IoTセキュリティ・ガイドライン・

エンドポイント・エコシステム

IoT セキュリティ・ガイドライン・エンドポイント・エコシステム

バージョン 2.0

2017 年 10 月 31 日

本文書はGSMA の拘束力のない恒久参照文書です。

セキュリティ区分： 公開可能

本文書の入手および配布は、セキュリティ区分で認められた者に限られます。本文書は GSM の機密文書であり、著作権保護が適用されます。本文書はその提供目的のためにのみ使用されるものとし、本文書の全部もしくは一部の情報を、GSM の書面による事前の承認なくセキュリティ区分によって認められている者以外に開示する、またはそれ以外の方法で利用可能にすることを禁じます。

著作権表示

免責事項

GSM Association（GSMA）は、本文書に記載する情報の正確性、完全性または適時性について、（明示、黙示を問わず）一切の表明、保証または約束を行わないものとし、それらに対する責任を本免責事項によって放棄します。本文書の情報は予告なしに変更されることがあります。

反トラスト法上の通知

本文書の情報は、GSM Association の反トラスト法コンプライアンス方針を全面的に遵守しています。

目次

1 はじめに

1.1 GSMA IoT セキュリティ・ガイドライン文書群

本文書は、黎明期の「モノのインターネット」（IoT）業界における IoT のセキュリティ問題に対する共通の理解を確立する一助となることを目的とした、GSMA による一連のセキュリティ・ガイドライン文書の一部となります。この拘束力のない一連のガイドライン文書は、サービスのライフサイクル全体を通じてセキュリティのベストプラクティスが実装されることを保証にするために、安全性の高い IoT サービスを開発するための方法論を示すもので、IoT サービスにおける一般的なセキュリティへの脅威と脆弱性を軽減する方法についての推奨事項を提示しています。

以下の図は、GSMA セキュリティ・ガイドライン文書群の構成を示しています。「CLP.11 IoT セキュリティ・ガイドライン概要書」[1]を手引きとして参照した後に、CLP.12 [2]および CLP.13 [3]（本文書）へと読み進めることをお勧めします。

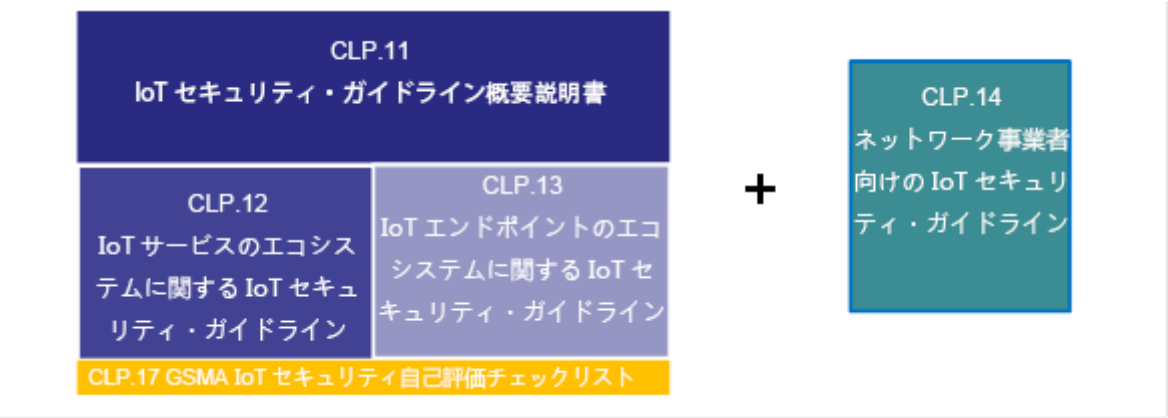

図1 -「GSMA IoT セキュリティ・ガイドライン」文書の構成

IoT エコシステムで活動するネットワーク事業者、IoT サービス提供者およびその他の関連事業者の皆様には、GSMA 文書 CLP.14「ネットワーク事業者のための IoT セキュリティガイドライン」[4]を参照することをお勧めします。同文書は、IoT サービス提供者にサービスを提供しようとしているネットワーク事業者向けに、システムのセキュリティやデータのプライバシーを保証するための最高水準のセキュリティ・ガイドラインを提示しています。

1.1.1 GSMA IoT セキュリティ評価チェックリスト

GSMA 文書 CLP.17 [19]には、評価チェックリストが添付されています。IoT 製品、サービスおよびコンポーネントのサプライヤーは、同チェックリストを使用して自社の製品、サービスおよびコンポーネントが「GSMA IoT セキュリティ・ガイドライン」を遵守しているかどうかを自己評価することができます。

「GSMA IoT セキュリティ評価チェックリスト」[19]に評価を記入することで、サプライヤーはサイバーセキュリティーのリスクから自社の製品、サービスおよびコンポーネントを守るために講じているセキュリティ対策を実証することができます。

同チェックリストへの回答は、記入したチェックリストを GSMA に提出することで完了することができます。チェックリストへの回答手順については、GSMA 公式ウェブサイトを参照してください。

https://www.gsma.com/iot/future-iot-networks/iot-security-guidelines/

1.2 文書の目的

本文書は、IoT エンドポイントデバイスの観点から IoT サービスのコンポーネントを評価するために使用されるものとします。IoT の観点からのエンドポイントとは、インターネット接続された製品やサービスの一部として機能またはタスクを実行する物理コンピューティングデバイスです。エンドポイントは、例えば、ウェアラブルなフィットネス用デバイス、産業用コントロールシステム、自動車用テレマティクスユニット、または個人用ドローンユニットでさえもなり得ます。物理デバイスの操縦に使用されるすべての技術は、セキュリティリスクについて評価される必要があります。その結果は、読者が IoT サービスに対する潜在的なリスクのほとんど全てを特定して修復が可能となる、実践的な設計ガイドライン一式となります。

本文書の適用範囲は、IoT エンドポイントデバイスの設計と実装に関する推奨事項に限定されます。

本文書は、新たな IoT 仕様や標準の作成を促すことを意図したものではなく、現時点で利用可能なソリューション、標準、ベストプラクティスを示すものです。

本文書には、既存の IoT サービスの陳腐化を加速させる意図はありません。ネットワーク事業者の既存 IoT サービスとの下位互換性は、安全性が適切に保証されていると見なされる場合は維持する必要があります。

特定地域の国内法令および規則を遵守することによって、本文書のガイドラインが無効となる場合がありますのでご注意ください。

1.3 想定読者

本文書が想定する主な読者は次の通りです。

- IoT サービス提供者 - 新たに革新的な接続機能を備えた新製品やサービスを開発しようとしている企業または組織。IoT サービス提供者が活動する分野の一例には、スマートホーム、スマートシティー、自動車、輸送、医療、公益事業、家電製品が含まれます。
- IoT デバイス製造業者 - IoT サービス提供者向けに IoT サービス対応の IoT デバイスを提供する業者。
- IoT 開発業者 - IoT サービス提供者向けに IoT サービスの構築を代行する業者。
- IoT サービス提供者にサービスを提供するネットワーク事業者。

1.4 用語の定義

用語	説明
アクセスポイント名	エンドポイントデバイスを取り付けるネットワーク接続ポイントの識別子。異なるサービス種類ごとに決められており、多くの場合、ネットワーク事業者別に設定される。
攻撃者	ハッカー、脅威エージェント、脅威アクター、詐欺師または IoT サービスに対して悪意のある脅威。脅威の発生源として、個々の犯人、組織犯罪、テロ、敵対国およびその代理人、産業スパイ、ハッカー集団、政治活動家、マニアハッカー、研究者、さらには意図的でないセキュリティとプライバシーの侵害などが考えられる。
移動体通信	3GPP 標準化モバイルネットワーク技術（GSM、UMTS、LTE（LTE-M を含む）および NB-IoT）。
クラウド	アプリケーションおよびデータのホスト、保存、管理および処理を行う、インターネット上にあるリモートサーバーのネットワーク。
複合エンドポイント	長距離通信リンク（移動体通信、衛星、イーサネット等の配線接続など）において、バックエンドのサーバーへの継続的な接続性を備えたエンドポイント・モデルの一種。注釈 3 を参照してください。
埋め込み SIM	容易にアクセスまたは交換できず、デバイス内での取り外しや交換を意図せず、安全性の高いプロファイルの変更が可能な SIM。

用語	説明
エンドポイント	IoT エンドポイントとは、インターネット接続された製品やサービスの一部として機能またはタスクを実行する物理コンピューティングデバイスです。IoT デバイスの一般的な 3 つのクラスに関する説明、及びエンドポイントの各クラスの事例については、セクション 3 を参照してください。
モノのインターネット	モノのインターネットとは、複数のネットワークを通じてインターネットに接続された様々なマシン、デバイス、器具が連動して動作することを指します。これらのデバイスには、タブレットや家電製品などの日用品のほか、データの送受信ができるマシンツーマシン同士（M2M）の通信機能を備えた車両、モニター、センサーなどのマシンが含まれます。
IoT サービス	サービスを実行するために IoT デバイスからのデータを利用するコンピュータープログラム。
IoT サービスのエコシステム	フィールドで展開するエンドポイントに機能を提供し、そこからデータを収集するために必要な一連のサービス、プラットフォーム、プロトコルおよびその他の技術。詳細については、CLP.11 [1]を参照。
IoT サービス提供者	新たに革新的な接続機能を備えた IoT 製品やサービスを開発しようとしている企業または組織。
ネットワーク事業者	IoT エンドポイントデバイスを IoT サービスのエコシステムに接続する、通信回線の運営者および所有者。
組織の信頼の基点（Root of Trust）	ID、アプリケーション、通信のセキュリティを暗号によっていかにして確保できるか（確保すべきか）を定める、一連の暗号化ポリシーおよび手順。
サービスのアクセスポイント	通信回線を経由した、IoT サービスのバックエンドにあるインフラストラクチャへのエントリーポイント。
加入者識別モジュール	モバイルネットワークとネットワークサービスへのアクセス時に、デバイス認証のためにモバイルネットワークが使用するスマートカード。
トラストアンカー	階層構造のある暗号システムにおけるトラストアンカーは、信頼が想定され、派生されていない、信頼できるエンティティです。

用語	説明
トラステッドコンピューティングベース	トラステッドコンピューティングベース（TCB）とは、製品またはサービス内のアルゴリズム、ポリシー、および秘密の集合体です。TCB は、製品やサービスが独自の信頼性を測定したり、ネットワークのピアについての確実性を評価したり、製品やサービスが送受信したメッセージの整合性を検証したりできるモジュールとして機能します。TCB は、基盤となるセキュリティプラットフォームとして機能し、安全な製品やサービスを構築することができます。TCB のコンポーネントは、コンテキスト（エンドポイント用ハードウェア TCB、またはクラウドサービス用ソフトウェア TCB）によって変わりますが、抽象的な目標、サービス、手順、およびポリシーは酷似している必要があります。
信頼できる実行環境（TEE）	高性能なオペレーティングシステムと一緒に実行し、そのオペレーティングシステムにセキュリティサービスを提供する環境。TEE を実装する上で使用できる技術は複数あり、その技術に応じて達成されるセキュリティレベルは異なります。
UICC	ETSI TS 102 221（欧州電気通信標準化機構の技術仕様）に規定されている、暗号が異なるセキュリティドメインにおいて複数の標準化されたネットワークまたはサービスの認証アプリケーションをサポートすることができる、セキュアエレメントのプラットフォーム。ETSI TS 102 671 に規定されている埋め込み式要素に埋め込まれることがある。

1.5 略語

用語	説明
3GPP	第 3 世代プロジェクト・パートナーシップ
AC	交流電流
API	アプリケーション・プログラム・インターフェイス
APN	アクセスポイント名
BLE	Bluetooth 低エネルギー
BT	Bluetooth
CLP	GSMA のコネクテッド・リビング・プログラム
CPE	カスタマ構内設備
CPU	中央処理装置
EEPROM	電気的に消去可能なプログラマブル読み取り専用メモリ
eUICC	埋め込み UICC

用語	説明
FIB	集束イオンビーム
GBA	汎用ブートストラッピング・アーキテクチャ
GPS	グローバル・ポジショニング・システム
GSMA	GSM Association
IoT	モノのインターネット
IP	インターネットプロトコル
ISM	産業・科学・医療
LAN	ローカルエリア・ネットワーク
LPWA	ローパワー・ワイドエリアネットワーク
LTE-M	マシン向けロング・ターム・エボリューション
MCU	マイクロコントローラーユニット
NB-IoT	狭帯域モノのインターネット
NVRAM	不揮発性ランダム・アクセス・メモリ
OMA	Open Mobile Alliance
PAN	パーソナル・エリア・ネットワーク
PSK	事前共有キー
RAM	ランダム・アクセス・メモリ
ROM	読み取り専用メモリ
SCADA	監視制御およびデータ収集
SPI	シリアル・ペリフェラル・インタフェース
SSH	セキュアシェル
SIM	加入者識別モジュール
SRAM	スタティック・ランダム・アクセス・メモリ
TCB	トラステッドコンピューティングベース
TTL	トランジスタ-トランジスタロジック
UART	汎用非同期送受信機/送信機

1.6 参考文献

参照	文書番号	タイトル
[1]	CLP.11	IoT Security Guidelines Overview Document
[2]	CLP.12	IoT Security Guidelines for IoT Service Ecosystem
[3]	CLP.13	IoT Security Guidelines for IoT Endpoint Ecosystem
[4]	CLP.14	IoT Security Guidelines for Network Operators
[5]	OMA FUMO	OMA Firmware Update Management Object www.openmobilealliance.org
[6]	該当なし	ST-LINK/V2 in-circuit debugger/programmer http://www.st.com/
[7]	該当なし	Mobile IoT Initiative https://www.gsma.com/iot/mobile-iot-initiative/
[8]	該当なし	Nmap Security Scanner https://nmap.org/
[9]	CLP.03	IoT Device Connection Efficiency Guidelines https://www.gsma.com/iot/gsma-iot-device-connection-efficiency-guidelines/
[10]	該当なし	Federal Information Processing Standards www.nist.gov/itl/fips.cfmwww.nist.gov/itl/fips.cfm
[11]	該当なし	EMVCo www.emvco.com/
[12]	該当なし	SIM Alliance - Open Mobile API simalliance.org/key-technical-releases/
[13]	GPD_SPE_013	GlobalPlatform Secure Element Access Control www.globalplatform.org/specificationsdevice.asp
[14]	GPD_SPE_024	GlobalPlatform Trusted Execution Environment API Specification www.globalplatform.org/specificationsdevice.asp
[15]	GPC_SPE_034	GlobalPlatform Card Specification www.globalplatform.org/specificationscard.asp
[16]	ISO/IEC 29192-1	Information technology -- Security techniques -- Lightweight cryptography www.iso.org/obp/ui/#iso:std:iso-iec:29192:-1:ed-1:v1:en
[17]	TS 33.220	Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)

参照	文書番号	タイトル
		www.3gpp.org
[18]	TS 33.222	Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) www.3gpp.org
[19]	CLP.17	GSMA IoT Security Assessment Checklist https://www.gsma.com/iot/iot-security-assessment/
[20]	TS-0003	oneM2M Security Solutions www.onem2m.org
[21]	3GPP TS33.163	Battery efficient Security for very low Throughput Machine Type Communication (MTC) devices (BEST) www.3GPP.org

2 IoT エンドポイントセキュリティチャレンジ

IoT サービスが提示するセキュリティチャレンジは、多くの場合、サービスが用いる IoT エンドポイントの特性に直接関係しています。例えば、IoT エンドポイントの多くには次のような特性があり、エンドポイントに関連した特有のセキュリティチャレンジがあります。

2.1 低消費電力

- 無停電電源装置がなく、リモートアクセスできないエンドポイントや、デバイスの電源が恒久的だが制限されている（太陽エネルギー供給など）理由により、長いバッテリー寿命（数年）を達成するには、低消費電力が求められる場合があります。
- 低消費電力のエンドポイントは通常、計算処理上単純な暗号操作（例えば、エンドポイントは ISO/IEC 29192 [16]内で定義されている軽量の暗号操作のみだけ対応する場合があります）のみを行うことができ、より高度な暗号操作に関連する高消費電力要件により、制限された帯域幅通信のみに対応し、暗号機能を再び制限することができます。

2.2 低コスト

- 数多くの IoT サービスにおけるビジネスケースでは、IoT エンドポイントのコストを低く抑えることが求められています。これは多くの場合、処理能力が低く、メモリ量の少ない制約されたオペレーティングシステムを搭載しているデバイスとなります。この結果、デバイスは「インターネットグレード」の暗号化を実行できない可能性があります。

2.3 長寿命（10 年以上）

- 多くのエンドポイントは、特に市民向け用途や産業用途（例えば、スマートガスメーター）の場合、長寿命でなければなりません。これは課題を提示しており、デバイスを設計する際に選択した暗号の設計は、デバイスの存続期間にわたって堅牢である必要があるからです。例えば、この 10 年間で攻撃者が利用できる$あたりの処理能力は、16 倍に増加する可能性がありますが、デバイスの能力は停滞する可能性があります。
- 長寿命デバイスの管理は、IoT エンドポイント内でパッチを当てることができないセキュリティ脆弱性が発見された場合も特に問題となります。

2.4 物理的にアクセス可能

- 多くの IoT エンドポイントは、攻撃者に対して物理的にアクセス可能です。従って、このエンドポイント上にあるハードウェアのコンポーネントとインターフェースは、すべて潜在的に攻撃の対象となるので、開発者が保護する必要があります。

上記の結果により、多くの IoT エンドポイントにおいて、IoT エンドポイントは広域通信網に直接接続されておらず、IoT エンドポイントの多くには、インターネットプロトコル（IP）機能がありません。例えば、IoT エンドポイントは、産業・科学・医療（ISM）用の無線トランシーバを用いてデータをローカルの IoT サービスゲートウェイに転送してからデータを取得し、IP を使用して通信ネットワークにそのデータを送信するので、エンドツーエンド通信を確保する工程が複雑になります。

本文書の残りの部分で説明する通り、IoT エンドポイントの機能と関連するセキュリティリスクに応じて、複雑度が変化する様々なセキュリティ方法を適用する必要があります。

3 IoT エンドポイントモデル

物理的な世界と対話し、インターネット上のどこかにサーバーを接続してメトリクスに関するガイダンスと送信を行うといった一連の技術が非常に異種であると見なされると、IoT エンドポイントモデルは劇的に変化しました。現代のエンジニアリングにおける IoT 技術は、いくつかのバリアントのみで構成され、予測可能なモデルへとまとめられました。IoT エンドポイントもより予測可能になりつつあり、いくつかの兆候の内から 1 つだけを利用することが期待されています。

- 軽量エンドポイント
- 複合エンドポイント
- ゲートウェイ（または「ハブ」）

以下の図では、一般的な IoT エンドポイント構成の一部を示しています。

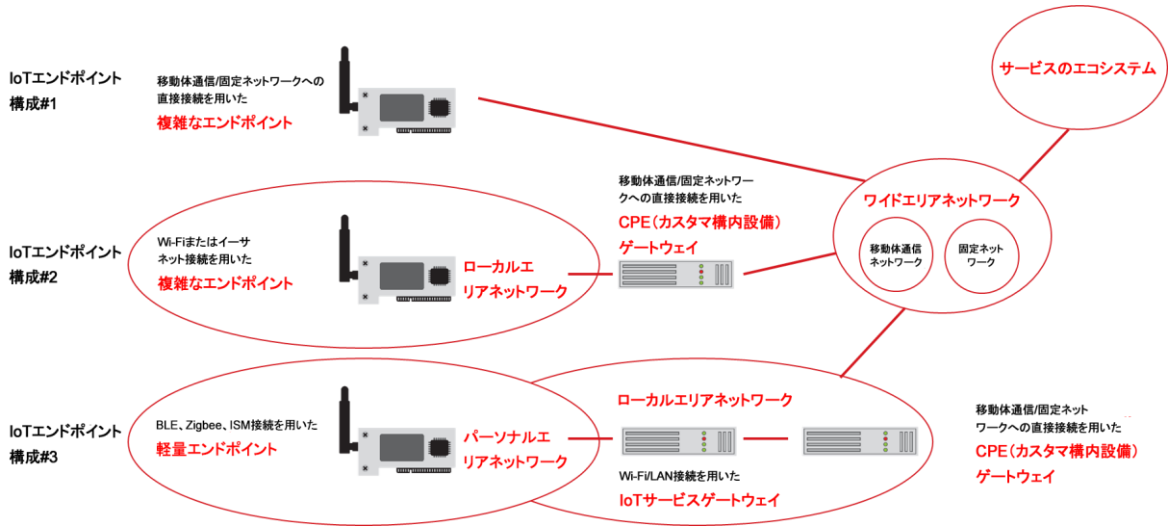

図2 - IoT エンドポイント構成の例

3.1 軽量エンドポイント

このタイプのエンドポイントは通常、照明スイッチやドアロックなど、機能がほとんどないセンサーや単純な物理デバイスで、物理的な目的を 1 つだけ果たし、サービスエコシステムや消費者にメトリクスを提供することが目的です。これは一般的に、Bluetooth Low Energy (BLE)、Thread、または Zigbee など、非常に安価な処理装置、恐らく 8 ビットマイクロコントローラや、接続用の短距離パーソナル・エリア・ネットワーク（PAN）またはキャピラリープロトコルを使用します。これは通常低電力であり、コイン形電池、太陽電池、または小型リチウムポリマー電池で動作することがあります。これらのデバイスは通常、図 2 の「エンドポイント構成#3 の例」で示している通り、IoT サービスのゲートウェイとカスタマ構内設備を介してサービスエコシステムに接続されています。

軽量エンドポイントの例は次の通りです。

- ウェアラブル
- ホームセキュリティセンサー・エンドポイント
- 近接ビーコン
- ノンセルラー式キャピラリーデバイス

軽量エンドポイントのコストが低いため、これらのデバイスで使用できるセキュリティ技術は最低限に抑えられています。回路基板に大量の電流、コスト、またはスペースを必要とするセキュリティ技術は通常、これらのシ

ステムでは利用できません。ただし、軽量エンドポイントは依然として、費用対効果の良い小型のトラストアンカーを用いて堅牢なセキュリティフレームワークを実装することができます。

3.2 複合エンドポイント

このエンドポイントモデルは、通常、移動体通信（LPWA ネットワークを含む）（図 2 の「エンドポイント構成#2」を参照）などの長距離通信リンクを介したバックエンドサーバーへの持続接続を有するか、Wi-Fi やイーサネットをカスタマ構内設備ゲートウェイを介して使用し接続します。このデバイスは、8 ビットマイクロコントローラであっても、基本的なプロセッサを備えていますが、交流（AC）電源に直接接続されるか、電池を含んでおり、電池充電システムへ定期的にアクセスするので、より信頼性の高い処理装置を動作させることができます。一部の複合エンドポイントでは、キャピラリープロトコルを介して通信しますが、ローカルアプリケーションを効率的に実行するには、ストリーミング音声デバイスなどより多くの電力が必要です。

複合エンドポイントの例は次の通りです。

- IoT で接続された照明システム
- 冷蔵庫や洗濯機などの電化製品
- 産業用制御システム（SCADA など）
- 後付 OBD2 移動体通信「コネクテッドカー」の追跡及び監視装置

複合エンドポイントは、より多くの電流を引き込むことができ、通常はより信頼性の高いプロセッサを実装し、セキュリティ技術に利用できる回路基板上に空間を増やします。その結果、複合エンドポイントでははるかに多くのことが可能になります。これらのデバイスは、ほぼあらゆる種類のトラストアンカーを使用できます。結果として、本文書で後述する通り、個別の事前共有鍵（PSK）またはトラステッドコンピューティングベース（TCB）モデルを容易に実装できます。

3.3 ゲートウェイ（または「ハブ」）

ゲートウェイは通常、専用の電源に接続されたデバイスで、概して軽量エンドポイントとそれらを駆動するバックエンドシステム間の通信を管理します。ゲートウェイは移動体通信（LPWA を含む）、衛星、固定系統、ファイバー、イーサネットなどの長距離通信リンクを管理します。サービスエコシステムに存在するバックエンドシステムからのコマンドを受け入れ、そのコマンドを軽量エンドポイントで消費可能なメッセージに変換します。エンドポイント

IoT ゲートウェイの主な機能は、軽量エンドポイントとの間でメッセージをルーティングすることですが、以下のようなクリティカルタスクを実行することもできます。

- デバイスの検出
- ネットワークドライバーの展開
- 管理機能
- ランタイム監視
- GBA や TLS などの認証やセキュリティの設定

ゲートウェイが技術的にエンドポイントとなりますが、エンドユーザーが必ずしも管理する必要はなく、IoT サービス提供者またはネットワーク事業者が管理できます（下記参照）。これに関わらず、ゲートウェイはまた、ローカライズしたネットワーク内の複数の軽量エンドポイントへのアップリンク配布をより有効利用するために、複合エンドポイントとして設計することもできます。

複合エンドポイントと同様に、ゲートウェイはより多くの処理能力、電流引き込みが可能であり、通常は回路基板でより多くの空間を利用できます。これにより、IoT ゲートウェイは複雑なトラステッドコンピューティングベースのソリューションや GBA 認証クライアントなどの技術を、比較的簡単に実装できます。

これらのゲートウェイ属性は、異種のネットワークデバイス間でメッセージをルーティングするために複数の通信技術を組み込むことも可能にします。これにより、通常は効果的にメッセージを交換できないエンドポイント間の通信が可能になります。この方法で、ゲートウェイは、ローカルエコシステム内にあるデバイスの集約ポイントとして機能し、お互いに、必要に応じてネットワークとサービスエコシステムとの通信を可能にします。

通常、「IoT サービスゲートウェイ」と「カスタマ構内設備（CPE）ゲートウェイ」の 2 種類のゲートウェイがあります。その違いを以下で説明します。

1. 「IoT サービスゲートウェイ」は、IoT サービス提供者によって提供されます。エンドユーザーが所有している場合もありますが、通常は IoT サービス提供者が管理しています。このようなゲートウェイは通常、軽量エンドポイントをサービスエコシステムに（固定/移動体通信接続経由で直接か、CPE ゲートウェイ経由のいずれかで）接続するハブとして使用され、エンドユーザーは IoT サービス提供者からマネージドサービスを購入します。
2. 「CPE ゲートウェイ」はネットワーク事業者によって提供されます。これは通常、移動体通信または固定ネットワークでインターネットに接続されたブロードバンドルーターです。これは、住居環境または企業環境で使用できます。この構成では、ゲートウェイの管理と構成は、ネットワーク事業者が行います。

3.4 包括的モデル

どのタイプのエンドポイントが評価または設計されているかに関わらず、ハードウェアとロジスティックの観点から類似したサブコンポーネントモデルを持っています。

- 中央処理装置（CPU）は、アプリケーションコードを実行するはず
- CPU は、データと実行可能コードを永続記憶装置から読み込み/格納する必要がある
- CPU は一時的な記憶装置内にあるデータを計算する必要がある
- トラステッドコンピューティングベースを使用して、環境を認証する必要がある
- デバイスは、IoT エコシステムと通信する必要がある

軽量エンドポイントは複合エンドポイントやゲートウェイよりも記憶容量と計算能力が下回ることが特徴的です。通常はセキュリティ機能もほとんどありません。

包括的モデルの最も重要な側面は、各タイプのエンドポイントデバイスが、特定のアプリケーションを実行するために、信頼性が高く、高品質でセキュアなプラットフォームを定義するプライマリジョブを 1 つ持っているということです。言い換えると、スマートフォン、クラウドサービス、メインフレームなどのより複雑なコンピューティングプラットフォームのように、高品質なアプリケーションが確実に実行できたり、ピアと安全にやり取りする前に、エンジニアチームは、ハードウェアがアプリケーションに対して*信頼できる*プラットフォームを提供していることを確実にする必要があります。

IoT エンドポイントは、本来、他のエンドポイントのネットワークに関与します。監視サービスの影響や関与のないアクションを実行するスタンドアロンのデバイスではありません。特定のデバイスの信頼性を高め、セキュリティや信頼性のギャップによる責任の可能性を軽減するため、すべてのエンドポイントは、IoT エコシステム全体の*信頼性*がエンドポイントのハードウェアを構築することで*始まる*という考えで設計される必要があります。

この観点を念頭に置いた場合、最も簡単に開発できるタイプのエンドポイントデバイスでさえも、最終的には数百万のデバイスの大きさに及ぶ可能性のあるネットワークに関与することが見込まれるので、信頼性が高く、高品質でセキュアな方法で動作する必要があることは明らかです。単一のエンドポイントが動作する方法は、IoT エコシステム全体へ確実に影響を与えます。その結果、エンジニアたちは、組み込まれた特定のデバイスに関連する物理的属性をはるかに超えたアーキテクチャ設計の影響を考慮する必要があります。エンジニアたちは、IoT エコシステム全体のセキュリティ、信頼性、品質のニーズについて考える必要があります。

4 セキュリティモデル

エンドポイントにおけるセキュリティは、コンポーネントの観点から評価できます。特定のエンドポイントを構築する上で必要な各コンポーネントを評価することで、エンジニアや敵対者は多大な労力をかけずに、全システム侵害という結果をもたらす可能性のある一連の攻撃を構築できます。

上記で定義したエンドポイントの包括的モデルを使用すると、使用されるコンポーネントを高いレベルから評価できます。各コンポーネントの高水準な観点は、アナリストを一般的に使用され、不適切に保護される可能性の高い技術へと向かわせるでしょう。最低限の専門知識、機器、成功するために必要な費用のコンポーネントを優先することで、アナリストや敵対者は、特定のエンドポイントを迅速に評価してセキュリティの欠陥を発見できる攻撃モデルを構築できます。

エンドポイントエコシステムでは、敵対者がリソース、インフラストラクチャへのアクセス、そして専門知識に応じて調査する脅威の側面がいくつかあります。これら脅威の側面は次の通りです。

- ネットワーク通信
- アクセス可能なネットワークサービス
- コンソールアクセス
- ローカルバス通信
- チップアクセス

4.1 ネットワーク通信攻撃

IoT エンドポイントを侵害しようとする最も単純な第 1 の手順は通常、通信モデルの弱点を含んでいます。アナリストたちは、通信モデルが通信セキュリティのベストプラクティスを組み込んでいるかどうかを観察します。アナリストがログイン資格情報、通信トークン、またはエンドポイントを識別するためにサービスエコシステムが使用するその他の識別子を簡単に取得できる場合、それらがデバイスを侵害しています。

この戦略は、非常に単純なものから極めて難しいものへと変動します。この理由は、アナリストや敵対者が通信チャネルを通過するプレーンテキストデータにアクセスするためです。十分に装備したアナリストは、BLE、802.15.4 などの人気のあるプロトコルの通信を傍受する技術をすでに持っています。エンドポイントの通信に対する監視や、介入者攻撃を実行することは、通常エンドポイントの変更をほとんど必要としないため、敵対者は非常に有益な立場にあります。この種の攻撃を実装するには、手間と労力をほとんど必要としません。

しかしながら、通信モデルがベストプラクティスを用いてデータの機密性と整合性を実施する場合、敵対者は貴重な秘密にアクセスするのが指数関数的に困難な時間となります。これにより、敵対者は次に簡単な攻撃モデルに移行します。

4.2 アクセス可能なネットワークサービスの攻撃

IoT エンドポイントを攻撃する次のステップは、オープンなネットワークサービスの評価です。最初のステップでは、エンドポイントから生じるアウトバウンドメッセージがキャプチャされ、すぐに使用できる秘密がメッセージ内でアクセス可能かどうかが識別されます。これにより、敵対者はエンドポイント自体から秘密を抽出する際に必要な作業量を減らすことができます。アウトバウンド通信のセキュリティモデルが安定している場合、ネットワークサービスをスキャンして、ネットワークからエンドポイントのオペレーティングシステムにアクセスできるまたは、機器を搭載しているかどうかを評価します。

ネットワークポートがオープンかどうかを判断するには、NMap [8]などのツールを用いて評価を実施します。ネットワークトポロジが、BLE や IEEE 802.15.4 ネットワークで一般的な IP ケーブルでない場合、敵対者はすぐにアクセスできるツールを引き続き使用し、適切な無線プロトコルを介してエンドポイントに接続します。

敵対者は次に、エンドポイントにメッセージを送信することを試みて、エンドポイントがコマンドの実行またはオペレーティングシステムへのリモートコンソールアクセスが実現できるかどうかを判断します。一般的な方法は、セキュアシェル（SSH）や Telnet などのネットワークログインインターフェースが利用可能かどうかを評価することです。デフォルトのログイン資格情報が使用されている場合、敵対者はエンドポイントにログインできる可能性があります。これにより、敵対者はローカルのオペレーティングシステムを操作し、潜在的にローカルの脆弱性を悪用して、権限を上げ、デバイスから秘密を抽出します。

別の一般的な事例には、不完全に設計された Web サービスの悪用が挙げられます。これは、ユーザー入力フィールドから制御文字を適切に取り除かない Common Gateway Interface（CGI）スクリプトを介してコマンドを注入し、ローカルのオペレーティングシステム上でコードを実行します。

4.3 コンソールアクセス攻撃

コンソールアクセスは厳密には攻撃ではなく、戦略です。通常、開発者と品質保証（QA）技術者にハードウェアまたはソフトウェアの以上を診断する機能を提供するためにエンドポイントでコンソールを有効にする必要があります。しかし、コンソールから提供される情報は敵対者にとって非常に貴重です。さらに、コンソールは敵対者にローカルおよびリモートの両方でエンドポイントにログインできる機能を提供する可能性があります。

通常、ローカルのハードウェアコンソールは、次の方法でエンドポイントデバイス上で検出することができます。

- TTL シリアルポートを示す回路基板上の 5 ピンヘッダーを探します。
- CPU または MCU の仕様を調べ、UART ピンを識別します。

ピンが TTL の標準電圧仕様に準拠するため、マルチメーターを使用して TTL ポートを識別することができます。あるいは、ロジックアナライザを使用して、ハードウェアピンを通過するシリアルデータのボーレートを推測することができます。アナリストは、コンソールがローカルのハードウェア上で利用可能かどうかを素早く見分けることができます。

多くの場合、コンソールポートにアクセスするだけで、アナリストはエンドポイントデバイスのコマンドプロンプトに直接アクセスできます。それ以外の場合では、ログイン資格情報が必要ですが、通常は推測できます。インターネット上で他の個人がログイン資格情報を識別し、すべてのエンドポイントのログイン資格情報が同じの場合、アナリストが行わなければならないことは、オンラインで Google 検索を実行して他の誰かが資格情報を掲載しているかどうかを確認するだけです。

リモートコンソールへのアクセスは、診断ネットワークプロトコル、コンソールアクセスプロトコル（例えば、SSH や telnet）、またはほかの手段を通じて取得することができます。これらのアクセス方法は、敵対者がアクセスチャンネルを操作でき、それによってリモートコンソールへ敵対するアクセスが許可されるかどうかを判断するために評価する必要があります。

4.4 ローカルバス通信攻撃

コンソールからコマンドプロンプトを取得できない場合、敵対者またはアナリストはハードウェアの調査を開始して、エンドポイントが簡単に侵入されているかどうか判断する必要があります。これには様々な方式がありますが、利用しやすい手順があります。

- 書き込み可能なメディアが存在し、変更可能である
- 暗号化された秘密がハードウェアバスを介してクリアに渡される
- ハードウェア回路にメッセージを挿入し、敵対者に有利に動くようにアプリケーションまたはオペレーティングシステムの動作に影響を与えることができる

1 番シンプルな攻撃は、書き込みメディアが存在するかどうかを特定することです。これは、書き込み可能な外部メモリ（SD/MMC）カードなど、変更しやすいメディアである可能性があります。あるいは、アプリケーションや構成を変更して NVRAM チップや EEPROM を変更することにより、コマンドプロンプトへのアクセスや安全に格納されたトークンへのアクセスが可能になります。

このベクターが適切に保護されている場合、アナリストは暗号化された秘密がハードウェアバスを介してクリアに渡されるかどうかを判断します。これには、ロジックアナライザを使用して、EEPROM と CPU、マイクロコントローラと SPI 接続ネットワークアダプタ、または他の攻撃との間のメッセージを傍受することが含まれます。これらの攻撃は、攻撃の複雑性や悪用される技術によっては、極めて単純で即効性のあるものから、複雑で高価なものまであります。

敵対者が上記の方法を使用して貴重な秘密を傍受できない場合、ハードウェアバスにメッセージを注入して、エンドポイント上で実行されているアプリケーションの動作を変更しようとする可能性があります。これは難解な攻撃で、高度な専門知識、機器、そしてアプリケーション固有のデータとそのコンテキストを評価できる能力を必要とします。

4.5 チップアクセス攻撃

上記の攻撃が複雑すぎるか高価すぎる場合、敵対者はハードウェアに対してさらに複雑な攻撃に移行する必要があります。これには通常、チップまたは回路基板上にある様々なコンポーネントのセキュリティを悪用することを含みます。これには次の内容が含まれます。

- マイクロコントローラまたは CPU のデキャッピング
- 内部 EEPROM または NVRAM からの秘密の抽出
- 内部 SRAM メッセージのインターセプト
- X 線解析または FIB リバースエンジニアリングの実行

これらの攻撃にはすべて、高度なスキル、電子工学の知識、そして高価な機器が必要です。攻撃者がこれらの方法を利用して製品をリバースエンジニアリングすることを大部分の組織は恐れる必要はありませんが、依然として考慮すべき重大な可能性です。その理由は、エンドポイントデバイスに一意の暗号秘密がプロビジョニングされていな場合、これらの攻撃を *1 度*だけ実行すればいいからです。

一意の暗号秘密がプロビジョニングされていない場合、このクラスの攻撃 1 つが製品ライン全体に影響する可能性のある秘密を抽出します。これは重大なリスクとなります。データが何らかの理由で一般に公開されると、技術はパッチがリリースされるまで、攻撃と悪用の対象となるからです。1 つリリースできれば対象とはならなくなります。

5 セキュリティに関するよくある質問

本書では、優先度に基づいてエンドポイントに関するセキュリティを推奨事項に分類しています。しかし、実用向きとして実質的な出発点から推奨事項を評価するほうが有益です。エンジニアは通常、技術目標またはビジネスに影響された目標に基づいて推奨事項のリストを作成し始めます。このセクションでは、エンドポイントの観点から見た共通の目標と、これらの目標の達成に向けた推奨事項の概要を説明します。

5.1 クローニングにどう立ち向かいますか。

知的財産の保護は、現代のビジネスにとって重要な目標です。エンドポイント製品を製造するために使用されるハードウェア技術、ファームウェア技術、そして通信技術は、時間や専門知識、そしてお金が必要ですが、企業は、良心的でない企業ブランドや事業を安易に構築したくはありません。しかし、企業が何をしようとも、誰かが全く同じハードウェアコンポーネントを使用して、特定の製品によく似た「盗作」や「クローン」を作ります。適法契約やパートナーシップの外側でこれを防ぐために企業ができることは何もありません。しかし、誰かにこのようなクローンを*使用*させないための費用対効果の高い方法があります。

エンドポイント通信に認証を構築することにより、各エンドポイントが IoT サービス提供者によって製造されていることが暗号で証明されます。バックエンドサービスまたはピアエンドポイントがエンドポイントデバイスと通信するたびに、エンドポイント自体の認証を強制することにより、有効なエンドポイントと*クローン*を区別することができます。デバイスがそうすることができない場合、ピアまたはサービスはエンドポイントを拒否できます。これを機能させるには、次の推奨事項が必要です。

- エンドポイント ID の認証
- 不適切な設計または未実装の相互認証

5.2 エンドポイント ID をどのように保護すべきか。

エンドポイントを適切に認証するには、エンジニアはエンドポイントの暗号化 ID を信用できなければなりません。これは見た目よりも複雑で、目標を達成するためのプロセス、ポリシー、技術の組み合わせが必要です。これは、「*トラステッドコンピューティングベースの推奨事項の実装*」でさらに詳細に述べていますが、エンドポイントで認証トークンをエンコードする方法によって、システム全体の安全性が決まります。

エンドポイントアーキテクチャの多くでは、敵対者は偽装するために、対象デバイスから暗号トークン（存在する場合）をたやすくコピーできます。IoT サービス提供者が製造した各エンドポイントが同じ暗号トークンセッ

トを活用する場合、敵対者は 1 つのトークンセットに侵入することであらゆるデバイスをたやすく偽装することができる場合があります。

従って、適切な TCB を構築するには、次の推奨事項が必要です。

- トラステッドコンピューティングベース（TCB）の実装
- トラストアンカーの活用
- 耐タンパートラストアンカーの使用
- TCB 用 API の活用
- 実績のある乱数ジェネレーターの使用
- 耐タンパー製品ケーシングの使用
- トラストアンカー間の機密性と整合性の強化

5.3 トラストアンカーに対する攻撃の影響をどのように低減しますか。

デバイスの製造およびプロビジョニングの方法では、生産におけるエンドポイントのセキュリティに強烈な影響を与えることに注意することも重要です。製造プロセスでは、エンドポイントが鍵で安全にエンコードされているかどうかを判断します。フルフィルメントおよびプロビジョニングのプロセスが、エンドポイントが特定のコンシューマーにどのように関連付けられているか、そして関連付けが作成される前また後にデバイスが侵害されるかどうかを決定します。

- サプライチェーンセキュリティの考慮
- フルフィルメントの前に各エンドポイントデバイスをカスタマイズ
- 各エンドポイントを一意にプロビジョニング
- プライバシーおよび一意のエンドポイント識別子

5.4 エンドポイントが偽装される可能性をどのように低減するか。

ビジネス上の理由からデバイスを複製した後の敵対者の観点からの望ましい攻撃は、人物または *特定の*デバイスの偽造です。これは、特定の個人の攻撃に直接関連している場合と、そうでない場合があります。これは、Bluetooth 対応のデジタルロックなどのセキュリティ制御の回避を目的としたデバイスの偽装にすぎません。

その根拠に関わらず、この攻撃に立ち向かうには、TCB、個人用設定、認証、そして以下の内容も使用して達成することができます。

- 完ぺき前方秘匿性（Perfect　Forward Secrecy）
- メモリのクリティカルセクションをロック

5.5　サービスやピアを偽装できる機能をどのように禁ずるか。

すべての IoT ネットワークは、エンドポイントデバイスだけでなく、ネットワークサービスとピアで構成されています。エンドポイントは、サービス別に認証される必要がありますが、サービスはエンドポイントによっても認証される必要があります。これにより、アプリケーションのアップデートなどの重要なサービスが妨害されてネットワークをさらに侵害することができなくなります。

- エンドポイント通信セキュリティ
- 完ぺき前方秘匿性（Perfect　Forward Secrecy）
- 実績のある乱数ジェネレーターの使用
- アプリケーションの OTA アップデート
- 不適切な設計または未実装の相互認証
- 不正なメタデータの収集

5.6 ファームウェアとソフトウェアの改ざんをどのように禁ずるか。

信頼のルートが確立されると、エンドポイントは信頼できるコンポーネントから認証できます。そうすることにより、エンドポイントは信頼の基準を確立でき、次のステージのアプリケーションが敵対者によって意図せずに（例えば、欠陥のある NVRAM を介して）または意図的に変更されていないことが保証できます。これは以下によって実現します。

- 実行可能な最小プラットフォーム（アプリケーションロールバック）
- 暗号署名アプリケーションイメージ
- 内部 EEPROM の外部へのブートローディング
- メモリのクリティカルセクションをロック
- 不安定なブートローダ
- 耐タンパー製品ケーシングの使用

5.7 リモートでコードが実行される可能性をどのように減らすか。

物理的なファームウェアまたはソフトウェアを改ざんしても適切な結果が得られない場合、敵対者はバスやネットワークのインターフェースを介して通信するブートローダやアプリケーションに対するコードの実行など、より複雑な攻撃に移行する場合があります。ネットワーク内にあるすべてのピアが認証されている場合、この章で前述したように、敵対者にとって悪質なコンテンツを注入するのははるかに困難になります。しかし、ほとんどのデバイスは、他の組織のデバイスとやり取りするために公衆通信に類似したものが必要です。従って、データ元に適切な制限を行うことはできない可能性があります。

それ故、リモートインターフェースと物理インターフェースの双方からコンピューターシステムへのデータ侵入を徹底的に精査する必要があります。アプリケーションが搾取される可能性を制限し、アプリケーションが侵害された後の発覚を制限するには、次の点を考慮してください。

- メモリ保護の強化
- 秘密用内部メモリの使用
- アプリケーションの OTA アップデート
- 適切な権限レベルでのアプリケーションの実行
- アプリケーションアーキテクチャにおける職務分離の実施
- 言語セキュリティの強化

- オペレーティングシステムレベルのセキュリティ強化実施
- ユーザーインターフェースのセキュリティ
- サードパーティコードの監査

5.8 アーキテクチャの不正なデバックやインストルメント化をどのように禁ずるか。

アーキテクチャの知識を持ち、デバックツールへアクセスできる攻撃者は通常、標準のデバックや診断ユーティリティのインストルメント化を試みて、システムの秘密にアクセスしたり、有益なコードを変更したり注入したりします。これを行う敵対者の能力を制限することにより、コンシューマーによって検出されない可能性のある、高速かつ内密の攻撃の可能性を減らします。

- 耐タンパートラストアンカーの使用
- ログおよび診断
- メモリのクリティカルセクションをロック
- 異常検知
- 耐タンパー製品ケーシングの使用
- デバックとテスト技術の無効化
- ユーザーインターフェースのセキュリティ

5.9 サイドチャネル攻撃をどのように処理すべきか。

敵対者が典型的な選択肢から外れている場合、彼らはデバイスから秘密を抽出するべく、より難解な攻撃に関心を向けます。これらの攻撃は、動作パターンが 1 か 0 などの値、または特定の命令と同等かどうかを確かめるためにハードウェアがどのように動作するかを評価します。これにより、時間の経過と共に、アナリストは埋め込みシステムで処理されたデータをリバースエンジニアリングすることができます。

また、敵対者は高価な解析技術を用いてデバイスから秘密を抽出したり、シリコンのセキュリティ層を介して接続をブリッジする極小の回路を構築したりすることができます。これらの攻撃に立ち向かうのは極めて困難ですが、実装者が攻撃を阻止するために行うことができるものがいくつかあります。

- フルフィルメントの前に各エンドポイントデバイスをカスタマイズ
- 秘密用内部メモリの使用
- 耐タンパー製品ケーシングの使用
- 周辺型攻撃を介して汚染されたメモリ

- 環境ロックアウトしきい値の実装
- 電力警告しきい値の実行
- デバイスの廃止と段階的廃止
- シャドウコンポーネントと信頼できないブリッジの無効化
- コールドブート攻撃の無効化
- 集束イオンビームと X 線への対抗

5.10 セキュアなリモート管理をどのように実装するべきか。

リモート管理は、IoT エンドポイントライフサイクルの重要部分であり、管理に使用されるチャネルが悪用されないよう、確実に保護する必要があります。これは、不明なサードパーティの敵対者による問題ではありません。コンシューマーのサークル内や IoT サービス提供者内でも内部での不正使用が発生する可能性があります。

- エンドポイントパスワード管理
- リモートエンドポイント管理
- ログおよび診断
- 完ぺき前方秘匿性（Perfect　Forward Secrecy）
- プライベート APN の使用

5.11 侵害されたエンドポイントをどのように検出するのか。

エンドポイントのアーキテクチャによっては、デバイスが正常に動作している場合にハードウェアまたはファームウェアが改ざんされているかどうかを判断するのはほとんど不可能です。ただし、異常が検出された場合、インフラストラクチャが追跡、ログ記録、および警告を行っている限り、侵害されたデバイスは異常な動作によって検出できます。以下のような推奨事項を考えてみましょう。

- 異常検知
- 耐タンパー製品ケーシングの使用
- 電力警告しきい値の実行

5.12 バックエンド接続なしでデバイスをどのように安全に展開するのか。

バンクエンド環境への接続が利用可能でも要求されてもいない特定の時間があります。これらの環境においては、セキュリティはより困難になります。それはセキュリティ、ユーザー情報、および動的な認証メカニズムを管理できないことが明らかになるからです。しかし、かなりのセキュリティレベルを実現することができます。以下のような点を考えてみましょう。

- トラステッドコンピューティングベース（TCB）の実装
- 組織の信頼の基点（Root of Trust）の定義
- フルフィルメントの前に各エンドポイントデバイスをカスタマイズ
- 完ぺき前方秘匿性（Perfect　Forward Secrecy）
- エンドポイント ID の認証
- バックエンド接続のない環境

5.13 コンシューマーのプライバシーをどのように確保するのか。

コンシューマーのプライバシーは、エンドポイントの技術だけでなく、IoT 製品またはサービス全体の詳細な解析が必要な複雑な問題です。システム全体の各コンポーネントは、プライバシーの潜在的なギャップを解析する必要があります。以下の推奨事項を検討して、プライバシーを強化する上でのより詳細な洞察を得てください。

- 完ぺき前方秘匿性（Perfect　Forward Secrecy）
- エンドポイント通信セキュリティ
- プライバシー管理
- プライバシーおよび一意のエンドポイント ID
- プライベート APN の活用
- 不正なメタデータの収集
- 不明確なセキュリティリスク（壁を通して見る）
- 合法的傍受

5.14 プライバシーとセキュリティを強化しながらユーザーの安全をどのように確保するのか。

安全性は、アプリケーション、その目的、アプリケーションが居住する予定の環境、コンシューマーの種類、使用される通信技術に関連して考慮する必要があるトピックです。安全性とセキュリティの間にトレードオフが

あるように見えることがよくあります。しかし、これは事実ではないかもしれません。それよりも、安全性およびセキュリティの両方を維持するために、アーキテクチャモデルをシフトする必要があるかもしれません。可能であれば、安全性のためにセキュリティを捨てるべきではありません。可能であれば、両方とも強化すべきです。これは哲学的な推奨事項ですが、安全性はエンジニアリングチームによって絶えずレビューされることが重要です。IoT における安全性について議論し始める上で、以下の推奨事項について考えてみましょう。

- 安全性に関するクリティカル解析
- 意図的なサービス妨害および意図的でないサービス妨害
- 合法的傍受
- サプライチェーンセキュリティの考慮

5.15 解決を期待すべきでない問題は何か。

あらゆるシステムにおいて、物理法則、コスト、または単に技術的なソリューションの欠如により解決できないリスクがあります。これら問題の一部をここに記載します。

- 意図的なサービス妨害および意図的でないサービス妨害
- シャドウコンポーネントと信頼できないブリッジの無効化
- 不明確なセキュリティリスク（壁を通して見る）
- 集束イオンビームと X 線への対抗
- サプライチェーンセキュリティの考慮
- 合法的傍受

6 重要な推奨事項

安全なエンドポイントを開発する際、以下の推奨事項は常に実装する必要があります。以下の重要な推奨事項は、安全なエンドポイントのアーキテクチャを定義します。これらの推奨事項がなければ、エンドポイントは敵対者が悪用する不完全なセキュリティプロファイルを持つようになります。

6.1 エンドポイントトラステッドコンピューティングベースの実装

埋め込みシステムを保護する際の最初の手順は、トラステッドコンピューティングベース（TCB）の定義です。エンドポイント（または類似した埋め込みデバイス）という状況下において、TCB は、エンドポイントの整合性を確保し、ネットワークピアと相互認証を実行し、通信およびアプリケーションのセキュリティを管理するハードウェア、ソフトウェア、およびプロトコルで構成されるスィートです。

TCB のコアはトラストアンカーで、事前共有鍵（PSK）や非対称鍵などの暗号鍵を格納して処理するセキュアなハードウェア技術です。UICC などのトラストアンカーは、ネットワーク通信中のピアの認証に使用されるだけでなく、エンドポイントのアプリケーションセキュリティに役立つデータを格納するために拡張することができます。

トラストアンカーが選択され、エンドポイントのソリューションに統合されると、TCB スイート全体にトラストアンカーを統合するライブラリを選択または設計することができます。TCB により、オペレーティングシステムおよびエンドポイントのプライマリアプリケーションは、デバイスだけでなく、ネットワーク全体のセキュリティをより簡単に管理できるようになります。

エンジニアリングチームは、ソリューションに対して正しいトラストアンカーを選択することが重要です。トラストアンカーと TCB のそれぞれの組み合わせによってセキュリティレベルが異なるからです。一部の組み合わせおよびトラストアンカーの実装は誤ったセキュリティをもたらします。

トラステッドコンピューティングベースの最も一般的なバリエーションは、「最も安全でない」から「最も安全である」の順となっています。

- 未実装（プレーンテキスト）
- 静的事前共有鍵（PSK）
- 静的公開鍵
- カスタマイズされた PSK
- カスタマイズされた静的公開鍵

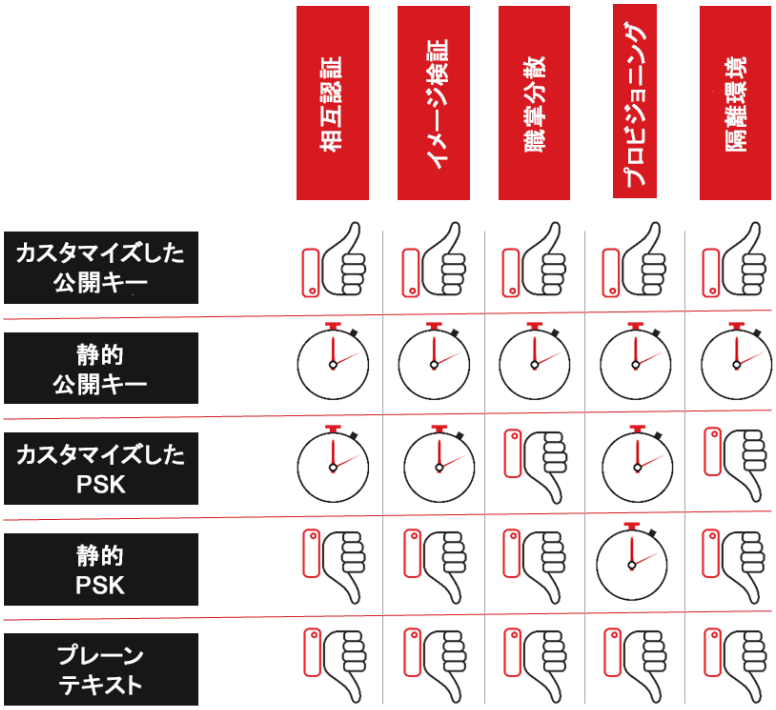

図3 - 各 TCB タイプ毎に提供されるセキュリティ保証

上記の図を考えてみましょう。この図では、各 TCB バリアントの機能に重みが与えられています。親指を下げているアイコンは、TCB モデルが一番上の行に示されたセキュリティ戦略に対応できないことを示しています。ストップウォッチのアイコンは、セキュリティ戦略を使用できることを示していますが、適当な時間内でセキュリティへ不法侵入される可能性があります。親指を上げているアイコンは、セキュリティ戦略を着実に実装できること、およびセキュリティ戦略の有効期間は恐らく永続するだろうということを示しています。

TCB を使用して IoT 製品およびサービス全体における多くの側面を守ることができますが、本書では、以下の 5 つの中心概念に焦点を当てます。

- 実行可能イメージの検証
- ネットワークピアの相互認証
- IoT セキュリティアーキテクチャ内の職務分離
- プロビジョニングおよびパーソナル化
- 分離された環境セキュリティ（またはコネクションレスサイトセキュリティ）

*実行可能イメージの検証*を実装する TCB は、デバイスがロードして実行する各実行可能イメージを暗号で検証することでエンドポイントデバイスを保護します。このプロセスは、ブートローダから始まります。次の実行段階、通常はオペレーティングシステムカーネルを暗号で検証する必要があります。ブートローダはオペレーティングシステムのイメージ、または NVRAM に格納されたファームウェアアプリケーションのイメージも検証できます。

*ネットワークピアの相互認証*を実装する TCB は、ネットワークコンポーネントの認証に信頼のルートを提供し、ネットワークピアに対して自身を暗号で認証します。これにより、ネットワーク上のピアが存在を要求する ID を示す可能性が高くなります。例えば、ネットワークピアがファームウェア更新サービスの提供を要求する場合、TCB はファームウェア更新をピアから受け入れる前に、IoT サービス提供者のコアなネットワークの一部としてピアを認証します。

*職務分離*を実装する TCB は鍵の階層を使用して、IoT サービス提供者の製品内の様々なコンポーネントまたはサービスを識別します。例えば、1 セットの暗号化キーは、ファームウェア更新サービスを表しながら、2 番目の暗号化キーセットは「プッシュ」サービスを表すことができます。これらのサービスは完全に異なる機能を持つため、通信に対して同じ暗号化キーと ID を使用すべきではありません。そのため、TCB は各 ID を管理・検証して、1 つのサービスまたは機能を別のものから分離する必要があります。これにより、暗号化キーの *1 つ*が侵害された場合、IoT サービスインフラストラクチャ全体を侵害する敵対者の能力が低減します。言い換えると、攻撃者が「プッシュサービス」の鍵を侵害すると、ファームウェア更新サービスを偽装する能力も持たないことになります。

*パーソナル化およびプロビジョニング*を実装する TCB は、エンドポイントがそのタイプの他のエンドポイントから暗号的に一意の ID を持つことを保証します。また、プライバシー漏洩や追跡の可能性を減らすため、すべての通信 ID を確実に保護します。

*分離した環境のセキュリティ*を実装する TCB は、プロセスの助けとなるバックエンドサービスがなくても、ピアの信頼性およびデータの機密性と整合性を検証するポリシーおよび手順を強化します。つまり、バックエンドサービスへの通信が長時間切断される場合、ローカライズされた IoT エコシステムは、それでも高度のセキュリテ

ィで機能することができます。分離された環境の整合性は、時間の経過と共に低下しますが、*分離された環境セキュリティ*を実装する、しっかりと設計された TCB はネットワークの回復性を強化し、環境が安全とみなすことができる時間を長くします。

これに関連して、*カスタマイズした*ものは、特定のトラストアンカーに関連付けされる一意の鍵セットを示します。パーソナル化のプロセスには、固有の鍵の生成およびインストール、鍵と一意のチップとの関連付け、適切な権限に対してこの情報と関連メタデータを安全に配布することが含まれます。これにより、各チップには一意の暗号化 ID が確保されます。これに関連して、*静的*とは、すべてのエンドポイントで使用される鍵と同じセットの鍵を指します。

TCB を使用して埋め込みシステムが抱えるほとんどのセキュリティ問題を解決できますが、TCB が解決できなければならない中核となる問題がいくつかあります

- エンドポイントアプリケーションのイメージ検証
- ネットワーク認証および/またはピア認証
- 職務分離
- プロビジョニングおよびパーソナル化
- 分離された環境（コネクションレスサイト）のプロビジョニングおよび通信
- ランダム化

TCB を実装しないことを選択すると、セキュリティが欠如することは明らかですが、対処すべき他の一般的な TCB の実装には微妙な点があります。これらの微妙な点に対処しなければ、セキュリティに相当なギャップが生じる場合があります。

6.1.1 トラストアンカーの鍵モデル

6.1.1.1 静的キー

静的キーの実装は、PSK であろうと非対称鍵であってもすべてのエンドポイントが同じ暗号秘密を活用して特定の問題を解決するソリューションとして定義されています。異なる鍵を使用してコアとなる様々な問題を解決することができますが、それでも鍵は各エンドポイントに対して同じセットとなります。

このモデルは、TCB が解決した各問題が効果的に行われているという点で安全と思われます。しかし、ソリューション全体の寿命は、長いものから極端に短いものまで様々あります。トラストアンカーのセキュリティおよ

び選択した暗号アルゴリズムおよび鍵サイズに応じて、敵対者は間髪入れずにソリューションを中断できるかもしれません。

キーの単一侵害により、すべてのエンドポイントシステムが侵害されるという点で、問題は実際に発生します。これは TCB の実装を低く評価し、エンドポイントおよび IoT アーキテクチャ全体でソリューションを実装するために使った時間とコストを否定します。従って、このモデルは事実上、時限爆弾として実装する危険な TCB なのです。

6.1.1.2 パーソナル化したキー

PSK または非対称ソリューションが実装されているかどうかに関わらず、TCB が効果的に機能するためにはパーソナル化が不可欠です。パーソナル化は、侵害したトラストアンカーを使用して IoT エコシステム全体のセキュリティを破壊する敵対者の能力を無効にします。敵対者が一度に単一のエンドポイントだけを侵害でき、それを行うために物理的なアクセスが必要な場合、IoT 技術の幅広い侵害を実装するには極めて遅く、高価で複雑になります。これはビジネスにとって意義ある勝利です。

過去数十年にわたって進化してきた移動体通信の基準により、ネットワーク事業者は UICC など、トラストアンカーのパーソナル化に対する PSK モデルを完成させました。この結果、IoT エンドポイントによるアプリケーショントラストアンカーとして機能するように UICC をプロビジョニングすることができるときがあり、IoT アプリケーションに対して費用対効果の高いセキュリティソリューションを形成するのに役立ちます。近い将来 eUICC が利用可能になると、現場にすでに展開されている eUICC にでもこの機能を有効にすることができます。

今日、パーソナル化した鍵の技術は、トラストアンカーにとって最も効果的なセキュリティソリューションとなっています。今日の IoT に実装された TCB は、パーソナル化した TCB ソリューションに基づいているはずです。IoT サービス提供者は、ネットワーク事業者と話し合いを持ち、UICC または SIM がアプリケーションレイヤーのトラストアンカーとして実装できるかどうかを判断する必要があります。

6.1.2 TCB プロトコルおよび技術

TCB はトラストアンカーと併せて、プロトコル、ポリシー、ソフトウェアライブラリを組み込んで、IoT 製品やサービスにセキュリティを提供する必要があります。移動体通信が支えている標準トラストアンカーを活用するメリットの 1 つは、ネットワーク事業者向けにすでに存在するプロビジョニングおよびパーソナル化したソフトウェアを投入できることです。以下のような技術、プロトコル、およびスイートは、エンドポイントをネットワークに認証することができる TCB の機能を支援します。

- oneM2M TS-0003 で指定された oneM2M SM UICC アプリケーション

- 汎用ブートストラッピング・アーキテクチャ（GBA）3GPP TS 33.220（付属書 A 参照）

これらの技術を使用すると、長年にわたり経験豊富なエンジニアやセキュリティアナリストがライブラリとプロトコルを綿密に調べているので、プロビジョニングおよびパーソナル化の実装を促進することができます。しかし、これらのプロトコルは、TCB がエンドポイントのアプリケーションを検証することや、エンドポイントがメッセージを適切に認証すること、またはアクションを承認することが完全にできない可能性があります。TCB はファームウェアの検証、OTA アップデートメッセージの検証などのタスクを成し遂げるために、他のプロトコルを組み込む必要があります。

近い将来、eUICC などの技術は、アプリケーションの観点からその機能を増強し、先を見越した UICC は、ネットワークセキュリティを管理しながらエンドポイント自体をブートストラップできるデュアルユース技術を可能にします。これは、ネットワーク事業者が IoT サービス提供者に代わり eUICC デバイスをリモートかつ安全に管理できるので、重要な増強となります。さらに、GlobalPlatform Card Specification [15]で指定された Confidential Card Content Management 機能では、ネットワーク事業者が許可して要る場合、IoT サービスエコシステムにおけるいくつかのアクターが互いに独立して独自のアプリケーションを管理することができます。

6.1.3 リスク

TCB を実装しないことを選択することは、IoT アーキテクチャ全体に対する重大な障害発生起点となります。明確に定義された TCB がなければ、トラストアンカーとコアアプリケーション間の相互作用は大まかに定義され、敵対者が破壊できるギャップを有する可能性があります。TCB は、トラストアンカー、コアアプリケーション、およびネットワークピア間の通信が安全で、信頼でき、そして最新であることを保証します。TCB がなければ、エンドポイントのセキュリティライフサイクルを管理する中心的なコンポーネントはありません。

6.2 トラストアンカーの活用

エンドポイントがエコシステムに加わるには、それ自体のプラットフォームの整合性を検証できなければならず、そのピアの ID を認証できなければなりません。これを行うには、エンドポイントはトラステッドコンピューティングベースに組み込まれたトラストアンカーが必要になります。

トラストアンカーはハードウェアのセキュアエレメントで、個別の物理チップ、または CPU 内のセキュアコアのいずれかで、暗号秘密を安全に格納して処理できます。UICC または eUICC デバイスは、認証秘密を格納するための信頼要素として使用できるセキュア技術の例です。

信頼要素を効果的に使用すると、データの格納、検証、更新、および処理が効果的に作用します。データは、暗号で検証する必要のある秘密または公開情報のいずれかとなります。いずれの場合でも、トラストアンカーは、メッセージおよび ID が認証できるかどうかを確実に判断できなければならず、TCB に全ての認証結果または暗号操作の結果を安全に伝えることができなければなりません。これにより、アプリケーションおよび TCB は、エンドポイント全体のセキュリティに影響を与える価値ある決定を行うことができるようになります。例えば、トラストアンカーは、ネットワークピアがパッチを展開するサーバーなどの重要なリソースを偽装しているかどうかをエンドポイントが判断するのに役立ちます。トラストアンカーがネットワークピアを検証できない場合、エンドポイント上の TCB およびアプリケーションは、そのようなピアと対話しないことを選択し、可能であれば、不正なネットワークリソースをユーザーに警告する必要があります。

コンポーネントのコスト削減と需要の急激な増加により、トラストアンカーはこれまで以上により利用できるようになっています。これには実際のトラストアンカー技術だけでなく、技術と一緒の使用が承認されたライブラリおよびインターフェースも含まれます。これにより、エンジニアリングチームはごくわずかな時間でトラストアンカーのソリューションをスピンアップすることができ、カスタムソフトウェアや不十分に実装された標準によって技術の寿命が低下しないようになります。可能であれば、セキュリティのギャップの可能性を減らすために標準を使用すべきです。

軽量なエンドポイントにトラストアンカーを実装する際のもう 1 つの課題は、コンポーネントのサイズです。外部のトラストアンカーを活用する場合、最小限のコンポーネントプロファイルを維持する必要があります。フォームファクターが UICC のような技術を組み込む場合、このプロファイルを実現するのは難しいです。但し、ETSI TS 102 671 規格は、約 6 ミリメートル×5 ミリメートルサイズの非常に小型なフォームファクターを導入することによってこの問題を解決しています。この「MFF1」および「MFF2」は、UICC スマートカードフォームファクターを増強し、物理的要件を最小限に抑えながら、UICC がサポートする技術への完全アクセスを可能しています。フィールドにプロビジョニングされ、デバイスに固く結合されたフォームファクターを利用することにより必要以上のセキュリティが加わるので、敵対者はデバイスの ID を別のデバイスに転送することがより困難になります。

トラストアンカーの開発および展開する際に発生する費用には、以下のようなものが挙げられます。

- CPU または個別チップに埋め込まれた基盤技術のコスト
- 技術を回路に統合するコスト（必要な場合）
- ドライバーを OS および TCB に設計または統合するコスト

- トラストアンカーを使用するアプリケーションを設計するコスト
- トラストアンカーの維持（必要な場合）
    - セキュリティキーの維持、キーの取り消し、ID の停止
    - キーおよびメタデータの保護と管理に必要なインフラストラクチャの維持
- サービス側でのトラストアンカーID の監視
    - デバイスのブラックリスト化の実装（必要な場合）
- UICC などのトラストアンカーを監視・管理するためのキャリアサービスの統合（利用可能な場合）

6.2.1 リスク

トラストアンカーを利用しない場合のリスクは数多くありますが、すべて同じ基本問題、つまり敵対者が IoT エコシステム全体に関連するキーを盗む能力に起因します。この結果、敵対者は以下のことを行うことができます。

- エンドポイント ID の複製
- IoT サービスの偽装
- 承認されていないパッチや更新の展開
- エンドポイントソフトウェアに不正な変更を行うこと

セキュリティにおけるこのようなギャップは、時間の経過と共にビジネスに対してコストのかかる問題を引き起こしますが、敵対者だけでなく、競合他社がインフラストラクチャを悪用して利益を得ることができるようになります。

6.3 耐タンパートラストアンカーの使用

一部のトラストアンカーは、FIB、サイドチャネル解析、グリッチなどの特定クラスの攻撃から保護するために、必要以上の物理的なセキュリティを備えています。FIB 利用などの一部の攻撃は、コストの観点からガードすることはほとんど不可能ですが、トラストアンカーの製造では、現代の技術を使用して攻撃のコストを高めることができます。攻撃にかかるコストが高ければ高いほど、ランダムなエンドポイントデバイスに対して攻撃されるか可能性は低くなります。代わりに、攻撃は費用が報酬に見合ったターゲットに焦点を当てます。

近い将来、一部のトラストアンカーのメーカーは、FIPS（Federal Information Processing Standards、連邦情報処理標準）[10]、EMVCo [11]およびコモンクライテリア（Common Criteria）が認可された技術のバリエーションを公開する予定です。新技術を開発しているエンジニアは、近い将来、現在の設計が準拠モジュールへの移行に対応しているかどうかを判断する必要があります。

詳細については、各標準の最新版を確認して、メーカーが提供する機能レベルを評価してください。実装のコストおよび複雑性により、コンシューマーベースのデバイスでは、一部のセキュリティレベルが意図的に不可能に近いことに注意してください。

6.3.1 リスク

耐タンパートラストアンカーを使用しないリスクは極めて高いです。例えば、トラストアンカーが NVRAM に埋め込まれた単なる暗号化キーの場合、そのキーを抽出するツールとスキルを備えた攻撃者はインフラストラクチャ全体を破壊できます。しかし、秘密が耐タンパートラストアンカーに格納されている場合、秘密を抽出する費用は高くなり、秘密が抽出される可能性はかなり低くなり、トラストアンカーは潜在的な攻撃対象としての価値が下がります。

トラストアンカーの実装が弱い場合、侵害をもたらす秘密の抽出は十分に高くなることは顕著です。侵害は、設計、アーキテクチャ、生産、および実行中に発生する費用は無効になります。これにより、著しい財政損失を招く場合があります。従って、組織が正しい実装を確実に設計することが不可欠です。

6.4 TCB 用 API の活用

信頼のルートが TCB 内で確立されたら、TCB の機能および信頼のルートを効果的に組み込んだプロファイルを使用する必要があります。API は以下を確保する必要があります。

- すべての署名検証は TCB によって実行されること
- 秘密キーは TCB から公開されていないこと
- キー交換は、アプリケーションに代わって TCB によって実行できること
- 暗号解読は TBC で実行できること
- 暗号化は TCB 上で実行できること
- メッセージの署名は TCB 上で実行できること
- セキュアなメッセージの埋め込みは TCB 上で実行できること
- TCB およびアプリケーション間の機密性および整合性

この一連の機能は、TCB が重要なセキュリティ資産を安全でないアプリケーションやハードウェア環境に決して公開しないことを保証するのに役立ちます。これは、これらの要件を統一した方法で適用する既存の仕様を使用することで実現できます。評価を検討してください。

- SIM Alliance Open Mobile API [12]
- GlobalPlatform Secure Element Access Control [13]
- GlobalPlatform Trusted Execution Environment (TEE) API Specification [14]
- Trusted Computing Group (TCG)
- oneM2M TS-0003 [20]

多くのトラストアンカーは、TCB として実装できるソフトウェアライブラリを備えています。これらのライブラリには、エンジニアが TCB と対話するために使用できる API があります。トラストアンカーが提供するライブラリは、トラストアンカー開発の分野における専門家によって入念に検査された可能性が高いので、利用可能な場合は優先されます。しかし、エンジニアリングチームはこの推奨事項に記載されている要件一覧を評価し、ライブラリがこれらの懸念事項を適切に説明することを確実にする必要があります。

さらに、TCB はエンドポイント上で実行されている特権アプリケーションから*のみ*アクセスできる必要があります。TCB インターフェースは、エンドポイント上で実行されている特権のない、または信頼されていない（サードパーティの）アプリケーションから決してアクセスできないようにする必要があります。すべてのアクセスは、要求を評価し、信頼できないアプリケーションによって疑わしい、またはプライバシー中心の要求が行われた際、オプションでユーザーに警告する、信頼できるサービスを介してプロキシされる必要があります。

このプロトコルを実装する際の課題は、データの起点と TCB 間ですべてのメッセージが改ざんできないこと、およびその逆を保証することです。最も効果的なのは、アプリケーションから呼び出すことができる EEPROM のセグメントがアプリケーションに代わってこれらの機能を実行できる場合です。API コードの矛先を内部 EEPROM に分離し、内部 RAM を使用してメッセージを処理することにより、重要性の低いデータが外部バスに公開されることになります。

6.4.1 リスク

アプリケーションプロトコルインターフェースが明確に定義されない場合、TCB を使用すると意図しない結果や副作用が生じる可能性があります。事前にプロトコルを定義し、ロジックとセキュリティの問題を入念に検査することで、エンジニアリングチームは後にセキュリティ問題を引き起こす可能性のある欠陥を迅速かつ効果的に特定できます。従って、プロトコルの定義は、IoT サービス提供者のニーズを盛り込んだ既存の API の評価を具現化する必要があります。既存の定着した技術が特定できれば、これはカスタムソリューションよりも常に有利になります。

6.5 組織の信頼の基点（Root of Trust）の定義

組織的な信頼のルートとは、ID、アプリケーション、通信のセキュリティを暗号によっていかにして確保できるか（確保すべきか）を定める、一連の暗号化ポリシーおよび手順です。独自の対象キー、証明書、公開キーのいずれかの形式で強力な暗号を使用する必要があります。これは、TCB で使用できるモデル、トラストアンカーの機能、およびエンジニアリングチームにとって意味あるものに依存します。

階層内で使用する他のキーにデジタル署名するには、対称または非対称いずれかのルート秘密キーを使用する必要があります。例えば、例示の組織である、Example IoT Company LLC が組織的な信頼のルートを作成したい場合、信頼できるマシンにルートキーを生成します。このキーは組織のルートを表します。次に、独立したセキュリティ階層を持つ必要のある各サブ組織を表す新しいキーを生成します。例としては以下のようになることがあります。

- コード署名キー
- サーバー通信キー
- ピアツーピア通信キー
- エンドポイント ID キー
- マスター失効キー

これらのキーのそれぞれは、組織のルートキーによって署名される必要があります。これらのキー、対応する署名、およびルートキーはすべて TCB が使用するトラストアンカーに格納する必要があります。次に、特定のキーにリンクされたアプリケーションが使用されるたびに、アプリケーションは特定のキーを使用して、通信チャネルを介して送信されたメッセージを検証できます。

このモデルは、すべてのメッセージが暗号階層を通じて安全性が確保されたことを確認するのに役立ちます。特定のキータイプ間で役割を分離することにより、同じ通信プロセスを通じて侵害されたキーを取り消すことができます。

このメソッドの展開を支援する既存のプロトコルには以下の通りです。

- トランスポート層セキュリティ（TLS）。有効な最新の仕様
- セキュアシェル（SSH2）
- オンライン証明書ステータスプロトコル（OCSP）IETF RFC 2560
- 汎用ブートストラッピング・アーキテクチャ（GBA）（付属書 A 参照）3GPP TS 33.220

暗号キーが必要なサービスを展開する必要がある場合、問題が生じます。サーバー通信キーなど、セキュリティが重要な資産をインターネットにアクセスできる Web サーバー上に配置する代わりに、個別の証明書またはキーのペアをサーバー層専用に生成する必要があります。その後、この証明書はサーバー通信キーで署名される場合があります。このように、エンドポイントはサービスが信頼のルートで認証されたことを検証できますが、重要な組織キーは敵対者に公開されません。

キーが侵害された場合、失効マスターキーを使用して失効を認証することにより、使用から失効させることができます。

組織的な信頼のルートの中核となるキーはすべて、インフラストラクチャのセキュリティにとって重要であることは言うまでもありません。これらのキーは厳重に守られ、コアチームの信頼できる内部メンバーによってのみ使用されなければなりません。承認されたハードウェアセキュリティモジュール（HSM）を利用して、キーの格納、アクセス、および利用することを強くお勧めします。

技術の展開開始時は多くの場合、HSM は多大な費用がかかりますが、長期的な財務効果は非常に肯定的です。TCB と HSM で解決できるであろう特定のリスクを診断して立ち向かうために、法医学分析とエンジニアリングの後に多大な費用を負担するのではなく、比較的少額な初期費用がかかります。

6.5.1 リスク

組織的な信頼のルートを使用しないリスクは、単一のキーに対する侵害がエコシステム全体の侵害をもたらします。組織を階層に分け、階層に個別のキーを配置することで、一定間隔で、キーが関連するアプリケーションまたはサブ組織の優先度に応じてキーを循環させることができます。これにより、組織のファセット間での職務分離が生成され、侵害されたキーがインフラストラクチャ全体のセキュリティを破壊する機能が低下します。

6.6 フルフィルメントの前に各エンドポイントデバイスをカスタマイズ

エンドポイントデバイスは、敵対者、競合他社、および愛好家たちが実稼働環境で他のユーザーやデバイスを侵害できないように、暗号化された一意の ID で有効にする必要があります。これを適切に実現するには、パーソナル化のプロセスを製造時に実行する必要があります。これは、特定の TCB ソリューションのメーカー経由、またはプリント基板アセンブリ（PCB/A）プロセス中のいずれかで行うことができます。

パーソナル化のプロセスを解決するには、以下の手順を実行します。

- 一意の暗号化キーを生成する

- 組織のエンドポイント署名キー（またはその派生物）を使用してキーに署名する
- TCB のトラストアンカーにキーを格納する
- その特定のエンドポイントに一意の内部 ID を生成（または使用）する
- TCB トラストアンカーに一意識別子を格納する
- 一意識別子、キー、および署名を IoT サービスのバックエンド認証システムに保存する

エンドポイントプラットフォームのパーソナル化は、ネットワーク ID のパーソナル化とは別のものであることに注意してください。ネットワーク認証に UICC を利用することは多くの理由から有益であり、可能であれば、UICC をトラストアンカーとして使用することができます。ただし、ネットワークのトラストアンカーがネットワークの認証にのみ使用できる場合、アプリケーションのトラストアンカーのパーソナル化は、個別に実行する必要があります。エンドポイントアプリケーションを実行する前にアプリケーションのプラットフォームが確実に検証されるようにするには、アプリケーションのトラストアンカーの暗号を一意性なものにする必要があります。

UICC は、ネットワーク事業者またはその他発行元当事者との適切な契約を活かして、アプリケーション中心のトラストアンカーとして配信する前にプロビジョニングされることがあります。近い将来、エンドポイントの開発者は、eUICC 技術が IoT 製品及びサービスでの使用に適しているかどうか評価する必要があります。これらの技術は、アプリケーション中心のトラストアンカーに類似した方法で暗号秘密の現地プロビジョニングを可能にします。モバイル業界はパーソナル化およびプロビジョニングプロセスにおけるリーダーであるため、eUICC をトラストアンカーとして利用することは大きな利点となることがあります。

さらに、これらの技術は、アプリケーションおよび eUICC トラストアンカーが安全に通信するために、リモートプロビジョニング機能およびセキュアチャネルが組み込まれています。これらの機能は現地でのパーソナル化を提供し、個々のエンドポイントごとのパーソナル化およびプロビジョニングの全体的なコストを削減します。

IoT サービスエコシステムにおける UICC カードの使用に関する簡易チュートリアルは付属書 B に記載しています。

課題となるのは、エンドポイント ID および署名プロセスを管理することです。各 ID は、改ざんできないシステム内で、ID に一致する一意識別子と併せてカタログ化される必要があります。プロセスは通常、PCB/A 施設で実行されますが、その施設からビジネスへの接続を設定して、ID データを安全にトラフィックする必要があります。

このソリューションを展開するには、暗号のパーソナル化に詳しい施設を使用すると簡単になることがあります。その他の製造設備には、これを実現するためのプロセスがないかもしれません。モバイル業界は、UICC

などの埋め込み技術の製造および実行を制御する機能があるため、このような形で成功することができました。かねてより、モバイル業界はこのプロセスのリーダーでしたが、IoT アプリケーションエンドポイントのパーソナル化およびプロビジョニングプロセスは依然として初期段階にあります。

エンドポイント ID がゲートウェイまたはアップリンクで管理されるべき（または管理できる）かどうか判断する準備をしてください。IoT 製品またはサービスのアーキテクチャを評価することは、この ID 管理の属性がパーソナル化プロセスに影響を与えるかどうかを判断するのに役立ちます。信頼はゲートウェイに配布されるかもしれませんが、組織は、通信および認証システムのセキュリティ全体を低下させることなく、信頼が適切に委任できるかどうかを判断する必要があります。

パーソナル化に関連する経費には通常、以下のものが含まれますが、これらに限定されません。

- チップメーカーでのパーソナル化プロセスの実装
- メーカーおよび IoT サービス提供者の双方でパーソナル化した固有の値の調整または配信
- パーソナル設定を行った ID の実装および管理

6.6.1 リスク

組織がエンドポイントデバイスのパーソナル化を実装しないことを選択した場合、エンドポイントを別のエンドポイントと区別できないリスクがあります。すべてのキーがエンドポイントシステムにわたって準拠している場合、シリアル番号が一意であるかどうかは問題になりません。この理由は、キーが単一のエンドポイントから抽出された場合、敵対者は任意のエンドポイントを偽装することができるからです。

パーソナル化は、敵対者が複製または偽装したい各エンドポイントから暗号秘密を敵対者に抽出させることでこれに立ち向かいます。このプロセスの経費は非常に高くなるので、トラストアンカーを利用したパーソナル化は、複製および偽装に立ち向かうためのたった一つの最強な方法です。

6.7 実行可能な最小プラットフォーム（アプリケーションロールバック）

実用最小限実行プラットフォーム（MVeP）は、トラストアンカーと通信するための信頼できる実行環境を作るために実行する必要のある最小限の作業量です。通常、これは以下のことを意味します。

- 内部クロックまたは発振器の設定
- コア周辺機器の構成（メモリ、ストレージ）

- 様々なハードウェアブリッジまはた周辺デバイスの有効化
- CPU が実行するコードの次のチャンクの認証
- 次の段階のコードの実行
- アプリケーションイメージロールバックの管理

この MVeP が定義されると、最小ブートローダはトラストアンカーを使用してより堅牢なブートローダを検証したり、外部アプリケーションを検証した後に残りのブートローダを実行したりすることができます。これにより、アプリケーションプラットフォームを定義する後続のコードチェーンを認証する最小限の労力で一貫した環境を定義することができます。

別の利点としては、MVeP モデルを使用することで、内部 NVRAM または EEPROM が最小量のプロセッサであっても、内部または外部のトラストアンカーを使用して信頼されたアーキテクチャをブートストラップすることができることです。

最後に、MVeP は、特定のプラットフォームの安定版にロールバックするために重要です。アプリケーションファームウェアイメージの整合性を検証し、実行環境を構成するために必要な最小限の機能を持つ MVeP を定義できる場合、その機能をコアアプリケーション機能から切り離すことができます。従って、ファームウェアの更新が何らかの理由で失敗した場合でも、MVeP を使用してバックエンドネットワークに再接続し、別のファームウェアイメージ（同じイメージまたは古いイメージ）をダウンロードできます。これにより、NVRAM チップが損傷したエンドポイントでも、バックエンドサービスと通信して、診断情報を送信できます。

6.7.1 リスク

それは安全のように思われるかもしれませんが、MVeP を定義することにより、エンドポイント全体のアーキテクチャがブートプロセスの各ステップを暗号で検証することが保証されます。このステップは、エンドポイントがネットワークおよびそのピアに対してエンドポイント自体を認証できることを保証する上で重要となります。
MVeP のアーキテクチャが適切でない場合、ブートプロセスのセキュリティギャップが敵対者に悪用され、セキュリティアーキテクチャが無効になる場合があります。

6.8 各エンドポイントを一意にプロビジョニング

パーソナル化は、各デバイスが製造されるとそのデバイスは一意であることを保証しますが、プロビジョニングでは、固有のデバイスがアクティブ化され、更新され、特定の顧客 ID に紐づけられます。プロビジョニングプロセスは、製造したデバイスを IoT 環境で購入および/または展開したデバイスから切り離すのに役立ちます。これは IoT サービス提供者に役立ちます。

- アクティブなデバイスと非アクティブなデバイスとの区別
- エンドポイントを特定の顧客にリンクされたネットワークまたはその他のリソースに関連付ける
- 顧客のニーズに応じてエンドポイントをカスタマイズ
- 特定の顧客またはエンドポイントが侵害されたかどうかをより簡単に判断

プロビジョニングプロセスは製造時には発生しませんが、製造中に展開したパーソナル化のプロセスに依存します。プロビジョニングは通常、アクティブ化プロセスを初期化する顧客に基づいて現場で行われます。ただし、プロセスを保護するために、プロビジョニングは、パーソナル化プロセス中に設定した一意のセキュリティトークンに依存し、一意のエンドポイントが一意の顧客に紐づけられるようにします。このように、敵対者はエンドポイントの詳細を推測するだけでは、エンドポイントデバイスを任意に登録（または登録解除）することはできません。代わりに、パーソナル化プロセス中に生成され、設定された各一意の暗号トークンを必要とし、それは計算することは不可能です。

このようにして、IoT サービス提供者は、敵対者が意のままにエンドポイントデバイスを任意になりすましたり、登録することは不可能であることを数学的に保証できます。これにより、よりセキュアで安定した IoT 環境が実現し、顧客とデバイスの関係はより信頼できるものになります。

6.8.1 リスク

プロビジョニングプロセスを実装しないと、組織およびそのエンドポイントのノード間の同期が解除される可能性があります。組織がエンドポイントを追跡することがより困難になり、そしてどのデバイスがエコシステムで使用できるようになっているか、または停止されているかを確証しました。さらに、どのデバイスが特定の顧客に関連付けられているかを証明することが困難になる場合があり、その結果、現場で問題のあるデバイスまたは潜在的に侵害されたデバイスを追跡するのがより困難になります。

6.9 エンドポイントパスワード管理

ユーザーインターフェースを組み込んだデバイスは、パスワードを効果的に管理できるようにする必要があります。これにはいくつかのことが必要です。

- ブルートフォース攻撃の軽減
- デフォルトのパスワードまたはハードコードされたパスワードの無効化
- パスワードのベストプラクティス実施
- ログインインターフェースでのユーザー資格情報の表示を禁止する

- 無効なパスワードの試行のしきい値と増分遅延の強制

別のユーザーがユーザーのパスワードを推測しようとする、考えられる最も簡単な攻撃からユーザーを保護する必要があります。これは、ブルートフォース攻撃の可能性を否定するだけで軽減することができます。これは、パスワードの試行間の時間制限を増やすことで行うことができます。失敗したログイン試行ごとに、次のパスワードの入力が許可されるまで遅延が増加するはずです。一度に施行できる回数が N 回を超えないように天井を実装する必要があります。それ以外の方法では、合理的なロックアウト期間を実行する必要があります。正しい資格情報が入力されると、ブルートフォース攻撃に対する警告をユーザーに行う必要があります。

IoT システムでは、ハードコードされたパスワードまたはデフォルトのパスワードを*絶対に*使用すべきではありません。システムに入るための管理用「バックドアパスワード」は*絶対に*あってはなりません。デフォルトの資格情報を持った特権アカウントは*絶対に*あってはなりません。これは、セキュリティが弱いインターネット上をランダムに行き来するユーザーによる不正な侵入からユーザーのデバイスを保護するために不可欠です。

パスワードは、現行の情報セキュリティにおけるベストプラクティスを代表する最低品質条件を満たす必要があります。これは、パスワードを力づくで破るのが困難になることを保証し、盗難からユーザーを保護するのに役立ちます。アプリケーションが最近のベストプラクティスに準拠していることを保証するために、パスワードセキュリティに関する OWASP または SANS のガイドラインの見直しを検討してください。

パスワードをユーザーの画面に絶対表示してはいけません。パスワードはアスタリスク文字または別の安全なグリフで常に非表示にしてください。

さらに、パスワードを受け入れるすべてのインターフェイスには、ブルートフォース軽減技術を利用する必要があります。パスワードを*検証する*技術は強制を行う必要があることも重要です。例えば、Web ブラウザ上でレンダリングされた Web ページに埋め込まれた JavaScript は、ブルートフォース軽減を実装*すべきではありません*。Web に精通した攻撃者は、インターネット経由でバックエンド認証サーバーとやり取りすることでこれらの制御を無視できます。このモデルでは、軽減技術をサーバー側で実装する必要があります。モバイルアプリケーションでは、アプリケーションのセキュアなストレージ領域にローカルピンまたはパスワードが埋め込まれているため、モバイルデバイスはこのインターフェイスでブルートフォース攻撃を軽減する必要があります。

さらに、軽減システムは、各無効なパスワードが試行された後、許可した試行間に必要な遅延を増加させる必要があります。無効なパスワードの試行に対する最大しきい値も必要です。このしきい値に達した後、ユーザーは、2 要素認証またはより侵襲性のある別のモデルのいずれかで承認待ちとロックアウトされる必要があります。困難

このプロセスの実装は極めて簡単であり、エンジニアリングチームの一員には労力がほとんどかかりません。

6.9.1 リスク

この推奨事項を実装しないリスクは以下の通りです。

- 力づくでパスワードを推測することによって盗まれたデバイスが破壊されること
- 「ドライブバイ」によるインターネット攻撃は、ハードコードされたパスワードを使用するだけで IoT システムのセキュリティを破壊できること
- ユーザーインターフェイスがシステムに入力する実際のパスワードを表示している場合、ユーザーは「ショルダーサーフィン」によって侵害される可能性があること

6.10 実績のある乱数ジェネレーターの使用

お使いの TCB が本当に乱数生成できるかどうかを見極めてください。これは重要です。それがなければ、暗号検証プロセスが損なわれる可能性があり、暗号化されたデータをより推測しやすく、データの整合性を弱めてしまいます。

これは、一意の暗号化キー生成にとっても極めて重要です。一連の環境条件が与えられると、敵対者はキー生成、署名、または暗号化メッセージの埋め込みの間、TCB に推測可能な数を生成させるために環境に影響を与えることができなくなるはずです。

このプロセスは、TCB が FIPS [10]、EMVCo [11]またはコモンクライテリアで承認された乱数生成ができるかどうかを特定するのと同じくらい簡単です。

6.10.1 リスク

強力な乱数生成ジェネレータなしで暗号を利用することは、多くの理由で危険です。その理由はあまりにも多くてここに記載することはできませんが、注意すべき主な弱点がいくつかあります。

- 暗号化キーの生成が侵害され、弱いキーまたは推測可能なキーが生成される可能性があること
- ワンタイムパスワード/パッドまたはキーが弱いか推測可能になる可能性があること
- メッセージ再生の可能性を否定するために使用されるメッセージ埋め込みが侵害される可能性があること

これらの問題は、IoT エコシステム全体の暗号セキュリティの整合性全体に重大な失敗をもたらす場合があります。このリスクはエンドポイントデバイスに影響するだけでなく、ネットワーク全体に影響します。

6.11 暗号署名アプリケーションイメージ

CPU コア EEPROM の外部に格納されているすべてのアプリケーションは、暗号で認証される必要があります。これを行うには、単に以下の手順に従ってください。

- アプリケーションイメージのバージョンを表すメタデータを特定する
- メタデータを含む、アプリケーションイメージの暗号化ハッシュを生成する
- アプリケーションメタデータが内部メタデータと一致することを検証する
- ハッシュ値がトラストアンカー内部の値と一致することを検証する
- アプリケーション署名キーで署名を暗号化して検証する
- アプリケーション署名キーが組織ルートで署名されていることを暗号で検証する

このプロセスは、まず最も不安定なアクティビティを実行し、最後に失敗する可能性が最も低いオペレーションを実行する順序です。このように、最も可能性が高いリスクを観察するために最小の作業量が実行されます。

このプロセスは、特に TCB がアプリケーションに代わって主な処理を実行できる場合は、非常に実装しやすいです。実際の課題はより捉えにくいです。*どのアプリケーション*がその操作を実行しているかです。

暗号で検証されていないアプリケーションは、操作を実行できません。それは、自身のコードが敵対者によって破壊されているかどうかを知る方法がないからです。埋め込みシステムがアプリケーションを検証しない場合、NVRAM のコードを変更することは、攻撃者が埋め込みシステムを操作する一般的な方法です。

代わりに、内部 EEPROM アプリケーションはまず、外部の永続ストレージのアプリケーションイメージ上でこの手順を実行する必要があります。その後、そのアプリケーションは、操作自体を実行するか、内部 EEPROM にエンコードされたアプリケーションを要求してその代わりにこれらのタイプのテストを実行します。

6.11.1 リスク

エンドポイントファームウェア（NVRAM）に格納されたアプリケーションイメージが暗号で署名されていない場合、システムは承認されたコードと敵対者が投入したコードを区別することができません。これにより、敵対者が物理的に侵害されたエンドポイントを操作する実行可能なコードを乱用できるようになるだけでなく、ライバル企業が自身のソフトウェアをエンドポイント上にインストールできるようになる可能性があります。

6.12 リモートエンドポイント管理

すべてのエンドポイントがリモート管理を必要とするわけではありませんが、リモート管理を行う人は、サードパーティが管理者の資格情報を悪用して、フィールド内の一部（またはすべて）のエンドポイントを確実に侵害できないような方法で設計する必要があります。適切なソリューションは、エンドポイントの機能に依存します。ただし、以下のガイドラインを使用する必要があります。

- SSH 秘密鍵、TLS 秘密鍵、パスワードなどの非公開の暗号コンポーネントを、エンドポイント上の安全ではないストレージに配置しないでください
- 可能であれば、各エンドポイントごとに管理トークン（暗号化キーまたはパスワード）を生成してください
- パスワードを使用する場合は、パスワードの複雑性と長さに関するベストプラクティスに準拠するパスワードの使用を実施してください
- 可能であれば、管理者に対して 2 要素認証を実施してください
- 管理者がリモートでエンドポイントにアクセスする場合、ユーザーに知らせるようにしてください
- 仮想プライベートネットワーク（VPN）への管理アクセスを制限することを検討してください
- 公的にアクセス可能なアプリケーションまたは API にリモート管理機能を埋め込むことはしないでください。区別できる個別の通信チャネルを使用してください
- 管理通信チャネルの機密性と整合性を強化してください
- 業界標準の通信プロトコルを使用して、通信プロトコルが適切なエントロピーを備えていることを確保することによって、管理コマンド再生の可能性を軽減してください

6.12.1 リスク

リモート管理に関するポリシーの定義、実装、実施に失敗すると、エンドポイントがリモートで侵害される場合があります。エンドポイントデバイスへのスーパーユーザーアクセス用の厳重なセキュリティモデルが存在しない場合、敵対者は技術をリバースエンジニアリングできたり、エンドポイントからセキュリティキーを抽出したりすることができ、これはエコシステム内にあるすべてのエンドポイントにアクセスすることを引き起こします。管理アクセスは多くの場合、誤った構成や技術的に弱いため、埋め込みシステムの敵対者が最初に悪用する技術の 1 つとなっています。

6.13 ログおよび診断

エンドポイントデバイスの問題を評価するために、IoT サービス提供者は、エンドポイントの動作を絶えず評価して、エンドポイントが承認された一連の動作内で機能しているかどうかを判断する必要があります。これを実現するには、3 つの戦略を利用する必要があります

- 異常検知
- エンドポイントのロギング
- エンドポイント診断

エンドポイントは独自の動作を記録し、このログを処理するためにバックエンドサービスへ断続的にアップロードする必要があります。このログは、カーネルログ、アプリケーションログ、その他のメタデータなどの通常のアクティビティで構成する必要があります。

診断情報も定期的に監視し、通常のログと一緒または別にバックエンドサービスへ配信する必要があります。診断メッセージには、温度、バッテリの寿命、メモリ使用量、実行時間、プロセスリスト（該当する場合）など、エンドポイントに関する環境データをできるだけ多く含める必要があります。この情報は、問題のあるイベントまたは異常イベントにどのサービスがいつ関係するのかを特定するのに役立ちます。

ネットワーク内の異常検知は、ログまたは診断解析によって明らかにできない問題を捕らえるのに役立つはずです。また、ログや診断で観察できる問題を分類したり、その問題を物理的な世界で上手く反応しないかもしれない特定のコンポーネントだと考えることもできます。例えば、ネットワークに再接続し続ける移動体通信モジュール、または不良データを生成するセンサーなどです。

この情報全体では、技術の欠陥がフィールドで観察されるかどうか特定するのに役立つだけではありません。異常な動作がセキュリティイベントを示すかどうかを特定するのにも役立ちます。

6.13.1 リスク

ロギングおよび診断の実装に失敗すると、組織は重要情報を見落とすことがあります。この情報はエコシステムのセキュリティに影響を与えるだけでなく、製品設計の重大な欠陥を診断するのにも役立ちます。

6.14 メモリ保護の強化

埋め込みシステムは、メモリ管理ユニット（MMU）やメモリ保護ユニット（MPU）などの堅牢な技術を備えていないマイクロコントローラで設計されることがよくあります。ただし、これらの技術は以下のようなプラットフォームで使用する*必要があります*。

- 特権のないアプリケーションの実行
- 信頼できない（サードパーティの）アプリケーションの実行
- 特権のないプロセスでエミュレータまたは仮想マシンの実行

特権のないアプリケーションを実行する必要がある環境では、不正なアプリケーションや侵害されたアプリケーションから自身を保護できなければなりません。これにより、これらの不正なアプリケーションや侵害されたアプリケーションが TCB、トラストアンカードライバ、ハードウェア周辺のレジスタなどの特権リソースを制御するメモリ領域にアクセスできなくなります。

この分野における課題は、多くの場合、8 ビットマイクロコントローラのプラットフォームから、32 ビットマイクロコントローラやフルプロセッサアーキテクチャなど、より堅牢なプラットフォームに移行することです。しかし、MPU または MMU のいずれかでメモリ保護を正しく実装する埋め込みシステムには、無料またはほんのわずかなライセンス料で利用できるオペレーティングシステムが数多く存在します。

6.14.1 リスク

これらの技術を使用しない場合、不正なアプリケーションや侵害されたアプリケーションは、ドライバ、周辺レジスタ、カーネルやその他アプリケーションのような特権サービスなどのコアリソースの変更が制限されません。メモリ保護が欠如することにより、どのようなアプリケーションでもマイクロコントローラやプロセッサに存在するメモリの全範囲にフルアクセスできるようになります。特権のないアプリケーションは、これらのリソースの悪用を制限する*必要があります*。

6.15 内部 EEPROM の外部へのブートローディング

ブートローダコードの大部分は、CPU 内部にある EEPROM（Electrically Erasable Read-Only Memory）内に埋め込まれています。しかし、これは必ずしもそうではありません。お使いの CPU がブートローダを外部ソースから読み込むかどうかを見極めてください。ブートローダコードを検証させる EEPROM が CPU にない場合、ローカルの攻撃者に操作され、攻撃者に有利な方法で CPU を構成される場合があります。

ブートローダをホストするチップまたはメモリ領域に提供される保護レベルに応じて、敵対者はローカルバス（シリアル・ペリフェラル・インターフェイス（SPI）など）やリモート API（ファームウェア OTA など）を使用して、埋め込みコードを操作できる場合があります。これにより、敵対者は実行可能コードの第一段階である、最も信頼できる実行ポイントにカスタムコードを配置することで、コンピューティングプラットフォームを破壊することができるようになります。別の攻撃は、新しいチップを取り外してから結合するカスタム命令を含む独自のチップ用のブートローダチップを交換するだけの敵対者かもしれません。外部コードの整合性を検証する方法がなければ、ユーザーはソフトウェアが承認されたのか未承認なのかを区別できなくなります。

ブートローダをカスタマイズするために、攻撃者はブートローダを開発するか、その開発をアウトソースする必要があります。使用可能なリソースおよび対象のプロセッサに応じて、このアクションの難しさは、極めて簡単なものから極端に難しいものまで広範囲に及ぶ可能性があります。

ブートローダを格納するための内部 EEPROM またはロック可能な NVRAM を備えた CPU または MCU/MPU を使用することを検討してください。これにより、プラットフォームが少なくともアーキテクチャによって読み込み実行された最初の実行可能ファイルを検証できるようになり、より信頼性の高いデバイスが得られるようになります。

6.15.1 リスク

信頼チェーンを評価せず、CPU で読み込んだ初期コードの整合性検証を強化することは、システム全体の侵害につながる可能性があります。このステップは、IoT エンドポイントデバイス、ひいてはエコシステムを保護する上で重要です。

6.16 メモリのクリティカルセクションをロック

第 1 段階のブートローダやトラステッドコンピューティングベースなど、メモリの実行可能領域に格納されている重要なアプリケーションは、読み取り専用で格納する必要があります。これにより、敵対者からの投入なく、有効な構成にデバイスをブートできるようになります。この保証がなければ、実行の第 1 段階後に読み込まれた実行可能コードは、有効な構成または有効な状態にブートされたことを信頼することができません。

敵対者がこれらメモリのクリティカルセクションを独自のコードに置き換えることでシステムを破壊する可能性が依然としてありますが、ソフトウェアの独自のカスタムバージョンを構築する必要があります。これは複雑で困難なプロセスになる場合があります。これにより、攻撃に関する全体的なコストと、成功するために必要なスキルが大幅に増加します。さらに、パーソナル化およびプロビジョニングを使用する場合、これらのステップは、攻撃者に各エンドポイントのプロセスの再作成させ、彼らのソリューションをローカルシステム固有の暗号

特性にカスタマイズすることをを余儀なくさせます。これにより、攻撃全体が極端に高価になり、実行可能性が低下します。

このリスクを修復するには、メモリのクリティカルセクションを格納する技術がロックできるかどうかを確認するだけです。あるいは、ロック可能な EEPROM 技術から始めてください。

ロックが使用されている場合は、ロックがソフトウェアで設定されていないことを確認してください。ソフトウェアで定義されたロックは、ソフトウェアがそれぞれの機能を実行してロックを掛けた後にもに有効になります。敵対者が自身の利得に対してロックされていない状態を悪用することができる数ミリ秒のウィンドウがあります。従って、ヒューズやロックビットなどのハードウェアロックは、可能であれば常に使用する必要があります。

6.16.1 リスク

ロックまたは読み取り専用の状態でなければ、メモリのクリティカルセクションは敵対者によって簡単に変更される可能性があります。これにより、システム内で使用される後続のセキュリティ制御のすべてを妨害するといったさらなるアクションを実行することなく、エンドポイントプラットフォーム全体を侵害するのに十分な特権を与える可能性があります。

6.17 不安定なブートローダ

ブートローダのジョブはプライマリアプリケーションの実行するための CPU を構成するだけでなく、実行制御を読み込んでアプリケーションに転送することです。これを実現するために、ブートローダは通常、メインアプリケーションを見つけてメインの CPU メモリに読み込みます。この問題は、デフォルトのブートローダが特定のタイプのシステムで使用されている場合に発生します。

例えば、マイクロコントローラのベンダーが使用するブートローダの多くは、実行するために外部ファームウェアを CPU メモリに読み込んだり、シリアルインターフェイスを介してファームウェアの更新を可能にしたりします。その他のブートローダは、アプリケーションイメージを含む場所をユーザーにプロンプトして、ユーザーが選択したアプリケーションを実行できるようにすることができます。

この機能は、デスクトップ、ラップトップ、サーバーなどの環境では期待されていますが、埋め込みシステムでは受け入れられません。これは、ブートローダが未検証で信頼されていないアプリケーションを読み込んで実行すると、実行したアプリケーションの信頼性やセキュリティが保証されｒず、埋め込みデバイスの状態が問題となるからです。

従って、この問題を修復するには：

- ブートローダは、実行するアプリケーションイメージを暗号で検証できる必要があります
- デフォルト/標準のブートローダは、別のイメージやファームウェアのフラッシングが可能な場合は使用*すべきではありません*
- ブートローダは、任意の格納場所からアプリケーションイメージを読み込ませてはいけません
- 第 1 段階のブートローダの実行可能なイメージは、EEPROM にロックする必要があり、セキュアなプロセスを介してのみ更新する必要があります

さらに、ブートローダの設計は、サードパーティのセキュリティアナリストによる精査の対象となる必要があります。ソフトウェアのバグを操作してブートローダを侵害すると、カスタムコードが実行されたり、整合性検証のチェックが無視されたりする可能性があります。これは、ビジネスに有利ではないかもしれない*ジェイルブレイク*につながる可能性があります。システムに使用されているすべてのブートローダが、セキュリティ上のリスクを招く恐れのあるソフトウェアプログラミングの欠陥について徹底的に監査されていることを確認してください。

6.17.1 リスク

不安定なブートローダは、不完全に設計されたブートローダプロセスと同様に損害を与える可能性があります。ブートローダを保護することは、IoT エンドポイントの整合性を確保するための重要なステップです。

6.18 完ぺき前方秘匿性（Perfect Forward Secrecy）

PFS（Perfect Forward Secrecy）は、2 つのエンドポイント間の通信を設定する間に交換する暗号化キーの開示を取り扱います。概してエンドポイントには、それ自体の ID を認証するために使用される非対称証明書があります。認証フェーズが完了すると、対称キーが生成され、キーのネゴシエーションを保護する非対称暗号を使用して相互に合意します。このキーが生成され合意されると、2 つのエンティティ間に残っているセッションを保護するために使用されます。これは、非対称暗号に関わるコンピューターの経費を削減するために使用されます。対称暗号は計算コストが低いので、埋め込み技術や低電力技術では高速かつ電力消費が少なくなります。

しかし、難点があります。この共通キー合意モデルは、非対称キーが*常に秘密に保たれている*ことを前提としています。こうであるとは限りません。将来的には、十分に資金調達を得たエンティティは、与えられた公開非対称キーの秘密キーを計算できるようになるかもしれません。攻撃者がターゲットのエンティティとそのピア間の*通信セッションのすべてを保存する*と、将来そのエンティティは、いつか秘密キーを生成して*過去からの通信メッセージをすべて*解読できるようになります。

さらに、サーバーの暗号化キーは、匿名のサードパーティまたはビジネス内部関係者によって侵害される可能性があります。これが発生すると、盗まれた非対称キーで保護された通信メッセージを格納しているすべてのユーザーが、これらのメッセージを解読できるようになります。

この問題に対する解決策の 1 つは、キーのネゴシエーションプロセス中に一時的な非対称キーの組を生成することです。この一時的なキーの組の公開キーのみが、通信リンクの各側に渡され、対称キーをトラフィックに使用できます。

この一時的なキーは、十分なエントロピーと合理的な期間内に計算が枯渇する攻撃の可能性を否定するのに十分な大きさのキーサイズで生成する必要があります。これにより、キーのネゴシエーションプロセスが持続可能になり、将来的に攻撃の対象になる可能性が低くなります。

さらに、この方法では、ピアが永続的な非対称キーを認証のためだけに使用し、機密性および整合性のために使用することはありません。この非対称キーが盗まれたり公開されたりすると、通信チャネルの機密性および整合性ではなく、認証プロセスにのみ影響します。

このプロセスを攻撃からさらに回復させるには、認証に使用される非対称キーは、エンドポイントがキーが侵害されたかどうかを検証できることを保証する安全な失効プロセスの対象となる必要があります。エンドポイントは、そのような侵害が発生したことが通知された場合、認証のためにそのキーを信頼しなくなります。

6.18.1 リスク

PFS を実装しないと、敵対者が通信チャネルを保護するために使用する秘密キーにアクセスできるようになると、すべてのネットワーク通信が敵対者に公開される可能性があります。将来的にはいつでも、敵対者が秘密キーを捉えた場合、過去に敵対者によって捉えられたすべての通信は解読されるでしょう。これは深刻な結果につながります。

6.19 エンドポイント通信セキュリティ

本ガイドでは、いくつかの推奨事項やリスクについて取り上げていますが、エンドポイント通信セキュリティが IoT のエンドポイントにとって最大の脅威であることを簡潔に言及することが重要です。敵対者が通信チャネルを操作する能力は、エンドポイントが侵害される最も簡単な方法です。

これに伴い、エンドポイント設計者は、以下の観点から通信チャネルを実装する必要があります。

- ネットワークピアの認証
- データの機密性

- メッセージの整合性

他の組織が設計したエンドポイントと相互運用するためにクリアテキストメッセージを送受信することはできますが、コマンド、ユーザーのプライバシーデータ、または重要なシステムメッセージを組み込んだ*すべての*チャネルを介して通信されるデータを保護する*必要があります*。最初のステップでは、ピアデバイスを認証して、要求しているものであることを確認します。ピアがシステムサービスを表している場合、これは特に重要です。

次に、データの機密性が要求されます。これは、サードパーティが通信チャネルを介して渡される重要なデータを読み取ることができないようにするためです。

最後に、メッセージの整合性が要求されます。これは、秘密のメッセージが敵対者によって改ざんされていないことを保証するためです。

これら 3 つの属性を組み合わせることで、エンジニアリングのわずかな変更だけで長期にわたり存続できる通信モデルをもたらします。

このプロセスは、以下のような（これらに限定されません）既存のよく解析されたセキュリティプロトコルの使用によって、はるかにシンプルになります。

- 承認された最新の TLS 標準
- 承認された最新の DTLS 標準
- 認証およびキー交換向け SSH2
- キーの生成および交換向け GBA
- 承認向け OAuth2
- BEST, Battery Efficient Security for very low Throughput Machine Type Communication (MTC) devices [21]

エンジニアリングチームは、前述の要件に準拠したスイートを使用できますが、標準の通信プロトコルスイートを利用することで、フィールドで観察されるエラー数が削減されます。これは、情報セキュリティおよび暗号の専門家がプロトコルの標準化開発に関与しているからです。

LPWA の標準化されたネットワーク技術 NB-IoT および LTE-M を含む、3GPP ベースの移動体通信技術のセキュリティ特性については、GSMA PRD CLP.14 [4]に記載されています。

6.19.1 リスク

通信セキュリティが要件であるということは言うまでもありませんが、それが*なぜ*要件なのかについては分かりにくいことがあります。通信セキュリティは、敵対者がデータを読み取ることができないようにすることだけではありません。以下の点も確保します。

- エンドポイントが偽装されないこと
- 重要なサービスが偽装されないこと
- 不正なメッセージが検出できること
- ソフトウェアまたはセキュリティ構成の変更が安全に実行できること

通信セキュリティがなければ、IoT 製品またはサービスの品質、信頼性、またはプライバシーに関する保証はありません。

6.20 エンドポイント ID の認証

各エンドポイントが一意のシリアル番号などの暗号化された一意の ID を持っている場合、デバイスは*そのシリアル番号を正しく示す*ことを証明する必要があります。これを行うには、TCB は TCB および IoT バックエンドサービスのみが把握しているキーを用いてメッセージに暗号署名する必要があり、GBA などの技術で管理できる複雑さがあります。メッセージには、エンドポイントに対してそれぞれ一意の ID（シリアル番号またはその他のトークン）およびメタデータが含まれている必要があります。

TCB が署名するメッセージには、バックエンドシステムが発行するチャレンジも含まれている必要があります。これは、敵対者が TCB からバックエンドに送信済みの認証メッセージを*再生する*能力を無効にします。チャレンジに十分な*エントロピー*が含まれている場合、メッセージ再生の可能性は打ち消されます。

エンドポイントの ID にチャレンジするには：

- 一意の ID トークンを含むエンドポイントから要求を受信する
- 一意のチャレンジを生成してエンドポイントに送信する
- 署名およびメッセージを含むエンドポイントからチャレンジ応答を受信する
- 共有キーを使用して署名が正しいことを検証する
- 署名済みメッセージに正しい ID トークンとその他関連するメタデータが含まれていることを確認する
- 検証済み署名を承認する

チャレンジを処理するには：

- バックエンドシステムに接続する
- バックエンドシステムの暗号化 ID を受信する
- TCB を使用してバックエンドシステムの ID を暗号によって認証する
- エンドポイント ID およびその他メタデータを含むメッセージをバックエンドに送信する
- バックエンドからチャレンジを受信する
- 一意の ID トークン、メタデータ、およびチャレンジを含むメッセージを生成する
- メッセージに署名する
- メッセージおよびその署名をバックエンドに送信する
- バックエンドが署名済みメッセージを承認したことを検証する

6.20.1 リスク

この推奨事項を実装しないリスクは、エンドポイントがなりすまし攻撃に対してクローン可能または脆弱となることです。これにより、組織のインフラストラクチャは競合企業および敵対者の両方から攻撃を受け始める可能性があります。競合企業は、エンドポイント ID 認証がないことを利用して、同じ部品表から競合プラットフォームを構築することができますが、コストは低くなります。

あるいは、競合企業は認証がないことを利用して、組織のインフラストラクチャにピギーバック（便乗）したハードウェアを販売する可能性があります。これらの問題はビジネス収益の損失をもたらし、競合企業が利用に対する支払いを行っていないのに、ビジネスのネットワークインフラストラクチャを利用することで利益を得ることができるため、運営費が増加します。ネットワーク帯域幅、クラウドサーバー、CPU 使用率、ディスク使用量、その他リソースには定量化できるコストがあるため、このような寄生性のあるビジネスは、脆弱な組織に深刻な影響を与える可能性があります。

7 高優先度の推奨事項

優先度の高い推奨事項は、実装すべき一連の推奨事項を示していますが、エンドポイントアーキテクチャで必要とされる場合のみに限ります。例えば、すべてのエンドポイントアーキテクチャが耐タンパー製品ケーシングを必要とするわけではありません。ビジネスケースがこれらの推奨事項を要件だと見なすかどうかを判断するために評価する必要があります。

7.1 秘密用内部メモリの使用

可能であれば、プロセッサは内部 CPU を使用して、トラストアンカーに含まれていないコアの秘密および暗号化キーを処理すべきです。これにより、敵対者がメモリバスを監視している場合、またはメモリバスを操作できる場合、彼らはコアの秘密を取得しませんが、実行中のアプリケーション上でこれらの秘密を使用した場合の効果のみ見ることができます。

このモデルは暗号の秘密に関して寿命を延ばし、攻撃者が秘密を明らかにしないようにします。代わりに、攻撃者は前述の秘密を使用した場合の*効果*に相当する RAM のビットの操作に依存する必要があります。これにより、攻撃者は秘密が内部で使用されるたびにメモリのビットを変更し、攻撃の複雑さを大幅に高める必要があります。

秘密の処理については、すべてのオペレーティングシステムが内部 RAM の利用に関するモデルを定義するわけではありません。従って、エンジニアリングチームはこれを実装する必要があるかもしれません。このプロセスは難しくありませんが、些細なことでもありません。実行可能コードは、そのメモリルーチンが、内部プロセッサメモリを表すことを保証するすべて特定の領域を使用することを確保する必要があります。これは、使用するオペレーティングシステムおよびコンパイラツールチェーンに応じて、余分な労力が必要になることがあります。

7.1.1 リスク

ほとんどのマイクロプロセッサと一部の CPU は、内部 EEPROM または内部 NVRAM から実行するコード専用の内部 SRAM を少量備えています。この SRAM は通常、DMA などの技術を利用して意図的に公開されていない限り、外部周辺機器にアクセスできません。非公開にしておくと、コードによって処理される暗号の秘密は、RAM 通信を傍受できる敵対者に公開される可能性がはるかに低くなります。

リスクは高くありませんが、暗号の秘密は、攻撃の可能性を減らすために、公けにアクセス可能なバスを渡すべきではありません。RAM 通信を潜在的に高速で傍受できる知識を身に着けた敵対者は、暗号の秘密などのデータを取り込むことが *できます*。ただし、暗号操作に起因する可能性のある RAM を介してメッセージを取り込むには、熟練したリバースエンジニアリングが必要となるかもしれません。

そのため、これは重要な推奨事項となりますが、物理的なセキュリティの確保には重要ではないかもしれません。コアな暗号化キーがトラストアンカー内に格納され、セッションキーのみがアプリケーションによって処理される場合、外部 RAM のキーを処理してもすぐに侵害される可能性はありません。ただし、これは、暗号化アーキテクチャがキーローテーション、セッションキー生成、および証明書失効などのコアな IoT 操作にとって重要でないキーに対して公開キーを制限することを前提としています。

7.2 異常検知

エンドポイント動作をモデリングすることは、IoT セキュリティの不可欠な部分です。これは、侵害されたエンドポイントは、デバイスとの正常な対話だけがログに記録されて解析されている場合、正常に動作するエンドポイントと区別できないからです。IoT 環境のより包括的な観点については、デバイス動作の完全なフィンガープリントをカタログし、敵対者の動作を示す可能性のある異常を特定する必要があります。

エンドポイントから発生する異常な動作には以下のものがあります。

- 不規則な再起動またはデバイスのリセット
- 不規則な間隔で通信ネットワークを離脱または接続する
- 異常なサービスエンドポイントへの接続、または不適切なタイミングでのサービスエンドポイントへの接続
- 通常とは著しく異なるネットワークトラフィックのフィンガープリント
- エンドポイントからサーバーエンドポイントに送信された複数の不完全に形成されたメッセージ

エンドポイントタイプの通常の動作が、IoT サービス提供者によってカタログされる場合、組織は異常動作を示すであろう動作パターンを特定することができます。動作のベースラインを設定し、潜在的な外れ値を継続的に監視することで、組織は本番環境でセキュリティのみならずパフォーマンスに関する問題を迅速に診断できます。

動作のフィンガープリントをカタログすることで、障害のある一連の機能を特定の機能や環境条件により迅速にリンクすることで、組織を支援することもできます。これは、動作データが収集されない場合よりも、より迅速なペースでエンジニアリングソリューションズにつながる可能性があります。

7.2.1 リスク

異常検知がなければ、IoT エコシステム内で侵害したエンドポイントを検出する時間が非常にかかる可能性があります。エンドポイントの異常動作が通常の操作外でしか見えない場合、管理チームはエンドポイントに不信感を抱く理由がない可能性があります。ただし、異常検知がエコシステム全体にわたって実装されていると、悪意のある動作は検出される可能性があり、その結果、かなり早く抑制できるかもしれません。

7.3 耐タンパー製品ケーシングの使用

物理デバイスは、チップレベルでの耐タンパー性だけでなく、製品レベルでも耐タンパー性を備えている必要があります。製品に使用するケースは、敵対的なユーザーまたは好奇心が強いユーザーから保護する必要があります。これはいくつかの方法で実現できます。

- ケーシングを開いたときに NVRAM を無効にする回路
- ライトが検出されたときにセキュリティヒューズをとばすセンサー
- 物理的に静的なデバイスの場所が移動されたときにアラートをトリガーするセンサー
- コア回路コンポーネントを覆っているエポキシ
- デバイスから取り外した内部コンポーネントまたはリムーバブルコンポーネントのいずれかで発生したアラート

これらの方法を利用すると、物理エンドポイントの耐タンパー性が向上します。ただし、回路の設計自体を改善するほうがより費用対効果があるかもしれません。これらの方法論は、アマチュアの愛好家や敵対者による侵害の可能性を減らすのに大いに役立ちますが、知識を身に着けた経験豊富なセキュリティアナリストを和らげるものではありません。

従って、これらの方法は、コンシューマーが所有していない間に製品自体が改ざんできないように組織の能力を向上させます。言い換えると、コンシューマーが自分のデバイスを自宅や現場に残す場合、敵対者は物理的なアクセスを得てデバイスを侵害するだけでなく、デバイスを改造してから置き換えるために耐タンパー性のあるセキュリティ制御にも立ち向かう必要があります。これはデバイスがすぐに改ざんされて置き換えられることを無効にします。これは物理デバイスのセキュリティにとって価値ある改善点です。

ただし、脅威モデルがこの側面を無視し、高度な攻撃者や態勢が整っている攻撃者を含め、一般的な物理攻撃を修復することに重点を置いている場合、その脅威を完全に修復わけではありません。その場合、耐タンパー性のあるこれらの添加物は敵対者を鈍化させますが、時間と専門知識を持った敵対者を止めることはありません。

従って、費用対効果のあるものと、特定デバイスの脅威モデルとのバランスを取る必要があります。ATM（現金自動支払い機）がこのようなデバイスの例に相当します。敵対者が磁気ストライプのデータを取り込み、アクセス番号を記録するために、物理的な収納を開けたり改造したりすることができないよう、ATM セキュリティには収納における耐タンパー性が必要です。しかしながら、精通した敵対者は、既存の ATM の上

に適合させるためにローカルが似ているコンポーネントである、スキマー（skimmer）を考案しました。従って、物理的な改ざん防止は望む結果の一部だけを実現できます。アプリケーションおよびハードウェアの設計は、物理的な攻撃を減らすためにもう一歩踏み込む必要があります。

エンジニアやビジネスリーダーは、特定の製品やサービスの脅威モデルを評価し、攻撃のリスクとデバイスに実装された耐タンパー対策とのバランスを取る必要があります。耐タンパー性の各タイプは、プロセス、エンジニアリング、関連する材料に応じてコストがかかります。しかし、その努力は必要とされるセキュリティレベルとはならない可能性があります。

この問題の一例は、エポキシでチップをコーティングすることです。このプロセスは価値がありますが、攻撃者がエポキシの利用を無視するために簡単に行うことができる 2 つのことがあります。

- エポキシで覆われたコンポーネントから生まれるタップ回路
- エポキシの除去

エポキシはチップのコンポーネントを視界から隠していますが、エポキシで覆われたチップから出てくる回路を横切って電子が移動するのを妨げることはしないし、妨げることもできません。従って、重要な秘密がハードウェアバスを介して通信されると、エポキシは敵対者がこのデータを傍受する能力を止めることはありません。

さらに、エポキシ自体は簡単に除去できます。愛好者が手作りした技術は過去数年間で浮上し、コンシューマーがすぐに使える化学物質とプロセスを利用して回路からエポキシを除去する実用的な方法を明確にまとめています。プロセスには腐食性があり、潜在的に危険ですが、熟練したリバースエンジニアが概説した手順は妥当であり、適切に換気された実験室や事務所を持つ人は誰でも実装できます。

従って、耐タンパー技術のメリットと侵害の容易さを明確に重視するリスク評価を実行する必要があります。各デバイスがランダムデバイスを容易く操作したり悪用したりすることを望んでいる敵対者から単に保護するだけの場合は、耐タンパー性を用いる必要があります。高度な攻撃者によるハードウェアバスを介したメッセージの傍受を軽減する必要がある要件の場合、アプリケーションとオペレーティングシステムに対してより回復性のあるセキュリティアーキテクチャを耐タンパー性よりも考慮する必要があります。

7.3.1 リスク

前章で述べたように、耐タンパー性を展開しないリスクは、デバイスの要件によって大きく異なります。物理デバイスが開かれている、破損している、または改造されている場合、デバイスはユーザーに警告するという要件の場合、耐タンパー性が重要です。アマチュア、熟練したセキュリティ研究者または敵対者による解析からデ

バイスを保護する必要があるという要件の場合、アーキテクチャのセキュリティは恐らくリスクに対する正しいソリューションです。

いずれの場合でも、ケースに耐タンパー性を展開しないリスクは、敵対者が物理デバイスを改ざんしたかどうかをユーザーが判断できないようなものです。これは、堅牢で強化されたハードウェアとアプリケーションセキュリティのアーキテクチャを備えたアプリケーションにはさほど意味がないかもしれませんが、医療機器、テレマティックスシステム、ホームセキュリティまたはオートメーションシステムなどの重要なサービスをユーザーに提供する製品には*大きな*意味があります。

7.4 トラストアンカー間の機密性と整合性の強化

トラストアンカー間におけるすべての通信は認証され、機密性と整合性を強化する必要があります。このモデルに対する唯一の例外は、トラストアンカーがプロセッサコアの内部にある場合です。UICC などの外部トラストアンカーは、送受信されたメッセージが信頼できる場合のみ信頼できます。

ンジに対する回答を含むすべてのメッセージが機密に、可能であれば、検証可能な整合性で送信されることを検証します。

セキュアチャネルで管理できる UICC は機密性と整合性が可能です。IoT サービス提供者は、UICC セキュアチャネル技術を用いてアプリケーションセキュリティを支援できるかどうかについてネットワーク事業者と話し合う必要があります。将来的に eUICC は、アプリケーションのセキュリティが可能になるでしょう。その後、セキュアチャネルを使用すると、ブートローダステージからネットワーク認証ステージへのエンドポイントアプリケーションのセキュリティを容易にすることができます。

これは単純な仕事のように見えるはずですが、このプロセスには細かな区別があります。通信レイヤーの各側面をテストする必要があります。様々なトラストアンカーからのメッセージの中には機密でなく、整合性を用いて有効になっていない場合があります。例えば、操作が成功したか失敗したかを示すメッセージは安全のように思われますが、敵対者がアプリケーションを騙して目的に合わせた応答を送信しないように保護する必要があります。

一部のトラストアンカーでは、通信チャネルにおいて整合性を取ることができないかもしれません。整合性を優先し、メッセージが改ざんされていないことを保証するために使用する必要があります。しかし、これを行うには、ホストプロセッサのみならずトラストアンカー上にも信頼の基盤が必要です。これは、アプリケーションにとって合理的でない場合があります。

すべての埋め込みシステムは、十分に装備された物理的な敵対者から侵害できるので、ローカルバス通信のためだけに双方のプロセッサに信頼のルートを要求するのはやり過ぎかもしれません。ただし、物理的なセキュリティが重要なアプリケーションにおいては、整合性を実装する必要があります。

7.4.1 リスク

機密性と整合性を強化しないリスクは興味深いものです。このリスクは、システム全体の侵害から安全な情報収集まで多岐にわたります。これは、特定のメッセージを操作 *できる*からです。例えば、トラストアンカーがメッセージの整合性を検証するように TCB が要求すると、ハードウェアバスを介してメッセージをトラストアンカーに渡します。

トラストアンカーが CPU 内部にある場合、攻撃者は高機能で高価な装置なしでこのメッセージを改造できるという可能性は低いです。ただし、トラストアンカーが回路基板上の個別チップである場合、敵対者は回路を接続し、自身のハードウェアを挿入してメッセージを改ざんする機会があるかもしれません。例えば、トラストアンカーがメッセージを受信し、整合性なしに「はい、メッセージは有効です」とクエリに単に応答する場合、TCB は、攻撃者がバスへ物理的にアクセスしてメッセージを操作したかどうかを検証することはできません。

さらに、応答*が*整合性を検証しても、バスへ物理的にアクセスする攻撃者は、回路を容易く侵害し、TCB からのメッセージ要求を吸収し、トラストアンカーに自信が信頼するメッセージを送信し、TCB を経由して実際のトラストアンカーに応答させることができます。ハードウェア通信バスが適切に保護されていない場合、この攻撃も可能となり、トラストアンカーのジョブ実行機能を無効にします。

しかし、CPU およびトラストアンカーの双方が内部トラストアンカーを個別に持つことを要求すると、矛盾が生まれます。起動可能な CPU が敵対者によって変更できる場合、その CPU はどのように自身を信頼するのでしょうか。けれども CPU は、自身の EEPROM を使用してトラストアンカーの整合性を検証する必要があるのです。これは解決困難な状況を生み出しますが、解決できるものです。

1 つのソリューションは、公開キーを CPU の ROM に挿入することです。このキーはトラストアンカーによって送信されたメッセージの整合性を検証するために使用できます。（検証される）任意のメッセージがハードウェアバスを介してトラストアンカーに送信される場合、トラストアンカーは、返信の一部として*元のメッセージを含む*署名済みメッセージで応答することができます。これは、メッセージが実際にトラストアンカーから発進されたこと、そして処理されるメッセージが本当に処理を要求されたメッセージであることを検証します。唯一残っている懸念事項は、メッセージパディングで使用されるノンスが、暗号メッセージを*再生*できないようにすることを確保することでしょう。

上記を念頭に置くと、暗号だけでなく、暗号通信をサポートするアルゴリズムの非常に微妙な問題により、暗号化が失敗することを簡単に特定できます。このため、機密性と整合性を（正しく）実装することがかなり重要です。

7.5 アプリケーションの OTA アップデート

エンドポイント*アプリケーション*イメージをリモートで更新することは、簡単で分かりやすいプロセスです。複雑さは、現実的なセキュリティの欠陥に実際は対処しない方法で、ソリューションをオーバースペックすることに由来します。永続的なストレージの観点からは、エンジニアリングプロセスは非常に簡単です。

- アクティブなアプリケーションイメージの場所を定義する
- バックアップアプリケーションイメージの場所を定義する（ある場合）
- 緊急アプリケーションイメージの場所を定義する
- バックアップアプリケーションイメージのスペースが存在する場合、このスペースをアクティブイメージで更新する
- TCB に格納されている署名を使用してアクティブイメージを暗号で検証する
    - これにより、ストレージメディアが破損していないだけでなく、敵対者が書き込みプロセス中にビットを修正しなかったことが保証される
- 新しいイメージの全体またはデルタのいずれかで、そしてメタデータと署名をダウンロードする
- アクティブイメージをデルタでパッチする
- TCB を使用して暗号署名を検証する
- 新しいイメージで再起動する

いずれかの時点でプロセスが失敗する場合、システムはバックアップイメージに戻り必要に応じてアプリケーションを実行するようにするか、緊急システムを用いて*自宅に電話をかけて*障害が発生したことを IoT サービス提供者に通知する必要があります。

難しさは、2 つの問題に対処するストレージモデルを作成することに由来します。

- 更新プロセスを操作しようとする攻撃者
- ハードウェアの異常

バックアップシステムや緊急パーティションがなければ、デバイスは故障せざるを得なくなります。埋め込みシステムは通常、堅牢なユーザーインターフェイスを備えていないので、ビジネスとその顧客との間に著しいストレス

がかかることがあります。可能な限り説得できる原因で失敗することは、ユーザーの信頼だけでなく、システムの信頼性にとっても不可欠です。

攻撃者の中には、更新プロセスを意図的に破損し、システムを永続的に脆弱な状態にしたいと思っている場合があることに注目すべきです。例えば、アプリケーションのアクティブなバージョンで悪用できる脆弱性が発見されても、最新バージョンのアプリケーションではパッチを利用できます。

このモデルのメリットは、攻撃者がネットワークのネゴシエーションプロセスを破損しても、バックエンドシステムにはこのイベントに気付く機会があるということです。更新を*除き*、ノードが正常に通信していることをバックエンドネットワークが識別する場合、管理者がそのエンドポイントのノードが悪用されているかどうかを判断するためにアラートを発する必要があります。

7.5.1 リスク

OTA アプリケーションの更新プロセスが適切に設計されていない場合、敵対者がエンドポイントに実行可能コードをリモートで投入する可能性があります。敵対者がネットワーク上で特権的な立場にある場合、一度に数千のエンドポイントへ潜在的に影響を与える可能性があります。攻撃結果は、単純なコードの実行からサービスの拒否（エンドポイントを動作不能にすること）、またはエンドポイントデバイスの目的を完全に改造することにまで及ぶ可能性があります。

7.6 不適切な設計または未実装の相互認証

通信環境において、ピアは表面上のプロトコル *ID* を通じて互いに会話します。これは異なるコンテキストで異なることを意味しますが、すべての環境では、ある種の*アドレス*がメッセージの宛先を識別します。特定のプロトコルを実装している通信モジュールは、*特定のアドレス*の所有者であることを示すことができます。プロトコル個別の*実装*がローカルラジオモジュールのハードウェアアドレスを使用するように設計または強要されていても、ユーザーがそのモジュールの EEPROM を物理的に変更してハードウェアアドレスを変更できることを示すルールはありません。ユーザーがハードウェアアドレスを動的に変更できるようにすることを実装が拒否する場合であっても、それは依然としてアドレスを変更するように操作することができます。この機能の結果は、本質的になりすましです。そのコンピューター宛のメッセージを傍受する目的で別のコンピューターID を取得する行為です。

7.6.1 クライアント認証

すべての環境はなりすましに対して脆弱です。例えば、移動体通信ラジオは、事実であるかに関わらず、定められた国際移動体加入者識別番号（IMSI）の所有者であることを知らせることができます。どんなラッ

プトップでも、イーサネットアドレスを変更して、ローカルエリア・ネットワーク（LAN）上の他のコンピューターになりすますことができます。トポロジが物理空間または放送電波空間を通過するかどうかに関わらず、通信エンドポイント *ID* を偽装することができます。

これに対する保護は認証です。例えば、移動体通信ネットワークでは、適切な機器を持つ人は誰でも、自身が選択する IMSI を所有すると要求できます。しかし、携帯電話会社は、加入者（IMSI）ごとに一意の加入者識別モジュール（SIM）に暗号化キーをエンコードすることで*認証*を実施します。移動体通信デバイスが特定の IMSI を表していることを示す基地局と通信する場合、その基地局は、その個別 ID 用にプロビジョニングされた SIM カードに格納された一意の暗号化キーを持つ人によってのみ解決できる暗号化チャレンジを発行します。攻撃者が暗号化チャレンジを解決できない場合、基地局は攻撃者が問題となる IMSI を示さ*ない*ことを検証でき、そのユーザーをネットワークに関連付けることを無効にすることができます。

上記のモデルは、*クライアントベースの認証*を示しています。これは、クライアントが自身の ID を暗号で認証できる限り、サーバーサブシステム（基地局を含む）がクライアント（エンドポイント）にネットワークへ参加したり離れたりすることを許可するモデルです。ただし、クライアントを操作に公開するという逆の問題があります。*サーバー認証*

7.6.2 サーバー認証

3GPP モデルでは、エンドポイント（3GPP のユーザー機器と呼ばれる）のみが認証されます。エンドポイントは、接続先の基地局を認証しません。従って、どの基地局も携帯電話会社に代わってサービスを提供できます。移動体通信の基地局を操作したり構築したりすることができる個人は、選択する携帯電話会社を偽装することができます。カスタムの移動体通信基地局は現在、1,000 ドル（USD）以下で構築されていますが、合成電力はローカルエリア内にあるメッセージの傍受のみを可能にしています。偽のタワーが構築されると、基地局はローカルの携帯電話会社を偽装し、ローカルエリアのエンドポイントから電話、テキストメッセージ、さらにはデータも傍受することができます。

UMTS や LTE などの新しい 3GPP ネットワークプロトコルは、両方のエンティティの相互認証を強化しています。これにより、エンドポイントは、基地局がサービスを提供する携帯電話会社に代わってサービスを行っていることを暗号で検証できます。敵対者は、携帯電話会社の暗号を破って基地局を偽装する必要があり、攻撃の複雑さ、難易度、およびコストが大幅に増加させることになります。

7.6.3 移動体通信のインテロゲータまたは偽の基地局

しかし、このルールには、移動体通信のインテロゲータなどの例外があります。政府の請負業者、政府機関、および情報局によって通常使用されるこれらのデバイスは、国家安全のために、特定の携帯電話会社によってこれらのエンティティに提供される暗号化キーでエンコードされています。これらのシステムでは、このキーを使用して双方間通信を受動的に傍受するか、特定のターゲットに対して積極的に中間者攻撃を実行します。

しかし、現代の通信脅威モデルにおいて、この技術へのアクセスは、政府および情報分野のアクターに限定されていません。今日、これらのシステムは、わずか数百米ドルの部品で構築でき、移動体通信を傍受または偽装できる、費用対効果の高い偽の基地局をもたらすことになります。

7.6.4 通信セキュリティはゲート・ツー・ゲートセキュリティです

移動体通信のインテロゲータを持ち出すことは、通信セキュリティが絶対的ではないという考え方に触れることで、本章をかなり的確に要約するのに役立ちます。2 つのエンティティ間の通信チャネルを保護するのみです。しかし、これらのエンティティは、接続先のエコシステムにデータが出入りできるようにするゲートとして機能します。

例えば、特定の SIM カードは、油井監視装置などの産業用制御システムで使用するためにプロビジョニングされる場合があります。SIM カードは、設計上、取り外し可能なコンポーネントです。油井監視装置に物理的にアクセスできる人は誰でも、SIM カードを取り出してラップトップに置くことができます。ラップトップがその油井装置の機能をシミュレートできるソフトウェアを備えている場合、バックエンドサーバーは、実際の油井装置とラップトップを区別できなくなります。それにもかかわらず、ラップトップは SIM カードによって、移動体通信ネットワークに認証されます。従って、移動体通信ネットワークは SIM カードを認証しましたが、ラップトップは認証しません。

7.6.5 相互認証に向けた解決

IoT エコシステムの各ピアは、そのエコシステムに関係するその他すべてのピアを認証する必要があります。これを実現するには、TCB を使用して、適切な暗号アーキテクチャが通信技術を推進していることを確実にする必要があります。キーが敵対者に容易く公開される場合、相互認証は行われません。詳細については、本文書の TCB の章を参照してください。

一度認証されると、各ピアはネットワーク内にある他のピアに送信済みのメッセージを暗号化して署名する必要があります。メッセージを受信する各ピアは、データに従って処理する前にそのデータを暗号で検証する必

要があります。すべての通信プロトコルが相互認証を行うことができるわけではなく、強力な暗号を備えているわけでもないため、アプリケーションエンジニアは、通信プロトコルに頼るのではなく、機密性と整合性を強化する十分なプロトコルを設計することが不可欠です。

LTE などの相互認証を組み込んだより堅牢なプロトコルでさえ、移動体通信ネットワークを超えたインフラストラクチャのセキュリティに対処していません。より高いレイヤーのプロトコルセキュリティのみが、携帯電話会社の制御を超えてインフラストラクチャにおける弱点のリスクに対処できます。

7.6.6 リスク

強力なアプリケーションセキュリティを忠実に守らないリスクは、エンドポイントが通信レイヤーのセキュリティを信頼する必要があるということです。本推奨事項で示したように、アプリケーションのセキュリティ問題を解決するためにネットワークを信頼するだけでは不十分な場合があります。MNO を信頼できる場合でも、メッセージは、データが IoT サービス提供者が所有するサーバーに到達する前に、MNO が所有または制御していない複数のネットワークインフラストラクチャを通過する場合があります。従って、IoT サービス提供者は、エンドポイントシステム間でメッセージを傍受、改造、または製造するシステムを管理する人を危険にさらします。

7.7 プライバシー管理

IoT 技術に不可欠な側面は、物理世界をデジタル世界につなげる機能です。ユーザーの物理環境が彼らが好んでオンライン上で閲覧するものと直接関連付けられると同じように、この結果はプライバシーのギャップです。これは、時間の経過と共に望ましくない結果を引き起こす場合があります。

結果的に、IoT サービス提供者は、コンシューマーのプライバシーを考慮し、可能であればエンドポイントと製品またはサービスの Web インターフェイスの両方に統合されたプライバシー管理のインターフェイスを開発することが重要となります。

この技術により、ユーザーはシステムが利用するプライバシーの属性、利用規約、およびビジネスやそのパートナーにこの情報を公開するのを無効にする機能を判断できるようになるはずです。この細分性およびオプトアウトシステムでは、ユーザーが自分自身と物理的な世界について共有する情報を制御する権利と機能を備えるようになります。

7.7.1 リスク

コンシューマーのプライバシーを保護しないことに関する潜在的なリスクは数多くあります。ストーカー行為、嫌がらせ、プロファイリング、脅威などから発生する問題は、ユーザーのデータを保護しないという現実的かつ実際的な結果です。

7.8 プライバシーおよび一意のエンドポイント ID

各エンドポイントは、フィンガープリントによってデジタル的に認識されます。このフィンガープリントは、特定のエンドポイントに対して一意の*アドレス*、*シリアル番号*、および*暗号化 ID* で構成されています。ただし、これらのトークンは、デバイスを特定の顧客、場所、またはサービスに直接関連付けることもできます。多くの状況において、これは望ましくありません。例えば、スマートフォンは、802.11 のアクセスポイントを積極的にスキャンする際に内蔵の Wi-Fi アドレスが使用されたため、追跡できます。このアドレスは、場所から場所へ移動した際に追跡できます。これにより、誰もが特定の Wi-Fi アドレスを特定ユーザーに関連付けて世界中の動きを見ることができるようになります。これに立ち向かうために、スマートフォンソフトウェアのメーカーは、アクセスポイントをスキャンする際にランダムな Wi-Fi クライアントアドレスを生成しました。この方法で電話を追跡することはほとんど不可能になりました。

IoT エンドポイントは、Bluetooth 低エネルギー（BLE）アドレス、802.15.4 アドレス、Wi-Fi、さらには移動体通信 IMSI によっても同様に追跡できます。可能であれば、IoT サービス提供者は、ランダムなラジオアドレスを使用して新しい環境に接続するといった方法でエンドポイント技術を開発し、ユーザーのプライバシーを損なわないようにする必要があります。

これはまた、SSH 公開キーなどの暗号化キーにも当てはまります。ユーザーは通常、自身の公開キーを一般に公開する必要がある一方で、エンドポイントの暗号化公開キーは、特定のエンドポイントのユーザーID を公開しますが、これは望ましく*ありません*。代わりに、ユーザーは、新しい環境に接続している際に自身の ID を知らせたいかどうかを選択できるようにする必要があります。

7.8.1 リスク

このリスクを適切に軽減しないと、デバイスがネットワークに出入りする際に、モバイルエンドポイントを持っているユーザーが追跡されます。これにより、法務チーム、議員、さらには保険会社が現在分析しているプライバシーに大きなギャップをもたらします。追跡に関する可能性を軽減するべくプライバシーを適切に実装していないと、近い将来、IoT サービス提供者に法的な影響をもたらす場合があります。

7.9 適切な権限レベルでのアプリケーションの実行

エンドポイント上で実行されているアプリケーションは通常、スーパーユーザー権限を必要としません。ほとんどの場合、アプリケーションはデバイスドライバまたはネットワークポートにアクセスする必要があります。これらのデバイス、ポート、またはその他オブジェクトの中には、初回アクセスのためにスーパーユーザー権限を必要とする場合がありますが、その後の操作実行にスーパーユーザー権限は必要ありません。従って、アプリケーションの開始時点でのみスーパーユーザー権限を使用し、そのリソースにアクセスするのがベストプラクティスです。その後、スーパーユーザー権限を削除する必要があります。

スーパーユーザー権限を削除することは、よく文書化されている一般的なプロセスであり、セキュアシェル（SSH）、apache2 などの技術的に優れたサーバーなどのアプリケーションでは、非常にうまく実装されています。このプロセスには通常、以下が含まれています。

- 昇格された特権でアプリケーションを起動する
- 昇格された特権を必要とするすべてのリソースへのアクセス
- アプリケーションが実行するユーザーID（UNIX ユーザーID やグループ ID など）を識別する
- プロセス ID をターゲットユーザー/グループ ID へと完全に変更して、実行中のアプリケーションからスーパーユーザー権限を削除する

より複雑なモデルは、*特権分離*の SSH 実装で見ることができます。これは、ターゲットユーザー/グループ ID の下でメインアプリケーションをブートストラップすることが唯一の目的である特権サービスを実行します。このように、サービスが終了する場合、特権リソースの侵害なく、簡単に再起動できます。

詳細については、SSH 特権分離（http://www.citi.umich.edu/u/provos/ssh/privsep.html）を参照してください。

7.9.1 リスク

昇格された特権レベルでアプリケーションを実行すると、単一のアプリケーションが侵害された場合、システム全体が侵害される可能性があります。スーパーユーザー権限は、実行中のシステム全体へのフルアクセス権をアプリケーションに付与するため、敵対者がこのようなアプリケーションを侵害すると、彼らを阻止する方法はありません。権限を削除することは敵対者を阻止するのに役立ち、埋め込みシステム内で彼らが権限を増やすことを制限します。これは、システム全体の侵害と些細な苛立ちとの違いかもしれません。

7.10 アプリケーションアーキテクチャにおける職務分離の実施

エンドポイント上で実行中のアプリケーションは、異なるユーザーID を各一意のプロセスに関連付けさせる必要があります。これにより、1 つのアプリケーションが侵害された場合、同じエンドポイント上にある個別のアプリケーションは第 2 の攻撃を受けず、侵害されることはありません。攻撃者のために必要とされるこの追加ステップは、セキュリティー上の弱点を突く手段を開発するプロセス全体にとって重大な妨害となることが多く、エンドポイントに対する攻撃のコストおよび複雑さを増加させます。

例えば、ユーザーがエンドポイントの状態に関する情報を取得できるネットワークサービスは、同じプロセス上で TCB を操作することもできないようにする必要があります。この機能は、サービスの目的に関連する*範囲外*になります。この 2 つの異なる操作は別々のアプリケーションで処理し、ローカルのオペレーティングシステム上にある別々のユーザーID で実行し、アプリケーションの任務を分離し、1 つのコンポーネントが侵害された場合の悪用のリスクを軽減する必要があります。

これを正しく実装するには、基本的なハードウェアアーキテクチャでメモリ保護を有効にする必要があり、オペレーティングシステムには権限レベルの概念が必要です。特権のないソフトウェアは、ドライバ、構成ファイル、またはその他オブジェクトなどの特権のあるリソースへのアクセスを制限する必要があります。

サービスは、すべてのメッセージが適切に定義され、セキュリティアーキテクチャの要件に適合するように、システムコールなどの制約された API を介して、特権のあるリソースにアクセスする要求を行う必要があります。

権限の複数階層に関する概念は、半世紀前の概念です。しかしながら、埋め込みシステムでは、ユーザーがコンソールにログインして独自のアプリケーションを実行することが許可されていないため、見過ごされることが多いです。結果的に、すべてのサービスが特権ユーザーとして展開されることがよくあります。しかし、これには欠陥があります。

各アプリケーションまたはサービスは、カスタム特権を使用して実装する必要があります。ほとんどの環境では、これは単独の*ユーザーID* です。別のユーザーID を強要することで職務を分離することにより、1 つのサービスが侵害された場合、同じシステム上にある別のサービスが使用するリソースに直接影響を与えることはできません。その他のサービスやユーザーを侵害するには、ローカルのオペレーティングシステムで第 2 の突破口を見つけ、特権を昇格させる必要があります。

これには、特権分離を正しく利用する計画と安定したアプリケーションアーキテクチャが必要です。

7.10.1 リスク

職務分離が実行されていない場合、エンドポイント上にある単一サービスに対する侵害は、デバイス上で実行中の各サービスまたはアプリケーションが同一のユーザーおよびグループ ID の両方またはいずれか一方を共有するため、デバイス全体の侵害をもたらします。推奨事項が実装*されている*場合、ネットワークを介して侵害された権限の低いサービスは、システム全体の侵害を即座にもたらすことは*ありません*。

この推奨事項は実装しやすいため、IoT エンドポイントのセキュリティにとって重要です。注目すべきは、ネットワークサービスをリモートで侵害するには、多くの場合、膨大な量の専門知識が必要であるということです。敵対者がカーネルレベルの突破口や別の第 2 の突破口を実装して特権を昇格させ、システム全体を支配する必要がある場合、敵対者は攻撃を実行するための時間、スキル、または機器を持ち合わせていない場合があります。

このような単純な構成変更で攻撃の難易度を上げるのは、デバイスの寿命を確保するのに大きな役割を果たします。

さらに、侵害されたサービスは、プロセス監視やその他の解析を通じて検出されるため、サービスの侵害は、デバイスの侵害が検出されたことをサービスエコシステムに警告することができます。これにより、管理者はシステム全体の侵害が達成される前にシステムのセキュリティを確保することができます。これはまた、管理者が特定の脆弱性からの悪用が横行する前に、脆弱なソフトウェアの診断とパッチを行うこともできます。これにより、ビジネスは熟練した攻撃者に対してでさえ有利な立場になります。

7.11 言語セキュリティの強化

プログラム言語は、言語の目的とレベルの高さに応じて様々なセキュリティレベルを備えています。一部の言語では、未加工メモリへのアクセスを制限する構造を提供し、メモリの使用方法に関する制約を強化します。エンジニアリングチームは、アプリケーションの実行時または結果として生じるバイナリにセキュリティを提供できる言語を特定する必要があります。

可能であれば、脆弱性が敵対者に悪用される可能性を制限するために、コンパイラまたは実行時にセキュリティを強化する必要があります。明確に定義された実行時環境では、引き起こしやすいプログラミング上の欠陥であっても完全に悪用するのは極めて困難になる場合があります。これは、アプリケーションの実行方法、メモリへのアクセス方法を保護するためにセキュリティ強化が用いられ、オペレーティングシステムのセキュリティ強化でサポートされることを前提としています。

7.11.1 リスク

プログラミング言語および結果として生じるアプリケーションを強化しないというリスクは、悪用しやすいアプリケーションです。PHP のようなプログラミングシステムの一部はバグが多いことで有名であり、プロのエンジニアリングチームが使用することはありません。Python などの他の言語は運用環境には適していますが、評価する必要のある微妙なセキュリティリスクがあります。従って、結果として生じるリスクのボラティリティは、クリティカルレベルから良性レベルまでの範囲になる場合があります。エンジニアリングチームは、リスク評価と脅威モデリングのプロセスを使用して、運用環境に最も適した言語を十分に評価する*必要があります*。

7.12 永続的なペンテストの実装

新しいエンドポイントを現場にリリースしていつでも構成できる IoT の展開にとって、展開時にのみセキュリティ監査を実行するだけでは不十分です。脆弱なエンドポイントソフトウェアおよびセキュリティ対策が施されていない構成を早期に検出するには、永続的なペンテスト手法を使用することをお勧めします。

永続的なペンテスト戦略を実装することで、特定された脅威を迅速に検出して早期に管理し、軽減するスピードを上げ、脅威がさらされている期間を短縮することができます。

永続的なペンテストを完備した戦略は、アクセス可能な資産、資産の識別および解析、既知の脆弱性の検証および調査、不安定な構成の確認、適切なレポートおよびアラートを作成し、修復に役立てる必要のある資産の発見を実行するための自動でスケジュールされた方法を提供する必要があります。

7.12.1 リスク

永続的なペンテスト戦略を実装しないリスクは、セキュリティ監査を展開時に 1 度だけ実行するだけかもしれませんが、新しいエンドポイントおよび構成では決して評価されないということです。このような状況は、攻撃者によって侵害されるまで、公開されないものとして識別できない一連の脆弱なエンドポイントを引き起こす可能性があります。

8 中優先度の推奨事項

中優先度の推奨事項には、エンドポイント技術設計の選択に応じて関連する推奨事項が含まれています。例えば、オペレーティングシステムレベルのセキュリティ強化を実施することは、エンドポイント上でオペレーティングシステムが実行中の場合にのみ有効です。エンドポイントがモノリシックのカーネルアプリケーション、ま

たは単一の埋め込みアプリケーションで埋め込みリアルタイムオペレーティングシステム（RTOS）で構成されている場合、推奨事項は適用されない場合があります。推奨事項がエンドポイント設計に適用*される*場合は、実装する必要があります。

8.1 オペレーティングシステムレベルのセキュリティ強化実施

オペレーティングシステム上で実行しているアプリケーションは、基礎となるオペレーティングシステムおよびカーネルのセキュリティ強化を（透過的にまたは意図的に）使用するように設計する必要があります。これには以下のような技術が含まれます。

- ASLR
- 非実行可能メモリ（スタック、ヒープ、BSS、ROData など）
- ユーザーポインタデリファレンス保護（UDREF）
- 構造漏洩（情報開示）保護

埋め込みシステムで使用される各オペレーティングシステムは、時には異なる名前の下で、これら技術の様々なバリエーションと組み合わせを提供します。オペレーティングシステムおよびカーネルが何を提供できるのかを判断し、可能であればこれらの技術を有効にして、アプリケーションのセキュリティを強化します。

課題は、各オペレーティングシステムができることを特定することにあります。例えば、メモリ管理ユニット（MMU）のないプラットフォーム上で実行されているアプリケーションでは、ASLR が可能でない場合があります。ただし、メモリ保護ユニット（MPU）のみの環境でも UDEREF に相当するものを実施することが*できます*。使用する技術およびその機能を評価し、アーキテクチャ、カーネル、オペレーティングシステム、およびアプリケーション保護の組み合わせによって実現できるセキュリティレベルを判断してください。

8.1.1 リスク

本推奨事項を実施しないと、アプリケーションのランタイム環境が実質的に悪用されやすくなります。これらの機能強化は、脆弱なサービスに対して信頼性の高い悪用を開発できる（仮にそうであっても）敵対者の数を大幅に制限します。

ートコード実行機能を乱用される可能性がある場合、ASLR、NX、UDEREF などの技術を実行して悪用の可能性をなくすことができます。これにより、悪用を開発する攻撃者は、個々のターゲットに対してカスタマイズが必要な高度で難しい技術を用いる必要があるため、攻撃者がある程度の時間で悪用を開発するこ

とを制限します。これは、難易度を上げるだけでなく、完全に機能する悪用を実現するために必要な時間と経費も増加させます。

これらの機能強化がなければ、完全に機能する悪用は、既製の無料で入手できるソフトウェアを用いて数時間で開発できます。

8.2 デバックとテスト技術の無効化

製品を開発しているときは、エンジニアリングプロセスを容易にするためにデバッグ技術およびテスト技術を用いて有効にすることがよくあります。これは実に標準です。ただし、デバイスが運用展開の準備を整えた際には、承認済みの構成を定義する前に、これらの技術を運用環境から撤去する必要があります。

製品が展開される承認済みの構成には、敵対者が悪用する可能性のあるデバッグ、診断、またはテストのインターフェイスを決して含めてはいけません。前述したインターフェイスは以下の通りです。

- コマンドラインコントロールインターフェイス
- 冗長なデバッグ、診断、またはエラーメッセージを含むコンソール
- JTAG や SWD などのハードウェアデバッグポート
- デバッグ、診断、テストに使用するネットワークサービス
- SSH や Telnet などの管理インターフェイス

承認済みの構成では、前述した技術はすべて無効にする必要があります。

システムで取り外しが可能なシリアルポートも、回路基板から物理的に取り外す必要があります。ただし、UART/USART などのシリアルポートは、マイクロコントローラまたはプロセッサ上のハードウェアピンで何度も有効になります。依然としてこれらのピンがコンソールとして有効になっている場合、敵対者はピンをタップするだけでコンソールと対話できます。DB9 インターフェイスなどの物理シリアルポート自体を取り外しても、コンソールは無効になりません。

さらに、JTAG や SWD などのデバッグポートは、単にソフトウェアを使用して無効にすべきではありません。これらのデバイスは、セキュリティヒューズやロックを変更して無効にする必要があります。これらの技術をソフトウェアから無効にすると、ソフトウェアがインターフェイスを無効にする前に、敵対者が JTAG、SWD、または類似したハードウェアデバッグインターフェイスに接続できる絶好の機会を与えてしまいます。この絶好の機会は、敵対者が成功するには十分なものです。

8.2.1 リスク

この推奨事項を実装しなければ、組織自体が中央処理装置から重要な秘密を抽出してしまいます。これにより、敵対者は自身のファームウェアを NVRAM または EEPROM に読み込ませ、IoT ネットワークまたはデバイスをさらに侵害する重要な秘密を抽出したり、変更したりできるようになります。

デバッグポートを無効にすることは、IoT 製品やサービスの整合性を確保するための重要なステップです。しかし、組織がこれらの技術を無効にするリスクを評価し、フィールドで特定された問題を診断してデバッグできるというメリットと比較検討することが重要です。実行中のシステムをデバッグする方法がない場合、製品における生産レベルの欠陥を修復することが極めて困難になる場合があります。

8.3 周辺型攻撃を介して汚染されたメモリ

処理システムは、アルゴリズムの出力が、指定した一連の入力に関して予測可能であることを確実にするために一貫性に依存します。処理システムはまた、コンポーネントが確実に動作することを期待しており、書き込まれたすべてのビットについては、プロセッサが変更するまでそのビットは安定し不変です。クローズドシステムでは、この理論が適用されます。このモデルに対して異常が発生すると、処理環境が侵害されるか、容易く損傷を受けたりする可能性があります。

情報セキュリティは、別の方法でアクセスできないオブジェクトへアクセスするために意図的に誘導された異常のクラスを提起しています。敵対者に有利な異常動作を誘導する乱用可能なウィンドウは、直接メモリアクセス（DMA）です。分かりやすく言うと、DMA は、外部コンポーネント（周辺機器）が CPU の干渉なしにメインのプロセッサメモリにアクセスできるようにプロセッサが使用できる技術です。つまり、CPU は、周辺機器がメモリ領域へ直接アクセスすることを許可することができます。この周辺機器は、そのメモリ領域に対して読み取りや書き込みを行うことができます。

プロセッサが周辺機器で使用可能なメモリ領域を適切に制限していない場合、周辺機器は目的の機能に必要とされるよりも多くのメインメモリにアクセスすることができます。言い換えると、周辺機器（例えば、イーサネットコントローラ）に、受信したイーサネットフレームの循環バッファとして使用するために DMA 領域が割り当てられ、割り当てられた DMA 領域がメインメモリの全範囲を含む場合、イーサネットコントローラのファームウェアは、すべてのシステムメモリに対して任意に読み取りおよび書き込みを行うことができるようになります。CPU には、イーサネットコントローラのファームウェアがメモリに書き込まれるのをブロックする方法がありません。

この攻撃の結果は 2 倍になります。データはメインメモリから漏洩し、ネットワークパケットまたはアプリケーション情報にエンコードされ、秘密または即時の引き出しが可能になります。あるいは、攻撃者は、アプリケーションの実行可能コードを上書きすることで、バックドア（マルウェア）をメインメモリにこっそりと挿入することができます。

プロセッサの観点からは、過度に許容するメモリのウィンドウが悪意のある周辺デバイスによって悪用されたかどうかを特定することはほとんどできません。この攻撃に立ち向かうには、エンドポイントシステムで使用するプロセッサが、予測可能な小さなメモリ領域に対して DMA を制限できるかどうかを特定します。その場合は、メモリの各領域が必要な周辺デバイスごとに定義されていることを確認してください。可能であれば、周辺機器に対して任意のウィンドウメモリを有効にしないでください。

一部のプロセッサでは、DMA ウィンドウの線形メモリまたは仮想メモリのサイズまたは場所に対して細かい制限ができない場合があります。DMA 攻撃は、重要なアプリケーションの IoT エンドポイントに対する現実的な脅威とみなす必要があるため、より細かな機能を備えた代替プロセッサを検討することが理にかなっているかどうかを評価してください。

IEEE1394、Thunderbolt（サンダーボルト）、Express Card（エクスプレスカード）、または PCI（Peripheral Component Interconnect）DMA へちょくせすアクセスできるその他のポートなどのポートを公開するプラットフォームでは、あらかじめ準備され、費用対効果の高い攻撃がすでに利用可能となっています。

DMA ベースの攻撃がローカルハードウェアコンポーネントの悪用を必要とするプラットフォームでは、難易度が確実に上がりますが、ローカルエンドポイントの侵害に対して DMA を破壊するために周辺機器のファームウェアをリフラッシュすることは、攻撃に基づいたセキュリティ関与の範囲外ではありません。しかし、コスト、時間、および専門知識が要因となり、この場合のアクターはスポンサー付き（有料）の敵対者となる可能性が高くなります。

8.3.1 リスク

DMA が外部コンポーネントによって悪用される性能を制限しないことを選択すると、プラットフォームが完全に侵害されるか、少なくとも秘密キー、プライバシー中心のデータ、または知的財産がエンドポイントから抽出される場合があります。

8.4 ユーザーインターフェースのセキュリティ

タッチスクリーン、リッチディスプレイ、または代替インターフェイス技術などのユーザーインターフェイスを備えている IoT エンドポイントは、ユーザーに情報を提供し、安全な方法でユーザーから情報を取得できる必要があります。

パスワードなどのユーザーインターフェイスの属性については、本書においてすでに取り上げていますが、考察する必要のある微妙な問題がいくつかあります。

- 警告システム
- アクションの確認

物理的な改ざんやアプリケーションが意図しない方法で動作するなどの異常が発生した場合、ユーザーは目に見える警告を受け取る必要があります。あるいは、ユーザーは、システムからのアラートをユーザーインターフェイス内から確認できる必要があります。

さらに、あるインターフェイスから別のインターフェイスへのエンコードまたはシームレスな移行によって動作するデバイスが実行するすべての操作は、ユーザーによって確認される必要があります。この例としては、デバイスカメラが QR コードを読み取るか、デバイスが URL に接続する NFC または RFID の対話要求がある場合です。このような場合、ユーザーは操作を確認し、実行した操作が望ましいことを検証するように指示する必要があります。操作を取り消すオプションをユーザーに提供する必要があります。ユーザーは、接続する URL 全体を含め、特定の操作に関するすべての詳細を表示できる必要があります。

8.4.1 リスク

この推奨事項が実装されていない場合、ユーザーは検出されない攻撃に対して脆弱になります。一部のシステム設計者は、RFID チップから対応する製品の Web サイトへの移行のシームレス性を高く評価していますが、この動作については、望ましくない影響がある場合があります。ユーザーは、望ましくない資料を同意なしに閲覧することを余儀なくされたり、騙されて Web サイトに訪問したり、セキュリティ状態やプライバシーを弱める操作をしたりする可能性があります。

また、アラートの確認に苦労するユーザーは、潜在的に改ざんされたデバイスを使用するリスクを理解できない場合があります。これにより、ユーザーの物理的セキュリティが低下し、危険にさらされる可能性があります。

8.5 サードパーティコードの監査

ブートローダなどのコードの一部が、安全なランタイムプラットフォームを構築する際に重要なコンポーネントとなるときはいつでも、リスクを監査する必要があります。敵対者がブートローダを操作し、信用できないコードを実行したり、認証シーケンスを無視する場合、ブートローダは役に立たなくなる。これは、この技術の展開において、組織が利用する財源、時間、および経験を台無しにし、エンジニアリング費用を無駄にします。

この分野におけるセキュリティのギャップはまた、なりすまし、API の乱用、データの傍受、デバイスのクローニング、さらにはデバイスのリブランディングを通じて、競合企業のビジネスに対する優位性をもたらす場合があります。従って、技術が乱用の危険にさらされていないことを確保するには、承認されたサードパーティがコードの重要部分を監査することが不可欠です。そこで、監査を実施するのに適切な情報セキュリティチームを見つけるには、どのタイプのコードを監査するのかを評価します。通常、このモデルにおいては、以下のことを意味します。C、アセンブリ、場合によっては C++または Java。

これらの言語に精通しているチームはもちろん、基礎となるアーキテクチャを特定します。多くの情報セキュリティチームがソースコードの監査を実施していますが、IoT ビジネスで使用する特定のプラットフォームについての監査を実施するチームはあまりないです。各プラットフォームには微妙な違いがあるので、使用するプラットフォームに精通したチームを活用するのに越したことはありません。

8.5.1 リスク

内部で開発した技術を評価するためにサードパーティのコンサルタントを雇うのは難しい場合がありますが、セキュリティの要件となっています。これは、技術を開発するエンジニアが、彼らのアーキテクチャが証明可能であることを示すことができなければならないからです。アーキテクチャを開発しているエンジニアだけがそれを検討している場合、これは難しくなります。エンジニアは、*実際の実装*からではなく、設計・実装を*試みた*アーキテクチャからコードベースを可視化する傾向があります。従って、セキュリティのギャップを引き起こす可能性のあるアーキテクチャおよび実装の微妙な点を見つけるために、サードパーティの目が度々必要となります。

8.6 プライベート APN の活用

3GPP 移動体通信ネットワークでは、アクセスポイント名（APN）は、認証済みデバイスのセット専用に構成されたプライベートネットワークとして機能します。通常、プライベート APN（「セキュア APN」とも呼ばれます）は、特定のビジネスに関連付けられた認証済みのデバイスにのみアクセスできるプライベートネットワークです。APN を利用することで、企業は、エンドポイントが移動体通信ネットワークを介してサービスインフラスト

ラクチャに接続できるようにすることを制限できます。これにより、バックエンドインフラストラクチャの IoT サービスに直接アクセスできるユーザー数を削減できます。

プライベート APN のその他の属性は、不正エンドポイントが IoT エコシステムを悪用する可能性を減らすのに役立ちます。ファイアウォールは、APN から接続できるサービスまたはコンピュータを制限できます。適切に構成された APN は、エンドポイントが互いに直接接続できないようにします。これにより、侵害されたエンドポイントはネットワークインフラストラクチャを介して他のエンドポイントへ水平に移行することができなくなります。

セキュアな APN 内で利用できる技術を判断するために、組織が携わっている携帯電話会社または仮想移動体通信事業者（MVNO）と連携します。監視、異常デバイスのブラックリスト化、ユーザーID と操作の紐づけなどの他のサービスも利用できる場合があります。

8.6.1 リスク

プライベート APN を利用することで、多様な攻撃を軽減できます。例えば、プライベート APN を使用すると、企業はエンドポイントからインターネットに直接接続できる接続数を削減できます。エンドポイントは、信頼できないインターネットリソースに直接接続することを決して許可するべきではありません。パートナ組織のみが信頼され、彼らのサービスが認証される必要があります。

プライベート APN を使用しないと、侵害されたエンドポイントは、任意のインターネットサービスやプロトコルと制限なく通信できます。これにより、敵対者は別のインフラストラクチャに第 2 の攻撃を開始するために、エンドポイントを悪用する場合があります。これには DoS（サービス拒否）攻撃が含まれる可能性があります。または、他の企業、政府、または民間人に対するより危険な攻撃を促進するのに役立つ可能性もあります。

しかし、注目すべきは、プライベート APN はエンドポイントおよびプライベート APN 間の通信リンクを侵害する敵対者のリスクを軽減しないことです。さらに、プライベート APN はバックエンドサービスへのゲートウェイとしてのみ機能し、APN および IoT サービス提供者のプライベートネットワーク上のバックエンドサービス間のセキュリティを強化しません。セキュリティにおけるこれら潜在的なギャップは、プライベート APN を使用することで得られる改善に関わらず、個別に対処する必要があります。

8.7 環境ロックアウトしきい値の実装

埋め込みシステム内のコンポーネントは、特定環境のしきい値内で使用するように設計されています。これには、電圧レベル、電流の流れ、アンビエントまたは動作温度、および湿度が含まれます。各コンポーネントは通常、承認済みレベルの特定ウィンドウに対して評価されます。デバイスが定められたウィンドウの上または

下の状態にさらされた場合、コンポーネントは不安定に動作したり、敵対者に役立つ方法で動作したりする場合があります。

従って、これらの環境レベルの変化を検出して、デバイスを実行し続けるべきなのか、電源を切るべきなのかを判断することが重要です。ただし、電源を切ることが望ましい結果であり、敵対者がサービス拒否を活用するためにこのエンジニアリングの決定を悪用する場合があることに注意する必要があります。エンジニアリングチームはこのモデルを評価して、シャットダウンするほうが有益であるか、オンラインの状態を試みるほうが有益であるかを判断する必要があります。

それにも関わらず、モデルは通常以下を組み込んでいます

- 節電および節電検出（電圧が低下し過ぎる場合）
- 電圧レベルがしきい値を超えないようにする電圧上限回路の保護
- 電流の流れが一定レベルを下回ることも上回ることもできないようにする、電流制限回路
- 内部レベルを監視する CPU、MCU などのコンポーネントの内部温度監視
- 環境が過度に湿ったり乾燥し過ぎているかどうかを判断するために評価できる湿度レベル（オプション）

高温は、ユーザー、環境、さらにはハードウェアまたはソフトウェアの問題によって引き起こされる回路の問題を示すことができるので、温度は非常に重要です。温度を監視することで、オペレーティングシステムやアプリケーションがリソース（またはデバイス全体）をシャットダウンして、火災や別の問題がエンドポイントによって引き起こされないようにすることができます。

また、低温レベルはデバイスの動作を変化させます。これにより、回路の動作が遅くなったり、コンポーネントが予期しない方法で反応したりする場合があります。温度が有益な方法でアプリケーションまたは回路に影響を与える予測可能な異常を引き起こす場合、これは攻撃者にとって有益です。

温度および湿度を解析する際に、ロックアウトしきい値の問題が明らかになります。電圧レベルおよび電流レベルは、回路基板上またはプロセッサ内の節電および節電回路で軽減する必要があります。エンジニアはチップの電圧と電流のしきい値に関連する数値を調べることができるので、これらの問題に対する保護を容易に実装できます。

温度および湿度については、敵対者は物理的なデバイスに触れることなくこれらのレベルを製造できるので、行動するという決定は少し難しいです。温度の場合、承認待ちの安全イベントを示す可能性のあるレベルは、温度を下げるための適切な措置をデバイスに講じる必要があります。ただし、産業用制御システムや医療機器などの重要な環境においては、デバイスは可能な限り、重要な操作の継続実行を試みる必要があります。エンジニアおよびビジネスリーダーが合意した定義済みの特定ポイントをレベルが超える場合になってから、デバイスはシャットダウンする必要があります。

8.7.1 リスク

電圧および電流の流れについては、悪用のリスクは、グリッチやこれらのレベルの変化から利益を得ることができるサイドチャネル攻撃に関連しています。節電および節電検出がプロセッサに実装されている場合、悪用のリスクは低下します。そうでなければ、リスクは、物理デバイスの安全性に関する問題を引き起こす可能性のある電圧または電流のスパイクに関連するか、攻撃者がグリッチ（および同様の）攻撃をインストルメント化して、コンポーネントのセキュリティを破壊できる可能性があることに関連しています。

これらの問題は、異常なスパイクまたは電圧や電流の低下からコンポーネントを保護する PCB 上の回路を使用することで修復する必要があります。

環境レベルの劇的な変化については、リスクはユーザーの安全性に関連しています。過度の CPU 使用やその他の異常によって引き起こされる高温は、火傷、化学火傷、さらには火災の原因となる可能性があります。

8.8 電力警告しきい値の実行

ユーザーに対して重要なサービスを提供するエンドポイントは、電力関連のイベントを示す警告しきい値で有効にする必要があります。これらのイベントには以下が含まれる場合があります。

- 低バッテリ状態
- 非常に低いバッテリ状態
- 停電イベント
- 節電イベント
- バッテリバックアップイベントへの切り替え

ユーザーは警告を受け、十分な時間をかけて電力喪失を補う必要があります。これは、OK の場合は緑、低の場合はオレンジ、クリティカルの場合は赤など、特定の電源状態を示す LED を有効にすることで実現できます。

交流の主電力に接続されているシステムは、停電または節電イベントが発生した場合、ユーザーに警告するように構成する必要があります。また、エンドポイントは、これらのイベントを永続的なメモリに記録して、ユーザーおよび管理者が後で情報を取得できるようにする必要があります。情報にはタイムスタンプを付ける必要があります。

このプロセスにおける課題は、バッテリの電源がどのくらいで枯渇するのか、そして電源状態の変化をユーザーに通知するために必要となる余分なエネルギーを特定することです。これはすべて電気工学によって実現することができますが、経験豊富なエンジニアリング企業のプロセスには挑戦すべきではありません。

8.8.1 リスク

明確に定義された電力警告システムがなければ、ユーザーは潜在的に重大な電力イベントに対して適切に準備することはできません。ペースカウンター、タイマー、およびその他のウェアラブルデバイスなどのシンプルなデバイスの場合、これは安全である場合がありますが、パーソナルトラッカー、テレマティクスシステム、およびホームセキュリティシステムなどのより重要なデバイスは、電力喪失によって深刻な影響を受ける可能性があります。

8.9 バックエンド接続のない環境

8.9.1 方法

エンドポイント、特にゲートウェイ、またはゲートウェイとして機能するエンドポイントは、バックエンドネットワークへの接続が利用できない環境でも通信セキュリティを実施できなければなりません。この接続の欠如が一時的であるかどうかに関わらず、ゲートウェイまたはエンドポイントは、バックエンドシステムが利用できるかのようにセキュリティを実施しなければなりません。

これを実現するには、TCB を使用してエンドポイントがプライバシー中心の構成、またはコマンドデータを通信する必要のあるすべてのピアを認証する必要があります。TCB は、ピアから送受信したメッセージが同じ組織によってプロビジョニングされたエンティティから送受信されていることを保証するために使用できます。これは、敵対者のデバイスが通信される可能性を低減します。

相互運用性は、認証できないその他のデバイスと通信することで今もなお実現できます。しかし、これらのデバイスに伝達される情報のタイプは、相互運用性のある非機密性のデータクラスに限定する必要があります。

課題は、認証するエンドポイントとクリアテキストで通信するエンドポイントを決定することにあります。組織は、分類するデータタイプおよび認証されていないピアから守るデータタイプを決定する必要があります。このデータ分類が実現できると、組織はコアとなる IoT サービスの手助けがなくても、どのピアが合理的に信頼できるのかを判断することができます。

8.9.2 リスク

通信の少ない環境にソリューションを展開するリスクは、競合関係がインフラストラクチャを悪用する機会をもたらすことです。競合企業は、相互運用性を提供し、接続の少ないサイトを性能試験場として使用することで、ビジネスの価値を下げる可能性があります。

代わりに、組織は相互運用性を可能にすることを選択できますが、ある程度です。コアとなる特定の知的財産およびサービスは、TCB を使用して検証される認証済みピアのみに確保することができます。これは、知的財産に関する問題や敵対者となる競合企業に対してビジネスを公開することを低減するのに役立ちます。

8.10 デバイスの廃止と段階的廃止

本文書の他の部分で述べたように、すべてのエンドポイントデバイスにはライフサイクルがあります。一部のデバイスは、ユーザーがサブスクリプションを取り消すために閉鎖する必要がありますが、その他のデバイスは、異常または敵対者の動作によって閉鎖する必要があります。理由に関わらず、企業は TCB および通信モデルを使用してデバイスを安全に閉鎖する準備をする必要があります。

本文書の他の部分で述べたように、段階的廃止（サンセット）とは、デバイスのネットワーク全体およびそれらのデバイスをサポートするサービスを閉鎖するプロセスです。企業によって廃止された製品やサービス、またはシャットダウンを決定する企業は、段階的に廃止するネットワークを敵対者が奪取して悪用するリスクを軽減するために、デバイスとネットワークを廃止する必要があります。

これを実現するには、TCB および対応するプロトコルを使用する必要があります。一般的にこのプロセスは以下のようになります。

- サービスエコシステムから閉鎖メッセージを作成する
- メッセージを受信する一意のエンドポイントに合わせてメッセージを調整する
- 閉鎖する PSK または非対称キーを使用してメッセージに署名する
- メッセージをエンドポイントにプッシュダウンする
- 閉鎖を暗号で承認するエンドポイントからメッセージを受信する
- 認証済みデバイスリストのエンドポイントを無効にする
- このエンドポイントからのさらなる通信を禁止する

デバイス側では、ソフトウェア上で実行中のアプリケーションは、

- サービスエコシステム上の重要なバックエンドサービスに接続する必要がある
- 重要なメッセージをサービスに問い合わせる必要がある
- 閉鎖メッセージを受信する必要がある
- TCB およびトラストアンカーを使用してメッセージの署名を検証する必要がある

- 受信確認メッセージを生成して、カスタマイズした PSK または非対称キーを使用して暗号署名する必要がある
- 閉鎖操作を実行する必要がある
- 重要なサービスにメッセージを返送する必要がある

閉鎖プロセスには、トラストアンカーからの無効化とセキュリティキーの削除が含まれているため、閉鎖前には、メッセージに署名し、送信準備をすることが重要です。このプロセスにより、閉鎖メッセージに署名するために使用するキーは使用できません。エンドポイントが実際にメッセージを受信して処理したことを保証するために、サービスは整合性が証明できるメッセージの受信を必要とします。

このプロセスの問題は主に、潜在的に侵害されたデバイスを閉鎖することは、デバイスが閉鎖コマンドを拒否するポイントまで侵害されていないことを前提としていることにあります。十分に侵害されている場合、閉鎖コマンドを受け取ることはできません。

結果的に、サービスエコシステムで実行するバックエンドシステムは、エンドポイントと重要なサービスとの通信を無効にすることが不可欠です。デバイスがネットワーク接続されたピアまたは重要なサービスと対話しようとする場合、バックエンドシステムはアラートを発し、管理者に異常イベントが発生したことを知らせる必要があります。

8.10.1 リスク

閉鎖および段階的停止を実装しないというリスクには多くのことがあり、敵対者がネットワーク全体を完全に乗っ取ることから、侵害されたデバイスがネットワークに接続されたサービスを利用し続けることまでに至ります。最も一般的なリスクは、IoT サービス提供者とのサブスクリプションを終了したユーザーに関連しています。このようなユーザーがネットワークからの使用を停止していない場合、IoT エンドポイントのネットワーク内の他のピアとの通信を継続することができたり、ユーザーがもはやアクセスできないサービスにアクセスできる場合があります。これは、サービスエコシステムにおける帯域幅、CPU 時間、およびストレージに支払う必要のある IoT サービス提供者に代わってコストが発生します。

8.11 不正なメタデータの収集

現代の IoT は、物理世界をデジタル世界に橋渡しするために設計されています。この近代的なモデルでは、技術の影響は過去よりもはるかに侵略的である可能性があります。メタデータを使用すると、企業または個人はランダムまたは特定のコンシューマーの動作を意図的に追跡および監視することができます。

メタデータ解析は、2 つのネットワークエンティティ間の通信が暗号化されているが、メッセージのタイプまたは送信者や受信者の ID を識別するプロトコル構造が公開されている場合に使用されます。このメタデータを使用してインテントを導き出すことができます。

自動車が特定のコンシューマーに帰属するメタデータを含んだメッセージを発するシナリオを考えていましょう。これらメタデータの要素を（ローカルまたはリモートのいずれかで）追跡できる人は、コンシューマーの動きを監視し、その動きから動作やインテントを導き出すことができる場合があります。自動車のテレマティクスシステムで攻撃に利用できるセキュリティ上の欠陥がある場合、特定のコンシューマーのテレマティクスシステムを追跡して標的にし、物理的な危害の危険にさらす可能性があります。

合法的な組織や保険会社は、これらのリスクが将来の自動車への財源にどのように影響するかを懸念しており、エンジニアがテレマティクス機器を設計する方法を決定する法律や基準に関与し始めています。この変化は、より多くの技術が開発されるにつれて、いずれは活発ではない IoT 業界へと徐々に浸透していくでしょう。

メタデータ収集に立ち向かうには、可能な限り多くのデータを暗号化し、通信モジュールに一意のバイナリ識別しを使用します。外部ユーザーが IoT システムの API を使用してユーザープロファイルからハードウェアのシリアル番号やその他追跡可能な ID を取得できないようにするポリシーを施行します。可能であれば、メッセージの構造をサードパーティに公開しないようにします。操作、アクティビティ、動作がサードパーティに公開されないようにします。ユーザーのプライバシーに関係するすべてのデータについて機密性と整合性を強化します。

8.11.1 リスク

弱い通信セキュリティを使用すると、エンドユーザーを危険にさらすデータまたはメタデータ収集を可能にしたり、エンドユーザーのプライバシーを公開してしまう場合があります。保険代理店は、技術におけるエンドユーザーのプライバシー要件を強化するケースを構築しているので、企業がデバイスが生成するデータに対して責任を追わない場合、その企業自体が危険にさらされる場合があります。

9 低優先度の推奨事項

優先度の低い推奨事項には、抵抗に対して非常にコストがかかるリスクに適用される推奨事項や、エンドポイントの設計に影響を与える可能性の低い推奨事項が含まれています。これらの推奨事項は有益であり、推奨事項の中で詳述している情報は重要ですが、説明する軽減戦略や修復戦略は、企業には範囲外である場合があります。各推奨事項を評価し、説明するリスクが企業および顧客に関連しているか、また

は重要であるかを判断します。顧客がこれらのリスクに対処する必要がある場合は、推奨事項を適用します。

9.1 意図的なサービス妨害および意図的でないサービス妨害

無線通信の場合、*ジャミング*の絶え間ない脅威や、正規の信号をスクランブルするために使用できるノイズやパターンの意図的なブロードキャストがあります。無線信号は、特定のパターンで空間を飛ぶ電子から構成されているだけなので、通信データを形成するパターンを中断または破壊する一連の信号を作るのはかなり簡単です。

通常、このような攻撃の目標は単に中断することで、正当なユーザーに対してサービスを許可しない、または拒否することです。その他の場合では、悪用はより目的があるかもしれません。例えば、認証メカニズムを備えていない通信プロトコルを*偽装する*ことができます。これを実現するには、敵対者がなりすました信号が計画したターゲットに到達する可能性が高くなるようにするために、実際の信号を*妨害させる*必要があります。

この一例として、グローバル・ポジショニング・システム（GPS）のなりすましがあります。民間の GPS 信号は、本質的に誰でも取得できるプレーンテキストのブロードキャスト信号なので、暗号化と認証が欠けています。また、比較的弱い無線信号でもあり、テレビ受信機やマイクロ波用の極超短波（UHF）プリアンプなどの環境異常によって容易に減衰します。

位置情報を適切に機能させる必要があるデバイスの場合、妨害された GPS 信号は、特になりすましが後に続いて使用される場合、情報セキュリティリスクへとつながる可能性のある、信頼性に関するリスクをもたらす可能性があります。

妨害や他の形態の意図的なサービス拒否（DoS）攻撃に立ち向かうには、サービスの中断を低く評価する方法に焦点を当てた堅牢な通信プロトコルを開発します。ネットワークは、デバイスが突然または異常にネットワークから離れたかどうかを検出する必要があります。各エンドポイントは、それがネットワークから離れようとする際は、「別れを告げる」する必要があります。そうでない場合は、統計解析のために異常を記録する必要があります。

さらに、通信セキュリティキーは、デバイスがネットワークに再び加わるたびに、再度ネゴシエートする必要があります。同じ通信セキュリティキーは使用すべきではありません。同じ非対称キーで*ブートストラップする*必要がありますが、キーのネゴシエーションから派生した対称キーは、各通信セッションごとに新規である必要があります。

意図的でない妨害は、信号伝搬を無効にする環境条件、機器の誤動作、さらには同じ周波数で動作する隣接機器など、多くの理由でラジオで発生する可能性があります。根本的な理由に関わらず、無線通信に依存するエンジニアは、信号劣化または信号損失を引き起こす一時的な条件が存在すると予想しています。これらの損失は、アプリケーション*および*ネットワーク通信プロトコルの設計によって補正する必要があります。

開発者には、意図しないサービスの拒否攻撃に対する保護方法についてのアドバイスを含んでいる GSMA's Connection Efficiency Guidelines [9]を読み、デバイスホスト ID レポート（DHIR）に関するガイダンスを提供することをお勧めします。

9.1.1 リスク

意図的な DoS のリスクに立ち向かうことができないと、エンドポイントの動作が異常となったり、不安定になります。エンドポイントが常に同じセッションキーを使用している場合、敵対者がネットワークアーキテクチャを悪用して通信を保護するために使用している対称キーに関する情報を収集する可能性があります。エンドポイント通信のセキュリティに関しては、セッションが切断された後にセキュリティで保護されたセッションを適切に構築することが不可欠です。

9.2 安全性に関するクリティカル解析

モノのインターネット製品の大多数は、物理世界の側面の一部をデジタル技術に組み込んでいます。結果的に、人間は、IoT エンドポイントが提供する情報に基づいた物理世界で意思決定を下す可能性が高くなります。あるいは、IoT エンドポイントは、デジタル世界を通じて得られる情報を用いて物理世界に影響を及ぼす意思決定を下す場合があります。

従って、IoT サービス提供者は、安全性の観点から製品を評価し、人間の生活が技術の影響を受けるかどうか、どのように影響を受けるか、そしていつ影響を受けるのかを見極めることが不可欠です。物理的な危害を引き起こすために技術を悪用できないように適切な予防手段を整備していない場合、顧客は危険にさらされる可能性があります。

安全上の問題を解決するには、IoT サービス提供者の幹部、法務、および保険チームと話し合いを持ちます。これらのチームが製品やサービスに使用している技術の機能および限界を理解していることを確認します。使用している技術が企業のニーズを満たすことができるかどうかを見極め、予定したアプリケーションに必要な安全レベルを顧客に提供します。

9.2.1 リスク

製品やサービスが顧客の安全に及ぼす影響を評価する時間を取らないと、明らかに、収益の損失、不測の事故、さらには人命の損失につながる可能性があります。

9.3 シャドウコンポーネントと信頼できないブリッジの無効化

物理回路上のコンポーネントは通常、互いに通信する際、または中央処理装置と通信する際、うわべだけの機密性および整合性は使用しません。結果として、敵対者は、これらのバス上で送信されたデータよ読み取りまたは書き込みを行うことができます。通信セキュリティにおけるこのギャップの影響は、敵対者が物理回路上の正当なデバイスに偽装できることです。敵対者が選択すれば、NVRAM、RAM、さらにはトラストアンカーなどの重要なコンポーネントを偽装する可能性があります。

この攻撃の目標は、バス上にある 2 つのコンポーネント間で使用するセキュリティを無視することでしょう。このシナリオの典型的な例は、この弱点を利用して NVRAM に格納したアプリケーションイメージを解析する整合性検証プロセスを無視することです。CPU が NVRAM に格納したメモリを取得する際、攻撃者はパススルーシステムを使用して、実際のメモリ内容を CPU に提供することができます。CPU 上で実行しているアプリケーションがアプリケーションイメージの整合性を検証すると、攻撃者は物理バス上の通信をインストルメント化して、攻撃者にとって有益となる NVRAM コンテンツを選択的にスワップアウトすることができます。つまり、CPU は 1 つのアプリケーションイメージ（元のイメージ）を検証しますが、その後攻撃者のイメージを RAM に読み込み実行します。

この攻撃から保護するため方法の 1 つは以下の通りです。

- NVRAM の内容を RAM に読み込む
- RAM に読み込んだアプリケーションイメージを検証する
- コードを RAM で直接実行するか、RAM の内容をキャッシュする

この時点で注意すべきは、攻撃者が RAM も破壊して、このプロセスを弱める可能性があることです。しかし、RAM に対する中間者攻撃を実行することは、NVRAM に対する攻撃よりもはるかに複雑でコストがかかります。それは、バスとアクセスパターンの速度が、主にブロックでアクセスされる NVRAM よりもはるかに高速で不安定だからです。

代わりに、攻撃者は検証済みの NVRAM コンテンツのより小さな領域に対してチェックサムを作成し、NVRAM から定期的に署名をチェックすることができます。作成したチェックサムが異なる場合、コンテンツが

操作されています。これは成功する可能性がありますが、敵対者は実行中のアプリケーションがランダムにチェックしない少量のデータしか操作できないため、成功する可能性は低くなります。

注意すべきことは、この攻撃から保護する最善の方法は、NVRAM のコンテンツを検証してから実行可能な RAM に読み込むことですが、この問題を完全に解決するソリューションはないということです。物理コンポーネントを保護するコストは非常に高いので、顧客がそのようなセキュリティ保証を必要としない限り、この攻撃をより完全な方法で解決することは現実的ではありません。

この攻撃は、I2C などの基本的な物理通信プロトコルを使用すると、さらに単純化されます。I2C などのバスは、本質的に物理的なブロードキャストシステムです。従って、I2C バスにあるコンポーネントは、他のコンポーネントを装うことができます。これにより、敵対者は通信チャネルを保護する上で機密性および整合性を強化していないバス上にある他のデバイスを偽装することができるようになります。この点が懸念される場合は、物理バスプロトコルの上で使用するアプリケーションプロトコルの機密性と整合性を強化します。

9.3.1 リスク

ソリューションを一切実装しないというリスクは、敵対者がアプリケーションの整合性チェックを無視できるようにしてしまいます。これにより、攻撃者はブートローダや TCB など、より権限のあるコードで実行しているアプリケーションを侵害することができるようになります。

ただし、注目すべき点は、この攻撃はブートローダに対する単純な攻撃よりもはるかに可能性が低いです。NVRAM のようなコンポーネントや RAM のような高速コンポーネントに対するハードウェアの中間者攻撃を実行することは困難で複雑、そして現在は高価となっています。攻撃者がこの方法で埋め込みシステムを破壊することはいつでも可能ですが、あまりにも桁違いなコストがかかるので行うことはできません。

従って、コードを RAM に読み込み、整合性を検証することは、攻撃がたとえあったとしても、その大半を無視できる妥当なソリューションとなるかもしれません。

また、上記およびその他の理由により、暗号化キーはこれらのような不安定な権限の中で保持されるべきではありません。暗号化キーはトラストアンカーの中で格納して TCB を用いて使用すべきで、偽装されたり侵害される可能性のある NVRAM などのメディア内に格納すべきではありません。

9.4 コールドブート攻撃の無効化

コールドブート攻撃[参照]は、実行中のコンピュータから物理メモリを取り出し、敵対者が管理するセカンダリシステムにそのメモリを配置することで秘密を抽出するといった、コンピュータシステムに対する物理的な攻撃

戦略です。この攻撃のメリットは、攻撃者が RAM のコンテンツを永続的なストレージにダンプするカスタムオペレーティングシステムを実行できることです。これにより、攻撃者は取得したデータを綿密にチェックし、使用できるセキュリティ関連のトークンが存在するかどうかを判断できます。これには次の内容が含まれます。

- 暗号化秘密または秘密キー
- ログイン資格情報（ユーザー名およびパスワード）
- 個人識別可能情報（PII）
- Web サービス用のアクセストークン

この攻撃の目標は、攻撃者が別の方法では手の届かないであろうリソースに対して長期的にアクセスできるようにする秘密を侵害することです。例えば、最新の TLS 標準で使用している暗号化アルゴリズムを破ることは、平均的な攻撃者にとっては不可能でしょう。しかし、相互認証 TLS で使用するプライベートクライアント証明書を侵害すると、攻撃者はより便利なシステムからクライアントをシミュレートできるようになります。

この攻撃を成功させるには、攻撃者はチップに格納されたビットを変更することなく、ターゲットのコンピュータシステムから RAM を取り外すことができなければなりません。研究論文に詳述されているように、これはメモリチップを冷却することで実現できます。ただし、RAM は容易に取り外すことができなければなりません。RAM が回路基板にはんだ付けされている場合、攻撃が非常に複雑になり、攻撃者が半田ごてを使用してメモリを抽出する必要があり、これはコンテンツを損傷する可能性があります。

シャットダウン時にメモリをスクラブすることは、エンドポイントのプライバシーを強化する上で常に有益で、知らされていることに注意することが重要です。しかし、コールドブート攻撃はシステムが実行中であっても、いつでも発生する可能性があります。従って、メモリをスクラブすることは役に立ちますが、現実の攻撃を打倒することは成功しない場合があります。

この攻撃に対するより効果的な軽減策は、CPU 内部にある RAM を使用してセキュリティ中心の操作を処理することです。多くの CPU、MCU、および MPU には、実行中のアプリケーションで使用できる少量の内部 SRAM があります。アプリケーションが重要なセキュリティトークン（秘密キーなど）をこの内部 RAM に使用することを制限する場合、取り外し可能な（または外部の）RAM のコンテンツは攻撃者にとって重要でなくなります。

9.4.1 リスク

コールドブート攻撃のリスクを考慮しないと、単純な攻撃モデルを用いて重要なセキュリティキーが抽出される場合があります。セキュリティキーが IoT サービス提供者のエコシステム内のすべてのエンドポイントに対して共通である場合、大規模な侵害の可能性があります。

詳細については、https://citp.princeton.edu/research/memory/を参照してください

9.5 不明確なセキュリティリスク（壁を通して見る）

通信ネットワークの相互認証、機密性、および整合性を有効にして強化するにも関わらず、トラフィックパターンはイベントに直接相互関係があります。データが特定の物理イベントに応答してトラフィックされると、最終的に物理イベントとデータ間に相関が生じます。これにより、敵対者は信号パターンを監視し、プレーンテキストデータに直接アクセスできるかどうかに関わらず、パターンから意味を引き出すことができるようになります。

これについての例として、特定の部屋におけるユーザーの物理的な存在に基づいて反応するホームオートメーション技術があります。通信システムをリモートで監視できる敵対者は、IoT エンドポイント、ゲートウェイ、およびバックエンドシステム間の通信パターンを見るだけで、特定の家にいるユーザー数、そのユーザーが家のどこにいるのか、そしてそのユーザーは誰なのかについて観察できる可能性があります。

敵対者は、人口密度の多い家、1 人の個人だけが独りでいる家、そしてその個人が家のどこにいるのかを簡単に区別することができるかもしれません。保険会社や法人は、このことが自家所有者や生活圏のテナントに対するリスクを潜在的にどのように高めているのかについて理解する必要があります。

このリスクに立ち向かうのは難しいことがあります。これを行うための最も一般的でシンプルなモデルは、サンプルを取得するユーザーが存在するかどうかに関わらず、事前に定義された速度でサンプルを送信することです。機密性および整合性が強化され、遠隔にいる敵対者がデータのプレーンテキストの評価を拒否されると、オブザーバーは、ユーザーのアクティビティを含んだサンプルと空のサンプルを区別できなくなります。

しかし、このモデルには、スペクトル飽和度の増加、低電力またはバッテリ対応技術の消費電力の増加、そして空のサンプルパケットの復号化、検証、および解釈に必要な処理レベルの増加などの懸念点があります。

代替手段として、バーストを変更して、ランダムな間隔でサンプルを送信する方法があります。このタイプのパターンは、コストがかからず、消費電力が少なく、処理能力が少なくて済みます。それでも、ユーザーの存在を示す微妙な変化を観察することは可能かもしれません。例えば、本当にエントロピーなシステムは完全にラン

ダムであり、予測不可能です。ただし、ユーザーの動作は完全に予測可能です。ユーザーが部屋に入り、その部屋のセンサーが反応して、ネットワーク内のピア IoT エンドポイントにデータを送信し始めると、一貫性のある動作の*導入*によってユーザーの存在が示される場合があります。

このタイプのリスクを対象とする技術を開発しているチームは、プライバシーの露出に関する潜在的な影響を調査し、法務チームに相談して、その技術が企業の法的立場や保険モデルに影響を及ぼすかどうかを見極める必要があります。

9.5.1 リスク

IoT サービス提供者がプライバシー露出の可能性およびセキュリティ上のリスクの観点から自身の技術を評価しない場合、対処*しなければならない*リスクに対して補償するために、アーキテクチャをしっかりと見直す必要があります。後でお金をかけてアーキテクチャを調整しようとするのではなく、作業段階の開始時にできるだけ早くこれらのソリューションを製品に設計します。

9.6 集束イオンビームと X 線への対抗

集束イオンビーム（FIB）は、半導体評価で一般的に使用される製造装置です。この技術は、ナノメートルレベルで回路を検査および変更することができ、アナリストが製造上の欠陥を特定できたり、製造プロセスを変更する前に回路パッチをテストすることができます。

情報セキュリティでは、FIB を使用して内部バスをタップすることで、内部コンポーネントを介してトラフィックされたデータを傍受できます。さらに、FIB を使用して、内部コンポーネントの動作方法を変更する内部回路を変更し、敵対者がセキュリティ制限を無視できるようにすることができます。

ほとんどすべてのデバイスは、FIB の攻撃対象です。しかし、特定のデバイスだけが FIB のプロセスを介して実行されます。これは、FIB 自体が単位当たり約 1,000,000 米国ドルする非常にコストのかかる技術であるからです。技術が高額なため、ツールキットにそのようなデバイスを備えている組織はほとんどありません。さらに、デバイスは自動化されていません。これを使用できるようにするには、操作に対する高度なスキルだけでなく、半導体解析における非常に高度な専門知識が必要になります。従って、FIB の現実的なコストは、単に 100 万ドルをはるかに上回り、ユーティリティ自体、教育、給与およびユーザーの専門知識のための数百万ドルにまで及びます。

ただし、組織はアウトソーシングに応じることができます。リバースエンジニアリングは大部分が合法なので、組織はデバイスのリバースエンジニアリングに関心のある顧客に半導体攻撃サービスを提供できます。これらの契約は、特定のコンポーネントを攻撃するのに必要なカスタマイズと専門知識のレベルに応じて、10,000～

1,000,000 米国ドルの費用がかかります。例えば、アウトソーシング企業は、共通のチップで保護を回避するための*戦略*を持っています。しかし、新しいセキュリティロック技術を備えているカスタム FPGA ソリューションは、既存の*戦略*が定義されていないため、コストがはるかにかかります。FIB を上手く使用するには、新しいプロセスが必要となり、かなりの時間とお金がかかります。

現代のトラストアンカーバリアントなどの新技術の中には、FIB プローブから抵抗を要求するものもあります。これらの要求にはいくつかの妥当性がありますが、バイパス技術の解析に十分な時間をかけた後、動的でない（多くの場合はそうではありません）ハードウェア保護が*戦略*になります。従って、これらの新しい要求は有効かもしれませんが、*時間枠に対して*のみ有効である可能性があります。

従って、このような侵略的だけれどもほとんどが常に成功する攻撃に対して補償するには、エンジニアリング組織がトラストアンカーだけでは成功しないセキュリティ戦略を設計することが不可欠です。それよりも、この技術をベースのトラストアンカーとして用いた十分なプロトコルを設計する必要がありますが、1 つのデバイスの侵害がエンドポイントのネットワーク全体の侵害という結果にならないように各エンドポイントの暗号化キーをカスタマイズします。

敵対者が FIB を用いて*ターゲットとしたいすべてのエンドポイント*から暗号を抽出する必要があるシナリオを考えてみましょう。これはすぐに極めてコストのかかる提案となり、大部分の敵対者の予算に関して範囲外になるでしょう。これらの攻撃方法は*十分に軽減*できないので、曖昧なことからではなく、アーキテクチャを介してリスクを減らし、*価値を下げる*必要があります。

9.6.1 リスク

FIB のリスクは、セキュリティで保護されたコンポーネントであっても、コンポーネントから暗号の秘密やその他の知的財産を抽出できるということです。コンシューマーの IoT に対して、費用対効果のある方法で FIB を打破するのは実現が困難なので、組織はエンドポイントシステムを保護する戦略を変更するか、エンドポイントエコシステム全体の侵害を覚悟する必要があります。

9.7 サプライチェーンセキュリティの考慮

どのようなコンピューティングシステムのセキュリティでも、回路基板を構成する未加工のコンポーネントから始まります。シリコン、暗号トークン、ROM（読み取り専用メモリ）、ファームウェア、およびその他埋め込みシステムのコア属性はすべてこのようなシステムのセキュリティに貢献します。これらコンポーネントのいずれかが改ざんされる場合、システム全体はセキュリティ侵害にさらされる可能性があります。

結果的に、セキュリティを意識している IoT サービス提供者は、コンポーネントのソース、アセンブリ、および組み立てた技術の出荷に使用した遂行プロセスを考慮する必要があります。技術を生むために使用するプロセスを慎重に計画しない場合、プロセスの SPOF（単一障害点）が重大なセキュリティ障害をもたらす結果となります。

以下のような問題点を考えてみましょう。

- どこで誰が製造したシリコンですか。
- シリコン設計は、信頼できるサードパーティの情報セキュリティチームによって解析されていますか。
- シリコンは安全な施設で製造されていますか。
- EEPROM または NVRAM にブートローダなどの実行可能イメージをどのように投入しますか。
- 実行可能イメージをフラッシュするプロセスは安全ですか。
- 実行可能イメージはどのようにして製造元へ配信されますか。
- 実行可能イメージは、EEPROM または NVRAM にフラッシュされた後で検証されますか。
- 暗号の秘密はチップ上でどのようにプロビジョニングされていますか。
- 秘密が製造元で生成されている場合、実績のある RNG を使用してキーを生成していますか。
- すべてのセキュリティキーは TCB の推奨事項ごとに一意ですか。
- 暗号の秘密は IoT サービス提供者とどのように共有されていますか。安全ですか。
- 一意のチップ識別子（シリアル番号など）は、暗号の秘密とどのように相関し、IoT サービス提供者とどのように共有していますか。

製品を構築して組み立てるためによりセキュリティの高い設備を選択するのは、コストが高くなる場合がありますが、それは組織にとって必要不可欠なステップです。これは、製品のユースケース、計画した展開環境、対象顧客、および人々の安全、軍事アプリケーション、および重要なインフラストラクチャの展開などの要因によって異なります。結果として生じる技術が人命に影響を与える場合、サプライチェーンは、セキュリティのギャップについて評価しなければなりません。

9.7.1 リスク

サプライチェーンのセキュリティがなければ、組織は数多くのリスクにさらされますが、その中にはまったく予期できないものがあり、その上企業にとって極めて重要な場合があります。

- エンドポイントのクローニング（不正な製造）
- 技術の盗難（競合企業がサービスプロバイダーから盗んだり、商品の価格を下げること）
- 資格情報の盗難（データ傍受または偽装攻撃）
- 埋め込みの投入（後で活性化される可能性のある悪意のある「バックドア」）

9.8 合法的傍受

合法的傍受とは、顧客とサービス提供者間の通信を合法的に傍受または操作する行為です。これは 2 つの方法のいずれかで動作できます。まず、最も典型的なシナリオは、法執行機関が通信事業者に法的要求を送付し、特定の購読者が行った通信からのメタデータまたは実際のデータへのアクセスを求めることです。第 2 のシナリオでは、法執行機関が、特定の購読者のデータとメタデータの両方、またはいずれか一方へのアクセスを IoT サービス提供者に求めます。機関が通信事業者経由でアクセスを要求するシナリオでは、IoT サービス提供者は法的要求の範囲に応じて、問題があることが全く通知されない場合があります。従って、サービス提供者は、このような機関が行う法的要求を実装する、または準拠する準備を整える必要があります。

従って、提供者は、法的拘束力のある要求から生じるプライバシー問題を特定し、組織の法的組織の行為能力の範囲内で各自の法的モデルおよびプライバシーポリシーに関連する情報を提供する準備を整える必要があります。

近年、Google、Apple などの大企業は、機関に代わって企業に秘密の要求が行われた際、ユーザーへ合法的に知らせるために *令状のカナリア*を採用しています。企業は、合法的な傍受代理人と接して*いない*ことを示すフレーズ、イメージ、またはその他加工物を削除する場合があります。このオブジェクトの削除は当然のことながら、要求が行われたことを示しています。

9.8.1 リスク

合法的傍受の要求に対して企業が準備していないと、そのような要求を企業が受けた場合、企業は不利な立場になります。企業は要求に従う必要があるけれども、法的なインフラストラクチャやプライバシーポリシーを準備していない場合は、潜在的に危険にさらされている可能性があります。

エンドポイントプロトコルおよび IoT プラットフォームが適切な機密性および整合性を準備していないと、企業が何の知識を得られないまま、通信がネットワーク側で傍受される可能性があります。これにより、企業がユーザーデータ漏洩のリスクにさらされる可能性があり、NSA のスノーデンなどのイベントに関連して、ユーザーデータを保護する組織の力量に対する世間の信頼が大幅に低下する可能性があります。

10 要約

要約すると、IoT 製品またはサービスにおけるほぼすべてのセキュリティリスクは、明確に定義されたアーキテクチャ、セキュリティ関連のイベントの前と最中にリスクを特定するインテリジェンス、そしてそのようなイベントを対処するためのポリシーや手順で立ち向かうことができます。IoT サービス提供者にとって重要な高レベルのセキュリティ概念について解析することで、セキュリティに関するよくある質問を見直すことができます。これは、セキュリティアーキテクチャのギャップの解決に最も関連性がある推奨事項へとエンジニアチームを導くはずです。

アーキテクチャの定義が進捗するにつれ、チームはセキュリティに関する質問や懸念事項が自身の実装でより一意になるので、独立した推奨事項を検討できます。

概して、すべてのエンジニアリングチームは、非常に類似したリスクに直面します。組織がリスクのみならず修復に関する戦略について共通の知識ベースを構築するには、懸念事項を同僚と共有することが不可欠です。私たちの組織は力を合わせて、IoT の将来に向けてセキュリティを構築する際にお互いを支援するための技術と知識の両方を構築することができます。

付録A　汎用ブートストラップアーキテクチャを使用した例

マルチホップネットワーク全体のセキュリティレベルは、チェーン内の最も弱いリンクで定義されます。従って、IoT エンドポイントおよびゲートウェイ間のローカルリンクは、セキュリティの全体レベルを同一に維持するために、ワイドエリアネットワークとして同等のセキュリティレベルで保護する必要があります。

これを実現するために候補となる技術の 1 つに、認証だけでなくデータ整合性にも使用できる汎用ブートストラップアーキテクチャ（GBA）[17]があります。これは事前共有キーに基づいており、認証のみならず暗号化の基盤として期限付きキー（トークン）を生成するために使用されます。

認証とは、ある人またはある物、実際にはそれが誰または何であるかと宣言されたかを判断するプロセスです。数十億のエンドポイントがアクティブになっている IoT 空間では、本物かつ信頼できる通信動作を判断することが最も重要です。この信頼関係を築くために確立されたメカニズムは、スケーラブルで維持可能であるいう要件を満たす必要があります。さらに、様々な IoT サービスは、これらのサービスに対応し、さらに共通のインフラストラクチャを維持する認証メカニズムを適合させるという要件を課しています。時間の経過と共に証明されたメカニズムは、SIM に基づいたネットワーク認証です。この認証インフラストラクチャには、認証だけでなく、事前共有秘密に基づく暗号化機能も提供するという長所があります。エンドポイント数の爆発的増加および IoT の世界的展開により、ネットワークローミングおよび無人のエンドポイントから SIM を物理的に削除できるセキュリティの弱さのために、SIM の使用が制限されています。埋め込み SIM のような技術の到来は、事前共有秘密に基づいた認証向けの実用的なインフラストラクチャを実現し、現在の SIM ベースのネットワーク認証を拡大しています。また、IoT の成長は、キャピラリーネットワークの形態で起こる可能性が最も高いです（本文書の前章における例 2、3 および 4 の構成で示した PAN）。これらのキャピラリーネットワークは、ゲートウェイに接続されたエンドポイントデバイスの大群です。これらのエンドポイントデバイスの大部分は軽量のエンドポイントデバイスとなります（つまり、SIM や移動体通信接続は含まれていません）。これらの軽量なエンドポイントデバイスは、それでもなお認証と暗号化機能を必要とします。キャピラリーネットワークにおける認証の主な責任はゲートウェイにあり、ネットワーク全体における複雑な SIM ベースのエンドポイントデバイス数を削減しています。この認証およびセキュリティをゲートウェイからエンドポイントデバイスに拡張し、特定のエンドポイントデバイスから IoT サービスプラットフォームにセキュリティ保護されたチャネルを作成する必要があります。

SIM ベースの認証は、単一のアプリケーション、つまりネットワーク接続のために一意のエンドポイントデバイスの認証の役割を果たすことを意味します。エンドポイントデバイスには、様々なサービスを備えており、それぞれ異なる認証と限定したニーズがあります。ネットワーク認証を複数のサービスに拡張するフレームワークが必

要です。このために設計されたフレームワークの 1 つが、汎用ブートストラッピング・アーキテクチャである GBA となります。GBA は、SIM ベースのインフラストラクチャを活用して、デバイスとネットワーク認証機能（NAF）間での時間ベースの共有キーを生成します。GBA は 3GPP 仕様 TS 33.220 [17]の 3GPP で標準化した認証方法です。この方法では、サービスに対する 3GPP サブスクリプションを備えたデバイスの認証を可能にします。サブスクリプションの資格情報はデバイス内にあり、通常は、UICC（Universal Integrated Circuit Card）などの SIM 上に格納されているか、リモートで管理する資格情報として、例えば GSMA specified Embedded SIM (eUICC) [5]などの埋め込み SIM（eUICC）に格納・管理されています。

このフレームワークの利点は以下の通りです。

- デバイスとネットワークアプリケーション機能間の独自の PSK、または承認書ベースの NAF 認証（TS 33.222）[18]による共有キーベースの UE 認証のいずれかに基づいた相互認証。
- 信頼できる環境で資格情報が保護される
- eUICC を使用すると、資格情報を OTA に変更できます。
- スケーラビリティ認証がフレームワークの「内部に組み込まれて」いるため、メンテナンスの複雑さと経済コストは、デバイス数に対して直線的に増加します。
- データ整合性認証時に生成された時間ベースのキーは、TLS-PSK トンネルを確立するために使用することができ、この接続は非常に強力なデータ整合性と機密性を提供することになります。

付録B　IoT サービスにおける UICC カードの使用に関するチュートリアル

ETSI TS 102 221 で標準化されている UICC は、相互運用可能なセキュアなファイルシステムインターフェイスとセキュアなアプリケーションフレームワークを UICC ホストデバイスに提供するスマートカードプラットフォーム（プログラミング可能な耐タンパーセキュアエレメント）です。ETSI TS 102 221 は、UICC ホストデバイスが UICC 上で関連するアプリケーションを発見するためのフレームワークを提供し、各 UICC アプリケーションは、既知のプロビジョニングと構成情報、および（認証やキーの導出などの）運用手順に対応し、必要に応じてホストデバイスでサポートできます。

IoT という観点から、UICC は、ETSI TS 102 671 に明記しているように複数のフォームファクターおよび環境動作範囲で利用できます。その最もシンプルな実施形態では、UICC は通常、ネットワーク事業者が所有しており、1 つのネットワークアクセスアプリケーション（3GPP TS 51.011 による SIM アプリケーション、3GPP TS 31.102 による USIM、3GPP2 で規定された CDMA CSIM,、WiMAX SIM など）のみをホストします。この場合、UICC は ETSI TS 102 225 / TS 102 226 を使用する UICC のコンテンツをリモートで管理するための追加のメカニズムを用いてネットワークアクセスを可能にするために、セキュリティのプロビジョニングと構成情報だけでなく、モバイルデバイス上の暗号化手順もホストする標準化されたホルダーを提供します。モバイルネットワークのエコシステムには、ネットワーク事業者の管理下で UICC の安全なパーソナル化と展開を確保するための手順があり、UICC ホストデバイスとインフラストラクチャ間に個別の共有対称キーを確立します。

UICC プラットフォームの重要な特徴の 1 つには、複雑なエコシステム内にいる複数のステークホルダーがそれぞれ UICC 上の独自の領域に割り当てられ、そのコンテンツを他のステークホルダーの機密性で管理できるようにする独立したセキュリティドメインのサポートがあります。この機能は、GlobalPlatform Card Specification [15] Amendment A の ETSI TS 102 226 で受け継がれています。従って、IoT の観点においては、単一の UICC により、複数のステークホルダーが互いに独立して独自の資格情報を格納および管理できます。

一般的に、UICC は、いくつかのネットワークアクセスアプリケーション（常時アクティブになっている 1 つのみ）と、IMS アクセス用の ISIM アプリケーション（3GPP TS 31.103 で定めています）のようなより精巧なサービスへのアクセスを保護する潜在的に他のアプリケーションを保持できます。IoT サービスの場合、oneM2M TS-0003 の付属書 D で定めている 1M2M SM アプリケーションを保持できます。1M2MSM アプリケーションは、専用の IoT サービス/アプリケーションの資格情報の直接プロビジョニングだけでなく、3GPP で定めている GBA メカニズムを使用する UICC の既存のネットワークアクセス資格情報からの導出をサポー

トできます。さらに、IoT サービス提供者が特定サービスの認証メカニズムをサポートするなど、特定のニーズに応じて暗号化手順をカスタマイズできるようにします。

単一の UICC は、複数の 1M2MSM アプリケーションを保持することもでき、各 IoT サービス提供者専用の対称キーを機密性高く展開できます。UICC の所有者（IoT の観点においては通常、ネットワーク事業者または OEM 製造者）は、UICC を要求する IoT サービス提供者と UICC のスペースを共有し、ネットワークアクセス資格情報の安全な展開を可能にする、認められた UICC パーソナル化チェーンおよびインフラストラクチャを IoT サービス提供者が独自の資格情報を展開するのに活用することもできます。

IoT アプリケーションセキュリティが非対称暗号に依存する場合、カスタム UICC を同様に使用して、特定の IoT サービスに必要な公開/秘密キーペアの展開を容易にすることができます。このような UICC アプリケーションは、IoT アプリケーション固有の基準でホストデバイスを指定し、サポートする必要があります。

付録C　文書管理

C.1　文書の履歴

バージョン	日付	変更事項の簡記	承認者	編集者/会社名
1.0	2016 年 2 月 8 日	New PRD CLP.13	PSMC	Ian Smith GSMA & Don A. Bailey Lab Mouse Security
1.1	2016 年 11 月 7 日	GSMA IoT セキュリティ評価スキームの参考資料の追加。 編集上の軽微な明確化。	PSMC	Ian Smith GSMA
2.0	2017 年 9 月 29 日	GSMA LPWA ネットワークリソースに加え、その他軽微な改訂を参考文献に追加します。	IoT Security Group	Rob Childs GSMA

C.2　その他の情報

種類	説明
文書の所有者	GSMA IoT プログラム

連絡先	Rob Childs – GSMA

GSMA は、お客様に高品質の情報をお届けしたいと考えています。誤記や記載漏れなど、お気づきの点がございましたら、ご意見をお寄せください。お問い合わせ先：prd@gsma.com

ご意見、ご提案、ご質問をお待ちしております。