课外实践作业一：SQL 注入实验

(1) 实验描述

在本次实验中，我们修改了 phpBB 的 web 应用程序，并且关闭了 phpBB 实现的一些对抗 SQL 注入的功能。因而我们创建了一个可以被 SQL 注入的 phpBB 版本。尽管我们的修改是人工的，但是它们代表着 web 开发人员的一些共同错误。学生的任务是发现 SQL 注入漏洞，实现攻击者可以达到的破坏，同时学习抵挡这样的攻击的技术。

(2) 实验环境

SEED Ubuntu 镜像

- 环境配置

实验需要三样东西，Firefox、apache、phpBB2（镜像中已有）：

① 运行 Apache Server：镜像已经安装，只需运行命令 %sudo service apache2 start

② phpBB2 web 应用：镜像已经安装，通过 http://www.sqllabmysqlphpbb.com 访问，应用程序源代码位于/var/www/SQL/SQLLabMysqlPhpbb/

③ 配置 DNS： 上述的 URL 仅仅在镜像内部可以访问，原因是我们修改了/etc/hosts 文件使 http://www.sqllabmysqlphpbb.com 指向本机 IP 127.0.0.1。如果需要在其他机器访问，应该修改 hosts 文件，使 URL 映射到 phpBB2 所在机器的 IP。

- 关闭对抗措施

PHP 提供了自动对抗 SQL 注入的机制，被称为 magic quote，我们需要关闭它。

1. 找到/etc/php5/apache2/php.ini
2. 找到 magic_quotes_gpc = On 这一行
3. 改为 magic_quotes_gpc = Off
4. 重启 Apache：”sudo service apache2 restart”

- Note for Instructors 最好拥有一些背景知识

1. 使用虚拟机，Firefox 的插件 LiveHttpHeaders 和 Tamper Data
2. 对 SQL 语句的一些了解
3. 如何操作 MySQL 数据库
4. 对 PHP 一些了解

(3) 实验任务

① 对 SELECT 语句的攻击

此次任务，你需要通过访问虚拟机内的 URL:www.sqllabmysqlphpbb.com。在进入 phpBB 之前系统会要求你登陆。这个登陆认证由服务器上的 login.php 实现，需要用户输入用户名和密码来通过认证。界面如下：

用户键入用户名和密码后，login.php 会将它们与 mysql 数据库中的 username 和 user_password 字段进行比较，如果匹配就登陆成功。和其他大多数 web 应用程序一样，PHP 程序使用 SQL 语言与背后的数据库交互。在 phpBB2 中，下面的语句实现了对用户的认证

```
SELECTuser_id,username,user_password,user_active,user_level,
    user_login_tries,user_last_login_try
    FROMUSERS_TABLE
    WHEREusername='$username'ANDuser_password='md5($password)';
if(foundonerecord)
then{allowtheusertologin}
```

对登陆的 SQL 注入攻击：以上语句存在 SQL 注入漏洞，你能利用这个漏洞来达到以下目的吗？

- 你能在不知道其他人的密码下登陆他的账号吗？
- 你能找到修改数据库的方法吗（依旧使用上面的 SQL 语句）？比如新建一个 database，或者删除一个用户的账号。很显然，上述语句是数据库查询语句，并不能修改（UPDATE）数据库。然而，使用 SQL 注入，你可以使上面的语句变成两句，用第二句来修改数据库。请尝试这种方法，看是否能够成功。

  坦白地说，我们没能够更改数据库，这是由于 MySQL 的防御措施。在报告中，你应该描述你如何尝试的，并找出为什么会失败，MySQL 使用了什么方法来抵御这样的攻击。你可以从网上寻找答案（第二手），当然你的第一手证据会有更多得分。万一你找到了方法成功地攻击了，那将得到一个额外加分。

② 对 UPDATE 语句的攻击

当用户想要在 phpBB2 中修改他们的资料时，可以点击 Profile，然后填写表单修改。用户发送修改请求后，会执行 include/usercp_register.php 中的一条 UPDATE SQL 语句。在这条语句中同样有一个 SQL 注入漏洞，请用它来达到以下目标：

在不知道其他人密码的情况下修改其资料。例如：你以 Alice 登陆，你的目标就是修改 Ted 的资料信息，包括他的密码。攻击成功后你将可以登陆 Ted 的账号。

③ 对抗 SQL 注入

SQL 注入漏洞的根本原因是没有将代码和数据区分开。当组建一个 SQL 语句时，程序（如 PHP 程序）知道哪个部分是代码哪个部分是数据。不幸的是当 SQL 语句送往数据库执行时，这个边界被打破，当程序被注入时，SQL 解释器看到的边界可能和最初的边界不一样。为了解决这个问题，保持服务端程序和数据库看到的边界一样就十分重要。

- 使用 magic_quotes_gpc 避开特殊字符。

PHP 代码中的字符串类型的数据需要用单引号(')引起来。比如上面的任务中 username='$username'，单引号用于把$username 从代码中区分出来。不幸的是如果$username 中含有单引号，这个区分将被打破。我们需要一个机制告诉数据库$username 中的单引号应该被当做数据，而不是 SQL 语句的特殊字符。为此我们只需要在单引号前加一个反斜杠(\)

PHP 提供了自动在单引号、双引号、反斜杠和空字符前添加反斜杠的机制，如果这个选项启用，那么所有这些从用户输入的特殊字符会被加反斜杠。启用方法为，修改 /etc/php5/apache2/php.ini 的 magic_quotes_gpc = On，然后重启 Apache。

请分别启用/禁用该功能来看这个机制是如何实现保护的。（php5.3.0 后该功能弃用）

- 使用 addslashes()来避开特殊字符。

PHP 的方法 addslashes()可以达到 magic quote 同样的功能。如果 magic quote 没有弃用，phpBB2 的代码会使用该功能来防止 SQL 注入攻击。请查看 /var/www/SQL/SQLLabMysqlPhpbb 目录下的 common.php。实际上为了使 SQL 注入攻击成功，我们注释掉了 phpBB2 的保护措施。

请修改回 phpBB2 的保护功能，并观察移除下面代码中"and FALSE"前后的区别，描述这个保护机制如果防止 SQL 注入攻击的。

if(!get_magic_quotes_gpc() and FALSE)

为了帮助描述区别，你需要打印出 SQL 语句。

- 使用 mysql_real_escape_string 避开特殊字符

一个较好的方法来防止 SQL 注入攻击是使用数据库的回避机制。MySQL 提供了一个机制，叫 mysql_real_escape_string()，它在一些特殊字符前加反斜杠，包括：\x00, \n, \r, \, ', ", 和 \x1A。

请使用该功能来修复前面任务中的 SQL 注入漏洞（需要禁用前面提到的保护机制）。

- Prepare Statement

一个更平常的解决方法是将数据和 SQL 逻辑区分开来准确地告诉数据库哪是数据部分以及哪是逻辑部分。MySQL 提供了 Prepare Statement 机制来达到这一目的。

```
$db=newmysqli("localhost","user","pass","db");
$stmt=$db->prepare("SELECT * FROMusersWHEREname=?ANDage=?");
$stmt->bind_param("si",$user,$age);
$stmt->execute();
```

使用该机制，我们将发送 SQL 语句分为几步。第一步先发送代码，如需要下一步填入数据的 SQL 语句，这是进行准备的一步。然后我们再用 bind_param()发送数据至数据库。这样数据库就可以区分代码和数据了。

请使用该机制来修复前面的 SQL 注入漏洞。（bind_param 函数中的第一个参数"si"的意思是第一个参数$user 是字符串类型，第二个参数$age 是整数类型）。