

Windows Client Assessment: Prerequisites and Configuration

This document explains the required steps to configure the Windows Client Assessment included with your Azure Log Analytics Workspace and entitled Microsoft On-Demand assessment.

There are configuration and setup tasks to be completed prior to executing the assessment setup tasks in this document. For all pre-work, follow the [Getting Started with On-Demand Assessments](#) in the Services Hub Resource Center.

Table of Contents

System Requirements and Configuration at Glance.....	2
Supported Versions.....	2
Common to Both Scenarios.....	2
Data Collection Machine.....	2
Powershell Remoting.....	2
User Profile Service.....	8
Setting up the Windows Client Assessment.....	8
Appendix.....	12
Data Collection Methods.....	12

System Requirements and Configuration at Glance

According to the scenario you want to use, review the following details to ensure that you meet the necessary requirements.

Supported Versions

- This service is available for Active Directory domain joined clients running Windows 8.1 or later.
- Clients purely joined to Azure Active Directory or Workgroups may not be supported. To support this scenario, you will have to assess each client machine individually. Please reach out to serviceshubteam@ppas.uservoice.com to get the latest package and [follow the steps in this article](#) to run in this scenario.

Common to Both Scenarios

- You will need a log analytics workspace
- **User account rights:**
 - A domain account with the following rights:
 - Member of the local Administrators group on all clients in the environment
 - Member of the local Administrators group of the tools machine
 - Unrestricted network access from the Tools machine to all clients

Data Collection Machine

- The **data collection machine** must be domain joined and have a Windows domain trust path to the domain joined clients to be assessed.
- **Data collection machine hardware:** Minimum 8 gigabytes (GB) of RAM, 2 gigahertz (GHz) dual-core processor, minimum 5 GB of free disk space, plus up to 6 GB for every target client in the assessed environment during data collection.
- The **data collection machine** is used to connect to all clients and retrieve information from it, communicating over Remote Procedure Call (RPC), Server Message Block (SMB), WMI, remote registry, SQL Database, Lightweight Directory Access Protocol (LDAP) and Distributed Component Object Model (DCOM).
- The CLR version on the data collection machine should be using .NET 4.0 or greater. This can be verified by running `$PSVersionTable.CLRVersion` in the PowerShell prompt
- Microsoft .NET Framework 4.6.2 or newer installed and running Windows Server 2012 R2 or newer.
- The data collection machine must have the Microsoft Monitoring Agent installed and configured for one of the deployment scenarios at the beginning of this document.

Powershell Remoting

To complete the assessment with the accurate results, you will need to configure all in-scope target machines for Powershell remoting.

PowerShell on the tools machine is used to scan the computers for installed security patches as well as audit policy configuration.

- Windows Update Agent must be running on all in-scope workstations for the security update scan

Additional requirements for Windows 7 and later Target Machines:

The following three items must be configured on target workstations to support data collection: PowerShell Remoting,

WinRM service and Listener, and Inbound Allow Firewall Rules.

Note1: *Windows 7 and Windows 10 have WinRM and PowerShell remoting enabled by default. The following configuration steps detailed below will only need to be implemented if the default configuration for target machines has been altered:*

- Execute **Enable-PSRemoting** Powershell cmdlet on each target machine within the scope of the assessment. This one command will configure PS-Remoting, WinRM service and listener, and enable required Inbound FW rules. A detailed description of everything Enable-PSRemoting does is documented [here](#).

OR

- Configure **WinRM / PowerShell remoting** via Group Policy (Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service)
 - **"Allow remote server management through WinRM"**.
- Configure **WinRM service for automatic start** via Group Policy (Computer Configuration\Policies\Windows Settings\Security Settings\SystemServices)
 - Define **Windows Remote Management (WS-Management)** service for **Automatic startup mode**
- Configure **Inbound allow Firewall Rules:** This can be done individually in the local firewall policy of every in-scope target workstation or via a group policy which allow communication from the tools machine.

Two steps are involved to configure a group policy to enable both WinRM listener and the required inbound allow firewall rules:

- A) Identify the IP address of the source computer where data collection will occur from.
- B) Create a new GPO linked to the workstation organizational unit(s), and define an inbound rule for the tools machine

A.) Log into the chosen data collection machine to identify its current IP address using IPConfig.exe from the command prompt.

An example output is as follows

```
C:\>ipconfig
```

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :

Link-local IPv6 Address : fe80::X:X:X:X%13

IPv4 Address. : **X.X.X.X**

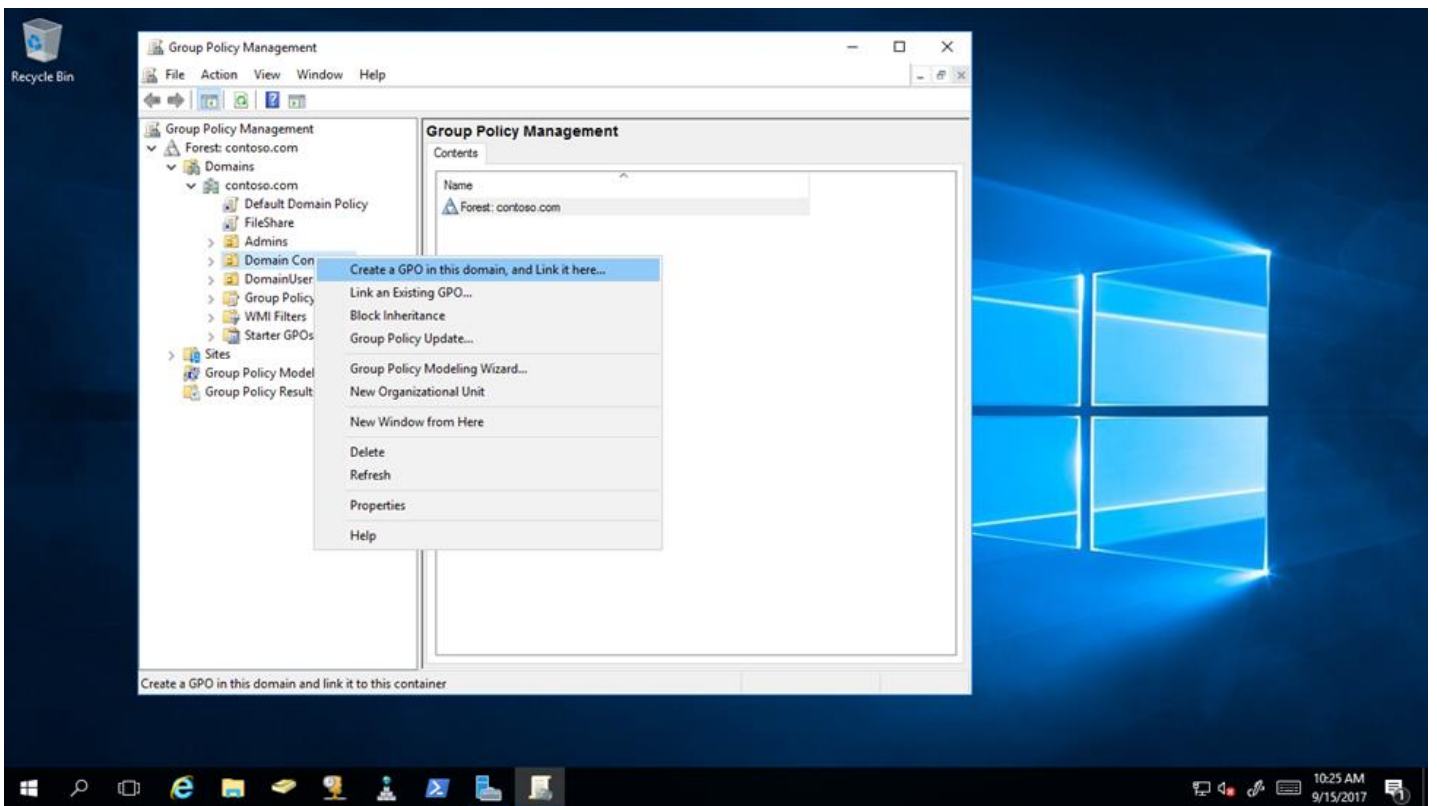
Subnet Mask : X.X.X.X

Default Gateway : X.X.X.X

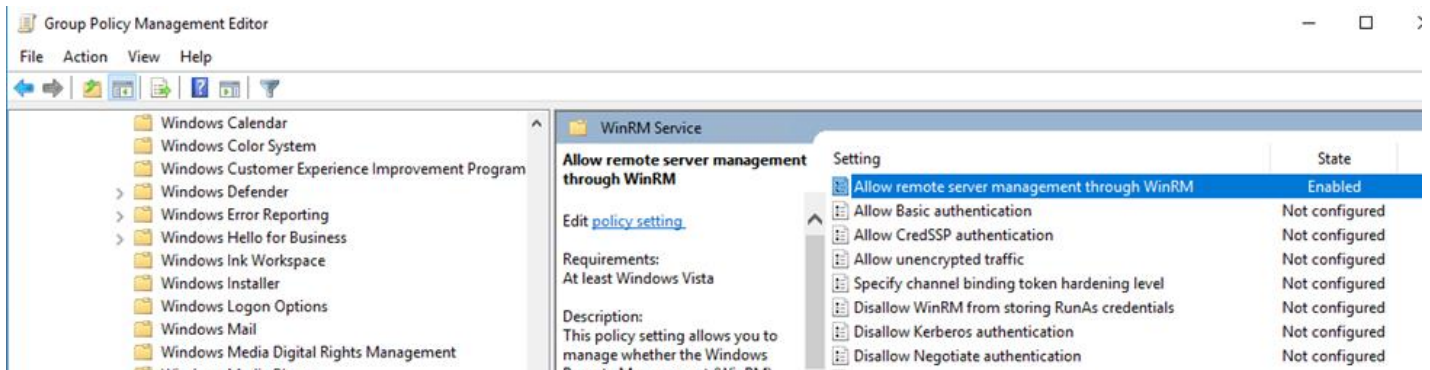
Make a note of the IPv4 address of your machine. The final step in the configuration will use this address to ensure only the data collection machine can communicate with the Windows Update Agent on the target workstations.

B.) Create, configure, and link a group policy object to the workstations OU(s) in each domain in the forest.

1. Create a new GPO. Make sure the GPO applies to the workstation organizational unit(s). Give the new group policy a name based on your group policy naming convention or something that identifies its purpose similar to "Windows Client Assessment"



2. Within the GPO open: (Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service). Enable "**Allow remote server management through WinRM**". You will need to specify IPv4 and IPv6 filters. ("*" will allow all inbound servers access, but specifying the IP address of the tools machine is preferred)



Allow remote server management through WinRM

Allow remote server management through WinRM Previous Setting Next Setting

☐ Not Configured
 ☒ Enabled
 ☐ Disabled

Comment:

Supported on: At least Windows Vista

Options:

IPv4 filter: *

IPv6 filter: *

Syntax:

Type "*" to allow messages from any IP address, or leave the field empty to listen on no IP address. You can specify one or more ranges of IP addresses.

Example IPv4 filters:

2.0.0.1-2.0.0.20, 24.0.0.1-24.0.0.22

*

Example IPv6 filters:

Help:

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.

If you enable this policy setting, the WinRM service automatically listens on the network for requests on the HTTP transport over the default HTTP port.

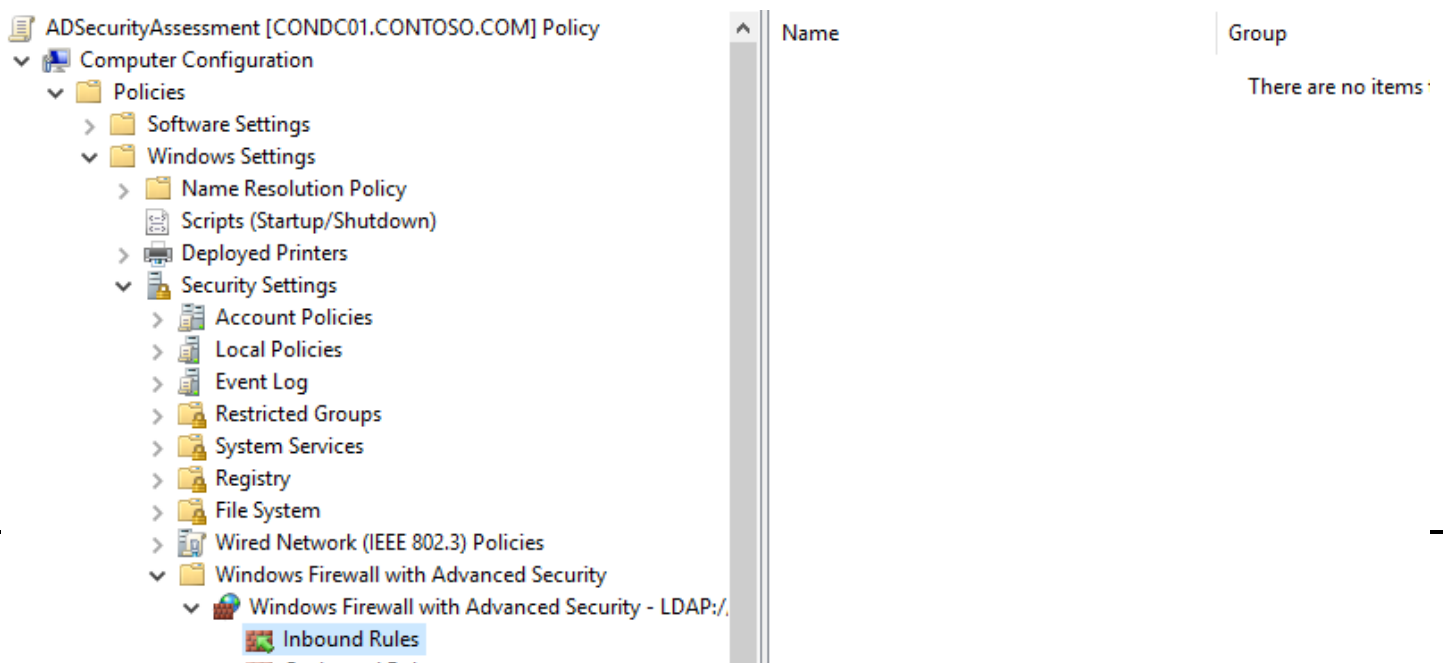
To allow WinRM service to receive requests over the network, configure the Windows Firewall policy setting with exceptions for Port 5985 (default port for HTTP).

If you disable or do not configure this policy setting, the WinRM service will not respond to requests from a remote computer, regardless of whether or not any WinRM listeners are configured.

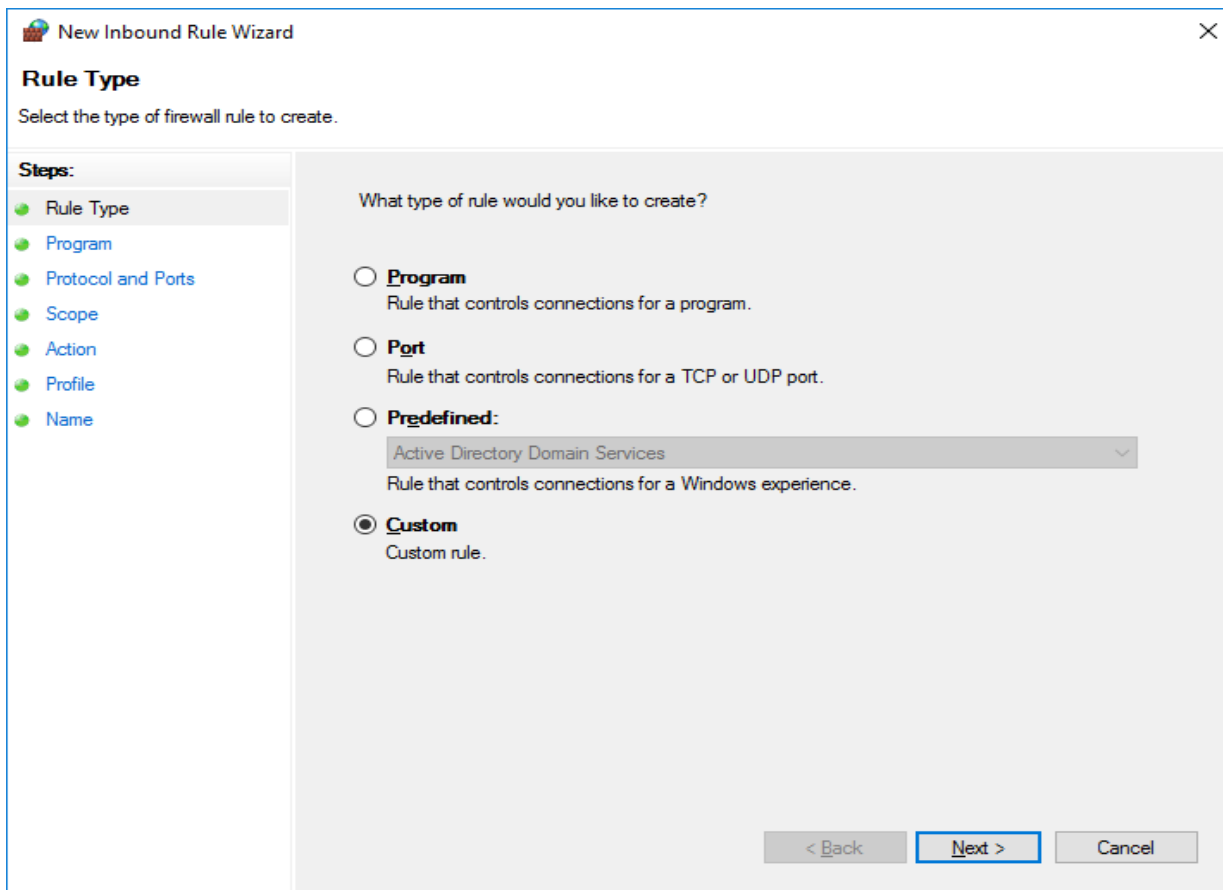
The service listens on the addresses specified by the IPv4 and IPv6 filters. The IPv4 filter specifies one or more ranges of IPv4 addresses, and the IPv6 filter specifies one or more ranges of IPv6 addresses. If specified, the service enumerates the available IP addresses on the computer and uses only addresses that fall within one of the filter ranges.

OK Cancel Apply

3. Create an advanced Inbound Firewall Rule to allow all network traffic from the tools machine to the target workstations. This can be applied to the same GPO that was used in step 1 above. (Computer Configuration\Policies\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Windows Firewall with Advanced Security – LDAP:/xxx\Inbound Rules)
4. To create the new rule, Right Click on "Inbound Rules" and select "New"

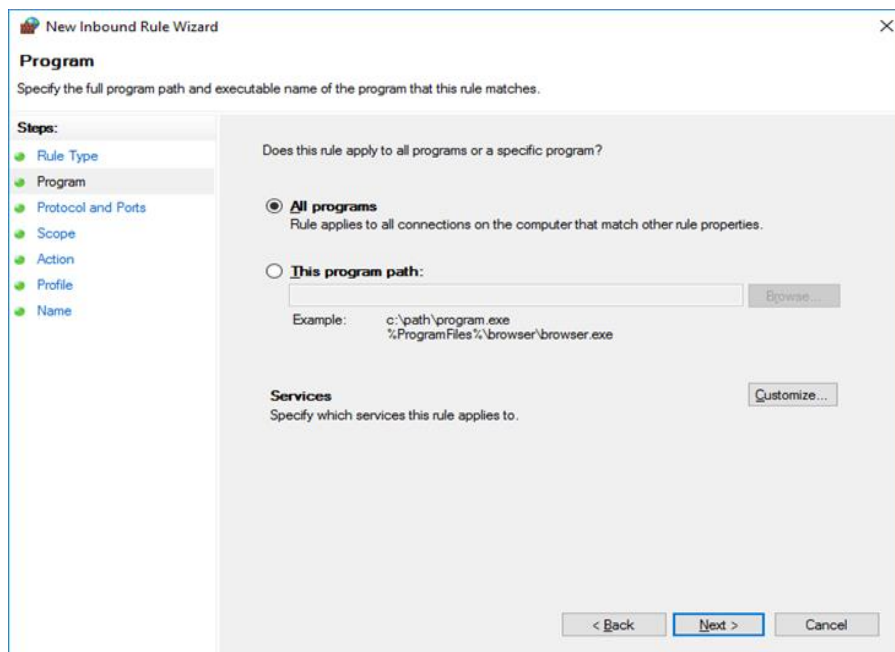


5. Create a custom rule and choose “Next”



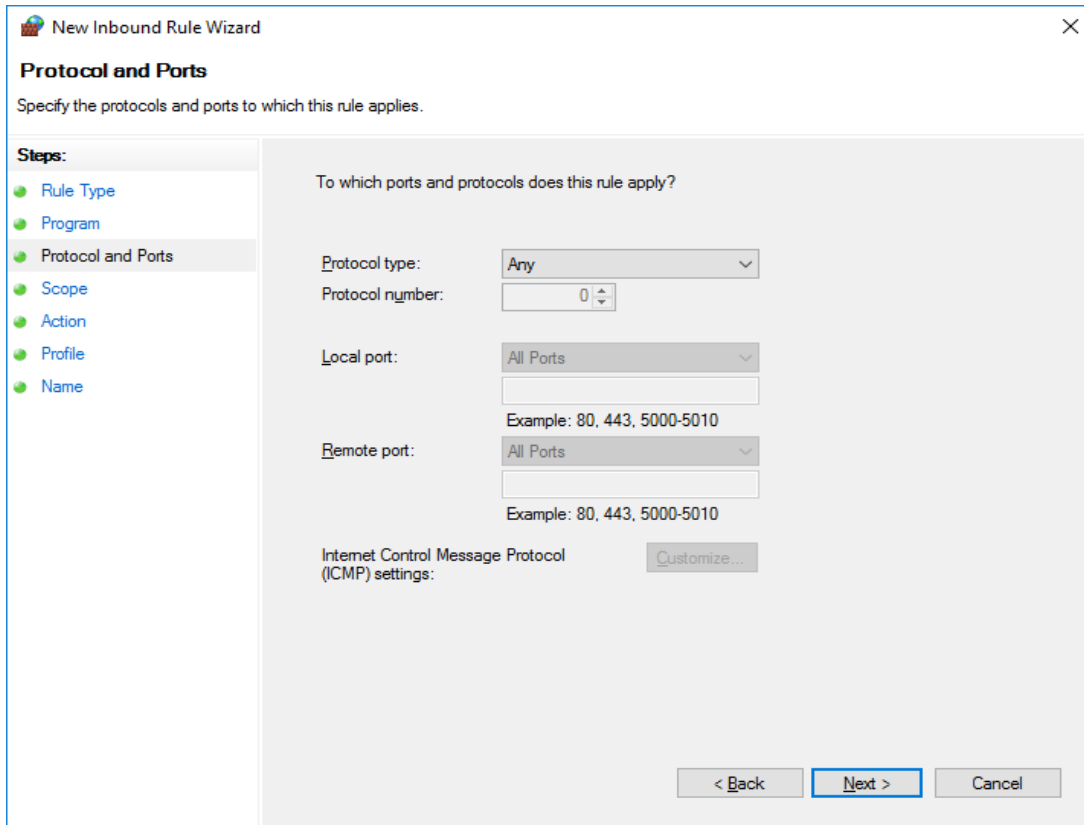
The screenshot shows the 'New Inbound Rule Wizard' window at the 'Rule Type' step. The title bar reads 'New Inbound Rule Wizard' with a close button. Below the title bar, the text 'Rule Type' is displayed, followed by the instruction 'Select the type of firewall rule to create.' On the left, a 'Steps:' pane lists 'Rule Type', 'Program', 'Protocol and Ports', 'Scope', 'Action', 'Profile', and 'Name', with 'Rule Type' selected. The main area asks 'What type of rule would you like to create?' and offers four options: 'Program' (Rule that controls connections for a program.), 'Port' (Rule that controls connections for a TCP or UDP port.), 'Predefined:' (with a dropdown menu showing 'Active Directory Domain Services' and the description 'Rule that controls connections for a Windows experience.'), and 'Custom' (Custom rule.), which is selected with a radio button. At the bottom right are '< Back', 'Next >', and 'Cancel' buttons.

6. Allow “All programs” from the tools machine and click “Next”.



The screenshot shows the 'New Inbound Rule Wizard' window at the 'Program' step. The title bar reads 'New Inbound Rule Wizard' with a close button. Below the title bar, the text 'Program' is displayed, followed by the instruction 'Specify the full program path and executable name of the program that this rule matches.' On the left, a 'Steps:' pane lists 'Rule Type', 'Program', 'Protocol and Ports', 'Scope', 'Action', 'Profile', and 'Name', with 'Program' selected. The main area asks 'Does this rule apply to all programs or a specific program?' and offers two options: 'All programs' (Rule applies to all connections on the computer that match other rule properties.), which is selected with a radio button, and 'This program path:' (with a text box and a 'Browse...' button). Below the text box is an example: 'Example: c:\path\program.exe %ProgramFiles%\browser\browser.exe'. At the bottom, there is a 'Services' section with the text 'Specify which services this rule applies to.' and a 'Customize...' button. At the bottom right are '< Back', 'Next >', and 'Cancel' buttons.

7. Allow all protocols and ports, then click "Next".

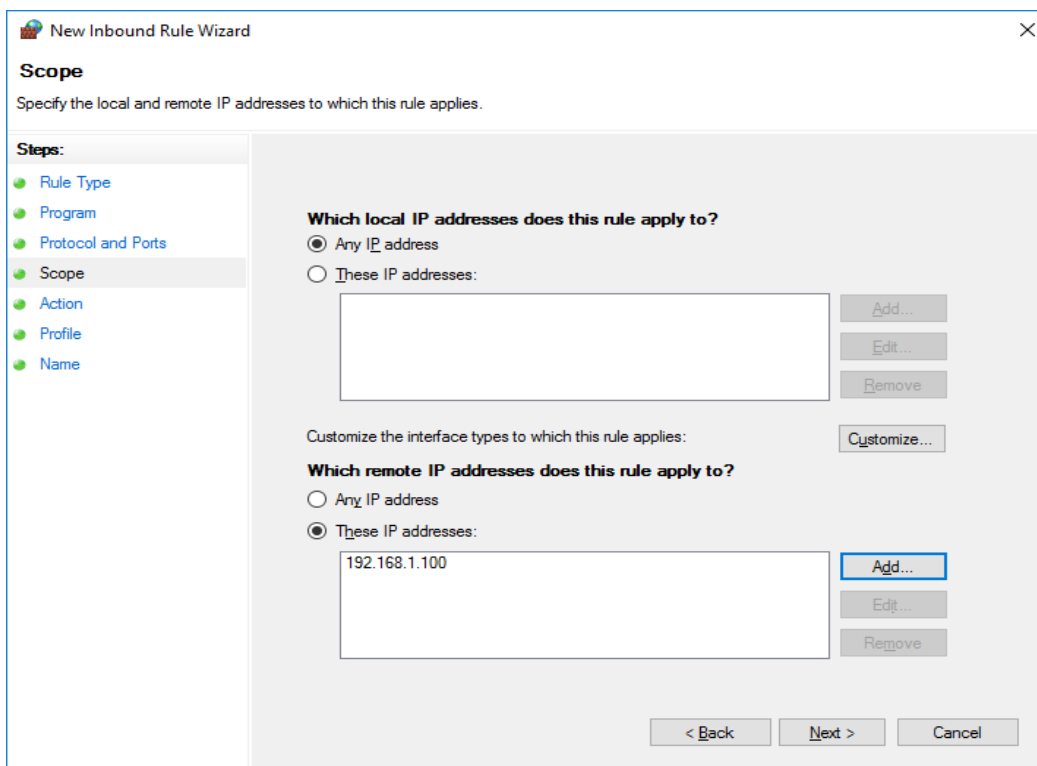


The screenshot shows the 'New Inbound Rule Wizard' window, specifically the 'Protocol and Ports' step. The left sidebar lists the steps: Rule Type, Program, Protocol and Ports (selected), Scope, Action, Profile, and Name. The main area is titled 'To which ports and protocols does this rule apply?'. It contains the following fields and controls:

- Protocol type:** A dropdown menu set to 'Any'.
- Protocol number:** A numeric input field set to '0'.
- Local port:** A dropdown menu set to 'All Ports'.
- Remote port:** A dropdown menu set to 'All Ports'.
- Example:** The text 'Example: 80, 443, 5000-5010' is displayed below both port dropdowns.
- Internet Control Message Protocol (ICMP) settings:** A section with a 'Customize...' button.

At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

8. Specify the IP address of the tools machine and click "Next".



The screenshot shows the 'New Inbound Rule Wizard' window, specifically the 'Scope' step. The left sidebar lists the steps: Rule Type, Program, Protocol and Ports, Scope (selected), Action, Profile, and Name. The main area is titled 'Specify the local and remote IP addresses to which this rule applies.' and contains the following sections and controls:

- Which local IP addresses does this rule apply to?**
 - ☒ Any IP address
 - ☐ These IP addresses: A text box for specifying IP addresses, with 'Add...', 'Edit...', and 'Remove' buttons to its right.
- Customize the interface types to which this rule applies:** A section with a 'Customize...' button.
- Which remote IP addresses does this rule apply to?**
 - ☐ Any IP address
 - ☒ These IP addresses: A text box containing '192.168.1.100', with 'Add...', 'Edit...', and 'Remove' buttons to its right.

At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

9. Choose to "Allow the connection" and click Next
10. Choose to select network profile "Domain" and click "Next"
11. Choose a name for the rule (Example: Windows ClientToolsMachine)

User Profile Service

It is necessary to modify the default behavior of the User Profile Service as it relates to user logoff. Windows, by default, forcibly unloads user registry hive on logoff even if there are applications with open handles to the user registry hive. This default behavior interferes with remote Powershell initialization routines during execution of the on-demand assessment via scheduled task and can prevent successful collection and submission of assessment data to the log analytics portal.

On the data collection machine, change the following setting in the group policy editor (gpedit.msc) from "not configured" to "enabled":

Computer Configuration->Administrative Templates->System-> User Profiles

'Do not forcefully unload the user registry at user logoff'

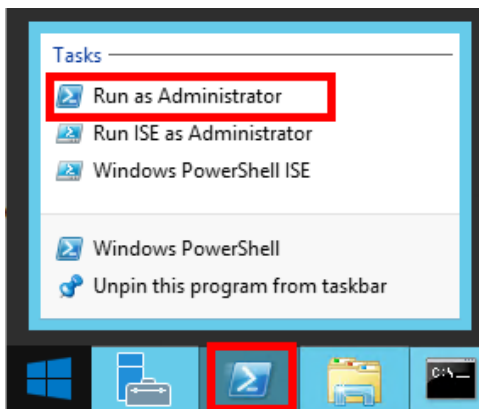
After you have finished the installation of the Microsoft Monitoring Agent/OMS Gateway, and configured Security Updates Prerequisites on the Data Collection machine and target machines, continue with the next section to set up the assessment.

Setting up the Windows Client Assessment

When you have finished the installation of the Microsoft Management Agent/OMS Gateway, you are ready to setup the Windows Client Assessment. There are two approaches to setting up the assessment scheduled task depending on whether the scheduled task account will be a managed service account or a user account (outlined in steps 2 and 3 below).

On the designated data collection machine, complete the following:

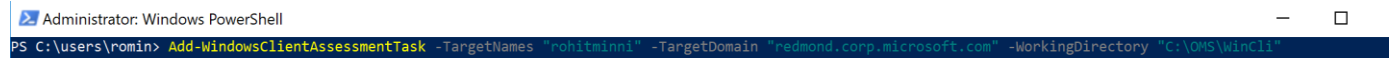
1. Open the Windows PowerShell command prompt as an Administrator



2. Using a User Account:

Run the **Add-WindowsClientAssessmentTask -TargetNames <YourClientNames> -TargetDomain <TargetDomain> -WorkingDirectory <Directory>** command where <YourClientNames> is the FQDN or NetBIOS name of one of the clients in the environment, <TargetDomain> is an optional input and specifies the domain from which target clients would be selected from and <Directory> is the path to an existing directory used to store the files created while collecting and analyzing the data from the environment.

NOTE: If the directory does not exist, it must be created before you continue with the execution



You can also import a list of computers from a text file by using the below approach:

PS C:\WINDOWS\system32> \$Clients = Get-Content "C:\Docs\ClientList.txt"
Add-WindowsClientAssessmentTask -TargetNames \$Clients -TargetDomain <TargetDomain> -WorkingDirectory "C:\OMS\WinCli"

where the text file would contain a list of multiple clients that are semicolon separated for eg: "Client01;Client02;Client03".

3. Using a Managed Service Account:

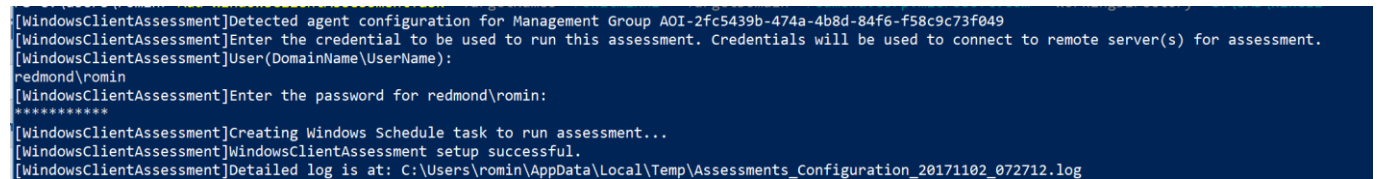
Managed service accounts are the preferred option for running the assessment due to their credential management and security related benefits over standard user accounts. Managed service accounts must be provisioned in Active Directory Domain Services and authorized in the environment.

1. Follow the instructions in the provisioning [KB article](#)
2. Authorize the account with the necessary environmental access per the User account rights section in this document. On the designated data collection machine, complete the following in an admin powershell prompt:

Add-WindowsClientAssessmentTask -TargetNames <YourClientNames> -TargetDomain <TargetDomain> -WorkingDirectory <Directory> -ScheduledTaskUsername <MSAname> -RunWithManagedServiceAccount \$True

command where <YourClientNames> is the FQDN or NetBIOS name of one of the clients in the environment, <TargetDomain> is an optional input and specifies the domain from which target clients would be selected from, <Directory> is the path to an existing directory used to store the files created while collecting and analyzing the data from the environment and <MSAname> is the SAM account name (ending with a \$ sign) of the provisioned and authorized managed service account.

4. Provide the required user account credentials. These credentials are used to run the Windows Client Assessment.



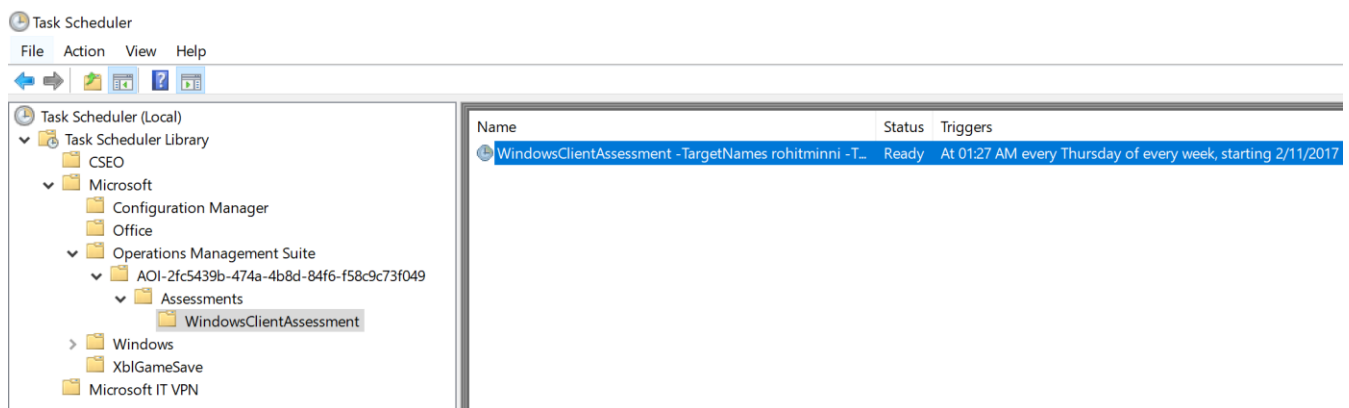
NOTE: This domain account must have all the following rights:

- Must be a local administrator on the data collection machine.
- Must be a local administrator on each of the target clients to be assessed.
- Unrestricted network access to every client to be assessed

- The script will continue with the necessary configuration. It will create a scheduled task that will trigger the data collection.

```
Administrator: Windows PowerShell
PS C:\Users\romin> Add-WindowsClientAssessmentTask -TargetNames "rohitminni" -TargetDomain "redmond.corp.microsoft.com" -WorkingDirectory "C:\OMS\WinCli"
[WindowsClientAssessment]Detected agent configuration for Management Group AOI-2fc5439b-474a-4b8d-84f6-f58c9c73f049
[WindowsClientAssessment]Enter the credential to be used to run this assessment. Credentials will be used to connect to remote server(s) for assessment.
[WindowsClientAssessment]User(DomainName\UserName):
redmond\romin
[WindowsClientAssessment]Enter the password for redmond\romin:
*****
[WindowsClientAssessment]Creating Windows Schedule task to run assessment...
[WindowsClientAssessment]WindowsClientAssessment setup successful.
[WindowsClientAssessment]Detailed log is at: C:\Users\romin\AppData\Local\Temp\Assessments_Configuration_20171102_072712.log
```

- Data collection is triggered by the **scheduled task** named "**WindowsClientAssessment**" within an hour of running the previous script and then every 7 days. The task can be modified to run on a different date/time or even forced to run immediately.

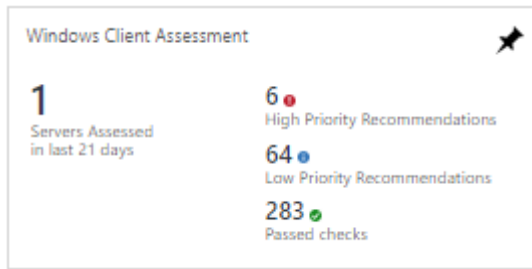


- During collection and analysis, data is temporarily stored under the **WorkingDirectory** folder that was configured during setup, using the following structure:

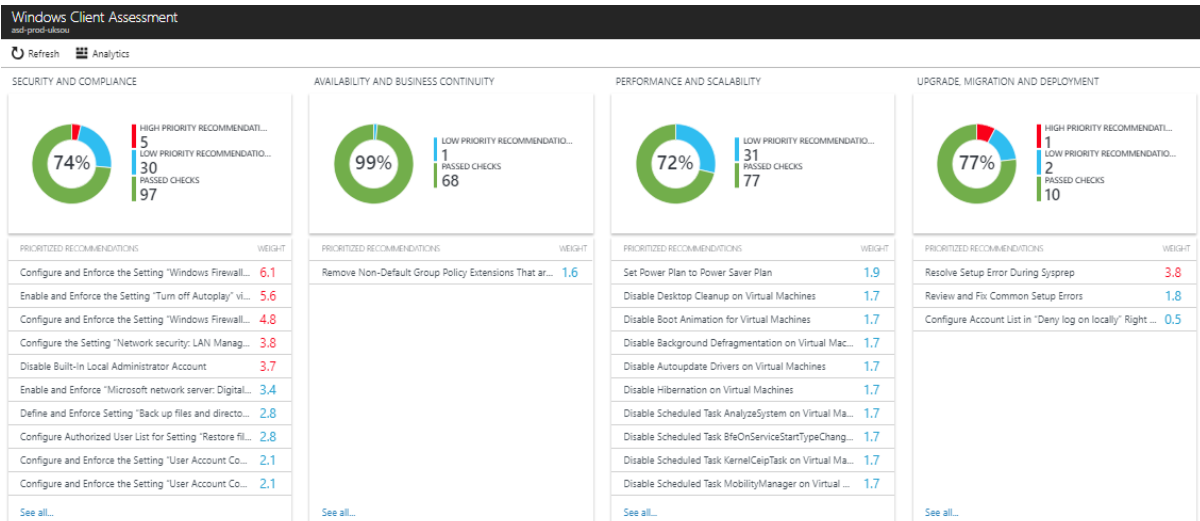
This PC > OSDisk (C:) > OMS > WinCli > WindowsClientAssessment > rohitminni

<input type="checkbox"/>	Name	Date modified	Type	Size
	3060735	2/11/2017 12:35 AM	File folder	
	OmsAssessment	2/11/2017 12:35 AM	File folder	
	run.cmd	2/11/2017 12:27 AM	Windows Command ...	1 KB

- After data collection and analysis is completed on the tools machine, it will be submitted to your log analytics workspace depending on the scenario you have chosen:
 - Directly** if the Data Collection Machine is connected to the Internet and configured to submit directly.
 - Through to the OMS Gateway Computer** if this option is configured, which will then submit the data to your log analytics workspace.
- After a few hours, your assessment results will be available on your log analytics dashboard. Click the **Windows Client Assessment** tile to review:



10. You will then be presented with findings grouped by the focus area.



Appendix

Data Collection Methods

The **Windows Client Assessment in the log analytics workspace** uses multiple data collection methods to collect information from your environment. This section describes the methods used to collect data from your environment. No Microsoft Visual Basic (VB) scripts are used to collect data.

1. Registry Collectors
2. Xperf
3. EventLogCollector
4. Windows PowerShell
5. FileDataCollector
6. WMI
7. Nltest

1. Registry Collectors

Registry keys and values are read from the data collection machine and all workstations. They include items such as:

- Service information from HKLM\SYSTEM\CurrentControlSet\Services
- Operating System information from HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion

2. XPerf

[Xperf](#) is a tool that is part of the [Windows Performance Toolkit](#) that can create boot time statistics. With Xperf the boot time is evaluated and the top 10 processes that utilize disk and/or cpu most.

3. EventLogCollector

Collects event logs from target machines. We mostly collect the last 7 days of different event logs.

4. Windows PowerShell

Collects various information, such as:

- BCD store boot configuration Data
- Defragmentation rate

5. FileDataCollector

Enumerates files in a folder on a remote machine, and optionally retrieves those files.

6. Windows Management Instrumentation (WMI) Collectors

[WMI](#) is used to collect various information such as:

- WIN32_Volume
Collects information on volume settings for each in-scope workstation. For example, the information is used to determine the system volume and drive letter, which allows the assessment to collect information on files located on the system drive.
- Win32_Process
Collect information on the processes running on each DC in the forest. The information provides insight on processes that consume a large amount of threads, memory, or have a large page file usage.
- Win32_LogicalDisk
Used to collect information on the logical disks. We use the information to determine the amount of free space on the disk where the database or log files are located.