# DVWA之暴力破解

## 漏洞概述

暴力破解指攻击者枚举其准备的用户名和密码字典同时进行登陆，通过响应结果从而得到正确用户名和密码的过程

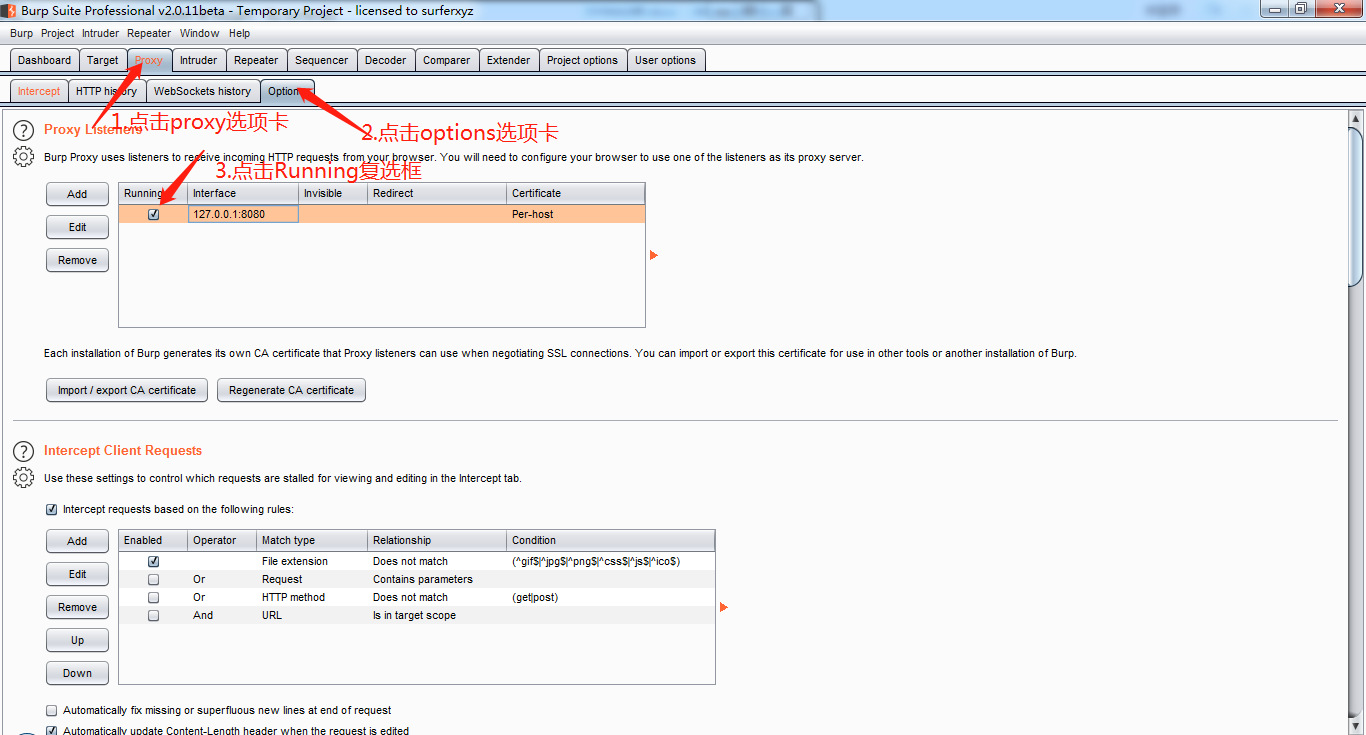
## 测试攻击

firefox浏览器，burpsuite

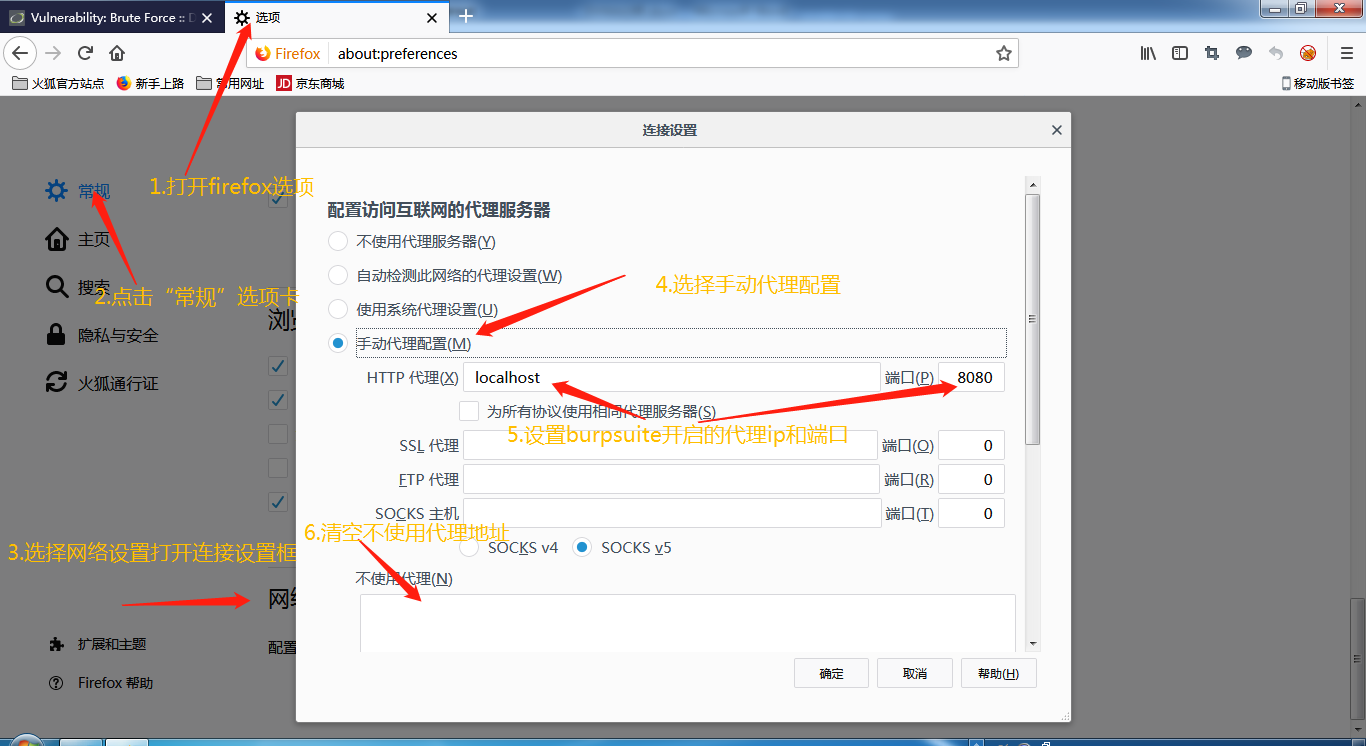
## 测试方法

### LOW级别

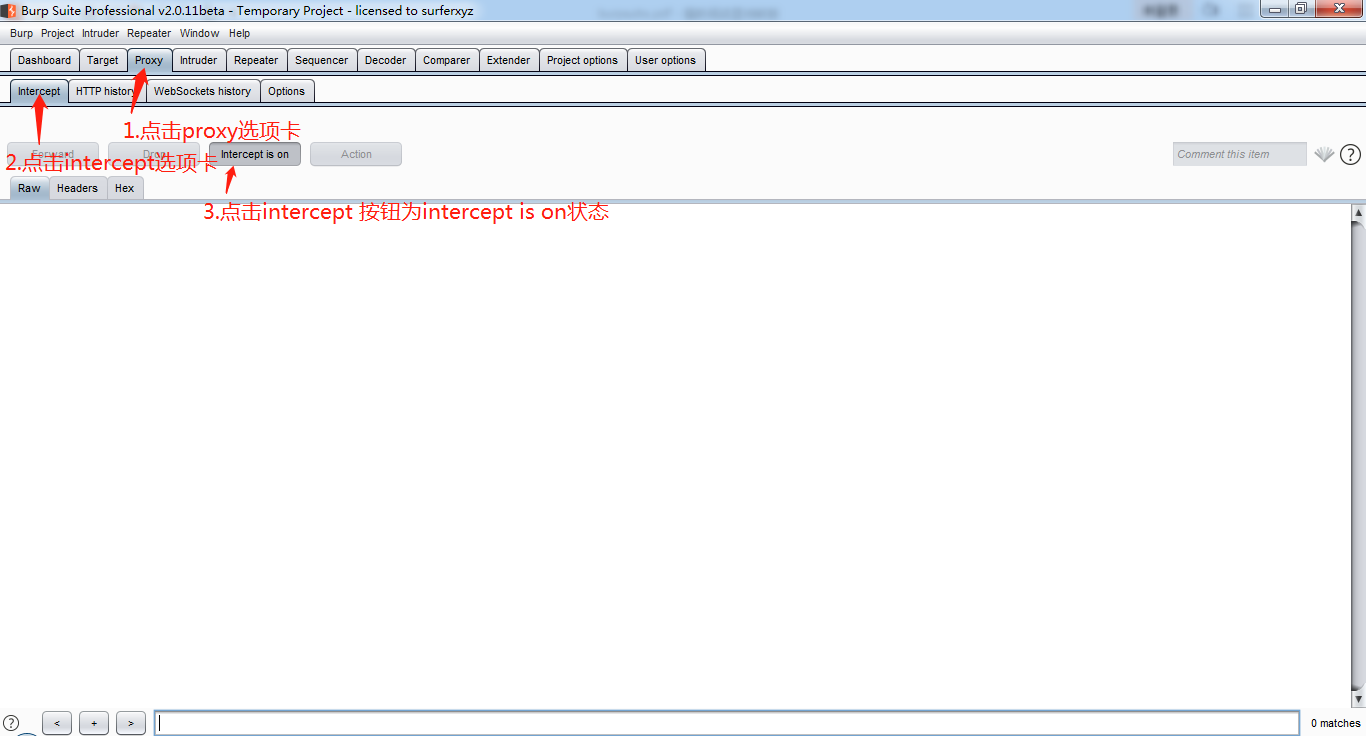
1. 设置DVWA安全级别为LOW
2. 启动burpsuite并开启代理



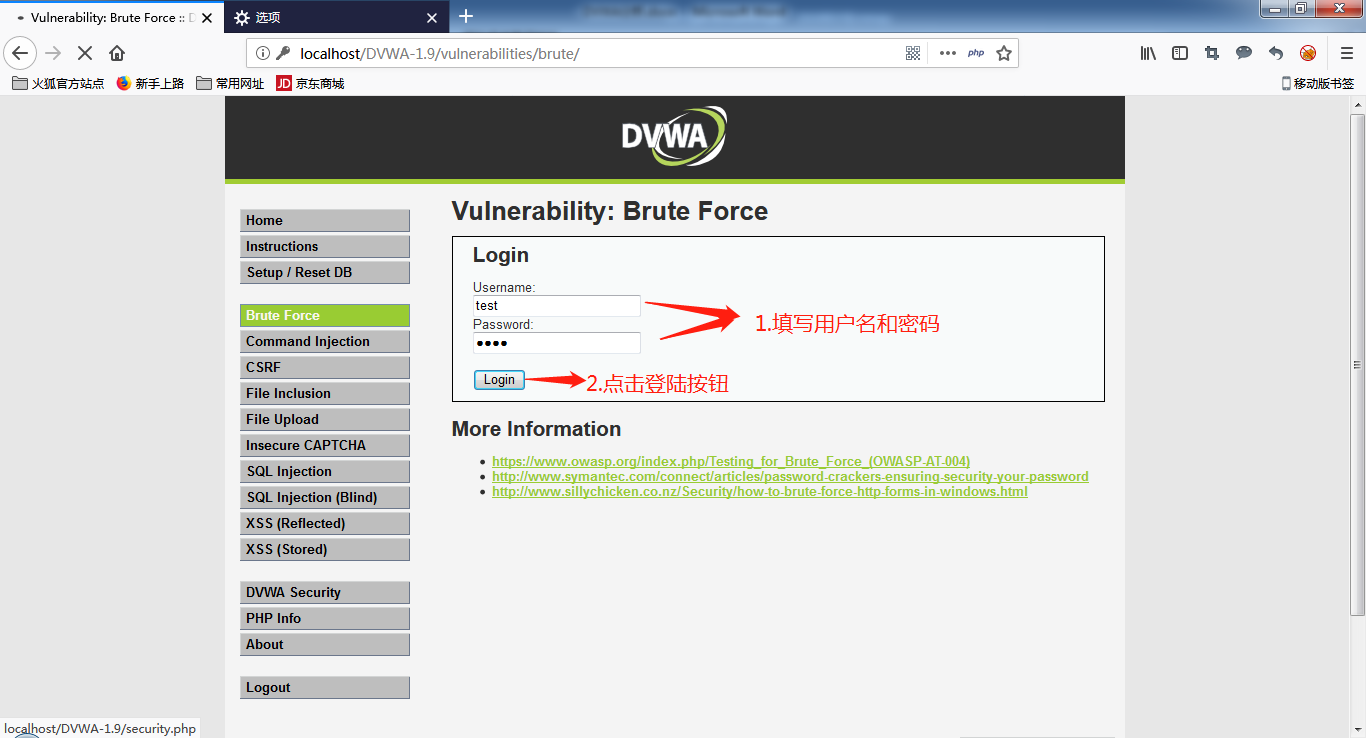
1. 设置firefox浏览器代理为127.0.0.1:8080



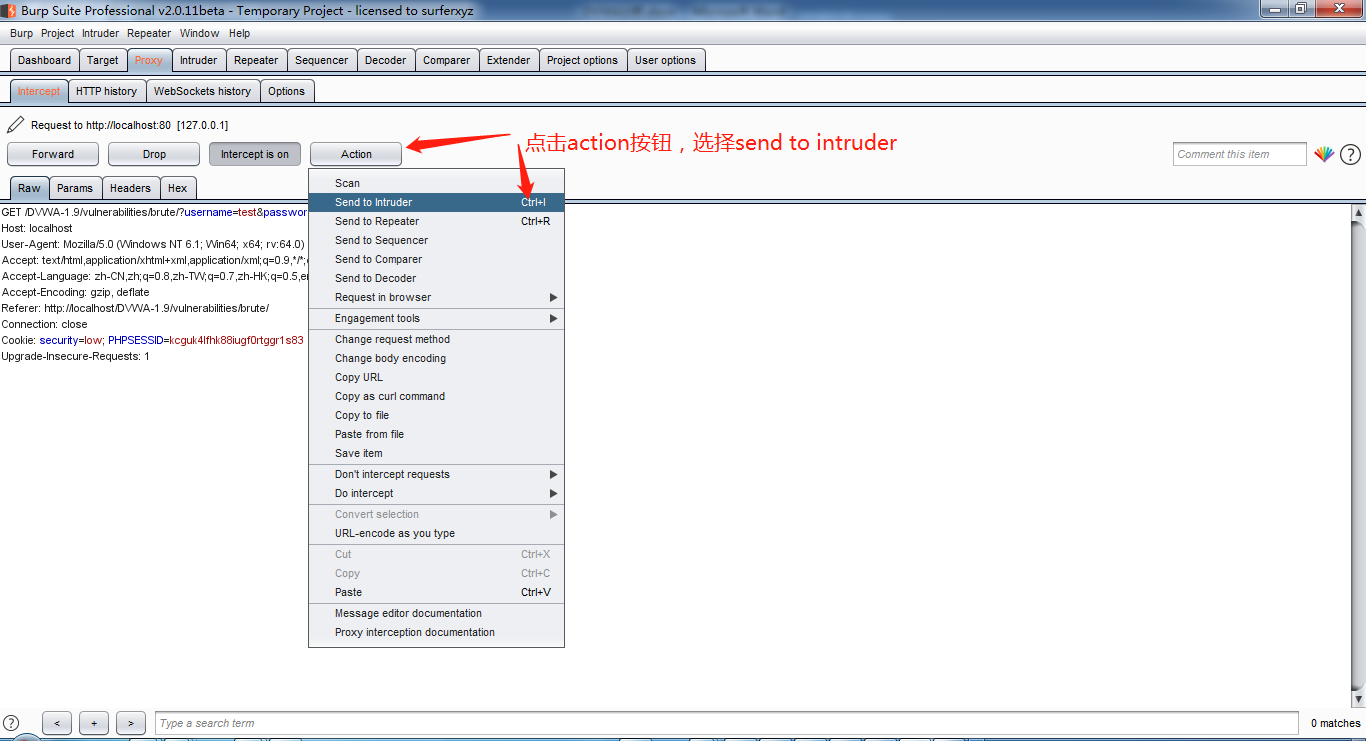
1. 开启burpsuite拦截



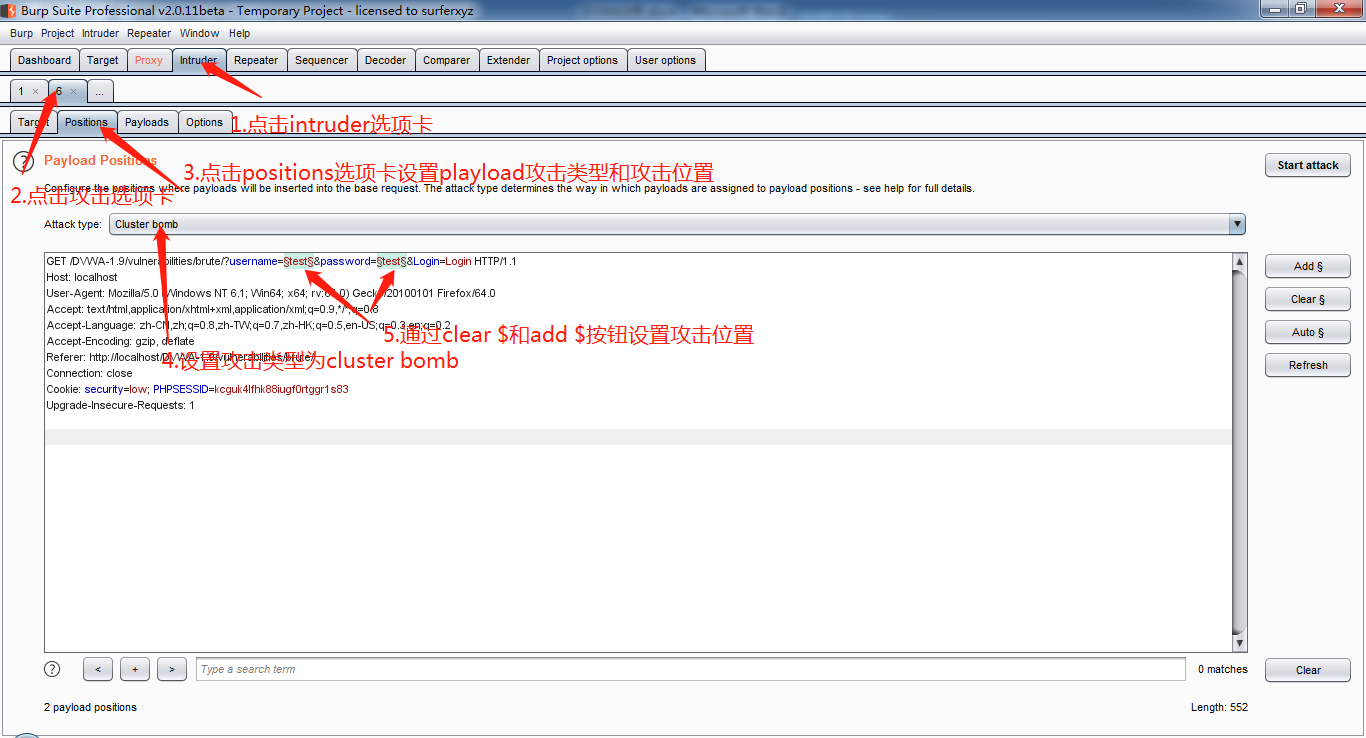
1. 使用firefox浏览器发起登陆请求



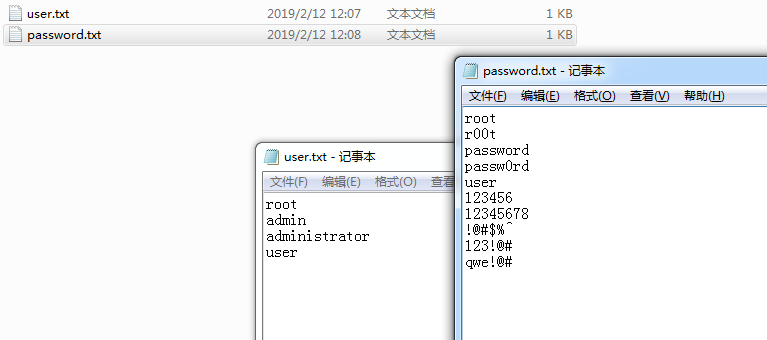
1. 使用burpsuite进行暴力破解



1. 设置攻击方式和攻击位置

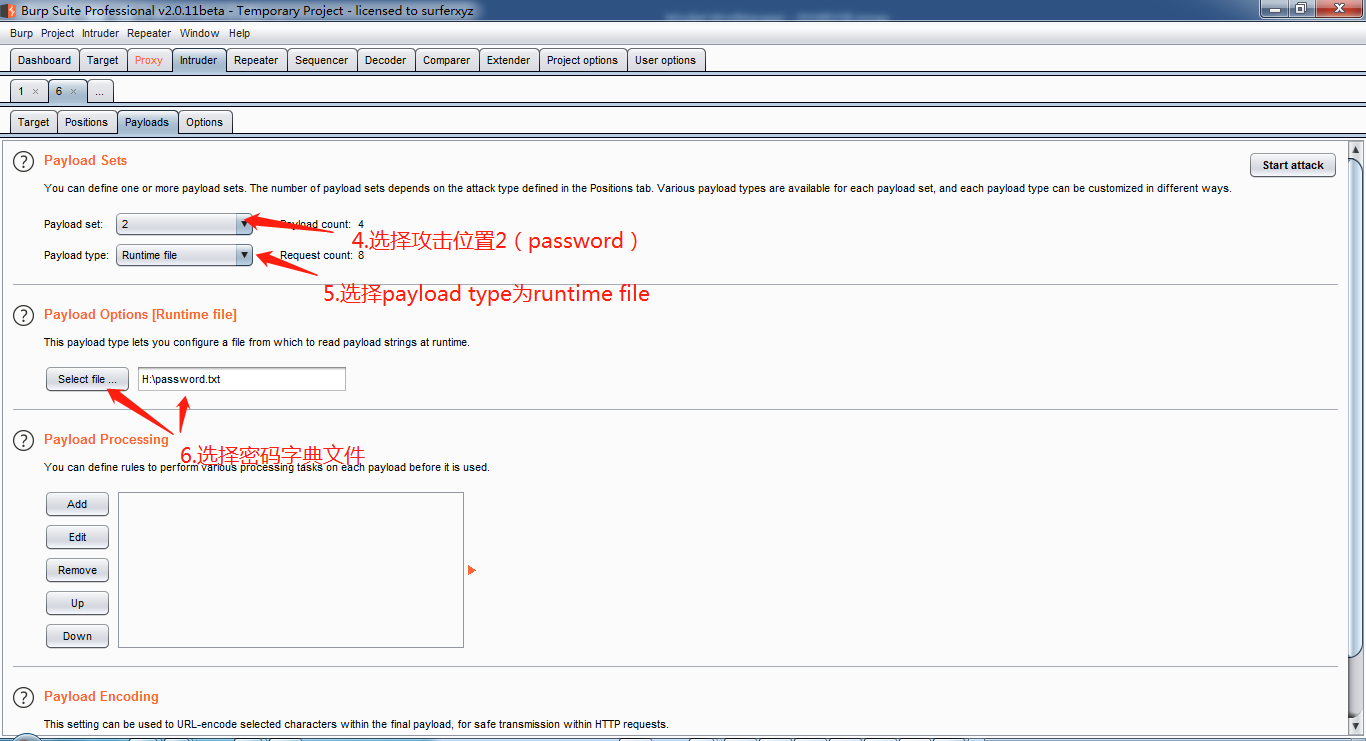


1. 准备用户名和密码字典

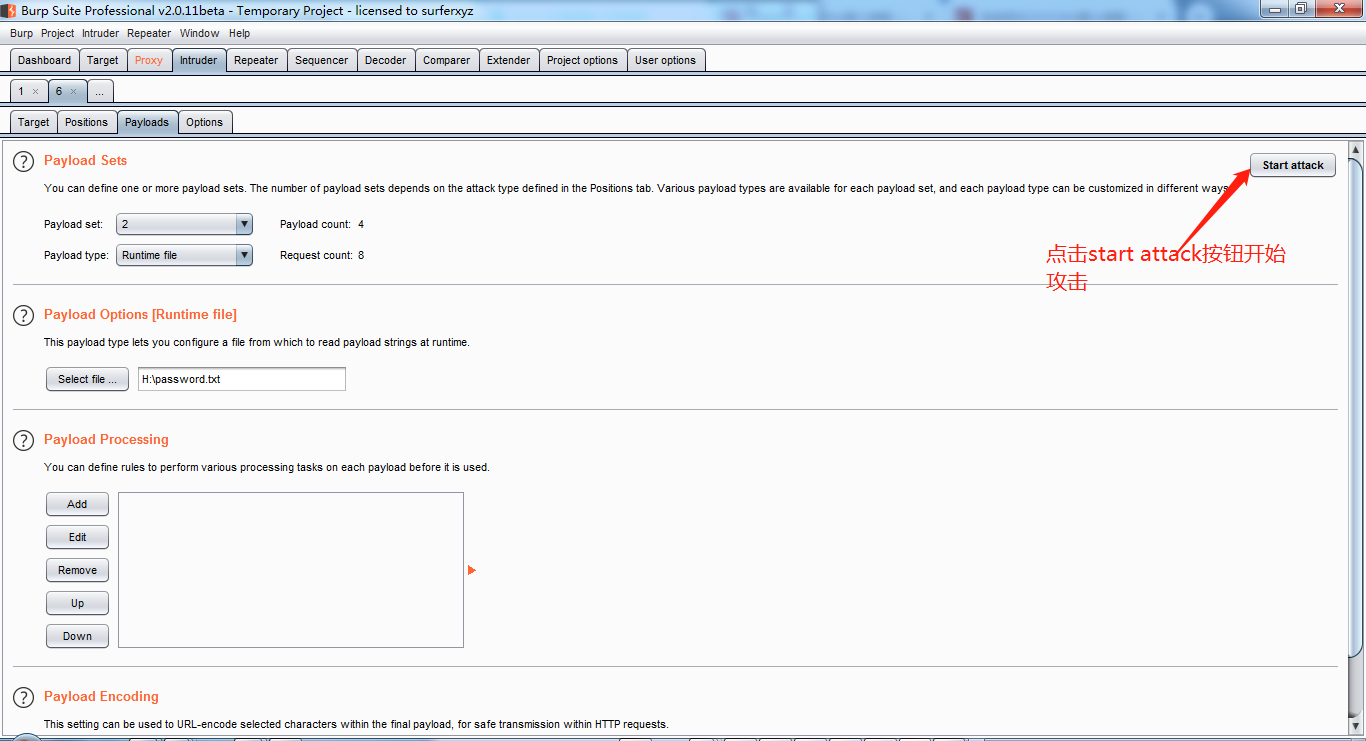


1. 设置playloads

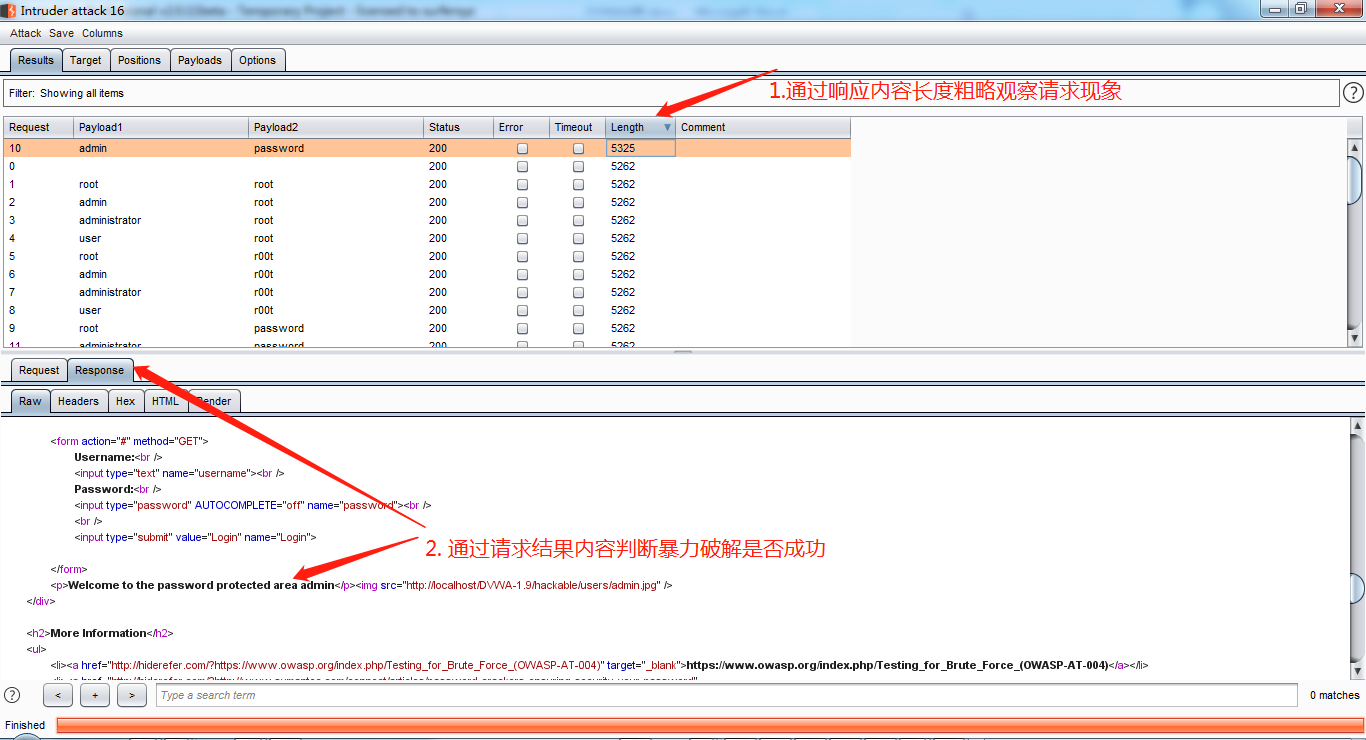




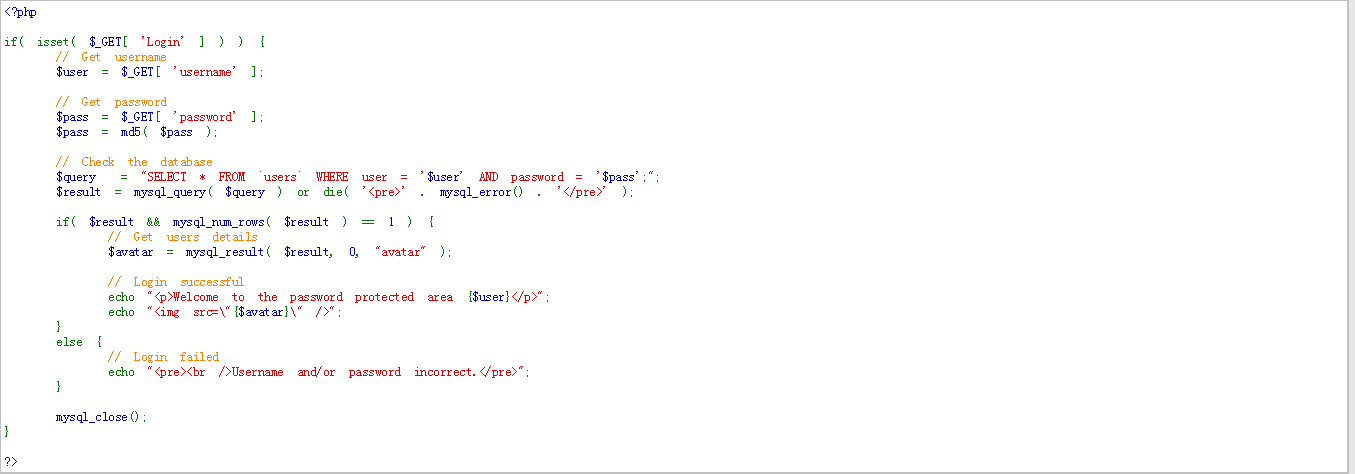
1. 开始攻击



1. 分析结果



1. 代码分析



说明：

登陆过程直接将输入数据拼写到SQL字符串中进行执行，无任何保护措施，可任意尝试登陆以及使用’闭合的SQL注入(admin’或admin’ or ‘1’=’1)

1. 自编Python脚本暴力破解



使用：

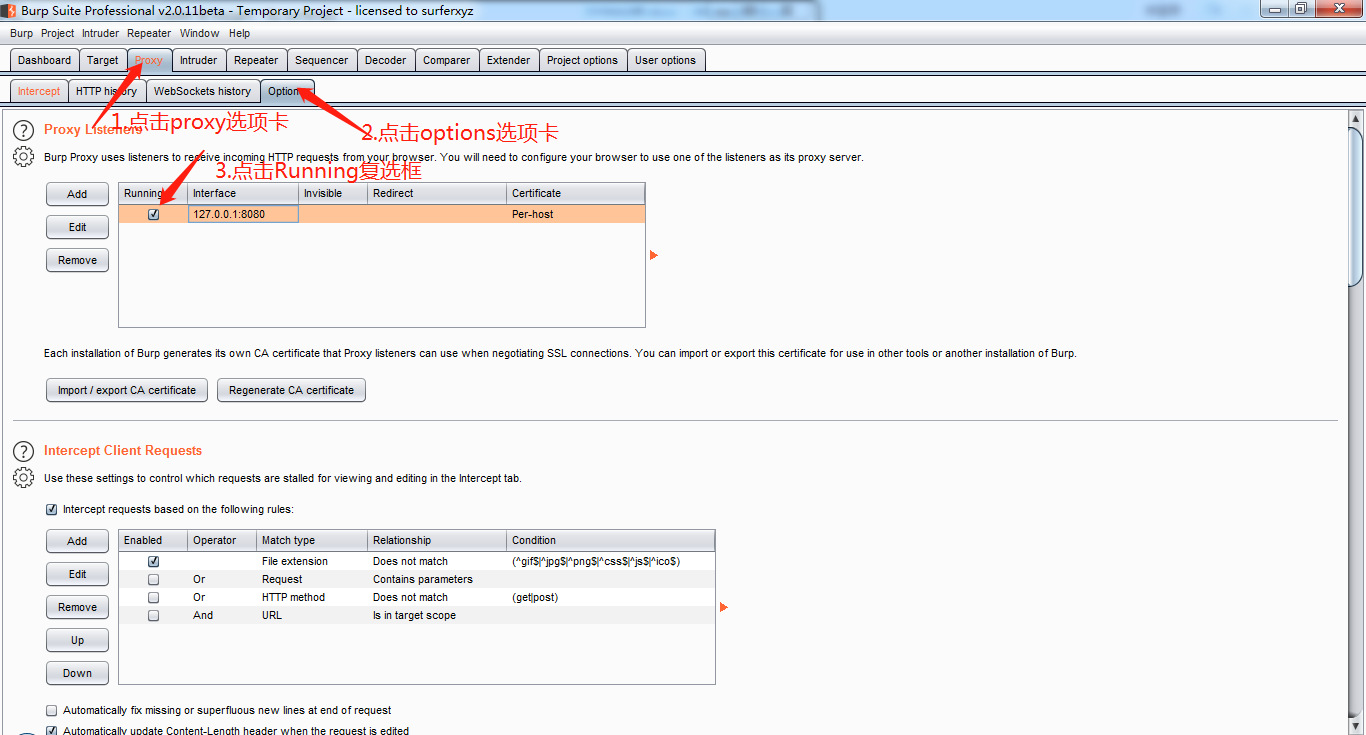
* 从浏览器中copy登陆成功后的cookie信息
* 使用python3运行脚本

说明：

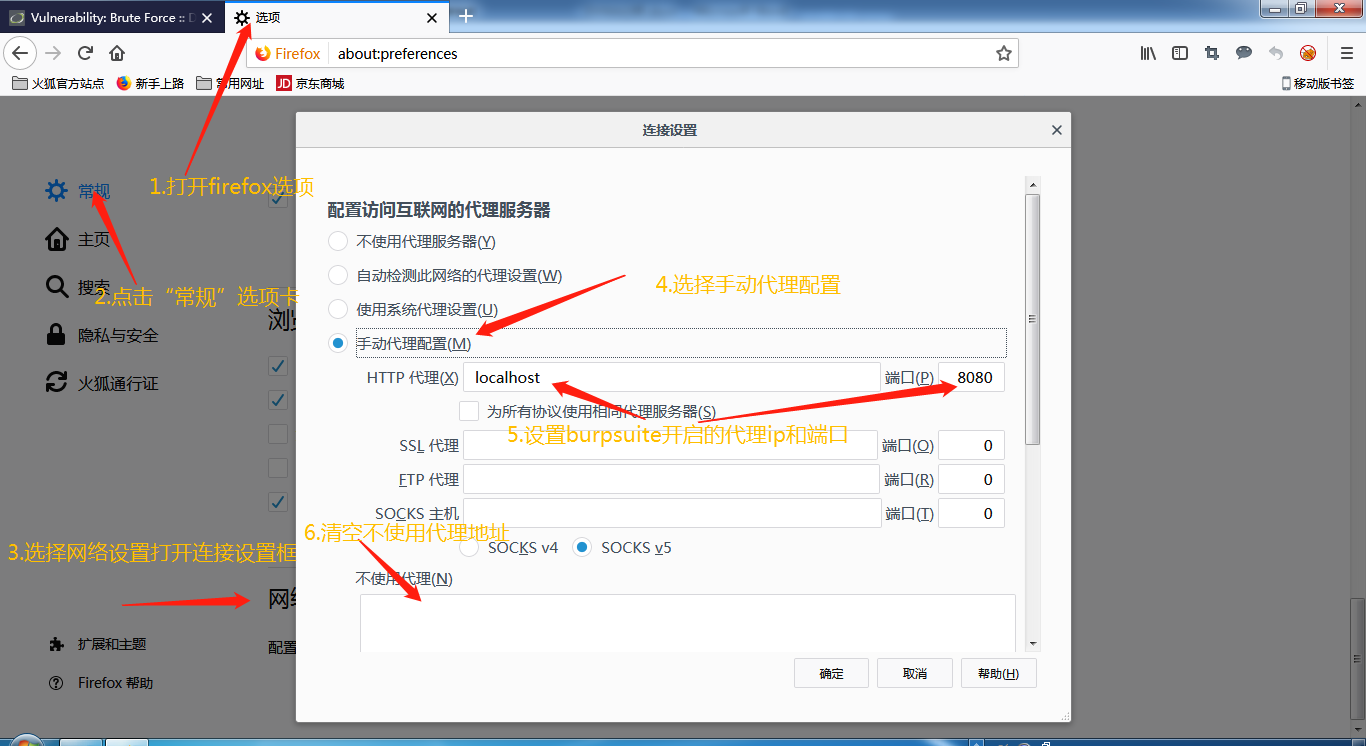
* 遍历用户名和密码字典文件
* 使用requests发送请求到dvwa，根据请求结果是否包含Welcome to password protected area字符串判断是否破解成功

### MEDIUM级别

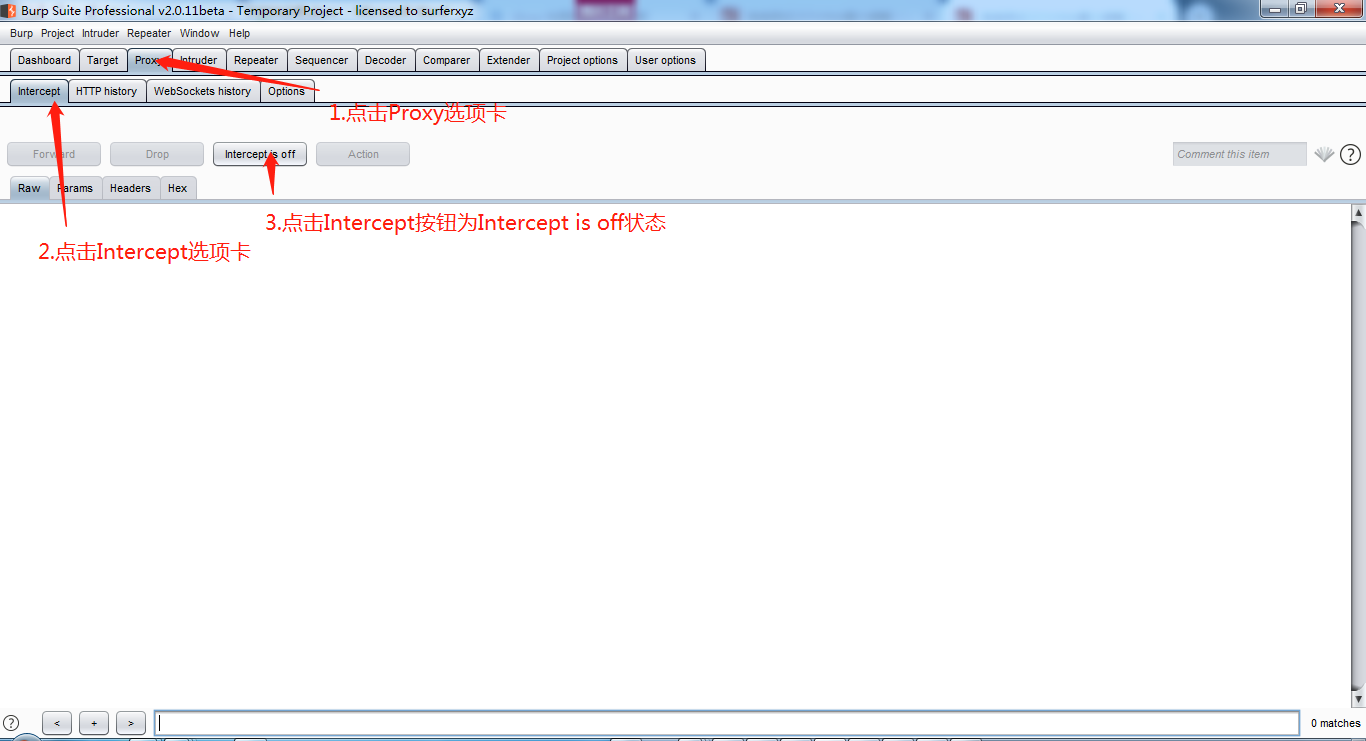
1. 设置DVWA安全级别为Medium
2. 启动burpsuite并开启代理



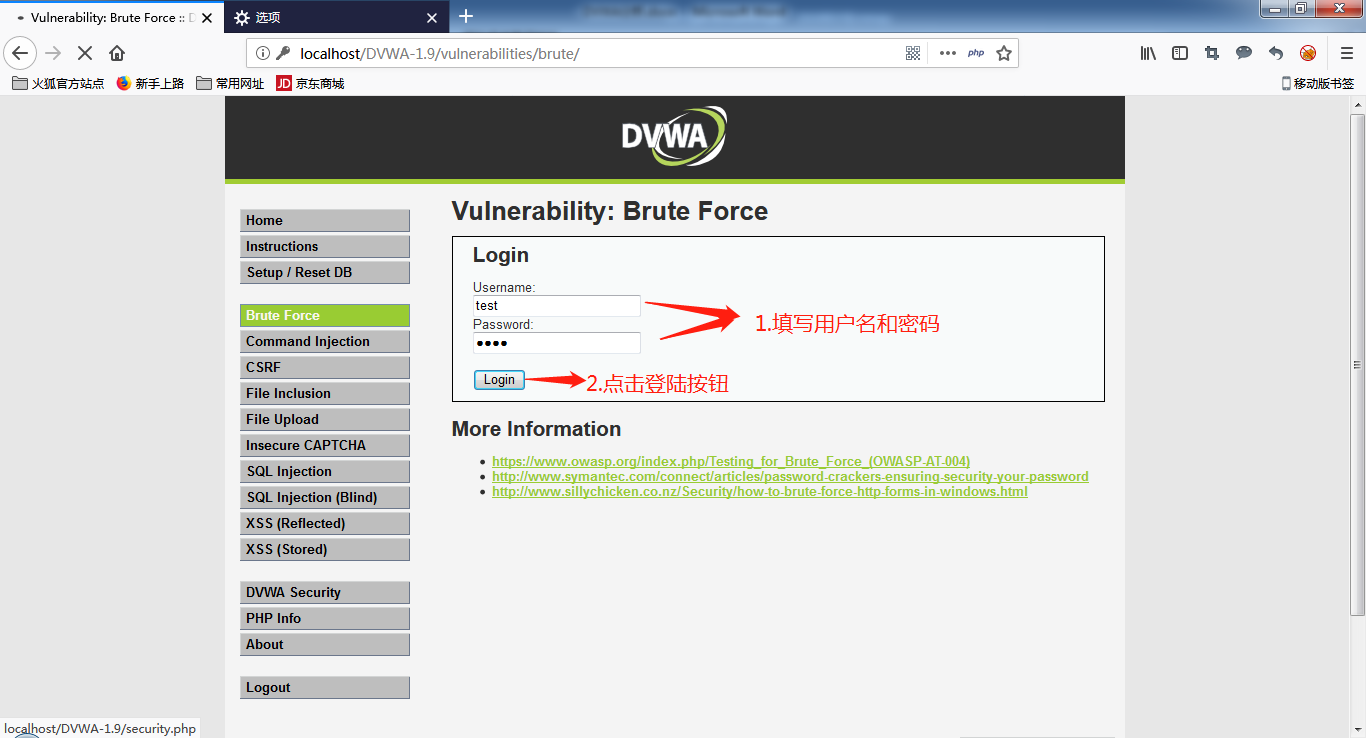
1. 设置firefox浏览器代理为127.0.0.1:8080



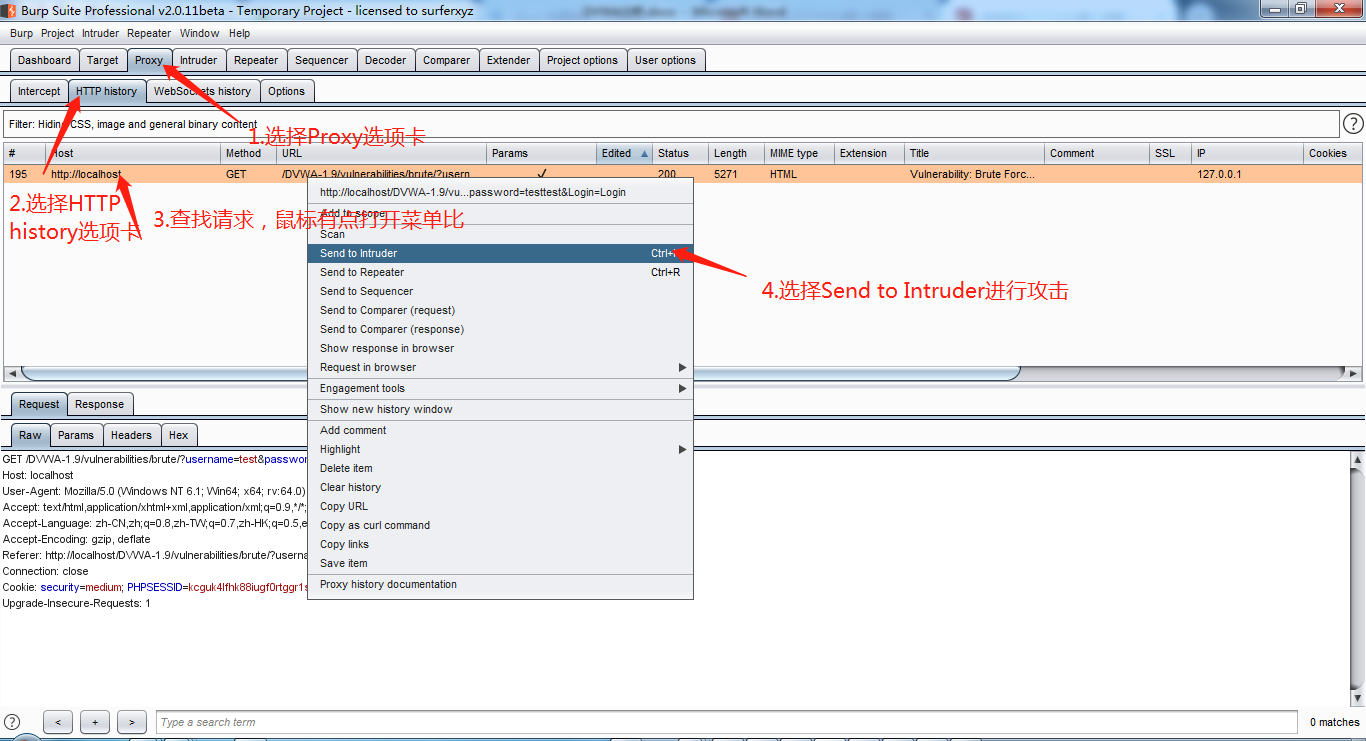
1. 关闭burpsuite拦截



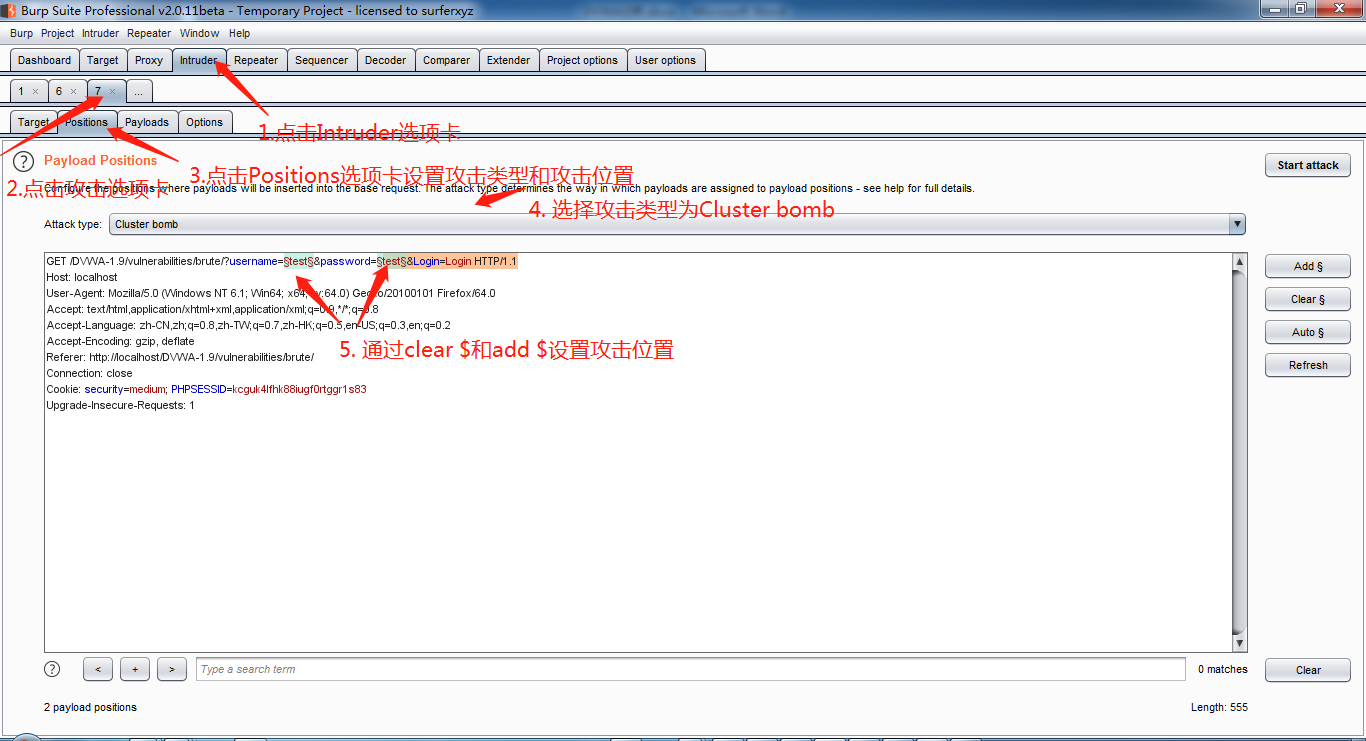
1. 使用firefox浏览器发起登陆请求



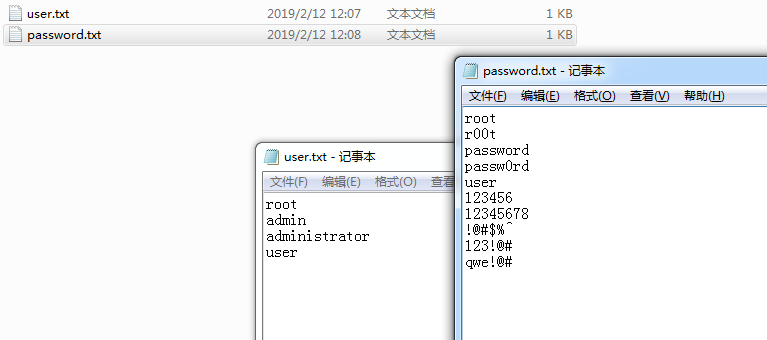
1. 使用burpsuite进行暴力破解



1. 设置攻击方式和攻击位置



1. 准备用户名和密码字典

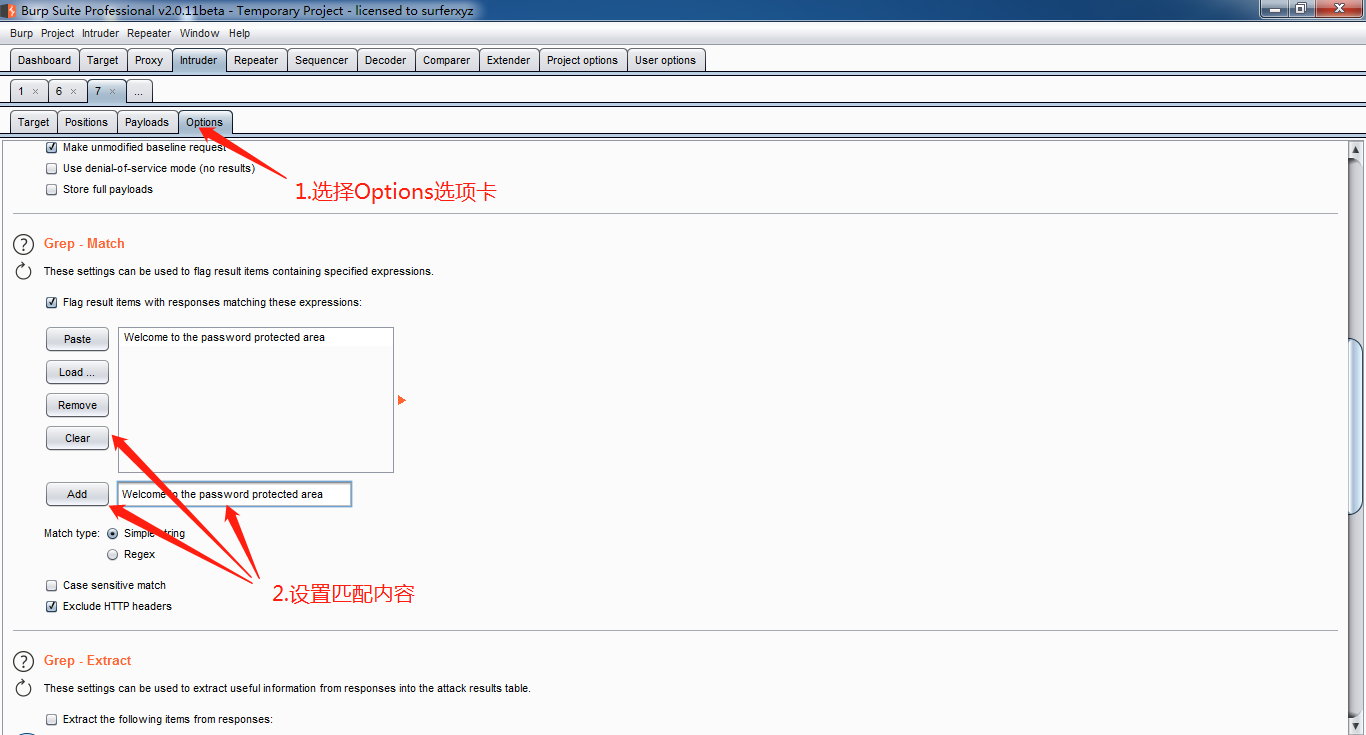


1. 设置playloads

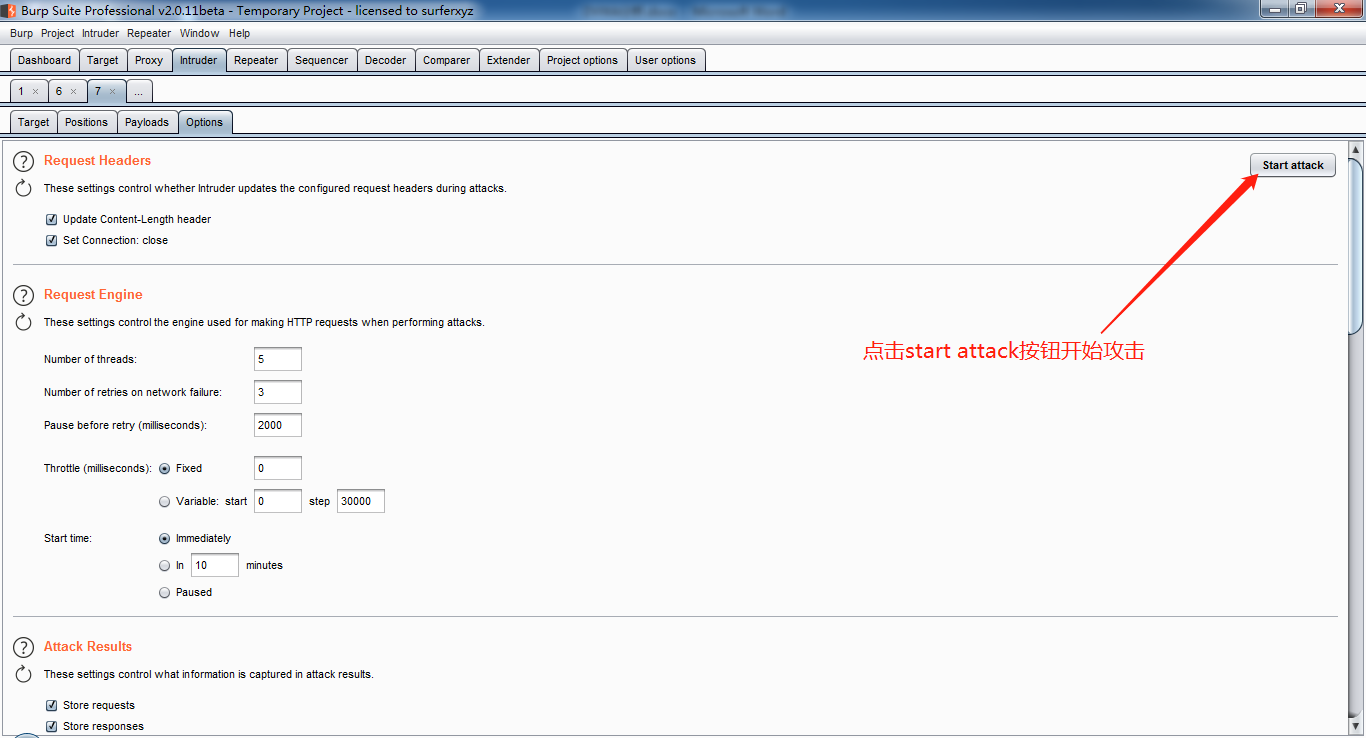




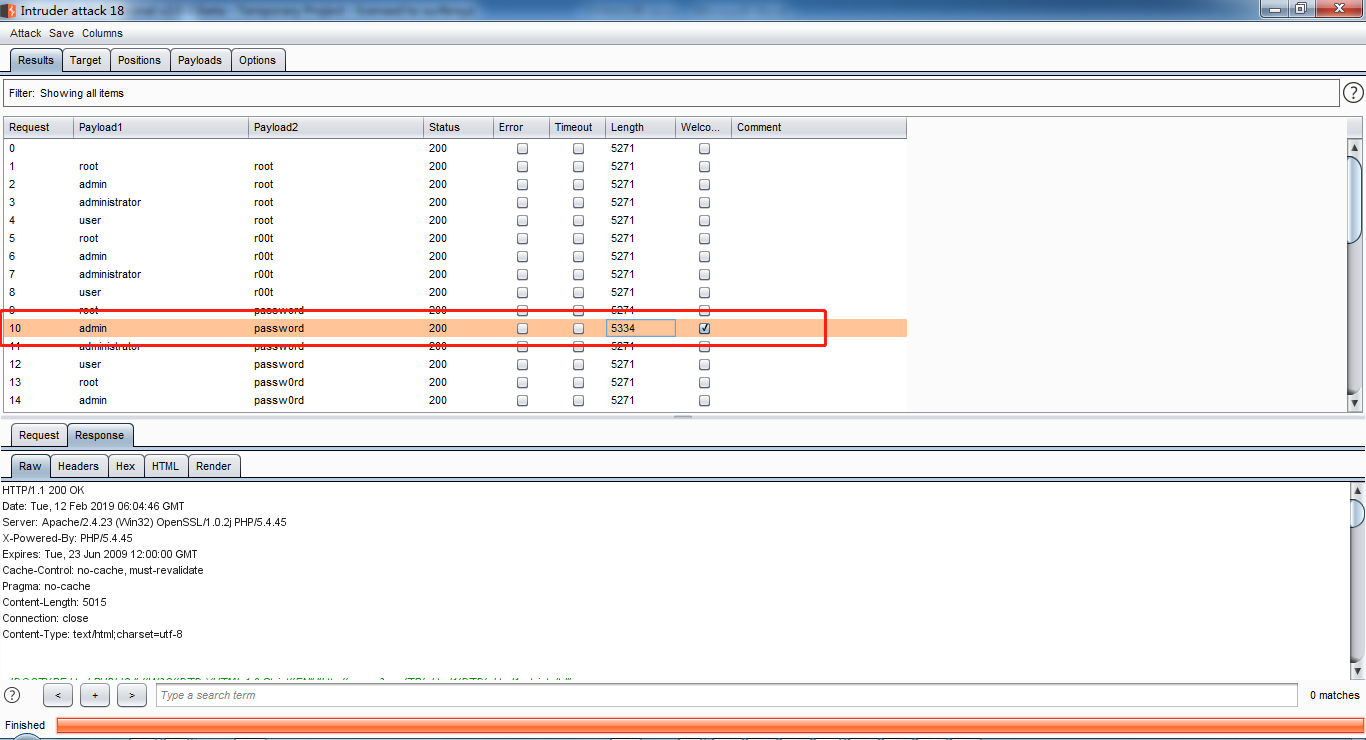
1. 设置结果匹配信息用以判断是否登陆成功



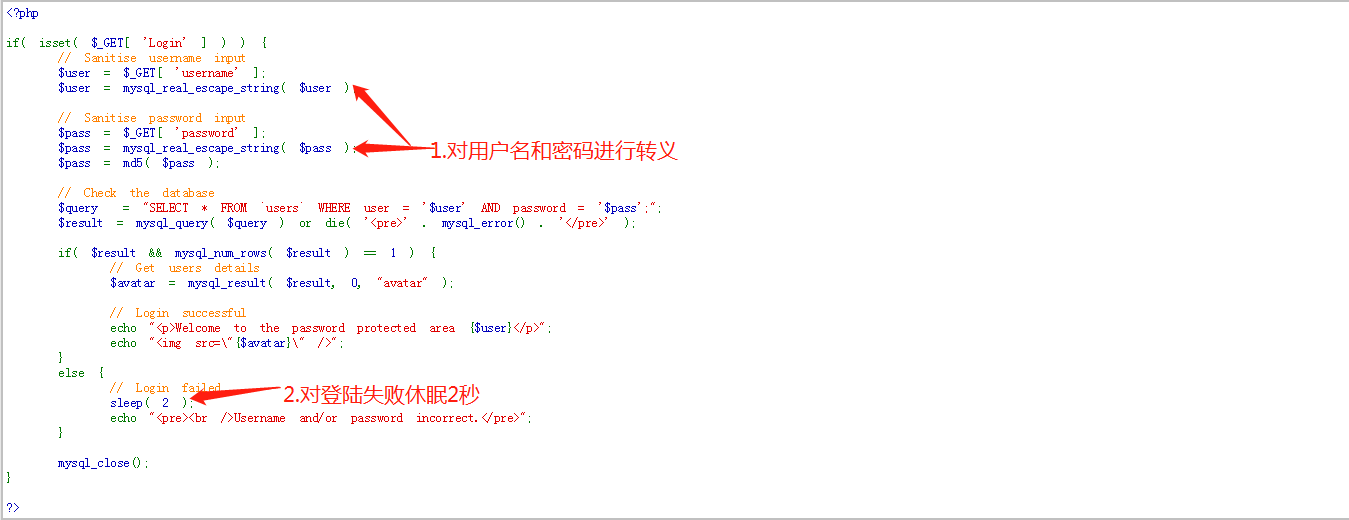
1. 开始攻击



1. 分析结果



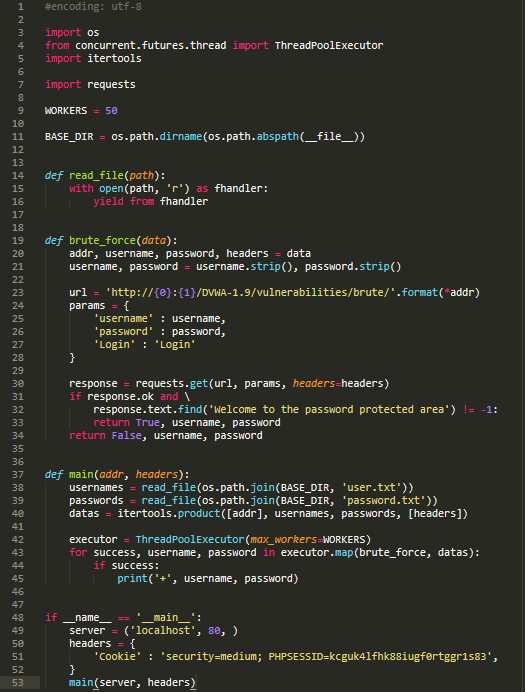
1. 代码分析



说明：

针对用户名和密码进行转义处理，预防SQL注入，同时对登陆过程失败的情况，休眠2秒返回结果，在一定程度上使破解攻击时间延长

1. 自编Python脚本暴力破解



使用：

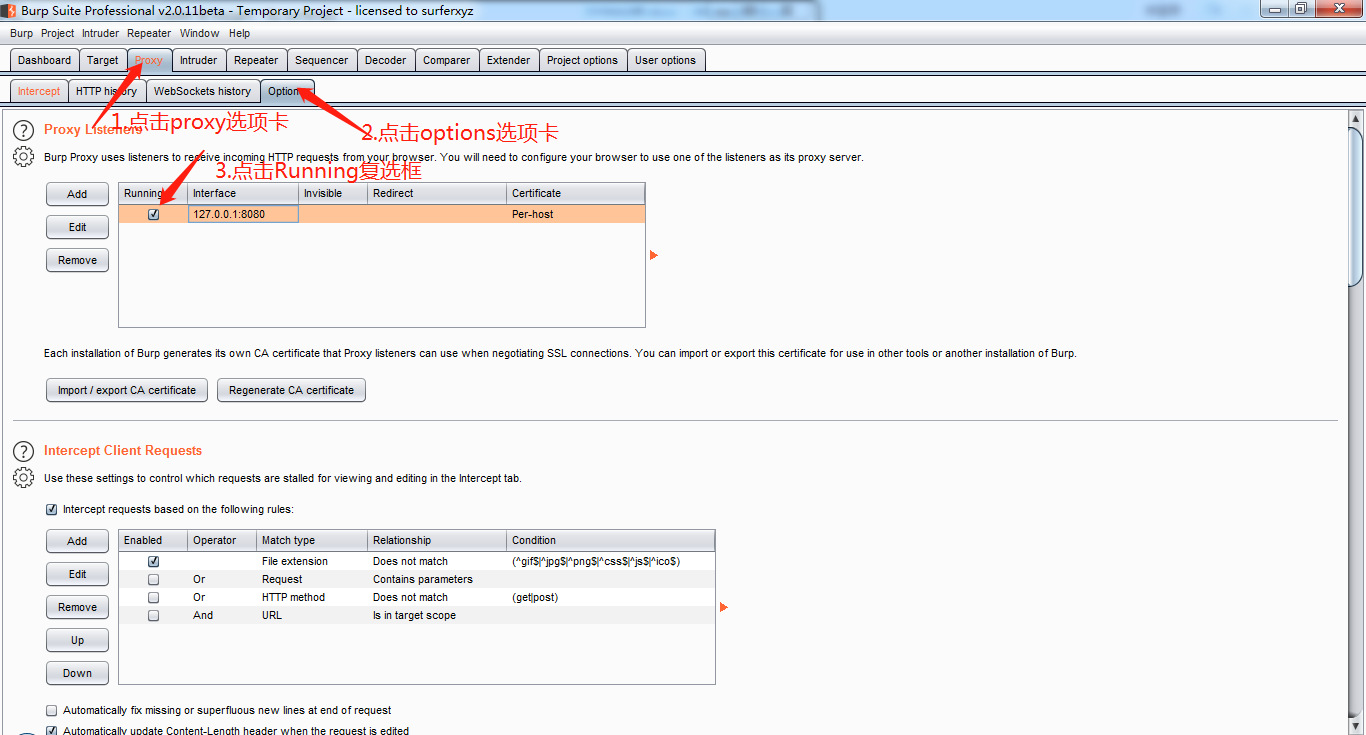
* 从浏览器中copy登陆成功后的cookie信息
* 使用python3运行脚本

说明：

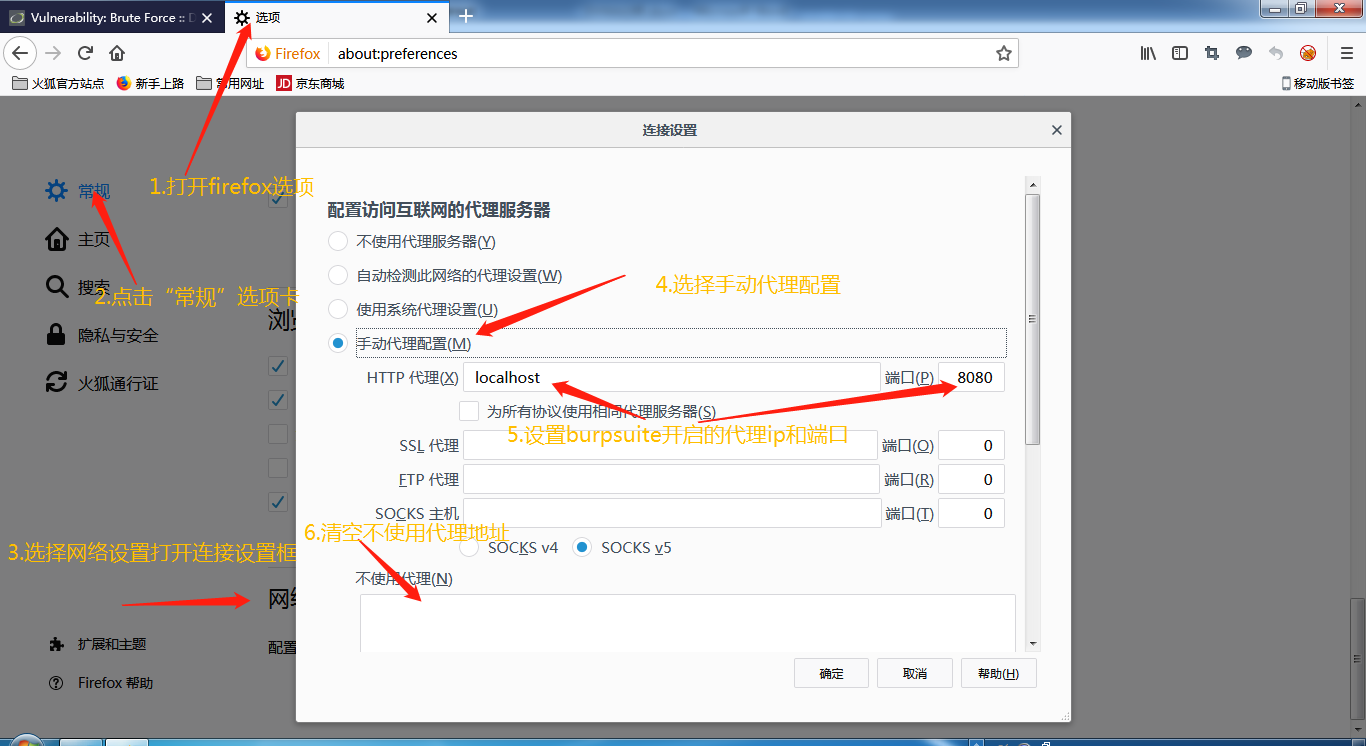
* 使用线程池机制同时发起多个请求
* 使用requests发送请求到dvwa，根据请求结果是否包含Welcome to password protected area字符串判断是否破解成功

### HIGH级别

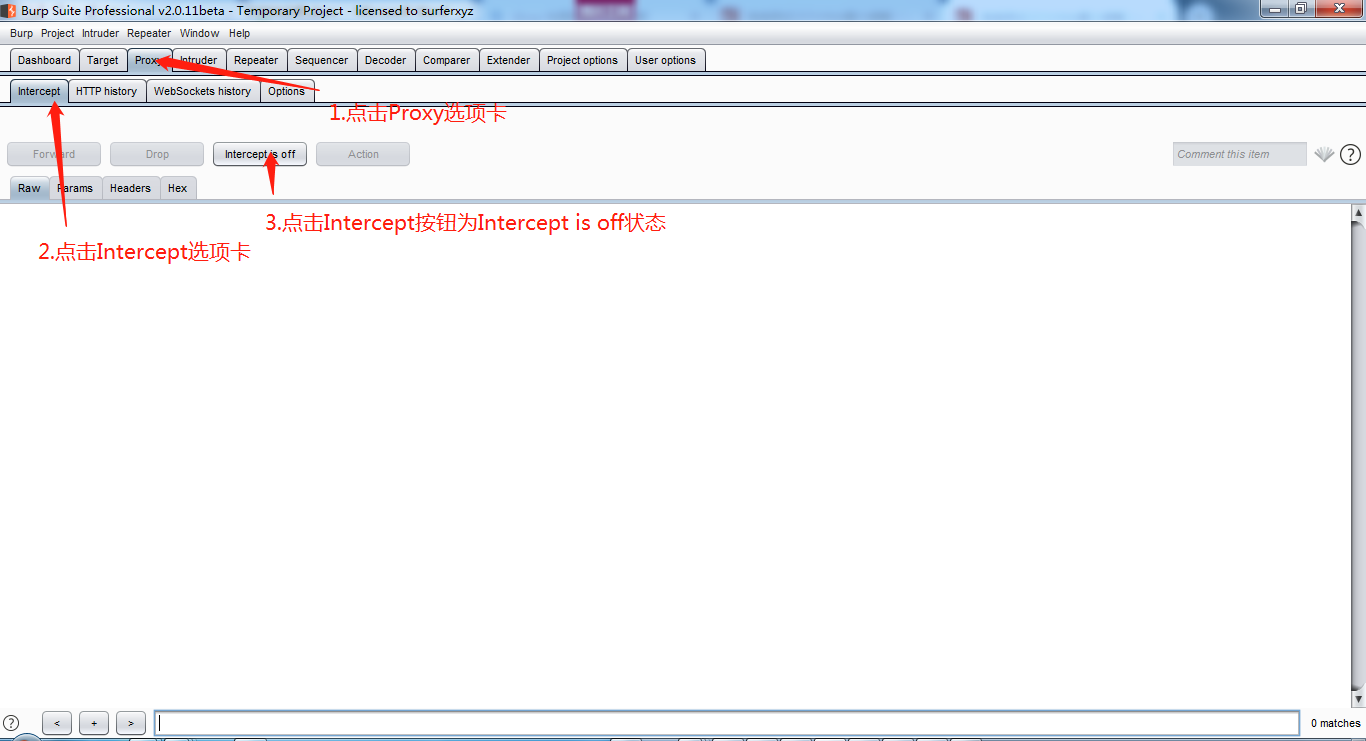
1. 设置DVWA安全级别为High
2. 启动burpsuite并开启代理



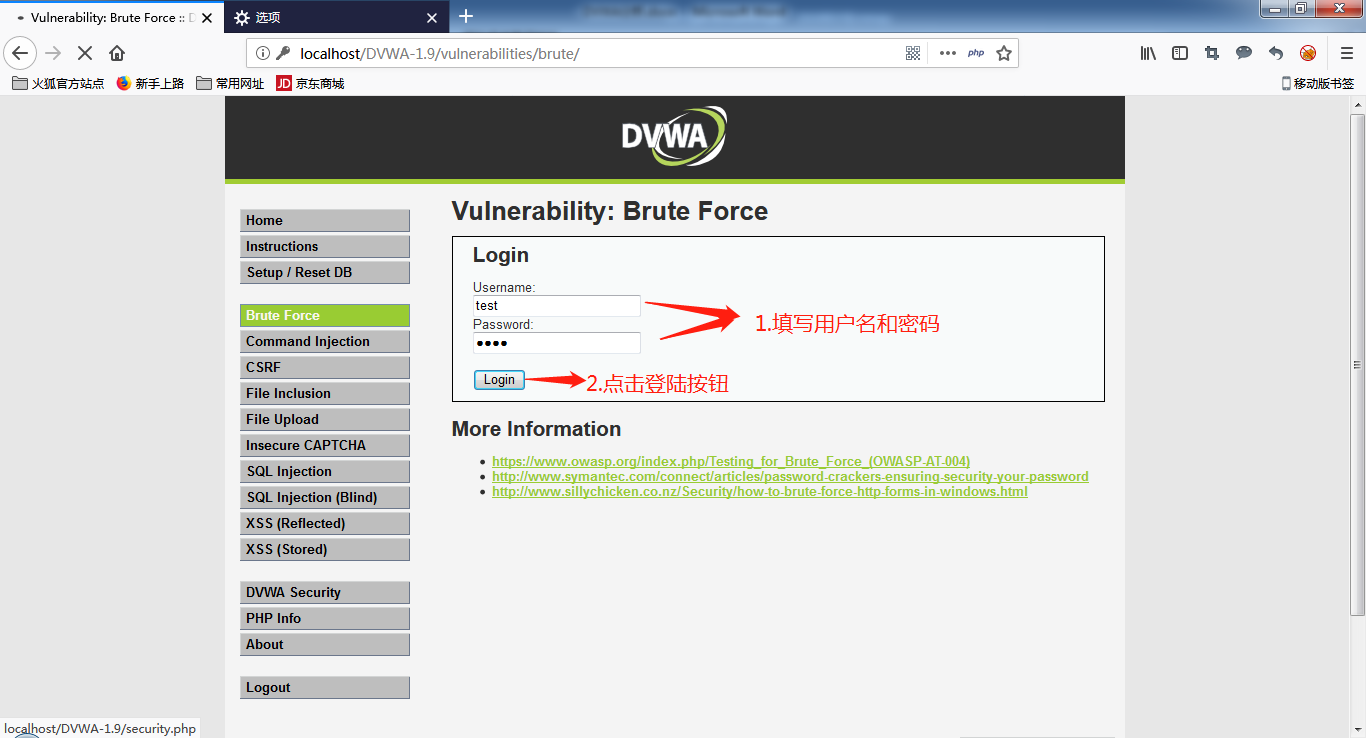
1. 设置firefox浏览器代理为127.0.0.1:8080



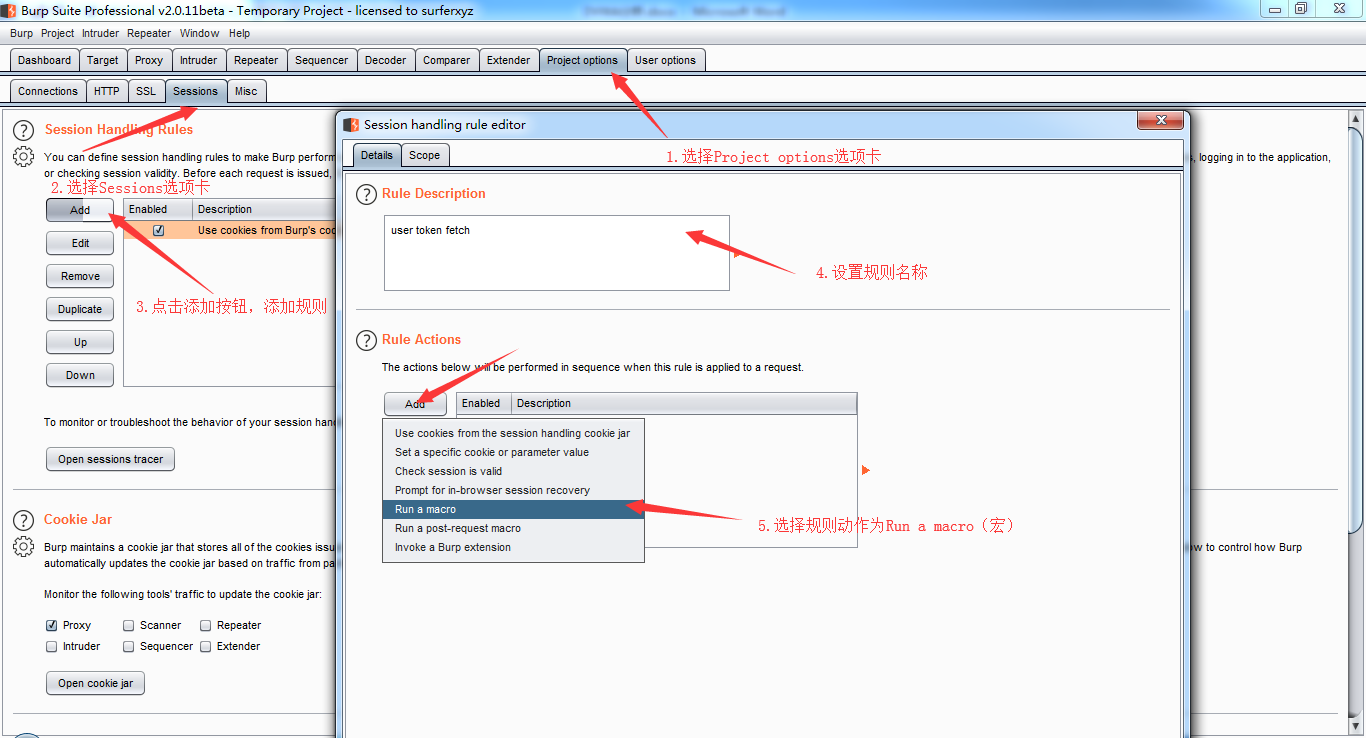
1. 关闭burpsuite拦截

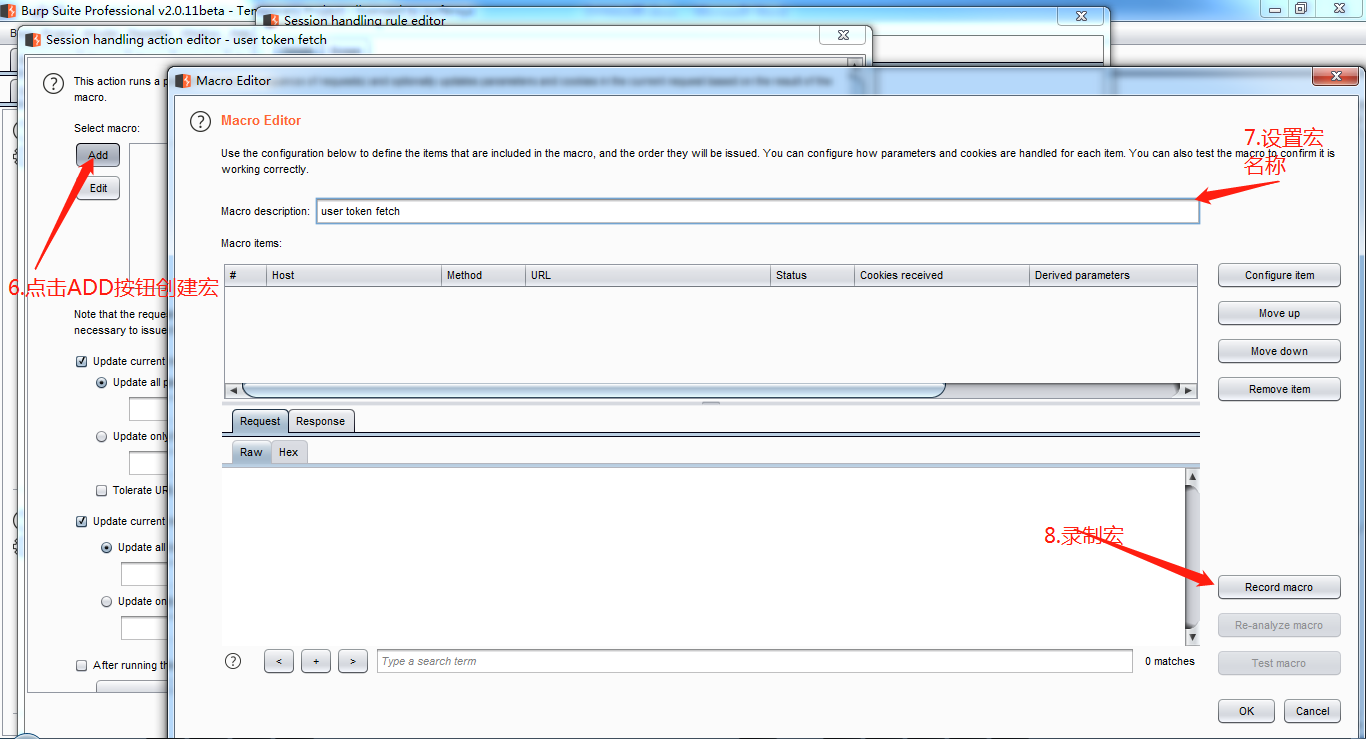


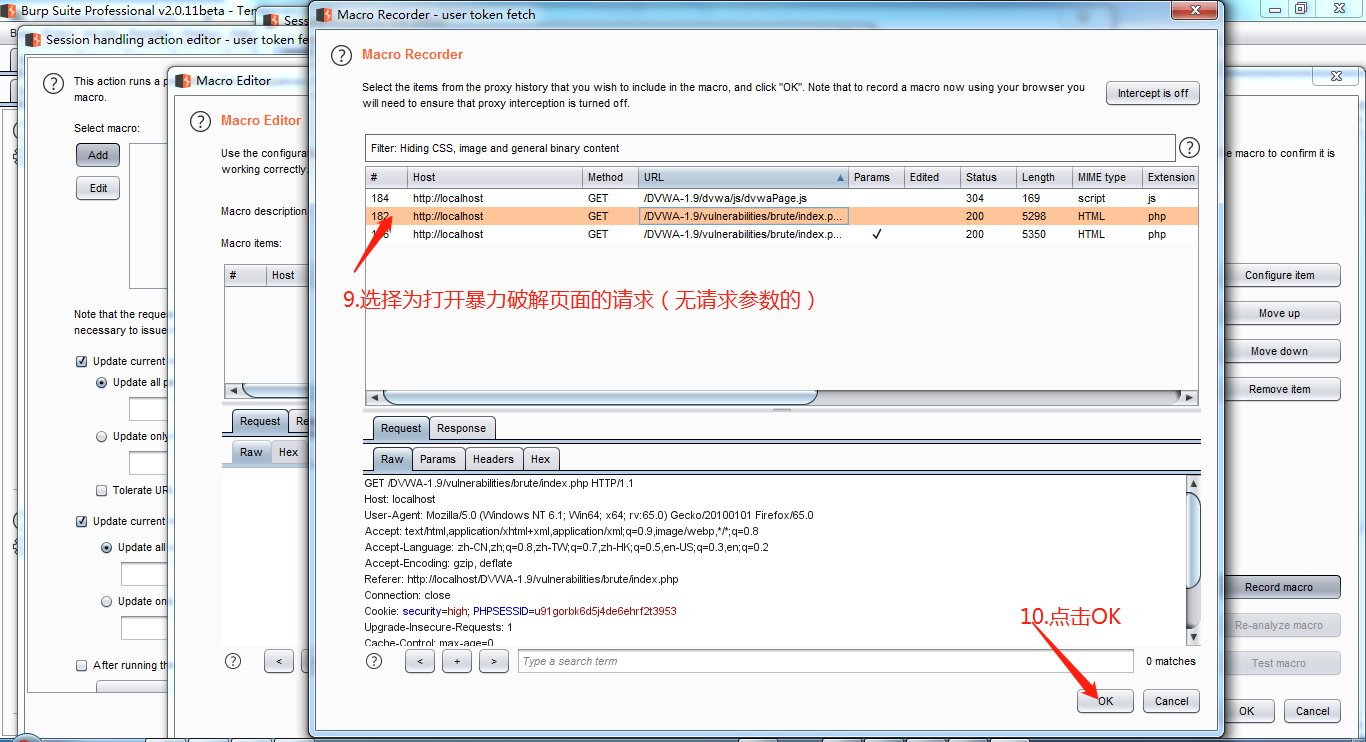
1. 使用firefox浏览器发起登陆请求

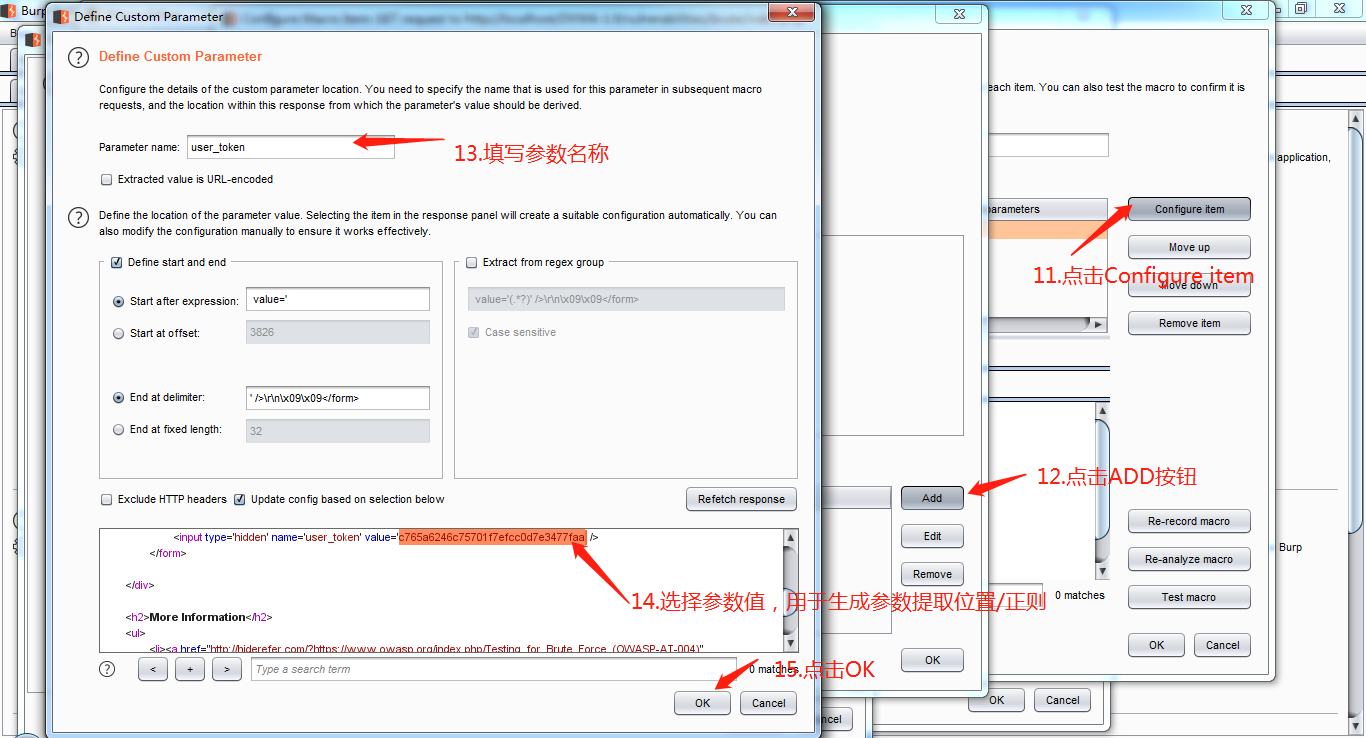


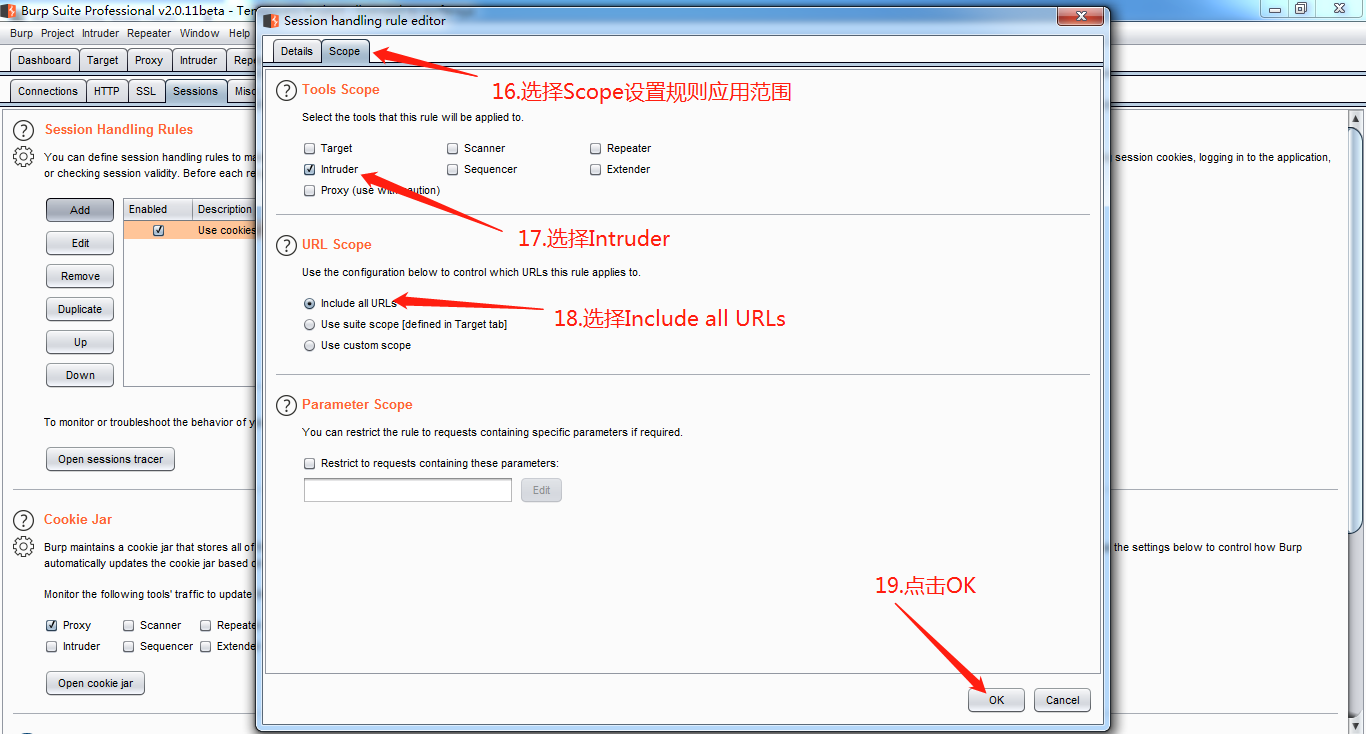
1. 定义burpsuite规则同时配置宏动作，用于user\_token自动提取和填充



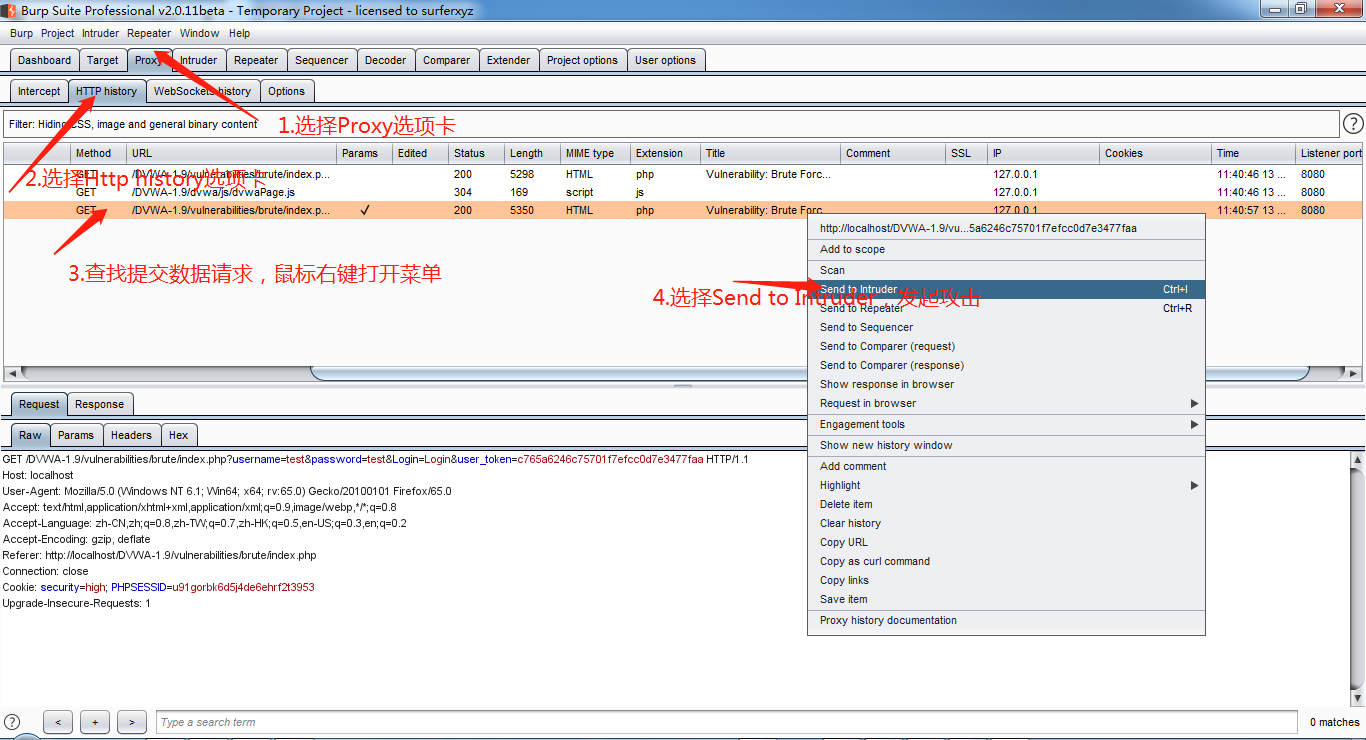




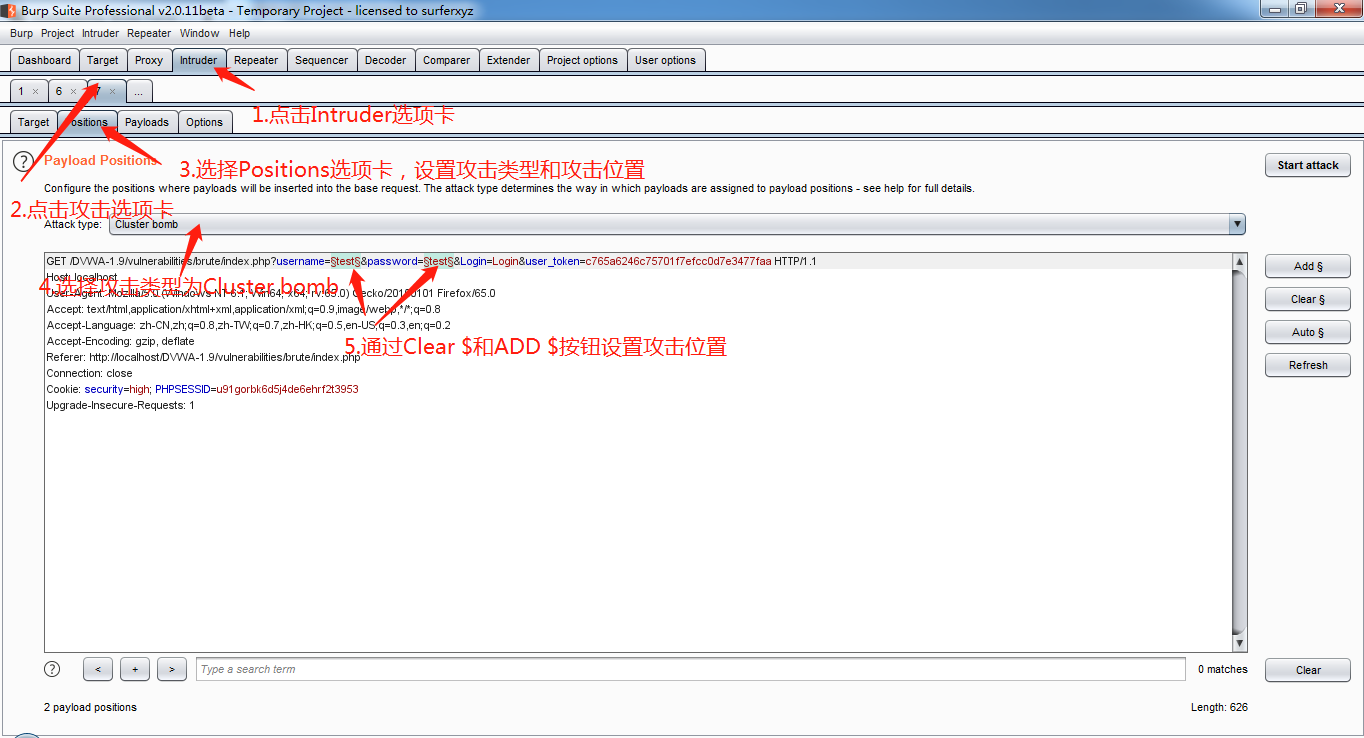




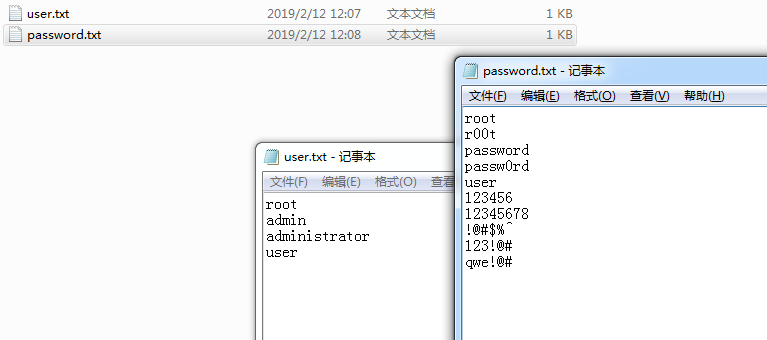
1. 使用burpsuite进行暴力破解



1. 设置攻击方式和攻击位置



1. 准备用户名和密码字典

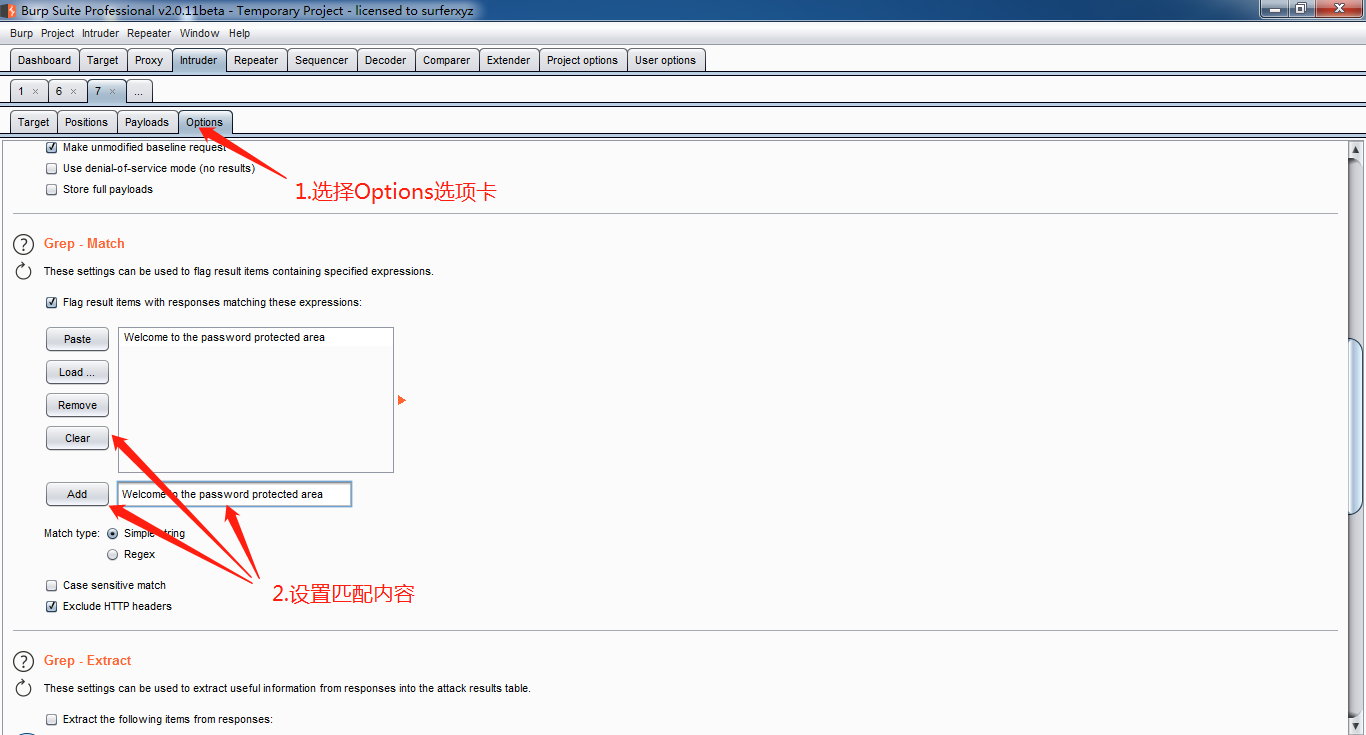


1. 设置playloads

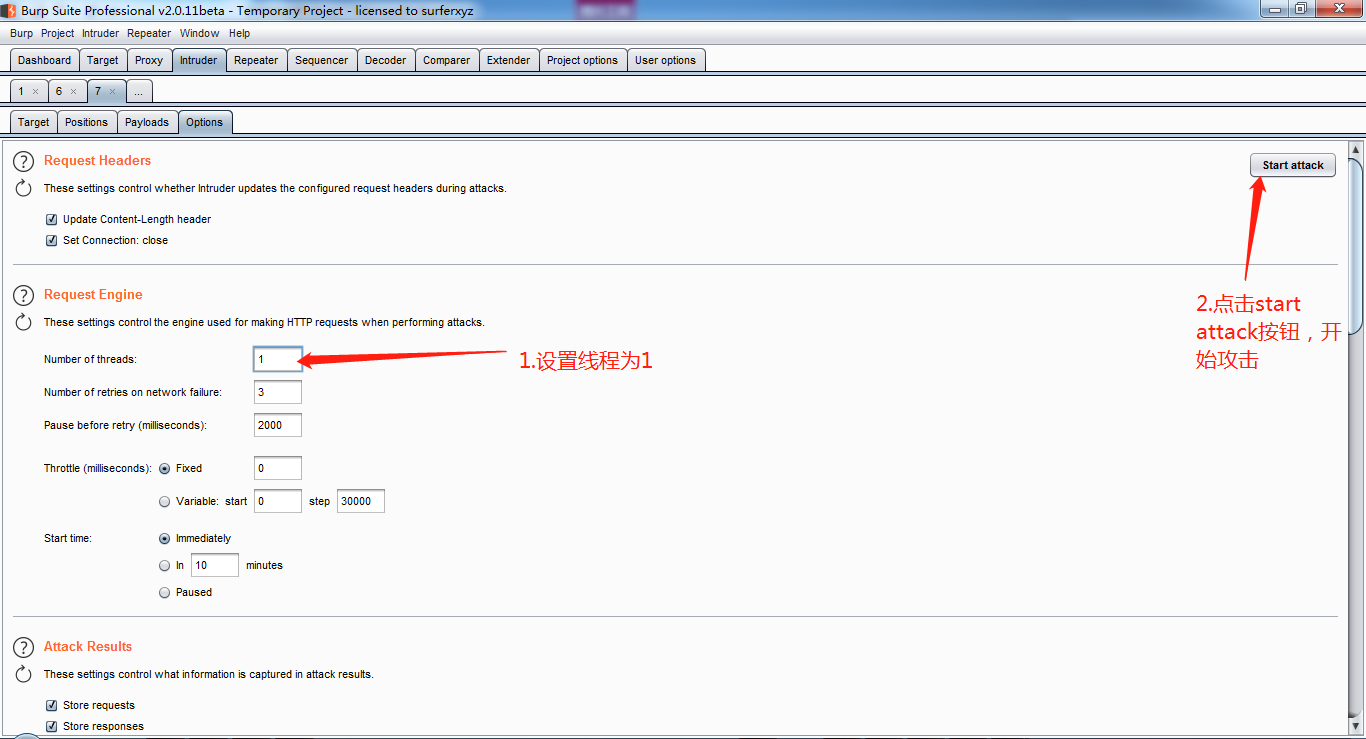




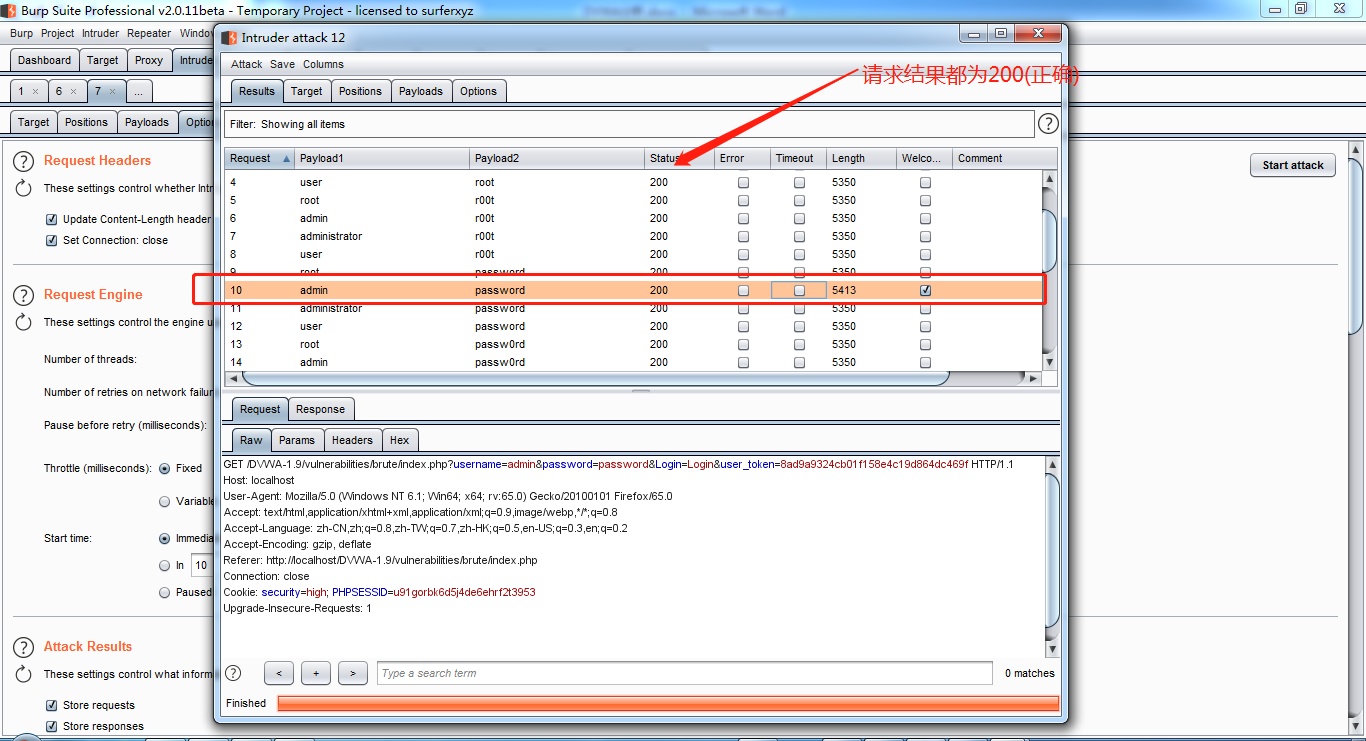
1. 设置结果匹配信息用以判断是否登陆成功



1. 开始攻击



1. 分析结果



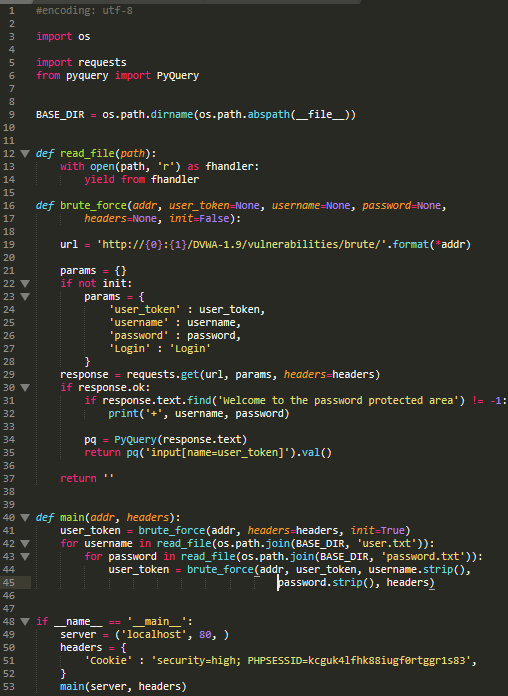
1. 代码分析



说明：

在每次请求检查随机参数token的正确性，在一定程度上增加了暴力破解的难度，同时对用户名和密码进行转义处理，预防SQL注入，在登陆过程失败的情况，休眠0-3秒返回结果，在一定程度上使破解攻击时间延长

1. 自编Python脚本暴力破解



使用：

* 从浏览器中copy登陆成功后的cookie信息
* 使用python3运行脚本

说明：

* 首次请求登陆页面html同时解析user\_token，在后续发起登陆请求时初始化user\_token参数，并从响应的html中user\_token作为下次请求参数的数据来源
* 使用requests发送请求到dvwa，根据请求结果是否包含Welcome to password protected area字符串判断是否破解成功

## 修复建议

1. 对于修改数据和登陆表单提交使用POST方式，同时数据通过POST方式读取
2. 添加随机token预防csrf攻击
3. 针对登陆功能可添加图形验证码，每提交一次数据，验证码改变一次，验证功能在服务端进行
4. 针对登陆次数进行限制，可使用登陆远程IP或用户名两种方式进行锁定，登录错误次数5分钟之内超过3次锁定1-3小时
5. 对于管理类系统配置登陆用户允许的IP范围
6. 可使用短信验证和邮箱验证方式实现双因子认证，注意对短信轰炸和邮件轰炸的防御
7. 密码等敏感字段进行加密后传输，例如密码使用加盐hash算法等加密后传输