gtworek/Priv2Admin

github.com/gtworek/Priv2Admin

gtworek

The idea is to "translate" Windows OS privileges to a path leading to:

1. administrator,
2. integrity and/or confidentiality threat,
3. availability threat,
4. just a mess.

Privileges are listed and explained at: https://learn.microsoft.com/en-us/windows/win32/secauthz/privilege-constants

If the goal can be achieved multiple ways, the priority is

1. Using built-in commands
2. Using PowerShell (only if a working script exists)
3. Using non-OS tools
4. Using any other method

You can check your own privileges with `whoami /priv`. Disabled privileges are as good as enabled ones. The only important thing is if you have the privilege on the list or not.

Note 1: Whenever the attack path ends with a token creation, you can assume the next step is to create new process using such token and then take control over OS.

Note 2:
a. For calling `NtQuerySystemInformation()`/`ZwQuerySystemInformation()` directly, you can find required privileges here.
b. For `NtSetSystemInformation()`/`ZwSetSystemInformation()` required privileges are listed here here.

Note 3: I am focusing on the OS only. If a privilege works in AD but not in the OS itself, I am describing it as not used in the OS. It would be nice if someone digs deeper into AD-oriented scenarios.

Feel free to contribute and/or discuss presented ideas.

Privilege	Impact	Tool	Execution path	Remarks
`SeAssignPrimaryToken`	***Admin***	3rd party tool	*"It would allow a user to impersonate tokens and privesc to nt system using tools such as potato.exe, rottenpotato.exe and juicypotato.exe"*	Thank you Aurélien Chalot for the update. I will try to re-phrase it to something more recipe-like soon.
`SeAudit`	**Threat**	3rd party tool	Write events to the Security event log to fool auditing or to overwrite old events.	Writing own events is possible with `Authz Report Security Event` API. - see PoC by @daem0nc0re
`SeBackup`	***Admin***	3rd party tool	1. Backup the `HKLM\SAM` and `HKLM\SYSTEM` registry hives 2. Extract the local accounts hashes from the `SAM` database 3. Pass-the-Hash as a member of the local `Administrators` group Alternatively, can be used to read sensitive files.	For more information, refer to the `SeBackupPrivilege` file. - see PoC by @daem0nc0re

Privilege	Impact	Tool	Execution path	Remarks
`SeChangeNotify`	None	-	-	Privilege held by everyone. Revoking it may make the OS (Windows Server 2019) unbootable.
`SeCreateGlobal`	?	?	?	
`SeCreatePagefile`	None	***Built-in commands***	Create hiberfil.sys, read it offline, look for sensitive data.	Requires offline access, which leads to admin rights anyway. - See PoC by @daem0nc0re
`SeCreatePermanent`	?	?	?	
`SeCreateSymbolicLink`	?	?	?	
`SeCreateToken`	***Admin***	3rd party tool	Create arbitrary token including local admin rights with `NtCreateToken`. - see PoC by @daem0nc0re	
`SeDebug`	***Admin***	**PowerShell**	Duplicate the `lsass.exe` token.	Script to be found at FuzzySecurity. - See PoC by @daem0nc0re
`SeDelegateSession-UserImpersonate`	?	?	?	Privilege name broken to make the column narrow.
`SeEnableDelegation`	None	-	-	The privilege is not used in the Windows OS.
`SeImpersonate`	***Admin***	3rd party tool	Tools from the *Potato family* (potato.exe, RottenPotato, RottenPotatoNG, Juicy Potato, SweetPotato, RemotePotato0), RogueWinRM, PrintSpoofer, etc.	Similarly to `SeAssignPrimaryToken`, allows by design to create a process under the security context of another user (using a handle to a token of said user). Multiple tools and techniques may be used to obtain the required token.
`SeIncreaseBasePriority`	Availability	***Built-in commands***	`start /realtime SomeCpuIntensiveApp.exe`	May be more interesting on servers.
`SeIncreaseQuota`	Availability	3rd party tool	Change cpu, memory, and cache limits to some values making the OS unbootable.	- Quotas are not checked in the safe mode, which makes repair relatively easy. - The same privilege is used for managing registry quotas.
`SeIncreaseWorkingSet`	None	-	-	Privilege held by everyone. Checked when calling fine-tuning memory management functions.

Privilege	Impact	Tool	Execution path	Remarks
`SeLoadDriver`	***Admin***	3rd party tool	1. Load buggy kernel driver such as `szkg64.sys` 2. Exploit the driver vulnerability Alternatively, the privilege may be used to unload security-related drivers with `fltMC` builtin command. i.e.: `fltMC sysmondrv`	1. The `szkg64` vulnerability is listed as CVE-2018-15732 2. The `szkg64` exploit code was created by Parvez Anwar
`SeLockMemory`	Availability	3rd party tool	Starve System memory partition by moving pages.	PoC published by Walied Assar (@waleedassar)
`SeMachineAccount`	None	-	-	The privilege is not used in the Windows OS.
`SeManageVolume`	***Admin***	3rd party tool	1. Enable the privilege in the token 2. Create handle to \.\C: with `SYNCHRONIZE | FILE_TRAVERSE` 3. Send the `FSCTL_SD_GLOBAL_CHANGE` to replace `S-1-5-32-544` with `S-1-5-32-545` 4. Overwrite utilman.exe etc.	`FSCTL_SD_GLOBAL_CHANGE` can be made with this piece of code.
`SeProfileSingleProcess`	None	-	-	The privilege is checked before changing (and in very limited set of commands, before querying) parameters of Prefetch, SuperFetch, and ReadyBoost. The impact may be adjusted, as the real effect is not known.
`SeRelabel`	**Threat**	3rd party tool	Modification of system files by a legitimate administrator	See: MIC documentation Integrity labels provide additional protection, on top of well-known ACLs. Two main scenarios include: - protection against attacks using exploitable applications such as browsers, PDF readers etc. - protection of OS files. `SeRelabel` present in the token will allow to use `WRITE_OWNER` access to a resource, including files and folders. Unfortunately, the token with IL less than *High* will have SeRelabel privilege disabled, making it useless for anyone not being an admin already. See great blog post by @tiraniddo for details.

Privilege	Impact	Tool	Execution path	Remarks
`SeRemoteShutdown`	Availability	***Built-in commands***	`shutdown /s /f /m \\server1 /d P:5:19`	The privilege is verified when shutdown/restart request comes from the network. 127.0.0.1 scenario to be investigated.
`SeReserveProcessor`	None	-	-	It looks like the privilege is no longer used and it appeared only in a couple of versions of winnt.h. You can see it listed i.e. in the source code published by Microsoft here.
`SeRestore`	***Admin***	**PowerShell**	1. Launch PowerShell/ISE with the SeRestore privilege present. 2. Enable the privilege with Enable-SeRestorePrivilege). 3. Rename utilman.exe to utilman.old 4. Rename cmd.exe to utilman.exe 5. Lock the console and press Win+U	Attack may be detected by some AV software. Alternative method relies on replacing service binaries stored in "Program Files" using the same privilege. - see PoC by @daem0nc0re
`SeSecurity`	**Threat**	***Built-in commands***	- Clear Security event log: `wevtutil cl Security` - Shrink the Security log to 20MB to make events flushed soon: `wevtutil sl Security /ms:0` - Read Security event log to have knowledge about processes, access and actions of other users within the system. - Knowing what is logged to act under the radar. - Knowing what is logged to generate large number of events effectively purging old ones without leaving obvious evidence of cleaning. - Viewing and changing object SACLs (in practice: auditing settings)	See PoC by @daem0nc0re
`SeShutdown`	Availability	***Built-in commands***	`shutdown.exe /s /f /t 1`	Allows to call most of NtPowerInformation() levels. To be investigated. Allows to call NtRaiseHardError() causing immediate BSOD and memory dump, leading potentially to sensitive information disclosure - see PoC by @daem0nc0re
`SeSyncAgent`	None	-	-	The privilege is not used in the Windows OS.

Privilege	Impact	Tool	Execution path	Remarks
`SeSystemEnvironment`	*Unknown*	3rd party tool	The privilege permits to use `NtSetSystemEnvironmentValue`, `NtModifyDriverEntry` and some other syscalls to manipulate UEFI variables.	The privilege is required to run sysprep.exe. Additionally: - Firmware environment variables were commonly used on non-Intel platforms in the past, and now slowly return to UEFI world. - The area is highly undocumented. - The potential may be huge (i.e. breaking Secure Boot) but raising the impact level requires at least PoC. - see PoC by @daem0nc0re
`SeSystemProfile`	?	?	?	
`SeSystemtime`	**Threat**	***Built-in commands***	`cmd.exe /c date 01-01-01` `cmd.exe /c time 00:00`	The privilege allows to change the system time, potentially leading to audit trail integrity issues, as events will be stored with wrong date/time. - Be careful with date/time formats. Use always-safe values if not sure. - Sometimes the name of the privilege uses uppercase "T" and is referred as `SeSystemTime`.
`SeTakeOwnership`	***Admin***	***Built-in commands***	1. `takeown.exe /f "%windir%\system32"` 2. `icacls.exe "%windir%\system32" /grant "%username%":F` 3. Rename cmd.exe to utilman.exe 4. Lock the console and press Win+U	Attack may be detected by some AV software. Alternative method relies on replacing service binaries stored in "Program Files" using the same privilege. - See PoC by @daem0nc0re
`SeTcb`	***Admin***	3rd party tool	Manipulate tokens to have local admin rights included.	Sample code+exe creating arbitrary tokens to be found at PsBits.
`SeTimeZone`	Mess	***Built-in commands***	Change the timezone. `tzutil /s "Chatham Islands Standard Time"`	
`SeTrustedCredManAccess`	**Threat**	3rd party tool	Dumping credentials from Credential Manager	Great blog post by @tiraniddo. - see PoC by @daem0nc0re

Privilege	Impact	Tool	Execution path	Remarks
SeUndock	None	-	-	The privilege is enabled when undocking, but never observed it checked to grant/deny access. In practice it means it is actually unused and cannot lead to any escalation.
SeUnsolicitedInput	None	-	-	The privilege is not used in the Windows OS.

Credits: