DENIAL OF SERVICE ATTACK MITIGATION ON AWS

Overview

A Distributed Denial of Service (DDoS) attack is a malicious attempt to make a targeted system, such as a website or application, unavailable to end users. To achieve this, attackers use a variety of techniques that consume network or other resources, interrupting access for legitimate end users.

AWS provides flexible infrastructure and tools that enable our customers to implement strong DDoS mitigations. In the event of a DDoS attack, AWS customers can leverage multiple capabilities to absorb and deflect unwanted traffic while also working with AWS Support to mitigate the issue. Note that the Business Support and Enterprise Support plans provide prioritized response times and allow customers to quickly engage AWS Support. This document describes common DDoS attack types and provides AWS customers with best practices and considerations for securing cloud resources.

General Best Practices

At its core, DDoS protection and mitigation involves establishing a secure perimeter around your infrastructure and allowing or denying certain traffic based on filters or rules. AWS customers can take advantage of the flexible nature of the cloud and adapt their infrastructure defensively in the event of an attack. When thinking about DDoS security, keep the following best practices in mind:

- **Be ready to scale.** Building infrastructure for scale is fundamental to a well-architected system, however it is also an effective DDoS mitigation technique. Scaling to meet the additional traffic volumes, whether valid or from a DDoS attack, will increase your application’s ability to keep running.

- **Minimize the attack surface area and safeguard exposed resources.** In other words: decouple your infrastructure. For example, customers running public websites should separate the application from the database and, if possible, the media and static content as well. Decoupled applications prevent Internet access to critical system components, protecting them from an attack and enabling teams to focus DDoS mitigation efforts on resources that are publicly accessible.

- **Know what is normal, alert on what is not.** In order to get the best help from AWS Support, it is critical that customers identify traffic correctly. For example, if a customer’s website gets media attention and is suddenly overwhelmed with traffic, blocking that traffic can cause more harm than good. Constant logging and monitoring of infrastructure helps customers quickly identify a legitimate attack and engage AWS.

- **Create a plan for attacks.** Don’t forget: we’re in this together. Customers who suspect they are under attack should immediately request the assistance of AWS. We strongly advise the use of Business Support, at a minimum, to ensure 24/7 access to a Support Engineer. Before contacting AWS, gather the following data:
  - AWS Account Number
  - IDs of affected resources (instances, IP addresses, load balancers, CloudFront distributions, etc.)
  - Nature of the attack (Increased volume? SYN flood? UDP flood?)
  - If the affected resources are accessible
  - If the sources have anything in common (Same IP? Contiguous IP addresses? Same country?)
  - If the traffic can be blocked using a NACL, Security Group, or black-hole routing without impacting customer traffic
  - The type(s) of traffic you are comfortable to have AWS block

1 Adapted from the whitepaper AWS Best Practices for DDoS Resiliency.
The following sections describe key AWS services involved in DDoS attack mitigation and outline mitigation techniques for common application types.

**AWS Services for DDoS Attack Mitigation**

AWS offers globally distributed, high network bandwidth and resilient services that, when used in conjunction with application-specific strategies, are key to mitigating DDoS attacks. For more information on how to leverage each of these services and details on how their various features help protect against DDoS attacks, see the whitepaper AWS Best Practices for DDoS Resiliency.

**Amazon Route 53**

One of the most common targets of DDoS attacks is the Domain Name System (DNS). Amazon Route 53 is a highly available and scalable DNS service designed to route end users to infrastructure running inside or outside of AWS. Route 53 makes it possible to manage traffic globally through a variety of routing types, and provides out-of-the-box shuffle sharding and Anycast routing capabilities to protect domain names from DNS-based DDoS attacks.

**Amazon CloudFront**

Amazon CloudFront distributes traffic across multiple Points of Presence (PoP) locations and filters requests to ensure that only valid HTTP(S) requests will be forwarded to backend hosts. CloudFront also supports geo restriction, also known as *geoblocking*, which can be useful for isolating attacks originating from a particular geographic location.

**AWS WAF**

AWS WAF is a web application firewall that helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. Using AWS WAF, customers can define customizable web security rules that control which traffic accesses their web applications. Web security rules that target specific DDoS request patterns can be very effective for minimizing the effect of a DDoS attack.

**Elastic Load Balancing**

Elastic Load Balancing (ELB) enables the automatic distribution of application traffic to several Amazon Elastic Compute Cloud (Amazon EC2) instances across multiple Availability Zones, which minimizes the risk of overloading a single EC2 instance. Elastic Load Balancing, like CloudFront, only supports valid TCP requests, so DDoS attacks such as UDP and SYN floods are not able to reach EC2 instances. It also offers a single point of management and can serve as a line of defense between the Internet and your backend, private EC2 instances.

**VPCs and Security Groups**

Amazon Virtual Private Cloud (Amazon VPC) allows customers to configure subnet routes, public IP addresses, security groups, and network access control lists in order to minimize application attack surfaces. ELB load balancers and EC2 instance security groups can be configured to allow only traffic that originates from specific IP addresses, such as that from CloudFront or AWS WAF, protecting backend application components from a direct attack.

**Mitigation Approaches**

In general, there are three major application patterns that we see from customers: *web*, *non-web and load balanceable*, and *non-web and non-load balanceable*. The following sections describe high-level DDoS attack mitigation approaches for each of these application patterns, and include reference architectures that portray the high-level configuration of related services. Note that these architectures assume a highly available design that uses subnets in multiple Availability Zones. For more detailed information, please see the AWS Best Practices for DDoS Resiliency whitepaper.
Web Applications

The reference architecture below represents a stateless web application that relies on HTTP/S for communication, such as a website, web-based API, or mobile application.

This approach leverages Amazon Route 53, AWS WAF, CloudFront, and Elastic Load Balancing to control and distribute traffic. Security groups or origin access identity (OAI) can also help minimize the attack surface of backend load balancers, EC2 instances, or Amazon Simple Storage Service (Amazon S3) buckets because they require attackers to make requests through AWS WAF and CloudFront rather than directly from the website origin.

In addition to AWS WAF, the AWS Marketplace offers additional web application firewall products that can be combined with externally facing and internally facing ELB load balancers to provide additional security. These include out-of-the-box web-filtering capabilities that protect against application-level exploits such as cross-site scripting (XSS) and SQL injection (SQLi), and additional features such as automatic evaluation against IP reputation lists or redirection to Completely Automated Public Turning test to tell Computers and Humans Apart (CAPTCHA) tests.

Non-Web, Load Balanceable Applications

The reference architecture below represents a client-server application that requires a TCP connection and general host or session affinity, such as an application using the WebSocket protocol. In this case, the client and the server have a stateful relationship, and the application requires that the client communicates with the same server during the session.
This approach leverages Amazon Route 53 and Elastic Load Balancing to control and distribute traffic. The AWS Marketplace offers additional firewall and intrusion-detection products that can be combined with externally facing and internally facing ELB load balancers to provide additional out-of-the-box traffic monitoring and filtering capabilities to help detect and filter invalid requests.

Another method to mitigate DDoS attacks against TCP-based applications is to vertically scale EC2 instances and leverage instances that support enhanced networking in order to absorb more connections and traffic. Additionally, host-based IDS/IPS agents can validate the incoming traffic against a rule set and make a decision to serve it or not. When combined with monitoring, timely alerting, and host-based agents to isolate offending traffic, larger instances sizes can provide relief to a targeted application until AWS Support is engaged for additional assistance.

Non-Web, Non-Load Balanceable Applications
The reference architecture below represents a TCP- or UDP-based, non-HTTP service or application, such as DNS, FTP, or a gaming application where user-to-host persistence is required for the users to be able to interact with each other in real-time or near real-time.

This approach leverages Amazon Route 53 and a highly available, scalable routing service to control and distribute traffic. This service acts as a queue for users who are ultimately grouped together by specific attributes. For example, in an online gaming scenario, the routing service acts as a lobby where players are directed to a particular application server according to their skill level. The routing application itself should be either a web application or a non-web, load balanceable application and should follow the previously mentioned techniques for DDoS mitigation.

Another method to mitigate DDoS attacks on these types of applications is to vertically scale EC2 instances and leverage instances that support enhanced networking in order to absorb more connections and traffic. Security groups can be configured to allow only traffic that originates from specific IP addresses, further protecting EC2 instances. Additionally, host-based IDS/IPS agents can also be used to validate and serve or deny incoming traffic against a rule set. Finally, additional application-connection logic can be introduced to the application’s architecture to validate incoming users through a highly-available lobby or routing server before providing clients with backend server connection details. This approach, especially when combined with dynamically assigned public IP addresses, obscures the backend servers from potential attackers.
Resources

**AWS Best Practices for DDoS Resiliency**
AWS whitepaper on best practices to improve resiliency against DDoS attacks

**Elastic Load Balancing Product Website**
https://aws.amazon.com/elasticloadbalancing/

**Amazon Route 53 Product Website**
https://aws.amazon.com/route53/

**AWS WAF Product Website**
https://aws.amazon.com/waf/

**AWS WAF Security Automations**
AWS-provided solution that enables customers to automatically deploy a set of AWS WAF rules designed to filter common web-based attacks

**Amazon CloudFront Product Website**
https://aws.amazon.com/cloudfront/

**EC2 Product Details**
https://aws.amazon.com/ec2/details/
AWS webpage describing EC2 features, including links to documentation and FAQs on Enhanced Networking.