ホワイトペーパー

API 侵害の防止策

5 つのタイプの API 侵害を探り、セキュリティを確保する

目次

はじめに

API は、パートナー、サプライヤー、顧客とのデータの交換を通してビジネスをつなぎます。しかし、API のセキュリティは、依然としてほとんどの組織で包括的とは言えない状態にあります。実際のところ、近年、脆弱な API は企業の弱点として狙われるようになっており、攻撃者は API を悪用して機微な情報にアクセスしたり、他の攻撃者に販売したり、世界中に公開したりしています。2024 年には、消費者向け通信、企業向けコンピューティング、仮想コラボレーション業界の世界的ブランドで、API 侵害によって大量の顧客データやその他の機微な情報が漏えいし、財務面と信用面で多額のコストを被っています。

API 侵害とは

簡単に言うと API 侵害とは、大抵は機微な情報にアクセスするために、意図的に API を悪用することを指します。API 侵害のタイプは、さまざまな基準に従って細分化できます。リスクを特定し、本番運用環境での侵害を回避するためには、リスクを 5 つのカテゴリーに分類する以下のようなスキームについて検討すると有益です。

1. **既知の脆弱性**
   - 攻撃者は、パッチが適用されていない既知の脆弱性を悪用します。
2. **シャドウ、不正、ゾンビ、非推奨の API**
   - 管理されていない API や忘れられた API によって運用が脆弱になっている可能性があります。
3. **外部への露出**
   - 資格情報やキーなど、外部に露出している情報が管理されていない場合があります。
4. **設定ミスやオペレーターによるエラー**
   - インフラやサービスのセキュリティ設定ミスによって攻撃者の侵入口が作成され、悪用される可能性があります。
5. **未発見の脆弱性やバグ**
   - 管理者が最善を尽くしても、攻撃者はバグや脆弱性を探し出して本番環境に侵入しようとします。

この e ブックでは、こうした 5 つのタイプの API 侵害のそれぞれにおいて、どこでセキュリティの失敗が発生するのか、そしてそれをどのように防止できるのかについて、解説します。また、API セキュリティを最大限高め、リスクを最小化するために、API セキュリティプログラムにおける特定の弱点を排除できるように支援することも目指します。

侵害のタイプ：既知の脆弱性

おそらく最も一般的なのが、（パッチが適用されていない）既知の脆弱性を悪用する API 侵害です。サイバー犯罪者は、データを入手したい場合、一般的な最初のステップとして、組織がバックドアを開けたままにしていないかを確認します。

2024 年 1 月、攻撃者は認証制御されていない API エンドポイントを悪用して、広く使用されているプロジェクト管理ツールに不正に侵入しました。攻撃者は API に侵入後、数百万人のユーザー情報に不正アクセスし、数か月後にはメールアドレスや取締役会の情報など、21GB 以上のデータがインターネット上に漏えいしました。

API の最も一般的な問題の 1 つに、認証と認可に関する問題があります。「OWASP Top 10 API セキュリティリスク」は、認証の不備など、組織が保護するべき最も重要な 10 タイプの API 脆弱性について情報を提供しています。

組織は、OWASP Top 10 に含まれている 10 タイプのリスクから API を保護するだけでなく、MITRE が運営する連邦政府出資研究開発センター（FFRDC）が作成した共通脆弱性識別子 CVE（Common Vulnerabilities and Exposures）の全リストに照らして API コードを保護する必要があります。よく知られている脆弱性として、Apache log4j 2（CVE-2021-44228）、別称「Log4Shell」を思い起こす方もいるでしょう。Log4j ライブラリ（Java プログラミング言語の一般的なオープンソースログライブラリ）にバグがあるために、攻撃者が任意のコードをリモートで実行してシステムにアクセスできる状態になっていました。攻撃者は、企業のシステムにこうした既知の脆弱性がないか定期的に調査しています。

米国では、Cybersecurity and Infrastructure Security Agency（CISA）が、既知の CVE のカタログを管理しています。その他の国でも同様のカタログが管理されていると思います。

OWASP Top 10 API セキュリティリスクのリストは 2019 年に作成され、2023 年に更新されました。有益なリストですが、アタックサーフェスの変化のスピードには追いつけません。2024 年だけでも 2 万 4,000 以上の新しい CVE が CISA のカタログに追加されており、そのうち 500 以上が API に関連しています（2024 年 8 月中旬現在）。

既知の脆弱性から組織を完全に保護するためには、次のような二面的な対策が必要です。

1. 開発およびテストプロセスが、既知の脆弱性の本番環境への導入を回避できるだけの堅牢性を備えているか確認する。

2. 新たな脆弱性が特定されたら、可能な限り迅速にパッチを適用する。

多くの組織は、この 2 つのステップの両方に苦戦しています。さらに、別の一連の脆弱性をもたらす可能性のあるサードパーティソースの API とコードを使用しています。2022 年にある研究チームが、自動車業界の複数のメーカーに影響を及ぼす API の重大な欠陥を発見しました。こうした欠陥によって機微な顧客情報や車両の位置情報までもが漏えいし、侵害されたリモート管理システムを介して車両のロック解除、始動、無効化が可能な状態になっていました。

防止策

既知の脆弱性による API 侵害から組織を保護する一般的な方法として、セキュリティパッチがリリースされた際は迅速にソフトウェアとシステムをアップデートすることが挙げられます。また、開発およびテストのプロセスが包括的であり、API セキュリティのベストプラクティスに従っていることを確認することも重要です。これには次のことが含まれます。

- **ソフトウェアサプライチェーンのセキュリティ確保：**使用するライブラリ、オープンソースソフトウェア（OSS）、その他のサードパーティコードがセキュアであることを確認します。
- **シフトレフトセキュリティテストの実装：**API セキュリティとソフトウェアテストに関連するタスクを、開発プロセスの早い段階に移行します。このようにすることで、コーディングのエラーや、ソフトウェアやアップデートを迅速にリリースするというプレッシャー下にある開発者チームによる設定ミスなどの脆弱性を発見しやすくできます。
- **API セキュリティ対策管理の活用：**API の探索、機微な情報の特定、脆弱性の検知を組み合わせて、まず最も重要な API に修復作業を集中させることができます。

Akamai API Security の機能

Akamai API Security を利用することで、スピードを犠牲にせずに、新しいビルドのたびに既知の脆弱性を減らすことができます。API Security は、API 特有の脆弱性に包括的に対応する、API セキュリティテスト専用に開発されたソリューションです。Active Testing を使用して、開発のあらゆる段階に API セキュリティテストを組み込むことができます。

- **すべての API を検索してテスト**：アプリケーションのビジネスロジックの理解に基づいてすべての API を検索してテストできます。
- **シフトレフト**：テストをソフトウェア開発ライフサイクル全体に統合し、通常よりも前倒しして実施できます。CI ／ CD プロセス全体を通じて、さまざまな状態や環境にわたり動的な API の可視性を確保できます。
- **開発者をサポート**：シンプルなセットアップ、自動化、インラインテスト結果、特定された問題を状況に応じて修正できるガイダンスなど、クラス最高の操作性で開発者をサポートします。

また、API Security の対策管理によって、トラフィック、コード、設定を包括的に把握して、自社の API のセキュリティ対策を評価できます。API Security は、ログファイル、トラフィック履歴の再現、設定ファイルなど、可能な限り広範なソースを参照して、脆弱性を検知します。また、OWASP Top 10 API セキュリティリスクに挙げられているすべての脆弱性も検知します（対策管理の詳細については、「設定ミスやオペレーターによるエラー」のセクションを参照）。

侵害のタイプ：シャドウ、不正、ゾンビ、非推奨の API

見えないものは保護できません。多くの企業では、API の大部分が管理されておらず、API 環境で認識されていない、または把握されていないシャドウ、不正、ゾンビ、非推奨の API ターゲット（次のページのサイドバーを参照）が生み出されています。さらに攻撃者は、組織の無防備な API に着目し、古いバージョンを見つけるために値を曖昧にしたり、修正したりして、悪用可能な API の亜種を頻繁に探し回っています。

氏名、住所、誕生日、政府発行の ID 番号を含む 1,120 万件以上の顧客記録を誤って漏えいさせたオーストラリアのある大手通信会社でも、こうしたことが起こっていました。この攻撃では、何らかの方法でオープンなインターネットからアクセスできるようになったテスト用の API が悪用されました。この不正な API には認証チェックが欠けていたため、攻撃者は数百万件ものレコードを要求して受信することができました。

ほとんどの組織では、さまざまなレガシー API と新しい API を使用して業務を行っています。こうした API とともに、企業をさまざまなサイバーセキュリティリスクや運用上の課題にさらす不正な API、ゾンビ API、シャドウ API が見つかることは、残念ながらよくあります。

こうした認識されていない API には、次のようなさまざまなソースがあります。

- **商用 API：**一部の商用ソフトウェアパッケージには、他のアプリケーションや外部データソースと接続するための API が含まれています。こうした API は、誰にも気付かれないままアクティベートされる可能性があります（API の徹底的な探索によって解決できる問題です）。
- **古いバージョンの API：**多くの場合、古いバージョンの API はセキュリティが弱い可能性や既知の脆弱性がある可能性が高いのですが、削除されない場合があります。ソフトウェアの更新中に古いバージョンを新しいバージョンと共存させる必要がある場合がありますが、プロセスの不備によって古い API がシャットダウンされないと、ゾンビ API になります。
- **ショートカットとプロセスの失敗：**シャドウ API は、適切な関係者に情報を提供していないことによって発生します。たとえば、事業チームが、IT チームやセキュリティチームに通知せずに特定のニーズに対応する API を作成したり、開発者が手順に従わなかったりすることがあります。
- **継承された API：**合併や買収の一環として継承された API も見落とされることが多く、シャドウ API になります。
- **再有効化されたコード：**古いバージョンの API を誤って再有効化してしまう場合もあります。

防止策

手動で API 監査を行い、正確なインベントリを作成する必要のあるすべてのインプットを文書化するためには、特に検出されたすべての API を評価して適切に処理する時間を考慮すると、数時間はかかります。すでに過重労働の状態にあるセキュリティチームにとって、これは現実的な方法ではありません。不正 API、ゾンビ API、シャドウ API の爆発的な増加からビジネスを保護するためには、使用しているあらゆるタイプのすべての API を識別できる自動 API 探索機能が必要です。運用環境全体のすべての API を検索してインベントリを作成し、API ゲートウェイによって管理されていない API と API ドメインを探索する必要があります。

Akamai API Security の機能

API Security は、幅広い統合ソースを活用して、未加工のトラフィック、ログなどの API データをインジェストします。API Security は、こうしたソースから生成されたデータによって、API、API の設定ミス、脆弱性、API の悪用を特定します。当社の探索ツールは、OWASP Top 10 API セキュリティリスクに挙げられているすべての脆弱性を検知します。

追加の探索機能により、次のことが可能になります。

- 設定やタイプ（RESTful、GraphQL、SOAP、XML-RPC、JSON-RPC、gRPC など）を問わず、すべての API を探索してインベントリを作成する
- 休眠 API、レガシー API、ゾンビ API を発見する
- 忘れられているドメイン、見落とされているドメイン、またはその他の不明なシャドウドメインを特定する
- API インベントリを管理し、API ドキュメントの正確性を確保する

攻撃者が探し回る、管理されていないハイリスクな API

シャドウ API（「文書化されていない API」）は、組織の公式の監視チャネルの外に存在し、稼働しています。善意の開発者によって作業を迅速化するために作成される場合もあれば、以前のソフトウェアバージョンの残余物である場合もあります。

不正な API は、システムやネットワークにセキュリティリスクをもたらす不正な API や悪性の API です。

ゾンビ API には、新しいバージョンの API や完全に別の API に置き換えられた後も稼働したままになっている API が含まれます。

非推奨の API は、API が変更されたために使用を推奨されなくなった API です。非推奨のクラス、メソッド、フィールドが依然として実装されており、今後の実装で削除される可能性があるため、新しいコードで使用するべきではありません。

侵害のタイプ：外部への露出

API の外部の脆弱性は通常、API キーや資格情報の漏えい、API コードやスキーマの露出、ルーズなドキュメント管理、リポジトリの脆弱性など、不適切な手法や手順上のエラーの結果として生み出されます。運用範囲外の潜在的な攻撃ベクトルを探索する機能が不可欠になっています。この 1 年間に、API キーなど、外部ソースの認証情報が誤って公開されたことで、注目を集める侵害が多数発生しています。たとえば、ハッカーはフィッシングキャンペーンによって Dropbox の 130 のソースコードリポジトリに不正にアクセスしました。ハッカーはこれを悪用して、GitHub に不適切に保存されている API キーにアクセスしました。こうした露出は非常に一般的になっているため、GitHub は API キーなどの機密の漏えいを防ぐ措置を講じましたが、他の公開リポジトリには依然として脆弱性が存在する可能性があります。

外部に情報が露出しているもう 1 つの事例として、Twitter の API のキーを露出させているモバイルアプリが 3,000 以上あることが判明しています。開発者は利便性のために開発時に API キーをアプリケーションコードに埋め込むことが多いため、こうしたミスは驚くほどよく見られます。そうした埋め込みキーをパブリックリリース前に削除しないと、それがキーが露出する潜在的な原因となります。

防止策

こうした外部への露出を削減または排除するためには、次のような二面的なアプローチが必要になります。

- 漏えいしたキーや認証情報、リポジトリの不適切な使用など、露出の原因を特定し、排除するための手順を強化します。
- 外部のアタックサーフェスを定期的にスキャンして、脆弱性を検知して修復します。

広範に及ぶ API の脅威から身を守るには、（「不正な API からの侵害」セクションで説明している）内部から外部への探索と、外部から内部への探索を両方行って露出を特定し、外部のアタックサーフェスを縮小させる必要があります。

Akamai API Security の機能

API Security は、ハッカーが使用する偵察テクニックをシミュレートし、問題を迅速に発見して修正できるようにすることで、常に攻撃者に先手を打てるようにサポートします。API Security は、外部から内部への探索により、攻撃を受ける前に、外部のアタックサーフェスを定期的かつ自動的にスキャンして、脆弱性を検知し、次のことをできるようにします。

- **公開されている脆弱性を発見：**API キーや認証情報の漏えい、コードの露出、設定ミス、リポジトリの脆弱性など、重要な問題を迅速に検出して修正できます。
- **自社に関連するドメインとサブドメインを探索：**インターネットレジストラ、認定レジストラ、オープンソースなど、さまざまなソースから収集されたデータを活用できます。
- **実際の攻撃手法を活用：**外部の偵察を実行する攻撃者をシミュレートし、企業のドメインまたはサブドメインに対して限定的なクエリーを実行して情報を収集できます。

侵害のタイプ：設定ミスやオペレーターによるエラー

多くのサイバー攻撃者は、API トラフィックを仲介し、保護するサーバー、ネットワーク、API ゲートウェイ、ファイアウォールの設定ミスを悪用して侵入します。IBM Security X-Force の調査によると、クラウド侵害の 3 分の 2 は、API の設定ミスに関連しています。セキュリティ設定ミスは、安全性の低いデフォルト設定、アクセス制御されていないクラウドストレージ（驚くべきことによくあります）、不完全な設定や場当たり的な設定によって発生する可能性があります。デジタルフットプリントが拡大するにつれて、複数のパブリッククラウドの可用性ゾーンやパブリッククラウド（AWS、Microsoft Azure、Google Cloud など）をはじめとした、より多くのロケーションに運用環境が拡大する可能性があります。多くの場合、こうした環境はさまざまなセキュリティ制御の下で稼働しているため、セキュリティを正しく設定することが複雑で困難になっています。

防止策

インフラ側でのセキュリティ設定ミスを防止する最善の方法の 1 つに、サーバー、ネットワークデバイス、ゲートウェイ、ファイアウォールの手動設定を可能な限り回避することが挙げられます。企業の管理チームがインフラとアプリケーションのセキュリティ制御を日常的に手動で設定している場合、または定期的に「微調整」している場合、設定上の脆弱性が生じる可能性が高まります。

セキュリティには、自動化が最善の策です。一部の企業は、手作業によるミスを回避する手段として、不変のインフラという考え方を取り入れています。

インフラ、サービス、API を確実に完全に保護するためにできる限りのことをしたとしても、API 体制の管理機能が必要です。体制管理機能は、API ライフサイクル全体を通じて API のセキュリティを管理、監視、維持するツールを備えています。

API Security の機能

API Security の体制管理モジュールは、API コールとインフラを分析して、設定ミスを特定します。こうした設定ミスは通常、Amazon S3 バケットに関する問題、認証されていない API 上の機微な情報、Kubernetes アクセスベースのさまざまな設定ミスに分類されます。

体制管理モジュールは、トラフィック、コード、設定の包括的なビューを提供し、API および Web アプリケーション全体のアタックサーフェス全体を把握できるようにします。こうした情報には、個人を特定できる情報など、API を介して移動するあらゆる形式の機微な情報も含まれます。また、体制管理モジュールを使用して、API 管理ツールが強力なプロトコルと暗号を使用していることを確認し、こうした機微な情報を露出させる可能性のある脆弱な暗号化を回避することもできます。さらに、有効期限が切れた JSON Web トークンを API が受け入れると、不正アクセスが可能になり、セキュリティリスクが増大するため、そうしたトークンを受け入れるべきではありません。このモジュールは、安全でないポートをリダイレクトせずにリッスンするアプリケーションロードバランサーなどの設定ミスも防止できます。こうしたあらゆる対策によって、API のセキュリティ体制を総合的に強化し、潜在的な脅威に対してより回復力の高い防御を実現できます。

侵害のタイプ：未発見の脆弱性

大半の侵害タイプと同様に、サイバー犯罪者はインフラを定期的にスキャンして、CVE や OWASP API Security Top 10 などの一般的な設定ミスに加えて、不正な API、ゾンビ API、シャドウ API も探し回っています。また、露出している　API も調査して、ライブラリやオープンソースコードなどの公開コード、API 環境のコーディングエラー、バグ、設定ミスに含まれる悪用可能な新たな脆弱性を探し回っています。こうした脆弱性によって、サイバー犯罪者は API コールを操作し、リクエストに文字列を挿入することができます。その結果、サイバー犯罪者が使用するテクニックは常に進化しています。

防止策

コードに可能な限りバグや脆弱性がないようにすることは、防御の重要な要素です（「既知の脆弱性」のセクションを参照）。しかし、攻撃者がバグを発見するということや、API を悪用するためにキーや認証情報にアクセスするということを前提にする必要があります。

API ランタイム保護は、既知、未知を問わず、あらゆる脆弱性を悪用するハッカーを特定するように設計されています。API ランタイム保護は、これまで特定されていなかったバグや設定ミスから API 環境を保護する唯一の手段であり、認証情報やキーが侵害された場合に最適な保護を提供します。

ランタイム保護によって、API の使用やデータアクセスにおける通常とは異なるパターンや異常を特定し、レーダーをすり抜ける可能性のある進行中の攻撃を特定して、数千件または数百万件のデータレコードが抽出される前に修正できるようにします。

API ランタイム保護によって、次のような悪性の API リクエストを特定してブロックできます。

- API から大量の機微な情報を引き出す攻撃
- オブジェクトレベル認可の不備（BOLA）攻撃

API ランタイム保護ソリューションは、以下を検知できます。

- データ漏えい
- データポリシー違反
- API セキュリティ攻撃
- データ改ざん
- 疑わしいふるまい

さらに、ランタイム保護は、API トラフィックのロギング、機微な情報へのアクセスの監視、脅威の検知、攻撃ベクトルのブロックや修復を行います。

API Security の機能

他の防御策が不足している場合は、ランタイム保護を防御の最後の手段として考える必要があります。ランタイム保護の主な機能は、API 攻撃をリアルタイムで検知してブロックすることです。自律的な機械学習（ML）ベースの監視を活用して、リアルタイムのトラフィック分析を行い、データ漏えい、データ改ざん、データポリシー違反、疑わしいふるまい、API セキュリティ攻撃に関する、コンテキストに沿った知見を提供します。API Security は、API トラフィックの異常や潜在的な脅威を検知し、事前に選択したインシデント対応ポリシーに基づいて修復を促進します。

API Security は ML を活用して、各 API の動作モデルを構築します。この正常な動作のベースラインを使用して、API ビジネスロジック攻撃を検知します。ランタイム保護によって生成されるすべての問題には、重大度、ステータス、OWASP API Security Top 10 へのマッピング、および該当する場合は攻撃者の詳細が含まれています。問題には、攻撃者のセッションの詳細や API リクエストのコピーなどの証拠も含まれており、問題の優先順位付けや修復に役立てることができます。

API Security のランタイム保護は、API 攻撃のリアルタイムの検知と防止に加えて、API の設定ミスの継続的な検知を行い、運用と修正をシンプル化する多くの一般的なワークフローの統合も実現します。

API Security が WAF、API ゲートウェイ、ITSM、SIEM、その他のワークフローツールと統合して、攻撃に対する総合的な防御を提供することは、おそらくチームにとって最大の朗報でしょう。脅威の修復を完全に自動化することも、さまざまなレベルの手動操作を選択して可視性や制御性を高めることもできます。

5 つの侵害タイプ、5 つの防止原則

API がサイバー犯罪者にどのように悪用されているか理解が深まり、これでその防止に取り組めるようになったと思います。ここで、併用する必要がある 5 つの予防ツールと戦略的視点を紹介したいと思います。

1. シフトレフトの API セキュリティ

- シフトレフトの API セキュリティとは、本番環境の脆弱性を露出させて、サイバー犯罪者がそれを発見することがないように、開発時に API を徹底的にテストすることを指します

2. 内部から外部への探索

- 運用環境全体のすべての API を特定します

3. 外部から内部への探索

- 漏えいしたキーや認証情報、不適切なリポジトリの使用などの露出源を特定して排除し、外部のアタックサーフェスを定期的にスキャンして、脆弱性を検知して修復します

4. 包括的な対策管理

- API セキュリティに関しては、設定ミスや脆弱性を回避することで、常に最善の対策を講じる必要があります

5. ランタイム保護

- 異常な API アクティビティを検知し、未確認の脆弱性やバグを含む、あらゆる脅威から保護します

デモのお申し込み

Akamai API Security の動作を実際にご覧いただくと、API の設定ミスを特定して修正し、悪質な API 攻撃から身を守ることがいかに簡単に実現できるかを、体験していただけます。大手エンタープライズ組織が当社の API セキュリティソリューションを選択している理由を、ぜひ直接ご確認ください。

デモを依頼する

Akamai のセキュリティは、パフォーマンスや顧客体験を損なうことなく、ビジネスを推進するアプリケーションをあらゆる場面で保護します。当社のグローバルプラットフォームの規模と脅威に対する可視性を活用して、お客様と Akamai が提携し、脅威を防止、検知、緩和することで、ブランドの信頼を構築し、ビジョンを実現できます。Akamai のクラウドコンピューティング、セキュリティ、コンテンツデリバリーの各ソリューションの詳細については、akamai.com および akamai.com/blog をご覧いただくか、X (旧 Twitter) と LinkedIn で Akamai Technologies をフォローしてください。公開日：2024 年 11 月。