OpenPnPy

Release 0.0.1

Sacha Boudjema

Aug 09, 2021

CONTENTS

CHAPTER

ONE

WHAT IS OPENPNPY?

OpenPnPy is a python package implementing Cisco's Network PnP protocol. It provides a server class, as well as utility functions to easily build PnP XML service messages.

To understand what Cisco Network PnP is all about, see https://developer.cisco.com/docs/network-plug-n-play/

CHAPTER

TWO

BASIC USAGE

A PnP server with custom behavior can be defined by subclassing the *openpnpy.server.PnpServer* class, and implementing the handler methods.

Service messages to be returned by the handler methods can be built using functions provided by the *openpnpy.messages* module.

A very basic server implementation could look like this:

```
# myserver.py

#!/usr/bin/env python3

from openpnpy.server import PnpServer
from openpnpy.messages import device_info, backoff, bye


class MyServer(PnpServer):

    def handle_work_request(self, work_request):
        return device_info(type='all')

    def handle_work_response(self, work_response):
        if work_response.success:
            return bye()
        else:
            return backoff(seconds=30)


server = MyServer(__name__)


if __name__ == '__main__':
    server.run(host='0.0.0.0', port='1234')
```

Which can then be run like so:

```
python3 myserver.py
```

2.1 API Reference

2.1.1 openpnpy.messages

class openpnpy.messages.**PnpMessage**(*element*)
Bases: object

PnP protocol XML message

Parameters **element** (*xml.etree.ElementTree.Element*) – XML parsed message element

property **body**
Body of the PnP message. Can be one of info or request or response messages for various PnP services.

property **correlator**
A unique string to match requests and responses between agent and server.

classmethod **from_string**(*xmlstring*)
Create an instance of PnpMessage from an XML string.

make_reply(*element*)
Builds a response to this message instance preserving session related attributes.

Parameters **element** (*xml.etree.ElementTree.Element*) – Response message body, as can be built using the openpnpy.services module

Returns New PnP message with given body

Return type openpnpy.server.PnpMessage

property **password**
Login password for the device. Applicable for all the messages that are sent from the PnP server to the agent. The only exception is the initial exchange when there is no configuration present on the new device.

property **success**
Success state of the last executed work request.

Returns Success state, None if the message is not a work response

Return type bool, None

to_string()
Returns the message as an XML string.

property **udi**
Unique Device Identifier. A built in device id consisting of product id, version id, and the serial number. It is mandatory for all the messages that get exchanged between the PnP agent and the PnP server.

property **username**
Login username for the device. Applicable for all the messages that are sent from the PnP server to the agent. The only exception is the initial exchange when there is no configuration present on the new device.

2.1.2 openpnpy.server

This module provides a base server class to be derived and implemented according to the user's needs.

class openpnpy.server.**PnpServer**(**args*, ***kwargs*)

Bases: object

Wrapped Flask web server implementing PnP protocol endpoints. This class should be derived and the handler methods overloaded to implement behavior. See https://flask.palletsprojects.com/en/2.0.x/api/#application-object for supported constructor params.

handle_hello()

What to do when a PnP Agent calls in to the server. Should not be overridden in most cases.

Returns HTTP response tuple as specified by Flask, defaults to empty http 200 by default.

Return type tuple

handle_work_request(*work_request*)

What to do when a PnP agent sends a work request.

Parameters **work_request** (*openpnpy.server.PnpMessage*) – Work request sent by the PnP agent

Returns Body element to be used in the repsonse message, as can be built using the openpnpy.services module

Return type xml.etree.ElementTree.Element

Raises **NotImplementedError** – To be implemented by user subclass

handle_work_response(*work_response*)

What to do when a PnP agent sends a work response.

Parameters **work_response** (*openpnpy.server.PnpMessage*) – Work response sent by the PnP agent

Returns Body element to be used in the repsonse message, as can be built using the openpnpy.services module

Return type xml.etree.ElementTree.Element

Raises **NotImplementedError** – To be implemented by user subclass

reply(*handler*)

Decorator to create replies to the PnP request currently held in the global request object, using the body returned by the handler method.

run(**args*, ***kwargs*)

Runs the Flask development server. See https://flask.palletsprojects.com/en/2.0.x/api/#flask.Flask.run for supported params

PYTHON MODULE INDEX

O

INDEX