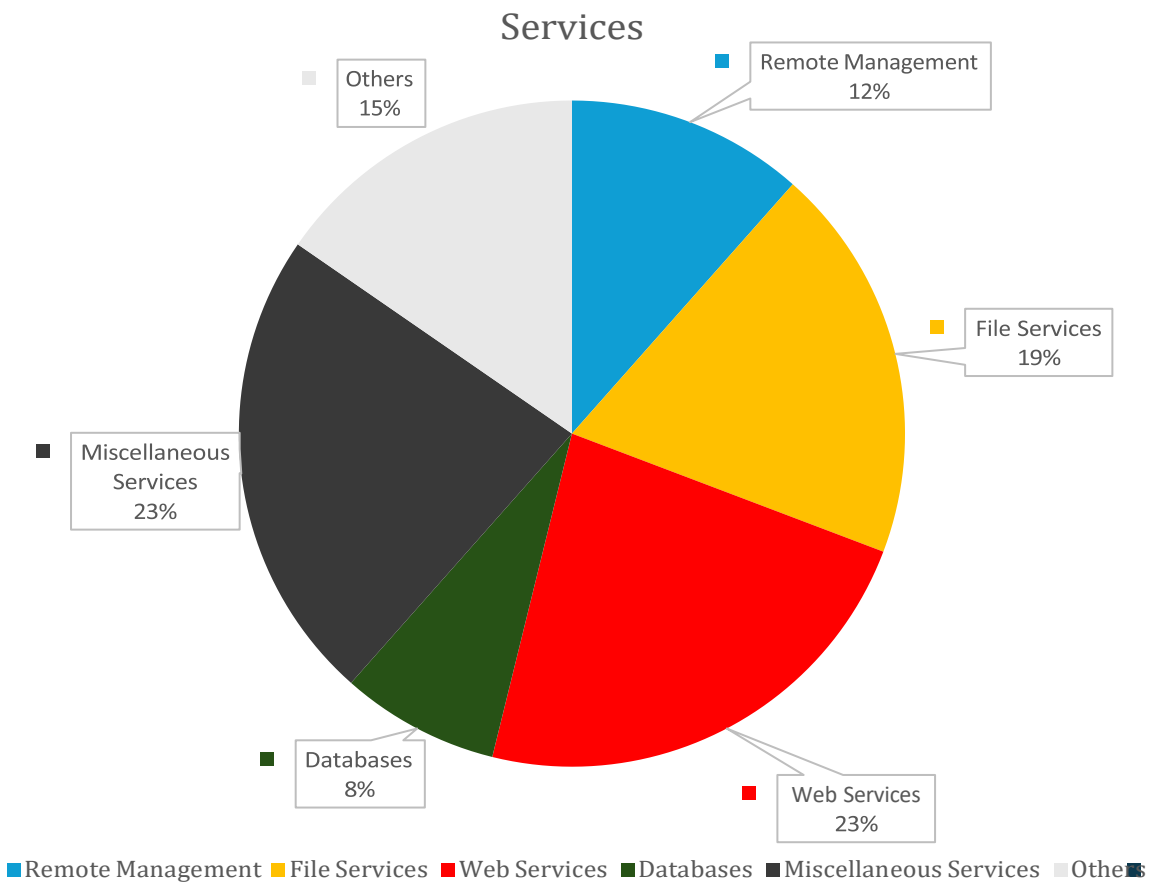
- Nmap Scan showed that the host has multiple services running on including Web services, Databases, Filesharing and Remote management services.
- It has web services such as Tomcat service, Damn Vulnerable Web-application, Mutillidae, Tikiwiki and Web Distributed Authoring and Versioning (WebDAV) which can potentially grant access to the server and can reveal critical information.
- The Nmap scan contains a lot of services. Identifies services are listed below and the Nmap result is attached at the end of the report.

Identified Services

- **Remote Management**: 3 services (SSH, Telnet, VNC)
- **File Services**: 5 services (FTP, NFS, Samba, NetBIOS, RPC)
- **Web Services**: 6 services (Apache, Tomcat, DVWA, WebDAV, Mutillidae, TikiWiki)
- **Databases**: 2 services (MySQL, PostgreSQL)
- **Mail Services**: 1 service (SMTP)
- **Miscellaneous Services**: 5 services (DNS, X11, IRC, Java RMI, Shell/Backdoor)
- **Others**: 4 services (exec, login, krb524, ajp13)

Most of these services haven't been updated for a long time and some of them are configured with very little security and few are just left open, which is too risky.

Refer to the graph shown below to identify which category has the most vulnerabilities and which one an attacker can use as a threat vector.



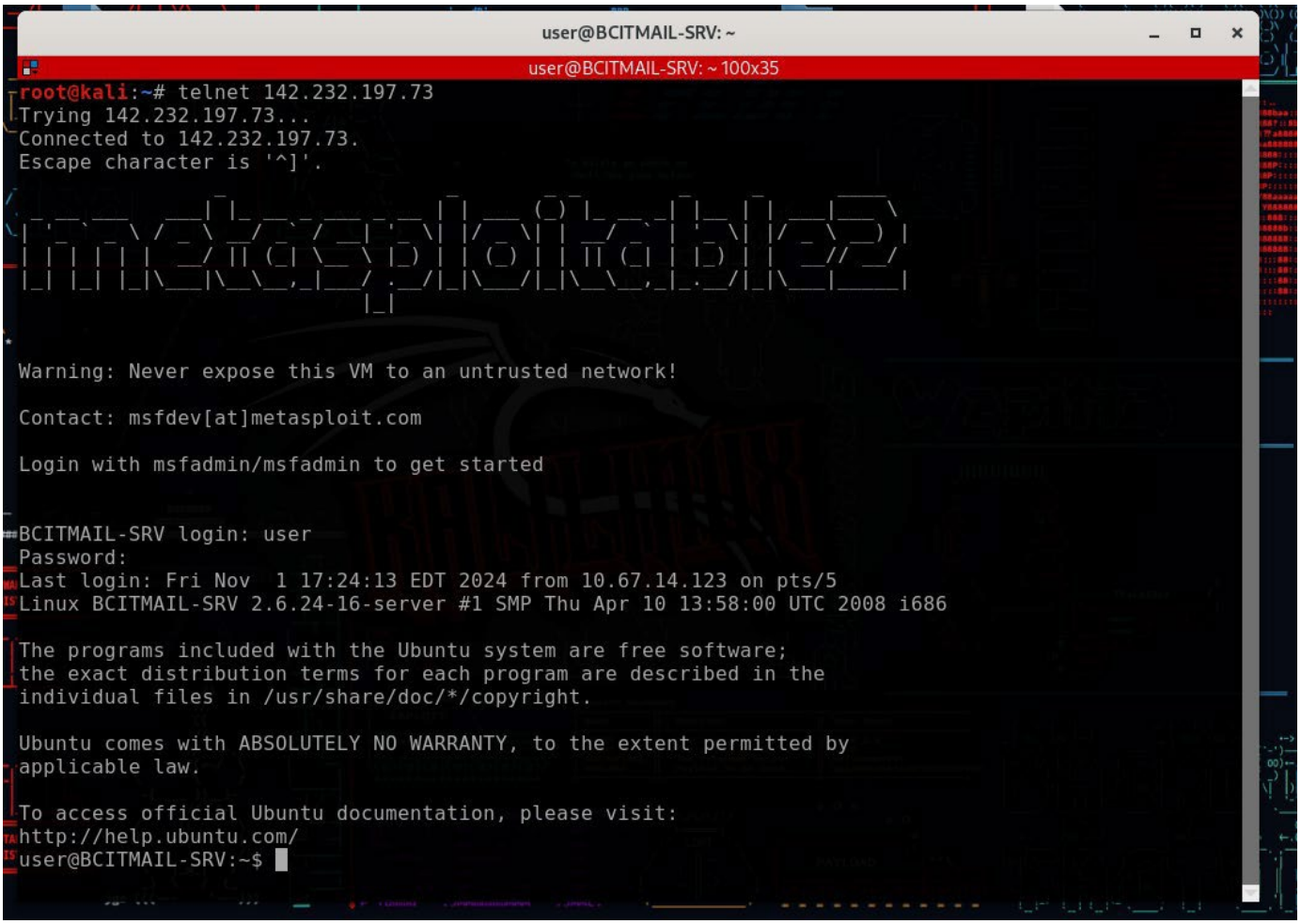
Exploitation

Remote Management Service

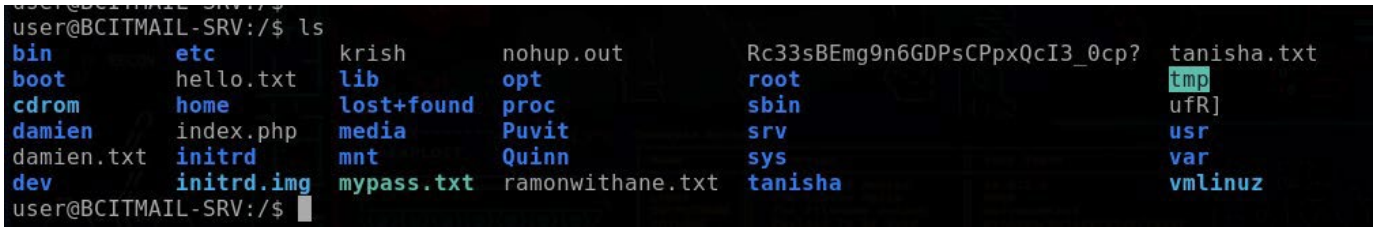
Remote Management Services (RMS) such as SSH, Telnet, RDP, and VNC are commonly used across various business environments. RMS is particularly useful for managing computers, servers, switches, routers, and firewalls from a distance. This service acts as a gateway from the public network to the organization's private or internal network, making it essential to implement proper security measures for these nodes.

In this scenario, we have identified that services like VNC, Telnet, and SSH are active on the target server. Let's evaluate whether these services are configured correctly.

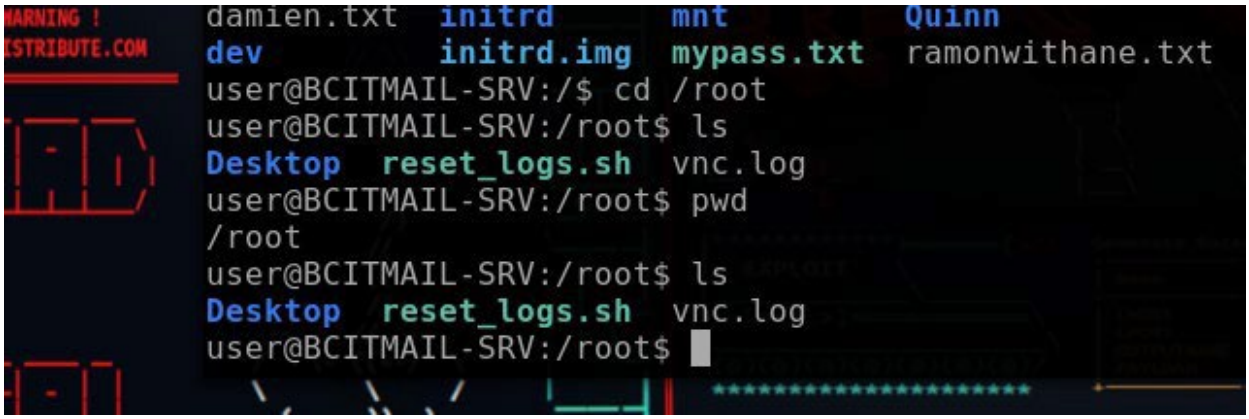
Service: Telnet
Port: 23
Risk Level: Critical



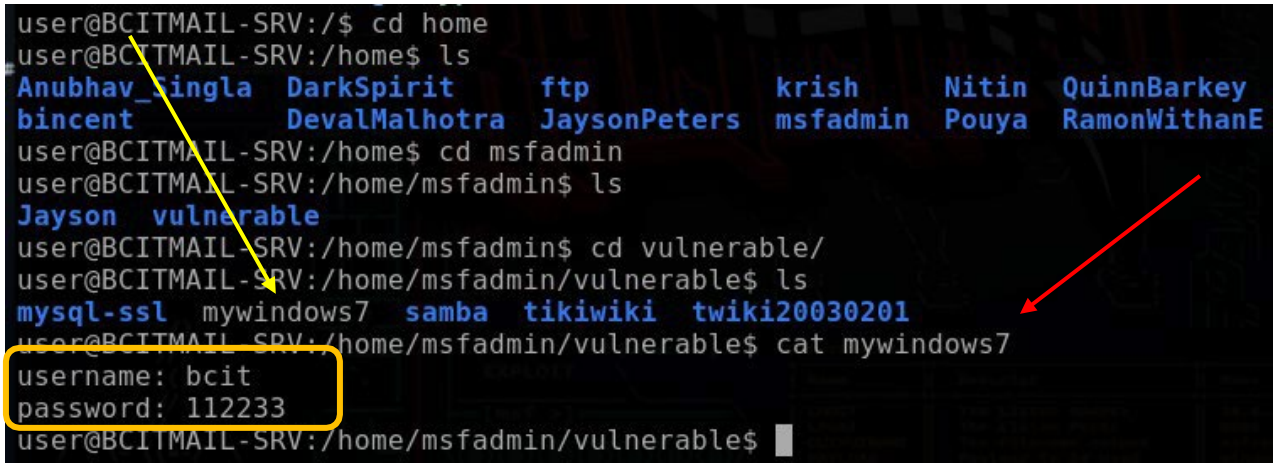
In the screenshot above, we can see that I was successfully able to log into 142.232.197.73 with a common username "user" and password "user12345". This is a critical leak, an individual can potentially access confidential information available on the server.



Using the normal user account, I can list the directories and files which are stored inside the root user's home directory. This is possible because the file permissions are not set correctly.



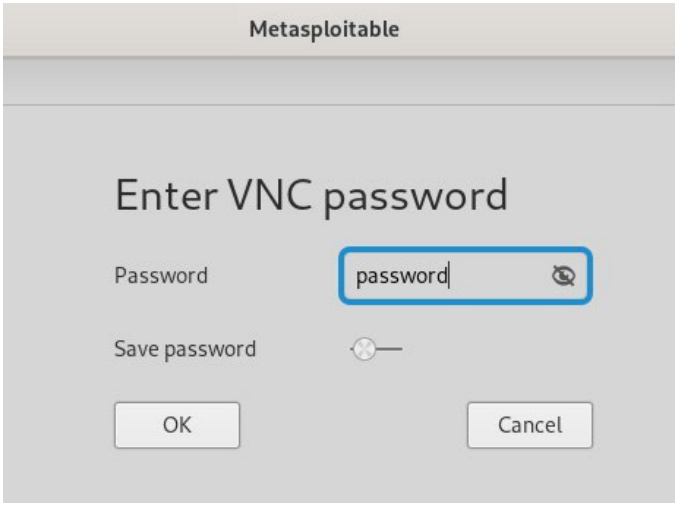
Over the further exploration I found the credentials for Windows 7 running on 142.232.197.72.



We have almost full control over this server, we have privilege to create and delete files and folders, modify things and etc.

Service: VNC
Port: 5900
Risk Level: Critical

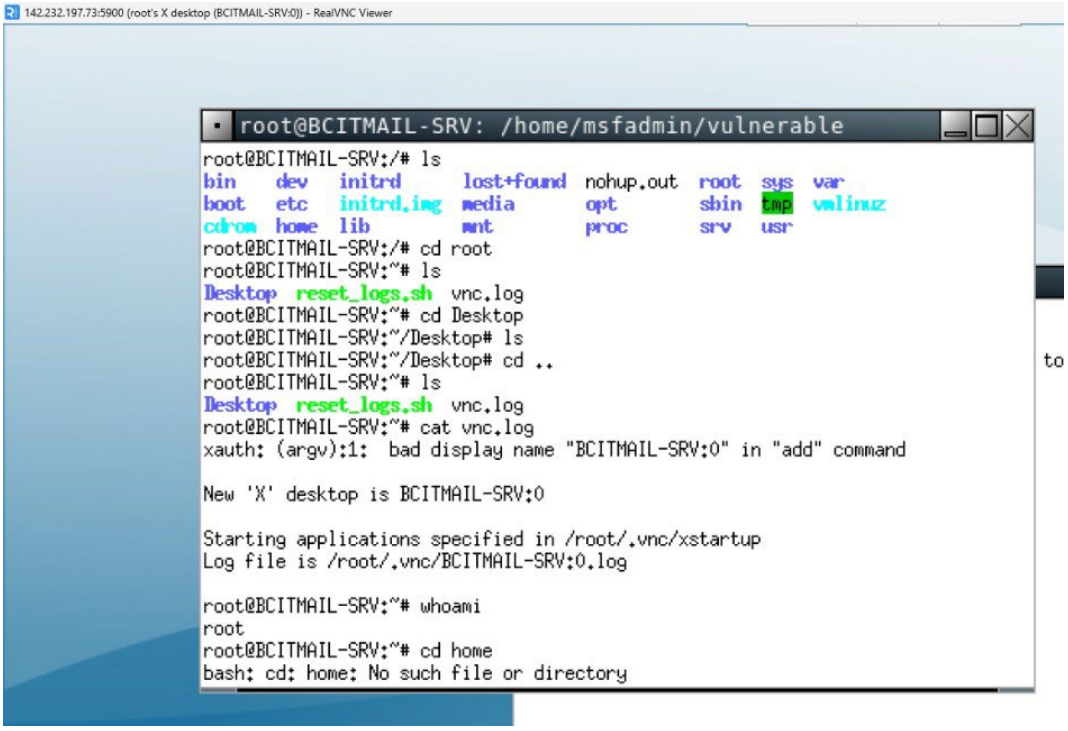
VNC (Virtual Network Computing) establishes a remote connection to a device. Unlike SSH (Secure Shell), which provides a command-line interface (CLI), VNC offers a graphical user interface (GUI) for remote management. It also serves as another public gateway to an organization's internal network.



In this scenario, I am using a VNC client from RealVNC, and for authentication, I will use commonly known passwords, such as "password."

By using this default password, I was able to log in as the root user. This represents a critical vulnerability, as root access grants full access to all files on the system.

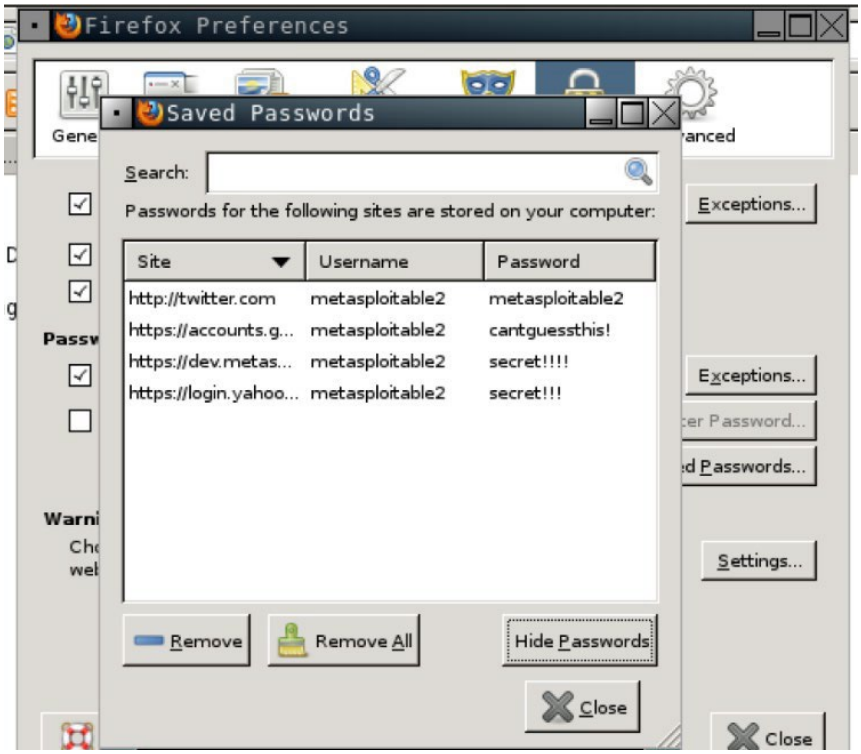
Using the default password, I was able to login as a **root.** This is a critical vulnerability. Root access allowed me to gain access to all the files in the system.



Using VNC connection, I completely searched the target machine to find what applications it is running, stored passwords, stored passwords in the browser, shadow files and much more.

I was able to see the logs, credentials stored inside the Tomcat directory for tomcat users. It is a critical flaw, a strong password should have been implemented to prevent an unauthorized access.

Example:



Security Steps:

In the example provided, I gained unauthorized access to the server using VNC and Telnet. Because of the default password, I had root privileges through VNC. These security vulnerabilities can be addressed by implementing a strong password and a unique username for the Telnet/SSH port. Additionally, services like VNC should be disabled when not in use. When they are necessary, it is important to utilize a strong password and a unique username. Furthermore, an access control list (ACL) should be used to allow connections to the Remote Management Services only from specific IP addresses.

File Services

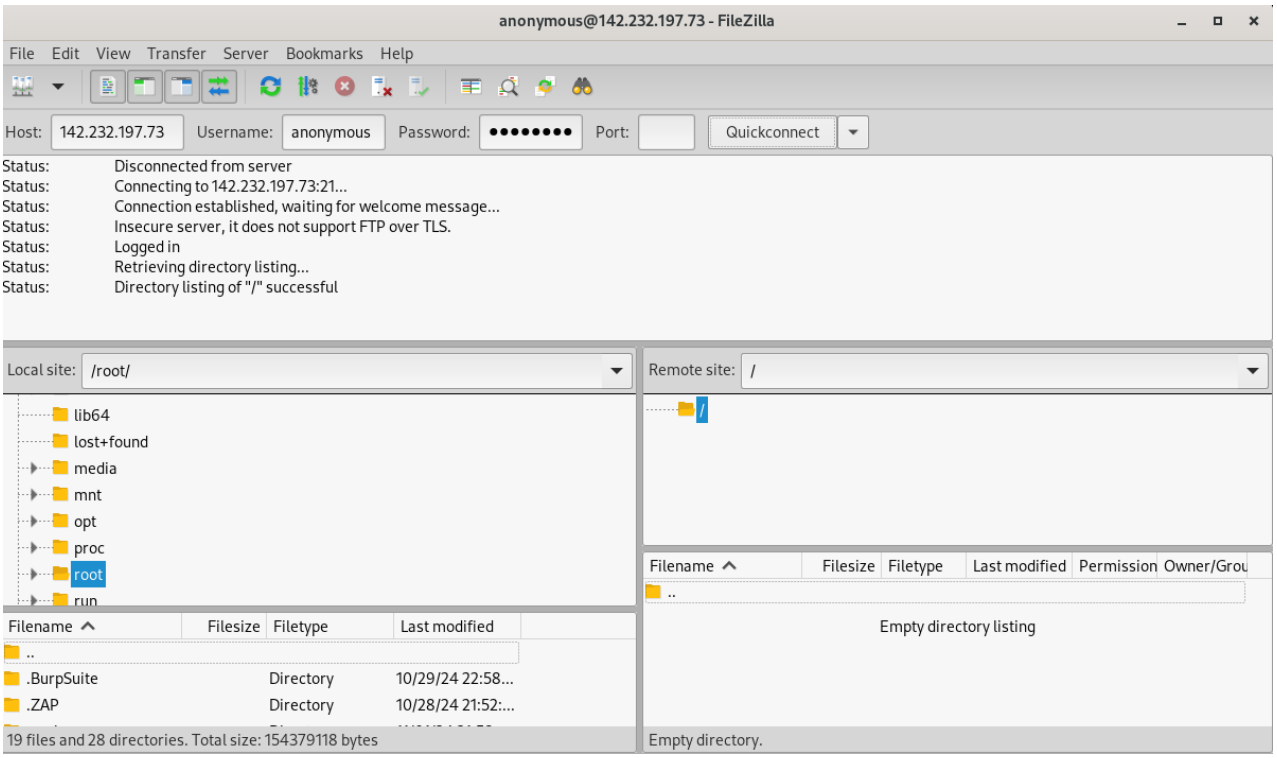
Unlike other servers, this server also runs file services, including FTP, NFS, and Samba. Failure to secure these services can lead to significant vulnerabilities, potentially resulting in threats such as ransomware attacks or unauthorized access through reverse shells if the permissions aren't correctly applied.

Service: FTP
Port: 21
Risk Level: Low

Nmap Result:

21/tcp open ftp
ftp-anon: Anonymous FTP login allowed (FTP code 230)

Sometimes, FTP servers are set up to allow anonymous login, enabling users to access files with limited permissions. While this typically restricts access to the server, it can pose risks. If a company has an anonymous login, users may share files on that server, which could allow unauthorized individuals to access and view all documents in the FTP folder.

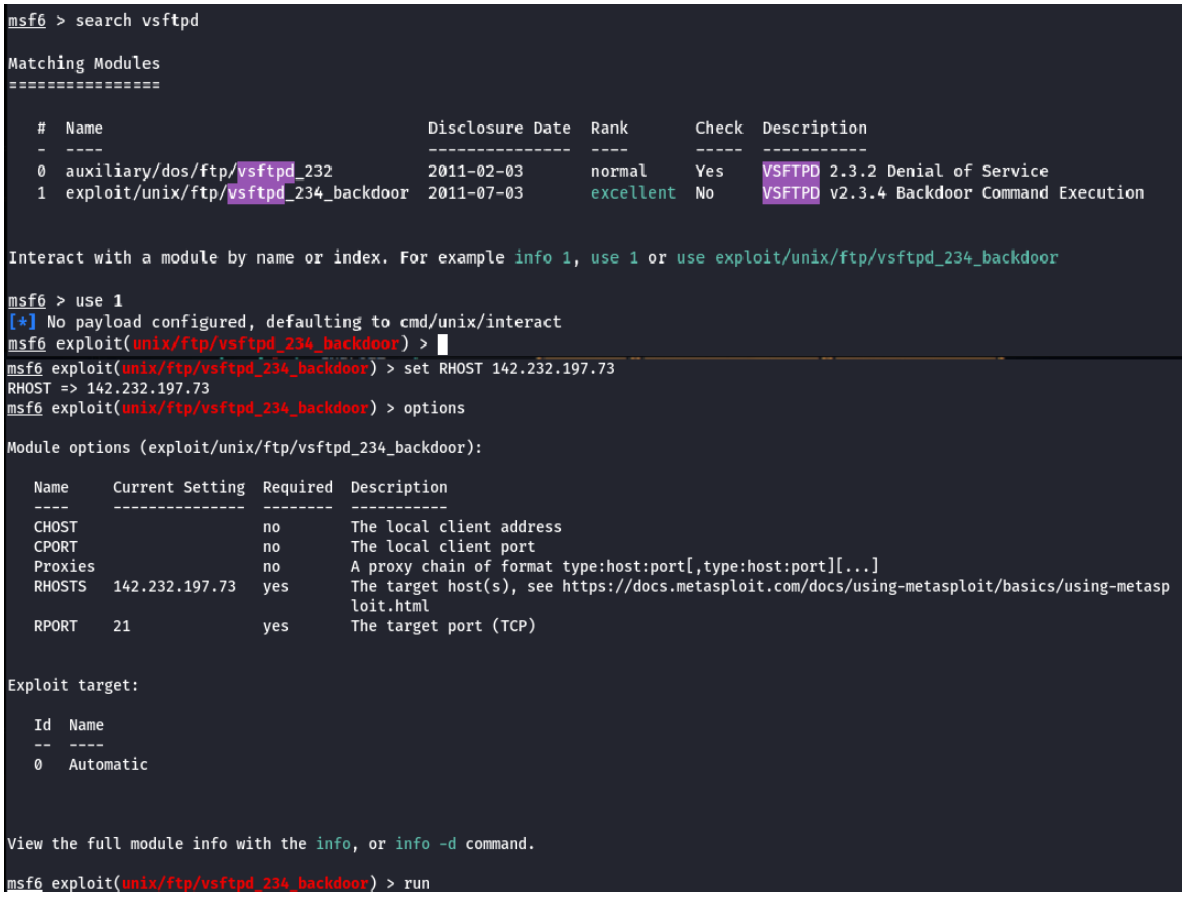


As discussed, it did not provide much information because the anonymous user does not have permission to access the directory, resulting in it being empty.

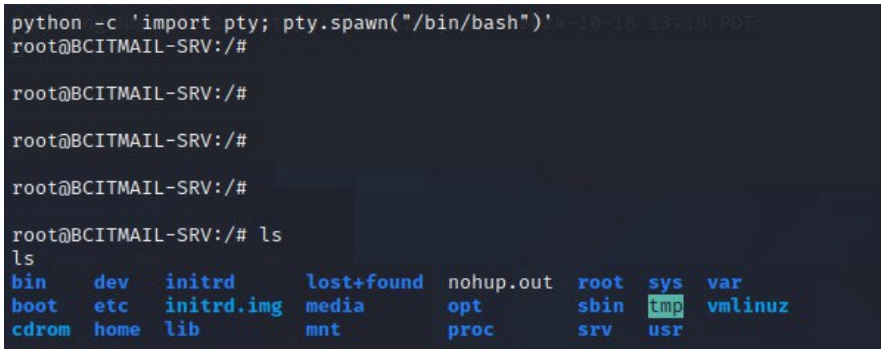
Service: FTP, VSFTPD 2.3.4
Port: 21
Risk Level: Critical
Nmap Result:

21/tcp open ftp vsftpd 2.3.4

VSFTPD 2.3.4 is vulnerable and can allow the successful execution of a backdoor, which will give full access to the server (root access). The backdoor can be performed by searching for the vsftpd and version in the Metasploit framework and could be exploited from there.



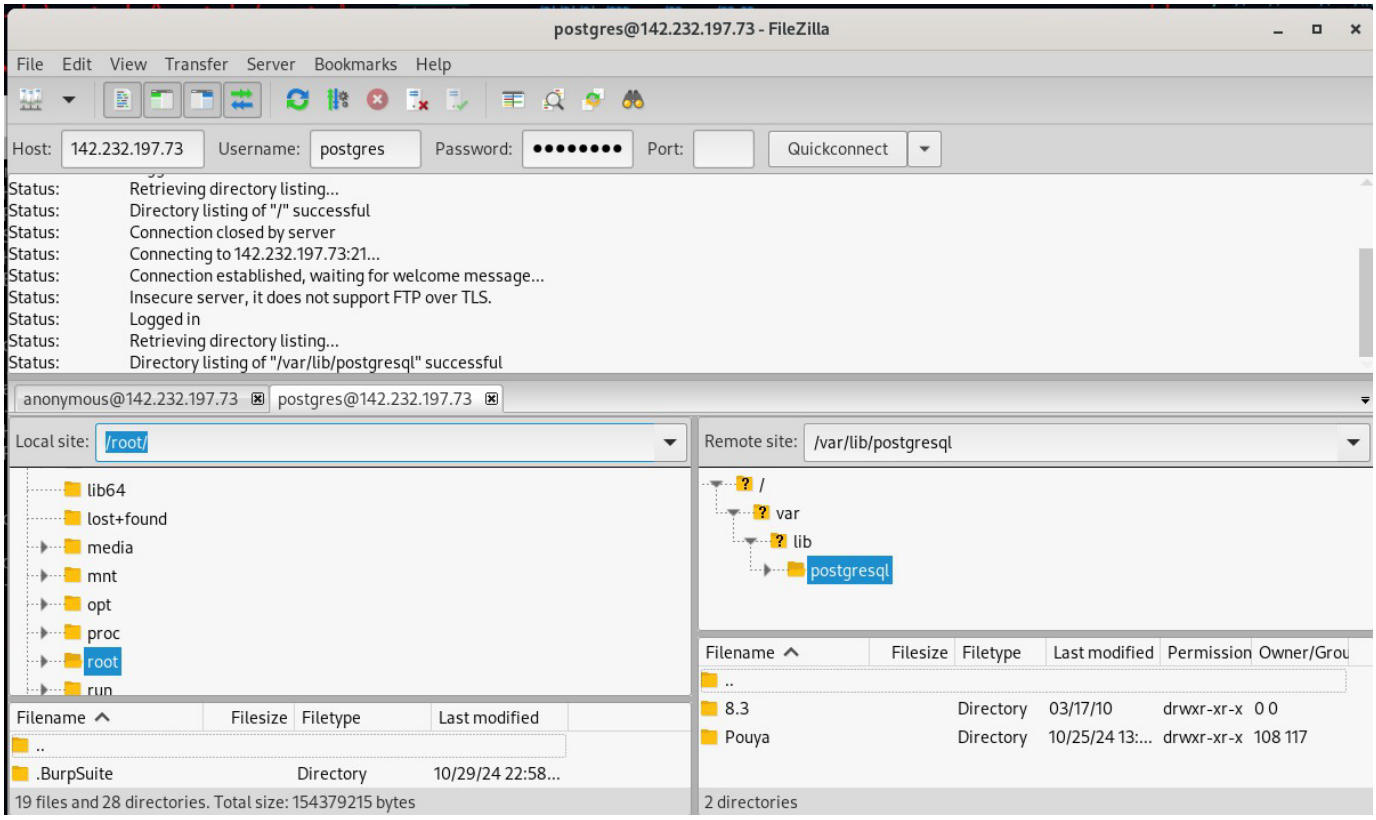
Furthermore, upon establishing a successful connection, we can import a more interactive Python shell.



Service: PostgreSQL
Port: 21
Risk Level: Moderate

Using FileZilla, similar to FTP, I was able to gain access to the PostgreSQL folder. This access reveals much more information than an anonymous login would allow. It enables me to list all the content within the directories but restricts me from making changes outside of the PostgreSQL folder. I was able to explore the /home directory and the remaining directories inside /var.

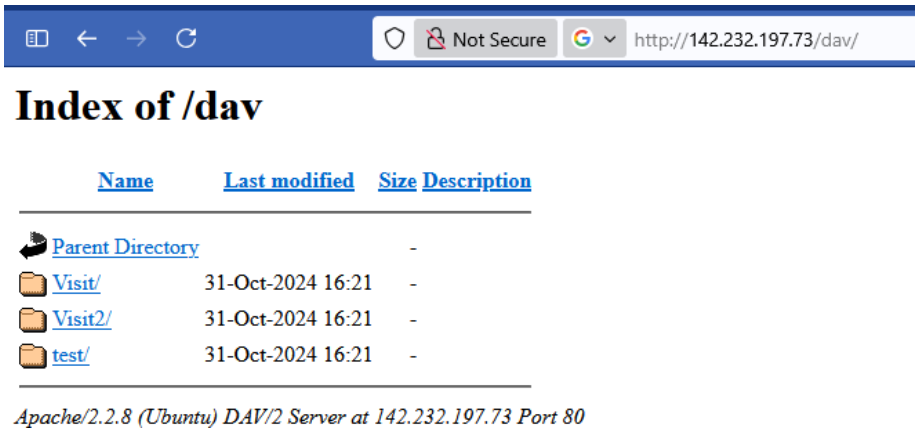
Credentials used: username: postgres password: postgres



Service: WebDAV Distributed Authoring and Versioning Port: 80
Risk Level: Low
Nmap Scan:

80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2

WebDAV (Web Distributed Authoring and Versioning) is an extension of the HTTP protocol that allows users to collaboratively edit and manage files on remote web servers.



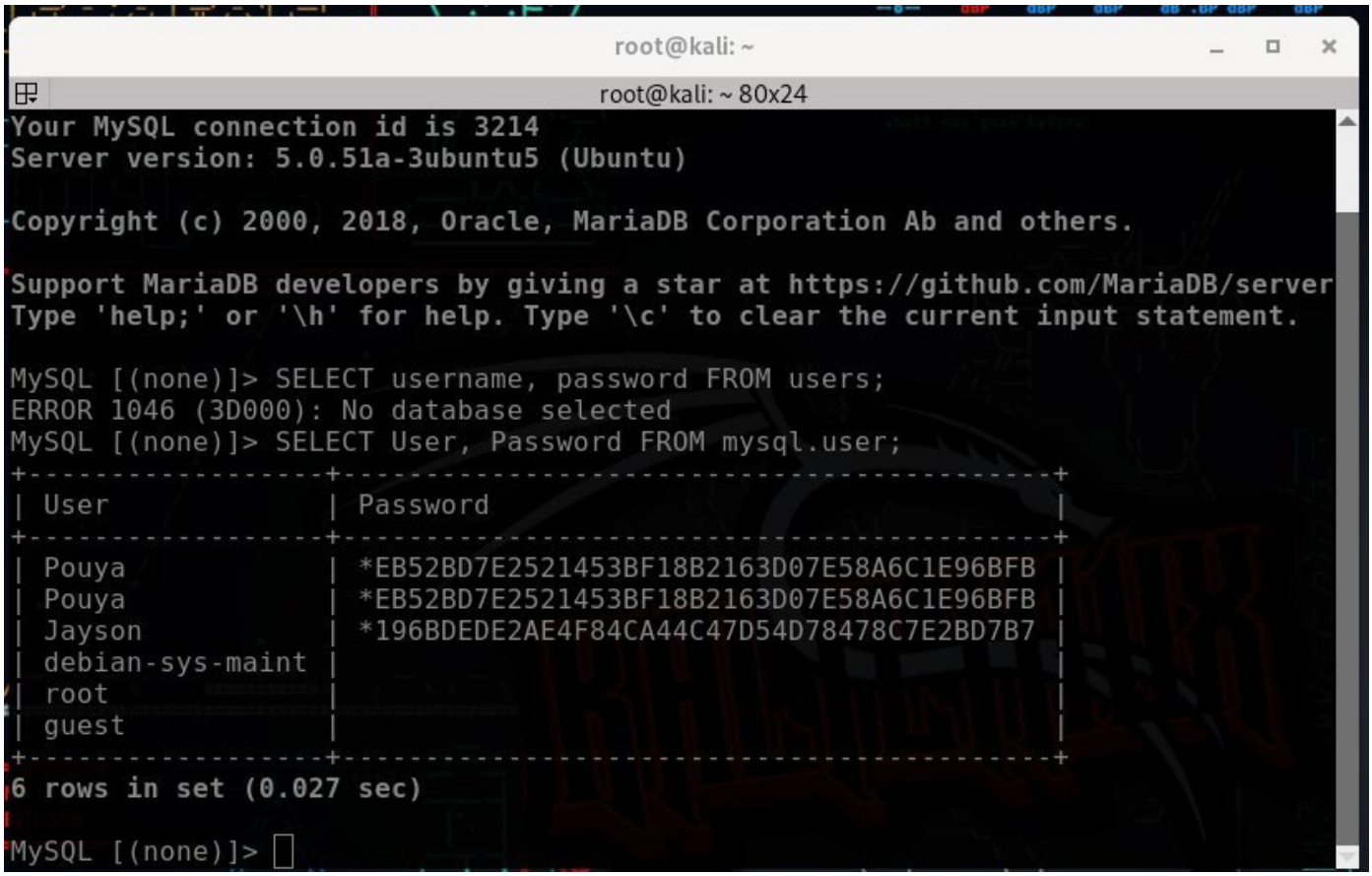
This service can be accessed using

Cadaver allows users to upload and download files from the server. While it restricts permissions for changes made within the dav folder, confidential information can be leaked if configured without a password.

Databases

Service: MySQL
Port: 3306
Risk Level: Critical
Nmap result:
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5
| mysql-info:
| Protocol: 10
| Version: 5.0.51a-3ubuntu5

Almost every server holds a type of database, this host has a MySQL database as well.



I successfully gained access to the MySQL server using the root account with a blank password. I now have access to this database, which is running a DVWA (Damn Vulnerable Web Application).

The accounts listed in the database were created through this web application. The passwords can be cracked using tools such as John the Ripper or Hashcat.

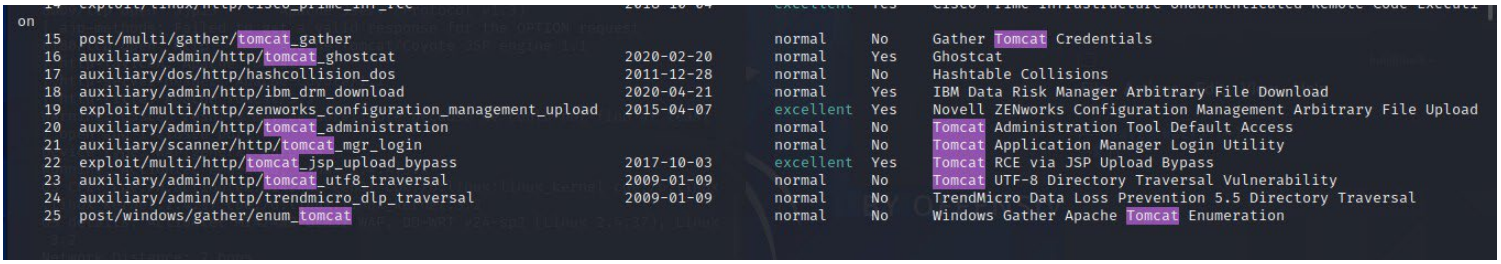
Web Services

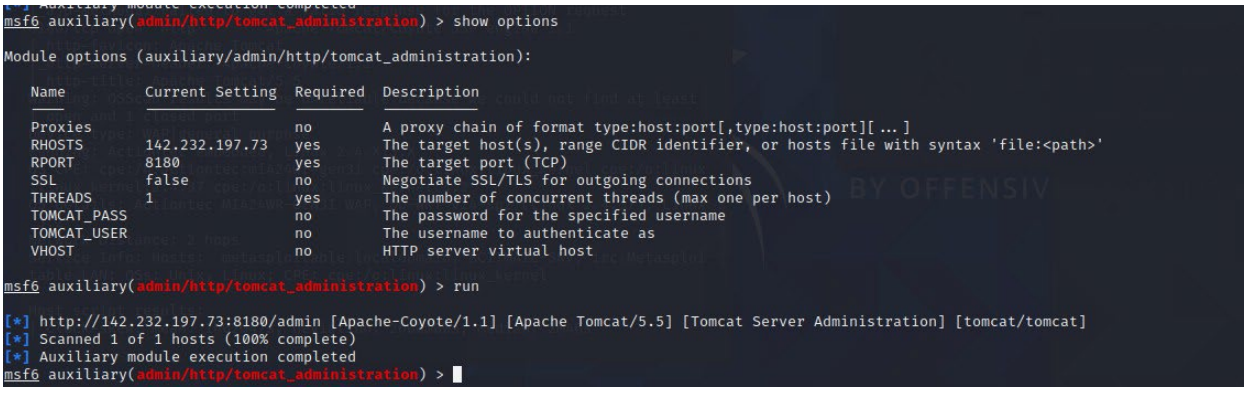
Host 142.232.197.73 has multiple web application running on it, such as DVWA, and Mutillidae which has all the vulnerabilities included in OWASP top 10 and can be exploited in multiple ways.

Service: Tomcat
Port: 8180
Risk Level: High
Nmap Result:
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

Apache Tomcat is an open-source web server and servlet container for Java applications. It supports Java Servlets and JavaServer Pages (JSP), allowing the creation of dynamic web applications. Tomcat serves static content, deploys WAR files, and is lightweight, cross- platform, and widely used in development and production for Java-based solutions.

Using MSF console to determine whether the default Tomcat credentials are used.

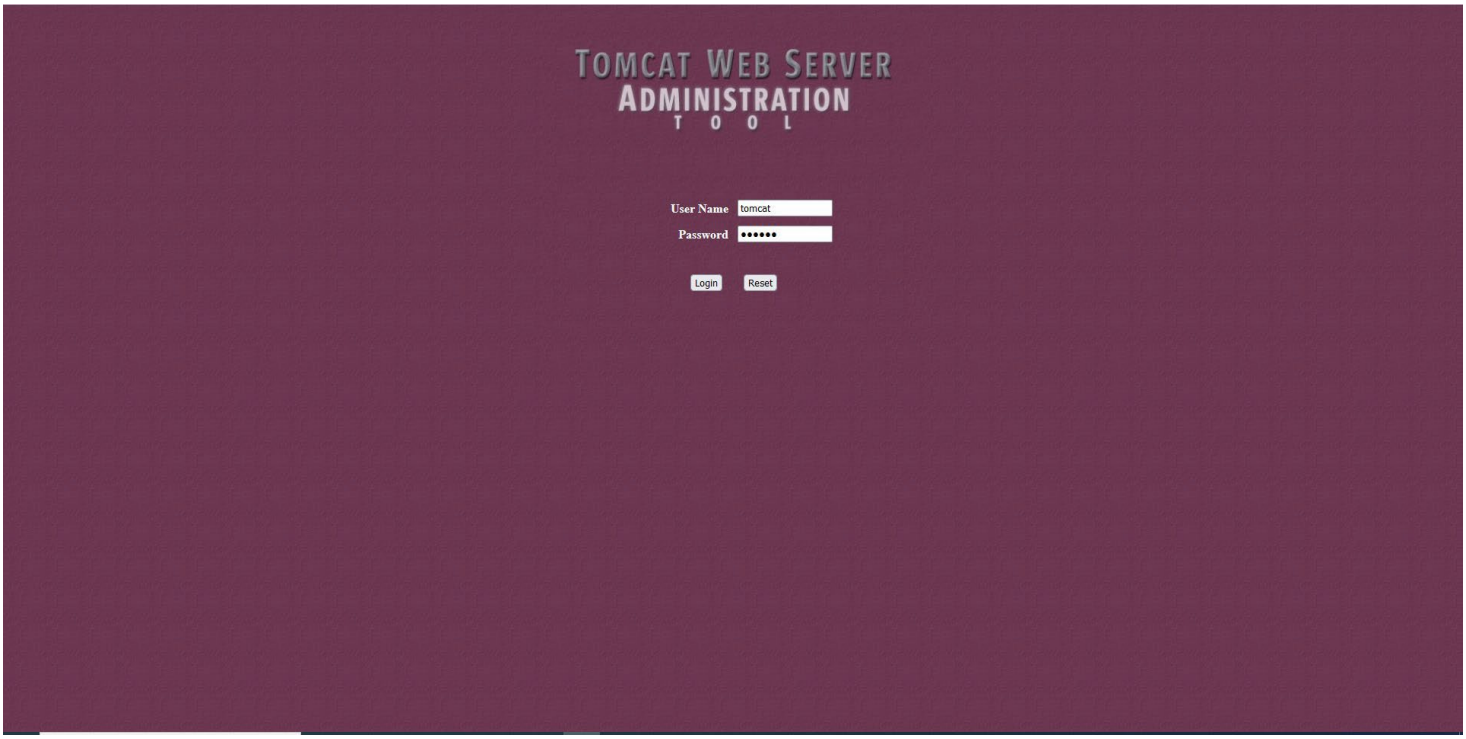




```
[-] 142.232.197.73:8180 - LOGIN FAILED: manager:s3cret (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: manager:vagrant (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:admin (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:manager (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:role1 (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:root (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:tomcat (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:s3cret (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: role1:vagrant (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:admin (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:manager (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:role1 (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:root (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:tomcat (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:s3cret (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: root:vagrant (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 142.232.197.73:8180 - Login Successful: tomcat:tomcat
[-] 142.232.197.73:8180 - LOGIN FAILED: both:admin (Incorrect)
[-] 142.232.197.73:8180 - LOGIN FAILED: both:manager (Incorrect)
```

It is using default username and passwords.

Successful login to Tomcat



Apache Tomcat/5.5

The Apache Software Foundation
http://www.apache.org/

Administration
Status
Tomcat Administration
Tomcat Manager

Documentation
Release Notes
Change Log
Tomcat Documentation

Tomcat Online
Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC

Examples
JSP Examples
Servlet Examples
WebDAV capabilities

Miscellaneous
Sun's Java Server Pages Site
Sun's Servlet Site

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

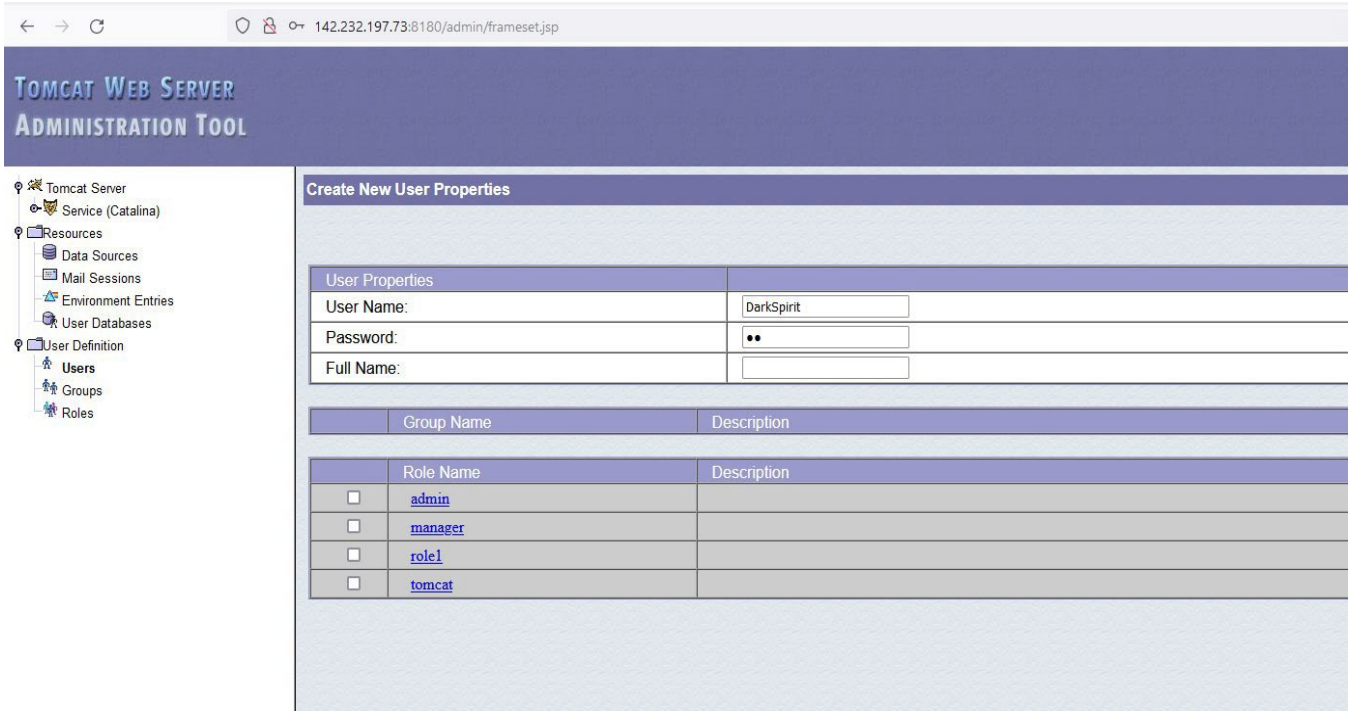
Tomcat mailing lists are available at the Tomcat project web site:

- **users@tomcat.apache.org** for general questions related to configuring and using Tomcat
- **dev@tomcat.apache.org** for developers working on Tomcat

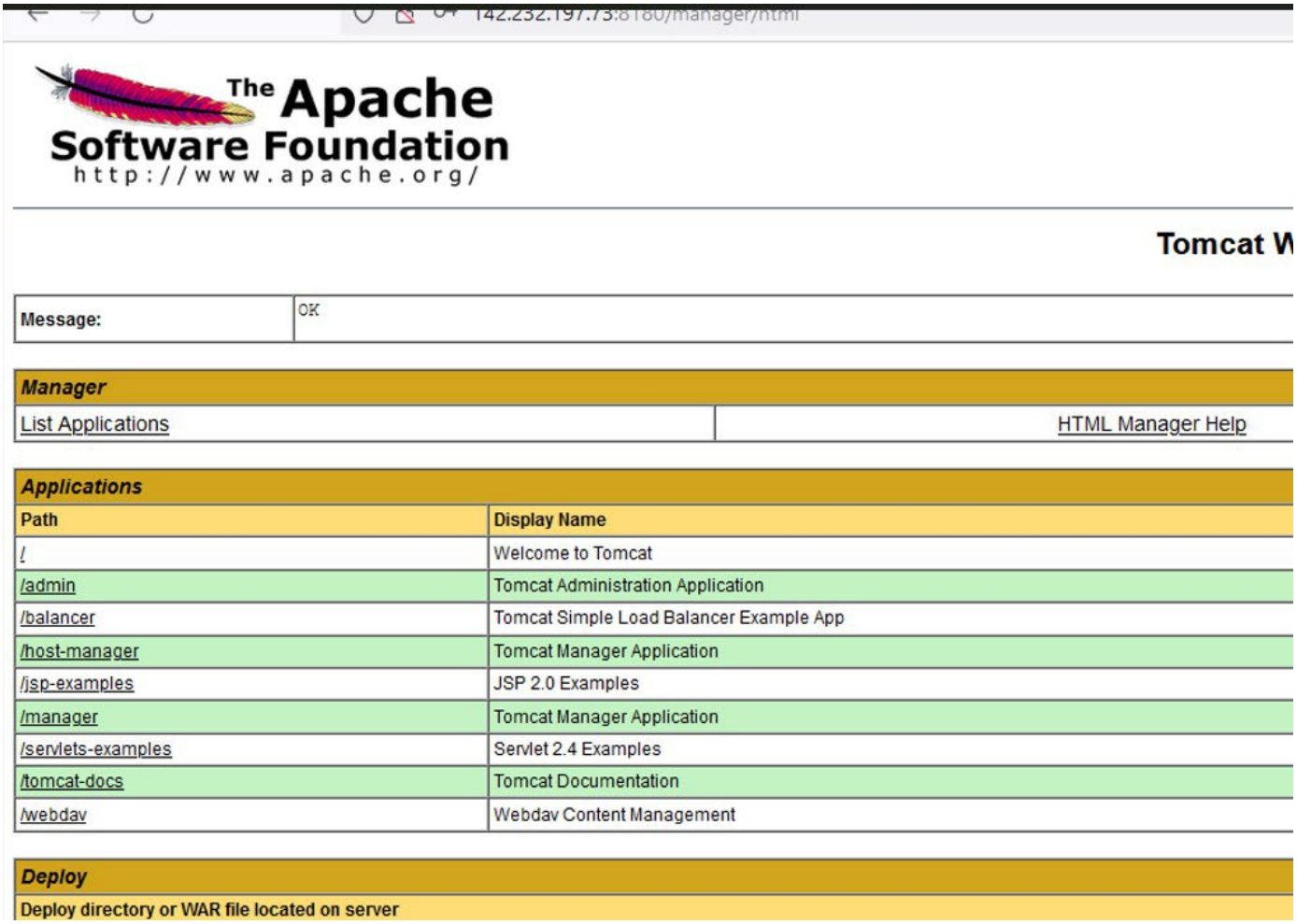
Thanks for using Tomcat!

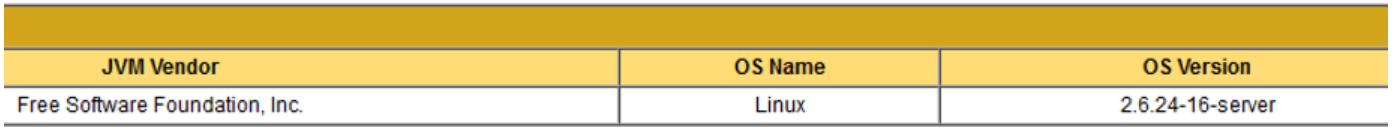
Powered by TOMCAT

Further exploring, I noticed there is an administrative tool, which uses the same credentials, it allowed me to create and remove users and assign roles to them.

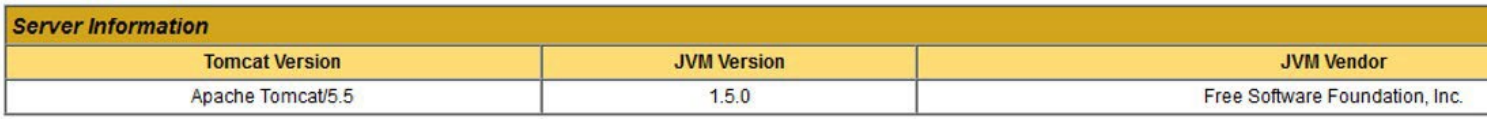


It reveals much more information about the Apache server.





JVM Vendor	OS Name	OS Version
Free Software Foundation, Inc.	Linux	2.6.24-16-server



Server Information		
Tomcat Version	JVM Version	JVM Vendor
Apache Tomcat/5.5	1.5.0	Free Software Foundation, Inc.

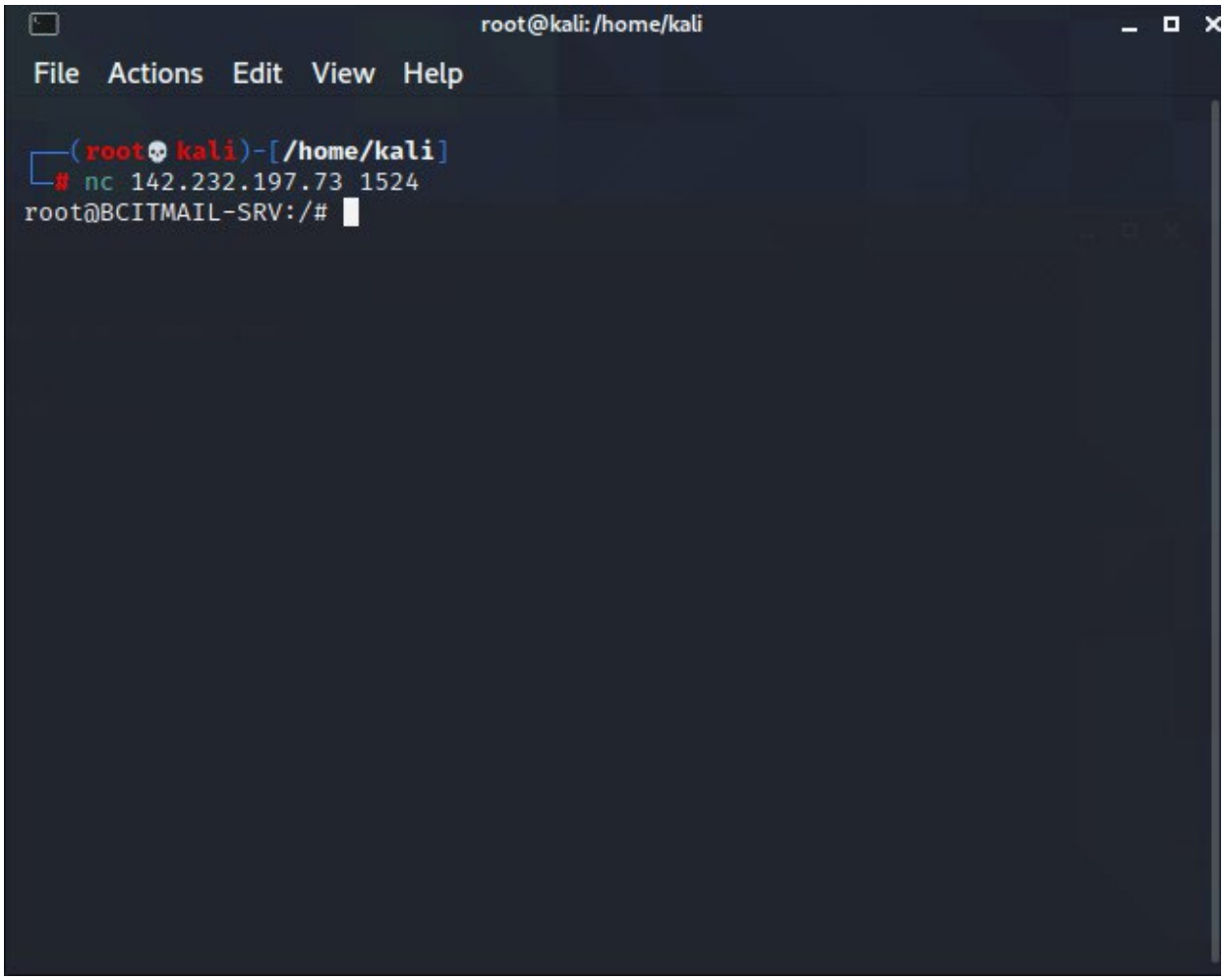
Backdoor

Service: Backdoor
Port: 1524
Risk Level: Critical

Nmap Result: 1524/tcp open bindshell Bash shell (**BACKDOOR**; root shell)

This backdoor allows a direct root connection to the Metasploitable machine, it will give full access of the server to an attacker.

```
513/tcp  open  login?
514/tcp  open  shell       Netapp ONTAP rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Bash shell (**BACKDOOR**; root shell)
2049/tcp open  nfs         2-4 (RPC #100003)
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
```



Security Steps

After reviewing all the vulnerabilities, I can conclude that most of them can be mitigated by updating services to the latest version and closing unnecessary ports. Regular auditing is essential in this context, as we discovered a running backdoor that could allow unauthorized access to the system.

To prevent access via SSH and VNC, it's important to implement strong passwords and unique usernames. Additionally, firewall rules should be established to block foreign IP addresses from accessing services like SSH/VNC and connecting to databases.

Antimalware software must also be installed on the host machine.

Host 2 – Windows 7

IP Address: 142.232.197.72

Objective: To enumerate active services and exploit potential vulnerabilities.

Overview

Host 142.232.197.72 is a Windows 7 PC with an IDS honeypot system. This system prevents attackers from enumerating the system and discovering active services by presenting well-known false open ports that host fake services. This tactic distracts attackers and conceals the actual services running on the system. The IDS honeypot operating on the Windows 7 machine is known as KFSensor.

Enumeration & Methodology

- Initial Nmap Scan: Conducted an initial Nmap scan using standard options to identify open ports and running services.
- Results of Initial Scan: The scan output indicated 1,000 ports in an ignored state, suggesting the possibility of an IDS or an environment configured to obscure actual services.
- Targeted Nmap Scan: A second Nmap scan was conducted with more specific options to determine actual active services. That provided a list of 100-200 services running, which indicated there is indeed a honeypot.
- Final Nmap Scan: To filter out actual open ports and services the final Nmap scan was performed, that showed fewer and active services (such as port 80, 3389, 445).

The Final Nmap Scan shows actual services.

nmap -sC 142.232.197.72

Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-24 21:15
CDT Nmap scan report for 142.232.197.72

Host is up (0.021s latency).
Not shown: **906 filtered tcp ports (no-response)**

PORT STATE SERVICE

80/tcp open http

3389/tcp open ms-wbt-server
|_ssl-date: 2024-10-25T02:15:44+00:00; -11s from scanner time.
| ssl-cert: Subject: commonName=mypc7-PC
| Not valid before: 2024-10-23T16:34:22
|_Not valid after: 2025-04-24T16:34:22

135/tcp open msrpc

445/tcp open microsoft-ds
Host script results:
| smb2-time:
| date: 2024-10-25T02:15:40
|_ start_date: 2024-10-10T19:52:02
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h44m50s, deviation: 3h30m04s, median: -11s
| smb2-security-mode:
| 2:1:0:
|_ Message signing enabled but not required
| smb-os-discovery:
| OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
| OS CPE: cpe:/o:microsoft:windows_7::sp1
| Computer name: mypc7-PC
| NetBIOS computer name: MYPC7-PC\x00
| Workgroup: WORKGROUP\x00
|_ System time: 2024-10-24T19:15:45-07:00

Nmap done: 1 IP address (1 host up) scanned in 353.12 seconds

This is not the complete scan, fake ports are omitted, and only legit ones are shown in the result. The full result is attached at the end of the report.

Actual Services

By looking at the nmap scan we can learn that the operating system of the device is Windows 7Ultimate 7601 Service Pack 1, PC's name is mypc7.

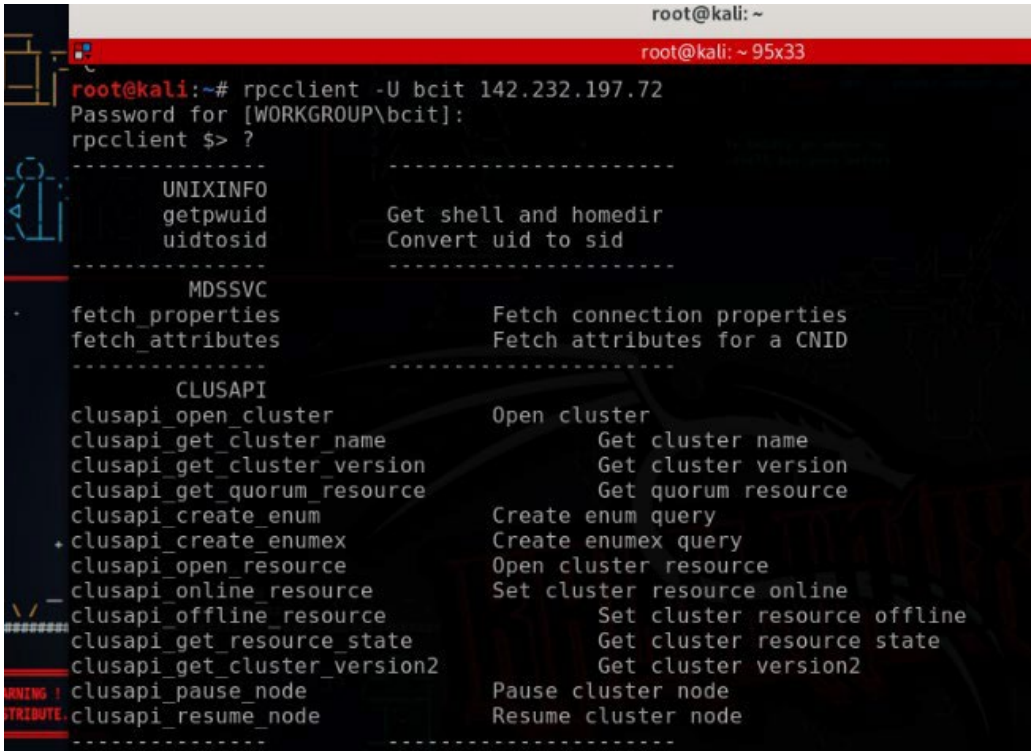
It has services like MSRPC, SMB, RDP and a Webpage running.

- Port 80 (http): IIS welcome page.
- Port 135 (MSRPC): Enables Windows inter-process communication for remote services; essential for administrative tasks.
- Port 445 (SMB): Server Message Block is used for files and printer sharing.
- Port 3389 (RDP): Allows remote desktop access to Windows systems, the device is currently listening on that port.

Exploitation

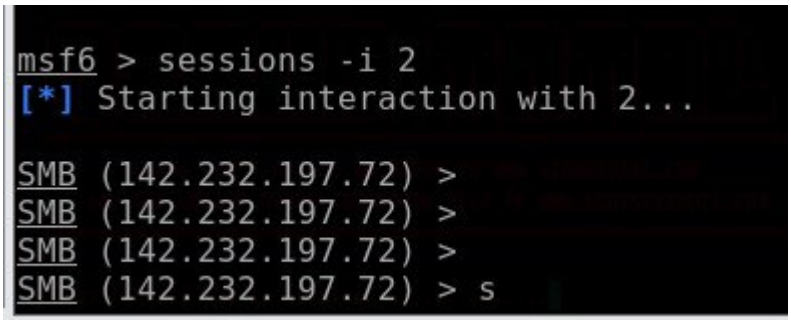
Based on the findings of ports, we can try SMB, MSRPC and RDP services to gain access to Windows 7. MSRPC and SMB will not provide much access to the system.

Using rpcclient to interact with MSRPC



As mentioned, there is not much, the commands will just help you to gather more information about the System.

Using SMB login to interact with the SMB service.



Metasploit's auxiliary smb login was used to connect but that didn't provided much information as well.

RDP Connection

Successful login provides complete control over Windows 7.

Risk Analysis

Risk Level: High

Impact: Unauthorized access to RDP could allow an attacker to control the system directly, potentially leading to data leakage, system manipulation, or lateral movement within the network.

To establish a connection, it is essential to avoid using Microsoft's RDP client and Windows itself. The reason for this is that their software employs a higher level of encryption and authentication that is not compatible with older versions of Windows 7.

Attempting to connect using Windows 7 PCs with Microsoft's RDP client will result in a CredSSP Encryption Error. Even configuring group policy and registry settings will not resolve this issue.

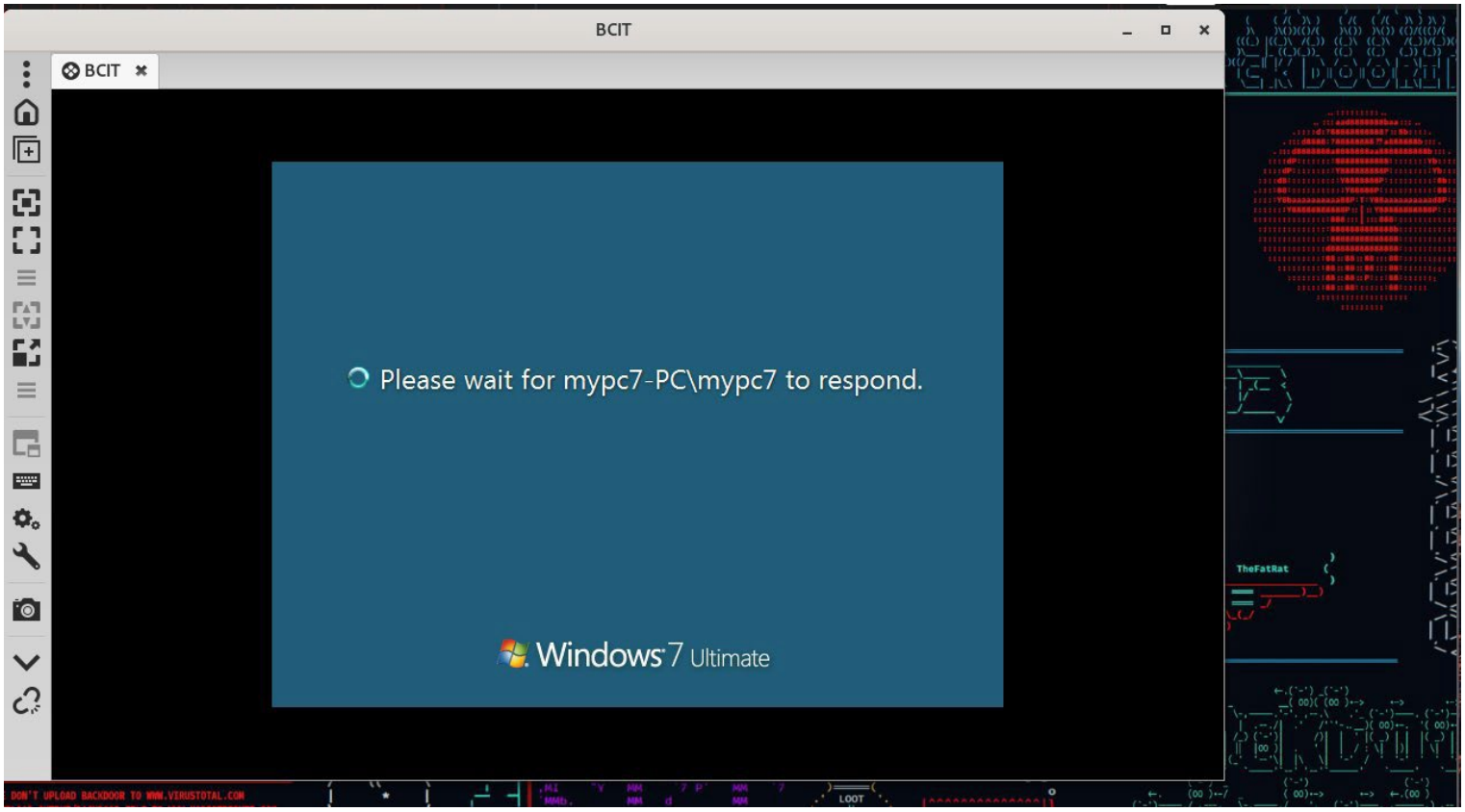
Instead, the **Remmina RDP client** can be used as it offers an option for Windows 7-based authentication, which allows a connection to the device. This tool is Linux-based.

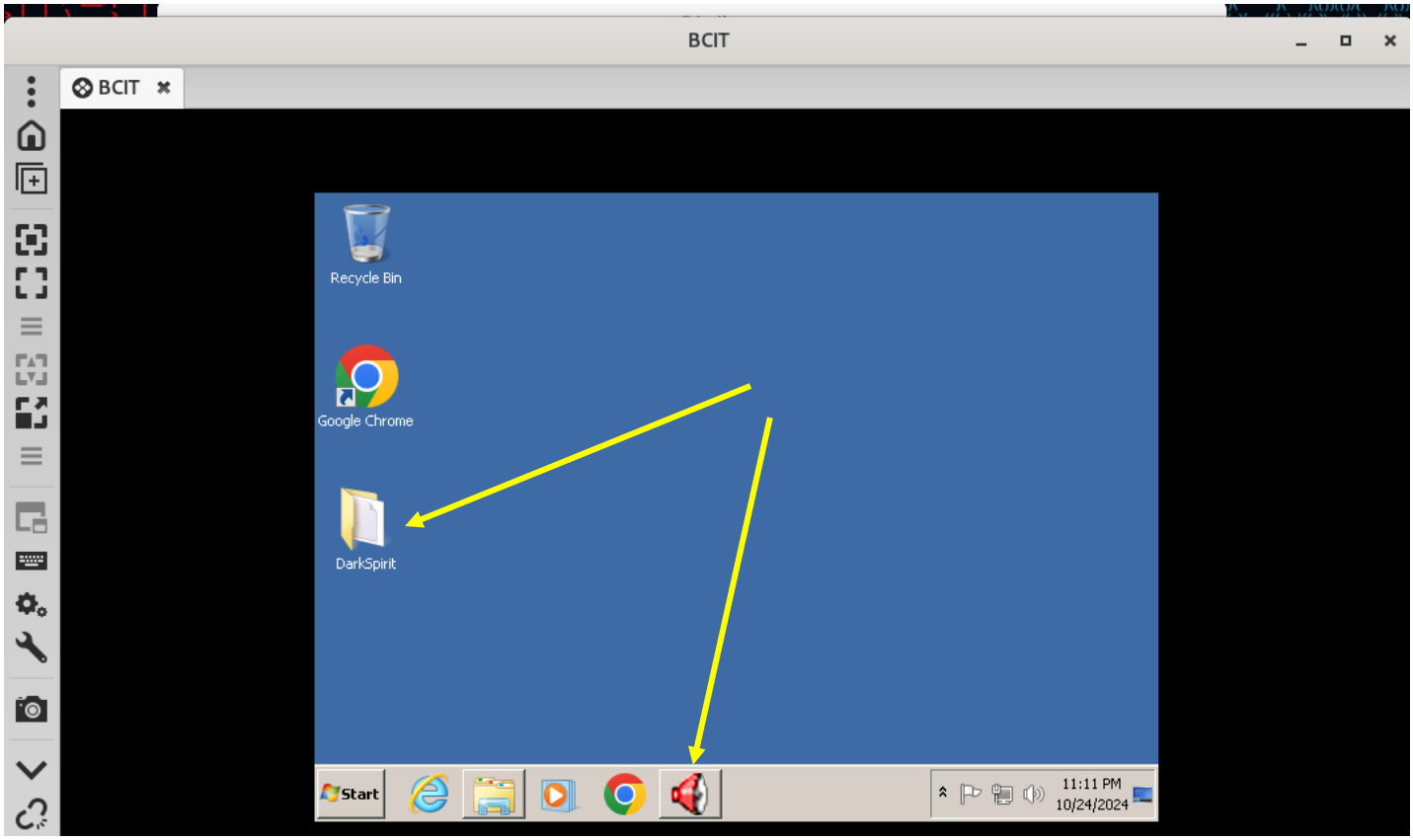
For authentication, we can use the credentials found in the previous machine (142.232.197.73). The credentials were

username: bcit

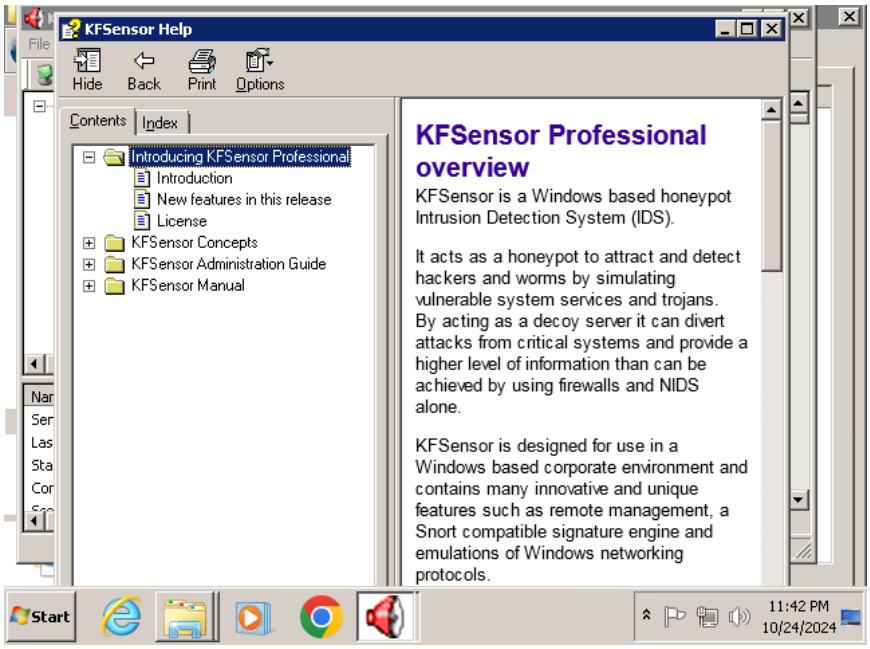
password: 112233

Successful login to mypc7 (Windows 7)

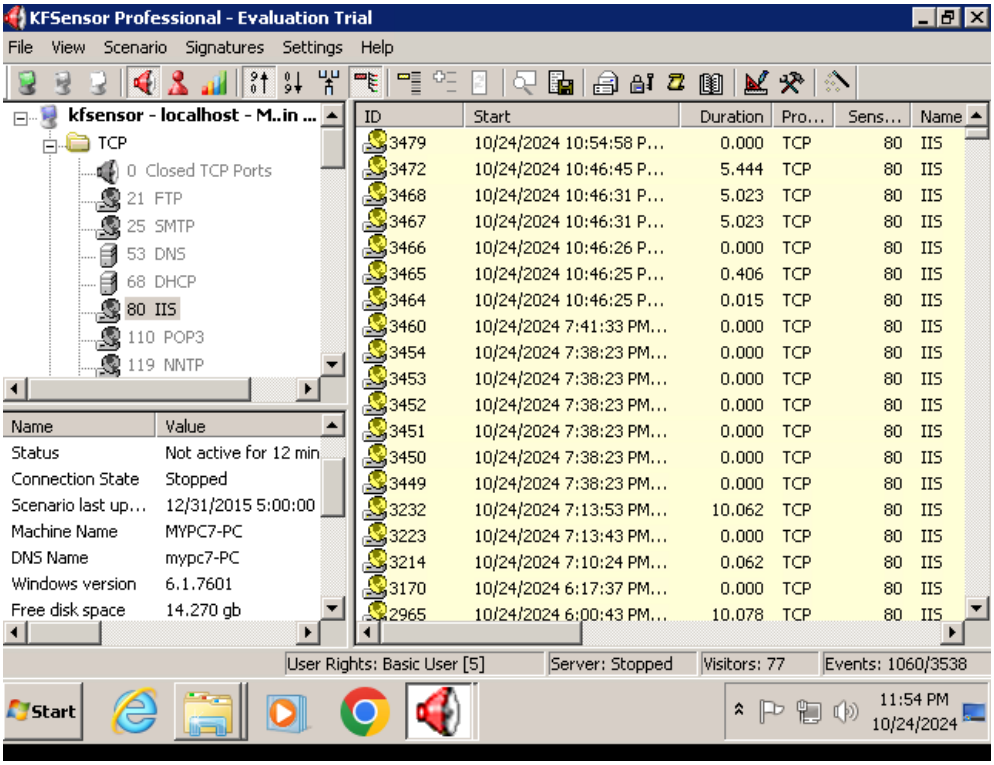




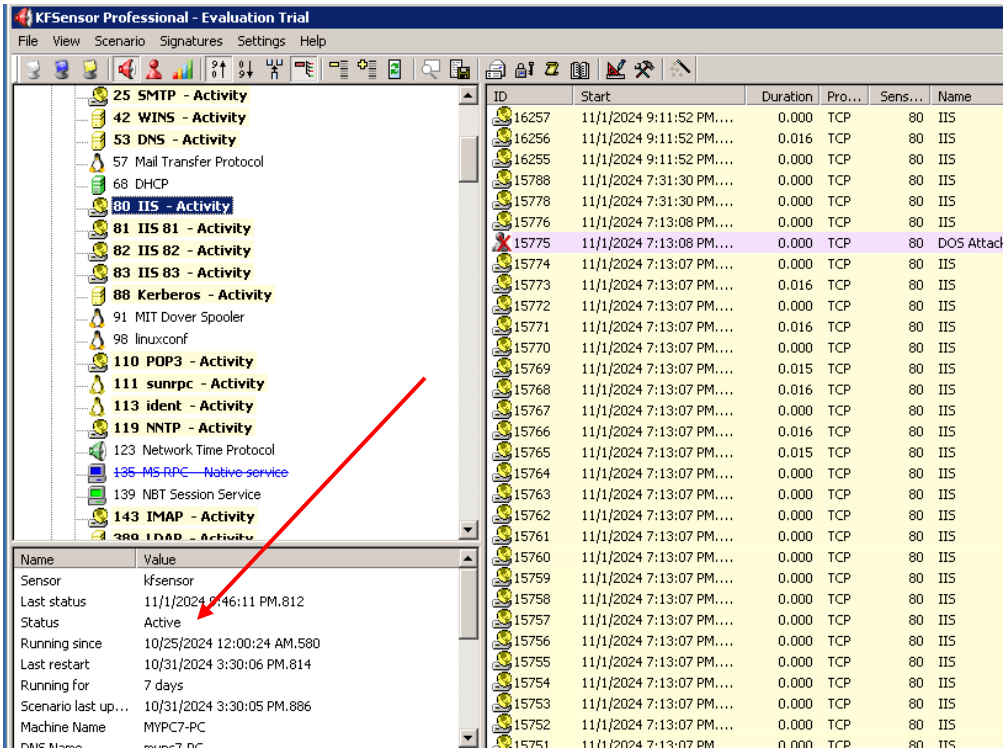
After successfully logging in, I discovered that this computer was running KFSensor, a honeypot IDS. This piqued my interest, and upon further exploration, I was able to see all the fake ports that appeared in the Nmap results.



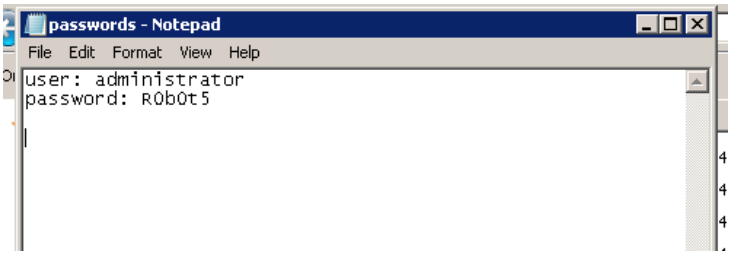
Fake services and ports



For some reason, port 80 was down, which affected the IIS webpage. This was why people were unable to access the site and thought the server was not working. I had the opportunity to learn how KFSensor software works, and I successfully restarted the services and reactivated port 80.



Further findings helped me to get new credentials.



Summary

The host 142.232.197.72 has been identified as a Windows 7 computer running a honeypot Intrusion Detection System (IDS) service called KFSensor. Credentials discovered on the host 142.232.197.73 were used to authenticate via the Remote Desktop Protocol (RDP) client. The Remmina RDP client was selected due to its support for legacy Windows 7 authentication. Additionally, new credentials were discovered in the password file, which can be used on another machine.

Host 3 – Ubuntu Server

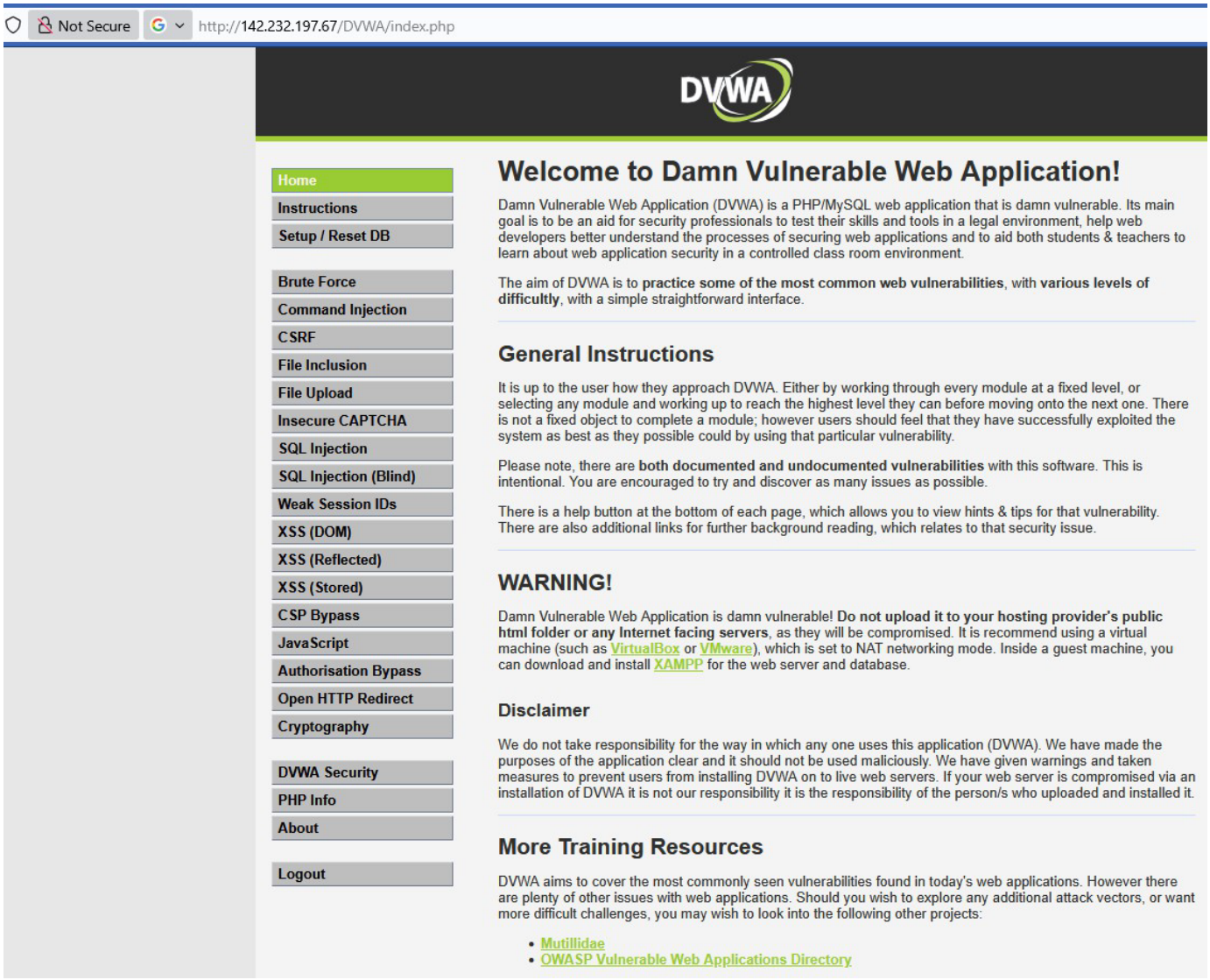
IP Address: 142.232.197.67

Objective: Testing Web Application security.

Overview

Host 142.232.197.67 is an Ubuntu server running DVWA web application. It has multiple vulnerabilities such as XSS, Command Execution, File Inclusion, File Upload and more.

DVWA is using default username and password that is admin and password.



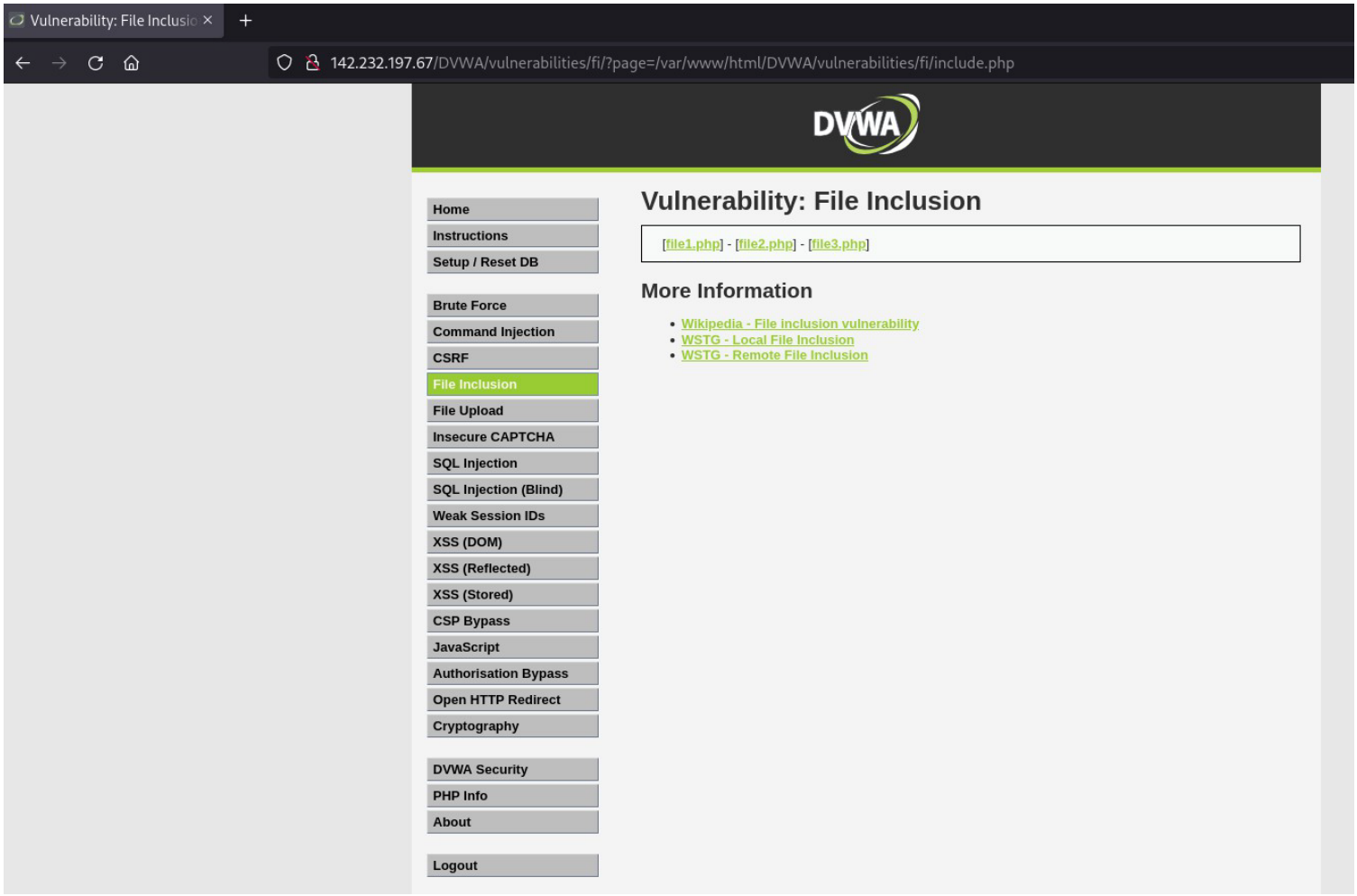
Exploitation

File Inclusion:

File Inclusion vulnerabilities allow an attacker to read and execute files on the victim server. There are two types of File inclusion

1. Local File Inclusion: which allows an attacker to explore or surf the local file system and execute files from there.
2. Remote File Inclusion: Allow an attacker to execute files from the remoter server to the target server.

In the example below I can specify the parent directories with the php page and still able to load it, which means I can also load other files located in the / directory.



Result of file inclusion.

Trying to go back f directories and going into /etc.

After five directories we are in the / directory.

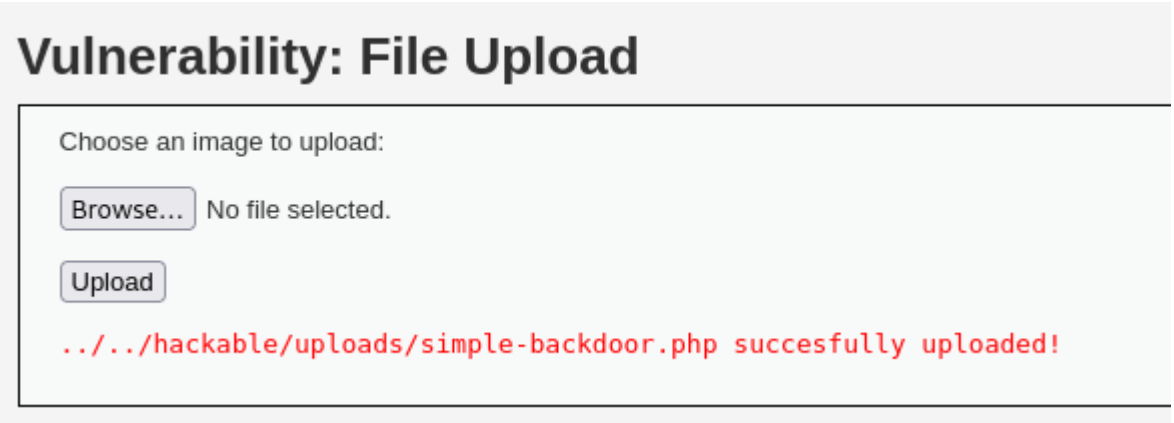
Result of passwd file.

Similar to that this vulnerability can allow an attacker to mention a remoter servers address and run a file from there on to this server, it can be remote connection via reverse shell or any other files.

File Upload:

File upload vulnerabilities allow attackers to upload files with incorrect extensions, such as a .pdf instead of a .jpg, if the server improperly accepts them then it is vulnerable.

This option here should only take files with jpg extension, but it will accept a php file.



As we can see, I can upload a php file, such as a backdoor and execute it to have a shell.

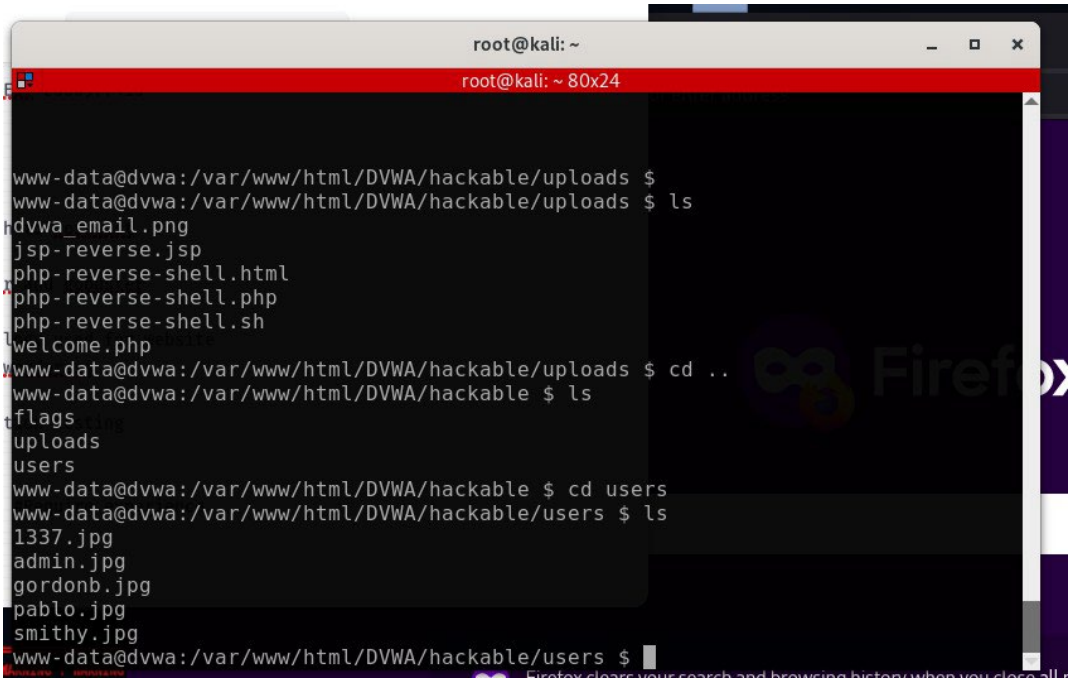
I am using the **weevely** tool to create a php shell which only I can access.

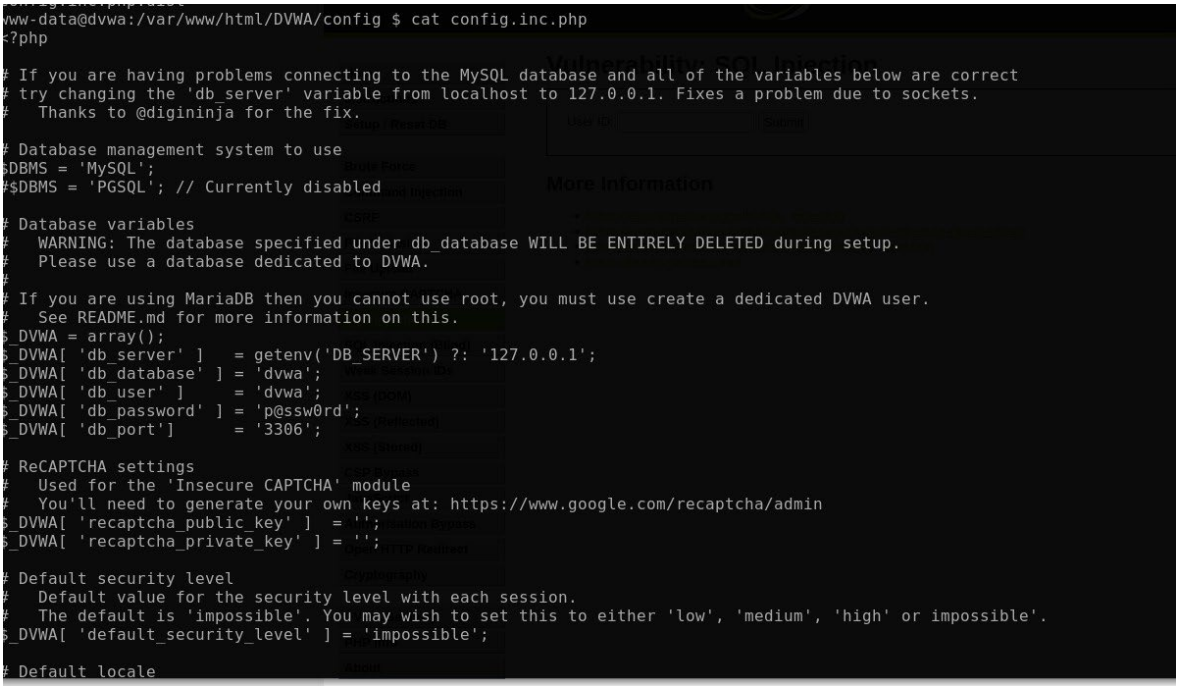
The link of the shell: http://142.232.197.67/DVWA/hackable/uploads/welcome.php 2356977413

Entering this link on the browser will execute the php file.

Usage: weevely http://142.232.197.67/DVWA/hackable/uploads/welcome.php 2356977413

successful shell execution.



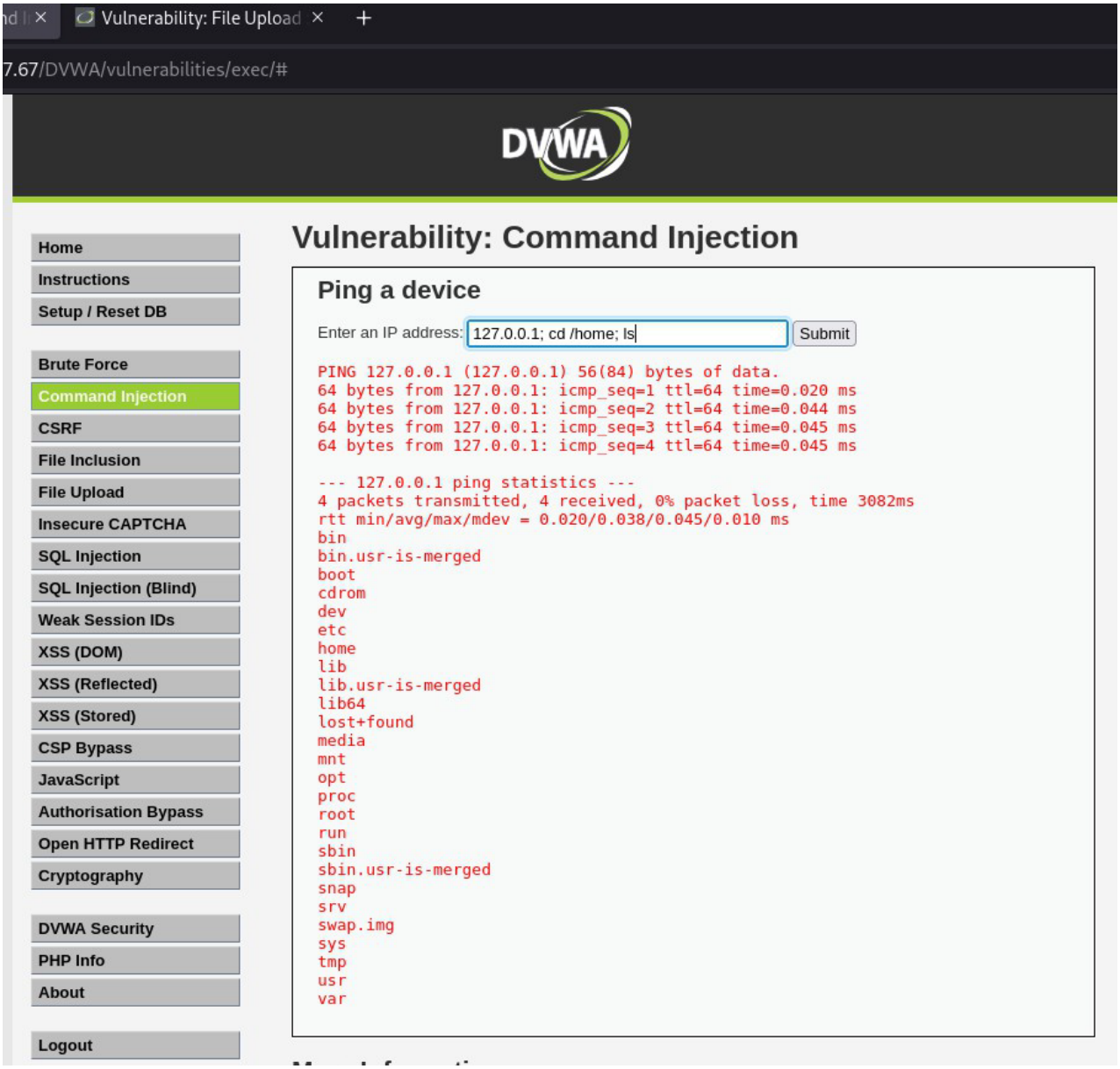


The above information reveals the database credentials.

This tells how important it is to prevent this kind of attack, it can cause greater damage to the web server and the service it is serving. This was just a demo, advanced backdoors can do many more things.

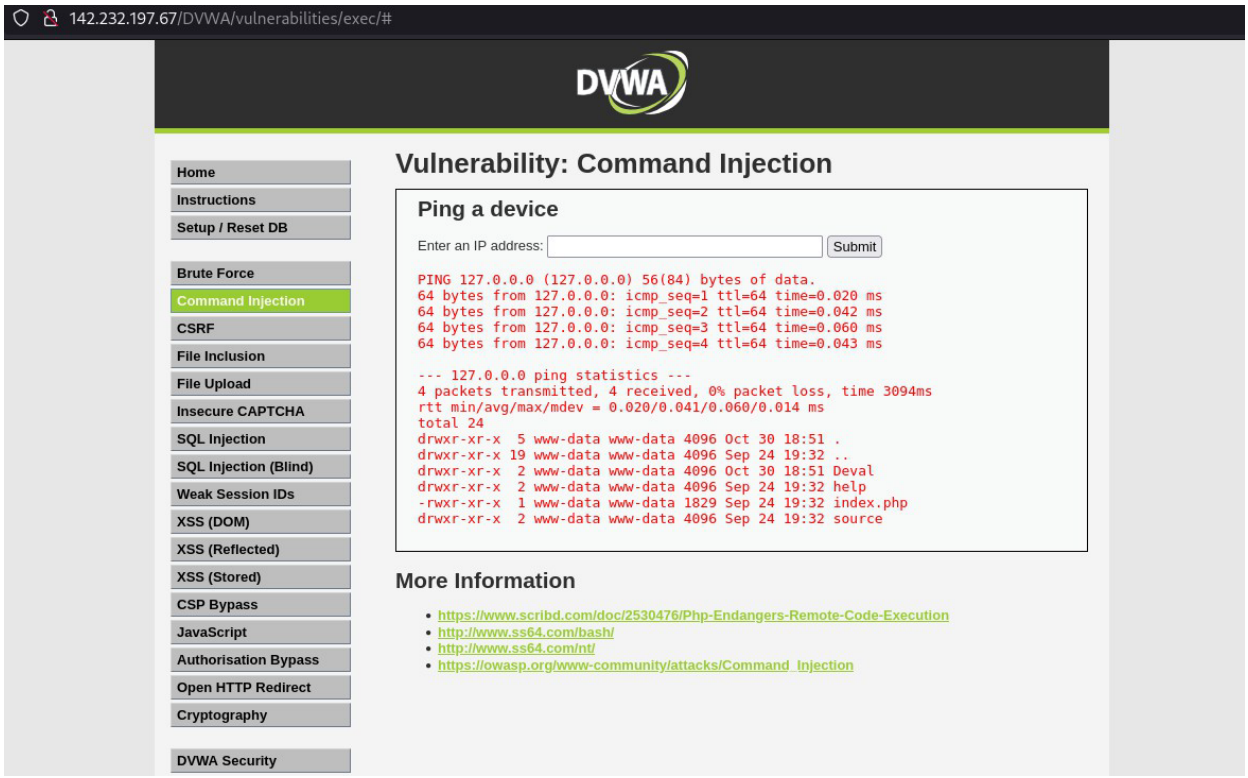
Command Execution

Command execution vulnerability in DVWA allows an attacker to run arbitrary system commands on the server through user input. If the application does not properly validate or sanitize this input, an attacker can exploit it to execute malicious commands, potentially compromising the server.



In the example above, we can see how with the ping I was able to list the files in the **/home** directory.

This input field will run any command, but you will be restricted because of the permissions are set.



This vulnerability can allow you to gather information about your target and help you design your attacking strategies with it.

HOST 4 – Honeypot (T-Pot)

IP Address: 142.232.197.39

Objective: To enumerate active services and exploit potential vulnerabilities.

Overview

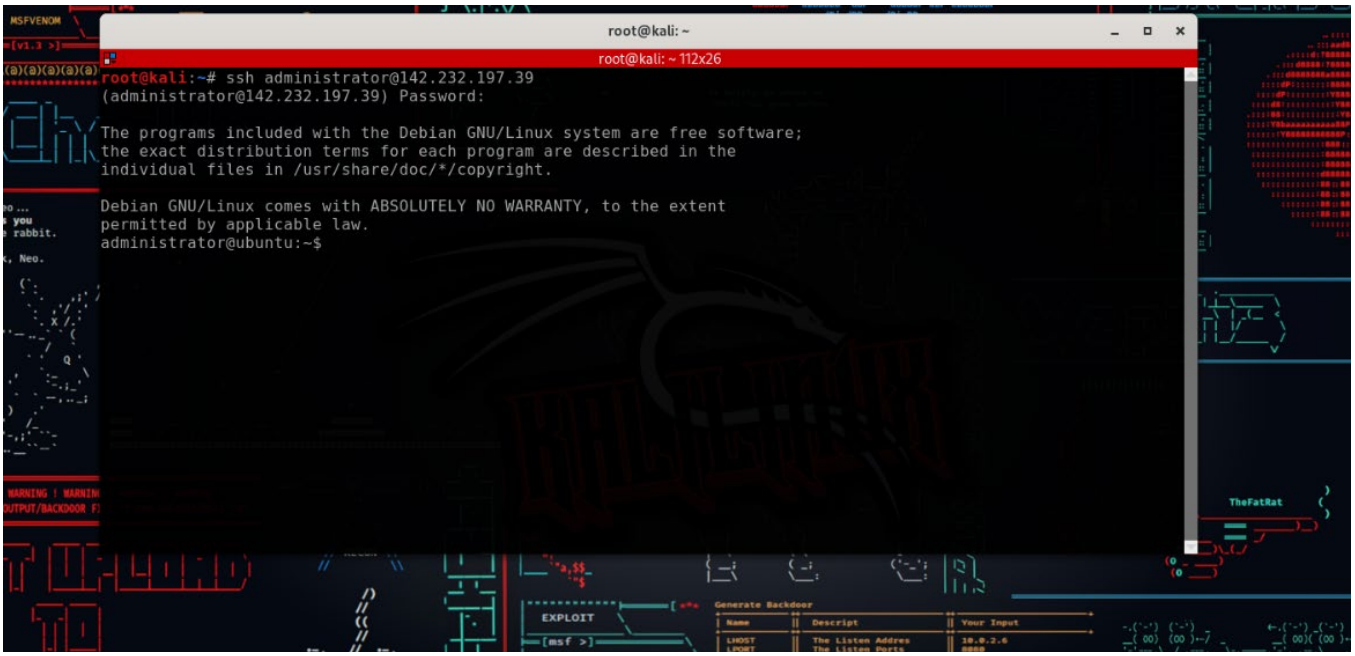
This device was found to be a honeypot. After interacting with it several times using Nmap and SSH, it is obvious that this system is a honeypot, specifically a T-Pot.

T-Pot is the all-in-one, optionally distributed, multi-arch (amd64, arm64) honeypot platform, supporting 20+ honeypots and countless visualization options using the Elastic Stack, animated live attack maps and lots of security tools to improve the deception experience further.

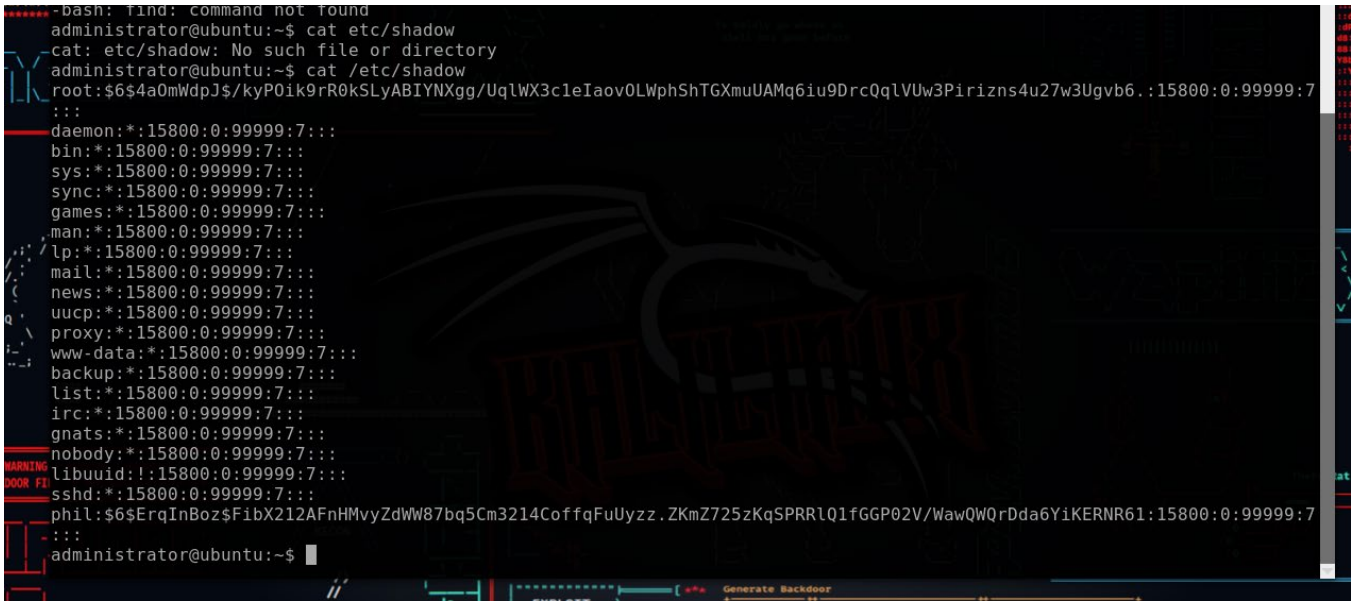
Enumeration & Methodology

- Initial Nmap Scan: Conducted an initial Nmap scan using standard options to identify open ports and running services.
- Results of Initial Scan: The scan output indicated multiple well-known ports, but Nmap was unable to get the banner, and scripts were failed. That was a possible indication that something was off. Every time a run the scan, Nmap would highlight show me a different name of honeypot.
- Targeted Nmap Scan: Among the open ports, I targeted SSH service to see how it responds.

SSH Service:



To log in, I used the credentials found in the previous machine, Windows 7. was able to gain access to it and could read the shadow file.



After the session time out, when I tried to reconnect, those credentials did not work. Therefore, I decided to guess a random username and password, and I got access. This behavior indicated that it is an SSH honeypot. It was just simulating the SSH service.

During my Nmap scans, I noticed different honeypots each time, indicating that this machine is running multiple honeypots, possibly a T-Pot.

THE END