



## Purpose

This document describes how to use Nagios XI to monitor for high priority updates from Microsoft on a remote Windows host. This includes critical and security updates, service packs, and update rollups. Monitoring for updates from Microsoft with Nagios XI allows for automated checks to alert you when a new Windows updates are released so you can apply them in a timely manner and ensure a safe and secure network environment.

## Target Audience

This document is intended for use by Nagios XI Administrators who are interested in monitoring Windows machines to determine if they require updates to ensure their network infrastructure is safe, secure, and up-to-date.

## Prerequisites

### NSClient++ and NRPE

You must have NSClient++ installed on the Windows machine you intend to monitor. NSClient++ itself does not know how to check the Windows updates status however it will be configured to execute a PowerShell script (a plugin) that will be able to check the Windows update status.

NSClient++ must be configured to allow NRPE checks from the Nagios XI server. These specific documents will show you how to install NSClient++ and configure it to accept NRPE requests:

- [Installing The XI Windows Agent](#)
- [Configuring The XI Windows Agent](#)
- [Enabling the NRPE Listener in NSClient](#)
- [Enabling the NRPE Listener in NSClient 0.4.x](#)

This guide is specifically aimed at NSClient++ v 0.4.x or newer, the previous 0.3.x version of NSClient++ is no longer supported by the developer of the application.

## PowerShell

The Windows machine also needs to have PowerShell installed for the `.ps1` script to run to report the Windows Update status. If you need to download PowerShell, access the following link. The downloads are listed at the bottom of the page:

<http://support.microsoft.com/kb/968929>

## Downloading The Required Plugin

The plugin that will be used to check for updates is found at:

<https://exchange.nagios.org/directory/Plugins/Operating-Systems/Windows/NRPE/Check-Windows-Updates-Powershell/details>

Download the plugin to your windows machine that has NSClient++ installed. Once downloaded, unzip the `Check-Updates.zip` file and put the `Check-Updates.ps1` script inside your `NSClient++/scripts` directory where it can be used for the check, usually the file path is as follows:

```
C:\Program Files\NSClient++\scripts\
```

## Configure nsclient.ini On The Windows Host

In order for you to properly use the plugin you must edit the `nsclient.ini` file in the NSClient++ directory. Open the file in your favorite text editor and locate the `[settings/external scripts/scripts]` section. If you are already using NRPE checks there should be commands listed here, if not simply add the section heading as well as the following command:

```
check_updates=cmd /c echo scripts\Check-Updates.ps1; exit $LastExitCode | powershell.exe -command -
```

It should look like this:

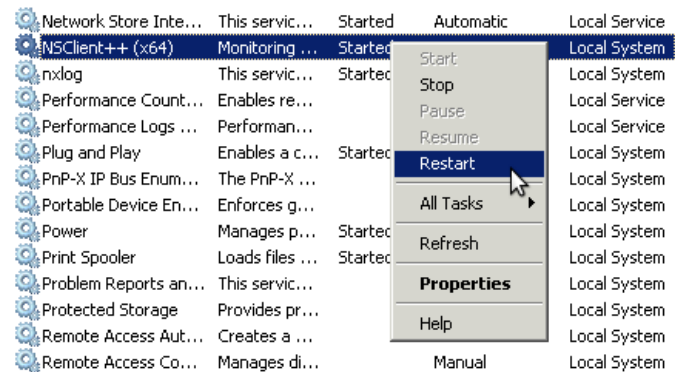
```
[/settings/external scripts/scripts]
check_updates=cmd /c echo scripts\Check-Updates.ps1; exit $LastExitCode | powershell.exe -command -
```

NSClient must now be restarted. In Windows open the **Services** console under **Administrative Tools**. If you cannot locate this, use `services.msc` to open the Services console.

Locate the NSClient++ service.

Right click the **NSClient++** service and select **Restart**.

You can close the Services console as it's no longer required.



## Set Execution Policy In PowerShell

In order for PowerShell to properly execute the `Check-Updates.ps1` script it must have permission to do so. You can change these permissions by first running PowerShell. Type into the **Start Menu** search field `PowerShell`, when it appears right click on **Windows PowerShell** and select **Run as administrator**. This will open the PowerShell command line interface.

Type the following command in PowerShell:

```
Set-ExecutionPolicy Unrestricted
```

You will be prompted to confirm the change to the execution policy, answer **Y** and press **Enter**.

This will allow all PowerShell scripts to be run by Windows. More information on the [Execution Policy commands](#) can be found at:

<http://technet.microsoft.com/en-us/library/ee176961.aspx>

Another policy option is `Bypass` and is configured a similar way:

```
Set-ExecutionPolicy Bypass
```

The next step will continue to use PowerShell so leave it open.

## Testing The Plugin From The Windows Command Line

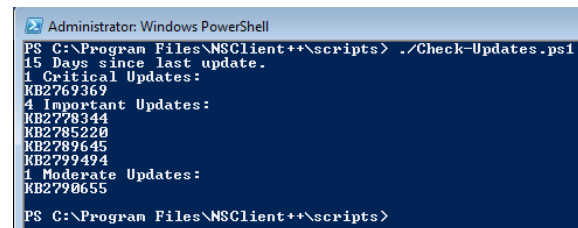
You are now ready to test the PowerShell script. First the directory must be changed to the `NSClient++\scripts` directory, execute the following command:

```
cd "C:\Program Files\NSClient++\scripts"
```

Next we will run the command to check for Windows Updates (the plugin you downloaded above). This may take a bit of time to run depending on how many pending updates exist. Run the following command:

```
./Check-Updates.ps1
```

If everything was configured correctly the return output will be similar to this:



```
Administrator: Windows PowerShell
PS C:\Program Files\NSClient++\scripts> ./Check-Updates.ps1
15 Days since last update.
1 Critical Updates:
KB2769369
4 Important Updates:
KB2778344
KB2785220
KB2789645
KB2799494
1 Moderate Updates:
KB2790655
PS C:\Program Files\NSClient++\scripts>
```

## Testing The Check From The Nagios XI Server

Now that we know the plugin works on our windows machine, we will test the command from the Nagios XI server. Login to your Nagios XI server as the root user and change the directory to the location of the `check_nrpe` plugin:

```
cd /usr/local/nagios/libexec
```

Next enter the command that will be used to run the NRPE check. You will need to replace `<Remote Windows IP address>` with the IP address of your remote windows machine:

```
./check_nrpe -H <Remote Windows IP address> -t 120 -p 5666 -c check_updates
```

If everything was configured correctly the return will be similar to this:

```
15 Days since last update.
1 Critical Updates:
KB2769369
4 Important Updates:
KB2778344
KB2785220
KB2789645
KB2799494
1 Moderate Updates:
KB2790655
```

## Creating The Check In Nagios XI

Now the check must be configured in the Nagios XI Web Interface using Core Configuration Manager (CCM).

The first step will be to create a custom command specifically for this check.

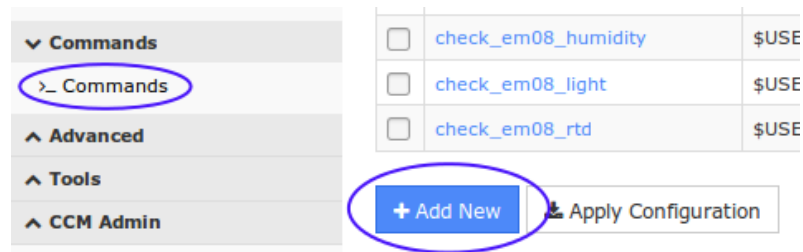
### Create Check Command

Navigate to **Configure > Core Config Manager**

In the left pane expand **Commands** and then click

**>\_ Commands**

Click the **Add New** button



The Command Management page will open, populate the fields with the following values:

Command Name:

check\_updates

Command Line:

\$USER1\$/check\_nrpe -H \$HOSTADDRESS\$ -t 120 -c check\_updates \$ARG1\$ \$ARG2\$

Command Type:

check command

Active:

checked

Click the **Save** button to create this new command.

Here is a screenshot that shows the command definition.

## Command Management

### Command Name \*

Example: check\_example

### Command Line \*

Example: \$USER1\$/check\_example -H \$HOSTADDRESS\$ -P \$ARG1\$ \$ARG2\$

### Command Type:

Active ?

### Available Plugins

 ?



## Create Service

The final step is to create a new service definition that is associated with the remote windows host. It is assumed that you are already monitoring the Windows host and there is a HOST object already created. If there isn't, go and run the Windows Server Configuration Wizard and then return to this step. This guide is going to use the host 10.25.14.52 as an example.

In the left pane expand **Monitoring** and then click **Services**.

Click the **Add New** button.

The screenshot shows the Nagios XI interface. The left sidebar is expanded to 'Monitoring' > 'Services'. The 'Add New' button is circled in blue. The main content area shows a table of services for host 10.25.14.52.

<input type="checkbox"/>	Service Name	Service Description
<input type="checkbox"/>	10.25.14.52	CPU Usage
<input type="checkbox"/>	10.25.14.52	Drive C: Disk Usage
<input type="checkbox"/>	10.25.14.52	Drive D: Disk Usage
<input type="checkbox"/>	10.25.14.52	Memory Usage
<input type="checkbox"/>	10.25.14.52	Uptime

At the bottom, there is an 'Add New' button (circled in blue), an '& Apply Configuration' button, and a 'With checked' dropdown menu with a 'Go' button.

**Common Settings** tab

Config Name:

10.25.14.52

Description:

**Windows Update Status**Click the **Manage Hosts** buttonSelect 10.25.14.52 in the left pane and click the **Add Selected >** buttonClick the **Close** button

For this service we will use the generic-service template as it has a lot of the required directives already configured

Click the **Manage Templates** buttonSelect **generic-service** in the left pane and click the **Add Selected >** buttonClick the **Close** buttonCheck command (*drop down list*)**check\_updates**

Active:

**Checked****Service Management**

Common Settings	Check Settings	Alert Settings	Misc Settings
<b>Config Name *</b> <input type="text" value="10.25.14.52"/>		<b>Check command</b> <input type="text" value="check_updates"/>	
<b>Description *</b> <input type="text" value="Windows Update Status"/>		<b>Command view</b> <pre>\$USER1\$/check_nrpe -H \$HOSTADDRESS\$ -t 120 -c check_updates \$ARG1\$ \$ARG2\$</pre>	
<b>Display name</b> <input type="text"/>		<input type="text" value="\$ARG1\$"/> <input type="text" value="\$ARG2\$"/> <input type="text" value="\$ARG3\$"/>	
<input type="button" value="Manage Hosts"/>			
<input type="button" value="Manage Templates"/>			



**Check Settings** tab

Check interval:

1440

Retry interval:

20

Max check attempts:

3

Click the **Save** buttonClick the **Apply Configuration** button at the bottom of the screen.




You may have noticed that the check interval is set to **1440**. There are 1440 minutes in a day, hence this setting causes the check to only run once a day, running any more frequently isn't really required. You could also take this a step further and create a custom **Time Period** which would restrict the check from only being run in the early hours of the morning.

**Service Management**
 Common Settings
  Check Settings
  Alert Settings
**Initial state**
 Warning
  Critical
  Ok
  Unreachable
**Check interval**
 min
**Retry interval**
 min
**Max check attempts**
 attempts



## End Result

Now that the service has been created, navigate to **Home > Service Detail** and search for the service. If the check has been configured correctly, you should see a result like the one below.

Host	Service	Status	Duration	Attempt	Last Check	Status Information
10.25.14.52   	Windows Update Status	Critical	1h 46m 34s	3/3	2016-10-28 15:43:25	0 Days since last update.

From the Critical status it looks like there has never been any Windows updates installed on this computer!

This is a good example of how you would be notified that there are pending Windows updates that need to be installed.

## Finishing Up

This completes the documentation on checking for Windows updates with Nagios XI.

If you have additional questions or other support related questions, please visit us at our Nagios Support Forums:

<https://support.nagios.com/forum>

The Nagios Support Knowledgebase is also a great support resource:

<https://support.nagios.com/kb>