

Planning and Preparing for Deployment

In this chapter, we'll cover how to plan and prepare for your deployment of **Microsoft Defender for Endpoint (MDE)**. Most organizations have some degree of project planning and management standardization. The goal here is not to supplant that at all, but rather to give broad, general guidance on how you might go about planning a deployment (with as many specific MDE considerations as possible, of course). Certainly, IT and security teams can vary greatly in scope and responsibilities, and there's no plan that's going to fit everyone's needs. The idea is more to ensure that anyone at any level of understanding or responsibility could pick up this book and have everything they need to be successful. As you read through the chapter, don't be afraid to discard the ideas that don't fit your environment or level of expertise, absorb any new ideas, or mold the concepts to your needs. This chapter isn't meant to be followed rigorously or be all-encompassing, and some parts may even seem elementary or obvious. Rest assured though, even if you're working in enterprise environments with well-articulated service adoption procedures, there's still going to be something for you in this chapter.

We'll begin by loosely framing what a deployment plan might include. We'll then discuss the concept of personas and define examples as they relate to this project. Then, we'll start our planning by talking about how to define your project scope and gather relevant data. Once gathered, we'll work through how to break it down into manageable chunks for both remediation and deployment, and move into the deployment approach from there. In the end, we'll bring to the surface some of the key considerations to have in mind for each MDE feature.

After reading this chapter, you should have a good understanding of how to successfully plan an MDE deployment, including some general thoughts on project planning and pitfalls to keep an eye out for.

In this chapter, we will cover the following:

- Architecting a deployment framework
- Understanding personas
- Gathering data and initial planning
- Planning your deployment
- Some key considerations per feature

Architecting a deployment framework

Many of you may already have experience in developing an architectural framework for deploying new infrastructure or services, while others may not. We won't go deep into it here, but after working through this chapter and capturing all the relevant data from your environment, you should aggregate the information into a structured project plan or deployment framework. For our purposes, at a minimum, you would want it to include the following:

- A design document:
 - Which systems you'll be targeting (also known as defining your scope)
 - Which antivirus will be used (if not yet adopting the full platform immediately)
 - Which **Endpoint Detection and Response (EDR)** product will be used (if not yet adopting the full platform immediately)
 - What tenant the data will reside in
- Two architectural diagrams:
 - The existing architecture
 - MDE added (with an overlay or new parts colorized)

- Select your configuration options: List them out fully and run a series of workshops to determine which configurations will be implemented
- Document prerequisites and dependencies:
 - MDE tenant provisioned
 - Network connectivity verified
 - Configuration management needs
 - Licensing
 - SIEM integration
 - Client telemetry configuration
 - Any third-party product concerns (co-existence or migration)
- Granularly define a technical implementation plan (this should be verbose and detailed):
 - Prepare for enrollment
 - Enroll MDE onto endpoints
 - Configure the MDE portal
 - Have a backout plan

This will by no means be a bulletproof approach and is simply meant as a high-level guideline of how you might structure a project plan for an MDE deployment. We will go over many of these concepts and details in this chapter, as well as the next two chapters. This chapter focuses heavily on early planning and approach for deploying MDE, and the next two dig deeper into technical details and operationalizing the product. Much like an exam in school, we suggest a full read-through before trying to put pen to paper (or worse, trying to turn things on). Once you have a good grasp of the information, come back and use these chapters as a reference to document your plan.

Understanding personas

Born of Agile development practices, personas are a great method for thinking critically about the key players in any project, their motivations, and their needs. In the case of security tool implementations, there are

four primary personas that encompass the most important stakeholders to project success. Those are as follows:

- Leadership
- IT admins
- Security admins
- Security operations

Let's define each in relation to this text so that we can be on the same page whenever they are referenced.

Leadership

We'll avoid delving into leadership variations too much because leadership structures can vary greatly from business to business. From our perspective, it is crucial to make sure that you have support from your leadership, that you are communicating with them at every step, and that you realize that they are almost always appreciative of thorough documentation and structured planning.

Even though beyond reporting cadences, leadership won't really factor into the granular planning steps themselves, that doesn't make them any less important as a stakeholder. Historically, there's a well-known lack of communication between security and IT teams. Though they rely on each other for input on best practices and implementation assistance, they work very independently in a lot of organizations. This has slowly improved over the years, each realizing the value of communicating with the other, but it's still paramount to solidify buy-in from leadership and engineering commitment *prior* to undertaking a project like this. This ensures that the project plan is not only thorough and thoughtful, as you'll need to convince leadership that it's worth pursuing, but also that the project has the support needed for the inevitable challenges along the way.

IT admins

For our purposes, an IT admin will be defined as any individual within your organization who stands up and manages infrastructure, networking, and/or devices. Though MDE is a security tool, it requires the configuration of the endpoints themselves, and the reality is that most security policies for endpoints are ultimately implemented by IT admins. These are the people that understand the management tools in play, the network changes that might be required, and generally, how things are going to impact the business and end users. Their situational awareness doesn't come from caring more than security administrators, but rather from their closeness to the frontline of support escalations, many of which likely resulted from poorly planned project implementations.

IT admins tend to own and care about patch management (sometimes with a friendly nudge from security admins), operating system development and deployment, software administration, device configuration, and integration of systems. Most of these skill sets will come into play in this project, so try to get someone who can be dedicated to its success throughout. In the end, their primary goal is IT environment stability and keeping costs down. The primary outcomes they'll be responsible for in this project are as follows:

- Ensuring that all devices are fully patched
- Delivering and/or configuring prerequisites
- Onboarding devices to MDE

Though network administrators could easily be their own persona, they still fall under the umbrella of IT admins in this context. They're part of making sure that the infrastructure and devices can support the service.

COLD SNACK

Network configuration issues are one of the most common blockers for successful deployments, and a solid understanding of your environment's network architecture will be necessary. If network engineering is a separate

team within your organization, then ensure that you also have a network engineer dedicated to your project.

Security admin

Security administrators hold the responsibility for vulnerability assessment, security management, security configuration, rights management, and identity management, and generally define the security policies for your organization (which are ultimately implemented by an IT admin). In general, their goal is to harden the environment against attacks and to be the proactive arm of your security organization. Security admins will likely prioritize quick remediation of impacted devices, reduction in exposure through software and firmware updates, and ensure that their security operations counterparts have the tools they need to do their job. Remember that IT admins value stability and cost minimization; the two complement and temper each other very well.

Security admins typically own the configuration of the MDE portal and **role-based access control (RBAC)** and will want to have input into the configuration of advanced features. An example would be allowing unsigned code execution via **live response (LR)**. Even though security operations professionals will be the primary users of LR, security admins would likely have defined the relevant code-signing policies for your organization. So, they'd be best equipped to speak about how it should be configured.

Security operations

The last of our example personas are the security operations analysts and engineers. These are the folks within your organization that monitor threats and respond to security events. They perform forensic analysis using tools and available logs to investigate suspicious behavior on systems and networks. They augment their understanding of the current threat landscape with threat intelligence from multiple sources, which

they can then combine with their expertise to design and execute hypothesis-based, targeted hunts for threats in their environment.

This group is the reactive arm of your security organization, handling the incoming response need by investigating and mitigating threats, and thus is ultimately the primary user of the MDE platform once it's implemented. Their input into the project plan includes an explanation of their operational model to inform configuration decisions (especially RBAC), review and input on advanced feature needs, and planning around how to operationalize the product once implemented.

The biggest initial focus will be feature needs and configuration settings, such as **automated investigation and response (AIR)** levels. Once those are outlined, time should be dedicated to familiarizing themselves with the product from an operational standpoint. Perhaps creating use-case playbooks, documentation around the approach, or developing a training plan to quickly ramp analysts on MDE as they onboard. ***Chapter 8, Establishing Security Operations***, is meant to be a primer for just this sort of familiarization: giving practical examples of how to monitor, investigate, and mitigate alerts and incidents. The Microsoft Learn modules related to MDE are also an excellent place to start. These can be found at <https://learn.microsoft.com/en-us/training/paths/sc-200-mitigate-threats-using-microsoft-defender-for-endpoint/>.

It's important to clarify that the personas we've outlined could be different than what is available in your organization. One person might handle all three of these roles and will need to put on each hat independently as they think through the needs and responsibilities of each aspect of their job. The opposite could also be true, where your organization is such a large enterprise that each of these roles is comprised of multiple teams with very granular focuses. The goal will be to take this outline and use it as a starting point to define your own personas using your knowledge about your own organization. Once defined, use them to think about whose expertise is required, what questions you need them to answer, and how they can be engaged to ensure the success of your project.

Once you have clearly defined personas, it's time to move on to discovery and initial planning.

Gathering data and initial planning

Once the decision is made to move forward with any new IT implementation project, the next step is to answer these four questions:

- What do you *need*?
- What do you *have*?
- What changes need to be made to what you *have* to get what you *need*?
- How and when are you going to make those changes?

This intentional oversimplification is helpful in understanding the goal of the discovery and preparation phases relative to your unique environment. Remember, you've already identified broadly what you want to do. You want to deploy MDE in your environment to improve your security posture. Now, you need to clarify what you need granularly (scope), discover what you have specifically (discovery), and solidify the changes required to close the gap into a plan. Then, of course, you must execute that plan to get what you need out of the project.

Defining scope

This is the *what you need* (or *want*) aspect of our simple model mentioned previously. In this step, you will need to consider exactly what systems you're focused on deploying MDE to and the features you need to be enabled on them. You'll want to define what systems you're going to target and ensure you choose compatible platforms (or work to upgrade any devices that you want to deploy to that don't meet the minimum requirements). Current compatible platforms are as follows:

- Windows 7 SP1, 8.1, 10, 11, and IoT.

- Windows Server 2008 R2 SP1, 2012 R2, 2016, LTSC 2016+, 2019+ (including Core edition), and 2022.
- Azure Virtual Desktop and Windows 365.
- Linux: Distributions that support systemd system management. The list of supported distributions is long, specific, and ever-evolving. You'll want to check online for the current list of supported distributions and kernels at the time of planning.
- macOS: The three most recent releases (also referred to as **n-2**).
- iOS 13.0+.
- Android 6.0+.

Note that, though some very old versions of Windows and Windows Server are listed, it is highly recommended that you upgrade to more recent versions. Windows 7 and server 2008 R2 require the purchase of extended support licenses to even be supported. **Down-level** operating systems like these may also have limited features in some aspects of MDE. For example, Windows 7 and versions of Windows 10 lower than 1703 only support OS vulnerability reporting in Microsoft Defender Vulnerability Management (of the six assessments it's capable of on more modern operating systems). We go deep into OS compatibility in the next chapter but wanted to mention it here to ensure that it's being considered as part of your planning phase.

You'll also want to be certain that the platforms you want aren't just supported but are also manageable by your organization's configuration management tools. If not, you won't be able to remediate any issues found, handle prerequisite configuration needs, or actually perform the onboarding to MDE when the time comes.

Performing discovery

The *what you have* step from our simple model could also be called the *discovery* phase. In this step, you'll use the tools available in your environment to gather data on the devices you've determined are in scope for deployment, working to glean a solid understanding of them, their OS

version, configuration, and risk if they were to be exposed or impacted by deployment.

The goal will ultimately be to subdivide these groups into buckets based first on remediation needs, such as patching and network configuration concerns, and then separately on the deployment method selected, including caveats for networking differences. You may even further divide them by speed of rollout, being more cautious with mission-critical systems. This is highly dependent on the complexity of your environment, but the rest of this chapter should give you more ideas on things to consider. Make sure you spend sufficient time on this step to truly understand your systems; it will help you avoid unexpected impacts later.

Identifying your device management architecture

Since you'll be leaning on it heavily for the previous discovery, it's important early on to get clear documentation on which device management architecture you use within your environment:

- **Cloud-only management:** Sometimes referred to as **cloud-native management**, this is where your management platform for the devices in question is strictly cloud-based. Microsoft's example of this is Intune. Unfortunately, Intune cannot manage Windows servers. You can gain some visibility through what's called **tenant attach** of the Microsoft **Configuration Manager (ConfigMgr)** to your Intune instance. However, you can't manage the servers from the Intune portal at all. So, if you have server operating systems in your environment, Intune will most likely not be your sole management approach.
- **On-premises management:** On-premises management, for our purposes, is the common combination of **Group Policy (GP)** and ConfigMgr in Microsoft **Active Directory Domain Services (AD DS)** environments.
- **Hybrid management:** Hybrid management can be achieved through two primary methods if using Microsoft tools. The first is what's known as co-management, which is Intune and ConfigMgr working in

tandem to manage any given device. The second would be extending ConfigMgr off-premises using an appliance called a **Cloud Management Gateway (CMG)**. This gives you the ability to put ConfigMgr sites in the cloud and allows you to manage clients seamlessly over both the internet and the intranet.

- **Non-Microsoft management:** There are lots of other options out there for management, such as **Jamf** for **macOS** clients, and tools such as **Ansible**, **Chef**, and **Puppet** for server management. Regardless of which your organization is using, these fall into the same bucket when it comes to the deployment approach. So, just make sure you note what method is being used for which devices so that you can later choose the recommended method for deployment.

Patching and device health

As a part of the discovery phase of your preparation, you'll also need to identify remediation needs and incompatibilities. The most important thing will be to get your operating systems up to date, but right behind that is ensuring that the devices are properly enrolled in and manageable by their relevant configuration management tool. For example, if you're using ConfigMgr, make sure the agent is healthy and reporting back to your ConfigMgr infrastructure.

We all know what needs to be said here: please, please have a patch plan in place. We cannot stress enough how important this is. Repeatedly, we see a lack of patching as one of the core reasons for security incidents, whether that be OS-based or application patches missing. Not only is patch management important for these reasons, but it is also how you'll keep your systems updated with the latest security intelligence updates, or perhaps an update to the EDR components!

Assessing application compatibility

You'll also want to assess the compatibility of applications beyond just the operating system. The first application compatibility assessment task will

be to ensure your teams can access the MDE portal. One would think it's obvious, but make sure that your security administration and operations teams are using supported browsers for accessing MDE. At the time of writing, that's only Microsoft Edge and Google Chrome. Though other browsers may work, they are technically unsupported.

On Windows clients, diagnostic data settings need to be enabled as well to be compatible with MDE. They are enabled by default, but if you've turned them off on certain devices, you'll need to plan on reenabling them. A great point to note here is that it doesn't matter what level you have diagnostic data settings configured for as long as they're enabled.

The last application compatibility check is potentially the most difficult. You need to take stock of any existing endpoint security solutions that you plan on having to coexist with any aspect of MDE. If you plan on migrating fully to MDE and are adopting the entire suite, then this is less concerning as the products only need to coexist in minor ways through the transition. However, if you are going to be adopting MDE slowly or selectively, then this can be a much more complex undertaking. Unfortunately, it's also a realm where we can't give as much guidance, as there are just too many possibilities.

Our biggest advice here would be to not attempt to use two similar products simultaneously, such as multiple antivirus programs in tandem. If nothing else, this would cause a significant resource hit, but is much more likely to cause more significant issues. However, for other scenarios, you will need to thoroughly test and ensure you're talking to your third-party application vendor about any known compatibility issues. This bleeds right into a conversation about migration approaches versus net new deployments, but we'll cover that later in the chapter.

COLD SNACK

The biggest thing to keep in mind with coexisting security products is that they will often detect each other. Make sure you have in your plan to add

exceptions to each for the other(s).

Reviewing network architecture

In addition to gaining a deep understanding of the state of your devices, you'll also want to assess your network infrastructure and take stock of changes or configurations that are needed. MDE leverages cloud services, which means that internet connectivity is key to maximizing the benefits of the platform. Not only will testing network connectivity to the appropriate endpoints be needed here, but you should also take stock of any non-transparent (traditional) proxying of internet traffic. In some cases, such as offline network segments, you may even need to add a proxy as a method to enable full MDE functionality.

As previously mentioned, network layout comprehension is most often one of the larger roadblocks when it comes to large-scale rollouts. This can be caused by many things: lack of documentation of current network architecture, lack of communication between teams because of compartmentalization inefficiencies, or even just employee attrition. Nonetheless, it is very important to understand the various network paths that devices need to take to reach the internet.

Some environments will have easy paths out, and some will need more complex planning if there are layers of firewalls or proxies between devices and the internet. This is the time to meet with network teams throughout the organization and express the intent of what you are looking to do. The goal is to get devices onboarded to MDE and talking to a well-defined list of URLs. The current list of URLs can be found at <https://aka.ms/MDEURL>.

The MDE client analyzer tool to validate functionality can be found at <https://aka.ms/mdeanalyzer> (more on this in [*Chapter 7, Managing and Maintaining the Security Posture*](#)).

A key consideration here is *offline* scenarios. Let's be specific: if you are using MDE, you'll want to think long and hard about how to make sure that your machines can access the relevant cloud services. In most cases, there is a possible path to those services, and you will need to make sure this path is facilitated. This is not *offline*, rather its access to the internet is controlled and limited (proxy, firewall, private peering, etc.).

COLD SNACK

Air-gapped, where there is literally no path to access anything that is not on the local network, is a different story. You can get some core capabilities going but be aware of the limitations! Essentially, the basic antimalware capabilities are available, but you need to ensure regular updates to be protected against the latest malware.

Analyzing the results

Now that you understand your estate well, you can start to gather a list of prerequisites required. This is the *what changes are needed* phase of discovery. Some OS versions are maybe unsupported and need to be upgraded, some may just need to be patched, some may need Defender configured in a certain way, or some may have special network considerations due to having a proxy or special firewall rules – the list of possibilities is near-limitless. Whatever the case may be, it's important to clearly understand the variations on the systems that you want to configure to avoid impact and ensure success. Now, document those findings and turn them into action items to get resolved. Create buckets of items and a *burn-down list* of remediation tasks required and get the work assigned out.

Once you're clear about what the work and stakeholders look like, you can also start to build a timeline around your project. This isn't a project management book (though this chapter may read a bit like one), so we won't go deep into timelining. We just believe it's worth saying out loud that projects with timelines are more likely to get done, especially when other teams are involved. Get yourself into their planning cycles. Get your

needs prioritized by giving examples of recent ransomware statistics – whatever you need to do to get those prerequisite requirements and remediation items on their to-do lists.

Now that all the project scoping and endpoint prerequisite needs are defined, all that's left is to plan your deployment of MDE itself.

Planning your deployment

Finally, you've done all the legwork, engaged all the stakeholders, and discovered and scoped your way to a clear understanding of your environment and its hurdles. Now to focus on what you came here for: planning your MDE deployment. Important considerations at this point are as follows:

- Logically grouping your target devices
- Determining a rollout cadence
- Selecting your deployment method
- Understanding SOC needs
- Creating a backout plan

Creating buckets

With your initial discovery done and your configuration management architecture understood, you should be ready to start to solidify the logical groups you've been using to understand your environment, based (at a minimum) on the operating system and management approach. Again, don't be afraid to create more granular buckets for critical systems that require a light or more careful approach.

Once finalized, you'll want to plan on creating these logical groups within your device management or identity infrastructure. For example, if you're focusing solely on Intune-managed Windows clients, you would create clearly named AAD device groups that represent your logical

groups and contain the relevant devices. Though your tools may vary, the idea will still follow. Use logic to get your devices into buckets that make deployment targeting easy. This will also help review your quantities and subdivide things, such as ring deployment methods, which, now that it's been brought up, should be talked about.

Taking a gradual approach

It is recommended that you take a ringed approach to implement any significant change. If you're unfamiliar with ring deployment approaches, it's really just another way of looking at a gradual rollout, and, just like any gradual method, is meant to allow for the early detection of issues and hopefully avoid major impact as a result.

Imagine a dartboard with concentric circles radiating out from the center. The bullseye in the middle is the beginning of your deployment; you're all ready to go, but you haven't deployed to a single system yet. The outer edge is the full deployment of your change to all relevant production systems. Everything in between is the progress to that end.

The first ring outside of the center is your group of test devices (often referred to as the **canary**, **certification (cert)**, or **development (dev)** ring). These systems are designed to get early, maybe even **beta**, versions of software and to act as *canaries in the coal mine* for potential impact on your environment. This first ring can contain as many devices as you're comfortable with, but make sure they belong to folks that have a direct line to support. A canary isn't useful if no one knows it died. You also need to make sure that this group has a representative sample of relevant devices in your environment (different operating systems and versions, significantly different configurations/applications, etc.) at a minimum.

The next ring outside of the center is just as representative of the overall population but includes more systems. Often referred to as the *pilot* or *early adopter* ring, this could perhaps include the entire IT department and a handful of informed users from different departments. You can

even have an approach where you allow anyone to sign up for this ring with a caveat that they might be impacted (but they are super helpful to the IT department). The next ring beyond that one would include a larger sample, and so on until you've built enough confidence that the change won't cause an undue impact on your environment.

With your newly discovered confidence in hand, you can go ahead and push the change fully to production. How many rings and how many systems are included in each are totally up to your organization's appetite for risk. Keep in mind that risk appetite should also vary from system to system, depending on the purpose or criticality. This means that not all ring deployment structures will be the same, even within the same environment. That said, don't be afraid to make changes to mission-critical systems. That's an old mentality that keeps systems vulnerable. You can still make changes to them; you maybe just need to go slower or test more thoroughly.

Selecting your deployment method

As mentioned in the IT admin persona, how best to roll out MDE is entirely dictated by this architecture. This section's goal is to explain what deployment method fits with each. In *[Chapter 6, Considerations for Deployment and Configuration](#)*, we will dig deeper into each configuration management option and provide insight into things you should be mindful of. The goal here is simply to assist with identification during the planning phase.

To get started, review the note where you captured which device management approaches are used in your environment earlier in this chapter. Odds are good that there are at least two in play. What follows are the recommended deployment approaches for each.

Non-Microsoft operating systems and evaluation

The non-Windows devices you are deploying to will require that MDE agents be downloaded and deployed to them. This is an excellent point to double-check that those operating systems are supported and to ensure your plan includes downloading the relevant installation packages from the **Microsoft 365 Defender (M365D)** portal (also referred to as **Microsoft Defender Security Center**).

This method includes leveraging scripts to automate the process. Since this scripted method doesn't require configuration of your device management infrastructure at all, it's also the preferred method for evaluating the product and deployment to network segments where your device management infrastructure isn't connected (such as DMZs). Even though you may deploy to a small number of devices at first, avoid using this method for piloting efforts (the step beyond evaluation) unless it's ultimately going to be your deployment method to those system types. Because, during a pilot, you'll of course also want to be piloting the effectiveness of the relevant deployment approach.

The steps you need to document in your project plan for this approach are as follows:

1. Go to the M365D portal (<https://security.microsoft.com>), **Settings | Endpoints | Device Management | Onboarding**.
2. Select the appropriate operating system and your preferred ** configuration management tool* option (where * is replaced with the operating system name).
3. Click **Download onboarding package**.
4. Use the documentation, both from Microsoft (if available) and for your configuration management tool, to plan the deployment based on best practices.
5. Deploy the package.

Intune

In a cloud-only or cloud-native deployment, you would use Microsoft Intune to deploy to your client systems. This is also a good option if you don't have any existing management or deployment tools that can support an MDE deployment, as the cost and effort to stand up an Intune instance is minimal compared to other options where you would need to deploy and manage your own infrastructure.

The steps you need to document in your project plan for this approach are as follows:

1. Configure **MDM User Scope** in **Azure Active Directory (AAD)** to enable automatic enrollment (*only required on a new Intune deployment*).
2. Assign licenses to users in AAD and make sure that the devices actually get enrolled (*only required on a new Intune deployment*).
3. Run the initial setup wizard in the M365D portal and connect MDE to Intune by turning on the **Microsoft Intune connection** setting. This is enabled under **Settings** | **Endpoints** | **Advanced features**, as shown in the following figure:

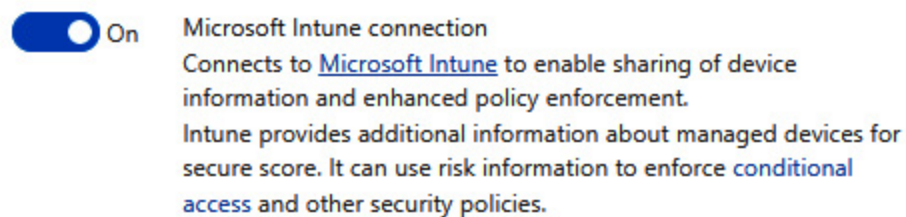


Figure 5.1 – Microsoft Intune connection enabled in the M365D portal

4. Create AAD device groups that bucket systems, as discussed earlier in the chapter.
5. Under the **Endpoint security** node in Intune, create **Antivirus**, **Endpoint detection and response**, and **Attack surface reduction** policies targeted at the relevant device groups you've created in AAD for the stage of your deployment you're in (i.e., cert, pilot, production, etc.).

COLD SNACK

Note that, even though you can manage macOS devices with Intune, if you use Intune to deploy MDE to them, there is no native deployment method. You will need to download the installation package from the M365D portal, upload it to Intune, then use a custom configuration to deploy it.

Group Policy or Configuration Manager

For your Windows servers and clients that are solely managed by on-premises resources, you have two options for deployment, GP or ConfigMgr. There's a chance that you may use these in tandem, so we'll describe them both together in this section.

The step you need to document in your project plan for this approach is as follows: Go into the M365D portal and download a Configuration Manager or Group Policy package from the **Device management** | **Onboarding** blade:

- **For ConfigMgr:** Deploy the package to a collection you've filled with devices per your logical grouping approach
- **For GP:** Extract the downloaded files to a shared location reachable by your target systems and target the **Organizational Unit (OU)** you've used for logical grouping with a **Group Policy Object (GPO)** that creates a scheduled task to run the **.cmd** file from the shared location

Co-management

If your organization is co-managing clients through both Intune and ConfigMgr, while managing on-premises servers through ConfigMgr and GP, then the steps you need to document in your project plan are just a combination of the previous Intune and ConfigMgr steps. You should just opt for the Intune method anywhere you can, and ConfigMgr (and GP if needed) as backup options for devices that can't be targeted with Intune.

Understanding security operations needs

Though **Chapter 8**, *Establishing Security Operations*, is dedicated to this topic, it's important to outline a few aspects of security operations during the planning phase, especially as it relates to access control.

SOC tiers

Microsoft recommends and employs a three-tiered model within the **security operations center (SOC)**. Let's define what that means for our context, as we'll use it in a practical example of access control implementation later in the chapter.

In a three-tiered SOC, the responsibility moves from efficient mitigation of a high volume of alerts all the way through to slower, more methodical threat hunting:

- **Tier 1 (T1):** Acting as the front-line defense, the first tier of the SOC focuses on delivering high-speed remediation of a large number of events. That inherently means they focus on well-defined, high-fidelity alerts. This means the alert is trustworthy, the mitigation is clearly known, and in many cases, the whole incident is automatable. An important note for access control is that a T1 team might only need visibility to investigate within their own geolocation if your organization is spread across multiple locations.
- **Tier 2 (T2):** The tier 2 team gets involved when a deeper analysis is required. This inherently means lower overall volume, but also more ambiguity in the type of incidents that are worked. It also means lower fidelity alerts in some cases. T2 will also work on escalations from T1. These may include requests for support on alerts that turn out to be more complex or more concerning, or for high-value assets. In a large enough SOC, T2 may also be restricted by geolocation, perhaps through a regional grouping of T1 SOC-responsible areas, with multiple T2 teams acting semi-independently to cover all regions of the organization.
- **Tier 3 (T3):** These are your threat hunters by trade, proactively searching for and mitigating hidden threats in your environment.

They will also generally have an escalation path for engagement from T2, especially when a broader environmental investigation is required. This is not because the T2 analysts don't have the skill set to widen investigations; it's because, in some cases, the volume of work at the T2 level is too high to allow the time it can take to deal with the broader investigation and mitigation of a widespread campaign, or it could be simply that T3 has broader visibility by nature of your SOC structure. Some other functions T3 might engage in are custom detection creation (as the result of a hunt, for example) or what's known as purple team operations. Purple-teaming is where they partner with an internal penetration testing team for adversarial practice, understanding of vulnerabilities discovered, improvements in defense processes and procedures, and mitigation action item creation. To facilitate their need for advanced use and broad visibility, T3 security experts should be authorized to perform all actions in the portal across all geolocations.

COLD SNACK

*Though well known in the security industry at this point, terms like **purple team** may not make immediate sense to someone new to information security. Historically, SOC teams are defensive and are designated as the **blue team**. Penetration testers are considered offensive and designated as the **red team**. From a primary color perspective, if you mix red and blue, you get purple.*

Though this three-tiered model is common, your organization may differ in approach. Use this as a baseline, much like the personas discussed earlier in the chapter, to define roles within your organization. Keep this defined SOC model in mind as you read on about role-based access control.

Understanding RBAC

If you've worked around cloud or identity for a while, you've likely heard of RBAC. This is the method employed by Microsoft and others to control

granular access to resources by defining a set of permissions as a role and assigning that role to those acting in the relevant capacity, thereby granting them all relevant permissions at once.

MDE's RBAC implementation lets you granularly define, though not always perfectly, groups of actions as a role definition. It also gives you the ability to separately define what those users have visibility to via **device groups**. So, in the context of MDE, you might give a group full access to take any action they want on a select few devices or give them broad visibility to devices but restrict them to read-only access. It depends on your model and the needs of each group involved.

In its simplest form, the steps to planning an RBAC policy are as follows:

1. Create device groups.
2. Define and create admin roles.
3. Assign the device groups to the appropriate users.
4. Assign the roles to the appropriate users.

COLD SNACK

*Until RBAC is defined for MDE, the only permissions are full access and read-only access. Full access is granted by a **Global Administrator (GA)** or **Security Administrator (SA)** from AAD. Read-only access is granted by a **Security Reader (SR)**. Though full access through a GA and SA will persist when you enable RBAC, read-only access will no longer be available via the SR role. This means that it's important to design your RBAC strategy during the planning phase and have all parties accounted for prior to cutting over so no one loses visibility.*

If device groups are going to be a part of your architecture, you should plan to get them assigned prior to assigning roles. Once roles are assigned, any devices in device groups without a user assignment are visible to everyone with portal access. If you're planning well ahead, as this chapter is recommending, then you'll be able to implement both at the

same time and avoid concerns about unwanted device visibility. Since we're on the subject, let's dig into device groups before we move on to designing your roles.

Creating device groups

Previously in the chapter, we discussed creating logical groups of systems for ease of deployment. Creating device groups is a similar exercise but based on security operations needs and AIR configuration, rather than remediation or deployment efforts.

Configuration of device groups exists within the M365D portal, under **Settings | Endpoints | Roles | Permissions**. This allows you to create a group that gets dynamically filled with devices. What devices get added to the group is based on logic you supply in the **Devices** tab when creating or editing a device group, as can be seen in the following figure:

Add device group

General **Devices** Preview devices User access

Specify the matching rule that determines which devices belong to this group.

| And/Or | Condition | Operator | Value | |
|--------|-----------|---------------|-------------|---|
| | Name | Starts with ▼ | Value | + |
| And | Domain | Starts with ▼ | Value | + |
| And | Tag | Starts with ▼ | Value | + |
| And | OS | In | Select... ▼ | |

Back Next Cancel

Figure 5.2 – Device group Devices tab

As you can see from the figure, you can add devices to the group based on a few logical boundaries: naming convention, domain, tags, and operating system (which allows multiple selections). Note that these are all **And** statements, so any inputs across these parameters will need to all be true for a device to fall into the group, though also note that each is optional. If you click the blue plus (+) symbol, you can add an **Or** statement for any given parameter to, for example, add two different domains or multiple tags to the same device group.

Once created, your device groups show as being ranked from **1** descending. The way this works logically is that **1** is the highest-ranked group, and devices only fall into the highest-ranked group that they match the logic for (even if they match the logic for multiple groups). This also means that a device can't be in more than one group. Your first thought might be that this is too restrictive, but consider that you can give a user group visibility to as many device groups as needed. Therefore, the goal with device groups will be to come up with the most granular or *atomic* groups required for your needs (don't forget that these are used for both automation and RBAC). If this still doesn't make complete sense, don't worry, we'll work through a practical example later in the chapter. Note that any devices not added to a custom device group will fall into a group called **Ungrouped devices (default)**.

Permissions available

Before defining your roles, you'll need to review the permissions available to understand your options. *Figure 5.3* shows what those options are. Note that roles are also in the M365D portal, located just above **Device groups**:

Add role

General Assigned user groups

Role name

MDE read only

Description

Describe the role

Permissions

☒ View Data

- ☒ Security operations
- ☒ Threat and vulnerability management

☐ Active remediation actions

- ☐ Security operations
- ☐ Threat and vulnerability management - Exception handling
- ☐ Threat and vulnerability management - Remediation handling
- ☐ Threat and vulnerability management - Application handling

☐ Threat and vulnerability management – Manage security baselines assessment profiles

☐ Alerts investigation

☐ Manage security settings in Security Center

☐ Manage endpoint security settings in Microsoft Endpoint Manager

☐ Live response capabilities

- ☒ Basic
- ☐ Advanced

Figure 5.3 – Role definition in MDE

View Data

First, we determine what data can be viewed by the role. Though it may expand in the future, at the time of writing, the **View Data** options are limited to **Security operations** and **Threat and vulnerability management**. The example in *Figure 5.3* could be a good configuration for an MDE reader role, giving read access to the full platform, to allow tertiary security teams visibility to review alerts for metrics, audits, or quality assurance.

Active remediation actions

Next, you define what remediation actions the role is able to take. The options here, and what they provide access to, are as follows:

- **Security operations:** This allows those assigned the role to take response actions within MDE. This includes dismissing or approving remediation actions from AIR and managing the **tenant allowed/blocked lists (TABL)** for both automation and indicators.
- **Threat and vulnerability management - Exception handling:** This allows those assigned this role to create and manage exceptions in TVM.
- **Threat and vulnerability management - Remediation handling:** This allows those assigned this role to submit and manage remediation requests, as well as create tickets within TVM.
- **Threat and vulnerability management - Application handling:** This allows the blocking and management of blocks of vulnerable applications.

Threat and vulnerability management - Manage security baselines assessment profiles

This setting is self-explanatory but also worth noting it is a pre-release feature at the time of writing, and the option may look different or not even exist yet in your portal.

Alerts investigation

Checking this box allows the role assignees to take any actions needed to investigate an alert. They can start automated investigations, run antivirus scans, collect investigation packages, download **portable executable (PE)** files, and manage tags.

Manage security settings in Security Center

Though you might assume this is more a security admin permission, it is very much a security operations permission. This allows the assignee to

manage the evaluation lab, email notifications, and automation folder exclusions, and to configure alert suppression settings.

Manage endpoint security settings in Microsoft Endpoint Manager

If connected to Intune, this setting allows the assignee to manage everything under the **Endpoint security** node within the Intune portal. The text of this setting will likely change soon, as Microsoft Endpoint Manager has been rebranded to simply Intune.

Live response capabilities

The **Basic** and **Advanced** options dictate what level of commands the assignee is allowed to run. Basic users can start a new LR session, perform read-only investigations, and download files from the device. Advanced users can perform all LR actions, including uploading scripts, viewing the script library, executing scripts, and downloading both PE and non-PE files from the file entity page within the portal.

Now that we know what permissions can be given and how to create device groups, let's further define our three-tier SOC, use geolocation as a division for device groups, and use both to create a practical example of what RBAC could look like.

A practical example of MDE RBAC

To get started, let's define a company to leverage for our example. Again, we'll expand on our three-tier SOC and our personas from earlier in the chapter.

Imagine a fictional business with its headquarters in France; we'll call it Graves Corporation. Over the years, Graves has acquired competitors and now has four other offices: one in Croatia, two in the United States (one on each coast), and one in the Netherlands. When acquired, Graves' other offices retained their local IT and security teams. The IT teams mostly just

provide simple, local, touch support. Each Graves site has a domain created within the primary Graves forest. The three-tiered SOC model was applied during acquisition and provides global consistency in security response and escalation procedures.

There is also a separate, global security admin team that focuses on vulnerability management and policy enforcement; both responsibilities are driven through engagement with IT to push updates and configurations to devices. That engaged IT admin team is also global, was created to drive consistency in device configurations, and is explicitly responsible for implementing security policies designed by the security admins. Both teams work out of the corporate headquarters and support the entire organization. They have beers at the weekends. They play video games together. They are friends... Be friends with your IT or security counterparts, it will make your life so much easier.

Creating device groups

To begin, let's think about our example's need for device groupings. In this case, the only differentiator is that there are different SOC teams per site and region. Laying that out, SOC geographic responsibilities look like this:

- **T1:** There are five T1 teams, each responsible for monitoring solely the devices at their different sites:
 - France
 - Croatia
 - Netherlands
 - US-east
 - US-west
- **T2:** There are two T2 teams, each responsible for all devices within a given region, as well as escalations from the relevant T1 SOC teams:
 - Europe
 - North America

- **T3:** There is one T3 team that supports and threat hunts across all regions and sites, that is Global

Don't forget we also have security and IT admin teams, but they both need full purview over the entire organization for their respective responsibilities. Remember from our description on device groups, you want to create your device groups at the most atomic level because devices can only exist in one group. In our example, the most atomic level is the site, so we will create the following device groups:

- **Devices:** France
- **Devices:** Croatia
- **Devices:** Netherlands
- **Devices:** US-east
- **Devices:** US-west

At this point, you can also decide what automation level you want for your devices. You can always come back and do it later, but if you're planning, it's a good idea to go ahead and get it implemented. If needed, refer to [*Chapter 4, Understanding Endpoint Detection and Response*](#), for a full description of automation levels. However, we will reiterate here that the confidence in the fidelity and response is high enough that Microsoft enables **Full - remediate threats automatically** by default in new tenants. So, for our example, we'll enable all five of our device groups for full automation, with the preceding names, and we'll target each domain separately by setting the **Domain** condition equal to the relevant domain name. With the infrastructure configuration we have, it's the easiest way to get the appropriate devices into each device group. With the minimum device groups created, it's time to define permissions.

Define and create admin roles

Though we described responsibilities earlier in our three-tiered SOC example, we need to define those responsibilities as explicit permissions

within MDE. Reviewing the permission options described previously, Graves landed on these permissions for each SOC tier:

- **T1:** As the lowest tier of the SOC, Graves security leadership wants their analysts to have the ability to investigate alerts. They also think they should be able to leverage LR to investigate the device's local filesystems and download files for further analysis as needed, but not upload or execute scripts from the library. To achieve this, a role is created in MDE called **SOC - T1 analyst**, which is given the following permissions:
 - **View Data: Security operations**
 - **Active remediation actions: Security operations**
 - **Alerts investigation**
 - **Live response capabilities: Basic**
- **T2:** As the next tier of the SOC, Graves security leadership believes that these analysts should be able to upload scripts to devices to automate investigation and mitigation tasks. To this end, a role is created in the M365D portal called **SOC - T2 analyst**, and given the same permissions as T1 analysts, but **Live response capabilities** is changed to **Advanced**.
- **T3:** As the top tier of the SOC, Graves security leadership believes that these analysts should, beyond investigative access, also have access to manage the evaluation lab, email notifications, and automation folder exclusions, and to configure alert suppression settings. A role is created called **SOC - T3 analyst** and given all the same permissions as T2, as well as **Manage security settings in Security Center**.

Again, don't forget our admins. Two more roles are created to cover their needs.

The first is given the role name **Security admin** and these permissions:

- **View Data:**
 - **Threat and vulnerability management**
- **Active remediation actions:**

- **Threat and vulnerability management - Exception handling**
- **Threat and vulnerability management - Remediation handling**
- **Threat and vulnerability management - Application handling**

This can obviously be divided between multiple teams if your organization has more granular vulnerability remediation responsibilities.

The second is given the role name **IT admin**, and this single permission to facilitate control from the Intune portal: **Manage endpoint security settings in Microsoft Endpoint Manager**.

This may seem strange. Why give the IT admin the responsibility for security configuration within the Intune portal? Remember that, at least for our personas, we have designated our security admins as the policy writers and our IT admins as the implementers. Make sure you fit your model to your needs.

Assign those roles and device groups

To finish our example, we'll need to get our roles and device groups assigned. To do so, we'll also quickly create some AAD groups that contain each of our teams respectively if they don't already exist. We recommend having a consistent naming convention wherever possible, both for easy identification and to avoid mistakes. For our example, we'll use these very self-explanatory AAD group names:

- **SOC-T1-France**
- **SOC-T1-Croatia**
- **SOC-T1-Netherlands**
- **SOC-T1-US-east**
- **SOC-T1-US-west**
- **SOC-T2-Europe**
- **SOC-T2-NorthAmerica**
- **SOC-T3**
- **SecurityAdmins**

- **ITAdmins**

Then, we'll assign our device groups to each group that requires visibility of those devices:

| Device group | Assigned AAD groups |
|----------------------|--|
| Devices: France | SOC-T1-France SOC-T2-Europe SOC-T3 SecurityAdmins ITAdmins |
| Devices: Croatia | SOC-T1-Croatia SOC-T2-Europe SOC-T3 SecurityAdmins ITAdmins |
| Devices: Netherlands | SOC-T1-Netherlands SOC-T2-Europe SOC-T3 SecurityAdmins ITAdmins |
| Devices: US-east | SOC-T1-US-east SOC-T2-NorthAmerica SOC-T3 SecurityAdmins ITAdmins |
| Devices: US-west | SOC-T1-US-west SOC-T2-NorthAmerica |

SOC-T3**SecurityAdmins****ITAdmins**

Table 5.1 – Device group assignments example

Now, all that's left to do is to assign our roles to the appropriate teams. Those assignments would look like this:

| Role | Assigned AAD groups |
|------------------|--|
| SOC - T1 analyst | SOC-T1-France SOC-T1-Croatia SOC-T1-Netherlands SOC-T1-US-east SOC-T1-US-west |
| SOC - T2 analyst | SOC-T2-Europe SOC-T2-NorthAmerica |
| SOC - T3 analyst | SOC-T3 |
| Security admin | SecurityAdmins |
| IT admin | ITAdmins |

Table 5.2 – Role assignment example

All done! This example may be basic on the surface, but we felt a practical walk-through could help you more clearly understand an approach, and with any luck, answer some questions we didn't even consider.

COLD SNACK

*Though further than we want to take the identity management within this text, you should consider **Privileged Identity Management (PIM)** as an additional layer in your RBAC model. The idea is to have analysts leverage PIM to gain access to things like LR without standing access to the capability, or have your admins do so prior to performing management tasks. This helps to greatly reduce the opportunity for abuse of compromised (admin) credentials.*

Creating a backout plan

No matter how much you plan, something can always go awry. It's for that reason that any good plan needs a solid backout plan – a way to revert all changes made and get back as close to the previous state as possible.

Fortunately, MDE makes the process easy. There are script-based, GP, and MDM options to offboard devices that can be downloaded from the portal. Just select the appropriate operating systems and deployment method. It's recommended that you don't use the script option for large-scale offboarding. Then, use the same configuration management tools you used for deployment to back it out.

COLD SNACK

Realize that deleting a device from AD, AAD, or Intune will not remove it from MDE. MDE is inherently independent of these management and identity platforms. Also note that offboarding a device from MDE will not delete existing data from the service. The data in the service will be retained until the retention period is reached (180 days from collection).

One lesser-known option for performing an offboard from MDE is by using the **application programming interface (API)**. You can use the API call referenced at <https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/offboard-machine-api> to offboard a single machine without any client-side interaction required. Best of all,

you can do so directly from the portal. Navigate to the **Partners and APIs** node in the M365D portal and select **API explorer**. This brings up a page where you can test different API calls against the platform.

At the top, you'll see the **Run Query** button, a dropdown for what type of **HTTP request** you're going to make, and a spot for the API endpoint URL (conveniently already populated with the first part of the URL).

Referencing the documentation, the call you need to make is an **HTTP POST** to this endpoint:

```
https://api.securitycenter.microsoft.com/api/machines/{id}/offboard
```

All we need to do is collect the MDE device ID. There's a cool option there since you're usually already looking at the device page for a device you want to offboard. Guess what? The MDE ID for the device is right in the URL for the device page (and all the subpages, such as the timeline, for example). It's found between the forward slashes just after the word **machines**. Copy this and insert it into the API path just as it's shown in the API documentation.

Let's do a test first to make sure we haven't made a mistake somewhere:

1. Take **offboard** off the end so that your URL looks like this:

```
https://api.securitycenter.microsoft.com/api/machines/{id}
```

2. Change the drop-down for the HTTP request type to **GET**.
3. Click **Run Query**.

This should return general information about the device and its status. Once you've confirmed that it is indeed the system you wanted to offboard, add **/offboard** back to the end of your URL. Change the HTTP request type to **POST**. Stop and note that the documentation says that a JSON comment is required in the request body. That may sound complicated, but just copy the syntax from the documentation and change the

comment to something relevant. In the end, your API request should look something like this, and then you can click **Run Query**:

API Explorer

Use the API explorer to test Microsoft Defender for Endpoint capabilities. Use the sample queries to get started.



Figure 5.4 – Using the API to offboard a device

The API can be great for automation or integrations, but that’s a bit beyond the scope of this book, so we won’t dig in any further. We wanted to provide this trick for offboarding single devices because it can be very useful in specific situations, and it gave us an excuse to talk about the API a bit.

With our discovery, remediation, and deployment well planned, let’s turn our focus to some particularly key considerations per feature.

Some key considerations per feature

Now that you have combined a solid understanding of how to plan your deployment with the primary capabilities of MDE that you’ve learned about in previous chapters, it’s time to begin framing your knowledge and bringing it further into focus in relation to your specific environment. What follows are some key considerations for each feature area that are especially valuable to consider during the early architecture and planning phases. A much more detailed view of operating systems and specifics will be provided in ***Chapter 6, Considerations for Deployment and Configuration***.

Adoption order

Most organizations these days have some sort of antivirus installed in their environment at a minimum. So, it's very likely that MDE will need to be migrated to or coexist alongside one or more non-Microsoft products. Though we've covered coexistence already to some extent (and will reiterate it in the feature-specific sections), we wanted to take a moment to discuss the recommended adoption order for migration. If, for whatever reason, you decide to adopt individual products separately or to take a slower approach, Microsoft recommends the following adoption order to maximize adoption success:

1. **Endpoint detection and response (EDR):** With built-in sensors on many Windows operating systems that only need to be activated, and cloud-based management, EDR is both easy to adopt and adds huge value (especially if you're coming from a non-EDR product)
2. **Threat and vulnerability management (TVM):** TVM is easily adopted right alongside EDR but requires coordination between IT and security admins to be truly effective. This means more planning and consideration for process flows, so it lands in the second spot.
3. **Next-generation protection (NGP):** Cloud-powered Microsoft Defender Antivirus is also built-in and integrates beautifully with EDR. This is only your third stop because you might need to have a solid migration plan outlined first.
4. **Attack surface reduction (ASR):** Ensuring configuration and creating attack resistance, ASR is next because prevention is key.
5. **Automated investigation and response (AIR):** You can enable AIR to handle the low-hanging fruit so your analysts can be freed up to focus on more complex threats. Keep in mind that this can really be adopted any time after, or in tandem with, EDR, but it's one that warrants a call out in case you avoided it initially.
6. **Defender Experts for Hunting:** Once your security organization has matured, consider the premium add-on where Microsoft threat experts will actively hunt in your environment and provide you with both insights and action plans.

That covers it for the adoption order. Now, let's move on to feature-specific considerations.

Next-generation protection

Though it may seem like a great idea to just enable everything in every instance, there are some common, highly-controlled circumstances where you may have a need for exceptions. For high-performance workloads in highly controlled circumstances, you may choose to forgo the real-time protection aspects. We all realize the importance of antimalware, so the core protection capabilities apply almost universally. Within the MDE family, there are specific benefits as the antimalware component is an integral part of the suite. This includes not just visibility but also reporting and response options – a better story when leveraging the full suite of prevention, detection, and response capabilities.

When it comes to next-generation protection, realistically, one of the first things you'll need to deal with is existing third-party products. In these situations, there are many things to consider, such as coexistence, in which we need to implement things in stages. In these situations, it is important to understand the *lift and shift* model (a term most used when referring to modernizing applications, but it fits here as well) as you want to minimize the gaps in coverage. The last thing you want during the roll-out of a new security stack is to create a weak security posture because of a lack of planning, or lack of understanding of how either the new or old product works with another similar product.

Staying on point with the coexistence rollout, let's assume we have a third-party antimalware in the environment and we're rolling Defender Antivirus out within the context of MDE. This means we're bringing it back to life as, in many cases, it's part of the operating system. Let's break down, as clearly as we can, what *bringing it back to life* means for each supported OS, starting with Server 2012 R2, and moving on to Server 2022 and Windows 11.

Server 2008 R2

Windows Server 2008 R2, like Windows 7, is still present in many environments. The key challenge with securing these operating systems is that it's not enough to run an antimalware EDR solution. Machines running these operating systems require an inordinate amount of investment, which is better directed toward modernization. Often referred to as technical debt, the risk is significant. Extended support plans can be a lifeline but a plan should be in place (and should have been in place since before 2020) to deal with the technical debt as soon as possible. Surely, if the machines are considered mission-critical, they should be prioritized.

COLD SNACK

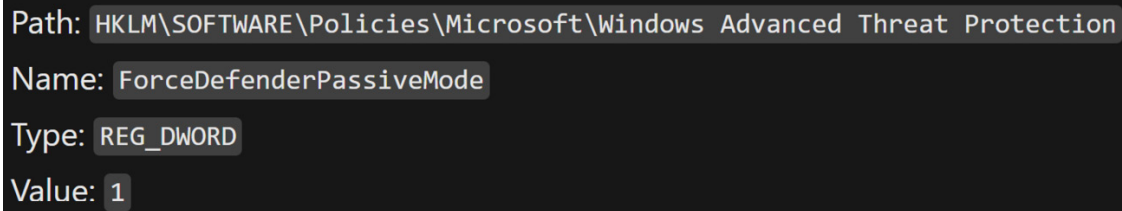
Extended support for Windows Server 2008/2008 R2 ended way back in January 2020. Extended security updates, which should not be confused with support, end in January 2023 (with a possible extension if you're running a machine on Azure). For 2012/2012 R2, extended support ends in October 2023 – migrating workloads from 2008 R2 to 2012 R2 is likely not a good long-term strategy.

*At the end of the day, paying because it is easier can very well lead to paying for an incident response team to come in and tell you how and why your business is down. That entry point is very likely one of your legacy systems. **Chapter 7**, *Managing and Maintaining the Security Posture*, talks about the importance of keeping up to date to prevent opportunistic attackers from leveraging your technical debt against you.*

Server 2012 R2

Defender Antivirus did not ship as part of this operating system, but we can change that now thanks to the release of the Unified Agent, discussed in **Chapter 2**, *Exploring Next-Generation Protection*. Since the Defender Antivirus bits are not there, they will need to be installed. Unlike on client operating systems, on Windows Server, Microsoft Defender Antivirus

doesn't enter **passive** mode automatically. So, if you're looking to coexist with a third-party AV solution, add the registry key shown in the following screenshot; that way Defender Antivirus will run in **passive** mode once the device onboards to MDE:



The screenshot shows a Windows Registry editor window with a dark background. The 'Path' field is set to 'HKLM\SOFTWARE\Policies\Microsoft\Windows Advanced Threat Protection'. The 'Name' field is 'ForceDefenderPassiveMode'. The 'Type' is 'REG_DWORD'. The 'Value' is '1'.

Figure 5.5 – Passive mode registry key

COLD SNACK

*While there are options to lift and shift from one product to another in one move, our experience in dealing with environments from a few thousand to a few hundred thousand, is that it is more practical to revitalize Defender Antivirus in **passive** mode and ensure that your feature enablement settings are properly in place before pulling the third-party antimalware. The last thing you want is to be left in a situation where your deployment failed, leaving the pre-existing product in a hampered state, and giving you essentially no protection at all.*

Let's look at the high-level steps to consider when preparing for deployment:

1. Create the **ForceDefenderPassiveMode** key with a value of **1**, as shown in *Figure 5.5*.
2. Install **md4ws.msi** (obtained from the onboarding section of the portal) with the **-passivemode** flag (which will install Defender Antivirus and modern EDR bits).
3. Onboard to MDE via the onboarding script.
4. Target configuration policies.
5. Ensure patches get applied regularly!

Server 2016

The Defender Antivirus feature exists in this OS and for many default installations, the feature was enabled by default; however third-party anti-malware typically forced it into a disabled state. For the longest time up until the unified agent, Defender Antivirus was unable to run in passive mode. This was due to the modern **Sense** agent not being present. With the unified agent, we can now leverage the **ForceDefenderPassiveMode** key. With that said, the preparation for Server 2016 can be broken down like this:

1. Create the **ForceDefenderPassiveMode** key with a value of **1**, as shown in *Figure 5.5*.
2. Enable or re-install Defender Antivirus and update the platform and engine to the latest versions. See [*Chapter 2, Exploring Next-Generation Protection*](#), if you need to read about the differences again.
3. Install **md4ws.msi** (obtained from the onboarding section of the portal) with the **-passivemode** flag (which installs EDR only and applies passive mode to the running Defender Antivirus).
4. Onboard to MDE via the onboarding script.
5. Target configuration policies.
6. Ensure patches get applied regularly!

Always ensure that the server is fully updated with the **Latest Cumulative Update (LCU)** and **Servicing Stack Update (SSU)** before installing the package.

Server 2019 and Server 2022

In both Server 2019 and Server 2022, Defender Antivirus is present in the operating system. Like Windows Server 2019, it ships as an optional feature. It's still possible to disable or uninstall Defender Antivirus on these operating systems, so like the steps for Server 2016, you'll follow suit to revive it:

1. Create the **ForceDefenderPassiveMode** key, as shown in *Figure 5.5*.

2. If applicable, enable or re-install Defender Antivirus and update the platform and engine to the latest versions.
3. Onboard to MDE via the onboarding script.
4. Target configuration policies.
5. Ensure patches get applied regularly!

COLD SNACK

*Extending the tamper protection feature to prevent abuse, from the September 2022 release of the Defender Antivirus platform on server OS, you can no longer toggle Antivirus into passive mode after onboarding when **tamper protection** is enabled. Either set the key before onboarding or temporarily disable **tamper protection** if you wish to transition to passive mode from active mode. The reverse is not true: if you remove the registry value or set it to 0, Defender Antivirus will toggle into active mode.*

Windows 10 and Windows 11

With Windows 10 and Windows 11, we have a slightly different approach when it comes to getting Defender Antivirus to run in **passive** mode. The difference here is that these two operating systems will natively move to **automatic disabled** mode when a third-party antimalware is detected. The detection is triggered when another AV registers itself with the Windows Security Center. Then, onboarding the device to MDE will automatically move Defender Antivirus to *passive* mode and out of *automatic disabled* mode.

While this book attempts to avoid sending you to public documentation, as the whole point is to dig deeper or show topics in a different light, some docs are just too good to ignore. The following gives you a very clear idea of what the expected outcomes are when it comes to third-party products and Defender Antivirus:

<https://aka.ms/MDAVCompat>

Attack surface reduction

When it comes to planning and preparation for ASR, the straightforward explanation is to leverage **audit** mode. ASR rules, network protection, and controlled folder access all have **audit** modes where you get to turn them on and see what the outcome would have been if they were to be set to their respective blocking modes.

While it is generally safe to set them all to **block** mode, with the exception being some ASR rules, it is always the right move to enable these things in phases. Upward of 30 days of auditing will provide a solid picture of what the impact on your environment could be. You can then proceed with an educated approach to enabling certain features across the environment.

One rule to be mindful of when it comes to ASR rules is the **Block process creations originating from PSEXEC and WMI commands** rule, especially when you are leveraging ConfigMgr. This is because the SCCM agent relies heavily on WMI for client actions. The new per-rule exclusions could potentially be leveraged and could exclude the SCCM agent.

***Chapter 8**, Establishing Security Operations*, will show some queries on how you can check for these audit events for these three features.

Note that not all ASR rules are supported on all operating systems; please look at ***Chapter 10**, Reference Guide, Tips, and Tricks*, for more details about this.

Endpoint detection and response

When it comes to planning or preparing for the EDR component of an MDE deployment, there are a few questions that should be answered first. Is this the only EDR component, or is the intent to co-exist with a third party here as well? While this is not advisable, as there is no way to predict how overlapping security solutions interact (and you will need both vendors to support the setup!), for a temporary setup, you would want to allow-list the MDE processes in the other product. Check ***Chapter***

10, *Reference Guide, Tips, and Tricks*, for a list of those processes. Note that in some situations when both security solutions depend on the same component, there is no path to coexistence, not even a temporary one.

If your security team needs alerts or events sent to a SIEM, you may need to consider how to plug in MDE. If it's Sentinel, the built-in connector makes this very easy. If you are a Splunk or other supported SIEM user, then the Microsoft 365 Defender API can be used for alert consumption needs – often, these SIEM solutions provide their own connector.

Other platforms

On non-Windows platforms, MDE is typically a more monolithic component with no clear separation between prevention and detection components. That said, for Linux and macOS, **passive** mode also exists. This allows you to only run the detection component and provide a possible transition path if you are already running another antimalware solution.

Linux

Extra attention should typically be given to Linux. Very often, production machines have been sized to run a specific workload, and adding a security solution comes with an unaccounted performance impact. That said, even correct sizing does not guarantee smooth sailing – Linux is notoriously diverse in combinations of distributions, kernels, packages, and libraries. It is *unregulated* as to what software is allowed to do, and there is a lot of freedom, so to speak. Not everyone has a standardized, controlled Linux environment, and Defender can only adapt to so many variables. Be prepared to fine-tune leveraging exclusions and spend some time fine-tuning for the best results.

macOS

If your Macs are on a recent version of macOS, there's not much you need to do to prepare for an MDE deployment – assuming you have device

management in place and are familiar with system extensions. If not, now is a good time to evaluate options. Both Intune and JAMF are popular options. However, like Windows client operating systems, any **mobile device management (MDM)** solution can be leveraged to deploy the app and its configuration.

iOS and Android

If your devices meet the requirements, the complexity here lies primarily in the fact that many organizations have **Bring Your Own Device (BYOD)** policies for mobile phones – and ultimately, the end user is in control of their personal devices. This comes with many caveats, pretty much all rooted in privacy concerns. MDE on Android and iOS comes with various options to limit the amount of management required, often working in tandem with access controls. For example, **AAD's** conditional access requires device compliance with company policies as opposed to strictly configuring the device. If you do not have a clear BYOD policy articulated in your organization, this is likely the biggest part of your implementation journey.

Summary

In this chapter, we discussed how to plan both your preparation and your MDE deployment. We talked about personas and how they can help you frame the key players in your project plan, how to take a gradual approach to roll things out, and which deployment method to choose based on your configuration management approach. We also took a deep dive into RBAC with a practical example to hopefully answer any lingering questions you had about the approach there. Most importantly, make sure you take the necessary time here. Planning always seems to drain folks, but the truth is, the more time and effort you put into planning, the more likely you are to succeed. Try to think of it as an interesting puzzle to solve and really lean into it. Your results will be better for it.

Again, many of the previous considerations are generic in nature, and chances are that you have tools and strategies already in place.

Operational excellence is important – not just to prepare for your MDE implementation, but as a driver for a more secure environment. Be prepared to put more work in after deployment – MDE should help you identify areas of significant improvement and offer a path to strengthen your overall security posture.

Wrapping up this chapter, it is time to prepare for production! Let's get those rollout plans squared away, your deployment rings, your collections, and your GPO deployments – whatever they may be, get excited! At this point, you should have your device management tooling ready to go and in tip-top shape. Patching should be in full effect and your devices should be nearing a fully patched state if not already patched up to the current release. You should be confident that devices around the environment have what they need to get out to the internet for proper communications and back to Defender backend services.

The next chapters will help you with your deployment and configuration, operationalizing, troubleshooting in case of issues, and finally, provide a reference guide to look up items that you need some quick clarification on. Let's get started!