SNMP_pwdecrypt.py

Specification & Documentation

store, organize, and provide SNMP credentials

Kostis Netzwerkberatung
Konstantinos Kostis
kosta@kostis.net

Inhalt

Preface

SNMP-credentials stores, organizes, and provides SNMP credentials to Python3 scripts.

Sensitive data like community is encrypted.

For the moment only SNMP v2c is support but v3 may be supported in the future.

The scripts were developed and tested with Python 3.8.2 on a Ubuntu 20.4 system.

Prerequisites

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

- Python 3.8.2 (earlier version 3 may work but not tested)
- pip3 module cryptography version 2.8 (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

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 (encrypted 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 neither use 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.

Example:

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

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

Usage

This module has two public functions:

- decrypt_password(crypted)
- get_credentials(DEBUG, device)

Function decrypt_password(crypted)

This function is imported using:

```
from SNMP_pwdecrypt import decrypt_password
```

The function has a single parameter containing encrypted data (e. g. credentials).

Example:

```
password = decrypt_password(encrypted)
```

Function get_credentials(DEBUG, device)

This function is imported using:

```
from SNMP_pwdecrypt import get_credentials
```

The function has two parameters, one containing a device name.

Example:

```
credentials = get_credentials(DEBUG, hostname)
```

If no credentials for a given device are found, the function returns {}.

DEBUG contains an error level. If it's 0, no debugging information is displayed.

Debugging information, if any, will go to sys.stderr.