

实验13 暴力破解http basic 认证

实验目的

通过实验，使学习者：

- 理解弱口令的危害；
- 掌握使用Burp suite intruder的基本方法；
- 掌握http basic auth的基本原理。

实验内容

- 打开或构建webgoat应用
- 使用burpsuite 攻击webgoat的http基本认证

实验步骤

启动 owasp bwa 靶机

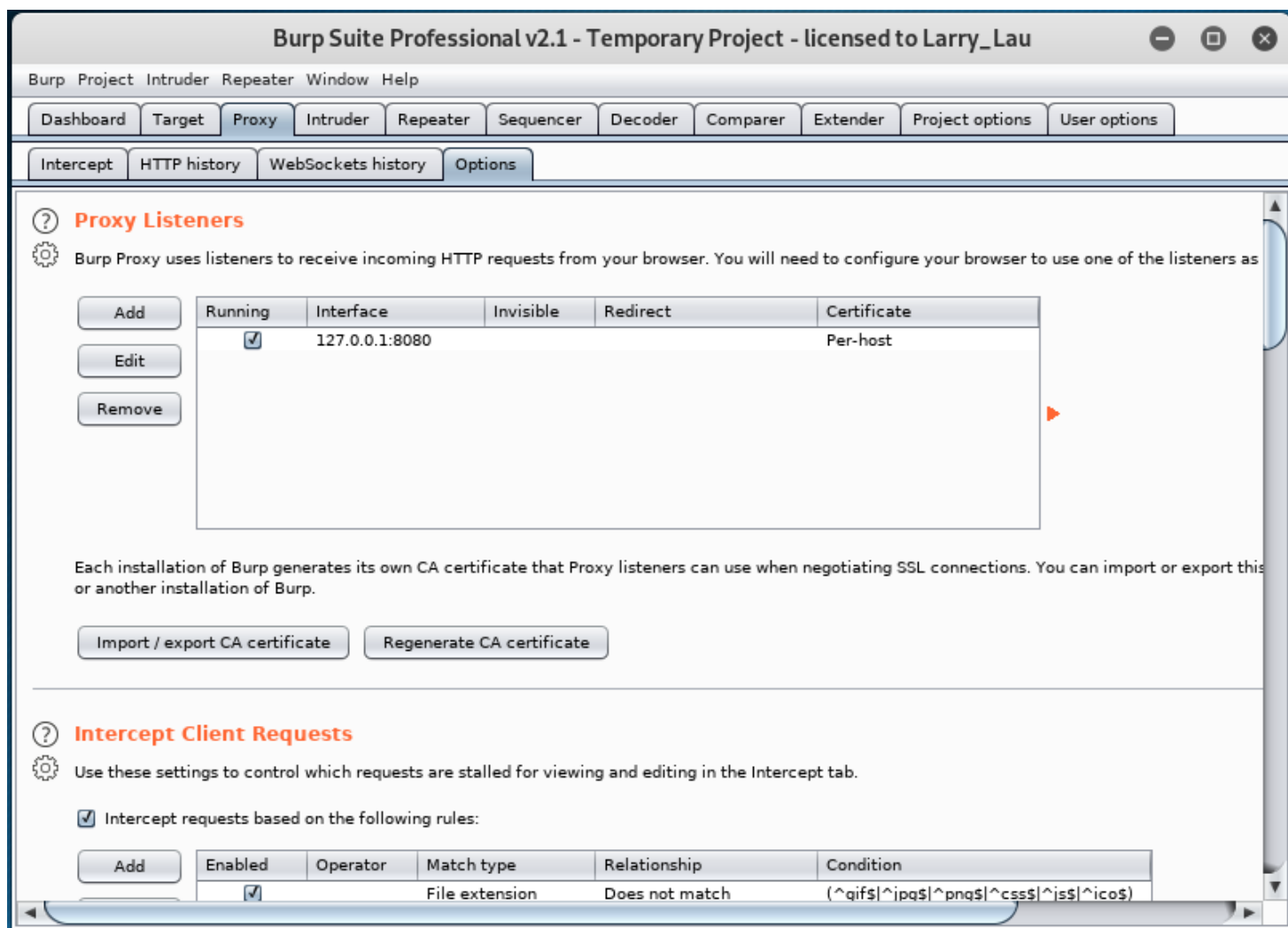
步骤：

1. 启动 owasp bwa v1.2 虚拟机。
2. 以 root用户名登录，密码owaspbwa。
3. 在 owasp bwa 虚拟机中运行ifconfig，确认其的IP地址。下面以10.10.10.135为例。
4. 在 kali linux 虚拟机 或 windows中的浏览器上访问 <http://10.10.10.135/>
5. 进入打开Web页面的Training applications中的OWASP WebGoat

启动 kali 中的 Burpsuite

步骤：

1. 启动Burpsuite pro 2.1 ， 查看器代理设置选项。



2. 打开 kali 中浏览器，以 firefox 为例，设置其 Preferences，找到 network proxy 设置，点击 Settings，按 burpsuite 中选项情况设置连接。

Connection Settings

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy Port

☐ Use this proxy server for all protocols

SSL Proxy Port

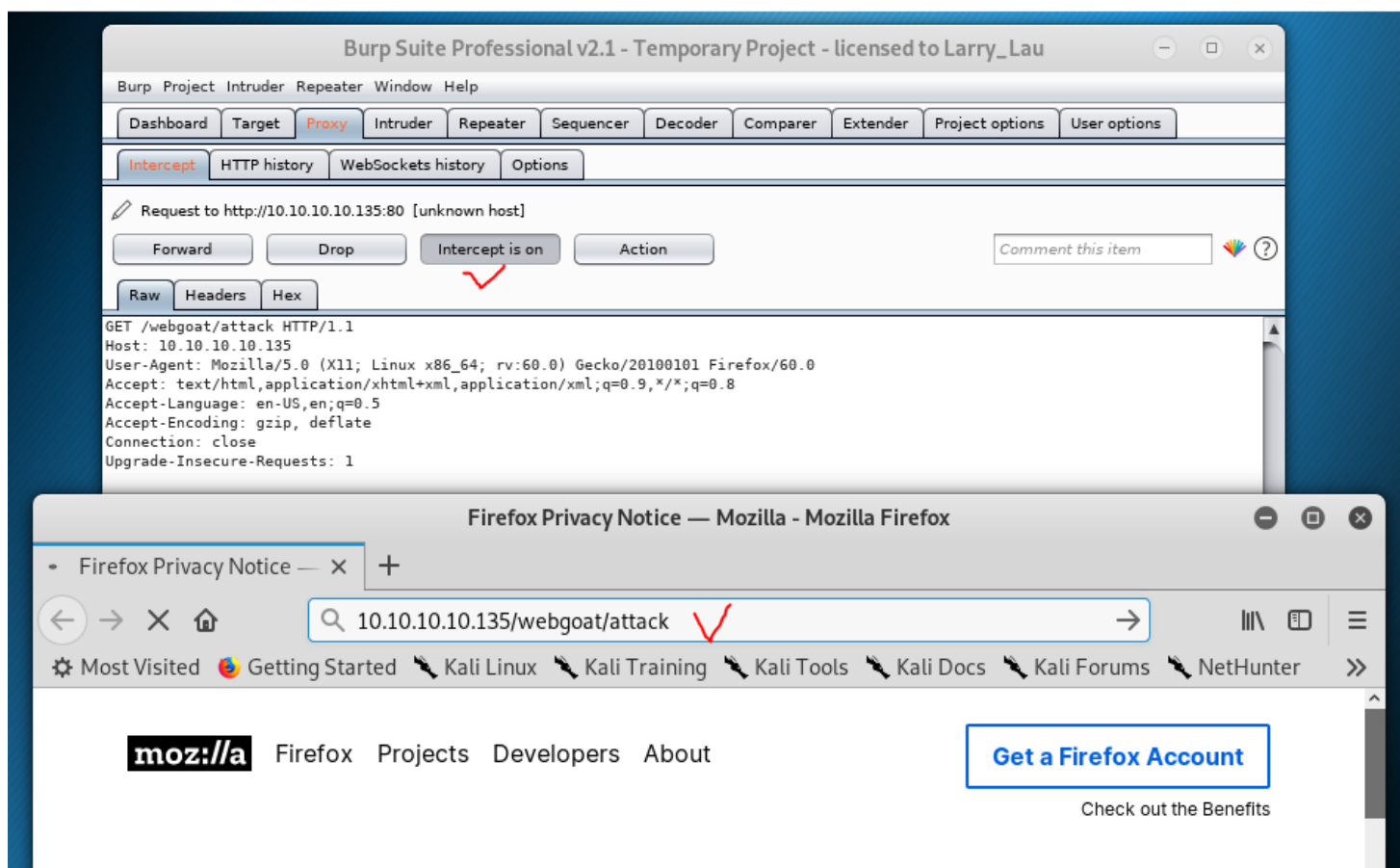
FTP Proxy Port

SOCKS Host Port

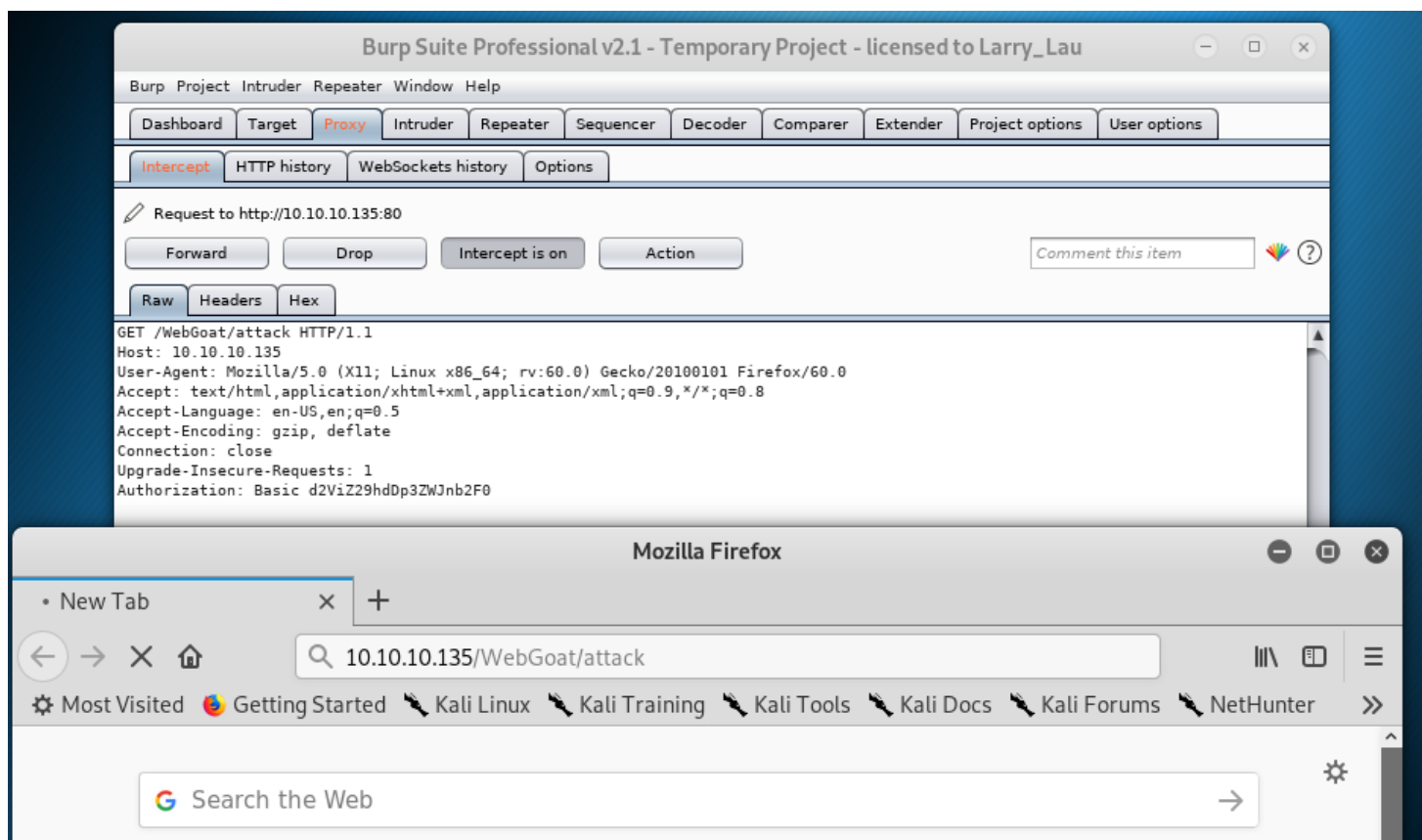
☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

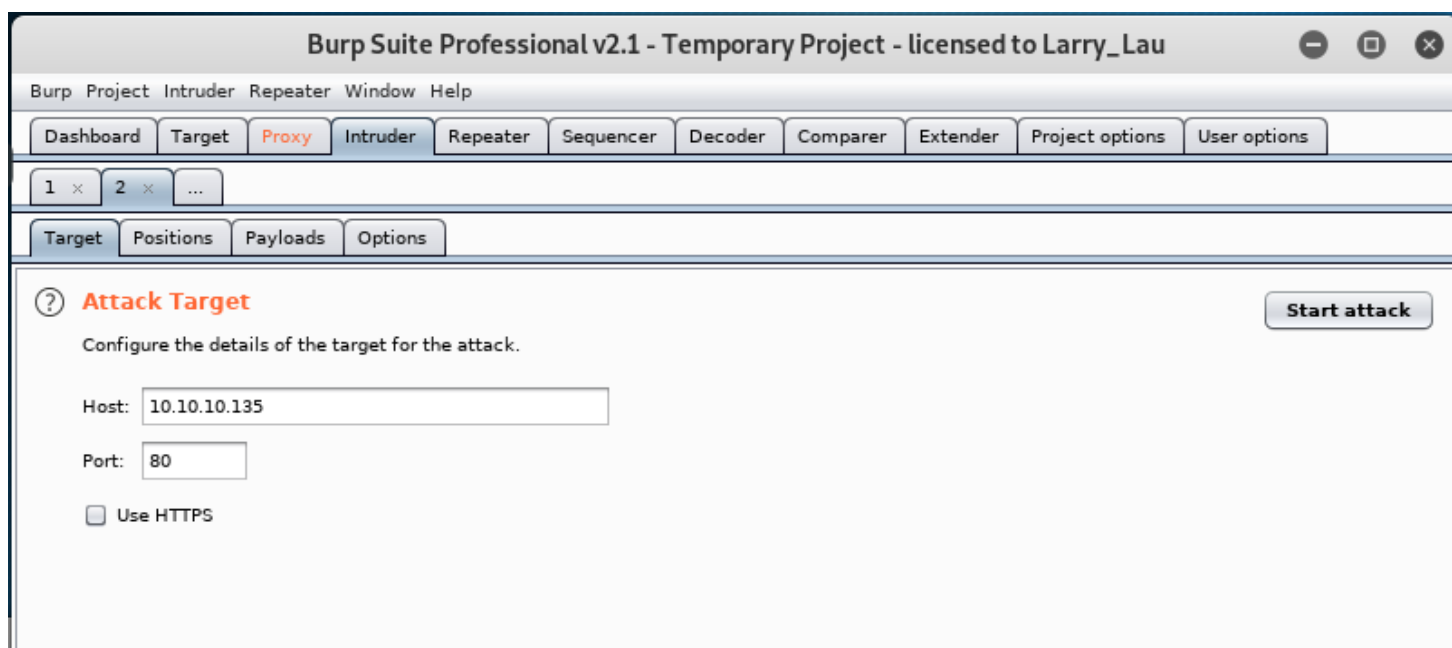
3. 在burpsuite中Proxy启动中断，即“Intercept is on”，然后从kali访问
<http://10.10.10.135/webgoat/attack>



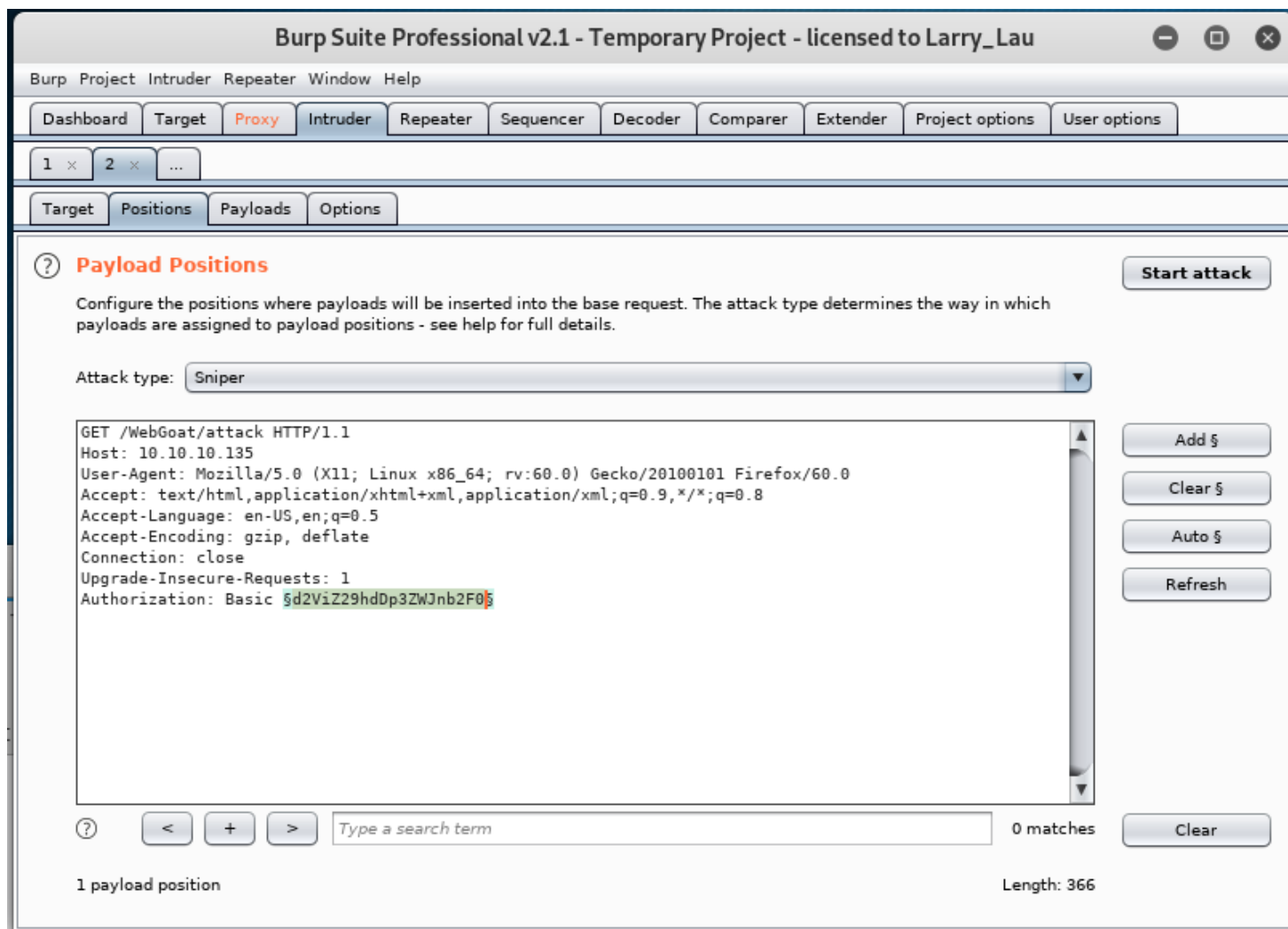
4. 点击 Forward，直到出现类似下图中的http请求。



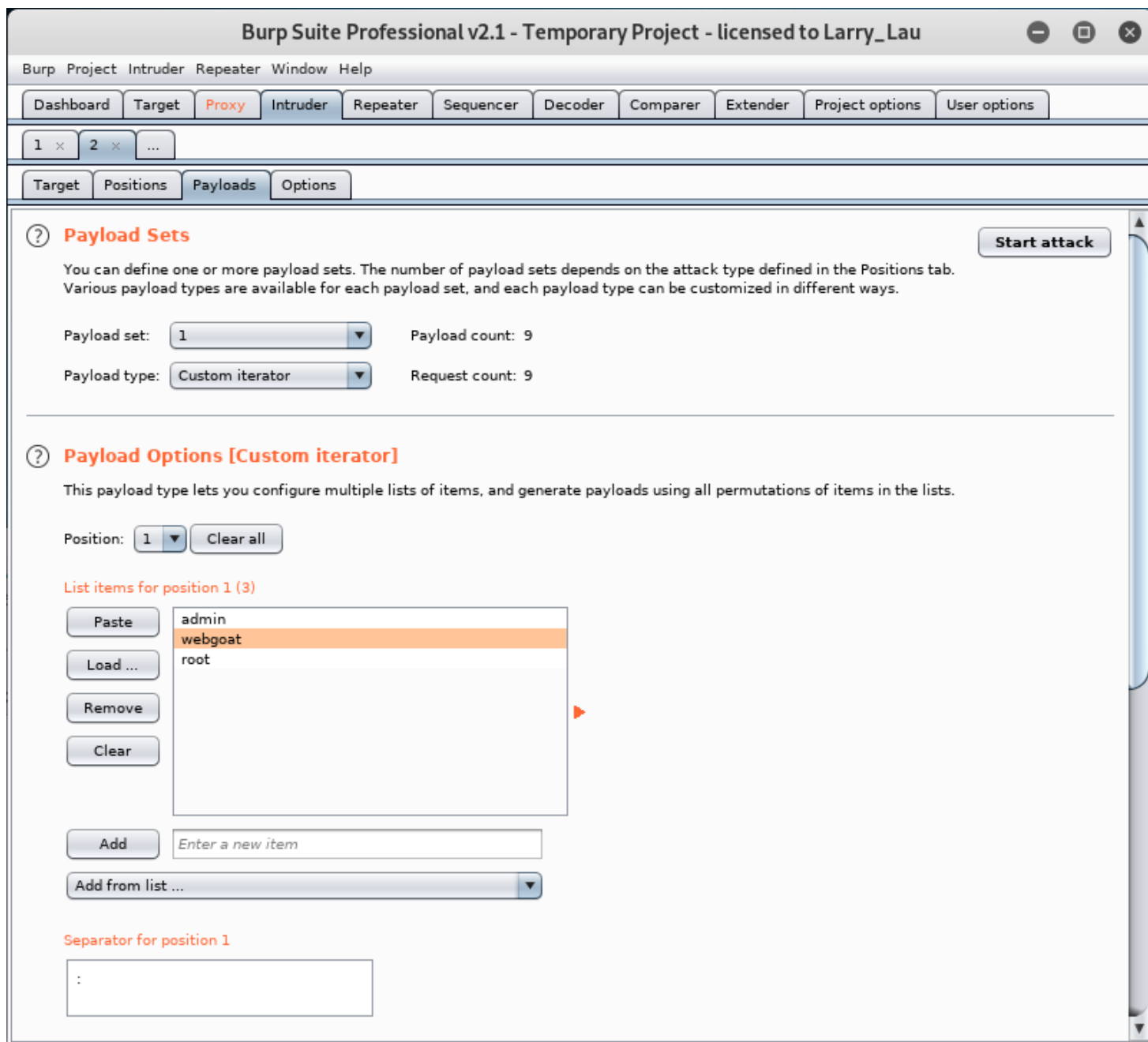
5. 点右键，将这个请求发送到 Send to Intruder（或者点快捷键 Ctrl+I）



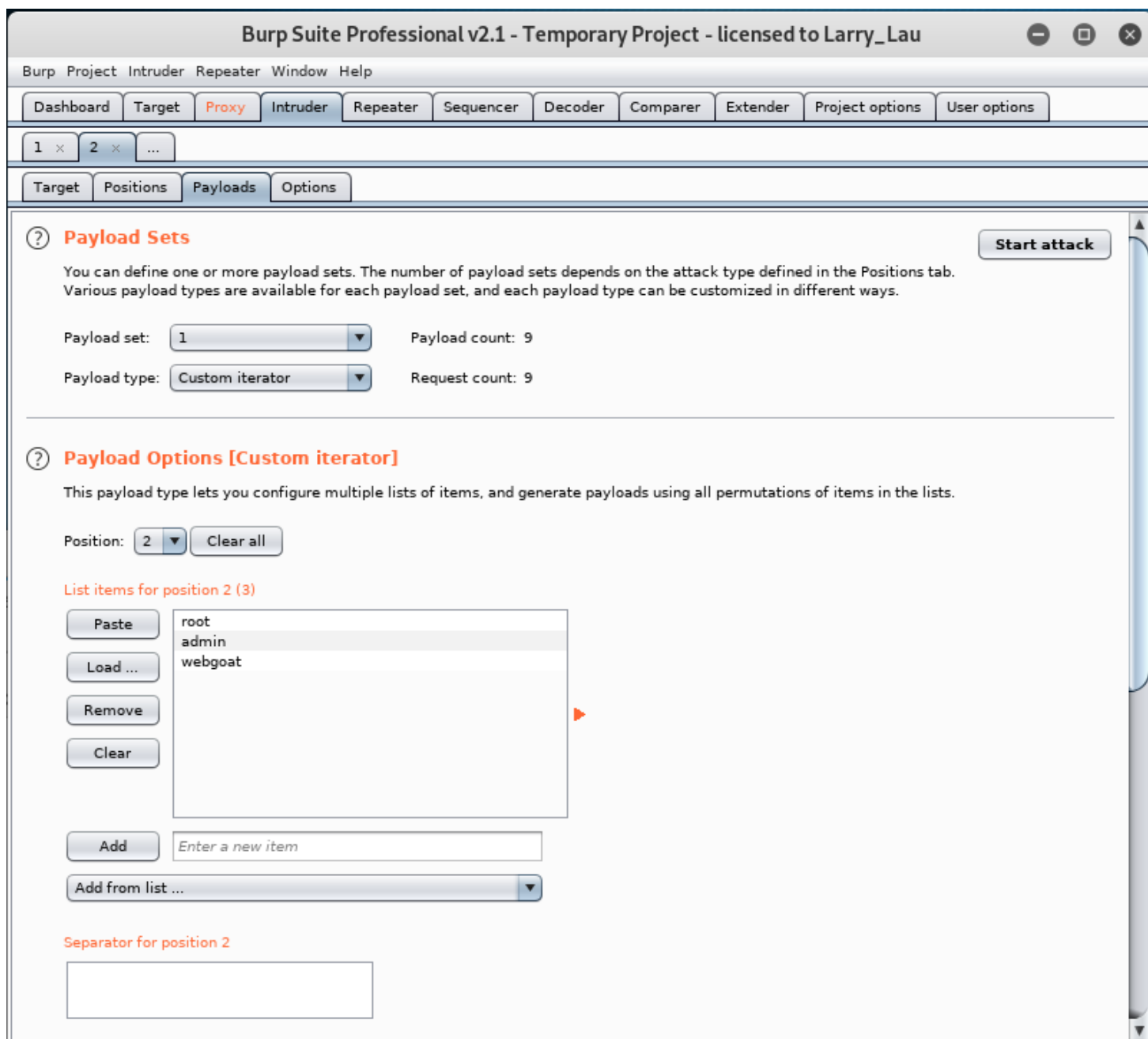
6. 在burp 的intruder的positions中设置注入位置参数。设置位置在Authorization: Basic 后的字段。



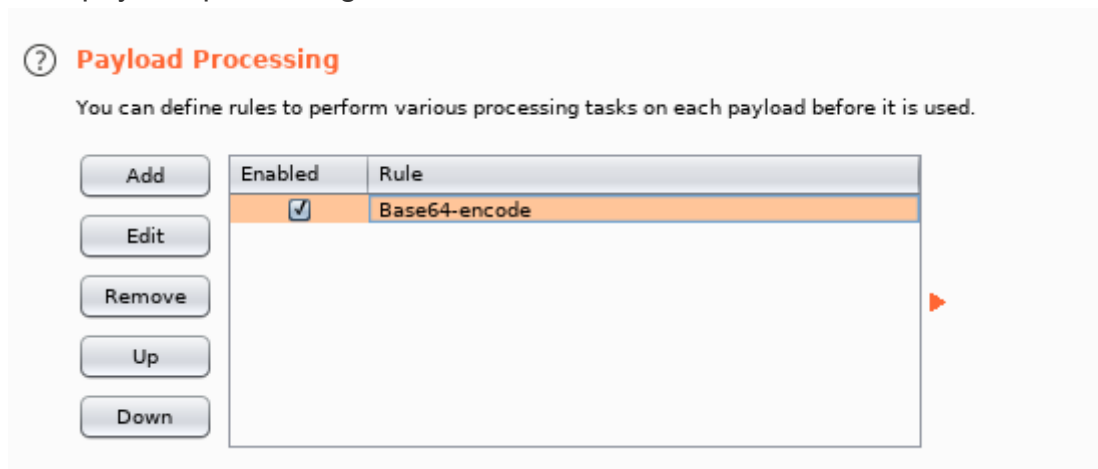
7. 设置burp intruder的payloads,首先选择payload type为"custom iterator"; 然后设置“payload options”的Position 1, 添加可能的用户名, 并在下方Separator for position 1中设置为“:”



8. 设置“payload options”的Position 2, 添加可能的密码。



9. 设置 payload processing，点击add，选择 Encode，然后选择 Base64-encode。



10. 设置 payload Encoding。去除url编码中的=、/、+。这些都是Base64中可能出现的字符。

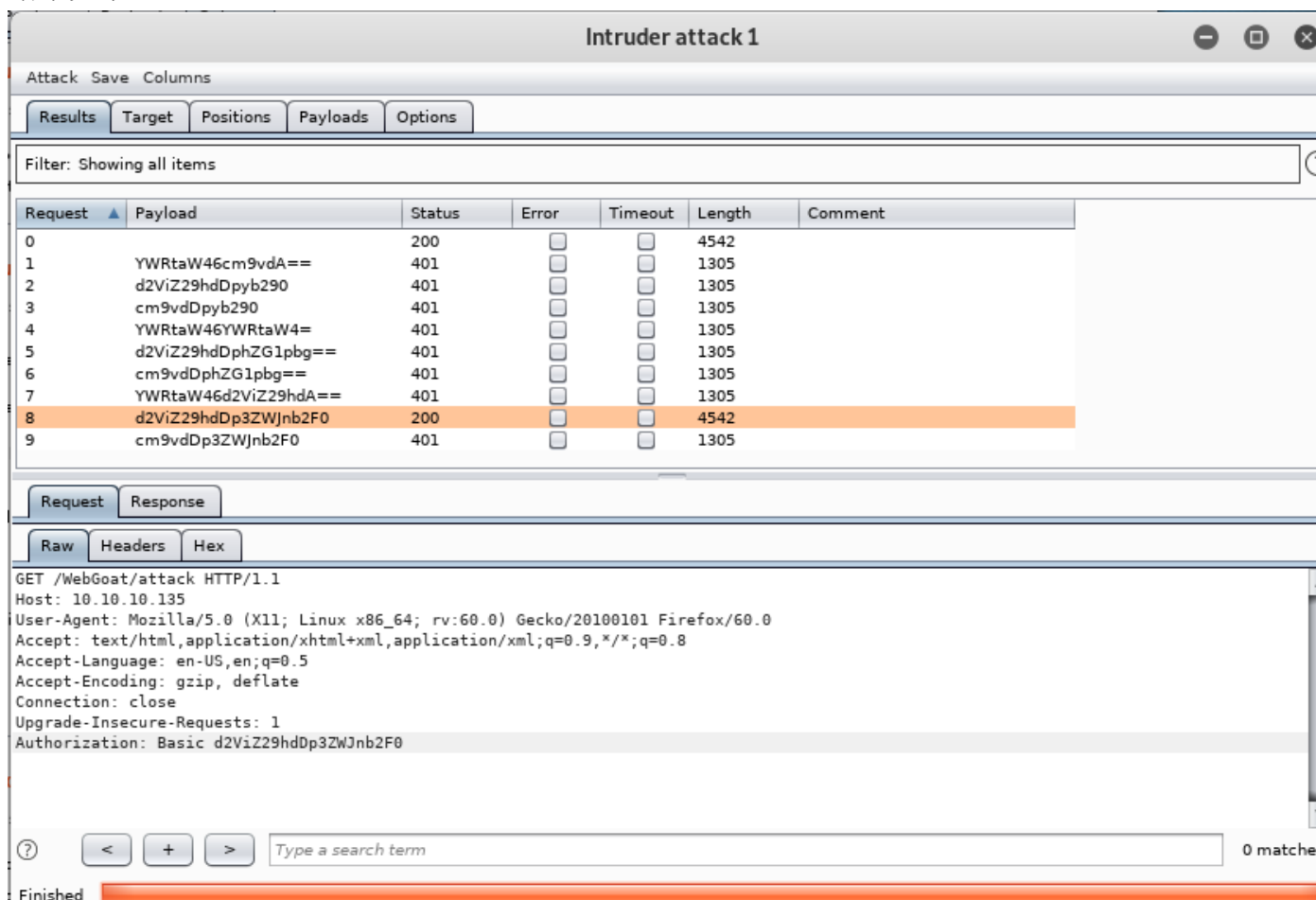
? Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

☒ URL-encode these characters:

11. 运行 start attack，启动 http basic auth 暴力破解攻击。

结果如下：



Intruder attack 1

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4542	
1	YWRtaW46cm9vdA==	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
2	d2ViZ29hdDpyb290	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
3	cm9vdDpyb290	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
4	YWRtaW46YWRtaW4=	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
5	d2ViZ29hdDphZG1pbG==	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
6	cm9vdDphZG1pbG==	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
7	YWRtaW46d2ViZ29hdA==	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	
8	d2ViZ29hdDp3ZWJnb2F0	200	<input type="checkbox"/>	<input type="checkbox"/>	4542	
9	cm9vdDp3ZWJnb2F0	401	<input type="checkbox"/>	<input type="checkbox"/>	1305	

Request Response

Raw Headers Hex

```
GET /WebGoat/attack HTTP/1.1
Host: 10.10.10.135
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
```

0 matches

Finished

实验结论

上面的第8行，payload是d2ViZ29hdDp3ZWJnb2F0，获得响应码为200，说明这个载荷可用，经过Base64解码可知是webgoat:webgoat。