Implementation Guide

Security Automations for AWS WAF

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Security Automations for AWS WAF: Implementation Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Solution overview **1**
Features and benefits 3
Secure your web applications with AWS Managed Rules rule groups 3
Provide layer 7 flood protection with predefined HTTP Flood custom rule 4
Block exploitation of vulnerabilities with predefined Scanners & Probes custom rule 4
Detect and deflect intrusion with predefined Bad Bot custom rule 4
Block malicious IP addresses with predefined IP reputations lists custom rule 5
Provide manual IP configuration with predefined allowed and denied IP lists custom rule 5
Build your own monitoring dashboard 5
Use cases 5
Concepts and definitions 6
Architecture overview **9**
Architecture diagram 9
AWS Well-Architected design considerations 12
Operational excellence 12
Security 13
Reliability 13
Performance efficiency 13
Cost optimization 14
Sustainability 14
Architecture details **15**
AWS services in this solution 15
Log parser options 16
AWS WAF rate-based rule 16
Amazon Athena log parser 16
AWS Lambda log parser 17
Component details 17
Log parser - Application 17
Log parser - AWS WAF 19
Log parser - Bad bot 21
IP lists parser 22
Plan your deployment **23**
Supported AWS Regions 23

Cost ... 24
Cost estimate of CloudWatch logs ... 27
Cost estimate of Athena ... 27
Security ... 28
IAM roles ... 28
Data ... 28
Protection capabilities ... 28
Quotas ... 29
Quotas for AWS services in this solution ... 29
AWS WAF quotas ... 30
Deployment considerations ... 30
AWS WAF rules ... 30
Web ACL traffic logging ... 30
Oversize handling for request components ... 31
Multiple solution deployments ... 31
Minimum role permissions for deployment (Optional) ... 31
Deploy the solution ... 39
Deployment process overview ... 39
AWS CloudFormation templates ... 40
Main stack ... 40
WebACL stack ... 40
Firehose Athena stack ... 40
Prerequisites ... 41
Configure a CloudFront distribution ... 41
Configure an ALB ... 41
Step 1. Launch the stack ... 42
Step 2. Associate the web ACL with your web application ... 75
Step 3. Configure web access logging ... 75
Store web access logs from a CloudFront distribution ... 75
Store web access logs from an Application Load Balancer ... 76
Update the solution ... 77
Update considerations ... 78
Resource type update ... 78
WAFV2 upgrade ... 78
Customizations at stack update ... 78
Bad bot Protection upgrade ... 78

CDK upgrade ... 79
Uninstall the solution ... 80
Use the solution ... 81
Modify the allowed and denied IP sets (optional) ... 81
Embed the Honeypot link in your web application (optional) ... 81
Create a CloudFront Origin for the Honeypot Endpoint ... 82
Embed the Honeypot endpoint as an external link ... 83
Use Lambda log parser JSON file ... 84
Use Lambda log parser JSON file for HTTP Flood protection ... 84
Use Lambda log parser JSON file for scanner and probe protection ... 85
Use country and URI in HTTP flood Athena log parser ... 87
View Amazon Athena queries ... 87
View WAF log queries ... 88
View application access log queries ... 89
View adding Athena partition queries ... 89
Configure IP retention on Allowed and Denied AWS WAF IP sets ... 90
How it works ... 90
Turn on IP retention ... 91
Build monitoring dashboard ... 92
Handle XSS false positives ... 93
Troubleshooting ... 95
Contact AWS Support ... 95
Create case ... 95
How can we help? ... 95
Additional information ... 95
Help us resolve your case faster ... 96
Solve now or contact us ... 96
Developer guide ... 97
Source code ... 97
Reference ... 98
Anonymized data collection ... 98
Related resources ... 99
Associated AWS whitepapers ... 99
Associated AWS Security Blog posts ... 99
Third-Party IP Reputation Lists ... 99
Contributors ... 100

Revisions **101**
Notices **102**

Automatically deploy a single web access control list that filters web-based attacks with Security Automations on AWS WAF

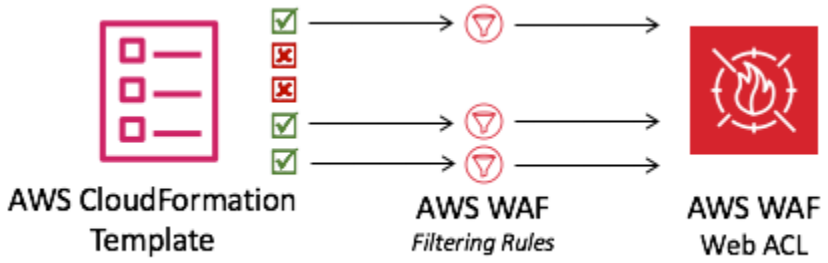
Important

Security Automations for AWS WAF will retire in December 2026. Deployments (via CloudFormation or GitHub) will remain operational, but customers will assume responsibility for maintenance and API-related updates post-retirement. Customers can explore using native AWS WAF capabilities including AWS Managed Rules, native rate-based rules, and the AWS WAF Console for intuitive configuration and management. These native capabilities provide superior functionality, better performance, real-time updates, and enhanced security features compared to the solution.
Customers running solution versions prior to v3.0.0 (which deployed AWS WAF Classic) should first migrate to AWS WAF v2 using the AWS WAF migration documentation. Customers on v3.0.0 or later are already on AWS WAF v2 and can directly adopt native managed rules and console-based management.

The Security Automations for AWS WAF solution deploys a set of preconfigured rules to help you protect your applications from common web exploits. This solution's core service, AWS WAF, helps protect web applications from attack techniques that can affect application availability, compromise security, or consume excessive resources. You can use AWS WAF to define customizable web security rules. These rules control which traffic to allow or block to web applications and application programming interfaces (APIs) deployed on AWS resources such as Amazon CloudFront, Application Load Balancer (ALB). For more supported resource types, see AWS WAF in the *AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide*.

Configuring AWS WAF rules can be challenging and burdensome to large and small organizations alike, especially for those who don't have dedicated security teams. To simplify this process, the Security Automations for AWS WAF solution automatically deploys a single web access control list (ACL) with a set of AWS WAF rules designed to filter common web-based attacks. During initial configuration of this solution's AWS CloudFormation template, you can specify which protective features to include. After you deploy this solution, AWS WAF inspects web requests to their existing CloudFront distribution(s) or ALB(s), and blocks them when applicable.

A CloudFormation template deploys a web ACL with AWS WAF filtering rules.



This implementation guide discusses architectural considerations, configuration steps, and operational best practices for deploying this solution in the Amazon Web Services (AWS) Cloud. It includes links to CloudFormation templates that launch, configure, and run the AWS security, compute, storage, and other services required to deploy this solution on AWS, using AWS best practices for security and availability.

The information in this guide assumes working knowledge of AWS services such as AWS WAF, CloudFront, ALBs, and AWS Lambda. It also requires basic knowledge of common web-based attacks and mitigation strategies.

> **Note**
>
> As of version 3.0.0, this solution supports the latest version of the AWS WAF service API (AWS WAFV2).

This guide is intended for IT managers, security engineers, DevOps engineers, developers, solutions architects, and website administrators.

> **Note**
>
> We recommend using this solution as a starting point for implementing AWS WAF rules. You can customize the source code, add new custom rules, and leverage more AWS WAF managed rules based on your needs.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution. The total cost for running this solution depends on the protection activated and the amount of data ingested, stored, and processed.	Cost
Understand the security considerations for this solution.	Security
Know which AWS Regions are supported for this solution.	Supported AWS Regions
View or download the CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template
Use Support to help you deploy, use, or troubleshoot the solution.	Support
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution	GitHub repository

Features and benefits

The Security Automations for AWS WAF solution provides the following features and benefits.

Secure your web applications with AWS Managed Rules rule groups

AWS Managed Rules for AWS WAF provides protection against common application vulnerabilities or other unwanted traffic. This solution includes AWS Managed IP reputation rule groups, AWS Managed baseline rule groups and AWS Managed use-case specific rule groups. You have the option of selecting one or more rules groups for your web ACL, up to the maximum web ACL capacity unit (WCU) quota.

Provide layer 7 flood protection with predefined HTTP Flood custom rule

The **HTTP Flood** custom rule protects against a web-layer Distributed Denial-of-Service (DDoS) attack for a customer-defined period of time. You can choose one of these options to activate this rule:

- AWS WAF rate-based rule
- Lambda log parser
- Amazon Athena log parser

The Lambda log parser or Athena log parser options allow you to define a request quota of less than 100. This approach can help you not reach the quota required by AWS WAF rate-based rules. For more information, see Log parser options.

You can also enhance the Athena log parser by adding a country and Uniform Resource Identifier (URI) to filtering conditions. This approach identifies and blocks HTTP flood attacks that have unpredictable URI patterns. For more information, refer to Use country and URI in HTTP Flood Athena log parser.

Block exploitation of vulnerabilities with predefined Scanners & Probes custom rule

The **Scanners & Probes** custom rule parses application access logs searching for suspicious behavior, such as an abnormal amount of errors generated by an origin. It then blocks those suspicious source IP addresses for a customer-defined period of time. You can choose one of these options to activate this rule: Lambda log parser or Athena log parser. For more information, see Log parser options.

Detect and deflect intrusion with predefined Bad Bot custom rule

The **Bad Bot** custom rule sets up a honeypot endpoint, which is a security mechanism intended to lure and deflect an attempted attack. You can insert the endpoint in your website to detect inbound requests from content scrapers and bad bots. Once detected, any subsequent requests from the same origins will be blocked. For more information, see Embed the Honeypot link in your web application.

Block malicious IP addresses with predefined IP reputations lists custom rule

The **IP reputation lists** custom rule checks third-party IP reputation lists hourly for new IP ranges to block. These lists include the Spamhaus Don't Route Or Peer (DROP) and Extended DROP (EDROP) lists, the Proofpoint Emerging Threats IP list, and the Tor exit node list.

Provide manual IP configuration with predefined allowed and denied IP lists custom rule

The **allowed and denied IP lists** custom rules allow you to manually insert IP addresses that you want to allow or deny. You can also configure IP retention on Allowed and Denied IP lists to expire IPs at a set time.

Build your own monitoring dashboard

This solution emits Amazon CloudWatch metrics such as allowed requests, blocked requests, and other relevant metrics. You can build a customized dashboard to visualize these metrics and gain insights into the pattern of attacks and protection provided by AWS WAF. For more information, refer to Build monitoring dashboard.

Use cases

The following are example use cases for using this solution. You can customize this solution in innovative ways that aren't limited to this list.

Automate the setup of AWS WAF rules

AWS WAF protects your web application from common attacks; however, setting up AWS WAF rules can be complicated and time consuming. To help you, this solution automatically deploys a set of AWS WAF rules into your account with a CloudFormation template. This way, you don't need to configure AWS WAF rules yourself, and you can get started with AWS WAF faster.

Customize layer 7 HTTP Flood protection

This solution provides three options to activate HTTP Flood protection. You can select the option that fits your needs to gain protection against DDoS attacks. For more information, see **Provide layer 7 flood protection with pre-defined HTTP Flood custom rule** in Features and benefits.

Leverage the source code for applying customization or building your own security automations

This solution provides an example for how to use AWS WAF and other services to build security automations on the AWS Cloud. Its open source code in GitHub makes it convenient for you to apply customizations or build your own security automations that fit your needs.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution.

ALB logs

This solution uses logs for the ALB resource. The **Scanner & Probe Protection** rule in this solution inspect these logs.

Athena log parser

Amazon Athena is a serverless, interactive analytics service that is built on open-source frameworks, supporting open-table and file formats. This solution runs a scheduled Athena query to inspect AWS WAF, CloudFront, or ALB logs if user chooses `yes - Amazon Athena log parser` when activating the **HTTP Flood Protection** rule or **Scanner & Probe Protection** rule, and can be used for **Activate Bad Bot Protection** through detection that operates through a structured logic chain.

AWS WAF rule

An AWS WAF rule defines:

- How to inspect HTTP(S) web requests
- The action to take on a request when it matches the inspection criteria

You define rules only in the context of a rule group or web ACL.

CloudFront logs

This solution uses logs for the CloudFront resource. The **Scanner & Probe Protection** rule in this solution inspects these logs.

IP set

An IP set provides a collection of IP addresses and IP address ranges that you want to use

together in a rule statement. IP sets are AWS resources.

Lambda log parser

This solution runs a Lambda function invoked by an Amazon Simple Storage Service (Amazon S3) object create event. The Lambda function initiates an inspection of AWS WAF, CloudFront, or ALB logs if the user chooses `yes - AWS Lambda log parser` when activating the **HTTP Flood Protection** , **Scanner & Probe Protection** and can be used for **Bad Bot Protection** rule through detection that operates through a structured logic chain.

Managed rule groups

Managed rule groups are collections of predefined, ready-to-use rules that AWS and AWS Marketplace sellers write and maintain for you. AWS WAF Pricing applies to your use of any managed rule group.

resource/endpoint type

You can associate AWS resources with web ACLs to protect them. These resources are CloudFront, ALB, AWS AppSync, Amazon Cognito, AWS App Runner, and AWS Verified Access resources. Currently this solution Amazon supports CloudFront and ALB.

WAF logs

This solution uses logs generated by AWS WAF for the resources associated with the web ACL. The **HTTP Flood Protection**, **Scanner & Probe Protection** and **Activate Bad Bot Protection** rules for this solution inspect these logs.

WCU

AWS WAF uses web access control list (ACL) capacity units (WCUs) to calculate and control the operating resources that are required to run your rules, rule groups, and web ACLs. AWS WAF enforces WCU quotas when you configure your rule groups and web ACLs. WCUs don’t affect how AWS WAF inspects web traffic.

web ACL

A web ACL gives you fine-grained control over the HTTP(S) web requests that your protected resource responds to.

> **Note**
>
> For a general reference of AWS terms, see the AWS Glossary.

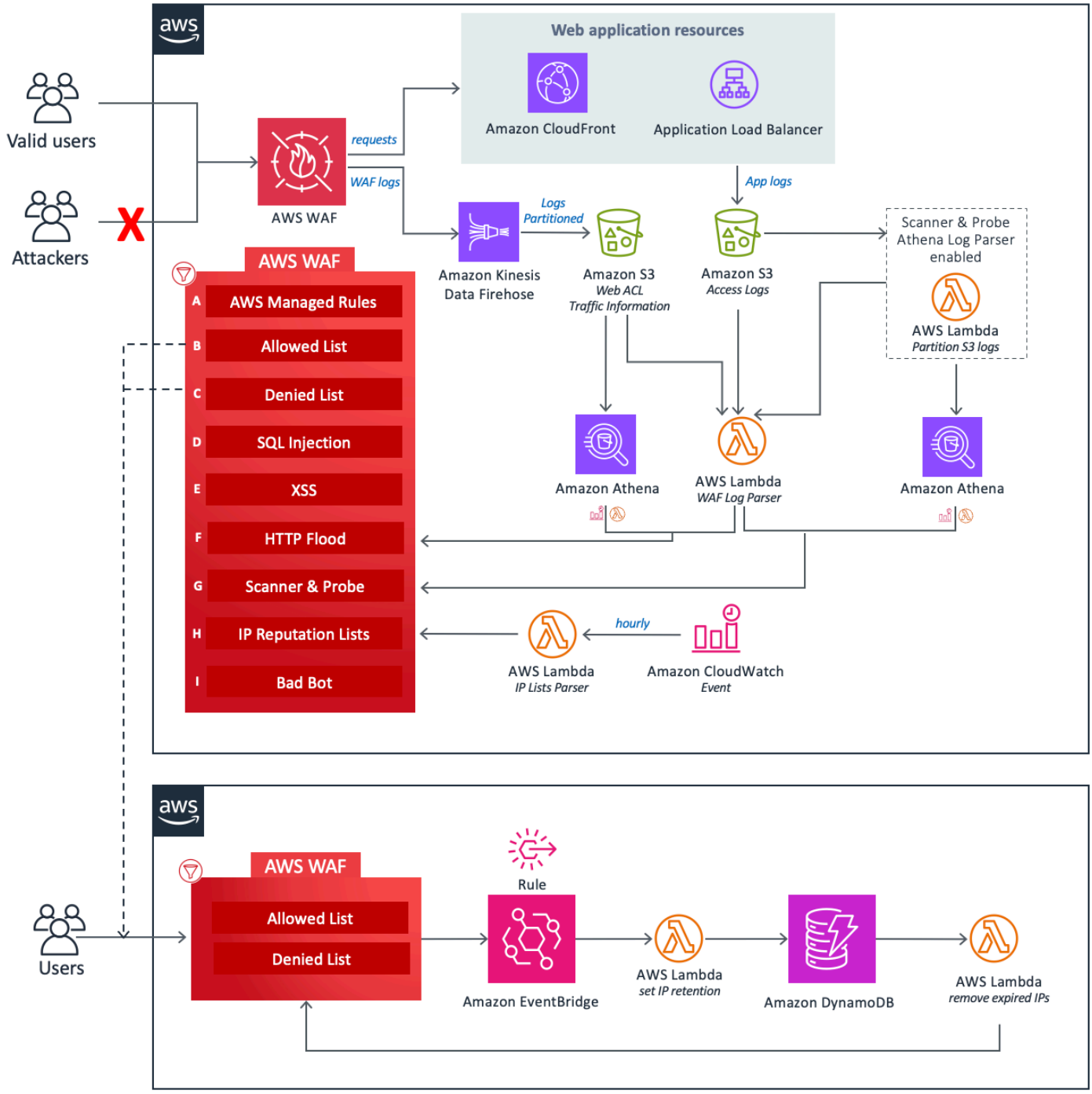
Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.

CloudFormation template deploys AWS WAF and other AWS resources to protect your web application from common attacks.



At the core of the design is an AWS WAF web ACL, which acts as the central inspection and decision point for all incoming requests to a web application. During initial configuration of the CloudFormation stack, the user defines which protective components to activate. Each component operates independently and adds different rules to the web ACL.

The components of this solution can be grouped into the following areas of protection.

> **Note**
>
> The group labels don't reflect the priority level of the WAF rules.

- **AWS Managed Rules (A)** - This component contains AWS Managed Rules IP reputation rule groups, baseline rule groups, and use-case specific rule groups. These rule groups protect against exploitation of common application vulnerabilities or other unwanted traffic, including those described in OWASP publications, without having to write your own rules.
- **Manual IP lists (B and C)** - These components create two AWS WAF rules. With these rules, you can manually insert IP addresses that you want to allow or deny. You can configure IP retention and remove expired IP addresses on allowed or denied IP sets using Amazon EventBridge rules and Amazon DynamoDB. For more information, refer to Configure IP retention on Allowed and Denied AWS WAF IP sets.
- **SQL Injection (D) and XSS (E)** - These components configure two AWS WAF rules that are designed to protect against common SQL injection or cross-site scripting (XSS) patterns in the URI, query string, or body of a request.
- **HTTP Flood (F)** - This component protects against attacks that consist of a large number of requests from a particular IP address, such as a web-layer DDoS attack or a brute-force login attempt. With this rule, you set a quota that defines the maximum number of incoming requests allowed from a single IP address within a default five-minute period (configurable with the **Athena Query Run Time Schedule** parameter). After this threshold is breached, additional requests from the IP address are temporarily blocked. You can implement this rule by using an AWS WAF rate-based rule, or by processing AWS WAF logs using a Lambda function or Athena query. For more information about the tradeoffs related to HTTP flood mitigation options, refer to Log parser options.
- **Scanner and Probe (G)** - This component parses application access logs searching for suspicious behavior, such as an abnormal amount of errors generated by an origin. Then it blocks those suspicious source IP addresses for a customer-defined period of time. You can implement this rule using a Lambda function or Athena query. For more information about the tradeoffs related to scanner and probe mitigation options, refer to Log parser options.
- **IP Reputation Lists (H)** - This component is the `IP Lists Parser` Lambda function that checks third-party IP reputation lists hourly for new ranges to block. These lists include the Spamhaus Don't Route Or Peer (DROP) and Extended DROP (EDROP) lists, the Proofpoint Emerging Threats IP list, and the Tor exit node list.

- **Bad Bot (I)** - This component enhances bad bot detection by monitoring direct connections to an Application Load Balancer (ALB) or Amazon CloudFront, in addition to the honeypot mechanism. If a bot bypasses the honeypot and attempts to interact with ALB or CloudFront, the system analyzes request patterns and logs to identify malicious activity. When a bad bot is detected, its IP address is extracted and added to an AWS WAF block list to prevent further access. Bad bot detection operates through a structured logic chain, ensuring comprehensive threat coverage:
  - HTTP Flood Protection Lambda Log Parser – Collects bad bot IPs from log entries during flood analysis.
  - Scanner & Probe Protection Lambda Log Parser – Identifies bad bot IPs from scanner-related log entries.
  - HTTP Flood Protection Athena Log Parser – Extracts bad bot IPs from Athena logs, using partitions across query run.
  - Scanner & Probe Protection Athena Log Parser – Retrieves bad bot IPs from scanner-related Athena logs, using the same partitioning strategy.
  - Fallback Detection – If both HTTP Flood Protection and Scanner & Probe Protection are disabled, the system relies on the Log Lambda parser, which logs bot activity based on WAF label filters.

Each of the three custom Lambda functions in this solution publish runtime metrics to CloudWatch. For more information on these Lambda functions, refer to Component details.

AWS Well-Architected design considerations

This solution uses the best practices from the AWS Well-Architected Framework, which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

- The solution pushes metrics to CloudWatch to provide observability into the infrastructure, Lambda functions, Amazon Data Firehose, Amazon S3 buckets, and the rest of the solution components.

- We develop, test, and publish the solution through an AWS continuous integration and continuous delivery (CI/CD) pipeline. This helps developers achieve high quality results consistently.
- You can install the solution with a CloudFormation template that provisions all the required resources in your account. To update or delete the solution, you only need to update or delete the template.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- All inter-service communications use AWS Identity and Access Management (IAM) roles.
- All roles used by the solution follow least-privilege access. In other words, they only contain minimum permissions required so that the service can function properly.
- All data storage, including Amazon S3 buckets and DynamoDB, have encryption at rest.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

- The solution uses AWS serverless services wherever possible (for example, Lambda, Firehose, Amazon S3, and Athena) to ensure high availability and recovery from service failure.
- We perform automated tests on the solution to detect and fix errors quickly.
- The solution uses Lambda functions for data processing. The solution stores data in Amazon S3 and DynamoDB, and it persists in multiple Avaialbility Zones by default.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- The solution uses a serverless architecture to ensure high scalability and availability at a reduced cost.

- The solution enhances database performance by parititioning data and optimizing query to reduce the amount of data scanning and achieve faster results.
- The solution is automatically tested and deployed every day. Our solution architects and subject matter experts review the solution for areas to experiment and improve.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- The solution uses a serverless architecture, and customers pay only for what they use.
- The solution's compute layer defaults to Lambda, which uses a pay-per-use model.
- The Athena database and queries are optimized to reduce the amount of data scanning, thereby reducing cost.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

- The solution uses managed and serverless services to minimize the environmental impact of the backend services.
- The solution's serverless design is aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

AWS services in this solution

AWS service	Description
AWS WAF	**Core**. Deploys an AWS WAF web ACL, AWS Managed Rules rule groups, custom rules, and IP sets. Makes AWS WAF API calls to block common attacks and secure web applications.
Amazon Data Firehose	**Core**. Delivers AWS WAF logs to Amazon S3 buckets.
Amazon S3	**Core**. Stores AWS WAF, CloudFront, and ALB logs.
AWS Lambda	**Core**. Deploys multiple Lambda functions to support custom rules.
Amazon EventBridge	**Core**. Creates events rules to invoke Lambda.
Amazon Athena	**Supporting**. Creates Athena queries and work groups to support the Athena log parser.
AWS Glue	**Supporting**. Creates databases and tables to support the Athena log parser.
Amazon SNS	**Supporting**. Sends Amazon Simple Notificat ion Service (Amazon SNS) email notifications to support IP retention on allowed and denied lists.

AWS service	Description
AWS Systems Manager	**Supporting**. Provides application-level resource monitoring and visualization of resource operations and cost data.

Log parser options

As described in the Architecture overview, there are three options to handle HTTP flood and scanner and probe protections. The following sections explain each of these options in more detail.

AWS WAF rate-based rule

Rate-based rules are available for HTTP flood protection. By default, a rate-based rule aggregates and rate limits requests based on the request IP address. This solution allows you to specify the number of web requests that a client IP allows in a trailing, continuously updated five-minute period. If an IP address breaches the configured quota, AWS WAF blocks new requests blocked until the request rate is less than the configured quota.

We recommend selecting the rate-based rule option if the request quota is more than 2,000 requests per five minutes and you don't need to implement customizations. For example, you don't consider static resource access when counting requests.

You can further configure the rule to use various other aggregation keys and key combinations. For more information, see Aggregation options and keys.

Amazon Athena log parser

Both **HTTP Flood Protection** and **Scanner & Probe Protection** template parameters provide the Athena log parser option. When activated, CloudFormation provisions an Athena query and a scheduled Lambda function responsible for orchestrating Athena to run, process result output, and update AWS WAF. This Lambda function is invoked by a CloudWatch event configured to run every five minutes. This is configurable with the **Athena Query Run Time Schedule** parameter.

We recommend selecting this option when you can't use AWS WAF rate-based rules and you have familiarity with SQL to implement customizations. For more information about how to change the default query, refer to View Amazon Athena queries.

HTTP flood protection is based on AWS WAF access log processing and uses WAF log files. The WAF access log type has a lower lag time, which you can use to identify HTTP flood origins more quickly when compared to CloudFront or ALB log delivery time. However, you must select the CloudFront or ALB log type in the **Activate Scanner & Probe Protection** template parameter to receive response status codes.

> **Note**
>
> If a bad bot bypasses the honeypot and directly interacts with ALB or CloudFront, the system detects malicious behavior through log analysis unless both HTTP Flood Protection and Scanner & Probe Protection are not using the Lambda log parser.

AWS Lambda log parser

The **HTTP Flood Protection** and **Scanner & Probe Protection** template parameters provide the **AWS Lambda Log Parser** option. Use the Lambda log parser only when the **AWS WAF rate-based rule** and **Amazon Athena log parser** options aren't available. A known limitation of this option is that information is processed within the context of the file being processed. For example, an IP might generate more requests or errors than the defined quota, but because this information is split into different files, each file doesn't store enough data to exceed the quota.

> **Note**
>
> Additionally, if a bad bot bypasses the honeypot and interacts directly with ALB or CloudFront, detection relies on the chosen log parser option to effectively identify and block malicious activity.

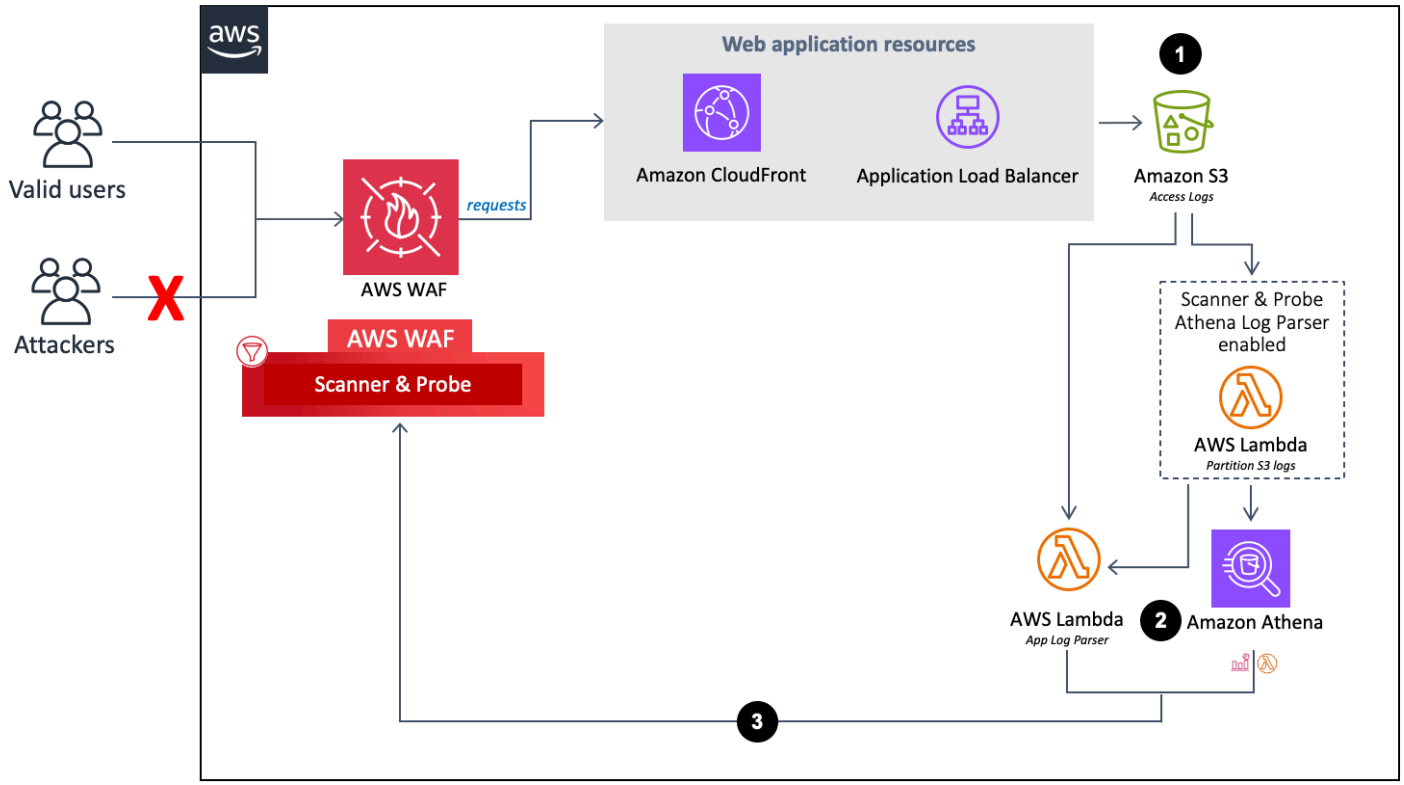
Component details

As described in the Architecture diagram, four of this solution's components use automations to inspect IP addresses and add them to the AWS WAF block list. The following sections explain each of these components in more detail.

Log parser - Application

The Application log parser helps protect against scanners and probes.

Application log parser flow.



1. When CloudFront or an ALB receives requests on behalf of your web application, it sends access logs to an Amazon S3 bucket.
   a. (Optional) If you select `Yes - Amazon Athena log parser` for the template parameters **Activate HTTP Flood Protection** and **Activate Scanner & Probe Protection**, a Lambda function moves access logs from their original folder `<customer-bucket>` `/AWSLogs` to a newly partitioned folder `<customer-bucket>` `/AWSLogs-partitioned/` `<optional-prefix>` `/year=` `<YYYY>` `/month=` `<MM>` `/day=` `<DD>` `/hour=` `<HH>`/ upon their arrival in Amazon S3.
   b. (Optional) If you select `yes` for the **Keep Data in Original S3 location** template parameter, logs remain in their original location and are copied to their partitioned folder, duplicating your log storage.

> **Note**
>
> For the Athena log parser, this solution only partitions new logs that arrive in your Amazon S3 bucket after you deploy this solution. If you have existing logs that you

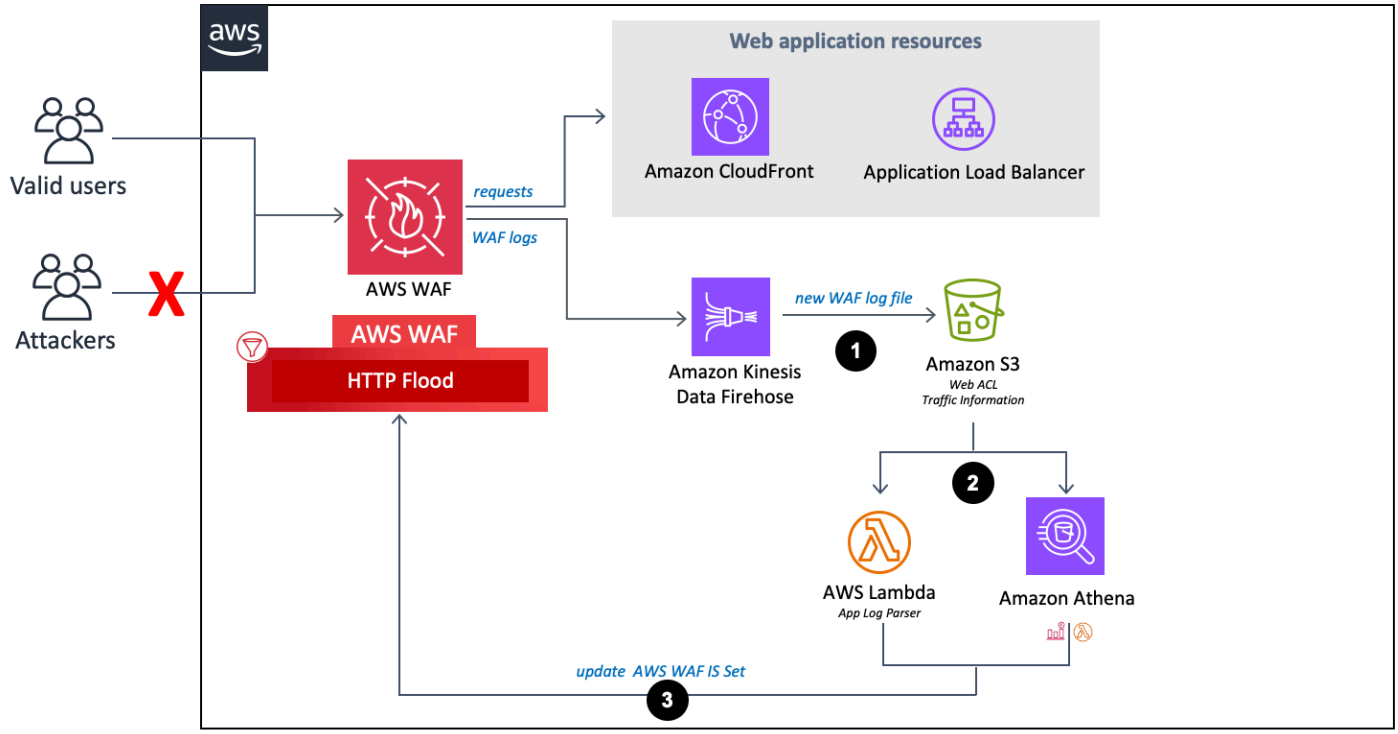
> want to partition, you must manually upload those logs to Amazon S3 after you deploy this solution.

2. Based on your selection for the template parameters **Activate HTTP Flood Protection** and **Activate Scanner & Probe Protection**, this solution processes logs using one of the following:
   a. **Lambda** - Each time a new access log is stored in the Amazon S3 bucket, the `Log Parser` Lambda function is initiated.
   b. **Athena** - By default, every five minutes the **Scanner & Probe Protection** Athena query runs, and the output pushes to AWS WAF. This process is initiated by a CloudWatch event, which starts the Lambda function responsible for running the Athena query and pushes the result into AWS WAF.
3. The solution analyzes the log data to identify IP addresses that generated more errors than the defined quota. The solution then updates an AWS WAF IP set condition to block those IP addresses for a customer-defined period of time.

Log parser - AWS WAF

If you select `yes - AWS Lambda log parser` or `yes - Amazon Athena log parser` for **Activate HTTP Flood Protection**, this solution provisions the following components, which parse AWS WAF logs to identify and block origins that flood the endpoint with a request rate greater than the quota you defined.

AWS WAF log parser flow.

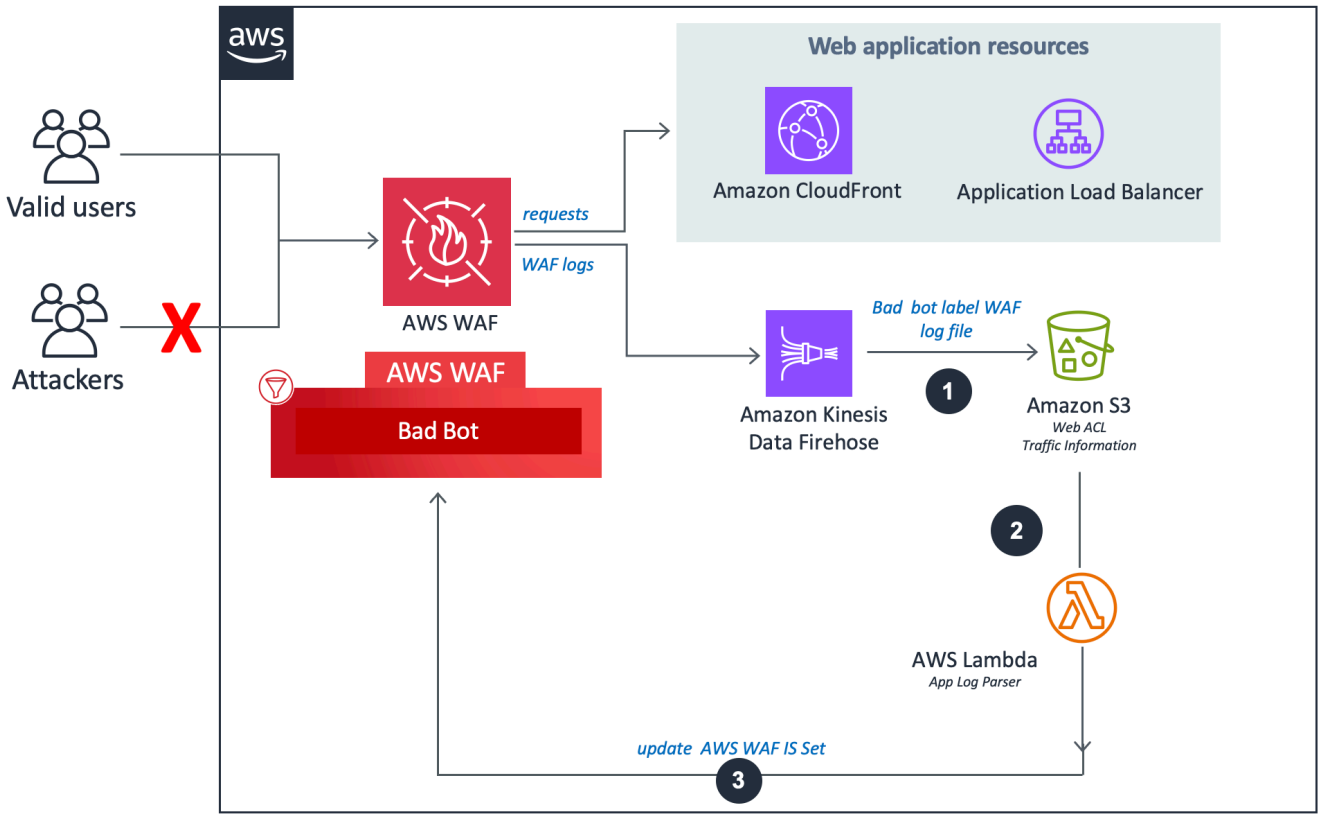


1. When AWS WAF receives access logs, it sends the logs to an Firehose endpoint. Firehose then delivers the logs to a partitioned bucket in Amazon S3 named `<customer-bucket>` / `AWSLogs/` `<optional-prefix>` `/year=` `<YYYY>` `/month=` `<MM>` `/day=` `<DD>` `/hour=` `<HH>` /
2. Based on your selection for the template parameters **Activate HTTP Flood Protection** and **Activate Scanner & Probe Protection**, this solution processes logs using one of the following:
   a. **Lambda**: Each time a new access log is stored in the Amazon S3 bucket, the `Log Parser` Lambda function is initiated.
   b. **Athena:** By default, every five minutes the scanner and probe Athena query is run and the output is pushed to AWS WAF. This process is initiated by an Amazon CloudWatch event, that then starts the Lambda function responsible for executing the Amazon Athena query, and pushes the result into AWS WAF.
3. The solution analyses the log data to identify IP addresses that sent more requests than the defined quota. The solution then updates an AWS WAF IP set condition to block those IP addresses for a customer-defined period of time.

Log parser - Bad bot

The Bad bot log parser inspects requests to the honeypot endpoint to extract their source IP address.

Bad bot log parser flow.

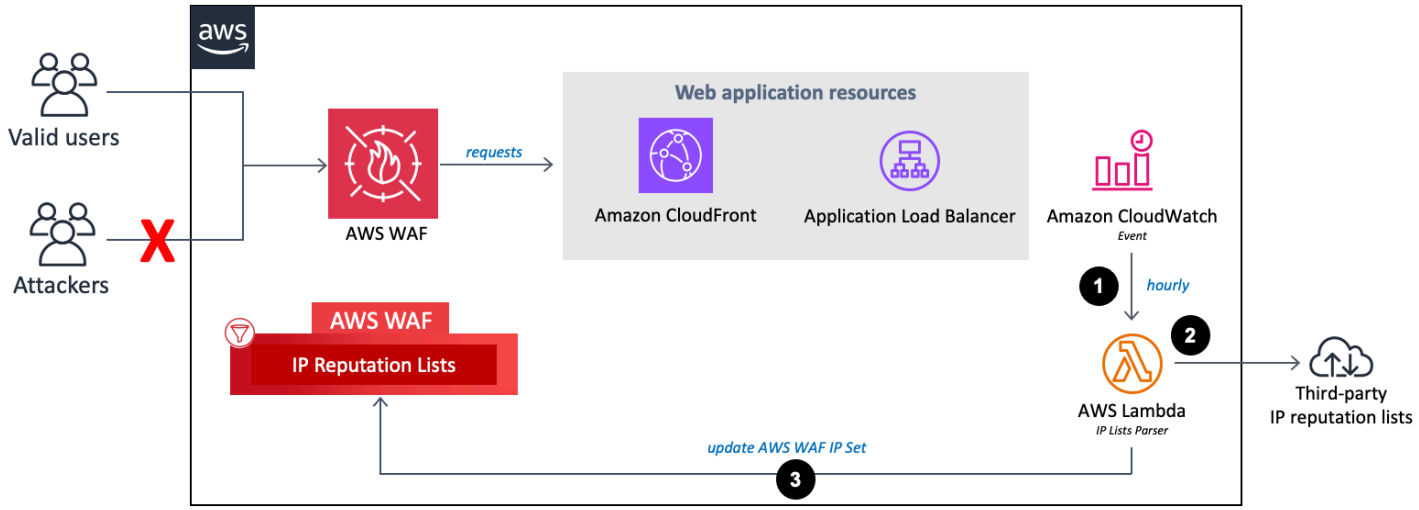


1. If `Bad Bot Protection` is activated and both HTTP Flood Protection and Scanner & Probe Protection features are disabled: The system will use the Log Lambda parser, which logs only bad bot requests based on WAF label filters.
2. The Lambda function intercepts and inspects request headers to extract the IP address of the source that accessed the trap endpoint.
3. The solution analyses the log data to identify IP addresses that sent more requests than the defined quota. The solution then updates an AWS WAF IP set condition to block those IP addresses for a customer-defined period of time.

IP lists parser

The `IP Lists Parser` Lambda function helps protect against known attackers identified in third-party IP reputation lists.

IP eputation lists parser flow.



1. An hourly Amazon CloudWatch event invokes the `IP Lists Parser` Lambda function.
2. The Lambda function gathers and parses data from three sources:
   - Spamhaus DROP and EDROP lists
   - Proofpoint Emerging Threats IP list
   - Tor exit node list
3. The Lambda function updates the AWS WAF block list with the current IP addresses.

Plan your deployment

This section describes the cost, security, Quotas, and other considerations prior to deploying the solution.

Supported AWS Regions

Depending on the template input parameters values you define, this solution requires different resources. These resources (listed in the following table) might not be available in all AWS Regions. Therefore, you must launch this solution in an AWS Region where these services are available. For the most current availability of AWS services by Region, see the AWS Regional Services List.

	AWS WAF Web ACL	AWS Glue	Amazon Athena	Amazon Kinesis Data Firehose
Endpoint type				
CloudFront	✓			
Application Load Balancer (ALB)	✓			
Activate HTTP Flood Protection				
yes - AWS Lambda log parser				✓
yes - Amazon Athena log parser		✓	✓	✓
Activate Scanner & Probe Protection				

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

> **Note**
>
> If you choose `CloudFront` as your **Endpoint**, you must deploy the solution in the US East (N. Virginia) Region (`us-east-1`).

Cost

You're responsible for the cost of the AWS services used while running the Security Automations for AWS WAF solution. The total cost for running this solution depends on the protection activated and the amount of data ingested, stored, and processed.

We recommend creating a budget through AWS Cost Explorer to help manage costs. For full details, refer to the pricing webpage for each AWS service you used in this solution.

The following tables are example cost breakdowns for running this solution in the US East (N. Virginia) Region (excludes AWS Free Tier). Prices are subject to change.

Example 1: Activate Reputation List Protection, Bad Bot Protection, AWS Lambda Log Parser for HTTP Flood Protection, and Scanner & Probe Protection

AWS service	Dimensions/Month	Cost [USD]
Amazon Data Firehose	100 GB	~$2.90
Amazon S3	100 GB	~$2.30
AWS Lambda	128 MB: 3 functions, 1M invocations, and average 500 millisecond duration per Lambda run	~$5.40

AWS service	Dimensions/Month	Cost [USD]
	512 MB: 2 functions, 1M invocations, and average 500 millisecond duration per Lambda run	
AWS WAF web ACL	1	$5.00
AWS WAF rule	4	$4.00
AWS WAF request	1M	$0.60
Total		**~$20.60 per month**

Example 2: Activate Reputation List Protection, Bad Bot Protection, Amazon Athena Log Parser for HTTP Flood Protection, and Scanner & Probe Protection

AWS service	Dimensions/Month	Cost [USD]
Amazon Data Firehose	100 GB	~$2.90
Amazon S3	100 GB	~$2.30
AWS Lambda	128 MB: 3 functions, 1M invocations ,and average 500 millisecond duration per Lambda run 512 MB: 2 functions, 7560 invocations, and average 500 millisecond duration per Lambda run	~$1.26
Amazon Athena	1.2M CloudFront objects hits or 1.2M ALB requests per day that generates a ~500 byte log record per hit or request	~$4.32

AWS service	Dimensions/Month	Cost [USD]
AWS WAF web ACL	1	$5.00
AWS WAF rule	4	$4.00
AWS WAF request	1M	$0.60
Total		**~$20.38 per month**

Example 3: Activate IP Retention for Allowed and Denied IP Sets

AWS service	Dimensions/Month	Cost [USD]
Amazon DynamoDB	1K writes and 1 MB data storage	~$0.00
AWS Lambda	128 MB: 1 function, 2K invocations, and average 500 millisecond duration per Lambda run 512 MB: 1 function, 2K invocations, and average 500 millisecond duration per Lambda run	~$0.01
Amazon CloudWatch	2K events	~$0.00
AWS WAF Web ACL	1	$5.00
AWS WAF Rule	2	$2.00
WAS WAF request	1M	$0.60
Total		**~$7.61 per month**

Cost estimate of CloudWatch logs

Some AWS services used in this solution, such as Lambda, generate CloudWatch logs. These logs incur charges. We recommend deleting or archiving logs to reduce the cost. For log archive detail, refer to Exporting log data to Amazon S3 in the *Amazon CloudWatch Logs User Guide*.

If you choose to use the Athena log parser on installation, this solution schedules a query to run against the AWS WAF or application access logs in your Amazon S3 bucket(s) as configured. You're charged based on the amount of data scanned by each query. The solution applies partitioning to logs and queries to minimize costs. By default, the solution moves application access logs from their original Amazon S3 location to a partitioned folder structure. You can also retain original, but you will be charged for duplicated log storage. This solution uses workgroups to segment workloads, and you can configure both to manage query access and costs. Refer to Cost estimate of Athena for a sample cost estimate calculation. For more information, refer to Amazon Athena Pricing.

Cost estimate of Athena

If you use the Athena log parser option while running the **HTTP Flood Protection**, **Scanner & Probe Protection** or **Bad Bot Protection** rules, you will be charged for Athena usage. By default, each Athena query runs every five minutes and scans the past four hours of data. The solution applies partitioning to logs and Athena queries to minimize costs. You can configure the number of hours of data that a query scans by changing the value for the **WAF Block Period** template parameter. However, increasing the amount of data scanned will likely increase the Athena cost.

> **Tip**
>
> The following is an example CloudFront logs cost calculation:
> On average, each CloudFront hit might generate around 500 bytes of data.
> If there are 1.2M CloudFront objects hit per day, then there will be 200K (1.2M/6) hits per four hours, assuming that data is ingested at a consistent rate. Consider your actual traffic patterns when calculate your cost.
> `[500 bytes of data] * [200K hits per four hours] = [an average 100 MB (0.0001TB) data scanned per query]`
> Athena charges $5.00 per TB of data scanned.
> `[0.0001 TB] * [$5] = [$0.0005 per query scan]`
> The Athena query runs every five minutes, which is 12 runs per hour.
> `[12 runs] * [24 hours] = [288 runs per day]`

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

Actual costs vary depending on your application's traffic patterns. For more information, refer to Amazon Athena Pricing.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit AWS Cloud Security.

IAM roles

With IAM roles, you can assign granular access, policies, and permissions to services and users on the AWS Cloud. This solution creates IAM roles with least privileges, and these roles grant the solution's resources with needed permissions.

Data

All data stored in Amazon S3 buckets and DynamoDB tables have encryption at rest. Data in transit with Firehose are also encrypted.

Protection capabilities

Web applications are vulnerable to a variety of attacks. These attacks include specially crafted requests designed to exploit a vulnerability or take control of a server; volumetric attacks designed to take down a website; or bad bots and scrapers programmed to scrape and steal web content.

This solution uses CloudFormation to configure AWS WAF rules, including AWS Managed Rules rule groups and custom rules, to block the following common attacks:

- **AWS Managed Rules** - This managed service provides protection against common application vulnerabilities or other unwanted traffic. This solution includes AWS Managed IP reputation rule groups, AWS Managed baseline rule groups, and AWS Managed use-case specific rule groups. You have the option of selecting one or more rules groups for your web ACL, up to the maximum web ACL capacity unit (WCU) quota.

- **SQL injection** - Attackers insert malicious SQL code into web requests to extract data from your database. We designed this solution to block web requests that contain potentially malicious SQL code.
- **XSS** - Attackers use vulnerabilities in a benign website as a vehicle to inject malicious client-site scripts into a legitimate user's web browser. We designed this to inspect commonly explored elements of incoming requests to identify and block XSS attacks.
- **HTTP floods** - Web servers and other backend resources are at risk of DDoS attacks, such as HTTP floods. This solution automatically invokes a rate-based rule when web requests from a client exceed a configurable quota. Alternatively, you can enforce this quota by processing AWS WAF logs using a Lambda function or Athena query.
- **Scanners and probes** - Malicious sources scan and probe internet-facing web applications for vulnerabilities, by sending a series of requests that generate HTTP 4xx error codes. You can use this history to help identify and block malicious source IP addresses. This solution creates a Lambda function or Athena query that automatically parses CloudFront or ALB access logs, counts the number of bad requests from unique source IP addresses per minute, and updates AWS WAF to block further scans from addresses that reached the defined error quota.
- **Known attacker origins (IP reputation lists)** - Many organizations maintain reputation lists of IP addresses operated by known attackers, such as spammers, malware distributors, and botnets. This solution leverages the information in these reputation lists to help you block requests from malicious IP addresses. In addition, this solution blocks attackers identified by IP reputation rule groups based on Amazon internal threat intelligence.
- **Bots and scrapers** - Operators of publicly accessible web applications need to trust that the clients accessing their content identify themselves accurately, and that they use services as intended. However, some automated clients, such as content scrapers or bad bots, misrepresent themselves to bypass restrictions. This solution helps you identify and block bad bots and scrapers.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, refer to AWS service quotas. To see the service quotas for all AWS services in the

documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS WAF quotas

AWS WAF can block a maximum of 10,000 IP address ranges in Classless Inter-Domain Routing (CIDR) notation per IP match condition. Each list that this solution creates is subject to this quota. For more information, refer to AWS WAF quotas. As of version 3.0, this solution creates two IP sets to attach to each rule, one for IPv4 and one for IPv6.

AWS WAF allows a maximum of one request per second, per account, per AWS Region for API calls to any individual `Create`, `Put`, or `Update` action. If you make these API calls outside the solution, you might encounter an API throttling issue. To prevent the issue, we recommend avoiding running other applications that make these API calls in the same account and Region where this solution is deployed.

Deployment considerations

The following sections provide constraints and considerations for implementing this solution.

AWS WAF rules

The web ACL that this solution generates is designed to offer comprehensive protection for web applications. The solution provides a set of AWS Managed Rules and custom rules that you can add to the web ACL. To include a rule, choose `yes` for the relevant parameters when launching the CloudFormation stack. See Step 1. Launch the stack for the list of parameters.

> **Note**
>
> The out-of-box solution doesn't support AWS Firewall Manager. If you want to use the rules in Firewall Manager, we recommend that you to apply customizations to its source code.

Web ACL traffic logging

If you create the stack in an AWS Region other than US East (N. Virginia) and set the **Endpoint** as `CloudFront`, you must set **Activate HTTP Flood Protection** to no or `yes - AWS WAF rate based rule`.

The other two options (`yes - AWS Lambda log parser` and `yes - Amazon Athena log parser`) require activating AWS WAF logs on a web ACL that runs in all AWS edge locations, and this isn't supported outside US East (N. Virginia). For more information about logging Web ACL traffic, refer to the AWS WAF developer guide.

Oversize handling for request components

AWS WAF doesn't support inspecting oversized content for the web request component's body, headers, or cookies. When you write a rule statement that inspects one of these request component types, you can choose one of these options to tell AWS WAF what to do with these requests:

- `yes` (continue) - Inspect the request component normally according to the rule inspection criteria. AWS WAF inspects the request component contents that are within the size limitations. This is the default option used in the solution.
- `yes - MATCH` - Treat the web request as matching the rule statement. AWS WAF applies the rule action to the request without evaluating it against the rule's inspection criteria. For a rule with `Block` action, this blocks the request with the oversize component.
- `yes - NO_MATCH` - Treat the web request as not matching the rule statement, without evaluating it against the rule's inspection criteria. AWS WAF continues its inspection of the web request by using the rest of the rules in the web ACL, like it would do for any non-matching rule.

For more information, refer to Handling oversize web request components in AWS WAF.

Multiple solution deployments

You can deploy the solution multiple times in the same account and Region. You must use a unique CloudFormation stack name and Amazon S3 bucket name for each deployment. Each unique deployment incurs additional charges and is subject to the AWS WAF quotas per account, per Region.

Minimum role permissions for deployment (Optional)

Customers can manually create an IAM role with the minimum required permissions for deployment:

- WAF Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "wafv2:CreateWebACL",
                "wafv2:UpdateWebACL",
                "wafv2:DeleteWebACL",
                "wafv2:GetWebACL",
                "wafv2:ListWebACLs",
                "wafv2:CreateIPSet",
                "wafv2:UpdateIPSet",
                "wafv2:DeleteIPSet",
                "wafv2:GetIPSet",
                "wafv2:AssociateWebACL",
                "wafv2:DisassociateWebACL",
                "wafv2:PutLoggingConfiguration",
                "wafv2:DeleteLoggingConfiguration",
                "wafv2:ListWebACLs",
                "wafv2:ListIPSets",
                "wafv2:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:wafv2:*:*:regional/webacl/*",
                "arn:aws:wafv2:*:*:regional/ipset/*",
                "arn:aws:wafv2:*:*:global/webacl/*",
                "arn:aws:wafv2:*:*:global/ipset/*"
            ]
        }
```

- Lambda Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:AddPermission",
                "lambda:RemovePermission"
            ],
```

```
            "Resource": "arn:aws:lambda:*:*:function:*"
        }
```

- Firehose Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "firehose:CreateDeliveryStream",
                "firehose:DeleteDeliveryStream",
                "firehose:DescribeDeliveryStream",
                "firehose:StartDeliveryStreamEncryption",
                "firehose:StopDeliveryStreamEncryption",
                "firehose:UpdateDestination"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/*"
        }
```

- S3 Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteBucketPolicy",
                "s3:GetBucketAcl",
                "s3:GetBucketPolicy",
                "s3:GetObject",
                "s3:PutBucketAcl",
                "s3:PutBucketPolicy",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketVersioning",
                "s3:PutEncryptionConfiguration",
                "s3:PutObject",
                "s3:PutBucketTagging",
                "s3:PutLifecycleConfiguration",
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
```

```
                "s3:ListMultipartUploadParts",
                "s3:PutBucketLogging",
                "s3:GetBucketLogging"
            ],
            "Resource": "arn:aws:s3:::*"
        }
```

- Athena Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "athena:CreateWorkGroup",
                "athena:DeleteWorkGroup",
                "athena:GetWorkGroup",
                "athena:UpdateWorkGroup",
                "athena:StartQueryExecution",
                "athena:GetQueryExecution",
                "athena:GetQueryResults",
                "athena:StopQueryExecution"
            ],
            "Resource": "arn:aws:athena:*:*:workgroup/WAF*"
        }
```

- Glue Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:UpdateDatabase",
                "glue:CreateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:UpdateTable"
            ],
```

```
            "Resource": [
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*",
                "arn:aws:glue:*:*:table/*/*",
                "arn:aws:glue:*:*:userDefinedFunction/*"
            ]
        }
```

- CloudWatch Logs Permissions

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

- CloudWatch Permissions

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

```
        }
```

- SNS Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sns:SetTopicAttributes"
            ],
            "Resource": "arn:aws:sns:*:*:*"
        }
```

- DynamoDB Permissions

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

- CloudFormation Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
```

```
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack",
                "cloudformation:ListStacks"
            ],
            "Resource": "arn:aws:cloudformation:*:*:stack/*/*"
        }
```

- Service Catalog App Registry Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:CreateApplication",
                "servicecatalog:DeleteApplication",
                "servicecatalog:GetApplication",
                "servicecatalog:TagResource",
                "servicecatalog:CreateAttributeGroup",
                "servicecatalog:DeleteAttributeGroup",
                "servicecatalog:GetAttributeGroup",
                "servicecatalog:AssociateAttributeGroup",
                "servicecatalog:DisassociateAttributeGroup",
                "servicecatalog:AssociateResource",
                "servicecatalog:DisassociateResource"
            ],
            "Resource": "arn:aws:servicecatalog:*:*:*"
        }
```

- X-Ray Permissions

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

- IAM Permissions

```
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreatePolicy",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListRoles",
                "iam:PassRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        }
```

- EventBridge Permissions

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

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation templates specify the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the templates.

Deployment process overview

Before you launch the CloudFormation template, review the architectural and configuration considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately 15 minutes.

> **Note**
>
> If you have previously deployed this solution, see Update the solution for update instructions.

Prerequisites

- Configure a CloudFront distribution
- Configure an ALB

Step 1. Launch the stack

- Launch the CloudFormation template into your AWS account.
- Enter values for the required parameters: **Stack Name** and **Application Access Log Bucket Name**.
- Review the other template parameters, and adjust if necessary.

Step 2. Associate the web ACL with your web application

- Associate your CloudFront web distribution(s) or ALB(s) with the web ACL that this solution generates. You can associate as many distributions or load balancers as you want.

Step 3. Configure web access logging

- Turn on web access logging for your CloudFront web distribution(s) or ALB(s), and send log files to the appropriate Amazon S3 bucket. Save logs in a folder matching the user-defined prefix. If no user-defined prefix is used, save logs to AWSLogs (default log prefix `AWSLogs/`). See the **Application Access Log Bucket Prefix** parameter in Step 1. Launch the stack for more information.

AWS CloudFormation templates

This solution includes one main AWS CloudFormation template and two nested templates. You can download the CloudFormation templates before deploying the solution.

Main stack

View template

aws-waf-security-automations.template - Use this template as the entry point to launch the solution in your account. The default configuration deploys an AWS WAF web ACL with preconfigured rules. You can customize the template based on your needs.

WebACL stack

View template

aws-waf-security-automations-webacl.template - This nested template provisions AWS WAF resources including a web ACL, IP, sets and other associated resources.

Firehose Athena stack

View template

aws-waf-security-automations-firehose-athena.template - This nested template provisions resources related to AWS Glue, Athena, and Firehose. It's created when you choose either the **Scanner & Probe** Athena log parser or the **HTTP Flood** Lambda or Athena log parser.

> **Note**
>
> AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

This AWS CloudFormation template deploys the Security Automations for AWS WAF solution in the AWS Cloud.

Prerequisites

This solution is designed to work with web applications deployed with CloudFront or an ALB. If you don't already have one of these resources configured, complete the applicable tasks before you launch this solution.

Configure a CloudFront distribution

Complete the following steps to configure a CloudFront distribution for your web application's static and dynamic content. Refer to the Amazon CloudFront Developer Guide for detailed instructions.

1. Create a CloudFront web application distribution. Refer to Creating a Distribution.
2. Configure static and dynamic origins. Refer to Using various origins with CloudFront distributions.
3. Specify your distribution's behavior. Refer to Values that you specify when you create or update a distribution.

> **Note**
>
> If you choose `CloudFront` as your endpoint, you must create your WAFV2 resources in the US East (N. Virginia) Region.

Configure an ALB

To configure an ALB to distribute incoming traffic to your web application, refer to Create an Application Load Balancer in the *User Guide for Application Load Balancers*.

Step 1. Launch the stack

This automated AWS CloudFormation template deploys the solution on the AWS Cloud.

1. Sign into AWS Management Console and select the **Launch Solution** to launch `waf-automation-on-aws.template` CloudFormation template.

   Launch solution

2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar. If you choose `CloudFront` as your endpoint, you must deploy the solution in the US East (N. Virginia) (`us-east-1`) Region.

   > **Note**
   >
   > Depending on the input parameters values you define, this solution requires different resources. These resources are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where these services are available. For more information, refer to Supported AWS Regions.

3. On the **Specify template** page, verify that you selected the correct template and choose **Next**.
4. On the **Specify stack details** page, assign a name to your AWS WAF configuration in the **Stack name** field. This is also the name of the web ACL that the template creates.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. To opt out of a particular feature, choose none or no as applicable. This solution uses the following default values.

Parameter	Default	Description
Stack name	`[.red]#<requires input>`	The stack name can't contain spaces. This name must be unique within your AWS account and is the name of the web ACL that the template creates.

Parameter	Default	Description
Resource Type		
Endpoint	`CloudFront`	Choose the type of resource being used. NOTE: If you choose `CloudFront` as your endpoint, you must launch the solution to create WAF resources in the US East (N. Virginia) Region (`us-east-1`).
AWS Managed IP Reputation Rule Groups		

Parameter	Default	Description
Activate Amazon IP reputation List Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Amazon IP reputation List Managed Rule Group** to the web ACL. This rule group is based on Amazon internal threat intelligence. This is useful if you want to block IP addresses typically associated with bots or other threats. Blocking these IP addresses can help mitigate bots and reduce the risk of a malicious actor discovering a vulnerable application. The required WCU is 25. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate Anonymous IP List Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Anonymous IP List Managed Rule Group** to the web ACL. This rule group blocks requests from services that permit the obfuscation of viewer identity. These include requests from VPNs, proxies, Tor nodes, and hosting providers. This rule group is useful if you want to filter out viewers that might be trying to hide their identity from your applicati on. Blocking the IP addresses of these services can help mitigate bots and evasion of geographic restrictions. The required WCU is 50. Your account should have sufficien t WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.
AWS Managed Baseline Rule Groups		

Parameter	Default	Description
Activate Core Rule Set Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Core Rule Set Managed Rule Group** to the web ACL. This rule group provides protection against exploitat ion of a wide range of vulnerabilities, including some of the high risk and commonly occurring vulnerabilities. Consider using this rule group for any AWS WAF use case. The required WCU is 700. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate Admin Protection Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Admin Protection Managed Rule Group** to the web ACL. This rule group blocks external access to exposed administrative pages. This might be useful if you run third-party software or want to reduce the risk of a malicious actor gaining administrative access to your application. The required WCU is 100. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate Known Bad Inputs Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Known Bad Inputs Managed Rule Group** to the web ACL. This rule group blocks external access to exposed administrative pages. This might be useful if you run third-party software or want to reduce the risk of a malicious actor gaining administrative access to your application. The required WCU is 100. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.
AWS Managed Use-case Specific Rule Group		

Parameter	Default	Description
Activate SQL Database Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **SQL Database Managed Rule Group** to the web ACL. This rule group blocks request patterns associate d with exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries. Evaluate this rule group for use if your applicati on interfaces with an SQL database. Using the SQL injection custom rule is optional if you already have AWS managed SQL rule group activated. The required WCU is 200. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate Linux Operating System Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Linux Operating System Managed Rule Group** to the web ACL. This rule group blocks request patterns associated with the exploitation of vulnerabilities specific to Linux, including Linux-specific Local File Inclusion (LFI) attacks. This can help prevent attacks that expose file contents or run code for which the attacker should not have had access. Evaluate this rule group if any part of your application runs on Linux. You should use this rule group in conjunction with the POSIX operating system rule group. The required WCU is 200. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate POSIX Operating System Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Core Rule Set Managed Rule Group Protection** to the web ACL. This rule group blocks request patterns associated with the exploitation of vulnerabilities specific to POSIX and POSIX-like operating systems, including LFI attacks. This can help prevent attacks that expose file contents or run code for which the attacker should not have had access. Evaluate this rule group if any part of your application runs on a POSIX or POSIX-like operating system. The required WCU is 100. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate Windows Operating System Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **Windows Operating System Managed Rule Group** to the web ACL. This rule group blocks request patterns associated with the exploitation of vulnerabilities specific to Windows, like remote execution of PowerShell commands. This can help prevent exploitation of vulnerabilities that permit an attacker to run unauthorized commands or run malicious code. Evaluate this rule group if any part of your application runs on a Windows operating system. The required WCU is 200. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate PHP Application Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **PHP Application Managed Rule Group** to the web ACL. This rule group blocks request patterns associated with the exploitation of vulnerabilities specific to the use of the PHP programming language, including injection of unsafe PHP functions. This can help prevent exploitation of vulnerabilities that permit an attacker to remotely run code or commands for which they are not authorized. Evaluate this rule group if PHP is installed on any server with which your application interfaces. The required WCU is 100. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.

Parameter	Default	Description
Activate WordPress Application Managed Rule Group Protection	no	Choose `yes` to turn on the component designed to add **WordPress Application Managed Rule Group** to the web ACL. This rule group blocks request patterns associated with the exploitation of vulnerabilities specific to WordPress sites. Evaluate this rule group if you are running WordPress. This rule group should be used in conjunction with the SQL database and PHP application rule groups. The required WCU is 100. Your account should have sufficient WCU capacity to avoid web ACL stack deployment failure due to exceeding the capacity limit. For more information, see AWS Managed Rules rule groups list.
Custom Rule - Scanner & Probes		

Parameter	Default	Description
Activate Scanner & Probe Protection	`yes - AWS Lambda log parser`	Choose the component used to block scanners and probes. Refer to Log parser options for more information about the tradeoffs related to the mitigation options.

Parameter	Default	Description
Application Access Log Bucket Name	`[.red]<requires input>`	If you chose `yes` for the **Activate Scanner & Probe Protection** parameter , enter the name of the Amazon S3 bucket (new or existing) where you want to store access logs for your CloudFront distribution(s) or ALB(s). If you're using an existing Amazon S3 bucket, it must be located in the same AWS Region where you are deploying the CloudFormation template. You should use a different bucket for each solution deployment. To deactivate this protection, ignore this parameter. NOTE: Turn on web access logging for your CloudFront web distribution(s) or ALB(s) to send log files to this Amazon S3 bucket. Save logs in the same prefix defined in the stack (default prefix `AWSLogs/`). See the **Application Access Log Bucket Prefix** parameter for more information.

Parameter	Default	Description
Application Access Log Bucket Prefix	`AWSLogs/`	If you chose `yes` for the **Activate Scanner & Probe Protection** parameter, you can enter an optional user defined prefix for the application access logs bucket above. If you chose `CloudFront` for the **Endpoint** parameter, you can enter any prefix such as `yourprefix/` . If you chose `ALB` for the **Endpoint** parameter, you must append `AWSLogs/` to your prefix such as `yourprefix/AWSLogs/` . Use `AWSLogs/` (default) if there isn't a user-defined prefix. To deactivate this protection, ignore this parameter.

Parameter	Default	Description
Is bucket access logging turned on?	no	Choose `yes` if you entered an existing Amazon S3 bucket name for the **Application Access Log Bucket Name** parameter and the server access logging for the bucket is already turned on. If you choose no, the solution turns on server access logging for your bucket. If you chose no for the **Activate Scanner & Probe Protection** parameter, ignore this parameter.
Error Threshold	`50`	If you chose `yes` for the **Activate Scanner & Probe Protection** parameter, enter the maximum acceptable bad requests per minute, per IP address. If you chose no for the **Activate Scanner & Probe Protection** parameter, ignore this parameter.

Parameter	Default	Description
Keep Data in Original S3 Location	no	If you chose `yes - Amazon Athena log parser` for the **Activate Scanner & Probe Protection** parameter, the solution applies partitioning to application access log files and Athena queries. By default, the solution moves log files from their original location to a partitioned folder structure in Amazon S3. Choose `yes` if you also want to keep a copy of the logs in their original location. This will duplicate your log storage. If you didn't choose `yes - Amazon Athena log parser` for the **Activate Scanner & Probe Protection** parameter, ignore this parameter.
Custom Rule - HTTP Flood		
Activate HTTP Flood Protection	`yes - AWS WAF rate-based rule`	Select the component used to block HTTP flood attacks. Refer to Log parser options for more information about the tradeoffs related to the mitigation options.

Parameter	Default	Description
Default Request Threshold	`100`	If you chose `yes` for the **Activate HTTP Flood Protection** parameter, enter the maximum acceptable requests per five minutes, per IP address. If you chose `yes - AWS WAF rate-based rule` for the **Activate HTTP Flood Protection** parameter, the minimum acceptable value is `10`. If you chose `yes - AWS Lambda log parser` or `yes - Amazon Athena log parser` for the **Activate HTTP Flood Protection** parameter, it can be any value. To deactivate this protection, ignore this parameter.

Parameter	Default	Description
Request Threshold by Country	*<optional input>*	If you chose `yes - Amazon Athena log parser` for the **Activate HTTP Flood Protection** parameter, you can enter a threshold by country following this JSON format `{"TR":50, "ER":150}` . The solution uses these thresholds for the requests originated from the specified countries . The solution uses the **Default Request Threshold** parameter for the remaining requests. NOTE: If you define this parameter, the country will automatically be included in Athena query group, along with IP and other optional group-by fields that you can select with the **Group By Requests in HTTP Flood Athena Query** parameter. + If you chose to deactivate this protection, ignore this parameter.

Parameter	Default	Description
Group By Requests in HTTP Flood Athena Query	None	If you chose `yes - Amazon Athena log parser` for the **Activate HTTP Flood Protection** parameter, you can choose a group-by field to count requests per IP and the selected group-by field. For example, if you choose `URI`, the solution counts the requests per IP and URI. If you chose to deactivate this protection, ignore this parameter.
WAF Block Period	`240`	If you chose `yes - AWS Lambda log parser` or `yes - Amazon Athena log parser` for the **Activate Scanner & Probe Protection** or **Activate HTTP Flood Protectio n** parameters, enter the period (in minutes) to block applicable IP addresses. To deactivate log parsing, ignore this parameter.

Parameter	Default	Description
Athena Query Run Time Schedule (Minute)	5	If you chose `yes - Amazon Athena log parser` for the **Activate Scanner & Probe Protection** or **Activate HTTP Flood Protection** parameters, you can enter a time interval (in minutes) over which the Athena query runs. By default, the Athena query runs every 5 minutes. If you chose to deactivate these protections, ignore this parameter.

Parameter	Default	Description
Rule Keys	`IP`	If you chose `yes - AWS WAF rate-based rule` for the **Activate HTTP Flood Protection** parameter, configure this rule to use various other combinati ons of aggregation keys. Available options: IP (Default) IP+Custom Header (if this option is selected, `Rule Keys Custom Header` is mandatory) IP+URI IP+HTTP METHOD For more information, see WAF rule rate based aggregation options.
Rule Keys Custom Header	no	If you chose `IP+Custom Header` for the **Rule Keys** parameter, enter the name of the custom header to use for request aggregation. For more information, see WAF rule statement type rate based aggregation options.

Parameter	Default	Description
Time Window Threshold (Minutes)	5	Time window threshold in minutes for HTTP flood protection. Applies to both rate-based rule and lambda log parser. Options available: [1, 2, 5, 10]. If you chose `yes - AWS WAF rate-based rule` for the **Activate HTTP Flood Protection** parameter will be used for evaluation time windows. For more informati on, see WAF web ACL rate based statement. If you chose `yes - AWS Lambda log parser` for the **Activate HTTP Flood Protection** parameter will be used for evaluation period in addition to the block period.
Custom Rule - Bad Bot		
Activate Bad Bot Protection	`yes`	Choose `yes` to turn on the component designed to block bad bots and content scrapers.

Parameter	Default	Description
ARN of an IAM role that has write access to CloudWatch logs in your account	*<optional input>*	Provide an optional ARN of an IAM role that has write access to CloudWatch logs in your account. For example: `ARN: arn:aws:iam::account_id:role/myrolename` . If you leave this parameter blank (default), the solution creates a new role for you.
Custom Rule - Third Party IP Reputation Lists		
Activate Reputation List Protection	`yes`	Choose `yes` to block requests from IP addresses on third-party reputation lists (supported lists include Spamhaus, Emerging Threats, and Tor exit node).
Legacy Custom Rules		

Parameter	Default	Description
Activate SQL Injection Protection	`yes`	Choose `yes` to turn on the component designed to block common SQL injection attacks. Consider activating it if you aren't using an AWS managed core rule set or AWS managed SQL database rule group. You can choose one of the options (`yes` (continue), `yes - MATCH`, or `yes - NO_MATCH`) that you want AWS WAF to handle oversized request exceeding 8 KB (8192 bytes). By default, `yes` inspects the request component contents that are within the size limitations according to the rule inspection criteria. For more information, refer to Handling oversize web request components. Choose no to deactivat e this feature. NOTE: The CloudFormation stack adds the selected oversize handling option to the default SQL injection protection rule and deploys it into your AWS account. If you customized the rule outside of CloudForm

Parameter	Default	Description
		ation, your changes will be overwritten after the stack update.

Parameter	Default	Description
Sensitivity Level for SQL Injection Protection	`LOW`	Choose the sensitivity level that you want AWS WAF to use to inspect for SQL injection attacks. `HIGH` detects more attacks, but might generate more false positives. `LOW` is generally a better choice for resources that already have other protectio ns against SQL injection attacks or that have a low tolerance for false positives. For more information, refer to AWS WAF adds sensitivi ty levels for SQL injection rule statements and Sensitivi tyLevel property in the *AWS CloudFormation User Guide*. If you choose to deactivat e SQL injection protection, ignore this parameter. NOTE: The CloudFormation stack adds the selected sensitivi ty level to the default SQL injection protectio n rule and deploys it into your AWS account. If you customized the rule outside of CloudFormation, your

Parameter	Default	Description
		changes will be overwritten after the stack update.

Parameter	Default	Description
Activate Cross-site Scripting Protection	`yes`	Choose `yes` to turn on the component designed to block common XSS attacks. Consider activatin g it if you aren't using an AWS managed core rule set. You can also select one of the options (`yes` (continue), `yes - MATCH`, or `yes - NO_MATCH`) that you want AWS WAF to handle oversized request exceeding 8 KB (8192 bytes). By default, `yes` uses the `Continue` option, which inspects the request component contents that are within the size limitatio ns according to the rule inspection criteria. For more information, refer to Oversize handling for request components. Choose `no` to deactivat e this feature. NOTE: The CloudFormation stack adds the selected oversize handling option to the default cross-site scripting rule and deploys it into your AWS account. If you customized the rule outside of CloudFormation, your

Parameter	Default	Description
		changes will be overwritten after the stack update.
Allowed and Denied IP Retention Settings		
Retention Period (Minutes) for Allowed IP Set	`-1`	If you want to activate IP retention for the Allowed IP set, enter a number (15 or greater) as the retention period (minutes) . IP addresses reaching the retention period expire, and the solution removes them from the IP set. The solution supports a minimum 15-minute retention period. If you enter a number between `0` and 15, the solution treats it as 15. Leave it as `-1` (default) to turn off IP retention.

Parameter	Default	Description
Retention Period (Minutes) for Denied IP Set	`-1`	If you want to activate IP retention for the Denied IP set, enter a number (15 or greater) as the retention period (minutes) . IP addresses reaching the retention period expire, and the solution removes them from the IP set. The solution supports a minimum 15-minute retention period. If you enter a number between `0` and 15, the solution treats it as 15. Leave it as `-1` (default) to turn off IP retention.
Email for receiving notification upon Allowed or Denied IP Sets expiration	*<optional input>*	If you activated the IP retention period parameters (see two previous parameters) and want to receive an email notification when IP addresses expire, enter a valid email address. If you didn't activate IP retention or want to turn off email notifications, leave it blank (default).
Advanced Settings		

Parameter	Default	Description
Retention Period (Days) for Log Groups	365	If you want to activate retention for the CloudWatch Log Groups, enter a number (1 or greater) as the retention period (days). You can choose a retention period between one day (1) and ten years (3650). By default logs will expire after one year. Set it to `-1` to keep the logs indefinitely.

6. Choose **Next.**
7. On the **Configure stack options** page, you can specify tags (key-value pairs) for resources in your stack and set additional options. Choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the boxes acknowledging that the template will create IAM resources and any additional capabilities required.
9. Choose **Submit** to deploy the stack.

   View the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a status of CREATE_COMPLETE in approximately 15 minutes.

> **Note**
>
> In addition to the `Log Parser` and `IP Lists Parser` AWS Lambda functions, this solution includes the `helper` and `custom-resource` Lambda functions, which run only during initial configuration or when resources are updated or deleted.
> When using this solution, you will see all functions in the AWS Lambda console, but only the three primary solution functions are regularly active. Don't delete the other two functions; they are necessary to manage associated resources.

To see details about the stack resources, choose the **Outputs** tab. This includes the **BadBotHoneypotEndpoint** value. Remember this value because you will be used it in Embed the Honeypot link in your web application.

Step 2. Associate the web ACL with your web application

Update your CloudFront distribution(s) or ALB(s) to activate AWS WAF and logging using the resources you generated in Step 1. Launch the stack.

1. Sign in to the AWS WAF console.
2. Choose the web ACL that you want to use.
3. On the **Associated AWS resources** tab, choose **Add AWS resources**.
4. Under **Resource type**, choose the CloudFront distribution or ALB.
5. Select a resource from the list, then choose **Add** to save your changes.

Step 3. Configure web access logging

Configure CloudFront or your ALB to send web access logs to the appropriate Amazon S3 bucket so that this data is available for the Log Parser Lambda function.

Store web access logs from a CloudFront distribution

1. Sign in to the Amazon CloudFront console.
2. Select your web application's distribution, and choose **Distribution Settings.**
3. On the **General** tab, choose **Edit**.
4. For **AWS WAF Web ACL**, choose the web ACL solution created (the **Stack name** parameter).
5. For **Logging**, choose **On**.
6. For **Bucket for Logs**, choose the S3 bucket that you want to use for storing web access logs. This can be a new or existing S3 bucket that is used in the main stack and has permission for CloudFront to write logs. The drop-down list enumerates the buckets associated with the current AWS account. For more information, see Getting started with a basic CloudFront distribution in the *Amazon CloudFront Developer Guide*.
7. Set the log prefix to the prefix used for deploying the solution. You can find the prefix in the main stack, **Parameters** tab, **AppAccessLogBucketPrefixParam** (default `AWSLogs/`).
8. Choose **Yes, edit** to save your changes.

For more information, refer to Configuring and using standard logs (access logs) in the *Amazon CloudFront Developer Guide*.

Store web access logs from an Application Load Balancer

1. Sign in to the Amazon Elastic Compute Cloud (Amazon EC2) console.
2. In the navigation pane, choose **Load Balancers**.
3. Select your web application's ALB.
4. On the **Description** tab, choose **Edit attributes**.
5. Choose **Enable access logs**.
6. For **S3 location**, type the name of the S3 bucket that you want to use for storing web access logs. This can be a new or existing S3 bucket that is used in the main stack and has permission for Application Load Balancer to write logs.
7. Set the log prefix to the prefix used for deploying the solution. You can find the prefix in the main stack, **Parameters** tab, **AppAccessLogBucketPrefixParam** (default `AWSLogs/`).
8. Choose **Save**.

For more information, refer to Access Logs for your Application Load Balancer in the *Elastic Load Balancing User Guide*.

Update the solution

If you previously deployed the solution, follow this procedure to update the solution's CloudFormation stack to get the latest version of the solution's framework. Before you update the stack, read Update considerations carefully.

1. Sign in to the AWS CloudFormation console.
2. Select **Stacks** in the left navigation menu.
3. Select your existing `aws-waf-security-automations` CloudFormation stack.
4. Choose **Update**.
5. Select **Replace current template**.
6. Under **Specify template**:
   a. Select **Amazon S3 URL**.
   b. Copy the link of the `aws-waf-security-automations.template` AWS CloudFormation.
   c. Paste the link in the **Amazon S3 URL** box.
   d. Verify that the correct template URL shows in the **Amazon S3 URL** text box.
   e. Choose **Next**.
   f. Choose **Next** again.
7. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the stack for details about the parameters.
8. Choose **Next**.
9. On the **Configure stack options** page, choose **Next**.
10. On the **Review** page, review and confirm the settings.
11. Select the box acknowledging that the template might create IAM resources.
12. Choose **View change set** and verify the changes.
13. Choose **Update stack** to deploy the stack.

You can see the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of UPDATE_COMPLETE in approximately 15 minutes.

Update considerations

The following sections provide constraints and considerations for updating this solution.

Resource type update

You must deploy a new stack to update the **Endpoint** parameter after creating the stack. Don't change the **Endpoint** parameter when updating the stack.

WAFV2 upgrade

Starting from version 3.0, this solution supports AWS WAFV2. We replaced all the AWS WAF Classic API calls with AWS WAFV2 API calls. This removes dependencies on Node.js and uses the most up-to-date Python runtime. To continue using this solution with the latest features and improvements, you must deploy version 3.0 or higher as a new stack.

Customizations at stack update

The out-of-box solution deploys a set of AWS WAF rules with default configurations into your AWS account with the CloudFormation stack. We don't recommend applying customizations to rules deployed by the solution. Stack updates overwrite these changes. If you need customized rules, we recommend creating separate rules outside of the solution.

Bad bot Protection upgrade

In version **4.1.0**, the **Access Handler** Lambda with **API Gateway** has been deprecated and replaced with enhanced log functionality from the `Log parser - Bad bot` feature. Instead of using direct requests through API Gateway, the solution now re-uses the log stream to detect bad bots.

Previous Implementation:

1. Required Access Handler Lambda and API Gateway.
2. Used honeypot endpoint for direct request handling.
3. Required embedding honeypot endpoint in websites.

New Implementation (4.1.0+): The **Bad Bot Protection** log parser now:

1. Inspects requests to the honeypot endpoint through logs.

2. Processes requests when **Bad Bot Protection** is activated.
3. Uses WAF filter **BadBotRuleFilter** to identify bad bot requests.
4. Analyzes log data to identify IP addresses exceeding defined quotas.
5. Updates AWS WAF IP set conditions to block identified addresses.

This change simplifies the architecture by eliminating duplicate functionality and leveraging existing log processing capabilities.

CDK upgrade

Starting from version v4.1.0, this solution is supported by CDK. If migrating from a version below v4.1.0 . Use the new template and update solution in Cloudformation . Then you can start to update the solution locally via your terminal using cdk deploy (see README for more information) If you try to use cdk deploy directly then you might see this error : Insufficient indentation in flow collection

The other way to update the solution will be to use the template provided by the solution and to go to the Cloudformation section of the AWS console and click update the solution and paste the new template there.

> **Note**
>
> If you are upgrading from version 3.0 or 3.1 to version 3.2 or newer of this solution, and you have manually inserted IP addresses into the allowed or denied IP set, you will be at risk of losing those IP addresses. To prevent that from happening, make a copy of the IP addresses in the allowed or denied IP set before upgrading the solution. Then after you complete the upgrade, add the IP addresses back to the IP set as needed. Refer to the get-ip-set and update-ip-set CLI commands. If you're already using version 3.2 or newer, ignore this step.

Uninstall the solution

To uninstall the solution, delete the CloudFormation stacks:

1. Sign in to the AWS CloudFormation console.
2. Select the solution's parent stack. All other solution stacks will be deleted automatically.
3. Choose **Delete**.

> **Note**
>
> Uninstalling the solution deletes all the AWS resources used by the solution except for the Amazon S3 buckets. If some IP sets fail to delete due to rate exceeded throttling issue caused by the AWA WAF API quotas, manually delete those IP sets, and then delete the stack.

Use the solution

This section provides detailed instructions to use the solution after you deploy the solution.

Modify the allowed and denied IP sets (optional)

After deploying this solution's CloudFormation stack, you can manually modify the allowed and denied IP sets to add or remove IP addresses as necessary.

1. Sign in to the AWS WAF console.
2. In the left navigation pane, choose **IP Sets**.
3. Choose **IP set for Allowed List** and add IP addresses from trusted sources.
4. Choose **IP set for Denied List** and add IP addresses you want to block.

Embed the Honeypot link in your web application (optional)

If you chose `yes` for the **Activate Bad Bot Protection** parameter in Step 1. Launch the stack, the CloudFormation template creates a trap endpoint to a low-interaction production honeypot. This trap is intended to detect and divert inbound requests from content scrapers and bad bots. Valid users won't attempt to access this endpoint.

This component enhances bad bot detection by monitoring direct connections to an Application Load Balancer (ALB) or Amazon CloudFront, in addition to the honeypot mechanism. If a bot bypasses the honeypot and attempts to interact with ALB or CloudFront, the system analyzes request patterns and logs to identify malicious activity. When a bad bot is detected, its IP address is extracted and added to an AWS WAF block list to prevent further access. Bad bot detection operates through a structured logic chain, ensuring comprehensive threat coverage:

- HTTP Flood Protection Lambda Log Parser – Collects bad bot IPs from log entries during flood analysis.
- Scanner & Probe Protection Lambda Log Parser – Identifies bad bot IPs from scanner-related log entries.
- HTTP Flood Protection Athena Log Parser – Extracts bad bot IPs from Athena logs, using partitions across query run.
- Scanner & Probe Protection Athena Log Parser – Retrieves bad bot IPs from scanner-related Athena logs, using the same partitioning strategy.

- Fallback Detection – If both HTTP Flood Protection and Scanner & Probe Protection are disabled, the system relies on the Log Lambda parser, which logs bot activity based on WAF label filters.

Use one of the following procedures to embed the honeypot link for requests from either a CloudFront distribution.

Create a CloudFront Origin for the Honeypot Endpoint

Use this procedure for web applications that are deployed with a CloudFront distribution. With CloudFront, you can include a `robots.txt` file to help identify content scrapers and bots that ignore the robots exclusion standard. Complete the following steps to embed the hidden link and then explicitly disallow it in your `robots.txt` file.

1. Sign in to the AWS CloudFormation console.
2. Choose the stack that you built in Step 1. Launch the stack
3. Choose the **Outputs** tab.
4. From the **BadBotHoneypotEndpoint** key, copy the endpoint URL.
   - The behavior path (`/ProdStage`)
5. Embed this endpoint link in your content pointing to the honeypot. Hide this link from your human users. As an example, review the following code sample: `<a href="/behavior_path" rel="nofollow" style="display: none" aria-hidden="true">honeypot link</a>`.
6. Modify the `robots.txt` file in the root of your website to explicitly disallow the honeypot link, as follows:

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

> **Important**
>
> No path registration in CloudFront is required as requests are: Blocked by WAF **BadBotRuleFilter**. Solution collected in logs automatically. Processed by the **Log parser** lambda. This simplified approach uses the WAF logs directly instead of requiring additional endpoint configuration, making the bad bot detection process more efficient through log analysis

> **Note**
>
> It's your responsibility to verify what tag values work in your website environment. Don't use `rel="nofollow"` if your environment doesn't observe it. For more information about robots meta tags configuration, refer to the Google developer's guide. Modify the `robots.txt` file in the root of your website to explicitly disallow the honeypot link, as follows:

Embed the Honeypot endpoint as an external link

> **Note**
>
> These rules use the source IP address from the web request origin. If you have traffic that goes through one or more proxies or load balancers, the web request origin will contain the address of the last proxy, and not the originating address of the client.

Use this procedure for web applications.

1. Sign in to the AWS CloudFormation console.
2. Choose the stack that you built in Step 1. Launch the stack.
3. Choose the **Outputs** tab.
4. From the **BadBotHoneypotEndpoint** key, copy the endpoint URL.

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

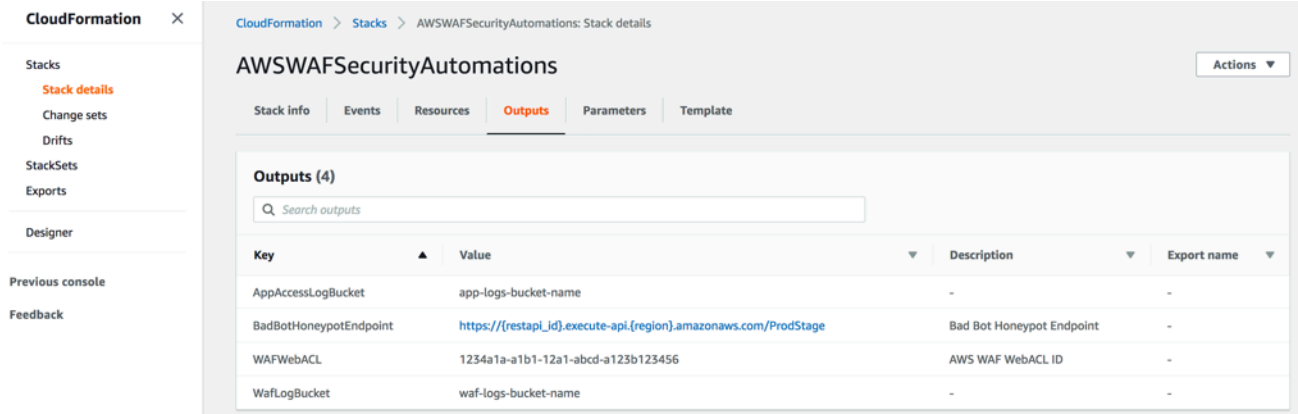
> **Note**
>
> This procedure uses `rel=nofollow` to instruct robots to not access the honeypot URL. However, because the link is embedded externally, you can't include a `robots.txt` file to explicitly disallow the link. It's your responsibility to verify what tags work in your website environment. Don't use `rel="nofollow"` if your environment doesn't observe it.

Use Lambda log parser JSON file

Use Lambda log parser JSON file for HTTP Flood protection

If you chose `Yes - AWS Lambda log parser` for the **Activate HTTP Flood Protection** template parameter, this solution creates a configuration file named `<stack_name>-waf_log_conf.json` and uploads it to the Amazon S3 bucket used to store the AWS WAF log files. To find the bucket name, refer to the **WafLogBucket** variable in the CloudFormation output. The following figure shows an example.

Screenshot depicting a screen labeled AWSWAFSecurityAutomations and listing four outputs



If you edit and overwrite the `<stack_name>-waf_log_conf.json` file on Amazon S3, the `Log Parser` Lambda function considers the new values when processing new AWS WAF log files. The following is a sample configuration file:

Screenshot of a sample configuration file

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

Parameters include the following:

- General:
  - **Request threshold (required)** - The maximum acceptable requests per five minutes, per IP address. This solution uses the value you define when provisioning or updating the CloudFormation stack.
  - **Block period (required)** - The period (in minutes) to block applicable IP addresses. This solution uses the value you define when provisioning or updating the CloudFormation stack.
  - **Ignored suffixes** - Requests accessing this type of resource don't count to request threshold. By default, this list is empty.
- **URI list** - Uuse this to define a custom request threshold and block period for specifics URLs. By default, this list is empty.

When WAF logs arrive in the **WafLogBucket**, they will be processed by Lambda log parser function using the configurations in your configuration file. The solution writes the result to an output file named `<stack_name>-waf_log_out.json` in the same bucket. If the output file contains a list of the IP addresses identified as attackers, the solution adds them to the WAF IP set for **HTTP Flood**, and they're blocked from accessing your application. If the output files have no IP addresses, check if your configuration file is valid or if the rate limit has exceeded according to the configuration file.

Use Lambda log parser JSON file for scanner and probe protection

If you chose `Yes - AWS Lambda log parser` for the **Activate Scanner & Probe Protection** template parameter, this solution creates a configuration file named `<stack_name>-app_log_conf.json` and uploads it to the defined Amazon S3 bucket used to store CloudFront or Application Load Balancer log files.

If you edit and overwrite on the `<stack_name>-app_log_conf.json` on Amazon S3, the `Log Parser` Lambda function considers the new values when processing new AWS WAF log files. The following is a sample configuration file:

Screenshot of configuration file

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

Parameters include the following:

- General:
  - **Error threshold (required)** - The maximum acceptable bad requests per minute, per IP address. This solution uses the value you defined when provisioning or updating the CloudFormation stack.
  - **Block period (required)** - The period (in minutes) to block applicable IP addresses. This solution uses the value you defined when provisioning or updating the CloudFormation stack.
  - **Error codes** - Return status code considered errors. By default, the list considers the following HTTP status codes as errors: `400 (Bad Request)`, `401 (Unauthorized)`, `403 (Forbidden)`, `404 (Not Found)`, and `405 (Method Not Allowed)`.
- **URI list** - Use this to define a custom request threshold and block period for specifics URLs. By default, this list is empty.

When application access logs arrive in the **AppAccessLogBucket**, the `Log Parser` Lambda function processes them using the configurations in your configuration file. The solution writes the result to an output file named `<stack_name>`-app_log_out.json` in the same bucket. If the output file contains a list of the IP addresses identified as attackers, the solution adds them to the WAF IP set for **Scanner & Probe** and blocks them from accessing your application. If the output files have no IP addresses, check if your configuration file is valid or if the rate limit has been exceeded according to the configuration file.

Use country and URI in HTTP flood Athena log parser

You can group by IPs along with country and URI in the Athena query to detect and block HTTP flood attacks that have unpredictable URI patterns. To do so, select one of the options (`Country`, `URI`, `Country and URI`) for the **Group By Requests in HTTP Flood Athena Query** parameter when launching the stack.

You can also enter a request threshold by country using the **Request Threshold by Country** parameter. For example, `{"TR": 50,"ER":150}`. The solution uses these thresholds on the requests originated from these specified countries. The solution uses the default threshold on the requests from other countries.

> **Note**
>
> If you define a threshold by country, the solution automatically includes the country in the Athena query group-by clause. For more information, see the parameters table in Step 1. Launch the stack.

The solution counts the request threshold in a five-minute period by default. This is configurable with the **Athena Query Run Time Schedule (Minute)** parameter.

> **Note**
>
> The Athena query calculates threshold per minute by dividing the request threshold by the time period. For example:
> *Request threshold (default threshold or threshold by country): 100*
> *Athena Query Run Time Schedule: 5*
> *Request threshold per minute: 20 = 100 / 5*

View Amazon Athena queries

If you selected `Yes - Amazon Athena log parser` for the **Activate HTTP Flood Protection** or **Activate Scanner & Probe Protection** template parameters, this solution creates and runs Athena queries for CloudFront or ALB (`ScannersProbesLogParser`) or AWS WAF logs (`HTTPFloodLogParser`), parses the output, and updates AWS WAF accordingly.

To improve performance and keep costs low, the solution partitions logs based on timestamps in the file names. The solution dynamically generates Athena queries to use partition keys (year, month, day, and hour). By default, queries run every five minutes. You can configure their run schedules by changing the value of the **Athena Query Run Time Schedule (Minute)** template parameter. Each query run scans the last four to five hours of data by default. You can configure the amount of data that a query scans by changing the value of the **WAF Block Period** template parameter. The solution also places queries in separate workgroups to manage query access and costs.

> **Note**
>
> Verify that Athena is configured to access the AWS Glue Data Catalog. This solution creates the access logs data catalog in AWS Glue and configures an Athena query to process the data. If Athena isn't configured correctly, the query doesn't run. For more information, refer to Upgrading to the latest AWSAWS Glue Data Catalog step-by-step.

Use the following procedure to view these queries:

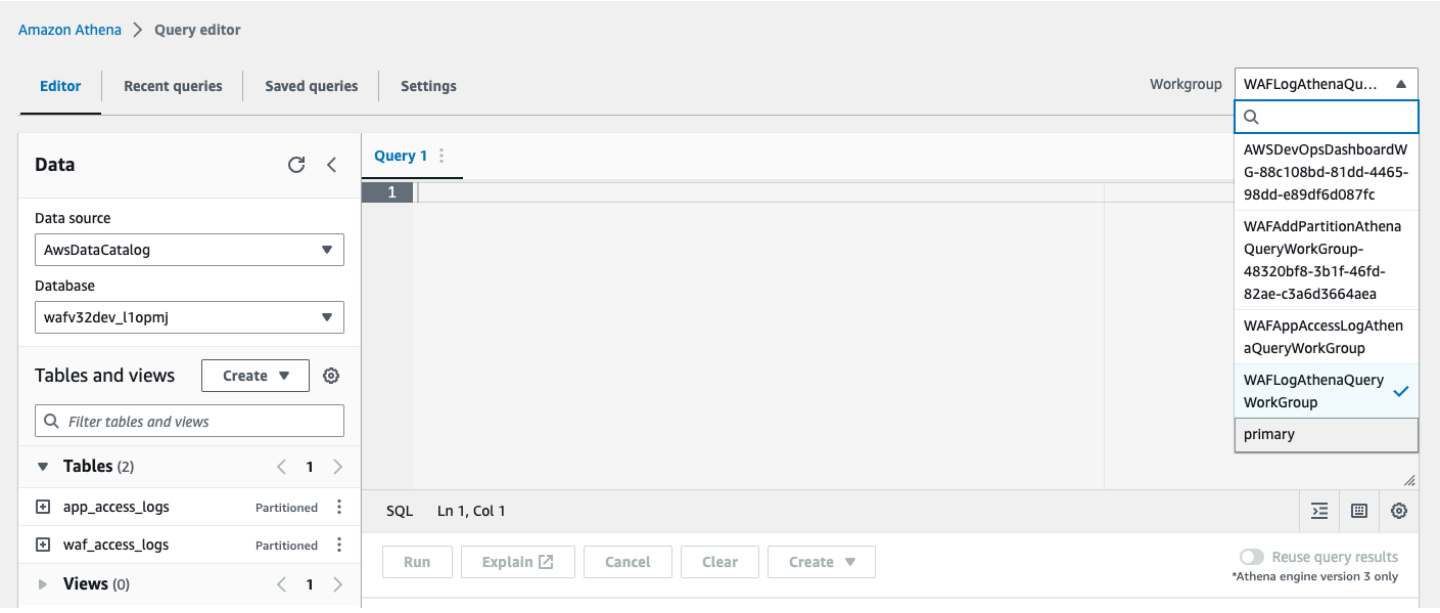
View WAF log queries

1. Sign in to the Amazon Athena console.
2. Choose **Launch query editor**.
3. Select the database for this solution.
4. Select **WAFLogAthenaQueryWorkGroup** from the dropdown list.

   > **Note**
   >
   > This workgroup exists only if you selected `Yes - Amazon Athena log parser` for the **Activate HTTP Flood Protection** template parameter.

5. Choose **Switch** to switch the workgroup.

Screenshot of Athena query editor showing no queries



1. Select the **History** tab.
2. Select and open SELECT queries from the list.

View application access log queries

1. Sign in to the Amazon Athena console.
2. Select the **Workgroup** tab.
3. Select **WAFAppAccessLogAthenaQueryWorkGroup** from the list.

> **Note**
>
> This workgroup exists only if you selected `Yes - Amazon Athena log parser` for the **Activate Scanner & Probe Protection** template parameter.

4. Choose **Switch workgroup**.
5. Select the **Recent queries** tab.
6. Select and open SELECT queries from the list.

View adding Athena partition queries

1. Sign in to the Amazon Athena console.

2. Select the **Workgroup** tab.
3. Select **WAFAddPartitionAthenaQueryWorkGroup** from the list.

> **Note**
>
> This workgroup exists only if you selected `Yes - Amazon Athena log parser` for the **Activate HTTP Flood Protection** and/or **Activate Scanner & Probe Protection** template parameter.

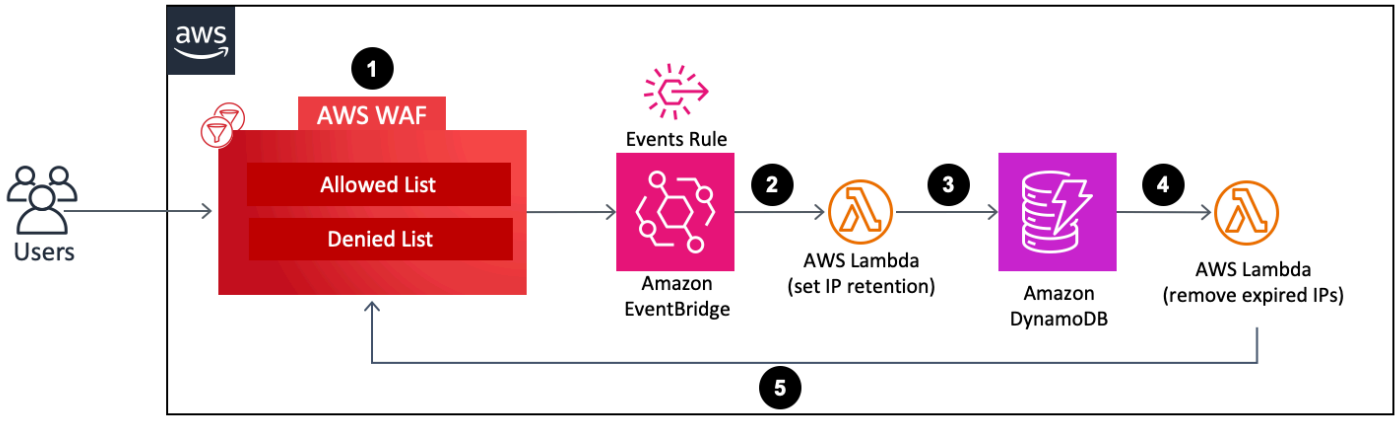
4. Select **Switch workgroup**.
5. Select the **History** tab.
6. Select and open `ALTER TABLE` queries from the list. These queries run every hour to add a new hourly partition to the Athena table.

Configure IP retention on Allowed and Denied AWS WAF IP sets

You can configure IP retention on Allowed and Denied AWS WAF IP sets that the solution creates. The following sections explain how it works and provide the steps to set it up.

How it works

Architecture diagram depicting the AWS WAF allowed and denied lists and other AWS resources



1. When a user updates (add or delete an IP address) the Allowed or Denied WAF IP set, this action invokes an AWS WAF `UpdateIPSet` API call and creates an event.

2. An Amazon EventBridge events rule detects the events based on a predefined event pattern, and invokes a Lambda function to set the retention period for all the IP addresses that exist in the IP set after the update.
3. The Lambda function processes the events, extracts relevant data to IP retention (such as IP set name, ID, scope, IP addresses), and inserts it into a DynamoDB table. It also inserts an `ExpirationTime` attribute for each DynamoDB item. The solution calculates the expiration time by adding a user-defined retention period to the event time. The table has DynamoDB Streams and Time to Live (TTL) turned on. The TTL attribute is `ExpirationTime`.
4. When an item reaches its expiration time, TTL is invoked and DynamoDB deletes the item from the table after its expiration time. Upon deletion of the item, the deleted item is added to the DynamoDB stream, which invokes a Lambda function for downstream processing.
5. The Lambda function obtains the information about the deleted item from the DynamoDB stream and makes an AWS WAF API call to remove the expired IP addresses included in the item from the target AWS WAF IP set.

Turn on IP retention

Follow these steps to turn on IP retention:

1. In the Cloudformation stack that you deploy or update, enter the **IP Retention Period (Minutes) for Allowed IP Set** and **IP Retention Period (Minutes) for Denied IP Set**. The minimum retention period is 15 minutes. The solution treats any number between `0` and 15 as 15. For more information about deployment configuration, refer to Step 1. Launch the stack.
2. Enter an email address if you want to receive an email notification when expired IP addresses are removed from the AWS WAF IP set. If you choose to receive an email notification, you must confirm subscription using the link in the email you receive after the solution successfully deploys. For more information about deployment configuration, refer to Step 1. Launch the stack.
3. Update the AWS WAF IP set by adding or deleting IP addresses. This initiates the IP retention process and creates an DynamoDB item, including an IP expiration list. This expiration list consists of IP addresses that exist in the AWS WAF IP set after your update it.
4. Once the DynamoDB item reaches its expiration time and is deleted from the table, the solution deletes the IP addresses included in the item's IP expiration list from the WAF IP set.

> **Note**
>
> Depending on the time when DynamoDB deletes an item expired by TTL, the actual delete operation of an expired IP address from the AWS WAF IP set can vary. DynamoDB TTL deletion mainly depends on the size and activity level of a table. Expect a delay in the AWS WAF delete operation because of the potential delay in the DynamoDB delete operation. In general, the solution deletes expired IP addresses from the AWS WAF IP set shortly after DynamoDB TTL deletion. For more information, refer to DynamoDB Time to Live (TTL) in the *Amazon DynamoDB Developer Guide*.

Build monitoring dashboard

AWS recommends that you configure a custom baseline monitoring system for each critical endpoint. For information on creating and using customized metric views, refer to CloudWatch Dashboards - Create & Use Customized Metrics Views and Using Amazon CloudWatch dashboards.

The following dashboard screenshot shows an example of a custom baseline monitoring system.

Screenshot of CloudFront dashboard

The dashboard displays the following metrics:

- **Allowed vs Blocked Requests** - Shows if you receive a surge in allowed access (twice the normal peak access) or blocked access (any period that identifies more than 1K blocked requests). CloudWatch sends an alert to a Slack channel. You can use this metric to track known DDoS attacks (when blocked requests increase) or a new version of an attack (when the requests are allowed to access the system).

> **Note**
>
> **Note:** The solution provides this metric.

- **BytesDownloaded vs Uploaded** - Helps identify when a DDoS attack targets a service that normally doesn't receive a large amount of access to exhaust resources (for example, search engine component sending MBs of information for one specific request parameters set).
- **ELB Spillover and Queue length** - Helps verify if a DDoS attack is causing damage to the infrastructure and the attacker is bypassing CloudFront or the AWS WAF layer, and attacking directly unprotected resources.
- **ELB Request Count** - Helps identify damage to the infrastructure. This metric shows if the attacker is bypassing the protection layer, or if you should review a CloudFront cache rule to increase the cache hit rate.
- **ELB Healthy Host** - You can use this as another system health check metric.
- **ASG CPU Utilization** - Helps identify if the attacker is bypassing CloudFront, AWS WAF, and Elastic Load Balancing. You can also use this metric to identify the damage of an attack.

Handle XSS false positives

This solution configures an AWS WAF rule that inspects commonly explored elements of incoming requests to identify and block XSS attacks. This detection pattern is less effective if your workload allows legitimate users to compose and submit HTML, for example, using a rich text editor in a content management system. In this scenario, consider creating an exception rule that bypasses the default XSS rule for specific URL patterns that accept rich text input, and implement alternate mechanisms to protect those excluded URLs.

Additionally, some image or custom data formats can cause false positives because they contain patterns indicating a potential XSS attack in HTML content. For example, an SVG file might contain a `<script>` tag. If you expect this type of content from legitimate users, narrowly tailor your XSS rules to allow HTML requests that include these other data formats.

Complete the following steps to update XSS rule to exclude URLs that accept HTML as input. Refer to the Amazon WAF Developer Guide for detailed instructions.

1. Sign in to the AWS WAF console.
2. Create a string match or regex condition.
3. Configure the filter settings to inspect URI and list values that you want to accept against the XSS rule.
4. Edit this solution's **XSS Rule** and add the new condition that you created.

   For example, to exclude all URLs in the list, choose the following for **When a request** :

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

Troubleshooting

If you need help with this solution, contact Support to open a support case for this solution.

Contact AWS Support

If you have AWS Business Support+, AWS Enterprise Support, or Unified Operations, you can use the AWS Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Open Support Center.
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Security Automations for AWS WAF**.
4. For **Severity**, the option that best matches your use case.
5. As you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail, including the name of this solution and the version you are using, such as this example: **Security Automations for AWS WAF vX.Y.Z**.
3. Choose **Attach files**.
4. Attach the information that Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can't resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Developer guide

This section provides the source code for the solution.

Source code

Visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

This solution's templates are generated using the AWS CDK. Refer to the README.md file for additional information.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution, pointers to related resources, and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When turned on, the solution collects the following information is collected and sends it to AWS during initial deployment of the CloudFormation template:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each deployment of this solution
- **Timestamp** - Data-collection timestamp
- **Solution configuration** - Features turned on and parameters set during initial launch
- **Lifecycle** - How long the customer used this solution (based on stack delete)
- **Log parser data**:
  - The number of IP addresses in the **Scanner & Probe** IP set, **Bad Bot** IP set and the **HTTP Flood** IP set to block
  - The number of requests processed and blocked
- **IP lists parser data**:
  - The number of IP addresses in the **Reputation Lists** IP set
  - The number of requests processed and blocked
- **IP retention data** - The number of expired IP addresses being removed from the **Allowed** or **Denied** IP set

AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the `aws-waf-security-automations.template` AWS CloudFormation to your local hard drive.

2. Open the CloudFormation template with a text editor.
3. Modify the CloudFormation template mapping section from:

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

   to:

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

4. Sign in the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Step 1. Launch the stack.

Related resources

Associated AWS whitepapers

- AWS Best Practices for DDoS Resiliency

Associated AWS Security Blog posts

- How to Prevent Hotlinking by Using AWS WAF, Amazon CloudFront, and Referer Checking

Third-Party IP Reputation Lists

- Spamhaus DROP List website
- Proofpoint Emerging Threats IP list
- Tor exit node list

Contributors

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

Revisions

Visit the CHANGELOG.md in our GitHub repository to track version-specific improvements and fixes.

Notices

This implementation guide is provided for informational purposes only. It represents current AWS product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers, or licensors. 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.

The Security Automations for AWS WAF solution is licensed under the terms of the Apache License Version 2.0.