プロトコルスタックを スクラッチ開発したエンジニアが考える セキュア通信をコンパクトに 実現する方法とは？

イー・フォース株式会社　四宮　充智

会社紹介

- **本社　　：東京**
- **開発拠点　　：東京 / インド**
- **事業内容**
  - Embedded事業
    - RTOSやミドルウェアの開発・販売
  - IoT事業
    - IoTプラットフォーム「iot-mos」の開発・販売
    - コンサルタント
- **沿革**

2006　:設立
国内のOSベンダとして初めて、ArmのCortex®-M/Aに対応し
「μC3/Compact」、「μC3/Standard」をリリース
2008　:TCP/IPスタック「μNet3」を開発
2013　:マルチコアデバイスに対応した「μC3/Standard+M」をリリース
2015　:産業用イーサネットへの取り組みをスタート
2016　:無線LANモジュール向けのSWパッケージを開発
2017　:RTOSとLinuxを共存させるソリューションをリリース
2018　:IoTプラットフォーム「iot-mos」を発表
2019　:Arm Cortex®-M33(TrustZone)に正式対応

アジェンダ

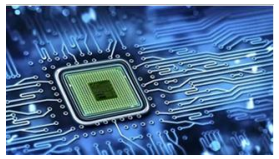
1. いまそこにある脅威
2. セキュア通信をコンパクトに実現するとは
3. セキュリティ対策を支えるための技術紹介
   - ➢ イー・フォースの取り組みのご紹介

1. いまそこにある脅威

サイバーセキュリティとは

サイバーセキュリティとは何を指すのか？

- ランサムウェア？
- SNSアカウント乗っ取り？
- 組込み機器のハッキング？
- etc.…



Wikipediaによると

サイバーセキュリティ

ページ　ノート

出典: フリー百科事典『ウィキペディア（Wikipedia）』

サイバーセキュリティ（英: cyber security）は、サイバー領域に関するセキュリティを指す。

概要 [編集]

サイバーセキュリティはサイバー領域のセキュリティを指し、その定義は論者によって異なるものの(後述)、この言葉は2010年ころから[1]情報セキュリティに変わるバズワード的な語として用いられるようになった。この言葉が登場した2010年頃はセキュリティにとってのターニングポイントになっており[2]、2010年のスタックスネットの事案や2011の三菱重工の事案からもわかるように[2]、ターニングポイント以降、以下の問題が顕在化した。

- 攻撃対象が産業システムにも広がった[2]
- 攻撃方法も高度化して特定組織を狙った標的型攻撃が行われるようになった[2]
- 攻撃目的も国家によるサイバー攻撃、犯罪者による金銭目的、ハクティビストによる主義主張の目的などに多様化した[2]

Wikipedia https://ja.wikipedia.org/wiki/サイバーセキュリティ

- 定義は論者によって異なる
  - 情報セキュリティの一部とみなしサイバー空間において機密性、完全性、可用性の確保を目指すもの
  - デジタル社会のリスクへの対応を指すとするもの
- 情報セキュリティに変わるバズワード
- 攻撃対象が産業システムにも広がった

【バズワード】
いかにも専門性・説得力のある言葉に聞こえていても、
曖昧な定義のまま広く世間で使われてしまう用語・造語・フレーズのこと

- サイバーセキュリティは組み込みシステムへにも影響
- しかしサイバーセキュリティの定義は曖昧

セキュリティ対策は必要だけど何をやればいい？

セキュリティ対策ありきがプロダクトの足かせになっている

情報セキュリティ 10大脅威

	個人
1	フィッシングによる個人情報等の詐取 ↑
2	ネット上の誹謗・中傷・デマ ↑
3	メールやSMS等を使った脅迫・詐欺の手口による金銭要求 ↑
4	クレジットカード情報の不正利用 ↑
5	スマホ決済の不正利用 ↓
6	偽警告によるインターネット詐欺 ↑
7	不正アプリによるスマートフォン利用者への被害 ↑
8	インターネット上のサービスからの個人情報の窃取 ↓
9	インターネットバンキングの不正利用 ↓
10	インターネット上のサービスへの不正ログイン →

	組織
1	ランサムウェアによる被害 →
2	標的型攻撃による機密情報の窃取 →
3	サプライチェーンの弱点を悪用した攻撃 ↑
4	テレワーク等のニューノーマルな働き方を狙った攻撃 ↓
5	内部不正による情報漏えい ↑
6	脆弱性対策情報の公開に伴う悪用増加 ↑
7	修正プログラムの公開前を狙う攻撃(new) ↑
8	ビジネスメール詐欺による金銭被害 ↓
9	予期せぬIT基盤の障害に伴う業務停止 ↓
10	不注意による情報漏えい等の被害 ↓

IPA 情報セキュリティ10大脅威2022 https://www.ipa.go.jp/security/vuln/10threats2022.html

- 社会的な**情勢の変化**に伴う脅威がランキングしている (組織 2, 3, 4)
- SNSやgit-hubで**ノウハウを共有**できることから**攻撃の敷居**が下がってきている(組織 6, 7)
- **情報技術(IT)**にフォーカスしているため**組み込みデバイス**との関連性が**見えづらい**

組込みデバイスの関連性を見るには脅威をブレイクダウン**する必要がある**

セキュリティ対策のハードルを考える

何がセキュリティ対策の障壁になっているのか？

攻撃者の立場や目的が多様化、複雑化(本来まだまだある)

- ✓ 環境(5G, クラウド)の利用
- ✓ 脆弱性の実証(PoC)コードの利用
- ✓ IoT機器(botネット)などプレイヤーの乱立

セキュリティ対策によって本来のユーザビリティを損なう

- ✓ パフォーマンスの低下
- ✓ 実装コードの肥大化

セキュリティ対策に費やすコストは投資という意識

- ✓ インシデントが発生するまで意識しない
- ✓ 実際は保険と同じ

そもそもセキュリティ技術は敷居が高い？

- ✓ 専門家の不在
- ✓ 敵の姿がみえない

組み込みデバイスに限らず一般的なセキュリティのはなし

問題は組み込み分野は情報セキュリティの耐性がない

制御システム 10大脅威

	制御システム
1	リムーバルメディアや**モバイルシステム**経由のマルウェア感染
2	**インターネット**やイントラネット経由のマルウェア感染
3	ヒューマンエラーと妨害行為
4	**外部ネットワーク**やクラウドコンポーネントの攻撃
5	ソーシャルエンジニアリングとフィッシング
6	**DoS/DDoS攻撃**
7	**インターネットに接続**された制御機器
8	**リモートメンテナンスアクセス**からの侵入
9	技術的な不具合と不可抗力
10	サプライチェーンにおけるソフトウェア／ハードウェアの脆弱性(new)

- ドイツのBSIから発表された産業用制御システム(ICS)のセキュリティ(IPA訳)
- 情報セキュリティに比べ特定のシチュエーションに言及されていることが多い
- ネットワークに接続する状況が多い (1, 2, 4, 6, 7, 8)
- トポロジやプロトコルは様々なのにネットワークへの接続自体が脅威として捉えられている

Industrial Control System Security: Top 10 threats and countermeasures 2022 [English] v1.5
https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_005E.html

ネットワーク技術と制御システムにおける脅威は密接に関係している

情報セキュリティでも繰り返し注意勧告されてきた内容

組み込みデバイスのセキュリティ対策はむずかしい？

組込みデバイスからと情報通信機器との比較

	情報通信機器	組込みデバイス
ユーザインターフェース	豊富な操作が可能	限定的な操作
ソフトウェア	汎用OS・アプリケーション	専用アプリケーション
ハードウェア	ハイエンドプロセッサ 大容量メモリ	マイコン～プロセッサ メモリは潤沢ではない
稼働	ユーザによる限定的な稼働	長期的・連続的な運用

- セキュリティ対策の**耐性不足**は**差分に起因** (攻撃不足, 対策不足)
- この状況に**Network**を介した攻撃が**激化**
- 組込みデバイスの**セキュリティ対策が難化**

どの脅威に対応する必要があるのか？

どうやって対策を実装するのか？

memcmpのはなし

int memcmp(const void *s1, const void *s2, size_t n)

usleep(random() % 10); /* 0～9μs停止 */

- 引数のサイズ(n)によっては該当領域のReadでオーバランが発生する？
- 複数のスレッドからリエントラントできる？
- 呼び出しコンテキストのスタックは足りる？
- 標準ライブラリをリンクすることで想定以上のメモリを消費する？

注意

暗号化された秘密などセキュリティ的に重要なデータの比較には **memcmp**() を使用しないこと。 必要な CPU 時間は値が等しいバイトの量に依存するからである。 その代わり、 一定時間で比較を実行する関数が必要である。 いくつかのオペレーティングシステムでは (例えば NetBSD の **consttime_memequal**() などの) 関数が提供されているが、 このような関数は POSIX では規定されていない。 Linux では、 このような関数自体を実装する必要があるかもしれない。

JM Project https://linuxjm.osdn.jp/html/LDP_man-pages/man3/memcmp.3.html

2. セキュア通信をコンパクトに実現するとは

セキュリティ対策を整理する

個人

1	フィッシングによる個人情報等の詐取
2	ネット上の誹謗・中傷・デマ
3	メールやSMS等を使った脅迫・詐欺の手口による金銭要求
4	クレジットカード情報の不正利用
5	スマホ決済の不正利用
6	偽警告によるインターネット詐欺
7	不正アプリによるスマートフォン利用者への被害
8	インターネット上の
9	インターネットバンキ
10	インターネット上の

組織

1	ランサムウェアによる被害
2	標的型攻撃による機密情報の窃取
3	サプライチェーンの弱点を悪用した攻撃
4	テレワーク等のニューノーマルな働き方を狙
5	内部不正による情報漏えい
6	脆弱性対策情報の公開に伴う悪用増加
7	修正プログラムの公開前を狙う攻撃
8	ビジネスメール詐欺による金銭被害
9	予期せぬIT基盤の障害に伴う業務停止
10	不注意による情報漏えい等の被害

制御システム

1	リムーバルメディアやモバイルシステム経由のマルウェア感染
2	インターネットやイントラネット経由のマルウェア感染
3	ヒューマンエラーと妨害行為
4	外部ネットワークやクラウドコンポーネントの攻撃
5	ソーシャルエンジニアリングとフィッシング
6	DoS/DDoS攻撃
7	インターネットに接続された制御機器
8	リモートメンテナンスアクセスからの侵入
9	技術的な不具合と不可抗力
10	サプライチェーンにおけるソフトウェア／ハードウェアの脆弱性

3つの10大脅威の対策をリストアップ

- ✓ クレジットカードの停止
- ✓ 警察への被害届の提出
- ✓ 利用していないサービスからの退会
- ✓ サービス運営者（コールセンター等）へ連絡
- ✓ 連携する金融機関へ連絡
- ✓ 警察に相談する
- ✓ 不用意にカレンダーの照会を追加しない 5
- ✓ カレンダー内の不審な予定は削除する
- ✓ 虚偽のサポート契約の解消 近くの消費生活センター6 に相談する
- ✓ クレジットカード会社へ連絡
- ✓ 利用しないアプリはアンインストールする
- ✓ サービス利用の必要性を判断する
- ✓ 利用していないサービスからの退会
- ✓ サービス運営者（コールセンター等）へ連絡
- ✓ 金融機関や公的機関から公開される注意喚 起を確認する
- ✓ 警察への被害届の提出
- ✓ 利用していないサービスからの退会
- ✓ クレジットカードの停止
- ✓ サービス運営者（コールセンター等）へ連絡
- ✓ パスワードの管理・認証の強化
- ✓ 設定の見直し
- ✓ 社内専用、制御ネットワーク専用
- ✓ 不正接続防止障壁（USBロック等）
- ✓ データ暗号化
- ✓ ネットワークのセグメント化
- ✓ 連携する金融機関へ連絡
- ✓ 警察に相談する

重複を含め約300項目の対策

セキュリティ対策を選択する

機械的にできないものは除外

- セキュリティ教育の実施
- 運用ルールの整備
- 警察に相談するなど

既存のセキュリティソフトを併用するものは除外

- アンチウィルス, EDRの導入など

除外した対策が無効なわけではない

組込みデバイスで実装できるもの

✓ ソフトウェアの更新
✓ パスワードの管理・認証の強化
✓ 設定の見直し
✓ 社内専用、制御ネットワーク専用
✓ 不正接続防止障壁（USBロック等）
✓ データ暗号化
✓ ネットワークのセグメント化
✓ ファイアウォールやIDSの設置
✓ アップデータの適用
✓ ログ監視
✓ 自動監視
✓ 信頼できる／認定されたサービスの利用
✓ 暗号技術を用いた外部保存データ保護

約130件

セキュリティ対策を分類する

脅威モデル
(STRIDE)

Spoofing(なりすまし)
Tampering(改ざん)
Repudiation(否認)
Information Disclosure(情報漏えい)
Denial of Service(サービス妨害)
Elevation of Privilege(権限昇格)

認証の強化 (Certification)
ネットワーク対策 (Network)
リフレッシュ動作 (Refresh)
痕跡の確認 (Trace)

セキュリティ
要素
(CIA+4)

Confidentiality(機密性)
Integrity(完全性)
Availability(可用性)
Authenticity(真正性)
Reliability(信頼性)
Accountability(責任追跡性)
non-repudiation(否認防止)

NIST CSF
Cyber Security Framework

Identify(特定) / Protect(防御) / Detect(検知) / Respond(対応) / Recover(復旧)

認証の強化

30大脅威における対策

- 多要素認証の使用
- 適切なパスワード運用(長さ、管理、更新)
- 信頼できるサービスの利用
- ネットワークレベル認証
- 公式アプリの利用
- フィッシングサイトへの注意
- 情報管理（need to knowの原則） など

脅威モデルでいうと**なりすまし**に該当

セキュリティ要素でいうと**真正性**、**完全性**に該当

認証は**される側**と**する側**のどちらの方向もある

認証される側は銀行口座やクレジットなどサービスを攻撃者がのっとるケース

認証する側はフィッシング詐欺、不明アプリ・URLをトリガーとするもの

組織、制御システムにおいても人の操作が介在するものの対策として有効

	個人		組織		制御システム
1	フィッシングによる個人情報等の詐取	1	ランサムウェアによる被害	1	リムーバルメディアやモバイルシステム経由のマルウェア感染
2	ネット上の誹謗・中傷・デマ	2	標的型攻撃による機密情報の窃取	2	インターネットやイントラネット経由のマルウェア感染
3	メールやSMS等を使った脅迫・詐欺の手口による金銭要求	3	サプライチェーンの弱点を悪用した攻撃	3	ヒューマンエラーと妨害行為
4	クレジットカード情報の不正利用	4	テレワーク等のニューノーマルな働き方を狙った攻撃	4	外部ネットワークやクラウドコンポーネントの攻撃
5	スマホ決済の不正利用	5	内部不正による情報漏えい	5	ソーシャルエンジニアリングとフィッシング
6	偽警告によるインターネット詐欺	6	脆弱性対策情報の公開に伴う悪用増加	6	DoS/DDoS攻撃
7	不正アプリによるスマートフォン利用者への被害	7	修正プログラムの公開前を狙う攻撃	7	インターネットに接続された制御機器
8	インターネット上のサービスからの個人情報の窃取	8	ビジネスメール詐欺による金銭被害	8	リモートメンテナンスアクセスからの侵入
9	インターネットバンキングの不正利用	9	予期せぬIT基盤の障害に伴う業務停止	9	技術的な不具合と不可抗力
10	インターネット上のサービスへの不正ログイン	10	不注意による情報漏えい等の被害	10	サプライチェーンにおけるソフトウェア／ハードウェアの脆弱性

個人に関してはほぼすべてが認証プロセスでの対策が有効

ネットワーク対策

30大脅威における対策

- ファイアーウォール, IDS/IPSの利用
- ネットワーク分離／セグメント化
- 重要サーバーの要塞化
- VPNの利用
- 冗長化接続
- 不要サービスの無効化
- 暗号化による伝送路の保護 など

脅威モデルでいうと情報の漏洩、DoS攻撃に該当

セキュリティ要素でいうと機密性、可用性に該当

NIST CSFでいうと防御に該当

L2(データリンク層), TCP/IP, ソケットにフォーカス

(デバイスドライバ, プロトコルスタック, ネットワークアプリケーション)

	個人		組織		制御システム
1	フィッシングによる個人情報等の詐取	1	ランサムウェアによる被害	1	リムーバルメディアやモバイルシステム経由のマルウェア感染
2	ネット上の誹謗・中傷・デマ	2	標的型攻撃による機密情報の窃取	2	インターネットやイントラネット経由のマルウェア感染
3	メールやSMS等を使った脅迫・詐欺の手口による金銭要求	3	サプライチェーンの弱点を悪用した攻撃	3	ヒューマンエラーと妨害行為
4	クレジットカード情報の不正利用	4	テレワーク等のニューノーマルな働き方を狙った攻撃	4	外部ネットワークやクラウドコンポーネントの攻撃
5	スマホ決済の不正利用	5	内部不正による情報漏えい	5	ソーシャルエンジニアリングとフィッシング
6	偽警告によるインターネット詐欺	6	脆弱性対策情報の公開に伴う悪用増加	6	DoS/DDoS攻撃
7	不正アプリによるスマートフォン利用者への被害	7	修正プログラムの公開前を狙う攻撃	7	インターネットに接続された制御機器
8	インターネット上のサービスからの個人情報の窃取	8	ビジネスメール詐欺による金銭被害	8	リモートメンテナンスアクセスからの侵入
9	インターネットバンキングの不正利用	9	予期せぬIT基盤の障害に伴う業務停止	9	技術的な不具合と不可抗力
10	インターネット上のサービスへの不正ログイン	10	不注意による情報漏えい等の被害	10	サプライチェーンにおけるソフトウェア／ハードウェアの脆弱性

制御システムの脅威では多くの対策で有効

リフレッシュ動作

30大脅威における対策

- 不要なアプリケーションの削除
- 当該ソフトウェアの一時的な使用停止
- 端末を初期化する
- 定期的なバックアップの取得
- セキュリティパッチの適用
- ソフトウェアの更新 など

ソフトウェア的なリスタートやリカバリー

組込みでいうとウォッチドッグタイマやウォームスタート

NIST CSFでいうと検知, 対応に該当

ソフトウェアの更新やバックアップもリフレッシュとする

NIST CSFでいうと防御に該当

	個人		組織		制御システム
1	フィッシングによる個人情報等の詐取	1	ランサムウェアによる被害	1	リムーバルメディアやモバイルシステム経由のマルウェア感染
2	ネット上の誹謗・中傷・デマ	2	標的型攻撃による機密情報の窃取	2	インターネットやイントラネット経由のマルウェア感染
3	メールやSMS等を使った脅迫・詐欺の手口による金銭要求	3	サプライチェーンの弱点を悪用した攻撃	3	ヒューマンエラーと妨害行為
4	クレジットカード情報の不正利用	4	テレワーク等のニューノーマルな働き方を狙った攻撃	4	外部ネットワークやクラウドコンポーネントの攻撃
5	スマホ決済の不正利用	5	内部不正による情報漏えい	5	ソーシャルエンジニアリングとフィッシング
6	偽警告によるインターネット詐欺	6	脆弱性対策情報の公開に伴う悪用増加	6	DoS/DDoS攻撃
7	不正アプリによるスマートフォン利用者への被害	7	修正プログラムの公開前を狙う攻撃	7	インターネットに接続された制御機器
8	インターネット上のサービスからの個人情報の窃取	8	ビジネスメール詐欺による金銭被害	8	リモートメンテナンスアクセスからの侵入
9	インターネットバンキングの不正利用	9	予期せぬIT基盤の障害に伴う業務停止	9	技術的な不具合と不可抗力
10	インターネット上のサービスへの不正ログイン	10	不注意による情報漏えい等の被害	10	サプライチェーンにおけるソフトウェア／ハードウェアの脆弱性

ソフトウェア自体を客観的にコントロール

痕跡の確認

30大脅威における対策

- ログイン履歴の確認
- クレジットカード利用明細の定期的な確認
- ログイン通知
- 適切なログの取得と継続的な監視 など

明細やイベントログのトレース

脅威モデルでいうと否認に該当

セキュリティ要素でいうと否認防止に該当

NIST CSFでいうと対応に該当

通知や監視は防御、検知に該当

	個人		組織		制御システム
1	フィッシングによる個人情報等の詐取	1	ランサムウェアによる被害	1	リムーバルメディアやモバイルシステム経由のマルウェア感染
2	ネット上の誹謗・中傷・デマ	2	標的型攻撃による機密情報の窃取	2	インターネットやイントラネット経由のマルウェア感染
3	メールやSMS等を使った脅迫・詐欺の手口による金銭要求	3	サプライチェーンの弱点を悪用した攻撃	3	ヒューマンエラーと妨害行為
4	クレジットカード情報の不正利用	4	テレワーク等のニューノーマルな働き方を狙った攻撃	4	外部ネットワークやクラウドコンポーネントの攻撃
5	スマホ決済の不正利用	5	内部不正による情報漏えい	5	ソーシャルエンジニアリングとフィッシング
6	偽警告によるインターネット詐欺	6	脆弱性対策情報の公開に伴う悪用増加	6	DoS/DDoS攻撃
7	不正アプリによるスマートフォン利用者への被害	7	修正プログラムの公開前を狙う攻撃	7	インターネットに接続された制御機器
8	インターネット上のサービスからの個人情報の窃取	8	ビジネスメール詐欺による金銭被害	8	リモートメンテナンスアクセスからの侵入
9	インターネットバンキングの不正利用	9	予期せぬIT基盤の障害に伴う業務停止	9	技術的な不具合と不可抗力
10	インターネット上のサービスへの不正ログイン	10	不注意による情報漏えい等の被害	10	サプライチェーンにおけるソフトウェア／ハードウェアの脆弱性

異変を察知、後から追跡

コンパクト化のまとめ

どの脅威に対応する必要があるのか？

脅威分析

- 分析対象となる脅威は幅広く情報を収集
- 技術主体でその分野でできることは何か？
- どの脅威に対応する必要があるのか？

どうやって対策を実装するのか？

対策の実装

- どのような原理を使って攻撃している？
  (似たようなものが多い)
- 適材適所の実装(必要最小限の実装)

コスト

- 実態は圧縮できない(？)
- 費用対効果(UP)
- フットプリント(DOWN)

3.セキュリティ対策を支えるための技術紹介

セキュリティ対策を支えるための技術紹介

認証の強化

- デジタル証明書
- 疑似乱数

ネットワーク対策

- TLS/暗号
- 受信フィルタと送信制御
- 帯域制御

リフレッシュ動作

- タイムアウト
- ファームウェアアップデート

痕跡の確認

- 統計情報

デジタル証明書①

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

デジタル証明書

デジタル証明書の利用シーン

- 暗号資産(仮想通貨やNFTなど)、デジタルサイン
- TLS/HTTPS, SSH, EAP(回線認証)など

パスワードとデジタル証明書の比較

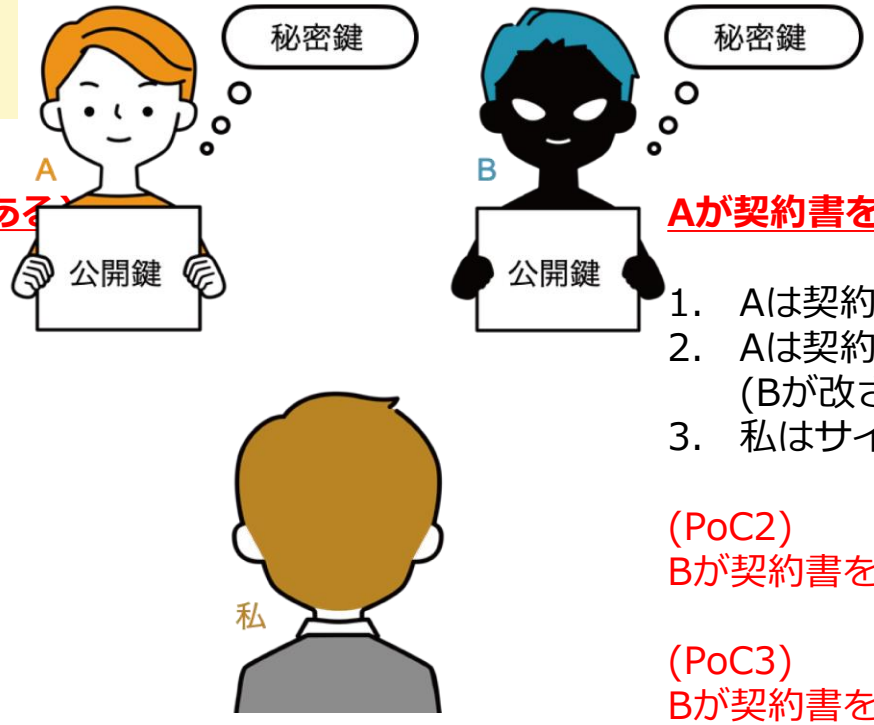
	パスワード	デジタル証明書
要素	知識を要素とする情報	所有を要素とする情報
発行者	使用者が設定する	第3者が発行する
管理	認証する側はアカウント数分の管理が必要	認証する側の管理は不要 認証される側は秘密鍵とペアで管理
機密性	ある	デジタル証明書(ない) 秘密鍵(ある)
使用方法	主にログイン時のUIで入力	プロトコル上の認証プロセス

デジタル証明書②

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

公開鍵暗号方式

公開鍵で暗号化したものは秘密鍵しか復号できない
秘密鍵で暗号化したものは公開鍵しか復号できない
(その目的から署名(サイン)／検証(ベリファイ)と呼ぶ)



私からAに資産を送る(資産は情報自体に価値がある)

1. 私はAの公開鍵で資産を暗号化してAに送る
   (Bが盗聴)
2. Aは秘密鍵で復号

(PoC1)
Bが盗聴→Bは復号できない→漏洩しない

Aが契約書を書いてサインをする(契約書に機密性はない)

1. Aは契約書を書き、Aの秘密鍵で暗号化(=サイン)
2. Aは契約書とサインをあわせて私に送る
   (Bが改ざん)
3. 私はサインをAの公開鍵で復号して契約書と比較

(PoC2)
Bが契約書を改ざん→私が復号しても異なる→改ざん検知

(PoC3)
Bが契約書を改ざん→Bの秘密鍵でサイン
→Aの公開鍵で復号できない→なりすまし検知

私は必ず本人(A)の公開鍵を使用すること！

デジタル証明書③

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

デジタル署名

「私は必ず本人(A)の公開鍵を使用する」をどうやって保証するか？

↓

Aの公開鍵にAの名前などの情報を添える　契約書

↓

信頼できる第3者がサインする　デジタル署名

↓

これらを合わせてA本人のデジタル証明書となる

↓

Aの公開鍵 + Aの情報　信頼できる第3者のサイン

紛れもないAのデジタル証明書

第3者への与信はデジタル証明書の利用者が決める

利用者が不特定多数の場合は第3者は認証局(CA)となる

疑似乱数

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

疑似乱数

プロトコルにおける利用シーン

① 通信負荷の**分散**

一斉にパケットを送信するケース(起動時、応答時など)

② **推測不能**な値の生成

- ✓ **一意なID**(DNSやDHCPのトランザクションIDなど)の生成
- ✓ **一時的なポート**(ローカル, FTPパッシブポートなど)選択
- ✓ TCPの**シーケンス番号**初期値
- ✓ ワンタイムパスワード
- ✓ TLSにおける**鍵(マスターシークレット)**

  - ➢ **一時的**に利用する値
  - ➢ **中間者攻撃**の標的

良質な暗号アリズムを使っても鍵の品質が悪いと意味がない

積極的なシード**の更新**

ハードウェアエンジンの使用

サーバ側の乱数の使用

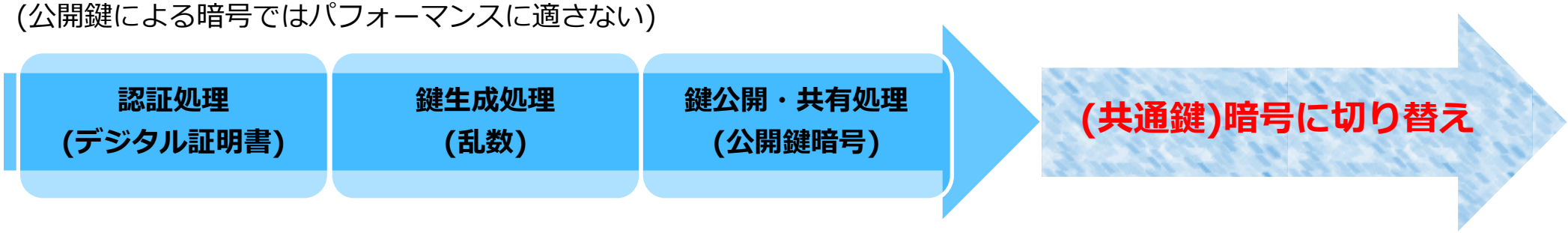
TLS/暗号

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

情報セキュリティではもはやTLSやHTTPSを使いなさいという対策はない
(制御システムではNo.4とNo.8には対策としてある)

デジタル証明書と乱数に暗号処理(HASH, CRYPT) を加えて実現される
(公開鍵による暗号ではパフォーマンスに適さない)

脅威モデル	保障されるもの	セキュリティ技術
なりすまし	相手の真正性	デジタル証明書
改ざん	情報の完全性	HASH
情報漏洩	情報の機密性	CRYPT, 乱数



- TLSを使用する一般的な上位プロトコルはHTTPS, MQTT, SMTPS, IMAPS, FTPSなど
- 逆にTLSではないセキュアなプロトコルはSSH, SFTP, SCP, IPsecなど
- 暗号処理はハードウェアエンジンを搭載するマイコンも多い (処理速度・メモリ効率はSWに比べ圧倒的に高い)

TLSはセキュリティ通信の要(ネットワークから要求されるプロトコル)

受信フィルタと送信制御

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

受信フィルタ（不慮の侵入は防げ！）

レイヤー	Source	Destination
MACアドレス	通信相手を特定可能なら指定	ユニキャスト(自身) マルチキャスト(参加中) ブロードキャスト
IPアドレス	通信相手が特定可能なら指定 同一ネットワークアドレス	
TCP/UDPポート	ANY	待ち受けるものだけ LISTENキューは必要な数

順位	宛先ポート番号	前四半期の順位
1	23/TCP（telnet）	1
2	6379/TCP（redis）	2
3	22/TCP（ssh）	3
4	80/TCP（http）	4
5	5555/TCP	8

JPCERT/CC インターネット定点観測レポート（2022年 7～9月）
https://www.jpcert.or.jp/tsubame/report/report202207-09.html

不要なパケットは受信しない

必要であっても**必要以上**に受信しない

送信制御（あまり目立つな！）

- Ping
- ICMP (unreachable etc.)
- IP Forwarding

- Padding
- Beacon
- meta-tag

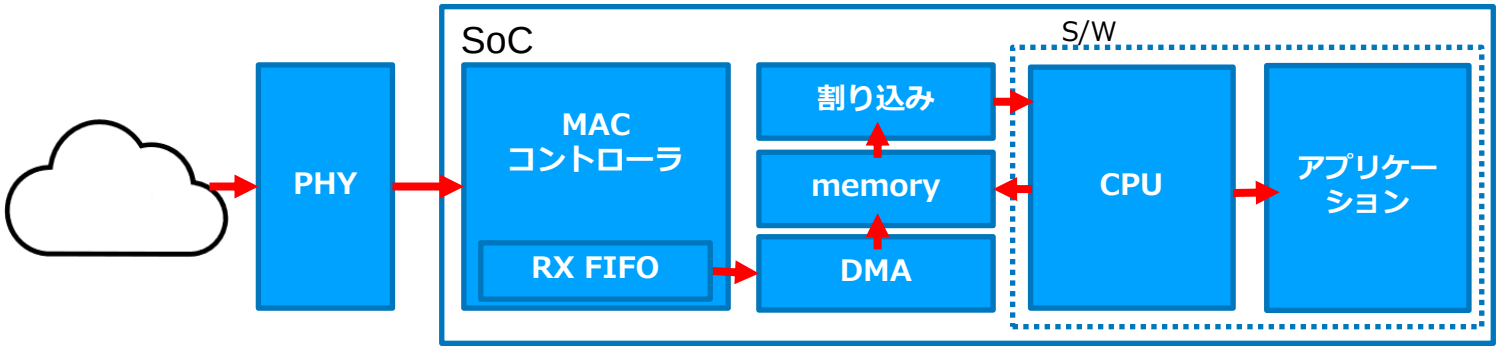
スキャンは**Botnet**の**常套手段**

不要な情報は発信しない

帯域制御

帯域制御

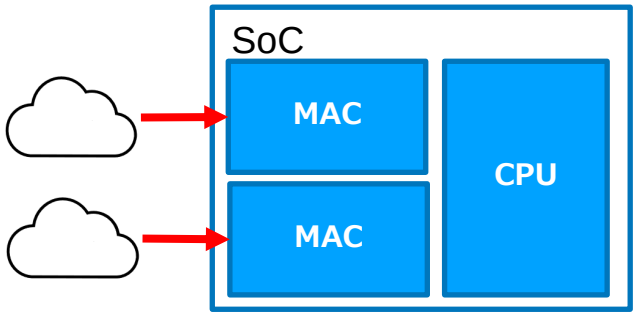
帯域を制御することはサービスを停止させるDoS攻撃の対策
ループ接続によるブロードキャストストームの対策にも有効



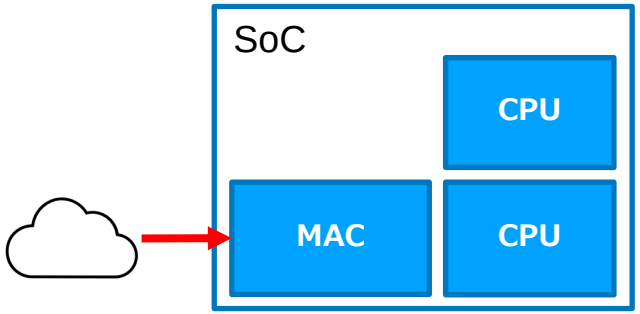
- DMA転送量(ディスクリプタやMACのFIFO)に閾値を設ける(HW依存)
- S/Wで単位時間の受信パケット数を計測しMACの受信を一時的に停止

通信動作とCPU稼働はフェールソフトに

- MACを複数内蔵するSoCはセグメント分断



- 通信自体を専用コアで行う



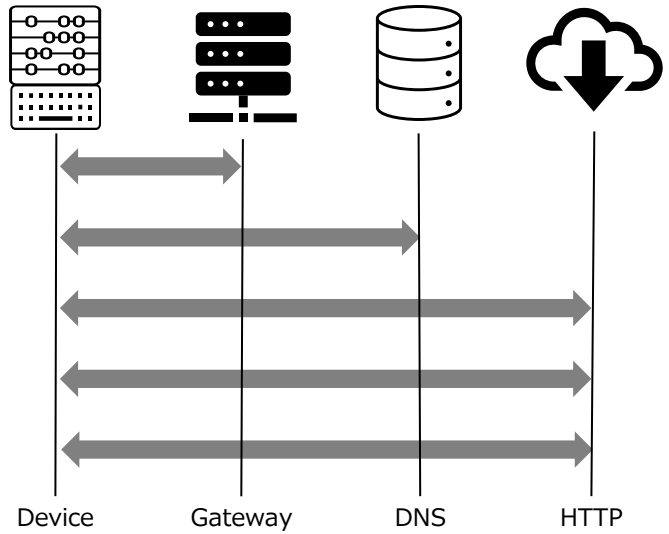
タイムアウト①

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

タイムアウトの目的は不要なリソースの解放

(例)HTTPSでGETリクエストを発行する手順

手順	プロトコル動作
IPアドレスからMACアドレスを取得	ARP Request送信／Response受信 ICMPv6 Neighbor Discovery送信／Advertise受信
URLのドメイン名からIPアドレスを取得	UDPでDNS Query送信／Response受信
IP上でTCPを接続	TCPでSYN送信／SYN/ACK受信／ACK送信
TCP上でTLSを接続	TCPでClient Hello送信～(TLS Handshake)～Finished受信
TLS上でHTTP GETリクエスト発行	TLS上でHTTP GET送信／HTTP Response受信



- 基本的に応答がないと手順は進まない
- 手順が先に進まなければ状態を持つプロトコルではリソースを占有
- ARP, N/D, DNS, HTTP GETはキャッシュを利用することがある (他にもNAT, PROXY, CDN, etc..)

タイムアウト②

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

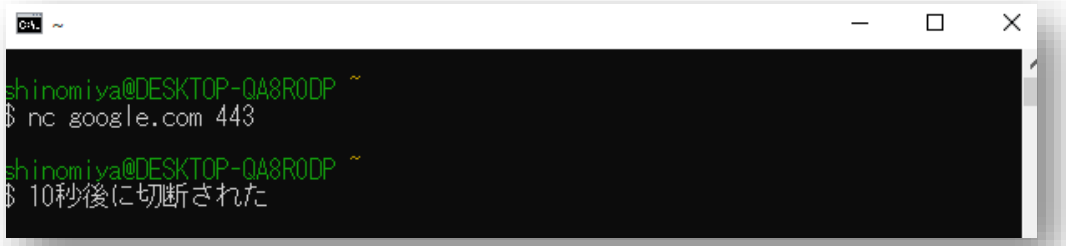
応答が止まる要因

- 物理な接続の問題(Wi-FiやLANのリンクダウンやセグメント変更)
- Flood攻撃などのリソースの占有を目的とした攻撃の被害
- キャッシュポイズニングによる不適切な通信の発生

有効な対策

- リンク状態の監視
- 受動的な動作をする場合はタイムアウトを指定
- 待つ必要がある場合は疎通確認(keep-Alive)
- 来ない応答を待ち続けるならリトライ
- キャッシュのTTLを有効化
  (用途によってはキャッシュを使用しない)

google.comにTCPだけ接続した場合10秒後に切断された



Time	SRC_port	DST_port	Protocol	seq	ack	dat	INFORMATION
REF	57941	53	DNS				Standard query 0xd1b2 A google.com
0.000240	63581	53	DNS				Standard query 0x8733 AAAA google.com
0.002971	53	63581	DNS				Standard query response 0x8733 AAAA google.com AAAA 2404:6800:4004:801::200e
0.026370	57941	53	DNS				Standard query 0xd1b2 A google.com
0.029029	53	57941	DNS				Standard query response 0xd1b2 A google.com A 142.250.199.110
0.037936	59205	443	TCP	0	0	0	59205 → 443 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM=1
0.040749	443	59205	TCP	0	1	0	443 → 59205 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
0.041173	59205	443	TCP	1	1	0	59205 → 443 [ACK] Seq=1 Ack=1 Win=263424 Len=0
0.082155	53	57941	DNS				Standard query response 0xd1b2 A google.com A 142.250.199.110
10.048427	443	59205	TCP	1	1	0	443 → 59205 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0
10.049214	59205	443	TCP	1	2	0	59205 → 443 [ACK] Seq=1 Ack=2 Win=263424 Len=0
10.050095	59205	443	TCP	1	2	0	59205 → 443 [FIN, ACK] Seq=1 Ack=2 Win=263424 Len=0
10.051490	59205	443	TCP	2	2	0	59205 → 443 [RST, ACK] Seq=2 Ack=2 Win=0 Len=0
10.053227	443	59205	TCP	2	2	0	443 → 59205 [ACK] Seq=2 Ack=2 Win=65536 Len=0
10.053383	59205	443	TCP	2	13577…	0	59205 → 443 [RST] Seq=2 Win=0 Len=0

タイムアウトの振る舞いを考えることがリフレッシュ動作で重要

ファームウェアアップデート

認証の強化	ネットワーク対策	リフレッシュ動作	痕跡の確認

- 不具合の改修、機能追加、セキュリティパッチ
- bootモードの変更やローカルストレージが使えない状況を想定
- ファームウェア更新自体がアタックサーフェスになるのでは？

POINT	
更新契機 (デバイス操作不可)	**同期型** HTTP GETなどで定期的に更新契機の有無を問い合わせる (サーバ側の負荷を分散させる必要がある) **非同期型** サーバが任意のタイミングで配信できる (MQTTなどの非同期型のプロトコルを使用する必要がある)
ファーム転送プロトコル	HTTPS, SFTP, SCPなど (バイナリファイルが転送できてTLSベースのもの)
ファームウェアの真正性・完全性	バージョン、ハッシュ、マジックコードなどで保証 (メモリ保存時やboot時に検証)
ROM	現行用と更新用の最低２つの領域が必要 (ロールバックやバックアップ用) Read/Writeのプロテクションがあるものを使用する

統計情報

ログ出力の問題点

- ログは低優先度でファイルに出力するのが一般的
- ログの解析は問題発生後の作業で有効
  (ログの解析作業はどちらかといえばよろしくないシチュエーション)
- 脅威を監視するにはリアルタイムでトレースが必要

SNMP(MIB)のメリット

- SNMPはMIB(データベース) を管理するプロトコル
- デバイスはSNMPエージェント
- デバイスの動作や状態、通信記録をSNMPマネージャで監視可能
- アプリケーション固有のMIB(プライベートMIB)も使用可能
- trapを使って任意のMIBの変化をイベント通知することも可能

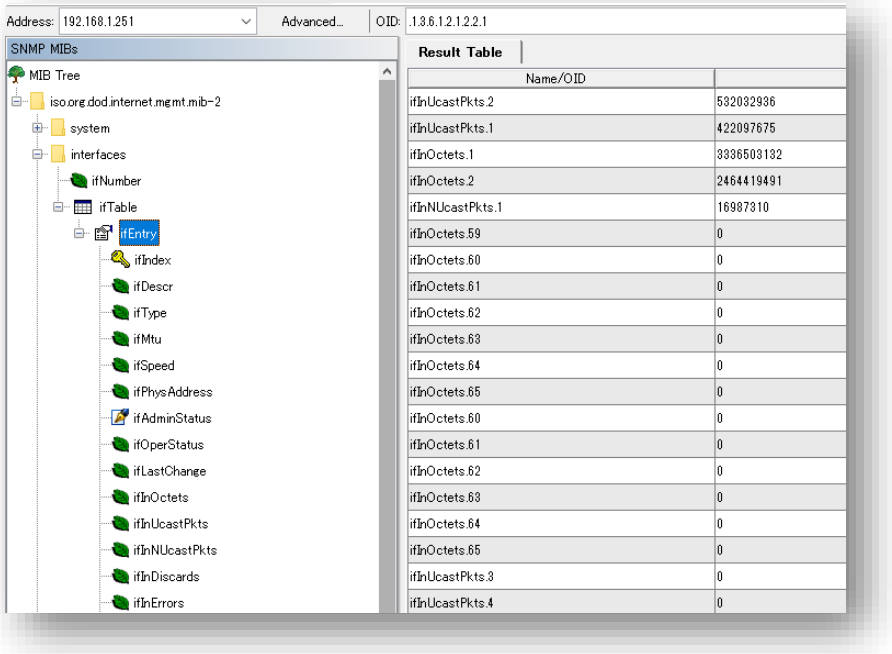
組込みシステム × モニター(DB)は今後のテーマ？

H/Wで統計情報を提供するものもある

- Supports IEEE 802.1Q VLAN tag detection for reception frames
- Separate transmission, reception, and control interfaces to the Application
- Supports mandatory network statistics with RMON/MIB counters (RFC2819/RFC2665)
- MDIO interface for PHY device configuration and management
- Detection of LAN wakeup frames and AMD Magic Packet™ frames
- Receive feature for checksum off-load for received IPv4 and TCP packets

STM32F7xシリーズのリファレンスマニュアルより

ルータの標準MIB



イー・フォースの取り組み(2022-2023)

- 機能安全
- IoT機器の脆弱性検証促進事業
- TLS1.3の開発

機能安全

機能安全認証取得RTOS

【特徴】

- μITRON4.0のスタンダードプロファイルに準拠したμC3/Standardにリアルタイム検知機能を追加

- 下記機能安全規格の認証を取得予定
  - IEC 61508 SIL-3
  - ISO 26262 ASIL D
  - IEC 62304 Class C

- ターゲットCPUコア
  - ARMv7-M：Cortex-M3, M4, M7
  - ARMv8-M：Cortex-M33

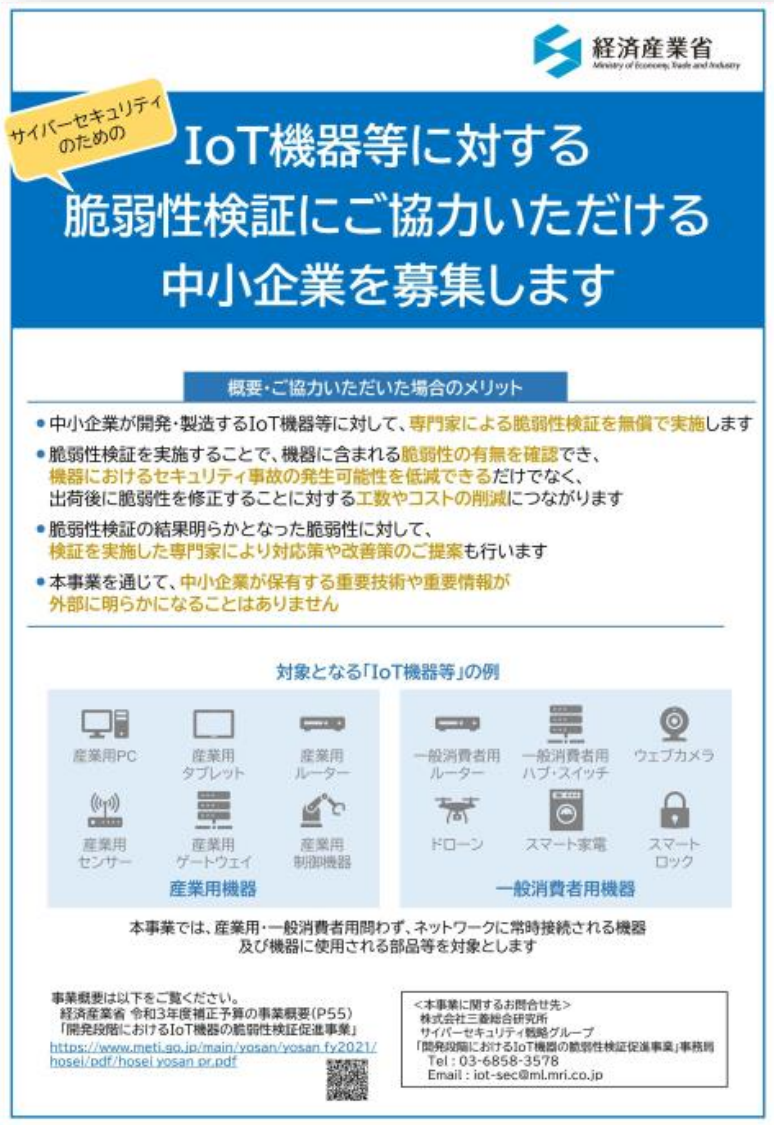
μC3 micro c cube

2023年秋
認証取得予定

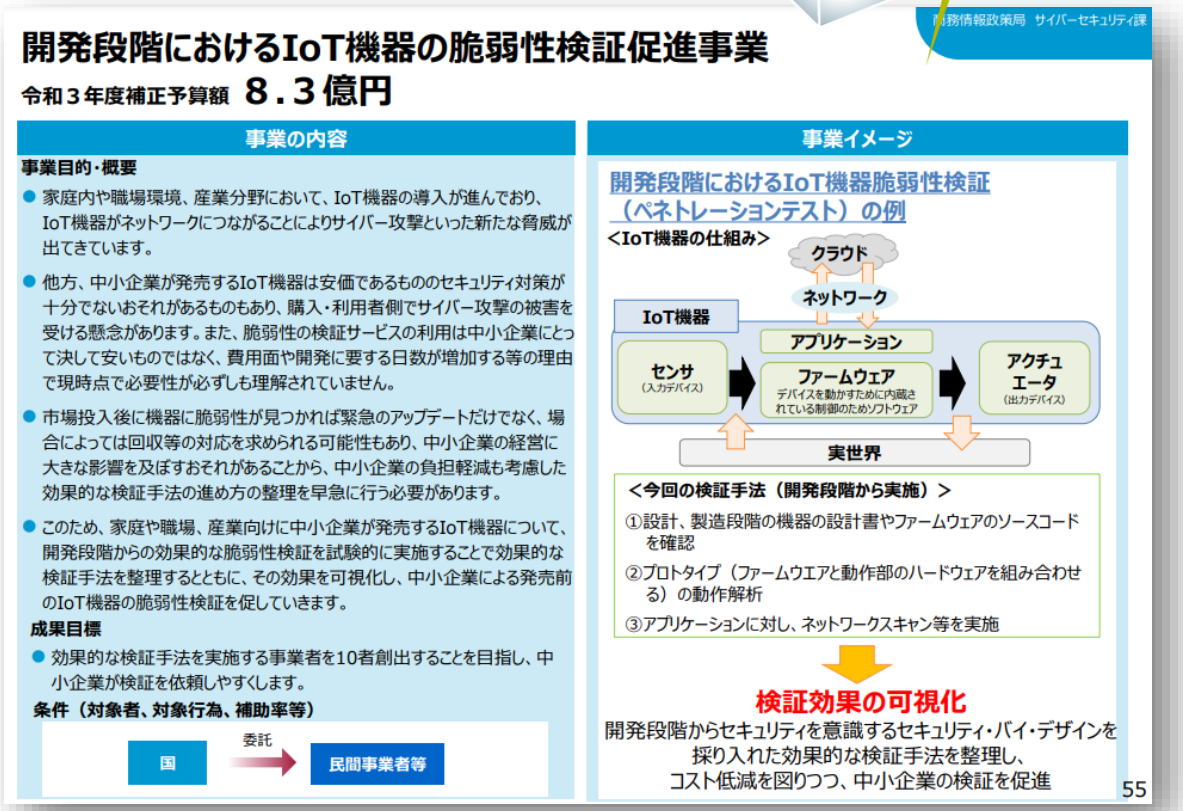
eForce Channel 第１回『機能安全認証にRTOSって必要なの？』～機能安全RTOSが今後の機能安
https://www.youtube.com/watch?v=F3vfahdb9J8

eForce

IoT機器の脆弱性検証促進事業



μNet3 **TCP/IP, TLS**を2022年12月から検証



MRI 三菱総合研究所 「開発段階におけるIoT機器の脆弱性検証促進事業」（経済産業省事業）においてIoT機器の脆弱性検証を希望する中小企業の募集のご案内について
https://pubpjt.mri.co.jp/publicoffer/20220425_2.html

TLS1.3の開発

μNet3/SSL, μNet3/TLS1.3(仮称) ロードマップ

μNet3/TLS1.3

μNet3/TLS

Crypt Library

TLS1.3

Crypt Function
(RSA-pss, ECDH, ECDSA, AES)

μNet3/TLS1.3
2023年1月リリース予定

μNet3/SSL(既存のTLS1.2)

Crypt Function
(RSA, ECDH, ECDSA, AES..)

μNet3/SSL→μNet3/TLS
2022年7月リリース
・Server機能追加
・サイファー追加

CAVP認証(米国調達安全基準)
2023年12月リリース予定

μNet3 micro net cube

eForce

ご清聴頂きありがとうございました

アンケートのご協力をお願いします

https://www.eforce.co.jp