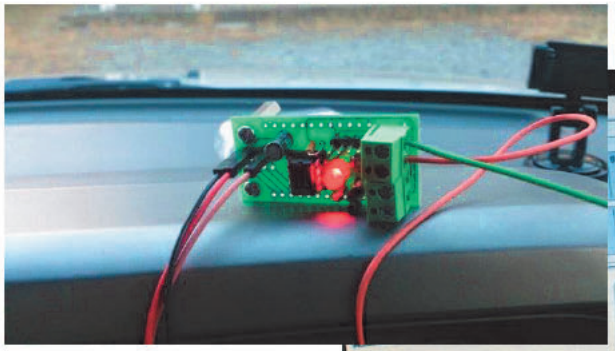
October 29, 2015
CODE BLUE 2015

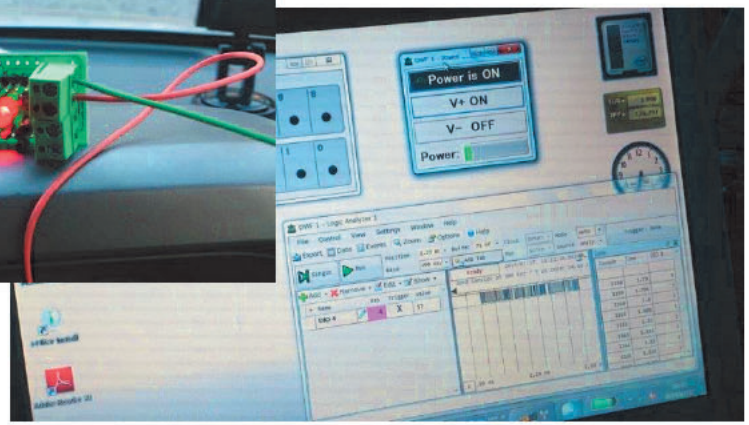
Windows 10 IoT Coreの脅威分析と実施すべきセキュリティ対策

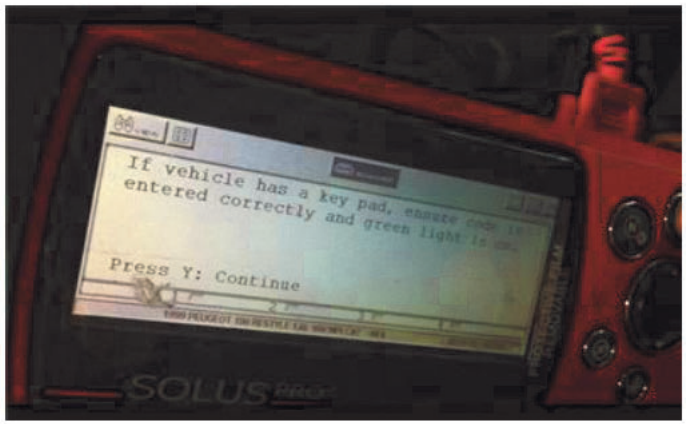
株式会社FFRI

自己紹介

- 元ネットワークエンジニアでマルチレイヤスイッチの品質評価やファームウェア開発（2007年くらいに IXIA を使った自動化やリグレッションテストの必要性を布教）
- 2013年に FFRI に入社し、 Type1 VMM を利用したドライバ保護システム開発や組み込み機器の検査、0-day 対策システムのプロトタイプを開発
- 2015年より現職で主に自動車セキュリティの研究を担当







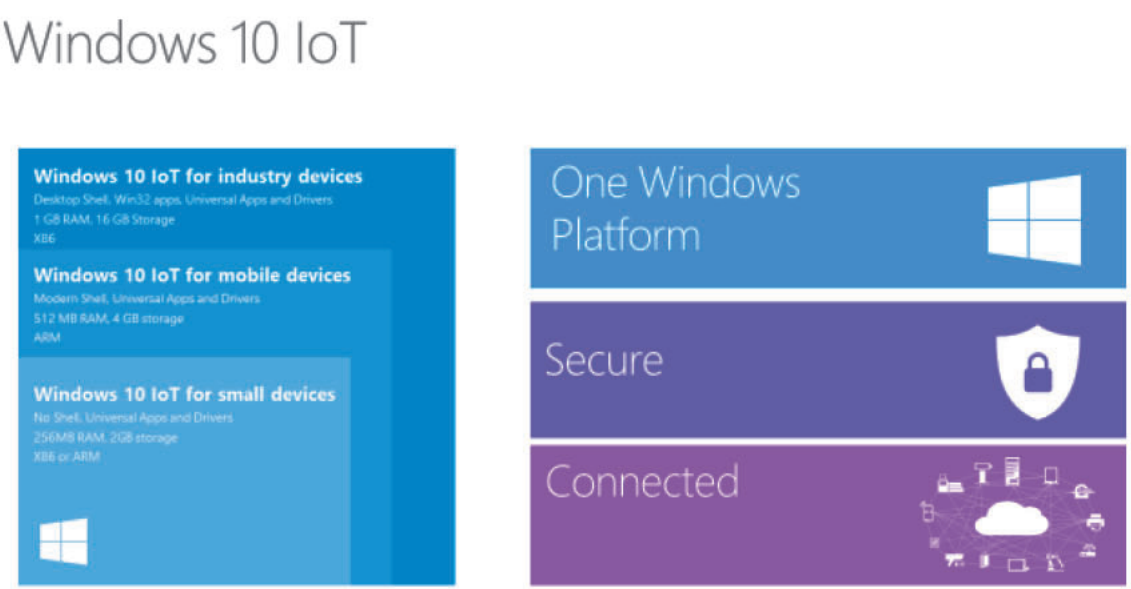
アジェンダ

- Windows 10 IoT Core 概要
- Windows 10 IoT Core の標準的なセキュリティ機能
- アタックベクタ調査 & 脅威分析
- Windows 10 IoT Core で行うべきセキュリティ対策
- まとめ

Windows 10 IoT Core 概要

Embedded から IoT へ

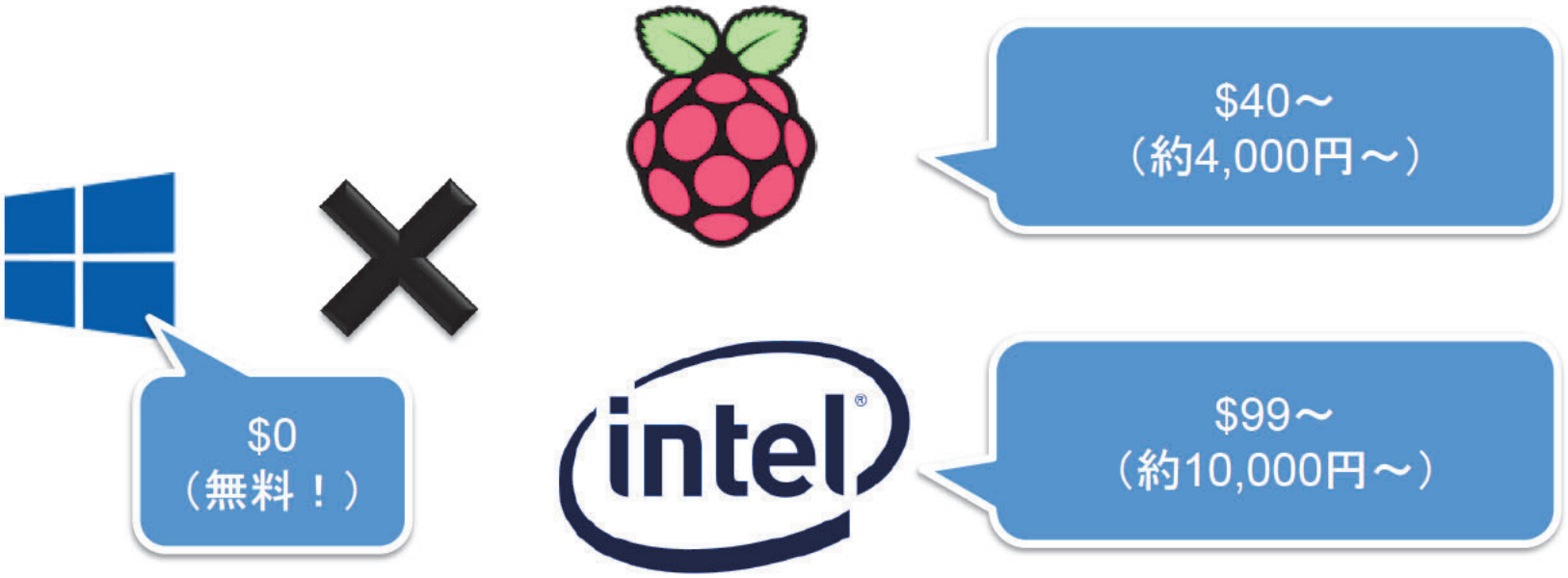
- Windows Embedded シリーズは、 Windows 10 の発売に併せて Windows IoT シリーズに一新された
- Windows 10 IoT Core はセンサーなどの小型デバイスをターゲットにしたシリーズ中最小構成のWindows



http://az648995.vo.msecnd.net/win/2015/03/IoT-1.png

Windows OSはSBCも対象に

- Windows 10 IoT Core は安価で人気のある Raspberry Pi 2 や Intel のMinnowBoard MAX のようなSBC(シングルボードコンピュータ)上で利用することができる（無償）



Windows 10 IoT Core の標準的なセキュリティ機能

Windows 10 IoT Coreとデスクトップ版の違い

Windows 10 IoT Core でもサポート

- DEP
- ASLR
- Control Flow Guard
  ※ビルド時に指定が必要
- Windows Firewall
  ※カスタマイズ前提

Windows 10 IoT Core では未サポート

- Windows Update
- Windows Defender
- User Account Control

Windows Updateが未サポート・・・。
既にリリースから2ヶ月経過しているが問題は
ないのか調査してみた。

マイクロソフトのセキュリティ情報（8～9月）

- Windows 10 が対象となるセキュリティ情報 ID
  - Windows 10 Systems
    - MS15-080, MS15-085, MS15-088, MS15-091, MS15-097, MS15-098, MS15-102, MS15-105
  - Microsoft .NET Framework
    - MS15-080(3.5), MS15-092(4.6), MS15-101(3.5/4.6)
  - Internet Explorer 11/Microsoft Edge
    - MS15-079, MS15-091, MS15-093, MS15-094, MS15-095

Windows 10 IoT Core に InternetExplorer と Edge は含まれないので除外

マイクロソフトのセキュリティ情報（8～9月）

- Windows 10 が対象となるセキュリティ情報 ID
  - Windows 10 Systems
    - MS15-080, MS15-085, MS15-097, MS15-098
  - Microsoft .NET Framework
    - MS15-080(3.5), MS15-092(4.6), MS15-101(3.5/4.6)
  - Internet Explorer 11/Microsoft Edge
    - MS15-079, MS15-091, MS15-093, MS15-094, MS15-095

Windows 10 IoT Coreでは
.NET Framework のサブセットである
CoreCLR が使用されているため、
直接的に影響はしないので除外

マイクロソフトのセキュリティ情報（8～9月）

- Windows 10 が対象となるセキュリティ情報 ID
  - Windows 10 Systems
    - MS15-080, MS15-085, MS15-088, MS15-091, MS15-097, MS15-098, MS15-102, MS15-105
  - Microsoft .NET Framew
    - MS15-080(3.5), MS1
  - Internet Explorer 11/Mic
    - MS15-079, MS15-091, MS15-093, MS15-094, MS15-095

MS15-102（タスク管理の脆弱性による権限昇格）のように Windows 10 IoT Core でも共通する可能性のあるパッチもあるが、アーキテクチャの違いなどから、確実に再現するとは言えない

アタックベクタ調査 & 脅威分析

ネットワークサービスの調査

- 今回はリモートからの攻撃可能性について調査をしてみた。
- ポートスキャンを使用して、TCP/UDPポートを調査した結果、いくつかのポートがデフォルトでオープンしていた。
- オープンしたポートの中でも、今回はもっとも攻撃対象になりやすいと考えられる FTP と リモートデバッグサービスに注目した。

ネットワークサービスの調査（cont.）

- デフォルトでオープンしていたポートと、該当ポートを使用している実行ファイルのコマンドラインは以下の通り。

ポートNo.	Nmapの判定	コマンドライン
21.tcp	ftp	ftpd.exe
22.tcp	ssh	C:¥windows¥System32¥svchost.exe -k SshSvcGroup
135.tcp	msrcp	C:¥windows¥system32¥svchost.exe -k RPCSS
445.tcp	microsoft-ds?	System
4020.tcp	trap?	C:¥RDBG¥msvsmon.exe /CHILDSERVER 188 "+:4020" {5D8A1EE3-3C96-4562-AD8A-8E4740A26577} 0x3 148 140 13c 144 /silent- /servicemode-
5985.tcp	wsman?	System
8080.tcp	http-proxy	System
9955.tcp 9955.udp	unknown	C:¥windows¥system32¥svchost.exe -k LocalService
47001.tcp	unknown	System

ネットワークサービスの調査（cont.）

- 今回注目したサービスは以下の用途での利用を目的としている。

ポートNo.	Nmapの判定	コマンドライン
21.tcp	ftp	ftpd.exe
22.tcp	ssh	
135.tcp	msrcp	
445.tcp	microsoft-ds?	System
4020.tcp	trap?	C:¥RDBG¥msvsmon.exe /CHILDSERVER 188 "+:4020" {5D8A1EE3-3C96-4562-AD8A-8E4740A26577} 0x3 148 140 13c 144 /silent- /servicemode-
5985.tcp	wsman?	
8080.tcp	http-proxy	
9955.tcp 9955.udp	unknown	
47001.tcp	unknown	

ＭＳの公式ページでは、スタートアップファイルの編集にＦＴＰを使用して解説している

Nmapでは trap の可能性を示唆しているが、実際には Visual Studio 2015 上でのリモートデバッグに使用される。タスクスケジューラに登録されていて、一定時間後に自動的に終了する

・・・FTPは認証不要？

- Nmapの実行結果を見ると、デフォルトで動作している FTP サービスは匿名でのログインが可能であることが分かる。

- バナー出力も、従来の Windows が提供しているものとは異なるため、バイナリを調べてみた。

```
Scanned at 2015-09-26 00:14:16 ???? (?W???) for 83s
PORT      STATE    SERVICE        REASON           VERSION
21/tcp    open     ftp            syn-ack ttl 128
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| d--------- 1 user group 0 Jul 10 13:13 CrashDump
| d--------- 1 user group 0 Jul 10 13:13 Data
| d--------- 1 user group 0 Jul 10 13:13 EFI
```

・・・FTPは本当の意味で認証不要だった

- ftpd.exe を調査した結果、そもそも認証ロジックがない！

```
SetUser
PUSH.W          {R11,LR}
MOV             R11, SP
LDR             R1, =a331UserNameOk_ ; "331 User name ok.\r\n"
POP.W           {R11,LR}
B.W             PostReply
; End of function SetUser
```

```
SetPassword
PUSH.W          {R11,LR}
MOV             R11, SP
LDR             R1, =a230UserLoggedI ; "230 User logged in.\r\n"
POP.W           {R11,LR}
B.W             PostReply
; End of function SetPassword
```

FTPサービスまとめ

- Windows 10 IoT Core の FTP サービスは認証機能を持たない
  - そもそも無いので後から認証させることも出来ない
- デフォルトではスタートアップファイルに記述されているため、デバイス起動時に FTP サービスも必ず起動する
- デフォルトのルートディレクトリは“C:¥”に設定されている
  - リモートデバッグ関連のファイル群が格納されている”C:¥RDBG¥”や”C:¥Windows¥System32¥”以下の一部のファイルを上書き可能

・・・リモートデバッグ“も”認証不要？

- FTP 同様に VS2015 から利用することの出来るリモートデバッグ機能も、デフォルトでは認証不要

- リモートデバッグに関する設定は、FTP 同様にスタートアップファイルに記述されていて、その中でセキュリティに対する設定が意図的に無効化されている

```
schtasks /create /f /tn "StartMsvsmon" /tr "%SystemDrive%¥RDBG¥msvsmon.exe
    /nowowwarn /noauth /anyuser /nosecuritywarn /timeout:36000" /ru
                DefaultAccount /sc onstart >nul 2>&1
```

Web UI

- Windows 10 IoT Core は一般的な IoT 機器に搭載されているようなWeb UI を標準で備えている
- FTP やリモートデバッグとは異なり、アクセスは Basic認証を経て行われるが、デフォルトでは HTTP による通信
- REST API による一部操作も可能
  - ドキュメントは /RestDocumentation.htm を参照
- Web UIではアプリのデプロイや、デバイスの一部設定が可能だが、今回推奨するようなセキュリティに関連する設定は一切出来ない

攻撃シナリオの調査

- Windows 10 IoT Core で考えられる脅威を、機密性（C）、完全性（I）、可用性（A）の3要素に基づいて検討してみた
- 加えて、これらの脅威を悪用するマルウェアについても考えてみた
- 最後に、必要な対策を検討する為に各脅威の関連性を示すダイアグラムを作成

機密性（C）に対する脅威

- Sniffing/Password cracking（or Steal）
  - Web UI は**デフォルトで HTTP 通信 + Basic 認証**を使用するため、通信を盗聴された場合パスワードが盗まれる可能性がある
  - 以前から問題視されている**ホームルーターに対する攻撃同様に、HTTP や SSH などのサービスに対するパスワードクラッキング**が試行される可能性がある
- Unauthorized access/Spoofing
  - セットアップ時にアカウント設定ウィザードなどが無い為、**ビルトインアカウントがデフォルトパスワードのまま運用される可能性**がある

完全性（Ｉ）に対する脅威

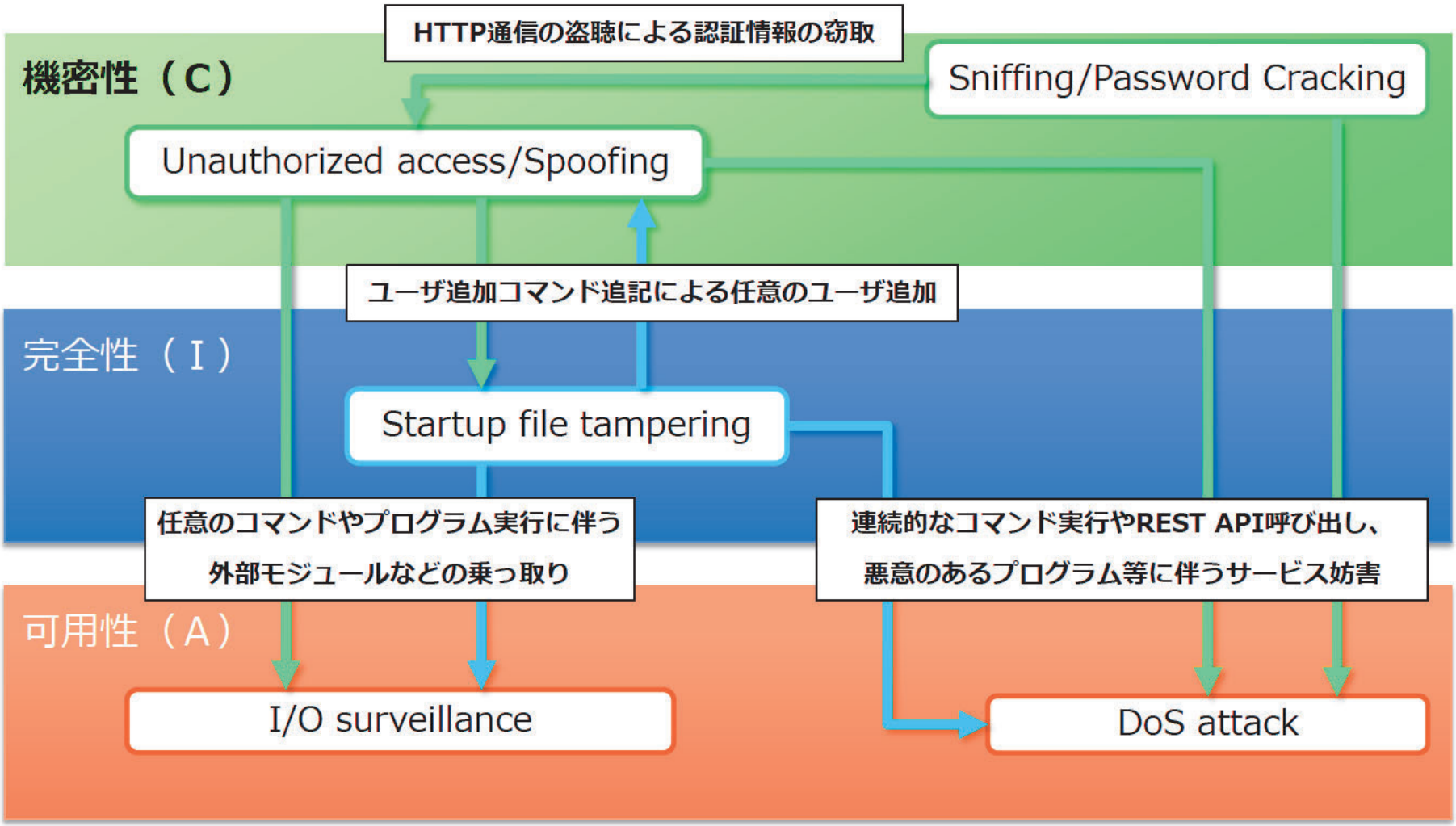
- Startup file tampering
  - デフォルト設定の FTP サービスを悪用することで、スタートアップファイルを書き換えることが可能
  - スタートアップファイルはバッチファイルとして実行されるため、”net use”コマンドで任意のユーザーが追加される可能性がある。

可用性（A）に対する脅威

- DoS Attack
  - パスワードクラッキングの試行や、連続的な REST API の実行は結果としてサービス妨害を誘発する可能性がある

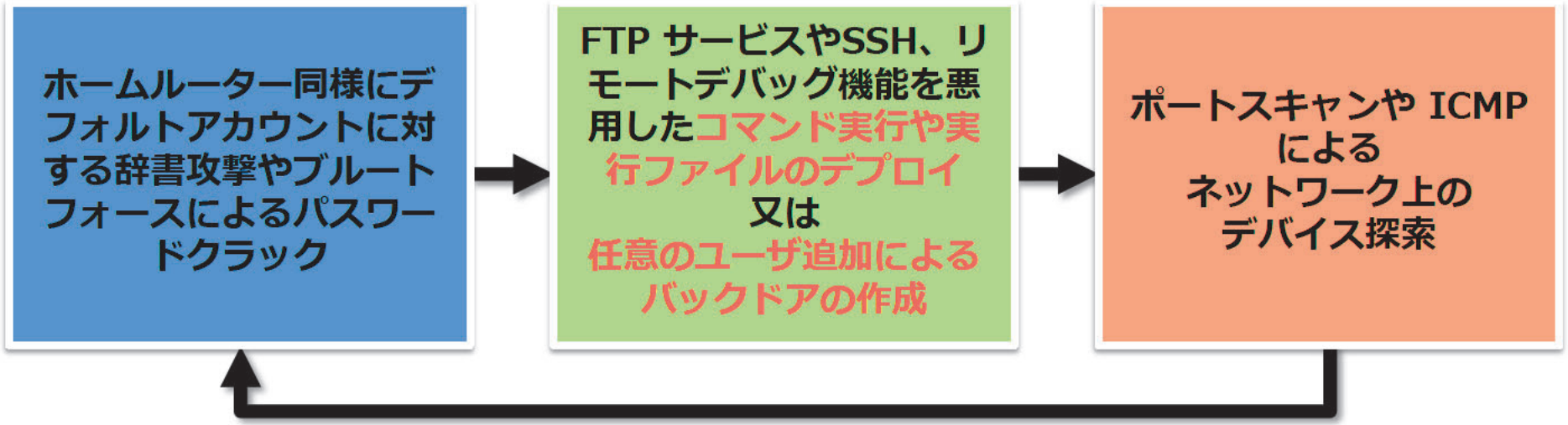
- I/O surveillance
  - 不正アクセスやなりすましは、結果としてデバイスに接続されているカメラモジュールの悪用（盗撮など）やセンサーの不正操作を誘発する可能性がある

脅威の相関関係


HTTP通信の盗聴による認証情報の窃取
機密性（C）
Sniffing/Password Cracking
Unauthorized access/Spoofing
ユーザ追加コマンド追記による任意のユーザ追加
完全性（I）
Startup file tampering
任意のコマンドやプログラム実行に伴う
外部モジュールなどの乗っ取り
連続的なコマンド実行やREST API呼び出し、
悪意のあるプログラム等に伴うサービス妨害
可用性（A）
I/O surveillance
DoS attack

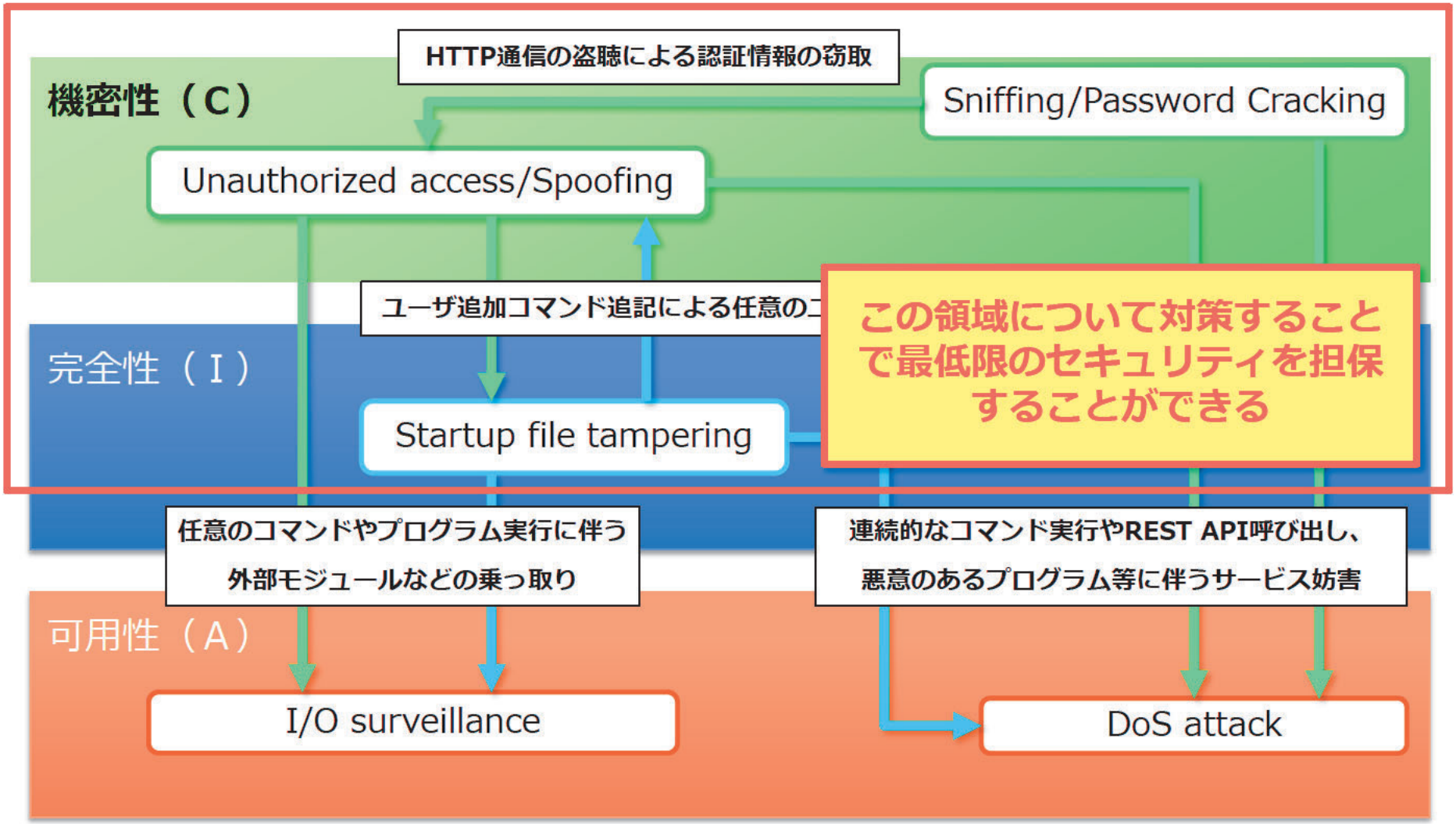
これらの脅威はマルウェアによって悪用される可能性大

- デフォルトのシステム構成における脅威が、ホームルーターなどの組み込み機器とだいたい同じである
  - デフォルトのアカウントで運用される可能性
  - Web UI への暗号化されないアクセス
  - 認証の無い FTP やリモートデバッグサービス
- そのため、侵入と感染を繰り返すワーム型マルウェアの標的になる可能性が非常に高い



Windows 10 IoT Core で行うべきセキュリティ対策

脅威の相関関係


HTTP通信の盗聴による認証情報の窃取
機密性（C）
Sniffing/Password Cracking
Unauthorized access/Spoofing
ユーザ追加コマンド追記による任意の
この領域について対策することで最低限のセキュリティを担保することができる
完全性（I）
Startup file tampering
任意のコマンドやプログラム実行に伴う外部モジュールなどの乗っ取り
連続的なコマンド実行やREST API呼び出し、悪意のあるプログラム等に伴うサービス妨害
可用性（A）
I/O surveillance
DoS attack

インストール後に最初にやるべきこと

Unauthorized access/Spoofing 対策

Sniffing/Password Cracking 対策

- パスワードを変更する
  - 不正アクセスやなりすまし対策として、インストール後は必ずSSH もしくは PowerShell 経由でビルトインアカウントのパスワードを変更する
  - 設定するパスワードはパスワードクラック対策として複雑なものを使用する

```
net user [username] [password]

ユーザーを追加する場合は:
net user [username] [password] /add
```

インストール後に最初にやるべきこと（cont.）

Sniffing/Password Cracking 対策

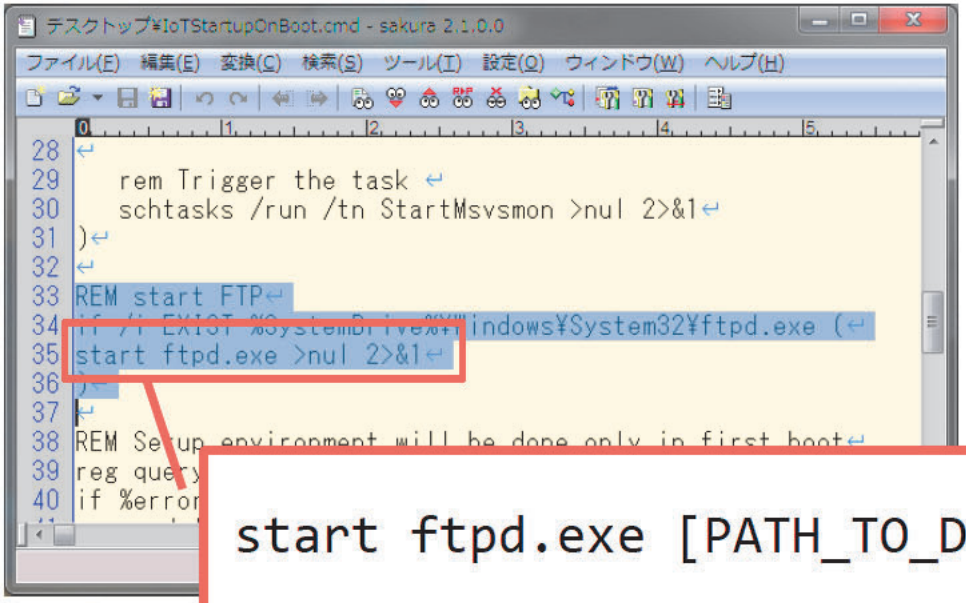
- HTTPS 通信の有効化
  - 盗聴による認証情報の窃取対策として、Web UI への HTTPS 接続設定を行う
  - 設定はレジストリの変更によって行うため、サービスもしくはデバイスの再起動が必要になる

```
Reg add
HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥IoT¥WebB /v
UseHttps /t REG_DWORD /d 1 /f
Reg add
HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥IoT¥WebB /v
HttpsPort /t REG_DWORD /d <your port number> /f
```

スタートアップファイルの編集

Startup file tampering 対策

- FTP を起動させない or ルートディレクトリの変更をする
  - IoTStartupOnBoot.cmd やその他の重要なファイル改ざん対策として、認証が不要な FTP サービスの起動設定をスタートアップファイルから削除するか、ルートディレクトリを変更する



start ftpd.exe [PATH_TO_DIRECTORY] >nul 2>&1 に変更

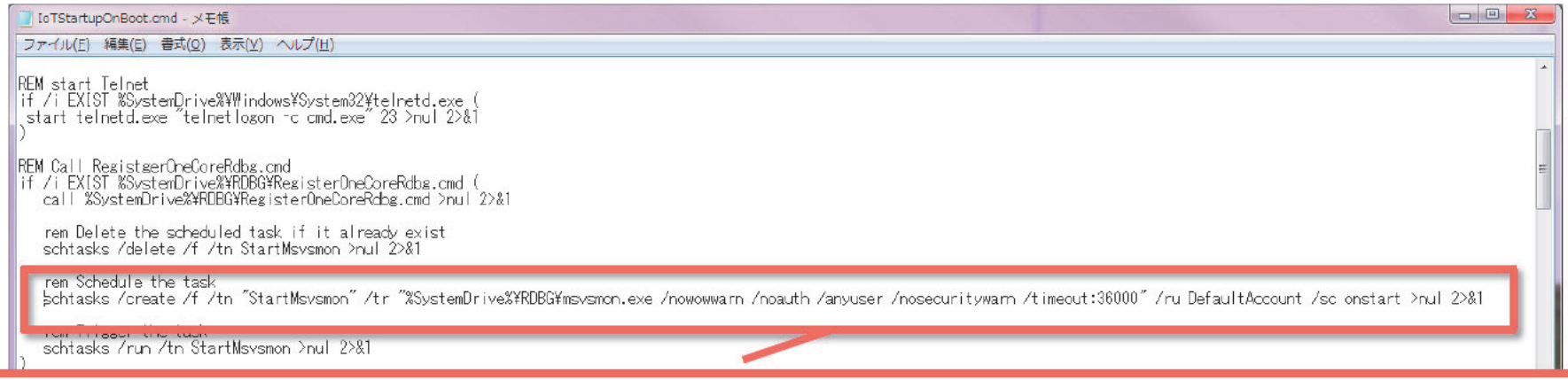
スタートアップファイルの編集（cont.）

Startup file tampering 対策

I/O surveillance 対策

DoS attack 対策

- リモートデバッグ時の認証を有効にする
  - リモートデバッグによって
    不正なプログラムを実行されてしまうのを防ぐ



%SystemDrive%¥RDBG¥msvsmon.exe /timeout:36000" /ru DefaultAccount /sc onstart >nul 2>&1に変更

Windows Firewall のルール設定

Unauthorized access/Spoofing 対策

DoS attack 対策

マルウェアによる探索 対策

- Windows Firewall のカスタマイズ
  - Inbound/Outbound 設定などの細かい通信許可/拒否設定が可能
  - スタートアップファイルに記述することも出来る
  - 例として以下に SSH 通信をブロックする簡易的な例を示す

```
Firewallの設定状態確認：
netsh advfirewall firewall show currentprofile

SSH(22)の通信をブロック：
netsh advfirewall firewall add rule name=[RULE_NAME]
protocol=TCP localport=22 action=block

設定を確認:
Netsh advfirewall firewall show rule name=[RULE_NAME]
```

まとめ

IoT向けのプラットフォームとして期待できると同時に、デスクトップ版のWindowsと同等に考えてはいけない

- デスクトップ版の Windows とは異なり、セキュリティ面でもユーザーがカスタマイズすることが前提となっているため、デフォルト設定のままインターネットにつなぐのは非常に危険

- 今回紹介した対策を最低限行うことが推奨されるが、設定が正直若干面倒くさい（特に Windows Firewall）上に情報量も Raspbian などの既存 OS にくらべ少ない

- 現在は、 Windows Update によるセキュリティパッチが自動的に適用されないため、将来的に既知の脆弱性が残ったままのデバイスが多く存在することになるかもしれない

最低限のセキュリティはプラットフォーム側で担保すべきでは？

- FTP や リモートデバッグはデフォルトで認証を有効にしておくべきで、認証の要否はユーザの設定によって選択できるようにするべき

- Raspberry Pi 2 はホビー用途で所有しているユーザも多く、すべてのユーザが認証不要で利用できる FTP やリモートデバッグサービスのリスクを理解しているとは限らない

- 組み込み用途である Embedded の系譜を継ぐ IoT シリーズであるであるとはいえ、Windows の名を関していることと、無償での提供よって多くのユーザに利用されることを期待するのであれば、最低限のセキュリティの担保は必要だと我々は考えている

ありがとうございました。

FFRI Inc.
http://www.ffri.jp