

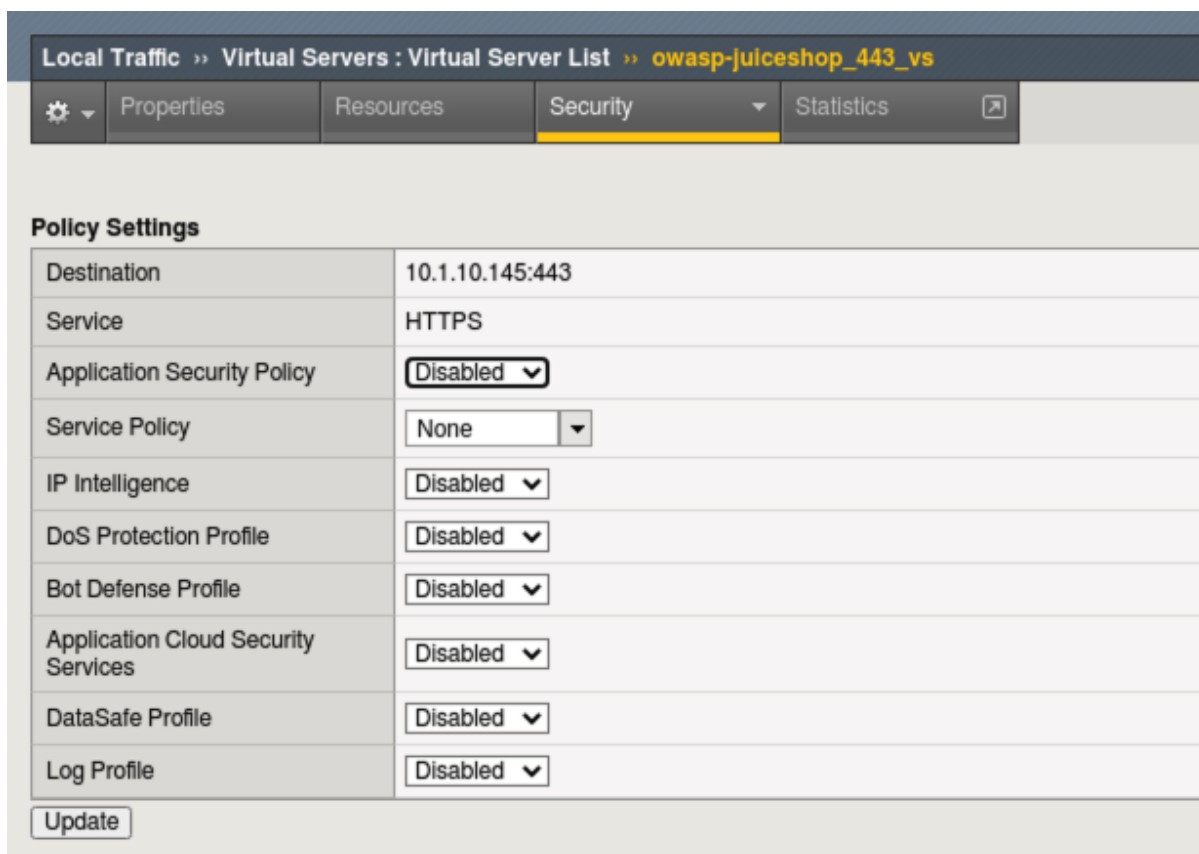
Lab 3: Behavioral DOS Protection¶

In this lab you will use a baseline traffic generation script and an Apache Bench based attack script against a Virtual Server in both transparent and blocking mode to trigger Behavioral DoS Protection. **Unlimited Behavioral DoS protection profiles are a feature of Advanced WAF. Legacy ASM customers are limited.**

Test Default Site Behavior¶

1. Navigate to **Local Traffic > Virtual Servers > owasp-juiceshop_443_vs > Security > Policies**.
2. **Disable** all Policies and Logging Profiles so that we can effectively demonstrate just the DoS mitigation.
3. Click **Update**.

Your Virtual Server should look like this:



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

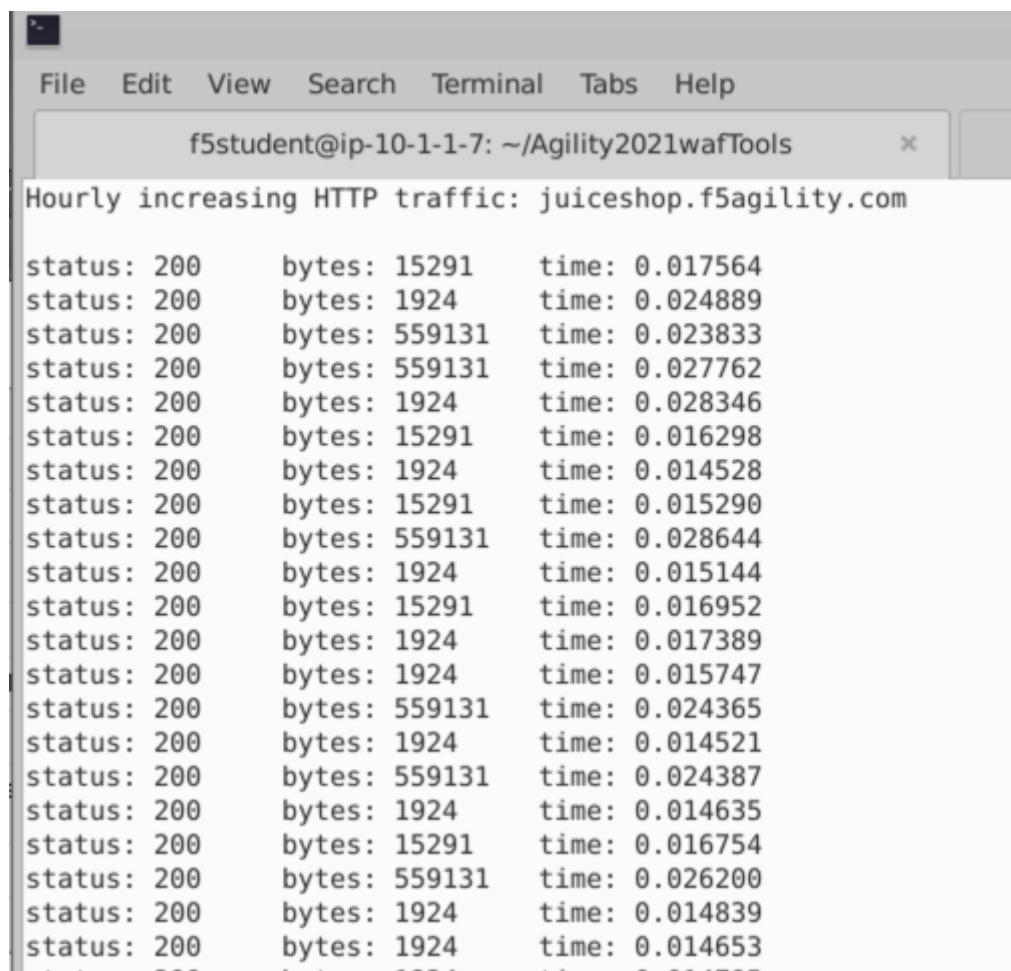
(../../../../_images/dos_vs.png)

4. Open a new **Terminal** window and run the **baseline_menu.sh** script in the **Agility2021wafTools** directory.

```
cd ~/Agility2021wafTools/  
./baseline_menu.sh
```



5. Choose **Option 2** and hit the **Return Key or Enter key**.
6. You will see the script start sending traffic to the site and you should see all responses with an http code of **status: 200**.



The screenshot shows a terminal window with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help) and a title bar (f5student@ip-10-1-1-7: ~/Agility2021wafTools). The terminal output displays the message "Hourly increasing HTTP traffic: juiceshop.f5agility.com" followed by a list of HTTP response statistics. Each line shows "status: 200", "bytes" (alternating between 15291 and 1924), and "time" (various decimal values).

```
Hourly increasing HTTP traffic: juiceshop.f5agility.com
status: 200      bytes: 15291      time: 0.017564
status: 200      bytes: 1924       time: 0.024889
status: 200      bytes: 559131     time: 0.023833
status: 200      bytes: 559131     time: 0.027762
status: 200      bytes: 1924       time: 0.028346
status: 200      bytes: 15291     time: 0.016298
status: 200      bytes: 1924       time: 0.014528
status: 200      bytes: 15291     time: 0.015290
status: 200      bytes: 559131     time: 0.028644
status: 200      bytes: 1924       time: 0.015144
status: 200      bytes: 15291     time: 0.016952
status: 200      bytes: 1924       time: 0.017389
status: 200      bytes: 1924       time: 0.015747
status: 200      bytes: 559131     time: 0.024365
status: 200      bytes: 1924       time: 0.014521
status: 200      bytes: 559131     time: 0.024387
status: 200      bytes: 1924       time: 0.014635
status: 200      bytes: 15291     time: 0.016754
status: 200      bytes: 559131     time: 0.026200
status: 200      bytes: 1924       time: 0.014839
status: 200      bytes: 1924       time: 0.014653
status: 200      bytes: 1924       time: 0.014705
```

(../_images/status.png)

7. While the script is still running, open a new browser tab and click on the **OWASP Juice Shop** bookmark.
8. Browse around the site and refresh the page a few times. You should not feel any lag in response time and everything should be loading quickly in the browser even with the baseline script running in the background.

Start the Attack¶

1. Open a new **Terminal** tab and ensure you are in the **~/Agility2021wafTools/** directory and run the **AB_SSL_DOS.sh** script.

```
cd ~/Agility2021wafTools/
./AB_SSL_DOS.sh
```



2. Choose **Option 1** and hit the **Return Key or Enter key**.
3. Back in Chrome browser, attempt to refresh the site. **There are no smoke and mirrors here.** The Apache Bench script almost instantly knocked this single container site offline. If it isn't offline, then it's horribly slow and un-usable. **There was no special configuration on the server side.**
4. Stop the Apache Bench script by hitting (Ctrl + C) and then type the # **4** and hit the **Return Key or Enter key**.
5. In Juice Shop refresh the page a few times. The site should recover quickly as the connections die down.
6. **Leave the baseline_menu script running.**

Questions: What do you do when you are shopping online and a site is behaving like this? Do you think attackers and/or competing entities ever use this to their advantage?

Advanced WAF to the Rescue¶

For demonstration purposes in this lab we will simply configure Source IP based DoS Protection, although there are several selectors available including DoS mitigation based on Device ID, Geolocation, URL and Site Wide.

1. In the Advanced WAF tab of Chrome, navigate to **Security > DoS Protection > Protection Profiles** and click **Create**.
2. Name the profile **juiceshop_dos** and click **Behavioral & Stress-based Detection** to the left.
3. Configure the **Behavioral & Stress-based Detection** settings like below:

- Operation Mode: **Transparent** (It is always best-practice to add new features to a policy in transparent mode until you fully understand the impacts.)
- Thresholds Mode: **Manual**
- Stress-based Detection and Mitigation:
 - By Source IP: Click **Edit** and check the box at the bottom for **Client Side Integrity Defense** which tells the WAF to send an Active JS challenge when under attack to verify the browser vs an attacking bot.

- Set the Relative Threshold to 500% and **15** transactions per second.
- Set the Absolute Threshold TPS to **20**.

Profile Information

General Settings

Application Security

General Settings

TPS-based Detection Off

Behavioral & Stress-based Detection Transparent

Record Traffic Off

Application Security » Behavioral & Stress-based (D)DoS Detection Edit All

This section configures the detection of DoS attacks based on server stress. The system automatically detects an increase in server stress and mitigates DoS attacks causing it.

Operation Mode	Specifies how the system reacts when it detects an attack.	Transparent	Close
Thresholds Mode	Specifies what type of thresholds to use.	Manual	Close
Stress-based Detection and Mitigation	By Source IP	<p>Consider an IP as an attacking entity if either of the following conditions occur:</p> <p>Relative Threshold: TPS increased by: <input type="text" value="500"/> % and reached at least <input type="text" value="15"/> transactions per second OR</p> <p>Absolute Threshold: TPS reached: <input type="text" value="20"/> transactions per second</p> <p>Set default criteria</p> <p>Select mitigation methods to use on the attacking IP's:</p> <p><input checked="" type="checkbox"/> Client Side Integrity Defense</p> <p><input type="checkbox"/> CAPTCHA Challenge</p> <p><input type="checkbox"/> Request Blocking</p>	Close
	By Device ID	No mitigation	Edit
	By Geolocation	No mitigation	Edit
	By URL	No mitigation	Edit
	Site Wide	No mitigation	Edit

(../../../../_images/behav.png)

- Behavioral Detection and Mitigation: > click **Edit**

- Check the box for **Use TLS patterns as part of host identification**
- Check the box for **TLS fingerprinting signatures**
- Under **Mitigation** read the description of **Standard Protection**.

- Prevention Duration: > click **Edit**

- Escalation Period: **90** seconds
- De-escalation Period: **360** seconds

Behavioral Detection and Mitigation	<div> By Bad Actors Behavior / Signatures Close </div> <div> <input checked="" type="checkbox"/> Bad actors behavior detection <small>Enables bad actors detection by behavioral analysis.</small> </div> <div style="border: 2px solid green; padding: 5px; margin: 5px 0;"> <input checked="" type="checkbox"/> Use TLS patterns as part of host identification <small>Expands identification key using individual TLS patterns.</small> </div> <div> <input checked="" type="checkbox"/> Request signatures detection <small>Enables signatures detection</small> </div> <div> <input type="checkbox"/> Accelerated HTTP signatures <small>Applicable only for HTTP VS(s).</small> </div> <div> <small>Enables signatures detection before the connection establishment. Automatically enables syn-cookie mechanism during attack.</small> </div> <div style="border: 2px solid green; padding: 5px; margin: 5px 0;"> <input checked="" type="checkbox"/> TLS fingerprinting signatures <small>Applicable only for HTTPS VS(s).</small> </div> <div> <small>TLS fingerprinting signatures. Utilizes syn-cookie mechanism if VS configuration allows SYN Challenge Handling.</small> </div> <div> <input type="checkbox"/> Use approved signatures only </div> <hr/> <div> Mitigation </div> <div> <div>Standard protection * ▼</div> </div> <div> <small>If "Bad actors detection" enabled, slows down requests from anomalous IP addresses based on its anomaly detection confidence and the server's health. Rate limits requests from anomalous IP addresses and, if necessary, rate limits all requests based on the server's health. Limits the number of concurrent connections from anomalous IP addresses and, if necessary, limits the number of all concurrent connections based on the server's health. If "Request signatures detection" enabled, blocks requests that match the attack signatures.</small> </div> <div> <small>Note: Behavioral DoS will not mitigate attacks because the Behavioral and Stress-Based operation mode is set to Transparent</small> </div>
Prevention Duration	<div> Specifies the time spent in each mitigation step until it is stopped, and the next one is started. </div> <div style="border: 2px solid green; padding: 5px; margin: 5px 0;"> Escalation Period: <input type="text" value="90"/> seconds De-escalation Period: <input type="text" value="360"/> seconds </div> <div> Set default duration </div> <div style="text-align: right;">Close</div>

(../../_images/bdos.png)

4. Click **Finished**

i Important

What you essentially just configured was a DoS "Client Source IP" traffic categorizer and step-up mitigations. In this case, mitigations will fire in the order that they are shown in the GUI and progress every 90 seconds. First, a client side integrity check is performed (JS injection) and if necessary the WAF will escalate to issuing a CAPTCHA to offending clients. Finally, Rate-Limiting kicks in as a last resort. You could use any combination of these three mitigation tactics with any of the DoS categorizer types in the profile.

Apply the Dos Profile and Test

1. Navigate to **Local Traffic > Virtual Servers > owasp-juiceshop_443_vs > Security > Policies.**

2. **Enable** the Dos Protection Profile and choose our new **juiceshop_dos** profile and also **Enable** the **local-dos** Logging Profile.

3. Click **Update**.

Your virtual server should look like this:

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

Properties Resources **Security** Statistics

Policy Settings

Destination	10.1.10.145:443
Service	HTTPS
Application Security Policy	Disabled
Service Policy	None
IP Intelligence	Disabled
DoS Protection Profile	Enabled... Profile: juiceshop_dos
Bot Defense Profile	Disabled
Application Cloud Security Services	Disabled
DataSafe Profile	Disabled
Log Profile	Enabled... Selected: /Common, local-dos Available: /Common, Log all requests, Log illegal requests, global-network, local-bot-defense

Update

(../_images/vs1.png)

4. Open a new tab to Juice Shop and browse the site. There should not be any perceivable effect of having applied the DoS profile.

5. Browse the photo wall from the hamburger menu (top left) and leave some arbitrary Customer Feedback. Spend about a minute browsing the site.

6. Open a new **Terminal** tab and start the attack script again and choose option **1**:

```
cd ~/Agility2021wafTools/  
./AB_SSL_DOS.sh
```

7. Navigate to **Security > Event Logs > DoS > Application Events** and review the entry. The system immediately picked up the attack due to the behavior.

8. Click on the Attack ID #.

Security » Event Logs : DoS : Application Events

Application

Protocol

Network

DoS

Bot Defense

Logging Profiles

Last Hour

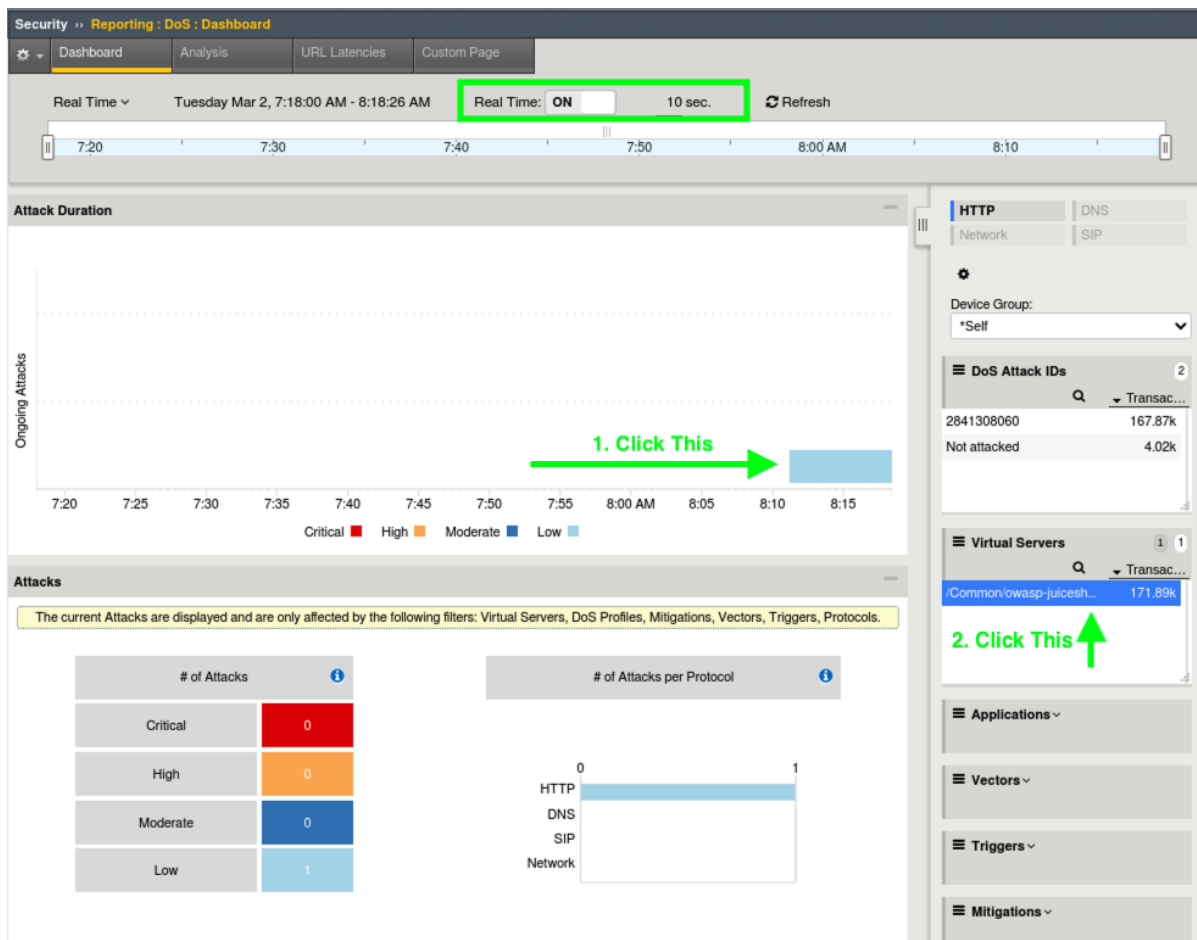
Search

Custom Search...

Time	Virtual Server	Profile Name	Event	Detection Mode	Mitigation	TPS	Detection Threshold	Mitigate To Threshold	Threshold Condition	Attack ID
2021-03-02 08:11:21	/Common/owasp-juiceshop_443_vs	/Common/juiceshop_dos	Attack started	Behavioral detection	Transparent	30 tps				2841308060

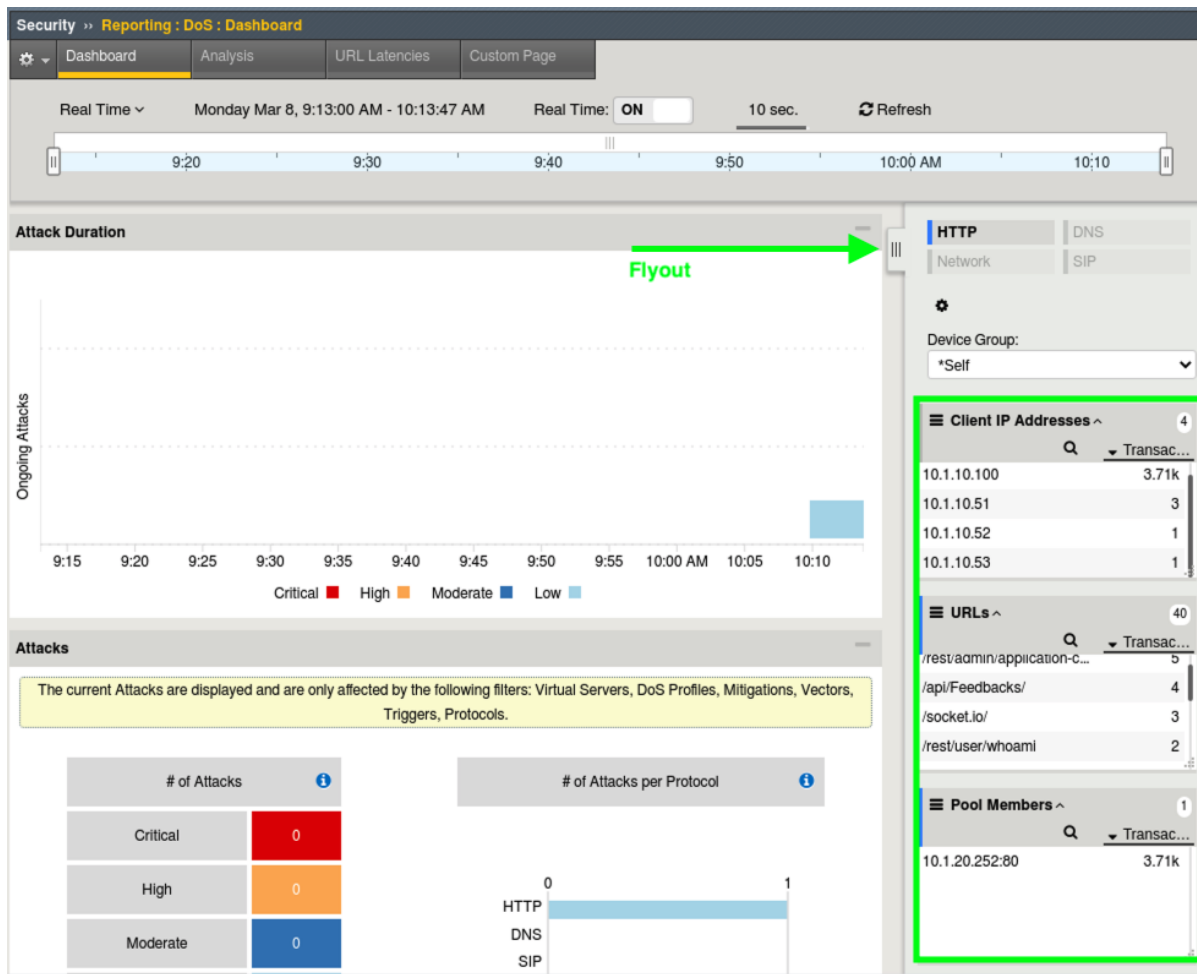
(../../../../_images/dos.png)

- Once the Dashboard loads, turn on **Real Time** by checking the box at the top under **DoS Attack IDs**. It will take a few moments for the data to populate. "Real Time" is relative here.



(../../../../_images/bdoslog.png)

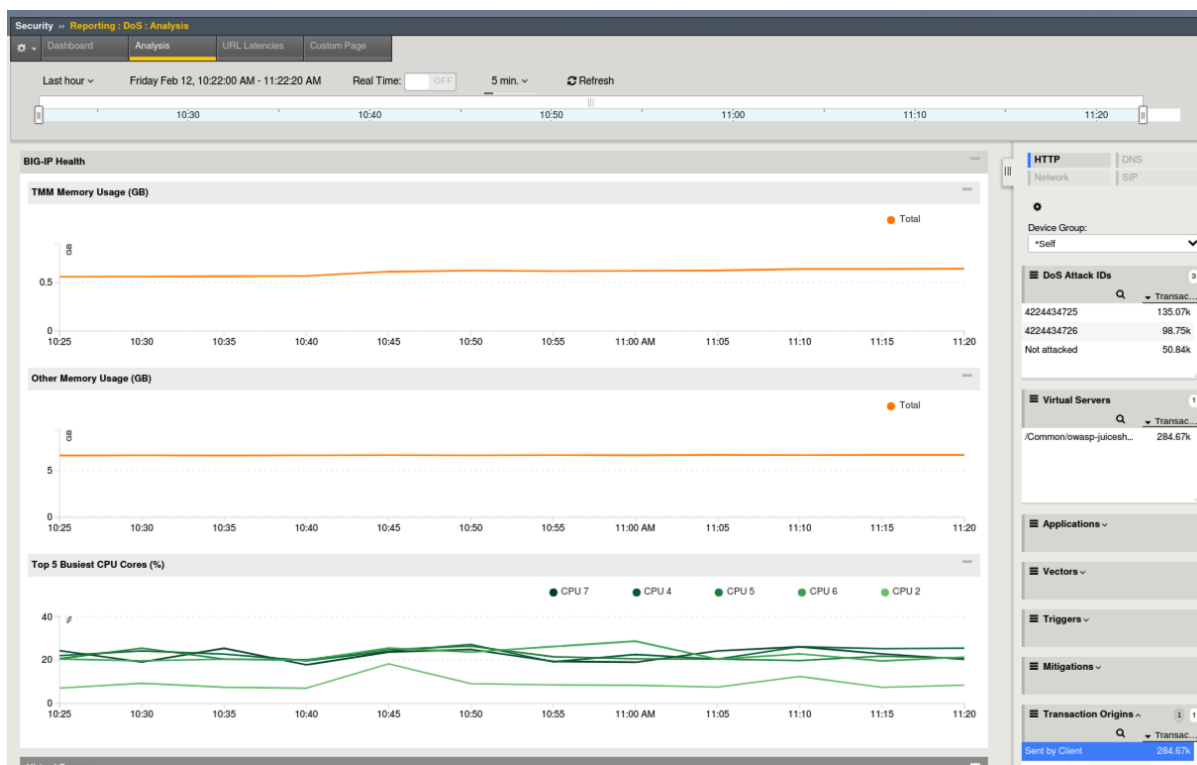
- Eventually...it may take several minutes, click on the attack graphic and then select the virtual server to the right. You may have to wait a few moments for the Virtual Server to appear.
- There is a lot of information on DoS Visibility Dashboard including the type of attack, the severity, duration and much more. You can use the **Real Time** filters on the right to further dissect the traffic and drill down for analysis. It may take some time for various data fields to load.
- From the right hand filters menu expand **Client IP Addresses**, **Pool Members** and **URLS** and review the attack data. You can drag the boxes to group them closer together as shown here and there is also a flyout.



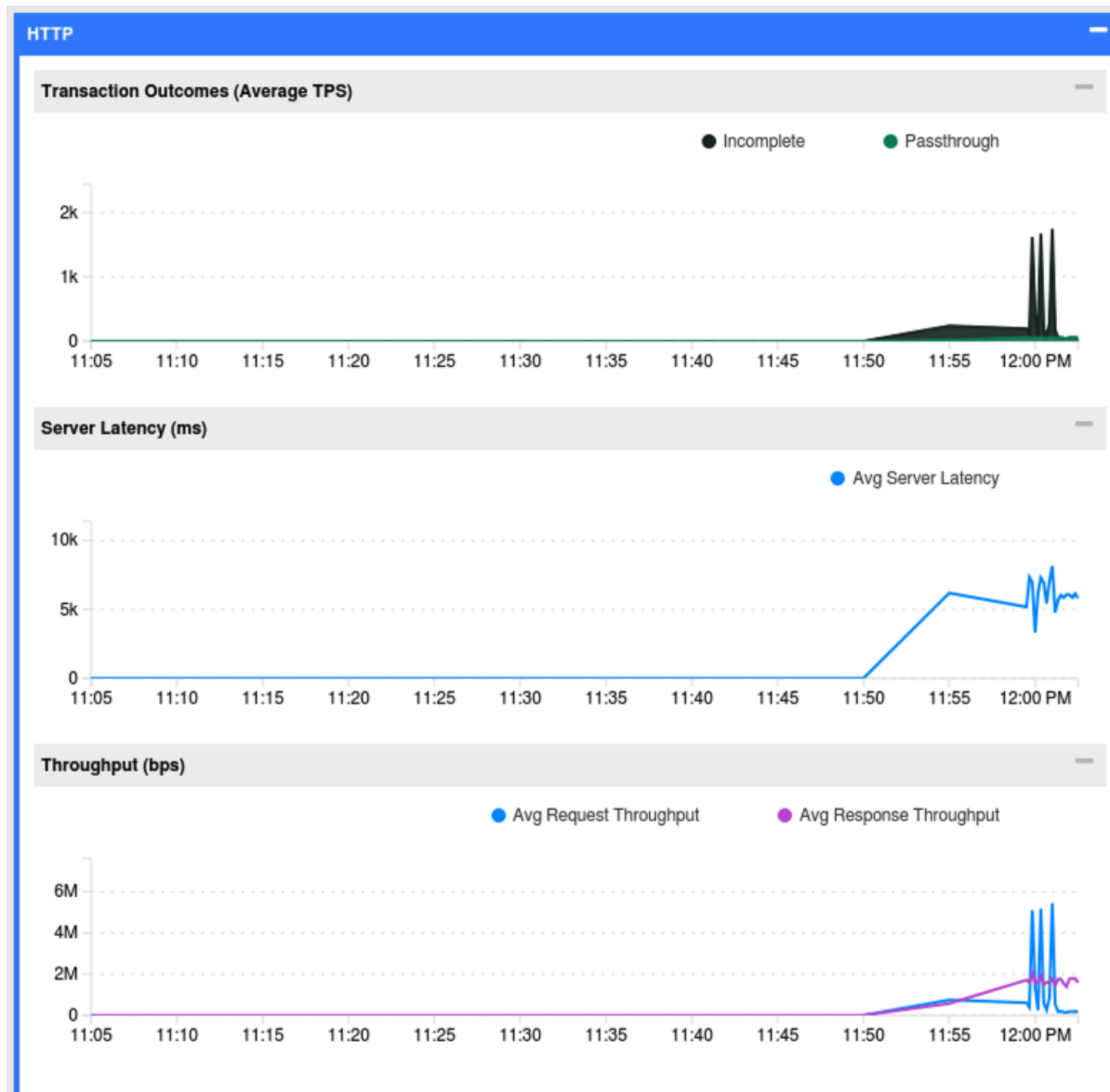
(../_images/client.png)

13. In the top middle of the GUI click the **Analysis** tab and review the system health overview of the BIG-IP device itself. Scroll down and check out the CPU, Connection and Throughput stats.

14. At the bottom you will get to the HTTP stats which should be of most interest.



(../_images/sys.png)



(../../_images/http_stats.png)

Stop the Baseline and Attack Scripts¶

1. In each of your terminal windows or tabs type **Ctrl+C** to terminate **all** the scripts including the baseline. The **AB_SSL_DOS.sh** script will require you to enter **4** to completely stop the attacks.

Enable Blocking in the DoS Profile¶

1. In the Advanced WAF tab, navigate to **Security > DoS Protection > Protection Profiles > juiceshop_dos** and click on **Behavioral & Stress-based Detection**.
2. Under **Operation Mode** click **Edit** and from the dropdown choose **Blocking** and click **Update**.
3. Open a new tab to Juice Shop and ensure the site is operating normally.
4. Open a new Terminal Tab and run the AB_SSL_DOS script again. Choose **Option 1**.

```
cd ~/Agility2021wafTools/  
./AB_SSL_DOS.sh
```



5. Attempt to refresh Juice Shop..initially it is down, but within about a minute and a few refreshes later, the WAF has figured out the attack and taken mitigation action.
6. Navigate to **Security > Event Logs > DoS > Application Events** and review the new entries. We can see that the attack was picked up by behavioral mitigation first.
7. Next we can see that the mitigation was changed to **DOS L7 Attack** with **Source IP-Based Client Side Integrity Defense**. This means that the WAF is actively challenging these IP addresses with JS.
8. Expand the + and you will be able to see more details about the mitigation for each of the client IP's.

Security > Event Logs > DoS > Application Events												
Application		Protocol	Network	DoS	Bot Defense	Logging Profiles						
Time		Virtual Server	Profile Name	Event	Detection Mode	Mitigation	TPS	Detection Threshold	Mitigate To Threshold	Threshold Condition	Attack ID	Entity Type
2021-03-08 10:19:32	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Change mitigation	DOS L7 attack	Source IP-Based Client Side Integrity Defense	140 tps				726104061	
2021-03-08 10:19:40	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Suspicious entity	DOS L7 attack	Source IP-Based Client Side Integrity Defense	39 tps	20 tps	17 tps	Absolute Manual Threshold	726104061	Source IP 10.1.10.52 0
2021-03-08 10:19:40	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Suspicious entity	DOS L7 attack	Source IP-Based Client Side Integrity Defense	44 tps	20 tps	17 tps	Absolute Manual Threshold	726104061	Source IP 10.1.10.51 0
2021-03-08 10:19:40	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Suspicious entity	DOS L7 attack	Source IP-Based Client Side Integrity Defense	44 tps	20 tps	17 tps	Absolute Manual Threshold	726104061	Source IP 10.1.10.53 0
2021-03-08 10:19:24	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Change mitigation	Behavioral detection	Behavioral mitigation	110 tps				726104061	
2021-03-08 10:17:41	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Change mitigation	DOS L7 attack	Source IP-Based Client Side Integrity Defense	121 tps				726104061	
2021-03-08 10:17:32	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Attack started	Behavioral detection	Behavioral mitigation	0 tps				726104061	
2021-03-08 10:17:30	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Attack ended	Behavioral detection	Transparent	145 tps				726104059	
2021-03-08 10:10:00	/Common/owasp-juiceshop_443_vs		/Common/juiceshop_dos	Attack started	Behavioral detection	Transparent	32 tps				726104059	

(../..../_images/events11.png)

Note

The Linux client uses 10.1.10.100 as it's primary source IP. This is the IP you are coming from when using the browser. The Apache Bench script is configured to use alternate source IP's (10.1.10.51, 52, and 53)

Verifying Behaviors

1. Notice Juice Shop continues to load fine in the browser now that mitigations are in place for the attacking IP's.
2. Open a new terminal tab and run the following command: `curl -k https://juiceshop.f5agility.com`

3. Notice the default HTML being returned for the site. You are coming from the **.100** IP address.

```
f5student@ip-10-1-1-7:~/Agility2021wafTools$ curl -k https://juiceshop.f5agility.com
<!--
  ~ Copyright (c) 2014-2021 Bjoern Kimminich.
  ~ SPDX-License-Identifier: MIT
-->

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon.js.ico">
  <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
/>
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "#546e7a", "text": "#ffffff" },
          "button": { "background": "#558b2f", "text": "#ffffff" }
        }
      })
    })
  </script>
</body>
</html>
```

(../../_images/curl1.png)

4. Stop the Apache Bench attack in the terminal window by typing **CTRL +C** and then **4** and hit **Return**.

5. Now run this command in terminal to send the request from an IP that is being mitigated by DoS profile. `curl -k --interface 10.1.10.51 https://juiceshop.f5aquility.com`

6. Notice the javascript challenge. This ip will continue to be challenged for the duration of the de-escalation period of 360 seconds that we set earlier or as long as the server is under stress from this IP.

```
f5student@ip-10-1-1-7:~/Agility2021wafTools$ curl -k --interface 10.1.10.51 https://juiceshop.f5agility.com

<!DOCTYPE html>
<html><head>
<meta http-equiv="Pragma" content="no-cache"/>
<meta http-equiv="Expires" content="-1"/>
<meta http-equiv="CacheControl" content="no-cache"/>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<link rel="shortcut icon" href="data:;base64,iVBORw0KGgo="/>

<script type="text/javascript">
(function(){
window["bobcmn"] = "101110101010102000000032000000052000000062000000012b7b0dc72000000962000000020000000130000000030000000
0300000006/TPSD/3000000008TPD 10130000000CSPD 101 DID300000005https3000000b0082a7ad0c3ab20002fb2ddb1c38d7c78e944ebd9d6b39a
420d7e2df5f6fad122552ee911aaf97b140821f911fa0a28005e9508d9cec95bc2e3bd4abb1e78dcd1f14fc19338776f97dbd116277514ac9738b3db723
db74606300000002TS200000000200000000";

window["failureConfig"] = "524f6f70732e2e2e2e736fd657468696e672077656e742077726f6e672e2e2e2e20796f757220737570706f72742069
642069733a2025444f534c372e6368616c6c656e67652e737570706f72745f6964252e143136343839333831373232393133343339333036062f5453504
42f171800";window.sWvc=!window.sWvc;try{(function(){(function OL(){var z=!1;function s(z){for(var s=0;--;)s+=I(document.d
ocumentElement,null):return s}function I(z,s){var l="vi";s=s||new J: return zL(z,function(z){z.setAttribute("data"+l,s.JsI
```

(../../_images/curl2.png)

This concludes Lab 3