IoT セキュリティチェックリストに基づいた脆弱性検査の自動化

瀧口萌花[1]　塩原孝弘[2]　加藤雅彦[1]

概要：近年，IoT 機器が急速に普及しているが，これらの中には脆弱なデバイスも多いことから IoT 機器を対象とする攻撃が増加している．このような攻撃から機器を保護するためには，機器の脆弱性検査を行い，脆弱性を取り除く必要がある．しかし，脆弱性検査を行うためには，IoT 機器のシステムやセキュリティに関する詳細な知識が必要であり，また，労力がかかることから，全ての開発者や利用者が脆弱性検査を行うことは難しい．そこで，本研究では IoT 機器に特化した脆弱性検査として，JPCERT/CC の「IoT セキュリティチェックリスト」を使った，脆弱性検査の自動化手法を提案し，プロトタイプを作成する．複数の IoT 機器に脆弱性検査を行った結果，手動で検査を行った場合と比較して，機器のセキュリティ対策状況を定量的に把握することができた．

キーワード：IoT，脆弱性検査，セキュリティ，自動化

Automating vulnerability checks based on the IoT Security Checklist

MOEKA TAKIGUCHI[1]　TAKAHIRO SHIOHARA[2]
MASAHIKO KATO[1]

Abstract: In recent years, IoT devices are rapidly spreading, but there are many vulnerable devices among them, and attacks targeting IoT devices are increasing. In order to protect it from such attacks, it is necessary to conduct vulnerability assessment and remove the vulnerability. However, in order to conduct vulnerability assessment, it is difficult for all developers and users to conduct vulnerability assessment because it requires detailed knowledge about the system and security of IoT devices, and it takes time. Therefore, in this paper, we propose and prototype an automated method of vulnerability assessment using JPCERT/CC IoT SECURITY CHECKLIST as a vulnerability assessment specialized for IoT devices. As a result of performing vulnerability assessment on multiple IoT devices, we quantitatively grasped the security measures of the devices compared with the case of performing the assessment manually.

Keywords: IoT，Vulnerability inspection，Security，Automation

1. はじめに

近年，IoT 機器が普及し，インターネットに接続される機器が増加している．IPA によると，2020 年には 200 億を超えるモノがインターネットを含む様々なネットワークと接続し，情報をやりとりすると言われている [1]. しかし，現在普及している IoT 機器の中には脆弱なデバイスも多く，今後さらに多くの IoT 機器が広く利用されるようになることから，IoT 機器を対象とする攻撃が増加することが予想される [2].

IoT 機器を対象とする攻撃として，米国でマルウェアに感染した IoT 機器が踏み台となり，大規模な DDoS 攻撃が発生し，一部サイトにアクセスできなくなる障害が発生した [3]．また，NICT(情報通信研究機構)が運用するサイバー攻撃観測網(NICTER)が観測したサイバー攻撃パケット，1504 億パケットのうち，半数以上が IoT 機器を狙ったものであった [4]．このように，IoT 機器を対象とする攻撃は，増加傾向にあることから，IoT 機器におけるセキュリティ対策が急務となっている．

このような攻撃から IoT 機器を保護するためには，機器の脆弱性検査を行い，脆弱性を取り除く必要がある．しかし，脆弱性検査を行うためには，IoT 機器のシステムやセキュリティに関する詳細な知識が必要であるため，全ての開発者や利用者自身が脆弱性検査を行うことは困難である．また，家庭用 IoT 機器の場合，検査や更新が利用者に委ねられていることが多い．従来の自動脆弱性検査ツール（以下，脆弱性スキャナ）では検査内容や結果が専門的であることから，利用者自身で検査を行うには扱いが難しいという問題がある．さらに，家電や防犯機器など家庭でも使用できる IoT 機器が年々増加しており，全ての機器の脆弱性検査を専門機関に依頼する場合，コストがかかるという問題がある．

そこで，本研究では IoT 機器を利用する全ての人が容易に脆弱性検査を行うことができるシステムを目指し，IoT の脆弱性検査を自動化する手法を提案する．提案機構は，IoT 機器に自動的に脆弱性を突く攻撃を仕掛け，攻撃結果から脆弱性を検査する．この脆弱性検査の自動化により，開発者にとっては検査時間を短縮することが可能となり，利用者にとっては詳細な知識がなくても脆弱性検査を行う

1 長崎県立大学
University of Nagasaki　bs216020@sun.ac.jp
2 TDK 株式会社
TDK Corporation

ことが可能となる．

本提案機構の有効性を示すために，自動脆弱性検査システムを開発し，複数の IoT 機器を用いて実験を行った．実験では，本システムを用いた自動化脆弱性検査と手動で脆弱性検査を行ったときの結果を比較し，結果が正しいことを確認した．以降，2 章で先行研究を紹介し，3 章で本研究における提案手法を述べる．4 章で提案手法を実装し，5 章で複数の IoT 機器を対象として評価を行う．6 章で課題等について考察を行い，7 章でまとめる．

2. 先行研究

本章では，脆弱性検査の自動化に関する研究で利用されている手法と，本研究で参考にした IoT 製品に対する脆弱性検査項目についての関連研究について述べる．

脆弱性検査の自動化で利用されている手法として，脆弱性を突く攻撃を自動的に生成する方法が多く用いられる[5]．この手法を用いて，ウェブアプリケーションのセッション管理における脆弱性検査の自動化を行った研究によると，HTTP リクエストやレスポンス内の Cookie 情報から，セッション ID の名前を抜き出し，セッション ID をどの方式で伝播しているかを判断することでセッション管理における脆弱性検査の自動化を行っている．しかしながらこの手法では，脆弱性検査に必要となるログイン等の操作手順はウェブアプリケーション毎に固有であることから，利用者が検査することを前提に手動での操作が一部残っており，検査対象毎に手動でログイン操作を行ったうえで脆弱性検査を行う必要がある．

また，脆弱性検査ツールとして Openvas [6],nmap [7],owasp zap [8]等が広く使われている．Openvas は，診断対象にインストールされているソフトウェアに脆弱性があるかどうかを調べる際に用いられ，検査結果として重大度や脆弱性の詳細を確認できるレポートが得られる．nmap は，ポートスキャンにより脆弱性検査を行う際に用いられ，単純なポートの開閉状況だけでなく，検査対象の OS 検出等単純なコマンドで多彩な検査が可能となる．owasp zap は，web アプリケーション型の脆弱性検査を行う際に用いられ，URL を指定して検査を行い，幅広い脆弱性に対する検査を行うことができる．これらの脆弱性スキャンは自動で検査を実行することができるうえに，幅広い検査を行うことができるが，検査項目が豊富なため機器によっては必要ない項目の検査を行ってしまい，余計な時間がかかってしまうという問題がある．そのため，機器によって検査項目を指定して検査を行うと，利用者自身で項目を選択する必要があり，検査項目を選定するのが難しいという問題がある．

また，IoT システムを構成する製品の開発時，および導入時に検査すべき項目として，IoT セキュリティガイドライン [9]が公開されているが，同ガイドラインから機器毎に個別に具体的なチェック方法を考える必要があり，自動化が難しいため IoT 機器の脆弱性検査が全体に網羅されているような IoT 機器に特化した検査項目がなく，十分な検査ができているか，保証できなかった．

そこで本研究では，IoT システムを構成する製品の開発時および導入時に検査すべき項目として，JPCERT/CC から 2019 年 6 月に報告された，IoT セキュリティチェックリスト[10]を用いて，JPCERT/CC が推奨する脆弱性検査を自動化する手法を提案する．本手法を用いて，先行研究を参考に，ログイン操作は手動で行い，ログインに必要となる ID，パスワードの情報もユーザが知っていることを前提に，IoT 機器に特化した自動脆弱性検査システムを開発する．

3. 提案手法

脆弱性検査をするために，[10]に記載されている項目を自動で検査する機構を提案し，IoT 機器を用いた実験・評価を行う．[10]に記載されている検査項目の一部を表 1 に示す．このチェックリストは，IoT 機器を脅威の存在する環境においても，安全に運用するために，完備する必要がある 39 種のセキュリティ機能を必要な背景とともにまとめて，一覧表にしたものである．図 1 に提案機構の概要を示す．図 1 のユーザは IoT 機器の利用者もしくは開発者であり，ユーザが IoT 機器の IP アドレスを自動検査プログラムに入力することで，脆弱性検査が行われ，検査結果が自動で表示される仕組みになっている．

提案機構では，あらかじめ検査対象の IP アドレス，ログインページの URL やログインの操作手順等のシステムの設計情報を取得した上で検査を行う．また，検査は管理者権限を持つユーザが実行することを前提としている．

表 1 :セキュリティチェックリスト

Table 1　Security checklist.

I	1	ユーザ管理	アカウントロックアウトメカニズム
	2		一定期間利用されていないアカウントの強制失効オプション
	3		パスワード強度の担保機能
	4		パスワードセキュリティオプション（二要素認証など）
	5		サービスやプロセスを起動するアカウントの権限管理
	6		共有ユーザアカウント
	7		管理ユーザへの適切な権限付与
	8		一般ユーザへの権限付与機能
	9		認可制御機能
	10		サービス連携
II	1	ソフトウェア管理	Webアプリケーションファイアウォール
	2		製品に含まれるファイアウォール機能
	3		ソフトウェアバージョン
	4		ウィルス対策機能

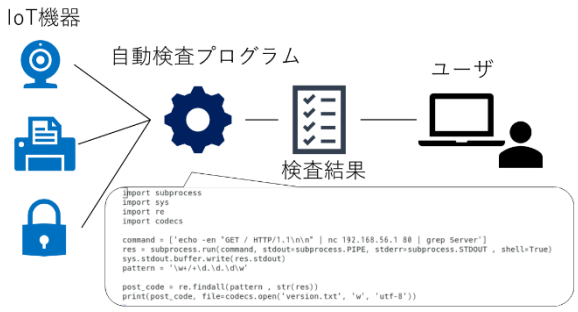

図 1:提案機構の概要

Figure 1　Overview of the proposed mechanism.

4. 実装

本研究では，IoT 機器の脆弱性検査を自動化するプログラムを作成し，検査結果を表示する自動脆弱性検査システムを開発した．本プログラムの開発環境は以下の通りである．

[ハードウェア]

・CPU : Intel(R)Core(TM) i7, 2.50GHz

・RAM : 1024MB

[ソフトウェア]

・OS : Windows 10 pro, Oracle VM VirtualBox Centos7

・プログラミング言語 : Python3.6,PHP7.1.28

また，検査対象は，自作のウェブサイトと，以下の５つの IoT 機器を対象に検査を行った．

・Canon プリンター(MF240 Series) [11]

・IODATA ネットワークカメラ(TS-WPTCAM2) [12]

・iRobot Roomba(e5) [13]

・amazon echo plus [14]

・Awair 空気品質モニター [15]

今回，検査対象として使用した IoT 機器はすべて無線接続の形で，同一の Wi-Fi(2.4GHz)ルーターに接続して，インターネットにアクセスしている．

今回はセキュリティチェックリストに記載されている項目の中で，検査対象の IoT 機器の IP アドレスや，システムの構成に関する情報で検査できる項目を対象に実装を行った．今回実装した項目を以下に示す．

4.1 ユーザ管理

(1)パスワード解析

John the Ripper [16]を用いてパスワード解析を自動で行い，結果を表示できるようにした．John the Ripper とは，パスワードクラックをするためのツールであり，提案機構では，脆弱なパスワードを使用していないかを検査するため，辞書に載っているパスワードを解読する word list mode を使用して検査を行う．

また，パスワード解析の際にログイン試行が複数回行われるため，脆弱なパスワードの使用を検査すると同時に，連続した規定回数以上のログイン失敗を確認した時に，検査対象がアカウントをロックし，一時的にログイン不可になる機能を持っているかを検査することができる．

(2)有効期限切れパスワード確認

lastlog コマンドを用いて最終ログインの記録を自動で表示できるようにして，存在するアカウントの最終ログインの記録から，一定期間利用されていないアカウントが確認できるようにした．

(3)Web アプリケーションファイアウォール利用の確認

検査対象の機器にログイン画面がある場合，ユーザ名の入力フォームに SQL インジェクションを行うシステムを自動化し，SQL 文を実行させることにより，検査対象が，SQL インジェクションの対策を行っているか，Web アプリケーションファイアウォールを使用しているかを検査することができる．

4.2 ソフトウェア管理

(1)製品に含まれるファイアウォール機能

検査対象の機器にファイアウォール機能が含まれる場合，ファイアウォール機能の状況が確認できる画面を自動で取得し，確認できるようにした．

(2)ソフトウェアのバージョン確認

脆弱性やバグ等に対応したバージョンであることを確認するために，検査対象が http 接続を可能とする場合，自動で http リクエストを行い，返ってくるレスポンスヘッダを取得することで，ソフトウェアのバージョンを確認することができる．

また，検査対象が http 接続を可能としない場合は，検査対象のソフトウェアのバージョンが確認できる画面を自動で取得し，確認できるようにした．

(3)データ転送量の確認

検査対象に向けて icmpflood を行うシステムを自動化し，検査対象のデータ転送量が DDoS 攻撃などを考慮した設計になっているかを確認できるようにした．検査対象の機器によってデータ転送量の設定は異なるため，一定時間に設定したデータ量のパケット転送を行い，応答時間の最短，最長，平均に加えて，どれだけの応答があるかを確認できるようなシステムを作成した．本研究における実験では，横浜国立大学・BBSSIoT サイバーセキュリティ共同研究プロジェクト [17]による，実際の IoT マルウェア(Bashlite)による DoS 攻撃を行った実験を参考に，DoS 攻撃の攻撃通信量を設定した．

提案機構では，icmp echo request の payload 長を 10byte（IP ヘッダを含めたパケットサイズは 38byte）とし，100 ミリ秒間隔で 10000 回送信し，検査対象からの応答を待たずに icmp echo request を送信して検査を行った．

4.3 セキュリティ管理

(1)ログ管理

インシデント発生時のログ情報を自動で表示し，検査対象が不正に操作されていないことや，異常な動作をしていないかを確認することができる．

(2)セッション管理

検査対象のシステムがセッション情報の管理に Cookie を使用している場合，検査対象に http 接続でアクセスしたときのヘッダにある Cookie の値を自動で取得することで，Cookie の値に secure 属性，HttpOnly 属性が設定されているかを確認することができる．

また，提案機構はログアウト前とログアウト後の Cookie の値を比較して表示している．これは，ログイン時や重要な確定処理の後に，新しいセッション ID が発行されているかを確認し，セッション情報を窃取された際の，機密情報の漏洩リスクを低減できる構造になっていることを確認するためである．

(3)SSL 証明書の取得

検査対象が SSL 通信を可能とする場合，利用している SSL 証明書を自動で取得し，確認する．提案機構では，openssl コマンドを用いて，検査対象の SSL 証明書を取得している．

4.4 アクセス制御

(1)リモートアクセス用ポートの確認

検査対象がリモート接続で外部から操作されないようにリモートアクセス用のポートがデフォルトポートになっていないことを確認する．提案機構では，後述するネットワークポートの開閉を確認する方法を自動で行う際に，リモートアクセス用のデフォルトポートである 22 番ポートが開いていないことを確認している．

4.5 不正な接続

(1)ネットワークポートの確認

検査対象のネットワークポートが利用用途を想定した適切なポートのみ開いているかを確認するために，ポートスキャンを自動で行うシステムを開発した．

(2)UPnP 設定

検査対象に UPnP の設定が可能な場合，検査対象の UPnP の設定が確認できる画面を自動で取得し，確認できるようにした．

4.6 暗号化

(1)通信の暗号化機能の確認

検査対象が SSL/TLS などを利用した暗号化通信を行っているかを確認する．提案機構では，4.3(3)より，SSL 証明書を取得できた場合，検査対象は暗号化通信を行っているとして判断している．

(2)暗号化方式・証明書の有効期限の確認

検査対象が利用している暗号化方式と証明書の有効期限を確認する提案機構では，openssl コマンドを用いて，利用している暗号化方式・証明書の有効期限を自動で取得し，表示することで確認できるようにした．

4.7 システム設定

(1)センサーの動作状況の確認

検査対象の機器にセンサーが作動している場合，センサーの動作状況が確認できる画面を自動で取得し，確認できるようにした．

4.8 通知

(1)セキュリティイベントのアラートと通知

last コマンドを用いて，システムのログイン失敗の履歴を自動で取得し，確認できるようにした．ログイン失敗の履歴は，検査対象の機器によって，通知先やログの保存先が異なるが，提案手法では，/var/log/wtmp 内にあるログイン履歴から，ログイン失敗の履歴を取得している．

5. 評価

プログラムを開発環境で実行し，自作の Web サイトと 5 種の IoT 機器を検査した．Roomba，amazon echo plus については，接続先クラウドが暗号化された通信を行っているかについて検査し，項目 4.6 に限定した検査を行った．

表 2 に提案機構による検査結果の例として，プリンターの検査結果を示す．

表 2:プリンターの検査結果

table 2　Inspection result of printer.

検査内容	該当項目番号	検査結果
ソフトウェアバージョン	Ⅱ-3	
証明書の有効期限	Ⅵ-4	Not Before: Jan 1 00:00:00 2012 GMT
		Not After : Dec 31 23:59:59 2037 GMT
暗号化方式	Ⅵ-3	New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
ネットワークポートの開閉	Ⅴ-1	80 open
		443 open
		515 open
		631 open
		8610 open
		9100 open
ログアウト前のセッションID	Ⅲ-2, Ⅲ-3, Ⅲ-4	#HttpOnly_192.168.100.120 FALSE / FALSE 0 view normal
ログアウト後のセッションID	Ⅲ-2, Ⅲ-3, Ⅲ-4	#HttpOnly_192.168.100.120 FALSE / FALSE 0 view normal
UPnPの設定	Ⅴ-2	
権限設定	Ⅰ-6	
ファイアウォール機能	Ⅱ-2	
センサーの動作状況	Ⅶ-1	
応答時間	Ⅱ-6	--- 192.168.100.120 ping statistics ---
		10000 packets transmitted, 661 received, 93% packet loss, time 1732ms
SSL証明書	Ⅲ-9, Ⅵ-1	-----BEGIN CERTIFICATE-----
		MIICuTCCAaGgAwIBAgIBATANBgkqhkiG9w0BAQsFADAgMR4wHAYDVQQDExVDYW5v
		biBJbWFnaW5nIFByb2R1Y3QwHhcNMTIwMTAxMDAwMDAwWhcNMzcxNjMxMjM1OTU5
		WjAgMR4wHAYDVQQDExVDYW5vbiBJbWFnaW5nIFByb2R1Y3QwggEiMA0GCSqGSIb3
		DQEBAQUAA4IBDwAwggEKAoIBAQDWGgK1ax0Q+tagG9W9bPDNkWN7Oklhk6+I2wFe
		8PChmFFYgu/6y2+yyDnZEUTn+XKX+r3zli5xrgv++TSR3aSQ2YCS6s+9X8qplxtS
		kct/EjsnukYBk9vsy6LOmYwwGwCwIYgsnce13HpiOyXAgm+uu1KOwigDXyWc+HCP
		QG3Oebe4Lo+4J57dndckE1FLSNFY9U7NWSh62e/3VcIiz61So+KV7JN9QPljzu7+
		WA9bSdWwCOADytRH5Q1Lu0ATaGUQBsq9MyuzWm1OYVZhUfGzedGS6VmGSM2iXq7X
		nCtEzk1m7VVzKUoaKGAeN4dJ6/g+EcQ7qYXYvCxIuMEp990VAgMBAAEwDQYJKoZI
		hvcNAQELBQADggEBAHCHh0YDW7Hq1dWTGpvloXr7Szs/BRLiK3Xz/eU9JCupIpnA
		Y/OAgkoCWjlF+4Brt7bw+0ClerEb5Bxu/yQraAc1JFeMsP9XFWJS5E1s6+FmzzLL
		hisA8TD+pd0fB96k/yN+0bi5oZCaqtrD7MhszCx64XlOeBL835bPrUTS/QRD+Yk7
		EQFs7jhX7XEmTZ8o2/FZ1eFm1qpUgVNIJtzAjXSd9d3bmNoJLdeZKN8lbU/9HjxM
		6QWCwgNeUKfrYniY2zdFsOo96c95I16f0ILyOIIAdjznHf+KZUUizFdFxvVYsPLz
		K1LxPkx4sNacpeym48AIjwNQKbICLmKom73I1BA=
		-----END CERTIFICATE-----

また，図 2~図 6 に表 2 で検査した項目を手動で検査した結果を示す．

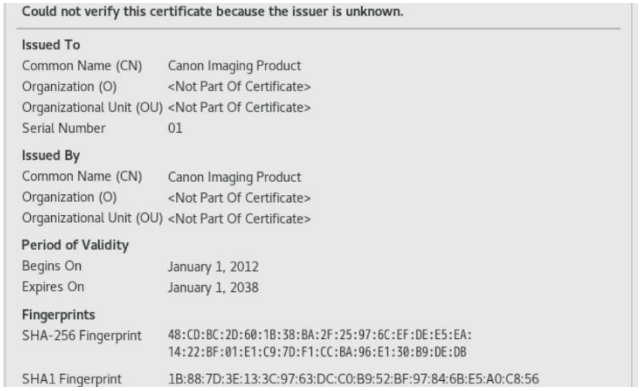

図 2:証明書の有効期限（手動）

Figure 2　Certificate expiration (manual).

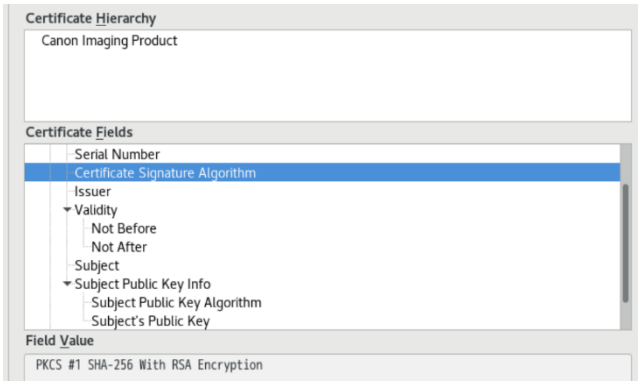

図 3:暗号化方式（手動）

Figure 3　Encryption method (manual).

```
[root@localhost html]# nmap -p 1-10000 192.168.100.120

Starting Nmap 6.40 ( http://nmap.org ) at 2019-08-02 03:11 JST
Nmap scan report for 192.168.100.120
Host is up (1.1s latency).
Not shown: 9994 closed ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
515/tcp  open  printer
631/tcp  open  ipp
8610/tcp open  unknown
9100/tcp open  jetdirect

Nmap done: 1 IP address (1 host up) scanned in 584.46 seconds
```

図 4:ネットワークポートの開閉（手動）

Figure 4　Open / close network port (manual).

```
[root@localhost html]# telnet 192.168.100.120 80
Trying 192.168.100.120...
Connected to 192.168.100.120.
Escape character is '^]'.
GET / HTTP/1.1
host: 192.168.100.120

HTTP/1.1 200 OK
Expires: Thu, 1 Jan 1998 00:00:00 GMT
Content-Type: text/html
Content-Length: 3543
Pragma: no-cache
Cache-Control: no-store, no-cache, max-age=0
Connection: close
Set-Cookie: view=normal;Version=;HttpOnly
```

図 5:セッション ID の属性（手動）

Figure 5　Session ID attribute (manual).

図 6:証明書の内容（手動）

Figure 6　Certificate contents (manual).

6. 考察

表 2 と図 2~図 6 の結果より，手動による検査の結果と，提案機構による IoT 機器の検査の結果が一致することから，脆弱性を正しく検査できるということがわかった．

しかし，表 2 に示したプリンターの検査結果(Ⅱ-3, Ⅴ-2, Ⅰ-6, Ⅱ-2, Ⅶ-1)には，内容が出力されなかった項目があるということが確認できた．これは，検査対象自体に検査項目の機能がない，もしくは，検査結果を確認できる記述が存在しないということが原因であると考えられる．ソフトウェアバージョン(II-3)については，セキュリティ対策として，サーバ情報を外部に公開しない設定になっていると考えられる．そのため，ソフトウェアバージョンが確認されなかった．また，今回検査したプリンターには UPnP・ファイアウォール・センサーの機能が製品に含まれていないこと(V-2, II-2, VII-1)，権限設定(I-6)では，ユーザの権限を確認できる機能がないことから，これらの項目は，検査結果が確認されなかった．

このことから，検査対象の機器によっては，セキュリティチェックリストに記載されているすべての項目を検査する必要はなく，利用用途を想定して検査する項目を決定する必要があると考えられる．

さらに，今回使用したセキュリティチェックリストには，自動で検査をすることが困難な項目があるため，今後の課題として以下に述べる．

・Ⅰ-10:サービス連携

本項目はログイン情報が，必要以上に他のサービスに渡らないようにすることを目的としており，検査対象の機器を他のサービスと連携して利用する際に，他のサービスに渡した情報が何かを利用者が確認できる機能があるかを検査する必要がある．

しかし，この検査を行うためには，連携先のサービスのシステムや，個々のサービスに渡す情報について，あらかじめ知る必要があり，一つの機器に対して，連携するサービスが膨大になると，検査を自動化するのは困難である．

・Ⅳ-1:管理されていない物理手段によるアクセス

本項目は，管理されていない物理手段によるアクセスを防ぐことを目的としているが，この検査を行うためには，検査対象の機器に対して，物理手段によるアクセスを行う必要があるため，検査を自動化することは困難である．

・Ⅳ-4:無線通信におけるセキュリティ(WPS)

本項目は，無線の設定ミスによるセキュリティの低下を防ぐことを目的としている．この検査に関してもⅣ-1 と同様に，WPS が動作するかを検査する方法が物理的であるため，自動化することは困難である．

・Ⅵ-1:データの暗号化機能

本項目は，データが平文で送られることにより，通信内容を読み取られることがないようにすることを目的としているが，データによって暗号化方式が異なることを考慮すると，検査対象の機器によっては，膨大な数の暗号化方式に対応した検査が必要となると考えられるため，すべての暗号化方式に対応すると考えると，自動化することは困難である．

これらの項目以外にも，今回実装した項目の中には，ssh などを利用して機器をリモートで操作することを前提に検査する項目が含まれているが，実際に今回検査対象とした機器は，ssh による接続が不可能ではあったが，外部からのリモート操作が可能な機器では，検査が可能になると考えられる．

また，本研究では，検査対象へのログイン方法や，製品に含まれる機能の設定画面の URL は，事前に分かっていることを前提として検査を行っているため，実際には，部分的に手動による事前情報の収集が，必要とされている．これは，機器によってログイン方法が異なることや，製品における情報の格納場所が異なることが原因と考えられるため，提案機構での脆弱性検査を行うためには，IoT 機器側の標準化も必要になると考えられる．

7. まとめ

本研究では，IoT システムを構成する製品の開発時および導入時に検査すべき項目として，JPCERT/CC から 2019 年 6 月に報告された，IoT セキュリティチェックリスト [4] を用いて，JPCERT/CC が推奨する脆弱性検査を自動化する手法を提案した．提案機構の有効性を示すために，自動脆弱性検査システムを開発し，5 種類の IoT 機器を用いて実験を行った．実験の結果，手動で脆弱性検査を行ったときの結果と比較して，機器のセキュリティ対策状況を定量的に把握することができた．

今後は，今回の評価結果で出力されていない項目に関しての検査項目の実装を行う．また，本研究の実験では，検査対象の機器にリモートアクセスができず，一部の実装に関しては，実機での実験ができなかったものがあるという問題があったため，リモートアクセス可能な機器での実験も行う必要がある．

参考文献

[1] “IoT のセキュリティ：IPA 独立行政法人 情報処理推進機構”. https://www.ipa.go.jp/security/iot/index.html, (参照 2019-06-25).

[2] “脆弱な IoT 機器及びマルウェアに感染している IoT 機器の利用者への注意喚起の実施状況 総務省”. http://www.soumu.go.jp/menu_news/s-news/01cyber01_02000001_00033.html, (参照 2019-08-10).

[3] “米国で「大規模 DDoS 攻撃」発生：Netflix，Twitter，Spotify がダウン”. https://wired.jp/2016/10/24/internet-down-dyn-october-2016/, (参照 2019-08-10).

[4] “IoT に関する取組 総務省”. http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd265230.html, (参照 2019-06-25).

[5] “セッション管理の脆弱性検査の自動化：慶応義塾大学大学院理工学研究所“. https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=77915&item_no=1&page_id=13&block_id=8, (参照 2019-0625).

[6] “OpenVAS - Open Vulnerability Assessment Scanner”. http://www.openvas.org/ ,(参照 2019-08-20)

[7] “Nmap security scanner”. https://nmap.org/ ,(参照 2019-08-20)

[8] “OWASP Zed Attack Proxy Project”. https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project,(参照 2019-08-12)

[9]“IoT セキュリティガイドライン 経済産業省”. https://www.meti.go.jp/press/2016/07/20160705002/20160705002-1.pdf ,(参照 2019-08-12)

[10] “IoT セキュリティチェックリスト:一般社団法人 JPCERT コーディネーションセンター”. https://www.jpcert.or.jp/research/IoT-SecurityCheckList.html, (参照 2019-06-27).

[11] “Canon MF244dw / MF242dwSatera MULTI FUNCTION PRINTER 概要”. https://cweb.canon.jp/satera/mfp/lineup/a4-mono/mf244dw-mf242dw/index.html, (参照 2019-08-20).

[12] “IO-DATA TS-WPTCAM2”. https://www.iodata.jp/product/lancam/lancam/ts-wptcam2/index.htm, (参照 2019-08-20).

[13] “irobot ルンバ e5 について”. https://www.irobot-jp.com/product/e5/index.html, (参照 2019-08-20).

[14] “amazon alexa”. https://alexa.amazon.co.jp, (参照 2019-08-20).

[15] “アウェア公式サイト : AWAIR Japan”. https://jp.getawair.com/, (参照 2019-08-20).

[16] “IoT セキュリティチェックリスト:一般社団法人 JPCERT コーディネーションセンター”. https://www.jpcert.or.jp/research/IoT-SecurityCheckList.html, (参照 2019-06-27).

[17] “横浜国立大学・BBSSIoT サイバーセキュリティ共同研究プロジェクト”. https://www.bbss.co.jp/business/service/pdf/YNU-BBSS_IoTSecPJReport.pdf, (参照 2019-07-29).