

《漏洞利用及渗透测试基础》实验报告

姓名：张洋 学号：2111460 班级：信安二班

实验名称：

WEB 开发实践

实验要求：

复现课本第十章的实验三（10.3.5 节）：利用 php，编写简单的数据库插入、查询和删除操作的示例。基于课本的完整的例子，进一步了解 WEB 开发的细节。

实验过程：

1. 安装 PHPnow

```
正在安装 Apache ...

正在启动 Apache ...

启动 Apache 完成;

正在启动 MySQL 5.0 ...

Service successfully installed.
MySQL5_pn 服务正在启动 .
MySQL5_pn 服务已经启动成功。

启动 MySQL 5.0 完成;

现在为 MySQL 的 root 用户设置密码. 重要! 请切记!
-> 设置 root 用户密码: 123456

MySQL root 用户的新密码为 "123456" , 请切记!

全部完成!! 你将可以看到 PHPnow 的默认页面!

- 按任意键继续... ■
```

安装成功后启动 PHPnow



打开网页，访问 <http://127.0.0.1> 如下：

127.0.0.1

Let's **PHP** now !

为何只能本地访问?
此服务器互联网 IP
221.238.245.39

| Server Information | |
|--------------------|---|
| SERVER_NAME | 127.0.0.1 |
| SERVER_ADDR:PORT | 127.0.0.1:80 |
| SERVER_SOFTWARE | Apache/2.0.63 (Win32) PHP/5.2.14 |
| PHP_SAPI | apache2handler |
| php.ini | D:\PHPnow-1.5.6\php-5.2.14-Win32\php-apache2handler.ini |
| 网站主目录 | D:/PHPnow-1.5.6/htdocs |
| Server Date / Time | 2023-05-09 00:02:02 (+08:00) |
| Other Links | phpinfo() phpMyAdmin |

| PHP 组件支持 | |
|----------------|-----------------------------------|
| Zend Optimizer | Yes / 3.3.3 |
| MySQL 支持 | Yes / client lib version 5.0.90 |
| GD library | Yes / bundled (2.0.34 compatible) |
| eAccelerator | No |

| MySQL 连接测试 | | | |
|------------|--|------------|-----------------------------------|
| MySQL 服务器 | <input type="text" value="localhost"/> | MySQL 数据库名 | <input type="text" value="test"/> |
| MySQL 用户名 | <input type="text" value="root"/> | MySQL 用户密码 | <input type="text"/> |
| | | | <input type="button" value="连接"/> |

2. 输入密码 123456 查看数据库连接是否正常：

| MySQL 连接测试 | | | |
|------------|-----------|------------|-----------------------------------|
| MySQL 服务器 | localhost | MySQL 数据库名 | test |
| MySQL 用户名 | root | MySQL 用户密码 | |
| | | | <input type="button" value="连接"/> |

| MySQL 测试结果 | |
|---------------|--------------------------|
| 服务器 localhost | OK (5.0.90-community-nt) |
| 数据库 test | OK |

3. 点击 phpMyAdmin, 用户名密码为 root/123456, 登录进数据库管理系统。创建一个新的数据库：

localhost ▶ testDB

结构 SQL 搜索 查询 导出 导入 操作 权限 删除

✓ 创建数据库 testDB 成功。

```
CREATE DATABASE `testDB` ;
```

4. 新建表 userinfo, 属性为 username 与 pwd, 点击保存：

localhost ▶ testDB ▶ userinfo

浏览 结构 SQL 搜索 插入 导出 导入 操作 清空 删除

✓ 创建数据表 `testDB`.`userinfo` 成功。

```
CREATE TABLE `testDB`.`userinfo` (
  `username` VARCHAR(30) NOT NULL,
  `pwd` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE = MYISAM ;
```

| | 字段 | 类型 | 整理 | 属性 | 空 | 默认 | 额外 | 操作 |
|--------------------------|----------|-------------|-------------------|----|---|----|----|----|
| <input type="checkbox"/> | username | varchar(30) | latin1_swedish_ci | | 否 | 无 | | |
| <input type="checkbox"/> | pwd | varchar(30) | latin1_swedish_ci | | 否 | 无 | | |

↑ 全选 / 全不选 选中项:

5. 插入数据, admin/admin

✓ 已插入 1 行。

```
INSERT INTO `testDB`.`userinfo` (
  `username`,
  `pwd`
)
VALUES (
  'admin', 'admin'
);
```

6. 使用 DW 软件创建 html 文件, 保存在 hpdocs 文件夹下。修改 html 文件文档。添加表单 form1, 添加 action, 添加表单对象为文本。添加 username, password 文本以及提交按钮。具体代码如下：

```

<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=gb2312" />
  <title>无标题文档</title>
</head>

<body>
  <form id="form1" name="form1" method="get" action="loginok.php">
    <label>username:
    <input name="username" type="text" id="username" />
  </label>
  <p>
    <label>password:
    <input name="password" type="text" id="password" />
  </label>
  </p>
  <p>
    <label>
    <input type="submit" name="Submit" value="提交" />
  </label>
  </p>
</form>
</body>
</html>

```

实现效果为：



← → ↻ ⓘ 127.0.0.1/login.html

username:

password:

7. 设置 username 为 admin，密码为 admin 得出如下效果：显然 get 模式会将具体的值显示在 URL 中。

127.0.0.1/loginok.php?username=admin&password=admin&Submit=%CC%E1%BD%BB

8. 在htdocs文件夹下新建 login.php 文件，打开并编辑代码如下：

```

<?php
$username=$_GET['username'];
$password=$_GET['password'];
echo $username;
?>

```

9. 再次刷新网页即可得到如下结果，输出 admin(username)。

← → ↻ ⓘ 127.0.0.1/loginok.php?username=admin&password=admin&Submit=%CC%E1%BD%BB
admin

10. 修改 html 文件中方式为 post，再次回到登陆页面并输入用户名密码为 admin/admin，结果如下：

← → ↻ ⓘ 127.0.0.1/loginok.php

Notice: Undefined index: username in D:\PHPnow-1.5.6\htdocs\loginok.php on line 2
Notice: Undefined index: password in D:\PHPnow-1.5.6\htdocs\loginok.php on line 3

11. 再次将 php 中获取方式改为 POST 方式得结果：

← → ↻ ⓘ 127.0.0.1/loginok.php
admin

12. 在 php? 范围外的文本将被直接输出：

```
1 助教老师您好
2  <br />
3  <?php
4  $username=$_POST['username'];
5  $password=$_POST['password'];
6  echo $username;
7  ?>
```

← → ↻ ⓘ 127.0.0.1/loginok.php
助教老师您好
admin

13. 添加构造 SQL 语句如下：

```
<?php

$conn=mysql_connect("localhost","root","123456");//连接数据库
$username=$_POST['username'];
$password=$_POST['password'];
$SQLstr="select * from userinfo where username=' $username' and
pwd=' $password' ";
echo $SQLstr;
$result=mysql_db_query("testDB",$SQLstr,$conn);//执行数据库语
句
//获取查询结果
if($row=mysql_fetch_array($result))
```

```
        echo "<br>OK<br>";
    else
        echo "<br>false<br>";
    //释放资源
    mysql_free_result($result);
    //关闭连接
    mysql_close($conn);
?>
```

结果如下:

← → ↻ ⓘ 127.0.0.1/loginok.php

```
select * from userinfo where username='1' and pwd='1'
false
```

若输入之前插入的数据 admin/admin, 结果为:

← → ↻ ⓘ 127.0.0.1/loginok.php

```
select * from userinfo where username='admin' and pwd='admin'
OK
```

14. 修改代码实现以下功能: 输入正确的用户名密码跳转至 sys.html 界面, 否则回到历史界面

```
<?php
$isOK=0;
$conn=mysql_connect("localhost","root","123456");//连接数据库
$username=$_POST['username'];
$password=$_POST['password'];
$SQLstr="select * from userinfo where username='$username' and
pwd='$password'";
echo $SQLstr;
$result=mysql_db_query("testDB",$SQLstr,$conn);//执行数据库语句
//获取查询结果
if($row=mysql_fetch_array($result))
    $isOK=1;
//释放资源
mysql_free_result($result);
//关闭连接
mysql_close($conn);

if($isOK==1)
{
```



```
?>
<script language="javascript">
alert("ok")
window.location.href="sys.php";
</script>
<?php
}else{
    ?>
    <script language="javascript">
    alert("false");
    history.back();
    </script>
    <?php
}
?>
```

输入之前插入的数据 admin/admin 结果为:

127.0.0.1 显示
ok

确定

← → ↻ 127.0.0.1/sys.php

Not Found

The requested URL /sys.php was not found on this server.

Additionally, a 404 Not Found error was encountered while trying to use an ErrorDocument to handle the request.

Apache/2.0.63 (Win32) PHP/5.2.14 Server at 127.0.0.1 Port 80

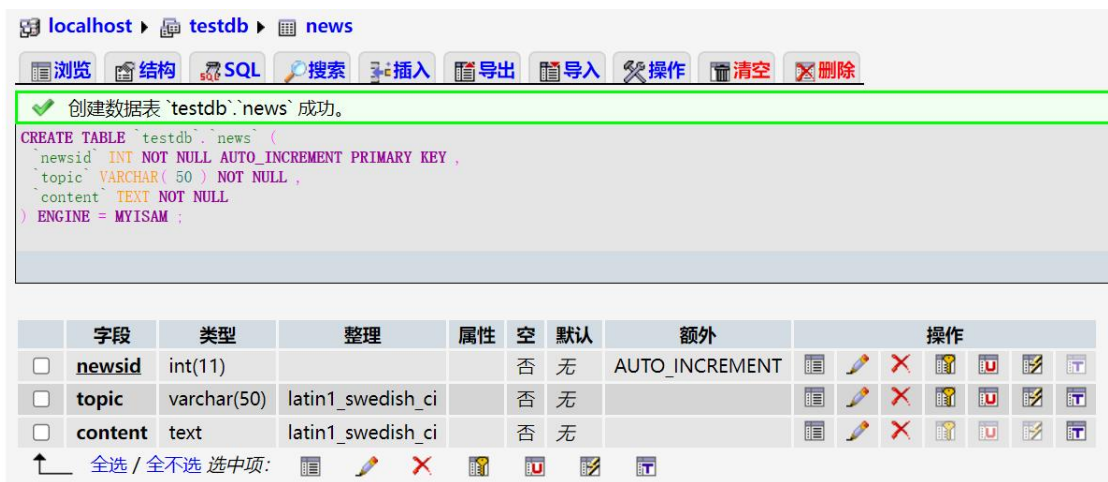
输入 1/1 结果为:

127.0.0.1 显示
false

确定

点击确定后返回上一页。

15. 在 testdb 数据库中新建 news 表, 属性为 newsid, topic, content。



新建 sys.php 页面，布局为 topic 为标签，一个文本输入框，content 为标签，一个文本区域用于输入。具体代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"
/>
<title>无标题文档</title>
</head>

<body>
<form id="form1" name="form1" method="post" action="addok.php">
  <label>topic:
  <input name="topic" type="text" id="topic" />
</label>
<p>
  <label>content:
  <textarea name="content" cols="60" rows="8"
id="content"></textarea>
</label>
</p>
<p>
  <label>
  <input type="submit" name="Submit" value="提交" />
</label>
</p>
</form>
</body>
</html>
```

新建 addok.php 文件，实现功能为对 news 表实现插入功能，其具体功能为读取 sys.php 页面中获取的 topic 和 content 值，并且插入到 news 表中。实现代码如下：

```
<?php
```



```

$conn=mysql_connect("localhost","root","123456");//连接数据库
mysql_select_db("testDB");
$topic=$_POST['topic'];
$content=$_POST['content'];
$SQLstr="insert          into          news(topic,          content)
values(' $topic',' $content)";
echo $SQLstr;
$result=mysql_query($SQLstr);//执行数据库语句

//关闭连接
mysql_close($conn);

if($result)
{
    ?>
    <script language="javascript">
    alert("add ok")
    window.location.href="sys.php";
    </script>
    <?php
} else {
    ?>
    <script language="javascript">
    alert("add false");
    history.back();
    </script>
    <?php
}
?>

```

在页面中成功登陆后即可实现插入操作：

←
→
↺
127.0.0.1/sys.php

topic:

content:

1111







+ 选项

| | | | |
|--|--------|-------|---------|
| ←T→ | newsid | topic | content |
| <input type="checkbox"/>   | 1 | 1 | 1111 |

16. 新建 html 文件，另存为为 news.php，实现功能为显示出 news 表中所存信息的 id 和 topic，在 topic 处显示链接。具体实现代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"
/>
<title>无标题文档</title>
</head>

<body>
<table width="600" border="1" align="center">
  <tr>
    <td>id</td>
    <td>topic</td>
  </tr>
  <?php
    $conn=mysql_connect("localhost","root","123456");
    $SQLstr="select newsid,topic from news";
    $result=mysql_db_query("testDB",$SQLstr,$conn);
    if($row=mysql_fetch_array($result))//通过循环读取数据内容
    {
      //定位到第一条记录
      mysql_data_seek($result,0);
      //循环取出记录
      while($row=mysql_fetch_row($result))
      {
        ?>
        <tr>
          <td><?php echo $row[0];?></td>
          <td><?php echo $row[1];?></td>
        </tr>
        <?php
          }
        }
      ?>
    </table>
  </body>
</html>
```

| ←T→ | newsid | topic | content |
|--|--------|-------|---------|
| <input type="checkbox"/>   | 1 | 1 | 1111 |
| <input type="checkbox"/>   | 2 | 22 | 2222 |
| <input type="checkbox"/>   | 3 | 444 | 4444 |

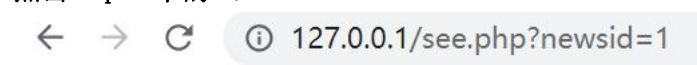
| id | topic |
|----|-------|
| 1 | 1 |
| 2 | 22 |
| 3 | 444 |

17. 新建 see.php, 实现对该信息的查看。

```
<body>
<?php
$conn=mysql_connect("localhost","root","123456");
$id=$_GET['newsid'];
$SQLstr="select * from news where newsid='$id'";
$result=mysql_db_query("testDB",$SQLstr,$conn);
if($row=mysql_fetch_array($result))//通过循环读取数据内容
{
    //定位到第一条记录
    mysql_data_seek($result,0);
    //循环取出记录
    while($row=mysql_fetch_row($result))
    {
        ?>
        topic:<?php echo $row[1];?><br />
        content:<?php echo $row[2];?><br />
    }
}
?>
</body>
```

| id | topic |
|----|---------------------|
| 1 | 1 |
| 2 | 22 |
| 3 | 444 |

点击 topic 中的 1:



topic:1
content:1111

18. 增加删除功能。更改 sys.php, 添加 delete.php。

sys.php:

```
<body>
<form id="form1" name="form1" method="post" action="delete.php">
  <label>delete topic:
  <input name="topic" type="text" id="topic" />
</label>
</p>
<p>
  <label>
    <input type="submit" name="Submit" value="提交" />
  </label>
</p>
</form>
</body>
```

delete.php:

```
<?php
$conn=mysql_connect("localhost","root","123456");//连接数据库
mysql_select_db("testDB");
$topic=$_POST['topic'];
$SQLstr="delete from news where topic='$topic'";
echo $SQLstr;
$result=mysql_query($SQLstr);//执行数据库语句

//关闭连接
mysql_close($conn);

if($result)
{
  ?>
  <script language="javascript">
    alert("delete ok");
    window.location.href="sys.php";
  </script>
  <?php
}else{
  ?>
  <script language="javascript">
    alert("delete false");
    history.back();
  </script>
  <?php
}
?>
```

Sys. php 的页面显示为:

← → ↻ ⓘ 127.0.0.1/sys.php

delete topic:

提交

删除 topic 等于 1 的新闻后显示:





127.0.0.1 显示

delete ok

确定

查看数据库，成功删除 topic=1 的新闻:

+ 选项

| ← T → | newsid | topic | content |
|--|--------|-------|---------|
| <input type="checkbox"/>   | 2 | 22 | 2222 |
| <input type="checkbox"/>   | 3 | 444 | 4444 |

心得体会:

这次实验让我学会了如何在 PHP 中利用 MySQL 数据库进行简单的插入、查询和删除操作，同时也熟悉了 Dreamweaver 的基本使用。

下面是我在本次实验中的收获和思考:

1. 编程语言的切换

通过这次实验，我更深刻地理解到编程语言的重要性。不同的编程语言适用于不同的场景和领域，在日常工作和学习应用时需要注意选择合适的编程语言，提高开发效率和代码质量。

2. 数据库的建立

对于建立一个有竞争力的 Web 应用程序来说，理解如何设计和管理数据库极为重要。学习如何有效地组织数据并应用 SQL 是核心技能之一。

3. 前后端分离

在本次实验中，我使用 PHP 编写服务器端代码，而在 Dreamweaver 中构建用户界面。整个系统的后端和前端都有明确的职责划分。这让我对前后端分离有了更加深刻的理解，并意识到这种思想在现代 Web 开发中越来越重要。

4. 熟练使用开发工具

我在本次实验中重温了并熟练掌握了 Dreamweaver 的基本功能，同时也更深刻地理解到代码编辑器在 Web 开发中的作用和重要性。良好的开发工具不仅可以提高开发效率，还可以优化代码结构和风格。

总之，这次实验对我的 Web 开发和数据库大作业有很大帮助，我学习到了很多知识。