LaunchDarkly

LaunchDarkly セキュリティ概要

組織で安全に利用するために
知っておくべきことについて

LaunchDarkly セキュリティ概要

LaunchDarkly は、業界をリードする SaaS 製品の機能管理です。私たちは、ソフトウェア チームがリスクを軽減してコードをデプロイできるようにすることで、より迅速なイノベーションを可能にします。LaunchDarkly は、幅広い業界や地域のお客様の厳格なセキュリティおよびプライバシー要件をサポートします。

このホワイト ペーパーでは、LaunchDarkly を組織に安全に統合するために知っておくべきことを概説します。内容は次のとおりです。

- ✓ セキュリティに関するアーキテクチャ上の考慮事項
- ✓ LaunchDarkly に送信されるユーザー データの最小化
- ✓ 安全な運用のためのベスト プラクティス

セキュリティに関するアーキテクチャ上の考慮事項

LaunchDarkly がアプリケーションと統合する仕組みについて

LaunchDarkly を使用するには、実行時に機能評価を有効にするために、アプリケーションに LaunchDarkly ソフトウェア開発キット（SDK）を埋め込む必要があります。開発者は、機能のコードパスを LaunchDarkly SDK の呼び出しでラップして、機能フラグを評価します。その後、機能フラグの状態を管理し、現在のユーザー、デバイスの詳細、またはビジネスに必要なその他の属性など、コンテキスト データに基づいて状態のルール セットを作成できます。フラグの状態とルール セットは、LaunchDarkly UI または REST API を介して変更できます。結果の状態は、機能のオン/オフなど、アプリケーションの動作を変更するために使用されます。

UI または API でフラグが更新されると、その特定の変更に関する新しいデータが、一方向のサーバー送信接続を使用して、メモリ内ストレージのために各 SDK にストリーミングされます。これらの更新は 200 ミリ秒以内にすべての SDK に伝播され、エンドユーザーは常に最新の状態を確実に体験できます。

クライアント側 SDK とサーバー側 SDK

LaunchDarkly には、クライアント側とサーバー側の 2 つの異なる SDK アーキテクチャを使用できます。クライアント側 SDK は、サーバー側 SDK とは異なるセキュリティ プロパティを備えています。

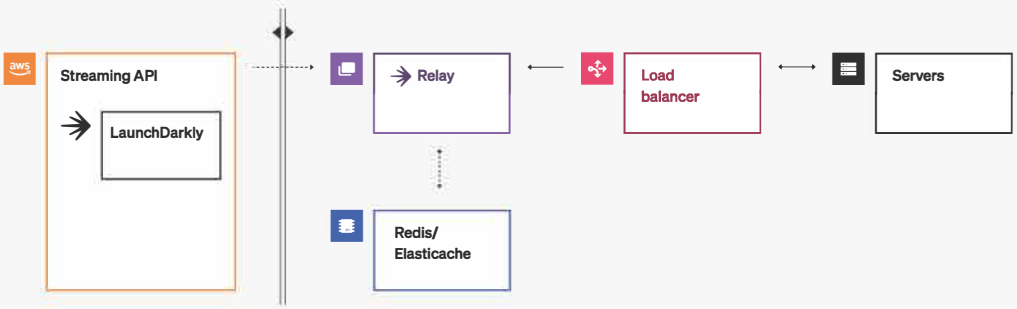
- 初期化中に、LaunchDarkly サーバー側 SDK はすべての機能フラグとターゲティング ルールを LaunchDarkly からフェッチし、それらをメモリにローカルに保存します。つまり、アプリケーションは LaunchDarkly に外部ポーリング コールを返すことなく、すぐに機能を有効にできます。セキュリティ上の理由から、クライアント側 SDK はルールセット全体をダウンロードして保存することはできません。クライアント側 SDK は通常エンドユーザーのデバイス上で実行されるため、技術的には送信されたすべてのデータを表示および変更できます。潜在的に機密性の高いデータを保存する代わりに、クライアント側 SDK は、ストリーミング接続または REST API 要求を通じて LaunchDarkly サーバーと通信することで、フラグルールを確認および更新します。
- デフォルトでは、クライアント側 SDK は認証されません。このため、あるユーザーが別のユーザーのアカウントを使用して、自分のものではないフラグを評価する可能性があります。ユーザー データを認証するには、SDK のセキュア モードを有効にすることができます。これには、ユーザー データとともにサーバー生成のハッシュを渡す必要があります。詳細については、セキュア モードのドキュメントを参照してください。（secure mode ドキュメント）
- クライアント側 SDK は、GET クエリ パラメータとして URL 内のコンテキスト データを送信します。そのデータがログまたは中間プロキシによって保存されることを懸念する場合は、useReport 設定を使用して HTTP REPORT 動詞を使用できます。これにより、コンテキスト データがヘッダーではなくリクエスト本文で送信されます。（useReport ドキュメント）
- クライアント側 SDK は、GET クエリ パラメータとして URL 内のコンテキスト データを送信します。そのデータがログまたは中間プロキシによって保存されることを懸念する場合は、useReport 設定を使用して HTTP REPORT 動詞を使用できます。これにより、コンテキスト データがヘッダーではなくリクエスト本文で送信されます。
- 「このフラグをクライアント側 SDK で使用できるようにする」設定を使用して、クライアント側 SDK がアクセスできるようにする各フラグを有効にするか、プロジェクトの構成画面内でフラグのデフォルト設定を構成する必要があります。
- 詳細については、クライアント側 SDK とサーバー側 SDK のドキュメントを参照してください。

> **Client-side SDKs 及び Server−Side SDKs ドキュメントについて**

Relay Proxy

Relay Proxy は、LaunchDarkly への直接接続の数を最小限に抑えるのに役立ちます。サーバーは LaunchDarkly のストリーミング API に直接接続する代わりに、独自のネットワーク内のホストに直接接続できます。

基本的なRelay Proxyのサンプル構成図



LaunchDarkly Relay Proxy は Go で書かれたオープンソース サービスであり、LaunchDarkly によってサポートされ、GitHub リポジトリで入手できます。Go がバイナリ形式で実行できる場所であればどこでも実行できます。Relay Proxy は Docker コンテナとしても提供され、DockerHub で入手できます。

Relay Proxy はサーバー側 SDK を対象としていますが、モバイルおよびクライアント側の評価エンドポイントも提供します。つまり、クライアント側 SDK を LaunchDarkly に直接接続する代わりに、Relay Proxy に対して直接初期化できます。これは、クライアントからの LaunchDarkly トラフィックを許可したくない場合、またはプライベート コンテキスト属性（以下を参照）を LaunchDarkly に送信したくない場合に意味があります。
詳細については、Relay Proxy のドキュメントまたはブログ投稿を参照してください。

リレー プロキシについて: Relay Proxy
ブログ記事: Blog post

オフライン モード

Relay Proxy でオフラインモードを有効にすると、LaunchDarkly に直接接続することなく実行できます。Relay Proxy は、LaunchDarkly のサーバーからフラグとセグメントの値を取得する代わりに、ローカルホストまたはファイルシステムにあるファイルからそれらを取得します。これにより、LaunchDarkly の機能管理ソリューションを利用しながら、FedRAMP High などの隔離された環境でアプリケーションを実行できます。ただし、値が変更されるたびに、フラグ設定をローカル ファイル システムに転送するように調整する必要があります。

> 詳細については、オフラインモード（Offline Mode）のドキュメントを参照してください。

LaunchDarkly に送信されるデータの最小化

組織のセキュリティおよびプライバシー要件によっては、LaunchDarkly に送信するデータを検討することが必要になる場合があります。

コンテキスト データとは?

コンテキスト データとは、アプリケーションが LaunchDarkly に送信する構成可能な属性であり、機能リリースのターゲティング ルールを構成するために使用されます。これには、顧客またはユーザーに関する個人情報、デバイスの詳細、場所の詳細、またはその他の構成可能なプロパティが含まれる可能性があります。

LaunchDarkly SDK を構成して、これらのプロパティに関する属性を収集し、フラグ ターゲティングの目的で LaunchDarkly に送信します。SDK で機能フラグを評価すると、評価にはオブジェクトに関連付けられたキーが含まれます。オブジェクトはコンテキスト データです。機能評価のコンテキストに関する情報を含むさまざまなキーと値のペアが含まれる場合があります。

コンテキストデータには、氏名、電子メールアドレス、その他の固有識別子など、個人を特定できる情報（PII）が含まれる場合があります（コンテキストの定義方法によって異なります）。このデータはビジネスに不可欠な情報である可能性があり、権限のない第三者に公開された場合、重大なリスクをもたらす可能性があります。

すべての組織が異なるリスク特性を持つデータを収集します。機密性の高いユーザーデータを LaunchDarkly に送信しないようにすることができますが、その場合はこの情報に基づいてフラグをターゲットにすることはできません。ただし、以下で説明するプライベートコンテキスト属性を使用して、LaunchDarkly に送信されない情報に基づいてターゲットを設定できます。

プライベート属性

プライベート属性機能を使用すると、サービスが LaunchDarkly に送信するコンテキスト データを制限しながら、そのデータをフラグ ターゲティングに引き続き使用できます。すべての属性をプライベートにすることも、プライベートにする特定の属性を選択することも、特定のコンテキストの属性をプライベートにすることもできます。

サーバー側 SDK でプライベート属性機能を使用すると、データがサーバーから流出することはありません。すべての機能フラグ評価はメモリ内でローカルに発生するため、コンテキスト データを LaunchDarkly に送信する必要はありません。クライアント側 SDK の場合、機能フラグ評価は LaunchDarkly によってホストされるエンドポイントで発生するため、プライベート属性は LaunchDarkly に送信する必要があります（ただし、LaunchDarkly によって保存されません）。前述のように、Relay Proxy を使用して、データを LaunchDarkly に送信せずにクライアント SDK とプライベート属性を使用することを検討できます。

サーバー側 SDK またはクライアント側 SDK のいずれかでユーザー コンテキストに対してプライベート属性機能を使用する場合、ユーザー ID は常に LaunchDarkly に送信されることに注意してください。このため、ユーザー ID は GUID、ハッシュ、またはその他の識別できないデータにすることをお勧めします。

詳細については、プライベート コンテキスト属性の使用に関するドキュメント（Using private context attributes）を参照してください。

匿名ユーザー

匿名ユーザーはコンテキスト リストにユーザーとして登録されないため、LaunchDarkly がユーザーに関して収集する通常のデータは匿名ユーザーには使用できません。匿名ユーザーを使用して個人を特定できる情報（PII）を隠すことができますが、代わりにプライベート ユーザー属性を使用することをお勧めします。ドメイン ユーザー オブジェクトから LaunchDarkly ユーザー オブジェクトを構築するときに、SDK 内の匿名ユーザー ビットを true に設定することで、すべてのユーザーを LaunchDarkly に匿名ユーザーとして強制的に登録できます。

匿名ユーザーまたはプライベート コンテキスト属性を使用する場合、コンテキスト リストには LaunchDarkly にアクセスするユーザーの完全なリストは設定されず、プライベート属性のオートコンプリートは LaunchDarkly では機能しません。

> **詳細については、匿名コンテキスト（Anonymous contexts）のドキュメントを参照してください。**

LaunchDarkly から PII を削除する

PII をコンテキスト データとして LaunchDarkly に送信する必要がある場合は、次の 2 つの方法のいずれかで影響を受けるデータを削除する必要があります。

- ユーザー インターフェイス（UI）のコンテキスト リスト（Context lists）
- コンテキスト インスタンスの削除またはユーザー API エンドポイントの削除の呼び出し（Delete context instance、delete user）

安全な運用のためのベスト プラクティス

アカウント セキュリティ

LaunchDarkly には、アカウント セキュリティのベスト プラクティスをサポートする多くの機能があります。デフォルトでは、ユーザー名とパスワードを使用して LaunchDarkly にサインインします。TOTP ベースの多要素認証（MFA）は、LaunchDarkly のすべてのユーザーに必須です。また、シングル サインオン（SSO）を使用して、LaunchDarkly へのアクセスを制御することもできます。

LaunchDarkly は、SAML 2.0、Okta、Azure AD、OneLogin、および Google Workspace を含む、幅広い SSO プロバイダーをサポートしています。SSO を有効にすると、LaunchDarkly はユーザーの認証と承認のために SSO プロバイダーに依存します。

その他の注目すべきアカウント セキュリティ機能には以下があります。

- REST API にアクセスするための API アクセス トークンを作成できます（API access tokens）
- Webセッションの期間を設定し、アクティブなセッションを取り消すことができます（Session duration）
- トラブルシューティングのために LaunchDarkly サポートチームにアカウントへのアクセスを許可する拡張サポートを有効にできます（Enhanced support）
- Integration, OAuth、webhook を使用して外部アプリケーションを LaunchDarkly アカウントに接続できます（Integrations, OAuth）

認可

LaunchDarkly は、リソースへのアクセスを制御するためにロール ベースのアクセス制御（RBAC）モデルを使用しています。デフォルトでは、LaunchDarkly プロジェクト内のすべてのアカウント メンバーに Reader ロールが割り当てられ、ターゲティング データを含むユーザーとフラグ情報を表示できます。その他のビルトイン ロールには、Writer、Admin、Owner、No Access ロールがあります。
カスタム ロールとポリシーを使用して、プロジェクト、環境、フラグ自体に対して詳細なアクセスを設定できます。詳細については、カスタム ロールのドキュメントを参照してください。（Custom roles）

チーム

ロールの管理を簡素化するために、LaunchDarkly はチームの概念をサポートしています。チームは組織のメンバーのグループです。管理者はチームにカスタムロールを割り当て、メンバーに個別にロールを割り当てるのではなく、グループレベルで権限を制御できます。

これにより、LaunchDarkly の権限を組織構造にマッピングできます。例えば、モバイル チームにモバイル フラグの権限を与え、デスクトップ チームにデスクトップ フラグの権限を与えたり、すべての組織メンバーにステージング環境へのアクセスを与えつつ、特定のチームのメンバーにのみ本番環境でのフラグ制御の権限を与えたりすることができます。

> 詳細については、チームのドキュメントを参照してください。（Teams）

監査ログ

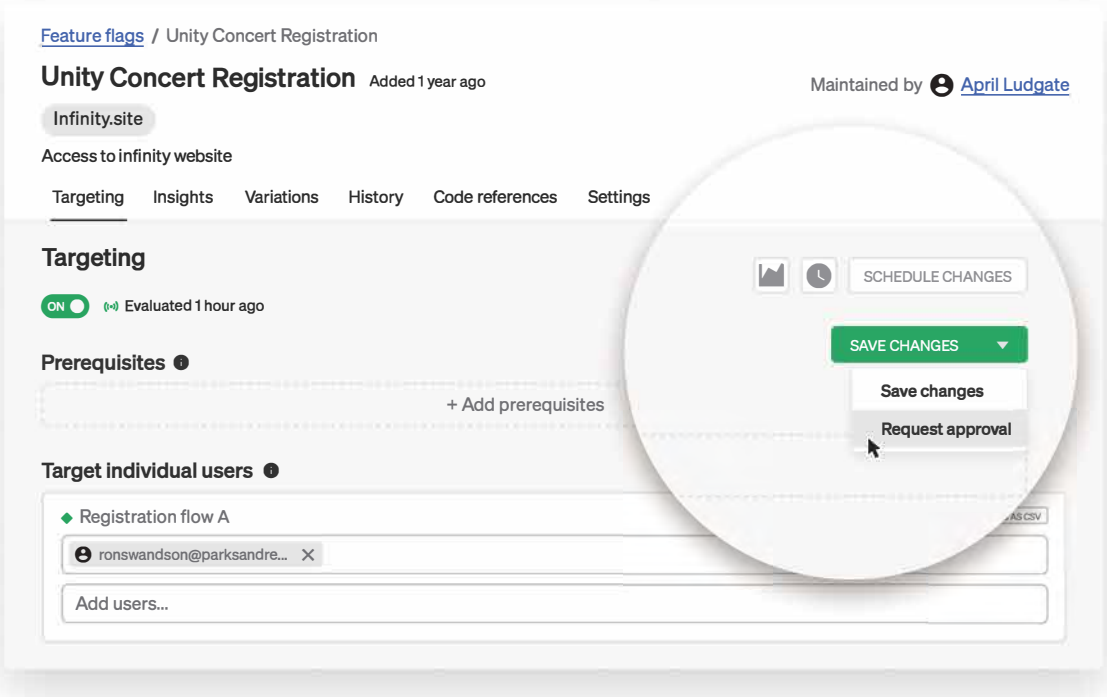
LaunchDarkly は、LaunchDarkly データモデルへのすべての変更の監査ログを維持しています。これらの監査ログは UI で表示したり、API を介してエクスポートしたり、Splunk、Log DNA、CloudTrail Lake、その他の宛先へのインテグレーションを通じて利用したりできます

カスタム ロールとポリシーを使用して、プロジェクト、環境、フラグ自体に対して詳細なアクセスを設定できます。

> 詳細については、監査ログと履歴タブのドキュメントを参照してください（Audit logs and history tabs）

承認

LaunchDarkly は、チームがソフトウェアをどのように構築、出力、制御するかをより適切にマッピングするためのワーク フローをサポートしています。セキュリティ関連のワーク フローの1つは、フラグの変更を行う前にユーザーが承認を要求または必要とすることです。例えば、本番環境で行おうとしている変更をマネージャーにレビューを受けた後、承認してもらいたい場合があります。



LaunchDarkly は、コンプライアンス目的で本番環境への変更を管理するために、すでに変更管理プラクティスを導入しているチームや、サードパーティ ツールを使用しているチームもサポートしています。承認ワーク フローは、ServiceNow や Jira などのツールと統合し、ユーザーがそれらのツール内でリクエストを管理できるようにします。

詳細については、承認のドキュメントを参照してください。(Approvals)

付録

認証

ISO 27001 および 27701

LaunchDarkly は、情報セキュリティ管理システム（ISMS）が ISO 27001（セキュリティ）および 27701（プライバシー）標準に準拠していることの認証を受けています。詳細情報は、署名済みの NDA で提供できます。

SOC2 Type II

LaunchDarkly は、AICPA Trust Services Criteria（TSC）のセットに対して、情報システム制御の定期的な第三者評価を受けています。 SOC 2 レポートは、署名済みの NDA で要求に応じて提供できます。

FedRAMP

LaunchDarkly の Federal インスタンス（LaunchDarkly Federal）は、FedRAMP マーケットプレイスに FedRAMP Moderate Authorized としてリストされています。

LaunchDarkly について

LaunchDarklyは 単なるフィーチャー マネジメントのリーダーではありません - 最初のスケーラブルなフィーチャーマ ネジメント プラットフォームです。フィーチャー マネジメントにより、開発チームは、ソフトウェアが顧客に提供される方法を根本的に変革することで、より速くイノベーションを行うことができます。任意のプラットフォーム上の任意のユーザー セグメントに新しいソフトウェア機能を段階的にリリースする能力により、DevOps チームは安全なリリースを大規模に標準化し、クラウドへの移行を加速し、ビジネス チームとより効果的に協力することができます。今日、LaunchDarkly は 1 日にピーク時 20 兆のフィーチャー フラグをデプロイしており、その数は増え続けています。2014 年にカリフォルニア州オークランドで Edith Harbaugh と John Kodumal によって設立された LaunchDarkly は、Forbes Cloud 100 リスト、InfoWorld の 2021 Technology of the Year リスト、Enterprise Tech 30 リストに名を連ねています。詳細は launchdarkly.com をご覧ください。


LaunchDarkly