RayAegis Japan

モバイルセキュリティを実現する！

Jamf Pro APIを活用したセキュリティ対応の『自動化』と
モバイルアプリのセキュリティ標準

株式会社レイ・イージス・ジャパン

マネージドセキュリティサービス部　廣井 辰哉

2025年8月8日

自己紹介

廣井 辰哉
Tatsuya Hiroi

＊＊＊ 出身 ＊＊＊

神奈川 西湘方面

ミカン農家生まれ

※当人は全く食べません

＊＊＊ 略歴 ＊＊＊

2010年　中小SES企業へ入社

非情報系学科からのIT業界就職

2023年　RayAegis Japanへ入社

入社の決め手：

マネージドセキュリティサービス部
SOCチーム所属、 現在に至る

RAY-SOC

RayAegis Japan

レイ・イージス・ジャパンについて

RayAegis Japan

会社名
株式会社レイ・イージス・ジャパン

住所
〒163-0532　東京都新宿区西新宿7-22-33 Polar西新宿4階

設立
2019年10月10日

資本金
9,800万円

従業員数（2025年7月現在）
39名

親会社
Ray Aegis Information Security（台湾）
設立：2011年11月　従業員：約350名

株式会社アリス
設立：2001年6月　従業員：約200名

レイ・イージス・ジャパンの事業内容

【主力サービス】

セキュリティ診断サービス

- AIクイック・ツール診断
- AIリモート脆弱性診断
  （Webアプリケーション診断）
- API診断
- モバイルアプリ診断
- プラットフォーム診断
- ファストペネトレーションテスト
- ペネトレーションテスト
- IoTペネトレーションテスト

＊1 **UTDS**：Undetected Threat Detection System
＊2 **EDR**：Endpoint Detection and Response
＊3 **SOC**：Security Operation Center
＊4 **ASM**：Attack Surface Management
＊5 **DWS**：Dark Web Search

RayAegis Japan

登壇に至ったきっかけ

会社の知名度もなければ、Jamf製品のユーザーでもない

なぜ？？

jamf | PROTECT for Mobile

×

RayAegis Japan

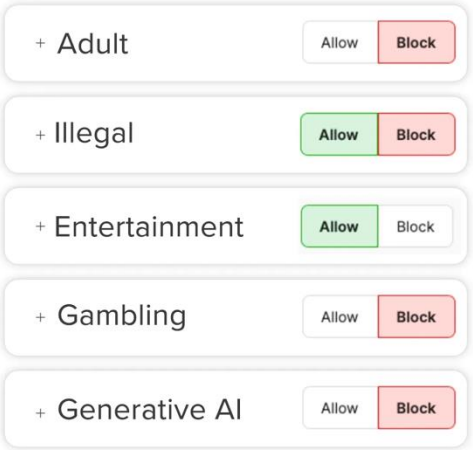
Jamf Protect for Mobileとは

jamf | PROTECT for Mobile

モバイル向けJamf Protect

Mobile Threat Defense（MTD：モバイル脅威防御）

コンテンツフィルタリング



脅威防御

- フィッシング
- ランサムウェア
- クリプトジャッキング
- マルウェアドメイン
- コマンド＆コントロール（C2）サーバトラフィック

etc

RayAegis Japan

iPhoneは安全って言うじゃない

セキュリティ・安全性は高いと言われている

しかし、**完璧ではない**

利用者の行動・リテラシー次第で**リスク**は高まる

モバイルデバイスを起因とした情報漏洩の事例

2025年7月　スミッシングによる不正アクセス事案

概要:

株式会社熊谷組にて、社員に貸与している社用iPhoneに紐づくApple Accountに対して
第三者による不正アクセスが発生し、1238件の個人情報が漏洩

原因:

スミッシング（ショートメッセージサービス：SMSを利用したフィッシング詐欺）
⇒運送会社を名乗るSMSを受信、配送の予定があったため疑うことなく送付されたリンクを開いてしまった

被害:

iCloudに同期されている電話帳（会社名、氏名、電話番号）

出典：株式会社熊谷組『個人情報漏えいの可能性について』(2025年7月4日)
※上記の掲載内容をもとに内容を平易化して再構成したものです、詳細は原文をご確認ください

あらためてモバイルセキュリティについて考える

モバイルデバイスはもはや**ビジネスシーンで欠かせない**

一方で「**iPhoneは安全**」という神話は**揺らぎつつある**

『**モバイルセキュリティ**』

あらためて考えてみる機会となれば幸いです

RayAegis Japan

モバイルデバイスのセキュリティ対策

とはいえ、何をすれば？

① MTD製品による脅威防御

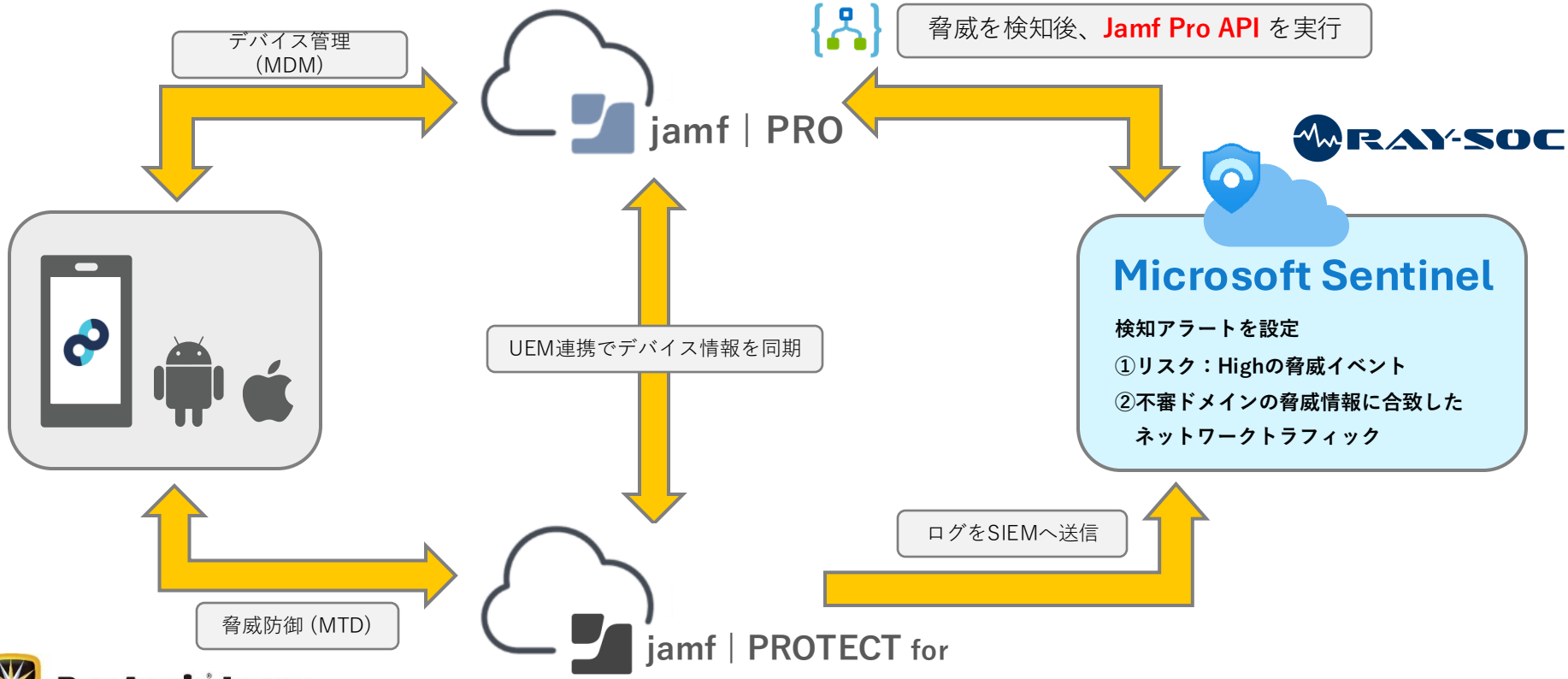
② 監視による早期検知・対応

セキュリティ対策の一例としてご紹介

③ APIによるセキュリティ対応

RayAegis Japan

Jamf x Ray-SOC 連携フロー



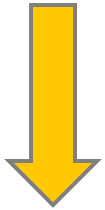
RayAegis Japan

なぜAPIを実行するのか？

脅威カテゴリ（一例）	ブロック可否
スパム	可
フィッシング	可
クリプトジャッキング	可
クレジットカード情報の漏洩	可
パスワードの漏洩	可
E メールアドレスの漏洩	可
マルウェア	可
中間者攻撃	不可
危険なホットスポット	不可
危険な証明書	不可
ジェイルブレイク（脱獄）	不可
危険な iOS プロファイル	可

脅威を検知できても対処が遅れることで

被害が拡大する恐れ



SIEM＊1での検知をトリガーにして

SOAR＊2でAPIを実行（一次対処を実施）

＊1 **SIEM**：Security Information and Event Management

＊2 **SOAR**：Security Orchestration, Automation and Response

RayAegis Japan

Jamf Pro API 使ってますか？

GUI上の管理コマンド（MDMコマンド）

RayAegis Japan

Jamf Pro API 使ってますか？

Jamf Pro API

デバイスロックコマンド実行例

（APIへリクエストを送信）

一から作るのは少し大変そう・・・

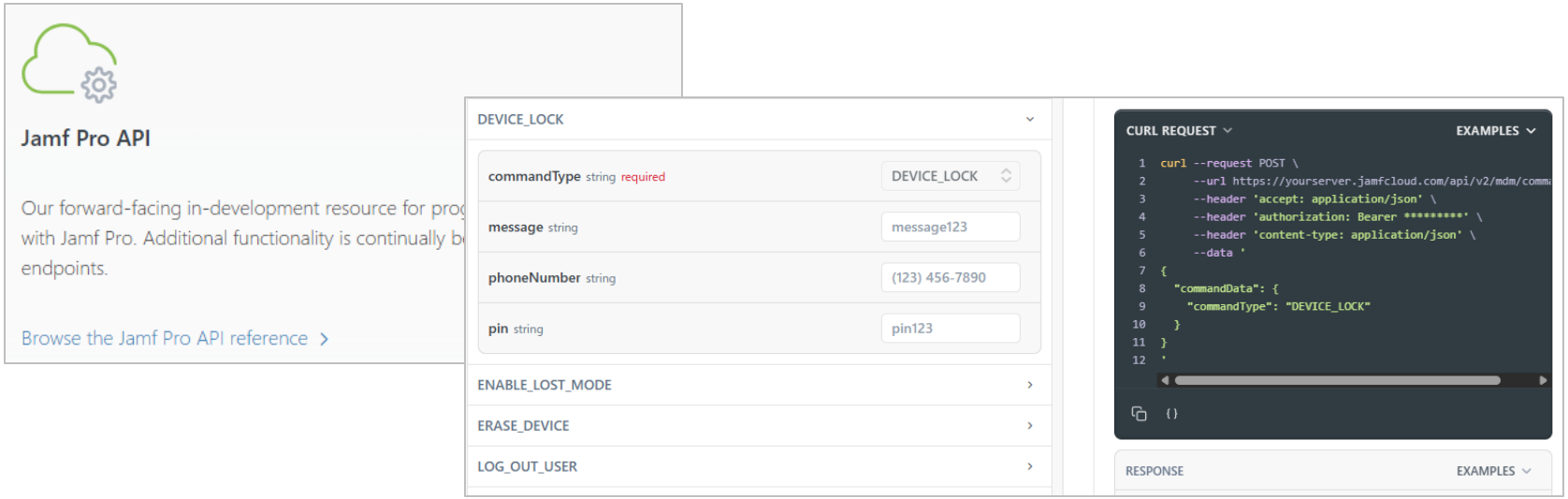
```
curl --request POST ¥
     --url https://yourserver.jamfcloud.com/api/v2/mdm/commands ¥
     --header 'accept: application/json' ¥
    --header 'authorization: Bearer *********' ¥
     --header 'content-type: application/json' ¥
     --data '
{
  "commandData": {
    "commandType": "DEVICE_LOCK",
    “message”: “脅威を検出したためデバイスをロックしました。ロックを解除せず、システム管理者へ連絡してください。"
  },
  "clientData": [
    {
      "managementId": "aaaaaaaa-3f1e-4b3a-a5b3-ca0cd7430937"
    }
  ]
}
'
```

RayAegis Japan

Jamf Pro API 使ってますか？

APIドキュメントでコードのサンプルを確認可能

https://developer.jamf.com/

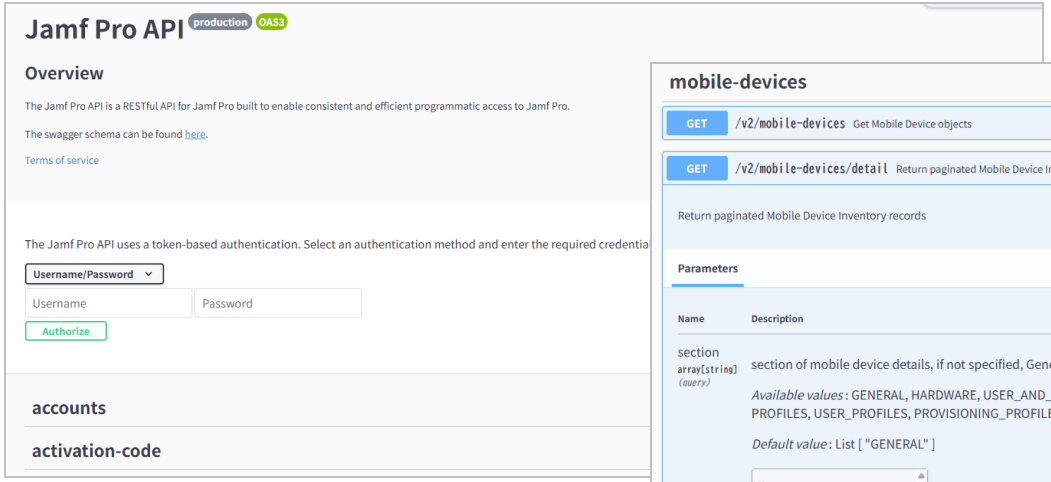


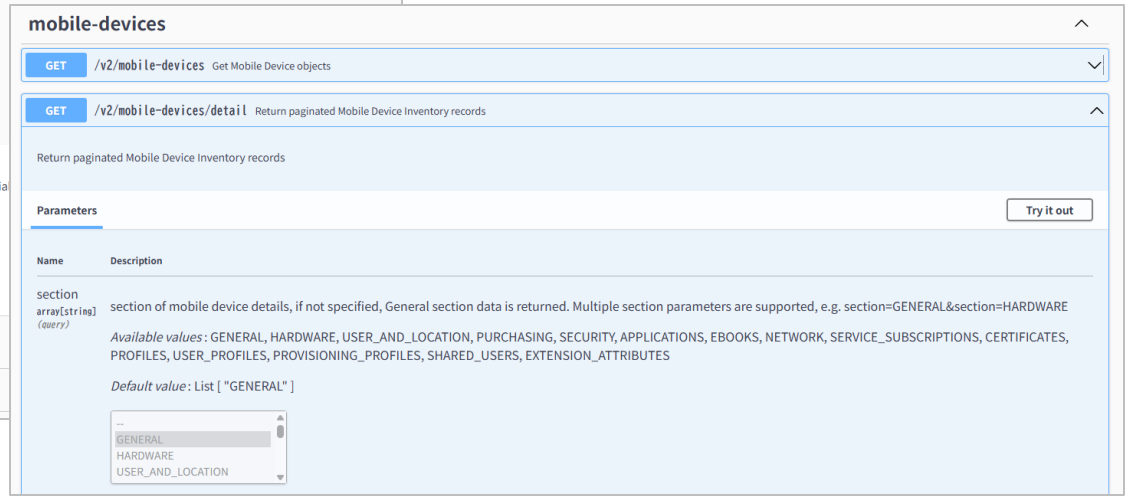
RayAegis Japan

Jamf Pro API 使ってますか？

Jamf ProのURLに「**api**」と入力することでAPIのテスト機能も利用可能

https://yourserver.jamfcloud.com/api/doc/





RayAegis Japan

Ray-SOCでのAPI活用例

アラート検知をトリガーに Logic Apps で以下のアクションを実行

1. 対象デバイスの詳細情報を取得
2. デバイスロックコマンドの実行

Ray-SOCでのAPI活用例

1. 対象デバイスの詳細情報を取得

Jamf Protectの検知ログに含まれる**UDID** ＊1 情報を使ってJamf Pro側のデバイス**管理ID**を取得

＊1 **UDID**：Unique Device Identifier
デバイスごとに一意に割り振られる固有の識別番号

① GET /mobile-devices/detail?section=GENERAL&filter=udid=="12345abcde67890fghij12345klmno67890pqrst"

② {"totalCount": 1, "results": [{ ・・・, "managementId": "aaaaaaaa-3f1e-4b3a-a5b3-ca0cd7430937", ・・・ }]}

Ray-SOCでのAPI活用例

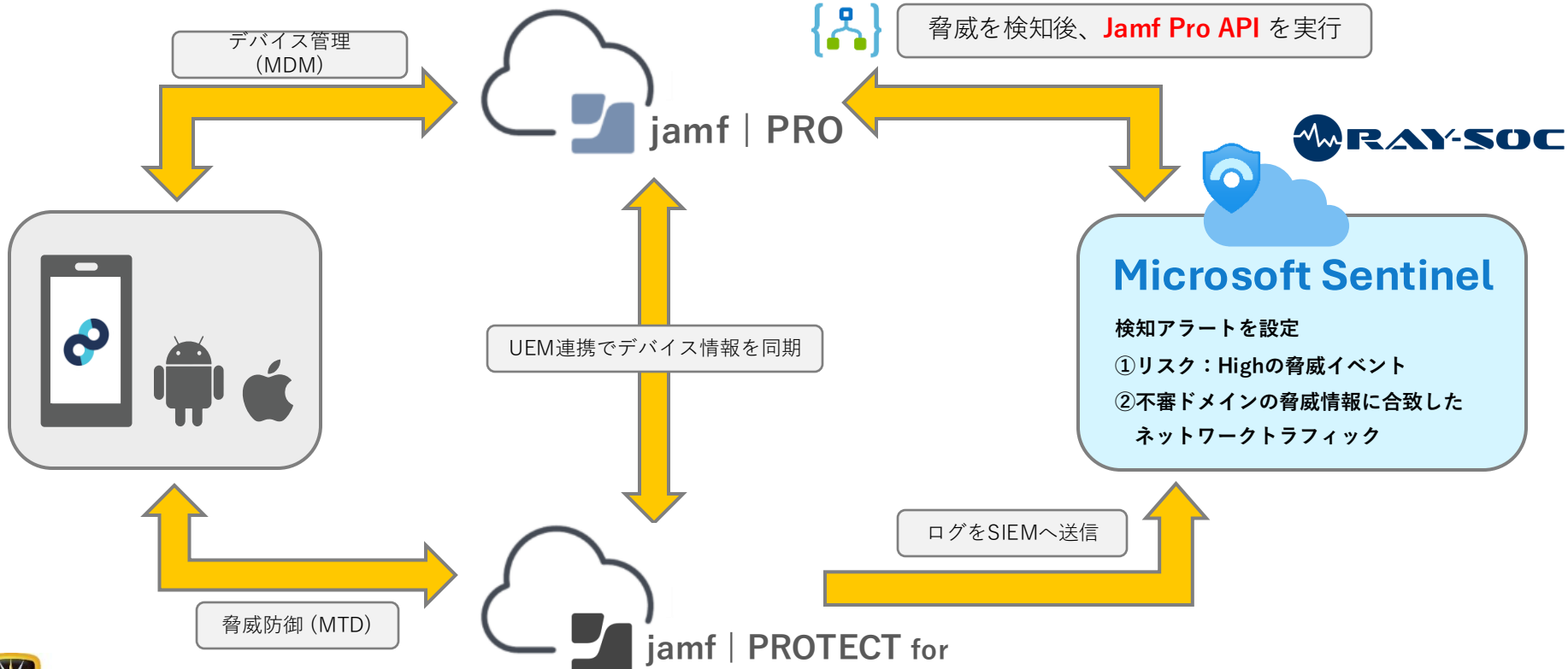
2. デバイスロックコマンドの実行

さきほど取得したデバイスの**管理ID**を使って対象デバイスをロック

① POST /mdm/commands

BODY {"commandData": {"commandType": "DEVICE_LOCK"},
"clientData": [{"managementId": "aaaaaaaa-3f1e-4b3a-a5b3-ca0cd7430937"}]}

Jamf x Ray-SOC 連携フロー



RayAegis Japan

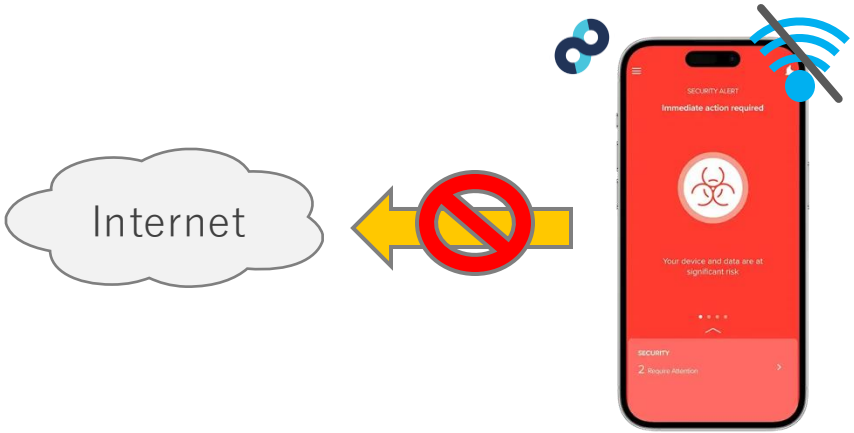
今後の展望

① デバイス隔離機能の実装

Jamf Pro **API** ＋ **拡張属性**

＋ Jamf Protect **コンテンツフィルタ**

＝ **<u>デバイス隔離を実現</u>**



② SOC対象範囲の拡大

モバイル向けJamf Protectに加え、

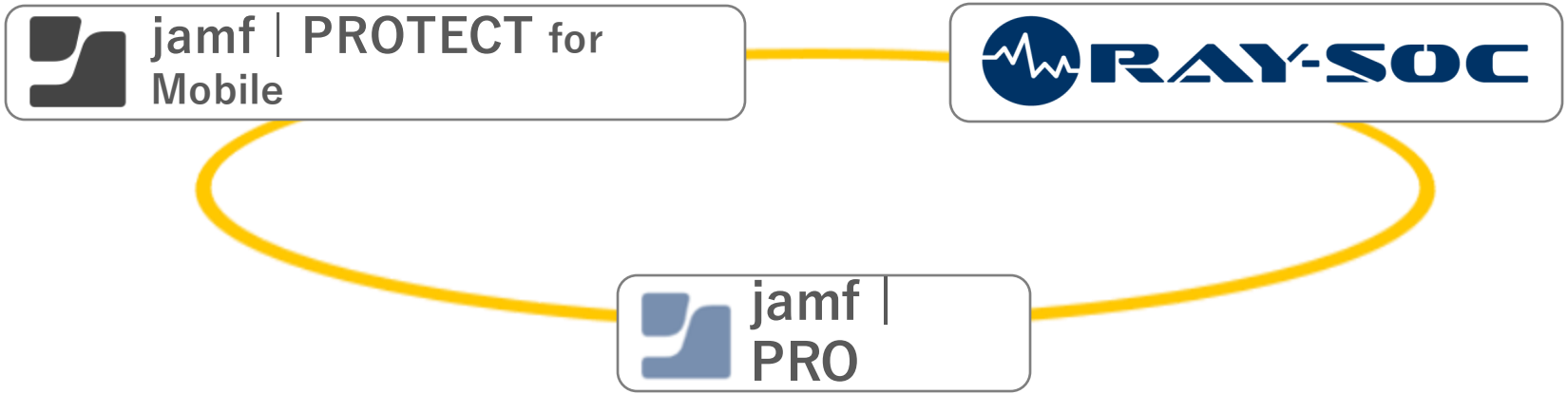
<u>Mac向けJamf Protect</u> もSOC対象へ

RayAegis Japan

モバイルデバイスのセキュリティ対策

① MTD製品による脅威防御

② 監視による早期検知・対応

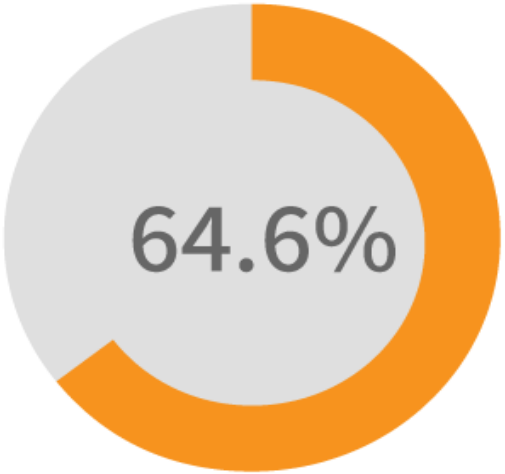


③ APIによるセキュリティ対応

MTD製品＋**監視運用**(＋**API**) でセキュリティ対策は万全・・・？

リスクはデバイスだけではない

Q. 何の数字だと思いますか？



「**対策が必要なレベル**」と診断されたスマホアプリの割合

※LAC社の「スマートフォンアプリケーション診断」で検出された脆弱性の統計データ

特に検出率の高い問題点 TOP5

1位：内部ストレージに重要情報を保存	39.1%
2位：中間者攻撃が可能	12.7%
3位：リクエスト改ざん	11.5%
4位：ログに重要情報を出力	11.5%
5位：認証機能における問題	10.4%

出典：一般社団法人 日本スマートフォンセキュリティ協会（JSSEC）『弱性診断結果をもとにわかったスマホアプリに潜む脆弱性の傾向』（2025年4月3日）

※上記の掲載内容をもとに内容を平易化して再構成したものです、詳細は原文をご確認ください

モバイルアプリを起因とした情報漏洩の事例

2025年1月　ハンズクラブアプリの不正アクセス事案

概要:

株式会社ハンズが運営する「ハンズクラブアプリ」が不正アクセスを受け、約12万件の会員情報が漏洩

原因:

アプリのシステムに使用されているソフトウェアの脆弱性を悪用

被害:

氏名、会員番号、メールアドレス、パスワード、住所、電話番号など
（※クレジットカード情報の漏洩は無し）

出典：株式会社ハンズ『「ハンズクラブアプリ」への不正アクセスによる個人情報漏洩に関する お詫びとお知らせ』（2025年1月27日）
※上記の掲載内容をもとに内容を平易化して再構成したものです、詳細は原文をご確認ください

モバイルアプリもセキュリティ対策を

Jamf Protect のようなMTD製品によるデバイス保護も重要

同時にモバイルアプリの開発者・提供者には

より「安全」なアプリの提供が期待されている

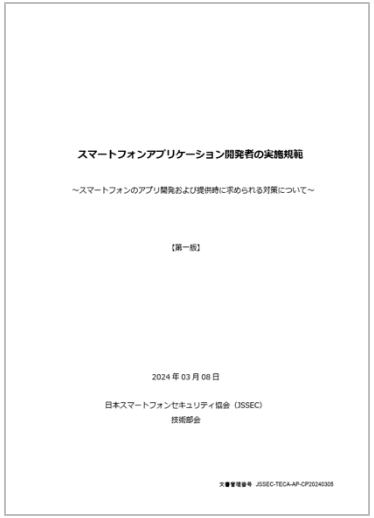
・・・とはいえ、何をすれば？

RayAegis Japan

スマートフォンアプリケーション開発者の実施規範

日本スマートフォンセキュリティ協会（JSSEC）が定めた、

アプリ開発者がアプリ提供に際して**利用者を保護するための実施規範**


スマートフォンアプリケーション開発者の実施規範

～スマートフォンのアプリ開発および提供時に求められる対策について～

【第一版】

2024 年 03 月 08 日

日本スマートフォンセキュリティ協会（JSSEC）
技術部会

文書管理番号 JSSEC-TECA-AP-CP20240305

スマホアプリ開発者向け実施規範
セキュリティとプライバシーの基本要件
アプリ公開後のメンテナンス
プライバシーの基本要件
利用規約の基本要件
ユーザサポート
セキュリティインシデント対応

出典：一般社団法人 日本スマートフォンセキュリティ協会（JSSEC）『スマートフォンアプリケーション開発者の実施規範（スマホアプリ開発者向け実施規範）』（2024年3月14日）より抜粋

「**安全なアプリ開発**」と「**継続的なリスク把握**」

出典：一般社団法人 日本スマートフォンセキュリティ協会（JSSEC）『スマートフォンアプリケーション開発者の実施規範 【第一版】』（2024年3月8日）より抜粋

RayAegis Japan

安全なアプリを開発するには

安全なアプリを開発するためには

セキュアなコーディングが重要

アプリ開発に際して推奨される、

セキュリティ要件が定められたガイドライン

MASVS

Mobile Application Security Verification Standard

モバイル アプリケーション セキュリティ検証標準

出典：一般社団法人 日本スマートフォンセキュリティ協会（JSSEC）『スマートフォンアプリケーション開発者の実施規範 【第一版】』（2024年3月8日）より抜粋

出典：OWASP『Mobile Application Security Verification Standard (MASVS)』(v2.1.0) より抜粋

MASVSとは

OWASP ＊1 が定めた

モバイルアプリ向けセキュリティ要件のワークフレーム

＊1 **OWASP**：Open Worldwide Application Security Project
ソフトウェアのセキュリティを向上させることを目的とした非営利団体
以前は Open “Web” Application Security Project という名称

モバイルの攻撃対象となりえる主要な領域を8つのグループに分類

MASVS-STORAGE	MASVS-PLATFORM
MASVS-CRYPTO	MASVS-CODE
MASVS-AUTH	MASVS-RESILIENCE
MASVS-NETWORK	MASVS-PRIVACY

出典：OWASP『Mobile Application Security Verification Standard (MASVS)』(v2.1.0) より抜粋

RayAegis Japan

MASVSとは

MASVS-STORAGE	デバイス上の機密データの安全な保管
MASVS-CRYPTO	機密データを保護するために使用される暗号化機能
MASVS-AUTH	モバイルアプリで使用される認証および承認メカニズム
MASVS-NETWORK	モバイルアプリとリモートエンドポイント間の通信におけるデータ保護
MASVS-PLATFORM	モバイルプラットフォームとインストール済みアプリとの安全なやり取り
MASVS-CODE	データ処理とアプリを最新の状態に保つためのベストプラクティス
MASVS-RESILIENCE	リバースエンジニアリングや改ざんの試みに対する耐性
MASVS-PRIVACY	ユーザーのプライバシーを保護するためのプライバシーコントロール

出典：OWASP『Mobile Application Security Verification Standard (MASVS)』(v2.1.0) より抜粋

MASVSとは

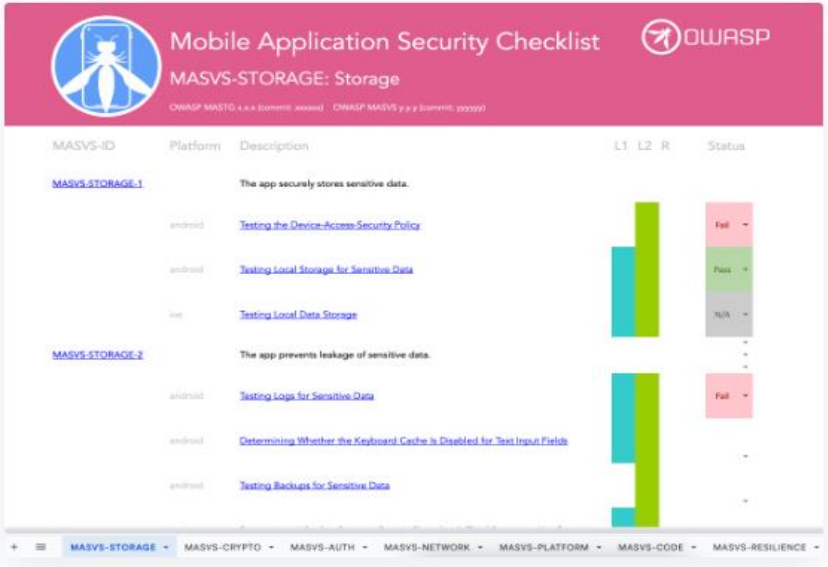
MASVS-STORAGE	• 端末内のストレージに平文でパスワードやトークンを置かない • 不要データは適切に削除し、キャッシュやログに残さない
MASVS-CRYPTO	• 標準的かつ十分な長さの暗号アルゴリズムを使用している • 鍵の生成、格納、破棄が安全に行われている
MASVS-AUTH	• パスワードやワンタイムトークンなどの認証情報が安全に管理されている • 認可ポリシーの実装ミス（権限昇格など）がない
MASVS-NETWORK	• すべての外部通信がTLSなどの暗号化プロトコルで保護されている • 中間者攻撃を防ぐために証明書ピンニングを行っている
MASVS-PLATFORM	• 機密情報が画面上に表示される際、キャプチャや録画をブロックする仕組みを適用している • WebViewにCSPを適用して許可済みスクリプト／リソースのみ実行している
MASVS-CODE	• 起動時にバージョン確認を行い、最新バージョンではない場合はアプリを停止している • 入力されたデータはすべて検証およびサニタイズを行っている
MASVS-RESILIENCE	• メソッド名、クラス名の難読化やメタデータ除去を行い、静的解析を困難化している • デバッガ検出などにより、解析ツールの実行を中断する動的解析防止を実装している
MASVS-PRIVACY	• アプリの機能に必須となるデータや権限についてユーザーから同意を取得している • 個人識別情報は匿名ID化やトークン化してユーザーの特定・追跡を防止している

出典：OWASP『Mobile Application Security Verification Standard (MASVS)』(v2.1.0) より抜粋

MASVSとは

各セキュリティ要件のチェックリストも有

参考にしてみてはいかがでしょうか？



出典：OWASP MAS Checklist (v2.0.0)より抜粋

出典：OWASP『Mobile Application Security Verification Standard (MASVS)』(v2.1.0)より抜粋

RayAegis Japan

リリース後も気を抜くことなかれ

アプリのリリース時点では存在しなかった脆弱性

しかし、**新たな脆弱性が日々発見されている**

脆弱性＝**リスク**の継続的な把握と対処も重要

特に注意すべきリスクをまとめたガイドライン

OWASP Mobile Top 10 2024

出典：OWASP Mobile Top 10 2024 より抜粋

RayAegis Japan

OWASP Mobile Top 10 2024とは

OWASPが定めた、モバイルアプリケーションにおける

セキュリティリスクの**上位10項目**を体系化したガイドライン

OWASP Mobile Top 10 2024	（日本語抄訳）
M1: Improper Credential Usage	不適切なクレデンシャルの使用
M2: Inadequate Supply Chain Security	不十分なサプライチェーンセキュリティ
M3: Insecure Authentication/Authorization	安全でない認証・認可
M4: Insufficient Input/Output Validation	不十分な入出力検証
M5: Insecure Communication	安全でない通信
M6: Inadequate Privacy Controls	不十分なプライバシーコントロール
M7: Insufficient Binary Protections	不十分なバイナリ保護
M8: Security Misconfiguration	セキュリティの設定ミス
M9: Insecure Data Storage	安全でないデータ保存
M10: Insufficient Cryptography	不十分な暗号化

出典：OWASP Mobile Top 10 2024 より抜粋

RayAegis Japan

OWASP Mobile Top 10 2024とは

OWASP Mobile Top 10 2024	脆弱な設定の例
M1: Improper Credential Usage 不適切なクレデンシャルの使用	認証情報がソースコードにハードコードされている
M2: Inadequate Supply Chain Security 不十分なサプライチェーンセキュリティ	外部ライブラリやSDK導入前に検証をしていない
M3: Insecure Authentication/Authorization 安全でない認証・認可	多要素認証（MFA）を実装していない
M4: Insufficient Input/Output Validation 不十分な入出力検証	入力データやサーバー応答をサニタイズしていない
M5: Insecure Communication 安全でない通信	証明書ピンニングを実装していない
M6: Inadequate Privacy Controls 不十分なプライバシーコントロール	不要な個人識別情報（PII）を取得している
M7: Insufficient Binary Protections 不十分なバイナリ保護	コードの難読化や改ざん検知機能を組み込んでいない
M8: Security Misconfiguration セキュリティの設定ミス	ログやデバッグ情報に機密データが含まれている
M9: Insecure Data Storage 安全でないデータ保存	ローカルストレージへ機密データを暗号化せずに保存している
M10: Insufficient Cryptography 不十分な暗号化	暗号鍵がアプリケーション内にハードコードされている

出典：OWASP Mobile Top 10 2024 より抜粋

RayAegis Japan

ご参考：当社モバイルアプリ診断項目の一例

診断項目	診断内容
権限とマニフェスト分析	攻撃者が特権昇格したり、アプリケーション内部のデータを変更したりすることを可能にする権限の問題があるかどうかをスキャンします。また、アプリケーションが過剰に不合理な権限を要求しているかどうかもスキャンします。 さらに、他の悪意のあるアプリケーションがテスト対象のアプリケーションにフックして感染し、データ漏洩やアプリケーションの侵害を引き起こす可能性があるかどうかもスキャンします。
バイナリとコードの分析	コードをデコンパイルし、アプリケーション内部の脆弱性をスキャンします。 SQLインジェクション、ハードコードされたパスワード、ストレージの安全でない使用方法、ロジックフロー、JailBreakやルート検出などの問題を探索し、アプリケーションの侵害を引き起こす可能性のある多数のベクトルを検査します。
ネットワーク通信診断	アプリケーション内部で通信されるドメイン名やIPアドレスを探します。 ドメイン名やIPアドレスの中には、侵害されたり、誤って指定されたり、悪者によって注入されたりするものがあります。当社のスキャナは、各通信先を自動的にチェックします。既知の問題と一致するマルウェアとC&Cサーバの識別も実施されます。
証明書の警告	証明書が有効かの確認のほか、隠し証明書がないか、また、中間者攻撃防止のために証明書のピン止めがされているかをスキャンします。 多くのアプリケーションでは証明書のチェックを適切に実施できていないことで中間者攻撃につながり、データ漏洩の原因となります。
ファイル解析	アプリケーションの実行中に、何らかの動作を達成するため、ファイルが読み込まれたり、書き込まれたり、実行されることがあります。 当社のスキャナは、攻撃者がファイルを利用してアプリケーションのセキュリティ問題を引き起こす可能性があるかどうかをチェックします。例えば、機密情報がファイル内に保存されているかどうか、攻撃者がファイルを変更してロジックフローを変更する可能性があるかどうかなどをチェックします。
コンポーネントの列挙	アプリケーションで使用されているコンポーネントをチェックします。 コンポーネントの中で、脆弱性が含まれていたり、古くなっているものを指摘します。

ご参考：当社モバイルアプリ診断項目の一例

診断項目	OWASP Mobile Top 10 2024 該当するリスク項目
権限とマニフェスト分析	M1: Improper Credential Usage、M3: Insecure Authentication/Authorization、M5: Insecure Communication、M7: Insufficient Binary Protections、M8: Security Misconfiguration
バイナリとコードの分析	M1: Improper Credential Usage、M2: Inadequate Supply Chain Security、M4: Insufficient Input/Output Validation、M6: Inadequate Privacy Controls、M7: Insufficient Binary Protections、M8: Security Misconfiguration、M9: Insecure Data Storage、M10: Insufficient Cryptography
ネットワーク通信診断	M2: Inadequate Supply Chain Security、M5: Insecure Communication、M10: Insufficient Cryptography
証明書の警告	M5: Insecure Communication、M10: Insufficient Cryptography
ファイル解析	M1: Improper Credential Usage、M2: Inadequate Supply Chain Security、M6: Inadequate Privacy Controls、M7: Insufficient Binary Protections、M8: Security Misconfiguration、M9: Insecure Data Storage、M10: Insufficient Cryptography
コンポーネントの列挙	M2: Inadequate Supply Chain Security

RayAegis Japan

モバイルアプリもセキュリティ対策を

「安全なアプリ開発」と「継続的なリスク把握」

① 安全なアプリ開発

- セキュアコーディング／セキュア開発
- リリース**前**の各種診断

（モバイルアプリ、Webアプリ、APIの脆弱性診断）

② 継続的なリスク把握

- リリース**後**の「**定期的な**」脆弱性診断
- アプリの不正アクセス／不正利用監視

セキュリティ標準に則ったアプリ開発と維持**でセキュリティ対策を！**

RayAegis Japan

総括

MTD製品＋監視運用による

モバイルデバイスの保護

×

セキュリティ標準に則った

モバイルアプリ開発と維持

モバイルセキュリティを実現！

RayAegis Japan

RayAegis Japan

より安全な環境の実現に、
より現実的なセキュリティ対策を

株式会社レイ・イージス・ジャパン

〒160-0023 東京都新宿区西新宿7-22-33 Polar西新宿 4階

https://www.rayaegis.co.jp/