Vol.2019-CSEC-86 No.9
Vol.2019-SPT-34 No.9
2019/7/23

擬似C&Cサーバを用いたIoTマルウェア駆除手法の検討

三須 剛史[1,a)]　高田 一樹[1,b)]

概要：近年，IoT 機器を狙ったマルウェアが猛威を奮っている．IoT 機器を狙ったマルウェアの中でも，特に活発な感染活動が確認されている mirai，bashlite，tsunami 等は，いずれも感染後に C&C サーバから指令（コマンド）を受信することで，そのコマンドに応じた動作を行うという特徴がある．我々はこれまでに，マルウェアのプロセス終了（駆除）を目的として，C&C サーバの通信を模擬したプログラム（擬似 C&C サーバ）から自滅指令（キルコマンド）を送信することで，感染した端末に直接手を加えることなく駆除する試みについて報告した．その報告において有効性を示すと共に，複数の課題が明らかになった．中でも本稿では課題の 1 つである擬似 C&C サーバのシステムを構築し，さらにシステムの運用における 2 つの検討を行なった．1 つ目はキルコマンドはマルウェア毎に異なるため，個々のマルウェアを静的解析する必要があるが，静的解析を行うには専門的な知識と多くの時間がかかる．よって，マルウェア内からキルコマンドの候補を自動で抽出する手法について検討した．2 つ目はキルコマンド等を持たないマルウェアも存在するため，キルコマンドによらない駆除手法について検討した．

Investigation of IoT malware disablement method using dummy C&C server

1. はじめに

近年，IoT 機器の急速な普及に伴い，IoT 機器を狙ったマルウェアが猛威を奮っている．IoT 機器固有の課題として，コストの観点からセキュリティ対策が省かれることが想定される点や，これまで考慮されていなかった様々な分野の機器の接続が想定される点が挙げられ，これらが被害拡大の要因となっていることが考えられる [1]．

IoT 機器を狙ったマルウェアの中でも，特に活発な感染活動が確認されている mirai[2]，bashlite[3]，tsunami[4] 等は，いずれも感染後にコマンド・アンド・コントロールサーバ（以下，C&C サーバと呼ぶ）から指令（コマンド）を受信することで，そのコマンドに応じた動作を行うという特徴がある．

我々はこれまでに，マルウェアのプロセス終了（以下，駆除と呼ぶ）を目的として，感染した端末に直接手を加えることなく IoT マルウェアを駆除する手法について報告した [5]．具体的には，IoT ゲートウェイ内に設置した C&C サーバを模擬したプログラム（以下，擬似 C&C サーバと呼ぶ）がマルウェアに対して駆除情報を送信することで駆除を行う．これにより，IoT 機器自体に手を加えることなくマルウェアの駆除を行えるため，利便性を損なうことなくセキュリティの向上を図ることが期待される．なお，駆除情報とはマルウェアに送信される自滅指令（以下，キルコマンドと呼ぶ）や，それに類するコマンド等を指す．予備調査では，mirai と bashlite について GitHub[6] 等で公開されているソースコードを静的解析し，mirai にはキルコマンドは存在しないが，特定のコマンドと機能を組み合わせることでプロセスを終了させられることが分かった．また，bashlite にはキルコマンドが存在することがわかった．実験の結果，IoTPOT[7] にて収集された 413 検体の内，それぞれ，mirai が約 48%，bashlite が約 91%の割合で駆除が可能であった．

この報告 [5] において，大別して 3 つの課題が明らかになった．

(1) 擬似 C&C サーバのシステム化

擬似 C&C サーバは，IoT マルウェアの駆除実験において簡易的に作られたものであり，実用性などは考慮されていない．このため擬似 C&C サーバのシステム化を行う必要がある．

[1] 株式会社セキュアブレイン
[a)] takeshi_misu@securebrain.co.jp
[b)] kazuki_takada@securebrain.co.jp

このシステムを運用する上で，キルコマンドを抽出しシステムに常に反映し続ける必要がある．しかし，キルコマンドの抽出には個々のマルウェアを手動で静的解析しており，専門的な知識と多くの時間がかかり，大量のマルウェアに対応できないため効率的な抽出手法を検討する必要がある．また，キルコマンドを持たないマルウェアに対しては駆除手法を適用できないため，新たな駆除手法を検討しシステムに実装する必要がある．

(2) 擬似 C&C サーバへの通信の転送

擬似 C&C サーバへの通信の転送は，事前に静的解析や動的解析などから得られた C&C サーバの情報（IP アドレスやドメイン名）を用いたブラックリスト形式を想定している．C&C サーバの情報については，IoTPOT で収集された通信データを用いることで，大量の IoT マルウェアの通信先を収集する等の解決策を検討している．しかし，C&C サーバの IP アドレスが変更された場合や，C&C サーバからのコマンドによって通信先を変更するようなマルウェアの場合は，ブラックリストに一致しないため転送が行われないという課題がある．

(3) 擬似 C&C サーバのインターネット上での運用

擬似 C&C サーバをインターネット上で運用すれば，IoT ゲートウェイ内に設置する場合と比較して，より広範囲な駆除が可能になると想定される．しかし，インターネット上で運用した場合，感染した IoT 端末の保有者の承諾無しに端末内のプロセスを終了させることになるため，法律面や倫理面での検討が必要になる．

本稿では，報告 [5] の IoT マルウェア駆除手法を検証するために構築した疑似 C&C サーバシステムについて報告する．加えて，“（1）擬似 C&C サーバのシステム化” について，以下の検討を行なった．1 つ目は，解析効率の向上のために，マルウェア内からキルコマンドの候補を自動で抽出する手法について検討した．2 つ目は，擬似 C&C サーバで対応できるマルウェアの種類を増やすために，キルコマンドによらない駆除手法について検討した．

本稿の構成を以下に示す．2 章では関連研究について述べる．3 章では提案手法について述べる．4 章では本研究における課題と検討について述べる．5 章ではまとめと今後の予定について述べる．

2. 関連研究

本章では，IoT マルウェア対策に関する研究とマルウェア対策に関する実例について述べる．

2.1 IoT マルウェア対策に関する研究

近年，IoT マルウェア対策に関する様々な研究 [8], [9], [10], [11] が行われている．

論文 [8] では，IoT マルウェアに感染した IoT 機器の通信に着目し，C&C サーバへ通信を開始した場合に，対象の IoT 機器に対して以下を行うことで IoT マルウェアの駆除を行なっている．

- IoT マルウェアに感染した IoT 機器へ Telnet コマンドでログインを行い reboot コマンドを実行する
- 機器の電源を物理的に落として再起動する
- 機器に備わっている機能で工場出荷前に復元する

論文 [9] では，多くの IoT 機器では動作するプロセスが一定である点に着目してプロセスの監視を行う手法を提案している．具体的には，ホワイトリスト方式でプロセスを監視するスクリプトを IoT 機器内に配置し，ホワイトリストに一致しないプロセスの起動を確認した際に該当プロセスを終了することで，マルウェア感染を防止している．

論文 [8], [9] はいずれもスクリプトの設置や機器の電源を物理的に落とすなど IoT 機器に対してのアプローチが取られている．本研究は，IoT 機器に手を加えることなく，ネットワーク経由で IoT マルウェアを駆除するという点で異なる．

論文 [10] では，IoT マルウェアによる DDoS 攻撃（Distributed denial of service attack）の実態を分析するために，ハニーポットで収集した IoT マルウェアをサンドボックス内で実行しその挙動を観測している．具体的には，挙動観測の際に解析環境上にダミー C&C サーバとダミー攻撃対象サーバを用意し攻撃を再現を行っている．攻撃の再現では，マルウェアから攻撃対象への通信をダミー攻撃対象サーバに転送を行い，逆にダミー攻撃対象サーバからの応答を攻撃対象からの応答としてマルウェア側に転送している．本研究においても，擬似 C&C サーバを用いているが，IoT マルウェアの挙動を観測する目的ではなく，IoT マルウェアに対してキルコマンドを送信する目的で用いている点で異なる．

論文 [11] では，インターネット上に存在する IoT マルウェアの C&C サーバを効率的に発見するシステムの提案を行っている．具体的には，マルウェアを動的解析し，C&C サーバとの通信時に送信するペイロードと，それに対するレスポンスから悪性サーバを検知するためのシグネチャを作成し，探索を行っている．本研究においても，C&C サーバとの通信の際に発生するペイロードを用いているが，C&C サーバを探索する目的ではなく，IoT マルウェアの通信を判別する目的で用いている点で異なる．

2.2 マルウェア対策に関する実例

マルウェアの対策に関する実例として，実例 [12], [13] がある．

擬似 C&C サーバを用いたマルウェア対策の実例として，バンキングマルウェアである VAWTRAK を無力化する目的で，警視庁が 2016 年に行ったテイクダウン（無力

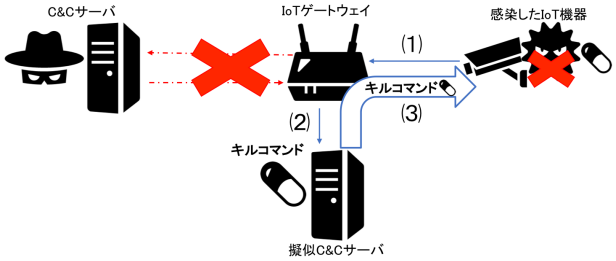

図 1　提案手法概要

化）作戦がある [12]．この作戦では，バンキングマルウェアに感染した PC に対して疑似 C&C サーバから何も行わない指令（無力化情報）を送信することで，マルウェアの攻撃活動を無力化する．また，テイクダウン作戦実行後に VAWTRAK による被害の減少が報告されている．本研究においても，擬似 C&C サーバを用いているが，マルウェアの無力化を行うだけでなく駆除をするという点で異なる．

IoT 機器に対するセキュリティ対策の実例として，総務省及び国立研究開発法人情報通信研究機構が 2019 年から行なっている取組がある [13]．この取組では，インターネット上の IoT 機器について，サイバー攻撃に悪用されるおそれのある機器を調査し，調査した IoT 機器の情報をインターネットプロバイダへ通知し，利用者に対して注意喚起を行っている．なお，この取組は平成 30 年 11 月 1 日に施行された「電気通信事業法及び国立研究開発法人情報通信研究機構法の一部を改正する法律」を元に実施されている．本研究で提案した擬似 C&C サーバをインターネット上で運用する場合，実例 [13] の様に法律面での対応が必要になると考える．

3. 提案手法

本章では，IoT マルウェア駆除手法の概要および，BOT 型の IoT マルウェアである tsunami の駆除手法，擬似 C&C サーバのシステム化について述べる

3.1 IoT マルウェア駆除手法の概要

本研究における提案手法は，「BOT 型の IoT マルウェアが C&C サーバと通信を行い特定の機能を実行する」ことに着目した駆除手法である．C&C サーバと IoT マルウェアの通信には，特定のパターンが存在する．このパターンを模擬した擬似 C&C サーバから IoT マルウェアに対してキルコマンドを送信し駆除を行う．なお，IoT マルウェアの通信を擬似 C&C サーバに転送する際は，C&C サーバのアドレスリストを用いて転送を行う．キルコマンドの送信手順を以下に示す（**図 1**）．

(1) IoT マルウェアに感染した機器が C&C サーバへ通信
(2) C&C サーバのアドレスリストを用いて擬似 C&C サーバへ通信を転送
(3) IoT マルウェアに感染した機器に対してキルコマンドを送信

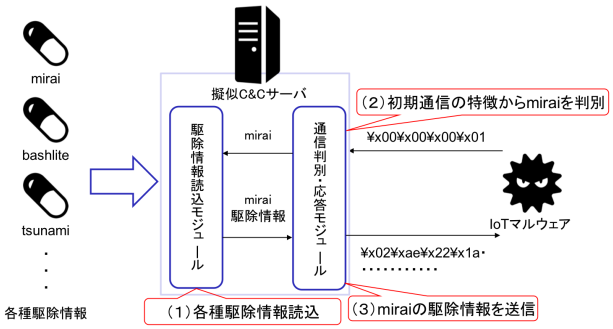

図 2　擬似 C&C サーバシステム

3.2 IoT マルウェア駆除手法適用範囲の拡大

IoT マルウェア駆除手法を用いた我々の調査 [5] において，駆除対象は bashlite と mirai であった．本稿では，これらに加えて IoT マルウェアである tsunami についても同様の調査を行い駆除が可能か実験を行なった．

tsunami には 1 種類のキルコマンドが存在する．tsunami は IRC サーバーに接続し，IRC プロトコル [14] を用いてコマンドの送受信を行う．公開されている tsunami のソースコードを静的解析して得られた，キルコマンドの送信に必要な項目とソースコード上の値（デフォルト値）を**表 1** に示す． IoTPOT から得られた tsunami に対して静的解析を行い，駆除に必要な情報を抽出した結果を**表 2** に示す．表 2 を元にして提案手法による駆除を行なったところ，5 検体中 2 検体を駆除することができた．表 2 の結果より，全ての検体でデフォルト値ではないチャンネル名が使われていた．IRC ユーザ名とキルコマンドについては，デフォルト値と同じ検体も存在した．

駆除できなかった検体に対して詳細に静的解析を行った所，キルコマンドは存在したが実行される関数が「Unable to comply」という文字列をサーバに返答するのみで，プロセスの終了は行わない実装となっていた．これは，攻撃者が意図的にキルコマンドの実装を削除したものと思われる．他の駆除できなかった検体では，IRC ユーザ名が「ADMIN USER」であり，ADMIN と USER の間にスペースが含まれていたため，tsunami 側でコマンドをパースする処理が失敗していた．IRC プロトコルにおいてユーザ名にスペースを含んではいけないという制約があり，これを想定した実装になっておらず，キルコマンドの実行が失敗したと考えられる．これらの駆除できない検体に対しては，キルコマンドが存在しない場合の駆除手法を適用する必要がある．

3.3 擬似 C&C サーバシステム

3.1 節で述べた提案手法のシステムを構築した（**図 2**）．システムの構成は，Windows7 に VirtualBox をインストールし，仮想環境上で擬似 C&C サーバとマルウェア実行環

表 1　tsunami のキルコマンド送信に必要な情報

項目名	デフォルト値	説明
IRC チャンネル名	#Channel	tsunami が接続する C&C サーバのチャンネル名
IRC ユーザ名	Fine	キルコマンドを実行するために必要なユーザ名
キルコマンド	Kkt9x4JApM0RuSqCLA	チャンネルに接続された tsunami のプロセスを終了するために必要な文字列

表 2　tsunami の駆除に必要な情報の静的解析結果

ハッシュ値	チャンネル名	IRC ユーザ名	キルコマンド
cbcba43d3***	##KTN	無し	KILL
799a55a73***	##Nix	Fine	Kkt9x4JApM0RuSqCLA
a6be88432***	##KTN	ADMIN USER	+botkill
5837488bc***	#AS	無し	KILL
cd1576680***	#death	Fine	Kkt9x4JApM0RuSqCLA

境を用意した．擬似 C&C サーバとマルウェア実行環境は Ubuntu18.04 上で動作する．擬似 C&C サーバは Python スクリプトで実装した．マルウェア実行環境では QEMU によって MIPS アーキテクチャをエミュレートした．擬似 C&C サーバは通信判別・応答モジュールと，駆除情報読み込みモジュールの 2 つからなる．各モジュールの詳細については，次項以降に述べる．擬似 C&C サーバへの通信の転送については，事前に収集した C&C サーバの情報を用いて iptables 等で転送を行う．この通信転送機能については，現在構築中でありまだ実装していない．なお，現在のシステムではマルウェア実行環境から発生する全ての通信を iptable を用いて擬似 C&C サーバへ転送している．本システムにおいて，mirai，bashlite，tsunami を各数検体を用いて実験を行った結果，マルウェアの種類を自動判別し，適切な駆除情報を用いてマルウェアを駆除できることを確認している．

3.3.1　通信判別・応答モジュール

通信判別・応答モジュールは，疑似 C&C サーバへ転送された初期通信を元にマルウェアの判別を行い，駆除情報の送信を行う．通信判別モジュールの動作手順を説明する．まず初めに，初期通信の内容からマルウェアの判別を行う．次に，3.3.2 項で述べる駆除情報読み込みモジュールに判別したマルウェア名を渡す．最後に，駆除情報読み込みモジュールから受け取った駆除情報リストを用いて順番にマルウェアへ応答し，マルウェアからの応答が確認されなくなった場合に駆除したとみなす．

通信判別モジュールでは，初期通信の内容からマルウェアの判別を行うため，マルウェア毎の初期通信の情報が必要になる．我々の研究 [5] において調査した mirai，bashlite における初期通信例を**表 3**，**表 4** に示す．今回新たに調査を行なった tsunami の初期通信例について**表 5** に示す．なお，初期通信の内容は動的解析および静的解析によって得られたものである．

mirai（表 3）の通信順序を以下に示す．

- 通信順序 1
  mirai であることを示すバイト列を送信する．
- 通信順序 2
  送信される BOT 固有のバイト列の長さを送信する．
- 通信順序 3
  一定間隔でパケットを送受信し生存確認を行う．

この様に，通信順序 1〜2 で，C&C サーバに対して BOT の登録が行われると考えられる．そして，通信順序 3 以降 C&C サーバからコマンドを待ち受ける状態となる．

表 3　mirai 初期通信例順序

通信順序	通信内容
1	0x00 0x00 0x00 0x01
2	0x00
3	0x00 0x00

表 4　bashlite 初期通信例一覧

通信内容
BUILD MIPS
BUILD MIPSEL
BUILD X86
BUILD ARM
BUILD PPC
BUILD UNKNOWN

bashlite（表 4）は C&C サーバに対して自身のビルドバージョンを示す文字列を送信し，以降は C&C サーバからコマンドを待ち受ける状態となる．ビルドバージョンの種類は MIPS[15] や ARM[16] 等があり，様々なアーキテクチャの IoT 機器で動作させることを想定した実装になっている．

tsunami（表 5）の通信順序を以下に示す．

- 通信順序 1
  C&C サーバへの接続処理と思われる内容を送信
- 通信順序 2
  ユーザーモードの設定と思われる内容を送信
- 通信順序 3
  IRC チャンネルへ接続

この様に，通信順序 1〜2 で IRC チャンネルに接続する前の準備を行うと考えられる．その後，通信順序 3 で IRC

表 5　tsunami 初期通信例順序

通信順序	通信内容
1	PASS NICK [STD]ROHT USER Remote localhost localhost :Remote IRC BOT
2	MODE [STD]ROHT +pixB
3	JOIN #death : WHO [STD]ROHT

チャンネルに接続を行い，以降は C&C サーバからコマンドを待ち受ける状態となる．

3.3.2　駆除情報読み込みモジュール

駆除情報読み込みモジュールでは，駆除情報の管理と通信モジュールへの駆除情報の受け渡しを行う．駆除情報読み込みモジュールの動作手順について説明する．まず初めに，マルウェア種類別の駆除情報を読み込む．駆除情報は管理の容易さの観点から擬似 C&C サーバとは別に保持する．次に，通信判別・応答モジュールで判別したマルウェア名を受け取り，合致したマルウェアの駆除情報のリストを返す．

現状は，mirai，bashlite，tsunami の一部に対応しているが，新種の IoT マルウェアや既存の IoT マルウェアの亜種に対応するために，駆除情報の恒常的な更新が必要である．

4.　本研究における課題と検討

本研究では，1 章で述べた通り 3 つの課題が存在している．それらの課題の中でも，“（1）擬似 C&C サーバのシステム化” について運用を行う上での 2 つの検討を行なった．1 つ目は，キルコマンドは手動による静的解析で抽出しており，多くの時間がかかるという課題である．2 つ目は，キルコマンドを持たないマルウェアに対しては駆除手法を適用できないという課題である．そこで，本章ではこれらの課題を解決する方法として，キルコマンド候補の自動抽出手法とキルコマンドによらない駆除手法について検討する．

4.1　キルコマンド候補の自動抽出手法

キルコマンド抽出を効率化するために，マルウェア内からキルコマンドの候補を自動で抽出する手法について検討した．

キルコマンド候補の自動抽出手法の概要を図 3 に示す．図 3 より，マルウェアの判別（1）では動的解析を行いサンドボックス環境で検体を実行し，C&C サーバへの通信を発生させることで，3.3.1 項で述べたような特徴的な通信パターンと比較し，マルウェアの判別を行う．

マルウェアの情報抽出（2）では，判別したマルウェアに対して適切な方法で静的解析を行い，バイナリデータからキルコマンドの候補を抽出する．例えば，キルコマンド文字列付近には，exit 関数や kill 関数等が存在していると考えられるため，これらの関数の呼び出し箇所をバイナリ

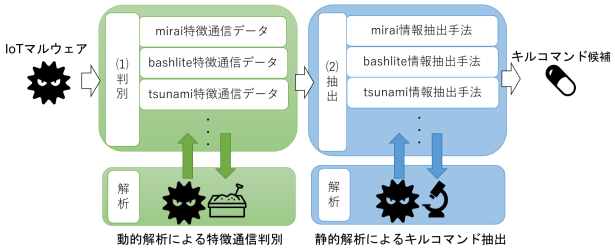

図 3　キルコマンド候補の自動抽出

```
if(!strcmp(argv[0], "LOLNOGTFO"))
{
        exit(0);
}
```

図 4　bashlite ソースコード（一部抜粋）

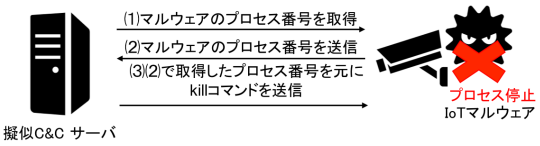

図 5　キルコマンドによらない駆除手法の例

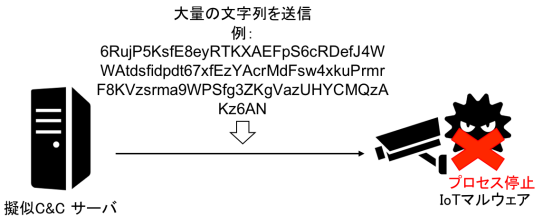

図 6　マルウェアの脆弱性を突いた駆除手法の例

上から探索し，周辺領域を含めて抽出・デコンパイルすることでキルコマンドを探索する方法を検討中である．我々は，キルコマンド候補の自動抽出の調査として公開されている bashlite のソースコードを静的解析した．図 4 より，キルコマンド文字列である “LOLNOGTFO” と exit 関数が近傍に存在していることを確認している．他の手法として，strings コマンドで文字列を抽出することも考えられるが，これはコマンドを示す文字列以外のノイズが多く含まれることや，コマンドがバイナリ文字列であった場合に抽出が行えないなどの課題がある．

4.2　キルコマンドによらない駆除手法

キルコマンド等を持たないマルウェアに対して，駆除手法を適用するためにキルコマンドによらない駆除手法につ

いて検討した．

4.2.1 shell を操作するマルウェアに対する駆除手法

BOT 型の IoT マルウェアの多くは，感染した IoT 機器内でコマンドを実行する機能（shell 操作機能）を持つものが一般的だと考えられる．この機能を用いてマルウェア自身のプロセスを停止するコマンド (kill コマンド) を実行することで駆除を試みる．

shell 操作機能を用いたプロセス停止手順例を以下に示す（**図 5** 参照）．

(1) マルウェアのプロセス（自分自身のプロセス）を取得するコマンドを送信
(2) マルウェアから送信されたプロセス番号を擬似 C&C サーバで保持
(3) （2）で保持したプロセス番号を元に kill コマンドを送信

shell 操作機能は図 5 に挙げた以外にも IoT 機器に存在する shell コマンドを実行できる．しかし，IoT 機器毎に実行できる shell コマンドは異なると考えられるため，プロセスを停止するために必要な shell コマンドのパターンを調査する．また，shell 操作機能実行に必要なコマンドのフォーマットも異なるためマルウェアの静的解析を行うことでフォーマットを調査する．

4.2.2 マルウェアの脆弱性を突いた駆除手法

BOT 型の IoT マルウェアには，コマンドの送受信を行うためのインターフェースが存在する．このインターフェースに対して，バッファオーバーフローなどの脆弱性を突くことで，プロセスの終了を試みる．

例えば，コマンドを受け付けるインターフェースに文字数制限がないと仮定し，大量の文字列を送信することでセグメンテーションフォルトを発生させ，プロセスの終了を促すことが考えられる（**図 6** 参照）．しかし，IoT 機器で動作する他のプロセスに影響がないかを検討する必要がある．この様に，安全性などの点から脆弱性を突く他の方法については更なる検討が必要である．

5. まとめと今後の予定

本章では，まとめと今後の予定について述べる．

- 擬似 C&C サーバシステム
  擬似 C&C サーバシステムを構築した．構築したシステムにおいて mirai，bashlite，tsunami の各数検体を駆除できることを確認した．今後は更に検体数を増やして実験を行いシステムの有効性を評価する．
- tsunami の駆除手法
  tsunami の静的解析を行い，駆除手法の調査を行った．調査の結果，mirai や bashlite と同様にキルコマンドによる駆除が可能であることを明らかにした．
- キルコマンド候補の自動抽出手法
  キルコマンド候補の自動抽出手法について検討した．今後は，既にキルコマンドの文字列がわかっている検体をデコンパイルし，デコンパイルした結果から exit などの終了関数の近傍に存在する文字列を抽出後，どの程度キルコマンドの文字列と一致するか検証する．次に，未知の検体に対しても同様の手法でキルコマンドの抽出が可能か検証する．
- shell を操作するマルウェアに対する駆除手法
  shell を操作するマルウェアに対する駆除手法について検討した．今後は，IoT 機器で用いられるプロセスを停止する，あるいはそれに該当する shell コマンドを調査する．次に，shell を操作するマルウェアの静的解析を行い，shell 操作機能を実行するために必要なコマンドのフォーマットを調査する．最後に，調査した結果を元にした駆除情報を検体に対して適用し駆除が可能か実験する．なお，コマンドのフォーマット抽出は，キルコマンド候補の自動抽出と同様に自動化を検討する必要がある．
- マルウェアの脆弱性を突いた駆除手法
  マルウェアの脆弱性を突いた駆除手法について検討した．今後は，マルウェアの静的解析を行い脆弱性を調査し，脆弱性が発見できた場合は Exploit コードの作成を試みる．そして，作成できた Exploit コードを，同じソースコードから作成されたと考えられるキルコマンドが異なる検体，あるいはキルコマンドが存在しない検体に対して適用し駆除が可能か実験する．

上記の他にも，1 章で述べた “（2）擬似 C&C サーバへの通信の転送” と “（3）擬似 C&C サーバのインターネット上での運用” について検討及び実験を行う予定である．

謝辞

本研究は，国立研究開発法人情報通信研究機構の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」の成果の一部です．

本研究を進めるにあたり，横浜国立大学の吉岡克成准教授，玉井達也氏から有益な助言とデータ提供の協力を頂きました．深く感謝致します．

参考文献

[1] IoT 開発におけるセキュリティ設計の手引き，https://www.ipa.go.jp/files/000052459.pdf
[2] IoT デバイスを狙うマルウェア「Mirai」とは何か――その正体と対策，https://techfactory.itmedia.co.jp/tf/articles/1704/13/news010.html
[3] threatpost，https://threatpost.com/bashlite-family-of-malware-infects-1-million-iot-devices/120230/
[4] New IoT/Linux Malware Targets DVRs, Forms Botnethttps://unit42.paloaltonetworks.com/unit42-new-iotlinux-malware-targets-dvrs-forms-botnet/
[5] 三須 剛史，桃井達明，“擬似 C&C サーバを用いた IoT マ

ルウェア駆除手法の提案”，暗号と情報セキュリティシンポジウム 2019，セッション 3E2-2，2019.
[6] GitHub,https://github.com/
[7] Y. M. P. Pa et al., “IoTPOT: A Novel Honeypot for Revealing Current IoT Threats”, J. Information Processing, vol. 24, no. 3, pp. 522-533, 2016.
[8] 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
[9] Chun-Jung Wu et al., “IoTProtect: Highly Deployable Whitelist-based Protection for Low-cost Internet-of-Things Devices”, J. Information Processing, vol. 26, pp. 662-672, 2018.
[10] 鉄 穎, 楊 笛, 保泉 拓哉, 中山 颯, 吉岡 克成, 松本 勉, “IoT マルウェアによる DDoS 攻撃の動的解析による観測と分析，” 情報処理学会論文誌，vol.59，no.5，pp. 1321-1333，2018
[11] 篠宮一真，山村翔，荒木翔平，張一凡，胡博，神谷和憲，谷川真樹，浜田泰幸，高橋健司，“IoT マルウェアと通信可能な悪性サーバの探索”，コンピューターセキュリティシンポジウム 2018，セッション 3B1-4，2018
[12] 高田一樹，“「ネットバンキングウイルス無力化作戦」の裏側と高度化する金融マルウェア”，CODE BLUE 2016, https://www.slideshare.net/codeblue_jp/cb16-takada-ja
[13] IoT 機器調査及び利用者への注意喚起の取組「NOTICE」の実施, http://www.soumu.go.jp/menu_news/s-news/01cyber01_02000001_00011.html
[14] Internet Relay Chat Protocol，https://tools.ietf.org/html/rfc1459
[15] MIPS Processors，https://www.mips.com/
[16] ARM Processors，https://www.arm.com/