AWS Best Practices for DDoS Resiliency

First Published June 2015
Updated September 21, 2021

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

It’s important to protect your business from the impact of Distributed Denial of Service (DDoS) attacks, as well as other cyberattacks. Keeping customer trust in your service by maintaining the availability and responsiveness of your application is high priority. You also want to avoid unnecessary direct costs when your infrastructure must scale in response to an attack. Amazon Web Services (AWS) is committed to providing you with the tools, best practices, and services to defend against bad actors on the internet. Using the right services from AWS helps ensure high availability, security, and resiliency.

In this whitepaper, AWS provides you with prescriptive DDoS guidance to improve the resiliency of applications running on AWS. This includes a DDoS-resilient reference architecture that can be used as a guide to help protect application availability. This whitepaper also describes different attack types, such as infrastructure layer attacks and application layer attacks. AWS explains which best practices are most effective to manage each attack type. In addition, the services and features that fit into a DDoS mitigation strategy are outlined and how each one can be used to help protect your applications is explained.

This paper is intended for IT decision makers and security engineers who are familiar with the basic concepts of networking, security, and AWS. Each section has links to AWS documentation that provides more detail on the best practice or capability.

Introduction

Denial of Service Attacks

A Denial of Service (DoS) attack is a deliberate attempt to make a website or application unavailable to users, such as by flooding it with network traffic. Attackers use a variety of techniques that consume large amounts of network bandwidth or tie up other system resources, disrupting access for legitimate users. In its simplest form, a lone attacker uses a single source to carry out a DoS attack against a target, as shown in the following image.

Diagram of a DoS Attack

In a DDoS attack, an attacker uses multiple sources to orchestrate an attack against a target. These sources can include distributed groups of malware infected computers, routers, IoT devices, and other endpoints. The following diagram shows a network of compromised host participates in the attack, generating a flood of packets or requests

to overwhelm the target.

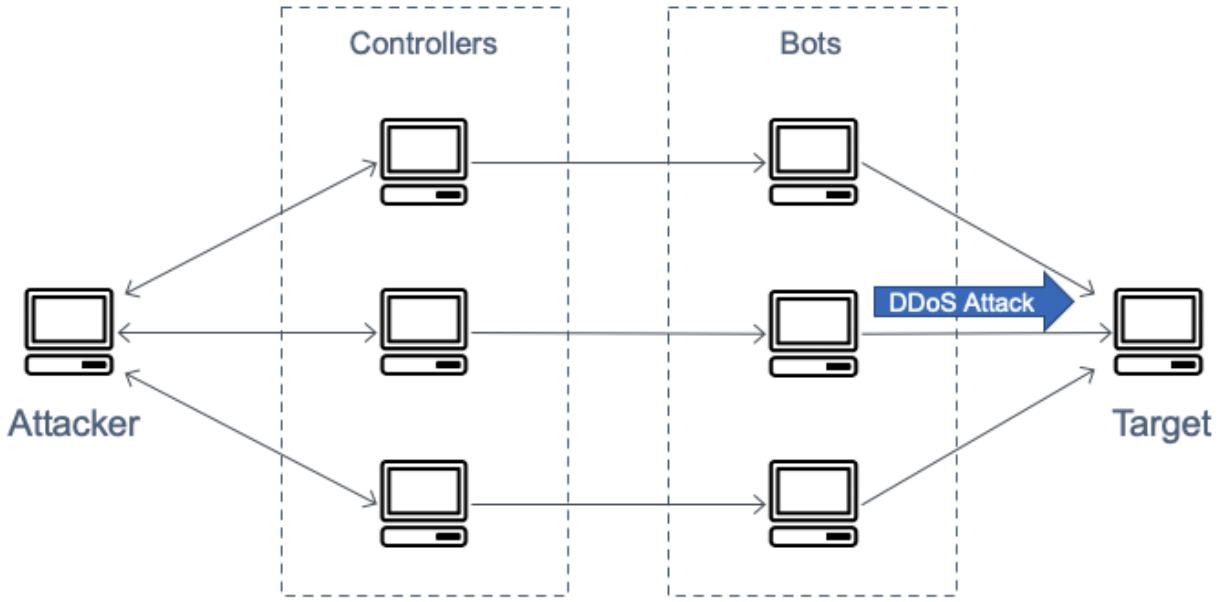


Diagram of a DDoS Attack

There are seven layers in the Open Systems Interconnection (OSI) model and they are described in the *Open Systems Interconnection (OSI) Model* table. DDoS attacks are most common at layers three, four, six, and seven. Layer three and four attacks correspond to the Network and Transport layers of the OSI model. Within this paper, AWS refers to these collectively as *infrastructure layer attacks*. Layers six and seven attacks correspond to the Presentation and Application layers of the OSI model. AWS will address these together as *application layer attacks*. Examples of these attack types are discussed in the following sections.

Open Systems Interconnection (OSI) Model

#	Layer	Unit	Description	Vector Examples
7	Application	Data	Network process to application	HTTP floods, DNS query floods
6	Presentation	Data	Data representation and encryption	TLS abuse

#	Layer	Unit	Description	Vector Examples
5	Session	Data	Interhost communication	N/A
4	Transport	Segments	End-to-end connections and reliability	SYN floods
3	Network	Packets	Path determination and logical addressing	UDP reflection attacks
2	Data Link	Frames	Physical addressing	N/A
1	Physical	Bits	Media, signal, and binary transmission	N/A

Infrastructure Layer Attacks

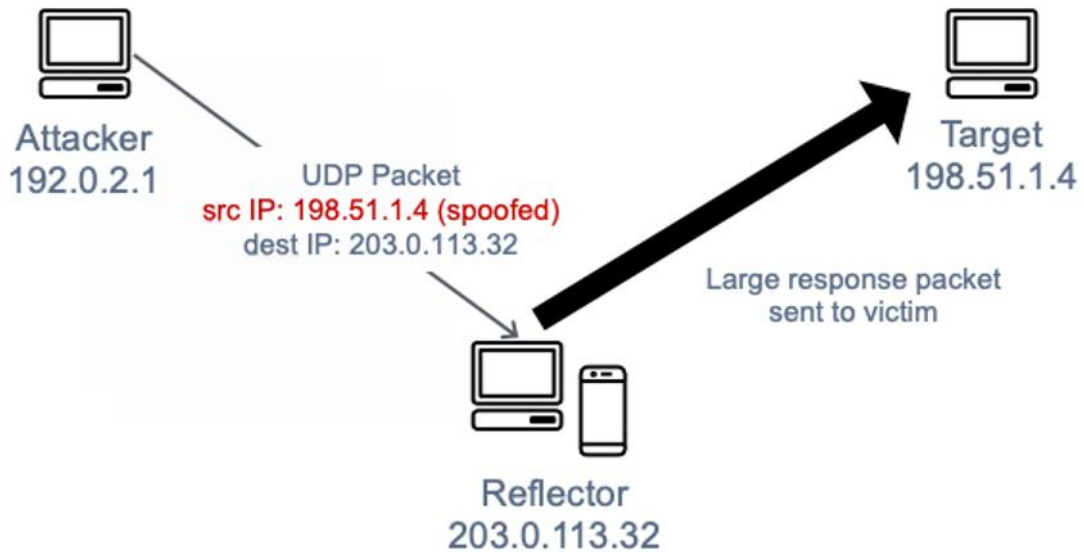
The most common DDoS attacks, User Datagram Protocol (UDP) reflection attacks and synchronize (SYN) floods, are infrastructure layer attacks. An attacker can use either of these methods to generate large volumes of traffic that can inundate the capacity of a network or tie up resources on systems such as servers, firewalls, intrusion prevention system (IPS), or load balancers. While these attacks can be easy to identify, to mitigate them effectively, you must have a network or systems that scale up capacity more rapidly than the inbound traffic flood. This extra capacity is necessary to either filter out or absorb the attack traffic freeing up the system and application to respond to legitimate customer traffic.

UDP Reflection Attacks

User Datagram Protocol (UDP) reflection attacks exploit the fact that UDP is a stateless protocol. Attackers can craft a valid UDP request packet listing the attack target's IP address as the UDP source IP address. The attacker has now falsified—spoofed—the UDP request packet's source IP. The UDP packet contains the spoofed source IP and is sent by the attacker to an intermediate server. The server is tricked into sending its UDP response packets to the targeted victim IP rather than back to the attacker's IP address. The intermediate server is used because it generates a response that is several times larger than the request packet, effectively amplifying the amount of attack traffic sent to the target IP address.

The amplification factor is the ratio of response size to request size and it varies depending on which protocol the attacker uses: DNS, NTP, SSDP, CLDAP,

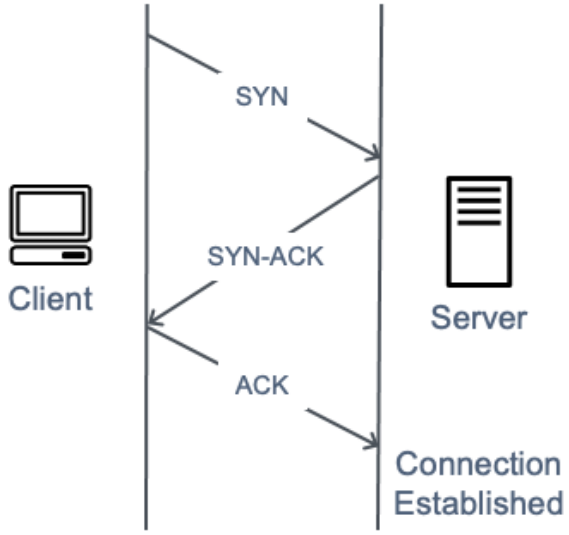
Memcached, CharGen, or QOTD. For example, the amplification factor for DNS can be 28 to 54 times the original number of bytes. So, if an attacker sends a request payload of 64 bytes to a DNS server, they can generate over 3400 bytes of unwanted traffic to an attack target. UDP reflection attacks are accountable for larger volume of traffic in comparison to other attacks. The UDP Reflection Attack figure illustrates the reflection tactic and amplification effect.



UDP Reflection Attack

SYN Flood Attacks

When a user connects to a Transmission Control Protocol (TCP) service, such as a web server, their client sends a SYN synchronization packet. The server returns a SYN-ACK packet in acknowledgement, and finally the client responds with an acknowledgement (ACK) packet, which completes the expected three-way handshake. The following image illustrates this typical handshake.



SYN 3-way Handshake

In a SYN flood attack, a malicious client sends a large number of SYN packets, but never sends the final ACK packets to complete the handshakes. The server is left waiting for a response to the half-open TCP connections and eventually runs out of capacity to accept new TCP connections. This can prevent new users from connecting to the server. The attack is trying to tie up available server connections so that resources are not available for legitimate connections. While SYN floods can reach up to hundreds of Gbps, the purpose of the attack is not to increase SYN traffic volume.

Application Layer Attacks

An attacker may target the application itself by using a layer 7 or application layer attack. In these attacks, similar to SYN flood infrastructure attacks, the attacker attempts to overload specific functions of an application to make the application unavailable or unresponsive to legitimate users. Sometimes this can be achieved with very low request volumes that generate only a small volume of network traffic. This can make the attack difficult to detect and mitigate. Examples of application layer attacks include HTTP floods, cache-busting attacks, and WordPress XML-RPC floods.

In an **HTTP flood attack**, an attacker sends HTTP requests that appear to be from a valid user of the web application. Some HTTP floods target a specific resource, while more complex HTTP floods attempt to emulate human interaction with the application.

This can increase the difficulty of using common mitigation techniques like request rate limiting.

Cache-busting attacks are a type of HTTP flood that use variations in the query string to circumvent content delivery network (CDN) caching. Instead of being able to return cached results, the CDN must contact the origin server for every page request, and these origin fetches cause additional strain on the application web server.

With a **WordPress XML-RPC flood attack**, also known as a WordPress pingback flood, an attacker targets a website hosted on the WordPress content management software. The attacker misuses the XML-RPC API function to generate a flood of HTTP requests. The pingback feature allows a website hosted on WordPress (Site A) to notify a different WordPress site (Site B) through a link that Site A has created to Site B. Site B then attempts to fetch Site A to verify the existence of the link. In a pingback flood, the attacker misuses this capability to cause Site B to attack Site A. This type of attack has a clear signature: **WordPress** is typically present in the **User-Agent** of the HTTP request header.

There are other forms of malicious traffic that can impact an application's availability. **Scraper bots** automate attempts to access a web application to steal content or record competitive information, such as pricing. **Brute force** and **credential stuffing** attacks are programmed efforts to gain unauthorized access to secure areas of an application. These are not strictly DDoS attacks; but their automated nature can look similar to a DDoS attack and they can be mitigated by implementing some of the same best practices covered in this paper.

Application layer attacks can also target Domain Name System (DNS) services. The most common of these attacks is a **DNS query flood** in which an attacker uses many well-formed DNS queries to exhaust the resources of a DNS server. These attacks can also include a cache-busting component where the attacker randomizes the subdomain string to bypass the local DNS cache of any given resolver. As a result, the resolver can't take advantage of cached domain queries and must instead repeatedly contact the authoritative DNS server, which amplifies the attack.

If a web application is delivered over Transport Layer Security (TLS), an attacker can also choose to attack the TLS negotiation process. TLS is computationally expensive so an attacker, by generating extra workload on the server to process unreadable data (or unintelligible (ciphertext)) as a legitimate handshake, can reduce server's availability. In a variation of this attack, an attacker completes the TLS handshake but perpetually renegotiates the encryption method. An attacker can alternatively attempt to exhaust server resources by opening and closing many TLS sessions.

Mitigation Techniques

Some forms of DDoS mitigation are included automatically with AWS services. DDoS resilience can be improved further by using an AWS architecture with specific services, covered in the following sections, and by implementing additional best practices for each part of the network flow between users and your application.

All AWS customers can benefit from the automatic protections of AWS Shield Standard at no additional charge. AWS Shield Standard defends against the most common and frequently occurring network and transport layer DDoS attacks that target your website or applications. This protection is always on, pre-configured, static, and provides no reporting or analytics. It is offered on all AWS services and in every AWS Region. In AWS Regions, DDoS attacks are detected and the Shield Standard system automatically baselines traffic, identifies anomalies, and, as necessary, creates mitigations. You can use AWS Shield Standard as part of a DDoS-resilient architecture to protect both web and non-web applications.

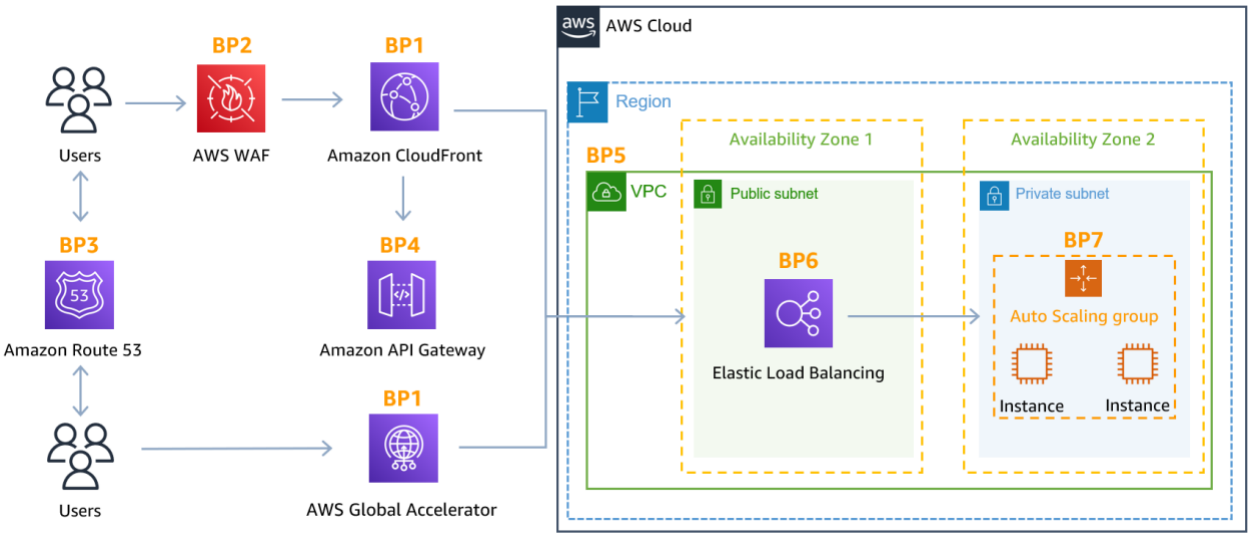
You can also utilize AWS services that operate from edge locations, such as Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 to build comprehensive availability protection against all known infrastructure layer attacks. These services are part of the AWS Global Edge Network and can improve the DDoS resiliency of your application when serving any type of application traffic from edge locations distributed around the world. You can run your application in any AWS Region and use these services to protect your application availability and optimize the performance of your application for legitimate end users.

Benefits of using CloudFront, AWS Global Accelerator, and Amazon Route 53 include:

- Access to internet and DDoS mitigation capacity across the AWS Global Edge Network. This is useful in mitigating larger volumetric attacks, which can reach terabit scale.
- AWS Shield DDoS mitigation systems are integrated with AWS edge services, reducing time-to-mitigate from minutes to sub second.
- Stateless SYN Flood mitigation techniques proxy and verify incoming connections before passing them to the protected service. This ensures that only valid connections reach your application while protecting your legitimate end users against false positives drops.

- Automatic traffic engineering systems that disperse or isolate the impact of large volumetric DDoS attacks. All of these services isolate attacks at the source before they reach your origin, which means less impact on systems protected by these services.
- Application layer defense when combined with AWS WAF that does not require changing current application architecture (for example, in an AWS Region or on-premises data center).

There is no charge for inbound data transfer on AWS and you do not pay for DDoS attack traffic that is mitigated by AWS Shield. The following architecture diagram includes AWS Global Edge Network services.



DDoS-resilient reference architecture

This architecture includes several AWS services that can help you improve your web application's resiliency against DDoS attacks. The *Summary of Best Practices* table provides a summary of these services and the capabilities that they can provide. AWS has tagged each service with a best practice indicator (BP1, BP2) for easier reference within this document. For example, an upcoming section discusses the capabilities provided by CloudFront and Global Accelerator that includes the best practice indicator BP1.

Summary of Best Practices

	AWS EDGE			AWS REGION		
	Using Amazon CloudFront (BP1) with AWS WAF (BP2)	Using AWS Global Accelerator (BP1)	Using Amazon Route 53 (BP3)	Using Elastic Load Balancing (BP6) with AWS WAF (BP2)	Using Security Groups and network ACLs in Amazon VPC (BP5)	Using Amazon EC2 Auto Scaling (BP7)
Layer 3 (for example, UDP reflection) attack mitigation	✔	✔	✔	✔	✔	✔
Layer 4 (for example, SYN flood) attack mitigation	✔	✔	✔	✔		
Layer 6 (for example, TLS) attack mitigation	✔	✔	✔	✔		
Reduce attack surface	✔	✔	✔	✔	✔	
Scale to absorb application layer traffic	✔	✔	✔	✔	✔	✔
Layer 7 (application layer) attack mitigation	✔	✔ (*)	✔	✔	✔ (*)	✔ (*)
Geographic isolation and dispersion of excess traffic and larger DDoS attacks	✔	✔	✔			

** If used with AWS WAF with AWS Application Load Balancer*

Another way to improve your readiness to respond to and mitigate DDoS attacks is by subscribing to AWS Shield Advanced.

Customers receive tailored detection based on:

- Specific traffic patterns of your application.
- Protection against Layer 7 DDoS attacks including AWS WAF at no additional cost.
- Access to 24x7 specialized support from the AWS SRT.
- Centralized management of security policies through AWS Firewall manager.
- Cost protection to safeguard against scaling charges resulting from DDoS-related usage spikes.

This optional DDoS mitigation service helps protect applications hosted on any AWS Region. The service is available globally for CloudFront, Amazon Route 53, and Global Accelerator. Using AWS Shield Advanced with Elastic IP addresses allows you to protect Network Load Balancer (NLBs) or Amazon EC2 instances.

Benefits of using AWS Shield Advanced include:

- Access to the AWS SRT for assistance with mitigating DDoS attacks that impact application availability.
- DDoS attack visibility by using the AWS Management Console, API, and Amazon CloudWatch metrics and alarms.
- Access to the history of all DDoS events from the past 13 months.
- Access to AWS web application firewall (WAF) at no additional cost for the mitigation of application layer DDoS attacks (when used with CloudFront or Application Load Balancer).
- Automatic baselining of web traffic attributes when used with AWS WAF.
- Access to AWS Firewall Manager, at no additional cost, for automated policy enforcement.
- Sensitive detection thresholds that route traffic into the DDoS mitigation system earlier and can improve time-to-mitigate attacks against Amazon EC2 or Network Load Balancer when used with an Elastic IP address.
- Cost protection that enables you to request a limited refund of scaling-related costs that result from a DDoS attack.
- Enhanced service level agreement that is specific to AWS Shield Advanced customers.
- Proactive engagement from the AWS SRT when a Shield event is detected.

- Protection groups that enable you to bundle resources, providing a self-service way to customize the scope of detection and mitigation for your application by treating multiple resources as a single unit. Resource grouping improves the accuracy of detection, minimizes false positives, eases automatic protection of newly created resources, and accelerates the time to mitigate attacks against many resources that comprise a single application. For information about protection groups, see Shield Advanced protection groups.

For a complete list of AWS Shield Advanced features and for more information about AWS Shield, refer to How AWS Shield works.

Best Practices for DDoS Mitigation

In the following sections, each of the recommended best practices for DDoS mitigation are described in more depth. For a quick and easy-to-implement guide on building a DDoS mitigation layer for static or dynamic web applications, see How to Help Protect Dynamic Web Applications Against DDoS Attacks.

Infrastructure Layer Defense (BP1, BP3, BP6, BP7)

In a traditional data center environment, you can mitigate infrastructure layer DDoS attacks by using techniques such as overprovisioning capacity, deploying DDoS mitigation systems, or scrubbing traffic with the help of DDoS mitigation services. On AWS, DDoS mitigation capabilities are automatically provided; but you can optimize your application's DDoS resilience by making architecture choices that best leverage those capabilities and also allow you to scale for excess traffic.

Key considerations to help mitigate volumetric DDoS attacks include ensuring that enough transit capacity and diversity are available and protecting AWS resources, like Amazon EC2 instances, against attack traffic.

Some Amazon EC2 instance types support features that can more easily handle large volumes of traffic, for example, up to 100 Gbps network bandwidth interfaces and enhanced networking. This helps prevent interface congestion for traffic that has reached the Amazon EC2 instance. Instances that support enhanced networking provide higher I/O performance, higher bandwidth, and lower CPU utilization compared to traditional implementations. This improves the ability of the instance to handle large volumes of traffic and ultimately makes them highly resilient against packets per second (pps) load.

To allow this high level of resilience, AWS recommends using Amazon EC2 Dedicated Instances or EC2 instances with higher networking throughput that have an N suffix and support for Enhanced Networking with up to 100 Gbps of Network bandwidth, for example, c6gn.16xlarge and c5n.18xlarge or metal instances (such as c5n.metal).

For more information about Amazon EC2 instances that support 100 Gigabit network interfaces and enhanced networking, see Amazon EC2 Instance Types.

The module required for enhanced networking and the required *enaSupport* attribute set are included with Amazon Linux 2 and the latest versions of the Amazon Linux AMI. Therefore, if you launch an instance with an HVM version of Amazon Linux on a supported instance type, enhanced networking is already enabled for your instance. For more information, see Test whether enhanced networking is enabled . For more information about how to enable enhanced networking, see Enhanced networking on Linux.

Amazon EC2 with Auto Scaling (BP7)

Another way to mitigate both infrastructure and application layer attacks is to operate at scale. If you have web applications, you can use load balancers to distribute traffic to a number of Amazon EC2 instances that are overprovisioned or configured to automatically scale. These instances can handle sudden traffic surges that occur for any reason, including a flash crowd or an application layer DDoS attack. You can set Amazon CloudWatch alarms to initiate Auto Scaling to automatically scale the size of your Amazon EC2 fleet in response to events that you define, such as CPU, RAM, Network I/O, and even Custom metrics. This approach protects application availability when there's an unexpected increase in request volume. When using CloudFront, Application Load Balancer, Classic Load Balancers, or Network Load Balancer with your application, TLS negotiation is handled by the distribution (CloudFront) or by the load balancer. These features help protect your instances from being impacted by TLS-based attacks by scaling to handle legitimate requests and TLS abuse attacks.

For more information about using Amazon CloudWatch to invoke Auto Scaling, see Monitoring CloudWatch metrics for your Auto Scaling groups and instances.

Amazon EC2 provides resizable compute capacity so that you can quickly scale up or down as requirements change. You can scale horizontally by automatically adding instances to your application by Scaling the size of your Auto Scaling group and you can scale vertically by using larger EC2 instance types.

Elastic Load Balancing (BP6)

Large DDoS attacks can overwhelm the capacity of a single Amazon EC2 instance. With Elastic Load Balancing (ELB), you can reduce the risk of overloading your application by distributing traffic across many backend instances. Elastic Load Balancing can scale automatically, allowing you to manage larger volumes when you have unanticipated extra traffic, for example due to flash crowds or DDoS attacks. For applications built within an Amazon VPC, there are three types of Elastic Load Balancing to consider, depending on your application type: Application Load Balancer (ALB), Classic Load Balancer (CLB) and Network Load Balancer.

For web applications, you can use the Application Load Balancer to route traffic based on content and accept only well-formed web requests. Application Load Balancer blocks many common DDoS attacks, such as SYN floods or UDP reflection attacks, protecting your application from the attack. Application Load Balancer automatically scales to absorb the additional traffic when these types of attacks are detected. Scaling activities due to infrastructure layer attacks are transparent for AWS customers and do not affect your bill.

For more information about protecting web applications with Application Load Balancer, see Getting started with Application Load Balancers.

For TCP-based applications, you can use Network Load Balancer to route traffic to targets (for example, Amazon EC2 instances) at ultra-low latency. One key consideration with Network Load Balancer is that any traffic that reaches the load balancer on a valid listener will be routed to your targets, not absorbed. You can use AWS Shield Advanced to configure DDoS protection for Elastic IP addresses. When an Elastic IP address is assigned per Availability Zone to the Network Load Balancer, AWS Shield Advanced will apply the relevant DDoS protections for the Network Load Balancer traffic.

For more information about protecting TCP applications with Network Load Balancer, see Getting started with Network Load Balancers.

Leverage AWS Edge Locations for Scale (BP1, BP3)

Access to highly scaled, diverse internet connections can significantly increase your ability to optimize latency and throughput to users, absorb DDoS attacks, and isolate faults while minimizing the impact on your application's availability. AWS edge locations provide an additional layer of network infrastructure that provides these benefits to any application that uses CloudFront, Global Accelerator, and Amazon Route 53. With these

services, you can comprehensively protect on the edge your applications running from AWS Regions.

Web Application Delivery at the Edge (BP1)

CloudFront is a service that can be used to deliver your entire website including static, dynamic, streaming, and interactive content. Persistent connections and variable time-to-live (TTL) settings can be used to offload traffic from your origin, even if you are not serving cacheable content. Use of these CloudFront features reduces the number of requests and TCP connections back to your origin, helping protect your web application from HTTP floods. CloudFront only accepts well-formed connections, which helps prevent many common DDoS attacks, such as SYN floods and UDP reflection attacks, from reaching your origin. DDoS attacks are also geographically isolated close to the source, which prevents the traffic from impacting other locations. These capabilities can greatly improve your ability to continue serving traffic to users during large DDoS attacks. You can use CloudFront to protect an origin on AWS or elsewhere on the internet.

If you're using Amazon S3 to serve static content on the internet, AWS recommends you use CloudFront to protect your bucket. You can use origin access identify (OAI) to ensure that users only access your objects by using CloudFront URLs.

For more information about OAI, see Restricting access to Amazon S3 content by using an origin access identity (OAI).

For more information about protecting and optimizing the performance of web applications with CloudFront, see Getting started with Amazon CloudFront.

Protect network traffic further from your origin using AWS Global Accelerator (BP1)

Global Accelerator is a networking service that improves availability and performance of users' traffic by up to 60%. This is accomplished by ingressing traffic at the edge location closest to your users and routing it over the AWS global network infrastructure to your application, whether it runs in a single or multiple AWS Regions.

Global Accelerator routes TCP and UDP traffic to the optimal endpoint based on performance in the closest AWS Region to the user. If there is an application failure, Global Accelerator provides failover to the next best endpoint within 30 seconds. Global Accelerator uses the vast capacity of the AWS global network and integrations with AWS Shield, such as a stateless SYN proxy capability that challenges new connection attempts and only serves legitimate end users, to protect applications.

You can implement a DDoS resilient architecture that provides many of the same benefits as the Web Application Delivery at the Edge best practices, even if your application uses protocols not supported by CloudFront or you are operating a web application that requires global static IP addresses. For example, you may require IP addresses that your end users can add to the allow list in their firewalls and are not used by any other AWS customers. In these scenarios you can use Global Accelerator to protect web applications running on Application Load Balancer and in conjunction with AWS WAF to also detect and mitigate web application layer request floods.

For more information about protecting and optimizing the performance of network traffic using Global Accelerator, see Getting started with AWS Global Accelerator.

Domain Name Resolution at the Edge (BP3)

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) service that can be used to direct traffic to your web application. It includes advanced features like Traffic Flow, Health Checks and Monitoring, Latency-Based Routing, and Geo DNS. These advanced features allow you to control how the service responds to DNS requests to improve the performance of your web application and to avoid site outages.

Amazon Route 53 uses techniques like shuffle sharding and anycast striping, that can help users access your application even if the DNS service is targeted by a DDoS attack.

With shuffle sharding, each name server in your delegation set corresponds to a unique set of edge locations and internet paths. This provides greater fault tolerance and minimizes overlap between customers. If one name server in the delegation set is unavailable, users can retry and receive a response from another name server at a different edge location.

Anycast striping allows each DNS request to be served by the most optimal location, dispersing the network load and reducing DNS latency. This provides a faster response for users. Additionally, Amazon Route 53 can detect anomalies in the source and volume of DNS queries and prioritize requests from users that are known to be reliable.

For more information about using Amazon Route 53 to route users to your application, see Getting Started with Amazon Route 53.

Application Layer Defense (BP1, BP2)

Many of the techniques discussed so far in this paper are effective at mitigating the impact that infrastructure layer DDoS attacks have on your application's availability. To also defend against application layer attacks, you need to implement an architecture that allows you to specifically detect, scale to absorb, and block malicious requests. This is an important consideration because network-based DDoS mitigation systems are generally ineffective at mitigating complex application layer attacks.

Detect and Filter Malicious Web Requests (BP1, BP2)

When your application runs on AWS, you can leverage both CloudFront and AWS WAF to help defend against application layer DDoS attacks.

CloudFront allows you to cache static content and serve it from AWS edge locations, which can help reduce the load on your origin. It can also help reduce server load by preventing non-web traffic from reaching your origin. Additionally, CloudFront can automatically close connections from slow reading or slow writing attackers (for example, Slowloris).

By using AWS WAF, you can configure web access control lists (web ACLs) on your CloudFront distributions or Application Load Balancers to filter and block requests based on request signatures. Each web ACL consists of rules that you can configure to string match or regex match one or more request attributes, such as the Uniform Resource Identifier (URI), query string, HTTP method, or header key. In addition, by using AWS WAF's rate-based rules, you can automatically block the IP addresses of bad actors when requests matching a rule exceed a threshold that you define.

Requests from offending client IP addresses will receive **403 Forbidden** error responses and will remained blocked until request rates drop below the threshold. This is useful for mitigating HTTP flood attacks that are disguised as regular web traffic. To block attacks based on IP address reputation, you can create rules using IP match conditions or use Managed Rules for AWS WAF offered by sellers in the AWS Marketplace. AWS WAF directly offers AWS Managed Rules as a managed service where you can choose IP reputation rule groups. The Amazon IP reputation list rule group contains rules that are based on Amazon internal threat intelligence. This is useful if you would like to block IP addresses typically associated with bots or other threats. The Anonymous IP list rule group contains rules to block requests from services that allow the obfuscation of viewer identity. These include requests from VPNs, proxies, Tor nodes, and cloud platforms (including AWS). Both AWS WAF and CloudFront also enable you to set geo-restrictions to block or allow requests from

selected countries. This can help block attacks originating from geographic locations where you do not expect to serve users.

To help identify malicious requests, review your web server logs or use AWS WAF's logging and Sampled Requests features. By enabling AWS WAF logging, you get detailed information about the traffic analyzed by the web ACL. AWS WAF supports log filtering, allowing you to specify which web requests are logged and which requests are discarded from the log after the inspection.

Information recorded in the logs includes the time that AWS WAF received the request from your AWS resource, detailed information about the request, and the matching action for each rule requested. Sampled Requests provide details about requests within the past three hours that matched one of your AWS WAF rules. You can use this information to identify potentially malicious traffic signatures and create a new rule to deny those requests. If you see a number of requests with a random query string, make sure to allow only the query string parameters that are relevant to cache for your application. This technique is helpful in mitigating a cache busting attack against your origin.

If you are subscribed to AWS Shield Advanced, you can engage the AWS Shield Response Team (SRT) to help you create rules to mitigate an attack that is hurting your application's availability. You can grant AWS SRT limited access to your account's AWS Shield Advanced and AWS WAF APIs. AWS SRT accesses these APIs to place mitigations on your account only with your explicit authorization. For more information, see the Support section of this document.

You can use AWS Firewall Manager to centrally configure and manage security rules, such as AWS Shield Advanced protections and AWS WAF rules, across your organization. Your AWS Organizations management account can designate an administrator account, which is authorized to create Firewall Manager policies. These policies allow you to define criteria, such as resource type and tags, which determine where rules are applied. This is useful when you have multiple accounts and want to standardize your protection.

For more information about:

- AWS Managed Rules for AWS WAF, see AWS Managed Rules for AWS WAF.
- Using geo restriction to limit access to your CloudFront distribution, see Restricting the geographic distribution of your content.
- Using AWS WAF, see

  - Getting started with AWS WAF
  - Logging web ACL traffic information
  - Viewing a sample of web requests
- Configuring rate-based rules, see Protect Web Sites & Services Using Rate-Based Rules for AWS WAF.
- How to manage the deployment of AWS WAF rules across your AWS resources with AWS Firewall Manager, see
  - Getting started with AWS Firewall Manager AWS WAF policies
  - Getting started with AWS Firewall Manager AWS Shield Advanced policies

Attack Surface Reduction

Another important consideration when architecting an AWS solution is to limit the opportunities an attacker has to target your application. This concept is known as *attack surface reduction*. Resources that are not exposed to the internet are more difficult to attack, which limits the options an attacker has to target your application's availability.

For example, if you do not expect users to directly interact with certain resources, make sure that those resources are not accessible from the internet. Similarly, do not accept traffic from users or external applications on ports or protocols that aren't necessary for communication.

In the following section, AWS provides best practices to guide you in reducing your attack surface and limiting your application's internet exposure.

Obfuscating AWS Resources (BP1, BP4, BP5)

Typically, users can quickly and easily use an application without requiring that AWS resources be fully exposed to the internet. For example, when you have Amazon EC2 instances behind an Elastic Load Balancing, the instances themselves might not need to be publicly accessible. Instead, you could provide users with access to the Elastic Load Balancing on certain TCP ports and allow only the Elastic Load Balancing to communicate with the instances. You can set this up by configuring Security Groups and network access control lists (network ACLs) within your Amazon Virtual Private Cloud (Amazon VPC). Amazon VPC allows you to provision a logically isolated section

of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

Security groups and network ACLs are similar in that they allow you to control access to AWS resources within your VPC. But security groups allow you to control inbound and outbound traffic at the instance level, while network ACLs offer similar capabilities at the VPC subnet level. There is no additional charge for using security groups or network ACLs.

Security Groups and Network Access Control Lists (Network ACLs) (BP5)

You can choose whether to specify security groups when you launch an instance or associate the instance with a security group at a later time. All internet traffic to a security group is implicitly denied unless you create an *allow* rule to permit the traffic. For example, if you have a web application that uses an Elastic Load Balancing and multiple Amazon EC2 instances, you might decide to create one security group for the Elastic Load Balancing (Elastic Load Balancing security group) and one for the instances (web application server security group). You can then create an *allow* rule to permit internet traffic to the Elastic Load Balancing security group and another rule to permit traffic from the Elastic Load Balancing security group to the web application server security group. This ensures that internet traffic can't directly communicate with your Amazon EC2 instances, which makes it more difficult for an attacker to learn about and impact your application.

When you create network ACLs, you can specify both allow and deny rules. This is useful if you want to explicitly deny certain types of traffic to your application. For example, you can define IP addresses (as CIDR ranges), protocols, and destination ports that are denied access to the entire subnet. If your application is used only for TCP traffic, you can create a rule to deny all UDP traffic, or vice versa. This option is useful when responding to DDoS attacks because it lets you create your own rules to mitigate the attack when you know the source IPs or other signature.

If you are subscribed to AWS Shield Advanced, you can register Elastic IP addresses as Protected Resources. DDoS attacks against Elastic IP addresses that have been registered as Protected Resources are detected more quickly, which can result in a faster time to mitigate. When an attack is detected, the DDoS mitigation systems read the network ACL that corresponds to the targeted Elastic IP address and enforce it at the AWS network border. This significantly reduces your risk of impact from a number of infrastructure layer DDoS attacks.

For more information about configuring Security Groups and network ACLs to optimize for DDoS resiliency, see How to Help Prepare for DDoS Attacks by Reducing Your Attack Surface.

For more information about using AWS Shield Advanced with Elastic IP addresses as Protected Resources, see the steps to Subscribe to AWS Shield Advanced.

Protecting Your Origin (BP1, BP5)

If you are using CloudFront with an origin that is inside of your VPC, you may want to ensure that only your CloudFront distribution can forward requests to your origin. With Edge-to-Origin Request Headers, you can add or override the value of existing request headers when CloudFront forwards requests to your origin. You can use the *Origin Custom Headers*, for example *X-Shared-Secret* header, to help validate that the requests made to your origin were sent from CloudFront.

For more information about protecting your origin with an *Origin Custom Headers*, see Adding custom headers to origin requests and Restricting access to Application Load Balancers.

For a guide on implementing a sample solution to automatically rotate the value of Origin Custom Headers for the origin access restriction, see How to enhance Amazon CloudFront origin security with AWS WAF and AWS Secrets Manager.

Alternatively, you can use an AWS Lambda function to automatically update your security group rules to allow only CloudFront traffic. This improves your origin's security by helping to ensure that malicious users cannot bypass CloudFront and AWS WAF when accessing your web application.

For more information about how to protect your origin by automatically updating your security groups, see How to Automatically Update Your Security Groups for Amazon CloudFront and AWS WAF by Using AWS Lambda.

Protecting API Endpoints (BP4)

Typically, when you must expose an API to the public, there is a risk that the API frontend could be targeted by a DDoS attack. To help reduce the risk, you can use Amazon API Gateway as an entryway to applications running on Amazon EC2, AWS Lambda, or elsewhere. By using Amazon API Gateway, you don't need your own servers for the API frontend and you can obfuscate other components of your application. By making it harder to detect your application's components, you can help prevent those AWS resources from being targeted by a DDoS attack.

When you use Amazon API Gateway, you can choose from two types of API endpoints. The first is the default option: edge optimized API endpoints that are accessed through a CloudFront distribution. The distribution is created and managed by API Gateway, however, so you don't have control over it. The second option is to use a regional API endpoint that is accessed from the same AWS region from which your REST API is deployed. AWS recommends that you use the second type of endpoint and associate it with your own CloudFront distribution. This gives you control over the CloudFront distribution and the ability to use AWS WAF for application layer protection. This mode provides you with access to scaled DDoS mitigation capacity across the AWS global edge network.

When using CloudFront and AWS WAF with Amazon API Gateway, configure the following options:

- Configure the cache behavior for your distributions to forward all headers to the API Gateway regional endpoint. By doing this, CloudFront will treat the content as dynamic and skip caching the content.
- Protect your API Gateway against direct access by configuring the distribution to include the origin custom header `x-api-key`, by setting the API key value in API Gateway.
- Protect the backend from excess traffic by configuring standard or burst rate limits for each method in your REST APIs.

For more information about creating APIs with Amazon API Gateway, see Amazon API Gateway Getting Started.

Operational Techniques

The mitigation techniques in this paper help you architect applications that are inherently resilient against DDoS attacks. In many cases, it's also useful to know when a DDoS attack is targeting your application so you can take mitigation steps. This section discusses best practices for gaining visibility into abnormal behavior, alerting and automation, managing protection at scale, and engaging AWS for additional support.

Visibility

When a key operational metric deviates substantially from the expected value, an attacker may be attempting to target your application's availability. Familiarity with the

normal behavior of your application, means you can take action more quickly when you detect an anomaly. Amazon CloudWatch can help by monitoring applications that you run on AWS. For example, you can collect and track metrics, collect and monitor log files, set alarms, and automatically respond to changes in your AWS resources.

If you follow the DDoS-resilient reference architecture when architecting your application, common infrastructure layer attacks will be blocked before reaching your application. If you are subscribed to AWS Shield Advanced, you have access to a number of CloudWatch metrics that can indicate that your application is being targeted. For example, you can configure alarms to notify you when there is a DDoS attack in progress, so you can check your application's health and decide whether to engage AWS SRT. You can configure the `DDoSDetected` metric to tell you if an attack has been detected. If you want to be alerted based on the attack volume, you can also use the `DDoSAttackBitsPerSecond`, `DDoSAttackPacketsPerSecond`, or `DDoSAttackRequestsPerSecond` metrics. You can monitor these metrics by integrating Amazon CloudWatch with your own tools or by using tools provided by third parties, such as Slack or PagerDuty.

An application layer attack can elevate many Amazon CloudWatch metrics. If you're using AWS WAF, you can use CloudWatch to monitor and activate alarms on increases in requests that you've set in AWS WAF to be allowed, counted, or blocked. This allows you to receive a notification if the level of traffic exceeds what your application can handle. You can also use CloudFront, Amazon Route 53, Application Load Balancer, Network Load Balancer, Amazon EC2, and Auto Scaling metrics that are tracked in CloudWatch to detect changes that can indicate a DDoS attack.

The *Recommended Amazon CloudWatch Metrics* table lists descriptions of Amazon CloudWatch metrics that are commonly used to detect and react to DDoS attacks.

Recommended Amazon CloudWatch Metrics

Topic	Metric	Description
AWS Shield Advanced	DDoSDetected	Indicates a DDoS event for a specific Amazon Resource Name (ARN).
AWS Shield Advanced	DDoSAttackBitsPerSecond	The number of bytes observed during a DDoS event for a specific ARN. This metric is only available for layer 3/4 DDoS events.

Topic	Metric	Description
AWS Shield Advanced	DDoSAttackPacketsPerSecond	The number of packets observed during a DDoS event for a specific ARN. This metric is only available for layer 3/4 DDoS events.
AWS Shield Advanced	DDoSAttackRequestsPerSecond	The number of requests observed during a DDoS event for a specific ARN. This metric is only available for layer 7 DDoS events and is only reported for the most significant layer 7 events.
AWS WAF	AllowedRequests	The number of allowed web requests.
AWS WAF	BlockedRequests	The number of blocked web requests.
AWS WAF	CountedRequests	The number of counted web requests.
AWS WAF	PassedRequests	The number of passed requests. This is only used for requests that go through a rule group evaluation without matching any of the rule group rules.
CloudFront	Requests	The number of HTTP/S requests.
CloudFront	TotalErrorRate	The percentage of all requests for which the HTTP status code is 4xx or 5xx.
Amazon Route 53	HealthCheckStatus	The status of the health check endpoint.
ALB	ActiveConnectionCount	The total number of concurrent TCP connections that are active from clients to the load balancer, and from the load balancer to targets.
ALB	ConsumedLCUs	The number of load balancer capacity units (LCU) used by your load balancer.
ALB	HTTPCode_ELB_4XX_Count HTTPCode_ELB_5XX_Count	The number of HTTP 4xx or 5xx client error codes generated by the load balancer.

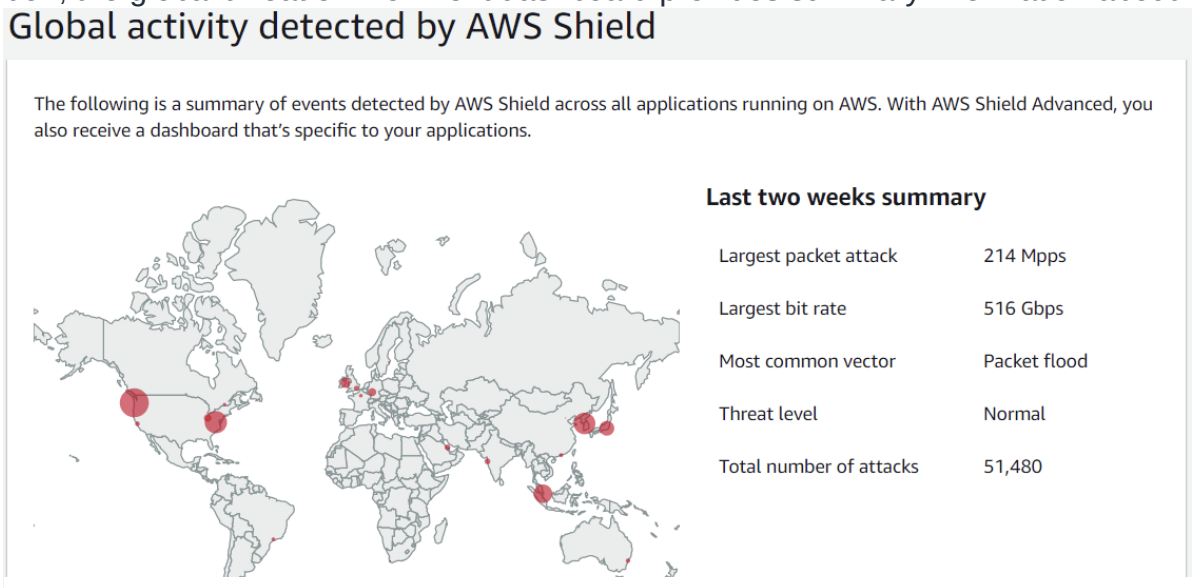
Topic	Metric	Description
ALB	NewConnectionCount	The total number of new TCP connections established from clients to the load balancer, and from the load balancer to targets.
ALB	ProcessedBytes	The total number of bytes processed by the load balancer.
ALB	RejectedConnectionCount	The number of connections rejected because the load balancer reached its maximum number of connections.
ALB	RequestCount	The number of requests that were processed.
ALB	TargetConnectionErrorCount	The number of connections that were not successfully established between the load balancer and the target.
ALB	TargetResponseTime	The time elapsed, in seconds, after the request leaves the load balancer until a response from the target is received.
ALB	UnHealthyHostCount	The number of targets that are considered unhealthy.
NLB	ActiveFlowCount	The total number of concurrent TCP flows (or connections) from clients to targets.
NLB	ConsumedLCUs	The number of load balancer capacity units (LCU) used by your load balancer.
NLB	NewFlowCount	The total number of new TCP flows (or connections) established from clients to targets in the time period.
NLB	ProcessedBytes	The total number of bytes processed by the load balancer, including TCP/IP headers.
Global Accelerator	NewFlowCount	The total number of new TCP and UDP flows (or connections) established from clients to endpoints in the time period.

Topic	Metric	Description
Global Accelerator	ProcessedBytesIn	The total number of incoming bytes processed by the accelerator, including TCP/IP headers.
Auto Scaling	GroupMaxSize	The maximum size of the Auto Scaling group.
Amazon EC2	CPUUtilization	The percentage of allocated EC2 compute units that are currently in use.
Amazon EC2	NetworkIn	The number of bytes received by the instance on all network interfaces.

For more information about using Amazon CloudWatch to detect DDoS attacks on your application, see Getting Started with Amazon CloudWatch.

To explore an example of a dashboard built using some of the metrics from the preceding table, see A custom baseline monitoring system.

AWS includes several additional metrics and alarms to notify you about an attack and to help you monitor your application's resources. The AWS Shield console or API provide a per-account event summary and details about attacks that have been detected. In addition, the global threat environment dashboard provides summary information about



Global threat environment dashboard

all DDoS attacks that have been detected by AWS. This information may be useful to better understand DDoS threats across a larger population of applications in addition to attack trends, and comparing with attacks that you may have observed.

If you are subscribed to AWS Shield Advanced, the service dashboard displays additional detection and mitigation metrics and network traffic details for events detected on protected resources. AWS Shield evaluates traffic to your protected resource along multiple dimensions. When an anomaly is detected, AWS Shield creates an event and reports the traffic dimension where the anomaly was observed. With a placed mitigation this protects your resource from receiving excess traffic and traffic that matches a known DDoS event signature.

Detection metrics are based on sampled network flows or AWS WAF logs when a web ACL is associated with the protected resource. Mitigation metrics are based on traffic that's observed by Shield's DDoS mitigation systems. Mitigation metrics are a more precise measurement of the traffic into your resource.

The network top contributors metric provides insight into where traffic is coming from during a detected event. You can view the highest volume contributors and sort by aspects such as protocol, source port, and TCP flags. The top contributors metric includes metrics for all traffic observed on the resource along various dimensions. It provides additional metric dimensions you can use to understand network traffic that's sent to your resource during an event.

The service dashboard also includes details about the actions automatically taken to mitigate DDoS attacks. This information makes it easier to investigate anomalies, explore dimensions of the traffic, and better understand the actions taken by AWS Shield Advanced to protect your availability.

Another tool that can help you gain visibility into traffic that is targeting your application is VPC Flow Logs. On a traditional network, you might use network flow logs to troubleshoot connectivity and security issues and to make sure that network access rules are working as expected. By using VPC Flow Logs, you can capture information about the IP traffic that is going to and from network interfaces in your VPC.

Each flow log record includes the following: source and destination IP addresses, source and destination ports, protocol, and the number of packets and bytes transferred during the capture window. You can use this information to help identify anomalies in network traffic and to identify a specific attack vector. For example, most UDP reflection attacks have specific source ports, such as source port 53 for DNS reflection. This is a clear attack signature that you can identify in the flow log record. In response, you might

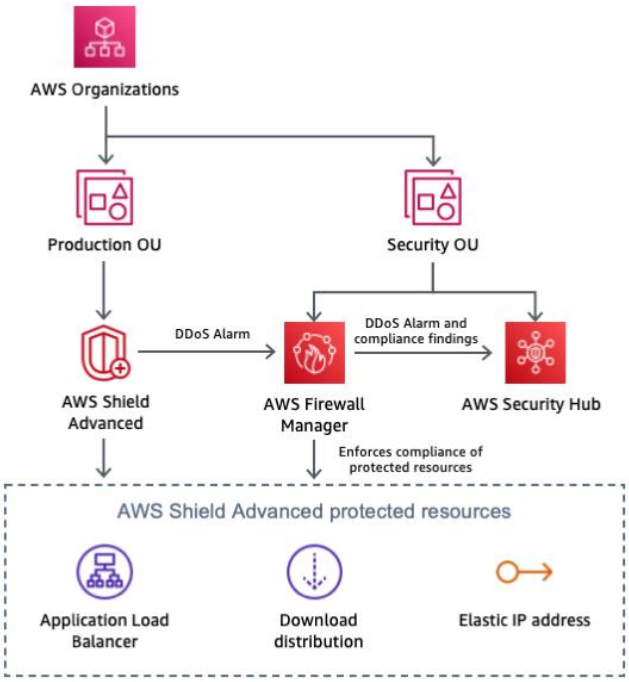
choose to block the specific source port at the instance level or create a network ACL rule to block the entire protocol if your application doesn't require it.

For more information about using VPC Flow Logs to identify network anomalies and DDoS attack vectors, see VPC Flow Logs and VPC Flow Logs – Log and View Network Traffic Flows.

Visibility and protection management across multiple accounts

In scenarios when you operate across multiple AWS accounts and have multiple components to protect, using techniques that enable you to operate at scale and reduce operational overhead increase your mitigation capabilities. When managing AWS Shield Advanced protected resources in multiple accounts, you can set up centralized monitoring by using AWS Firewall Manager and AWS Security Hub. With Firewall Manager, you can create a security policy that enforces DDoS protection compliance across all your accounts. You can use these two services together to manage your protected resources across multiple accounts and centralize the monitoring of those resources.

Security Hub automatically integrates with Firewall Manager, allowing AWS Shield Advanced customers to view security findings in a single dashboard, alongside other high priority security alerts and compliance statuses. For instance, when AWS Shield Advanced detects anomalous traffic destined for a protected resource in any AWS account within the scope, this finding will be visible in the Security Hub console. If configured, Firewall Manager can automatically bring the resource into compliance by creating it as an AWS Shield Advanced–protected resource, and then update Security Hub when the resource is in a compliant state.



Monitoring AWS Shield protected resources with Firewall Manager and Security Hub architecture diagram

For more information about central monitoring of AWS Shield protected resources, see [Set up centralized monitoring for DDoS events and auto-remediate noncompliant resources.]

Support

If you experience an attack, you can also benefit from support from AWS in assessing the threat and reviewing the architecture of your application, or you might want to request other assistance. It is important to create a response plan for DDoS attacks before an actual event. The best practices outlined in this paper are intended to be proactive measures that you implement before you launch an application, but DDoS attacks against your application might still occur. Review the options in this section to determine the support resources that are best suited for your scenario. Your account

team can evaluate your use case and application, and assist with specific questions or challenges that you have.

If you're running production workloads on AWS, consider subscribing to Business Support, which provides you with 24/7 access to Cloud Support Engineers who can assist with DDoS attack issues. If you're running mission critical workloads, consider Enterprise Support which provides the ability to open critical cases and receive the fastest response from a Senior Cloud Support Engineer.

If you are subscribed to AWS Shield Advanced and are also subscribed to either Business Support or Enterprise Support, you can configure AWS Shield proactive engagement. It allows you to configure health checks, associate to your resources, and provide 24/7 operations contact information. When AWS Shield detects signs of DDoS and your application health checks are showing signs of degradation, AWS SRT will proactively reach out to you. This is our recommended engagement model because it allows for the quickest AWS SRT response times and empowers AWS SRT to begin troubleshooting even before contact has been established with you.

The proactive engagement feature requires you to configure an Amazon Route 53 health check that accurately measures the health of your application and is associated with the resource protected by AWS Shield Advanced. Once a Route 53 health check is associated in the AWS Shield console, the AWS Shield Advanced detection system uses the health check status as an indicator of your application's health. AWS Shield Advanced's health-based detection feature will ensure that you are notified and that mitigations are placed more quickly when your application is unhealthy. AWS SRT will contact you to troubleshoot whether the unhealthy application is being targeted by a DDoS attack and place additional mitigations as needed.

Completing configuration of proactive engagement includes adding contact details in the AWS Shield console. AWS SRT will use this information to contact you. You can configure up to 10 contacts and provide additional notes if you have any specific contact requirements or preferences. Proactive engagement contacts should hold a 24/7 role, such as a security operations center or an individual who is immediately available.

You can enable proactive engagement for all resources or for select key production resources where response time is critical. This is accomplished by assigning health checks only to these resources.

You can also escalate to AWS SRT by creating an AWS Support case using the AWS Support console or Support API if you have a DDoS-related event that affects your application's availability.

Conclusion

The best practices outlined in this paper can help you build a DDoS resilient architecture that protects your application's availability by preventing many common infrastructure and application layer DDoS attacks. The extent to which you follow these best practices when you architect your application will influence the type, vector, and volume of DDoS attacks that you can mitigate. You can incorporate resiliency without subscribing to a DDoS mitigation service. By choosing to subscribe to AWS Shield Advanced you gain additional support, visibility, mitigation, and cost protection features that further protect an already resilient application architecture.

Contributors

The following individuals and organizations contributed to this document:

- Jeffrey Lyon, AWS Perimeter Protection
- Rodrigo Ferroni, AWS Security Specialist TAM
- Dmitriy Novikov, AWS Solutions Architect
- Achraf Souk, AWS Solutions Architect
- Yoshihisa Nakatani, AWS Solutions Architect

Further Reading

For additional information, see:

- Best Practices for DDoS Mitigation on AWS
- Guidelines for Implementing AWS WAF
- SID324 – re:Invent 2017: Automating DDoS Response in the Cloud
- CTD304 – re:Invent 2017: Dow Jones & Wall Street Journal's Journey to Manage Traffic Spikes While Mitigating DDoS & Application Layer Threats
- CTD310 – re:Invent 2017: Living on the Edge, It's Safer Than You Think! Building Strong with Amazon CloudFront, AWS Shield, and AWS WAF

- SEC407 - re:Invent 2019: A defense-in-depth approach to building web applications
- SEC321 - re:Invent 2020: Get ahead of the curve with DDoS Response Team escalations
- William Hill: High-performance DDOS Protection with AWS

Document revisions

Date	Description
September 21, 2021	Updated to include latest recommendations and features. AWS Global Accelerator is added as part of comprehensive protection at the edge. AWS Firewall Manager for centralized monitoring for DDoS events and auto-remediate noncompliant resources.
December 2019	Updated to clarify cache busting in Detect and Filter Malicious Web Requests (BP1, BP2) section, and Elastic Load Balancing and Application Load Balancer usage in Scale to Absorb (BP6) section. Updated diagrams and Table 2, marked “Choice of Region.” as BP8. Updated BP7 section with more details.
December 2018	Updated to include AWS WAF logging as a best practice.
June 2018	Updated to include AWS Shield, AWS WAF features, AWS Firewall Manager, and related best practices.
June 2016	Added prescriptive architecture guidance and updated to include AWS WAF.
June 2015	Whitepaper published.