RSAConference™2023
San Francisco | April 24 – 27 | Moscone Center

SESSION ID: HTA-R05

The Hidden Risk in Undocumented API Behavior

#RSAC

Bahaa Naamneh
Technical Fellow
Crosspoint Labs
@bahaanaamneh

https://t.me/learningnets

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference™ or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2023 RSA Conference LLC or its affiliates. The RSA Conference logo and other trademarks are proprietary. All rights reserved.

This presentation is being given in the presenter's individual capacity and reflect the subjective views and opinions of the presenter. This presentation is not meant to express any views of Crosspoint Labs or any of its affiliates. The statements contained herein cannot be independently verified and are subject to change.

Session Outline

- Background
- A look at Microsoft documentation
- Vulnerability
- Conclusion & recommendations

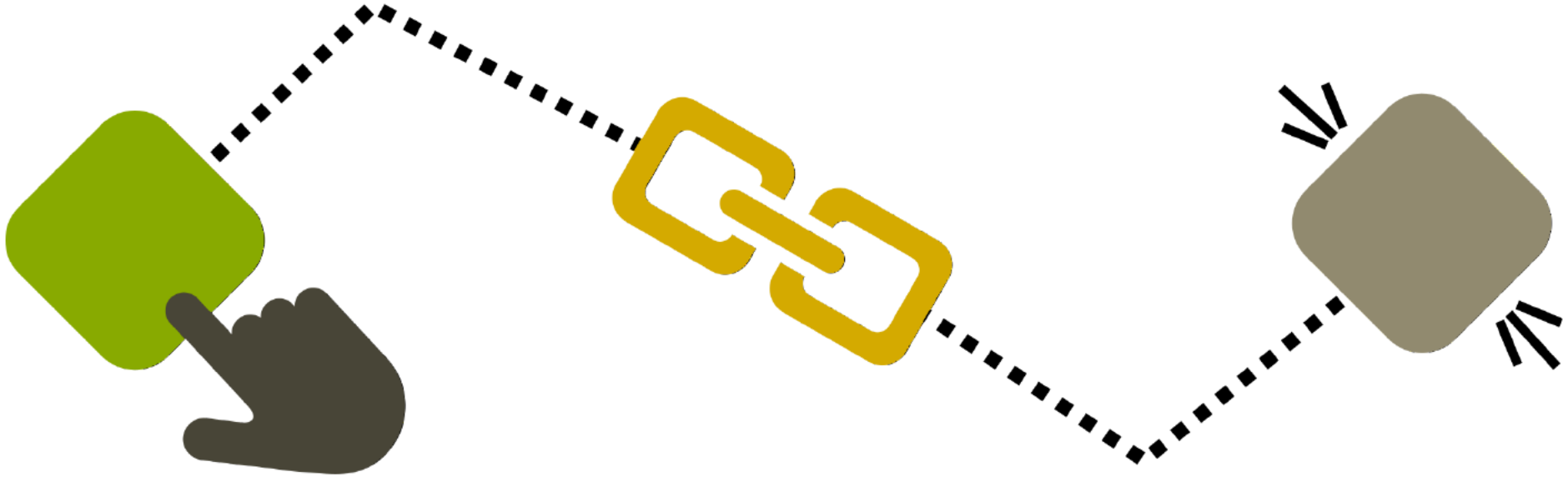
https://t.me/learningnets

Background

Links in Windows, with a focus on registry links

https://t.me/learningnets

Symbolic Links



SYMBOLIC LINKS CAN POSE SECURITY RISKS

Have been exploited in various attacks, including privilege escalation, and denial of service. Attackers exploit these vulnerabilities by, for example, bypassing permissions or creating race conditions.

https://t.me/learningnets

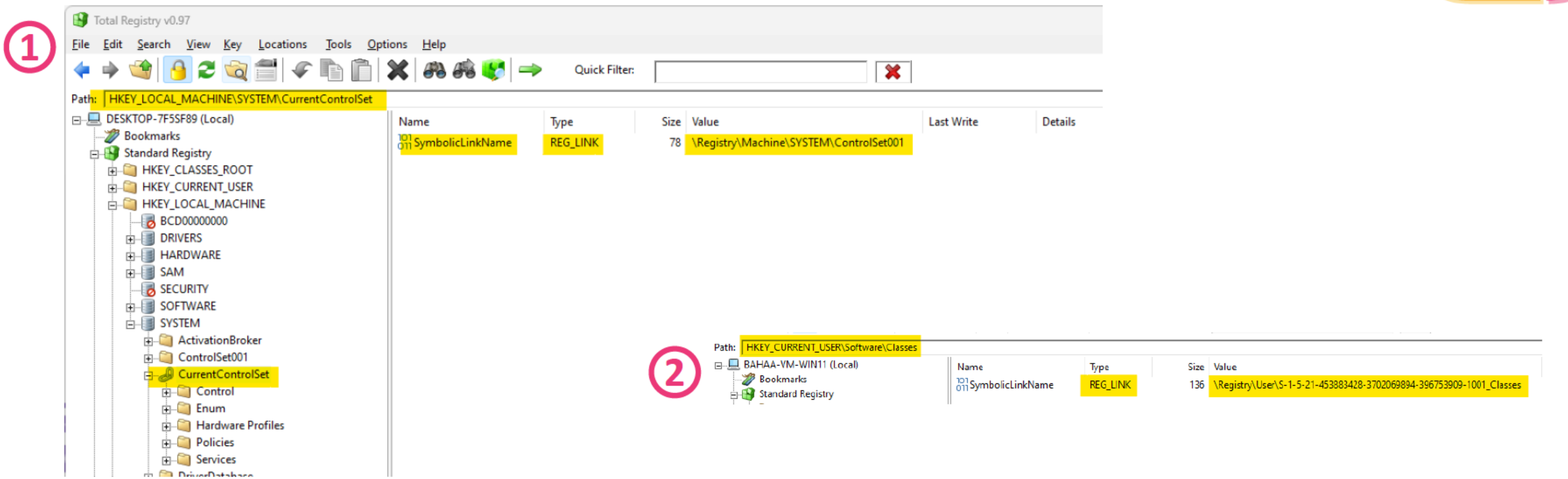
(Symbolic) Links in Windows

Some known types of links in Windows

Link Type	User Privilege	Delete Op	On Delete
NTFS Symbolic Links	Admin	DeleteFile	Link
NPFS Symbolic Links	Admin	DeviceIoControl	Link
NTFS Volume Mount Points	Standard	DeleteVolumeMountPoint	Link
Directory Junctions	Standard	RemoveDirectory	Link
Hard Links	Standard*	DeleteFile	Link
Object Manager Symbolic Links	Standard*	Zero handles	Link
Registry Symbolic Links	**Standard***	**RegDeleteKey[Ex]**	**Target**

* In some cases, admin privilege is required

Registry Symbolic Links



A FEATURE IN THE WINDOWS REGISTRY THAT ENABLES THE CREATION OF A LINK FROM A SOURCE KEY TO A TARGET KEY

Creating Registry Symlink

Win32 API	Native API
RegCreateKeyEx	Nt/ZwCreateKey

✚ REG_OPTION_CREATE_LINK

- Microsoft discourages developers from creating them
- Documentation is ambiguous and unclear (on purpose?)
  - Makes it difficult for developers to test against registry symlinks
- Links between keys in different registry hives are not allowed
  - For security reasons
  - But CreateKey APIs do not document this (issue #1)

Creating Registry Symlink: issue #2

RegCreateKeyEx

REG_OPTION_CREATE_LINK
0x00000002L

Note Registry symbolic links should only be used for for application compatibility when absolutely necessary.

This key is a symbolic link. The target path is assigned to the L"SymbolicLinkValue" value of the key. The target path must be an absolute registry path.

"absolute registry path" – what does that mean?

- **HKLM\...** - ✗
- **HKEY_LOCAL_MACHINE\...** - ✗
- **\REGISTRY\MACHINE\...** - ✓

Creating Registry Symlink: issue #3

`REG_LINK`	A null-terminated Unicode string that contains the target path of a symbolic link that was created by calling the RegCreateKeyEx function with **REG_OPTION_CREATE_LINK**.

VS.

Symbolic link source keys are created when a client creates a registry key with the registry option REG_OPTION_CREATE_LINK. After creating the symbolic link source key, a client MUST create a new value under the source key named "SymbolicLinkValue". The SymbolicLinkValue value contains the Object Name of the target of the symbolic link, which MUST NOT be NULL-terminated. The type of the value named SymbolicLinkValue MUST be REG_LINK.

[MS-RRP] Technical Documentation for Windows Remote Registry Protocol

Opening Registry Symlink

	API	W/o REG_OPTION_OPEN_LINK	W/ REG_OPTION_OPEN_LINK (N/A for *Create*)
Win32	RegOpenKeyEx	Target	Link
	RegCreateKeyEx	Target	Target
Native	Nt/ZwOpenKeyEx	Target	Link
	Nt/ZwCreateKey	Target	Target

→ Unexpected behavior: while CreateKey APIs can create **or** open standard registry keys, they cannot open existing symbolic link keys

Deleting Registry Symlink

	API	W/o REG_OPTION_OPEN_LINK	W/ REG_OPTION_OPEN_LINK
Win32	RegDeleteKey[Ex] RegDeleteTree	Target	Target
Native	Nt/ZwDeleteKey	Target	Link

→ Does not behave in a similar manner to other types of links in Windows

Documentation: Delete APIs

Delete registry keys vs. other objects in Windows

https://t.me/learningnets

Documentation of Links Deletion

DeleteFile

> Symbolic link behavior—
>
> If the path points to a symbolic link, the symbolic link is deleted, not the target. To delete a target, you must call CreateFile and specify **FILE_FLAG_DELETE_ON_CLOSE**.

> To delete a hard link, use the **DeleteFile** function. You can delete hard links in any order regardless of the order in which they are created.

> **RemoveDirectory** removes a directory junction, even if the contents of the target are not empty; the function removes directory junctions regardless of the state of the target object. For more information on junctions, see Hard Links and Junctions.

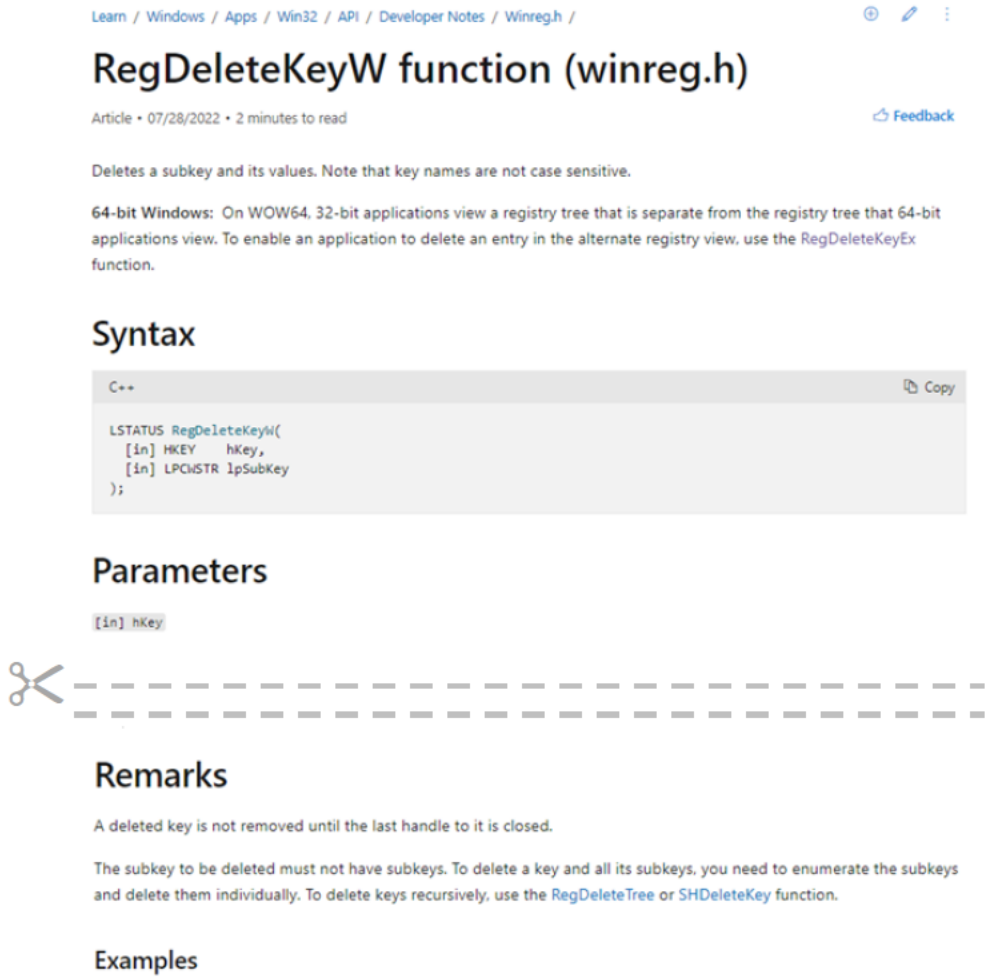
DeleteVolumeMountPoint

> Deleting a mounted folder does not cause the underlying directory to be deleted.

> A temporary object has a name only as long as its handle count is greater than zero. When the handle count reaches zero, the system deletes the object name and appropriately adjusts the object's pointer count.
>
> Removes the specified definition for the specified device.

Object Manager Symlink

RegDeleteKey[Ex] / RegDeleteTree



Learn / Windows / Apps / Win32 / API / Developer Notes / Winreg.h /

RegDeleteKeyW function (winreg.h)

Article • 07/28/2022 • 2 minutes to read

Feedback

Deletes a subkey and its values. Note that key names are not case sensitive.

64-bit Windows: On WOW64, 32-bit applications view a registry tree that is separate from the registry tree that 64-bit applications view. To enable an application to delete an entry in the alternate registry view, use the RegDeleteKeyEx function.

Syntax

C++ Copy

```
LSTATUS RegDeleteKeyW(
  [in] HKEY    hKey,
  [in] LPCWSTR lpSubKey
);
```

Parameters

`[in] hKey`

Remarks

A deleted key is not removed until the last handle to it is closed.

The subkey to be deleted must not have subkeys. To delete a key and all its subkeys, you need to enumerate the subkeys and delete them individually. To delete keys recursively, use the RegDeleteTree or SHDeleteKey function.

Examples

Vulnerability

Understanding the security risks and attack vectors

https://t.me/learningnets

Vulnerability: WYSINWYD

- What You See Is Not What You Delete!
- A non-privileged attacker can trick a privileged process to follow a registry symbolic link and delete a restricted/protected registry key
- Privileged AV/EDR products are more susceptible to this issue, as they typically attempt to remediate and remove threats that are detected including unwanted (malicious) registry keys

An Industry Failure?

#	Vendor Name	Vulnerable?	Patched?	CVE?
1	Avast	✓	✓	CVE-2022-4294
2	AVG	✓	✓	CVE-2022-4294
3	Avira	✓	✓	CVE-2022-4294
4	Bitdefender	✓	✓	CVE-2022-3369
5	Checkpoint	✓	✓	×
6	CrowdStrike	✓	✓	×
7	ESET	×	N/A	N/A
8	Kaspersky	✓	✓	×
9	Malwarebytes	✓	✓	×
10	McAfee	✓	✓	CVE-2023-24577
11	Microsoft	×	N/A	N/A
12	Norton	✓	✓	CVE-2022-4294
13	Sentinel One	✓	✓	×
14	Sophos	✓	✓	×
15	Symantec	✓	✓	CVE-2022-37016
16	Trellix	✓	✓	×
17	Trend Micro	✓	✓	×
18	Webroot	✓	✓	×

- 16 out of 18 tested vendors were vulnerable
- The list Includes top AV/EDR vendors
- Not all vendors have adequate security advisory/CVE publishing processes
  - Traditional AV vendors were better than EDR vendors in this regard

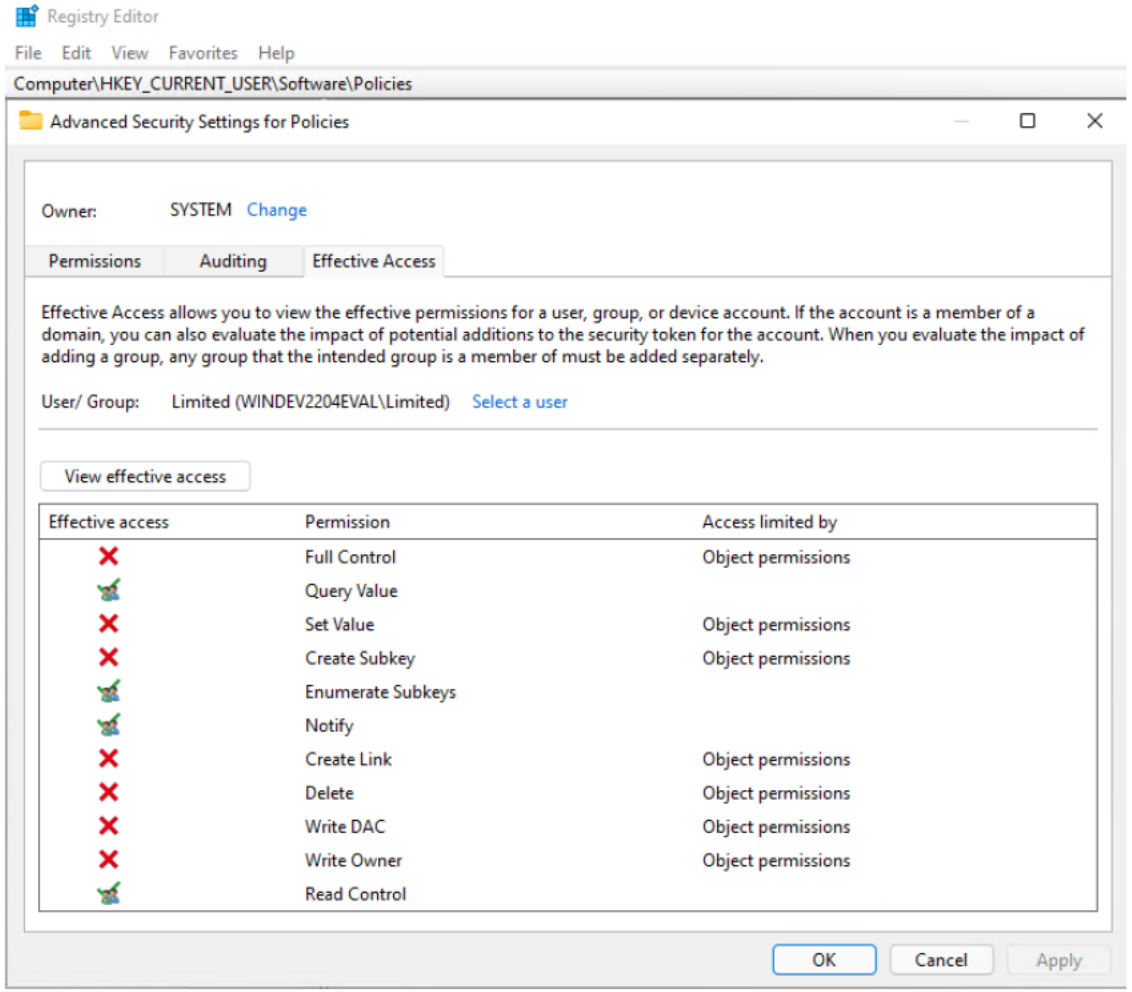
https://t.me/learningnets

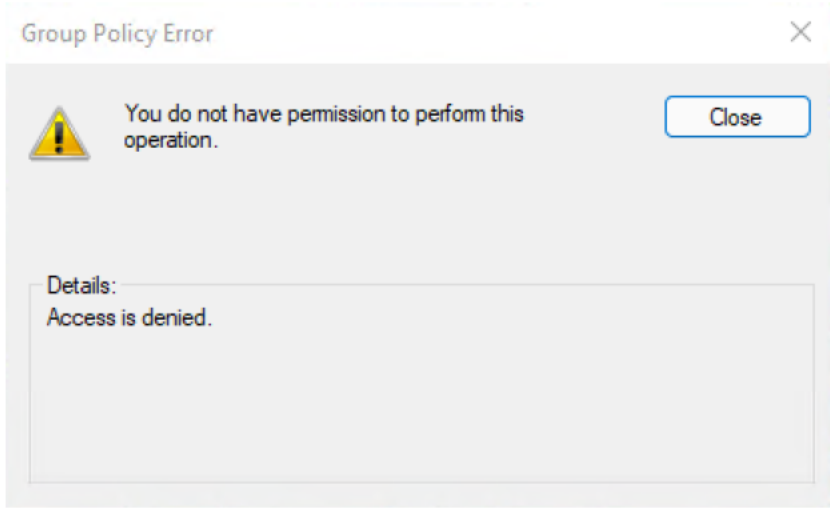
Read-Only Registry Keys


```
C:\demo>accesschk.exe -krs Limited HKCU | findstr /b /c:"R "
R  HKCU\Software\Google\Chrome\NativeMessagingHosts\com.microsoft.browsercore
R  HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy
R  HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\GroupMembership
R  HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\History
R  HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\PolicyApplicationState
R  HKCU\Software\Microsoft\Windows\CurrentVersion\Policies
R  HKCU\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Applications\Windows.SystemToast.CloudExperienceHostLauncherCustom
R  HKCU\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Applications\Windows.SystemToast.DisplaySettings
R  HKCU\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Applications\Windows.SystemToast.FodHelper
R  HKCU\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Applications\Windows.SystemToast.MobilityExperience
R  HKCU\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Applications\Windows.SystemToast.Suggested
R  HKCU\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Applications\Windows.SystemToast.WindowsTip
R  HKCU\Software\Policies
R  HKCU\Software\Policies\Microsoft
R  HKCU\Software\Policies\Microsoft\Windows
R  HKCU\Software\Policies\Microsoft\Windows\CloudContent
R  HKCU\Software\Policies\Microsoft\Windows\CurrentVersion
R  HKCU\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings
```

REGISTRY KEYS WITH RESTRICTIVE ACCESS TO STANDARD USERS WITHIN THE HKCU HIVE

Risk #1: Group Policies





gpedit.msc

Risk #1: Some Interesting Policies

HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer

Value name	Value data types	Description
AlwaysInstallElevated	REG_DWORD	If this value is set to "1" and the corresponding computer value is also set, the installer always installs with elevated privileges. Otherwise, the installer uses elevated privileges to install managed applications and uses the current user's privilege level for nonmanaged applications.

- Requires that the same registry key is also set in HKEY_LOCAL_MACHINE

Risk #1: Some Interesting Policies

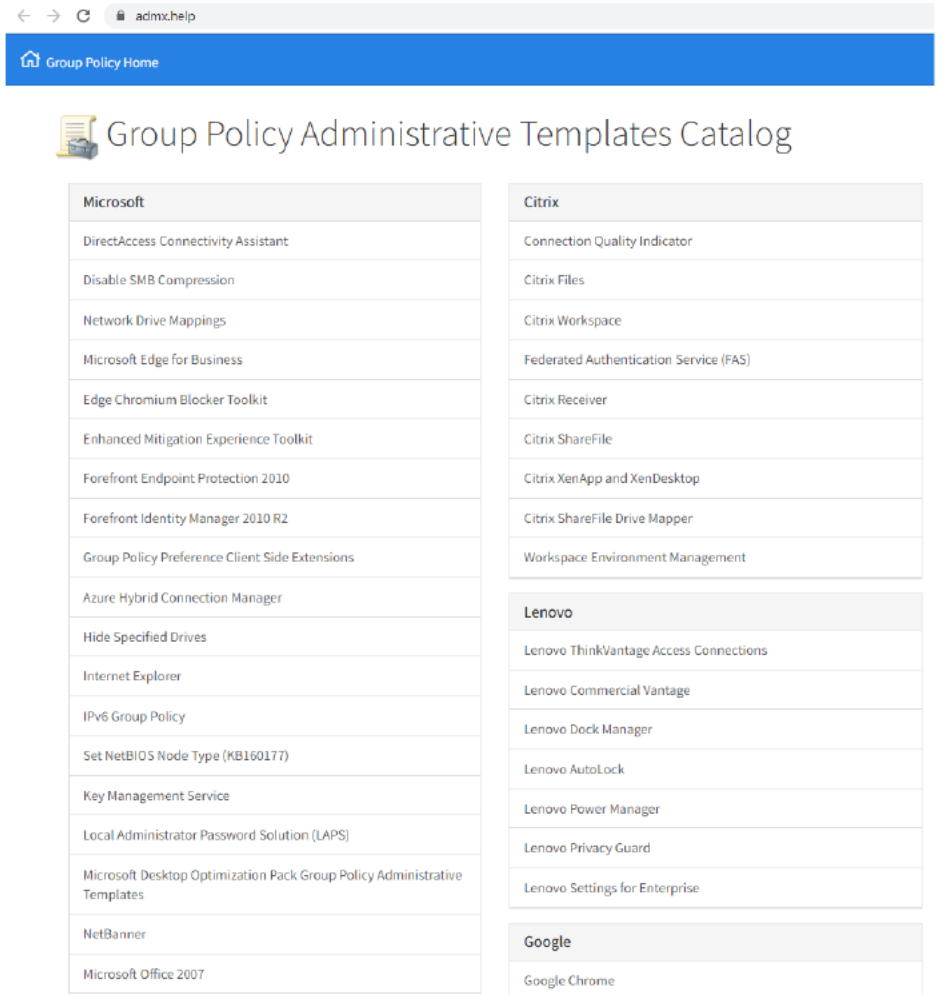
- HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows NT\MitigationOptions**ProcessMitigationOptions**
  - Disable
    - Data Execution Prevention (DEP)
    - Structured Exception Handling Overwrite Protection (SEHOP)
    - Address Space Layout Randomization (ASLR)

https://t.me/learningnets

Risk #1: Thousands of Group Policies

Microsoft	RSA
Citrix	Dell
Lenovo	KnowBe4
Google	Oracle
VMware	SAP
Adobe	RealVNC
Duo Security	HP
Mozilla	Symantec

And many more…

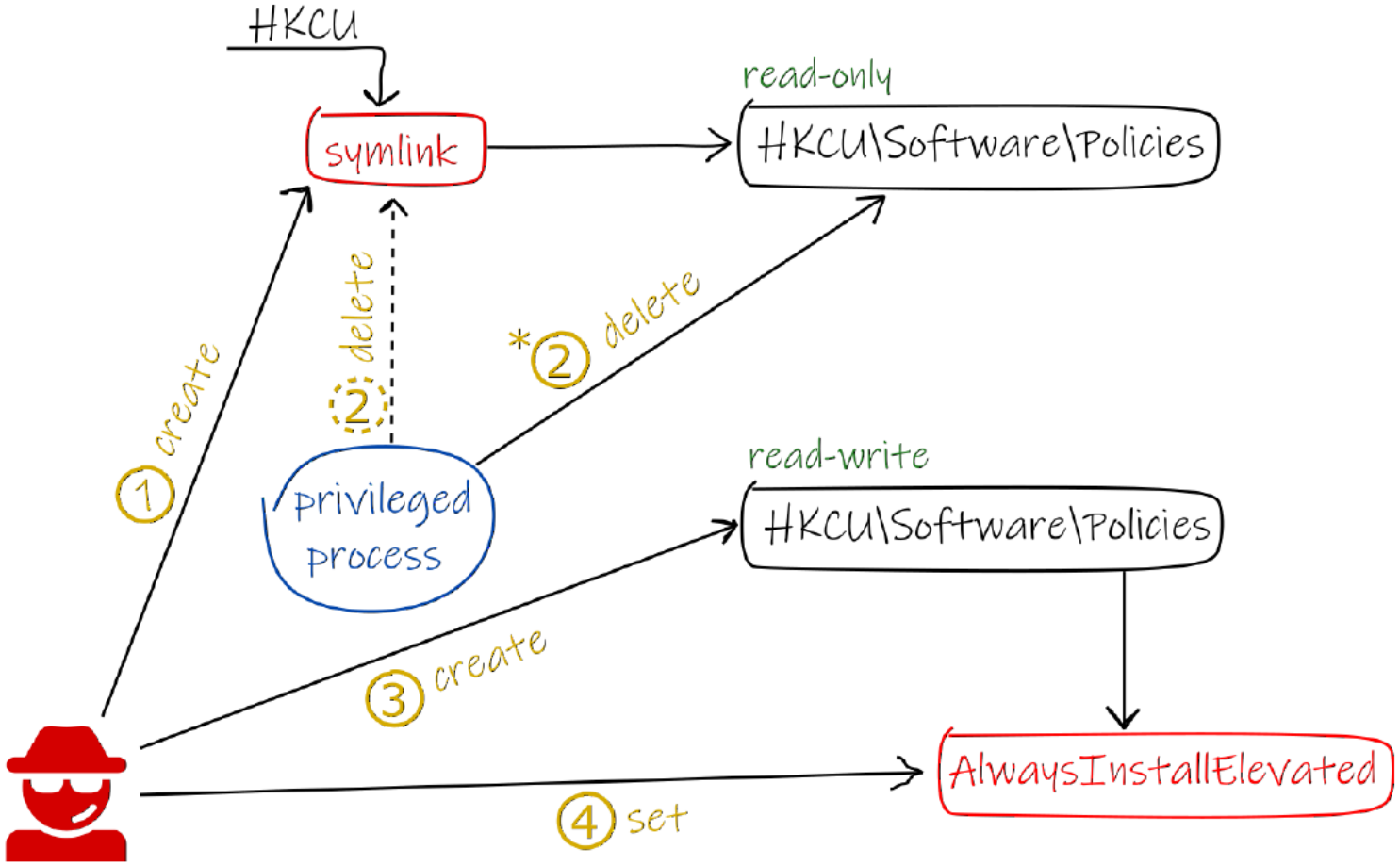


Risk #1: Attack Vector

1. Attacker creates a registry symlink to **HKCU\Software\Policies** in a location where a privileged process deletes it
2. Triggers/waits for the privileged process to delete the symbolic link registry key

   → The target key **HKCU\Software\Policies** will be deleted instead
3. Attacker creates **HKCU\Software\Policies** with permissive write permissions (in most cases, the first process that accesses this key will inadvertently create it with more permissive write access)
4. Attacker can now set any local group policies they wish, including **AlwaysInstallElevated**

Risk #1: Attack Vector (Diagram)



Risk #1: DEMO

DEMO

https://t.me/learningnets

Risk #2: Attack Vector

1. AV/EDR products typically implement anti-tamper protection
2. Attacker creates a registry symlink to point to the AV/EDR system service registry key:

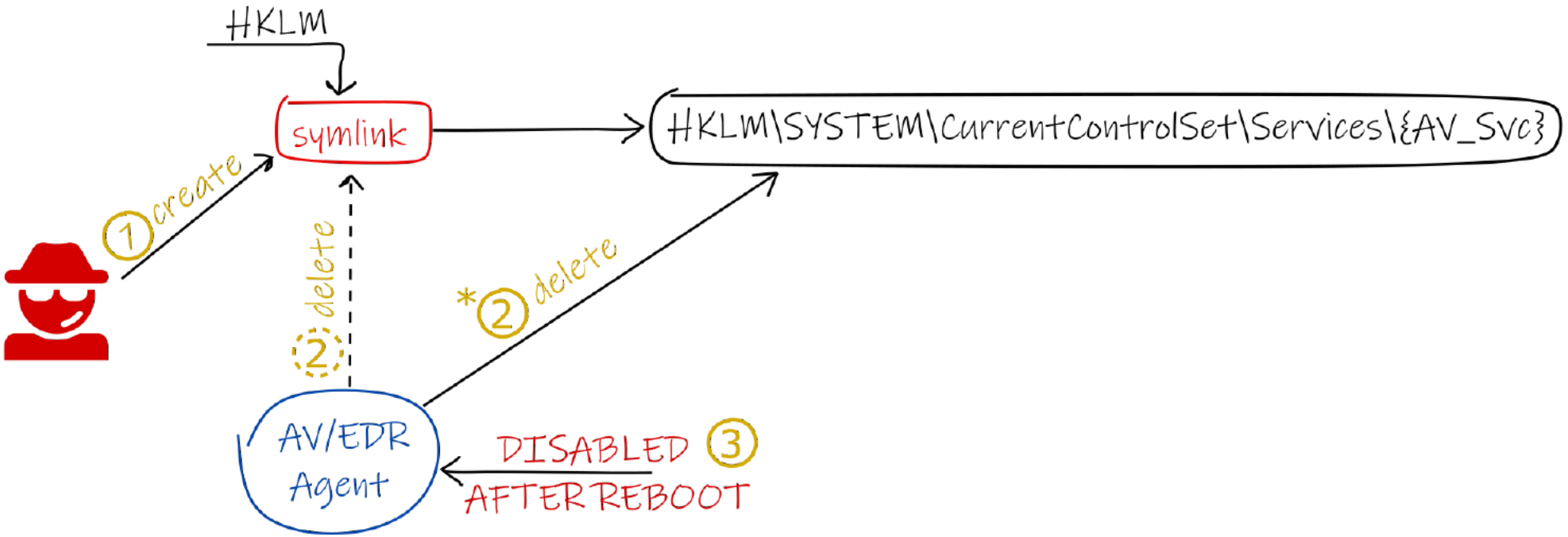
   HKLM\PATH\SymLink → **HKLM\SYSTEM\CurrentControlSet\Services\{AV_Svc}**

3. Triggers scan/waits for the privileged process (AV/EDR) to delete the symbolic link key

→ AV/EDR deletes its own **HKLM\SYSTEM\CurrentControlSet\Services\{AV_Svc}**

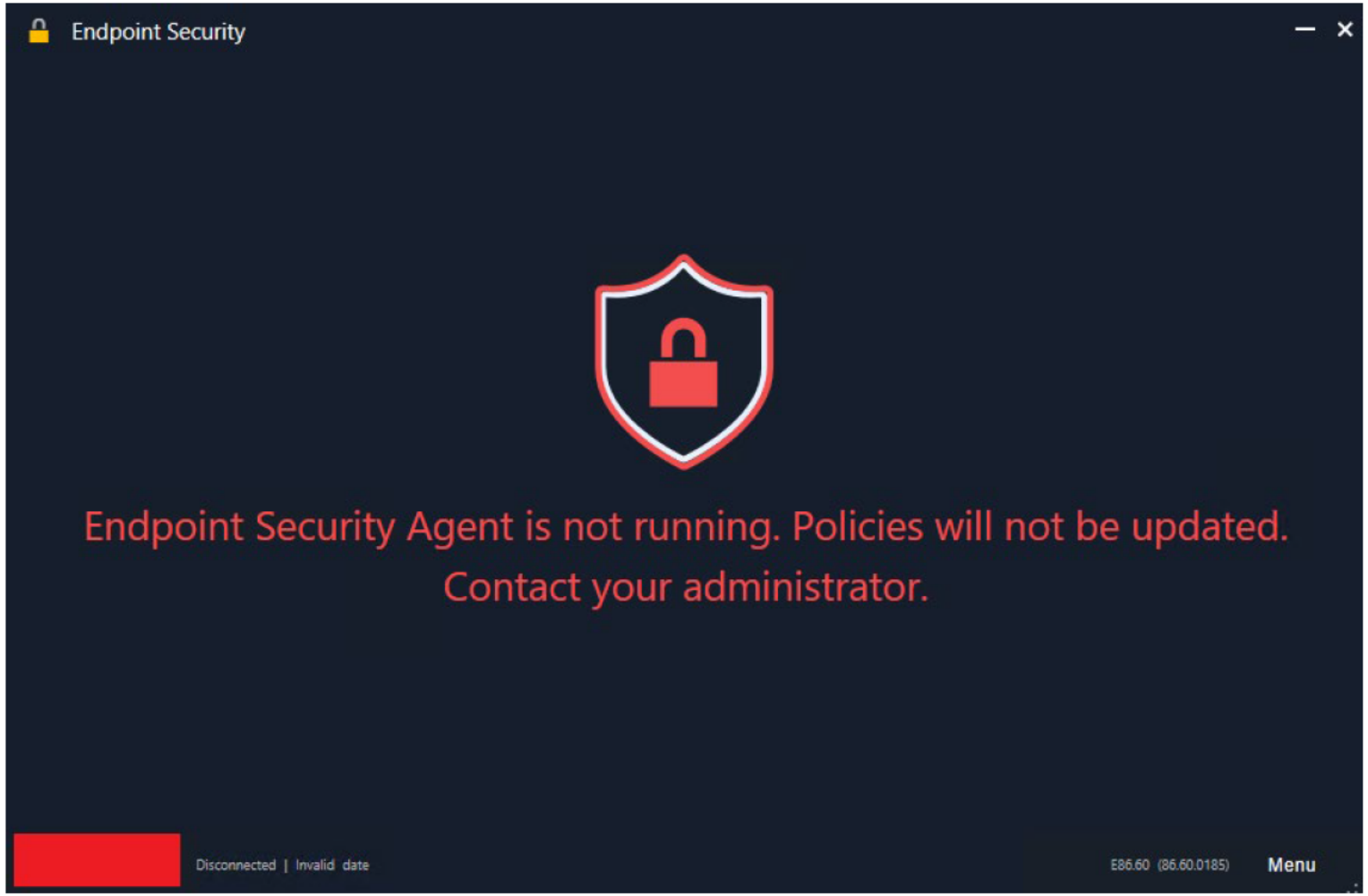
4. On reboot the AV/EDR will be disabled

Risk #2: Attack Vector (Diagram)



AV/EDR PRODUCTS TYPICALLY IMPLEMENT ANTI-TAMPER PROTECTION

Risk #2: EDR Agent is Disabled After Reboot



Security Impact

- Standard (non-admin) users can configure group policies that would otherwise be inaccessible to them
- Local privilege escalation to run code as SYSTEM
- Disable AV/EDR

Conclusion & Best Practices

Key takeaways and recommendation for future actions

https://t.me/learningnets

Who is to Blame?

BUG

A security vulnerability was inevitable, leading 16 out of 18 tested vendors to make the same mistake

Lack of Testing

Software vendors could have exercised greater care in writing and testing their software

Inadequate Documentation

It didn't warn about the security risks of deleting registry key symbolic links

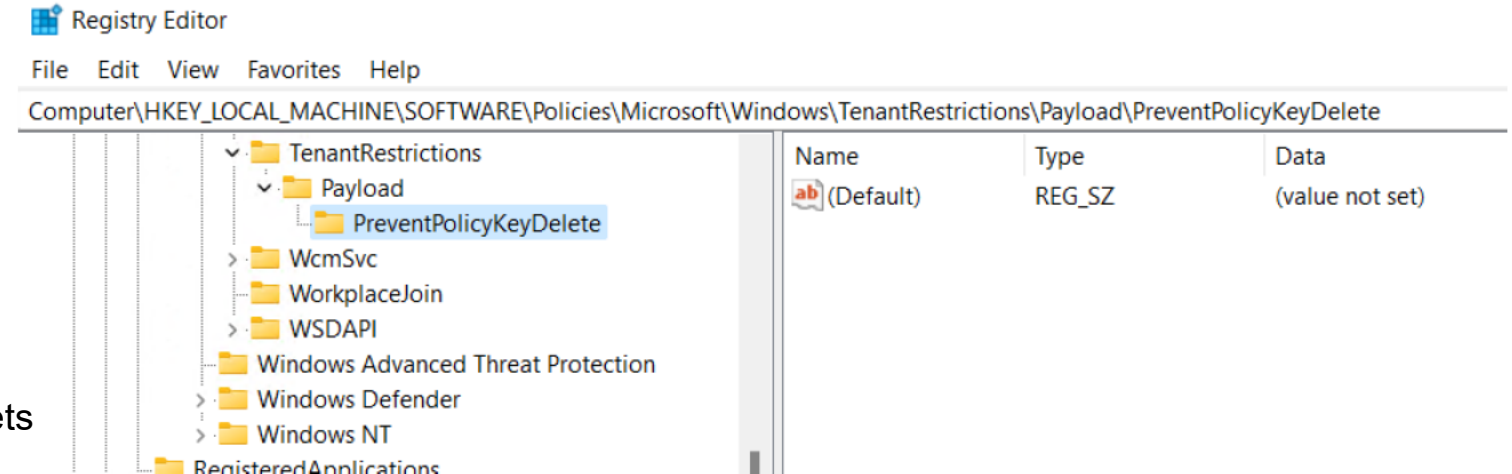
Unexpected Behavior

APIs behavior is unexpected for developers – not documented

https://t.me/learningnets

Mitigation / Microsoft

- Update the documentation of APIs that are used to delete registry keys to include the likely-to-be-overlooked risk of deleting registry symlinks
- Consider changing the default behavior of these APIs when deleting registry symbolic links
- Prevent deleting **HKEY_CURRENT_USER\Software\Policies**



Apply What You Have Learned Today: Defenders

- Create a rule to check if "**HKEY_CURRENT_USER\Software\Policies**" get deleted
- As TrustedInstaller, create a subkey under "**HKCU\Software\Policies**" with DELETE permission removed for all other user accounts
- Needless to say that "**AlwaysInstallElevated**" should never ever be used
- Maintain an overview of offline AV/EDR agents on active endpoints

Apply What You Have Learned Today: Vendors

- Review your code to ensure it does not allow deleting or writing to privileged registry keys via registry symlink manipulation
- Always use user impersonation when accessing unprotected registry keys or objects in general from privileged processes
- Set the correct ACL on your created registry keys

Summary

- Registry links, like other symbolic links, can pose potential security risks, underscoring the need for cautious access and appropriate security measures to mitigate these risks

API documentation should include potential security risks to promote secure coding practices and reduce the likelihood of security incidents

https://t.me/learningnets

RSAConference™2023

Thank You!

Q&A

https://www.linkedin.com/in/bahaanaamneh

https://t.me/learningnets