Raspberry Piを用いたIoTハニーポットの開発

坂川 巧将[1,a)]　廣友 雅徳[2,b)]　福田 洋治[3]　毛利 公美[4]　白石 善明[5]

概要：本研究では，Raspberry Pi を用いた IoT ハニーポットの開発を行う．本研究の目的は，IoT 機器に対するサイバー攻撃を観測，分析することである．このシステムでは ssh と telnet を稼働させ，多くの攻撃を調査する．入手したマルウェアの分析を行い，IoT 機器への攻撃の実情を調べる．また，攻撃者には通信を行わせないことで，侵害が広まることを防ぐ．

キーワード：ハニーポット，IoT，不正アクセス，サイバー攻撃

IoT honeypot Using Raspberry Pi

Kosuke Sakagawa[1,a)]　Masanori Hirotomo[2,b)]　Youji Fukuta[3]　Masami Mohri[4]
Yoshiaki Shiraishi[5]

Abstract: In this research, we improve the IoT honeypot with Raspberry Pi. The purpose of this study was observe and analyze cyber attacks on IoT devices. This system runs ssh and telnet and investigates many attacks. The obtained malware is analyzed to determine the actual status of attacks on IoT devices. And, banning the connect by attacker for do not spread infringement.

Keywords: honeypot, IoT

1. はじめに

近年，IoT 対応製品が普及し，その数は 2020 年に 400 億台を超すといわれている．IoT によって，生活から事業までの幅広い情報が収集され，利用されている [1]．これに伴い，Mirai と呼ばれるマルウェアを中心とした脅威も増加している [2]．Mirai は IoT 機器を標的としたウイルスで，ソースコードが公開されているため様々な亜種が存在している．また，Mirai は侵入した機器の CPU アーキテクチャを確認し，それに応じた攻撃バイナリをダウンロードする．2016 年には Mirai に感染したボットネットによる DDoS 攻撃が行うわれ, この攻撃は最大 620Gbps にも達した．

2018 年 9 月には Chalubo とよばれるマルウェアが活発になった [3]．このマルウェアの目的は，IoT 機器を含む ssh サーバに対して不正アクセスを行うことである．最終的に，DDoS 攻撃用のマルウェアをダウンロードし，DDoS 攻撃を実行する．Chalubo は広範囲の IP アドレスをスキャンし，ポート 22 番で ssh を稼働しているデバイスを探索する．その後，一般的なデフォルト設定を使った辞書攻撃か脆弱なパスワードに対する総当り攻撃などで認証情報を入手する．Chalubo には Mirai のコードが使われていることから主な標的は IoT 機器と考えられる．

つまり IoT 機器の telnet や ssh への攻撃はこれからも続くと考えられる．そのため企業や個人は IoT 機器への攻撃を監視する必要があるが，中小企業などは情報セキュリティに使える予算が少ないという問題点がある [4]．そこで本研究では IoT 機器のための安価なハニーポットを提案す

[1] 佐賀大学大学院理工学研究科
Graduate of Science and Engineering, Saga University
[2] 佐賀大学理工学部
Faculty of Science and Engineering, Saga University
[3] 近畿大学理工学部
Faculty of Science and Engineering, Kindai University
[4] 岐阜大学工学部
Faculty of Engineering, Gifu University
[5] 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
[a)] sakagawk@ma.is.saga-u.ac.jp
[b)] hirotomo@cc.saga-u.ac.jp

る．本システムは高対話・サーバ型ハニーポットであり，ハードウェアに Raspberry Pi を用いる．これにより安価で手軽にハニーポットを構築できる．

2. 先行研究

小寺らは，QEMU を用い 6 つの CPU をエミュレーションし telnet ベースの IoT 向けハニーポットを開発している [5].

Pa らは，フロントエンドで telnet を模しバックエンドで IoTBOX を動作させたハニーポットを開発している [6]. 81 日間の運用で IoT 機器への攻撃を観測し，80 万の IP アドレスから 48 万件のマルウェアのダウンロードがあったことを報告している．これらのマルウェアの中には既存の telnet ハニーポットでは補足できないものもあった．

鈴木らは，より多くの攻撃を観測するために文献 [6] の機能拡張を行った [7]．DVR やルータ，IP カメラのサービスを模擬することでこれらに対する攻撃を確認した．プロキシを用い，多数の IP アドレスを割り当てれるように改良した．

坂野らは，アノマリ手法を用い IoT 機器の通信からマルウェアの感染を検知する方法を提案している [8]．この研究では事前調査として IoT 機器を模したハニーポットで telnet と ssh への攻撃を観測した．約 2 週間の運用で telnet には 2.7 万回，ssh には 4.6 万回のログインが行われた．

3. IoT ハニーポットの構成

本章では本研究で提案するハニーポットの構成と機能について述べる．

3.1 ハードウェア

ハードウェアには Raspberry Pi を用いる．理由は入手しやすく，安価であるためである．また，環境の複製が容易であるためクリーンな環境を保つことができ，かつ短時間で複製することができるためである．Raspberry Pi の CPU アーキテクチャは ARM であるため，Mirai と Mirai から派生したマルウェアの標的になりやすく，IoT ハニーポットのハードウェアの条件として適している．

3.2 システム構成

提案システムは攻撃を待ち受けるハニーポット，攻撃の記録を保存する管理サーバ，マルウェアを入手するダウンロードサーバの 3 つから成る．ダウンロードサーバは管理サーバ上で動作する．提案システムのネットワークの構成を図 1 に示す．

ハニーポットの OS には Raspberry Pi OS (32-bit) with lite を用いる．

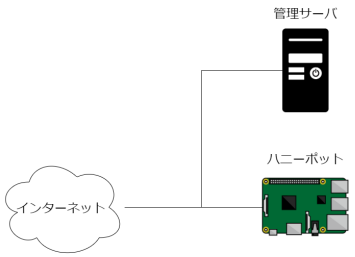

図 1 ネットワーク構成

3.3 機能

本節では本研究のハニーポットの機能について述べる．

3.3.1 動作するプロトコル

Mirai は telnet に対し攻撃を行うため，ハニーポットでは telnetd を用いて telent(23/tcp) を受け付ける．

また ssh への攻撃も予想されるため，openssh-server を用いて ssh(22/tcp) を受け付ける．デバイス検索エンジンである Shodan[9] で，ポート 22 番で稼働している ssh の数を調査した結果，ssh 全体の数は 20,877,142 個，ポート 22 番で稼働している ssh の数は 18,517,642 個だった．よって IoT 機器の ssh は 9 割以上がポート 22 番で稼働しているため，本研究のハニーポットもポート 22 番で ssh を稼働させる．

3.3.2 ログイン時のメッセージ

ssh を用いて Raspberry Pi OS へログインを行うと OS のバージョン情報が表示される．このメッセージの中に raspberrypi という単語あるため，攻撃者に Raspberry Pi であることが気づかれてしまう．そこでメッセージを変更し，攻撃者に与える情報を減らす．まず OS のバージョン情報を隠すために/etc/pam.d/sshd を編集する．また最終ログインの情報を隠すために，ユーザのホームディレクトリに.hushlogin を作成する．警告メッセージを隠すために，/etc/profile.d/sshpwd.sh の標準出力を空にする．

telent を用いて Raspberry Pi OS へログインを行うと以下のメッセージが表示される．こちらもメッセージの中に raspberrypi という単語あるため，攻撃者に Raspberry Pi であることが気づかれてしまう．

```
Trying アドレスIP...
Connected to アドレスIP.
Escape character is '^]'.
Raspbian GNU/Linux 10
raspberrypi login:
```

そこで攻撃者に与える情報を減らすために/etc/inetd.conf の telnetd に h オプションを追加し、ログインバナーを隠す．また hostnamectl コマンドで Hostname を ubuntu に変更し以下のようにメッセージを変更する．

```
ubuntu login:
```

表 1　ユーザ名とパスワードの組み合わせ

ユーザ名	パスワード
admin	admin
support	support
user	user
Administrator	admin
service	service
supervisor	supervisor
guest	guest
admin1	password
666666	666666
888888	888888
ubnt	ubnt
tech	tech
mother	fucker

3.3.3　ユーザ名とパスワード

表 1 は Mirai のソースコード [10] に記載されたユーザ名とパスワードの組み合わせの一部である．この組み合わせのアカウント情報をハニーポットに追加した．これにより攻撃者がハニーポットにログインできる可能性が増え，観測できる攻撃が増えると考えられる．

また管理者である root を一般ユーザに変更し，root でログインしても管理者権限でコマンドを実行できないようにする．これにより root への攻撃を安全に観測できる．root で vipw コマンドでユーザ情報ファイル/etc/passwd を編集し，root を別のユーザ名に変更する．次に vipw -s コマンドでパスワードファイル/etc/shadow を編集し，root のユーザ名を先程と同じものに変更する．passwd コマンドで，パスワード情報を更新する．vigr コマンドでグループ情報ファイル/etc/sgroup を編集し，root のグループを一般のグループに変更する．次に vipw -s コマンドでパスワードファイル/etc/group を編集し，root のグループを先程と同じものに変更する．これにより，ユーザ名が root である一般ユーザが作成できる．

3.3.4　コマンドの記録

Snoopy Logger[11] を用い，攻撃者がログイン後に実行するコマンドを記録する．このログにより攻撃者が IoT 機器を乗っ取る過程とその後の動きを知ることができる．ログは管理サーバに FTP で送信し，保存する．

3.3.5　マルウェアのダウンロード

ログを毎分監視し，ftp，curl，wget のいずれかのコマンドが実行されていれば攻撃者がマルウェアのダウンロードを試みたとし，ダウンロードサーバでマルウェアのダウンロードを行う．

3.4　活動の制限

本節では攻撃者にハニーポットを乗っ取られる可能性を減らすための制限について述べる．

3.4.1　ログインの制限

管理者へのリモートログインを禁止し，攻撃者が一般ユーザでのみ活動するように制限する．またハニーポットの設定が変更されることを防ぐため，一般ユーザは sudo コマンドを利用できないようにする．通常 telnet での管理者へのリモートログインは禁止されている．ssh は/etc/ssh/sshd_config の PermitRootLogin の項目を NO にし，管理者へのリモートログインを禁止する．

3.4.2　通信の制限

iptables を用い，管理サーバ以外の通信を禁止する．ログインがあった場合はログインから 5 分後に通信を遮断し，5 分後に再び通信を受け付ける．

3.4.3　バイナリの実行制限

攻撃者は/tmp や/home にマルウェアをダウンロードする傾向がある．そのため/tmp と/home に置かれた実行ファイルを実行不可能にする．カーネルのパーティションとは別のパーティションを用意し，noexec オプションを付けてマウントする．noexec オプションを付けてマウントしたパーティションはバイナリの実行ができなくなる．しかし noexec オプションを使っても ld-linux.so コマンドを使うとバイナリの実行ができる問題がある [12]．

3.4.4　一部のコマンドの実行制限

一般ユーザのシェルを rbash にする．シェルが rbash であると bash の組み込みコマンドは実行できるが，cd コマンドを用いたディレクトリの移動や/を含むコマンドの実行ができなくなる．これにより，攻撃者の不正な操作を防げる．

4.　まとめと今後の課題

本研究では，IoT 機器向けの安価なハニーポットの開発を行うにあたって，IoT 機器に対するセキュリティ侵害事例をもとにシステムの機能と構築方法を示した．今後の課題として，提案ハニーポットを実装し実験を行うことで有効性を示すことが挙げられる．またハニーポットの複製が手動で行うことになるため，ネットワークを介した自動インストールの機能を追加し作業者の負担を減らす必要がある．

参考文献

[1] ”総務省　平成 30 年版　情報通信白書　IoT デバイスの急速な普及”, 入手先 ⟨http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd111200.html⟩(2020.08.20).

[2] 独立行政法人情報処理推進機構. 情報セキュリティ白書.2019 年版，　2019. 入手先 ⟨https://www.ipa.go.jp/files/000079041.pdf⟩(2020.08.20)

[3] ”安全性の低い SSH サーバーがボットネットの標的に”.naked security. 入手先 ⟨https://nakedsecurity.sophos.com/ja/2018/10/24/poorly-secured-ssh-servers-targeted-by-chalubo-

botnet/⟩(2020.08.20).
[4] IPA. 入手先 ⟨https://www.ipa.go.jp/files/000058502.pdf⟩ (2020.08.20)
[5] Kodera, Tateki and Izumi, Takashi. ”IoT デバイス用ハニーポットの構築”
[6] Pa, Yin Minn Pa and Suzuki, Shogo and Yoshioka, Katsunari and Matsumoto, Tsutomu and Kasama, Takahiro and Rossow, Christian. ”IoTPOT: A Novel Honeypot for Revealing Current IoT Threats”.”Journal of Information Processing”.2016.vol. 24, no .3.p. 522–533.(2020.08.20).
[7] 将吾, 鈴木 and イン ミン, パパ and 優太, 江澤 and 穎, 鉄 and 楓, 中山 and 克成, 吉岡 and 勉, 松本. ”組込み機器への攻撃を観測するハニーポット IoTPOT の機能拡張”.”研究報告セキュリティ心理学とトラスト（SPT）”.2016.vol. 2016-SPT-17, no .1.p.1–6.(2020.08.20).
[8] 坂野, 加奈 and 上原, 哲太郎. ”アノマリ手法を用いた IoT 機器のマルウェア感染検”.2018. no .3.(2020.08.20).
[9] ”Shodan”，入手先 ⟨https://www.shodan.io/⟩，(2020.08.20).
[10] Mirai-Source-Code/scanner.c，入手先 ⟨https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/bot/scanner.c⟩，(2020.08.20).
[11] a2o/snoopy: Log every executed command to syslog (a.k.a. Snoopy Logger)，入手先 ⟨https://github.com/a2o/snoopy⟩，(2020.08.20).
[12] ”4.10. パーティションを正しくマウントする”.debian，入手先 ⟨https://www.debian.org/doc/manuals/securing-debian-manual/ch04s10.ja.html⟩，(2020.08.20).