

laan D'Souza-Wiltshire

Contributors

Yong Rhee, Chris Jackson, Amitai Rottem, Bhavna Soman

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it.

Copyright © 2019 Microsoft Corporation. All rights reserved.

Please refer to <u>Microsoft Trademarks</u> (https://aka.ms/MSTrademarks) for a list of trademarked products.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners

Contents

Disconnected scenario definitions	6
Considerations for a disconnected device security policy	8
Types of disconnected scenarios	9
Gatekeeping	14
Defense in depth	16
Protection technologies for disconnected device scenarios	16
Device lockdown and restriction	16
Offline updates and shared security locations	19
Application control and access	20
Conclusion	20

Disconnected devices

The use of disconnected devices among enterprise customers is a common scenario, and one that is becoming more and more important. For example, the rise in ransomware – often highly targeted to specific industries like healthcare – alongside state-sponsored actions designed to sabotage or steal information has highlighted the importance of protection against these threats across multiple device types and deployment scenarios.

The phrase 'disconnected devices' is itself host to a high level of complexity. Alongside the usual configuration concerns (including how patched (or unpatched) the devices are, and the operating system version), the devices themselves are spread across the deployment spectrum, from what their function is, how disconnected they are, to how they are managed. Therefore it becomes extremely hard to generalize the actual threat risk for any individual disconnected machine – or for any individual customer.

The top deployments that we hear about from customers with disconnected devices are in the public sector, defense, and manufacturing industries. We are also aware of customers with disconnected devices in critical services sectors – such as power and water.

By definition, most of these devices aren't regularly connected to the Internet – and therefore may or may not be intermittently connected to a management server. This has lead to the belief historically that these devices have a lower attack surface when compared to an Internet-connected PC.

However, ransomware and highly targeted attacks, as mentioned above, new and emerging physical-based attacks, and data theft, sabotage, and even traditional worms spreading through network shares and removable devices has caused a re-evaluation of the historical belief that these machines aren't at as high a risk as a connected device.

Microsoft Defender ATP, as part of the wider Microsoft Threat Protection umbrella, works together across multiple scenarios – and this includes disconnected devices. However, a defense in depth strategy, especially when applied to the multiple ways in which disconnected deployments occur, requires a strong understanding of how the technologies can work together.

It's important to note that machine learning – while it can be used in a disconnected environment – still benefits from regular updates to ensure the algorithms are updated with the

latest attack techniques seen in the wild. For this reason, we strongly recommend the use of a defense-in-depth strategy, even if you are using a dedicated offline machine-learning protection product.

While we invest heavily in machine learning, it should never be used as the sole or main layer of protection. Instead, it should be thought of as a last resort – if malware has managed to evade human, behavioral, and contextual analysis along with client-based attack surface reduction, exploit mitigation, and operating system hardening, then machine learning might be your best hope to catch and prevent the malware from spreading.

This is why Microsoft starts with cloud-delivered and client-based behavioral detection and protection, backed up by machine learning, human analysis, and cross-product telemetry to understand the entire malware lifecycle – both within the wider ecosystem and down to individual threats.

When Microsoft Defender ATP is connected to the cloud, intel can also be shared with other cloud-enabled machines. However, if a machine isn't connected, it still has client-based machine learning, behavioral analysis, heuristics, fileless detection, and process monitoring. This forms part of a defense-in-depth strategy that sees protection provided at the client level, even if there is no connection to a network or the Internet.

This whitepaper will describe some of the key characteristics of disconnected devices, how they require special attention from a security policy standpoint, and provide guidance to enterprise customers on how to use a defense in depth strategy to keep disconnected devices protected with technology in Windows 10.

Disconnected scenario definitions

With Microsoft Defender ATP are two main types of device protection – online (which uses our cloud-based service to analyze files and deliver security intelligence updates) and offline (which includes <u>attack surface reduction technology</u> to prevent threats from making changes at the device level, even if they aren't detected by our antivirus engine).

At a high level, the breakdown between technology can be summarized in Table 1: Network requirements for Microsoft Defender ATP technology. Note that technology that is listed as requiring Internet connectivity for some options generally refers to the ability to use the technology on the disconnected device for Internet-facing scenarios (such as visiting websites or using web browsers, restoring or resetting passwords, or obtaining reporting into usage).

Table 1: Network requirements for Microsoft Defender ATP technology

Protection technology	Requires Internet connectivity on the device for management, configuration, or usage
Attack surface reduction	
Hardware-based isolation	Not required
Bitlocker	Not required
Removable device control policies	Not required
Windows Defender System Guard (used to be known as Device Guard Kernel Mode Code Integrity (KMCI)).	Not required
Local Administrator Password Solution (LAPS)	Not required
Windows Defender Credential Guard	Not required

Windows Defender <u>Application Control</u> (used to be known as Configurable Code Integrity(CCI))	Required for some configuration options
Exploit protection	Not required
Network protection	Required
Controlled folder access	Not required
Attack surface reduction rules	Not required (some rules require Internet connectivity)
Powershell transcription	Not required
Next generation protection	
Windows Defender Antivirus, can run in a sandbox along with the VDI shared file feature	Required for some configuration options
Windows Defender SmartScreen	Required
IPsec rules and configuration alongside Windows Defender Firewall	Required for some configuration options
Endpoint detection and response	
All features	Required
Other features	
Windows Defender Application Control	Required for some configuration options
<u>Applocker</u>	Not required
Manually move WSUS configuration and files to ensure your operating system is kept up to date.	Not required

Certificate Revocation: Downloading or
storing the Certificate Trust List (CTL) files
on the virtual machine

Not required

With this breadth of technology, we've given you the customization that enables you to determine the right level of protection depending upon your actual deployments. For example, if you can't connect to the cloud for online protection, we have a broad and robust series of offline protection capabilities within the Attack surface reduction category.

If you have disconnected devices in your organization, you'll need to create a disconnect device security policy, which will define what technologies you need to apply, how they should be applied and kept up to date, and how future management of the device will impact (or be impacted by) security settings.

Furthermore, a strong disconnected device security policy should be one that relies on multiple technologies to provide a defense in depth strategy. No single technology (whether it's client-based machine learning or AI, signatures, heuristics, or manual analysis) should be relied on.

Considerations for a disconnected device security policy

This section will describe some of the high level considerations and tactics that can be used when defining a disconnected device security policy.

The first thing that should be considered is how updates impact security when delivering them to disconnected devices. In most deployments, a disconnected device is updated from an auxiliary device. The way in which those devices interact determine the general scope of updates, and this will inform how security should be considered within the realm of updates and connectivity between devices.

The following considerations will help you to understand the scope of update management between your devices:

- 1. Any auxiliary device that is connected to the disconnected device at any level of relationship (primary, secondary, tertiary) needs to be protected. This includes all devices that have any sort of relationship with the disconnected device.
- 2. Determine where or how updates get onto the disconnected device. Is it with a USB thumb drive? A management repository that can only connect to the disconnected

- device? Or is the device connected to a management console, such as Configuration Manager?
- 3. Next, consider how updates get onto the auxiliary device? Are they downloaded onto a PC, and then copied to the USB thumb drive? Are they distributed via Config Manager onto your management repo? What gates are involved and what other devices are connected to the auxiliary updating device?
- 4. Who uses the disconnected device? What admin privileges do they have?
- 5. Do users ever need to plug in removable devices? What's the core purpose of the device?

For example, if users never need to plug in removable devices, or only need to plug in a specific device that delivers updates, then removable device control will be a key configuration component. On the other hand, if you have disconnected devices that aren't connected to the Internet, but are managed by a local or disconnected Configuration Manager server, then you'll be need to be aware of how updates get onto the Configuration Manager server, and if there are opportunities for Internet access from other machines that might be connected to that server.

Types of disconnected scenarios

Due to the multiple ways in which disconnected deployments can occur, we've identified the three main categories that deployments can fall under, and provide our suggestions on which you should use, and how you can protect your devices.

The main categories that disconnected scenarios fall into are:

 Airgapped devices that are completely locked down and require physical interaction to apply updates (such as a USB device), as shown in Figure 1: Airgapped disconnected device. These can be semi-hybrid scenarios as well, where it might only connect to the network or Internet on an intermittent basis, or might only connect to the network but not the Internet, shown in Figure 2: Intermittently connected device.

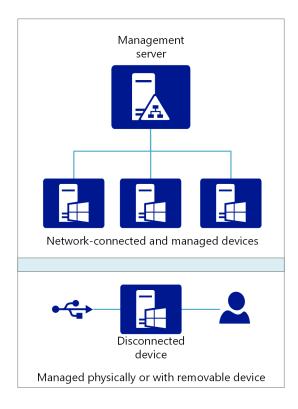


Figure 1: Airgapped disconnected device

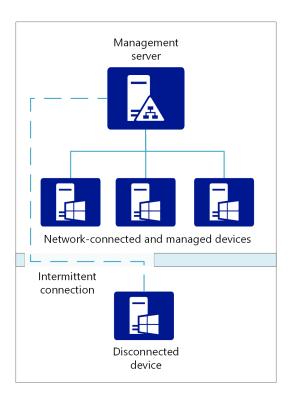


Figure 2: Intermittently connected device

Gated devices that are themselves not connected to a network, but are connected to a
device that is connected to the network (a gatekeeper). The gatekeeper then controls
what information can be sent to or from the Internet or the rest of the network. This is
demonstrated in Figure 3: Gatekeeper model for disconnected devices and Figure 5:
Gatekeeper architecture model.

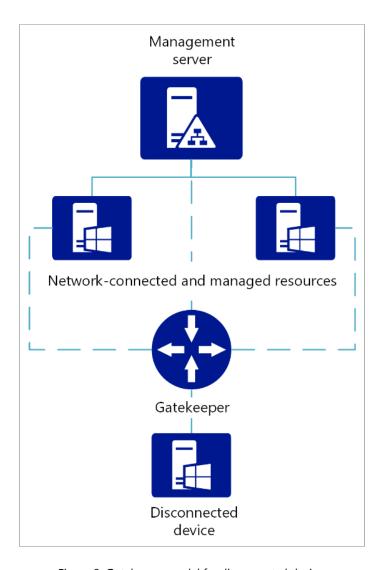


Figure 3: Gatekeeper model for disconnected devices

Pinholed devices that, for example, allow outbound connections only over SSL through a
proxy to a specific set of destinations, demonstrated in Figure 4: Pinholed disconnected
devices.

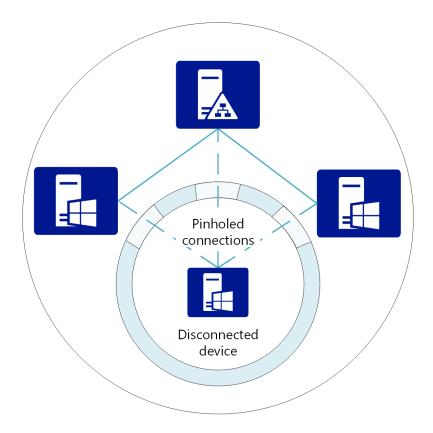


Figure 4: Pinholed disconnected devices

Out of all of these, we recommend you use some form of <u>gatekeeping</u> to validate the integrity of software that crosses the trust boundary to your disconnected devices.

Gatekeeping gives you the control over what goes in and out of the disconnected device, prevents it from contacting the Internet or other locations that could transmit malware, and still allows you to deliver timely updates.

Gatekeeping

The following image illustrates how a gatekeeper pattern might work – further on in this blog I dive into some ideas on how you can implement a pattern with existing Microsoft tech.

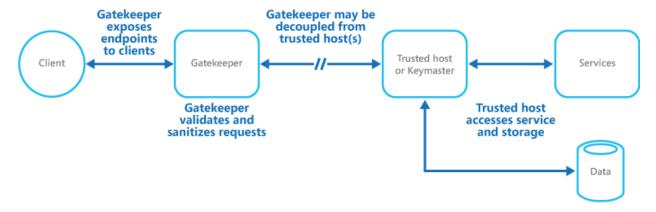


Figure 5: Gatekeeper architecture model

In this pattern, the client device is connected only to a dedicated machine, known as the gatekeeper. This could be a server or a standard Windows 10 client. It could also be an Azure Application Gateway, which can take requests from the machine, validate it, and then determine whether to pass the request to other resources (like update files).

The trusted host or keymaster is the owner of the data or services. In a full gatekeeper pattern as described in the illustration above, this would be where credentials are stored and confirmed.

However, in a disconnected device scenario the architecture needs to be flipped, in a way, such that the client is protected from any external or internal resources (as opposed to the internal resources being protected from the client and external resources). In which case the trusted host is more like a dedicated file share or management machine that has strict security policies in place to add an additional layer of prevention against threats.

Essentially, the 'disconnected' device is only connected to the gatekeeper, and has no direct interaction with any other resource, whether it's the Internet or the network.

The gatekeeper's function is to act as a guard and only allow through any requests that pass the rules.

This also means that the gatekeeper runs in a limited privilege mode. Its function is to review requests for data from the client. If it determines the request is permissible, then it passes that request to the trusted host, which then obtains the data.

The following is one way in which a gatekeeper model (using an Azure Application Gateway) could be used to only allow security intelligence updates to be downloaded onto the client device:

- 1. Create an Application Gateway with Azure
- 2. <u>Create and assign a virtual machine as a backend pool</u> to the application gateway. Ensure the VM is on a <u>virtual network that can connect to an external network</u>.
- 3. Enable WAF (or customize it) for the application gateway.
- 4. Configure Windows Defender Antivirus on the disconnected machine to <u>only take</u> <u>security intelligence updates from a UNC file path</u> or the <u>new shared security intelligence</u> <u>feature</u>. Use a path on the virtual machine you created as the backend pool.

Now when the semi-disconnected device attempts to obtain security intelligence updates from the virtual machine, it has to go through the application gateway.

You could also set up a public IP or internal IP that points to a machine that has the intelligence update files.

Defense in depth

Once you have set up your gatekeeper architecture, you should consider how you will lock down each of the devices.

At a minimum, you should have attack surface reduction technologies enabled on the client, gatekeeper, and any trusted hosts, in addition to an antimalware service. Windows Defender AV allows for offline loading (through shared security intelligence update locations) and can be set to perform scans as soon as new signatures are loaded.

Windows Defender AV also <u>performs real-time scanning</u> – identifying threats as soon as they are seen on the device. It doesn't require Internet connectivity to perform this and other behavioral detection activities.

You should also consider other technologies – such as firewalls, patching and inventory policies, drive encryption, application control, and web protection. This is where defense in depth is so important.

While most reporting features require Internet connectivity (such as those for the Microsoft Defender ATP console and Intune), some reporting can be achieved with network-connected management consoles (like Configuration Manager). You can also use similar methods to offloading intelligence updates and WSUS updates (as described in the Offline updates and shared security locations section), and <u>use Windows Event Forwarding to load events onto a separate machine</u>, from which your SIEM or event monitoring service can retrieve and create reports.

As an example, the following features could all be layered on both the client machine and the trusted host or resource machines.

Protection technologies for disconnected device scenarios

The remainder of this section details the technologies that can be used in addition to Windows Defender AV.

Device lockdown and restriction

BitLocker

<u>Use BitLocker and other data loss prevention</u> technology to control what information can be copied to or from the device.

BitLocker works at the hardware layer – it helps prevent physical theft of storage devices from their environments. You can use it on both main devices (workstations) and on removable drives (like USB thumb drives). If a malicious actor attempts to physically remove the drive, they'll be unable to access the data on the drive with the associated decryption keys – which you can store on a separate, offline device (such as a USB drive that is physically secured in a separate location).

You deploy BitLocker first by auditing your environment (which includes identifying the required hardware), and then enabling drive-level encryption on the devices you want to deploy to. You can manage the de-encryption policy in a fully offline scenario by keeping the recovery keys and information stored on separate devices and in separate locations.

If you need to recover a locked drive, you can use offline recovery methods to unlock or temporarily suspend the protection.

Windows Defender System Guard

<u>Windows Defender System Guard</u> also uses hardware-based configurations (such as the trusted platform module - TPM) to determine the validity of a request from a machine that is requesting resources.

It is enabled by default, although it can be further configured both within the Windows Security app locally on machines, and with management consoles including Configuration Manager and GPOs.

It uses a <u>management layer to understand if a device should be allowed to access resources</u>; the management system illustrated in Figure 6: Windows Defender System Guard resource request flow could be any management console – it doesn't have to be cloud based (for example, Configuration Manager or local Group Policy management).

WINDOWS DEFENDER SYSTEM GUARD

DEVICE HEALTH ATTESTATION

Management
system to gate
access based on device integrity and health

| Device Integrity Health State s (WD SG cloud service) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Windows Defender System Guard cloud service and management system
| Request | 3 | 4 | Approved |
| Here is my proof |
| Email | Wireless |
| Prove to me you are healthy |
| Access please |
| Wireless |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Important resources |
| File Servers |
| OneDrive |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Windows Defender System Guard cloud service and management system |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Windows Defender System Guard cloud service and management system |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Windows Defender System Guard cloud service and management system |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (Intune) |
| Client Policies (AV, Firewall, Patch state (In

Figure 6: Windows Defender System Guard resource request flow

USB and removable device control

<u>Use removable device control policies</u> to lock down the types of devices that can connect to the disconnected device – you can customize this to certain devices as well (<u>read more in our device control blog</u>).

The way this works is to prevent users from copying files to or from removable devices. You can specify the type, make, or model of the removable device, or you can lock out all removable devices. Combine this with BitLocker to create a secure infrastructure – remember that it's vital that all components of your disconnected deployment are secured.

Attack surface reduction

Use <u>advanced attack surface reduction features</u> which work directly on the device to prevent suspicious and unauthorized behaviors:

- Controlled folder access prevents modifications to files in certain folders you can customize the folders that are added so that very few folders can be modified or changed.
- Attack surface reduction rules prevent typical behaviors in common apps and frameworks (such as Office, macros, JavaScript, and more) that are exploited by malware from working.

• Exploit protection provides strict mitigation against known exploit behaviors.

All of these can be configured locally on the machine and don't require Internet connectivity to function.

Windows Defender Firewall

Use <u>IPsec rules and configuration alongside Windows Defender Firewall</u> to prevent access to untrusted network resources. This is valuable for semi-disconnected devices.

Offline updates and shared security locations

Shared security intelligence update location

<u>Utilize the VDI shared file feature</u> to allow your disconnected device to install security intelligence updates without having to download, unpack, and extract them.

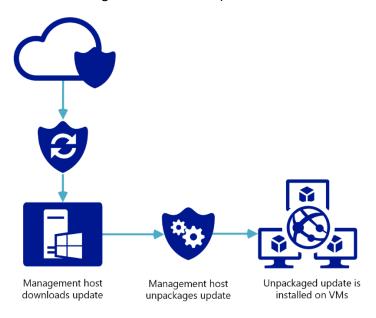


Figure 7: VMs with the shared security intelligence feature

Certificate trust lists

Ensure that certificate trust lists can be obtained. You could do this in a similar way to using the gatekeeper model to install security intelligence updates, by <u>downloading or storing the CTL</u> <u>files on the virtual machine</u> which can then be <u>requested by the disconnected machine</u>.

Windows Update configuration files and metadata

Manually <u>move WSUS configuration and files to ensure your operating system is kept up to date</u>. As with the security intelligence updates and CTL files, you could do this with the gatekeeper method.

Application control and access

Windows Defender Application Control

<u>Use Windows Defender Application Control</u> and <u>Applocker</u> to restrict the apps that are allowed to run on the device (it makes it significantly harder for malware executables to run if you block all unknown apps).

Application inventory and patching policies

You should have a clear patching policy for all the software on the devices. This includes maintaining an inventory of what is installed, and builds on using Windows Defender Application Control on locking down application installs to specific users or groups of users. Using Windows 10S, or optionally preventing installation of apps, would help here.

For example, create a recurring requirement that disconnected devices are reviewed for which apps are currently installed. You can use WD Application Control and AppLocker, or you can manually identify which apps have been installed. Note, however, that malicious apps often hide themselves and can't easily be found – which is why having antimalware protection (furnished via the shared security intelligence location) is key.

Regularly auditing the app inventory augments these two suggestions and helps identify irregular app installation either from unauthorized individuals or malware.

Conclusion

There is no one single golden path for how to best protect disconnected devices, and no single technology will provide adequate protection.

Multiple protection capabilities must be used together, in conjunction with strong access control, application installation and usage, and regularly inventorying of assets, access, and usage by employees.

If you'd like to share feedback about disconnected devices in your environment, please contact your Microsoft sales representative. Teams within Microsoft regularly meet to discuss customer needs and requests – across the entire product stack, and we are committed to providing the best security protection for all of our customers.