Dane Stuckey Follow
DFIR @ Palantir | Blue Team | Windows Security Fanboy | All views are my own, not my employer
Apr 22 · 16 min read

Endpoint Isolation with the Windows Firewall

Over the last few weeks, I've had conversations with several individuals around mitigating lateral movement in a Windows environment. In all of these cases, I was surprised to learn that these defenders were not using the native Windows Firewall as one of their defense-in-depth layers. This was curious to me as the firewall is both present by default and is one of the easiest ways to limit remote access to many commonly-abused services.

This post is going to focus on using the Windows Firewall for isolating and securing endpoints in an Active Directory environment. The goal is to limit the potential attack surface of endpoints by limiting access to potentially exploitable services, require IPSEC authentication to management services, and provide additional encryption to services which rely on plaintext or weak ciphers. The configurations listed in this post should be immediately deployable in a production environment and ultimate make our adversary's lateral movement attempts that much more frustrating.

An important note: you should go watch Jessica Payne's 'Demystifying the Windows Firewall' talk from Ignite 2016. Most of the content in this post is simply a re-hash of the best practices and strategies that she has outlined in her presentation. Her talk is *the* reference for the Windows Firewall.

Seriously, go watch it right now.

Firewall Principles

There are some basic principles to adhere to when developing a comprehensive firewall policy:

- **Manage centrally**. We're going to rely on using Group Policy Objects (GPO) for managing our firewall rules. This has the benefit of native integration within Active Directory and, if using Advanced Group Policy Management (AGPM), change control, rollback, and auditing features.
- **Block by default.** We're going to block services by default. We will whitelist critical services we need with appropriate scope. Everything else can get tossed to the void.
- **Require authentication.** We're going to challenge remote devices and users to authenticate before they can communicate to our critical services. If they fail authentication, they get tossed to the void.
- **Encrypt plaintext protocols**. We're going to apply encryption and integrity protections on plaintext protocols.
- **Use strong cryptographic settings**. We're going to use strong ciphers, key sizes, and protocols when establishing IPSec tunnels.
- **Ignore local policies.** By default, any user or application can create local firewall rules. To ensure our ruleset isn't accidentally or intentionally subverted, we're going to deny local rules from applying in the firewall.

Firewall Base Settings (Endpoints)

From these principles, we can now start the process of building our firewall settings. These will configure how the firewall operates, determine logging settings, and set authentication and encryption options for IPSEC.

To start, we'll open up the **Group Policy Management** editor (gpmc.msc) and make a new GPO called **Workstation Firewall Policy.**

Scope this to the OU containing your workstations of choice **but do not apply it yet** (remove "Authenticated Users" from the security filter)**.**

In the GPO manager, navigate to **Computer Configuration > Windows Settings > Security Settings > Windows Defender Firewall with Advanced Security**

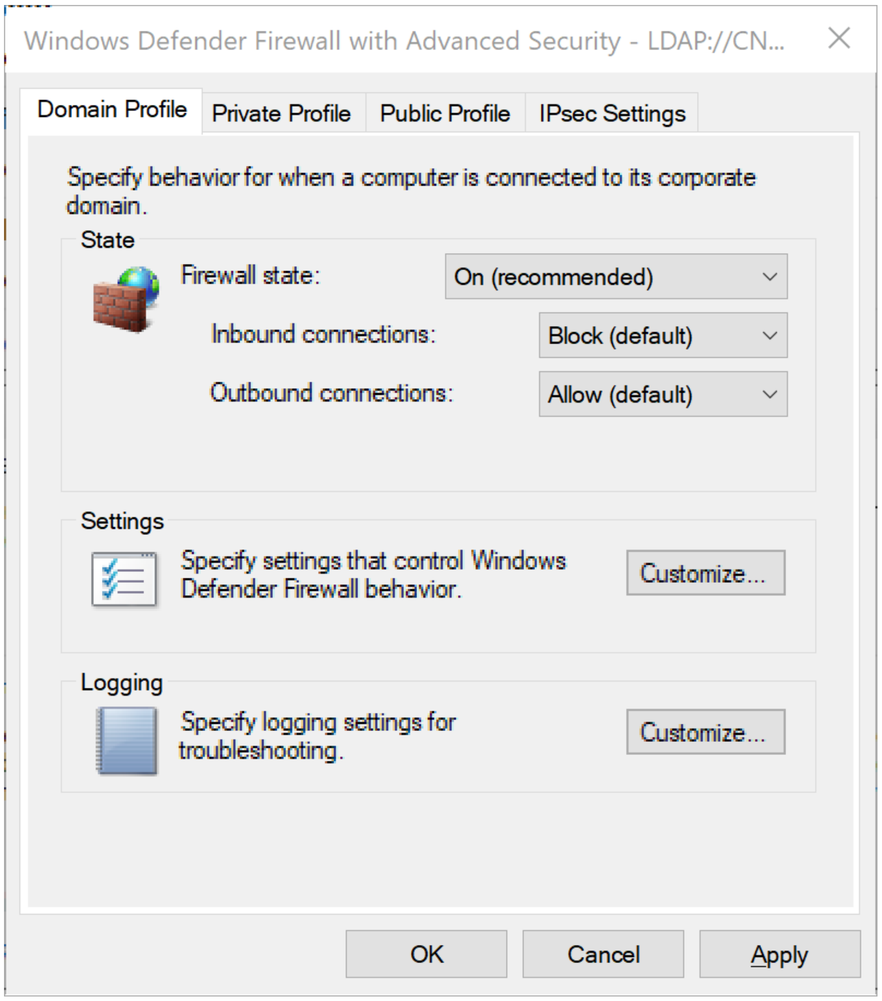
Configuring Firewall Settings

The first option we'll set is to enforce the Firewall to be enabled on all profiles. Navigate to each of the Profile Tabs and set the following settings:

Firewall State: On
Inbound Connections: Block
Outbound Connections: Allow

Settings:
Display a Notification: No
Allow Unicast Response: Yes
Apply Local Firewall Rules: No
Apply Local Connection Security Rules: No

Logging:
Name: <Default>
Size Limit: 4096 (or higher—dealer's choice)
Log Dropped Packets: Yes
Log Successful Connections: No


Windows Defender Firewall with Advanced Security - LDAP://CN...

Specify behavior for when a computer is connected to its corporate domain.

State

Firewall state: On (recommended)

Inbound connections: Block (default)

Outbound connections: Allow (default)

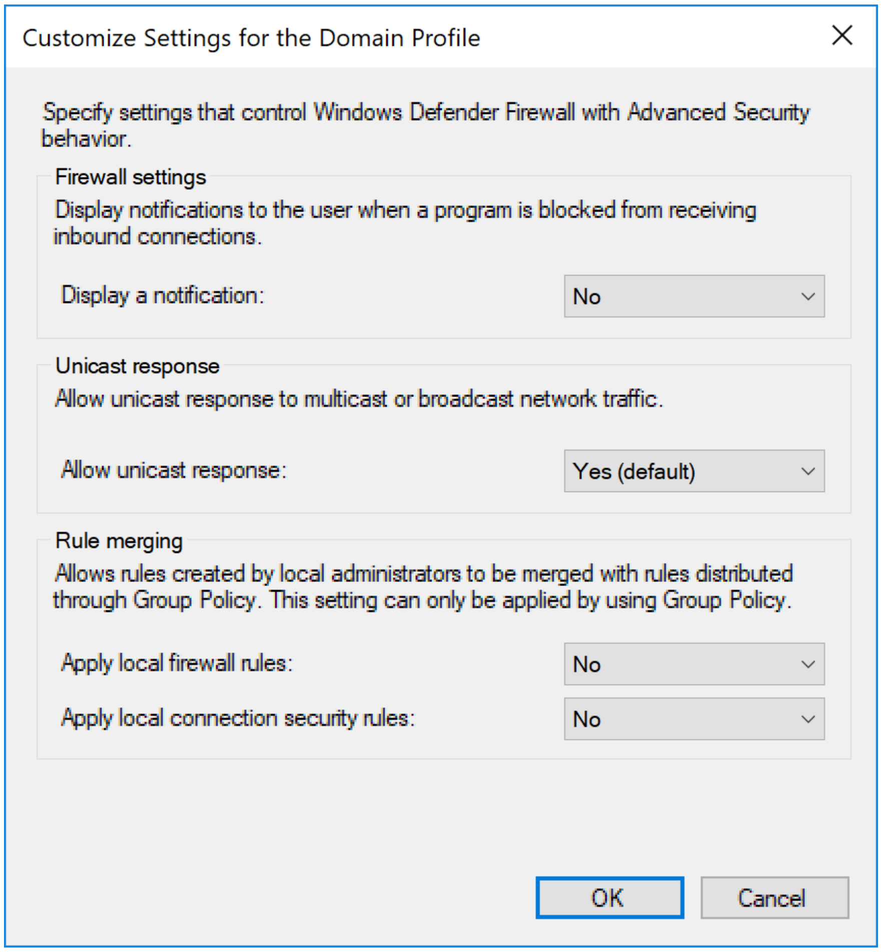
Settings

Specify settings that control Windows Defender Firewall behavior.

Logging

Specify logging settings for troubleshooting.

Desired Firewall Settings



Customize Settings for the Domain Profile

Specify settings that control Windows Defender Firewall with Advanced Security behavior.

Firewall settings

Display notifications to the user when a program is blocked from receiving inbound connections.

Display a notification: No

Unicast response

Allow unicast response to multicast or broadcast network traffic.

Allow unicast response: Yes (default)

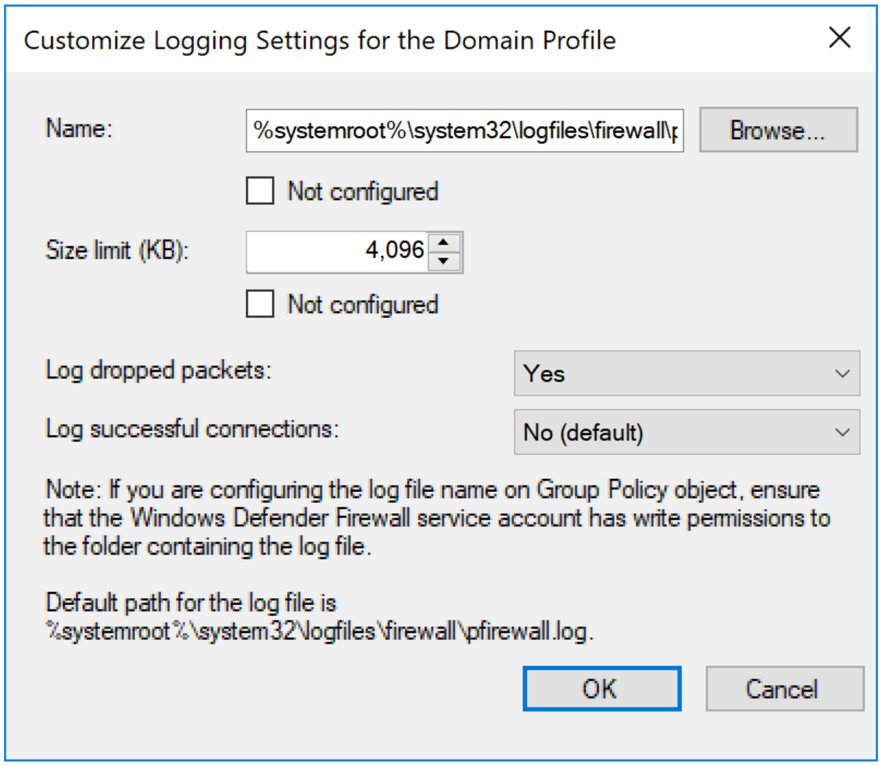
Rule merging

Allows rules created by local administrators to be merged with rules distributed through Group Policy. This setting can only be applied by using Group Policy.

Apply local firewall rules: No

Apply local connection security rules: No

Desired Firewall Settings.



Customize Logging Settings for the Domain Profile

Name: %systemroot%\system32\logfiles\firewall\p

Browse...

☐ Not configured

Size limit (KB): 4,096

☐ Not configured

Log dropped packets: Yes

Log successful connections: No (default)

Note: If you are configuring the log file name on Group Policy object, ensure that the Windows Defender Firewall service account has write permissions to the folder containing the log file.

Default path for the log file is %systemroot%\system32\logfiles\firewall\pfirewall.log.

OK Cancel

Desired Firewall Settings.

Configuring IPSEC Settings

The second option we'll set are the IPSEC settings. Navigate to the IPSEC Settings Tab and set the following settings:

> ***Note: IPSEC ciphers, key sizes, and exchange protocols may need to be tailored to your environment and needs. These selections assume a modern fleet with strong security requirements. Avoid DES and MD5 in any configuration.***

Key Exchange Main Mode: Advanced
Security Methods:

- SHA-384-AES-CBC-256-Elliptic Curve Diffie-Hellman-P-384
- SHA-384-AES-CBC-192-Elliptic Curve Diffie-Hellman-P-384
- SHA-384-AES-CBC-128-Elliptic Curve Diffie-Hellman-P-384
- SHA-256-AES-CBC-256-Elliptic Curve Diffie-Hellman-P-256
- SHA-256-AES-CBC-192-Elliptic Curve Diffie-Hellman-P-256

- SHA-256-AES-CBC-128-Elliptic Curve Diffie-Hellman-P-256

Key Lifespan: 480 minutes
Use Diffie-Hellman: Yes

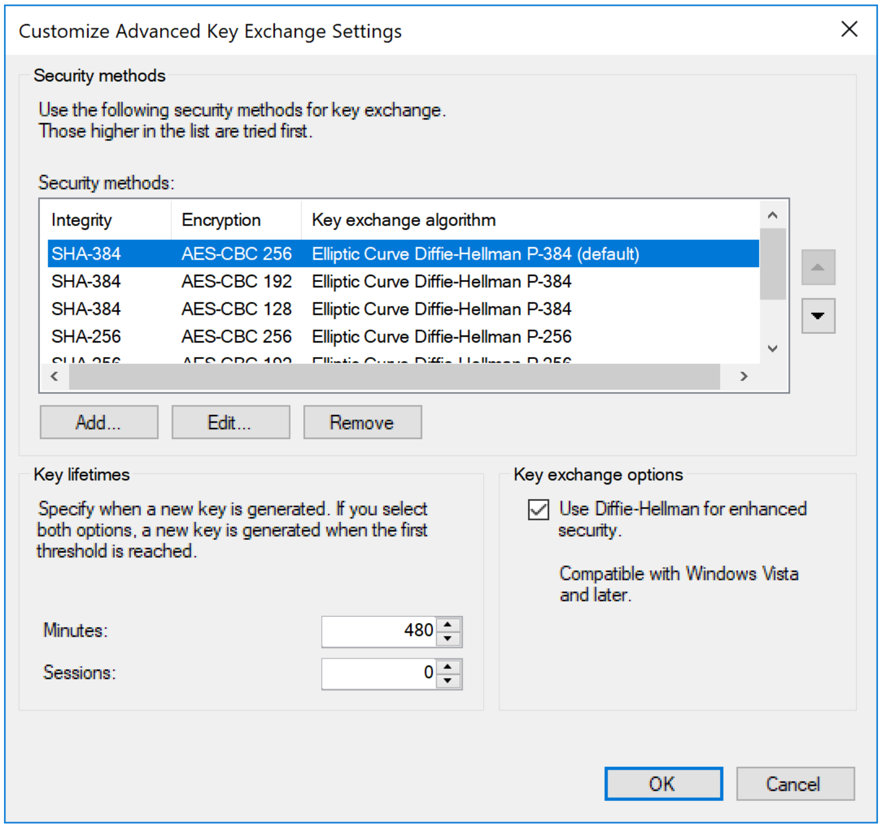
Data Protection Quick Mode: Advanced
Security Methods (Data Integrity):

- ESP-AES-GMAC-256 (60/100,000)
- ESP-AES-GMAC-192 (60/100,000)
- ESP-AES-GMAC-128 (60/100,000)

Security Methods (Data Integrity/Encryption):

- ESP-AES-GCM-256 (60/100,000)
- ESP-AES-GCM-192 (60/100,000)
- ESP-AES-GCM-128 (60/100,000)

Authentication Method: Computer and Kerberos (v5)



Customize Advanced Key Exchange Settings

Security methods

Use the following security methods for key exchange.
Those higher in the list are tried first.

Security methods:

Integrity	Encryption	Key exchange algorithm
SHA-384	AES-CBC 256	Elliptic Curve Diffie-Hellman P-384 (default)
SHA-384	AES-CBC 192	Elliptic Curve Diffie-Hellman P-384
SHA-384	AES-CBC 128	Elliptic Curve Diffie-Hellman P-384
SHA-256	AES-CBC 256	Elliptic Curve Diffie-Hellman P-256

Key lifetimes

Specify when a new key is generated. If you select both options, a new key is generated when the first threshold is reached.

Minutes: 480

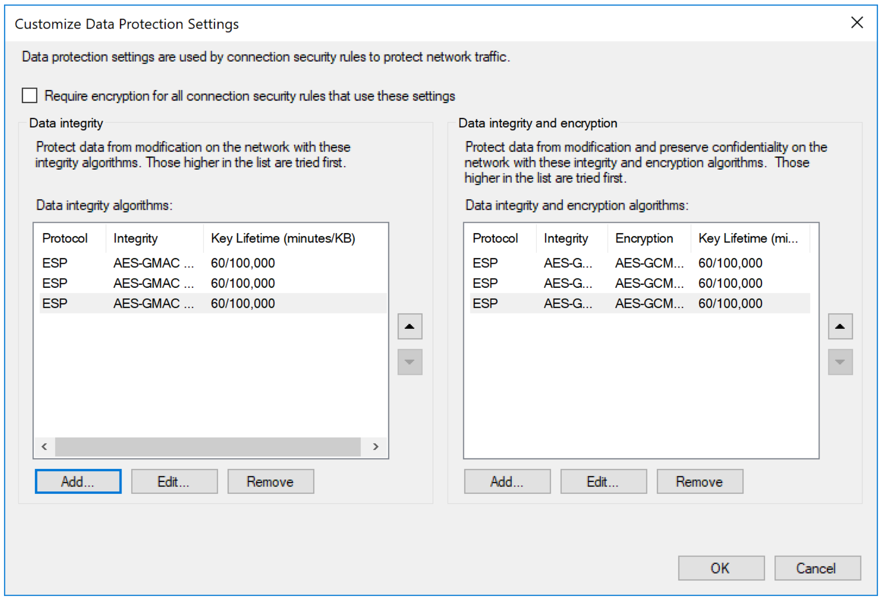
Sessions: 0

Key exchange options

[✓] Use Diffie-Hellman for enhanced security.

Compatible with Windows Vista and later.

Desired IPSec Key Exchange Settings.



Customize Data Protection Settings

Data protection settings are used by connection security rules to protect network traffic.

[ ] Require encryption for all connection security rules that use these settings

Data integrity

Protect data from modification on the network with these integrity algorithms. Those higher in the list are tried first.

Data integrity algorithms:

Protocol	Integrity	Key Lifetime (minutes/KB)
ESP	AES-GMAC ...	60/100,000
ESP	AES-GMAC ...	60/100,000
ESP	AES-GMAC ...	60/100,000

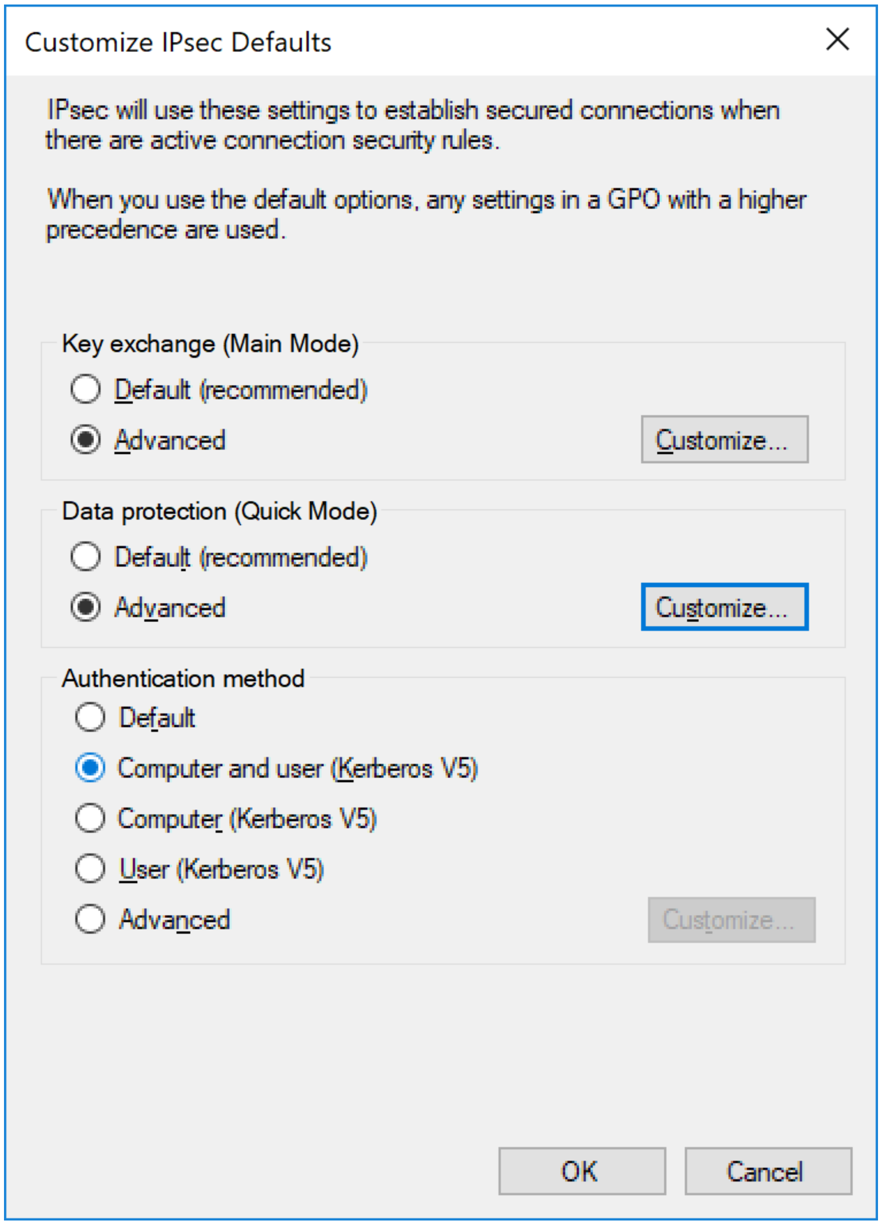
Data integrity and encryption

Protect data from modification and preserve confidentiality on the network with these integrity and encryption algorithms. Those higher in the list are tried first.

Data integrity and encryption algorithms:

Protocol	Integrity	Encryption	Key Lifetime (mi...
ESP	AES-G...	AES-GCM...	60/100,000
ESP	AES-G...	AES-GCM...	60/100,000
ESP	AES-G...	AES-GCM...	60/100,000

Desired IPSec Quick Mode Settings.


Customize IPsec Defaults

IPsec will use these settings to establish secured connections when there are active connection security rules.

When you use the default options, any settings in a GPO with a higher precedence are used.

Key exchange (Main Mode)
- ○ Default (recommended)
- ● Advanced — Customize...

Data protection (Quick Mode)
- ○ Default (recommended)
- ● Advanced — Customize...

Authentication method
- ○ Default
- ● Computer and user (Kerberos V5)
- ○ Computer (Kerberos V5)
- ○ User (Kerberos V5)
- ○ Advanced — Customize...

Desired IPSec Settings.

At this stage, we've gone ahead and configured the base policies for the firewall. This has the following effects:

- The firewall will be enabled on all profiles.
- The firewall will ignore locally applied rules.
- The firewall will log dropped packets for debugging purposes.
- Local firewall rules will not be applied.
- IPSec is configured to use strong ciphers, keys, and protocols.

- IPSec authentication will use kerberos for user and computer accounts.

Firewall Connection Security Rules (Endpoints)

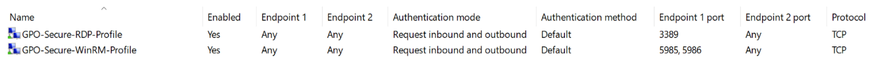
We will now start the process of implementing connection security rules for the endpoints. Connection security rules allow for complex management of IPSEC tunnels to include authentication from users or computers prior to establishment of sessions.

In essence, we will use these rules to require both computers on each end of a given session to establish an authenticated (and optionally: encrypted) session to protect critical services. These rules do not deal with directionality or access controls, so they will be used in conjunction with standard firewall rules later in this post.

> ***Note: As you develop new rules, you should deploy to a subset of systems using the security filter. Failure to adequately test firewall rules could cause a loss of availability for critical systems.***

In the GPO editor, open up the **Workstations Firewall Policy** and navigate to **Computer Configuration > Windows Settings > Security Settings > Windows Defender Firewall with Advanced Security > Connection Security Rules.**

We'll use a standard naming scheme for our rules to make it easier to understand and troubleshoot rule issues:
GPO-{Secure}-{Service}-Profile



Name	Enabled	Endpoint 1	Endpoint 2	Authentication mode	Authentication method	Endpoint 1 port	Endpoint 2 port	Protocol
GPO-Secure-RDP-Profile	Yes	Any	Any	Request inbound and outbound	Default	3389	Any	TCP
GPO-Secure-WinRM-Profile	Yes	Any	Any	Request inbound and outbound	Default	5985, 5986	Any	TCP

Connection Security Rules.

Securing WinRM

The first rule we'll implement will secure WinRM by requiring successful kerberos authentication from trusted computers and users. This is designed to gate access to WinRM only from trusted machines on the network, such as Windows Event Collectors, bastion hosts, and other management devices. This additionally mitigates the risk of exploits targeting WinRM and associated services.

> ***Note: As you develop new rules, you should use the "Request Authentication for Inbound and Outbound Settings" option, as it will gracefully fall back for troubleshooting purposes. This should be changed to "Require Authentication" once it is stable.***

Implement this rule with the following details:

Rule Type: Custom
Endpoint 1: Any
Endpoint 2: Any
Requirements: Request Authentication for Inbound and Outbound Settings
Authentication Method: Default (User/Computer Kerberos v5)
Protocol Type: TCP
Endpoint 1 Port: 5985, 5986
Endpoint 2 Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Profile: All
Name: GPO-Secure-WinRM-Policy

Securing RDP

The next rule we'll implement will secure RDP by requiring successful kerberos authentication from trusted computers and users. This is designed to gate access to RDP only from trusted machines on the network, such as help-desk machines, bastion hosts, and other management devices. This additionally mitigates the risk of exploits targeting RDP and associated services.

> ***Note: As you develop new rules, you should use the "Request Authentication for Inbound and Outbound Settings" option, as it will gracefully fall back for troubleshooting purposes. This should be changed to "Require Authentication" once it is stable.***

Implement this rule with the following details:

Rule Type: Custom
Endpoint 1: Any
Endpoint 2: Any
Requirements: Request Authentication for Inbound and Outbound Settings
Authentication Method: Default (User/Computer Kerberos v5)
Protocol Type: TCP
Endpoint 1 Port: 3389
Endpoint 2 Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Profile: All
Name: GPO-Secure-RDP-Policy

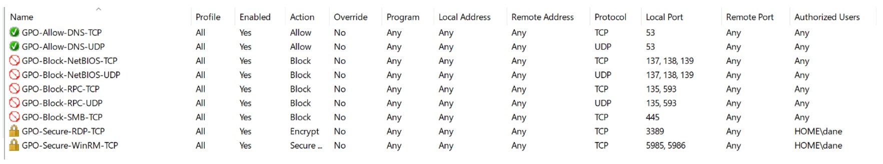
Firewall Rules (Endpoints)

We will now start the process of implementing firewall rules for the endpoints.

> ***Note: As you develop new rules, you should deploy to a subset of systems using the security filter. Failure to adequately test firewall rules could cause a loss of availability for critical systems.***

In the GPO editor, open up the **Workstations Firewall Policy** and navigate to **Computer Configuration** > **Windows Settings** > **Security Settings** > **Windows Defender Firewall with Advanced Security** > **Inbound Rules.**

We'll use a standard naming scheme for our rules to make it easier to understand and troubleshoot rule issues:
GPO-{Allow,Block,Secure}-{Service}-{Protocol}



Name	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port	Authorized Users
GPO-Allow-DNS-TCP	All	Yes	Allow	No	Any	Any	Any	TCP	53	Any	Any
GPO-Allow-DNS-UDP	All	Yes	Allow	No	Any	Any	Any	UDP	53	Any	Any
GPO-Block-NetBIOS-TCP	All	Yes	Block	No	Any	Any	Any	TCP	137, 138, 139	Any	Any
GPO-Block-NetBIOS-UDP	All	Yes	Block	No	Any	Any	Any	UDP	137, 138, 139	Any	Any
GPO-Block-RPC-TCP	All	Yes	Block	No	Any	Any	Any	TCP	135, 593	Any	Any
GPO-Block-RPC-UDP	All	Yes	Block	No	Any	Any	Any	UDP	135, 593	Any	Any
GPO-Block-SMB-TCP	All	Yes	Block	No	Any	Any	Any	TCP	445	Any	Any
GPO-Secure-RDP-TCP	All	Yes	Encrypt	No	Any	Any	Any	TCP	3389	Any	HOME\dane
GPO-Secure-WinRM-TCP	All	Yes	Secure ...	No	Any	Any	Any	TCP	5985, 5986	Any	HOME\dane

Example of My Firewall Rules.

Blocking SMB / Remote Named Pipes

The first rule we'll implement will block incoming Server Message Block (SMB) connections. This is designed to prevent remote access to file shares on the workstation, as well as mitigate the risk of exploits targeting SMB and associated services.

Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: TCP
Local Port: 445
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block the Connection
Profile: All
Name: GPO-Block-SMB-TCP

Blocking NetBIOS/NBT

The next rule we'll implement will block incoming NetBIOS and NetBIOS over TCP (NBT) connections. As NetBIOS is not required in a modern environment, this rule will mitigate the risk of exploits targeting NetBIOS and associated services.

Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: TCP
Local Port: 137, 138, 139
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block the Connection
Profile: All
Name: GPO-Block-NetBIOS-TCP

We'll repeat this rule, but for UDP. Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: UDP
Local Port: 137, 138, 139
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block the Connection
Profile: All
Name: GPO-Block-NetBIOS-UDP

Blocking RPC/DCOM/WMI

The next rule we'll implement will block incoming Remote Procedure Call (RPC) connections. RPC allows access to the Distributed COM (DCOM) service, the WMI service, and many other windows services. This rule will mitigate the risk of exploits targeting RPC and associated services.

Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: TCP
Local Port: 135, 593
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block the Connection
Profile: All
Name: GPO-Block-RPC-TCP

We'll repeat this rule, but for UDP. Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: UDP
Local Port: 135, 593
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address

Action: Block the Connection
Profile: All
Name: GPO-Block-RPC-UDP

Securing WinRM

The next rule we'll implement will secure incoming WinRM connections with an authenticated IPSEC tunnel using the security connection rule we previously built.

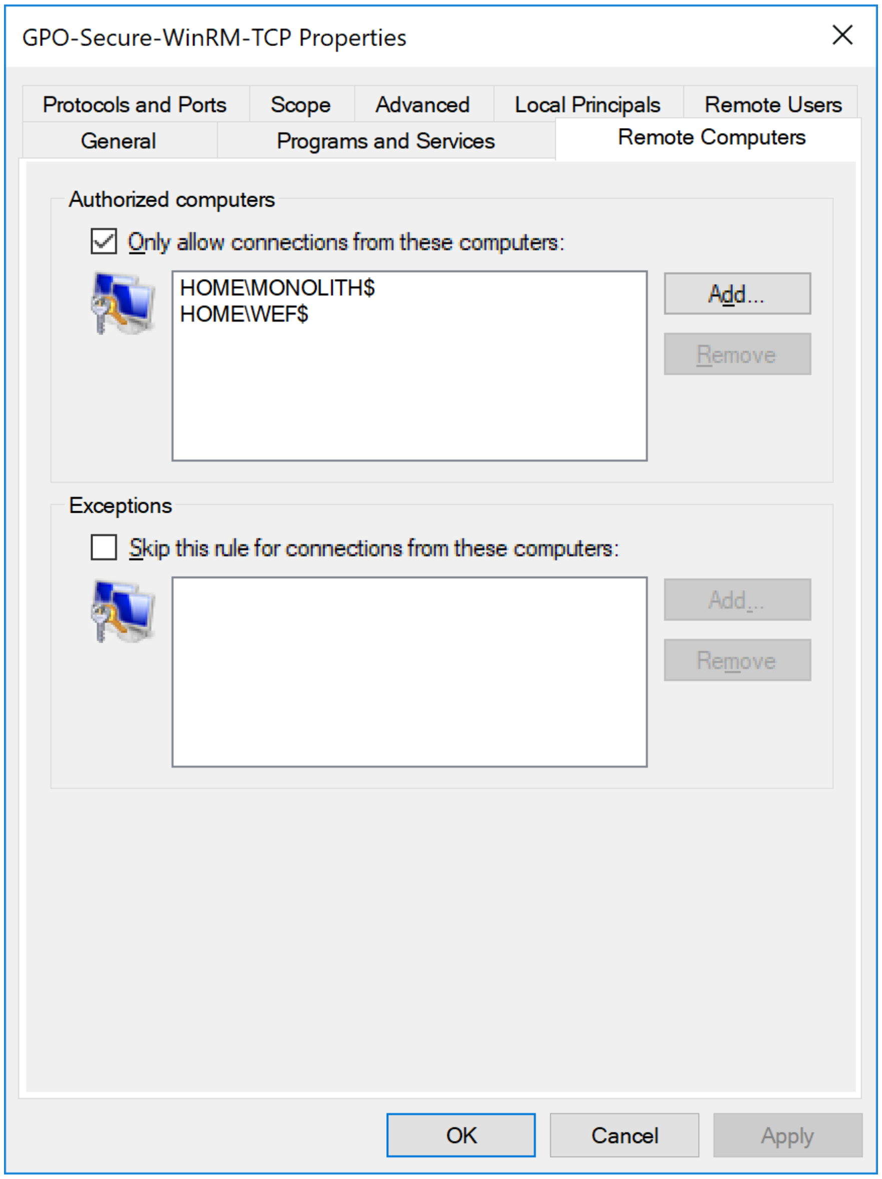
Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: TCP
Local Port: 5985, 5986
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Allow if Secure
Profile: All
Name: GPO-Secure-WinRM-TCP

Then, **right-click** on the rule and navigate to **properties**. Select the **Remote Computers** tab. This allows you to specify which computers in your Active Directory environment are allowed to connect to this machine remotely.

In this example, I am going to allow my desktop **monolith** to connect remotely via WinRM to any device in my network via WinRM. I am also going to allow my Windows Event Collector machine **WEF** to connect for event forwarding purposes.

You can substitute this for any privileged access workstations (PAWs), bastion hosts, Windows Event Collectors, or other management hosts in your fleet.


GPO-Secure-WinRM-TCP Properties

Authorized computers

☑ Only allow connections from these computers:

HOME\MONOLITH$
HOME\WEF$

Exceptions

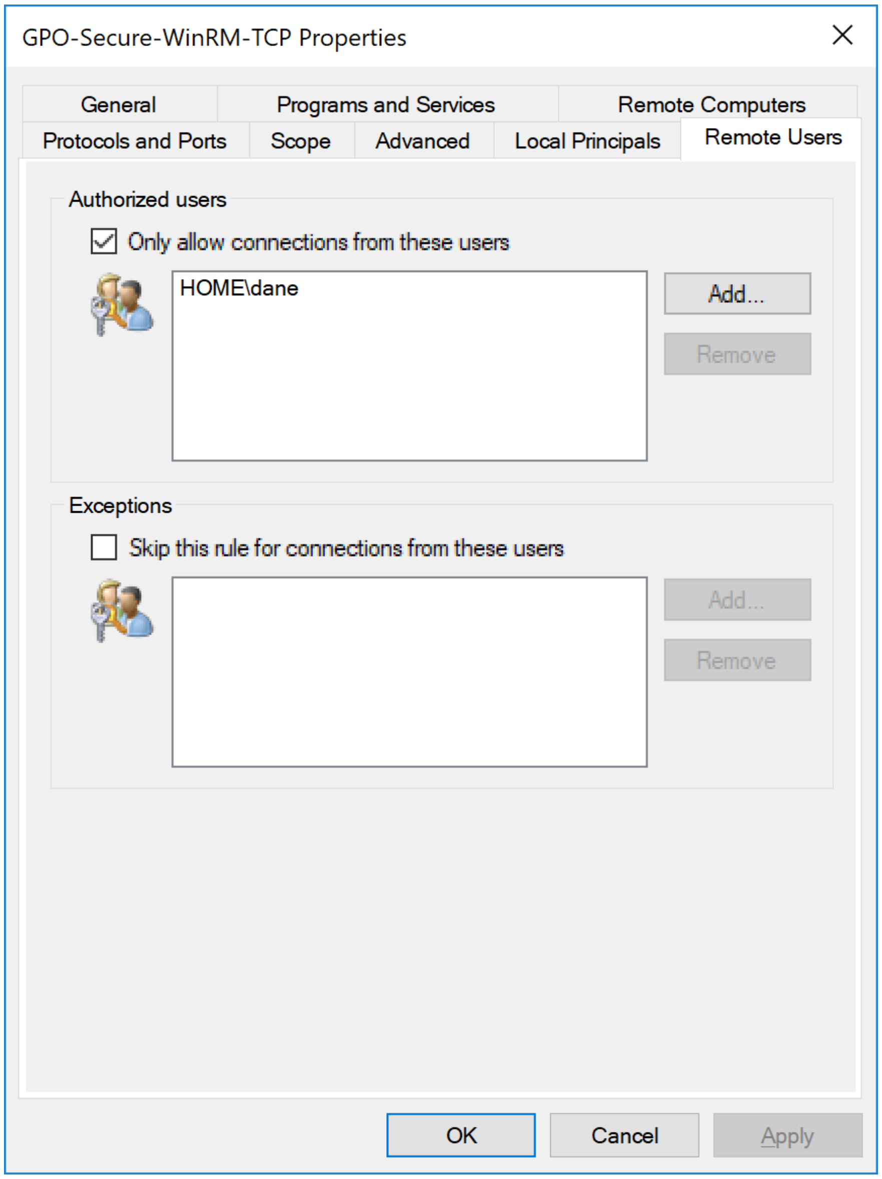
☐ Skip this rule for connections from these computers:

Remote Computer Requirements.

Next, we'll specify legitimate remote users. Select the **Remote User** tab. This allows you to specify which users in your Active Directory environment are allowed to connect to this machine remotely.

In this example, I am only going to allow myself to connect remotely via WinRM.

You can substitute this for any user or user group in your fleet. These could include help-desk, workstation administrators, or other user types.


GPO-Secure-WinRM-TCP Properties

Authorized users

☑ Only allow connections from these users

HOME\dane

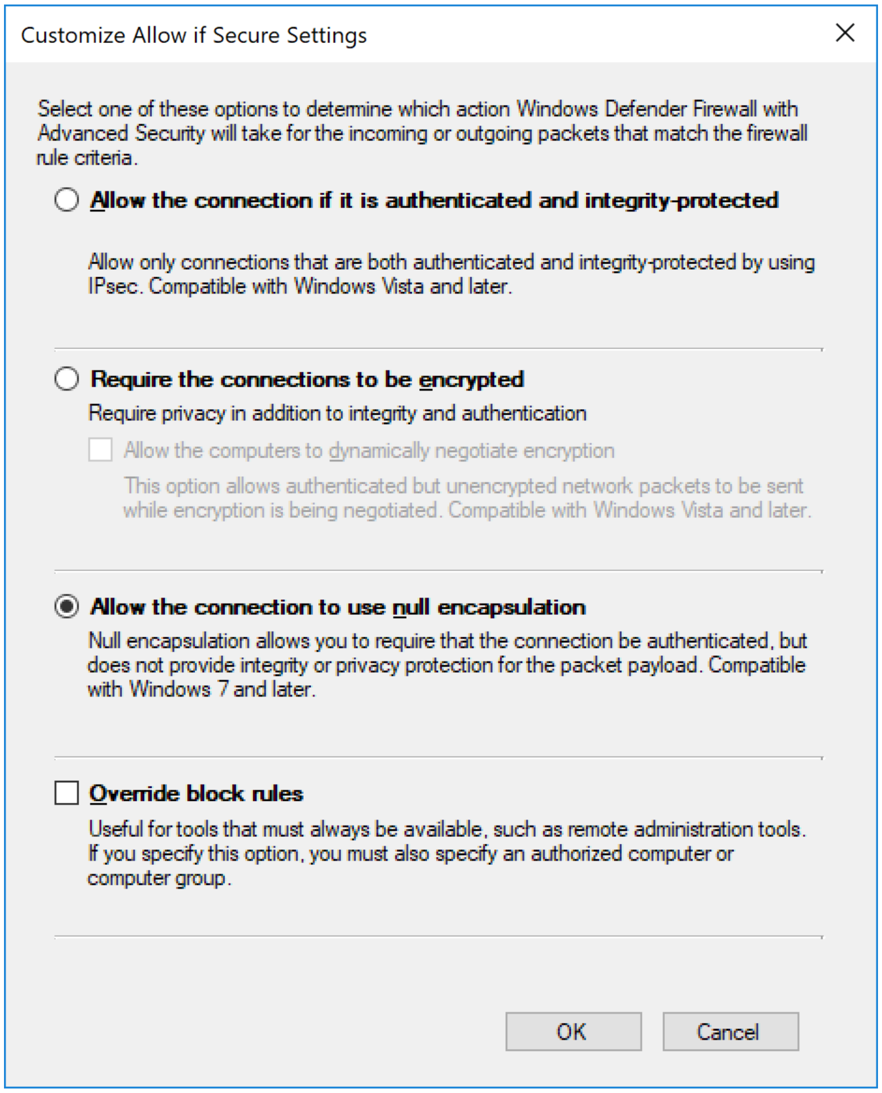
Exceptions

☐ Skip this rule for connections from these users

Remote User Requirements.

Lastly, I am going to specify the requirements needed to secure this connection. Select the **General Tab** and click **Customize** under the **Allow Connection if it is secure** option.

As WinRM uses encryption natively (e.g. kerberos or HTTPS), I do not need to double-encrypt the payload with IPSEC encryption. Additionally, I feel pretty confident that the integrity of these sessions will be guaranteed by the underlying protocol. As such, I am going to select **Allow the Session to Use Null Encapsulation,** which will ensure we authenticate the connection, but do not apply any encryption or integrity protections.


Customize Allow if Secure Settings

Select one of these options to determine which action Windows Defender Firewall with Advanced Security will take for the incoming or outgoing packets that match the firewall rule criteria.

( ) **Allow the connection if it is authenticated and integrity-protected**

Allow only connections that are both authenticated and integrity-protected by using IPsec. Compatible with Windows Vista and later.

( ) **Require the connections to be encrypted**

Require privacy in addition to integrity and authentication

[ ] Allow the computers to dynamically negotiate encryption

This option allows authenticated but unencrypted network packets to be sent while encryption is being negotiated. Compatible with Windows Vista and later.

[✓ **Allow the connection to use null encapsulation**]

Null encapsulation allows you to require that the connection be authenticated, but does not provide integrity or privacy protection for the packet payload. Compatible with Windows 7 and later.

[ ] **Override block rules**

Useful for tools that must always be available, such as remote administration tools. If you specify this option, you must also specify an authorized computer or computer group.

WinRM IPSEC Settings.

The result of this rule is that:

- Only kerberos-authenticated connections originating from HOME\dane can connect to WinRM on this machine.
- Only kerberos-authenticated connections originating from HOME\WEF$ and HOME\monolith$ can connect to WinRM on this machine.
- When the IPSEC session is established, it will be authenticated, but no other protections will be applied.

Securing RDP

The next rule we'll implement will secure incoming RDP connections with an authenticated IPSEC tunnel using the security connection rule we previously built.

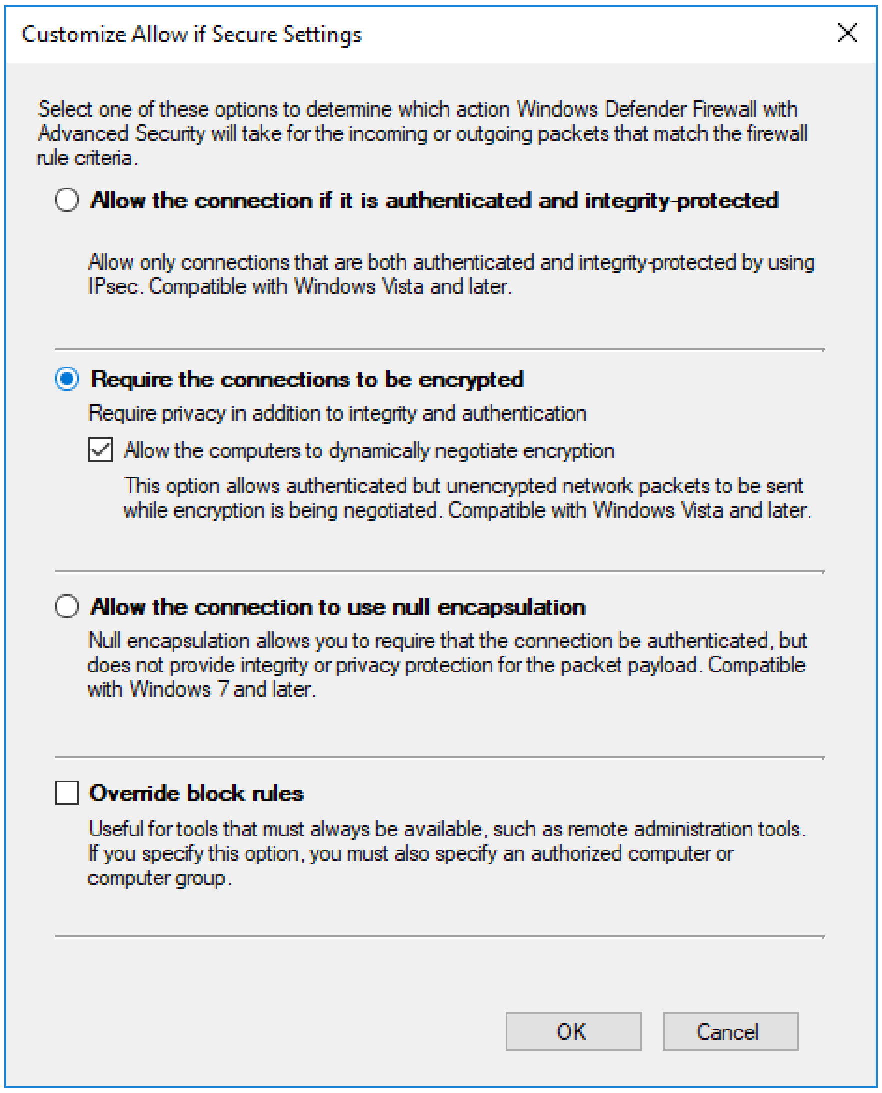
Implement this rule with the following details:

Rule Type: Custom
Programs: All Programs
Protocol Type: TCP
Local Port: 3389
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Allow if Secure
Profile: All
Name: GPO-Secure-RDP-TCP

I will repeat the **Remote Users** and **Remote Machines** configuration as per the previous rule. However, since I don't trust RDP encryption or authentication, I am going to specify additional IPSEC security requirements.

Select the **General Tab** and click **Customize** under the **Allow Connection if it is secure** option.

I am going to select **Require the connections to be encrypted,** which will not only authenticate the session, but force integrity and encryption protections. Additionally, select **Allow the Computers to Dynamically Negotiate Encryption**.


Customize Allow if Secure Settings

Select one of these options to determine which action Windows Defender Firewall with Advanced Security will take for the incoming or outgoing packets that match the firewall rule criteria.

○ **Allow the connection if it is authenticated and integrity-protected**

Allow only connections that are both authenticated and integrity-protected by using IPsec. Compatible with Windows Vista and later.

● **Require the connections to be encrypted**

Require privacy in addition to integrity and authentication

☑ Allow the computers to dynamically negotiate encryption

This option allows authenticated but unencrypted network packets to be sent while encryption is being negotiated. Compatible with Windows Vista and later.

○ **Allow the connection to use null encapsulation**

Null encapsulation allows you to require that the connection be authenticated, but does not provide integrity or privacy protection for the packet payload. Compatible with Windows 7 and later.

☐ **Override block rules**

Useful for tools that must always be available, such as remote administration tools. If you specify this option, you must also specify an authorized computer or computer group.

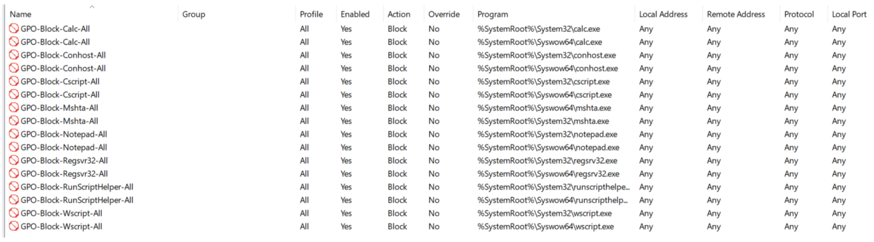
Secure Settings.

The result of this rule is that:

- Only kerberos-authenticated connections originating from HOME\dane can connect to RDP on this machine.
- Only kerberos-authenticated connections originating from HOME\monolith$ can connect to RDP on this machine.
- When the IPSEC session is established, it will be authenticated, but also require encryption and integrity protection.

Blocking Outbound Connections (Endpoints)

Lastly, the firewall provides an excellent interface for developing rules to prevent outbound connections from binaries on endpoints. These can be used to prevent services from communicating on untrusted profiles (e.g. Public, Private networks), preventing binaries from communicating to non-private resources (e.g. Non-RFC1918 addresses), or even from communicating at all. These rules can be very useful for catching unsophisticated techniques, especially those which rely on living off the land principles.



Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port
GPO-Block-Calc-All		All	Yes	Block	No	%SystemRoot%\System32\calc.exe	Any	Any	Any	Any
GPO-Block-Calc-All		All	Yes	Block	No	%SystemRoot%\Syswow64\calc.exe	Any	Any	Any	Any
GPO-Block-Conhost-All		All	Yes	Block	No	%SystemRoot%\System32\conhost.exe	Any	Any	Any	Any
GPO-Block-Conhost-All		All	Yes	Block	No	%SystemRoot%\Syswow64\conhost.exe	Any	Any	Any	Any
GPO-Block-Cscript-All		All	Yes	Block	No	%SystemRoot%\System32\cscript.exe	Any	Any	Any	Any
GPO-Block-Cscript-All		All	Yes	Block	No	%SystemRoot%\Syswow64\cscript.exe	Any	Any	Any	Any
GPO-Block-Mshta-All		All	Yes	Block	No	%SystemRoot%\Syswow64\mshta.exe	Any	Any	Any	Any
GPO-Block-Mshta-All		All	Yes	Block	No	%SystemRoot%\System32\mshta.exe	Any	Any	Any	Any
GPO-Block-Notepad-All		All	Yes	Block	No	%SystemRoot%\System32\notepad.exe	Any	Any	Any	Any
GPO-Block-Notepad-All		All	Yes	Block	No	%SystemRoot%\Syswow64\notepad.exe	Any	Any	Any	Any
GPO-Block-Regsvr32-All		All	Yes	Block	No	%SystemRoot%\System32\regsrv32.exe	Any	Any	Any	Any
GPO-Block-Regsvr32-All		All	Yes	Block	No	%SystemRoot%\Syswow64\regsrv32.exe	Any	Any	Any	Any
GPO-Block-RunScriptHelper-All		All	Yes	Block	No	%SystemRoot%\System32\runscripthelpe...	Any	Any	Any	Any
GPO-Block-RunScriptHelper-All		All	Yes	Block	No	%SystemRoot%\Syswow64\runscripthelp...	Any	Any	Any	Any
GPO-Block-Wscript-All		All	Yes	Block	No	%SystemRoot%\System32\wscript.exe	Any	Any	Any	Any
GPO-Block-Wscript-All		All	Yes	Block	No	%SystemRoot%\Syswow64\wscript.exe	Any	Any	Any	Any

Outbound Firewall Deny Rules.

In the GPO editor, open up the **Workstations Firewall Policy** and navigate to **Computer Configuration** > **Windows Settings** > **Security Settings** > **Windows Defender Firewall with Advanced Security** > **Outbound Rules.**

Blocking Calc.exe

While not the most glamorous of defensive strategies, calc.exe is commonly abused by default behaviors for process migration and injection techniques. This is an example rule which prevents calc.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\calc.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Calc-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\calc.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Calc-All

Blocking Notepad.exe

While not the most glamorous of defensive strategies, notepad.exe is commonly abused by default behaviors for process migration and injection techniques. This is an example rule which prevents notepad.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\notepad.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Notepad-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\notepad.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports

Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Notepad-All

Blocking Conhost.exe

While not the most glamorous of defensive strategies, conhost.exe is commonly abused by default behaviors for process migration and injection techniques. This is an example rule which prevents conhost.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\conhost.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Conhost-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\Conhost.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Conhost-All

Blocking Mshta.exe

Mshta.exe is commonly abused by attackers to proxy execution of malicious .hta files and Javascript or VBScript. This is an example rule which prevents mshta.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\mshta.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Mshta-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\mshta.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Mshta-All

Blocking Cscript.exe

Cscript.exe is commonly abused by attackers to execute malicious scripts. This is an example rule which prevents Cscript.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\Cscript.exe

Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Cscript-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\Cscript.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Cscript-All

Blocking Wscript.exe

Wscript.exe is commonly abused by attackers to execute malicious scripts. This is an example rule which prevents Wscript.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\Wscript.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Wscript-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\Wscript.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-Wscript-All

Blocking RunScriptHelper.exe

RunScriptHelper.exe is commonly abused by attackers to execute malicious scripts. This is an example rule which prevents RunScriptHelper.exe from communicating with any network resources.

Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\System32\RunScriptHelper.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports
Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-RunScriptHelper-All

We'll repeat this rule, but for the syswow64 path. Implement this rule with the following details:

Rule Type: Custom
Programs: %SystemRoot%\Syswow64\RunScriptHelper.exe
Protocol Type: Any
Local Port: Any
Remote Port: All Ports

Scope (Local): Any IP Address
Scope (Remote): Any IP Address
Action: Block
Profile: All
Name: GPO-Block-RunScriptHelper-All

Server Configuration

The steps to require servers are mostly identical to the steps listed above for workstations. That said, there are a few important caveats to be mindful of:

- You will want to create a base Server Firewall Policy GPO with all of the default firewall settings configurations and connection security rules.
- You will want multiple GPOs for different server types. For example, file servers will want their own policy GPO which allows SMB inbound (preferably with IPSEC authentication, integrity, and encryption).
- Remember that Active Directory Organizational Units (OUs) are policy boundaries for GPOs. Try to put like-kind servers in the same OU or, failing that, use security filtering and security groups to apply firewall rules.

Again, it is very important to use the **request** option for security connection rules when testing. This will allow graceful fallback to the underlying protocol, which will prevent service availability issues.

Protips

There are a few protips that I've learned while deploying this across my home environment:

- Test slowly using security filtering. It is easy to accidentally roll out a bad firewall rule or security configuration profile, which could cause loss of access to your device. I recommend having a low frequency for GPO refreshes and a slow roll-out to quickly recover from any bad firewall rules.

- These rules will override locally set rules. If you have specific applications that require **inbound connectivity**, you will need to ensure it's managed via a relevant GPO. Failure to plan out what applications you're using on your network will break connectivity.
- If you intend to use Hyper-V on machines running the Windows Firewall, you will need to create a rule allowing **inbound DNS (TCP/UDP 53) on all interfaces from all profiles**. This is required to process DNS packets from Hyper-V guests to the hypervisor. Failure to do so will break default DNS resolution within Hyper-V guests.
- You will need to ensure that UDP Port 500 (IKE) and ESP are not filtered by any network firewalls. Using IPSEC to protect protocols will initiate the tunnel via IPSEC (UDP 500) and then tunnel traffic over ESP, which could be filtered by firewalls inside your network. Use wireshark and validate all the connections look sane and reasonable.
- If you're looking to use bloodhound or Windows ATA on your network, you'll need to ensure you have a rule allowing RPC from your trusted hosts. This is required to allow SAMR requests from these tools, which can provide deeper insight into the configuration of your endpoints.
- I highly recommend installing Glasswire, a freemium application which provides deeper insight into network connectivity on windows hosts. The paid upgrade for glasswire additionally allows for more restrictive firewall policies (e.g. ask for connectivity for new binaries), but these may require enabling local rule processing in your GPO.

Further Reading and Acknowledgements

- Demystifying the Windows Firewall
- Windows Firewall Documentation
- Securing RDP with IPSEC
- Creating a IPSEC Tunnel with the Windows Firewall

- Windows Hardening Script
- The kind people who reviewed this post.