AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

SEC313-R2

Protecting against automated bots with AWS WAF Bot Control

Kevin Lee
Sr. Product Manager (Technical)
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Learn about different types of bots and how they can affect your business

Get insight into how you can mitigate against common and pervasive bots using AWS Web Application Firewall (WAF) Bot Control

Dive deep into advance use cases for Bot Control

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s talk about bots

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Real-world scenarios



E-commerce operation



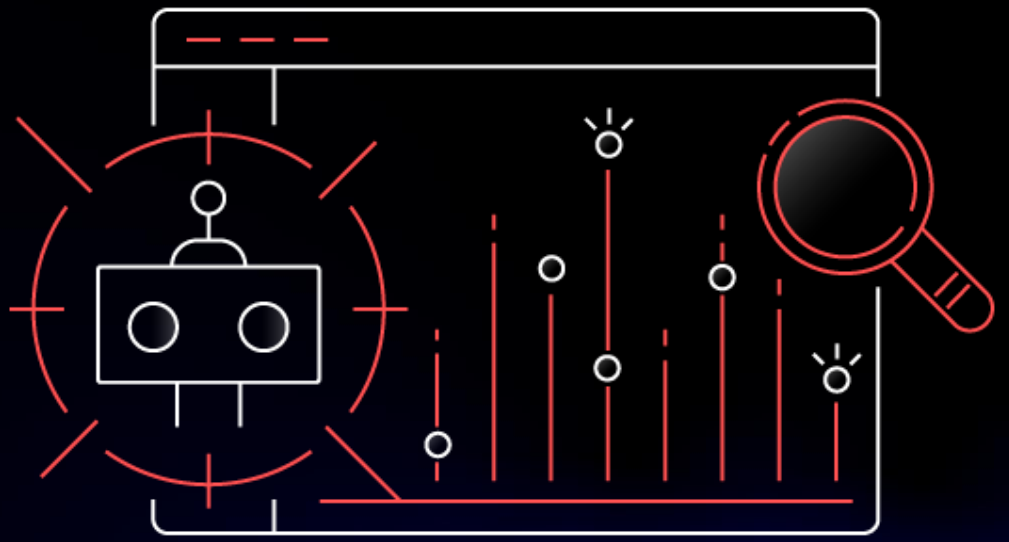
Web hosting provider

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is a bot?

A software programmed to do specific task repetitively

- Automated
- Mimics human behavior
- Precise and deliberate



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What can a bot do?



Crawlers
Traverse and index accessible webpages



Scrapers
Download contents from webpages, deplete resources



Spammers
Post unwanted content, make undesirable requests



Monitors
Check application status, perform automated tests

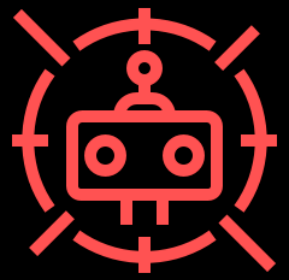
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Good bots vs. bad bots



Good traits

- Identifies itself
- Rate-limits itself
- Uses API
- Follows **robots.txt**
- Caches data between requests



Bad traits

- Obscures itself
- Tests application’s limit
- Abuses API
- Does not respect **robots.txt**
- Does not cache data between requests

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Quick recap

There are tons of bots on the internet

- Some are good and some are bad, and all of them have a purpose
- Sometimes even with good intention, bots can have a negative effect on your application
- Not dealing with these bots can lead to unintended security consequences

So what can you do about them?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS WAF and Bot Control

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Detecting a bot is challenging . . .



Architecture changes



Finding the right signal



Constant maintenance

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

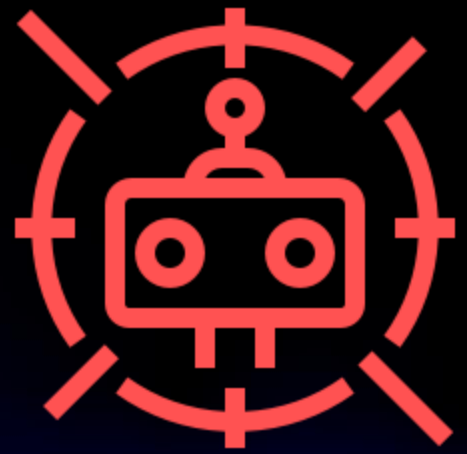
What is Bot Control?

Designed to help stop common and pervasive bot traffic on your application

Identify and categorize bots

Combine with natively available mitigation techniques on AWS WAF

Insight and visibility into bot traffic



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Bot Control is part of AWS WAF

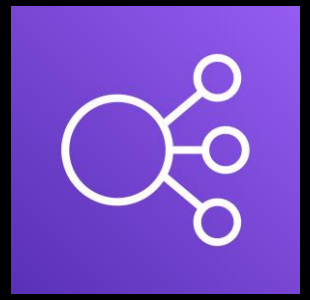


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

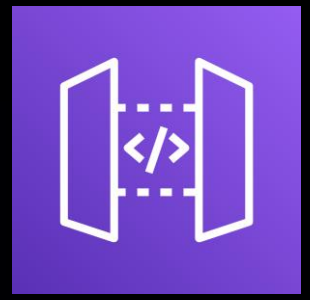
Where AWS WAF can be deployed



Amazon CloudFront



Application Load Balancer



Amazon API Gateway



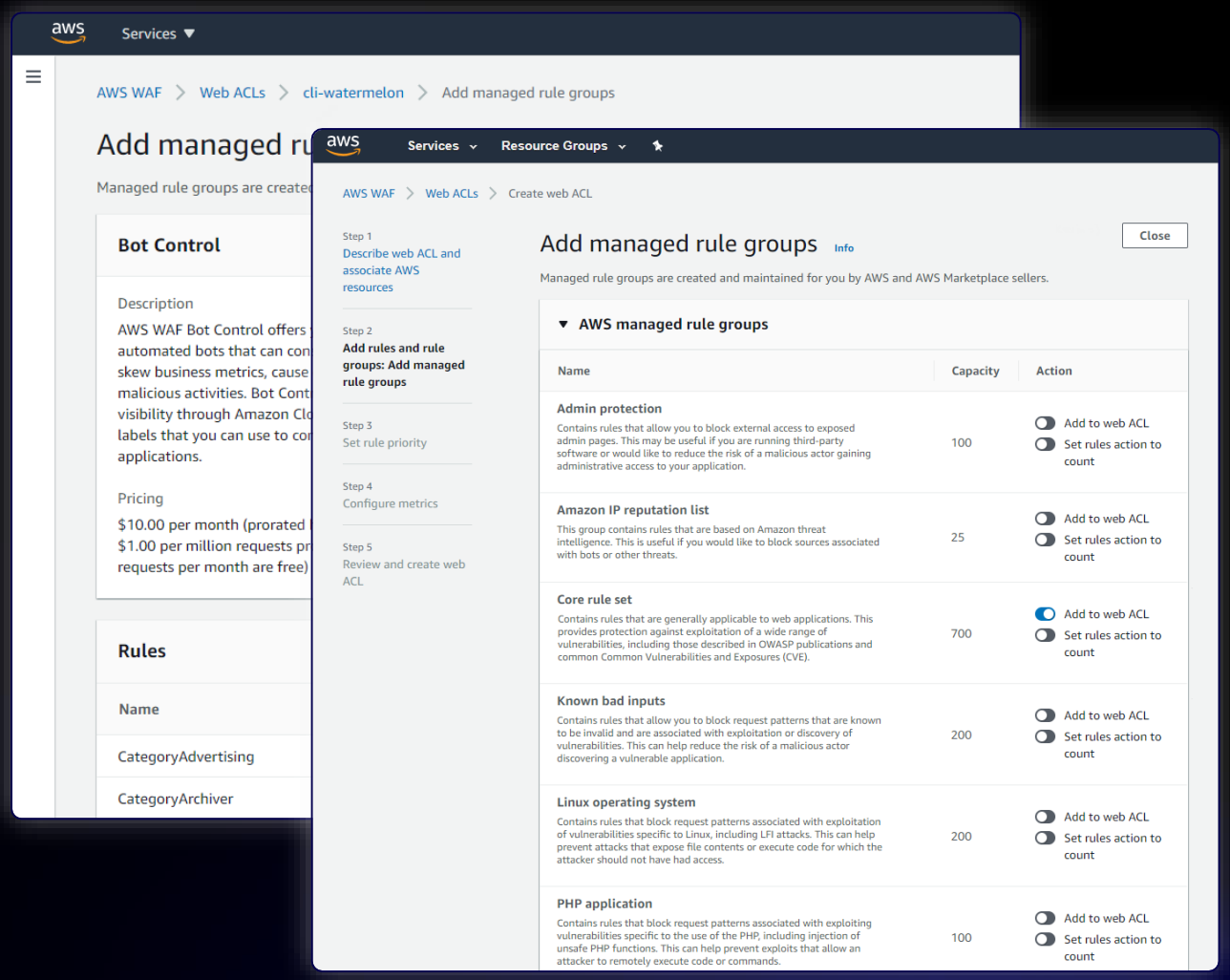
AWS AppSync

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Activating Bot Control

Part of AWS managed rules

- Guided step-by-step wizard
- Pre-configured
- Customizable
- Support for AWS Firewall Manager

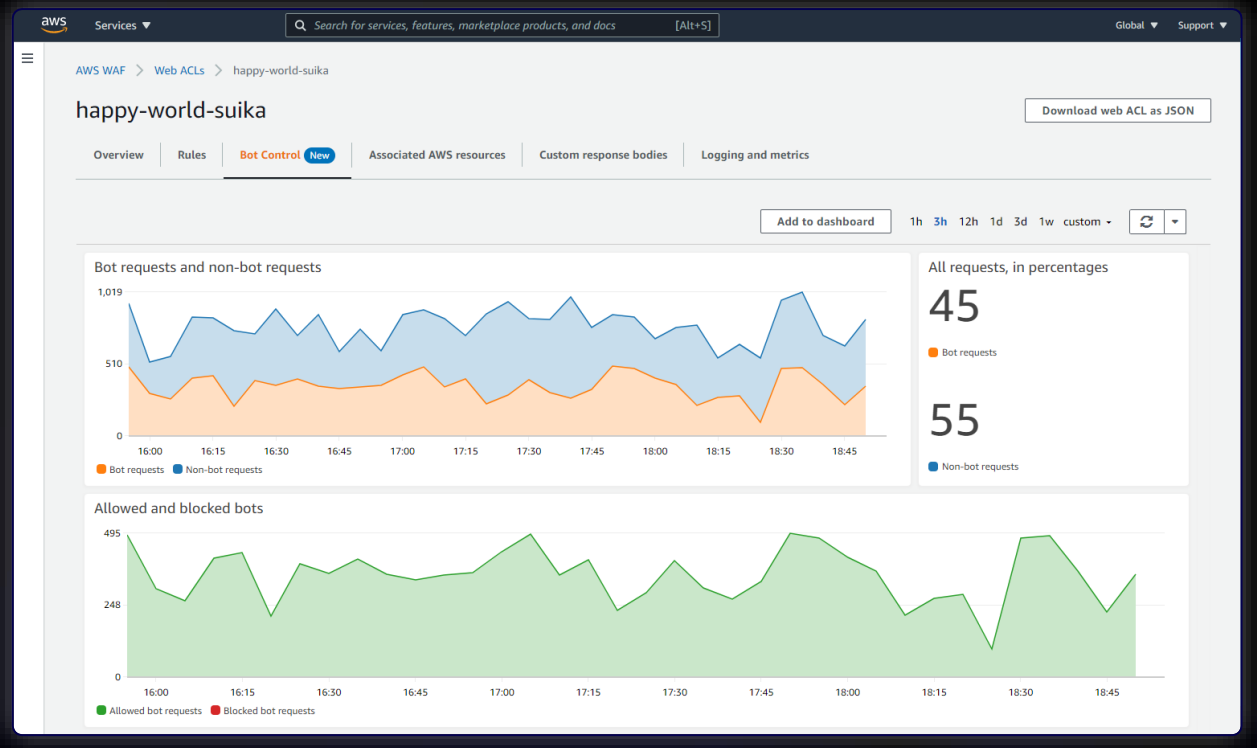


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New signatures are added to Bot Control automatically

Built on a dedicated data plane

- Real-time heuristic analysis
- Reverse DNS lookup
- Amazon.com intelligence
- Free bot visibility

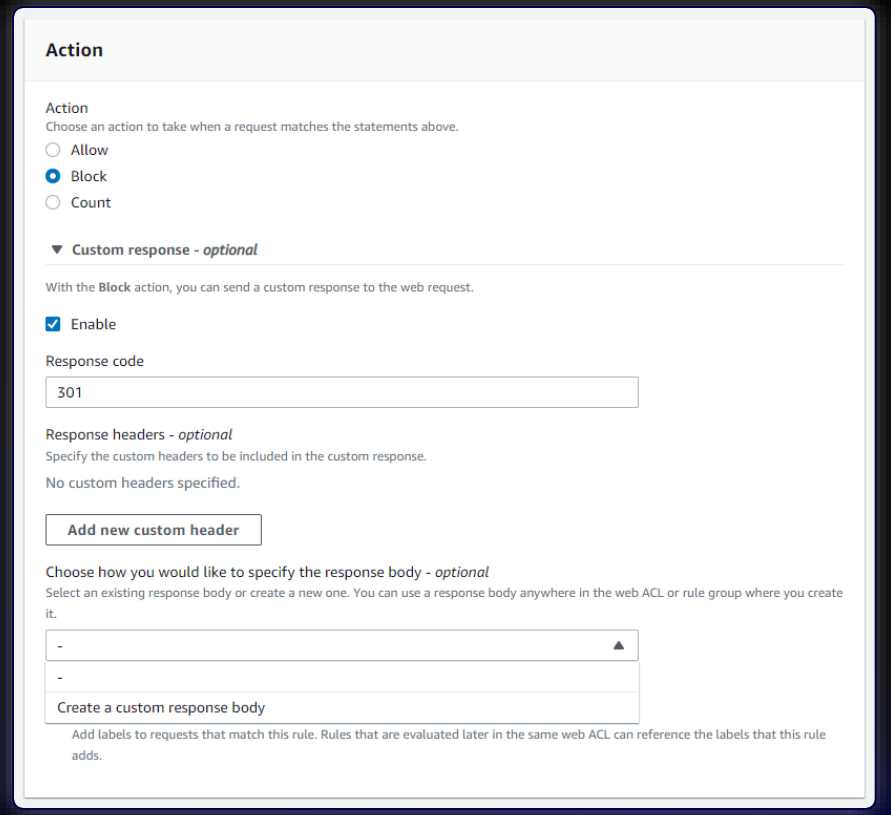


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Leveraging AWS WAF and Bot Control together

AWS WAF offers multiple ways to mitigate bots

- Simple actions (e.g., block, allow, count)
- Labels
- Rate-based rules
- Custom response
- Request header insertion



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Quick recap – What does this all mean for me?



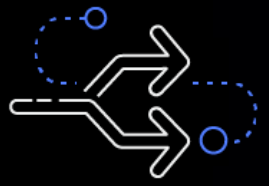
Reduce backend infrastructure cost

Lessen the burden from bots like scrapers, scanners, and crawlers



Fast deployment

No architecture change required, get started immediately



Simple yet highly customizable

Take advantage of WAF rule engine and its mitigation techniques



Gain additional insight

Find out about bots that are roaming within your traffic

aws

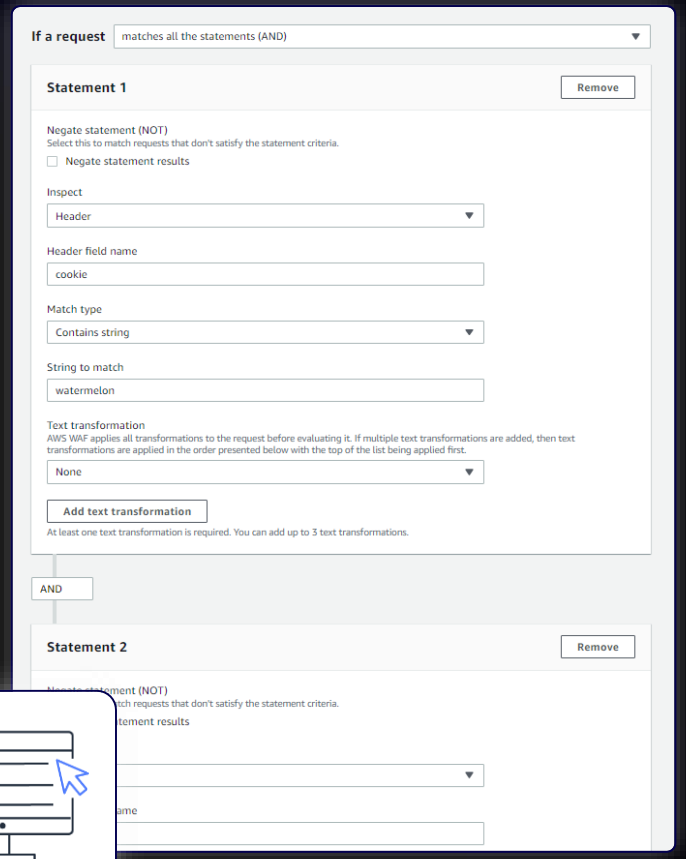
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Bot Control deep dive

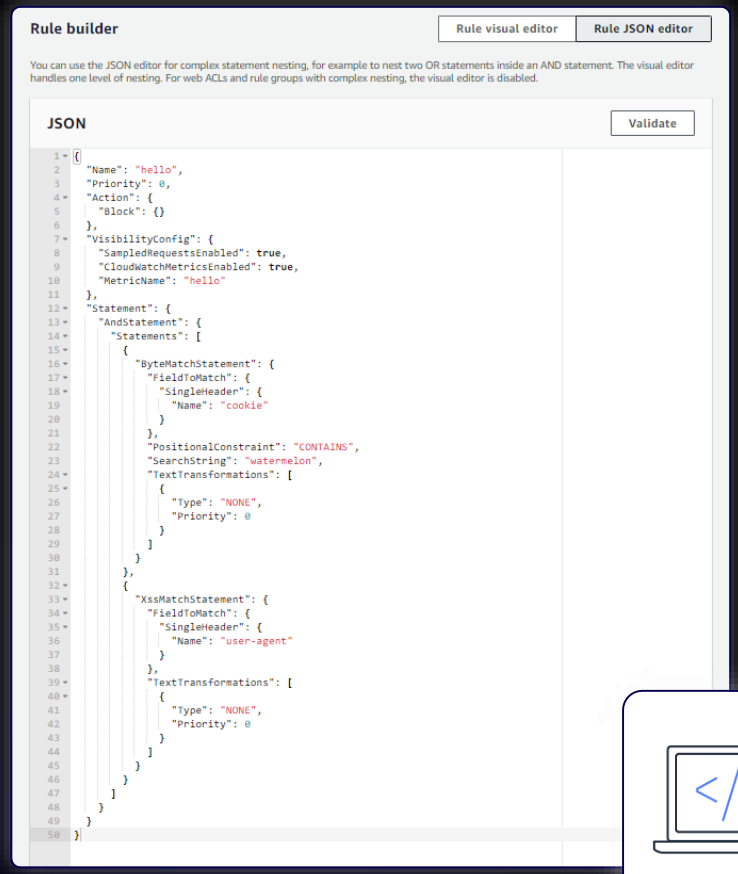
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Writing rules in AWS WAF

Visual rule builder



Code (JSON)



https://youtu.be/ll-uvVgQ3Jg

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are labels?

Essentially a tag that can be applied to individual request

- Act on signals generated by Bot Control
- Store outcome of rule evaluation
- Write exception around managed rules



Visibility through Amazon CloudWatch metrics and AWS WAF logging

- Aggregate metric from multiple rules

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Bot Control and labels

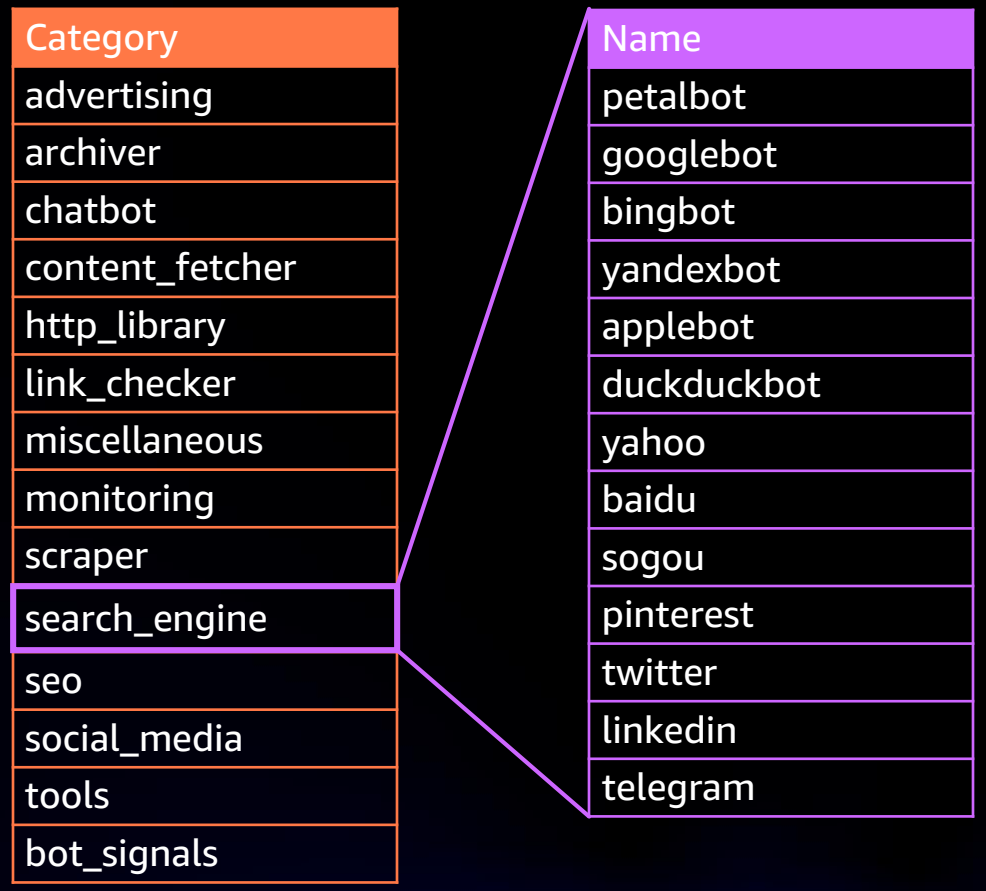
- When enabled, various labels are emitted
- bot:category: – Type of bot
- bot:name: – Name of the bot (if available)
- bot:verified: – Indication whether it's a good bot or not
  - Validated by looking up rDNS and AS number
  - Verified bots are not blocked by default!
- signal: – Various attributes that are indicative of bot activity



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Examples of Bot Control labels

- Label information can be found under detail page for Bot Control
- DescribeManagedRuleGroup for API



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

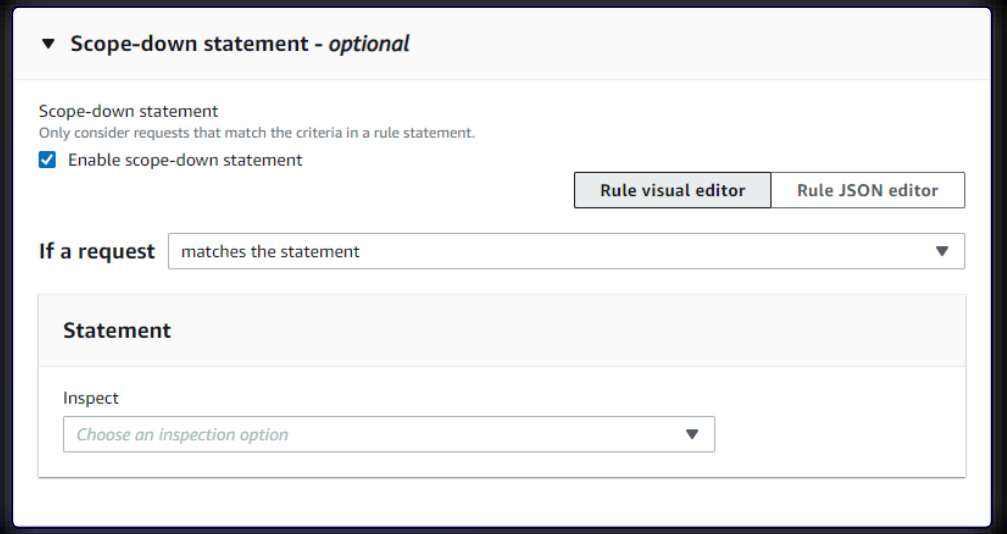
What is scope-down statement?

Not all traffic needs to be analyzed

- E.g., images and favicons, allowlisted IPs

Limits the inspection to specific conditions provided

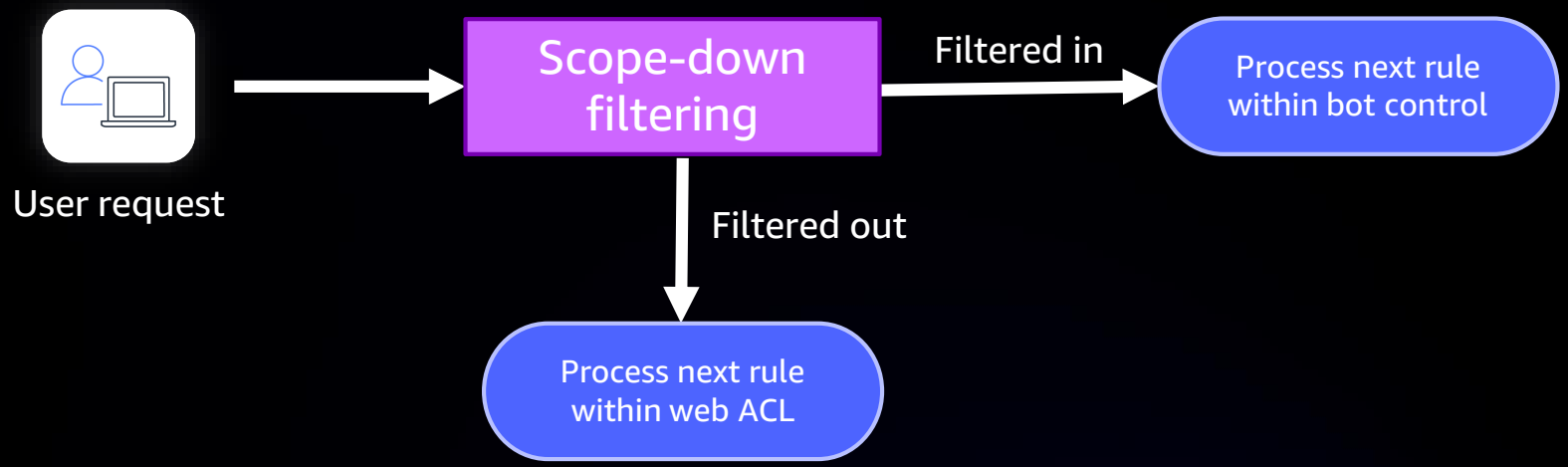
Helps control cost for paid managed rules



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Inspecting dynamic content only

"I want Bot Control to ignore static content like images."



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Inspecting dynamic content only (JSON)

```
"ScopeDownStatement": {
  "NotStatement": {
    "Statement": {
      "RegexMatchStatement": {
        "FieldToMatch": {
          "UriPath": {}
        },
        "TextTransformations": [
          {
            "Type": "NONE",
            "Priority": 0
          }
        ],
        "RegexString": "(?i)\\.(jpe?g|gif|png|svg|ico|css|js|woff2?)$"
      }
    }
  }
}
```

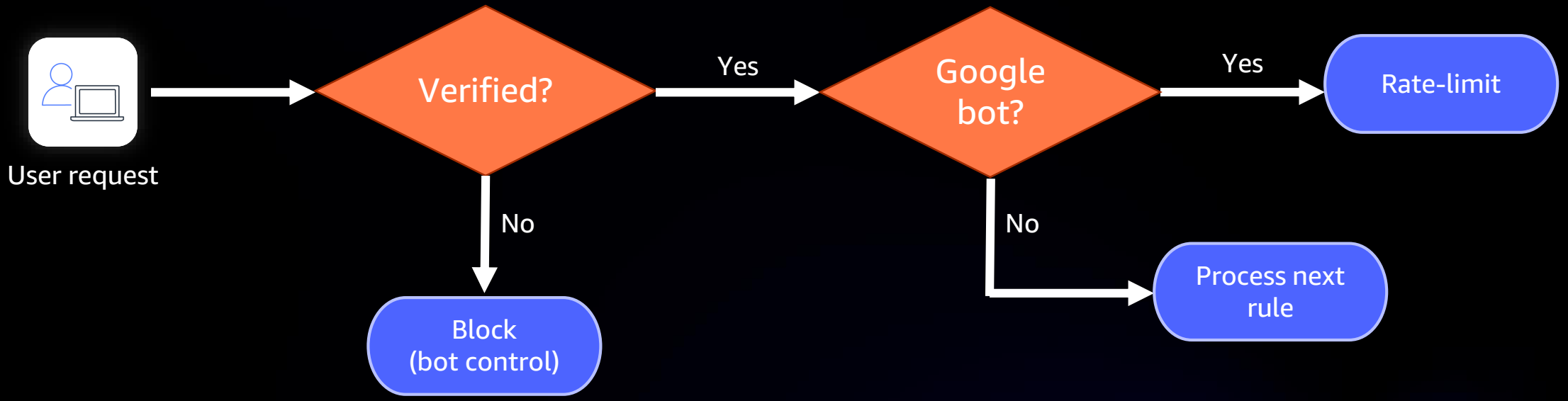
NOT(RegexString A)

← Regex that looks for static content

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Slowing down verified bot

"I want to slow down how much Googlebot crawls my application."



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Slowing down verified bot (JSON)

[(Label A) AND (Label B)]

```
...
"Statement": {
    "RateBasedStatement": {
      "Limit": "100",
      "AggregateKeyType": "IP",
      "ScopeDownStatement": {
        "AndStatement": {
          "Statements": [
            {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:name:googlebot"
              }
            },
            {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:verified"
              }
            }
          ]  }  }  }  }  }
```

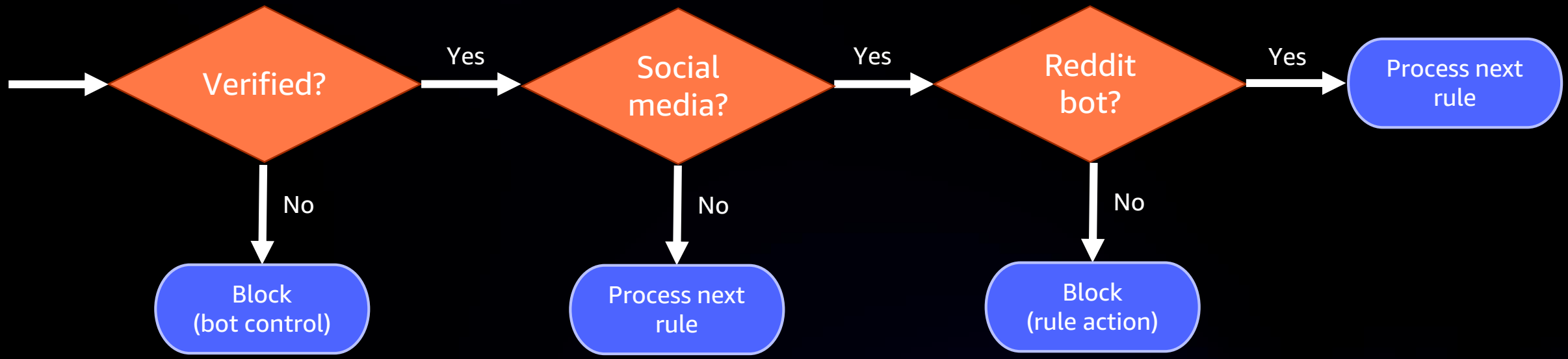
Indicates rate-limiting rule type

Rate-based rule also has optional scope-down statement

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Allowing specific bot only

"I want to block all social media bots except redditbot."



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Allowing specific bot only (JSON)

```
{
  "Name": "social-media-bot",
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "Statement": {
    "AndStatement": {
      "Statements": [
        {
          "LabelMatchStatement": {
            "Scope": "LABEL",
            "Key": "awswaf:managed:aws:bot-control:bot:category:social_media"
          }
        },
        {
          "NotStatement": {
            "Statement": {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:name:redditbot"
              } } } } ... ] } } }
```

Complex action can be defined here

[(Label A) AND NOT(Label B) AND (Label C)]

Not shown (verified)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Leveraging complex actions with Bot Control

Simple action alone might not be sufficient for thwarting out bots

- May change tactic to avoid detection

Consider:

- Sending custom response with fake page
- Flagging request to warn downstream application
- Rate limiting on specific condition
- Ask to solve Captcha

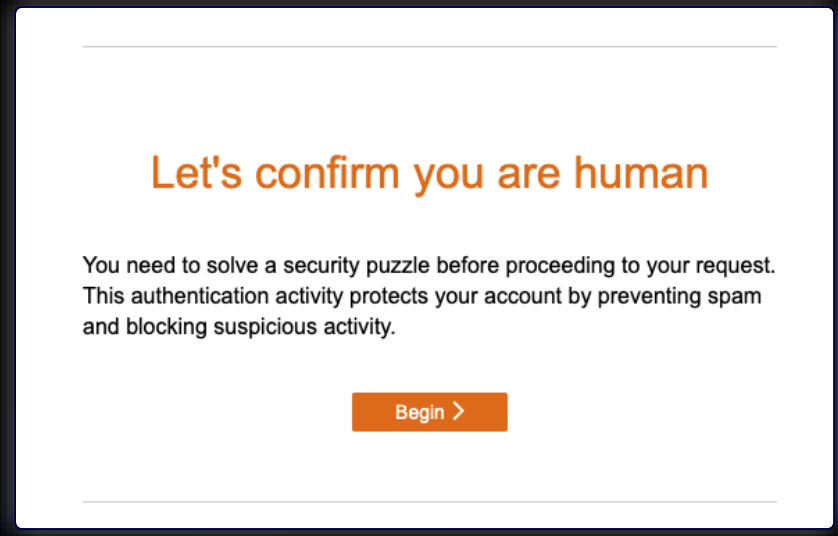


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Captcha for AWS WAF

Self-explanatory . . . ☺

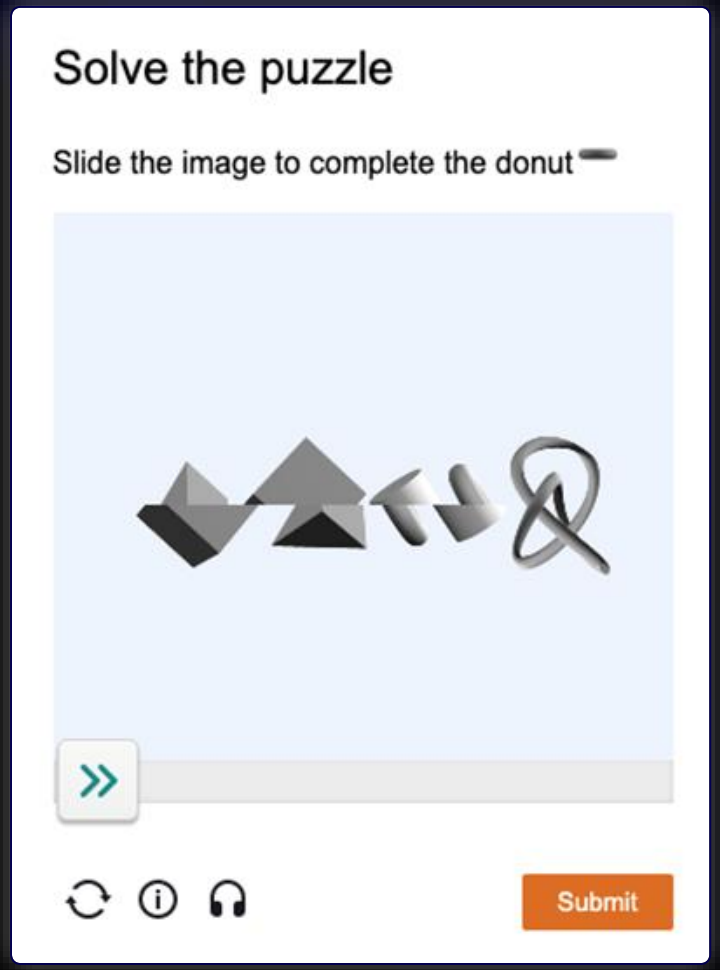
- Original
- Rule action
- Changeable immunity time
- Currently available in:
  - US East (N. Virginia)
  - US West (Oregon)
  - Europe (Frankfurt)
  - South America (Sao Paulo)
  - Asia Pacific (Singapore)

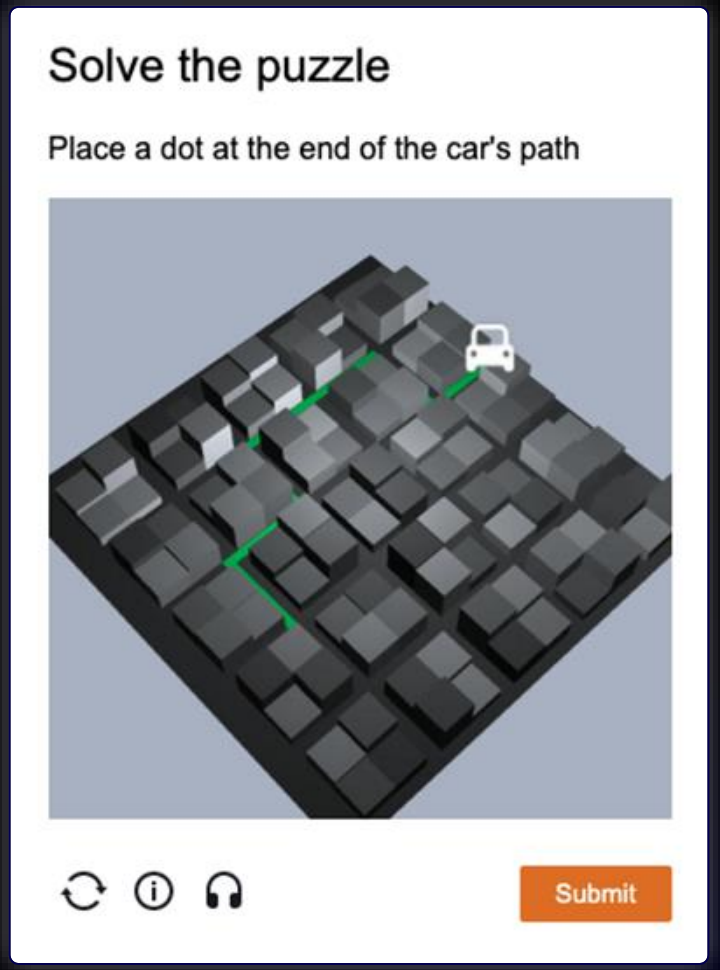


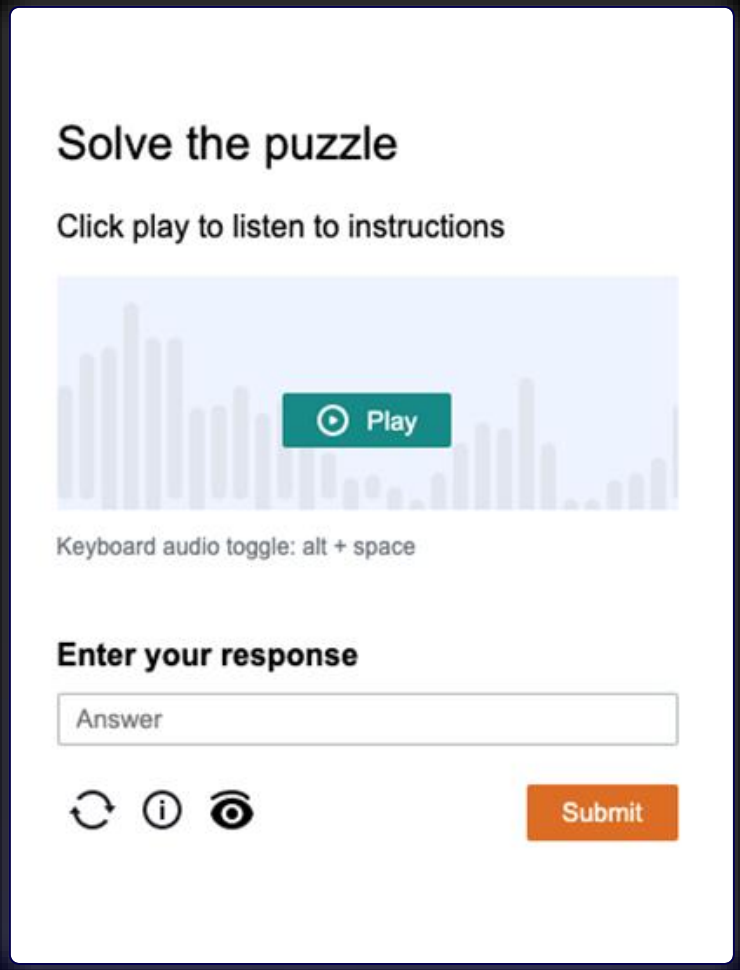
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Supported puzzles







© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Custom response (JSON)

```
{
  "Name": "social-media-bot",
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "Statement": {
    "AndStatement": {
      "Statements": [
        {
          "LabelMatchStatement": {
            "Scope": "LABEL",
            "Key": "awswaf:managed:aws:bot-control:bot:category:social_media"
          }
        },
        {
          "NotStatement": {
            "Statement": {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:name:redditbot"
              } } } } ... ] } } }
```

Complex action can be defined here

[(Label A) AND NOT(Label B) AND (Label C)]

Not shown (verified)

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Custom response (JSON)

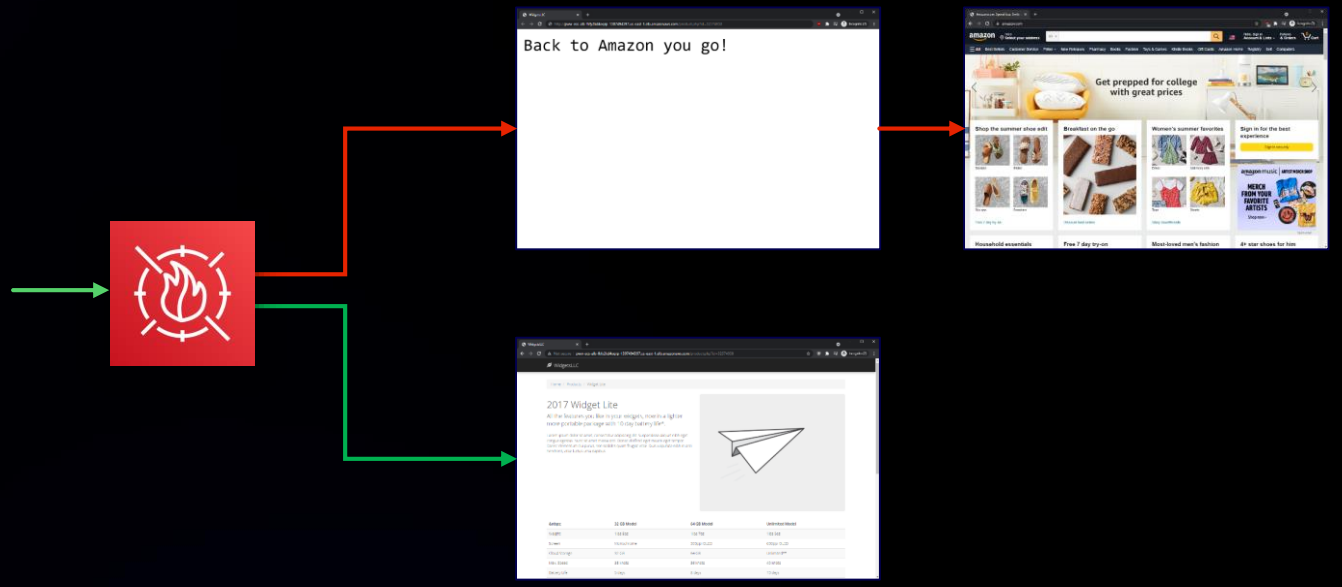
```
{
  "Name": "social-media-bot",
  "Priority": 0,
  "Action": {
    "Block": {
      "CustomResponse": {
        "ResponseCode": 301,
        "CustomResponseBodyKey": "some-fake-page",
        "ResponseHeaders": [
          {
            "Name": "location",
            "Value": "https://www.amazon.com/"
          }
        ]
      }
    }
  },
  "Statement": {
    "AndStatement": {
       ...
   } }
}
```

```
"CustomResponseBodies": {
  “some-fake-page": {
    "ContentType": "TEXT_PLAIN",
    "Content": “Back to Amazon you go!”
  }
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Custom response (JSON)

```
{
  "Name": "social-media-bot",
  "Priority": 0,
  "Action": {
    "Block": {
      "CustomResponse": {
        "ResponseCode": 301,
        "CustomResponseBodyKey": "some-fake-page",
        "ResponseHeaders": [
          {
            "Name": "location",
            "Value": "https://www.amazon.com/"
          }
        ]
      }
    }
  },
  "Statement": {
    "AndStatement": {
      ...
    } }
}
```



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Helpful blog posts for reference

AWS Security Blog

Customize requests and responses with AWS WAF

by Kaustubh Phatak and EJ Chen | on 21 JUN 2021 | in AWS WAF, Intermediate (200), Security, Identity, & Compliance | Permalink | Comments | Share

In March 2021, AWS introduced support for custom responses and request header insertion with AWS WAF. This blog post will demonstrate how you can use these new features to customize your AWS WAF solution to improve the user experience and security posture of your applications.

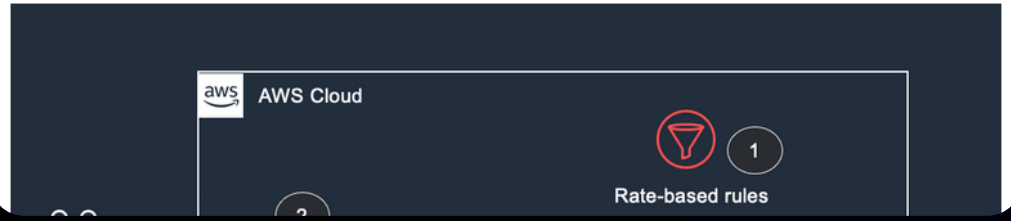
HTTP response codes are standard responses sent by a server in response to a client request. When AWS WAF blocks a request, the default response code sent back to the client is HTTP 403 (Forbidden). The HTTP 403 response code is associated with a default error page built by the web server engine. This page is typically generic and not user-friendly. With the Custom Response feature, AWS WAF now allows you to modify the status code from HTTP 403 to HTTP 2xx, 3xx, 4xx, and 5xx, and to return a custom body when the request is blocked by AWS WAF. The custom responses unique to AWS WAF also allow you to differentiate blocked requests generated by AWS WAF or your server.

When inspected HTTP requests are allowed by AWS WAF, the request is passed through to the associated resource. Now you have the ability to insert custom HTTP request headers for each rule inside your web access control list (web ACL) set to **allow** or **count**, and you can create additional logic with your application by tagging these requests with the headers.

We will be outlining three different use cases to show how you can use these AWS WAF features.

Use case 1: Custom response code

In this example, you will use the custom response code feature to redirect a viewer request to a different webpage. You use HTTP 3xx response codes to redirect the incoming request, and use the HTTP header **Location** to specify the website URL for redirection. Figure 1 shows an overview of this workflow.



AWS Security Blog

The three most important AWS WAF rate-based rules

by Artem Lovan and Jesse Lepich | on 22 JUL 2021 | in Advanced (300), Amazon Athena, AWS WAF, Security, Identity, & Compliance | Permalink | Comments | Share

In this post, we explain what the three most important AWS WAF rate-based rules are for proactively protecting your web applications against common HTTP flood events, and how to implement these rules. We share what the Shield Response Team (SRT) has learned from helping customers respond to HTTP floods and show how all AWS WAF customers can benefit from these learnings.

When you have business-critical applications that are internet-facing, you need to protect them from risks such as distributed denial of service (DDoS) attacks. AWS Shield Advanced is a managed DDoS protection service that safeguards applications that are running behind Amazon Web Services (AWS) internet-facing resources. The backend origin of your application can exist anywhere, including on premises, and Shield Advanced can protect it. Shield Advanced provides DDoS protection for Layers 3–7. It also includes 24/7 access to the SRT to help you quickly respond to sophisticated unauthorized activity scenarios that might be unique to your application. To learn more about what resource types are supported to associate AWS WAF, see AWS WAF.

Increasingly, the SRT has been assisting customers in protecting against Layer 7 HTTP flood occurrences that negatively impact application availability or performance by overloading the application with an unusually high number of HTTP requests. In many cases, these malicious events can be automatically mitigated by using AWS WAF. In addition, AWS WAF has an easy-to-configure native rate-based rule capability, which detects source IP addresses that make large numbers of HTTP requests within a 5-minute time span, and automatically blocks requests from the offending source IP until the rate of requests falls below a set threshold. In this post, we show how you can pull insights from the AWS WAF logs to determine what your rate-based rule threshold should be.

The top three most important AWS WAF rate-based rules are:

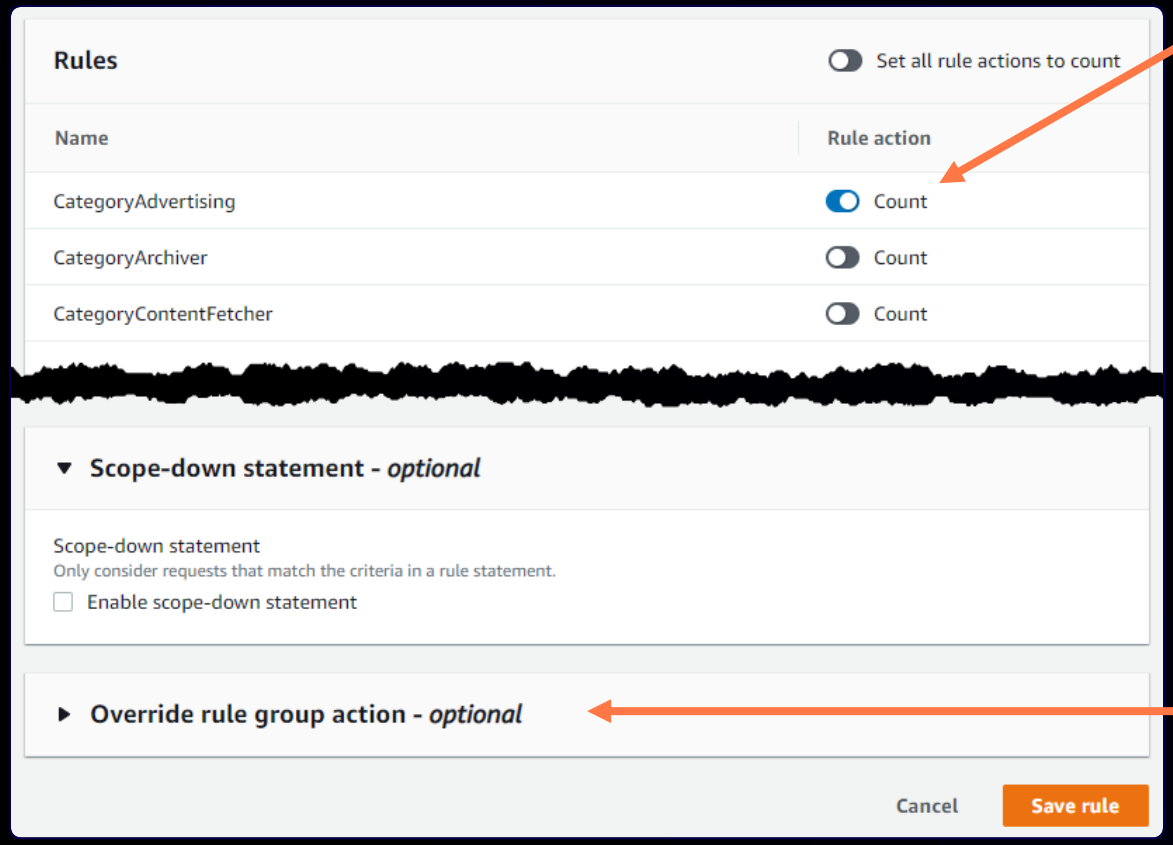
- A blanket rate-based rule to protect your application from large HTTP floods.
- A rate-based rule to protect specific URIs at more restrictive rates than the blanket rate-based rule.
- A rate-based rule to protect your application against known malicious source IPs.

Solution overview

AWS WAF is a web application firewall that helps protect your web applications against common web exploits that might affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over which web

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Mitigating false-positive scenarios



1. Exclude the rule to count mode
2. Write a custom rule at web ACL level using the label match statements
3. Use sampled request or logging to verify desired outcome

Overrides action taken by the rule group (useful for monitoring only)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

General tips and guidance

Profile your traffic first by putting into count mode

When customizing, rule priority matters

JSON is more flexible and powerful than console for rule writing

Use scope-down statement to limit what only needs to be inspected



Quick recap – Where do we go from here?

Bot Control is simple to setup yet offers many customization options

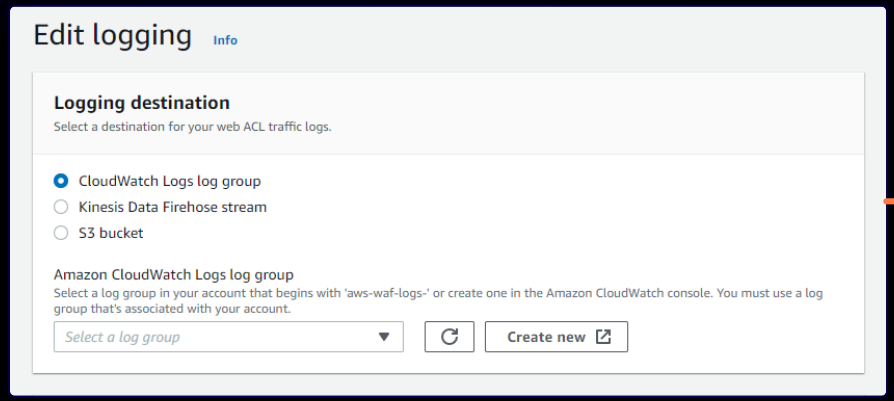
- Effective strategy against reducing common and pervasive bots

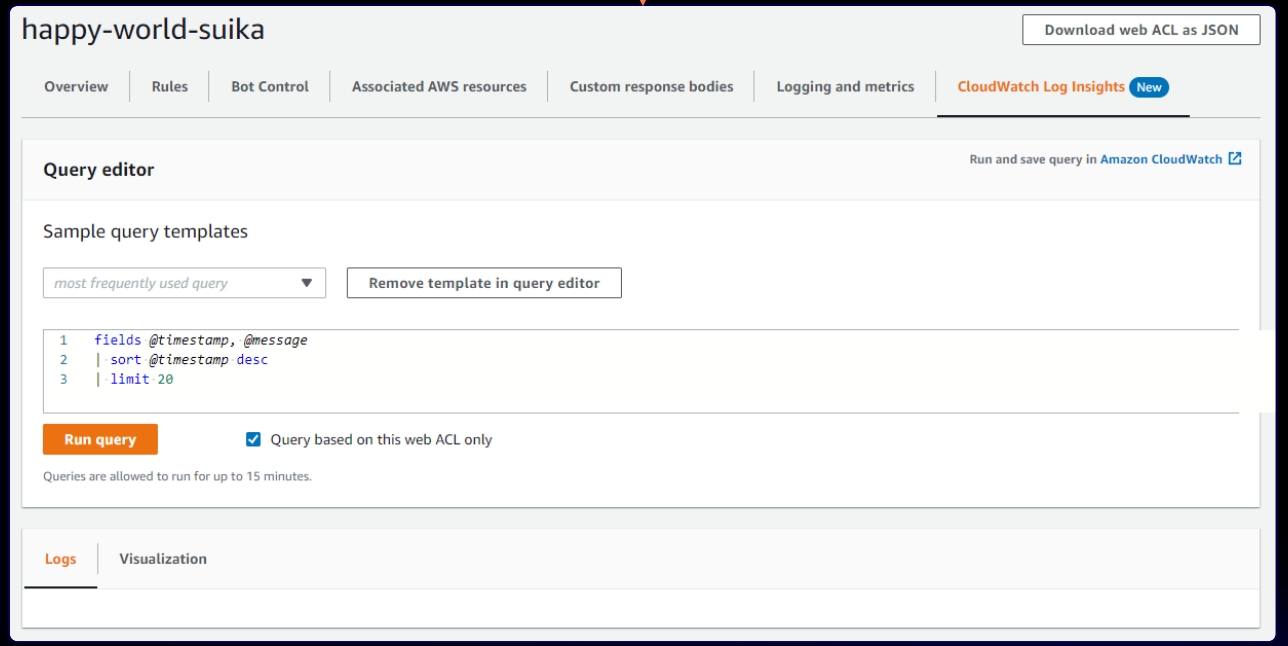
Looking ahead:

- Machine learning
- Additional bot telemetry
- Targeted bot attacks

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New feature: CloudWatch Logs





© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Kevin Lee

Sr. Product Manager (Technical)
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.