PowerShell Remoting Configuration

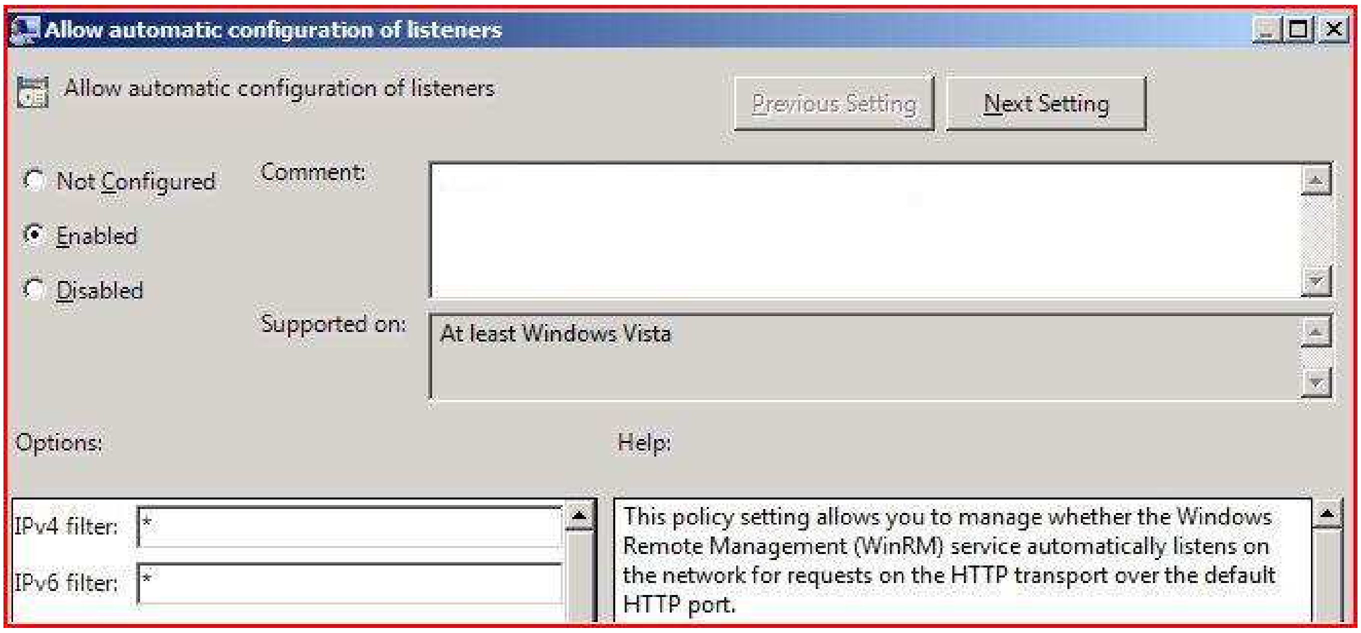
This document will explain the proposed steps to configure PowerShell Remoting. Because there are two methods of configuring Remoting (HTTP and HTTPS), this document will explain both configurations, beginning with the HTTP recommendations.

HTTP Remoting Configuration

Because HTTP-based PowerShell Remoting can be configured directly through Group Policy, it makes sense to enable it this way, as it results in the least administrative overhead. The following steps will describe Group Policy configuration and application as well as security application.

The policy to configure PowerShell Remoting is located under Computer Configuration -> Policies -> Administrative Templates -> Windows Components -> Windows Remote Management -> WinRM Service.

The only policy object that needs to be configured to enable Remoting, is “Allow automatic configuration of listeners”, which must be set to “Enabled”.


Allow automatic configuration of listeners
Allow automatic configuration of listeners
Previous Setting
Next Setting
Comment:
Supported on:
Options:
Help:

Not Configured [✓ Enabled] Disabled

Supported on: At least Windows Vista

IPv4 filter: *

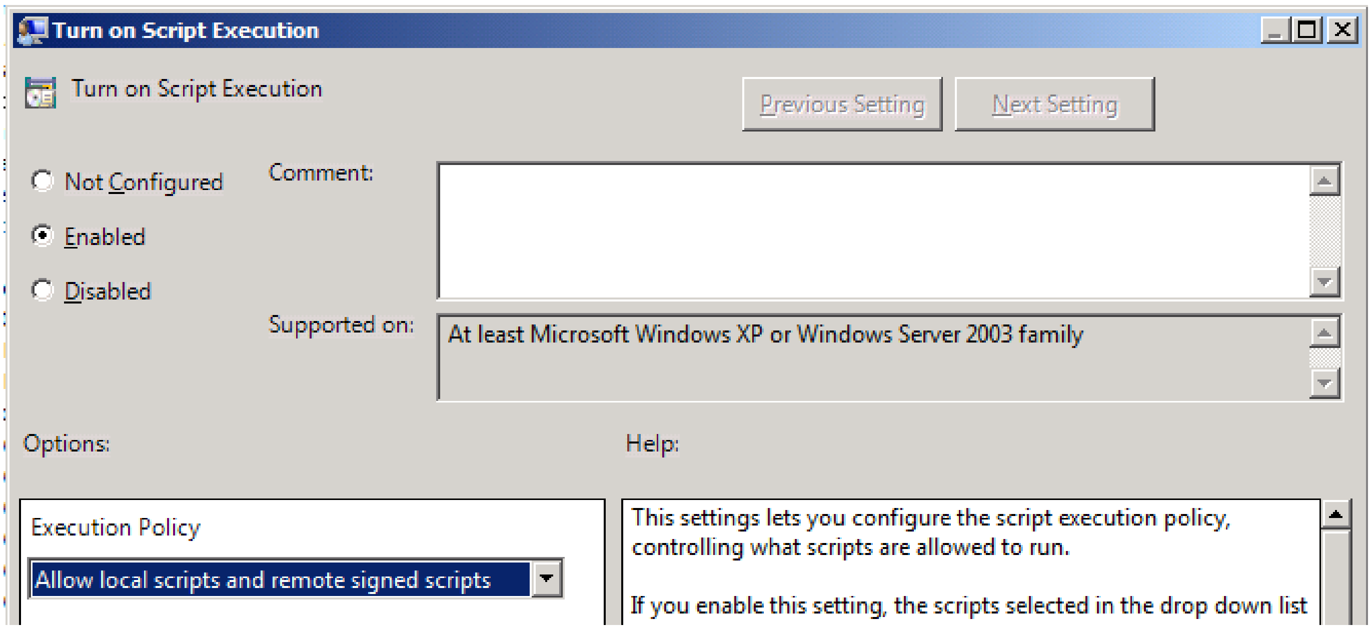
IPv6 filter: *

This policy setting allows you to manage whether the Windows Remote Management (WinRM) service automatically listens on the network for requests on the HTTP transport over the default HTTP port.

Despite the deceptive names, the “IPv4 Filter” and “IPv6 Filter” only specify an IP or IP range on an interface to listen on. They cannot be used to limit IP addresses from connecting to a machine, only the range the machine will accept Remoting connections on.

By default, this listener will use HTTP as the transport protocol. This is because an HTTPS listener cannot be directly configured through Group Policy.

Because it isn’t as useful to have Remoting but not be able to run scripts, the setting under Computer Configuration -> Policies -> Administrative Templates -> Windows Components -> Windows PowerShell named “Turn on Script Execution” should be set to “Enabled” with a value of “Allow local scripts and remote signed scripts”.


Turn on Script Execution

Turn on Script Execution

( ) Not Configured
(●) Enabled
( ) Disabled

Comment:

Supported on: At least Microsoft Windows XP or Windows Server 2003 family

Options:

Execution Policy

Allow local scripts and remote signed scripts

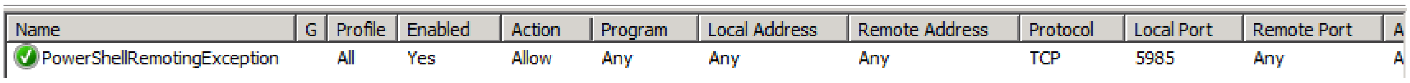
Help:

This settings lets you configure the script execution policy, controlling what scripts are allowed to run.

If you enable this setting, the scripts selected in the drop down list

This will allow scripts to run on the machine, while requiring that scripts downloaded from the internet to have been signed with a code signing certificate.

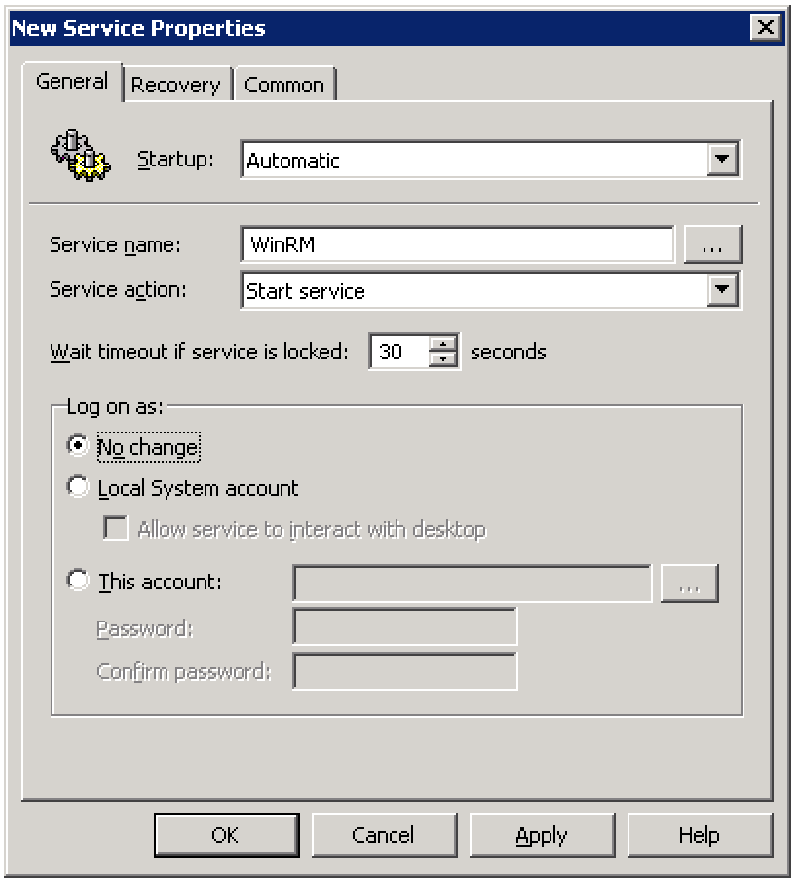
The machines in the test environment used for this proposal retained their Windows Firewall, and as such, needed to be configured to allow PowerShell Remoting using Group Policy. The settings for the Windows Firewall can be located under Computer Configuration -> Windows Settings -> Security Settings -> Windows Firewall with Advanced Security -> Windows Firewall with Advanced Security -> Inbound Rules. A rule needs to be added for port 5985, which is the default port for HTTP Remoting. Create a new inbound rule using the New Inbound Rule Wizard with the Port option.



Name	G	Profile	Enabled	Action	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port	A
✓ PowerShellRemotingException		All	Yes	Allow	Any	Any	Any	TCP	5985	Any	A

The Windows Firewall could be used to restrict access from IP addresses or computers, but may not be used on the computers, so this may not be practical. This Windows Firewall example is purely for demonstration purposes. Networking could configure their hardware firewalls to allow traffic to and from port 5985 only from specific IP addresses as well, limiting the computers that would be able to use the Remoting service.

A setting should also be created under Computer Configuration -> Preferences -> Control Panel Settings -> Services to ensure that the WinRM service is started automatically at boot:


New Service Properties

General | Recovery | Common

Startup: Automatic

Service name: WinRM

Service action: Start service

Wait timeout if service is locked: 30 seconds

Log on as:

[✓ No change]

Local System account

Allow service to interact with desktop

This account:

Password:

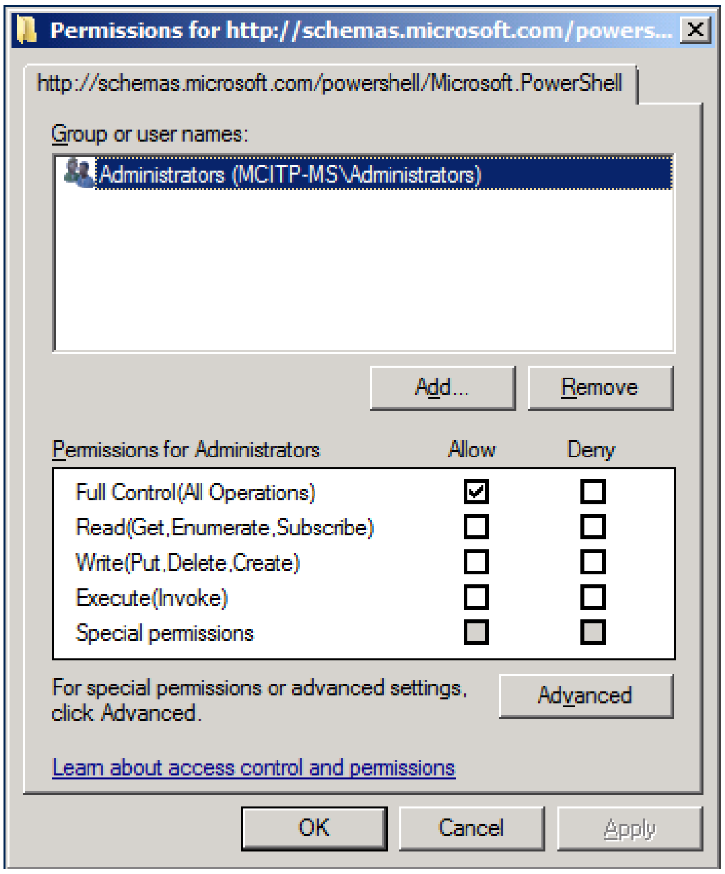
Confirm password:

If this is not configured, any machines that have this service set to “Disabled”, “Manual”, or “Automatic(Delayed Start)” on the local machine will hang when applying the permissions script to the PowerShell Remoting listener. This is because the WinRM service must be running before the listener configuration can be changed.

These Group Policy settings, applied to a container, will be enough to configure HTTP PowerShell Remoting on domain computers, but with just the Group Policy configuration, there are no access restrictions to internal users who are members of the Administrators group on a particular domain computer. The next steps to configuring HTTP Remoting involve configuring access restrictions on the Remoting listener.

An Active Directory global security group should be created that all users who need access to PowerShell Remoting should be added to. The group in the demonstration environment was named PSRemotingUsers.

The next step in configuring security is to change the properties of the Remoting listener on a single machine. Entering the command `Set-PSSessionConfiguration –ShowSecurityDescriptorUI –Name Microsoft.PowerShell –Force` will open the Security configuration window for the PowerShell Remoting listener.



Permissions for http://schemas.microsoft.com/powers...

http://schemas.microsoft.com/powershell/Microsoft.PowerShell

Group or user names:

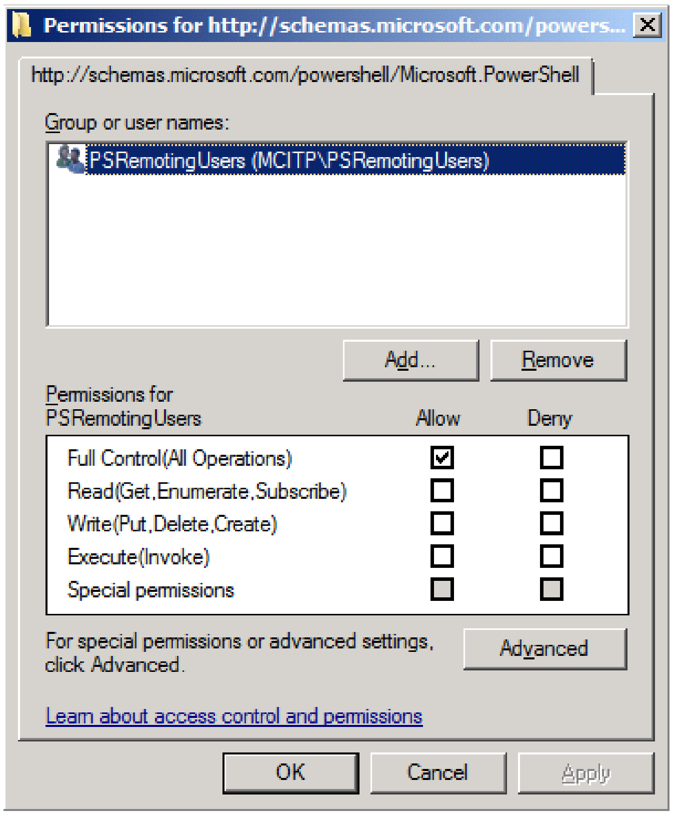
Administrators (MCITP-MS\Administrators)

Permissions for Administrators	Allow	Deny
Full Control(All Operations)	☑	☐
Read(Get,Enumerate,Subscribe)	☐	☐
Write(Put,Delete,Create)	☐	☐
Execute(Invoke)	☐	☐
Special permissions	☐	☐

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

From this UI, you can add the PowerShell Remoting security group from Active Directory created in the first step. In order to take full advantage of Remoting features, it is best if the group created has "Full Control" access rights. To further limit access to Remoting, the Administrators group can be removed.



Permissions for http://schemas.microsoft.com/powers...

http://schemas.microsoft.com/powershell/Microsoft.PowerShell

Group or user names:

PSRemotingUsers (MCITP\PSRemotingUsers)

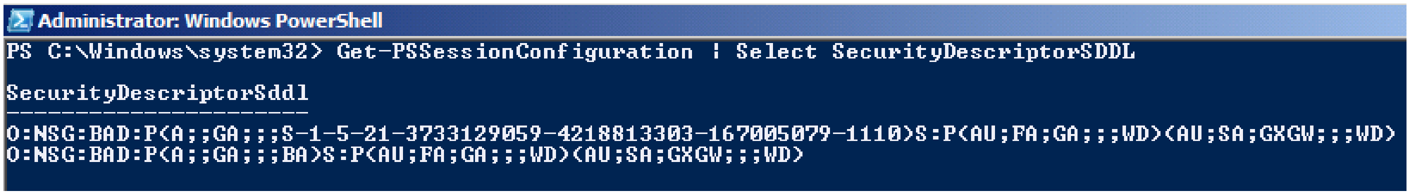
Permissions for PSRemotingUsers	Allow	Deny
Full Control(All Operations)	☑	☐
Read(Get,Enumerate,Subscribe)	☐	☐
Write(Put,Delete,Create)	☐	☐
Execute(Invoke)	☐	☐
Special permissions	☐	☐

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

The machine's listener is now configured to only allow access from users in the PSRemotingUsers Active Directory security group. However, this change is not domain-wide; it only applies to the machine it was run on. To add these security settings to the domain, you will need to apply the security descriptors for the Remoting listener to each machine. Since there are likely many machines, the UI process is not useful for this portion of the configuration. Fortunately, PowerShell has commands built-in that can set these permissions for us.

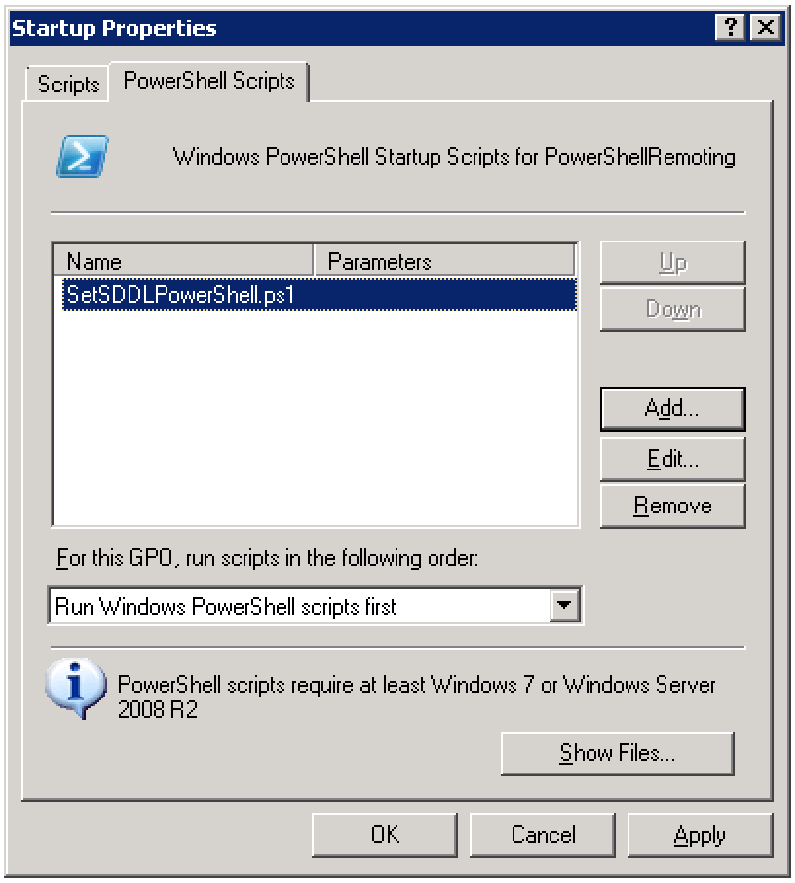
On the server where the security permissions were changed, run the command `Get-PSSessionConfiguration | Select SecurityDescriptorSDDL` to show the Windows SDDL for the security configured on the listener.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Get-PSSessionConfiguration | Select SecurityDescriptorSDDL

SecurityDescriptorSddl
----------------------
O:NSG:BAD:P(A;;GA;;;S-1-5-21-3733129059-4218813303-167005079-1110)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
O:NSG:BAD:P(A;;GA;;;BA)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
```

Because the SDDL has to be set at an elevated command prompt, to do the SDDL configurations remotely will require a Group Policy. A PowerShell script was created with the contents `Set-PSSessionConfiguration -Name Microsoft.PowerShell -SecurityDescriptorSDDL "O:NSG:BAD:P(A;;GA;;;S-1-5-21-3733129059-4218813303-167005079-1110)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)" -Force`. This script will apply the PowerShell Remoting group restrictions to the PowerShell listener on the machines that it runs on. The Group Policy was created under Computer Configuration -> Windows Settings -> Scripts (Startup/Shutdown). The PowerShell script was added to the SYSVOL share and configured as a Startup script in the Policy object under the "PowerShell Scripts" tab.



Startup Properties

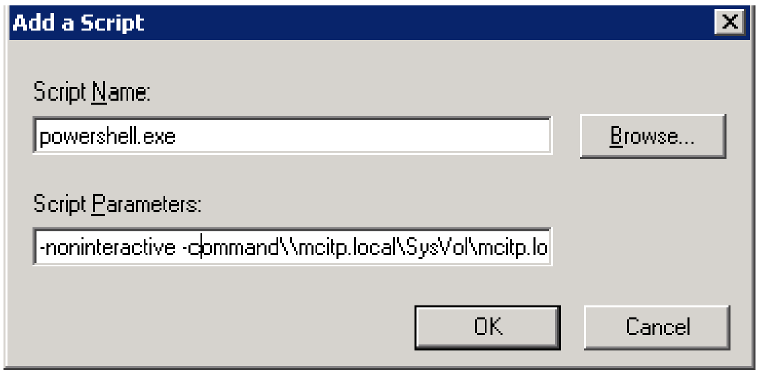
Windows PowerShell Startup Scripts for PowerShellRemoting

Name	Parameters
SetSDDLPowerShell.ps1	

For this GPO, run scripts in the following order: Run Windows PowerShell scripts first

PowerShell scripts require at least Windows 7 or Windows Server 2008 R2

If there was a need to apply the configuration to Server 2003 or Server 2008, an alternative method would be used to apply the script. Instead of using the PowerShell Scripts tab in the Group Policy Manager, the regular Scripts tab would need to be used. A reference to the SYSVOL share would be used to run a PowerShell script located there. The "Script Name" would be `powershell.exe`, while the "Script Parameters" would be `-noninteractive –command \\mcitp.local\SysVol\mcitp.local\Policies\{976DE4C5-17CF-40EB-AEEB-048D70FFEC62}\Machine\Scripts\Startup\SetSDDLPowerShell.ps1`.



Script Name: powershell.exe

Script Parameters: -noninteractive -command\\mcitp.local\SysVol\mcitp.lo

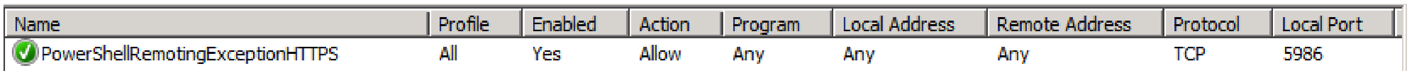
This will run the PowerShell script from the SYSVOL on the Server 2003 or Server 2008 machine. Since Server 2003 does not come with PowerShell installed, and Server 2008 is being phased out in favor of Server 2008 R2, the merits of this method are debatable.

HTTPS Listener Configuration

The HTTPS listener configuration starts out using the above steps to configure the HTTP listener, with the exception of the "Allow automatic configuration of listeners" policy. This is because the HTTPS listener cannot be directly configured through Group Policy, so some scripting was needed to enable the HTTPS listener, as well as a change to the Windows Firewall configuration.

Ensure that the computers are configured to have valid Computer certificates. In the demonstration environment, a Certificate Authority was created for this purpose, along with a Group Policy Object to automatically assign Computer Certificates. It is important that these certificates come from a trusted authority to prevent issues with untrusted Remoting.

A new Windows Firewall exception will need to be created for port 5986, which is the HTTPS listener port.



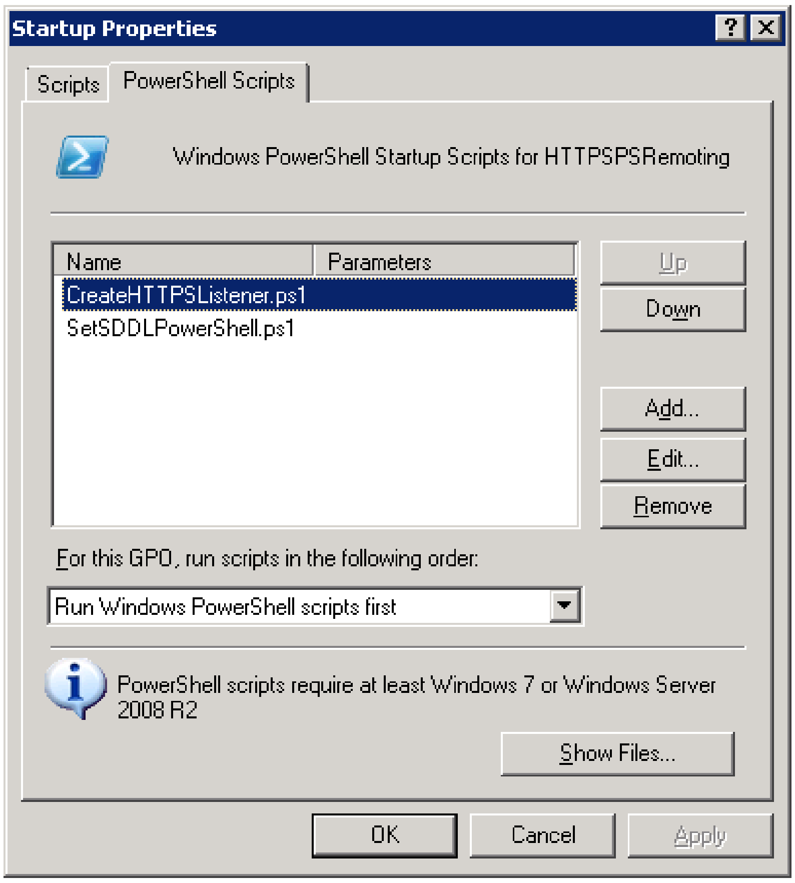
Name	Profile	Enabled	Action	Program	Local Address	Remote Address	Protocol	Local Port
✓ PowerShellRemotingExceptionHTTPS	All	Yes	Allow	Any	Any	Any	TCP	5986

As before, the use of Windows Firewall in this proposal is for demonstration only. The Networking team could configure a similar rule in their hardware firewalls, possibly with IP address restrictions if desired.

Once the HTTP listener is configured, a script was created with the following commands:

```
$ipProperties =  $ipProperties =
[System.Net.NetworkInformation.IPGlobalProperties]::GetIPGlobalProperties()
$Hostname = "{0}.{1}" -f $ipProperties.Hostname,$ipProperties.DomainName
$CertThumbprint = Get-ChildItem "Cert:\LocalMachine\My" | Select -First 1
$CertThumbprintValue = $CertThumbprint | foreach-Object {$_.Thumbprint}
New-WSManInstance winrm/config/listener -SelectorSet @{Address="*";Transport="HTTPS"} -
ValueSet @{Hostname=$Hostname;CertificateThumbprint=$CertThumbprintValue}
```

This script was then placed in the SYSVOL directory and added as a PowerShell logon script in the HTTPSPSRemoting Group Policy. This policy must come before the SDDL permissions script so the permission changes are applied correctly.



Windows PowerShell Startup Scripts for HTTPSPSRemoting

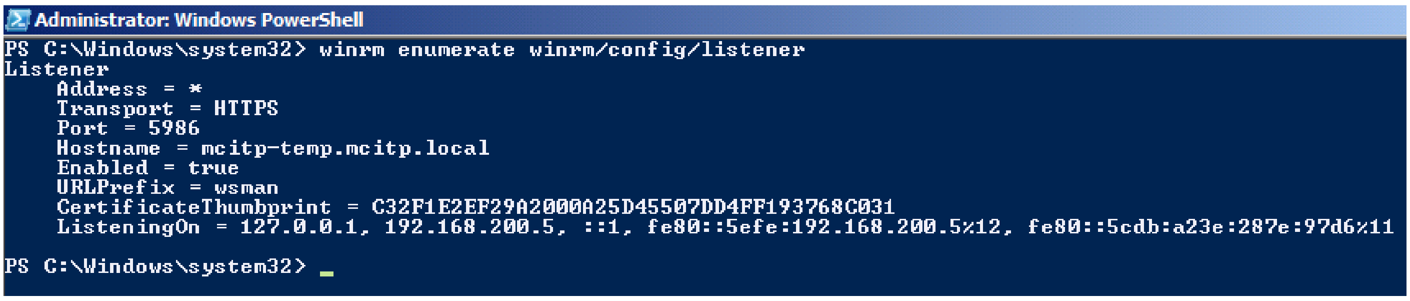
Name	Parameters
CreateHTTPSListener.ps1	
SetSDDLPowerShell.ps1	

For this GPO, run scripts in the following order:

Run Windows PowerShell scripts first

PowerShell scripts require at least Windows 7 or Windows Server 2008 R2

When this Group Policy applies to the machines with HTTPS, the command `winrm enumerate winrm/config/listener` should display the HTTPS listener information.

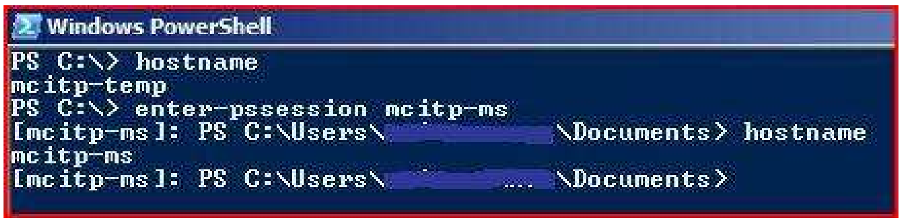


```
PS C:\Windows\system32> winrm enumerate winrm/config/listener
Listener
    Address = *
    Transport = HTTPS
    Port = 5986
    Hostname = mcitp-temp.mcitp.local
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint = C32F1E2EF29A2000A25D45507DD4FF193768C031
    ListeningOn = 127.0.0.1, 192.168.200.5, ::1, fe80::5efe:192.168.200.5%12, fe80::5cdb:a23e:287e:97d6%11

PS C:\Windows\system32> _
```

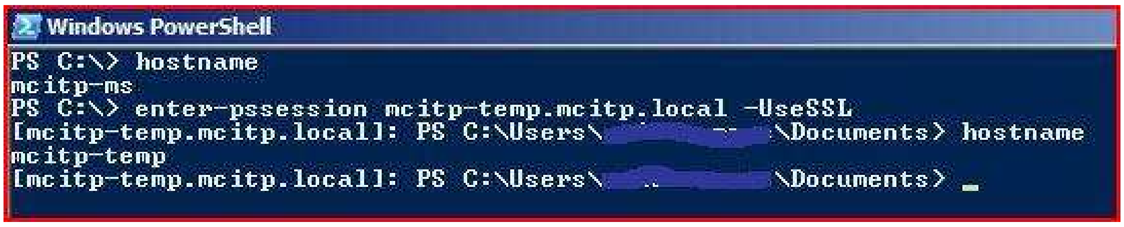
Making Connections via PowerShell Remoting

Successful connections using an HTTP listener are made using the command `Enter-PSSession <hostname>`, where <hostname> is the name of the computer to connect to.



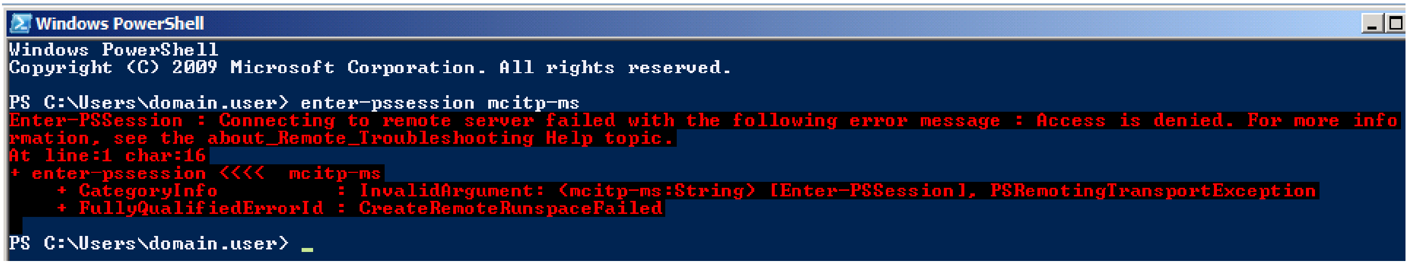
```
PS C:\> hostname
mcitp-temp
PS C:\> enter-pssession mcitp-ms
[mcitp-ms]: PS C:\Users\[redacted]\Documents> hostname
mcitp-ms
[mcitp-ms]: PS C:\Users\[redacted]\Documents>
```

Successful connections using an HTTPS listener are made using the command `Enter-PSSession <hostname> -UseSSL`, where <hostname> is the name of the computer to connect to.



```
PS C:\> hostname
mcitp-ms
PS C:\> enter-pssession mcitp-temp.mcitp.local -UseSSL
[mcitp-temp.mcitp.local]: PS C:\Users\[redacted]\Documents> hostname
mcitp-temp
[mcitp-temp.mcitp.local]: PS C:\Users\[redacted]\Documents> _
```

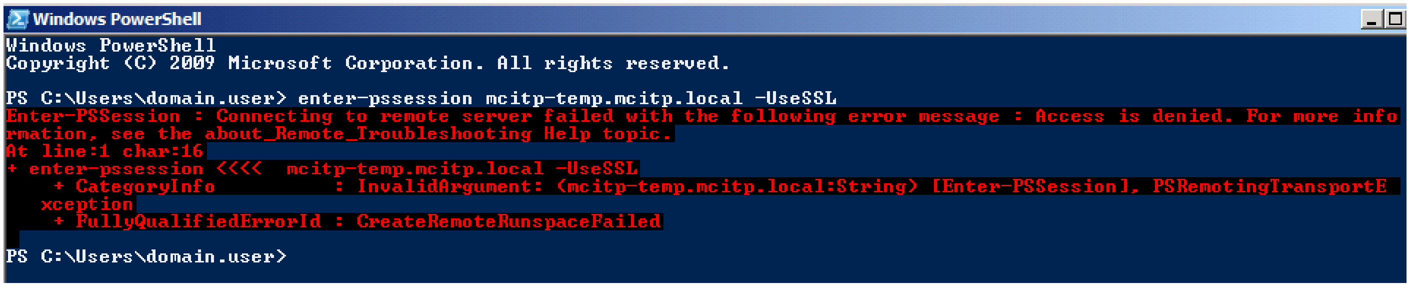
Note that a user that is not part of the PSRemotingUsers group is unable to connect via HTTP or HTTPS.



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\domain.user> enter-pssession mcitp-ms
Enter-PSSession : Connecting to remote server failed with the following error message : Access is denied. For more info
rmation, see the about_Remote_Troubleshooting Help topic.
At line:1 char:16
+ enter-pssession <<<<  mcitp-ms
    + CategoryInfo          : InvalidArgument: (mcitp-ms:String) [Enter-PSSession], PSRemotingTransportException
    + FullyQualifiedErrorId : CreateRemoteRunspaceFailed

PS C:\Users\domain.user> _
```



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\domain.user> enter-pssession mcitp-temp.mcitp.local -UseSSL
Enter-PSSession : Connecting to remote server failed with the following error message : Access is denied. For more info
rmation, see the about_Remote_Troubleshooting Help topic.
At line:1 char:16
+ enter-pssession <<<<  mcitp-temp.mcitp.local -UseSSL
    + CategoryInfo          : InvalidArgument: (mcitp-temp.mcitp.local:String) [Enter-PSSession], PSRemotingTransportE
   xception
    + FullyQualifiedErrorId : CreateRemoteRunspaceFailed

PS C:\Users\domain.user>
```

Conclusion and Recommendations

To conclude, I'd like to offer a few recommendations on general decision making for PowerShell Remoting. Because some networks are considered trusted, HTTPS may not be completely necessary. It is more work to configure and more error-prone due to administrator-designed Policy application scripts, but offers slightly better security due to its encryption. Remote connections to a machine with an HTTPS listener must use the FQDN.

Due to the migration away from Windows Server 2003 and 2008, it may not be worth the headache to eschew the direct PowerShell scripting method of logon scripts in Group Policy. It would be a good deal of extra work to install PowerShell 2.0 on Windows Server 2003 machines, and wouldn't offer a reasonable return on investment if you are planning to migrate off those platforms. If you plan to keep them however, it may be prudent to create a second organizational unit for Server 2008 and 2003 machines to ensure you are using the most reliable method of Group Policy application for the majority of machines that use Server 2008 R2.