JC-STAR★1 開発ガイド

Version 1.0.0
2025/08/28

株式会社アットマークテクノ [https://www.atmark-techno.com]

Armadillo サイト [https://armadillo.atmark-techno.com]

JC-STAR★1 開発ガイド

株式会社アットマークテクノ

Version 1.0.0
2025/08/28

目次

図目次

JC-STAR★1 開発ガイド

表目次

1. はじめに

1.1. 背景

1.1.1. IoT 製品におけるセキュリティと JC-STAR

近年、 IoT 製品の数が急速に増加し、それと同時に IoT 製品を狙った攻撃も増加傾向にあり、 IoT 製品の脆弱性を狙ったサイバー脅威が高まってきています。こうしたことを踏まえて、諸外国では IoT 製品のセキュリティ対策に関する制度検討が進んでいます。すでに EU では、一定以上のセキュリティ機能が施されていない IoT 製品は EU 圏内で販売を禁止する法律が発効され始め、セキュリティ対策が十分でない製品は市場からの退場を余儀なくされる状況が進んできており、 IoT 製品においてセキュリティ対策は急務の課題となっています。

日本においては、独立行政法人情報処理推進機構(以下、 IPA)が、インターネットとの通信が行える幅広い IoT 製品を対象に、共通的な物差しで製品に具備されているセキュリティ機能を評価・可視化することを目的として、 JC-STAR(Labeling Scheme based on Japan Cyber-Security Technical Assessment Requirements)制度を開始しました。JC-STAR は将来的に諸外国のセキュリティ要件と相互認証の関係を持つと経済産業省及び IPA は発表しており、 IoT 製品にセキュリティが必須となるこれからの時代に向けて、 JC-STAR の取得がとても重要な第一歩となります。

1.1.2. セキュリティ対策の課題と Armadillo の意義

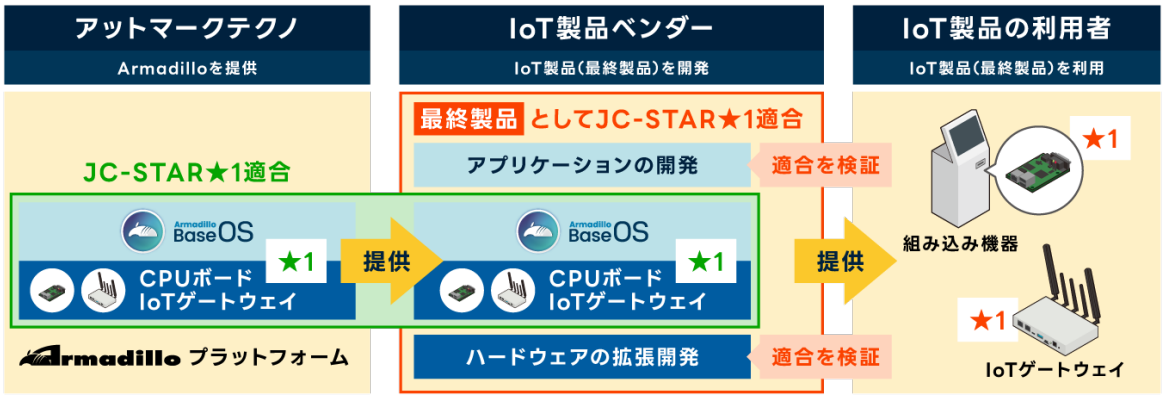
一般的な IoT 製品で JC-STAR が求める要件を満たすには、IoT 製品が本来の働きとして具備する機能に加えて、セキュリティ機能を実装する必要があり、それだけ開発の工数が増加してしまいます。そこで、各種セキュリティ機能も初めから実装している Armadillo Base OS(以下、ABOS)を搭載する Armadillo シリーズを用いて開発していただくことにより、セキュリティ機能の実装に工数をかけず、本来実現したいアプリケーションの機能の実装だけにご注力いただけます。

ABOS を搭載した Armadillo での開発における注意点として、標準状態の ABOS では開発を進めやすくするために、様々な機能やインターフェースが有効化されており、開発者が最終製品である IoT 機器を開発する中で不要な機能やインターフェースを無効化する必要があります。本書では、 ABOS を搭載した Armadillo シリーズを用いた IoT 製品の開発を行う IoT ベンダーの方が JC-STAR★1 を取得する場合に、 ABOS が具備するセキュリティ機能を最大限に活かし、よりセキュアな IoT 製品を開発・量産・運用する方法をご紹介します。

また、JC-STAR★1 では、製品自体に求められる基準以外にも、セキュリティ対策の社内体制の構築なども求められる要件があります。本書ではそれらに対して、参考情報としてアットマークテクノではどのような体制を構築しているかも紹介します。

1.1.3. ABOS 搭載製品におけるセキュリティ的な責任分界点

Armadillo を用いて開発された IoT 製品は、一般的に以下の図のようにエンドユーザー(IoT 製品の利用者)に提供されるパターンが多いです。



一連の流れの中で生じる各ソフトウェア・ハードウェアの開発やメンテナンス、脆弱性対応及び JC-STAR★1 の取得のための検証は、アットマークテクノと製品ベンダーそれぞれの責任において行われます。

1.1.3.1. アットマークテクノが行うこと

アットマークテクノは、ベースとなる Armadillo のハードウェア及び ABOS の開発を行います。さらに、 ABOS についてはリリース後のセキュリティアップデートや脆弱性対応などのメンテナンスも行います。

ABOS の特徴として、 IoT 製品のアプリケーションはコンテナ化技術で ABOS とは分離されており、 ABOS 部分とアプリケーション部分はそれぞれ独立してアップデートなどのメンテナンスを実施することができます。そのため、 ABOS 部分のメンテナンスは完成品の IoT 製品においてもアットマークテクノに任せることができます。

1.1.3.2. IoT 製品ベンダーが行うこと

IoT 製品ベンダーは、 IoT 製品の追加ハードウェア部分とアプリケーション部分を開発する必要があります。リリース後も、アプリケーション部分の継続的なセキュリティアップデートや脆弱性対応を行う必要がありますが、前述の通り、アプリケーション部分は ABOS とは分離されているため、 ABOS 部分のアップデートとは独立して行うことができます。

また、 完成品である IoT 製品が JC-STAR★1 に適合していることを検証も、 IoT 製品ベンダーもしくは IoT 製品ベンダーが検証を委託した第三者検証機関の責任の下で行われなければなりません。

1.2. 本書の目的

本書は、 JC-STAR★1 の取得に向けて、 ABOS を搭載した Armadillo シリーズを用いた IoT 製品の開発・量産・運用時に利用できる ABOS のセキュリティ機能や仕様について紹介します。

JC-STAR★1 を申請する際のチェックリストのエビデンスとしても利用可能です。

1.3. 対象読者

本書は、以下のような方を対象としています。

- ABOS を搭載している Armadillo シリーズを用いた IoT 製品で、 JC-STAR★1 の取得を考えている方
- JC-STAR★1 の取得は考えていないが、ABOS を搭載している Armadillo シリーズを用いた IoT 製品でセキュアな開発・運用をしたい方

本書は、JC-STAR★1 の各要件を満たすために、どの程度のセキュリティ機能を有していれば良いかの基準を解説するものではなく、本書に記載の内容を実施したことで、必ず JC-STAR★1 を取得できることを保証するものではありません。本書を読み進める前に、必ず以下の IPA が提供する JC-STAR の制度概要、要求事項などをお読みいただき、理解した上で本書をお読みください。

IPA - セキュリティ要件適合評価及びラベリング制度（JC-STAR）

https://www.ipa.go.jp/security/jc-star/index.html

JC-STAR★1 の評価や申請のコンサルティング、適合性評価を委託できる業者をアットマークテクノからご紹介することも可能です。詳細は弊社の営業部まで個別にお問い合わせください。

アットマークテクノ営業部 E-mail

sales@atmark-techno.com [mailto:sales@atmark-techno.com]

1.4. 本書の構成

「2. 用語説明」では、本書内で使用される用語の説明をしています。

「3. S1.1-01」から「18. S1.1-16」では、標準状態の Armadillo やアットマークテクノは各要件にどのように対応しているか、 Armadillo を用いて開発した完成品において要件を満たすために、 Armadillo が具備するどのような機能が役立つかを、 JC-STAR★1 の 16 個の要件ごとに紹介しています。

1.5. フォント

本書では以下のような意味でフォントを使いわけています。

表 1.1 使用しているフォント

フォント例	説明
本文中のフォント	本文
[PC ~]$ ls	プロンプトとユーザ入力文字列
text	編集する文字列や出力される文字列。またはコメント

1.6. コマンド入力例

本書に記載されているコマンドの入力例は、表示されているプロンプトによって、それぞれに対応した実行環境を想定して書かれています。「 / 」の部分はカレントディレクトリによって異なります。各ユーザのホームディレクトリは「 ~ 」で表します。

表 1.2 表示プロンプトと実行環境の関係

プロンプト	コマンドの実行環境
[PC /]#	作業用 PC の root ユーザで実行

プロンプト	コマンドの実行環境
[PC /]$	作業用 PC の一般ユーザで実行
[ATDE ~/]#	ATDE 上の root ユーザで実行
[ATDE ~/]$	ATDE 上の一般ユーザで実行
[armadillo /]#	Armadillo 上 Linux の root ユーザで実行
[armadillo /]$	Armadillo 上 Linux の一般ユーザで実行
[container /]#	Podman コンテナ内で実行
⇒	Armadillo 上 U-Boot の保守モードで実行

コマンド中で、変更の可能性のあるものや、環境により異なるものに関しては以下のように表記します。適宜読み替えて入力してください。

表 1.3 コマンド入力例での省略表記

表記	説明
[VERSION]	ファイルのバージョン番号

1.7. アイコン

本書では以下のようにアイコンを使用しています。

注意事項を記載します。

役に立つ情報を記載します。

用語の説明や補足的な説明を記載します。

2. 用語説明

本書で使用する用語について以下のように定義します。

本書内では IPA が提供する JC-STAR 関連資料内で用いられる用語も使用しています。IPA が提供する JC-STAR 評価ガイドもあわせて参照してください。

表 2.1 用語説明

用語	説明
ABOSDE	Armadillo Base OS Development Environment の略称で、ABOS 搭載製品での開発をしやすくするための Visual Studio Code(以下、VS Code)で使用できる拡張機能です。
ABOS Web	ABOS にデフォルトでインストールされている、 Armadillo の各種設定を Web UI 及び REST API 経由で行うことができるツールです。
Armadillo Base OS(ABOS)	アットマークテクノが提供する多面的なセキュリティ機能を有する Linux ベースのディストリビューションです。詳細は「Armadillo Base OS とは [https://armadillo.atmark-techno.com/guide/armadillo-base-os]」を参照してください。
Armadillo Twin	ABOS 搭載のデバイスをリモートから運用管理することができるクラウドサービスです。詳細は「Armadillo Twin とは [https://armadillo.atmark-techno.com/guide/armadillo-twin]」を参照してください。
IoT 製品	IoT 機器や付随サービスなどをひとまとめにパッケージングした、エンドユーザーに提供される製品そのものを指します。
IoT 製品ベンダー	IoT 製品を開発する企業を指します。
IoT 機器	IoT 製品に含まれる物理的な機器自体を指します。
IPA	独立行政法人情報処理推進機構の略称です。
エンドユーザー	IoT 製品ベンダーが開発した IoT 製品を実際に利用する個人・企業・団体等を指します。
開発者	Armadillo を使用して IoT 製品を開発する人を指します。
技術文書	IoT 製品の設計書や仕様書等を指します。
標準状態の ABOS	Armadillo 購入後の何も設定がなされていない、デフォルトで Armadillo に書き込まれている ABOS を指します。

3. S1.1-01

S1.1-01 では、IoT 機器またはユーザからの守るべき情報資産へのアクセスにおいて、適切な認証に基づくアクセス制御を行っていることが求められます。

開発した IoT 製品において、以下を明確にする必要があります。

- 開発した IoT 製品において守るべき情報資産が何か
- 守るべき情報資産へのアクセス手段がどういったものがあるか
- 各アクセス手段に対する認証の仕様(適切な認証によるアクセス制御ができているかどうか)

守るべき情報資産や適切なアクセス制御がどういったものを指すかについては、評価ガイドおよびチェックリストをご参照ください。

3.1. 守るべき情報資産をリストアップする

本要件を満たすために、まずは IoT 製品が持つ守るべき情報資産が何かを定義する必要があります。

標準状態の ABOS における守るべき情報資産としては以下が挙げられます。

標準状態の ABOS における守るべき情報資産

- 通信機能に関する設定情報
  - 無線 LAN アクセスポイントの SSID とパスワード
  - LTE の APN などの認証情報
- セキュリティ機能に関する設定情報
  - Linux ユーザーのログインパスワード
  - SE050 に格納している秘密鍵や証明書

これらに加えて、開発した IoT 製品が持つ守るべき情報資産をリストアップしてください。

3.2. 守るべき情報資産へのアクセス方法をリストアップする

次に、 IoT 製品が持つ IP 通信を介した守るべき情報資産へのアクセス方法をリストアップする必要があります。

標準状態の ABOS において、 「標準状態の ABOS における守るべき情報資産」で挙げた Armadillo 内の情報資産に IP 通信を介してアクセスする手段として以下を提供しています。

標準状態の ABOS における Armadillo 内の情報資産に IP 通信を介してアクセスする手段

- ABOS Web によるアクセス
  - Web UI 経由でのアクセス
  - REST API 経由でのアクセス

これらに加えて、開発した IoT 製品が持つ IP 通信を介した守るべき情報資産へのアクセス方法をリストアップしてください。

3.2.1. ABOS Web を無効化する

ABOS には、ネットワークの設定や SWU イメージのインストールなど各種設定が行うことが可能な ABOS Web という機能があります。ABOS Web は、 Web UI 経由で各種設定を行えるほか、 REST API を介して各種設定を行うことも可能です。デフォルトでは Armadillo と作業用 PC が同一 LAN 内に存在している場合に限り、 HTTPS 通信を介してアクセスすることができます。

この ABOS Web は、 Armadillo が持つ守るべき情報資産に対して IP 通信を介してアクセスできる方法の一つとして挙げられます。

最終的な IoT 製品において、この ABOS Web が提供する機能が不要である場合、 ABOS Web を無効化することができます。

「図 3.1. ABOS Web を停止する」に示すコマンドを実行することで、 ABOS Web を停止・無効化することができます。

```
[armadillo ~]# rc-update | grep abos-web ❶
             abos-web |      default
[armadillo ~]# rc-service abos-web status ❷
 * status: started
[armadillo ~]# rc-service abos-web stop ❸
abos-web                 | * Stopping abos-web ... [ ok ]
[armadillo ~]# rc-update del abos-web ❹
 * service abos-web deleted from runlevel default
[armadillo ~]# persist_file -d /etc/runlevels/default/abos-web ❺
```

図 3.1 ABOS Web を停止する

❶ OpenRC に ABOS Web のサービスが登録されていることを確認します。

❷ ABOS Web のサービスが起動していることを確認します。

❸ ABOS Web のサービスを停止します。

❹ サービスを管理している OpenRC から ABOS Web のサービスの登録を解除します。

❺ サービス設定ファイルの削除を永続化します。

ABOS Web を停止すると ABOS Web の Rest API も使用できなくなります。

3.3. 各アクセス方法に対するユーザー認証の仕様を明確化する

「3.2. 守るべき情報資産へのアクセス方法をリストアップする」でリストアップした各アクセス方法における認証の仕様が JC-STAR★1 の要件を満たした上で、 IoT 製品の技術文書に記載されている必要があります。

開発者が実装した認証の仕様が JC-STAR★1 の要件を満たしているかどうかは、評価ガイド等を参照してください。

以下では、「標準状態の ABOS における Armadillo 内の情報資産に IP 通信を介してアクセスする手段」で挙げた ABOS が持つ守るべき情報資産への各アクセス方法に対するユーザー認証の仕様について説明します。

3.3.1. ABOS Web のアクセス制御

ABOS Web はデフォルトで同一 LAN 内の 機器からのアクセスのみを許可するように実装しています。Armadillo 内に /etc/atmark/abos_web/init.conf を作成して内容を記述することで、 ABOS Web のアクセス制御を変更することができます。

```
[armadillo ~]# vi /etc/atmark/abos_web/init.conf
command_args="--allowed-subnets '10.88.0.0/16 127.0.0.0/8 ::1/128'" ❶
[armadillo ~]# persist_file -v /etc/atmark/abos_web/init.conf ❷
'/mnt/etc/atmark/abos_web/init.conf' -> '/target/etc/atmark/abos_web/init.conf'
[armadillo ~]# rc-service abos-web restart ❸
```

図 3.2 /etc/atmark/abos_web/init.conf の作成例

❶ Armadillo 内のコンテナとループバックからのみアクセスを許可します

❷ 再起動後も設定を維持するようにします

❸ 設定を反映させるために ABOS Web を再起動します

3.3.2. ABOS Web の Web UI におけるアクセス制御

ABOS Web はログイン時にパスワードを入力することを要求します。初回ログイン時にパスワードが設定されていない場合、ログイン用パスワードの設定を求めます。

図 3.3 パスワード登録画面

この時に設定できるパスワードは以下の条件を満たすように実装しています。

・ 8 文字以上のパスワード長にする

3.3.3. ABOS Web の REST API におけるアクセス制御

ABOS Web に REST API 経由でアクセスする場合、以下の 2 つの認証方式が提供されています。

・ Basic 認証(パスワード認証)

・ Bearer 認証(トークン認証)

  ・ トークンに付与する権限によっては守るべき情報資産へのアクセスが伴わない場合もあります

Basic 認証を使用する場合、「3.3.1. ABOS Web のアクセス制御」で設定したパスワードを入力することが求められるため、同様に以下の条件を満たすパスワードが用いることになります。

・ 8 文字以上のパスワード長にする

パスワードが流出すると認証を突破されてしまうので、設定したパスワードは流出しないように保存・使用してください。

Bearer 認証を使用する場合においても、取得したトークンが流出すると認証を突破されてしまうので、流出しないように保存・使用してください。

3.4. 技術文書への記載

S1.1-01 はドキュメント評価です。

IoT 製品の技術文書に以下を明確に記載してください。アットマークテクノが提供する機能を使用する場合は、上記をご参照ください。

・ 開発した IoT 製品において守るべき情報資産が何か

・ 守るべき情報資産へのアクセス手段がどういったものがあるか

・ 各アクセス手段に対する認証の仕様(適切な認証によるアクセス制御ができているかどうか)

4. S1.1-02

S1.1-02 では、ネットワークを介したユーザ認証の仕組み、または IoT 機器初期設定時のクライアント認証の仕組みとしてパスワードによる認証を実装している場合に、パスワード強度を一定以上に保つための仕組みを要求しています。

これは総当たり攻撃などの自動化されたパスワードクラッキングに対して耐性を持たせるためです。

4.1. ネットワークを介したユーザ認証をリストアップする

Armadillo Base OS が提供する機能でネットワークを介したユーザ認証が必要なものは以下が挙げられます。

標準状態の ABOS におけるネットワークを介したユーザ認証

- ABOS Web

これらに加えて、開発した IoT 製品が持つネットワークを介したユーザ認証をリストアップしてください。

4.2. 使用しない機能の無効化

リストアップした機能のうち、使用しないものは無効化してください。

アットマークテクノが提供する機能を使用する場合は、以下の手順で無効化できます。

4.2.1. ABOS Web の無効化

ABOS Web の無効化は、「3.2.1. ABOS Web を無効化する」を参照してください。

4.2.2. sshd の無効化

sshd を運用時に使用することは推奨しません。sshd サービスは ABOS の標準状態では無効化されていますが、開発中に有効化される場合があります。量産に入る前に忘れずに無効化することを推奨します。sshd の無効化は、以下の手順で行います。

```
[armadillo ~]# rc-service sshd stop
[armadillo ~]# rc-update del sshd
[armadillo ~]# persist_file -d /etc/runlevels/default/sshd
```

図 4.1 sshd の無効化

4.3. パスワードを使用する認証機能への対応

Armadillo を無線 LAN アクセスポイントとして動作させる際のパスワードと、ABOS Web のログインパスワード、 Armadillo Twin のログインパスワードにおいて、アットマークテクノでは以下の方法でパスワードの強度を一定以上に保つようにしています。

4.3.1. 無線 LAN アクセスポイントのパスワード

Armadillo を無線 LAN アクセスポイントとして動作させる際のパスワードは、ABOS Web の「WLAN 設定」から設定できます。ABOS Web では、無線 LAN アクセスポイントのパスワードを設定する際に、以下のような制限を設けています。

- 8 文字以上のパスワード長にする

4.3.2. ABOS Web のパスワード

ABOS Web のパスワード設定には以下の方法があります。

- initial_setup.swu を使用する
- ABOS Web の初回ログイン時にパスワードを設定する
- エンドユーザーに ABOS Web のパスワードを設定してもらう

それぞれの設定方法において、 ABOS Web のログインパスワードの強度を一定以上に保つための仕組みを提供しています。

4.3.2.1. initial_setup.swu を使用する場合

開発前に initial_setup.swu をインストールします。

initial_setup.swu を生成するタイミングで、ABOS Web のパスワードを設定します。

パスワードの強度を一定以上に保つために以下のような対応を行っています。

- 辞書に載っている言葉を使用しない
- 単調な文字列を使用しない
- 8 文字以上のパスワード長にする

4.3.2.2. ABOS Web の初回ログイン時にパスワードを設定する場合

initial_setup.swu を使用しない場合、ABOS Web の初回ログイン時にパスワードを設定します。この場合も、パスワードの強度を一定以上に保つために以下のような対応を行っています。

- 8 文字以上のパスワード長にする

4.3.2.3. エンドユーザーに ABOS Web のパスワードを設定してもらう場合

エンドユーザーに ABOS Web の初回ログイン時にパスワードを設定してもらう場合は、現在のパスワードをリセットする必要があります。ABOS Web の「設定管理」の「パスワードリセット」を使用します。

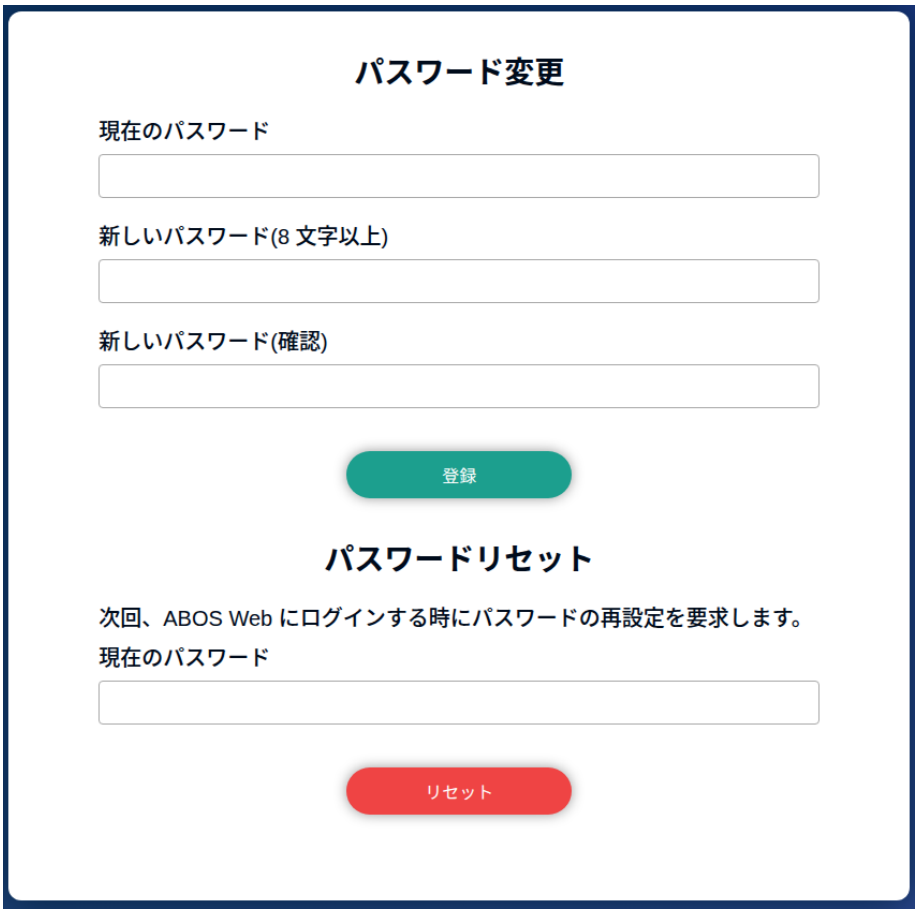


図 4.2 ABOS Web のパスワードリセット

現在のパスワードを入力して「リセット」ボタンを押すと、次回 ABOS Web ログイン時にパスワードの再設定が要求されます。この場合も、「4.3.2.2. ABOS Web の初回ログイン時にパスワードを設定する場合」と同様の基準でパスワードの強度を担保します。

4.3.3. Armadillo Twin のパスワード

Armadillo Twin のアカウント作成時にパスワードを設定します。

アカウントの作成方法は、Armadillo Twin のマニュアルを参照してください。

Armadillo Twin のマニュアル [https://manual.armadillo-twin.com/create-account-and-user/]

Armadillo Twin では、パスワードの強度を一定以上に保つために以下のような対応を行っています。

- 8 文字以上、99 文字以下のパスワード長にする
- 数字、大文字、記号を含める

4.4. 技術文書への記載

S1.1-02 はドキュメント評価です。

アットマークテクノが提供する機能を使用する場合は、上記をご参照ください。

また、開発者が開発するアプリケーションにおいて、守るべき情報資産に対してパスワードによるアクセス制御を施す場合は、同様に評価項目 1 または評価項目 2 を満たすパスワード強度を担保する仕組みを実装する必要があります。

チェックリストではその実装内容を明示する必要がありますので、実装した仕組みについて技術文書に記載してください。

5. S1.1-03

S1.1-03 では、ユーザ認証に使用される認証値の変更を可能にすることを要求しています。

5.1. ネットワークを介したユーザ認証機能をリストアップする

「4.1. ネットワークを介したユーザ認証をリストアップする」 と同様に、ネットワークを介したユーザ認証を必要とする機能をリストアップします。

さらに、開発者が開発するアプリケーションにおいて、守るべき情報資産に対してパスワードによるアクセス制御を施す場合は、ユーザ認証に使用される認証値の変更を可能にする仕組みを実装する必要があります。

5.2. 使用しないユーザ認証機能の無効化

「4.2. 使用しない機能の無効化」 と同様に、使用しないユーザ認証機能を無効化します。

5.3. ユーザ認証に使用される認証値の変更を可能にする

ユーザ認証に使用される認証値は、エンドユーザーが変更できるように設計・開発を行ってください。

アットマークテクノが提供するネットワークを介したユーザ認証に使用される認証値の変更方法を以下に示します。

5.3.1. 無線 LAN アクセスポイント

ABOS Web から無線 LAN アクセスポイントのユーザ認証に使用される認証値を変更するには、「WLAN 設定」画面から一度現在の設定を削除した後、改めて設定を行う必要があります。

5.3.2. ABOS Web

ABOS Web におけるネットワークを介したユーザ認証に使用される認証値として、ログインパスワードと REST API 用のトークンの 2 つが挙げられます。

5.3.2.1. ABOS Web のログインパスワードの変更

ABOS Web のログインパスワードは、 ABOS Web の「設定管理」の「パスワード変更」画面から変更できます。

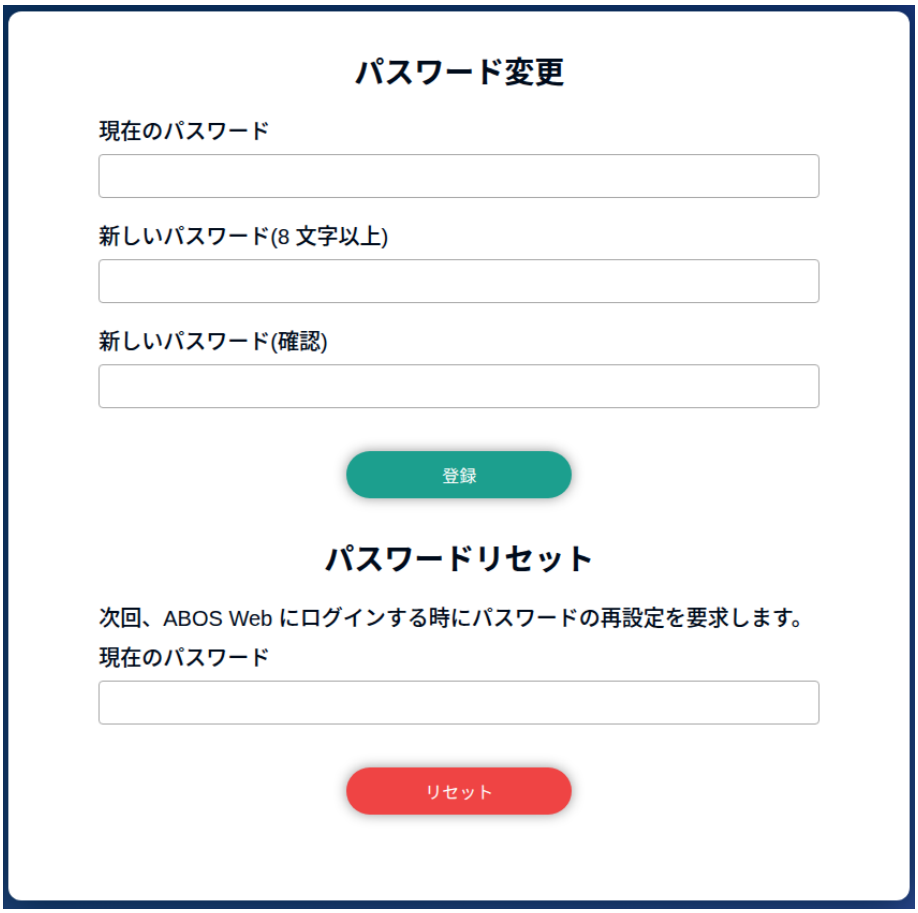


図 5.1 ABOS Web のパスワード変更画面

5.3.2.2. ABOS Web の REST API 用トークンの変更

ABOS Web の REST API 用トークンは、 ABOS Web の「設定管理」の「Rest API トークン一覧」画面から削除、追加できます。

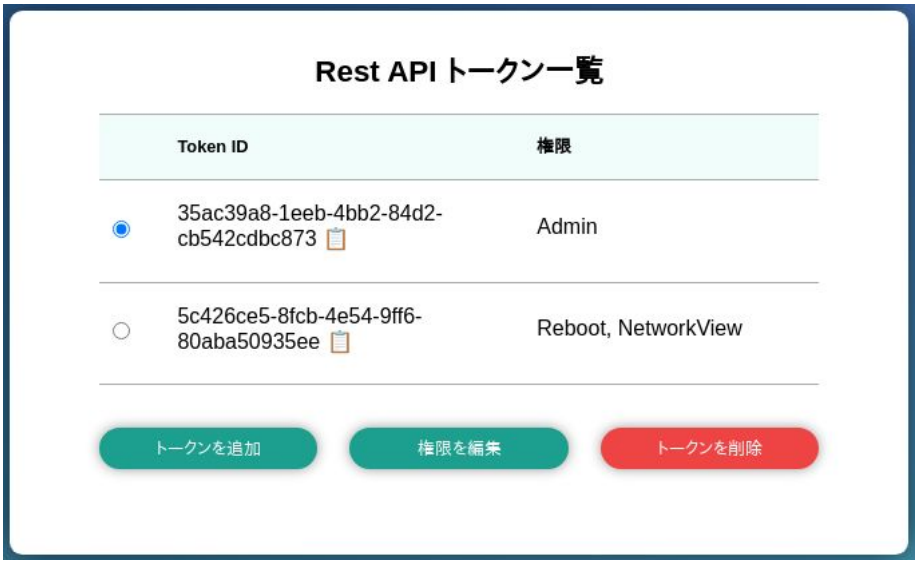


図 5.2 ABOS Web の REST API 用トークンの削除・追加

5.3.3. Armadillo Twin

Armadillo Twin では、パスワードの変更を「ユーザー設定」画面から行うことができます。

詳細は Armadillo Twin のユーザーマニュアルを参照してください。

Armadillo Twin ユーザマニュアル「パスワードを変更する」 [https://manual.armadillo-twin.com/change-user-config/#toc4]

Armadillo Twin は REST API 機能を提供しています。API をコールする際に使用するトークンは以下のコマンドで変更・取得することができます。

```
[PC ~]$ curl -X POST -H "Content-Type: application/json" -d '{"username":"ユーザー名","password":"パスワード"}' "https://api.armadillo-twin.com/login"
```

図 5.3 Armadillo Twin の REST API トークンを取得するコマンド

もう一度、同じコマンドを実行すると、トークンが更新されます。前のトークンは無効化されますので、注意してください。

Armadillo Twin の REST API 機能に関する情報は、以下のリンクを参照してください。

Armadillo Twin ユーザマニュアル「Armadillo Twin REST API」 [https://manual.armadillo-twin.com/about-armadillo-twin-rest-api/]

5.4. エンドユーザーが各認証値を変更する方法を技術文書に明記する

S1.1-03 はドキュメント評価です。

アットマークテクノが提供する機能の変更方法については、上記をご参照ください。

また、開発者が開発するアプリケーションにおいて、守るべき情報資産に対してパスワードによるアクセス制御を施す場合は、ユーザ認証に使用する認証値の変更を可能にする仕組みを実装する必要があります。

チェックリストではその実装内容を明示する必要がありますので、実装した仕組みについて技術文書に記載してください。

6. S1.1-04

S1.1-04 では、ネットワークを介したユーザ認証の仕組みが総当たり攻撃に対して耐性を持つことを要求しています。

6.1. ネットワークを介したユーザ認証機能をリストアップする

「4.1. ネットワークを介したユーザ認証をリストアップする」 と同様に、ネットワークを介したユーザ認証を必要とする機能をリストアップします。

さらに、開発者が開発するアプリケーションにおいて、守るべき情報資産に対してパスワードによるアクセス制御を施す場合は、総当たり攻撃に対して耐性を持たせる必要があります。

6.2. 使用しないユーザ認証機能の無効化

「4.2. 使用しない機能の無効化」 と同様に、使用しないユーザ認証機能を無効化します。

6.3. 総当たり攻撃への対策を行う

開発者が開発するアプリケーションにおいて、ネットワークを介したユーザ認証を行う仕組みを提供する場合、総当たり攻撃に対して耐性を持たせる必要があります。

アットマークテクノが提供するネットワークを介したユーザ認証の仕組みでは、以下の対策を行っています。

6.3.1. ABOS Web

ABOS Web では、パスワードの認証処理において、認証が失敗した場合に 2 秒以上の待ち時間を設けることで、総当たり攻撃に対して耐性を持たせています。

6.3.2. Armadillo Twin

Armadillo Twin では、ログイン画面でパスワード認証が 5 回失敗すると 1 秒間ロックアウトします。その後、失敗する度にロックアウト時間を延長することで、総当たり攻撃に対して耐性を持たせています。

6.4. 実機テストによる評価

S1.1-04 は実機テストによる評価です。

開発するアプリケーションにおいて、守るべき情報資産に対してパスワードによるアクセス制御を施す場合は、同様に総当たり攻撃に対して耐性を持たせる必要があります。

例えば、アットマークテクノ では hydra を使用して対象のユーザ認証に対して総当たり攻撃を実施し、認証失敗時に再認証可能まで 2 秒以上時間がかかることを確認しています。

hydra については以下をご参照ください。

https://github.com/vanhauser-thc/thc-hydra

7. S1.1-05

S1.1-05 は、IoT 機器のセキュリティ機能についてではなく、機器メーカーとして脆弱性に対応するための体制を整えることを求める要件です。

S1.1-05 はドキュメント評価になります。

製造業者は脆弱性開示ポリシーを策定し、公開する必要があります。脆弱性開示ポリシーには以下の内容を含める必要があります。

- 問題を報告するための連絡先情報
- 以下のタイムラインに関する情報
  - 最初の受領確認
  - 報告された問題が解決されるまでの状況の更新

詳細な脆弱性開示ポリシーに求められる内容については、評価ガイドおよびチェックリストをご参照ください。

7.1. アットマークテクノと IoT 製品ベンダーの責任分界点

基本的なセキュリティ機能をはじめとした OS の機能は ABOS としてアットマークテクノが提供・メンテナンスします。アプリケーションはコンテナとして OS と分離された環境で動作し、アップデートも OS とは別々に実施することができます。そのため、 Armadillo を用いて開発された IoT 機器において、アットマークテクノがメンテナンスする部分と IoT 製品ベンダーがメンテナンスする部分に分かれます。詳細は「1.1.3. ABOS 搭載製品におけるセキュリティ的な責任分界点」を参照してください。

ABOS は基本的にはアットマークテクノがメンテナンスし、アップデートを提供します。開発者は ABOS の設定を変更することで、利用する ABOS の機能を選択できます。ABOS の設定は ABOS 自体をアップデートしても引き継がれます。

ただし、開発者が ABOS の機能自体の改修をした場合、 アットマークテクノが提供するアップデートをそのまま適用することができなくなる場合があります。ABOS の機能改修を行う場合は、一度アットマークテクノにご相談いただくことをお勧めします。

アプリケーションは開発者がメンテナンスし、エンドユーザーにアップデートを提供する必要があります。

7.2. 参考情報

開発者および IoT 製品ベンダーが IoT 製品の脆弱性開示ポリシーを策定する際の参考情報として、アットマークテクノの脆弱性開示ポリシーを紹介します。

アットマークテクノの脆弱性開示ポリシー

https://www.atmark-techno.com/vulnerability-disclosure-policy/

8. S1.1-06

S1.1-06 では、ファームウェアのアップデート機構を備えており、かつ、エンドユーザがバージョン確認を行うことができ、再起動後もバージョンが維持されることを要求しています。

これは、ファームウェアのダウングレード攻撃を防止するためです。

8.1. SWUpdate によるソフトウェアアップデート方法を理解する

Armadillo のファームウェアアップデート機構として、 SWUpdate を提供しています。SWUpdate は、ソフトウェアコンポーネントをアップデートするためのツールです。desc ファイルというインストール実行時に行う処理を記載したファイルから SWU イメージというアップデートファイルを生成します。この SWU イメージを様々な方法で Armadillo にインストールすることでアップデートを実現できます。

SWU イメージを Armadillo にインストールする方法は以下の通りです。

- 手元でイメージをインストールする方法
  - CUI コマンドを使用した手動インストール
  - ABOS Web を使用した手動インストール
  - ABOSDE から ABOS Web を使用した手動インストール
  - USB メモリまたは microSD カードからの自動インストール
  - 外部記憶装置からの手動インストール
- リモートでイメージをインストールする方法
  - Armadillo Twin を使用した自動インストール
  - ウェブサーバーからの手動インストール
  - ウェブサーバーからの定期的な自動インストール

上記の方法で SWU イメージをインストールしてソフトウェアアップデートを行うことで、電源 OFF 後もバージョンは維持されます。

それぞれのインストール方法の詳細については、以下に記載しています。

8.1.1. SWU イメージのインストール方法

8.1.1.1. CUI コマンドを使用した手動インストール

Armadillo 上で以下のコマンドを使用することで最新の ABOS にアップデートできます。

```
[armadillo ~]# abos-ctrl update
```

8.1.2. ABOS Web を使用した手動インストール

ABOS Web から PC 上の SWU イメージや HTTP サーバー上の SWU イメージを Armadillo にインストールできます。

ABOS Web のトップページから、"SWU インストール"をクリックすると、「図 8.1. SWU インストール」の画面に遷移します。

図 8.1 SWU インストール

この画面では、PC 上の SWU イメージファイルまたは HTTP サーバー上の SWU イメージファイルの URL を指定して、Armadillo にインストールできます。

"現在の SWU で管理されているバージョン" 欄には、ABOS の各ソフトウェアコンポーネントの名前とバージョン情報を一覧表示します。

図 8.2 SWU 管理対象ソフトウェアコンポーネントの一覧表示

8.1.2.1. ABOSDE から ABOS Web を使用した手動インストール

VS Code 拡張機能の ABOSDE を使用することで、Armadillo で動作している ABOS Web 経由でアップデートできます。

ローカルネットワーク上の Armadillo をスキャンした後に、「図 8.3. ABOSDE で Armadillo に SWU をインストール」 の赤枠で囲まれているボタンをクリックすることで、選択した Armadillo に SWU をインストールできます。SWU インストールのログは VS Code 画面下部の OUTPUT に表示されます。

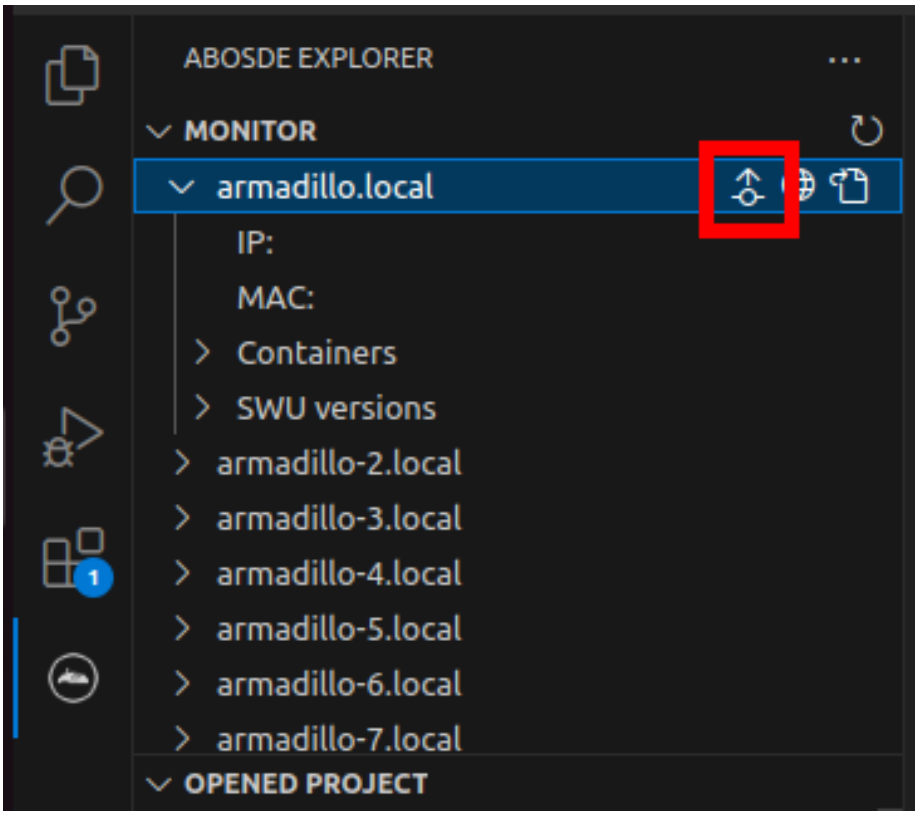


図 8.3 ABOSDE で Armadillo に SWU をインストール

8.1.2.2. USB メモリまたは microSD カードからの自動インストール

Armadillo に USB メモリを接続すると自動的にアップデートが始まります。アップデート終了後に Armadillo は自動で再起動します。

USB メモリや microSD カードを vfat もしくは ext4 形式でフォーマットし、作成した SWU イメージをディレクトリを作らずに配置してください。

ATDE 上で USB メモリ/microSD カードのパーティションを作成・フォーマットする方法

https://armadillo.atmark-techno.com/howto/atde-partition-howto

```
[ATDE ~/mkswu]$ df -h
Filesystem           Size  Used Avail Use% Mounted on
: (省略)
/dev/sda1             15G  5.6G  9.1G  39% /media/USBDRIVE ❶
[ATDE ~/mkswu]$ cp initial_setup.swu /media/USBDRIVE/ ❷
[ATDE ~/mkswu]$ umount /media/USBDRIVE ❸
```

❶ USB メモリがマウントされている場所を確認します。

❷　ファイルをコピーします。

❸　/media/USBDRIVE をアンマウントします。コマンド終了後に USB メモリを取り外してください。

8.1.2.3. 外部記憶装置からイメージの手動インストール

USB メモリや microSD カード等の外部記憶装置のルートディレクトリ以外に SWU イメージを保存して、手動でイメージのインストールを行います。なお、ルートディレクトリに保存すると自動アップデートが実行されます。

```
[armadillo ~]# swupdate -i /mnt/swu/initial_setup.swu
SWUpdate v5f2d8be-dirty

Licensed under GPLv2. See source distribution for detailed copyright notices.

[INFO ] : SWUPDATE running :  [main] : Running on AGX4500 Revision at1
[INFO ] : SWUPDATE started :  Software Update started !
[INFO ] : SWUPDATE running :  [read_lines_notify] : No base os update: copying current os over
[INFO ] : SWUPDATE running :  [read_lines_notify] : Removing unused containers
[INFO ] : SWUPDATE running :  [read_lines_notify] : swupdate triggering reboot!
Killed
```

8.1.2.4. Armadillo Twin を使用した自動インストール

Armadillo Twin を使用することで、自身でサーバー構築を行うことなくネットワーク経由で SWU イメージを配信し、デバイスのソフトウェアを更新できます。

複数台管理してアップデートすることも可能です。

また、Armadillo Twin を使用したソフトウェアアップデートの実施方法については、 Armadillo Twin ユーザマニュアル「デバイスのソフトウェアをアップデートする」 [https://manual.armadillo-twin.com/update-software/] を参照してください。

8.1.2.5. ウェブサーバーからイメージの手動インストール

SWU イメージをウェブサーバーにアップロードして、イメージのインストールを行います。以下は、http://server/initial_setup.swu のイメージをインストールする例です。

```
[armadillo ~]# swupdate -d '-u http://server/initial_setup.swu'
SWUpdate v5f2d8be-dirty

Licensed under GPLv2. See source distribution for detailed copyright notices.

[INFO ] : SWUPDATE running :  [main] : Running on AGX4500 Revision at1
[INFO ] : SWUPDATE running :  [channel_get_file] : Total download size is 25 kB.
[INFO ] : SWUPDATE started :  Software Update started !
[INFO ] : SWUPDATE running :  [read_lines_notify] : No base os update: copying current os over
[INFO ] : SWUPDATE running :  [read_lines_notify] : Removing unused containers
[INFO ] : SWUPDATE running :  [read_lines_notify] : swupdate triggering reboot!
Killed
```

8.1.2.6. ウェブサーバーからの定期的な自動インストール

swupdate-url サービスを有効にすると、定期的に登録した URL をチェックして指定した時間に SWUpdate を実行します。以下はサービスの有効化とタイミングの設定の例です。

```
[armadillo ~]# rc-update add swupdate-url ❶
[armadillo ~]# persist_file /etc/runlevels/default/swupdate-url ❷
[armadillo ~]# echo https://armadillo.atmark-techno.com/files/downloads/armadillo-iot-g4/image/ ⏎
baseos-x2-latest.swu ¥
> /etc/swupdate.watch ❸
[armadillo ~]# echo 'schedule="0 tomorrow"' > /etc/conf.d/swupdate-url
[armadillo ~]# echo 'rdelay="21600"' >> /etc/conf.d/swupdate-url ❹
[armadillo ~]# persist_file /etc/swupdate.watch /etc/conf.d/swupdate-url ❺
```

❶ swupdate-url サービスを有効にします。

❷ サービスの有効化を保存します。

❸ イメージの URL を登録します。一行ごとにイメージの URL を設定でき、複数行にイメージの URL を設定できます。

❹ チェックやインストールのスケジュールを設定します。

❺ 変更した設定ファイルを保存します。

USB メモリからのアップデートと同様に、ログは /var/log/messages に保存されます。

8.2. バージョン確認方法を理解する

SWUpdate によるアップデート後、Armadillo のソフトウェアコンポーネントのバージョンを確認する方法として、以下の方法があります。

8.2.1. CUI による確認

SWUpdate によってインストールされたバージョンは、 /etc/sw-versions に記載しています。内容は以下のコマンドで確認できます。

```
[armadillo ~]# cat /etc/sw-versions
base_os <ABOS のバージョン>
boot <ブートローダーのバージョン>
:(省略)
```

8.2.2. ABOS Web による確認

ABOS Web では「図 8.4. SWU 管理対象ソフトウェアコンポーネントの一覧表示」の画面のように、インストールした SWU イメージの最新のバージョンを確認できます。

図 8.4 SWU 管理対象ソフトウェアコンポーネントの一覧表示

8.3. アップデート方法を決定する

開発者は、「8.1.1. SWU イメージのインストール方法」のいずれかの方法で SWU イメージを使用して ABOS や開発したコンテナイメージをアップデートする必要があります。

アップデート方法を決定した後、攻撃面を減らすために使用しないアップデート方法は無効化してください。

CUI で手動で SWUpdate を実行する方法を除き、「8.1.1. SWU イメージのインストール方法」のアップデート方法は大きく以下の 4 つに分類できます：

- ABOS Web 経由で SWUpdate を実行する
- USB メモリまたは microSD カード経由で SWUpdate を実行する
- Armadillo Twin 経由で SWUpdate を実行する
- ウェブサーバー経由で SWUpdate を実行する

8.3.1. ABOS Web 経由の SWUpdate の無効化

ABOS Web で SWUpdate を無効化する手順を以下に示します。

まず、ABOS Web にログインして、「設定管理」画面に移動します。「設定管理」画面の「カスタマイズ」欄にある「メニュー項目を変更する」をクリックします。

図 8.5 ABOS Web の設定管理

次に、「メニュー項目の変更」画面で「SWU インストール」の項目とその説明を空欄にします。

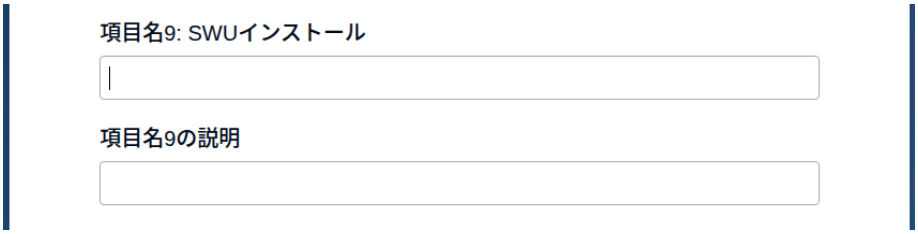


図 8.6 ABOS Web の SWU インストールの無効化

その後、一番下までスクロールして「メニューを設定」ボタンをクリックします。

これで、ABOS Web の SWU インストール機能が無効化されます。

8.3.2. USB メモリまたは microSD カード 経由の SWUpdate の無効化

USB メモリまたは microSD カードから SWUpdate を実行することを禁止するには、Armadillo 上で以下のコマンドを実行してください。

```
[armadillo ~]# mv /usr/lib/udev/rules.d/swupdate-usb.rules /usr/lib/udev/rules.d/swupdate-
usb.rules.bk
```

```
[armadillo ~]# persist_file -d /usr/lib/udev/rules.d/swupdate-usb.rules
[armadillo ~]# persist_file /usr/lib/udev/rules.d/swupdate-usb.rules.bk
[armadillo ~]# reboot
```

図 8.7 USB メモリまたは microSD カードによる SWUpdate の無効化

8.3.3. Armadillo Twin の無効化

Armadillo Twin の無効化は、以下の手順で行います。

まず、Armadillo 上で armadillo-twin-agentd を停止し、自動起動を無効にします。

```
[armadillo ~]# rc-service armadillo-twin-agentd stop
[armadillo ~]# rc-update del armadillo-twin-agentd
[armadillo ~]# persist_file -d /etc/runlevels/default/armadillo-twin-agentd
```

図 8.8 Armadillo Twin の無効化

さらに、Armadillo Twin にデバイスを登録している場合、登録している Armadillo を削除する必要があります。削除方法は、Armadillo Twin のマニュアルを参照してください。

Armadillo Twin ユーザマニュアル「デバイスを削除する」 [https://manual.armadillo-twin.com/delete-device/]

8.3.4. ウェブサーバーによる自動アップデートの無効化

ウェブサーバーからの SWUpdate の自動インストールを無効化するには、swupdate-url サービスを停止し、自動起動を無効にします。

```
[armadillo ~]# rc-service swupdate-url stop
[armadillo ~]# rc-update del swupdate-url
[armadillo ~]# persist_file -d /etc/runlevels/default/swupdate-url
```

図 8.9 swupdate-url サービスの無効化

8.4. バージョンの確認方法を決定する

開発者は、エンドユーザーに対してファームウェアのバージョンを確認する方法を提供する必要があります。

「8.2. バージョン確認方法を理解する」では、ABOS Web や CUI コマンドを使用してバージョンを確認する方法を説明しました。

これら以外の方法では、開発者がエンドユーザー向けにバージョンを確認する方法を別途提供する必要があります。

例えば、/etc/sw-versions ファイルからバージョン情報を取得して GUI に表示することで、エンドユーザーがバージョンを確認できるようにする実装が考えられます。

8.5. 実機テストによる評価

S1.1-06 は実機テストによる評価項目です。

開発者は上記のいずれかの方法で SWU イメージを使用して、ABOS や開発したコンテナイメージをアップデートする必要があります。

SWU イメージによるアップデートを実行した後、Armadillo のソフトウェアコンポーネントのバージョンが更新され、再起動後もそのバージョンが維持されることを確認してください。

9. S1.1-07

S1.1-07 では、ユーザがアップデートを実施する手順を明示することを要求しています。

製品のマニュアルやウェブサイトなどのユーザがアクセス可能な媒体に、アップデート手順を明確に記載する必要があります。

ファームウェアを含むセキュリティアップデートを自動で行うか、それともユーザが手動で行うかは開発者の設計に委ねられています。開発者はその設計に基づいて、ユーザがアップデートを実施する手順を明示する必要があります。

9.1. アップデートの提供形態を決定する

開発者は、セキュリティアップデートの提供形態を決定する必要があります。主な選択肢は以下の通りです。

9.1.1. SWU イメージを開発者が作成しエンドユーザーがインストールする

この形態では、開発者がアップデートファイル（SWU イメージ）を作成してエンドユーザー（メンテナー）に提供し、エンドユーザーが任意のタイミングでアップデートを適用します。

エンドユーザーが desc ファイルを記述することはほぼないため、開発者は完成したアップデートファイルをエンドユーザーに提供します。

この形態を選択した場合、以下のいずれかの評価項目に対応したアップデート手順を明示する必要があります。

- **評価項目 2**: 必須付随サービス（モバイルアプリケーション等）を利用したアップデート手順
- **評価項目 3**: 製品のインタフェース（ウェブインタフェース等）を介したアップデート手順
- **評価項目 4**: 開発者のウェブサイトからのアップデートファイルのダウンロードとインストール手順

9.1.2. SWU イメージを開発者が作成し開発者がインストールする

この形態では、開発者が SWU イメージを作成し、開発者側でリモートアップデートを実施します。

開発者が主に管理している機器については、開発者のタイミングでアップデートを配信・適用することができます。エンドユーザーの視点では、この形態は評価項目 1 の「自動アップデート」に該当します。

この形態を選択した場合、以下の評価項目に対応した内容を明示する必要があります・

- **評価項目 1**: 自動的にアップデートが実行されることが明示されていること。また、自動アップデートに失敗した場合の対応方法が明示されていること。

開発者は、システムの運用方針やエンドユーザーの技術レベルを考慮して、適切な提供形態を選択してください。

9.2. エンドユーザーが簡単にアップデートできる方法を決定する

エンドユーザーが簡単にアップデートを実施する方法は、開発者が選択したアップデートの提供形態に応じて異なります。

開発者が独自にアップデート方法を実装する場合は、その手順を明示する必要があります。

以下はアットマークテクノが提供する SWU イメージのアップデート方法の例です。

9.2.1. SWU イメージを開発者が作成しエンドユーザーがインストールする形態の場合

主な方法として以下があります。

- ABOS Web を使用した GUI での操作
- Armadillo Twin を使用した GUI での操作
- コマンドラインツールを使用した操作

ユーザの技術レベルや運用環境に応じて、最適な方法を選択してください。

9.2.1.1. ABOS Web を使用した GUI での操作

ABOS Web を使用した GUI での SWU イメージのインストール方法については、「8.1.2. ABOS Web を使用した手動インストール」をご参照ください。

9.2.1.2. Armadillo Twin を使用した GUI での操作

Armadillo Twin を使用した GUI での操作については、以下のリンクを参照してください。

Armadillo Twin ユーザマニュアル「デバイスのソフトウェアをアップデートする」 [https://manual.armadillo-twin.com/update-software/]

9.2.2. SWU イメージを開発者が作成し開発者がインストールする形態の場合

開発者は自動アップデートの方法を決定し、実装する必要があります。

swupdate-url サービスは、アットマークテクノが提供するアップデート方法の一つです。「8.1.2.5. ウェブサーバーからイメージの手動インストール」 で swupdate-url サービスの使用方法について説明していますので、参照してください。

9.3. ユーザがアクセス可能な技術文書にアップデート手順を記載する

決定したアップデート手順について、ユーザがアクセス可能な技術文書に詳細な手順を記載する必要があります。

技術文書に含める情報の例は以下の通りです。

- アップデートの実施タイミング
- 必要な事前準備

- 具体的な操作手順
- アップデート後の確認方法
- トラブルシューティング情報

10. S1.1-08

S1.1-08 では、ソフトウェアをネットワーク経由でアップデートする際、ソフトウェアの完全性をアップデート前に確認できる仕組みを有することを要求しています。

これは、アップデートで書き込むデータが改ざんされていないこと、および通信中に情報が盗聴されないことを保証するためです。

10.1. ネットワーク経由でのアップデート方法を把握する

まず、システムで使用するソフトウェアアップデート方法をリストアップし、その中からネットワーク経由でアップデートする方法を抽出する必要があります。

「8. S1.1-06」 で説明した SWUpdate によるアップデート方法のうち、ネットワーク経由でのアップデート方法は以下の通りです。

- ABOS Web を使用した手動インストール
- Armadillo Twin を使用した自動インストール
- ウェブサーバーからの定期的な自動インストール（swupdate-url サービス）

これらの方法は、いずれもネットワークインタフェースを介してソフトウェアアップデートが配信されるため、S1.1-08 の要件対象となります。

10.2. ABOS が提供するセキュアなアップデートメカニズム

10.2.1. SWUpdate のセキュアなアップデートメカニズム

アットマークテクノが提供する ABOS では、SWUpdate を使用してセキュアなソフトウェアアップデートを実現しています。

SWUpdate は、以下のセキュリティ機能を提供します。

10.2.1.1. デジタル署名による真正性・完全性検証

SWUpdate は、SWU イメージに付与された ECDSA デジタル署名を使用して、アップデートソフトウェアイメージの真正性および完全性を検証します。

アップデートソフトウェアをインストールする前に、更新ソフトウェアに付与された ECDSA デジタル署名による検証を行い、検証の結果、検証に失敗した場合にはインストールを中止します。

10.2.1.2. ハッシュ値による完全性確認

SWUpdate は、SWU イメージ内の各コンポーネントに対して SHA-256 ハッシュ値を使用した完全性確認を実行します。

アップデートソフトウェアをインストールする前またはインストール中に、更新ソフトウェアに付与された SHA-256 ハッシュ値との照合を行い、照合の結果、不一致が確認された場合にはインストールを中止します。

10.2.1.3. 暗号化による機密性保護

SWUpdate は、SWU イメージの暗号化に AES-256-CBC を使用して、アップデートデータの機密性を保護します。

これにより、ネットワーク経由でのアップデート時に、第三者による盗聴からアップデートデータを保護します。

10.2.1.4. 「電子政府推奨暗号リスト」の準拠

SWUpdate で使用するハッシュ関数やデジタル署名は、「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）」のうち「電子政府推奨暗号リスト」に記載されたアルゴリズムを利用しています。

これにより、ベストプラクティスの暗号技術を使用したセキュアなアップデートメカニズムを実現しています。

10.2.2. 推奨されるアップデート方法

特殊な事情がない限り、ネットワーク経由でのソフトウェアアップデートには SWUpdate の使用を強く推奨します。

SWUpdate は、セキュアなアップデートメカニズムを容易にするために必要な暗号技術とプロトコルを実装しており、信頼関係を介して各アップデートの真正性および完全性を検証できます。

10.3. 開発者独自のアップデート方法を使用する場合の留意点

開発者が独自のソフトウェアアップデート方法を実装する場合、以下の点に留意してください。

10.3.1. セキュリティ要件の実装

独自のアップデート方法では、以下のセキュリティ要件を満たす必要があります。

- アップデートソフトウェアの完全性をアップデート前に確認する仕組み
- デジタル署名またはハッシュ値による検証機能
- 検証失敗時のインストール中止機能
- CRYPTREC 暗号リストに記載された暗号アルゴリズムの使用

10.3.2. サポート対象外となることの理解

開発者独自のアップデート方法を使用する場合、そのアップデートメカニズムに関する問題や脆弱性については、アットマークテクノのサポート対象外となります。

開発者は、独自のアップデートメカニズムのセキュリティ評価、脆弱性対応、および継続的なセキュリティ維持を自身の責任で行う必要があります。

10.4. 技術文書への記載要件

S1.1-08 の要件を満たすため、技術文書には以下の評価項目を満たす内容を明示する必要があります。

評価項目 1〜4 のいずれかに類する仕組み、かつ評価項目 5 を満たすことが適合の条件となります。

10.4.1. SWUpdate による要件の満足

SWUpdate を使用する場合、以下により上記の評価項目を満たすことができます。

- **評価項目 1 を満足**: SHA-256 ハッシュ値による完全性確認の実装
- **評価項目 2 を満足**: ECDSA デジタル署名による検証の実装
- **評価項目 5 を満足**: SHA-256 および ECDSA は CRYPTREC 暗号リストの電子政府推奨暗号リストに記載されたアルゴリズム

11. S1.1-09

S1.1-09 では、製品のセキュリティアップデートの優先度を決定するための方針や指針を文書化することが求められます。

この要件では、以下の内容について検討・決定し、文書化しておく必要があります。

- セキュリティアップデートの優先度を決定するための基準
- インシデントレスポンスをハンドリングするための組織体制
- 脆弱性情報の収集、トリアージや分析、対策、アップデートなどの一連の対応プロセス
- 製品のステークホルダー間の連絡体制

本節では、 Armadillo を用いた IoT 機器において、上記の内容の決定に役立つ機能やサービスについての説明をします。また、「11.3. アットマークテクノの方針」では、参考情報としてアットマークテクノが上記の内容に対してどのような方針で対応しているかを説明します。

11.1. 脆弱性情報の収集方法

一般的に脆弱性の収集は、使用している各種ライブラリの提供元や、セキュリティ関連の情報を提供する組織から取得します。しかし、 IoT 機器を構成するソフトウェアの数と種類は膨大であり、それらの情報を手動で収集するには多大な労力が必要です。

ABOS ではアットマークテクノが提供する OS 部分と、開発者が作成するコンテナアプリケーション部分の両方において、 SBOM (Software Bill of Materials) を活用して脆弱性情報を収集します。

OS 部分の SBOM は、アットマークテクノがリリースしています。コンテナアプリケーション部分の SBOM は、 ABOS の専用開発ツールである ABOSDE を用いてアプリケーションを開発することで自動的に生成されます。

11.1.1. SBOM スキャンツールを用いた脆弱性情報の収集

SBOM をスキャンツールにかけることで、その時点でのソフトウェアの脆弱性情報を収集することができます。

ABOS の開発環境である ATDE では、デフォルトで SBOM スキャンツールである OSV-Scanner [1] がインストールされています。

「図 11.1. OSV-Scanner を用いて SBOM をスキャンする」では、例として ATDE 上で OSV-Scanner を用いて ABOSDE で開発したアプリケーションの SBOM (development.swu.spdx.json)をスキャンします。

```
[ATDE ~]$ osv-scanner scan --sbom ~/my_project/development.swu.spdx.json --format markdown ❶
Scanned /home/atmark/my_project/development.swu.spdx.json as SPDX SBOM and found 97 packages
5 unimportant vulnerabilities have been filtered out.
Filtered 5 vulnerabilities from output
| OSV URL | CVSS | Ecosystem | Package | Version | Source |
```

[1] OSV-Scanner: https://github.com/google/osv-scanner/

```
| --- | --- | --- | --- | --- | --- |
| https://osv.dev/CVE-2022-3715 | 7.8 | Debian | bash | 5.1-2+deb11u1 | development.swu.spdx.json |
| https://osv.dev/CVE-2016-2781 | 6.5 | Debian | coreutils | 8.32-4 | development.swu.spdx.json |
| https://osv.dev/CVE-2021-33560 | 7.5 | Debian | libgcrypt20 | 1.8.7-6 | development.swu.spdx.json ⏎
|
| https://osv.dev/CVE-2024-2236 |  | Debian | libgcrypt20 | 1.8.7-6 | development.swu.spdx.json |
```

図 11.1 OSV-Scanner を用いて SBOM をスキャンする

❶ 今回は見やすさのために format を markdown に設定しています

OSV URL 列の URL にアクセスすることで各脆弱性の詳細を確認することができます。

11.1.2. 継続的な SBOM スキャンによる脆弱性情報の収集

SBOM のスキャンは、そのスキャンを実行した時点での脆弱性情報のみを収集します。しかし、脆弱性は日々新たに発見され続けるため、 SBOM をスキャンした時点では脆弱性情報が存在しなかったライブラリに対しても、後から脆弱性情報が公開される可能性があります。そのため、 製品をリリースした後も定期的に SBOM をスキャンして脆弱性情報を収集する必要があります。

「11.1.1. SBOM スキャンツールを用いた脆弱性情報の収集」で紹介したスキャン手順を手動、あるいは CI/CD パイプラインの一部として定期的に実行することで、継続的に脆弱性情報を収集する仕組みを構築しても良いですが、 ABOS では Armadillo Twin と連携することで、 SBOM を Armadillo Twin 上にアップロードするだけで定期的にスキャンを実行し、脆弱性情報を収集する仕組みを提供していますので、そちらを利用することを推奨します。

Armadillo Twin による継続的な SBOM スキャンについては、Armadillo Twin ユーザーマニュアル「SWU イメージを管理する」 [https://manual.armadillo-twin.com/management-swu-image/]を参照してください。

11.2. セキュリティアップデートの優先度の決定

脆弱性の収集方法について紹介しましたが、必ずしも検出した全ての脆弱性に対応する必要はありません。実際には脆弱性や製品の性質によって、対応する必要のない脆弱性や、逆に他の脆弱性よりも優先して対応する必要のある脆弱性があります。

セキュリティアップデートの優先度は、脆弱性の深刻度や影響範囲、製品の利用状況などを考慮して決定します。脆弱性には基本的に CVE (Common Vulnerabilities and Exposures) と呼ばれる識別子が付与されており、この識別子を用いて検出した脆弱性のより詳細な情報を調べることができます。また、脆弱性には深刻度を表す CVSS (Common Vulnerability Scoring System) スコアが付与されていることもあります。この CVSS スコアも脆弱性の深刻度を判断するための参考になります。

SBOM をスキャンして取得できる情報と、開発した IoT 機器の利用状況を考慮して、セキュリティアップデートの優先度を決定することが重要です。

11.3. アットマークテクノの方針

この節では、アットマークテクノがどのように脆弱性情報を収集し、セキュリティアップデートの優先度を決定しているかについて説明します。開発ベンダーが社内体制を整備する際の参考情報としたり、製品のステークホルダーとしてアットマークテクノの方針を理解したりするためにご活用ください。

11.3.1. アットマークテクノの脆弱性情報の収集方法

アットマークテクノでは、 ABOS のリリースイメージの SBOM を毎日スキャンし、脆弱性情報を収集しています。

また、以下のメーリングリストを購読することでも脆弱性情報を収集しています。

- Open Source Software Security Wiki
  - https://oss-security.openwall.org/subscribe
- Debian Mailing Lists: debian-security-announce
  - https://lists.debian.org/debian-security-announce/
- Alpine Linux announcements
  - https://lists.alpinelinux.org/~alpine/announce

さらに、アットマークテクノでは第三者からの脆弱性情報の報告を受け付けるための窓口(https://www.atmark-techno.com/form/vulnerability_report)も設けています。

11.3.2. アットマークテクノの脆弱性の分析方法

アットマークテクノでは、収集した脆弱性情報を以下の基準に従って分析し、優先度を決定しています。

- 基本的には収集した脆弱性情報を一つ一つ確認する
  - アナウンス内容およびソースコード、修正パッチの内容を確認し、重要度を判断する
- 発見された脆弱性の数が多い場合は、 CVSS (Common Vulnerability Scoring System) のスコアを参考に深刻度を測る

11.3.3. アットマークテクノの脆弱性に対する対応優先度

アットマークテクノでは、発見された全ての脆弱性に対して対応を実施するわけではありません。以下の基準に従って、対応の優先度を決定しています。

- 任意コマンド実行の可能性がある脆弱性
  - 対応優先度が高い脆弱性と判断し、直ちに修正を行ってお客様に可及的速やかに提供する
  - アップデートニュースには以下を記載する
    - 脆弱性の内容と、それがもたらす影響
    - 脆弱性の対象となるかどうかの判断方法
    - (可能であれば)アップデート以外のワークアラウンド
- 製品の動作の妨害につながる脆弱性
  - 悪意のある攻撃者が、その脆弱性を利用して妨害を行う難易度が低いと判断した場合は、即時アップデートで対応する
  - 妨害を行う難易度が高いと判断した場合は、月末の定期アップデートで対応する

    - アップデートニュースに脆弱性の内容と影響を記載するが、ワークアラウンドは基本的に記載しない

- 製品の運用上無関係な脆弱性
    - 何も対応を行わない

12. S1.1-10

S1.1-10 では、IoT 製品の型番をユーザへ提供することを要求しています。

製品のモデル名称は、製品上のラベルまたは GUI アプリケーションなどのソフトウェアを介して、ユーザに対して明確に認識可能でなければなりません。

12.1. 型番提供方法の選択

IoT 製品の型番は、以下のいずれかの方法でユーザへ提供する必要があります。

評価項目 1: IoT 製品本体に、IoT 製品の型番を直接記載すること

評価項目 2: IoT 製品の GUI、ウェブ UI 等や、IoT 製品に付帯するソフトウェア、アプリケーション（スマホアプリなど）の GUI、ウェブ UI 等から、ユーザが型番を認識できるようにすること

12.2. IoT 製品に製品型番を表示させる方法

12.2.1. 物理的な製品型番の記載

IoT 製品本体に製品型番を直接記載する方法です。

開発した実機にケースを取り付ける場合は、型番を記載したシールやラベルを貼り付けることで要件を満たすことができます。

12.2.1.1. BTO サービスによるシール・ラベル対応

アットマークテクノでは、ケースありモデルについては BTO サービスにおいて、お客様用の型番シールを貼り付けるサービスを提供しています。

貼り付けるラベルの内容は、お客様の要件に応じてカスタマイズすることが可能です。

このサービスを利用することで、製品出荷時点で適切な型番表示を実現できます。

12.3. GUI に製品型番を表示させる方法

12.3.1. ABOS Web での型番表示

ABOS Web を使用して、製品型番をユーザに表示する方法を説明します。

ABOS Web にログイン後、「設定管理」画面の「製品型番」欄に 24 文字以下で製品型番を入力してください。

図 12.1 「製品型番」欄

「製品型番」欄に例えば、"ABCD-1234-EFGH" のように入力して、一番下の「アップロード」ボタンをクリックします。

再ログインすると、ヘッダのタイトルの下に製品型番が表示されるようになります。

図 12.2 製品型番の表示例

12.3.2. その他の GUI での実装

開発者が独自に開発した GUI アプリケーションや WebUI においても、製品型番を表示することができます。

・ デスクトップアプリケーションの情報画面

- Web アプリケーションのシステム情報ページ
- モバイルアプリケーションの設定画面

12.4. 開発者自身で対処する場合

開発者が独自の方法で型番表示を実装する場合の留意点について説明します。

12.4.1. ソフトウェア的な型番識別方法

物理的なシール・ラベルを使用しない場合は、ソフトウェア的に型番を識別する方法を実装する必要があります。

例えば以下のような実装方法があります。

- 設定ファイルに型番情報を記録
- ファームウェア内に型番情報を埋め込み
- 起動時に型番情報を読み込む仕組みの実装

12.4.2. 実装時の考慮事項

独自実装する場合は、以下の点を考慮してください。

- **ユーザアクセシビリティ**: ユーザが容易に型番を確認できること
- **情報の正確性**: 表示される型番が実際の製品と一致していること
- **表示の永続性**: システム再起動後も型番情報が維持されること
- **セキュリティ**: 型番情報が不正に改変されないこと

12.5. 実機テストでの評価

S1.1-10 は実機テストによる評価項目です。

評価では、以下のいずれかの方法で型番が確認できることが求められます。

- **評価項目 1**: IoT 製品本体を確認し、IoT 製品の型番が記載されていることを確認できること
- **評価項目 2**: IoT 製品の GUI、ウェブ UI 等や、IoT 製品に付帯するソフトウェア、アプリケーションの GUI、ウェブ UI 等に実際にアクセスすることで、当該 IoT 製品の型番を確認できること

実装した型番表示方法が、実際にユーザによって確認可能であることを事前に検証してください。

13. S1.1-11

S1.1-11 では、IoT 製品のストレージに保存される守るべき情報資産をセキュアに保存することを要求しています。

製品のストレージにある機密セキュリティパラメータは、製品によってセキュアに保存されなければなりません。

守るべき情報資産がどういったものを指すかについては、評価ガイドおよびチェックリストをご参照ください。

13.1. ストレージに保存する守るべき情報資産をリストアップする

まず、システムで取り扱う守るべき情報資産を特定し、リストアップします。

13.1.1. 情報資産の分類

守るべき情報資産を以下の観点で分類します。

機密性保護が必要な情報資産

外部に不正に漏えいしないように保護する必要がある情報。

- 暗号化鍵
- 認証情報（パスワード、証明書など）
- 個人情報
- センサーデータ（機密性が求められる場合）

完全性保護が必要な情報資産

改ざんされないように保護する必要がある情報。

- 設定ファイル
- ファームウェアイメージ
- ログファイル
- デバイス固有の識別情報

真正性確認が必要な情報資産

情報資産の作成者や提供者をなりすまされないようにする必要がある情報。

- デジタル署名
- 証明書

・認証トークン

13.2. Armadillo における守るべき情報資産

標準状態の Armadillo では、以下の情報資産が守るべき情報資産として挙げられます。

・通信機能に関する設定情報

・/etc/NetworkManager/system-connections/ 以下のファイル

13.3. リストアップした守るべき情報資産の保存先を確認する

特定した情報資産について、実際の保存先を確認します。

13.3.1. Armadillo における標準的な保存先

・オンボード eMMC

Armadillo の標準構成では、主にオンボード eMMC に情報資産を保存します。

「13.2. Armadillo における守るべき情報資産」で挙げた標準状態の ABOS における守るべき情報資産も eMMC 上に保存します。

・外部ストレージメディア

SD カードや USB メモリ等の外部ストレージメディアも使用可能です。

13.4. Armadillo 上のストレージ以外にある守るべき情報資産の保護対策を確認する

外部ストレージメディアに保存される情報資産については、以下の保護対策を検討します。

13.4.1. 想定される脅威

JC-STAR★1 では、以下のような非正規のアクセス方法による脅威を想定しています。

・マルウェアが不正な権限でファイルを窃取

・廃棄された IoT 機器のストレージに直接アクセス

・スクリプトキディレベルの攻撃

13.4.2. セキュアな保存の実現方法

以下のいずれかに類する保護対策またはそれ以上の対策により「セキュアな保存」を実現します。

セキュアな保存の実現方法のために使用する暗号化方式、署名方式、ハッシュ関数については、以下の資料を参照してください。

・CRYPTREC 暗号リスト: https://www.cryptrec.go.jp/list.html

13.4.2.1. 評価項目 1: 暗号化による機密性保護

機密性の保護が必要な守るべき情報資産は、「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）」のうち「電子政府推奨暗号リスト」に記載の暗号技術を採用した暗号化方式によって暗号化された上で保存します。

Armadillo では、セキュアエレメントである SE050 を使用することで、セキュアな暗号化を実現できます。SE050 は、暗号鍵の生成や管理を安全に行うことができ、機密性の高い情報を保護するための強力な手段となります。

秘密鍵は SE050 内で安全に管理され、外部に漏洩することはありません。そのため、もし暗号化したファイルと SE050 内の秘密鍵にアクセスする参照鍵が搾取された場合でも、暗号化されたデータを復号することはできません。

以下にコンテナ上で SE050 を使用して暗号化を行う例を示します。

まず、コンテナ自動起動用設定ファイルを作成します。

```
[armadillo ~]# vi /etc/atmark/containers/plug-and-trust.conf
set_image docker.io/alpine
add_armadillo_env
add_devices "${AT_SE_PARAM%:*}"
add_volumes /etc/apk:/etc/apk:ro
set_command sleep infinity
```

図 13.1 plug-and-trust.conf ファイルの作成

ベースとなるコンテナイメージをプルします。

```
[armadillo ~]# abos-ctrl podman-rw pull docker.io/alpine
```

図 13.2 alpine コンテナのインストール

作成したコンテナ自動起動用設定ファイルを使用して、コンテナを起動します。

```
[armadillo ~]# podman_start plug-and-trust
[armadillo ~]# podman exec -it plug-and-trust sh
```

図 13.3 コンテナの起動

コンテナの中で必要なパッケージをインストールして、環境設定を行います。

```
[container /]# cd
[container ~]# apk add se05x-tools plug-and-trust-tools
```

図 13.4 必要パッケージのインストール

```
[container ~]# export OPENSSL_CONF=/etc/plug-and-trust/openssl11_sss_se050.cnf
[container ~]# export EX_SSS_BOOT_SSS_PORT="$AT_SE_PARAM"
```

図 13.5 環境変数の設定

正しく SE050 が動作しているか確認します。

```
[container ~]# se05x_GetInfo
:（省略）
```

図 13.6 SE050 動作確認

SE050 内に保存する RSA 鍵ペアを生成します。

```
[container ~]# openssl genpkey -algorithm RSA -out private.pem -pkeyopt rsa_keygen_bits:3072
[container ~]# openssl pkey -in private.pem -pubout -out public.pem
```

図 13.7 RSA 鍵ペア生成（3072 ビット、128 ビットセキュリティ）

SE050 に秘密鍵を登録します。登録した秘密鍵は、ルートファイルシステム上から削除します。

```
[container ~]# se05x_setkey -f 0x10 private.pem "$AT_SE_PARAM"
[container ~]# rm private.pem
```

図 13.8 秘密鍵の SE050 登録と削除

登録した秘密鍵を参照するための参照鍵を生成します。

```
[container ~]# se05x_getkey 0x10 refkey.pem "$AT_SE_PARAM"
[container ~]# ls
refkey.pem public.pem
```

図 13.9 参照鍵の生成

ここからは、 ATDE など Linux OS の PC 上での操作となります。

公開鍵 public.pem を ATDE に転送後、OAEP パディング方式でデータを暗号化します。

```
[ATDE ~]# echo "This is test" > message.txt
[ATDE ~]# openssl pkeyutl -encrypt -in message.txt -out message.enc -pubin -inkey public.pem -
pkeyopt rsa_padding_mode:oaep
[ATDE ~]# ls
message.enc message.txt public.pem
```

図 13.10 公開鍵による暗号化

上記で暗号化したデータを Armadillo に転送し、 SE050 による復号を試してみます。暗号文 message.enc を Armadillo に転送後、message.enc を以下のコマンドで復号します。

```
[container ~]# openssl pkeyutl -decrypt -in message.enc -out message.dec -inkey refkey.pem -pkeyopt ⏎
rsa_padding_mode:oaep
[container ~]# cat message.dec
This is test
```

図 13.11 SE050 秘密鍵による復号

上記の例では、message.enc と refkey.pem が搾取されても、refkey.pem は Armadillo の SE050 内の秘密鍵への参照鍵にすぎないため、他のデバイスでは復号できません。

SE050 についての詳細は製品マニュアルを参照してください。

13.4.2.2. 評価項目 2: デジタル署名による完全性保護

完全性の保護が必要な守るべき情報資産は、「電子政府推奨暗号リスト」に記載の暗号技術を採用した署名によってデータの完全性が確認できる形で保存します。

13.4.2.3. 評価項目 3: ハッシュ関数による完全性保護

完全性の保護が必要な守るべき情報資産は、「電子政府推奨暗号リスト」に記載の暗号技術を用いたメッセージダイジェストによってデータの完全性が確認できる形で保存します。

13.4.2.4. 評価項目 4: セキュア領域での保存

守るべき情報資産を以下のセキュア領域に保存します。

- 仮想化技術によるセキュア領域
- OS の機能として提供されるサンドボックス
- セキュリティチップによるセキュア領域（SE050 等）

13.4.2.5. 評価項目 5: 容易に取り外せないストレージでの保存

守るべき情報資産を、IoT 機器に組み込まれた容易に取り外せないストレージ領域にあって、外部から呼び出すインタフェースを経由した直接的なデータの読み書きができない領域に保存します。

Armadillo における守るべき情報資産の保護対策

「13.2. Armadillo における守るべき情報資産」で挙げた、標準状態の ABOS における守るべき情報資産は、以下のような保護対策によりこの評価項目 5 を満たすため、セキュアに保存されていると判断できます。

- オンボード eMMC という容易に取り外せないストレージに保存している
- root および abos-web-admin ユーザしか読み書きできない権限設定をしている
- root および abos-web-admin ユーザは「4. S1.1-02」で規定されている強度の高いパスワードで保護している

上記より、守るべき情報資産は容易に取り外せないストレージ領域にあって、外部から呼び出すインタフェースを経由した直接的なデータの読み書きができない領域に保存されていると判断できます。

13.5. 技術文書に保護対策を明記する

S1.1-11 の要件を満たすため、技術文書には以下の内容を明示する必要があります。

13.5.1. 記載すべき内容

- 守るべき情報資産のリスト
- 各情報資産の保存先
- 採用した保護対策（評価項目 1〜5 のいずれか）
- 使用する暗号技術の詳細
- セキュア領域の実装方法

13.5.2. Armadillo を使用する場合の技術文書記載例

Armadillo の標準構成の場合、以下のように記載できます。

「守るべき情報資産は、Armadillo のオンボード eMMC に保存されます。このストレージは容易に取り外せない構造となっており、root および abos-web-admin アカウントのパスワード強度が保証されています。これにより、外部から呼び出すインタフェースを経由した直接的なデータの読み書きができない領域での保存を実現し、評価項目 5 の要件を満たしています。」

14. S1.1-12

S1.1-12 では、ネットワーク経由で伝送される守るべき情報資産を、情報の盗聴から保護することを要求しています。

守るべき情報資産や必要な保護対策の定義は、チェックリストや評価ガイドを参照してください。

14.1. ネットワーク伝送経路のリストアップ

守るべき情報資産がネットワーク経由で伝送されるかを確認します。

標準状態の ABOS において守るべき情報資産が伝送される経路は以下の通りです。

- ABOS Web
- Armadillo Twin (使用している場合)

これらに加えて、開発した IoT 製品で下記のようなネットワーク経由で守るべき情報資産が伝送される経路をリストアップしてください。

- 開発したアプリケーションによるサーバとの通信
- クラウドサービス（AWS 等）との通信
- 他の IoT 機器との通信
- モバイルアプリとの通信

14.2. 保護対策の確認

守るべき情報資産をネットワーク経由で伝送する場合、盗聴を防ぐ保護対策が必要です。

14.2.1. Armadillo での実装

ABOS Web

ABOS Web はデフォルトで TLS 1.2 および 1.3 に対応しており、公開鍵暗号方式は ECDSA (secp384r1) を使用しています。

Armadillo Twin

Armadillo Twin は Armadillo 本体との通信と、 Armadillo Twin のページを表示する PC 間との通信でそれぞれ異なる TLS バージョンを使用します。

- Armadillo と Armadillo Twin 間: TLSv1.2
- Armadillo Twin と PC 間: TLSv1.2 および v1.3

暗号化スイートは TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 などを使用しています。

14.2.2. 開発したアプリケーションでの実装

以下の対策を実装してください。

- TLS 等の暗号化通信プロトコルを使用
- CRYPTREC 推奨暗号リストに準拠した暗号技術を採用

より高いセキュリティが必要な場合、 SE050 を利用して伝送するファイル自体を暗号化することを推奨します。

14.2.3. SE050 による暗号化の利点

SE050 を使用してファイルを暗号化した場合、以下の利点があります。

- 暗号化キーがハードウェア内に安全に保管される
- 他の Armadillo や PC では復号できない（デバイス固有の暗号化）
- より強固なセキュリティを実現

SE050 の使用方法に関しては製品マニュアルを参照してください。

14.3. 技術文書への明記

S1.1-12 の要件を満たすため、技術文書に以下のような保護対策の詳細を記載する必要があります。

- ネットワーク経由で伝送する守るべき情報資産の有無
- 採用した保護対策（TLS、ファイル暗号化、VPN など）
- 保護対策が初期設定で有効であること
- （保護された通信環境で利用する場合）ユーザ向けの利用環境制限の明示

15. S1.1-13

S1.1-13 では、IoT 製品のセキュリティ強化のため、不要なインタフェースの無効化と脆弱性検査の実施を要求しています。これにより、外部からのサイバー攻撃を受けるリスクを低減できます。

以下の 2 つの基準を満たす必要があります。

- 不要かつ攻撃リスクがあるインタフェース（TCP/UDP ポート、Bluetooth、USB）の無効化
- 脆弱性スキャンツールによる既知の脆弱性検査の実施（攻撃に悪用される可能性がある脆弱性が検出されないこと）

15.1. TCP/UDP ポートの確認と管理

JC-STAR★1 では、攻撃面を最小化するために、不要な TCP/UDP ポートを無効化することを要求されます。IoT 機器の利用上必要なポートのみを開放し、不要なポートは閉じた状態にする必要があります。

15.1.1. ABOS がデフォルトで開放しているポート

標準状態の ABOS において、外部からの通信に対して開放しているポートは以下の通りです。

表 15.1 ABOS のデフォルト開放ポート

ポート番号	プロトコル	使用目的
58080	TCP	ABOS Web
5353	UDP	avahi (mDNS)

ABOS Web が不要な場合は、ABOS Web を無効化することで、ポート 58080 を閉じることができます。ABOS Web を無効化する方法は、 「3.2.1. ABOS Web を無効化する」を参照してください。

avahi (mDNS) は、ローカルネットワーク内で Armadillo を検出するために使用されます。開発完了後は使用しないので、avahi を無効化することを推奨します。avahi を無効化するには、以下のコマンドを実行します。

```
[armadillo ~]# rc-update | grep avahi-daemon ❶
         avahi-daemon |      default
[armadillo ~]# rc-service avahi-daemon status ❷
 * status: started
[armadillo ~]# rc-service avahi-daemon stop ❸
avahi-daemon             | * Stopping avahi-daemon ... [ ok ]
[armadillo ~]# rc-update del avahi-daemon ❹
 * service avahi-daemon deleted from runlevel default
[armadillo ~]# persist_file -d /etc/runlevels/default/avahi-daemon ❺
```

図 15.1 avahi-daemon を停止する

❶ OpenRC に avahi のサービスが登録されていることを確認します。

❷ avahi のサービスが起動していることを確認します。

❸ avahi のサービスを停止します。

❹ サービスを管理している OpenRC から avahi のサービスの登録を解除します。

❺ サービス設定ファイルの削除を永続化します。

15.1.2. 開発した IoT 製品でのポート管理

開発者が新たにポートを開放する際は、以下の点を確認してください。

- ポート番号とプロトコルを明確にする
- 開放する理由と利用目的を明確にする
- 不要なポートは開放しない

15.1.3. nmap を使用したポート確認

IoT 機器が開放している TCP/UDP ポートは、外部から機器に対して nmap というツールを使用することで確認できます。

ATDE ではデフォルトで nmap がインストールされていないため、以下のコマンドを実行してインストールすることで使用できます。

```
[ATDE ~]$ sudo apt install nmap
```

図 15.2 nmap のインストール

nmap コマンドではプロトコルに合わせて以下のコマンドでポートスキャンを行うことができます。

```
[ATDE ~]$ nmap -sT <Armadillo の IP アドレス>
```

図 15.3 TCP ポートの確認

```
[ATDE ~]$ nmap -sU --min-rate 1000 <Armadillo の IP アドレス>
```

図 15.4 UDP ポートの確認

15.2. HTTP/HTTPS プロトコルの脆弱性テスト

IoT 機器に HTTP/HTTPS プロトコルを使用する設定や機能が実装されている場合、脆弱性テストを実施して特定の脆弱性が検知されないことを確認する必要があります。

標準状態の ABOS では、 ABOS Web にて HTTPS プロトコル、Armadillo Twin にて HTTP/HTTPS プロトコルを使用しています。

開発者が HTTP/HTTPS プロトコルを使用する設定や機能を追加した場合は、その追加した設定・機能に対しても脆弱性テストが必要です。確認すべき項目は、チェックリストや評価ガイドを参照してください。

検証に使用できる脆弱性検査ツールの例

ウェブアプリケーションに対して脆弱性の検査を行うツールは有償・無償ともに様々なものがあります。その一例を示します。

- OWASP ZAP
- Burp Suite
- Nessus
- OpenVAS

15.3. Bluetooth プロファイルの確認と管理

IoT 機器が、 Bluetooth 機能を使用する場合は、以下を確認してください。

- 使用する Bluetooth プロファイルと利用目的を明確にする
- 不要なプロファイルは無効化または削除する
- 廃止された Bluetooth プロファイルが使用できないことを確認する

15.3.1. ABOS における Bluetooth プロファイル

標準状態の ABOS には、 Bluetooth プロファイルは含まれていません。

15.4. USB デバイスクラスの確認と管理

IoT 機器が USB デバイスを使用する場合、利用しない USB デバイスクラスは無効化する必要があります。

15.4.1. USB 接続制御機能

ABOS は、 USB デバイス固有の情報をもとに、そのデバイスの接続を許可・拒否する USB 接続制御機能を搭載しています。IoT 機器に接続してもいいと判断した USB デバイスのみ許可リストに入れ、それ以外の USB デバイスは接続を拒否します。この機能を使用することで、意図しない不正な USB デバイスの接続を防止できます。

ABOS の USB 接続制御機能は、大きく分けて以下の 2 つの USB デバイスが持つ情報によって接続の許可・拒否設定を行うことができます。

- USB デバイス固有の情報（ベンダー ID、プロダクト ID、シリアル番号など）
- USB デバイスクラス（HID、Mass Storage など）

15.4.2. 標準状態の ABOS において接続を許可する USB デバイス

標準状態の ABOS では、開発者が開発を行うために最低限必要であろう USB デバイスのみ接続を許可しています。具体的な標準状態で接続を許可している USB デバイスクラスは各製品マニュアルを参照してください。

15.4.3. ABOS Web を使用した USB 接続制御機能の設定確認

USB デバイスクラスの設定は、ABOS Web の「USB 接続制御」画面で確認・変更できます。

図 15.5 USB 接続制御画面

15.4.4. USB デバイスの接続拒否手順

接続が許可されている USB デバイスの接続を拒否する手順は以下の通りです。

図 15.6 許可済みの USB デバイスルールを選択

「図 15.6. 許可済みの USB デバイスルールを選択」に示すように、許可済みの USB デバイスルールを選択した状態で "削除" ボタンを押すと、「図 15.7. 許可ルールを削除する」に示す画面に遷移します。

図 15.7 許可ルールを削除する

確認画面が表示されます。問題なければ改めて "削除" ボタンを押して許可ルールを削除してください。

USB デバイスの許可ルールが削除されると、そのルールに該当するデバイスの接続は永続的に拒否されます。「接続済みの USB デバイス」画面では、「可否」が block に変更されます。

15.4.5. USB デバイスクラス単位での接続許可・拒否

「図 15.8. 許可済みの USB デバイスクラス一覧」の「許可済みの USB デバイスクラス」では、デバイスクラス単位で接続設定を管理できます。

図 15.8 許可済みの USB デバイスクラス一覧

一覧には現在許可されている USB デバイスクラスが表示されます。

15.4.6. 特定の USB デバイスクラスの接続を許可する

IoT 機器が動作するために必要な USB デバイスクラスは接続を許可する必要があります。特定の USB デバイスクラスの接続を許可する場合は、 ABOS Web の「USB 接続制御」画面の「許可済みの USB デバイスクラス」欄にて、 "追加" ボタンを押してください。

"追加" ボタンを押すと、「図 15.9. USB デバイスクラスの追加」に示す画面が表示されます。追加可能な USB デバイスクラスを選択し、"追加" ボタンを押してください。

図 15.9 USB デバイスクラスの追加

USB デバイスクラスを新たに追加する際は、以下を確認してください。

- 使用する USB デバイスクラスと利用目的を明確にする
- 不要なデバイスクラスは無効化する
- 追加したデバイスクラスを技術文書に記載する

15.4.7. 特定の USB デバイスクラスの接続を拒否する

IoT 機器が動作するために不要な USB デバイスクラスは接続を拒否する必要があります。特定の USB デバイスクラスの接続を拒否する場合は、 ABOS Web の「USB 接続制御」画面の「許可済みの USB

デバイスクラス」欄にて、 "削除" ボタンを押してください。接続を拒否したいデバイスクラスを選択し、 "削除" ボタンを押すことでそのデバイスクラスの接続を拒否できます。削除手順は「図 15.7. 許可ルールを削除する」と同様です。

15.4.8. IoT 機器に必要な USB デバイスのみを許可する

上記で紹介した USB 接続制御機能を使用して、IoT 機器に必要な USB デバイスのみを許可し、その他の接続は拒否するように設定してください。接続を許可した USB デバイスは、使用目的を明確にしてください。

15.5. 技術文書への記載要件

S1.1-13 の要件を満たすため、技術文書に各評価項目を満たすような内容を明示してください。評価項目の詳細はチェックリストや評価ガイドを参照してください。

16. S1.1-14

S1.1-14 では、停電やネットワーク停止により IoT 機器の電源が OFF になった後、復帰時に認証値及びアップデートが維持されることを要求します。

16.1. 変更可能な認証値をリストアップする

まず、 IoT 製品における変更可能な認証値をリストアップします。

標準状態の ABOS における変更可能な認証値に該当するものは以下の通りです。

- root ユーザのログインパスワード
- atmark ユーザのログインパスワード (initial_setup で設定した場合)
- ABOS Web のログインパスワード
- WLAN の SSID とパスワード
- インストール済み各種暗号化鍵・証明書

開発者は、これらの項目に加えて、新たに認証値などを追加した場合には、確認項目を追加する必要があります。

インストール済み各種暗号化鍵・証明書の発見

インストール済みの各種暗号化鍵・証明書は、以下のコマンドで検索できます。

```
[armadillo ~]# find / -type f ¥( ¥
   -iname "*.key" -o ¥
   -iname "*.pem" -o ¥
   -iname "*.crt" -o ¥
   -iname "*.cer" -o ¥
   -iname "*.der" -o ¥
   -iname "*.p12" -o ¥
   -iname "*.pfx" -o ¥
   -iname "id_rsa*" -o ¥
   -iname "id_dsa*" -o ¥
   -iname "id_ecdsa*" -o ¥
   -iname "id_ed25519*" ¥
   ¥)
```

16.2. ソフトウェアのアップデート情報の確認方法

ABOS における、ソフトウェアのバージョン情報の確認方法は、「8.4. バージョンの確認方法を決定する」で紹介しているので、そちらを参照してください。

16.3. 再起動時にファイルが削除されないための TIPS

16.3.1. ルートファイルシステム設定変更

Armadillo Base OS では、一部のディレクトリを除いてオーバーレイファイルシステム（overlayfs）を使用しています。このため、rootfs の内容を変更したり、新規にファイルを追加して、それらを再起動後にも維持したい場合は以下のコマンドを実行してください。

```
[armadillo ~]# persist_files -P <file path>
```

図 16.1 オーバーレイファイルシステムの更新

16.3.2. コンテナでの注意事項

ABOS においてコンテナは起動時に、コンテナイメージから再作成されるため、結果としてコンテナ内のデータは電源断で失われます。

そのため、コンテナ内で動的に作成されるデータや設定は以下の ABOS のユーザーデータディレクトリをコンテナにマウントして、認証値はそのディレクトリに保存してください。

表 16.1 電源を切っても保持されるディレクトリ(ユーザーデータディレクトリ)

ディレクトリ	備考
/var/app/volumes	SWUpdate の最中や後も保持され続けます。ロールバックが発生しても、アップデート前の状態には戻りません。ログやデータベースなど、アプリケーションが動作中に作成し続けるようなデータはこのディレクトリに保存してください。
/var/app/rollback/volumes	SWUpdate の最中や後も保持され続けます。ロールバックが発生すると、アップデート前の状態に戻ります。コンフィグファイルなど、アプリケーションのバージョンに追従してアップデートするようなデータはこのディレクトリに保存してください。

16.4. 実機テストを行う

本要件では実機テストを行う必要があります。開発した IoT 機器に対してテストを実施し、その結果をエビデンスとして記録・保存しておく必要があります。

テストは以下の流れで実施するとよいでしょう。

1. テスト前の各種設定状態を確認・記録
2. 機器の電源を切断 or ネットワークを切断
3. 再起動 or ネットワークを復帰させる
4. 各種設定状態が維持されていることを確認

17. S1.1-15

S1.1-15 は、 IoT 機器の利用が終了し、廃棄・譲渡された後にデータが窃取されないような対応を求める要件です。

S1.1-15 はドキュメント評価と実機テストによる評価のどちらも対象になります。

17.1. 製品内の削除すべきデータを定義する

一般に IoT 機器は、アプリケーション本体やライブラリなどの、製品が製品として動作するために必要なアプリケーション部分と、エンドユーザーが製品を利用しているときに保存されるログやデータなどのデータ部分に分かれます。本要件では、前者のアプリケーション部分を残しつつ後者のデータ部分を削除することで、製品の出荷状態に戻し、廃棄や譲渡を行っても IoT 機器が稼働中に収集したデータが流出しないようにすることを期待しています。

この要件を満たすためには、まずは製品の中の残したいアプリケーション部分と、削除したいデータ部分を明確に定義する必要があります。

削除したいデータ部分として、具体的には以下のようなものが考えられます。

- IoT 機器が収集した個人情報
- IoT 機器が収集したログ情報
- ユーザーが設定した項目
- ユーザーが設定したパスワードなどの認証値
- 利用中に取得した暗号鍵やデジタル署名

上記以外にも様々なデータが削除対象となる可能性がありますので、製品の特性に応じて決定してください。

17.2. Armadillo 内の指定したデータを削除する方法

ABOS では、容易にデータの削除を行うための「ユーザーデータ削除機能」を提供しています。

ユーザーデータ削除機能は、以下のいずれかの方法で実行できます。

- `abos-ctrl reset-default` コマンドを実行する
- ABOS Web の UI から「ユーザーデータ削除」を実行する
- ABOS Web の REST API を利用して、ユーザーデータ削除を実行する

ユーザーデータ削除機能では、デフォルトで以下のデータを削除します。

- ネットワーク設定
  - LAN、WLAN、WWAN の設定を全て削除します。WLAN はクライアント設定とアクセスポイント設定の両方を削除します。

- DHCP 設定
- NAT 設定
- VPN 設定
- NTP 設定
- /var/app/volumes ディレクトリ下のファイルを全て
- /var/log ディレクトリ下のファイルを全て

また、上記のデータに加えて /etc/atmark/reset_default_list.txt ファイルに記載されているファイルも削除します。書き方のサンプルファイルとして、/etc/atmark/reset_default_list.txt.example も用意されていますので、リネーム・編集してご利用ください。

17.2.1. ユーザーデータ削除機能の削除レベル

JC-STAR★1 では、どの程度までデータを削除すべきかを以下のように定義しています。

単純な非侵襲のデータ回復技術（市販のデータ復旧ソフトによるサルベージ等）から保護できるセキュリティレベル（NIST SP800-88 Rev.1 での「Clear」レベル）での削除を求める。

ABOS が提供するユーザーデータ削除機能は、NIST SP800-88 Rev.1 での「Clear」レベルを満たすように設計されています。

ABOS では、データの保存領域のファイルシステムとして、 ext4 と btrfs を採用しています。ext4 の領域に保存されたデータは、ユーザーデータ削除機能を実行すると、 discard 処理を含むフォーマットを行いデータをセキュアに削除します。btrfs の領域に保存されたデータは、ユーザーデータ削除機能を実行すると、当該のファイルや subvolume を削除した後、 sync と fstrim を実行することで metadata を含むデータをセキュアに削除します。

17.3. ABOS におけるユーザーデータの保存場所

「17.2. Armadillo 内の指定したデータを削除する方法」で紹介したユーザーデータ削除機能は、任意の場所のデータを削除できますが、データの保存場所として以下を推奨しています。

表 17.1 ABOS におけるユーザーデータの推奨保存場所

ディレクトリ	保存するデータの内容
/var/app/volumes	ログやデータベースなど、アプリケーションが動作中に作成するデータはこのディレクトリに保存してください。
/var/app/rollback/volumes	コンフィグファイルなど、アプリケーションのバージョンに追従してアップデートするようなデータはこのディレクトリに保存してください。

17.4. アプリケーションからユーザーデータ削除機能を利用する

前述の通り、ユーザーデータ削除機能はコマンドラインや ABOS Web の UI から実行できますが、アプリケーションから実行する場合は ABOS Web の REST API を利用します。REST API 実行用の token には、 ResetDefault 権限を持たせ、それ以外の不要な権限は持たせないようにしてください。

具体的な設定方法や実行方法は 各製品マニュアルを参照してください。

18. S1.1-16

S1.1-16 では、製品に関する情報提供をユーザに対して行うことを要求しています。

ここで言う製品に関する情報とは、以下を指します。

- 初期設定の方法やパスワード変更の実施手順等、サイバーセキュリティに影響が生じる設定や使用方法について、安全に利用できる手順を示した情報
- 製品のセキュリティアップデートの重要性や必要性、アップデートを行わない場合の影響等を周知する仕組みや実施方法
- アップデートを行わなかったときに想定される事故や障害・一般的に想定される事故や障害に対する免責事項
- 製品のサポート期限
- 製品内に守るべき情報資産が残留したまま廃棄や中古販売されることで想定されるリスクや、データ消去を含む製品の安全な利用終了方法

18.1. 参考情報

標準状態の ABOS では、製品に関する情報提供を以下のように行っています。例として Armadillo-IoT ゲートウェイ G4 の製品マニュアルへのリンクを示します。情報の提供方法や方針の参考にしてください。

- **初期設定の方法やパスワード変更の実施手順等**
  - 各製品マニュアルに記載
  - Armadillo の初期化と ABOS のアップデート [https://manual.atmark-techno.com/armadillo-iot-g4/armadillo-iotg-g4_product_manual_ja/ch03.html#sct.initialize-armadillo-update-abos]
  - Armadillo に初期設定をインストールする [https://manual.atmark-techno.com/armadillo-iot-g4/armadillo-iotg-g4_product_manual_ja/ch03.html#sct.setup-armadillo-with-vscode]
- **製品のセキュリティアップデートの重要性や必要性、アップデートを行わない場合の影響等**
  - 各製品マニュアルに記載
  - Armadillo Base OS のメンテナンスポリシーとアップデートの推奨 [https://manual.atmark-techno.com/armadillo-iot-g4/armadillo-iotg-g4_product_manual_ja/ch02.html#sct.armadillo-base-os-update-policy]
- **アップデートを行わなかったときに想定される事故や障害・一般的に想定される事故や障害に対する免責事項**
  - 各製品マニュアルに記載
  - ソフトウェア使用に関しての注意事項 [https://manual.atmark-techno.com/armadillo-iot-g4/armadillo-iotg-g4_product_manual_ja/ch01.html#sct.caution-software]

- **製品のサポート期限**
  - Armadillo Base OS 対応製品のサポート期限 [https://armadillo.atmark-techno.com/abos-support-timeline]
- **情報資産が残留したまま廃棄・転売されるリスク、製品の安全な利用終了方法**
  - 各製品マニュアルに記載
  - Armadillo を廃棄する [https://manual.atmark-techno.com/armadillo-iot-g4/armadillo-iotg-g4_product_manual_ja/ch05.html#sct.discard-armadillo]
  - Armadillo の初期化と ABOS のアップデート [https://manual.atmark-techno.com/armadillo-iot-g4/armadillo-iotg-g4_product_manual_ja/ch03.html#sct.initialize-armadillo-update-abos]

19. チェックリストと申請書を作成する

この章では、IPA に提出するチェックリストと申請書を作成する際の注意点等を紹介します。

19.1. 申請書: IoT 製品の関連企業について

JC-STAR★1 の申請書には、以下の項目があります。

- 申請企業、申請代行企業、製造ベンダー、ファームウェア開発企業のいずれにおいても過去 5 年以内に我が国の法令や国際的に受け入れられた基準等に違反したまたはその疑いをかけられたことはないか。
- 申請企業、申請代行企業、製造ベンダー、ファームウェア開発企業のいずれにおいても申請製品のサイバーセキュリティ確保について外国の法的環境等により影響を受ける懸念はないか。

Armadillo を用いて開発された IoT 機器において、アットマークテクノはファームウェア開発企業として位置付けられる可能性が高いです。

JC-STAR 申請時点で、アットマークテクノに上記のような違反歴や影響を受ける懸念があるかどうかは、お手数をおかけしますがアットマークテクノの営業部までお問い合わせの上ご確認ください。

JC-STAR★1 開発ガイド

改訂履歴

バージョン	年月日	改訂内容
1.0.0	2025/08/28	・初版発行

JC-STAR★1 開発ガイド
Version 1.0.0
2025/08/28