

get-snmp-mac.py

Specification & Documentation

gather mac table from a device using snmp

Kostis Netzwerkberatung
Konstantinos Kostis
kosta@kostis.net

Inhalt

Preface.....	3
Prerequisites.....	4
Specification	5
SNMP OIDs	5
OID iso.3.6.1.2.1.31.1.1.1.1 (ifName)	5
OID iso.3.6.1.2.1.17.7.1.2.2.1.2 (dot1qTpFdbEntry)	5
OID iso.3.6.1.2.1.47.1.2.1.1.2 (entLogicalDescr)	6
OID iso.3.6.1.4.1.9.9.46.1.3.1.1.2 (vtpVlanState).....	6
OID iso.3.6.1.2.1.17.4.3.1.2 (dot1dTpFdbPort)	6
OID iso.3.6.1.2.1.17.1.4.1.2 (dot1dBasePortIfIndex)	6
SNMP Version.....	7
Algorithm.....	7
Additional Data.....	7
Credentials.....	8
SNMP-credentials.txt.....	8
SNMP-devicelist.txt	8
Devices Tested On	9

Preface

get-snmp-mac.py is based on get-arp-snmp2.pl from the same author made in 2009. Before the author had made Perl scripts that gather such data from Cisco devices using expect as early as 2000.

get-arp-snmp2.pl was inspired by cammer.pl written by by Tobi Oetiker (author of mrtg et al).

For the moment get-arp-snmp.py uses SNMP V2c only but may support v3 in the future.

This script was developed and tested with Python 3.8.2 on a Ubuntu 20.4 system.

The SNMP library used is easysnmp. It has the reason for its use in its name.

Prerequisites

You need to have the following installed to run this script:

- Net-SNMP 5.7.x (or better)
- Python 3.8.2 (earlier version 3 may work but not tested)
- Python module easysnmp 0.2.5 (or better)

There must be an environment variable pointing to a directory containing user configuration files.

Example:

```
export DO_DEVICE = "$HOME/do-device/cfg"
```

Specification

The script shall query the ARP cache of a given device using SNMP and return:

IEEE-MAC;ifName;VlanId

Example:

52-54-00-1D-94-7A;Po2;101

The script shall have only one CLI parameter: hostname

Credentials shall be retrieved from a user configuration file containing all hostnames and their respective credentials (e.g. community).

SNMP OIDs

If possible only ISO SNMP OIDs shall be used.

The following SNMP OIDs are used to accomplish the goal:

Name	OID	community
ifName	iso.3.6.1.2.1.31.1.1.1.1	community
dot1qTpFdbEntry	iso.3.6.1.2.1.17.7.1.2.2.1.2	community
entLogicalDescr	iso.3.6.1.2.1.47.1.2.1.1.2	communiy
vtpVlanState	iso.3.6.1.4.1.9.9.46.1.3.1.1.2	community
dot1dTpFdbPort	iso.3.6.1.2.1.17.4.3.1.2	community@vlan
dot1dBasePortIfIndex	iso.3.6.1.2.1.17.1.4.1.2	community@vlan

OID iso.3.6.1.2.1.31.1.1.1.1 (ifName)

An SNMP walk of this OID returns:

oid => iso.3.6.1.2.1.31.1.1.1.1.7 (last octet is ifIndex)
value = ifName

OID iso.3.6.1.2.1.17.7.1.2.2.1.2 (dot1qTpFdbEntry)

An SNMP walk of this OID returns:

oid => iso.3.6.1.2.1.17.7.1.2.2.1.2.101.4.92.108.18.99.8 (last octets are vlan, mac)
value = ifIndex

OID iso.3.6.1.2.1.47.1.2.1.1.2 (entLogicalDescr)

An SNMP walk of this OID returns:

oid = iso.3.6.1.2.1.47.1.2.1.1.2.1 (last octet is index starting with 1)
value = vlan string (e.g. 'vlan101')

OID iso.3.6.1.4.1.9.9.46.1.3.1.1.2 (vtpVlanState)

An SNMP walk of this Cisco specific OID returns:

oid = iso.3.6.1.4.1.9.9.46.1.3.1.1.2.1.101 (last octets are '1' and vlan)
value = status (1 is operational)

OID iso.3.6.1.2.1.17.4.3.1.2 (dot1dTpFdbPort)

An SNMP walk of this OID using community@vlan returns:

oid = iso.3.6.1.2.1.17.4.3.1.2.24.232.41.77.248.140
value = port

OID iso.3.6.1.2.1.17.1.4.1.2 (dot1dBasePortIfIndex)

An SNMP get of this OID using community@vlan returns:

oid = iso.3.6.1.2.1.17.1.4.1.2.72 (last octet is port)
value = ifIndex

SNMP Version

At the moment SNMP v2c is used. Future versions may support v3. easysnmp does.

Algorithm

The gathering of MAC table data is done in steps:

1. Gather indexed device interface names (ifNames)
=> dict ifNames
2. Gather mac;vlan to ifName map (dot1qTpFdbEntry) – will not work with Cisco
3. If no data yielded, assume a Cisco device
4. Gather activevlans (vtpVlanState or entLogicalDescr)
5. Use extended community@vlan
6. Gather mac to port mappings (dot1dTpFdbPort)
7. Gather mac;vlan to ifName (dot1dTpFdbPort, dot1dTpFdbPort)

Additional Data

A MAC may be associated with a physical interface. Without additional information (like eth0.42) you cannot determine the Vlan for this interface. In order to properly associate such interface with Vlans there shall be an option to provide this information in a file "in2vlan.csv".

The format for the file should look like this:

```
hostname;interface;vlan
```

Example:

```
fwkostis0;ge-0/0/0.0;3  
fwkostis0;ge-0/0/1.0;100  
fwkostis0;ge-0/0/2.0;203  
fwkostis0;ge-0/0/3.0;243  
fwkostis0;ge-0/0/5.0;101  
router-isp;Gi0;3
```

Note that interface must match the format provided by ifName.

Credentials

In order to make administration of device and list containing device information manageable we split the information in two files

SNMP-credentials.txt

This file contains “global” credentials organized in “realms”.

```
*realm;community;port
```

Realm names start with ‘*’, the realm ‘*’ is the default realm.

Each realm holds the following information:

- community (SNMP v1/2c read-only community)
- port (*=161 as default port)

Example (using SNMP v2c):

```
*,example;*
```

SNMP-devicelist.txt

This file contains a list of all devices, optionally allows to map hostnames to ipaddr (when you don’t have DNS nor a hosts file). Any device can either use credentials from a realm or use explicit values, including the SNMP port.

```
hostname;ipaddr;community;port
```

If an ipaddr is given it is used rather than the hostname, if you want to use DNS/hosts for resolving the hostname, leave the ipaddr empty.

For SNMP v1/v2c you can (and should) leave username and password empty.

Example:

```
router-isp;;*;
```

router-isp is resolved using DNS/hosts and credentials are used from the default realm.

Devices Tested On

The script was designed for for devices support ISO OIDs.

It has been tested on:

Vendor	Model	OS	OS Version
Cisco	WS-C3560CG-8TC-S	ISO	15.2(2)E10
Ubiquiti	ES-16-XG		1.9.0