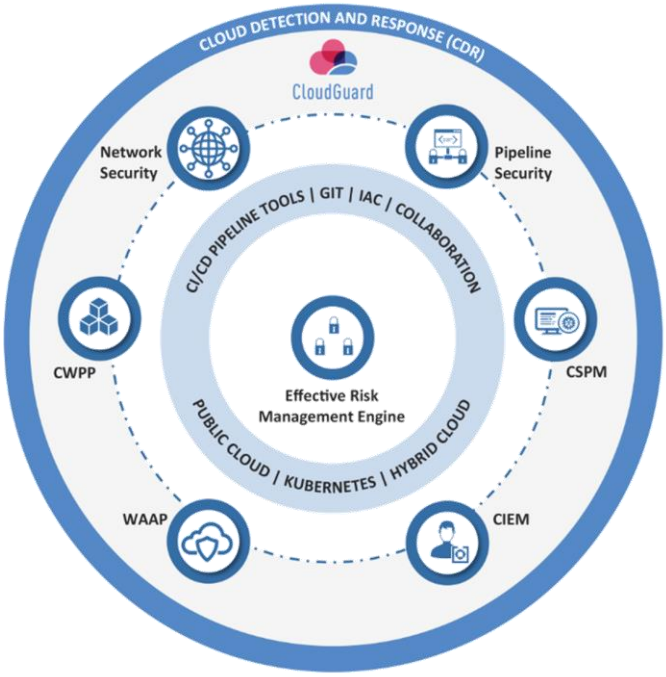
mind Cloud Program



Check Point CloudGuard WAF on AWS

Scaled and dynamically secured web services

ABSTRACT

Secure your AWS Cloud Infrastructure with CloudGuard WAF. Total Deployment time ~60 minutes.

Carlos Díaz

Peter Griekspoor

Version 3.0

Revision Control

Version 1.0	First release with CFT based on wiki demo guide and LATAM bootcamp	Peter Griekspoor	Aug. 29, 2024
Version 2.0	Adding detailed diagram and packet flow, and troubleshooting	Peter Griekspoor	Sept. 22, 2024
Version 3.0	Major Layout and textual changes	Peter Griekspoor	Oct. 15, 2024
Version 3.1	Changing screenshots for subscription plus comments	Peter Griekspoor	Oct. 15, 2024

Check Point CloudGuard – a CNAPP solution

Reading time: 5”

Cloud Native Application Protection Platform (CNAPP)

Check Point CloudGuard is a market leading CNAPP solution to unify cloud security with deeper security insights to prioritize risks and prevent critical attacks —providing more context, smarter security, faster—from code to cloud. Specifically, your customers will benefit from:

1. Deep security Intelligence:
   Use deep security intelligence across cloud workloads and users for greater insights
2. Prioritization remediation of critical risks:
   Focus on 1% of alerts that comprise 99% of critical cloud security risks.
3. Streamlined cloud security:
   Streamline cloud security across the SDLC and workloads to speed up risk reduction
4. True Prevention with WAF and Network Security
   Whatever doesn’t enter your Cloud doesn’t need to be managed.

Cloud Security Threat Landscape

Data breaches are one of the most significant threats facing cloud computing today, reports the Cloud Security Alliance. In 2023, it's predicted that cybercriminals will continue to target the cloud as a means of gaining access to sensitive information. This could include customer data, financial records, and proprietary business intelligence.

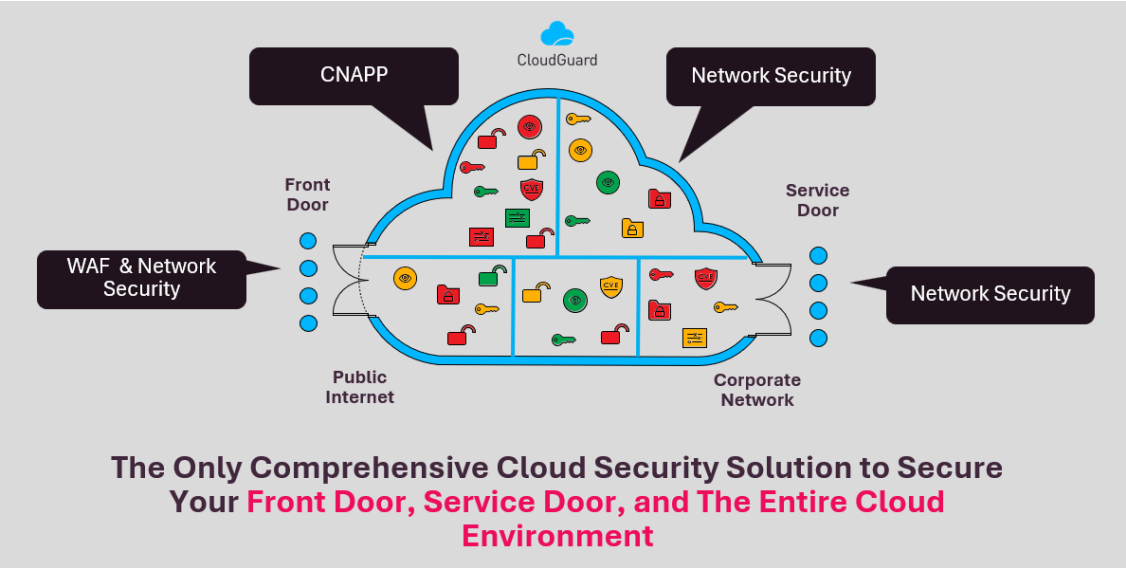


Figure 1 Check Point CloudGuard Security solution

AWS Bootcamp

In this Bootcamp we will demonstrate deployment of WAF agent, configuration and Dashboard with OWASP attacks.

CloudGuard WAF

Amongst the many features CloudGuard WAF offers, we include:

- OWASP TOP10 protection
- APIs protection, by validating their schemas and thus providing another layer of positive security that won't allow attackers to send malicious payloads.
- Web application protection: we can stop automated attacks by injecting a script into the client side and verifying human behavior.
- Check Point's Threat Cloud service to provide security for files and detect malware before they are uploaded to your application.

The Defense layers:

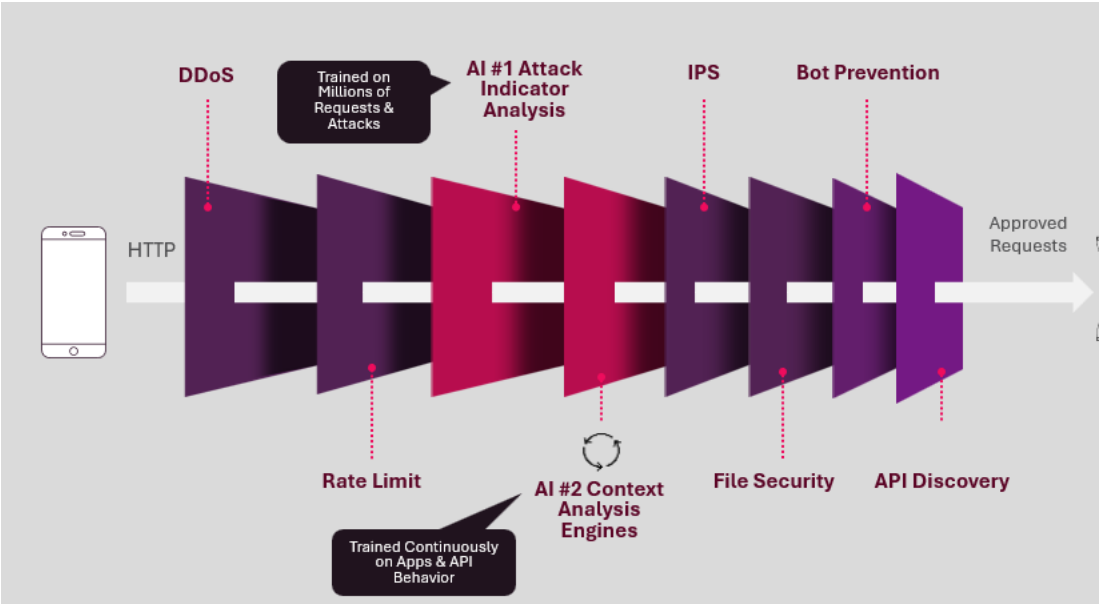


Figure 2 Check Point CloudGuard WAF Defense layers.

Contents

Section 1 – Lab Preparations

Time estimate: 15”

Objective:

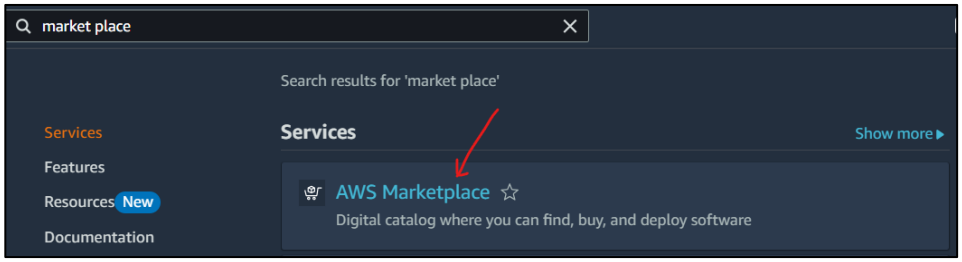
In this step you will make the required lab preparations:

1. Access your AWS Cloud Account
2. Validate subscriptions
3. Setup the AWS console
4. Diagram

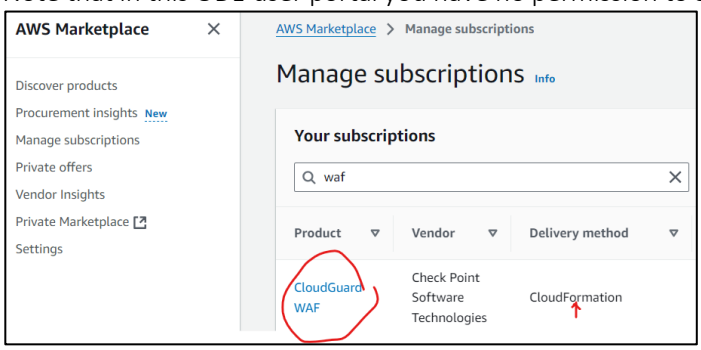
Validate Subscriptions

Before you can use any product from the Market Place you will need to subscribe to the product first.

1. Access your Cloud Account with the credentials provided.
2. Search for Market Place and open the Market Place



3. In Manage Subscriptions search for WAF, and confirm that WAF is subscribed and delivered through CFT. Note that in this ODL-user portal you have no permission to subscribe manually.



4. If WAF is not subscribed you cannot launch the WAF CFT, contact support at the portal owner Spektra: support@spektrasystems.com

Setup your AWS Console Prerequisites

In case you use your own account

- Permission to create an EC2 Instances (c5.large, t2.micro), NAT Gateway (not mandatory)
- Subscribe to WAF
- Linux Knowledge
- AWS VPC creation permissions
- Infinity portal account
- Basic OpenSSL knowledge

High Level Diagram

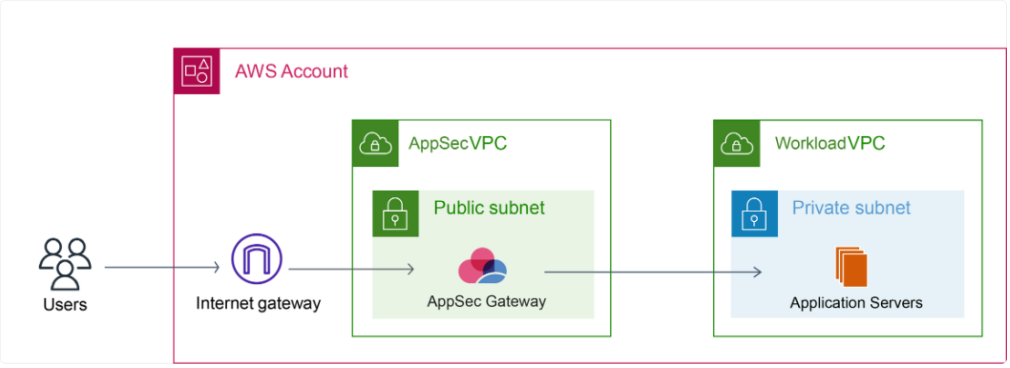


Figure 2: Quick Start Architecture for Check Point CloudGuard WAF on the AWS cloud.

Detailed Diagram

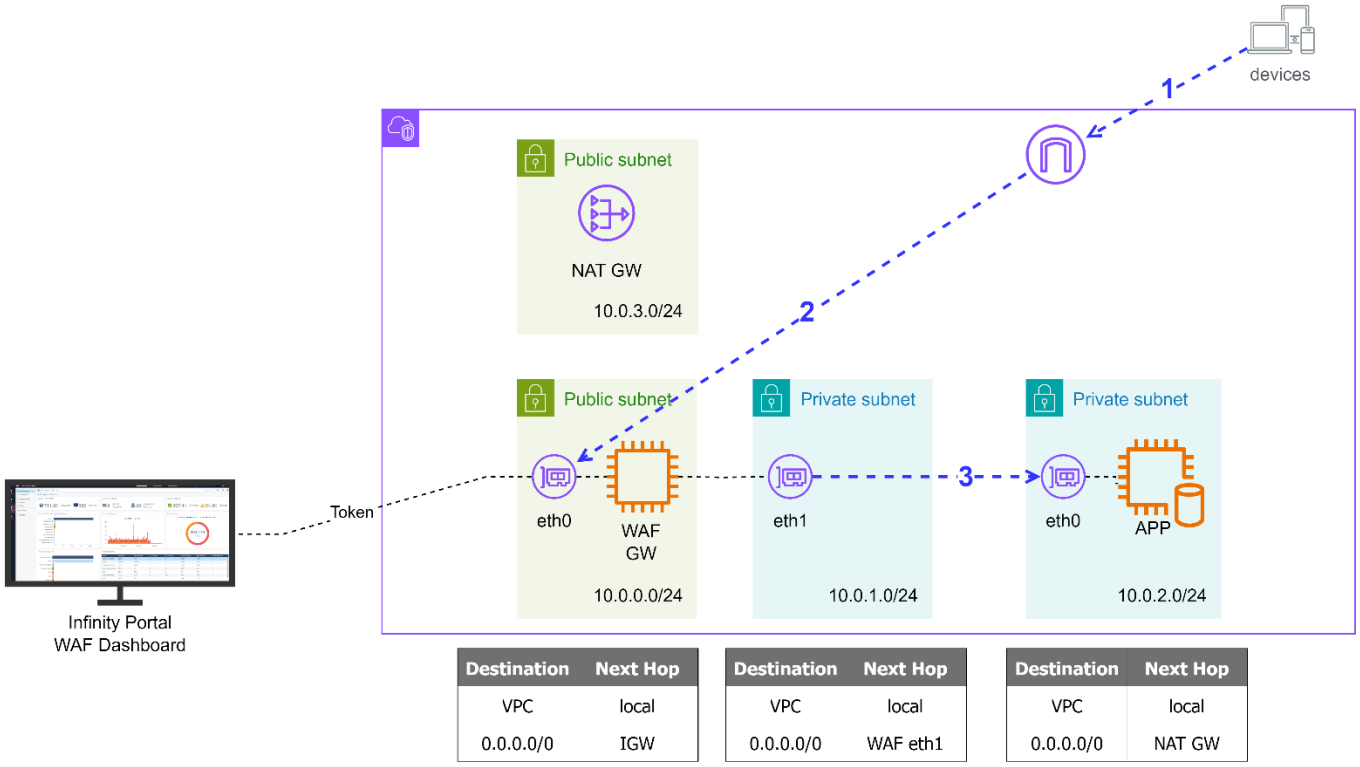


Figure 3: Low level diagram or lab architecture

WAF GW eth0	10.0.0.27
WAF GW eth1	10.0.1.220
APP eth0	10.0.2.220

Example values based on the lab settings:

Traffic Flow	source	destination
1	44.196.251.9 (PIP user)	44.196.251.9 (PIP WAF GW)
2	44.196.251.9 (PIP WAF GW)	10.0.0.27
3	10.0.1.220	10.0.2.220 (port 8000 or 3000)

Section 1 – Lab AWS Creation

Time estimate: 20”

Objective:
In this step you will configure all the network and application components for the training:

1. Create EC2 Key
2. Network Infrastructure with CFT: VPC, Subnets, Routing, NAT GW, Ubuntu server with Docker installed and website
3. Optional: Install an EC2 Connect Endpoint for Docker troubleshooting

Create Keypair

1. Loggin in the portal using the information provided by the trainer
2. The Lab is built and tested in **N. Virginia Region**
3. Download the CFT “Bootcamps-CloudGuard-WAF_AWS-1*.yaml” from here
4. Or go to github account getin2cloudnow, in repo WAF-Bootcamp

   https://github.com/getin2cloudnow/WAF-Bootcamp/blob/main/Bootcamps-CloudGuard-WAF_AWS-10.yaml

5. Open the EC2 Dashboard
6. From the left menu choose **Network & Security > Key Pairs**
7. Select **Create Key Pair** in upper right corner, i.e.: **wafkey**
   Keep all defaults and hit the **Create Key Pair** button to complete the key creation
8. Your **wafkey.ppk** file should be in your **download folder on your PC**.
   Don’t delete this key file as you can only download it once.

Launch Network Stack in AWS Portal

1. Open the Cloud Formation Dashboard and select the orange button **Create Stack**
2. In create stack, check the following radio buttons:
   - Choose and existing template
   - Upload a template file
   - Choose a file
   - Select the yaml file Bootcamps-CloudGuard-WAF_AWS-10
3. Select **Next** and enter stackname **WAFstack**
4. Enter the **EC2 Ubuntu 20.04 LTS image ID for N-Virginia:** default (**ami-0e86e20dae9224db8)**

5. Enter your EC2 Key name: **wafkey**
6. Select **Next twice**
7. Select **Submit**

It will take ~5 minutes for the stack to complete

PRIVATE IP ADDRESS OF DOCKER SERVER

1. Open the EC2 Dashboard
2. From the left menu choose **Instances > Instances**
3. Select your **DockerEC2Instance**
4. From the Instance summary copy the **Private IPv4 address** of your Docker Instance to your notepad. You will need this address to configure WAF reverse proxy

YOU HAVE SUCCESFULLY LAUNCHED YOUR APPLICATION!

!! The following steps are OPTIONAL, and only required if you wish to access the Docker machine for troubleshooting

Create EC2 Connect Endpoint

1. Open the VPC Dashboard
2. From the left menu, choose Virtual Private Cloud > Endpoints
3. Select Create Endpoint and name it: Dockeraccess
4. In Service Category choose EC2 Instance Connect Endpoint
5. In VPC, select your VPC: MyWAFVPC
6. In Security Group Select the security group from the WAFstack (allow all traffic)
7. In Subnet select DockerPrivateSubnet2
8. Select Create Endpoint
   It may take a few minutes for the endpoint to be available

Connect with your Docker Server

1. Open the EC2 Dashboard
2. From the left menu, choose **Instances > Instances**
3. Select your **DockerEC2Instance**
4. Select **Connect** from the upper EC2 menus
5. In **EC2 Instance Connect**, select radio button **Connect using EC2 Instance Connect Endpoint**
6. Choose **Connect** to connect with the Docker Server
7. A browser tab should open with the Ubuntu prompt
8. Run the following command to verify if your Docker is installed correctly:

```
sudo docker ps -a
```

You should see two ports for the Juice Shop (port 3000) and an appsec side with attacks (port 8000):

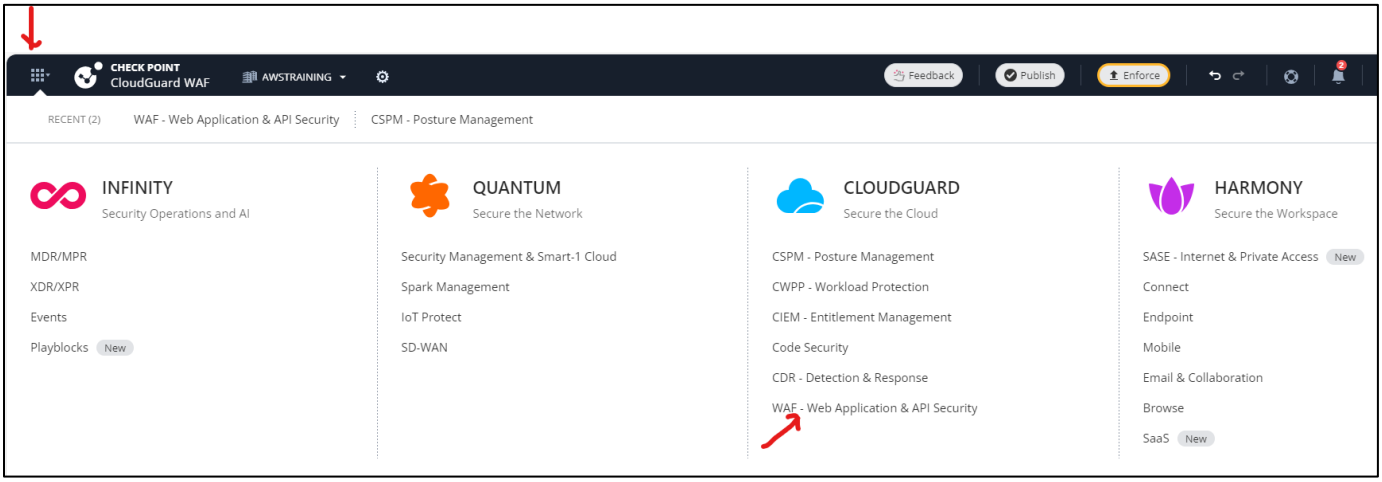
```
*** System restart required ***
Last login: Tue Aug 27 13:18:24 2024 from 10.0.2.37
ubuntu@ip-10-0-2-7:~$ sudo docker ps -a
CONTAINER ID   IMAGE                   COMMAND                  CREATED          STATUS          PORTS                                          NAMES
683adb37c197   appsecco/dsvw           "python /dsvw.py"        22 minutes ago   Up 22 minutes   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp      happy_spence
57f0106d4f6a   bkimminich/juice-shop   "/nodejs/bin/node /j…"   22 minutes ago   Up 22 minutes   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp      gallant_chaplygin
ubuntu@ip-10-0-2-7:~$
```

9. To test NAT GW routing for your private Docker server, run command: `ping 8.8.8.8`
   You should see ping replies.

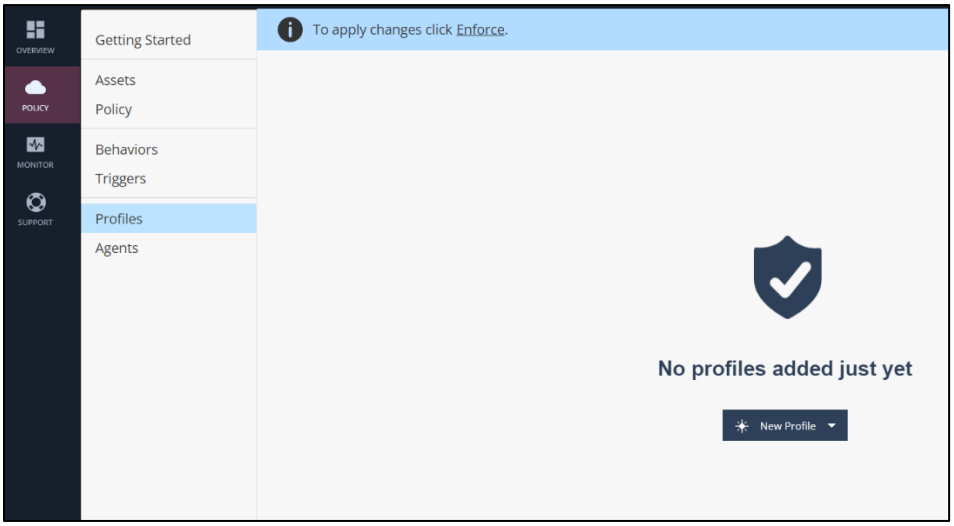
Section 2 – Infinity Portal Account Creation and Asset Creation

INFINITY PORTAL CREATION AND CONFIGURATION OF AN ASSET

1. Open a Web Browser and open https://portal.checkpoint.com/signin
2. Create an account if is necessary
3. Click of the grid icon in the left upper corner and go to **CloudGuard** section and **WAF- Web Application & API Security**.

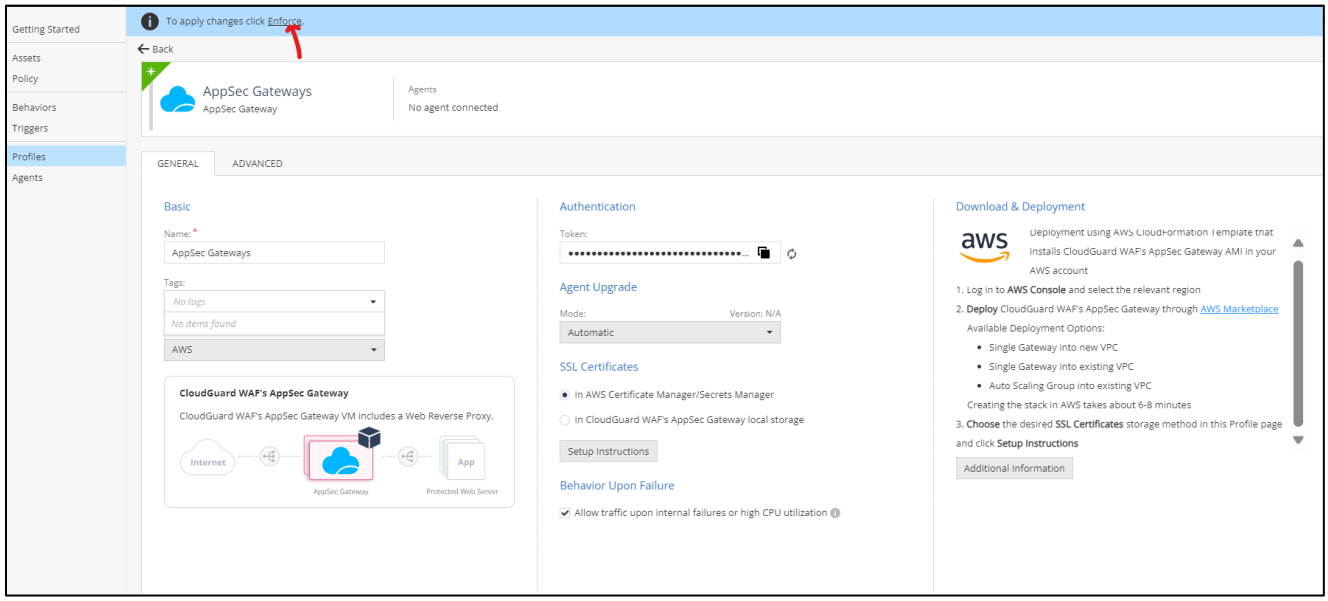


4. From the left menu, select policy and profiles add a new profile- **AppSec Gateway Profile**

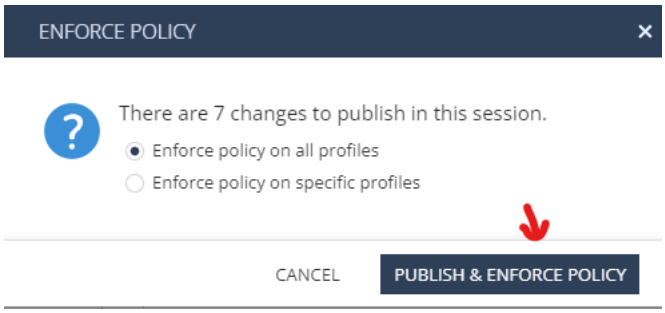


CONFIGURATION APPSEC GATEWAY PROFILE:

1. Name: **AppSec Gateways**
2. Environment: **AWS**
3. **Copy the secret Token** for portal authentication to your clipboard
4. Click on the **enforce** in the left upper corner

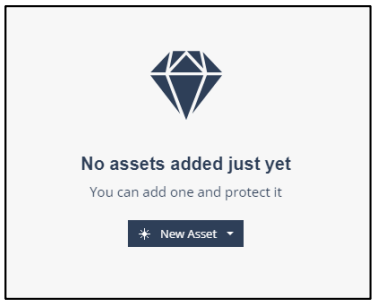


5. Confirm the radio button **Enforce policy on all profiles** is selected
6. Choose **Publish & Enforce Policy**



CONFIGURE THE JUICE SHOP WEB ASSET

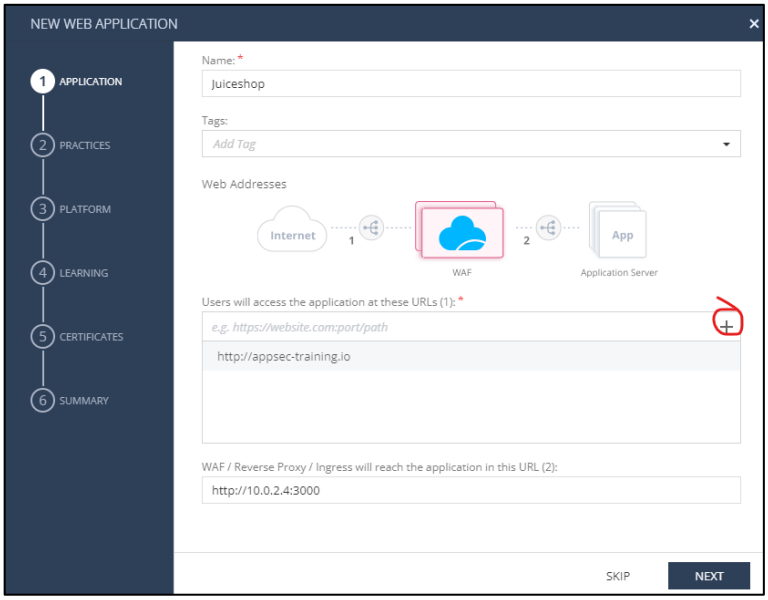
1. On the left menu Select **assets**



2. Choose **New Asset**
3. Choose **Web Application**
4. Configure the wizard:

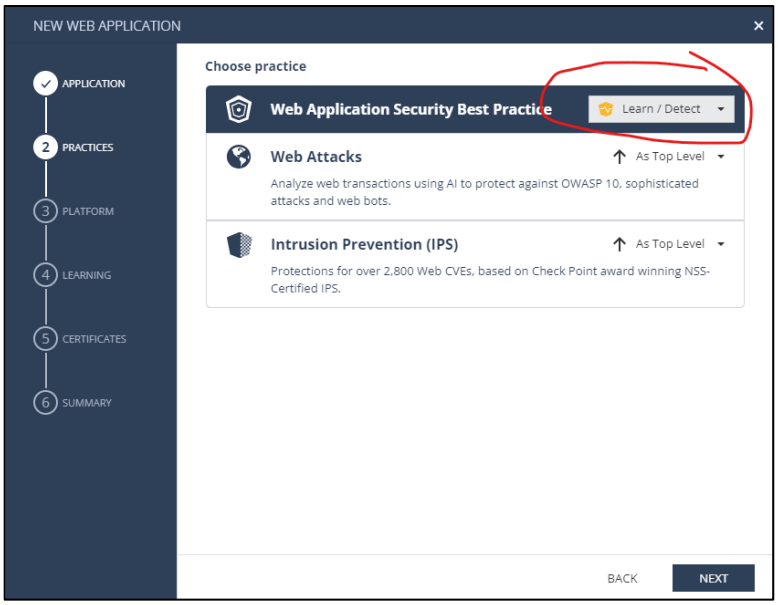
<u>Step 1 Application</u>

- Name: **Juiceshop**
- In field "**Users will access the application at these URLs (1)**", enter the URL for the application: http://appsec-training.io
  press the + button
- In field "**WAF / Reverse Proxy / Ingress will reach the application in this URL (2)**" enter the private IPv4 address of **your Docker instance you saved earlier**, i.e.: http://10.0.2.4:3000



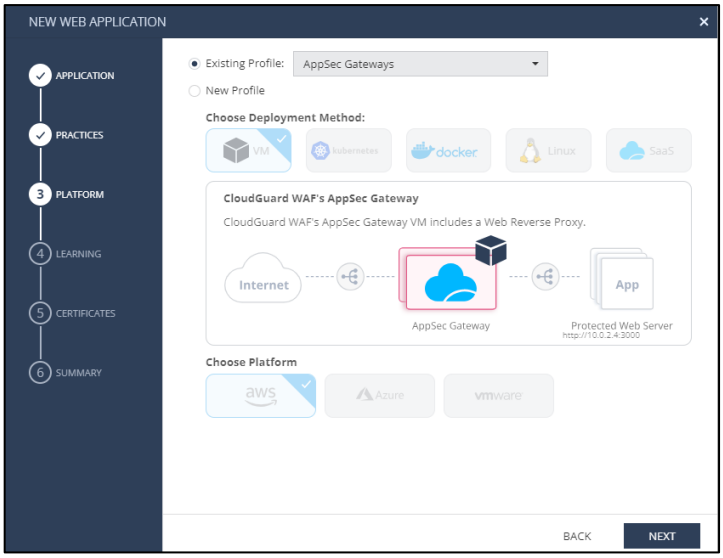
Step 2 Application

- In Practices, choose “Web Application Security Best Practice” and set in **Learn / Detect**
- **Choose Next**



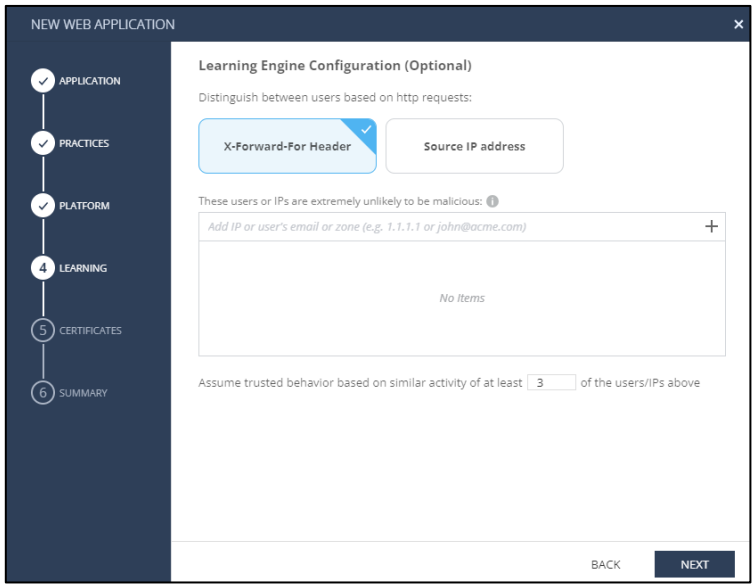
Step 3 Platform

- Choose existing profile, choose **AppSec Gateways**
- **Choose Next**



Step 4 Learning

- Choose **X-Forward-For Header**
- **Choose Next**

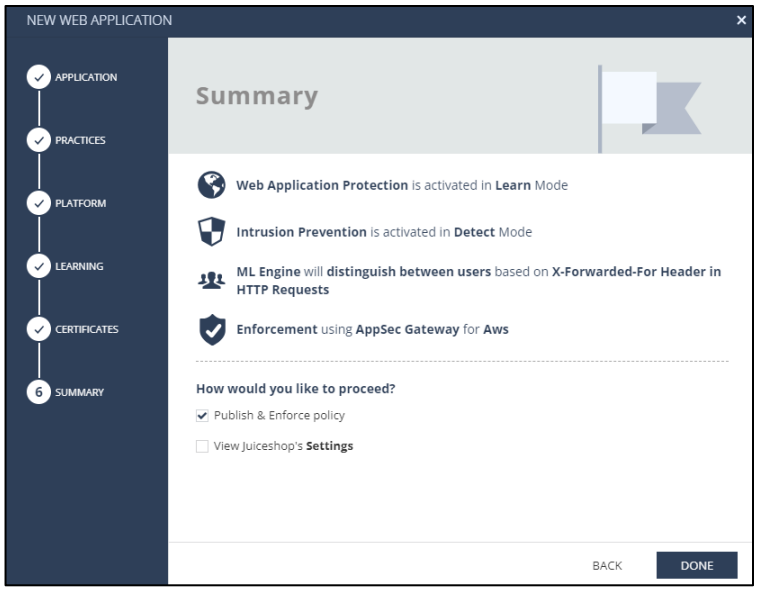


Step 5 Certificates

- **Choose Next**

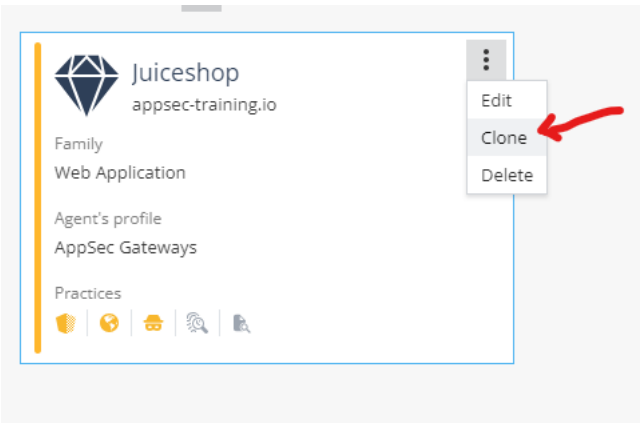
Step 6 Summary

- Verify the settings, confirm that the **Publish & Enforce Policy** box is checked.
- Click on **Done** to complete the wizard.

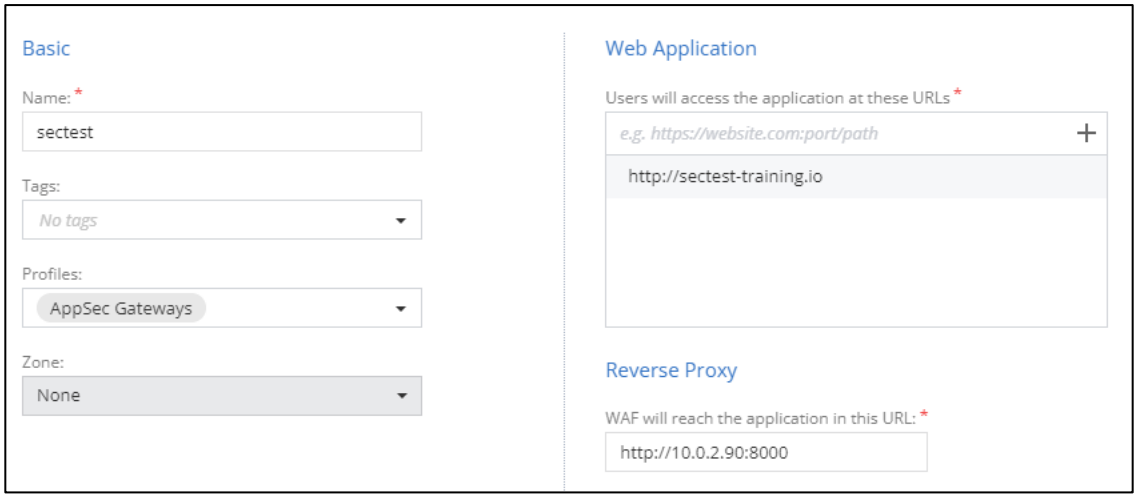


Configure the Malicious Web Asset

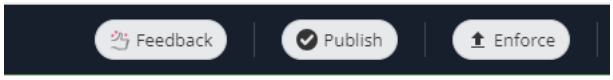
1. Go back to the Assets overview and clone the Juiceshop web asset as shown below:



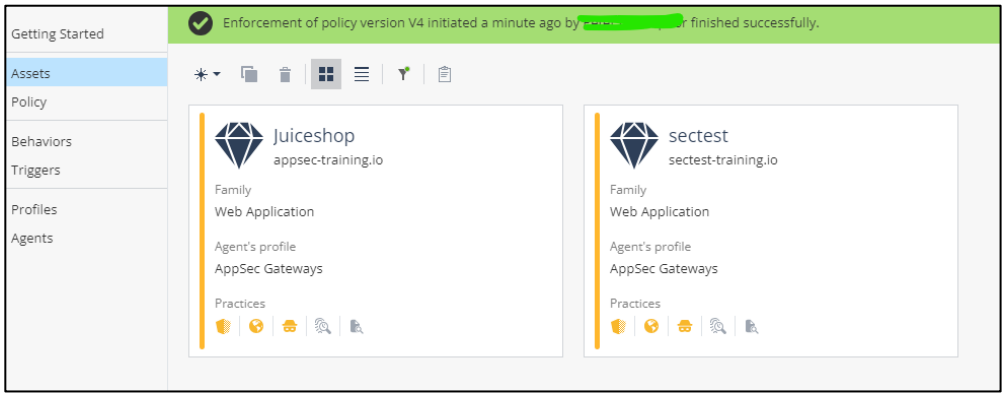
2. Choose Edit in the Asset you just created and change the name to **sectest**
3. Change **Users will access the application at these URLs** to http://sectest-training.io
4. Change **WAF will reach the application in this URL** to http://10.0.2.90:8000*
   *Change to <u>your</u> private IPv4 address of the Docker Server



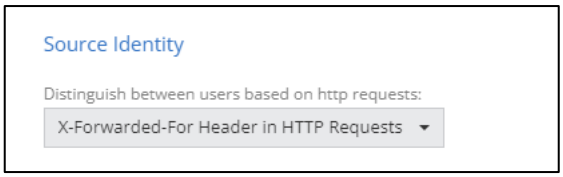
5. Publish and **Enforce** the Policy from the left upper corner or from the top menu.



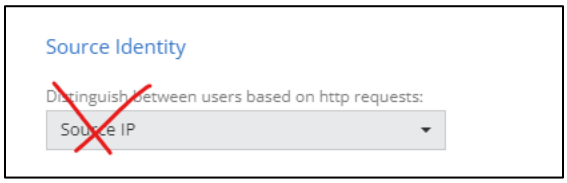
6. You have now successfully configured two assets:



7. Confirm that in your clone the Source Identity is set to **X-Forward-For Header in HTTP Requests**[1]



8. And not to Source IP:



[1] See also in common errors "The AppSec Gateways Reverse Proxy has stopped working unexpectedly."

Section 3 – Resource Deployments

DEPLOY WAF

1. Make sure you verified if you have a valid CloudGuard WAF subscription as explained at the start of this guide.
2. Open the https://support.checkpoint.com/results/sk/sk111013
3. Choose Deploys and configures a **CloudGuard Infinity Next Gateway into an existing VPC**.

CloudGuard WAF (formerly AppSec)

Description	Notes	CloudFormation Template	Direct Launch
Deploys and configures a CloudGuard Infinity Next Gateway	Creates a **new VPC** and deploys a CloudGuard Infinity Next Gateway into it.	AWS Marketplace	Launch Stack
	Deploys a CloudGuard Infinity Next Gateway into an **existing VPC**.	AWS Marketplace	Launch Stack

4. In the VPC dashboard, select **Virtual Private Cloud > Route tables > WAFInternalRouteTable**
5. Copy the **WAF Internal Route table ID** to your notepad, you will need it to configure the WAF CFT.
   Note: For this lab you can also skip this step.
6. Configure the CFT:

Stack Name	Default
VPC	Select the created WAF VPC
Public subnet	WAFPublicSubnet1
Private subnet	WAFPrivateSubnet1
Internal route table	Leave blank or <Your WAF internal Route Table ID>
EC2 Instance details - Gateway Name	Default
Instance Type	**c5.large**
Allow Access from	**0.0.0.0/0**
Key Name	The key your created earlier, i.e. **wafkey**
Assign public ip	**true**
Enable Instance Connect	**true**
Gateway's Password hash (optional)	Leave blank
Infinity Next Agent Token	<Your Infinity authentication Token> (Infinity portal: WAF >Policy > Profiles)
Fog address (optional)	Leave blank
Gateway's Hostname (optional)	Leave blank
Bootstrap script (optional)	Leave blank
IAM role - optional	Leave blank

7. **Check both acknowledgements** under capabilities and submit the stack
   The launch takes ~ 15 minutes

Configure static route in GW and hosts file

1. Open the EC2 Dashboard
2. From the left menu, choose **Instances > Instances**
3. Select the instance **Check-Point-Infinty-Next**
4. Copy the **Public IPv4** address to your notepad
5. Choose **Connect** from the right upper corner menu
6. In tab **EC2 Instance Connect**, confirm the radio button **Connect using EC2 Instance Connect** is selected.
7. Change the **username** to **admin**
8. Choose **Connect**
9. In the browser ssh session run the following clish commands:

```
clish
set static-route 10.0.2.0/24 nexthop gateway address 10.0.1.1 on
save config
show route
```

confirm the static route is added.

```
gw-c3d157> show route
Codes: C - Connected, S - Static, R - RIP, B - BGP (D - Default),
       O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA),
       A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed,
       U - Unreachable, i - Inactive

S               0.0.0.0/0           via 10.0.0.1, eth0, cost 0, age 201
C               10.0.0.0/24         is directly connected, eth0
C               10.0.1.0/24         is directly connected, eth1
S               10.0.2.0/24         via 10.0.1.1, eth1, cost 0, age 38
C               44.196.251.9/32     is directly connected, eth0
C               127.0.0.0/8         is directly connected, lo
gw-c3d157>
```

10. Modify the /etc/hosts in the PC to test the lab and use the URL **appsec-training.io** with the same public IP for both assets.

11. Open **notepad++ or other editor as administrator** in windows to open and save the hosts file

Operative System	Path
Windows (Open Notepad as Admin)	`C:\Windows\System32\drivers\etc\hosts`
Linux	/etc/hosts

12. Scroll to the bottom of the hosts file and add <u>your public ip</u> of the infinity gateway with the domain names:
    <Public IP> appsec-training.io
    <Public IP> sectest-training.io

Example:

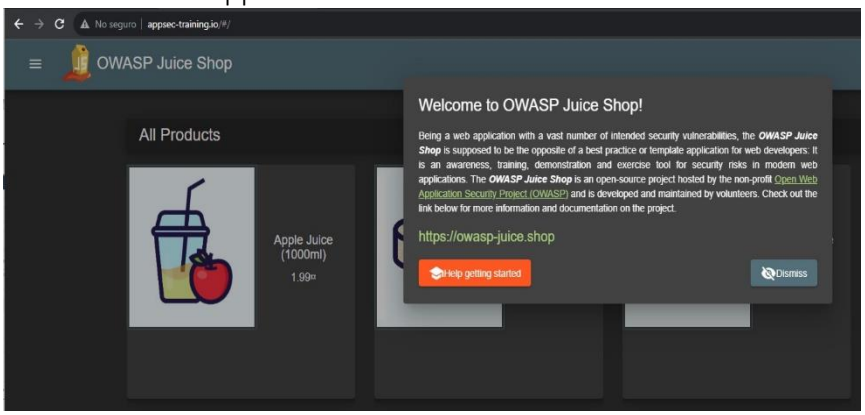
```
# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
54.225.64.241   appsec-training.io
54.225.64.241   sectest-training.io
```

Figure 4 - Public IP Address for WAF, every lab has a unique public IP address

Launch the Web Sites

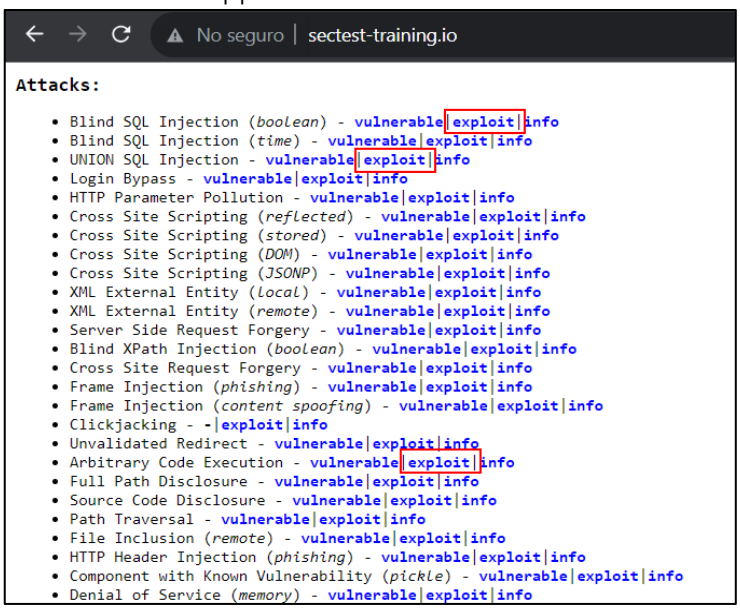
1. In your Web Browser, paste the following URL:
   http://appsec-training.io

   This site should appear:



2. In your Web Browser, paste the following URL:
   http://sectest-training.io

   This site should appear:



Section 4 – Attack Protection – SQL Injection

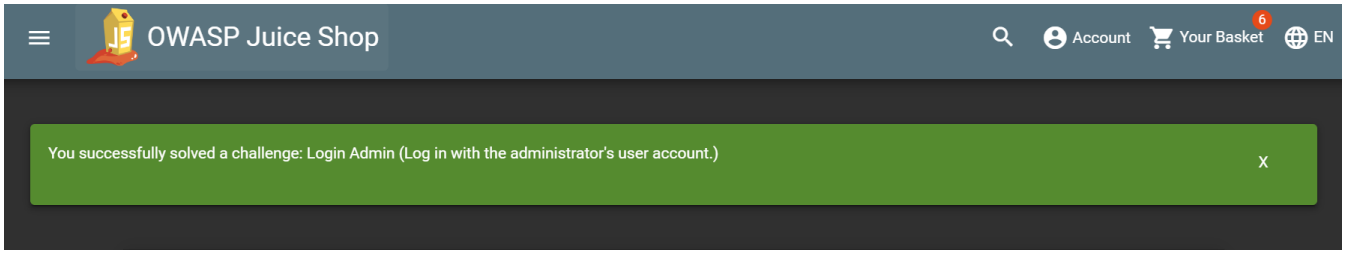
Juice Shop is susceptible to several attacks as SQL injection at the login, open a **browser using incognito** for http://appsec-training.io

The first phase is running the attack without protection, with CloudGuard WAF on **learn and detect**.

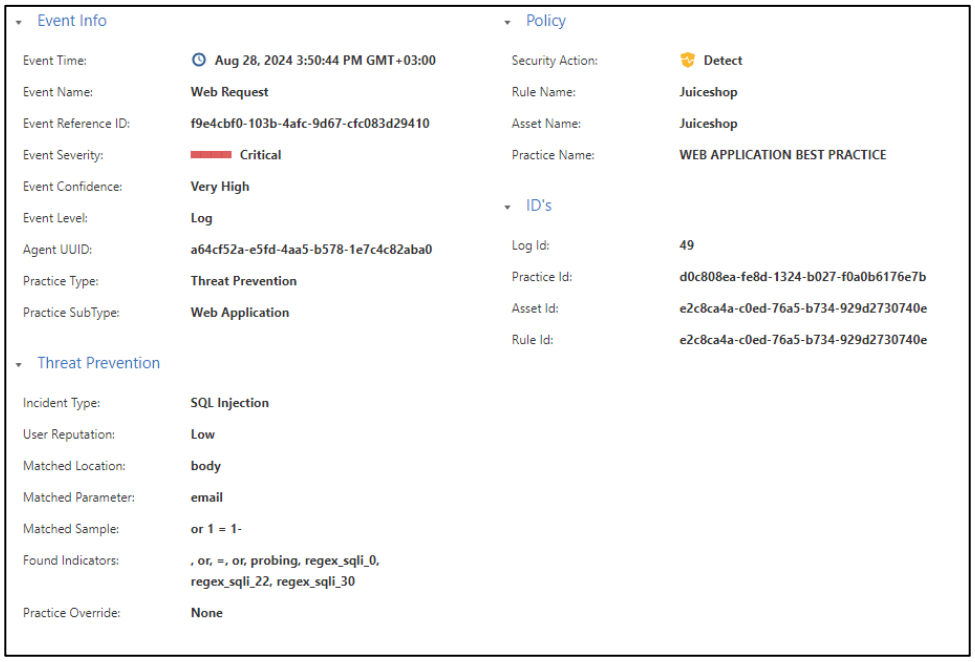
1. Open **the Juice shop** portal and go to account and in the login screen enter:
   Note: if copy-paste **'or 1=1--** in the email field doesn't work, type in manually

   Email : `'or 1=1--`
   Password : `abc123`

2. The attack should pass:



3. On WAF portal check the logs related to the attack Monitor and important events, and open a log. You should see information below:



Event Info

Event Time:	Aug 28, 2024 3:50:44 PM GMT+03:00
Event Name:	Web Request
Event Reference ID:	f9e4cbf0-103b-4afc-9d67-cfc083d29410
Event Severity:	Critical
Event Confidence:	Very High
Event Level:	Log
Agent UUID:	a64cf52a-e5fd-4aa5-b578-1e7c4c82aba0
Practice Type:	Threat Prevention
Practice SubType:	Web Application

Threat Prevention

Incident Type:	SQL Injection
User Reputation:	Low
Matched Location:	body
Matched Parameter:	email
Matched Sample:	or 1 = 1-
Found Indicators:	, or, =, or, probing, regex_sqli_0, regex_sqli_22, regex_sqli_30
Practice Override:	None

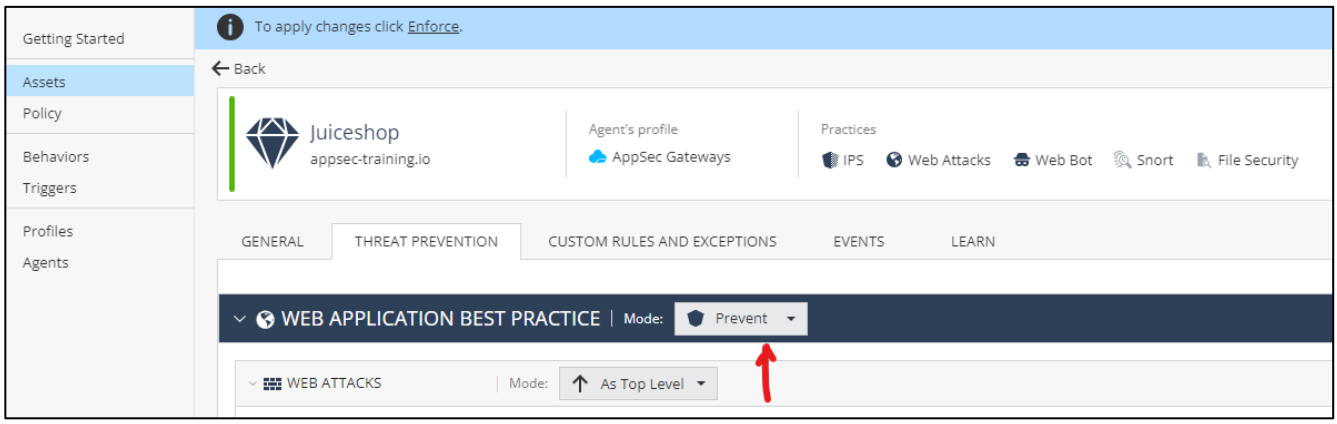
Policy

Security Action:	Detect
Rule Name:	Juiceshop
Asset Name:	Juiceshop
Practice Name:	WEB APPLICATION BEST PRACTICE

ID's

Log Id:	49
Practice Id:	d0c808ea-fe8d-1324-b027-f0a0b6176e7b
Asset Id:	e2c8ca4a-c0ed-76a5-b734-929d2730740e
Rule Id:	e2c8ca4a-c0ed-76a5-b734-929d2730740e

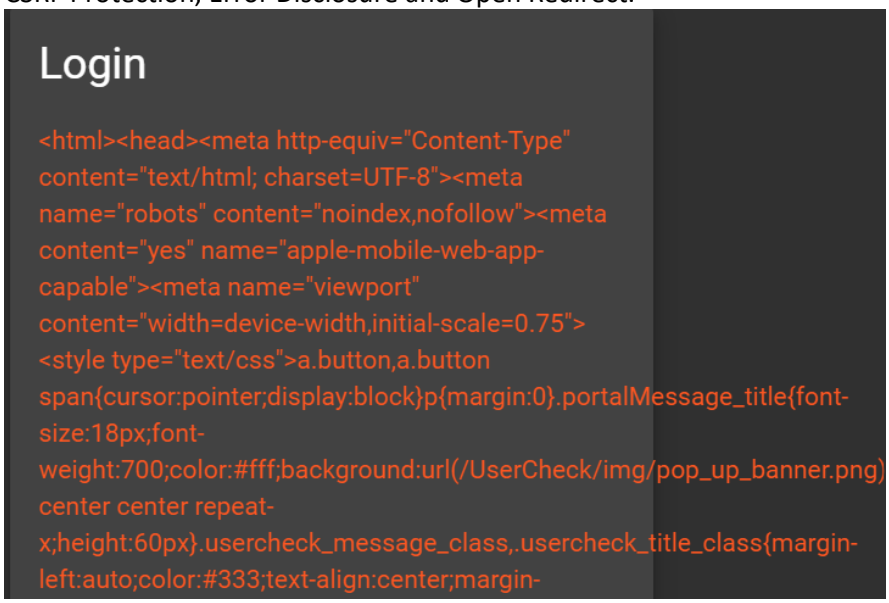
4. In the WAF portal, go to Policies > Assets and open the Juiceshop. Set mode to Prevent and enforce the policy:



5. Recreate the SQL injection in the juice shop (logout, and re-login with ‘or 1=1--)

6. The attack should now fail, with a popup like below :

CSRF Protection, Error Disclosure and Open Redirect.



7. Check the log again and confirm the **prevent** log.

Time	Event Severity	Asset Name	Security Action	Incident Type	Source Identifier
Aug 28, 2024 7:12:47 PM GMT+03:00	Critical	Juiceshop	Prevent	SQL Injection	77.137.79.16
Aug 28, 2024 7:12:09 PM GMT+03:00	Critical	Juiceshop	Detect	SQL Injection	77.137.79.16

8. In addition, DSVW has several vulnerabilities test, open a browser using incognito for http://sectest-training.io and exploit some tests

YOU HAVE SUCCESFULLY COMPLETED THIS LAB!

Section 5 – Deep Dive

The steps in this section assume that you had allowed "AWS EC2 connect" during the launch of the WAF GW CFT. If EC2 connect does not work, use PuTTY instead.

Debugging information at: https://waf-doc.inext.checkpoint.com/references/agent-cli#cpnano-command
DO NOT RUN DEBUGS IN A PRODUCTION ENVIRONMENT WITH TAC SUPPORT!

1. Open the EC2 Dashboard
2. From the left menu, choose **Instances > Instances**
3. Select the instance **Check-Point-Infinty-Next**
4. Choose **Connect** from the right upper corner menu
5. In tab **EC2 Instance Connect**, confirm the radio button **Connect using EC2 Instance Connect** is selected.
6. Change the **username** to **admin**
7. Choose **Connect**

   To check the status of the agent, run the following command: `cpnano -s`
   ('cpnano -help' for all options)

```
[Expert@gw-c3d157:0]# cpnano -s --e
---- Check Point Nano Agent ----
Version: 1.2430.970712
Status: Running
Last update attempt: 2024-08-29T07:38:39.467455
Last update: 2024-08-29T07:38:39.503127
Last update status: Succeeded
Policy version: 36
Last policy update: 2024-08-29T02:51:48.912186
Last manifest update: 2024-08-28T14:15:01.339585
Last settings update: 2024-08-29T02:51:48.912186
Registration status: Succeeded
Manifest status: Succeeded
Upgrade mode: automatic
Fog address: https://inext-agents.cloud.ngen.checkpoint.com
Agent ID: 8513d96f-d2b7-4936-b5a7-5c1642725f16
Profile ID: 8ac8cd00-13e9-e1bf-5922-1d57800f8a43
Tenant ID: a70598dd-308b-430e-b2ae-10c8507d01d3
Registration details:
    Name: gw-c3d157
    Type: AppSecGateway
    Platform: gaia
    Architecture: x86_64
```


8. To display the current policy, run following command: `cpnano -dp`

```
/etc/cp/conf/waap/waap.policy:
------------------------------
{
  "WAAP": {
    "WebAPISecurity": [],
    "WebApplicationSecurity": [
      {
        "antiBot": {
          "injected": [],
          "validated": []
        },
        "applicationUrls": "http://sectest-training.io",
        "assetId": "b8c8cd16-a7bb-90f6-0f2d-275880c4e01a",
        "assetName": "sectest",
        "botProtection": false,
        "botProtection_v2": "Detect",
        "context": "Any(All(Any(EqualHost(sectest-training.io)),EqualListeningPort(80)))",
        "csrfProtection": "Disabled",
        "errorDisclosure": "Disabled",
        "openRedirect": "Disabled",
        "overrides": [],
        "practiceAdvancedConfig": {
          "httpHeaderMaxSize": 102400,
          "httpIllegalMethodsAllowed": 0,
          "httpRequestBodyMaxSize": 1000000,
          "jsonMaxObjectDepth": 40,
          "urlMaxSize": 32768
```


Common Error's

Error:

The AppSec Gateways Reverse Proxy has stopped working unexpectedly.

During the cloning if the Juice shop asset, the source identity changes from **X-Forwarder-For Header in HTTP Requests** to **Source IP**:

Change Source Identity back to **X-Forwarder-For Header in HTTP Requests**