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

飯田 正樹[†1] 永見 健一[†1] 遠藤 貴裕[†1] 舟根 優作[†1]

概要：近年，マルウェア“Mirai”の感染など，モノのインターネット（IoT，Internet of Things）で利用される多種大量のデバイスを狙ったサイバー攻撃による被害が増加している．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，適切な認証設定やメーカーから提供されるセキュリティアップデートの適用など，基本的な対策が極めて重要である．しかしながら，IT 技術に精通していない一般的なユーザが対策の必要性に気付くのは難しく，IoT デバイスが適切に管理されているとは言い難い．本稿では，IT 技術に精通していないユーザであっても，簡単に脆弱性診断や必要な情報収集が行える IoT デバイス管理システムを提案する．提案システムによって，IoT デバイスの適切な管理を助け，ユーザ側からセキュアな IoT を実現する．

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

The Proposal of a Convenient IoT Device Manager for Secure IoT

Masaki Iida[†1] Kenichi Nagami[†1] Takahiro Endo[†1] Yuusaku Funane[†1]

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

1. はじめに

1.1 IoT に対する脅威と IoT デバイス管理の現状

近年，コネクテッドカーやスマート TV 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デバイスが 500 億台に達す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．最近では，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マルウェア“Mirai”の事例が報告されている[2]．また、マルウェア“WannaCry”のように水平感染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利用される IoT デバイス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，適切な認証設定やメーカーから提供されるセキュリティアップデートの適用など，基本的なセキュリティ対策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．

1.2 本研究の貢献

IoT 全体のセキュリティ向上のためには，我々がこれまで取り組んできたような IoT システム側のセキュリティ対策技術[5]だけではなく，ユーザ側の適切な IoT デバイス管理を支援することとの両輪で進める必要があると考える．

そこで本研究では，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，適切な IoT デバイス管理のために必要な機能を検討する．そして，それらの機能を実装した簡便な IoT デバイス管理システムを提案する．

1.3 関連する先行研究

高橋の研究[6]では，組織内の IT 資産について収集された各種情報から IT 資産情報の識別子を特定し，その特定した識別子をもとに関連する脆弱性情報を取得するという，2 段階の検索処理を用いた脆弱性自動配信システムを構築している．我々の研究においても，この手法を参考にして，IoT デバイスの各種情報から当該 IoT デバイスの識別子を特定し，それをもとに関連する脆弱性情報を取得するという方法を検討する．

一方で，我々の研究対象とする IT 技術に精通していないユーザによる IoT デバイス管理という状況においては，それ特有の課題を加味した手法の追加検討が必要となる．

†1 株式会社インテック
INTEC Inc.

2. IoT デバイス管理の課題

IT 技術に精通していないユーザが，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 デバイスを適切に管理できない要因を検討し，3 つの課題として整理する．

2.1 管理すべき IoT デバイス自体の認識

IoT デバイスが適切に管理されていない原因の一つとして，サイバー攻撃の対象となり得る IoT デバイスがネットワークに接続されていることを，ユーザ自身がそもそも認識できていない可能性が考えられる．

IT 技術に精通したユーザを対象として, 2016 年に実施された企業内ネットワークにおける IoT に関する意識調査[7]では，ネットワークに接続された全ての IoT デバイスを認識および管理できている自信があると回答したのは，全体の僅か 15%にとどまっている．また，IoT デバイスを保有していないと回答したユーザであっても，実際には平均して約 8 個の IoT デバイスがネットワーク上に存在していたと報告している. 本調査が IT 技術に精通したユーザに対して実施されたものであることを考えれば，IT 技術に精通していないユーザでは，管理すべき IoT デバイス自体を認識している割合は更に低くなると推測される．

2.2 IoT デバイスの設定不備や脆弱性の認識

前述したように，“管理者不在”の野良 IoT デバイスの存在が指摘されており，適切な認証設定などが行われないまま利用されている IoT デバイスが存在する．この原因として，IoT デバイスのセキュリティに関する設定が適切に行われていないにもかかわらず，ユーザが“適切に設定・管理できている”と思い込み，設定不備の問題に気付いていない可能性が考えられる．

また，メーカーから提供されるセキュリティアップデートなどの情報にユーザが気付いていない可能性も考えられる．経年劣化による重大製品事故の発生の恐れが高い製品を対象とした長期使用製品安全点検制度では，メーカーへのユーザ登録率が 2017 年 3 月末時点で 38.7%にとどまることが報告されている[8]．メーカーへのユーザ登録を行わない場合，脆弱性やセキュリティアップデートの情報を得るためには，基本的にユーザ自身が能動的に情報収集を行う必要がある．一人のユーザにより複数種類かつメーカーも異なる IoT デバイスが利用されることを考えると，ユーザ自らが必要な情報を全て収集するのは現実的ではない．IoT デバイスの設定不備や脆弱性などの存在については，ユーザの認識を補助する仕組みが必要である．

2.3 難しい操作の簡略化および自動化

IT 技術に精通したユーザであれば，Nmap[9]などのセキュリティスキャナを活用して，ネットワークに接続された IoT デバイスの種類や台数，それらの不適切な設定を検知できる可能性がある．しかしながら，一般家庭などで利用される IoT デバイスの多くは IT 技術に精通していない一般的なユーザが利用すると思われる．そのため，専門知識が必要なセキュリティ検査ツールを使いこなし，IoT デバイスの管理を継続して実施するのは困難であると考える．

仮に, ユーザが管理すべき IoT デバイスの存在を認識し，更に脆弱性の存在を認識して，メーカーから提供されるセキュリティアップデートや，推奨される設定値を適用しようと思い至ったとしても，その適用するための操作自体が難しければ，実施することを諦めてしまう可能性もある．

IoT デバイスの管理に必要な難しい操作の簡略化および自動化を行い，ユーザが IoT デバイスの管理を無理なく継続できる仕組みが必要となる．

3. 提案システムのコンセプトと設計概要

我々は，第 2 章で検討した IoT デバイス管理の課題を仮定し，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 デバイス管理の仕組みが必要と考え，次の特徴を持つ IoT デバイス管理システムを提案する．

3.1 エージェントレスでの IoT デバイスの検知と判別

ユーザが管理すべき IoT デバイス自体を認識するためには，IoT デバイスの検知と判別を行う仕組みが必要である．特に IoT デバイスに関しては，使用できる計算リソースや自由度に制限があるため，エージェントなどの情報収集のための追加ソフトウェアをインストールすることが難しい．そのため，IoT デバイスの外部からの観測のみで検知および判別を行う“エージェントレス”が必須の要件となる．

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．

IoT デバイスの判別については，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試みる．

3.2 IoT デバイス単位での設定不備や脆弱性情報の通知

IoT デバイスの設定不備として，適切なパスワード設定が行われていない状態で利用されている例が見られる[10]. 例えば，工場出荷時点の初期設定パスワードのまま利用されていたり，“password”や“12345”などの単純なパスワードが設定されていたりする場合，特に攻撃を受けやすいと考えられる．そのため，提案システムでは，脆弱とされる

パスワードリストを用いて IoT デバイスのパスワード設定を検査し，その結果をユーザに通知する．

また，ユーザによる IoT デバイス情報の入力や，第 3.1 項に記載した IoT デバイスの判別によって，IoT デバイスの型名やメーカー名などを取得できる場合には，関連する脆弱性情報をユーザに通知する．具体的には，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）[11]を活用して実現する．以下に SCAP バージョン 1.0 を構成する 6 つの標準仕様を概説する．

- **CVE（Common Vulnerabilities and Exposures）**
  個別製品の脆弱性に対して付与される識別子である．脆弱性情報同士の関連付けなどにも利用できる．
- **CCE（Common Configuration Enumeration）**
  パスワードの長さや複雑さなど，システムのセキュリティ設定項目に対して付与される識別子である．
- **CPE（Common Platform Enumeration）**
  個別のハードウェア，OS，アプリケーションなどに対して付与される識別子である．
- **CVSS（Common Vulnerability Scoring System）**
  脆弱性の深刻度を算出するための評価手法であり，脆弱性の深刻度を定量的に比較することができる．
- **XCCDF（eXtensible Configuration Checklist Description Format）**
  セキュリティ設定のチェックリストなどを記述する仕様言語である．CCE などと組み合わせて使用する．
- **OVAL（Open Vulnerability and Assessment Language）**
  脆弱性対策のためのセキュリティ設定の確認作業などを自動化するための言語である．

この CVE や CPE を活用したサービスの一例として，Saucs[12]というサービスが公開されている．Saucs は，Web ベースの GUI を備え，予め CPE 識別子を登録しておくことで関連する CVE 情報を一覧表示する．IoT デバイスのユーザは，Saucs に自身が保有する IoT デバイスの CPE 識別子を一度登録しておくことで，関係する新たな脆弱性が報告された際に通知を受け取ることができる．しかしながら，Saucs のような Web ベースの脆弱性情報通知サービスは，ユーザのネットワークに接続されている実際の IoT デバイス群と連動した管理を行うことができない．そのため，そもそもユーザが管理すべき IoT デバイス自体を認識できていない場合，脆弱性情報通知サービスに当該 IoT デバイスが登録されず，IoT デバイスの管理漏れが発生してしまう．そこで，本稿で扱う IoT デバイス管理の状況においては，第 3.1 項で述べた IoT デバイスの検知と判別を組み合わせた脆弱性情報通知の仕組みを検討する．

3.3 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つつあると言われている[13]．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，本提案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できる IoT デバイス管理システムを考えたとき，スマートフォンは有力なプラットフォームの一つであると考える．

4. 提案システムの実装

本章では，提案システムのコンセプトと設計概要をもとに実装した IoT デバイス管理システムの全体構成，並びにユーザが使用する各画面とその機能について説明する．

4.1 システム全体構成

実装した提案システム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（以下，スマホアプリ）”，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．

スマホアプリでは，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，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ことの検査，そして 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収集する．

スマホアプリで収集された 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．この蓄積された反応情報を用いて機械学習を行うことで，IoT デバイスの型名やファームウェアバージョンを判別するためのフィンガープリンティングを実施する．型名などが推定されない場合、または推定された型名などが間違っていた場合には，ユーザがスマホアプリ上で修正を行うことができる．この修正情報もまた蓄積して機械学習に利用することにより，判別精度の向上を図っている．

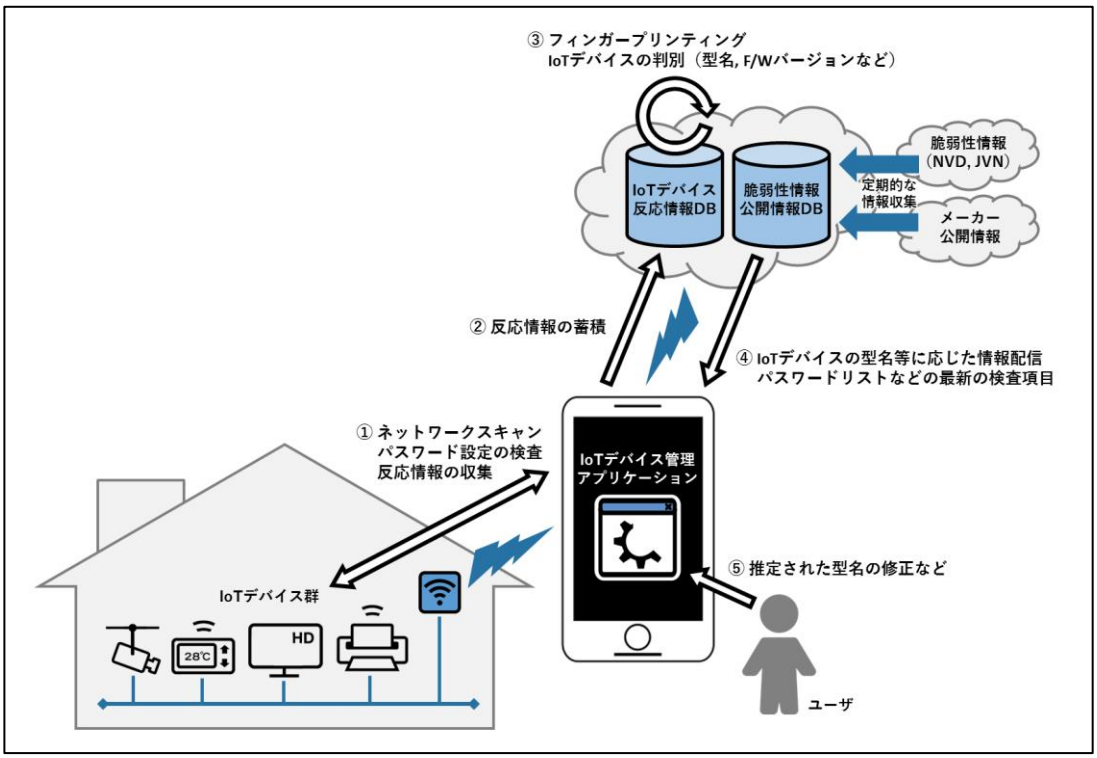

図 1　IoT デバイス管理システムの全体構成と処理の流れの一例

推定された IoT デバイスの型名やファームウェアバージョンなどの情報は，CPE 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 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）[14]および JVN（Japan Vulnerability Notes）[15]から定期的に収集して構築している．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．

4.2　IoT デバイス一覧画面

スマホアプリを起動すると，最初に IoT デバイス一覧画面が表示される．ユーザは本画面にて，右上のネットワークスキャンボタン，そして左上の脆弱性診断ボタンをタップすることで，ネットワークに接続された IoT デバイスの検出および一覧表示，そして検出された 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．

図 2 は，脆弱性診断を実施した後の IoT デバイス一覧画面の様子である．脆弱性診断の結果は，“安全（緑色）”と“危険（赤色）”の 2 種類のインジケータで表す．“危険”と判定されたデバイスが存在する場合，図 3 に示す脆弱性診断結果の通知画面でユーザに問題への対処を促す．

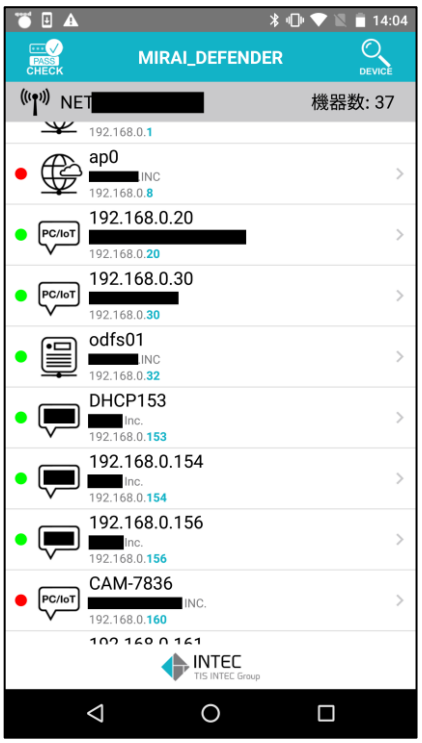

図 2　脆弱性診断実施後の IoT デバイス一覧画面

図 3　脆弱性診断結果の通知画面

脆弱性診断内容については，IoT デバイスに対して下記の 3 点を検査し，何れか一つでも問題が見つかれば“危険”と判定する．

- **パスワードの安全性**

  IoT デバイスのポート“23/tcp”と“2323/tcp”で待ち受ける TELNET サービスが有効な場合に，工場出荷時点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[16]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 から配信する．

  HTTP/HTTPS サービスの基本認証やダイジェスト認証に関するパスワードの診断については，IoT デバイスによっては，ログインの試行により認証機能がロックアウトしてしまうものが存在したため，現状では診断の実施対象外としている．

- **セキュリティアップデートの適用状況**

  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取得し，IoT デバイスに対して最新のセキュリティアップデートが適用されているか否かを確認する．最新のセキュリティアップデートが適用されていない場合，IoT デバイスが脆弱な状態にあると判定する．

- **脆弱性への対応状況**

  IoT デバイスに関係する脆弱性情報を取得し，対応済みか否かを確認する．実際には，当該 IoT デバイスに関係する脆弱性情報が登録されているにも関わらず，メーカーのセキュリティアップデート情報を取得できない場合，もしくは本システムがそのメーカーの情報取得に未対応の場合などに，IoT デバイスが脆弱な状態にあると判定している．この場合には，ユーザに対して，メーカーに対応方法を確認するように促す．

4.3　IoT デバイス詳細画面

図 4 に示す IoT デバイス詳細画面は，IoT デバイスの詳細な情報を確認したい場合に，IoT デバイス一覧画面で対象の IoT デバイスをタップして画面遷移することで利用できる．IT 技術に精通したユーザを想定して用意した画面であり，必ずしも利用する必要はない．

IoT デバイス詳細画面では，脆弱性情報やセキュリティアップデート情報の一覧表示，フルポートスキャンの実行などが可能である．また，脆弱性情報やセキュリティアップデート情報をタップすることで，情報提供元の Web ページを表示し，詳細な内容を確認することができる．

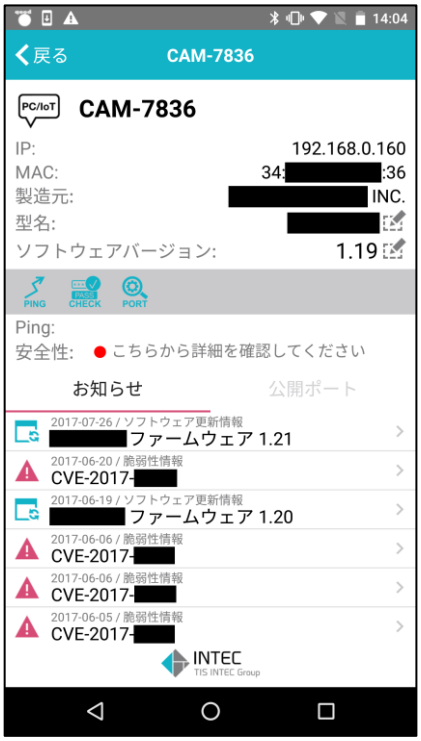

図 4　IoT デバイス詳細画面

5.　今後の取り組みと課題

今後の取り組みと課題について，IoT デバイス判別精度の向上，脆弱性診断内容の充実，ユーザによる確認・設定作業の自動化，提案システムの有効性の検証，以上の 4 点について詳細を述べる．

5.1 IoT デバイス判別精度の向上

提案システムでは，IoT デバイスの反応情報を蓄積し，機械学習を用いることで，IoT デバイスの型名判別を行うためのフィンガープリンティング技術の実装を進めている．しかしながら，現段階では十分な学習データを得られておらず，判別精度も十分ではない．今後は，学習データの“質”と“量”，特に“質”を向上させるための検討を進める．

5.2 脆弱性診断内容の充実

パスワードの安全性に関する脆弱性診断は TELNET サービスに対してのみ実施しているが，SSH や HTTP/HTTPS 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．特に HTTP/HTTPS サービスについては，IoT デバイスの管理コンソールとして使用されることが多いため，パスワード設定の診断は必要であると考える．しかしながら，前述したように，ログインの試行により認証機能がロックアウトしてしまう問題があるため，現時点では実施しておらず，解決策を検討している最中である．

また，図 5 に示すように，蓄積している IoT デバイスの反応情報を可視化して統計的に分析することで，より効果的な脆弱性診断に活用できる可能性がある．例えば，IoT デバイスの公開ポート種別毎の台数によって，より影響範囲の大きいサービスに比重を置いた脆弱性診断を実施するなどが考えられる．また，こうした分析情報を利用することで，ユーザに対するセキュリティ対策の注意喚起を効果的に行うなどの活用も考えられる．

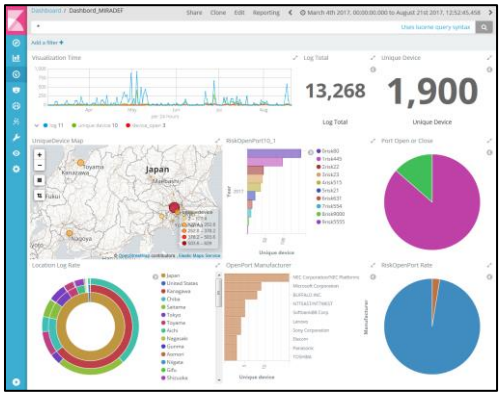

図 5　IoT デバイス情報の可視化と分析

5.3 ユーザによる確認・設定作業の自動化

実装した提案システムでは，IoT デバイスに関連する脆弱性情報の一覧表示を行うことができる．しかしながら，その脆弱性が実際にユーザ自身の環境で影響するのか否か，どれほどの深刻度であるのかなど，IoT デバイス個別の設定内容などに依存した確認はユーザ自身で行う必要がある．前述した SCAP の標準仕様に沿った実装を更に進めていくことで，こうした作業を自動化し，必要最小限の脆弱性情報のみをユーザに提示することで，ユーザの負担を軽減することが期待できる．また，IoT デバイスの設定値が脆弱な場合に，自動的に適切な設定値に変更するなどの自動化も考えられる．

5.4 提案システムの有効性についての評価

本稿では，IT 技術に精通していないユーザを対象とした IoT デバイス管理の課題検討と，それを解決するためのシステムの提案にとどまり，提案システムの有効性を評価できていない．今後，アンケート調査を実施するなどして，提案システムの有効性について評価する必要がある．

6. おわりに

本稿では，IT 技術に精通していないユーザを対象として，適切な IoT デバイス管理のために必要な機能を検討した．また，それらの機能を実装した簡便な IoT デバイス管理システムを提案した．提案システムによって，脆弱性診断の自動化，そして保有する IoT デバイスに関する各種情報の一覧性が高まることで，IoT デバイス管理にかかるユーザの負担を軽減させることが期待できる．

本稿で提案したシステムは，“MIRAI_DEFENDER”という名称で，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[17][18]しており，誰でも自由に利用することができる．アプリケーションストアを通じ，ユーザから使い勝手などのフィードバックを得ることで，より簡便に利用できる仕組みへと継続的に改善していく．

本研究を進めることで，IoT システム側のセキュリティ対策技術だけではなく，ユーザによる IoT デバイスの適切な管理を助け，IoT 全体のセキュリティ向上に貢献する．

参考文献

[1] Dave Evans. “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything”, Cisco Internet Business Solutions Group, 2011. https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, (参照 2016-08-08).

[2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. “Understanding the Mirai Botnet”, 26th USENIX Security Symposium, 2017. https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf, (参照 2017-08-22).

[3] “ランサムウェア「WannaCry」対策ガイド rev.1”, 株式会社ラック コーポレート・コミュニケーション室, 2017. https://www.lac.co.jp/lacwatch/report/20170519_001289.html, (参照 2017-08-24).

[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] 高橋健志. “ネットワーク上の IT 資産に関する脆弱性情報自動配信システム”, 情報通信研究機構研究報告 Vol.62 No.2, 2016.
[7] Steven Taylor. “The Internet of Things isn’t coming. It’s here.”, Webtorials, 2016. https://www.forescout.com/wp-content/uploads/2016/06/ForeScout-Webtorials-IoT-Security-Survey-Results-June-2016.pdf, (参照 2017-08-10).
[8] “長期使用製品安全点検制度の登録率向上に向けた取組みについて”, 経済産業省 商務流通保安グループ 製品安全課, 2017. http://www.meti.go.jp/committee/sankoushin/shojo/seihin_anzen/pdf/005_04_00.pdf, (参照 2017-08-25).
[9] “Nmap: the Network Mapper - Free Security Scanner”, Nmap.Org. https://nmap.org/, (参照 2017-08-25).
[10] Mario Ballano Barcena, and Candid Wueest. “Insecurity in the Internet of Things”, Symantec Security Response, 2015. https://www.symantec.com/content/dam/symantec/docs/white-papers/insecurity-in-the-internet-of-things-en.pdf, (参照 2016-08-08).
[11] “The Security Content Automation Protocol (SCAP) - NIST”, National Institute of Standards and Technology. https://scap.nist.gov/, (参照 2017-08-25).
[12] “Saucs”, saucs.com. https://www.saucs.com/, (参照 2017-08-04).
[13] “平成 29 年版情報通信白書”, 総務省, 2017. http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/29honpen.pdf, (参照 2017-08-22).
[14] “NVD - Data Feeds”, National Institute of Standards and Technology. https://nvd.nist.gov/vuln/data-feeds, (参照 2017-06-20).
[15] “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 2017-06-20).
[16] “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 2017-03-24).
[17] 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 2017-08-27).
[18] INTEC Inc. “MIRAI_DEFENDER を App Store で”. https://itunes.apple.com/jp/app/mirai-defender/id1219842704, (参照 2017-08-27).