

# Advanced Microsoft Defender Antivirus for Windows

Our dive into MDAV continues in this chapter. In the previous chapter, you learned about some of its basic features, such as scanning and exclusion management. In this chapter, we take a closer look under the hood at the capabilities that really make MDAV powerful.

You will learn about the following:

- How **cloud-delivered protection** improves MDAV's layered approach to endpoint security
- How this manifests itself in features such as **block at first sight (BAFS)**
- The protection MDAV can provide against gray-area applications
- The different **running modes** for MDAV
- **Tamper protection**, important defense in your fight against unauthorized manipulation of MDAV, even by local administrators
- Ongoing management of MDAV – troubleshooting and reporting

To kick off our deep dive into MDAV's additional features, we'll start with one that sits at the heart of them all: cloud-delivered protection.

## Cloud-delivered protection

You've seen the marketing emails. You've read the white papers. You've watched the webinars. They all preach one buzz phrase you're probably now numb to: *the power of the cloud*. As much of a cliché as this now is, we cannot discuss MDAV and MDE without emphasizing the importance of its cloud-delivered protection.

As you learned in *Chapter 2*, Microsoft Defender Antivirus has a layered approach to threat protection, with layers beyond the client using cloud-delivered protection for defense. Detonation, reputation, file classification, behavioral, and metadata-based machine learning engines are all dependent on it. Even client-side capabilities, such as **Antimalware Scan Interface (AMSI)** are enhanced by it, to analyze potential fileless attacks. Suffice it to say, without cloud-delivered protection enabled, you severely limit the system's ability to guard against threats that are not yet included (or cannot be included) in traditional signatures.

Other services that rely on the cloud-delivered protection service include these:

- **BAFS** uses it for rapid assessment of whether a new executable should run

- **EDR in block mode**, as covered in the *Running modes* section of this chapter
- **Custom indicators**, or **indicators of compromise (IoCs)**, which need cloud-delivered protection to assess your customized allow and block lists
- **Tamper protection**, explained later in this chapter, uses it to pull settings from the Microsoft 365 Defender portal
- Some **ASR rules**, covered in the next chapter, are dependent on cloud-delivered protection

#### THE LAND OF PRODUCT NAME CHANGES

*You may see the cloud-delivered protection service referred to as **Microsoft Advanced Protection Service (MAPS)** or *SpyNet* in some literature, system resources, and user interfaces. It probably won't shock you that many of the services, features, and products in the Microsoft Defender branding have gone by several names and, in all probability, will go by several more in the future.*

Whether you use Intune, Configuration Manager, or Group Policy to centrally manage Windows, you should be making sure cloud-delivered protection is enabled in them. You will learn how to do this in the next section. When configuring cloud-delivered protection, we also get the opportunity to set up BAFS, which leverages cloud-delivered protection to control and investigate the execution of new and potentially risky executables. Let's get into it!

## Block at first sight

BAFS is one of the most visible ways cloud-delivered protection can be applied practically. Malicious executable files are a serious concern, and this includes non-portable ones such as Office macros. As of Windows 10 1803, an MDAV client with BAFS enabled will query the hash value of executables with **mark of the web (MOTW)** against the cloud protection service. If the file is new to the cloud telemetry dataset but may pose a risk, it is locked for up to 1 minute and a sample is uploaded (based on your tenant's geography) for further analysis. If a verdict of *malicious* is returned before this timeout period, execution is blocked. Decisions are usually returned in milliseconds based on metadata, but the additional time allows for analysis at levels further along the processing chain.

#### WHAT IS MOTW?

*Files downloaded from the internet (based on zones) are given a MOTW, contained in an NTFS stream, by Windows, if the downloading/extracting software supports it. Most will (such as web browsers), though things get messy with some apps if we're then extracting files from ZIPs or VHDs, and so on. MOTW can be seen in the **Properties** menu of a file in Explorer, stating **This file came from another computer and might be blocked to help protect this computer.***

There is no individual tick box or configuration field for BAFS in any of the admin consoles: it is a combination of settings. First, we must ensure that **Cloud-delivered protection** is enabled. Secondly, we must specify the timeout value. Lastly, we must choose a *cloud-delivered protection level*, also sometimes seen as the *file-blocking level*. Combined, these tell MDAV how to behave if it encounters a new executable.

The timeout value is a minimum of 10 seconds, and you can add an additional 50 seconds for a maximum of 1 minute.

The cloud-delivered protection level can be left in its default state or heightened to **high**, **high plus**, or **zero tolerance** (ooh, scary!). At least **high** is required for BAFS to apply. Microsoft's description of the levels is vague, but what you should know is they translate, approximately, to the level of confidence the cloud-delivered protection service has that they are malware, and to block it:

- A protection level of **high** is the minimum required for BAFS.
- A protection level of **high plus** may have some performance implications but improves security (at the risk of false positives) by blocking malware at a lower level of likelihood of being malicious.
- The protection level of **zero tolerance** won't allow any executables the cloud protection service doesn't recognize for an even greater level of protection. This is ideal for devices that are highly secured, with minimal changes in binaries, such as **privileged access workstations (PAWs)**.

Now that you know what BAFS is and how it can be customized, let's have a look at how you can configure it and enforce cloud-delivered protection using Intune, Configuration Manager, and Group Policy.

## Intune

The following steps will let you configure things for Intune, security management, co-management, or tenant attach scenarios:

1. Navigate to [endpoint.microsoft.com](https://endpoint.microsoft.com), then go to **Endpoint security | Antivirus**.
2. Create a new policy or edit an existing one for the **Windows 10, Windows 11, and Windows Server** platform and the **Microsoft Defender Antivirus** profile. Note that the platform ending with **(ConfigMgr)** is for tenant-attach configuration.
3. The settings we're interested in are noted here:
  1. Ensure **Allow Cloud Protection** is set to **Allowed**.
  2. Set **Cloud block level** to the level you need; at least **High**. An architectural decision for you to make is if you can scope different levels to different types of devices. For example, your privileged users and PAWs should probably have a level of zero tolerance. Developers may require a lower level due to the nature of their work generating executables. Remember, you can change the level later, so potentially start with **High** and move upwards through pilot schemes.

3. Set **Defender Cloud Extended Timeout In Seconds** to, ideally, 50 seconds.
4. It's recommended that you set **Submit Samples Consent to Send all samples automatically**. You do have the option to send safe samples automatically instead. Use of **Always prompt** or **Never send** is strongly discouraged.
4. Proceed to assign your policy to devices in scope:

[Home](#) > [Endpoint security](#) >

### Create profile ...

Microsoft Defender Antivirus

|                          |  |
|--------------------------|--|
| Allow Cloud Protection ⓘ | Allowed. Turns on the Microsoft Active Protection Service. ▼ |
| Cloud Block Level ⓘ      | High ▼   |
| Cloud Extended Timeout ⓘ | <input checked="" type="checkbox"/> Configured               |
| *                        | 50 ▼   |

Figure 6.1 – Creating an MDAV profile with BAFS enabled

When devices next sync with Intune, they will apply the settings and BAFS will be active.

#### WHAT MAKES A SAMPLE “SAFE?”

*The general recommendation is to enable sending all samples, but you may have heightened requirements to respect file privacy. The send all safe samples option for **Cloud-delivered protection** limits uploads to file types that likely won't contain personal information, such as EXE files. When you change to all samples, the scope increases to include files such as Office documents, which although are more likely to contain personal information, you really should consider including them due to the threat of malicious macros. If you're debating between safe samples or all samples, remember that Microsoft stores and retains these based on your own retention and residency choices when starting an MDE instance, and follow regulatory guidance on protecting your privacy. If you are using Office 365 to store data anyway, it logically follows there is no reason to disable all sample submissions.*

## Configuration Manager

If you can't use any of the Intune admin center options in your environment, you can use on-premises management services such as Configuration Manager.

MDAV policies can be managed in the Configuration Manager console under **Assets and Compliance** | **Endpoint Protection** | **Antimalware policies**. You can have one or more policies and can tune each according to your needs. Be careful if you're modifying the default one as, by its nature, you may impact a lot of devices. Let's look at the steps for BAFS in Configuration Manager:

1. In a new or existing antimalware policy, navigate to the **Cloud Protection Service** tab.

2. The settings on this page correspond to the similarly named settings in the *Intune* section:
  1. **Cloud Protection Service membership** should be **Advanced** or **Basic membership**. Windows 10 and later treat these the same.
  2. **Allow users to modify Cloud Protection Service settings** should be set to **No** to avoid end users with elevated rights disabling it.
  3. The **Level for blocking suspicious files** setting should correspond to your needs, using the information explained earlier.
  4. The **Allow extended cloud check to block and scan suspicious files for up to (seconds)** value can be up to 50. Remember that the total is this plus 10 that you cannot edit for up to 1 minute in total.
3. Now, head to the **Advanced** tab. For the **Enable auto sample file submission to help Microsoft determine whether certain detected items are Malicious** setting, choose **Yes**.
4. You can now deploy this antimalware policy to devices that should have your BAFS settings:

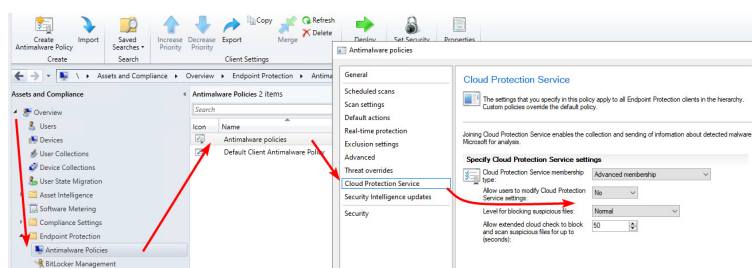


Figure 6.2 – Setting up BAFS using a Configuration Manager antimalware policy

If you don't manage devices with Configuration Manager but they're domain joined, you can also use Group Policy to configure BAFS, which we'll cover next.

## Group Policy

The following settings must be configured in Group Policy to enable BAFS. They can all be found by going to **Computer Configuration | Policies | Administrative Templates | Windows Components | Microsoft Defender Antivirus | MAPS**. Remember that MAPS is the old name for the cloud-delivered protection service:

- **Send file samples when further analysis is required** should be enabled and should be set to **(0x3) Send all samples automatically**. See the guidance in the *Intune* section for more information on the different options.
- **Join Microsoft MAPS** can be set to either **0x2 Advanced** or **0x1 Basic**. Both are processed the same as in Windows 10 and this setting effectively turns on **Cloud-delivered protection**.
- **Configure the 'Block at First Sight' feature** must be enabled.
- Optionally but recommended, **Configure local setting override for reporting to Microsoft MAPS** should be enabled. As a general rule, if you see a setting to disable overrides, it improves security by preventing local administrators from superseding your Group Policy settings.

Now, we'll switch Group Policy paths to **Computer Configuration** | **Policies** | **Administrative Templates** | **Windows Components** | **Microsoft Defender Antivirus** | **MpEngine**:

- **Select cloud protection level** must be enabled and should be set to your preferred blocking level. See the guidance in the *Intune* section for advice on making this decision.
- **Configure extended cloud check** must be set to a value in seconds of up to 50:

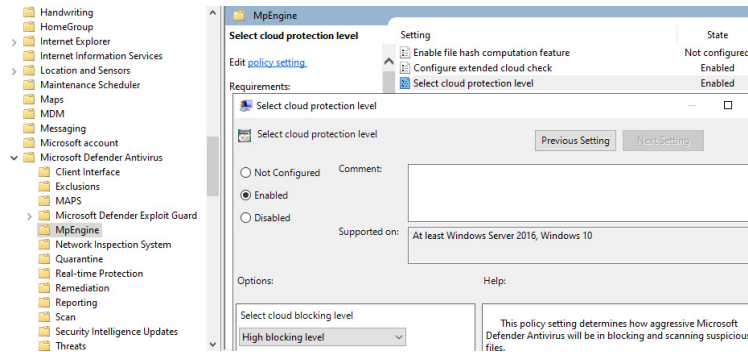


Figure 6.3 – Setting the cloud protection level with Group Policy

With the configuration of your GPO complete, you can link the object to computer OUs for MDAV so that you now have BAfS enabled.

## Confirming BAfS is active on clients

If you are on a client and want to confirm your BAfS settings, you can use PowerShell:

```
Get-MpPreference | FL CloudBlockLevel, CloudExtendedTimeOut, SubmitSamplesConsent, MAPSReporting
```

What's going on here? We are running a function to get all of MDAV's settings, then selecting which of those we're specifically interested in: the BAfS settings. How do you interpret the results?

- **CloudBlockLevel** reports the **Cloud-delivered protection** level. **0** is the default, which means BAfS is not supported. **2**, **4**, and **6** correspond to **high**, **high plus**, and **zero tolerance** respectively.
- **CloudExtendedTimeOut** reports the additional seconds to hold execution.
- **SubmitSamplesConsent** confirms that samples can be uploaded. **3** means all samples will be sent, **2** means no samples will be sent, **0** means prompt the user, and **1** means send safe samples.
- **MAPSReporting** confirms if **Cloud-delivered protection** is enabled. A value of either **1** or **2** in Windows 10 or later is what you're looking for here. **0** means disabled.

To confirm whether clients can access the cloud protection services (or troubleshoot generic problems), you can also use the following command, which must be run with elevated privileges; in my example, I'm doing this using PowerShell:

```
& "C:\Program Files\Windows Defender\MpCmdRun.exe" -ValidateMAPSConnection
```

A successful connection will return **ValidateMapsConnection successfully established a connection to MAPS**. Other messages or errors can be the starting point of your troubleshooting.

To stay on top of configuration drift or missed devices, you can use the **Recommendations** area within **Vulnerability management** in Microsoft 365 Defender ([security.microsoft.com](https://security.microsoft.com)) to track the **Enable cloud-delivered protection** recommendation across your onboarded devices:

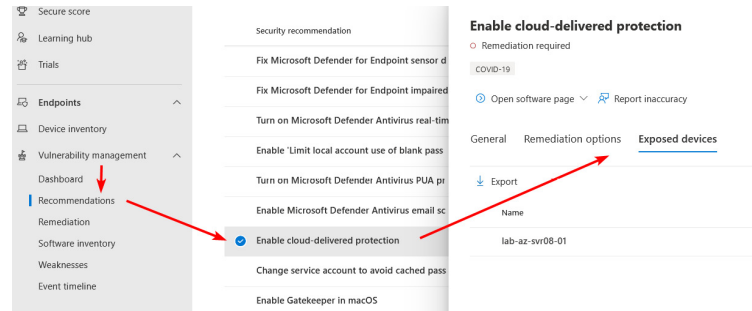


Figure 6.4 – Reviewing devices without cloud-delivered protection in Microsoft 365 Defender

The **Exposed devices** tab shown in the preceding screenshot lists devices that are inadequately configured. However, note that this does not go as far as to enable BAFS, which there is currently no recommendation for.

## Always-on protection

In addition to **Cloud-delivered protection**, MDAV has **always-on protection**. This refers to client-based protective layers used to identify risky files and processes and includes **real-time protection (RPT)** and **behavior monitoring**. Always-on protection is enabled by default but should be force-enabled using your central management tool. Additionally, you can use these management tools to **disable local setting override**. This prevents a local administrator's local settings from overriding your centrally managed settings.


You can find the *always-on protection* settings for Intune in **Endpoint security | Antivirus**, then within the **Microsoft Defender Antivirus** profile type. The settings available include **Allow Realtime Monitoring**, **Allow Behavior Monitoring** (identify threats based on risky behavior such as process, registry, and file activity), and **Allow Intrusion Prevention System** (inspects network traffic for exploits). The general recommendation is to force enable all types of scanning and protection in this profile.

If you're managing MDAV with Configuration Manager, you'll find always-on protection settings within the antimalware policies under **Assets and Compliance | Endpoint Protection | Antimalware Policies |**



<your new or existing policy> | **Real-time protection**. The same guidance mentioned previously applies:

### Real-time protection

 The settings that you specify in this policy apply to all Endpoint Protection clients in the hierarchy. Custom policies override the default policy.

**Specify real-time protection settings**


|  |                                  |
|--|----------------------------------|
|  Enable real-time protection: | Yes                              |
| Monitor file and program activity on your computer:  | Yes                              |
| Scan system files:   | Scan incoming and outgoing files |
| Scan all downloaded files and enable exploit protection for Internet Explorer:                                 | Yes                              |
| Enable behavior monitoring:  | Yes                              |
| Enable protection against network-based exploits:  | Yes                              |
| Allow users on client computers to configure real-time protection settings:                                    | No                               |
| Configure detection for potentially unwanted applications:   | Enabled                          |

Figure 6.5 – Always-on protection settings for MDAV in Configuration Manager

Group Policy environments can find similar settings under **Policies | Administrative Templates | Windows Components | Microsoft Defender Antivirus | Real-time Protection**. Due to Group Policy having a deeper legacy than Intune, you will find settings for GPOs' *always-on protection* that aren't available in Intune or at least don't map directly to the settings in Intune's MDAV endpoint security profile. Generally, you still want to prevent local setting overrides and enforce always-on protection settings.

Always-on/real-time protection is a vital component of any endpoint security service, and you should follow the guidance provided to ensure it is enabled to protect against malware. What constitutes malware is not always obvious, and in the next section, we'll cover a gray area: potentially unwanted apps.

## Potentially unwanted application protection

**Potentially unwanted applications (PUAs)** is a catch-all term to refer to software that is not, strictly speaking, malware but still compromises the machine's integrity in several ways, particularly in the context of enterprise environments. PUAs might bundle lots of other unrelated apps in their installer and obfuscate the fact you're also getting them. PUAs might also spam the user with excessive advertisements, put roadblocks in the way of uninstalling them, or consume excessive resources, potentially sideloaded intensive processes such as hidden cryptocurrency miners.

In short, you probably don't want these touching your devices.

Later in this chapter, you will learn about SmartScreen, a capability targeted at Windows 10/11 that can block PUAs based on reputation and



heuristics. This prevents installation, but what about PUAs already installed when you deploy MDE and MDAV?

For Windows 10 E5 licensed devices (included as part of Microsoft 365 E5, the SKU that most Microsoft 365 Defender customers tend to have), MDAV has its setting to block PUAs on by default as of Windows 10 2004. If PUAs are detected, they'll be quarantined.

There is the risk that supported applications in your environment may get classified as PUAs and quarantined when you deploy MDE/MDAV. If this is a concern, you can set PUA to run in audit mode. You can then review the data to make an informed decision about the consequences of enabling them, potentially requiring exclusions.

Let's review the Intune, Configuration Manager, and Group Policy configuration for managing PUA protection centrally.

## Intune

Follow these steps:

1. Navigate to [endpoint.microsoft.com](https://endpoint.microsoft.com), then go to **Endpoint security | Antivirus**.
2. Create a new policy or edit an existing one for the **Windows 10, Windows 11, and Windows Server** platform and the **Microsoft Defender Antivirus** profile. Use the platform ending in **(ConfigMgr)** for tenant-attach scenarios.
3. You'll find **PUA Protection** as a setting. From here, you can choose **PUA Protection off**, **PUA Protection on**, or **Audit mode**. Disabling it will reverse the automatic enablement of this feature in Windows 10 E5:

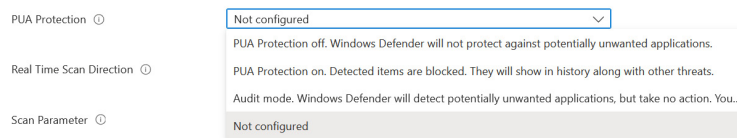


Figure 6.6 – PUA Protection options in an MDAV endpoint security profile

4. Proceed to assign the policy.

## Configuration Manager

Follow these steps:

1. In the Configuration Manager console, head to your antimalware policy or create a new one by going to **Assets and Compliance | Endpoint Protection | Antimalware Policies | <your new or existing policy>**.
2. In the **Real-time protection** section, change **Configure detection for potentially unwanted applications** to **Disabled**, **Enabled**, or **Audit**.

## Group Policy

Follow these steps:

1. In your GPO, navigate to **Computer Configuration | Policies | Administrative Templates | Windows Components | Microsoft Defender Antivirus**.
2. Choose the **Configure detection for potentially unwanted applications** setting.
3. Set it to **Enabled** with the option of **Block**, **Disable**, or **Audit** as required.
4. Link the policy to the devices that should have the PUA protection you've specified:

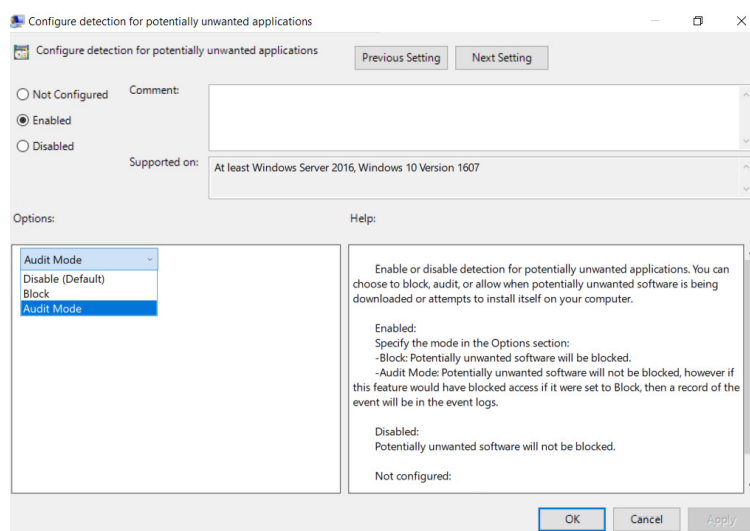


Figure 6.7 – Configuring PUA protection with Group Policy

When devices next sync (Intune) or perform a Group Policy update, the **PUAProtection** value for MDAV, visible with **Get-MpPreference**, will update to **0** (disabled), **1** (enabled), or **2** (audit). If you allow users to see the **Windows Security** app, within **App & browser control | Reputation-based protection**, the settings for **Potentially unwanted app blocking** will be grayed out to restrict alteration. If you're only in audit mode, the **Block apps** box will not be checked:

**This setting is managed by your administrator.**

### Potentially unwanted app blocking

Protect your device from low-reputation apps that might cause unexpected behaviour.



Figure 6.8 – PUA Protection enabled as visible in Windows Security

If you have deployed in audit mode, you'll want to remain on top of monitoring PUAs. Microsoft provides an advanced hunting query for this, which is based on the way MDAV prefixes potentially unwanted apps with PUA:

```
DeviceEvents
| where ActionType == "AntivirusDetection"
| extend x = parse_json(AdditionalFields)
| project Timestamp, DeviceName, FolderPath, FileName, SHA256, ThreatName = tostring(x.ThreatName)
| where ThreatName startswith_cs 'PUA'
```

You now have the knowledge to control PUA protection with the main three management tools. As stated, although it may appear an obvious setting to enable, during migration to MDE/MDAV, you should proceed with some caution as there is a risk of what you perceive to be false positives, which is where audit mode comes in useful.

Next up, we'll look at MDAV scanning, an area that may seem obvious from the surface, but with some nuances to be aware of.

## Running modes

As it is built into the OS, MDAV has different *running modes* to provide compatibility with other endpoint protection software. If no other anti-malware service is running, **normal mode** is used, and MDAV provides its configured threat protection capabilities.

In the presence of a third-party service for endpoint protection, MDAV can enter **passive mode**. This is only an option if the device is onboarded to MDE: consumer or unlicensed devices cannot leverage it, and instead use **disabled mode**. In passive mode, many of the features you will learn about in this chapter enter a state you can think of as hibernating: they are not explicitly disabled but will not be active either, on the assumption the third-party service has been chosen to replace them. The following will not be available in passive mode:

- **Real-time protection** and **Cloud-delivered protection**, and anything that has those as a prerequisite
- **Attack surface reduction (ASR)** rules
- Network protection and services that depend on it, such as web content filtering

MDE-onboarded client OSs will enter passive mode automatically if they identify a third-party replacement and return to active mode if it is removed. Server OSs must be put into passive mode by administrators by setting **REG\_DWORD HKLM\SOFTWARE\Policies\Microsoft\Windows Advanced Threat Protection\ForceDefenderPassiveMode** to **1**. There is no dedicated GPO or cmdlet to achieve this, so you may wish to consider scripts, Group Policy, or **Desired State Configuration (DSC)** if you need to do this at scale.

You may be wondering: if passive mode disables so much, what differentiates it from disabled mode? Unlike disabled mode, devices in passive mode continue to receive updates, but can also run scans on demand. Disabled mode does, however, have a capability targeted at consumer devices called **side-by-side (SxS) passive mode**. This is also called *limited*

*periodic scanning*, which explains its purpose: to provide a reduced scope of protection but still scan devices even in the presence of third-party antimalware. Microsoft does not recommend SxS passive mode for production enterprise use.

If you're using either passive mode or active mode, you should still not attempt to remove or stop services related to Windows Security, such as **SecurityHealthService** or **WSCSVC**. These should remain untouched to allow active mode to resume if needed – for example, if the third-party service is uninstalled.

The big difference between passive mode and disabled mode is that passive mode can be extended into the last mode we'll cover: **EDR in block mode**.

This provides a layered defense against attacks, primarily in scenarios that use a third party for endpoint protection, but it can be enabled for devices that use active mode too. Microsoft describes EDR in block mode as *post-breach protection*, which succinctly describes its use case: for instances when a threat evades your primary endpoint antimalware, but Microsoft Defender for Endpoint's cloud processing suspects something malicious. Defenders are then made aware of these with alerts or incidents in Microsoft 365 Defender. MDAV can also take automated action in this case.

You can enable EDR in block mode for all onboarded devices simply by toggling the radio button in **Microsoft 365 Defender | Settings | Endpoints | Advanced features**. Cloud-delivered protection, which we discussed at the start of this chapter, is a requirement for it to function:

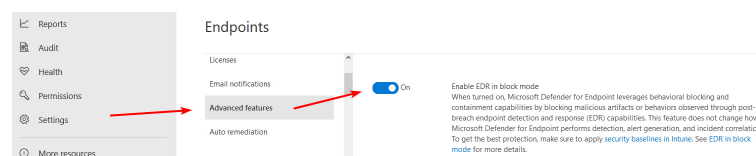


Figure 6.9 – Enabling EDR in block mode for all supported devices

Experience has revealed very few scenarios where you would choose not to enable EDR in block mode, though if you are configuring custom indicators as exclusions, beware that it will not acknowledge them. Currently, only local MDAV exclusions are respected by EDR in block mode.

You can limit EDR in block mode to specific devices, such as a pilot group, using a custom profile for Intune-managed devices (it does not apply to other methods such as Security Management). Alternatively, you can enable it tenant-wide and disable it per device. You cannot do this with Group Policy or other methods. To do so, you must create and assign a custom configuration profile:

1. Navigate to the Microsoft Intune admin center at [endpoint.microsoft.com](https://endpoint.microsoft.com).
2. Browse to **Devices | Windows | Configuration profiles | + Create profile**.

3. Choose **Windows 10 and later** as the platform, **Templates** as the profile type, and then the **Custom** template.
4. After giving the profile a name and description, choose to **Add** the following **OMA-URI settings**:
  1. **Name:** EDR in block mode
  2. **OMA-URI:**

```
./Vendor/MSFT/Defender/Configuration/PassiveRemediation
```
  3. **Data type:** Integer
  4. **Value:** 0 or 1 (1 will enable EDR in block mode; 0 will disable it)
5. Proceed through the custom profile wizard to assign and create this policy for the Azure AD groups of Intune-enrolled devices:

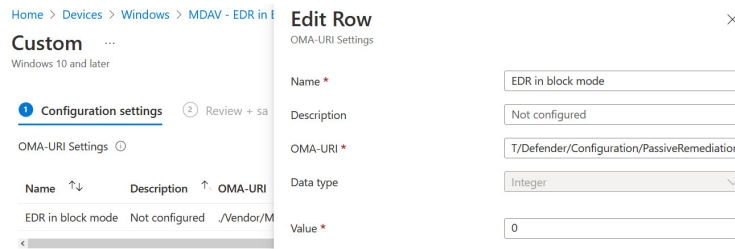


Figure 6.10 – Configuring EDR in block mode with a custom profile for Intune devices

On clients, you can use PowerShell to query the running mode of the device with `Get-MpComputerStatus | FL AMRunningMode`. This information is also sent to the cloud in MDE's sensory information and can be queried with advanced hunting's `DeviceTvmSecureConfigurationAssessment` table.

As you start your deployment of MDE and migrate to it from other solutions, passive and EDR in block mode provide value and make the transition much easier. You do not have to time migrations in a way that leaves the system exposed: you roll out passive mode, remove the existing protection when you're ready, and clients automatically transition to active mode and MDAV.

Now that you understand MDAV's running modes and how they complement third-party services, we'll move on to one of its core features for defense that was briefly touched on earlier: tamper protection.

## Tamper protection

The MITRE ATT&CK tactic of **defense evasion** is one seen frequently in security incidents. This tactic refers to all techniques that avoid, disable, or otherwise circumvent security mechanisms. If an attacker can simply turn off MDE/MDAV when trying to compromise a system, life gets a lot easier. As defenders, we want to stop that.

**Tamper protection** for Microsoft Defender Antivirus is an on-by-default capability to make evasion harder. It is available for Windows 10/Server 2016 or later, and Windows Server 2012 R2 with the unified agent.

Malware or intruders can try to evade MDAV in several ways. The registry editor, PowerShell, Intune, Group Policy (local or Active Directory), and **MpCmdRun.exe**: these all allow a legitimate or illegitimate user to tamper with protection. When enabled, tamper protection restricts such methods of editing settings. Let's reiterate that: when tamper protection is enabled, you cannot disable certain features of MDAV, even as an administrator, regardless of your management tool. When a change is attempted, it is not accepted. Some of the most important tamper-protected values are as follows:

- **DisableIOAVProtection**, which controls how all downloaded files and attachments are scanned
- **DisableEmailScanning**, which controls mail file and mailbox file scanning, such as PST and DBX files
- **DisableScriptScanning**, which excludes scripts from MDAV scans
- **DisableBlockAtFirstSeen**, which can override other settings that, when combined, enable BAFS
- **DisableRealtimeMonitoring**, which can be used to disable real-time protection
- **DisableBehaviorMonitoring**, which disables one element of MDAV's layer-based approach, potentially exposing the system to fileless or living-off-the-land attacks
- **MAPSReporting**, which controls if cloud-delivered protection is enabled
- **RemoveDefinitions**, which is a parameter of **MpCmdRun.exe** that deletes MDAV definitions from the device

Fabian Bader of [cloudbrothers.info](https://cloudbrothers.info) has published very useful research for defenders regarding what kinds of protection and alerts tamper protection provides. Key among those findings is what tamper protection *doesn't* cover:

- **SharedSignaturesPath**, which VDI environments can use to download signatures, rather than every client querying cloud services for them
- **ThreatIDDefaultAction**, which controls the action to take based on the specific ID of the threat

Exclusions not being in the scope of tamper protection jumps out as the most interesting finding. Although Microsoft may change this in the future, this design decision is based on the reality that administrators will have to add and manage exclusions based on real-world problems that software faces, and for vendors to support them. This is not an endorsement of exclusions: you still need to limit them as much as possible. You will also need to continue to monitor them, as we can't use tamper protection as our shield against attackers leveraging them.

If using PowerShell, errors are not presented to the user if tamper protection stops an action. For example, using **Set-MpPreference -DisableRealtimeMonitoring \$true** does not return any output. Instead, it's as if the command is accepted and ignored. What you may find, however, are event ID 5013 entries in **Event Viewer** under **Applications and Services Logs | Microsoft | Windows | Windows Defender | Operational**. There is some inconsistency with which tamper actions

generate logs. For example, **Set-MpPreference -DisableEmailScanning \$true** does not log an event (even though it is blocked), while **Set-MpPreference -DisableRealTimeMonitoring \$true** does. Expect this to evolve as MDAV updates come along:

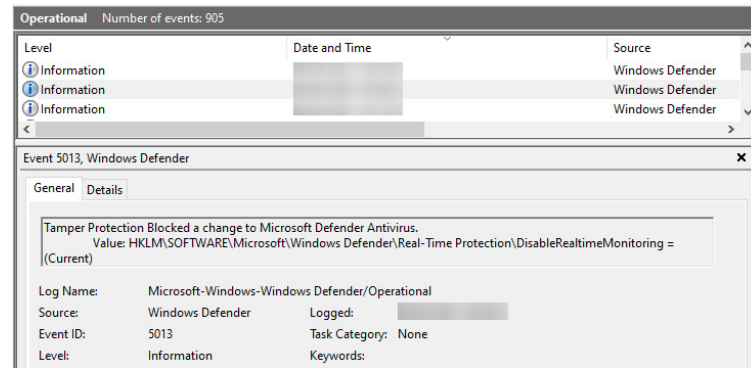


Figure 6.11 – Tamper protection in action, viewed within Event Viewer

Bader’s research continues to point out something important: how are administrators alerted to tamper protection events?

When an attacker attempts to edit the registry using PowerShell (**New-ItemProperty**), **alerts** are created in Microsoft 365 Defender for disabling real-time and cloud-delivered protection. However, these same alerts are not created if you’re using PowerShell directly (**Set-MpPreference**), which generates no alerts for tamper protection events.

In either scenario, **advanced hunting** and the **device timeline** area can be used to uncover most events that tamper protection blocks but don’t create alerts, such as behavior monitoring or removing security intelligence updates. These can also be used for tamper events that are beyond the scope of tamper protection, such as exclusion paths. In [Chapter 19](#), you will learn how you can create custom alerts for scenarios such as tamper protection events.

#### PROTECTION EVOLUTION – AN IMPORTANT CONSIDERATION

*While some of the inconsistencies of tamper protection are true at the time of writing, as an always-changing cloud service, you can and should expect changes over time. As one example, **DisableEmailScanning** was once an unprotected value, but now, it is protected. The point to be aware of is that Microsoft 365 Defender may not necessarily include all configurations in tamper protection: you should still implement hardening techniques already described, such as disabling local policy merge and removing local administrators.*

Now that you are aware of tamper protection’s objectives and scope, let’s dive into how it is applied. There are two ways to deploy tamper protection:

- Tenant-wide using the Microsoft 365 Defender portal
- Scoped using a Microsoft Intune Windows Security experience profile



You'll notice that other management tools are notably absent, which is indicative of the direction of travel for configuring devices.

Tenant-wide is the easiest method: it is as simple as ensuring a button is checked, which, as of September 2022, Microsoft has enabled by default. This relies on cloud-delivered protection to pull down the policy from the cloud and apply it using MDAV:

1. Navigate to **Microsoft 365 Defender | Settings | Endpoints | Advanced features.**
2. Toggle **Tamper protection** on or off:

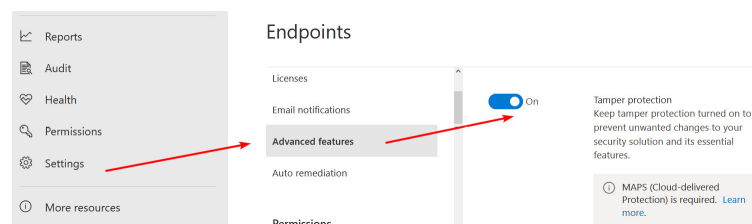


Figure 6.12 – Configuring Tamper protection in the Microsoft 365 Defender portal

Easy! This book recommends enabling tamper protection this way. The negative of this approach is that it's one size fits all: what if you need exceptions?

This is where the ability to control tamper protection with Intune comes in. It can be enabled or disabled here, and either setting takes precedence over your tenant-wide setting. For example, you can disable it for a group of devices if it is enabled for the tenant or you can enable it for groups of devices if it is disabled for the tenant. The general recommendation is to keep it enabled for the tenant and use Intune for any exclusions you may have, ideally keeping this managed by a dedicated group that you can easily remove devices from during configuration changes:

1. Navigate to **Microsoft Intune admin center | Endpoint security | Antivirus.**
2. Choose + **Create Policy** or edit an existing **Windows Security experience** profile.
3. Change the **Enable tamper protection to prevent Microsoft Defender being disabled** setting to either **Enable** or **Disable**.
4. Proceed to assign the profile to the devices you wish the tamper protection setting to apply:

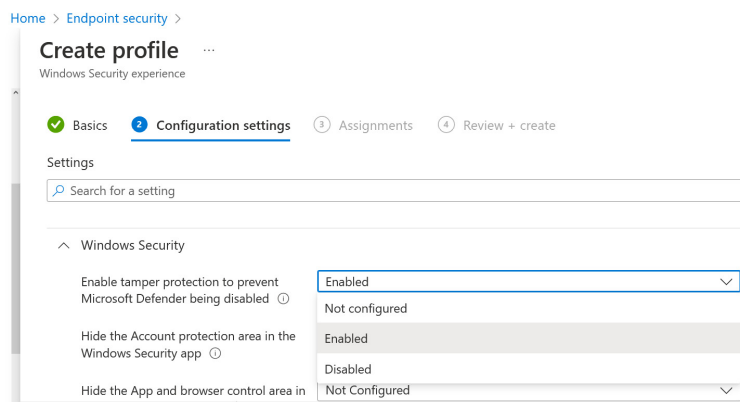


Figure 6.13 – Configuring tamper protection in Microsoft Intune

On clients, the tamper protection status can be reviewed using the following command:

```
Get-MpComputerStatus | FL IsTamperProtected,TamperProtectionSource
```

**IsTamperProtected** returns a binary **true** or **false** value and **TamperProtectionSource** lets you know where that is coming from, such as Microsoft 365 Defender (represented as **ATP**) or Intune.

As an administrator, you can also track the tamper protection status of onboarded devices using Microsoft 365 Defender. The recommendation is to **Turn on Tamper Protection** in the security recommendations ([security.microsoft.com/security-recommendations](https://security.microsoft.com/security-recommendations)). This will show you **exposed devices**, which are those that are not tamper-protected.

It is important to keep on top of having tamper protection enabled as widely as possible. As you've learned, it protects against several attack methods, such as disabling antivirus features. It is not a silver bullet against evasion, regrettably due to the need to still allow administrators to manage exceptions, but remains a fundamental tool for defenders.

This brings us to our section on diagnosing and investigating faults as part of ongoing MDAV management.

## Ongoing management of MDAV

In the following subsections, you'll learn about how to troubleshoot MDAV and work with its reports. This information will assist in your day-to-day operations, service desk issues, and ongoing security posture reviews.

### Troubleshooting

The reality of enterprise IT management is that we're going to run into problems sooner or later. It would be naïve to try and cram every possible problem you'll experience into this book, so in this section, we'll look at some guidance *specific* to MDAV, which you can use, as well as *general*

troubleshooting tools such as **Windows Performance Recorder** and **Process Monitor**.

## Troubleshooting mode

After enabling central policy and tamper protection, you may struggle to troubleshoot MDAV endpoints. For example, a user reports trouble updating a Microsoft 365 Apps for Enterprise add-in because of ASR rules (see [Chapter 7](#)); or after changing the cloud-delivered protection level to **High Plus**, you find that a developer cannot launch some of their apps. The problem? Your Intune, Group Policy, or Configuration Manager policy cannot be amended, and tamper protection stops you – by design – from changing MDAV.

This is where **troubleshooting mode** comes in. In the Microsoft 365 Defender portal page for a device, an administrator can choose to **Turn on troubleshooting mode**:

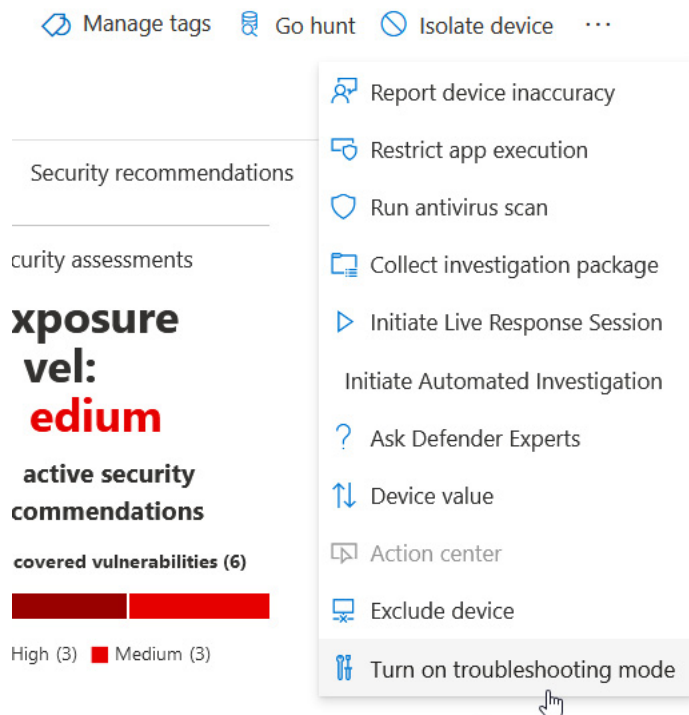


Figure 6.14 – Turning on troubleshooting mode in the Microsoft 365 Defender portal

What happens after activating troubleshooting mode? The following chain of events occurs, typically within 15 minutes of turning it on:

1. A snapshot of the pre-troubleshooting mode MDAV settings is recorded. The user is notified that troubleshooting mode has been enabled by an OS notification.
2. Then, with local admin rights, changes that tamper protection would normally block are allowed, excluding the ability to uninstall or completely disable MDAV – for example, disable tamper protection, then change exclusions and real-time protection using **Set-MpPreference**.
3. After 3 hours, troubleshooting mode expires and the user is notified once more. Before expiration, a warning notification is also displayed.

4. At expiration, a closing snapshot is taken and MDAV returns to its protected state. Any changes are set back to the original snapshot:

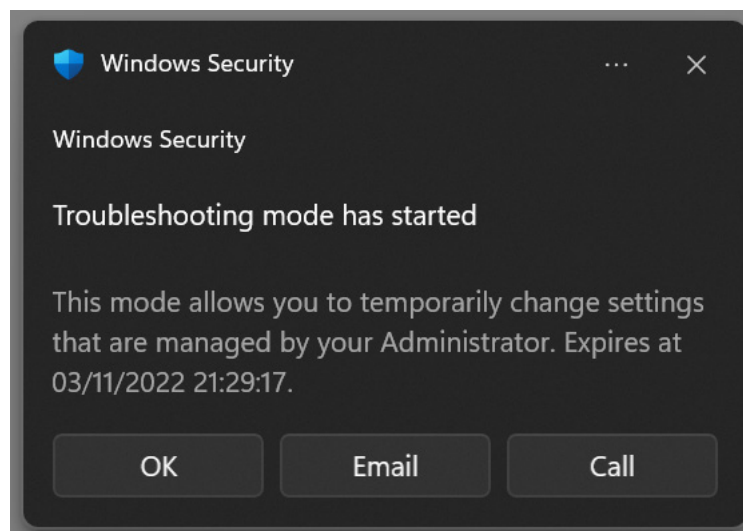


Figure 6.15 – Device notification for troubleshooting mode activation

During troubleshooting mode, events are logged and available from the device's **Timeline** tab in Microsoft 365 Defender, investigation packages, or with advanced hunting. This includes both entries for troubleshooting mode being enabled/expiring, and the activities conducted during it, such as adding exclusions. When hunting, the **DeviceEvents** table's **ActionType** can be filtered to **AntivirusTroubleshootModeEvent** for troubleshooting mode events. Advanced hunting will be covered in additional depth in [\*Chapter 19\*](#).

Enabling troubleshooting mode isn't without risk due to allowing configuration changes, and it also stops updates when active, but it is a necessary option for investigating problematic devices.

The biggest concern many have when deploying a new endpoint security capability is its performance implications. If you need to investigate performance problems, you can supplement troubleshooting mode with **Performance Analyzer**.

### Performance Analyzer

A common problem for any new protection software deployment is performance. Antimalware services, by their nature, have to intercept and monitor a lot of activity, which adds overhead. All going well, it's not particularly noticeable to end users. If things aren't working properly, though, you can expect sluggish performance caused by increased demands on the processor and memory.

Available for MDAV since the 4.18.2108 platform released in 2021, Performance Analyzer is a PowerShell capability that introduces two cmdlets: **New-MpPerformanceRecording** and **Get-MpPerformanceReport**. They are currently unavailable for Windows Server 2012 R2. The cmdlets are used back-to-back to create and review a report of MDAV's performance of a recreated scenario:

1. Either locally or on a remote session, with elevated rights, initiate troubleshooting by executing **New-MpPerformanceRecording** and using **-RecordTo** to specify where the output ETL file should be saved.
2. Reproduce the problem, such as by launching the program with degraded performance or performing actions that are degraded. After this, return to the PowerShell window and press the *Enter* key to complete the recording:

```
PS C:\WINDOWS\system32> New-MpPerformanceRecording -RecordTo C:\MDAVreport.etl
Starting Microsoft Defender Antivirus performance recording... ok.

Recording has started.

=> Reproduce the scenario that is impacting the performance on your device.

    Press <ENTER> to stop and save recording or <Ctrl-C> to cancel recording:

Stopping Microsoft Defender Antivirus performance recording...
ok.

Recording has been saved to 'C:\MDAVreport.etl'.

The performance analyzer provides insight into problematic files that could
cause performance degradation of Microsoft Defender Antivirus. This tool is
provided "AS IS", and is not intended to provide suggestions on exclusions.
Exclusions can reduce the level of protection on your endpoints. Exclusions,
```

Figure 6.16 – Running a performance recording with New-MpPerformanceRecording

3. The results can be interpreted with PowerShell by using **Get-MpPerformanceReport** pointed to the file using **-Path** (as shown in *Figure 6.17*). You will need to use at least one of the following to present the data:
  1. **-TopFiles**
  2. **-TopExtensions**
  3. **-TopProcesses**
  4. **-TopScans**

These correspond to the duration for which MDAV is scanning them, thus potentially lowering performance. More parameters are available and can be discovered using **Get-Help Get-MpPerformanceReport**. Each parameter should be followed with **:X**, where **X** is the number of items to return. You can also nest the results. For example, you can use the following command to see the top five processes and the top three consuming extensions of each:

```
Get-MpPerformanceReport -Path $file -TopProcesses:5 -TopExtensionsPerProcess:3
```

```
PS C:\WINDOWS\system32> Get-MpPerformanceReport -Path C:\MDAVreport.etl -TopProcesses:5 -TopExtensionsPerProcess:3

TopProcesses
=====
Count TotalDuration MinDuration AverageDuration MaxDuration MedianDuration ProcessPath
-----
34 4381.0666ms 2.4891ms 128.8549ms 2934.9266ms 23.0432ms

Extensions:
Count TotalDuration MinDuration AverageDuration MaxDuration MedianDuration Extension
-----
14 3677.5403ms 2.4891ms 262.6814ms 2934.9266ms 26.5506ms .dll
13 354.7271ms 13.6905ms 27.2867ms 71.3529ms 23.5879ms .exe
7 348.7992ms 3.5131ms 49.8284ms 317.3174ms 5.6856ms .xml

Count TotalDuration MinDuration AverageDuration MaxDuration MedianDuration ProcessPath
-----
1 394.8480ms 394.8480ms 394.8480ms 394.8480ms 394.8480ms C:\Program Files\Google\Chrome\Application\chrome.exe

Extensions:
Count TotalDuration MinDuration AverageDuration MaxDuration MedianDuration Extension
-----
1 394.8480ms 394.8480ms 394.8480ms 394.8480ms 394.8480ms .exe
```

Figure 6.17 – Using Get-MpPerformanceReport to see files processed by MDAV

Using the results of Performance Analyzer, you can identify the culprit files that are damaging performance. How you proceed depends on the situation. For example, a vendor may suggest adding exclusions. Ideally, you should avoid them, so consider using the data to raise support requests to implement fixes.

Another tool in your belt for troubleshooting is the age-old favorite of event logs, so let's check it out.

Logs

Don't you just love crawling through logs? MDAV stores logs under **Event Viewer | Applications and services logs | Microsoft | Windows | Windows Defender**.

There are too many event IDs to list in this book without boring you to sleep, but some examples are noted in the following table. These examples are mostly based on failures for further investigation, but also include attack surface reduction, which you will learn about in the next chapter. If you are ingesting events into a central log system, they might come in useful:

| Event ID               | Why It's Useful  |
|------------------------|--|
| 1005                   | Find out if an antimalware scan failed   |
| 1008,<br>1118,<br>1119 | MDAV failed to apply a remediation action  |
| 1010                   | Quarantine restoration failed; useful if initiated from the Microsoft 365 Defender portal but it doesn't appear to have worked |
| 1121                   | An ASR rule block action occurred  |
| 1122                   | An ASR audit mode action occurred  |
| 1123                   | A controlled folder access action was audited  |

|               |   |
|---------------|---|
| 1124          | A controlled folder access action was blocked   |
| 2001          | Updates for security intelligence failed  |
| 2003          | Updates for the engine failed   |
| 2006          | Updates for the platform failed   |
| 2004          | Definitions couldn't be loaded to MDAV attempted to load the last-known good set                                |
| 2005          | The engine couldn't load due to an unsupported platform version, so a last-known good restoration was attempted |
| 2040,<br>2041 | MDAV's support for this OS will end soon or has ended   |
| 5001          | Real-time protection has been disabled  |
| 5010          | Malware and PUA scanning has been disabled  |
| 5012          | Virus scanning has been disabled  |
| 5013          | Tamper protection prevented a change to a protected configuration   |

Table 6.1 – MDAV Event IDs

A reference guide is maintained on Microsoft Docs for other important event IDs, available at [learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/troubleshoot-microsoft-defender-antivirus](https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/troubleshoot-microsoft-defender-antivirus).

Per-device troubleshooting is useful for specific problem scenarios, but you're also going to want to keep on top of your devices at scale, checking for problems before you receive a helpdesk ticket. There are several reports available to help with this out of the box, which we'll explore next.

## Understanding reports

For a glancing view with the ability to drill into the details if required, there are several reports you can find across your Microsoft 365 Defender management tools. In this section, we will check out some of them.

*BUT WAIT – THERE'S MORE!*

*The reports in this section aren't your only option for investigating devices and surfacing potential problems. Throughout this book, you'll find references to advanced hunting queries that can help you out. You can also integrate Microsoft 365 Defender APIs with Power BI for bespoke reporting.*



## Device health and compliance report

Found in **Microsoft 365 Defender** | **Reports** | **Device health and compliance**, this is a report for getting a quick overview of the status of your onboarded devices. At a glance, you can see trends over the length of your MDE data retention period for the following:

- Device health, such as impaired communications
- Antivirus status, such as being disabled or out of date
- OS platforms in use
- Versions of Windows in use

In practical terms, this can be useful for identifying unexpected trends, such as a sudden decline in updated devices (is there a network problem?) or sensor state (has a configuration change broken something?):

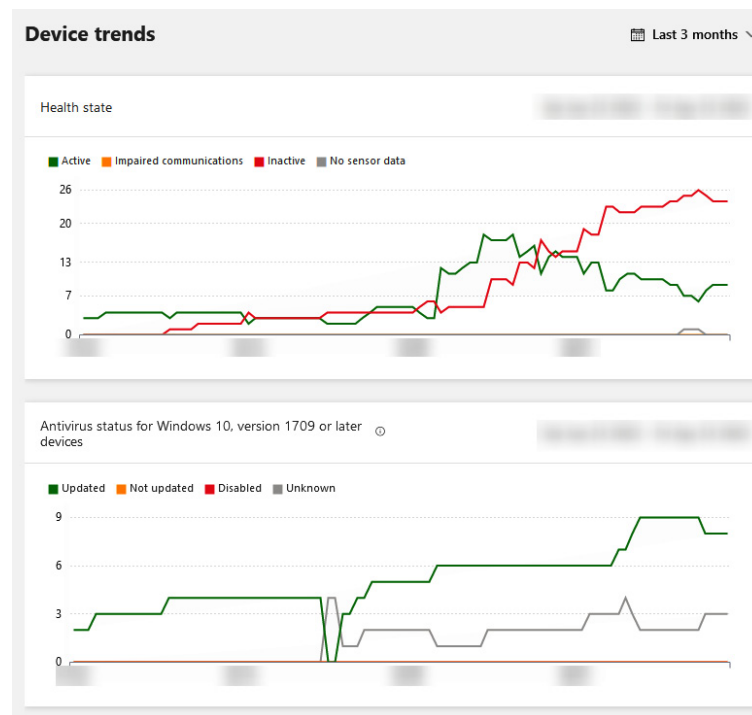


Figure 6.18 – Device trends as seen in the report

## Intune reports

For Intune devices, Intune's **Endpoint security** page has three reports.

The **Summary** page, which is also where new MDAV policies are made, reports metrics of endpoints, such as the number of pending updates, scans, restarts, or critical failures and inactive agents.

The **Unhealthy endpoints** and **Active malware** pages present a list of devices that should have some attention for those respective reasons.

Within these tabs, an Intune administrator may proceed to act, such as initiating scans, restarts, or updates. This is done using the toolbar at the top of the device list:

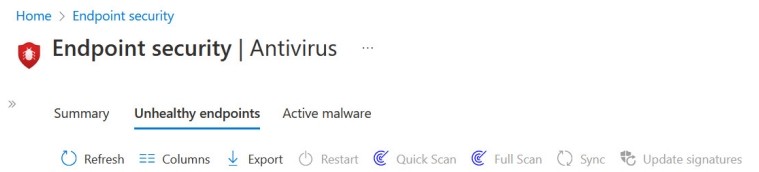


Figure 6.19 – Intune actions available for devices

You can also use the **Reports | Microsoft Defender Antivirus** and **Firewall** reports found in the Microsoft Intune admin center. In the case of MDAV, you'll get a **Detected malware** report and an **Antivirus agent status** report. You can drill into these and use the **Generate report** button to get data by device. For example, the antivirus agent status reports return information on the device's anti-malware, signature, and engine version, whether or not a scan is in progress, tamper protection status, real-time protection status, and more:

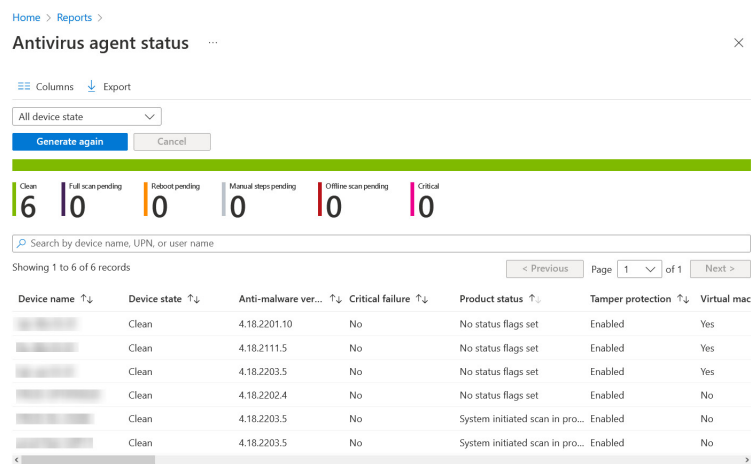


Figure 6.20 – Intune's Antivirus agent status report

The reports you've just read about can be combined with the capabilities you'll learn about in [Chapter 17](#) and [Chapter 19](#) to stay on top of your endpoint's security posture, ensuring they're behaving as expected.

## Summary

This chapter expanded on many of the advanced and cloud-powered capabilities of Microsoft Defender Antivirus. You learned how cloud-delivered protection drastically improves the security of the OS, and how it facilitates things such as BAFS and EDR in block mode. You also learned how to fight against evasive action using tamper protection, to control unauthorized changes to MDAV.

In the next chapter, your understanding of MDAV will continue to grow as we explore ASR to minimize risk as early in the attack chain as possible.

## Questions

To test your knowledge of protecting Windows clients and servers with Microsoft Defender for Endpoint, you can try answering the following

questions. The answers can be found toward the end of this book:

1. A developer is generating test versions of their new application and reports it is not launching successfully. You run **Get-MpPreference** and discover that its **CloudBlockLevel** is **6**. What does this mean?
  1. Block at first sight is in zero-tolerance mode
  2. Block at first sight is in high plus mode
  3. Block at first sight is in its default configuration
2. A bad actor has local administrative rights to a Windows 11 device and is trying to evade defenses using PowerShell. You have enabled tamper protection on the device using Intune. Which of the following can the attacker not disable or evade? Choose all that apply.
  1. Real-time protection
  2. Cloud-delivered protection
  3. Default action based on threat ID
  4. Attack surface reduction rules
3. You are migrating Windows Server 2016 from a well-known third-party antivirus provider to Microsoft Defender Antivirus and Microsoft Defender for Endpoint. Of the following, which is false?
  1. You can enable EDR in block mode before the migration to benefit from MDE before the migration
  2. You can deploy the unified agent and Microsoft Defender Antivirus will enter passive mode automatically
  3. You can deploy the unified agent and Microsoft Defender Antivirus passive mode must be configured manually
  4. Attack surface reduction rules will be unavailable until Microsoft Defender Antivirus is in active mode
4. To minimize risk and disruption, which of the following PUA protection modes might be appropriate during an initial deployment?
  1. PUA Protection off
  2. PUA Protection on
  3. Audit mode
5. True or false: you can confirm the status of EDR in block mode on clients using **Get-MpPreference**.
  1. True
  2. False

## Further reading

To go into even further detail about some of the topics in this chapter, you can refer to the following online material:

- Fabian Bader's research into tamper protection was mentioned in this chapter, and you can find that blog and an incredible level of depth into tamper protection here: [cloudbrothers.info/en/current-limits-defender-av-tamper-protection](https://cloudbrothers.info/en/current-limits-defender-av-tamper-protection)
- Microsoft's official blog has some additional details on how cloud-delivered protection comes together to benefit MDAV: [microsoft.com/security/blog/2019/06/24/inside-out-get-to-know-the-advanced-technologies-at-the-core-of-microsoft-defender-atp-next-generation-protection](https://microsoft.com/security/blog/2019/06/24/inside-out-get-to-know-the-advanced-technologies-at-the-core-of-microsoft-defender-atp-next-generation-protection)

- This article, published in the Microsoft TechCommunity, provides an interesting real-life example of how EDR in block mode defends against threats: [techcommunity.microsoft.com/t5/microsoft-defender-for-endpoint/introducing-edr-in-block-mode-stopping-attacks-in-their-tracks/ba-p/1596617](https://techcommunity.microsoft.com/t5/microsoft-defender-for-endpoint/introducing-edr-in-block-mode-stopping-attacks-in-their-tracks/ba-p/1596617)