L-023

プロセス構造の解析による IoT ボットの検知
A proposal of the detecting method of IoT bot based on analysis of process structure

小西 達也† 小林 孝史‡
Tatsuya Konishi Takashi Kobayashi

1. はじめに

近年，IoT 機器が注目を浴び，多くの場面で使われている．その反面，IoT 機器に対するセキュリティ対策が不十分であるために多くの攻撃の踏み台に利用されている．2016 年 9 月に米国で IoT 機器に感染する Mirai と呼ばれる新種のボットが出現し，現在も世界中で多くの被害をもたらしている．その後，Reaper，Satori 等の多くの Mirai の亜種が出現したことで IoT 機器を狙う攻撃が増加している．2017 年 2 月の警察庁の調査 [1]ではインターネット定点観測システムに対する 52%のアクセス元が IP カメラ等の IoT 機器であったことを報告している．そのため早急な IoT 機器へのセキュリティ強化が必要とされている．

本研究では，IoT ボットに感染していない正常時に IoT 機器内で発生するプロセスを解析・学習することでプロセスの観点から IoT ボットの動作を検知し，ユーザに報告するシステムを実現することを目的とする．

2. ボット

ボットとは，感染したコンピュータを C&C（Command and Control）サーバを中継して外部から遠隔操作できる状態にするマルウェアの一種である．その中でも IoT ボットは，IoT 機器に感染するボットで，最近の IoT ボットの多くが 2016 年 6 月に GitHub 上で公開された Mirai のソースコードを元に作成されている．

IoT ボットの基本的な動きとしては，まず最初に脆弱な IoT 機器を発見するためのスキャン活動を実行する．その後，ブルートフォースアタック又は IoT 機器内で動作しているサービスの脆弱性を突くことで機器内部に侵入し，外部レポートサーバに脆弱な IoT 機器の情報を送信する．レポートサーバはその機器にダウンローダ又はシェルコードをダウンロードさせる．このダウンローダ等を通して IoT 機器のアーキテクチャに適合する IoT ボットの実行バイナリを外部から取得し，起動する．起動された IoT ボットは，C&C サーバへの接続を行いスキャン活動を開始する．上記の一連の動作を繰り返すことで感染を拡大する．

IoT ボットには，従来のボット同様に C&C サーバに接続し，攻撃者の命令を受け，外部サーバに攻撃を行う．しかし，IoT ボットは，従来のボットと異なる点が三つある．まず最初にボット自身に脆弱な IoT 機器を発見するスキャン機能を持つ点である[2]．この機能により IoT ボットは，短時間に多くの脆弱な IoT 機器を自身のボットネット（ボットで構築したネットワーク）に組み込むことができる．次に，複数機器への感染を行うことができる点である．IoT ボットは，arm，mips，mipsel 等の複数アーキテクチャの実行バイナリを所持しており調査を通して感染対象の機器にあったものをダウンロードさせて感染を広げる．これにより多くの種類の機器に感染することが可能である．最後に，ボット自体が複数の役割を持っている点である．Mirai 以前のボット（Bashlite 等）では，脆弱な機器を発見するためのスキャン活動等の機能は他のサーバが行っていた．しかし，IoT ボットではスキャン活動や脆弱な IoT 機器の情報をレポートサーバに伝える動作も IoT ボット自身で行われている．さらに，これらの C&C サーバへの接続機能，攻撃機能，スキャン機能，送信機能等は fork を使用することで並列に実行している．これにより IoT ボットは巨大なボットネットを効率よく形成することを可能にしている．

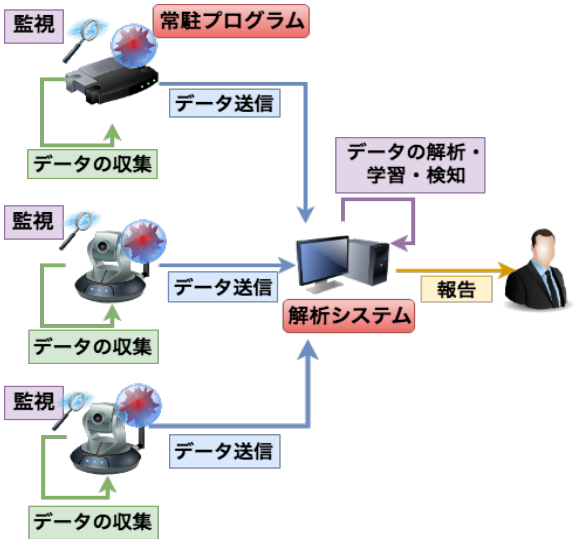

図 1: 提案システムの構成

3. 本研究のシステム

2 節で説明したように，IoT ボットは多くの機能を並列に実行するために fork を用いて複数のプロセスを生成する．そのため，一つのプロセスを起点にした親と子を持つプロセスが複数出現すると考えられる．本研究では，この親と子の関係を持つプロセスを追跡し集めたプロセス群をプロセス構造と定義する．また，これらの生成されたプロセスは wget コマンドによるダウンローダの取得やダウンローダ自身による感染対象の調査，レポートサーバや C&C サーバとのやり取りを行うために多くの通信を発生させる．そのため，IoT ボットが動作した機器のプロセス構造は，正常時のプロセス構造と比較し，多くの通信を持つものになると考えられる．そこで本研究では，正常時に発生する通信とプロセス構造を学習することで，IoT ボットの動作により発生する異常な通信やプロセスの検知を行う．

本システムの構成を図 1 に示す．本システムは，機器内のデータを収集する常駐プログラムと送信されたデータから通信の情報とプロセス構造を抽出し，学習・検知する解析システムで構成されている．本システムの解析システム

† 関西大学大学院 Kansai University Graduate School
‡ 関西大学 Kansai University

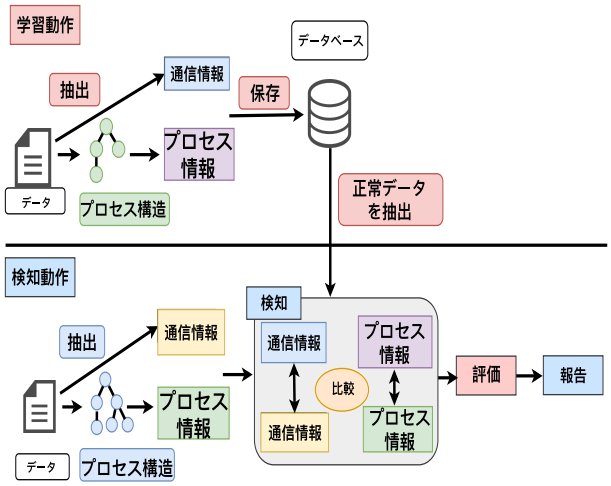

図 2:　解析システムの流れ

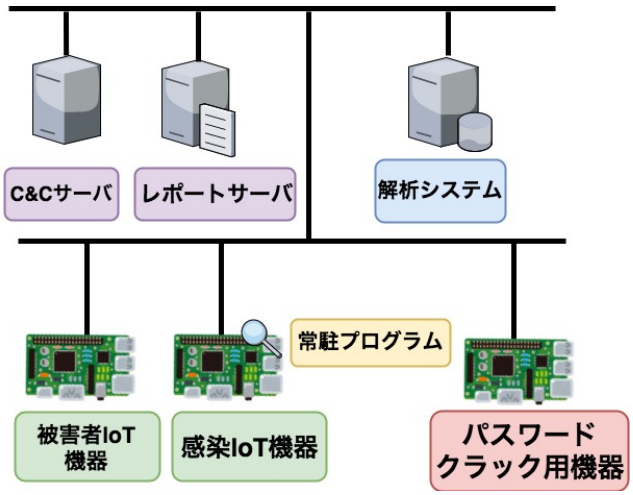

図 3:　検証環境

では，IoT 機器内のデータを学習する学習動作と検知動作の二つから IoT ボットの動作を検知する．解析システムの流れを図 2 に示す．

3.1 学習動作

学習動作では，収集したデータから二つの情報を学習する．一つ目は，IoT 機器の通信の情報である．具体的には，宛先 IP アドレス・宛先ポート番号・ 通信時間間隔・通信回数を特徴量として学習する．これらの値は， 1 日ごとのデータを用いて学習を行う．二つ目は，IoT 機器内で発生するプロセスの構造の情報である．具体的には，プロセスから抽出したプロセスの構造が持つ宛先 IP アドレス・ポート番号のセット，通信数，通信しているプロセスの数，分岐後のプロセス数，分岐後で通信を行っているプロセスの数を特徴量として取得して学習を行う．なお，この特徴量は決定木を用いて悪意のあるプロセス構造と通常のプロセス構造に対して良悪判断を行い，判断に利用された変数の上位五つを使用している．

3.2 検知動作

検知動作では，通信による検知とプロセス構造による検知の 2 段階の検知を行う．最初に通信による検知を実行する．ここでは，送信されたデータから前述した通信の情報を取得し，学習した値と比較する．データが学習の値と異なる場合 は異常な通信が発生したと判断し，プロセス構造による検知を実行する．プロセス構造による検知では，学習動作と同様の方法でプロセス構造から前述した特徴量を取得する．取得した値が学習した値の最頻値・ 基準値・最大値（最頻値と基準値の差の半分を基準値に足した値）を超える場合は順に 0.5，1，2 と評価し，それ以外は 0 と評価する．五つの変数の評価の値を加算したものを評点とする．その後，プロセス名の偽装を行っているプロセスの有無を調査し，存在する場合は評点に 2 を加算する．そして，合計した評点を特徴量の数の 2 倍で割ったものを悪性度とする．この値が 0.6 以上であった場合に 発生したプロセスを異常と判断しユーザに報告する．

4.　検証と考察

本研究では，Raspberry Pi 3 を用いて研究室の気温を一定間隔ごとに送信する IoT 機器を作成した．この機器で収集したデータを正常データとして検証に使用する．本検証では，研究室内のネットワークに図 3 のような検証環境を構築した．本研究の検証では，感染 IoT 機器に対し外部からの侵入・調査・ボットの感染・被害者 IoT 機器への攻撃といった一連の動作を行い，提案システムが検知対象を検知できるかを検証する．検知対象として Mirai・ Telnet へのパスワードクラック攻撃・Reaper（Reaper の感染シェルスクリプト）を使用した． 検証結果を表 1 に示す．

検証の結果，全ての検知対象で 0.6 以上の悪性度となり，全ての検知対象を検知することができた．また，Mirai が行うプロセス名の偽装動作を偽装であると判断することができた．

表 1:　検証結果

検知対象	悪性度
Mirai	0.92
Telnet による パスワードクラック攻撃	0.71
Reaper	0.82

5.　おわりに

本研究では，IoT 機器の内部で発生したプロセスの情報を収集・解析することでプロセスの観点から IoT ボットの動作及びプロセスの偽装を検知することができた．現段階では，検知にプロセスと通信数，通信の時間間隔，宛先 IP アドレス，宛先ポート番号の情報のみを使用しているが，今後は通信の状態や通信量の大きさ等の情報を追加することで検知率を上げていきたい．

参考文献

[1] 警察庁，“「Mirai」ボットの亜種等からの感染活動と見られるアクセスの急増について”，
https://www.npa.go.jp/cyberpolice/topics/?seq=19824,
2018 年 6 月 27 日確認.

[2] Manos Antonakakis,Tim April,Micahel Bailey,etc.al “Understanding the Mirai Botnet”， USENIX Association, 26th USENIX Security Syposium(2017 Agust 16-18).