AWS WAF for Bot control and Fraud prevention

Harith Gaddamanugu

Sr. Solutions Architect- Edge Services
Amazon Web Services

Mun Hossain

Principal Networking & Security Specialist
Amazon Web Services

Agenda

AWS WAF

AWS Bot Control

AWS Fraud Control

Client side Integration

Best Practices

Call to Action

Expanding threat landscape….

Denial of Service

- SYN Floods
- Reflection Attacks
- Web Request Floods

App Vulnerabilities

- SQL Injection
- Cross-site Scripting (XSS)
- OWASP Top 10
- Common Vulnerabilities and Exposures (CVE)

Good Bots

- Search Engine Crawling
- Website Health Monitoring
- Vulnerability Scanning

Bad Bots

- Comment Spam
- SEO Spam
- Fraud
- Account Takeover

AWS WAF

AWS WAF is a web application firewall that protects against common web exploits and bots that may affect availability, compromise security ,or consume excessive resources.

AWS WAF

Elastic Load Balancer

Amazon CloudFront

Amazon API Gateway

AWS AppSync

Amazon Cognito

AWS WAF

Multiple Rules & Condition types
Security Automations
AWS WAF Managed Rules
Third Party Managed Rules
Rate limiting rules
Custom Response Codes
Custom Header Insertion
Labels

Flexible Rule Configuration

AWS WAF

Intelligent Threat Mitigation

AWS Bot Control
Scope Down Statements
Account Take Over Protection (ATP)
Mobile and Java SDK Integration
CAPTCHA Integration
Challenge Action Integration

Monitoring

Logs

Alarms

Events

Amazon CloudWatch

AWS CloudTrail

AWS Config

Logging

Amazon S3

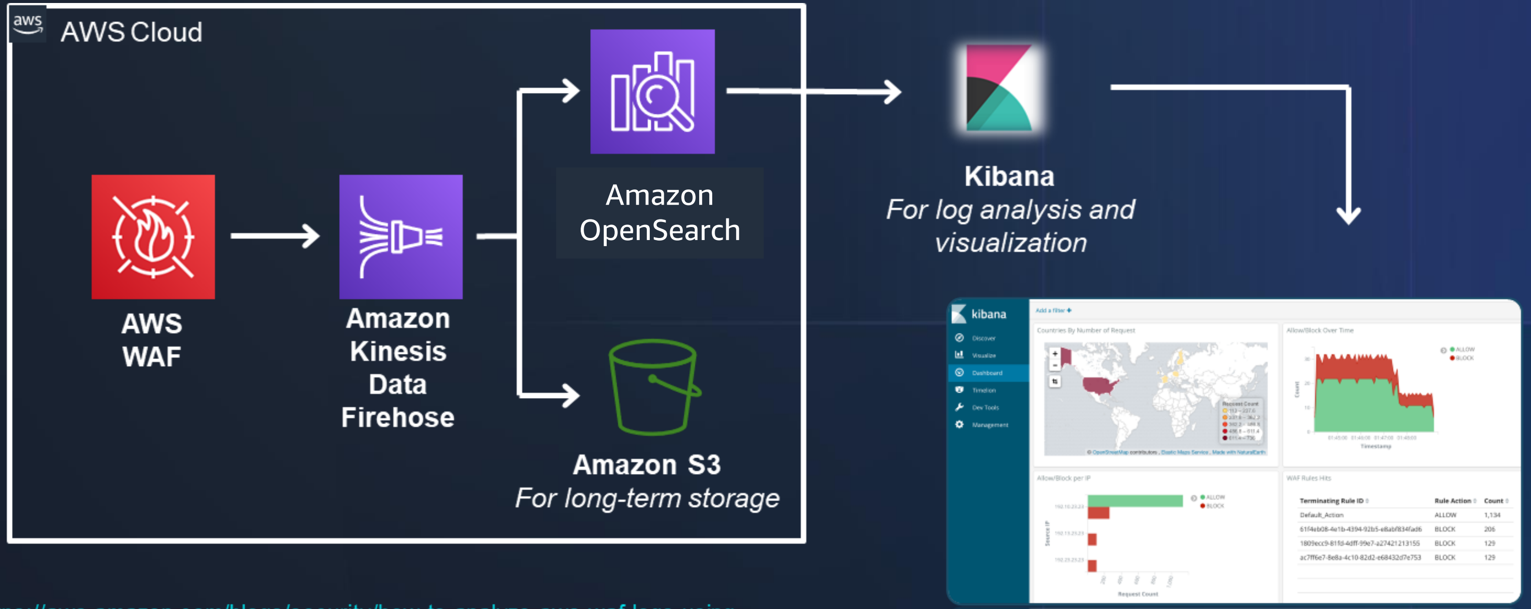
Amazon CloudWatch Logs

Amazon Redshift

Amazon OpenSearch

Splunk

Analytics



AWS Cloud

AWS WAF

Amazon Kinesis Data Firehose

Amazon OpenSearch

Amazon S3
For long-term storage

Kibana
For log analysis and visualization

WAF Rules Hits

Terminating Rule ID	Rule Action	Count
Default_Action	ALLOW	1,134
61f4eb08-4e1b-4394-92b5-e8abf834fad6	BLOCK	206
1809ecc9-81fd-4dff-99e7-a27421213155	BLOCK	129
ac7ff6e7-8e8a-4c10-82d2-e68432d7e753	BLOCK	129

https://aws.amazon.com/blogs/security/how-to-analyze-aws-waf-logs-using-amazon-elasticsearch-service/

Insights

Amazon CloudWatch

Amazon Athena

AWS WAF Bot Control

What is a bot ?

An internet bot, web robot, robot or simply bot, is a software application that runs automated tasks (scripts) over the internet.

Types of bots ?

Broadly divided into two types

Good Bot

Bad Bot

Understanding impact of bot attacks

Infrastructure attacks

Application attacks

How AWS WAF can help?

Common Bots

- Detects self-identifying, simple bots
- Signature-based detection
- IP Reputation lists
- Request headers
- Reverse DNS lookup

Targeted Bots

- Detects bots that try to evade detection
- Behavior-based detection capabilities
- Browser Fingerprinting and Interrogation
- Dynamic low-latency rate-limiting
- ML-based behavior analysis

Common Bot Control

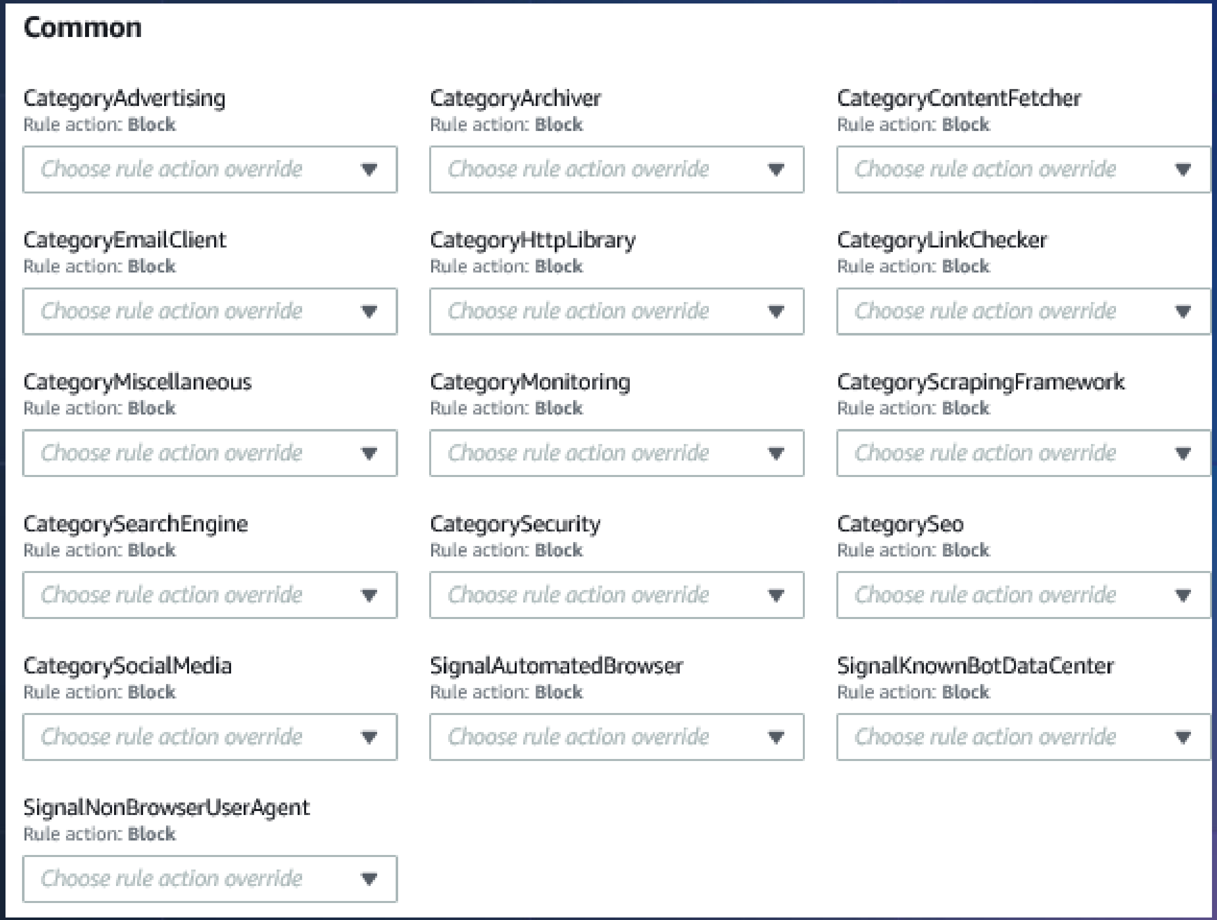
Ideal for generic bot problems such as scanners and crawlers

Exposed as Bot Control Amazon Managed ruleset (AMR)

Categorizing and verifying common bots e.g.

bot:category:search_engine such as wpscan

bot:category:social_media such as redditbot



Common

CategoryAdvertising
Rule action: **Block**
Choose rule action override ▼

CategoryArchiver
Rule action: **Block**
Choose rule action override ▼

CategoryContentFetcher
Rule action: **Block**
Choose rule action override ▼

CategoryEmailClient
Rule action: **Block**
Choose rule action override ▼

CategoryHttpLibrary
Rule action: **Block**
Choose rule action override ▼

CategoryLinkChecker
Rule action: **Block**
Choose rule action override ▼

CategoryMiscellaneous
Rule action: **Block**
Choose rule action override ▼

CategoryMonitoring
Rule action: **Block**
Choose rule action override ▼

CategoryScrapingFramework
Rule action: **Block**
Choose rule action override ▼

CategorySearchEngine
Rule action: **Block**
Choose rule action override ▼

CategorySecurity
Rule action: **Block**
Choose rule action override ▼

CategorySeo
Rule action: **Block**
Choose rule action override ▼

CategorySocialMedia
Rule action: **Block**
Choose rule action override ▼

SignalAutomatedBrowser
Rule action: **Block**
Choose rule action override ▼

SignalKnownBotDataCenter
Rule action: **Block**
Choose rule action override ▼

SignalNonBrowserUserAgent
Rule action: **Block**
Choose rule action override ▼

Targeted Bot Control

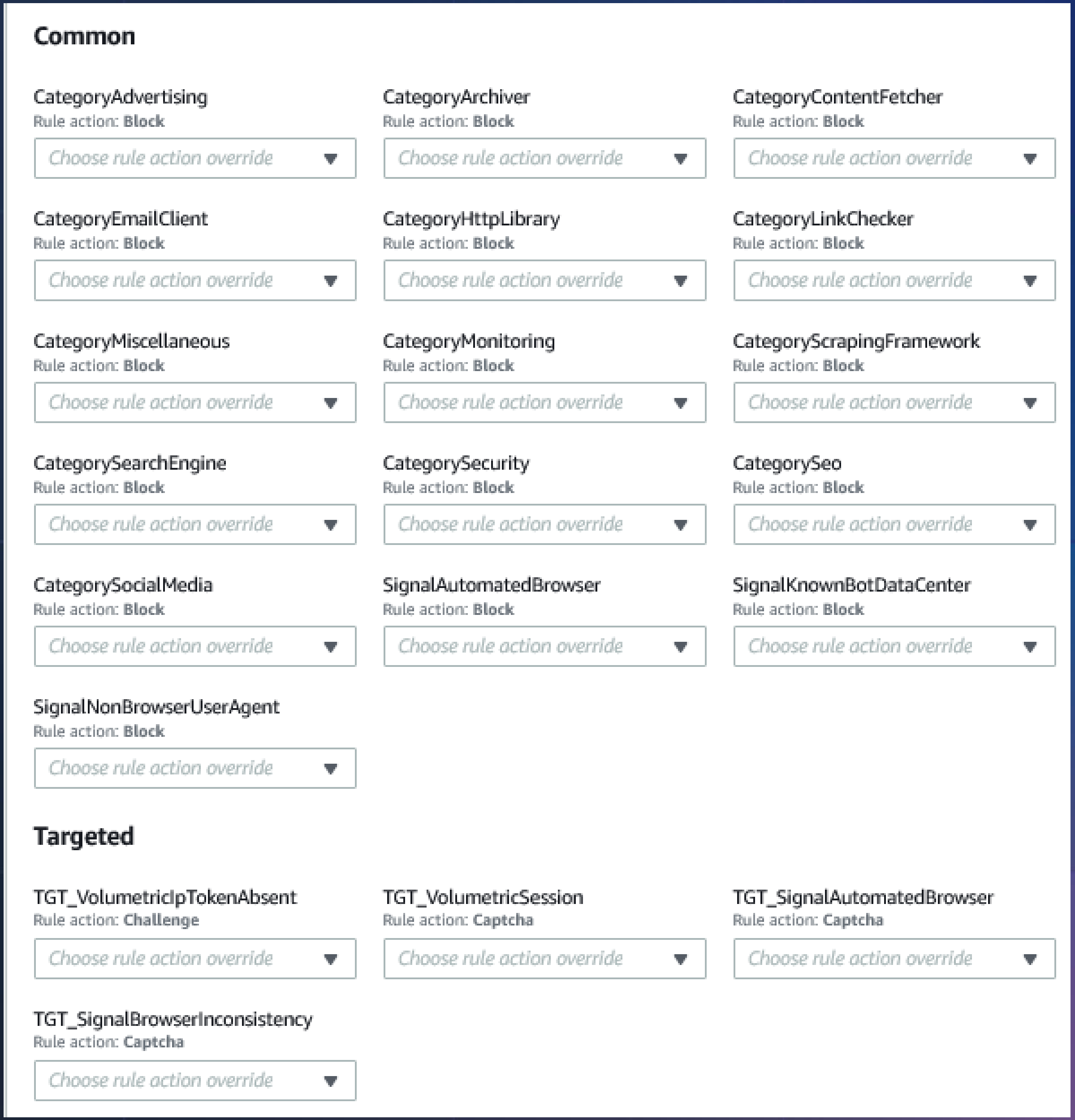
Allows tracking users across sessions and Ips

Automatically detects attack vectors across devices and allows blocking the patterns rather than individual fields like IP address

Done by generating cryptographic tokens by a process known as Challenge Interstitial (exposed as Challenge rule action on the console)

Challenge rule action available for all WAF rules including custom rules

Supports client-side embedding with Application Integration SDK



Common

CategoryAdvertising
Rule action: **Block**
Choose rule action override ▼

CategoryArchiver
Rule action: **Block**
Choose rule action override ▼

CategoryContentFetcher
Rule action: **Block**
Choose rule action override ▼

CategoryEmailClient
Rule action: **Block**
Choose rule action override ▼

CategoryHttpLibrary
Rule action: **Block**
Choose rule action override ▼

CategoryLinkChecker
Rule action: **Block**
Choose rule action override ▼

CategoryMiscellaneous
Rule action: **Block**
Choose rule action override ▼

CategoryMonitoring
Rule action: **Block**
Choose rule action override ▼

CategoryScrapingFramework
Rule action: **Block**
Choose rule action override ▼

CategorySearchEngine
Rule action: **Block**
Choose rule action override ▼

CategorySecurity
Rule action: **Block**
Choose rule action override ▼

CategorySeo
Rule action: **Block**
Choose rule action override ▼

CategorySocialMedia
Rule action: **Block**
Choose rule action override ▼

SignalAutomatedBrowser
Rule action: **Block**
Choose rule action override ▼

SignalKnownBotDataCenter
Rule action: **Block**
Choose rule action override ▼

SignalNonBrowserUserAgent
Rule action: **Block**
Choose rule action override ▼

Targeted

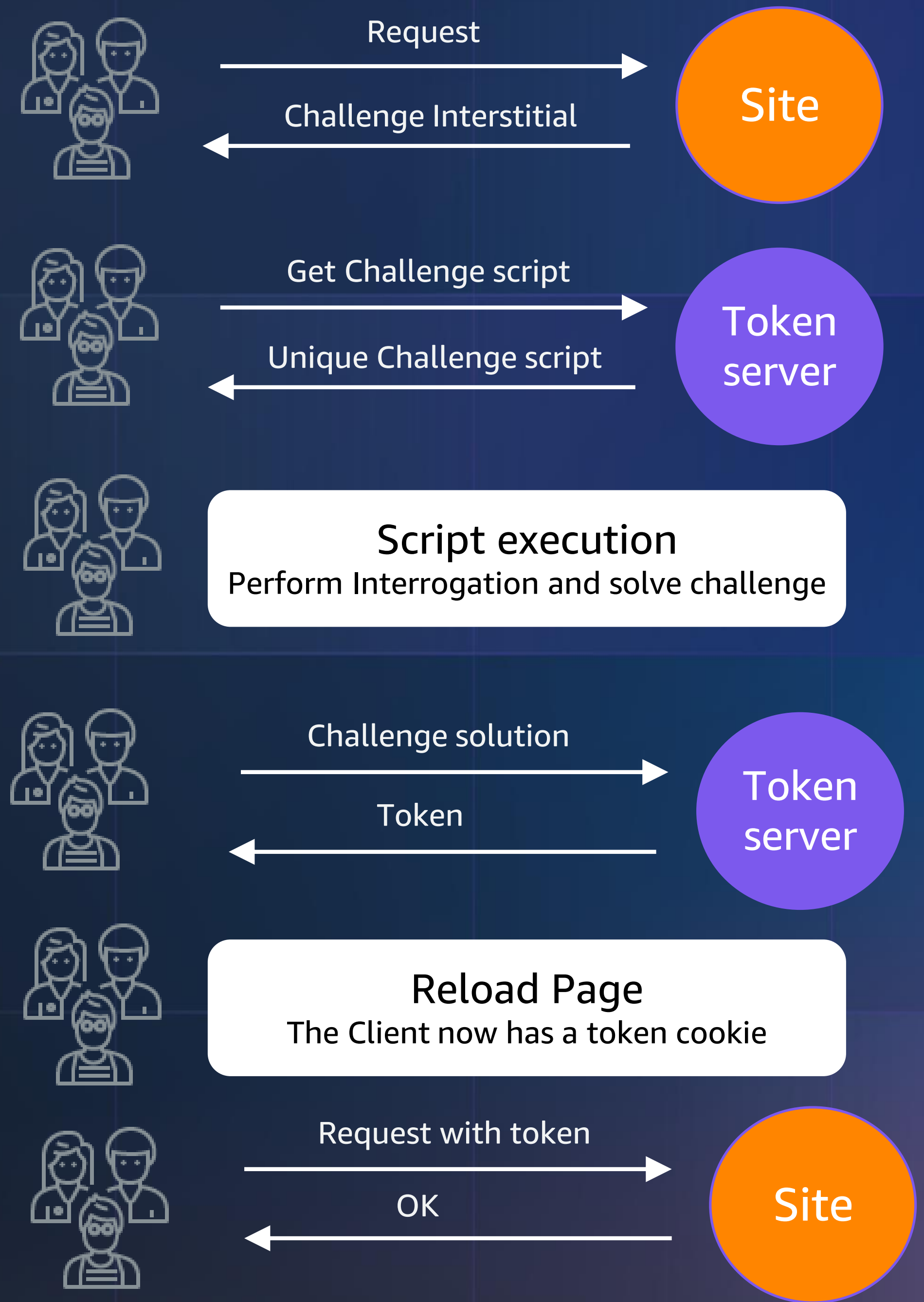
TGT_VolumetricIpTokenAbsent
Rule action: **Challenge**
Choose rule action override ▼

TGT_VolumetricSession
Rule action: **Captcha**
Choose rule action override ▼

TGT_SignalAutomatedBrowser
Rule action: **Captcha**
Choose rule action override ▼

TGT_SignalBrowserInconsistency
Rule action: **Captcha**
Choose rule action override ▼

Targeted Bots - Challenge Rule Action



Request

Challenge Interstitial

Site

Get Challenge script

Unique Challenge script

Token server

Script execution

Perform Interrogation and solve challenge

Challenge solution

Token

Token server

Reload Page

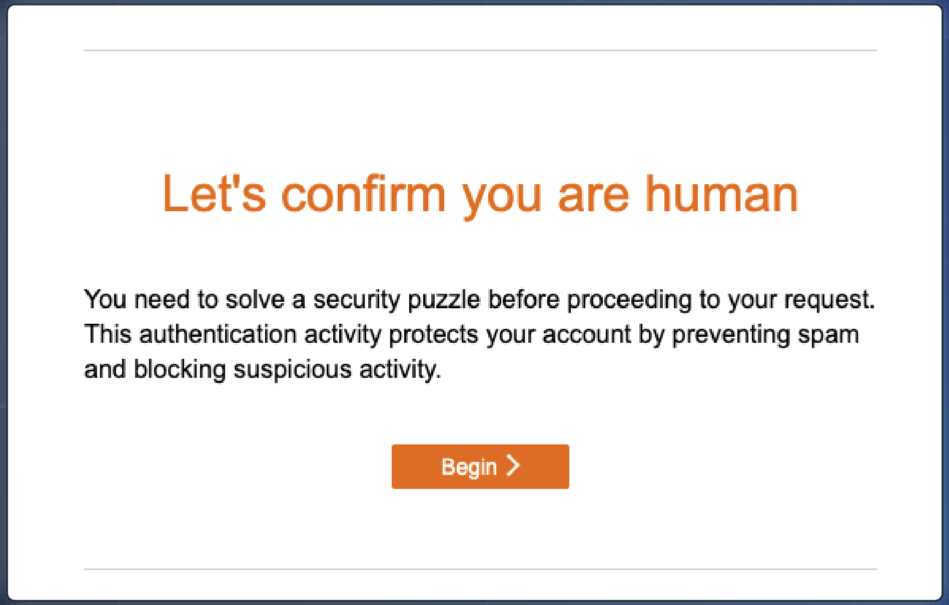
The Client now has a token cookie

Request with token

OK

Site

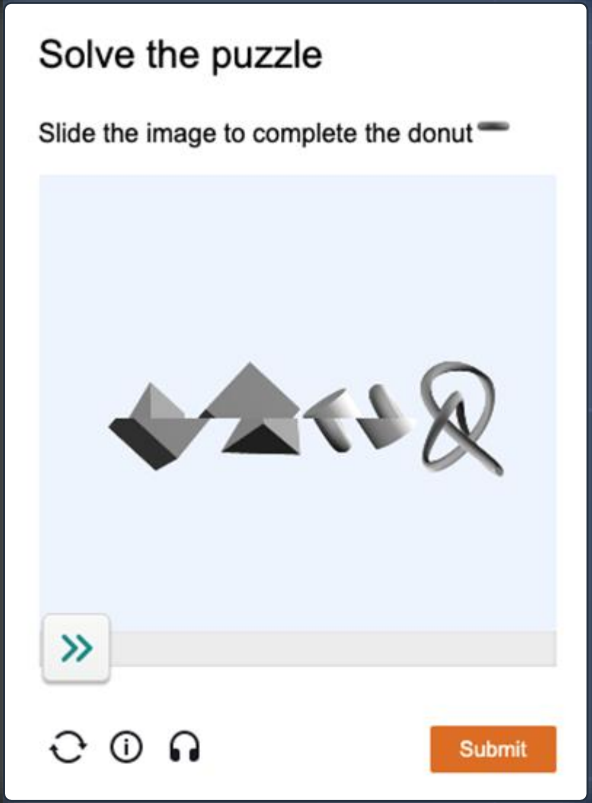
Targeted Bots - CAPTCHA Rule Action



Let's confirm you are human

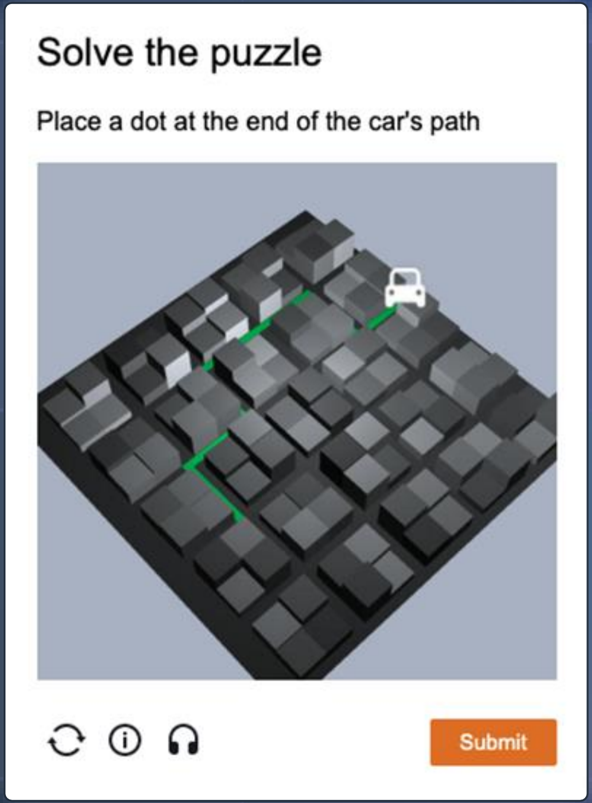
You need to solve a security puzzle before proceeding to your request. This authentication activity protects your account by preventing spam and blocking suspicious activity.

Targeted Bots - CAPTCHA Puzzle options



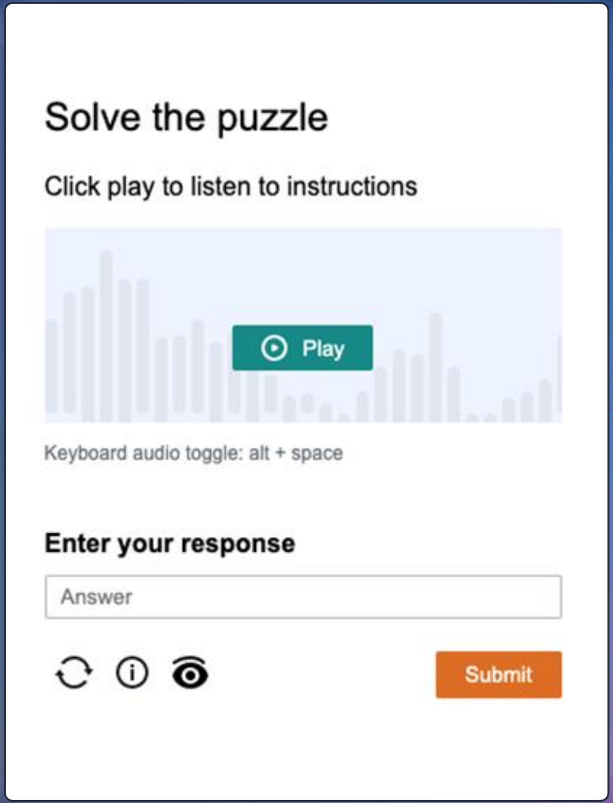
Solve the puzzle

Slide the image to complete the donut



Solve the puzzle

Place a dot at the end of the car's path



Solve the puzzle

Click play to listen to instructions

Keyboard audio toggle: alt + space

Enter your response

AWS Fraud Control

Some AWS Fraud Protection solutions

AWS WAF

Amazon Fraud Detector

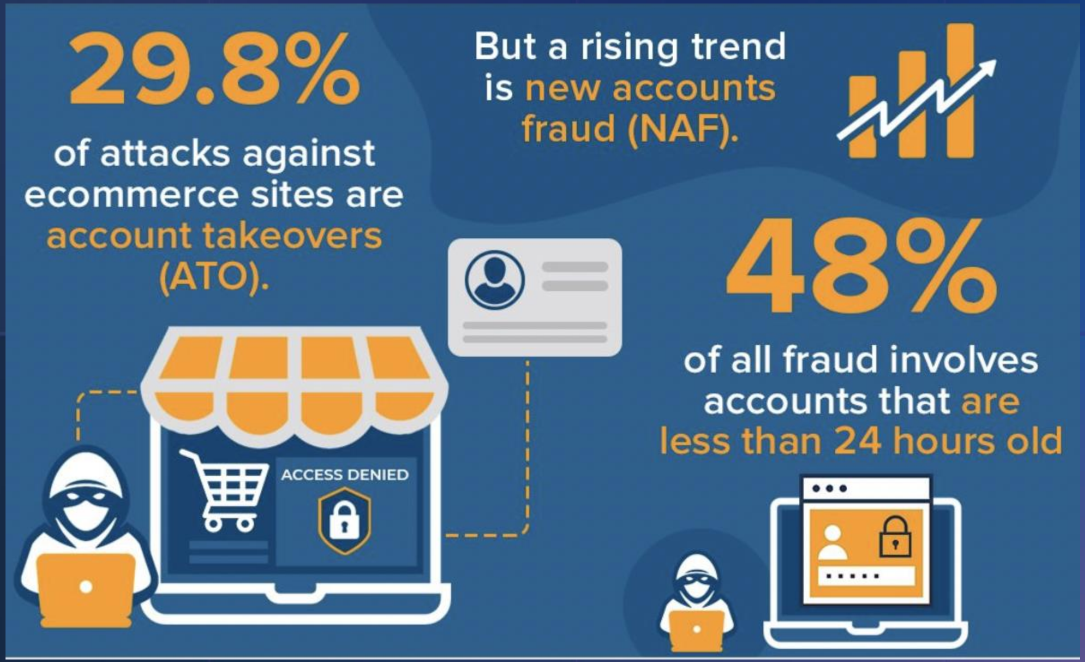
What are Account Take Over attacks?

Credential stuffing attacks:

- Attacker uses a collection of stolen login credentials and uses a botnet to try all combinations
- Assumes the reuse of breached credentials on multiple sites

Brute force password attacks:

- Attacker uses passwords found in cracking dictionaries over and over again
- Relies on guessing passwords at a high volume



29.8% of attacks against ecommerce sites are account takeovers (ATO).

But a rising trend is new accounts fraud (NAF).

48% of all fraud involves accounts that are less than 24 hours old

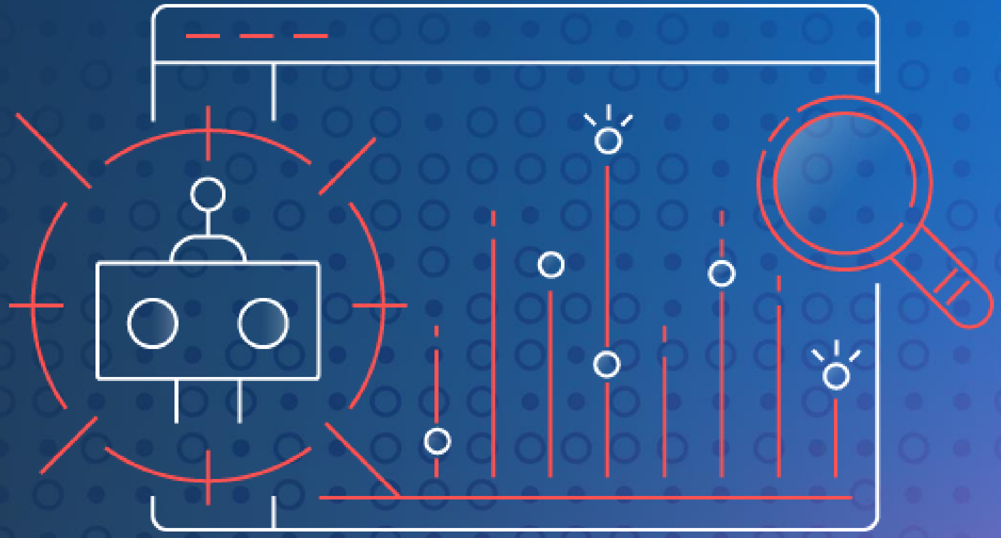
*Forbes report

Account Takeover Prevention (ATP) Managed Rule Group

Detect and mitigate against login attempts made by malicious actors using stolen credentials

- Monitor and verify credentials
- Check to see if user’s credentials have been leaked on dark web

- Prevent automated login attempts from bots
- Stop attacks like brute-force and credential stuffing



Can also inspects your application's responses to login attempts, to track success and failure rates

Account Takeover Prevention (ATP)

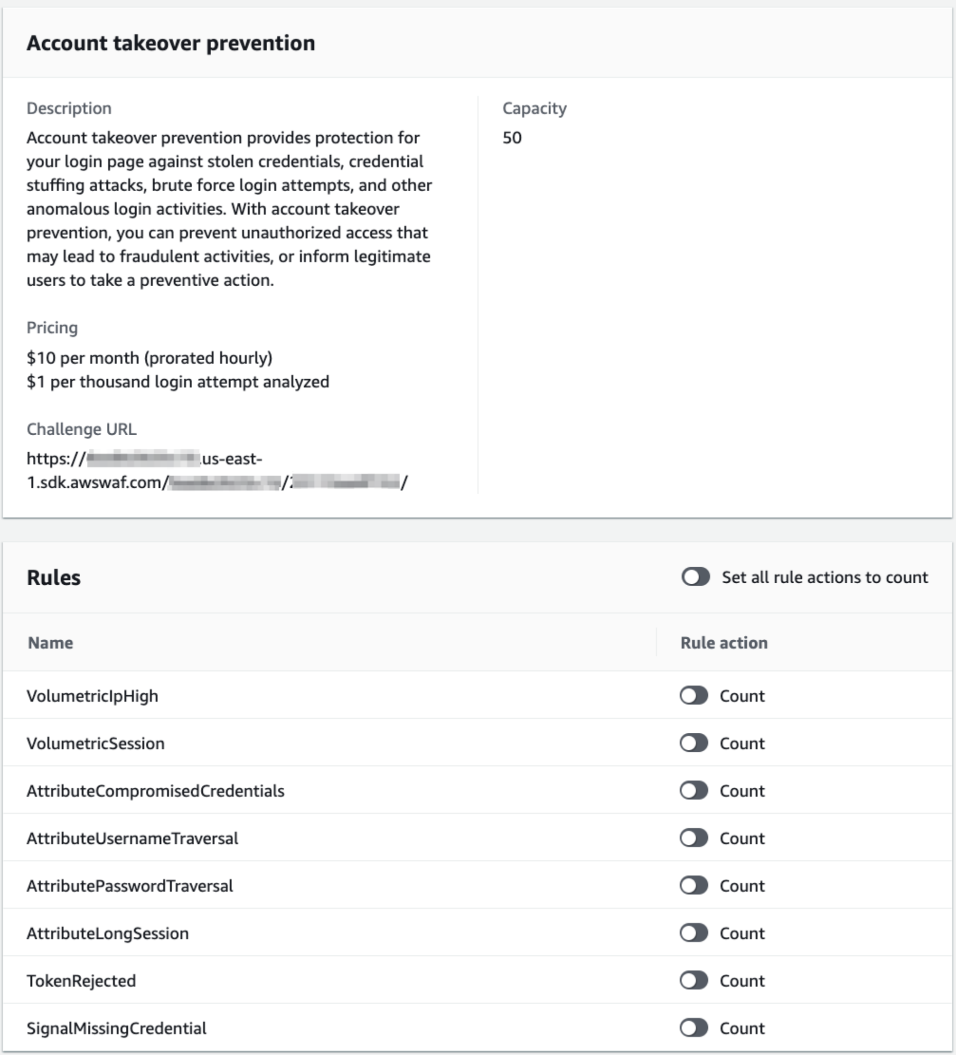
A Set of login-oriented controls

SDK for active challenges for mobile and web

Credential lookup in a stolen credential DB for each login

Detects brute force and bot access

Checks Response codes and signals



Account takeover prevention

Description

Account takeover prevention provides protection for your login page against stolen credentials, credential stuffing attacks, brute force login attempts, and other anomalous login activities. With account takeover prevention, you can prevent unauthorized access that may lead to fraudulent activities, or inform legitimate users to take a preventive action.

Pricing

$10 per month (prorated hourly)
$1 per thousand login attempt analyzed

Challenge URL

https://[redacted].us-east-1.sdk.awswaf.com/[redacted]/[redacted]/

Capacity

50

Rules

Set all rule actions to count

Name	Rule action
VolumetricIpHigh	Count
VolumetricSession	Count
AttributeCompromisedCredentials	Count
AttributeUsernameTraversal	Count
AttributePasswordTraversal	Count
AttributeLongSession	Count
TokenRejected	Count
SignalMissingCredential	Count

Account Takeover Prevention (ATP)

Additional configuration needs to be provided

Required Parameters	Description
Login Path	Your application’s sign-in page URI
Payload Type	JSON or Form encoded body
Username Field	Where username is located within body
Password Field	Where password is located within body

Client Side Integration

Client Side Integration

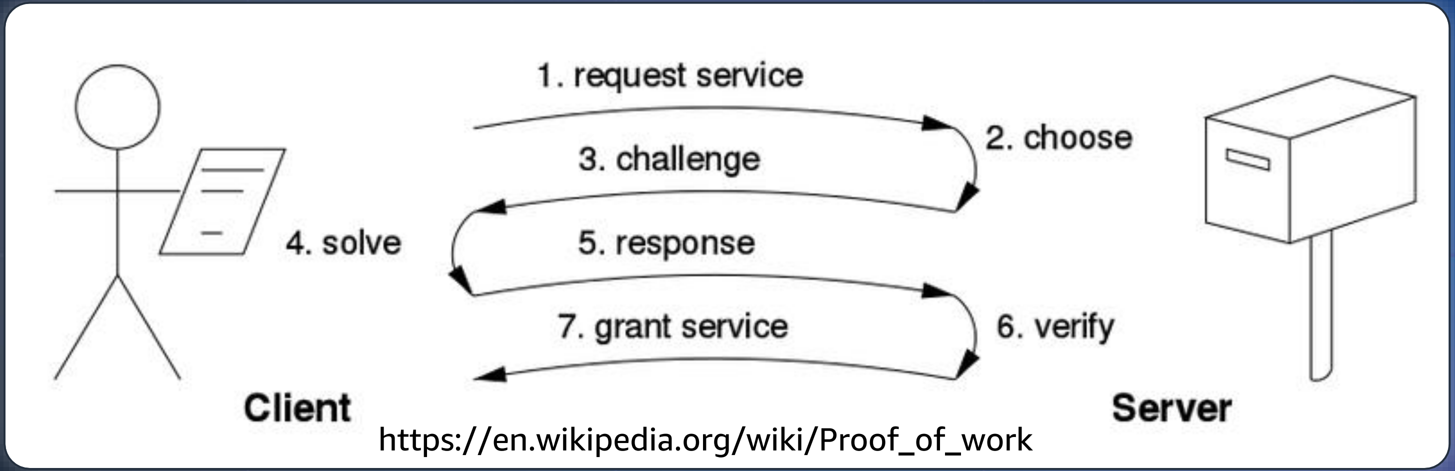
Provides enhanced detection against bots and automations

Silent challenge (proof-of-work)

Gathers additional telemetry

(Optional)

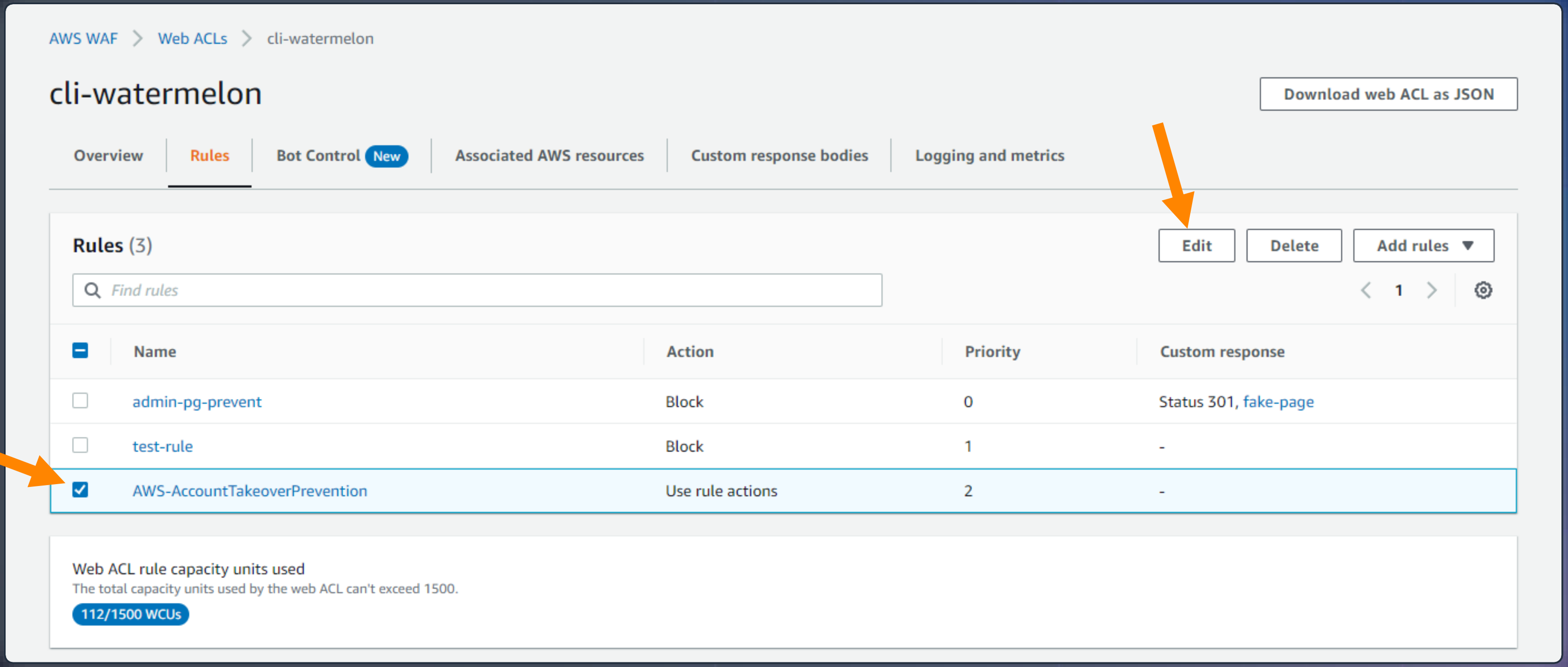
Similar to Challenge Action



https://en.wikipedia.org/wiki/Proof_of_work

Example ATP JavaScript SDK Setup

After adding ATP to your web ACL, navigate to the detail page.



AWS WAF > Web ACLs > cli-watermelon

cli-watermelon

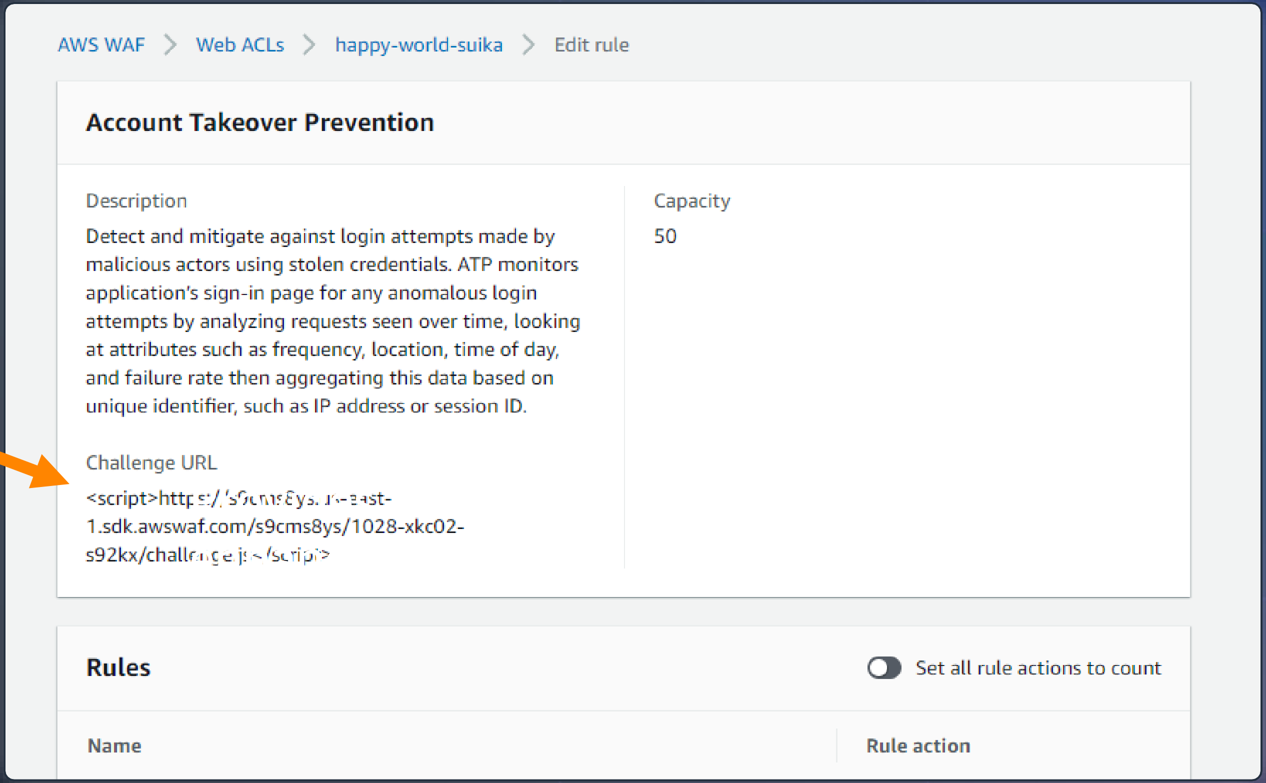
Rules (3)

☐	Name	Action	Priority	Custom response
☐	admin-pg-prevent	Block	0	Status 301, fake-page
☐	test-rule	Block	1	-
☑	AWS-AccountTakeoverPrevention	Use rule actions	2	-

Web ACL rule capacity units used
The total capacity units used by the web ACL can't exceed 1500.
112/1500 WCUs

Example ATP JavaScript SDK Setup

2. Copy the unique Challenge URL for your web ACL.



AWS WAF > Web ACLs > happy-world-suika > Edit rule

Account Takeover Prevention

Description

Detect and mitigate against login attempts made by malicious actors using stolen credentials. ATP monitors application's sign-in page for any anomalous login attempts by analyzing requests seen over time, looking at attributes such as frequency, location, time of day, and failure rate then aggregating this data based on unique identifier, such as IP address or session ID.

Capacity

50

Challenge URL

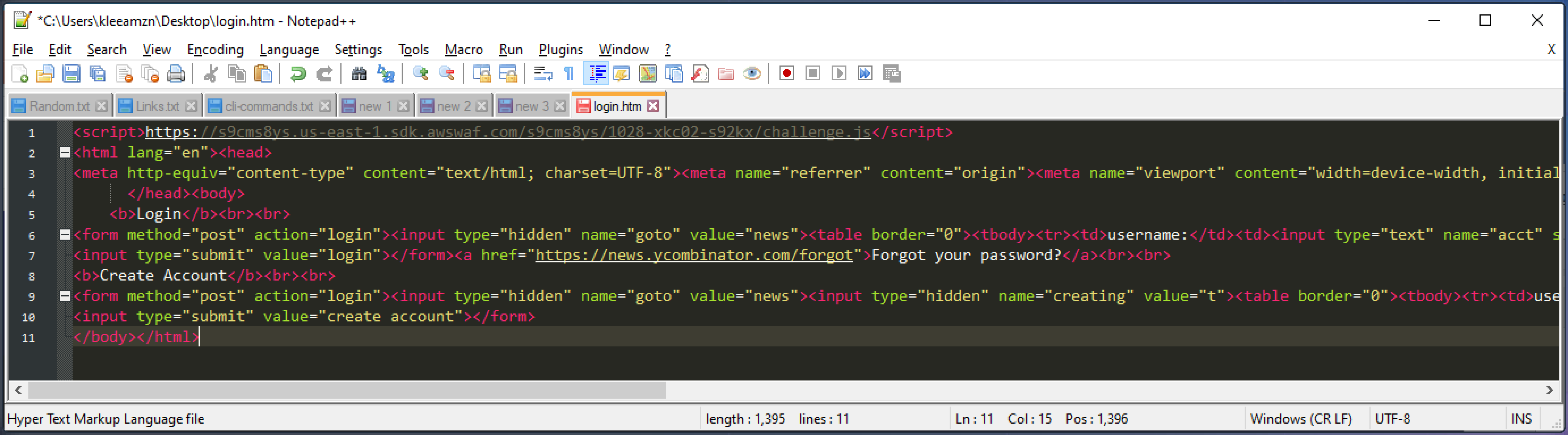
```
<script>https://s9cms8ys.us-east-
1.sdk.awswaf.com/s9cms8ys/1028-xkc02-
s92kx/challenge.js</script>
```

Rules

Name	Rule action

Example ATP JavaScript SDK Setup

3. Paste this challenge URL into your sign-in page.


*C:\Users\kleeamzn\Desktop\login.htm - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Random.txt
Links.txt
cli-commands.txt
new 1
new 2
new 3
login.htm
Hyper Text Markup Language file
length : 1,395 lines : 11
Ln : 11 Col : 15 Pos : 1,396
Windows (CR LF)
UTF-8
INS

```html
<script>https://s9cms8ys.us-east-1.sdk.awswaf.com/s9cms8ys/1028-xkc02-s92kx/challenge.js</script>
<html lang="en"><head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"><meta name="referrer" content="origin"><meta name="viewport" content="width=device-width, initial
      </head><body>
    <b>Login</b><br><br>
<form method="post" action="login"><input type="hidden" name="goto" value="news"><table border="0"><tbody><tr><td>username:</td><td><input type="text" name="acct" s
<input type="submit" value="login"></form><a href="https://news.ycombinator.com/forgot">Forgot your password?</a><br><br>
<b>Create Account</b><br><br>
<form method="post" action="login"><input type="hidden" name="goto" value="news"><input type="hidden" name="creating" value="t"><table border="0"><tbody><tr><td>use
<input type="submit" value="create account"></form>
</body></html>
```

Best Practices with Intelligent threat mitigation

Few best practices for Intelligent threat mitigation

- ✓ Implement the JavaScript and mobile application integration SDKs
- ✓ Using Scope Down Statements
- ✓ Reject requests with arbitrary host specifications
- ✓ Using Shield Advanced Automitigation for L7 DDoS Attacks
- ✓ Proper testing and understanding the right solution
- ✓ Fine tuning token immunity times

Managing False positives

Managing False positives

1. Allow Listing trusted requests

2. Use Labels as needed

3. Scope Down statements

4. Changes in existing request flow

5. Emerging user behavior changes

Configuring for cost optimization

“Limit the requests that you send to the ATP and Bot Control rule groups”

Exclude requests using scope down statements

Exclude requests by stacking other rules before

If you are using both ATP & Bot Control- Run bot control first

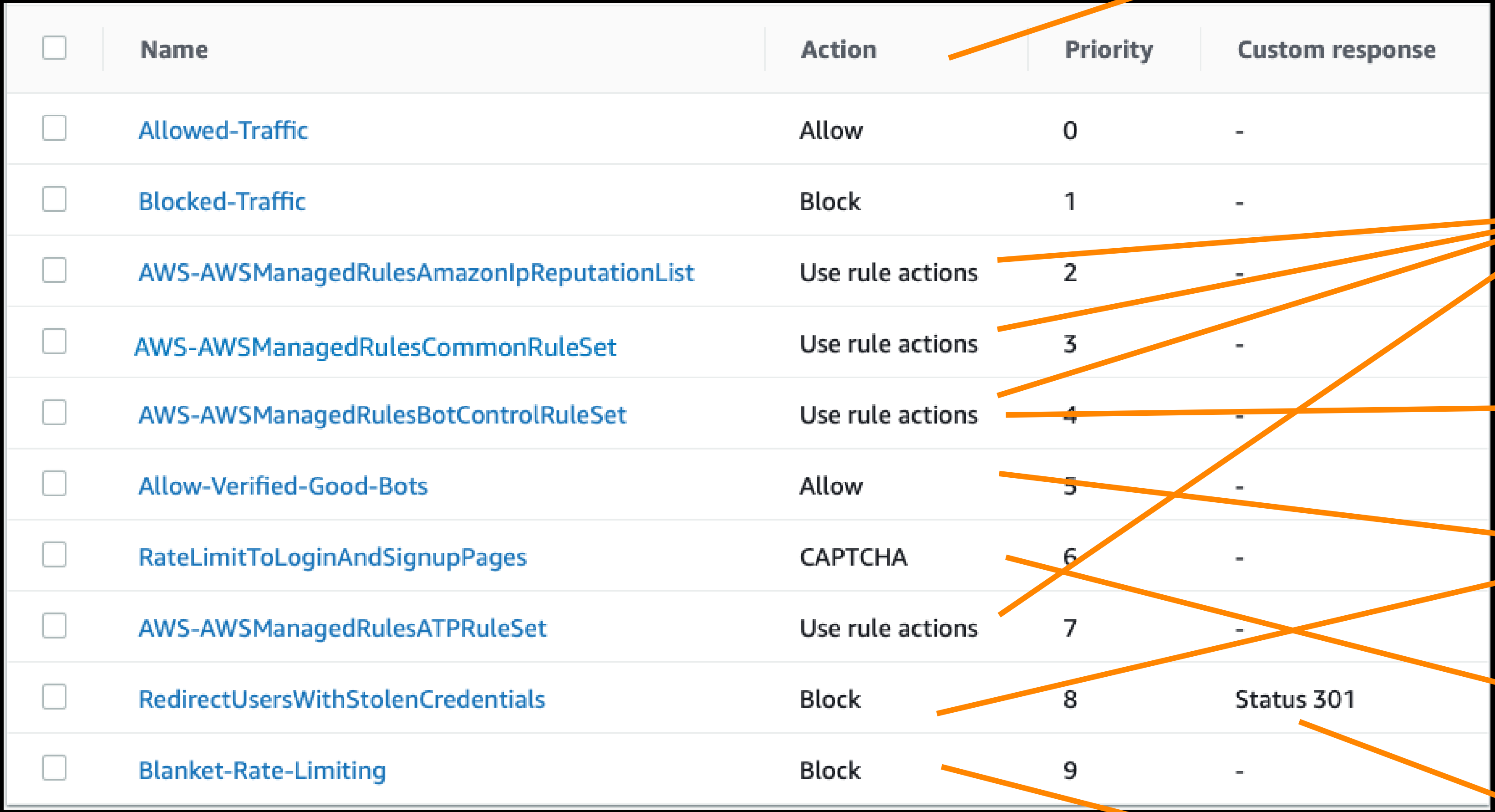
Adding a rate based rule to prevent abuse inspection fee abuse

Fine tune Token immunity time

Using IP Reputation Lists

Example Webacl Configuration

Web App Security WebACL Example



□	Name	Action	Priority	Custom response
□	Allowed-Traffic	Allow	0	-
□	Blocked-Traffic	Block	1	-
□	AWS-AWSManagedRulesAmazonIpReputationList	Use rule actions	2	-
□	AWS-AWSManagedRulesCommonRuleSet	Use rule actions	3	-
□	AWS-AWSManagedRulesBotControlRuleSet	Use rule actions	4	-
□	Allow-Verified-Good-Bots	Allow	5	-
□	RateLimitToLoginAndSignupPages	CAPTCHA	6	-
□	AWS-AWSManagedRulesATPRuleSet	Use rule actions	7	-
□	RedirectUsersWithStolenCredentials	Block	8	Status 301
□	Blanket-Rate-Limiting	Block	9	-

Reusable rule group. Contains IP CIDR blocks and countries - should be automatically populated through **automations**. Uses XFF as well. Set to “Block”

Set to “Count” (if you wish)

Use “**Scope Down**” to control costs and reduce logging noise

Label match rule

CAPTCHA when rate exceeded, Also helps to **prevent abuse** of ATP AMR

Redirect to stolen password

Protect **Availability** of the application

Call to action

1 Visit the AWS WAF Product Page at: https://aws.amazon.com/waf/

2 Getting Started Guide: https://aws.amazon.com/getting-started/?sc_icontent=awssm-evergreen-getting_started&sc_iplace=2up&trk=ha_awssm-evergreen-getting_started&sc_ichannel=ha&sc_icampaign=evergreen-getting_started

3 **WAF- Proof of Concept**

AWS WAF

Thank you!

Harith Gaddamanugu
harithg@amazon.com

Mun Hossain
munahoss@amazon.com