《信息安全基础》实验报告

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. 完成对特定网站的 SQL 注入以获取数据库信息，对以下网站进行 SQL 注入：
http://pu2lh35s.ia.aqlab.cn/

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

3. 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。

1. SQL 注入

SQL 注入是一种常见的安全漏洞，它允许攻击者通过在应用程序的输入字段中插入恶意的 SQL 代码来执行未经授权的数据库操作。SQL 注入的原理是利用应用程序没有正确验证和过滤用户输入数据的漏洞。当应用程序接收用户输入并将其直接拼接到 SQL 查询语句中时，如果没有对输入进行适当的处理和过滤，攻击者可以在输入中注入恶意的 SQL 代码。

攻击者可以利用 SQL 注入漏洞执行各种恶意操作，例如：

（1）数据泄露：攻击者可以通过执行恶意的 SQL 查询语句获取敏感数据，如用户名、密码、个人信息等；

（2）数据篡改：攻击者可以修改数据库中的数据，包括插入、更新或删除记录，从而破坏数据的完整性；

（3）绕过身份验证和授权：攻击者可以通过修改 SQL 查询语句中的条件来绕过应用程序的身份验证和授权机制，获得未经授权的访问权限；

SQL 注入的关键在于攻击者能够将恶意的 SQL 代码注入到应用程序的查询语句中，并使数据库执行该恶意代码。为了防止 SQL 注入攻击，应用程序应该采用参数化查询或预编译语句，对输入数据进行适当的验证和过滤，并使用最小权限原则来限制数据库用户的权限。

2. 基本方法

SQL 注入攻击方法主要包括基于字符串的注入、数字型注入和布尔型注入。它们的原理和特点如下：

2.1 基于字符串的注入（String-based Injection）

原理：攻击者通过在字符串类型的输入字段中插入特殊的字符或字符串，改变 SQL 查询语句的结构，从而执行恶意的数据库操作。

特点：攻击者常常利用单引号（'）来终止正常的字符串，然后插入自己构造的 SQL 代码。这种注入方式适用于字符串类型的输入，如用户名、密码等。

2.2 数字型注入（Numeric-based Injection）

原理：攻击者通过在数字类型的输入字段中插入特殊的数值或运算符，改变 SQL 查询语句的语义，从而执行恶意的数据库操作。

特点：攻击者利用数值型注入来绕过数字输入的过滤和验证。他们可能使用注释符号、逻辑运算符或数学运算符，来构造恶意的 SQL 查询语句。

2.3 布尔型注入（Boolean-based Injection）

原理：攻击者通过在布尔类型的输入字段中插入特殊的逻辑表达式，利用应用程序的响应结果来推断 SQL 查询的真假，从而执行恶意的数据库操作。

特点：攻击者通过构造布尔型注入的查询条件，可以根据应用程序返回的不同响应结果来推断出 SQL 语句的真假。这种注入方式常用于布尔类型的查询条件，如登录验证等。

这些不同类型的注入攻击方法都是利用应用程序对输入数据的不正确处理和过滤而产生的。攻击者通过插入恶意的 SQL 代码来改变查询语句的结构或语义，从而执行未经授权的数据库操作。为了防止这些注入攻击，应用程序应该采用参数化查询或预编译语句来处理用户输入，并对输入数据进行适当的验证和过滤。

3. 攻击步骤

SQL 注入攻击的基本步骤如下：

（1）识别注入点：攻击者首先需要确定应用程序中存在潜在的注入点，即用户输入被直接拼接到 SQL 查询语句中的位置。这可以是表单字段、URL 参数、Cookie 等用户可控的输入；

（2）构造恶意的 SQL 代码：攻击者通过在注入点插入特定的恶意字符、语句或逻辑来构造恶意的 SQL 代码。目标是改变原始的查询语句结构或语义，以实现攻击者想要的数据库操作；

（3）执行注入攻击：构造好恶意的 SQL 代码后，攻击者将其提交给应用程序的注入点。应用程序在执行 SQL 查询时，会将恶意代码与正常的查询代码一起执行，从而导致注入攻击的发生；

（4）获取所需数据或控制权：成功执行注入攻击后，攻击者可以根据攻击的目的来获取所需的数据或控制权。这可能包括获取敏感数据、修改数据库内容或绕过身份验证和授权机制等；

需要注意的是，不同的注入攻击类型和具体的应用程序会有一些细微的差别，因此攻击者可能需要针对具体的情况进行一些调整和尝试。然而，以上步骤概括了 SQL 注入攻击的基本过程，攻击者通过构造恶意的 SQL 代码来利用应用程序的漏洞执行未经授权的数据库操作。为了防止注入攻击，应用程序应该采用参数化查询或预编译语句，并对用户输入进行适当的验证和过滤。

4. 防御措施

防御 SQL 注入的方法主要有参数化查询、预编译语句、输入验证、安全编码实践、安全漏洞扫描和漏洞修复、安全意识培训等。

（1）参数化查询或预编译语句：使用参数化查询或预编译语句是防止 SQL 注入的一种有效方法。这些技术通过将用户输入作为参数传递给 SQL 查询，而不是直接将输入拼接到查询语句中，从而避免了注入攻击。数据库系统能够正确处理参数化查询，确保输入数据被视为数据值而不是 SQL 代码；

（2）输入验证：进行输入验证是防止 SQL 注入的重要步骤。验证用户输入的数据类型、长度、格式和范围等，确保输入符合预期的规范。可以使用正则表达式、白名单验证或自定义验证逻辑来过滤和拒绝不符合规范的输入。此外，还可以对特殊字符进行转义或编码，以确保输入数据不会被误解为 SQL 代码；

（3）安全编码实践：编写安全的代码是防御 SQL 注入的关键。遵循安全编码实践，如使用准确的权限控制、最小权限原则、避免动态构造 SQL 查询语句、避免将用户输入直接拼接到查询中等，有助于减少注入漏洞的风险。应当使用框架和库提供的安全函数和方法，而不是自行编写解析和处理用户输入的代码；

（4）安全漏洞扫描和漏洞修复：定期进行安全漏洞扫描和测试，包括 SQL 注入的检测和评估。使用专门的安全工具或服务来扫描应用程序，识别潜在的注入漏洞，并及时修复这些漏洞；

（5）安全意识培训：提供给开发人员和系统管理员关于 SQL 注入和其他安全漏洞的培训和教育。加强他们的安全意识，使他们能够理解和遵循最佳的安全实践，从而减少注入漏洞的风险。

四、实验过程或算法

1. 检查单/双引号闭合

进入网站 http://pu2lh35s.ia.aqlab.cn/，点击页面上的链接后可以跳转到 http://pu2lh35s.ia.aqlab.cn/?id=1：

此时可以先利用 GET 进行传入 SQL 语句，分别检查是否存在单引号闭合或双引号闭合。传入单引号后发现页面有异常，因此不存在单引号闭合：

同理，发现也不存在双引号闭合，因此可以放心传入 SQL 代码了。

2. 查看 SQL 查询字段数

使用 Order by 语句看 SQL 语句向该表查询了多少字段，为后面的联合查询做准备，因为联合查询需要有相同的字段数。先向 URL 中传入 order by 1，即 http://pu2lh35s.ia.aqlab.cn/?id=1 order by 1，页面回显正常：

继续向 URL 中传入 order by 1,2，即 http://pu2lh35s.ia.aqlab.cn/?id=1 order by 1,2，页面回显也正常。当输入 order by 1,2,3 时，页面回显异常：


至此可以判断出当前 SQL 语句只查询了两个字段。

3. 查询数据库名称

通过 Order by 知道了字段数后，就可以进行联合查询。想要查询数据库名字，只需要加入 and 1=2 union select 1,database()：


可以看到数据库名称为 maoshe 。

4. 查询数据库中所有表的名称

继续注入 SQL 语句 and 1=2 union select 1,table_name from information_schema.tables where table_schema='maoshe'：


可以查询到一个名为 admin 的表。但该数据库中不一定只有一张数据表，上述 SQL 注入默认只查找第一个表，如果想要查找所有表，需要使用 limit 语句，如 limit 1,1 来跳过第 1 张表查询后面 100 张表。因此从 0,1，1,1，一直向后遍历，直到 4,1 才出现空页面：

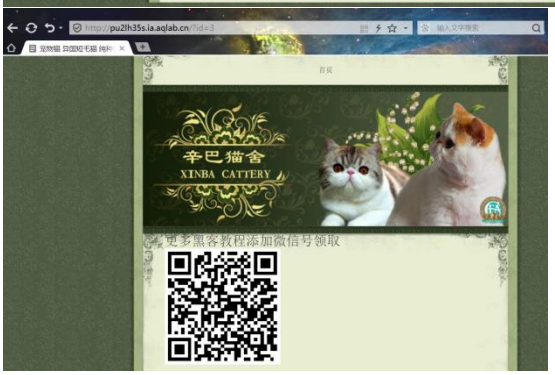
根据查询结果可知，maoshe 数据库下共有 admin、dirs、news、xss 四张数据表。

5. 查询表中字段

（1）admin 数据表

注入 SQL 语句 and 1=2 union select 1,column_name from information_schema.columns where table_name='admin'：

此处查询得到了一个 id 字段。但开始页面中点击超链接跳转的是 id=1，因此还可能有其他的字段。替换 URL 中的 id 查看，发现 id=0，id=1，id=2 和 id=3 字段都有数据，id=4 及以后就都是空页面：



因此断定该数据表 id 从 0 到 3 共有 4 列，其中第 0 列是主页。因此 admin 数据表有 3 个字段。

继续注入 union select 1,group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='admin' limit 1,100，可以查看表的字段名称：

可以看到 admin 数据表中字段的名称为 id、username 和 password。

（2）dirs 数据表

操作同上，可以得出 dirs 数据表有 1 个字段：

字段名称是 paths。

（3）news 数据表

操作同上，可以得出 news 数据表有 2 个字段：

字段名称分别是 id 和 content。

（4）xss 数据表

操作同上，可以得出 news 数据表有 3 个字段：

字段名称分别是 id、user 和 path。

6. 查看管理员用户密码

通过上面查询数据表中字段名称，想要获取管理员用户密码，就需要获取 admin 数据表中 username 和 password 字段的具体数据。注入 SQL 语句 and 1=2 union select 1,group_concat(username) from admin，可以查询登录用户名：

查询到用户名分别为 admin 和 ppt 领取微信。

继续注入 SQL 语句 and 1=2 union select 1,group_concat(password) from admin 可以查询密码：

查询到密码分别为 hellohack 和 zkaqbanban。

综上，管理员用户名和密码为：(admin, hellohack)。

五、实验过程中遇到的问题及解决情况

1. 问题：查询数据库中所有表的名称时，一开始只注入了 and 1=2 union select 1,table_name from information_schema.tables where table_schema='maoshe'，页面上只呈现了一张 admin 表；

解决办法：默认呈现第一张，需要使用 limit 语句遍历就可以查询到所有数据表。

2. 问题：某次访问网站还出现了’504 Gateway Timeout connect to remote host time out’的错误，无法访问网站；

解决办法：后来发现是网站服务器出现问题，只能等待网站修复后继续完成实验。

六、实验结果及分析和（或）源程序调试过程

1. 实验结果

通过 SQL 注入，获取到 http://pu2lh35s.ia.aqlab.cn/网站的数据库名称是 maoshe，数据库中有 4 张表分别是 admin、dirs、news 和 xss。admin 表中有 id、username 和 password 三个字段；dirs 表中有 paths 一个字段；news 表中有 id 和 content 两个字段；xss 表中有 id、user 和 path 三个字段。管理员的用户密码是 (admin, hellohack)。

防御 SQL 注入的方法主要有参数化查询、预编译语句、输入验证、安全编码实践、安全漏洞扫描和漏洞修复、安全意识培训：

（1）参数化查询或预编译语句：使用参数化查询或预编译语句是防止 SQL 注入的一种有效方法。这些技术通过将用户输入作为参数传递给 SQL 查询，而不是直接将输入拼接到查询语句中，从而避免了注入攻击。数据库系统能够正确处理参数化查询，确保输入数据被视为数据值而不是 SQL 代码；

（2）输入验证：进行输入验证是防止 SQL 注入的重要步骤。验证用户输入的数据类型、长度、格式和范围等，确保输入符合预期的规范。可以使用正则表达式、白名单验证或自定义验证逻辑来过滤和拒绝不符合规范的输入。此外，还可以对特殊字符进行转义或编码，以确保输入数据不会被误解为 SQL 代码；

（3）安全编码实践：编写安全的代码是防御 SQL 注入的关键。遵循安全编码实践，如使用准确的权限控制、最小权限原则、避免动态构造 SQL 查询语句、避免将用户输入直接拼接到查询中等，有助于减少注入漏洞的风险。应当使用框架和库提供的安全函数和方法，而不是自行编写解析和处理用户输入的代码；

（4）安全漏洞扫描和漏洞修复：定期进行安全漏洞扫描和测试，包括 SQL 注入的检测和评估。使用专门的安全工具或服务来扫描应用程序，识别潜在的注入漏洞，并及时修复这些漏洞；

（5）安全意识培训：提供给开发人员和系统管理员关于 SQL 注入和其他安全漏洞的培训和教育。加强他们的安全意识，使他们能够理解和遵循最佳的安全实践，从而减少注入漏洞的风险。

2. 实验总结

本次实验通过对 http://pu2lh35s.ia.aqlab.cn/ 网站的 SQL 注入攻击，获取了其数据库的相关信息并总结如何进行防范。在进行实验之前，我先了解了 SQL 注入攻击的原理与方法。SQL 注入是一种常见的网络安全漏洞，攻击者通过构造恶意的 SQL 查询语句，绕过应用程序的输入验证，直接访问或修改数据库的数据。攻击者可以利用这个漏洞获取敏感信息，甚至完全控制数据库。

在本次实验中，对目标网站进行了测试，发现该网站存在 SQL 注入漏洞。通过对网站 URL 进行检查，发现了可以通过参数注入的漏洞点。通过构造恶意的 SQL 查

询语句，我们成功地获取了数据库名称、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以及管理员用户密码。

通过本次实验，得出了一些对于 SQL 注入攻击的防范总结，主要有参数化查询、预编译语句、输入验证、安全编码实践、安全漏洞扫描和漏洞修复、安全意识培训等。总之，SQL 注入攻击是一种常见而严重的安全威胁，只有综合运用这些措施，才能有效地减少 SQL 注入攻击的风险，保障系统的安全性。

参考：https://www.bilibili.com/read/cv21888913/