LOGPOINT

EMERGING THREATS PROTECTION REPORT

Defending Against OS Credential Dumping: Threat Landscape, Strategies, and Best Practices

www.logpoint.com

FOREWORD

Cybersecurity is a top priority in today's linked world, where digital change accelerates at unprecedented speeds. Threat actors are unrelenting in pursuing unauthorized access to vital systems and sensitive information. OS Credential Dumping is a persistent and widespread danger among attackers who use various approaches.

This research from Logpoint's Emerging Threat Protection team digs into the complexities of OS Credential Dumping (T1003), detailing its multiple sub-techniques in the Windows environment. By collecting usernames and passwords from compromised systems, attackers get a footing to elevate privileges, travel laterally across networks, and carry out more complex cyber operations.

Understanding the complexities of OS Credential Dumping is critical for enterprises looking to improve their cybersecurity posture. It is more than just a technical matter; it is essential to ensure business continuity, data integrity, and consumer confidence.

As threats develop, so do our defenses. This paper explains the threat environment and offers practical insights and solutions to strengthen defenses against OS Credential Dumping. Organizations may reduce the dangers of this widespread threat by using attentive monitoring, robust authentication processes, and proactive vulnerability management.

At Logpoint, we are committed to providing companies with the information and resources to battle new cyber threats effectively. Our specialized security researchers are ready to assist you with your cybersecurity journey, providing bespoke detection rules and personalized playbooks designed to identify, analyze, and respond to OS Credential Dumping situations quickly and decisively.

I welcome you to read this study, acquire insights, and evaluate how you may strengthen your organization's resistance against OS Credential Dumping and other growing cybersecurity risks. Let's handle the complexity of the digital environment with caution, teamwork, and invention.

TABLE OF CONTENTS

Swachchhanda Shrawan Poudel

Logpoint Security Research

Swachchhanda Shrawan Poudel is a cybersecurity professional specializing in purple teaming, reverse engineering, and malware analysis. Currently a Security Researcher at Logpoint Security Research, he leads the Emerging Threat Protection initiative. His focus includes detection engineering, threat hunting, and remediation, with a special passion for crafting effective detection rules, threat reports and playbooks.

Ujwal Thapa

Logpoint Security Research

Ujwal Thapa is a cybersecurity enthusiast who has been working as a Security Researcher at Logpoint since 2021. His expertise includes threat hunting, response, detection engineering, and cloud security. Ujwal holds several notable certifications such as SAA-CO3, SC200, AZ104, and CEH (practical).

ABOUT LOGPOINT EMERGING THREATS PROTECTION

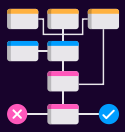
The cybersecurity threat landscape continuously changes while new risks and threats are constantly discovered. Only some organizations have enough resources or the know-how to deal with evolving threats.

Emerging Threats Protection is a managed service provided by a Logpoint team of highly skilled security researchers who are experts in threat intelligence and incident response. Our team informs you of the latest threats and provides custom detection rules and tailor-made playbooks to help you investigate and mitigate emerging incidents.

All new detection rules are available as part of Logpoint's latest release and through the **Logpoint Help Center. Customized investigation and response playbooks are available to all Logpoint Emerging Threats Protection customers.

Below is a rundown of the incident, potential threats, and how to detect any potential attacks and proactively defend using Logpoint Converged SIEM capabilities.

1. Research for emerging threats such as malware families, threat actors and vulnerabilities
2. Data retrieval e.g., malware samples, IOCs, and TTP



1. Analysis of the collected data and malware and, tracking of threat actors' activities
2. Creation and update analytics and playbooks
3. Writing of ETP report

1. Publishing of report

1. Continuous monitoring for other emerging threats to create next ETP report

BACKGROUND

In the current technical landscape, where security is paramount, credential dumping has become a significant concern because of the increase in complex cyber threats. Credentials, such as passwords, API keys, and tokens, are essential for accessing devices, systems, and networks. Threat actors continuously seek ways to obtain these credentials to gain unauthorized access to valuable resources. Over time, techniques for credential dumping have evolved, ranging from simple password guessing to more advanced methods like phishing, token manipulations, exploiting vulnerabilities, and many more.

Credential access attempts are typical in threat investigations related to ransomware, Advanced Persistent Threat (APT) groups, malware, and other cyber threats. Threat actors seek to obtain valid credentials to move laterally within networks, escalate privileges, and access sensitive data. This focus on **Credential Access** is so significant that **MITRE ATT&CK®** has a dedicated column among its 14 tactics table.

OS CREDENTIAL DUMPING (T1003): A COMMON AND CRITICAL THREAT

Among the techniques under Credential Access, **OS Credential Dumping (T1003)** is the most common and prevalent technique among threat actors. OS Credential Dumping allows adversaries to steal usernames and passwords from compromised systems. This technique is a significant initial step for attackers who have gained access to a system, empowering them to move laterally within a network and escalate privileges. By leveraging various methods, adversaries aim to extract credentials stored within the victim's operating system or software. This stolen information grants them access to valuable resources, often with higher privileges.

According to MITRE ATT&CK v15.1, it has eight constituent sub-techniques. However, this report only focuses on the techniques related to the Windows OS ecosystem.

T1003.001
LSASS Memory

T1003.002
Security Account Manager

T1003.003
NTDS

T1003.004
LSA Secrets

T1003.005
Cached Domain Credentials

T1003.006
DCSync

T1003.001: LSASS MEMORY

LSASS, or the Local Security Authority Subsystem Service, is an essential Windows operating system component. It is responsible for enforcing the system's security policies. Here are some crucial data about lsass.exe and why it is targeted by threat actors (TAs):

What is lsass.exe?

1. **Function:** lsass.exe enforces security regulations and manages user authentication, including verifying logins and password modifications. It also oversees the generation of access tokens, which enable users and services to access system resources.
2. **Service Role:** LSASS, as a system process, is an essential component of Windows' security architecture. It communicates with the Security Accounts Manager (SAM) database, Active Directory, and many authentication services.
3. **Process:** The process operates in the background with elevated privileges, granting it extensive access to system resources and user credentials.

Details Held by lsass.exe:

User Credentials: After logging in, lsass.exe stores the user's credentials in memory. This encompasses hashed passwords, Kerberos tickets, and NTLM hashes.

Authentication Tokens: LSASS contains access tokens with user rights and groups required for user and service authentication.

Session Data: LSASS manages information about active sessions, including Terminal Services sessions.

Why do threat actors target lsass.exe?

Credential Harvesting: Because lsass.exe stores user credentials in memory, it is a popular target for attackers looking to harvest these credentials. Accessing this data enables attackers to gain unauthorized access to accounts without cracking passwords.

Lateral Movement: After obtaining credentials from lsass.exe, attackers can use these credentials to move laterally across the network and gain access to more systems and resources. This is especially useful when the hijacked account has administrative access.

Privilege Escalation: Attackers can use the credentials held in lsass.exe to elevate their rights within the network, from lower-privileged users to domain administrators.

Persistence: By getting credentials, attackers can continue accessing the network even if their initial entry point is found and closed.

Common Tools and Techniques for Credential Dumping from LSASS

The LSASS is a precious target, so several tools and approaches exist for extracting credentials from its memory. These include LOLBAS (Living Off the Land Binaries and Scripts) like rundll32, comsvcs.dll, and taskmgr.exe, as well as bespoke tools like Mimikatz, Cobalt Strike, Nanodump, and Dumpert. Other essential SysInternals utilities include Procdump, Process Explorer, and Microsoft SqlDumper. Adversaries utilize these tools and tactics to dump the LSASS memory and obtain access to sensitive credential data.

In-Memory Techniques:

1. **Procdump:** A legitimate Windows Sysinternals tool used to dump the memory of the LSASS process.
   - Command: `procdump -ma lsass.exe lsass_dump`
2. **Mimikatz:** A very common credential-dumping tool.
   - Command
   - `sekurlsa::Minidump lsassdump.dmp`
   - `sekurlsa::logonPasswords`
3. **comsvcs.dll:** `comsvcs.dll` is a dynamic link library (DLL) file associated with the Component Services (COM+) infrastructure in Microsoft Windows. It exports a MiniDump function, which in turn calls MiniDumpWriteDump. `MiniDumpWriteDump` is a function provided by the Windows API that allows developers to create a minidump file of a specified process. Adversaries often use this technique to dump the memory space of lsass.exe
   - Command: `rundll32.exe C:\Windows\System32\comsvcs.dll MiniDump PID lsass.dmp full`
4. **Silent Process Exit:** Werfault.exe is the Windows Error Reporting process in Windows 10, utilized by many applications to report errors. For example, Werfault.exe forwards the crash report to Microsoft if an application crashes on your computer. These files are often described as Windows Fault Reporting components. Similarly, wer.dll is the Windows Error Reporting DLL. Abusing the Windows Error Reporting (WerFault.exe) mechanism to create a memory dump of LSASS. To effectively monitor and manage Windows Error Reporting, it is essential to keep track of the processes WerFault.exe and WerFaultSecure.exe, along with the DLL file “wer.dll.” This ensures a comprehensive understanding of error-reporting activities and aids in diagnosing potential issues.

Attackers Use This Technique

Adversaries across various threat groups utilize tools like Mimikatz to capture credentials, leveraging the sensitive information stored in the Local Security Authority Subsystem Service (LSASS) memory. This tactic is seen in numerous high-profile attacks and by various Advanced Persistent Threat (APT) groups:

- **Sandworm Team:** Mimikatz was used to get and use legitimate credentials during the 2016 Ukraine Electric Power Attack.
- **APT1:** Known to use Mimikatz for credential dumping.
- **APT28 (Fancy Bear):** Regularly deploys tools like Mimikatz and custom password retrieval tools, including dumping LSASS process memory using the MiniDump function.
- **APT3:** Uses a tool that injects itself into lsass.exe and dumps credentials with the argument "dig."
- **APT32 (OceanLotus):** Utilizes Mimikatz and customized versions of Windows Credential Dumper to harvest credentials.
- **APT33:** Uses publicly available tools like LaZagne, Mimikatz, and ProcDump to dump credentials.
- **APT39:** Uses Mimikatz, Windows Credential Editor, and ProcDump to dump credentials.
- **APT41:** Utilizes hashdump, Mimikatz, and the Windows Credential Editor to dump password hashes from memory.
- **APT5:** Targets LSASS process memory via Task Manager to obtain NTLM password hashes and uses Mimikatz hosted through an RDP mapped drive to dump clear text passwords and hashes.
- **Aquatic Panda:** Attempts to harvest credentials through LSASS memory dumping.
- **Bad Rabbit:** Uses Mimikatz to harvest credentials from the victim's machine.
- **Blue Mockingbird:** Uses Mimikatz to retrieve credentials from LSASS memory.
- **BRONZE BUTLER:** Utilizes tools like Mimikatz and Windows Credential Editor (WCE) for credential dumping.
- **C0032 (TEMP.Veles):** Uses Mimikatz and a custom SecHack tool to harvest credentials.
- **Cleaver:** Known to use Mimikatz and Windows Credential Editor to dump credentials.
- **Cobalt Strike:** Can start a task that dumps password hashes into LSASS memory.
- **CozyCar:** Executes Mimikatz to harvest stored credentials from the victim.
- **Cutting Edge:** Uses Task Manager to dump LSASS memory from Windows devices to disk.
- **Daserf:** Utilizes Windows Credential Editor and Mimikatz to steal login credentials.
- **Earth Lusca:** Uses ProcDump to obtain credential hashes by dumping LSASS memory.
- **Emotet:** Drops password grabber modules, including Mimikatz.
- **Empire:** Has a Mimikatz implementation that pulls login credentials from memory.
- **FIN13:** Obtains memory dumps with ProcDump to parse and extract credentials using Mimikatz.
- **FIN6:** Uses Windows Credential Editor for credential dumping.
- **FIN8:** Harvests credentials using Invoke-Mimikatz or Windows Credential Editor (WCE).
- **Fox Kitten:** Uses ProcDump to dump credentials from LSASS.
- **GALLIUM:** Uses a modified version of Mimikatz and a PowerShell-based Mimikatz to dump credentials.
- **GreyEnergy:** includes a module for Mimikatz that gathers Windows login information.
- **HAFNIUM:** Uses ProcDump to dump the LSASS process memory.
- **Impacket:** Modules like SecretsDump and Mimikatz perform credential dumping to obtain account and password information.
- **Indrik Spider:** Uses Cobalt Strike to carry out credential dumping using ProcDump.
- **Ke3chang:** Uses Mimikatz for credential dumping.
- **Kimsuky:** Gathers credentials using Mimikatz and ProcDump.
- **LaZagne:** Can dump credentials from memory to obtain account and password information.

- **Leafminer:** Uses tools like LaZagne and Mimikatz to retrieve login and password information.
- **Leviathan:** Uses publicly available tools like ProcDump and WCE for password dumping.
- **Lizar:** Can run Mimikatz to harvest credentials.
- **Lslsass:** Hashes of active login session passwords can be extracted from the LSASS process.
- **Mafalda:** Password hashes from LSASS.exe can be dumped.
- **Magic Hound:** Steals domain credentials by dumping LSASS memory using Task Manager, comsvcs.dll, and Mimikatz from a domain controller.
- **Mimikatz:** Carries out credential dumping to get password and account information from LSASS memory.
- **MuddyWater:** Uses Mimikatz and procdump64.exe for credential dumping.
- **Net Crawler:** Uses tools like Mimikatz and Windows Credential Editor to extract cached credentials.
- **NotPetya:** Consists of a modified Mimikatz variant designed to obtain credentials for sidestepping security measures.
- **OilRig:** Uses Mimikatz to steal credentials from the compromised system and Outlook Web Access.
- **Okrum:** Uses MimikatzLite for credential dumping.
- **Olympic Destroyer:** Contains a module similar to Mimikatz to obtain credentials from LSASS for propagation.
- **Operation Wocao:** Uses ProcDump to dump credentials from memory.
- **PLATINUM:** Uses keyloggers capable of dumping credentials.
- **PoetRAT:** Uses voStro.exe, a compiled version of pypykatz (Python Mimikatz), to steal credentials.
- **PoshC2:** Contains an implementation of Mimikatz for credential harvesting.
- **PowerSploit:** Contains modules that can harvest credentials using Mimikatz.
- **Pupy:** Can execute LaZagne and Mimikatz via PowerShell.
- **Pysa:** Uses Mimikatz for OS credential dumping.
- **Sandworm Team:** Uses tools to dump Windows credentials, such as comsvcs.dll, a modified version of Mimikatz, and plainpwd.
- **Silence:** Uses the Farse6.1 utility (based on Mimikatz) to extract credentials from lsass.exe.
- **SILENTTRINITY:** The MiniDumpWriteDump Win32 API call can generate an LSASS memory dump.
- **Threat Group-3390:** Uses Wrapikatz, a modified version of Mimikatz, and dumps domain controller credentials.
- **Triton Safety Instrumented System Attack (TEMP.Veles):** Uses Mimikatz for credential dumping.
- **Volt Typhoon:** Attempts to access hashed credentials from the LSASS process memory space.
- **Whitefly:** Uses Mimikatz to obtain credentials.
- **Windows Credential Editor:** Can dump credentials.
- **Wizard Spider:** Uses LaZagne to dump the lsass.exe memory for credential harvesting.

Detection for OS Credential Dumping: LSASS Memory

Enabling detailed logging and auditing on the system is crucial for detecting such activities. Below are the steps to configure and enable the necessary logging.

Enabling Audit File System

1. **Access Local Security Policy Settings:**
   - Navigate to `Security Settings > Advanced Audit Policy Configuration > Audit Policies > Object Access > Audit File System`.
   - Enable auditing for both success and failure events.

> **Please be aware that enabling this auditing can generate significant log data, potentially increasing noise in the audit logs.**

Configuring Audit Policies for LSASS

1. **Default Process SACL for LSASS.exe:**
   - In **Windows 10 Enterprise LTSC 2015**, a default process SACL was added to `LSASS.exe` to log processes attempting to access `LSASS.exe`. The SACL is `"S:(AU;SAFA;0x0010;;;WD)"`. This helps identify attacks that steal credentials from the memory of a process.
   - Logging can be enabled from `Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies > Object Access > Audit Kernel Object`.

Enabling Process Creation Auditing

1. **Access Local Security Policy Settings:**
   - Navigate to `Security Settings > Advanced Audit Policy Configuration > System Audit Policies - Local Group > Detailed Tracking > Audit Process Creation`.
   - Enable auditing for both success and failure events.
   - To include command line auditing, ensure the `Audit Command Line` is enabled. This can be configured via:
     - `Computer Configuration > Administrative Templates > System > Audit Process Creation > Include command line in process creation events`.

Enabling Registry Auditing

1. **Access Local Security Policy Settings:**
   - Navigate to `Security Settings > Local Policies > Audit Policy > Audit Object Access`.
   - Enable auditing for both successful and failed events.

> **Please be aware that enabling this auditing can generate significant log data, potentially increasing noise in the audit logs.**

2. **Configure Specific Registry Key Auditing:**
   - Open Registry Editor.
   - Browse to the registry key to be audited, e.g., `HKEY_LOCAL_MACHINE\SAM`.
   - Right-click the key, select `Permissions`, then `Advanced`, and go to the `Auditing tab`.
   - Add a new audit entry with the Query Value, Enumerate Subkeys, and Read Control permissions.
   - Set the Principal for `Everyone` to ensure all users are audited.
   - Repeat the above steps for the `SYSTEM` and `SECURITY` registry hives.

Enabling Windows Sysmon

1. **Windows Sysmon Configuration:**
   - Ensure Windows Sysmon is installed.

Enabling Windows Powershell Logging

- Enable `event_id 4104` and `event_id 4103` logging for script block and module logging.

Implementing Detection Strategies

Once the required logging and auditing are enabled, it's essential to implement detection strategies to identify anomalies indicative of credential dumping activities. Below is an approach to effectively detect such anomalies according to MITRE:

1. **Baseline Normal Behavior:**
   - Understand what constitutes regular access and activity on your systems to distinguish between legitimate and suspicious actions.
2. **Command Execution**:
   - Monitor commands and arguments being executed to identify potential attempts to obtain credentials kept in the Local Security Authority Subsystem Service (LSASS) process memory. Remote access tools can integrate preexisting tools such as Mimikatz or have built-in capabilities. Additional programs in PowerShell, such as the Invoke-Mimikatz module in PowerSploit, can dump credentials.
3. **File Creation**:
   - Monitor for the unexpected creation of memory dump files associated with the LSASS process, such as files named lsass{*}.dmp.

```
label=File label=Create
file IN ["lsass*.dmp", "lsass.rar", "lsass.zip", "lsassdump", "lsassdmp","lsass*"]
```

4. **Process Creation:**
   - Monitor for newly executed processes that might signal credential dumping.
5. **Detect Abnormal LSASS Access and Injection:**
   - Detect the reading of LSASS memory using Sysmon events by monitoring specific access permissions (0x1010, 0x1410, etc.) on the lsass.exe process.

```
norm_id=WindowsSysmon event_id=10 "image"="C:\windows\system32\lsass.exe"
access IN ["0x1fffff", "0x01000", "0x1010", "0x1038", "0x1438", "0x143a", "0x40", "0x1400",
"0x1410"]
call_trace IN ["*dbghelp.dll*", "*dbgcore.dll*", "*ntdll.dll*"]
-"process"="*\Sysmon64.exe"
```

6. **Monitor LSASS child processes like `cmd.exe`, `powershell.exe`, `regsvr32.exe`, `mstsc.exe`, `dllhost.exe`, or unsigned binaries.**

```
label="Process" label=Create
"parent_process"="*\lsass.exe"
"process" IN ["*\cmd.exe", "*\powershell.exe", "*\regsvr32.exe",
"*\mstsc.exe", "*\dllhost.exe"]
```

```
norm_id=WindowsSysmon event_id=7 image="*\lsass.exe"
(-is_signed="true" OR status IN ["errorChaining",
"errorCode_endpoint*", "errorExpired", "trusted"])
```

Detection Queries for LSASS Credential Dumping Attempts

Here is a list of detection queries crafted after considering the detection strategy. They can be used for alerting on LSASS credential dumping attempts:

Process Dump via Rundll32 and Comsvcs
Transfering Files with Credential Data via Network Shares
Unsigned Image Loaded Into LSASS Process
CreateMiniDump Hacktool Detected
Credential Dump Tools Dropped Files Detected
LSASS Access from Non System Account Detected
LSASS Memory Dump Detected
Lsass Memory Dump with MiniDumpWriteDump API Detected
Mimikatz Command Line Detected
Password Dumper Remote Thread in LSASS
Credential Access via LaZagne
Usage of Procdump Detected
HandleKatz Duplicating LSASS Handle
Process Dump via Resource Leak Diagnostic Tool
Suspicious Execution of Dump64
Process Dump via Sqldumper Detected
Suspicious Execution of XORDump Utility for LSASS Memory Dump
Dumpert Process Dumper Execution
Possible LSASS Memory Dump Via Windows Task Manager
Suspicious Execution of Createdump Utility for Memory Dump
Possible LSASS Dump Via SilentProcessExit Technique

Here are Microsoft's recommendations for hardening the LSASS process to prevent or at least try to add an extra layer of defense depth.

- Enable **Protected Process Light (PPL)** for the LSASS process. This is already enabled by default for new, enterprise-joined Windows 11 installations (22H2 update).
- Enable Windows **Defender Credential Guard**, enabled by default on the enterprise edition of Windows 11.
- Enable Remote Desktop Protocol (RDP) restricted admin mode.
- Turn off WDigest's "UseLogonCredential" setting.

Following these configurations and monitoring strategies can enhance the detection and mitigation of LSASS dumping activities on your systems.

T1003.002: SECURITY ACCOUNT MANAGER

Another technique threat actors utilize for the credential dump is the Security Account Manager (SAM). SAM is a database file in Windows that stores the user credentials of the local user defined on the system. While the SAM database is technically stored in a file located at "%systemroot%\System32\config\SAM", it's accessible through a corresponding registry hive at "HKEY_LOCAL_MACHINE\SAM." The file located at %systemroot%\repair\sam._ functions as a safeguard, serving as a backup for the main SAM file. This backup becomes crucial should system recovery be necessary during a repair procedure. However, regardless of the access method, reading or extracting data from the SAM database requires the highest level of system access - SYSTEM privileges.

Attackers utilizing this technique

Various APT groups use different tools and methods to extract this information. Here are some notable examples:

- **APT29:** Utilized the reg save command to save registry hives.
- **APT41:** Extracted user account data from the SAM by copying the database from the registry using the reg save command or exploiting volume shadow copies. During campaign C0017, APT41 copied the SAM and SYSTEM Registry hives for credential harvesting.
- **APT5:** Copied and exfiltrated the SAM Registry hive from targeted systems.
- **Cobalt Strike:** Capable of recovering hashed passwords from the SAM database.
- **CosmicDuke:** Collected Windows account hashes from the SAM database.
- **CozyCar:** Used password and NTLM stealer modules to harvest stored credentials, including those used in Windows NTLM user authentication.
- **CrackMapExec:** Can dump usernames and hashed passwords from the SAM database.
- **Dragonfly:** Utilized SecretsDump to dump password hashes.
- **Fgdump:** Can dump Windows password hashes from the SAM database.
- **FIN13:** Extracted the SAM and SYSTEM registry hives using the reg.exe binary to obtain password hashes from compromised machines.
- **GALLIUM:** Used `reg` commands to dump specific hives from the Windows Registry, such as the SAM hive, to obtain password hashes.
- **gsecdump:** Dumps Windows password hashes from the SAM database.
- **HOPLIGHT:** Capable of harvesting credentials and passwords from the SAM database.
- **IceApple:** Its Credential Dumper module can dump encrypted password hashes from SAM registry keys, including `HKLM\SAM\SAM\Domains\Account\F and HKLM\SAM\SAM\Domains\Account\Users\*\V`.
- **Impacket:** Modules like SecretsDump and Mimikatz can dump credentials to extract account and password information from the SAM database.
- **Ke3chang:** Used gsecdump to dump credentials from the SAM database.
- **Koadic:** Can gather hashed passwords by dumping the SAM/SECURITY hive.
- **menuPass:** A modified version of pen-testing tools like wmiexec.vbs and secretsdump.py was used to dump credentials from the SAM database.
- **Mimikatz:** Dumps credentials from the SAM database table to extract account and password details.
- **Mivast:** Capable of gathering NTLM password information from the SAM database.
- **Night Dragon:** Dumped account hashes using gsecdump.
- **Operation CuckooBees:** Leveraged a custom tool to dump OS credentials using commands like `reg save HKLM\SYSTEM system.hiv`, `reg save HKLM\SAM sam.hiv`, and `reg save HKLM\SECURITY security.hiv`.
- **POWERTON:** Can dump password hashes from the SAM database.
- **pwdump:** This can dump credentials from the SAM database.
- **Remsec:** Can dump the SAM database to retrieve credentials.
- **Threat Group-3390:** Used gsecdump to dump credentials from the SAM database and is also known to dump credentials from domain controllers.
- **Wizard Spider:** Acquired credentials from the SAM/SECURITY registry hives.

These groups utilize publicly available tools and custom-developed methods to extract credentials from the SAM database, enabling them to gain unauthorized access to systems and propagate further within networks.

Detection

You are fine if you enable the above audit logging in LSASS Dumping Detection.

Monitoring commands and arguments that might target the SAM for credential extraction is crucial. Threat actors widely use tools like mimikatz and reg, aka regedit (Registry Editor), for credential dumping. Watch on program opening `%SystemRoot%\system32\config\SAM` or any access and modifications on SAM file `%SystemRoot%\system32\config\SAM`.

Detection Queries for Security Account Manager

Here is a list of detection queries crafted after considering the detection strategy. They can be used for alerting on Security Account Manager credential dumping attempts:

Alert Rules for Security Account Manager:

Transfering Files with Credential Data via Network Shares
Copying Sensitive Files with Credential Data
Credential Dump Tools Dropped Files Detected
Mimikatz Command Line Detected
Microsoft Build Engine Loading Credential Libraries
NTDS or SAM Database Copy Operation

T1003.003: NTDS

The NTDS.dit file is a database of domain controllers that holds all Active Directory data. This file is replicated among domain controllers within a domain or forest. Their hashed passwords are stored in the NTDS for user accounts in the Active Directory.dit file, enabling authentication across all domain-joined machines. Threat T1003.003 is a potent threat vector among the numerous techniques employed. This technique involves the extraction of the Active Directory (AD) database, NTDS.dit, which contains crucial credential information. Understanding how threat actors leverage this technique is paramount for enhancing defensive strategies and mitigating risks.

As per MITRE's ATT&CK framework, **T1003.003** encompasses various methods adversaries employ to extract the Active Directory database. This database is a goldmine for attackers, as it houses essential user credential information, including password hashes.

Thread Actors Utilizing NTDS

1. **APT28:** Known for its sophisticated tactics, APT28 has utilized ntdsutil.exe to export the AD database, facilitating credential access.
2. **APT41:** Similar to APT28, APT41 utilized ntdsutil to obtain a copy of the victim's NTDS.dit file, indicating a focus on credential acquisition.
3. **Chimera:** This threat group utilized the NtdsAudit tool with ntdsutil to dump password hashes, demonstrating a multifaceted approach to credential theft.
4. **CrackMapExec:** Utilizes Windows' Directory Replication Services API (DRSUAPI) to obtain hashed passwords related to Active Directory and then dumps them.
5. **Cutting Edge:** To extract NTDS.dit, threat actors accessed and mounted virtual hard drive backups, showcasing the exploitation of backup mechanisms for credential theft.
6. **Dragonfly:** Using SecretsDump, Dragonfly successfully extracted password hashes and obtained NTDS.dit from domain controllers.
7. **FIN13:** This threat actor harvested the NTDS.DIT file was locally decrypted using the Impacket tool on the compromised domain controller.
8. **FIN6:** Utilizing Metasploit's PsExec NTDSGRAB module, FIN6 obtained copies of victims' AD databases, emphasizing ease of access to sensitive information.
9. **Fox Kitten:** Utilizing Volume Shadow Copy, Fox Kitten accessed credential information from NTDS, showcasing a stealthy extraction technique.
10. **HAFNIUM:** Stolen copies of the Active Directory database (NTDS.DIT) indicate a strategic focus on credential access.
11. **Ke3chang:** Utilizing NTDSDump and other password-dumping tools, Ke3chang gathered credentials, highlighting a persistent threat.
12. **Koadic:** Koadic gathered hashed passwords by gathering domain controller hashes from NTDS, emphasizing exploiting system weaknesses.
13. **LAPSUS$:** Leveraging Windows built-in tool ntdsutil, LAPSUS$ extracted the AD database, showcasing the simplicity of exploitation.
14. **menuPass:** Utilizing Ntdsutil, menuPass dumped credentials, indicating a targeted approach towards credential theft.
15. **Mustang Panda:** Utilizing vssadmin and reg save, Mustang Panda created volume shadow copies and extracted NTDS.dit, emphasizing resourcefulness in data exfiltration.
16. **Sandworm Team:** Using ntdsutil.exe, the Sandworm Team backed up the AD database, showcasing a systematic approach to data theft.
17. **Scattered Spider:** Scattered Spider successfully extracted NTDS.dit through volume shadow copies of virtual domain controller disks, highlighting the exploitation of system features.
18. **Volt Typhoon:** Utilizing ntds.util, Volt Typhoon created domain controller installation media containing credentials, showcasing a unique extraction method.
19. **Wizard Spider:** By exporting copies of NTDS.dit and creating volume shadow copies, Wizard Spider gained access to valuable credentials, emphasizing adaptability in attack methodologies.

Detection

You are fine if you enable the above audit logging in LSASS Dumping Detection.

Focusing on executed commands and arguments, particularly those involving known Windows utilities and parameters for such actions is essential to effectively monitor and detect attempts to access or create a copy of the Active Directory domain database (NTDS.dit).

By taking advantage of Windows Security Auditing and Windows Sysmon, one can identify and watch for any commands and arguments being run that might try to access or duplicate the NTDS.dit file to steal credential information or obtain other domain member information (devices, users, access rights).

```
label="Process" label=Create
(("process" IN ["*\NTDSDump.exe", "*\NTDSDumpEx.exe","*\ntdsutil.exe"])
OR (command="*ntds.dit*" command="*system.hiv*")
OR (command="*NTDSgrab.ps1*"))
OR (command="*ac i ntds*" command="*create full*")
OR (command="*/c copy *" command="*\windows\ntds\ntds.dit*")
OR (command="*activate instance ntds*" command="*create full*")
OR (command="*powershell*" command="*ntds.dit*")
OR (command="*ntds.dit*" "process" IN ["*\apache*", "*\tomcat*", "*\AppData\*", "*\Temp\*", "*\Public\*", "*\PerfLogs\*"]
OR "parent_process" IN ["*\apache*", "*\tomcat*", "*\AppData\*", "*\Temp\*", "*\Public\*", "*\PerfLogs\*"])
```

Additionally, the SACL should be set up to audit both read and write access to the NTDS.dit file, and monitoring access rights for these operations helps to detect attempts to access or copy the NTDS.dit file.

```
(label=Object label=Access access_list IN ["%%4416","%%4419", "%%4417", "%%4424"])
OR
(label=Create label=File label=Overwrite file="*ntds.dit")
```

Alert Rules for NTDS:

Transfering Files with Credential Data via Network Shares
Copying Sensitive Files with Credential Data
Credential Dump Tools Dropped Files Detected
Invocation of Active Directory Diagnostic Tool Detected
Activity Related to NTDS Domain Hash Retrieval
Active Directory Database Dump Attempt
NTDS or SAM Database Copy Operation
Suspicious Activities Associated with NTDS Exfiltration

T1003.004: LSA SECRETS

The Local Security Authority (LSA) is a core component of the Windows operating system responsible for enforcing security policies and managing user authentication. One of LSA's key responsibilities is to authenticate users and ensure that users have the necessary permissions to access specific resources. When users log into a Windows system, LSA authenticates their credentials and assigns them a security token. This token determines the user's access level to various resources, such as files, applications, and network shares. The LSA securely stores sensitive information, including passwords and cryptographic keys, which are critical for maintaining the integrity and security of the authentication process. These stored secrets are used for authenticating services, scheduling tasks, and performing other password-required activities. By managing these credentials, LSA is vital in ensuring users can only access the resources they are authorized to use and protecting the system from unwanted access.

Attacks utilizing LSA Secrets

1. **AADInternals:** By exploiting vulnerabilities within the Local Security Authority, AADInternals can dump secrets effectively, highlighting the potential risks associated with insider threats.
2. **APT29:** Utilizing the reg save command, APT29 extracts LSA secrets offline, demonstrating a sophisticated approach to credential theft.
3. **APT33:** Leveraging publicly available tools like LaZagne, APT33 gathers credentials, showcasing the utilization of off-the-shelf software for malicious purposes.
4. **CosmicDuke:** By collecting LSA secrets, CosmicDuke demonstrates a focused effort to compromise system integrity and acquire sensitive information.
5. **CrackMapExec:** With the capability to dump hashed passwords from LSA secrets, CrackMapExec poses a significant risk to targeted systems, emphasizing the need for robust security measures.
6. **Dragonfly:** Employing SecretsDump, Dragonfly successfully extracts password hashes, indicating a persistent threat to system security.
7. **gsecdump:** Threat actors can dump LSA secrets using gsecdump, showcasing the exploitation of system vulnerabilities for credential theft.
8. **IceApple:** IceApple's Credential Dumper module targets registry keys to dump LSA secrets, highlighting the diverse range of attack vectors utilized by adversaries.
9. **Impacket:** SecretsDump and Mimikatz modules within Impacket perform credential dumping, showcasing the versatility of this toolset in extracting sensitive information.
10. **Ke3chang:** By leveraging gsecdump, Ke3chang dumps credentials, underscoring the ongoing threat posed by persistent adversaries.
11. **LaZagne:** LaZagne is employed for credential dumping from LSA secrets, showcasing its widespread usage among threat actors for malicious activities.
12. **Leafminer:** Utilizing tools like LaZagne, Leafminer retrieves login and password information, highlighting the need for comprehensive security measures to thwart such attacks.
13. **menuPass:** menuPass utilizes modified pen-testing tools to dump credentials, showcasing the customization of off-the-shelf software for malicious purposes.
14. **Mimikatz:** With its multifaceted functionality, Mimikatz performs credential dumping from LSA secrets, emphasizing the need for proactive detection and response strategies.
15. **MuddyWater:** Utilizing LaZagne for credential dumping, MuddyWater demonstrates a persistent threat to system integrity and data confidentiality.
16. **OilRig:** Employing tools like LaZagne, OilRig steals credentials for unauthorized access, underscoring the need for enhanced authentication mechanisms and access controls.
17. **Pupy:** Pupy utilizes LaZagne for credential harvesting, showcasing this tool's adaptability in various malicious scenarios.
18. **Threat Group-3390:** Utilizing gsecdump, Threat Group-3390 actors dump credentials, highlighting the widespread adoption of this technique among threat groups for malicious activities.

Detection

Logging must be enabled for the Audit File System, permission on the **object was changed**, and Registry auditing for HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets

After enabling logging, monitor executed commands and arguments that may attempt to access Local Security Authority (LSA) secrets on a host. Additionally, monitor attempts to access LSA secrets stored in the registry at HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets.

Alert Rules for LSA Secrets:
Credential Dump Tools Dropped Files Detected
DPAPI Domain Backup Key Extraction Detected
DPAPI Domain Master Key Backup Attempt
Grabbing Sensitive Hives via Reg Utility
Mimikatz Command Line Detected
Password Dumper Activity on LSASS

T1003.005: CACHED DOMAIN CREDENTIALS

T1003.005, as defined by MITRE's ATT&CK framework, encompasses techniques adversaries utilize to extract cached domain credentials from compromised systems. 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. Threat actors can target them to gain unauthorized access to sensitive information. This allows the user to continue accessing network resources if the domain controller becomes unavailable. These cached credentials are typically stored in the LSASS memory. They can be used to authenticate the user even if the domain controller is not reachable.

Threat Actors Utilizing Cached Domain Credentials

1. **APT33:** APT33 utilizes publicly available tools like LaZagne to gather credentials, indicating a reliance on off-the-shelf software for malicious activities.
2. **Cachedump:** Cachedump extracts cached password hashes from cache entry information, providing threat actors access to valuable credential data.
3. **LaZagne:** LaZagne performs credential dumping from MSCache to obtain account and password information, showcasing its widespread usage among threat actors for extracting cached domain credentials.
4. **Leafminer:** This tool retrieves login and password information, utilizing tools like LaZagne, highlighting its versatility in credential theft scenarios.
5. **MuddyWater:** MuddyWater performs credential dumping with LaZagne, indicating a persistent threat to system integrity and data confidentiality.
6. **OilRig:** OilRig employs credential dumping tools like LaZagne to steal credentials, emphasizing the threat posed by sophisticated adversaries targeting cached domain credentials.
7. **Okrum:** Okrum utilizes modified Quarks PwDump for credential dumping, showcasing the adaptation of existing tools for malicious purposes.
8. **Pupy:** Pupy leverages LaZagne for credential harvesting, indicating the widespread usage of this tool among various threat actors for extracting cached domain credentials.

Detection

File System Auditing should be enabled to monitor object access, deletion, and permission change.

Registry Auditing should be enabled for `HKLM\Security`, `HKLM\Security\Cache`, `HKLM\System`.

To detect attempts to steal cached domain credentials, which are used for offline authentication, focus on two areas:

1. **Monitoring Executed Commands and Arguments:** Look for commands associated with remote access tools or credential dumping scripts (e.g., Mimikatz) that might target cached domain credentials.
2. **Identifying Compromised Accounts:** Monitor valid accounts used in suspicious contexts, as attackers might leverage compromised credentials to access cached information.

Alert Rules for Cached Domain Credentials:

Cmdkey Cached Credentials Recon Detected

Credential Dump Tools Dropped Files Detected

Grabbing Sensitive Hives via Reg Utility

Mimikatz Command Line Detected

T1003.006: DCSYNC

DCSync is a technique utilized by adversaries to perform domain replication and extract credentials from Active Directory Domain Controllers (DCs) using DCSync commands. Members of the Administrators, Domain Admins, and Enterprise Admins groups or computer accounts on the domain controller can run DCSync, which can be used to retrieve password data from Active Directory, including current and historical hashes of important accounts like KRBTGT and Administrators. These hashes can then be utilized to generate a Golden Ticket for Pass the Ticket attacks or to change an account's password, as mentioned in Account Manipulation. To launch a DCSync attack and steal password hashes, attackers target Active Directory accounts with permissions like "Replicating Directory Changes" or "Replicating Directory Changes All," which are typically granted to accounts in the Administrators, Domain Admins, Enterprise Admins, and Domain Controllers groups by default; or any account with privileges like "GenericAll (Full Control)" or "AllExtendedRights." However, improper privilege management can lead to over-privileged accounts being vulnerable to exploitation by DCSync.

How it works

The core of DCSync is the DsGetNCChanges function. This function, part of the Directory Replication Service API (DRSUAPI), is used for data replication between Domain Controllers (DCs) in AD. The attacker uses a compromised account with the necessary permissions to send a Remote Procedure Call (RPC) request. This request targets the DRSUAPI on a target DC. Then, the attacker tailors the request to retrieve specific data, primarily password hashes (like NTLM) and Kerberos keys. Unlike directly dumping the NTDS.dit file (which stores AD data), DCSync leverages replication. It extracts a copy of the desired data (credentials, domain secrets) during replication. This makes it stealthier. Furthermore, the "lsadump" module in Mimikatz now provides **DCSync capabilities**. Additionally, Lsadump comes with NetSync, which facilitates DCSync attacks by performing DCSync over an antiquated replication protocol.

On Windows, mimikatz can be used with the lsadump::dcsync command to perform a DCSync attack and recover the krbtgt keys for a golden ticket attack. For this attack to work, the following mimikatz commands should be run in an elevated context (i.e., through runas with plaintext password, pass-the-hash, or pass-the-ticket).

```
lsadump::dcsync /dc:$DomainController /domain:$DOMAIN /user:srlab
```

The above command extracts a specific user srlab.

```
lsadump::dcsync /dc:$DomainController /domain:$DOMAIN /all /csv
```

The second command dumps everything in a readable format.

Threat Actor Utilizing DCSync

During a financially motivated campaign (C0027) linked to Scattered Spider, domain replication was performed, highlighting the utilization of DCSync for credential extraction.

LAPSUS$, a cyber criminal threat group, utilized DCSync attacks to gather credentials for privilege escalation routines, emphasizing the strategic use of this technique in sophisticated attack campaigns.
Earth Lusca, a cyber espionage group based in China, used the DCSync command with Mimikatz to extract credentials from a compromised controller, demonstrating this method's stealthy capabilities.
Mimikatz, a widely used tool, performs credential dumping via DCSync to acquire account and password information, demonstrating its versatility in malicious activities.
During **Operation Wocao**, threat actors utilized Mimikatz's DCSync functionality to dump credentials from the memory of targeted systems, underscoring the persistent threat posed by this technique.
In the **SolarWinds Compromise**, APT29 conducted a complex supply chain cyber operation uncovered in mid-December 2020. APT29 utilized privileged accounts to replicate directory service data with domain controllers, showcasing the exploitation of trusted relationships for credential extraction.

Detection for the DCSync Attacks

To identify DCSync attacks, monitor network traffic for replication activities from an IP address that doesn't belong to a domain controller. Watch for DsGetNCChanges requests and traffic using the DRSUAPI protocol. Event ID 4662 on DCs, which indicates a replication event occurrence, is commonly found in Windows event logs. To obtain insights about the replication event and pinpoint the accounts that need to be targeted, filter on **GUIDs** linked to DS-Replication-Get-Changes (1131f6aa-9c07-11d1-f79f-00c04fc2dcd2) and DS-Replication-Get-Changes-All (1131f6ad-9c07-11d1-f79f-00c04fc2dcd2). Logging must be enabled for both Audit Directory Service Access, object operation, and Audit File System, and monitoring for **Event ID 4670**, which indicates changes to permissions on an object, is required. In addition, Process Creation with **Command Line Auditing** should be enabled.

Alert Rules for DCSync:

Active Directory Replication User Backdoor

Mimikatz Command Line Detected

DCSync detected

Suspicious DsInternals Get-ADReplAccount Activities

To adequately protect accounts with elevated permissions, thoroughly examine account memberships within the Administrators, Domain Admins, and Enterprise Admins groups.

Ensure that the only admin accounts maintained in these groups are dedicated ones with solid passwords and adequate account security. Furthermore, locate and safeguard accounts authorized for "Replicating Directory Changes" at the domain's root. It is imperative to defend domain controllers (DCs) against DCSync attacks by using NTLMv2, installing security patches regularly, keeping the operating system up to date, and keeping an eye out for frequent user account access. One DC can be used to attack another, so protecting domain controllers (DCs) from DCSync attacks is foremost.

RECOMMENDATION AND BEST PRACTICES

To successfully tackle the issue of OS credential dumping, companies must adopt a comprehensive security policy that includes both technological and operational safeguards. The following are crucial guidelines and best practices to help you strengthen your defenses against this significant threat.

Recommendations

1. **Implement Multi-Factor Authentication (MFA):**
   - Utilize multi-factor authentication for all critical accounts to provide an extra layer of protection, making it more difficult for attackers to re-utilize stolen credentials.
2. **Regularly Update and Patch Systems:**
   - To reduce risks, verify that all operating systems, apps, and firmware have installed the most recent security updates. Keep in mind that business-critical systems must also be functioning and vulnerability-free.
3. **Limit Privileged Account Use:**
   - Restrict access to privileged accounts and apply the principle of least privilege (PoLP) to decrease the risk of credential dumping and misuse.
4. **Monitor and Audit Account Activities:**
   - Monitor account activity continuously for abnormal behavior and audit access logs regularly to discover potential credential dumping efforts.
5. **Deploy Endpoint Detection and Response (EDR) Solutions:**
   - Use EDR solutions to identify, examine, and respond to real-time suspicious activity on endpoints.
6. **Use Secure Credential Storage:**
   - Ensure credentials are securely saved using robust encryption techniques, and avoid saving plaintext passwords.
7. **Conduct Regular Security Training:**
   - Conduct periodic phishing tests to educate staff on the dangers of phishing, social engineering, and other credentials-stealing strategies and foster a security-conscious culture.
8. **Conduct Vulnerability Assessment and Penetration Testing:**
   - Conduct regular vulnerability assessment and penetration testing exercises to determine the vulnerability and penetrability of internal systems to external threats. Add the flavor of purple teaming, which can check if defense controls detect and respond to threats as expected and improve in both areas.

Best Practices

1. **Network Segmentation:**
   - Segment your network to prevent attackers' lateral movement, limiting the potential harm even if credentials are compromised.
2. **Implement Strong Password Policies and Avoid Reuse of Passwords:**
   - Enforce strong password practices, such as using complicated passwords, changing passwords regularly, and preventing password reuse.
3. **Disable Unnecessary Services:**
   - Reduce the attack surface by turning off any extra services and accounts.
4. **Utilize Threat Intelligence:**
   - Utilize threat intelligence to remain current on the newest credential dumping tactics and apply this knowledge to your security procedures.
5. **Regularly Test Security Posture:**
   - Perform frequent penetration testing and vulnerability assessments to uncover and address flaws in your security defenses.
6. **Employ Advanced Logging and SIEM Solutions:**
   - Use sophisticated logging and Security Information and Event Management (SIEM) technologies to collect and analyze logs from several sources, offering complete visibility into possible threats.
7. **Develop Incident Response Playbooks:**
   - Develop and uphold incident response playbooks tailored to credential dumping scenarios to ensure your team can respond swiftly and effectively to such incidents.

Organizations implementing these suggestions and best practices can dramatically improve their capacity to identify, prevent, and respond to OS Credential Dumping threats. Staying attentive, proactive, and aware is critical for maintaining a strong cybersecurity posture in the face of emerging threats.

CONCLUSION

As we negotiate an increasingly complicated and dynamic cybersecurity landscape, the significance of understanding and mitigating OS Credential Dumping (T1003) must be considered. This common practice among threat actors poses a substantial danger to enterprises by allowing unauthorized access to crucial systems and data. Throughout this paper, we have looked at the numerous sub-techniques utilized in credential dumping within the Windows environment, shedding light on their mechanics and consequences.

OS credential dumping is not a stand-alone danger but a necessary step in a larger attack chain, frequently preceding more damaging activities like ransomware distribution or data exfiltration. Its capacity to promote lateral movement and privilege escalation within networks emphasizes the need for robust security measures.

To combat these risks, companies must take a multifaceted approach that involves constant monitoring, robust authentication methods, frequent patch management, and extensive user education. By employing these measures, firms may drastically minimize their vulnerability to credential dumping attacks while improving their overall cybersecurity posture.

Logpoint's Emerging Threat Protection team is committed to staying ahead of these developing threats. We enable enterprises to successfully identify, analyze, and respond to OS Credential Dumping incidents by providing advanced threat information, configurable detection rules, and personalized response playbooks. Our mission is to provide the tools and expertise necessary to protect your digital assets while preserving operational integrity.

Although OS Credential Dumping signifies a severe obstacle, it is not insurmountable. Organizations may protect vital resources and build resilience against future cyber attacks by understanding their methodologies and proactively adopting suggested measures. We want to encourage you to use the insights and recommendations in this study to improve your security posture and stay ahead of enemies.

Thank you for reading this report. We can build a more secure digital future together through vigilance and innovation.

ABOUT LOGPOINT

Logpoint is the creator of a reliable, innovative cybersecurity operations platform — empowering organizations worldwide to thrive in a world of evolving threats.

By combining sophisticated technology and a profound understanding of customer challenges, Logpoint bolsters security teams' capabilities while helping them combat current and future threats.

Logpoint offers SIEM, UEBA, and SOAR technologies in a complete platform that efficiently detects threats, minimizes false positives, autonomously prioritizes risks, responds to incidents, and much more.

Headquartered in Copenhagen, Denmark, with offices around the world, Logpoint is a multinational, multicultural, and inclusive company.

For more information visit **www.logpoint.com**