資産情報自動収集ツールの利用方法について

本ドキュメントは資産情報の収集および資産台帳と現状の資産の突合を自動化する資産情報自動収集ツール（自動化ツール）の利用方法についてまとめたものである。

目次

1　自動化ツールの概要

ネットワーク経由で資産情報を収集し、台帳作成/更新および不正な端末、通信の検出が可能なツールである。なおアクティブスキャンは制御システムの可用性に影響を及ぼす可能性があるため、使用開始時に警告表示される。

- 検証済み OS 環境
  - ・macOS 10.14
  - ・CentOS7
  - ・raspbery pi Debian version 10.3
- 開発プログラミング言語
  - ・Python 3.7.7
- 機能
  - ・2 種類のアクティブスキャン（非認証型）による資産情報の収集
  - ・パッシブスキャンによる資産情報の収集
  - ・スキャン実行時に台帳ファイル(CSV ファイル)を指定することで前回スキャン結果との比較が可能（IP アドレス、MAC アドレスの変化を検知）
  - ・資産情報の取得および CSV 形式の台帳作成/更新

（それぞれの対応機能は表 1 のとおりである。）

表 1　自動化ツールで取得可能な資産情報一覧

資産情報項目	簡易アクティブスキャン	詳細アクティブスキャン	パッシブスキャン ※1
資産名(ホスト名)		●※2	●※2
IP アドレス	●	●	●
MAC アドレス	●	●	●
ベンダー情報	●	●	●
OS 種類、バージョン		●※3	●※3
通信先			●
通信プロトコル（ポート含む）			●
制御通信プロトコル 固有詳細情報		●※4	

※1:正確に資産情報を把握するためには、スキャンする前にミラーポートの準備が必要である。取得するパケットが大量通信の場合、パケットを拾えない場合がある。

※2:ホスト名はアクティブスキャンの場合は NetBIOS 名を取得しており、対象端末が未対応の場合は取得できない。パッシブスキャンの場合は DHCP、NBNS プロトコルから判定しているため通信が流れていない場合は判定できない。

※3:OS 種類、バージョンは TTL(Time To Live)から Windows 系、Linux or MAC 系、Unix or Network 機器系と簡易的に判定を実施している（判定できない場合は unknown）。

※4: BACnet のみ実装

自動化ツールの実行フローは以下の図のようになっている。

- 簡易アクティブスキャンでは arp を利用して資産情報を取得する
- 詳細アクティブスキャンでは arp に加えて、NetBIOS、PING を利用して資産情報を取得する
- 制御通信プロトコルスキャンでは BACnet プロトコルを用いて資産情報を取得する
- パッシブスキャンでは tshark を用いて pcap ファイルを作成し、作成したファイルの解析により資産情報を取得する
- PCAP スキャンは pcap ファイルの解析により資産情報を取得する

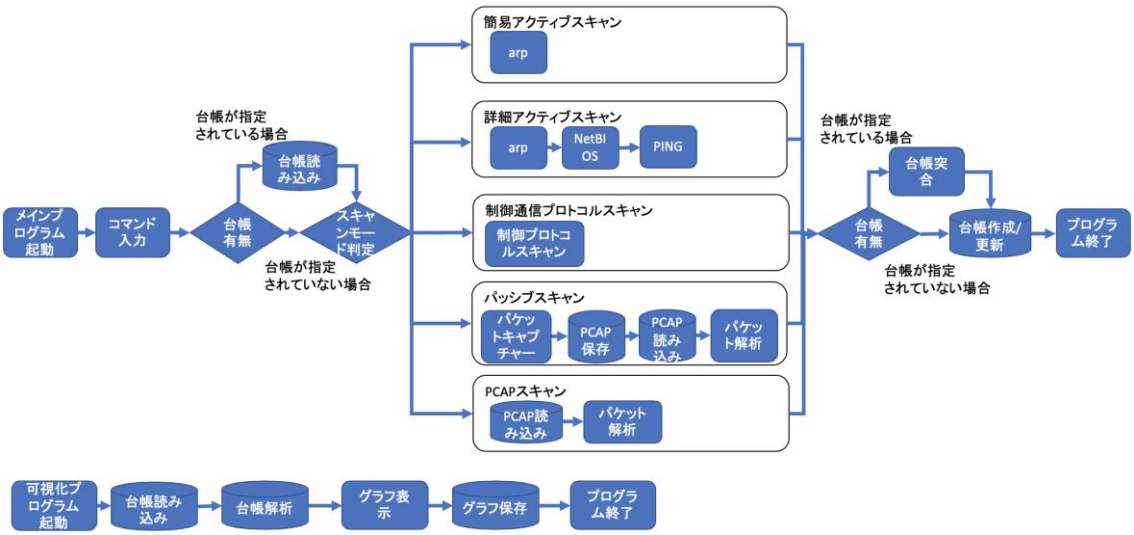

図 1　自動化ツールのフロー図

2 準備

自動化ツールの実行にあたり事前準備について記載する。

2.1 自動化ツールをダウンロードする

自動化ツールをダウンロードする。ダウンロードフォルダは書き込み権限がある場所とする。

（pcap ファイルや、xml ファイル、png ファイルを作成するため）

2.2 必要ソフトウェアのインストールする

「８章　動作環境」に記載した前提アプリケーション・ライブラリをインストールする。

2.3 ネットワーク設定

自動化ツールを実行する IP アドレスのセグメントをスキャン対象のセグメントに変更する

例）スキャン対象が 192.168.1.0/24 なら、実行 PC を 192.168.1.X にする。

3 実行画面

メインファイルを python で実行後、コマンド入力して実行する（メインファイル起動時に引数入力で実行開始も可能）。例は簡易アクティブスキャン実行時とヘルプ実行時である。

図 2 実行画面（簡易アクティブスキャン ）

図 3 実行画面（ヘルプ）

4 実行結果

以下はパッシブスキャンを実施したときの出力結果である※値はサンプル

hostname	ip_src	mac_src	vendercode	osname	ip_dst	protocol	port_src	port_dst	communication	status
hostA	192.168.1.10	aa:bb:cc:dd:ee:01	samplevender	windows	192.168.1.25	UDP(4117),UDP(4114),BACnet-AF	411,447,808	4,117,411,447,808	192.168.1.10:4114-192.168.1.255:UDP(4117),	OK
hostB	192.168.1.11	aa:bb:cc:dd:ee:02	samplevender	windows	255.255.255.	UDP(4117),UDP(4114)	4114	41,174,114	192.168.1.11:4114-255.255.255.255:UDP(411	OK
unknown	unknown	aa:bb:cc:dd:ee:03	samplevender	unix or network	unknown	STP	unknown	unknown	unknown	OK
hostC	192.168.1.30	aa:bb:cc:dd:ee:04	samplevender	windows	192.168.1.20	BACnet-APDU(47808),ARP	47808	47808	192.168.1.30:47808-192.168.1.209:BACnet-AF	OK
hostD	192.168.1.209	aa:bb:cc:dd:ee:05	samplevender	windows	192.168.1.30	BACnet-APDU(47808),ARP	47808	47808	192.168.1.209:47808-192.168.1.30:BACnet-AF	OK
hostE	192.168.1.203	aa:bb:cc:dd:ee:06	samplevender	windows	192.168.1.40	BACnet-APDU(47808),ARP,ICMP,I	478,081,750,054,129	47,808,175,001,900	192.168.1.203:47808-192.168.1.40:BACnet-AF	OK
hostF	192.168.1.40	aa:bb:cc:dd:ee:07	samplevender	windows	192.168.1.20	ARP,BACnet-APDU(47808)	47808	47808	192.168.1.40:47808-192.168.1.203:BACnet-AF	OK
hostG	192.168.1.70	aa:bb:cc:dd:ee:08	samplevender	windows	192.168.1.25	BACnet-APDU(47808)	47808	47808	192.168.1.70:47808-192.168.1.255:BACnet-AF	OK
hostF	192.168.1.12	aa:bb:cc:dd:ee:09	samplevender	windows	192.168.1.25	BACnet-APDU(47808)	47808	47808	192.168.1.12:47808-192.168.1.255:BACnet-AF	OK
hostH	192.168.1.21	aa:bb:cc:dd:ee:10	samplevender	windows	192.168.1.25	BACnet-APDU(47808)	47808	47808	192.168.1.21:47808-192.168.1.255:BACnet-AF	OK

図 4　スキャン機能の CSV 出力結果

またスキャン結果のネットワークトポロジー図を出力することが可能である。過去の資産台帳と比較することで、図のように新規資産台帳にしかないデータは赤色、逆に古い資産台帳にしかないデータは緑色で表示することが可能で、目視でも変化に気づくことが容易。また MAC アドレスのみの表示、IP アドレスのみの表示も可能である。

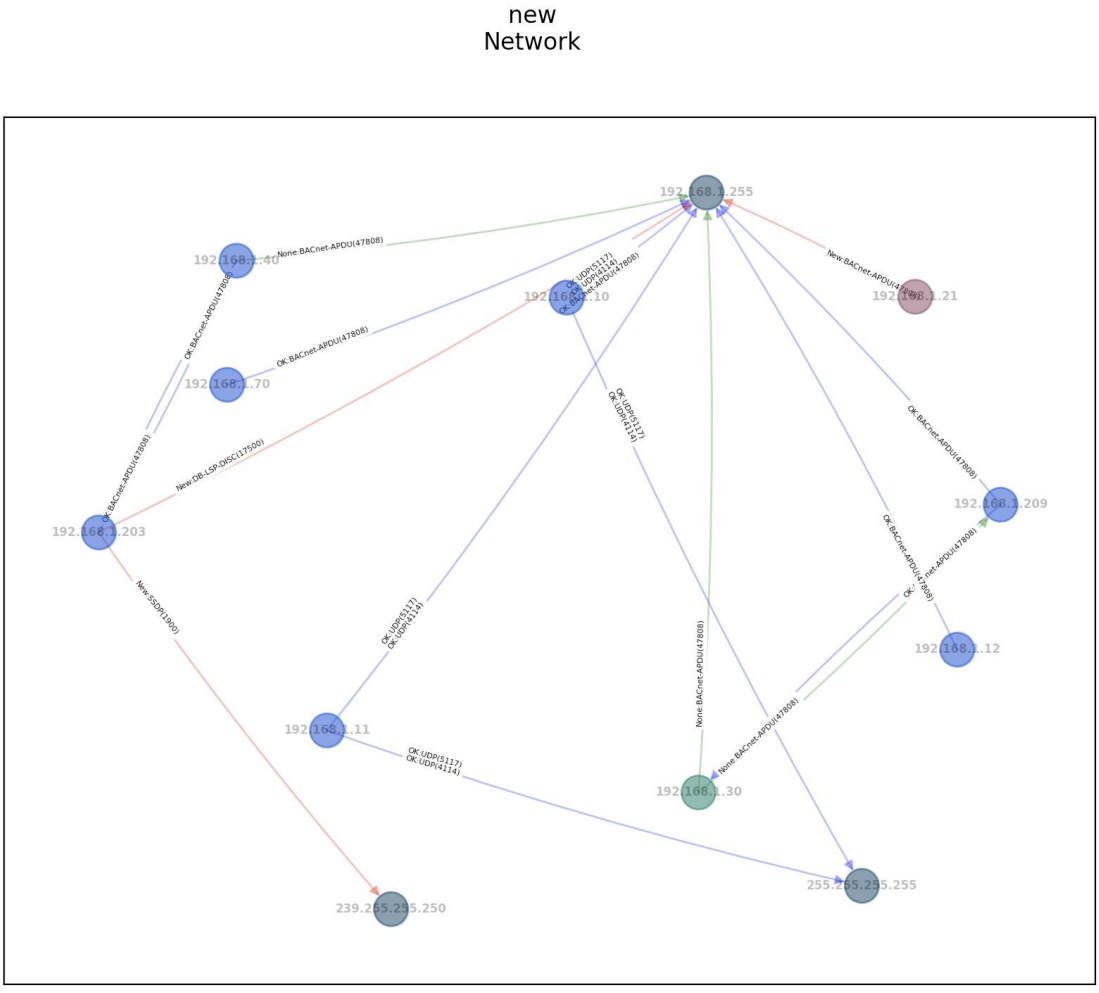

図 5　ネットワーク可視化機能の出力結果

さらに可視化機能は XML 出力ができ、外部ツール（図 6 は Cytoscape を使用）を用いることでノードの移動、拡大・縮小やフィルタリング操作が可能のため可視性を上げることができる。例では新しい通信を赤く表示している。

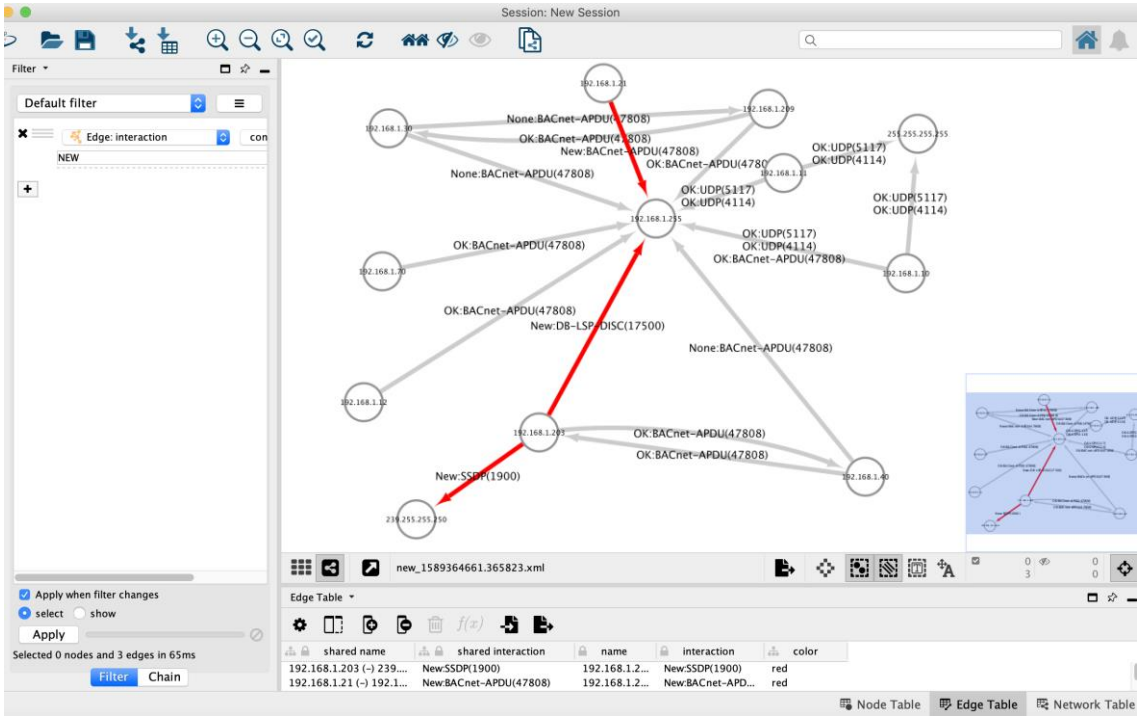

図 6 Cytoscape での出力結果

5 実行コマンド

5.1 スキャン機能

1. coe_assetmanagement_main.py 実行

   >sudo python3 coe_assetmanagement_main.py

2. コンソールモードが起動するので以降は「表　スキャン機能コマンド一覧」参照

   以下のようにコンソールモードを省略することも可能。

   >sudo python3 coe_assetmanagement_main.py ass en7 192.168.0.1/24

表 2　スキャン機能コマンド一覧

項番	スキャンモード	コマンド※1,2
1	簡易アクティブスキャン （台帳無し）	ass [NIC] [IP アドレス or セグメント] ass= active scan simple >ass en7 192.168.0.0/24
2	簡易アクティブスキャン （台帳有り）	assl [NIC] [IP アドレス or セグメント]　[台帳パス] >assl en7 192.168.0.0/24 inlist.csv
3	詳細アクティブスキャン （台帳無し）	asd [NIC] [IP アドレス or セグメント] >asd en7 192.168.0.0/24
4	詳細アクティブスキャン （台帳有り）	asdl [NIC] [IP アドレス or セグメント]　[台帳パス] >asdl en7 192.168.0.0/24 inlist.csv
5	パッシブスキャン （台帳無し）※3	ps [NIC] [計測時間(秒)] >ps en7 360
6	パッシブスキャン （台帳有り）※3	psl [NIC] [計測時間(秒)] [台帳パス] >psl en7 360 inlist.csv
7	PCAP スキャン （台帳無し）	pcaps [pcap ファイルパス] >pcaps input.pcap
8	PCAP スキャン （台帳有り）	pcaps [pcap ファイルパス] [台帳パス] >pcaps input.pcap inlist.csv
9	BACnet スキャン	bacnet [NIC] [IP アドレス or セグメント] >bacnet en7 192.168.0.0/24
10	ヘルプ	＞help

※1 セグメントは/24 を推奨する

※2 [***]内に自身の環境に合わせてパラメータ入力する

※3 リアルタイム処理ではなく、計測時間の pcap ファイルを一度作成し、それを pcap スキャンモードで読み込んでいる（一つのツールで完結することがメリット）

表 3　スキャン機能コマンド　共通オプション

項番	オプション名	コマンド※1
1	CSV 出力機能	それぞれのスキャンコマンド後に「-o 出力先」を指定する >ass en7 192.168.0.10/24 -o outlist.csv
2	アクティブスキャン 時の警告非表示※1	末尾「-q」をつける >ass en7 192.168.0.10/24 -o outlist.csv -q

※1 通常はアクティブスキャン 、BACnet スキャンを実施する場合、以下は警告が表示される

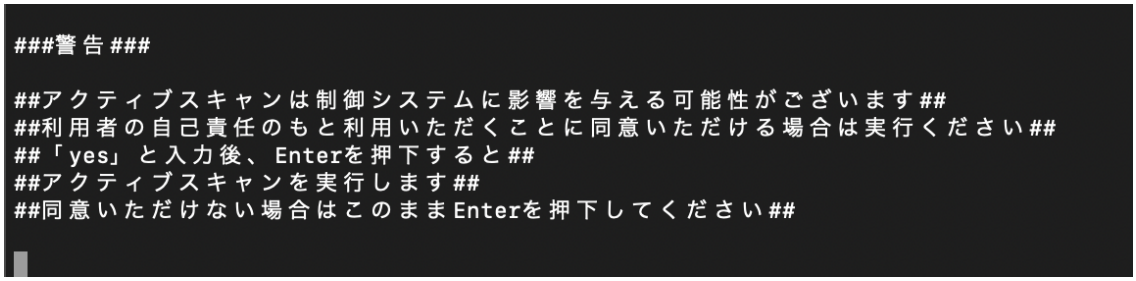

図 7　警告表示

5.2　ネットワーク可視化機能

スキャンモードで出力した台帳ファイルを可視化させる。

一つの台帳の結果を描画する「通常モード」と新旧の台帳を比較した結果を描画する「比較モード」がある。比較モードは新しく出力した資産台帳にしかないデータは赤色、逆に古い資産台帳にしかないデータは緑色で表示する。

なお、実行フォルダに XML ファイルと PNG ファイルを出力する。

表 4　ネットワーク可視化コマンド一覧

項番	モード	
1	通常	ノードが MAC アドレスのデータを表示する python3 可視化ファイル名 MAC [台帳パス] ＞python3 coe_assetmanagement_networkviewer.py MAC hoge.csv
2		ノードが IP アドレスのデータを表示する python3 可視化ファイル名 IP [台帳パス] ＞python3 coe_assetmanagement_networkviewer.py IP hoge.csv
3		ノードが IP アドレスで通信情報を付与したデータを表示する python3 可視化ファイル名 COMM [台帳パス] ＞python3 coe_assetmanagement_networkviewer.py COMM hoge.csv
4	比較	ノードが MAC アドレスで比較したデータを表示する python3 可視化ファイル名 MAC2 [旧台帳パス] [新台帳パス] ＞python3 coe_assetmanagement_networkviewer.py MAC2 old.csv new.csv
5		ノードが IP アドレスで比較したデータを表示する python3 可視化ファイル名 IP2 [旧台帳パス] [新台帳パス] ＞python3 coe_assetmanagement_networkviewer.py IP2 old.csv new.csv
6		ノードが IP アドレスで通信情報を付与した状態を比較したデータを表示する python3 可視化ファイル名 COMM2 [旧台帳パス] [新台帳パス] ＞python3 coe_assetmanagement_networkviewer.py COMM2 old.csv new.csv

6 資産台帳形式について

資産台帳形式は以下の通りになる（入出力される CSV 形式）

[凡例]

判定：台帳ありコマンド実行時に、比較を行い、異なった場合にメッセージを表示する

未判定：台帳ありコマンド実行時に、比較を行い、異なった場合にメッセージを表示しない

表 5　資産台帳形式

No	資産情報項目	簡易アクティブ	詳細アクティブ	パッシブ	備考
1	資産名(ホスト名)	未判定	判定	判定	
2	IP アドレス	判定	判定	判定	
3	MAC アドレス	判定	判定	判定	
4	ベンダー情報	未判定	未判定	未判定	出力と列を合わせるためであり、空白でも OK
5	OS 種類、バージョン	未判定	判定	判定	
6	通信先(MAC アドレス)	未判定	未判定	判定	複数の場合、””で囲ってカッコ内をカンマ区切り
7	通信先(IP アドレス)	未判定	未判定	判定	複数の場合、””で囲ってカッコ内をカンマ区切り
8	通信プロトコル（ポート含む）	未判定	未判定	判定	上記同様
9	送信元ポート	未判定	未判定	未判定	上記同様
10	送信先ポート	未判定	未判定	判定	上記同様
11	通信情報（送信元：送信元ポート：送信先：通信プロトコル（送信先ポート）	未判定	未判定	判定	上記同様
12	ステータス	未判定	未判定	未判定	１：MAC/IP が完全一致の場合：OK ２：MAC/IP 両方とも一致しなかった場合：NoDiscovery ３：IP のみ一致した場合：IPOnlyMatch ４：MAC のみ一致した場合:MACOnlyMatch ５：新規端末検出時：DiscoveryNewDevice

7 BACnet スキャンの取得情報について

BACnet/IP プロトコルを使用しているビルシステムに対して、BACnet プロトコルの who-Is と readProperty の 2 種類のサービスを活用し BACnet 端末の資産情報の収集を行う。

1. 自動化ツールから who-Is をブロードキャストで送信し BACnet 端末から i-Am 応答を受信することにより、ネットワーク情報（IP アドレス、MAC アドレス）と BACnet Device オブジェクト ID、BACnet ベンダ名を取得する。ただし BACnet ベンダ名は BACnet ベンダ ID を取得し、対応表に従って BACnet ベンダ名に変換する。
2. 自動化ツールが送信した who-Is に対して BACnet デバイスが i-Am を応答することより得られた BACnet 端末の MAC アドレス、IP アドレスの情報を活用して、ツールから各デバイスに対して表 6 に記載の Device オブジェクトのプロパティ ID を設定した readProperty を順番に送信する。BACnet 端末は対応しているプロパティ ID については readProperty に対する Complex-ACK 応答に該当 ID のプロパティ値を含めて応答を返すため、この値を自動化ツールで取得する。

   Device オブジェクトに対応したプロパティ ID は BACnet の規格書によると 46 種類あるが、資産管理に必要な資産情報収集の目的を考慮して、下記の 8 種類のオブジェクトのプロパティ値を取得するものとする。

表 6　BACnet の Device オブジェクトプロパティ取得情報一覧

プロパティ ID	BACnet プロパティ	BACnet データ型
12	Application Software Version	character string
44	Firmware Revision	character string
70	Model Name	character string
77	Object Name	character string
98	Protocol Revision	unsigned
112	System Status	enumerated
139	Protocol Version	unsigned
155	Database Revision	unsigned

8 動作環境

8.1 管理者権限

管理者権限で実行する

8.2 動作環境済み OS

以下の OS については動作検証済み

- macOS 10.14
- CentOS7
- raspbery pi Debian version 10.3

8.3 前提アプリケーション、ライブラリ

以下のアプリケーション、ライブラリをインストールしておく必要がある。

なお本アプリケーション、ライブラリはそれぞれのマニュアルを確認し、自己責任にてインストールを行う。

- python バージョン 3.x 系
- tshark ※1
- Python ライブラリ
  - ・scapy
  - ・pyshark
  - ・netaddr
  - ・netifaces
  - ・networkx ※2
  - ・matplotlib ※2

※1 PATH を通した状態であること（./tshark の「./」が不要な状態）

※2 ネットワーク可視化機能を使う場合

8.4　フォルダ構成

メインファイルを含む以下全てのファイルが同一フォルダに格納している必要がある。またメインファイルが格納されたフォルダにはファイル書き込み権限を有している必要がある。

表 7　ファイル一覧

No.	ファイル名	備考
1	coe_assetmanagement_activescan.py	アクティブスキャン機能
2	coe_assetmanagement_bacnet.py	BACnet スキャン機能
3	coe_assetmanagement_class.py	資産台帳クラス
4	coe_assetmanagement_common.py	共通関数
5	coe_assetmanagement_main.py	メイン関数
6	coe_assetmanagement_match.py	台帳との比較機能
7	coe_assetmanagement_networkviewer.py	ネットワーク可視化機能
8	coe_assetmanagement_passivescan.py	パッシブスキャン機能
9	vender_mac_v3.txt	MAC アドレスとベンダーコード紐付け
10	service-names-port-numbers.csv	ポート番号とプロトコル紐付け
11	ieee-802-numbers-1.csv	EtherType とプロトコル紐付け
12	protocol-numbers-1.csv	IP ヘッダーのプロトコル番号とプロトコル紐付け
13	bacnet_vendor_id_list.csv	BACnet Vendor ID とベンダ名の紐付け

9 参考情報

Cytoscape

https://cytoscape.org/

10 免責事項

本ツールは商用レベルではありません。11.著作権に記載の各著作権等保有者は、本ツールの利用に起因または関連して利用者に生じたトラブルや損失、損害等に対して、一切の責任を負いません。作成者は本ツールに不具合がある場合出来るだけ早期に改善を試みる予定ですが、時間を要するか、改善できない場合もあります。作成者は、不具合の修正の義務を負いません。

11 著作権

【版管理】

2020 年 6 月 29 日	初版