IoTセキュリティ評価のための チェックリストを使った取組み

JPCERTコーディネーションセンター
早期警戒グループ
輿石 隆

目次

- はじめに
- IoT関連のセキュリティ事例
- セキュリティに関する対応を行う上での課題
- IoTセキュリティ評価のためのチェックリストについて
- まとめ

はじめに

JPCERT/CC とは

- **一般社団法人 JPCERT コーディネーションセンター**
  Japan Computer Emergency Response Team / Coordination Center
  - コンピュータセキュリティインシデントへの対応、国内外にセンサをおいたインターネット定点観測、ソフトウエアや情報システム・制御システム機器等の脆弱性への対応など**日本の「セキュリティ向上を推進する活動」**を実施
  - **サービス対象: 日本国内のインターネット利用者やセキュリティ管理担当者、ソフトウエア製品開発者等（主に、情報セキュリティ担当者）**
  - インシデント対応をはじめとする、国際連携が必要なオペレーションや情報連携に関する、**日本の窓口となる「CSIRT」**
    ※各国に同様の窓口となるCSIRTが存在する
    (例、米国のUS-CERT, CERT/CC, 中国のCNCERT, 韓国のKrCERT/CC)
- 経済産業省からの委託事業として、**サイバー攻撃等国際連携対応調整事業を実施**

「JPCERT/CCをご存知ですか？」

JPCERT/CCの活動

インシデント予防 → インシデントの予測と捕捉 → 発生したインシデントへの対応

脆弱性情報ハンドリング

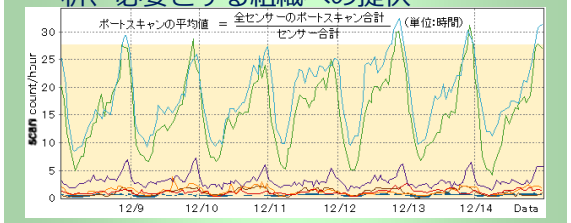
- 未公開の脆弱性関連情報を製品開発者へ提供し、対応依頼
- 関係機関と連携し、国際的に情報公開日を調整
- セキュアなコーディング手法の普及
- 制御システムに関する脆弱性関連情報の適切な流通

JVN Japan Vulnerability Notes

情報収集・分析・発信

定点観測（TSUBAME）

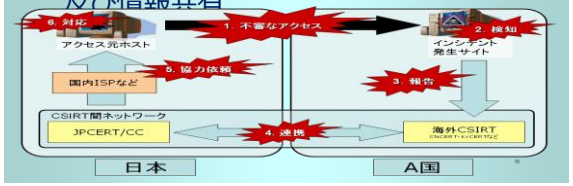
- ネットワークトラフィック情報の収集分析
- セキュリティ上の脅威情報の収集、分析、必要とする組織への提供



インシデントハンドリング

（インシデント対応調整支援）

- マルウエアの接続先等の攻撃関連サイト等の閉鎖等による被害最小化
- 攻撃手法の分析支援による被害可能性の確認、拡散抑止
- 再発防止に向けた関係各関の情報交換及び情報共有



早期警戒情報	重要インフラ、重要情報インフラ事業者等の特定組織向け情報発信
CSIRT構築支援	海外のNational-CSIRTや企業内のセキュリティ対応組織の構築・運用支援
制御システムセキュリティ	制御システムに関するインシデントハンドリング/情報収集,分析発信
アーティファクト分析	マルウエア（不正プログラム）等の攻撃手法の分析、解析
国内外関係者との連携	日本シーサート協議会、フィッシング対策協議会の事務局運営等
国際連携	各種業務を円滑に行うための海外関係機関との連携

サイバーセキュリティ・インシデントに沿った活動

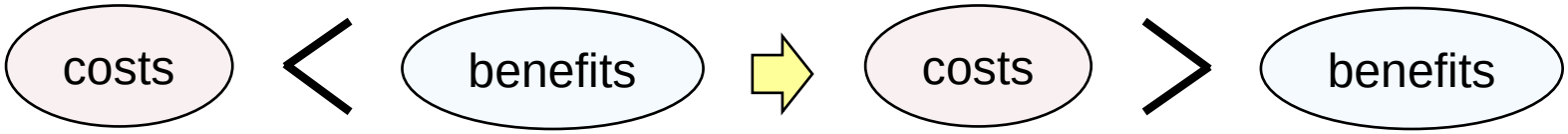
- **発生したインシデントへの対応 (主に CSIRT)**
  - インシデント初動対応への技術的な支援、助言
    例）インシデントレスポンス、アーティファクト分析など
  - CSIRT ：Computer Security Incident Response Team
- **製品の脆弱性への対応 (主に PSIRT)**
  - 届けられた製品の脆弱性についての調整、脆弱性情報の公開
  - PSIRT : Product Security Incident Response Team
- **事前、事後への対応**
  - 情報提供、情報共有、インターネット上の脅威への分析

制御システムセキュリティ分野 (ICS) についての活動も行っています

サイバー攻撃のリスクを下げる活動

■ 攻撃：コストとベネフィット

- 攻撃にもコストが必要 (攻撃インフラ、ツール等)
- **環境を改善しない限り攻撃は終わりません**



利益の方が大きい
→攻撃動機を助長

コストの方が大きい
→攻撃動機が減退

■ インシデントを防ぎ、サイバー攻撃のリスクを下げるには、攻撃のコストを上げていけるかが鍵

- 適切かつ迅速な初動活動・対応
- 正確な技術情報
- 普段からの脆弱性への対応 (作る側・使う側)

JPCERT/CCは、サイバー攻撃のリスクを下げるための活動を展開しています

攻撃のリスクを下げる取り組みの例

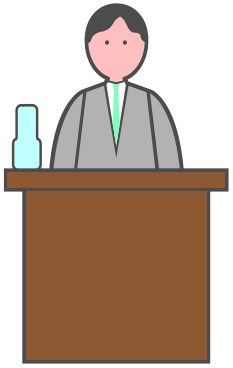
- **脆弱性情報への適切な対応 (主に事前)**
  - 報告者、開発者と脆弱性情報を適切に取り扱い、対応策、アップデート情報をJVNに公開
  - ユーザへの早期アップデート実施の呼びかけ
  - 脆弱性の低減方策の調査研究・開発、セキュアコーディングセミナー等による製品ベンダへの普及啓発
- **正しい技術情報に基づいた適切かつ迅速なインシデント対応 (主に事中・事後)**
  - アーティファクト分析に基づいた迅速な対応
- **その後の脅威へのモニタリングと集団防護 (主に事後)**
  - 継続的なモニタリングによる脅威の発見
  - 同じタイプの攻撃を防ぐため、グループでの情報共有

製品開発者にとって、脆弱性の存在は不可避

- **全ての攻撃の脅威に備えるのは不可能**
  - 新たな攻撃手法が出現

- **開発が大規模になればなるほど、既知の脆弱性対策の反映が困難**
  - 脆弱性情報の収集と取り纏め
  - 外部委託先での管理

- **製品に脆弱性が発見された場合、**
  ユーザに不安を与えず、冷静に対処してもらうことが重要



適切な情報公開のためのプロセスが存在

脆弱性対策情報ポータルサイト
JVN（Japan Vulnerability Notes）

公開日：2016/04/27　最終更新日：2016/04/27

JVNVU#91375252
Apache Struts2 に任意のコード実行の脆弱性

概要
Apache Struts2 には、任意のコードを実行可能な脆弱性が存在します。

影響を受けるシステム
- Struts 2.3.20 から 2.3.28 まで (Struts 2.3.20.3 および Struts 2.3.24.3 を除く)

詳細情報
Apache Struts2 には、Dynamic Method Invocation を有効にしている場合、任意のコードを実行可能な脆弱性が存在します。

なお、本脆弱性を使用した proof-of-concept コードが公開されています。

想定される影響
遠隔の第三者によって、当該製品が動作しているサーバ上で任意のコードを実行される可能性があります。

対策方法
アップデートする
開発者が提供する情報をもとに、最新版へアップデートしてください。
本脆弱性は Apache Struts 2.3.20.3、2.3.24.3、2.3.28.1 で修正されています。

国内外の脆弱性について記載
- JPCERT/CCで調整、公表
- US-CERTの注意喚起の翻訳
  など

- 開発ベンダに修正パッチなどの作成を依頼
- 脆弱性公表日などの調整

早期警戒パートナーシップ

脆弱性発見者 → 受付機関（IPA）：**届出**

受付機関（IPA） → 調整機関（JPCERT）：**通知**

調整機関（JPCERT） → ユーザ：**調整**

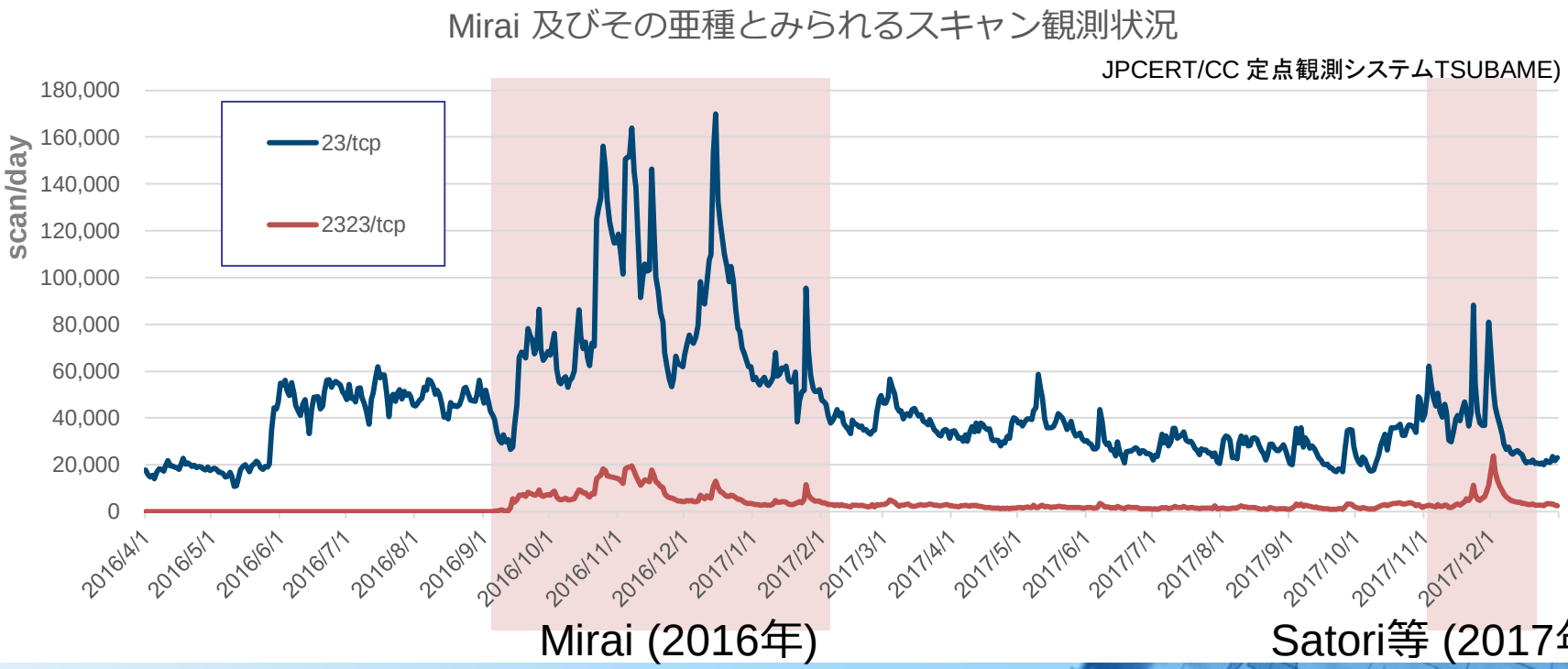
パッチができ、関係者合意の元 JVN発行

早期警戒パートナーシップに沿って脆弱性が扱われることで、未修正の脆弱性が公開されるのを防いでいる

IoT関連のセキュリティ事例

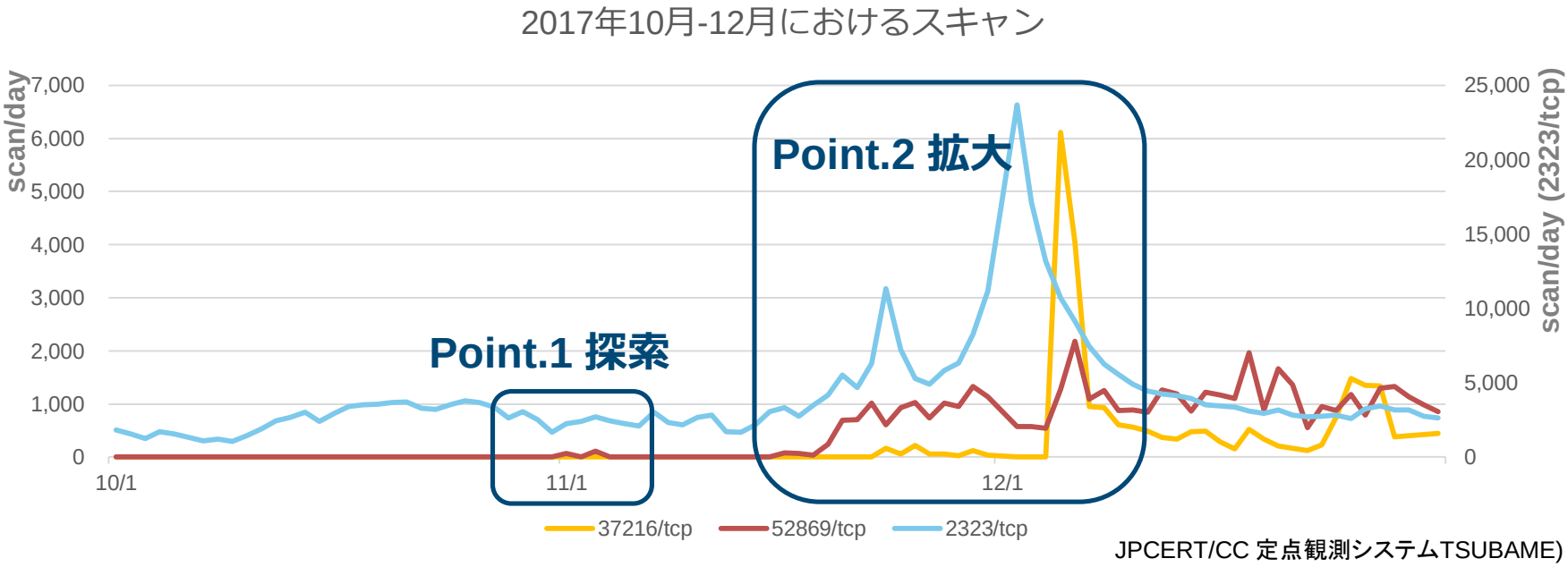
IoT botnet の問題

- **IoT botnet の脅威**
  - 2017年の脅威予測として、2016年末・2017年始に多く取り上げられていた問題
- **1Tbps 規模の DDoS は観測されなかったが、IoT botnet の問題は継続している**



Mirai 亜種の感染拡大の観測 (2017年12月)

- Mirai (2016年) → Satori等 (2017年) での変化
  - 従来、Mirai や Mirai 亜種の感染には、機器の管理コンソール (telnet等) におけるデフォルトの ID / PASSWD の悪用が主な拡大原因であった
  - 従来の脅威に加えて、脆弱性を悪用して感染を拡大させる
- 52869/tcp および 37216/tcp のアクセス増加



JPCERT/CC 定点観測システムTSUBAME)

感染拡大に悪用されている脆弱性

CVE-2014-8361 の悪用

Realtek SDK - Miniigd UPnP SOAP Command Execution (Metasploit), https://www.exploit-db.com/exploits/37169/

```
  register_options(
    [
      Opt::RPORT(52869) # port of UPnP SOAP webinterface
    ], self.class)
end

def check
  begin
    res = send_request_cgi({
      'uri' => '/picsdesc.xml'
    })
    if res && [200, 301, 302].include?(res.code) && res.headers['Server'] =~ /miniupnpd\/1.0 UPnP\/1.0/
      return Exploit::CheckCode::Detected
    end
```

- UPnP SOAP webinterface における OS command injection

Point1：SoC 開発元より提供されるファームウェア内に存在する問題

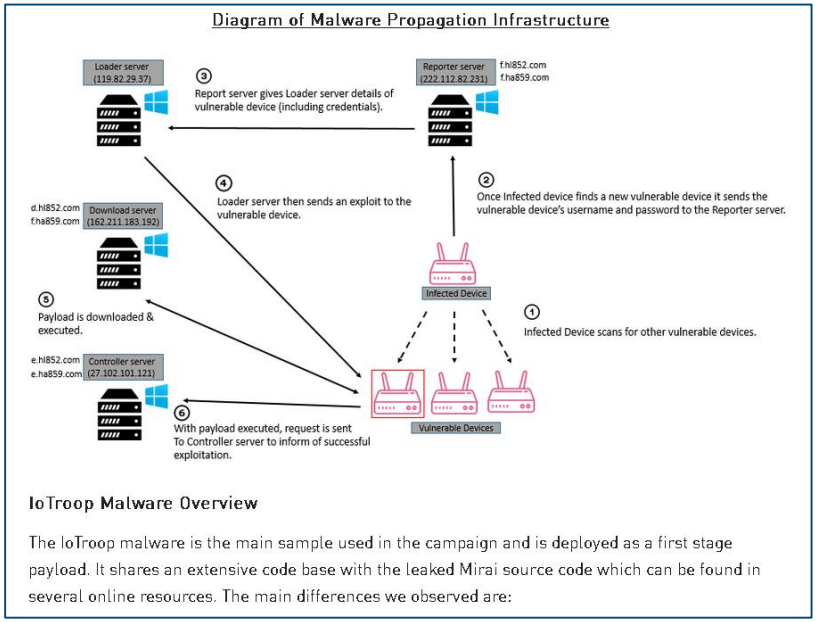
国内で感染が確認された機器

- ロジテック製 300Mbps無線LANブロードバンドルータおよび、セットモデル（全11モデル）
- 同製品に対するUPnP に対する 応答は、2014年に解消済み

Point2：過去の対策が行き届かなった機器が攻撃の影響を受けた

2017年におけるその他の IoT 関連 botnet の観測例

- **IoT reaper**
  2017年9月におけるクラウド上ハニーポット上における IoT reaper によるものとみられる脆弱性スキャンの観測
  - Mirai とは異なるマルウエア IoT reaper
  - 脆弱なパスワードを悪用するのではなく、**機器の既知の脆弱性を悪用**し感染を拡大 (2017年)
    - D-Link, Netgear, Vacron, Linksys, AVTECHなどの製品に対する脆弱性が挙げられている
  - Lua実行環境を内包
    - DDoS攻撃実行機能 (DNS amp攻撃) が確認されている



引用: Check Pont, IoTroop Botnet: The Full Investigation, https://research.checkpoint.com/iotroop-botnet-full-investigation/

2017年におけるその他の IoT 関連 botnet の観測例

IoT_reaperとみられる脆弱性スキャンの観測状況

- Web管理コンソールにおける脆弱性が悪用されている
- 認証なく遠隔からコードが実行可能
- 多くが OS コマンドインジェクション

脆弱性スキャンの一連の流れ

メソッド	URL	ヘッダ・クエリー	脆弱性が悪用される可能性のある製品
GET	/		
POST	/command.php	cmd = cat /var/passwd	D-Link DIR300/DIR600
GET	/system.ini		GoAhead HTTP server
GET	/upgrade_handle.php	uploaddir = ';echo nuuo 123456;' cmd = writeuploaddir	NetGear Surveillance
POST	/board.cgi	cmd = cat /etc/passwd	Vacron NVR
POST	/hedwig.cgi	cookie:uid=qDxppsreSd	D-Link 850L
POST	/apply.cgi		Linksys E1500/E2500
GET	/setup.cgi	todo=syscmd currentsetting.htm=1 cmd=echo dgn 123456 next_file=netgear.cfg curpath=/	NetGear DGN DSLモデム
GET	/cgi-bin/user/Config.cgi		AVTech IP cameras, DVRs and NVRs
GET	/shell		

2017年における IoT 機器への攻撃

- **脆弱性の悪用が加わりつつある**
  - 管理コンソールへのアクセスは依然として狙われているものの、既知の脆弱性を狙った攻撃が観測されている

- **対策は有効に機能しているか？**
  - ユーザが全員対策が可能とは限らない
    - アップデートなどの対策は、完全にはできない
  - IoT botnet は気が付きにくい問題
    - 「まさか、自分の機器が？」
    - 感染したからといって、目に見えた影響が発生しにくいため、どうしても感染に気が付きにくく、対策も遅れがち

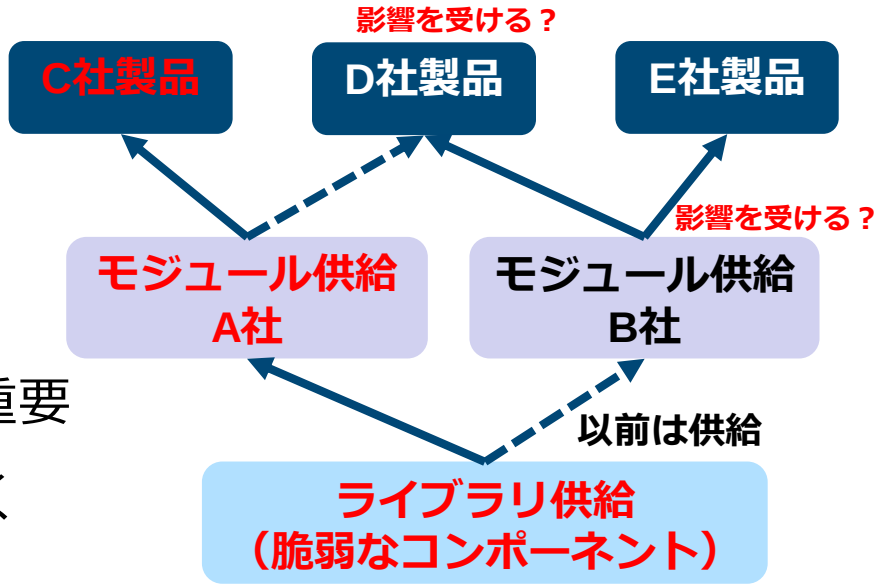
利用者だけでなく、開発者においても対策を施していくことが重要

セキュリティに関する対応を行う上での課題

脆弱性に対する製品の管理における課題

- **ファームウエアや、サードパーティのライブラリなどでの脆弱性をどう考えるか？**
  - アップデートに対する製造者にかかる責任/負担
  - 最終的な製品がどの脆弱性の影響を受けるのか、影響範囲がどこまでなのかといった脆弱性の管理が必要
  - 開発が大規模になればなるほど、既知の脆弱性対策の反映が困難
    - 脆弱性情報の収集と取り纏め
    - 外部委託先での管理

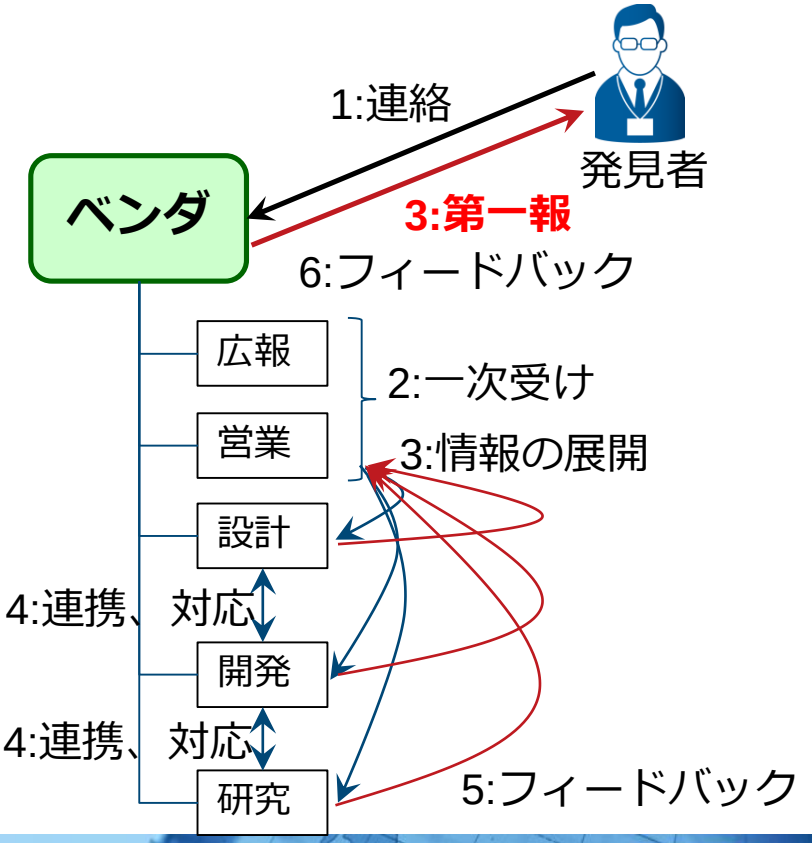
- **IoT では、影響が広い範囲に及ぶケースも考えられる**
  - 製品に脆弱性が発見された場合、ユーザに不安を与えず、冷静に対処してもらうことも重要
  - 業界全体で対応を検討していくことも必要



サイバー攻撃を受けることを前提とした対策の検討が肝要

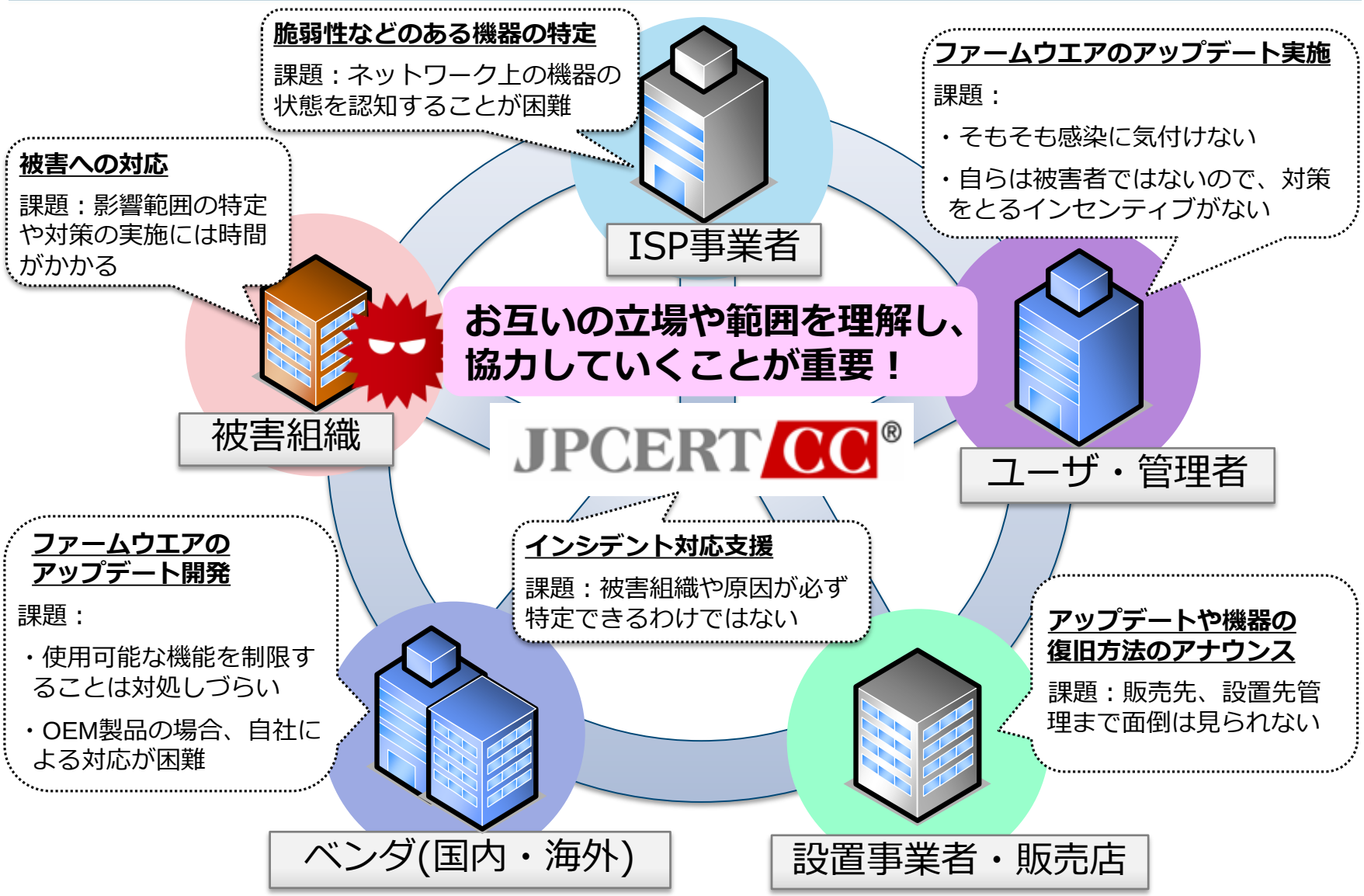
組織内での部門連携の問題

- セキュリティに関する情報が外部から届いたら・・・
  - セキュリティ担当者が直接外部からの情報を受け取るケースは少ない
  - 情報を受け取った人が適切な部署に情報を展開する必要がある
    - 情報のトリアージ
  - さらに内容によって組織内の様々な部門との連携が必要
    - 設計の見直し
    - コード修正、パッチ作成
    - テスト、リリース
    - 発見者への報告



対応完了前に発見者がセキュリティに関する情報(脆弱性の詳細やPoCコードなど)を公開してしまう場合もあるので慎重な対応が求められる

IoT・製品を取り巻くエコシステムにおける課題



IoT全般の課題｜1

- **IoT機器の課題**
  - 性能・機能の向上により“スマートな”機器に対する脅威はPCとほぼ変わらない
  - 設置された後、適切にメンテナンスされない可能性も考慮する必要がある

- **Internet of Threatsに関する課題**
  - IoTには、何かしらの“モノへの制御機能”が備わる
  - モノのスケールが大きくなるとCyber-Physical Threatな問題に発展する可能性がある
  - インターネットに“モノ”を直接接続することでセーフティに影響が及ばないかを考える必要がある
    - セーフティとセキュリティの両方の観点を考慮する必要がある

業界や関係者で情報共有を行うことの重要性

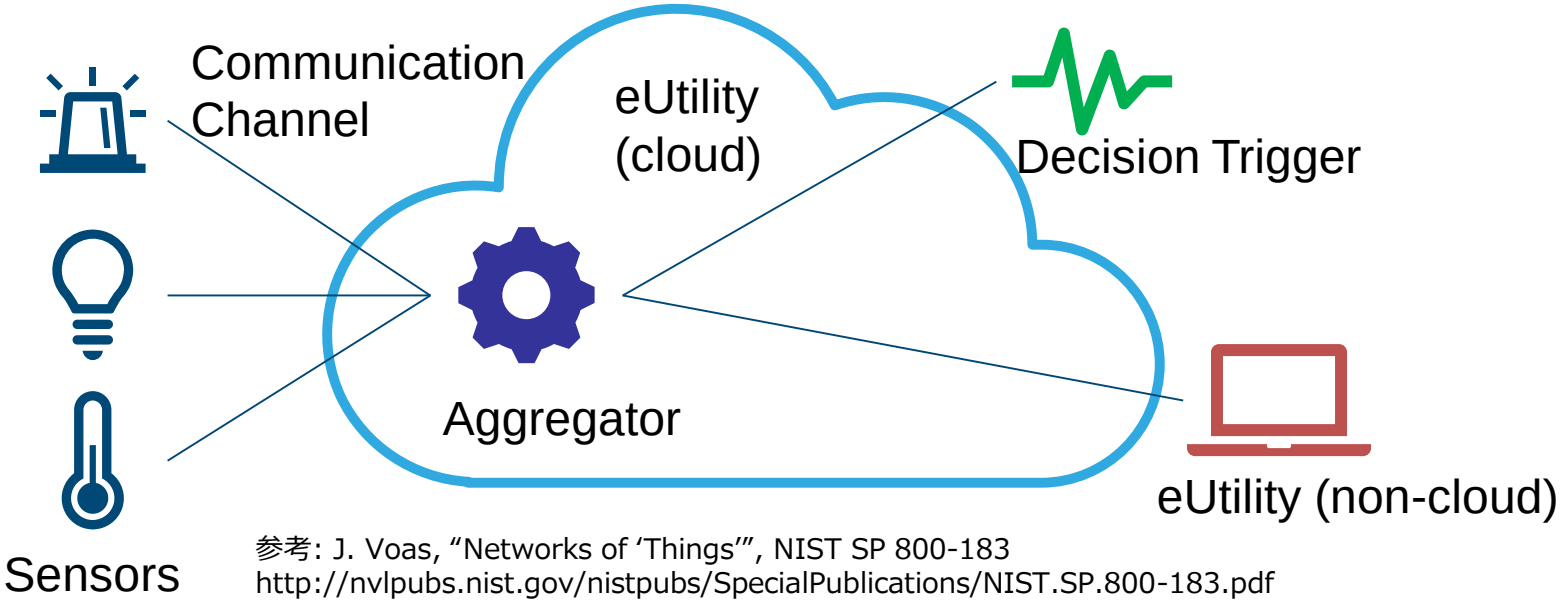
- **対策を進める上で、PSIRT の構築や、情報共有 (PSIRT間、同業他社、コミュニティ) がカギ**
  - 情報セキュリティの分野では、ISAC (Information Sharing and Analysis Center) による脅威情報の共有が進められている
  - 特に、脅威や影響などに関して共通の話題がある場合には情報共有は進みやすい

- **コミュニティでの取り組み例**
  - コンシューマ向けIoTセキュリティガイド
    http://www.jnsa.org/result/iot/
    - JNSA IoT セキュリティ WG にてとりまとめ (2016年)
    - 業界を横断して横のつながりで情報を整理
      - セキュリティベンダ、メーカ、その他

コミュニティでの活動がIoTセキュリティをとりまく環境を改善していく上で大きな役割を担うのではないかと考えられる

IoT全般の課題｜2

- **IoTという言葉で認識する対象範囲の違いによる課題**
  - IoTの問題は機器だけでなく、クラウド、サーバ、ネットワークなどシステム全体の問題である
  - 開発者はそれぞれの立場で自分と繋がる“何か”を意識し、影響を考慮する必要がある
  - 開発者 (ベンダ) 、利用者 (ユーザ) 双方がシステム全体の特徴を知る必要がある



参考: J. Voas, “Networks of ‘Things’”, NIST SP 800-183
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-183.pdf

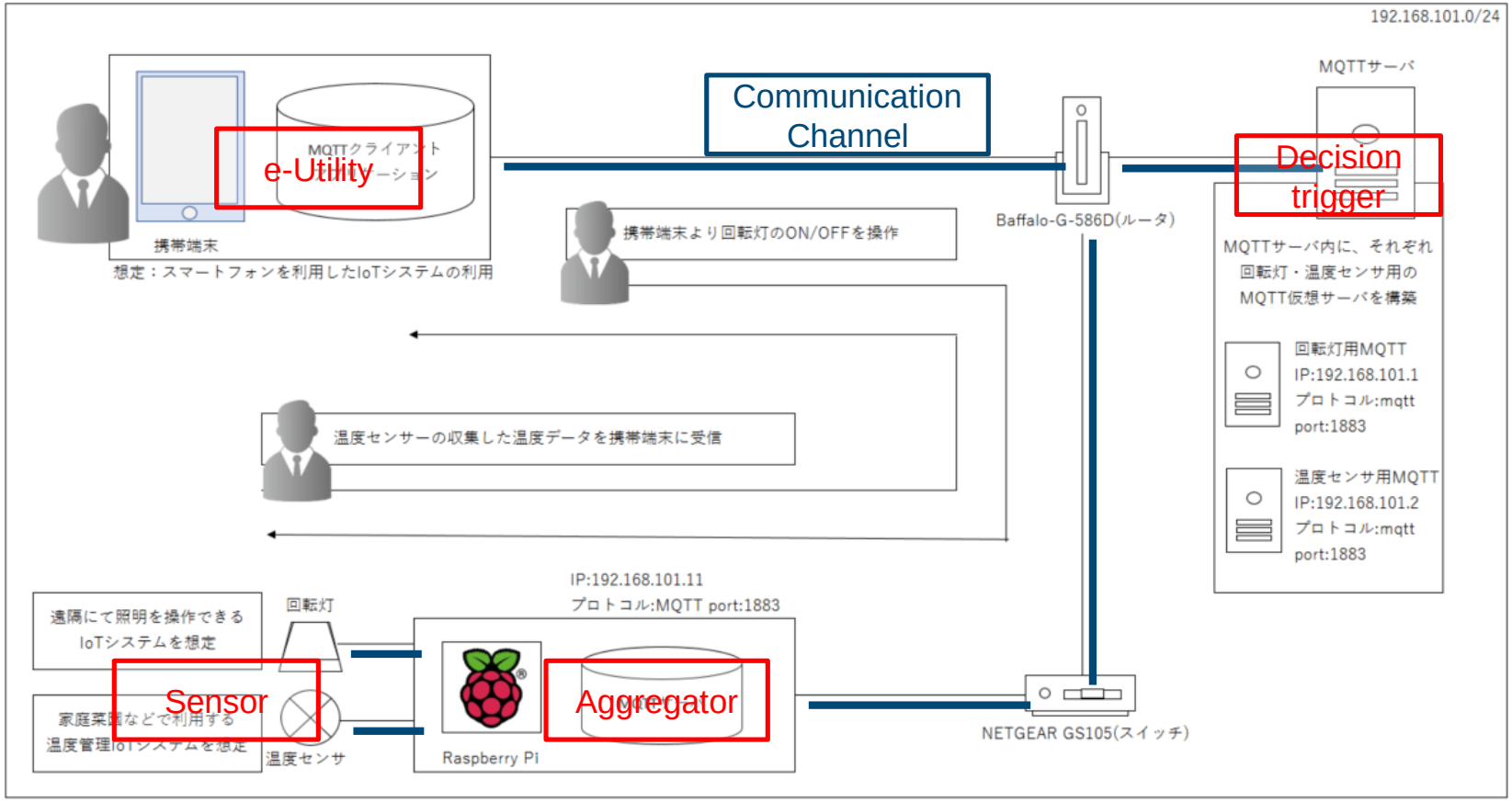
IoT の構成要素

- NIST SP 800-183 によると IoT は以下のように機能ごとに要素を分類できる
  - Sensor：温度、加速度、重量、音、位置などを測定するための機器
  - Aggregator：センサからのデータを集約して処理をする機器
  - Communication channel：データの送受信を行うための通信路
  - eUtility(external utility)：サービスを提供する／受ける機器
  - Decision trigger：データの加工や計算をするための機器

IoTシステムの構成要素ごとの特徴を把握し、それぞれの要素ごとに適切な実装を行っていく必要がある

IoT の構成要素

- 利用している IoT システムのパーツごとの機能を考えて、分類をしてみる



IoTセキュリティ評価のための チェックリストについて

IoT のセキュリティガイドライン

- **IoTに関するセキュリティガイドラインが多く公開されている**
  - IoT にまつわるサイバーでの問題が出始めた 2016 年ごろより国内外の複数の組織より IoT のセキュリティに関するガイドラインが公開されはじめた
  - 公開している組織により対象とする **想定読者**や**レイヤー**に違いがある

ポリシーレベルのガイドラインではなく、運用レベルにフォーカスしたドキュメントの作成を検討した

IoT セキュリティ評価のためのチェックリスト

- 運用レベルでの IoTの脅威と対策を理解するため、
  開発者・利用者双方が確認したい項目をなるべく具体的に
  まとめたチェックリストを作成中

	大項目	小項目	本項目の目的	開発する際に気を付けること	利用する際に気を付けること	チェック欄	理由	関連するコンポーネント S:Sensor、A:Aggregator、C:Communication Channel、E:e-utility、D:Decision trigger				
								S	A	C	E	D
1	ユーザ管理	アカウントロックアウトメカニズム	第三者が端末を不正に操作できないようにする	続した規定回数以上のログイン失敗や多重ログインなどの挙動を確認し、ら、アカウントをロックし、ログインが不可になる機能を持たせる	アカウントロックに関する設定可能な内容を確認し、自身で設定したとおりにアカウントがロックされるか確認				○		○	○
2		有効期限切れパスワードへの強制失効オプション	一定期間利用されていない アカウントからのログインをできないようにする	定した有効期限を超過したパスワードを失効させ、アカウントをロックする機能を持たせる	有効期限後にパスワードが失効することを確認する						○	○
3		パスワード強度の担保機能	ブルートフォース、辞書攻撃などにより不正にログインされないようにする	数の文字種の利用や一定文字数以上などの条件を満たしたパスワードの登録できる機能を持たせる	条件に満たないパスワードの登録ができないことを確認する						○	○
4		パスワードセキュリティオプション（2要素認証など）	第三者がシステムにログインすることを困難にする	スワードセキュリティオプションを利用できるようにする 例：2要素認証など）	パスワードセキュリティオプションが利用できることを確認する						○	○
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	アカウント等にサービスやプロセスを動かす権限を限定してインシデント発生時の影響範囲をサービスやプロセスの範囲内におさ	ービスやプロセスの起動にスーパーユーザを求めない作りにする	サービスやプロセス専用のユーザで動作することを確認する						○	○
6		共有ユーザアカウント	用途に応じて適切な権限を付与できるようにする	カウント管理機能を持たせる	共有するアカウントが適切な権限と共有範囲で利用されていることを確認する						○	○
7		管理ユーザへの適切な権限付与	管理ユーザが必要な権限を使えるようにする	限を管理	管理者							○
8		一般ユーザへの権限付与機能	ユーザに必要な権限のみを使えるようにする	ユーザに	ユーザ						○	○
9		認可制御機能	役割に応じたアクセス権を付与するできるようにする	カウントの役割に応じたアクセス権を付与する機能を持たせる	認可されていない						○	○
10		サービス連携	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	ービス連携時に他のサービスに渡す	他のサービスに渡した							○
11		Webアプリケーションファイアウォール	Webアプリケーションファイアウォールを利用できるようにする	ebアプリケーションファイアウォールを利用できるようにする	Webアプリケーションファイアウォールが利用できることを確認する						○	○
12	ソフトウェア管理	製品に含まれるファイアウォール機能	製品に含まれるファイアウォール機能を利用し、よりセキュアな状態にする	製品に含まれるファイアウォール機能を利用できるようにする	製品に含まれるファイアウォール機能が利用できることを確認する						○	○
13		ソフトウェアバージョン	脆弱性やバグ等に対応したバージョンのソフトウエアを利用し、セキュアな状態にしておく	フトウエアのアップデートを行う機能とバージョンを確認できる機能を持たせる	ソフトウエアのアップデートとバージョン確認ができることを確認する						○	○
14		ウィルス対策機能	製品に含まれるウイルス対策機能を利用し、よりセキュアな状態にする	製品に含まれるウイルス対策機能を利用できるようにする	製品に含まれるウイルス対策機能が利用できることを確認する						○	○
15		不正なデータ処理	システムが意図しない動作をしないようにする	受け付けるデータを制限する機能を持たせる							○	○
16		データ転送量	システムのデータ転送量はDDoS攻撃などを考慮した設計にする	受け付けるデータ転送量の制限を行うなどの機能を持たせる	受け付けるデータ転送量を確認する						○	○

チェック項目

開発する際に気を付けること

利用する際に気を付けること

対象構成要素

IoT セキュリティ評価のためのチェック項目

- 複数の IoT、ITのセキュリティのガイドラインを参考に確認すべき 7 つの大項目、39 のチェック項目を作成

大項目	チェック項目
ユーザ管理	アカウントロックアウトメカニズム
ソフトウェア管理	有効期限切れパスワードへの強制失効オプション
セキュリティ管理	パスワード強度の担保機能
アクセス制御	パスワードセキュリティオプション（2要素認証など）
不正な接続	サービスやプロセスを起動するアカウントの権限管理
暗号化	共有ユーザアカウント
システム設定	・・・
通知	

以下のようなガイドラインを参考に要素を抽出

- **IoT Security Guidance**
  - https://www.owasp.org/index.php/IoT_Security_Guidance
- **The Penetration Testing Execution Standard**
  - http://www.pentest-standard.org/index.php/Main_Page
- **The STRIDE Threat Model**
  - https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

チェック項目のポイント

- **全ての項目を確認する必要はない**
  - — 想定する利用シーンや目的によっては対策を実施しない項目もありうる
  - → 理由を明記し、要件を満たさなくてよい理由を関係者にわかるようにしておく

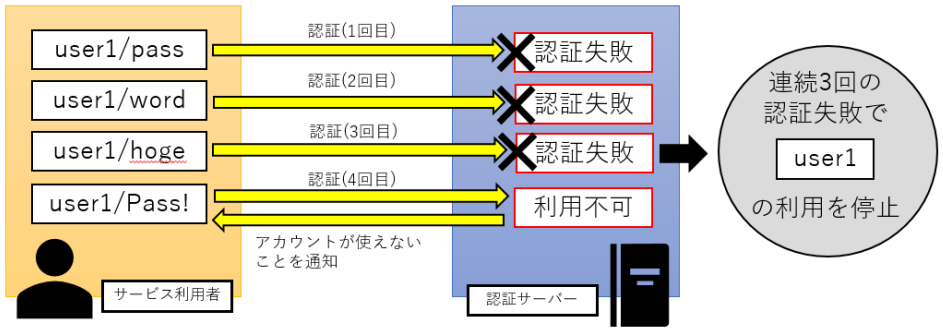
- **各項目についてイメージしやすいように解説図を添付**
  - — 確認する際にセキュリティに関する言葉がわからない場合がある
  - → 簡単な確認内容とその例を図で説明し、具体的なイメージを持てるようにしている

例：ユーザ管理 / アカウントロックメカニズム

1.アカウントロックアウトメカニズム

総当たり攻撃、辞書攻撃などを用いたID/PWの不正アクセスからアカウントを守るために
一定回数以上の認証失敗時に、その機器やサービスを利用できなくする仕組み

例)

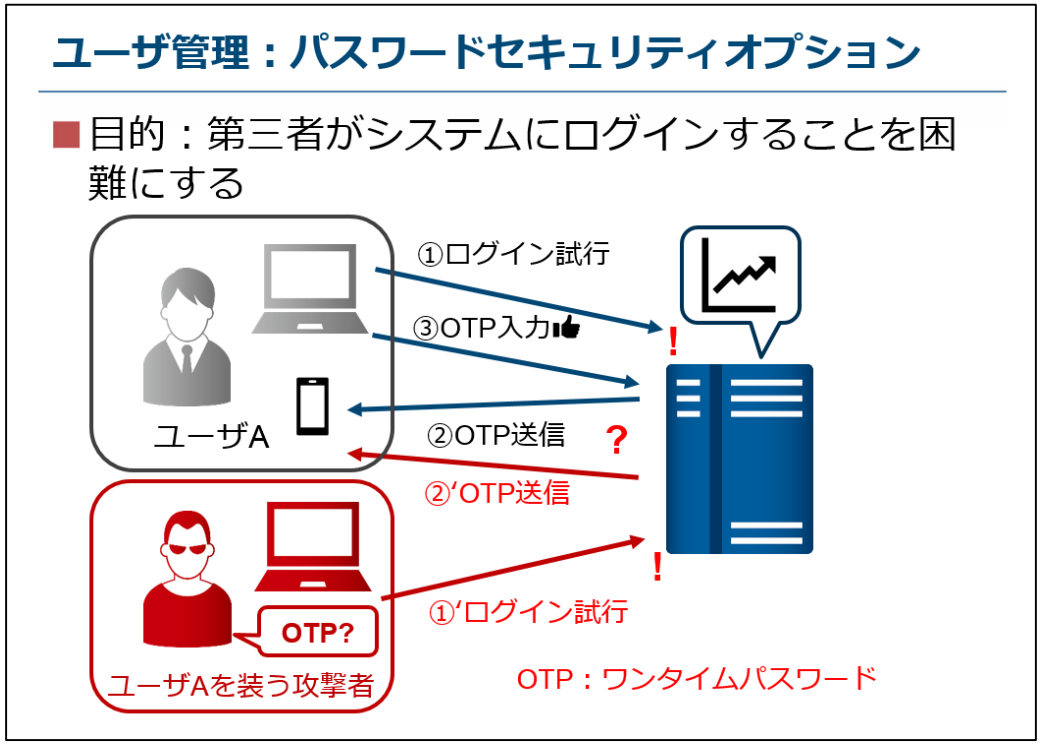


参考情報)

第三者が端末を不正に操作できないようにする

- **【開発の際に気を付けること】**
  連続した規定回数以上のログイン失敗や多重ログインなどの痕跡を確認したら、アカウントをロックし、ログインが不可になる**機能を持たせる**
- **【利用する際に気を付けること】**
  アカウントロックに関する**設定可能な内容を確認**し、自身で設定したとおりにアカウントがロックされるか確認する

例：ユーザ管理 / パスワードセキュリティオプション



第三者がシステムにログインすることを困難にする

- **【開発の際に気を付けること】**
  パスワードセキュリティオプションを**利用できるようにする**
  （例：二要素認証など）
- **【利用する際に気を付けること】**
  パスワードセキュリティオプションが**利用できることを確認する**

IoTを構成する要素ごとの評価

- デバイスだけを評価すればいいわけではない
  - NIST SP 800-183 に定義されている IoT の構成要素
    (Sensor, Aggregator, Communication Channel, e-utility, Decision trigger)
  - IoT の構成要素ごとに確認すべき項目を抽出

チェック項目
アカウントロックアウトメカニズム
有効期限切れパスワードへの強制失効オプション
パスワード強度の担保機能
パスワードセキュリティオプション（2要素認証など）
サービスやプロセスを起動するアカウントの権限管理
共有ユーザアカウント
・・・

本項目は次の構成要素となる製品が確認
- Sensor
- Aggregator
- e-utility
- Decision trigger

本項目は次の構成要素となる製品が確認
- Aggregator
- e-utility
- Decision trigger

活用

本チェックリストの対象者は

- **IoT の製品開発者**
- **ビジネスレベルの IoT の製品利用者**
  - **ある程度の IT などの知識が必要**

どんな場面で本チェックリストを使うのか

- **例：製品開発時の要件のチェック**

  開発工程の各段階の仕様確認の際に確認
  - **設計段階で特に確認したい項目を選んでおく**
    確認したい項目を満たしているか各工程の確認者が改めて確認をする

課題

- **公開に向けて作成したチェックリストについて意見を募集しています**
  - 現在の項目数に抜け漏れがないか確認をしたい
    - 今回作成した基本的な項目の他に足りないものがあるかもしれない
  - セキュリティの観点からリストを作成している
    - ITのセキュリティに関する知識と製品側のセキュリティ(セーフティ)に関する知識は大きく違うのかもしれない

まだまだ、多くの方に意見を頂いていく必要がございます。
チェックリストの評価など興味を持たれた方は、
ぜひ会話させてください

まとめ

2017年におけるIoTセキュリティの動向

インシデントに関する問題

- IoT botnetの問題は継続中
  コンソールへのアクセスだけでなく、製品の脆弱性を悪用するケース (IoT reaper) も観測されている
- botnet だけでなくフィッシングサイトやサイバー攻撃の踏み台にされている事例も観測されている

脆弱性に関する問題

- 仕様や実装の問題や特徴がよく考えられた指摘が続いた
- 現在はまだ、Web管理インターフェースにある脆弱性が悪用されているものが多いが、今後はより攻撃が複雑になっていく可能性がある

おわりに

- **インターネットに接続されたモノに対する脅威は増加している**
  - 脆弱性の問題の悪用にも注意
  - 脆弱性への対策は、製品の運用時だけでなく、製品を作る際においても考慮が不可欠
- **実際の攻撃では一瞬のスキを突かれてしまう**
  - アップデートまでの時間差
  - 仕様上の問題、実装上の問題
  - 共通ライブラリにおける脆弱性などの問題
- **IoT・製品を取り巻くエコシステムに対する課題**
  - インシデントを防止、軽減するためにも同業他社、業界を超えた協力が不可欠

セキュリティに関して何かございましたら、JPCERT/CCまでご一報ください！

お問合せ、インシデント対応のご依頼は

JPCERTコーディネーションセンター

- Email：pr@jpcert.or.jp
- Tel：03-3518-4600
- https://www.jpcert.or.jp/

インシデント報告

- Email：info@jpcert.or.jp
- https://www.jpcert.or.jp/form/

制御システムインシデントの報告

- Email：icsr-ir@jpcert.or.jp
- https://www.jpcert.or.jp/ics/ics-form.html

ご静聴ありがとうございました