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 2](#_Toc44239999)

[2 準備 5](#_Toc44240000)

[3 実行画面 6](#_Toc44240001)

[4 実行結果 7](#_Toc44240002)

[5 実行コマンド 9](#_Toc44240003)

[5.1 スキャン機能 9](#_Toc44240004)

[5.2 ネットワーク可視化機能 11](#_Toc44240005)

[6 資産台帳形式について 12](#_Toc44240006)

[7 BACnetスキャンの取得情報について 13](#_Toc44240007)

[8 動作環境 14](#_Toc44240008)

[8.1 管理者権限 14](#_Toc44240009)

[8.2 動作環境済みOS 14](#_Toc44240010)

[8.3 前提アプリケーション、ライブラリ 14](#_Toc44240011)

[8.4 フォルダ構成 15](#_Toc44240012)

[9 参考情報 16](#_Toc44240013)

[10 免責事項 16](#_Toc44240014)

[11 著作権 16](#_Toc44240015)

# 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

スクリーンショットの画面

自動的に生成された説明

図 1　自動化ツールのフロー図

# 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。

# 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。

スクリーンショット, 抽象, 座る, モニター が含まれている画像

自動的に生成された説明

図 2 実行画面（簡易アクティブスキャン ）

文字と写真のスクリーンショット

自動的に生成された説明

図 3 実行画面（ヘルプ）

# 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

スクリーンショットの画面

自動的に生成された説明

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。　　　地図, 写真, ボート, テーブル が含まれている画像

自動的に生成された説明

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。

地図のスクリーンショット

自動的に生成された説明

図 6 Cytoscapeでの出力結果

# 実行コマンド

## スキャン機能

1. coe\_assetmanagement\_main.py実行

>sudo python3 coe\_assetmanagement\_main.py

1. 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

コンピュータ, ブラック, キーボード が含まれている画像

自動的に生成された説明

図 7　警告表示

## 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 |

# 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 |

# 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 |

# 動作環境

## 管理者権限

管理者権限で実行する

## 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

## 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

## 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 |

# 参考情報

Cytoscape

<https://cytoscape.org/>

# 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。

# 著作権

本ツールに関する著作権その他すべての知的所有／財産権は、「情報処理推進機構 産業サイバーセキュリティセンター 中核人材育成プログラム3期生 資産管理プロジェクト」に帰属します。

利用者は10.免責事項のすべての内容に同意された場合に限り、本ツールを、自己利用または、自身の所属組織での内部利用のため必要な範囲で複製し、或いはそのサーバー上に搭載して閲覧等に供することができます。これらの範囲を超える利用は、各権利者の明示の同意がない限り、禁止されています。

**【版管理】**

|  |  |
| --- | --- |
| 2020年6月29日 | 初版 |
|  |  |
|  |  |