Amazon MWS の
デスクトップアプリケーション

お知らせ

Amazon の出品者と開発者には、このドキュメントの情報を個別に評価する責任があります。この文書は、(a)情報提供のみを目的として、(b)現在の慣行を示しており、これらは予告なく変更されることがあります。また、(c)Amazon.com Services LLC(Amazon)およびその関連会社、サプライヤー、ライセンサーのコミットメントまたは保証を発生させるものではありません。Amazon マーケットプレイス Web サービス(Amazon MWS)の商品またはサービスは、明示または黙示を問わず、いかなる種類の保証、表明、条件もなく、「現状のまま」提供されます。Amazon MWS に関する Amazon の責任は、Amazon の MWS 契約(Amazon セリングパートナー API 開発者契約、Amazon 出品パートナーAPI ライセンス契約など)によって管理されており、この文書は Amazon と第三者間の契約の一部ではなく、それを修正するものでもありません。

目次

概要

このドキュメントでは、Amazon マーケットプレイス Web サービス（Amazon MWS）を使用して構築されたデスクトップベースのアプリケーションの要件について説明し、データセキュリティの許容可能な認証モデルとベストプラクティスに焦点を当てています。このドキュメントでは、Amazon MWS 適正使用ポリシー（AUP）およびデータ保護ポリシー（DPP）に準拠するためのガイダンスを提供しています。

デスクトップアプリケーションとは、単一のコンピュータにインストールされるソフトウェアとして定義されています。対照的に、**Web アプリケーション**はクライアント/サーバーアーキテクチャで構築され、Web ブラウザをクライアントインターフェイスとして使用します。デスクトップアプリケーションには 2 つのタイプがあります。

- **ネイティブのデスクトップアプリケーション**は、ユーザーが自分のコンピュータにダウンロードし、アプリケーション開発者のシステムには接続されません。
- **ハイブリッドアプリケーション**は、開発者が所有するリモートホストで実行されます。ハイブリッドアプリケーションは、Web ブラウザインターフェイスとユーザーのコンピュータにダウンロードされたデスクトップコンポーネントの両方を使用します。

Amazon MWSアプリケーションを構築するための要件

Amazon MWS でデスクトップベースのアプリケーションを構築するための要件を以下に示します。

a. 適切な**認証モデル**を使用してアプリケーションを開発します。このモデルでは、Amazonの出品者に代わってあなたがAmazon MWS フルフィルメント APIの呼び出しを行えるように出品者が認証します。Amazon MWS フルフィルメント APIの承認には、出品者が提供したAmazon MWS認証トークンに加えて、開発者IDとアクセスキーを使用します。API呼び出しは、出品者のデバイスではなく、あなたのサーバーから発信する必要があります。

b. 必要な**データ暗号化**制御を使用してアプリケーションを開発します。Amazonの情報がすべてのネットワークチャネルを通じて暗号化され、アプリケーションに保存されていることを確認してください。Amazonカスタマー PIIデータは、常に暗号化する必要があります。アプリケーション内のすべての通信チャネルは、ファイアウォールまたはセキュリティグループを使用して保護する必要があります。

c. 安全な**ログ記録とモニタリング**ソリューションを使用してアプリケーションを開発します。ログがアプリケーションで安全にキャプチャされ、保護されたネットワーク経由でシステムに返されます。アプリケーションのあらゆる側面を対象とする自動アラームと監視を実装します。

d. 簡単にアクセスできる**ソフトウェアアップデートとセキュリティパッチ**を使用して、アプリケーションを開発します。Webアプリケーションおよびデスクトップアプリケーションに、最新のソフトウェアアップデートとセキュリティパッチが適用されていることを必ず確認してください。また、ウイルス、マルウェア、ハッキングから保護するアプリケーションを設計し、セキュリティ侵害の可能性を低減することが大切です。

次のセクションでは、デスクトップアプリケーションを構築する際の、これらの各要件についての特別な考慮事項について説明します。

認証モデル

グラント認証モデル

グラントモデルは、Amazon MWS フルフィルメント API について認証を行う、唯一の許容できる方法です。このモデルでは、出品者が MWS 認証トークンを提供するため、開発者は出品者に代わって Amazon MWS を呼び出すことができます。Amazon MWS への API 呼び出しは、出品者のコンピュータではなく、あなたのシステムから呼び出す必要があります。このため、純粋なネイティブのデスクトップアプリケーションモデルは許容されません。Amazon MWS フルフィルメント API 呼び出しが、あなたが管理するホストから発信されるように、アプリケーションをビルドする必要があります。このホストは、ご利用の環境内でリモートサーバーとして機能します。API 呼び出しに対するレスポンスを受け取るため、Amazon の情報を保護するのに必要なネットワーク制御と、セキュリティ制御が必要です。グラントモデルを使用すると、DPP 規制により Amazon の情報を管理できます。出品者があなたのお客様である場合に限り、出品者の認証トークンを後で使用できるようデータベースにセキュアな方法で保存することができます。

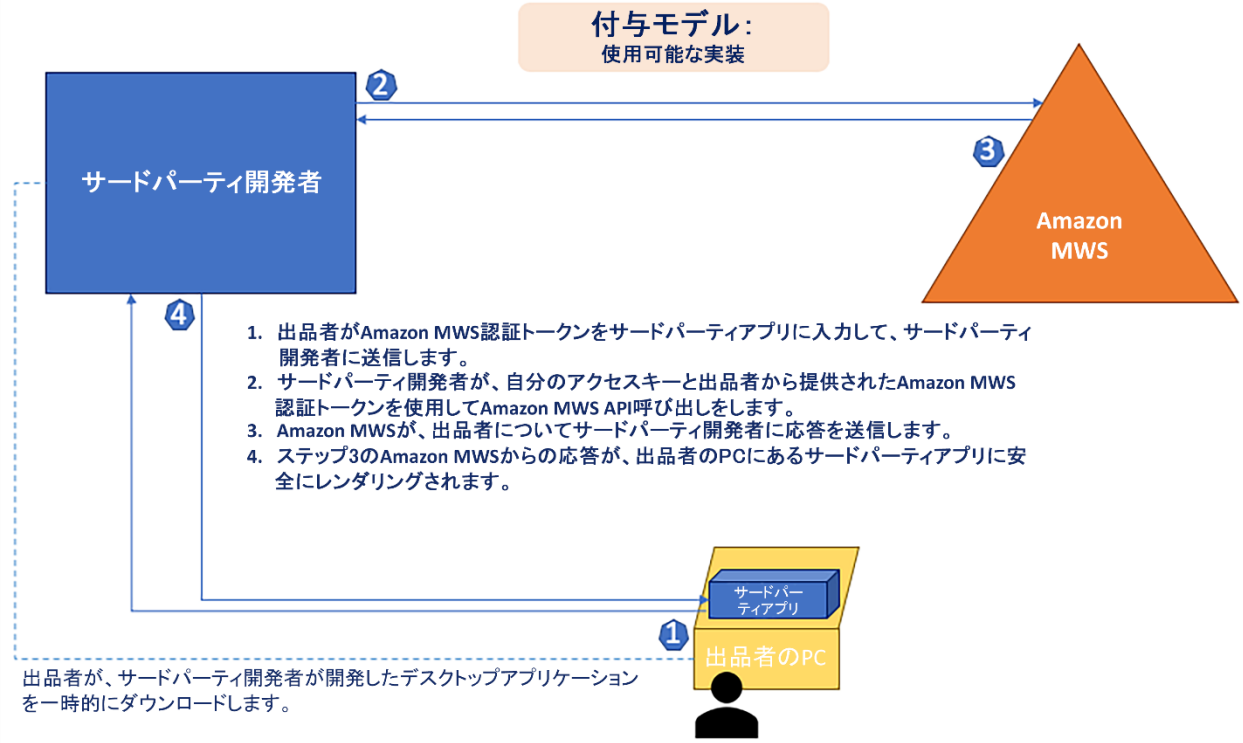

図1: グラント認証モデルの実装

許容できない認証モデル

認証情報を共有する認証モデルは、利用規約に違反し、Amazon では許可されていません。これは、出品者が認証情報を共有する場合や、開発者が認証情報を共有する場合に生じます。

出品者が認証情報を共有している

出品者が認証情報を共有するための Amazon ポリシーに違反しています。図 2 では、出品者は自分の Amazon MWS アクセスキーと秘密キーをアプリケーションのインターフェイスに入力し、Amazon MWS フルフィルメント API 呼び出しを行います。これは、「*いかなる目的のためにも、出品者の秘密キーを求めない、または受け入れてはいけない。*」とする AUP 条項に違反します。アクセスキーと秘密キーにアクセスできない場合でも、Amazon カスタマーの PII データはセキュリティ上のリスクにさらされます。出品者は、アプリケーションのすべてのチャネルでデータを保護し、暗号化するための知識やツールを持っていない場合があります。

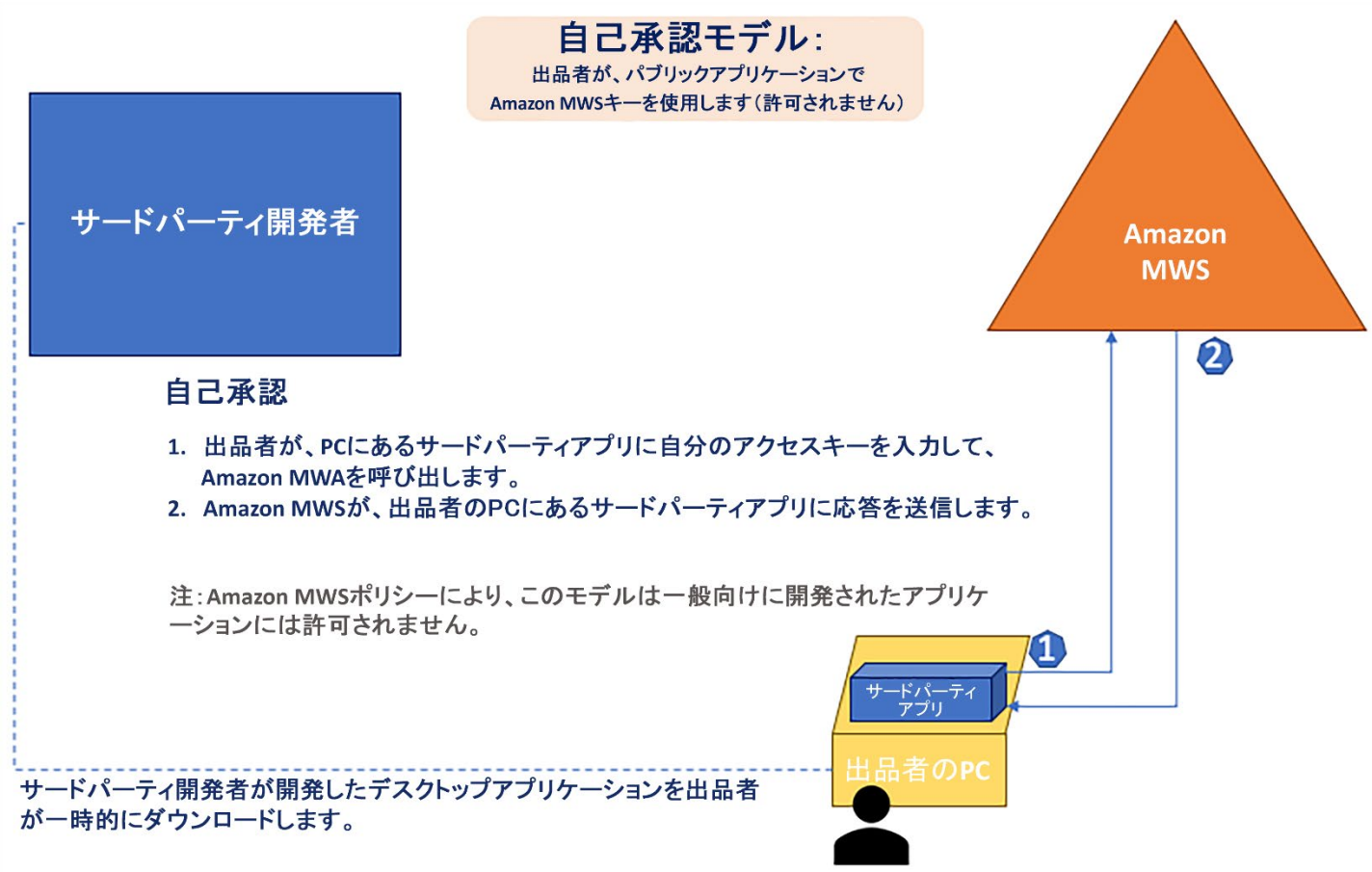

図2: 自己認証 — 許容できない

開発者が認証情報を共有する

開発者が認証情報を共有することは、Amazon ポリシーに違反しています。図 3 では、開発者 ID とアクセスキーがハードコードされているか、アプリケーションのデスクトップコンポーネントに埋め込まれているか、GitHub などのパブリックリポジトリで共有されています。これにより、Amazon MWS アクセスキーは、出品者のデバイスまたはパブリックリポジトリで公開されます。これは、「*キーまたはパスワードを共有してはいけない*」という AUP の条項に違反します。

このモデルには、次の 2 つのリスクがあります。

a. 開発者のアクセスキーと秘密キーは、アプリケーションまたは公開リポジトリの複数のインスタンス間で共有されます。そのためすべての出品者のAmazon情報へのアクセスを許してしまい、違反となる危険性があります。

b. Amazon情報は、出品者のコンピュータのデスクトップアプリケーションに保存されます。これは、Amazonの情報を管理し、保護する機能がないことを意味します。

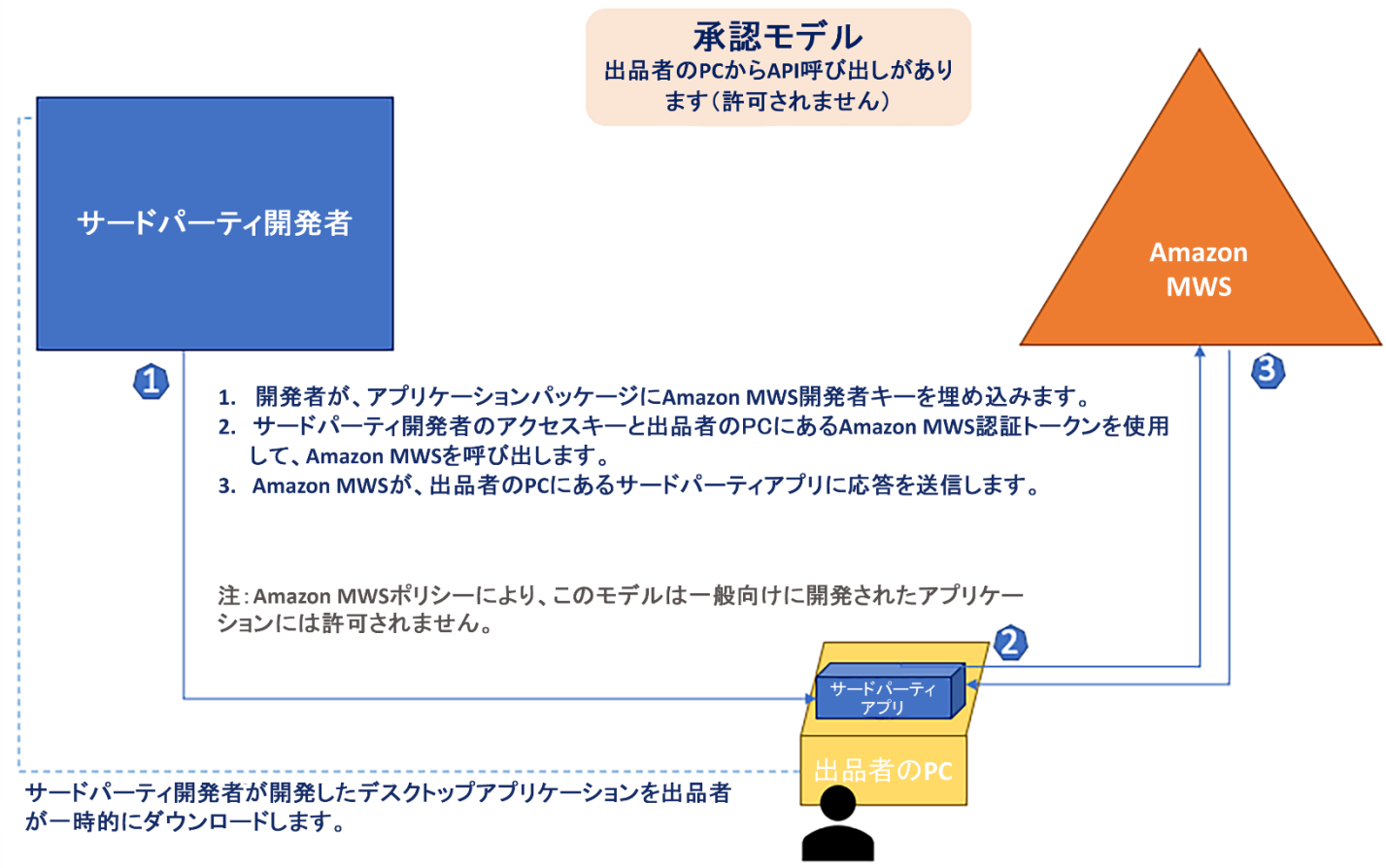

図 3: 開発者のMWSアクセスキーと秘密キーを公開する認証モデル — 許容できない

データの暗号化と保存

すべての Amazon MWS 開発者は、アプリケーションの種類にかかわらず、データ保護ポリシーを遵守する必要があります。Amazon では、アプリケーションに関する Amazon 情報の保護を開発者の責任であるとみなしています。アプリケーションは、以下に準拠している必要があります。

1. 転送中のデータの暗号化
2. 保管中のデータの暗号化
3. 自動化されたデータ保存とライフサイクルのポリシー

転送中のデータの暗号化

転送中のデータとは、あるシステムから別のシステムに送信される任意のデータを指します。これには、環境内のリソース間の通信、および出品者のデバイスにインストールされたネイティブのデスクトップコンポーネント間の通信が含まれます。転送中のデータに対して適切なレベルの保護を提供することで、Amazon 情報の機密性と完全性を維持することができます。通常、TLS（Transport Layer Security）、SFTP（SSH ファイル転送プロトコル）などのセキュアなプロトコルを選択することが、最も効果的な方法となります。たとえば AWS サービスでは、通信に TLS を使用する HTTPS のエンドポイントが提供されるため、AWS API との通信時には、データ転送中に暗号化が行われます。また、サーバー側の暗号化プロトコルとクライアント側の暗号化プロトコルを使用すると、SSL（Secure Sockets Layer）を使用したり、クライアント側のデータを暗号化したり、暗号化データをデータストアにアップロードしたりして、データを保護することができます。

保管中のデータの暗号化

保管中のデータとは、任意の期間にわたって保持するデータのことを指します。これには、ブロックストレージ、オブジェクトストレージ、データベース、およびデータが保持されるその他のストレージメディアが含まれます。暗号化とアクセス制御で保管中のデータを保護することで、不正アクセスのリスクを軽減できます。デスクトップアプリケーションにより、出品者のデバイスで Amazon PII データをローカルにダウンロードできる場合、これらのダウンロードの暗号化と、DPP で定義されているアクセスコントロールが必要です。データを複数のチャネルを介して永続化する場合は、データのメッセージレベルの暗号化を考慮する必要があります。（複数のチャネルを介してデータを永続化するアプリケーション）たとえば、データベース、ウェブサーバー、および出品者のコンピューターのデスクトップアプリでは、属性レベルでデータを暗号化することを検討する必要があります。属性レベルでデータを暗号化すると、データを暗号化するためのデータストアへの依存がなくなります。

自動化されたデータ保存とライフサイクルのポリシー

アプリケーションは、データ保持要件を満たしている必要があります。「適正利用規約」には、*「開発者は注文を履行するために必要な期間（注文出荷後 30 日間以内）、PII を保持します。このアーカイブされたストレージは、「コールド」または「オフライン」である必要があります。」*という記述があります。Amazon では、出品者の端末に Amazon PII データを保存しないことを推奨しています。デスクトップアプリケーションで Amazon PII データの永続性が許可されている場合は、保持期間の終了時に自動データ削除メカニズムを実装する必要があります。保存期間の満了後にデータを確実に削除するために、デスクトップアプリケーションの検証メカニズムを開発する必要があります。たとえば Amazon S3 では、オブジェクトバージョンの「保持期間」の日数を設定し、保持期間が終了するまでオブジェクトバージョンを保護できます。

ログの記録とモニタリング

アプリケーションのログ記録

アプリケーション、サービス、リソースログなど、アプリケーションスタック全体のログを設定します。すべてのログを一元的に収集し、自動的に分析して、悪意のあるアクティビティや侵害による異常や指標を検出する必要があります。デスクトップアプリケーションの場合、エージェントベースのツールを使用してログを収集できます。たとえば、CloudWatch エージェントまたは API を使用し、ハイブリッドアーキテクチャで CloudWatch を使用して、リソースからログファイルを集約し、デスクトップアプリケーションをモニタリングできます。Amazon PII データのアクセスや変更など、重要なコンテンツを含む詳細なログ記録を実装します。環境への変更を自動的にログ記録すると、信頼性に影響を与えている可能性のあるアクションを監査し、迅速に特定できるようになります。問題が発生したときにログを取得し、ログファイルを自分の環境に返送するよう出品者に依頼する「ファイルへのログ記録」メカニズムを実装します。オペレーティングシステムのイベントログに直接安全にログ記録することは、デスクトップアプリケーションのログを高速かつ信頼性の高いものにするうえで大変重要です。例外またはクラッシュを認識し、例外ログファイルの詳細を、指定されたサーバーに安全に送信する自動メカニズムを実装します。アラートは、できるだけシステムから切り離す必要があります。

モニタリングとアラーム

可用性の要件が満たされていることを確認するうえで、モニタリングは不可欠です。リモートのロケーションからすべての外部エンドポイントをモニタリングし、ベースの実装から独立していることを確認します。デスクトップアプリケーションのすべてのインスタンスをモニタリングします。「ユーザーカナリア」アプリケーションの使用に関する問題を検出するまでの時間が改善されました。これらは、デスクトップアプリケーションと Web アプリケーションの両方で実装できます。カナリアデプロイは、少数のお客様を新しいバージョンに誘導し、発生した動作の変更やエラーを精査することを目的としています。重大な問題が発生した場合は、カナリアデプロイか

らトラフィックを削除し、出品者を古いバージョンに誘導できます。アプリケーションの過負荷を避けるため、これらの導入作業は極めて短い時間で慎重に選択して完了する必要があります。導入作業を短期間で行うことで、テストを短期間実施できます。これにより、ユーザーが気づく前に問題を検出できるようになります。

ソフトウェアアップデートとセキュリティパッチ

ソフトウェアを最新の状態に保つことは大切です。ソフトウェア更新プログラムは、セキュリティ上の問題やアプリケーションで検出されたマイナーなバグに対処し、ハードウェアや周辺機器の動作を改善し、コンピューターの新しいモデルへのサポートを強化します。オペレーティングシステムの更新プログラムの多くは、セキュリティ更新プログラムです。これは、ハッカーやウイルスによって悪用される可能性のある脆弱性からコンピュータを保護する目的で発行されます。常に変化する脅威からシステムを可能な限り保護するために、セキュリティ更新プログラムがリリースされたら必ずインストールすることが大切です。デスクトップアプリケーションがインターネットに常時接続されておらず、強制メカニズムがないために必須の更新が行われず、アプリケーションが古い状態に留まっていることがあります。Web アプリケーションがこの問題に直面することはありませんが、デスクトップアプリケーションはダウンロードされるため、古いバージョンがインストールされたまま放置される場合があります。そのため、アプリケーションが実行中のホストの保有者であれば、パッチを強制的に適用することができます。出品者のコンピュータで実行されるデスクトップアプリケーションの場合、各インスタンスへのソフトウェアアップデートはケースバイケースで行います。デスクトップアプリケーションでは、ソフトウェアが最新バージョンであることを確認するための検証チェックが行われます。たとえば、AWS Patch Manager がオペレーティングシステムにインストールされている場合、セキュリティ関連パッチはデフォルトでサーバーで更新されます。

結論

このデスクトップアプリケーションのホワイトペーパーでは、データセキュリティと許容できる認証モデルに関するアーキテクチャ上のベストプラクティスについて説明しました。そして、デスクトップベースのアプリケーションを開発するための正しい方法について説明しました。認証および承認の強力な制御機能、セキュリティイベントに対する自動応答、複数レベルでのインフラストラクチャ保護、適切なデータ管理と暗号化により、あらゆるアプリケーションで必要な保護が実現します。

その他のリソース

- MWS利用規約
- MWSデータ保護ポリシー
- AWS DFS
- AWSの優れたアーキテクチャフレームワーク
- AWSセキュリティの中核

ドキュメントの変更履歴

日付	説明
2019 年 12 月	初版