Monthly Research

Windows 10 IoT Core のセキュリティについて

株式会社ＦＦＲＩ
http://www.ffri.jp

Ver 1.00.01

はじめに

- 組み込み向けに最適化された Windows Embedded が Windows 10 の登場とともに Windows 10 IoT にリニューアルされる。

- 「Windows 10 IoT Core Insider Preview」というプレビュー版が既に Raspberry Pi 2 を始めとする小型マイコンボード向けに提供されている。

- 今回は Windows 10 IoT Core for Raspberry Pi 2 を用いたセキュリティ検証のチュートリアルを紹介する。

※本レポートの記載内容はプレビュー版を対象としているため、
正式版 Windows 10 IoT とは異なる可能性がある。

Windows 10 のエディションによる違い

- Windows 10 は一般向けの Home や、仮想化技術の Hyper-V、暗号化機能の BitLocker などの機能が追加された Pro など下記の 7 種類のエディションが存在する。
  - Home, Mobile, Pro, Enterprise, Education, Enterprise Mobile, Windows 10 IoT

- Windows 10 Mobile と Windows 10 IoT には ARM, x86/x64 版がある。

- Windows 10 IoT は今後開発される IoT デバイスをターゲットにしており、一般向けエディションとの違いは、フットプリントの小ささとハードウェアの制御に用いる GPIO を操作する Windows.Devices API が利用可能な点である。

Windows 10 IoT のエディションについて

エディション	概要	要求スペック
Windows 10 IoT for Industry Devices	X86/x64 プロセッサで動作 デスクトップシェルを搭載し 比較的高機能	メモリー：1 GB ストレージ：16 GB
Windows 10 IoT for Mobile Devices	Windows Embedded Handheld の後継 ARM 32bit プロセッサで動作し、 モダンシェルを搭載	モバイル端末 メモリー：512 MB ストレージ：4 GB
Windows 10 IoT for Small Devices / Windows 10 IoT Core	x86, ARM 32bit で動作 より軽量な小型デバイスの 制御用軽量 OS シェルなし	メモリー：256MB ストレージ：2GB

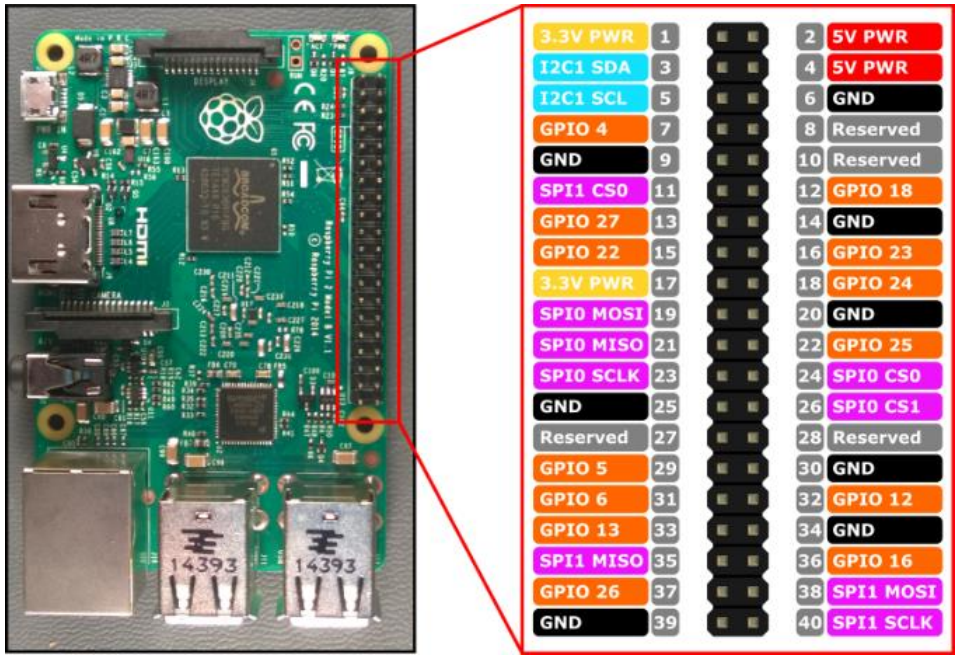
Windows 10 IoT 対応マイコンボード

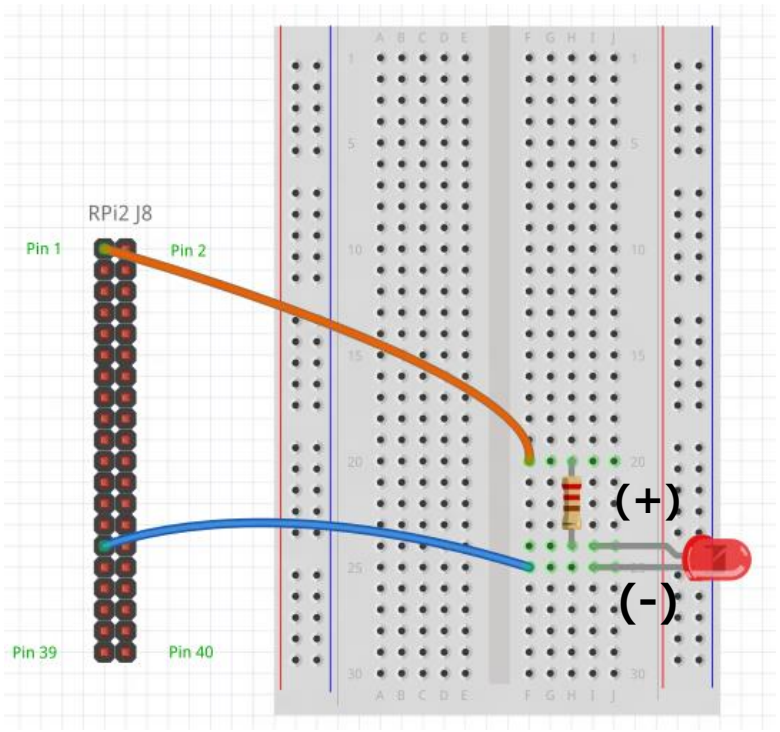
名称	CPU	RAM
Raspberry Pi 2	900MHz quad-core ARM Cortex-A7 CPU	1GB
MinnowBoard Max	Atom E3815-1.46GHz/E3825-1.33GHz	1 or 2GB
Galileo	Intel® Quark™ SoC X1000 (16K Cache, 400 MHz)	256 MB
Windows Remote Arduino	ATmega2560 16 MHz	256 KB
Windows Virtual Shields for Arduino	ATmega328 16 MHz	32 KB

- 人気のマイコンボードである Raspberry Pi 2 に対応しており普及する可能性がある。
- C++, C#, Python を用いてアプリケーションの開発を行うことができる。

Raspberry Pi 2 を使った LED 点滅

- Windows 10 IoT Core を用いて Raspberry Pi 2 の GPIO を制御し LED を点滅させる。





ms-iot.github.ioより

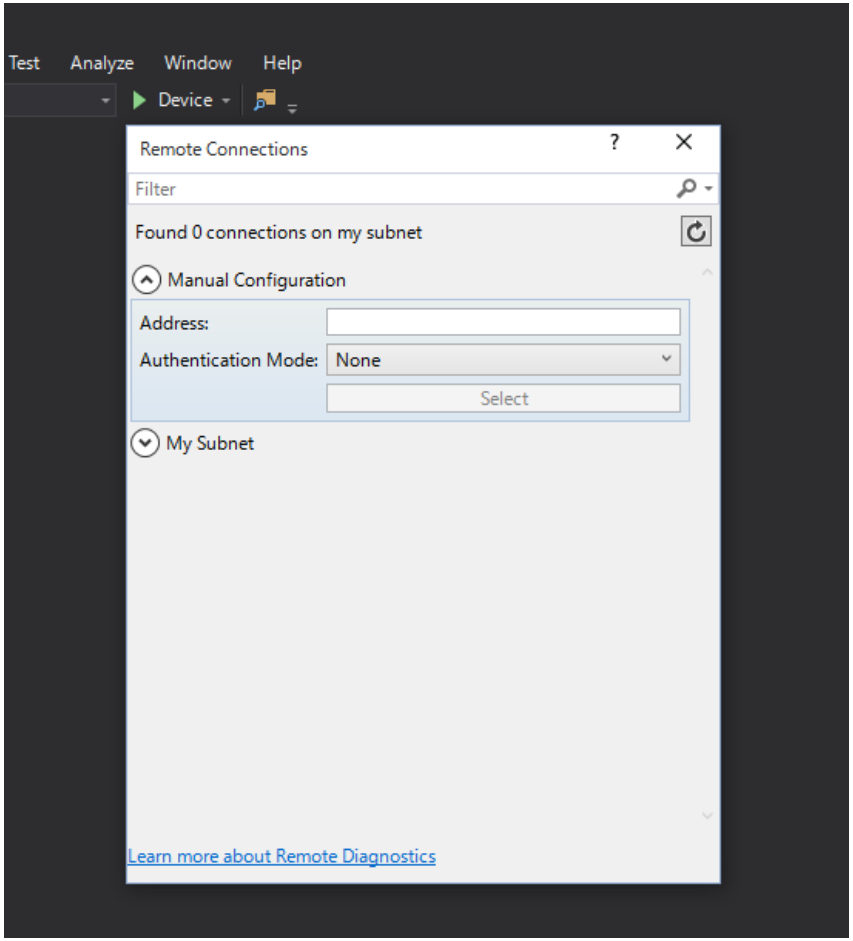
Windows 10 IoT Core でのプログラム実行手順

- Python を用いて LED を点滅させる手順について解説する。
  1. PC に Visual Studio 2015 (本レポート作成時の最新版は Visual Studio 2015 RC, 以降 VS2015) をインストール
  2. PC に Python 3.x, PTVS (Python Tools for Visual Studio), Python UWP(Universal Windows Platform) SDK をインストール
  3. 前スライドで示した図のように**赤色 LED, 200Ωの抵抗、ブレッドボード、ジャンパワイヤ**を組み立て、プラスを 3.3v の PIN1, マイナスを GPIO の PIN5 に接続
  4. Raspberry Pi 2 を PC と同一のネットワークに接続し、電源 ON
  5. PC で VS2015 とともにインストールされた Windows 10 IoT Core Watcher を実行し、Raspberry Pi 2 の IP アドレスを確認
  6. VS2015 上で 9 ページに示す Python スクリプトを作成し、 Raspberry Pi 2 の IP アドレスを指定して実行

VS2015 からのコード実行

- VS2015 のツールバーの「Device」ボタン右側のプルダウンから「Remote Device」を選択すると、対象デバイスの IP アドレスと認証モードの設定ができる。

- デフォルトでは「Authentication Mode」を「None」(認証を行わない設定)で問題なくコードを実行する事ができた。



LED を点滅させる Python スクリプト

```
import _wingpio as gpio       // GPIOコントロール用のモジュールをロード
import time

led_pin = 5                   // GPIOのPIN番号を指定
ledstatus = 0

gpio.setup(led_pin, gpio.OUT, gpio.PUD_OFF, gpio.HIGH)

while True:
    if ledstatus == 0:
        ledstatus = 1
        gpio.output(led_pin, gpio.HIGH)      // 指定したGPIOのPINをHIGHに設定(点灯)
    else:
        ledstatus = 0
        gpio.output(led_pin, gpio.LOW)       // 指定したGPIOのPINをLOWに設定(消灯)

    time.sleep(0.5)    // 点滅の間隔を指定

gpio.cleanup()
```

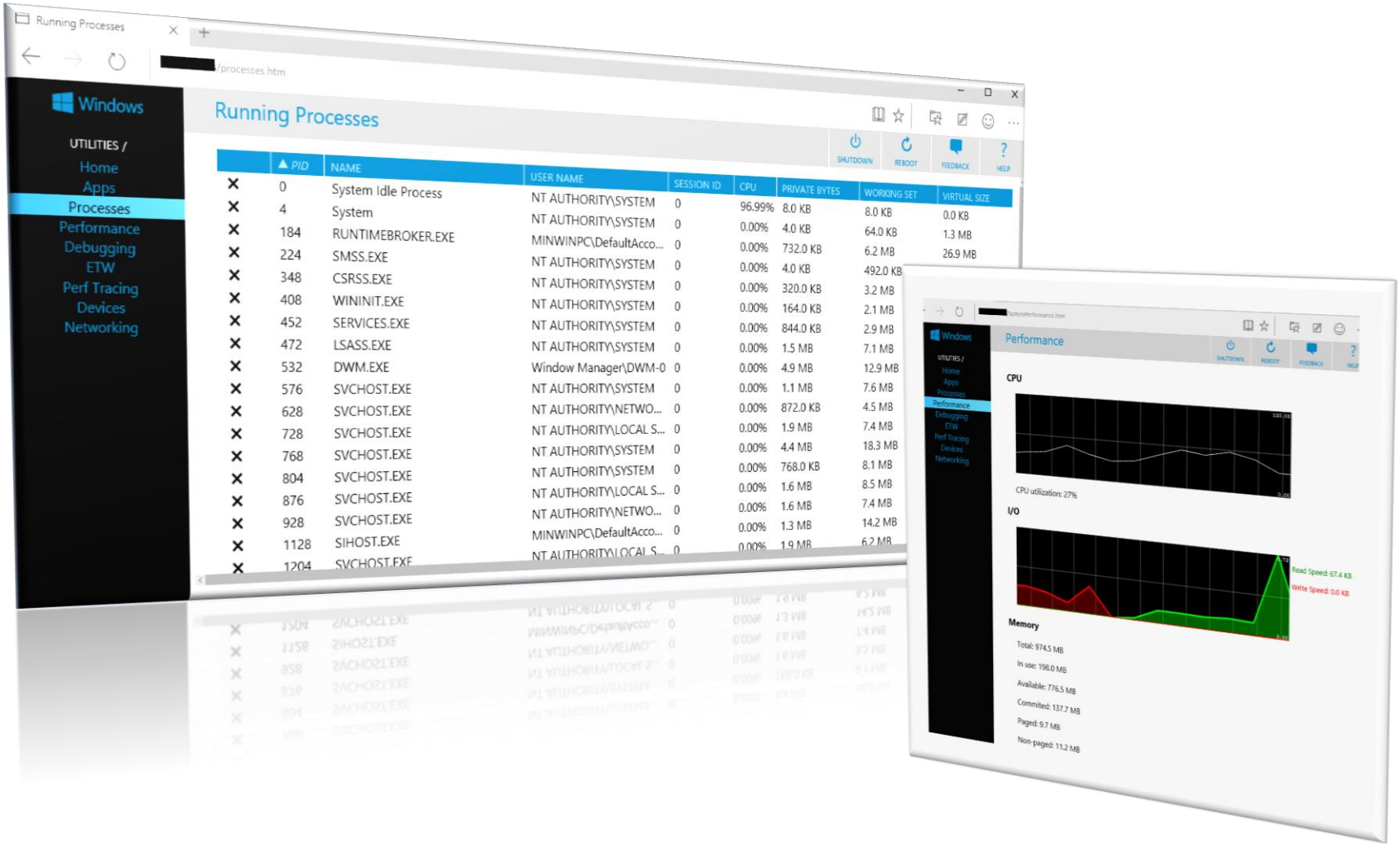
LED 点滅のまとめ

- Windows 10 IoT Core では、 Windows.Devices API を用いて、非常に簡単に GPIO をコントロールすることができる。

- 高機能 IDE の Visual Studio と Python や C# という人気のあるプログラミング言語が使えるため、PC アプリや Web アプリ開発者も気軽に組み込みアプリの開発に挑戦できる。また、GUI アプリの開発も可能である。
  - http://ms-iot.github.io/content/en-US/win10/samples/HelloWorld.htm

- Visual Studio で作成したコードをデバイス上で実行することができる。
  - デフォルトでは認証不要(「Authentication Mode」が「None」)で実行可能であった。

Web インターフェイスについて



Web インターフェイスの概要

- http://<デバイスのIPアドレス> でアクセスできる Web インタフェースを介して下記の機能が利用できる。(要ベーシック認証)

項目名	概要
Apps	アプリのインストール、起動、削除などが行える。 また、現在実行中のユーザーアプリケーションを確認できる。
Processes	実行中のプロセスの一覧が確認できる。
Performance	リアルタイムで　CPU、メモリーの使用率や I/O 状況を確認できる。
Debugging	カーネルダンプやプロセスダンプと言った情報の取得ができる。 また、クラッシュレポートの設定と、これまでに作成したレポートの詳細が確認できる。
ETW	ETW(Event Tracing Windows)が確認できる。
Perf Tracing	WPR(Windows Performance Recorder)によるメモリーリーク等の確認ができる。
Devices	デバイスマネージャー
Networking	ネットワークへの接続状態を確認できる。

ポートスキャンによるネットワークサービスの確認

- Windows 10 IoT Core 起動直後に nmap を用いてポートスキャンを行ったところ、下記のサービスが稼働していることが分かった。

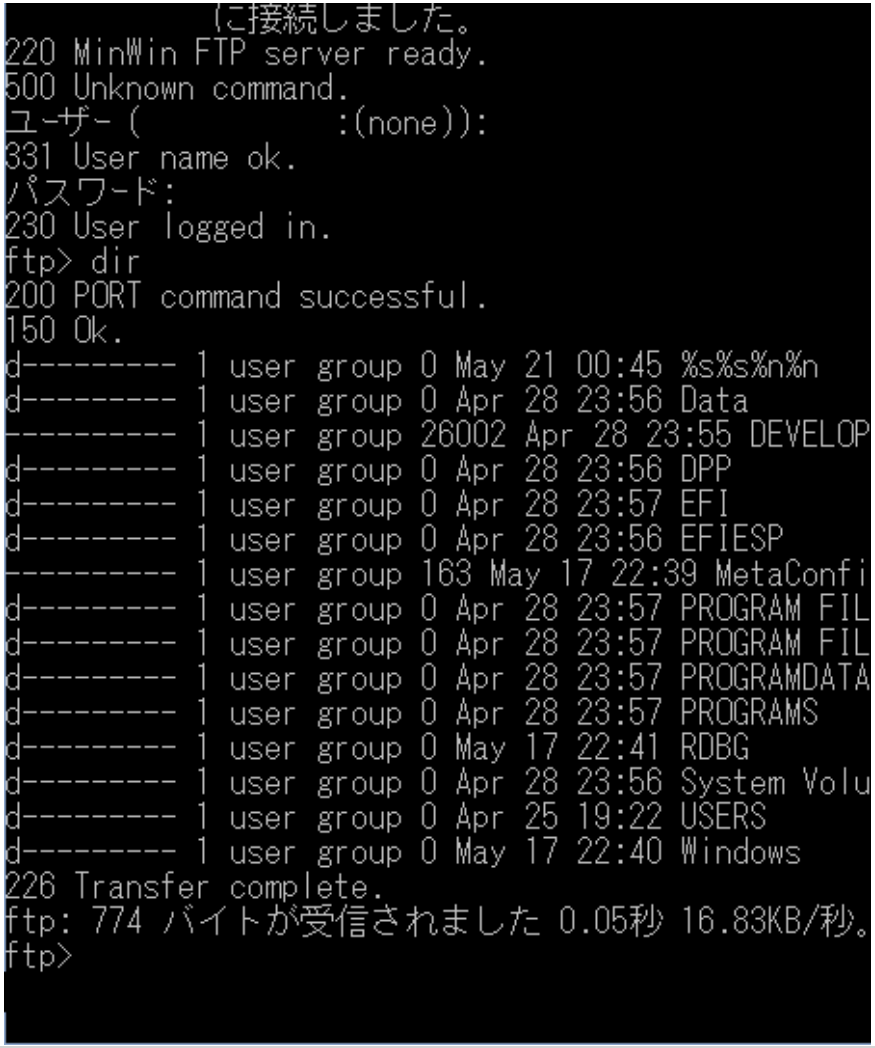
PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	
22/tcp	open	ssh	(protocol 2.0)
80/tcp	open	http	Microsoft-HTTPAPI/2.0
135/tcp	open	msrpc	Microsoft Windows RPC
445/tcp	open	microsoft-ds?	
4020/tcp	open	trap	
5985/tcp	open	wsman	
9956/tcp	open	unknown	
29817/tcp	open	unknown	
29819/tcp	open	unknown	
29820/tcp	open	unknown	

ポートスキャンによるネットワークサービスの確認

- nmap で OS fingerprint を行った結果、Windows OS である事は特定可能であった。
- SSH, HTTP は認証が必要であったが、FTP は任意のユーザーが接続可能(詳細後述)
- SSH は公開鍵認証は使用できなかった。また SCP, SFTP, ポートフォワーディングも利用不可であった。リモートコマンドの実行のみが可能と思われる。

FTP で任意のユーザーが全ファイルを読み取り可能

- FTP のユーザー名とパスワードは空文字も含む任意の文字列で接続可能
- ルートディレクトリは「/」で、Windowsのファイルシステムのルートであり、全ディレクトリに対して読み取り可能であるが、書き込みは不可
  - 有効な認証情報を入力し、ネットワークドライブとしてマウントした場合には読み書きが可能
- 任意ユーザーでのアップロードは不可のため、FTP の悪用によるマルウェア感染の可能性は低い。しかし、重要な設定情報などが読み取られる可能性がある。



Windows 10 IoT Core セキュリティ機能の確認

- Windows Firewall はデフォルトで無効
  - Web インタフェースからは設定できないが、netsh コマンドにて設定可能
- Windows Update は非対応
- マルウェア対策ソフト Windows Defender は非搭載
- UAC は無効
  - インタラクティブな UI がないエディションでは不要と思われる。デスクトップシェルを有するエディションでは有効の可能性あり
- DEP, ASLR はデフォルトで有効
- Control Flow Guard はサポートされており、CFG が有効な PE ファイルを実行可能
  - アプリケーションに適用するには、VS2015 でプロジェクトのプロパティからリンカーオプションの設定が必要
  - Project>Property>C/C++>Code Generation>Control Flow Guard

スタートアッププログラムの確認(startupコマンド)

```
C:¥>startup
startup
Startup Editor

Options:
  /d - display the list of startup apps
  /r <Name> - remove an app from the list of startup apps
  /a <Name> <Command> - add an app into the list of startup apps

Where:
  <Name> is the name of the app in the startup registry
  <Command> is the full command line for the app

Example:
  Startup /a EbootPinger "start ¥windows¥system32¥EbootPinger.exe"

C:¥>
```

- startup コマンドにてスタートアップで起動するプログラムの一覧取得、登録、削除をすることができる。
  - “C:¥Windows¥system32¥STARTUP.EXE” が実体
- デフォルトでは FTP サーバーである ftpd.exe がスタートアップ登録されている。

脅威分析（１）

- 不正アクセス(管理者へのなりすまし)や乗っ取り
  - セットアップ時にアカウント設定ウィザードがないため、ビルトインアカウント Administrator 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。
  - そのようなデバイスは、発見されると容易に Web インタフェースに不正アクセスされ、乗っ取られる恐れがある。
  - また、デフォルトパスワード設定のデバイスを標的とした自動攻撃ツールなどが出回ることも考えられる。
- パスワードクラッキング
  - ftp, http, ssh の認証に対してパスワードクラッキングが行われる恐れがある。
- 盗聴によるアカウント情報の流出
  - ftp, http 通信を盗聴される恐れがある。
    平文で認証情報が流れるためアカウント情報が流出し、不正アクセスや乗っ取りが行われる恐れもある。

脅威分析（２）

- プログラムやデータの流出
  - VS2015 付属の「Windows IoT Core Watcher」から、LAN 内の Windows 10 IoT 搭載デバイスを探索可能であり、デフォルトで稼働している FTP サーバーに任意ユーザーでアクセス可能な上、デバイス上の全ファイルの閲覧が可能であることから、攻撃者がデバイスと同一 LAN に接続できた場合、プログラムやデータが流出する恐れがある。

- マイコンボードに接続されたハードウェアの不正操作
  - VS2015 を用いて簡単に GPIO をコントロールできるコードを作成でき、作成したコードをデバイス上で実行可能(デフォルトでは認証不要)であるため、マイコンボードに接続されたハードウェア（カメラ、スイッチ等）を不正操作される恐れがある。

脅威分析（３）

- プログラムやデータの改ざん、マルウェア感染
  - 認証不要でVisual Studioからコードが実行できるため、それを悪用したワーム等が流行する恐れがある。
  - 攻撃者が任意の OS コマンドを実行可能になった場合、前述の startup コマンドを用いてスタートアップにマルウェアを追加される恐れがある。

  - 考えられる攻撃のシナリオ
    1. IoT デバイスが接続されたネットワーク上の PC が RAT に感染
    2. 感染 PC が遠隔操作され、Windows 10 IoT デバイスを探索される
    3. IoT デバイス上でバックドアプログラムを実行される
    4. Iデバイスのスタートアップへのマルウェア追加や正規のアプリを不正なプログラムに上書きするなどしてバックドアを作成し、継続的にコントロール
    5. デバイス制御 API を利用して接続されたハードウェアを不正操作

対策案

- 想定される脅威に対して下記の対策が考えられる。
  - 管理用ユーザーを新たに追加し、複雑なパスワードを設定する
    - リモートシェルから「net user ユーザー名 /add」で追加
  - ファイアウォールで必要最低限の通信のみ許可する
    - netsh コマンドで接続できる端末やポートを制限
  - 不必要なサービスを停止する
    - 必要がなければ FTP を停止
  - 安全な暗号化通信を行う
    - インターネットを越しに Web インタフェースを使用せず、ssh で管理
    - Wi-Fi 接続では WEP 以外の暗号化を使用
  - 物理セキュリティを確保する
    - GPIO PIN などへ周辺機器を不正接続されることが考えられるため、アプリケーション側で入力信号を検証

参考情報

- Introducing Windows 10 Editions
  http://blogs.windows.com/bloggingwindows/2015/05/13/introducing-windows-10-editions/
- WinHEC Shenzhen 2015
  https://channel9.msdn.com/Events/WinHEC/2015
- Internet of Things Overview(Build 2015)
  https://channel9.msdn.com/Events/Build/2015/2-652
- Python Tools for Visual Studio
  https://pytools.codeplex.com/
- Windows IoT - Python Blinky Sample
  https://ms-iot.github.io/content/en-US/win10/samples/PythonBlinky.htm
- Windows IoT - Blinky Sample
  https://ms-iot.github.io/content/en-US/win10/samples/Blinky.htm
- Nmap
  https://nmap.org/

Contact Information

E-Mail ： research—feedback@ffri.jp
Twitter： @FFRI_Research