

Configuring Additional LSA Protection

By Archiveddocs

Archived: 2026-04-05 16:24:51 UTC

Applies To: Windows 8.1, Windows Server 2012 R2

This topic for the IT professional explains how to configure additional 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. The Windows 8.1 operating system provides additional protection for the LSA to prevent reading memory and code injection by non-protected processes. This provides added security for the credentials that the LSA stores and manages. The protected process setting for LSA can be configured in Windows 8.1, but it cannot be configured in Windows RT 8.1. When this setting is used in conjunction with Secure Boot, additional protection is achieved 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:

1. Signature verification

Protected mode requires that any plug-in that is loaded into the LSA is digitally signed with a Microsoft signature. Therefore, any plug-ins that are unsigned or are not signed with a Microsoft signature will fail to load in LSA. Examples of these 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 \(Windows Drivers\)](#).

LSA plug-ins that do not have a WHQL Certification process, must be signed by using the [file signing service for LSA](#).

2. Adherence to the Microsoft Security Development Lifecycle (SDL) process guidance

All of the plug-ins must conform to the applicable SDL process guidance. For more information, see the [Microsoft Security Development Lifecycle \(SDL\) Appendix](#).

Even if the plug-ins are properly signed with a Microsoft signature, non-compliance with the SDL process can result in failure to load a plug-in.

Recommended practices

Use the following list to thoroughly test that LSA protection is enabled before you broadly deploy the feature:

- Identify all of the LSA plug-ins and drivers that are in use within your organization. This includes non-Microsoft drivers or plug-ins such as smart card drivers and cryptographic plug-ins, and any internally developed software that is 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 that the plug-in will not fail to load.
- 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.

How to identify LSA plug-ins and drivers that fail to run as a protected process

The events described in this section are located in the Operational log under Applications and Services Logs\Microsoft\Windows\CodeIntegrity. They can help you identify LSA plug-ins and drivers that are failing 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 \[Vista\]](#).

Before opting in: How to identify plug-ins and drivers loaded by the lsass.exe

You can use the audit mode to identify LSA plug-ins and drivers that will fail to load in LSA Protection mode. While in the audit mode, the system will generate event logs, identifying all of the plug-ins and drivers that will fail to load under LSA if LSA Protection is enabled. The messages are logged without blocking the plug-ins or drivers.

To enable the audit mode for Lsass.exe on a single computer by editing the Registry

1. Open the Registry Editor (RegEdit.exe), and navigate to the registry key that is located 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.

Analyze the results of event 3065 and event 3066.

- **Event 3065:** This event records that a code integrity check determined that a process (usually lsass.exe) attempted to load a particular driver that did not meet the security requirements for Shared Sections. However, due to the system policy that is set, the image was allowed to load.
- **Event 3066:** This event records that a code integrity check determined that a process (usually lsass.exe) attempted to load a particular driver that did not meet the Microsoft signing level requirements. However, due to the system policy that is set, the image was allowed to load.

Important

These operational events are not generated when a kernel debugger is attached and enabled on a system. If a plug-in or driver contains Shared Sections, Event 3066 is logged with Event 3065. Removing the Shared Sections should prevent both the events from occurring unless the plug-in does not meet the Microsoft signing level requirements.

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 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe registry key.

To create the AuditLevel value setting in a GPO

1. Open the Group Policy Management Console (GPMC).
2. Create a new Group Policy Object (GPO) that is linked at the domain level or that is linked to the organizational unit that contains your computer accounts. Or you can select a GPO that is already deployed.
3. Right-click the GPO, and then click **Edit** to open the Group Policy Management Editor.
4. Expand **Computer Configuration**, expand **Preferences**, and then expand **Windows Settings**.
5. Right-click **Registry**, point to **New**, and then click **Registry Item**. The **New Registry Properties** dialog box appears.
6. In the **Hive** list, click **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, click to select the **REG_DWORD**.
10. In the **Value data** box, type **00000008**.
11. Click **OK**.

Note

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

To opt-in for additional LSA protection on multiple computers, you can use the Registry Client-Side Extension for Group Policy by modifying HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa. For steps about how to do this, see [How to configure additional LSA protection of credentials](#) in this topic.

After opting in: How to identify plug-ins and drivers loaded by the lsass.exe

You can use the event log to identify LSA plug-ins and drivers that failed to load in LSA Protection mode. When the LSA protected process is enabled, the system generates event logs that identify all of the plug-ins and drivers that failed to load under LSA.

Analyze the results of Event 3033 and Event 3063.

- **Event 3033:** This event records that a code integrity check determined that a process (usually lsass.exe) attempted to load a driver that did not meet the Microsoft signing level requirements.
- **Event 3063:** This event records that a code integrity check determined that a process (usually lsass.exe) attempted to load a driver that did not 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. This can create security vulnerabilities.

How to configure additional LSA protection of credentials

On devices running Windows 8.1 (with or without Secure Boot or UEFI), configuration is possible by performing the procedures described in this section. For devices running Windows RT 8.1, lsass.exe protection is always enabled, and it cannot be turned off.

- For information about changes in Secure Boot in Windows 8 and Windows 8.1, see [Secure Boot](#).
- For information about UEFI in Windows 8 and Windows 8.1, see [What's Changed in Security Technologies in Windows 8.1 \[Win 8.1\]](#).

On x86-based or x64-based devices using Secure Boot and UEFI or not

On x86-based or x64-based devices that use Secure Boot and UEFI, a UEFI variable is set in the UEFI firmware when LSA protection is enabled by using the registry key. When the setting is stored in the firmware, the UEFI variable cannot be deleted or changed in the registry key. The UEFI variable must be reset.

x86-based or x64-based devices that do not support UEFI or Secure Boot are disabled, cannot store the configuration for LSA protection in the firmware, and rely solely on the presence of the registry key. In this scenario, it is possible to disable LSA protection by using remote access to the device.

You can use the following procedures to enable or disable LSA protection:

To enable LSA protection on a single computer

1. Open the Registry Editor (RegEdit.exe), and navigate to the registry key that is located at:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa.
2. Set the value of the registry key to: "RunAsPPL"=dword:00000001.

3. Restart the computer.

To enable LSA protection using Group Policy

1. Open the Group Policy Management Console (GPMC).
2. Create a new GPO that is linked at the domain level or that is linked to the organizational unit that contains your computer accounts. Or you can select a GPO that is already deployed.
3. Right-click the GPO, and then click **Edit** to open the Group Policy Management Editor.
4. Expand **Computer Configuration**, expand **Preferences**, and then expand **Windows Settings**.
5. Right-click **Registry**, point to **New**, and then click **Registry Item**. The **New Registry Properties** dialog box appears.
6. In the **Hive** list, click **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, click the **REG_DWORD**.
10. In the **Value data** box, type **00000001**.
11. Click **OK**.

To disable LSA protection

1. Open the Registry Editor (RegEdit.exe), and navigate to the registry key that is located at:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa.
2. Delete the following value from the registry key: "RunAsPPL"=dword:00000001.
3. Use the Local Security Authority (LSA) Protected Process Opt-out tool to delete the UEFI variable if the device is using Secure Boot.

For more information about the opt-out tool, see [Download Local Security Authority \(LSA\) Protected Process Opt-out from Official Microsoft Download Center](#).

For more information about managing Secure Boot, see [UEFI Firmware](#).

Warning

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.

Verifying LSA protection

To discover if LSA was started in protected mode when Windows started, search for the following WinInit event in the **System** log under **Windows Logs**:

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

Additional resources

[Credentials Protection and Management](#)

[File signing service for LSA](#)

Source: <https://technet.microsoft.com/en-us/library/dn408187.aspx>