ホワイトペーパー

Fastly で OWASP トップ10の脅威を乗り切る方法

Web アプリケーションセキュリティの強化に向けたインサイト、戦略、実例

fastly

はじめに

あらゆる組織が攻撃対象：攻撃者は、Web アプリや API が宝の島に通じる扉の鍵を握っていることをよく知っています。アプリや一般公開されている API の侵害に成功した攻撃者は、目的の達成に向けて行動することができます。開発者やセキュリティ担当者の間で脅威や脆弱性に関する理解を高めるために作成された OWASP トップ10リストは、セキュアなソフトウェア開発環境の促進に貢献しています。

OWASP トップ10は、非営利組織の OWASP (Open Web Application Security Project) Foundation によって作成されたスタンダードで、Web アプリケーションにおける最も深刻なセキュリティリスクのトップ10ランキングとそれらのリスクへの対策に関するガイダンスを提供しています。同リストは世界中のセキュリティ専門家のコンセンサスに基づいており、Web アプリケーションセキュリティの複雑な世界において、集中して取り組むべき領域を明確にすることを目的としています。

OWASP は2003年以降、アプリケーションセキュリティ業界における技術の進化や変化に基づいて同リストを数年ごとに更新しています。2021年に大幅に更新された最新バージョンでは、トップ10リストに新たに3つのカテゴリーが追加されたほか、4つのカテゴリーで名称とスコープが変更され、統合されたものもいくつかありました。

「**Verizon DBIR 2023**」では、すべてのセキュリティ侵害のうち、60%が Web アプリケーション経由で発生したという驚くべき結果が報告されています。OWASP トップ10のリスクに単一のプロダクトまたはプロセスで対処できるという考えは魅力的ですが、実際に複数の場所に異なる種類のアプリケーションを抱える組織は、アプリケーションセキュリティについて包括的な視点で考える必要があります。攻撃者は最も脆弱な場所を常に探しています。そのため、OWASP トップ10のリスクに対処するために、オンプレミスとクラウド環境全体にわたって一貫したセキュリティポリシーを適用する必要があることが多く、プロダクトやプロセス、開発者の啓蒙の強化が含まれます。

このホワイトペーパーでは、OWASP トップ10の各脅威について例を交えながら詳しく解説し、これらのカテゴリーの脅威から組織を保護する Fastly のソリューションをご紹介します。

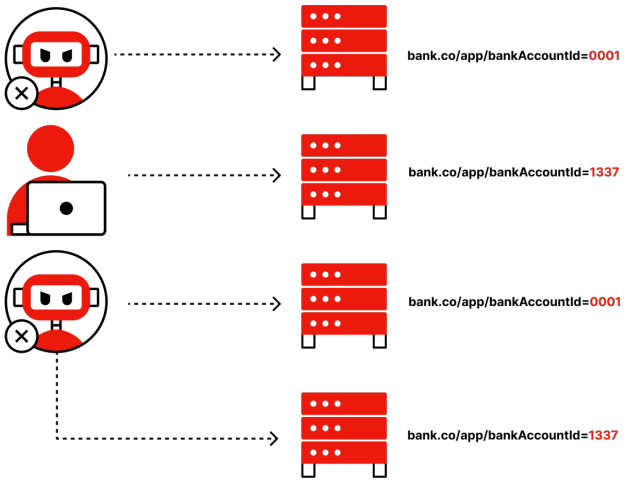
A01:2021

アクセス制御の不備

このカテゴリーは、アプリケーションにおける最も深刻なセキュリティリスクを意味します。攻撃者が必要なレベルの権限を得てアクセス制御をバイパスできる場合、深刻な被害をもたらす可能性があるためです。OWASP によると、アプリケーションの94%で何らかのアクセス制御の不備が確認され、提供されたデータセットの中で最も発生数の多いリスクでした。このカテゴリーには、Common Weakness Enumerations (CWE : 共通脆弱性タイプ一覧) で指定されている脆弱性タイプのうち最多の34件が含まれ、本来アクセスが許可されるべきでないデータやリソース、ユーザーアカウント、操作に攻撃者がアクセスすることを可能にします。

攻撃シナリオの例

攻撃者 (ユーザー ID: 1337) が攻撃対象の URL にアクセスし、アプリケーションに自分が別のユーザー (ユーザー ID: 1001) であると伝えます。アプリケーションはアクセス制御の強制に失敗し、リクエストがこの「別のユーザー」によって送信されたものとみなされることを許可してしまいます。



Fastly のソリューション

攻撃者 (ユーザー ID: 1337) が攻撃対象の URL にアクセスし、アプリケーションに自分が別のユーザー (ユーザー ID: 1001) であると伝えます。アプリケーションはアクセス制御の実行に失敗し、リクエストがこの「別のユーザー」によって送信されたものとみなされることを許可してしまいます。

- **Next-Gen WAF** の設定では、許可/ブロックのリストを使用できますが、認証済みのユーザーまたは管理者のみがこれらにアクセスできるよう強制することができます。この機能は、Fastly のエッジクラウドプラットフォームと連動します。これにより、アクセス制御リスト (ACL) を使用して Next-Gen WAF によるセキュリティ決定を攻撃者の近くで実行し、アプリケーションに達するはるか手前で悪意のある攻撃者をブロックできます。

- **Next-Gen WAF** は、強制ブラウジングやディレクトリトラバーサル、プライベートファイルや「/web_app/WEB-INF/webapp.properties」のようなディレクトリへのアクセスを試みるリクエストをデフォルトで検出します。簡単な設定によってこの機能を拡張し、他のディレクトリやファイル拡張子を検出に含めることができます。こうすることで、以下の OWASP の推奨事項を実施できます。

  - *Web サーバーのディレクトリリスティングを無効にし、ファイルのメタデータ (.git など) やバックアップファイルが Web ルート内に存在しないことを確認します。*

- **Next-Gen WAF** は、クレデンシャルスタッフィングなどのブルートフォース攻撃を検出し、攻撃者のログイン試行をブロックしてアプリケーションを保護します。失敗したログイン試行はすべてログに記録され、攻撃が検出されるとすぐに管理者に警告します。これにより、以下の OWASP の推奨事項を実施できます。

  - *アクセス制御の失敗をログに記録し、必要に応じて (失敗が繰り返されている場合など) 管理者に警告します。*

- **Next-Gen WAF** は API リクエストにレート制限を適用し、しきい値を超えるとアラートが発生します。これにより、以下の OWASP の推奨事項を実施できます。

  - *API とコントローラーのアクセスにレート制限を適用し、自動攻撃ツールによる被害を最小限に抑えます。*

対策

セキュアな開発ライフサイクルを念頭に置いてカスタムビルトのアプリケーションを構築し、セキュリティ・ペネトレーション・テストを実施することで、組織はアクセス制御の不備からアプリケーションを保護できます。さらに、WAF などのセキュリティツールの実装によって、セキュリ体制を強化できます。

アクセス制御は、攻撃者がアクセス制御のチェックやメタデータを変更できないサーバーレス API や信頼できるサーバーサイドコードでのみ効果があります。この例では、**ルートコントローラーが承認を確認**し、ユーザーがレコードの読み取り、作成、更新、削除を実行できることを受け入れるのではなく、所有権へのアクセス制御を強制する必要がありました。

A02:2021

暗号化の失敗

このカテゴリーは、暗号化 (および暗号化の欠如) による失敗に焦点が当てられています。暗号化の欠如、または暗号化が適切に実装されていない場合、機密性の高いデータ (パスワードやクレジットカード番号、健康記録、個人情報、企業の機密情報など) の漏えいにつながることがよくあります。

攻撃シナリオの例

以下のような一般的な例のシナリオが多くの環境でみられます。

パスワードデータベースで、ソルトなしのハッシュ、または単純なハッシュを使用して全ユーザーのパスワードを保存されています。ファイルアップロードの欠陥を悪用して攻撃者がパスワードのデータベースを取得します。事前に計算されたレインボーテーブルを利用して、ソルトなしのすべてのハッシュが解読される可能性があります。単純な、または高速なハッシュ関数を使用して生成されたハッシュは、ソルトありでハッシュ化されていても、GPU によって解読されることがあります。

対策

自前の機能ではなく、標準化された暗号化ライブラリを使用することで、組織は暗号化失敗のリスクを軽減できます。これに徹底したコードレビューとペネトレーションテストを組み合わせることで、このような脅威に対する初期の保護レイヤーを実装することが可能になります。

こうした攻撃シナリオを回避するため、Argon2、scrypt、bcrypt、PBKDF2 などのワークファクター (遅延要因) を伴う強力かつ適応可能なソルト付きハッシュ関数を使用してパスワードを保存することを OWASP は推奨しています。

優れたベストプラクティスの実例として、Blowfish 暗号アルゴリズムをベースにした **bcrypt 関数** (好みの言語のライブラリからが理想) の使用が挙げられます。この関数ではソルトと適応関数が使用されるため、時間が経つと共に反復数が増加し、遅くなります。すなわち、長期的にブルートフォース攻撃に対する抵抗力を維持できるわけです。

Fastly のソリューション

- セキュア・バイ・デフォルトを実践する (TLS 1.3 の使用や脆弱な暗号の排除など) **Fastly のエッジクラウドプラットフォーム**は、TLS 終端を処理し、グローバル規模の高速化を実現します。同プラットフォームでは、アプリケーションの機密性が高い領域へのリクエストを含むアプリケーションリクエストに脆弱な暗号化やプロトコルが使用されることがありません。これにより、以下の OWASP の推奨事項を実施できます。
  - *完全前方秘匿性 (PFS) が有効化された TLS、サーバーサイドによる暗号スイートの優先度決定、セキュアなパラメータなどの安全性の高いプロトコルを使用して通信経路上のすべてのデータを暗号化します。HTTP Strict Transport Security (HSTS) のようなディレクティブによって暗号化を強制します。*
  - *さらに、Fastly のプログラマブルなエッジネットワークは、暗号化ハッシュの生成やデジタル署名の認証を行う開発ツールを提供します。一般的なツールを使用してセキュリティを一元化することで、リスクの増大につながる恐れがある、自社製の暗号化を使用する危険を削減できます。*

- **Next-Gen WAF** は、必要なレスポンスセキュリティヘッダーが見当たらない場合に、リクエストをログに記録してアラートを発信するよう設定することが可能です。

A03:2021

インジェクション

インジェクション攻撃でも特に一般的なものに、SQL、シェルおよびコマンドライン、オブジェクト・リレーショナル・マッピング (ORM)、LDAP、OGNL 式 (struts テンプレートなど) のインジェクションがあります。2021年に OWASP はクロスサイトスクリプティング (XSS) も、このカテゴリーに追加しています。それまでは XSS というカテゴリーが存在しました。XSS がインジェクションのカテゴリーに含まれたからといって、その脅威の深刻度が下がったわけではありません。ただ近年、他の脆弱性がより拡大していることが背景にあります。

インジェクションは2番目に発生件数の多いアプリケーション攻撃です。インジェクションがどのような状況で発生する攻撃かを具体的にみてみましょう。

- ユーザーが提供したデータが、アプリケーションによって検証、フィルタリング、サニタイズされない。
- コンテキストに応じたエスケープが行われず、動的クエリまたはパラメータ化されていない呼出しがインタープリタで直接使用される。
- 悪意のあるデータが直接または連結して使用される。動的クエリやコマンド、ストアドプロシージャにおいて SQL やコマンドにそのような構造と悪意を持ったデータが含まれる。

攻撃シナリオの例

以下のような脆弱な SQL の呼び出しの作成において、受信したユーザークエリから抽出された、信頼できないデータがアプリケーションによって使用される場合があります。

```
String query = "SELECT \* FROM accounts WHERE custID='" + request.getParameter("id") + "'";
```

対策

インジェクション攻撃を防止する最善策は、パラメータ化されたクエリや似たような構造体 (ORM ツール) を使用し、コマンドやクエリからデータを常に切り離しておくことです。他の防御対策として、ポジティブ検証、すなわち「許可リスト」によるサーバーサイドの入力検証を用いることが挙げられます。モバイルアプリケーション用のテキスト領域や API など、多くのアプリケーションで特殊文字が必要とされるため、これは完全な防御方法とは言えません。それでも、前バージョンのトップ10リストに比べてこのカテゴリーのランクが下がった理由のひとつとして、多数のフレームワークでこのようなアプローチがデフォルトで使用されていることが考えられます。

Fastly のソリューション

- Next-Gen WAF は、デフォルトでインジェクション攻撃を検出するので、ルールの追加やチューニングが不要です。従来型の WAF テクノロジーは、SQLi などのインジェクション攻撃を検出するのに、何万もの正規表現 (regex) パターンを必要とします。インジェクション攻撃の検出に必要なパターンの数は、各 HTTP リクエストを、増える一方の正規表現パターンと照合して検査するコンピューティング能力の拡大と共に経時的に増大し続けます。正規表現のパターンマッチングでは通常、多くの誤検知が発生するため、WAF をブロックモードで使用し続けるには、WAF のチューニングを行う専門チームを社内に設けるか、または高いコストを費やして WAF の管理をアウトソーシングする必要があります。新しいバージョンのアプリケーションが迅速にリリースされる環境では、この問題の影響がより深刻になります。
- Next-Gen WAF は、Smart Parse と呼ばれる独自の検出方法を使用し、悪質または異常なペイロードが存在するかどうかを、インラインで即座に判断します。SmartParse はリクエストのコンテキストや実際の実行内容を評価することで、非常に正確な判断を行うことができます。有効性を高めるため、社内リサーチとクローズドループのプロダクト内の誤検知フィードバック機能に基づいて、SmartParse は継続的に更新されます。正規表現に依存し、ブロックモードではほとんど使用されない WAF に対し、Fastly のお客様の90%が、チューニングなしでデフォルトのすべての攻撃タイプに対してフルブロックモードを有効にしています。

A04:2021

安全が確認されない不安な設計

この広範なカテゴリーには、「欠落した、または不十分な制御設計」による、さまざまな脆弱性が含まれます。不十分な制限、不正確な特権の割り当て、機密情報を含むキャッシュされたクッキーの使用、その他多くの直接テストできない脆弱性の欠陥など、このカテゴリーの CWE の多くがグループ化されています。残念ながら多くの開発者にとって、これらのカテゴリーグループの一部は、単一のカテゴリーとして考えるのが容易ではなく、見落とされる可能性があります。そこで、開発者がアプリケーションの各コンポーネント (認証や認可など) を確認し、最小のパーツに分解することが役立つかもしれません。こうすることで各パーツを個別にテストしやすくなり、一般的にセキュリティと信頼性の両方の点においてより望ましい結果が得られます。通常、攻撃者は特定の攻撃ベクトル (欠陥のある認証設計など) に狙いを定め、同じ脆弱性を抱えるその他の攻撃対象を探すためです。

攻撃シナリオの例

ある映画館チェーンでは団体予約による割引を提供していますが、最大15名までは予約保証金を支払う必要がありません。攻撃者は、この人数制限を確信すると、予約システムを悪用して数回のリクエストで600席とすべての映画館を一度に予約できるかどうかをテストし、多額の売り上げの損失を引き起こす可能性があります。

対策

このカテゴリーの性質上、テストを実施するのが困難です。この種の悪用を最も効果的に阻止するには、「シフトレフト」のアプローチを実践し、コードを記述してテストする方法を変更する必要があります。以下は、コードにおけるこのような種類の脆弱性を回避する方法の例です。

- 重要な認証、アクセス制御、ビジネスロジック、および暗号鍵の管理フローに脅威モデルを使用する。
- ユニットテストおよび統合テストを実施し、すべての重要なフローが脅威モデルに対して耐性があることを検証する。アプリケーションの各階層のユースケースとミスユースケースをまとめる。
- ユーザーやサービスによるリソースの消費やアクセスを制限する。

Fastly のソリューション

- **Next-Gen WAF** ではカスタムシグナルを作成し、アプリケーションの狙われやすい経路やフロー周辺のアクティビティをモニタリングできます。例えば、決済フローに関連するアプリケーションが構築され、機密性が高い経路が存在する場合、アプリケーションのその部分に対するクライアントリクエストにカスタムシグナルをアタッチすることが可能です。カスタムシグナルにより、アクティビティをモニタリングし、リクエストの属性に基づき、悪意のある動作を行うと判断された場合にブロックまたはレート制限を行うルールを作成できます。

A05:2021

セキュリティの設定ミス

このカテゴリーは、アプリケーションにおいてセキュリティの堅牢化が不足している、またはディレクトリリスティングやデフォルトアカウントが有効になっている、セキュリティヘッダーが欠落しているなど、設定の一部が適切に行われていないことを意味します。また、前バージョンでは独立したカテゴリーが割り当てられていた XXE (XML External Entity) インジェクションも、このカテゴリーに含まれています。

攻撃シナリオの例

ディレクトリリスティングがサーバー上で無効になっていません。そのため、簡単にディレクトリを表示できることが攻撃者に見つかってしまいます。攻撃者はコンパイル済みの Java クラスを見つけてダウンロードし、デコンパイルしてからリバースエンジニアリングによってコードを確認します。これにより、アプリケーションに存在する深刻なアクセス制御の欠陥が攻撃者に気づかれてしまいます。

Fastly のソリューション

- Next-Gen WAF は、サーバーのレスポンスに必要なセキュリティヘッダーが含まれていない場合、管理者に警告します。これにより、OWASP が懸念する以下のアプリケーション脆弱性の問題に対応できます。
    - *サーバーによってセキュリティヘッダーやディレクティブが送信されなかったり、安全な値に設定されていなかったりする。*
- XXE インジェクション対策として、Next-Gen WAF は SQLi やコマンドインジェクションなどのインジェクション攻撃が XML ペイロードに存在しないか検査します。このような機能は、以下の OWASP の推奨事項に対応するのに役立ちます。
    - *ポジティブセキュリティモデル (許可リスト) に基づくサーバーサイドの入力検証やフィルタリング、サニタイズを実装し、XML ドキュメントやヘッダー、ノード内に潜む悪意のあるデータをブロックする。*
    - アプリケーションが HTTP リクエスト内の XML を受け入れるべきでない場合、Fastly によって XML ペイロードを含むリクエストを見つけて警告またはブロックできます。
- また、JSON ペイロードに存在する機密性の高いデータパターンを探してブロックまたはログに記録されるよう Next-Gen WAF を設定することも可能です。ブロックモードまたはログモードにかかわらず、機密性の高いデータパターンがペイロードに見つかると、Fastly はすぐに管理者に警告します。このような機能は、以下の OWASP の推奨事項に対応するのに役立ちます。
    - *可能な限り、JSON などの複雑でないデータ形式を使用し、機密性の高いデータのサニタイズを避ける。*
- アプリケーションが修正されるまで、新たなデータ漏えい問題に対処するのに仮想パッチの適用が役立ちます。
- さらに Fastly のエッジクラウドプラットフォームによって `Content-Security-Policy` や `Strict-Transport-Security` などのセキュリティヘッダーを追加することも可能です。エッジからクライアントにレスポンスヘッダーが返されるよう自動的に設定されるので、不適切に設定されたアプリケーションやオリジンサーバーに対して追加の保護レイヤーを提供できます。

対策

このカテゴリーの脅威を阻止するのは困難ですが、いくつかの簡単なプロセスを導入することで、このカテゴリーの脆弱性が攻撃者に悪用される前に脅威を検出してリスクを緩和できる可能性が高まります。その例として、デフォルトで堅牢化されている (「ゴールデンイメージ」など) apache や nginx の設定ファイルの使用など、反復可能な堅牢化プロセスが挙げられます。これにより、適切にロックダウンされた別の環境を素早く簡単にデプロイすることが可能になります。

開発や QA、本番環境にまったく同じ設定を適用し、それぞれの環境で異なる認証情報を使用するようにします。また、このプロセスを自動化してセキュアな環境を新たに設定する手間を最小限に抑えます。さらに、あらゆる環境において構成や設定が有効であることを検証する自動化されたプロセスも必要です。
これらの環境にとって、不要な機能やコンポーネント、ドキュメント、サンプルの無い最小限のプラットフォームが理想的です。

A06:2021

脆弱で古くなったコンポーネント

脆弱で古くなったソフトウェアコンポーネントには、既知のセキュリティ欠陥が存在する、またはセキュリティアップデートが期限切れのライブラリやサポートサービスが含まれます。特に大規模なレガシーコードベースを抱える大企業にとって、このようなコンポーネントを常にアップデートするのは、至難の業です。

攻撃シナリオの例

一般的にコンポーネントはアプリケーション自体と同じ権限で実行されるため、コンポーネントのいずれかに欠陥があると、深刻な影響が生じる可能性があります。そのような欠陥は偶発的なもの (コーディングのミスなど) もあれば、意図的なもの (コンポーネントに設置されたバックドアなど) もあります。以下は、コンポーネントにおける既知の悪用可能な脆弱性の例です。

- サーバー上で任意のコードの実行を可能にする、Struts 2 におけるリモートコード実行の脆弱性 (CVE-2017-5638) によって、重大なセキュリティ侵害がもたらされています。
- モノのインターネット (IoT) では、パッチの適用が困難または不可能なことが多いのですが、公開されたままの状態にある場合 (例 : NVR) や、重要な機能を提供する場合 (例 : バイオ医療機器) など、パッチ適用の重要性を無視できないことがあります。

対策

このカテゴリーの脆弱性によるリスクの緩和に役立つツールやプロダクトが存在するものの、このような脅威からの保護は、いまだにきわめて困難です。まずは、アプリケーションやポートフォリオのライフタイムを通じて、モニタリング、トリアージ、アップデートや設定変更の適用を継続的に計画することをお勧めします。これを実行するためのステップのひとつとして、Versions Maven Plugin や OWASP Dependency Check、Retire.js などのツールを使用して、クライアントおよびサーバの両方のコンポーネント（フレームワークやライブラリなど）とその依存関係のインベントリ取得を継続的に行うことが挙げられます。

また、コンポーネントの脆弱性について CVE (Common Vulnerability and Exposures) や NVD (National Vulnerability Database) などの情報ソースを継続的にモニタリングすることで、重要なバグのパッチに関する情報が得られます。さらに、ソフトウェア構成分析 (SCA) ツールによってプロセスを自動化し、必要な作業を軽減できます。使用しているコンポーネントに関するセキュリティ脆弱性のメールアラートに登録することで、CVE のモニタリングだけでは得られないコンテキストがもたらされ、環境をより把握しやすくなります。

Fastly のソリューション

- Next-Gen WAF は、複数の CVE (CVE-2023-34362 MOVEit SQL インジェクション脆弱性など) に対する仮想パッチを提供します。また、お客様のアプリケーション特有の脆弱性が見つかった場合、社内でパッチ/修正が開発されリリースされるまで、Fastly で仮想パッチを作成することが可能です。このような機能は、以下の OWASP の推奨事項に対応するのに役立ちます。
- *メンテナンスされていない、もしくはセキュリティパッチが作成されていない古いバージョンのライブラリとコンポーネントを監視する。パッチの適用が不可能な場合は、発見された問題を監視、検出したり、問題から保護したりするために仮想パッチの使用を検討する。*

A07:2021

識別と認証の失敗

識別と認証の失敗には、不適切なパスワード管理やクレデンシャルスタッフィング攻撃に対するレート制限の失敗、安全でないセッション管理、認証のバイパスなどが含まれます。

攻撃シナリオの例

クレデンシャルスタッフィングはこのカテゴリーの代表例です。一般的な攻撃のひとつに、よく使われるパスワードのリストやデータ侵害によって得たパスワードを使用して既知のパスワードを持つアカウントへのアクセスを試みることが挙げられます。自動化された脅威やクレデンシャルスタッフィングに対する保護対策が実装されていないアプリケーションは、認証情報が有効化どうかを判断する神託機械として利用される可能性があります。また、アプリケーションのパスワードやセッションポリシー、保護対策が適切に設計されていてセキュアであっても、それほど安全性が高くないアプリケーションから盗まれた認証情報を利用するクレデンシャルスタッフィング攻撃に対して脆弱である場合があります。

対策

阻止する方法は理論的にはかなりシンプルですが、実際に実装するのは容易ではありません。このカテゴリーでは細かい防御戦略が必要です。まずは以下を実践することをお勧めします。

- 可能な限り多要素認証を実装します。これにより、自動化されたクレデンシャルスタッフィング攻撃、ブルートフォース攻撃、盗まれた認証情報の再利用などを防ぐことができます。
- 弱いパスワードを設定していないか確認する機能を実装します。例えば、パスワードの新規設定や変更時には、「弱いパスワード トップ10,000」などのリストを使用して検証します。
- ログインの試行回数に制限を設けるか、またはログインの失敗が続く場合に徐々に処理を遅延させます。ただし、サービス拒否の機会を作らないように注意する必要があります。ログインの失敗をすべてログに記録し、クレデンシャルスタッフィングやブルートフォースなどの攻撃が検出された場合、管理者に警告します。

Fastly のソリューション

- **Next-Gen WAF** は、クレデンシャルスタッフィング攻撃などのブルートフォース攻撃を検出し、攻撃者によるログイン試行をブロックしてシステムを保護します。失敗したログイン試行はすべて記録され、攻撃が検出されるとすぐに管理者に警告します。これにより、以下の OWASP の推奨事項を実施できます。
    - *ログイン試行回数に制限を設けるか、ログインの失敗が続く場合に徐々に処理を遅延させる。ログインの失敗をすべてログに記録し、クレデンシャルスタッフィングやブルートフォースなどの攻撃が検出された場合、管理者に警告する。*
- 「セッション ID が URL に含まれないようにする」という OWASP が推奨する防止対策が実行されるように、または単にセッション ID がリクエスト URL に見つかった場合、管理者に警告するように **Next-Gen WAF** を設定できます。

A08:2021

ソフトウェアとデータの整合性の不具合

ソフトウェアとデータの整合性の不具合は、コードやインフラストラクチャが整合性違反から保護されていないことに関連しています。 例として、アプリケーションが信頼されていないソースに由来するプラグインやライブラリ、モジュール、レポジトリ、コンテンツ配信ネットワークに依存している場合が挙げられます (ua-parser-js の乗っ取りなど)。安全でない CI/CD パイプラインも、権限のないアクセスや悪意のあるコード、システム侵害の可能性を高めます (Kaseya のサプライチェーン攻撃など)。2017年にカテゴリーのひとつに加えられた「安全でないデシリアライゼーション」も、現在このカテゴリーに含まれています。

攻撃シナリオの例

SolarWinds での悪意のあるアップデート：最近注目を浴びた **SolarWinds の Orion に対する攻撃** からも分かるように、国家がアップデートメカニズムを攻撃する場合があることが知られています。Solarwinds は、安全なビルドとアップデートの整合プロセスを備えていました。にもかかわらず、それらは破壊され、同社は数か月にわたって高度に標的化された悪意のあるアップデートを1万8,000を超える組織に配信し、そのうち100組織ほどが影響を受けました。この一件は、この類の侵害としては史上最も影響が広範に及び、重大な被害をもたらした攻撃のひとつでした。

対策

- デジタル署名または類似のメカニズムを利用してソフトウェアやデータが意図されたソースから取得されたものであり、改ざんされていないことを検証します。
- npm や Maven など、ライブラリや依存関係が信頼されたリポジトリを使用していることを確認します。リスクの高いプロファイルがある場合は、社内で入念に検査された既知の信頼できるリポジトリをホストすることを検討します。
- CI/CD パイプラインが適切に分離されて設定され、アクセス制御が行われていること、また、ビルドやデプロイのプロセスに至るコードフローの整合性が確保されていることを確認します。

Fastly のソリューション

- **Next-Gen WAF** は、インジェクション攻撃のような攻撃のペイロードを含む、あらゆるリクエストを検査します。これにより、XSS や SQLi、ディレクトリトラバーサル、コマンド実行などの攻撃を含む、シリアライズされたオブジェクトのブロックが可能になります。

- また **Next-Gen WAF** は、単一のソースから送信された過度のデシリアライゼーションリクエストも検出できるので便利です。しきい値を超えるとすぐに管理者に通知し、レート制限によって後続のデシリアライゼーションリクエストの一時的なブロックを有効にできます。このような機能は、以下の OWASP の推奨事項に対応するのに役立ちます。
  - *デシリアライゼーションをモニタリングし、ユーザーが何度もデシリアライズを行う場合、管理者に警告する。*

- さらに、リクエストのペイロードに、Base64 でエンコードされ、シリアライズされたペイロードが含まれているか検出するように **Next-Gen WAF** を設定できます。以下を含む他のシリアライズパターンを特定して検出することも可能です。
  - Base64 で使用される「rO0」
  - Content-type = 'application/x-java-serialized-object'

- Base64 でエンコードされ、シリアライズされたペイロードや、特定のコンテンツタイプがリクエストに含まれるべきでない場合、そのようなリクエストを簡単にブロックできます。また、Base64 でエンコードされたシリアライゼーションリクエストをデコードし、拒否リストに含まれる Java クラスが存在しないか検査することも可能です。これにより、先を見越したクラスの検証を行い、以下の OWASP の推奨事項に対応できます。
  - *通常コードは定義可能なクラスに基づくため、オブジェクトを生成する前に、デシリアライゼーションにおいて厳密な型制約を強制する。*

- 不正な形式のデシリアライゼーションリクエスト - **Next-Gen WAF** は、デフォルトで XML や JSON のペイロードを解析し、ペイロードに不正な形式が使用されている場合、そのリクエストをフラグするので、そのためのルール作成は不要です。
- さらに、デシリアライゼーションの新たな欠陥が見つかった場合、アプリケーションが修正されるまで仮想パッチを適用することで対応できます。

A09:2021

セキュリティログとモニタリングの失敗

このカテゴリーは、アクティブな侵害の検出、エスカレーション、および対応を促進することを目的としています。ログへの記録とモニタリングがなければ、侵害を検知することはできません。このカテゴリーで失敗が起きると、可視性、インシデントアラート、フォレンジックなどに直接影響します。

攻撃シナリオの例

ある児童医療プランのプロバイダーは、攻撃者が350万人以上の子どもたちの数千もの機密性の高い健康記録にアクセスし、改ざんしたという連絡を、外部の関係者から受けました。インシデント後のレビューでは、医療プランプロバイダーのWebサイト開発者が重要な脆弱性に対処していなかったことが判明しました。システムのログ記録やモニタリングが行われていなかったため、データ侵害は2013年から7年以上にわたって進行していた可能性があります。

対策

アプリケーションのリスクに応じて、以下のコントロールを開発者が実装する必要があります。

- ログイン、アクセス制御の失敗、サーバーサイドの入力検証の失敗をすべてログに記録するようにします。不審なアカウントや悪意のあるアカウントを特定するのに必要なユーザコンテキストがログに含まれるようにし、後日フォレンジック分析を行えるよう十分な期間、保持されるようにします。
- 価値の高いトランザクションに関して監査証跡を取得するようにし、その際、追記型データベースのテーブルなど、完全性を確保できるコントロール方法を用いて、改ざんや削除を防止します。
- NIST 800-61rev2（またはそれ以降）のような、インシデント対応および復旧計画を策定または採用します。

Fastly のソリューション

- **Next-Gen WAF** は、受信したすべての HTTP リクエストとレスポンスをモニタリングし、攻撃や異常の有無を確認します。これには、成功と失敗の両方を含むあらゆるログイン試行のモニタリングとログへの記録が含まれます。自動化されたログ機能で記録されたり、ブロックの判断が下されたりした場合、メールまたはウェブフックの統合経由で、設定されたチャンネルを通じて管理者にアラートが自動的に送信されます。また API 経由で、Next-Gen WAF のすべてのイベントとリクエストデータを任意のシステムにインポートし、JSON 形式のログアーカイブとして長期間保存できます。このような機能は、以下の OWASP の推奨事項に対応するのに役立ちます。
  - *ログイン、アクセス制御の失敗、サーバーサイドの入力検証の失敗をすべてログに記録するようにする。不審なアカウントや悪意のあるアカウントを特定するのに必要なユーザコンテキストがログに含まれるようにし、後日フォレンジック分析を行えるよう十分な期間、保持されるようにする。*
  - *統合ログ管理ソリューションで簡単に使用できる形式でログが生成されていることを確認する。*
  - *価値の高いトランザクションに関して監査証跡を取得するようにし、その際、追記型データベースのテーブルなど、完全性を確保できるコントロール方法を用いて、改ざんや削除を防止する。*
  - *疑わしいアクティビティをすぐに検出して対応できるよう、効果的なモニタリングとアラートのしくみを確立する。*
- **Next-Gen WAF** は、ダッシュボードで設定されたデフォルトのウェブフック統合経由でリアルタイムのイベント通知を送信します。通知のチャンネルには、ウェブフックをサポートするあらゆる通知受信ソリューションに加えて、Slack、PagerDuty、Datadog が含まれます。
- SIEM との統合 (Elastic や ELK スタック、Sumo Logic など) はシンプルで、セキュリティ関連のイベントや詳細情報を必要な期間保存することができます。
- Fastly の Managed Security Service では、Fastly の Customer Security Operations Center (CSOC) が24時間365日体制でプロアクティブなモニタリングと脅威の緩和を行います。

A10:2021

サーバーサイド・リクエスト・フォージェリ

SSRF の欠陥は、Web アプリケーションがリモートのリソースを取得する際に、ユーザーから提供された URL を検証しないことで発生します。ファイアウォールや VPN あるいはその他の種類のネットワークアクセス制御リスト (ACL) によってアプリケーションが保護されている場合でも、攻撃者は SSRF により、アプリケーションが意図しない宛先へ細工されたリクエストを送信するよう強制することができます。SSRF は攻撃目的で使用される一種の邪悪なリバースプロキシを可能にするものとして捉えることができます。

最先端の Web アプリケーションは、エンドユーザーに便利な機能 (複数のバックエンドシステムと通信するなど) を提供するようになり、アプリケーション側で URL を取得することも珍しくなくなりました。しかしその結果、SSRF の発生が増加しています。また、クラウドサービスやアーキテクチャの複雑性を背景に、 SSRF の深刻度も徐々に増しています。

攻撃シナリオの例

クラウドサービスのメタデータストレージへのアクセス - 大半のクラウドプロバイダーは http://169.254.169.254/ のようなメタデータストレージを提供しています。攻撃者はこのメタデータを読み取り、機密性の高い情報を取得できます。

対策

SSRF 攻撃の攻撃対象領域は複雑なため、多層防御の実装がこの種類の攻撃に対する防御の最善策です。

ネットワークレイヤーにて、リモートのリソースにアクセスする機能を分離されたネットワークにセグメント化することで、SSRF の影響を軽減できます。

アプリケーションレイヤーでは、クライアントが提供したすべての入力データのサニタイズと検証、HTTP リダイレクトの無効化、明確な許可リストの使用による URL スキーム、ポート、宛先の強制などの対策を実行できます。
SSRF への対策に、拒否リストや正規表現を使用しないようにしてください。攻撃者は、拒否リストを回避するためのペイロードのリストやツール、技術を利用できるためです。

Fastly のソリューション

- **Next-Gen WAF :** NLX と Fastly のセキュリティリサーチチームを通じて、SSRF をはじめ、新たに拡散する脅威に対する保護対策を迅速にロールアウトします。その例として Next-Gen WAF には、AWS を狙った SSRF に対処するテンプレートルールが含まれています。
- また **Next-Gen WAF** では、無効なホストヘッダー (ドメインの許可リストにないものなど) を含むリクエストをすべてブロックするカスタムルールを追加し、ホストヘッダーを介した SSRF を阻止できます。

まとめ

Next-Gen WAF が提供するデフォルトの検出機能とカスタマイズ可能な機能を組み合わせ、さらに Fastly のエッジクラウドプラットフォームの機能を併用することで、Fastly のお客様は OWASP トップ10の脅威に対する保護対策を強化できます。OWASP と Fastly が推奨するベストプラクティスを実践し、保護領域を拡大することによって、アプリケーションセキュリティのスキルと能力を高め、お客様と企業のデータをより効果的に保護することが可能になります。

Fastly によって OWASP トップ10の脅威に対する保護を強化する方法の詳細については、**Fastly までお問い合わせ**ください。

参考リソース

- **OWASP.org**
- **Next-Gen WAF の検出・ブロック機能**
- **Fastly Next-Gen WAF の主な10機能**