唯快不破的分块传输绕WAF

原创 队员编号042 酒仙桥六号部队 1周前

这是 **酒仙桥六号部队** 的第 **42** 篇文章。
全文共计1595个字，预计阅读时长6分钟。

1 前言

某重保项目，需要进行渗透，找到突破口，拿起sqlmap一顿梭，奈何安全设备在疯狂运转，故祭起绕过注入的最强套路–分块传输绕过WAF进行SQL注入。安全人员当然安全第一，拿到渗透授权书，测试时间报备等操作授权后：

2 神马探测

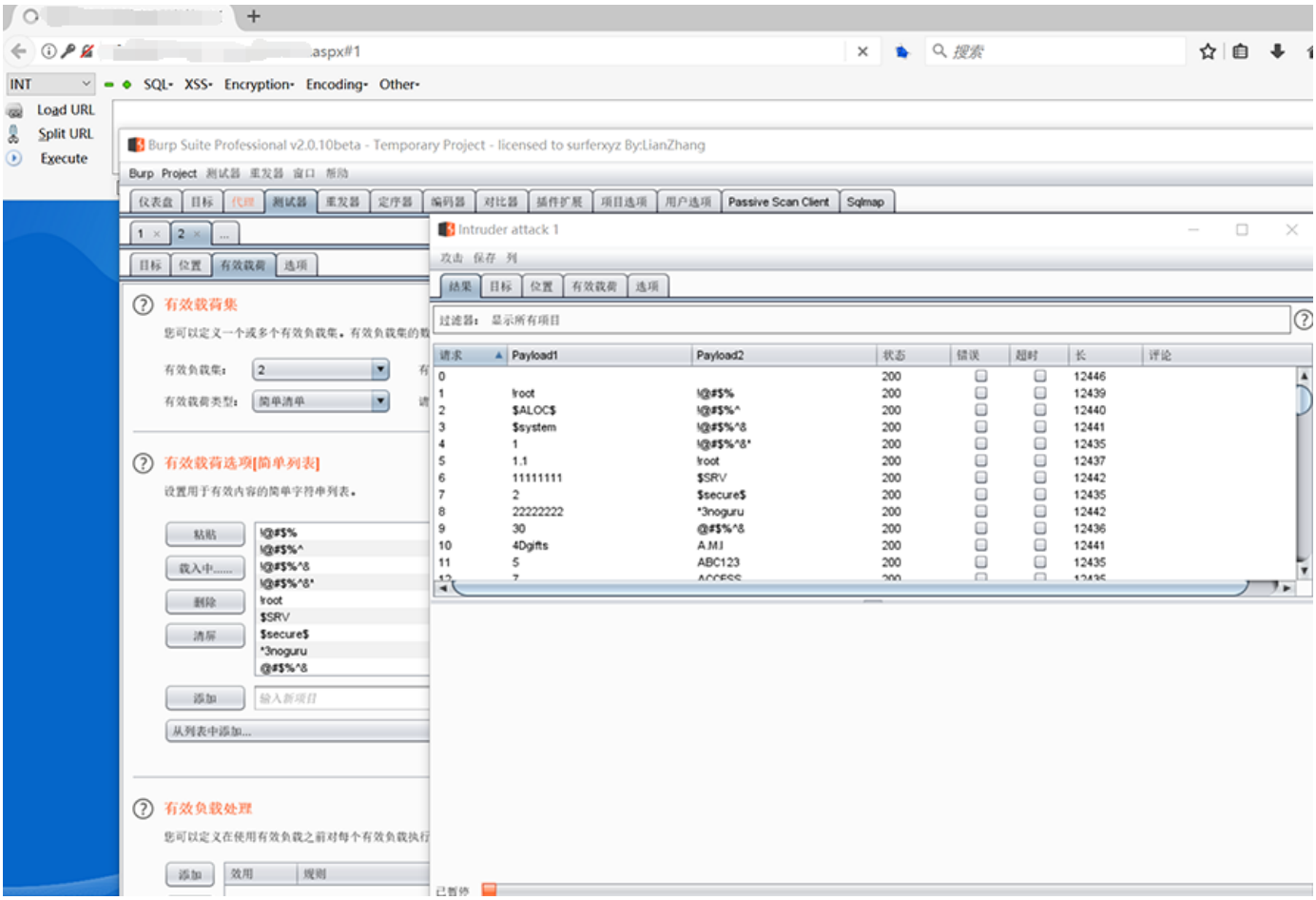
因为客户授权的是三个段，资产众多，且时间紧张，多工具搭配同时进行资产探测。故先对三个段使用资产探测神器goby和端口神器nmap一顿怼，还有静悄悄不说话的主机漏扫神器Nessus。因此也就结合探测出来的ip和端口及其他资产详情，信息探测进行时，先根据目前得到的web网站一顿梭。在浏览器输入IP+端口，滴，开启web世界。喝了一口肥宅快乐水并咪咪眼开始端详起这几个web网站。

界面是这个样子：

定睛一看，先抓个包跑跑注入，神器sqlmap一片红。卒，遂放弃。

```
[17:39:20] [CRITICAL] unable to connect to the target URL ('')
[17:39:20] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:20] [CRITICAL] unable to connect to the target URL ('')
[17:39:20] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:20] [CRITICAL] unable to connect to the target URL ('')
[17:39:21] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace'
[17:39:21] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:21] [CRITICAL] unable to connect to the target URL ('')
[17:39:21] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:21] [CRITICAL] unable to connect to the target URL ('')
[17:39:21] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[17:39:21] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:21] [CRITICAL] unable to connect to the target URL ('')
[17:39:21] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:21] [CRITICAL] unable to connect to the target URL ('')
[17:39:21] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
[17:39:21] [CRITICAL] unable to connect to the target URL ('')
[17:39:21] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[17:39:21] [CRITICAL] unable to connect to the target URL (''). sqlmap is going to retry the request(s)
```

再次定睛一看，妥妥的用户登录页面，试试弱口令，burp神器走一波。



嗯，用户名密码可爆破漏洞，提交，收工。

报告提交后，我领导看到后，嗯，如下图：

挨了一顿锤之后，手里的肥宅快乐水不香了，继续努力搬砖吧。

3 继续杠不要怂

作为男子汉，肿么能因为sqlmap一片红就继续放弃呢？是男人就继续用sqlmap杠，这次祭起分块WAF进行绕过。

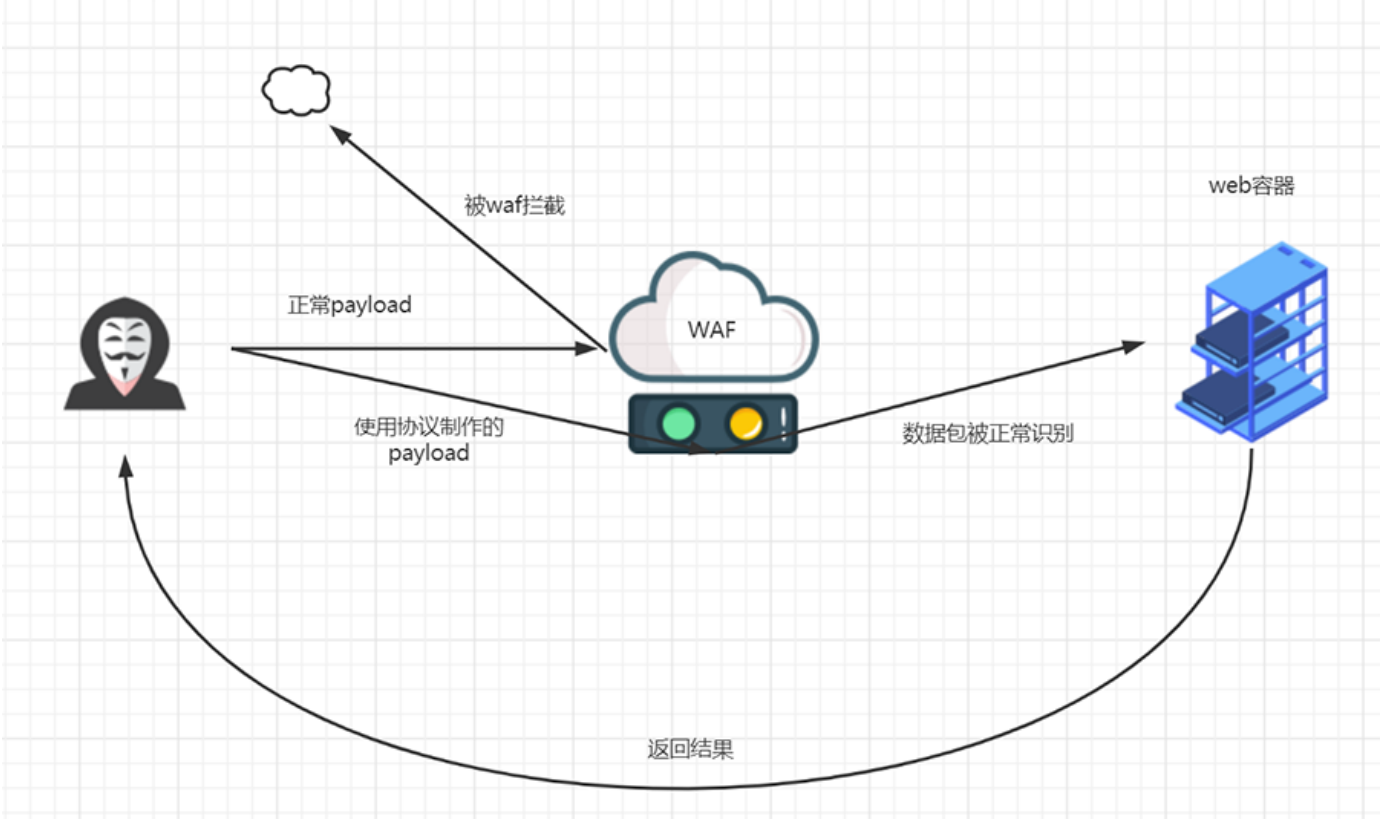
4 what is 分块传输?

分块传输编码（Chunked transfer encoding）是超文本传输协议（HTTP）中的一种数据传输机制，允许HTTP由应用服务器发送给客户端应用（ 通常是网页浏览器）的数据可以分成多个部分。分块传输编码只在HTTP协议1.1版本（HTTP/1.1）中提供。通常，HTTP应答消息中发送的数据是整个发送的，Content-Length消息头字段表示数据的长度。数据的长度很重要，因为客户端需要知道哪里是应答消息的结束，以及后续应答消息的开始。然而，使用分块传输编码，数据分解成一系列数据块，并以一个或多个块发送，这样服务器可以发送数据而不需要预先知道发送内容的总大小。通常数据块的大小是一致的，但也不总是这种情况。

一般情况HTTP请求包的Header包含Content-Length域来指明报文体的长度。有时候服务生成HTTP回应是无法确定消息大小的，比如大文件的下载，或者后台需要复杂的逻辑才能全部处理页面的请求，这时用需要实时生成消息长度，服务器一般使用chunked编码。在进行Chunked编码传输时，在回复消息的Headers有Transfer-Encoding域值为chunked，表示将用chunked编码传输内容。

这在http协议中也是个常见的字段，用于http传送过程的分块技术，原因是http服务器响应的报文长度经常是不可预测的，使用Content-length的实体搜捕并不是总是管用。

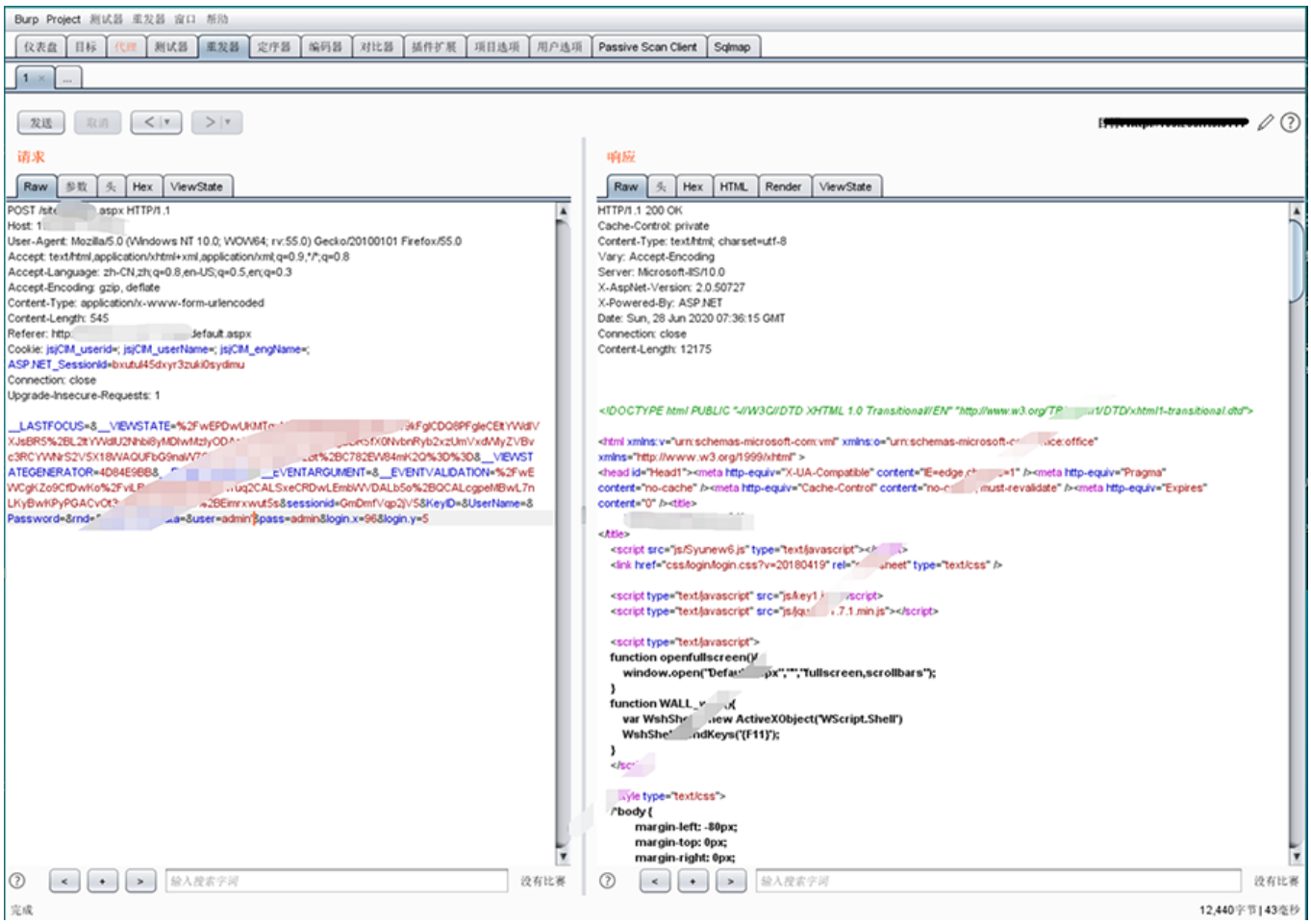
分块技术的意思是说，实体被分成许多的块，也就是应用层的数据，TCP在传送的过程中，不对它们做任何的解释，而是把应用层产生数据全部理解成二进制流，然后按照MSS的长度切成一分一分的，一股脑塞到tcp协议栈里面去，而具体这些二进制的数据如何做解释，需要应用层来完成。



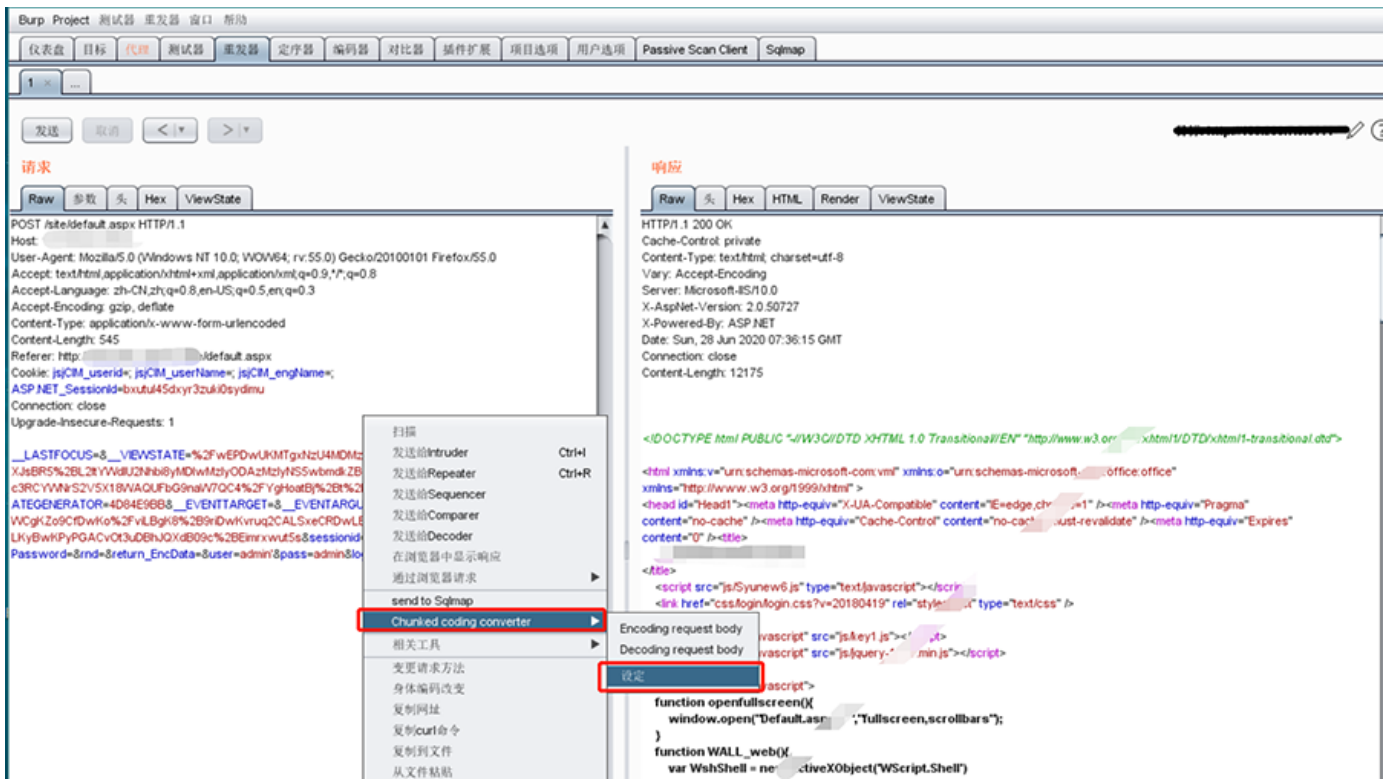
简而言之，就是把数据包分成一块一块的丢过去，骗骗死脑筋的WAF。

5 分块传输开启绕过

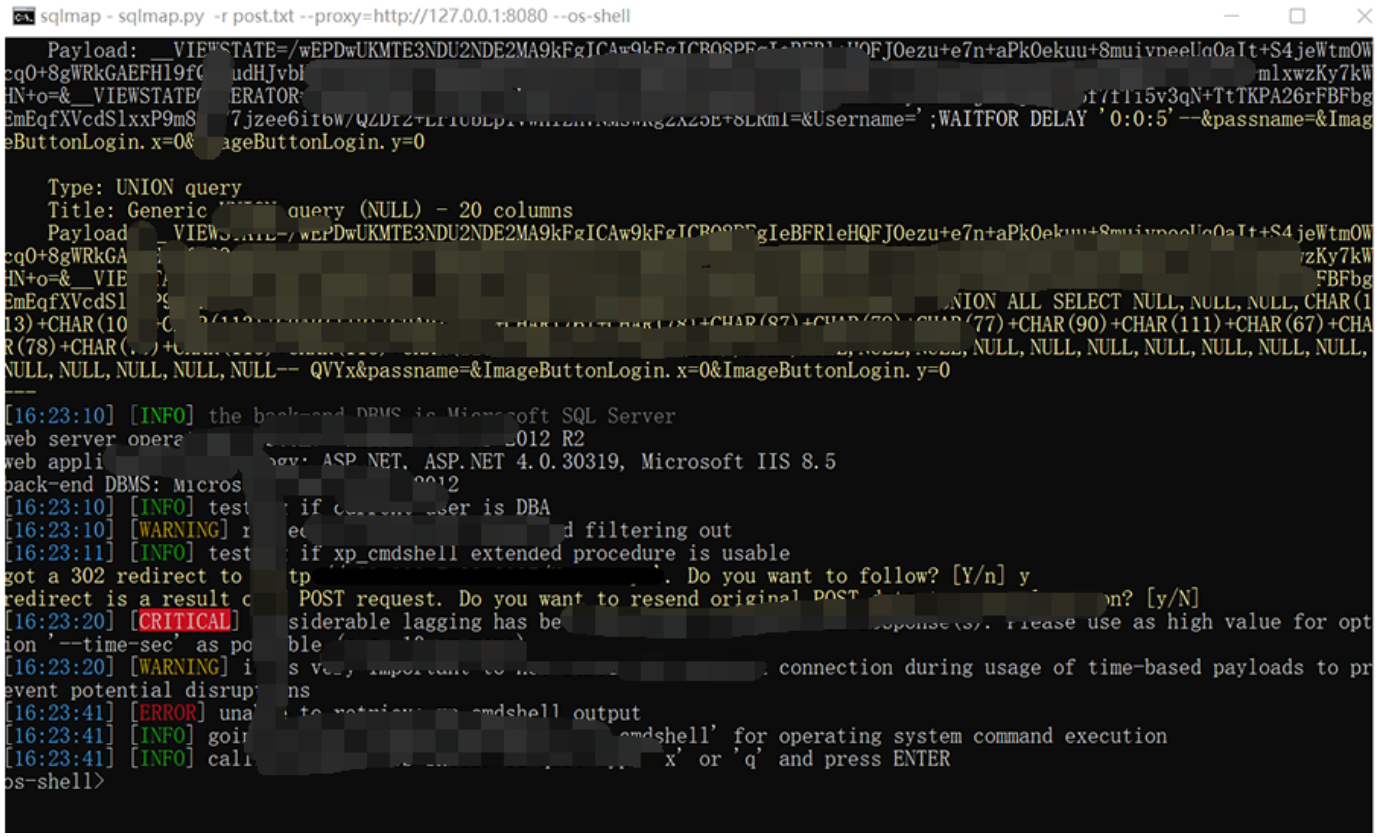
手工进行分块绕过较为繁琐，且花费时间长，面对大量资产的情况，项目时间较为紧张的情况下，还是使用自动化工具来的快捷方便。这里使用sqlmap+burp+burp插件（chunked-coding-converter）。祭出我二表哥工具的项目地址：https://github.com/c0ny1/chunked-coding-converter。快速使用：burp获取post包后，复制post包，做成post.txt,并放置于sqlmap工具文件下。（忽略在下负一级的打马赛克技术）



使用burp 设定插件，开启插件代理：



使用Sqlmap进行代理：sqlmap语句sqlmap.py -r post.txt --proxy=http://127.0.0.1:8080 --os-shell



什么？为什么不继续了？因为客户不让了，表演结束了，谢谢大家。

6 让我再多说一句

当然为了更加快速化，和方便快捷一步到位，可使用sqlmap参数batch自动进行注入。

sqlmap.py –r post.txt ––proxy=http://127.0.0.1:8080 —batch

当然，我们再可以提高速度，进行一步到位，可使用sqlmap参数threads提高并发数。

sqlmap.py –r post.txt ––proxy=http://127.0.0.1:8080 ––batch ––threads 10

当当当然可以修改sqlmap配置文件将默认最高10改成9999，具体根据现场实际情况进行修改。

Sqlmap配置文件settings.py，将MAX_NUMBER_OF_THREADS = 9999。

多线程sqlmap效果如下：

Ok，以上是面对大量资产绕过waf进行注入的姿势。

知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你

长按二维码关注 酒仙桥六号部队