摘要

一个数据库系统包括数据库本身和数据库应用程序，MySQL可以用来管理数据库，实现数据库应用程序的功能。SQL注入是针对数据库应用程序漏洞的一种对数据库本身的网络攻击，可以通过数据库应用程序的交互接口，向数据库管理代码或数据库本身中注入恶意代码，实现攻击目标。在本次实验项目中，我们将构建一个关于SQL注入的网络攻防场景，搭建数据库和数据库对应的应用网站，分析网站存在的SQL注入相关漏洞，用不同方法实现不同种类的SQL注入攻击，达成不同的攻击目标。同时，针对我们对SQL注入相关漏洞的分析，我们也实施了对不同种类的SQL注入预防措施，有效预防了SQL注入攻击。

1. 攻防所需知识
   1. 数据库与SQL语言概述
   2. SQL语言存在的漏洞

在进行基础的SQL注入攻击之前，我们需要再分析一下How Web Applications Interact with Database。

网页在获取用户输入的键值之后，用获取到的键值构造SQL语言的查询字符串，然后将这个查询字符串完整地发送到数据库端，执行该SQL命令。也就是说，表单中键入的数据最终将成为数据库执行的SQL命令字符串的一部分。即使用户不直接在数据库端与数据库进行命令交互，但在网页输入端处，用户和数据库之间确实存在一个通道。

用户和数据库之间存在的这个隐含的通道为数据库创造了攻击条件。如果对数据库接口编写方式保护不当，用户可能会在分析了解数据库接口编写代码的基础上，利用代码中对字符串的获取，对数据库本身发起攻击。

* 1. SQL注入的介绍

SQL注入即是指web应用程序对用户输入数据的合法性没有判断或过滤不严，攻击者可以在web应用程序中事先定义好的查询语句的结尾上添加额外的SQL语句，在管理员不知情的情况下实现非法操作，以此来实现欺骗数据库服务器执行非授权的任意查询，从而进一步得到相应的数据信息。

我们将构建一个关于SQL注入的网络攻防场景，搭建数据库和数据库对应的应用网站，分析网站存在的SQL注入相关漏洞，用不同方法实现不同种类的SQL注入攻击，达成不同的攻击目标。同时，针对我们对SQL注入相关漏洞的分析，我们也实施了对不同种类的SQL注入预防措施，有效预防了SQL注入攻击。

1. 攻击场景搭建

再开始攻防之前，我们需要先构建一个攻击场景，假设现在有一个数据库用于存储东南大学学生网络安全期末考试的成绩，学生可以通过一个老师搭建好的查询网页查询自己的期末考试成绩。

具体来说，这个数据库需要存储学生的考试成绩和相关身份信息，而查询网页则需要我们实现登录功能、用户管理功能（主要是密码修改）和成绩信息显示功能。

因此我们分别从数据库搭建、查询网站搭建和查询网站连接数据库三个步骤出发，介绍攻击场景搭建的实验过程

* 1. 数据库搭建

我们需要搭建一个存储学生考试成绩的数据库，该数据库除了要存储学生的考试成绩以外，还有对应存储学生的姓名、学号、登录密码和QQ，用于识别学生，将学生信息与其考试成绩匹配起来，同时学生的学号可以作为用户名，和登录密码一起用于成绩查询网页的登录和用户管理。

数据库的搭建主要由以下步骤实现：

1)

* 1. 查询网站搭建

搭建好数据库之后，我们需要搭建配套的成绩查询网站，这个网站由三个网页组成：用户登录页面、用户成绩信息页面和用户管理页面。

首先网站需要一个基本的用户登录页面，允许学生输入学号和登录密码进行登录。在学生输入学号和登录密码进行登录后，可以跳转到该学生学号所对应的成绩信息页面，该页面的内容即包括学生的考试成绩，也学生的相关信息，如学生的姓名、学号和QQ。最后，网站需要一个用户管理页面，为学生提供登录密码的修改接口。

在具体搭建之前，我们需要分析数据库的信息与查询网页之间的联系。主要有三个联系：

* 将网站的用户登录功能与数据库中的学号和登录密码建立联系：用户登录功能要求将输入的学号和登录密码与数据库中的数据进行对比，以确定是否登陆成功
* 将网站的用户成绩信息与数据库中的学生成绩和学生相关信息建立联系：网站需要从数据库中调取登录者所对应的姓名、学号、考试成绩和QQ，并在该学生登录后的成绩信息页面中显示出来
* 将网站的密码修改功能与数据库中的登录密码建立联系：应实现在用户管理页面上对密码进行修改的功能，操作者在网页上修改密码时，需要在数据库中对该操作者的登录密码进行同步修改。

之后，可以基于以上联系，搭建查询网站。查询网站的搭建主要由以下步骤实现：

1. SQL注入攻击

我们在搭建好的攻击场景中，进行SQL注入攻击的实验。依据既有的场景，可以进行的攻击主要分为三种，首先是Basic SQL Injection Attacks，其次是SQL Injection Attacks using cURL，最后是Modify Database。

利用SQL语言建立数据库时的不同漏洞，这三种攻击可以达成不同的攻击目标：

* 我们可以不需要密码直接登入别人的成绩查询页面，查看别人的成绩
* 我们可以直接使用URL获得完整的学生成绩数据库的信息
* 我们在修改密码的页面中，利用数据和代码不分离的SQL编写漏洞，更改数据库中自己的成绩，让自己的成绩更高。
* 我们在修改密码的页面中，利用数据和代码不分离的SQL编写漏洞，恶意更改数据库中别人的成绩，把别人的成绩改低。
  1. Basic SQL Injection Attacks

web应用程序开发人员开发了数据输入接口，意图在于让用户在接口内输入一些数据，实现交互功能。

本来用户在网页的数据输入接口内只应输入数据，但这些输入的数据都将成为代码中SQL命令的一部分。因此，用户可以通过在输入接口内注入伪装为数据的特殊代码片段，这种代码片段中包含有特殊的字符，如：“#”的注释字符，这样就可以利用这些特殊字符来更改代码中SQL语句的含义，以达成攻击目标。

我们将利用这个漏洞，在用户登录页面的学生学号输入接口内，注入伪装为数据的代码片段，实现“不需要密码直接登入别人的成绩查询页面，查看别人的成绩”的攻击目标。具体实现如下：

* 1. SQL Injection Attacks using cURL

除了直接在网页的交互接口中注入代码，使用命令行工具老发起攻击则更加方便。这样，我们无需进入图形用户界面，即可轻松地自动化攻击。使用cURL命令即可实现在命令行工具中发送伪装为数据的恶意代码。

我们将利用cURL命令，用命令行工具中编写URL，在URL中实现对数据交互接口的恶意代码注入。具体实现如下。

* 1. Modify Database

如果在网页的数据接口功能中，有修改数据的功能接口，该功能要求用户在网页端输入数据，然后用用户输入的数据更新数据库中的数据，在这类会插入或更改数据库中数据的接口中，我们将有机会在网页的接口端更改数据库。

在我们的攻击场景下，网站的用户管理页面可以对学生的密码进行修改，这就是一个会更新数据库数据的接口，我们可以利用这个接口，实现更改自己或他人成绩的攻击目标。

1. 修改自己的成绩
2. 修改别人的成绩
3. SQL注入防御

针对

* 1. Filtering and Encoding Data

SQL注入攻击时，攻击者通过各种特殊字符编写SQL注入语句。因此防范SQL注入要对用户输入进行规范，确保数据输入的安全性，在具体检查网页端接口中输入或提交的变量时，需要对会影响SQL语句功能的特殊字符进行转换或者过滤，从而有效防止SQL注入。

在我们的实验场景中，在将用户提供的学号等数据与SQL查询代码混合之前，先检查输入的数据，过滤掉其中任何可能被解释为代码的字符，如“#”，对特殊字符进行编码会告诉解析器将编码字符视为数据而不是代码。

PHP的mysqli扩展有一个名为mysqli：：real\u escape\u string（）的内置方法。

它可以用于对SQL中具有特殊含义的字符进行编码。下面的代码片段显示了如何使用此API。

* 1. Prepared Statement

通过告诉数据库参数的数据类型，可以降低 SQL 注入的风险。