AWS Best Practices for DDoS Resiliency

2016 年 6 月

注意

本書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、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。

目次

要約

本書は、アマゾン ウェブ サービス (AWS) で実行するアプリケーションの分散サービス妨害 (DDoS) 攻撃に対する弾力性を高めることをお考えのお客様を対象としています。DDoS 攻撃の概要、AWS が提供する機能、緩和技術、DDoS 弾力性のリファレンスアーキテクチャを説明しており、アプリケーションの可用性を保護するための参考になります。

はじめに

本書の対象者は IT に関する意思決定者およびセキュリティ担当者であり、ネットワーキング、セキュリティ、AWS におけるセキュリティの基本的な概念を理解していることを前提として書かれています。各セクションには、ベストプラクティスや機能の詳細が記された AWS ドキュメントへのリンクがあります。AWS re:Invent conference セッションの SEC307 – AWS による DDoS に強いアーキテクチャの構築[1] および SEC306 – DDoS 攻撃に対する防御[2] で詳細を確認することもできます。

DDoS 攻撃

サービス拒否 (DoS) 攻撃が行われると、お客様のウェブサイトやアプリケーションをエンドユーザーが利用できなくなる場合があります。攻撃者は、これを行うために、ネットワークやその他のリソースを消費するさまざまな手法を用いて、正規のエンドユーザーのアクセスを妨げます。DoS 攻撃の最もシンプルな形式では、図 1 に示すように、単独の攻撃者が 1 つのソースから標的を攻撃します。

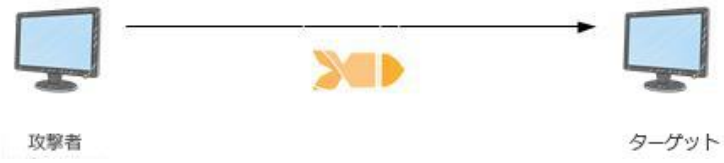

図 1: DoS 攻撃の図

分散サービス妨害 (DDoS) 攻撃では、攻撃者は複数のソースを使用して標的に対する攻撃を指揮します。これらのシステムは、協力者のグループによって侵害または制御されている場合があります。図 2 に示すように、DDoS 攻撃では、各協力者または侵害されたホストが攻撃に参加し、標的に負荷をかけるため大量のパケットやリクエストを生成します。

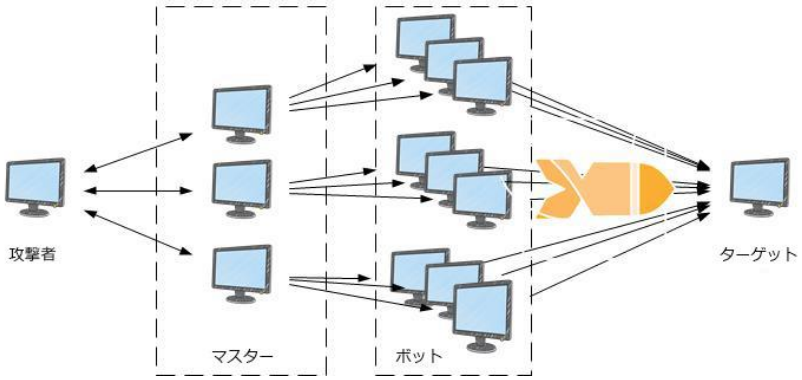

図 2: DDoS 攻撃の図

表 1 の説明にあるとおり、DDoS 攻撃は、開放型システム間相互接続 (OSI) モデルのレイヤー 3、4、6、7 で最もよく見られます。レイヤー 3 および 4 の攻撃は OSI モデルのネットワークおよびトランスポートレイヤーに対応します。このドキュメントでは、これをインフラストラクチャレイヤー攻撃と呼びます。レイヤー 6 および 7 の攻撃は OSI モデルのプレゼンテーションおよびアプリケーションレイヤーに対応します。このドキュメントでは、これをアプリケーションレイヤー攻撃と呼びます。

#	レイヤー	単位	説明	ベクトルの例
7	アプリケーション	データ	アプリケーションへのネットワークプロセス	HTTP フラッド、DNS クエリフラッド
6	プレゼンテーション	データ	データ形式および暗号化	SSL 不正使用
5	セッション	データ	ホスト間の通信	該当なし
4	トランスポート	セグメント	エンドツーエンド接続と信頼性	SYN フラッド
3	ネットワーク	パケット	パスの決定と論理アドレス指定	UDP リフレクション攻撃
2	データリンク	フレーム	物理アドレス指定	該当なし
1	物理	ビット	メディア、信号、バイナリの送信	該当なし

表 1: 開放型システム間相互接続 (OSI) モデル

これらのレイヤーへの攻撃タイプは異なるため、弾力性を構築するための異なる手法を使用するので、この区別は重要です。

インフラストラクチャレイヤー攻撃

最も一般的な DDoS 攻撃、User Datagram Protocol (UDP) リフレクション攻撃、および、同期 (SYN) フラッドは、インフラストラクチャレイヤー攻撃です。攻撃者はこれらのいずれかのメソッドを使用し、ネットワーク、またはサーバー、ファイアウォール、IPS、ロードバランサーなどのシステムを圧迫するほど大量のトラフィックを生成します。これらの攻撃には、検出を容易にする明確な署名があります。これらの攻撃を効率的に緩和するには、攻撃者が生成した量を上回るネットワークやシステムのリソースが必要です。

UDP はステートレスなプロトコルです。これにより攻撃者は、より大きなレスポンスを引き起こすようサーバーに送信されたリクエストのソースをごまかすことができます。増幅係数 (応答サイズに対するリクエストサイズの比率) は、ドメインネームシステム (DNS)、ネットワークタイムプロトコル (NTP)、またはシンプルサービスディスカバリープロトコル (SSDP) など、使用されているプロトコルによって異なります。たとえば、DNS の増幅係数は 28 ～ 54 の範囲です。この場合、攻撃者は 64 バイトのリクエストペイロードを DNS サーバーに送信して、3400 バイト以上の不要なトラフィックを生成することができます。この考えを図 3 に示します。

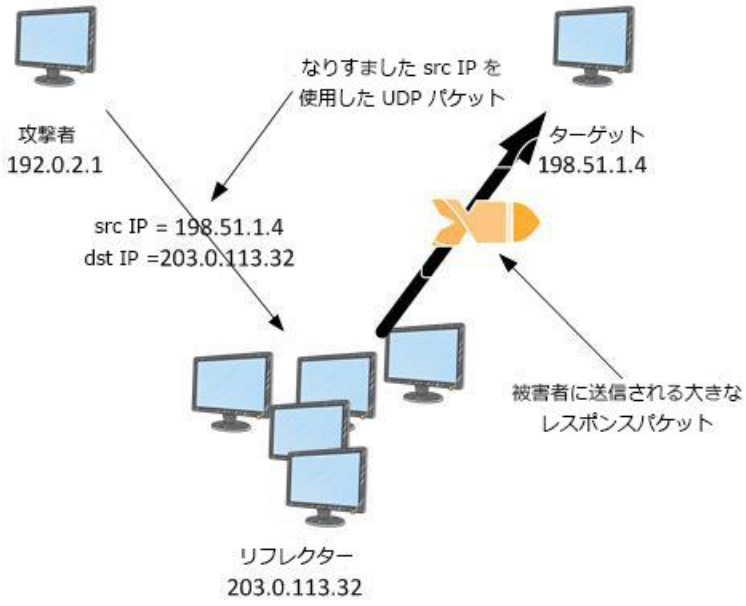

図 3: UDP リフレクション攻撃

SYN フラッドは数十 Gbps 程度ですが、この攻撃が意図しているのは、接続を半分開いた状態にすることでシステムの利用可能なリソースを枯渇させることです。図 4 に示すとおり、エンドユーザーがウェブサーバーのような TCP サービスに接続すると、クライアントは SYN パケットを送信します。サーバーが SYN-ACK を返し、クライアントが ACK を返すことで、3 ウェイハンドシェイクが完了します。

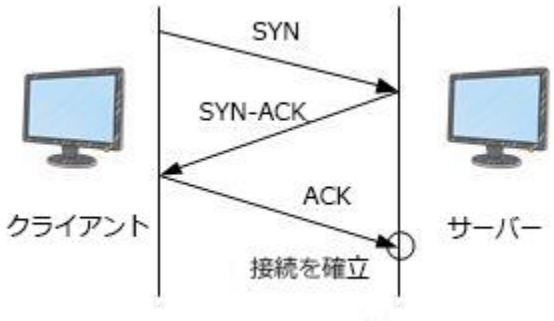

図 4: SYN 3 ウェイハンドシェイク

SYN フラッドでは、ACK が返されることはなく、サーバーはレスポンスを待っている状態になります。このため、新規ユーザーはサーバーに接続できません。

アプリケーションレイヤー攻撃

頻度は低くなりますが、攻撃者はレイヤー 7 またはアプリケーションレイヤー攻撃でアプリケーション自体を標的にする場合があります。これらの攻撃はインフラストラクチャレイヤー攻撃とは異なり、攻撃者は、アプリケーションの特定の機能を過剰に作動させ、使用不可にすることを目的としています。場合によっては、大量のネットワークトラフィックを生成しない非常に少量のリクエストによりこれが実行されることもあります。このような場合は攻撃を検出し緩和することがより難しくなります。アプリケーションレイヤー攻撃の例には、HTTP フラッド、キャッシュバスティング攻撃、WordPress XML-RPC フラッドがあります。

HTTP フラッドでは、攻撃者は、ウェブアプリケーションの正規ユーザーからと思えるような HTTP リクエストを送信します。HTTP フラッドの中には、特定のリソースを標的にしたものもありますが、一方でより複雑な HTTP フラッドは人間の行動をエミュレートしようと試みます。このため、リクエストのレート制限などの一般的な緩和技術を使用することがより難しくなります。キャッシュバスティング攻撃は、HTTP フラッドの一種で、コンテンツ配信ネットワーク (CDN) のキャッシュを回避するためにクエリ文字列のバリエーションを使用します。結果として、オリジンフェッチが生じオリジンウェブサーバーの負荷が増大します。

WordPress ピンバックフラッドとしても知られる WordPress XML-RPC フラッドでは、攻撃者は WordPress ブランドのコンテンツ管理ソフトウェアによりホストされているウェブサイトの XML-RPC API 関数を悪用し、大量の HTTP リクエストを生成します。ピンバック機能により、WordPress にホストされているウェブサイト (サイト A) から、別の WordPress サイト (サイト B) に、サイト A がサイト B へのリンクを作成したことを通知できます。その結果、サイト B はサイト A を取得してリンクの存在を確認しようとします。ピンバックフラッドの場合、攻撃者はこの機能を悪用し、サイト B がサイト A を攻撃するようにします。このタイプの攻撃には、「WordPress」が HTTP リクエストヘッダーの User-Agent に表示されるので、明確な署名があります。

アプリケーションレイヤー攻撃はドメインネームシステム (DNS) も標的にします。これらの攻撃で最も一般的なのは DNS クエリフラッドで、攻撃者は大量の正しい形式の DNS クエリを使用して DNS サーバーのリソースを枯渇させます。これらの攻撃には、キャッシュバスティングの要素も含まれていて、攻撃者はサブドメインの文字列をランダムにし、指定したリゾルバーのローカル DNS キャッシュをバイパスします。結果として、リゾルバーは権威ある DNS サーバーに対する攻撃に徴用されます。

Secure Sockets Layer (SSL) で配信されたウェブアプリケーションの場合、攻撃者は SSL ネゴシエーションプロセスを攻撃することを選べます。SSL 計算コストが高く、攻撃者が解読できないデータを送信することでサーバーの可用性に影響を及ぼすことができます。この攻撃のその他のバリエーションとしては、攻撃者が SSL ハンドシェイクを完了しても、継続的に暗号化方法のネゴシエーションを繰り返すものがあります。同様に、攻撃者が多数の SSL セッションを開いたり閉じたりすることで、サーバーのリソースを枯渇させることがあります。

緩和テクニック

AWS インフラストラクチャは、DDoS 弾力性があるよう設計されていて、過剰なトラフィックを自動的に検出しフィルタリングする DDoS 緩和システムがサポートされています。アプリケーションの可用性を保護するため、これらの機能を活用できるアーキテクチャを実装する必要があります。

最も一般的な AWS ユースケースの 1 つは、インターネット上で静的および動的コンテンツをユーザーに提供するウェブアプリケーションです。ウェブアプリケーションで一般的に使用されている DDoS 弾力性のあるリファレンスアーキテクチャについては、図 5 を参照してください。

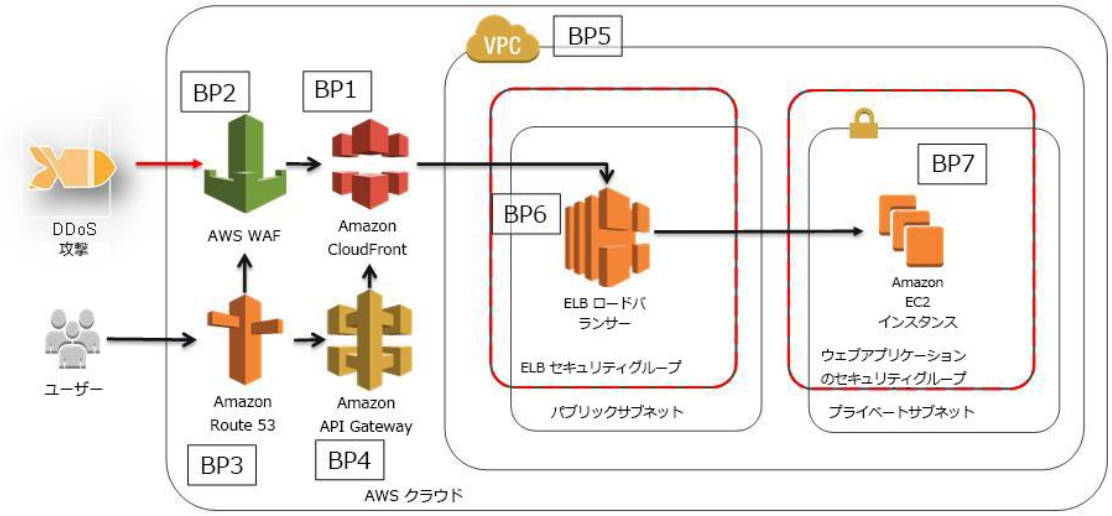

図 5: DDoS 弾力性のあるリファレンスアーキテクチャ

このリファレンスアーキテクチャには、ウェブアプリケーションの DDoS 攻撃に対する弾力性を高めるのに役立つ数多くの AWS のサービスが含まれています。このアーキテクチャのベストプラクティスは、ドキュメント内で説明される際、参照しやすいよう列挙されています。たとえば、Amazon CloudFront が提供する機能を説明するセクションには、ベストプラクティスを参照するための記号 (BP1 など) があります。これらのサービスおよび提供されている機能の概要については、テーブル 2 を参照してください。

	AWS エッジロケーション			**AWS リージョン**		
	AWS WAF を使用した Amazon CloudFront (BP1、BP2)	Amazon API Gateway (BP4)	Amazon Route 53 (BP3)	Elastic Load Balancing (BP6)	Amazon VPC (BP5)	Auto Scaling を使用した Amazon EC2 (BP7)
レイヤー 3 (例: UDP リフレクション) 攻撃の緩和	✓	✓	✓	✓	✓	
レイヤー 4 (例: SYN フラッド) 攻撃の緩和	✓	✓	✓	✓		
レイヤー 6 (例: SSL) 攻撃の緩和	✓	✓	該当なし	✓		
攻撃領域を削減する	✓	✓	✓	✓	✓	
アプリケーションレイヤーのトラフィックをスケールして吸収する	✓	✓	✓	✓		✓
レイヤー 7 (アプリケーションレイヤー) 攻撃の緩和	✓	✓	✓			

過剰なトラフィックの地理的な分離と分散およびより大規模な DDoS 攻撃	✓	✓	✓			

表 2: ベストプラクティスのまとめ

Elastic Load Balancing および Amazon Elastic Compute Cloud (EC2) などの AWS リージョン内で利用可能なサービスにより、DDoS 弾力性を構築でき、リージョン内での想定外の量のトラフィックをスケールし処理できます。Amazon CloudFront、AWS WAF、Amazon Route 53、および Amazon API Gateway などの AWS エッジロケーションで利用可能なサービスにより、エッジロケーションのグローバルなネットワークを活用して、アプリケーションの耐障害性を改善し、より大量のトラフィックをスケールし管理できます。インフラストラクチャレイヤー DDoS 攻撃に対する弾力性を構築するために、これらの各サービスを利用することの利点については、以下のセクションで説明されています。

インフラストラクチャレイヤーディフェンス (BP1, BP3, BP6, BP7)

従来のデータセンターの環境では、過剰なキャパシティを備えること、DDoS 緩和システムをデプロイすること、または DDoS 緩和サービスによるトラフィックのスクラブなどの技術を使用してインフラストラクチャレイヤー DDoS 攻撃を緩和します。AWS では、多くの資本を要する投資や不必要な複雑さはなく、より大量のトラフィックをスケールし吸収できるアプリケーションを構築するという選択肢があります。ボリューメトリック DDoS 攻撃の緩和において考慮すべき重要なポイントには、利用可能な伝送能力と多様性、および攻撃ト

ラフィックに対する Amazon EC2 インスタンスのような AWS リソースの保護が含まれます。

インスタンスサイズ (BP7)

AWS のお客様の多くは、規模を自在に変更できるコンピューティング性能のために Amazon EC2 を使用しており、要件の変更に応じてスケールアップまたはスケールダウンできます。必要に応じてアプリケーションにインスタンスを追加することで水平にスケールできます。より大きなインスタンスを使用することで垂直にもスケールできます。インスタンスタイプによっては、より大量のトラフィックへの処理能力を向上できる、10 ギガビットネットワークインターフェイスや拡張ネットワーキングなどの機能をサポートしています。

10 ギガビットネットワークインターフェイスでは、各インスタンスがより大量のトラフィックをサポートできます。これは、Amazon EC2 インスタンスに到達したトラフィックによるインターフェイスの輻輳を防ぐのに役立ちます。拡張ネットワーキングをサポートするインスタンスでは、従来の実装と比較し、I/O パフォーマンスが高く、CPU 利用率が低くなります。これにより、パケットのボリュームがより大きなトラフィックを処理するインスタンスの能力が向上します。AWS では、インバウンドデータ転送の料金を負担する必要はありません。

10 ギガビットネットワークインターフェイスおよび拡張ネットワーキングをサポートする Amazon EC2 インスタンスの詳細については、Amazon EC2 インスタンスタイプ[3] を参照してください。拡張ネットワーキングを有効にする方法については、「VPC 内の Linux インスタンスで拡張ネットワーキングを有効化する[4]」を参照してください。

リージョンの選択 (BP7)

Amazon EC2 などの AWS のサービスの多くは、世界中の複数の場所で利用できます。これらの地理的に分けられたエリアは AWS リージョンと呼ばれています。アプリケーションを構築する際、要件に基づいて 1 つ、または複数のリージョンを選択することができます。一般的に考慮される点には、パフォーマンス、コスト、データの主権が含まれます。各リージョンで、AWS は独自のインターネット接続およびピア関係へのアクセスを提供し、同様の状況にあるエンドユーザーに対して最適なレイテンシーおよびスループットが得られるようにしています。

DDoS 弾力性の点からもリージョンの選択は重要です。リージョンの多くは大規模なインターネットのやり取りが行われている場所のより近くにあります。多くの DDoS 攻撃は国際的に行われるので、国際キャリアおよび大規模なピアが頻繁な活動を維持している場所の近くでのやり取りは役に立ちます。これは、より大量のトラフィックを処理する際に、エンドユーザーがお客様のアプリケーションを使用する上で役立ちます。

リージョンの選択の詳細については、リージョンとアベイラビリティーゾーン[5]を参照してください。また、十分な情報に基づいた決定ができるよう、各リージョンの特徴についてアカウントチームにお問い合わせください。

ロードバランシング (BP6)

より大規模な DDoS 攻撃では、1 つの Amazon EC2 インスタンスのサイズを超過することがあります。このような攻撃を緩和するため、過剰なトラフィックのロードバランシングのオプションについて考慮できます。Elastic Load Balancing (ELB) では、多くのバックエンドインスタンスにトラフィックを分配することにより、アプリケーションに過剰な負荷がかかるリスクを軽減できます。ELB は自動的なスケーリングを行い、フラッシュクラウドや DDoS 攻撃など、より大量の予期しないトラフィックを処理できます。

ELB は正しい形式の TCP 接続のみを受け取ります。つまり、SYN フラッドや UDP リフレクション攻撃など多くの一般的な DDoS 攻撃は、ELB に受け取られず、アプリケーションにも渡されません。ELB がこれらのタイプの攻撃を検出したら、自動的にスケールして追加のトラフィックを吸収しますが、追加の課金は発生しません。

ELB を使用して負荷を分散し Amazon EC2 インスタンスを保護することの詳細については、Elastic Load Balancing の開始方法[6] を参照してください。

AWS エッジロケーションを使用した大規模配信 (BP1, BP3)

スケーリングが高く多様なインターネット接続にアクセスすることで、レイテンシーとエンドユーザーへのスループットを最適化し、DDoS 攻撃を吸収し、可用性への影響を最小限にとどめながら障害を分離する能力が大幅に向上します。AWS エッジロケーションはネットワークインフラストラクチャの追加のレイヤーを提供し、Amazon CloudFront および Amazon Route 53 を使用するウェブアプリケーションにこれらのメリットをもたらします。これらのサービスにより、通常エンドユーザーにより近い場所からコンテンツは提供され DNS クエリは解決されます。

エッジでのウェブアプリケーションの配信 (BP1)

Amazon CloudFront はコンテンツ配信ネットワーク (CDN) サービスで、静的、動的、ストリーミング、インタラクティブコンテンツなどのウェブサイト全体の配信に使用できます。持続的な TCP 接続と変更可能な有効期限 (TTL) を使用して、エッジロケーションでキャッシュできない場合でも、コンテンツの配信を高速化できます。これにより、静的コンテンツを提供していなくても、Amazon CloudFront を使用してウェブアプリケーションを保護できます。SYN フラッドや UDP リフレクション攻撃など多くの一般的な DDoS 攻撃がオリジンに達するのを防ぐため、Amazon CloudFront は正しい形式の接続のみを受け取ります。DDoS 攻撃は地理的にソースの近辺に隔離されるので、トラフィックがその他の場所に影響することを防ぎます。このような機能により、より大規模な DDoS 攻撃の間、エンドユーザーへトラフィックを処理し続ける能力は大幅に向上します。Amazon CloudFront を使用して AWS またはその他のインターネット上の場所にあるオリジンを保護できます。

Amazon CloudFront を使用してウェブアプリケーションのパフォーマンスを最適化することの詳細については、「CloudFront の使用開始[7]」を参照してください。

エッジでのドメイン名の解決 (BP3)

Amazon Route 53 は可用性と拡張性が非常に高いドメインネームシステム (DNS) サービスで、トラフィックをウェブアプリケーションへダイレクトするために使用できます。トラフィックフロー、レイテンシーに基づくルーティング、Geo DNS、ヘルスチェック、およびモニタリングなど、多くの高度な機能が含まれています。これらの機能では、レイテンシー、ヘルス、その他の考慮すべき要素に応じて最適化するために、サービスが DNS リクエストにどのように応答するかを管理できます。これらの機能を使用して、ウェブアプリケーションのパフォーマンスを改善し、サイトの機能停止を避けることができます。

Amazon Route 53 はシャッフルシャーディングとエニーキャストストライピングを使用して、DNS サービスが DDoS 攻撃の標的になっても、エンドユーザーがアプリケーションにアクセスできるようにします。シャッフルシャーディングでは、委託セットの各ネームサーバーがエッジロケーションおよびインターネットパスの一意のセットに対応します。これにより、耐障害性が向上し、お客様間の重複が最小化されます。委託セットの 1 つのネームサーバーが利用できない場合、エンドユーザーは異なるエッジロケーションの別のネームサーバーを再試行し、レスポンスを受け取ることができます。エニーキャストストライピングは、各 DNS リクエストを最適な場所で処理するために使用されます。これには負荷を分散して DNS レイテンシーを減らす効果があり、エンドユーザーはより迅速にレスポンスを受け取ることができます。さらに、Amazon Route 53 は DNS クエリのソースとボリュームの異常を検出し、信頼できることが分かっているユーザーからのリクエストを優先させることができます。

Amazon Route 53 のホストゾーンが多くある場合は、再利用可能な委託セットを作成し、各ドメインに正式なネームサーバーの同じセットを提供できます。これにより、ホストゾーンの維持がより簡単にできます。DDoS 攻撃の際は、AWS が再利用可能な委託セットが使用されているホストゾーンすべてに対応する単一の緩和方法を適用することもできます。

Amazon Route 53 を使用してエンドユーザーをアプリケーションにダイレクトすることの詳細については、「Amazon Route 53 の開始方法[8]」を参照してください。再利用可能な委託セットの詳細については、「再利用可能な委任セットでのアクション[9]」を参照してください。

アプリケーションレイヤーの防御 (BP1, BP2, BP6)

本書で説明している手法の多くはインフラストラクチャレイヤー DDoS 攻撃による可用性への影響を緩和するのに効果的です。アプリケーションレイヤー攻撃からアプリケーションを防御するには、悪意のあるリクエストを検出し、スケールして吸収し、ブロックできるアーキテクチャを実装することが必要です。一般的にネットワークベースの DDoS 緩和システムは複雑なアプリケーションレイヤー攻撃の緩和には効果がないため、この点を考慮するのは重要です。

悪意のあるウェブリクエストの検出とフィルタリング (BP1, BP2)

ウェブアプリケーションファイアーウォール (WAF) は、アプリケーションの脆弱性を悪用しようとする攻撃からウェブアプリケーションを保護するためによく使用されます。一般的な例としては、SQL インジェクションまたはクロスサイトリクエストフォージェリがあります。WAF を使用してウェブアプリケーションレイヤー DDoS 攻撃を検出し緩和することもできます。

AWS では、Amazon CloudFront および AWS WAF を使用してアプリケーションをこれらの攻撃から保護できます。Amazon CloudFront では、静的コンテンツをキャッシュして AWS エッジロケーションから提供でき、オリジンへの負荷を減らすのに役立ちます。さらに、Amazon CloudFront では、読み込みや書き込みが遅い攻撃者 (例: Slowloris) からの接続を自動的に閉じることができます。Amazon CloudFront の地域制限を使用して、特定の地理的場所のユーザーからのコンテンツへのアクセスを制限できます。これは、エンドユーザーにサービスを提供するつもりのない地理的な場所から来る攻撃をブロックする場合に有効です。

HTTP フラッドまたは WordPress ピンバックフラッドなどのその他のタイプの攻撃には、AWS WAF を使用して独自の緩和方法を作成できます。ブロックしたいソースの IP アドレスが分かっていれば、ブロックするアクションのルールを作成してウェブ ACL に関連付けることができます。その後、攻撃に参加しているソース IP アドレスをブロックするために、ウェブ ACL に IP アドレス一致の条件を作成できます。また、URI、クエリ文字列、HTTP メソッド、またはヘッダーキーによりブロックする条件をつけたルールを作成することもできます。後者の方法は、明確な署名がある攻撃の場合に有効です。たとえば、WordPress ピンバック攻撃には、必ず User-Agent に "WordPress" という語があります。

DDoS 攻撃の署名を見分けること、または、攻撃に参加している IP アドレスを正確に特定するのは難しい場合があります。ウェブサーバーのログを確認することでこの情報を入手できる場合もあります。AWS WAF コンソールを使用して Amazon CloudFront が AWS WAF に転送したリクエストの例を表示することもできます。リクエストの例は、アプリケーションレイヤー攻撃を緩和するために必要なのはどのようなルールかを決めるのに役立ちます。ランダムなクエリ文字列の多くのリクエストがある場合は、Amazon CloudFront でクエリ文字列の転

送を無効にする決定ができます。これは、オリジンへのキャッシュバスティング攻撃を緩和するのに有効です。

通常のエンドユーザーのトラフィックのように見せかけたウェブトラフィックによる攻撃もあります。このタイプの攻撃を緩和するには、AWS Lambda 関数を使用してレートに基づくブラックリストを実装できます。レートに基づくブラックリストでは、ウェブアプリケーションが処理できるリクエストの数のしきい値を設定できます。ボットまたはクローラーがこの制限を超過すると、AWS WAF を使用してそれ以上のリクエストを自動的にブロックできます。

Amazon CloudFront ディストリビューションを制限するための地理制限の使用の詳細については、「コンテンツの地理的ディストリビューションの制限[10]」を参照してください。

AWS WAF の使用の詳細については、「AWS WAF の開始方法[11]」および「CloudFront が AWS WAF に転送したリクエストの例の表示[12]」を参照してください。

AWS Lambda および AWS WAF でのレートに基づくブラックリストの設定方法の詳細については、「AWS WAF および AWS Lambda でのレートに基づくブラックリストの設定方法[13]」を参照してください。

スケールして吸収する (BP6)

アプリケーションレイヤー攻撃に対処する別の方法はスケーリングの運用です。ウェブアプリケーションの場合、ELB を使用すれば、フラッシュクラウドまたはアプリケーションレイヤー DDoS 攻撃のいずれの原因であれ、トラフィックの急増に対処するために過剰プロビジョニングされた、または自動スケーリングが設定された多くの Amazon EC2 インスタンスへトラフィックを分配できます。Amazon CloudWatch アラームを使用して Auto Scaling を開始すると、定義したイベントに応じて Amazon EC2 群のサイズを自動的にスケールします。これにより、予期しない量のリクエストを処理する場合でも、アプリケーションの可用性が保護されます。Amazon CloudFront または ELB を使用することで、SSL ネゴシエーションはディストリビューションまたはロードバランサーにより行われ、インスタンスが SSL ベースの攻撃に影響を受けないようにします。

Auto Scaling を開始するための Amazon CloudWatch の使用の詳細については、「Amazon CloudWatch を使用したインスタンスおよびグループの Auto Scaling のモニタリング[14]」を参照してください。

攻撃領域の削減

AWS の構築の際に考慮できる別の重要な点は、攻撃者がアプリケーションを標的にする機会を制限することです。たとえば、エンドユーザーが特定のリソースに直接アクセスすることを想定していなければ、インターネット上からそれらのリソースへのアクセスが決してできないようにします。同様に、エンドユーザーまたは外部アプリケーションがアプリケーションの特定のポートやプロトコルにアクセスすることを想定していなければ、トラフィックを決して受け取らないようにします。この考えは攻撃対象領域の削減として知られています。このセクションでは、攻撃領域の削減およびアプリケーションがインターネットにさらされる度合いの制限に役立つベストプラクティスを紹介します。インターネットにさらされていないリソースは攻撃がより難しくなり、攻撃者がアプリケーションの可用性を標的にするための手段が限られることになります。

AWS リソースの難読化 (BP1, BP4, BP5)

多くのアプリケーションで、AWS リソースはインターネットに完全に公開される必要はありません。たとえば、ELB の背後の Amazon EC2 インスタンスは、パブリックにアクセス可能である必要はないでしょう。このシナリオでは、エンドユーザーが ELB に特定の TCP ポートでアクセスできるようにし、ELB だけが Amazon EC2 インスタンスと通信できるようにするかもしれません。これは、Amazon Virtual Private Cloud (VPC) 内でセキュリティグループとネットワークアクセスコントロールリスト (NACL) を作成することにより達成できます。Amazon VPC では、AWS クラウドの論理的に隔離されたセクションを使用可能です。そこでは、ユーザーが定義した仮想ネットワーク内で AWS リソースを起動することができます。

セキュリティグループとネットワーク ACL は、VPC 内での AWS リソースへのアクセスの管理ができるという点で似ています。セキュリティグループは、インスタンスレベルでインバウンドトラフィックとアウトバウンドトラフィックを制御でき、ネットワーク ACL は同様の機能を VPC レベルで提供します。さらに、Amazon EC2 セキュリティグループ (SG) ルールまたはネットワーク ACL でのインバウンドデータ転送に料金は発生しません。これにより、セキュリティグループまたはネットワーク ACL によりドロップされたトラフィックに追加料金が発生しないようにできます。

セキュリティグループ (BP5)

インスタンスを起動する際にセキュリティグループを指定するか、または後でインスタンスにセキュリティグループを関連付けることができます。許可ルールを決めてトラフィックを許可しない限り、インターネットからセキュリティグループへのトラフィックはすべて暗黙的に拒否されます。たとえば、ELB と多くの Amazon EC2 インスタンスから成るウェブアプリケーションがある場合、ELB に 1 つのセキュリティグループ (「ELB セキュリティグループ」) を、インスタンスにもう 1 つのセキュリティグループ (「ウェブアプリケーションサーバーセキュリティグループ」) を作成できます。その後、許可ルールを作成し、インターネットから ELB セキュリティグループへのトラフィックを許可し、ELB セキュリティグループからウェブアプリケーションサーバーセキュリティグループへのトラフィックを許可できます。その結果、インターネットからのトラフィックは Amazon EC2 インスタンスと直接通信できなくなり、攻撃者がアプリケーションについての情報を得にくくなります。

ネットワークアクセスコントロールリスト (ACL) (BP5)

ネットワーク ACL では、許可および拒否の両方のルールを設定できます。これは、アプリケーションへの特定のタイプのトラフィックを明示的に拒否する場合に役立ちます。たとえば、サブネット全体で拒否する IP アドレス (CIDR 範囲)、プロトコル、および送信先ポートを指定できます。アプリケーションを TCP トラフィックにのみ使用している場合、すべての UDP トラフィックを拒否するルールを作成できますし、その逆もできます。ソース IP アドレスまたはその他の署名を知っているなら、独自のルールを作成して攻撃を緩和できるので、DDoS 攻撃へ対応する場合にこのツールは役立ちます。

オリジンの保護 (BP1)

VPC 内のオリジンで Amazon CloudFront を使用している場合、AWS Lambda 関数を使用して Amazon CloudFront からのトラフィックだけを許可するようセキュリティグループルールを自動的に更新します。これにより、Amazon CloudFront と AWS WAF をバイパスすることはできなくなり、オリジンの安全性を向上できます。

セキュリティグループの自動更新によるオリジンの保護の詳細については、「AWS Lambda を使用して Amazon CloudFront および AWS WAF のためにセキュリティグループを自動更新する方法[15]」を参照してください。

Amazon CloudFront ディストリビューションだけがオリジンにリクエストを転送していることを確認することもできます。Edge-to-Origin のリクエストヘッダーで、Amazon CloudFront からオリジンにリクエストが転送された場合、既存のリクエストヘッダーの値の追加またはオーバーライドを行えます。X-Shared-Secret ヘッダーを使用すると、オリジンに対して行われたリクエストが Amazon CloudFront から送信されたものであることを検証するのに役立ちます。

X-Shared-Secret ヘッダーでオリジンを保護することの詳細については、「カスタムヘッダーをオリジンへ転送する[16]」を参照してください。

API エンドポイントの保護 (BP4)

通常、API を一般公開する必要がある場合、API フロントエンドが DDoS 攻撃の標的になる危険性があります。Amazon API Gateway は完全マネージド型サービスで、Amazon EC2 や AWS Lambda で実行するアプリケーション、またはその他のウェブアプリケーションの正面玄関のような役割を果たす API を作成できます。Amazon API Gateway では、API フロントエンドのためにサーバーを独自に運用する必要はなく、アプリケーションのその他のコンポーネントをパブリックから難読化できます。これは、AWS リソースが DDoS 攻撃の標的になるのを避けるのに役立ちます。Amazon API Gateway は Amazon CloudFront に統合されていて、その特徴であるさらなる DDoS 弾力性の利点を受けられます。REST API で各メソッドに標準またはバーストレートの制限を設定することで、バックエンドを過剰なトラフィックから保護できます。

Amazon API Gateway での API の作成の詳細については、「Amazon API Gateway の開始方法[17]」を参照してください。

オペレーションテクニック

本書にある緩和のテクニックで、DDoS 攻撃に対する弾力性を本来備えたアプリケーションを構築できます。多くの場合、DDoS 攻撃がいつアプリケーションを標的にするかを知り、このデータに基いてアクションを取るのは効果的です。脅威を評価し、アプリケーションのアーキテクチャを確認し、その他のサポートを受けるため、他のリソースを活用することもできます。このセクションでは、異常な動作の可視性を得ること、アラートと自動化、およびその他のサポートのための AWS の使用について説明します。

可視性

アプリケーションの通常の動作を理解すると、異常を検出したときによりすばやくアクションを取ることができます。キーメトリクスが期待値とは大きく異なる場合、攻撃者がアプリケーションの可用性を標的にしていることが考えられます。Amazon CloudWatch を使用して、AWS で実行中のアプリケーションをモニタリングすることができます。メトリクスの収集とトラッキング、ログファイルの収集とモニタリング、アラーム設定、AWS リソースの変更に自動的に対応することができます。DDoS 攻撃を検出し対応するために一般的に使用されている Amazon CloudWatch メトリクスの説明については、表 3 を参照してください。

トピック	メトリクス	説明
Auto Scaling	GroupMaxSize	Auto Scaling グループの最大サイズ
Amazon CloudFront	リクエスト	HTTP/S リクエストの数
Amazon CloudFront	TotalErrorRate	HTTP ステータスコードが 4xx または 5xx であるすべてのリクエストの割合
Amazon EC2	CPUUtilization	割り当てられた EC2 コンピュートユニットのうち、現在使用されているものの比率
Amazon EC2	NetworkIn	すべてのネットワークインターフェースでの、このインスタンスによって受信されたバイトの数
ELB	SurgeQueueLength	ロードバランサーによりキューに入れられ、バックエンドインスタンスが接続を受け入れ処理するのを待っているリクエストの数
ELB	UnHealthyHostCount	各アベイラビリティーゾーンの異常なインスタンス数
ELB	RequestCount	受信され、登録されたインスタンスにルーティングされた、完了したリクエスト数
ELB	レイテンシー	リクエストがロードバランサーから送信され、応答を受信するまでの経過時間 (秒)
ELB	HTTPCode_ELB_4xx HTTPCode_ELB_5xx	ロードバランサーで生成される HTTP 4xx または 5xx エラーコードの数
ELB	BackendConnectionErrors	成功しなかった接続の数
ELB	SpilloverCount	キューがいっぱいなため、拒否されたリクエストの数
Amazon Route 53	HealthCheckStatus	ヘルスチェックのエンドポイントのステータス

表 3: 推奨される Amazon CloudWatch のメトリクス

図 5 に示された DDoS 弾力性のあるリファレンスアーキテクチャに基づいて構築されたアプリケーションでは、一般的なインフラストラクチャレイヤー攻撃はアプリケーションに達する前にブロックされます。結果として、これらの攻撃は Amazon CloudWatch メトリクスに現れません。

アプリケーションレイヤー攻撃はこれらのメトリクスの多くを高くする場合があります。たとえば、HTTP フラッドでは、Amazon CloudFront、ELB、および Amazon EC2 のメトリクスのリクエストと CPU とネットワークの使用率が高くなります。バックエンドインスタンスが過剰なリクエストを処理できない場合は、Amazon CloudFront で TotalErrorRate が、また、ELB で SurgeQueueLength、UnHealthyHostCount、Latency、BackendConnectionErrors、SpilloverCount、または HTTPCode が高くなっているのを確認できます。この場合、アプリケーションが通常のエンドユーザーにサービスを提供できていないため、HTTP リクエストのボリュームは減少しないかもしれません。この状況を解決するには、本書で先述したとおり AWS WAF を使用して、アプリケーションのバックエンドのスケーリング、または過剰なトラフィックのブロックを実行できます。

Amazon CloudWatch を使用してアプリケーションに対する DDoS 攻撃を検知することの詳細については、「Amazon CloudWatch の開始方法[18]」を参照してください。

アプリケーションを標的にしたトラフィックの可視性を得るために使用できる別のツールは VPC フローログです。従来のネットワークでは、ネットワークフローログを使用して、接続およびセキュリティの問題のトラブルシューティングをし、ネットワークアクセスのルールが想定通りに機能していることを確認しました。VPC フローログでは、VPC のネットワークインターフェイスとの間で行き来する IP トラフィックに関する情報をキャプチャできます。

各フローログのレコードには、送信元と送信先の IP アドレス、送信元と送信先のポート、プロトコル、キャプチャウィンドウ中に転送されたパケットおよびバイトの数が含まれます。この情報はネットワークトラフィックの異常を検知し、特定の攻撃進路を見分けるのに使用できます。たとえば、ほとんどの UDP リフレクション攻撃には特定の送信元ポートがあります (例: DNS リフレクション攻撃では送信元ポート 53)。これは、フローログレコードで識別できる明確な署名です。それに応じて、インスタンスレベルで特定の送信元ポートをブロックするか、または、必要がなければプロトコル全体をブロックするためのネットワーク ACL ルールを作成できます。

VPC フローログを使用してネットワークの異常および DDoS 攻撃進路を検出することの詳細については、「VPC フローログ[19]」および「VPC フローログ – ネットワークトラフィックフローのログと確認[20]」を参照してください

サポート

実際に攻撃が起きる前に DDoS 攻撃に対するプランを作成することは重要です。本書で概要が説明されているベストプラクティスは、事前対策を想定してあり、DDoS 攻撃の標的になりうるアプリケーションを起動する前に実装されるべきです。アカウントチームはユースケースおよびアプリケーションを確認し、特定の質問または直面するかもしれない問題においてサポートします。

場合によっては、DDoS 攻撃の際、AWS に連絡し追加のサポートを求めるかもしれません。お客様の事例にはすばやく解決策が示され、エキスパートによるサポートが提供されます。ビジネスサポートに登録すると、毎日 24 時間対応のクラウドサポートエンジニアへ E メール、チャット、電話で相談できます。

ミッションクリティカルなワークロードを AWS で実行しているなら、エンタープライズサポートを検討してください。エンタープライズサポートでは、緊急のケースは最優先され、シニアクラウドサポートエンジニアが対応します。さらに、エンタープライズサポートでは、お客様専属の顧問で技術的な相談を受けるテクニカルアカウントマネージャ (TAM) がつきます。エンタープライズサポートでは、計画されたイベント、製品の導入、および移行の際、リアルタイムでオペレーション関連のサポートを受けられるインフラストラクチャイベント管理へのアクセスも提供されます。

お客様独自のニーズに合わせたサポートプランの選択の詳細については、「AWS サポートプランの比較[21]」を参照してください。

まとめ

本書で概要を説明したベストプラクティスで、DDoS 弾力性のあるアーキテクチャを構築でき、多くの一般的なインフラストラクチャおよびアプリケーションレイヤー DDoS 攻撃からアプリケーションの可用性を保護できます。どの程度これらのベストプラクティスに従ってアプリケーションを構築できるかが、緩和できる DDoS 攻撃のタイプ、方向性、ボリュームに影響します。AWS は、一般的な DDoS 攻撃からアプリケーションの可用性をより保護するため、これらのベストプラクティスを使用することをお勧めします。

寄稿者

本書の執筆に当たり、次の人物および組織が寄稿しました。

- Andrew Kiggins (AWS ソリューションアーキテクト)
- Jeffrey Lyons (AWS DDoS オペレーションエンジニアリング)

注意

[1] https://www.youtube.com/watch?v=OT2y3DzMEmQ

[2] https://www.youtube.com/watch?v=Ys0gG1koqJA

[3] https://aws.amazon.com/ec2/instance-types/

[4] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html

[5] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

[6] http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-getting-started.html

[7] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/GettingStarted.html

[8] http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/getting-started.html

[9] http://docs.aws.amazon.com/Route53/latest/APIReference/actions-on-reusable-delegation-sets.html

[10] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/georestrictions.html

[11] http://docs.aws.amazon.com/waf/latest/developerguide/getting-started.html

[12] http://docs.aws.amazon.com/waf/latest/developerguide/web-acl-testing.html#web-acl-testing-view-sample

[13] https://blogs.aws.amazon.com/security/post/Tx1ZTM4DT0HRH0K/How-to-Configure-Rate-Based-Blacklisting-with-AWS-WAF-and-AWS-Lambda

[14] http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-monitoring.html

[15] https://blogs.aws.amazon.com/security/post/Tx1LPI2H6Q6S5KC/How-to-Automatically-Update-Your-Security-Groups-for-Amazon-CloudFront-and-AWS-W

[16] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/forward-custom-headers.html

[17] https://aws.amazon.com/api-gateway/getting-started/

[18] http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/GettingStarted.html

[19] http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html

[20] https://aws.amazon.com/blogs/aws/vpc-flow-logs-log-and-view-network-traffic-flows/

[21] https://aws.amazon.com/premiumsupport/compare-plans/