実装ガイド

AWS WAF のセキュリティオートメーション

AWS WAF のセキュリティオートメーション: 実装ガイド

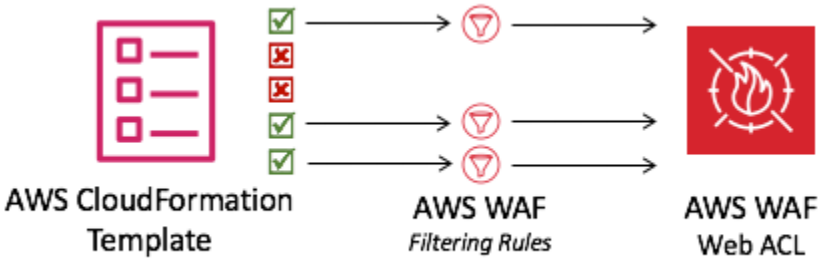
Table of Contents

AWS WAF で Security Automations を使用してウェブベースの攻撃をフィルタリングする単一のウェブアクセスコントロールリストを自動的にデプロイする

AWS WAF のセキュリティオートメーションソリューションでは、事前設定されたルールのセットをデプロイして、一般的なウェブエクスプロイトからアプリケーションを保護します。このソリューションのコアサービスである AWS WAF は、アプリケーションの可用性、セキュリティの侵害、リソースの過剰消費に影響を及ぼす可能性のある攻撃手法からウェブアプリケーションを保護するのに役立ちます。AWS WAF を使用して、カスタマイズ可能なウェブセキュリティルールを定義できます。これらのルールは、Amazon CloudFront、Application Load Balancer (ALB) などの AWS リソースにデプロイされたウェブアプリケーションとアプリケーションプログラミングインターフェイス (API) に対して、どのトラフィックを許可またはブロックするかを制御します。サポートされているその他のリソースタイプについては、「AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide」の「AWS WAF」を参照してください。

AWS WAF ルールの設定は、組織の大小に関わらず、特に専任のセキュリティチームを持たない組織にとって困難で負担が大きい場合があります。このプロセスをシンプルにするために、AWS WAF のセキュリティオートメーションソリューションでは、一般的なウェブベースの攻撃をフィルタリングするよう設計された AWS WAF ルールを使用して、単一のウェブアクセスコントロールリスト (ACL) を自動的にデプロイします。このソリューションの AWS CloudFormation テンプレートの初期設定時に、どの保護機能を含めるか指定できます。このソリューションをデプロイすると、AWS WAF は既存の CloudFront ディストリビューションまたは ALB へのウェブリクエストを検査し、該当する場合はそれらをブロックします。

CloudFormation テンプレートは、AWS WAF フィルタリングルールを使用してウェブ ACL をデプロイします。



この実装ガイドでは、Amazon Web Services (AWS) クラウドにこのソリューションをデプロイする際のアーキテクチャ上の考慮事項、設定手順、および運用上のベストプラクティスについて説明しま

す。セキュリティと可用性に関する AWS のベストプラクティスを使用して、このソリューションを AWS にデプロイするために必要な AWS セキュリティ、コンピューティング、ストレージ、およびその他のサービスを起動、設定、実行する CloudFormation テンプレートへのリンクが含まれています。

このガイドに記載されている情報は、AWS WAF、CloudFront、ALB、AWS Lambda などの AWS のサービスに関する実務知識があることを前提としています。また、一般的なウェブベースの攻撃と緩和戦略に関する基本的な知識も必要です。

> **Note**
>
> バージョン 3.0.0 以降、このソリューションは最新バージョンの AWS WAF サービス API (AWS WAFV2) をサポートしています。

このガイドは、IT マネージャー、セキュリティエンジニア、DevOps エンジニア、開発者、ソリューションアーキテクト、ウェブサイト管理者を対象としています。

> **Note**
>
> AWS WAF ルールを実装するための出発点として、このソリューションを使用することをお勧めします。ソースコードをカスタマイズしたり、新しいカスタムルールを追加したり、必要に応じてより多くの AWS WAF マネージドルールを活用したりできます。

このナビゲーションテーブルを使用すると、次の質問に対する回答をすばやく見つけることができます。

質問内容	参照先
このソリューションの実行に必要なコストを確認する。このソリューションを実行するための合計コストは、アクティブ化された保護、取り込み、保存、および処理されるデータの量によって異なります。	コスト
このソリューションのセキュリティ上の考慮事項を理解する。	セキュリティ

質問内容	参照先
どの AWS リージョンでこのソリューションをサポートしているか知りたい場合。	サポートしている AWS リージョン
このソリューションに含まれている CloudFormation テンプレートを表示またはダウンロードして、このソリューションのインフラストラクチャリソース (スタック) を自動的にデプロイしたい場合。	AWS CloudFormation テンプレート
ソリューションのデプロイ、使用、トラブルシューティングについて、サポートを使用します。	サポート
ソースコードにアクセスし、オプションで AWS Cloud Development Kit (AWS CDK) を使用してソリューションをデプロイします。	GitHub リポジトリ

機能とメリット

AWS WAF のセキュリティオートメーションソリューションには、次の機能と利点があります。

AWS マネージドルールのルールグループを使用してウェブアプリケーションを保護する

AWS WAF 用の AWS マネージドルールは、一般的なアプリケーションの脆弱性やその他の望ましくないトラフィックからの保護を提供します。このソリューションには、AWS マネージド IP 評価ルールグループ、AWS マネージドベースラインルールグループ、AWS マネージドユースケース固有のルールグループが含まれます。ウェブ ACL のキャパシティユニット (WCU) クォータの上限まで、ウェブ ACL に 1 つまたは複数のルールグループを選択するオプションがあります。

事前定義された HTTP Flood カスタムルールを使用してレイヤー 7 フラッド保護を提供する

HTTP Flood カスタムルールは、お客様が定義した期間、ウェブレイヤー分散型サービス妨害 (DDoS) 攻撃から保護します。このルールを有効にするには、次のいずれかのオプションを選択できます。

- AWS WAF レートベースのルール
- Lambda ログパーサー
- Amazon Athena ログパーサー

Lambda ログパーサーまたは Athena ログパーサーオプションを使用すると、100 未満のリクエストクォータを定義できます。このアプローチは、AWS WAF レートベースのルールに必要なクォータに達しないようにするのに役立ちます。詳細については、「ログパーサーオプション」を参照してください。

フィルタリング条件に国と Uniform Resource Identifier (URI) を追加することで、Athena ログパーサーを強化することもできます。このアプローチは、予測不可能な URI パターンを持つ HTTP フラッド攻撃を特定してブロックします。詳細については、「HTTP フラッド Athena ログパーサーで国と URI を使用する」を参照してください。

事前定義された Scanners & Probes カスタムルールを使用して脆弱性の悪用をブロックする

スキャナー & プローブ のカスタムルールは、アプリケーションアクセスログを解析して、オリジンによって生成された異常な量のエラーなどの疑わしい動作を検索します。その後、疑わしい送信元 IP アドレスは、お客様が定義した期間、ブロックされます。このルールを有効にするには、Lambda ログパーサーまたは Athena ログパーサーのいずれかのオプションを選択できます。詳細については、「ログパーサーオプション」を参照してください。

定義済みの Bad Bot カスタムルールで侵入を検出して回避する

Bad Bot のカスタムルールは、試みられた攻撃を誘き寄せることを目的としたセキュリティメカニズムであるハニーポットエンドポイントを設定します。エンドポイントをウェブサイトに挿入して、コンテンツスクレイパーや悪質なボットからのインバウンドリクエストを検出できます。検出されると、同じオリジンからの後続のリクエストはブロックされます。詳細については、「ウェブアプリケーションにハニーポットリンクを埋め込む」を参照してください。

事前定義された IP 評価リストのカスタムルールを使用して悪意のある IP アドレスをブロックする

IP 評価リストカスタムルールは、ブロックする新しい IP 範囲について、サードパーティー IP 評価リストを 1 時間ごとにチェックします。これらのリストには、Spamhaus Don't Route Or Peer (DROP) および Extended DROP (EDROP) リスト、Proofpoint Emerging Threats IP リスト、Tor exit node リストが含まれます。

許可および拒否された IP リストのカスタムルールが事前定義された手動 IP 設定を提供する

許可および拒否された IP リストのカスタムルールを使用すると、許可または拒否する IP アドレスを手動で挿入できます。また、許可および拒否された IP リストで IP 保持を設定し、設定した時間に IP を期限切れにすることもできます。

独自のモニタリングダッシュボードを構築する

このソリューションは、許可されたリクエスト、ブロックされたリクエスト、その他の関連メトリクスなどの Amazon CloudWatch メトリクスを出力します。カスタマイズされたダッシュボードを構築して、これらのメトリクスを視覚化し、AWS WAF が提供する攻撃と保護のパターンに関するインサイトを得ることができます。詳細については、「モニタリングダッシュボードを構築する」を参照してください。

ユースケース

このソリューションのユースケースの例を次に示します。このソリューションは、このリストに示している方法以外にもさまざまな革新的な方法でカスタマイズできます。

AWS WAF ルールのセットアップを自動化する

AWS WAF は一般的な攻撃からウェブアプリケーションを保護しますが、AWS WAF ルールの設定は複雑で時間がかかる場合があります。このソリューションでは、CloudFormation テンプレートを使用して、一連の AWS WAF ルールをアカウントに自動的にデプロイします。これにより、AWS WAF ルールを自分で設定する必要がなくなり、AWS WAF をより迅速に開始できます。

レイヤー 7 の HTTP フラッド保護をカスタマイズする

このソリューションには、HTTP Flood 保護をアクティブ化するための 3 つのオプションがあります。DDoS 攻撃に対する保護を得るために、ニーズに合ったオプションを選択できます。詳細につ

いては、「機能と利点」の「事前定義された HTTP Flood カスタムルールを使用してレイヤー 7 フラッド保護を提供する」を参照してください。

カスタマイズの適用や独自のセキュリティオートメーションの構築にソースコードを活用する

このソリューションは、AWS WAF やその他のサービスを使用して AWS クラウドでセキュリティオートメーションを構築する方法の例を示します。GitHub のオープンソースコードは、カスタマイズを適用したり、ニーズに合った独自のセキュリティオートメーションを構築したりするのに便利です。

概念と定義

このセクションでは、重要な概念について説明し、このソリューションに固有の用語を定義します。

ALB ログ

このソリューションでは、ALB リソースのログを使用します。このソリューションのスキャナーとプローブの保護ルールは、これらのログを検査します。

Athena ログパーサー

Amazon Athena は、オープンソースフレームワーク上に構築されたサーバーレスのインタラクティブな分析サービスで、オープンテーブル形式とファイル形式をサポートしています。このソリューションは、ユーザーが `yes - Amazon Athena log parser` を選択して HTTP フラッド保護ルールまたはスキャナーとプローブ保護ルールをアクティブ化した場合、スケジュールされた Athena クエリを実行して AWS WAF、CloudFront、または ALB ログを検査します。また、構造化されたロジックチェーン経由で動作する検出により、不正なボットからの保護のアクティブ化にも使用できます。

AWS WAF ルール

AWS WAF ルールは以下を定義します:

- HTTP(S) ウェブリクエストを検査する方法
- 検査基準に一致する場合にリクエストに対して実行するアクション

ルールは、ルールグループまたはウェブ ACL のコンテキストでのみ定義されます。

CloudFront ログ

このソリューションは、CloudFront リソースのログを使用します。このソリューションの Scanner & Probe Protection ルールは、これらのログを検査します。

IP セット

IP セットは、ルールステートメントで一緒に使用する IP アドレスと IP アドレス範囲の

コレクションを提供します。IP セットは AWS リソースです。

Lambda ログパーサー

このソリューションは、Amazon Simple Storage Service (Amazon S3) オブジェクト作成イベントによって呼び出される Lambda 関数を実行します。Lambda 関数は、ユーザーが `yes - AWS Lambda log parser` を選択して HTTP フラッド保護、スキャナーとプローブ保護をアクティブ化した場合、AWS WAF、CloudFront、または ALB の各ログの検査を開始します。構造化されたロジックチェーン経由で動作する検出により、不正なボットからの保護ルールにも使用できます。

マネージドルールグループ

マネージドルールグループは、AWS と AWS Marketplace の販売者がお客様に変わって作成および管理する、すぐに使用可能な事前定義済みのルールの集まりです。AWS WAF 料金は、すべてのマネージドルールグループの使用に適用されます。

リソース/エンドポイントのタイプ

AWS リソースをウェブ ACL に関連付けて保護することができます。これらのリソースは、CloudFront、ALB、AWS AppSync、Amazon Cognito、AWS App Runner、AWS Verified Access の各リソースです。現在、このソリューション Amazon は CloudFront と ALB をサポートしています。

WAF ログ

このソリューションは、ウェブ ACL に関連付けられたリソースに対して AWS WAF によって生成されたログを使用します。このソリューションの HTTP フラッド保護、スキャナーとプローブ保護、不正なボットからの保護のアクティブ化の各ルールは、これらのログを検査します。

WCU

AWS WAF はウェブアクセスコントロールリスト (ACL) キャパシティユニット (WCU) を使用して、ルール、ルールグループ、ウェブ ACL の実行に必要な運用リソースを計算およびコントロールします。AWS WAF は、ルールグループとウェブ ACL を設定するときに WCU クォータを適用します。WCU は、AWS WAF によるウェブトラフィックの検査方法には影響しません。

ウェブ ACL

ウェブ ACL を使用すると、保護されたリソースが応答する HTTP(S) ウェブリクエストをきめ細かく制御できます。

> **Note**
>
> AWS 用語の一般的なリファレンスについては、「AWS 用語集」を参照してください。

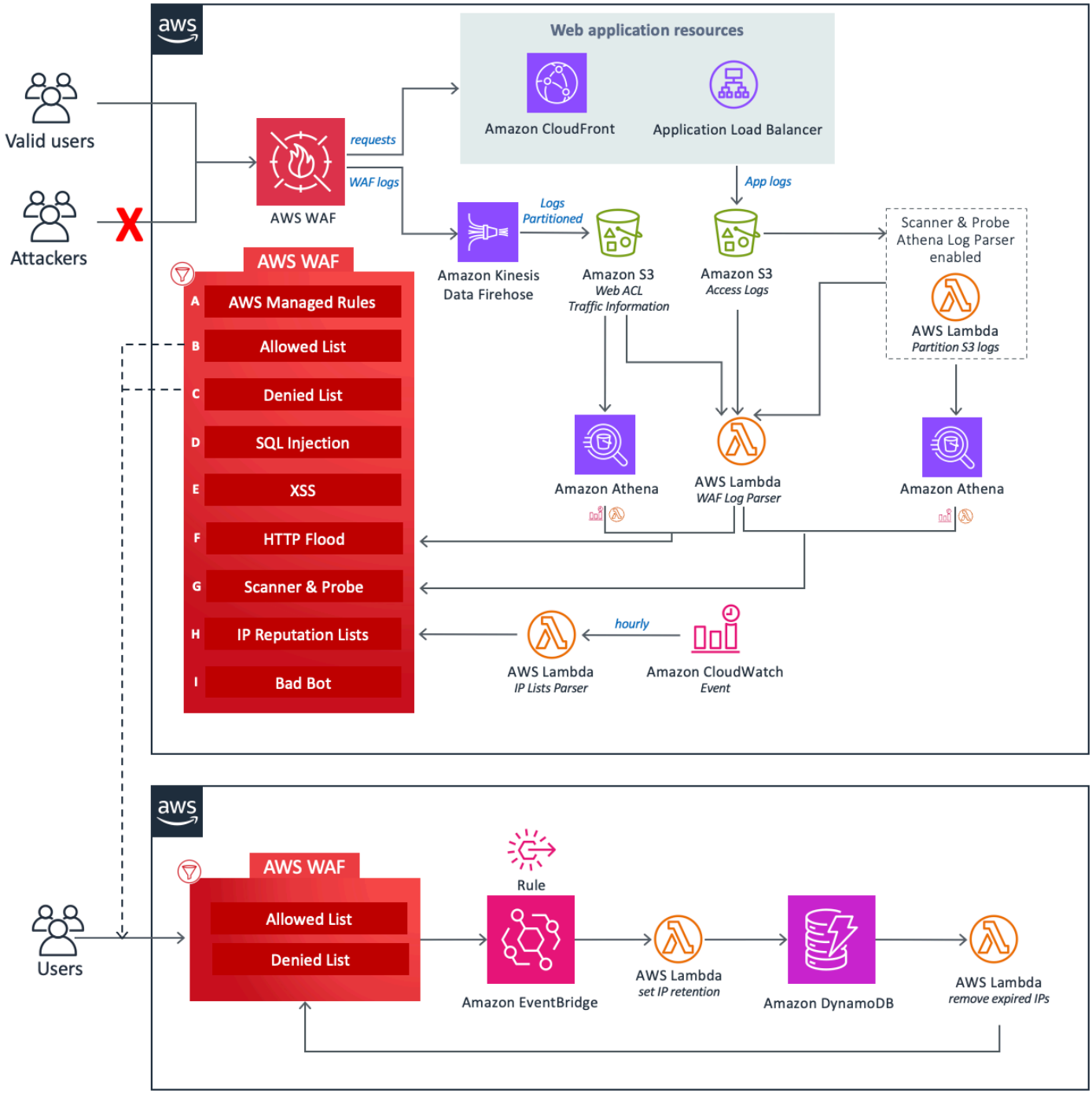
アーキテクチャの概要

このセクションでは、このソリューションで導入されるコンポーネントのリファレンス実装のアーキテクチャ図を示します。

アーキテクチャ図

このソリューションをデフォルトのパラメータを使用してデプロイすると、AWS アカウントに次のコンポーネントがデプロイされます。

CloudFormation テンプレートは、AWS WAF およびその他の AWS リソースをデプロイして、一般的な攻撃からウェブアプリケーションを保護します。



設計の中核となるのは AWS WAF ウェブ ACL です。このウェブ ACL は、ウェブアプリケーションへのすべての受信リクエストの一元的な検査と判断ポイントとして機能します。CloudFormation スタックの初期設定時に、ユーザーはどの保護コンポーネントをアクティブにするのかを定義します。各コンポーネントは独立して動作し、ウェブ ACL に異なるルールを追加します。

このソリューションのコンポーネントは、次の保護領域に分類できます。

> **Note**
>
> グループのラベルは WAF ルールの優先度レベルを反映していません。

- AWS マネージドルール (A) – このコンポーネントには、AWS マネージドルールの IP 評価ルールグループ、ベースラインルールグループ、およびユースケース固有のルールグループが含まれます。これらのルールは独自のルールを作成しなくても、OWASP で公開されているものを含め、一般的なアプリケーションの脆弱性やその他の望ましくないトラフィックが悪用されるのを防ぐことができます。
- 手動 IP リスト (B および C) – これらのコンポーネントは 2 つの AWS WAF ルールを作成します。これらのルールを使用すると、許可または拒否する IP アドレスを手動で挿入できます。Amazon EventBridge ルールと Amazon DynamoDB を使用して、許可または拒否された IP セットの IP 保持を設定し、期限切れの IP アドレスを削除できます。詳細については、「許可および拒否された AWS WAF IP セットの IP 保持を設定する」を参照してください。
- SQL インジェクション (D) および XSS (E) – これらコンポーネントにより、URI、クエリ文字列、リクエストボディ内の一般的な SQL インジェクションやクロスサイトスクリプティング (XSS) パターンから保護するよう設計された 2 つの AWS WAF ルールが設定されます。
- HTTP フラッド (F) – このコンポーネントは、アプリケーションレイヤーの DDoS 攻撃や総当たりのログインの試行など、特定の IP アドレスから大量の要求を行う攻撃から保護します。このルールでは、デフォルトの 5 分間に 1 つの IP アドレスから許可される受信リクエストの最大数を定義するクォータを設定します (Athena Query Run Time Schedule パラメータで設定可能)。このしきい値を超えると、IP アドレスからの追加のリクエストは一時的にブロックされます。このルールは、AWS WAF レートベースのルールを使用するか、Lambda 関数または Athena クエリを使用して AWS WAF ログを処理することで実装できます。HTTP フラッドの緩和策オプションに関連するトレードオフの詳細については、「ログパーサーオプション」を参照してください。
- スキャナーとプローブ (G) – このコンポーネントは、アプリケーションアクセスログを解析して、オリジンによって生成された異常な量のエラーなどの疑わしい動作を検索します。その後、疑わしい送信元 IP アドレスは、お客様が定義した期間、ブロックされます。このルールは、Lambda 関数または Athena クエリを使用して実装できます。スキャナーとプローブの緩和策オプションに関連するトレードオフの詳細については、「ログパーサーオプション」を参照してください。
- IP 評価リスト (H) – このコンポーネントは、ブロックする新しい範囲について、毎時サードパーティーの IP 評価リストをチェックする `IP Lists Parser` Lambda 関数です。これらのリストには、Spamhaus Don't Route Or Peer (DROP) および Extended DROP (EDROP) リスト、Proofpoint Emerging Threats IP リスト、Tor exit node リストが含まれます。

- 不正なボット (I) - このコンポーネントは、ハニーポットメカニズムに加えて、Application Load Balancer (ALB) または Amazon CloudFront への直接接続をモニタリングすることで、不正なボットの検出を強化します。ボットがハニーポットをバイパスして ALB または CloudFront とやり取りしようとすると、システムはリクエストパターンとログを分析して悪意のあるアクティビティを特定します。不正なボットを検出すると、その IP アドレスを抽出して AWS WAF ブロックリストに追加し、以降のアクセスを防ぎます。不正なボット検出は、構造化されたロジックチェーンを介して動作し、脅威に対して包括的に対応します。
  - HTTP フラッド保護 Lambda ログパーサー – フラッド分析中にログエントリから不正なボットの IP を収集します。
  - スキャナーとプローブ保護 Lambda ログパーサー – スキャナー関連のログエントリから不正なボットの IP を特定します。
  - HTTP フラッド保護 Athena ログパーサー – クエリ実行全体でパーティションを使用して、Athena ログから不正なボットの IP を抽出します。
  - スキャナーとプローブ保護 Athena ログパーサー – 同じパーティショニング戦略を使用して、スキャナー関連の Athena ログから不正なボットの IP を取得します。
  - フォールバック検出 - HTTP フラッド保護およびスキャナーとプローブ保護の両方が無効になっている場合、システムは Log Lambda パーサーにより、WAF ラベルフィルターに基づいてボットアクティビティをログに記録します。

このソリューションの 3 つのカスタム Lambda 関数はそれぞれ、ランタイムメトリクスを CloudWatch に公開します。これらの Lambda 関数の詳細については、「コンポーネントの詳細」を参照してください。

AWS Well-Architected の設計に関する考慮事項

このソリューションでは、AWS Well-Architected フレームワークのベストプラクティスを使用しています。これにより、お客様は信頼性が高く、安全で、効率的で、コスト効率の高いワークロードをクラウド上で設計し運用することができます。

このセクションでは、Well-Architected Framework の設計原則とベストプラクティスがこのソリューションにどのように役立つかについて説明します。

オペレーショナルエクセレンス

このセクションでは、オペレーショナルエクセレンスの柱に関する原則とベストプラクティスを用いてこのソリューションをどのように設計したかを説明します。

- このソリューションは、メトリクスを CloudWatch にプッシュして、インフラストラクチャ、Lambda 関数、Amazon Data Firehose、Amazon S3 バケット、その他のソリューションコンポーネントへのオブザーバビリティを提供します。
- 当社では、AWS 継続的インテグレーションおよび継続的デリバリー (CI/CD) パイプラインを通じてソリューションを開発、テスト、公開します。これにより、デベロッパーは一貫して高品質の結果を達成できます。
- アカウントに必要なすべてのリソースをプロビジョニングする CloudFormation テンプレートを使用してソリューションをインストールできます。ソリューションを更新または削除するには、テンプレートを更新または削除するだけで済みます。

セキュリティ

このセクションでは、このソリューションを設計する際に、セキュリティの柱の原則とベストプラクティスをどのように適用したかについて説明します。

- すべてのサービス間通信は、AWS Identity and Access Management (IAM) ロールを使用します。
- このソリューションで使用されるすべてのロールは、最小権限の原則に従います。つまり、サービスが正しく機能するために必要な最小限のアクセス許可のみが含まれます。
- Amazon S3 バケットと DynamoDB を含むすべてのデータストレージは保管時に暗号化されます。

信頼性

このセクションでは、信頼性の柱に関する原則とベストプラクティスを用いてこのソリューションをどのように設計したかを説明します。

- このソリューションは、AWS のサーバーレスサービス (Lambda、Firehose、Amazon S3、Athena など) をできるだけ使用し、高可用性の維持とサービス障害からの復旧を確保します。
- ソリューションで自動化されたテストを実行して、エラーを神速に検出して修正します。
- このソリューションでは、データ処理に Lambda 関数を使用します。このソリューションはデータを Amazon S3 と DynamoDB に保存しており、デフォルトで複数のアベイラビリティーゾーンに保持します。

パフォーマンス効率

このセクションでは、パフォーマンス効率の柱に関する原則とベストプラクティスを用いてこのソリューションをどのように設計したかを説明します。

- このソリューションでは、サーバーレスアーキテクチャを使用して、高いスケーラビリティと可用性を低コストで実現します。
- このソリューションでは、データをパーティション分割し、クエリを最適化してデータスキャンの量を減らし、より高速な結果を実現することで、データベースのパフォーマンスを向上させます。
- このソリューションは毎日自動的にテストされて、デプロイされます。ソリューションアーキテクトと対象分野の専門家が、実験と改善が必要な分野についてこのソリューションをレビューします。

コスト最適化

このセクションでは、このソリューションを設計する際に、コスト最適化の柱の原則とベストプラクティスをどのように適用したかを説明します。

- このソリューションはサーバーレスアーキテクチャを使用しており、ユーザーは使用した分のみを支払います。
- コンピューティングレイヤーのデフォルトは Lambda で、従量課金制モデルを使用しています。
- Athena データベースとクエリは、データスキャンの量を減らすように最適化されているため、コストが削減されます。

持続可能性

このセクションでは、このソリューションを設計する際に、持続可能性の柱の原則とベストプラクティスをどのように適用したかを説明します。

- このソリューションは、マネージドサービスとサーバーレスサービスを使用して、バックエンドサービスの環境への影響を最小限に抑えます。
- このソリューションのサーバーレス設計は、継続的に運用されているオンプレミスサーバーのフットプリントと比較して、二酸化炭素排出量を削減することを目的としています。

アーキテクチャの詳細

このセクションでは、このソリューションを構成するコンポーネントと AWS のサービス、およびこれらのコンポーネントがどのように連携するのかについてのアーキテクチャの詳細について説明します。

このソリューションで使用している AWS のサービス

AWS のサービス	説明
AWS WAF	コア。AWS WAF ウェブ ACL、AWS マネージドルールのルールグループ、カスタムルール、および IP セットをデプロイします。AWS WAF API コールを行って、一般的な攻撃をブロックし、ウェブアプリケーションを保護します。
Amazon Data Firehose	コア。Amazon S3 バケットに AWS WAF ログを配信します。
Amazon S3	コア。AWS WAF ログ、CloudFront ログ、ALB ログを保存します。
* AWS Lambda*	コア。カスタムルールをサポートするために複数の Lambda 関数をデプロイします。
Amazon EventBridge ()	コア。Lambda を呼び出すイベントルールを作成します。
Amazon Athena	サポート。Athena ログパーサーをサポートする Athena クエリとワークグループを作成します。
AWS Glue	サポート。Athena ログパーサーをサポートするデータベースとテーブルを作成します。
Amazon SNS	サポート。Amazon Simple Notification Service (Amazon SNS) の E メール通知を送信して、許

AWS のサービス	説明
	可リストと拒否リストの IP 保持をサポートします。
AWS Systems Manager	サポート。アプリケーションレベルのリソースの監視と、リソースの操作とコストデータの可視化を提供します。

ログパーサーオプション

「アーキテクチャの概要」で説明した通り、HTTP フラッドとスキャナーとプローブの保護を処理するには 3 つのオプションがあります。以下のセクションでは、これらの各オプションについて詳しく説明します。

AWS WAF レートベースのルール

レートベースのルールは、HTTP フラッド保護に使用できます。デフォルトでは、レートベースのルールはリクエスト IP アドレスに基づき、リクエストを集約してレート制限します。このソリューションを使用すると、クライアント IP が連続的に更新される 5 分の間に許可するウェブリクエストの数を指定できます。IP アドレスが設定されたクォータに違反した場合、AWS WAF は、リクエストレートが設定されたクォータを下回るまで、ブロックされた新しいリクエストをブロックします。

リクエストクォータが 5 分あたり 2,000 リクエストを超え、カスタマイズを実装する必要がない場合は、レートベースのルールオプションを選択することをおすすめします。例えば、リクエストのカウント時に静的リソースアクセスを考慮しません。

さらに、他のさまざまな集約キーやキーの組み合わせを使用するようにルールを設定できます。詳細については、「集約オプションとキー」を参照してください。

Amazon Athena ログパーサー

HTTP Flood Protection と Scanner & Probe Protection の両方のテンプレートパラメータには、Athena ログパーサーオプションが用意されています。アクティブにすると、CloudFormation は Athena を実行し、結果出力を処理し、AWS WAF を更新するオーケストレーションを担当する Athena クエリとスケジュールされた Lambda 関数をプロビジョニングします。この Lambda 関数は、5 分ごとに実行されるように設定された CloudWatch イベントによって呼び出されます。これは、Athena Query Run Time Schedule パラメータで設定できます。

AWS WAF レートベースのルールを使用できず、カスタマイズを実装するために SQL に精通している場合は、このオプションを選択することをお勧めします。デフォルトのクエリを変更する方法については、「Amazon Athena クエリを表示する」を参照してください。

HTTP フラッド保護は、AWS WAF アクセスログ処理に基づいており、WAF ログファイルを使用します。AWS WAF アクセスログのタイプは遅延時間が短いため、CloudFront や ALB のログ配信時間と比較して、HTTP フラッドのオリジンをより迅速に特定できます。ただし、レスポンスステータスコードを受信するには、Activate Scanner & Probe Protection テンプレートパラメータで CloudFront や ALB のログタイプを選択する必要があります。

> **Note**
>
> 不正なボットがハニーポットをバイパスして ALB または CloudFront と直接やり取りする場合、システムはログ分析を通じて悪意のある動作を検出します (ただし、HTTP フラッド保護およびスキャナーとプローブ保護の両方が Lambda ログパーサーを使用していない場合を除きます)。

AWS Lambda ログパーサー

HTTP Flood Protection と Scanner & Probe Protection のテンプレートパラメータには、AWS Lambda ログパーサーオプションが用意されています。Lambda ログパーサーは、AWS WAF レートベースのルールおよび Amazon Athena ログパーサーオプションが利用できない場合にのみ使用します。このオプションの既知の制限は、処理中のファイルのコンテキスト内で情報が処理されることです。例えば、あるIPでは、定義されたクォータよりも多くのリクエストやエラーが生成される場合がありますが、この情報は複数のファイルに分割されているため、各ファイルにはクォータを超えるほどのデータが保存されません。

> **Note**
>
> さらに、不正なボットがハニーポットをバイパスして ALB または CloudFront と直接やり取りする場合、検出では、選択したログパーサーオプションにより、悪意のあるアクティビティを効果的に特定してブロックします。

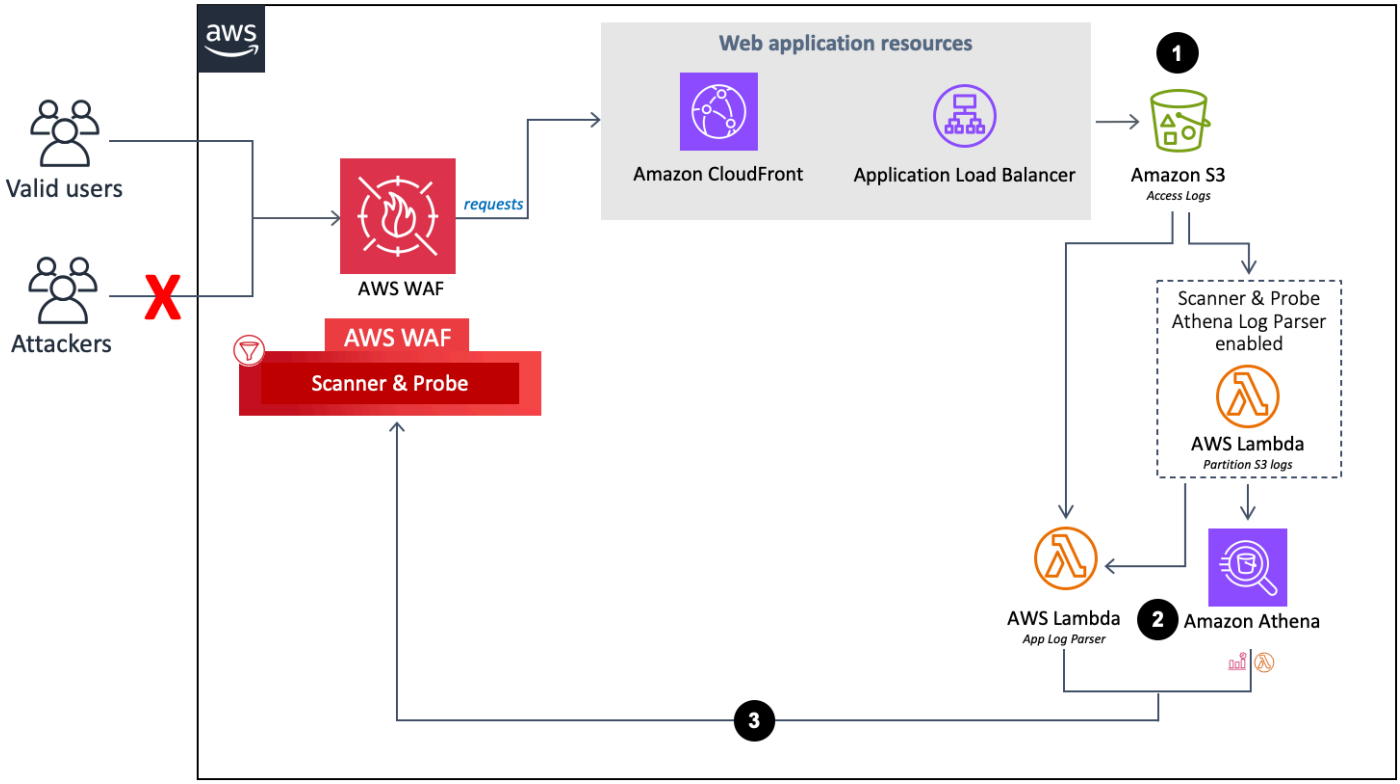
コンポーネントの詳細

アーキテクチャの図で説明した通り、このソリューションの 4 つのコンポーネントはオートメーションを使用して IP アドレスを検査し、AWS WAF ブロックリストに追加します。次のセクションでは、これらの各コンポーネントについて詳しく説明します。

ログパーサー – アプリケーション

アプリケーションログパーサーは、スキャナーとプローブから保護するのに役立ちます。

アプリケーションログパーサーフロー。



1. CloudFront または ALB がウェブアプリケーションに代わってリクエストを受信すると、アクセスログを Amazon S3 バケットに送信します。
   a. (オプション) [Activate HTTP Flood Protection] と [Activate Scanner & Probe Protection] テンプレートパラメータで `Yes - Amazon Athena log parser` を選択すると、Lambda 関数はアクセスログが Amazon S3 に到着した時点で元のフォルダ *`<customer-bucket>`* / AWSLogs から新しくパーティション分割されたフォルダ *`<customer-bucket>`* /AWSLogs-partitioned/ *`<optional-prefix>`* /year= *`<YYYY>`* /month= *`<MM>`* /day= *`<DD>`* /hour= *`<HH>`*/ に移動します。

b. (オプション) Keep Data in Original S3 location テンプレートパラメータで [yes] を選択すると、ログは元の場所に残り、パーティション分割されたフォルダにコピーされ、ログストレージが複製されます。

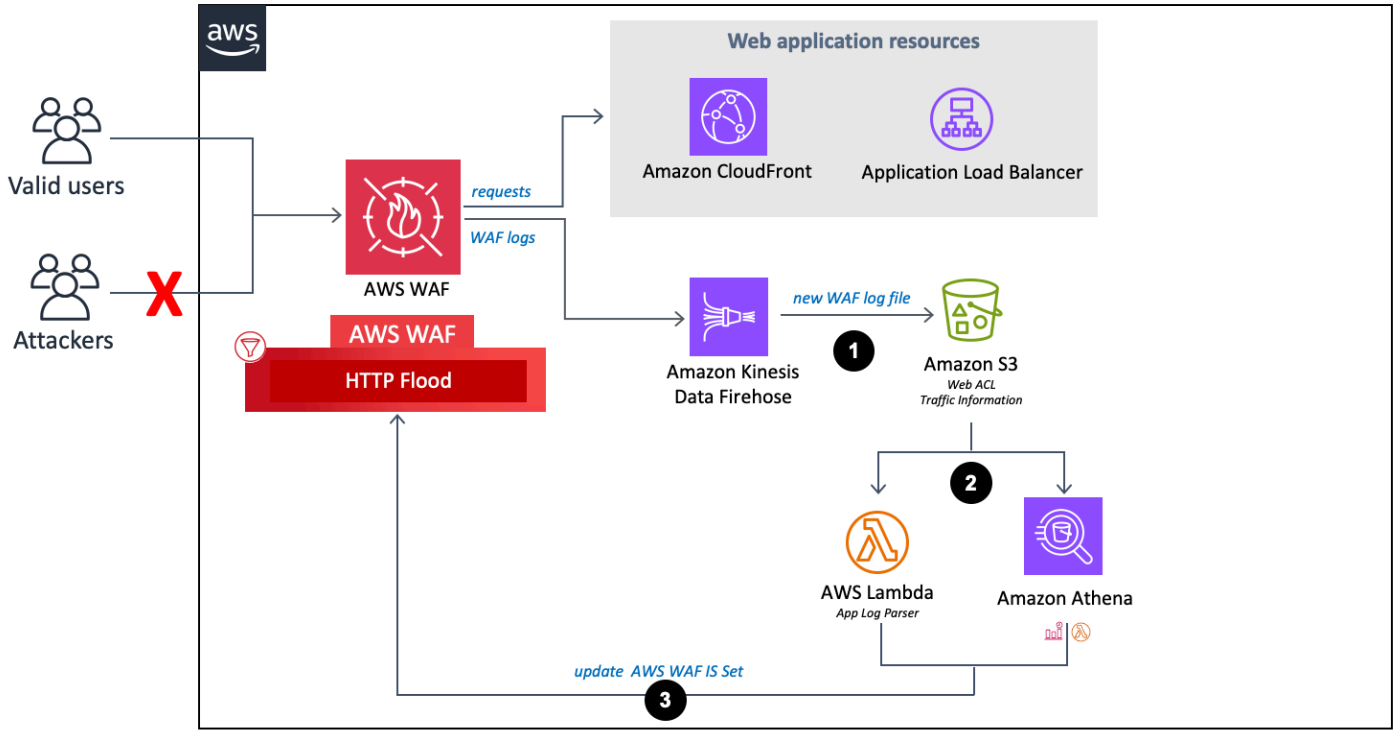
> **Note**
>
> Athena ログパーサーの場合、このソリューションは、このソリューションのデプロイ後に Amazon S3 バケットに到着する新しいログのみをパーティション分割します。パーティション分割したい既存のログがある場合は、このソリューションをデプロイした後、これらのログを手動で Amazon S3 にアップロードする必要があります。

2. Activate HTTP Flood Protection と Activate Scanner & Probe Protection のテンプレートパラメータでの選択に基づいて、このソリューションは次のいずれかを使用してログを処理します。
   a. Lambda – 新しいアクセスログが Amazon S3 バケットに保存されるたびに、`Log Parser` Lambda 関数が開始されます。
   b. Athena – デフォルトでは、Scanner & Probe Protection Athena クエリが 5 分ごとに実行され、出力が AWS WAF にプッシュされます。このプロセスは CloudWatch イベントによって開始されます。このイベントは Athena クエリの実行を担当する Lambda 関数を開始し、その結果を AWS WAF にプッシュします。
3. このソリューションは、ログデータを分析して、定義されたクォータよりも多くのエラーを生成した IP アドレスを特定します。次に、このソリューションは AWS WAF IP セット条件を更新して、お客様が定義した期間、それらの IP アドレスをブロックします。

ログパーサー – AWS WAF

Activate HTTP Flood Protection で [`yes - AWS Lambda log parser`] または [`yes - Amazon Athena log parser`] を選択すると、このソリューションは次のコンポーネントをプロビジョニングします。これらのコンポーネントは AWS WAF ログを解析して、定義したクォータを超えるリクエストレートでエンドポイントをフラッドするオリジンを識別してブロックします。

AWS WAF ログパーサーフロー。

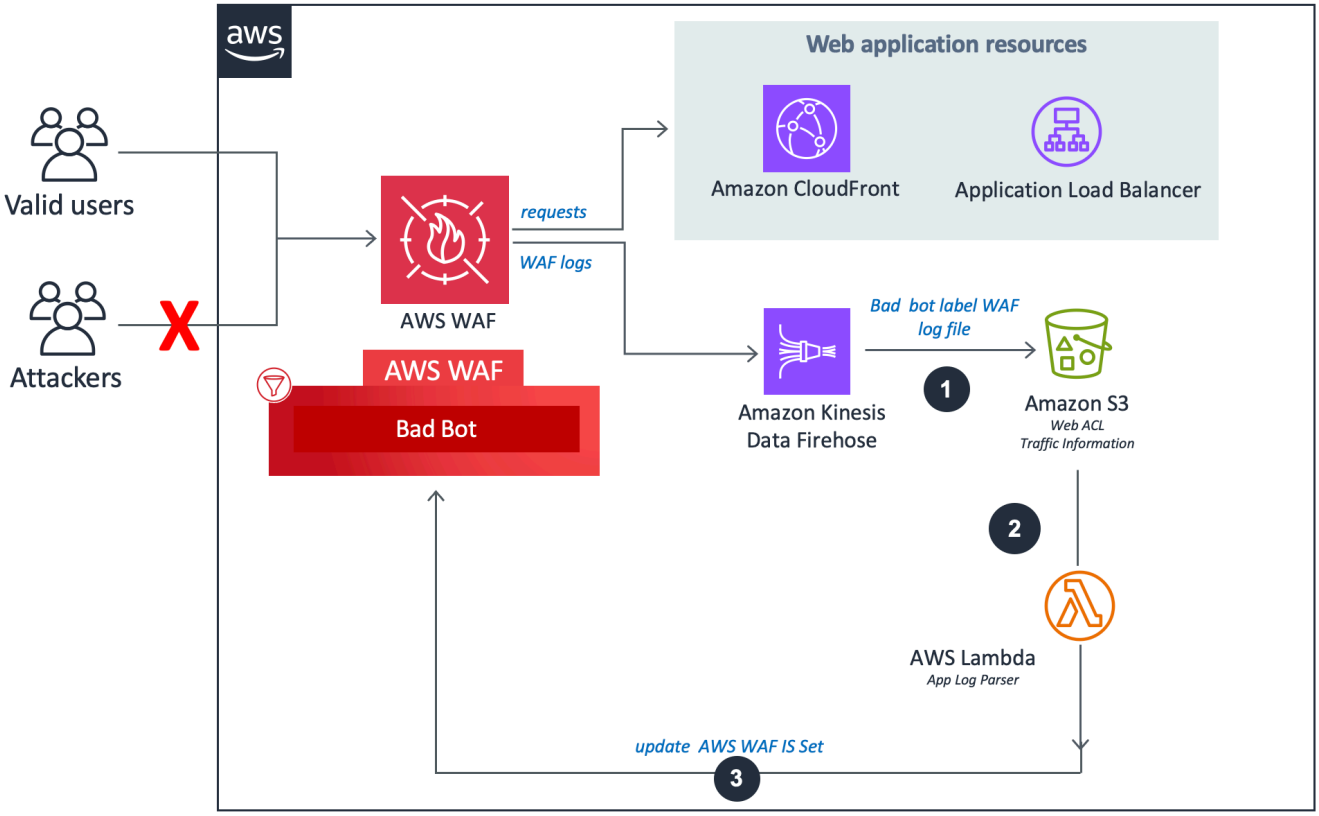


1. AWS WAF はアクセスログを受信すると、ログを Firehose エンドポイントに送信します。その後、Firehose は Amazon S3 内のパーティション分割されたバケット *<customer-bucket>* / AWSLogs/ *<optional-prefix>* /year= *<YYYY>* /month= *<MM>* /day= *<DD>* /hour= *<HH>* / にログを配信します。
2. Activate HTTP Flood Protection と Activate Scanner and Probe Protection のテンプレートパラメータでの選択に基づいて、このソリューションは次のいずれかを使用してログを処理します。
   a. Lambda: 新しいアクセスログが Amazon S3 バケットに保存されるたびに、`Log Parser` Lambda 関数が開始されます。
   b. Athena: デフォルトで、スキャナーとプローブの Athena クエリが 5 分ごとに実行され、その出力が AWS WAF にプッシュされます。このプロセスは、Amazon CloudWatch イベントによって開始され、その後 Amazon Athena クエリの実行を担当する Lambda 関数が開始され、その結果が AWS WAF にプッシュされます。
3. このソリューションは、ログデータを分析して、定義されたクォータよりも多くのリクエストを送信した IP アドレスを特定します。次に、このソリューションは AWS WAF IP セット条件を更新して、お客様が定義した期間、それらの IP アドレスをブロックします。

ログパーサー - 不正なボット

不正なボットのログパーサーは、ハニーポットエンドポイントへのリクエストを検査して、不正なボットの送信元 IP アドレスを抽出します。

不正なボットのログパーサーフロー。

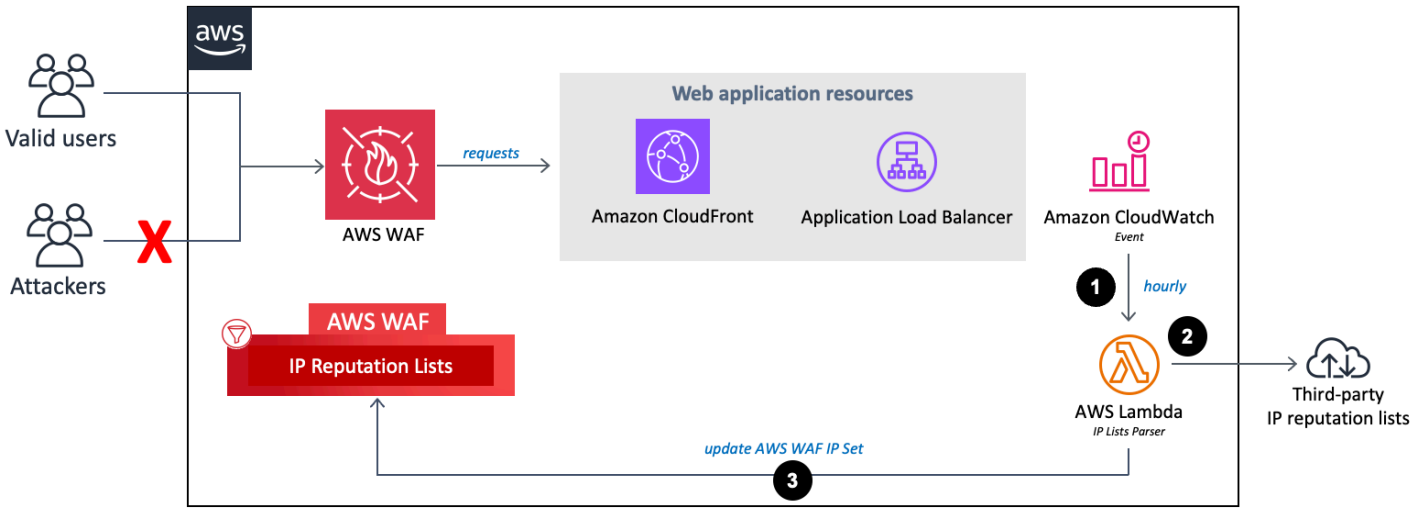


1. Bad Bot Protection がアクティブ化され、HTTP フラッド保護およびスキャナーとプローブ保護の両機能が無効になっている場合: システムは Log Lambda パーサーを使用し、WAF ラベルフィルターに基づいて不正なボットリクエストのみをログに記録します。
2. Lambda 関数は、リクエストヘッダーを傍受して検査し、トラップエンドポイントにアクセスした送信元の IP アドレスを抽出します。
3. このソリューションは、ログデータを分析して、定義されたクォータよりも多くのリクエストを送信した IP アドレスを特定します。次に、このソリューションは AWS WAF IP セット条件を更新して、お客様が定義した期間、それらの IP アドレスをブロックします。

IP リストパーサー

`IP Lists Parser` Lambda 関数は、サードパーティーの IP 評価リストで識別された既知の攻撃者からの保護に役立ちます。

IP 評価リストパーサーフロー。



1. 1 時間ごとの Amazon CloudWatch イベントにより `IP Lists Parser` Lambda 関数が起動されます。
2. Lambda 関数は、次の 3 つのソースからデータを収集して解析します。
   - Spamhaus DROP と EDROP リスト
   - Proofpoint Emerging Threats IP リスト
   - Tor exit node リスト
3. Lambda 関数は、AWS WAF ブロックリストを現在の IP アドレスで更新します。

デプロイを計画する

このセクションでは、このソリューションをデプロイする前に、コスト、セキュリティ、クォータ、およびその他の考慮事項について説明します。

サポートしている AWS リージョン

定義したテンプレート入力パラメータ値に応じて、このソリューションに必要なリソースは異なります。これらのリソース (次の表を参照) は、すべての AWS リージョンで利用できるとは限りません。そのため、これらのサービスが利用可能な AWS リージョンでこのソリューションを起動する必要があります。リージョン別の AWS サービスの最新情報については、AWS リージョン別のサービスのリストを参照してください。

	AWS WAF ウェブ ACL	AWS Glue	Amazon Athena	Amazon Kinesis Data Firehose
エンドポイントタイプ				
CloudFront	✓			
Application Load Balancer (ALB)	✓			
Activate HTTP Flood Protection				
yes – AWS Lambda ログパーサー				✓
yes - Amazon Athena log parser		✓	✓	✓
Activate Scanner & Probe Protection				

	AWS WAF ウェブ ACL	AWS Glue	Amazon Athena	Amazon Kinesis Data Firehose
yes - Amazon Athena log parser		✓	✓	

> **Note**
>
> エンドポイントとして CloudFront を選択した場合は、ソリューションを米国東部 (バージニア北部) リージョン (us-east-1) にデプロイする必要があります。

Cost

AWS WAF のセキュリティオートメーションソリューションの実行中に使用した AWS のサービスのコストは、お客様の負担となります。このソリューションを実行するための合計コストは、アクティブ化された保護、取り込み、保存、および処理されるデータの量によって異なります。

AWS Cost Explorer を使用して予算を作成することをお勧めします。これはコスト管理に役立ちます。詳細については、このソリューションで使用する AWS のサービスごとに料金ウェブページを参照してください。

次の表に、米国東部 (バージニア北部) リージョンでこのソリューションを実行した場合の、(AWS 無料利用枠は除く) コストの内訳例を示します。価格は変更されることがあります。

例 1: HTTP Flood Protection および Scanner & Probe Protection の Reputation List Protection、Bad Bot Protection、AWS Lambda、Log Parser をアクティブにする場合

AWS のサービス	ディメンション/月	コスト [USD]
Amazon Data Firehose	100 GB	~ 2.90 USD
Amazon S3	100 GB	~ 2.30 USD
AWS Lambda	128 MB: 3 つの関数。100 万回の呼び出しと、平均で 500	~ 5.40 USD

AWS のサービス	ディメンション/月	コスト [USD]
	ミリ秒の各 Lambda の実行時間 512 MB: 2 つの関数。100 万回の呼び出しと、平均で 500 ミリ秒の各 Lambda の実行時間	
AWS WAF ウェブ ACL	1	5.00 USD
AWS WAF ルール	4	4.00 USD
AWS WAF リクエスト	100 万回	0.60 USD
合計		~20.60 USD/月

例 2: HTTP Flood Protection および Scanner & Probe Protection の Reputation List Protection、Bad Bot Protection、Amazon Athena Log Parser をアクティブにする場合

AWS のサービス	ディメンション/月	コスト [USD]
Amazon Data Firehose	100 GB	~2.90 USD
Amazon S3	100 GB	~2.30 USD
AWS Lambda	128 MB: 3 つの関数。100 万回の呼び出しと、平均で 500 ミリ秒の各 Lambda の実行時間 512 MB: 2 つの関数。7560 回の呼び出しと、平均で 500 ミリ秒の各 Lambda の実行時間	~1.26 USD
Amazon Athena	Amazon CloudFront オブジェクトに対する 120 万回のヒット、もしくは 1 日あたり 120	~4.32 USD

AWS のサービス	ディメンション/月	コスト [USD]
	万回の ALB リクエストが、ヒットもしくはリクエストごとに、最大で 500 バイトのログレコードを生成	
AWS WAF ウェブ ACL	1	5.00 USD
AWS WAF ルール	4	4.00 USD
AWS WAF リクエスト	100 万回	0.60 USD
合計		~20.38 USD/月

例 3: 許可された IP セットと拒否された IP セットの IP 保持をアクティブにする

AWS のサービス	ディメンション/月	コスト [USD]
Amazon DynamoDB	1,000 回の書き込みと 1 MB のデータストレージ	~ 0.00 USD
AWS Lambda	128 MB: 1 つの関数。2000 回の呼び出しと、平均で 500 ミリ秒の各 AWS Lambda の実行時間 512 MB: 1 つの関数。2000 回の呼び出しと、平均で 500 ミリ秒の各 AWS Lambda の実行時間	~ 0.01 USD
Amazon CloudWatch	2,000 件のイベント	~ 0.00 USD
AWS WAF ウェブ ACL	1	5.00 USD
AWS WAF ルール	2	2.00 USD
WAS WAF リクエスト	100 万回	0.60 USD

AWS のサービス	ディメンション/月	コスト [USD]
合計		~7.61 USD/月

CloudWatch ログのコスト見積もり

このソリューションで使用される一部の AWS のサービス (Lambda など) は CloudWatch ログを生成します。これらのログには料金が発生します。コストを削減するために、ログを削除またはアーカイブすることをお勧めします。ログのアーカイブについては、Amazon CloudWatch Logs ユーザーガイドの「Amazon S3 へのログデータのエクスポート」を参照してください。

インストール時に Athena ログパーサーの使用を選択した場合、このソリューションは設定された Amazon S3 バケットの AWS WAF またはアプリケーションアクセスログに対してクエリが実行されるようにスケジュールされます。各クエリでスキャンされるデータ量に基づいて課金されます。このソリューションは、コストを最小限に抑えるために、ログとクエリにパーティション化を適用します。デフォルトでは、ソリューションはアプリケーションアクセスログを元の Amazon S3 の場所からパーティション化されたフォルダ構造に移動します。オリジナルを保持することもできますが、重複したログストレージに対しては課金されます。このソリューションでは、ワークグループを使用してワークロードをセグメント化し、クエリアクセスとコストの両方を管理するように構成できます。コスト見積もりの計算例については、「Athena のコスト見積もり」を参照してください。詳細については、「Amazon Athena の料金表」を参照してください。

Athena のコスト見積もり

HTTP フラッド保護、スキャナーとプローブ保護、不正なボットからの保護のいずれかのルールを実行中に Athena ログパーサーオプションを使用すると、Athena の使用料金が発生します。デフォルトでは、各 Athena クエリは 5 分ごとに実行され、過去 4 時間のデータをスキャンします。このソリューションは、コストを最小限に抑えるために、ログと Athena クエリにパーティション化を適用します。WAF Block Period テンプレートパラメータの値を変更することで、クエリがスキャンするデータの時間数を設定できます。ただし、スキャンされるデータ量を増やすと、Athena のコストが増加する可能性があります。

> **Tip**
>
> CloudFront ログのコスト計算の例を次に示します。
> 平均して、CloudFront のヒットごとに約 500 バイトのデータが生成される可能性があります。

1 日あたり 120 万件の CloudFront オブジェクトのヒット数がある場合、データが一定の速度で受信すると仮定すると、4 時間あたり 200 万件 (1.2 M/6) のヒットが発生します。コストを計算するときは、実際のトラフィックパターンを考慮してください。

`[500 bytes of data] * [200K hits per four hours] = [an average 100 MB (0.0001TB) data scanned per query]`

Athena では、スキャンされたデータ 1 TB あたり 5.00 USD が課金されます。

`[0.0001 TB] * [$5] = [$0.0005 per query scan]`

Athena クエリは 5 分ごとに実行され、1 時間あたり 12 回実行されます。

`[12 runs] * [24 hours] = [288 runs per day]`

`[$0.0005 per query scan] * [288 runs per day] * [30 days] = [$4.32 per month]`

実際のコストは、アプリケーションのトラフィックパターンによって異なります。詳細については、「Amazon Athena の料金表」を参照してください。

セキュリティ

AWS インフラストラクチャでシステムを構築すると、お客様と AWS の間でセキュリティ上の責任が分担されます。この責任共有モデルにより、ホストオペレーティングシステムと仮想化レイヤーからサービスが運用されている施設の物理的なセキュリティに至るまでのコンポーネントを AWS が運用、管理、制御するため、お客様の運用上の負担を軽減するのに役立ちます。AWS セキュリティの詳細については、「AWS クラウドセキュリティ」を参照してください。

IAM ロール

IAM ロールを使用すると、AWS クラウドのサービスとユーザーに、きめ細かなアクセスポリシーとアクセス許可を割り当てることができます。このソリューションでは、ソリューションのリソースに対して必要なアクセス許可を付与する、最小特権の IAM ロールが作成されます。

データ

Amazon S3 バケットと DynamoDB テーブルに保存されたすべてのデータは、保管時に暗号化されます。Firehose で転送中のデータも暗号化されます。

保護機能

ウェブアプリケーションは、さまざまな攻撃に対して脆弱です。これらの攻撃には、脆弱性を悪用したりサーバーを制御したりするように特別に設計· 作成されたリクエスト、ウェブサイトを破壊する

ように設計された帯域幅消費型攻撃、ウェブコンテンツを破壊し盗むようにプログラムされた悪質なボットやスクレイパーなどが含まれます。

このソリューションでは、CloudFormation を使用して、AWS マネージドルールのルールグループやカスタムルールなどの AWS WAF ルールを設定し、次のような一般的な攻撃をブロックします。

- AWS マネージドルール – このマネージドサービスは、一般的なアプリケーションの脆弱性やその他の望ましくないトラフィックに対する保護を提供します。このソリューションには、AWS マネージド IP 評価ルールグループ、AWS マネージドベースラインルールグループ、AWS マネージドユースケース固有のルールグループが含まれます。ウェブ ACL のキャパシティユニット (WCU) クォータの上限まで、ウェブ ACL に 1 つまたは複数のルールグループを選択するオプションがあります。
- SQL インジェクション – 攻撃者は、データベースからデータを抽出するために、ウェブリクエストに悪意のある SQL コードを挿入します。このソリューションは、悪意のある可能性がある SQL コードを含むウェブリクエストをブロックするように設計されています。
- XSS – 攻撃者は悪質なウェブサイトの脆弱性を利用して、悪意のあるクライアントサイトスクリプトを正当なユーザーのウェブブラウザに挿入します。このソリューションは、XSS 攻撃を識別してブロックするために、一般的に調査される受信リクエストの要素を検査するように設計されています。
- HTTP フラッド – ウェブサーバーやその他のバックエンドリソースには、HTTP フラッドなどの DDoS 攻撃のリスクがあります。このソリューションでは、クライアントからのウェブリクエストが設定したクォータを超えると、レートベースのルールが自動的に起動されます。または、Lambda 関数または Athena クエリを使用して AWS WAF ログを処理することで、このクォータを適用することもできます。
- スキャナーとプローブ – 悪意のある送信元は、HTTP 4xx エラーコードを生成する一連のリクエストを送信することで、インターネットに接続しているウェブアプリケーションの脆弱性をスキャンして調査します。この履歴を使用して、悪意のある送信元 IP アドレスを識別してブロックできます。このソリューションは、CloudFront または ALB のアクセスログを自動的に解析し、1 分あたりの一意の送信元 IP アドレスからの不正リクエストの数をカウントして、定義されたエラークォータに達したアドレスからのさらなるスキャンをブロックするように AWS WAF を更新する Lambda 関数または Athena クエリを作成します。
- 既知の攻撃元 (IP 評価リスト) – 多くの組織が、スパマー、マルウェアディストリビューター、ボットネットなど、既知の攻撃者が用いる IP アドレスの評価リストを保持しています。このソリューションは、これらの評価リストの情報を活用して、悪質な送信元 IP アドレスからのリクエストをブロックするのに役立ちます。さらに、このソリューションは、Amazon の内部脅威インテリジェンスに基づいて IP 評価ルールグループによって識別される攻撃者をブロックします。

- ボットとスクレイパー – 一般的に公開されたウェブアプリケーションの管理者は、コンテンツにアクセスするクライアントが自分自身を正確に識別し、意図したとおりにサービスを使用することを信頼する必要があります。ただし、コンテンツスクレイパーや悪質なボットなどの一部の自動化されたクライアントは、制限を回避するために自分自身を偽装しています。このソリューションは、悪質なボットやスクレイパーを識別してブロックするのに役立ちます。

クォータ

サービスクォータ (制限とも呼ばれます) は、AWS アカウント用のサービスリソースまたはオペレーションの最大数です。

このソリューション内の AWS サービスのクォータ

このソリューションに実装されている各サービスに十分なクォータがあることを確認してください。詳細については、「AWS サービスクォータ」を参照してください。ページを切り替えずにドキュメント内のすべての AWS のサービスのサービスクォータを表示するには、こちらの PDF にある「Service endpoints and quotas」ページの情報を確認してください。

AWS WAF クォータ

AWS WAF は、IP の一致条件ごとに、Classless Inter-Domain Routing (CIDR) で最大 10,000 の IP アドレス範囲をブロックできます。このソリューションが作成する各リストには、このクォータが適用されます。詳細については、「AWS WAF クォータ」を参照してください。バージョン 3.0 以降、このソリューションは各ルールにアタッチする 2 つの IP セット (IPv4 用、IPv6 用) を作成します。

AWS WAF では、個々の `Create`、`Put`、または `Update` アクションへの API コールに対して、アカウントごとに、1 秒あたり、AWS リージョンごとに最大 1 つのリクエストが許可されます。これらの API コールをソリューションの外部で行うと、API スロットリングの問題が発生する可能性があります。この問題を回避するには、このソリューションがデプロイされているのと同じアカウントとリージョンでこれらの API コールを行う他のアプリケーションを実行しないことをお勧めします。

デプロイに関する考慮事項

次のセクションでは、このソリューションを実装するための制約と考慮事項について説明します。

AWS WAF ルール

このソリューションが生成するウェブ ACL は、ウェブアプリケーションを包括的に保護するように設計されています。このソリューションは、ウェブ ACL に追加できる一連の AWS マネージドルールおよびカスタムルールを提供します。ルールを含めるには、CloudFormation スタックを起動するときに関連するパラメータで [yes] を選択します。パラメータのリストに関しては「ステップ 1. スタックを起動する」を参照してください。

> **Note**
>
> すぐに使用できるソリューションは AWS Firewall Manager をサポートしていません。Firewall Manager でルールを使用する場合は、ソースコードにカスタマイズを適用することをお勧めします。

ウェブ ACL トラフィックロギング

米国東部 (バージニア北部) 以外の AWS リージョンでスタックを作成して、エンドポイント を [CloudFront] に設定する場合は、Activate HTTP Flood Protection を [no] または [`yes - AWS WAF rate based rule`] に設定する必要があります。

他の 2 つのオプション (`yes - AWS Lambda log parser` と `yes - Amazon Athena log parser`) では、すべての AWS エッジロケーションで実行されるウェブ ACL で AWS WAF ログをアクティブ化する必要があります。これは米国東部 (バージニア北部) 以外ではサポートされていません。ウェブ ACL トラフィックのロギングに関する詳細については、「AWS WAF 開発者ガイド」を参照してください。

過剰サイズのリクエストコンポーネントの処理

AWS WAF は、ウェブリクエストコンポーネント本文、ヘッダー、または cookie のオーバーサイズのコンテンツの検査をサポートしていません。これらのリクエストコンポーネントタイプの 1 つを検査するルールステートメントを作成する場合、これらのオプションのいずれかを選択して、これらのリクエストの処理方法を AWS WAF に指示できます。

- `yes (continue)` – ルール検査基準に従って、リクエストコンポーネントを通常どおり検査します。AWS WAF は、サイズ制限内のリクエストコンポーネントのコンテンツを検査します。このソリューションで使用するデフォルトのオプションです。

- yes - MATCH - ウェブリクエストをルールステートメントと一致するものとして扱います。AWS WAF は、ルールの検査基準に照らして評価することなく、ルールアクションをリクエストに適用します。Block アクションのルールの場合、オーバーサイズコンポーネントでのリクエストをブロックします。
- yes - NO_MATCH – ルールの検査基準に対して評価せずに、ウェブリクエストをルールステートメントに一致しないものとして扱います。AWS WAF は、一致しないルールの場合と同様に、ウェブ ACL の残りのルールを使用してウェブリクエストの検査を続行します。

詳細については、「AWS WAF でのオーバーサイズウェブリクエストコンポーネントの処理」を参照してください。

複数のソリューションデプロイ

ソリューションは、同じアカウントとリージョンに複数回デプロイできます。デプロイごとに一意の CloudFormation スタック名と Amazon S3 バケット名を使用する必要があります。それぞれの固有のデプロイメントには追加料金が発生し、アカウントごと、リージョンごとの AWS WAF クォータが適用されます。

デプロイに必要なロールの最小限のアクセス許可 (オプション)

お客様は、デプロイに必要な最小限のアクセス許可を持つ IAM ロールを手動で作成できます。

- WAF のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "wafv2:CreateWebACL",
                "wafv2:UpdateWebACL",
                "wafv2:DeleteWebACL",
                "wafv2:GetWebACL",
                "wafv2:ListWebACLs",
                "wafv2:CreateIPSet",
                "wafv2:UpdateIPSet",
                "wafv2:DeleteIPSet",
                "wafv2:GetIPSet",
                "wafv2:AssociateWebACL",
                "wafv2:DisassociateWebACL",
                "wafv2:PutLoggingConfiguration",
```

```
                "wafv2:DeleteLoggingConfiguration",
                "wafv2:ListWebACLs",
                "wafv2:ListIPSets",
                "wafv2:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:wafv2:*:*:regional/webacl/*",
                "arn:aws:wafv2:*:*:regional/ipset/*",
                "arn:aws:wafv2:*:*:global/webacl/*",
                "arn:aws:wafv2:*:*:global/ipset/*"
            ]
        }
```

- Lambda のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:AddPermission",
                "lambda:RemovePermission"
            ],
            "Resource": "arn:aws:lambda:*:*:function:*"
        }
```

- Firehose のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "firehose:CreateDeliveryStream",
                "firehose:DeleteDeliveryStream",
                "firehose:DescribeDeliveryStream",
                "firehose:StartDeliveryStreamEncryption",
                "firehose:StopDeliveryStreamEncryption",
```

```
                "firehose:UpdateDestination"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/*"
        }
```

- S3 のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteBucketPolicy",
                "s3:GetBucketAcl",
                "s3:GetBucketPolicy",
                "s3:GetObject",
                "s3:PutBucketAcl",
                "s3:PutBucketPolicy",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketVersioning",
                "s3:PutEncryptionConfiguration",
                "s3:PutObject",
                "s3:PutBucketTagging",
                "s3:PutLifecycleConfiguration",
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:PutBucketLogging",
                "s3:GetBucketLogging"
            ],
            "Resource": "arn:aws:s3:::*"
        }
```

- Athena アクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "athena:CreateWorkGroup",
```

```
                "athena:DeleteWorkGroup",
                "athena:GetWorkGroup",
                "athena:UpdateWorkGroup",
                "athena:StartQueryExecution",
                "athena:GetQueryExecution",
                "athena:GetQueryResults",
                "athena:StopQueryExecution"
            ],
            "Resource": "arn:aws:athena:*:*:workgroup/WAF*"
        }
```

- Glue のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:UpdateDatabase",
                "glue:CreateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:UpdateTable"
            ],
            "Resource": [
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*",
                "arn:aws:glue:*:*:table/*/*",
                "arn:aws:glue:*:*:userDefinedFunction/*"
            ]
        }
```

- CloudWatch Logs のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
```

```
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DeleteLogGroup",
                "logs:DeleteLogStream",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/lambda/*",
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:*:log-group:/aws/kinesisfirehose/*"
            ]
        }
```

- CloudWatch のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DeleteDashboards",
                "cloudwatch:GetDashboard",
                "cloudwatch:ListDashboards",
                "cloudwatch:PutDashboard",
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*"
        }
```

- SNS のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sns:SetTopicAttributes"
            ],
```

```
            "Resource": "arn:aws:sns:*:*:*"
        }
```

- DynamoDB のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:DescribeTable",
                "dynamodb:PutItem",
                "dynamodb:GetItem",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteItem"
            ],
            "Resource": "arn:aws:dynamodb:*:*:table/*"
        }
```

- CloudFormation のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack",
                "cloudformation:ListStacks"
            ],
            "Resource": "arn:aws:cloudformation:*:*:stack/*/*"
        }
```

- Service Catalog App Registry のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:CreateApplication",
```

```
                "servicecatalog:DeleteApplication",
                "servicecatalog:GetApplication",
                "servicecatalog:TagResource",
                "servicecatalog:CreateAttributeGroup",
                "servicecatalog:DeleteAttributeGroup",
                "servicecatalog:GetAttributeGroup",
                "servicecatalog:AssociateAttributeGroup",
                "servicecatalog:DisassociateAttributeGroup",
                "servicecatalog:AssociateResource",
                "servicecatalog:DisassociateResource"
            ],
            "Resource": "arn:aws:servicecatalog:*:*:*"
        }
```

- X-Ray のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords"
            ],
            "Resource": "*"
        }
```

- IAM 許可

```
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreatePolicy",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListRoles",
                "iam:PassRole",
```

```
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        }
```

- EventBridge のアクセス許可

```
        {
            "Effect": "Allow",
            "Action": [
                "events:PutTargets",
                "events:RemoveTargets",
                "events:DescribeRule",
                "events:EnableRule",
                "events:ListRules",
                "events:PutRule",
                "events:DeleteRule",
                "events:ListEventSources",
                "events:DescribeEventSource",
                "events:ActivateEventSource",
                "events:DeactivateEventSource"
            ],
            "Resource": "arn:aws:events:*:*:rule/*"
        }
```

ソリューションをデプロイする

このソリューションは、AWS CloudFormation テンプレートとスタックを使用してデプロイを自動化します。CloudFormation テンプレートは、このソリューションに含まれる AWS リソースとそのプロパティを指定します。CloudFormation スタックは、テンプレートに記述されているリソースをプロビジョニングします。

デプロイプロセスの概要

CloudFormation テンプレートを起動する前に、このガイドで説明しているアーキテクチャと設定の考慮事項を確認してください。このセクションのステップバイステップの手順に従って、ソリューションを設定してアカウントにデプロイします。

デプロイ時間: 約 15 分

> **Note**
>
> すでにこのソリューションをデプロイしている場合は、「ソリューションのアップデート」でアップデートの手順を参照してください。

前提条件

- CloudFront ディストリビューションを設定する
- ALB を設定します

ステップ 1. スタックを起動する

- AWS アカウントで CloudFormation テンプレートを起動します。
- 必須パラメータの値を入力します: スタック名、Application Access Log Bucket Name
- 他のテンプレートパラメータを確認して、必要に応じて調整します。

ステップ 2. ウェブ ACL をウェブアプリケーションに関連付ける

- CloudFront ウェブディストリビューションまたは ALB を、このソリューションが生成するウェブ ACL に関連付けます。ディストリビューションまたはロードバランサーを必要な数だけ関連付けることができます。

ステップ 3. ウェブアクセスロギングを設定する

- CloudFront ウェブディストリビューションまたは ALB のウェブアクセスロギングをオンにして、ログファイルを適切な Amazon S3 バケットに送信します。ユーザー定義のプレフィックスに一致するフォルダにログを保存します。ユーザー定義プレフィックスが使用されていない場合は、ログを AWSLogs (デフォルトのログプレフィックス AWSLogs/) に保存します。詳細については、「ステップ 1. スタックを起動する」の Application Access Log Bucket Prefix パラメータを参照してください。詳細については「Step 1. Launch the stack」を参照してください。

AWS CloudFormation テンプレート

このソリューションには、1 つのメイン AWS CloudFormation テンプレートと 2 つのネストされたテンプレートが含まれています。ソリューションをデプロイする前に CloudFormation テンプレートをダウンロードできます。

メインスタック

View template

aws-waf-security-automations.template - このテンプレートをエントリポイントとして使用して、アカウントでソリューションを起動します。デフォルト設定では、事前設定されたルールを使用して AWS WAF ウェブ ACL がデプロイされます。また、ニーズに応じてテンプレートをカスタマイズすることもできます。

WebACL スタック

View template

aws-waf-security-automations-webacl.template – このネストされたテンプレートは、ウェブ ACL、IP、セット、その他の関連リソースなどの AWS WAF リソースをプロビジョニングします。

Firehose Athena スタック

View template

aws-waf-security-automations-firehose-athena.template – このネストされたテンプレートは、AWS Glue、Athena、および Firehose に関連するリソースをプロビジョニングします。これは、Scanner

& Probe Athena ログパーサー、HTTP Flood Lambda または Athena ログパーサーのいずれかを選択したときに作成されます。

> **Note**
>
> AWS CloudFormation のリソースは、AWS Cloud Development Kit (AWS CDK) のコンストラクトで作成されています。

この AWS CloudFormation テンプレートは、AWS WAF ソリューションのセキュリティオートメーションを AWS クラウドにデプロイします。

前提条件

このソリューションは、CloudFront または ALB でデプロイされたウェブアプリケーションで動作するように設計されています。これらのリソースのいずれかが設定されていない場合は、このソリューションを起動する前に該当するタスクを完了してください。

CloudFront ディストリビューションを設定する

次の手順に従い、ウェブアプリケーションの静的および動的コンテンツ用に CloudFront でディストリビューションを設定します。詳細な手順については、「Amazon CloudFront 開発者ガイド」を参照してください。

1. CloudFront のウェブアプリケーション用のディストリビューションを作成します。「ディストリビューションを作成する」を参照してください。
2. 静的オリジンおよび動的オリジンを設定します。「CloudFront ディストリビューションでさまざまなオリジンを使用する」を参照してください。
3. ディストリビューションの動作を指定します。「ディストリビューションを作成または更新する場合に指定する値」を参照してください。

> **Note**
>
> エンドポイントとして `CloudFront` を選択した場合は、米国東部 (バージニア北部) リージョンに WAFV2 リソースを作成する必要があります。

ALB を設定します

着信トラフィックをウェブアプリケーションに分散するように ALB を設定するには、「Application Load Balancer ユーザーガイド」の「Application Load Balancer の作成」を参照してください。

ステップ 1. スタックを起動する

この自動 AWS CloudFormation テンプレートは、AWS クラウドにソリューションをデプロイします。

1. AWS マネジメントコンソールにサインインして、[ソリューションを起動] を選択して `waf-automation-on-aws.template` CloudFormation テンプレートを起動します。

   Launch solution

2. テンプレートはデフォルトで米国東部 (バージニア北部) リージョンで起動します。別の AWS リージョンでこのソリューションを起動するには、コンソールのナビゲーションバーのリージョンセレクターを使用します。エンドポイントとして `CloudFront` を選択した場合は、ソリューションを米国東部 (バージニア北部) (`us-east-1`) リージョンにデプロイする必要があります。

   > **Note**
   >
   > 定義する入力パラメータ値に応じて、このソリューションに必要なリソースは異なります。これらのリソースは現在、特定の AWS リージョンでのみ使用できます。そのため、これらのサービスが利用可能な AWS リージョンでこのソリューションを起動する必要があります。詳細については、「サポートしている AWS リージョン」を参照してください。

3. [テンプレートの指定] ページで、正しいテンプレートを選択したことを確認し、[次へ] を選択します。
4. [スタックの詳細を指定] ページの [スタック名] フィールドで AWS WAF 設定に名前を割り当てます。これは、テンプレートが作成するウェブ ACL の名前にもなります。
5. [パラメータ] で、テンプレートのパラメータを確認し、必要に応じて変更します。特定の機能をオプトアウトするには、必要に応じて none または no を選択します。このソリューションでは、次のデフォルト値を使用します。

パラメータ	デフォルト	説明
スタック名	[.red]#<入力必須>`	スタック名にスペースを含めることはできません。この名前は AWS アカウント内で一意である必要があり、テンプレートが作成するウェブ ACL の名前です。
リソースタイプ		
Endpoint	CloudFront	使用するリソースのタイプを選択します。注意: エンドポイントに CloudFront を選択した場合は、米国東部 (バージニア北部) リージョン (us-east-1) で WAF リソースを作成してソリューションを起動する必要があります。
AWS マネージドの IP 評価ルールグループ		

パラメータ	デフォルト	説明
Activate Amazon IP reputation List Managed Rule Group Protection	no	Amazon IP 評価リストマネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、Amazon の内部脅威インテリジェンスに基づきます。これは、通常、ボットやその他の脅威に関連付けられている IP アドレスをブロックする場合に便利です。これらの IP アドレスをブロックすることで、ボットを緩和し、悪意のあるアクターが脆弱なアプリケーションを発見するリスクを緩和できます。 必要な WCU は 25 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate Anonymous IP List Managed Rule Group Protection	no	ウェブ ACL に匿名 IP リストマネージドルールグループを追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、ビューワー ID の難読化を許可するサービスからのリクエストをブロックします。これには、VPN、プロキシ、Tor ノード、ホスティングプロバイダーなどからのリクエストが含まれます。このルールグループは、アプリケーションから ID を隠そうとするビューワーを除外する場合に便利です。これらのサービスの IP アドレスをブロックすると、ボットの緩和や地理的制限の回避に役立ちます。 必要な WCU は 50 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
AWS マネージドのベースラインルールグループ		
Activate Core Rule Set Managed Rule Group Protection	no	コアルールセット (CRS) マネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、リスクが高く一般的に発生するいくつかの脆弱性を含む、さまざまな脆弱性の悪用に対する保護を提供します。すべての AWS WAF ユースケースでこのルールグループを使用することを検討してください。 必要な WCU は 700 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate Admin Protection Managed Rule Group Protection	no	管理者保護マネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、公開されている管理ページへの外部アクセスをブロックします。これは、サードパーティーのソフトウェアを実行している場合や、悪意のあるアクターがアプリケーションへの管理アクセスを得るリスクを緩和したい場合に便利です。 必要な WCU は 100 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate Known Bad Inputs Managed Rule Group Protection	no	ウェブ ACL に既知の不正な入力マネージドルールグループを追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、公開されている管理ページへの外部アクセスをブロックします。これは、サードパーティーのソフトウェアを実行している場合や、悪意のあるアクターがアプリケーションへの管理アクセスを得るリスクを緩和したい場合に便利です。 必要な WCU は 100 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。
AWS マネージドのユースケース固有のルールグループ		

パラメータ	デフォルト	説明
Activate SQL Database Managed Rule Group Protection	no	SQL データベースマネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、SQL インジェクション攻撃などの SQL データベースの悪用に関連するリクエストパターンをブロックします。これにより、不正なクエリのリモートインジェクションを防ぐことができます。アプリケーションが SQL データベースと連結している場合は、このルールグループを評価します。AWS マネージド SQL ルールグループが既に有効になっている場合、SQL インジェクションカスタムルールの使用はオプションです。 必要な WCU は 200 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate Linux Operating System Managed Rule Group Protection	no	Linux オペレーティングシステムマネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、Linux 固有のローカルファイルインクルージョン (LFI) 攻撃など、Linux 固有の脆弱性の悪用に関連するリクエストパターンをブロックします。これにより、攻撃者がアクセスしてはならないファイルの内容を公開したり、コードを実行したりする攻撃を防ぐことができます。アプリケーションの一部が Linux で実行されている場合は、このルールグループを評価します。このルールグループは、POSIX オペレーティングシステムルールグループと組み合わせて使用する必要があります。 必要な WCU は 200 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。

パラメータ	デフォルト	説明
		詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate POSIX Operating System Managed Rule Group Protection	no	コアルールセット (CRS) マネージドルールグループの保護をウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、POSIX および POSIX と同等のオペレーティングシステムに固有の脆弱性の悪用 (LFI 攻撃など) に関連するリクエストパターンをブロックします。これにより、攻撃者がアクセスしてはならないファイルの内容を公開したり、コードを実行したりする攻撃を防ぐことができます。アプリケーションの一部が POSIX または POSIX と同等のオペレーティングシステムで実行されている場合は、このルールグループを評価します。 必要な WCU は 100 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグ

パラメータ	デフォルト	説明
		ループリスト」を参照してください。
Activate Windows Operating System Managed Rule Group Protection	no	Windows オペレーティングシステムマネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、PowerShell コマンドのリモート実行など、Windows 固有の脆弱性の悪用に関連するリクエストパターンをブロックします。これにより、攻撃者が不正なコマンドまたは悪意のあるコードを実行できる脆弱性の悪用を防ぐことができます。アプリケーションの一部が Windows オペレーティングシステムで実行されている場合は、このルールグループを評価します。 必要な WCU は 200 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate PHP Application Managed Rule Group Protection	no	PHP アプリケーションマネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、安全でない PHP 関数のインジェクションなど、PHP プログラミング言語の使用に固有の脆弱性の悪用に関連するリクエストパターンをブロックします。これにより、攻撃者が許可されていないコードまたはコマンドを遠隔で実行できる脆弱性の悪用を防ぐことができます。アプリケーションが連結するサーバーに PHP がインストールされている場合は、このルールグループを評価します。 必要な WCU は 100 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。

パラメータ	デフォルト	説明
Activate WordPress Application Managed Rule Group Protection	no	WordPress アプリケーションマネージドルールグループをウェブ ACL に追加するように設計されたコンポーネントをオンにするには、yes を選択します。 このルールグループは、WordPress サイト固有の脆弱性の悪用に関連するリクエストパターンをブロックします。WordPress を実行している場合は、このルールグループを評価します。このルールグループは、SQL データベースおよび PHP アプリケーションルールグループと組み合わせて使用する必要があります。 必要な WCU は 100 です。アカウントには、容量制限を超えたためにウェブ ACL スタックのデプロイが失敗するのを避けるために、十分な WCU 容量が必要です。 詳細については、「AWS マネージドルールのルールグループリスト」を参照してください。
カスタムルール – スキャナー & プローブ		

パラメータ	デフォルト	説明
Activate Scanner & Probe Protection	`yes - AWS Lambda log parser`	スキャナーとプローブをブロックするために使用するコンポーネントを選択します。緩和策オプションに関連するトレードオフの詳細については、「ログパーサーオプション」を参照してください。

パラメータ	デフォルト	説明
Application Access Log Bucket Name	`[.red]<requires input>`	[Activate Scanner & Probe Protection] パラメータに yes を選択した場合は、CloudFront ディストリビューション (複数可) または ALB (複数可) のアクセスログを保存する Amazon S3 バケット (新規または既存) の名前を入力します。既存の Amazon S3 バケットを使用している場合は、CloudFormation テンプレートをデプロイしているのと同じ AWS リージョンにある必要があります。ソリューションのデプロイごとに異なるバケットを使用する必要があります。 この保護を無効にするには、このパラメータを無視します。注意: CloudFront ウェブディストリビューション (複数可) または ALB (複数可) のウェブアクセスログを有効にして、この Amazon S3 バケットにログファイルを送信します。スタックで定義されているのと同じプレフィックス (デフォルトのプレフィックス `AWSLogs/`) にログを保存します。詳細については、「Application Access Log Bucket Prefix」パラメータを参照してください。

パラメータ	デフォルト	説明
Application Access Log Bucket Prefix	`AWSLogs/`	[Activate Scanner & Probe Protection] パラメータに `yes` を選択した場合は、上記のアプリケーションアクセスログバケットにオプションのユーザー定義プレフィックスを入力できます。 Endpoint パラメータに `CloudFront` を選択した場合は、`yourprefix/` などの任意のプレフィックスを入力できます。 [Endpoint] パラメータに ALB を選択した場合は、`yourprefix/AWSLogs/` などのプレフィックスに `AWSLogs/` を追加する必要があります。 ユーザー定義のプレフィックスがない場合は、`AWSLogs/` (デフォルト) を使用します。 この保護を無効にするには、このパラメータを無視します。

パラメータ	デフォルト	説明
Is bucket access logging turned on?	no	[Application Access Log Bucket Name] パラメータに既存の Amazon S3 バケット名を入力し、バケットのサーバーアクセスログ記録が既にオンになっている場合は、yes を選択します。 no を選択すると、ソリューションはバケットのサーバーアクセスログ記録を有効にします。 [Activate Scanner & Probe Protection] パラメータに no を選択した場合は、このパラメータを無視します。
Error Threshold	50	[Activate Scanner & Probe Protection] パラメータに yes を選択した場合は、IP アドレスごとに 1 分あたりに許容される不正なリクエストの最大数を入力します。 [Activate Scanner & Probe Protection] パラメータに no を選択した場合は、このパラメータを無視します。

パラメータ	デフォルト	説明
Keep Data in Original S3 Location	no	[Activate Scanner & Probe Protection] パラメータに `yes - Amazon Athena log parser` を選択した場合、ソリューションはアプリケーションアクセスログファイルと Athena クエリにパーティション化を適用します。デフォルトでは、ソリューションはログファイルを元の場所から Amazon S3 のパーティション化されたフォルダ構造に移動します。 ログのコピーも元の場所に保持する場合は、`yes` を選択します。これにより、ログストレージが複製されます。 [Activate Scanner & Probe Protection] パラメータに `yes - Amazon Athena log parser` を選択しなかった場合は、このパラメータを無視します。
カスタムルール – HTTP フラッド		

パラメータ	デフォルト	説明
Activate HTTP Flood Protection	`yes - AWS WAF rate-based rule`	HTTP フラッド攻撃をブロックするために使用するコンポーネントを選択します。緩和策オプションに関連するトレードオフの詳細については、「ログパーサーオプション」を参照してください。
Default Request Threshold	`100`	[Activate HTTP Flood Protection] パラメータに `yes` を選択した場合は、IP アドレスごとに 5 分あたりの最大許容リクエスト数を入力します。 [Activate HTTP Flood Protection] パラメータに `yes - AWS WAF rate-based rule` を選択した場合、許容される最小値は `10` です。 [Activate HTTP Flood Protection] パラメータに `yes - AWS Lambda log parser` または `yes - Amazon Athena log parser` を選択した場合、任意の値にすることができます。 この保護を無効にするには、このパラメータを無視します。

パラメータ	デフォルト	説明
Request Threshold by Country	<オプション入力>	[Activate HTTP Flood Protection] パラメータに `yes - Amazon Athena log parser` を選択した場合は、この JSON 形式 `{"TR":50, "ER":150}` に従って国別にしきい値を入力できます。このソリューションは、指定された国から発信されたリクエストにこれらのしきい値を使用します。このソリューションは、残りのリクエストに [Default Request Threshold] パラメータを使用します。注意: このパラメータを定義すると、[Group By Requests in HTTP Flood Athena Query] パラメータで選択できる IP フィールドやその他のオプションのグループ別フィールドとともに、国が Athena クエリグループに自動的に含まれます。 + この保護を無効にすることを選択した場合は、このパラメータを無視します。

パラメータ	デフォルト	説明
Group By Requests in HTTP Flood Athena Query	None	[Activate HTTP Flood Protection] パラメータに yes - Amazon Athena log parser を選択した場合は、グループ別フィールドを選択して、IP あたりのリクエストと選択したグループ別フィールドをカウントできます。例えば、URI を選択した場合、ソリューションは IP および URI あたりのリクエストをカウントします。 この保護を無効にすることを選択した場合は、このパラメータを無視します。
WAF Block Period	240	[Activate Scanner & Probe Protection] または [Activate HTTP Flood Protection] パラメータに yes - AWS Lambda log parser または yes - Amazon Athena log parser を選択した場合は、該当する IP アドレスをブロックする期間 (分単位) を入力します。 ログ解析を無効にするには、このパラメータを無視します。

パラメータ	デフォルト	説明
Athena Query Run Time Schedule (Minute)	5	[Activate Scanner & Probe Protection] または [Activate HTTP Flood Protection] パラメータに `yes - Amazon Athena log parser` を選択した場合は、Athena クエリが実行される時間間隔 (分単位) を入力できます。デフォルトでは、Athena クエリは 5 分ごとに実行されます。 これらの保護を無効にすることを選択した場合は、このパラメータを無視します。

パラメータ	デフォルト	説明
ルールキー	IP	Activate HTTP Flood Protection パラメータに yes - AWS WAF rate-based rule を選択した場合は、他のさまざまな集計キーの組み合わせを使用するように、このルールを設定します。利用可能なオプション: IP (デフォルト) IP+カスタムヘッダー (このオプションを選択した場合、Rule Keys Custom Header は必須) IP+URI IP+HTTP メソッド 詳細については、「WAF rule rate based aggregation options」を参照してください。
ルールキーのカスタムヘッダー	no	Rule Keys パラメータに IP +Custom Header を選択した場合は、リクエストの集約に使用するカスタムヘッダーの名前を入力します。 詳細については、「WAF rule statement type rate based aggregation options」を参照してください。

パラメータ	デフォルト	説明
時間枠のしきい値 (分)	5	HTTP フラッド保護の時間枠のしきい値です。レートベースのルールと Lambda ログパーサーの両方に適用されます。使用可能なオプション: [1, 2, 5, 10]。 Activate HTTP Flood Protection パラメータに `yes - AWS WAF rate-based rule` を選択した場合、評価時間枠に使用されます。詳細については、「WAF web ACL rate based statement」を参照してください。 Activate HTTP Flood Protection パラメータに `yes - AWS Lambda log parser` を選択した場合、ブロック期間に加えて評価期間にも使用されます。
カスタムルール – 悪意のあるボット		
Activate Bad Bot Protection	`yes`	`yes` を選択すると、悪意のあるボットやコンテンツスクレイパーをブロックするように設計されたコンポーネントが有効になります。

パラメータ	デフォルト	説明
ARN of an IAM role that has write access to CloudWatch logs in your account	<オプション入力>	アカウントの CloudWatch ログへの書き込みアクセス権を持つ IAM ロールのオプション ARN を指定します。 例: `ARN: arn:aws:iam::account_id:role/myrolename` 。 このパラメータを空白 (デフォルト) のままにすると、ソリューションによって新しいロールが作成されます。
カスタムルール – サードパーティーの IP 評価リスト		
Activate Reputation List Protection	`yes`	`yes` を選択すると、サードパーティーの評価リスト (サポートされているリストは Spamhaus、Emerging Threats、Tor exit ノードを含みます) にある IP アドレスからのリクエストがブロックされます。
レガシーカスタムルール		

パラメータ	デフォルト	説明
Activate SQL Injection Protection	yes	yes を選択すると、一般的な SQL インジェクション攻撃をブロックするように設計されたコンポーネントが有効になります。AWS マネージドコアルールセットまたは AWS マネージド SQL データベースルールグループを使用していない場合は、有効にすることを検討してください。 また、8 KB (8,192 バイト) を超えるオーバーサイズのリクエストを AWS WAF に処理させるオプション (yes (continue)、yes - MATCH、または yes - NO_MATCH) の 1 つを選択することができます。デフォルトでは、yes はルールの検査基準に従ってサイズ制限内にあるリクエストコンポーネントのコンテンツを検査します。詳細については、「AWS WAF でのオーバーサイズウェブリクエストコンポーネントの処理」を参照してください。 この機能を無効にするには、no を選択します。注意 : CloudFormation スタックは、選択したオーバーサイズ処理オプションをデフォルトの SQL インジェクション保護ルールに追加し、AWS

パラメータ	デフォルト	説明
		アカウントにデプロイします。CloudFormation 以外でルールをカスタマイズした場合、変更内容はスタックの更新後に上書きされます。

パラメータ	デフォルト	説明
Sensitivity Level for SQL Injection Protection	LOW	AWS WAF が SQL インジェクション攻撃の検査に使用する感度レベルを選択します。 HIGH はより多くの攻撃を検出しますが、より多くの誤検出を生成する可能性があります。 LOW は、通常 SQL インジェクション攻撃に対する他の保護をすでに備えているリソースや、誤検知に対する許容度が低いリソースにとって、より適切な選択肢です。 詳細については、「AWS CloudFormation ユーザーガイド」の「AWS WAF で SQL インジェクションルールステートメントの感度レベルを追加」および「SensitivityLevel プロパティ」を参照してください。 SQL インジェクション保護を無効にする場合は、このパラメータを無視してください。注意: CloudFormation スタックは、選択した機密レベルをデフォルトの SQL インジェクション保護ルールに追加し、AWS アカウントにデプロイします。CloudFormation 以外でルールをカスタマイズした場合、変更内容はスタッ

パラメータ	デフォルト	説明
		クの更新後に上書きされます。

パラメータ	デフォルト	説明
Activate Cross-site Scripting Protection	yes	yes を選択すると、一般的な XSS 攻撃をブロックするように設計されたコンポーネントが有効になります。AWS マネージドコアルールセットを使用していない場合は、有効にすることを検討してください。また、8 KB (8192 バイト) を超えるオーバーサイズのリクエストを AWS WAF に処理させるオプション (yes (continue)、yes - MATCH、yes - NO_MATCH) の 1 つを選択することができます。デフォルトで、yes は Continue オプションを使用し、ルールの検査基準に従ってサイズ制限内にあるリクエストコンポーネントのコンテンツを検査します。詳細については、「AWS WAF でのオーバーサイズウェブリクエストコンポーネントの処理」を参照してください。 この機能を無効にするには、no を選択します。注意 : CloudFormation スタックは、選択した過剰サイズの処理オプションをデフォルトのクロスサイトスクリプティングルールに追加して、AWS アカウントにデプロイします。CloudFormation 以外でルールをカスタマイズした場

パラメータ	デフォルト	説明
		合、変更内容はスタックの更新後に上書きされます。
許可および拒否された IP 保持設定		
Retention Period (Minutes) for Allowed IP Set	-1	許可された IP セットの IP 保持を有効にする場合は、保持期間 (分) として数値 (15 以上) を入力します。保持期間に達する IP アドレスは期限切れになり、ソリューションは IP セットから IP アドレスを削除します。このソリューションは、最低 15 分の保持期間をサポートしています。0 ~ 15 の数値を入力すると、ソリューションはそれを 15 として扱います。 IP 保持をオフにするには、-1 (デフォルト) のままにします。

パラメータ	デフォルト	説明
Retention Period (Minutes) for Denied IP Set	-1	拒否された IP セットの IP 保持を有効にする場合は、保持期間 (分) として数値 (15 以上) を入力します。保持期間に達する IP アドレスは期限切れになり、ソリューションは IP セットから IP アドレスを削除します。このソリューションは、最低 15 分の保持期間をサポートしています。0 ~ 15 の数値を入力すると、ソリューションはそれを 15 として扱います。 IP 保持をオフにするには、-1 (デフォルト) のままにします。
Email for receiving notification upon Allowed or Denied IP Sets expiration	<オプション入力>	IP 保持期間パラメータ (前述の 2 つのパラメータを参照) を有効にして、IP アドレスの有効期限が切れたときに E メール通知を受信する場合は、有効な E メールアドレスを入力します。 IP 保持をアクティブにしなかった場合、または E メール通知を無効にする場合は、空白 (デフォルト) のままにしてください。
詳細設定		

パラメータ	デフォルト	説明
Retention Period (Days) for Log Groups	365	CloudWatch ロググループの保持を有効にする場合は、保持期間 (日数) として数値 (1 以上) を入力します。保持期間は 1 日 (1) から 10 年 (3650) の間で選択できます。デフォルトでは、ログは 1 年後に期限切れになります。 ログを無期限に保持するには、-1 に設定します。

6. [次へ] を選択します。
7. [スタックオプションの設定] ページでは、スタック内のリソースのタグ (キー値のペア) を指定し、追加オプションを設定できます。[次へ] を選択します。
8. [確認および作成] ページで、設定を確認して確定します。テンプレートによって IAM リソースと必要な追加機能が作成されることを承認するボックスを選択します。
9. [送信] を選択してスタックをデプロイします。

   AWS CloudFormation コンソールの [ステータス] 列でスタックのステータスを確認します。約 15 分で CREATE_COMPLETE ステータスが表示されます。

> **Note**
>
> このソリューションには、AWS Lambda 関数 `Log Parser` と `IP Lists Parser` に加えて、Lambda 関数 `helper` と `custom-resource` が含まれています。この 2 つの関数は、初期設定時、またはリソースの更新時や削除時にのみ実行されます。
> このソリューションを使用すると、AWS Lambda コンソールにすべての関数が表示されますが、3 つの主要なソリューション関数のみが定期的にアクティブになります。他の 2 つの関数は関連付けられたリソースを管理するために必要になるため削除しないでください。

スタックリソースの詳細を表示するには、[出力] タブを選択します。これには、BadBotHoneypotEndpoint 値が含まれます。この値を覚えておいてください。ウェブアプリケーションにハニーポットリンクを埋め込む際に使用します。

ステップ 2. ウェブ ACL をウェブアプリケーションに関連付ける

CloudFront ディストリビューションまたは ALB を更新して、「ステップ 1. スタックを起動する」で生成したリソースを使用して AWS WAF とロギングをアクティブにします。

1. AWS WAF コンソールにサインインします。
2. 使用したいウェブ ACL を選択します。
3. [Associated AWS resources (関連付けられた AWS リソース)] タブで [Add AWS resources (AWS リソースの追加)] を選択します。
4. [リソースタイプ] で、CloudFront ディストリビューションまたは ALB を選択します。
5. リストからリソースを選択し、[追加] を選択して変更を保存します。

ステップ 3. ウェブアクセスロギングを設定する

ウェブアクセスログを適切な Amazon S3 バケットに送信して、このデータを Log Parser Lambda 関数で使用できるように、CloudFront または ALB を設定します。

Amazon CloudFront ディストリビューションからのウェブアクセスログを保存する

1. Amazon CloudFront コンソールにサインインします。
2. ウェブアプリケーションのディストリビューションを選択し、[ディストリビューション設定] を選択します。
3. [全般] タブで、[編集] を選択します。
4. [AWS WAF ウェブ ACL] で、作成されたウェブ ACL ソリューション (スタック名パラメータ) を選択します。
5. [Logging] で、[On] を選択します。
6. [ログ用のバケット] で、ウェブアクセスログの保存に使用する S3 バケットを選択します。これは、メインスタックで使用され、CloudFront がログを書き込むアクセス許可を持つ新規または既存の S3 バケットにすることができます。ドロップダウンリストに、現在の AWS アカウントに関

連付けられているバケットが一覧表示されます。詳細については、Amazon CloudFront 開発者ガイドの「基本的な CloudFront ディストリビューションの開始方法」を参照してください。

7. ログプレフィックスを、ソリューションのデプロイに使用されるプレフィックスに設定します。プレフィックスは、メインスタックの [パラメータ] タブ、[AppAccessLogBucketPrefixParam] (デフォルト `AWSLogs/`) にあります。
8. [Yes, edit] を選択して変更を保存します。

詳細については、「Amazon CloudFront デベロッパーガイド」の「標準ログ (アクセスログ) の設定および使用」を参照してください。

Application Load Balancer からのウェブアクセスログを保存する

1. Amazon Elastic Compute Cloud (Amazon EC2) コンソールにログインします。
2. ナビゲーションペインで、[ロードバランサー] を選択します。
3. ウェブアプリケーションの ALB を選択します。
4. [Description] (説明) タブで、[Edit attributes] (属性の編集) を選択します。
5. [Enable access logs] を選択します。
6. [S3 location] に、ウェブアクセスログの保存に使用する S3 バケットの名前を入力します。これは、メインスタックで使用され、Application Load Balancer がログを書き込むアクセス許可を持つ新規または既存の S3 バケットにすることができます。
7. ログプレフィックスを、ソリューションのデプロイに使用されるプレフィックスに設定します。プレフィックスは、メインスタックの [パラメータ] タブ、[AppAccessLogBucketPrefixParam] (デフォルト `AWSLogs/`) にあります。
8. [保存] を選択します。

詳細については、「Elastic Load Balancing ユーザーガイド」の「Application Load Balancer のアクセスログ」を参照してください。

ソリューションを更新する

このソリューションを既にデプロイ済みの場合は、この手順に従ってソリューションの CloudFormation スタックを更新し、ソリューションのフレームワークの最新バージョンを取得します。スタックを更新する前に、「更新に関する考慮事項」を注意深くお読みください。

1. AWS CloudFormation コンソールにサインインします。
2. 左側のナビゲーションメニューで [スタック] をクリックします。
3. 既存の `aws-waf-security-automations` CloudFormation スタックを選択します。
4. [更新] を選択します。
5. [既存テンプレートを置き換える] を選択します。
6. [テンプレートを指定] で、以下を実行します。
   a. [Amazon S3 URL] を選択します。
   b. `aws-waf-security-automations.template` AWS CloudFormation のリンクをコピーします。
   c. [Amazon S3 URL] ボックスにリンクを貼り付けます。
   d. テンプレートの正しい URL が [Amazon S3 URL] テキストボックスに表示されていることを確認します。
   e. [次へ] を選択します。
   f. [次へ] をもう一度選択します。
7. [パラメータ] で、テンプレートのパラメータを確認し、必要に応じて変更します。パラメータの詳細については、「ステップ 1. スタックを起動する」を参照してください。
8. [次へ] を選択します。
9. [スタックオプションの設定] ページで、[次へ] を選択します。
10. [レビュー] ページで、設定を確認して確定します。
11. テンプレートによって IAM のリソースが作成されることを承認するチェックボックスをオンにします。
12. [変更セットの表示] を選択して、変更を確認します。
13. [スタックの更新] を選択してスタックをデプロイします。

AWS CloudFormation コンソールの [ステータス] 列でスタックのステータスを確認できます。約 15 分で UPDATE_COMPLETE のステータスが表示されます。

更新に関する考慮事項

次のセクションでは、このソリューションを実装するための制約と考慮事項について説明します。

リソースタイプの更新

スタックの作成後に Endpoint パラメータを更新するには、新しいスタックをデプロイする必要があります。スタックの更新時に Endpoint パラメータを変更しないでください。

WAFV2 の更新

バージョン 3.0 以降、このソリューションでは AWS WAFV2 をサポートしています。すべての AWS WAF Classic API コールは、AWS WAFV2 API コールに置き換えられました。Node.js への依存関係が削除され、最新の Python ランタイムが使用されています。このソリューションを最新の機能と改善とともに引き続き使用するには、バージョン 3.0 以降を新しいスタックとしてデプロイする必要があります。

スタック更新時のカスタマイズ

既成のソリューションでは、AWS CloudFormation スタックを介して、デフォルト設定の AWS WAF ルールセットを AWS アカウントにデプロイします。ソリューションによってデプロイされたルールにカスタマイズを適用することはお勧めしません。スタックの更新では、これらの変更を上書きします。カスタマイズされたルールが必要な場合は、このソリューションとは別に個別のルールを作成することをお勧めします。

不正なボットからの保護のアップグレード

バージョン 4.1.0 では、API Gateway を使用する Access Handler Lambda は廃止され、Log parser - Bad bot の拡張ログ機能に置き換えられました。API Gateway 経由で直接リクエストを使用する代わりに、ソリューションがログストリームを再利用して不正なボットを検出するようになりました。

以前の実装:

1. アクセスハンドラーとして Lambda と API Gateway が必要。
2. リクエストの直接処理にハニーポットエンドポイントを使用。
3. ウェブサイトにハニーポットエンドポイントの埋め込みが必要。

新しい実装 (4.1.0 以降): 現在は不正なボットからの保護ログパーサー:

1. ログを介してハニーポットエンドポイントへのリクエストを検査。
2. 不正なボットからの保護をアクティブ化したときにリクエストを処理。
3. WAF フィルター BadBotRuleFilter を使用して不正なボットリクエストを特定。
4. ログデータを分析して、定義したクォータを超える IP アドレスを特定。
5. AWS WAF IP セット条件を更新して、特定したアドレスをブロック。

この変更により、重複する機能を排除し、既存のログ処理機能を活用することで、アーキテクチャが簡素化されます。

CDK のアップグレード

バージョン v4.1.0 以降、このソリューションは CDK でサポートされています。v4.1.0 より前のバージョンから移行する場合。Cloudformation で新しいテンプレートと更新ソリューションを使用します。その後、cdk デプロイを使用してターミナル経由でソリューションの更新をローカルで開始できます (詳細については「README」を参照してください)。cdk デプロイを直接使用しようとすると、フロー収集のインデントが不十分というエラーが表示されることがあります。

ソリューションを更新するもう 1 つの方法は、ソリューションが提供するテンプレートを使用し、AWS コンソールの [Cloudformation] セクションに移動して [ソリューションの更新] をクリックし、そこに新しいテンプレートを貼り付けることです。

> **Note**
>
> このソリューションのバージョン 3.0 または 3.1 からバージョン 3.2 以降にアップグレードする場合で、許可または拒否された IP セットに IP アドレスを手動で挿入した場合、それらの IP アドレスが失われるリスクがあります。これを防ぐには、ソリューションをアップグレードする前に、許可または拒否された IP セット内の IP アドレスのコピーを作成します。アップグレードが完了したら、必要に応じて IP アドレスを IP セットに追加し直します。get-ip-set および update-ip-set CLI コマンドを参照してください。バージョン 3.2 以降をすでに使用している場合は、このステップを無視してください。

ソリューションをアンインストールする

ソリューションをアンインストールするには、CloudFormation でスタックを削除します。

1. AWS CloudFormation コンソールにサインインします。
2. ソリューションの親スタックを選択します。他のすべてのソリューションスタックは自動的に削除されます。
3. [削除] を選択します。

> **Note**
>
> このソリューションをアンインストールすると、Amazon S3 バケットを除いた、このソリューションで使用されている AWS リソースがすべて削除されます。AWS WAF API の制限が原因でレート超過のスロットリングの問題のために一部の IP セットが削除されない場合は、それらの IP セットを手動で削除してから、スタックを削除します。

ソリューションを使用する

このセクションでは、ソリューションをデプロイ後に使用する手順について詳しく説明します。

許可セットと拒否セットを変更する (オプション)

このソリューションの CloudFormation スタックをデプロイした後で、許可セットと拒否セットを手動で変更することにより、必要に応じて IP アドレスを追加または削除できます

1. AWS WAF コンソールにサインインします。
2. 左側のナビゲーションペインで、[IP セット] を選択します。
3. [許可リストに IP セット] を選択し、信頼できる送信元の IP アドレスを追加します。
4. [拒否リストに IP セット] を選択し、ブロックする IP アドレスを追加します。

ウェブアプリケーションにハニーポットリンクを埋め込む (オプション)

「ステップ 1. スタックを起動する」で Activate Bad Bot Protection パラメータで `yes` を選択した場合、CloudFormation テンプレートは、低インタラクションの本番稼働用ハニーポットにトラップエンドポイントを作成します。このトラップは、コンテンツスクレーパーや悪質なボットからのインバウンドリクエストを検出して迂回することを目的としています。有効なユーザーは、このエンドポイントにアクセスできません。

このコンポーネントは、ハニーポットメカニズムに加えて、Application Load Balancer (ALB) または Amazon CloudFront への直接接続をモニタリングすることで、不正なボットの検出を強化します。ボットがハニーポットをバイパスして ALB または CloudFront とやり取りしようとすると、システムはリクエストパターンとログを分析して悪意のあるアクティビティを特定します。不正なボットを検出すると、その IP アドレスを抽出して AWS WAF ブロックリストに追加し、以降のアクセスを防ぎます。不正なボット検出は、構造化されたロジックチェーンを介して動作し、脅威に対して包括的に対応します。

- HTTP フラッド保護 Lambda ログパーサー – フラッド分析中にログエントリから不正なボットの IP を収集します。
- スキャナーとプローブ保護 Lambda ログパーサー – スキャナー関連のログエントリから不正なボットの IP を特定します。

- HTTP フラッド保護 Athena ログパーサー – クエリ実行全体でパーティションを使用して、Athena ログから不正なボットの IP を抽出します。
- スキャナーとプローブ保護 Athena ログパーサー – 同じパーティショニング戦略を使用して、スキャナー関連の Athena ログから不正なボットの IP を取得します。
- フォールバック検出 - HTTP フラッド保護およびスキャナーとプローブ保護の両方が無効になっている場合、システムは Log Lambda パーサーにより、WAF ラベルフィルターに基づいてボットアクティビティをログに記録します。

以下の手順のいずれかを使用して、CloudFront ディストリビューションからのリクエストのハニーポットリンクを埋め込みます。

ハニーポットエンドポイント用の Amazon CloudFront オリジンを作成する

CloudFront ディストリビューションでデプロイされるウェブアプリケーションには、この手順を使用します。CloudFront では、`robots.txt` ファイルを含めることで、ロボット排除規約を無視するコンテンツスクレイパーやボットを識別できます。次の手順に従い、非表示のリンクを埋め込み、`robots.txt` ファイルで明示的に禁止します。

1. AWS CloudFormation コンソールにサインインします。
2. 「ステップ 1 スタックを起動する」で構築したスタックを選択します。
3. [出力] タブを選択します。
4. [BadBotHoneypotEndpoint] キーから、エンドポイント URL をコピーします。
   - 動作パス (`/ProdStage`)
5. ハニーポットを指すコンテンツにこのエンドポイントリンクを埋め込みます。このリンクを人間のユーザーから隠します。例として、次のコードサンプルを参照してください: `<a href="/behavior_path" rel="nofollow" style="display: none" aria-hidden="true">honeypot link</a>`。
6. ウェブサイトのルートにある `robots.txt` ファイルを変更して、次のようにハニーポットリンクを明示的に禁止します。

```
User-agent: <*>
        Disallow: /<behavior_path>
```

> **Important**
>
> リクエストは WAF BadBotRuleFilter によってブロックされるため、CloudFront でのパス登録は必要ありません。ログで自動的に収集されるソリューション。ログパーサー Lambda による処理。この簡略化されたアプローチでは、追加のエンドポイント設定を必要とせずに WAF ログを直接使用するため、ログ分析による不正なボット検出プロセスがより効率的になります。

> **Note**
>
> どのタグ値がウェブサイト環境で機能するかを確認するのはユーザーの責任です。ご使用の環境でタグ値を監視しない場合は、`rel="nofollow"` を使用しないでください。ロボットのメタタグ設定の詳細については、「Google 開発者ガイド」を参照してください。ウェブサイトのルートにある `robots.txt` ファイルを変更して、次のようにハニーポットリンクを明示的に禁止します。

ハニーポットエンドポイントを外部リンクとして埋め込む

> **Note**
>
> これらのルールは、ウェブリクエストの発信元のソース IP アドレスを使用します。トラフィックが 1 つ以上のプロキシまたはロードバランサーを通過する場合、ウェブリクエストの発信元には、クライアントの発信アドレスではなく、最後のプロキシのアドレスが含まれます。

次の手順は、ウェブアプリケーションに使用します。

1. AWS CloudFormation コンソールにサインインします。
2. 「ステップ 1 スタックを起動する」で構築したスタックを選択します。
3. [出力] タブを選択します。
4. [BadBotHoneypotEndpoint] キーから、エンドポイント URL をコピーします。

```
<a href="<BadBotHoneypotEndpoint value>" rel="nofollow" style="display: none" aria-hidden="true"><honeypot link></a>
```

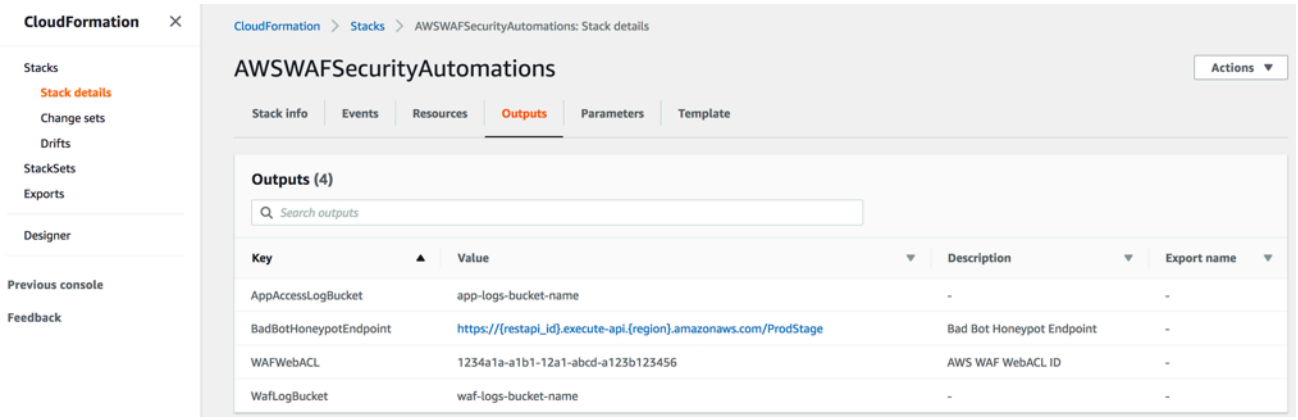
> **Note**
>
> この手順では、rel=nofollow を使用して、ハニーポット URL にアクセスしないようロボットに指示します。ただし、リンクは外部に埋め込まれているため、robots.txt ファイルを含めてリンクを明示的に禁止することはできません。どのタグがウェブサイト環境で機能するかを確認するのはユーザーの責任です。ご使用の環境で rel="nofollow" を監視しない場合は、使用しないでください。

Lambda ログパーサーの JSON ファイルを使用する

HTTP フラッド保護に Lambda ログパーサー JSON ファイルを使用する

Activate HTTP Flood Protection テンプレートパラメータで Yes - AWS Lambda log parser を選択した場合、このソリューションは <stack_name>-waf_log_conf.json という名前の設定ファイルを作成し、AWS WAF ログファイルの保存に使用される Amazon S3 バケットにアップロードします。バケット名を確認するには、CloudFormation 出力の WafLogBucket 変数を参照してください。次の図はその一例です。

AWSWAFSecurityAutomations というラベルの画面を示し、4 つの出力を一覧表示するスクリーンショット



Amazon S3 で <stack_name>-waf_log_conf.json ファイルを編集して上書きすると、Log Parser Lambda 関数は新しい AWS WAF ログファイルを処理するときに新しい値を考慮します。以下は、サンプルの設定ファイルです。

サンプル設定ファイルのスクリーンショット

```
{
 "general": {
    "requestThreshold": 2000,
    "blockPeriod": 240,
    "ignoredSufixes": [".css", ".js", ".jpg", "png", ".gif"]
  },
 "uriList": {
    "/search": {
        "requestThreshold": 500,
        "blockPeriod": 600
    }
  }
}
```

パラメータには、以下が含まれます。

- 全般:
  - リクエストしきい値 (必須) – IP アドレスごとに 5 分あたりの最大許容リクエスト数。このソリューションは、CloudFormation スタックをプロビジョニングまたは更新する時に定義した値を使用します。
  - ブロック期間 (必須) – 該当する IP アドレスをブロックする期間 (分単位)。このソリューションは、CloudFormation スタックをプロビジョニングまたは更新する時に定義した値を使用します。
  - 無視されたサフィックス – このタイプのリソースにアクセスするリクエストは、リクエストのしきい値にはカウントされません。デフォルトでは、このリストは空です。
- URI リスト – これを使用して、特定の URL のカスタムリクエストのしきい値およびブロック期間を定義します。デフォルトでは、このリストは空です。

WAF ログが WafLogBucket に到着すると、設定ファイルの設定を使用して Lambda ログパーサー関数によって処理されます。このソリューションは、同じバケット内の `<stack_name>-waf_log_out.json` という名前の出力ファイルに結果を書き込みます。出力ファイルに攻撃者として特定された IP アドレスのリストが含まれている場合、ソリューションはそれらを HTTP Flood の WAF IP セットに追加し、アプリケーションへのアクセスをブロックします。出力ファイルに IP アドレスがない場合は、設定ファイルが有効かどうか、または設定ファイルに従ってレート制限を超えたかどうかを確認します。

スキャナーとプローブ保護に Lambda ログパーサー JSON ファイルを使用する

Activate Scanner & Probe Protection テンプレートパラメータで `Yes - AWS Lambda log parser` を選択した場合、このソリューションは設定ファイル `<stack_name>-app_log_conf.json` を作成し、CloudFront または Application Load Balancer のログファイルを保存するのに使用される定義済みの Amazon S3 バケットにアップロードします。

Amazon S3 で `<stack_name>-app_log_conf.json` を編集して上書きすると、`Log Parser` Lambda 関数は新しい AWS WAF ログファイルを処理するときに新しい値を考慮します。以下は、サンプルの設定ファイルです。

設定ファイルのスクリーンショット

```
{
    "general": {
        "errorThreshold": 50,
        "blockPeriod": 240,
        "errorCodes": ["400", "401", "403", "404", "405"]
    },
    "uriList": {
        "/login": {
            "errorThreshold": 5,
            "blockPeriod": 600
        },
        "/api/feedback": {
            "errorThreshold": 10,
            "blockPeriod": 240
        }
    }
}
```

パラメータには、以下が含まれます。

- 全般:
  - エラーしきい値 (必須) – 各 IP アドレスが 1 分あたりに許容する不正なリクエストの数。このソリューションは、CloudFormation スタックをプロビジョニングまたは更新する時に定義した値を使用します。
  - ブロック期間 (必須) – 該当する IP アドレスをブロックする期間 (分単位)。このソリューションは、CloudFormation スタックをプロビジョニングまたは更新する時に定義した値を使用します。
  - エラーコード – エラーとみなされるステータスコードです。デフォルトでは、リストは次の HTTP ステータスコードをエラーと見なします。`400 (Bad Request)`、`401 (Unauthorized)`、`403 (Forbidden)`、`404 (Not Found)`、`405 (Method Not Allowed)`。

- URI リスト – これを使用して、特定の URL のカスタムリクエストのしきい値およびブロック期間を定義します。デフォルトでは、このリストは空です。

アプリケーションアクセスログが AppAccessLogBucket に到着すると、`Log Parser` Lambda 関数は設定ファイルの設定を使用してログを処理します。このソリューションは、同じバケット内の `<stack_name>`-app_log_out.json` という名前の出力ファイルに結果を書き込みます。出力ファイルに攻撃者として識別される IP アドレスのリストが含まれている場合、ソリューションはそれらをスキャナーとプローブの WAF IP セットに追加し、アプリケーションへのアクセスをブロックします。出力ファイルに IP アドレスがない場合は、設定ファイルが有効かどうか、または設定ファイルに従ってレート制限を超えたかどうかを確認します。

HTTP フラッドの Athena ログパーサーで国と URI を使用する

Athena クエリで国と URI とともに IP 別にグループ化し、予測不可能な URI パターンを持つ HTTP フラッド攻撃を検出してブロックできます。これを行うには、スタックの起動時に Group By Requests in HTTP Flood Athena Query パラメータのオプション (`Country`、`URI`、`Country and URI`) のいずれかを選択します。

国別のリクエストしきい値は、Request Threshold by Country パラメータを使用して入力することもできます。例えば、`{"TR": 50,"ER":150}`。このソリューションは、これらの指定された国から発信されたリクエストにこれらのしきい値を使用します。このソリューションは、他の国からのリクエストにデフォルトのしきい値を使用します。

> **Note**
>
> 国別にしきい値を定義すると、ソリューションには Athena クエリの group-by 句にその国が自動的に含まれます。詳細については、「ステップ 1 スタックを起動する」のパラメータテーブルを参照してください。

このソリューションは、デフォルトで 5 分間のリクエストしきい値をカウントします。これは、Athena Query Run Time Schedule (Minute) パラメータで設定できます。

> **Note**
>
> Athena クエリは、リクエストしきい値を期間で割って 1 分あたりのしきい値を計算します。例えば、次のようになります。
> Request threshold (default threshold or threshold by country): 100

Athena Query Run Time Schedule: 5
Request threshold per minute: 20 = 100 / 5

Amazon Athena クエリを表示する

Activate HTTP Flood Protection または Activate Scanner & Probe Protection テンプレートパラメータに `Yes - Amazon Athena log parser` を選択した場合、このソリューションは CloudFront または ALB (`ScannersProbesLogParser`) または AWS WAF ログ (`HTTPFloodLogParser`) の Athena クエリを作成して実行し、出力を解析し、それに応じて AWS WAF を更新します。

パフォーマンスを向上させ、コストを低く抑えるために、ソリューションはファイル名のタイムスタンプに基づいてログを分割します。このソリューションは、パーティションキー (年、月、日、時間) を使用する Athena クエリを動的に生成します。デフォルトでは、クエリは 5 分ごとに実行されます。Athena Query Run Time Schedule (Minute) テンプレートパラメータの値を変更することで、実行スケジュールを設定できます。各クエリの実行は、デフォルトで最後の 4 ~ 5 時間のデータをスキャンします。WAF Block Period テンプレートパラメータの値を変更することで、クエリがスキャンするデータ量を設定できます。また、このソリューションでは、クエリを別々のワークグループに配置して、クエリのアクセスとコストを管理します。

> **Note**
>
> Athena が AWS Glue データカタログにアクセスするように設定されていることを確認します。このソリューションは、AWS Glue でアクセスログのデータカタログを作成して、そのデータを処理するように Athena クエリを設定します。Athena が正しく設定されていないと、クエリは実行されません。詳細については、「Amazon Athena とは」を参照してください。

次の手順に従って、これらのクエリを更新します。

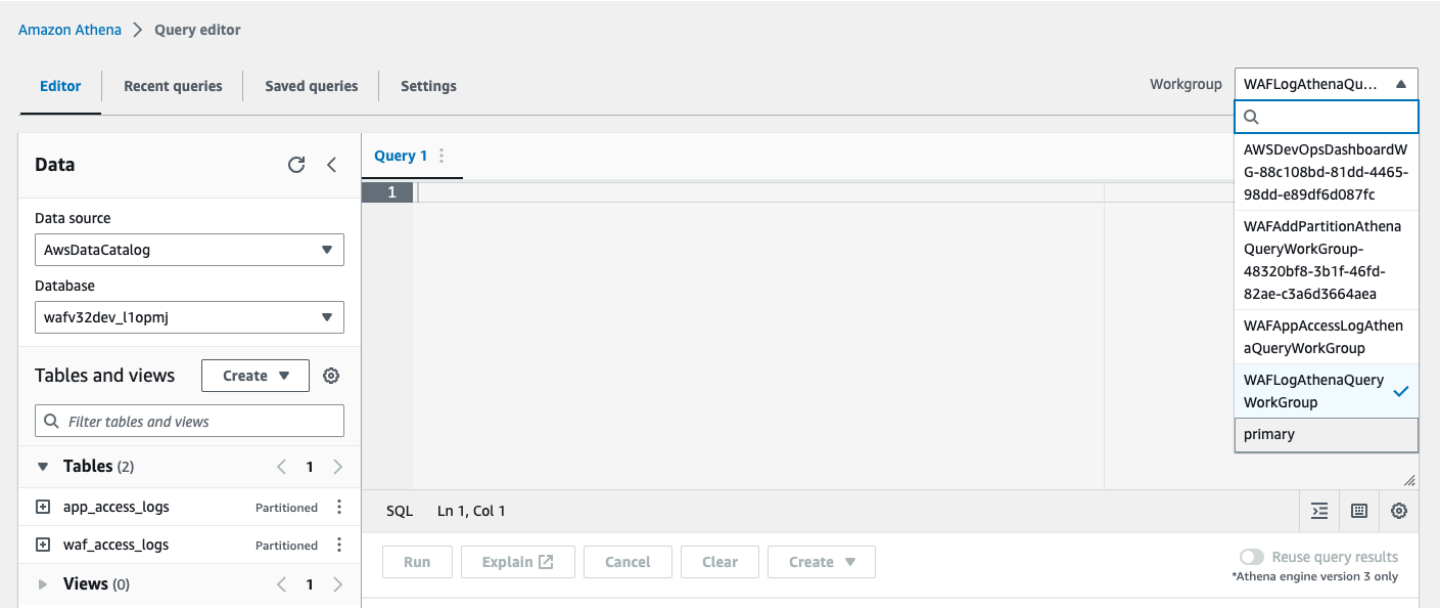
WAF ログクエリを表示する

1. Amazon Athena コンソールにサインインします。
2. [クエリエディタの起動] を選択します。
3. このソリューションのデータベースを選択します。
4. ドロップダウンリストから [WAFLogAthenaQueryWorkGroup] を選択します。

> **Note**
>
> このワークグループは、Activate HTTP Flood Protection テンプレートパラメータに `Yes - Amazon Athena log parser` を選択した場合にのみ存在します。

5. ワークグループを切り替えるには、[切り替え] を選択します。

クエリがないことを示す Athena クエリエディタのスクリーンショット



1. [履歴] タブを選択します。
2. リストから [SELECT クエリ] を選択して開きます。

アプリケーションアクセスログクエリを表示する

1. Amazon Athena コンソールにサインインします。
2. [ワークグループ] を選択します。
3. リストから [WAFAppAccessLogAthenaQueryWorkGroup] を選択します。

> **Note**
>
> このワークグループは、Activate Scanner & Probe Protection テンプレートパラメータで `Yes - Amazon Athena log parser` を選択した場合にのみ存在します。

4. [ワークグループの切り替え] を選択します。
5. [最近のクエリ] タブを選択します。
6. リストから [SELECT クエリ] を選択して開きます。

Athena パーティションクエリの追加を表示する

1. Amazon Athena コンソールにサインインします。
2. [ワークグループ] を選択します。
3. リストから [WAFAddPartitionAthenaQueryWorkGroup] を選択します。

> **Note**
>
> このワークグループは、Activate HTTP Flood Protection または Activate Scanner & Probe Protection のテンプレートパラメータで `Yes - Amazon Athena log parser` を選択した場合にのみ存在します。

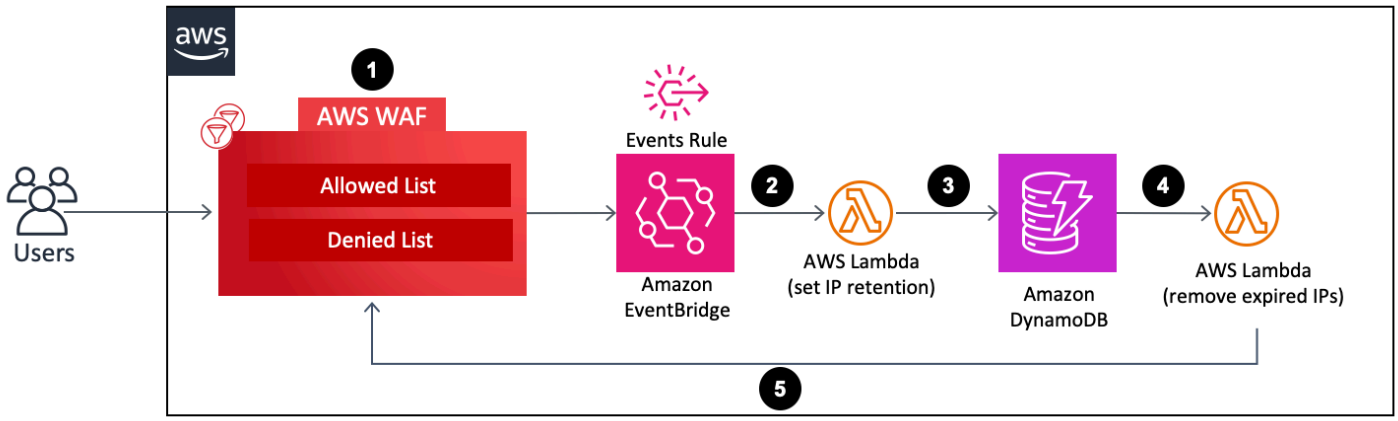
4. [ワークグループを切り替える] を選択します。
5. [履歴] タブを選択します。
6. リストから [ALTER TABLE クエリ] を選択して開きます。これらのクエリは 1 時間ごとに実行され、Athena テーブルに新しい時間ごとのパーティションが追加されます。

許可および拒否された AWS WAF IP セットの IP 保持を設定する

このソリューションが作成する許可および拒否された AWS WAF IP セットの IP 保持を設定できます。次のセクションでは、その仕組みを説明し、設定する手順を説明します。

仕組み

AWS WAF の許可リストと拒否リスト、およびその他の AWS リソースを示すアーキテクチャ図



1. ユーザーが、許可または拒否された AWS WAF の IP セットを更新 (IP アドレスを追加または削除) すると、このアクションは、 AWS WAF の `UpdateIPSet` API コールを呼び出して、イベントを作成します。
2. Amazon EventBridge イベントルールは、事前定義されたイベントパターンに基づいてイベントを検出し、Lambda 関数を呼び出して、更新後に IP セットに存在するすべての IP アドレスの保持期間を設定します。
3. Lambda 関数はイベントを処理し、IP 保持に関連するデータを (IP セット名、ID、スコープ、IP アドレスなど) を抽出し、DynamoDB テーブルに挿入します。また、各 DynamoDB の項目に対して `ExpirationTime` 属性も挿入されます。このソリューションは、ユーザー定義の保存期間をイベント時間に加算して有効期限を計算します。テーブルには DynamoDB Streams があり、Time to Live (TTL) が有効になっています。TTL 属性は `ExpirationTime` です。
4. 項目が有効期限に達すると、TTL が呼び出され、Amazon DynamoDB は有効期限が切れた後にその項目をテーブルから削除します。項目を削除すると、削除された項目が DynamoDB ストリームに追加され、ダウンストリーム処理のために Lambda 関数が呼び出されます。
5. Lambda 関数は、DynamoDB ストリームから削除された項目に関する情報を取得し、AWS WAF の API コールを行って、項目に含まれる期限切れの IP アドレスをターゲットの AWS WAF の IP セットから削除します。

IP 保持を有効にする

IP 保持を有効にするには、次の手順に従います。

1. デプロイまたは更新する Cloudformation スタックで、IP Retention Period (Minutes) for Allowed IP Set とIP Retention Period (Minutes) for Denied IP Set を入力します。最小保持期間は 15 分です。このソリューションは、`0` と 15 の間の任意の数を 15 として扱います。デプロイ設定の詳細

については、「ステップ 1. スタックを起動する」で生成したリソースを使用して AWS WAF と口ギングをアクティブにします。

2. 期限切れの IP アドレスが AWS WAF の IP セットから削除された場合に E メール通知を受け取りたい場合は、E メールアドレスを入力します。E メール通知の受信を選択した場合は、このソリューションが正常にデプロイされた後に受信する E メール内のリンクを使用して、サブスクリプションを確認する必要があります。デプロイ設定の詳細については、「ステップ 1. スタックを起動する」で生成したリソースを使用して AWS WAF と口ギングをアクティブにします。
3. IP アドレスを追加または削除して AWS WAF IP セットを更新します。これにより、IP 保持プロセスが開始され、IP 有効期限リストを含む DynamoDB 項目が作成されます。この有効期限リストは、更新後に AWS WAF IP セットに存在する IP アドレスで構成されます。
4. DynamoDB 項目が有効期限に達し、テーブルから削除されると、ソリューションは項目の IP 有効期限リストに含まれている IP アドレスを WAF IP セットから削除します。

> **Note**
>
> DynamoDB が TTL によって期限切れの項目を削除する時間に応じて、AWS WAF IP セットから期限切れ IP アドレスの実際の削除操作が変わる場合があります。DynamoDB の TTL による削除は、主にテーブルのサイズとアクティビティレベルに依存します。DynamoDB の削除操作では遅延が発生する可能性があるため、AWS WAF の削除操作に遅延が発生することが予想されます。一般的に、このソリューションは DynamoDB TTL を削除した直後に AWS WAF IP セットから期限切れの IP アドレスを削除します。詳細については、「Amazon DynamoDB 開発者ガイド」の「DynamoDB での Time to Llive (TTL) の使用」を参照してください。

モニタリングダッシュボードを構築する

AWS では、重要なエンドポイントごとにカスタムのベースラインモニタリングシステムを設定することをお勧めしています。カスタマイズされたメトリクスビューの作成と使用の詳細については、「CloudWatch Dashboards – Create & Use Customized Metrics Views」および「Amazon CloudWatch ダッシュボードの使用」を参照してください。

次のダッシュボードスクリーンショットは、カスタムのベースラインモニタリングシステムの例を示しています。

CloudFront ダッシュボードのスクリーンショット

ダッシュボードには、以下のメトリクスが表示されます。

- Allowed vs Blocked Requests – 許可されたアクセス (通常のピークアクセスの 2 倍) またはブロックされたアクセス (ブロックされた 1K を超えるリクエストを識別する期間) が急増したかどうかが表示されます。CloudWatch は Slack チャネルにアラートを送信します。このメトリクスを使用して、既知の DDoS 攻撃 (ブロックされたリクエストが増加した場合) や新しいバージョンの攻撃 (リクエストがシステムへのアクセスを許可されている場合) を追跡するために使用できます。

> **Note**
>
> 注記: ソリューションではこのメトリクスを提供しています。

- BytesDownloaded vs Uploaded – 通常は大量のアクセスを受け取らないサービスが DDoS 攻撃の標的になっていないかを特定するのに役立ちます (例えば、ある特定のリクエストパラメータセットに関する MB の情報を送信する検索エンジンコンポーネントなど)。
- ELB Spillover and Queue length – DDoS 攻撃がインフラストラクチャにダメージを与えており、攻撃者が CloudFront または AWS WAF レイヤーをバイパスし、保護されていないリソースを直接攻撃しているかどうかを確認するのに役立ちます。
- ELB Request Count – インフラストラクチャへのダメージを特定するのに役立ちます。このメトリクスは、攻撃者が保護レイヤーをバイパスしているかどうか、またはキャッシュヒット率を上げるために CloudFront キャッシュルールを確認する必要があるかどうかを示します。

- ELB Healthy Host – これを別のシステムヘルスチェック指標として使用できます。
- ASG CPU Utilization – 攻撃者が CloudFront、AWS WAF、Elastic Load Balancing をバイパスしているかどうかを特定するのに役立ちます。このメトリクスを使用して、攻撃のダメージを特定するためにも使用できます。

XSS フォールスポジティブを処理する

このソリューションでは、受信リクエストでよく調べられる要素を検査して XSS 攻撃を特定してブロックする AWS WAF ルールを設定します。この検出パターンは、コンテンツ管理システムのリッチテキストエディタなどで HTML を作成し送信することをユーザーに許可しているワークロードでは、あまり効果的ではありません。このシナリオでは、リッチテキスト入力を受け入れる特定の URL パターンに対してデフォルトの XSS ルールをバイパスする例外ルールを作成し、除外された URL を保護する代替のメカニズムを実装することを検討してください。

さらに、一部の画像またはカスタムデータ形式では、HTML コンテンツに潜在的な XSS 攻撃を示すパターンが含まれているため、誤検出が発生する可能性があります。例えば、SVG ファイルには `<script>` タグが含まれる場合があります。このタイプのコンテンツが正当なユーザーから得られることが予想される場合は、これらの他のデータ形式を含む HTML リクエストを許可するように XSS ルールを細かく調整します。

次の手順に従い、HTML をインプットとして受け入れる URL を除外するために XSS ルールを更新します。詳細な手順については、「Amazon WAF 開発者ガイド」を参照してください。

1. AWS WAF コンソールにサインインします。
2. 文字列一致または正規表現条件を作成します。
3. URI を検査し、XSS ルールに対して受け入れる値をリストするようにフィルター設定を構成します。
4. このソリューションの XSS ルールを編集し、作成した新しい条件を追加します。

   例えば、リスト内のすべての URL を除外するには、When a request に対して以下を選択します。

   - does not
   - match at least one of the filers in the string match condition
   - XSS Allowlist

トラブルシューティング

このソリューションに関するヘルプが必要な場合は、サポートに連絡して、このソリューションに関するサポートケースを開いてください。

AWS サポートに問い合わせる

AWS ビジネスサポート+、AWS エンタープライズサポート、または Unified Operations をご利用の場合は、AWS サポートセンターを利用して、このソリューションに関する エキスパートのサポートを受けることができます。次のセクションで、その方法を説明します。

ケースを作成する

1. サポートセンターを開きます。
2. [ケースを作成] を選択します。

どのようなサポートをご希望ですか?

1. [技術] を選択します。
2. [サービス] で、[ソリューション] を選択します。
3. [カテゴリ] で、[AWS WAF のセキュリティオートメーション] を選択します。
4. [重要度] で、ユースケースに最も適したオプションを選択します。
5. [サービス]、[カテゴリ]、[重要度] を入力すると、インターフェイスに一般的なトラブルシューティングの質問へのリンクが表示されます。これらのリンクを使用しても問題を解決できない場合は、[次のステップ: 追加情報] を選択してください。

追加情報

1. [件名] に、質問または問題を要約したテキストを入力します。
2. [説明] で、このソリューションの名前と使用しているバージョン (例: AWS WAF バージョン X.Y.Z でのセキュリティオートメーション) を含めて、問題を詳しく説明します。
3. [ファイルを添付] を選択します。
4. リクエストを処理するためにサポートに必要な情報を添付します。

ケースの迅速な解決にご協力ください

1. 必要な情報を記入します。
2. [次のステップ: 今すぐ解決またはお問い合わせ] を選択します。

今すぐ解決またはお問い合わせ

1. [今すぐ解決] で解決策を確認します。
2. これらの解決策で問題を解決できない場合は、[お問い合わせ] を選択し、必要な情報を入力して [送信] を選択します。

デベロッパーガイド

このセクションでは、ソリューションのソースコードを提供します。

ソースコード

GitHub リポジトリにアクセスして、このソリューションのテンプレートとスクリプトをダウンロードし、カスタマイズした上で他のユーザーと共有できます。

このソリューションのテンプレートは AWS CDK を使用して生成します。詳細については、README.md ファイルを参照してください。

参照資料

このセクションには、このソリューション固有のメトリクスを収集するためのオプション機能、関連リソースへのポインタ、このソリューションに貢献したビルダーのリストに関する情報が含まれています。

匿名化されたデータの収集

このソリューションには、運用メトリクスを AWS に送信するオプションが含まれています。AWS ではこのデータを使用して、ユーザーがこのソリューション、関連サービスおよび製品をどのように使用しているかをよりよく理解し、提供するサービスや製品の改善に役立てます。有効にすると、このソリューションは次の情報が収集し、CloudFormation テンプレートの初回デプロイ時に AWS に送信します。

- Solution ID - AWS ソリューションの識別子
- Unique ID (UUID) – このソリューションのデプロイごとにランダムに生成された一意の識別子
- Timestamp - データ収集タイムスタンプ
- Solution configuration – 有効化された機能と初期起動時に設定されるパラメータ
- Lifecycle – このソリューションを使用した期間 (スタックの削除に基づいた)
- Log parser data:
  - ブロックするスキャナーとプローブの IP セット、不正なボットの IP セット、HTTP フラッドの IP セットの IP アドレス数
  - 処理およびブロックされたリクエストの数
- IP lists parser data:
  - 評価リスト IP セット内の IP アドレスの数
  - 処理およびブロックされたリクエストの数
- IP retention data – 許可または拒否された IP セットから削除される期限切れの IP アドレスの数

AWS は、このアンケートを通じて収集されたデータを所有します。データ収集には、AWS プライバシーポリシーが適用されます。この機能をオプトアウトするには、AWS CloudFormation テンプレートを起動する前に次の手順を実行します。

1. `aws-waf-security-automations.template` AWS CloudFormation をローカルハードドライブにダウンロードします。

2. テキストエディタで CloudFormation テンプレートを開きます。
3. CloudFormation テンプレートのマッピングセクションを次のように変更します。

```
Solution:
  Data:
   SendAnonymizedUsageData: "Yes"
```

変更後:

```
Solution:
  Data:
   SendAnonymizedUsageData: "No"
```

4. AWS CloudFormation コンソールにサインインします。
5. [スタックの作成] を選択します。
6. [スタックの作成] ページの [テンプレートの指定] セクションで、[テンプレートファイルのアップロード] を選択します。
7. [テンプレートファイルのアップロード] で、[ファイルの選択] を選択し、ローカルドライブから編集したテンプレートを選択します。
8. [次へ] を選択し、「ステップ 1 スタックを起動する」の手順に従います。

関連リソース

関連する AWS ホワイトペーパー

- DDoS に対する回復性に関する AWS のベストプラクティス

関連する AWS セキュリティブログ記事

- AWS WAF、Amazon CloudFront、Referer Checking を使用してホットリンクを防止する方法

サードパーティーの IP 評価リスト

- Spamhaus DROP List website
- Proofpoint Emerging Threats IP list

- Tor exit node list

寄稿者

- Heitor Vital
- Lee Atkinson
- Ben Potter
- Vlad Vlasceanu
- Aijun Peng
- Chaitanya Deolankar
- Shu Jackson
- William Quan
- Mykhailo Markhain

リビジョン

GitHub リポジトリの CHANGELOG.md にアクセスして、バージョン固有の改善と修正を追跡します。

注意

この実装ガイドは情報提供のみを目的としています。本書の発行時点における AWS の現行製品と慣行を表したものであり、それらは予告なく変更されることがあります。お客様は本書の情報、および AWS 製品またはサービスの利用について、独自の評価に基づき判断する責任を負います。いずれの AWS 製品またはサービスも、明示または黙示を問わずいかなる保証も伴うことなく、「現状のまま」提供されます。本書のいかなる内容も、AWS、その関係者、サプライヤー、またはライセンサーからの保証、表明、契約的責任、条件や確約を意味するものではありません。お客様に対する AWS の責任は AWS 契約によって規定されています。また、本文書は、AWS とお客様との間の契約に属するものではなく、また、当該契約が本文書によって修正されることもありません。

AWS WAF のセキュリティオートメーションソリューションは、Apache ライセンスバージョン 2.0 の条件に基づいてライセンスされています。