SQLMAP JSON格式检测

坏虾 / 2017-07-10 01:43:00 / 浏览数 5957 安全技术 技术讨论 顶(0) 踩(0)

现在越来越多的网站采取RESTFUL架构，传输的数据多是JSON格式，那么会对传统的安全测试有什么改变呢？

首先，我们模拟一个这种网站试试。

```
import web
import json
import requests

urls = (
■    '/', 'hello'
)
app = web.application(urls, globals())

class hello:
■    def POST(self):
■        payload = web.input()
■        data = web.data()
■        data = json.loads(data)
■        print data
■        #payload = {"artist": "1"}
■        url = '<http://testphp.vulnweb.com/artists.php?artist='>; + data["artist"]
■        print url
■        req = requests.get(url)
■        return req.content

if **name** == "**main**":
■    app.run()
```

然后我们提交正常的请求。

可以看到网站正常响应。

然后使用burpsuite捕获数据包。

```
POST / HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 14
Connection: close

{"artist":"1"}
```

使用SQLMAP -r 进行注入。
SQLMAP会自动识别JSON格式SQL注入。

成功注入。

对于这些JSON数据，也一样可以使用FUZZ的手段进行漏洞挖掘。
之前就在我们公司的一个产品中发现，恶意修改id的值，造成数据处理异常的问题。

下面开始谣言粉碎。

百度搜索 ：SQLMAP注入json格式数据

嗯，出现一堆文章，但是这种直接 --data 然后加星号的手段有效么？

```
C:\Users\xi4okv>sqlmap  -u "[http://127.0.0.1:8080/"  --data={"artist":"1*"}
```

不好意思，在服务端直接报错，数据格式不识别的错误。所以http://127.0.0.1:8080/%22%C2%A0%C2%A0--data=%7B%22artist%22:%221*%22%7D[/code]%E4%B8
--data 是不能直接提交JSON格式数据的。因为服务端只接受json格式的数据。

```
File "D:\Python27\lib\json__init__.py", line 339, in loads
■     return _default_decoder.decode(s)
 File "D:\Python27\lib\json\decoder.py", line 364, in decode
■     obj, end = self.raw_decode(s, idx=_w(s, 0).end())
 File "D:\Python27\lib\json\decoder.py", line 382, in raw_decode
■     raise ValueError("No JSON object could be decoded")
ValueError: No JSON object could be decoded

127.0.0.1:6338 - - [10/Jul/2017 09:38:48] "HTTP/1.1 POST /" - 500 Internal Server Error
```

所以以后遇到json格式的数据，想检测SQL注入，请使用 sqlmap -r的操作。 还有sqlmapapi怎么调用检测json，请各位思考。

本文描述的是个人见解。或许会有错误，欢迎评论中指出。

谢谢大家。

上一篇：【译】黑夜的猎杀-盲打XXE 下一篇：通过样本分析之四CVE-2012-...

1. 1 条回复

CGY 2019-07-05 09:54:55

```
python sqlmap.py -u x --data="{\"aid\":\"7*\",\"numid\":218518900}"
```

0 回复Ta