

nu774 / sudo-for-cygwin

- <> Code
- Issues 8
- Pull requests
- Actions
- Projects
- Wiki
- Security
- Insights

This repository has been archived by the owner on Jan 4, 2024. It is now read-only.

sudo-for-cygwin

Public archive

1 Branch

2 Tags

Go to file

t

Go to file

Code

About

DanKaplanSES

Update README.rst

2a8c7d3 · 5 months ago

<div></div> README.rst	Update README.rst	5 months ago
<div></div> sudo	fix EOL	9 years ago
<div></div> sudoserver.py	fix EOL	9 years ago

poor sudo clone for cygwin

sites.google.com/site/qaacpage/

Readme

Activity

72 stars

16 watching

17 forks

Report repository

Releases

2 tags

Packages

No packages published

You can use this like:

```
$ sudo vim /etc/hosts
$ sudo cp foo.txt /cygdrive/c/Program Files/
$ suco cygstart cmd # open elevated standard command prompt
$ sudo cygstart regedit
$ sudo # just invoke elevated shell
```



This might be handy if you are running cygwin on Vista or Windows 7 with UAC. By this program, you can run processes as an administrator, from normal, non-elevated cygwin shell.

Caution

UAC Elevation is usually done through UI prompt for good reasons. By this program, you can run elevated process without UI prompt that does not go along well with Cygwin shell environment. However, it also means that you are weakening the system in terms of security.

How it works

This is in fact a client/server application.

It looks as if the child process is running in the current terminal. However, in fact, it's invoked by the server, and running remotely (though "remote" is in the same PC).

You must launch a python script named **sudoserver.py** beforehand, in desired privileges. If you want function like "Run as administrator", just run **sudoserver** as administrator. For this purpose, Windows built-in Task Scheduler is handy.

Contributors 3



nu774



DanKaplanSES Daniel Kaplan



weinhold Ingo Weinhold

Languages

● Python 100.0%

sudoserver.py opens a listening port 127.0.0.1:7070 (by defaults), then sits and wait for connections from **sudo**.

sudo, when invoked, connects to the **sudoserver**. Then it sends it's command line arguments, environment variables, current working directory, and terminal window size, to the **sudoserver**.

When **sudoserver** accepts connection from **sudo**, **sudoserver** forks a child process with pty, set up environments, current working directory or something, then execute the process.

The child process is spawned by the **sudoserver**, therefore it runs in the privileges same as the server.

And, as the child process runs in a pty, it *acts* as if running in ordinary terminals. Therefore you can run cygwin's interactive console-based program like vim or less.

After execution, **sudo** and **sudoserver** bridges user's tty and the process I/O.

Requirement

Both sudo and sudoserver.py is written in python, therefore you need to install Python.

Also, you need Python module named greenlet, and eventlet. These are not packaged in cygwin, therefore you must manually install them.

How to setup

1. Install python with cygwin installer.
2. Download greenlet. It can be downloaded from <http://pypi.python.org/pypi/greenlet/>

3. Download eventlet. It can be downloaded from <http://pypi.python.org/pypi/eventlet/>
4. If you don't have setuptools installed, you also need it. <https://pypi.python.org/pypi/setuptools>
5. Install greenlet package. Extract the archive, and cd to the directory. then you type in the cygwin shell:

📖 README

6. Install eventlet package. Extract the archive, and do the same with the above instruction for greenlet. If this doesn't work, probably you need setuptools. Download setuptools and install it. setuptools can be installed in same way as greenlet.
7. You can place sudo and sudoserver.py where you like. You will want to execute sudo via command line, therefore /usr/local/bin or somewhere in the PATH will be good.
8. If you want to use the TCP portnumber other than 7070 (default value), you have to edit the both script manually. It is written like:

```
PORT = 7070
```



9. At first, probably you want to test it. From cygwin shell, invoke sudoserver.py like:

```
$ /path/to/sudoserver.py
```



10. And then, test sudo command like:

```
$ sudo ls -l
```



11. If it seems to work, you can register `sudoserver.py` to the Windows task scheduler. I recommend you the following setup.

- Action: "Start a program"
- Triggers: "At log on"
- "Run with highest privileges": checked.
- "Run only when user is logged on": checked.
- "Program/script": `C:\cygwin\bin\python.exe`
- "Add arguments(optional)": `/path/to/sudoserver.py -nw`

Notes

With argument `"-nw"` is specified, **sudoserver** hides it's console window.

sudoserver sets an additional environment variable `"ELEVATED_SHELL"` when spawning child processes. You can use this variable for changing your elevated shell prompt (PS1), to see which environment you are in. For example, you can put the following in your `.bashrc`:

```
case $ELEVATED_SHELL in
1) PS1='\[\033[31m\][\u@\h]#\[\033[0m\] ';; # elevated
*) PS1='\[\033[32m\][\u@\h]$[\033[0m\] ';;
esac
```

