LIFARS
your digital world, **secured**

Mitigating Credential Dumping on Windows Clients

June 2021

MITIGATING CREDENTIAL DUMPING ON WINDOWS CLIENTS

Credential dumping from compromised Windows clients allows the attacker to perform lateral movement and gain control even after more sensitive hosts and eventually compromise the domain controller.

Mitigating various types of credential dumping on Windows is not easy. Not every permission that is abused by various post-exploitation tools can be safely removed and even the latest virtualization-based features in Windows 10 cannot protect all credentials.

In this paper we will look at most abused techniques, as well as their detection and mitigation.

WDigest Authentication – plaintext passwords

Prior to the Windows 8.1 (or Windows Server 2012 R2), LSA stored plaintexts of user passwords. This was due to the WDigest authentication, which was enabled by default. But even though it is disabled by default in all supported Windows versions, the components are still present in the operating system. Attacker with admin privileges can reenable WDigest and extract plaintext passwords from the mimikatz by adding the following registry key and setting its value to one.:

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential

Extracting plaintext passwords stored by WDigest is also a supported feature by mimikatz. Since it is such a critical registry key, monitoring changes via System Access Control List (SACL) might be beneficial.

While enabling Credential Guard does offer some limited protection against this attack, security researched presented a bypass making monitoring essential even with Credential Guard.

Additional LSA Protection

Many of the techniques consist of dumping the Local Security Authority Subsystem Service process (lsass.exe), which is responsible for authenticating users and enforcing security policies. Dumping the contents of lsass.exe is commonly abused by malware, including the Trickbot trojan.

Credentials stored by the LSASS process include domain NTLM hashes, Kerberos credentials, passwords of logged-in users, and much more. However, not all sensitive credentials are stored in memory. Some are stored on disk or potentially in plaintext (WDigest).

Protected processes were originally introduced in Windows Vista as a DRM measure. Protected processes cannot be debugged or managed by other, non-protected processes. Other processes also cannot open their virtual memory. In Windows 8.1, this feature was expanded to protect anti-malware services and LSASS. Anti-malware vendors have to opt-in into this feature (it is enabled by default in Microsoft Defender), while protected process for LSASS must be enabled via registry. Once enabled, the variable is bound to the system's UEFI firmware and disabling it requires an opt-out tool distributed by Microsoft. More information about enabling, disabling and auditing the policy can be found in the official documentation.

Even if Additional LSA Protection is enabled, LSA drivers or plug-ins can still load if they are signed with the required digital certificate. This allows post-exploitation tools to dump contents of lsass.exe even when

LSA protection is enabled. Mimikatz includes a kernel driver signed with the required certificate, although using it is bit tricky, since the detection rate with anti-malware solutions is high. Attackers can still deploy their own certificates.

Mimikatz kernel driver can be blocked either manually in Application Control, or simply by enabling HVCI (labeled as Memory Integrity in the Windows Security). HVCI features a blocklist for known-vulnerable drivers and drivers used to bypass security policies, such as mimikatz driver. HVCI has similar system requirements as Credential Guards (see below).

An alternative to Additional LSA Protection is the "Block credential stealing from the Windows local security authority subsystem (lsass.exe)" policy included in the Attack Surface Reduction ruleset (ASR). This protection is equivalent to the Additional LSA Protection and provides no additional security benefit. The only downside to the ASR rule compared to the Additional LSA Protection is that it provides no UEFI lock.

Windows 10, version 1511, added default SACL to LSASS.exe to detect access to the process and both policies support logging of events. LSASS policies can create a lot of noise in logs, as even non-malicious apps tend to enumerate LSASS.

Credential Guard

Credential Guard protects LSASS by isolating it in a virtualized container. This offers a strong level of protection, which protects secrets even from a compromised kernel. Credential Guard was first introduced with Windows 10, version 1507 and further improved with newer versions. Similarly, to Additional LSA Protection, Credential Guard can be configured with UEFI lock,

Because Credential Guard is using virtualization, there are additional hardware requirements:

- Intel VT-D\AMD-Vi and SLAT support
- At least Kaby Lake or Zen 2 based CPU for best performance
- TPM 2.0 for best security
- Compatible firmware
- Compatible drivers

To verify the system's drivers, you can use the Device Guard and Credential Guard hardware readiness tool.

Credential Guard is based on Hyper-V platform; therefore, it might cause compatibility issues with other hypervisors. The major benefit is support for enabling it inside other Hyper-V powered guests.

However, Credential Guard will not protect against all kinds of credential dumping attacks. It is important to realize that Credential Guard protects only in-memory stored credentials. Newly entered credentials can still be stolen either via a keylogger or registering a new Security Support Provider (SSP). This attack allows logging of credentials of any account login in plain-text and cannot be mitigated with neither Credential Guard nor Additional LSA Protection.

LIFARS
your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

While such attacks cannot be easily mitigated, they can be detected by monitoring changes to the registry key:

HKEY_LOCAL_MACHINE|System|CurrentControlSet|Control|Lsa|Security Packages,

which needs to be updated to install new SSPs.

Other than that, Credential Guard does not protect credentials for NTLM authentication, Kerberos service tickets and WDigest credentials. Credential Guard also does not offer any protection for the Active Directory database on domain controllers.

Hashes stored by SAM

Security Account Manager (SAM) stores the LM or NT hashes of local user passwords. This includes Microsoft accounts potentially with access to BitLocker recovery keys, OneDrive, and Outlook.

LM hashes are known to be cryptographically weak and have major limitations to the password complexity. When computing the LM hash, user passwords are limited to fourteen characters and are not case sensitive. This makes LM hashes very easy to crack even without expensive and specifically built hardware. Storing of LM hashes is disabled by default since Windows Vista, but support is kept for backward compatibility reasons.

Storing of LM hash can be configured with the following policy:

Windows Settings|Security Settings|Local Policies|Security Options|Network Security: Do Not Store LAN Manager Hash Value On Next Password Change.

NT hash is significantly stronger by being case sensitive and cryptographically much stronger. Nonetheless, NT hashes are still vulnerable to rainbow tables and or cracking with dedicated hardware.

SAM database is not readable even for Administrators in the default settings, configured access control lists can be easily bypassed with backup permission (SeBackupPrivilege) and mimikatz supports dumping of SAM hashes with its lsadump module. This feature works even with Credential Guard and Additional LSA Protection enabled as it simply dumps data stored on the disk. As a mitigation it is strongly recommended to use complex passwords and long password to thwart attacks using brute force method and rainbow tables.

Another mitigation to restrict mimikatz would be removing the debug privilege (SeDebugPrivilege), which is heavily used by it. The permission allows malicious software to attach debuggers and dump contents of sensitive processes. By default, it is restricted to users in the administrators group, so malicious applications will first have to elevate their permissions. The best practice is to remove the permission from the administrators group. Debug privilege can be removed with the following policy:

Policy path: *Windows Settings|Security Settings|Local Policies|User Rights Assignment|Debug programs.*

Removing the privilege will not stop mimikatz entirely, as most of the commands can still be run with SYSTEM permissions. Elevating from Administrator to SYSTEM is not considered a security boundary.

Auditing access to registry files

While not every form of credential dumping can be stopped, most of them can be logged. Apart from logging LSASS access attempts, you can apply SACL auditing to sensitive registry keys, such as:

HKLM|Security, HKLM|Security|Cache, HKLM|Sam, HKLM|System.

Properly configured SACL for sensitive registry keys can help detect and stop attacks before attackers had enough time to utilize stolen credentials. SACLs can be configured to just keys, or both keys and subkeys. For these keys, it is recommended to monitor querying, reading, and enumerating.

Non-registry files containing sensitive data include the DPAPI master key, which resides in the following location:

*C:|Users|*Username*|AppData|Roaming|Microsoft|Protect|*UserSID*.*

Credential files are in two locations:

C:|Users|Username|AppData|Local|Microsoft|Credentials|

C:|Users|Username|AppData|Roaming|Microsoft|Credentials|

It is also worth setting up SACL for sensitive non-Windows directories, such as directories containing Keepass databases or directories containing Chrome data.

Conclusion

While Windows 10 offers a robust set of technologies designed to prevent credential dumping, no protection is bulletproof and covers every set of credentials. Even a well configured system still must be monitored and protected against malware.

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com