Tracemac Documentation

Release 2.0.0

Mattias Måhl

April 26, 2016

CONTENTS

Contents:

Specifications of modules in project Tracemac.

Modules:

CHAPTER

ONE

SOURCES

1.1 libs

1.1.1 Switch_Object module

Created 2013-08-27

@author Mattias Måhl

Class Switch_Object

This is an object to store Switchdata in.

class Switch_Object.**Sw_Object**(**ipaddress*)
Bases: object

Initilization of the Switch Object setting up the switch and getting the information from the switch through SNMP.

Defining the tuples and variables to store the switch data.

append_mac_to_interface(*interface*, *macaddress*)
function append_mac_to_interface

a function to add mac-address to a specific interface. if interface not found add new interface and add the mac_address to it.

append_neighbor(*interface*, **args*)
function append_neighbor

function to append Neighbor to the switchobjects array och neighbors.

check_if_alive(*ipaddress*)
function check_if_alive

function to ping host to see if it's alive.

raises error if it's a fail.

check_ip_address(*ipaddress*)
function check_ip_address

function to check if the ip-address provided is acceptable.

i.e. number(dot)number(dot)number(dot)number

return False if not and returns the same string if True

class **cl_switch_interface**(*interface*, *mac_address*)
Bases: object

class cl_switch_interface

Switch_objects interface object to store mac-adresses assosiated with the interface.

class Sw_Object.**cl_switch_neighbors**(*interface*)
Bases: object

class cl_switch_neighbores

Switch_Objects neighbors object to store the switches neighbors and witch interface their on.

interface = 0

ip_address = ‘’

name = ‘’

remote_interface = 0

Sw_Object.**find_my_mac_address**(*mac_address*)
function to search the Mac-table of the switch to find a specific MAC-adress.

Sw_Object.**find_switch_mac_address**()
function find_switch_mac_address()

Do snmp request för oid: 1.3.6.1.4.1.11.2.14.11.5.1.1.6.0 (BaseMacAddress)

Sw_Object.**get_interface**(*interface*)
function get_interface

searches registered interfaces and returns the interface_object

Sw_Object.**get_mac_address_list**()
function get_mac_address_list

Does a SNMP-request to gather the mac-address-table from this switch-object.

Sw_Object.**get_neighbor_at_interface**(*interface*)
function get_neighbor_at_interface

function to get the neighbor at a specific interface.

searches registered neighbors and returns a hit.

Sw_Object.**get_neighbors**()
function get_neighbores

gets the list of neighbors and stores them in the list ‘switch_neighbors’

Sw_Object.**get_switch_data**()
Do snmp-recuest to get the system name of the target switch.

Sw_Object.**is_interface_a_neighbor**(*interface*)
function is_interface_a_neighbor

Check to see if the interface has neighbors registered.

Returns True or False

exception Switch_Object.**sw_error**(*error_msg*)
Bases: Exception

1.1.2 Trace_Functions module

Created 2013-08-27

@author Mattias Måhl

Class Tracefunctions

Functions to administrate search for MAC-adress

class Trace_Functions.**Tracefunctions**
Bases: object

class **Trace_arguments**
Bases: object

Class Trace_arguments Object to store the supplied arguments.

dump_file = ''

in_file = ''

start_ip_address = ''

target_ip_address = ''

target_mac_address = ''

verbose = False

class Tracefunctions.**Trace_result**
Bases: object

Class Trace_result Object to store a single result from the search. This stores the path the program took to find the port witch has the mac-adress.

SW_O = []

failed = False

search_ip = ''

search_mac = ''

trace_end = ''

Tracefunctions.**chk_system_args**(*argv*)
function chk_system_args function to check if argumest supplied are correct and that mandatory argurments are supplied.

Tracefunctions.**fix_macaddress**(*MAC*)
function fix_macaddres this function fixes the mac-address to be exactly the same even if the user supplies it in different formats. i.e 121212-121212 will become 121212121212 and likewise 12:12:12:12:12:12 will become 121212121212.

Tracefunctions.**get_mac_address_from_ip**(*ipaddress*)
function get_mac_address_from_ip function to arping an ip address to get the Mac-address assosiated woth it. It's important that the target ip address is on the same network as the machine running the program and NOT routed! If it's routed the routers mac address will be the one reported by the arping!

Tracefunctions.**get_system_args**(*argv*)
function get_system_args function to parse system arguments in cli-envioronment. Args:

-h, –help= Display helptext for cli-command.

-i, –ipaddress= Target ip-address to find in the network. *note* this implies access to both mgmnt- and target network (if separate)

-s, –startingip= IP-address of the first switch in the cascade.

-o, –out= Output logging top specified file.

-m, –macaddress= Target MAC-address to find in the network.

-v, –verbose= Enebles verbose output to standard output and logging file if '-o/-out= is used.

-f, --in-file Enables function to loop through a list of targets in a text file. *note* One target per line.

`Tracefunctions.`**`ping_my_address`**(*ipaddress*, *cnt*)
function ping_my_address simple function to ping the target IP-address to keep the Mac-address table up-to-date.

`Tracefunctions.`**`printhelp`**`()`
function printhelp duh!

1.1.3 frm_main module

Created 2016-04-12 @author: Mattias Måhl

class `frm_main.`**`frm_main`**
Bases: `tkinter.Tk`

Start object to render the application window.

use:

import libs.frm_main as mainwindow

if __name__ == "__main__":

app = mainwindow.App()

app.mainloop()

`createWidgets`(*frame*)
Creates and lays out the widgets for the mainwindow.

`quit`(**event*)

1.1.4 tracemac module

** TraceMac ** Traces a mac-address to a specific port in a switch. ** Version: 1.7.2 ** License: GPLv2

`tracemac.`**`pr`**(*pr_str*)
function pr function to print out messages to stdout using a specified format. This is used for verbose output to stdout and logfile.

`tracemac.`**`printout`**(*msg*, **w*)
function printout stardart output to stdout and logfile.

`tracemac.`**`read_infile`**(*filename*)
function read_infile this function serves to read the input file an store the targets for the search engine.

`tracemac.`**`split_string_into_chunks`**(*text*, *length=94*)
function split_string_into_chunks this function serves to sprit output into chunks that's under specified length. Default value is 94 chars.

tracemac.**vreport**(*header*, **msg*)
function vreport creates and outputs verbose output of progress.

tracemac.**write_to_file**(*line*)
function write_to_file if there is a file specified in options this will write to it.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

f

s

t