

POLLER on OS Windows (Installation)

Context

Having a Poller running directly on a Windows system can be interesting to take advantage of Windows rights and environment.

For example, a request launched with WMI (with the check_wmic.exe) will be able to use the user rights corresponding to the user that started the service and not need a password for the request/

- On a Active Directory environment, if you start the Poller service with an administrator account, you will be able to send requests from this Poller without needing a password.
- The new Windows Poller can be tagged to get only checks corresponding to Windows specific commands.

The following procedure describes the installation and update process for the Windows Poller.

Installation

Getting and extraction data

Shinken Enterprise delivery comes with a **.tar.gz** archive specifically for the Windows Poller.

The first step is to connect with an Administrative account on the Windows server and to extract the **.tar.gz** archive (with 7zip for example).

Dependencies installation

To install the Windows Poller, you first have to install the following dependencies on your system (contained in the root of the archive):

- python-2.7.6.amd64.msi [**install first**]
- pywin32-217.win-amd64-py2.7.exe
- pycurl-7.19.3.1.win-amd64-py2.7.exe
- egenix-pyopenssl-0.13.3.1.0.1.6.win-amd64-py2.7.msi
- psutil-5.2.1.win-amd64-py2.7.exe

Note : leave default installation paths

Manual installation

- CherryPy-3.2.4 installation:
 - Open DOS command prompt with **Administrator** rights
 - Move into the **CherryPy-3.2.4** folder
 - Execute the command: **C:\python27\python.exe setup.py install**
- From the Windows explorer :
 - Copy the "shinken" directory to **C:\shinken**
 - Copy **tools\srvany.exe** dans **c:\shinken\srvany[replace file]**
 - Copy **context.json** dans **c:\shinken\var[replace file]**

Installation of the automatic Windows service

Open a DOS command line with **Administrator** rights and launch the command:

- **sc create "Shinken-Poller" binpath= "c:\shinken\srvany\srvany.exe" DisplayName= "Shinken-Poller"**
 - (*WARNING: spaces after = are required*)

From Windows explorer :

- Import the **poller.reg** file into the registry by double-clicking it
- From the MMC console of the Windows services (accessible from Administration Tools ou by launching **services.msc** directly), change the Shinken-Poller service to **"Automatic"** in the service properties and **change the account** that launches the services (via a **local or domain administrator account**)

Sommaire

- [Context](#)
- [Installation](#)
 - [Getting and extraction data](#)
 - [Dependencies installation](#)
 - [Manual installation](#)
 - [Installation of the automatic Windows service](#)
- [Updating a Windows Poller](#)
 - [Stopping the Poller daemon](#)
 - [Backup](#)
 - [Copying source files](#)
 - [Restarting the Poller daemon](#)
- [Troubleshooting](#)
 - [SSL configuration](#)
 - [Manual start of the Poller \(for testing\)](#)
 - [Network](#)
 - [User rights](#)
 - [Poller limitations](#)

- Start the service by clicking "**Start**" on the Shinken-Poller line in the MMC console, or launch the following command: **net start Shinken-Poller**.
- Check the service has been launched correctly:
 - 1/ Log file: *C:\shinken\var\pollerd.log*
 - 2/ The port **7771** must be open. This can be checked with following command: **netstat -an**
 - 3/ Once the new Windows Poller declared in the configuration on the Arbiter server, launch the **shinken-healthcheck** command. Your pollers should be valid and accessible.
 - Note : one or two minutes can pass before all the configuration has been dispatched and taken into account by the daemons

Updating a Windows Poller

The update of a Windows Poller is manual. On a Windows server, extract the archive and follow the following steps.

Stopping the Poller daemon

From the Services management tool, stop the "Shinken-Poller" service, or via the command line:

- **net stop Shinken-Poller**

From the Windows task manager check that the "python.exe" process linked to the Shinken service are not present anymore. If after stopping the service "python.exe" are still present, they can be killed.

Backup

Because manipulations on the Poller files can lead to errors, we advise to proceed to a complete backup of the **C:\shinken** directory before upgrading.

Copying source files

- Copy the **context.json** file (which is at the root of the extracted archive) to **C:\shinken\var [replace file]**

You have to delete the old Shinken source files by deleting the following folder:

- **C:\shinken\shinken**

WARNING: Do not delete the entire C:\shinken folder, but delete the "shinken" folder that is INSIDE C:\shinken.

Copy the new Shinken source files:

- *windows - XXX\shinken\shinken* to copy to *c:\shinken\shinken*
- Check that the *c:\shinken\shinken\daemon.py* file is still present (to check files have been copied to the correct place).

Restarting the Poller daemon

From the Windows Service management console, start the "**Shinken-Poller**" service, or from the command line with Administrator rights:

- **net start Shinken-Poller**

Troubleshooting

SSL configuration

To configure communications to use SSL, you have to edit the **C:\shinken\etc\daemons\pollerd-windows.ini** file and change the following block:

```
#-- HTTPS configuration --
```

You can then enable SSL and configure your certificates.

Manual start of the Poller (for testing)

If the Windows service does not start, you can manually launch the Shinken Poller to have further information. Open a command prompt with Administrator rights and launch the command:

```
C:\Python27\python.exe c:\shinken\bin\shinken-poller.py -c c:\shinken\etc\daemons\pollerd-windows.ini
```

Network

Check that the connection between the other Shinken daemons and this new Windows Poller is working correctly. A firewall might block important communications, that can cause issues between the Shinken daemons.

The 7771 listening port has to be open on the Windows Poller machine?

If needed, according to the daemons configuration, name resolution has to allow the Windows Poller to contact the other daemons (and vice versa).

User rights

During the installations, make sure you are connected with a local or domain administrator account.

Poller limitations

On Linux machines, to avoid overloading the machine, the Poller limits its checks executions according to CPU and memory consumption. On Windows, due to technical limitations, the Poller cannot use the CPU measurements to limit its checks execution rate. As a consequence, the "max_cpu_queue_per_cpu" parameter in the Poller configuration will be taken into account.

However, the Windows Poller can still limit its execution rate by checking memory consumption.