IoT デバイスにおけるセキュリティソフト導入の検討

Consideration of security software implementation in IoT devices

池上瑛世†

Akise IKEGAMI†

† 明星大学情報学部情報学科

† School of Information Science, Meisei University

1. はじめに

総務省の情報通信白書令和 3 年版[1]によれば，世界中の IoT デバイス数は 2020 年までに 253 億台存在している．また，その中でも医療分野や家庭向け家電(コンシューマー)，産業用途といった分野に対して高成長が予想されている．IoT 機器は数や種類，使用用途が豊富であることや，スペック等の問題からアンチウイルスソフトの導入が困難とされており，ユーザ側で有効な対策が実施できないとされている．

また，情報通信研究機構(NICT)が運用するサイバー攻撃観測網(NICTER)が 2020 年に観測したサイバー攻撃関連通信について，約 4 割が IoT 機器を狙った攻撃であったとしている．

こうしたことを背景に，近年の CPU，メモリ等の軽量化が進んでいることに着目し，IoT 機器のスペックを考慮したセキュリティ対策ツールの調査および導入を検討し，問題点についても明らかにする．

2. 関連研究

組み込み機器向けホワイトリスト型マルウェア対策ツール SecNucleus® WhiteEgret(以下，WhiteEgret™)が先行研究として挙げられる．Linux カーネルに標準搭載されている Linux Security Module(以下，LSM)のセキュリティフレームワークを利用した，ホワイトリスト型実行制御機能を有したセキュリティツールとなっている[2]．

また，IoT デバイスにおけるマルウェアの最新動向を調査した論文[3]によると，IoT デバイスでの有効なセキュリティ対策として，IoT デバイスに感染する多くのマルウェアが揮発性メモリに残ることを利用し，デバイスの再起動でマルウェアを除去する方法や，ファームウェアの更新による脆弱性対策といったものが主流となっているようである．

3. 事前準備

ブラックリスト方式:

マルウェアなどの危険な対象をパターンファイルとしてあらかじめリストに定義することで該当する通信やプログラム，アプリケーションの実行などを検知や防御，削除を行うことが可能となる．ブラックリスト方式として代表的なものが，アンチウイルスソフトである

ホワイトリスト方式:

あらかじめ安全な対象をリストに定義することで，リストに登録された以外のプログラムやアプリケーションの実行を防ぎ，それにより，不正な動作を抑制することが可能となる．IP アドレスのアクセス制限や，フィルタリングで使用されるケースが多い．

4. 課題

本研究における課題は以下の 3 点である．

1，IoT 機器を使用するユーザは自発的なセキュリティ対策を実施することができず，現状では有効な対策が存在しないという点．

2，マルウェアに感染したかどうかの確認ができないことから，マルウェアの侵入に際してユーザに通知を行う必要がある点．

3，購入した IoT 機器の設定が初期状態のまま使用されていることが多く，その点を狙った攻撃から侵入されてしまっているという点である．

5. 提案手法

本研究では，ブラックリスト方式の「ClamAV」とホワイトリスト方式の「WhiteEgret™」，「iptables」の導入を試み，処理時間や CPU 使用率，空きメモリ量の調査を行う．

ClamAV はアンチウイルスソフトで，CLI での操作・設定が可能であり，OSS での提供であるため，組み込み Linux にも対応できると思われる．今回は RaspberryPi4 を用いた実験を行う．

WhiteEgret™は東芝情報システム株式会社が開発した，組み込み機器向けの Linux に特化したホワイトリスト型マルウェア対策ツールである．Linux カーネルのビルドに際して WhiteEgret™を導入する必要がある．

iptables は、ファイアウォール機能であり，サーバーへ接続させる通信のルールを設定できるパケットフィルタ．今回では 5000 個から 60000 個のルール設定を行う際の処理性能の違いについて調査を行う．ホワイトリスト方式の実験として RaspberryPi3B+を使用する．

6. 実験

ブラックリスト方式:

使用した Raspberry Pi3B+のディレクトリ階層は右図 1 の通りで，Desktop 以下の eicar.com というファイルがスキャン対象となる擬似ウイルスファイルとなる．

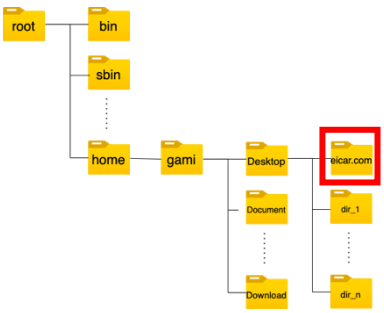

図 1：ディレクトリ階層

スキャンに使用するコマンドは「$ clamscan -r (directory)」.

スキャンは指定した(directory)以下を走査する．今回は3種類(/，～/Desktop，～/Desktop/eicar.com)の指定方法でスキャンを行い，それぞれスキャンにかかった時間，CPU 使用率，空きメモリ量の調査を行った．

ホワイトリスト方式：

・WhiteEgret™：Linux カーネルビルドから行う必要があるため，導入過程についてや，ホワイトリストがどのような仕様になっているかを調査を行った．

・iptables：INPUT(入力)に対して各ポートの 0/tcp から 5000 個ずつ増やし，DROP(拒否)としてフィルターをかけていった際の実行時間や CPU 使用率，空きメモリ量を調べた．5000 個から 60000 個までを「# iptables -A INPUT -p tcp --dport 0 -j DROP」のルールで設定を行い，「--dport 0 ～--dport 60000」を bash スクリプトにて登録を行う．今回の実験ではそれぞれのルール登録数での実験を 10 回ずつ行い，time コマンドと vmstat コマンドを bash スクリプトにてまとめて計測を行った．

7. 実験結果

ブラックリスト方式：

下図 2,3 に ClamAV で実験を行った結果を示す．

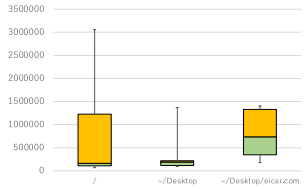

図 2:空きメモリ

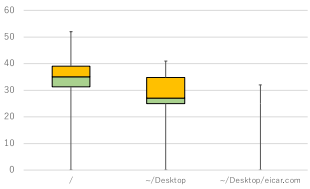

図 3:CPU 使用

それぞれの図の左から/以下，～/Desktop 以下，～/Desktop/eicar.com の走査結果となっている．

Raspberry Pi 4 は使用可能なメモリ量も約 3.2GB あり(仕様上は 4GB)，IoT デバイスとしては高スペックであるため，図2の結果から Raspberry Pi 4 より低スペックのデバイスでの運用は難しいことがわかる．また図3より CPU 使用率に関しては最大でも 52%であったため，より低スペックなプロセッサでも動作できる可能性がある．

ホワイトリスト方式：

・WhiteEgret™：下図 4 は WhiteEgret™を用いた，ホワイトリスト作成時終了後の様子である．root の状態で # wectl create のコマンドで/以下の実行ファイル等を走査しホワイトリストを作成することができた．7028 個のファイルを走査しリストを作成するために要した時間は約 200 秒であった．しかし，kernel や gcc, OS,バージョン，プロセッサの種類によって大きく異なるため，導入後の動作が不安定であることが分かった．

```
/media/pi/UNTITLED/WhiteEgret/src
/media/pi/UNTITLED/WhiteEgret/src/signature
/media/pi/UNTITLED/WhiteEgret/src/signature/include
/media/pi/UNTITLED/WhiteEgret/src/signature/signature_sample
/media/pi/UNTITLED/WhiteEgret/src/signature/signature_sample/src
!! WARNING !! 1 directories are skipped.
!! WARNING !! See /var/log/weusr.log for more details.
Elapsed time : 196.74713119 (sec).
successfully finished.
```

図 4：ホワイトリスト作成

・iptables：

右図5はルール設定にかかる実行時間を示している．

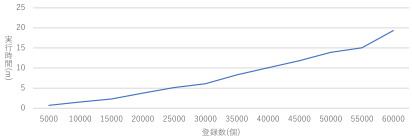

図 5：iptables ルール設定における実行時

5000 個のルールを設定にかかる時間が約 40 秒であり，60000 個では約 19 分であった．また，下図 6 にルール 5000 個と 60000 個の設定時の CPU 使用率を示す．最大でも 52%の使用率で設定が可能になっていることがわかる．

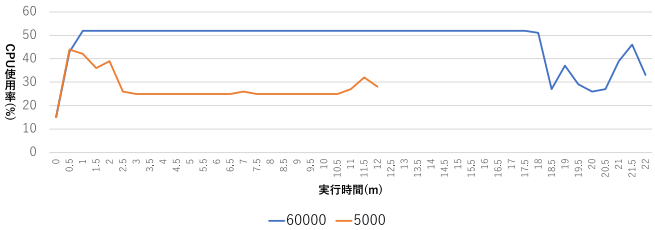

図 6：iptables, 5000 個, 60000 個 CPU 使用率

8. 考察

実験結果から，Raspberry Pi 4 以上のスペックであればアンチウイルスソフトを動作させることが可能である．

また，今回使用した Raspbian は GUI で動作するものを使用しており，メモリ消費が大きいものであった可能性もあることから，サーバー用の OS での検証や，セキュリティソフトのソースコードからどの位置でどの程度のメモリ消費がなされているかといった検証も必要であることが分かった．

9. 今後の課題

消費メモリを抑える方法を模索する必要があることから，既存のウイルスソフトで使用されているマルウェアのパターンを定義ファイルとして格納するブラックリスト型ではなく，実行可能なアプリケーションを定義ファイルとして格納するホワイトリスト型での導入を検討しようと考えている．

また，ネットワーク側でマルウェアの種別を判定・分類を行い，IoT機器への侵入を防ぐといった方法でのセキュリティ維持を検討しても良いかと考えている．

10. 参考文献

[1] 総務省 ．”令和 3 年版 情報通信白書”，https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/pdf/index.html (Accessed on 12/02/2022)

[2] Arwa Abdulkarim Al Alsadi, Kaichi Sameshima, Jakob Bleier, Katsunari Yoshioka, Martina Lindorfer, Michel van Eeten, and Carlos H. Gañán. “No Spring Chicken: Quantifying the Lifespan of Exploits in IoT Malware Using Static and Dynamic Analysis”, In *ASIA CCS '22: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 309-321, May 2022.

[3] 小池正修，小椋直樹，内匠真也，花谷嘉一，春木洋美．“Linux 上でのホワイトリスト型実行制御機能 WhiteEgret™の開発”，コンピュータセキュリティシンポジウム 2017 論文集，2017 1317-1323，2017-10-16