How To Use The Nagios XI Windows SSH Wizard

Purpose

This document describes the Windows SSH Configuration Wizard for Nagios XI, a replacement for the deprecated WMI Wizard. The Windows SSH Wizard allows administrators to monitor Windows servers remotely without installing any agents on the target machine.

Target Audience

This guide is intended for system administrators who want to monitor Windows servers without installing agents on the target machines.

Supported Operating Systems

- Windows 10
- Windows 11

Disclaimer: Security Considerations

Please note that while the Windows SSH Configuration Wizard provides a convenient way to monitor Windows servers without installing agents, it is not the most secure method. We recommend using an agent such as NCPA for a more secure monitoring solution.

Using the Windows SSH Configuration Wizard may expose your servers to potential security risks. If the Nagios XI machine is compromised, all the servers that you can SSH into become vulnerable as well. It is essential to ensure that your Nagios XI machine is secure and up to date to minimize these risks.

Nagios Enterprises is not responsible for any security breaches, data loss, or other issues that may arise from using the Windows SSH Configuration Wizard. By choosing to use this Wizard, you acknowledge and accept the potential security risks involved.

For a more secure monitoring solution, consider using an agent such as NCPA, which provides encrypted communication and additional security features. For more information on NCPA, please visit the official NCPA documentation.

Monitoring Features

The Windows SSH Wizard offers the following monitoring features:

1. Ping (ICMP ping): This feature checks the availability and responsiveness of the target Windows machine by sending ICMP echo requests and measuring the response time.
2. Monitoring Disk Volume: This feature monitors the disk volume usage on the target Windows machine, helping administrators identify potential storage issues and manage disk space effectively.
3. Monitoring Memory Usage: This feature monitors the memory usage on the target Windows machine, allowing administrators to detect memory leaks or high memory consumption that may affect system performance.
4. Monitor CPU Usage: This feature monitors the CPU usage on the target Windows machine, helping administrators identify potential performance bottlenecks and optimize resource allocation.
5. Monitor Disk Usage (read/write/total operations/sec): This feature monitors the disk read, write, and total operations per second on the target Windows machine, providing insights into disk performance and potential issues.
6. Monitor Windows Services: This feature monitors the status of Windows services on the target machine, allowing administrators to detect service failures and ensure that critical services are running as expected.
7. Monitor Windows Processes: This feature monitors the status of Windows processes on the target machine, helping administrators identify potential issues with running applications and services.

Each of these monitoring features provides valuable insights into the health and performance of the target Windows machine, allowing administrators to proactively address potential issues and optimize system performance.

Windows Machine Requirements

To use the Windows SSH Wizard, the target Windows machine must meet the following requirements:

- Firewall rules must be set up to allow incoming SSH connections.
- OpenSSH must be installed and configured on the target Windows machine.

Instructions

Generating an SSH Key on the Nagios XI System

1. Log in to your Nagios XI system and switch to the nagios user by running the following in the terminal: su nagios
2. Generate an SSH key by running: ssh-keygen -t ed25519
3. Follow the prompts and save the key in the default location: /home/nagios/.ssh/

Installing OpenSSH on the Target Windows Machine

1. Open PowerShell as an administrator on the target Windows machine.
2. Install OpenSSH by running the following commands:

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

3. Verify that OpenSSH is installed by running:

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

   The output should show that both the OpenSSH Client and Server are installed.

4. Run the following commands to configure OpenSSH:

```
winrm quickconfig
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine
Restart-Service sshd
Set-Service -Name sshd -StartupType 'Automatic'
```

5. Create a firewall exception for OpenSSH by running the following commands:

   ```
   if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction
   SilentlyContinue | Select-Object Name, Enabled)) { Write-Output "Firewall Rule 'OpenSSH-
   Server-In-TCP' does not exist, creating it..."
   New-NetFirewallRule -Name 'OpenSSH-Server-In-TCP' -DisplayName 'OpenSSH
   Server (sshd)' -Enabled True -Direction Inbound -Protocol TCP -Action Allow
   -LocalPort 22
   } else {
   Write-Output "Firewall rule 'OpenSSH-Server-In-TCP' has been created and exists."
   }
   ```

6. Set PowerShell as the default shell by running the following command:

   ```
   New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value
   "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -
   Force
   ```

Configuring SSH Keys on the Target Windows Machine

Step 1: Create the authorized_keys file

1. Open PowerShell and run the following command to create the administrators_authorized_keys file:

   ```
   New-Item -Path "C:\ProgramData\ssh\administrators_authorized_keys" -ItemType File
   ```

Step 2: Edit the authorized_keys file

1. Open the file in Notepad:

   notepad C:\ProgramData\ssh\administrators_authorized_keys

2. Go back to your Nagios XI machine and display the SSH key you generated:

   cat /home/nagios/.ssh/id_ed25519.pub

3. Copy the displayed SSH key and paste it into the administrators_authorized_keys file on the Windows machine. Save and close the file.

Step 3: Verify the file extension

1. In File Explorer, ensure the file is not saved as a .txt file. If it is, go to **View > Options** and uncheck the **Hide extensions for known file types**. Rename the file to administrators_authorized_keys (without the .txt extension).

Step 4: Set file permissions

1. Right-click on administrators_authorized_keys in File Explorer and click **Properties > Security > Advanced > Change Permissions > Disable Inheritance > Remove all inherited permissions from this object**.
2. Click **Add** and then **Select a principal**. Enter the username of the Windows user. If you don't know it, go to **Task Manager > Users** to find it.
3. After adding the user, check all the permission boxes (except for **Special**). Click **OK** and then **Apply**.
4. Repeat the process for the SYSTEM user, ensuring it has full privileges. You should now have two users with all permissions.

Step 5: Test the SSH connection (required)

1. Return to the terminal on your Nagios XI machine and attempt to SSH into the Windows machine. It should prompt you for a fingerprint initially. This fingerprint must be acknowledged for the Wizard to work. Remember, you must be the nagios user.
2. If everything is set up correctly, you can now SSH into the Windows machine without a password.

Note: Finding correct user

To find the current user on the target Windows machine, follow these steps:

1. Open PowerShell on the target Windows machine.
2. Type *whoami* and press Enter.
3. The command will return the current user in the format DOMAIN\username.

Now that you have the correct user, you can test the SSH connection from your Nagios XI machine using the following command:

```
ssh <username>@<target_windows_machine_ip>
```

Replace username with the user you found using the *whoami* command and *target_windows_machine_ip* with the IP address of the target Windows machine.

If the SSH connection is successful, you will be logged into the target Windows machine without being prompted for a password. This confirms that the Windows SSH Configuration Wizard is set up correctly and ready for use.