PROJECT REPORT: Vulnerability Discovery and Exploitation [Kian Mojami]

Northern Virginia Community College

# Introduction

The two vulnerabilities I selected are the UnrealIRCd Backdoor and Bind Shell Backdoor. I chose these vulnerabilities because they are both critical vulnerabilities with high CVSS scores that provide easy backdoor access into a system. Because of this, understanding how the exploit is executed and how to patch it to stop it from happening is vital.

# Lab Topology

The topology of this lab includes two Linux machines as the attack and target machines. The attack machine is a Kali Linux machine which runs Debian, and the target machine is the Metasploitable 2 machine which runs Ubuntu Linux. The Nessus vulnerability scanner runs on the Kali Linux machine to scan the Metasploitable machine, so it can read the vulnerabilities and exploit them. Both of these machines run on my Windows laptop which is running them using VMware. My laptop is then connected to my router which connects to the internet and is protected by a firewall.

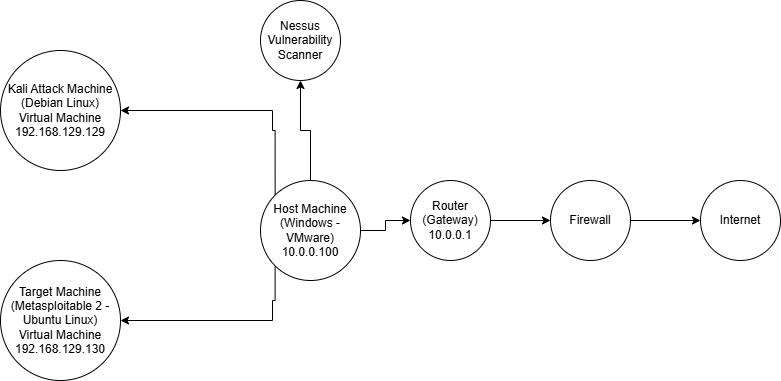


Figure 1: Logical topology of lab

# Mapping and Scanning

The two network scanners I used were Nmap and Masscan which show the open ports and services on a specific system. The vulnerability scanner software I used was Nessus to list out the vulnerabilities and their scores based on severity. Each of these tools helps with finding openings to exploit the system and gain root access.

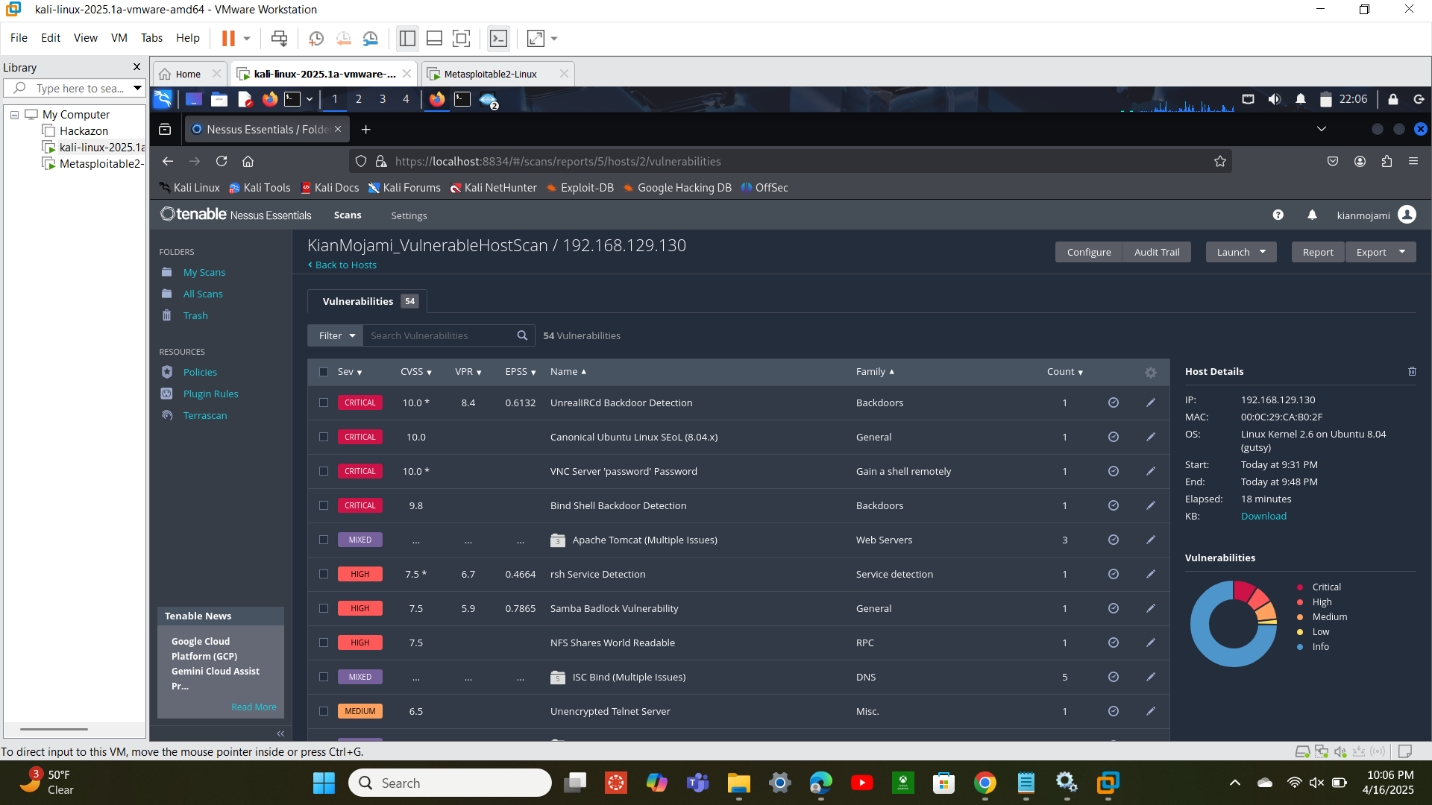


Figure 2: Vulnerability scan of target machine using Nessus

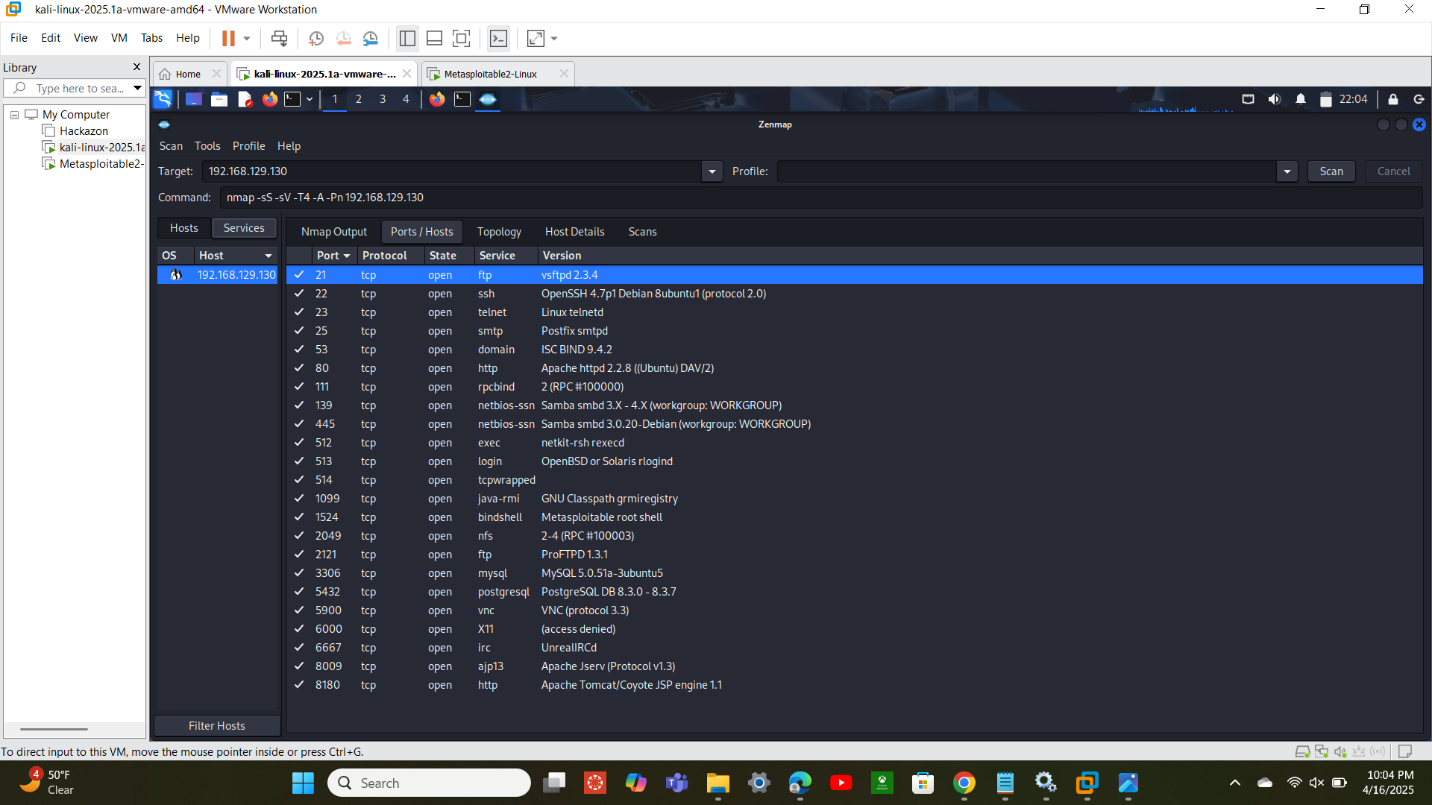
****

Figure 3: Nmap scan of target machine

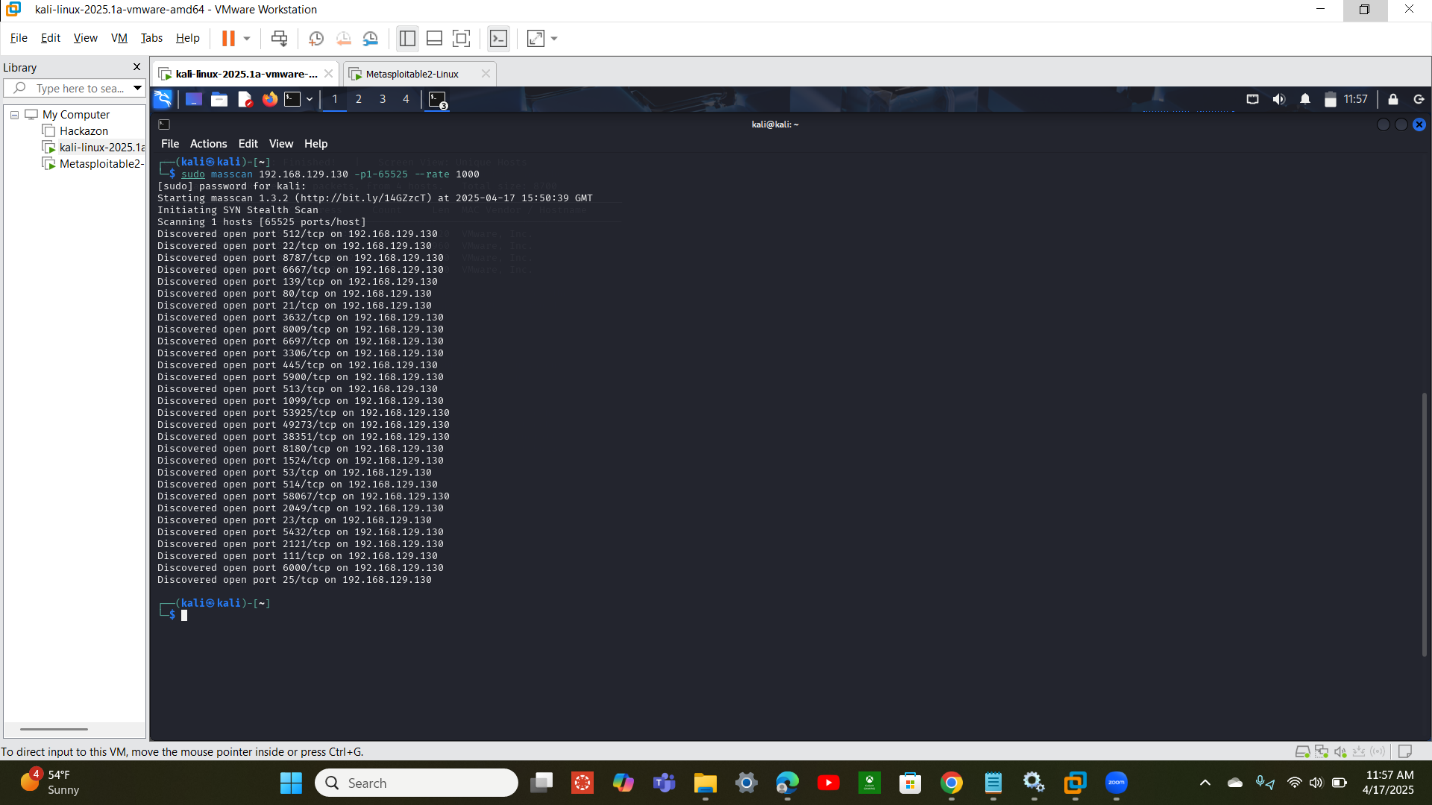
****

Figure 4: Masscan of target machine

# Vulnerability Research

The UnrealIRCd backdoor vulnerability exists because of a supply chain attack where specifically version 3.2.8.1 of this software was replaced with a version that contained a backdoor. This software, which is an internet relay chat daemon, is used for real-time communications such as private messaging and file sharing which was used by many people. The exploit worked by entering a specific string to bypass the normal processing of IRC and instead treating it as a shell command. Whatever command was sent after this specific string was then sent as the root user allowing the user to execute anything they wanted to. The bind shell backdoor vulnerability is used to gain access to a system through an open port that requires no authentication. This allows an attacker to bind a shell to the port while it listens for TCP connections then once a connection is established it can run commands without being authenticated.

# Vulnerability Analysis

CVSS v2 scores the UnrealIRCd vulnerability a 10/10 on its scale which marks it as a critical vulnerability. This is because as the command is treated as a system shell command, it allows root privileges to be accessed. It can also be done easily by using port 6667 and a Metasploit module which creates a shell on the specified system. You can do this as any type of user on another system and gain full control on another system that uses specifically version 3.2.8.1 of UnrealIRCd. The bind shell backdoor vulnerability scores a 9.8/10 on the CVSS v3 because of the amount of access that can be gained due to no authentication being required. As long as the remote port, like port 1524, is open and listening for communication, an attacker can execute a bind shell to attach to the port so once communication is started it can execute commands without authenticating the user first. This allows any user to gain root access and execute system-level commands which is what makes this vulnerability so critical.

# Vulnerability Exploitation

To exploit the UnrealIRCd vulnerability, I used Metasploit which has a specific module for this vulnerability that uses TCP port 6667 to create a backdoor. To configure this exploit, I set the remote host as the IP address of the target machine. I then set the payload to use a reverse shell for a Linux system to bypass any firewall and set the listener host to the attacker IP address and the listener port to an open port such as TCP 4444. After setting my configurations, I ran the exploit to create a backdoor on TCP port 6667 using a reverse shell to grant me root access to run commands such as hostname and ifconfig.

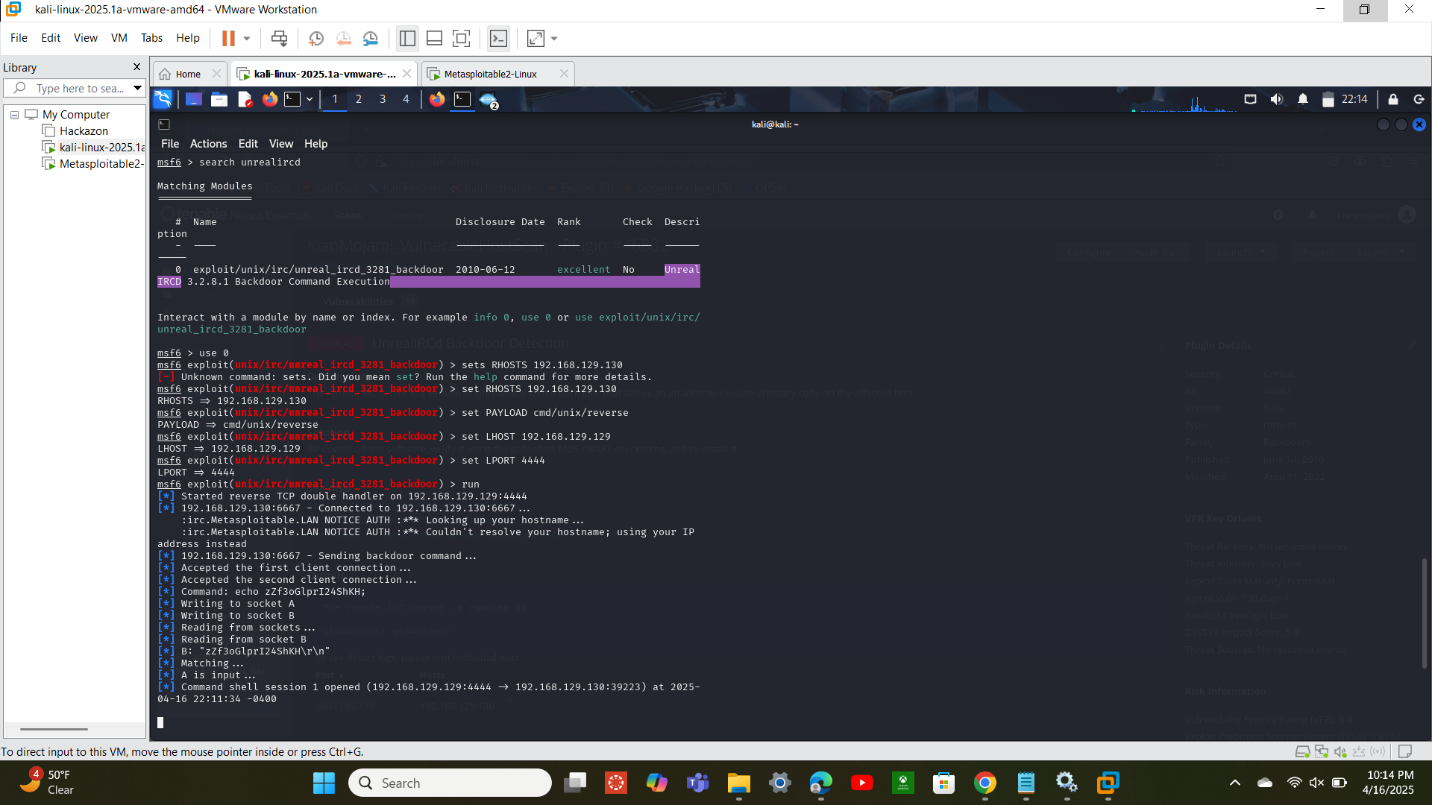


Figure : Configuration of and successful exploitation using RealIRCd exploit

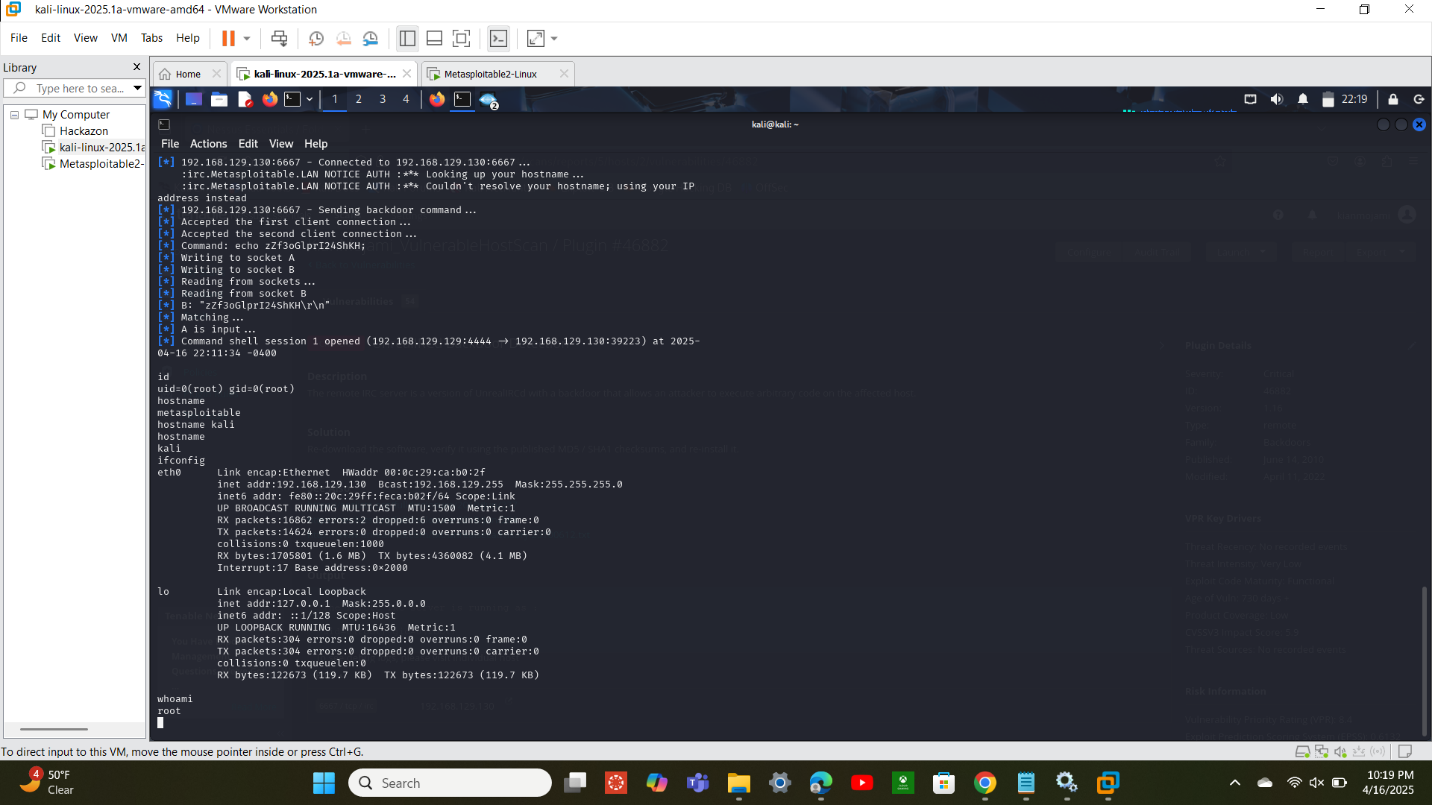


Figure 6: Successful root access using UnrealIRCd exploit

To exploit the bind shell backdoor vulnerability, I first set my exploit as multi handler which allows me to use a payload for a TCP bind shell on a Linux system. Afterwards, I set the remote host as the IP of the target machine and the listener port as the port that was identified as being vulnerable to a bind shell which was TCP port 1524. Once I’ve set this, I can run a bind shell to the open port from the attacking system to gain root access and run system-level commands to view the network and system configurations.

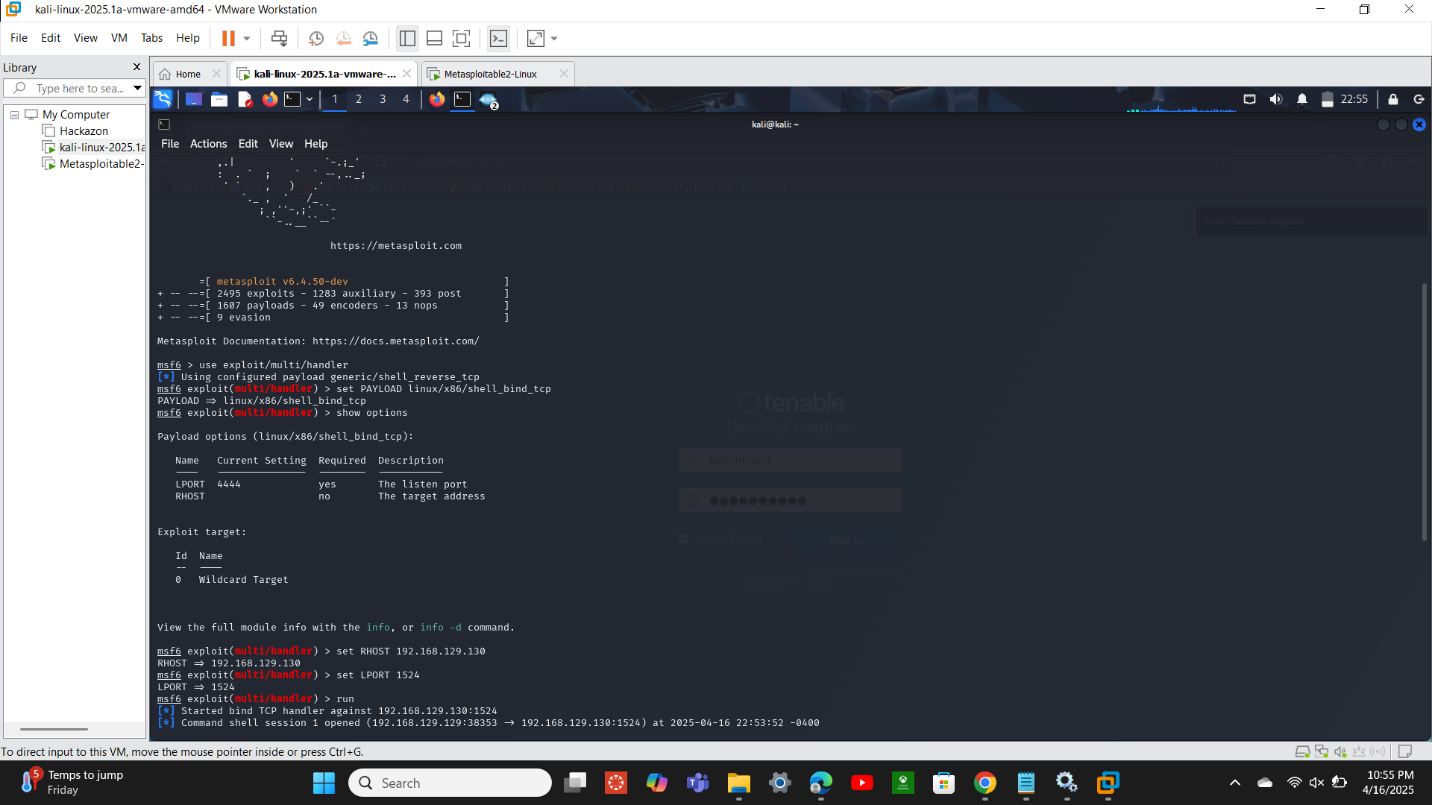


Figure 7: Configuration of and successful connection using Bind Shell backdoor exploit

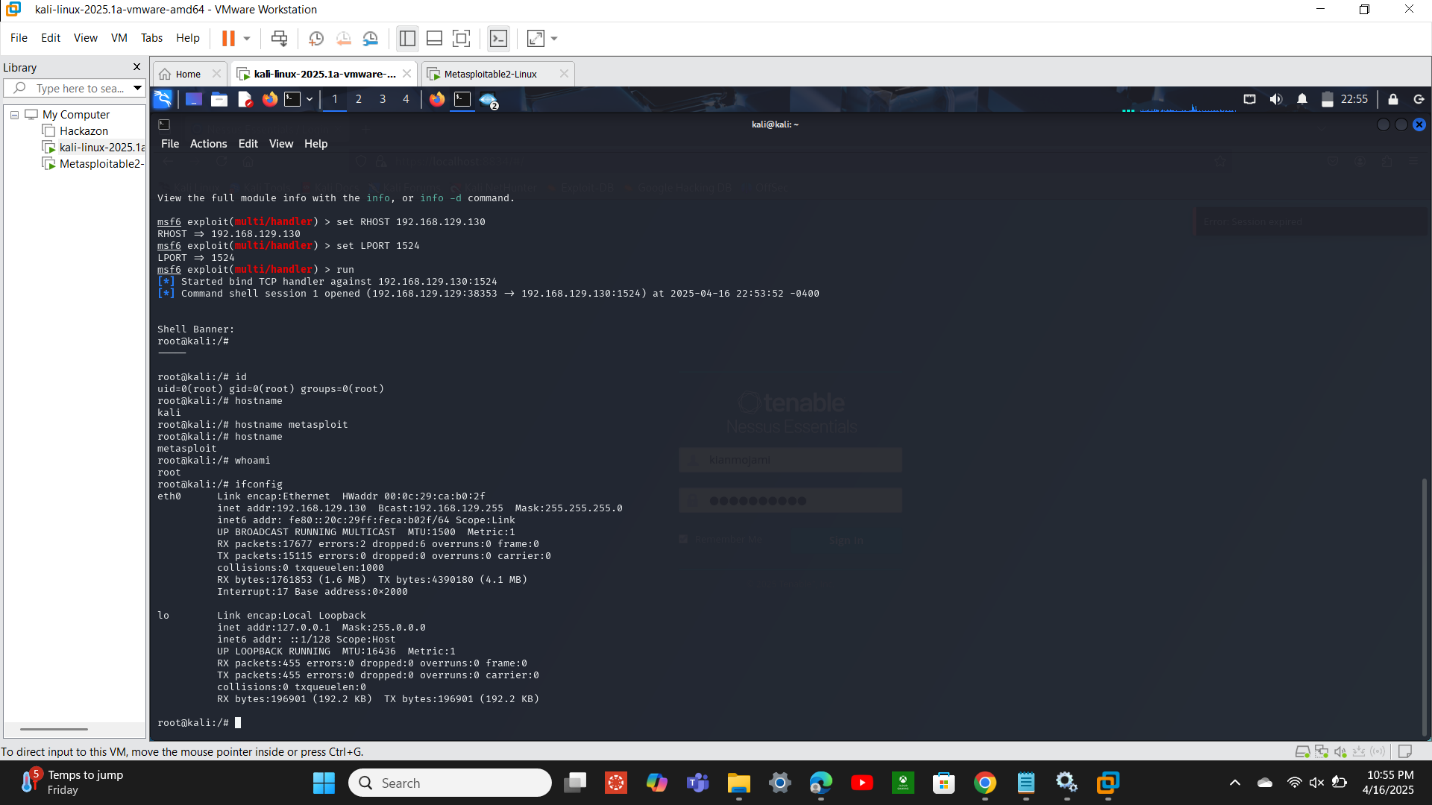


Figure 8: Root-level access using Bind Shell backdoor exploit

# Risk Assessment and Analysis

The risks these vulnerabilities present to an organization are great as they allow a user comprehensive access to the system with root abilities which allows them to find and change any information they want to. The risk becomes even higher if it were on a public-facing server compared to an internal server because anyone who knows of this vulnerability would then have the ability to exploit the vulnerabilities. If the exploit were to be successful, they would have admin-level access to any part of the system which could destroy an organization’s entire infrastructure. In a business environment, UnrealIRCd or an open remote port could be installed on a public-facing server which would allow external users the opportunity to exploit the vulnerability or installed on an internal application server which allows an insider threat to exploit the vulnerability and have access to more confidential information than they do usually.

# Mitigation and Security Control Recommendation

To fix the UnrealIRCd vulnerability, completely avoiding version 3.2.8.1 would mitigate this risk because only this version of the software is known to be vulnerable to backdoor access. Mitigating the risk of a bind shell backdoor can be done by closing any open ports that are not used or not needed to prevent them from being open listeners that an attacker can attach to and create a bind shell. Installing a firewall can also block access to unused ports which prevents attackers from having easy access to them, and IDS/IPS systems can be used to detect a breach in the system and help with eradicating them. A best practice would also be to implement network segmentation so that if an attack were to happen, the impact would be much smaller as it would be a lot harder to access further parts of the system.

**Citations**:

<https://nvd.nist.gov/vuln/detail/CVE-2010-2075>

<https://medium.com/@josegpach/detecting-and-exploiting-bind-shell-backdoor-on-metasploitable-2-f88ed3251a9b>

<https://www.infosecmatter.com/nessus-plugin-library/?id=46882>

<https://en.wikipedia.org/wiki/UnrealIRCd>

<https://medium.com/@S3Curiosity/understanding-shell-reverse-shell-and-bind-shell-a-comprehensive-guide-6bad2169edbd>