

Lab Instructions

Site: [Barracuda CloudGen WAF Training Portal](#)

Course: API Security with Barracuda WAF - automated and simplified

Book: Lab Instructions

Printed by: aravindan a

Date: Monday, 24 May 2021, 1:28 AM

Table of contents

1. Lab Objective
2. Lab environment setup
3. Familiarize with the Backend REST API
4. Access and configure the Barracuda WAF
5. Protecting the API-Part 1: Configure the Virtual Service
6. Protecting the API-Part 2: Import the swagger file and create the rules on the WAF
7. Deploy the front end application
8. Petstore Frontend App User Manual
9. Check the logs and fine tune the WAF settings
10. Change the Security Policy Mode
11. Conclusion and Next Steps

1. Lab Objective

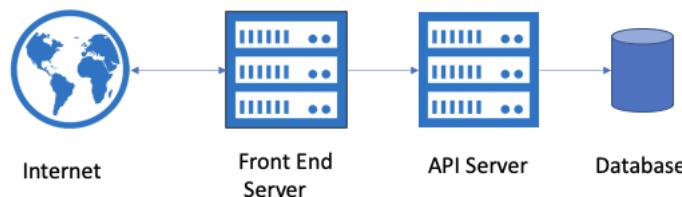
Objective

The objective of this hands on session is to deploy a Web Application called the PetStore and provide security at each layer.

About the Petstore Application

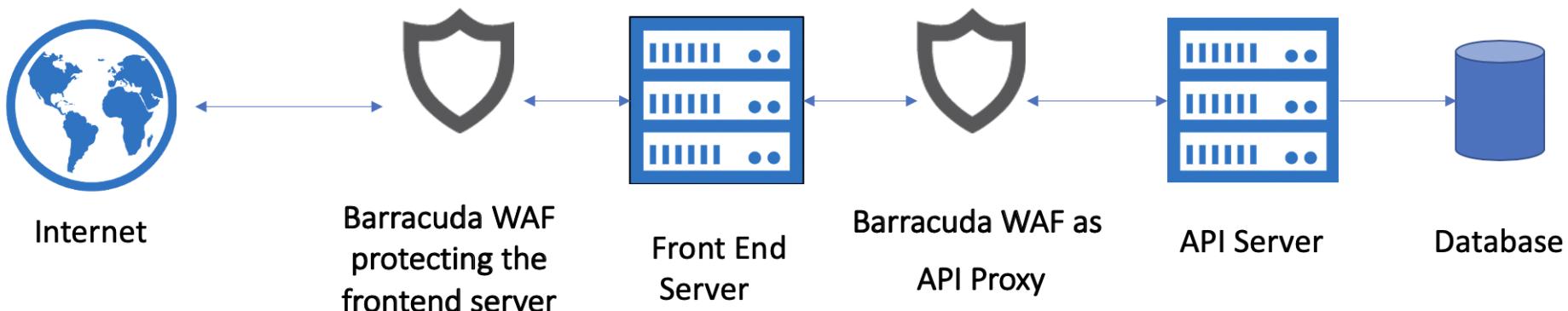
Pet Store comprises of a backend application i.e API Server and a Front End Server.

The Frontend Server communicates to the backend application through REST API. The frontend server stores and retrieves data from the database through APIs. Clients can connect to the application via the front end server.



Deploying security for the Pet Store application

We will use the Barracuda WAF Virtual Machine solution to deploy security for the front end and the API server of Pet Store. The intended deployment can be schematically represented as below:



This design provides the following benefits:

1. Defence in depth: Barracuda WAF will protect the Pet Store application from any malicious requests from the attackers.
2. The API server can be scaled up as per demand or migrated to a different network without too much of network / deployment changes.
3. If the Pet Store company wants to change the front end server to adopt a modern web application framework, it can be done without changing the network design.

4.The Pet Store company can expose the API Server to another service provider to integrate their services, for example, to add order payment feature or to support auxillary services like pet care accessories.

2. Lab environment setup

The environment for the lab will have two resources auto created in the AWS Oregon Region:

1. Barracuda WAF

Make a note of the AMI-ID (this is the password to login to the WAF), Public IP and Private IP of the Barracuda WAF.

Instance ID / WAF Password	
Description	Status Checks
Instance ID	i-0b2c406a57 [REDACTED]
Instance state	running
Instance type	m4.large
Finding	You may not have permission to access AWS Compute Optimizer.
Private DNS	ip-1[REDACTED]-69.us-west-1.compute.internal
Private IPs	[REDACTED].69
Secondary private IPs	
VPC ID	vpc-9163fcf5
Public IP	
Public DNS (IPv4)	ec2-54-219-176-165.us-west-1.compute.amazonaws.com
IPv4 Public IP	54.219.176.165
IPv6 IPs	-
Elastic IPs	
Availability zone	us-west-1b
Security groups	Backend-WAFSecurityGroup-9X8TVJG99AP9. view inbound rules. view outbound rules
Scheduled events	No scheduled events
AMI ID	CudaWAF-p5-vm4.6.3-fw10.1.0.015-20200629-BYOL-9d8bd938-dd50-4899-

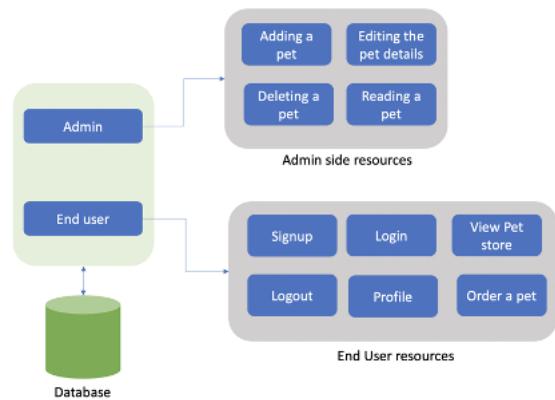
2. Backend API Server

Make a note of the public IP and the private IP of the Backend API Server

Description	Status Checks	Monitoring	Tags
Instance ID	i-0a6368cdc[REDACTED]		
Instance state	running		
Instance type	t2.medium		
Finding	You may not have permission to access AWS Compute Optimizer.		
Private DNS	ip-172-[REDACTED]us-west-1.compute.internal		
Private IPs	172.01.00.100[REDACTED]		
Secondary private IPs			
Public IP		Public IP	
		Public DNS (IPv4)	ec2-13-[REDACTED]8.us-west-1.compute.amazonaws.com
		Public IP	13.0.117.0.218
		IPv6 IPs	-
		Elastic IPs	
		Availability zone	us-west-1c
		Security groups	Backend-WebServerSecurityGroup-1F7L6P3XBQYQQ. view inbound rules. view outbound rules
		Scheduled events	No scheduled events

3. Familiarize with the Backend REST API

A schematic representation of the PetStore REST API is shown here:



PetStore's API Documentation is available at the below address:

<https://petstore.swagger.io/>

The screenshot shows the Swagger Petstore API documentation. At the top, there's a navigation bar with the Swagger logo, the URL `/api/petstore/1.0.0/openapi.json`, and an **Explore** button. Below the header, the title "Swagger Petstore" is displayed with version **1.0.0** and **OAS3**. A link to the JSON file (`/api/petstore/1.0.0/openapi.json`) is also present. A note states: "This is a sample Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net, #swagger](#)". Below this, there are links for "Terms of service", "Contact the developer", "Apache 2.0", and "Find out more about Swagger".

Servers: `http://api_server/api/petstore/1.0.0 - SwaggerHub API Auto Mocking` ▾ Authorize

pet Everything about your Pets Find out more: <http://swagger.io> ▾

POST `/pet` Add a new pet to the store

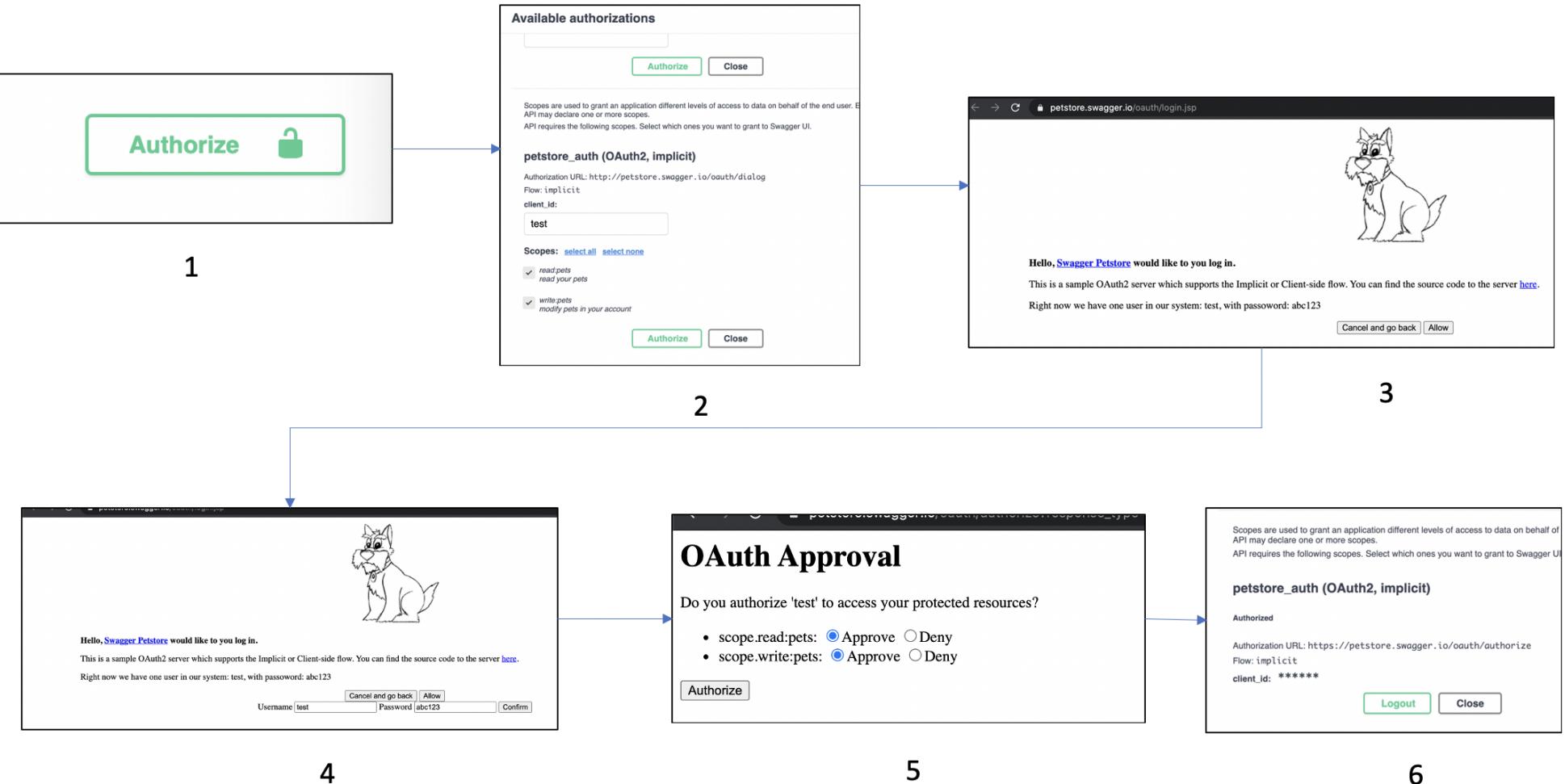
PUT `/pet` Update an existing pet

GET `/pet/findByStatus` Finds Pets by status

GET `/pet/findByTags` Finds Pets by tags

DELETE `/pet/{id}` Deletes a pet

1. Authenticating as Admin



2. Creating, Updating, Reading and Deleting Resources (CRUD Operations)

Read : Getting Information about a resource

Uses the GET HTTP Method

API Resource: pet

1

2

3

Execute request for /pet/findByStatus

Create:

Creating a resource: Uses the POST HTTP Method

3

Responses

Curl

```
curl -X GET "https://petstore.swagger.io/v2/pet/findByStatus?status=available" -H "accept: application/json" -H "authorization: Bearer"
```

Request URL

```
https://petstore.swagger.io/v2/pet/findByStatus?status=available
```

Server response

Code	Details
200	Response body <pre>{ "id": 15435006004244, "category": { "id": 0, "name": "string" }, "name": "doggie", "photoUrls": ["string"], "tags": [{ "id": 0, "name": "string" }], "status": "available" }, { "id": 1234, "category": { "id": 10, "name": "string" }, "name": "doggie", "photoUrls": ["string"], "tags": [{ "id": 10, "name": "string" }], "status": "available" }</pre>

API Response for the Request

POST /pet/{petId} Updates a pet in the store with form data

Parameters

Name	Description
petId <small>* required</small>	integer(\$int64) ID of pet that needs to be updated (path)
	petId - ID of pet that needs to be updated
name	Updated name of the pet (<i>formData</i>)
	name - Updated name of the pet
status	Updated status of the pet (<i>formData</i>)
	status - Updated status of the pet

Cancel

1

POST /pet/{petId} Updates a pet in the store with form data

Parameters

Name	Description
petId <small>* required</small>	integer(\$int64) ID of pet that needs to be updated (path)
	10
name	Updated name of the pet (<i>formData</i>)
	cat
status	Updated status of the pet (<i>formData</i>)
	sold

Execute

Responses

2

4. Access and configure the Barracuda WAF

Introduction

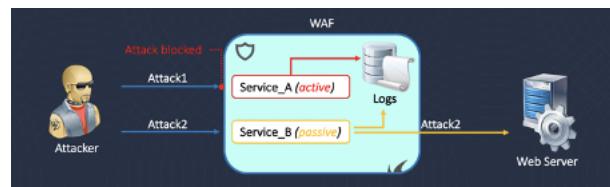
The Barracuda Web Application Firewall blocks an ever-expanding list of sophisticated web-based intrusions and attacks that target applications hosted on web servers and in the cloud. The Barracuda Web Application Firewall scans all inbound web traffic to block attacks and inspects the HTTP responses from the configured back-end servers for Data Loss Prevention (DLP).

Security Policy Modes on the Barracuda WAF

Mode – The mode determines how the service responds to offending traffic. It can either be Active Mode or Passive Mode.

- *Passive* mode logs violating events but allows the request to pass through. This is the **default** mode.
- *Active* mode performs the action configured in association with the perceived threat.

Note: *Passive* mode is recommended in the initial stages of deployment so that traffic to the service is not broken due to false positives. All traffic translation rules continue to work in both Active and Passive modes.



Getting Familiar with the Barracuda WAF

To connect to your Barracuda WAF : <http://<publicIP of the Barracuda WAF>:8000/>

After the boot process is complete, the **Licensing** page displays with the following options:

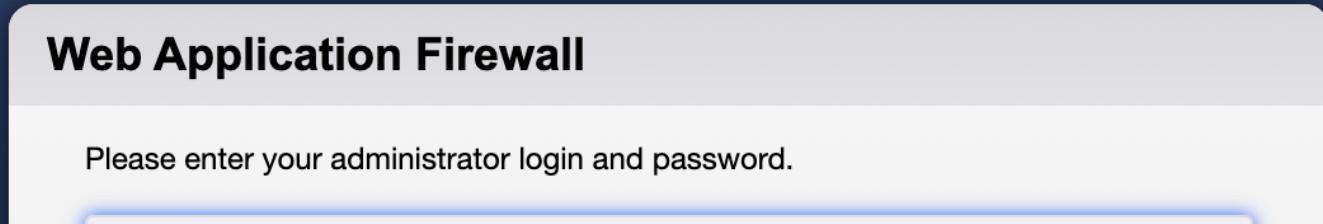


From the 3 options, choose the last i.e. "I would like to request a free Evaluation".

Use this option to get 30 days free evaluation of the Barracuda CloudGen WAF. Provide the required information in the form, accept the terms and conditions, and click **Evaluate**.

The Barracuda CloudGen WAF connects to the Barracuda Update Server to get the required information based on your license, and then reboots automatically. Allow a few minutes for the reboot process. Once the instance is provisioned, you are redirected to the login page.

.219.176.165:8000/cgi-mod/index.cgi

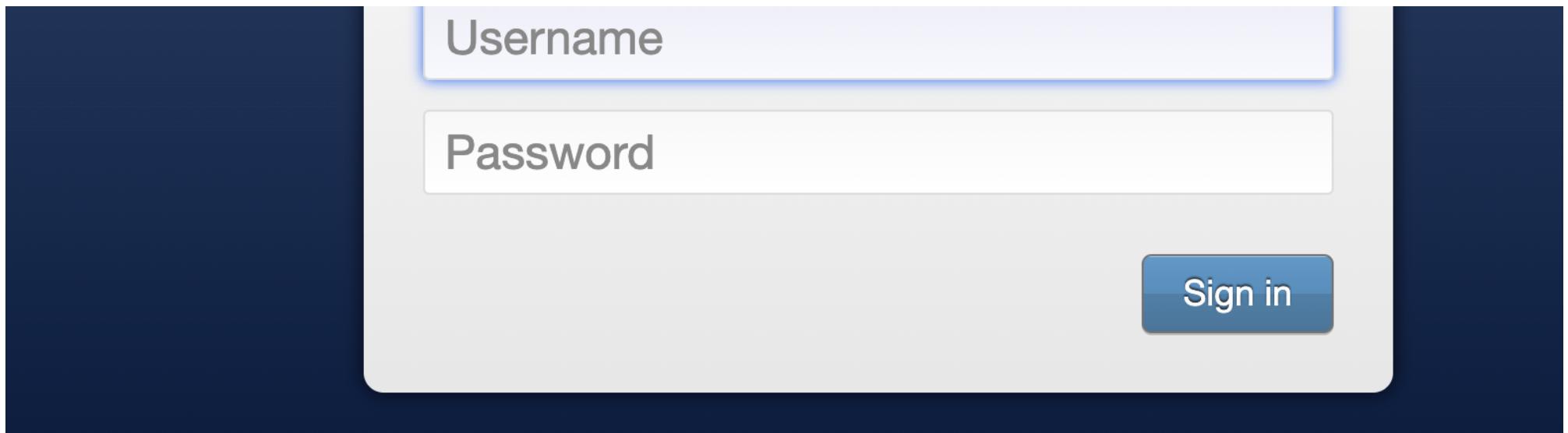


The image shows a screenshot of a web browser window. The address bar at the top contains the URL ".219.176.165:8000/cgi-mod/index.cgi". The main content area has a dark blue background. In the center, there is a white logo consisting of a stylized white 'B' shape followed by the word "Barracuda" in white lowercase letters. To the right of the logo, the words "CloudGen WAF" are written in large, bold, white uppercase letters. Below this, a white rectangular box contains the text "Web Application Firewall" in bold black font. Underneath that, the text "Please enter your administrator login and password." is displayed in a smaller black font. At the very bottom of the page, the URL "52.53.255.67/moodle/mod/book/tool/print/index.php?id=9" is visible.

Barracuda | CloudGen WAF

Web Application Firewall

Please enter your administrator login and password.



Username : admin

Password: AMI ID of the Barracuda WAF VM instance (you can get the id from the AWS Console)

Please set your System Alerts Email Address under Administration.

notes for the latest firmware version 1.175 downloaded and applied. Vulnerabilities and remediate them.

[Dismiss](#)

Welcome to Barracuda Web Application Firewall

To publish and protect your first application through the Barracuda Web Application Firewall, you will need to configure a Service. A Service is the application endpoint that is exposed to all clients. To get started click on the **Create your first service** button below.

If you need help with any configuration at any point, simply click on the **Help** dropdown menu on the top right corner. You will find links to the relevant documentation in this location. For more help, you can visit our [community forums](#) or reach out to our support team.

Thank you for choosing the Barracuda Web Application Firewall!

[Do it later](#)

[Create your first service](#)



Data Path Status:

Active

Parent Tabs on the GUI:

The screenshot shows the Barracuda Web Application Firewall's main dashboard. At the top, there's a dark blue header with the Barracuda logo and the text "Barracuda | Web Application Firewall". Below the header is a navigation bar with seven tabs: BASIC, SECURITY POLICIES, WEBSITES, BOT MITIGATION, ACCESS CONTROL, NETWORKS, and ADVANCED. A search bar is located on the far right of the navigation bar. The "BASIC" tab is currently selected.

Each of the parent tabs has its set of child tabs :

Basic Tab:

Provides the necessary options/features to get started with the product. Covers basic system administration, reporting and logging.

This screenshot shows the "SECURITY POLICIES" tab selected in the navigation bar. Below it, a sub-navigation bar lists several sub-options: Dashboard, Services, Default Security, Certificates, IP Configuration, Administration, Web Firewall Logs, Access Logs, Audit Logs, Notifications, and Reports. The "Dashboard" tab is currently active.

Security Policies Tab

A collection of 9 sub-policies which make up the global security layer for the virtual service. If no other configuration is enabled, the security policy will still guard the application from external threats.

This screenshot shows the "SECURITY POLICIES" tab selected. It displays a grid of nine sub-options: Policy Manager, Request Limits, Cookie Security, URL Protection, Parameter Protection, Cloaking, Data Theft Protection, URL Normalization, and Global ACLs. The "Policy Manager" and "Client Profile" sections are partially visible.

WebSites Tab

Covers the advanced security options. The features work in relation to the security policy but cover specific parts of the application in more detail. URLs, Parameters, Headers can be secured with a lot more granular control here. Application DDoS, IP Reputation, Anti Virus checks, Session tracking checks etc are enabled here.

This screenshot shows the "WEBSITES" tab selected. It displays a grid of eight sub-options: Allow/Deny/Redirect, Website Profiles, Advanced Security, JSON Security, IP Reputation, URL Encryption, Website Translations, and Trusted Hosts. The "Allow/Deny/Redirect" and "Adaptive Profiling" sections are partially visible.

Bot Mitigation Tab

Advanced Bot Management features can be configured here. Features include credential stuffing protection, referer header/comment/form spamming protection, web scraping protection, bot allow/deny settings.

BASIC SECURITY POLICIES WEBSITES BOT MITIGATION ACCESS CONTROL NETWORKS ADVANCED Search help topics

Bot Mitigation Bot Spam Mitigation Application DDoS Mitigation Libraries

Access Control Tab

Authentication and Authorization features can be enabled for pre-authentication requirements. Integration is available for protocols like LDAP, RADIUS, SAMLv2, OpenID Connect and Kerberos.

ACCESS CONTROL BASIC SECURITY POLICIES WEBSITES BOT MITIGATION NETWORKS ADVANCED Search help topics

Authentication Services Authentication Policies Local Users/Groups Client Certificates

Networks Tab

Covers the system networking options

NETWORKS BASIC SECURITY POLICIES WEBSITES BOT MITIGATION ACCESS CONTROL ADVANCED Search help topics

NAT ACL Network Configuration Network Firewall Logs

Advanced Tab

Covers more granular system administration tasks like Backup/Restore, High Availability, Troubleshooting, Role Based Access Control.

ADVANCED BASIC SECURITY POLICIES WEBSITES BOT MITIGATION ACCESS CONTROL NETWORKS Search help topics

Backups	Energize Updates	Firmware Update	Export Logs	System Logs	Templates	View Internal Patterns	Libraries
Admin Access Control	High Availability	Appearance	System Configuration	Secure Administration	Troubleshooting	Vulnerability Reports	
CloudGen Firewall Settings	Cloud Control	Task Manager					

5. Protecting the API-Part 1: Configure the Virtual Service

The first step to protecting the Backend API is to create a virtual service on the Barracuda WAF to process and validate the request connections.

Getting Started with the Virtual Service

You can configure the Barracuda Web Application Firewall to secure web servers from incoming traffic threats. To do this, create a service to receive the incoming traffic type (for example: an HTTP service can receive HTTP data), and then associate security settings with that service to address the security risks of that traffic type. The service also receives responses from the servers and applies security before returning responses to the client.

The Barracuda Web Application Firewall acts as a server for the client connection on the front end, and the service acts as a client to the real servers on the back end. The Barracuda Web Application Firewall fulfills each of these roles using the service and its associated configuration settings.

A service is configured with a Virtual IP (VIP) address and a TCP port. Traffic arriving at the designated VIP and port is validated, subjected to security checks configured for the service, and then passed to one of the real servers associated with that service.

To configure a Service

1. Go to Basic->Services Page

2.

Service Name : Backend_API

Service Type: HTTP

Virtual IP Address: Leave as is

Port: 8080

Identifier: IP Address

Real Servers: <Private IP of the Backend API Server>

Add New Service

Service Name	Type	Virtual IP Address	Port	Identifier	Real Servers
Backend_API	HTTP	[REDACTED]	8080	IP Address	[REDACTED]

Create Group Service Groups Web Firewall Policy

Yes No default default Add

↓

Name	Status	Hostname	IP Address	Port	Interface	Domain	URL	Type	Mode	Policy	Add	Actions
default					WAN							
default												
Backend_API	✓	[REDACTED]	[REDACTED]	8080				HTTP	Passive	default	Server Rule	Edit Disable Delete
Server [REDACTED] 80	✓	[REDACTED]	[REDACTED]	80								Edit Disable Delete

Previous 1 Next

3. As seen above the port number for the server is 80, which is incorrect. Click edit for the server entry and change to 8080

Server Configuration

Server Name:

IP Address:

Port:

Status:

Keep the default setting of In Service, or click the drop-down list and select one of the out-of-service settings as required.

Backup server:

 Yes No

This server will be used as a last resort server when all other servers fail or found to be out of service.

Weight:

1

Weight of this server to be used when the Load Balancing algorithm is Weighted Round Robin.

Comments:

SSL

Server uses SSL:

 Yes No

Set to Yes if the server protocol is over SSL. Recommended: No

SSL Protocols:

PROTOCOL	ENABLE	DISABLE
SSL 3.0 (Insecure)	<input type="radio"/>	<input checked="" type="radio"/>
TLS 1.0 (Insecure)	<input type="radio"/>	<input checked="" type="radio"/>
TLS 1.1	<input checked="" type="radio"/>	<input type="radio"/>
TLS 1.2	<input checked="" type="radio"/>	<input type="radio"/>
TLS 1.3	<input type="radio"/>	<input checked="" type="radio"/>

SaveCancel

4. Final Configuration

The port number for both service and server should be 8080

Name	Status	Hostname	IP Address	Port	Interface	Domain	URL	Type	Mode	Policy	Add	Actions
default					WAN							
default												
Backend_API	✓			8080				HTTP	Passive	default	Server Rule	Edit Disable Delete
Server_8080	✓			8080								Edit Disable Delete

Previous 1 Next

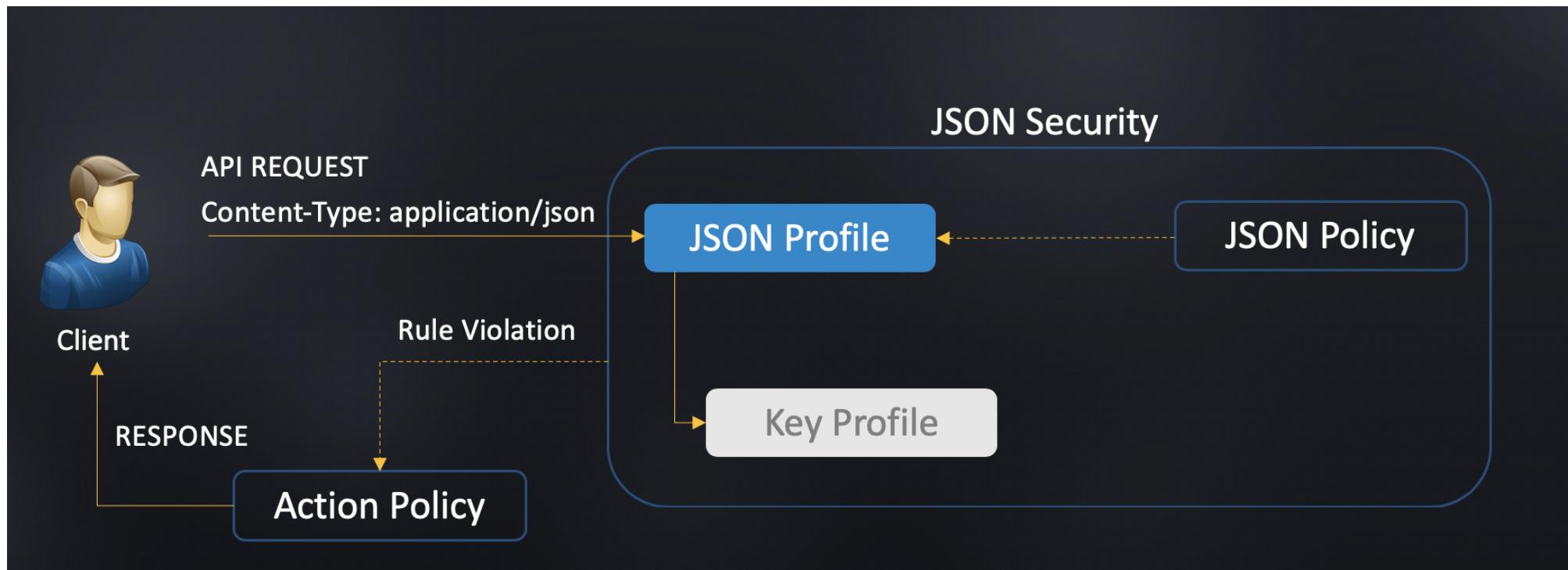
To access the API Server visit, http://<waf_publicIP>:8080/api/petstore/1.0.0/ui

Introduction to API Security on Barracuda WAF

The following are the security capabilities offered by the Barracuda WAF for APIs:

- Quota Management/Traffic Throttling
- Content Inspection
- Content Validation
- Automated Attack / Bot Detection
- Transport Layer Security
- Signature Validation
- Third Party identity provider (IdP)
- Integration with access management
- XML / SOAP Security
- SIEM Integration

Request processing workflow on the Barracuda WAF for API requests



1. Client sends a request with the Content-Type header whose value is "application/json"
2. WAF processes the request, identifies that the request is for an API from the Content-Type header.
3. The request is sent to the JSON Security module. The JSON Security module provides JSON profiles and JSON Key profiles to create API Key/Value specific rules. The Barracuda WAF uses this module in addition to features like protocol security, rate limiting, SSL encryption amongst a suite of other features to effectively safeguard an API.
4. If the request is valid, its sent to the backend server for further processing. If there is a rules violation, the request is blocked and a suitable JSON response is sent to the client by the WAF.

Since, the creation of JSON profiles and JSON key profiles can be a lengthy process, Barracuda WAF provides a way to automate this using the OpenAPI Specification File.

Proceed to the next section to use the API Spec file to create the rules on the WAF.

6. Protecting the API-Part 2: Import the swagger file and create the rules on the WAF

OpenAPI Specification

The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.

An OpenAPI definition can then be used by documentation generation tools to display the API, code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases.

[Source: <https://swagger.io/specification/>]

Importing the OpenAPI Specification File into Barracuda WAF

OpenAPI File Import

Barracuda WAF parses through the swagger specs and extracts the data which can be used to configure API Security rules.

WAF Rules affected: JSON Profile, URL Profiles, Header ACLs

Request form	Path/ Form data/ Body	Query	Headers
WAF Rules	JSON Profile, key Profiles	URL Profile, param Profile	Header ACLs

- Enables brute force protection rules for rate limiting
- Optionally enables Access Control

Getting Started

1. Download the OpenAPI Specification File for PetStore from here: https://raw.githubusercontent.com/aravindan-acct/awsdevdays_sep2020/master/petStore_req_check.yaml
2. Login to your WAF appliance as 'admin'
3. Go to WebSites -> JSON Security

The screenshot shows the Barracuda CloudGen WAF web interface. At the top, there's a navigation bar with tabs: BASIC, SECURITY POLICIES, WEBSITES (which is currently selected), BOT MITIGATION, ACCESS CONTROL, NETWORKS, and ADVANCED. To the right of the tabs is a search bar labeled "Search help topics" with a magnifying glass icon. Below the tabs, there are several links: Allow/Deny/Redirect, Website Profiles, Advanced Security, JSON Security (which is highlighted with a grey box), IP Reputation, URL Encryption, Website Translations, Trusted Hosts, XML Protection, Traffic Management, Adaptive Profiling, Exception Profiling, Exception Heuristics, and XML Validations. The main content area has a title "JSON SECURITY" and includes buttons for "Import API Spec", "Edit", and "Help". There are also some smaller buttons at the bottom left.

4. Delete the existing default json profile with the name "default-json-profile". Click on Delete to remove the rule.

This screenshot shows a list of JSON security rules. One rule is visible: "default-json-profile" with a status of "Active". The rule details are: Path: /*, Status: On, Method: POST, Policy: default-policy. To the right of the rule, there are buttons for "Add Key Profile", "Edit", and "Delete".

5. Import the OpenAPI Specification file by clicking on Import API Spec and follow the Wizard in to complete the configuration

API Discovery Wizard

Service: backend

API Specs to be used: Import new Spec file Associate existing Spec file

Select File: Choose whether upload new or use existing
C:\fakepath\petStore_req.json

API Spec Name: petstore

API Specs to be used: Name for configuration

Host/Domain Name:

Base URL: api.petstore/1.0.0

API End Points to Configure

	End Point	Method	Parameters	Rate Limit
<input checked="" type="checkbox"/>	/user/logout	get	0	10 Valid Requests OR 10 Invalid request IN 60 seconds
<input checked="" type="checkbox"/>	/user/createWithList	post	8	10 Valid Requests OR 10 Invalid request IN 60 seconds
<input checked="" type="checkbox"/>	/store/order	post	7	10 Valid Requests OR 10 Invalid request IN 60 seconds
<input checked="" type="checkbox"/>	/pet	put	6	10 Valid Requests OR 10 Invalid request IN 60 seconds

Showing 1 to 20 of 20 entries

JSON Profile

Mode Active Passive

Set to On if you want to enforce checks on requests using this JSON Profile.
default-policy

Show Advanced Settings

URL Profile

Mode Active Passive

Mode for this URL profile.
Active: Allows or blocks the requests by validating against the URL profile.
Passive: Validates the requests against the URL profile and allows to pass through, but logs the request errors. Note: The Passive mode setting will not affect the Parameter profiles under that URL profile.

Show Advanced Settings

Header ACLs

Mode Active Passive

Active: Blocks any request when an anomaly or intrusion is observed.
Passive: Logs all anomalies and intrusions but allows traffic to pass through.

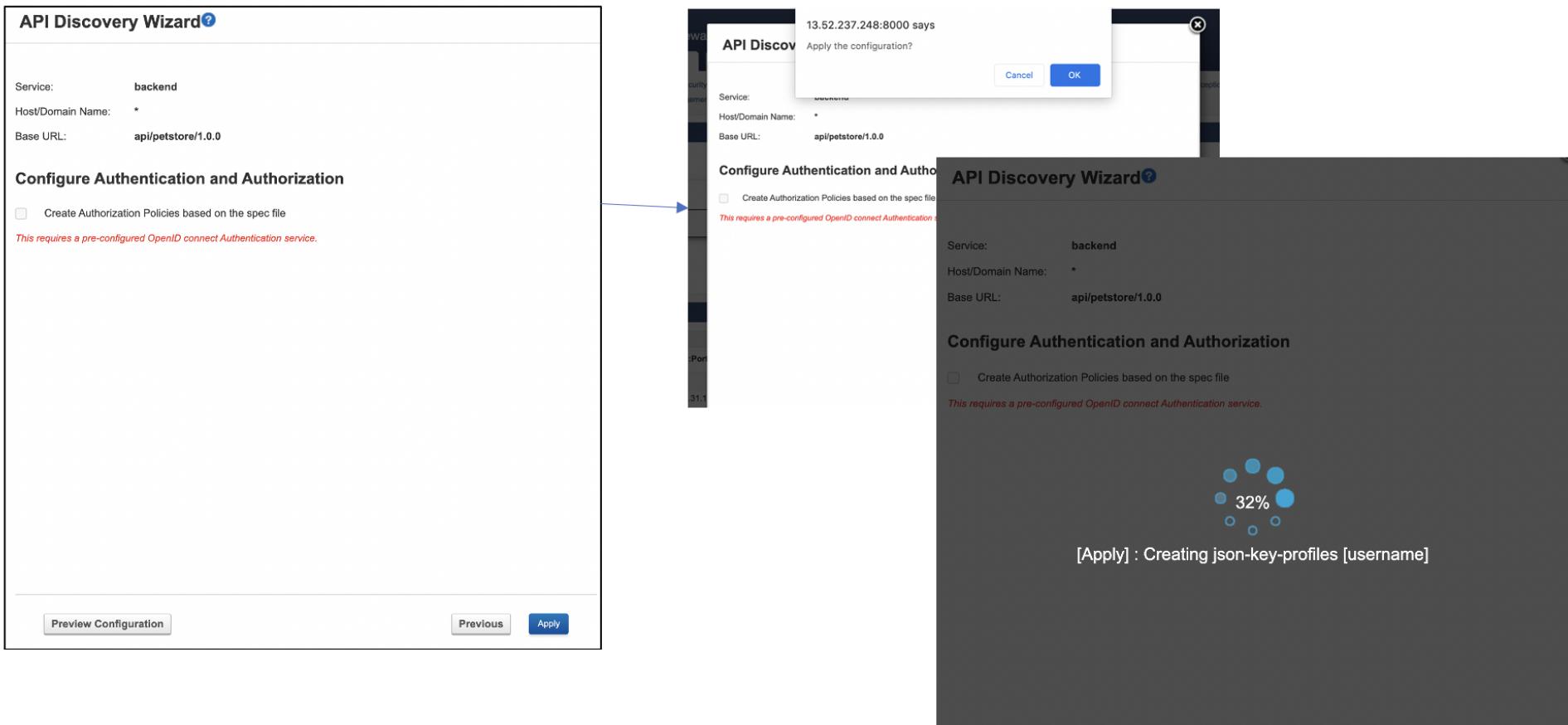
Show Advanced Settings

Application DDoS Prevention

Status On Off

Set to On to enable Slow Client Attack validation to this Service.

Show Advanced Settings



API Spec Name	Type	Import Time	Associated Services	Host/Domain Name	Base URL	Actions
petstore	openapi_3.0.0	2020-08-17 23:20:41	backend	*	api/petstore/1.0.0	Select ▾

Showing 1 to 1 of 1 entries

Previous 1 Next

JSON SECURITY						
Service	IP:Port/URI	Status	Mode	Methods	Policy/Validation	Actions
default						
backend	172.31.13.240:8080					Add JSON Profile
default-json-profile	/*	On	Active	POST	default-policy	Add Key Profile Edit Delete
restApi-user-createWithList	/api/petstore/1.0.0/u...	On	Active	POST	default-policy	Add Key Profile Edit Delete
id		On			Number	Edit Delete
username		On			String	Edit Delete
firstName		On			String	Edit Delete
lastName		On			String	Edit Delete
email		On			String	Edit Delete
password		On			String	Edit Delete
phone		On			String	Edit Delete
userStatus		On			Number	Edit Delete
restApi-user-createWithArra	/api/petstore/1.0.0/u...	On	Active	POST	default-policy	Add Key Profile Edit Delete
id		On			Number	Edit Delete
username		On			String	Edit Delete
firstName		On			String	Edit Delete

This import operation results in the rules getting configured on the Barracuda WAF for the Backend API Server

7. Deploy the front end application

Frontend Application Deployment

To deploy the Front end Application, please use the Cloud Formation Template from the following link:

https://raw.githubusercontent.com/aravindan-acct/awsdevdays_sep2020/master/frontend_app_template.yaml

To get started with the CFT deployment, please follow the instructions below:

1. Go to the Cloud Formation console

The screenshot shows the AWS CloudFormation console interface. At the top, there are navigation tabs: 'Services' (highlighted in orange) and 'Resource Groups'. Below the tabs is a search bar with the placeholder text: 'Find a service by name or feature (for example, EC2, S3 or VM, storage)'. The main content area displays several service categories with their respective icons and links:

- Compute**:
 - EC2
 - Lightsail
 - Lambda
 - Batch
 - Elastic Beanstalk
 - Serverless Application Repository
 - AWS Outposts
 - EC2 Image Builder
- Blockchain**:
 - Amazon Managed Blockchain
- Satellite**:
 - Ground Station
- Quantum Technologies**:
 - Amazon Braket
- Storage**:
 - S3
 - EFS
 - FSx
- Management & Governance**:
 - AWS Organizations
 - CloudWatch

The screenshot shows the AWS CloudFormation service interface. At the top, there's a navigation bar with 'CloudFormation' and 'Stacks'. Below it is a search bar with 'Filter by stack name' and a dropdown menu for 'Active' status. To the right are buttons for 'Create stack' (with options for 'With new resources (standard)' and 'With existing resources (import resources)'), 'Stack actions' (with a dropdown menu), 'Update', 'Delete', and a 'Close' button.

On the left, there's a sidebar with a 'Database' icon and a list of services: RDS, DynamoDB, and ElastiCache.

The main content area lists various AWS services under 'AWS Auto Scaling' and 'CloudFormation'. The 'CloudFormation' service is highlighted with a blue border.

AWS Auto Scaling
CloudFormation

Other listed services include CloudTrail, Config, OpsWorks, Service Catalog, Systems Manager, AWS AppConfig, and Trusted Advisor.

2. Select "Upload a template file" and select the file downloaded from the link above:



Specify template

Specify stack details

Configure stack options

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

 Template is ready Use a sample template Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

 Amazon S3 URL Upload a template file

Upload a template file

Choose file No file chosen

JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

[View in Designer](#)

[Cancel](#)

[Next](#)

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL

Upload a template file

Upload a template file

Choose file  **frontendapp_template.json**

JSON or YAML formatted file

S3 URL: https://s3-us-west-1.amazonaws.com/cf-templates-1ffa5vfypunam-us-west-1/20202311o0-frontendapp_template.json

View in Designer

Cancel

Next

3. Configure the launch parameters:

- WAF Password: **WAF admin password is the Instance ID of the Barracuda WAF**. This can be copied from the AWS EC2 instances page.
- StackID: **This is the name of the stack which was used to launch the api and the waf in the lab**. Please note this stack would have been auto-created. This stack name can be seen from the CloudFormation service page on the AWS Console. For example, see the following screenshot of a stack created with the name "backend". In your environment, the name will be different, so please use the name seen in your lab account.

The screenshot shows the AWS CloudFormation console with the following interface elements:

- Left Panel:** Displays a list of 11 stacks. One stack, named "backend", is highlighted with a blue border and has a blue arrow pointing to its name. The stack status is shown as "CREATE_COMPLETE".
 - Stack Name:** "backend" (highlighted with a blue box and arrow)
 - Creation Time:** 2020-09-01 08:46:52 UTC+0530
 - Status:** CREATE_COMPLETE (indicated by a green checkmark icon)
- Top Bar:** Includes navigation icons (Back, Forward, Home), a search bar ("Filter by stack name"), and filter options ("Active" dropdown and "View nested" toggle).
- Right Panel - Stack Details:** For the selected "backend" stack.
 - Title:** backend
 - Actions:** Delete, Update, Stack actions ▾, Create
 - Tab Navigation:** Stack info (selected), Events, Resources, Outputs, Parameters, Template, Change sets
 - Overview Section:** Contains the Stack ID and Description.
 - Stack ID:** arn:aws:cloudformation:us-west-2:634426045170:stack/backend/94b19590-ec01-11ea-80ff-021f9cf80633 [Copy]
 - Description:** AWS CloudFormation Sample Template
PetStore_Backend_Single_Instance: PetStore is web application that you can use to demo an API server with a frontend. This template installs PetStore Backend API Server and MySQL database for storage. In addition to this, the template also installs a Barracuda WAF instance which will be used to protect the application.

Please proceed with "Next" without any other changes in CFT creation steps in the screen i.e. Step 3 and Step 4 in the AWS console.

frontend

[Delete](#)[Update](#)[Stack actions ▾](#)[Create st...](#)[Stack info](#)[Events](#)[Resources](#)[Outputs](#)[Parameters](#)[Template](#)[Change sets](#)

Events (1)

*Search events*

Timestamp	Logical ID	Status	Status reason
2020-08-18 16:32:22 UTC+0530	frontend	CREATE_IN_PROG RESS	User Initiated

The screenshot shows the AWS CloudFormation console with the 'frontend' stack selected. The left sidebar lists three stacks: 'frontend', 'backend', and 'devops'. The 'frontend' stack is expanded, showing it was created on 2020-08-18 16:32:22 UTC+0530 and is in a 'CREATE_COMPLETE' state. The main area displays the 'frontend' stack details. The 'Events' tab is active, showing 9 events. The first event is for the stack creation, and the second is for a 'WebServer' resource.

Timestamp	Logical ID	Status	Status reason
2020-08-18 16:33:34 UTC+0530	frontend	CREATE_COMPLETE	-
2020-08-18 16:33:32 UTC+0530	WebServer	CREATE_COMPLETE	-

To access the Front End application visit: <https://<WAF Public IP>/>

The CFT creates the front end application VM as well as configures the virtual service on the WAF. Confirm that the virtual service is created on the WAF by logging into the WAF GUI. The service should be created with the name as "frontend_svc" on port 443.

The following changes are made to the Barracuda WAF configuration by the script:

1. Creates a self signed certificate called "petstore"
2. Creates a HTTPS virtual service on port 443 and binds the self signed certificate
3. TLS 1.3 is disabled since we are using a self signed certificate
4. Authentication is enabled for the frontend service
5. A local user is created for the admin portal
6. A response rewrite rule is created to handle redirects

For reference, the script can be found here:

https://raw.githubusercontent.com/aravindan-acct/frontend_UI_app/master/waf_configuration.py

The API documentation for the Barracuda WAF is available here: <https://campus.barracuda.com/product/webapplicationfirewall/api/>

If successfully deployed, you should be able get to the home page for the frontend app by visiting [**https://<PublicIP of WAF>/**](https://<PublicIP of WAF>/):



8. Petstore Frontend App User Manual

Getting Started with the application

The application provides two types of logins - admin login and end user login

Administrator login

The admin login allows the administrator of Petstore to **add/remove/edit the pets**

To access the admin portal go to : <https://<WAF publicIP>/admin>

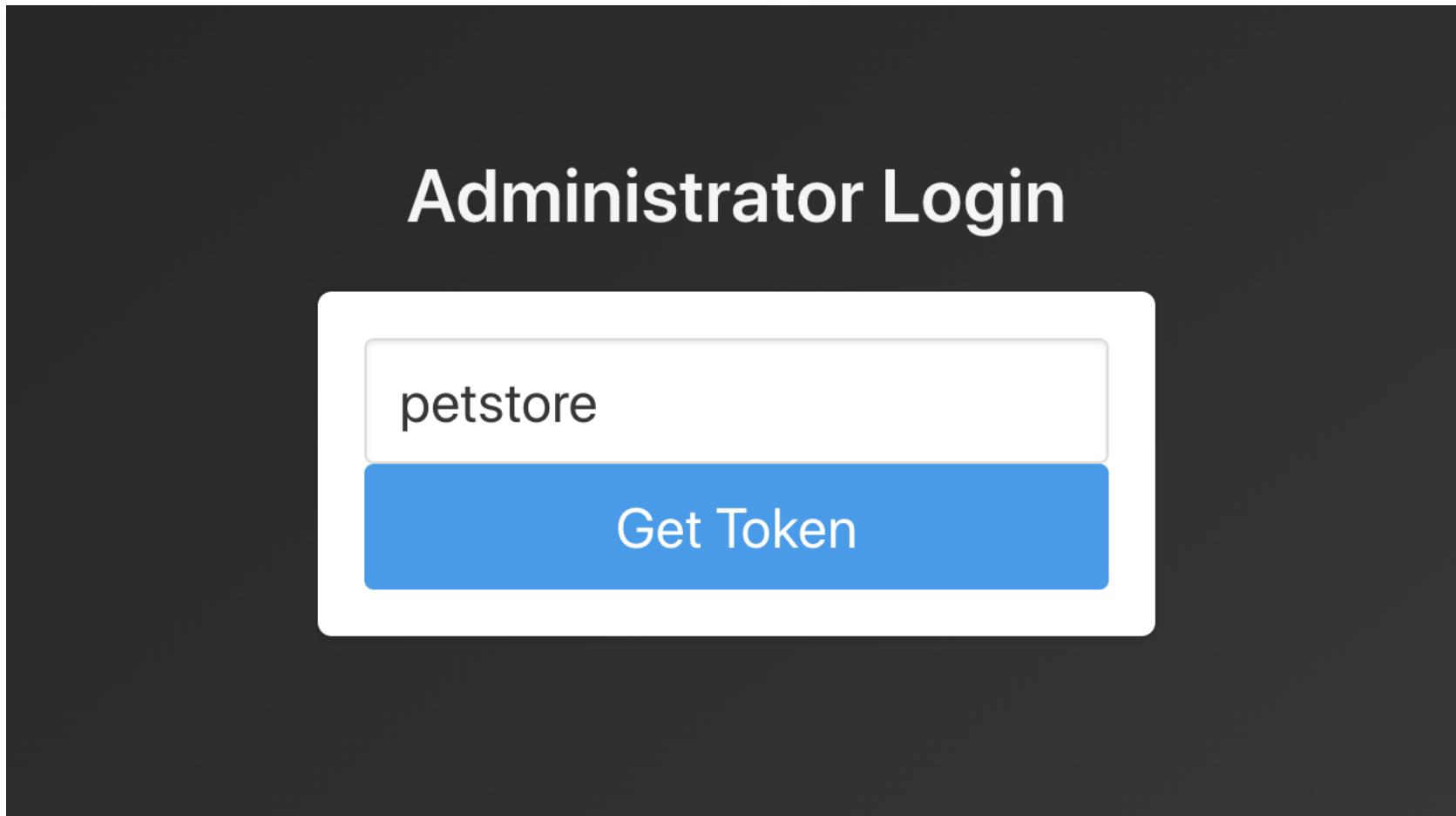
The credentials for the login page are

Username: administrator

Password: administrator

Note: Please note that the above credentials are created on the Barracuda WAF during the deployment of the frontend app

Oauth Token: Implicit grant flow OAUTH authentication is enabled for the /admin URL. Type a string, for example, "petstore" and click on Get Token to login to the admin portal.



Loading sample data

Once you are logged in to the admin portal, load the sample data by clicking on the option "Get Started by Loading Sample Data" in the landing page:

[Home](#) [Add Pet](#)

This is administrator space !

Hello World Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla accumsan, metus ultrices eleifend gravida, nulla nunc varius lectus, nec rutrum justo nibh eu lectus. Ut vulputate semper dui. Fusce erat odio, sollicitudin vel erat vel, interdum mattis neque. Second level Curabitur accumsan turpis pharetra augue tincidunt blandit. Quisque condimentum maximus mi, sit amet commodo arcu rutrum id. Proin pretium urna vel cursus venenatis. Suspendisse potenti. Etiam mattis sem rhoncus lacus dapibus facilisis. Donec at dignissim dui. Ut et neque nisl. In fermentum leo eu lectus mollis, quis dictum mi aliquet. Morbi eu nulla lobortis, lobortis est in, fringilla felis. Aliquam nec felis in sapien venenatis viverra fermentum nec lectus. Ut non enim metus.

Get Started by Loading sample data

Other Admin options:

Once the sample data is loaded, the administrator can add/remove/edit the data.

End User login

- The end user login allows Petstore clients to **signup to the portal and access the store inventory and complete an order.**

Note:

There is no end user configured by default. Complete the user registration page and signup to login to the application

Security Concerns / Vulnerabilities in PetStore

The frontend application has a few security concerns:

1. The site is not SSL enabled

This is overcome by enforcing SSL on the virtual service created on the Barracuda WAF. The certificate is a self signed certificate which can be easily replaced with a 3rd party certificate if required.

2. The admin portal is open and needs to be protected

This is overcome by enforcing form based authentication on the Barracuda WAF. Unauthorized access to the /admin portal can be prevented using this step.

Note: Both the above security concerns are handled during the application provisioning phase i.e. as part of the deployment of the Cloud Formation Template in Chapter 7.

9. Check the logs and fine tune the WAF settings

Barracuda WAF provides lots of features around logging and reporting.

Events related to HTTP traffic, actions of the Barracuda Web Application Firewall, and user actions are captured in logs. These log messages enable a system administrator to do the following:

- Obtain information about the Barracuda Web Application Firewall traffic and performance.
- Analyze logs for suspicious activity.
- Troubleshoot problems.

The following types of logs are available in the Barracuda Web Application Firewall:

- Web Firewall logs
- Access logs
- Audit logs
- System logs
- Network Firewall logs

Each log in **Web Firewall Logs**, **System Logs**, and **Network Firewall Logs** is associated with a log level that indicates the severity of the log. An administrator can configure the severity level based on the error messages/information that needs to be recorded in the logs. You can export the logs in .csv format and save the file to your desktop using **Generate CSV File** and **Download CSV File** options.

You can use the Web Firewall Logs to evaluate rule violations and, when warranted, to create exceptions to the rule violated. Exceptions can apply globally if they modify the security policy, which affects all services using that policy. Or, you can apply an exception locally that only applies to a specific website or URL.

Once logged in to the unit, select **Web Firewall Logs** from the **BASIC** tab to search for a log entry believed to be a false positive. These log entries are in red and have an action of DENY (active mode) or LOG (passive mode).

Time	Event Details	Client Details	Attack Details	Actions
↑ DENIED	URL /index.html	Service IP:Port 99.99.224.2:80 Client IP 99.99.48.1 Country US Service Name service1 Protocol HTTP Method GET	Attack Name Title in URL Path Attack Detail Rule security-policy	Fix Details
↑ DENIED	URL /index.html	Service IP:Port 99.99.224.2:80 Client IP 99.99.48.1 Country US Service Name service1 Protocol HTTP Method GET	Attack Name SQL Injection in Parameter Attack Detail type="sql-injection-medium" pat Rule security-policy	Fix Details
↑ LOGGED	URL /index.html	Service IP:Port 99.99.224.2:80 Client IP 99.99.48.1 Country US Service Name service1 Protocol HTTP Method GET	Attack Name SQL Injection in Parameter Attack Detail type="sql-injection-medium" pat Rule security-policy	Fix Details
↑ LOGGED	URL /index.html	Service IP:Port 99.99.224.2:80 Client IP 99.99.48.1 Country US Service Name service1 Protocol HTTP Method GET	Attack Name SQL Injection in Parameter Attack Detail type="sql-injection-medium" pat Rule security-policy	Fix Details

Scroll over to the right of the selected log and click **Fix**. A Policy Fix window appears.

The fix recommended by the Barracuda Web Application Firewall may be localized or global, depending upon which rule was violated. Accepting a recommendation can have the following impact:

1. **Web site profile (localized) modification:** As the most fine-grained security, changes impact only a given URL or parameter.
2. **Security Policy (global) modification:** As a policy shared by multiple applications, changes impact all applications using the security policy.

Examples of Fixes Suggested by the Barracuda Web Application Firewall

Example 1: Recommendation to Configure a Fine-Grained Rule.

Policy Fix

Cross-Site Scripting in Parameter

The parameter \$NONAME_PARAM, contained javascript:alert(attack); which is a Cross-Site Scripting pattern. This is a Blocked Attack type that is enabled in the Default Parameter Protection of the corresponding Security Policy, or in the Parameter Class of the matching Parameter Profile.

Recommended Fix

Create a new URL Profile for URL /index.html and Parameter Profile for parameter \$NONAME_PARAM and add script-in-tag-attribute to the Exception Patterns List of website service1.

Buttons: Apply Fix | Close Window

Following the recommendation to create a URL profile for /modules.php creates an exception only for that particular page.

Example 2: Recommendation to Change the Configuration in Security Policy.

Policy Fix

Metacharacter in Parameter

The parameter username contained %08, which is set as a Denied Metacharacter under Parameter Protection of the "default" Security Policy, or in the Parameter Class used by the Parameter Profile security-policy.

Recommended Fix

Modify Parameter Protection of "default" Security Policy by removing "%08" from the Denied Metacharacter List.

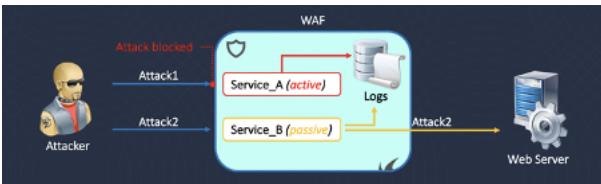
Buttons: Apply Fix | Close Window

The suggested change to the Parameter Protection sub-policy of the default Security Policy would allow the meta-character (%08) in any parameter for any application using this security policy. To avoid an exception that applies globally, you can add an exception that only applies to the URL or parameter noted in the log.

10. Change the Security Policy Mode

Now that the rules have been setup and the web firewall logs have been analysed, we can be sure that there will be no false positives, and so, the security policy mode can be changed to Active.

In Active mode, any requests that match rules on the Barracuda WAF will be blocked and a response will be sent back to the client.



To change the mode of the security policy, go to Basic -> Services and edit the service configuration and change the mode to "Active":

Service ?

Basic Security Help

Web Firewall Policy: Select a policy from the drop-down list to be used for this Service.

Web Firewall Log Level: Select the level for logging Web Firewall events. The lower level has higher priority and demands higher attention with lesser information logging.

Mode: **Passive** **Active**
*Passive: Logs the intrusions but allows traffic to pass through. Active: Logs and blocks the intrusions.
 Recommended: Initially set the Mode to 'Passive', and after fine-tuning the policies turn it to 'Active'.*

Trusted Hosts Action: Allow Passive Default
When set to Allow or Passive, all requests from trusted hosts, including those that are possible attacks, are ignored and passed through. Allow mode does not log events, whereas in Passive mode, events are logged. Set to Default if trusted hosts requests need no special handling.

Trusted Hosts Group:

Barracuda WAF has over 215 categories of attack rules for which a request may be blocked. Each rule category has an event ID which is logged in the Web Firewall Logs.

Based on the attack group, a response action is triggered. These response actions are configured in action policies page under the security policy.

11. Conclusion and Next Steps

Summary

Barracuda Web Application Firewall



Ensure Protection from
Web Attacks and DDoS



Block advanced Bots



Protect APIs and Mobile
Apps



Control Access and
Authentication



Secure App Delivery and
Increase Availability



Automate and
Orchestrate Security

Ensure Protection from Web Attacks and DDoS

The Barracuda Web Application Firewall protects applications, APIs, and mobile app backends against a variety of attacks including the OWASP Top 10, zero-day threats, data leakage, and application-layer denial of service (DoS) attacks. By combining both positive signature-based policies with robust anomaly detection capabilities, Barracuda WAF can defeat today's most sophisticated attacks targeting your web applications.

Barracuda Active DDoS Prevention—an add-on service for the Barracuda Web Application Firewall—filters out volumetric DDoS attacks before they ever reach your network and harm your apps. It also protects against sophisticated application DDoS attacks without the administrative and resource overhead of traditional solutions, to eliminate service outages while keeping costs manageable for organizations of all sizes.

Additional References

1. <https://www.barracuda.com/cap>
2. <https://campus.barracuda.com/>
3. <https://www.barracuda.com/demos>
4. <https://www.barracuda.com/products/webapplicationfirewall>
5. <https://campus.barracuda.com/product/webapplicationfirewall/api/#/>

Next Steps

Want to know more about WAF and WAF-as-a-Service?

- We can talk to your teams about Application Security
- We can help your teams with your evaluations and PoC's

Congratulations !! You have successfully deployed the Petstore in AWS and secured it using the Barracuda Web Application Firewall !