The MITRE ATT&CK T1003 OS Credential Dumping Technique and Its Adversary Use

By Sıla Özeren Hacıoğlu

Published: 2022-03-23 · Archived: 2026-04-05 18:29:08 UTC

Obtaining credentials is critical in adversaries' attack campaigns as it allows them to access other resources and systems in the target environment. Dumping credentials from operating systems and utilities is the most prevalent technique for adversaries to obtain account logins and credentials. Therefore, this technique has again secured its place as one of the top ten most frequently used techniques by adversaries in the Red Report 2024.

The Red Report 2024
The 10 Most Prevalent MITRE ATT&CK Techniques Used by Adversaries

Where are Windows OS Credentials Stored?

In a Windows operating system, credentials are stored in several places:

- **Security Account Manager (SAM) database:** The SAM is a protected system file located on the local machine, which stores the hashed versions of the password for all local user accounts on the system.
- **Local Security Authority Subsystem Service (LSASS) memory:** LSASS is a Windows process responsible for authenticating user logins and enforcing security policies. When a user logs in, the LSASS process retrieves the user's credentials from the SAM database and stores them in memory for the duration of the session.
- **NTDS.dit:** NTDS.dit is a database file on domain controllers containing all of the Active Directory data. The data in the NTDS.dit file is replicated between domain controllers in a domain or forest. If a user's account is in Active Directory, the hashed passwords are stored in the NTDS.dit file. This allows users to authenticate across all domain-joined machines.
- **Local Security Authority (LSA) Secrets:** LSA secrets is a mechanism that allows storing secrets, such as passwords, in the Windows Registry. These secrets can be used to authenticate services, schedule tasks, and other tasks that require a password.

- **Cached Domain Credentials:** When a user logs into a Windows computer that is part of a domain, the user's domain credentials are cached on the local machine so that the user can continue to access resources on the network if the domain controller is unavailable. The cached credentials are typically stored in the LSASS memory and can be used to authenticate the user even if the domain controller is not reachable.
- **Credentials Manager:** Credential Manager is the built-in Windows feature that allows users to store and manage their credentials, like passwords or certificates. These credentials will be used when a user wants to access a network resource, web page, or application that requires a username and password.
- **Group Policy:** In certain situations, credentials may be stored in Group Policy to allow automatic login for a specific user or group of users. This can be useful in cases where a user needs to access a resource that requires a username and password, but the user is not present to enter the information manually.

Where are Linux and macOS OS Credentials Stored?

In Linux and macOS operating systems, user credentials are typically stored in the following places. It's important to note that the exact locations and names of these files may vary depending on the specific Linux distribution or macOS version you are using.

- **/etc/passwd:** This file is used to store user information, including username, user ID (UID), group ID (GID), and home directory path.
- **/etc/shadow:** This file is used to store the password hashes and other information related to user authentication, such as the last time the password was changed and the date on which the account will expire. This file is only readable by the root user.
- **PAM (Pluggable Authentication Modules):** PAM is a framework that allows Linux and macOS systems to use multiple authentication methods, such as local password authentication, Kerberos, and smart cards. PAM is configured through a series of files located in the /etc/pam.d directory.
- **NSS (Name Service Switch):** This is a facility provided by the operating system that allows switching between different sources of information. For example, information about users, groups, and hosts. It is configured via the /etc/nsswitch.conf file. It can include the files /etc/passwd and /etc/shadow or an external database like LDAP, AD, or NIS.
- **Kerberos:** Kerberos is an authentication protocol that uses tickets to establish secure connections between clients and servers. Kerberos is typically used in enterprise environments and is configured through the krb5.conf file, usually located in the /etc directory.

Adversary Use of OS Credential Dumping

After gaining access and elevated privileges to a target system, adversaries harvest as many credentials as possible. Adversaries utilize the OS Credential Dumping technique to collect account login and password from the compromised system's operating system and utilities. These credentials could allow threat actors to gain access to other systems and services in the network with new privileges. Adversaries use the harvested credential information for:

- accessing restricted data and critical assets
- moving laterally to other hosts in the network
- creating new accounts and removing them to impede forensic analysis
- figuring out password patterns and policies to harvest other credentials

Sub-techniques of OS Credential Dumping

There are 8 sub-techniques under the OS Credential Dumping technique in ATT&CK v14:

ID	Name
T1003.001	LSASS Memory
T1003.002	Security Account Manager
T1003.003	NTDS
T1003.004	LSA Secrets
T1003.005	Cached Domain Credentials
T1003.006	DCSync
T1003.007	Proc Filesystem
T1003.008	/etc/passwd and /etc/shadow

Each of these sub-techniques will be explained in the next sections.

#6.1. T1003.001 LSASS Memory

Windows operating systems store the credentials of logged-in users in the Local Security Authority Subsystem Service (LSASS). LSASS allows users and services to access network resources seamlessly without re-entering their credentials. Adversaries harvest credentials by dumping LSASS memory.

LSASS verifies users logging into a Windows system, handles password changes, and creates access tokens. To authenticate users, the lsass.exe process stores and uses credentials in different forms, such as Kerberos tickets, reversibly encrypted plain text, LM hashes, and NT hashes. Users with SYSTEM privilege can interact with the lsass.exe process and dump its memory.

Adversary Use of LSASS Memory

Since LSASS memory contains valuable credentials, adversaries utilize various methods and tools to dump LSASS memory and extract credentials:

- **Mimikatz:** Mimikatz is the most common tool for credential dumping. Mimikatz can extract plaintext passwords, password hashes, PIN codes, and Kerberos tickets from memory. Adversaries also use Mimikatz to perform pass-the-hash, pass-the-ticket, and Golden tickets attacks [1].
- **gsecdump:** Gsecdump is a credential dumping tool that can harvest password hashes from LSA secrets, Active Directory (AD), Security Account Manager (SAM), and logon sessions [2].
- **ProcDump:** ProcDump is a legitimate tool that is part of the Microsoft Sysinternals suite [3]. ProcDump monitors applications for CPU spikes and generates a memory dump of processes. However, adversaries abuse ProcDump to dump LSASS memory and extract credentials from the memory dump.
- **Windows Task Manager:** Users can create memory dumps for processes using Windows Task Manager's Create Dump File feature. Adversaries with SYSTEM privilege can use this feature to dump LSASS memory.
- **Direct System Calls and API Unhooking:** Adversaries may use direct system calls to avoid security controls. By executing the system calls directly, adversaries bypass Windows and Native API and may also bypass any user-mode hooks used by security controls. For example, Dumpert can dump LSASS memory via direct system calls and API unhooking [4].

Below, you'll discover a list of APT groups and threat actors who are utilizing LSASS memory-dumping techniques.

For instance, as evident in CISA's cybersecurity advisory released in September 2023 (AA23-250A), APT actors utilized ProcDump, a tool typically employed for monitoring and creating crash dumps of processes, to execute a sophisticated cyber attack [5].

They placed ProcDump in the c:\windows\system32\prc64.exe directory for two key purposes: enumerating running processes and applications and, more crucially, dumping credentials from the LSASS. This technique demonstrates the attackers' adeptness in repurposing legitimate system tools for malicious objectives, a tactic often employed to blend in with normal network activities and avoid detection.

Additionally, in another CISA advisory released in March 2023 (AA23-075A), it was seen that LockBit 3.0 ransomware group also used Microsoft Sysinternals ProcDump to dump the contents of LSASS.exe [6].

Following this, a subsequent advisory in May 2023 (AA23-136A) highlighted the actions of the BianLian ransomware group, which similarly targeted the lsass.exe process but chose to create a memory dump and save it as a CSV file.

cmd.exe /Q /c **for** /f "tokens=1,2 delims= " ^%A **in** ('"tasklist /fi "Imagename eq lsass.exe" | find "lsass""') **do** rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump ^%B \Windows\Temp\<file>.csv full

The command uses cmd.exe to list processes matching 'lsass.exe,' then employs rundll32.exe to invoke comsvcs.dll for creating a full memory dump of LSASS into a specified file, leveraging a Windows built-in function for detailed process examination.

#6.2. T1003.002 Security Account Manager

The Security Account Manager (SAM) database stores information related to local accounts, including usernames and hashed passwords. This database resides on the local disk as a file, and adversaries use various methods to access the SAM file and extract credentials.

The SAM is used to store credentials for local accounts. It was introduced with Windows XP and is still in use for the latest versions of Windows. The SAM file is located in %systemroot%\system32\config\SAM and is mounted on the HKLM/SAM registry hive. Also, the same password hashes are stored in %systemroot%\system32\config\SYSTEM, and backup copies can be found in %systemroot%\repair directory.

The SAM database stores hashes of user passwords instead of plaintext versions. While the hash format used for password storage changed over time, the SAM database is still used by the latest versions of Windows.

- **LM:** (Legacy systems): Introduced in 1987. While turned off by default since Windows Vista/Server 2008, users can enable it afterward.
- **NTLMv1:** Introduced in 1993. It is an improved version of LM but still contains vulnerabilities.
- **NTLMv2:** This updated version of NTLMv1 includes additional security features, such as a challenge/response mechanism to provide message integrity and replay protection. It's mainly used in Windows operating systems and older versions than Windows NT3.1 and Windows 2000.
- **NTLMv2 Session Security:** This is an update of NTLMv2 that includes additional security features, like signing and sealing the messages, more robust encryption keys, and secure channel protection.
- **Kerberos:** This is an industry-standard authentication protocol used in Windows operating systems.
- **bcrypt:** A more advanced password hashing algorithm designed to replace md5crypt, Blowfish-based crypt(3) algorithm.
- **scrypt:** Another advanced password hashing algorithm, designed to be more computationally expensive than bcrypt, better suited for usage with stronger user authentication.

Since the password is stored in a one-way format (i.e., irreversible), it is not feasible to get the original password from the hashed output as long as the length and complexity of the password are not susceptible to birthday attacks.

A **one-way hash function** is a mathematical function that converts an input string of variable length into a fixed-length binary sequence. This sequence is difficult to reverse, meaning it is difficult to use the output (the hash) to determine the original input string. One-way hash functions are commonly used to securely store passwords and other sensitive data.

Adversary Use of Security Account Manager

Several tools can retrieve the SAM file through in-memory techniques, such as pwdumpx.exe, Gsecdump, Mimikatz, and secretsdump.py.

In addition to these above, adversaries can extract the SAM from the Registry via Reg:

reg save HKLM\sam sam

reg save HKLM\system system

For instance, TrickBot's ADll module takes advantage of the "Install from Media" command to dump the Active Directory database and various Registry hives to the %Temp% folder with the following command [7].

reg save HKLM\SAM %TEMP%\[generated-id]1.dat /y

These files are then compressed and sent back to the attackers.

Another example comes from CISA's cybersecurity advisory, which was released in December 2023 [8]. In this operation, the Russian Foreign Intelligence Service (SVR) had a specific target: their victims' Windows Registry. They concentrated on extracting sensitive data from the SYSTEM, SAM, and SECURITY hives. To accomplish this, they utilized the reg save command to generate copies of these hives in the C:\Windows\temp\ directory, successfully capturing vital system and user data.

```
reg save HKLM\SYSTEM ""C:\Windows\temp\1\sy.sa"" /y
reg save HKLM\SAM ""C:\Windows\temp\1\sam.sa"" /y
reg save HKLM\SECURITY ""C:\Windows\temp\1\se.sa"" /y
```

Subsequently, PowerShell was employed to compress these files into a .zip archive, staged in the same directory.

powershell Compress-Archive -Path C:\Windows\temp\1\ -DestinationPath C:\Windows\temp\s.zip -Force & del C:\Windows\temp\1 /F /Q

This methodical approach not only allowed them to systematically gather vital system information but also facilitated smooth exfiltration through their backdoor capabilities, highlighting a calculated and efficient strategy for sensitive data exfiltration.

Another credential dumping example is from the CISA's cybersecurity advisory (AA23-144A) on Volt Typhoon, which was released in May 2023 [9].

```
reg save hklm\sam ss.dat
reg save hklm\system sy.dat
```

These two commands are used to export the SAM and SYSTEM hives from the Windows Registry into respective files, ss.dat, and sy.dat.

So far, we have seen several examples of SAM database dumping through registry manipulation. However, it is essential to note that passwords within the SAM file are not stored in cleartext but in a hashed format. And even though hash functions are designed to be one-way, having an output makes it impossible to learn the input; hashing passwords does not guarantee a foolproof security measure. With a list of dumped valid account credentials, adversaries can perform offline password cracking attacks to find the cleartext password by trying many combinations of characters and comparing the resulting hashes to the stored password hash.

There are several ways to perform an offline password-cracking attack:

1. Brute-Force Attacks

In this attack type, adversaries try all possible combinations of characters up to a certain length and character set. The length and set of characters are generally defined by adversaries through gaining knowledge of the organization's password policy. It is important to note that as a password's complexity increases, brute-force attacks become significantly time-consuming and inefficient. For instance, adversaries can crack passwords with 12 characters using ChatGPT hardware in 8 months [10]. However, it would take them 3000 years to crack 14-character passwords with the same tools.

2. Dictionary Attack

In dictionary attacks, an adversary tries a predefined list of words and phrases commonly used as passwords.These attacks can be effective, but how fast they can be achieved depends on the information about a target, such as their birthday and birthplace, the name of their children or pet, which sports team they are a fan of, etc. In some cases, adversaries leverage hybrid attacks using brute force and dictionary attacks by trying a combination of commonly used words and randomly generated characters.

3. Rainbow Table Attacks

A rainbow table is a precomputed table of hash values that can be used to speed up the process of cracking passwords. A rainbow table attack works by comparing the target password hash with the hashes in the rainbow table to see if there is a match. The corresponding password can be retrieved from the table and used to log in if a match is found.

A significant benefit of using rainbow tables as an adversary is that they can avoid the hash generation process. For example, if all sets of passwords of 1-8 characters, consisting of the ASCII-32-95 characters, get hashed by the NTLM hashing algorithm, the key space would become 6,704,780,954,517,120 ≈ 2^52.6, which is approximately 460 GB.

Hence, instead of trying to generate a hash of all plaintext within a specific range, attackers can directly look up the hash in the table (some algorithms speed up the checking process) and retrieve the corresponding password.

#6.3. T1003.003 NTDS

The NTDS.dit file is a database that stores information about Active Directory Domain Services (AD DS), including user objects, groups, and group membership. It also contains the hashed passwords for all users within the domain, making it another juicy target from which adversaries can dump credentials.

Adversary Use of NTDS

Adversaries commonly leverage the following methods and tools to capture the NTDS.dit file:

1. Utilizing NTDSUtil.exe

The use of ntdsutil.exe, a built-in command-line utility located in the %systemroot%\system32\ directory on Windows systems, is a notable method for extracting sensitive data, particularly from Active Directory environments. This utility, which requires administrative privileges to operate, is capable of exporting a copy of the Active Directory database (NTDS.dit) from a Domain Controller.

It achieves this through the Install From Media (IFM) backup functionality, providing a potent tool for adversaries to access a wealth of sensitive organizational data, including user credentials and system configurations, assuming they have gained the necessary elevated access.

Threat actors often leverage the ntdsutil.exe utility to capture the NTDS.dit file.

For instance, as evident in CISA's cybersecurity advisory (AA23-319A) on Rhysida ransomware group, threat actors used the ntdsutil.exe utility to extract and dump the NTDS.dit database from the domain controller containing hashes for all AD users [11].

Once more, as revealed in a different CISA cybersecurity advisory (AA23-144A) issued in May 2023, the Volt Typhon APT employed the following commands to replicate the ntds.dit file and SYSTEM registry hive by utilizing ntdsutil.exe. Each of the subsequent actor commands stands as an individual example, with multiple instances provided to illustrate variations in syntax and file paths that may be encountered in different environments [9].

wmic **process** call create "ntdsutil \"ac i ntds\" ifm \"create full C:\Windows\Temp\pro

wmic **process** call create "cmd.exe /c ntdsutil \"ac i ntds\" ifm \"create full C:\Windows\Temp\Pro"
wmic process call create "cmd.exe /c mkdir C:\Windows\Temp\tmp & ntdsutil \"ac i ntds\" ifm \"create full C:\Windows\Temp\tmp\"

"cmd.exe" /c wmic **process** call create "cmd.exe /c mkdir C:\windows\Temp\McAfee_Logs & ntdsutil \"ac i ntds\" ifm \"create full C:\Windows\Temp\McAfee_Logs\"

cmd.exe /Q /c wmic process call create "cmd.exe /c mkdir C:\Windows\Temp\tmp & ntdsutil \"ac i ntds\" ifm \"create full C:\Windows\Temp\tmp\" 1> \\127.0.0.1\ADMIN$\<timestamp value> 2>&1

2. Leveraging Shadow Copies

Adversaries exploit shadow copies for credential dumping by targeting the ntds.dit file, the primary database of Active Directory, and the SYSTEM registry hive from Windows domain controllers. The ntds.dit file, located by default at %SystemRoot%\NTDS\ntds.dit, contains crucial information like user details, group memberships, and password hashes. The SYSTEM registry hive holds the boot key, which is essential for decrypting data in the ntds.dit file. Since the ntds.dit file is typically locked during Active Directory's operation, adversaries create a Volume Shadow Copy, a snapshot of the file system, to access a copy of this locked file.

The process typically involves using commands to create a shadow copy of the volume where the ntds.dit file resides and then copies the ntds.dit file from this shadow copy to a location where it can be exfiltrated.

For example, the following commands are run by Volt Typhoon APT in their attack campaigns that were disclosed in May 2023 [9].

cmd /c vssadmin create shadow /**for**=C: > C:\Windows\Temp\<filename>.tmp

cmd /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3\Windows\NTDS\ntds.dit C:\Windows\Temp > C:\Windows\Temp\<filename>.tmp

These commands create a shadow copy of the C: drive and then copy the ntds.dit file from the shadow copy to a temporary directory, logging the operations in a temporary file, often for surreptitious data access purposes.

#6.4. T1003.004 LSA Secrets

Local Security Authority (LSA) Secrets are sensitive information, such as credentials and secrets, that the LSA of a Windows operating system stores. The LSA is an operating system component responsible for managing security-related functions, such as authentication and authorization. LSA Secrets may include various information, such as password hashes, security keys, and other sensitive data.

LSA Secrets are stored in a protected location on the system and are typically only accessible to the operating system and trusted applications. The LSA uses them to perform various security-related tasks, such as authenticating users and granting access to resources.

LSA Secrets may be stored in a number of locations, including the system memory and the registry. On Windows systems, LSA Secrets may be stored in the HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets registry key.

Adversary Use of LSA Secrets

Adversaries leveraging Mimikatz for LSA Secrets extraction follow a structured approach. They use Mimikatz's lsadump::secrets command to target and extract sensitive data, including password hashes and security keys from system memory. This operation requires elevated privileges, typically achieved by impersonating a SYSTEM token with Mimikatz's privilege::debug command, as LSA Secrets are only accessible to the operating system and trusted applications. This process is carefully executed, taking into account potential detection mechanisms.

1. Initial Access with CrackMapExec

Adversaries begin by gaining initial access to the target system. Using CrackMapExec, they authenticate using compromised credentials.

crackmapexec smb <host address> -u "domain_admin" -p "password"

This step involves leveraging the credentials of a previously compromised domain admin.

2. Elevating Privileges and Logging Output

Upon running Mimikatz, their first command is to elevate privileges to manipulate system processes, using:

Then, they prepare to log the output to a file, anticipating extensive data that may not be fully visible in the console:

3. Extracting Logon Passwords and LSA Secrets

With elevated privileges, the adversaries execute the command to dump logon passwords and LSA secrets:

This process results in the extraction of LSA secrets, including plain text credentials.

If we were to provide some real-life examples, it is known that, in their operation, the Russian Foreign Intelligence Service (SVR) leveraged Mimikatz with **lsadump::secrets** option to dump LSA secrets from the system memory [8].

Moreover, in June 2023, it was revealed that the Chinese APT15 utilized the SharpSecDump tool to extract LSA credentials from victim systems [12]. SharpSecDump, developed in .NET, is a port of Impacket's secretsdump.py, which is part of a widely used Python toolkit for network protocols. Its main function mirrors that of secretsdump.py: to dump SAM and LSA secrets from Windows systems, targeting sensitive data extraction.

Download the Red Report - Top Ten MITRE ATT&CK Techniques

#6.5. T1003.005 Cached Domain Credentials

In situations where a domain-joined computer encounters difficulty connecting to AD DS during a user's logon process, the system caches domain credentials in the registry for authentication purposes. This local caching of logon information for domain accounts ensures that users can still access their accounts even when a connection to a domain controller is unavailable during subsequent logons.

The storage mechanism for these cached credentials is referred to as DCC2, which stands for Domain Cached Credentials version 2. DCC2 serves as a security feature within the Microsoft Windows operating system, enabling the caching of domain credentials on a system. This functionality empowers users to log in to the domain even when they are not connected to the network, enhancing the usability of Windows systems. It acts as a secondary authentication method when a connection to the domain controller cannot be established.

When DCC2 is activated on a system, domain credentials are stored either in the SAM database or the Credential Manager, depending on the Windows version in use. These cached credentials are encrypted, ensuring their security, and can only be accessed by the system when a user attempts to log in to the domain. DCC2 encompasses two types of cached domain credentials, both of which are employed by the system for authentication purposes:

1. mscache2

mscache2 is a cached domain credential used by Windows systems running Windows 2000 and later. It stores the password hash of the user's domain account, salt value, and other metadata. When a user attempts to log in to the domain, the system uses the mscache2 credentials to authenticate the user to the domain controller.

2. mcash2

mcash2 (Microsoft CAched haSH) is a newer version of the mscache2 credential, and it is used by Windows systems running Windows 8 and later. It stores the password hash of the user's domain account, additional metadata, and a more robust encryption key.

Adversary Use of Cached Domain Credentials

An adversary may use the [T1003.005] technique as part of their attack campaign to obtain cached domain credentials and use them to gain unauthorized access to the domain or other systems on the network. An adversary may use the following tools to extract the cached domain credentials from a compromised system:

- **LaZagne** can extract credentials from various sources, including the system memory, the Windows Credential Manager, and various configuration files. LaZagne can also extract cached credentials from a compromised system.
- **Cachedump**, Metasploit's post-exploitation module, extracts cached credentials from a compromised system. The cachedump module can extract cached domain credentials from the Security Accounts

Manager (SAM) database and can be used to extract mscache2 and mcash2 credentials, depending on the version of Windows.

- The **reg.exe** is not typically used to extract cached credentials, but it may be possible to extract cached credentials from the registry if they are stored there.
- **secrestdump.py** is a tool used to extract secrets, including credentials, from a system. It can also extract cached credentials.
- **Mimikatz** is also used by adversaries to extract cached credentials. It can extract credentials from various sources, including the system memory, the Security Accounts Manager (SAM) database, and the Windows Credential Manager.
- **Windows Credential Editor** (WCE) extracts credentials from the system memory or local storage, such as the SAM database. Adversaries often use it to extract cached credentials.
- Adversaries may use many **other tools** to extract cached credentials from a compromised system. Some examples include **creddump**, **Pwdump**, **Fgdump**, and **LsaDump2**.

#6.6. T1003.006 DCSync

DCSync is a feature in Microsoft Domain Controllers (DC) that allows replication of the Active Directory (AD) database from a primary to a secondary DC, ensuring all DCs have the latest directory copy. It's commonly used by adversaries with sufficient permissions to extract sensitive information like credentials.

DCSync employs the Remote Procedure Call (RPC) protocol, requiring "Replicate Directory Changes" permission on the domain object in AD. It can replicate the entire database or specific parts, either in real-time or on a schedule. Primarily used by administrators to maintain AD database integrity and availability, DCSync is integral for keeping DCs updated and often works alongside other replication technologies.

Adversary Use of DCSync

Attackers can abuse DCSync in their attack campaigns to obtain sensitive information from the AD database. An adversary can do this with sufficient permissions and credentials, using DCSync to replicate the AD database from a primary DC to a secondary DC and then extracting sensitive information such as user passwords and other credentials.

There are several ways in which attackers may abuse DCSync in their attack campaigns:

1. Obtaining User Credentials

An attacker may use DCSync to replicate the AD database and extract user credentials, such as passwords, to gain unauthorized access to the system. This can be done without leaving any trace of the operation on the primary DC, making it difficult to detect.

2. Conducting Lateral Movement

An attacker may use DCSync to obtain credentials for other systems and services on the network to move laterally within the network and potentially compromise other systems.

3. Escalating Privileges

An attacker may use DCSync to obtain credentials for high-privilege accounts, such as administrator accounts, to escalate their privileges on the system. This can allow them to perform actions that would otherwise be restricted to them.

Below, you are going to see the steps required to perform a DCSync attack with Mimikatz.

Step 1: Compromise an Account with Replication Rights

To make this attack work, adversaries first compromise an administrative account (e.g., "PrivUser1") capable of replicating data from Active Directory [13].

PS> .\mimikatz.exe "privilege::debug" "sekurlsa::msv"
PS> .\mimikatz.exe "sekurlsa::pth /user:PrivUser1 /ntlm:<hash> /domain:domain.com"

Step 2: Replicate Data from Active Directory

Using Mimikatz, they replicate credentials from Active Directory, targeting the krbtgt account:

PS> .\mimikatz.exe "lsadump::dcsync /user:DOMAIN\krbtgt"

Step 3: Execute a Golden Ticket Attack

With the krbtgt hash, they generate a **Golden Ticket** for extensive access to Active Directory:

PS> .\mimikatz.exe "kerberos::golden /domain:domain.com /sid:<SID> /krbtgt:<krbtgt_hash> /user:Administrator /id:500 /ptt"

Finally, tools like PsExec may be used for remote execution:

PS> PSExec.exe \\fileserver1 powershell.exe

This approach allows attackers to escalate privileges and achieve broad network access, underscoring the need for strong security protocols.

#6.7. T1003.007 Proc Filesystem

The proc filesystem (procfs) is a virtual filesystem in the Linux kernel that provides information about processes and other system information. It is a pseudo-filesystem, meaning it does not exist on a physical storage device but rather is generated dynamically by the kernel as needed. Adversaries may attempt to use the procfs to obtain credentials and other sensitive information from a system.

The procfs is typically mounted in the /proc directory, and it consists of a series of virtual files and directories that provide information about various aspects of the system. Some examples of the types of information that can be

found in the procfs include:

- **Process information:** The /proc directory contains a subdirectory for each running process on the system, with a numeric name corresponding to the process ID (PID). These directories contain virtual files with information about the process, such as its command line arguments, current working directory, and open file descriptors.
- **Kernel information**: The procfs also contains virtual files and directories with information about the kernel and its configuration. This can include information about the version of the kernel, the system architecture, and the loaded kernel modules.
- **Hardware information:** The procfs contains virtual files with information about the hardware on the system, such as the processor type and model, the amount of memory installed, and the configured interrupts and I/O ports.

Adversary Use of Proc Filesystem

The proc filesystem (procfs) can potentially be used by attackers to obtain credentials and other sensitive information about the operating system and its processes. There are several ways in which attackers may use the procfs for this purpose:

1. Extracting Command-line Arguments

The procfs contains virtual files with the command-line arguments of each running process on the system. An attacker may attempt to read these files in order to obtain any sensitive information that may have been passed as command-line arguments, such as passwords or API keys.

2. Reading Environment Variables

The procfs contains virtual files with the environment variables of each running process. An attacker may attempt to read these files in order to obtain sensitive information that may be stored in environment variables, such as credentials for external services or database servers.

3. Obtaining Process Information

The procfs contains virtual files with information about the processes running on the system, including the current working directory, open file descriptors, and other details. An attacker may use this information to gather intelligence about the system and potentially identify processes that may be of interest.

4. Reading Kernel Information

The procfs contains virtual files and directories with information about the kernel and its configuration. An attacker may attempt to read these files in order to obtain information about the version of the kernel, the system architecture, and the loaded kernel modules. This information may be used to tailor an attack to the specific system and potentially exploit known vulnerabilities.

An adversary may use the following tools to extract credentials using the proc file system:

- **MimiPenguin** is an open-source tool capable of dumping process memory and harvesting passwords and hashes by searching for text strings and regular expressions.
- **LaZagne** can extract credential information from process memory with the memorydump.py module. It includes regex patterns for passwords of common websites, such as Gmail, Dropbox, Salesforce, PayPal, Twitter, Github, and Slack. Lazagne uses these patterns to dump cleartext passwords from the browser's memory. Its mimipy.py module is a Python port of MimiPenguin.
- **Procdump** for Linux is a Linux reworking of the classic ProcDump tool from the Sysinternals suite of tools for Windows. It provides Linux developers with a straightforward method for generating core dumps of their applications in response to performance triggers. Naturally, adversaries utilize this tool to dump process memory and extract credentials from dumped memory.

#6.8. T1003.008 /etc/passwd and /etc/shadow

The /etc/passwd and /etc/shadow files store information about user accounts on a Unix-like system. While the /etc/passwd file stores user account information, the /etc/shadow file consists of password hashes. MD5, SHA-256, and SHA-512 are hash algorithms used for these passwords. The contents of these files are dumped by adversaries for offline password cracking.

The /etc/passwd file stores information about each user account, including the username, user ID (UID), group ID (GID), and home directory. It does not contain the user's password, however. The /etc/shadow file stores the hashed password for each user account, along with other information, such as the password expiration date and any password reset flags. This file is typically readable only by the root user, as it contains sensitive information.

Adversary Use of /etc/passwd and /etc/shadow

Adversaries may attempt to access or modify the contents of the /etc/passwd and /etc/shadow files on a Unix-like system in order to compromise user accounts and gain unauthorized access to the system. There are several ways in which attackers may use these files for malicious purposes:

1. Adding new user accounts

An attacker may add a new user account to the /etc/passwd file with a known or easily guessable password. This would allow them to log in to the system using the new account and potentially escalate their privileges.

2. Modifying existing accounts

For example, an attacker may modify an existing user account in the /etc/passwd file by changing the home directory or group membership to escalate their privileges. They may also modify the user's password in the /etc/shadow file by changing it to a known password or setting it never to expire.

3. Gaining access to encrypted passwords

An attacker may attempt to gain access to the /etc/shadow file in order to obtain the encrypted passwords for offline cracking. They may then use tools such as John the Ripper or Hashcat to try to crack the passwords and gain access to user accounts.

4. Using these files as part of a larger attack

An attacker may use the /etc/passwd and /etc/shadow files in combination with other tactics and techniques as part of a larger attack against a system, such as moving laterally within the network.

Tools Used by Adversaries to Dump Credentials from /etc/passwd and /etc/shadow Files

- **Chntpw**

This versatile utility, originally designed for resetting passwords on Windows systems, can also be repurposed to dump password hashes from Unix/Linux systems. When used in a Unix-like environment, Chntpw can access and read the contents of /etc/passwd and /etc/shadow. The command syntax required for this tool is as follows.

chntpw -E /etc/passwd > passwd_hashes.txt
chntpw -S /etc/shadow >> passwd_hashes.txt

- **Unshadow**

The Unshadow tool is a specialized utility in Linux environments designed to merge the contents of the /etc/passwd and /etc/shadow files. Combining these two files, Unshadow creates a single file with usernames and associated hashed passwords. The command required for this tool is as follows [14].

unshadow /etc/passwd /etc/shadow > password_file

- **LaZagne**

LaZagne stands out as a versatile credential extraction tool, capable of retrieving sensitive information from various systems, including Unix-like systems. Specifically, on Linux, its shadow.py module, found under the /Linux/lazagne/softwares/sysadmin directory, is adept at accessing credential data from the /etc/shadow file. This file is critical as it contains hashed passwords of system users. The command used in this technique is as follows [15].

LaZagne's capability extends to performing dictionary attacks on several hash formats stored in /etc/shadow, including MD5, Blowfish, SHA-256, and SHA-512. This functionality allows it to potentially crack and reveal user passwords, assuming it operates with the necessary root privileges.

References

[1] “GitHub - ParrotSec/mimikatz,” GitHub. https://github.com/ParrotSec/mimikatz

[2]“gsecdump.” https://jpcertcc.github.io/ToolAnalysisResultSheet/details/gsecdump.htm

[3]“procdump.” https://jpceahttps://learn.microsoft.com/en-us/sysinternals/downloads/procdump

[4] “GitHub - outflanknl/Dumpert: LSASS memory dumper using direct system calls and API unhooking,” GitHub. https://github.com/outflanknl/Dumpert

[5] “Multiple Nation-State Threat Actors Exploit CVE-2022-47966 and CVE-2022-42475,” Cybersecurity and Infrastructure Security Agency CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-250a

[6] “StopRansomware: LockBit 3.0,” Cybersecurity and Infrastructure Security Agency CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-075a

[7] L. Abrams, “TrickBot Now Steals Windows Active Directory Credentials,” BleepingComputer, Jan. 23, 2020. https://www.bleepingcomputer.com/news/security/trickbot-now-steals-windows-active-directory-credentials/

[8] “Russian Foreign Intelligence Service (SVR) Exploiting JetBrains TeamCity CVE Globally,” Cybersecurity and Infrastructure Security Agency CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-347a

[9]“People’s Republic of China State-Sponsored Cyber Actor Living off the Land to Evade Detection,” Cybersecurity and Infrastructure Security Agency CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-144a

[10] “Hive Systems Password Table,” Hive Systems. https://www.hivesystems.io/password-table

[11] “StopRansomware: Rhysida Ransomware,” Cybersecurity and Infrastructure Security Agency CISA. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-319a

[12] B. Toulas, “Chinese APT15 hackers resurface with new Graphican malware,” BleepingComputer, Jun. 21, 2023. https://www.bleepingcomputer.com/news/security/chinese-apt15-hackers-resurface-with-new-graphican-malware/

[13]S. Özeren, “DCShadow Attack Explained - MITRE ATT&CK T1207,” Aug. 22, 2023. https://www.picussecurity.com/resource/blog/dcshadow-attack-explained-mitre-attack-t120

[14]“Unshadow Command Examples in Linux.” https://www.thegeekdiary.com/unshadow-command-examples-in-linux/

[15]H. C. Yuceel, “The MITRE ATT&CK T1003 OS Credential Dumping Technique and Its Adversary Use,” Mar. 23, 2022. https://www.picussecurity.com/resource/the-mitre-attck-t1003-os-credential-dumping-technique-and-its-adversary-use

---