Simditor 与 xss 绕过

前言：

最近做测试的时候遇到一个 xss，输出的内容被实体化编码了但仍然造成了 xss，于是研究了一下原因，最终定位到了所用编辑器 Simditor 上。

Simditor 的简介：

Simditor 是团队协作工具 Tower 使用的富文本编辑器， 功能精简，加载快速。使用方法很简单，首先引入依赖，然后实例化 Simditor 类，再将需要使用编辑器的文本域 id 设置为实例化的类即可。

```
<link rel="stylesheet" type="text/css" href="[style path]/font-awesome.css"
/>
<link rel="stylesheet" type="text/css" href="[style path]/simditor.css" />
<script type="text/javascript" src="[script path]/jquery.min.js"></script>
<script type="text/javascript" src="[script path]/module.js">
</script><script type="text/javascript" src="[script path]/uploader.js">
</script><script type="text/javascript" src="[script path]/simditor.js"></sc
ript>

<textarea id="editor" placeholder="Balabala" autofocus>这是文本</textarea>

<script>
var editor = new Simditor({
  textarea:  $('#editor'),
});
</script>
```

XSS 绕过：

造成 xss 绕过的根本原因是 jquery 的自解码机制，先来看看一个小例子。示例代码如下：

```
<textarea id="editor" placeholder="Balabala" autofocus>&#x3c;svg
onload=alert()&#x3e;</textarea>
```

```
<script>

alert('innerhtml --'+ document.getElementById("editor").innerHTML);

alert('value --' + document.getElementById("editor").value);

alert('jquery --' + $('#editor').val());

</script>
```

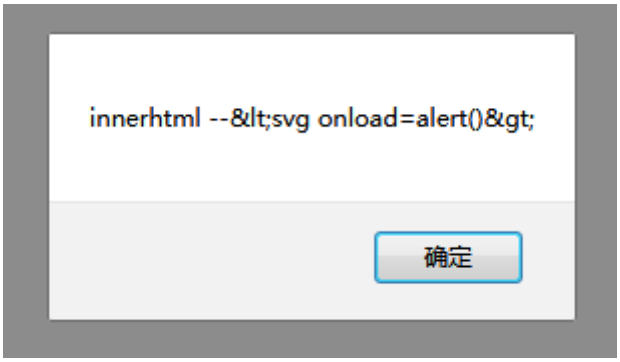
分别用

```
document.getElementById("editor").innerHTML)

document.getElementById("editor").value)

$('#editor').val())
```

三种方法获取 textarea 文本域中的内容，而文本域中的尖括号经过了实体化处理。测试结果如下：



document.getElementById("editor").value) 和 $('#editor').val())两种方法获取的值都经过了解码处理，而 simditor 官方说明中是使用 jQuery 的方式获取对象的（也就是$('#editor')）。

目前主流的防 xss 的方式有实体化编码输出和关键字处理（如去掉尖括号，onload 等事件关键字）两种，但单独使用这两种方式在 simditor 编辑器中均会被绕过。

Ps：以下说的输入不是指在编辑器框中输入，而是直接提交到后端的数据，也就是抓包输入的数据，因为编辑器前端会进行一些编码处理

方式一：实体化编码输出

绕过：输入 payload

<svg onload=alert(1)>

后端经过实体化处理输出到前端页面，编辑器获取文本域时解码处理让 payload 被还原，造成 xss。

方式二：关键字处理

绕过：将 payload 进行 html 编码处理

<svg onload=alert()>

这样 payload 中不含检测的关键字，后端不做处理，编辑器加载文本域时解码操作，还原的 payload，造成 xss。

正确防御方式：

可以看到，两种防御方式分开使用均会被绕过，正确的防御方式也很简单，两种方式一起使用即可。

总结：

有两个关键点，第一点是 payload 输入为抓包直接输入；第二点是能调用 simditor 编辑器编辑自己输入 payload。测试了 ckeditor 无此问题，其他编辑器有没有类似的情况没有测试。