IoTサービスのエコシステム

に関するIoTセキュリティ・ガイドライン

IoT サービスのエコシステムに関する IoT セキュリティ・ガイドライン

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 6

1.1 GSMA IoT セキュリティ・ガイドライン文書群 6

1.2 文書の目的 7

1.3 想定読者 8

1.4 用語の定義 8

1.5 略語 10

1.6 参考文献 10

2 サービスモデル 11

3 セキュリティモデル 14

3.1 ネットワークインフラストラクチャ攻撃 16

3.2 クラウドまたはコンテナインフラストラクチャ攻撃 18

3.3 アプリケーションサービス攻撃 20

3.4 プライバシー 20

3.5 悪意のあるオブジェクト 20

3.6 認証と承認 21

3.7 フォールスポジティブとフォールスネガティブ 22

4 セキュリティに関するよくある質問 22

4.1 クローニングにどう立ち向かいますか。 22

4.2 エンドポイントを介してユーザーが認証を受ける仕組みは。 23

4.3 サービスは匿名のエンドポイント動作をどのように特定しますか。 24

4.4 サービスはエンドポイントの異常動作をどのように制限しますか。 25

4.5 サーバーやサービスがハッキングされたかどうかを判断するにはどうすればよいですか。 25

4.6 サーバーがハッキングされた場合はどうすればよいですか。 26

4.7 管理者はサーバーやサービスとどのように通信を行う必要がありますか。 26

4.8 サービスアーキテクチャは、セキュリティ侵害の影響をどのように抑えることができますか。 27

4.9 サービスアーキテクチャは、セキュリティ侵害によるデータ損失をどのように軽減することができますか。 28

4.10 サービスアーキテクチャは、権限のないユーザーによる接続をどのように制限できますか。 29

4.11 リモートからの攻撃の可能性を軽減するにはどうすればよいですか。 29
4.12 サービスはユーザーのプライバシーをどのように管理することができますか。 30
4.13 サービスはその可用性をどのように向上できますか。 30
5 重要な推奨事項 32
5.1 サービスのトラステッド・コンピューティング・ベース（TCB）の実行 32
5.2 組織の信頼の基点（Root of Trust）の定義 33
5.3 ブートストラップ法の定義 35
5.4 パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義 36
5.5 永続ストレージモデルの定義 38
5.6 管理モデルの定義 39
5.7 システムロギングとモニタリング手法の定義 40
5.8 インシデント対策モデルの定義 41
5.9 復旧モデルの定義 42
5.10 サンセットモデルの定義 43
5.11 セキュリティ区分の定義 44
5.12 データタイプ区分の定義 45
6 高優先度の推奨事項 47
6.1 明確な承認モデルの定義 47
6.2 暗号化アーキテクチャの管理 47
6.3 通信モデルの定義 49
6.4 ネットワーク認証サービスの使用 51
6.5 可能であればサーバープロビジョニング 52
6.6 更新モデルの定義 53
6.7 流出データに対する違反ポリシーの定義 54
6.8 サービスエコシステムによる強制認証 55
6.9 入力検証の実行 56
6.10 出力フィルタリングの実行 57
6.11 強力なパスワードポリシーの施行 58
6.12 アプリケーションレイヤの認証と承認の定義 61
6.13 デフォルトオープンまたはフェイルオープンのファイアウォールのルールとシステム強化 62

6.14 通信プライバシーモデルの評価 63

7 中優先度の推奨事項 65

7.1 アプリケーション実行環境の定義 65

7.2 パートナーの強化モニタリングサービスの利用 66

7.3 セルラー接続に対するプライベート APN の利用 67

7.4 サードパーティーのデータ配布ポリシーの定義 68

7.5 サードパーティーのデータフィルターの構築 69

8 低優先度の推奨事項 71

8.1 Rowhammer や類似の攻撃 71

8.2 仮想マシンのセキュリティ侵害 72

8.3 ユーザーがプライバシー属性を管理するための API の構築 72

8.4 フォールスネガティブとフォールスポジティブの評価モデルの定義 73

9 要約 75

付録 A 文書管理 76

文書の履歴 76

その他の情報 76

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。この一連の拘束力のない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・ガイドライン概要説明書」[1]を手引きとして参照した後に、その他の関係文書へと読み進めることをお勧めします。

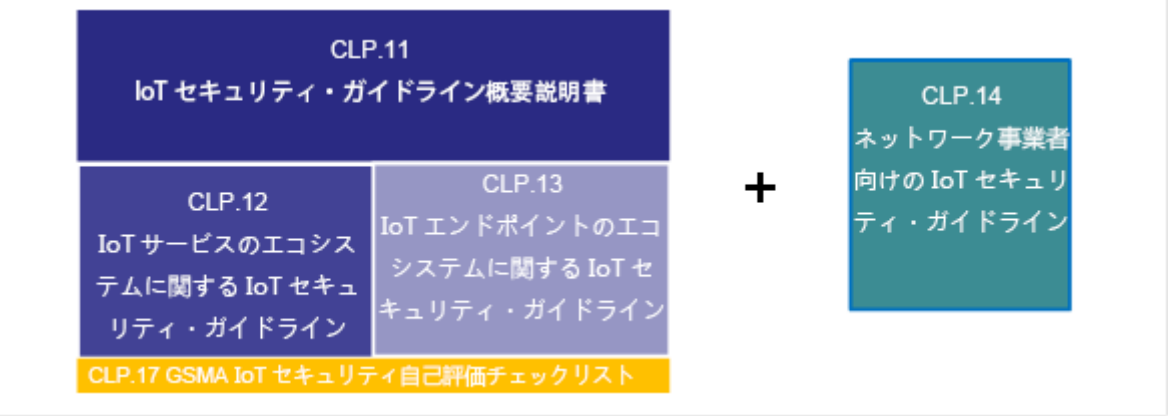

図1-「GSMA IoT セキュリティ・ガイドライン」文書の構成

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 [13]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」[13]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 製品またはサービスにおけるすべてのコンポーネントを、サービスエコシステムの観点から検証するために使用することを目的としています。サービスエコシステムには、IoT インフラストラクチャの中核を成すあらゆるコンポーネントが含まれています。このエコシステムのコンポーネントの一例には、サービス、サーバー、データベースクラスタ、ネットワーク要素など、製品やサービスの内部コンポーネントの動作に使用される技術があります。

本文書の適用範囲は、IoT サービスとネットワーク要素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 エンドポイントデバイス製造業者 - IoT サービス提供者向けに IoT サービス対応の IoT エンドポイント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
アクセスコントロールリスト	計算オブジェクトに付随する権限のリスト。
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。
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。
コンテナ	複数の独立したシステムやコンテナを 1 つのホストで実行できるようにするための技術。
埋め込み UICC（eUICC）	GSMA の指定に従い、認証するネットワークやサービス契約のリモート・プロビジョニングに対応する UICC。
最終顧客	IoT サービス提供者によって提供された IoT サービスを利用する消費者。公益事業会社のように、最終顧客と IoT サービス提供者が同じである場合があります。
エンドポイントのエコシステム	斬新な方法で現実世界をデジタル世界につなぐ、複雑性の低いデバイス、リッチデバイス、ゲートウェイからなる構成。詳細については、CLP.11 [1]を参照。
前方秘匿性（Forward Secrecy）	安全通信プロトコルの属性。安全通信プロトコルは、長期鍵の安全性が破れたとしても過去のセッションキーの安全性が保たれる場合、前方秘匿性があるとみなされます。

用語	説明
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 エンドポイント	複雑な IoT エンドポイントデバイスや IoT ゲートウェイデバイスの総称。
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。
セキュリティグループ	1 つ以上のインスタンスのトラフィックを制御する仮想ファイアウォール機能。
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。
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。

用語	説明
バーチャルプライベートネットワーク	特定のサービスセットが利用できる、隔離された安全で仮想的な専用ネットワーク。他のネットワークから隔離されているプライベートな空間であることから「VPN」と呼ばれており、仮想化されたネットワークとして機能します。

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
ACL	アクセスコントロールリスト
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
CERTS	コンピューター緊急対処チーム
CLP	GSMA のコネクテッド・リビング・プログラム
DDoS	分散型サービス拒否攻撃
GSMA	GSM Association
HSM	ハードウェアセキュリティーモジュール
IoT	モノのインターネット
IP	インターネットプロトコル
SQL	構造化問い合わせ言語
TCB	トラステッドコンピューティングベース
VM	仮想マシン
VPN	バーチャルプライベートネットワーク
WAF	ウェブアプリケーションファイアウォール

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

参照	文書番号	タイトル
[5]	該当なし	OWASP Secure Application Design Project https://www.owasp.org
[6]	該当なし	TCG Trusted Platform Module http://www.trustedcomputinggroup.org
[7]	該当なし	TCG Guidance for Securing IoT http://www.trustedcomputinggroup.org
[8]	該当なし	OAuth 2.0 http://oauth.net/2/
[9]		OpenID Foundation http://openid.net/foundation/
[10]	該当なし	GSMA Mobile Connect https://mobileconnect.io/
[11]	GPC_SPE_034	GlobalPlatform Card Specification www.globalplatform.org/specificationscard.asp
[12]	GPD_SPE_010	GlobalPlatform TEE Internal Core API Specification www.globalplatform.org/specificationsdevice.asp
[13]	CLP.17	GSMA IoT Security Assessment Checklist https://www.gsma.com/iot/iot-security-assessment/
[14]	該当なし	ETSI TC SmartM2M specifications www.etsi.org
[15]	該当なし	oneM2M Specifications www.onem2m.org
[16]	3GPP TS 33.220	Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) www.3gpp.org

2 サービスモデル

現代の IoT 製品やサービスには、エンドポイント、パートナー、ユーザーに意義、機能、価値を提供するサービスエコシステムが必要です。インフラストラクチャは、IoT サービスが提供するアプリケーションの複雑性に応じて異なり、様々な種類のサービスとサービスアクセスポイントから構成され得ます。より単純なアプリケーションの場合は、インフラストラクチャは基本的なものとなるでしょう。

サービスエコシステムは、そのフォーマットにかかわらず、IoT 技術全体の各コアファセットに対する機能と通信の中心的な役割を果たします。他のあらゆるエコシステムにおける階層的認証、ユーザーの接続性、可用性、管理など、IoT に関する日常業務に欠かせないタスクは、サービスエコシステムに依存します。これらのタスクを遂行するために、サービスエコシステムはインフラストラクチャの目標達成に必要な多くの階層から構成

されています。データベースクラスタ、アプリケーションサーバー、アプリケーションプロキシサーバー、その他のインフラストラクチャは、多くの展開で見られる階層の例です。下の図からわかるように、ネットワークとエンドポイントエコシステムは、サービスエコシステムのコア機能に依存します。

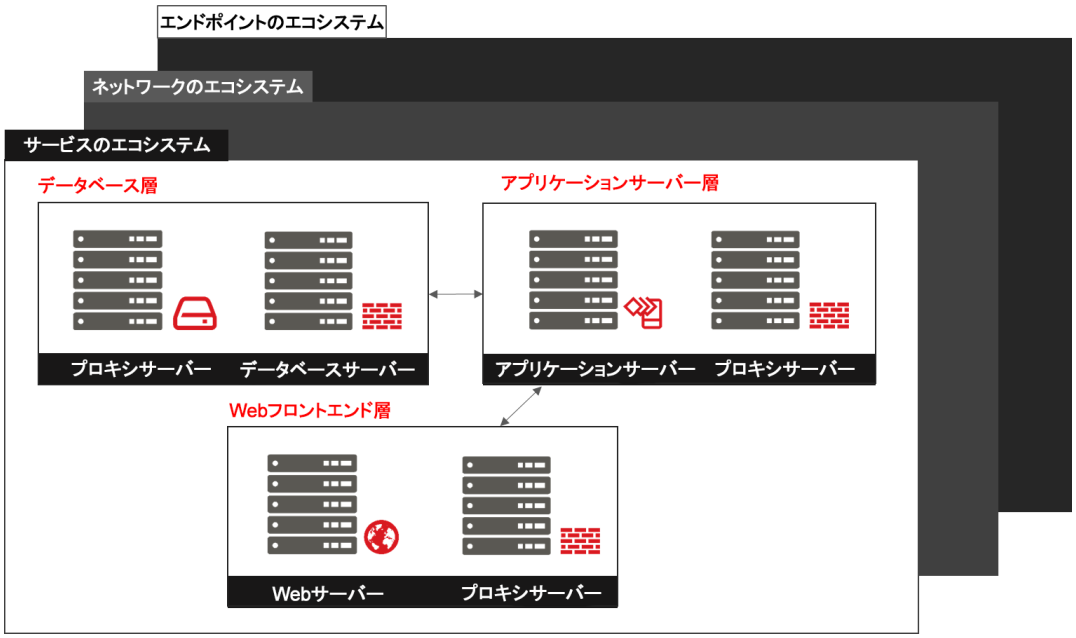

図2-サービスエコシステムに基づく依存関係

現代のサービスエコシステムの例には以下のものが含まれますが、これらに限定されません。

- クラウドインフラストラクチャベースのソリューション
- コンテナベースのアプリケーション展開
- 従来のデータセンターサーバー環境
- データベースクラスタ
- ウェブアプリケーションフレームワークのサービスクラスタ

上記の環境はそれぞれの設計、トポロジー、実装によって大きく異なるように見えますが、これらはすべて、アプリケーション内外の情報フローに関する同じ理論に基づいています。

現代のすべてのコンピューターシステムには、サービスアクセスポイントと呼ばれる、アプリケーションのインフラストラクチャへの入口が必要です。アプリケーションのコンテンツやコンテキストを作成する内部サブシステムは、信頼できる安全な環境やネットワーク内でデータを処理できる必要があります。データは任意の場所に保存

された後、サービスレイヤに返されます。サービスレイヤは、同じエコシステムか、その他のエコシステムや関連するネットワーク内で、認証されたコマンドを様々なコンポーネントに送信します。

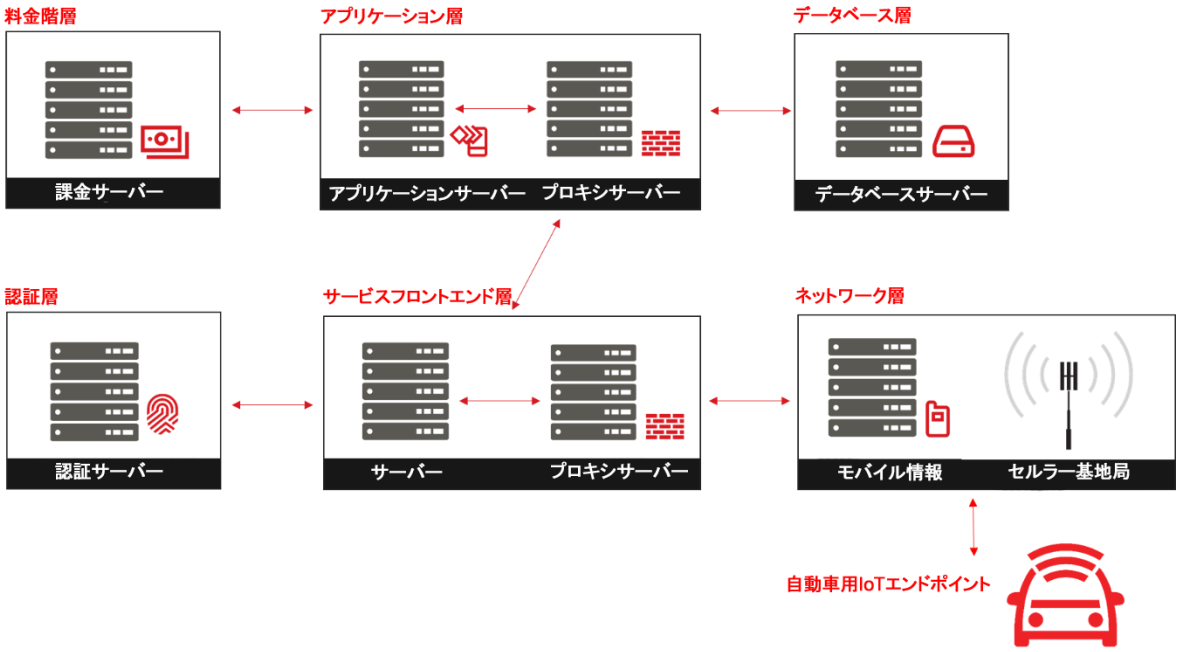

図3-サンプルサービスエコシステム

最新型であれ旧式であれ、この標準フレームワークを実装するために使用する技術の種類にかかわらず、情報は実証済みのプロトコルや技術によって処理、伝送および認証されます。処理環境に必要なトポロジーやアブストラクションは、速度、計算力、ストレージなどの最新の要件に応じて変化していますが、これらのイノベーションを実装するために使用する技術は、根本的には変わりません。例えば、各階層には、特定の種類のサーバー群に対する接続を管理するプロキシやファイアウォールが含まれています。請求層には請求サービスがあり、アプリケーション層にはアプリケーションサーバーがあります。データベースサービスは、データベース層で管理する必要があります。これらのシステムはすべて、プロキシサーバーに適用される入口・出口ルールに基づいて機能します。

そのため、サービスエコシステムのセキュリティモデルは、複数のコンポーネント群に分けることができます。本文書では、これらのコンポーネントについて説明します。

3 セキュリティモデル

サービスエンドポイント環境のセキュリティは、アプリケーションアーキテクチャの構築に使用されるトポロジーやイノベーションの種類にかかわらず、共通のインフラストラクチャ、戦略、ポリシーによって設計することができます。サービスエコシステムは複数のコンポーネントに分けることができ、これらのコンポーネントは個々にセキュリティを確保する必要がありますが、同じ手法を利用できます。

例えば、エンドポイント、パートナー、ユーザーに対してクエリの処理や応答の送信を行う、シンプルなサービスを構築するためのコンポーネントを考えてみましょう。このモデルには、以下の階層を含める必要がありますが、これらに限定されません。

- ウェブサービス層
- アプリケーションサーバー層
- データベース層
- 認証層
- ネットワーク層
- 請求層などのサードパーティー・アプリケーション層

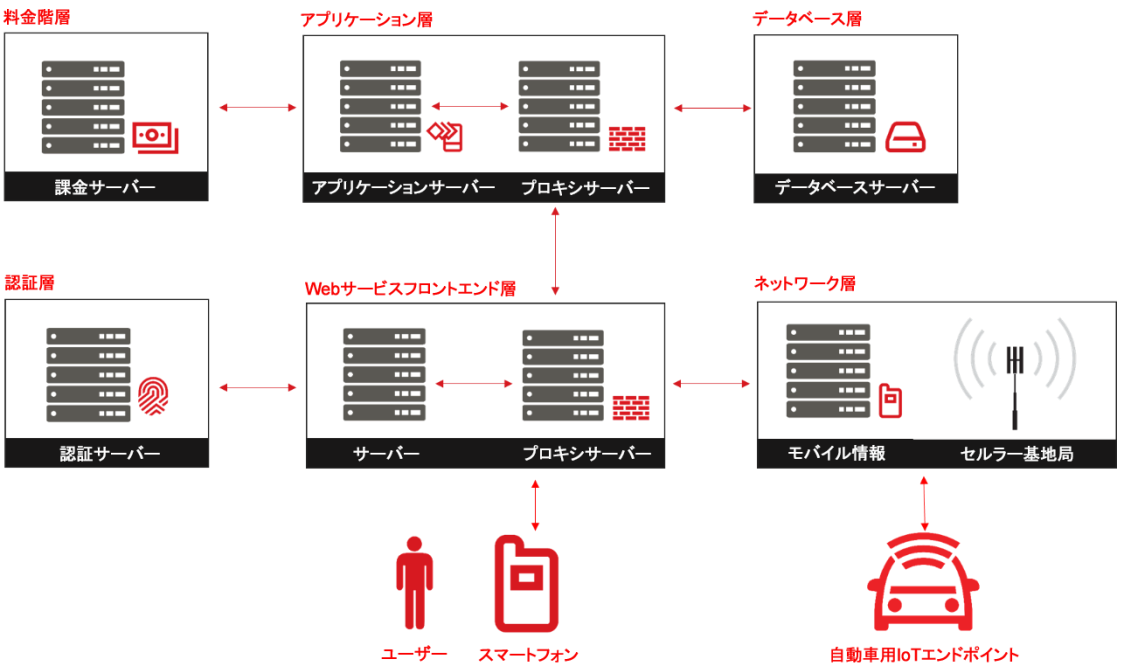

図4- 階層を区分したサービスエコシステムの例

各階層には 1 つのサーバーしかありませんが、論理的概念を該当する階層に割り当てる方が構造上より効果的です。また、セキュリティ侵害が発生した場合や、より多くのリクエストを処理するためにシステムを拡張する必要がある場合は、技術レイヤを他のレイヤから隔離します。

システムの種類が*階層式*であると思われる場合、セキュリティ保護、オンデマンド拡張、デコミッション処理、サンセット措置を容易に行うことができます。唯一の要件は、API が、階層の存続期間中に拡張または調整できるほど汎用性が高いことです。本文書では、API の定義については取り扱いませんが、組織が採用または定義するべき API の高いセキュリティ属性について、推奨事項を説明します。

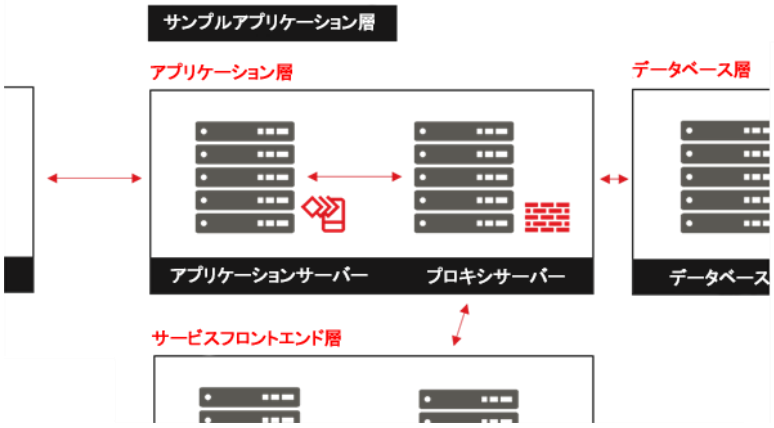

図5- ファイアウォールによって保護されているアプリケーション層

上記の例では、階層がより詳しく説明されています。階層を説明する上で不可欠なものは、プロキシサーバーです。このプロキシサーバーは、階層内で実際に展開されるセキュリティ技術を表す記述子にすぎません。実際の制御がハードウェアファイアウォール、ソフトウェアファイアウォール、セキュリティグループ、アクセスコントロールリスト（ACL）、その他のテクノロジーかどうかにかかわらず、階層の代わりに入力と出力を制御するコンポーネントがあります。

API を選択または定義する場合、組織は、技術チームの懸念を解決するための既存の仕様を検討する必要があります。特に、組織は以下の仕様を考慮するべきです。

- ETSI SmartM2M TS 102 690、ETSI SmartM2M TS 102 921 [14]
- oneM2M TS-0001、oneM2M TS-0003 [15]
- 3GPP TS 33.220 [16]

サービスフロントエンド層などの一般に公開されているコンポーネントについては、モデルが強化すべき点は以下の項目に対するセキュリティコンポーネントです。

- 分散型サービス拒否攻撃（DDoS）保護
- ロードバランシング（負荷分散）
- 冗長性
- オプションのウェブアプリケーションファイアウォール（WAF）機能

上記の技術を実装することで、サービスが適切に機能するとともに、リソースが極めて限られている状況においてもサービスが利用できるようにする必要があります。これらのコンポーネントの定義については、このドキュメントの範囲外ですが、詳細については以下のガイダンスや規格を参照してください。

- Cloud Security Alliance
- NIST Cloud Computing Standards
- FedRAMP
- Cisco Network Management Guidelines

この層が安全に機能するために必要な他の属性は、サーバー自体の定義です。これは、技術チームが選択したプラットフォーム内の管理者、アプリケーション、オペレーティングシステム制御によって定義されます。

一例にすぎませんが、プラットフォーム環境における主な問題は以下のとおりです。

- 集中ログサービスへのログ記録
- 管理者認証と承認
- 通信セキュリティの確保
- データのバックアップ、復元、複製
- アプリケーション業務の区分
- システムのモニタリングと完全性（インテグリティ）

3.1 ネットワークインフラストラクチャ攻撃

ネットワークの観点からサービスエンドポイントのセキュリティを侵害しようとす攻撃者は、組織の通信方法や、サービスのアクセスポイントを介して公開されているサービスに脆弱性があることを想定しています。また、これらの攻撃は、ネットワーク上の特権的な立場は、通信チャネルを管理できる立場であると想定しています。

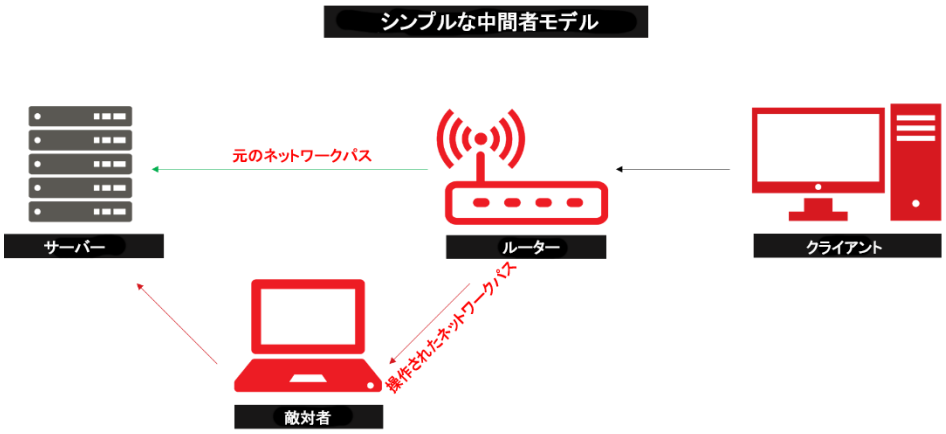

図6-「中間者」攻撃モデルの例

このモデルにおける最も一般的な攻撃形態は、中間者（MITM）攻撃です。この攻撃は、ピア認証または片方のピア認証がないか、通信チャネル上の相互認証データが破損していることを前提としています。攻撃者の目的は、1 つのホストになりすまして、攻撃者の代わりにもう 1 つのホストにアクションを強制的に実行させることです。この攻撃は、相互認証を実施することで緩和できます。相互認証には、適切に定義された組織の信頼の基点（Root of Trust）、トラステッド・コンピューティング・ベース（TCB）、通信モデルが必要です。

その他の攻撃には、前方秘匿性（Forward Secrecy）に対する攻撃、暗号化通信分析、サイドチャネル攻撃などがあります。これらの攻撃を緩和するには、適切な暗号化プロトコルやアルゴリズム、規格を使用する必要があります。

これらの攻撃への対応は容易ではなく、組織内、組織とそのパートナー間の中核となるインターネットインフラストラクチャ内、またはエンドポイントのエコシステム内のネットワークインフラストラクチャか、エンドポイント付近のインフラストラクチャへのアクセスを必要とします。最もシンプルかつ一般的な攻撃は、Wi-Fi、イーサネットまたは携帯電話のネットワークなどのエンドポイントのネットワークインフラストラクチャを操作して、サービスとそのピアとの間にある特権的な立場を獲得しようとすることです。

1 つのエンドポイントのインフラストラクチャに対する攻撃は、そのエンドポイント、または物理的に利用できるエンドポイントグループに限定されます。中核となるインターネットインフラストラクチャに対する攻撃には、通常ボーダーゲートウェイプロトコル（BGP）のハイジャック、コアルーターへの攻撃、またはドメインネームサービス（DNS）のインフラストラクチャの悪用を伴います。これらの攻撃により、特定のターゲットとは関係のない特権的な立場を獲得でき、攻撃者が 1 度に多数のシステムへのアクセス権を持つ恐れがあります。内部ネ

ットワークインフラストラクチャに対する攻撃を行うには、内部ネットワークにアクセスする必要があります。つまり、インサイダー攻撃や、企業内の既存の特権的な立場を獲得する必要があり、企業内のシステムはすでに深く侵害されている可能性があります。

攻撃の種類にかかわらず、この攻撃モデルは相互認証、前方秘匿性（Forward Secrecy）、および適切な暗号化プロトコルとアルゴリズムを使用して容易に緩和できます。これにより、攻撃者がこのインフラストラクチャを悪用する能力を無効にするか、このタイプの攻撃にかかるコストを大幅に引き上げることで、一般的な攻撃者が攻撃を実施できないようにします。

3.2 クラウドまたはコンテナインフラストラクチャ攻撃

これらの攻撃は、クラウドまたはコンテナインフラ環境における特権的な立場を想定しています。例えば、攻撃者がクラウドサービスのネットワークに侵入できる場合、ゲスト仮想マシン（VM）のシステムを実行しているホストにアクセスしている場合があります。これにより、攻撃者は VM のシステムを調べたり、変更することができます。攻撃者は具体的な目的があるか、有益なデータがある様々な種類のシステムにアクセスするためだけに、運よくクラウドサービス提供者のセキュリティを侵害できた可能性があります。

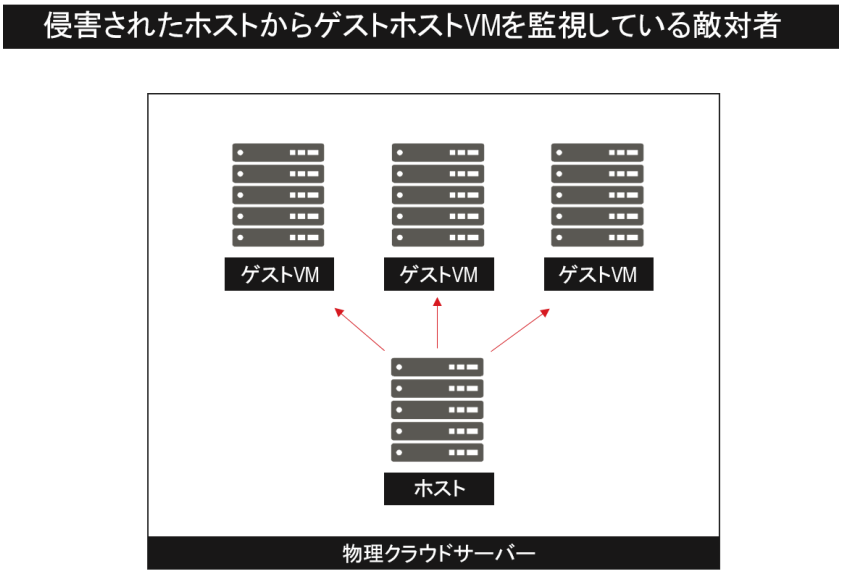

図7- VM 攻撃モデルの例

別のクラウドまたはコンテナのインフラストラクチャの攻撃は、攻撃者が同一の物理サーバー上で、ターゲットとして VM を制御することを前提としています。攻撃者は複数の手法を用いて、物理サーバー上の他の VM のセキュリティを侵害する場合があります。具体的には、以下のような攻撃を行う可能性があります。

- VM インフラストラクチャの脆弱性を利用して、ホストシステムにゲストを侵入させる
- サイドチャネル攻撃を利用して、別のゲスト VM から秘密鍵を推測する
- 物理サーバー上でリソースを過剰消費し、ターゲットの VM を、攻撃者が高いレベルの権限を持っている物理サーバーに移行させる

利用される攻撃モデルにかかわらず、企業はこのようなリスクを防ぐためにできることはほとんどありません。代わりに、クラウドサービス提供者は、攻撃者がクラウドまたはコンテナのインフラストラクチャを攻撃できる確率を減らすために、十分な機能を実装する必要があります。

このリスクを軽減する方法の 1 つは、各コンテナを特定のユーザーと一意の暗号化 ID に制限する、コンテナベースのアーキテクチャを実装することです。これは非常に多くのリソースを消費するアクティビティであり、追加のコストが発生する可能性がありますが、攻撃者が複数のユーザーまたはサービスに 1 度にアクセスするために、VM のインフラストラクチャを悪用する能力を軽減します。

クラウドまたはコンテナ環境での特権的な立場は、ゲスト VM 内で実行中のアプリケーションにとって大きな脅威ではあるものの、この立場にアクセスするには、高レベルの技術だけでなく、多くの時間とリソースが必要です。攻撃者はアクセス権を取得した後、ターゲットの VM が含まれているシステムを識別するまでこのアクセス権を維持する必要があります。また、攻撃者は、クラウドサービス提供者のインシデントのサブシステムによって検出されることなく、VM を監視または変更できる必要があります。これは攻撃者にとって非常に困難であり、セキュリティ侵害のリスク軽減につながるはずです。

しかし、ここで重要なことは、このセキュリティ侵害は、ゲスト仮想マシンまたはそこで実行されているアプリケーションによってほとんど検出されないことです。そのため、収集されたメトリクスによって、特定のクラウド VM またはコンテナの異常な動作が検出されたとしても、セキュリティ侵害が実際に発生したかどうかを識別することは非常に困難な場合があります。VM インフラストラクチャのホストレイヤに十分な権限を持つ攻撃者は、不正操作の検出を困難にするためにゲストを操作できるからです。

クラウドサービス提供者でさえも、ゲストからゲストへの攻撃を検出することは極めて困難です。しかし、これらの攻撃が可能であることは、あくまで理論上の話です。サイドチャネル攻撃は理論上は可能ですが、実行できるかどうかは賛否両論があります。これらの攻撃を実行するには基盤となる実行プラットフォームでの一貫性が求められますが、実世界ではこの一貫性は保証されていないからです。また、VM、コンテナまたはハイパーバイザー環境におけるゲストからホストへの権限昇格攻撃を検出するのは難しいですが、悪用することはさらに困難です。そのため、脆弱性によって大勢のゲストや特定のターゲットの悪用が発生する可能性が低くなります。

このように、特権的な立場は攻撃者にとって重要である一方、特権的な立場を利用した攻撃の困難さ、コスト、実行がほぼ不可能であるという事実により、攻撃が成功する可能性を軽減しています。

3.3 アプリケーションサービス攻撃

アプリケーションの実行アーキテクチャは本文書の対象外ですが、このレイヤは攻撃のリスクが高いため、注意が必要です。サービスのエコシステムが、本ガイドで推奨されているように正しく設定されている場合、攻撃者は攻撃対象をネットワークインフラストラクチャからアプリケーションに変更します。

製品やサービスのアプリケーションは最大かつ複雑なレイヤであり、攻撃者が複数の技術を用いて自分の権限を昇格するリスクを常に伴います。そのため、本文書の目的はネットワークインフラストラクチャへの攻撃回避について注意喚起を行うことだけでなく、攻撃が成功する可能性が高い*場所*にも注意を促すことにあります。

攻撃のリスクを減らすには、アプリケーションのセキュリティに関する参考文献（OWASP Secure Application Design Project [5]など）を参照し、アプリケーションの実行アーキテクチャを可能な限り安全に実装してください。

3.4 プライバシー

パートナーシステムはデータ/メトリクスまたは他のユーザーを中心としたコンポーネントを消費し、システム全体に付加価値を提供するように設計されていますが、パートナーが実装するセキュリティレベルが保証されるわけではありません。情報を第三者に単に渡すのではなく、どのような種類のデータを渡すべきか、具体的なメリットは何か、情報をどのように保護するかを評価する必要があります。

契約や保険約款によって法的責任は軽減され得ますが、第三者の失態によって顧客喪失が引き起こされる可能性があります。このような事業損失のリスクを避けるために、組織は、第三者の技術チームの評価を行い、インフラストラクチャ、アプリケーションおよび API に適用されるセキュリティレベルを特定する必要があります。セキュリティレベルが十分でない場合、別のパートナーを探すことをお勧めします。

3.5 悪意のあるオブジェクト

サードパーティー・システムは、消費者の個人情報またはマルチメディアを提示するよう設計されています。これを実現する 1 つの明白な方法は、広告です。広告は様々な種類のファイルによる複雑な構成となっており、ソフトウェアによって適切に解析することは困難です。広告ネットワークは、マルウェア配布に有効なチャネルです。コンテンツ配信ネットワーク（CDN）もまた、マルウェア配布に利用される可能性が高いチャネルで

す。動的情報のレンダリングのために複雑なマルチメディアやコードバンドル（ウェブまたは実行可能ファイル）を提供するシステムは、マルウェアを提供する恐れがあります。

そのため、企業は特定のチャネルを介して渡される技術サービスの種類を評価することが不可欠となります。また、顧客への提供を許可または拒否するサービスを決定する必要があります。例えば、広告会社は、IoT 企業によってパートナーに提供されたプロキシサービスアプリケーションを介して、Java コードをクライアントシステムに提供する場合があります。広告会社は、特定の環境で実行しているクライアントシステムが、Java テクノロジーからの攻撃を受けやすいかどうかを判断する必要があります。攻撃を受けやすいことが判明した場合、広告会社は Java を許可しないようにする必要がありますが、ハイパーテキストマークアップ言語（HTML）などの他の技術によって許可する場合があります。

多様なファイル、Adobe Flash、Java、マルチメディアの悪用など、マルウェアは様々な形式を取るため、エンドユーザーの安全を保証するための統一された唯一の方法はありません。技術チームが利用できる簡単なソリューションは、チャネル上で使用*される*テクノロジーとユーザーへの影響に関するポリシーを施行*する*ことです。監視サブシステムやサンドボックスを導入することで、クライアントシステム上でレンダリングされるオブジェクトが悪用される可能性を軽減できます。

3.6 認証と承認

パートナーは、ユーザーのサブセット専用のサービスを提供することが多いです。これには、ユーザーが任意で登録できる有料サービスも含まれる場合があります。つまり、ユーザーは、ネットワークサービス提供者からの既存の認証 API、ソーシャルネットワークインフラストラクチャ、既存の M2M または IoT 管理エンティティをはじめとした、よく知られている個別の技術によって共有される資格情報を使用して、システム認証を行うことができます。

これらのサービスはプラットフォーム間で技術を共有する優れた方法ですが、技術者は、サードパーティー・サービスに明示的に付与されていない権限の悪用に利用できる資格情報を、技術が意図せず使用しないようにする必要があります。例えば、特定のプラットフォームの API は、ユーザーから許可または拒否されているクラスへの権限を制限することができます。これにより、ユーザーのプライバシーのニーズに応じてユーザーエクスペリエンスを調整できます。プラットフォームが細分化されたセキュリティ権限を提供できない場合、アクセスを希望*する*テクノロジーを一覧表示する必要があります。

技術チームはパートナーに対して、サービス上で細分化された権限を提供することで、サービス契約が終了した後もユーザーのデータが流出することを意図せず可能にしないように依頼する必要があります。

3.7 フォールスポジティブとフォールスネガティブ

監視およびログのサービスは、既存のセキュリティインフラストラクチャを強化するための優れた方法ですが、フォールスポジティブとフォールスネガティブについて慎重に検証を行わなければなりません。これらのシステムは、IoT 製品またはサービス内の様々なエコシステムから発信されたデータを認識するだけであるため、社内の技術チームによって開発されません。また、イベントに人為的な情報のみを提供できます。しかし、悪意のあるイベントが実際に発生しているかどうかを正確に判断することができない場合があります。

そのため、IT および技術チームは、問題のあるイベントが悪意のある動作に起因するかどうかを検証することが重要です。これは、モニタリングチームが、正当な権限を持つユーザーによるシステムへのアクセスを拒否するリスクを軽減するのに役立ちます。このプロセスが自動的に行われ、かつプロセスが不適切な場合、クライアントアプリケーションやインフラストラクチャの異常によって引き起こされるフォールスポジティブにより、多くのユーザーが正当なアクセス権限があるサービスを利用できなくなる恐れがあります。重大なイベントが発生していることが疑われる場合、IT および技術チームはデータを確認し、攻撃が実際に発生しているかどうかを評価する必要があります。

また、技術者はアナログチャネルを介して取得されたモデル情報に注意する必要があります。取得したデータが完全に信頼できない場合の最も安全な措置をアプリケーションが適切に評価していない場合、データが高速で処理される必要があるエコシステムでフォールスポジティブとフォールスネガティブが発生すると重大な影響を及ぼす恐れがあります。十分な時間、技術および専門知識がある場合、すべてのアナログデータはデジタルシステムに偽装される可能性があります。

4 セキュリティに関するよくある質問

本文書には、優先順位別にサービスのセキュリティに関する推奨事項が記載されています。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。

4.1 クローニングにどう立ち向かいますか。

IoT サービス提供者によって製造された有効なデバイスと、複製または「模造品」（クローン）であるデバイスを見分けることは容易ではありません。CPU 時間、帯域幅、ディスクストレージなどのリソースに対してコストがかかっているため、不正なエンドポイントに対してサービスを提供したいと考えている IoT サービス提供者は

いません。デバイスが IoT サービス提供者によって製造されたものか否かにかかわらず、組織は支払いを行う必要があります。

また、組織は、エンドポイントのアーキテクチャが攻撃を受けているかどうかを見分けることができる必要があります。これにより、組織は同じデバイスの複数のインスタンスに*クローンされた*デバイスに対応することができます。これは、悪徳業者や、特定のユーザーになりすまそうとしている攻撃者によって行われる可能性があります。

サービスを利用してクローンに対応する場合、以下の推奨事項を確認してください。

- 組織の信頼の基点（Root of Trust）の定義
- ネットワーク認証サービスの使用
- サービスエコシステムによる強制認証
- アプリケーションレイヤの認証と承認の定義

4.2 エンドポイントを介してユーザーが認証を受ける仕組みは。

IoT で最も重要な概念の 1 つは、ユーザー認証からエンドポイント認証を分離することです。エンドポイントはトラステッド・コンピューティング・ベースによって認証できますが、*ユーザー*の認証方法はそれとは別に行われ、通信セキュリティを確保するためのエンドポイントの TCB に依存しています。ここで最も重要なことは、ユーザー認証に使用される通信チャネルが信頼できるかどうかを検証することです。

例えば、エンドポイント TCB がない、または弱いエンドポイント TCB が実装されているため、エンドポイントの信頼性が低い場合、エンドポイントのソフトウェア/ファームウェアに依存するユーザー認証メカニズムは信頼できません。つまり、エンドポイントデバイスによるユーザー認証は、認証されたとみなすことができません。

別の観点から見ると、認証スキームを容易にバイパスできる場合、適切に設計されているエンドポイント TCB であっても、エンドユーザーを適切に認証できません。そのため、サービスのエコシステムはエンドポイントの信頼性と認証メカニズムの実装に依存することで、正当な権限を持つユーザーがシステムにログインしているかどうかについて保証できる必要があります。

これらの問題に対応する場合は、以下の推奨事項を考慮してください。

- サービスのトラステッド・コンピューティング・ベース（TCB）の実行
- 組織の信頼の基点（Root of Trust）の定義

- 明確な承認モデルの定義
- ネットワーク認証サービスの使用
- サービスエコシステムによる強制認証
- 強力なパスワードポリシーの施行
- アプリケーションレイヤの認証と承認の定義

4.3 サービスは匿名のエンドポイント動作をどのように特定しますか。

分散型 IoT ネットワークにおけるエンドポイントの管理に関する最も難しい問題の 1 つは、エンドポイントが異常な方法で動作しているかどうかを特定することです。これはセキュリティの観点からだけでなく、信頼性の観点から見ても重要です。多くの場合、異常な動作はファームウェアやハードウェアの問題によって引き起こされる可能性があり、組織は予期しない問題を修正するための準備が必要です。しかし、異常な動作がネットワークから分離されており、IoT サービス提供者が分析できない場合、これらのメトリクスは失われ、組織にはメリットがほとんどなくなります。

この問題を解決するためには、エンドポイント、ネットワークレイヤおよびサービスのエコシステム上における動作を検査する必要があります。しかし、これらのデータポイントを収集するために適切なインフラストラクチャやサービス、パートナーシップが構築されていない場合、組織は問題があるか、また問題がセキュリティや信頼性に関連しているかどうかを決定するために必要な情報を入手できません。

サービスのエコシステムの観点から、以下の推奨事項を検証してください。

- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- システムロギングとモニタリング手法の定義
- 通信モデルの定義
- ネットワーク認証サービスの使用
- 入力検証の実行
- 出力フィルタリングの実行
- パートナーの強化モニタリングサービスの利用
- ワイヤレス接続に対するプライベート APN の利用
- フォールスネガティブとフォールスポジティブの評価モデルの定義

4.4 サービスはエンドポイントの異常動作をどのように制限しますか。

エンドポイントの異常動作が検出された場合、サービスはどのリソースを制限すべきか決定する必要があります。この質問は、サービスインフラストラクチャのすべてのレイヤに関連しています。

例えば、モバイルネットワークへの接続と切断を無限に繰り返すセルラー対応のエンドポイントは、異常動作が解決されるまで強制的に無効にする必要があります。もう 1 つの例は、攻撃者がバックエンドサービスに攻撃するために利用する、セキュリティ侵害されたエンドポイントです。この状況では、バックエンドサービスは、悪用されているエンドポイントがサービスにアクセスできないようにする必要があります。

それぞれの状況の対処方法は、IoT サービス提供者とその事業目標、インシデントへの対応方法に応じて異なります。これらのガイドラインを開発する場合は、以下の推奨事項を考慮してください。

- 組織の信頼の基点（Root of Trust）の定義
- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- インシデント対策モデルの定義
- 復旧モデルの定義
- サンセットモデルの定義
- 通信モデルの定義
- 流出データに対する違反ポリシーの定義
- サービスエコシステムによる強制認証
- ワイヤレス接続に対するプライベート APN の利用
- フォールスネガティブとフォールスポジティブの評価モデルの定義

4.5 サーバーやサービスがハッキングされたかどうかを判断するにはどうすればよいですか。

エンドポイントの異常は対応が困難であり、大部分の攻撃を特定するためには多くの動作分析を行う必要がありますが、サービスのエコシステムの場合は比較的容易に対応できます。サービスとサーバーは、クラウドやサーバーインフラストラクチャを管理する IoT サービス提供者やそのパートナーが厳正に管理する環境で展開されています。そのため、組織とそのパートナーは、利用可能な監視・診断システムを使用して潜在的な問題を検出・抑制できます。

参考までに、以下の推奨事項を確認してください。

- 管理モデルの定義

- システムロギングとモニタリング手法の定義
- インシデント対策モデルの定義
- 入力検証の実行
- 出力フィルタリングの実行

4.6 サーバーがハッキングされた場合はどうすればよいですか。

サーバーがセキュリティ侵害を受けていることが特定された場合、管理チームは可能な限り迅速かつ効率的に問題を解決する必要があります。リソース、情報およびアカウントが危険にさらされているかどうかを特定することは、難しい場合が多いです。一部の不適切な設計環境では、セキュリティ侵害の影響を定量化できないことが多いです。そのため、組織はセキュリティ上の脆弱性を解決するとともに、フィールドで危険にさらされているアセットの安全を確保する計画を実行しなければなりません。エコシステムのセキュリティを確保し、脆弱性を解決した後、組織は影響を受けた技術を再構築するための計画を進めることができます。

詳細については、以下の推奨事項を確認してください。

- インシデント対策モデルの定義
- 復旧モデルの定義
- サンセットモデルの定義
- セキュリティ区分の定義
- データタイプ区分の定義

4.7 管理者はサーバーやサービスとどのように通信を行う必要がありますか。

サービスのエコシステムに危険を及ぼさない管理モデルの開発は、IoT サービスのアーキテクチャの重要な部分です。管理モデルは複数のレイヤから構成されており、技術チームとセキュリティチームは各レイヤについて検討する必要があります。例えば、サーバーを管理する管理者は、（仮想、マイクロサービス、ユニカーネルアーキテクチャが使用されているかどうかにかかわらず）信頼性の高い安全な通信チャネルを介してライブサーバーと通信できる必要があります。ウェブアプリケーションを管理する管理者は、同じウェブ通信レイヤ上にあるアプリケーションと通信を行いますが、コードに埋め込まれた特殊なアプリケーションを使用します。

管理上のニーズにかかわらず、攻撃者が技術に通信または悪用する能力を制限するために、インターフェイスへのアクセスを制限する必要があります。以下のリソースについて検討してみましょう。

- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- 管理モデルの定義
- 明確な承認モデルの定義
- 通信モデルの定義
- ワイヤレス接続に対するプライベート APN の利用

4.8 サービスアーキテクチャは、セキュリティ侵害の影響をどのように抑えることができますか。

IoT ネットワークの有益な特徴の 1 つは、特定の消費者にサービスを提供できることです。ウェブサービスでは、各ユーザーは世界中のどこからでも、どのタイプのデバイスからでもサービスに接続できる必要があります。これは、IoT 技術では実現できません。通常 IoT 技術は、IoT サービスに接続するために特定のエンドポイントデバイスを必要とします。この相違があるため、サーバーのエコシステム設計者は、エンドポイントと消費者間の 1 対 1 の関係を利用して、エンドポイントのバックエンドデータへのアクセスを制限できます。

エンドポイントがセンサーメトリクスをバックエンドサービスに適用するとします。マイクロサービス・アーキテクチャでは、サービスのエコシステムは、特定の消費者に対応するために特定のマイクロサービスまたはユニカーネルを展開する可能性があります。技術者はこのアーキテクチャを使用して、*個々の消費者に特定*のデータやサービスを提供するために必要なリソースとアクセス機能で*のみ*、マイクロサービスがプロビジョニングされていることを確認できます。

つまり、サービスがセキュリティ侵害を受け、特定のサービスと通信できる唯一の技術がエンドポイントである場合、セキュリティ侵害によって得られるアクセス権限はエンドポイントで*既に*利用可能となっているリソースに限られるため、サービスを攻撃するメリットがなくなります。攻撃から得られるメリットは実質的にゼロです。

参考までに、以下の推奨事項を確認してください。

- サービスのトラステッド・コンピューティング・ベース（TCB）の実行
- ブートストラップ法の定義
- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- 永続ストレージモデルの定義
- 管理モデルの定義
- サンセットモデルの定義
- 明確な承認モデルの定義
- 可能であればサーバープロビジョニング

- アプリケーション実行環境の定義
- 仮想マシンのセキュリティ侵害

4.9 サービスアーキテクチャは、セキュリティ侵害によるデータ損失をどのように軽減することができますか。

IoT アーキテクチャのもう 1 つの興味深い特徴は、データ損失を軽減できることです。これは、特定のユーザーごとにサービスを分離する方法に類似しています。データは、ユーザーが認証された後に個別に分離することもできます。ただし、データベースおよびストレージのインフラストラクチャはコストがかかるため、データストレージをユーザーごとに実装することは容易ではありません。

代わりに、一意のトークンをサービスにプロビジョニングしてから、ストレージインフラストラクチャ内で特定のユーザーの代わりに動作できるようにする必要があります。この方法では、データストレージ環境へのアクセス権を持つ攻撃者はサービスに接続できることがありますが、セキュリティ侵害を受けたユーザー以外、ユーザーデータへの接続、取得または改ざんはできないはずです。

ネットワークレイヤの観点から見ると、サーバーのエコシステムからインターネットへのトラフィックのフローを抑えることも必要です。出力コントロールにより、攻撃者は特定のチャネルから知的財産権や顧客データを取得する必要があり、大量のデータを移動させることが一層困難となるか、インシデント発生中に通信を切断・検出できる通信レイヤを使用せざるを得なくなります。

詳細については、以下の推奨事項を考慮してください。

- ブートストラップ法の定義
- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- 永続ストレージモデルの定義
- セキュリティ区分の定義
- データタイプ区分の定義
- 可能であればサーバープロビジョニング
- アプリケーション実行環境の定義
- デフォルトオープンまたはフェイルオープンのファイアウォールのルール

4.10 サービスアーキテクチャは、権限のないユーザーによる接続をどのように制限できますか。

一般的な IoT アーキテクチャを活用するメリットの 1 つは、権限のないユーザーがバックエンドサービスに直接接続する能力を制限することです。ほとんどのウェブアプリケーションにはこの機能はなく、一般利用できるようにする必要があります。しかし、IoT では、エンドポイントは特定のサービスに接続する必要があるエンティティであるため、VPN（バーチャルプライベートネットワーク）を使用してバックエンドサービスにアクセスできるユーザーを制限できます。これは標準的なインターネットプロトコル上で実装するか、プライベート APN などのモバイルサービスを使用して実装できます。詳細については、以下の推奨事項を確認してください。

- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- ワイヤレス接続に対するプライベート APN の利用

4.11 リモートからの攻撃の可能性を軽減するにはどうすればよいですか。

ウェブアプリケーションおよびサービスへのリモートからの攻撃は、インフラストラクチャ管理者にとって常に懸念される問題点となっています。攻撃者が内部ネットワークや重要なリソースにアクセスできないよう、日々対策を講じる必要があります。攻撃者によるサービスのエコシステムへのセキュリティ侵害が発生するリスクを低減するための唯一の方法は、迅速かつ容易に維持・管理できるサービスセットによって標的となる可能性を軽減することです。アーキテクチャを強化するための 2 番目に重要な要素は、基盤となるアーキテクチャの設計です。アプリケーションが安全に実行できるかどうかは、実行アーキテクチャ、オペレーティングシステムの構成、展開ツールチェーン、プログラミング言語のセキュリティなどのオプションにかかっています。アプリケーションクラッシュとインフラストラクチャのセキュリティ侵害とでは、利用できるオプションが異なる場合があります。

リモートからの攻撃リスクを低減する方法の詳細については、以下を参照してください。

- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- 更新モデルの定義
- 入力検証の実行
- 出力フィルタリングの実行
- デフォルトオープンまたはフェイルオープンのファイアウォールのルール
- アプリケーション実行環境の定義
- Rowhammer や類似の攻撃
- 仮想マシンのセキュリティ侵害

4.12 サービスはユーザーのプライバシーをどのように管理することができますか。

IoT サービス提供者は事業成長に伴い、革新的な方法で消費者データを活用する組織とパートナーシップを締結することが不可欠となります。しかし、このデータにより、消費者のプライバシーが犠牲になっています。消費者は、どのようなデータがパートナーと共有され、それがどのように使用されるかどうかを決定する権利があります。また、パートナーは特定の方法でデータを使用することが求められます。承認モデルはこれらの要件をサポートしますが、プライバシーに関する検討事項は、法的な影響や事業保険など、多岐にわたります。

組織で検討を開始するにあたり、以下の推奨事項を確認してください。

- セキュリティ区分の定義
- データタイプ区分の定義
- 明確な承認モデルの定義
- 流出データに対する違反ポリシーの定義
- 通信プライバシーモデルの評価
- サードパーティーのデータ配布ポリシーの定義
- サードパーティーのデータフィルターの構築
- ユーザーがプライバシー属性を管理するための API の構築

4.13 サービスはその可用性をどのように向上できますか。

サービス拒否攻撃（DoS）または分散型サービス拒否攻撃（DDoS）は、現代のインターネットでは一般的な攻撃であり、すべての企業がこのクラスの主な攻撃に直面する準備を行い、長時間攻撃を受ける状況でもオンライン状態を保つことができるようにする必要があります。これらの攻撃が一般的にみられるようになった理由は、攻撃を実行するのに特別な技術やツールは必要なく、オンライン上でいつでも行うことができるからです。オンラインサービスでは、悪意のある者が攻撃者に報酬を提供し、特定のターゲットに対して DDoS 攻撃を実行させる可能性があります。

このような脅威に対応するために、サービスの可用性を高めるための新しいモデルが構築されました。サービスのエコシステムを構築する場合は、以下の推奨事項を考慮してください。

- パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義
- システムロギングとモニタリング手法の定義
- インシデント対策モデルの定義

- 復旧モデルの定義
- 通信モデルの定義
- デフォルトオープンまたはフェイルオープンのファイアウォールのルール

5 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。

5.1 サービスのトラステッド・コンピューティング・ベース（TCB）の実行

トラステッド・コンピューティング・ベース（TCB）は、ハードウェア、ソフトウェア、プロトコルおよびポリシーのセットです。TCB は特定のコンピューティングプラットフォームの基盤となるもので、高品質を保ちながらアプリケーションを確実かつ安全に実行できる環境を定義する必要があります。

TCB は、モバイル機器（スマートフォン）、IoT エンドポイント、サービスのエコシステム上で稼働するサーバーなどのシステムに対して構築・展開できます。TCB はすべて、類似した技術で構成されていますが技術の特性はシステムクラスに応じて異なります。例えば、クラウドサーバーへの TCB のブートストラップは、エンドポイントのブートストラップと大きく異なります。

サービスのエコシステムに TCB を構築する場合、アプリケーションイメージを展開する方法を定義します。この場合のイメージとは、実行可能なアプリケーションとその構成ファイル、メタデータで構成されたローバイナリデータです。これらを総称してアプリケーションイメージ、または単にイメージと呼びます。最新のサービスのエコシステムでは、コンピューティング環境の変化に応じてシステムが複製、強化またはスピンダウンします。つまり、TCB は、システムが永続的なセキュリティモデルを維持しながら効率的に対応できる方法を定義する必要があります。

これを正常に行うには、チームは以下の項目を実施する必要があります。

- コンピューティングプラットフォームの標準化：
  - 物理的なサーバーモデルのセットを選択します
  - クラウドプラットフォームまたは仮想マシン（VM）イメージセットの選択
- コンピューティングプラットフォーム上で実行するアプリケーション、ライブラリ、構成ファイルのセットの定義：
  - コンテナ環境を必要に応じて定義します
- 上記のように定義されたセットに基づいた、アプリケーションイメージの生成
- 階層 TCB 署名鍵を使用したイメージアーカイブの暗号署名

- アーカイブと署名の安全な保管

このタスクセットを実行することで、特定の階層に展開できるアプリケーションイメージを承認できます。各階層には、特定の階層に最適なハードウェアとアプリケーションモデルがあります。例えば、データベースのハードウェアには、アプリケーション層とは異なるパフォーマンスとストレージのニーズがあります。ストレージ層のハードウェアストレージ要件はデータベース層と似ていますが、パフォーマンス要件は異なります。各階層の定義を標準化すると、各ハードウェアプラットフォーム上でイメージを展開・検証できます。

以下の理由から、TCB の展開は困難が伴います。

- イメージの暗号署名を管理するために、組織の信頼の基点（Root of Trust）を設定する
- 各イメージの署名手順を設定する
- 各イメージの検証手順を設定する
- イメージの検証とともに、イメージの自動展開手順を設定する

以下の組織が提供する参考資料を参照することをお勧めします。

- GlobalPlatform Card Specification [11]
- Trusted Computing Group’s TPM Specification [6]
- GlobalPlatform TEE Internal Core API Specification [12]

5.1.1 リスク

適切に定義されたトラステッド・コンピューティング・ベースがない場合、コンピューティングプラットフォームは、技術チームにより承認された構成で現在実行されていることを確認することができません。アプリケーションのサブシステムは、システムが攻撃者によるセキュリティ侵害を受けているかどうかを判断できる必要があるため、この点は非常に重要です。TCB はこのリスクを修正するとともに、すべてのネットワーク通信に対してセキュリティレイヤを提供するために使用できます。

5.2 組織の信頼の基点（Root of Trust）の定義

組織の信頼の基点（Root of Trust）は、組織内のコンピューティングプラットフォームエンティティを認証するための証明書または公開鍵をベースにしたシステムです。サービスのエコシステムにおける各コンピュータプラットフォームは、ネットワーク通信時に暗号方式で認証される必要があります。これにより、内部者またはネットワーク上の特権的な立場にあるユーザーが、特権的なシステムの信頼を偽装または悪用する能力を抑制できます。

組織の信頼の基点を構築するには、次の操作を実行します。

- 組織の基点の秘密鍵を保管するために、ハードウェアセキュリティーモジュール（HSM）などを構築または取得する
- 基点の秘密鍵および／または証明書を生成する
- 秘密鍵のプライベートファセットが安全に保管されていることを確認する
- 階層 TCB 署名鍵に使用される 1 つまたは複数の署名鍵を生成する
- 組織の基点により、署名鍵の公開ファセットに署名する
- これらの鍵が、経営陣および技術責任者からの認証と承認なしに使用できないようにする

新しい階層システムが定義されているたびに、その一意の暗号鍵や証明書は、署名鍵によって署名できるようになります。別のシステムがこの新しいシステムに接続している場合、組織の基点で定義された信頼のチェーンを確認することで、システム ID を検証できます。

メッセージがシステムの公開鍵によって署名されたことを暗号方式で検証した後、署名鍵が生成した署名が、システムの一意の公開鍵によるものであることを確認します。その後、クライアントは、署名鍵が組織の基点によって認証された署名鍵であることを確認する必要があります。

証明書や秘密鍵の各セットは、組織内の少数の個人に使用が限定されており、定義されたポリシーと手順が秘密鍵の利用者と利用時期を制限しているため、クライアントが基点のチェーンを降下するにつれて、信頼の各レベルが上昇する必要があります。

サービスは、サービスのエコシステム内で承認されたピアに認証機能を提供するよう定義される必要があります。例えば、証明書または秘密チェーンによる認証は、自身のセキュリティを保証するために使用することはできません。サービスは、証明書が現在有効かどうかを確認できる必要があります。場合によっては、基盤となるインフラストラクチャの要件に応じて、短い寿命を持つサーバーまたはサービスの身元を認証するために別のサービスを使用する必要があります。

信頼の基点を定義する場合、以下の項目を考慮してください。

- 各秘密鍵を悪用から守る
- 各秘密鍵の内部使用を追跡・監視する必要がある
- 秘密鍵の利用を承認された個人は、秘密鍵にアクセスする際にマルチファクター認証を使用する必要がある

- ポリシーと手順のセットを定義し、一貫性のある安全な使用を徹底することが困難となる恐れがある
- 証明書を廃止または取り消す手順を策定することが困難となる恐れがある
- 鍵が悪用されているかどうかを特定することが困難となる恐れがある
- 暗号化アルゴリズムの適切なセットの選択が直感的でない恐れがある

信頼の基点に関する詳細については、以下の情報元を参照してください。

- Trusted Computing Group
  - TPM Specification [6]
  - TCG Guidance for Securing IoT [7]
  - ISO 11889
- PKI Specifications
  - RFC 2510
  - RFC 3647

5.2.1 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。

5.3 ブートストラップ法の定義

アプリケーションを正常に実行するには、信頼性の高い、高品質の安全なプラットフォーム上にアプリケーションを一貫した方法で読み込み、実行する必要があります。TCB はこのプラットフォームを策定する方法を定義しますが、ブートストラップモデルは、アプリケーションを実行する方法を定義します。

ブートストラップモデルを効果的に定義するには、以下の項目を考慮する必要があります。

- ピアがアプリケーションを暗号識別できるよう API を定義する
  - 信頼できる業界の大手企業によって定義された、既存の API の利用を検討する
- アプリケーションがエンドポイント、サービスピア、パートナーを認証する方法を定義する
- アプリケーションの適切な構成を定義する
- 特に階層ごとに異なるアプリケーションを実行する場合、各アプリケーションに一意の ID を割り当てる

ピアがアプリケーションを暗号識別できるようにするために、API は必ずしも必要ではないため、本番環境での工程は少し非直感的となります。これは、ブートストラップモデルでは、暗号化された ID をアプリケーション*に*プロビジョニングする*方法*を検討する必要があるからです。アプリケーションは ID をどのように取得するのでしょうか。ID は安全に取得できるでしょうか。秘密鍵を更新または変更する場合に ID が使用する秘密鍵を取り消す手順は何でしょうか。

アプリケーションを効果的に実行するためには、特定のリソースが必要です。アプリケーションは、この工程で必要なすべての外部サービス、エンドポイントおよびパートナーを用いて通信を確立し、相互認証を実行する必要があります。

アプリケーションの構成により、本番環境の安全性が決定されることが多いです。アプリケーションの構成は読み取り専用で策定される必要があります。アプリケーションのインフラストラクチャを悪用するアプリケーションや第三者が、アプリケーションの構成を簡単に変更できないようにする必要があります。

組織の信頼の基点（Root of Trust）を使用して、エコシステム全体に展開されている各階層の信頼モデルを定義します。これにより、それぞれのアプリケーションに一意の暗号化された ID を割り当てることができます。これは、例えばデータベースサービスとアプリケーションサービスを区別する機能をピアに提供します。

5.3.1 リスク

適切に定義されたブートストラップモデルがない場合、システムは動作に必要な各レイヤを確認できません。テクノロジーの各ファセットには、実質的に信頼レイヤがありません。信頼レイヤがないため、構造上の複雑性によって生じた欠陥を攻撃者が悪用する可能性があります。

5.4 パブリックネットワークを使用するシステムに対するセキュリティインフラストラクチャの定義

一般利用可能なサービスは、サービスの可用性、機密性および完全性を維持するためのセキュリティと信頼性に関する技術を必要とします。

- DDoS 耐性インフラストラクチャ
- ロードバランシング（負荷分散）インフラストラクチャ
- 冗長化システム
- ウェブアプリケーションファイアウォール（任意）
- 従来のファイアウォール

これらの技術をアプリケーション階層の前に配置し、外部の攻撃者が操作できないようにする必要があります。通信セキュリティモデルは、匿名の第三者がシステムにアクセスする可能性を軽減または是正しますが、これらの技術は、攻撃者がシステムを使用できないようにする能力を抑制します。

フロントエンドのセキュリティは、サービスで実装されている*すべての*プロトコルに適用する必要があります。例えば、サービスが IPV4 および IPv6 上で利用できる場合、同じセキュリティ制約を両方のプロトコル上のサービスに適用する必要があります。TCP およびストリーム制御伝送プロトコル（SCTP）からサービスにアクセスできる場合、セキュリティ制約をこれらのプロトコルの両方に適用する必要があります。IoT 製品またはサービスに特化した公共サービスを提供していないポートを、アクセス不可にする必要があります。

可能であれば、入力*および*出力フィルタリングの両方が管理されていることを確認します。入力フィルタリングは攻撃範囲の拡大を阻止しますが、一般利用可能なサービスに対する攻撃により、サービスのエコシステムへのセキュリティ侵害が発生する可能性があります。攻撃者がサービスのエコシステム内を移動するために、セキュリティ侵害を受けたコンポーネントを利用できないようにするには、出力フィルタリングが不可欠です。また、出力フィルタリングは、攻撃者がエコシステムから重要なデータを攻撃者が管理するサーバーに密かに移動させ、管理者による攻撃者の特定・隔離にかかる時間を増やす能力を抑制するのに役立ちます。

一部の組織は、特定の技術にドロップできる単純な API モデルにこれらのサービスを提供しています。これにより、技術の利用にかかる労力を削減できます。必要な作業は、サービス提供者のシステム内でアプリケーションの登録や設定を行うだけです。ご利用のサービス提供者に相談して、自社の環境に最適なセキュリティ技術を実装方法を決定します。

以下の組織が提供する参考資料を参照することをお勧めします。

- Amazon Best Practices for DDoS Resiliency：
  - https://d0.awsstatic.com/whitepapers/DDoS_White_Paper_June2015.pdf
- Arbor Networks DDoS Mitigation Best Practices：
  - https://www.arbornetworks.com/images/documents/Arbor%20Insights/AI_DDoSMitigation_EN2013.pdf
- Cisco DDoS Defence Guide：
  - http://www.cisco.com/web/about/security/intelligence/guide_ddos_defense.html

5.4.1 リスク

インターネットの不安定な性質を考慮すると、一般利用可能なサービスやアプリケーションには、安全なインフラストラクチャが不可欠です。ランダム DDoS 攻撃は、特別な理由もなく頻繁に行われます。DDoS サービスは、「闇市場」で数百米ドルで購入することができます。そのため、企業またはその顧客をターゲットにしている攻撃者は、このような攻撃の唯一の加害者ではありません。システムダウンを引き起こすことができるかどうかを確認するために、ランダム攻撃が行われる可能性があります。重要な IoT サービスの予期しない利用停止を防ぐために、このような攻撃への対策を講じるべきでしょう。可用性は、IoT 製品またはサービスの重要なコンポーネントです。

5.5 永続ストレージモデルの定義

現代のコンピューティングのアプリケーション環境は、多くの場合、コンテナベースのシステムなどのエフェメラルポートか、クラウド環境です。そのため、これらのシステムに割り当てられたストレージが十分な大きさではないか、アプリケーションがこれらの技術を永続的なストレージとして使用できるよう、長期間利用できるように設計されていません。また、これらのシステムはオンデマンドエンティティとして定義されており、一元化されていない場合があります。つまり、*他のシステム*は、永続的な使用のために十分なストレージシステムを定義できません。

そのため、中央のストレージシステムが不可欠であり、その安全性を確保しなければなりません。このような環境下でストレージシステムは特定の一時的なシステムにアクセスできる必要があるため、セキュリティ侵害を受けた寿命が短いサーバーまたはサービスは、他の多くのサーバーまたはサービスによって使用される永続的なストレージエンティティ（または階層）にアクセスできます。これは多くの場合、攻撃者が任意のネットワークを横断的に（または縦断的に）攻撃できる効率的な方法となります。

これを抑制するには、各サーバーまたはサービスは永続的なストレージにアクセスするだけでなく、対象となるアプリケーション、およびアプリケーションが代理を務める *一意のエンドポイント*、*パートナーまたはユーザー*に基づいて情報を保存する必要があります。最後に、そして最も重要な点は、特定の *ID* の代わりに永続的なストレージにアクセスして、寿命が短いサーバーまたはサービスのデータへのアクセスを制限することです。

つまり、攻撃者が*寿命の短いシステム*へのセキュリティ侵害を行った場合、その影響を受けるのは、*同じシステム*に関連した ID の代わりに保存されたデータのみです。このシステムが単一の ID のデータにのみアクセスできる場合、攻撃者はこのシステムへのセキュリティ侵害を利用して、他のアカウントへ横断的に移動することができません。単一の ID に関する情報へのアクセスのみに制限されます。これにより、攻撃者が脆弱性を利用してシステムを悪用する能力が大幅に制限されます。

5.5.1 リスク

安全かつ永続的なストレージモデルが定義されていない場合、他のアセットから安全に分離された一意のユーザー属性を適用するアーキテクチャはありません。そのため、攻撃者がトークンへのセキュリティ侵害を行い、ストレージデバイスへのアクセス権限を取得した場合、複数のユーザーのデータがセキュリティ侵害を受ける可能性があります。しかし、永続的なストレージモデルはデータの暗号化によって、セキュリティ侵害の被害を 1 人のユーザーまたは単一のストレージ技術に限定できます。いずれの場合も、セキュリティ侵害の範囲は大幅に縮小され、組織はユーザーとビジネスの両方に対する脅威の対応により多くの時間を費やすことができます。

5.6 管理モデルの定義

管理者は各システムにアクセスし、アプリケーション障害のトラブルシューティングと診断を行うことができる必要があります。サービスまたはサーバーの寿命が短い環境において、管理モデルが適切に設計されていない場合、これは困難となる場合があります。

アプリケーション障害のトラブルシューティングと診断を実現するには、管理チームによる各階層内の各システムとの通信方法を特定します。独立したシステムを分離するために、VPN などの認証の境界が必要です。管理チームは、各階層の認証を行う必要があります。

また、管理者によるシステムの通信方法を特定します。システムは、VM のようにスナップショットできますか。端末を使用していますか。リモートのセキュアシェル（SSH）は、システムの通信に使用されていますか。CPU 使用率、ディスク使用率、ネットワーク使用率など、システムメトリクスを監視・分析するための API はありますか。これらの API は、異常のトラブルシューティングまたは検出に使用できますか。

モデルにかかわらず、定義する*必要*がある項目が複数あります。

- 管理者による環境の認証方法
- 認証を行う管理者の身体的特徴による識別方法
    - 2 ファクタ認証（2FA）の使用
- システムによるスナップショットの作成方法
- 変更方法と変更を追跡する方法

5.6.1 リスク

管理者のアクセスに対して適切に構築されたパスがない環境では、本番環境のシステムにアクセスするための暫定的な措置を行うことになります。このため、管理ポートへのパブリック接続が可能となるか、サービスが診断を提供するにもかかわらず、第三者による利用が制限されていない状況が引き起こされます。明確な管理モデルにより、攻撃者が重要な IoT リソースへの特権的なアクセス権限を得るための手段を制限できます。

5.7 システムロギングとモニタリング手法の定義

各システムの監視を行うことで、管理者および情報技術（IT）部門が異常を検出・診断できるようにする必要があります。監視は、複数の観点から実行する必要があります。例えば、インフラレベルでのネットワーク監視は、アプリケーションへの攻撃やネットワークコンポーネントに対する DDoS 攻撃の診断に役立ちます。階層の監視は、特定のアプリケーションやインフラストラクチャがセキュリティ侵害を受けているかどうかを検証します。システムレベルの監視は、個々のアプリケーションやアプリケーションのプラットフォームが攻撃されている、またはセキュリティ侵害されているかどうかを特定します。

このように、監視には複数のレベルがあり、監督チームに提供されるリソースに情報を統合します。この技術を提供するとともに、IT 担当者やシステムエンジニアが利用できる視覚的システムにメトリクスを変換するアプリケーションは複数あります。

敵対的な動作を示す異常には以下のものが含まれますが、これらに限定されません。

- ネットワークトラフィックの増加
- *奇数*の方向（特に出力）でのネットワークトラフィックの増加
- 出力する必要がないリソースからのネットワークトラフィックの出力
- 異常な CPU 使用率
- 視覚的なインターフェイスを持たないが、CPU の一部として GPU があるシステムに対する GPU の使用率
- ディスクやネットワークストレージの使用率
- 特定のホスト上でのシステム時間の異常な変化

異常を検出するための監視システムは既に利用可能ですが、組織で使用されるアプリケーションやインフラストラクチャによって状況は変わります。監視システムを提供する企業と相談して、特定の実装に最も効果的な方法でメトリクスを取得および解釈する方法を決定します。

各階層には、攻撃やセキュリティ侵害を示す様々な異常が発生する可能性があります。各階層ごとに指標を評価します。

以下の組織が提供する参考資料を参照することをお勧めします

- Amazon EC2 Monitoring Documentation
  - http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html
- Google Cloud Monitoring
  - https://cloud.google.com/monitoring/
- Microsoft Azure Monitoring
  - https://azure.microsoft.com/en-us/documentation/articles/best-practices-monitoring/
- DigitalOcean Monitoring Tutorials (General)
  - https://www.digitalocean.com/community/tags/monitoring?type=tutorials

5.7.1 リスク

技術を監視するシステムは、IoT セキュリティモデルの主な特徴の 1 つです。監視以外に、重要なサービスコンポーネントに脆弱性が発見されたかどうかを確認する方法はありません。監視を行うことで、管理者はサービスとインフラストラクチャの問題点を迅速に診断し、セキュリティインシデントとソフトウェアバグの区別をサポートできます。

5.8 インシデント対策モデルの定義

潜在的なセキュリティ侵害や進行中の攻撃を検出するだけでは不十分です。組織は、攻撃に対応し、阻止できる必要があります。システムがセキュリティ侵害された場合、システムのクレンジングやシャットダウンを行うだけでは十分ではありません。組織は、セキュリティ侵害を行った攻撃元を診断し、システムにパッチを適用するとともに、すべての既存のインフラストラクチャ上にパッチを展開できる必要があります。

コンテナベースの環境において、脆弱性のある構成でクローンされたアプリケーションが動作している場合、この措置は困難な場合があります。アプリケーション接続がクラウド上で別のシステムに渡される、またはユーザー

が強制的にログアウトされ、アップデートが許可される場合、アプリケーションシステムは「再起動」または「アップデート」イベントを検出できる必要があります。

実行モデルの種類にかかわらず、技術チームはフォレンジック分析が可能な方法でメトリクスを取得できる必要があります。これらのポリシーと手順を策定し、情報が法執行官（LEO）が認める方法で記載されていることを確認するために、法執行機関（および保険チーム）による承認を受ける必要があります。コンプライアンスは、企業が現地法および連邦法を遵守しているだけでなく、裁判所にセキュリティ侵害のサンプルを提供するのに役立ちます。

サンプルを収集した後、問題となっているイベントを裏付けるログやメトリクスなど、システム上のあらゆるデータを検証する必要があります。法的検証を行うために、これらのデータをすべて収集し、安全なシステムに保存します。

以下の組織が提供する参考資料を参照することをお勧めします。

- CERT Recommendations for Creating a CSIRT
  - http://www.cert.org/incident-management/products-services/creating-a-csirt.cfmhttp://www.cert.org/incident-management/products-services/creating-a-csirt.cfm

5.8.1 リスク

インシデント対応モデルがない組織は、リソースの整理、セキュリティ侵害を受けたシステムの特定と隔離、情報収集のためのシステムの検証に、はるかに多くの時間を要します。また、システムにパッチを適用して復元する時間もかかります。このような無防備の状態では、攻撃者がセキュリティ侵害を悪用し、特定の環境下で横断的または縦断的に移動する機会を与えてしまいます。対応にかかる時間が長くなると、深刻なセキュリティ侵害を招く恐れがあります。組織は、インシデントの発生時に直ちに対応できるよう準備を整え、攻撃者がサービスの重要な部分を制御する時間を短縮する必要があります。

5.9 復旧モデルの定義

セキュリティ侵害やハードウェア障害の影響を受けるのがユーザーかアプリケーションかどうかにかかわらず、復旧を行う必要があります。アプリケーション層内の情報と機能を復旧する手順を策定しなければなりません。この手順は、各アプリケーションと階層のコンテキストに合わせて調整する必要があります。

例えば、アプリケーションが特定のアクションの出力に関する情報をエンドポイントから収集した際に、アプリケーションが永続的なストレージのデータ出力を確立できないようにするストレージ障害が発生した場合、アプリケーションは、以下の項目を実行できます。

- ストレージが成功するまで再試行する（場合によっては無限に繰り返す）
- 成功または失敗のしきい値に達するまで、試行回数の上限までストレージを再試行する
- 直ちにストレージに失敗し、メトリクスを失う恐れがある
- エンドポイントに同じデータを依頼する（利用不可となっている場合がある）

アプリケーションと企業の要件に最も適した方法を選択する必要があります。繰り返しになりますが、方法はアプリケーションのコンテキストに応じて異なるため、システム外でモデル化することが困難となる恐れがあります。

技術責任者と経営陣は、特にユーザーアクティビティの場合、障害が発生した、またはセキュリティ侵害を受けたアプリケーションの復旧方法を決定します。

攻撃者によるセキュリティ侵害を受けたシステムの場合、復旧の前に、アプリケーションやシステムに十分なパッチが適用されていることを検証するためのモデルを策定する必要があります。このようなポリシーや手順が策定されていない場合、脆弱なシステムはサービスのエコシステムに再展開され、さらなるセキュリティ侵害を助長する恐れがあります。

5.9.1 リスク

復旧モデルは、情報、アプリケーションおよび構成の正常な復元をサポートします。復旧モデルがない場合、チームはサーバーやインフラストラクチャに脆弱なサブシステムを意図せず再展開する可能性があります。また、データベースやストレージ環境で攻撃者によって操作された可能性がある汚染されたデータは、複数のシステム上でレプリケートされ、意図せずマルウェアを伝搬したり、データを改ざんする恐れがあります。復旧手順を実施することにより、攻撃者がインシデントからの復旧における弱点を悪用し、被害を拡大させる能力を制限します。

5.10 サンセットモデルの定義

組織が展開しているすべてのシステムと階層には、寿命があります。組織が同じ製品やサービスを数十年にわたり展開していたとしても、技術によって製品やサービスの仕様が変更されます。製品やサービスの設計と提供に関する計画だけでなく、製品やサービスの*廃止*に関する計画も必要です。

このプロセスは、攻撃者が技術の詳細を把握したり、その利便性を利用できないように、すべての技術を確実に撤回・廃止するのをサポートします。例えば、単純な事例は、親会社による事業買収後の特定の製品用のドメインです。製品の名前が変更され、ドメインが親会社のドメインに移行された場合、攻撃者は古いドメインの所有権を取得できる可能性があります。攻撃者がそのドメインの暗号化証明書を発行することができる場合、この古いドメインの下に展開された技術にアクセスでき、該当する製品またはサービスの廃止手順が欠落していることにより、重大なセキュリティ欠陥が発生します。

特定の製品またはサービスのアーキテクチャ、実装および管理で使用されるそれぞれの技術について、目録を作成して評価する必要があります。技術が利用できなくなると、そのモデルに従って技術を廃止できます。これにより、技術者と経営陣は、基盤となるプラットフォーム間のずれを引き起こすことなく、技術をより適切なものに移行できます。また、パートナーおよびユーザーに提供されなくなった製品の寿命が終了し、事業終了後に攻撃者によって悪用される可能性を確実に排除します。

5.10.1 リスク

廃止工程が欠落している場合、エンドポイントとサービスは、競合他社や攻撃者によるセキュリティ侵害を受ける恐れがあります。組織はドメイン名、電話番号、他の更新可能なサービスなどへのアクセス権限を公開している場合、攻撃者や競合他社にはこれらの情報やサービスを取得する権利があるため、非倫理的な行為にみえますが、合法的に行うことが可能です。そのため、デバイスやサービスが悪用や悪意のある行為などの脅威にさらされる恐れがあります。

5.11 セキュリティ区分の定義

パートナー企業との取り決めを適切かつ効果的に管理するためには、セキュリティの分類を定義する必要があります。これは、データセキュリティに関する内部ポリシーを策定するだけでなく、パートナー企業が貴社のデータ、自社のデータ、顧客データに適用するセキュリティレベルを定義するのに役立ちます。

このプロセスは検証を行い、組織に合わせてカスタマイズする必要がありますが、ほとんどのデータセキュリティ区分ポリシーは、まず以下の区分を採用しています。

- 公開 - 任意のエンティティに付与されたアクセス権限
- 分類 - ユーザーは開示認証を受ける必要があります
- 秘 - ユーザー固有のデータ
- 最高機密 - 組織固有のデータであり、開示されることはありません

基本的な区分を定義した後、組織は各セキュリティ区分はどのようにデータ区分に帰属するかを評価する必要があります。つまり、分類だけでなく、実際に利用される必要があるかどうかを評価します。事業*および*技術の観点から、どのようなポリシーと手順を策定するかを決定します。

これにより、組織は技術ポリシーだけでなく、技術的な要件に対応する事業ポリシーを制定できます。また、技術チームは、意図的か否かにかかわらずポリシーに違反しようとするパートナーや内部組織にこれらの要件を伝達しやすくなります。

セキュリティ区分を標準化した後、企業とそのユーザーのプライバシー要件がセキュリティ区分モデルに与える影響を検証することが重要です。組織は、時間をかけてセキュリティ区分にプライバシーモデルを適用し、ユーザーのデータに意味を持たせ、ユーザーのデータが流出する可能性がある特定のリソースへのアクセスをパートナーが希望する場合にユーザーのプライバシーを保護する必要があります。パートナーは、プライバシーにかかわる特定のデータを取得したい場合、セキュリティ区分のコンテキストにおけるプライバシーの状況を説明して、経営陣*および*ユーザーによる承認を求める必要があります。ユーザーは、自分のプライバシーにかかわるデータを保護する選択肢があり、第三者へのデータの開示を制限できる*必要があります*。

5.11.1 リスク

効果的にセキュリティを利用するソリューションを構築するためには、セキュリティに関するモデルの分類が不可欠です。情報を保護するためには、情報を定量化し、対応するポリシーや手順に基づいて適切な管理体制を整える必要があります。これらのモデルがない場合、技術者たちは、リスクの捉え方に応じて極端な方法でセキュリティを実装するか、セキュリティを全く実装できない傾向があります。技術者や経営陣を含むチーム全体が、企業に対するデータの意味と、適切な費用対効果の管理範囲におけるデータの安全性を把握しなければなりません。

5.12 データタイプ区分の定義

セキュリティ区分を定義した後、組織は IoT 製品またはサービスによって使用されるデータの種類を定義する必要があります。これにより、組織は IoT システム内で収集および生成し、ピアに伝達する情報の種類と、これらのデータを処理する方法を明確に定義できます。このデータは、IoT 環境全体で使用されるすべてのコンポーネントにコンテキストと値を提供します。

本文書では、特定の組織に関連するすべての種類のデータをモデル化しませんが、一部のデータの種類を以下に紹介します。

- ユーザー
- アクション
- 画像
- 編集可能な文書
- 個人識別情報
- 保護医療情報

情報は、1 つまたは複数の*種類*に区分されることがあります。しかし、データ自体は 1 つの*セキュリティクラス*にのみ帰属する必要があります。*種類*はデータの内容と処理方法を識別する一方、*セキュリティクラス*は*どのように*、*どこで*、*いつ*情報を使用でき、*誰*に共有される可能性があるかを示しています。

データの種類と区分を定義するには、時間と手間がかかります。この作業によって組織内の基準を定め、技術チームがデータとその区分に関する技術的な管理を実行できるようにします。これは、技術チームと経営陣が、データの共有方法と処理方法についてパートナーと交渉する際に役立ちます。

5.12.1 リスク

セキュリティ区分と同様に、データの種類と事業との関係を定量化する必要があります。これらのクラスは、システム上での情報の使用方法と、適切なセキュリティ対策を維持するためにデータに適用する必要がある保護措置を定義します。これらのクラスがない場合、技術者は規制が厳しすぎる、または緩すぎるセキュリティ措置を適用する傾向があります。事業に対するデータの重要性と管理のバランスを取るために、セキュリティ措置の策定には、技術チームと経営陣の合意が必要です。

6 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。

6.1 明確な承認モデルの定義

プライバシーモデルはユーザー情報がパートナーに提供される方法を定義する一方、認証モデルはパートナーがユーザーの代理を務める方法を定義します。これらのモデルは、パートナーのメトリクスが指定された家庭の冷暖房を最適化できる、ホームオートメーションシステムなどに役立ちます。認証モデルにより、パートナーは特定のメトリクスを検出した場合、ユーザーの自宅の冷暖房を調整できます。

これを実現するために、細分化された認証機能の詳細とこれらの機能をパートナーに提供する GUI を準備します。また、ユーザーが必要に応じて特定の機能に対するアクセス権限を取得または取り消すことができるようにします。悪用されるリスクを軽減するために、取り消された機能の処理を直ちに行います。

パートナーが許可されていない行為を行うことを防ぐために、システムの監視を徹底する必要があります。認証モデルの管理を細分化することにより、パートナーが特定の機能にアクセスする時期と頻度をユーザーが設定できるようにします。このような属性により、パートナーによる悪用またはセキュリティ侵害（ハッキング）の恐れがある場合に、ユーザーが自分のシステムを管理できる能力を高めることができます。

6.1.1 リスク

認証モデルがない場合、第三者はユーザーの機能に無制限でアクセスできます。これにより、悪意のある、またはセキュリティ侵害を企てる第三者が、ユーザーの技術やデータへの完全なアクセス権限を取得する可能性があります。認証モデルを作成することによって、ユーザーが許可する属性のみにアクセスが制限されます。ユーザーは、第三者が利用できる機能やデータの管理を強化するとともに、セキュリティ侵害の拡大を緩和することで、IoT サービス提供者のリスクを軽減できます。

6.2 暗号化アーキテクチャの管理

IoT 環境で展開されるすべての技術は、それが初歩的な低消費電力のエンドポイントか頑強なクラウドサービスかどうかに関わらず、暗号化を使用する必要があります。IoT 製品やサービスでセキュリティを適切に実

装するには、時間の経過とともに変化する仕様に対応できるよう、暗号化の適切な設計、管理および調整を行わなければなりません。

技術チームは、以下の項目を確認する必要があります。

- 使用する暗号化アルゴリズムが非推奨であるか
- ビットの長さが十分な暗号鍵を使用しているか
- ハッシュアルゴリズムは衝突攻撃の対象となるか
- 強力な乱数生成器を使用しているか
- メッセージに十分なランダムデータがあるか
- TLS などの暗号化プロトコルは、ベストプラクティスによって最新の状態になっているか
- 前方秘匿性（Forward Secrecy）などのプライバシーを第一に考えたコンセプトを採用しているか
- プレーンテキストのパスワードや暗証番号が、ネットワークを介して渡されているか
- カスタム暗号化アルゴリズムを使用したか

これらの項目は、IoT 製品またはサービス内で高品質な暗号化アーキテクチャを維持するために重要です。適切な暗号化ソリューションを展開できるかどうかは、技術チームが、回復力の低いソリューションを使用する技術にパッチを適用するために、回復力が最も高い暗号化ソリューションを活用する能力にかかっています。

例えば、先日 RC4 アルゴリズムに重要なセキュリティ上の欠陥があることが判明しました。RC4 を使用するクライアントに RC4 を AES-256 に交換するパッチを安全に配布できる場合、RC4 に対する懸念は少なくなります。ディフィー・ヘルマン鍵交換や非対称鍵などの回復力の高い技術か、UICC セキュリティトークンを使用して相互認証が実行された場合、脆弱な暗号化アルゴリズムを使用せずにパッチを検証できます。

通信チャネルが暗号化によって保護されている場合であっても、ユーザーまたはエンドポイントによって使用されるパスワードと暗証番号は、プレーンテキストでネットワーク上で伝送すべきではありません。代わりに、パスワードや暗証番号の暗号ハッシュを使用して、暗号化トンネルにおける設定ミスにより、パスワードが公開されないようにする必要があります。ハッシュは、パスワードと 1 つ以上の一意のワンタイムトークンによって生成する必要があります。このトークンは一般的にネットワークセッションから取得されますが、エンドポイントとサービスインフラストラクチャの両方に保存されているローリングコードから値を取得するほうがより安全です。この方法により、ネットワーク上で特権的な立場にある攻撃者が、自分にとって有利な値をハッシュに適用することで、署名を強制することを阻止できます。

カスタム暗号化アルゴリズム（社内で設計されたアルゴリズム）を使用しないでください。暗号化セキュリティを専門とする監督機関が推奨する、暗号作成者によって開発されたアルゴリズムを常に使用してください。また、不適切に設計されたアルゴリズムや、非推奨のアルゴリズム、圧縮およびバイナリテキスト変換、一般的に暗号化アルゴリズムであると誤解されているその他のアルゴリズム（LZO、base64、ROT13、XORなど）を使用しないでください。

このトピックの詳細については、以下のガイドや参考資料を参照してください。

- ISO 18033-1:2015 – Encryption Algorithms
- ISO 18033-2:2015 – Asymmetric Ciphers
- ISO 18033-3:2015 – Block Ciphers
- www.owasp.org/index.php/Guide_to_Cryptography
- csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf
- csrc.nist.gov/groups/ST/toolkit/key_management.html

6.2.1 リスク

暗号化アーキテクチャを伴うソリューションを適切に展開することで、使用されているアルゴリズム、プロトコルおよび秘密鍵が、すべて現在推奨されているものであることが保証されます。また、推奨されるアルゴリズム、プロトコルおよび秘密鍵は、時間の経過とともに変化します。暗号化アーキテクチャがない場合、非推奨となった技術をすべて特定することがより難しくなり、セキュリティ上の欠陥を引き起こします。

6.3 通信モデルの定義

サービスのエコシステム内の各システムは、相互認証に対応している必要があります。このエコシステム内のコンピューティングプラットフォームはすべて、匿名のパブリックユーザーにアクセスできないはずです。各エンドポイント、パートナーまたはユーザーは、相互認証を必要とする技術により、サービスのエコシステムとの通信を行います。ユーザーインターフェイスを構成するサービスは、一般的に別の環境で展開・管理されているため、一般利用可能なインターフェイスをこの空間に限定する必要があります。しかし、サービスのエコシステムは、すべての認証されたリソースへのサービスを展開するために使用される、すべてのシステムセットで構成されています。

ハードウェアの製造とパーソナライズ工程において、企業で展開されるリソースとして*認証できる*ようにハードウェアを適切に構成する*必要がある*ため、このシステムセットにはシステムによってまだプロビジョニングされていないエンドポイントが含まれます。

そのため、コミュニケーションモデルは以下の項目を提供する*必要があります*。

- 相互認証
- 機密性
- 完全性

これらの項目を効果的に提供するには、コミュニケーションモデルは以下の項目も提供する必要があります。

- 一元管理された信頼の基点、または分散化された信頼の基点
- ID のプロビジョニングおよび失効
- 完ぺき前方秘匿性（Perfect　Forward Secrecy）

コミュニケーションモデルの各エンティティがピアと同じ組織によって認証されていることを確認するために、信頼の基点を使用する必要があります。これは、すべてのエンティティが 1 つの中央組織によってプロビジョニングおよび認証されていることを確認するのに役立ちます。この信頼の基点を確保するために使用する技術は、（TLS 証明書のように）一元化されるか、または（IBM/Samsung の ADEPT プロジェクト、Tilepay など、ビットコインのブロックチェーンに基づく IoT モデルのように）分散される場合があります。いずれの場合も、1 つの中央組織がモデルの*所有者*となり、プロビジョニングシステムを保護する必要があります。

セキュリティ侵害された秘密鍵や ID を最小限の労力でシステムから削除できることを保証するために、プロビジョニングと失効は通信モデルの一部である必要があります。オンライン証明書セキュリティプロトコル（OCSP）などの技術は、この作業に役立ちます。

通信プロトコルは、*過去*から遡って通信へのセキュリティ侵害リスクを軽減する技術を導入する*必要があります*。これは、通信の秘密鍵を交換するために使用されるエフェメラル非対称暗号鍵を作成することによって行われます。証明書がセキュリティ侵害を受けた場合、エフェメラル秘密鍵は存在しません。これは、暗号化されたメッセージを長時間保存することで、証明書のプライベート秘密鍵がセキュリティ侵害を受けたり、公開された場合に、攻撃者がメッセージを解読できないようにします。

通信のセキュリティにおける課題は、技術の実装と寿命です。権威のある機関によって認められている、信頼性の高い暗号化アルゴリズムを採用することで、障害が発生するリスクを軽減できます。

技術機関によって設計・承認されたライブラリとアルゴリズムを実装する*必要があります*。アルゴリズムのカスタム実装を*行ってはいけません*。これにより、技術チームの負担を軽減するだけでなく、不適切に設計または実装されたシステムによってアルゴリズムの暗号化が弱体化するリスクを軽減できます。

以下の組織が提供する参考資料を参照することをお勧めします。

- CafeSoft Apache Mutual Authentication How-To Guide：
  - http://www.cafesoft.com/products/cams/ps/docs32/admin/ConfiguringApache2ForSSLTLSMutualAuthentication.html

6.3.1 リスク

通信セキュリティは、IoT の基盤となっています。通信セキュリティがない場合、組み込みデバイスが適切なバックエンドサービスと通信していることは保証されません。通信セキュリティは、テレマティクス、医療機器、産業用制御システムなどの装置の誘導、構成およびコマンドの送信を行う重要なサービスに欠かせません。通信セキュリティがない場合、適切なエンドポイントにコマンドが送信されることは保証されません。意図しているピアとメッセージを確実に送受信するために、通信セキュリティを確保しましょう。

6.4 ネットワーク認証サービスの使用

パートナーであるネットワーク事業者は、独自のトークンを使用してユーザーの認証を行うことができます。ネットワーク事業者の UICC にあるこれらのトークンは、ネットワークレイヤに関するユーザー認証を行いますが、アプリケーションレイヤでは、必ずしもユーザー認証を行うわけではありません。以下の技術を使用して、ネットワーク認証を容易に行うことができます。

- Generic Bootstrap Architecture（3GPP TS 33.220）
- M2M SM（ETSI TS 102 921）

認証技術が、認証のためにアプリケーションレイヤに意味を与えるかどうかを検証します。トークンをセキュリティストアとして使用できる場合、トークンを使用して TCB を構築する物理的エンドポイントに対する認証レイヤとして、デバイスを使用できるかどうかを確認します。

多くのネットワーク事業者はネットワークベースの認証を実施しますが、ユーザーまたはエンドポイントを認証するためにこの API へのアクセス権限を取得することは、極めて新しい技術です。提携しているネットワーク事業者が、この技術に関して十分な経験を有しているかどうかを確認しましょう。その場合、複数の技術ではなく 1 つのセキュリティストア技術を利用する方が容易であるため、この技術をネットワークレイヤの認証トークンとしてだけでなく、他の用途にも利用することを検討しましょう。

6.4.1 リスク

ネットワーク認証サービスに UICC などのトラストアンカーが含まれている場合、これらのサービスを利用 *せずに* アプリケーションレイヤのセキュリティを確保しようとすると、アプリケーションが確実にユーザー認証を行う機能が制限され、基盤となるエンドポイントのプラットフォームのコストが増大します。これにより展開コストが増大し、組織が利用できるネットワーク事業者からの情報も減少します。

6.5 可能であればサーバープロビジョニング

サーバーのプロビジョニングには、本番環境でのサーバーの定義、構成、パーソナライズ、展開が含まれます。サービスの観点からみると、プロビジョニングのプロセスは、攻撃される危険性がある環境において、サーバーのセキュリティが強化され、展開の準備が整っていることを保証します。

サーバーは、クラウドインフラストラクチャ、専用ホスティングプロバイダー、または企業の個人的なラックスペースで展開されているかどうかにかかわらず、内部と外部の両方の脅威を受ける可能性があります。サービスインフラストラクチャにサーバーを展開する前に、攻撃に対するサーバーのセキュリティを強化する必要があります。

これを実現するには、周囲の環境にアクセスできるようにする必要があるサービスを特定します。サーバーの環境がパブリックかプライベートかを設定し、サーバーセキュリティのコンテキストにおける意味を定義します。サーバー上で実行する各サービスが一般のアクセスを許可するかどうか、または認証されたクライアントのみサービスへの接続を許可するかどうかを決定します。

サーバー上で実行するオペレーティングシステムのライフサイクルを評価します。セキュリティパッチが迅速に展開され、本番環境におけるサーバーの正常な動作をサポートできるよう、ソフトウェアのアップデートを適切に管理する方法を決定します。一部のライブラリやアプリケーションのアップデートによって、意図しない副作用が生じる場合があるため、本番サービスでアップデートの失敗や予期しない問題が発生した場合のロールバックモデルを評価します。

最後に、プロビジョニングされたサーバーの廃止モデルを評価し、システムからアセットを削除する最も安全な方法を決定します。これには、異常なサービスまたはクライアントの動作を評価するために必要なシステムのログが含まれます。

この推奨事項は、組織がパッチ管理プロセスを行い、脆弱なサービスを特定してパッチを適用し、これらのパッチの実装が正常に行われたかを監視する必要があることを意味しています。

パッチ管理に関する以下の参考資料を確認してください。

- http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

6.5.1 リスク

サーバーのプロビジョニングは、IoT 環境の全体的なセキュリティに不可欠です。サーバーのプロビジョニングが行われない場合、サーバーのアーキテクチャに対する組織の管理能力は大幅に低下します。これは、アーキテクチャ仕様の不足によるセキュリティ上の欠陥につながる恐れがあります。仕様がない場合、組織は展開された技術が現在のベストプラクティスに準拠しているかどうかを確認できません。また、これらの技術の向上には、展開された各システムを調査し、展開されたアセット間の差分を評価する必要があります。これは非効率的であり、重要なセキュリティ更新プログラムを展開する際に大きな懸念材料となります。サービスの定義に一貫性とアーキテクチャがない場合、早急な対応が必要なシステムはどれかを容易に追跡する方法はなく、1 つずつ手動で確認しなければなりません。

6.6 更新モデルの定義

実行環境、アプリケーションイメージ、または TCB のアップデートは、容易ではありません。全体のプロセスを簡素化する次のサンプルモデルを考えてみましょう。

- 実行プラットフォームの各レイヤについて、新しいアプリケーションイメージに一意の URL などのネットワークリソースを定義する
- それぞれの特定のレイヤに対して署名鍵を生成する
- 各レイヤの認証された新しいバージョンに対して、該当するレイヤのイメージを生成する
- レイヤイメージに、イメージを記述したメタデータ（バージョン、タイムスタンプ、ID など）を含める
- 署名鍵を使用してレイヤイメージに署名する
- 一意のネットワークリソースまたはアップデートサービスを通じて、イメージ、署名および公開鍵を利用可能にする

新しいシステムを展開する場合、新しいシステムは以下を行う必要があります。

- 各レイヤに対して、
  - 展開するバージョンを取得する
  - イメージを暗号方式で検証する
  - システム上にイメージレイヤを展開する

任意のアプリケーションレイヤに、非公開の秘密鍵を保存してはいけません。各システムの展開時に、該当するシステムをパーソナライズするために秘密鍵を動的にプロビジョニングする必要があります。システムの寿命期間にかかわらず、システムの廃止時にこれらの ID も無効にする必要があります。

この推奨事項は、インフラストラクチャ内のサービスや技術を維持するために、パッチ管理プロセスを実施する必要があることを意味しています。

詳細については、以下のドキュメントを確認してください。

- http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

6.6.1 リスク

適切に定義された更新モデルがない場合、更新手続きの悪用によってサービスとアプリケーションがセキュリティ侵害を受ける恐れがあります。攻撃者は、更新プロセスにカスタムアプリケーションを注入し、クラウドシステムや他のサーバー上で独自のソフトウェアを展開できる可能性があります。通信セキュリティのインフラストラクチャが保護されていない場合、ドメインネームサービス（DNS）などのネットワークサービスを操作することによってこの攻撃を容易に行うことができます。ボーダーゲートウェイプロトコル（BGP）攻撃など、ルーティングに対するより高度な攻撃は、保護されていないサービスへのセキュリティ侵害を引き起こすために、過去に何度も実行されています。

6.7 流出データに対する違反ポリシーの定義

データ区分のためのポリシーおよび手順を定義するだけでは、十分ではありません。データがパートナーによって公開されているかどうかを特定するためのモデルが必要です。組織は、ユーザーのデータとプライバシーを保護するために策定された技術的管理やポリシーに違反している事業慣行に、パートナーが関与しているかどうかを評価するための計画を策定する必要があります。

そのためには、技術チームはユーザーデータではなく、*セキュリティ区分*に適用される監視とロギング技術を定義しなければなりません。これにより、監査証跡を情報だけでなく情報*区分*にも適用できます。これは、ユーザー情報が流出した場合に組織が防御策を講じるのに役立ちます。組織はセキュリティ区分だけでなく、これらの区分を管理し、ポリシーに従ってデータを管理、保存および伝達するための技術的管理を提示できるようになります。

組織は、パートナーがセキュリティ区分の規則にいつ違反したかを証明するために、監視とロギング技術を利用することが推奨されます。その場合経営陣は、パートナーに罰金、契約解除などの罰則を与えるべきかどうかを決定する必要があります。

6.7.1 リスク

違反ポリシーがない場合、第三者によって流出されたデータに対する法的責任から組織を保護する法的な保護措置はほとんどありません。企業が流出したデータの管理元である場合、第三者はデータを失う可能性がありますが、企業はパートナーにデータを渡したことに対する責任を負います。

違反ポリシーは、パートナーが提供されたデータに対して十分なセキュリティレベルを維持する必要があることを保証します。セキュリティ違反が発生した場合、IoT サービス提供者は自社のセキュリティ要件に従う限り、セキュリティ違反に対する責任を免除されます。これは、パートナーがポリシーを遵守するか否かにかかっています。

モデルを導入することにより、組織が厳格なセキュリティポリシーや手順を遵守することで自社の法的責任を軽減できることを保証するために、法務および保険チームはこれらのポリシーを検証する必要があります。製品やサービスの性質上、一部の企業は、規制、法令、その他の命令により例外的な扱いとなる場合があります。

6.8 サービスエコシステムによる強制認証

ユーザ-インターフェイスは、ユーザ-を直接認証し*てはいけません*。システムは、一元的に利用できるサービスを使用して常にユーザーを認証できる必要があります。この規則の唯一の例外は、モバイルデバイス上でアプリケーションを実行している場合、ローカルのパスコードで保護されることです。このパスコードは、ローカルのアプリケーションにアクセスするのに使用される場合があります。しかし、リモートサービスやリソースへのアクセスは、別の認証トークンで検証する必要があります。

この認証方法を利用するリスクに関する十分な情報がユーザーに提供される場合、技術チームは可用性の理由から、これらの 2 つの認証方法を 1 つに統合する場合があります。この手法により、リモートサービスで稼働する認証トークンを含むローカルデータベースの暗号を解読するために、認証されたユーザーのローカルアプリケーションのパスワードを使用できます。ほとんどのユーザーにとって、このマルチステップ認証モデルは十分に機能するでしょう。

しかし、中央の認証サービスがまず最初にローカルアプリケーションのユーザーを認証し、認証トークンの利用方法と期間を定めるポリシーと手順を策定する必要があります。また、メトリクスを収集し、ユーザーが同じト

ークンを使用して別のコンピューティングプラットフォームに移動したかどうか、または、ユーザーが同じトークンを使用して短時間で別の場所に移動したかどうかを特定する必要があります。動作の種類と速度に応じて、これらのメトリクスはトークンのセキュリティ侵害の可能性を示すことができます。セキュリティ侵害が発覚した場合、トークンを無効にするとともに、マルチファクター認証によってユーザーに強制的に再ログインさせる必要があります。

6.8.1 リスク

エンドポイントシステムにおける悪用リスクにより、アーキテクチャの保護方法にかかわらず、バックエンドシステムによる確認がないユーザー認証は、常に信頼性に欠けます。これは、ユーザーが自分の資格情報を更新していないか、複数のデバイス間で資格情報を分割できることを前提としています。これは非効率的であり、セキュリティ侵害されたデバイスがユーザーの資格情報の古いバージョンを使用している場合、欠陥が生じる恐れがあります。

6.9 入力検証の実行

エンドポイント、ユーザーまたは自称ユーザーから取得したすべてのデータについて、異常分析を行う必要があります。攻撃者にとって最も容易な攻撃経路は、ユーザーインターフェイスを構成するサービスのウェブアプリケーションの入力値を悪用することです。これは、この技術では、ユーザーごとに異なる場所、エンコーディングおよびその他のパラメータに基づいて情報を動的にレンダリングする必要があるためです。熟練したユーザーがエンコードの特定の属性を操作することで、処理サブシステムの異なるレイヤで攻撃者に有利な予期しない副作用を引き起こす可能性があります。

実行しやすい攻撃の一例には、高レベルの言語で文字列として処理されたメッセージに null バイトをエンコードすることが挙げられます。一部の高レベルの言語は、区切り文字としてではなくバイナリ文字列の一部として null バイトを受け入れます。このバイナリ文字列が低レベルのライブラリに渡された場合、組み込み null バイトは文字列の区切り文字として解釈されるため、アプリケーションによる文字列の解釈とは全く異なるものを意味する文字列を切り捨てます。以前は、これは特定のユーザーが利用できなかったファイルシステムリソースにアクセスする賢明な方法でした。

悪意のある入力値の形態は無数に存在しますが、技術者はすべての可能性をテストする必要はありません。必要なプロセスは極めてシンプルです。

- 内部でのデータの使用方法を特定する

- 内部利用のモデルに準拠するエンコーディングと文字に関するポリシーを策定する
- このポリシーに基づいてデータを分析する API を設計する
- データがモデルに準拠していない場合に例外を発生させる
- 悪意のある動作の検出をサポートするために、セッションに関するメタデータとともにイベントを内部で記録する

システム内に保存されているすべてのデータを最初に処理し、静的モデルに組み込む必要があります。これを行う効果的な手法は、base64 アルゴリズムによってすべてのデータをエンコードしてから、データベースに配置する方法です。これにより、データがデータベースを操作できないようにします。

6.9.1 リスク

入力の検証が採用されていないシステムは、SQL インジェクション（SQLi）などの OWASP Top 10 に記載されている問題を含む様々な攻撃や、リモートコード実行攻撃を受ける恐れがあります。潜在的な悪用の範囲は多岐にわたるため、本文書ではすべてのリスクを定量化できません。入力検証は、エンドポイント上で実行中のクラウドサービスやアプリケーションにとって、安全なアプリケーションの重要な属性となっています。

6.10 出力フィルタリングの実行

出力フィルタは、入力検証を補完します。このプロセスは悪意のある操作から表示レイヤを保護するだけでなく、システムが特権と見なされる必要があるユーザーに情報をレンダリングできないようにします。

前者では、表示レイヤでレンダリングされるすべてのデータがサービスレイヤを離れる前に、これらのデータを評価する必要があります。これにより、JSON メッセージやエンコードされた JavaScript における表示レイヤにエンコードされたデータには、エンコードされている場合、データの表示を破壊または無効にする恐れがあるフォーマットが含まれなくなります。つまり、レンダリングによって表示モデルを破壊する恐れがある、システム上に保存されている文字が、想定外の方法で表示を改ざんしないようにフィルタリングかエンコードする必要があります。

この問題を修正するには、制限された文字をフィルタリングするか、すべての文字のエンコードを強制することで、これらの文字の表示が GUI を改ざんしない（文字がレンダリングエンジンによって制御コードとして解釈されない）、またはメッセージを表示しないようにする必要があります。どちらも有効な方法ですが、特定のアプリケーションについてはどちらかがより適している場合があります。メッセージフォーラムの例を見ると、攻撃者

がスクリプトを配置し、ユーザーがそれに気づかずにスクリプトをコピーおよび実行する恐れがあります。そのため、表示レイヤに HTML や他のスクリプトを挿入しない方法で情報をレンダリングするのではなく、悪意のある情報を取り除き、他のユーザーに影響を与えないようにする必要があります。

データをユーザーに再表示するべきでない場合、これは攻撃者によって保存およびレンダリングされたデータとは無関係です。この問題は、管理者や技術者が管理するべき、一般利用に適していないデータの開示に関連しています。例えば、情報処理中に内部エラーが発生した場合、そのエラーを完全なデバッグデータとともにユーザーにレンダリングしてはいけません。これにより、ユーザーはバグを特定・利用して、アプリケーションの脆弱性を悪用できる可能性があります。この情報を内部に記録し、ユーザーがバグを悪用するために十分なコンテキストを提供しなかった場合は一般的なエラーが発生するようにしておく必要があります。ユーザーがバグを複製できるとしても、攻撃手法への対策を強化したアプリケーションからの出力を評価できないはずです。

6.10.1 リスク

出力検証は、IoT セキュリティの重要な特徴です。システムが出力検証を実行しない場合、重要なユーザーデータやプライバシーデータ、診断データ、詳細なエラーメッセージなどが流出する恐れがあります。これらのメッセージは、ユーザー情報を公開するか、ネットワーク化されたサービスに対して攻撃を確実に行うために利用される可能性があります。

6.11 強力なパスワードポリシーの施行

すべての認証システムは、ユーザー認証にパスワードが必要となる場合に、強力なパスワードを適用する必要があります。パスワードの複雑性は、情報セキュリティの研究者、技術者、経営陣が常に苦慮している問題です。経営陣は多くの場合、ユーザーが自分のパスワードを簡単に覚えることができるようにしたいと考えています。技術者は、特に表示レイヤの設計者のために、インターフェイスの複雑性を軽減する必要があります。情報セキュリティの研究者は攻撃者の技術を過大評価し、特定の技術の複雑性を過度に高めてしまうことが多いです。

解決の糸口は、各グループのすべての要件のどこかにあります。パスワードを強制的に長くする必要がありますが、シンプルでなければなりません。8 文字のパスワードが標準となっていますが、一部のシステムでは 6 文字を許容しています。パスワードの長さは、最新のベストプラクティスの標準に基づいて検討する必要がありますが、8 文字をはるかに超える可能性があります。長いパスワードの長さを適用することによって、複雑性の

要件が軽減されます。様々な文字セットの奇妙な組み合わせを適用する代わりに、ユーザーは文章を覚えておくことができます。空白、大文字と小文字、数字、句読点を使用できるため、攻撃者がブルートフォース攻撃を仕掛ける難易度が格段に上がります。

攻撃者は、主に次の 4 つの方法でパスワード攻撃を行います。

- パスワードのデータベースを盗み、個々のパスワードを解読する
- アプリケーションの認証サービスにブルートフォース攻撃を仕掛ける
- マルウェアをインストールする
- ハードコードされた、またはデフォルトのパスワードを使用する

長いパスワードの強制は、初期のリスクの軽減に役立ちます。しかし、サービスのエコシステムでのセキュリティは、さらに多くのメリットがあります。攻撃者は、最初の段階でパスワードのデータベースを取得できないはずです。

アプリケーションのパスワードに対するブルートフォース攻撃は、攻撃者にとってパスワードを悪用する最も効果的な方法です。適切に設計された認証サービスによって、このリスクは大幅に軽減されます。1 つの悪意のあるパスワードが推測されている場合、システムは推測間に必要な遅延時間を自動的に増加させる必要があります。次に、試行の合計数を制限するためのしきい値を定義する必要があります。攻撃者がこのしきい値に達した場合、アカウントはロックされ、2 ファクタ認証または別のモデルを使用して、ユーザーがアカウントのロック解除および検証できるようにする必要があります。このタイプのセキュリティは、ネットワークベースの攻撃によるメリットを軽減させます。

最後に、クライアントシステム上のマルウェアは、コンピューティングプラットフォームか、適切なアンチマルウェア技術をインストールしているユーザーによって対処されなければなりません。これは一般的に、アプリケーションによって保護できるものではありません。アプリケーションがこのリスクに対してできることはほとんどなく、マルウェアが攻撃者にとって*唯一の実行可能な攻撃*である場合、アプリケーションエンジニアは、2FA 認証を適用することで認証システムに対するパスワード攻撃の脅威を十分軽減できます。

しかし、この推奨事項を実施するメリットは*それほどありません*。これは、パスワード認証への攻撃リスクを減らすために使用される技術が何であれ、パスワードは基本的に無形のリソースであるからです。パスワードは、一個人のみが取得できる物理的トークンではありません。コンピュータシステム間や視覚的な観察を通じてコピーできる、抽象的なオブジェクトなのです。そのため、パスワードは特定のユーザーを適切に指名しない、非

常に脆弱な認証ソースであり、パスワード自体が弱点であるため、パスワードを使用するすべての技術は、パスワードがその性質上伴うリスクを受けることになります。

パスワードは、システム上でハードコードすべきではありません。エンドポイントの場合は、一意の暗号鍵を生成する必要があります。エンドポイントのプロビジョニングの詳細については、エンドポイントのドキュメントを参照してください。サービスおよびユーザーインターフェイスの場合は、ユーザー登録の際にユーザーがパスワードを定義する必要があります。その場合、パスワードは強力なパスワードのセキュリティ要件を遵守する必要があります。ユーザーが、デフォルト、脆弱、または不適切に設計されたパスワードを使用できないようにする必要があります。

ユーザーがいつでも自分のパスワードを変更できるようにしてください。ユーザーが自分のパスワードを変更する場合に、強力な認証要件と通信セキュリティを適用してください。可能な場合は、ユーザーがパスワードを変更する前に、2 ファクタ認証（2FA）によってユーザーID を確認してください。ユーザーがシステムに新しいパスワードを送信する場合は、ユーザーに元のパスワードを再入力することを常に義務付けてください。これにより、別のユーザーがロックを解除したノートパソコンや盗んだウェブアプリケーションのセッショントークンを利用して、開いているウェブアプリケーションを不正利用していないことを保証できます。

6.11.1 リスク

適切なパスワード管理を実施していないシステムでは、攻撃者がシステム上のユーザーのパスワードを容易に推測できるリスクがあります。

6.12 アプリケーションレイヤの認証と承認の定義

組織の信頼の基点（Root of Trust）とそのサービスは、ネットワーク通信レイヤ、ユーザー、管理者を保護する認証技術を定義します。パートナーの認証技術は個別に構成する必要があります。これらのエンティティの通信チャネルは組織の信頼の基点（Root of Trust）によって保護されますが、その*アクション*と *ID* は別のシステムによって認証する必要があります。

一般的に、このアプリケーションレイヤの認証は、同じサービスによってサポートされます。しかし、情報は別のリソースから収集されます。例えば、それは個別のデータベース内のユーザーおよび管理者の認証データを配置することをお勧めします。これは、アプリケーションレイヤを通じてデータベースを操作する方法がある場合（SQL インジェクションなど）、攻撃者は、ユーザーのデータベースを介して横断的にのみ移動できるようにします。攻撃者は縦断的に移動できないため、データベースのセキュリティ侵害を行うことができず、管理者に権限を昇格できません。これにより、組織のセキュリティは大幅に向上します。

可能であれば、次の項目に対して別のストレージシステムを定義します。

- エンドポイント ID
- ユーザー
- 管理者の資格情報
- パートナー

これは、アプリケーションとインフラストラクチャに対して論理的な職務の分担を行いますが、組織の信頼の基点（Root of Trust）サービスによって同じ認証 API 内に限定されます。

以下の組織が提供する参考資料を参照することをお勧めします。

- OAuth 2.0 [8]
- OpenID Foundation [9]
- GSMA Mobile Connect [10]

6.12.1 リスク

アプリケーションレイヤの認証と承認を適用する方法がない場合、システムは、ユーザーが主張するアクションが、ユーザーによって実際に認証されているかどうかを確認できません。この推奨事項を実行することで、各アクションが認証されたユーザーと認証を追跡可能であることが保証されます。これらのメトリクスは保存でき、セキュリティ侵害が疑われる場合に確認できます。これらの手順がない場合、悪用のリスクを最小限に抑えるための対策はありません。

6.13 デフォルトオープンまたはフェイルオープンのファイアウォールのルールとシステム強化

一部のサービスインフラストラクチャ環境では、入力と出力の保護機構がデフォルトで設定されていません。つまり、技術者は手動でファイアウォールやネットワークトラフィックのルールセットを展開しなければなりません。これらのルールは、サービスが一般に展開される前に、インフラストラクチャに設定する必要があります。

しかし、これらの技術ではサービスインフラストラクチャを保護するのに十分でない場合があります。ファイアウォールやその他のネットワークトラフィックの保護システムに、障害が発生することがあります。これらのシステムに障害が発生した場合、システムの起動に失敗することが多いです。この問題が発生する理由は、他のコンピューティング環境に対応したトラフィックは、IoT サービス提供者のトラフィックを伴うインフラストラクチャを介してルーティングされるため、システムに障害が発生した場合でもトラフィックは正常に動作できる必要があります。そのため、トラフィックは突然停止できません。その結果、多くの場合システムが起動に失敗し、多くのサービスが動作を続行できなくなります。

技術チームはオペレーティングシステムを強化し、障害が発生したインフラストラクチャにより、致命的なセキュリティイベントが発生しないようにする必要があります。つまり、既存のサービスインフラストラクチャに対してより多くの接続を確立できるようにします。

例えば、非表示のサービスをファイアウォールなどの技術の背後に配置すべきではありません。バーチャルプライベートネットワーク（VPN）や他の高セキュリティ保護を使用して、攻撃者からサービスを保護できます。

ソフトウェアのファイアウォールは、精通した攻撃者によって操作されるリスクを伴うことに注意してください。ソフトウェアのファイアウォールが使用されている場合、不適切に強化されたサーバーインフラストラクチャが、攻撃者に操作される可能性があります。すなわち、サーバー上で稼働しているパブリックサービスに不必要な権限（スーパーユーザー権限など）があり、セキュリティ侵害を受けた場合、攻撃者はソフトウェアのファイアウォールを無効化できる可能性が高いです。そのため、技術チームは、選択されたアーキテクチャに対してソフトウェアのファイアウォールのリスクが高すぎるかどうかを評価する必要があります。

6.13.1 リスク

ネットワークのトラフィックセキュリティシステム上の障害を修正するための戦略がない場合、セキュリティ戦略を強化する標準的なサービスで容易に阻止できる、不要な攻撃の対象となる場合があります。

6.14 通信プライバシーモデルの評価

通信のプライバシーは、アプリケーションのプライバシー（上記参照）や通信情報セキュリティとは少し異なります。プライバシーは、第三者がデータを*効果的に読み取る*、または*傍受する能力に関して評価されますが*、機密性と完全性は、通信プライバシー全体を対象としているわけではありません。

通信プライバシーに影響を及ぼすその他の問題は、次のとおりです。

- 各メッセージの暗号化の一意性
- 伝送パターン
- プレーンテキスト形式のメタデータ
- ハードウェアアドレスまたは寄与シリアル番号

各メッセージは機密情報であり、検証可能な完全性を有しているはずですが、暗号化を一意にする必要があります。攻撃者が予測可能なイベントへの応答で特定のメッセージが送信された場合、攻撃者によって暗号化が一意ではない応答が返信される場合があります。攻撃者が有益なメッセージを取得および返信できないようにするために、各メッセージは一意である必要があります。

伝送パターンは、攻撃者が特定のユーザーを識別したり、動作を特定の寄与アクションと一致させることができるようにする恐れがあります。例えば、ユーザ-が特定の物理ゾーンを入力するとメッセージを表示する技術は、ネットワークを介して伝送されるこれらのメッセージを受信できる「スニッファ」によって採取される場合があります。これは直観的でない場合がありますが、攻撃者が物理的位置にいる人物と所在地を特定できる場合、組織は潜在的な法的責任を負うことになります。ネットワークパターンを評価して、攻撃者が伝送パターンを実行可能データに変換できる簡単な方法があるかどうかを確認する必要があります。

インテリジェンスサービスでは、暗号化されたデータへの保証や他の合法的なアクセスを必要とせずに、メッセージングシステムのコンテキストを評価するために、長きにわたってメタデータを使用してきました。多くの場合、メタデータは、組織が実行可能なインテリジェンスを作成するための十分な情報です。しかし、現在はホビースト、犯罪組織、好奇心旺盛なユーザーが、追跡やその他の悪意のある目的のためにメタデータを使用することができます。そのため、第三者が利用できるメタデータの量を制限することが、これまで以上に重要です。可能な場合は、メタデータ量を、通信ピアがメッセージが目的に適したものであるかどうかを検証できるだけの十分な情報量のみに限定してください。

この推奨事項に従って、通信モジュールのハードウェアアドレスと一意のシリアル番号は、できる限り保護または無作為化する必要があります。例えば、Apple は Wi-Fi アクセスポイントをプロービングするために、iOS

モデルを変更しました。静的なハードウェアアドレスを使用するのではなく、自社技術が無作為化したハードウェアアドレスを使用するように変更し、第三者が Wi-Fi アクティブスキャンに基づいてユーザーの位置を追跡するリスクを軽減しました。IoT 技術も同様に機能しますが、多くの通信技術がこの問題の影響を受けます。一部の技術は、セルラーなどのランダムなハードウェアアドレスを生成できません。しかし、802.15.4、Wi-Fi、Bluetooth などの他の技術は、ファームウェアの機能に応じてランダムなハードウェアアドレスを生成できます。

6.14.1 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。

7 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。

7.1 アプリケーション実行環境の定義

アプリケーションの実行環境について、以下の項目にご注意ください。

- 使用されているプログラミング言語は、セキュリティと直接関係がある場合があります。
  - PHP やルビーのような言語には、セキュリティ上の問題がある場合がある
  - GoLang や Erlang などの言語は、リスクを軽減できる
- サードパーティ製のライブラリに伴うリスクの監視、管理および監査を行う必要がある
  - 一部のライブラリは、適切に管理されていない
  - 一部のライブラリは、セキュリティ上の欠陥について監査を受けたことがない
  - 一部のライブラリは、セキュリティ上の欠陥が判明した旧式の依存関係を必要とする
- 常に非特権ユーザーとしてアプリケーションを実行します。
  - アプリケーションに特権リソースが必要な場合、権限を降格して完全なアプリケーションを実行する前に、リソースをプロビジョニングするラッパーを使用する
- 適切に定義された TCB とブートストラップのモデルを使用します。
  - 適切に定義された環境を持つアプリケーションは、より*信頼性が高く安全*です

以下の組織が提供する参考資料を参照することをお勧めします。

- OWASP [5]

7.1.1 リスク

安全なアーキテクチャが展開されているアプリケーションは、攻撃元を簡単に追跡することができないセキュリティ侵害を受ける恐れがあります。サービスとアプリケーションのセキュリティ侵害を引き起こすツールと技法

は、過去 10 年間で高度化が進んでいます。Metasploit などの一部のオープンソース技術は、カスタムしたエクスプロイトを開発し、攻撃対象のプラットフォームに統合することで、ステルス攻撃の成功率を高めることができます。

安全なアプリケーションの実行環境は、アプリケーションの実行および通信方法、ランタイム時に使用される技術の種類を保護することで、このリスクを阻止できます。これらの属性は、セキュリティ侵害が発生する可能性を軽減するだけでなく、悪用される脆弱性を追跡・診断するためのトレーサビリティと重要なログ機能を追加できます。

7.2 パートナーの強化モニタリングサービスの利用

パートナーがモバイルネットワーク事業者の場合、パートナーが監視サービスを提供できるのかどうか特定します。一部のネットワーク事業者は、ネットワーク経由で通信するエンドポイントの動作を分析することができます。この分析を行うことができる事業者は、異常および悪意のある動作を示すメトリクスを評価する経験を有しています。

これにより、IoT 企業は特定のユーザーまたはエンドポイントが、脅威、または攻撃者によってセキュリティ侵害されているかどうかを、より迅速に特定することができます。そのため、企業は自社のインフラストラクチャの他の領域に対する攻撃に対して先手を打つために、より効果的に対応できます。

このサービスの複雑性は、ネットワーク事業者がインテリジェンスを迅速に提供できるかどうかに応じて異なります。攻撃者が IoT 企業を攻撃した後にのみネットワーク事業者がインテリジェンスを提供できる場合、知性のみを提供している場合は、IoT 企業のインフラストラクチャに導入されている監視およびロギングシステムは、この動作を検出できる必要があります。しかし、ネットワーク事業者がネットワークレイヤ上の悪意のある動作を企業に通知するとともに、異常なネットワークトラフィックを流している加入者を特定できる場合、企業は該当するユーザーのトラフィックを遮断して、IoT エコシステムへの流出を制限できる場合があります。

7.2.1 リスク

IoT サービス提供者が依存しており、かつ IoT サービス提供者が監視できない技術があります。このような技術の 1 つは、エンドポイントをサービスとネットワークのエコシステムに接続する通信ネットワークです。IoT サービス提供者はサービスを監視せずに、ネットワーク内で発生するイベントを特定できません。そのため、アプリケーションレベルのユーザーA がサービスを攻撃しようとする場合、組織は、エンドポイント B が通信ネットワーク

に接続されているユニットであることを特定できません。この情報の欠落は深刻な問題であり、組織はユーザーA ではなくセキュリティ侵害を受けたエンドポイント B が攻撃元であると誤解する可能性があります。

7.3 セルラー接続に対するプライベート APN の利用

アクセスポイント名（APN）は、ワイヤレスネットワークでインターネットに接続するセルラー通信コンポーネントです。エンドポイントが通信する必要があるセルラーエンドポイント装置とサービスインフラストラクチャの間にある、バーチャルプライベートネットワーク（VPN）が、実質的にこの役割を果たしています。プライベート APN（セキュア APN とも呼ばれる）は、複数の望ましい制御を実装するために、セキュリティが強化されている APN バージョンです。

- 認証されたクライアントにのみアクセスを制限
- ファイアウォール
- エンドポイント間の通信を強制的に無効化
- 異常検出のための監視サービス
- オプションのセキュリティや監視サービス

APN へのアクセスを制限することで、組織は認証されたエンドポイントのみが、APN を通じて利用できるサービスインフラストラクチャに接続できるようにします。これは、悪意のある、または無作為の無線クライアントが、APN やアクセス制限サービスに接続するリスクを軽減します。さらに、組織は異常動作を行っているクライアントを特定し、不適切な動作を特定のハードウェアやユーザーと関連付けることができます。

ファイアウォールは、クライアント側（エンドポイントのエコシステム）とサービス側（サービスのエコシステム）から APN に追加されたエンティティに対する、承認されていないチャネルを使用した通信からの接続を制限します。これは、エンドポイントがインターネットに接続するチャネルとして APN を悪用する能力を制限し、承認された特定のサービスセットへのトラフィックを遮断します。

エンドポイントの通信を制限することで、悪意のあるエンドポイントがワイドエリアネットワークとして APN を使用することで、他のエンドポイントに攻撃できないようにします。代わりに、すべての通信は、組織によって承認されたサービスを介してピボットする必要があります。希望する場合は、組織はポイント間の通信を完全に禁止することができます。

監視サービスは、既存のクラウドサービスやサービスインフラストラクチャの監視について、組織が実施したセキュリティの改善を強化します。既存の監視サービスを、APN、ネットワーク事業者が提供するネットワーク監

視技術と組み合わせることで、組織はより簡単に異常動作の原因を突き止めることができます。これにより、組織はエンドポイントやサービスインフラストラクチャに対して発生するインシデントを、より徹底的に調査できます。例えば、アプリケーションレイヤが、ユーザーA がセキュリティ侵害を受けた可能性を示しているものの、ユーザーB の装置が APN への接続認証を行っている場合、組織は APN の監視サービスを使用して、ユーザーB がユーザーA のセキュリティ侵害を引き起こしたか、別の攻撃者がユーザーA と B の両方を攻撃したかを特定できます。

ネットワーク事業者は、上記のサービスをサポートする追加サービスを提供しています。これらのサービスは、ネットワーク上で悪意のある動作を行うユーザーに関するブラックリストを作成し、特定のユーザーを監視するとともに、異常が発生する可能性があるトラフィックのルートを変更できます。その他のオプションが利用できる場合もあります。ネットワーク事業者に、貴社に適したサービスの選定を依頼します。

これらのサービスをすべて同時に活用することは難しいようにみえますが、ネットワーク事業者と提携することでプロセスを簡略化し、企業の既存のインフラストラクチャにこれらのサービスをより簡単に統合することができます。データの効率的な利用は容易ではなく、技術チームは、合理的な方法でデータを処理および管理できる必要があります。一部のサービスでは、追加料金が発生する場合があります。貴社に最適な価格モデルとサービスを特定してください。

7.3.1 リスク

プライベート APN がない場合、エンドポイントデバイスは、APN 上の他のエンドポイントや、インターネット上の任意のサービスに直接接続するなど、ほとんどのサービスや技術に接続できます。そのため、セキュリティ侵害を受けたエンドポイントもインターネット上のほぼすべてのサービスに通信でき、エンドポイントがプロキシとして利用され、より安全なネットワークやサービスを攻撃する可能性があるため、この推奨事項を実施して、エンドポイントが悪意のある不正な接続を行う能力を制限する必要があります。エンドポイントを承認済みのサービスのみに接続させることは、企業にとっても、IoT エコシステム全体のセキュリティにとっても非常に有益です。

7.4 サードパーティーのデータ配布ポリシーの定義

セキュリティ区分を定義した後、データの種類に有効な区分を適用し、違反ポリシーを策定し、データ配布ポリシーを生成する必要があります。データ配布ポリシーは、技術コントロールによる情報の処理方法と、データにアクセスするためのアクセス権限が付与されているアプリケーションに情報を提供する方法について規定し

ています。権限モデルはデータ配布ポリシーの一部であり、ユーザーがデータのアクセス権限を細分化できる権限も付随しています。

データ配布ポリシーは非常に詳細に各項目を規定していますが、優れたポリシーを策定するための重要な要素があります。

- このデータの提供・取得に必要な相互認証レベル
- 必要なデータの機密性と完全性
- 企業がデータを保持するために必要な能力
- パートナーがデータを保持するために必要な能力
- データの保持が可能な場合の、データを保持できる期間
- データに適用するべきストレージのセキュリティレベル
- データに適用するべきアクセスのセキュリティ区分

7.4.1 リスク

データ配布ポリシーは、IoT サービス提供者と同等のセキュリティレベルを遵守していないパートナーに関するセキュリティ要件を策定します。IoT サービス提供者は、パートナーの内部サービスおよびネットワークに実装されているセキュリティを管理できないため、パートナーに提供されたデータが安全な方法で提供されることのみを規定できます。この定義がない場合、パートナーは安全でない構成を適用し、IoT サービス提供者がユーザーデータを管理しているにもかかわらず、データが攻撃者に流出する恐れがあります。通信チャネルの厳正なセキュリティ制御を適用することで、IoT サービス提供者は、データ保護のために最大限の努力を行っていることを証明できます。

7.5 サードパーティーのデータフィルターの構築

パートナーから広告などの動的に生成されたデータを取得するには、データの品質とセキュリティについて、ある程度推測する必要があります。技術チームは、推測に基づいてデータを表示レイヤに適用するのではなく、適切な手順に従って、サービスアプリケーションやパートナーからのデータの配布が適切な方法で行われており、悪意のある可能性があるコンテンツが含まれていないことを確認しなければなりません。

そのためには、技術チームは以下のモデルを検討する必要があります。

- データは、パートナーがデータモデルのために指定したフォーマットに適しているか

- データは適切に生成されているか
- データは、クライアントによって誤解される可能性があるポリモフィック型のオブジェクトを表しているか
- データは、クライアントによる表示レイヤのレンダリング方法に影響を与えるか
- データは、クライアントによる表示レイヤの*解釈方法*に影響を与えるか
- データは、ユーザーにセキュリティを弱体化するような動作を実行することを要求するか
- データは、クライアントの GUI のコンポーネント（パスワード入力フィールド）のスプーフィングまたはなりすましを行うか

承認されたモデルに適合していないデータを拒否します。このようなデータが検出された場合、データの出所と形式に関するできる限り多くのメトリクスとともに、管理者に直ちに通知します。可能であれば、安全なデータベースにサンプルを記録します。

7.5.1 リスク

第三者が動的に生成したデータには、意図的か否かにかかわらず、マルウェア、不適切なコンテンツ、またはその他の予期しないデータが含まれている可能性があります。サードパーティのサービスの定義に合わせた入力フィルターを使用しない場合、組織はマルウェアやその他の悪意のあるコンテンツがエンドユーザーに攻撃を仕掛けることを意図せず許可する恐れがあります。これにより、このようなデータの副作用が原因でシステムがセキュリティ侵害を受けたり、顧客喪失につながる可能性があります。

8 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。

8.1 Rowhammer や類似の攻撃

動的ランダム・アクセス・メモリ（DRAM）や静的ランダム・アクセス・メモリ（SRAM）などの現代の RAM テクノロジーの実装は、特定のメモリアクセスのシーケンスによって誘発される恐れがあるエラーに脆弱な場合があります。このタイプのエラーを悪用すると、メモリの予測可能な領域で特定のビットの改ざんが発生する可能性があります。この攻撃に成功すると、ソフトウェアによって表示される権限タイプを表すメモリ内のビットを変更することができます。

つまり、攻撃が計画通りに行われた場合、攻撃者は DRAM や SRAM などの実装においてハードウェアの欠陥を操作することで、あるユーザーを別のユーザーに権限を昇格できます。多くの DRAM と SRAM の実装を、この脆弱性によって悪用可能であることが実証されています。しかし、このバグを引き起こすことができるメモリのアクセスシーケンスを作成するには、ローカルシステム上でコードを実行できる必要があります。

また、サンドボックス化された GoLang、Python、Erlang などのランタイム言語を介して、リモートでこの動作を実行できる場合があります。しかし、これらの攻撃の精度はまだ文書化されていないため、攻撃として効果的に実行される可能性は極めて低いでしょう。

この攻撃は、ハードウェアレベルで解決する必要があります。しかし、技術者は、クライアントが仮想マシンやランタイム上で特定のサービスに対してのコードを実行できないようにすることで、悪用のリスクを軽減できます。この機能を制限することで、技術者は攻撃者によるこの攻撃に必要なメモリのアクセスシーケンスの作成を阻止できます。

8.1.1 リスク

この攻撃に対して十分な保護対策を講じていない場合、攻撃者はターゲットのホストに対して、リモートで権限昇格を行ったり、任意のコードを実行できる可能性があります。ただし、攻撃を成功させるには、ハードウェ

アやオペレーティングシステム、攻撃のベクトルなど、この攻撃の実行可能性を低下させる要素に関する深い知識が必要です。

8.2 仮想マシンのセキュリティ侵害

現代のサービスインフラストラクチャは、仮想マシンを利用してオンデマンドサービスを展開することが多いです。このモデルは極めて利便性が高く、展開しやすいことが実証されていますが、全体的なインフラストラクチャのセキュリティに関する問題があります。技術チームは、検討を重ねたアーキテクチャを導入することに成功するかもしれませんが、組織が仮想インフラストラクチャの管理と展開に失敗する恐れがあります。

仮想サーバー環境での展開に関する主な懸念材料の 1 つは、ホストへのセキュリティ侵害を行う能力や、サーバー（仮想ゲスト）が同じインフラストラクチャ上で稼働している他のゲストのデータを傍受する能力です。

これらの攻撃は、IoT サービス提供者によって検証されるべき問題ですが、防御対策には多くの技術と時間を要する場合が多いです。そのため、攻撃が実行される可能性はあるものの、発生する確率は極めて低くなります。しかし、サービスインフラストラクチャが十分に保護されていない場合、攻撃者が仮想マシンへの管理者アクセスを侵害できる可能性があります。このような侵害は、少ない技術で成功させることができます。

この問題に対処する方法の 1 つは、サーバーのプロビジョニングを活用することです。このプロセスは、各サーバーが一意の暗号鍵セットによってエンコードされていることを保証します。このプロセスを実行することで、1 つのサーバーへのセキュリティ侵害が他のサーバーに拡大することを阻止します。

8.2.1 リスク

この攻撃に対応できない場合、サービスインフラストラクチャが多くの攻撃に対して脆弱のままとなる恐れがあります。サービスインフラストラクチャからアクセス可能な鍵を使用したサーバーのなりすまし、データの不正取得、プライバシーの侵害、ユーザーのなりすましを実行できる可能性があります。

8.3 ユーザーがプライバシー属性を管理するための API の構築

すべてのユーザーは、サービスの API を開始て第三者に提供する情報を制御できる必要があります。情報をデータ別に分類し、セキュリティ区分を適用する必要があります。ユーザーは、アカウントのモデリングで使用されるデータの種類と区分を取得できる必要があります。また、ユーザーは、データの種類に制約を適用し、データへのアクセス権限をパートナーに付与する、または取り消すことができる必要があります。

これは、認証された API や、パートナーごとに「はい」または「いいえ」で許可を決定する GUI の形態を取り得ます。

8.3.1 リスク

ユーザーが IoT サービス提供者に提供するデータを制御できない場合、サービス提供者またはそのパートナーの 1 人に対してセキュリティ侵害が発生した場合、ユーザーデータが流出するリスクがあります。特定のユーザーは他のユーザーよりもはるかに高いリスクがあるため、各ユーザーが個人的なニーズに応じて、各自のプライバシー制限を調整できるようにする必要があります。このインターフェイスを利用可能にすることで、ユーザーが自分のプライバシー制限を確実に調整できるようにします。ユーザーは、自分のニーズに合わせて調整する必要があります。例えば、（TS-0003 を介した） oneM2M では、ユーザーはサービス提供者に対してプライバシー設定を調整できます。

8.4 フォールスネガティブとフォールスポジティブの評価モデルの定義

フォールスポジティブ分析は非常に複雑ですが、技術がフォールスポジティブを示す可能性が高いかどうかを識別する簡単な方法があります。それは、次の項目を評価することです。

- データ元は*信頼できるか*
- データ元は改ざんまたはスプーフィングされる可能性があるか
- データ元は、アナログのドメインからのものか
- データの信頼性は、複数の出所から裏付けることができるか
- 証拠となるデータ元は、同じエンドポイントシステム上に存在しているか
- 証拠となるデータ元は、容易に改ざんまたはスプーフィングできるか
- データ元を操作するために、簡単に利用できるツールはあるか
- データ元を操作するためには、どの程度の専門性やコストが必要か
- データ元に接続しているデバイスは、*信頼できるか*

これらの項目はすべて、データが*信頼できる*ものかどうかを評価するために使用できます。実世界に影響を与える重要な意思決定は、有害な影響を引き起こす恐れがあるため、この評価は非常に重要です。技術チームは、データの信頼性に関するモデルを作成し、重要な意思決定に関する各データ元に適用することが不可欠です。評価の結果データ元が信頼できないと判断された場合、最も安全かつ合理的な措置を講じる必要があります。

技術チームは、この意思決定を行う唯一の責任者 *ではありません*。経営陣、法務チーム、および保険のチームもまた、リスクがある状況における適切な対応に関する意思決定に関与する必要があります。技術者は、*検証可能かつ再現可能な方法*で、適切な意思決定プロセスを技術に反映させる必要があります。

組織全体が、有事における技術による対応方法を慎重に検討する必要があるため、このプロセスは極めて困難です。データの信頼性は、技術、特に組み込み技術への適用が難しい項目です。

8.4.1 リスク

フォールスポジティブに関する評価モデルがない場合、技術者はより深刻なイベントが発生している場合でも、問題のないイベントの分析に多くの時間を費やさなければならない場合があります。これにより、組織によって分析されたメトリクスが、本番環境で発生しているイベントの種類について明確な情報を提供しないリスクが高まります。ロギングや監視インフラストラクチャの価値が下がり、組織はコストが高いこれらのリソースを有効に活用できなくなります。

9 要約

要約すると、IoT 製品またはサービスのほぼすべてのセキュリティ上のリスクは、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が発生する前や発生した際にリスクを特定するための情報、セキュリティ関連のイベントに対応するためのポリシーや手順によって解決できます。IoT サービス提供者にとってどの高レベルセキュリティコンセプトが重要かを分析することで、「セキュリティに関するよくある質問」を確認できます。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 文書管理

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.12	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。 軽微な修正。	PSMC	Ian Smith GSMA
2.0	2017 年 9 月 29 日	oneM2M の参考資料の追加。	IoT Security Group	Rob Childs GSMA

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。