

实验9 Web攻防基础-页面访问与分析

1 实验目的

了解HTTP的基本概念和通信过程。

2 实验内容

- Burpsuite安装
- 设置Burpsuite proxy
- 使用Burpsuite 访问 webgoat5.4

3 实验前提

需要准备下列虚拟机

- kali 2019
- owasp bwa v1.2

4 实验步骤

4.1 BurpSuite pro 2.1 破解版的安装

1.安装 vmware workstation 。

2.安装 kali 虚拟机.

以 2019.02为例,下载地址：<https://mirrors.neusoft.edu.cn/kali-images/kali-2019.2/kali-linux-2019.2-amd64.iso>

3.下载 BurpSuite_pro_v2.1.rar包，我们下发的虚拟机kali2019中已经安装好了，如果有问题可以从百度网盘地址链接：<https://pan.baidu.com/s/1t47Tw11fw8Riu6h0sABAsA> 提取码：wyex 。假设下载并解压到~/Downloads/BurpSuite_pro_v2.1目录下。

4.建立工作目录，复制文件，允许文件运行

```
mkdir /usr/local/BurpSuite_pro_v2.1
```

```
cp ~/Downloads/BurpSuite_pro_v2.1/*.* /usr/local/BurpSuite_pro_v2.1/
```

```
chmod 755 /usr/local/BurpSuite_pro_v2.1/*.*
```

BurpSuite_pro_v2.1中有两个jar包，运行条件是jre 11.0以上，burpsuite_pro_v2.1.jar是主文件，burpsuite_pro_v2.1_BurpHelper.jar是破解文件。

5.改变kali中原有的 /usr/bin/burpsuite 命令

```
mv /usr/bin/burpsuite{,.old}
```

```
ln -s /usr/local/BurpSuite_pro_v2.1/burpsuite_pro_v2.1_BurpHelper.jar /usr/bin/burpsuite
```

6.运行burpsuite

```
burpsuite
```

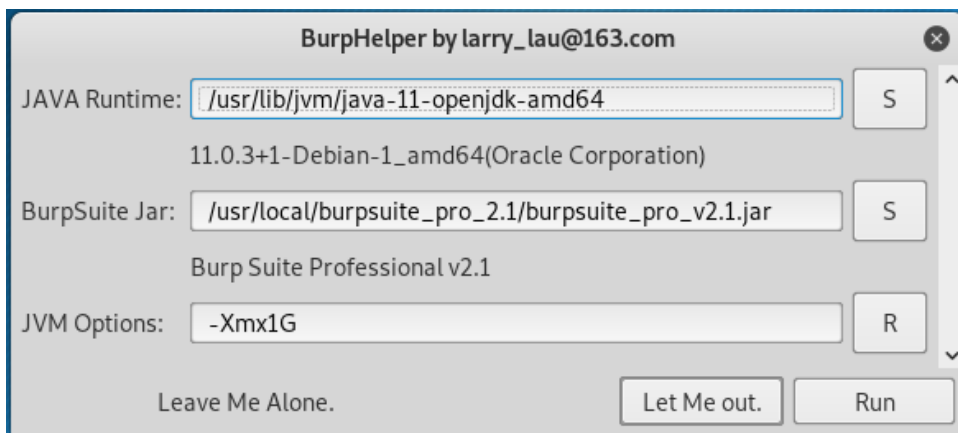
如果显示Burpsuite过期 (Expires)，可以关闭burpsuite pro，然后在命令行允许下列命令：

```
sudo date -s 01/01/2019
```

修改系统时间后，重新启动burpsuite。

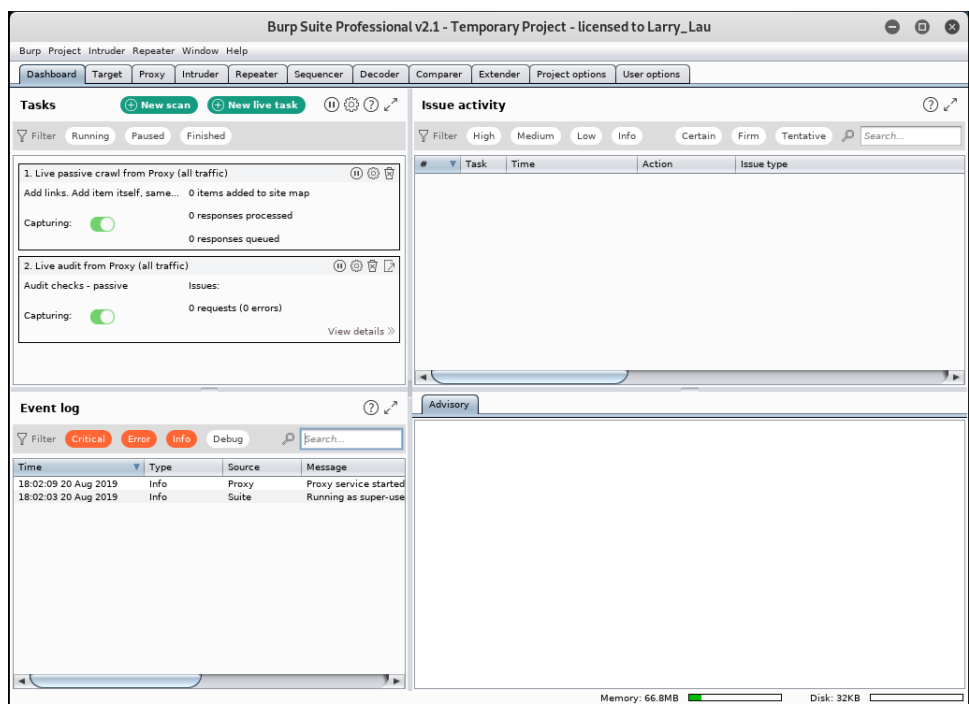
7.设置破解对象

主要是设置中间一行的BurpSuite Jar：



然后点击 Run，之后点击“I accept”，接受协议开始使用。

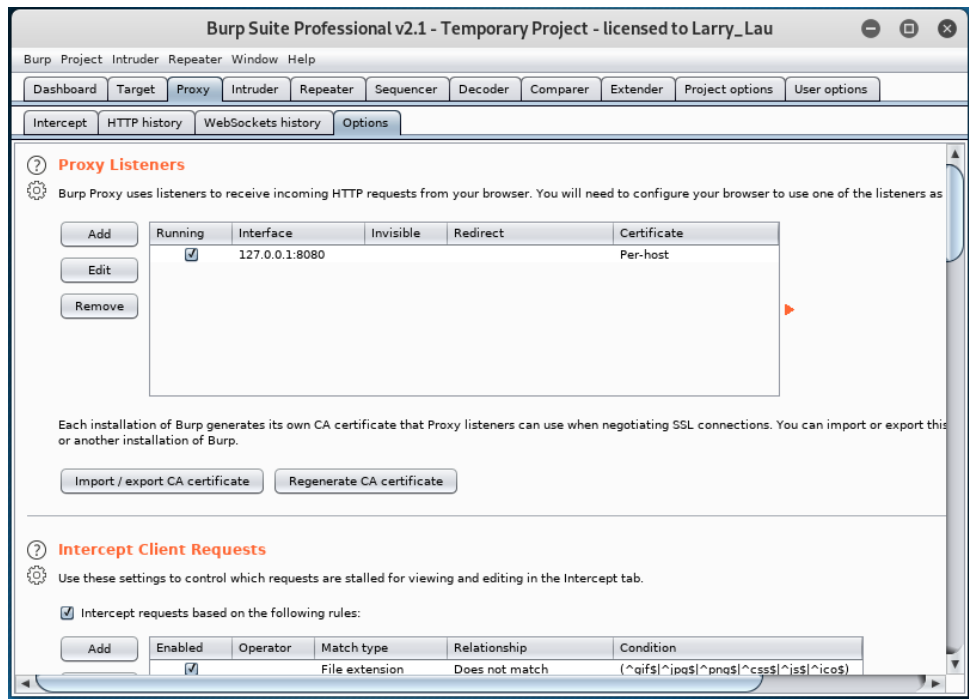
8.查看主界面



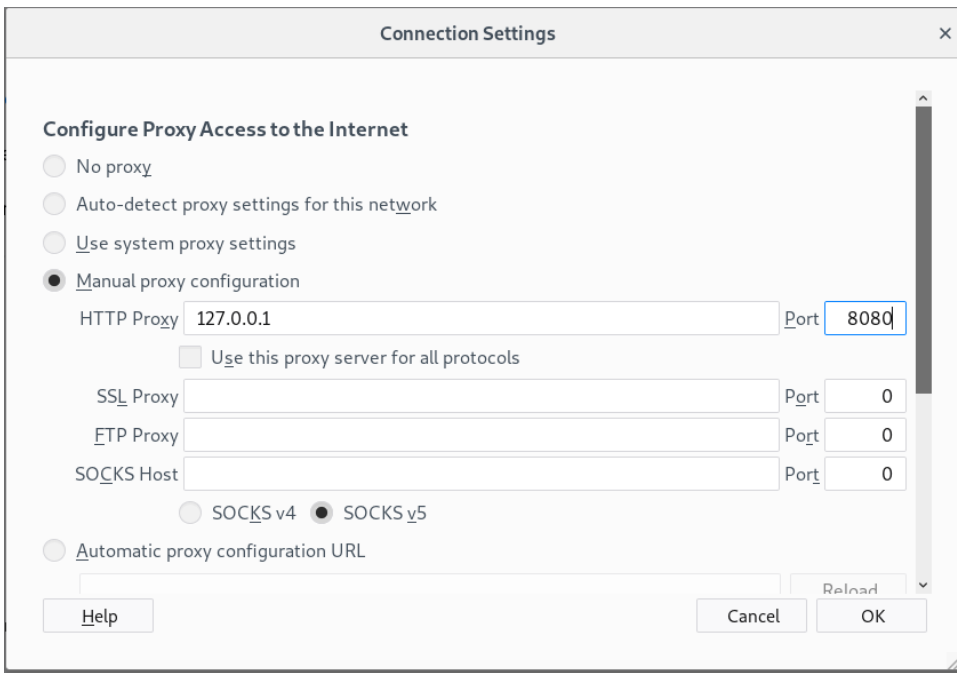
4.2 设置Burpsuite proxy 截获http请求

步骤：

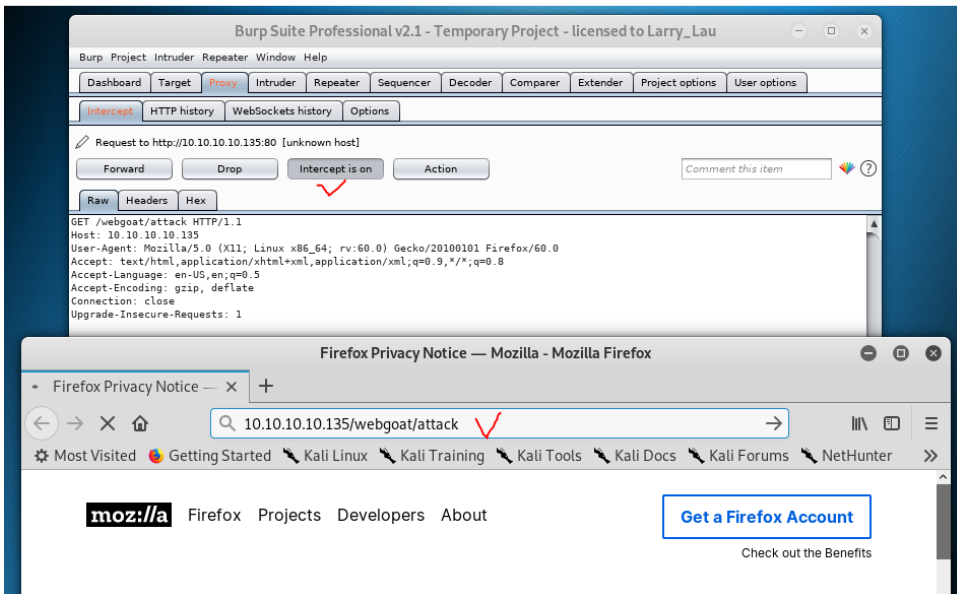
1.启动Burpsuite pro 2.1 ，查看器代理设置选项。



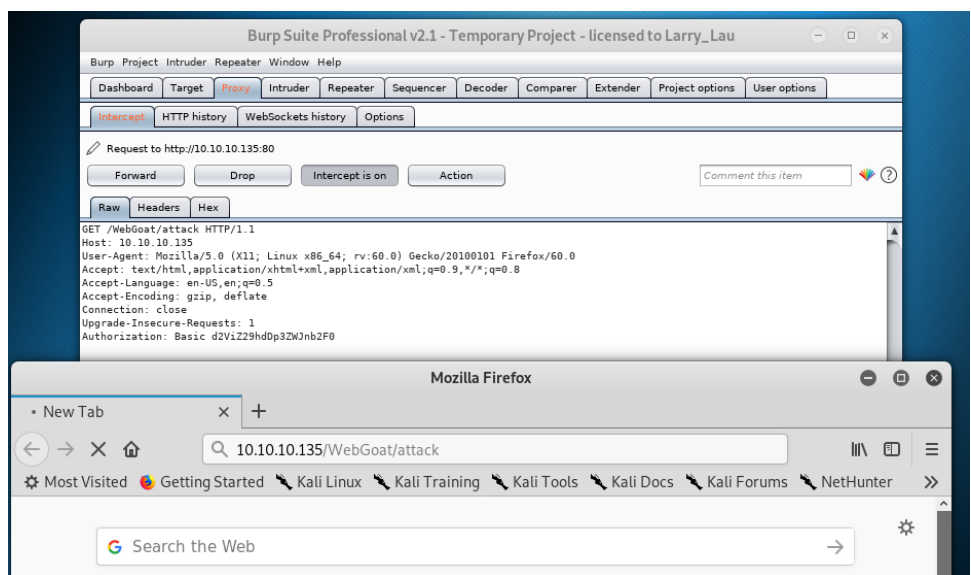
2.打开 kali中浏览器，以firefox为例，设置其Preferences，找到 network proxy设置，点击 Settings，按 burpsuite中选项情况设置连接。



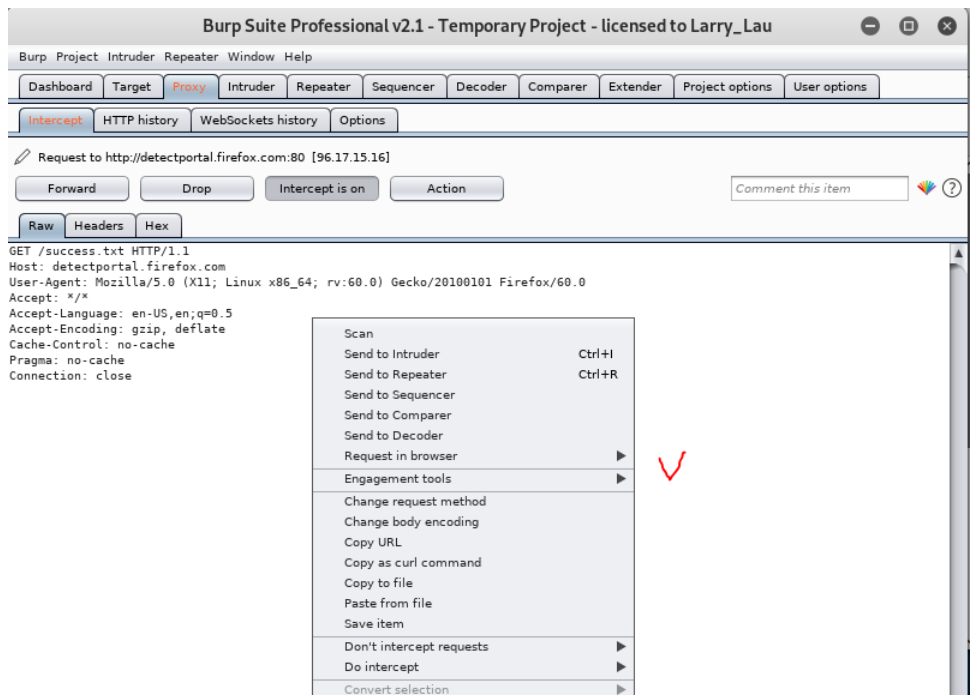
3.在burpsuite中Proxy启动中断，即“Intercept is on”，然后从kali访问某个链接，例如<http://10.10.10.135/webgoat/attack>。



4.点击 Forward，直到出现类似下图中的http请求。



5.对于感兴趣的请求，可以右键点击请求原文，将当前请求发送到其它burp模块。

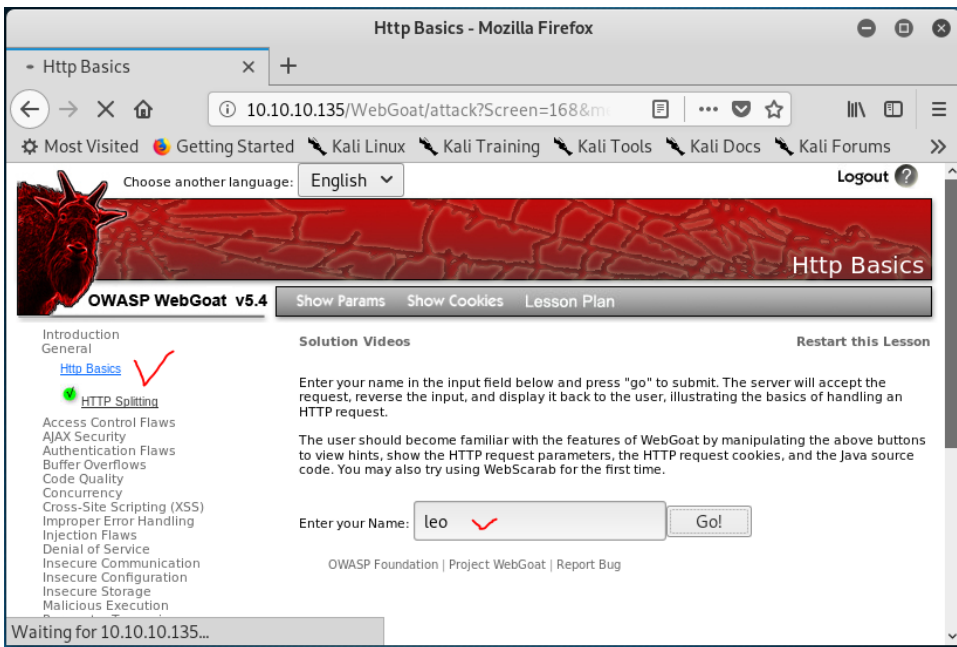


4.3 使用Burpsuite 访问 webgoat5.4

通过访问Webgoat General-Http Basic，掌握Burpsuite工具的基本应用方法，记录并分析访问过程。

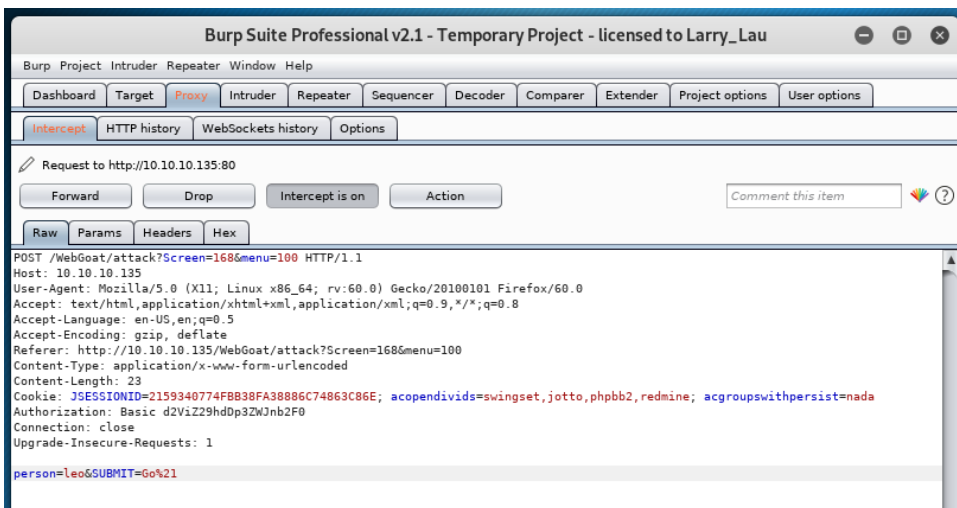
1.打开webgoat网站，点击左侧导航“General -http basics”。在页面输入框中输入任意字符串。但不点“go”。

说明：在下面的输入框中输入您的姓名，然后按“Go！”提交。服务器将接受请求，反转输入，并将其显示回用户，说明处理HTTP请求的基础知识。用户应该通过操作上面的按钮来查看提示、显示HTTP请求参数、HTTP请求cookie和Java源代码，从而熟悉WebGoat的特性。



2.启动 Burpsuite ，设置其“proxy - intercept is on”，使其处在监听状态。具体方法参考《Burpsuite 安装使用》。

3.再次操作浏览器，点击“go！”。此时，访问过程将被burp拦截，在burp suite的proxy中可以看到如下信息。



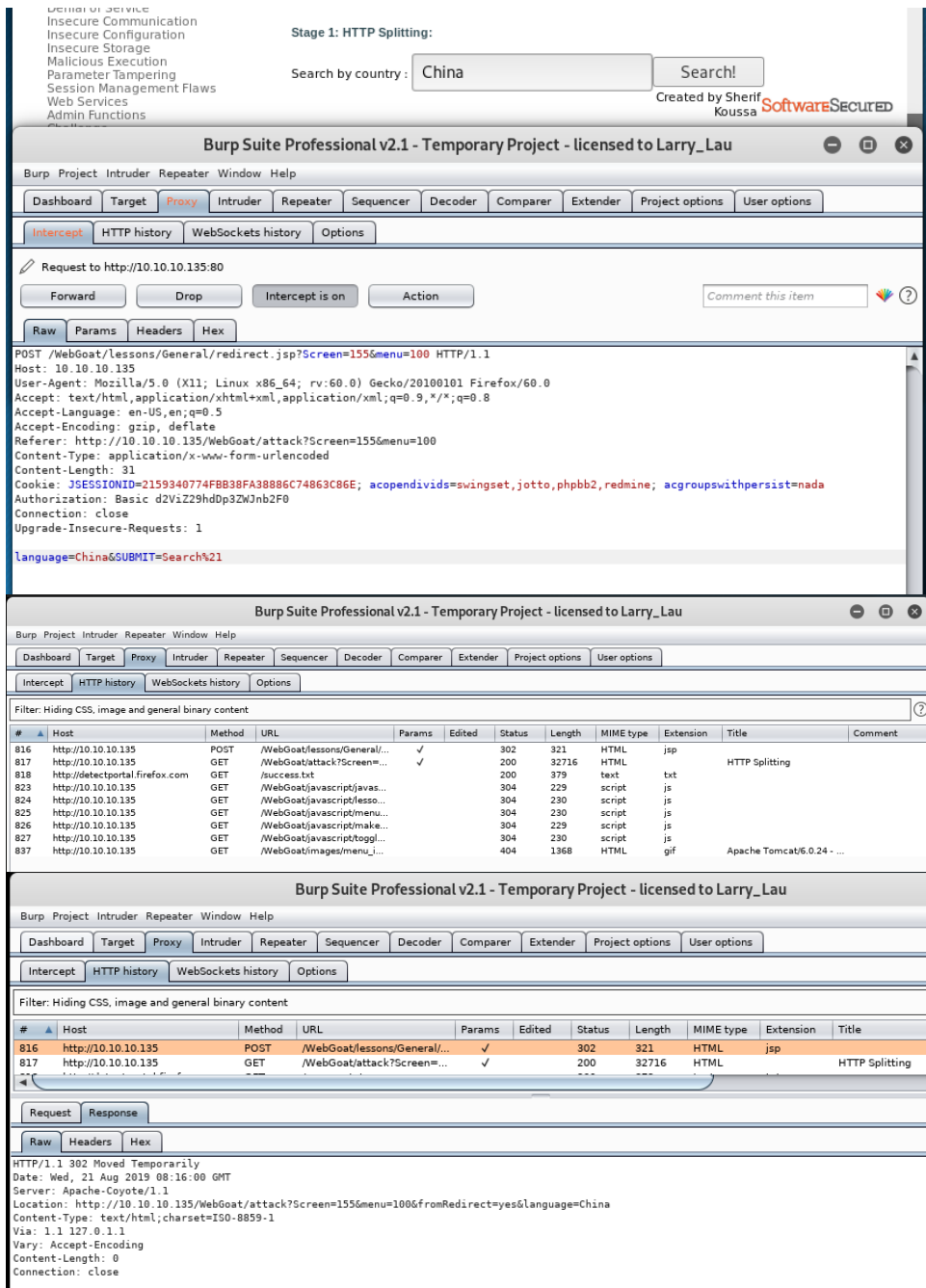
4.可以看到信息中包含了我们提交的字符串，以及请求原文。点击Proxy中的Params、Headers、Hex等页面，分析其中信息，然后点击“Forward”查看变化。

5.所有历史请求和响应，可以通过burp suite proxy中的“http history”查看。

6.点击webgoat中general第二部分http splitting（http 分片）。在输入框中输入China，启动burpsuite - proxy - interrupt is on 之后再点击"Search"。

说明：这一课有两个阶段。阶段1，告诉你如何做HTTP拆分攻击；阶段2 基于你所学，使用HTTP拆分执行Cache投毒。在上面的搜索框中键入一个搜索字符，然后点“search！”。你会注意到这个应用重定向了你的请求到另一个服务资源上。你需要使用CR（%0d）和LF（%0a）字符来实现这个攻击。你的攻击目标是强制使服务器发出“200 OK”这一http response。如果屏幕因你的攻击而改变

了输出结果，那么就会回到主页。在第二阶段渗透成功后，你会在左侧菜单中发现绿色的check。你可以查看PHP字符编码，使用Encode和Decode URI Component按钮来翻译CR和LF。



可以看到在历史记录中，有302 Moved Temporarily，表示服务器告诉浏览器，URL临时改变了，应该采用Location返回的重定向地址，重新发送一次请求。这里有

Location:<http://10.10.10.135/WebGoat/attack?Screen=231&menu=100&fromRedirect=yes&language=china>

重定向地址里面的Language参数，恰好就是通过浏览器里提交的参数“china”，这就存在着HTTP分拆漏洞的可能性。

7.在输入框中输入如下内容，实现http splitting攻击（%0d%0a即回车换行）。

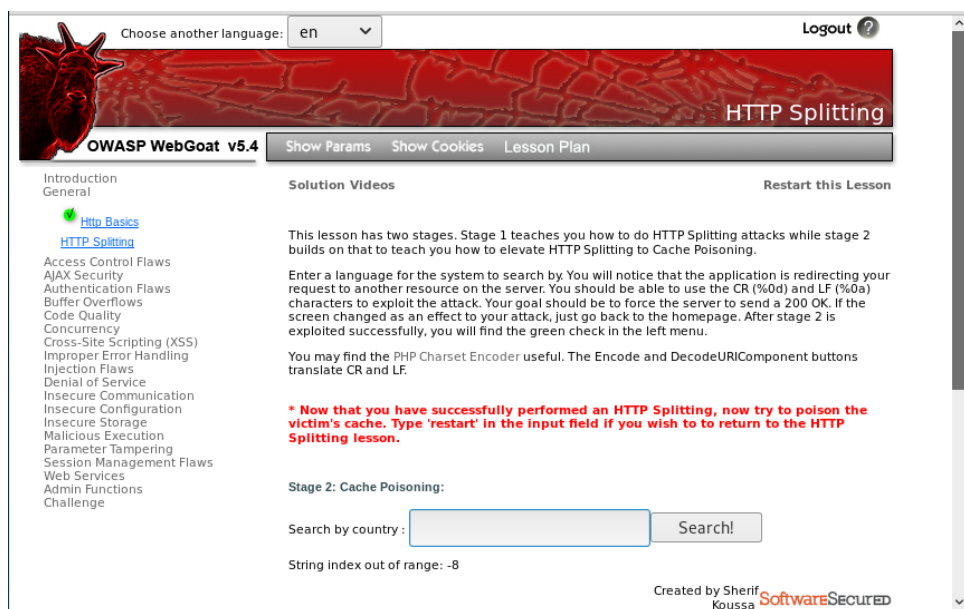
china%0aContent-Length:%200%0a%0aHTTP/1.1%20200%200K%0aContent-Type:%20text/html%0aContent-Length:%2047%0a<html>

如果对url编码不熟悉，可以使用在线转换工具：<http://tool.oschina.net/encode?type=4>

上述注入代码内容即一个http请求头：

```
language:china
Content-Length: 0
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 47
<HTML>stage 1 success</html>
```

提交后，会显示页面“stage 1 success”，然后回到webgoat-general会有以下界面。



上面的载荷还可以更为复杂一些，例如：

china%0aContent-Length:%200%0a%0aHTTP/1.1%20200%200K%0aContent-Type:%20text/html%0aContent-Length:%2047%0a<html>

8.接下来，尝试stage 2——Cache Poisoning。

主要是设置Last-Modified参数为未来时间，比如2020年1月1日。意思是服务器告诉浏览器，这个网页最后一次修改是在2020年1月1日。

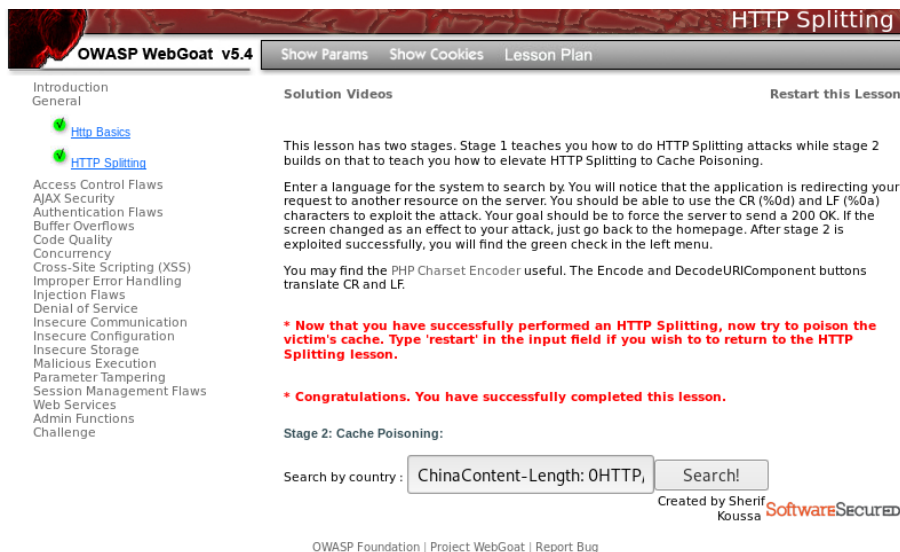
如果浏览器不清除缓存，下一次请求这个页面会加上if-modify-since字段，值是2020年1月1日，也就是说浏览器会问服务器，这个网页在2020年1月1日之后修改过嘛？

服务器可能会对这条信息莫名其妙，怎么会传过来一个未来的时间值？但是因为很多web程序都不进行审核，一般http服务器会对比最后一次修改时间和传过来的时间，发现传过来的时间较新，就会回送304即Not Modified，表示这个网页没有被修改过。浏览器接受到了这个响应会从本地缓存读取网页，当然

是之前注入的错误网页。如果浏览器不清除缓存，再也别想读取到正确的网页信息了（所以为了加强终端的web安全，要设定较短的缓存清理周期）。这就是HTTP的Cache Poisoning(缓存毒化)。

我们准备的Stage2 攻击字串为：

China%0aContent-Length:%200%0aHTTP/1.1%20200%20OK%0aContent-Type:%20text/html%0aContent-Length:%2047%0aLast-Modified:%202000%0a



5 实验结论

实验第一部分，演示了注入非法字符，获得意外结果的情况。这个情况可用于欺骗用户输入敏感信息。

这个实验的第二部分演示了使用Last-Modified这一http请求头参数欺骗服务器。Http cache毒化的常见于中间人攻击，即某个代理劫持了服务器内容，并修改其中信息，然后传递给用户。用户不知情的情况下，可能会泄露个人信息或提交受控的资源。