

get-snmp-arp.py

Specification & Documentation

gather arp cache 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.3.6.1.2.1.4.22.1.1 (ipNetToMediaIfIndex)	5
OID iso.3.6.1.2.1.4.22.1.2 (ipNetToMediaPhysAddress)	5
SNMP Version.....	5
Algorithm.....	6
Additional Data.....	6
Credentials.....	6
Devices Tested On	7

Preface

get-snmp-arp.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;IPv4Addr;VlanId

Example:

B4-FB-E4-1F-F0-BF;10.8.0.1;800

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
ifName	iso.3.6.1.2.1.31.1.1.1.1
ipNetToMediaIfIndex	iso.3.6.1.2.1.4.22.1.1
ipNetToMediaPhysAddress	iso.3.6.1.2.1.4.22.1.2

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.3.6.1.2.1.4.22.1.1 (ipNetToMediaIfIndex)

An SNMP walk of this OID returns:

oid = iso.3.6.1.2.1.4.22.1.1.17.192.168.178.1 (ifIndex, IPv4 address)
value = ifIndex

OID iso.3.6.1.2.1.4.22.1.2 (ipNetToMediaPhysAddress)

An SNMP walk of this OID returns:

oid = iso.3.6.1.2.1.4.22.1.2.17.192.168.178.1 (ifIndex, IPv4 address)
value = MAC address (binary format, 6 octets)

SNMP Version

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

Algorithm

The gathering of ARP cache data is done in steps:

1. Gather indexed device interface names (ifNames)
=> dict ifNames
2. Gather IPv4 address to interface association (ipNetToMediaIfIndex)
=> dict ipaddr2ifIndex
3. Gather MAC address to IPv4 address association (ipNetToMediaPhysAddress)
=> dict mac2ipaddr
4. Using ipaddr2ifIndex and ifNames as reference, list mac;ipaddr;vlan using mac2ipaddr

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

See SNMP_pwdecrypt.

Devices Tested On

The script was designed for for devices support ISO OIDs.

It has been tested on:

Vendor	Model	OS	OS Version
Cisco	C926-4P	IOS	15.8(3)M2
Cisco	WS-C3560CG-8TC-S	ISO	15.2(2)E10
Devolo	WiFi pro 1200e	firmware	4.0.1
Juniper	SRX300	JUNOS	20.2R1.10
Ubiquiti	ES-16-XG		1.9.1