IoT デバイス管理システムによる 家庭 LAN 内の IoT デバイス脆弱性調査

舟根 優作[†1]　永見 健一[†1]　遠藤 貴裕[†1]　時﨑 涼輔[†1]

概要：近年，IoT デバイスを悪用したサイバー攻撃が増加している．総務省及び国立研究開発法人情報通信研究機構（NICT）では，WAN 側からアクセスできる IoT デバイスに対して脆弱性を調査し，注意喚起を行う取組「NOTICE」を実施している．WAN 側から不正にアクセスされた場合，それが踏み台になってマルウェアが家庭内に広がる可能性がある．そのため，インターネットから直接接続できない LAN 内のデバイスの安全性も考慮する必要がある．この問題に対して我々は，独自に開発した脆弱性診断を行う IoT デバイス管理システムを用いて，LAN 内のデバイスに対して脆弱性を調査した．調査内容は，ポートスキャンによるポートの空き状況，http と telnet に対して容易に推測されるパスワードを入力した際のログインの可否である．これに加え，デバイスの種別などについても統計情報を取得している．本調査により，http 認証が設定されているデバイスに関して約 18.8%，telnet が空いているデバイスに関して約 13.6%の割合で脆弱なパスワードが設定されていることがわかった．

キーワード：モノのインターネット（IoT），セキュリティ，デバイス管理，脆弱性

Investigation of IoT Device Vulnerability in Home LAN by IoT Device Management System

Yusaku Funane[†1]　Kenichi Nagami[†1]　Takahiro Endo[†1]　Ryosuke Tokizaki[†1]

Abstract:
In recent years, cyber-attacks that exploit IoT devices are increasing. The Ministry of Internal Affairs and Communications and National Institute of Information and Communications Technology (NICT) investigated vulnerabilities on IoT devices and implement initiatives to alert them. It's called "notice". If it's accessed illegally from the WAN side, it may become a springboard and spread malware in the home. Therefore, we also need to consider the safety of devices in the LAN. To this issue, we investigated vulnerabilities of devices using the IoT device management system. The contents of the investigation are port scans and logins when entering easily guessed passwords for http and telnet. In addition, we collect statistical information on port availability and device type. As a result, we found that weak passwords were set at about 18.8% for devices with http authentication and about 13.6% for devices with open telnet.

Keywords: IoT （Internet of Things）, Security, Device Management, vulnerability

1. はじめに

1.1 IoT デバイスに対する脅威と管理の現状

近年，Web カメラや AI スピーカー，テレビなど，様々なデバイスがインターネットに接続されるようになってきている．2020 年には，インターネットに接続される IoT(Internet of Things)デバイスが 400 億台を超えると予測されている[1]．こうした IoT デバイスの増加に伴い，IoT デバイスを対象としたサイバー攻撃が増加している．2016 年，マルウェア“Mirai”は Telnet サービスに脆弱なパスワードが設定された IoT デバイスを踏み台として悪用し，非常に大規模な DDoS（Distributed Denial-of-Service）攻撃を引き起こした. 最近では,“Mirai”の亜種が発見されている．“Mirai”の亜種には法人向けのワイヤレスプレゼンテーションシステム“WePresent WiPG-1000”と“LG Supersign TV”の脆弱性を攻撃するコードが組み込まれていた. Mirai は標的を企業へとシフトしながらも未だに猛威を振るっている[2]. また, ランサムウェア“BitPaymer”や“Dharma”, “Ryuk”のようにラテラルムーブメントにより内部拡散する事例も報告されており[3]，企業内ネットワークや家庭内ネットワークのような，閉鎖的なネットワークであっても安心はできない．

IoT デバイスに限らず，サイバー攻撃に対する防御策としては，適切なパスワードの設定やアップデートの適用など，利用者自身が適切なセキュリティ対策を講じることが極めて重要である．しかしながら，適切な認証設定が行われていないような，いわゆる“管理者不在”の野良 IoT デバイスの存在が指摘されており[4]，IoT デバイスが正しく管理されているとは言い難い．

†1 株式会社インテック
INTEC Inc.

1.2 本研究の貢献

IoT 全体のセキュリティ向上のため，我々はこれまで IoT システム側のセキュリティ対策技術[5]を提案している．また，これだけでなく，ユーザ側の適切な IoT デバイス管理を支援するため，IoT システムのユーザの大部分を占めると思われる IT 技術に精通していないユーザを対象として，簡便な IoT デバイス管理システムを提案した[6]．我々はこの IoT デバイス管理システムを用いて，ユーザからデータを収集した．本稿では収集したデータについて，調査・分析した結果を報告する．

1.3 関連する調査

1.1 節で述べたように，IoT デバイスを悪用した大規模なサイバー攻撃（DDoS 攻撃）により，インターネットに障害が生じるなどの深刻な被害が発生している．この現状を踏まえ，総務省及び国立研究開発法人情報通信研究機構（NICT）では，2019 年 2 月，WAN 側からアクセスできる IoT デバイスに対して調査を実施し，利用者への注意喚起を行う取組“NOTICE（National Operation Towards IoT Clean Environment）”を実施している[7]．

“NOTICE”では，IoT デバイスに対して，容易に推測されるパスワードを入力することにより，サイバー攻撃に悪用されるおそれのある機器を特定し，当該機器の情報をインターネットプロバイダへ通知する．これを受けて，インターネットプロバイダは，当該機器の利用者を特定し，注意喚起を実施している．

調査の結果，約 9,000 万の IP アドレスのうち，ID・パスワードが入力可能であったものは約 3,1000～約 4,2000 件であった．このうち，ID・パスワードによりログインでき，注意喚起の対象となった IP アドレスは延べ 147 件である[8]．

一方で，IoT デバイスが WAN 側から不正にアクセスされた場合，家庭内のデバイスが感染し，それが踏み台になって家庭内に広がる可能性がある．そのため，インターネットから直接接続できない LAN 内の機器の安全性も考慮する必要がある．

2. IoT デバイス管理システム

我々は，企業内ネットワークや家庭内ネットワークに接続された IoT デバイスが適切に管理されていない現状を踏まえ，IT 技術に精通していない一般的なユーザでも容易かつ継続的に扱える IoT デバイス管理システムを提案した．この章では提案した IoT デバイス管理システムについて説明する．

2.1 スマートフォン用のアプリケーションとして提供

現在，コンシューマ向け情報通信デバイスの主役が，パーソナルコンピュータからスマートフォンにシフトしている[9]．スマートフォンは，ほぼ一人が一台を持ち，何時でも利用できる利便性と手軽さから，様々な新しいサービスやアプリケーションのプラットフォームとして活用されている．

また，スマートフォンでは，アプリケーションストアによるアプリケーション配信プラットフォームが発達しており，ユーザは手軽に利用したいアプリケーションを導入できる．アプリケーションのバージョンアップも自動的に行われることから，本システムに最新のセキュリティ脅威動向を反映した追加機能を実装した際にも，ユーザ側は特に意識することなく自動的にアプリケーションが更新されて利用できるようになる．

こうした理由から，ユーザが容易に無理なく継続的に利用するために，IoT デバイス管理システムはスマートフォンアプリケーション“MIRAI_DEFENDER”として提供している．“MIRAI_DEFENDER”はスマートフォンのアプリケーションストアにて無償で公開[10][11]しており，誰でも自由に利用することができる．

2.2 IoT デバイス管理アプリケーション

IoT デバイス管理アプリケーションの全体構成および処理の流れの一例を図 1 に示す．本システムは，ユーザのスマートフォン上で稼動する“IoT デバイス管理アプリケーション MIRAI_DEFENDER（以下，MIRAI_DEFENDER）”，そしてクラウド上に配備された“IoT デバイス反応情報 DB”および“脆弱性情報・公開情報 DB”を主な要素として構成される．

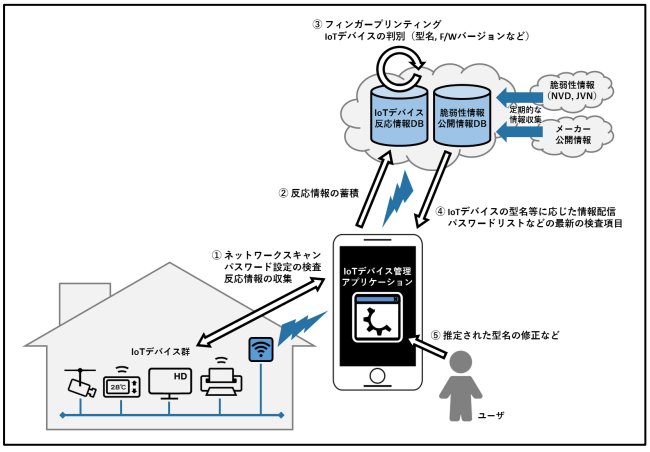


図 1 全体構成と処理の流れの一例

“MIRAI_DEFENDER”では，Ping スキャンと ARP テーブルの確認によるネットワークスキャンを行って機器を検知する．また，IoT デバイスで待ち受けている各種サービスに関するバナー情報などの反応情報をもとに機器を判別

する．

“MIRAI_DEFENDER”で収集された IoT デバイスの反応情報は，ユーザの個人情報を含まない IoT デバイス固有の情報のみをクラウド上の IoT デバイス反応情報 DB に蓄積する．この反応情報から，IoT デバイスの型名やファームウェアバージョンを特定するためのフィンガープリンティングを実施する．この結果から，メーカーが設定した初期パスワードなどの脆弱なパスワードが設定されていないかを検査する脆弱性診断や，ソフトウェアアップデートが実施されているかについての検査を実施する．

また，反応情報を学習データとして AI によるデバイスタイプの判別も行う．この結果は“MIRAI_DEFENDER”でアイコンとして表示している．“MIRAI_DEFENDER”では反応情報やデバイスタイプ判別の結果から，図 2 のように IoT デバイス一覧画面を作成する．

一覧画面の上部には，接続されている LAN の SSID（Service Set Identifier）や接続機器数を表示している．デバイスごとの情報としては，デバイスタイプのアイコン，ホスト名，型名，メーカー名，IP アドレスを表示する．これらの情報が間違っていた場合，または表示されていない場合には，ユーザがアプリ上で修正を行うことができる．ここで修正された情報もまた蓄積され，デバイスタイプ判別に利用している．

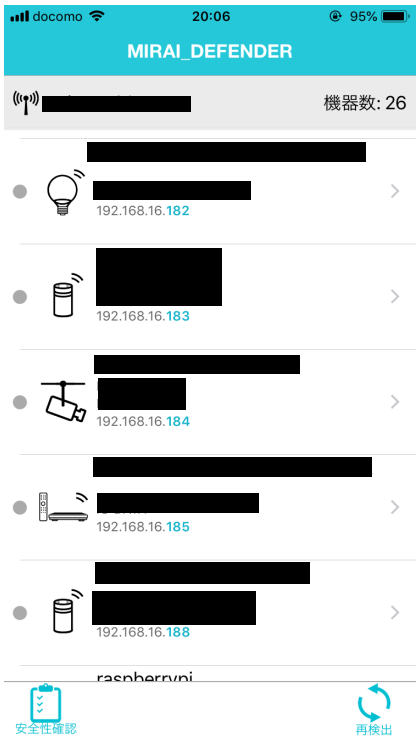


図 2 IoT デバイス一覧画面

一覧画面の左下の安全性確認をタップすると，ネットワークに接続された IoT デバイスに対する脆弱性診断を実施することができる．脆弱性診断を実施した結果を図 3 に示す．診断結果は，“安全（緑色）”と“危険（赤色）”の 2 種類のインジケータで表す．

脆弱性診断内容については，IoT デバイスの TCP ポート 23 番と 2323 番で待つ Telnet サービス，TCP ポート 80 番で待つ HTTP サービスが有効な場合に，工場出荷時の初期設定パスワードや脆弱とされるパスワードリストを用いて，実際にログインが可能か否か確認する．ログインに成功した場合，IoT デバイスのパスワードが脆弱であると判定する．診断に使用するパスワードリストは，マルウェア“Mirai”で用いられたパスワードリスト[12]など，最新の脅威動向に応じた情報をクラウド上の公開情報 DB から取得している．

図 3 脆弱性診断後の一覧画面

IoT デバイスのファームウェアバージョンなどの情報の通知は，セキュリティ対策の自動化および標準化を目指して策定が進められている SCAP（Security Content Automation Protocol）[13]を活用して実現する．“MIRAI_DEFENDER”では，CPE（Common Platform Enumeration）[14]情報を文字列検索することにより特定の CPE 識別子に対応させる．この CPE 識別子を使って関連する CVE（Common Vulnerabilities and Exposures）[15]情報を取り出すことで，当該 IoT デバイスに関連する脆弱性情報のみを絞り込んで配信する．また，同様に IoT デバイスの型名やファームウェアバージョンなどから，メーカーによるソフトウェアのセキュリティアップデート情報を検索して配信する．脆弱性情報については，NVD（National Vulnerability Database）[16]および JVN（Japan Vulnerability Notes）[17]から定期的に収集して構築している．

また，メーカーの公開情報については，定期的に Web ク

ローリングすることで，一部メーカーのファームウェアなどのソフトウェアに関するセキュリティアップデート情報を収集して構築している．

これらの脆弱性情報は，デバイスが対応しているファームウェアバージョンの日付以降の情報が表示されるようになっており，IoT デバイス一覧画面でデバイスをタップした際に表示される IoT デバイス詳細画面（図４）で確認することができる．

図 4 IoT デバイス詳細画面

スマートフォンアプリ“MIRAI_DEFENDER”は他のアプリに組み込めるように SDK（Software Development Kit）としても開発している．（以下，MIRADEF_SDK）

2.3 ユーザの利便性を重視したアプリの提供

2.2 節で述べた MIRADEF_SDK を組み込んだ“家電管理アプリケーション家電手帳（以下，家電手帳）”について紹介する．

“家電手帳”はメーカー名，モデル名，保証書やレシートの画像など，家電の情報をまとめて簡単に管理するためのアプリであり，図 5 のようにユーザの利便性向上を重視したアプリケーションである．

“家電手帳”では MIRADEF_SDK を家電登録方法の一つとして利用している．“家電手帳”では，家電登録の際に手動登録と Wi-Fi を使った自動登録を選択することができる．手動で登録を行う場合でも，画像や QR コードから登録することで比較的簡単に家電の登録を行うことができる．

図 5 家電手帳

Wi-Fi を使った自動登録を選択すると，MIRADEF_SDK を使ったデバイスの自動登録が実施され，デバイスの登録と共に脆弱性診断やソフトウェアアップデート検査などのセキュリティチェックも実施される．“家電手帳”では，家電の取扱説明書やリコール情報の通知，保証期限の通知など，ユーザにとって便利な機能を搭載している．

“家電手帳”は AppStore にて無償で公開[18] しており，誰でも自由に利用することができる．

3. IoT デバイスの検知と判別

2 章で述べた IoT デバイス管理システムでの IoT デバイス検知手法, デバイスタイプの判別手法について説明する．

3.1 IoT デバイスの検知

IoT デバイスの検知については, IoT デバイスがネットワークに接続されていることから，ICMP（Internet Control Message Protocol）エコー要求パケットを利用した Ping スキャンによる検知が可能である．ただし，Ping スキャンに応答しない IoT デバイスも存在するため，Ping スキャンに加えて ARP（Address Resolution Protocol）テーブルの確認を行うなどして，IoT デバイスの検知漏れを防ぐ．

また，ネットワーク上の機器を自動的に発見・接続する UPnP(Universal Plug and Play)で用いられる通信プロトコルである SSDP(Simple Service Discovery Protocol) [19]や, Apple Inc.が開発したゼロ・コンフィギュレーション技術の実装である Bonjour[20]などの様々な要求パケットに対する IoT デバイスの反応などを観測することで，IoT デバイスのフィンガープリンティングを行なっている．

3.2 IoT デバイスタイプの判別

IoT デバイス管理アプリケーションでは, IoT デバイスの検知時に収集する反応情報から, デバイスタイプを判別し, アイコンとして表示する．以下をデバイスタイプとして定義する．

・無線 LAN アクセスポイント　・PC
・スマートフォン　・タブレット
・ネットワークカメラ　・ゲーム機
・プリンター　・AI スピーカー
・メディアストリーミング端末　・ロボット
・スマートライト　・スマートプラグ
・メディアサーバ　・テレビ
・BD/DVD プレイヤー　・TV チューナー

デバイスタイプ判別の流れを図 6 に示す．サーバでは以下の流れでデバイスタイプを判別している．

1. デバイスの検知時にデバイス反応情報がサーバに送信される．
2. 反応情報を入力データとしてデバイスタイプ判別アルゴリズムによって判別を行う．
3. 判別されたデバイスタイプはアプリに送信され，デバイス一覧画面のアイコンとして表示される．

図 6 デバイスタイプ判別の流れ

4. IoT デバイス脆弱性調査

本章では，2 章で説明した IoT デバイス管理システムによる脆弱なパスワードリストを用いた IoT デバイスの脆弱性調査について説明し，結果を報告する．

また，家庭内の機器数やデバイスタイプの比率，デバイスタイプ毎のポートの空き状況などの統計情報についても紹介する．

4.1 調査対象データの定義

今回の調査対象データは，2.2 節で紹介した IoT デバイス管理アプリケーション“MIRAI_DEFENDER”と 2.3 節で紹介した家電管理アプリケーション“家電手帳”で収集したものである．それぞれのアプリケーションを利用するユーザのデータを収集し，分析することで脆弱性調査を実施した．なお，収集するデータには，ユーザの個人情報は含まず，デバイスの脆弱性等の情報のみである．収集期間は 2019 年 4 月から 2019 年 7 月の 4 ヶ月間であり，データ数は，83439 件である．

同一 LAN 内で，複数回アプリによる脆弱性確認をしているユーザもいるため，収集したデータには同一のデバイスのデータや家庭内以外のデータも含まれている．そこで我々は，デバイス毎に一意な ID として以下のようなデバイス ID を定義した．

デバイス ID : BSSID + IPv4 アドレス

BSSID とは，無線 LAN におけるアクセスポイントおよび無線ネットワークの識別子の一つで，48 ビットの値である．通常はアクセスポイントの MAC アドレスである．

本来，デバイス毎に一意な ID として，MAC アドレスを用いるべきである．しかし，iOS アプリケーションでは MAC アドレスを取得できない場合が多く，MAC アドレスでの集計は不可能である．そこで，本調査ではアクセスポイントの MAC アドレスである BSSID と，動的に割り振られて変化する場合はあるが，その家庭内で一意であると考えられる IPv4 アドレスを組み合わせた文字列をデバイス ID として定義する．

デバイス ID 単位で集計を行ったところ，83439 件のデータのうち，ユニークなデバイスのデータは 17260 件であった．本調査では 1 家庭に対してアクセスポイントは 1 つである場合が多いと仮定し，1 つの BSSID を 1 家庭と定義する．

このユニークなデバイスのデータのうち，1 つの BSSID に対して極端にデバイス数が多い場合があった．一般家庭では 50 件以上のデバイスが存在することは稀であると考える．そこで，本調査では 1 つの BSSID あたりのデバイス数が 50 件未満である場合に家庭内のデバイスであるとみなす．

集計の結果，1 つの BSSID に対してデバイス ID が 50 件未満であるデータは 4583 件であった．本調査では，このデータを調査対象データとする．また，調査対象に該当する家庭数は 535 件であった．

4.2 調査対象データの分析

本節では 4.1 節で定義した調査対象データについて分析する．

家庭内のデバイス数と家庭数の関係を図 7 に示す．535 家庭のうち，デバイス数が 10 機器未満の家庭が最も多く，1 家庭あたりの平均デバイス数は約 8.5 機器であった．

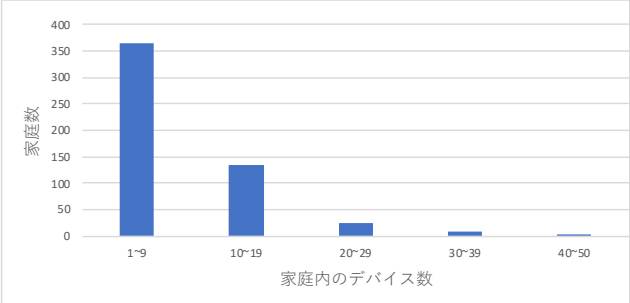


図 7　家庭内のデバイス数と家庭数

調査対象データ 4583 件のうち，3.2 節で定義したデバイスタイプのいずれかに判別できたデータは，2120 件である．このデータについて，デバイスタイプ毎のデバイス数の集計結果を図 8 に示す．

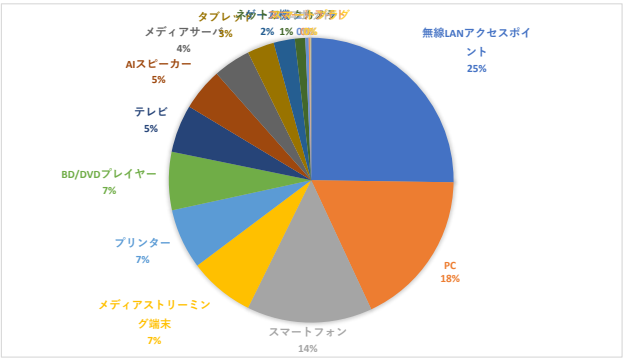


図 8 デバイスタイプ毎の集計結果

アクセスポイント，PC，スマートフォンが約 7 割を占めている．次いで多いのはメディアストリーミング端末，プリンター，BD/DVD プレイヤーであった．メディアストリーミング端末とは，Amazon.com Inc の“Fire TV Stick”や Google の“Chromecast”などのテレビに接続して直接アプリケーションを操作できるようにするデバイスの総称である．

次に，ポートスキャンの結果を示す．ポートスキャンの対象ポートは，主に国立研究開発法人情報通信研究機構（NICT）による NICTER 観測レポート[21]において攻撃対象として報告されているポートとしている．調査対象データ 4583 件の空きポート状況を図 9 に示す．

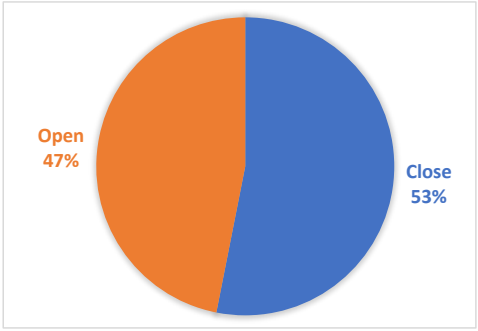
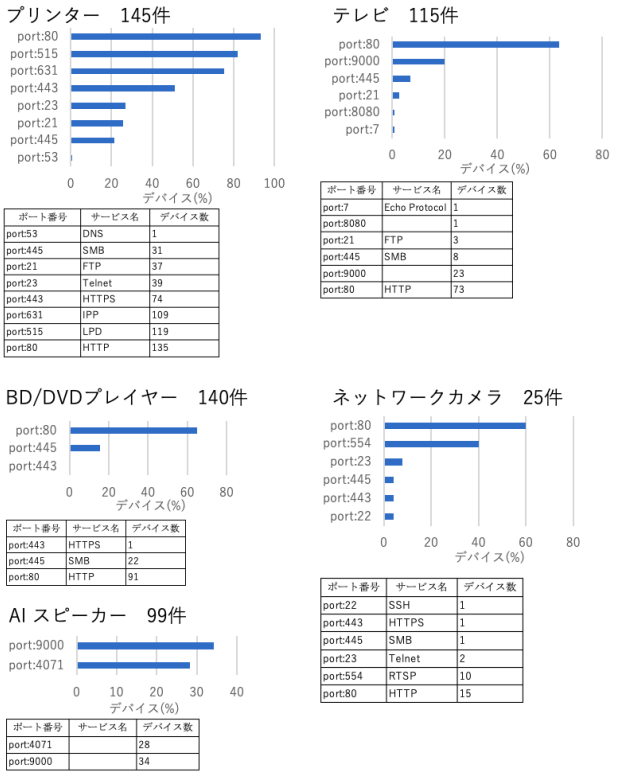


図 9 IoT デバイスの空きポート状況

調査対象データ 4583 件のうち，対象ポートが空いていたのは 2149 件であった．

また，ポートスキャンとデバイスタイプ判別により，デバイスタイプによって空きポート状況に特徴があることがわかった．デバイスタイプによって特徴のあった結果を図 10 に示す．



ポート番号	サービス名	デバイス数
port:53	DNS	1
port:445	SMB	31
port:21	FTP	37
port:23	Telnet	39
port:443	HTTPS	74
port:631	IPP	109
port:515	LPD	119
port:80	HTTP	135

ポート番号	サービス名	デバイス数
port:7	Echo Protocol	1
port:8080		1
port:21	FTP	3
port:445	SMB	8
port:9000		23
port:80	HTTP	73

ポート番号	サービス名	デバイス数
port:443	HTTPS	1
port:445	SMB	22
port:80	HTTP	91

ポート番号	サービス名	デバイス数
port:22	SSH	1
port:443	HTTPS	1
port:445	SMB	1
port:23	Telnet	2
port:554	RTSP	10
port:80	HTTP	15

ポート番号	サービス名	デバイス数
port:4071		28
port:9000		34

図 10 デバイスタイプ別の空きポート状況

4.3 脆弱性調査

4.1 節で述べた 4583 件の調査対象データに対してログインパスワードの脆弱性調査を実施した．脆弱性調査の内容は，デバイスの TCP ポート 23 番と 2323 番で待つ Telnet サービス，TCP ポート 80 番で待つ HTTP サービスが有効な場合に，工場出荷時の初期設定パスワードや脆弱とされるパスワードリストを用いて，実際にログインが可能か否かである．ログインに成功した場合，IoT デバイスのパスワードが脆弱であると判定する．

4.4 HTTP の脆弱性判定方法

この節では HTTP 認証における認証の種類と脆弱性の判定方法について説明する．HTTP 認証にはベーシック認証，ダイジェスト認証，フォーム認証の 3 つの認証が存在する．

ベーシック認証とダイジェスト認証と判定するのは，認証ページにアクセスした際のステータスコードが“401”の場合にレスポンスヘッダの“WWW-Authenticate”の項目に“Basic”や“Digest”が指定されている場合である．

ベーシック認証とダイジェスト認証における脆弱性については，脆弱とされるパスワードリストを使ってログインを試行し，レスポンスのステータスコードが 200 番台になった場合，脆弱性有りと判定している．

フォーム認証と判定するのは，認証ページにアクセスした際のステータスコードが 200 番台であり，該当ページにユーザ ID とパスワードを入力する項目のみがある場合である．

フォーム認証における脆弱性有無の判定について以下に述べる．

まず，意図的にログインできないユーザ ID とパスワードを用いてログインを試行し，到達したページの URL とテキストを記録する．その後に，脆弱とされるパスワードリストを使ってログインを試行する．この 2 回のログイン試行によるレスポンスステータスや到達した URL，テキストから脆弱性の有無を判定する．

脆弱性有りと判定するのは以下の 3 つの場合である．

1. 一度目のログイン試行でのレスポンスステータスが 200 番台ではなく，二度目のログイン試行で 200 番台のレスポンスステータスが返ってきた場合．
2. 二度のログイン試行において両方とも 200 番台のレスポンスが返ってくるが，到達した URL が異なる場合．
3. 二度のログイン試行において両方とも 200 番台のレスポンスかつ到達した URL が同一であっても，テキストが異なる場合．

また，フォーム認証の脆弱性判定時に，パスワードを入力するフォームは存在するが，ログインとは無関係なフォームが含まれるなどして，処理が異常終了した場合はフォーム認証であることは判明しているが，“脆弱性の有無は不明”としている．

上記以外で，ログインを試行した際にレスポンスステータスが 200 番台でも“401”でもない場合やレスポンスステータスが取得できなかった場合は“認証種別不明”としている．

4.5 脆弱性調査結果

脆弱性調査の結果を表 1 に示す．Telnet サービスに対する脆弱性調査の結果， TCP ポート 23 番または 2323 番が空いていたデバイスは 81 件であり，このうち脆弱なパスワードが設定されていたデバイスは 11 件であった．

4.4 節で説明した判定方法により HTTP サービスに対する脆弱性調査を実施した結果， TCP ポート 80 番が空いていたデバイスは 616 件であり，このうち認証が設定されていたデバイスは 351 件であった．認証が設定されているデバイスのうち脆弱なパスワードが設定されていたデバイスは 66 件であった．設定されている認証の内訳としては Basic 認証が 124 件，ダイジェスト認証が 9 件，フォーム認証が 126 件であった．

表 1 脆弱性調査結果

		脆弱性無し	脆弱性有り	
Telnet		86.4%(70/81)	13.6%(11/81)	
HTTP	ベーシック認証	54.8%(68/124)	45.2%(56/124)	18.8%(66/351)
	ダイジェスト認証	66.7%(6/9)	33.3%(3/9)	
	フォーム認証	66.7%(14/21)	33.3%(7/21)	
	フォーム認証(脆弱性の有無は不明)	105件		
	認証種別不明	92件		

上記の結果から，HTTP 認証が設定されているデバイスに関して約 18.8%，Telnet が空いているデバイスに関して約 13.6%の割合で脆弱なパスワードが設定されていることがわかった．認証方式別ではベーシック認証が設定されているデバイスの約 45.2%において脆弱なパスワードが設定されていることが判明した．

脆弱なパスワードが設定されていたデバイスのうち，機器情報からメーカー名が判明しているデバイスについて，メーカー別の内訳を図 11 に示す．

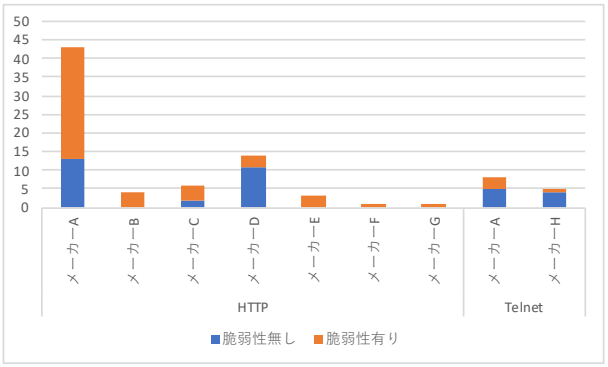


図 11 メーカー別の脆弱性

HTTP および Telnet のパスワードの脆弱性において，共にメーカーA の無線 LAN ルータが多くみつかった．脆弱性のある無線 LAN ルータについて調査したところ，初期ユーザ名と初期パスワードが推測されやすいものに設定されていることがわかった．

5. 考察

本調査により，HTTP 認証が設定されているデバイスに関して約 18.8%，Telnet が空いているデバイスに関して約 13.6%の割合で脆弱なパスワードが設定されていることがわかった． HTTP 認証に関しては，ベーシック認証やダイジェスト認証，フォーム認証などの認証が設定されている機器の約 25.5%において脆弱なパスワードが設定されていることがわかった．脆弱なパスワードが設定されている原因としては，初期設定のパスワードがそのまま用いられていることが挙げられる．

また，ポートスキャンの結果から，攻撃対象となり得るポートが空いたままのデバイスが多く存在していることが

判明した．このような適切な管理がされていない IoT デバイスは今後も増加し，マルウェアの脅威に晒されることが考えられる．

IoT デバイスを提供するメーカー側として，初期のユーザ名やパスワードを推測されにくいものにする必要はあるが，ユーザとしては IoT デバイスにおける脅威を認識し，適切なパスワードを設定するなど，ユーザ自身での適切な管理が必要であると考える．

6. まとめと今後の課題

現在，IoT デバイス管理アプリケーションにおけるパスワードの脆弱性診断は Telnet サービスと HTTP サービスに対して実施しているが，HTTPS サービスや SSH サービスに対しても同様に実施すべきであると考えている．また，家庭内 LAN からの調査と並行して外部 WAN からの脆弱性を診断する機能の実装も検討中である．

これらを実現することにより，IoT デバイス管理アプリケーション単体で WAN 側と LAN 側双方からの脆弱性診断が可能となり，ユーザはより安心して IoT デバイスを使用できるようになる．

本稿で紹介した IoT デバイス管理アプリケーションである“MIRAI_DEFENDER”や，“家電手帳”は無償で公開している．これらを利用することで，ユーザによる IoT デバイスの適切な管理を助け，IoT セキュリティの向上に貢献できると考える．

また，本稿のように IoT デバイス管理アプリケーションによって収集したデータを調査・分析した結果を公開することで，ユーザに対して IoT セキュリティ対策の助言や注意喚起を実施し，IoT セキュリティの向上に貢献する．

参考文献

[1] “平成 30 年版情報通信白書”，総務省, 2018. http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/pdf/30point.pdf （参照 2019-08-19）.

[2] “新しい Mirai 亜種”，パロアルトネットワークス, 2019. https://www.paloaltonetworks.jp/company/in-the-news/2019/new-mirai-variant-targets-enterprise-wireless-presentation-display-systems（参照 2019-08-19）.

[3] “SophosLabs 2019 年版脅威レポート”，ソフォス株式会社 SophosLabs, 2019. https://www.sophos.com/ja-jp/medialibrary/pdfs/technical-papers/sophoslabs-2019-threat-report.pdf, （参照 2019-08-19）.

[4] 中澤祐樹, 佐々木良一, 猪俣敦夫. “野良 IoT の地域特性の調査と分析”，マルチメディア、分散、協調とモバイルシンポジウム, 2017.

[5] 飯田正樹, 亀谷聡, 永見健一, 遠藤貴裕, 古瀬正浩. “IoT のための PKI によるシステム構築方法の提案”，コンピュータセキュリティシンポジウム, 2016.

[6] 飯田正樹, 永見健一, 遠藤貴裕, 舟根優作. “セキュアな IoT を実現する簡便な IoT デバイス管理システムの提案”，コンピュータセキュリティシンポジウム, 2017.

[7] “IoT デバイス調査及び利用者への注意喚起の取組「NOTICE」について”，総務省, 2019. http://www.soumu.go.jp/main_content/000597680.pdf（参照 2019-08-19）.

[8] “脆弱な IoT デバイス及びマルウェアに感染している IoT デバイスの利用者への注意喚起の実施状況”，総務省, 2019. http://www.soumu.go.jp/menu_news/s-news/01cyber01_02000001_00033.html（参照 2019-08-19）.

[9] “平成 30 年通信利用動向調査の結果”，総務省, 2019. http://www.soumu.go.jp/johotsusintokei/statistics/data/190531_1.pdf（参照 2019-08-19）

[10] INTEC Inc. “MIRAI_DEFENDER - Google Play の Android アプリ”. https://play.google.com/store/apps/details?id=jp.co.intec.miraidefender, （参照 2019-08-19）.

[11] INTEC Inc. “MIRAI_DEFENDER を App Store で”. https://itunes.apple.com/jp/app/mirai-defender/id1219842704, （参照 2019-08-19）.

[12] “Mirai-Source-Code/scanner.c at master · jgamblin/Mirai-Source-Code · GitHub”. https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/bot/scanner.c, (参照 2019-08-19).

[13] “The Security Content Automation Protocol （SCAP） - NIST”, National Institute of Standards and Technology. https://scap.nist.gov/,（参照 2019-08-19）.

[14] “Common Platform Enumeration: CPE”, The MITER Corporation. http://cpe.mitre.org/, （参照 2019-08-19）.

[15] “CVE - Common Vulnerabilities and Exposures (CVE)”, The MITER Corporation. https://cve.mitre.org/, （参照 2019-08-19）.

[16] “NVD - Data Feeds”, National Institute of Standards and Technology. https://nvd.nist.gov/vuln/data-feeds, （参照 2019-08-19）.

[17] “JVN iPedia - 脆弱性対策情報データベース”，情報処理推進機構. http://jvndb.jvn.jp/#jvndbrss, （参照 2019-08-19）.

[18] INTEC Inc. “「家電手帳 - 家電をまとめて安全に管理」を App Store で”. https://apps.apple.com/jp/app/id1453357489, （参照 2019-08-19）.

[19] “UPnP Device Architecture 2.0”, UPnP Forum, 2015. http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v2.0.pdf, （参照 2019-08-19）.

[20] “Bonjour - Apple Developer”, Apple Inc. https://developer.apple.com/bonjour/, （参照 2019-08-19）.

[21] “NICTER 観測レポート 2018”，国立研究開発法人情報通信研究機構. https://www.nict.go.jp/cyber/report/NICTER_report_2018.pdf, （参照 2019-08-21）.