

Web Application Hacking Lesson XSS Filter Evasion and Fuzzing

Lesson Objectives

-
- Review lecture on the Cross Site Scripting(below)
- Complete lab on Cross Site Scripting(below)

Cross Site Scripting Lecture:

XSS Filter Evasion

Takeaway: Sometimes a site will employ filters to try to prevent the XSS strings you have learned so far. This lecture will cover some ways to try to evade these security measures..

Sometimes, website owner use XSS filters(WAF) to protect against XSS vulnerability.

For eg: if you put the <scirpt>alert("hi")</script> , the Filter will escape the "(quote) character , so the script will become

```
<script>alert(>xss detected<)</script>
```

Now this script won't work. Likewise Filters use different type of filtering method to give protection against the XSS. In this case, we can use some tricks to bypass the filter. Here i am going to cover that only.

1.Bypassing magic_quotes_gpc

The magic_quotes_gpc=ON is a PHP setting(configured in PHP.ini File) , it escapes the every ' (single-quote), " (double quote) and \ with a backslash automatically.

For Eg:

<scirpt>alert("hi");</script> will be filtered as <script>alert(\hi\)</script>.so the script won't work now.

This is well known filtering method, but we can easily bypass this filter by using ASCII characters instead.

For Eg: alert("hi"); can be converted to

```
String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 104, 105, 34, 41, 59)
```

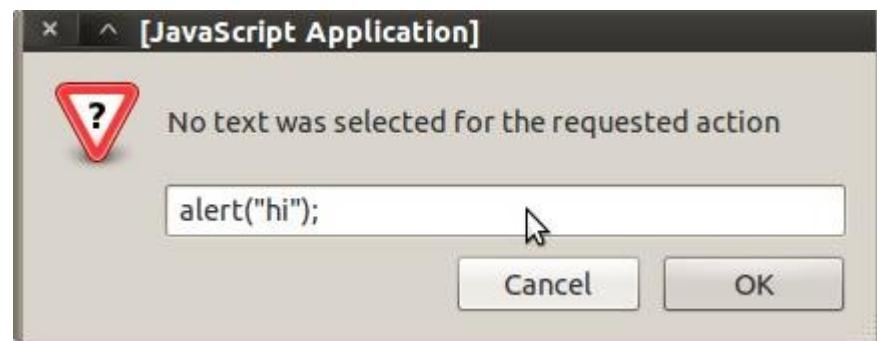
so the script will become <script>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 104, 105, 34, 41, 59)</script>. In this case there is no "(quotes) or '(single quotes) or / so the filter can't filter this thing. Yes, it will successfully run the script.

String.fromCharCode() is a javascript function that converts ASCII value to Characters.

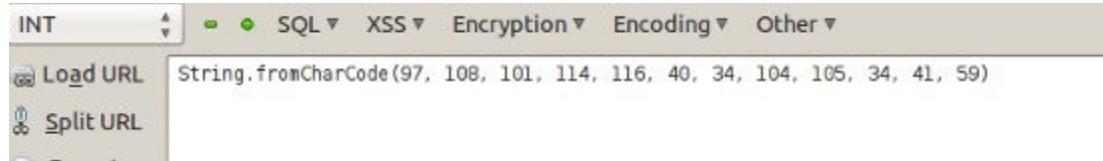
How to convert to ASCII values?

There are some online sites that converts to ASCII character. But i suggest you to use [Hackbar Mozilla addon](#).

After installing hackbar add on ,press F9. It will open the small box above the url bar. click the XSS->String.fromCharCode()



Now it will popup small window. enter the code for instance alert("Hi"). click ok button. Now we got the output.



copy the code into the <script></script> inside and insert in the vulnerable sites

For eg:

```
hxpx://vulnerable-site/search?q=<script>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 104,  
105, 34, 41, 59)</script>
```

2.HEX Encoding

we can encode our whole script into HEX code so that it can't be filtered.

For example: <script>alert("Hi");</script> can be convert to HEX as:

```
%3c%73%63%72%69%70%74%3e%61%6c%65%72%74%28%22%48%69%22%29%3b%3c%2f  
%73%63%72%69%70%74%3e
```

Now put the code in the vulnerable site request.

For ex:

```
hxpx://vulnerable-site/search?q=%3c%73%63%72%69%70%74%3e%61%6c  
%65%72%74%28%22%48%69%22%29%3b%3c%2f%73%63%72%69%70%74%3e
```

Converting to HEX:

This site will convert to hex code: <http://centricle.com/tools/ascii-hex/>

3.Bypassing using Obfuscation

Some website admin put the `script,alert` in restricted [word list](#). so whenever you input this keywords, the filter will remove it and will give error message like "you are not allowed to search this". This can bypassed by changing the case of the keywords(namely Obfuscation).

For eg:

```
<ScRipt>ALeRt("hi");</sCRipT>
```

This bypass technique rarely works but giving trial is worth.

4. Closing Tag

Sometimes putting ">" at the beginning of the code will work.

><script>alert("Hi");</script>

This will end the previous opened tag and open our script tag.

Example:

```
hxxp://vulnerable-site/search?q=><script>alert("Hi");</script>
```

Conclusion:

From above article, it is clear that XSS filters alone not going to protect a site from the XSS attacks. If you really want to make your site more [secure](#), then ask PenTesters to test your application or test yourself.

Also there are lot of different filter bypassing technique, i just covered some useful techniques for you.

Lab for Cross Site Scripting Introduction

Lab setup

- **Instructions:**
 1. Our target site is here <http://webscantest.com>

The screenshot shows a web page titled "Web Scanner Test Site". At the top right is a "Login" button. Below the title, a message says "Tell us a little about yourself". There are three input fields: "First Name" (empty), "Nick Name" (empty), and "Last Name" (empty). A "submit" button is below the last field. At the bottom of the page, a note states "The form based credentials are testuser/testpass, and the HTTP Basic credentials are btestuser/btestpass." and a "Privacy Policy" link is present.

Web Scanner Test Site

[Login](#)

Tell us a little about yourself

First Name:

Nick Name:

Last Name:

The form based credentials are testuser/testpass, and the HTTP Basic credentials are btestuser/btestpass.

[Privacy Policy](#)

Lab – Automated XSS Scanning, and Fuzzing

Fuzz testing or **fuzzing** is a [software testing](#) technique, often automated or semi-automated, that involves providing invalid, unexpected, or [random data](#) to the inputs of a web application. The input is then monitored for exceptions such as an error message etc.

Launch ZAP 2.0

1. in the directory that holds the zap file, type: java -jar zap.jar
2. Open firefox and configure it to use the ZAP 2.0 Proxy http://www.youtube.com/watch?v=Xp_PBH7wjiw&list=PLEBitBW-Hlsv8cEIUntAO8st2UGHmrjUB&index=2
3. Open the link from Firefox <http://webscantest.com/crosstraining/aboutyou.php>
4. In the first name text box enter in the word “test”

The screenshot shows a web page titled "Web Scanner Test Site". At the top right is a "Login" button. Below the title, a placeholder text "Tell us a little about yourself" is visible. There are three text input fields: "First Name:" containing "test", "Nick Name:" (empty), and "Last Name:" (empty). A red rectangular box highlights the "First Name:" field and its associated dropdown menu. Below the form is a note: "The form based credentials are testuser/testpass, and the HTTP Basic credentials are btestuser/btestpass." At the bottom is a "Privacy Policy" link.

Web Scanner Test Site

First Name: test

Nick Name:

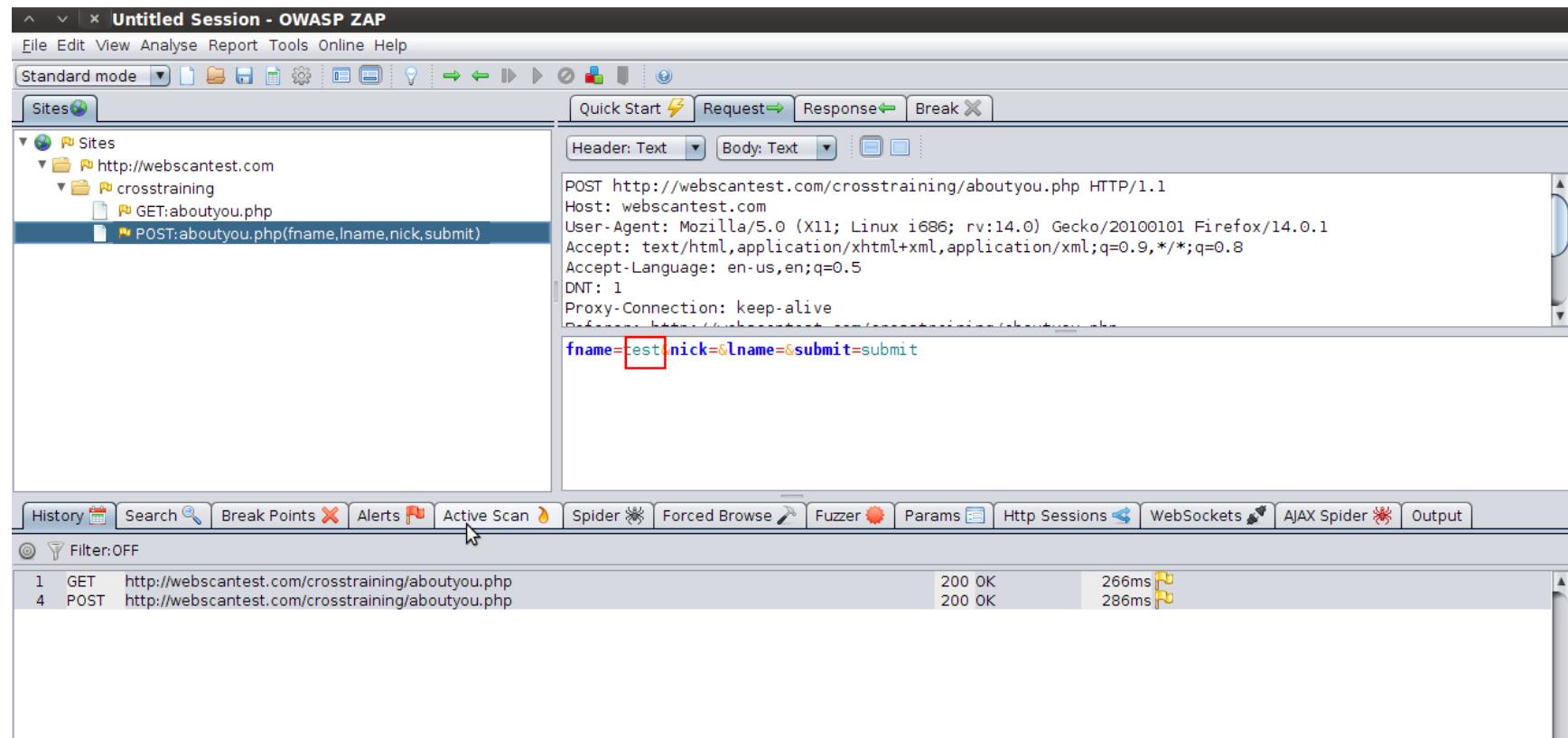
Last Name:

submit

The form based credentials are testuser/testpass, and the HTTP Basic credentials are btestuser/btestpass.

[Privacy Policy](#)

5 Switching to ZAP, notice which parameter was written with the word test



The screenshot shows the OWASP ZAP interface in Standard mode. The left sidebar displays a tree structure under 'Sites' with a node for 'http://webscantest.com/crosstraining'. Underneath it, there are two entries: 'GET:aboutyou.php' and 'POST:aboutyou.php(fname, lname, nick, submit)'. The 'POST' entry is selected. The main pane shows the request details. The 'Header: Text' section contains the following headers:

```
POST http://webscantest.com/crosstraining/aboutyou.php HTTP/1.1
Host: webscantest.com
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
DNT: 1
Proxy-Connection: keep-alive
```

The 'Body: Text' section shows the POST data: `fname=test&nick=&lname=&submit=submit`. The word 'test' in the 'fname' parameter is highlighted with a red rectangular box. At the bottom of the interface, there is a toolbar with various buttons like History, Search, Break Points, Alerts, Active Scan, Spider, Fuzzer, Params, Http Sessions, WebSockets, AJAX Spider, and Output. Below the toolbar, a status bar shows the results of two recent requests:

Method	URL	Status	Time
1 GET	http://webscantest.com/crosstraining/aboutyou.php	200 OK	266ms
4 POST	http://webscantest.com/crosstraining/aboutyou.php	200 OK	286ms

6. right click the parameter to be tested, (test in this case) and select fuzz

File Edit View Analyse Report Tools Online Help

Standard mode Sites Quick Start Request Response Break

Sites http://webscantest.com crosstraining GET:aboutyou.php POST:aboutyou.php(fname, lname, nick, submit)

Header: Text Body: Text

POST http://webscantest.com/crosstraining/aboutyou
Host: webscantest.com
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:14.0)
Accept: text/html, application/xhtml+xml, applicatio
Accept-Language: en-us,en;q=0.5
DNT: 1
Proxy-Connection: keep-alive

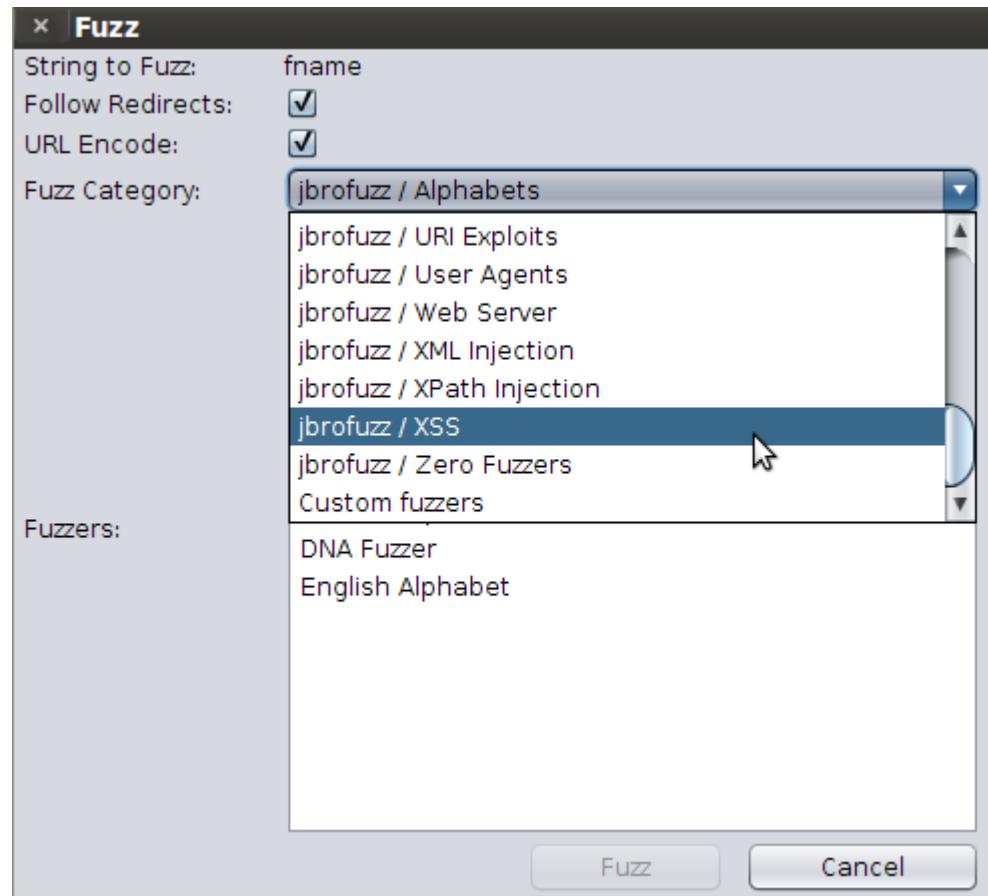
fname=test&nick=&lname=&submit=submit

Find...
Encode/Decode/Hash...
Fuzz
Syntax
View
Can't Undo Ctrl+Z
Can't Redo Ctrl+Y
Cut Ctrl+X

History Search Break Points Alerts Active Scan Params Http

Filter: OFF

7. Scroll down and select jbro/xss list. What this is , is the list of XSS Attack strings that we will automatically pass to the website. This is called "Fuzzing"



8. At the bottom, any instance where the word "reflected" is listed, means that the attack string to the right was successful against the tested parameter

Method	URI	Status	Reason	RTT (ms)	Size	State	Fuzz
POST	http://webscantest.co...	200	OK	194	1015	Reflected	<script>alert('xss')</s...
POST	http://webscantest.co...	200	OK	93	1039	Reflected	<script>alert(string.fr...
POST	http://webscantest.co...	200	OK	97	1019	Reflected	</title><script>alert(...
POST	http://webscantest.co...	200	OK	97	1014	Reflected	'> <script>alert(3)</s...
POST	http://webscantest.co...	200	OK	93	1014	Reflected	'> <script>alert(5)<...>
POST	http://webscantest.co...	200	OK	99	1013	Reflected	> <script>alert(4)</s...
POST	http://webscantest.co...	200	OK	93	1019	Reflected	</title><script>alert(...
POST	http://webscantest.co...	200	OK	93	1020	Reflected	<<script>alert("xss");...
POST	http://webscantest.co...	200	OK	94	989	Reflected	>"

Proof of Lab

- **Proof of Lab Instructions:**
 1. Do a <PrtScn> of all input, and results
 2. Paste into a word document
 3. Email to me
-

Questions:

1. What is fuzzing?
2. What are two ways to defeat an XSS filter?
3. What command is used to launch ZAP?