

The Windows Security Journey

Version 8.0
November-2025

By Dr. Shlomi Boutnaru



Table of Contents

Table of Contents.....	2
Introduction.....	13
SID (Security Identifier).....	14
SD (Security Descriptor).....	15
Securable Objects.....	16
Privileges.....	17
SAM (Security Account Manager).....	18
Access Token.....	19
Primary Access Token.....	20
Impersonation Access Token.....	21
SRM (Security Reference Monitor).....	22
Job Object.....	23
ACL (Access Control List).....	24
DACL (Discretionary Access Control List).....	25
SACL (System Access Control List).....	26
Mandatory Integrity Control (MIC).....	26
UAC (User Account Control).....	28
User Interface Privilege Isolation (UIPI).....	29
File Virtualization.....	30
Registry Virtualization.....	31
Share Permissions (Network Shares).....	32
SysMon (System Monitor).....	33
ELAM (Early Launch AntiMalware).....	34
Local System (NT AUTHORITY\SYSTEM).....	35
Network Service (NT AUTHORITY\NETWORK SERVICE).....	36
Local Service (NT AUTHORITY\LOCAL SERVICE).....	37
Per-User Windows Services.....	38
Windows Defender Firewall.....	39
TrustedInstaller.....	40
Windows Sessions.....	41
Secure Desktop.....	42
MSRC (Microsoft Security Response Center).....	43
MSTIC (Microsoft Threat Intelligence Center).....	44
PEL (Protected Event Logging).....	45
DSE (Driver Signature Enforcement).....	46
EMET (Enhanced Mitigation Experience Toolkit).....	47
Windows Defender Antivirus.....	48
Windows Defender SmartScreen.....	49
MSDE (Microsoft Defender for Endpoint).....	50

SRP (Software Restriction Policies).....	51
AppLocker (Application Locking).....	52
AppIDSvc (Application Identity Service).....	53
Differences between AppLocker and SRP.....	54
Microsoft ISG (Microsoft Intelligent Security Graph).....	55
WDAC (Windows Defender Application Control).....	56
Differences between WDAC and AppLocker.....	57
AMSI (Anti-Malware Scan Interface).....	58
SGRM (System Guard Runtime Monitor).....	59
WFP (Windows Filtering Platform).....	60
Windows Workgroup.....	61
Windows Homegroup.....	62
AD (Active Directory).....	63
RODC (Read Only Domain Controller).....	64
GPO (Group Policy Object).....	65
AAD (Azure Active Directory).....	66
Microsoft Entra ID.....	67
HGS (Host Guardian Service).....	68
Guarded Host.....	69
Microsoft Hyper-V Shielded VM.....	70
Null Page Mitigation.....	71
UMFD (User Mode Font Driver).....	72
WDEG (Windows Defender Exploit Guard).....	73
ASR (Attack Surface Reduction).....	74
ACG (Arbitrary Code Guard).....	75
CIG (Code Integrity Guard).....	76
Block Low Integrity Images.....	77
Block Remote Images.....	78
Block Untrusted Fonts.....	79
IAF (Import Address Filtering).....	80
Controlled Folder Access.....	81
SAC (Smart App Control).....	82
No Child Process Mitigation.....	83
Windows Hello.....	84
DEP (Data Execution Prevention).....	85
SafeSEH (Safe Structured Exception Handling).....	86
SEHOP (Structured Exception Handling Overwrite Protection).....	87
NTFS (New Technology File System) Permissions.....	88
Null Sessions.....	89
Local User Account.....	90
Domain User Account.....	91

Microsoft Account (MSA).....	92
Local Group.....	93
WDigest (Windows Digest).....	94
Protected Processes.....	95
PPL (Protected Processes Light).....	96
Windows Authentication.....	97
LSA (Local Security Authority).....	98
LSA Protection (Local Security Authority Protection).....	99
RestrictedAdminMode for RDP (Remote Desktop Protocol Restricted Admin Mode).....	100
Protected View.....	101
Hotpatching.....	102
RunAs (Secondary Logon).....	103
Sudo for Windows.....	104
Sudo for Windows vs RunAs.....	105
OpenSSH for Windows.....	106
Microsoft Intune.....	107
Microsoft LAPS (Microsoft Local Administrator Password Solution).....	108
Windows LAPS (Windows Local Administrator Password Solution).....	109
Windows Defender Antivirus Emulator.....	110
PatchGuard (Kernel Patch Protection).....	111
LocalAccountTokenFilterPolicy (User Account Control Remote Restrictions).....	112
NLA (Network Level Authentication).....	113
GINA (Graphical Identification and Authentication).....	114
Credential Providers.....	115
Credential Providers Settings.....	116
SSPI (Security Support Provider Interface).....	117
SSP (Security Support Provider).....	118
EFS (Encrypted File System).....	119
Virtual User Accounts (Virtual User).....	120
DPAPI (Data Protection API).....	121
VBS (Virtualization Based Security).....	122
CG (Credentials Guard).....	123
HVCI (Hypervisor Protected Code Integrity).....	124
Managed Service Accounts (MSA).....	125
Standalone Managed Service Account (sMSA).....	126
Group Managed Service Account (gMSA).....	127
KDS Root Keys (Key Distribution Service Root Keys).....	128
MP (Memory Protector).....	129
MemGC (Memory Garbage Collection).....	130
"/GS" (Buffer Security Check).....	131
Azure Arc.....	132

Windows File Protection (WFP).....	134
Replacement of WFP (Windows File Protection) Protected Files.....	135
How Does Windows File Protection (WFP) Work?.....	136
WRP (Windows Resource Protection).....	137
Replacement of WRP (Windows Resource Protection) Protected Files.....	138

Introduction

When starting to learn OS security I believe that there is a need for understanding multiple technologies and concepts. Because of that I have decided to write a series of short writeups aimed at providing the security vocabulary.

Overall, I wanted to create something that will improve the overall knowledge of Windows' different security mechanisms included with Windows in writeups that can be read in 1-3 mins. I hope you are going to enjoy the ride.

Lastly, you can follow me on twitter - @boutnaru (<https://twitter.com/boutnaru>). Also, you can read my other writeups on medium - <https://medium.com/@boutnaru>. Lastly, You can find my free eBooks at <https://TheLearningJourneyEbooks.com>.

Lets GO!!!!!!

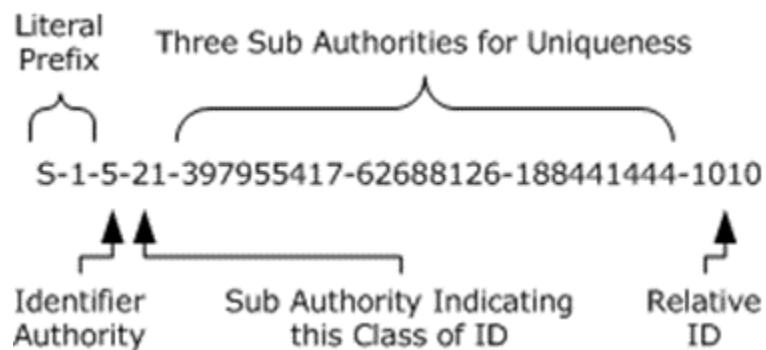
SID (Security Identifier)

The goal of an SID is to uniquely identify a security principal/group. When talking about a security principal we mean any entity that can authenticate to the operating system like: user/computer account or a thread/process that runs with the security context of one of those. Every time a user is logged on the system creates an access token for that user (more on that in a future writeup). This access token holds the user's SID, privileges and the SIDs for any groups that the user is part of¹.

Moreover, there is a specific format for an SID. We can split it into three main parts: revision, identifier authority and sub authorities. Revision, which specifies the version of the SID structure. Identifier authority, which specifies the highest level of authority that can issue an SID for a security principal. Sub authorities, which hold the most important information (can identify a local computer/domain) and its last part is an RID (relative identifier) that identifies a specific user/group in a local computer/domain - as shown in the diagram below².

An example of some well-known SIDs are: "S-1-1-0" (group that includes all users), "S-1-0-0" (aka NULL SID, a group with no members). They are called "Universal well-known SIDs"³. Also, there are well-known RIDs such as: 500 (Administrator), 501 (Guest). Since Windows 2008/Vista most of the system files are owned by the "TrustedInstaller" SID, in order to prevent a process running with Administrator/Local System permissions to overwrite the OS files⁴.

Lastly, there are also "Capability SIDs" which grant access to specific resources (like cameras, documents, location and more). Those type of SIDs that the system is aware of are stored in the registry under "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SecurityManager\CapabilityClasses"⁵.



¹ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>

² https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-azod/ecc7dfba-77e1-4e03-ab99-114b349c7164

³ <https://learn.microsoft.com/en-us/windows/win32/secauthz/well-known-sids>

⁴ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>

⁵ <https://renenyfenegger.ch/notes/Windows/security/SID/index>

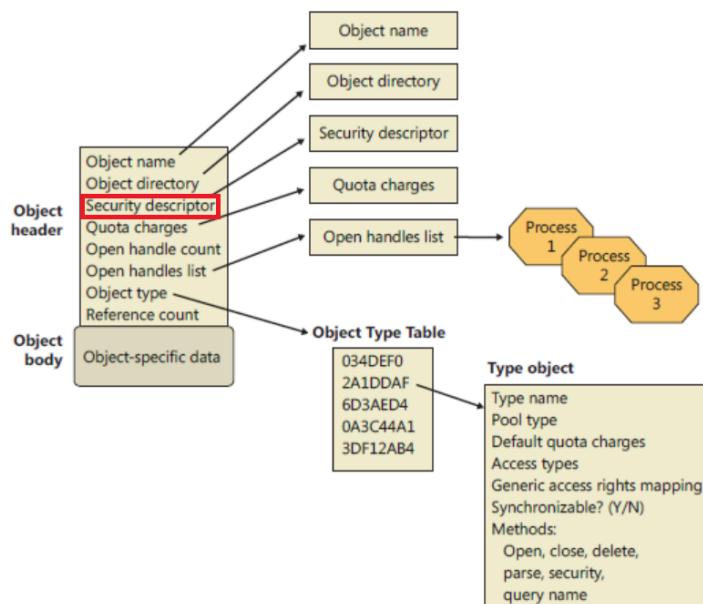
SD (Security Descriptor)

The goal of a security descriptor (SD) is to hold the security information that is related with a specific securable object. Examples for securable objects are: file, folder, network share, printer, registry key, synchronization object, active directory objects and more. The structure which describes a SD is defined in “`winnt.h`” and is named “`SECURITY_DESCRIPTOR`”⁶.

Overall, every object created by the “Object Manager” in Windows has a SD. Each objects has an header with different fields (like object name, reference count, object type and more) one of them is the security descriptor⁷. You can see an illustration of that in the diagram below⁸.

Moreover, we can think about an SD as containing four main fields: an owner, group, DACL (Discretionary Access Control List) and SACL (System Access Control List) . DACL is used for allowing/denying permissions which SACL is used for auditing⁹. The description of each entity in the structure is stored in the form of an SID (Security Identifier). More on those in future writeups.

Lastly, the “`SECURITY_DESCRIPTOR`” is a compact binary representation of the security associated with a specific object. Because it is not convenient to use it, there is a text-based form for representing it. This format is called SSDL (Security Descriptor Description Language). It has specific text tokens in order to describe: access rights, user accounts, user-mode drivers and more¹⁰.



⁶ https://learn.microsoft.com/en-us/windows/win32/api/winnt/ns-winnt-security_descriptor

⁷ https://www.geoffchappell.com/studies/windows/km/ntoskrnl/inc/ntos/ob/object_header/index.htm

⁸ https://www.tophertimzen.com/resources/cs407/slides/week02_01-KernelObjects.html#slide16

⁹ https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-azod/ec52bde3-9c86-4484-9080-e72148a2d53b

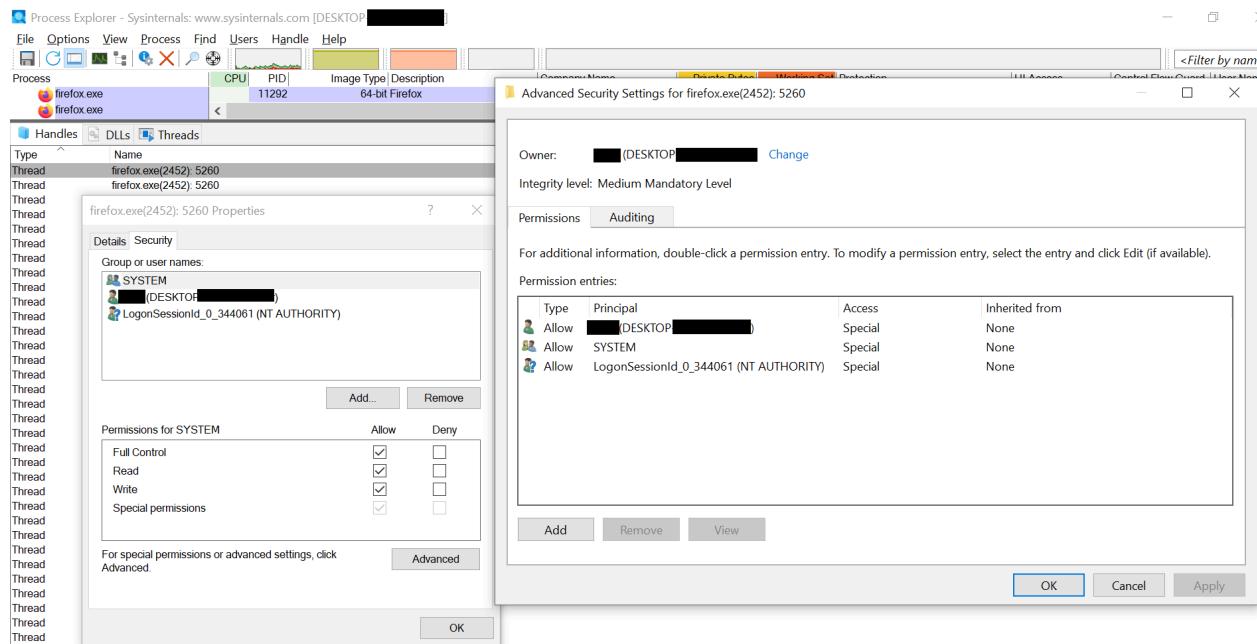
¹⁰ <https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/sddl-for-device-objects?redirectedfrom=MSDN>

Securable Objects

Overall, “securable objects” are Windows objects¹¹ that can have a “security descriptor”¹². All named Windows objects are securable. There are also unnamed objects which are securable like processes and threads¹³ - as shown in the screenshot below.

Moreover, each securable object has specific elements that are elaborated next. An owner’s (user/group) SID¹⁴. A DCAL (Discretionary Access Control List) which contains a list of group/user SIDs and the access rights each of them has - as shown in the screenshot below in the permissions tab. A SACL (System Access Control List) which states what logging/auditing should be done when accessing the object. Also, there is a group associated with the object which is for POSIX compatibility only¹⁵.

Lastly, examples of securable objects (but not limited to) are: files, directories, desktops, processes, threads, named pipes, mailslots, network shares, printers, private objects, events, semaphores, WMI namespaces and waitable timers and windows stations¹⁶..



¹¹ <https://medium.com/@boutnaru/windows-objects-2c289da600bf>

¹² <https://medium.com/@boutnaru/windows-security-security-descriptor-sd-ba95b8fa048a>

¹³ <https://learn.microsoft.com/en-us/windows/win32/secauthz/securable-objects>

¹⁴ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

¹⁵ <https://renenyffenegger.ch/notes/Windows/development/objects/securable/index>

¹⁶ http://winapi.freitechsecrets.com/win32/WIN32Securable_Objects.htm

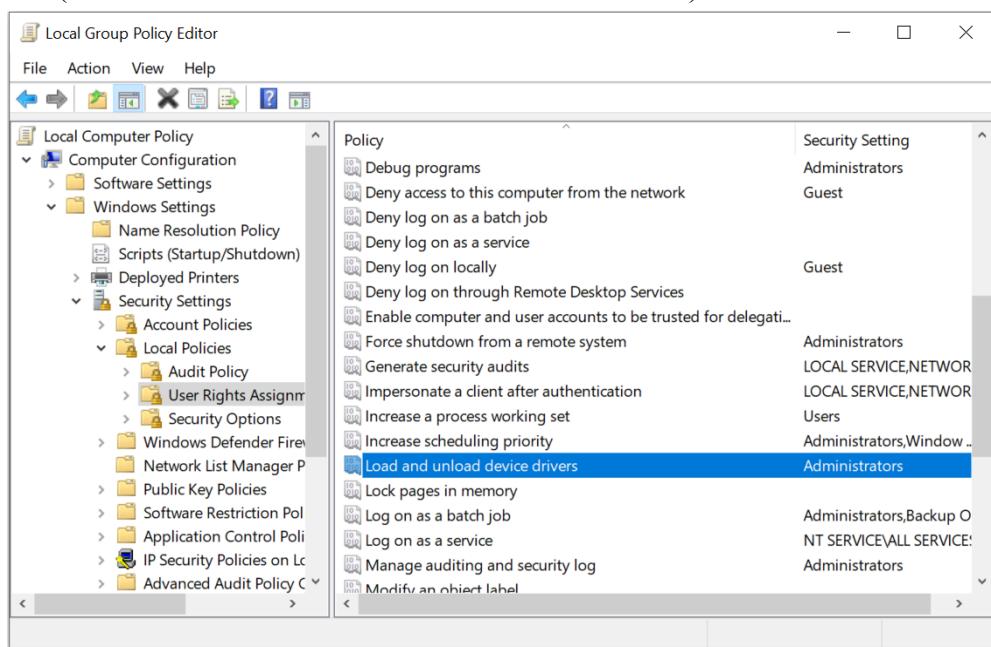
Privileges

Privileges are rights given for a specific account (user/group) which allows performing different system related operations on the local computer. Think about: changing the system time, loading a device driver, shutting down the system and more. There is a difference between access rights to privileges¹⁷.

Thus, we can say that privileges control the access to system resources/system related tasks while access rights control access to securable objects (such as files, directories, registry keys and more). We assign privileges to user/group accounts whereas access rights are granted as part of DACLs.

Moreover, the operating system represents a privilege in a category of “User Rights Assignments”. We can modify them using the “Local Group Policy” (or the “Group Policy”) MMC snap-in¹⁸ - as shown in the screenshot below.

Lastly, the privileges are defined using constants in the following pattern “SE_[DESCRIPTION]_NAME” and also has a text format which is in the pattern of “Se[DESCRIPTION]Privilege”. A couple of examples are: “SE_CREATE_PAGEFILE_NAME”\“SeCreatePagefilePrivilege” which enables creating a new pagefile, “SE_DEBUG_NAME”\“SeDebugPrivilege” which is required for debugging/adjusting the memory of a processes owned by a different user account and “SE_LOAD_DRIVER_NAME”\“SeLoadDriverPrivilege” which is required to load/unload a device driver (it is also the one marked in the screenshot below).



¹⁷ <https://learn.microsoft.com/en-us/windows/win32/secauthz/privileges>

¹⁸ <https://learn.microsoft.com/en-us/windows/win32/secauthz/privilege-constants>

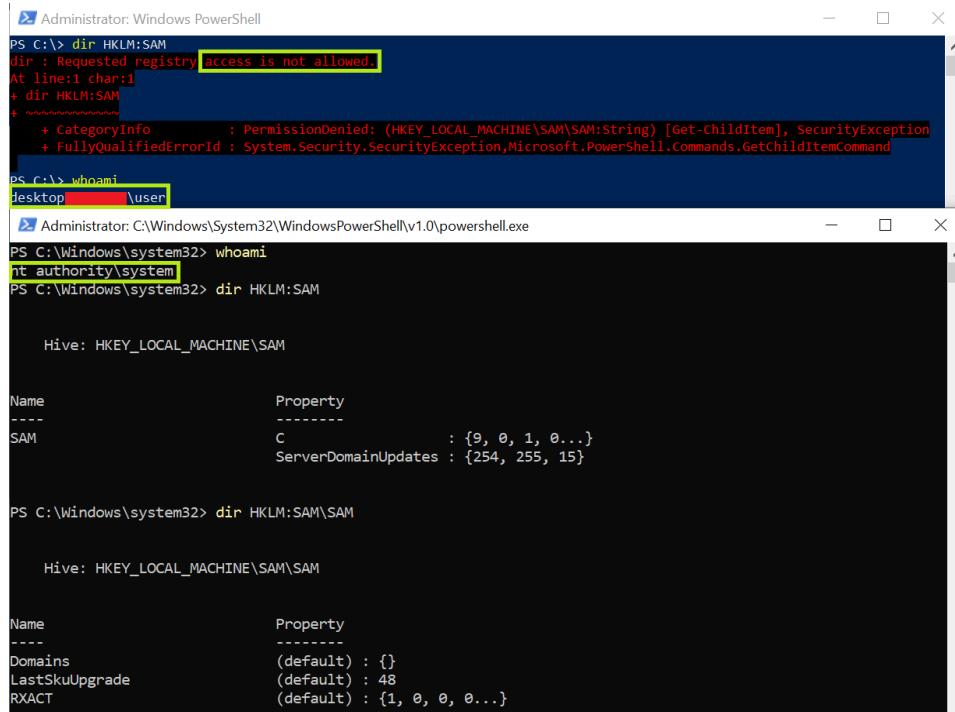
SAM (Security Account Manager)

SRM (Security Account Manager) is the DB in Windows that stores the user names/passwords of the local user defined on the system. By configuring SAM we allow users to authenticate to the local system¹⁹.

Moreover, the SAM file is located at “%windir%\System32\config\SAM” which is mounted in the registry in the following “HKEY_LOCAL_MACHINE\SAM”²⁰. In order to view its content we need to run as SYSTEM and Local Administrator is not enough - as shown in the screenshot below.

Thus, different hashes can be stored in SAM like LM hash and NTLM hash (more on those and others in future writeups). We can think about SAM as the equivalent of “/etc/passwd”, “/etc/shadow” and “/etc/group” files under Linux.

Because Microsoft wanted to increase the security around the hashes stored in SAM they have created SYSKEY. It uses a system key for encrypting to protect the account password information stored in the SAM. The system key can be saved locally, on a floppy disk or stored locally but protected by an admin password²¹. Lastly, SYSKEY support was removed from Windows 10 version 1709²².



The screenshot shows two separate Windows PowerShell windows. The top window is titled 'Administrator: Windows PowerShell' and has the command PS C:\> dir HKLM:SAM. It outputs:

```
PS C:\> dir HKLM:SAM
dir : Requested registry access is not allowed.
At line:1 char:1
+ dir HKLM:SAM
+ ~~~~~
+ CategoryInfo          : PermissionDenied: (HKEY_LOCAL_MACHINE\SAM\SAM:String) [Get-ChildItem], SecurityException
+ FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.GetChildItemCommand
PS C:\>
```

The bottom window is also titled 'Administrator: Windows PowerShell' and has the command PS C:\Windows\system32> whoami. It outputs:

```
PS C:\Windows\system32> whoami
nt authority\system
PS C:\Windows\system32> dir HKLM:SAM
```

Both windows then show the contents of the SAM registry hive:

```
Hive: HKEY_LOCAL_MACHINE\SAM

Name          Property
----          -----
SAM           C : {9, 0, 1, 0...}
              ServerDomainUpdates : {254, 255, 15}

PS C:\Windows\system32> dir HKLM:SAM\SAM

Hive: HKEY_LOCAL_MACHINE\SAM\SAM

Name          Property
----          -----
Domains       (default) : {}
LastSkuUpgrade (default) : 48
RXACT         (default) : {1, 0, 0, 0...}
```

¹⁹ <https://www.calcomsoftware.com/what-is-windows-security-accounts-manager/>

²⁰ <https://viperone.gitbook.io/pentest-everything/everything/everything-active-directory/credential-access/credential-dumping/security-account-manager-sam>

²¹ <https://learn.microsoft.com/en-us/windows-server/security/kerberos/system-key-utilty-technical-overview>

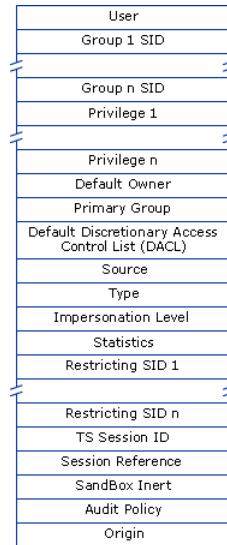
²² <https://learn.microsoft.com/en-us/troubleshoot/windows-server/identity/syskey-exe-utility-is-no-longer-supported>

Access Token

“Access Token” is an object which represents the access rights/privileges/identity for a specific process/thread. The operating system uses the access token in order to identify the user when a specific thread interacts with a securable object²³ or when it tries to perform a system task (that requires some kind of privilege)²⁴.

Thus, if a user authenticates to a system, the Local Security Authority (LSA) creates an access token (to be accurate it is the primary access token, as described in more detail later). It contains the SID of the user, the SIDs of all the groups the user belongs to, a list of privileges, the SID of the owner (user/group), the primary group (for POSIX subsystems), default DACL, source (process that caused the token to be created - RPC/LAN Manager/Session Manager/etc), type (primary/impersonation), impersonation level, restricting SIDs, Terminal service session ID (if relevant), session reference, SandBox inert, audit policy and origin - as shown below²⁵.

Moreover, using different Win32 API functions we can read/manipulate access tokens. As example we can use “OpenProcessToken”²⁶ or “OpenThreadToken”²⁷ to get a handle to the access token of the process/thread. Also, we can use “DuplicateTokenEx”²⁸ for duplicating the access token of the current process and “CreateProcessWithTokenW”²⁹ which allows creation of a process with a specified token. The access token is stored in kernel mode using “struct _TOKEN”³⁰.



²³ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

²⁴ <https://learn.microsoft.com/en-us/windows/win32/secauthz/access-tokens>

²⁵ [https://learn.microsoft.com/pt-pt/previous-versions/windows/server/cc783557\(v=ws.10\)](https://learn.microsoft.com/pt-pt/previous-versions/windows/server/cc783557(v=ws.10))

²⁶ <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-openprocessstoken>

²⁷ <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-openthreadtoken>

²⁸ <https://learn.microsoft.com/en-us/windows/win32/api/securitybaseapi/nf-securitybaseapi-duplicateokenex>

²⁹ <https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-createprocesswithtokenw>

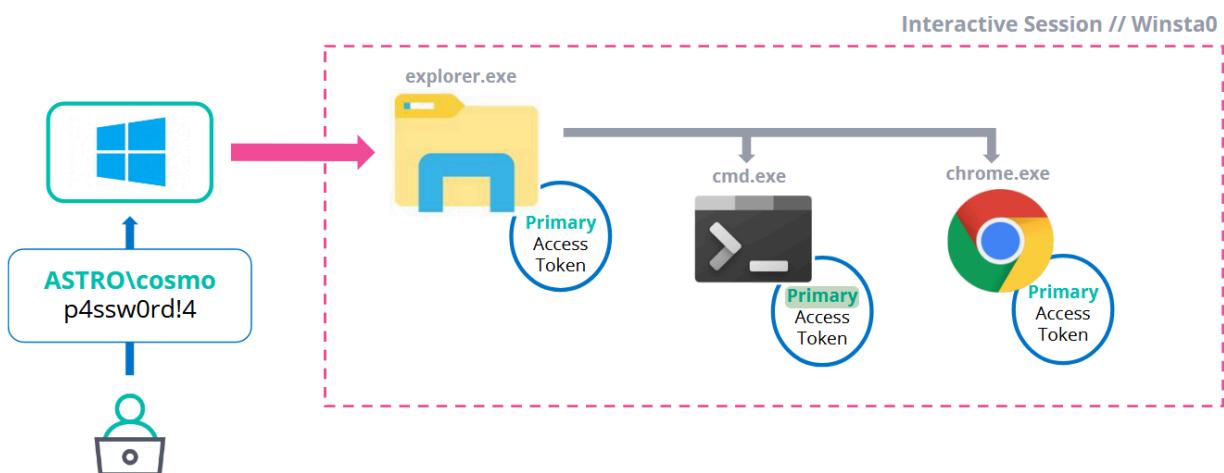
³⁰ <https://www.ired.team/miscellaneous-reversing-forensics/windows-kernel-internals/how-kernel-exploits-abuse-tokens-for-privilege-escalation>

Primary Access Token

Overall, there are two types of access tokens³¹ - as stated in the type field of the access token. Those are “Primary Token” and “Impersonation Token”. In this writeup I am going to focus on the first one.

A primary token can only be associated with a process. Processes inherit a copy of the parent’s process primary token³². Due to that, when a thread is attempting to access a securable object³³ by default this token is checked (threads can have an impersonation token but that is for a different writeup). Also, this token belongs to the user account that created the process³⁴.

Thus, every process has a primary token that it gets from its parent process - as shown in the diagram below³⁵. Because primary tokens are associated with processes they are also known as “Process Tokens”. We can also state that they are created while authenticating interactively³⁶.



³¹ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

³² <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/access-tokens>

³³ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

³⁴ <https://security101.medium.com/better-know-a-data-source-access-tokens-and-why-theyre-hard-to-get-7bc951eae0b9>

³⁵ <https://i.blackhat.com/USA-20/Thursday/us-20-Burgess-Detecting-Access-Token-Manipulation.pdf>

³⁶ <https://sensepost.com/blog/2022/abusing-windows-tokens-to-compromise-active-directory-without-touching-lsass/>

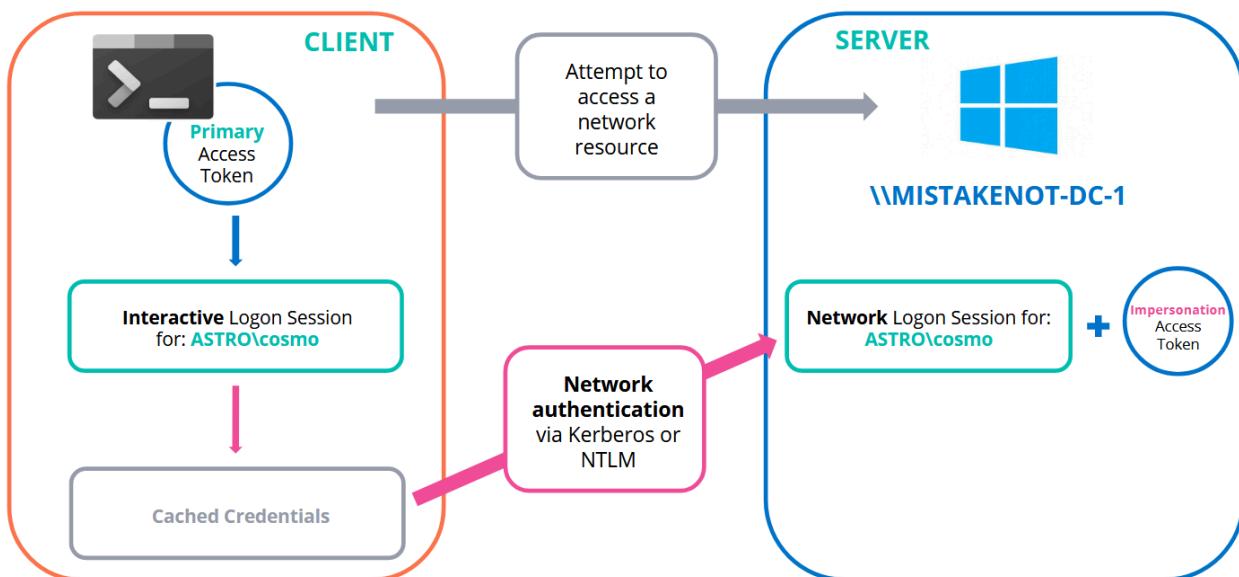
Impersonation Access Token

Overall, there are two types of access tokens³⁷ - as stated in the type field of the access token. Those are “Primary Token” and “Impersonation Token”. In this writeup I am going to focus on the second one.

Basically, impersonation is a mechanism which allows server processes to run by using the credentials of some client. Meaning using credentials of another user than its primary token³⁸.

Thus, when can that impersonation allow a thread to switch to a different security context. By default, threads inherit the same security context as the primary token³⁹ of the process⁴⁰.

One of the main use-cases for impersonation is asking the a server to execute code on behalf of the user performing a network authentication - as shown in the diagram below⁴¹. It can also be used for cases where we want an application/process to have a thread running code with a different security context (than the other threads). However, we need to be careful because all the threads share the same memory space so one thread can hijack the execution flow of another.



³⁷ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

³⁸ <https://medium.com/@boutnaru/windows-security-primary-access-token-e295a35796a9>

³⁹ <https://medium.com/@boutnaru/windows-security-primary-access-token-e295a35796a9>

⁴⁰ <https://i.blackhat.com/USA-20/Thursday/us-20-Burgess-Detecting-Access-Token-Manipulation.pdf>

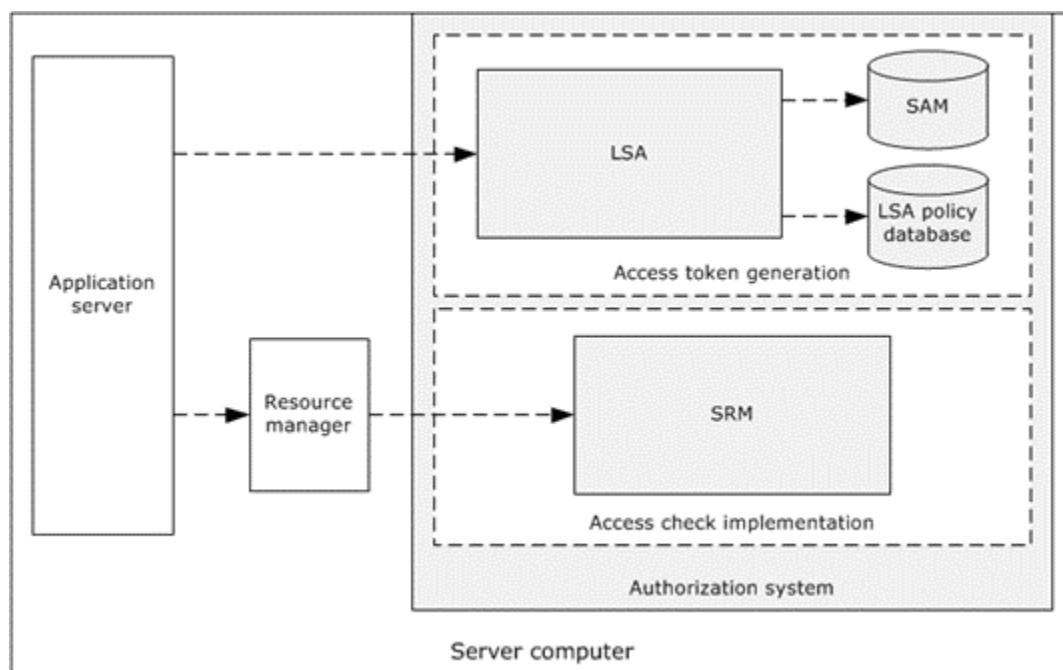
⁴¹ <https://i.blackhat.com/USA-20/Thursday/us-20-Burgess-Detecting-Access-Token-Manipulation.pdf>

SRM (Security Reference Monitor)

SRM (Security Reference Monitor) is a component that is part of the Windows executive (stored in %systemroot%\System32\ntoskrnl.exe). SRM is responsible for implementing the authorization system (together with LSA as shown in the diagram below). Also, SRM implements the access check algorithm⁴². This means it checks the access to different resources by getting the access token⁴³ of the subject and comparing it to the ACEs (Access Control Lists) in the security descriptor of the securable object⁴⁴.

Moreover, the routines that provide a direct interface with the SRM are those prefixed with “Se”⁴⁵. An example of such function is: “SeAccessCheck” which determines if the requested access to an object can be granted⁴⁶. If we want we can go over a reference implementation of “SeAccessCheck” as part of ReacOS⁴⁷.

Lastly, we can say that the “Object Manager” uses SRM to check if a specific process/thread has the proper rights to execute a certain action on an object. Also, it is part of the flow when implementing auditing functionality when objects are being accessed⁴⁸.



⁴² https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-azod/d28d536d-3973-4c8d-b2c9-989e3a8ba3c5

⁴³ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

⁴⁴ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

⁴⁵ <https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-security-reference-monitor>

⁴⁶ <https://learn.microsoft.com/en-us/windows-hardware/drivers/ddi/wdm/nf-wdm-seaccesscheck>

⁴⁷ <https://github.com/reactos/reactos/blob/master/ntoskrnl/se/accesschk.c#L1966>

⁴⁸ <https://cs.gmu.edu/~menasce/osbook/nt/sld034.html>

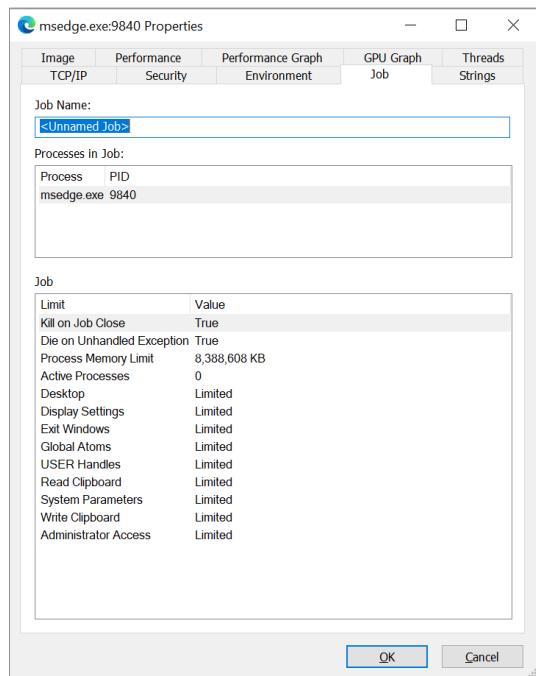
Job Object

By using a job object we can manage a group of processes as one unit. Thus, an operation performed on a job object affects all the processes which are part of that job⁴⁹.

Moreover, in order to create a job object we can use the Win32 API call “CreateJobObjectA”⁵⁰. When creating a job it has no processes associated with it, so we need to use the function “AssignProcessToJobObject”⁵¹. Until Windows 8/Windows Server 2012 a process could be associated with one job only. By the way, for getting an handle for an existing object we can use the “OpenJobObjectA” function⁵².

Overall, by using jobs we can limit the usage of system resources by processes like: process priority, time limit, working set, number of child processes, desktop creation, writing data to the clipboard and more. For setting the limits we use the function “SetInformationJobObject”⁵³.

Lastly, job objects are being used in sandboxes like in web browsers (shown in the screenshot below) and are one of the building blocks of “Windows Containers”. There are also named/unnamed, securable objects⁵⁴ and shareable objects - as shown in the screenshot below taken from Sysinternals’ “Process Explorer”⁵⁵.



⁴⁹ <https://learn.microsoft.com/en-us/windows/win32/procthread/job-objects>

⁵⁰ <https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-createjobobjecta>

⁵¹ <https://learn.microsoft.com/en-us/windows/win32/api/jobapi2/nf-jobapi2-assignprocessesjobobject>

⁵² <https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-openjobobjecta>

⁵³ <https://learn.microsoft.com/en-us/windows/win32/api/jobapi2/nf-jobapi2-setinformationjobobject>

⁵⁴ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

⁵⁵ <https://learn.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite>

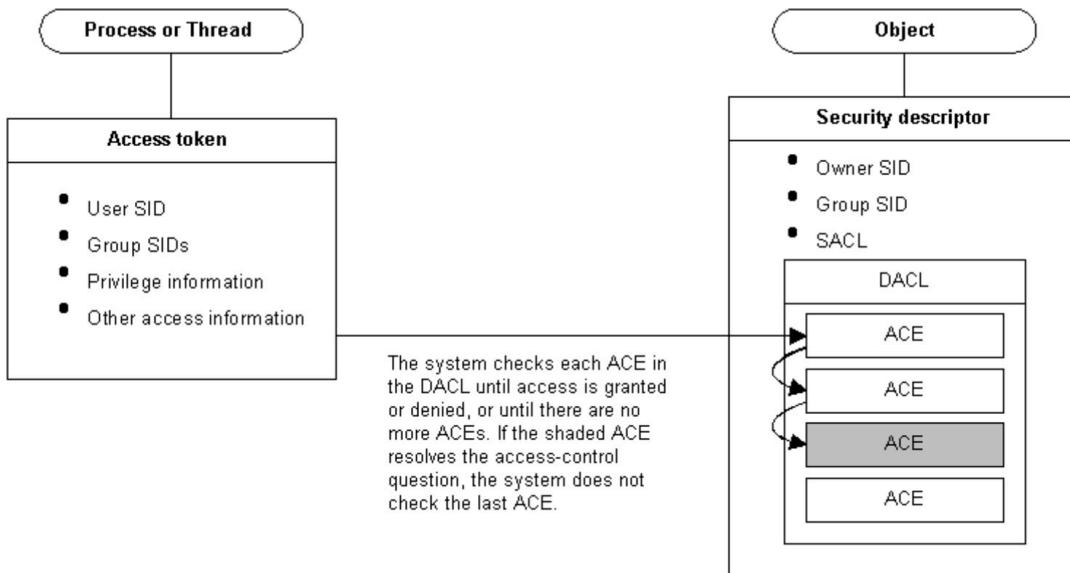
ACL (Access Control List)

ACL (Access Control List) is a list of ACEs (Access Control Entries). Every ACE identifies a trustee (user account/group/logon session) and the relevant allowed/denied/audited access for that trustee⁵⁶.

Overall, there are two types of ACLs which are in use in Windows systems: DACL aka as “Discretionary Access Control List” and SACL aka “System Access Control List”⁵⁷. More information about those types in future writeups. Those types of ACLs are part of the security information stored as part of the “Security Descriptor”⁵⁸ related to securable objects⁵⁹ - as shown in the diagram below⁶⁰.

Moreover, every ACE has four main components. The first, the SID⁶¹ to whom the access information in this ACE is relevant for. Second, a flag denoting the type of ACE (deny/allow/audit). Third, flags regarding the inheritance of the specific ACE. Forth, an access mask which is a 32 bit that describes the rights relevant for this ACE⁶².

Lastly, due to the fact we have DACL and SACL, usually when saying ACE we talk about the first one and when saying System ACE we mean the second one.



⁵⁶ <https://learn.microsoft.com/en-us/windows/win32/secauthz/access-control-lists>

⁵⁷ <https://www.securew2.com/blog/windows-access-control-acl-dacl-sacl-ace>

⁵⁸ <https://medium.com/@boutnaru/windows-security-security-descriptor-sd-ba95b8fa048a>

⁵⁹ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

⁶⁰ <https://developer.aliyun.com/article/747446>

⁶¹ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

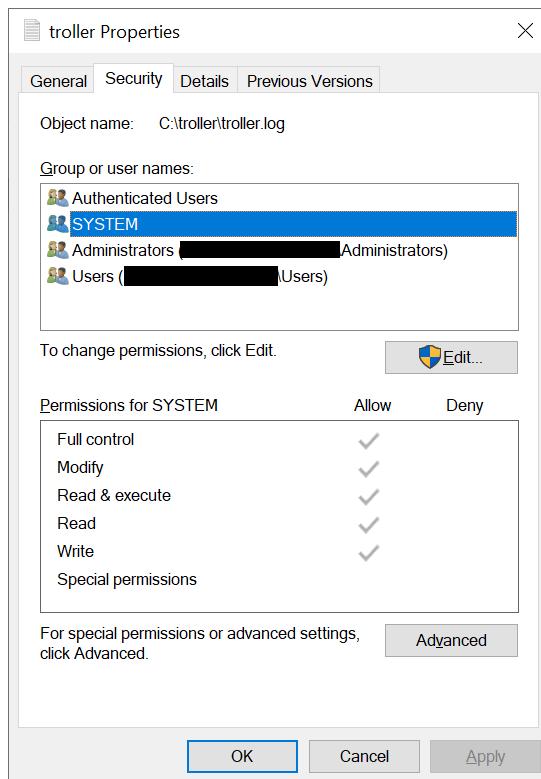
⁶² <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/acls-dacls-sacs-aces>

DACL (Discretionary Access Control List)

In general DACL (Discretionary Access Control List) is an ACL⁶³ which identifies the trustees that allowed/denied access to a securable object⁶⁴. Thus, if the securable object does not have any DACL (Null) the SRM⁶⁵ allows everyone full access to it. If the list of ACL is empty no one has any access to the object⁶⁶.

Moreover, when a thread tries to access a securable object, the system goes over the ACEs in the DACL until it finds one that allows/denies the access (think about it like firewall rules). The predefined order of ACEs are as follows: all explicit ACEs are before inherited ACEs and the inherited ones are placed in the order in which they are inherited. By the way, in every level access denied ACEs are placed before the access allowed ACEs ones⁶⁷.

Lastly, for configuring a DACL using the UI we just go to the properties of the object and select the “security tab”, there we can edit the DACL of that specific object - as shown in the screenshot below. We can also use CLI tools like cacls.exe/icacls.exe (but that is for a different writeup).



⁶³ <https://medium.com/@boutnaru/the-windows-security-journey-acl-access-control-list-b7d9a6fe4282>

⁶⁴ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

⁶⁵ <https://medium.com/@boutnaru/windows-security-srm-security-reference-monitor-d715f96d9fd6>

⁶⁶ <https://learn.microsoft.com/en-us/windows/win32/secauthz/dacls-and-aces>

⁶⁷ <https://www.tenouk.com/ModuleH2.html>

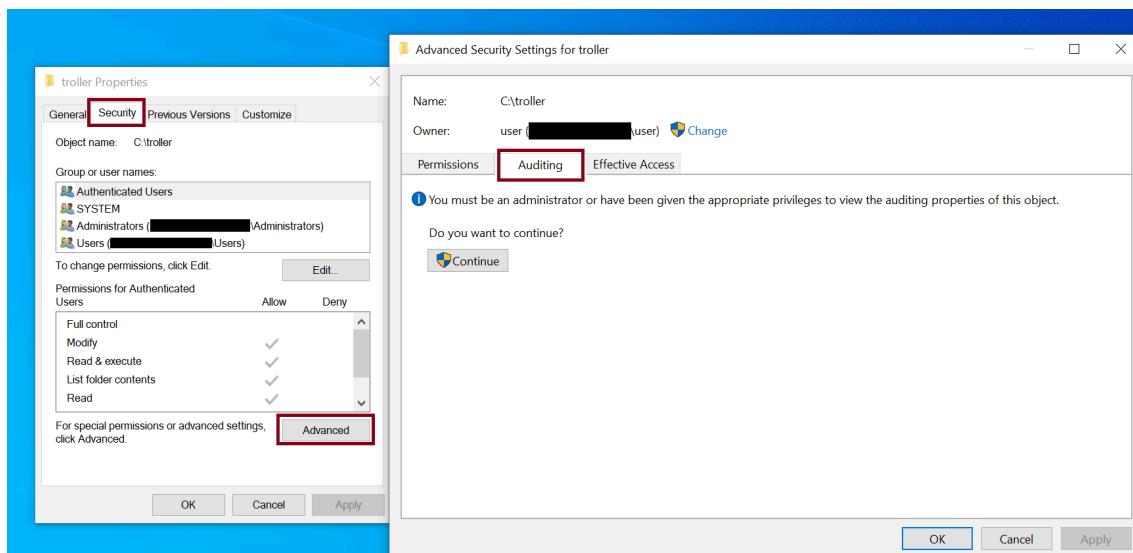
SACL (System Access Control List)

Overall, a SACL (System Access Control List) is an ACL⁶⁸ which enables the administrators of a system to audit attempts of accessing securable objects⁶⁹. Every ACE (Access Control Entry) defines the type of access attempt that causes to generate an audit trail while performed by a trustee⁷⁰.

Thus, an ACE as part of an SACL can emit an audit record when an access attempt is failed/succeeds/both. The system writes audit messages to the security event log⁷¹. In order to read/write object's SACL the relevant thread/process should enable as part of its access token⁷² the “SE_SECURITY_NAME” privilege⁷³.

Moreover, the “SE_SECURITY_NAME” privilege is defined as managing auditing and the security log⁷⁴. We can use “SetNamedSecurityInfoA”/“SetNamedSecurityInfoW”⁷⁵ or “GetNamedSecurityInfoA”/“GetNamedSecurityInfoW” in order to access the SACL. Those functions enable the “SE_SECURITY_NAME” privilege.

Lastly, in order to configure an SACL on a securable object like a file/directory we go to its properties and then we go to the “security tab”. In the “security tab” we need to press the “Advanced” button - as shown in the screenshot below. In the advanced security setting we can go to the “auditing tab” - also shown in the screenshot below.



⁶⁸ <https://medium.com/@boutnaru/the-windows-security-journey-acl-access-control-list-b7d9a6fe4282>

⁶⁹ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

⁷⁰ <https://learn.microsoft.com/en-us/windows/win32/secauthz/access-control-lists>

⁷¹ <https://learn.microsoft.com/en-us/windows/win32/secauthz/audit-generation>

⁷² <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

⁷³ <https://medium.com/@boutnaru/windows-security-privileges-b8fe18cf3d5a>

⁷⁴ <https://jeffpar.github.io/kbarchive/kb/188/Q188855/>

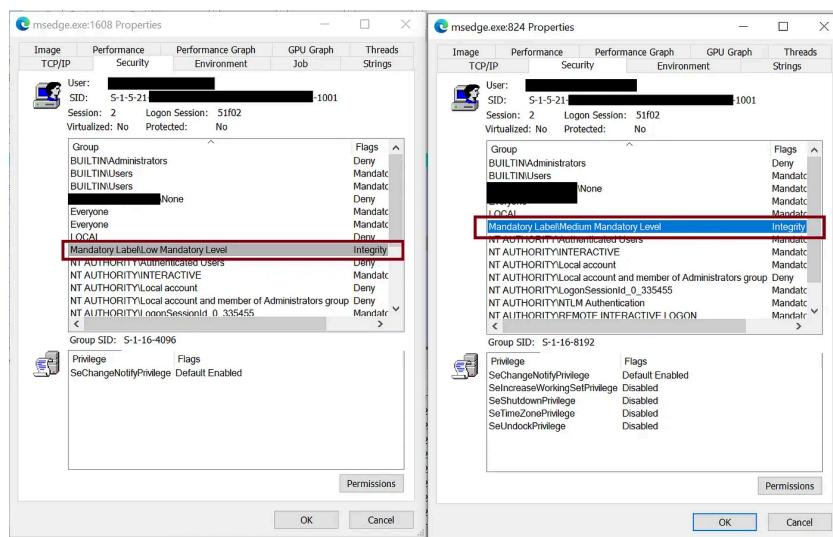
⁷⁵ <https://learn.microsoft.com/en-us/windows/win32/api/aclapi/nf-aclapi-setnamedsecurityinfow>

Mandatory Integrity Control (MIC)

In general, “Mandatory Integrity Control” (MIC) has been added to Windows from Vista for adding support of MAC (Mandatory Access Control) to running processes⁷⁶. This is done using a new attribute called “Integrity Level” (IL). MIC is designed to control access to securable objects⁷⁷. The mechanism works in conjunction with DACL⁷⁸. It is important to know that MIC evaluates access before the access check is made versus the object’s DACL, and itself is implemented as ACEs (Access Control Entries) using special SIDs⁷⁹.

Moreover, each security principal⁸⁰ and any securable object is marked with an integrity level which is aimed at determining their level of access/protection. In Windows we have different integrity levels: “untrusted” (S-1-16-0), “low” (S-1-16-4096), “medium” (S-1-16-8192), “high” (S-1-16-12288) and “system” (S-1-16-16384). By default, standard users are given an integrity level of “medium” while elevated users get “high”. Also, objects which lack an integrity level are treated as “medium”⁸¹.

Lastly, the integrity level SIDs (as shown above) are stored in the SACL⁸² of the secure object⁸³. The Windows security policy states that a process can’t interact with another process that has a higher integrity level⁸⁴, due to that it is also used by different sandbox implementations (like with Web Browsers). By the way, the integrity level is stored in the access token⁸⁵ of a process/thread - as shown in the screenshot below.



⁷⁶ <https://learn.microsoft.com/en-us/windows/win32/secauthz/mandatory-integrity-control>

⁷⁷ <https://medium.com/@boutnaru/windows-securable-objects-311a9d6c83ad>

⁷⁸ <https://medium.com/@boutnaru/the-windows-security-journey-dacl-discretionary-access-control-list-c74545e472ec>

⁷⁹ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

⁸⁰ <https://medium.com/@boutnaru/windows-security-privilege-escalation/integrity-levels>

⁸² <https://medium.com/@boutnaru/the-windows-security-journey-sacl-system-access-control-list-32488dcc80d7>

⁸³ <https://learn.microsoft.com/en-us/windows/win32/secauthz/mandatory-integrity-control>

⁸⁴ https://en.wikipedia.org/wiki/Mandatory_Integrity_Control

⁸⁵ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

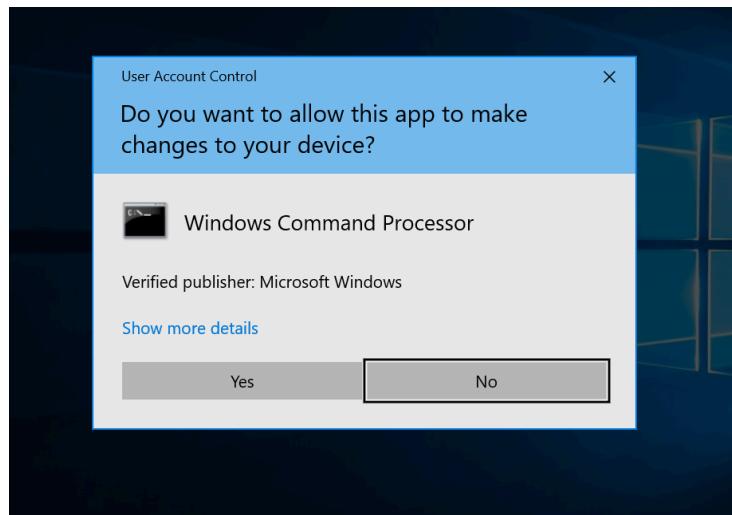
UAC (User Account Control)

The goal of UAC (User Account Control) is to reduce the risk of malware by limiting the ability of malicious code from running with administrator permissions. When UAC is used an application requests an access token⁸⁶ with administrator permissions must prompt the user for consent⁸⁷ - as shown in the screenshot below.

UAC (User Account Control) provides MAC (Mandatory Access Control) which was introduced as part of Windows Vista/Server 2008. Together with UIPI⁸⁸ UAC is used to isolate between applications with the same user on the same session. When a user tries to perform an operation that requires admin access it will trigger UAC (if it's enabled). Examples of such operations (but not limited to) are: executing an application as an administrator, changing system-wide settings, installing a device driver, changing UAC settings, configuring windows update, opening the registry editor, changing power setting and turning on/of Windows features⁸⁹.

Moreover, when UAC is enabled when an administrator logs on to a system two separate access tokens are created (standard access token and administrator access token). The difference between them is that the administrative privileges and SIDs are removed from the standard one⁹⁰.

Lastly, UAC is composed of several technologies in order to provide its capabilities, among them are: file and registry virtualization, same desktop elevation, filtered token, UIPI, protected internet explorer and installer detection⁹¹.



⁸⁶ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

⁸⁷ <https://medium.com/@boutnaru/the-windows-process-journey-consent-exe-consent-ui-for-administrative-applications-d8e6976e8e40>

⁸⁸ <https://medium.com/@boutnaru/windows-security-user-interface-privilege-isolation-uipi-db790ad173eb>

⁸⁹ https://en.wikipedia.org/wiki/User_Account_Control

⁹⁰ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/user-account-control/how-it-works>

⁹¹ <https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/disable-user-account-control>

User Interface Privilege Isolation (UIPI)

User Interface Privilege Isolation (UIPI) was introduced in Windows 2008/Vista with the goal of mitigating “Shatter Attacks”. Those types of attacks leverage the Windows’s message passing system which can be used to inject arbitrary commands/code to any application/service running in the same session, those we are using a “message loop”⁹².

UIPI allows isolating processes running as a full administrator from processes running as an account with lower permissions than an administrator on the same interactive desktop. UIPI is specific to the windowing/graphic subsystem (aka Windows USER). Thus, a process with lower privileges can’t perform operations on a process with higher privileges like: DLL injection, thread hooks for attaching, journal hooks for attaching, use window messages API (SendMessage/PostMessage) and more⁹³.

However, there are still resources that are shared between processes at different privilege levels like: clipboard, global atom table, desktop window and the desktop heap read-only shared memory. Also, painting on a screen is not controlled using UIPI, so a lower privilege application can paint over the surface region of a higher privilege application window - the GDI model does not allow control over painting surfaces⁹⁴.

Lastly, we can control the configuration of UIPI using the “EnableUIPI” value under the “HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\” registry path - as shown in the screenshot below⁹⁵. A value of “0” disables UIPI, and if the value is not present by default it means UIPI is enabled⁹⁶.

Policies	{a42c2ccb-67...}	REG_SZ	Microsoft Winc
Attachment:	{b9815375-5d...}	REG_SZ	Microsoft Winc
Explorer	{DC6EFB56-9...}	REG_SZ	Adobe PDF Pre
NonEnum	{f8b8412b-de...}	REG_SZ	Microsoft Winc
System	EnableUIPI	REG_DWORD	0x00000000 (0)

⁹² <https://www.slideserve.com/milek/shoot-the-messenger-win32-shatter-attacks-by-brett-moore>

⁹³ [https://learn.microsoft.com/en-us/previous-versions/aa905330\(v=msdn.10\)](https://learn.microsoft.com/en-us/previous-versions/aa905330(v=msdn.10))

⁹⁴ <https://learn.microsoft.com/en-us/windows/win32/gdi/painting-and-drawing>

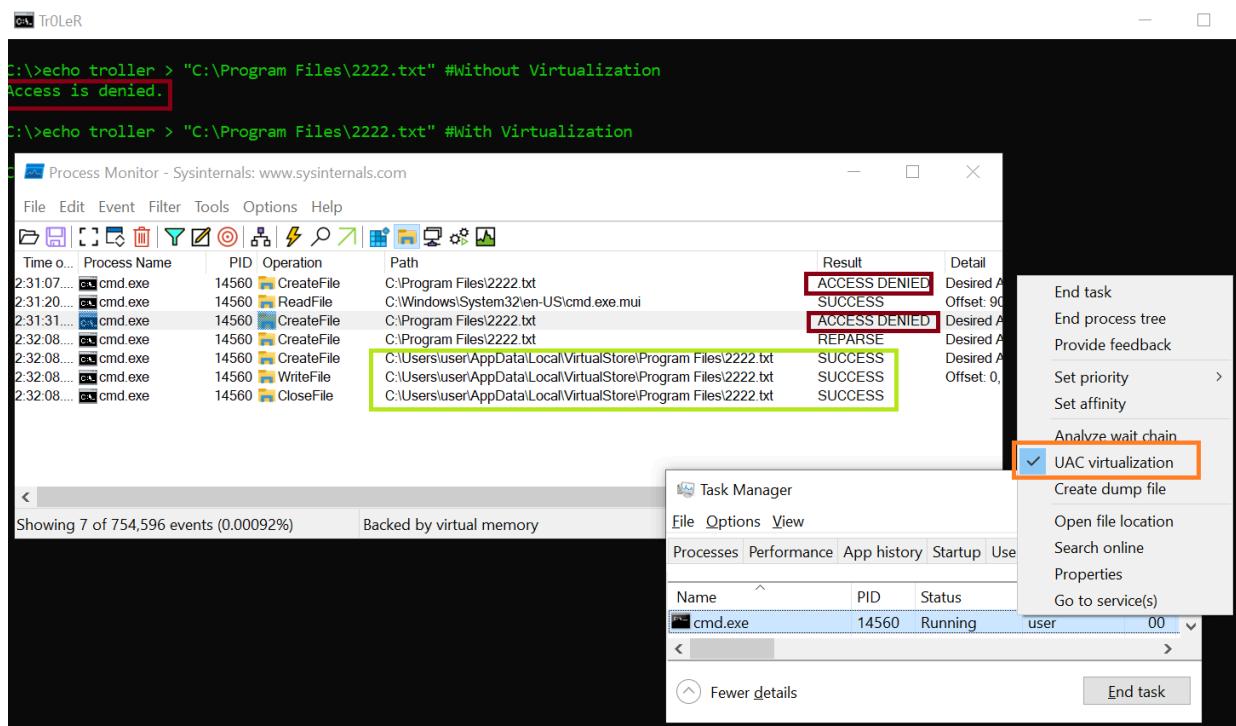
⁹⁵ <https://www.tipandtrick.net/fix-third-party-input-language-method-editor-ime-issues-in-ie-and-windows-vista-by-disabling-uipi/>

⁹⁶ <http://nferrie.epizy.com/papers/antidebug.pdf>

File Virtualization

Due to security considerations (UAC enabled), it is also known as “UAC File Virtualization”) as of Windows Vista it does allow standard (non-administrator) users to access/manipulate folders (like “Program Files” and the “Windows” directory) or specific registry areas - as was allowed in previous Windows versions. However, because there are legacy applications which expect doing those operations Windows include “File and Registry Virtualization”⁹⁷.

Thus, if we have an application running with-out administrative permissions and it tries to write to “Program Files” it will be redirected to “C:\Users\%username%\AppData\Local\VirtualStore\Program Files\” and the operation will succeed⁹⁸. Without the file virtualization the operation is going to fail - as shown in the screenshot below. By the way, I have enabled the virtualization on “cmd.exe” using “Task Manager”⁹⁹.



⁹⁷ <https://www.c-sharpcorner.com/uploadfile/GemingLeader/windows-file-and-registry-virtualization/>

⁹⁸ <https://flylib.com/books/en/2.955.1.34/1/>

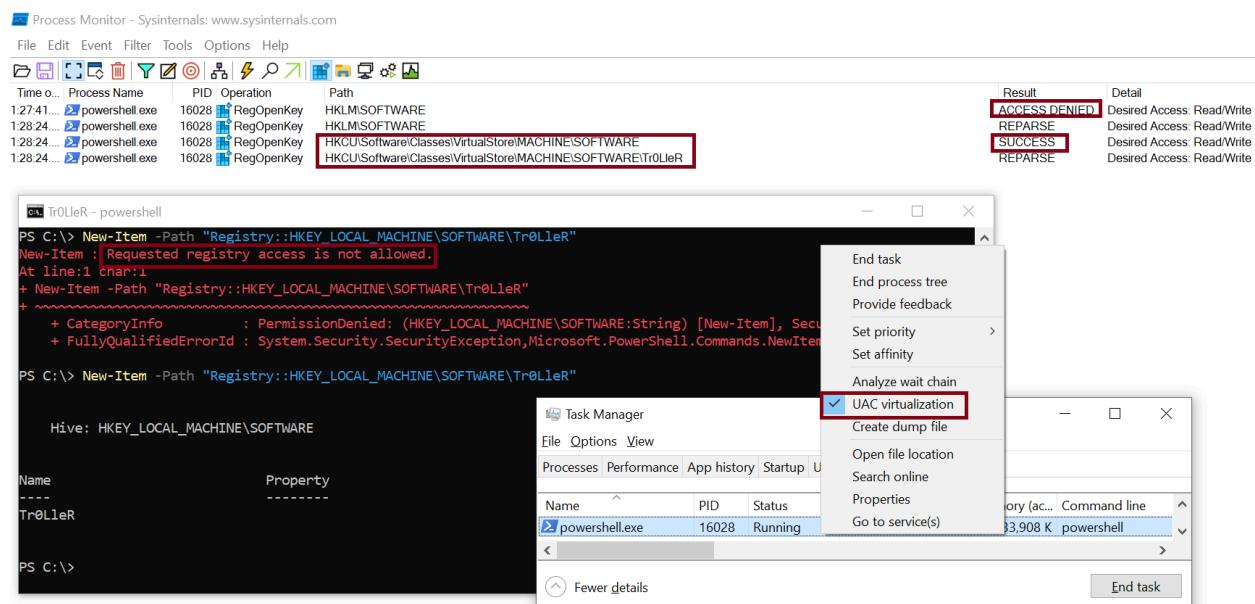
⁹⁹ <https://www.experts-exchange.com/questions/28943516/What-is-UAC-Virtualization-in-the-Process-TASK-Manager.html>

Registry Virtualization

Due to security considerations (like UAC enabled) as of Windows Vista it does allow standard (non administrator) users to access/manipulate specific registry areas - as was allowed in previous Windows versions. However, because there are legacy applications which expect doing those operations Windows include “File and Registry Virtualization”¹⁰⁰.

Thus, registry virtualization is an application compatibility technology. It helps registry write operations that can have global impact to be redirected to a per-user location. The redirection is transparent to the application writing/reading from the registry. There are specific cases in which it is enabled by default like for 32-bit processes which are interactive¹⁰¹.

For example, if we have an application running with-out administrative permissions and it tries to write/read to “HKEY_LOCAL_MACHINE\SOFTWARE” it will be redirected to “HKEY_CURRENT_USER\Software\Classes\VirtualStore\MACHINE\SOFTWARE” and the operation will succeed. Without the registry virtualization the operation is going to fail - as shown in the screenshot below. By the way, I have enabled the virtualization on “powershell.exe” using “Task Manager”¹⁰².



¹⁰⁰ <https://www.c-sharpcorner.com/uploadfile/GemingLeader/windows-file-and-registry-virtualization/>

¹⁰¹ <https://learn.microsoft.com/en-us/windows/win32/sysinfo/registry-virtualization>

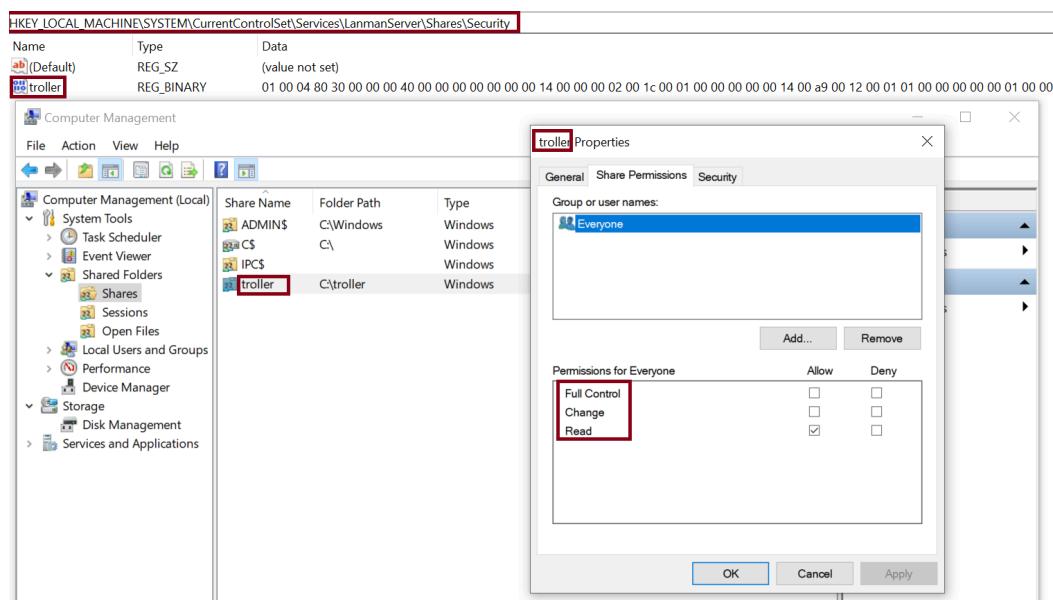
¹⁰² <https://www.experts-exchange.com/questions/28943516/What-is-UAC-Virtualization-in-the-Process-TASK-Manager.html>

Share Permissions (Network Shares)

In general share permissions are used in order to control the access from Windows shares¹⁰³, which are folders exposed over the network. It is important to understand that share permissions apply to all the files/folders in the share. Also, share permissions are relevant for different file systems (NTFS/FAT/FAT32) as opposed to file/directory permissions which are only relevant for NTFS based filesystems¹⁰⁴.

Moreover, there are three types of share permissions: “Read”, “Change” or “Full Control” - as shown in the screenshot below. When accessing a share that exposes a directory stored on an NTFS file system we use share permissions together with NTFS permissions. Thus, the most restrictive permission wins¹⁰⁵.

Lastly, share permissions are stored in the registry at the following location: “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Shares\Security”¹⁰⁶ - as shown in the screenshot below. By the way, if we want we can also limit the number of users that can access the share simultaneously.



¹⁰³ <https://medium.com/@boutinaru/the-windows-concepts-journey-windows-shares-8f9b60b8efd1>

¹⁰⁴ <https://blog.netwrix.com/2018/05/03/differences-between-share-and-ntfs-permissions/>

¹⁰⁵ <https://www.varonis.com/blog/ntfs-permissions-vs-share>

¹⁰⁶ <https://learn.microsoft.com/en-us/answers/questions/228360/share-folder-access-issue>

SysMon (System Monitor)

In general SysMon is a device driver and a Windows service which allows monitoring and logging System activities to the Windows event log¹⁰⁷ - as shown in the screenshot below¹⁰⁸. It is part of the Sysinternals software package published by Microsoft¹⁰⁹. After installing SysMon (“sysmon64/sysmon -i [<configfile>]”) it remains resident across system reboots.

Moreover, SysMon has different capabilities such as logging process creation with full command line, record hash of process image files, logging loading of DLLs/Drivers with their signatures and hashes, logging the opening of disks/volumes in raw mode, logging network connections, generate events from early in the boot process (to capture activity by kernel mode malware), rule filtering, logging registry events, logging file deletion events and more¹¹⁰. Examples are shown also in the screenshot below.

Overall, the configurations of SysMon are stored in XML files, which allow event filtering based on type and their specific payload¹¹¹. There are different templates for SysMon in Github which we can use like: SwiftOnSecurity SysMon config¹¹² and a repository of SysMon configuration modules by Olaf Hartong¹¹³.

Lastly, it is recommended to send the events generated by SysMon to a SIEM (Security Information and Event Management). By the way, there is also a version of SysMon for the Linux operating system¹¹⁴.

The screenshot shows the SysMon application interface. On the left is a tree view of system components, with 'Sysmon' expanded and its 'Operational' sub-node selected. The main window displays a table of events. One event is highlighted in blue, showing details in a modal dialog. The event details include:

Level	Date and Time	Source	Event ID	Task Category
Information	5/19/2017 1:29:10 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)
Information	5/19/2017 1:29:10 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)
Information	5/19/2017 1:29:04 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)
Information	5/19/2017 1:28:50 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	5/19/2017 1:28:49 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	5/19/2017 1:28:04 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)
Information	5/19/2017 1:27:42 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)
Information	5/19/2017 1:27:42 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)
Information	5/19/2017 1:27:40 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)

Detailed event information for the highlighted event:

```
Process Create:  
UtcTime: 2017-05-19 18:28:49.603  
ProcessGuid: {53472fad-3961-591f-0000-00101d2d9e16}  
ProcessId: 18400  
Image: C:\Windows\System32\mmc.exe  
CommandLine: "C:\Windows\system32\mmc.exe" "C:\Windows\system32\eventvwr.msc" /s  
CurrentDirectory: C:\Windows\system32  
User: .pdelgado  
LogonGuid: {53472fad-ed5e-591a-0000-0020983ece00}
```

¹⁰⁷ <https://medium.com/@bouthnary/the-windows-concept-journey-windows-event-logs-a9945bca421f>

¹⁰⁸ <https://www.syspanda.com/index.php/2017/05/19/sysmon-getting-started/>

¹⁰⁹ <https://www.blumira.com/glossary/system-monitor-sysmon/>

¹¹⁰ <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>

¹¹¹ <https://cqueacademy.com/blog/hacks/sysmon>

¹¹² <https://github.com/SwiftOnSecurity/sysmon-config>

¹¹³ <https://github.com/olafhartong/sysmon-modular>

¹¹⁴ <https://github.com/Sysinternals/SysmonForLinux>

ELAM (Early Launch AntiMalware)

ELAM (Early Launch AntiMalware) is a feature that Microsoft provides which allows anti-malware software to start before other 3rd-party components. We can use it for developing drivers that are initialized before other boot start drivers for ensuring that subsequent drivers do not contain malicious code such as malware¹¹⁵.

Overall, ELAM allows the registration of a kernel mode driver which is guaranteed to load early in the boot flow before other drivers. There is a signature data store for ELAM in the registry located at “HKLM\ELAM\[VENDOR_NAME]” in which the vendor stores the allowlist/blocklist AV signatures for the ELAM driver to use. This hive is located at “%windir%\System32\config\ELAM” and loaded by the OS bootloader and unloaded after the ELAM has finished its job. Due to that, it does not appear after the boot process¹¹⁶.

Moreover, as mentioned above ELAM allows antivirus software to scan boot drivers for malware before they are loaded. We can configure what boot start drivers we want to load based on the scan result of ELAM. The options are: “good only” (load only signed drivers which are not known as malware), “good and unknown” (loads only signed drivers or those not detected as malware) and “good, unknown, and bad but critical” (this is the defaults, drivers loaded if Windows does not start without it) and “all” (loads every driver). This can be set using “Local Computer Policy”/GPO in the following path “Computer Configuration\Administrative Templates\System\Early Launch Antimalware”¹¹⁷.

Lastly, the above configuration is stored in the registry in the following location “HKLM\SYSTEM\CurrentControlSet\Policies\EarlyLaunch\DriverLoadPolicy”¹¹⁸. Also, ELAM vendors should be members of MVI which is the “Microsoft Virus Initiative” and their pass the WHQL which is the “Windows Hardware Quality Lab” for their driver¹¹⁹.

```
C:\>sc qc afd
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: afd
        TYPE               : 1 KERNEL_DRIVER
        START_TYPE          : 1 SYSTEM_START
        ERROR_CONTROL       : 1 NORMAL
        BINARY_PATH_NAME    : \\SystemRoot\\system32\\drivers\\afd.sys
        LOAD_ORDER_GROUP   : PNP_TDI
        TAG                :
        DISPLAY_NAME        : Ancillary Function Driver for Winsock
        DEPENDENCIES        :
        SERVICE_START_NAME  :

C:\>sc qc wdboot
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: wdboot
        TYPE               : 1 KERNEL_DRIVER
        START_TYPE          : 0 BOOT_START
        ERROR_CONTROL       : 1 NORMAL
        BINARY_PATH_NAME    : \\SystemRoot\\system32\\drivers\\wd\\wdBoot.sys
        LOAD_ORDER_GROUP   : Early-Launch
        TAG                :
        DISPLAY_NAME        : Microsoft Defender Antivirus Boot Driver
        DEPENDENCIES        :
        SERVICE_START_NAME  :
```

¹¹⁵ <https://learn.microsoft.com/en-us/windows-hardware/drivers/install/early-launch-antimalware>

¹¹⁶ <https://www.anoopcnair.com/understanding-windows-trusted-boot/>

¹¹⁷ <https://www.bleepingcomputer.com/tutorials/configure-early-launch-antimalware-protection/>

¹¹⁸ <https://learn.microsoft.com/en-us/windows-hardware/drivers/install/elam-driver-requirements>

¹¹⁹ <https://learn.microsoft.com/en-us/windows-hardware/drivers/install/elam-prerequisites>

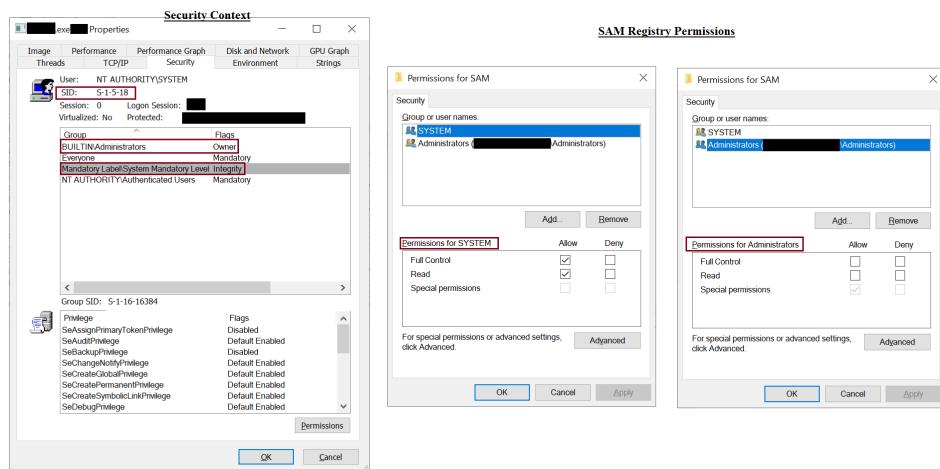
Local System (NT AUTHORITY\SYSTEM)

Local System (NT AUTHORITY\SYSTEM) is a built-in user account (sometimes also called LocalSystem) which is used as a security context by different processes like Windows services¹²⁰ or scheduled tasks¹²¹.

Overall, this user account has unlimited rights to a specific server/computer. By the way, when accessing a resource over the network (like a network share) with a thread/process holding a “SYSTEM” security context, the computer account is used. The name of the user in that case would be “[COUNTER_NAME]\$”¹²².

Moreover, The Local System user is more powerful than the builtin local administrator user. One example of that is that the local Administrator can't read the content of “KEY_LOCAL_MACHINE\SAM\SAM” while the System user can - as shown in the screenshot below. By the way, that subkey holds the db of the “Security Account Manager”¹²³.

Lastly, “NT AUTHORITY\SYSTEM” has a specific SID¹²⁴ which is relevant for all Windows systems in the world that is “S-1-5-18” - as shown below. Also, an access token¹²⁵ that belongs to a process with a security context of SYSTEM (for Windows VISTA+) has a mandatory level¹²⁶ of “system” and it contains the “Builtin\Administrators” group- as shown in the screenshot below. We can also see the list of privileges that are part of the token like: “SeDebugPrivilege” and “SeBackupPrivilege”¹²⁷.



¹²⁰ <https://medium.com/@boutnaru/windows-services-part-1-5d6c2d25b31c>

¹²¹ <https://medium.com/@boutnaru/windows-scheduler-tasks-84d14fe733c0>

¹²² <https://www.libe.net/en-local-system>

¹²³ <https://medium.com/@boutnaru/windows-security-sam-security-account-manager-c93ddadff388a>

¹²⁴ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

¹²⁵ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

¹²⁶ <https://medium.com/@boutnaru/the-windows-security-journey-mandatory-integrity-control-mic-f7963550c0e7>

¹²⁷ <https://learn.microsoft.com/en-us/windows/win32/services/localsystem-account>

Network Service (NT AUTHORITY\NETWORK SERVICE)

Network Service (NT AUTHORITY\NETWORK SERVICE) is a builtin user account which is used as a security context by different services on Windows¹²⁸. Examples of such services are: “RPC Endpoint Mapper”, “Remote Desktop Services”, “Network Location Awareness”, “DNS Client” and “Workstation”. By the way, the SID¹²⁹ of the “Network Service” account is “S-1-5-20”.

Overall, “Network Service” has less permissions/privileges than the LocalSystem account¹³⁰, for example it is not part of the built-in Administrators group. However, the “Network Service” account can still interact over the network using the identity of the computer account¹³¹.

Thus, we can say that the “Network Service” is a predefined local account used by the SCM (Service Control Manager). Also, this account is not recognized by the security subsystem in Windows, so we can’t specify it’s name while using “LookupAccountName” function¹³².

Moreover, when using “net.exe”¹³³ to create a new user named “NetworkService” we get the following error “The account already exists”. However, when trying to retrieve information about the “NetworkService” account we can “The user name could not be found.” - as shown in the screenshot below. Lastly, we can use the “Network Service” account when calling the “CreateService”¹³⁴ function for creating a new service or when using “ChangeServiceConfig”¹³⁵ function to alter configuration parameters of a service.

```
C:\>net user NetworkService /add
The account already exists.

More help is available by typing NET HELPMSG 2224.

C:\>net user NetworkService
The user name could not be found.

More help is available by typing NET HELPMSG 2221.

C:\>
```

¹²⁸ <https://medium.com/@boutnaru/windows-services-part-1-5d6c2d25b31c>

¹²⁹ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

¹³⁰ <https://medium.com/@boutnaru/the-windows-security-journey-local-system-nt-authority-system-f087dc530588>

¹³¹ <https://stackoverflow.com/questions/1301080/confused-over-localsystem-and-localservice-accounts>

¹³² <https://learn.microsoft.com/en-us/windows/win32/services/networkservice-account>

¹³³ <https://medium.com/@boutnaru/the-windows-process-journey-net-exe-net-command-91e4964f20b8>

¹³⁴ <https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-createservicew>

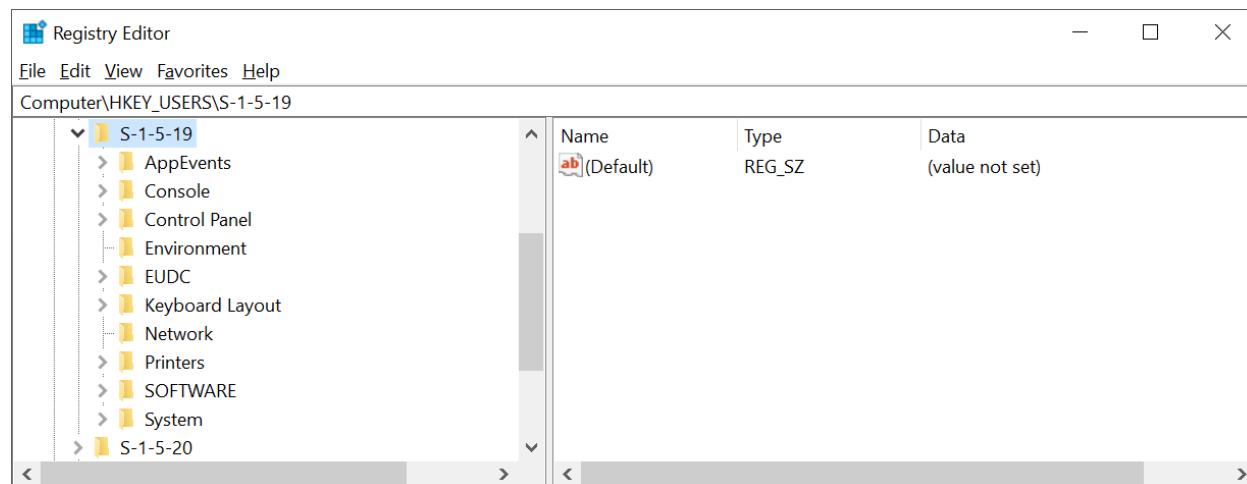
¹³⁵ <https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-changeserviceconfigw>

Local Service (NT AUTHORITY\LOCAL SERVICE)

Local Service (NT AUTHORITY\LOCAL SERVICE) is a builtin user account which is used as a security context by different Windows services¹³⁶. Examples of such services are: “Windows Time”, “SSDP Discovery”, “DHCP Client”, “Windows Audio”, “Windows Defender Firewall”, and “Windows Image Acquisition”. By the way, the SID¹³⁷ of the “Local Service” account is “S-1-5-19”.

Overall, “Local Service” is a built-in user account like “Network Service” built-in user account¹³⁸ which is used to run least-privileges services. However, the user is limited to the local computer (as opposed to the “Network Service” account). Thus, this user should be used anytime the worker process/thread does not need access outside the specific workstation/server¹³⁹, because it accesses the network as an anonymous user.

Lastly, we can use the “Network Service” account when calling the “CreateService”¹⁴⁰ function for creating a new service or when using “ChangeServiceConfig”¹⁴¹ function to alter configuration parameters of a service. As with “LocalSystem” and “NetworkService” the “LocalService” has its own profile in “HKEY_USERS\S-1-5-19”¹⁴².



¹³⁶ <https://medium.com/@boutnaru/windows-services-part-1-5d6c2d25b31c>

¹³⁷ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

¹³⁸ <https://medium.com/@boutnaru/the-windows-security-journey-network-service-nt-authority-network-service-e8706688e383>

¹³⁹ <https://stackoverflow.com/questions/1301080/confused-over-localsystem-and-localservice-accounts>

¹⁴⁰ <https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-createserviceew>

¹⁴¹ <https://learn.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-changeserviceconfigw>

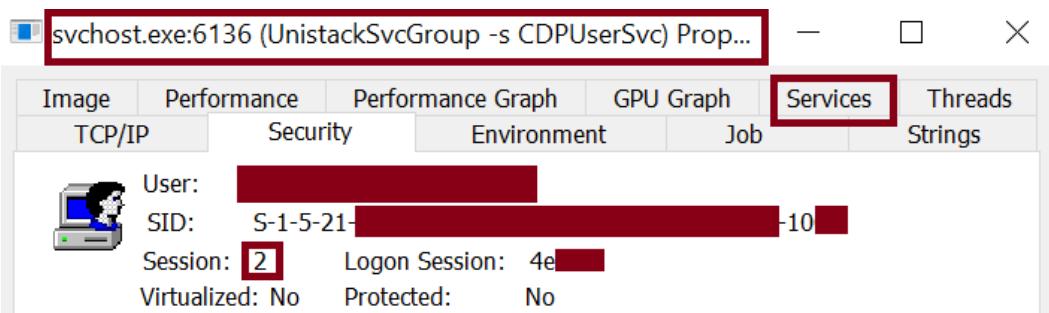
¹⁴² <https://www.cnblogs.com/talentzemin/p/16845794.html>

Per-User Windows Services

As opposed to standard Windows services¹⁴³ per-user windows services are created when a user signs on to the system. In case the user signs out the services are stopped and deleted. Thus, it provides better resource management. Also, those types of services are executed with the security context of the logged on user account and not built-in users (like LocalSystem, NetworkService and LocalService). Also, they run within the session¹⁴⁴ of the user (as shown below) and not session “0” (like normal services)¹⁴⁵.

Overall, per-user Windows services is relevant for editions of Windows servers only if “Desktop Experience”¹⁴⁶ (GUI and all tools, including client experience features) is installed. The OS displays the per-user services using the following format with the service name: <ServiceName>_<LUID> like: "CaptureService_2f53ce4", "ConsentUxUserService_2f53ce4" and "DevicePickerUserService_2f53ce4"¹⁴⁷. LUID¹⁴⁸ is a locally unique identifier for the user context.

Lastly, examples of per-user Windows services are (but not limited to): “Agent Activation Runtime” (AarSvc), “Bluetooth User Support Service” (BluetoothUserService), “Clipboard User Service” (cbdhsvc), “Connected Devices Platform User Service” (CDPUserService), “Devices Flow” (DevicesFlowUserService) , “Clipboard User Service” (cbdhsvc), “Pen Service” (PenService), and “Web Threat Defense User Service” (webthreatdefusersvc). Of course those services can change between operating system versions. As with standard Windows services per-user services the configuration is also stored in the following registry location: “HKLM\System\CurrentControlSet\Services\<ServiceName>”¹⁴⁹.



¹⁴³ <https://medium.com/@boutnaru/windows-services-part-2-7e2bdab5bce4>

¹⁴⁴ <https://medium.com/@boutnaru/windows-sessions-fd5330911d54>

¹⁴⁵ <https://helgeklein.com/blog/per-user-services-in-windows-info-and-configuration/>

¹⁴⁶ <https://learn.microsoft.com/en-us/windows-server/get-started/install-options-server-core-desktop-experience>

¹⁴⁷ <https://learn.microsoft.com/en-us/answers/questions/401557/per-user-services>

¹⁴⁸ <https://medium.com/@boutnaru/the-windows-concept-journey-luid-locally-unique-identifier-8e96aca9b1f1>

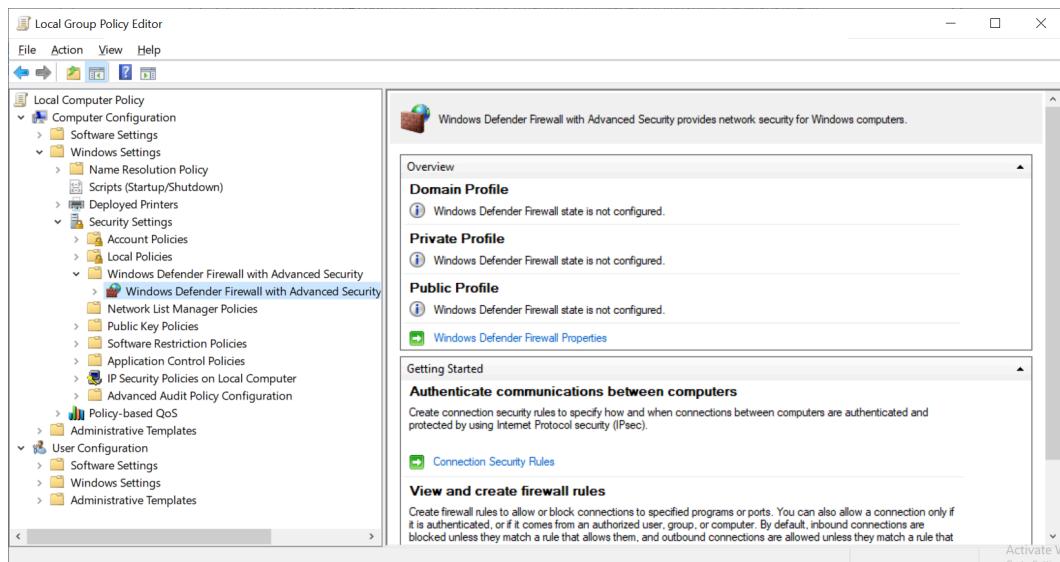
¹⁴⁹ <https://learn.microsoft.com/en-us/windows/application-management/per-user-services-in-windows>

Windows Defender Firewall

The “Windows Defender Firewall” (aka “Windows Firewall”) is available as part of the Windows operating system is used in helping to prevent malicious software/hackers from gaining access to a workstation/server through the Internet\other networks to which the system is connected¹⁵⁰. The firewall component was first included with Windows XP SP2\Windows 2003 Server SP1. Also, before Windows XP it was called “Internet Connection Firewall”. We can manage the firewall using the local/domain group policy - as shown in the screenshot below. We can also leverage the GUI (firewall.cpl), “netsh.exe” or specific COM objects.

Moreover, each network interface has one of three profiles activated automatically: “Public”, “Private” and “Domain”. The “Public” profile is the most restrictive one and it assumes the interface has been connected to a shared network. The “Private” profile assumes the interface has been connected to a network isolated from the Internet, by the way only an administrator can set a network to this profile. The “Domain” is the least restrictive and is set when an interface is connected to a network with a domain trusted by the local computer¹⁵¹. By the way, we can also enable the firewall to log dropped packets and/or successful connections¹⁵².

Lastly, it managed using the “Windows Defender Firewall” Windows service (MpsSvc) which is executed with the permissions/privileges of the “Local Service” user¹⁵³. The service is dependent both on the “Base Filtering Engine” (%windir%\System32\BFE.DLL) service and the "Windows Defender Firewall Authorization" driver (%windir%\system32\drivers\mpsdrv.sys).



¹⁵⁰ <https://learn.microsoft.com/en-us/mem/intune/user-help/you-need-to-enable-defender-firewall-windows>

¹⁵¹ https://en.wikipedia.org/wiki/Windows_Firewall

¹⁵² <https://learn.microsoft.com/en-us/windows/security/operating-system-security/network-security/windows-firewall/configure-logging?tabs=intune>

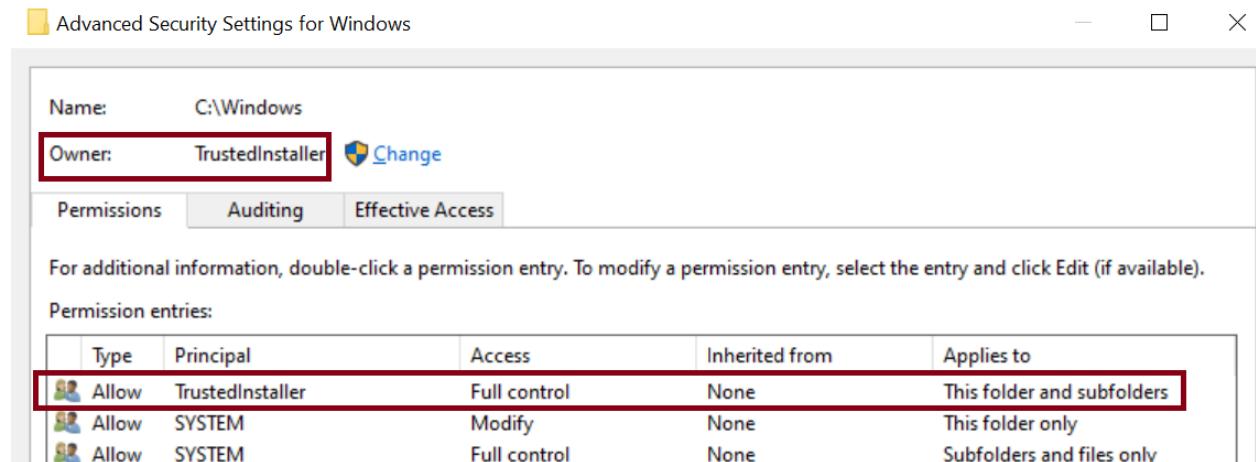
¹⁵³ <https://medium.com/@boutnaru/the-windows-security-journey-local-service-nt-authority-local-service-b1a624472931>

TrustedInstaller

Since Windows 2008/Windows 7 most of the operating system files are owned by the “Trusted Installer” entity and thus in the ACL we can see the SID of that security principal¹⁵⁴ - as shown in the screenshot below. The reason of doing that is to prevent a process running under the “Administrator”/“Local System” account from replacing/altering operating system files¹⁵⁵.

Overall, the “TrustedInstaller” is a hidden built-in user of the operating system. It is used in the process of installing/removing updates and other Windows components¹⁵⁶. By the way, the SID of the “NT Service\TrustedInstaller” user is “S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464”¹⁵⁷.

Lastly, in some cases “Trusted Installer” can be abbreviated as “TI”¹⁵⁸. Also, there is a binary with the same name “%windir%\servicing\TrustedInstaller.exe”, the description of the binary is “Windows Modules Installer” - more on that in a future writeup.



¹⁵⁴ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

¹⁵⁵ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>

¹⁵⁶ <https://softwarekeep.com/blogs/what-is/what-is-trustedinstaller-and-should-i-remove-it-from-windows-10>

¹⁵⁷ <https://learn.microsoft.com/en-us/archive/msdn-magazine/2008/november/access-control-understanding-windows-file-and-registry-permissions>

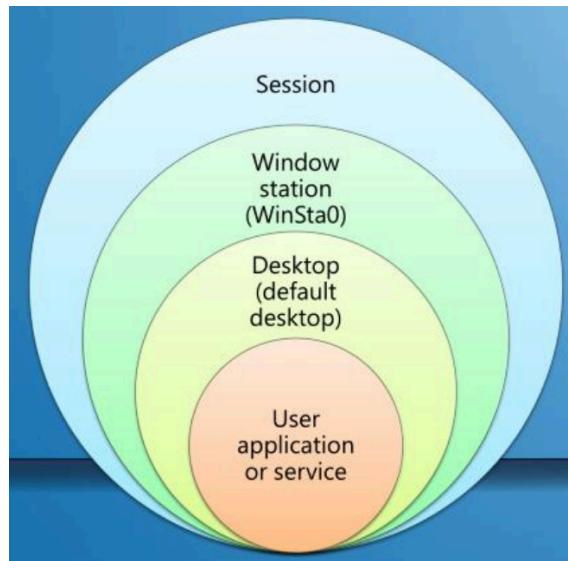
¹⁵⁸ <https://renenvfenegger.ch/notes/Windows/security/SID/S-1-5-80/956008885-3418522649-1831038044-1853292631-2271478464>

Windows Sessions

Every process started on a system belongs to a specific user maintained by an access token¹⁵⁹ and every process also belongs to a specific session. Each Windows session hosts processes, windows, Window Stations, Desktops and other resources¹⁶⁰. A diagram that demonstrates that is shown below¹⁶¹.

Also, we can create a process in another session using the Win32 API function “CreateProcessAsUser”¹⁶². We use the API call “SetTokenInformation”¹⁶³ while setting the “TokenInformationClass” with “TokenSessionId”¹⁶⁴. The privilege¹⁶⁵ named SeTcbPrivilege (Act as part of the operating system) is needed for launching a process in a different session¹⁶⁶.

Lastly, its session has its own ID aka “Session ID”. The first user that performs login is connected to session 1, the next is session 2 and so on¹⁶⁷. By the way, session 0 is saved for services but I am going to elaborate about it in a different writeup. As part of the creation flow of a session an instance of “smss.exe”¹⁶⁸ is started.



¹⁵⁹ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

¹⁶⁰ <https://brianbondy.com/blog/100/understanding-windows-at-a-deeper-level-sessions-window-stations-and-desktops>

¹⁶¹ <https://www.slideserve.com/ellery/windows-7-training>

¹⁶² <https://learn.microsoft.com/en-us/windows/win32/api/processsthreadsapi/nf-processsthreadsapi-createprocessasuser>

¹⁶³ <https://learn.microsoft.com/en-us/windows/win32/api/securitybaseapi/nf-securitybaseapi-settokeninformation>

¹⁶⁴ https://learn.microsoft.com/en-us/windows/win32/api/winnt/ne-winnt-token_information_class

¹⁶⁵ <https://medium.com/@boutnaru/windows-security-privileges-b8fe18cf3d5a>

¹⁶⁶ <https://stackoverflow.com/questions/3128017/launching-a-process-in-user-s-session-from-a-service>

¹⁶⁷ <https://renenyffenegger.ch/notes/Windows/development/session/index>

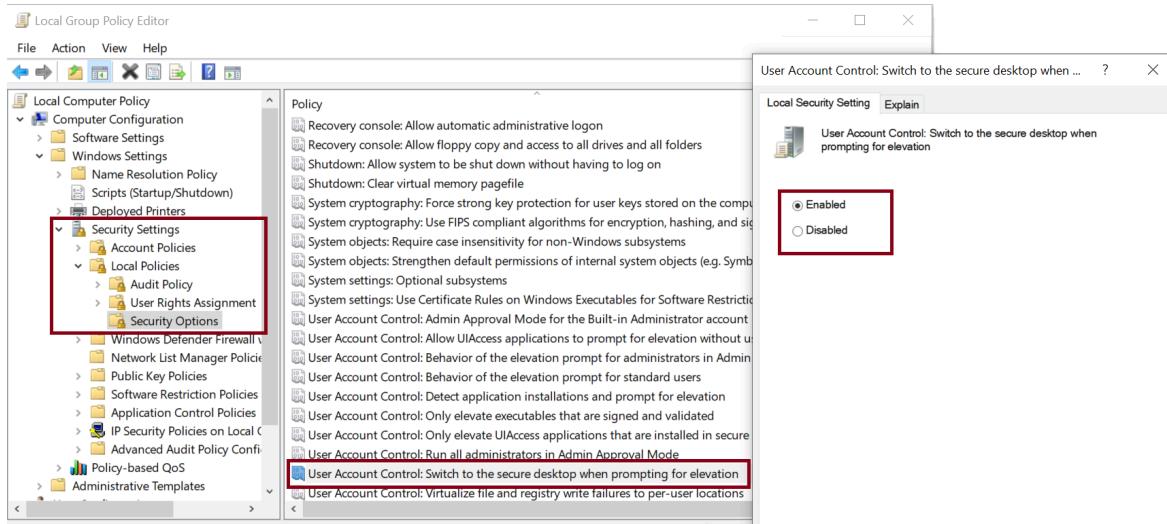
¹⁶⁸ <https://medium.com/@boutnaru/the-windows-process-journey-smss-exe-session-manager-subsystem-bca2cf748d33>

Secure Desktop

The “Secure Desktop” is used for displaying the sign-in UI and thus restricts the access and functionality until the sign-in is performed. The main difference between the “Secure Desktop” and the user’s desktop is that only trusted processes running as SYSTEM are allowed to execute in the “Secure Desktop”. It can also be used when prompting for elevation¹⁶⁹.

Overall, by using the “Secure Desktop” we can protect against input/output spoofing by presenting a credentials dialog or sending keys to it, which is required by UAC¹⁷⁰ to reduce the risk of malware by limiting the ability of malicious code from running with administrator permissions¹⁷¹.

Thus, the consent prompt¹⁷² and the credentials prompt by default are displayed on the “Secure Desktop”. The “Secure Desktop” dims the user desktop and shows the elevation prompt which must be responded before continuing. After the user selects Yes/No the desktop is switched back to the user’s desktop from the “Secure Desktop”¹⁷³. Lastly, since Windows 2019 it is not possible to paste the content of the clipboard¹⁷⁴ on the “Secure Desktop”. The “Secure Desktop” can be disabled/enabled using domain/local group policy¹⁷⁵.



¹⁶⁹ <https://tinyurl.com/2nuf65wt>

¹⁷⁰ <https://medium.com/@boutnaru/the-windows-security-journey-uac-user-account-control-ce395df5c784>

¹⁷¹ <https://security.stackexchange.com/questions/3759/how-does-the-windows-secure-desktop-mode-work>

¹⁷² <https://medium.com/@boutnaru/the-windows-process-journey-consent-exe-consent-ui-for-administrative-applications-d8e6976e8e40>

¹⁷³ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/user-account-control/how-it-works>

¹⁷⁴ <https://medium.com/@boutnaru/windows-clipboard-c63e369d35d9>

¹⁷⁵ <https://answers.microsoft.com/en-us/windows/forum/all/disable-secure-desktop-without-changing-uac-level/7c2522df-7beb-49d7-8793-e762facbd252>

MSRC (Microsoft Security Response Center)

MSRC (Microsoft Security Response Center) is part of the defender community. For more than 20 years MSRC has engaged with security research in order to protect Microsoft's customers¹⁷⁶. Thus, MSRC's mission is defined as protecting customers and Microsoft from current and emerging threats related to security and privacy¹⁷⁷.

Overall, MSRC also provides a bug bounty program. Among the bounty program we can find: cloud programs ("Microsoft Azure", "Microsoft Identity", "Xbox", "M365", "Microsoft Azure DevOps Services", "Microsoft Dynamics 365 and Power Platform", Microsoft AI"and "Microsoft Defender"), platform programs ("Microsoft Hyper-V", "Microsoft Windows Insider Preview", "Microsoft Applications and On-Premises Servers", "Windows Defender Application Guard", "Microsoft Edge (Chromium-based)" and "Microsoft 365 Insider") and defense & grant programs ("Mitigation Bypass and Bounty for Defense", "Grant: Microsoft Identity" and "SIKE Cryptographic Challenge"). Each program has its own scope, however they share the same high level requirements - as shown below¹⁷⁸.

Lastly, MSRC investigates all reports of security vulnerabilities affecting Microsoft products and services. Also, MSRC provides the information here as part of the ongoing effort to help manage security risks and help keep systems protected¹⁷⁹. By the way, there is a GitHub repository which stores security research conducted by MSRC¹⁸⁰.



¹⁷⁶ <https://www.microsoft.com/en-us/msrc>

¹⁷⁷ <https://www.microsoft.com/en-us/msrc/mission?rte=1>

¹⁷⁸ <https://www.microsoft.com/en-us/msrc/bounty>

¹⁷⁹ <https://msrc.microsoft.com/update-guide/en-us>

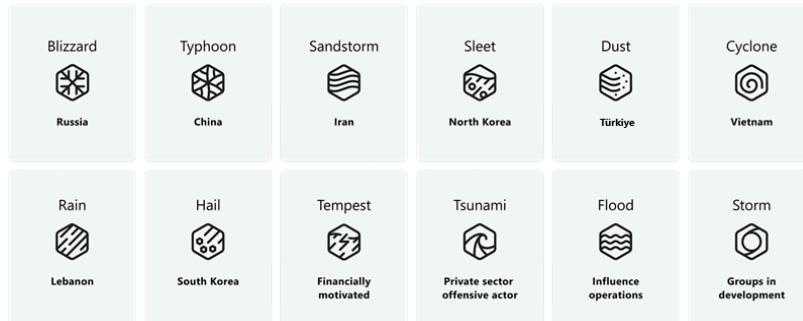
¹⁸⁰ <https://github.com/microsoft/MSRC-Security-Research>

MSTIC (Microsoft Threat Intelligence Center)

MSTIC (Microsoft Threat Intelligence Center) consistently tracks threat actor activity. Also, it works across Microsoft Security products and services in order to improve customer protections. In case Microsoft observes nation-state actor activity, customers are directly notified of being targeted\compromised. They also receive information for securing their accounts¹⁸¹.

Overall, MSTIC is one of Microsoft's prime security operations centers. MSTIC leverages vast data sets and cutting-edge analytical tools used with the widespread infrastructure of Microsoft. The core responsibilities of MSTIC includes: threat detection, incident response, threat education and security advisories. One example is the SolarWinds attack, MSTIC played a crucial role in understanding the extent of the breach, the actors behind it and more¹⁸².

Lastly, MSTIC provides unique insight on threats to protect Microsoft and our customers and is responsible for delivering timely threat intelligence across our product and services teams¹⁸³. Based on publications it seems that MSTIC has teams that are threat-focused, each responsible for a specific threat actor¹⁸⁴. Microsoft uses a naming taxonomy for actors with the theme of weather - as shown in the diagram below¹⁸⁵.



¹⁸¹ <https://www.microsoft.com/en-us/security/blog/2022/11/10/microsoft-threat-intelligence-presented-at-cyberwarcon-2022/>

¹⁸² <https://socket.dev/glossary/microsoft-threat-intelligence-center-mstic>

¹⁸³ <https://www.microsoftrnd.co.il/JobDetails?JobSeqNo=1807399>

¹⁸⁴ <https://www.technologyreview.com/2019/11/06/238375/inside-the-microsoft-team-tracking-the-worlds-most-dangerous-hackers/>

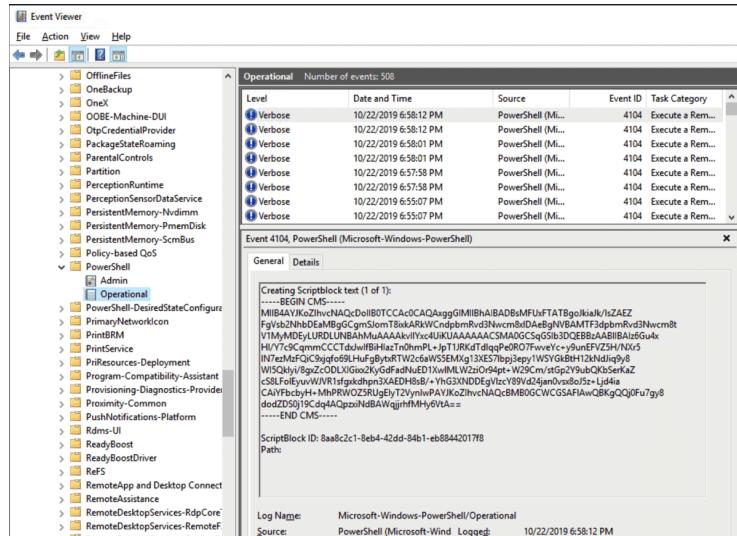
¹⁸⁵ <https://learn.microsoft.com/en-us/unified-seccoms-platform/microsoft-threat-actor-naming>

PEL (Protected Event Logging)

“Protected Event Logging” is a new security feature added in Windows 10. Its goal is to use encryption in order to protect sensitive data written to event logs. The data is encrypted using CMS (IETF Cryptographic Message Syntax) while it is written to the logs - as shown in the screenshot below¹⁸⁶. The data can be decrypted when the records are moved to a central server using “Windows Event Forwarding”¹⁸⁷.

Moreover, “Protected Event Logging” is not enabled by default on Windows. As of Windows 10 the only powershell¹⁸⁸ uses PEL as part of its advanced logging¹⁸⁹.

Lastly, the main goal of PEL is to allow performing script block logging while preserving security. This is done by encrypting the content of scripts while writing it to the event logs.



¹⁸⁶ <https://4sysops.com/archives/encrypt-event-logs-and-files-with-powershell-and-group-policies/>

¹⁸⁷ <https://petri.com/windows-10-protected-event-logging/>

¹⁸⁸ <https://medium.com/@boutnaru/the-windows-process-journey-powershell-exe-windows-powershell-36daabaa74c4>

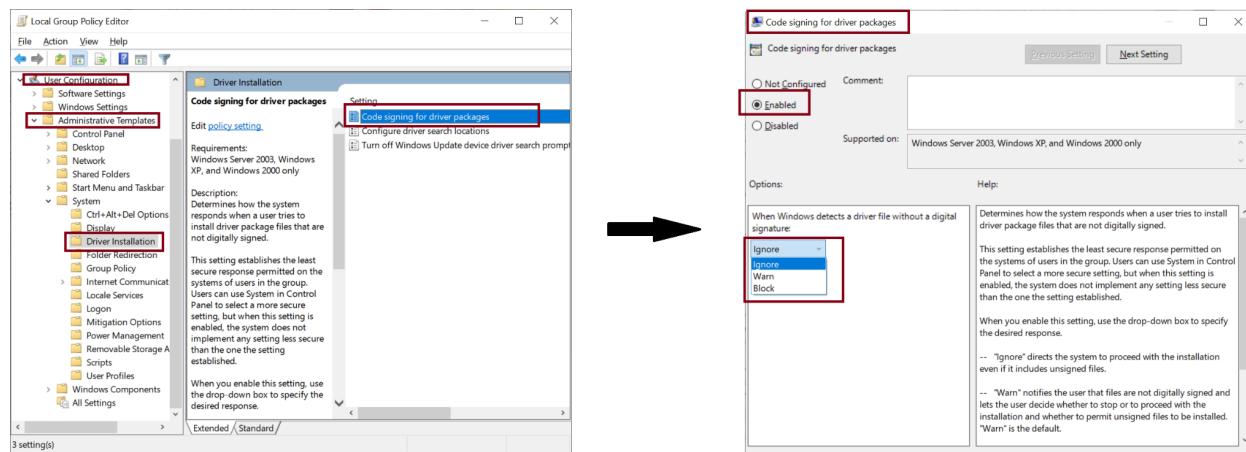
¹⁸⁹ https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_logging_windows?view=powershell-7.4

DSE (Driver Signature Enforcement)

“DSE” (Driver Signature Enforcement) is a security feature that was introduced in Windows Vista 64 bit. It ensures that only signed drivers can be loaded by the operating system kernel¹⁹⁰. It is important to know that starting with Windows 10/Windows Server 2016 kernel-mode drivers must be signed by the “Windows Hardware Dev Center Dashboard”. In order to do that we need an EV certificate - this is called the “Driver Signing Policy”¹⁹¹.

Moreover, one of the ways of bypassing this feature is by loading a vulnerable signed driver and overwriting the variable and thus disabling DSE¹⁹². By the way, from Windows 10 (1507) all drivers are signed by the “Hardware Dev Center” by leveraging the SHA2 hash function¹⁹³.

Lastly, in order to disable DSE we have a couple of choices. We can use the “bcdedit /set TESTSIGNING OFF” (using Windows test mode). Also, we can use the “Local Group Policy”/”Group Policy” for that, this can be done “User Configuration > Administrative Templates > System > Driver Installation” - as shown in the screenshot below. Finally, we can also reboot the system and go to the troubleshooting menu (pressing F7) and selecting “Disable driver signature enforcement”¹⁹⁴.



¹⁹⁰ <https://www.codeproject.com/Articles/5348168/Disable-Driver-Signature-Enforcement-with-DSE-Patc>

¹⁹¹ <https://learn.microsoft.com/en-us/windows-hardware/drivers/install/kernel-mode-code-signing-policy--windows-vista-and-later->

¹⁹² <https://github.com/hfiref0x/DSEFix>

¹⁹³ <https://learn.microsoft.com/en-us/windows-hardware/drivers/install/driver-signing>

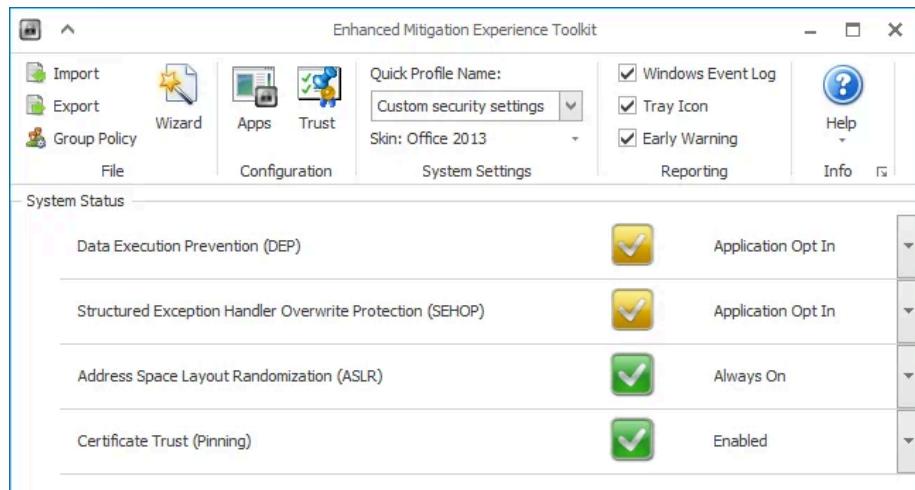
¹⁹⁴ https://www.maijorgeeks.com/content/page/how_to_disable_driver_signature_enforcement.html

EMET (Enhanced Mitigation Experience Toolkit)

EMET (Enhanced Mitigation Experience Toolkit) is a utility created by Microsoft for helping in preventing vulnerabilities in software from being successfully exploited. This is done by leveraging mitigation techniques¹⁹⁵. EMET was released in 2009 as a standalone tool from Windows 10 Microsoft has implemented many features and mitigations that can make EMET unnecessary¹⁹⁶.

Overall, by using EMET an administrator can enable different security mitigations such as: DEP¹⁹⁷ (Data Execution Prevention), ASLR¹⁹⁸ (Address Space Layout Randomization), Certificate Trust (Pinning), SEHOP (Structured Exception Handler Overwrite Protection) - as shown in the screenshot below. We can configure each mitigation as relevant for a specific application or system wide. The second which is called “Always On” is more secure¹⁹⁹.

Lastly, the biggest benefit of EMET is the ability to enable different security mitigations without the need of recompiling the binary. It is important to know that EMET is implemented using the Windows shim infrastructure called the “Application Compatibility Framework”²⁰⁰.



¹⁹⁵ <https://support.microsoft.com/en-us/topic/emet-mitigations-guidelines-b529d543-2a81-7b5a-d529-84b30e1ecee0>

¹⁹⁶ <https://msrc.microsoft.com/blog/2016/02/enhanced-mitigation-experience-toolkit-emet-version-5-5-is-now-available/>

¹⁹⁷ <https://medium.com/@boutnaru/security-nx-bit-non-executable-18759fd2802e>

¹⁹⁸ <https://medium.com/@boutnaru/security-aslr-address-space-layout-randomization-part-1-overview-3aec7fec01e0>

¹⁹⁹ <https://insights.sei.cmu.edu/blog/life-beyond-microsoft-emet/>

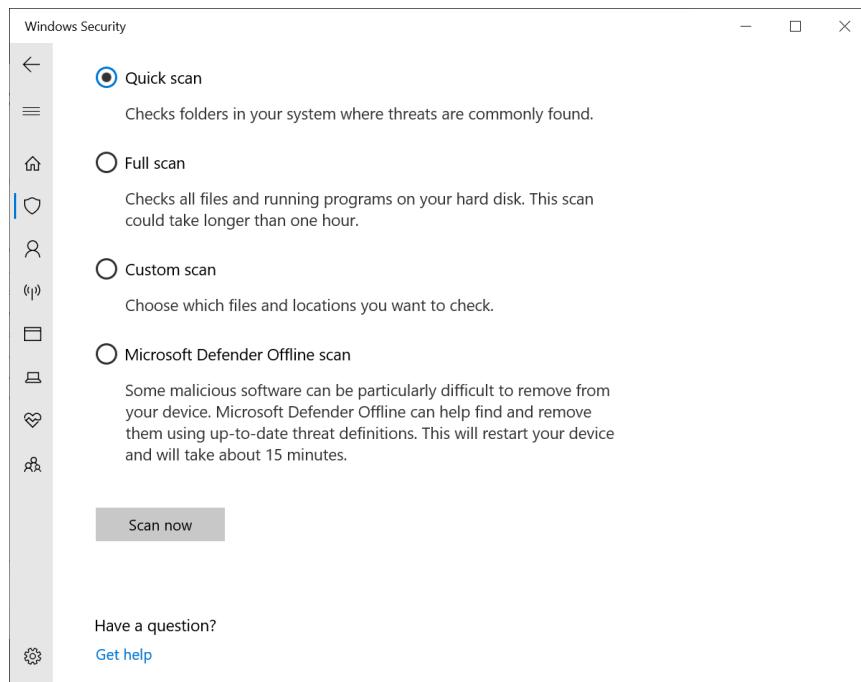
²⁰⁰ <https://techcommunity.microsoft.com/t5/configuration-manager-archive/deploying-and-configuring-the-enhanced-mitigation-experience/ba-p/273099>

Windows Defender Antivirus

“Windows Defender Antivirus” is available for Windows 10/11/Server. It uses advanced techniques for protecting from known/unknown malware. It does that by monitoring suspicious activities and blocking threats based on behavior. This is achieved through machine learning/anomaly detection (until 2015 it used only statically signatures), cloud-based intelligence, and real-time updates. “Windows Defender Antivirus” works both online and offline²⁰¹ - as shown in the screenshot below. By the way, it is also called MDA (Microsoft Defender AntiVirus).

Overall, we can think about “Windows Defender Antivirus” as a next-generation protection solution that comes with Windows. Microsoft Defender Antivirus is real-time, always-on antivirus protection²⁰². By the way, there is also a version of “Microsoft Defender: Antivirus” for Android²⁰³, however it is out of the scope of the current writeup.

Lastly, since 2021 “Windows Defender Antivirus” is part of a larger Microsoft Defender brand. Among that brand are included: “Microsoft Defender Vulnerability Management”, “Microsoft Defender for Cloud Apps”, “Microsoft Defender for Identity”, “Microsoft Defender Endpoint” and more²⁰⁴.



²⁰¹ <https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-antivirus-windows?view=o365-worldwide>

²⁰² <https://www.microsoft.com/en-us/windows/comprehensive-security>

²⁰³ <https://play.google.com/store/apps/details?id=com.microsoft.scmx&hl=en&gl=US>

²⁰⁴ https://en.wikipedia.org/wiki/Microsoft_Defender_Antivirus

Windows Defender SmartScreen

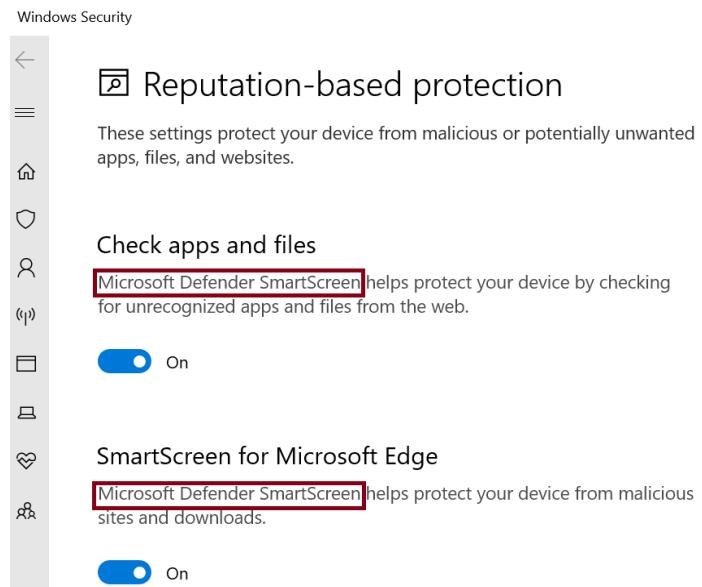
Windows SmartScreen is a cloud-based anti malware/phishing solution included in different Microsoft products (Windows/Internet Explorer/Microsoft Edge). The SmartScreen intelligence is also in use as part of Microsoft's online services backend (like Outlook.com/Bing search engine).

Since Windows 10 the setting of SmartScreen is part of “Windows Defender Security Center”²⁰⁵ - as shown in the screenshot below

Thus, Windows Defender SmartScreen is a filtering mechanism which verifies files/websites in order to prevent malicious activities on the system. The different features which are provided by SmartScreen are: protection from unauthorized applications (checks the publisher’s signature), keeping the user safe from phishing/malware attacks (based on a DB of malicious websites), automatic blocking of malicious URLs, reputation checks and safe downloading²⁰⁶.

Moreover, SmartScreen has different benefits such as the operating system integration, improved heuristics and diagnostic data and management using Microsoft Intune and group policy. It is important to understand it does not protect against malicious files on internal locations or network shares such as shared folders with UNC paths or SMB/CIFS²⁰⁷.

Lastly, in case we think there is a false positive/negative we can submit the specific file to Microsoft for further analysis²⁰⁸.



²⁰⁵ https://en.wikipedia.org/wiki/Microsoft_SmartScreen

²⁰⁶ <https://signmycode.com/blog/what-is-windows-defender-smartscreen>

²⁰⁷ <https://learn.microsoft.com/en-us/windows/security/operating-system-security/virus-and-threat-protection/microsoft-defender-smartscreen/>

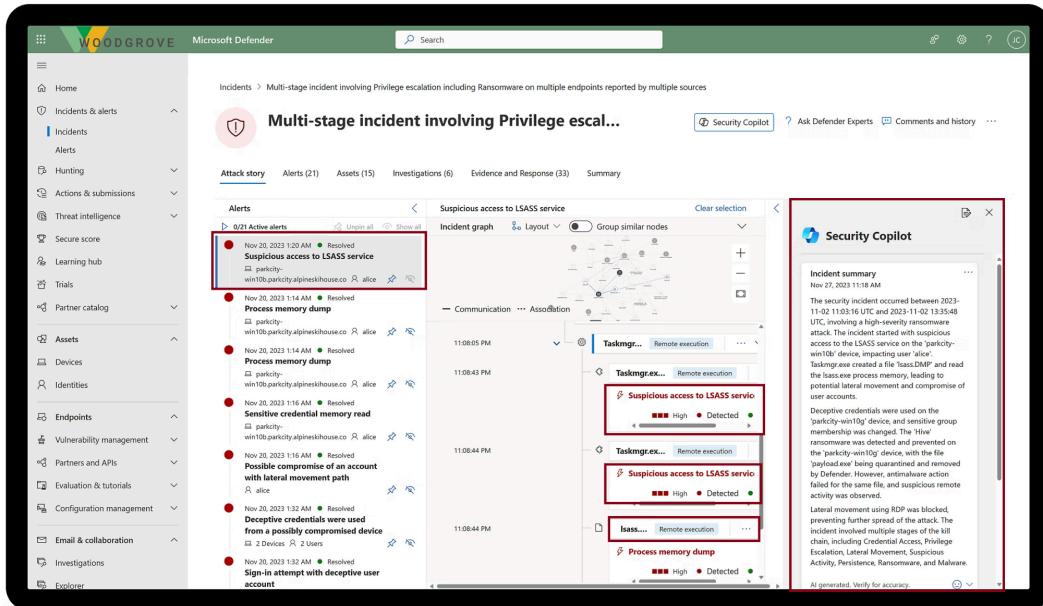
²⁰⁸ <https://www.microsoft.com/wdsi/filesubmission/>

MSDE (Microsoft Defender for Endpoint)

MSDE (Microsoft Defender for Endpoint) is an enterprise endpoint security solution\platform. It is used for helping enterprises prevent\detect\investigate\respond to advanced threats. “Defender for Endpoint” leverages technologies which are built into Windows 10+ and/or Microsoft’s cloud services. Among those technologies are: threat intelligence, cloud security analytics and endpoint behavioral sensors. Those provide users: attack surface reduction, next generation protection and more²⁰⁹.

Overall, “Defender for Endpoint” has the following key capabilities: auto-deployed deception (automatically generate and disperse deception techniques at scale to expose cyberattackers), copilot for security (security-specific generative AI to rapidly investigate and respond, prioritize alerts, and learn new skills - as shown in the screenshot below), network detection and response, simplified endpoint management and more²¹⁰.

Lastly, “Microsoft Defender for Endpoint” supports not only Windows as it also supports Linux²¹¹, Android²¹², macOS²¹³ and iOS²¹⁴.



²⁰⁹ <https://learn.microsoft.com/en-us/defender-endpoint/microsoft-defender-endpoint>

²¹⁰ <https://www.microsoft.com/en-us/security/business/endpoint-security/microsoft-defender-endpoint>

²¹¹ <https://learn.microsoft.com/en-us/defender-endpoint/linux-whatsnew>

²¹² <https://learn.microsoft.com/en-us/defender-endpoint/android-whatsnew>

²¹³ <https://learn.microsoft.com/en-us/defender-endpoint/mac-whatsnew>

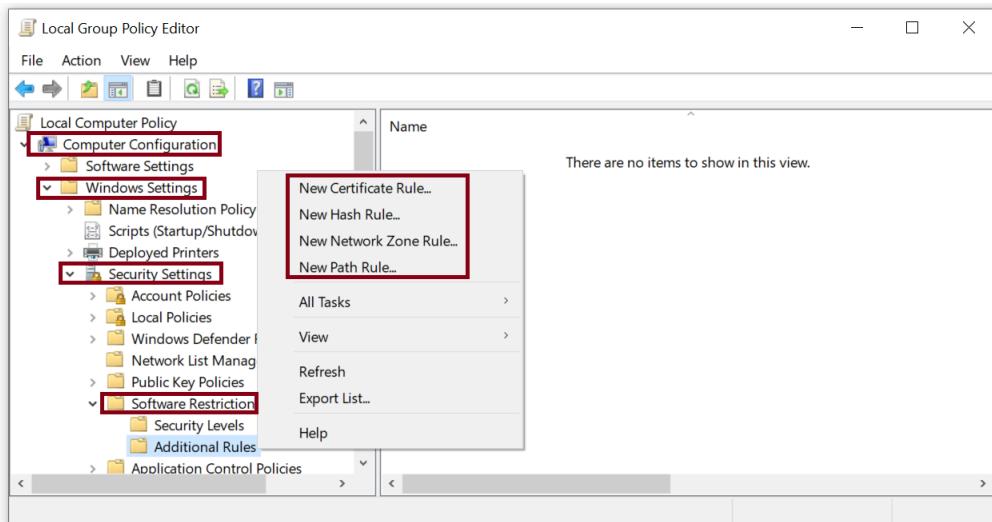
²¹⁴ <https://learn.microsoft.com/en-us/defender-endpoint/ios-whatsnew>

SRP (Software Restriction Policies)

SRP (“Software Restrictions Policies”) is a security feature available in Windows that we can configure using “Group Policy”/“Local Security Policy”. The main goal of SRP is to allow administrators to control which applications/programs are allowed to run on a Windows based device. We can set different restriction policies using SRP like: which programs (executables) can run, who can add trusted publishers, configure if the policies affect all users or just specific ones, preventing specific executable files from being executed based on OU, domain and more²¹⁵.

Moreover, SRP has been supported since Windows XP\Windows Server 2003. We can think about SRP as “Trust Policies” that can restrict the execution of scripts and other code (think about macros, ActiveX controls and more) from running. Starting from Windows 7/Windows 2008 R2 we should use “AppLocker” instead of SRP. From Windows 10 we can also use WDAC²¹⁶.

Lastly, each SRP (Software Restriction Policy) rule can have one of three security levels²¹⁷: “Disallowed” (software will not run), “Basic User” (programs can execute as a user without Administrator access rights) and “Unrestricted” (the access rights are determined by the access rights of the user). Overall, we have four types of SRP rules that we can define: “Certificate Rule”, “Hash Rule”, “Network Zone Rule” and “Path Rule” - as shown in the screenshot below. For each of them one security level is defined as an action.



²¹⁵ <https://learn.microsoft.com/en-us/windows-server/identity/software-restriction-policies/software-restriction-policies>

²¹⁶ <https://learn.microsoft.com/en-us/windows-server/identity/software-restriction-policies/software-restriction-policies-technical-overview>

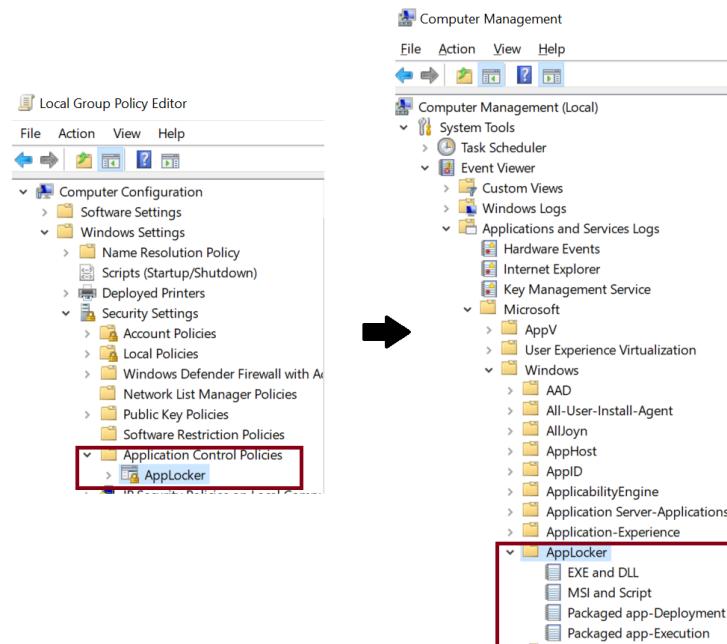
²¹⁷ <https://www.adamcouch.co.uk/software-restriction-policies/>

AppLocker (Application Locking)

“AppLocker” is a security feature introduced in Windows 7. The goal of “AppLocker” is to allow a system administrator to configure (control) which applications are allowed to be executed on a Windows system. Thus, preventing users from executing unwanted/unapproved applications. AppLocker’s settings can apply for all users of the computer or just specific users/groups²¹⁸. The configuration can be done using “GPO”\”Local Group Policy”->”Computer Configuration”->”Security Settings”->”Application Control Policies”->”AppLocker” - as shown in the screenshot below.

Moreover, AppLocker’s rules can be configured based on: the path from which the app/script is launched, attributes of the code signing certificate used for signing the binary, metadata of the file like the original file name, the hash of the file and more. The rules can target different file formats such as: scripts (“*.ps1”, “*.bat”, *.cmd”, “*.vbs” and “*.js”), executables (“*.exe” and “*.com”), libraries (“*.dll” and “*.ocx”), Windows installer files (“*.msi”, “*.msp” and “*.mst”) and packaged apps\packaged app installers (“*.appx”)²¹⁹.

Lastly, it is important to understand that the AppLocker rules can apply for any PE file²²⁰ regardless of the extension. Also, in case a DLL rule is configured we need to create an allow rule that covers every DLL used by all allowed apps. In case of a violation relevant entries are logged to the Windows eventlog²²¹ - as shown in the screenshot below.



²¹⁸ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/wdac-and-applocker-overview>

²¹⁹ <https://tinyurl.com/mrxn94t>

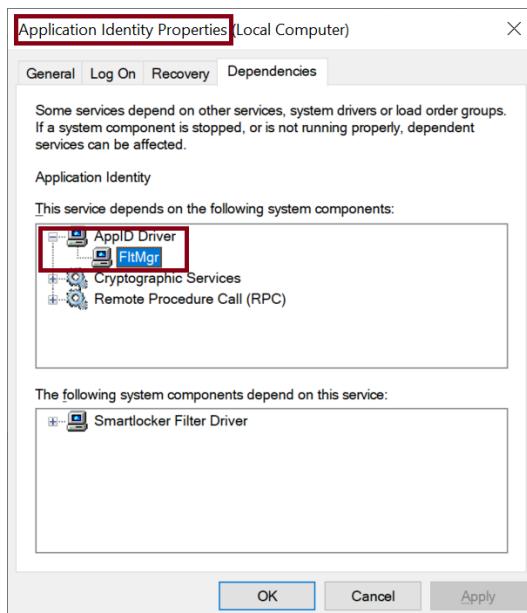
²²⁰ <https://medium.com/@boutnaru/the-portable-executable-journey-file-header-struct-image-file-header-00360271f147>

²²¹ <https://medium.com/@boutnaru/the-windows-concept-journey-windows-event-logs-a9945bca421f>

AppIDSvc (Application Identity Service)

AppIDSvc (Application Identity Service) is a Windows service hosted by “svchost.exe”²²². The description of the service states it is used for determining and verifying the identity of an application. Thus, if we disable this service it will prevent AppLocker²²³ from being enforced. Moreover, the service is launched using the access permissions of the “Local Service” user²²⁴. Also, the process of the service is defined with the protection level of “PsProtectedSignerWindows-Light”, this is also known as “Protected Process Light” (PPL). It is important to understand that the actual protection level is a combination of “Type” and “Signer”²²⁵.

Lastly, the “Application Identity Service” is dependent on: RPC (“Remote Procedure Call”), “Cryptographic Services” (think about verifying digital signatures of executables) and the “AppID Driver” (%windir%\system32\drivers\appid.sys), which itself is dependent on “FltMgr” (%windir%\system32\drivers\filtmgr.sys) - as shown in the screenshot below. The user-mode part of the service is implemented as part of “%windir%\System32\appidsvc.dll”, which is digitally signed by Microsoft.



²²² <https://medium.com/@boutnaru/the-windows-process-journey-svchost-exe-host-process-for-windows-services-b18c65f7073f>

²²³ <https://medium.com/@boutnaru/the-windows-security-journey-applocker-application-locking-b9547fb9cbbd>

²²⁴ <https://medium.com/@boutnaru/the-windows-security-journey-local-service-nt-authority-local-service-b1a624472931>

²²⁵ <https://uberagent.com/blog/uberagent-7-preview-edr-antivirus-windows-defender-protected-process-performance-monitoring/>

Differences between AppLocker and SRP

In general, both SRP²²⁶ and Applocker²²⁷ are built in security features of the Windows operating system used for application control/whitelisting in order to increase the security posture of a Windows based device. There are some differences between the two, part of those differences are documented in this writeup - an example of a table comparing the both is shown below²²⁸.

Overall, SRP is relevant since Windows XP/Windows Server 2003 while AppLocker is relevant since Windows 7/Window Server 2008 R2. It is important to know that AppLocker has an audit mode (which allows testing the configuration in production) and rule exceptions while SRP does not. Both SRP and AppLocker “Windows Installer” files, however only AppLocker package apps and installers. SRP has the ability to set different security levels for approved applications (like as limited/basic user) this is not supported by AppLocker²²⁹.

Lastly, SRP’s enforcement is done in user-mode while AppLocker’s enforcement occurs in kernel-mode. Moreover, AppLocker supports exporting importing policies, rule collections, PowerShell support and the ability to define custom error messages. The SRP policies are relevant for all users (per GPO) while in AppLocker we can specify specific users/groups per rule²³⁰.

Feature	SRP (Windows XP & W2K3)	AppLocker
Rule scope	Specific user or group (per GPO)	Specific user or group (per rule)
Rule conditions provided	File hash, path, certificate, registry path, and Internet zone rules	File hash, path, and publisher rules
Rule types provided	Allow and deny	Allow and deny
Default rule action	Allow and deny	Deny
Audit-only mode	No	Yes
Wizard to create multiple rules at one time	No	Yes
Policy import or export	No	Yes
Rule collection	No	Yes
PowerShell support	No	Yes
Custom error messages	No	Yes

²²⁶ <https://medium.com/@boutnaru/the-windows-security-journey-srp-software-restriction-policies-9f658a4ed648>

²²⁷ <https://medium.com/@boutnaru/the-windows-security-journey-applocker-application-locking-b9547fb9cbdb>

²²⁸ <https://www.cievo.sk/2012/11/30/quickie-software-restriction-policy-vs-applocker/>

²²⁹ <https://learn.microsoft.com/en-us/windows-server/identity/software-restriction-policies/software-restriction-policies-technical-overview>

²³⁰ <https://slideplayer.com/slide/11890654/>

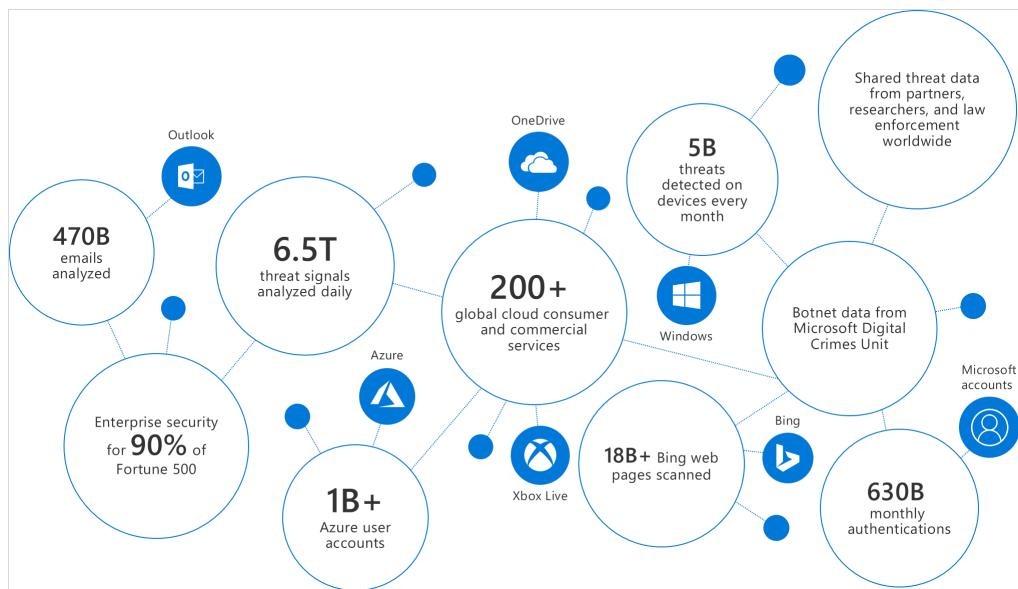
Microsoft ISG (Microsoft Intelligent Security Graph)

Microsoft ISG (Microsoft Intelligent Security Graph) is a collection of data (like security intelligence), people, and machine learning used to efficiently and effectively protect and strengthen Microsoft products and services²³¹.

Overall, as developers we can consume data from ISG using a REST API (Microsoft Security Graph API). Using the API we can retrieve different events in various categories such as: “Anonymous IP Risks”, “Leaked Credentials Risks”, “Suspicious IP Risks”, “Malware Risks”, “Impossible Travel Risks” and “Unfamiliar Location Risks”²³².

Thus, we can use the “Microsoft Graph Security API” in order to build security applications that can: provide visibility and enable proactive risk management, respond to new security threats, automate security actions, provide contextual data, enrichment information and more²³³.

Lastly, ISG pulls data from Microsoft endpoints (like Windows/Office 365), services and from companies part of the Microsoft Intelligent Security Association. Examples are the founding members like “Anomoli”, “PwC” and “Palo Alto Networks” which add signals to the graph²³⁴. For better understanding the data included in the graph you can check out the diagram below, which is relevant for 2019²³⁵.



²³¹ <https://www.linkedin.com/pulse/microsoft-intelligent-security-graph-peter-ginnegar>

²³² <https://www.sharepointeurope.com/consuming-microsoft-intelligent-security-graph-custom-developed-solutions/>

²³³ <https://github.com/microsoftgraph/microsoft-graph-docs-contrib/blob/main/api-reference/beta/resources/security-api-overview.md>

²³⁴ <https://www.forbes.com/sites/patrickmoorhead/2018/04/17/microsoft-announces-iot-security-silicon-and-platform-in-its-march-to-deliver-end-to-end-security/>

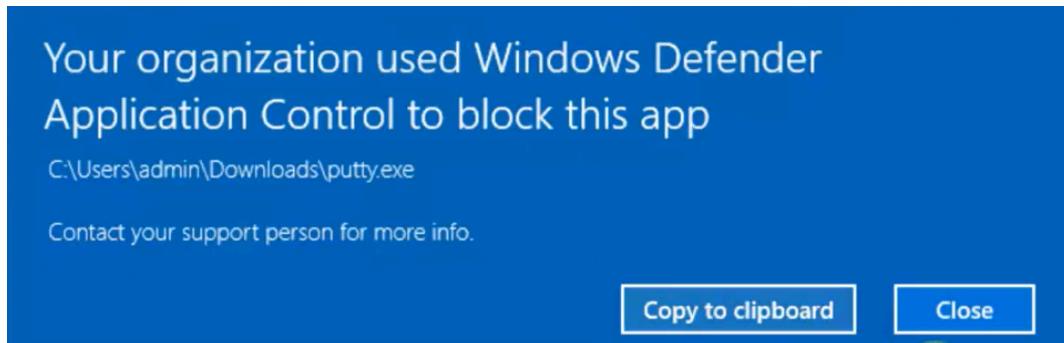
²³⁵ <https://web.archive.org/web/20190529062503/https://www.microsoft.com/security/blog/2019/05/14/executing-vision-microsoft-threat-protection/>

WDAC (Windows Defender Application Control)

WDAC (Windows Defender Application Control) is a security feature which was introduced as part of Windows 10. We can use WDAC for controlling which applications/drivers are allowed (or blocked) to be executed/loaded on a Windows base device - as shown in the screenshot below²³⁶. We can think about it as a replacement/enhancement of “Software Restriction Policy”²³⁷ and/or AppLocker²³⁸.

Overall, the configuration of WDAC is relevant and affects all users connected to the system. Also, when it was released it was part of “Device Guard” under the name of “Configurable Code Integrity”. WDAC supports defining rules based on different attributes such as: the reputation of the application (based on Microsoft’s Intelligent security graph), the process which launched the binary/application, the path from which the application was launched, signed metadata of the binary (like the original file name), the identity of the process which started the installation flow of the application, attributes of the code-signing certificates used to signed the application²³⁹.

Lastly, WDAC can be configured using different mechanisms/tools like: “Group Policy”, “Script”, “Microsoft Configuration Manager” or “Microsoft Intune”²⁴⁰. IT professionals can create powerful WDAC policies for deployment using “The WDAC Policy Wizard” tool created by Microsoft²⁴¹.



²³⁶ <https://blog.ciaops.com/category/windows/>

²³⁷ <https://medium.com/@boutnaru/the-windows-security-journey-srp-software-restriction-policies-9f658a4ed648>

²³⁸ <https://medium.com/@boutnaru/the-windows-security-journey-applocker-application-locking-b9547fb9cbdb>

²³⁹ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/wdac-and-applocker-overview>

²⁴⁰ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/feature-availability>

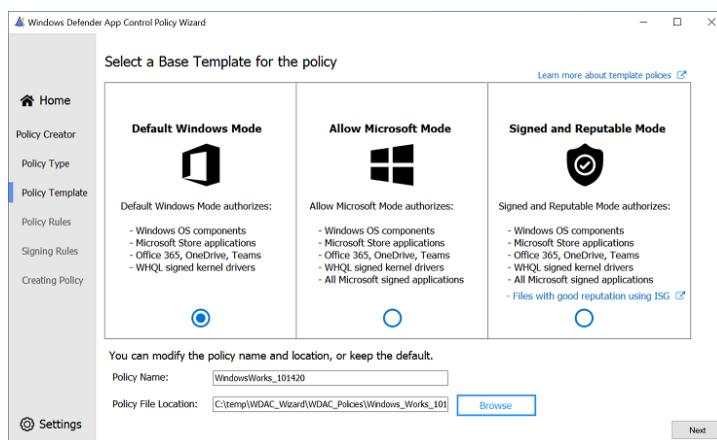
²⁴¹ <https://github.com/MicrosoftDocs/WDAC-Toolkit>

Differences between WDAC and AppLocker

In general, both AppLocker²⁴² and WDAC²⁴³ are built in security features of the Windows operating system used for application control/whitelisting in order to increase the security posture of a Windows based device. There are some differences between the two, part of those differences are documented in this writeup.

Overall, AppLocker is supported since Windows 8 while WDAC is available for Windows 10/Windows Server 2016. There are a couple of features which are only supported by WDAC and not AppLocker such as: kernel mode policies, per app rules, reputation based intelligence, COM object whitelisting, application ID tagging, packaged app rules²⁴⁴ - more on those in future writeups. In the case of WDAC it is recommended to start with a template policy and remove/add rules on top of it. WDAC wizard provides three basic policy templates - as shown in the screenshot below²⁴⁵.

Lastly, WDAC is suitable for a highly secured environment. As opposed to AppLocker in WDAC, administrators can be excluded by rules for executing specific applications. Think about a case in which we have not allows the execution of an installer, even an admin can't uninstall the application²⁴⁶. Thus, if we want to enforce different policies for users/groups on a shared device or we don't want to set application control rules on DLLs/drivers we should used AppLocker and not WDAC²⁴⁷.



²⁴² <https://medium.com/@boutnaru/the-windows-security-journey-applocker-application-locking-b9547fb9cbbd>

²⁴³ <https://medium.com/@boutnaru/the-windows-security-journey-wdac-windows-defender-application-control-26955abe4c01>

²⁴⁴ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/feature-availability>

²⁴⁵ <https://tinyurl.com/cxdb82e>

²⁴⁶ https://www.reddit.com/r/Intune/comments/lapqjip/applocker_vs_wdac/

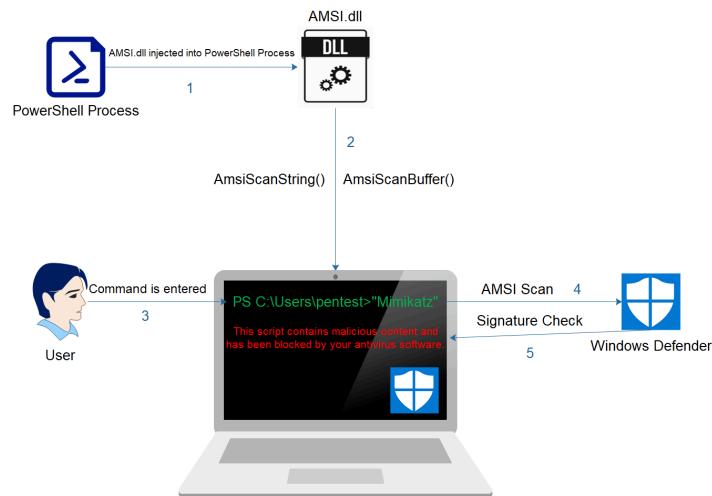
²⁴⁷ <https://learn.microsoft.com/en-us/windows/security/application-security/application-control/windows-defender-application-control/wdac-and-applocker-overview>

AMSI (Anti-Malware Scan Interface)

“AMSI” (Anti-Malware Scan Interface) is used by “Microsoft Defender for Endpoint” in order to enhance protection against fileless malware, dynamic script-based attacks, and other nontraditional cyber threats. AMSI supports different scripting languages such as: VBScript, JScript, PowerShell, WMI (Windows Management Instrumentation), .NET 4.8 assemblies scanning and more²⁴⁸.

Moreover, AMSI is used by major antivirus engines (like BitDefender, ESET and Kaspersky) for evaluating the content for malware and detecting possible threats. Also, 3-rd party AVs can replace the default AMSI engine, thus every AMSI call ends up invoking the third-party engine. However, in case there is not a 3-party aware AMSI engine the security evaluation requested over AMSI is provided by Defender²⁴⁹.

Lastly, when a script is executed (like using PowerShell) the “%windir%\System32\amsi.dll” (or “%windir%\SysWOW64\amsi.dll” in case of a 32-bit process) is loaded into the memory address space of the process (powershell.exe in our example). Before the script is executed the two APIs are called by the relevant AV engine to scan the buffer and string (AmsiScanBuffer() and AmsiScanString()). In case a known signature is triggered the execution doesn’t initiate and a message appears that the script has been blocked by the antivirus software - as shown in the diagram below²⁵⁰.



²⁴⁸ <https://learn.microsoft.com/en-us/defender-endpoint/amsi-on-mdav>

²⁴⁹ <https://github.com/MirekVales/MVsDotNetAMSIClient>

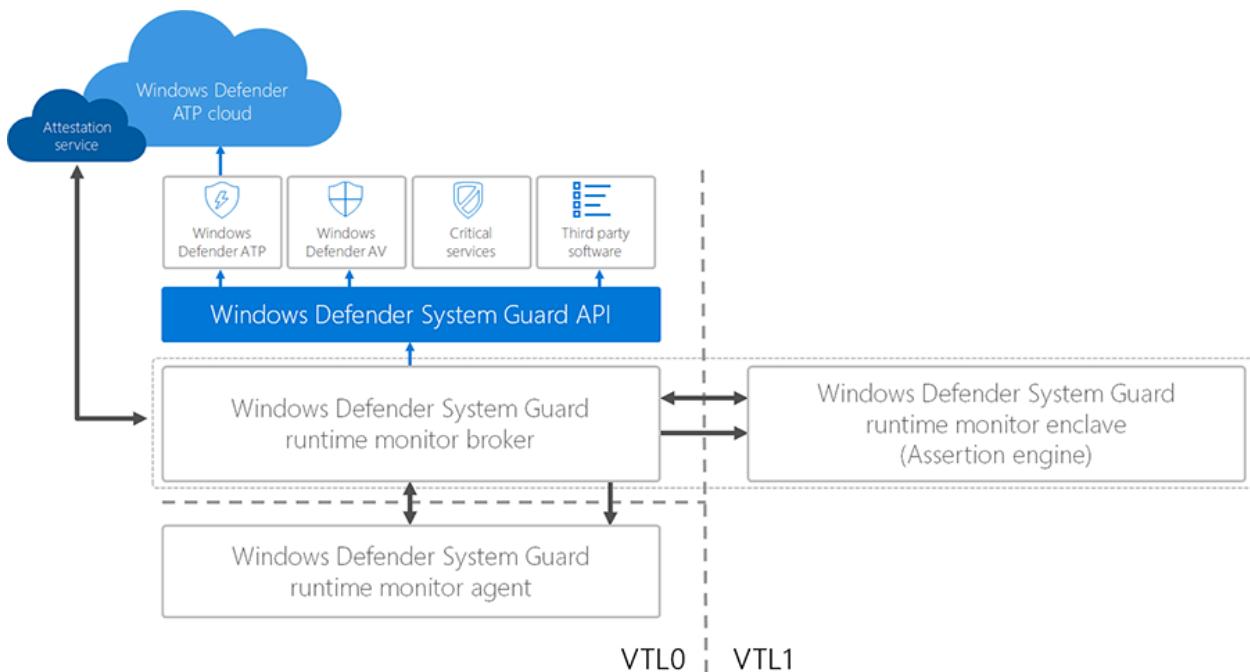
²⁵⁰ <https://pentestlaboratories.com/2021/05/17/amsi-bypass-methods/>

SGRM (System Guard Runtime Monitor)

SGRM (System Guard Runtime Monitor) is part of Windows Defender, which is used to ensure the operating system integrity. It was introduced as part of Windows 10 1709 update. By the way, SGRM has a Microsoft internal project name which is “Octagon”²⁵¹.

Overall, the main goal of SGRM is ensuring the security of the operating system. The runtime attestation of SGRM can assist with use cases such as: detecting artifacts of exploits/rootkits/kernel tampering, providing conditional access based on device posture, helping with game anti-cheating capabilities and more - a block diagram of SGRM is shown below²⁵².

Lastly, as part of Windows 10 Fall Creators Update all system integrity features were merged into “Windows Defender System Guard”. It provides an API that can be used for attesting the device in a point of time (based on different measurements). For the attestation to be trustable it must be: isolated from an attacker, signed and the isolation should be attestable. In order to provide that VBS and enclaves are leveraged²⁵³.



²⁵¹ <https://blog.syscall.party/2022/08/02/inside-windows-defender-system-guard-runtime-monitor.html>

²⁵² <https://www.microsoft.com/en-us/security/blog/2018/04/19/introducing-windows-defender-system-guard-runtime-attestation/>

²⁵³ (<https://www.slideshare.net/IgorKorkin/protected-process-light-will-be-protected-memoryranger-fills-the-gap-again>)

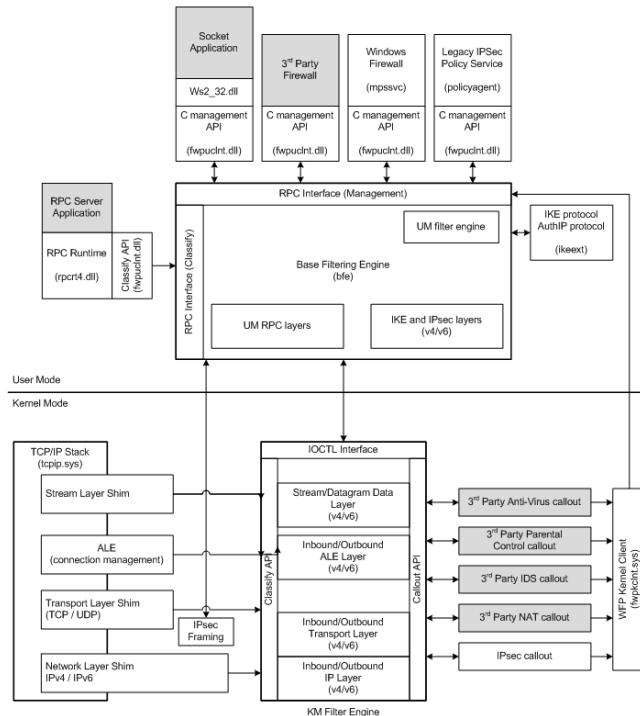
WFP (Windows Filtering Platform)

The WFP (Windows Filtering Platform) is a network traffic processing platform. It replaces Windows XP/Server 2003 (thus since Windows Vista) network traffic filtering interface. WFP is dependent on Windows Vista's "Next Generation TCP/IP Stack", which provides sets of hooks into the network stack/filtering engine²⁵⁴. It resembles "NetFilter" used under Linux²⁵⁵.

Overall, WFP is based on four main components: "Shims", "Filtering Engine", "Base Filtering Engine" and "Callouts". Shims are placed between the Network Stack and the filter engine in the kernel. Filtering engine is the core multi-layer filtering infrastructure, which resides both in kernel-mode and user-mode. Base filtering engine is a user-mode service managing WFP. Callouts are functions exposed by a driver and used for performing the filtering operations - more information on each one of those components in future writeups. A diagram showcasing the WFP architecture is shown below²⁵⁶.

Lastly, among the different features included as part of WFP are: stateful connection filtering, supports packet and stream processing, support both IPv4 and IPv6, allows integration of IPSec and firewall filtering, handles most packet reassembly and state tracking and more²⁵⁷.

Windows Filtering Platform Architecture Overview



²⁵⁴ https://en.wikipedia.org/wiki/Windows_Filtering_Platform

²⁵⁵ <https://medium.com/@aboutnaru/the-linux-security-journey-netfilter-90c6cf12ca40>

²⁵⁶ <https://learn.microsoft.com/en-us/windows/win32/fwp/windows-filtering-platform-architecture-overview>

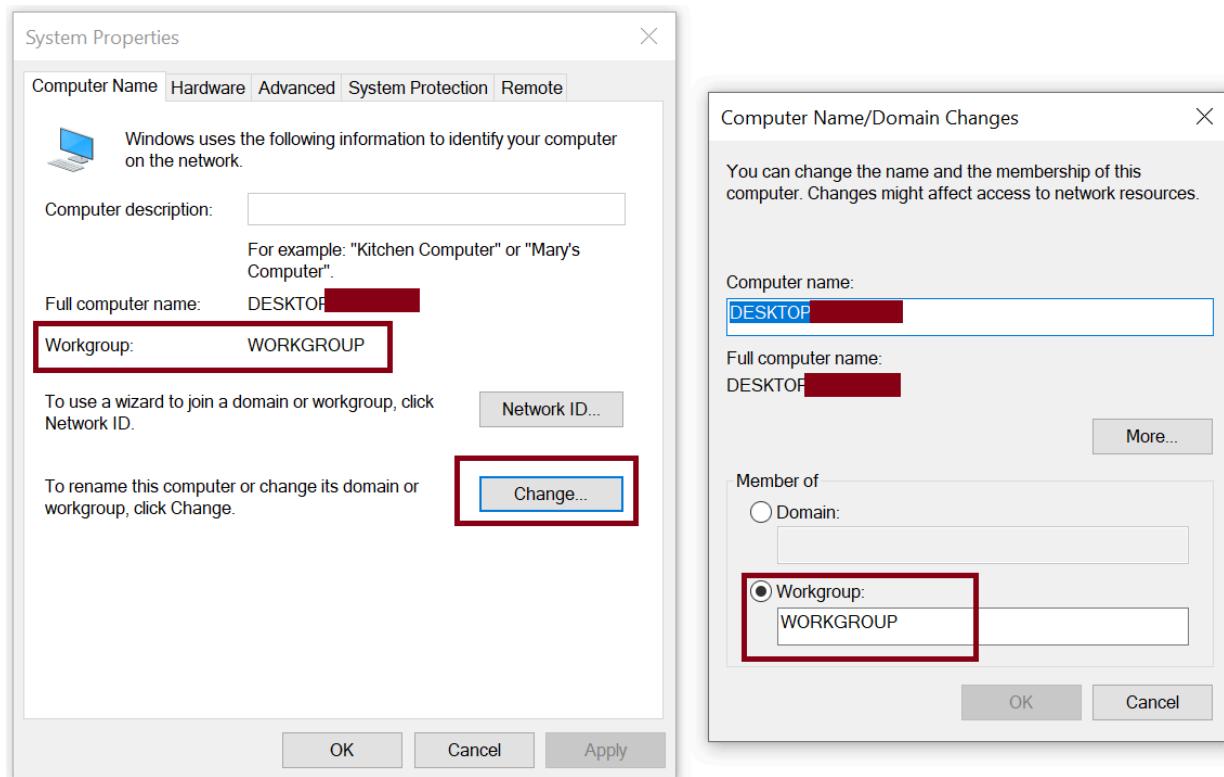
²⁵⁷ <https://learn.microsoft.com/en-us/windows/win32/fwp/about-windows-filtering-platform>

Windows Workgroup

Overall, a workgroup is a peer-to-peer network based on nodes running the Microsoft Windows operating system. We can think about it as a group of computers/servers on the LAN (Local Area Network), which don't have to go through a specific server in order to share resources²⁵⁸. Examples of such resources are: printers, files and internet connections.

Moreover, in the case of a workgroup there is no central management of users and groups. Both of them are stored locally on every system as part of the local SAM²⁵⁹. Thus, an administrator must create an account on each system if we want users to access them²⁶⁰.

Lastly, each workgroup is identified by a name that we can change if we want - as shown in the screenshot below. Also, it is dependent on the SMB (Server Message Block) protocol over NetBIOS (Network Basic Input Output System) - more on them in future writeups.



²⁵⁸ <https://www.windows-active-directory.com/what-is-a-workgroup-and-how-is-it-set-up.html>

²⁵⁹ <https://medium.com/@boutnaru/windows-security-sam-security-account-manager-c93ddadff388a>

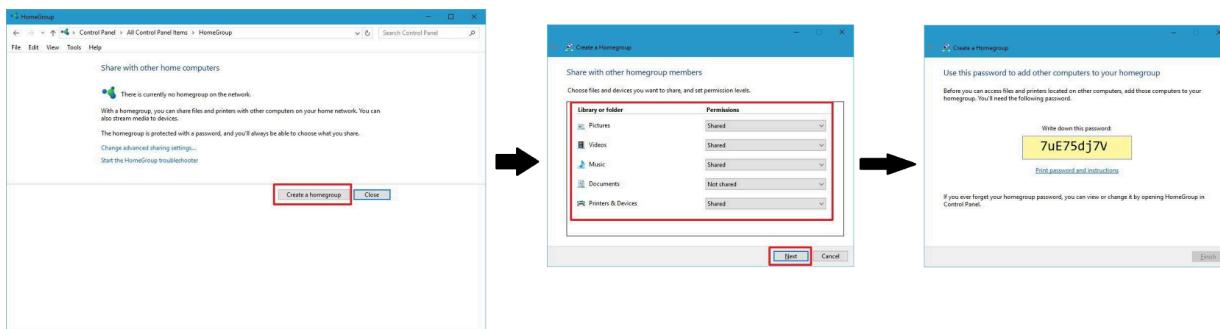
²⁶⁰ <https://networkencyclopedia.com/windows-workgroup/>

Windows Homegroup

Basically, a homegroup is a group of PCs on a home network that can share resources between them (like printers and files). We can protect a home using a password (that we can modify anytime). After we create/join a homegroup we need to select the libraries we want to share (like “My Documents” and “My Pictures”). Homegroup has been introduced since Windows 7²⁶¹. We can configure a homegroup on Windows 10 as shown below²⁶².

Overall, the homegroup feature has been removed from Windows 10 (Version 1803) . Thus, “Homegroups” won’t appear as part of the “File Explorer” user interface²⁶³ and as part of the “Control Panel” user interface²⁶⁴ anymore. It is important to know that folders/files/printers that were previously shared using HomeGroup will continue to be shared. They can be accessed using “\\[PC]\\[SharedFolder]”²⁶⁵.

Lastly, although the “HomeGroup” feature is deprecated we can still share printers/files/folders using other features built into Windows 10 (Version 1803) and above²⁶⁶.



²⁶¹ <https://support.microsoft.com/en-us/windows/homegroup-from-start-to-finish-9f802e8c-900f-60fb-826f-6fe06add8fe9>

²⁶² <https://www.windowscentral.com/how-setup-and-manage-windows-10-homegroup-local-network>

²⁶³ <https://medium.com/@boutnaru/the-windows-concept-journey-file-explorer-previously-windows-explorer-e48077b135a0>

²⁶⁴ <https://medium.com/@boutnaru/the-windows-concept-journey-control-panel-34bf84ca7ff0>

²⁶⁵ <https://support.microsoft.com/en-us/windows/homegroup-removed-from-windows-10-version-1803-07ea5db1-7bea-4d11-68a3-a31ff4a09979>

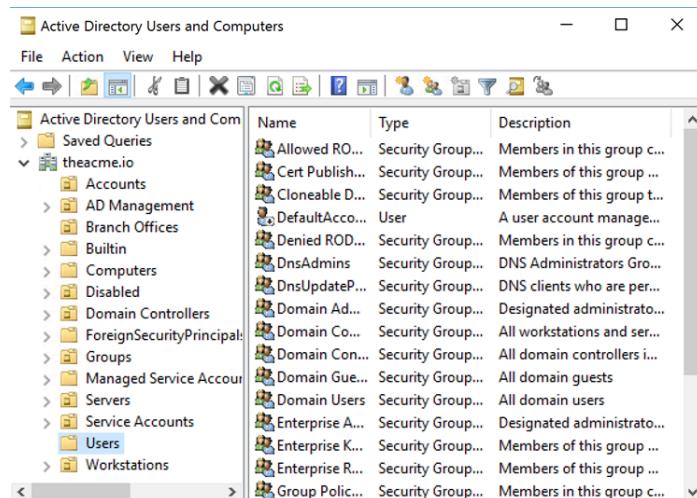
²⁶⁶ <https://tinyurl.com/3sz87nv>

AD (Active Directory)

AD (Active Directory) is an hierarchical database which is used for storing information about different objects on the network (users, computer, printers, shares and more). AD is composed of “schema” (defines the classes of objects and attributes stored by them), “global catalog” (contains a subset of attributes of every object in the directory), “query and index mechanism” and a “replication service” for distributing the data across the network²⁶⁷. Probably the most command UI for managing AD is the “Active Directory Users and Computers” snap-in (das.msc) - as shown in the screenshot below²⁶⁸.

Overall, AD provides a security boundary and is used to federate authentication (aka “domain authentication” vs local SAM based) and policy enforcement (GPO). The AD database is stored on a DC (Domain Controller) inside the “ntds.dit” file - more on that in future writeups. By the way , there are multiple best practices (patching, monitoring, recovery plans and more) for security AD as part of Microsoft’s documentation²⁶⁹.

Lastly, due to the fact the AD database is stored on DCs we can access it remotely using the LDAP (Lightweight Directory Access Protocol) over TCP port 389 or 636 in case of LDAPS (LDAP over SSL). LDAP supports an interface of queries in order to ask the directory service for data²⁷⁰ - more on that in future writeups.



²⁶⁷ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>

²⁶⁸ <https://www.torchsec.org/what-is-azure-active-directory-active-directory-security/>

²⁶⁹ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory>

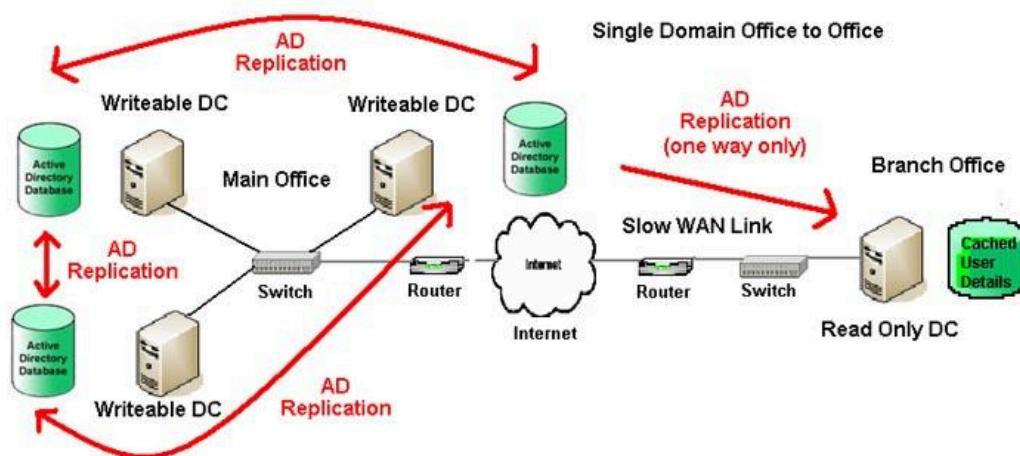
²⁷⁰ <https://www.varonis.com/blog/the-difference-between-active-directory-and-ldap>

RODC (Read Only Domain Controller)

RODC (Read Only Domain Controller) is a type of DC (domain controller) that has read-only partitions of AD DS (Active Directory Domain Services) database. It is mostly used in cases which: there is not enough physical security to the datacenter and\or limited bandwidth for establishing network connections. This feature is relevant since “Windows Server 2008”²⁷¹ - as described in the diagram below²⁷².

Overall, RODC stores all the “Active Directory”²⁷³ objects and attributes that a writable DC holds (except for account passwords). It is important to understand that modifications\alterations can't be made to the database stored on the RODC. In case we want to make changes we need to perform them on a writable DC, which are then replicated back to the RODC²⁷⁴.

Lastly, RODC also stores a read-only copy of the “SYSVOL” share²⁷⁵. A RODC is designed primarily to be deployed in a branch office environment²⁷⁶. Thus, RODC provides different benefits such as: mitigating and helping in the removal of replication concerns, preventing accidental deletion of AD objects/SYSVOL and preventing rogue application from changing the AD schema²⁷⁷.



²⁷¹ <https://www.windows-active-directory.com/read-only-domain-controller.html>

²⁷² https://www.mcmse.com/microsoft/guides/70-412/rodc_and_rrp.shtml

²⁷³ <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-af9e795f86b4>

²⁷⁴ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/planning-domain-controller-placement>

²⁷⁵ <https://medium.com/@boutnaru/the-windows-security-journey-epo-group-policy-object-f8757933217a>

²⁷⁶ [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc755058\(v=ws.10\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc755058(v=ws.10))

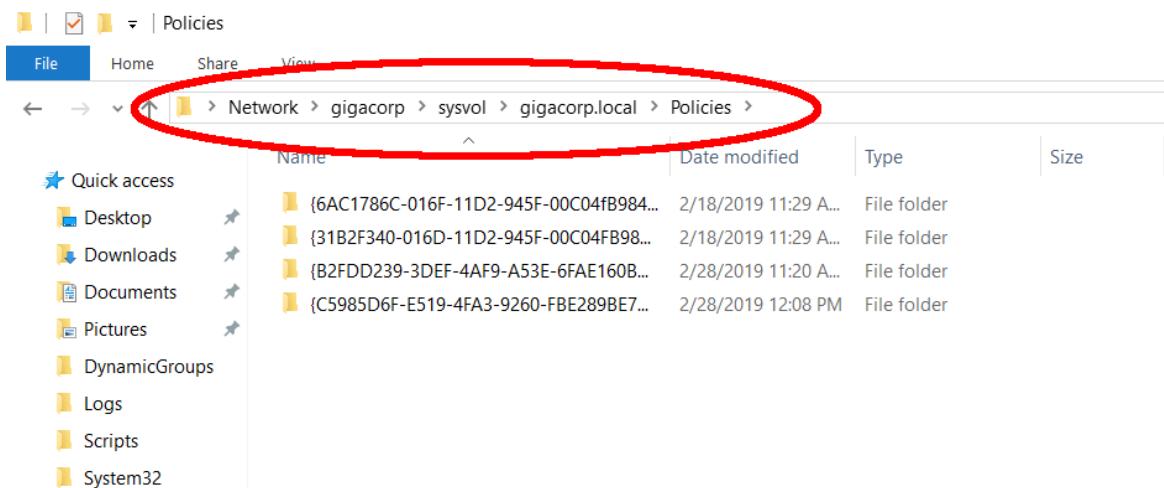
²⁷⁷ <https://www.veeam.com/blog/read-only-domain-controller-rodc-configuration-active-directory-best-practices.html>

GPO (Group Policy Object)

GPO (Group Policy Object) is a collection of policy settings (aka “Group Policy Settings”), which are represented in the file system and in the Active Directory. Each GPO has a unique name and a GUID²⁷⁸. GPO can be used for customizing settings in different areas such as: registry based policy, security options and folder redirection²⁷⁹. Policies can be assigned to a computer (like startup/shutdown scripts) or to a user (logon/logoff scripts).

Moreover, in order to configure a GPO we can use the “Group Policy Editor” which is a tool that hosts MMC extension snap-ins that manage policy settings²⁸⁰. We can link (associate) a GPO with one or more containers (site/domain/organizational unit). Thus, multiple containers can be linked to the same GPO, and a single container can have more than one GPO linked to it²⁸¹.

Lastly, a GPO is stored both in a “Group Policy Container” (GPC) and a “Group Policy Template” (GPT). GPC is an Active Directory container (located at [Domain]\System\Policies) that contains GPO properties (like version information and GPO status). GPT is a file system folder that includes policy data specified by .adm files, security settings, script files, and information about applications that are available for installation. The GPT is located in the system volume folder (sysvol) in the domain \Policies subfolder. By the way, sysvol is a network share exposed by domain controllers²⁸² - as shown in the screenshot below²⁸³ - the pattern of the location is \\[Domain]\Sysvol\[Domain]\Policies.



²⁷⁸ <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects>

²⁷⁹ <https://www.varonis.com/blog/group-policy-objects>

²⁸⁰ <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-object-editor>

²⁸¹ <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/linking-gpos-to-active-directory-containers>

²⁸² <https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-storage>

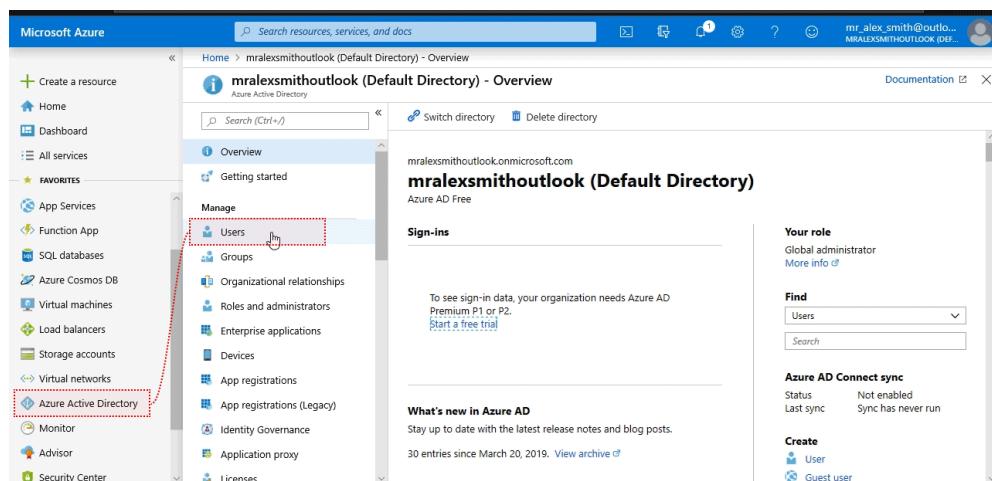
²⁸³ <https://www.easy365manager.com/where-are-gpos-stored/>

AAD (Azure Active Directory)

AAD (Azure Active Directory) is a cloud directory and authentication service. It is leveraged by “Office 365” and “Azure” for account/groups/role management. AAZ can act as an IdP (Identity Provider) and supports federation (like SAML and more). As of 2023 AAD is deployed in over 30 datacenters around the world leveraging “Azure Availability Zones” present²⁸⁴.

Overall, AAZ is not a cloud version of AD²⁸⁵. For example AAZ does not support NTLM and/or kerberos authentication methods and uses cloud based authentication mechanisms (like OAuth2, WS-Security and SAML). Also, LDAP is not used and REST (Representational State Transfer) API is used instead for communicating with web services. AAZ can provide mobile device management with Microsoft Intune while AD does not support mobile devices²⁸⁶.

Lastly, for managing AAD we can use the “Microsoft Azure Portal” - as shown in the screenshot below²⁸⁷. By the way, on 2023 the name of “Azure Active Directory” was changed to “Microsoft Entra ID”²⁸⁸.



²⁸⁴ <https://www.torchsec.org/what-is-azure-active-directory-active-directory-security/>

²⁸⁵ <https://medium.com/@bouthnaru/the-windows-concept-journey-ad-active-directory-af9e795f86b4>

²⁸⁶ <https://www.varonis.com/blog/azure-active-directory>

²⁸⁷ <https://stealthmail.com/help/deployment-guide/add-user-to-azure-active-directory>

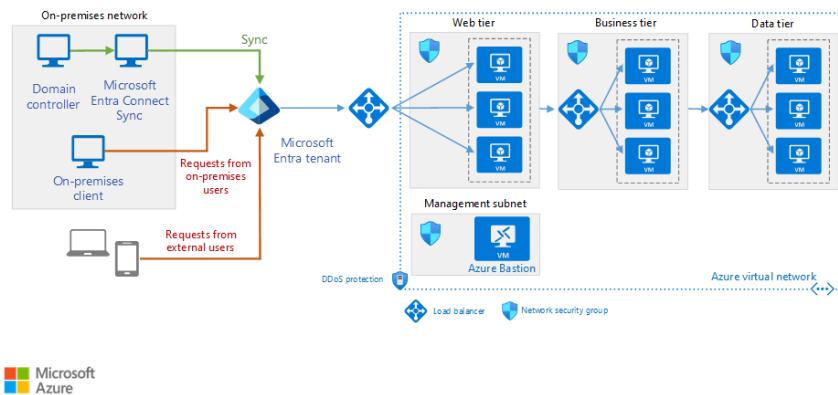
²⁸⁸ <https://www.microsoft.com/en-us/security/blog/2023/07/11/microsoft-entra-expands-into-security-service-edge-and-azure-ad-becomes-microsoft-entra-id/>

Microsoft Entra ID

“Microsoft Entra ID” is a cloud identity and access management solution which provides secure adaptive access, unified identity management and seamless user experience. Among “Microsoft Entra ID” capabilities we can find: applications integration and SSO (single sign on), unified admin center, passwordless and MFA (multi-factor authentication), conditional access, identity protection, privileged identity management and end-user self service²⁸⁹.

Overall, “Microsoft Entra ID” is the new name of “Azure Active Directory”²⁹⁰. It can be used for providing access to external resources like “Azure Portal”, “Office 365” and many other SaaS (Software as a Service) applications²⁹¹. For guidelines of how to create a new tenant in “Microsoft Entra ID” we should check out Microsoft’s documentation²⁹².

Lastly, we can sync between an on-premises “Active Directory” to “Microsoft Entra ID” (Microsoft Entra Connect Sync service). By doing so we ensure that identity information stored in the cloud is consistent with the identity information stored on-premises. The sync service is installed using “Microsoft Entra Connect” software on a virtual machine/physical host as part of the on-premises environment - as shown in the diagram below²⁹³.



²⁸⁹ <https://www.microsoft.com/en-us/security/business/identity-access/microsoft-entra-id>

²⁹⁰ <https://medium.com/@bouthnaru/the-windows-security-journey-aad-azure-active-directory-b7e33d2fb16b>

²⁹¹ <https://learn.microsoft.com/en-us/entra/fundamentals/whatis>

²⁹² <https://learn.microsoft.com/en-us/entra/fundamentals/create-new-tenant>

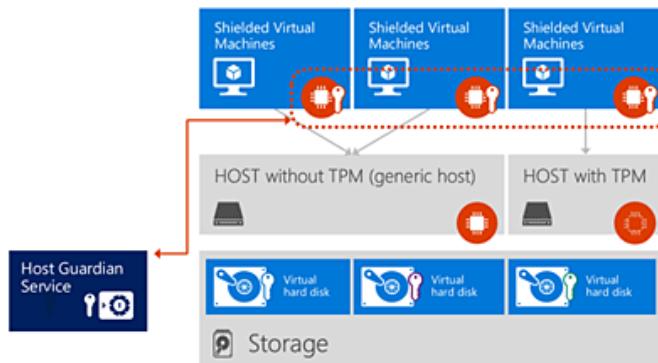
²⁹³ <https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/identity/azure-ad>

HGS (Host Guardian Service)

HGS (Host Guardian Service) is a server role supported by the Windows operating system. It is used for configuring guarded hosts and running shielded VMs (which are VMs leveraging BitLocker for protecting their disks and state). This role is included in Windows from “Windows Server 2016”. HGS provides mainly attestation and key protection services²⁹⁴ - as shown in the diagram below.

Overall, the attestation service is used for validating a Hyper-V host as a “guarded host”. In case of a validation which the key protection services enabled. It is used to provide the transport key required to unlock and subsequently run shielded virtual machines²⁹⁵.

Lastly, to install HGS we can use the dashboard of the “Server Manager” and by selecting in the “Server Roles” menu the “Host Guardian Service”, we can also use PowerShell for doing that. By the way, the security best practice (by Microsoft) states the domain admins and HGS admins should be different persons with different user accounts²⁹⁶.



²⁹⁴ <https://www.webopedia.com/definitions/host-guardian-service/>

²⁹⁵ <https://argonsys.com/microsoft-cloud/library/step-by-step-configuring-the-host-guardian-service-in-windows-server-2016/>

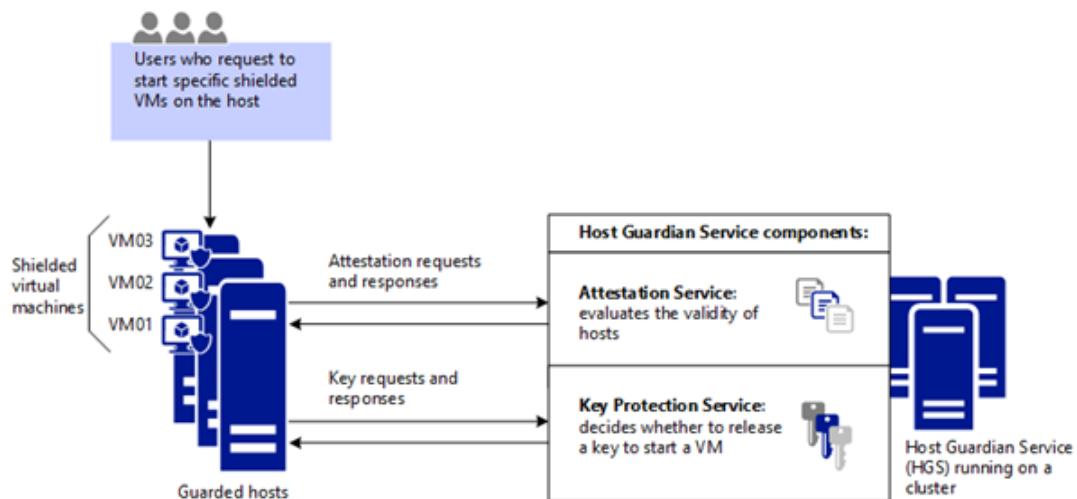
²⁹⁶ <https://www.vladan.fr/what-is-host-guardian-service/>

Guarded Host

The goal of a “Guarded Host” is to provide a trusted environment/server on which we can create and run “Shielded VMs”²⁹⁷. The trustworthiness of a “Guarded Host” is verified/attested by the “Host Guardian Service” (HGS) in order to provide a protection key for decrypting the virtual machine²⁹⁸.

Overall, the “Guarded Host” is the host server on which Hyper-V is installed. Thus, virtual machine on that host can become “Shielded VMs”²⁹⁹. By the way, together with HGS and shielded virtual machines it composes a “Guarded Fabric” - as shown in the diagram below³⁰⁰.

Lastly, for example a guarded host is running a “Windows 2016 Datacenter” edition of the operating system with Hyper-V installed³⁰¹. Due to the fact “Azure Confidential Computing” and “Azure Security Center” are becoming a significant focus, Microsoft has deprecated the “Guarded Fabric” and “Shielded VMs” features as part of “Windows 2022”³⁰².



²⁹⁷ <https://medium.com/@boutnaru/the-windows-security-journey-microsoft-hyper-v-shielded-vm-2f7c86dbdf56>

²⁹⁸ <https://medium.com/@boutnaru/the-windows-security-journey-hgs-host-guardian-service-e641ce2f8dde>

²⁹⁹ <https://www.entrust.com/sites/default/files/documentation/integration-guides/microsoft-host-guardian-server-shielded-vm-nshield-hsm-ig.pdf>

³⁰⁰ <https://github.com/MicrosoftDocs/windowsserverdocs/blob/main/WindowsServerDocs/security/guarded-fabric-shielded-vm/guarded-fabric-and-shielded-vms.md>

³⁰¹ <https://tinyurl.com/4takxsfb>

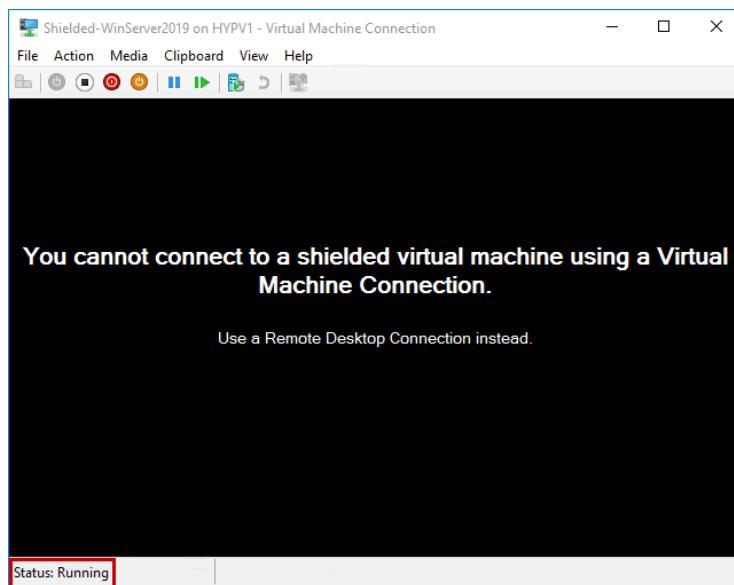
³⁰² <https://www.parallels.com/blogs/ras/windows-server-2019-vs-2022/>

Microsoft Hyper-V Shielded VM

A “Hyper-V Shielded VM” is a security feature for protecting Hyper-V second generation virtual machines. This is done by leveraging different technologies such as: secure boot, TPM (Trusted Platform Module), BitLocker and “Host Guardian Service”³⁰³.

Overall, a shielded virtual machine protects the state/data by leveraging a vTPM (Virtual Platform Module) device that enables BitLocker to encrypt the virtual machine disk/s. By the way, the vTPM device is itself encrypted with a transport key³⁰⁴. In order to protect the transport key, HGS is used.

Lastly, the goal of “Shielded VMs” is to provide protection from Hyper-V administrators and malware attacks. Shielded virtual machines are based on: code integrity, virtual secure mode (VSM), TPMv2 (Trusted Platform Module V2) and HGS (Host Guardian Service). Due to the encryption of a shielded VM VHD (Virtual Hard Disk) an administrator is not able to attach it³⁰⁵. Also, we can't connect to the video stream of a shielded VM using the “Hyper-V Manager” - as shown in the screenshot below³⁰⁶. If we want we can disable that feature (while still encrypting the disk/s). By the way, we can still use RDP (Remote Desktop Protocol) to connect to the shielded VM.



³⁰³ <https://www.techtarget.com/searchwindowsserver/definition/Microsoft-Hyper-V-Shielded-VM>

³⁰⁴ <https://argonsys.com/microsoft-cloud/library/step-by-step-configuring-the-host-guardian-service-in-windows-server-2016/>

³⁰⁵ <https://techcommunity.microsoft.com/discussions/windowsserver/introduction-to-shielded-virtual-machines-in-windows-server-2016/12753>

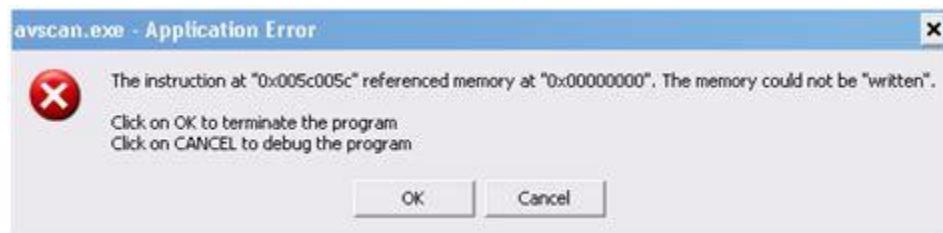
³⁰⁶ <https://www.red-gate.com/simple-talk/sysadmin/powershell/hyper-v-and-powershell-shielded-virtual-machines/>

Null Page Mitigation

Since Windows 8, Microsoft blocks the ability to map/allocate the “Null Page” (this corresponds to the virtual address “0x00000000”). This is done for mitigating “NULL Pointer dereference” vulnerabilities. It does not matter where the bug/vulnerability exists this mitigation reduces the possibility for arbitrary code execution to a DoS (Denial of Service) at best³⁰⁷. By the way, this feature was also backported to Vista+ for x64 only³⁰⁸.

Overall, a “NULL Pointer dereference” vulnerability occurs when a pointer has a NULL value and it is used as a valid pointer (for example in case of function pointers) to memory³⁰⁹ - as shown in the screenshot below³¹⁰. This feature was also supported in EMET (Enhanced Mitigation Experience Toolkit) for user-mode only. In that case the mitigation worked by allocating the first page of memory before a program starts, which of course is not³¹¹.

Lastly, the only exception to the mitigation is by enabling NTVDM which is disabled by default³¹². Thus, enabling NTVDM decreases protection against Null dereference vulnerabilities³¹³.



³⁰⁷ <https://securityintelligence.com/exploitation-vulnerabilities-allow-diverting-kernel-execution-flow-windows/>

³⁰⁸ <https://seclists.org/fulldisclosure/2014/May/112>

³⁰⁹ <https://www.mista.nu/research/nullpage.pdf>

³¹⁰ <https://iso26262fs.wordpress.com/2015/06/18/null-pointer-dereference-%EB%9E%80/>

³¹¹ <https://msrc.microsoft.com/blog/2009/10/announcing-the-release-of-the-enhanced-mitigation-evaluation-toolkit/>

³¹² <https://www.abatchy.com/2018/01/kernel-exploitation-6>

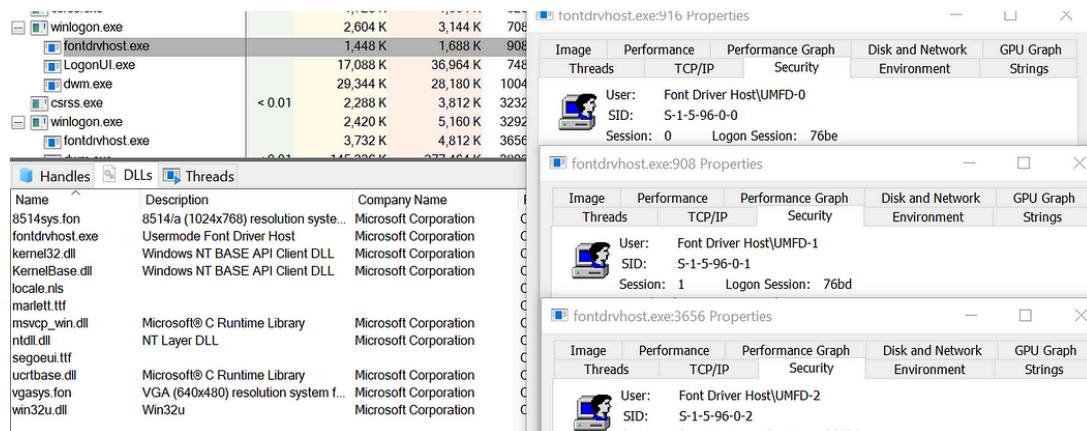
³¹³ <https://learn.microsoft.com/en-us/windows/security/threat-protection/overview-of-threat-mitigations-in-windows-10>

UMFD (User Mode Font Driver)

The UMFD (User Mode Font Driver) security mitigation has been released as part of Windows 10. It moves font processing into user-mode as part of an “App Container”³¹⁴. By the way, we can also block untrusted fonts from being processed³¹⁵.

Overall, the user-mode process which is responsible for the font parsing is “fontdrvhost.exe”. It is executed with the permissions of a user in the following pattern: “Font Driver Host\UMFD[SessionID]”. Also, the SID of the user is in the pattern of “S-1-5-96-[SessionID]”³¹⁶ - as shown in the screenshot below.

Lastly, UMFD is used for preventing local escalation of privileges and/or sandbox escapes using untrusted fonts. Also, a memory corruption in font processing won’t influence the kernel directly³¹⁷.



³¹⁴ <https://www.ghacks.net/2017/09/02/windows-10-threat-mitigation-advancements/>

³¹⁵ <https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/block-untrusted-fonts-in-enterprise>

³¹⁶ <https://medium.com/@boutnaru/the-windows-process-journey-fontdrvhost-exe-usermode-font-driver-host-a5dd4f6159ba>

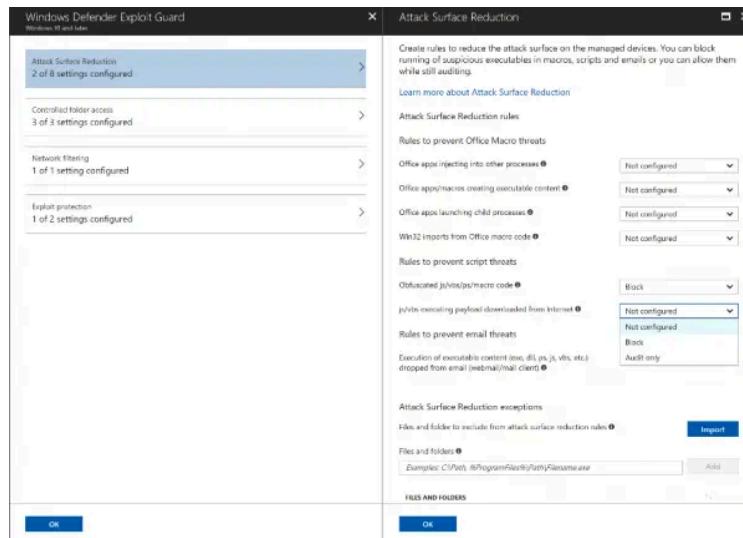
³¹⁷ <https://www.slideserve.com/tait/10>

WDEG (Windows Defender Exploit Guard)

WDEG (Windows Defender Exploit Guard) is based on four different components: “Attack Surface Reduction” (ASR) , “Controlled Folder Access”, “Exploit Protection” and “Network Protection” - as explained later in detail. It is basically an IPS (Intrusion Detection Prevention) capability part of Windows³¹⁸. Security personnel can view events of WDEG in Windows Defender Advanced Threat Protection (WD ATP) - as shown in the screenshot below³¹⁹.

Overall, “Attack Surface Reduction” (ASR) allows enterprises blocking Office/scripts/email based threats for preventing malware infecting the device. “Controlled Folder Access” is leveraged for protecting sensitive data from ransomware by blocking untrusted apps for accessing protected folders³²⁰. “Exploit Mitigations” is a replacement of EMET providing a set of exploit mitigations³²¹.

Lastly, “Network Protection” is used for protecting the endpoint against web-based threats, this is done by blocking any outbound processes while leveraging “Windows Defender SmartScreen”³²² .By the way, WDEG leverages the “Microsoft Intelligent Security Graph (ISG)”³²³.



³¹⁸ <https://learn.microsoft.com/en-us/mem/configmgr/protect/deploy-use/create-deploy-exploit-guard-policy>

³¹⁹ <https://www.thewindowsclub.com/windows-defender-exploit-guard>

³²⁰ <https://medium.com/@boutnaru/the-windows-security-journey-controlled-folder-access-d9972de1b9ea>

³²¹ <https://www.microsoft.com/en-us/security/blog/2017/10/23/windows-defender-exploit-guard-reduce-the-attack-surface-against-next-generation-malware/>

³²² <https://medium.com/@boutnaru/the-windows-security-journey-windows-defender-smartscreen-afdda6fb63e1>

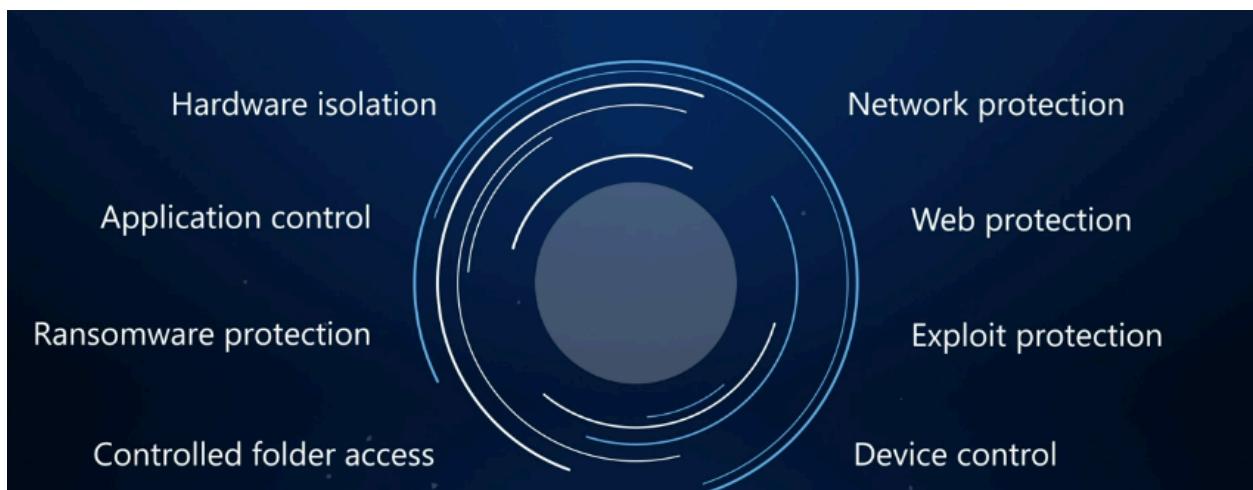
³²³ <https://medium.com/@boutnaru/the-windows-security-journey-microsoft-isg-microsoft-intelligent-security-graph-a32d787122b9>

ASR (Attack Surface Reduction)

ASR (Attack Surface Reduction) are security features that are included as part of the Windows operating system. The goal of those features is to block risky actions on Windows based systems. Thus, help in preventing different type of ransomware attacks and malware infection³²⁴ - as shown in the diagram below³²⁵.

Overall, we can configure the following as part of ASR: enabling hardware-base isolation for Microsoft Edge (Microsoft Defender Application Guard), enabling attack surface reduction, enabling application control (WDAC), enabling controlled folder access³²⁶, enabling device control, enabling exploit protection, setting up the network firewall³²⁷, enabling Web protection and turning on network protection - more information about those features in future writeups.

Lastly, we can configure the attack surface reduction capabilities in different methods like: “Microsoft Intune”, “Group Policy”, “PowerShell cmdlets” and “Microsoft Configuration Manager”. Also, we can set part of the features (attack surface reduction rules, exploit protection, controlled folder access and device control) to audit mode. The audited entries can be found in “Applications and Services > Microsoft > Windows > Windows Defender > Operational”³²⁸.



³²⁴ <https://www.cscoonline.com/article/570103/how-to-use-windows-defender-attack-surface-reduction-rules.html>

³²⁵ <https://learn.microsoft.com/en-us/defender-endpoint/overview-attack-surface-reduction>

³²⁶ <https://medium.com/@boutnaru/the-windows-security-journey-controlled-folder-access-d9972de1b9ea>

³²⁷ <https://medium.com/@boutnaru/the-windows-security-journey-windows-defender-firewall-7e596b369661>

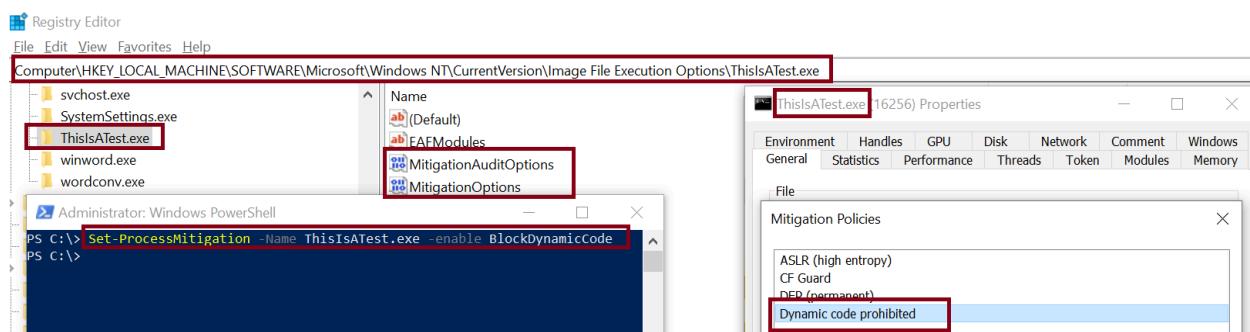
³²⁸ <https://learn.microsoft.com/en-us/defender-endpoint/overview-attack-surface-reduction>

ACG (Arbitrary Code Guard)

ACG (Arbitrary Code Guard) is a security mitigation included as part of the Windows operating system. It is designed for preventing/detecting that: existing code can't be modified (code pages can't become writable) and writing code to data segments and executing them (data cannot become code). This is done by enforcing memory from having both W (write) and X (execute) bits which is also referenced as "W^X"³²⁹.

Overall, we can set ACG on the current process using the "SetProcessMitigationPolicy" API³³⁰. ACG is also called "ProcessDynamicCodePolicy". When enforcing it, code pages of the process are immutable, thus calling "VirtualProtect" with "PAGE_EXECUTE_READ_WRITE" will fail. Also, new unsigned code can't be created which means that calling "VirtualAlloc" with "PAGE_EXECUTE_READWRITE" is going to fail³³¹.

Lastly, we can also use the "Set-ProcessMitigation" PowerShell cmdlet³³² in order to enable ACG - as shown below. By the way, the ACG configuration is saved in the registry per process in the following location: "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options" with the "MitigationOptions" under the relevant process name³³³ - as shown in the screenshot below (the right side is from SystemInformer³³⁴).



³²⁹ <https://www.countercraftsec.com/blog/arbitrary-vs-kernel/>

³³⁰ <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-setprocessmitigationpolicy>

³³¹ <https://www.ired.team/offensive-security/defense-evasion/acg-arbitrary-code-guard-processdynamiccodepolicy>

³³² <https://learn.microsoft.com/en-us/powershell/module/processmitigations/set-processmitigation?view=windowsserver2022-ps>

³³³ <https://learn.microsoft.com/en-us/defender-endpoint/exploit-protection-reference>

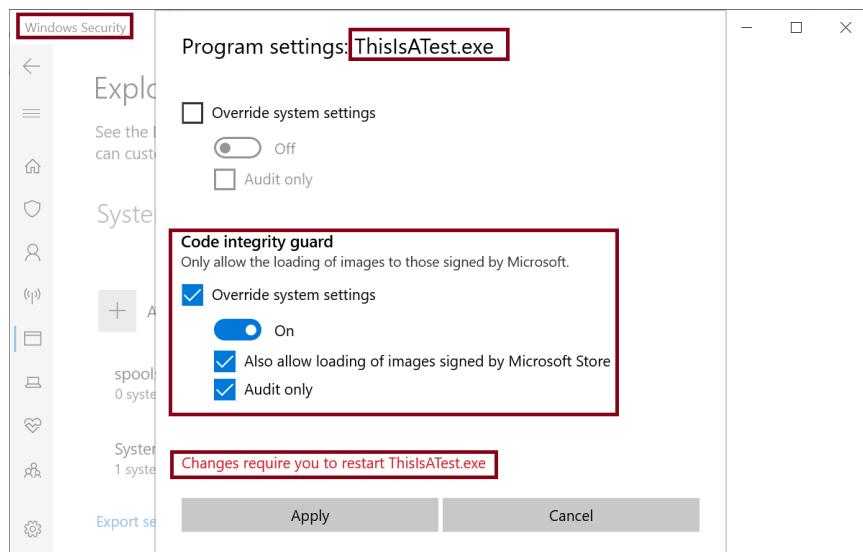
³³⁴ <https://github.com/winsiderss/systeminformer>

CIG (Code Integrity Guard)

CIG (Code Integrity Guard) is a security feature of the Windows operating system (that was launched in 2015 with the launch of Windows 10). It first started as part of “Device Guard” used for preventing the tampering of OS drivers loaded by Windows 10. Later Microsoft expanded CIG to “Microsoft Edge” and after that to any application (based on the developer decision)³³⁵.

Overall, an application with CIG enabled can only load binaries/DLLs that are signed by Microsoft. By the way, CIG is implemented as part of the memory manager of Windows. Thus, CIG is relevant for binaries loaded by the process or those injected to the process. The integrity check is done while trying to map the binary into memory. In case of a non signed binary by Microsoft, the memory manager returns the error “STATUS_INVALID_IMAGE_HASH”³³⁶.

Lastly, we can configure (audit/blocking mode) CIG per process as part of the “Exploit Protection” menu of “System Settings” (Windows Security->App & Browser Control->Exploit Protection->Programs Settings). The settings are relevant from the next time the executable is launched - as shown in the screenshot below. Like with ACG³³⁷ the configuration is saved in the registry per process in the following location: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options”.



³³⁵ <https://www.bleepingcomputer.com/news/security/cigslip-attack-bypasses-windows-code-integrity-guard-cig/>

³³⁶ <https://learn.microsoft.com/en-us/defender-endpoint/exploit-protection-reference>

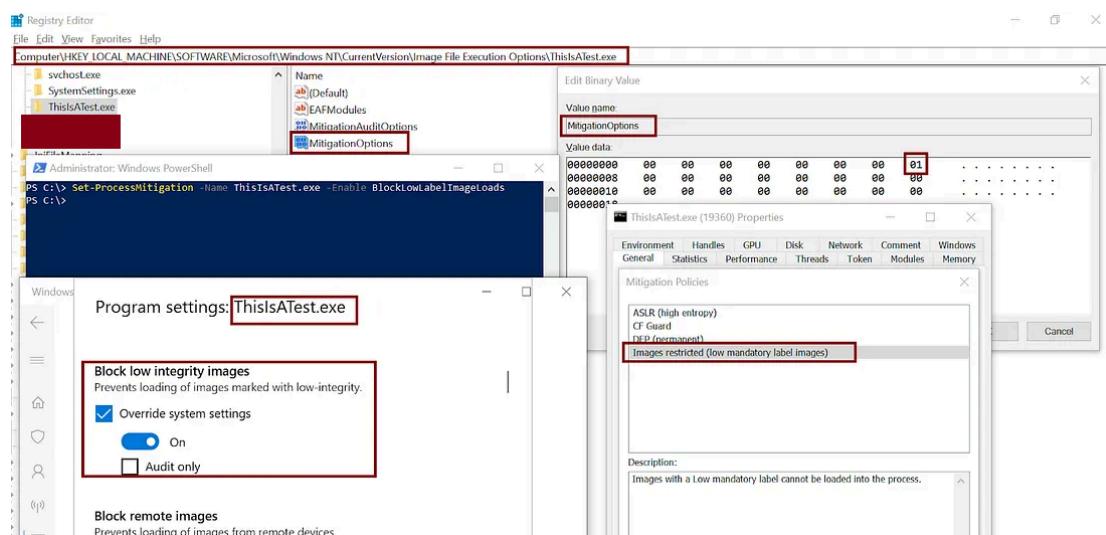
³³⁷ <https://medium.com/@boutnaru/the-windows-security-journey-acg-arbitrary-code-guard-74b08a8bd1e5>

Block Low Integrity Images

“Block Low Integrity Images” is a security feature of the Windows operating system that is used in order to prevent applications from loading files that are untrusted. Think for example about files downloaded from the Internet by a sandboxed browser³³⁸.

Overall, this security mitigation is used for blocking the loading of images that are marked with a “Low Integrity Level” label³³⁹. Like with CIG³⁴⁰ also “Block Low Integrity Images” is handled by the memory manager in the flow of mapping a file into the memory of the system. We can configure this mitigation (audit/block) per process as part of the “Exploit Protection” menu of “System Settings” (Windows Security->App & Browser Control->Exploit Protection->Programs Settings) - as shown in the screenshot below.

Lastly, we can also use the “Set-ProcessMitigation” PowerShell cmdlet³⁴¹ in order to enable “Block Low Integrity Images”- as shown below. By the way, the as with other mitigations (like ACG) the configuration is saved in the registry per process in the following location: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options” with the “MitigationOptions” under the relevant process name³⁴² - as shown in the screenshot below (the right side is from SystemInformer³⁴³).



³³⁸ <https://learn.microsoft.com/en-us/defender-endpoint/exploit-protection-reference#block-low-integrity-images>

³³⁹ <https://medium.com/@boutnaru/the-windows-security-journey-mandatory-integrity-control-mic-f7963550c0e7>

³⁴⁰ <https://medium.com/@boutnaru/the-windows-security-journey-cig-code-integrity-guard-7e410c8d2304>

³⁴¹ <https://learn.microsoft.com/en-us/powershell/module/processmitigations/set-processmitigation>

³⁴² <https://medium.com/@boutnaru/the-windows-security-journey-acg-arbitrary-code-guard-74b08a8bd1e5>

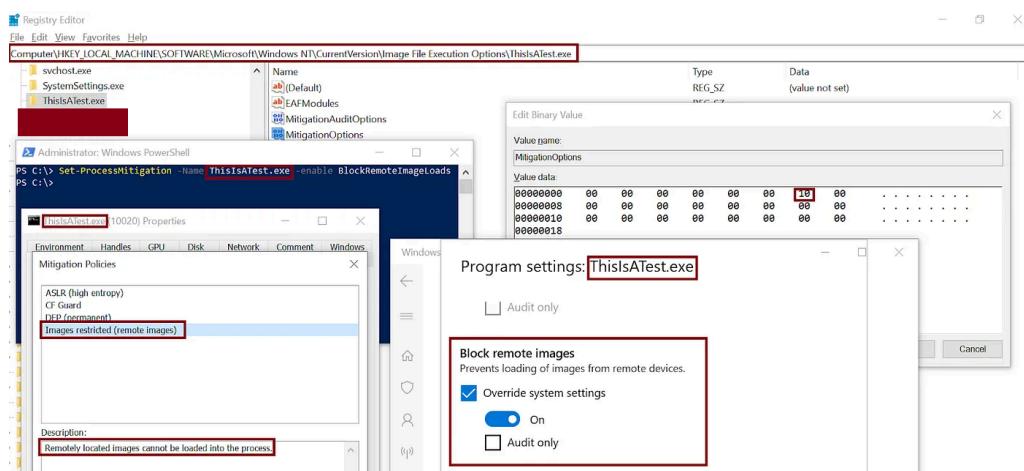
³⁴³ <https://github.com/winsiderss/systeminformer>

Block Remote Images

“Block Remote Images” is a security feature of the Windows operating system that is used for preventing applications from loading files stored on remote systems (think about UNC shares). By doing so it protects against loading binaries into the memory address space of processes from external devices which are controlled by attackers. In case an application attempts to map a remote file, a “STATUS_ACCESS_DENIED” error is returned³⁴⁴.

Overall, like with “Block Low Integrity Level”³⁴⁵ also “Block Remote Images” is handled by the memory manager in the flow of mapping a file into the memory of the system. We can configure this mitigation (audit/block) per process as part of the “Exploit Protection” menu of “System Settings” (Windows Security->App & Browser Control->Exploit Protection->Programs Settings) - as shown in the screenshot below.

Lastly, we can also use the “Set-ProcessMitigation” PowerShell cmdlet³⁴⁶ in order to enable “Block Remote Images”- as shown below (Set-ProcessMitigation -Name <ProcessName> -enable BlockRemoteImageLoads). By the way, the as with other mitigations (like ACG) the configuration is saved in the registry per process in the following location: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options” with the “MitigationOptions” under the relevant process name³⁴⁷ - as shown in the screenshot below (the left side is from SystemInformer³⁴⁸).



³⁴⁴ <https://learn.microsoft.com/en-us/defender-endpoint/exploit-protection-reference#block-remote-images>

<https://medium.com/@boutnaru/the-windows-security-journey-block-low-integrity-images-cd1fabe641cd>

<https://learn.microsoft.com/en-us/powershell/module/processmitigations/set-processmitigation>

³⁴⁷ <https://medium.com/@boutnaru/the-windows-security-journey-acg-arbitrary-code-guard-74b08a8bd1e5>

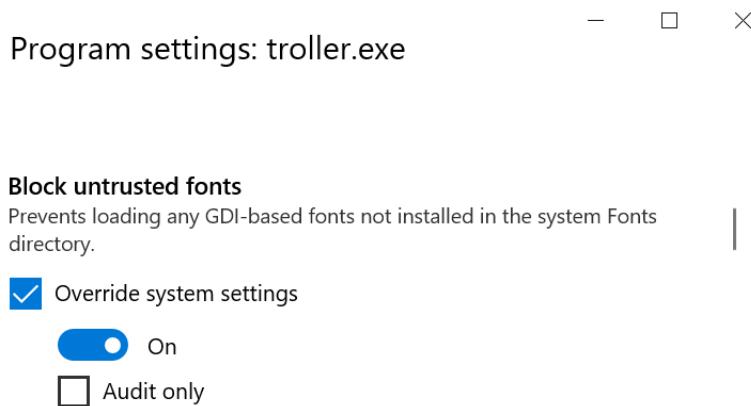
³⁴⁸ <https://github.com/winsiderss/systeminformer>

Block Untrusted Fonts

“Block Untrusted Fonts” is a security feature of the Windows operating system that is used for preventing applications from loading arbitrary font files. By default this feature is not turned on. There are three options of configuration: “On” (stops loading fonts outside the %windir%\Fonts directory and turns on event logging), “Audit” (turns on event logging, without blocking fonts from loading) and “Exclude apps to load untrusted fonts”³⁴⁹.

Overall, this security mitigation is used for blocking the load of GDI based fonts not installed in the system fonts directory. Thus, using it can reduce functionality in different use cases such as: applications using memory based fonts, documents which have embedded fonts, using “Internet Explorer” to view websites which embedded fonts, sending documents for printing with fonts not stored on the printing server and more³⁵⁰.

Lastly, we can configure “Block Untrusted Fonts” per process as part of the “Exploit Protection” menu of “System Settings” (Windows Security->App & Browser Control->Exploit Protection->Programs Settings). The settings are relevant from the next time the executable is launched - as shown in the screenshot below. Like with ACG³⁵¹ the configuration is saved in the registry per process in the following location: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options”.



³⁴⁹ <https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/block-untrusted-fonts-in-enterprise>

³⁵⁰ <https://learn.microsoft.com/en-us/troubleshoot/windows-client/shell-experience/feature-to-block-untrusted-fonts>

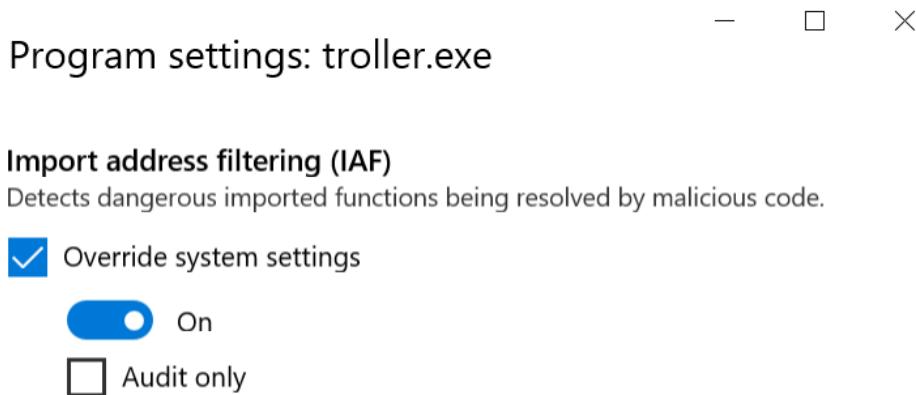
³⁵¹ <https://medium.com/@boutnaru/the-windows-security-journey-acg-arbitrary-code-guard-74b08a8bd1e5>

IAF (Import Address Filtering)

IAF (Import Address Filtering) is a security mitigation used for preventing code not backed by an image on disk to lookup addresses from the IAT (Import Address Table). Due to that, IAF is also referred to as “Import Address Table Access Filtering”³⁵². It is implemented by marking the relevant pages as “PAGE_GUARD” and validating the above condition as part of the exception handler. We can configure this mitigation (audit/block) per process as part of the “Exploit Protection” menu of “System Settings” (Windows Security->App & Browser Control->Exploit Protection->Programs Settings) - as shown in the screenshot below.

Overall, as with other mitigations IAF had originally implemented as part of EMET and later added to “Windows Defender”³⁵³. Among the APIs protected by IAF we can find: “GetProcAddress”, “GetProcAddressForCaller”, “LoadLibraryA”, “LoadLibraryExA”, “LoadLibraryW”, “LoadLibraryExW”, “VirtualProtect”, “VirtualProtectEx”, “WinExec”, “CreateProcessAsUserA”, “CreateProcessAsUserW”, “CreateProcessA”, “CreateProcessW” and more³⁵⁴.

Lastly, it is important to know that .NET 2.0 is not compatible with “Import Address Filtering”. Thus, if we want to execute .NET 2.0 code we can’t leverage “Import Address Filtering”³⁵⁵. Also, we can use the “Set-Processmitigation” PowerShell cmdlet for setting the IAF (using the “EnableImportAddressFilter” mitigation parameter keyword) mitigation³⁵⁶. By the way, the as with other mitigations (like ACG) the configuration is saved in the registry per process in the following location: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options” under the relevant process name³⁵⁷.



³⁵² <https://i.blackhat.com/USA21/Wednesday-Handouts/us-21-Rope-Bypassing-Behavioral-Detection-Of-Malware-With-Distributed-ROP-Driven-Execution-wp.pdf>

³⁵³ <https://github.com/vardenshafir/MitigationFlagsCliTool>

³⁵⁴ <https://learn.microsoft.com/en-us/defender-endpoint/exploit-protection-reference>

³⁵⁵ <https://learn.microsoft.com/en-us/defender-endpoint/enable-exploit-protection>

³⁵⁶ <https://learn.microsoft.com/en-us/powershell/module/processmitigations/set-processmitigation>

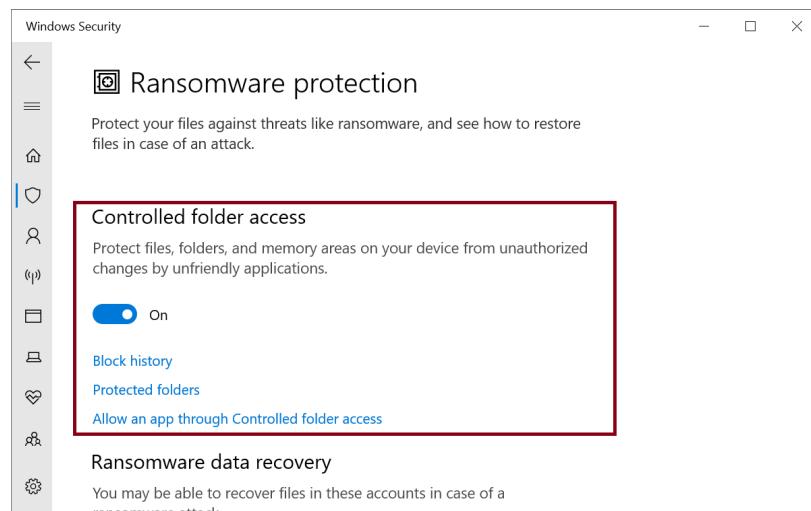
³⁵⁷ <https://medium.com/@boutnaru/the-windows-security-journey-acg-arbitrary-code-guard-74b08a8bd1e5>

Controlled Folder Access

“Controlled Folder Access” is a ransomware protection feature included as part of the Windows operating system (since Windows 10). It is used for blocking suspicious programs from writing to protected folders. In order for it to work it requires “Windows Defender AV” but it works best with “Windows Defender ATP”. Example for enabling it are (but not limited to): “System Settings->Windows Security->Ransomware Protection” (as shown in the screenshot below), GPO (“Windows components->Windows Defender Antivirus->Windows Defender Exploit Guard->Controlled Folder Access”) or by leveraging PowerShell with the command “Set-MpPreference -EnableControlledFolderAccess Enabled”³⁵⁸.

Overall, “Controlled Folder Access” operates based on a list of trusted applications. Applications which are part of the list work as expected while applications that are not part of the list are blocked/prevented from making any changes to files located at protected folders. Applications are added to the list based on their prevalence and/or reputation. If an application is prevalent and has never displayed any behavior deemed malicious it is added automatically to the list. Of course we can also add applications manually to the list³⁵⁹.

Lastly, the list of the protected folders by default contains the following: “%userprofile%\Documents”, “c:\Users\Public\Documents”, “%userprofile%\Pictures”, “c:\Users\Public\Pictures”, “c:\Users\Public\Videos”, “%userprofile%\Videos”, “%userprofile%\Music”, “c:\Users\Public\Music” and “%userprofile%\Favorites”³⁶⁰.



³⁵⁸ <https://slideplayer.com/slide/13416611/>

³⁵⁹ <https://learn.microsoft.com/en-us/defender-endpoint/controlled-folders>

³⁶⁰ https://wiki.nexusmods.com/index.php/Controlled_Folder_Access

SAC (Smart App Control)

SAC (Smart App Control) is a security feature part of the Windows operating system which is used for protecting against untrustworthy applications. Thus, when trying to execute an application on Windows its credibility is being analyzed by SCA. This is done while leveraging the Microsoft's Intelligent Security Graph³⁶¹.

Overall, if SCA is not able to make a confident prediction, it then checks the application for a valid signature (that identifies where the file originates from and if it's a trusted source). In case the application is from a known publisher or with a valid signature it will execute (if not it is blocked - as shown in the screenshot below). By the way, there's no manual override option, nor can you add an app to an exemption list³⁶².

Lastly, smart app control is only available on clean installations of Windows 11. It is started in evaluation mode, which can determine that the specific device is not a good candidate for SAC. It is important to understand that we have to enable “optional diagnostic data” in Windows for SCA to work. The settings of SCA can be found in the “App & Browser Security” panel of the Windows security app³⁶³.



³⁶¹ <https://medium.com/@boutnaru/the-windows-security-journey-microsoft-isg-microsoft-intelligent-security-graph-a32d787122b9>

³⁶² <https://www.howtogeek.com/smart-app-control-windows-11-explained/>

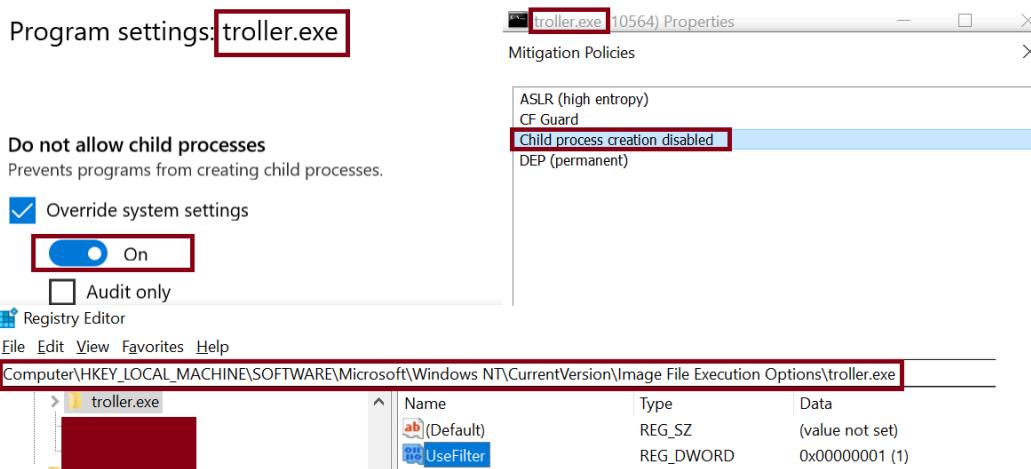
³⁶³ <https://support.microsoft.com/en-gb/topic/what-is-smart-app-control-285ea03d-fa88-4d56-882e-6698afdb7003>

No Child Process Mitigation

The “No Child Process” security mitigation has been designed to block code execution by preventing the launching of child processes³⁶⁴. The mitigation is enforced as a flag in the token of the process, we can check for the flag using the “SeTokenIsNoChildProcessRestrictionEnforced” kernel function³⁶⁵. The mitigation blocks the creation of a token for a child process (and thus a creation of a process) while returning “STATUS_CHILD_PROCESS_BLOCKED”³⁶⁶.

Overall, one of the common techniques leveraged by attacks is to initiate a trusted process with malicious parameters/input (aka “living off the land”). This causes the launching/starting of another executable /application by using the “No Child Process” mitigation to prevent (we can also use “audit mode” only if we want) this attack vector³⁶⁷. By the way, the as with other mitigations (like ACG) the configuration is saved in the registry per process in the following location: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options” with the “UseFilter” under the relevant process name³⁶⁸ - as shown in the screenshot below (the left side is from SystemInformer³⁶⁹).

Lastly, this mitigation is used for example as part of “Microsoft Edge”. It is set for the content processes which ensures that a child process cannot be created. By having the mitigation set as part of the token it ensures both direct (think about calling WinExec) and indirect (for example out-of-process COM server) process launches are blocked³⁷⁰.



³⁶⁴ <https://www.ghacks.net/2017/09/02/windows-10-threat-mitigation-advancements/>

³⁶⁵ <https://learn.microsoft.com/en-us/windows-hardware/drivers/ddi/ntifs/nf-ntifs-setokenisnochilprocessrestrictionenforced>

³⁶⁶ https://codemachine.com/downloads/win10_1709/nstatus.h

³⁶⁷ <https://learn.microsoft.com/en-us/defender-endpoint/exploit-protection-reference>

³⁶⁸ <https://medium.com/@boutinaru/the-windows-security-journey-acg-arbitrary-code-guard-74b08a8bd1e5>

³⁶⁹ <https://github.com/winsiderss/systeminformer>

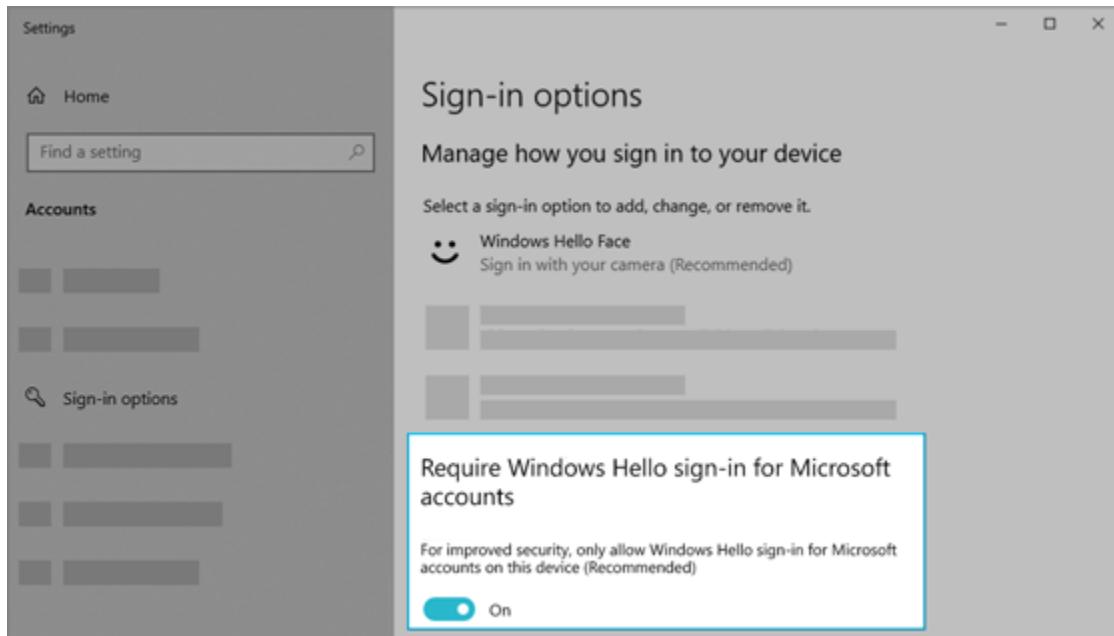
³⁷⁰ <https://blogs.windows.com/msedgedev/2017/02/23/mitigating-arbitrary-native-code-execution/>

Windows Hello

In general, “Windows Hello” is a more secure and personal way to access Windows devices. It provides the ability to use PIN code, face recognition or fingerprint authentication. When setting up a fingerprint/facial recognition we will need to set up a PIN also. After that we can sign in with just the PIN code. This is safer because the PIN code is only associated with one device and is not back up for recovery using a Microsoft account³⁷¹.

Overall, by using Windows it eliminates the need for passwords. Thus, it is a new way to sign into devices, apps, online services and networks. Even if our Windows 10/11 device supports Windows Hello we are not required to use it. For facial recognition it requires a specialized illuminated infrared camera or a fingerprint reader which supports the Window-Biometric Framework³⁷².

Lastly, “Windows Hello” provides the ability to go passwordless on Windows based devices. We just need to configure “Require Windows Hello sign-in for Microsoft Accounts”. To do so go to “WinKey->Settings->Accounts->Sign-in Options” - as shown in the screenshot below³⁷³.



³⁷¹ https://support.microsoft.com/en-us/windows/learn-about-windows-hello-and-set-it-up-dae28983-8242-bb2a-d3d1-87c9d265a5f0#WindowsVersion=Windows_11

³⁷² <https://www.dell.com/support/kbdoc/en-us/000148929/what-is-windows-hello>

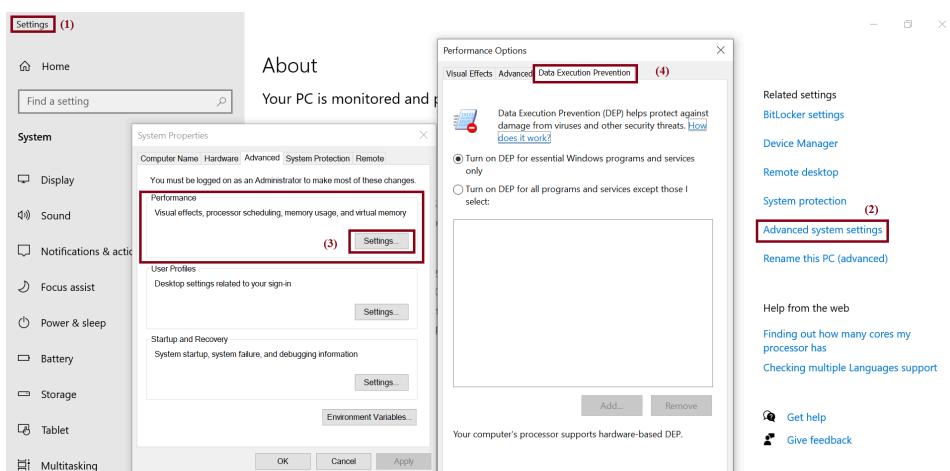
³⁷³ <https://support.microsoft.com/en-us/windows/go-passwordless-on-your-device-5035693d-8f87-c5d5-d16c-b2f75d37bb37>

DEP (Data Execution Prevention)

DEP (Data Execution Prevention) is a security feature built into the Windows operating system (since Windows XP/2003). It allows marking memory pages as non-executable, this means the code cannot be run from those regions of memory (for example the stack and the heap). Thus, making it harder to exploit security bugs such as buffer overflow (of course this is not the only feature for that). DEP is basically the implementation of the OS side required for completing the support for “NX bit” in the CPU³⁷⁴. For configuring DEP on Windows we can go to “Settings”->“Advanced System Settings” -> “Performance” (Settings)->“Data Execution Prevention” - as shown in the screenshot below.

Overall, if an application tries to jump/call an address which resides in a protected page an exception of “STATUS_ACCESS_VIOLATION” is raised. To avoid that we need to allocate memory using one of the following (using the VirtualAlloc/VirtualAllocEx API function) flags: “PAGE_EXECUTE”, “PAGE_EXECUTE_READ”, “PAGE_EXECUTE_READWRITE” or “PAGE_EXECUTE_WRITECOPY”³⁷⁵.

Moreover, a process can get its current DEP policy using the “GetSystemDEPPolicy” function call the return value can be: “0” (always off), “1” (always on), “2” (opt in) or “3” (opt out)³⁷⁶. In order to enable/disable DEP for a specific process we can call the “SetProcessDEPPolicy” API function³⁷⁷. Lastly, we can also change the protection of a the current process using “VirtualProtect”³⁷⁸ or even any other process (based on permissions) using “VirtualProtectEx”³⁷⁹. By the way there is also a software implementation of DEP which we have not focused on³⁸⁰.



³⁷⁴ <https://medium.com/@boutinari/security-nx-bit-non-executable-18759fd2802e>

³⁷⁵ <https://learn.microsoft.com/en-us/windows/win32/Memory/memory-protection-constants>

³⁷⁶ <https://learn.microsoft.com/en-us/windows/win32/api/WinBase/nf-winbase-getsystemdeppolicy>

³⁷⁷ <https://learn.microsoft.com/en-us/windows/win32/api/WinBase/nf-winbase-setprocessdeppolicy>

³⁷⁸ <https://learn.microsoft.com/en-us/windows/win32/api/memoryapi/nf-memoryapi-virtualprotect>

³⁷⁹ <https://learn.microsoft.com/en-us/windows/win32/api/memoryapi/nf-memoryapi-virtualprotectex>

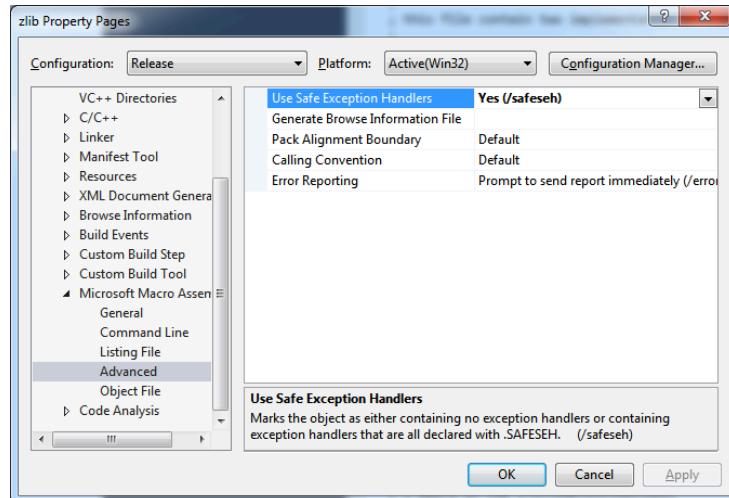
³⁸⁰ <https://learn.microsoft.com/en-us/windows/win32/memory/data-execution-prevention>

SafeSEH (Safe Structured Exception Handling)

The goal of the SafeSEH (Safe Structured Exception Handling) mechanism is to protect stack-based SEH³⁸¹ chains from overwriting. It is important to know that in case of x86/Itanium/ARM architectures the exception handlers are already stored in a table as one of the PE sections (i.e. PDATA). Thus, it can't be overwritten directly as part of a stack overflow³⁸².

Overall, for x86 code projects in case we use “Visual Studio” we can provide the “/SAFESEH” flag. This causes a creation of a PE image with a table of safe exception handlers³⁸³. We can configure it as part of the “Microsoft Macro Assembler” settings as shown in the screenshot below³⁸⁴.

Lastly, this means that if the flow of an application is changed within the exception handler chain and the address of the execution handler is not within such a table the application is terminated. By doing so, SafeSEH prevents the code execution through an exploited exception handler chain. As with other security mechanisms also SafeSEH can be paypassed using techniques like: finding a module which is not protected by SafeSEH and/or using addresses outside the range of the loaded module/the application image which are consider safe by SafeSEH³⁸⁵.



³⁸¹ <https://medium.com/@boutnaru/the-windows-concept-journey-seh-structured-exception-handling-10458086118a>

³⁸² <https://security.stackexchange.com/questions/23315/safeseh-and-x64>

³⁸³ <https://learn.microsoft.com/en-us/cpp/build/reference/safeseh-image-has-safe-exception-handlers?view=msvc-170>

³⁸⁴ <https://www.gangofcoders.net/solution/error-lnk2026-module-unsafe-for-safeseh-image/>

³⁸⁵ <https://www.reesecurity.com/2012/11/bypassing-safeseh-memory-protection-in-zoner-photo-studio-v15/>

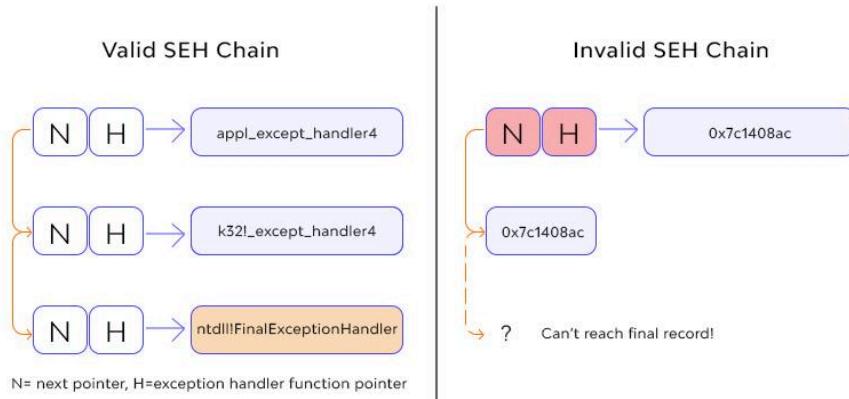
SEHOP (Structured Exception Handling Overwrite Protection)

The goal of the SEHOP (Structured Exception Handling Overwrite Protection) mechanism is to protect stack-based exception chains from overwriting. It is important to know that in case of x64/Itanium architectures the exception handlers are stored in a different PE section³⁸⁶ (by default it is called “PDATA”).

Overall, SEHOP verifies the integrity of SEH³⁸⁷ chains before executing exception handling code. This mechanism works at runtime regardless if we use SafeSEH³⁸⁸ or not. SafeSEH is a build time solution (the support is added by the linker).

Moreover, SEHOP is based on the assumption that an SEH overwrite has side effects. When used (in most of stack-based buffer overflows) the next pointer of the exception registration is overwritten. Since the next pointer is corrupted the it invalidates the integrity exception chain³⁸⁹ - as shown in the diagram below³⁹⁰.

Lastly, SEOP adds a dummy record at the top of the SEH list. In case an exception exception is triggered the dispatcher walks up the list and verifies that the dummy record is valid (was not changed), if not the process is terminated³⁹¹.



³⁸⁶ <https://security.stackexchange.com/questions/23315/safeseh-and-x64>

³⁸⁷ <https://medium.com/@boutnaru/the-windows-concept-journey-seh-structured-exception-handling-10458086118a>

³⁸⁸ <https://medium.com/@boutnaru/the-windows-security-journey-safeseh-safe-structured-exception-handling-835b6b938468>

³⁸⁹ <https://msrc.microsoft.com/blog/2009/02/preventing-the-exploitation-of-structured-exception-handler-seh-overwrites-with-sehop>

³⁹⁰ <https://www.wallarm.com/what/buffer-overflow-attack-preventing-and-mitigation-methods-part-2>

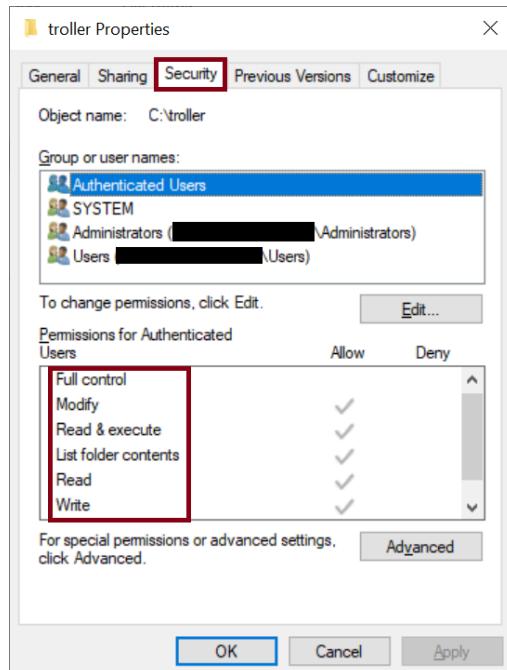
³⁹¹ <https://www.slideserve.com/christyrose/control-hijacking-defenses-powerpoint-ppt-presentation>

NTFS (New Technology File System) Permissions

NTFS (New Technology File System) is the default file system used on Windows based devices³⁹². NTFS supports different features like: compression, quotas, ads, journaling and more. Here we are going to focus on the ability of setting permissions to files/folders.

Overall, we can group NTFS permissions to the following categories: “Full Control”, “Modify”, “Read & Execute”, “List folder contents”, “Read” and “Write” - as shown in the screenshot below. All of those are composed of a specific set of advanced (special) permissions³⁹³.

Lastly, for each file/folder we have a security descriptor³⁹⁴ which holds the access control list relevant for permissions aka DACL³⁹⁵ - as seen in the screenshot below. We can also see that information using CLI tools like “cacls.exe”³⁹⁶ and “icacls.exe”³⁹⁷.



³⁹² <https://medium.com/@boutnaru/the-windows-concept-journey-ntfs-new-technology-file-system-433e27a2256a>

³⁹³ <https://www.permissionsreporter.com/ntfs-permissions>

³⁹⁴ <https://medium.com/@boutnaru/windows-security-security-descriptor-sd-ba95b8fa048a>

³⁹⁵ <https://medium.com/@boutnaru/the-windows-security-journey-dacl-discretionary-access-control-list-c74545e472ec>

³⁹⁶ <https://medium.com/@boutnaru/the-windows-process-journey-cacls-exe-control-acls-program-296ba9e7761c>

³⁹⁷ <https://ss64.com/n/icacls.html>

Null Sessions

A null session means that we can access a network resource without any authentication (user->null and password->null). One way of doing that is to connect to the hidden IPC\$ share³⁹⁸ over SMB (Server Message Block)CIFS (Common Internet File System) - as shown in the screenshot for Wireshark below³⁹⁹. This can be done for example by leveraging the “net.exe” utility⁴⁰⁰ in the following manner: net use \\[IPv4|IPv6|FQDN|NetBios Name]IPC\$ "" /u: "" - also shown below.

Overall, based on the system configuration (described later) using null sessions we can obtain different information such as (but not limited to): groups, shares, permissions, services, usernames⁴⁰¹, patch information, audit policy, domain information and account policy⁴⁰².

Lastly, the relevant configuration is stored as part of the registry⁴⁰³ as follows: “HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymous” (if set to “1”, null sessions can not be used to enumerate shares), “HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymousSAM” (“1” is the default setting. It means null sessions can not enumerate usernames) and “HKLM\System\CurrentControlSet\Control\Lsa\EveryoneIncludesAnonymous” (“0” is the default setting. It means null sessions don’t have any special rights). We could also configure those setting using GPO⁴⁰⁴ in the following location: “Computer configuration\Policies\Windows settings\Security Settings\Local Policies\SecurityOptions”⁴⁰⁵

The Wireshark screenshot displays a sequence of network frames. The traffic starts with a TCP SYN from port 57500 to port 445. This is followed by an ACK from port 445 to port 57500. A Negotiate Protocol Request is sent from the client, followed by a Negotiate Protocol Response from the server. An ACK from port 57500 to port 445 is shown, indicating the session setup is complete. The next frame is a Session Setup AndX Request from the client, with the user field set to "anonymous". This is followed by a Session Setup AndX Response from the server. Subsequent frames show Tree Connect AndX requests and responses, specifically targeting the "\\192.168.57.105\IPC\$". DCERPC Bind and Write AndX requests are also visible, along with a Read AndX request. The final frame is a DCERPC Bind_ack response.

³⁹⁸ <https://medium.com/@boutnaru/the-windows-concept-journey-ipc-inter-process-communication-share-2497a899b1ff>

³⁹⁹ <https://sensepost.com/blog/2018/a-new-look-at-null-sessions-and-user-enumeration/>

⁴⁰⁰ <https://medium.com/@boutnaru/the-windows-process-journey-net-exe-net-command-91e4964f20b8>

⁴⁰¹ <https://www.blumira.com/glossary/null-session>

⁴⁰² <https://www.utm.edu/sites/default/files/2021-04/4660-lab4.pdf>

⁴⁰³ <https://medium.com/@boutnaru/the-windows-concept-journey-registry-0767e79387a9>

⁴⁰⁴ <https://medium.com/@boutnaru/the-windows-security-journey-gpo-group-policy-object-f8757933217a>

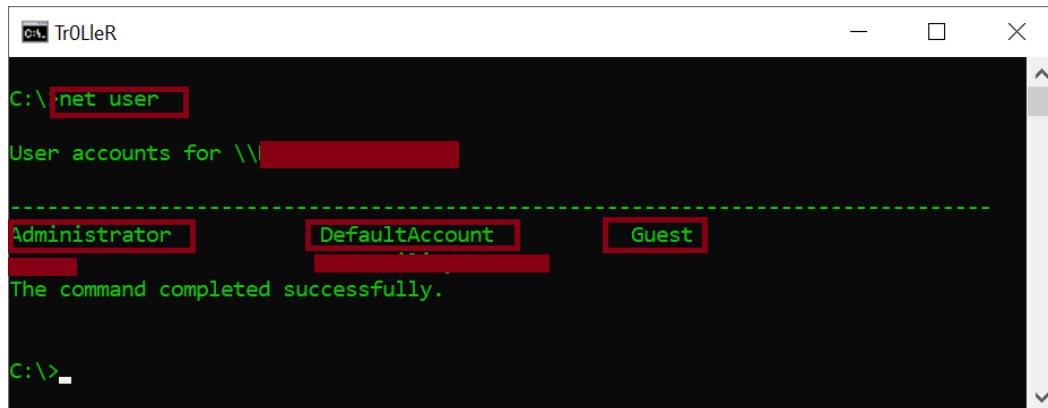
⁴⁰⁵ <https://www.blumira.com/integration/how-to-disable-null-session-in-windows/>

Local User Account

In general, a “Local User Account” is a Windows account which was created on the local device. This type of account can only logon to a specific device (the local machine) and due to that is used for managing/securing resources on a specific device or as service users⁴⁰⁶.

Moreover, the information regarding local user accounts on Windows is stored as part of the SAM⁴⁰⁷ (Security Account Manager) which itself is part of the registry. Also, Windows has default local accounts that are created when the operating system is installed. Examples of such users are: Administrator, Guest, DefaultAccount, Local System, Network Service, Local Service⁴⁰⁸.

Lastly, we can also list the local users accounts of a specific computer using the “user” argument of the “net.exe” command line utility - as shown in the screenshot below⁴⁰⁹. It is important to note that there are also local groups that can be created.



The screenshot shows a terminal window titled "Tr0LleR". The command "net user" is entered at the prompt "C:\>". The output shows "User accounts for \\" followed by a list of accounts: "Administrator", "DefaultAccount", and "Guest". A message "The command completed successfully." is displayed at the bottom. The entire terminal window is framed by a red border.

⁴⁰⁶ <https://learn.microsoft.com/en-us/windows/security/identity-protection/access-control/local-accounts>

⁴⁰⁷ <https://medium.com/@boutnaru/windows-security-sam-security-account-manager-c93ddadf388a>

⁴⁰⁸ <https://learn.microsoft.com/en-us/windows/security/identity-protection/access-control/local-accounts>

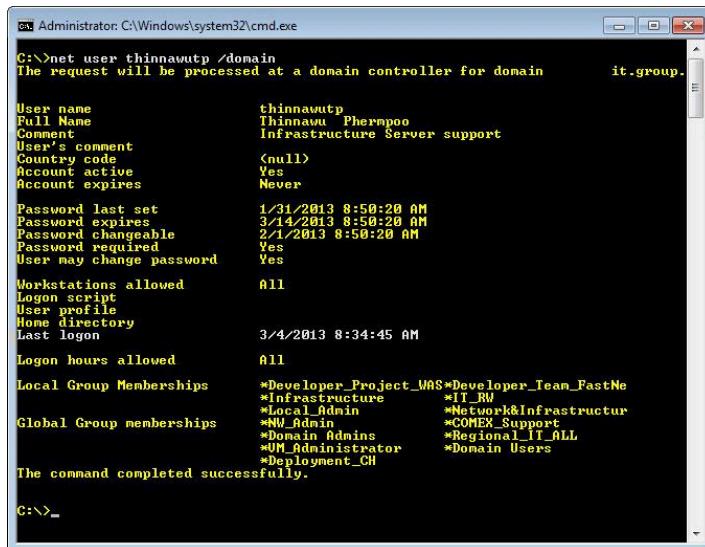
⁴⁰⁹ <https://medium.com/@boutnaru/the-windows-process-journey-net-exe-net-command-91e4964f20b8>

Domain User Account

By using a domain user account we get the full advantage of the administration and security features provided by “Microsoft Active Directory”. Thus, a domain user account provides the ability to log on into systems that are part of a MS domain⁴¹⁰. As opposed to a local user account which is stored locally in the SAM⁴¹¹, a domain user account is stored remotely (can be cached locally) in the AD database as part of a “Domain Controller”.

Overall, for administering domain user accounts we can use both CLI tools and GUI based tools. Probably the most commonly used GUI tool is “Active Directory and Computers” aka “dsa.msc”⁴¹². In the case of CLI we can use also the “net.exe”⁴¹³ utility which is relevant for local users\group management we just need to add the “/domain” switch to part of the commands - as shown in the screenshot below⁴¹⁴.

Lastly, a domain user account has two name formats: the DN (distinguished name) of the user object and “"\<username>"⁴¹⁵, where <domain> is the NETBIOS name of the domain or the FQDN (Fully Qualified Domain Name) of the domain. Also, a user in a domain can have a UPN (User Principal Name) which we can think of as a username in an email address format: <username>@<domain>⁴¹⁶.



The screenshot shows a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The command entered is "net user thinnavutp /domain". The output displays detailed information about the user account "thinnavutp" including its full name ("Thinnavu Phermppoo"), comment ("Infrastructure Server support"), and various account settings like password expiration and logon hours. It also lists local and global group memberships. The command concludes with "The command completed successfully."

```
C:\>net user thinnavutp /domain
The request will be processed at a domain controller for domain      it.group.

User name          thinnavutp
Full Name          Thinnavu Phermppoo
Comment           Infrastructure Server support
User's comment
User's comment
Country code       <null>
Account active    Yes
Account expires   Never
Password last set 1/31/2013 8:50:20 AM
Password expires   3/14/2013 8:50:20 AM
Password changeable 2/1/2013 8:50:20 AM
Password required  Yes
User may change password Yes
Workstations allowed All
Logon script
User profile
Home directory
Last logon        3/4/2013 8:34:45 AM
Logon hours allowed All
Local Group Memberships
*Developer_Project_WAS *Developer_Team_FastNet
*Infrastructure *IT_BU
*Local_Admin *Network&Infrastructure
*NU_Admin *COMEX_Support
*Domain Admins *Regional_IT_ALL
*UM_Administrator *Domain Users
*Deployment_CH

Global Group memberships
The command completed successfully.

C:\>_
```

⁴¹⁰ <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-a9e795f86b4>

⁴¹¹ <https://medium.com/@boutnaru/the-windows-security-journey-local-user-account-effe75db39d8>

⁴¹² <https://woshub.com/install-active-directory-users-computers-aduc-console/>

⁴¹³ <https://medium.com/@boutnaru/the-windows-process-journey-net-exe-net-command-91e4964f20b8>

⁴¹⁴ <http://serveradmin-tools.blogspot.com/2013/03/details-and-examples-net-user-domain.html>

⁴¹⁵ <https://learn.microsoft.com/en-us/windows/win32/ad/domain-user-accounts>

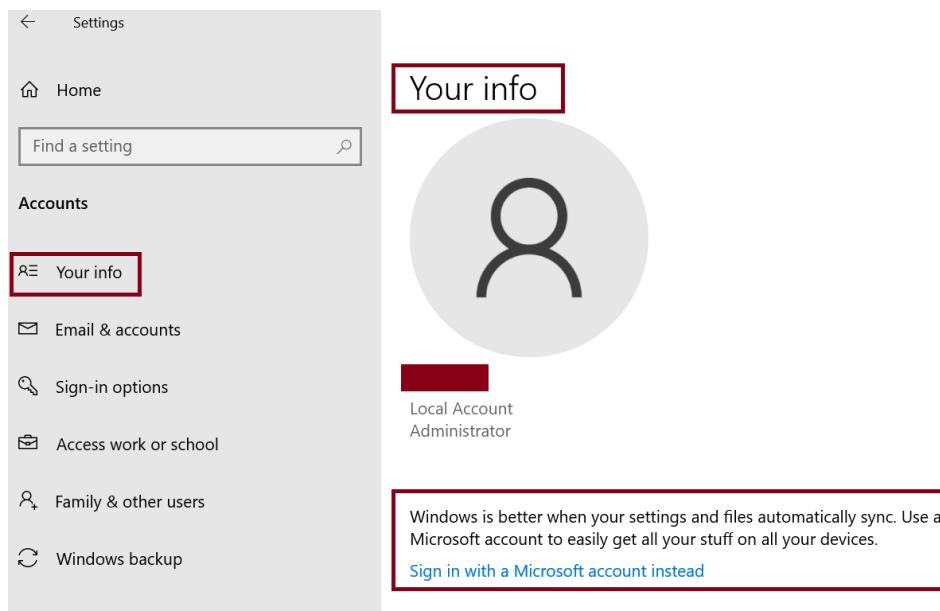
⁴¹⁶ <https://www.codetwo.com/kb/upn/>

Microsoft Account (MSA)

MSA (Microsoft Account) is an SSO (Single Sign On) account that can be used for accessing different Microsoft's services (Bing, Outlook.com, Microsoft Azure and more), devices running the Windows OS (Windows 8 and later\Windows Server 2012 and later) and other software created by Microsoft (Visual Studio, Skype, Windows Movie Maker and more). It was previously named “Microsoft Passport”, “.NET Passport” and “Windows Live ID”⁴¹⁷.

Overall, in order to create an MSA we can use an existing email and/or sign up for a Microsoft email. address. The usage of MSA can also be leveraged by organizations by allowing: the download of Microsoft store apps (those which the enterprise wants to distribute), allow for personalized settings across devices, integrated social media services and more⁴¹⁸.

Lastly, is it important to understand that even if we use MSA to log on to the system, we can still leverage local user accounts⁴¹⁹ as a login option. By the way, the same is also true if we use a domain account to log on to our system. If an administrator wants he\she can block the usage for MSA for user authentication using configurations as part of GPO⁴²⁰.



⁴¹⁷ https://en.wikipedia.org/wiki/Microsoft_account

⁴¹⁸ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-microsoft-accounts>

⁴¹⁹ <https://medium.com/@boutnaru/the-windows-security-journey-local-user-account-cffe75db39d8>

⁴²⁰ <https://medium.com/@boutnaru/the-windows-security-journey-gpo-group-policy-object-f8757933217a>

Local Group

A “Local Group” on Windows is a group defined as part of the SAM database⁴²¹ on a specific workstation\server. Thus, a local group is only relevant for the system on which it is configured (as opposed to a domain group account). Hence, is a workstation\server which is part of a “Workgroup”⁴²² and not a domain member we only can use the local users⁴²³ and local groups.

Overall, a local group can have members which are: local user accounts, local groups and global groups from one or more domains⁴²⁴. As with local\domain users and domain groups also local groups have an SID⁴²⁵ associated with them which can be included as part of an access token⁴²⁶. For example the SID of the BUILTIN\Administrators group (the local administrator group) is “S-1-5-32-544”⁴²⁷.

Lastly, the NetAPI.dll located at “%windir%\System32\netapi32.dll” (on 64-bit systems we also have a 32-bit version “%windir%\SysWOW64\netapi32.dll”) exports different functions for managing local groups (delete/add/modify/etc) such as: “NetLocalGroupAdd”, “NetLocalGroupAddMembers”, “NetLocalGroupDel”, “NetLocalGroupDelMembers” and more⁴²⁸. Also, we can use the “net.exe”⁴²⁹ command line utility for local group management using the “localgroup” argument - as shown in the screenshot below.

The screenshot shows an Administrator Command Prompt window. The user runs the command `net localgroup`, which lists various local groups like Administrators, Backup Operators, etc. Then, the user runs `net localgroup | findstr /i troller` to filter the output for the word 'troller'. Finally, the user adds the 'troller' group to the Administrators group with the command `net localgroup troller /add`, and the output shows the command completed successfully. The entire session is as follows:

```
C:\>net localgroup
Aliases for \\[REDACTED]
-----
*Access Control Assistance Operators
*Administrators
*Backup Operators
*Cryptographic Operators
*Device Owners
*Distributed COM Users
*Event Log Readers
*Guests
*Hyper-V Administrators

[REDACTED]

*Power Users
*Remote Desktop Users
*Remote Management Users
*Replicator

*System Managed Accounts Group
*Users
The command completed successfully.

C:\>net localgroup | findstr /i troller
[REDACTED]
E:\>net localgroup troller /add
The command completed successfully.

E:\>net localgroup | findstr /i troller
troller
```

⁴²¹ <https://medium.com/@boutnaru/windows-security-sam-security-account-manager-c93ddadfc388a>

⁴²² <https://medium.com/@boutnaru/the-windows-security-journey-windows-workgroup-b886a0f07f83>

⁴²³ <https://medium.com/@boutnaru/the-windows-security-journey-local-user-account-cffe75db39d8>

⁴²⁴ <https://learn.microsoft.com/en-us/windows/win32/netmgmt/local-group-functions>

⁴²⁵ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

⁴²⁶ <https://medium.com/@boutnaru/windows-security-access-token-81cd00000c64>

⁴²⁷ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-identifiers>

⁴²⁸ https://windows10dll.mirsoft.net/netapi32_dll.html

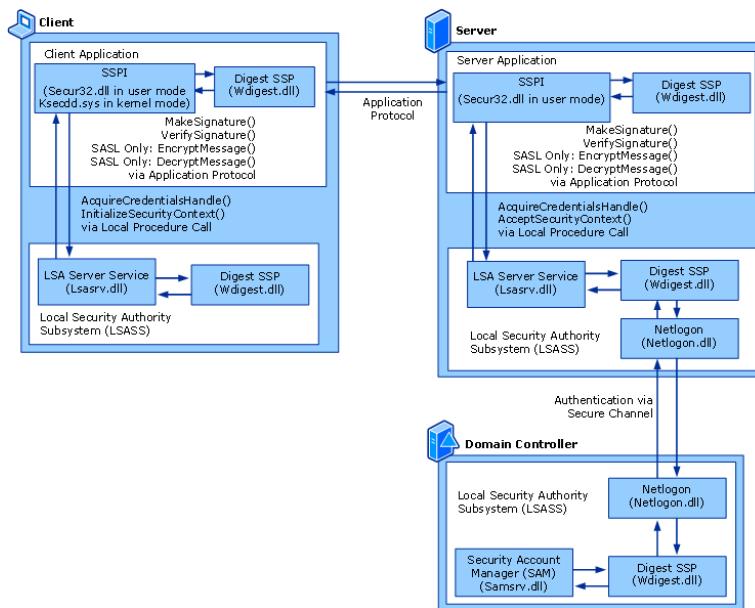
⁴²⁹ <https://medium.com/@boutnaru/the-windows-process-journey-net-exe-net-command-91e4964f20b8>

WDigest (Windows Digest)

WDigest is an authentication protocol which was introduced as part of Windows XP. It is designed for HTTP (Hypertext Transfer Protocol) authentication or SASL (Simple Authentication Security Layer) based on RFC 2617 and 2831⁴³⁰. WDigest is enabled by default from Windows XP to Windows 8 in case of clients and from Windows 2003 to Windows 2012 in case of servers⁴³¹.

Moreover, WDigest is implemented as part of “%windir%\System32\wdigest.dll” (which is digitally signed by Microsoft). By the way, in the case of a 64-bit version of Windows there is also a 32-bit implementation located at “%windir%\SysWOW64\wdigest.dll” - the digest authentication architecture is shown below⁴³². In order to enable WDigest we need to set the “UseLogonCredential” value located at the registry location “HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest” to “1”⁴³³.

Lastly, using WDigest causes “lsass.exe”⁴³⁴ to store the passwords of those which logged on in cleartext (which can be extracted when reading/dumping memory like with Mimikatz). Thus, using the credentials stored in cleartext is used by the OS to authenticate without requesting users to enter their passwords again⁴³⁵.



⁴³⁰ <https://www.elastic.co/guide/en/security/current/modification-of-wdigest-security-provider.html>

⁴³¹ <https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections>

⁴³² [https://learn.microsoft.com/pt-pt/previous-versions/windows/server/cc778868\(v=ws.10\)](https://learn.microsoft.com/pt-pt/previous-versions/windows/server/cc778868(v=ws.10))

⁴³³ <https://learn.microsoft.com/en-us/answers/questions/463368/disabling-credentials-caching-in-wdigest>

⁴³⁴ <https://medium.com/@boutnaru/the-windows-process-journey-lsass-exe-local-security-authority-process-24166cb0358f>

⁴³⁵ <https://www.triskelelabs.com/blog/wdigest-extracting-passwords-in-cleartext>

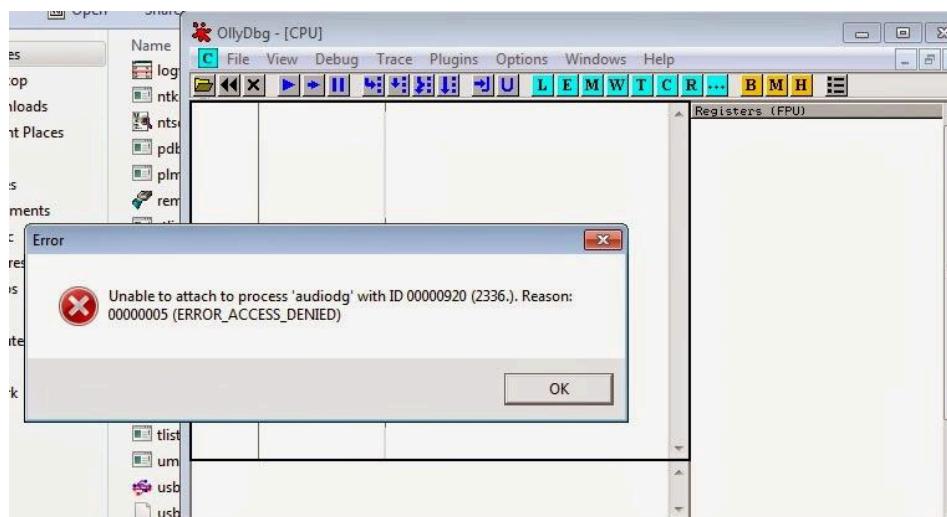
Protected Processes

“Protected Processes” is a security feature introduced as part of Windows Vista in order to enhance support for DRM (Digital Right Management) functionality on Windows. Those processes are running in parallel to normal processes⁴³⁶.

Overall, the difference between a normal process and a protected process is the level of access that other processes in the system can obtain to protected processes. Thus, example of limitations of operations performed by a normal process on protected process are: inability to debug an active protected process, can’t duplicate an handle from a protected process, accessing the virtual memory of a protected process and injecting a thread into a protected process⁴³⁷.

Thus, in case of protected processes we can’t debug them in user mode even if we are running as an administrator, “Local System”⁴³⁸ or “TrustedInstaller”⁴³⁹ - as shown in the screenshot below⁴⁴⁰. However, we can use a kernel debugger/driver in order to disable the protection on a specific process or alter the protection level of our user-mode debugger⁴⁴¹.

Lastly, from Vista only a bit field (ProtectedProcess) as part “struct _EPROCESS”⁴⁴² was used to mark a process as protected. Since Windows 8.1, due to signer types and PPL (Protected Processes Light) we have a “Protection” field from type “struct _PS_PROTECTION”⁴⁴³.



⁴³⁶ https://web.archive.org/web/20070515091802/http://www.microsoft.com/whdc/system/vista/process_Vista.mspx

⁴³⁷ https://web.archive.org/web/20090124094639/http://download.microsoft.com/download/a/f/7/af7777e5-7ddc-4800-8a0a-b18336565f5b/process_Vista.doc

⁴³⁸ <https://medium.com/@boutnaru/the-windows-security-journey-local-system-nt-authority-system-f087dc530588>

⁴³⁹ <https://medium.com/@boutnaru/the-windows-security-journey-trusted-installer-dc44c73ef04c>

⁴⁴⁰ <http://rce4fun.blogspot.com/2014/01/a-quick-dive-into-kernel-debugging.html>

⁴⁴¹ <https://itm4n.github.io/debugging-protected-processes/>

⁴⁴² <https://medium.com/@boutnaru/the-windows-kernel-data-structures-journey-struct-eprocess-executive-process-d43e93c51996>

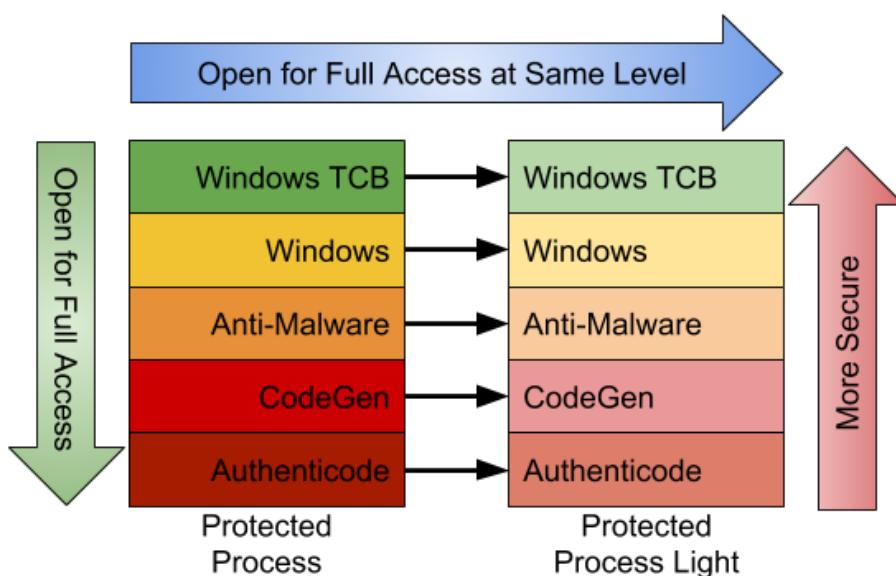
⁴⁴³ <https://www.crowdstrike.com/blog/evolution-protected-processes-part-1-pass-hash-mitigations-windows-81/>

PPL (Protected Processes Light)

PPL (Protected Processes Light) was introduced as a security feature in Windows 8.1. We can think about them as an extension to the concept of protected processes⁴⁴⁴. PPL allows binaries signed by specific keys to execute in such a way that they are immune from tampering\termination even from users with administrative permissions. Also, PPL supports levels of a hierarchy which means higher-privilege applications can't be tampered by lower-privilege applications, but not vice-versa⁴⁴⁵.

Overall, the extension of protected processes to PPL was in order to protect system processes which are not related to DRM (Digital Rights Management). This is done by leveraging digital signatures and code signing, which resembles the entitlement model of iOS⁴⁴⁶.

Lastly, the signing levels were also added to protected processes. Thus, a protected process from one level can open all protected processes and PPLs at the same signing level or below. However, a PPL at any signing level can't open a protected process with full access⁴⁴⁷ - as shown in the diagram below.



⁴⁴⁴ <https://medium.com/@boutnaru/the-windows-security-journey-protected-processes-6451a5fff277>

⁴⁴⁵ <https://www.elastic.co/blog/protecting-windows-protected-processes>

⁴⁴⁶ https://nosuchcon.org/talks/2014/D3_05_Alex_ionescu_Breaking_protected_processes.pdf

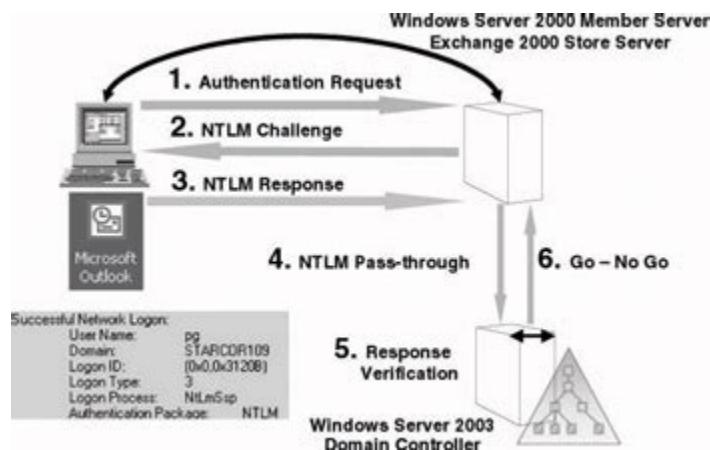
⁴⁴⁷ <https://googleprojectzero.blogspot.com/2018/10/injecting-code-into-windows-protected.html>

Windows Authentication

The goal of an authentication flow is to validate the login information of a specific user. For example comparing a username and password to an authorized list. Microsoft provides different authentication technologies which can be used by developers for authenticating clients/users⁴⁴⁸.

Overall, among the Windows authentication technologies we can find the following: credentials management (obtaining and managing credential data), LSA authentication (handles the logon process to a system), network provider API (interacting with other types of networks), smart card authentication, SSPI, Winlogon and credentials providers⁴⁴⁹.

Lastly, Windows supports different authentication protocols such as: LM, NTLM and Kerberos⁴⁵⁰ - more on them in future writeups. Example is shown in the diagram below⁴⁵¹. Those authentication methods can be used with other protocols like: LDAP, SMB and HTTP⁴⁵².



⁴⁴⁸ <https://learn.microsoft.com/en-us/windows/win32/secauthn/authentication-portal>

⁴⁴⁹ <https://learn.microsoft.com/en-us/windows/win32/secauthn/about-authentication>

⁴⁵⁰ <https://blog.certcube.com/windows-authentication-protocols/>

⁴⁵¹ <https://flylib.com/books/en/2.244.1.26/1/>

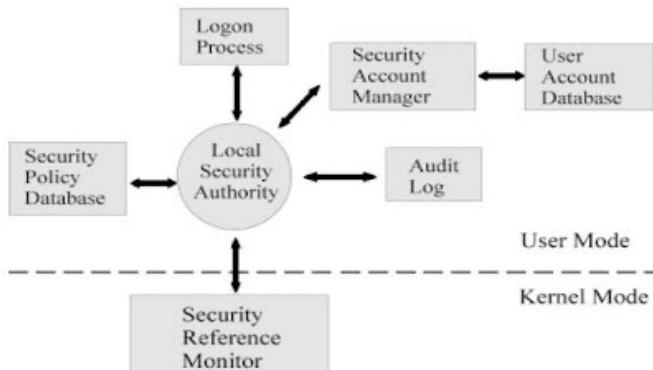
⁴⁵² <https://www.blumira.com/blog/authentication-protocols-101>

LSA (Local Security Authority)

LSA (Local Security Authority) is the security subsystem part of the Windows operating system⁴⁵³. LSA is used in order to authenticate and log users to the local system. LSA also maintains information regarding all aspects of the local security of the operating system⁴⁵⁴.

Overall, it is implemented as part of “%windir%\System32\lsasrv.dll” (LSA Server DLL). Also, it is hosted by the “lsass.exe” (Local Security Authority SubSystem) process⁴⁵⁵. It is one of the basic security components of the Windows operating system together with SAM (Security Account Manager)⁴⁵⁶ and SRM (Security Reference Monitor)⁴⁵⁷.

Lastly, LSA is used in different use cases such as: the logon process, user authentication, auditing, security policy enforcement and more - as demonstrated in the flows diagram below⁴⁵⁸. We can check the reference implementation of “lsasrv.dll” as part of ReactOS⁴⁵⁹.



⁴⁵³ https://learn.microsoft.com/en-usopenspecs/windows_protocols/ms-azod/d28d536d-3973-4c8d-b2c9-989e3a8ba3c5

⁴⁵⁴ <https://learn.microsoft.com/en-us/windows/win32/secgloss/l-gly>

⁴⁵⁵ <https://medium.com/@boutnaru/the-windows-process-journey-lsass-exe-local-security-authority-process-24166cb0358f>

⁴⁵⁶ <https://medium.com/@boutnaru/windows-security-sam-security-account-manager-c93ddadff388a>

⁴⁵⁷ <https://medium.com/@boutnaru/windows-security-srm-security-reference-monitor-d715f96d9fd6>

⁴⁵⁸ <https://youthtalentauzzar.blogspot.com/2012/01/its-windows-security-hackers.html>

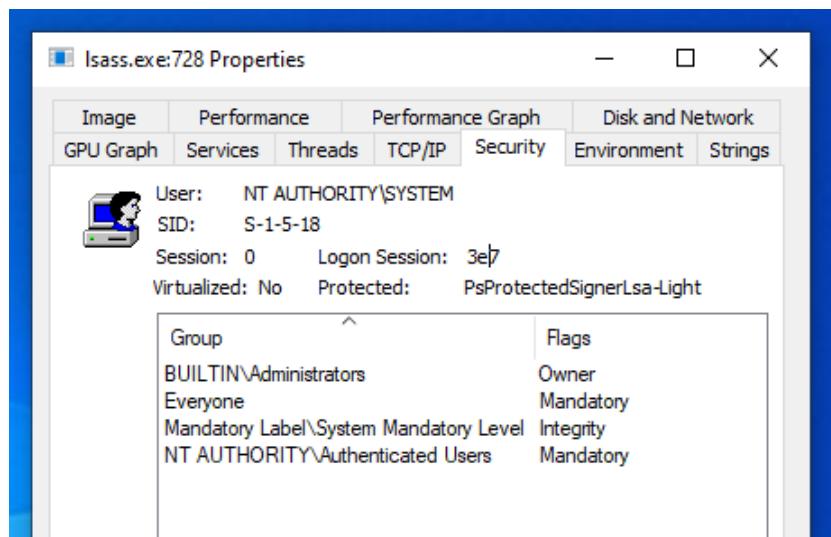
⁴⁵⁹ <https://github.com/reactos/reactos/tree/master/dll/win32/lsasrv>

LSA Protection (Local Security Authority Protection)

“LSA Protection” (Local Security Authority Protection) is a security feature of the Windows operating system which is used to disallow memory reads/code injection targeting the “lsass.exe”⁴⁶⁰ system process. This feature is supported since Windows 8.1 however just since Windows 11 (22H2) it is enabled by default. The last is in case of a newly installed device or if the device is enterprise-joined to a AD and/or Azure domain⁴⁶¹.

Overall, this feature is needed due the fact the memory address spaces of “lsass.exe” contains very sensitive data such as: NT hashes (which can be used for pass the hash attacks), kerberos tickets (which can be used for pass the ticket attacks), credentials in clear text⁴⁶² and more. This security feature can be bypassed in case of kernel level access (arbitrary write/drive). This is also supported by “Mimikatz” which is based on the “mimidrv.sys” driver⁴⁶³.

Lastly, in order to enable LSA protection we need to set “1”/“2” to the value “RunAsPPL” in the following registry location “HKLM\SYSTEM\CurrentControlSet\Control\Lsa”⁴⁶⁴. In this case the LSA protection is based on the PPL protected processes (PsProtectedSignerLsa-Light) mechanism⁴⁶⁵ - as shown in the screenshot below⁴⁶⁶. In case of Windows 11 “lsass.exe” could be protected by “PsProtectedSignerNone” which is based on the protected processes mechanism⁴⁶⁷.



⁴⁶⁰ <https://medium.com/@boutnaru/the-windows-process-journey-lsass-exe-local-security-authority-process-24166cb0358f>

⁴⁶¹ <https://www.mdmandganswers.com/blogs/view-blog/3-ways-to-enable/disable-lsa-on-windows-10-and-11>

⁴⁶² <https://medium.com/@boutnaru/the-windows-security-journey-wdigest-windows-digest-3be028607f62>

⁴⁶³ <https://book.hacktricks.xyz/windows-hardening/staling-credentials/credentials-protections>

⁴⁶⁴ <https://learn.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/configuring-additional-lsa-protection>

⁴⁶⁵ <https://medium.com/@boutnaru/the-windows-security-journey-ppl-protected-processes-light-831d5f371004>

⁴⁶⁶ <https://www.bordergate.co.uk/bypassing-lsa-protections/>

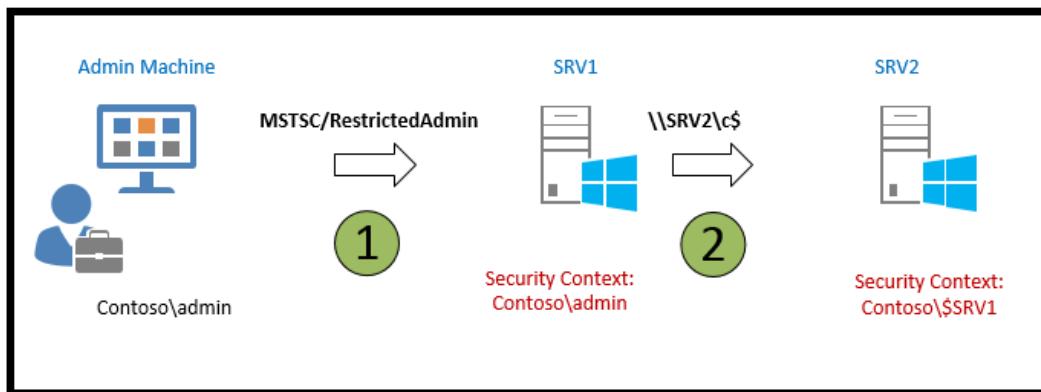
⁴⁶⁷ <https://medium.com/@boutnaru/the-windows-security-journey-protected-processes-6451a5ffl277>

RestrictedAdminMode for RDP (Remote Desktop Protocol Restricted Admin Mode)

RestrictedAdminMode for RDP (Remote Desktop Protocol Restricted Mode) is a security feature that was introduced as part of Windows 8.1/Windows Server 2012 R2 (and later ported to Windows 7). The goal of the feature is to prevent credentials harvesting by avoiding the transmission of reusable credentials for the connecting admin user⁴⁶⁸. By default, when connecting to a remote device using RDP our credentials are also stored there. Due to that, it limits the access of the user to other servers on the network⁴⁶⁹.

Thus, when passing the “/RestrictedAdmin” command line argument to “mstsc.exe”⁴⁷⁰ we are authenticating to a but our credentials are not stored on the remote device. In that case any access over the network would be done using the computer account of the RDP server - as shown in the diagram below⁴⁷¹.

Lastly, we can enable the support of RestrictedAdminMode by setting the value of “DisableRestrictedAdmin” to “0” in the following registry location: “HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa”⁴⁷². We can also enable that using local policy/GPO by setting the “ Restrict delegation of credentials to remote servers” which can be found in “Computer Configuration > Policies > Administrative Templates > System > Delegation of Credentials”⁴⁷³.



⁴⁶⁸ <https://learn.microsoft.com/en-us/archive/technet/wiki/32905.remote-desktop-services-enable-restricted-admin-mode>

⁴⁶⁹ <https://blog.ahasaven.com/restricted-admin-mode-for-rdp/>

⁴⁷⁰ <https://medium.com/@boutnaru/the-windows-process-journey-mstsc-exe-remote-desktop-connection-981bae774bac>

⁴⁷¹ <https://www.anyviewer.com/how-to/restricted-admin-mode-for-remote-desktop-connection-2578.html>

⁴⁷² <https://github.com/GhostPack/RestrictedAdmin>

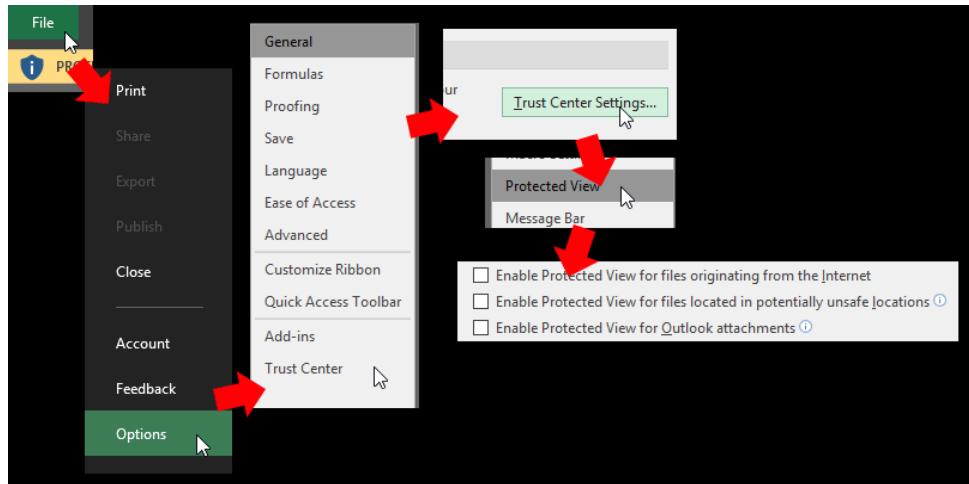
⁴⁷³ <https://4sysops.com/archives/secure-rdp-connections-using-remote-credential-guard/>

Protected View

Due to the fact that opening files downloaded from the Internet (browser/outlook attachment/etc) are unsafe (can contain malware). Because of that files from these potentially unsafe locations are opened as read only or in “Protected View”. Using “Protected View” we can read a file and see its contents and enable editing while reducing the risks. This security feature is relevant for docs opened by: “Excel for Microsoft 365”, “Word for Microsoft 365”, “PowerPoint for Microsoft 365”, “Excel 2024”, “Word 2024”, “PowerPoint 2024”, “Excel 2021”, “Word 2021”, “PowerPoint 2021”, “Excel 2019”, “Word 2019”, “PowerPoint 2019”, “Excel 2016”, “Word 2016” and “PowerPoint 2016”⁴⁷⁴.

Moreover, there was a time that if our system supported “Application Guard for Microsoft 365”⁴⁷⁵ files which previously opened using “Protected View” were opened in “Application Guard for Microsoft 365”. However, “Microsoft Defender Application Guard for Office” is being deprecated and is no longer being updated⁴⁷⁶ - more information about that in future writeups.

Lastly, “Protected View” acts like a sandbox while disabling macros⁴⁷⁷. Also, while opening a document in “Protected View” a yellow dialog box pops up with an “Enable Editing” option. We can also modify the “Protected View” configuration as part of the “Trust Center Settings”- as shown below⁴⁷⁸.



⁴⁷⁴ <https://support.microsoft.com/en-us/office/what-is-protected-view-d6f09ac7-e6b9-4495-8e43-2bbcdcb6653>

⁴⁷⁵ <https://support.microsoft.com/en-us/office/application-guard-for-office-9e0fb9c2-ffad-43bf-8ba3-78f785fdb4a6>

⁴⁷⁶ <https://support.microsoft.com/en-us/office/application-guard-for-office-9e0fb9c2-ffad-43bf-8ba3-78f785fdb4a6>

⁴⁷⁷ <https://superuser.com/questions/1771627/why-does-microsoft-words-protected-view-prevent-editing>

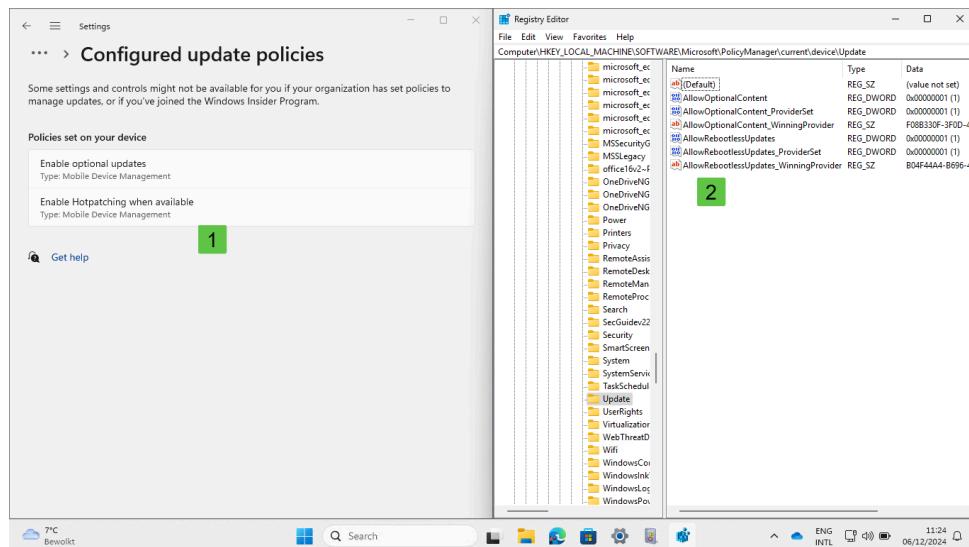
⁴⁷⁸ <https://www.digitalwelt.org/en/tips/software/disable-protected-view>

Hotpatching

By leveraging hotpatching we can install OS security updates on a Windows server based device without the need of restarting it⁴⁷⁹. This feature has been available since 2022 (February) for “Windows Server 2022 Datacenter” (Azure Version) and for “Windows Server Azure Edition” core virtual machines (as part of the GA). In september 2024⁴⁸⁰.

Overall, hotpatching provides different benefits such as (but not limited to): simplified IT management, reporting, increased uptime and productivity, enhanced security and faster protection and smoother user experience⁴⁸¹. We can configure hotpatching in different ways like (but not limited to): using “Microsoft Intune”, using the Setting application or directly modifying the relevant keys in the registry - as shown below⁴⁸².

Lastly, Hotpatch updates are the monthly security updates (also known as the B-release) that can be installed without restarting the device. Hence, it is important to understand that there is a difference between baseline releases and hotpatch releases. Baseline releases contain the latest security fixes, new features, and other enhancements, and still require a reboot, while the hotpatch releases only contain security fixes and don't require a reboot. Also, the baseline releases are released every quarter of the year⁴⁸³.



⁴⁷⁹ <https://learn.microsoft.com/en-us/windows-server/get-started/hotpatch>

⁴⁸⁰ <https://www.bleepingcomputer.com/news/microsoft/microsoft-windows-server-hotpatching-to-require-subscription/>

⁴⁸¹ <https://blog.sones.cloud/windows-hotpatching-is-now-possible-on-clients-rebootless-updates-is-here/>

⁴⁸² <https://petervanderwoude.nl/post/enabling-hotpatch-for-windows-11-enterprise/>

⁴⁸³ <https://learn.microsoft.com/en-us/windows/deployment/windows-autopatch/manage/windows-autopatch-hotpatch-updates>

RunAs (Secondary Logon)

RunAs allows a user to execute specific programs/tools with different permissions than the logged-on user. We can leverage it using the “runas.exe” executable aka “Run As Utility”, which is located at “%windir%\System32\runas.exe”. On 64 bit systems there is also a 32-bit version located at “%windir%\SysWow64\runas.exe”. It has multiple parameters that can be used like passing credentials from a smartcard instead of a password, loading the user’s profile and more⁴⁸⁴.

Overall, RunAs is dependent on the “Secondary Logon” service. The description of the service states that it “enables starting processes under alternate credentials. If this service is stopped, this type of logon access will be unavailable. If this service is disabled, any services that explicitly depend on it will fail to start”. As described if the service is disabled RunAs will fail - as shown in the screenshot below.

Lastly, the “Secondary Logon” service is started with the following command line: “%windir%\system32\svchost.exe -k netsvcs -p -s seclogon” with the permissions of the “Local System” user. Also, in this case “svchost.exe” will load “%windir%\System32\seclogon.dll” (Secondary Logon Service DLL). For a reference implementation of “runas.exe” we can check out the source code of ReactOS⁴⁸⁵.

```
C:\>echo %username%
user

C:\>runas /u:test "cmd"
Enter the password for test:
Attempting to start cmd as user [REDACTED]\test" ...
RUNAS ERROR: Unable to run - cmd
1058: The service cannot be started, either because it is disabled or because it has no enabled devices associated with it.
```

⁴⁸⁴ [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771525\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771525(v=ws.11))

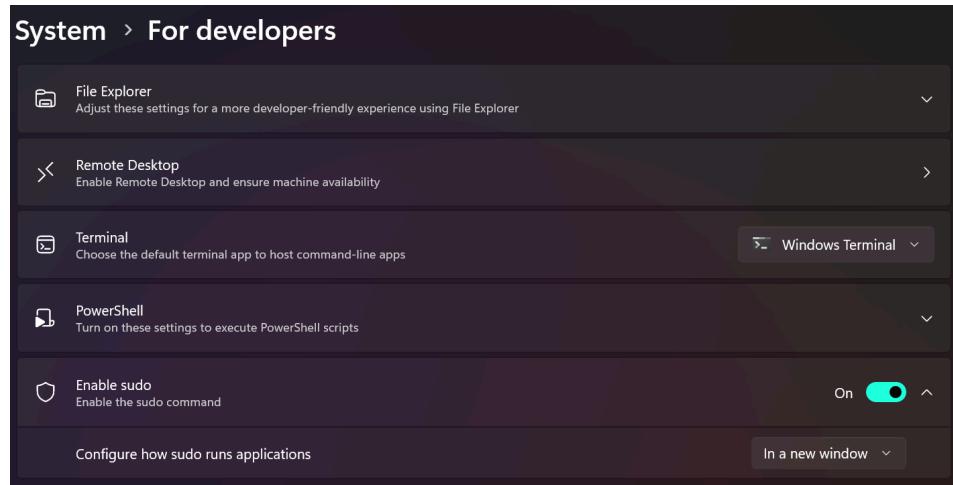
⁴⁸⁵ <https://github.com/reactos/reactos/tree/master/base/applications/runas>

Sudo for Windows

“Sudo for Windows” is a new security feature included as part of Windows (since Windows 11, version 24H2 and higher). It provides a new way for users to run elevated commands directly from an unelevated console session. As of today sudo supports three different configuration options: creating a new elevated console window (forceNewWindow), disallowing the read of the input by the elevated process (disableInput) and inline⁴⁸⁶.

Overall, sudo for Windows is basically a transformation of a security concept\feature (SuperUser Do) from the Linux operating system⁴⁸⁷. It is important to understand that if we execute sudo in input closed\inline there are security risks which can lead to malicious code running elevated commands - more on that in future writeups. For enabling it we can go to “Settings-> System-> For Developers” and toggle the “Enable Sudo” option⁴⁸⁸ - as shown in the screenshot below.

Lastly, as opposed to sudo under Linux and “runas.exe”⁴⁸⁹ which allow running programs as other users, as for today sudo for Windows does not support that - more differences between runas and sudo for Windows are going to be detailed in a future writeup. Sudo for Windows is not a fork of the Unix\Linux sudo project and is by itself open source⁴⁹⁰. There are also other sudo implementations for Windows which are not provided by Microsoft like “gsudo”⁴⁹¹.



⁴⁸⁶ <https://devblogs.microsoft.com/commandline/introducing-sudo-for-windows/>

⁴⁸⁷ <https://medium.com/@boutnaru/the-linux-security-journey-sudo-superuser-do-48d8f7c490d>

⁴⁸⁸ <https://learn.microsoft.com/en-us/windows/sudo/>

⁴⁸⁹ <https://medium.com/@boutnaru/the-windows-process-journey-runas-exe-run-as-utility-3c1e0b8aaa67>

⁴⁹⁰ <https://github.com/microsoft/sudo>

⁴⁹¹ <https://github.com/gerardog/sudo>

Sudo for Windows vs RunAs

While both RunAs⁴⁹² and “Sudo for Windows”⁴⁹³ provide the ability for executing command/script/application/etc with elevated privileges there are several differences between them. Thus, there are specific use-cases in which one is better than the other.

Overall, RunAs is able to execute a program as any other user both local users⁴⁹⁴ and domain users⁴⁹⁵. It is not limited to administrators only, by the way this feature is also part sudo’s (for Windows) roadmap. Also, while using runas.exe the user is prompted for providing a password using CLI while sudo is elevated using UAC. Using the “forceNewWindow” option with “sudo.exe” provides a similar behavior as “runas.exe”⁴⁹⁶.

Lastly, “sudo.exe” provides the ability to elevate a process while connecting it to the current console (with/without input), this feature is not supported by “runas.exe”⁴⁹⁷ - as shown below⁴⁹⁸. Moreover, although “Sudo for Windows” was introduced with Windows 11 it can also execute and work fine on Windows 10 and even Windows 7 but without the configuration options⁴⁹⁹.

```
PS C:\Users\weebo> sudo sudo config --enable disableInput ←
Sudo mode set to DisableInput mode
PS C:\Users\weebo> sudo cmd ←
Microsoft Windows [Version 10.0.26058.1100]
(c) Microsoft Corporation. All rights reserved.

C:\Users\weebo> ←
PS C:\Users\weebo>
```

⁴⁹² <https://medium.com/@boutnaru/the-windows-process-journey-runas-exe-run-as-utility-3c1e0b8aaa67>

⁴⁹³ <https://medium.com/@boutnaru/the-windows-security-journey-sudo-for-windows-82f0961eefdb>

⁴⁹⁴ <https://medium.com/@boutnaru/the-windows-security-journey-local-user-account-effe75db39d8>

⁴⁹⁵ <https://medium.com/@boutnaru/the-windows-security-journey-domain-user-account-d01486510e03>

⁴⁹⁶ <https://learn.microsoft.com/en-us/windows/sudo/>

⁴⁹⁷ <https://www.minitool.com/news/sudo-for-windows-vs-runas.html>

⁴⁹⁸ <https://cheapwindowsvps.com/blog/comparing-sudo-for-windows-vs-runas-and-sudo-for-linux/>

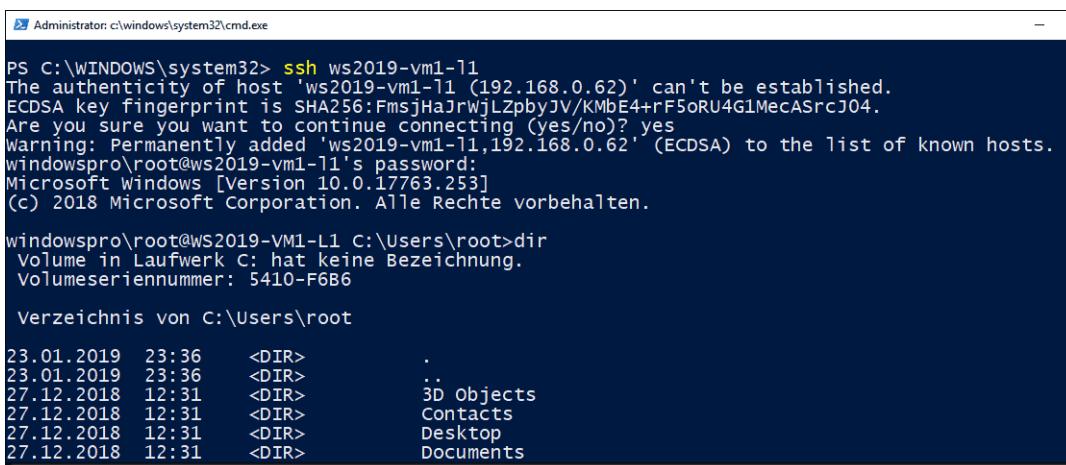
⁴⁹⁹ <https://winaero.com/sudo-for-windows-11-actually-runs-on-windows-10-and-windows-7/>

OpenSSH for Windows

OpenSSH provides the ability for users\administrators logging in to a remote Windows system by leveraging the SSH (Secure Shell) protocol - as shown in the screenshot below⁵⁰⁰. Since “Windows Server 2025” OpenSSH is installed by default⁵⁰¹. Thus, we can enable/disable the “ssh” service using the service control manager⁵⁰².

Overall, OpenSSH is an open-source implementation of SSH tools⁵⁰³ which are in use among different Linux distributions and other non-Windows systems. Beside authentication SSH protects communications security and integrity with strong encryption⁵⁰⁴. Starting with Windows 10 (build 1809) and “Windows Server 2019” we have OpenSSH available as a feature on demand⁵⁰⁵.

Lastly, it is important to know that OpenSSH for Windows includes different tools such as: “ssh” (SSH client), “sshd” (SSH server), “ssh-keygen” (used for generating/managing/converting authentication keys), “ssh-agent” (stores private keys), “sftp” (secure file transfer over SSH), “scp” (file copy over SSH) and “ssh-keyscan” (collecting public SSH host keys from hosts). With those tools we can not only use for username/password authentication. We can also leverage key-based authentication for coping with password brute-force attacks⁵⁰⁶.



The screenshot shows a Windows Command Prompt window titled "Administrator: c:\windows\system32\cmd.exe". The command entered is "ssh ws2019-vm1-11". The output shows the authenticity of the host cannot be established, and a warning message about adding the host to the list of known hosts. It then asks for the password, which is provided. Finally, a directory listing for "C:\Users\root" is shown, containing ".", "..", "3D Objects", "Contacts", "Desktop", and "Documents".

```
PS C:\WINDOWS\system32> ssh ws2019-vm1-11
The authenticity of host 'ws2019-vm1-11 (192.168.0.62)' can't be established.
ECDSA key fingerprint is SHA256:FmsjHaJrWjLZpbvJV/KMBe4+rF5oRU4G1MecASrcJ04.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ws2019-vm1-11,192.168.0.62' (ECDSA) to the list of known hosts.
windowspro\root@ws2019-vm1-11's password:
Microsoft Windows [Version 10.0.17763.253]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

windowspro\root@ws2019-VM1-L1 C:\Users\root>dir
 Volume in Laufwerk C: hat keine Bezeichnung.
 Volumeseriennummer: 5410-F6B6

 Verzeichnis von C:\Users\root

23.01.2019 23:36    <DIR>      .
23.01.2019 23:36    <DIR>      ..
27.12.2018 12:31    <DIR>      3D Objects
27.12.2018 12:31    <DIR>      Contacts
27.12.2018 12:31    <DIR>      Desktop
27.12.2018 12:31    <DIR>      Documents
```

⁵⁰⁰ <https://4sysops.com/archives/installing-openssh-on-windows-10-1803-and-higher-and-server-2019/>

⁵⁰¹ https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse

⁵⁰² <https://medium.com/@boutanru/the-windows-process-journey-services-exe-service-control-manager-45e9d2e0e867>

⁵⁰³ <https://github.com/openssh>

⁵⁰⁴ <https://www.ssh.com/academy/ssh/protocol>

⁵⁰⁵ <https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh-overview>

⁵⁰⁶ https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_keymanagement

Microsoft Intune

“Microsoft Intune” is a command center for endpoint management which can be leveraged to empower IT administrators and protect endpoints. Thus, it can help in strengthening security, boosting IT agility, accelerating business performance and reducing overall cost of management. It also provides the ability for cloud based helpdesk to user connections and allows standard users to perform IT approved privileged tasks⁵⁰⁷.

Overall, Microsoft Intune (formerly known as Windows Intune) is a cloud-based unified endpoint management tool for organizations. It provides the ability for: cross platform endpoint management (Windows/Linux/ChromeOS/macOS/Android/iOS/iPadOS), builtin endpoint security (conditional access, device compliance, privilege access and more)⁵⁰⁸ - as shown in the screenshot below.

Lastly, Intune can improve end-user experience using AI and analytics. This allows detecting and remediating endpoint issues like (but not limited to): preventing device and application anomalies (using AI to identify pattern and trends), getting deep visibility into device state and configuration and resolving issues quickly that reduces downtime. We can also use it for automating cloud certificate management aka “Microsoft Cloud PKI”⁵⁰⁹.

The screenshot shows the Microsoft Endpoint Manager admin center interface. The left sidebar includes navigation links such as Home, Dashboard, All services, Devices, Apps, Endpoint security, Reports, Users, Groups, Tenant administration, Troubleshooting + support, Device enrollment, Smart devices, Provisioning, and Windows 365. The main content area is titled "Devices | All devices" and displays a table of 23 devices. The columns include Device name, Managed by %, Ownership %, Compliance %, OS, OS version %, Last check-in %, and Privacy score UTM %. The table lists various devices with their respective details, such as Unknown, APPLICANT, APPLICANTERIA, and several desktops and mobile devices running Windows, macOS, or Chrome OS. The compliance status varies from Not Compliant to Compliant, and the privacy scores range from 0.00 to 100.00.

⁵⁰⁷ <https://www.microsoft.com/en-us/security/business/microsoft-intune>

⁵⁰⁸ <https://www.microsoft.com/en-us/security/business/endpoint-management/microsoft-intune>

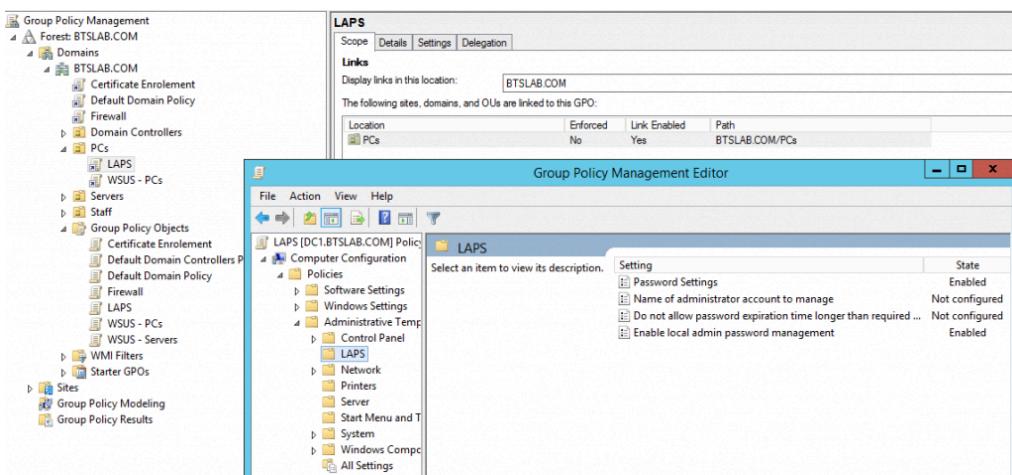
⁵⁰⁹ <https://www.microsoft.com/en-us/security/business/endpoint-management/microsoft-intune-advanced-analytics>

Microsoft LAPS (Microsoft Local Administrator Password Solution)

“Microsoft LAPS” (Microsoft Local Administrator Password Solution) is a Windows feature that is deprecated as of Windows 11 23H2 (and later). Thus, the installation of the MSI package (LAPS.x64.msi|LAPS.ARMS64.msi|LAPS.x86.msi) is blocked by a new version of the operating system. LAPS provides management of local account passwords for domain joined devices. Passwords are stored in AD (Active Directory) and protected by ACL, so only eligible users can read it or request its reset. It is replaced by “Windows LAPS”⁵¹⁰.

Overall, by using LAPS, passwords for the local administrator are unique for each managed device, the passwords are randomly generated and securely stored as part of the AD infrastructure. The feature is based on GPO CSE (Client Side Extension). The CSE checkers if the password expired or not, generates a new password if needed, changes the password of the administrator account and notifies AD regarding the password and expiration date⁵¹¹.

Lastly, LAPS helps in providing the ability to avoid reusable passwords of the local administrator account (which also helps in coping with PtH attacks). We can configure LAPS using different methods like: PowerShell and GPO. Among the settings are the name of the administrator account to manage (in case it was renamed), password settings and more - as shown in the screenshot below⁵¹².



⁵¹⁰ <https://www.microsoft.com/en-us/download/details.aspx?id=46899>

⁵¹¹ https://download.microsoft.com/download/C/7/A/C7AAD914-A8A6-4904-88A1-29E657445D03/LAPS_Datasheet.docx

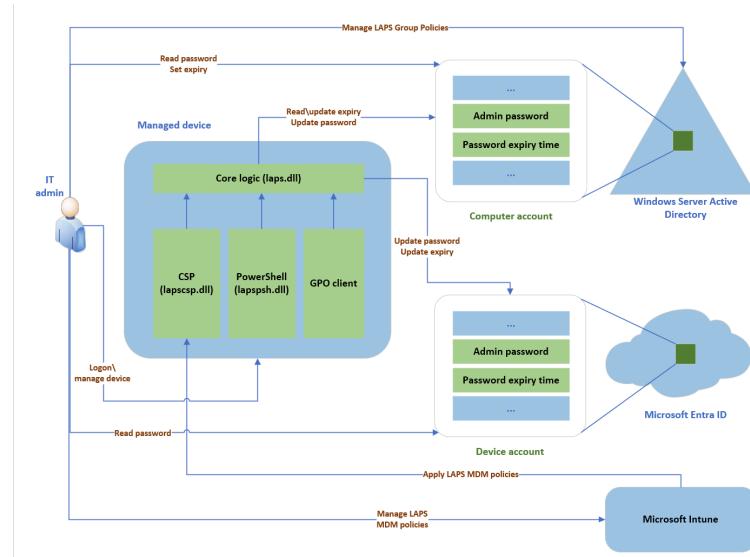
⁵¹² <https://www.adamcouch.co.uk/microsoft-laps/>

Windows LAPS (Windows Local Administrator Password Solution)

LAPS (Local Administrator Password Solution) is a Windows feature which is used to manage/backup passwords of local user accounts for “Active Directory”⁵¹³ or ”Microsoft Entra ID”⁵¹⁴ joined devices. It is relevant both for server and workstation editions of the Windows operating system.

Overall, if a device is joined to an active directory domain we can backup the password of the local administrator account only to a Windows server AD. Also, if the device is only joined to Microsoft Entra ID we can backup the passwords only there⁵¹⁵. The core logic of LAPS is implemented as part of “%windir%\System32\laps.dll” (LAPS Server DLL) while “%windir%\System32\lapscsp.dll” contains the configuration service provider (CSP) logic⁵¹⁶ - as shown in the diagram below.

Lastly, due to the fact LAPS is used for rotating the password of a local Windows account. We can configure the complexity of the password (such as if we want to use large letters+small letters+numbers and special characters)⁵¹⁷. By the way, “Windows LAPS” is the replacement of “Microsoft LAPS”⁵¹⁸.



⁵¹³ <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-af9e795f86b4>

⁵¹⁴ <https://medium.com/@boutnaru/the-windows-security-journey-microsoft-entra-id-617489ca3d27>

⁵¹⁵ <https://learn.microsoft.com/en-us/windows-server/identity/laps/laps-overview>

⁵¹⁶ <https://learn.microsoft.com/en-us/windows-server/identity/laps/laps-concepts-overview>

⁵¹⁷ <https://learn.microsoft.com/en-us/windows-server/identity/laps/laps-concepts-passwords-passphrases>

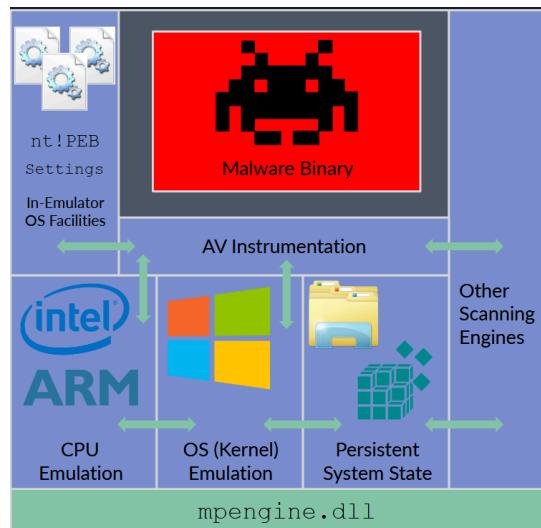
⁵¹⁸ <https://medium.com/@boutnaru/the-windows-security-journey-microsoft-laps-microsoft-local-administrator-password-solution-7e08c9d19fd4>

Windows Defender Antivirus Emulator

Due to the fact AV signatures are easily bypassed (like by using packers) antimalware solutions have added emulation capabilities. The goal of an emulation engine is to execute an unknown binary/script in an emulated environment and check at runtime for malicious behavior/scan for signatures in memory/files dropped to disk/etc⁵¹⁹.

Overall, an analysis conducted on “Windows Defender Antivirus Emulator” showed that some emulated stubs had hardcoded returns like: “GetComputerName” returning “HAL9TH” and “GetUserName” returning “JohnDoe”. Also, the emulator has specific files like “C:\INTERNAL__empty”. Moreover, there is also a concept of VDLLs which are modified versions of Windows’ DLLs which export extra functions. All of those artifacts (and more) can be used by binaries/scripts to identify they are being emulated⁵²⁰.

Lastly, because we can execute the binary/script forever the emulation can stop due one of the following reasons: termination of the process, limitation of time, max number of CPU instructions, number of API calls, amount of memory used and more. We can summarize that the emulation environment should provide user-mode and kernel-mode emulation, CPU emulation, file-systems and registry emulation and more - as shown in the diagram below⁵²¹



⁵¹⁹ <https://www.kaspersky.com/enterprise-security/wiki-section/products/emulator>

⁵²⁰ <https://harfanglab.io/insidethelab/raspberry-robin-and-its-new-anti-emulation-trick/>

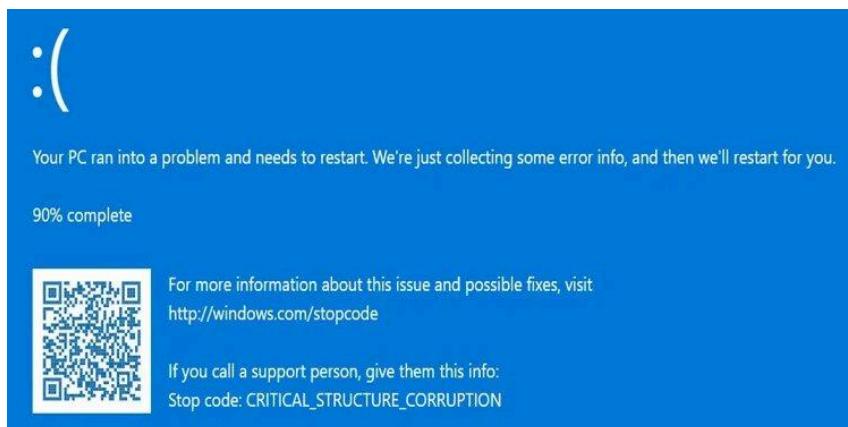
⁵²¹ <https://i.blackhat.com/us-18/Thu-August-9/us-18-Bulazel-Windows-Offender-Reverse-Engineering-Windows-Defenders-Antivirus-Emulator.pdf>

PatchGuard (Kernel Patch Protection)

“PatchGuard” (aka “Kernel Patch Protection” or KPP) is a security feature as part of 64-bit versions of the Windows operating system. The goal of “PatchGuard” is preserving the integrity of the operating system (mostly hooking, patching and direct object kernel manipulation attacks). Kernel code and data structures are verified to block inconsistent modifications or code injections into the kernel by developers and/or attackers⁵²².

Overall, “PatchGuard” periodically checks different kernel code areas/data structures. Among those data structures are: SSDT, IDT/GDT, HAL, NDIS, debug routines, loaded module list and more. “PatchGuard” can compute a checksum of sensitive memory areas regularly during the execution time of the system and compare it with the one obtained at boot time (before any user driver load). In case of a mismatch a BSOD (Blue Screen Of Death) is triggered with a BugCheck code 0x109 which is also known as “CRITICAL_STRUCTURE_CORRUPTION”⁵²³ - as shown in the screenshot below⁵²⁴.

Lastly, “PatchGuard”\”KPP” was introduced in 2005 with the launch of 64-bit versions of “Windows XP” and “Windows 2003 Server”. Also, there is no public publication by Microsoft about the specific internal in which “PatchGuard” performs its checks but it is believed to be randomized and probably varies between seconds to minutes⁵²⁵. For more information about the causes of “BugCheck code 0x109” I suggest going over the Microsoft documentation⁵²⁶. It is important to know that along the years different techniques for bypassing “PatchGuard” were introduced like: “GhostHook”, “InfinityHook” and “ByePg”⁵²⁷.



⁵²² <https://web.archive.org/web/20230427020624/https://zerocondition.com/posts/demystifying-patchguard/>

⁵²³ https://blog.tetraene.com/downloads/Tetraene_PatchGuard_Analysis_RS4_v1.01.pdf

⁵²⁴ <https://tech4gamers.com/critical-structure-corruption-windows-11/>

⁵²⁵ <https://blog.nviso.eu/2021/10/21/kernel-karnage-part-1/>

⁵²⁶ <https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/bug-check-0x109---critical-structure-corruption>

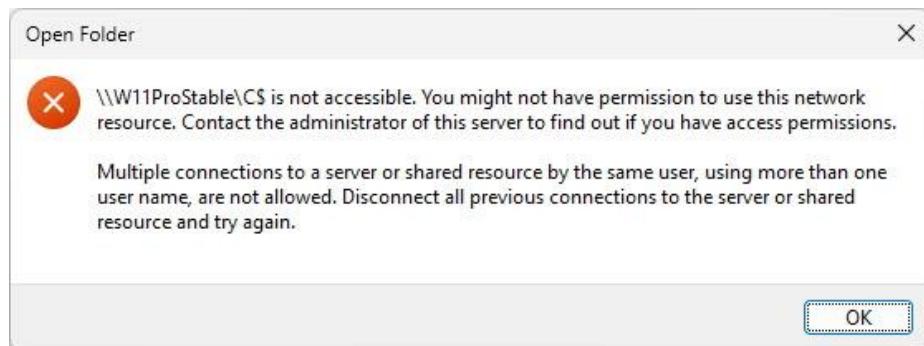
⁵²⁷ <https://therecord.media/poc-published-for-new-microsoft-patchguard-kpp-bypass>

LocalAccountTokenFilterPolicy (User Account Control Remote Restrictions)

UAC (User Account Control) is a security feature of the Windows operating system (since Windows Vista) that, if enabled, gives a filtered token to the user (in case the user belongs to the local administrators group). If the user wants the privileged token he needs to consent to in order to get the unfiltered token with the administrator rights⁵²⁸. To better protect administrator users there are UAC restrictions for network connections. The goal of those is to prevent “loopback attacks” and malicious software for executing code remotely using the administrative rights

Overall, the “LocalAccountTokenFilterPolicy” is a registry⁵²⁹ value (DWORD) stored as part of “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System”. In case it has a “0” value (which is the default one) the administrator rights are removed for network connections. In case of “1” the eveneted token is used and not the filtered one⁵³⁰. For example, let's say we try to access a remote machine (“\\[RemoteDevice]\\[ShareName]”) using “net use” with a local account (SAM defined user account) which has administrator rights we will get an “Access is Denied” error - as shown in the screenshot below⁵³¹.

Lastly, it is important to know that in case of a domain user the restriction is not relevant. In case a domain user (which is a member of the local administrators group) connects remotely it will get the full administrator access token of the remote device without any UAC restrictions. If we want to administer a remote system with a local user we need to logon interactively like by using “Remote Desktop” (RDP) and/or “Remote Assistant”⁵³².



⁵²⁸ <https://medium.com/@boutnaru/the-windows-security-journey-uac-user-account-control-ce395df5c784>

⁵²⁹ <https://medium.com/@boutnaru/the-windows-concept-journey-registry-0767e79387a9>

⁵³⁰ <https://kb.cybertecsecurity.com/knowledge/localaccounttokenfilterpolicy>

⁵³¹ <https://www.itechtics.com/localaccounttokenfilterpolicy>

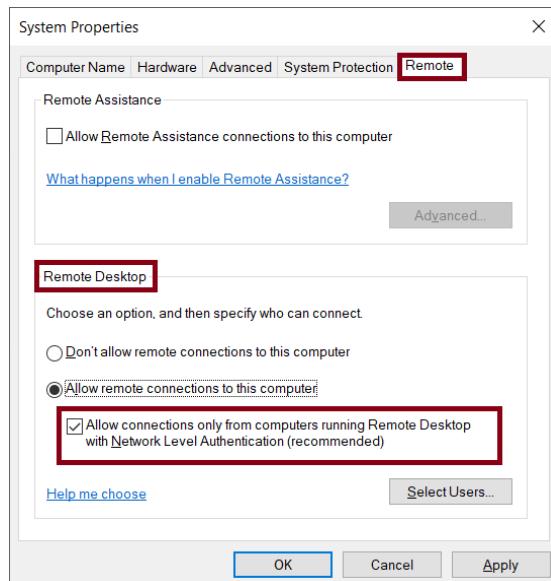
⁵³² <https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/user-account-control-and-remote-restriction>

NLA (Network Level Authentication)

NLA (Network Level Authentication) is a security feature as part of RDP (Remote Desktop Protocol). It is used for providing an additional layer of authentication before an RDP session is established. This feature is supported from Windows Vista/Windows Server 2008 as part of RDP 6.0⁵³³. We can enable NLA as part of the “Remote Desktop” settings in System Properties (sysdm.cpl) under the “Remote” tab - as shown in the screenshot below.

Overall, before the usage of NLA when a user opened an RDP session the login screen from the remote server would load. This caused the server to allocate resources before the user performed any authentication. NLA leverages CredSSP through SSPI⁵³⁴. Using NLA helps in mitigating RDP vulnerabilities that are exploited prior to authentication and reduces the resources needed by RDP session before authentication⁵³⁵.

Lastly, NLA is also supported with RDS (Remote Desktop Services) - more on that in future writeups. It is important to understand that NLA adds an extra layer of protection by validating the user's credentials\identity before creating the remote desktop connection⁵³⁶.



⁵³³ <https://www.portnox.com/cybersecurity-101/network-level-authentication-nla/>

⁵³⁴ <https://medium.com/@boutnaru/the-windows-security-journey-sspi-security-support-provider-interface-9d0b455fc593>

⁵³⁵ <https://superops.com/rmm/what-is-network-level-authentication>

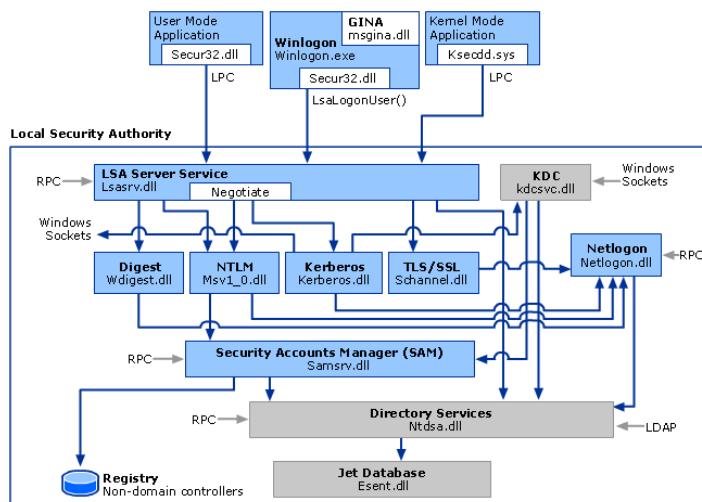
⁵³⁶ <https://www.avica.com/blog/network-level-authentication-nla/>

GINA (Graphical Identification and Authentication)

GINA (Graphical Identification and Authentication) is a component that is a replaceable DLL component loaded by “winlogon.exe”⁵³⁷ - as shown in the diagram below⁵³⁸. It is used for implementing the authentication policy of interactive logon flow and more⁵³⁹.

Overall, GINA was discontinued in Windows Vista and was replaced by credential providers. Thus, GINA is relevant for “Windows NT 3.51”, “Windows NT 4.0”, “Windows 2000”, “Windows XP” and “Windows 2003 Server”. By default Microsoft provided “msgina.dll” as part of the OS⁵⁴⁰. Developers could also create their own GINA implementation for alternate authentication methods and policy controls (such as custom password complexity rules). An example of such an open source project is “pGINA”⁵⁴¹.

Lastly, for a reference implementation of Microsoft’s “msgina.dll” we can check out the source code of ReactOS⁵⁴². By the way, GINA gets the credentials and uses LSALogonUser⁵⁴³ for performing an authentication flow for a security principal⁵⁴⁴.



⁵³⁷ <https://medium.com/@boutharu/the-windows-process-journey-winlogon-exe-windows-logon-application-88a1d4d3e13c>

⁵³⁸ <https://spy-sofi.net/how-windows-security-works/>

⁵³⁹ <https://learn.microsoft.com/en-us/windows/win32/secgloss/g-gly>

⁵⁴⁰ https://en.wikipedia.org/wiki/Graphical_identification_and_authentication

⁵⁴¹ <http://pgina.org/>

⁵⁴² <https://github.com/reactos/reactos/tree/master/dll/win32/msgina>

⁵⁴³ <https://github.com/reactos/reactos/blob/master/dll/win32/msgina/lsa.c#L200>

⁵⁴⁴ <https://learn.microsoft.com/en-us/windows/win32/api/ntsecapi/inf-ntsecapi-lsalogonuser>

Credential Providers

Credential providers are the primary way for authentication as part of the Windows operating system. A credential provider is implemented as a DLL⁵⁴⁵, providing different COM interfaces that are invoked by “LogonUI.exe”. It is the process which is shown in the “secure desktop” as part of the logging flow of users⁵⁴⁶. Microsoft provides as part of the OS a couple of credentials such as: password, PIN, smart card and “Windows Hello” (Fingerprint, Face, and Iris recognition) - as shown in the screenshot below (taken using Sysinternals’ “Process Monitor”). Those are sometimes called “system credential providers” vs “custom credential providers” that can be created by users/developers⁵⁴⁷.

Overall, credential providers can also be loaded into “Windows Explorer”⁵⁴⁸, and\or “conhost.exe”⁵⁴⁹ and\or “CredentialUIBroker.exe”. It can happen in different conditions due to UAC prompts or cases when the user selects the “Run as different User” option⁵⁵⁰. Also, we can use the “CredUIPromptForWindowsCredentials” API call (implemented as part of %windir%\system32\credui.dll) for loading a credential provider to our one process⁵⁵¹.

Lastly, we can also check out samples from Microsoft for implementing a credential provider⁵⁵². By the way, there is also a credential provider of Google for Windows (GCPW). It allows connecting to Windows based devices using a google account⁵⁵³. It is important to know that until Windows Vista (which added the support of credentials providers) Windows supported GINA aka “Graphical Identification and Authentication”⁵⁵⁴.

11:29:3... LogonUI.exe	15932	Load Image	C:\Windows\System32\Windows.Devices.Enumeration.dll
11:29:3... LogonUI.exe	15932	Load Image	C:\Windows\System32\SmartcardCredentialProvider.dll
11:29:3... LogonUI.exe	15932	Load Image	C:\Windows\System32\StructuredQuery.dll
11:29:3... LogonUI.exe	15932	Load Image	C:\Windows\System32\FontGlyphAnimator.dll
11:29:3... LogonUI.exe	15932	Load Image	C:\Windows\System32\BioCredProv.dll

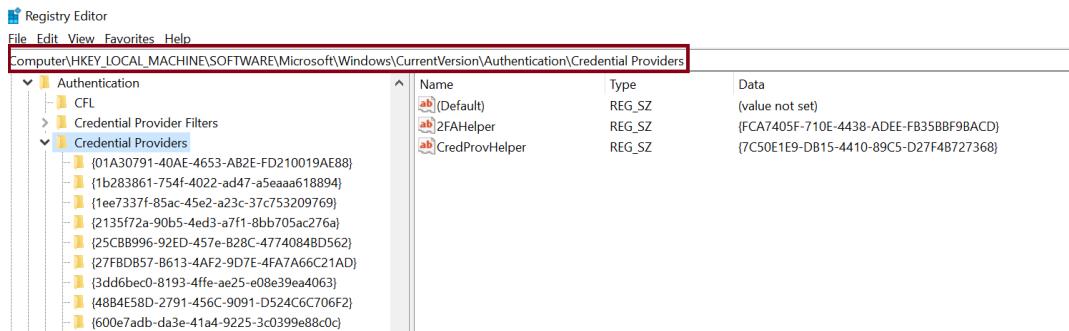
⁵⁴⁵ <https://medium.com/@boutnaru/the-windows-concept-journey-dll-dynamic-link-library-dcecd545416d>
⁵⁴⁶ <https://medium.com/@boutnaru/the-windows-process-journey-logonui-exe-windows-logon-user-interface-host-4b5b8b6417cb>
⁵⁴⁷ <https://learn.microsoft.com/en-us/windows/win32/secauthn/credential-providers-in-windows>
⁵⁴⁸ <https://medium.com/@boutnaru/the-windows-concept-journey-file-explorer-previously-windows-explorer-e48077b135a0>
⁵⁴⁹ <https://medium.com/@boutnaru/the-windows-process-journey-conhost-exe-console-window-host-f03fdb35574>
⁵⁵⁰ <https://dennissbabkin.com/blog/?t=primer-on-writing-credential-provider-in-windows>
⁵⁵¹ <https://learn.microsoft.com/en-us/windows/win32/api/wincred/nf-wincred-creduipromptforwindowscredentialsw>
⁵⁵² <https://learn.microsoft.com/en-us/samples/microsoft/windows-classic-samples/credential-provider/>
⁵⁵³ <https://tools.google.com/dlpage/gcpw/>
⁵⁵⁴ <https://medium.com/@boutnaru/the-windows-security-journey-gina-graphical-identification-and-authentication-9450b639559b>

Credential Providers Settings

We can configure credential providers⁵⁵⁵ using the Windows registry⁵⁵⁶. The “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers” key holds the list of available credential providers (each described using CLSID\GUID) as different sub keys - as shown in the screenshot below.

Overall, we can take as an example three of the most common credential providers: “Password Provider” (which is used for username and password based authentication), “NGC Credential Provider” (used for the PIN of Windows Hello for Business) and “FIDO Credential Provider” (used for FIDO2 security keys). The first has a CLSID of “{60b78e88-ead8-445c-9cf0-0b87f74ea6cd}” which based on the settings in “HKEY_CLASSES_ROOT” leads to “%windir%\system32\credprovs.dll”. The second, has a CLSID of “{D6886603-9D2F-4EB2-B667-1971041FA96B}” which is configured by default to load “%windir%\System32\ngccredprov.dll”. The third, with a CLSID of “{F8A1793B-7873-4046-B2A7-1F318747F427}” that is configured by default to load “%windir%\system32\fidocredprov.dll”. Also, we can set credential providers using “Microsoft Intune”⁵⁵⁷.

Lastly, the credential providers configuration is relevant both for builtin\system providers and for custom providers⁵⁵⁸. We can enable and\or disable credential providers using the different settings⁵⁵⁹.



⁵⁵⁵ <https://medium.com/@boulnaru/the-windows-security-journey-credential-providers-fb6b809ce2f5>

⁵⁵⁶ <https://medium.com/@boulnaru/the-windows-concept-journey-registry-0767e79387a9>

⁵⁵⁷ <https://petervanderwoude.nl/post/configuring-the-default-credential-provider/>

⁵⁵⁸ https://kb.swivelsecure.com/w/index.php/Windows_Credential_Provider

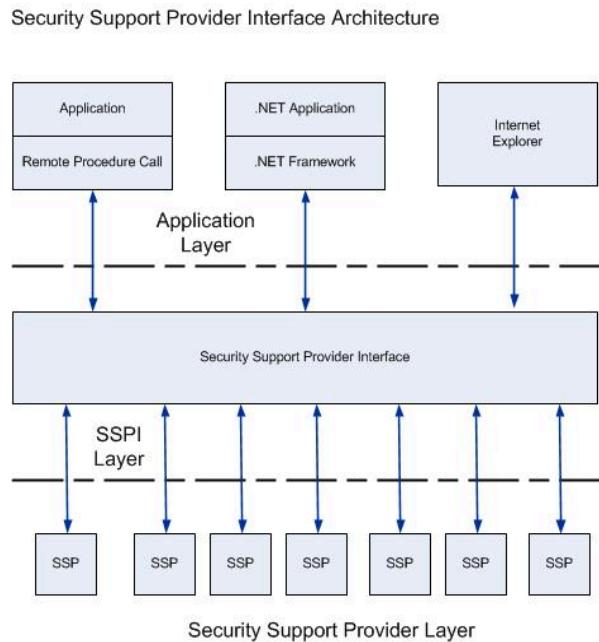
⁵⁵⁹ <https://answers.microsoft.com/en-us/windows/forum/all/how-do-you-disable-windows-11-credential-provider/b500f260-23c6-4360-8875-31db512daed4>

SSPI (Security Support Provider Interface)

SSPI (Security Support Provider Interface) is an authentication framework for Windows, we can think about it as similar PAM⁵⁶⁰ on Linux. SSPI is an API that can be used to perform different tasks like: message integrity, message privacy, authentication and security quality of service for distributed protocols. SSPI is a Windows implementation of GSS-API⁵⁶¹.

Overall, SSPI is a software interface that can be used by distributed programming libraries (like RPC) for authenticated communication. Each SSPI module is called SSP (Security Support Provider) that is implemented as a DLL (Dynamic Link Library) which can provide one/more security packages. Examples of security packages are: NTLM and Kerberos⁵⁶². For a reference implementation of loading SSP DLLs we can check out the source code of ReactOS⁵⁶³.

Lastly, the goal of SSPI is to serve as a standard interface to different SSPs for applications⁵⁶⁴ - as shown in the diagram below⁵⁶⁵. SSPI works together with LSA (Local Security Authority) which is the protected system used for authentication and signing in users to computers/server⁵⁶⁶.



⁵⁶⁰ <https://medium.com/@boutnaru/the-linux-security-journey-pam-pluggable-authentication-module-388496a8785c>

⁵⁶¹ <https://docs.oracle.com/cd/E19683-01/816-1331/overview-6/index.html>

⁵⁶² <https://learn.microsoft.com/en-us/windows/win32/rpc/sspi-architectural-overview>

⁵⁶³ <https://github.com/reactos/reactos/blob/master/dll/win32/secur32/sspi.c>

⁵⁶⁴ <https://www.sentinelone.com/blog/how-attackers-exploit-security-support-provider-ssp-for-credential-dumping/>

⁵⁶⁵ <https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/security-support-provider-interface-architecture>

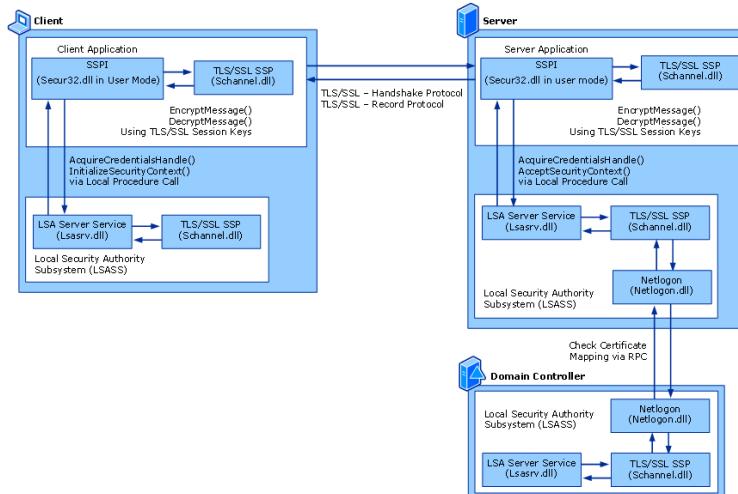
⁵⁶⁶ <https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-architecture>

SSP (Security Support Provider)

SSP (Security Support Provider) is a DLL (Dynamic Link Library) that performs security-related operations like authentication and makes one\more security packages available⁵⁶⁷. We can think about it as similar (but not identical as explained later) to a PAM module⁵⁶⁸.

Overall, SSP is implemented using SSPI⁵⁶⁹. It is important to understand that SSPs are not authentication mechanisms, thus they are used for gathering\serialize credentials. LSA and Windows authentication packages are responsible for performing authentication. Examples of SSPs are: “Negotiate SSP”, “Kerberos SSP”, “NTLM SSP”, “Schannel SSP”, “Digest SSP” and “CredSSP”⁵⁷⁰

Lastly, the security packages implemented by SSPs provide mappings between an application's SSPI function calls and an actual security model's functions⁵⁷¹. The relationship between the different components (including SSP) is demonstrated regarding the “Schannel SSP” in the diagram below⁵⁷².



⁵⁶⁷ <https://www.sentinelone.com/blog/how-attackers-exploit-security-support-provider-ssp-for-credential-dumping/>

⁵⁶⁸ <https://medium.com/@boutnaru/the-linux-security-journey-pa>

⁵⁶⁹ <https://medium.com/@boutnaru/the-windows-security-journey-sspi-security-support-provider-interface-9d0b455fc593>

⁵⁷⁰ <https://ldapwiki.com/wiki/Wiki.jsp?page=Security%20Support%20Provider>

⁵⁷¹ <https://learn.microsoft.com/en-us/windows/win32/seccgloss/s-gly>

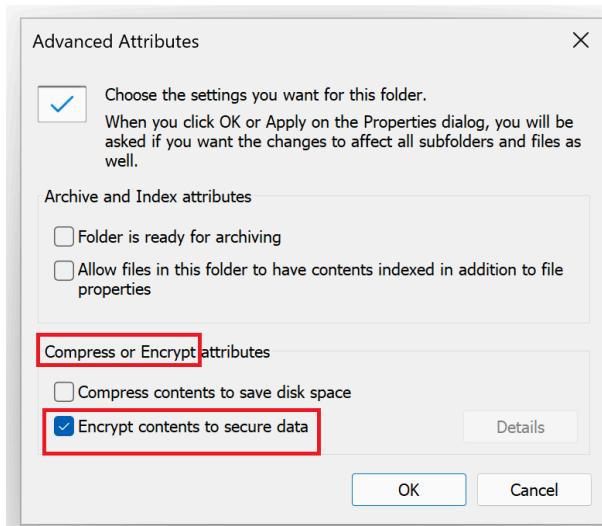
⁵⁷² [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/dn786429\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/dn786429(v=ws.11))

EFS (Encrypted File System)

EFS (Encrypted File System) is a security feature of NTFS⁵⁷³. EFS can be used for provisioning cryptographic protection of individual files\directories (stored on NTFS volumes). By the way, EFS can't encrypt: system files, compressed files, system directories, root directories and transactions⁵⁷⁴.

Overall, in order to check if our volume supports EFS we can call the "GetVolumeInformationA"\\"GetVolumeInformationW API function⁵⁷⁵ and check the "FS_FILE_ENCRYPTION" bit flag of "lpFileSystemFlags". By the way, EFS also appears in some places as the acronym for "Encrypting File System"⁵⁷⁶.

Also, in case we are copying an encrypted file ("CopyFile"\\"CopyFileEx") the EFS (Encrypting File System) service is leveraged for creating the target file and applying relevant keys⁵⁷⁷. The EFS service is hosted by "lsass.exe"⁵⁷⁸. For enabling EFS on a specific file or folder we can do the following: Right Click (on the file/folder) -> Properties -> Advanced -> Encrypt contents to secure data (Compress or Encrypt attributes) - as shown in the screenshot below.



⁵⁷³ <https://medium.com/@boutnaru/the-windows-concept-journey-ntfs-new-technology-file-system-433e27a2256a>

⁵⁷⁴ <https://learn.microsoft.com/en-us/windows/win32/fileio/file-encryption>

⁵⁷⁵ <https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getvolumeinformationw>

⁵⁷⁶ <https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getvolumeinformationw>

⁵⁷⁷ <https://appuals.com/cng-key-isolation-lsass-exe/>

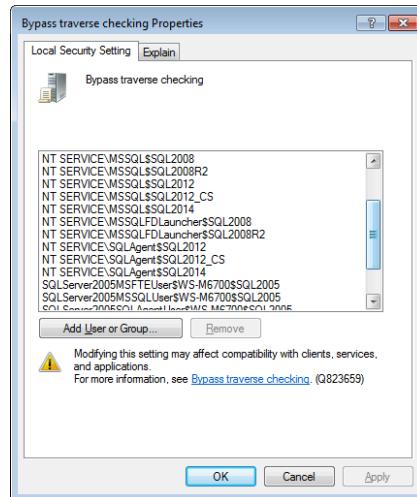
⁵⁷⁸ <https://medium.com/@boutnaru/the-windows-process-journey-lsass-exe-local-security-authority-process-24166cb0358f>

Virtual User Accounts (Virtual User)

Virtual user accounts have been introduced as part of Windows 7. They are similar to built-in accounts such as: “NT AUTHORITY\Network Service”⁵⁷⁹ and “NT AUTHORITY\Local Service”⁵⁸⁰ with extra features (as going to be described). We can customize virtual accounts to have the minimum privileges (user rights\access permissions\local group membership) needed to perform their specific tasks⁵⁸¹.

Overall, virtual accounts are local user accounts⁵⁸², which are used in order to simplify service administrations. This is done due to the following reasons: the accounts are automatically managed, they can access the network in a domain environment and no password management is required⁵⁸³. By the way, it is important to understand that DC (Domain Controllers) can't leverage virtual accounts due to the fact they don't allow the usage of local groups⁵⁸⁴.

Lastly, virtual accounts also use the naming convention of <Account Class>\<Account Name> (like “NT AUTHORITY\Network” Service”) - as shown in the screenshot below⁵⁸⁵. Examples of account classes are (but not limited to): “NT Service\<ServiceName>” (used by Windows services), “IIS APPPOOL\<NameOfPool>” (used for IIS application pool identity), “NT VIRTUAL MACHINE\<GUID>” (created for each Hyper-V virtual machine), “WINDOW MANAGER\DWM-<SessionID>”⁵⁸⁶, “Font Driver Host\UMFD-<SessionID>”⁵⁸⁷. Also, when virtual accounts access the network they leverage the context of the domain computer account: “<Domain>\<ComputerName>\$”⁵⁸⁸.



⁵⁷⁹ <https://medium.com/@boutnaru/the-windows-security-journey-network-service-nt-authority-network-service-e8706688e383>

⁵⁸⁰ <https://medium.com/@boutnaru/the-windows-security-journey-local-service-nt-authority-local-service-b1a624472931>

⁵⁸¹ <https://www.ibm.com/docs/en/qradar-common?topic=10-wincollect-virtual-accounts>

⁵⁸² <https://medium.com/@boutnaru/the-windows-security-journey-local-user-account-cffe75db39d8>

⁵⁸³ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-service-accounts#virtual-accounts>

⁵⁸⁴ <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-af9e795f86b4>

⁵⁸⁵ <https://www.sqlservercentral.com/blogs/windows-virtual-accounts-and-sql-server>

⁵⁸⁶ <https://medium.com/@boutnaru/the-windows-process-journey-dwm-exe-desktop-window-manager-e6802129c8d>

⁵⁸⁷ <https://medium.com/@boutnaru/the-windows-security-journey-umfd-user-mode-font-driver-03ea907a14e9>

⁵⁸⁸ <https://docs.delinea.com/online-help/privilege-manager/install/upgrades/virtual-accounts.htm>

DPAPI (Data Protection API)

The goal of DPAPI (Data Protection API) is to provide the ability for encrypting data using information related to the current user/computer account. By using DPAPI we avoid the need for explicitly generating and storing cryptographic keys⁵⁸⁹. We can use the “ProtectedData” in .NET for encrypting/decrypting data using DAPI⁵⁹⁰ - as shown in the screenshot below (taken from Visual Studio).

Overall, it is used for symmetric encryption of asymmetric private keys. This is done by leveraging user/system secrets as a source of entropy. The keys are derived from the user’s logon secrets or the system’s domain logon secrets⁵⁹¹. We select each of those based on whom we want to be able to decrypt the data (every user on the computer or only a specific user).

Lastly, among the data which is protected by DPAPI we can find the following (examples only): “Google Chrome”/“Internet Explorer” password and auto-compilation, passwords of remote desktop connections, email/FTP accounts used by Outlook, passwords of shared folders, wireless networks, .NET passport and personal data in applications such as Skype\MSN messenger⁵⁹².

A screenshot of Visual Studio showing a C# program named 'tester'. The code uses the System, System.Text, and System.Security.Cryptography namespaces to demonstrate DPAPI. It defines a class 'Program' with a static Main method. The method reads the string 'Tr0LeR' from the command line, converts it to bytes, protects it using DPAPI, and then unprotects it to get back the original string. The output window shows the encrypted data as a long Base64 string and the decrypted data as 'Tr0LeR'.

```
1  using System;
2  using System.Text;
3  using System.Security.Cryptography;
4  namespace tester
5  {
6      internal class Program
7      {
8          static void Main(string[] args)
9          {
10             byte[] cleartextBytes = Encoding.UTF8.GetBytes("Tr0LeR");
11             //never use null as entropy - THIS IS JUST AN EXAMPLE
12             byte[] chiperBytes = ProtectedData.Protect(cleartextBytes, null, DataProtectionScope.CurrentUser);
13             Console.WriteLine("Encrypted data (Base64): " + Convert.ToBase64String(chiperBytes) + "\n");
14             byte[] decryptedBytes = ProtectedData.Unprotect(chiperBytes, null, DataProtectionScope.CurrentUser);
15             Console.WriteLine("Decrypted data (Base64): " + Convert.ToBase64String(decryptedBytes) + "\n");
16             byte[] decodedBytes=Convert.FromBase64String(Convert.ToBase64String(decryptedBytes));
17             Console.WriteLine("Decrypted data is: " + System.Text.Encoding.UTF8.GetString(decodedBytes));
18         }
19     }
20 }
```

C:\Windows\system32\cmd.exe
Encrypted data (Base64): AQAAANCmnd8BFdERjHoAwE/C1+sBAAAukCUG+51kABwF0Vak8wAAAAACAAAAAAAQZgAAAAEAACAAAACMai^znEEA+2X+yhKzN8Gb77rv510bhZviwmlSUCEPiigAAAAAgAAAAIAACAAAAAR0fyAcJ0y8TzQ5fdxLzqAnf025xqvQqRunYxbIkOUSRAAACFZnkMMbcpnjOkGMotyja0QAAAJJD1Z5PQcwsUJM43icQ+BBJmNmMt0tWgrZHX/FHnjU60KD8wU9vJaQq1rGm/gqmgwdmCHOfzy0c1lc1lw+m+E=
Decrypted data (Base64): VHIwTGVS
Decrypted data is: Tr0LeR
Press any key to continue . . .

⁵⁸⁹ <https://learn.microsoft.com/en-us/dotnet/standard/security/how-to-use-data-protection>

⁵⁹⁰ <https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography.protecteddata>

⁵⁹¹ <https://www.thehacker.recipes/ad/movement/credentials/dumping/dpapi-protected-secrets>

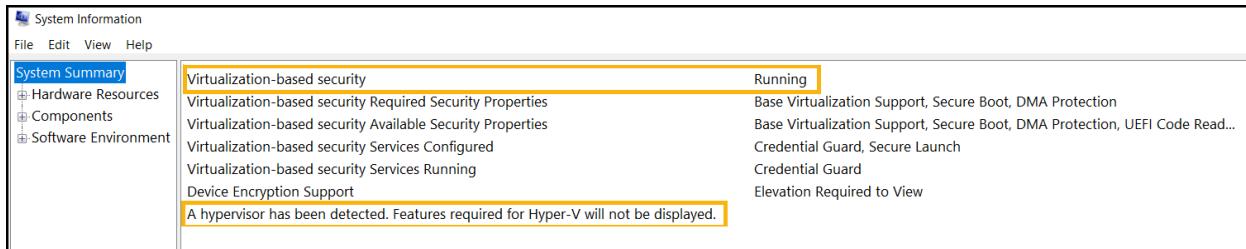
⁵⁹² <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/dpapi-extracting-passwords>

VBS (Virtualization Based Security)

VBS (Virtualization Based Security) is a security feature as part of the operating system that is based on hardware virtualization and the Windows hypervisor. It provides an isolated virtual environment which acts as a root of trust for the operating system. VBS assumes that the kernel of Windows can be compromised and it should provide security guardrails even in those scenarios⁵⁹³.

Overall, VBS provides the base layer for different security solutions provided by new version of Windows such as (but not limited to): HVCI (Hypervisor Protected Code Integrity), CG (Credentials Guard), IUM (Isolated User Mode) and more⁵⁹⁴. VBS is based on the Windows hypervisor for creating an isolated environment for providing the different security capabilities. We can check if VBS is enabled and running using the “System Information” application - as shown below⁵⁹⁵.

Lastly, in order for VBS to work there are hardware requirements which are needed: 64-bit CPU, SLAT (Second Level Address Translation), IOMMU/SMMU, TPM (Trusted Platform Module) 2.0, secure boot, UEFI memory reporting and more⁵⁹⁶.



⁵⁹³ <https://www.jaspreet.net/2024/10/12/2734/virtualization-based-security-pros-cons-and-use-cases-for-windows-10-11-and-server-environments/>

⁵⁹⁴ <https://learn.microsoft.com/en-us/answers/questions/245071/disable-virtualization-based-security-without-disa>

⁵⁹⁵ <https://www.intel.com/content/www/us/en/support/articles/000094905/processors.html>

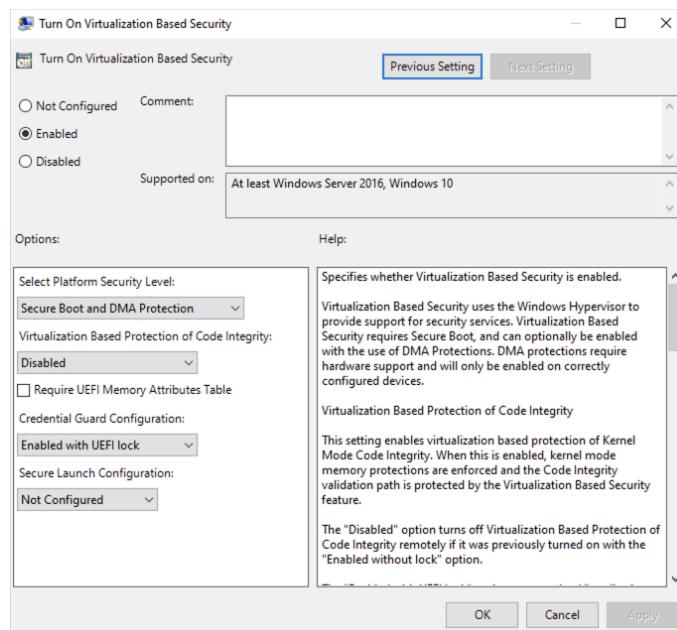
⁵⁹⁶ <https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs>

CG (Credentials Guard)

CG (Credentials Guard) is a security feature of the Windows operating system used for protecting against credentials theft (that can lead to pass-the-hash and pass-the-ticket attacks). Hence, NTLM password hashes\Kerberos TGTs (Ticket Granting Tickets)\other secrets are isolated using VBS (Virtual Based Security) so that only privileged system software can access them⁵⁹⁷. For enabling it we need a 64-bit CPU, secure boot enabled, VBS enabled and TPM (Trusted Platform Module) is also preferred⁵⁹⁸.

Overall, CG (aka Windows Defender Credential Guard) was introduced with Windows 10\Windows Server 2016 as a virtualization-based isolation technology for “lsass.exe”⁵⁹⁹. However, there are still techniques to bypass CG like (but not limited to): running a key-logger for capturing typed passwords, installing a malicious SSP⁶⁰⁰ and more⁶⁰¹.

Lastly, we can enable credentials guard in different ways such as (but not limited to): “Microsoft Intune”, “Registry” and “Group Policy” - as shown below⁶⁰². We can verify that credentials guard is enabled using the event viewer, PowerShell, system information and more⁶⁰³. By the way, in case “LsaIso.exe” is running it is also an indication that CG is enabled⁶⁰⁴.



⁵⁹⁷ <https://learn.microsoft.com/en-us/windows/security/identity-protection/credential-guard/>

⁵⁹⁸ <https://blog.netwrix.com/2023/02/06/using-windows-defender-credential-guard-to-protect-privileged-credentials/>

⁵⁹⁹ <https://medium.com/@boutnaru/the-windows-process-journey-lsass-exe-local-security-authority-process-24166cb0358f>

⁶⁰⁰ <https://medium.com/@boutnaru/the-windows-security-journey-ssp-security-support-provider-537668471d21>

⁶⁰¹ https://en.wikipedia.org/wiki/Credential_Guard

⁶⁰² <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/credential-guard.html>

⁶⁰³ <https://learn.microsoft.com/en-us/windows/security/identity-protection/credential-guard/configure>

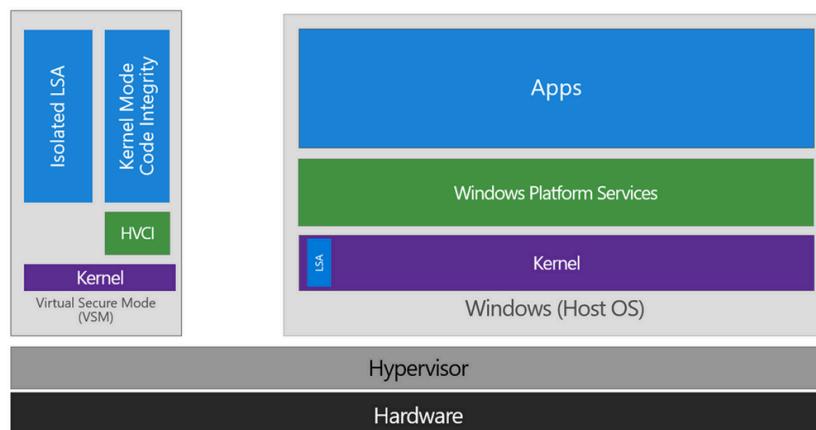
⁶⁰⁴ <https://medium.com/@boutnaru/the-windows-process-journey-lsaiso-exe-credential-guard-key-guard-99d8d558e3b8>

HVCI (Hypervisor Protected Code Integrity)

HVCI (Hypervisor Protected Code Integrity) is a feature based on VBS⁶⁰⁵ which is supported as part of Windows 10\Windows 11\Windows Server 2016 and later. HVCI is also called\referred to as “Memory Integrity”. It is a crucial component in protecting\hardening Windows by running kernel mode code integrity as part of VBS. This is done by ensuring a kernel page can be marked as executable only after passing specific code integrity checks (inside a secure environment) and that they are never writeable⁶⁰⁶.

Overall, the feature is also called HECl (Hypervisor Enforced Code Integrity). By the way, disabling “Memory Integrity” is recommended by Microsoft for boosting gaming performance⁶⁰⁷. Among the memory integrity features we can find different capabilities like the following examples (but not limited to). First, protecting from the modification of CGF (Control Flow Graph) bitmap for kernel mode drivers. Second, protecting the kernel mode code integrity process which ensures other trusted kernel processes have a valid certificate⁶⁰⁸.

Lastly, we can summarize that HVCI leverages hardware technology and virtualization to isolate CI (Code Integrity) decision making from the rest of the Windows operating system⁶⁰⁹ - as shown in the screenshot below⁶¹⁰. The memory integrity feature is part of “Core Isolation” feature, hence we can enable\disable it from “Settings->Privacy & Security->Windows Security->Device Security->Core Isolation->Memory Integrity”⁶¹¹.



⁶⁰⁵ <https://medium.com/@boutnaru/the-windows-security-journey-vbs-virtual-based-security-d4d7b1f60475>

⁶⁰⁶ <https://learn.microsoft.com/en-us/windows-hardware/drivers/bringup/device-guard-and-credential-guard>

⁶⁰⁷ <https://www.neowin.net/news/microsofts-vbshvci-still-hurts-windows-11-performance-even-on-latest-versions/>

⁶⁰⁸ <https://learn.microsoft.com/en-us/windows/security/hardware-security/enable-virtualization-based-protection-of-code-integrity?tabs=security>

⁶⁰⁹ [https://learn.microsoft.com/en-us/windows-hardware/test/hlk/testref driver-compatibility-with-device-guard?source=recommendations](https://learn.microsoft.com/en-us/windows-hardware/test/hlk/testref	driver-compatibility-with-device-guard?source=recommendations)

⁶¹⁰ <https://www.windowspro.de/wolfgang-sommergut/secured-core-windows-server-2022-hvci-dma-schutz-system-guard-vbs-konfigurieren>

⁶¹¹ <https://technoresearch.com/how-to-enable-or-disable-memory-integrity-in-windows-11/>

Managed Service Accounts (MSA)

MSA (Managed Service Account) is a feature introduced as part of Windows Server 2008 R2\Windows 7. It provides the ability to create an account as part of “Active Directory”⁶¹² which is tied to a specific computer account - as shown in the screenshot below. The account can be used for running Windows services⁶¹³ on the computer while allowing it to connect to network resources with a specific user. Also, the account has a complex password which is automatically maintained⁶¹⁴.

Overall, MSA can log to AD (Active Directory) using NTLM or Kerberos. In order for it to work, when creating a MSA we configure which other accounts can read its password (PrincipalsAllowedToRetrieveManagedPassword). This is done by managing the ACL on that specific attribute⁶¹⁵.

Lastly, there are different types of “Managed Service Accounts”⁶¹⁶: dMSA (Delegated Managed Service Account), gMSA (Grouped Managed Service Account) and sMSA (Standalone Managed Service Account).

The screenshot shows the Windows Active Directory Users and Computers (ADUC) interface. The left pane displays a tree view of the directory structure under 'Active Directory Users and Computers'. A 'Managed Service Accounts' folder is selected. The right pane lists the contents of this folder, showing a single entry: 'MSA1'. The table has three columns: 'Name', 'Type', and 'Description'. The 'Name' column shows 'MSA1', the 'Type' column shows 'msDS-ManagedServiceAccount', and the 'Description' column shows 'What's up, ya'll?'. The entire screenshot is framed by a blue border.

Name	Type	Description
MSA1	msDS-ManagedServiceAccount	What's up, ya'll?

⁶¹² <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-af9e795f86b4>

⁶¹³ <https://medium.com/@boutnaru/windows-services-part-2-7e2bdb5bce4>

⁶¹⁴ <https://techcommunity.microsoft.com/blog/askds/managed-service-accounts-understanding-implementing-best-practices-and-troublesh/397009>

⁶¹⁵ <https://syfuhs.net/how-managed-service-accounts-in-active-directory-work>

⁶¹⁶ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/delegated-managed-service-accounts/delegated-managed-service-accounts-overview>

Standalone Managed Service Account (sMSA)

”Standalone Managed Service Accounts” (sMSAs) are domain user accounts⁶¹⁷. In case we can’t leverage a gMSA⁶¹⁸ we can use a sMSA. sMSAs require at least “Windows Server 2008 R2”. As opposed to gMSA a sMSA can run on only one server but can be used for multiple services⁶¹⁹.

Overall, an example use-case in which we can just use sMSA (and not gMSA) is of failover clusters. sMSA provides automatic password management, simplified service principal name (SPN) management, and the ability to delegate the management to other administrators⁶²⁰.

Lastly, for listing all the sMSAs account in a domain we can leverage the following command “Get-ADServiceAccount -Filter * | where { \$_.objectClass -eq "msDS-ManagedServiceAccount" }”⁶²¹. We cannot use this type of account for interactive logons. Also, we can’t leverage “runas.exe”⁶²² for using an sMSA, however we can use “psexec.exe” for that. It is done by using the “~” symbol as a password which means that the computer needs to get the account password from “Active Directory”⁶²³ - as shown in the screenshot below.

The screenshot shows a Windows command prompt window. The command entered is:

```
PS C:\tools> ./PsExec64.exe -i -u r...ce\gmsaMunSQL1$ -p ~ cmd.exe
```

PsExec v2.32 - Execute processes remotely
Copyright (C) 2001-2021 Mark Russinovich

S: [ca] Administrator: C:\Windows\system32\cmd.exe

Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
r...e\gmsamunsql1\$

⁶¹⁷ <https://medium.com/@boutnaru/the-windows-security-journey-domain-user-account-d01486510e03>

⁶¹⁸ <https://medium.com/@boutnaru/the-windows-security-journey-group-managed-service-account-gmsa-467bd387f5a6>

⁶¹⁹ <https://docs.azure.cn/en-us/entra/architecture/service-accounts-on-premises>

⁶²⁰ <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-managed-service-accounts/group-managed-service-accounts-overview>

⁶²¹ <https://learn.microsoft.com/en-us/entra/architecture/service-accounts-standalone-managed>

⁶²² <https://medium.com/@boutnaru/the-windows-process-journey-runas-exe-run-as-utility-3c1e0b8aaa67>

⁶²³ <https://woshub.com/group-managed-service-accounts-in-windows-server-2012/>

Group Managed Service Account (gMSA)

”Group Managed Service Accounts” (gMSAs) are domain user accounts⁶²⁴ which are used for securing services. They can be used on one or more servers, when configured the password management of the account is done by the operating system. Thus, the benefits are: setting strong passwords (240-bytes randomly generated), cycling password regularly (30 days), supporting server farms and supporting simplified service principal name (SPN) management⁶²⁵.

Overall, we should use gMSA whenever we can. In case it is not supported we can use sMSA (Standalone Managed Service Account) - more on that in future writeups. gMSA have a class of “msDS-GroupManagedServiceAccount”, which is used to create an account which can be shared by different computers to run Windows services⁶²⁶.

Lastly, gMSAs have the following attributes: “msDS-ManagedPassword” (a blob containing the gMSA’s password), “msDS-ManagedPasswordID” (key ID used for generating the current gMSA password), “msDS-ManagedPasswordPreviousID” (the key ID used for generating the previous password) and “msDS-GroupMSAMembership” (list of objects which have permissions to query the password) and “msDS-ManagedPasswordInterval” (interval in days for rotating the password). We can leverage the “Get-ADServiceAccount” PowerShell’s cmdlet for finding gMSAs⁶²⁷ - as shown in the screenshot below.

```
PS C:\Users\notadmin> Get-ADServiceAccount -Filter *

DistinguishedName : CN=ManServAcct,CN=Managed Service Accounts,DC=sbpmlab,DC=net
Enabled          : True
Name             : ManServAcct
ObjectClass      : msDS-GroupManagedServiceAccount
ObjectGUID       : ddfb459e-e399-41ec-8c50-df388023df12
SamAccountName   : ManServAccts
SID              : S-1-5-21-1291669774-2486919405-3319405862-6272
UserPrincipalName :

DistinguishedName : CN=PasswordTest,CN=Managed Service Accounts,DC=sbpmlab,DC=net
Enabled          : True
Name             : PasswordTest
ObjectClass      : msDS-GroupManagedServiceAccount
ObjectGUID       : 5e384514-aa64-4d9c-9f12-8d1342ab3e64
SamAccountName   : PasswordTest$
SID              : S-1-5-21-1291669774-2486919405-3319405862-6613
UserPrincipalName :

DistinguishedName : CN=gMSA,CN=Managed Service Accounts,DC=sbpmlab,DC=net
Enabled          : True
Name             : gMSA
ObjectClass      : msDS-GroupManagedServiceAccount
ObjectGUID       : d8893925-b034-48f8-8bea-df2f3faac963
SamAccountName   : gMSA$
SID              : S-1-5-21-1291669774-2486919405-3319405862-6421
UserPrincipalName :
```

⁶²⁴ <https://medium.com/@boutnaru/the-windows-security-journey-domain-user-account-d01486510e03>

⁶²⁵ <https://learn.microsoft.com/en-us/entra/architecture/service-accounts-group-managed>

⁶²⁶ <https://learn.microsoft.com/en-us/windows/win32/adschema/c-msds-groupmanagedserviceaccount>

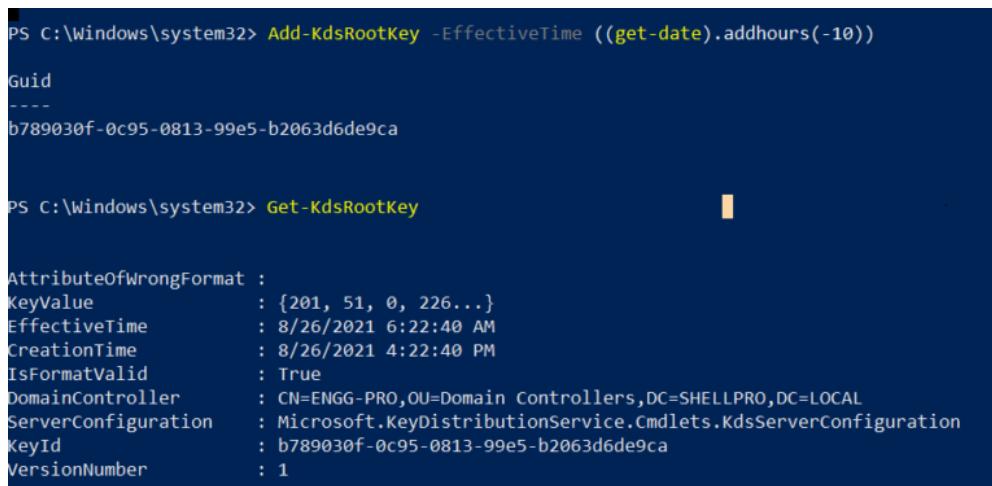
⁶²⁷ <https://blog.netwrix.com/2022/10/13/group-managed-service-accounts-gmsa/>

KDS Root Keys (Key Distribution Service Root Keys)

“KDS Root Keys” (Key Distribution Service Root Keys) are used to generate group managed service account passwords aka “gMSA”⁶²⁸. The KDS root keys are stored in active directory⁶²⁹ as part of “CN=Master Root Keys,CN=Group Key Distribution Service,CN=Services,CN=Configuration,DC=<forest name>;”. Also, the keys have an attribute named “msKds-DomainID” which links to the domain controller that created the object.

Overall, we can use the PowerShell⁶³⁰ cmdlet named “Add-KdsRootKey” for generating a new KDS root key - as shown in the screenshot below. Using the “EffectiveImmediately” causes the KDS root key to be added to the target DC (domain controller) immediately. However, other DCs will not be able to use it until replication is successful. This is due to the fact DCs wait for 10 hours from the time of creation to allow all DC to converge their AD replication⁶³¹.

Lastly, for ensuring the key was created we can leverage the “Get-KdsRootKey” cmdlet⁶³² - also shown in the screenshot below. For more information regarding the PowerShell KDS module I suggest going over the relevant Microsoft documentation⁶³³.



```
PS C:\Windows\system32> Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10))

Guid
-----
b789030f-0c95-0813-99e5-b2063d6de9ca

PS C:\Windows\system32> Get-KdsRootKey

AttributeOfWrongFormat :
KeyValue          : {201, 51, 0, 226...}
EffectiveTime     : 8/26/2021 6:22:40 AM
CreationTime      : 8/26/2021 4:22:40 PM
IsFormatValid    : True
DomainController  : CN=ENGG-PRO,OU=Domain Controllers,DC=SHELLPRO,DC=LOCAL
ServerConfiguration : Microsoft.KeyDistributionService.Cmdlets.KdsServerConfiguration
KeyId             : b789030f-0c95-0813-99e5-b2063d6de9ca
VersionNumber     : 1
```

⁶²⁸ <https://medium.com/@boutnaru/the-windows-security-journey-group-managed-service-account-gmsa-467bd387f5a6>

⁶²⁹ <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-af9e795f86b4>

⁶³⁰ <https://medium.com/@boutnaru/the-windows-process-journey-powershell-exe-windows-powershell-36daabaa74c4>

⁶³¹ <https://shellgeek.com/group-managed-service-accounts/>

⁶³² <https://learn.microsoft.com/en-us/powershell/module/kds/get-kdsrootkey>

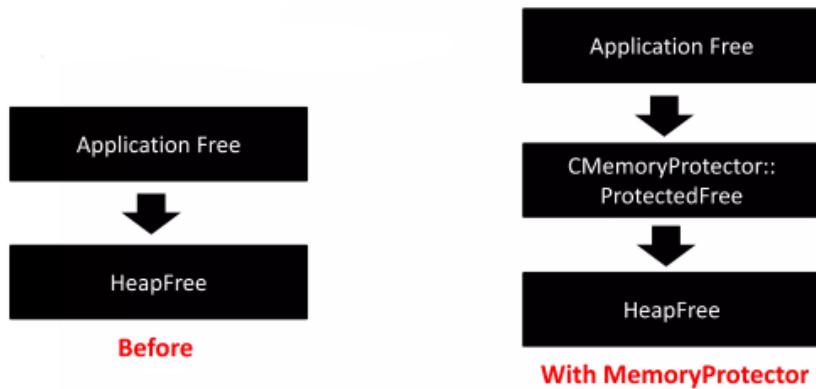
⁶³³ <https://github.com/MicrosoftDocs/windows-powershell-docs/blob/main/docset/winserver2025-ps/kds/kds.md>

MP (Memory Protector)

MP (Memory Protector) is a security mitigation focused on protecting against a sub-set of UAF (Use-after-Free) security issues as part of “Internet Explorer”. UAF are caused in case an instance of an object is freed while a pointer to the object is still kept. In case the dangling pointer is stored on the stack or in registers MP strongly mitigates that. However, in case of an heap reference based UAF there is a need for a security update⁶³⁴.

Overall, MP works in the following manner. When an object instance is deleted, its content is zeroed out and immediately placed into a queue. The actual freeing process is delayed until this queue reaches a threshold size. At that point, the system begins a safety check for each object: it scans registers and all pointer-aligned stack entries to determine if any active pointer still references the object. An object is freed only if no referencing pointer is found; otherwise, it is kept in the queue for a later check⁶³⁵.

Lastly, until MemGC had been introduced MP had been supported by any version of IE. However, now it only applies to IE 10 and below - more on MemGC in future writeups. Hence, the application call to free does not go directly to HeapFree⁶³⁶ but rather to “CMemoryProtector::ProtectedFree” - as shown in the screenshot below⁶³⁷.



⁶³⁴<https://www.blackhat.com/docs/us-15/materials/us-15-Yason-Understanding-The-Attack-Surface-And-Attack-Resilience-Of-Project-Spartans-New-EdgeHTML-Rendering-Engine-wp.pdf>

⁶³⁵<https://www.microsoft.com/en-us/msrc/blog/2016/01/triaging-the-exploitability-of-ieedge-crashes>

⁶³⁶<https://learn.microsoft.com/en-us/windows/win32/api/heapapi/nf-heapapi-heapfree>

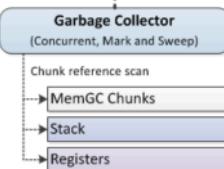
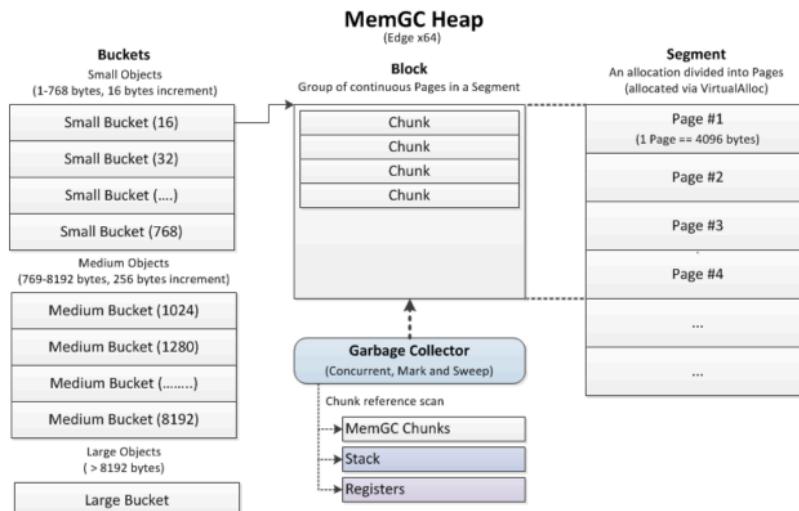
⁶³⁷<https://www.slideshare.net/slideshow/ie-memory-protector/47463710>

MemGC (Memory Garbage Collection)

MemGC (Memory Garbage Collection) is a security mitigation included as part of “Microsoft Edge” and “Internet Explorer 11”. Its goal is to help in protecting against UAF (User-after-Free) security bugs⁶³⁸. UAF are caused in case an instance of an object is freed while a pointer to the object is still kept. It is a successor of the MP (Memory Protector) security mitigation⁶³⁹.

Overall, while MP does not protect in case of a heap reference based UAFs the goal of MemGC is to support that also. It offers additional protection by scanning the heap for references to protected object types (as well as the stack and registers). MemGC will zero write upon free and delay the actual free until garbage collection (GC) is triggered. Thus, protected objects are only freed when no references exist on the stack\heap\registers, offering complete coverage⁶⁴⁰.

Lastly, MemGC was introduced in Windows 10 as a feature of the EdgeHTML and MSHTML engines. It leverages a separate managed heap (also known as the MemGC heap) for object allocation. Its concurrent garbage collector performs a “mark-and-sweep” operation to locate and recover unreferenced memory chunks within this heap - as shown in the diagram below. Crucially, MemGC relies on the Chakra JavaScript engine's memory management routines for the majority of its operations⁶⁴¹.



⁶³⁸ <https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/overview-of-threat-mitigations-in-windows-10>

⁶³⁹ <https://medium.com/@boutnaru/the-windows-security-journey-mp-memory-protector-258fbe93dcc6>

⁶⁴⁰ <https://www.microsoft.com/en-us/msrc/blog/2016/01/triaging-the-exploitability-of-fiedge-crashes>

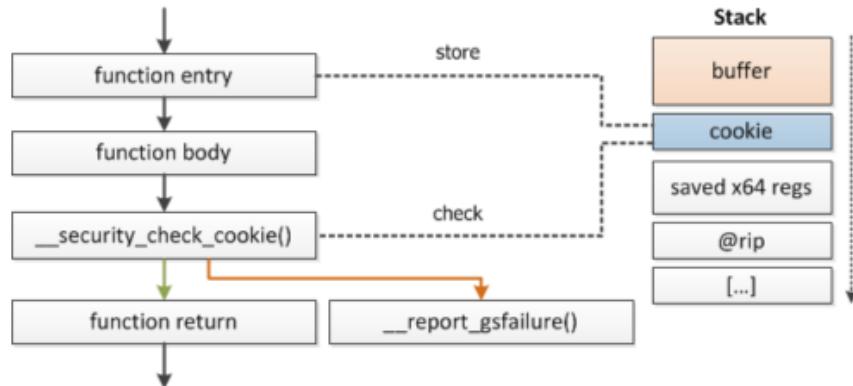
⁶⁴¹ <https://www.blackhat.com/docs/us-15/materials/us-15-Yason-Understanding-The-Attack-Surface-And-Attack-Resilience-Of-Project-Spartans-New-EdgeHTML-Rendering-Engine-wp.pdf>

“/GS” (Buffer Security Check)

“/GS” (Buffer Security Check) is a compiler option used to mitigate stack-based buffer overflows aiming to control the program flow. It is done by placing a “security cookie” on the stack (which should be hard to predict). The cookie is verified before the function exits to confirm the return address and saved registers are intact - as shown in the diagram below. Additionally, it can reorder variables so that critical parameters and local variables precede buffers, thus shielding them from overflow corruption⁶⁴².

Overall, Typically, the C Runtime (CRT) library automatically calls the “`__security_init_cookie`” function during its initialization. However, if the CRT initialization is bypassed (like using the `/ENTRY` compiler option to define a custom entry point) it's the developer's responsibility to manually call “`__security_init_cookie`”. Failing to do so results in the global “security cookie” being assigned a default value. It compromises the buffer overrun protection since this default value is an exploitable weakness that can allow an attacker to bypass security checks⁶⁴³.

Lastly, the “GS Cookie” security feature is not relevant for user-mode code only but also for kernel-mode code⁶⁴⁴. It is also called “stack canaries” and should contain random values generated placed specifically between the function's local variables and its return address⁶⁴⁵.



⁶⁴²<https://www.blackhat.com/docs/us-15/materials/us-15-Yason-Understanding-The-Attack-Surface-And-Attack-Resilience-Of-Project-Spartans-New-EdgeHTML-Rendering-Engine-wp.pdf>

⁶⁴³<https://learn.microsoft.com/en-us/cpp/c-runtime-library/reference/security-init-cookie?view=msvc-170>

⁶⁴⁴<https://j00ru.vexillium.org/2011/01/windows-kernel-mode-gs-cookies-and-1-bit-of-entropy/>

⁶⁴⁵<https://nordsecurity.com/blog/binary-memory-protection-windows-os>

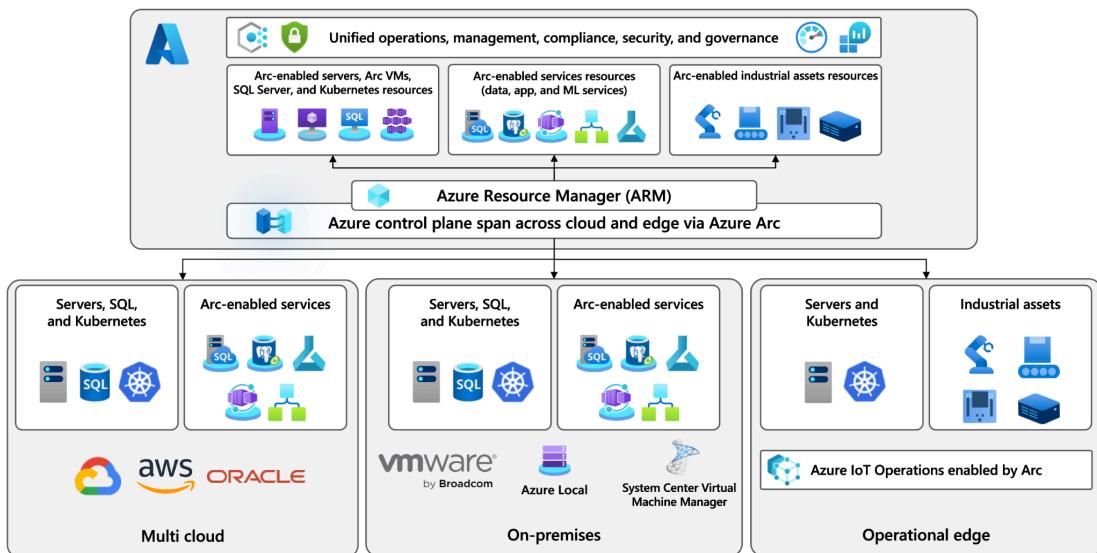
Azure Arc

“Azure Arc” provides the ability for managing multi-cloud and on-premises entities, which simplifies governance. Thus, we can use “Azure Arc” for different use-cases such as (but not limited to): projecting non-Azure and/or on premises resources into “Azure Resource Manager”, and managing VMs (Virtual Machines)\K8s clusters\databases\etc as if they are running on Azure - as described in the diagram below⁶⁴⁶.

Overall, it is important to understand that Arc supports managing Windows\Linux machines hosted outside of Azure like managing native Azure servers (for Arc enabled servers). There are various benefits for using Arc like (but not limited to): leveraging Azure security products for non-Azure resources, getting all the benefits of Azure tools and capabilities, extending “Azure Update Management” to non-Azure servers and because all the data is consolidated in Azure we can query the “Azure Resource Graph”⁶⁴⁷.

Lastly, we can summarize the main features of Azure Arc as: develop cloud native applications and operating them anywhere, meeting regulatory and connectivity needs with flexibility, leveraging data insights from the cloud to the edge and protecting\controlling applications\data\infrastructure across diverse environments⁶⁴⁸.

Azure Arc solution overview architecture



⁶⁴⁶ <https://learn.microsoft.com/en-us/azure/azure-arc/overview>

⁶⁴⁷ <https://www.techielass.com/what-is-azure-arc/>

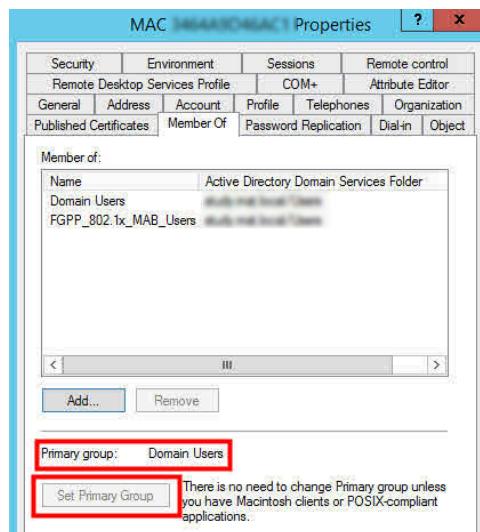
⁶⁴⁸ <https://www.plainconcepts.com/azure-arc/>

Primary Group

In order to be POSIX compliant every user account\group account part of “Active Directory”⁶⁴⁹ has a “Primary Group ID” (PGID) associated with it. This PGID determines the user’s default group membership. By default when a user is created the “Domain Users” group is set as the primary group⁶⁵⁰. It was designed for cross-platform compatibility with other operating systems like Linux⁶⁵¹.

Overall, the program group is stored in an attribute called “primaryGroupID”. Hence, we can read the attribute using PowerShell’s “Get-ADUser” cmdlet⁶⁵² and modify it using PowerShell’s “Set-ADUser” cmdlet⁶⁵³. By the way, we can also leverage the commonly used “Active Directory Users and Computers” console for that - as shown in the screenshot below⁶⁵⁴

Lastly, the primaryGroupID attribute holds the RID aka “Relative Identifier”⁶⁵⁵ from of the primary group⁶⁵⁶. Depending on the object class\operation there are different default values such as: user accounts 513 (Domain Users), computer accounts 514 (Domain Guests), domain controllers accounts 516 (Domain Controllers), read-only domain controllers⁶⁵⁷ 521 (Read-only Domain Controllers) and more⁶⁵⁸.



⁶⁴⁹ <https://medium.com/@boutnaru/the-windows-concept-journey-ad-active-directory-a9e795f86b4>

⁶⁵⁰ <https://www.semperis.com/blog/ad-security-101-primary-group-ids/>

⁶⁵¹ <https://medium.com/@boutnaru/the-linux-security-journey-primary-groups-de2b4d6bd27b>

⁶⁵² <https://learn.microsoft.com/en-us/powershell/module/activedirectory/get-aduser>

⁶⁵³ <https://learn.microsoft.com/en-us/powershell/module/activedirectory/set-aduser>

⁶⁵⁴ <http://www.windows-infrastructure.de/change-users-primary-group-powershell/>

⁶⁵⁵ <https://medium.com/@boutnaru/windows-security-sid-security-identifier-d5a27567d4e5>

⁶⁵⁶ <https://learn.microsoft.com/en-us/windows/win32/adschema/a-primarygroupid>

⁶⁵⁷ <https://medium.com/@boutnaru/the-windows-security-journey-rode-read-only-domain-controller-50560725d4d9>

⁶⁵⁸ <https://learn.microsoft.com/en-us/defender-for-identity/accounts-with-non-default-pgid>

Windows File Protection (WFP)

WFP (Windows File Protection) is a security feature of the Windows operating system. Its goal is preventing the replacement of critical system files. Those are different files based on various file-types are usually installed as part of the OS such as: “*.dll”, “*.ocx”, “*.sys” and some “True Type” fonts. WFP leverages file signatures and catalog files which are created by code signing. This is done in order to verify that protected system files are the in the correct Microsoft versions⁶⁵⁹ - as shown in the screenshot below⁶⁶⁰. We need to not be confused with “Windows Filtering Platform” which can have the same abbreviation of “WFP”⁶⁶¹.

Overall, in case a replacement of such a file is needed (bugs/security vulnerabilities/etc) there are different mechanisms which allow that like “Windows Update” - more on that in future writeups. WFP integrates with the “sfc.exe” utility which is the “System File Checker”⁶⁶². Due to the fact WFP was not designed based on file-systems ACLs⁶⁶³ it is also relevant for non NTFS⁶⁶⁴ systems.

Lastly, WFP was available in Microsoft Windows Server 2003 and Windows XP. WRP (Windows Resource Protection) replaces WFP from Windows Server 2008 and Windows Vista⁶⁶⁵.



⁶⁵⁹ <https://support.microsoft.com/en-us/topic/description-of-the-windows-file-protection-feature-db28f515-6512-63d1-6178-982ed2022fb>

⁶⁶⁰ <http://www.oszone.net/display.php?id=3761>

⁶⁶¹ <https://medium.com/@boutnaru/the-windows-security-journey-wfp-windows-filtering-platform-2e1092e2e55e>

⁶⁶² https://en.wikipedia.org/wiki/Windows_File_Protection

⁶⁶³ <https://medium.com/@boutnaru/the-windows-security-journey-acl-access-control-list-b7d9a6fe4282>

⁶⁶⁴ <https://medium.com/@boutnaru/the-windows-concept-journey-ntfs-new-technology-file-system-433e27a2256a>

⁶⁶⁵ <https://learn.microsoft.com/en-us/windows/win32/wfp/about-windows-file-protection>

Replacement of WFP (Windows File Protection) Protected Files

WFP (Windows File Protection) is a security feature of the Windows operating system which aims at preventing the replacement of critical system files. However, there are specific use-cases in which we need to replace such files for example to solve software bugs/security vulnerabilities⁶⁶⁶.

Overall, to preserve system integrity a file protected can only be successfully replaced if the modification is triggered by one of the few sanctioned Microsoft processes (targeting Windows XP/Windows Server 2003). These authorized mechanisms include installing a “Windows Service Pack” (using Update.exe), applying a system Hotfix (via Hotfix.exe), carrying out an operating system upgrade with “winnt32.exe”⁶⁶⁷ or executing automatic updates delivered through “Windows Update”⁶⁶⁸. - as shown in the screenshot below⁶⁶⁹.

Lastly, Any attempt to alter or replace a WFP-protected file outside of ways described above (officially verified methods will) be blocked and the original file restored. This is done for ensuring the operating system's stability and security⁶⁷⁰.



⁶⁶⁶ <https://medium.com/@boutnaru/the-windows-security-journey-windows-file-protection-wfp-222ee320015c>

⁶⁶⁷ <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/winnt32>

⁶⁶⁸ <https://learn.microsoft.com/en-us/windows/win32/wfp/supported-file-replacement-mechanisms>

⁶⁶⁹ <https://www.petechguide.com/how-to-do-a-clean-installation-of-win98/how-to-update-microsoft-windows-with-patches-and-service-packs>

⁶⁷⁰ https://microsoft.fandom.com/wiki/Windows_File_Protection

How Does Windows File Protection (WFP) Work?

WFP (Windows File Protection) is used for preventing the replacement of critical system files such as (but not limited to): “*.dll”, “*.exe”, “*.sys” and “*.ocx” from being deleted or replaced by unauthorized methods or older versions. Hence, when an attempt is made to replace or delete an unprotected system file, WFP immediately and silently restores the original version from a cached folder containing backup copies⁶⁷¹.

Overall, in the Windows NT family (up to XP\Server 2003), this cache is located at “%SystemRoot%\System32\dllcache”. By the way, Windows Me stores its compressed cabinet setup files in %windir%\Options\Install. WFP validates these files using code signing and checks their digital signatures against catalog files stored in “%SystemRoot%\system32\CatRoot\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}” - shown below⁶⁷². Only specific OS components, such as the Package Installer (Update.exe) or Windows Installer (Msieexec.exe) are authorized to replace protected files. Any other replacement attempts are reverted, and the original file is silently restored from the cache. If WFP can't find the file in the cache, it will search the network path or ask the user to insert the Windows installation disc to restore the correct version. WFP also works with the System File Checker (sfc.exe)⁶⁷³.

Lastly, Windows Vista and later systems no longer use WFP; they feature Windows Resource Protection (WRP). WRP uses ACLs⁶⁷⁴ to protect OS files and aims to protect core registry keys and values to prevent damaging system configuration changes - more on that in future writeups. It is important to understand that WFP was intentionally designed without ACLs. This choice allowed it to function on non-NTFS⁶⁷⁵ systems and prevented "bad" installers from immediately failing due to file access errors. Also, for specifying a different location for the cache we need to create the following registry value: “HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SFCDllCacheDir”⁶⁷⁶.

Name	Date modified	Type	Size
Networking-MPSSVC-Rules-Enterpri...	3/18/2019 3:33 PM	Security Catalog	9 KB
ntprint.cat	3/18/2019 10:43 PM	Security Catalog	17 KB
ntprint4.cat	3/18/2019 10:43 PM	Security Catalog	11 KB
oem0.cat	2/16/2018 1:14 PM	Security Catalog	11 KB
oem2.cat	3/18/2019 10:43 PM	Security Catalog	11 KB
oem3.cat	3/18/2019 10:45 PM	Security Catalog	10 KB
OpenSSH-Client-Package-31bf3856...	3/18/2019 2:42 PM	Security Catalog	9 KB
OpenSSH-Client-Package-31bf3856...	10/31/2020 1:34 PM	Security Catalog	9 KB
OpenSSH-Client-Package-Wrapper...	10/31/2020 1:34 PM	Security Catalog	9 KB

⁶⁷¹ <https://medium.com/@boutnaru/the-windows-security-journey-windows-file-protection-wfp-222ee320015e>

⁶⁷² https://support.arcserve.com/s/article/How-To-Remove-and-Replace-the-ShadowProtect-Snapshot-Driver?language=en_US

⁶⁷³ https://en.wikipedia.org/wiki/Windows_File_Protection

⁶⁷⁴ <https://medium.com/@boutnaru/the-windows-security-journey-acl-access-control-list-b7d9a6fe4282>

⁶⁷⁵ <https://medium.com/@boutnaru/the-windows-concept-journey-ntfs-new-technology-file-system-433e27a2256a>

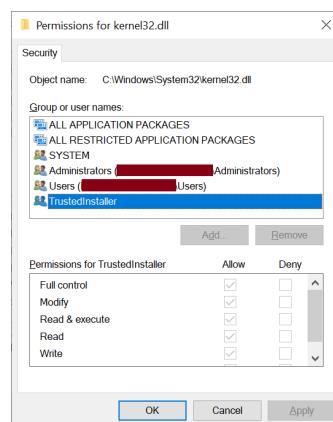
⁶⁷⁶ <https://learn.microsoft.com/en-us/windows/win32/wfp/protected-file-list>

WRP (Windows Resource Protection)

WRP (Windows Resource Protection) is a security feature of the Windows OS. It is used for preventing replacement of essential system files\folders\registry keys that are installed as part of the operating system⁶⁷⁷. It protects critical files with various extensions: “*.acm”, “*.ade”, “*.adp”, “*.app”, “*.asa”, “*.asp”, “*.aspx”, “*.ax”, “*.bas”, “*.bat”, “*.bin”, “*.cer”, “*.chm”, “*.clb”, “*.cmd”, “*.cnt”, “*.cnv”, “*.com”, “*.cpl”, “*.cpx”, “*.crt”, “*.csh”, “*.dll”, “*.drv”, “*.dtd”, “*.exe”, “*.fpx”, “*.grp”, “*.h1s”, “*.hlp”, “*.hta”, “*.ime”, “*.inf”, “*.ins”, “*.isp”, “*.its”, “*.js”, “*.jse”, “*.ksh”, “*.lnk”, “*.mad”, “*.maf”, “*.mag”, “*.mam”, “*.man”, “*.maq”, “*.mar”, “*.mas”, “*.mat”, “*.mau”, “*.mav”, “*.maw”, “*.mda”, “*.mdb”, “*.mde”, “*.mdt”, “*.mdw”, “*.mdz”, “*.msc”, “*.msi”, “*.msp”, “*.mst”, “*.mui”, “*.nls”, “*.ocx”, “*.ops”, “*.pal”, “*.pcd”, “*.pif”, “*.prf”, “*.prg”, “*.pst”, “*.reg”, “*.scf”, “*.scr”, “*.sct”, “*.shb”, “*.shs”, “*.sys”, “*.tlb”, “*.tsp”, “*.url”, “*.vb”, “*.vbe”, “*.vbs”, “*.vsmacros”, “*.vss”, “*.vst”, “*.vsw”, “*.ws”, “*.wsc”, “*.wsf”, “*.wsh”, “*.xsd” and “*.xsl”⁶⁷⁸.

Overall, WRP sets ACLs⁶⁷⁹ with permissions for full access to modify the protected resources to “TrustedInstaller”⁶⁸⁰ - as shown in the screenshot below. Moreover, WRP safeguards crucial registry keys from Windows Server 2008 and Windows Vista, and it can also protect all of a protected key's subkeys and values⁶⁸¹. “Windows Resource Protection” is the new name of “Windows File Protection”⁶⁸²

Lastly, WRP stores the files necessary for a Windows restart in a fixed-size, unchangeable cache directory located at “%Windir%\winsxs\Backup”. Files considered critical but not required for restarting Windows are excluded from this cache, and the list of cached files cannot be altered. It is also integrated with “sfc.exe” (System File Checker)⁶⁸³. There are specific mechanisms which support resource replacement like in case of Windows update⁶⁸⁴.



⁶⁷⁷ <https://learn.microsoft.com/en-us/windows/win32/wfp/about-windows-file-protection>

⁶⁷⁸ <https://learn.microsoft.com/en-us/windows/win32/wfp/protected-file-list>

⁶⁷⁹ <https://medium.com/@boutnaru/the-windows-security-journey-acl-access-control-list-b7d9a6fe4282>

⁶⁸⁰ <https://medium.com/@boutnaru/the-windows-security-journey-trusted-installer-dc44c73ef04c>

⁶⁸¹ https://www.reddit.com/r/sysadmin/comments/4ziu4w/take_ownership_of_registry_key_from/

⁶⁸² <https://medium.com/@boutnaru/the-windows-security-journey-windows-file-protection-wfp-222ee320015c>

⁶⁸³ https://en.wikipedia.org/wiki/Windows_Resource_Protection

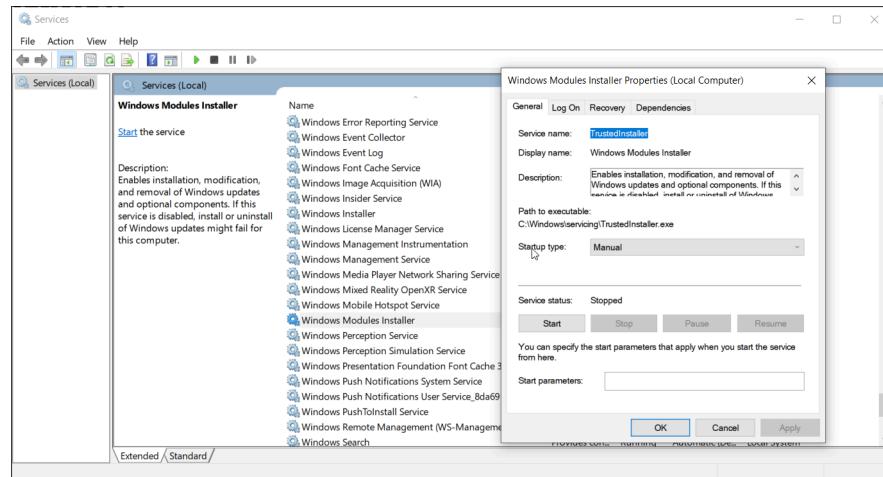
⁶⁸⁴ <https://learn.microsoft.com/en-us/windows/win32/wfp/supported-file-replacement-mechanisms>

Replacement of WRP (Windows Resource Protection) Protected Files

WRP (Windows Resource Protection) is a security feature of the Windows OS which aims at preventing the replacement of critical system files\folder\registry keys. Windows Vista\Windows Server 2008 (and later) full modification access to WRP-protected resources is strictly limited to the “TrustedInstaller” user account. However, there are specific use-cases in which we need to replace such files for example to solve software bugs/security issues⁶⁸⁵.

Overall, the replacement of critical files (protected by WRP) can be done by: “Windows Service Packs” installed by “TrustedInstaller”, hotfixes installed by “TrustedInstaller”, operating system upgrades installed by “TrustedInstaller” and “Windows Update” installed by “TrustedInstaller”. Thus, any attempt to modify the resource outside of these specified methods is blocked and will result in an access denied error message. This is done by leveraging the “Windows Module Installer” service⁶⁸⁶ - as shown below⁶⁸⁷.

Lastly, to ensure backward compatibility a special application compatibility layer (or "shim") is used for well-known installers. This shim prevents from receiving an "access denied" error when it attempts to modify a WRP-protected resource. Instead the installer is fooled with a fake "success" code (the actual modification is not performed). We can think about it as similar to file virtualization⁶⁸⁸ and registry virtualization⁶⁸⁹ supported by UAC⁶⁹⁰. It is relevant for: a legacy installer (lacking a compatible manifest and requestedExecutionLevel) or when the application is run by an administrator⁶⁹¹.



⁶⁸⁵ <https://medium.com/@boutnaru/the-windows-security-journey-wrp-windows-resource-protection-7e3cde332379>

⁶⁸⁶ <https://learn.microsoft.com/en-us/windows/win32/wfp/supported-file-replacement-mechanisms>

⁶⁸⁷ <https://www.makeuseof.com/fix-windows-resource-protection-could-not-start-the-repair-service/>

⁶⁸⁸ <https://medium.com/@boutnaru/the-windows-security-journey-file-virtualization-fdeb68e7e174>

⁶⁸⁹ <https://medium.com/@boutnaru/the-windows-security-journey-registry-virtualization-7b8599093746>

⁶⁹⁰ <https://medium.com/@boutnaru/the-windows-security-journey-uac-user-account-control-ce395df5c784>

⁶⁹¹ <https://www.microsoftpressstore.com/articles/article.aspx?p=2201310&seqNum=2>