IoT デバイス向けホスト型 IDS の開発

20T315　田中瑠星（最所研究室）

１．背景

近年,IP カメラやルータといった IoT デバイス数は増加傾向にあり,多種多様な場面で多くの人の生活を豊かにしている.その一方で，IoT デバイスを狙った攻撃が数多く確認されている．今現在の主流の対策方法は，外部ネットワークと内部ネットワークを隔てるファイアウォールや IDS などを設置することにより，外部からの攻撃を未然に防ぐネットワーク境界型でのセキュリティ対策である．しかし，ネットワーク境界型だけのセキュリティ対策では，ネットワーク貫通型攻撃やラテラルムーブメントに対して不十分であるため，IoT デバイス向けのホスト型のセキュリティ対策が必要になってくる．しかし，ホスト型 IDS は，CPU やメモリなどの制約や，マルウェアを検知する方法の制約があり，導入が難しい.

2. 関連研究とその問題点

関連研究[1]では，IoT デバイスを攻撃目標とするマルウェアは， SYN スキャンや TCP スキャンで感染先のホストを見つけ，SSH や Telnet へのブルートフォース攻撃を行うことで，IoT デバイスに侵入を試みる特徴を用いて検知を行う．しかし近年は SSH や Telnet へのブルートフォース攻撃を行わず，ソフトウェアの脆弱性を用いて IoT デバイスに侵入するマルウェアが発見されており，従来の検知方法では検知漏れが起きる.

関連研究[2]では，LSM を用いてホワイトリスト上にあるアプリケーション以外の実行をブロックするシステムである．このシステムは，ホワイトリストを一台毎に作らないといけないため大きな負担となる．さらに，管理者への通達機能がないため，同じネットワークにあるほかの機器へ攻撃の影響が出てしまうとその脅威に気が付けず対応できないという課題がある.

3. ホスト型 IDS を起用するにあたっての課題

課題①汎用機で使うセキュリティ対策ツールは使えない

IoT デバイスは，汎用機のように大きな処理能力を持ち合わせておらず，“ClamAV”のような汎用機向けのセキュリティソフトは IoT デバイスで使うことは，厳しいといえる[3].

課題②想定外攻撃パターンによる検知漏れ

IoT デバイスを狙うマルウェアは，デバイスに侵入する方法やデバイスに侵入後の隠蔽方法に手を加えた Mirai マルウェアの亜種が多い．そのため，これら 2 つでマルウェアを検知すると検知漏れが起こる可能性がある.

課題③画面出力がないので気づきにくい

IoT デバイスは，汎用機のように画面出力が無く，異変に気付くことが困難である．そのため，攻撃の影響が甚大化しやすいという性質がある[4].

4. 要件

Edgerunner の要件は，以下の 4 つである.

要件①小リソースでも動かせること

IoT デバイスは,持ちうるリソースが限られている.そのため,小リソースで動かせることを要件とする.

要件②検知漏れの可能性をなくす

IoT デバイスを狙うマルウェアは開発者によって様々な箇所に手を加えられている．そのため，マルウェアが必ず行うことを検知条件にすることを要件とする.

要件③頻繁な更新を必要としないこと

IoT デバイスが設置されるネットワーク環境は様々であり，頻繁にアップデートを実施できる環境でない場合もある．さらに，汎用機の利用期間やメンテナンス頻度と比べても，IoT デバイスは長期間利用され放置されてしまう傾向があるため，頻繁に更新を必要としないことを要件とする.

要件④感染後に速やかに対処できること

IoT デバイスは，攻撃の影響が甚大化しやすいという性質がある．そのため，マルウェアを検知後，直ぐに管理者に通達することで被害は最小限に留めることができる．よって，管理者が異常に気付くことができ，かつその後の対応で必要な侵入の痕跡を残せることを要件とする.

5. 実装

本研究では，eBPF を使って実装を行った．従来の eBPF は Python で記述するため多くのリソースを消費してしまう．そのため本研究では C 言語で eBPF プログラムを記述し，実行バイナリのサイズをより小さくするために CO-RE を用いてコンパイルを行った．本研究は図 1 のような構成になっている． IoT デバイスを狙うマルウェアは，標的の環境調査の 1 つとして，IoT デバイスにシェルがあるのか，コマンドが利用可能なのかを調べるために，コマンドではない入力を行う．この特徴は長年変わっていない．検知機能ではこの特徴を用いて検知する．マルウェアを検知すると，管理者が速やかに対処できるようにログ出力を行う．syslog を用いてログ出力を行うことで，ログを独自に管理する必要が無くなる．

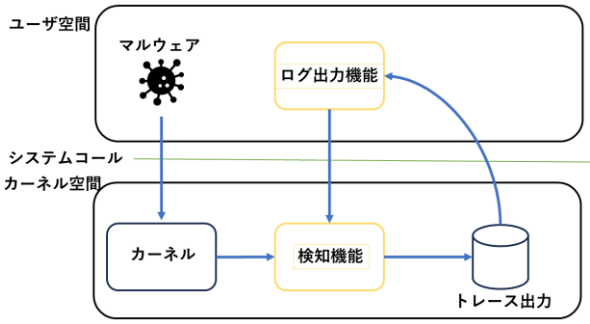

図 1 Edgerunner の概要

6. 評価実験

評価実験ではネットワーク境界型が検知困難なラテラルムーブメントの検知，ログの出力の機能評価を行った．結果は図 2 である．きちんと検知できており，ログも出力出来ている．

攻撃ホスト

Executing command: system at 2024-01-19 23:11
標準出力:
標準エラー:
bash: line 1: system: command not found

被害者ホスト

Jan 19 23:11:24 Raspberrypi EdgerunnerLog[16240]: input command　: bash: line 1: system: command not found#012)#012if [-z "${debian_chr

図 2 攻撃・被害者ホストのログ

また，使用するストレージサイズは 1.4MB となった．

図 3 は CPU 使用率の推移を示している．縦軸は CPU 使用率，横軸は経過時間である．平常時は使用率はほぼ 0%である．しかし，起動時に約 10%に，マルウェアを検知したときは約 30%まで CPU 使用率が跳ね上がっている．

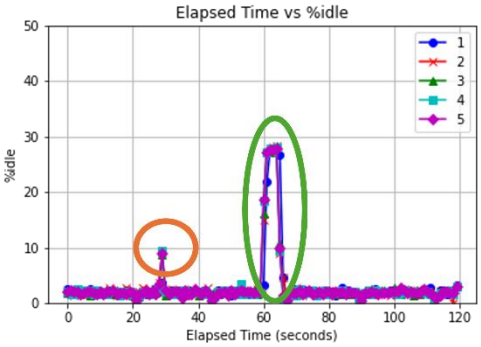

図 3　CPU 使用率の推移

図 4 はメモリ使用量を表している．縦軸はメモリ使用量(KB)，横軸は経過時間である．平常時と検知時はそれぞれ 1300KB，30,000KB になることがわかった．

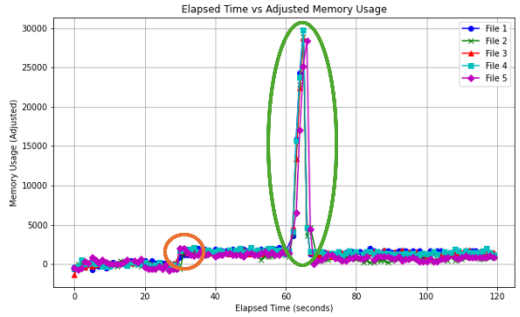

図 4　メモリ使用量の推移

7.考察

本研究で開発したシステムは，ネットワーク境界型では検知が難しい攻撃を検知することができた．さらに，IoT デバイスに導入できる範囲のリソースで動作することも確認ができた．しかし，リソースの小さい IoT デバイスでは，マルウェアを検知した時のリソースの消費量が多いため悪影響が出る可能性がある．今後の課題は，管理者に通達する機能の実装，検知した時のオーバーヘッドの改善を行う．

参考文献

[1] 小池正修，大分大学大学院工学研究科 Linux カーネルモジュールを用いた IoT 機器向けホスト型不正通信検知システムの開発

[2] 秋月 沢竜生，株式会社東芝 研究開発センター Linux 上でのホワイトリスト型実行制御機能 WhiteEgret™の開発

[3] 池上 瑛世，明星大学，“IoT デバイスにおけるセキュリティソフト導入の検討”

[4] 総務省，経済産業省，IoT 推進コンソーシアム，“IoT セキュリティガイドライン ver 1.0 ”