

syscon 0.8

syscon 0.8 documentation

July 27, 2011

Contents

1	Introduction	2
2	Usage	3
2.1	Command Line Usage	3
2.1.1	Client	3
2.1.2	Control	3
2.1.3	Screenshots	4
2.2	GUI Usage	5
2.2.1	Client	5
2.2.2	Control	5
A	Changelog	7

Chapter 1

Introduction

syscon is a python package allowing you to control another computer remotely using the local network. It uses the Python AES implementation of Josh Davis (<http://www.josh-davis.org/pythonAES>) to encrypt the commands and the replies of the remote machine. The image data is not encrypted. For key exchange, syscon uses the Diffie-Hellman-Merkle-Algorithm.

Important Note: If your country does not allow strong cryptography, you should use syscon-nc, which has the same functionality except from the cryptography tools.

Chapter 2

Usage

`syscon 0.8` can be used both from the command line and from a graphical user interface (GUI). On the command line, you can use as many connections as you like. Because the graphical user interface would mix up the screenshots of the different connections, you can use only one connection in the GUI.

2.1 Command Line Usage

2.1.1 Client

On the remote machine, you just need to execute the `client.py`. If you want to have a “silent” remote control, you can rename the `client.py` to `client.pyw`, so there is no console window.

2.1.2 Control

For controlling the remote computer from a python console, you need to import the *`syscon.control`* module.

```
from syscon import control
```

Then you need to connect to the remote machine. For this, you define a new *`Connection`* object:

```
con = control.Connection('ComputerName')
```

with `'ComputerName'` being a string containing the name of the remote machine.

Now, the *Connection* object can be used to send commands to the remote machine:

```
con.send(command, parameter1, parameter2)
```

command specifies the command name, parameter 1 and 2 are optional, their meaning depends on the command:

COMMAND	PARAMETER 1	PARAMETER 2	DOES WHAT?
execpy	python code	-	executes the python code on the remote machine with <i>exec</i>
returnvalue	python code	-	returns the value of the python expression when evaluated
executefile	file path	-	executes the file specified in the first parameter.
download	local path	URL	downloads the file from the URL specified in the second parameter to the local path in the first parameter
screenshot	port number	-	Deprecated: See 2.1.3 - Screenshots
shutdown	-	-	Start the shutdown process on the remote machine (works only with Windows NT to 7)
logoff	-	-	Logs the current user off (also only Windows NT to 7)
newconnect	-	-	Allows to connect to the same client with another command line or the GUI (with another key).

2.1.3 Screenshots

syscon also has the ability to make screenshots of the remote computer. To use this function, you need to start the screen.py on the controlling computer and enter a path name where the images will be saved. To make a screenshot, you call the *screenshot()* function of the *Connection* object. Also, you could use the screenshot command with the port number 56000 - it's more complicated, that's why the screenshot command is deprecated... To make screenshots, the Python Imaging Library (PIL), a python package with imaging functions, needs to be installed on the remote machine. It is available to download from <http://www.pythonware.com/products/pil/> .

2.2 GUI Usage

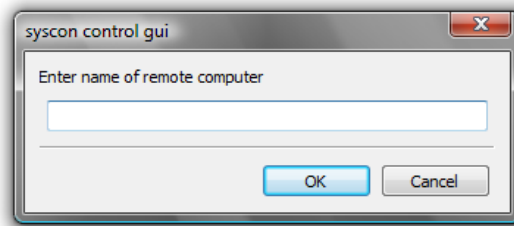
syscon 0.8 comes with a graphical user interface (GUI).

2.2.1 Client

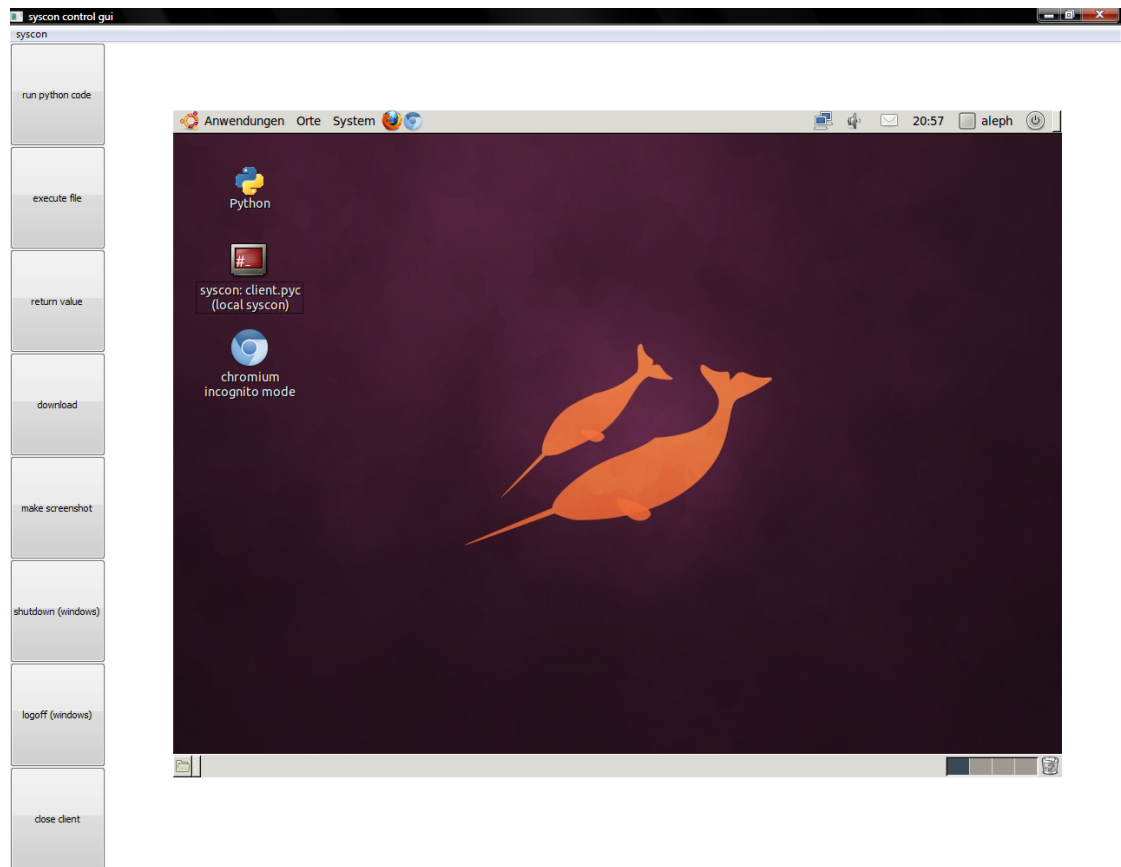
On the remote machine, you just have to execute the `client.py` (or `client.pyw`), just as if you used a python console. Because the GUI uses the screenshot functionality, the Python Imaging Library must be installed on the client computer.

2.2.2 Control

The controlling computer needs to have wxPython installed (<http://www.wxpython.org>) in order to be able to show the GUI. Since the screenshot function is used, the `screen.py` must be executed before starting the GUI. To start the graphical user interface, just execute the `control.gui.py`.



Enter the name of the remote machine



Main window.

Click on any of the buttons and the chosen command will be executed, if there are parameters needed, you will be asked to enter them. When you click on the screenshot on the right, it's important the window has exactly the size at which there is neither a grey border round the image nor the image is only partially shown because of a too small window. Then a mouse click is simulated on the remote computer. To use this function on remotely controlled linux systems, you need to install the **xautomation** program package there.

Appendix A

Changelog

to 0.8

added GUI
bugfix of major security breach (!)

to 0.4

added encryption
added “returnvalue” command

to 0.3.1

bugfixes (work on linux systems, usage without PIL)

0.3

initial release