社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

IoT機器に対するマルウェア持続感染性の診断手法の提案

添田　隼喜† 　井上　貴弘† 　田辺　瑠偉† 　YinMinn Pa Pa† 　吉岡　克成†
松本　勉†

† 横浜国立大学
E-mail: †{soeda-shunki-jf,inoue-takahiro-yt}@ynu.jp, ††{tanabe-rui-xj,yinminn-papa-jp,yoshioka,tsutomu}@ynu.ac.jp

あらまし　IoT 機器を標的としたマルウェアの中には, 機器を再起動した後も永続的に感染し続けるものがあり, 持続感染型 IoT マルウェアと呼ばれている．これらの脅威は近年拡大しているが, 持続感染の標的となりうる機器や持続感染した場合の駆除方法は十分に検証されていない．本研究では, 持続感染型 IoT マルウェアの挙動を模擬した持続感染性診断プログラムを作成し, 対象の IoT 機器の持続感染の可能性を検査する手法を提案する．検証実験では, セキュリティベンダーの解析レポート等で持続感染することが報告されている 14 種類の IoT マルウェアファミリーからなる合計 101 検体を収集し,11 種類の IoT 機器を用いて実機動的解析を行った．持続感染挙動を分類した結果, 持続感染を試みる挙動が 42 検体から観測され, 実際に 6 種類のファミリからなる合計 13 検体から 3 つの機器において持続感染が確認された．解析結果を基に持続感染性診断プログラムを作成して動的解析に用いた 11 機器と新たに追加した 1 機器を検査したところ,7 機器において持続感染の可能性が確認された．このうち,1 機器については持続感染性診断プログラムでのみ持続感染が確認された．これらの機器について持続感染を駆除する方法として工場出荷状態に戻す方法や, 持続感染の原因となるファイルの削除や設定の変更を試したところ, 全ての機器において駆除が成功した．しかし，最も有効性が高いと思われた工場出荷状態へ戻す操作が常に有効ではないことも示された．
キーワード　持続感染型 IoT マルウェア, 実機動的解析, 持続感染性診断

Assessment Method for Persistent Malware Infection on IoT Devices

Soeda SHUNKI†, Takahiro INOUE†, Rui TANABE†, Yin MINN PA PA†, Katsunari YOSHIOKA†, and Tsutomu MATSUMOTO†

† Yokohama National University
E-mail: †{soeda-shunki-jf,inoue-takahiro-yt}@ynu.jp, ††{tanabe-rui-xj,yinminn-papa-jp,yoshioka,tsutomu}@ynu.ac.jp

Abstract　Some malware that targets IoT devices can persistently infect devices even after they are rebooted, and are referred to as persistent IoT malware. These threats have been growing in recent years, but the devices that can be the target of persistent infection and the disinfection methods for persistent infection have not been sufficiently verified. In this study, we propose a method to check the possibility of persistent infection of target IoT devices by creating a diagnostic program that simulates the behavior of persistent infection-type IoT malware. In the verification experiment, we collected a total of 101 samples consisting of 14 IoT malware families that have been reported to be persistently infected by security vendors' analysis reports, and conducted the dynamic analysis using 11 types of bare-metal IoT devices. As a result of classifying the persistent infection behavior, 42 specimens were observed to attempt persistent infection, and three devices from a total of 13 specimens from six different families were actually confirmed to be persistently infected. Based on the analysis results, we created a diagnostic program for persistent infection and tested 11 devices used in the dynamic analysis and one newly added device. Of these, one device was confirmed to have persistent infection only by the diagnostic program for persistent infection. We tried several methods to eliminate the persistent infection from these devices, including restoring the devices to factory defaults, deleting files that cause persistent infections, and changing settings. However, the factory reset method thought to be the most effective, was not always effective.
Key words　Persistent IoT malware, Bare-metal dynamic analysis, Persistence diagnosis

1. はじめに

脆弱な IoT 機器に侵入し,DDoS 攻撃等の不正な活動を行う IoT マルウェアのうち,Mirai を代表とする IoT マルウェアの多くは機器の再起動によりプロセスが停止するため, 感染前の状態に戻すことができる [1]．しかし, 機器を再起動しても駆除できない持続感染性を有する IoT マルウェアの報告が増えており, その脅威が高まりつつある．実際に,2022 年だけでも Cyclops Blink [2] や Kaiji [3] と呼ばれる持続感染型 IoT マルウェアが報告されている．持続感染の脅威はウイルス対策ソフトなどの対策が適用できない IoT 機器の利用者にとって大きな脅威であるため, 早急な対応が求められている．

これまでに, 持続感染型 IoT マルウェアの実際を明らかにする研究が行われている．文献 [30] では,1 万件以上の Linux マルウェア検体を解析して持続感染の対象となるファイルパスを明らかにしている. 文献 [31] では,166 万件以上の IoT マルウェア検体を解析して持続感染挙動が観測されたことを明らかにしている．文献 [8] では,10 種類の持続感染型 IoT マルウェアを解析してその持続感染挙動を明らかにしている．文献 [32] では,IoT マルウェアの挙動を効果的に解析するシステムが提案されており, 持続感染型 IoT マルウェアの挙動観測にも有効である．文献 [33] では, 持続感染型 IoT マルウェアの挙動を解析するシステムが提案されている．しかし, 実機を用いた持続感染の検証は行われていない．また, 持続感染型 IoT マルウェアを機器から駆除する手順は検討されていない.

一方で,IoT 機器の持続感染の可能性を明らかにする研究が行われている．文献 [4] [5] では, 実機を用いて持続感染型 IoT マルウェアの実現性を調査している．文献 [6] では,IoT 機器における持続感染のメカニズムを 6 つに分類し, 実機を用いて持続感染の再現をおこなっている．文献 [7] では, 機器の不正ログイン後の悪用可能性を調査し, マルウェアの悪性挙動の実現可能性を調査している．しかし,IoT 機器が持続感染する可能性を検査する方法は我々の知る限り提案されていない．そこで本研究では, 持続感染型 IoT マルウェアを実機を用いて解析することで持続感染のメカニズムを明らかにするとともに, その挙動の一部を模擬した持続感染性診断プログラムを作成して IoT 機器の持続感染の可能性を検査する方法を提案する．また, 持続感染が確認された機器については駆除手順を調査する．検証実験では, 持続感染性を持つことが報告されている [3] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] マルウェア 14 ファミリ 101 検体を VirusTotal [23] から収集した．そして, 実機 11 機種を用いて合計 157 回の動的解析を行い, 持続感染挙動を調査した．組み込み機器の多くは Linux [24] をベースとしたものであり，Linux には起動時に実行される処理を設定する機能が複数存在し，そのほかにも特定の処理を定期実行を設定する機能等が存在する．持続感染性を持つ IoT マルウェアの多くは，このような Linux の自動実行に関連するファイルを持続感染のために利用することが報告されている．その結果, 自動実行に関係するファイルを参照しており, 持続感染の可能性がある解析結果 (レベル 1) が 10 件, 自動実行に関係するファイルの参照, 書き換え, 生成を行っており, 持続感染の可能性が高い解析結果 (レベル 2) が 19 件, 実際に持続感染が観測された解析結果 (レベル 3) が 13 件確認された．これらの解析結果を基に, 自動実行に関連するファイルを対象とした 8 パターンの持続感染手法を検査する診断プログラムを作成し, 動的解析に用いた 11 機器と新たに追加した 1 機器, 合計 12 種類の機器に対して持続感染の可能性を調査した．そして, 実 IoT マルウェア検体による持続感染が確認されていなかった 1 機種を含む 7 機種で持続感染を確認した．また, これらの機器に対して手動でのコマンド操作を用いた改ざん箇所の修復は全ての機器に対して有効な駆除方法であったが，工場出荷状態に戻す操作においては，１機種において有効ではない事例が確認された.

2. 関連研究

前述の通り,IoT マルウェアの持続感染性に関する研究が進められている. 文献 [30] では,1 万件以上の Linux マルウェアを解析してその挙動を把握するとともに, 持続感染挙動についても明らかにしている．文献 [31] では,166 万件以上の IoT マルウェアのを解析してその挙動を把握するとともに, 持続感染挙動について明らかにしている. 文献 [8] では, 様々な IoT 機器のファームウェアを展開した複数のサンドボックスで動的解析を行うことで,10 種類の IoT マルウェアの持続感染挙動を明らかにしている．これらの研究は持続感染型 IoT マルウェアについて有益な知見を提供するが, 実機を用いた解析が行われていないため, 持続感染の可能性がある機器についてはさらなる調査が必要である.

文献 [32] では,IoT マルウェアの挙動を効果的に解析するシステムが提案されており, 実装したシステムを用いて持続感染型 IoT マルウェアの挙動を明らかにしている．文献 [33] では, 持続感染型 IoT マルウェアの挙動を解析するシステムが提案されており, 実験に用いた検体の持続感染先ファイルパス情報を明らかにしている．これらの研究は持続感染型 IoT マルウェアを解析する上で有効であるが, 持続感染型 IoT マルウェアを機器から駆除する手順は検討されていない.

一方で,IoT 機器の持続感染性に関する研究が行われている．文献 [5] では, 実機 10 機種に対してシステム起動時に実行されるスクリプトの改ざんを行う擬似マルウェアを用いて持続感染の実証実験を行い,4 機種において擬似マルウェアプロセスの機器再起動後の残存を確認している．文献 [4] でも実機を用いて持続感染型 IoT マルウェアの実現性を調査しており, 改変したファームウェアに機器を更新させることで持続感染の実証に１機種で成功したと報告されている．文献 [6] では,IoT 機器における持続感染のメカニズムを 6 つに分類するとともに, 実際に 4 つの実機に対して持続感染の再現に成功している．文献 [7] では, インターネット上で発見された脆弱な ID/パスワードが設定された 17 ベンダ 36 機種について侵入後の悪用可能性を調査しており,14 機種で持続感染に成功したと報告されている．これらの研究は持続感染型 IoT マルウェアへの対策を考える上で有効であるが,IoT 機器が持続感染する可能性を検査する方法は我々の知る限り提案されていない.

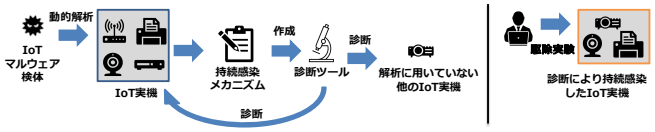

図 1　本研究の流れ

IoT マルウェアにおける手動での持続感染手法は大別すると,(1) 書き換え可能な領域に存在するファイルを改ざんすること,(2) ファイルの読み取りのみ可能な状態を書き換え可能に変更すること,(3) 機器固有の仕様を利用すること, の 3 つに分類できる．本研究では, 上記の手法のうち (1)(3) に着目し, 持続感染型 IoT マルウェアの実機での解析結果を基に持続感染性を診断するプログラムを作成して機器の持続感染の可能性を検査する手法を提案する．

3. IoT マルウェア実機動的解析システム

本研究では, 実際の IoT 機器を用いて構築した動的解析環境で IoT マルウェア検体を解析し, 観測された持続感染挙動を分析する．次に, 観測された持続感染挙動を参考に持続感染診断プログラムを作成し,IoT 機器の持続感染性の診断を行う．図 1 に本研究の流れを示す．本章では, 実機を用いた動的解析方法と持続感染挙動の分類方法を提案する．

3.1 IoT マルウェア検体と IoT 機器の選定

はじめに, 持続感染挙動を観測する可能性を高めるため, 持続感染する可能性の高い IoT マルウェア検体と IoT 機器のペアの選定を行う．具体的には, 実験に用いた各機器に対して, 各マルウェアファミリが持続感染のためにアクセスするファイルパスが存在するか, かつ, 不揮発領域であるかを調査する．その結果を基に, 各マルウェアファミリに対して以下の 2 つの条件を満たす機器でのみ動的解析を行う．

- 収集した検体の CPU アーキテクチャが機器と一致する．
- IoT マルウェアが持続感染に成功するための必要最低限のファイルパスが機器に存在していること．

3.2 実機動的解析の流れ

続いて，機器を工場出荷状態に戻し, システムコールのログを取得するための strace [25] のバイナリと検体を機器内に配置し, 実行権限を付与する．次に,LAN と WAN の通信の観測を開始する．そして，strace コマンドを用いて検体を 2 分間実行したのち strace のプロセスを全て終了させる．その後,strace のログを解析用のマシンに移動させたのち機器を再起動する．最後に,strace の結果を用いて持続感染挙動の分析を行う．本研究では, 持続感染には複数の段階が存在すると考え, 動的解析の結果を表 1 に示す 6 段階に分類する．再起動後にマルウェアによる外部への通信が発生していた場合に持続感染が確認されたと分類する（レベル 3）．また，実際に持続感染が確認されなかった場合でも，自動実行に関するファイルの書き換えや生成が観測された場合に持続感染の可能性が高いと分類し（レベル 2），自動実行に関係するファイルを参照していた場合に持続感染の可能性があると分類する (レベル 1)．一方，シェルスクリプトなど, ソースコードが確認できた検体に関しては静的解析を行うことで持続感染性を持つか判定した (レベル-2)．

表 1　持続感染レベル一覧

持続感染 Lv	詳細
-2	静的解析によって持続感染性はないと確認された
-1	本動的解析において持続感染挙動が観測されなかった
0	実行時エラーによって検体の持続感染性が不明なもの
1	自動実行に関するファイルへの参照が観測された
2	自動実行に関するファイルの書き換えや生成が観測された (write,writev,rename)
3	解析機器に持続感染したことが本動的解析において確認された

3.3 実機動的解析システムの実装

マルウェアのシステムコールの実行履歴を取得して持続感染挙動の詳細を把握するために, あらかじめ機器内に静的コンパイル済の strace のバイナリを配置した．これにより本来 strace が使用できない機器でも詳細な持続感染挙動の観測が可能とした．本研究は持続感染挙動の観測が目的のため, 解析対象の IoT 機器はインターネットには接続せず検体,strace バイナリの配置,webUI の操作, 解析結果の保存を目的とした windows マシン 1 台と接続して動的解析を行った．接続する際は,LAN ポートと WAN ポートが存在する場合は 2 つとも接続し, 両方の通信を wireshark [29] で観測した．また, 再起動後に検体自身または自身のコピーを再度実行するマルウェアファミリの検体は, 検体を機器に不揮発領域が存在する場合は不揮発領域に配置して実行するようにした．

4. IoT 機器の持続感染性診断手法

本研究では, 実機動的解析によって得られた持続感染挙動を参考に持続感染診断プログラムを作成し,IoT 機器の持続感染性の診断を行う．本章では持続感染診断手法を提案する．

4.1 持続感染診断の流れ

はじめに，実機動的解析の結果，持続感染が確認された（レベル 3），あるいは，持続感染の可能性が高い（レベル 2）検体の持続感染挙動を分析し, 持続感染の原因となる仕組みを特定する．そして, 実行された IoT 機器内に持続感染の原因となる自動実行に関するファイルが不揮発領域に存在した場合に自動実行の設定を変更することで, 持続感染性を診断するプログラムを作成する．以降では,IoT 機器の持続感染の可能性を診断プログラムを用いて検証する流れを説明する．

（1） 評価対象機器の任意のディレクトリに実行権限を付与した状態で診断プログラムを配置する．診断プログラムを実行すると, 本プログラムによって機器内の全てのディレクトリに空ファイルが生成されるため，生成完了後に，手動で機器を再起動する．

（2） 再度診断プログラムを実行し，手順 (1) で作成した空ファイルが残っているディレクトリを探索することで書き換え可能な不揮発領域の特定を行う．診断プログラムによって該当の領域が見つかった場合には，本プログラムは該当領域内の全てのファイルに対し表 7 に示す持続感染手法の中の Systemd,cron 登録スクリプトを除く 7 つを試す．その後，手動

表 2　駆除手法一覧

ID	詳細
コマンド操作による復旧	Linux コマンドを用いて改ざんされた部分を修復する
ファームウェア再インストール	ファームウェアの再インストールを行う
工場出荷状態へのリセット	マニュアルに従い機器を工場出荷時の状態に戻す

表 3　診 断 結 果

機器 ID	有効持続感染手法 (解析)	有効持続感染手法 (診断)	有効であった駆除手法
I	cron,cron 登録スクリプト	cron	コマンド操作による復旧 工場出荷状態へのリセット
II	なし	none	–
III	なし	none	–
IV	なし	none	–
V	なし	none	–
VI	なし	inittab	コマンド操作による復旧
VII	なし	none	–
VIII	cron	cron	コマンド操作による復旧 工場出荷状態へのリセット
IX	rcX	bashrc,rclocal,profile,rcX	〃
X	cron	cron,profile	〃
XI	cron,systemd	cron,profile	〃
XII	未解析	rclocal	〃

で機器を再起動する．

（3） 揮発領域に指定したテキストファイルが生成されているかを手動で確認し，生成されていれば持続感染性を検出したと診断する. 5 章の実験では/tmp 下にあるファイルを確認し, どの持続感染手法が有効であったかを確認する．診断プログラムが自動実行された際に生成される空ファイルがないかを確認し, どの手法が有効であったかを確認する．

（4） 持続感染が確認できた場合は表 2 に示す 3 通りの駆除方法を 1 回ずつ行い結果を確認する（4.3 節にて説明）．

4.2　持続感染診断プログラムの実装

検証実験では, 動的解析結果から特定した手法を含む 7 つの持続感染手法および不揮発領域の調査を行うことができる診断プログラムを作成した．診断プログラムは, 4.1 節にて述べたように，機器内に配置して実行することで，書き換え可能な不揮発領域を特定し，その領域内に存在するファイルに対して持続感染の可能性を検査する機能を有する．

本診断プログラムは，C 言語で作成したプログラムを静的リンクでクロスコンパイルしたバイナリまたはシェルスクリプトを用いた．クロスコンパイルの際は，機器のカーネルのバージョン，CPU アーキテクチャによって動作するバイナリが異なるため，*mips-linux-gnu-gcc* 等の GCC(GNU Compiler Collection) のクロスコンパイラおよび文献 [26] で Mirai ファミリのバイナリの作成に用いられる uClibc [27] のクロスコンパイラを利用した．また, 本プログラムは, 機器内に配置したのち chmod コマンドによって実行権限を付与した上で実行しているが, 書き込み不可の領域や権限不足等の理由により空ファイルの生成が行えない領域に関しては, 診断対象外となっている．本プログラムを実行する際は, 機器のマニュアルや WebUI, インターネットの情報を確認した上で最も権限が高いユーザでログインし, 診断プログラムを実行した．また, ユーザの情報がなくとも Mirai が用いた脆弱なパスワードリスト [28] を試行するこ

表 4　収集した IoT マルウェア検体一覧

ファミリ名	CPU アーキテクチャ	検体数
Hajime	mips	1
HideNSeek	i386, mips, x86_64	7
Icnanker	i386	4
Momentum	i386	1
QSnatch	shell script,x86$_6$4	8
RHOMBUS	shell script, x86_64	3
vbot	i386	4
XORDDoS	i386	3
Amnesia	mips,x86_64,i386	14
Torii	mips,aarch64,x86_64,i386	39
VPNfilter	ASCII text, i386, arm, mips	6
Chaos	arm,i386	2
Kaiji	arm	2
ZeroBOT	mips, arm, aarch64	7
合計	-	101

とで root ユーザまたは同等のユーザにログインできた際はそのユーザで診断を行なった．

4.3　持続感染の駆除

診断プログラムにより持続感染が確認された機器については, 持続感染を駆除するために 3 つの手法を試す．表 2 に駆除手法の内訳を示す．手法「コマンド操作による復旧」は Linux の rm コマンドや vi コマンドを用いて検体の削除，または, 攻撃対象となったファイルを再編集することで駆除を行う．手法「ファームウェア再インストール」は, 新たな Firmware を再インストールすること駆除を試みる方法である．ただし, この方法は,Firmware のバージョンを変更後に，実験開始時のバージョンに戻すことができる場合のみ行う．手法「工場出荷状態へのリセット」は, 機器をマニュアルや web ページに記載されている方法を用いて工場出荷状態に戻すことで駆除を行う．これには, 機器の設定ファイルのみの初期化や機器の完全初期化などが含まれるが, 本実験では対象機器で行える最も初期化できる領域が多い方法を手法「工場出荷状態へのリセット」とする．

5.　検 証 実 験

5.1　実機動的解析

動的解析には, 表 4 に示す 14 ファミリ 101 検体をオンライン上のファイル検査サービスである VirusTotal[5] から収集した．これらの検体はセキュリティベンダの解析レポート等 [9] 〜 [19] にて持続感染性が示唆されている持続感染型 IoT マルウェアのみで構成されている．また, 持続感染する可能性の高い検体と機器のペアの選定を行った．以降では,101 検体 11 機種のうち,124 の検体と機種のセットの動的解析を行なった．表 5 に機器の一覧を示す.

はじめに, 持続感染挙動の分類手法を用いて検体ごとに結果を分類した．その結果を表 6 に示す. また, ※のついている QSnatch と Chaos の結果に関しては,VirusTotal から入手した検体やマルウェア自身が生成したソースコードに一部改良を加えることで持続感染に成功したため条件付きの Lv.3 としている. 加えて，Lv.2 と 3 の検体に関しては，その挙動が観測された機器の ID を検体数の横に記載している．

解析した 101 検体のうち,13 検体で持続感染レベル 3 を確認

表 5　実機動的解析に用いた IoT 機器一覧

機器 ID	ベンダ ID	機器の種類	CPU アーキテクチャ
Ⅰ	A	NAS	x86_64
Ⅱ	B	ルータ	mips
Ⅲ	C	ルータ	mipsel
Ⅳ	D	ルータ	mips
Ⅴ	E	ルータ	mips
Ⅵ	F	ルータ	mips
Ⅶ	G	ルータ	mips
Ⅷ	A	NAS	x86_64
Ⅸ	D	ルータ	mips64
Ⅹ	H	NAS	aarch64
Ⅺ	H	NAS	aarch64

表 6　検体毎の持続感染レベルの調査結果

	各レベル毎の検体数 (その挙動が確認された機器の ID)					
ファミリ名	Lv.-2	Lv.-1	Lv.0	Lv.1	Lv.2	Lv.3
Hajime	0	0	0	0	0	1(Ⅴ)
HideNSeek	0	0	2	5	0	0
Icnanker	0	0	4	0	0	0
Momentum	0	0	1	0	0	0
QSnatch	0	3	1	0	2(Ⅰ, Ⅷ, Ⅺ)	2(Ⅰ) ※
RHOMBUS	0	2	0	1	0	0
vbot	0	4	0	0	0	0
XORDDoS	0	0	0	0	3(Ⅰ)	0
Amnesia	0	0	11	0	3(Ⅰ, Ⅷ)	0
Torii	1	8	15	4	8(Ⅰ, Ⅲ)	3(Ⅸ)
VPNfilter	1	2	1	0	0	2(Ⅰ, Ⅷ)
Chaos	0	0	0	0	1(Ⅰ)	1(Ⅹ, Ⅺ) ※
Kaiji	0	0	0	0	0	2(Ⅹ, Ⅺ)
ZeroBOT	0	3	0	0	2(Ⅸ, Ⅹ, Ⅺ)	2(Ⅺ)
合計	2	22	35	10	19	13

※一部改良を加えた

表 7　自動実行の手法

ID	詳細
cron	指定したプログラムの定期実行設定を行う cron ファイルの書き換え
cron 登録スクリプト	cron ファイルによって自動実行されるように登録されているスクリプトの書き換え
initd	起動時に自動実行されるファイルを格納する init．d ディレクトリ以下へのファイル生成
inittab	起動時に自動実行するファイルの設定を行う inittab ファイルの書き換え
bashrc	対話シェル (bash) を起動した際に自動実行されるファイルの設定を行う.bashrc ファイルの書き換え
profile	対話シェルを起動した際に自動実行されるファイルの設定を行う．bash_profile, . profile の書き換え
rcX	起動時に自動実行されるファイルを格納する rc から始まるファイルまたはディレクトリ内のファイルの編集
rclocal	起動時に自動実行する処理の設定を行う rc.local の書き換え
systemd	起動時に実行する処理設定用ファイルの作成

することができた．加えて, レベル 2,3 の合計 32 検体が自動実行のために取った手法は表 7 の cron，cron 登録スクリプト，initd，inittab，bashrc,rcX,Systemd の 7 つにまとめられることがわかった．この結果より, 今回解析した IoT マルウェアがとる持続感染挙動は,Linux の自動実行に関する設定ファイルの書き換えや生成を行うものがほとんどであることがわかった．

5.2　持続感染性診断

本実験では,3 節にて解析に使用していない新たな 1 機器 (機器 ID：Ⅻ, ベンダ：I，Arm のルータ) を加えた合計 12 機種に対して,4 節で述べた診断プログラムを実行することで機器に持続感染の可能性があるかを調査する．持続感染に成功した場合, その原因の特定まで行う．3 節にて述べた動的解析に用いた機器のいくつかと新たな機器を含めた表 5 に示す 12 機種を用いて診断実験を行った．

診断の結果を以下の表 3 に示す．結果として,12 機種のうち 7 機種で有効な持続感染手法が発見された．加えて,1 つの機器に対して複数の持続感染手法が有効であるケースが 3 機種で発見された．持続感染性を検出した 7 機種全てに対し, 手動でのコマンド操作による検体の削除が有効であり，6 機種に対して工場出荷状態へ戻す操作が有効な駆除方法であることがわかった．また実マルウェアの動的解析によって発見された持続感染手法以外の方法での持続感染性の検出に 2 機種において成功した．

5.3　考　　察

検証実験の結果, 今回診断を行なった 12 機種のうち 7 機種において持続感染の可能性があることが確認された．今回は, 書き換え可能かつ不揮発である領域に存在する自動実行に関連するファイルへの手法 8 つのみを用いた診断であったが, それでも持続感染が確認されたことから, 診断プログラムは IoT 機器の持続感染の可能性を十分に発見できることが示された．また, 機器Ⅸ, Ⅹ, Ⅺでは複数種類の持続感染手法が有効であることが確認されており, ベンダー側で 1 つの危険性に気づき修正をした場合でも, 持続感染への対策としては不十分であることも示された．また，全ての機器でコマンド操作による駆除が有効であったが，この手法は一般のユーザが実行するには難易度が高くなることが予想される．加えて, 改ざんされたファイルの特定を行う必要があるため, 新規のマルウェアファミリに対しては適用が難しいことが考えられる．次に, 工場出荷状態に戻す方法は 6 機種に対して有効であったが, 全ての機器に対して有効ではなかった．WebUI からの操作や, 説明書にも記載があるため, 一般のユーザによる実行難易度も低く, 有効なケースも多いため現時点では最も効果の高い駆除方法だと考えられる．

5.4　研 究 倫 理

本研究は, 持続感染の可能性を診断するプログラムを作成することで, 持続感染型 IoT マルウェアの標的となりうる IoT 機器のセキュリティ向上に資することを目的とする．そのため, 持続感染やその可能性が示された機器のベンダーに対しては情報提供を行う予定である．また, 診断プログラムや本研究成果の一部を研究者向けに公開する予定である．論文の情報が攻撃者に直接的に悪用されないようにするため, 機器の名前やベンダーの名前などの記述は避けるとともに, 持続感染に有効なファイルパス情報を記述することは避けた．

6.　まとめと今後の課題

セキュリティベンダーの解析レポート等で持続感染が報告されている 101 検体を 11 種類の実機で動的解析し,13 検体から 5 つの機器で持続感染が確認された．実機動的解析で観測された持続感染挙動とセキュリティベンダー等の解析レポートに記載されている情報を基に持続感染手法を 9 つに分類し, 持続感染性を診断するプログラムを作成したところ,7 つの機器で持続感染が確認された．これらの機器について持続感染の駆除を試したところ, 全ての機器で駆除が成功した．しかし，工場出荷状態に戻す操作が有効でない駆除手法である機器が存在することも確認された．今後は,(1) より多くの IoT マルウェ

アと実機の組み合わせで持続感染挙動の調査,(2) 持続感染診断方法の追加,(3) より多くの IoT 機器での持続感染性診断等を行う予定である.

謝辞: 本研究は総務省の「電波資源拡大のための研究開発 (JPJ000254)」における委託研究「電波の有効利用のための IoT マルウェア無害化／無機能化技術等に関する研究開発」によって実施した成果を含みます．本研究成果の一部は, 国立研究開発法人情報通信研究機構の委託研究（05201）により得られたものです.

文　献

[1] 田宮 和樹, 中山 颯, 江澤 優太, 鉄 穎, 呉 俊融, 楊 笛, 吉岡 克成, 松本 勉,"IoT マルウェア駆除と感染防止に関する実機を用いた実証実験", 暗号と情報セキュリティシンポジウム 2017, セッション 3E1-5, 2017

[2] National Cyber Security Centre, "Cyclops Blink Malware Analysis-Report", https://www．ncsc．gov．uk/files/Cyclops-Blink-Malware-Analysis-Report．pdf, 参照 Dec． 2022

[3] INTEZER, "Kaiji: New Chinese Linux malware turning to Golang", https://www.intezer.com/blog/research/kaiji-new-chinese-linux-malware-turning-to-golang/, 参照 Dec. 2022

[4] 原 悟史, 渡辺 露文, 田宮 和樹, 吉岡 克成, 松本 勉,"IoT マルウェアの持続的感染の成立要因の分析と実機による検証", コンピュータセキュリティシンポジウム 2017 論文集, 2017

[5] 原 悟史, 田宮 和樹, 鉄 穎, 渡辺 露文, 吉岡 克成, 松本 勉, 感染持続型 IoT マルウェアの実態調査と実機による概念実証. 電子情報通信学会論文誌 B J102-B(8), 524-535, 2019

[6] C． Brierley, J． Pont, B． Arief, D． J． Barnes, and J． Hernandez-Castro,"Persistence in Linux-Based IoT Malware," The 25th Nordic Conference on Secure IT Systems, pp． 3 − 19, Oct． 2020.

[7] 村上　恍介, 笠間 貴弘, 井上 大介,"実機を使用した不正ログイン後の IoT 機器悪用可能性の調査", 電子通信情報学会,2022

[8] 井上貴弘, 岡田英造, 岡田晃市郎, 塩治榮太朗, 秋山満昭, 田辺瑠偉, 吉岡克成, 中尾康二, 松本勉, "IoT 機器のファイル構成を模したサンドボックスによる持続感染型 IoT マルウェアの実行環境依存性の分析", 電子通信情報学会, 2022

[9] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, Dave Levin, "Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet",NDSS,2019

[10] Unit 42,"Hide ʻN Seek Botnet Updates Arsenal with Exploits Against Nexus Repository Manager & ThinkPHP",https://unit42.paloaltonetworks.com/hide-n-seek-botnet-updates-arsenal-with-exploits-against-nexus-repository-manager-thinkphp/, 参照 Dec. 2022

[11] Netlab 360,"Icnanker, a Linux Trojan-Downloader Protected by SHC",https://blog.netlab.360.com/icnanker-trojan-downloader-shc-en, 参照 Dec. 2022

[12] TRENDMICRO,"Momentum Botnet's Newest DDoS Attacks and IoT Exploits",https://www.trendmicro.com/en_us/research/19/l/ddos-attacks-and-iot-exploits-new-activity-from-momentum-botnet.html, 参照 Dec. 2022

[13] Security Scorecard, "QSnatch Technical Report", https://securityscorecard.pathfactory.com/cybersecurity/qsnatch-technical-report, 参照 Dec. 2022

[14] APNIC,"RHOMBUS: a new IoT malware",https://blog.apnic.net/2020/05/22/rhombus-a-new-iot-malware/, 参照 Dec. 2022

[15] Netlab 360,"The Gafgyt variant vbot seen in its 31 campaigns",https://blog.netlab.360.com/the-gafgyt-variant-vbot-and-its-31-campaigns/, 参照 Dec. 2022

[16] Microsoft Security, "Rise in XorDdos: A deeper look at the stealthy DDoS malware targeting Linux devices", https://www.microsoft.com/security/blog/2022/05/19/rise-in-xorddos-a-deeper-look-at-the-stealthy-ddos-malware-targeting-linux-devices, 参照 2022

[17] Unit 42, "New IoT/Linux Malware Targets DVRs, Forms Botnet", https://unit42.paloaltonetworks.com/unit42-new-iotlinux-malware-targets-dvrs-forms-botnet/, 参照 Dec. 2022

[18] Avast Blog, "Torii botnet - Not another Mirai variant", https://blog.avast.com/new-torii-botnet-threat-research, 参照 Dec. 2022

[19] Cisco, "New VPNFilter malware targets at least 500K networking devices worldwide", https://blog.talosintelligence.com/2018/05/VPNFilter.html, 参照 Dec. 2022

[20] LUMEN,"Chaos Is A Go-Based Swiss Army Knife Of Malware",https://blog.lumen.com/chaos-is-a-go-based-swiss-army-knife-of-malware/, 参照 Dec． 2022

[21] ,INTEZER,"Kaiji: New Chinese Linux malware turning to Golang",https://www.intezer.com/blog/research/kaiji-new-chinese-linux-malware-turning-to-golang/, 参照 Dec. 2022

[22] Zane Gittins, Michael Soltys, "Malware Persistence Mechanisms","24th International Conference on Knowledge − Based and Intelligent Information & Engineering Systems", 176, 88-97, 2020

[23] "VirusTotal",https://www. virustotal. com/, 参照 Dec. 2022

[24] "Linux",https://www.linuxfoundation.org/, 参照 Feb. 2023
"init",https://linux.die.net/man/8/init, 参照 Feb. 2023
"crond",https://man7.org/linux/man-pages/man8/cron.8.html, 参照 Feb. 2023

[25] "strace",https://strace.io/, 参照 Dec. 2022

[26] "Build Mirai botnet (I): Compile Mirai Source", https://www.cdxy.me/?p=746, 参照 Feb. 10, 2023.

[27] https://uclibc.org/downloads/, 参照 Feb. 10, 2023.

[28] https://github.com/danielmiessler/SecLists/blob/master/Passwords/Malware/mirai-botnet.txt, 参照 Dec． 2022

[29] "wireshark",https://www.wireshark.org/

[30] Brierley, Calvin, Jamie Pont, Budi Arief, David J. Barnes, and Julio Hernandez-Castro. "Persistence in linux-based iot malware." In Secure IT Systems: 25th Nordic Conference, NordSec 2020, Virtual Event, November 23 − 24, 2020, Proceedings 25, pp. 3-19. Springer International Publishing, 2021.

[31] Alrawi, Omar, Charles Lever, Kevin Valakuzhy, Ryan Court, Kevin Z. Snow, Fabian Monrose, and Manos Antonakakis. "The Circle Of Life: A Large-Scale Study of The IoT Malware Lifecycle." In USENIX Security Symposium, pp. 3505-3522. 2021.

[32] Darki, Ahmad, and Michalis Faloutsos. "RIoTMAN: a systematic analysis of IoT malware behavior." In Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies, pp. 169-182. 2020.

[33] 井上貴弘, 原悟史, 榊博史, 岡田晃市郎, 塩治榮太朗, 秋山満昭, 佐々木貴之, 田辺瑠偉, 吉岡克成, 中尾康二 and 松本勉, 2021. 適応的サンドボックスによる持続感染型 IoT マルウェアの解析. 研究報告セキュリティ心理学とトラスト (SPT), 2021(21), pp.1-6.