/

Managing Attack Surface Reduction for Windows

Attack surface reduction (**ASR**) refers to a group of capabilities that (wait for it!) reduce the attack surface of your devices by limiting their known areas of weakness. ASR first made its way to Windows 10 with feature update 1709, branded as **Exploit Guard**. You will still see this term referenced in some UIs and literature. In general, the term ASR has superseded it.

In this chapter, we will cover ASR capabilities for Windows that MDE customers have available:

- ASR rules
- Controlled folder access
- Exploit protection
- Network protection, including SmartScreen and web protection

Combined, ASR capabilities minimize the risk your device faces against threats such as zero days, exploits, and unauthorized activity. As before, you will learn how to configure and deploy these in the context of Windows in the enterprise, using central management tools and monitoring.

Our exploration of ASR begins with the most notable and widely deployed of them: ASR rules.

Understanding ASR rules

ASR rules restrict system behaviors often used by attackers, whether the intent is malicious or not.

By taking the determination of intent out of the equation, you significantly harden the device, albeit with the potential for disruption if legacy activities are still performed. Fortunately, you can plan for that disruption by deploying ASR rules in **Audit mode** (**2**) to review the scale of the problem before applying the rules in **Block mode** (**1**) or **Warn mode** (**6**). Warn mode, available for most but not all ASR rules since Windows 10 1809, allows the user to override the block for 24 hours at a time.

As general guidance, these three modes for ASR rules combined make a deployment road map:

- Start in **Audit** mode, leveraging the data that clients produce to understand what problems may present themselves when enabled
- After mitigating problems identified in **Audit** mode, or accepting the risks, proceed to **Warn** mode so that users can proceed without break-

ing their workflows
- When you are comfortable taking the next step, move your rules into full **Block** mode

You deploy ASR rules centrally using Intune, Group Policy, or Configuration Manager, which means your roadmap doesn't have to be all-or-nothing and can be scoped to Azure AD groups, organizational units, or collections. Instead of taking a big bang for each mode across your whole estate, roll out ASR rules in rings. Starting with pilot devices, progress out gradually as their effect becomes understood and managed.

During the rollout, you may find that some business processes are simply incompatible with ASR rules. You can exclude folders and files from ASR rules, including those with wildcards and variables. When you add an exclusion, it applies to *all* ASR rules. You cannot specify an individual ASR rule for the exclusion. These exclusions are managed separately from other MDAV exclusions and are covered in the *Deploying ASR rules* section.

ASR rules require MDAV to be in **Active** mode with **Cloud-delivered protection** enabled. On Windows 10, feature update 1709 or later is required as a baseline, and they are also supported on Windows Server 2012 R2/2016 (with the unified agent installed) or later. Some ASR rules require a later version of Windows 10, and you'll see a note of those when we move on to list them. Strictly speaking, ASR rules do not need an MDE license, but the value of MDE in the context of ASR rules is the visibility of their audits and blocks. If you're using Configuration Manager, the newer the version, the more ASR rules are available as they have been added over time; older versions may be missing some of the new options.

One of the great things about ASR rules is they *do what it says on the tin*. Each ASR rule has a descriptive name and a **GUID**. This GUID is referenced when deploying them with Group Policy or managing and viewing them with PowerShell. You should not need to know the GUID to manage them with Intune or Configuration Manager, which will show you the descriptive name.

ASR rules overview

Now that you have a solid background on ASR rules, let's get into each one and look at the intricacies you may need to know. ASR rules can be grouped into two categories: **standard protection rules** and **other rules**. Standard protection rules are ones Microsoft recommends are enabled by all customers: they are low risk and high reward. Other rules pose a higher risk of disruption, so you'll learn about managing that risk in the *Deploying ASR rules* and *Monitoring ASR rules* sections.

Standard protection rules

Here are the standard protection rules:

- **Block credential stealing from the Windows local security authority subsystem (lsass.exe)** (`9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2`)

on Windows 10 requires at least feature update 1803. With tools such as Mimikatz, LSASS can be exploited to retrieve the secrets it stores. You can implement this rule in **block** mode in scenarios that cannot use Windows Defender Credential Guard, which is another method of protecting LSASS. This rule can be quite noisy when looking at logs, but an entry does not necessarily mean there's a problem.

- **Block abuse of exploited vulnerable signed drivers** (`56a863a9-875e-4185-98a7-b882c64b5ce5`) leverages Microsoft-managed information on drivers that could be exploited to mitigate against those threats. The vulnerable drives come in part, but not entirely, from submissions made to the **Vulnerable and Malicious Driver Reporting Center** at **microsoft.com/en-us/wdsi/driversubmission**. This rule isn't currently available in Configuration Manager.
- **Block persistence through WMI event subscription** (`e6db77e5-3df2-4cf1-b95a-636979351e5b`) on Windows 10 requires at least feature update 1903. This ASR rule does not support **warn** mode or exclusions and can only be configured locally (PowerShell) or with Group Policy. WMI event subscription is a popular persistence mechanism used by attackers, so unless you have a user for it, such as management tools, you should consider this rule.

Other rules

These are the other rules:

- **Block Adobe Reader from creating child processes** (`7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c`) on Windows 10 requires at least feature update 1809. This ASR rule isn't available for deployment with Configuration Manager. Phishing emails often include PDF files that go on to create child processes deploying malware, which this ASR rule mitigates.
- **Block all Office applications from creating child processes** (`d4f940ab-401b-4efc-aadc-ad5f3c50688a`). Just like PDFs in the previous rule, Office documents are commonly used in phishing attacks; macros are an infamous tool for attackers to start malicious programs. For Access, Excel, OneNote, PowerPoint, and Word, this ASR rule protects against that kind of exploit.
- **Block executable content from email client and webmail** (`be9ba2d9-53ea-4cdc-84e5-9b1eeee46550`) applies to the Outlook desktop app and even some webmails such as `Outlook.com`. This controls the execution of files such as EXEs, PowerShell, and other scripts.
- **Block executable files from running unless they meet a prevalence, age, or trusted list criterion** (`01443614-cd74-433a-b99e-2ecdc07bfc25`) on Windows 10 requires at least feature update 1803. Using telemetry in the cloud-delivered protection service, this rule is useful when application control is not present to restrict the execution of files that do not meet reputation and prevalence checks.
- **Block execution of potentially obfuscated scripts** (`5beb7efe-fd9a-4556-801d-275e5ffc04cc`) applies to JavaScript, VBScript, PowerShell, and macros to limit their execution if the author has implemented obfuscation techniques to mask the true intent of the script.

- **Block JavaScript or VBScript from launching downloaded executable content** (`d3e037e1-3eb8-44c8-a917-57927947596d`) is one of the ASR rules that does not support **warn** mode. Scripts might be trying to download other executables that go on to deliver malware, and this ASR rule can defend you from that.
- **Block Office applications from creating executable content** (`3b576869-a4ec-4529-8536-b80a7769e899`) returns us to Office as a launch pad for attackers. For example, macros can be configured to execute other processes such as PowerShell scripts. This can be quite a challenging rule to deploy in environments that allow macros (most of them!) because users and developers may have automation processes that leverage this capability.
- **Block Office applications from injecting code into other processes** (`75668c1f-73b5-4cf0-bb93-3ecf5cb7cc84`) is generally simpler to roll out because injecting code can rarely be justified or used legitimately. Enabling this ASR rule can mitigate against malware that attempts to disguise itself as a safe process.
- **Block Office communication application from creating child processes** (`26190899-1602-49e8-8b27-eb1d0a1ce869`) applies to Outlook and Outlook on the web to prevent child processes being launched but does have logic to allow legitimate cases such as opening attachments.
- **Block process creations originating from PSExec and WMI commands** (`d1e49aac-8f56-4280-b9ba-993a6d77406c`) on Windows 10 requires at least feature update 1803. Although legitimately used by management tools and administrators for remote code execution and device management, there are many cases of APTs using PSExec and WMI in their attacks. Given its dependence on WMI, you can't deploy this rule with Configuration Manager.
- **Block untrusted and unsigned processes that run from USB** (`b2b3f03d-6a65-4f7b-a9c7-1c7ef74a9ba4`) falls firmly in the *does what it says on the tin* camp. Environments with a mature security posture will be limiting the use of USB drives anyway, but using this rule as well can further harden devices against the threat of malware, which is rarely signed and likely low in prevalence before having a signature.
- **Block Win32 API calls from Office macros** (`92e97fa1-2edf-4476-bdd6-9dd0b4dddc7b`) is usually a low-noise ASR rule to enable. Although macros might be prevalent, they shouldn't be making Win32 API calls.
- **Use advanced protection against ransomware** (`c1db55ab-c21a-4637-bb3f-a12568109d35`) on Windows 10 requires at least feature update 1803 and does not support **warn** mode. This rule's name is vaguer than the others. It uses heuristics to conclude if something could potentially be ransomware. To minimize false positives, it does not apply blocks to files that are signed, prevalent in the Microsoft telemetry dataset, or known to be safe.

There is a variance in ASR rules regarding the generation of alerts in Microsoft 365 Defender or toast notifications to the user (if enabled). For any alerts to come to Microsoft 365 Defender, cloud-delivered protection (also known as the file blocking level) must be at least **High**. This was covered in the *Block at first sight* section in the previous chapter. For a ma-

trix of when alerts or user notifications are generated, refer to Microsoft's documentation at

learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/attack-surface-reduction-rules-reference?#per-rule-alert-and-notification-details.

Deploying ASR rules

Consistent with recommendations throughout this book, if you can use Intune, then do so. However, Group Policy and Configuration Manager are also available. PowerShell can be used on individual devices but doesn't scale well, so it isn't covered in this section.

Intune

ASR rules are best managed using Intune by creating an **Attack Surface Reduction rules** profile via **Endpoint security** | **Attack surface reduction** | + **Create Policy** | platform of **Windows 10 and later** | profile of **Attack Surface Reduction Rules**.

Each ASR rule is listed, with a drop-down of the available options. You will also find the ability to create exceptions. If there are multiple policies assigned to a device, they are merged to create the final applicable list of ASR rule settings. This does not include settings between policies that clash, which are ignored entirely and do not apply, which leaves a gap in your protection and must be avoided. This is relevant in scenarios where **security baselines** in Intune have been deployed, or the legacy **Endpoint protection** profile has been used:

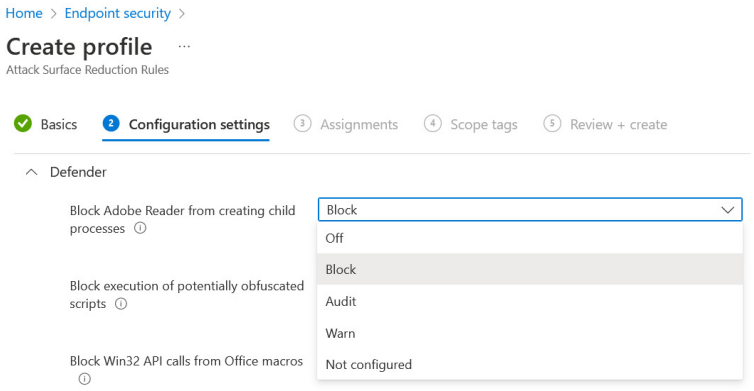

Figure 7.1 – Configuring ASR rules in Intune

Configuration Manager

If you're managing ASR rules with Configuration Manager, follow these steps:

1. In the console, navigate to **Assets and Compliance** | **Endpoint Protection** | **Exploit Guard** | **<your new or existing policy>**.
2. Each policy can have several Exploit Guard components. Select, at a minimum, **Attack Surface Reduction**. You will learn more about the others in this chapter:

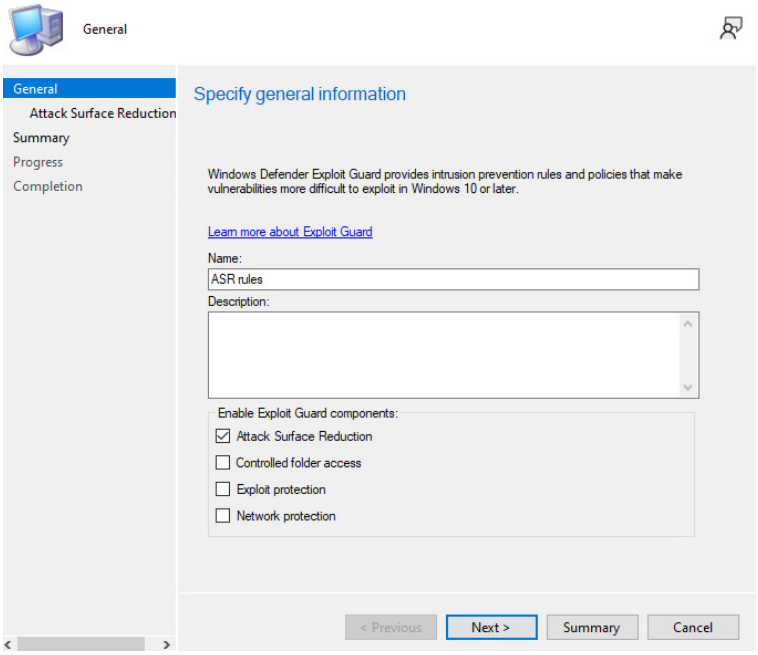

Figure 7.2 – Creating an Exploit Guard policy in Configuration Manager for ASR rules

3. Configuration is simple and, like Intune, each ASR rule is listed and there's a drop-down for options. You can also apply exclusions. As explained in the *ASR rules overview* section, you should have the latest version of Configuration Manager to get as many rules as possible, and not all rules are available in Configuration Manager.
4. Proceed to configure the policy and assign it to the required collections.

Group Policy

Group Policy objects can deploy ASR rules within **Computer Configuration** | **Policies** | **Administrative Templates** | **Windows Components** | **Microsoft Defender Antivirus** | **Microsoft Defender Exploit Guard** | **Attack Surface Reduction**.

When the **Configure Attack Surface Reduction rules** setting is enabled, you must also configure a list of **value names** (ASR rule GUIDs) and **values** (enforcement mode). These GUIDs can be found in the *ASR rules overview* section, and the values available (if supported for the rule) are as follows:

- **0**: Off
- **1**: Block
- **2**: Audit
- **5**: Not configured
- **6**: Warn:

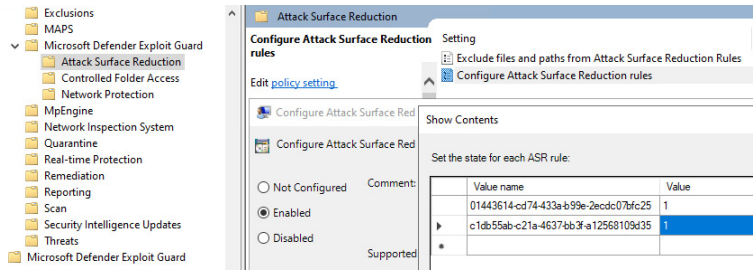

Figure 7.3 – Setting ASR rules in a Group Policy Object

Exclusions are configured by GPO with the **Exclude files and paths from Attack Surface Reduction Rules** setting. This uses a similar list of values, where each **Value name** would be a file or folder path. You must also enter **0** for **Value** for each value name in exclusions.

Monitoring ASR rules

As you go through the stages of **Audit**, **Warn**, and **block** mode for ASR rules, you'll want to keep up to date with the data to see the consequences of each.

A good starting point is **Microsoft 365 Defender** | **Reports** | **Attack surface reduction rules**. This is broken down into three sections: **Detections**, **Configuration**, and **Add exclusions**:

- The **Detections** tab reports ASR rule events, with filter options to view specific rules, and grouping options to choose how it is represented. For example, you can view line-level detail of ASR rules by application, rule, user, and so on. This is particularly useful when you're deciding when modes are ready to go from audit to enforced, and what you may need to plan for:

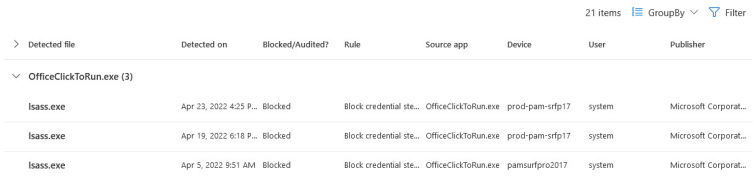

Figure 7.4 – ASR events as viewed in the Microsoft 365 Defender portal

- The **Configuration** tab presents each onboarded device and information on their ASR configuration – for example, which rules are in block or audit mode. This is useful for troubleshooting or identifying gaps in protection:

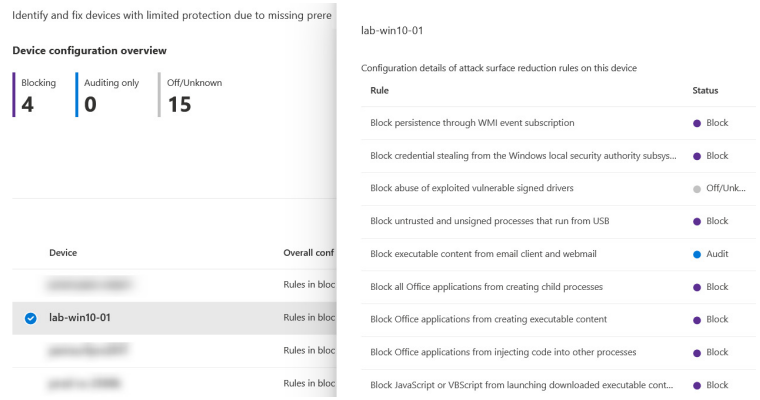

Figure 7.5 – ASR configurations by device as viewed in the Microsoft 365 Defender portal

- Lastly, the **Add exclusions** page uses the telemetry gathered for the **Detections** page to recommend what files might need exclusions. With any exclusions, you want to proceed with caution. For example, LSASS will likely be present here, and you don't want to exclude that. When you click on a filename, a popup will appear with options for you to download a CSV of exclusion paths (**Get exclusion paths**) or a shortcut to Microsoft Intune to use it (**Add exclusions**).

In addition to the dedicated ASR rule reports of Microsoft 365 Defender, you can refer to some event IDs in the *Troubleshooting* section of ***Chapter 5***.

Advanced hunting also surfaces ASR events (limited to the first per hour for a device). Each **`ActionType`** for ASR rules begins with **`Asr`**. As an example of reviewing them, Microsoft provides the following query to see the total of each event:

```
DeviceEvents
| where Timestamp | ago(30d)
| where ActionType startswith "Asr"
| summarize EventCount=count() by ActionType
```

Chapter 18, *Advanced Hunting with KQL*, covers advanced hunting with KQL in additional detail so that you can build queries for ASR rules.

There are ASR measures available for Windows devices beyond ASR rules. In the next section, we'll review another hardening capability closely linked to ASR rules.

Controlled folder access

Primarily a defense against ransomware, **controlled folder access** (**CFA**) is another ASR capability. It works by limiting folder write access to allow-listed applications only. If an app isn't trusted, it can't modify or delete files in the controlled folders.

Trusted apps are a combination of the ones you specify, and the ones deemed prevalent in Microsoft's massive telemetry data. Any other apps are forbidden from editing the contents of the folders. Thanks to the vastness of Microsoft's reputation system, you may not even have to add custom apps. Regardless of the applications you choose to trust, the system will not trust script engines such as PowerShell, even if you add them as exclusions.

The folders are a combination of the ones you specify, and the ones listed by Microsoft by default (public and user profile **Documents**, **Pictures**, **Videos**, **Music**, and **Favorites**; including OneDrive redirected versions).

As with ASR rules, there is a dependency on MDAV in **active** mode, and the same OSs are supported.

As you configure CFA, you will see **audit-only** mode as an option, and two additional modes: **block disk modification only** and **audit disk modification only**. These modes expand the standard **block** and **audit** modes to also include writes to disk sectors (for example, `\Device\HarddiskVolume1`).

Deploying CFA

Enabling controlled folder access, in either auditing or enforcing mode, can be achieved centrally using Intune, Configuration Manager, or Group Policy. Wildcards are supported for allowed applications, but not allowed folders. UNC paths and mapped drives are supported, as are environmental variables. For example, a commonly allowed app is `%userprofile%\AppData\Local\Microsoft\OneDrive\OneDrive.exe`. When you deploy a policy to allow an app with one of these management tools, it won't take effect until that app is restarted.

Intune

For Intune devices, CFA is managed using the same profile type as ASR rules, which we covered in the previous section. This means it does not yet support the security management scenario.

Head to **Endpoint security** | **Attack surface reduction** | + **Create Policy** (or choose an existing policy) | platform of **Windows 10 and later** | profile of **Attack Surface Reduction Rules**.

In the ASR rules profile, you manage CFA with three settings:

- **Enable Controlled Folder Access**, which gives you the option to choose the enforcement mode, such as audit or enabled
- **Controlled Folder Access Protected Folders**, which is your list of additional folders; system folders mentioned earlier don't need to be added manually and cannot be excluded
- **Controlled Folder Access Allowed Applications**, which is your list of additional applications to have permitted access:

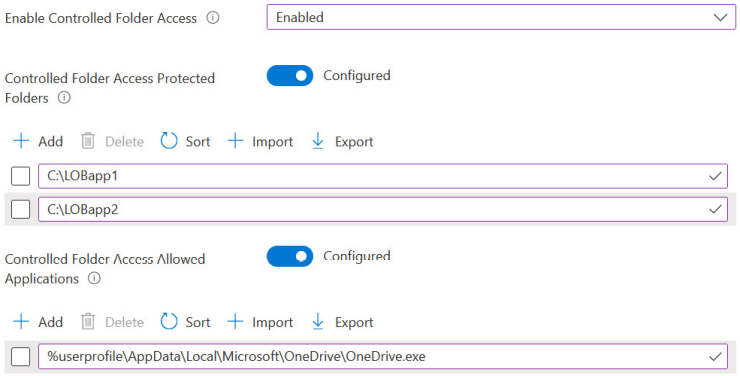

Figure 7.6 – Configuring controlled folder access with Intune

Configuration Manager

Configuration Manager administrators manage CFA in the same type of Exploit Guard policy used for ASR rules:

1. Navigate to **Assets and Compliance** | **Endpoint Protection** | **Exploit Guard** | <**your new or existing policy**> in the console.
2. On the **General** page, ensure **Controlled Folder Access** is selected as a component.
3. Clicking through the wizard, on the **Configure Controlled folder access** page, you can choose the enforcement mode, allowed apps, and protected folders:

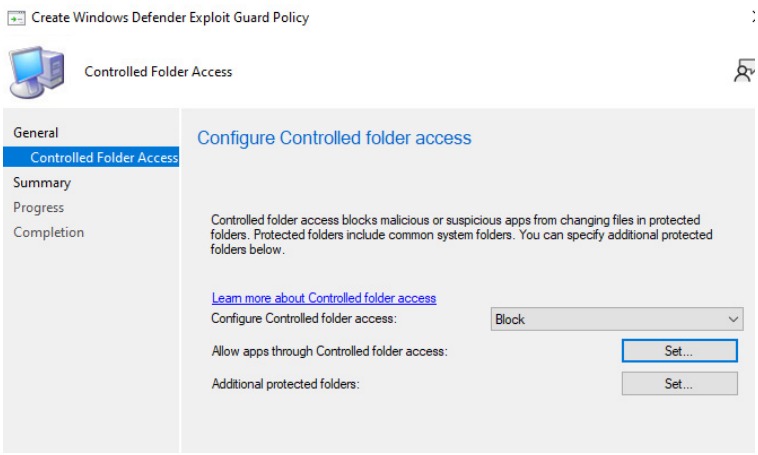

Figure 7.7 – Configuring controlled folder access with Configuration Manager

4. Proceed to finish the wizard and deploy the policy.

Group Policy

The last method of centrally deploying CFA is Group Policy. In a GPO, settings for CFA can be found in **Computer Configuration** | **Policies** | **Administrative Templates** | **Windows Components** | **Microsoft Defender Antivirus** | **Microsoft Defender Exploit Guard** | **Controlled Folder Access**. You will find three settings that are the same as those discussed in the other methods:

- **Configure allowed applications**, where you list apps you explicitly want to allow

- **Configure Controlled folder access**, which specifies the enforcement mode
- **Configure protected folders**, where you list the folders you explicitly want to protect (system-defined folders do not need to be re-specified here)

Monitoring CFA

Audits are useful before deployment so that you can identify apps that should be added to allow lists. Log information is also retained during enforcement after you progress from **audit** mode to **block** mode.

Data can be accessed using the event IDs mentioned in the *Troubleshooting* section of ***Chapter 5***, or via advanced hunting. In your advanced hunting queries, reference `ControlledFolderAccessViolationAudited` and `ControlledFolderAccessViolationBlocked` for `ActionType` to see events and understand what applications should be trusted:

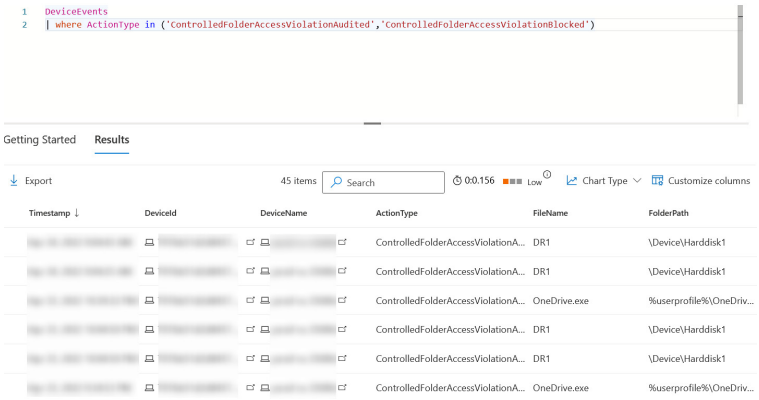

Figure 7.8 – Using advanced hunting to track controlled folder access audit events

Controlled folder access is a powerful tool but should be deployed gradually after you have used audit events to confirm any production impact. Next on our ASR list is **exploit protection**, which, unlike CFA, is enabled by default.

Exploit protection

Exploit protection succeeded the **Enhanced Mitigation Experience Toolkit (EMET)** from Windows 10 1709 onwards as a collection of mitigations against potential OS and app exploits. Exploit protection includes mitigations such as **Data Execution Prevention (DEP)**, **block untrusted fonts** and **remote images**, and **code integrity guard**.

You can also configure **system settings** and **program settings**. System settings apply across the operating system, while program settings are scoped to specific executables. By default, exploit protection is already turned on system-wide for system settings except for **Force randomization for images (Mandatory ASLR)**. Each system setting can be overridden at the executable level to work around problems they may cause.

Exploit protection has many protections enabled by out-of-the-box settings, but you can customize it to address specific concerns. To reduce the risk if you do want to make changes, exploit protection can be evaluated by using **audit** mode, just like with other ASR features. A list of exploit protection settings and their types can be found in *Table 7.1*:

Exploit Protection Setting	Setting Type
Control flow guard (CFG)	System and app
Data Execution Prevention (DEP)	System and app
Force randomization for images (Mandatory ASLR)	System and app
Randomize memory allocations (Bottom-Up ASLR)	System and app
Validate exception chains (SEHOP)	System and app
Validate heap integrity	System and app
Arbitrary code guard (ACG)	App only
Block low-integrity images	App only
Block remote images	App only
Block untrusted fonts	App only
Code integrity guard	App only
Disable extension points	App only
Disable Win32k system calls	App only
Do not allow child processes	App only
Export address filtering (EAF)	App only
Import address filtering (IAF)	App only
Simulate execution (SimExec)	App only
Validate API invocation (CallerCheck)	App only
Validate handle usage	App only

Exploit Protection Setting	Setting Type
Validate image dependency integrity	App only
Validate stack integrity (StackPivot)	App only

Table 7.1 – Exploit protection mitigations by type

EXPLOIT PROTECTION INTERNALS

For a full and detailed reference of how each exploit protection setting functions, check out the reference guide at **learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/exploit-protection-reference**.

Deploying exploit protection

Configuration is managed with XML files. On a reference device, you can configure exploit protection to its required state from **Windows Security** | **App & browser control** | **Exploit protection** or by using `Set-ProcessMitigation` in PowerShell. As you'd expect, using the Windows Security GUI is a bit easier. You will have options for **Audit** mode if they're available, and some settings have additional relevant options too:

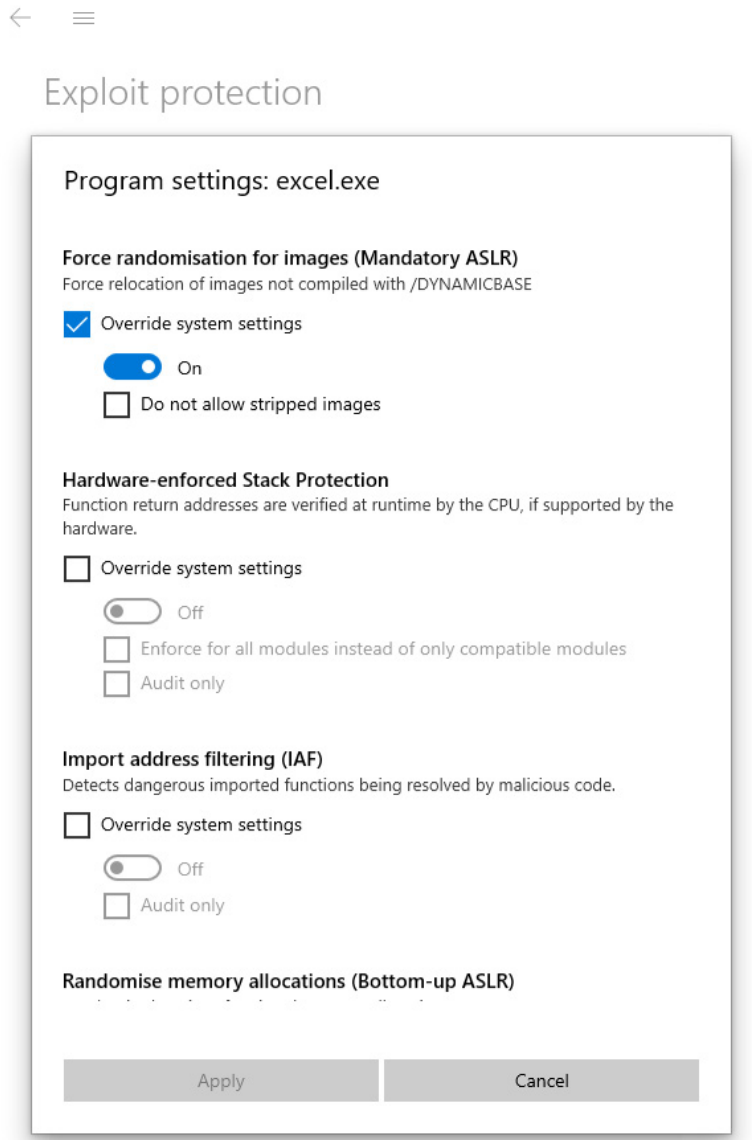

Figure 7.9 – Configuring exploit protection for excel.exe using Windows Security

Using a management tool such as Intune or Group Policy, you can distribute the XML to devices, and they take place when the apps in scope restart. The XML file can be extracted from the reference device using the **Export settings** button in Windows Security or by using `Get-ProcessMitigation -RegistryConfigFilePath Export.xml`, where `Export.xml` is the XML file you create.

Intune

To configure exploit protection with Intune, navigate to **Endpoint security** | **Attack surface reduction** | + **Create policy** and choose **Exploit Protection** as the profile type. Proceeding to the **Configuration settings** page, this is where you upload the XML file and disable local admin merge with **Disallow Exploit Protection Override** set to **Enable**:

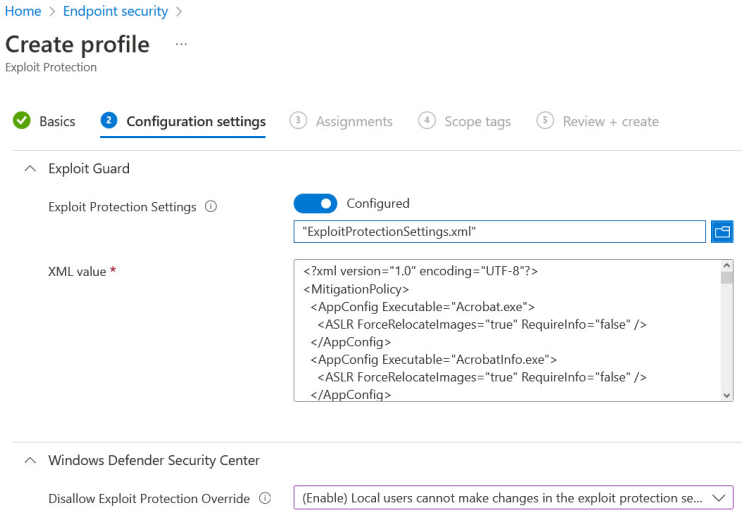

Figure 7.10 – Creating an exploit protection profile in Intune

Configuration Manager

Configuration Manager remains an option for deploying exploit protection settings. In the console, head to **Assets and Compliance** | **Endpoint Protection** | **Exploit Guard** | **<your new or existing policy>**. Within the components to choose from, ensure that **Exploit protection** is selected and specify your XML file:

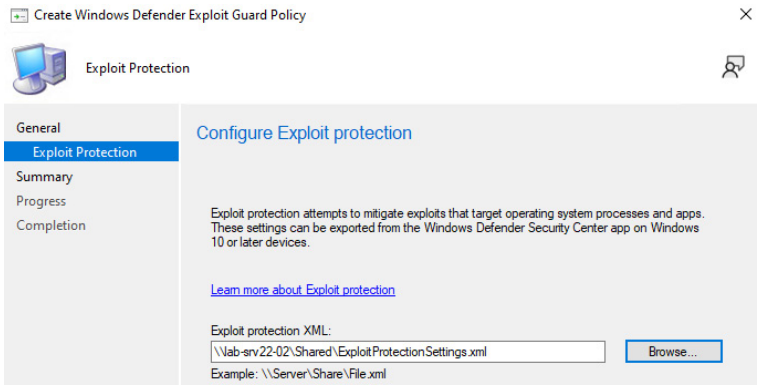

Figure 7.11 – Creating an exploit protection policy with Configuration Manager

Group Policy

If you're using Group Policy, point your GPO toward **Computer configuration** | **Policies** | **Administrative templates** | **Microsoft Defender Exploit Guard** | **Exploit Protection**. Set **Use a common set of exploit protection** settings to **Enabled** and specify a path to the XML file. A UNC path should be accessible to the clients; if you're specifying a local path, you need to get that onto the device in that same path – it isn't uploaded to the GPO and distributed:

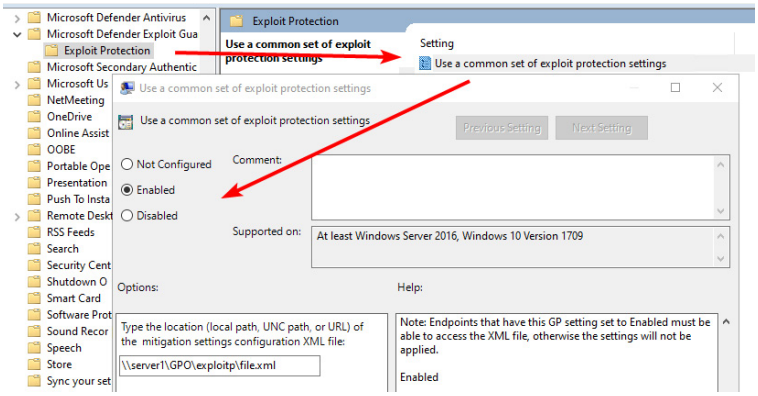

Figure 7.12 – Creating an exploit protection GPO

Monitoring exploit protection

Whether in audit or enforcement mode, you can use advanced hunting in Microsoft 365 Defender to monitor exploit protection using the following Microsoft-provided example:

```
DeviceEvents
| where ActionType startswith 'ExploitGuard' and ActionType !contains 'NetworkProtection'
```

This brings us to the end of this section on exploit protection and ASR. As mentioned consistently in this chapter, you don't have to rush into any of this: use the audits as your friends so that you minimize surprises. Roll out enforcement with rings and keep on top of the logs to track the repercussions of it.

The next section continues the MDE Windows security story with three features closely related to ASR, using the same principles of reputation-based and layered protection.

ASR at the network layer

In this section, you'll learn about **Microsoft Defender SmartScreen**, the closely related **network protection** (which is the last of our ASR features to discuss), and **web protection**.

SmartScreen

Available to both consumers and MDE customers, SmartScreen protects risky websites and applications before Microsoft Defender Antivirus needs to step in. Using a combination of suspicious indicators, user reports, and popularity telemetry, SmartScreen can either warn or block access to websites and applications it identifies as potentially malicious. For example, SmartScreen can identify unsafe advertising frames in websites and prevent them from loading. Or, if a user downloads an application with a low or poor reputation, it can prevent it from executing.

SmartScreen's scope is limited to content that originates from the internet. For example, it can block the execution of a rarely seen application

from a download website, but if this file was copied to the device by an administrator's USB storage and executed, it would fall out of scope. In this regard, SmartScreen is a very simple first line of defense against end user-derived threats, rather than a bad actor who has hands-on-device access. App protection requires feature update 1703 for Windows 10 at a minimum.

For website protection, SmartScreen is available natively in Edge and is available even if Microsoft Defender Antivirus is in passive mode. Google Chrome also has an official extension for SmartScreen that is branded in the Chrome web store as **Microsoft Defender Browser Protection**. This too functions in passive mode; MDAV is not a prerequisite:

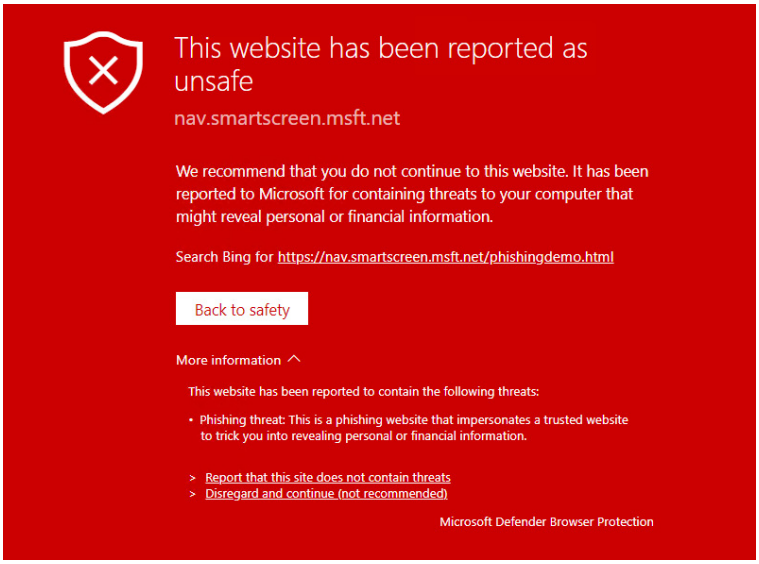

Figure 7.13 – SmartScreen protecting access to an unsafe website

The preceding ominous red warning screen will be familiar to most and shows SmartScreen in action.

Those of you who are eagle-eyed may have noticed that the SmartScreen screenshot includes the option to **Disregard and continue (not recommended)**. For both websites and downloaded apps, this is SmartScreen's default stance: warn the user, but allow them to proceed, albeit after jumping through a few warnings and confirmation clicks. As a defender, you're going to want to remove that ability for users.

Now, let's look at some ways to centrally manage SmartScreen.

Intune

Unlike MDAV and other Windows 10 security settings, there is no dedicated page in the Microsoft Intune admin center's endpoint security section to manage SmartScreen, other than the legacy Edge browser. This means we are limited to Intune-enrolled devices. The way this book encourages you to manage SmartScreen with Intune is via the **Settings catalog** area, which is Microsoft's attempt to start having an all-in-one location for any MDM settings:

1. Navigate to **Microsoft Intune admin center** | **Devices** | **Windows** and click + **Create profile**.
2. Choose **Windows 10 and later** as the platform and **Settings catalog** as the profile type.
3. After entering a name and description, proceed to the **Configuration settings** tab and click + **Add settings**.
4. You will be presented with a list of categories. We'll be using two categories: **Microsoft Edge/SmartScreen settings** and **SmartScreen**. These correspond to Edge settings for SmartScreen (to protect web use) and OS settings (to protect application use). Ensure that you do not choose the **users can override** category for Edge, for obvious reasons. In the settings catalog, as you check the box next to **Settings**, the settings will be added to the list of settings for the profile.
5. The policies it's recommended you configure for **Microsoft Edge/SmartScreen settings** are noted in the following screenshot. In Intune, you will have two choices for most, one of which has the **(User)** suffix. To apply at the device level rather than the user level, choose the option that does not mention "user." All settings should be set to **Enabled**:

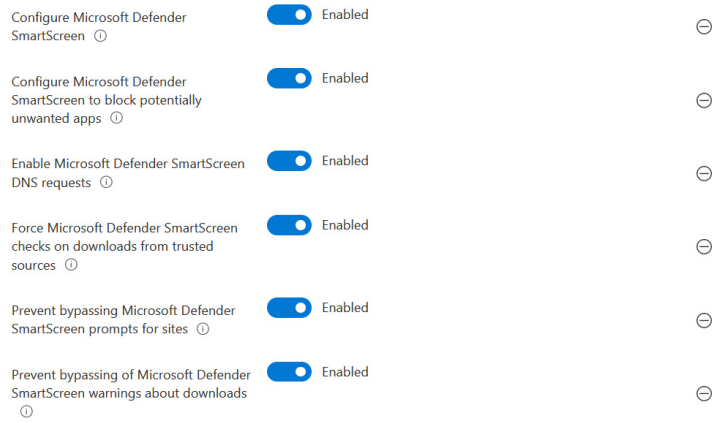

Figure 7.14 – Configuring SmartScreen for the web with Intune

6. The policies this book recommends you configure for **SmartScreen** are noted in the following screenshot. In Intune, you will have two choices for most, one of which has the **(User)** suffix. To apply at the device level rather than the user level, choose the option that does not mention "user." All settings should be set to **Enabled**:

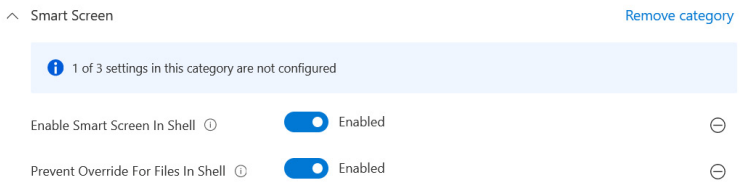

Figure 7.15 – Configuring SmartScreen shell settings with Intune

You may notice the omission of the **Enable App Install Control** option. App Install Control allows you to control if apps can be installed from locations other than the Microsoft Store. By turning it on, you block non-Store apps from being installed. While application control is something all defenders should be implementing, this setting is generally not the way to do it.

7. Proceed to assign the policy to the Azure AD groups you want to protect with SmartScreen.

With these policies, users cannot ignore SmartScreen and proceed to risky websites or apps. In Edge, the option to ignore SmartScreen is replaced with a warning that *You're blocked from continuing to this site*. If attempting a download, the option to **Keep** a file will be grayed out. If a browser that is unprotected by SmartScreen is used to download a file that SmartScreen would have otherwise prevented, it will still block it from being launched (thanks to the **Prevent Override For Files In Shell** setting), and the user's only option will be **Don't run**:

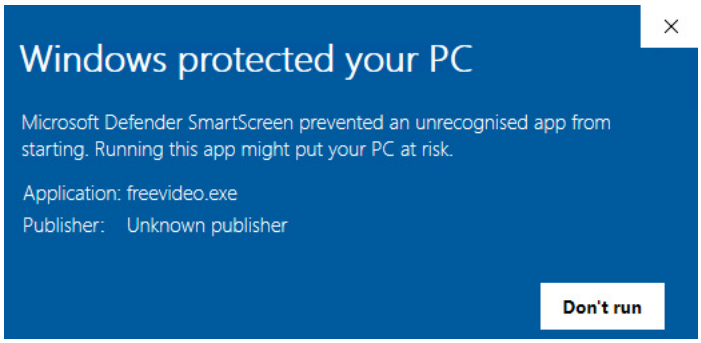

Figure 7.16 – SmartScreen stopping a risky app from running, with no option to ignore and proceed

You may not have the option of controlling SmartScreen using Intune. Under these circumstances, Group Policy remains an option.

Group Policy

As with Intune, we'll need to configure Group Policy in two sections: Edge for websites, and the general OS for app protection. Edge can be configured as either a **Computer Configuration** or **User Configuration** policy; app protection is limited to **Computer Configuration**. Your Group Policy may require the addition of the Microsoft Edge ADM and ADMX files, available at **aka.ms/EdgeEnterprise**.

Here's how to get going with the recommended settings:

1. We'll start with app protection. In your Group Policy Object, expand **Computer Configuration** and navigate to **Policies** | **Administrative Templates** | **Windows Components** | **Windows Defender SmartScreen** | **Explorer**. Ignore the **Microsoft Edge** option here; it is for the legacy version of Edge.
2. Configure your policies as follows:
    1. **Configure App Install Control** can generally be ignored. As explained in the *Intune* section, App Install Control can be used to limit app installs to Store apps only. However, this shouldn't be how you manage app control: a combination of limited administrative rights, AppLocker, Windows Defender Application Control, or third-party services is preferable. Only if you really think this is appropriate should it be used as part of defense in depth.
    2. **Configure Windows Defender SmartScreen** should be **Enabled** and the **Warn and prevent bypass** option should be selected:

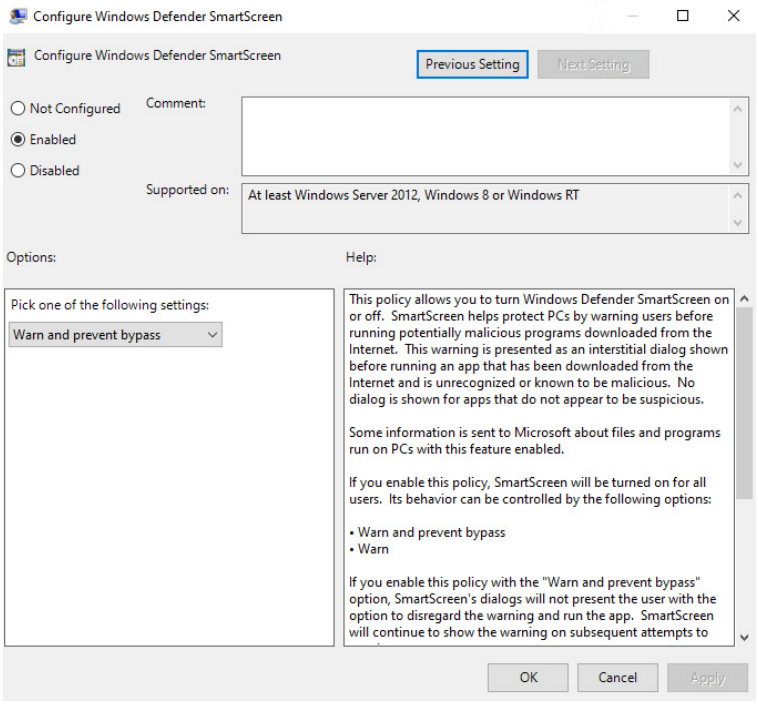

Figure 7.17 – Configuring a GPO for SmartScreen for programs

3. Next up are the policies for Microsoft Edge. In your Group Policy Object, expand either **User Configuration** or **Computer Configuration** (depending on your architecture, though **Computer Configuration** is generally more suitable as it applies to all users using a device) and navigate to **Policies** | **Administrative Templates** | **Microsoft Edge** | **SmartScreen settings**. Avoid the **users can override** version for obvious reasons.
4. The following are the recommended settings, with each set to **Enabled**:
    1. Prevent bypassing Microsoft Defender SmartScreen prompts for sites
    2. Prevent bypassing Microsoft Defender SmartScreen warnings about downloads
    3. Enable Microsoft Defender SmartScreen DNS requests
    4. Configure Microsoft Defender SmartScreen
    5. Force Microsoft Defender SmartScreen checks on downloads from trusted sources
    6. Configure Microsoft Defender SmartScreen to block potentially unwanted apps:

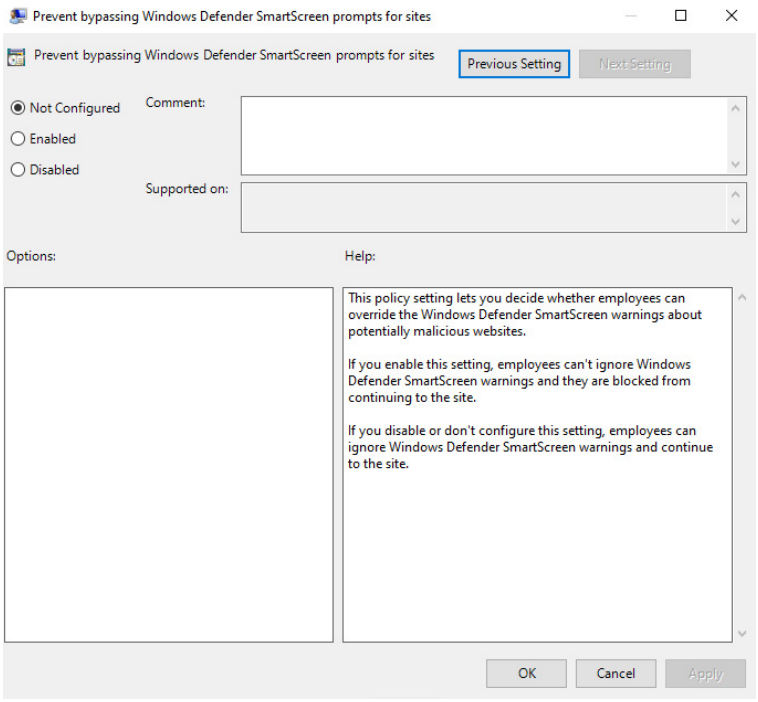

Figure 7.18 – Configuring a GPO for SmartScreen for Edge

5. Proceed to link your Group Policy Object to the devices or users it should apply to.

SmartScreen, as you've just learned, adds another line of defense to risky apps and websites. But it has limitations. For other browsers, such as Firefox, to benefit from SmartScreen capabilities, we need to turn to an MDAV active mode feature: **network protection**.

Network protection

Part of the ASR *stack* you learned about earlier, network protection can be looked at as an extension of SmartScreen beyond first-party web apps. Network protection applies across layer 3, defending against low reputation, C2, or exploitive outbound HTTP(S) traffic. Additionally, network protection powers network indicators of compromise (such as blocking IPs and websites based on your custom list), web content filtering (such as blocking entire web categories based on corporate policy), C2 detection and remediation, and unsanctioned web app enforcement from Microsoft Defender for Cloud Apps.

Network protection is provided by MDAV into Windows 10 1709 onwards and Windows Server equivalents (including with the unified agent). MDAV must be in **active** mode with **Cloud-delivered protection** enabled. Settings are applied to the entire device and cannot be limited at the user level. On Windows Server, network protection settings are, by default, ignored, so in the guidance that follows, you'll find additional steps to enforce it. In general, you should proceed with caution with network protection for servers, as there could be performance and compatibility implications. Azure Virtual Desktop multi-session hosts do not support network protection.

With this background information out of the way, let's get into how to deploy network protection.

Deploying network protection

Continuing with the trend of other ASR capabilities, network protection can be enabled in **audit** mode or **block** mode.

Intune

For Intune environments, network protection is enabled in the same MDAV profile as many other settings in an **Endpoint security** | **Antivirus** | **Microsoft Defender Antivirus** profile. This profile type only supports enabling network protection for clients, not servers, as the additional server options aren't available:

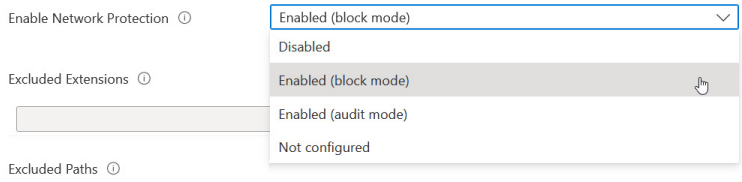

Figure 7.19 – Configuring network protection using an Intune Microsoft Defender Antivirus profile

Configuration Manager

In Configuration Manager, you can enable network protection for client devices, but you don't get the option to enable server support.

You will find network protection in the Configuration Manager console by clicking through **Assets and Compliance** | **Endpoint Protection** | **Exploit Guard** | **<your new or existing policy>**. Ensure **Network protection** is selected as an Exploit Guard component, then simply choose **Block**, **Audit**, or **Disabled** mode:

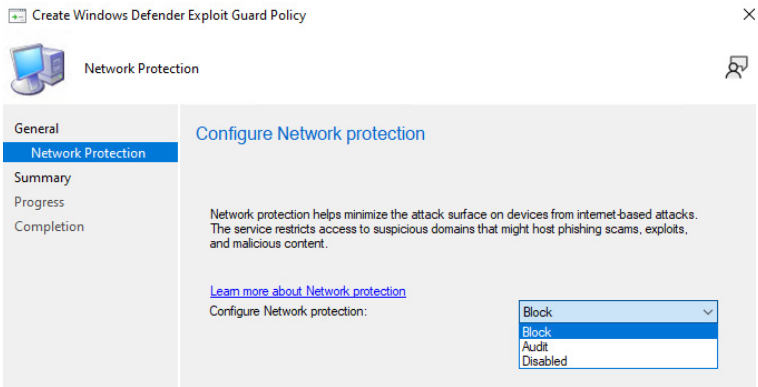

Figure 7.20 – Configuring network protection in a Configuration Manager Exploit Guard policy

Group Policy

Lastly, if you're using Group Policy, you can configure which mode network protection runs in by pointing your GPO to **Computer Configuration** | **Policies** | **Administrative Templates** | **Windows**

Components | **Microsoft Defender Antivirus** | **Microsoft Defender Exploit Guard** | **Network Protection** and enabling **Prevent users and apps from accessing dangerous websites**. Within this setting, you can choose **Disable (Default)**, **Block**, or **Audit Mode**.

In the GPO, you'll also see the **This settings controls whether Network Protection is allowed to be configured into block or audit mode on Windows Server** setting. As network protection settings are ignored by default on Windows Server, you should set this to **Enabled** if that's what you're assigning to:

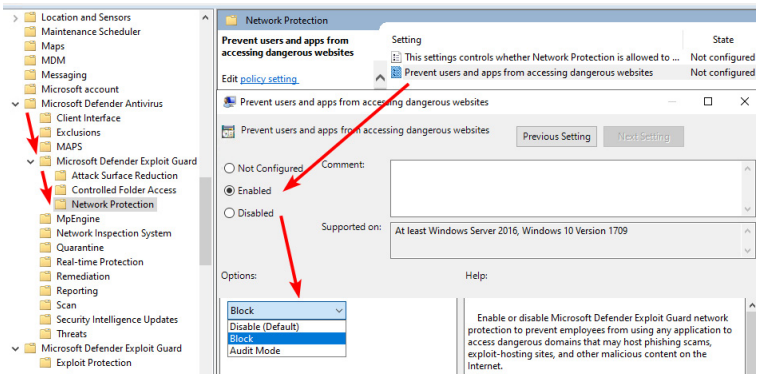

Figure 7.21 – Configuring network protection with Group Policy

If scoping to Windows Server 2012 R2/2016, additional steps are required. Using PowerShell, ensure the following values are configured:

```
Set-MpPreference -AllowNetworkProtectionDownLevel 1
Set-MpPreference -AllowDatagramProcessingOnWinServer 1
```

Monitoring network protection

Microsoft provides the following example of an advanced hunting query for looking into network protection logs. It could be edited to provide more specific results for your investigation; for example, by querying only `ExploitGuardNetworkProtectionAudited` while in **audit** mode:

```
DeviceNetworkEvents
| where ActionType in ('ExploitGuardNetworkProtectionAudited','ExploitGuardNetworkProtectionBloc
```

The `ConnectionSuccess` event can be misleading. This is based on networking data rather than MDE data; the block by network protection comes after the TCP 3-way handshake process. Therefore, a `ConnectionSuccess` event can exist, but network protection has prevented the device from actually accessing the resource.

Web protection

Web protection is a collection of features that, combined, protect users from risky websites by enforcing organizational policy and security. The

three features are **web threat protection**, **custom indicators for IPs and URLs/domains**, and **web content filtering (WCF)**.

Web threat protection

There is a degree of overlap in definitions of web threat protection and network protection. Ultimately, you should think of web threat protection as the element of network protection that protects Microsoft Edge and non-Microsoft third-party browsers against threats such as phishing, exploitation, and poor reputation. It is enabled as part of network protection, and in configuring network protection and SmartScreen, you configure web threat protection too without additional steps.

Custom indicators for IPs and URLs/domains

As you learned in ***Chapter 5***, indicators are objects that can have their access controlled. Specific to web protection, you can create indicators for public IP addresses and URLs/domains. Both can be managed in **Microsoft 365 Defender** | **Settings** | **Endpoints** | **Indicators** | **IP addresses** or **URLs/Domains** | + **Add item**. Indicators are applied to clients using MDAV's network protection feature when protection is active with **Cloud-delivered protection** enabled and are generally applied within a couple of hours of creation:

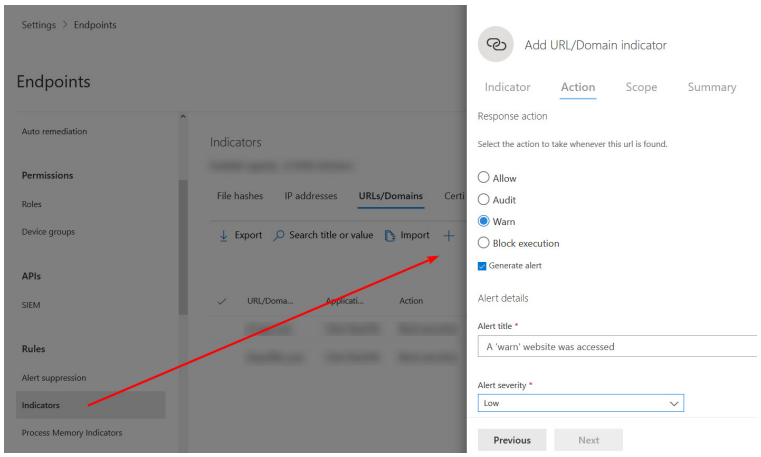

Figure 7.22 – Adding a URL/domain indicator in Microsoft 365 Defender

These indicators can be set up in **Allow** (that is, exempt), **Audit**, **Warn**, or **Block execution** mode. Except for **Allow** mode, a Microsoft 365 Defender alert can be generated too. **Warn** mode allows the user to ignore the content block and proceed by choosing **Unblock** in Edge (toast notification in other browsers). This is valid for the **Bypass duration (Hours)** value specified by an administrator when creating the indicator. For both **Warn** and **Block** modes, toast notifications have a **Feedback** button, allowing a user to raise a ticket using the contact details specified in MDAV:

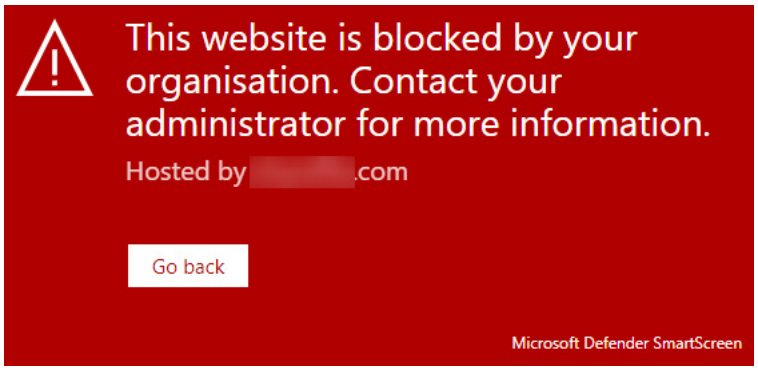

Figure 7.23 – Accessing an indicator-blocked website in Microsoft Edge

There are some intricacies with these types of indicators. For IP addresses, you cannot add a range (for example, a CIDR notation). For URLs, HTTPS supports the full path in Microsoft web browsers such as Edge, but only the FQDN in third-party browsers such as Firefox. HTTP URLs support the full path for both.

Web content filtering

In Microsoft 365 Defender, web content filtering allows an administrator to choose the categories of a website to restrict, and the device group(s) to apply those restrictions to. The classification of websites is driven by NetSTAR, an organization that focuses on URL categorization. Classifications are grouped into adult content, high bandwidth, legal liability, leisure, and uncategorized. Network protection powers web content filtering for third-party browsers, but for Microsoft Edge, SmartScreen is used.

To begin, web content filtering must be enabled for your MDE instance. This can be done in **Microsoft 365 Defender** | **Settings** | **Endpoints** | **Advanced features**. Ensure that **Web content filtering** is set to **On** and click **Save preferences**:

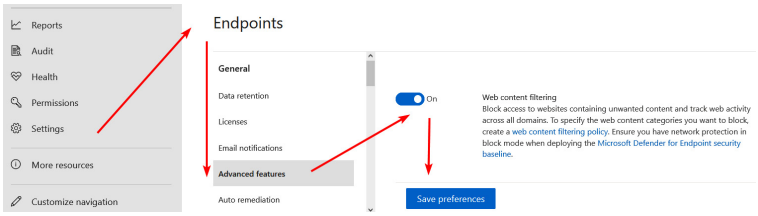

Figure 7.24 – Enabling web content filtering for MDE devices

Next, to customize your settings, head to **Microsoft 365 Defender** | **Settings** | **Endpoints** | **Web content filtering** | **+ Add item**. Each item (policy) entry in the web content filtering page has a list of categories to block and the scope to apply those blocks to. Therefore, if you want different rules for different device groups, you can create new policies. If a parent category is selected in the UI, all sub-categories are also selected by default, though you can edit this:

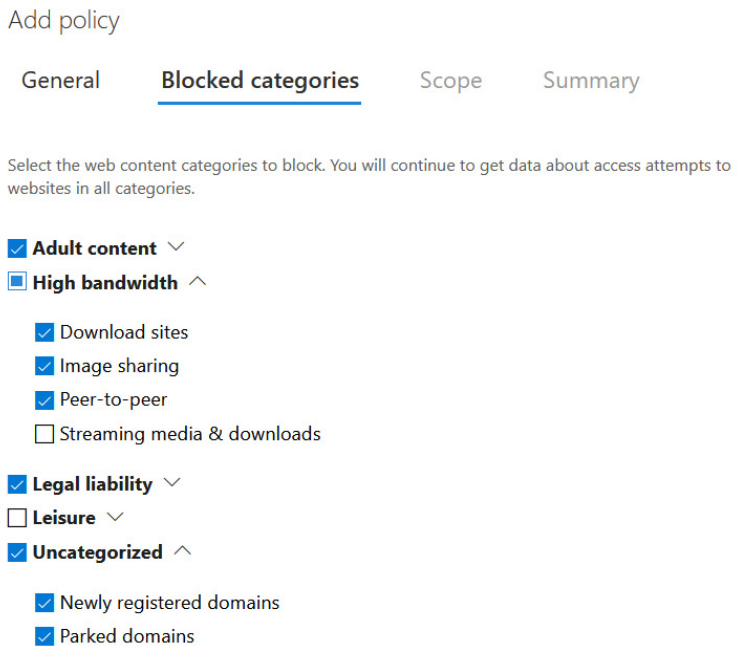

Figure 7.25 – Creating a web content filtering policy in Microsoft 365 Defender

Of particular note is the **Uncategorized** group. This benefits security as much as corporate compliance. For example, the **Newly registered domains** category may reduce access to dynamically provisioned phishing URLs, at the risk of blocking access to legitimate websites. Alternatively, if you are only interested in auditing and not blocking user access to website categories, you can create a policy without categories.

Once a policy has been created, devices will start enforcing the web content filters, generally within a couple of hours. Microsoft Edge will display a red warning screen if a website is blocked by web content filtering. Third-party browsers will display a general error message, such as `SSL_ERROR_NO_CYPHER_OVERLAP` in Firefox. This is true even for Chrome if the Microsoft Defender Browser Protection extension is installed:

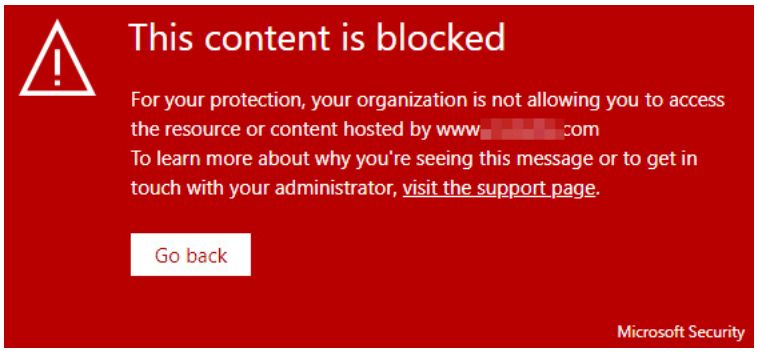

Figure 7.26 – Block message caused by web content filtering in Microsoft Edge

Using allow or warn custom indicators for IPs and URLs/domains, it is possible to override web content filtering blocks. The policy defined in an indicator is respected over the filtering policy, including threats or Microsoft Defender for Cloud App unsanctioned apps. This is useful in scenarios where you feel a website has been miscategorized, or if you

have a legitimate exception to the rule. Alternatively, if a website is obviously miscategorized, you can file a dispute that Microsoft will review:

1. Head to **Microsoft 365 Defender** | **Reports** | **Web Protection** | **Web content filtering categories details** | **Domains**.
2. You'll find a list of domains that have been found by web content filtering, which can be sorted by requests, blocks, trends, and machine counts. You can also filter by attributes such as category, device group, and the result: allowed or blocked.
3. Click the ellipsis (**...**) next to the domain you want to dispute and choose **Dispute category**.
4. Fill in the form and submit it:

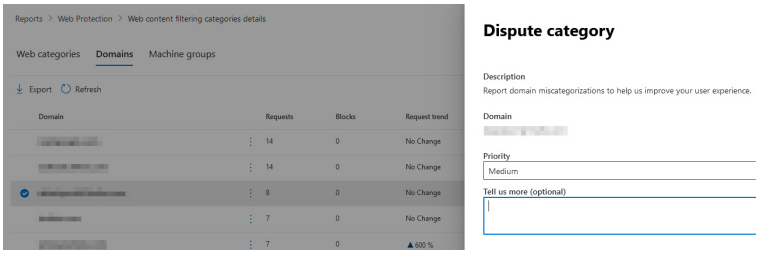

Figure 7.27 – Disputing a web content filtering categorization

Disputing categories are powered by the telemetry-gathering capabilities that web protection leverages, so let's move on to look at the additional reporting benefits that it provides.

Monitoring web protection

Heading to **Microsoft 365 Defender** | **Reports** | **Web Protection**, you'll find a dashboard of visual summaries based on web protection data, including the following sections:

- **Web threat detections over time** displays metrics of access to malicious, phishing, or custom indicator URLs over approximately the last month.
- **Web threat summary** shows similar information to **Web threat detections over time** but in a stacked bar chart so that you can see the quantity of threat category types relative to others. There is also an option to dive into its **Details**, which chose line-level detail of domains accessed, block results, and trends over time.
- The **web content filtering summary** report, which you learned about in the previous section, shows you blocks by category, with the option to dive into **Details** on the sites detected.
- **Web activity by category** shows similar information to the web content filtering summary but focuses on trends so that you can see spikes in access to certain types of websites that may be indicative of malicious activity.
- Lastly, **web activity summary** visualizes the number of requests by category rather than just the number of blocks:

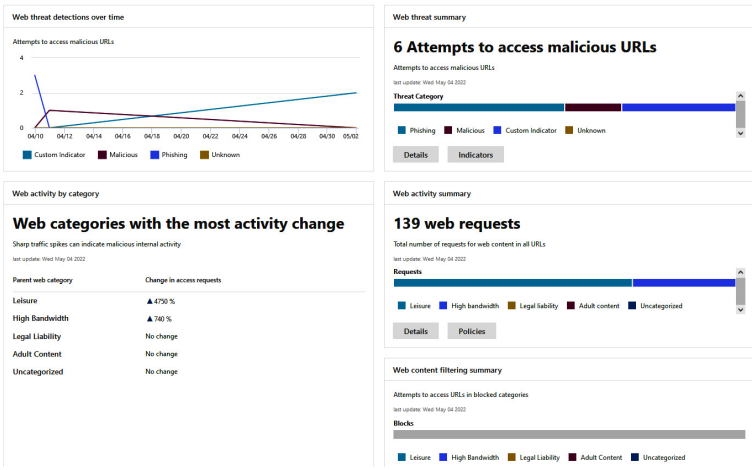

Figure 7.28 – Web protection reports in Microsoft 365 Defender

You can also use advanced hunting to find out about web protection events. For Microsoft Edge, these can be found under the **`SmartScreenUrlWarning` `ActionType`**; for other browsers, you'll get them under **`ExploitGuardNetworkProtectionBlocked`**. This lets us build a query such as the following:

```
DeviceEvents
| where ActionType in ('SmartScreenUrlWarning','ExploitGuardNetworkProtectionBlocked')
| extend ParsedFields = parse_json(AdditionalFields)
| extend BlockType = strcat(Experience=tostring(ParsedFields.Experience), ResponseCategory=tostr
| project DeviceName, BlockType, ActionType, RemoteUrl, Timestamp, InitiatingProcessFileName
```

The results of this are a list of each website blocked, regardless of browser, and a column called **`BlockType`** that returns either **`CustomBlockList`**, which represents a custom indicator (including Microsoft Defender for Cloud Apps unsanctioned apps), or **`CustomPolicy`**, which represents web content filtering.

In this section, you learned how SmartScreen, network protection, and web protection combine to defend endpoints and users against risky network resources, websites, and even apps. You can deploy them gradually by leveraging the **Audit** and **Warn** mode options to minimize business disruption, with the end objective being **Enforcement** and **Block** mode.

Summary

In this chapter, we dove into ASR, and you found out how to lower the likelihood of exploits and risk of vulnerabilities. You learned about how ASR, originally branded Exploit Guard, is comprised of four core features: ASR rules, controlled folder access, exploit protection, and network protection.

To recap, ASR rules are individually defined options that audit or prohibit (including the option to override) certain types of operations, such as Office applications creating child processes or running obfuscated scripts. CFA is primarily a ransomware protection feature that protects user fold-

ers from malicious applications of all kinds. Exploit protection lives on from the EMET to defend against potential OS and app exploits. Last of the four ASR features, network protection, guards the network layer against low reputation, C2, and exploitation. It powers the ability of MDE to block web content and sits alongside SmartScreen as a defense against low-reputation resources.

In addition to the four ASR features previously known as Exploit Guard, we explored how network protection compares and works with SmartScreen and web protection, rounding off your knowledge about proactive Windows security.

In the next chapter, we will round off our journey into securing Windows with MDE by exploring additional capabilities.

Questions

The following questions will let you test your knowledge of ASR for Windows. The answers can be found toward the end of this book:

1. You are testing web content filtering on a Windows Server 2022 server, but you find it is not blocking any websites. Which of the following may be a reason why? Choose all that may apply.
    1. The `AllowNetworkProtectionDownLevel` value is not configured
    2. Microsoft Defender Antivirus is in passive mode
    3. Network protection is only available for client devices
    4. Web content filtering has not been enabled for the tenant
2. Which of the following actions can you include in an advanced hunting query to review events involving a network protection block?
    1. `ExploitGuardNetworkProtectionAudited`
    2. `ExpoitProtectionNetworkAccessBlocked`
    3. `ExploitGuardNetworkProtectionBlocked`
    4. `NetworkProtectionExploitBlocked`
3. Web content filtering has been configured to block social media websites, but you have one website in this category that all employees are allowed to access. How should you proceed?
    1. Create an allow indicator
    2. Disable web content filtering
    3. Disable SmartScreen
    4. Create an exception within web content filtering
4. True or false: the **Block credential stealing from the Windows local security authority subsystem (lsass.exe)** ASR rule is still recommended when Windows Defender Credential Guard is active.
    1. True
    2. False
5. Which of the following can you use to begin obtaining an exploit protection configuration from a reference device? Choose all that apply.
    1. In PowerShell, use `Get-MpPreference`
    2. In PowerShell, use `Get-ProcessMitigation`
    3. In Windows Security, use **Export settings**
    4. In Command Prompt, use `MpCmdRun.exe`

Further reading

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

- Read Microsoft's announcement of (what was once called) Windows Defender Exploit Guard: **microsoft.com/security/blog/2017/10/23/windows-defender-exploit-guard-reduce-the-attack-surface-against-next-generation-malware**.
- Rudy Ooms MVP has some great insights into using Intune to harden Windows devices. You can read his article on CFA here, including a look under the hood at things such as events and registry entries: **call4cloud.nl/2021/06/married-with-controlled-folder-accesscfa**.
- For some insight into a wide-scale enterprise deployment approach for ASR rules, you can read about Palantir's journey and insights: **blog.palantir.com/microsoft-defender-attack-surface-reduction-recommendations-a5c7d41c3cf8**.
- Jonathan Gregson has a great example of an exploit protection XML on GitHub, which combines Microsoft recommendations, industry good practices, and his own experience: **github.com/jdgregson/Exploit-Protection-Settings**.