10

Reference Guide, Tips, and Tricks

You've reached the end of the book – well done and thank you for investing your time! As we mentioned at the start of the book, the intent of this publication is to remain relevant for a long time. So, we couldn't just stop at telling you all about the product features, helping you find out which deployment and configuration tools are most suitable for your environment, and helping you get to grips with what the day to day looks like after implementation. This chapter is where we hope to provide you with a great reference guide that should keep you coming back long after you're up and running.

Here's what to expect from this chapter:

- Useful commands for use in daily operations
- Tips and tricks from the experts
- Reference tables
- Logs and other useful output

Useful commands for use in daily operations

In this section, we will go over some of the most useful commands that should be part of your basic toolset.

PowerShell reference

You can use PowerShell to control many aspects of the product. The most important cmdlets are listed as follows.

Get-MPComputerStatus

This command outputs the status of the product running on the machine. Some of these settings are used to identify circumstances that are useful for troubleshooting purposes or support cases. The following table shows the output of the command and a description of the values:

Name	Example	Description
`AMEngineVersion`	`1.1.19700.3`	The version of the engine.
`AMProductVersion`	`4.18.2209.3`	The version of the anti-malware platform.

`AMRunningMode`	`Normal`	Can be **disabled**, **normal**, **passive**, or in **EDR block mode** depending on the third-party solution running and the configuration.
`AMServiceEnabled`	`True`	Returns `True` when the antimalware service is in an enabled state.
`AMServiceVersion`	`4.18.2209.3`	Should be the same as the antimalware platform.
`AntispywareEnabled`	`True`	(Legacy) In the past, you would have separate antispyware and antivirus solutions; this value should be the same as `AntivirusEnabled`.
`Antispyware SignatureAge`	`0`	(Legacy) How many days ago spyware definitions were last updated; this value should be the same as `AntivirusSignatureAge`.
`AntispywareSignature LastUpdated`	`1/1/2022 9:00:00 AM`	(Legacy) The last update time for spyware definitions; this value should be the same as `AntivirusSignatureAge`.
`Antispyware SignatureVersion`	`1.375.1808.0`	(Legacy) The version of the spyware definitions; this value should be the same as `Antivirus SignatureVersion`.
`AntivirusEnabled`	`True`	This would show `False` if so configured by group policy – note that this is only possible on servers today, as clients toggle to disabled or running mode through Windows Security Center only.

Name	Example	Description
`AntivirusSignatureAge`	`0`	How many days ago malware definitions were last updated.
`AntivirusSignature LastUpdated`	`1/1/2022 9:00:00 AM`	Last update time for malware definitions.
`AntivirusSignature Version`	`1.375.1808.0`	The version of the malware definitions.
`BehaviorMonitorEnabled`	`True`	Whether the behavior monitoring component is enabled. Note that this is different from **Endpoint Detection and Response** (**EDR**).
`ComputerID`	`C2C9CC10-84F6-1C44-1ABB-839CBDEF09AD`	The ID that **Microsoft Active Protection Service** (**MAPS**) created/uses for the device.
`ComputerState`	`0`	Used in the MDM reporting of the current state according to Defender – for example, whether a scan is pending or a reboot is required for remediation.
`DefenderSignatures OutOfDate`	`False`	Whether Defender determined its definitions are out of date – the default for this is 7 days but this time is configurable.

`DeviceControlDefaultEnforcement`	`Unknown`	What default enforcement mode is configured (default deny/default allow) as the foundation for more granular allow or deny lists.
`DeviceControlPoliciesLastUpdated`	`1/1/2022 9:00:00 AM`	The last update time of device control policies.
`DeviceControlState`	`Disabled`	Whether the device control feature is active.
`FullScanAge`	`4294967295`	Seconds since the last full scan.
`FullScanEndTime`	`1/1/2022 9:00:00 AM`	The time the last full scan ended.
`FullScanOverdue`	`False`	Whether the deadline was missed.
`FullScanRequired`	`False`	If a scan was scheduled but didn't run yet (a missed maintenance window or the machine hasn't been idle).
`FullScanSignatureVersion`	`1.375.1808.0`	What definition update version was loaded at the time of the last full scan.

Name	Example	Description
`FullScanStartTime`	`1/1/2022 8:00:00 AM`	The time the last full scan started.
`IoavProtectionEnabled`	`True`	Whether `IOfficeAntivirus` integration is enabled. This determines whether files are scanned on download and/or open.

`IsTamperProtected`	`True`	Whether **tamper protection (TP)** is enabled.
`IsVirtualMachine`	`False`	Whether the device is a virtual machine.
`LastFullScanSource`	`0`	What triggered the last full scan (scheduled/on-demand).
`LastQuickScanSource`	`2`	What triggered the last full scan (scheduled/on-demand).
`NISEnabled`	`True`	(Legacy) Whether the **network inspection system (NIS)** component is enabled. Now a part of the antimalware service/engine.
`NISEngineVersion`	`1.1.19700.3`	(Legacy) The version of the NIS engine.
`NISSignatureAge`	`0`	(Legacy) How many days ago NIS definitions were last updated; this value should be the same as `AntivirusSignatureAge`.
`NISSignatureLastUpdated`	`1/1/2022 9:00:00 AM`	(Legacy) The last update time for NIS definitions; this value should be the same as `AntivirusSignatureAge`.
`NISSignatureVersion`	`1.375.1808.0`	(Legacy) The version of the NIS definitions; this value should be the same as `AntivirusSignatureVersion`.
`OnAccessProtectionEnabled`	`True`	Whether files are scanned automatically on access – part of real-time protection.
`ProductStatus`	`524288`	The bitmask flag value that represents the cur-

		rent state for use in health reporting.
`QuickScanAge`	`1`	The number of days since the last quick scan.

Name	Example	Description
`QuickScanEndTime`	`1/1/2022 9:01:00 AM`	The time the last quick scan ended.
`QuickScanOverdue`	`False`	Whether a scan was scheduled but hasn't run yet (a missed maintenance window or the machine hasn't been idle).
`QuickScanSignature Version`	`1.375.1710.0`	What definition update version was loaded at the time of the last quick scan.
`QuickScanStartTime`	`1/1/2022 9:00:00 AM`	The time of the last quick scan.
`RealTimeProtection Enabled`	`True`	Whether real-time protection is enabled.
`RealTimeScan Direction`	`0`	Determines whether a particular scan direction was configured – incoming or outgoing files only, or both (`0` = default).
`RebootRequired`	`False`	If there is a pending reboot for remediation purposes, this will show `True`.
`TamperProtection Source`	`Intune`	From which source TP was applied; this can be E5, which is the MDE portal,

		Intune, or ConfigMgr.
`TroubleShootingExpireMinutes`	N/A when **Troubleshooting (TS)** mode is off; otherwise, a number value	The number of minutes left before TS mode expires.
`TroubleShootingMode`	`enabled` or `disabled`	Whether TS mode is active.
`TroubleShootingStartTime`	`2022-07-21 10:00:00 AM`	When TS mode started.
`PSComputerName`	Empty or the hostname of the remote computer	The name of the computer when running the cmdlet through remote PowerShell.

Table 10.1 – Get-MpComputerStatus output

As you can tell, `Get-MpComputerStatus` provides a good view of the overall status of Defender Antivirus prevention capabilities. To retrieve the status of configurable settings, `Get-MpPreference` is used, as we shall see shortly.

Update-MpSignature

This updates antimalware definitions on a computer.

This command will attempt to reach out to not only various sources to update definitions but also the antimalware platform. Success will depend on which update sources are available. If you have defined a proxy server and Windows Update is not reachable, Defender will fall back on the **alternate download location (ADL)** when using this command.

COLD SNACK

In general, Defender will attempt to use the last known successful location for definition updates if the current one is unreachable. So, if you are testing with a proxy server, please note that you may run into this fallback behavior, intended to increase the chances of successfully updating definitions.

Set-MpPreference

`Set-MpPreference` is the cmdlet to configure any type of preference, locally.

`Get-MpPreference` allows you to query the setting whereas `Set-MpPreference` allows you to configure.

`Add-MpPreference` is used for instances where you need to add settings to an existing array, such as ASR rules and exclusions.

The following table contains all the options for `Set-MpPreference`, their descriptions, and some additional information to help you decide whether and how to use them:

PowerShell Set-MpPreference	Description	Notes
`-ProxyBypass <String[]>`	Defines addresses that will bypass the proxy server.	This will allow you to instruct Defender Antivirus on which specific destinations you don't want to use the proxy.
`-ProxyPacUrl <String>`	Defines the proxy auto-config (`.pac`) file location.	If you have a `.pac` file hosted somewhere, you can use this. This applies to Defender Antivirus connections only.
`-ProxyServer <String>`	Defines the proxy server for Defender Antivirus.	This applies to Defender Antivirus connections only. Make sure to put the right notation here; it must have either `http://` or `https://`, depending on your proxy setup.
`-RandomizeScheduleTaskTimes <Boolean>`	Randomizes scheduled task times.	This is the default setting, which ensures that scans and updates don't all occur at the same time across your environment.
`-SchedulerRandomizationTime <UInt32>`	Configures the scheduled task times randomization window – in-	This is especialy useful in shared compute scenarios, such as when

	terval can be anywhere between 0 and 23 hours.	managing your own **virtual desktop infrastructure (VDI)** – ensuring there's not heavy resource contention from all of your VMs simultaneously.
`-PUAProtection <PUAProtectionType>`	Enable/disables detection for potentially unwanted applications. The default is on.	Keep this enabled in an enterprise.
`-DisableAutoExclusions <Boolean>`	Turn off automatic exclusions.	Automatic exclusions apply to server roles on Windows Server 2016 and later.
`-ExclusionExtension <String>`	Define exclusions by file extension – for example, `.csv`, `.jpg`, and `.xyz`.	Not recommended, as it leads to very big blind spots. Consider *contextual exclusions* instead.
`-ExclusionPath <String>`	Defines exclusions by file path. This can apply to either files, folders, or both.	Consider *contextual exclusions* here as well.

PowerShell Set-MpPreference	Description	Notes
`-ExclusionProcess <String>`	Defines process exclusions.	Note that process exclusions don't exclude the process itself but the files it touches. Add a file exclusion for the process if you don't want it scanned (not commonly necessary).

`-DisableBlockAtFirstSeen <Boolean>`	Configures the **block at first sight (BAFS)** feature.	Recommended – does require a sample upload.
`-MAPSReporting <MAPSReportingType>`	Joins the MAPS program for malware submission.	This is required for BAFS.
`-SubmitSamplesConsent <SubmitSamplesConsentType>`	Enables or disables sending file samples when further analysis is required.	Also required for BAFS.
`-AttackSurface ReductionRules_Ids <String[]> and -AttackSurface ReductionRules_Actions <ASRRuleActionType[]>`	Configures **ASR** rules in block, audit, or disabled mode.	Make sure to configure both the rule ID and the action type.
`-AttackSurfaceReduction OnlyExclusions <String[]>`	Excludes files and paths from ASR rules.	Generic exclusions that will apply to all rules.
`-EnableControlled FolderAccess <ControlledFolderAccessType>`	Enables or disables **controlled folder access (CFA)**.	Ensure to also specify the type.
`-ControlledFolderAccess AllowedApplications <String[]>`	Configures which applications are allowed to write to CFA-protected folders.	Used to allow a list of additional applications.
`-ControlledFolderAccess ProtectedFolders <String[]>`	Configures folders to protect with CFA.	Tip – you can define network paths.

PowerShell Set-MpPreference	Description	Notes
`-AllowNetworkProtectionOnWinServer <Boolean>`	This setting is required to allow **network protection** (**NP**) to be configured into block or audit mode on Windows Server.	Additional considerations that apply to servers; high network throughput may lead to a high-performance impact. Consider an alternative approach (such as a network appliance) to cover this functionality if you encounter this.
`-CloudBlockLevel <Cloud BlockLevelType>`	Selects the block level for cloud-delivered protection.	Consider dialing this up from the default if you have a tightly controlled application landscape. For more information, see ***Chapter 2***, *Exploring Next-Generation Protection*.
`-CloudExtendedTimeout <UInt32>`	Configures an extended cloud check in seconds.	For use with BAFS to provide additional time for analysis before the file is allowed to run.
`-EnableFileHash Computation <Boolean>`	Enables the file hash computation feature. The default is off.	Calculate hashes for every file encountered – this will benefit custom indicators to cover more previously unencountered files but comes with a performance trade-off.
`-DisableDatagram Processing <Boolean>`	This setting controls datagram processing for NP.	Enable/disable UDP traffic analysis. Can be performance heavy, so tread

		lightly, particularly on servers.
`-QuarantinePurgeItemsAfterDelay <UInt32>`	Used to configure your own delay, after which items are removed from the quarantine folder.	Use if you want to hold on shorter or longer to quarantined items.
`-DisableScriptScanning <Boolean>`	Enables or disables script scanning.	Keep it on. Toggles AMSI scanning of scripts, which aids in defending against attacks using scripting languages and engines, including obfuscated scripts.

PowerShell Set-MpPreference	Description	Notes
`-DisableBehaviorMonitoring <Boolean>`	Enables or disables behavior monitoring.	The next line of defense; whereas antimalware looks at specific files and processes, behavior monitoring looks at sequences of events to find out whether something is malicious. Keep on whenever possible.
`-DisableIOAVProtection`	Enables or disables scanning of downloaded files and attachments.	Should be on at all times.
`-DisableRealtimeMonitoring <Boolean>`	Enables or disables real-time protection.	Disable for troubleshooting only.
`-RealTimeScanDirection <ScanDirection>`	Used to specify a scan direction (the default is	Useful on file-servers. You can tweak perfor-

	both incoming and outgoing).	mance by selectively disabling the direction for scans – for example, to only scan if a file is placed.
`-RemediationScheduleDay <Day>`	Configures the day of the week that a full scan should run – if it was determined that it was necessary to complete the remediation.	When a threat is cleaned up, but the recommended action is to run a full scan and it's postponed – it will be run according to this schedule.
`-RemediationScheduleTime <DateTime>`	Configures the time that a full scan should run – if it was determined that it was necessary to complete the remediation.	When a threat is cleaned up, but the recommended action is to run a full scan and it's postponed – it will be run according to this schedule.
`-ReportingAdditionalActionTimeOut <UInt32>`	Sets the timeout for detections that require additional action.	The amount of time that's allowed to expire before considering the detection timed out when waiting on additional actions.
`-ReportingCriticalFailureTimeOut <UInt32>`	Sets the timeout for detections that are in a critically failed state.	The amount of time (in minutes) that's allowed to expire before a critically failed detection moves to a cleared state (or of additional actions, if needed).

PowerShell Set-MpPreference	Description	Notes

-ReportingNonCriticalTimeOut <UInt32>	Sets the timeout for detections that are in a non-critical failed state.	The amount of time that's allowed to expire before a non-critical failed detection moves from a failed state to cleared.
-ScanAvgCPULoad Factor <Byte>	Configures the maximum percentage of CPU utilization for scheduled scans.	This is typically more of an average across all cores. You may see it spike, which is not necessarily indicative of a problem. This does not apply to manually started scans! You may see different results if the scan occurs during system-idle periods – see also **DisableCpuThrottleOnIdleScans**.
-CheckForSignaturesBeforeRunningScan <Boolean>	Allows you to configure whether the system should check for the latest security intelligence, before running a scheduled scan.	Useful particularly if real-time protection and cloud-delivered protection are turned off.
-DisableArchiveScanning	Scans archive files.	Leave this on. Note that, by default, the scan will attempt to scan all levels of the archive file.
-DisableCatchupFullScan <Boolean>	Enables or disables a catch-up full scan.	A catch-up scan will be performed if the specified number of scheduled scans has been missed.
-DisableCatchupQuickScan <Boolean>	Enables or disables a catch-up quick scan.	A catch-up scan will be performed if the specified number of sched-

		uled scans has been missed.
`-DisableRestorePoint <Boolean>`	Create a system restore point.	To define whether a restore point should be created before remediation.

PowerShell Set-MpPreference	Description	Notes
`-DisableScanningMappedNetworkDrivesForFullScan <Boolean>`	Enables or disables running full scans on mapped network drives.	The default is off. Consider setting up scans selectively on the file-server hosting the shares, or a particular machine that can perform the scan. Real-time protection on both the client and server provides great coverage as is.
`-DisableScanningNetworkFiles <Boolean>`	Enables or disables the scanning of network files.	Off by default for performance reasons. Consider this for specific scenarios.
`-ScanPurgeItemsAfterDelay <UInt32>`	Configures the period after which items will be removed from the scan history folder.	Likely mostly useful on high-traffic machines that typically encounter a lot of threats.
`-ScanOnlyIfIdleEnabled <Boolean>`	Only runs the scheduled scan if the machine is idle.	Idle is determined by Windows.
`-ScanParameters <ScanType>`	Configures the scan type when configuring a scheduled scan.	Used for setting up a quick or full scan.

`-ScanScheduleDay <Day>`	Configures which day of the week to run a scheduled scan.	Used for setting up a quick or full scan.
`-ScanScheduleQuickScanTime <DateTime>`	Configures the time for a daily quick scan.	Note that daily scans are influenced by the randomization window.
`-EnableLowCpuPriority <Boolean>`	Configures low CPU priority for scheduled scans.	Note that this setting could make a full scan take a very long time, especially on busy machines. Consider quick scans and real-time protection as the go-to.
`-MeteredConnectionUpdates <Boolean>`	Enables Defender Antivirus to perform updates and communicate over a metered connection.	In case the Windows machine considers that it's on a metered – for example, cellular – connection, where data usage can incur cost.

PowerShell Set-MpPreference	Description	Notes
`-SignatureDefinitionUpdateFileShares Sources <String> and '-SignatureBlobFile SharesSources <String>`	Defines from which file share Defender Antivirus should download security intelligence updates.	Used for situations where you don't directly download updates from online sources, or don't use a patch management solution.
`-SharedSignaturesPath <String>`	Defines from which file share Defender Antivirus should down-	Point to a file share with unpacked updates – this will reduce the resources

	load security intelligence updates.	consumed for the unpacking operations in dense environments, such as VDI environments.
-SignatureDisableUpdateOnStartupWithoutEngine <Boolean>	Gives the ability to disable signature updates on startup if needed.	If enabled (set to **True**), a definition update will not occur on startup.
-SignatureFallbackOrder	Configures the fallback order of sources for downloading security intelligence updates.	Set the download source check order. Note that if the first source says there are no updates, the check ends. Only if a source is unavailable will the next source be attempted.
-SignatureScheduleDay <Day>	Configures on which day of the week to check for security intelligence updates.	Used in tandem with the fallback order.
-SignatureScheduleTime <DateTime>	Configures at which time to check for security intelligence updates.	Note that this setting impacts the frequency of applying updates – packages are released multiple times a day, and if a new one is found, it will be applied.
-SignatureUpdateCatchupInterval <UInt32>	Configures the number of days after which a security intelligence update	The default is 7 days. This will be a *catch-up* update, meaning the package will likely be much

	will be required.	larger (no deltas).
`-SignatureUpdateInterval <UInt32>`	Configures how many hours should pass between checks for security intelligence updates.	Alternative to specifying a time of day.

PowerShell Set-MpPreference	Description	Notes
`-ThreatIDDefaultAction_Ids <Int64[]>`	Configures the specific threats upon which default action should not be taken.	For specific named threats only. Not recommended other than for temporarily mitigating false positives.
`-ThreatIDDefault Action_Actions <ThreatAction[]> and -UnknownThreat DefaultAction <ThreatAction> and -LowThreatDefault Action <ThreatAction> and -ModerateThreat DefaultAction <ThreatAction> and -HighThreat DefaultAction <ThreatAction> and -SevereThreatDefault Action <ThreatAction>`	Configures the threat alert levels at which the default action should not be taken.	Protected by the **TP** feature, which is best to leave in a default state.
`-ScanScheduleOffset <UInt32>`	Configures the number of minutes after midnight to execute a scheduled scan.	The way you specify this may seem strange,

		but it helps to not have to create different policies for different time zones.
`-SignatureFirstAuGracePeriod <UInt32>`	Configures the grace period, in minutes, for a security intelligence update. If an update is successful in this period, Defender Antivirus will not initiate any service-initiated updates. This will override the value of `CheckForSignaturesBeforeRunningScan`.	Useful for when there are multiple potential sources for updates and to avoid duplication.
`-DisablePrivacyMode <Boolean>`	A legacy setting that is no longer in use.	The intent of this parameter was to disable privacy mode, which prevented a non-admin user from displaying threat history.

PowerShell Set-MpPreference	Description	Notes
`-Force`	Forces the command to run without any user confirmation.	The standard option for most PowerShell commands. Particularly useful in scripts for commands that could request confirmation.

`-EnableNetworkProtection <ASRRuleActionType>`	Enables or disables NP.	The main enablement for the NP feature. Note that you need to specify whether you want to use audit, block, or disabled, as with ASR rules.
`-EnableFullScanOnBatteryPower <Boolean>`	Enabled or disabled to allow Defender Antivirus to perform a full scan when the machine is not plugged in (running on battery power).	In general, this presents another potential use case where you may want to consider real-time protection and quick scans as the best balance between performance and protection.
`-ForceUseProxyOnly <Boolean>`	Defender Antivirus will only use the proxy as specified in `-ProxyServer`.	Defender Antivirus is opportunistic and will attempt to use any method that was/is (previously) successful. Disallow this behavior using this setting.
`-DisableTlsParsing <Boolean>`	Disables the inspection of TLS (HTTPS) traffic by NP.	Like `DisableDatagramProcessing`, but enabled by default. This can have a significant performance impact on busy machines. By default, NP inspects TLS traffic.
`-DisableHttpParsing <Boolean>`	Disables the inspection of HTTP traffic by NP.	Another control to fine-tune the NP feature.

PowerShell Set-MpPreference	Description	Notes
`-DisableDnsParsing <Boolean>`	Disables the inspection of	Like `DisableDatagram`

	DNS traffic (UDP) by NP.	**`Processing`**, but enabled by default. This can have a significant performance impact on busy machines. This capability can be disabled by setting this value to **`$true`**.
`-DisableDnsOverTcpParsing <Boolean>`	Disables the inspection of DNS traffic (TCP) by NP.	Another control to fine-tune the NP feature.
`-DisableSshParsing <Boolean>`	Disables the inspection of SSH traffic by NP.	Another control to fine-tune the NP feature.
`-Platform UpdatesChannel <UpdatesChannelType>`	Chooses from which channel Microsoft Defender platform updates arrive	Part of gradual rollout controls. For more information, see ***Chapter 7***, *Managing and Maintaining the Security Posture*.
`-Engine UpdatesChannel <UpdatesChannelType>`	Chooses from which channel Microsoft Defender engine updates arrive.	Part of gradual rollout controls. For more information, see ***Chapter 7***, *Managing and Maintaining the Security Posture*.
`-Signatures UpdatesChannel <UpdatesChannelType>`	Chooses from which channel devices receive daily Microsoft Defender definition updates.	Part of gradual rollout controls. For more information, see ***Chapter 7***, *Managing and Maintaining the Security Posture*.
`-DisableGradualRelease <Boolean>`	Disables the gradual rollout of Windows Defender Antivirus updates.	Part of gradual rollout controls and typically used only for more critical devices. For more information, see ***Chapter 7***, *Managing and Maintaining the Security Posture*.

PowerShell Set-MpPreference	Description	Notes
`-AllowNetworkProtectionDownLevel <Boolean>`	Allows NP to be enabled on Windows versions older than 1709.	Additional considerations apply to servers; high network throughput may lead to a high-performance impact. Consider an alternative approach (such as a network appliance) to cover this functionality if you encounter this.
`-AllowDatagramProcessingOnWinServer <Boolean>`	Disables the inspection of UDP connections on Windows servers.	Additional considerations apply to servers; high network throughput may lead to a high-performance impact. Consider an alternative approach (such as a network appliance) to cover this functionality if you encounter this.
`-EnableDnsSinkhole <Boolean>`	This will let NP examine and *sink-hole* (stop) DNS exfiltration attempts and other DNS-based malicious attacks.	Another control to fine-tune the NP feature. Set this configuration to `$true` to enable this feature.
`-DisableInboundConnectionFiltering <Boolean>`	Configures NP to only outbound connections to reduce performance impact (the default is both inbound and outbound).	Another control to fine-tune the NP feature.
`-DisableRdpParsing <Boolean>`	Does not inspect RDP connections.	Another control to fine-tune the NP

feature.

Table 10.2 – The Set-MpPreference options and descriptions

`Set-MpPreference` is a great tool to perform local configuration – note the `Preference` part, indicating that any policy would override these local settings.

MpCmdRun

`MpCmdRun.exe`, also known as the Microsoft Malware Protection Command-Line Utility, is a utility that allows you to perform certain operations that may not be available through PowerShell. Note that administrative permissions are required to use this tool, and the location is not normally in the path (meaning you need to navigate to `%ProgramFiles%\Windows Defender` in your console window or use the full path when calling the utility).

Using the `-?` parameter, you can pull up a list of possible commands. Some useful commands are as follows:

Flag	Description
`-getfiles`	Creates a support `.cab` file, containing a lot of information about the current state of the product by combining many sources of information
`-wdenable`	Attempts to enable services. Deprecated in modern Windows client operating systems (Windows 10+), but still a troubleshooting step when trying to reenable Defender on some server operating systems, such as Server 2016.
`-Trace, -CaptureNetworkTrace`	Useful for capturing traces of specific components for troubleshooting purposes
`-RemoveDefinitions`	Allows you to roll back to the previous engine or remove engine plus definitions. Also provides an option to remove dynamic signatures. Useful for troubleshooting issues; often used to *start fresh* and immediately update definitions to get back to a working state.
`-revertplatform`	Rolls back to the previous antimalware platform
`-resetplatform`	Roll back to the antimalware platform that shipped with the Windows version you are

	running
`-Restore`	Allows you to get files from quarantine
`-CheckExclusion`	Allows you to check whether a path or file is excluded
`-ValidateMapsConnection`	Allows you to check whether the machine can successfully connect to the Defender Antivirus cloud service

Table 10.3 – Common MpCmdRun.exe commands

COLD SNACK

The latest version of `mpcmdrun.exe` *can always be found in* `%programdata\Microsoft\Windows Defender\Platform\<VERSION>` *– this is particularly useful on Windows Server 2012 R2 and 2016, where you may find an outdated version in* `c:\Program Files\Windows Defender`.

macOS/Linux

In macOS and Linux, you can use Terminal/the console to operate the product. The `mdatp` command is available on both operating systems.

mdatp health

The `mdatp health` command provides you with a similar status overview as `Get-MpComputerStatus` on Windows. The following table provides a reference to the possible values and the description:

Value	Description
`automatic_definition_update_enabled`	Will return `true` if automatic antimalware definition updates are enabled.
`cloud_automatic_sample_submission_consent`	The sample submission level. Used to define what happens when a file is determined as malicious and has not been seen before: • **None**: No samples are submitted to Microsoft

	• **Safe**: Only samples that don't typically contain **personally identifiable information** (**PII**) are submitted automatically (default) • **All**: All samples are submitted to Microsoft
`Cloud_diagnostic_enabled`	Enables or disables diagnostic data collection.
`Cloud_enabled`	Enables or disables cloud-delivered protection.
`Conflicting_applications`	A list of applications that could possibly conflict with MDE. This list can include other security products and applications known to cause compatibility issues.
`Definitions_status`	Used to display the status of antimalware definitions.
`Definitions_updated`	Displays the date and time of the last antimalware definition update.
`Definitions_updated_minutes_ago`	Displays the number of minutes that have passed since the last antimalware definition update.
`Definitions_version`	The version of the antimalware definition.

Value	Description
`Edr_client_version`	The version of the EDR client component.
`Edr_configuration_version`	The version of the EDR configuration.
`Edr_device_tags`	Lists the applied tag that can be used for device grouping.
`Edr_group_ids`	The group ID is commonly used for preview feature enablement.
`Edr_machine_id`	The device identifier you will find in the Microsoft 365 Defender portal.
`Engine_version`	The version of the antimalware engine.
`Healthy`	Will be `false` if any of the components is in a bad state.
`Licensed`	Reflects the onboarding status of the device.
`Log_level`	The log level (diagnostic logging, not EDR event capture) that was configured.
`Machine_guid`	Displays the unique machine identifier used by the antimalware component.
`Network_protection_status`	Status of the NP component. This can display the following: • `starting`: NP is starting • `failed_to_start`: There is an error preventing NP from starting • `started`: NP is running • `restarting`: NP is restarting • `stopping`: NP is stopping • `stopped`: NP isn't running
`org_id`	The organization ID of the tenant that the device is onboarded to. If the device isn't

	onboarded, it will display `Unavailable`.
`Passive_mode_enabled`	Displays whether the antimalware component is running in passive mode.
`Product_expiration`	The end-of-support date for the currently installed version of the product
`Real_time_protection_available`	Will display `False` if there is something that is affecting real-time protection.
`Real_time_protection_enabled`	Whether real-time protection is enabled.
`Real_time_protection_subsystem`	The subsystem of the real-time protection component.
`Release_ring`	The release ring. See ***Chapter 7**, Managing and Maintaining the Security Posture*, for more information.

Table 10.4 – mdatp health output and descriptions

mdatp

`mdatp` is the macOS and Linux equivalent of `Set-MpPreference` for Windows. Some useful commands are as follows:

- Update security intelligence:

```
mdatp definitions update
```

- Turn on debug logging:

```
mdatp log level set --level debug
```

- Collect diagnostic logs:

```
mdatp diagnostic create
```

- Revert the log level to informational:

```
mdatp log level set --level info
```

- Test cloud connectivity:

```
mdatp connectivity test
```

COLD SNACK

mdatp connectivity test *is the equivalent of* **mpcmdrun.exe -ValidateMapsConnection**. *Note that the MDATP client analyzer script provides much more extensive coverage for testing all MDE cloud endpoints (and more), but these commands are great for a spot check.*

macOS-specific commands

The following commands are specific to MDE on macOS:

- Change channel to a different ring:

```
defaults write com.microsoft.autoupdate2 ChannelName InsiderFast
```

- Update the app/platform to the most recent version:

```
/Library/Application\ Support/Microsoft/MAU2.0/Microsoft\ AutoUpdate.app/Contents/MacOS/msupd
```

```
/Library/Application\ Support/Microsoft/MAU2.0/Microsoft\ AutoUpdate.app/Contents/MacOS/msupd
```

- Restart **wdavdaemon**:

```
sudo killall -9 wdavdaemon
```

- Collect network provider logs:

```
sudo log stream --predicate 'process MATCHES "netext"' --level debugs
```

- Uninstall MDE:

```
sudo '/Library/Application Support/Microsoft/Defender/uninstall/uninstall'
```

Linux-specific commands

The following commands are specific to MDE on Linux:

- Restart `wdavdaemon`:

```
sudo systemctl restart mdatp / pkill wdavdaemon
```

- Upgrade the platform to the latest version:
  - `sudo yum update mdatp` for Red Hat Enterprise Linux and variants (CentOS and Oracle Linux)
  - `sudo zypper update mdatp` for SUSE Linux Enterprise Server and variants
  - `sudo apt-get install –only-upgrade mdatp` for Ubuntu and Debian systems
- Uninstall the platform:
  - `sudo yum remove mdatp` for Red Hat Enterprise Linux and variants (such as CentOS and Oracle Linux)
  - `sudo zypper remove mdatp` for SUSE Linux Enterprise Server and variants
  - `sudo apt-get purge mdatp` for Ubuntu and Debian systems

Now that you've added those to your toolkit, let's take a look at some useful tips and tricks!

Tips and tricks from the experts

Here are some handy tips and tricks we've collected, with some help from the community:

- Use **https://security.microsoft.com/preferences2** to go straight to the MDE settings in the portal.
- If you are using command-line utilities to troubleshoot, you can use the pipe character to output to the clipboard:

```
"c:\Program Files\Windows Defender\MpCmdRun.exe" -ValidateMapsConnection | clip
```

- **https://gpsearch.azurewebsites.net/** is a great resource to look up Defender settings and their descriptions.
- `@NathanMcNulty` shared the following:
  - **Learning KQL is one of the highest ROI things you can do:**

```
// Find ingestion delay

| extend IngestTime = ingestion_time()
```

```
| project-reorder TimeGenerated,IngestTime
```

  - **The API is incredible, use it**
  - **Live Response can download and execute applications if you wrap them with scripts ;)**
- `@rakidbrahman` shared the following: **Device tags from Intune gives you so much free information. Location, division, use-case (kiosk) and so on.**
- `@JeffreyAppel7` shared the following: **Web content filtering in audit mode. (create policy without any checkbox enabled) to view the category impact/ usage.**
- `@ManuelHauch` shared the following: **Use Procmon to troubleshoot custom applications that seem to interfere with AV/EDR. Use tagging for offboarded machines.**
- `@reprise_99` shared the following: **Use the externaldata operator to enrich your KQL with all kinds of awesome stuff like IP reputation, or lists of LOLbins etc**.
- `@rcegann` shared the following: **Consider using something like Sysmon in conjunction with MDE - Defender records a lot but also has many gaps!**
- `@rpargman` shared the following: **GitHub - olafhartong/sysmon-modular: A repository of sysmon configuration modules (https://github.com/olafhartong/sysmon-modular):**
  - **Never stop checking DeviceEvents for new ActionType values: it will keep surprising you with treasures!**
  - **Use Olaf Hartong's WDACMe to get more visibility of DLL and EXE loading for free**
- `@BertJanCyber` shared the following: **Use the FileProfile function to enrich your file results with prevalence and signer information. Can be done based on the filehash. | invoke FileProfile(SHA1, 1000).**
- `@jmukari` shared the following: **Enable wfp logging to hunt example port scanning.**

Online resources

Of course, you will want to make sure that you have the most up-to-date information available – while the content in this book is written in such a way that it should stay relevant for quite some time, a lot can and will change.

Keep these links handy, and remember that you can always access the learning hub section at **https://security.microsoft.com** to find more learning resources:

- **https://aka.ms/mdeninja** for an excellent collection of learning links
- **https://aka.ms/ninjashow** for some cool videos, including multiple ones about MDE
- **https://aka.ms/mdeblog** for not only blogs but also to access the tech community for MDE
- **https://learn.microsoft.com**, which is the official location for MDE technical documentation

Tips and tricks are always handy, but what about if you just need to look up something?

Reference tables

The following section contains some useful reference tables for various aspects of MDE.

Processes

Here's an overview of the MDE processes per operating system.

Windows 11, Windows 10, Windows Server 2022, and Windows Server 2019, (Server 2012 R2 and Server 2016 with the unified agent)

COLD SNACK

On Windows Server 2012 R2 and 2016, EDR components initially get installed in `C:\Program Files`*. However, you will find that after monthly updates for the EDR, sensor services will start running from the* `C:\Programdata\Microsoft\Windows Defender Advanced Threat Protection\Platform\<VERSION>` *directory instead.*

The following table shows the processes, their location, and their purposes:

Process	Location	Purpose
`MpCmdRun.exe`	`C:\Program Files\Windows Defender`	Antivirus command-line utility
`MpDlpCmd.exe`	`C:\Program Files\Windows Defender`	**Data loss prevention (DLP)** command-line utility
`MsMpEng.exe`	`C:\Program Files\Windows Defender`	Defender Antivirus service
`ConfigSecurityPolicy.exe`	`C:\Program Files\Windows Defender`	Microsoft Security Client Policy Configuration Tool kit
`NisSrv.exe`	`C:\Program Files\Windows Defender`	Defender Antivirus Network Real-Time Inspection/Network Protection service

`MsSense.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	Defender for Endpoint EDR sensor service
`SenseCnCProxy.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	EDR communication module – receives commands
`SenseIR.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	Sense **Incident Response (IR)** module – used for LR and all other commands
`SenseCE.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	Sense **Classification Engine** (**CE**) module – used for DLP
`SenseSampleUploader.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	EDR sample upload module

Process	Location	Purpose
`SenseNdr.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	Sense **Network Detection and Response** (**NDR**) module
`SenseSC.exe`	`C:\Program Files\Windows Defender Advanced Threat Protection`	Sense **Screenshot Capture** (**SC**) module
`SenseCM.exe`	`C:\Program Files\Windows Defender`	Sense **Configuration Management** (**CM**)

`Advanced Threat Protection`	module

Table 10.5 – MDE processes on modern Windows operating systems

Windows 7 SP1, Windows Server 2012 R2, and Windows Server 2008 R2 (SCEP/MMA)

The following table shows the processes, their location, and their purposes on older Windows operating systems, using the legacy, MMA-based client:

Process	Location	Purpose
`MpCmdRun.exe`	`C:\Program Files\Microsoft Security Client`	Antivirus command-line utility
`MsMpEng.exe`	`C:\Program Files\Microsoft Security Client`	Antivirus service
`ConfigSecurityPolicy.exe`	`C:\Program Files\Microsoft Security Client`	Microsoft Security Client Policy Configuration Tool
`NisSrv.exe`	`C:\Program Files\Microsoft Security Client`	Defender Antivirus Network Real-Time Inspection
`MonitoringHost.exe`	`C:\Program Files\Microsoft Monitoring Agent\Agent`	MMA service host
`HealthService.exe`	`C:\Program Files\Microsoft Monitoring Agent\Agent`	MMA communication module
`MsSenseS.exe`	`C:\Program Files\Microsoft Monitoring Agent\Agent\Health Service State\Monitoring`	EDR sensor service – dynamically downloaded by MMA

	`Host Temporary Files *\****`	
`TestCloudConnection.exe`	`C:\Program Files\Microsoft Monitoring Agent\Agent`	MMA cloud connection test utility

Table 10.6 – MDE processes on legacy Windows operating systems

Linux

The following table shows the MDE processes, their location, and their purposes on Linux operating systems:

Process	Location	Purpose
`wdavdaemon`	`/opt/microsoft/mdatp/sbin/`	Core daemon (service). Uses `fanotify` for both antimalware and EDR purposes (TALPA on older RHEL).
`wdavdaemon enterprise`	`/opt/microsoft/mdatp/sbin/`	EDR engine. Used for enrichment, and also leverages `auditd` on most Linux platforms.
`wdavdaemon unprivileged`	`/opt/microsoft/mdatp/sbin/`	AV engine.
`mdatp_audisp_plugin`	`/opt/microsoft/mdatp/sbin/`	Auditd log ingestion.
`crashpad_handler`	`/opt/microsoft/mdatp/sbin/`	Collects crash dumps.

`mdatp`	`/opt/microsoft/mdatp/sbin/ Wdavdaemonclient`	Command-line utility.
`telemetryd_v2`	`/opt/microsoft/mdatp/sbin/`	Telemetry daemon for EDR.
`mde_netfilter`	`/opt/microsoft/mde_netfilter/sbin`	Packet filter for NP, and also used for response capabilities.

Table 10.7 – MDE for Linux processes

macOS

The following table shows the MDE processes, their location, and their purposes on macOS:

Process	Location
`wdavdaemon_enterprise`	`/Library/Application Support/Microsoft/Defender/`
`wdavdaemon_unprivileged`	`/Library/Application Support/Microsoft/Defender/`
`telemetryd_v1`	`/Library/Application Support/Microsoft/Defender/`
`Netext`	`/Library/SystemExtensions/*/com.microsoft.wdav.netext.systemextension/C`
`Epsext`	`/Library/SystemExtensions/*/com.microsoft.wdav.epsext.systemextension/C`
`msupdate`	`/Library/Application\ Support/Microsoft/MAU2.0/Microsoft\ AutoUpdate.app/Contents/MacOS`

Table 10.8 – MDE processes on macOS

ASR rules

Some ASR rules are not available on all Windows operating systems, and for fine-tuning and settings exclusions, looking at the event logs will help

you identify whether exclusions may be needed.

Rules by operating system

The following table is a good reference for what ASR rules are supported on what operating system:

ASR rule	Windows 10+	Windows Server 2012 R2+
Block abuse of exploited vulnerable signed drivers	Y	Y*
Block Adobe Reader from creating child processes	Y*	Y
Block all Office applications from creating child processes	Y	Y
Block credential stealing from the Windows local security authority subsystem (lsass.exe)	Y*	Y
Block executable content from email client and webmail	Y	Y
Block executable files from running unless they meet a prevalence, age, or trusted list criterion	Y*	Y
Block execution of potentially obfuscated scripts	Y	Y
Block JavaScript or VBScript from launching downloaded executable content	Y	Y**
Block Office applications from creating executable content	Y	Y
Block Office applications from injecting code into other processes	Y	Y
Block Office communication application from creating child processes	Y	Y
Block persistence through WMI event subscription	Y*	Y*/**

Block process creations originating from PSExec and WMI commands	Y*	Y
Block untrusted and unsigned processes that run from USB	Y	Y
Block Win32 API calls from Office macros	Y	Y
Use advanced protection against ransomware	Y*	Y

Table 10.9 – ASR rule availability by operating system

**Requires a recent version of the operating system (e.g., 1809 or later)*

***Not available on Windows Server 2012 R2 or 2016 (any ASR rule requires you to be on the unified agent, however)*

ASR rule events and exclusions

If you are fine-tuning rules for your environment, these examples based on events in the Windows Defender event log will help you determine what exclusions to configure.

Look for event ID **`1121`** (**`Warn`**, **`Block`**) or event ID **`1122`** (**`Audit`**). Typically, you will find the ID associated with the rule, the path to the executable, the process name, and the versions of security intelligence. The following table provides some examples of exclusions you may consider:

Rule name	Event log entry with file/folder exclusion example
Block Adobe Reader from creating child processes	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Antivirus has blocked an operation that is not allowed by your IT administrator. For more information, please contact your IT administrator.** **ID: 7674BA52-37EB-4A4F-A9A1-F0F9A1619A2C** **Detection time: xxxx** **User: xxxx** **Path: C:\Users\xxxx\AppData\Local\Microsoft\Edge SxS\ Application\msedge.exe** **Process Name: C:\Program Files (x86)\Adobe\Acrobat Reader 2018\Reader\AcroRd32.exe** Exclusion example(s): • **`%localappdata%\Microsoft\Edge SxS\Application\msedge.exe`**

- `%localappdata%\Microsoft\Edge SxS\Application`

Rule name	Event log entry with file/folder exclusion example
Block all Office applications from creating child processes	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: D4F940AB-401B-4EFC-AADC-AD5F3C50688A** **Detection time: xxxx** **User: xxxx** **Path: C:\Users\xxxx\AppData\Local\Microsoft\Teams\current\Teams.exe** **Process Name: C:\Program Files\Microsoft Office\root\Office15\WINWORD.EXE** Exclusion example(s): • `%localappdata%\Microsoft\Teams\current\Teams.exe` • `%localappdata%\Microsoft\Teams`
Block credential stealing from the Windows local security authority subsystem (lsass.exe)	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Antivirus has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: 9E6C4E1F-7D60-472F-BA1A-A39EF669E4B2** **Detection time: xxxx** **User: xxxx** **Path: C:\Windows\System32\lsass.exe** **Process Name: C:\Program Files (x86)\Google\Update\GoogleUpdate.exe** **Exclusions are generally not required unless the functionality of the blocked application is affected.** Exclusion example(s): • `C:\Program Files (x86)\Google\Update\GoogleUpdate.exe` • `%programfiles(x86)%\Google\Update`

Rule name	Event log entry with file/folder exclusion example
Block executable content from	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: BE9BA2D9-53EA-4CDC-84E5-9B1EEEE46550** **Detection time: xxxx**

email client and web-mail	**User: xxxx** **Path: C:\Users\xxxx\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\XS59XHHJ\dias.zip->dias.exe** **Process Name: C:\Program Files\Microsoft Office\root\Office16\OUTLOOK.EXE** Exclusion example(s): • `%localappdata%\Microsoft\Windows\INetCache\Content.Outlook\*\dias.zip`
Block executable files from running unless they meet a prevalence, age, or trusted list criterion	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: 01443614-CD74-433A-B99E-2ECDC07BFC25** **Detection time: 2021-02-26T01:01:16.000Z** **User: xxxx** **Path: C:\Users\xxxx\AppData\Local\Figma\Figma.exe** **Process Name: C:\Windows\explorer.exe** Exclusion example(s): • `%localappdata%\Figma\Figma.exe` • `%localappdata%\Figma`
Rule name	**Event log entry with file/folder exclusion example**
Block execution of potentially obfuscated scripts	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information please contact your IT administrator.** **ID: 5BEB7EFE-FD9A-4556-801D-275E5FFC04CC** **Detection time: xxxx** **User: xxxx** **Path: C:\Windows\System32\WindowsPowerShell\v1.0\Modules\SmbShare\Smb.types.ps1xml->(SCRIPT0000)** **Process Name: C:\Windows\System32\WindowsPowerShell\ v1.0\powershell.exe** Exclusion example(s): • `C:\Windows\System32\WindowsPowerShell\v1.0\Modules\SmbShare\Smb.types.ps1xml` • `%windir%\System32\WindowsPowerShell\v1.0\Modules`
Block JavaScript or VBScript from launching	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: D3E037E1-3EB8-44C8-A917-57927947596D** **Detection time: xxxx**

downloaded executable content	**User: xxxx** **Path: C:\Program Files (x86)\Tanium\Tanium Client\Downloads\Action_709762\ add-enhanced-tags.vbs** **Process Name: VBScript** Exclusion example(s): • `C:\Program Files (x86)\Tanium\Tanium Client\Downloads\Action_709762\add-enhanced-tags.vbs` • `%programfiles(x86)%\Tanium\Tanium Client\Downloads`
Rule name	**Event log entry with file/folder exclusion example**
Block Office applications from creating executable content	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: 3B576869-A4EC-4529-8536-B80A7769E899** **Detection time: xxxx** **User: xxxx** **Path: C:\Users\xxxx\AppData\Roaming\Grammarly\Updates\ GrammarlyAddInSetup6.7.223.exe** **Process Name: C:\Program Files\Microsoft Office\root\Office16\ WINWORD.EXE** Exclusion example(s): • `%appdata%\Grammarly\Updates\Grammarly AddInSetup6.7.223.exe` • `%appdata%\Grammarly\Updates`
Block Office applications from injecting code into other processes	**Microsoft-Windows-Windows Defender Information 1122** **Windows Defender Exploit Guard audited an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: 75668C1F-73B5-4CF0-BB93-3ECF5CB7CC84** **Detection time: 2021-03-22T13:23:41.365Z** **User: xxxx** **Path: C:\Users\xxxx\Documents\Insights\Predictive_Model_v1.pptx** **Process Name: C:\Program Files (x86)\Microsoft Office\root\Office16\POWERPNT.EXE** Exclusion example(s): • `%userprofile%\Documents\Insights\Predictive_Model_v1.pptx`
Rule name	**Event log entry with file/folder exclusion example**

Block Office communication application from creating child processes	**Microsoft-Windows-Windows Defender Information 1122** **Windows Defender Exploit Guard audited an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: 26190899-1602-49E8-8B27-EB1D0A1CE869** **Detection time: xxxx** **User: xxxx** **Path: C:\Users\xxxx\AppData\Roaming\Grammarly\Updates\GrammarlyAddInSetup6.7.223.exe** **Process Name: C:\Program Files\Microsoft Office\root\Office16\OUTLOOK.EXE** Exclusion example(s): • `%appdata%\Grammarly\Updates\GrammarlyAddInSetup6.7.223.exe` • `%appdata%\Grammarly\Updates`
Block persistence through WMI event subscription	For this rule, exclusions are not supported.
Block process creations originating from PSExec and WMI commands	**Microsoft-Windows-Windows Defender Information 1122** **Windows Defender Exploit Guard audited an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.** **ID: D1E49AAC-8F56-4280-B9BA-993A6D77406C** **Detection time: xxxx** **User: xxxx** **Path: C:\Tools\MDATPClientAnalyzerPreview\Tools\MDATPClientAnalyzer.exe** Exclusion example(s): • `%systemdrive%\Tools\MDATPClientAnalyzer Preview\Tools\MDATPClientAnalyzer.exe` • `%systemdrive%\Tools\MDATPClientAnalyzer Preview\Tools`
Rule name	**Event log entry with file/folder exclusion example**
Block untrusted and unsigned	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.** **For more information, please contact your IT administrator.**

processes that run from USB	**ID: B2B3F03D-6A65-4F7B-A9C7-1C7EF74A9BA4** **Detection time: 2020-08-20T17:29:15.283Z** **User: xxxx** **Path: C:\Users\xxxx\Documents\COMSEC8.2\DIAS\dias.exe** **Process Name: C:\Windows\explorer.exe** Exclusion example(s): • `%userprofile%\Documents\COMSEC8.2\DIAS\dias.exe` • `%userprofile%\Documents\COMSEC8.2`
Block Win32 API calls from Office macros	**Microsoft-Windows-Windows Defender Information 1122** **Windows Defender Exploit Guard audited an operation that is not allowed by your IT administrator. For more information, please contact your IT administrator.** **ID: 92E97FA1-2EDF-4476-BDD6-9DD0B4DDDC7B** **Detection time: 2021-02-24T12:09:32.806Z** **User: xxxx** **Path: C:\Program Files (x86)\Microsoft Office\Office16\Library\SparklinesWMC.xlam->xl/vbaProject.bin** **Process Name: C:\Program Files (x86)\Microsoft Office\root\Office16\EXCEL.EXE** Exclusion example(s): • `C:\Program Files (x86)\Microsoft Office\Office16\Library\SparklinesWMC.xlam` • `%programfiles(x86)%\ Microsoft Office\Office16\Library`
Rule name	**Event log entry with file/folder exclusion example**
Use advanced protection against ransomware	**Microsoft-Windows-Windows Defender Warning 1121** **Windows Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator. For more information, please contact your IT administrator.** **ID: C1DB55AB-C21A-4637-BB3F-A12568109D35** **User: xxxx** **Path: C:\SMARTLINK_DRIVERS\Verification\NEW_INST.EXE** **Process Name: C:\Windows\explorer.exe** Exclusion example(s): • `C:\SMARTLINK_DRIVERS\Verification\NEW_INST.EXE` • `%systemdrive%\SMARTLINK_DRIVERS`

Table 10.10 – ASR events and exclusion suggestions

Using the preceding table as a reference, you should be able to examine event logs on your machines to make informed decisions about setting ASR exclusions.

Settings

The best place to look up all possible settings and descriptions would be an online resource. That said, here are some important and useful things to know about MDE settings.

Interdependent settings

The following table shows which settings have an interdependency, meaning that for a configuration to be valid, all items in the **Depends on** column must be true aside from the enablement flag for the feature itself:

NOTE

Note that for alignment across platforms and simplicity's sake, we've used the generic terms On and Off in the following tables for the required state; not all management interfaces will use this vernacular (it could be enabled/disabled, allowed/not allowed, blocked/not blocked, and so on).

Platform	Setting	Linux and macOS equivalent	Depends on
Windows Server, Linux, or macOS	Passive mode	Enforcement level for the antivirus engine	Doesn't depend on a setting but the Defender Antivirus feature installed/enabled
Windows, Linux, or macOS	Real-time monitoring	Enforcement level for the antivirus engine	Passive mode = Off Enforcement level = Real-time
Windows, Linux, or macOS	Behavior monitoring	Enables/disables behavior monitoring	Passive mode = Off Enforcement level = Real-time
Windows	Script scanning	Not a separate setting	Passive mode = Off Real-time monitoring = On
Windows, Linux, or macOS	Cloud protection	Cloud-delivered protection preferences	Passive mode = Off Real-time monitoring = On

Windows	PUA protection	Not a separate setting	Passive mode = Off Real-time monitoring = On
Windows	IOAV protection	Not a separate setting	Passive mode = Off Real-time monitoring = On
Windows	Cloud block level	Not a separate setting	Passive mode = Off Real-time monitoring = On Cloud protection = On
Windows	Cloud extended timeout	Not a separate setting	Passive mode = Off Real-time monitoring = On Cloud protection = On
Windows or Linux	Submit samples consent	Enables/disables automatic sample submissions	Passive mode = Off Real-time monitoring = On Cloud protection = On
Windows	Real-time scan direction	Not a separate setting	Passive mode = Off Real-time monitoring = On Cloud protection = On
Platform	**Setting**	**Linux and macOS equivalent**	**Depends on**
Windows	NP	Enforcement level for NP	Passive mode = Off Real-time monitoring = On Cloud protection = On
Windows	ASR rules	None	Passive mode = Off Real-time monitoring = On Cloud protection = On

Table 10.11 – Interdependent features

Configuration locations

MDE has various locations, depending on the operating system, where you can find out which settings have been configured.

Windows

In Windows, MDE configuration can be found in the following registry locations:

Location	Purpose
`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender\`	Contains settings coming from group policies
`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender\Policy Manager`	Contains settings coming from MDM solutions
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender`	Contains preferences
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Advanced Threat Protection`	Contains EDR settings delivered from MDE. Also the location for setting `ForceDefenderPassiveMode` (third-party antivirus coexistence on server OS)

Table 10.12 – MDE for Windows Registry locations

COLD SNACK

The order of precedence is group policy wins over MDM, which wins over preferences. The only deviation to this is during a troubleshooting mode session. Note that group policy has its own precedence order where the last policy that applies wins, starting with local group policy.

macOS

On macOS, you can find configuration files in the following locations:

- Preferences:

```
/Library/Application Support/Microsoft/Defender/
```

- Managed configuration:

```
/Library/Managed Preferences
```

Linux

On Linux, you can find configuration files in the following locations:

- The managed configuration file:

```
/etc/opt/microsoft/mdatp/managed/mdatp_managed.json
```

- The effective configuration file:

```
/etc/opt/microsoft/mdatp/wdavcfg
```

COLD SNACK

DO NOT modify **`wdavcfg`**, *but looking at it does provide some insights into what could be configured using* **`mdatp_managed.json`**.

Now you know where to find configurations so that you know what was applied. If something is still not right, you may want to dive into some logs.

Logs and other useful output

Logs are often a great source of information to find out whether everything is going well – or what went wrong.

Useful logs

Here are some of the most useful MDE logs:

- Windows:
  - **`C:\Windows\Temp\MpSigStub.log`**: This is the update log for Windows Defender
  - **`C:\ProgramData\Microsoft\Windows Defender\Support`** contains various useful logs – particularly, **`MPLog`** can tell you a lot about what Windows Defender is up to

In the Windows event logs, the following locations are useful sources of information as well:

- **`Microsoft-Windows-SENSE/Operational`**: EDR sensor events
- **`Microsoft-Windows-Windows Defender/Operational`**: Protection events

- Linux:
  - **`/var/log/microsoft/mdatp/`**: This is the default log output folder
  - **`install.log`** contains information about the installation
  - **`Microsoft_defender_core_err.log`** contains error output logging
- macOS:
  - **`/Library/Logs/Microsoft/mdatp/`**: This is the default log output folder

- **`install.log`** contains information about the installation

Having access to logs really helps to narrow down what's happening on an endpoint and finding them goes a long way.

Summary

In this chapter, you were able to discover many useful commands, tips, tricks, and references to help you in the day-to-day operation of MDE.

There are many, many more useful commands, tips, and tricks that are not covered in this book. That said, the Defender for Endpoint community is quite extensive, and the product group is always looking for new ways to engage. We encourage you to reach out to us (the authors of this book) and share whatever helps you successfully defend your organization, in depth, every single day.

Thank you for all that you do.

Paul Huijbregts

Joe Anich

Justen Graves