Configure added LSA protection

Article • 09/27/2023 • 19 contributors Feedback

In this article

This article explains how to configure added protection for the Local Security Authority (LSA) process to prevent code injection that could compromise credentials.

The LSA, which includes the Local Security Authority Server Service (LSASS) process, validates users for local and remote sign-ins and enforces local security policies. Starting with Windows 8.1 and later, added protection for the LSA is provided to prevent reading memory and code injection by nonprotected processes. This feature provides added security for the credentials that LSA stores and manages. Further protection is achieved when using UEFI lock and Secure Boot, because disabling the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa** registry key has no effect.

Protected process requirements for plug-ins or drivers

For an LSA plug-in or driver to successfully load as a protected process, it must meet the following criteria:

Signature verification

Protected mode requires any plug-in that's loaded into the LSA to be digitally signed with a Microsoft signature. Any plug-ins that are unsigned or aren't signed with a Microsoft signature fail to load in LSA. Examples of plug-ins are smart card drivers, cryptographic plug-ins, and password filters.

- LSA plug-ins that are drivers, such as smart card drivers, need to be signed by using the WHQL Certification. For more information, see WHQL Release Signature.
- LSA plug-ins that don't have a WHQL Certification process must be signed by using the file signing service for LSA.

Adherence to Microsoft Security Development Lifecycle (SDL) process guidance

- All plug-ins must conform to the applicable SDL process guidance. For more information, see the Microsoft Security Development Lifecycle (SDL) – Process Guidance.
- Even if the plug-ins are properly signed with a Microsoft signature, noncompliance with the SDL process can result in failure to load a plug-in.

Recommended practices

Use the following list to thoroughly test enabling LSA protection before you broadly deploy the feature:

- Identify all of the LSA plug-ins and drivers that your organization uses. Include non-Microsoft drivers or plug-ins such as smart card drivers and cryptographic plug-ins, and any internally developed software that's used to enforce password filters or password change notifications.
- Ensure that all of the LSA plug-ins are digitally signed with a Microsoft certificate so they don't fail to load under LSA protection.
- Ensure that all of the correctly signed plug-ins can successfully load into LSA and that they perform as expected.
- Use the audit logs to identify LSA plug-ins and drivers that fail to run as a protected process.

Limitations of enabling LSA protection

If added LSA protection is enabled, you can't debug a custom LSA plug-in. You can't attach a debugger to LSASS when it's a protected process. In general, there's no supported way to debug a running protected process.

Audit for LSA plug-ins and drivers that won't load as a protected process

Before you enable LSA protection, use audit mode to identify LSA plug-ins and drivers that will fail to load in LSA protected mode. While in audit mode, the system generates event logs that identify all of the plug-ins and drivers that fail to load under LSA if LSA protection is enabled. The messages are logged without actually blocking the plug-ins or drivers.

The events described in this section are located in Event Viewer in the **Operational** log under **Applications and Services Logs** > **Microsoft** > **Windows** > **CodeIntegrity**. These events can help you identify LSA plug-ins and drivers that fail to load due to signing reasons. To manage these events, you can use the **wevtutil** command-line tool. For information about this tool, see Wevtutil.

> **Important**
>
> Audit events aren't generated if Smart App Control is enabled on a device. To check or change the enablement state of Smart App Control, open the Windows Security Application and go to the **App & browser control** page. Select **Smart App Control settings** to check the enablement state, and change the configuration to **Off** if you're trying to audit added LSA protection.

> **Note**
>
> Audit mode for added LSA protection is enabled by default on devices running Windows 11 version 22H2 and higher. If your device is running this build or later, no other actions are needed to audit added LSA protection.

Enable audit mode for LSASS.exe on a single computer

1. Open the Registry Editor (RegEdit.exe) and navigate to the registry key at **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe**.
2. Set the value of the registry key to **AuditLevel=dword:00000008**.
3. Restart the computer.

After taking these steps, analyze the results of event 3065 and event 3066. In Event Viewer, check for these events in the **Operational** log under **Applications and Services Logs** > **Microsoft** > **Windows** > **CodeIntegrity**.

- **Event 3065** records that a code integrity check determined that a process, usually LSASS.exe, attempted to load a driver that didn't meet the security requirements for Shared Sections. However, due to the system policy currently set, the image was allowed to load.
- **Event 3066** records that a code integrity check determined that a process, usually LSASS.exe, attempted to load a driver that didn't meet the Microsoft signing level requirements. However, due to the system policy currently set, the image was allowed to load.

If a plug-in or driver contains Shared Sections, Event 3066 is logged with Event 3065. Removing the Shared Sections should prevent both events from occurring unless the plug-in doesn't meet the Microsoft signing level requirements.

> **Important**
>
> These operational events aren't generated when a kernel debugger is attached and enabled on a system.

Enable audit mode for LSASS.exe on multiple computers

To enable audit mode for multiple computers in a domain, you can use the Registry Client-Side Extension for Group Policy to deploy the LSASS.exe audit-level registry value. You need to modify the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe** registry key.

1. Open the Group Policy Management Console (GPMC) by entering *gpmc.msc* in the Run dialog box or selecting **Group Policy Management Console** from the Start menu.
2. Create a new Group Policy Object (GPO) that's linked at the domain level or linked to the organizational unit that contains your computer accounts. Or, select a GPO that's already deployed.
3. Right-click the GPO, and then select **Edit** to open the Group Policy Management Editor.
4. Expand **Computer Configuration** > **Preferences** > **Windows Settings**.
5. Right-click **Registry**, point to **New**, and then select **Registry Item**. The **New Registry Properties** dialog box appears.
6. In the **Hive** list, select **HKEY_LOCAL_MACHINE.**
7. In the **Key Path** list, browse to **SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe**.
8. In the **Value name** box, type **AuditLevel**.
9. In the **Value type** box, select **REG_DWORD**.
10. In the **Value data** box, type **00000008**.

11. Select **OK**.

> **Note**
>
> For the GPO to take effect, the GPO change must be replicated to all domain controllers in the domain.

To opt in for added LSA protection on multiple computers, you can use the Registry Client-Side Extension for Group Policy to modify **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa**. For instructions, see Configure added LSA credentials protection later in this article.

Identify plug-ins and drivers that LSASS.exe fails to load

When LSA protection is enabled, the system generates event logs that identify all of the plug-ins and drivers that fail to load under LSA. After you opt in to added LSA protection, you can use the event log to identify LSA plug-ins and drivers that failed to load in LSA protection mode.

Check for the following events in Event Viewer **Applications and Services Logs** > **Microsoft** > **Windows** > **CodeIntegrity** > **Operational**:

- **Event 3033** records that a code integrity check determined that a process, usually LSASS.exe, attempted to load a driver that didn't meet the Microsoft signing level requirements.
- **Event 3063** records that a code integrity check determined that a process, usually LSASS.exe, attempted to load a driver that didn't meet the security requirements for Shared Sections.

Shared Sections are typically the result of programming techniques that allow instance data to interact with other processes that use the same security context, which can create security vulnerabilities.

Enable and configure added LSA credentials protection

You can configure added LSA protection for devices running Windows 8.1 or later, or Windows Server 2012 R2 or later, by using the procedures in this section.

Devices that use Secure Boot and UEFI

When you enable LSA protection on x86-based or x64-based devices that use Secure Boot or UEFI, you can store a UEFI variable in the UEFI firmware by using a registry key or policy. When enabled with UEFI lock, LSASS runs as a protected process and this setting is stored in a UEFI variable in the firmware.

When the setting is stored in the firmware, the UEFI variable can't be deleted or changed to configure added LSA protection by modifying the registry or by policy. The UEFI variable must be reset by using the instructions at Remove the LSA protection UEFI variable.

When enabled without a UEFI lock, LSASS runs as a protected process and this setting isn't stored in a UEFI variable. This setting is applied by default on devices with a new install of Windows 11 version 22H2 or later.

On x86-based or x64-based devices that don't support UEFI or where Secure Boot is disabled, you can't store the configuration for LSA protection in the firmware. These devices rely solely on the presence of the registry key. In this scenario, it's possible to disable LSA protection by using remote access to the device. Disablement of LSA protection doesn't take effect until the device reboots.

Automatic enablement

For client devices running Windows 11 version 22H2 and later, added LSA protection is enabled by default if the following criteria are met:

- The device is a new install of Windows 11 version 22H2 or later, not upgraded from a previous release.
- The device is enterprise joined (Active Directory domain joined, Microsoft Entra domain joined, or hybrid Microsoft Entra domain joined).
- The device is capable of Hypervisor-protected code integrity (HVCI).

Automatic enablement of added LSA protection on Windows 11 version 22H2 and later doesn't set a UEFI variable for the feature. If you want to set a UEFI variable, you can use a registry configuration or policy.

> **Note**
>
> For devices running Windows RT 8.1, added LSA protection is always enabled, and it can't be turned off.

Enable LSA protection on a single computer

You can enable LSA protection on a single computer by using the registry or by using Local Group Policy.

Enable by using the registry

1. Open the Registry Editor RegEdit.exe, and navigate to the registry key **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa**.
2. Set the value of the registry key to:

   - **"RunAsPPL"=dword:00000001** to configure the feature with a UEFI variable.
   - **"RunAsPPL"=dword:00000002** to configure the feature without a UEFI variable, only enforced on Windows 11 build 22H2 and higher.

3. Restart the computer.

Enable by using Local Group Policy on Windows 11 version 22H2 and later

1. Open the Local Group Policy Editor by entering *gpedit.msc*.
2. Expand **Computer Configuration** > **Administrative Templates** > **System** > **Local Security Authority**.
3. Open the **Configure LSASS to run as a protected process** policy.
4. Set the policy to **Enabled**.
5. Under **Options**, select one of the following options.

   - **Enabled with UEFI Lock** to configure the feature with a UEFI variable.
   - **Enabled without UEFI Lock** to configure the feature without a UEFI variable.

6. Select **OK**.
7. Restart the computer.

Enable LSA protection by using Group Policy

1. Open the GPMC by entering *gpmc.msc* in the Run dialog box or selecting **Group Policy Management Console** from the Start menu.
2. Create a new GPO that's linked at the domain level or linked to the organizational unit that contains your computer accounts. Or, select a GPO that's already deployed.
3. Right-click the GPO, and then select **Edit** to open the Group Policy Management Editor.
4. Expand **Computer Configuration** > **Preferences** > **Windows Settings**.
5. Right-click **Registry**, point to **New**, and then select **Registry Item**. The **New Registry Properties** dialog box appears.
6. In the **Hive** list, select **HKEY_LOCAL_MACHINE.**
7. In the **Key Path** list, browse to **SYSTEM\CurrentControlSet\Control\Lsa**.
8. In the **Value name** box, type **RunAsPPL**.
9. In the **Value type** box, select **REG_DWORD**.
10. In the **Value data** box, type:

- **00000001** to enable LSA protection with a UEFI variable.
- **00000002** to enable LSA protection without a UEFI variable, only enforced on Windows 11 version 22H2 and later.

11. Select **OK**.

Enable LSA protection by creating a custom device configuration profile

For devices running Windows 11 version 22H2 and later, you can enable and configure LSA protection by creating a custom device configuration profile in the Microsoft Intune admin center.

1. In the Intune admin center, navigate to **Devices** > **Windows** > **Configuration profiles** and select **Create profile**.
2. On the **Create a profile** screen, select the following options:

   - **Platform**: Windows 10 and later
   - **Profile type**: Select **Templates**, and then select **Custom**.

3. Select **Create**.
4. On the **Basics** screen, enter a **Name** and optional **Description** for the profile, and then select **Next**.
5. On the **Configuration settings** screen, select **Add**.
6. On the **Add row** screen, provide the following information:

   - **Name**: Provide a name for the OMA-URI setting.
   - **OMA-URI**: Enter *./Device/Vendor/MSFT/Policy/Config/LocalSecurityAuthority/ConfigureLsaProtecte*
   - **Data type**: Select **Integer**.
   - **Value**: Enter *1* to configure LSASS to run as a protected process with UEFI lock, or *2* to configure LSASS to run as a protected process without UEFI lock.

7. Select **Save**, and then select **Next**.
8. On the **Assignments** page, configure the assignments, and then select **Next**.
9. On the **Applicability Rules** page, configure any applicability rules, and then select **Next**.
10. On the **Review + create** page, verify the configuration, and then select **Create**.
11. Restart the computer.

For more information about this Policy CSP, see LocalSecurityAuthority - ConfigureLsaProtectedProcess.

Disable LSA protection

You can disable LSA protection by using the registry or by using Local Group Policy. If the device is using Secure Boot and you set the LSA protection UEFI variable in the firmware, you can use a tool to remove the UEFI variable.

Disable by using the registry

1. Open the Registry Editor, RegEdit.exe, and navigate to the registry key **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa**.
2. Set the value of the registry key to **"RunAsPPL"=dword:00000000**, or delete the **DWORD**.
3. If PPL was enabled with a UEFI variable, use the Local Security Authority Protected Process Opt-out tool to remove the UEFI variable.
4. Restart the computer.

Disable by using local policy on Windows 11 version 22H2 and later

1. Open the Local Group Policy Editor by entering *gpedit.msc*.
2. Expand **Computer Configuration** > **Administrative Templates** > **System** > **Local Security Authority**.
3. Open the **Configure LSASS to run as a protected process** policy.
4. Set the policy to **Enabled**.
5. Under **Options**, select **Disabled**.
6. Select **OK**.
7. Restart the computer.

> **Note**
>
> If you set this policy to **Not Configured** and the policy was previously enabled, the prior setting doesn't get cleaned up and continues to be enforced. You must set the policy to **Disabled** under the **Options** dropdown to disable the feature.

Remove the LSA protection UEFI variable

You can use the Local Security Authority (LSA) Protected Process Opt-out tool (LSAPPLConfig) from the Microsoft Download Center to delete the UEFI variable if the device is using Secure Boot.

> **Note**

The Download Center offers two files named *LsaPplConfig.efi*. The smaller file is for x86-based systems and the larger file is for x64-based systems.

For more information about managing Secure Boot, see UEFI Firmware.

> ⊗ **Caution**
>
> When Secure Boot is turned off, all the Secure Boot and UEFI-related configurations are reset. You should turn off Secure Boot only when all other means to disable LSA protection have failed.

Verify LSA protection

To determine whether LSA started in protected mode when Windows started, check **Windows Logs** > **System** in Event Viewer for the following **WinInit** event:

- **12: LSASS.exe was started as a protected process with level: 4**

LSA and Credential Guard

LSA protection is a security feature that defends sensitive information like credentials from theft by blocking untrusted LSA code injection and process memory dumping. LSA protection runs in the background by isolating the LSA process in a container and preventing other processes, like malicious actors or apps, from accessing the feature. This isolation makes LSA Protection a vital security feature, which is why it's enabled by default in Windows 11.

Starting in Windows 10, Credential Guard also helps prevent credential theft attacks by protecting NTLM password hashes, Kerberos Ticket Granting Tickets (TGTs), and credentials stored by applications as domain credentials. Kerberos, NTLM, and Credential Manager isolate secrets by using virtualization-based security (VBS).

With Credential Guard enabled, the LSA process talks to a component called the isolated LSA process, or LSAIso.exe, that stores and protects secrets. Data stored by the isolated LSA process is protected by using VBS and isn't accessible to the rest of the operating system. LSA uses remote procedure calls to communicate with the isolated LSA process.

Starting in Windows 11 version 22H2, VBS and Credential Guard are enabled by default on all devices that meet the system requirements. Credential Guard is supported on 64-bit Secure Boot devices only. LSA protection and Credential Guard are complementary, and systems that support Credential Guard or have it enabled by default can also enable and benefit from LSA protection. For more information about Credential Guard, see Credential Guard overview.

More resources

- Credential protection and management
- Partner Center for Windows Hardware

Feedback

Was this page helpful? Yes No