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

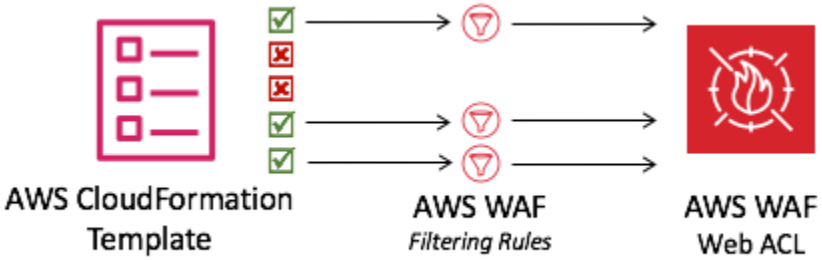
Table of Contents

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

公開日: 2016 年 9 月 (最終更新日: 2024 年 12 月)

Security Automations for AWS WAF 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に影響を与えたり、セキュリティを侵害したり、過剰なリソースを消費したりす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)、Amazon API Gateway 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) に対して許可またはブロックするトラフィックを制御します。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」、「AWS Shield Advanced デベロッパーガイド」の「AWS WAF」を参照してください。

AWS WAF ルールの設定は、大規模な組織にとっても小規模な組織にとっても、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、困難で負担がかかる可能性があります。このプロセスを簡素化するために、Security Automations for AWS WAF 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に設計された一連の AWS WAF ルールを含む単一のウェブアクセスコントロールリスト (ACL) が自動的にデプロイされます。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、含める保護機能を指定できます。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 (複数可) または ALB (複数可) へのウェブリクエストを検査し、必要に応じてブロックします。



AWS WAF ウェブ ACL の設定

この実装ガイドでは、Amazon Web Services (AWS) クラウド上にこのソリューションをデプロイするためのアーキテクチャ上の考慮事項、設定手順、および運用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 のセキュリティ、コンピューティング、ストレージ、その他さまざまなサービスを起動、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。

このガイドの情報は、AWS WAF、CloudFront、ALB、AWS Lambda などの AWS サービスに関する実用的な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サービス AWS WAF 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、デベロッパー、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、以下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。

目的	参照先
このソリューションの運用コストを確認する。 このソリューションを実行するための総コストは、アクティブ化された保護と、取り込まれ、保存され、処理されたデータの量によって異なります。	コスト

目的	参照先
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
このソリューションでサポートされている AWS リージョンを把握します。	サポートされる 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ます。	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、Support を使用します。	Support
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

特徴と利点

Security Automations for AWS WAF 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 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 マネージドルール for AWS WAF 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 Flood 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

Scanners & Probes カスタムルールは、オリジンによって生成された異常な量のエラーなど、疑わしい動作を検索するアプリケーションアクセスログを解析します。次に、お客様が定義した期間、疑わしい送信元 IP アドレスをブロックします。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 カスタムルールは、攻撃の試みを誘発し、回避す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。ウェブサイトにエンドポイントを挿入して、コンテンツスクレイパーや不正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) リストと拡張 DROP (EDROP) リスト、Proofpoint Emerging Threats IP リスト、Tor exit ノード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の構築」を参照してください。

Service Catalog AppRegistry および AWS Systems Manager Application Manager との統合

このソリューションには、AWS Service Catalog AppRegistry および AWS Systems Manager Application Manager の両方でアプリケーションとして、CloudFormation テンプレートと基礎となるリソースを登録するための Service Catalog AppRegistry リソースが含まれています。この統合により、ソリューションのリソースを一元管理できます。

ユースケース

公開日: 2016 年 9 月 (最終更新日: 2023 年 5 月)

このソリューションを使用する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、このリストに限定されない革新的な方法でカスタマイズできます。

AWS WAF ルールの設定を自動化する

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 HTTP Flood 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。

ソースコードを活用してカスタマイズを適用したり、独自のセキュリティオートメーションを構築する

このソリューションでは、AWS WAF および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を使用すると、カスタマイズを適用したり、ニーズに合わせて独自のセキュリティオートメーションを構築したりできます。

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 Scanner & Probe Protection 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、HTTP Flood Protection ルールまたは Scanner & Probe Protection ルールをアクティブ化するときにユーザーが `yes - Amazon Athena log parser` を選択した場合、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。

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。Lamba 関数は、HTTP Flood Protection ルールまたは Scanner & Probe Protection ルールをアクティブ化するときに、ユーザーが `yes - AWS Lambda log parser` を選択した場合、AWS WAF、CloudFront、または ALB ログの検査を開始します。

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、事前定義されたすぐに使用できるルールのコレクションで、AWS および AWS Marketplace 販売者が記述して管理します。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、API Gateway、ALB、AWS AppSync、Amazon Cognito、AWS App Runner、AWS Verified Access 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口グを使用します。このソリューションの HTTP Flood Protection ルールは、これらの口グを検査します。

WCU

AWS WAF は、ウェブアクセスコントロールリスト (ACL) 容量ユニット (WCU) を使用して、ルール、ルールグループ、およびウェブ ACL の実行に必要なオペレーティングリソースを計算および制御します。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。

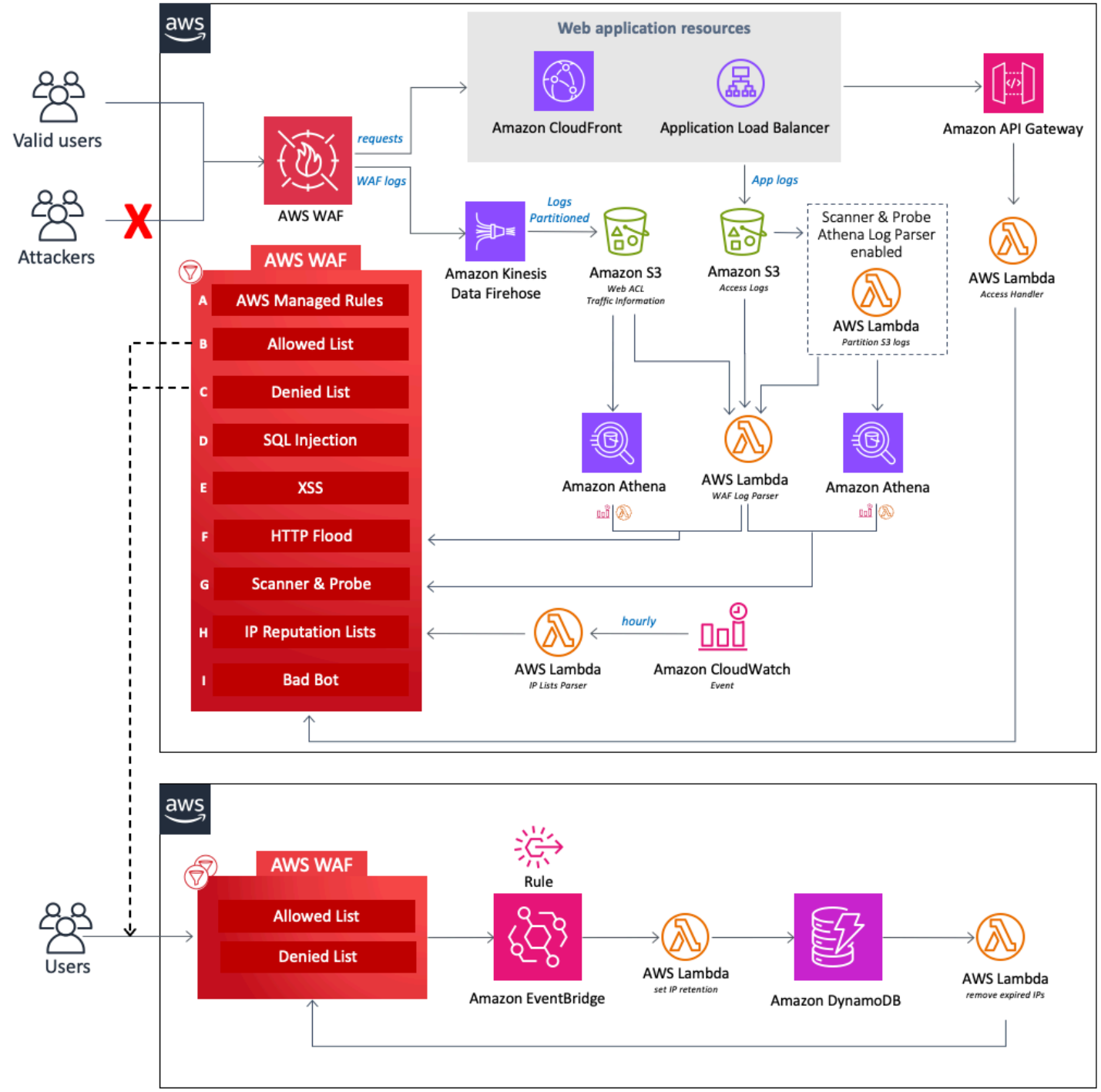
> **Note**
>
> AWS の用語の一般的なリファレンスについては、「AWS 用語集」を参照してください。

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デプロイ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。



AWS 上の AWS WAF アーキテクチャ向けセキュリティオートメーション

設計の中核となるのは、AWS WAF ウェブ ACL です。これは、ウェブアプリケーションへのすべての受信リクエストを一元的に検査および決定するポイントとして機能します。CloudFormation スタックの初期設定時に、ユーザーはアクティブ化する保護コンポーネントを定義します。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、次の保護領域にグループ化できます。

> **Note**
>
> グループラベルは、WAF ルールの優先度レベルを反映しません。

- AWS Managed Rules (A) – このコンポーネントには、AWS マネージドルール IP レピュテーションルールグループ、ベースラインルールグループ、ユースケース固有のルールグループが含まれます。これらのルールグループは、独自のルールを記述することなく、OWASP の出版物に記載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不要なトラフィックの悪用から保護します。
- Manual IP lists (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 Injection (D) および XSS (E) – これらのコンポーネントは、URI、クエリ文字列、またはリクエスト本文の一般的な SQL インジェクションまたは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に設計された 2 つの AWS WAF ルールを設定します。
- HTTP Flood (F) – このコンポーネントは、ウェブレイヤーの DDoS 攻撃や総当たりのログイン試行など、特定の IP アドレスからの多数のリクエストで構成される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、その 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 フラッド緩和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。
- Scanner and Probe (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。次に、お客様が定義した期間、それらの疑わしいソース IP アドレスをブロックします。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 Reputation Lists (H) – このコンポーネント `IP Lists Parser` Lambda 関数で、サードパーティーの IP レピュテーションリストを 1 時間ごとにチェックして、ブロックする新しい範囲を探

します。これらのリストには、Spamhaus Don't Route Or Peer (DROP) リストと Extended DROP (EDROP) リスト、Proofpoint Emerging Threats IP リスト、Tor 出口ノードリストが含まれます。

- Bad Bot (I) – このコンポーネントは、攻撃の試みを誘惑し、そらす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を自動的にセットアップします。このソリューションのハニーポットは、コンテンツスクレイパーや不正なボットからのインバウンドリクエストを検出するためにウェブサイトに挿入することができるトラップエンドポイントです。ソースがハニーポットにアクセスすると、`Access Handler` Lambda 関数がリクエストをインターセプトして検査し、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。

Well-Architected 設計上の考慮事項

このソリューションでは、AWS Well-Architected Framework のベストプラクティスを使用しています。これにより、ユーザーは信頼性、セキュリティ、効率、コスト効果が高いワークロードを設計し、クラウドで運用できます。

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、運用上の優秀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、インフラストラクチャ、Lambda 関数、Amazon Data Firehose、API Gateway、Amazon S3 バケット、およびその他のソリューションコンポーネントのオブザーバビリティを提供するために、メトリクスを CloudWatch にプッシュします。
- 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、セキュリティ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。

- 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。
- ソリューションで使用されるすべてのロールは、最小特権アクセスに従います。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 バケットや DynamoDB を含むすべてのデータストレージは、保管時の暗号化を使用します。

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 サーバーレスサービス (Firehose、API Gateway、Amazon S3、Athena など) を可能な限り使用して、高可用性とサービス障害からの回復を確保します。
- エラーを迅速に検出して修正するために、ソリューションに対して自動テストを実行します。
- 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では、Amazon S3 と DynamoDB にデータを保存するため、デフォルトで複数のアベイラビリティーゾーンに保持されます。

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およびデプロイされます。当社のソリューションアーキテクトと対象分野の専門家によって、実験と改善の余地がある領域についてソリューションのレビューが行われます。

コスト最適化

このセクションでは、コスト最適化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サーバーレスアーキテクチャを使用し、お客様への請求は使用した分に対してのみ行われます。
- ソリューションのコンピューティングレイヤーはデフォルトで Lambda に設定されます。Lambda は従量課金制モデルを使用します。
- 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、コストを削減できます。

持続可能性

このセクションでは、持続可能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、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、オンプレミスサーバーを継続的に運用する場合のフットプリントと比較して、カーボンフットプリントを低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の AWS 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、Cloud Front、および ALB ログを保存します。	
AWS Lambda	Core。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	

AWS のサービス	説明	
	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 API Gateway	サポート。不正なボットのハニーポットエンドポイントを作成します。	
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モニタリングと、リソースオペレーションおよび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」で説明されているように、HTTP フラッドとスキャナーおよび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では、継続的に更新される 5 分間にクライアント IP が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 リクエストを超えており、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勧めします。例えば、リクエストをカウントするときに静的リソースアクセスは考慮しません。

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は、[Athena Log Parser] オプションを提供します。アクティブ化すると、CloudFormation は Athena クエリと、Athena の実行、結果出力の処理、AWS WAF の更新を調整する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の詳細については、「Amazon Athena クエリの表示」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。WAF アクセスログタイプは遅延時間が短いため、CloudFront または ALB ログ配信時間と比較して、HTTP フラッドの発生元をより迅速に特定できます。ただし、応答ステータスコードを受け取るには、[Activate Scanner & Probe Protection] テンプレートパラメータで CloudFront または ALB ログタイプを選択する必要があります。

AWS Lambda ログパーサー

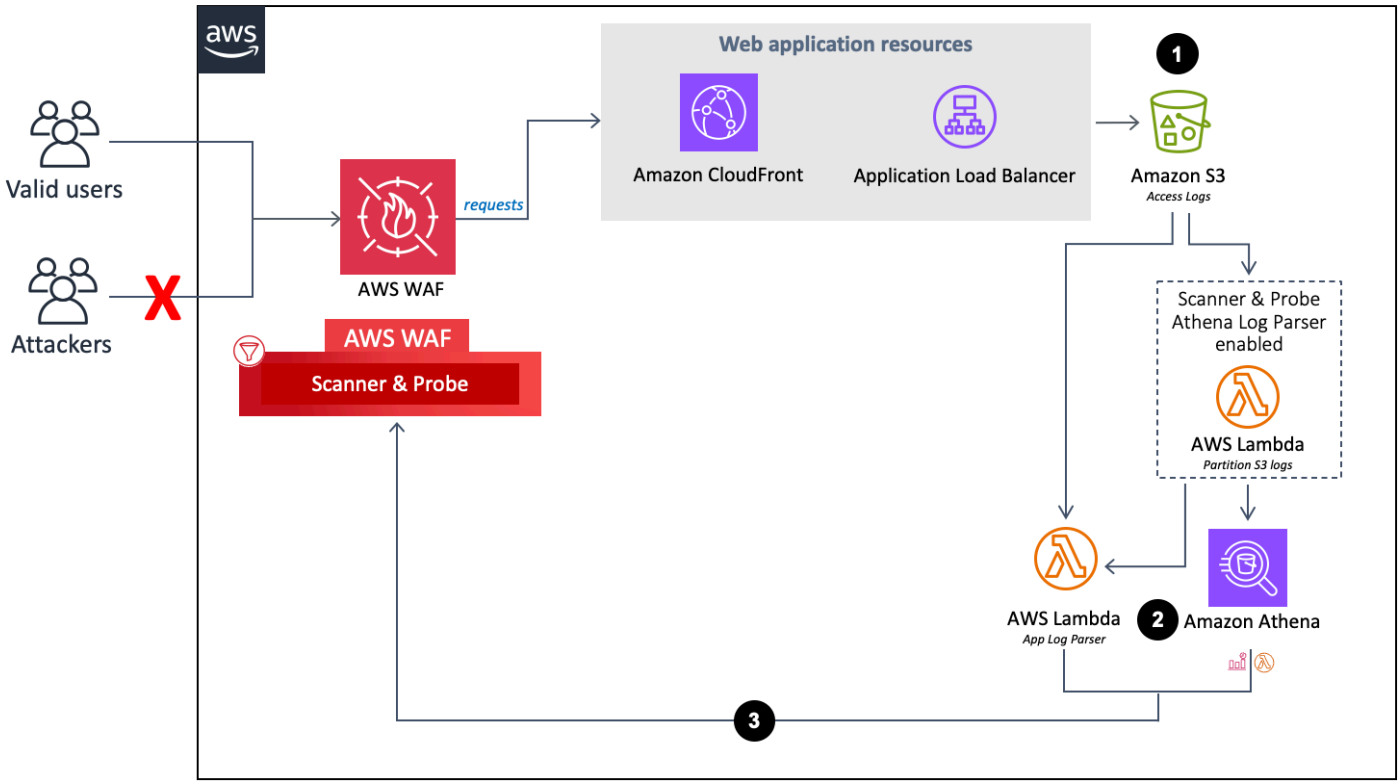
[HTTP Flood Protection] および [Scanner & Probe Protection] テンプレートパラメータは、[AWS Lambda Log Parser] オプションを提供します。Lambda ログパーサーは、AWS WAF レートベースのルールおよび Amazon Athena ログパーサーオプションが利用できない場合にのみ使用します。このオプションの既知の制限は、処理中のファイルのコンテキスト内で情報が処理されることです。例えば、IP は定義されたクォータを超えるリクエストまたはエラーを生成する可能性がありますが、この情報は異なるファイルに分割されているため、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。

コンポーネントの詳細

アーキテクチャ図で説明されているように、このソリューションの 4 つのコンポーネントは自動化を使用して IP アドレスを検査し、AWS WAF ブロックリストに追加します。以下のセクションでは、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、スキャナーやプローブからの保護に役立ちます。



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、Amazon S3 バケットにアクセスログを送信します。
   a. (オプション) [Activate HTTP Flood Protection] および [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、Lambda 関数はアクセスログを元のフォルダ *`<customer-bucket>`*`/AWSLogs` から、Amazon S3 に到着すると新しくパーティション化されたフォルダ *`<customer-bucket>`*`/AWSLogs-partitioned/`*`<optional-prefix>`*` /year=`*`<YYYY>`*`/month=`*`<MM>`*` /day=`*`<DD>`*`/hour=`*`<HH>`*`/` に移動します。

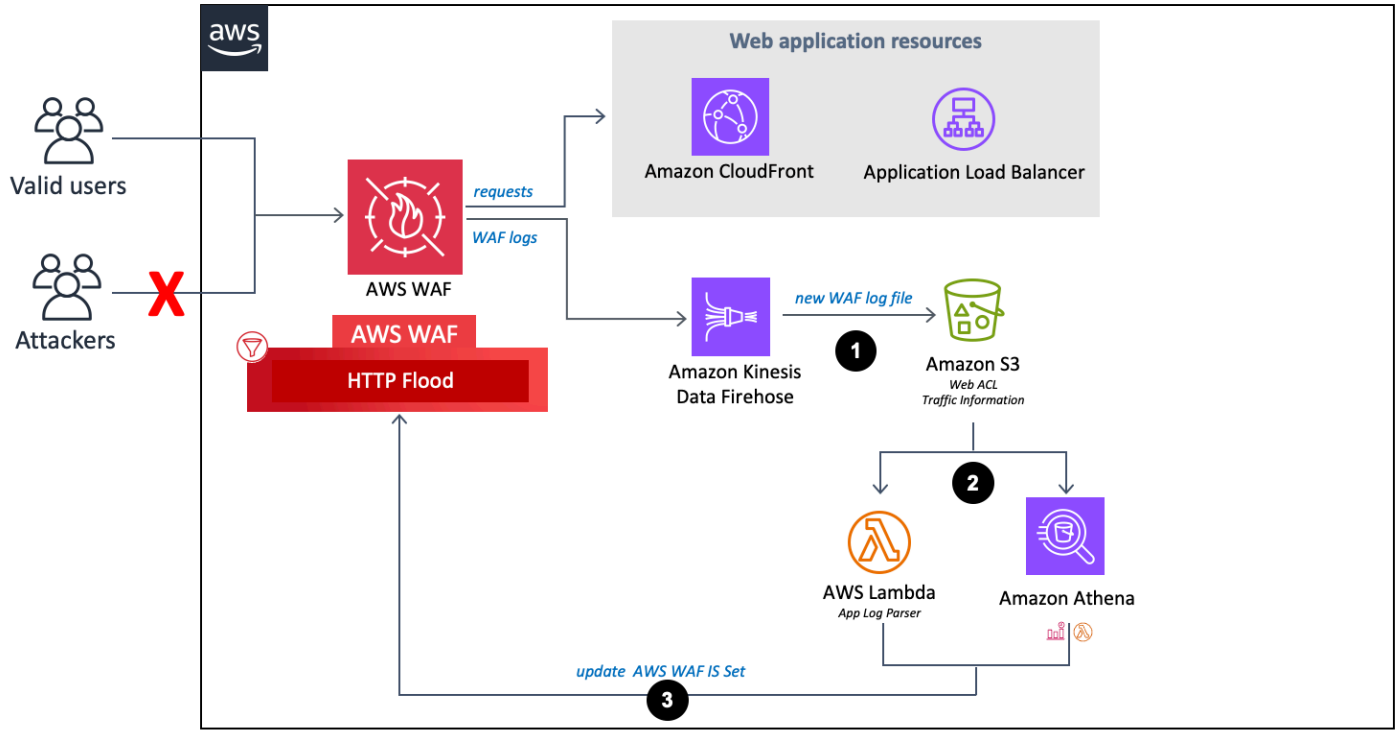
b. (オプション) [Keep Data in Original S3 location] テンプレートパラメータで `yes` を選択した場合、ログは元の場所に残り、パーティション化されたフォルダにコピーされ、ログストレージが複製されます。

> **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をデプロイした後に Amazon S3 バケットに到着した新しいログのみをパーティション化します。パーティション化する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、それらのログを Amazon S3 に手動でアップロードする必要があります。

2. [Activate HTTP Flood Protection] および [Activate Scanner & Probe Protection] テンプレートパラメータの選択に基づいて、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。これにより、Athena クエリの実行を担当する Lambda 関数が開始され、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] に `yes - AWS Lambda log parser` または `yes - Amazon Athena log parser` を選択した場合、このソリューションでは次のコンポーネントがプロビジョニングされます。これらのコンポーネントは AWS WAF ログを解析し、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。

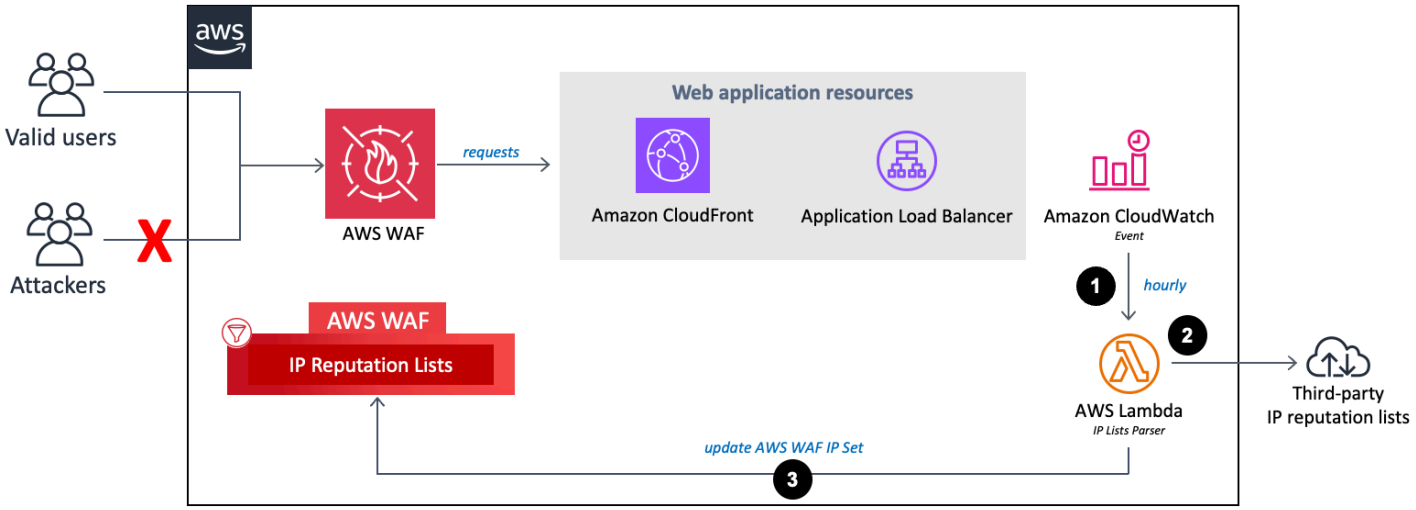


AWS WAF ログパーサーフロー

1. AWS WAF が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。次に、Firehose はログを *<customer-bucket>*/AWSLogs/ *<optional-prefix>*/year=*<YYYY>* /month=*<MM>*/day=*<DD>*/hour= *<HH>*/ という名前の Amazon S3 のパーティション化されたバケットに配信します。
2. [Activate HTTP Flood Protection] および [Activate Scanner & Probe Protection] テンプレートパラメータの選択に基づいて、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、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、出力は 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、Amazon Athena クエリの実行を担当する Lambda 関数を起動し、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 レピュテーションリストで特定された既知の攻撃者から保護するのに役立ちます。

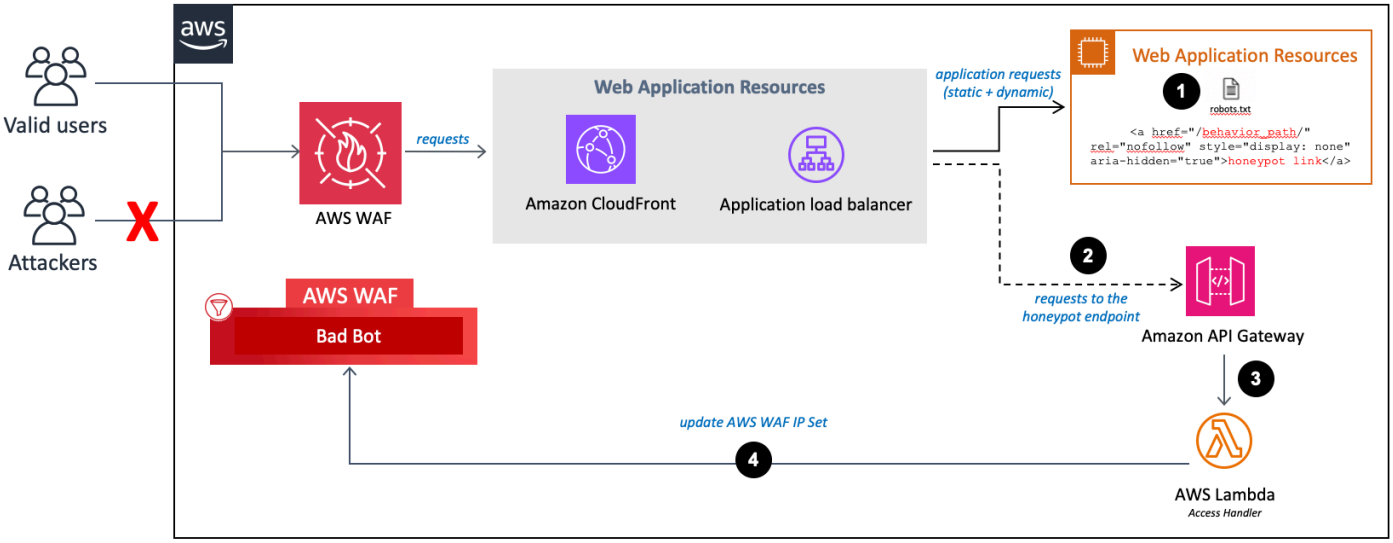


IP レピュテーションリストパーサーフロー

1. 1 時間ごとの Amazon CloudWatch イベントが `IP Lists Parser` Lambda 関数を呼び出します。
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 および EDROP リスト
   - Proofpoint Emerging Threats IP リスト
   - Tor 出口ノードリスト
3. Lambda 関数は、現在の IP アドレスを使用して AWS WAF ブロックリストを更新します。

アクセスハンドラー

`Access Handler` Lambda 関数は、ハニーポットエンドポイントへのリクエストを検査し、送信元 IP アドレスを抽出します。



アクセスハンドラーとハニーポットエンドポイント

1. 「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)」の説明に従って、ウェブサイトにハニーポットエンドポイントを埋め込み、ロボットの除外標準を更新します。
2. コンテンツスクレイパーまたは不正なボットがハニーポットエンドポイントにアクセスすると、`Access Handler` Lambda 関数が呼び出されます。
3. Lambda 関数は、リクエストヘッダーをインターセプトして検査し、トラップエンドポイントにアクセスしたソースの IP アドレスを抽出します。
4. Lambda 関数は AWS WAF IP セット条件を更新して、それらの IP アドレスをブロックします。

デプロイを計画する

このセクションでは、コスト、セキュリティ、the section called “クォータ”、およびソリューションのデプロイ前のその他の考慮事項について説明します。

サポートされている 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は異なるリソースが必要です。これらのリソース (次の表を参照) は、すべての AWS リージョンで使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。AWS のサービスのリージョンごと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)	✓			
HTTP Flood 保護を有効にする				
はい - AWS Lambda ログパーサー				✓
はい - Amazon Athena ログパーサー		✓	✓	✓
Scanner & Probe 保護を有効にする				
はい - Amazon Athena ログパーサー		✓	✓	

> **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。

コスト

Security Automations for AWS WAF 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総コストは、アクティブ化された保護と、取り込まれ、保存され、処理された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各 AWS のサービスの料金ウェブページを参照してください。

次の表は、米国東部 (バージニア北部) リージョン (AWS 無料利用枠を除く) でこのソリューションを実行する場合のコスト内訳の例です。料金は変更されることがあります。

例 1: 評価リスト保護、Bad Bot 保護、HTTP Flood 保護の AWS Lambda ログパーサー、Scanner & Probe 保護を有効にする

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: Lambda 実行あたり 3 つの関数、100 万回の呼び出し、平均 500 ミリ秒の期間 512 MB: Lambda 実行あたり 2 つの関数、100 万回の呼び出し、平均 500 ミリ秒の期間	~ 5.40 USD
Amazon API Gateway	100 万回のリクエスト	~ 3.40 USD
AWS WAF ウェブ ACL	1	5.00 USD

AWS のサービス	ディメンション/月	コスト [USD]
AWS WAF ルール	4	4.00 USD
AWS WAFリクエスト	+1M	0.60 USD
合計		1 か月あたり ~23.60 USD

例 2: 評価リスト保護、Bad Bot 保護、HTTP Flood 保護の Amazon Athena ログパーサー、Scanner & Probe 保護を有効にする

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: Lambda 実行あたり 3 つの関数、100 万回の呼び出し、平均 500 ミリ秒の期間 512 MB: Lambda 実行あたり 2 つの関数、7560 回の呼び出し、平均 500 ミリ秒の期間	~1.26 USD
Amazon API Gateway	100 万回のリクエスト	~3.40 USD
Amazon Athena	CloudFront オブジェクトが 1 日あたり 120 万件、または 1 日あたり 120 万件の ALB リクエストで、ヒットまたはリクエストごとに最大 500 バイトのログレコードが生成されます。	~4.32 USD
AWS WAF ウェブ ACL	1	5.00 USD
AWS WAF ルール	4	4.00 USD

AWS のサービス	ディメンション/月	コスト [USD]
AWS WAFリクエスト	+1M	0.60 USD
合計		1 か月あたり ~23.78 USD

例 3: 許可された IP セットと拒否された IP セットの IP 保持を有効にする

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: Lambda 実行あたり 1 つの関数、2,000 回の呼び出し、平均 500 ミリ秒の期間 512 MB: Lambda 実行あたり 1 つの関数、2,000 回の呼び出し、平均 500 ミリ秒の期間	~0.01 USD
Amazon CloudWatch	2,000 回のイベント	~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	+1M	0.60 USD
合計		1 か月あたり ~7.61 USD

CloudWatch ログのコスト見積もり

Lambda など、このソリューションで使用される一部の AWS サービスでは、CloudWatch ログが生成されます。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。ログアーカイブの詳細については、「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 ログパーサーを使用することを選択した場合、このソリューションでは、設定に従って Amazon S3 バケット (複数可) 内の AWS WAF またはアプリケーションアクセスログに対して実行されるクエリが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を管理するように両方を設定できます。サンプルコスト見積り計算については、「Athena のコスト見積り」を参照してください。詳細については、「Amazon Athena 料金」を参照してください。

Athena のコスト見積もり

HTTP Flood 保護ルールまたは Scanner & Probe 保護ルールの実行中に Athena ログパーサーオプションを使用すると、Athena の使用に対して課金されます。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、各 CloudFront ヒットは、約 500 バイトのデータを生成する可能性があります。
> 1 日あたり 120 万の CloudFront オブジェクトがヒットした場合、データが一貫したレートで取り込まれると仮定すると、4 時間あたり 20 万 (120 万/6) ヒットになります。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 は、スキャンされたデータ 1 TB あたり 5.00 USD を請求します。
> `[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 料金」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では、AWS が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セキュリティに至るまでのコンポーネントを運用、管理、および制御するため、お客様の運用上の負担を軽減するのに役立ちます。AWS の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 内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権限を割り当てることができます。このソリューションでは、最小特権を持つ IAM ロールが作成され、これらのロールはソリューションのリソースに必要なアクセス許可を付与します。

[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、サーバーをコントロールしたりするために特別に作成されたリクエスト、ウェブサイトを破壊するために設計されたボリューム攻撃、またはウェブコンテンツをスクレイプして盗むようにプログラムされた不正なボットやスクレイパーが含まれます。

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 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、次の一般的な攻撃をブロックします。

- AWS マネージドルール – これは、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するマネージドサービスです。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、悪意のある可能性のある 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 – 攻撃者は悪意のあるクライアントサイトスクリプトを他の正当なユーザーのウェブブラウザに挿入するための手段として、悪意のないウェブサイトの脆弱性を利用します。これは、受信リクエストの一般的に調査される要素を検査し、XSS 攻撃を識別してブロック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は、クライアントからのウェブリクエストが設定可能なクォータを超えると、レートベースのルールを自動的に呼び出します。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 – 悪意のあるソースは、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、インターネット向けウェブアプリケーションの脆弱性をスキャンしてプローブします。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では、CloudFront または ALB アクセスログを自動的に解析する Lambda 関数または Athena クエリを作成し、1 分あたりの一意の送信元 IP アドレスからの不正なリクエストの数をカウントし、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ために AWS WAF を更新します。
- 既知の攻撃者のオリジン (IP 評価リスト) – 多くの組織は、スパム送信者、マルウェアディストリビューター、ボットネットなどの既知の攻撃者が運用する 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、悪意のある 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 – パブリックにアクセス可能なウェブアプリケーションのオペレーターは、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していると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不正なボットなどの一部の自動化クライアントは、制限を回避するために自分自身を誤って表現します。このソリューションは、不正なボットやスクレイパーを特定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 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に記載されているすべての AWS のサービスのサービスクォータを確認するには、PDF の「サービスエンドポイントとクォータ」ページにある情報を表示してください。

AWS WAF のクォータ

AWS WAF は、IP 一致条件ごとに Classless Inter-Domain Routing (CIDR) 表記で最大 10,000 個の 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 のクォータ」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 セットを作成します。1 つは IPv4 用、もう 1 つは IPv6 用です。

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、AWS リージョンごとに 1 秒あたり最大 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

以下のセクションでは、このソリューションを実装する際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に包括的な保護を提供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として `yes` を選択します。パラメータのリストの「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 トラフィックのログ記録

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、エンドポイントを CloudFront として設定する場合は、Activate HTTP Flood 保護を 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 トラフィックのログ記録の詳細については、「AWS WAF デベロッパーガイド」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 (続行) – ルールの検査基準に従って、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 では、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が検査されます。これはソリューションで使用され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 では、ルールの検査基準に対してリクエストを評価せずに、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 – ウェブリクエストを、ルールの検査基準に対して評価せずに、ルールステートメントと一致しないものとして処理します。AWS WAF では、一致しないルールの場合と同様に、ウェブ ACL 内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の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のデプロイ

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。一意のデプロイ

ごとに追加料金が発生し、リージョンごとにアカウントごとの AWS WAF のクォータが適用されます。

ソリューションをデプロイする

このソリューションでは、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、このガイドで説明されている、アーキテクチャや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**
>
> 以前にこのソリューションをデプロイしたことがある場合、更新手順については「ソリューションを更新する」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のを設定する
- ALB を設定する

Step 1. スタックを起動する

- AWS アカウント に CloudFormation テンプレートを起動します。
- 必須パラメータの値として、[スタック名] と [アプリケーションアクセスログバケット名] を入力します。
- 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。

Step 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 (複数可) または ALB (複数可) 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。必要な数のディストリビューションまたはロードバランサーを関連付けることができます。

ステップ 3 ウェブアクセスログ記録を設定する

- 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、適切な Amazon S3 バケットにログファイルを送信します。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 AWS Logs (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 [アプリケーションアクセスログバケットプレフィックス] パラメータ 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。CloudFormation テンプレートは、ソリューションをデプロイする前に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。テンプレートはニーズに基づいてカスタマイズできます。

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。

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のを設定する

ウェブアプリケーションの静的コンテンツと動的コンテンツ用に CloudFront ディストリビューションを設定するには、次のステップを実行します。詳細な手順については、「Amazon CloudFront デベロッパーガイド」を参照してください。

1. CloudFront ウェブアプリケーション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の作成」を参照してください。
2. 静的オリジンと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 を設定する

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

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

1. AWS Management Console にサインインし、[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) リージョン (`us-east-1`) にデプロイする必要があります。

   > **Note**
   >
   > 定義した入力パラメータ値に応じて、このソリューションには異なるリソースが必要です。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、「サポートされている AWS リージョン」を参照してください。

3. [テンプレートを指定] 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] ページで、[スタック名] フィールドに AWS WAF 設定の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
スタックの名前	*<requires input>*	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

パラメータ	デフォルト	説明
		レートが作成するウェブ ACL の名前です。
リソースタイプ		
エンドポイント	`CloudFront`	使用するリソースのタイプを選択します。 **Note** エンドポイントとして `CloudFront` を選択した場合は、ソリューションを起動して、米国東部 (バージニア北部) リージョン (`us-east-1`) に WAF リソースを作成する必要があります。
AWS マネージド IP 評価ルールグループ		

パラメータ	デフォルト	説明
Amazon IP 評価リストマネージドルールグループを有効にする	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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
匿名 IP リストマネージドルールグループの保護を有効にする	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 ルールグループのリスト」を参照してください。
AWS マネージドベースラインルールグループ		

パラメータ	デフォルト	説明
コアルールセットマネージドルールグループの保護を有効にする	no	コアルールセッ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、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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
管理者保護マネージドルールグループの保護を有効にする	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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
既知の不正な入力マネージドルールグループの保護を有効にする	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 ルールグループのリスト」を参照してください。
AWS マネージド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
SQL データベースマネージドルールグループの保護を有効にする	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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
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の保護を有効にする	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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
POSIX オペレーティングシステムマネージドルールグループの保護を有効にする	no	コアルールセット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 ルールグルー

パラメータ	デフォルト	説明
		プのリスト」を参照してください。
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の保護を有効にする	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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
PHP アプリケーションマネージドルールグループの保護を有効にする	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 ルールグループのリスト」を参照してください。

パラメータ	デフォルト	説明
WordPress アプリケーションマネージドルールグループの保護を有効にする	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 ルールグループのリスト」を参照してください。
カスタムルール – Scanner & Probes		

パラメータ	デフォルト	説明
Scanner & Probe 保護を有効にする	`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されるコンポーネントを選択します。緩和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
アプリケーションアクセスログバケット名	*<requires input>*	[Scanner & Probe 保護] 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するのと同じ AWS リージョンに配置する必要があります。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。 **Note** 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

パラメータ	デフォルト	説明
		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、「アプリケーションアクセスログバケットプレフィックス」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
アプリケーションアクセスログバケットプレフィックス	AWSLogs/	[Scanner & Probe 保護] 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。 [エンドポイント] 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。 [エンドポイント] 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
バケットアクセスのログ記録はオンになっていますか	no	[アプリケーションアクセスログバケット名] 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。 [Scanner & Probe 保護を有効にする] 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。
エラーのしきい値	50	[Scanner & Probe 保護を有効にする] 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。 [Scanner & Probe 保護を有効にする] 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
元の S3 ロケーションにデータを保持する	no	[Scanner & Probe 保護を有効にする] 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、ログストレージが重複します。 [Scanner & Probe 保護を有効にする] 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 Flood		
HTTP Flood 保護を有効にする	`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されるコンポーネントを選択します。緩和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
デフォルトのリクエストしきい値	100	[HTTP Flood 保護を有効にする] 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。 [HTTP Flood 保護を有効にする] 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 100 です。 [HTTP Flood 保護を有効にする] 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
国別のリクエストしきい値	<オプション入力>	[HTTP Flood 保護を有効にする] 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 [デフォルトのリクエストしきい値] パラメータを使用します。 **Note** このパラメータを定義すると、[HTTP Flood Athena クエリでのリクエストによるグループ化] 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。

パラメータ	デフォルト	説明
		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。
HTTP Flood Athena クエリでのリクエストによるグループ化	None	[HTTP Flood 保護を有効にする] 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。

パラメータ	デフォルト	説明
WAF Block Period	240	[Scanner & Probe 保護を有効にする] または [HTTP Flood 保護を有効にする] 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。
Athena クエリ実行時間スケジュール (分)	5	[Scanner & Probe 保護を有効にする] または [HTTP Flood 保護を有効にする] 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。
カスタムルール - Bad Bot		

パラメータ	デフォルト	説明
Bad Bot 保護を有効にする	yes	不正な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をオンにするには、yes を選択します。
アカウント内の CloudWatch ログへの書き込み権限がある IAM ロールの ARN	<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 。ロールの作成方法については、「API Gateway での CloudWatch による REST API の口グの設定」を参照してください。 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。

パラメータ	デフォルト	説明
デフォルトのリクエストしきい値	100	[HTTP Flood 保護を有効にする] 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。 [HTTP Flood 保護を有効にする] 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 100 です。 [HTTP Flood 保護を有効にする] 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。
カスタムルール – サードパーティー IP 評価リスト		
評価リスト保護を有効にする	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 出口ノードを含みます) 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
SQL インジェクション保護を有効にする	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。 AWS WAF が 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処理するようにするには、オプション (yes (続行)、yes - MATCH または yes - NO_MATCH) のいずれかを選択します。デフォルトでは、yes は、ルール検査基準に従って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。詳細については、「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。 Note CloudFormation スタックは、選択したオーバーサイズ処理

パラメータ	デフォルト	説明
		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 アカウントにデプロイします。CloudFormation の外部でルールをカスタマイズした場合、スタックの更新後に変更が上書きされます。

パラメータ	デフォルト	説明
SQL インジェクション保護の感度レベル	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ます。 **Note** 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

パラメータ	デフォルト	説明
		に追加し、AWS アカウントにデプロイします。CloudFormation の外部でルールをカスタマイズした場合、スタックの更新後に変更が上書きされます。

パラメータ	デフォルト	説明
クロスサイトスクリプティング保護を有効にする	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。AWS WAF が 8 KB (8192 バイト) を超えるオーバーサイズのリクエストを処理するようにするには、オプション (yes (続行)、yes - MATCH または yes - NO_MATCH) のいずれかを選択することもできます。デフォルトでは、yes は Continue オプションを使用し、ルール検査基準に従って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。詳細については、「オーバーサイズ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。 **Note** 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クロスサイ

パラメータ	デフォルト	説明
		トスクリプティングルールに追加し、AWS アカウントにデプロイします。CloudFormation の外部でルールをカスタマイズした場合、スタックの更新後に変更が上書きされます。
許可および拒否された IP 保持設定		
許可された IP セットの保持期間 (分)	-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
拒否された IP セットの保持期間 (分)	-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。
許可または拒否された IP セットの有効期限が切れたときに通知を受信するための E メール	<オプション入力>	IP 保持期間パラメータを有効にし (前の 2 つのパラメータを参照)、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 保持を有効にしていない場合、または E メール通知をオフにする場合は、空白のままにします (デフォルト)。
詳細設定		

パラメータ	デフォルト	説明
ロググループの保持期間 (日数)	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. [Next] 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。[Next] 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。テンプレートが IAM リソースや必要な追加機能を作成することを確認するチェック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**
>
> `Log Parser`、`IP Lists Parser`、`Access Handler` AWS Lambda 関数に加えて、このソリューションには、`helper` および `custom-resource` Lambda 関数が含まれています。これらの関数は、初期設定中、またはリソースが更新または削除されたとき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 つの関数は削除しないでください。関連するリソースを管理するために必要です。

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、API Gateway のハニーポットエンドポイントである BadBotHoneypotEndpoint 値が含まれます。「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」で使用するため、この値を覚えておいてください。

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 (複数可) または ALB (複数可) を更新して、「Step 1. Launch the stack」で生成したリソースを使用して AWS WAF およびログ記録を有効にします。スタックを起動する」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. 使用するウェブ ACL を選択します。
3. [関連付けられた AWS リソース] タブで、[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. ウェブアクセスログ記録を設定する

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。

CloudFront ディストリビューションから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は、現在の AWS アカウントに関連付けられているバケットが列挙されます。詳細については、「Amazon CloudFront デベロッパーガイド」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) コンソールにサインインします。
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 の場所] には、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. [Save] 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。

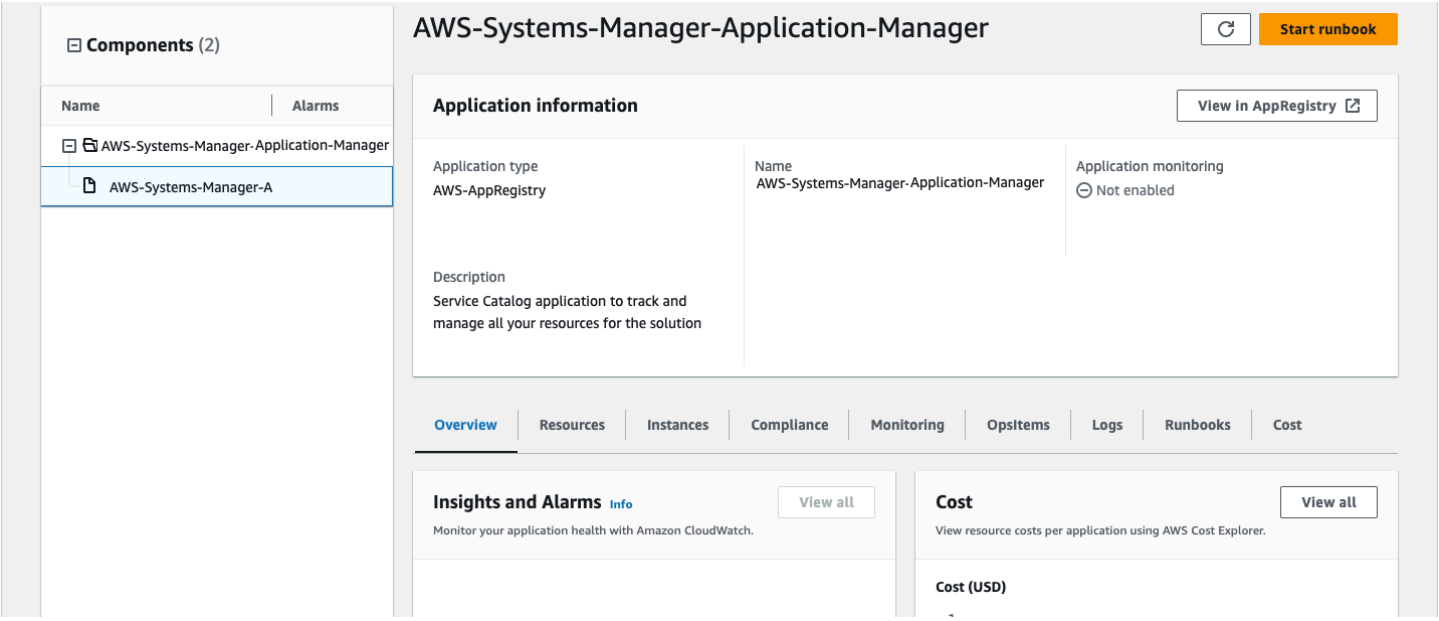
AppRegistry によるソリューションのモニタリング

ソリューションには、Service Catalog AppRegistry および AWS Systems Manager Application Manager の両方のアプリケーションとして、CloudFormation テンプレートと基礎となるリソースを登録するための Service Catalog AppRegistry リソースが含まれています。

AWS Systems Manager Application Manager では、このソリューションとそのリソースをアプリケーションレベルで表示できるため、以下の操作を行うことができます。

- リソース、スタックと AWS アカウント にデプロイされたリソースのコスト、およびソリューションに関連するログを、一元化された場所からモニタリングします。
- このソリューションのリソースの運用データをアプリケーションのコンテキストで表示します。これには、デプロイステータス、CloudWatch アラーム、リソース設定、運用上の問題などが含まれます。

次の図は、Application Manager のソリューションスタックに関するアプリケーションビューの例を示しています。



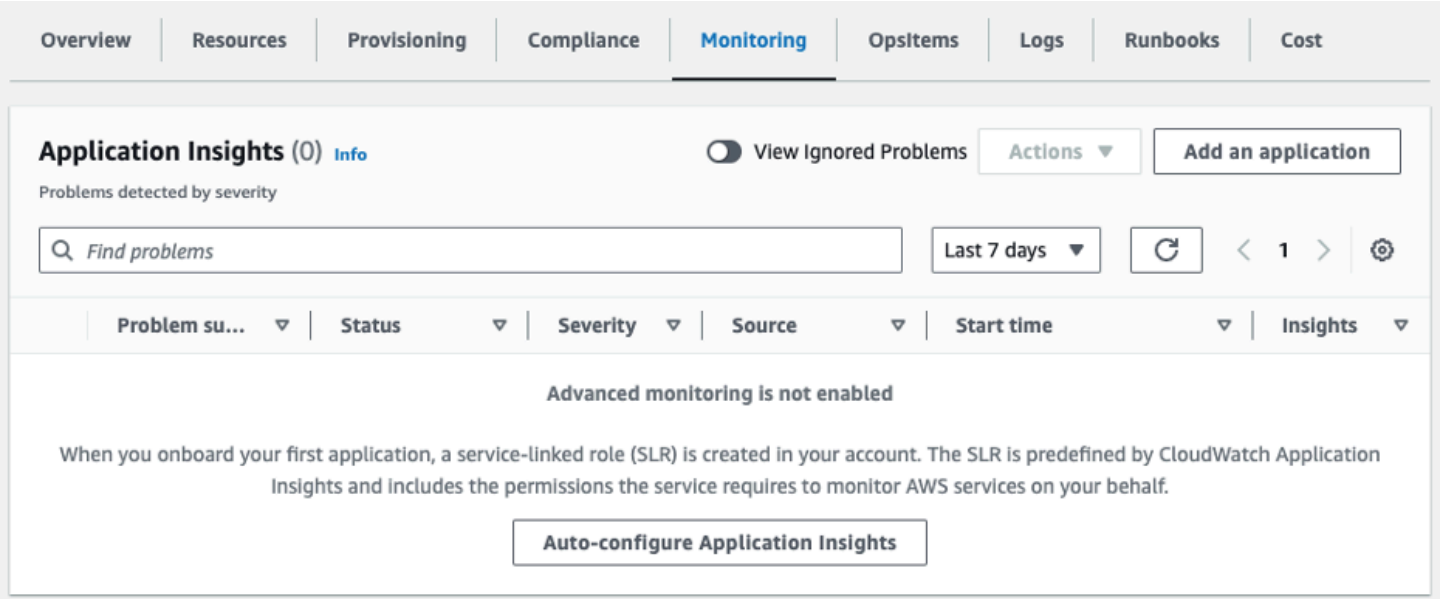
Application Manager のソリューションスタック

CloudWatch Application Insights アクティブ化する

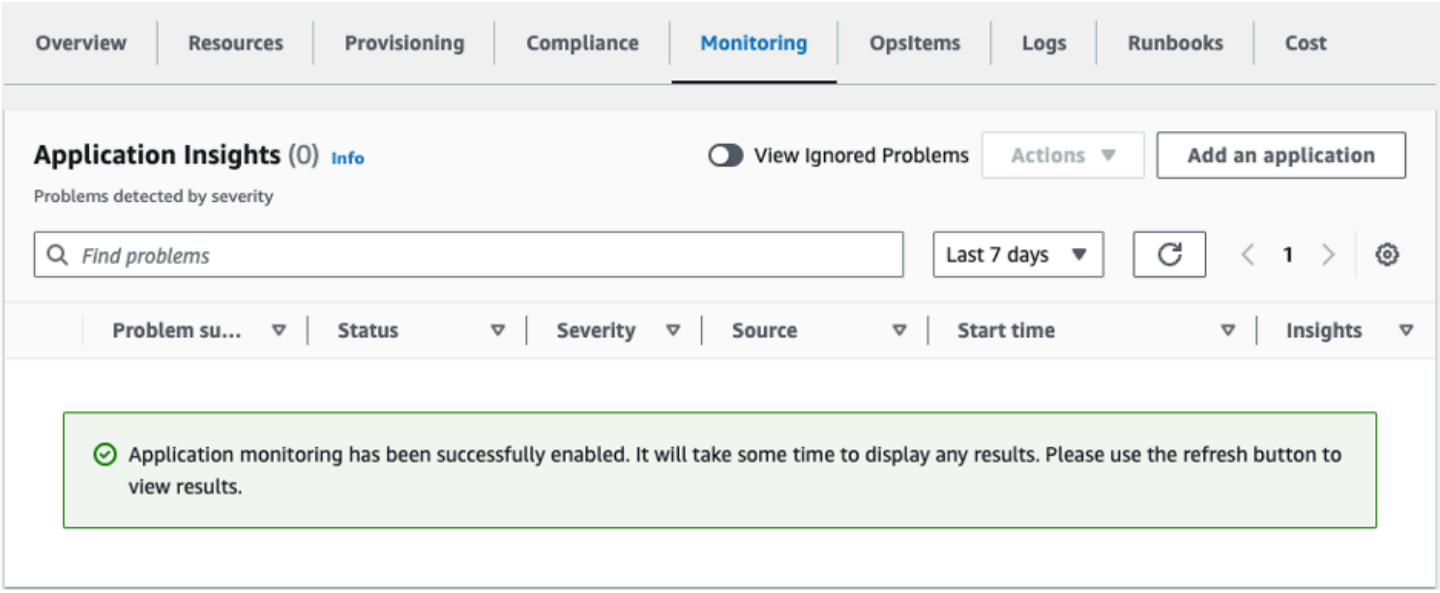
1. Systems Manager コンソールにサインインします。

2. [Application Manager] を選択します。
3. [アプリケーション] で、このソリューションのアプリケーション名を検索して選択します。

   アプリケーション名は、[アプリケーションソース] 列の [App Registry] と、ソリューション名、リージョン、アカウント ID、またはスタック名の組み合わせで構成されます。
4. [コンポーネント] ツリーで、アクティブにするアプリケーションスタックを選択します。
5. [モニタリング] タブの [Application Insights] で、[Application Insights を自動設定] を選択します。



アプリケーションのモニタリングが有効になり、次のステータスボックスが表示されます。

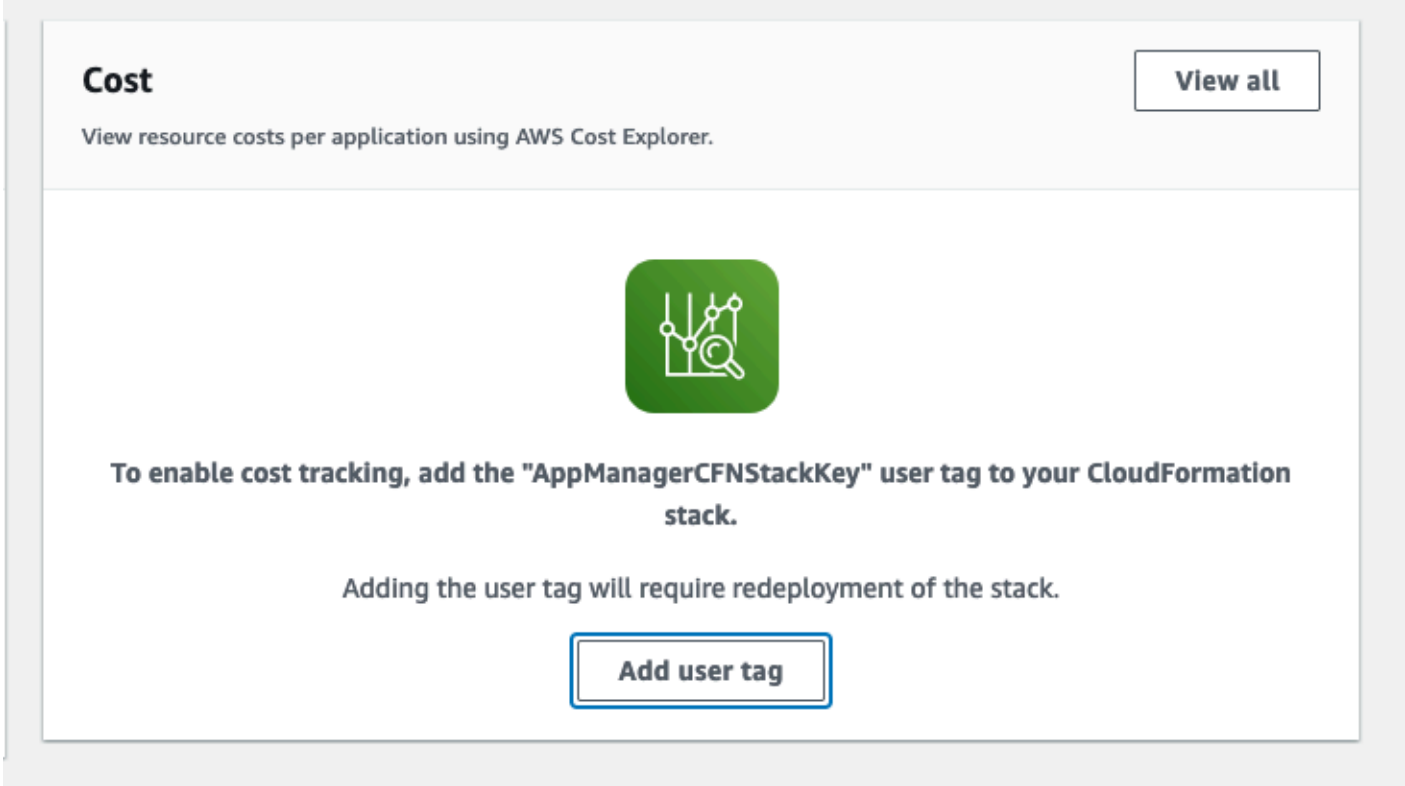


ソリューションに関連するコストタグを確認する

このソリューションに関連するコスト配分タグをアクティブ化した後、コスト配分タグを確認して、このソリューションのコストをチェックする必要があります。コスト配分タグを確認するには:

1. Systems Manager コンソールにサインインします。
2. [Application Manager] を選択します。
3. [アプリケーション] で、このソリューションのアプリケーション名を検索して選択します。

   アプリケーション名は、[アプリケーションソース] 列の [App Registry] と、ソリューション名、リージョン、アカウント ID、またはスタック名の組み合わせで構成されます。
4. [概要] タブの [コスト] で、[ユーザータグを追加] を選択します。



5. [ユーザータグを追加] ページで、「`confirm`」と入力し、[ユーザータグを追加] を選択します。

アクティベーションプロセスが完了して、タグデータが表示されるまでに最大 24 時間かかることがあります。

ソリューションに関連するコスト配分タグをアクティブ化する

Cost Explorer をアクティブ化したら、このソリューションに関連するコスト配分タグをアクティブ化して、このソリューションのコストを確認する必要があります。コスト配分タグは、組織の管理アカウントからのみアクティブ化できます。コスト配分タグをアクティブ化するには:

1. AWS Billing and Cost Management コンソールにサインインします。
2. ナビゲーションペインで、[コスト配分タグ] を選択します。
3. [コスト配分タグ] ページで、AppManagerCFNStackKey タグをフィルタリングし、表示された結果から同タグを選択します。
4. [有効化] を選択します。

AWS Cost Explorer

アプリケーションおよびアプリケーションコンポーネントに関連するコストの概要は、AWS Cost Explorer との統合 (最初にアクティブ化する必要があります) により、Application Manager コンソール内で確認できます。Cost Explorer では、時間経過に伴う AWS リソースのコストと使用量を提供するころにより、コストを管理できます。ソリューションに対して Cost Explorer をアクティブ化するには:

1. AWS Cost Management コンソールにサインインします。
2. ナビゲーションペインで [Cost Explorer] を選択し、ソリューションの経時的なコストと使用状況を表示します。

ソリューションを更新する

過去にソリューションをデプロイしたことがある場合は、次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てください。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. 左のナビゲーションメニューで [スタック] を選択します。
3. 既存の `aws-waf-security-automations` CloudFormation スタックを選択します。
4. [Update] (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. [Specify template] (テンプレートを指定) で、以下を実行します。
   a. [Amazon S3 URL] (Simple Storage Service (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. 正しいテンプレート 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. [Next] を選択します。
   f. [次へ] をもう一度選択します。
7. [Parameters] (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. [Next] 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. [確認] ページで、設定を確認して確定します。
11. テンプレートが IAM リソースを作成する可能性があることを確認するチェックボックスを選択します。
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表示できます。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

以下のセクションでは、このソリューションを更新する際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エンドポイント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エンドポイントパラメータを変更しないでください。

WAFV2 のアップグレード

バージョン 3.0 以降、このソリューションは AWS WAF V2 をサポートしています。すべての AWS WAF Classic API コールを AWS WAF V2 API コールに置き換えました。これにより、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ます。最新の機能や改善点とともにこのソリューションを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

すぐに使えるソリューションでは、CloudFormation スタックを使用して、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。スタックの更新によって、これらの変更が上書きされます。カスタマイズされたルールが必要な場合は、ソリューションの外部に個別のルールを作成することをお勧めします。

> **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 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**
>
> ソリューションをアンインストールすると、Amazon S3 バケットを除く、ソリューションで使用されているすべての AWS リソースが削除されます。AWA WAF API クォータによってレートがスロットリングを超えたために一部の IP セットを削除できない場合は、これらの 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した後にソリューションを使用するための詳細な手順について説明します。

許可および拒否された IP 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、許可および拒否された IP セットを手動で変更し、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. [許可リストの IP セット] を選択し、信頼できるソースから IP アドレスを追加します。
4. [拒否リストの 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 で Activate Bad Bot Protection パラメータで yes を選択した場合。スタックを起動すると、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して転送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へのアクセスを試みません。

ただし、セキュリティの脆弱性をスキャンして E メールアドレスをスクレイピングするマルウェアなどのコンテンツスクレイパーやボットは、トラップエンドポイントへのアクセスを試みる可能性があります。このシナリオでは、`Access Handler` Lambda 関数はリクエストを検査してオリジンを抽出し、関連する AWS WAF ルールを更新して、その IP アドレスからのその後のリクエストをブロックします。

次のいずれかの手順を使用して、CloudFront ディストリビューションまたは ALB からのリクエストにハニーポットリンクを埋め込みます。

Honeypot エンドポイントの 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 を使用すると、ロボット除外標準を無視するコンテンツスクレイパーやボットを特定するのに役立つ `robots.txt` ファイルを含めることができます。次のステップを完了して隠しリンクを埋め込んでから、`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 で構築したスタックを選択します。スタックを起動する
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。この手順を完了するために次の 2 つのコンポーネントが必要です。
   - エンドポイントのホスト名 (例: `xxxxxxxxxx.execute-api.region.amazonaws.com`)
   - リクエスト URI (`/ProdStage`)
5. Amazon CloudFront コンソールにサインインします。
6. 使用するディストリビューションを選択します。
7. [Distribution Settings] を選択します。
8. [Origins] (オリジン) タブで、[Create Origin] (オリジンの作成) を選択します。
9. [オリジンドメイン名] フィールドに、ステップ 2 でコピーしたエンドポイント URL のホスト名コンポーネントを貼り付けます。ウェブ ACL をウェブアプリケーションに関連付けます。
10. [オリジンパス] で、ステップ 2 でコピーしたリクエスト URL を貼り付けます。ウェブ ACL をウェブアプリケーションに関連付けます。
11. その他のフィールドはデフォルト値を受け入れます。
12. [Create] (作成) を選択します。
13. [Behaviors] (動作) タブで、[Create Behavior] (動作の作成) を選択します。
14. 新しいキャッシュ動作を作成し、それを新しいオリジンに指定します。ウェブアプリケーションの他のコンテンツに似たフェイク製品名などのカスタムドメインを使用できます。
15. このエンドポイントリンクを、ハニーポットを指すコンテンツに埋め込みます。このリンクを人間のユーザーに非表示にします。例として、次のコードサンプルを確認します。

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

> **Note**
>
> ウェブサイト環境で機能するタグ値を確認するのはお客様の責任です。お客様の環境で監視していない場合は、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。

16次のように、ウェブサイトのルートにある robots.txt ファイルを変更して、ハニーポットリンクを明示的に禁止します。

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

Honeypot エンドポイントを外部リンクとして埋め込む

ALB でデプロイされたウェブアプリケーションには、この手順を使用します。

1. AWS CloudFormation コンソール にサインインします。
2. ステップ 1 で構築したスタックを選択します。スタックを起動する」を参照してください。
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。
5. このエンドポイントリンクをウェブコンテンツに埋め込みます。ステップ 2 でコピーした完全な URL を使用します。ウェブ ACL をウェブアプリケーションに関連付けます。このリンクを人間のユーザーに非表示にします。例として、次のコードサンプルを確認します。

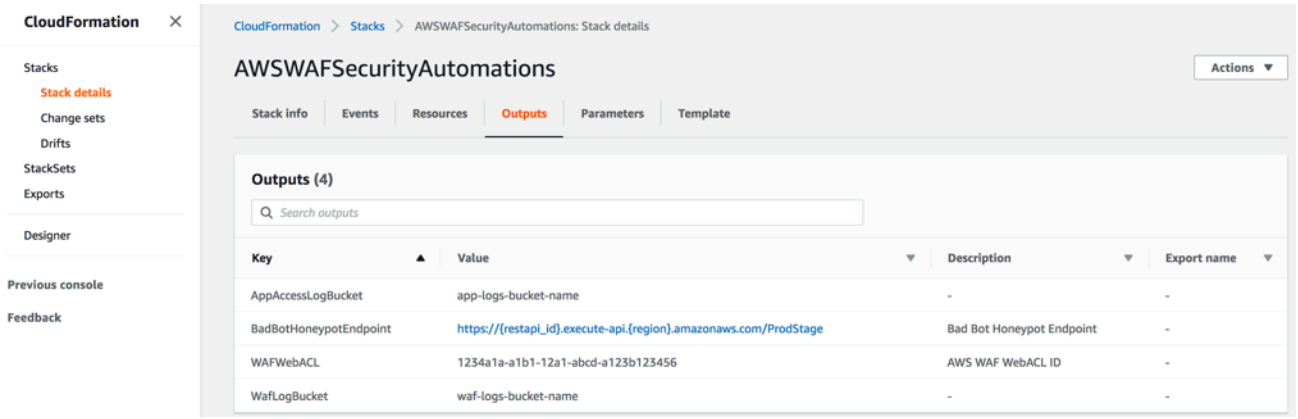
```
<a href="<BadBotHoneypotEndpoint value>" rel="nofollow" style="display: none" aria-
hidden="true"><honeypot link></a>
```

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

Lambda ログパーサー 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。次の図に例を示します。



スタック出力

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。

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

HTTP フラッド設定ファイル

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 スタックのプロビジョニングまたは更新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 スタックのプロビジョニングまたは更新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 – 特定の URL に対するカスタムリクエストのしきい値とブロック期間を定義するために使用します。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 *<stack_name>*-app_log_conf.json という名前の設定ファイルを作成し、CloudFront または Application Load Balancer ログファイルの保存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。

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

スキャナーとプローブの設定ファイル

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) — IP アドレスごとの 1 分あたりに許容される不正なリクエストの最大数。このソリューションは、CloudFormation スタックのプロビジョニングまたは更新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 スタックのプロビジョニングまたは更新時に定義した値を使用します。
  - エラーコード – Teturn ステータスコードはエラーと見なされます。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 – 特定の URL のカスタムリクエストのしきい値とブロック期間を定義するために使用します。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 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 HTTP Flood Athena クエリでのリクエストによるグループ化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、国別のリクエストしきい値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}` と指定します。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 クエリのグループ化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 のパラメータ表を参照してください。スタックを起動する」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 クエリ実行時間スケジュール (分) 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：
> リクエストしきい値 (デフォルトのしきい値または国別のしきい値): 100
> Athena クエリ実行時間スケジュール: 5
> 1 分あたりのリクエストしきい値: 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パーティション分割します。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 クエリ実行時間スケジュール (分) 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。デフォルトでは、各クエリ実行は直近の 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の量を設定できます。このソリューションは、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 AWS Glue Data Catalog 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 でアクセスログデータカタログを作成し、データを処理するための Athena クエリを設定します。Athena が正しく設定されていない場合、クエリは実行されません。詳細については、「Upgrading to the latest AWSAWS Glue Data Catalog step-by-step」を参照してください。

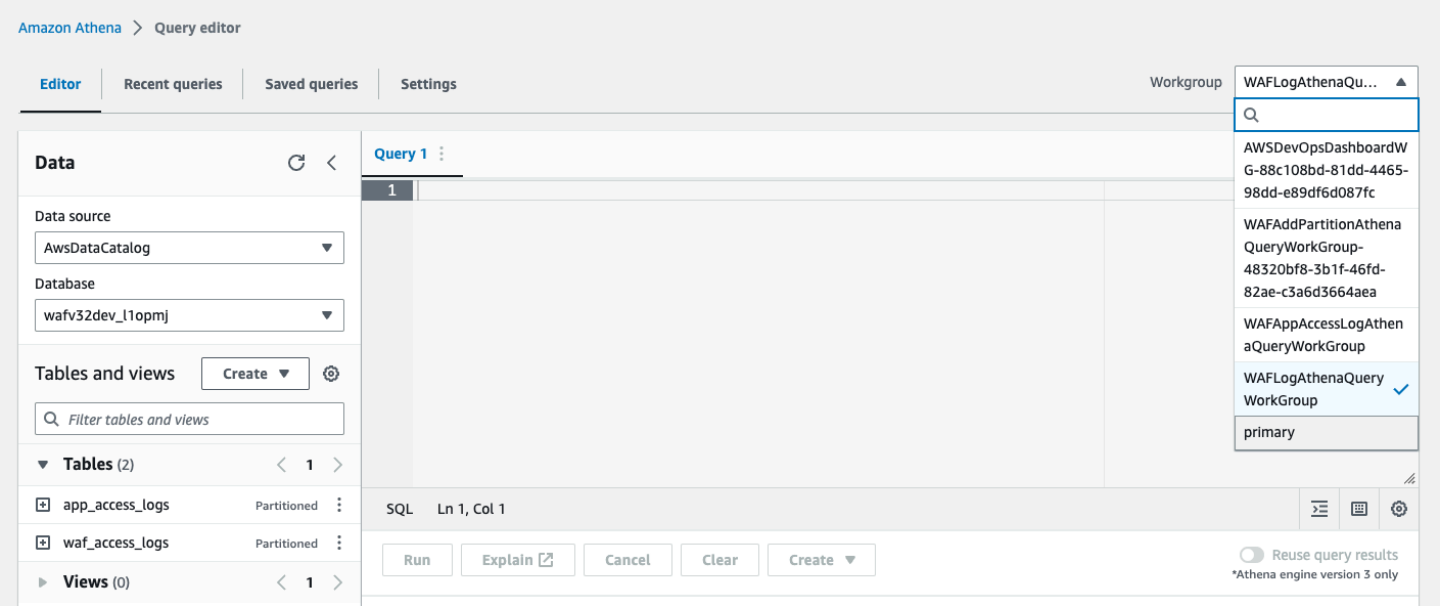
これらのクエリを表示するには、次の手順に従います。

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. [クエリエディタを起動] 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。



6. [履歴] タブを選択します。
7. 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に `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。

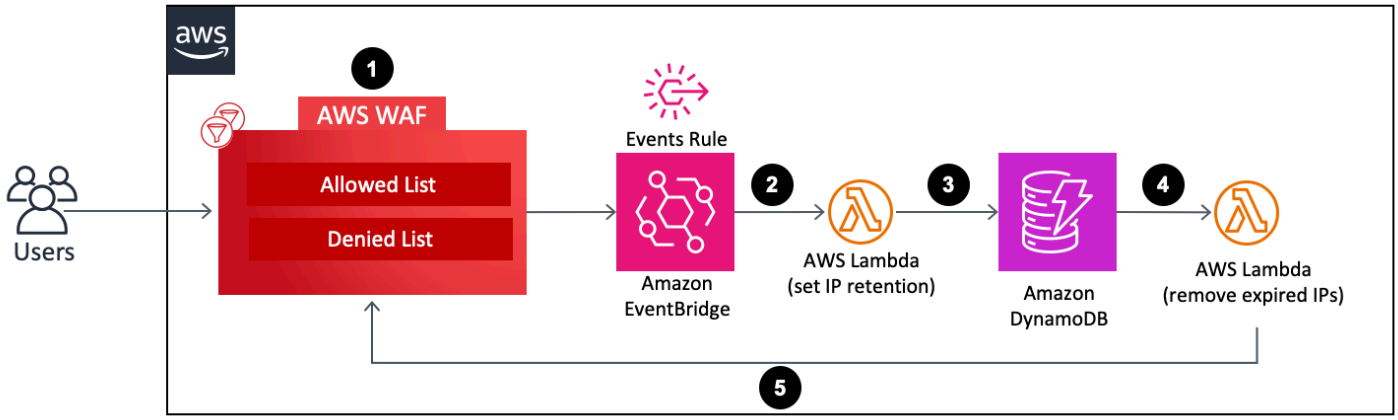
> **Note**
>
> このワークグループは、Activate HTTP Flood Protection または Activate Scanner & Probe Protection テンプレートパラメータに `Yes - Amazon Athena log parser` を選択した場合にのみ存在します。

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 1 時間ごとのパーティションを追加します。

許可および拒否された AWS WAF IP セットで IP 保持を設定する

IP 保持は、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 セットで設定できます。以下のセクションでは、その仕組みについて説明し、設定する手順を示します。

仕組み



許可および拒否された WAF IP セットの IP 保持

1. ユーザーが許可または拒否された WAF IP セットを更新 (IP アドレスを追加または削除) すると、このアクションにより AWS WAF `UpdateIPSet` API コールが呼び出され、イベントが作成されます。

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 項目に `ExpirationTime` 属性を挿入します。このソリューションでは、ユーザー定義の保持期間をイベント時間に追加することで有効期限を計算します。このテーブルでは、DynamoDB Streams と Time to Live (TTL) が有効になっています。TTL 属性は `ExpirationTime` です。
4. 項目の有効期限に達すると、TTL が呼び出され、DynamoDB は有効期限後にテーブルから項目を削除します。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 WAFAPI コールを実行して、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 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、許可された IP セットの IP 保持期間 (分) と拒否された IP セットの IP 保持期間 (分) 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.」を参照してください。スタックを起動する」を参照してください。
2. 期限切れの IP アドレスが AWS WAF IP セットから削除されたときに E メール通知を受信する場合は、E メールアドレスを入力します。E メール通知を受信する場合は、ソリューションが正常にデプロイされた後に受信した E メール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.」を参照してください。スタックを起動する」を参照してください。
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 によって期限切れになった項目を削除する時間によっては、AWS WAF IP セットからの期限切れ IP アドレスの実際の削除オペレーションが異なる場合があります。DynamoDB TTL の削除は、主にテーブルのサイズとアクティビティレベルによって異なります。DynamoDB 削除オペレーションに遅延が生じる可能性があるため、AWS WAF の削除オペレーションに遅延が発生する可能性があります。一般的に、このソリューションは、DynamoDB TTL 削除の直後に期限切れの IP アドレスを AWS WAF IP セットから削除します。詳細については、「Amazon DynamoDB デベロッパーガイド」の「DynamoDB Time to Live (TTL)」を参照してください。

モニタリングダッシュボードの構築

AWS では、重要なエンドポイントごとにカスタム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のスクリーンショットは、カスタム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。

ダッシュボードには、以下のメトリクスが表示されます。

- 許可されたリクエストとブロックされたリクエスト – 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 (ブロックされたリクエストが 1K を超える期間) が急増したかどうかを示します。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) または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) を追跡できます。

> **Note**
>
> 注: ソリューションはこのメトリクスを提供します。

- BytesDownloaded と Uploaded – DDoS 攻撃が、通常は大量のアクセスを受けずリソースを枯渇させることがないサービス (例: 特定のリクエストパラメータセットの MB 単位の情報を送信する検索エンジンコンポーネント) をいつターゲットにするかを特定するのに役立ちます。
- ELB の過剰数とキューの長さ — DDoS 攻撃がインフラストラクチャにダメージを与え、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 リクエスト数 — 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 CloudFront キャッシュルールを確認してキャッシュヒットレートを上げる必要があるかどうかを示します。
- ELB の正常なホスト – これは、別のシステムヘルスチェックメトリクスとして使用できます。
- ASG CPU 使用率 — 攻撃者が CloudFront、AWS WAF、および Elastic Load Balancing をバイパスしているかどうかを識別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こともできます。

XSS 誤検出を処理する

このソリューションは、受信リクエストの一般的に調査される要素を検査して XSS 攻撃を識別してブロックする 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でリッチテキストエディタを使用するなど、正規ユーザーが HTML を作成および送信できるワークロードの場合、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、画像またはカスタムデータ形式によっては、HTML コンテンツで XSS 攻撃の可能性があること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> タグが含まれている場合があります。正規ユーザーからこのタイプのコンテンツが予想される場合は、XSS ルールを絞り込んで、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します。

HTML を入力として受け入れる URL を除外するように XSS ルールを更新するには、次のステップを実行します。詳細な手順については、「Amazon WAF デベロッパーガイド」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一覧表示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、リクエストがに対して以下を選択します。

   - 一致しない
   - 文字列一致条件のフィルターの少なくとも 1 つに一致
   - XSS 許可リスト

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、Support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。

Supportに問い合わせる

AWS デベロッパーサポート、AWS ビジネスサポート、または AWS エンタープライズサポートをご利用の場合は、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工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。

ケースの作成

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、[WAF] または [AWS WAF] を選択します。
3. [カテゴリ] で、[WAF セキュリティオートメーション] または [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] を選択します。

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. 説明に、問題の詳細を入力します。
3. [ファイルを添付] を選択します。
4. Support がリクエストを処理するため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. 今すぐ解決の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。

開発者ガイド

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 リポジトリでは、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、カスタマイズ内容を他のユーザーと共有できます。

リファレンス

このセクションでは、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に関する情報、関連リソースへのポインタ、およびこのソリューションに貢献したビルダーのリストが含まれています。

匿名化されたデータ収集

このソリューションには、オペレーションメトリクスを AWS に送信するオプションが含まれています。このデータを使用して、ユーザーがこのソリューションおよび関連サービスや製品をどのように使用しているかをよりよく理解します。オンにすると、ソリューションは CloudFormation テンプレートの初期デプロイ時に次の情報を収集し、AWS に送信します。

- Solution ID – AWS ソリューション識別子
- 一意の ID (UUID) – このソリューションの各デプロイごとにランダムに生成された一意の識別子
- Timestamp - データ収集のタイムスタンプ
- Solution configuration – 初回起動時にオンにされた機能と設定されたパラメータ
- Lifecycle – お客様がこのソリューションを使用した期間 (スタックの削除にもとづく)
- ログパーサーデータ:
  - Scanner & Probe IP セットとブロックする HTTP Flood 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 リストパーサーデータ:
  - レピュテーションリスト 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
- アクセスハンドラーデータ:
  - Bad Bot 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 – Allowed または Denied IP セットから削除される期限切れの IP アドレスの数

このアンケートで収集されたデータは AWS が所有します。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。この機能を無効にするには、AWS CloudFormation テンプレートを起動する前に次のステップを実行します。

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"
```

変更先:

```
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、編集したテンプレートをローカルドライブから選択します。
8. [次へ] を選択し、ステップ 1 の手順に従います。スタックを起動します。

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の投稿

- 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 レピュテーションリスト

- Spamhaus DROP List ウェブサイト
- Proofpoint Emerging Threats IP リスト
- Tor 出口ノードリスト

寄稿者

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

リビジョン

日付	変更
2016 年 月 9 日	初回リリース
2017 年 1 月	このソリューションの IP アドレス制限を明確化しました。
2017 年 3 月	キャッシュ動作の作成に関する追加のガイダンス。AWS セキュリティブログ投稿の URL を更新しました。
2017 年 6 月	ALB サポートを追加し、製品制限を更新しました。
2017 年 11 月	HTTP フラッド保護のレートベースのルールサポートを追加しました。リソースアクセスログを保存するための追加のリンク。
2018 年 1 月	Application Load Balancer の AWS WAF のリージョン別の可用性に関するコンテンツを更新しました。
2018 年 12 月	IPv6 サポートを追加、CIDR 範囲を拡張、モニタリングダッシュボードを追加しました。
2019 年 4 月	AWS WAF ログを統合、Amazon Athena を統合、設定可能なログパーサーを追加しました。
2019 年 12 月	Node.js の更新のサポートに関する情報を追加しました。
2020 年 2 月	バグ修正と RequestThreshold パラメータを更新しました。
2020 年 6 月	パーティショニングを使用した Athena のコスト最適化を追加、README 指示を更新、Bad

日付	変更
	Bots X-Forward-For ヘッダー内の潜在的な DoS の問題を修正しました。
2020 年 7 月	AWS WAF Classic から AWS WAF V2 サービス API にアップグレードしました。
2020 年 11 月	リリースバージョン 3.1.0: 特定のリージョンの HTTP Flood Protection と Scanner & Probe Protection のルールを明確化し、S3 パスタイプを仮想ホスト形式に置き換え、すべての ARN にパーティション変数を追加しました。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。
2021 年 9 月	リリースバージョン 3.2.0: 許可および拒否された IP セットの IP 保持サポートが追加されました。バグ修正されました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2022 年 8 月	リリースバージョン 3.2.1: リクエストコンポーネントの WAF オーバーサイズ処理のサポートが追加されました。また、SQL インジェクションルールステートメントの WAF 機密レベルのサポートが追加されました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2022 年 9 月	ソリューションの CloudFormation スタック外でのカスタマイズに関するドキュメントを更新しました。

日付	変更
2022 年 12 月	リリースバージョン 3.2.2: Service Catalog AppRegistry および AWS Systems Manager Application Manager との統合を追加しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2022 年 12 月	リリースバージョン 3.2.3: AWS で始まる名前との競合を避けるため、アプリケーション属性グループ名にプレフィックスとしてリージョンを追加します。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2023 年 2 月	リリースバージョン 3.2.4: CVE を軽減するために pytest とリクエストをアップグレードしました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2023 年 3 月	IP アドレスを許可または拒否したバージョン 3.0 または 3.1 から 3.2 以降にソリューションをアップグレードするためのドキュメントを更新しました。
2023 年 4 月	リリースバージョン 3.2.5: すべての新しい Amazon S3 バケットの Amazon S3 オブジェクト所有権の新しいデフォルト設定 (ACL は無効) による影響を軽減しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。

日付	変更
2023 年 5 月	リリースバージョン 4.0.0: 新しい AWS マネージドルール ルールグループのサポートを追加し、カスタムルールを更新しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2023 年 5 月	リリースバージョン 4.0.1: `.gitignore` ファイルを更新して、ファイルが見つからない問題を解決しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2023 年 9 月	リリースバージョン 4.0.2: 品質を向上させるためにコードをリファクタリングしました。リクエストパッケージの脆弱性にパッチを適用しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2023 年 10 月	リリースバージョン 4.0.3: セキュリティの脆弱性を解決するためにパッケージバージョンを更新しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2023 年 11 月	ドキュメントの更新: AWS デベロッパーサポートを追加し、AWS サポートへの問い合わせをトラブルシューティングセクションに統合しました。
2023 年 11 月	ドキュメントの更新: 「AWS Service Catalog AppRegistry によるソリューションのモニタリング」セクションに「ソリューションに関連するコストタグを確認する」を追加しました。

日付	変更
2024 年 4 月	ドキュメントの更新: デプロイステップ 3 で S3 バケットを追加する手順を明確にしました。
2024 年 9 月	リリースバージョン 4.0.4: セキュリティの脆弱性を解決するためにパッケージバージョンを更新しました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2024 年 10 月	リリースバージョン 4.0.5: 依存関係管理に Poetry を使用しました。ネイティブ Python ロガーを aws_lambda_powertools ロガーに置き換えました。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。
2024 年 12 月	リリースバージョン 4.0.6: lambda を python 3.12 に更新します。詳細については、GitHub リポジトリ内の CHANGELOG.md ファイルを参照してください。

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。これは、このドキュメントの発行日現在の AWS 製品およびプラクティスを示すもので、予告なく変更される場合があります。お客様は、本書に記載されている情報と AWS の製品やサービスの使用について独自の独立した評価を行う責任があります。各製品やサービスは、明示的か黙示的かを問わず、いかなる種類の保証もなしに「現状有姿」で提供されます。本文書内のいかなるものも、AWS、その関係者、サプライヤ、またはライセンサーからの保証、表明、契約的なコミットメント、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 License Version 2.0 の条件に基づいてライセンスされます。