

How to test for DOM-based cross-site scripting

The majority of DOM XSS vulnerabilities can be found quickly and reliably using Burp Suite's [web vulnerability scanner](#). To test for DOM-based cross-site scripting manually, you generally need to use a browser with developer tools, such as Chrome. You need to work through each available source in turn, and test each one individually.

Testing HTML sinks

To test for DOM XSS in an HTML sink, place a random alphanumeric string into the source (such as `location.search`), then use developer tools to inspect the HTML and find where your string appears. Note that the browser's "View source" option won't work for DOM XSS testing because it doesn't take account of changes that have been performed in the HTML by JavaScript. In Chrome's developer tools, you can use `Control+F` (or `Command+F` on MacOS) to search the DOM for your string.


For each location where your string appears within the DOM, you need to identify the context. Based on this context, you need to refine your input to see how it is processed. For example, if your string appears within a double-quoted attribute then try to inject double quotes in your string to see if you can break out of the attribute.

Note that browsers behave differently with regards to URL-encoding, Chrome, Firefox, and Safari will URL-encode `location.search` and `location.hash`, while IE11 and Microsoft Edge (pre-Chromium) will not URL-encode these sources. If your data gets URL-encoded before being processed, then an XSS attack is unlikely to work.

Dom Lab

Bang in some random characters into the search field (input)

0 search results for
'sfsdfsafsdfasdfsdfasdfsdfasfwwfsegehhjadhga'



Using developer tools, track down the string and check the tags.

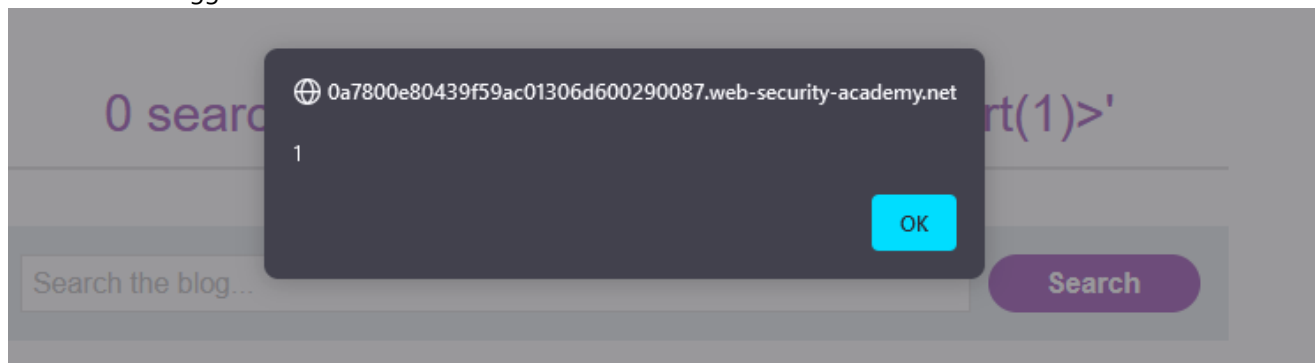
```
</section>
▶ <script> ... </script>

▶ <section class="blog-list"> ... </section>
</div>
</section>
</div>
```

We want to try and escape the img tags and trigger an alert. We can use the following tag to do so.

```
"><svg onload=alert(1)>
```

Alert box has triggered when the search button is clicked. This is DOM-XSS



Congratulations, you solved the lab!