

Lab 1: Elevated Bot Defense¶

We started in the 141 class with a Relaxed or “Signature Only” Bot Defense Profile that did not include any active challenges. Now to build on that knowledge, and **Elevate** our security posture, we will create a Balanced Bot Mitigation Profile that includes active JavaScript challenges.

Balanced Mode Bot Profile Template Benefits

- Defines a moderate security policy that performs advanced verification of Browsers
- Blocks Malicious Bots that bypass signature checks
- Initiates a CAPTCHA challenge for Suspicious Browsers
- Strong verification of Mobile Apps using Anti-Bot Mobile Security SDK (Add-on required)
- Limits the total request rate produced by Unknown bots and allows Trusted and Untrusted Bots.
- Malicious Bots and Suspicious Browsers are identified by using both anomaly detection algorithms and bot signatures.
- This mode provides an advanced protection level with reduced latency impact because Browser verification is performed by injecting challenge in HTTP response.

	Relaxed	Balanced	Strict
Verification	Challenge-Free Verification	Verify After Access (Blocking)	Verify Before Access
Mitigation Settings	None	Generate After Access	Generate Before Access
Trusted Bot	Alarm	Alarm	Alarm
Untrusted Bot	Alarm	Alarm	Block
Suspicious Browser	Alarm	CAPTCHA	Block
Malicious Bot	Block	Block	Block
Unknown	None	Rate Limit	Block
DoS Attack Mitigation Mode	Disabled	Enabled	Enabled
API Access for Browsers and Mobile Applications	Disabled	Enabled	Enabled

(../../_images/prof_types.png)

- Estimated time for completion: **20 minutes**

Important

If you are continuing your lab session from 141 with the same deployment, please disable any previously configured security profiles on the Virtual Server and skip down to "Configuring Bot Defense". New students start at step 1.

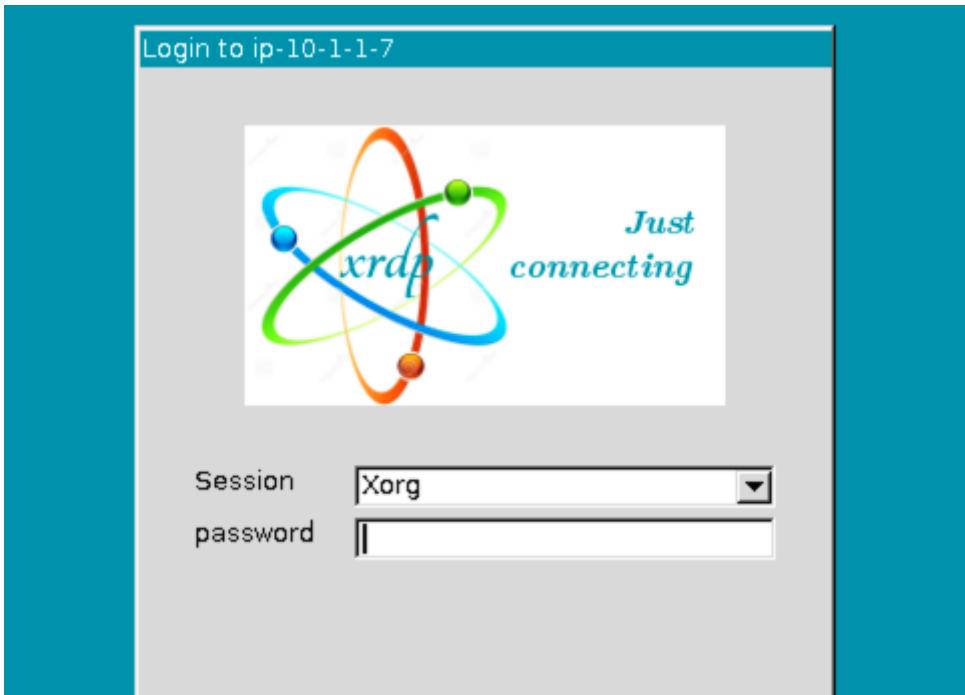
Local Traffic » Virtual Servers : Virtual Server List » **owasp-juiceshop_443_vs**

Policy Settings	
Destination	10.1.10.145:443
Service	HTTPS
Application Security Policy	Disabled
Service Policy	None
IP Intelligence	Disabled
DoS Protection Profile	Disabled
Bot Defense Profile	Disabled
Application Cloud Security Services	Disabled
DataSafe Profile	Disabled
Log Profile	Disabled

Update

(../../../../_images/blank_vs1.png)

1. RDP to the Linux Client by choosing the RDP access method from your UDF environment page. You will be presented with the following prompt where you will enter the password only. The **f5student** account is hard-coded into XRDP for your convenience.



(../../../../_images/xrdp1.png)

2. Once logged in, launch Chrome Browser. You can double-click the icon or right click and choose execute but **do not click multiple times**. It does take a few moments for the browser to launch the first time.
3. Click the **F5 Advanced WAF bookmark** and login to TMUI. admin/[password].
4. On the Main tab, click **Local Traffic > Virtual Servers** and you will see the Virtual Servers that have been pre-configured for your lab. Essentially, these are the listening IP's that receive requests for your application and proxy the requests to the backend "real" servers.

You will see 3 Virtual Servers:

	Status	Name	Description	Application	Destination	Service Port	Type
<input checked="" type="checkbox"/>	■	juiceshop-test.f5agility.com	for sending spoofed traffic to j...		10.1.10.146	443 (HTTPS)	Standard
<input checked="" type="checkbox"/>	■	owasp-juiceshop_443_vs	main juiceshop site		10.1.10.145	443 (HTTPS)	Standard
<input checked="" type="checkbox"/>	■	owasp-juiceshop_80_vs	standard redirect to owasp-juicesho...		10.1.10.145	80 (HTTP)	Standard

(../../../../_images/virtual_servers1.png)

- * **juiceshop-test.f5agility.com** - Will be used later to send spoofed traffic to the main site
- * **owasp-juiceshop_443_vs** - Main Site - Status of green indicates a healthy backend pool of real servers
- * **owasp-juiceshop_80_vs** - Standard port 80 redirect to main site

Configuring Bot Defense¶

The first step in enabling Bot Defense is to set up the log profile so we can capture all of the events we need to see. We will then create and apply the Bot Defense profile to our Juice Shop Virtual Server.

1. Navigate to **Security > Event Logs > Logging Profiles** and click **Create** to setup a new Logging Profile named: **Balanced_Bot_Log**.
2. Configure the profile per the screenshot below and when finished click **Create**. You may need to resize the browser for the “Create” button to be visible.

Note

Initially, we are logging everything so we get can a feel for traffic patterns where normally in the “real world” you would scale this back to log only essential requirements and not necessarily valid human or mobile devices.

The screenshot shows the 'Edit Logging Profile' dialog for a 'Balanced_Bot_Log'. The 'Logging Profile Properties' section includes fields for Profile Name (Balanced_Bot_Log), Partition / Path (Common), Description (empty), and various security features like Application Security, Protocol Security, Network Firewall, DoS Protection, and Bot Defense (which is checked). The 'Bot Defense' tab is selected, showing settings for Request Log, Log Requests by Classification (Human Users, Bots, Unknown), Log Requests by Mitigation Action (None, Alarm, CAPTCHA, Rate Limit, Block, TCP Reset, Honeypot Page, Redirect to Pool), Log Requests by Browser Verification Action, Log Device ID Collection Request, and Log Challenge Failure Requests. Both the 'Request Log' and 'Log Requests by Classification' sections have their 'Enabled' checkboxes checked.

Logging Profile Properties	
Profile Name	Balanced_Bot_Log
Partition / Path	Common
Description	
Application Security	<input type="checkbox"/> Enabled
Protocol Security	<input type="checkbox"/> Enabled
Network Firewall	<input type="checkbox"/> Enabled
DoS Protection	<input type="checkbox"/> Enabled
Bot Defense	<input checked="" type="checkbox"/> Enabled
Data Protection	<input type="checkbox"/> Enabled

Bot Defense	
Request Log	
Local Publisher	<input checked="" type="checkbox"/> Enabled
Remote Publisher	none
Log Requests by Classification	<p>Human Users: <input checked="" type="checkbox"/> Browser <input checked="" type="checkbox"/> Mobile App</p> <p>Bots: <input checked="" type="checkbox"/> Trusted Bot <input checked="" type="checkbox"/> Untrusted Bot <input checked="" type="checkbox"/> Suspicious Browser <input checked="" type="checkbox"/> Malicious Bot</p> <p>Unknown: <input checked="" type="checkbox"/> Enabled</p>
Log Requests by Mitigation Action	<input checked="" type="checkbox"/> None <input checked="" type="checkbox"/> Alarm <input checked="" type="checkbox"/> CAPTCHA <input checked="" type="checkbox"/> Rate Limit <input checked="" type="checkbox"/> Block <input checked="" type="checkbox"/> TCP Reset <input checked="" type="checkbox"/> Honeypot Page <input checked="" type="checkbox"/> Redirect to Pool
Log Requests by Browser Verification Action	<input checked="" type="checkbox"/> Enabled
Log Device ID Collection Request	<input checked="" type="checkbox"/> Enabled
Log Challenge Failure Requests	<input checked="" type="checkbox"/> Enabled

(../../../../_images/balanced_bot.png)

3. Navigate to **Security > Bot Defense > Bot Defense Profiles** and click **Create**.
4. Name: **Balanced_Bot_Profile**

5. Enforcement Mode: **Blocking** (If the enforcement mode is set to Transparent, browser verification challenges are not performed.)

6. Profile Template: **Balanced**

7. Click the **Learn more** link to see an explanation of the options.

The screenshot shows the 'Create New Bot Profile' page. At the top, there are 'Save' and 'Cancel' buttons, and a note: 'Note: Click Save to retain any changes you made in this profile.' On the left, a sidebar titled 'Bot Profile Configuration' lists several tabs: General Settings (selected), Bot Mitigation Settings, Microservice Protection, Browsers, Mobile Applications, Signature Enforcement, and Whitelist. The main area contains the following configuration:

- Profile Name ***: Balanced_Bot_Profile
- Partition/Path:** Common
- Description**: (empty text area)
- Enforcement Mode**: Transparent (radio button) is selected, while Blocking (radio button) is checked.
- Profile Template**: Balanced (dropdown menu, with 'Learn more' link)
- Signature Staging upon Update**: Disabled (button)
- Enforcement Readiness Period**: 7 days
- Redirect to Pool**: None (dropdown menu)
- Response and Blocking Pages** section:
 - First CAPTCHA Response**: Default (button)
 - Failure CAPTCHA Response**: Default (button)
 - Blocking Page Response**: Default (button)

(../../../../_images/bot_prof.png)

8. Click on the **Bot Mitigation Settings** tab and review the default Mitigation Settings for various classifications of bots and browsers. We will see these settings in action shortly.

9. Click on the **Browsers** tab and under **Browser Verification** and note the settings as well as the setting for **Device ID Mode**.

- The grace period allows web pages (including complex pages such as those which include images, JS, and CSS) the time to be recognized as non-bots, receive validation, and completely load without unnecessarily dropping requests.
- The grace period begins after the client is validated, a configuration change occurs, or when proactive bot defense starts as a result of a detected DoS attack or high latency.

10. Click on the **Help** tab at the top left of the screen and scroll down to the **Browsers > Browser Verification** section for a more detailed explanation of each of the settings for **Browser Verification**.

Note

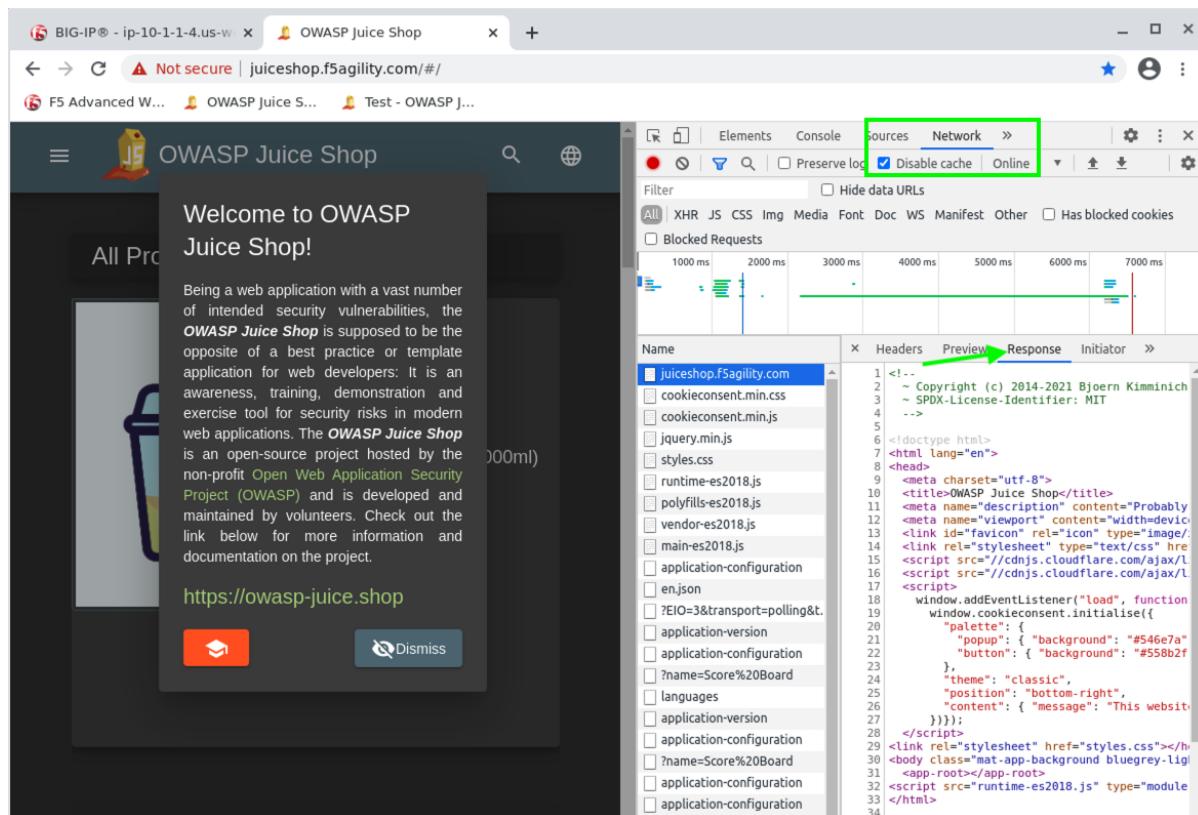
It is important to understand what these settings are capable of and how they operate. Inline help is always a great option for more information!

Verify after Access (Blocking): JavaScript is injected in the response. The JavaScript performs browser verification tests. If the tests fail, browser verification anomalies are reported and the mitigation is performed according to the selected mitigation settings. If the tests pass, the request is passed to the server.

11. Click **Save**.

Verify Normal HTTP Response¶

1. Open a new Tab in Google Chrome and **Right Click** anywhere on the page and choose **Inspect** from the menu. Click on the **Network Tab** and click **Disable cache** as shown in the screenshot below step 4.
2. Click the **OWASP Juice Shop Bookmark** in the Bookmark menu and click through the certificate warning. **DO NOT Dismiss the popup or click on anything else yet**. You will see a lot of output under the Network Tab Log as the page loads. Scroll to the top of this log until you find the entry named **juiceshop.f5agility.com** and click on it.
3. Click on the **Response** tab and note the default HTML response when no Bot Profile is applied to the Virtual Server. This is the normal Juice Shop web page HTML.
4. Do not close this tab.



The screenshot shows a Google Chrome window with the Network tab selected in the developer tools. The 'Disable cache' checkbox is checked. An arrow points to the 'Response' tab in the Network tab header. The response body displays the HTML code for the OWASP Juice Shop homepage, starting with the doctype declaration and various script and style tags.

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>OWASP Juice Shop</title>
6     <meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0">
7     <meta name="description" content="Probably the most secure and feature-rich open source web application for security professionals. It's designed to be a complete testbed for security researchers and a great learning tool for students and hobbyists. The OWASP Juice Shop is a web application with a vast number of intended security vulnerabilities, the OWASP Juice Shop is supposed to be the opposite of a best practice or template application for web developers: It is an awareness, training, demonstration and exercise tool for security risks in modern web applications. The OWASP Juice Shop is an open-source project hosted by the non-profit Open Web Application Security Project (OWASP) and is developed and maintained by volunteers. Check out the link below for more information and documentation on the project.">
8     <link id="favicon" rel="icon" type="image/png" href="/juice.png">
9     <link rel="stylesheet" type="text/css" href="/styles.css">
10    <script src="//cdnjs.cloudflare.com/ajax/libs/react/16.13.1/umd/react.development.js">
11    <script src="//cdnjs.cloudflare.com/ajax/libs/react-dom/16.13.1/umd/react-dom.development.js">
12    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.min.js">
13    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
14    <script src="https://unpkg.com/react-transition-group@4.3.2/dist/react-transition-group.min.js">
15    <script src="https://unpkg.com/react-transition-group@4.3.2/dist/react-transition-group.development.js">
16    <script src="https://unpkg.com/react-router-link@5.1.2/dist/react-router-link.min.js">
17    <script src="https://unpkg.com/react-router-link@5.1.2/dist/react-router-link.development.js">
18    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.min.js">
19    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
20    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
21    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
22    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
23    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
24    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
25    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
26    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
27    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
28    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
29    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
30    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
31    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
32    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
33    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">
34    <script src="https://unpkg.com/react-router-dom@5.1.2/dist/react-router-dom.development.js">

```

(../../../../_images/juice.png)

Applying Bot Defense¶

1. In Advanced WAF tab, click the **Main** tab and navigate to **Local Traffic > Virtual Servers > owasp-juiceshop_443_vs > Security > Policies**
2. Enable the Bot Defense Profile and select the **Balanced_Bot_Profile**.

3. Enable the Log Profile and select the **Balanced_Bot_Log** profile.

4. Click **Update**

The screenshot shows the 'Local Traffic > Virtual Servers : Virtual Server List > owasp-juiceshop_443_vs' screen. The 'Security' tab is selected. In the 'Policy Settings' section, the 'Log Profile' dropdown is set to 'Enabled...' and has 'Balanced_Bot_Profile' selected. Below this, two dropdown menus show available log profiles: 'Selected' contains '/Common/Balanced_Bot_Log', and 'Available' contains '/Common/Log all requests', '/Common/Log illegal requests', '/Common/global-network', and '/Common/local-bot-defense'. At the bottom left is a 'Update' button.

(../../../../_images/vs.png)

Verify Browser Challenges¶

1. Back in the JuiceShop tab, click the Browsers **Refresh** button. **Do not dismiss the popup or interact with the site in any way.** (Inspection tools should still be open and focused on the Network tab)
2. Find the 1st entry named **juiceshop.f5agility.com** at the top and click on it. There will be two. The top one is empty (Failed to load response data) because there was none, but if you look at the headers you can see this is actually a 307 temp redirect back to "/" with 2 **TS** cookies set by the WAF. The **TSPD_101** cookie is the one set as part of the challenge. This was the first phase of the Active challenge and similar in a way to how our TCP SYN cookies work at Layer 4.

The screenshot shows the OWASP Juice Shop application running in a browser. The page displays a welcome message and a product list. A cookie consent banner at the bottom states: "This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!". A "Me want it!" button is present. The browser's developer tools Network tab is open, showing a request to "juiceshop.f5agility.com" with a status code of 307 Temporary Redirect. The response headers include "Content-Type: text/html" and "Location: /". The response body contains obfuscated JavaScript code for setting cookies, including "Set-Cookie: TSPD_101_R0=082a7ad0c3ab20002235d4bc9b11f5ea9000443eb8f7cb3f95f134382925ded43901e6bc9f2870868ff84fe1430004dfa0de82ba8f6c8072bf3fb7547f280044363453048c19f43ebf7bac6fc7b0912aaacabd144acb8c48c0df0fccadb4a; Max-Age=5; Path=/", "Set-Cookie: T5c97ac518027=082a7ad0c3ab2000fd04e6ead21044a670af7f01fd2ea9d8aa39b4d51d4a8a0a8bf3317fd8a1a108d93e4a3f11300897fac974f0fda34c7f82a597d9e0b0bc805637d6ccbca6921f7c26a4a6bb0c11d33c762006f6f41c4b62a61699cc2; Path=/", and "X-Content-Type-Options: nosniff", "X-Frame-Options: SAMEORIGIN", "X-XSS-Protection: 1; mode=block".

(../../../../_images/first.png)

- Under the second request for **juiceshop.f5agility.com** you will see quite a different HTML response this time as the Advanced WAF has inserted obfuscated JS to challenge and verify the browser.
- You may need to resize the Inspect > Response pane to get a better look at the JS. This code is not easy to reverse engineer and is updated often via the Advanced WAF **Live Update** feature.

The screenshot shows the OWASP Juice Shop application running in a browser. The page displays a product list. A cookie consent banner at the bottom states: "This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!". A "Me want it!" button is present. The browser's developer tools Network tab is open, showing a request to "juiceshop.f5agility.com" with a status code of 200 OK. The response headers include "Content-Type: text/html" and "Cache-Control: no-store, must-revalidate, no-cache, max-age=0". The response body contains obfuscated JavaScript code, including "Set-Cookie: TSPD_101_R0=082a7ad0c3ab20002235d4bc9b11f5ea9000443eb8f7cb3f95f134382925ded43901e6bc9f2870868ff84fe1430004dfa0de82ba8f6c8072bf3fb7547f280044363453048c19f43ebf7bac6fc7b0912aaacabd144acb8c48c0df0fccadb4a; Max-Age=5; Path=/", "Set-Cookie: T5c97ac518027=082a7ad0c3ab2000fd04e6ead21044a670af7f01fd2ea9d8aa39b4d51d4a8a0a8bf3317fd8a1a108d93e4a3f11300897fac974f0fda34c7f82a597d9e0b0bc805637d6ccbca6921f7c26a4a6bb0c11d33c762006f6f41c4b62a61699cc2; Path=/", and "X-Content-Type-Options: nosniff", "X-Frame-Options: SAMEORIGIN", "X-XSS-Protection: 1; mode=block".

(../../../../_images/with_bot.png)

5. Now that we have verified the Bot Profile is actively inserting the challenge, you can **Close** the **Inspection tools** in the browser and **Refresh** the Juice Shop site. **Dismiss** the popup and click on one of the first items for sale such as the Apple or Banana Juice.

6. Back in the Advanced WAF tab navigate to **Security > Event Logs > Bot Defense > Bot Requests** and review the event logs. You will see all valid and/or challenged requests from **"Chrome Browser"**.

7. Click on some of the requests and then click the **All Details** tab on the right and review the **Verification Action and Challenge Status**. You will also see a unique DeviceID was assigned per the Balanced_Bot_Profile default settings. Also note the Bot Details and the full text visibility of the request below.

The screenshot shows the FortiWeb interface under the 'Bot Defense' tab. On the left, a list of 'Bot Requests' is displayed, with one item selected. On the right, a detailed view of the selected request is shown. The 'Request Details' section includes fields like Requested URL, Host, Time, Geolocation, Source IP Address, Destination IP Address, and Device ID. The 'Verification Action and Challenge Status' section shows configured verification actions and actual verification results. The 'Bot Details' section provides information about the bot's name, class, and categories. The 'Request' section displays the raw HTTP traffic. Several parts of the interface are highlighted with green boxes: the 'Verification Action and Challenge Status' section, the 'Bot Details' section, and the 'Request' section.

([..../../_images/goodbot.png](#))

Testing with a Bot¶

1. Open a Terminal on the Linux Client and run the following command:

```
curl -k https://juiceshop.f5agility.com
```

2. Refresh **Security > Event Logs > Bot Defense > Bot Requests** and review the event logs. Was the Request blocked?

The screenshot shows the FortiWeb interface under the 'Bot Defense' tab. On the left, a list of 'Bot Requests' is displayed, with one item selected. On the right, a detailed view of the selected request is shown. The 'Request Details' section includes fields like Requested URL, Host, Time, Geolocation, Source IP Address, Destination IP Address, and Device ID. The 'Verification Action and Challenge Status' section shows configured verification actions and actual verification results. The 'Bot Details' section provides information about the bot's name, class, and categories. The 'Request' section displays the raw HTTP traffic. The 'Request Details' section is highlighted with a green box.

([..../../_images/untrust.png](#))

- This request was not blocked but did produce an alarm. **Click** on the **Mitigation Action** in Request Details for more information around the enforcement.

Mitigation Action	Alarm (Untrusted Bot)
Virtual Server	
Bot Defense Profile	
Microservice	
<input checked="" type="checkbox"/> Bot Details	Configured Action Alarm Actual Action Alarm Actual Action Reason None Enforced By Profile Mitigation and Verification Settings

(../../../../_images/mitver.png)

Note

Curl is an untrusted bot, but not necessarily malicious. By default, the Balanced policy is set to only alarm on untrusted bot access. This can be tuned per your environment.

- Now we will test with a request that is formatted to appear as if it is coming from a malicious user-agent (Nikto). In the terminal run the following curl command:

```
curl https://juiceshop.f5agility.com/ -k -H "User-Agent: Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:Port Check)"
```

You should get a **Request Rejected** response in the Terminal window.

```
f5student@ip-10-1-1-7: ~
File Edit View Search Terminal Help
f5student@ip-10-1-1-7:~$ f5student@ip-10-1-1-7:~$ f5student@ip-10-1-1-7:~$ f5student@ip-10-1-1-7:~$ f5student@ip-10-1-1-7:~$ curl https://juiceshop.f5agility.com/ -k -H "User-Agent: Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:Port Check)"
<html><head><title>Request Rejected</title></head><body>The requested URL was rejected. Please consult with your administrator.<br><br>Your support ID is: <14154075482861105067><br><br><a href='javascript:history.back();'>[Go Back]</body></html>
f5student@ip-10-1-1-7:~$ f5student@ip-10-1-1-7:~$
```

(../../../../_images/reject.png)

- Refresh **Security > Event Logs > Bot Defense > Bot Requests** and review the event logs. You will see that the Bot was categorized as malicious and blocked. Also notice that there is no DeviceID because the bot was blocked immediately due to its categorization. No challenge necessary. DeviceID is provided via the JS Challenge.
- Examine **All Details** and review the **Verification Action and Challenge Status**. Notice there is none. Why?

The screenshot shows the FortiWAF interface under the 'Bot Defense' tab. On the left, a list of bot requests is displayed, with one entry selected and highlighted in blue. The selected request is shown in a detailed view on the right. The detailed view includes sections for Request Details, Verification Action and Challenge Status, and Bot Details. The 'Request Status' is 'Denied' and the 'Mitigation Action' is 'Block (Malicious Bot)'. The 'Bot Defense Profile' is set to 'Balanced_Bot_Profile'. The 'Verification Action and Challenge Status' section shows that the configured verification action is 'None' and the actual verification action reason is also 'None'. The 'Bot Details' section is partially visible at the bottom.

(../../../../_images/blocked.png)

i Note

The reason there are no challenges for this request is because these requests have user-agents associated with them that are well-known by our Bot Signatures so there is no reason to challenge them. They match the signature at the time of the request so an action is immediately taken based on the categorization of the bot.

So what if an attacker spoofs user-agents to look legitimate? Let's try to trick the WAF by using curl and spoofing a legitimate user-agent.

Spoofing a legitimate UA¶

1. Select one of the **Accepted Requests** in Bot Requests Log and scroll down to examine the request. We will “borrow” the user-agent from that request since we know it is a valid browser UA.

Security > Event Logs : Bot Defense - Bot Requests

Order by Date ▾ Newest ▾

Bot Requests

- [HTTPS] / favicon.ico (Malicious Bot) 10.1.10.100 10:02:16 2021-02-04
- [HTTPS] /favicon.ico (Chrome (Browser)) 10.1.10.100 10:02:04 2021-02-04
- [HTTPS] /assets/public/images/products/artw... (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04
- [HTTPS] /assets/public/images/products/per... (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04
- [HTTPS] /assets/public/images/products/gree... (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04
- [HTTPS] /socket.io/ (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04
- [HTTPS] /assets/public/images/products/item... (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04
- [HTTPS] /assets/public/images/products/ccg... (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04
- [HTTPS] /assets/public/images/products/egg... (Chrome (Browser)) 10.1.10.100 10:02:02 2021-02-04

Request

```

GET /favicon.ico HTTP/1.1
Host: juiceshop.f5agility.com
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: https://juiceshop.f5agility.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: language=en; welcomebanner_status=dismiss; TS0000000000076=082a7ad0c3ab280097d4d4170969991f4da1ed9160d854beb7a79c9f82eb7
6c67fd49a724c08cdec9c99c879fecfa9f98f690b86c09d000c:35319fb6978fd3d09a0adaddb645d6238140da5344138480f19742f5f16f7db6850df16c4
ba0b0eec284c814991d9568c0fc62b4eacecf998bb0aed7514ae172f87bd4fb81196a0b3fe38f7eb9288315f499af13864a7e992480d7691698cb1d1776b14
115f7a73cb987560ffdbe17b052f9d4a8a8d52c318b760162b69fe2e2a29917b676fc565d0344888f10f64e19af2510e51593b3211d1245c7fca03be3ee1
2624390d1ab0b1376b83aea233864f65bb110384c8038bb62865ffd4c5e23e7f2f6215e509b3bd0484167b; TSPD_101_ID=082a7ad0c3ab280097d4d4
170969991f4da1ed9160d854beb7a79c9f82e76cfcfd49a724c08cdec9c99c879fecfa9f08f6908b86c063800fc1032716a8f91f9419ea4c4885c159ca337
75886e846ea06caa30f48f99fc3b1d4954e65203c1d6ee6cd8af0f266149168264b167fb08519; TSPD_101=082a7ad0c3ab280097f83287937115d00f2aafa7
0c627dd2799acb942116131a3e600b128a7f7767fb4935f087ca869450875d4de160518001d351e0e7f83b3ee84cd5917da691ce882f832c475e65; io=
RWhu3T_KNWK9AvIkAAK; Ts7b780dc7029-082a7ad0c3ab28001548c2068f358ab98a0e6535f2edfb812577a8d7f257c386f03d04256345ea126f3d6e5ab
531pe5; Ts7b780dc7077-082a7ad0c3ab2800b7ce8bf1f6bia397464abc3882c13bb77a88519cef49b3f9844337ffccdf1e1e081701d29becd08ea12ce
56172090427787d44a53eb697dd80897b63ed07ea4b14884c69fe1e42ea79212c26111c99; TSc7ac518027=082a7ad0c3ab200018f6b2a44ad6424be28ece
6772755b1bea0e093978aa7446b3df7a5c1cbf0406142616ad113090c11b64b05451e2d17576405c9499995fb6c6eb0de1ca5dd7304fdaede5a374f3826e
470275a8b8bc6a25b10541e542b
```

(../../../../_images/legit.png)

2. In the terminal run the following command:

```
curl https://juiceshop.f5agility.com/ -k -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36" | more
```

Here we see a response but it isn't the default HTML of the Juiceshop page we saw earlier or a **Request Rejected** page as seen in the previous example. Continue to hit the space bar to see the remainder of the response/challenge.

```
f5student@ip-10-1-1-7:~$ curl https://juiceshop.f5agility.com/ -k -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36"
<!DOCTYPE html>
<html><head>
<meta http-equiv="Pragma" content="no-cache"/>
<meta http-equiv="Expires" content="-1"/>
<meta http-equiv="CacheControl" content="no-cache"/>
```

(../../../../_images/js.png)

3. Refresh **Security > Event Logs > Bot Defense > Bot Requests**. You should now see a **Challenged Event**.

Security > Event Logs > Bot Defense : Bot Requests

Order by Date ▾ Newest ▾

[HTTPS] / Presenting as Chrome (Suspicious Browser) 10.1.10.100 10:35:16 2021-02-04

Load Related Requests

[HTTPS] / nikto (Malicious Bot) 10.1.10.100 10:02:16 2021-02-04

[HTTPS] / favicon.ico Chrome (Browser) 10.1.10.100 10:02:04 2021-02-04

[HTTPS] /assets/public/images/products/artw... Chrome (Browser) 10.1.10.100 10:02:02 2021-02-04

[HTTPS] /assets/public/images/products/per... Chrome (Browser) 10.1.10.100 10:02:02 2021-02-04

[HTTPS] /assets/public/images/products/gree... Chrome (Browser) 10.1.10.100 10:02:02 2021-02-04

[HTTPS] /socket.io/ Chrome (Browser) 10.1.10.100 10:02:02 2021-02-04

[HTTPS] /assets/public/images/products/ern... Chrome (Browser) 10.1.10.100 10:02:01 2021-02-04

[HTTPS] /assets/public/images/products/cgc...

Delete

Request Details

Requested URL	[HTTPS] /	Request Status	Challenged
Host	juiceshop.f5agility.com	Mitigation Action	CAPTCHA (Suspicious Browser)
Time	2021-02-04 10:35:16	Virtual Server	owasp-juiceshop_443_vs
Geolocation	N/A	Bot Defense Profile	Balanced_Bot_Profile
Source IP Address	10.1.10.100:45268	Microservice	N/A

Bot Details

Bot Name	Presenting as Chrome	Detected Anomalies	Suspicious HTTP Headers Presence or Order
Bot Class	Suspicious Browser	Detected Bot Signature	N/A
Bot Categories	Suspicious Browser Types		

Request

```
GET / HTTP/1.1
Host: juiceshop.f5agility.com
Accept: /*
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36
```

(../../../../_images/challenge.png)

4. Look under **Bot Details > Detected Anomalies** and note the anomaly of Suspicious HTTP Headers Presence or Order.

Bot Details

Bot Name	Presenting as Chrome	Detected Anomalies	Suspicious HTTP Headers Presence or Order
Bot Class	Suspicious Browser	Detected Bot Signature	N/A
Bot Categories	Suspicious Browser Types		

Request

```
GET / HTTP/1.1
Host: juiceshop.f5agility.com
Accept: /*
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.141 Safari/537.36
```

(../../../../_images/anomaly.png)

❶ Note

The Bot Profile identified the requesting party as a suspicious browser and issued a Captcha Response due to Suspicious HTTP Headers Presence or Order. That was the Javascript we saw returned in the terminal window.

At this point it should be getting clearer to you as to how a bot profile operates from a mitigation settings perspective and how to validate your configurations. We “could” start blocking or redirecting Untrusted Bots to another pool of servers for forensics. We “could” also send Suspicious browsers or Malicious Bots to a honeypot page.

Mitigation Settings

Trusted Bot	Alarm
Untrusted Bot	Redirect to Pool
⚠ Note: This action requires configuring an existing pool in "Redirect Pool" field (General Settings)	
Suspicious Browser	Honeypot Page
Malicious Bot	Block
Unknown	Rate Limit for 30 transactions per second

(../../../../_images/mitig.png)

This concludes Lab 1