Understanding LSA Protection

blog.netwrix.com/2022/01/11/understanding-lsa-protection

Securing your Windows servers and Windows 10 running is vital, especially given today's sophisticated threat landscape. These are usually the first machines to be compromised in an attack through exploitation of the weakest link in the chain — the user. Through trickery and social engineering, threat actors gain access to these machines and then seek to move laterally and elevate their privileges. Therefore, enhancing endpoint and server security can significantly reduce your risk of a security breach.

Handpicked related content:
[Free Guide] Windows Server Security Best Practices

One thing you can do to harden a server is to protect the Local Security Authority (LSA). The LSA controls and manages user rights information, password hashes and other important bits of information in memory. Attacker tools, such as mimikatz, rely on accessing this content to scrape password hashes or clear-text passwords. Enabling LSA Protection configures Windows to control the information stored in memory in a more secure fashion — specifically, to prevent non-protected processes from accessing that data.

What is a Protected Process?

A process is considered protected when it meets the criteria described in this Microsoft documentation. To summarize, a process is considered protected if it has a verified signature from Microsoft and it adheres to the Microsoft Security Development Lifecycle (SDL). If those two criteria are not met, the process cannot access the content being used by the LSA in memory.

How to Enable LSA Protection

Since LSA Protection is controlled via the registry, you can enable it easily across all your devices using Group Policy: Simply set the value of **RunAsPPL** to 1. This setting can be found in the registry at **SYSTEMCurrentControlSetControlLsa**.

LSA Protection

The following code can be leveraged as a .reg file to set this value to 1:

```
*Code Block*

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINESYSTEMCurrentControlSetControlLsa]

"RunAsPPL"=dword:00000001

*Code Block*
```

Reviewing Your Settings

To verify that each server is protected, you can navigate to the section of the registry mentioned above and confirm that the value is set to 1.

However, to make the job somewhat easier, I've provided a PowerShell script below that enables you to query the value for a specific machine remotely (assuming the correct access exists). Just replace *[SAMPLEHOST]* with the hostname of the machine. This code will return 0, 1 or an exception that indicates the RunAsPPL property doesn't exist. Unless 1 is returned, the setting is not enabled on the targeted computer.

```
*Code Block*

invoke-command -Computer [SAMPLEHOST] {Get-itempropertyvalue -Path
"HKLM:SYSTEMCurrentControlSetControlLsa" -Name RunAsPPL}

*Code Block*
```

Kevin Joyce

Senior Technical Product Manager at Netwrix. Kevin is passionate about cyber-security and holds a Bachelor of Science degree in Digital Forensics from Bloomsburg University of Pennsylvania.