实验名称：实现本地 Web 攻击

实验原理：大多数 Web 应用程序攻击都是来源于 XSS、CSRF 和 SQL 注入攻击，这些攻击通常指的是通过利用网页开发时留下的漏洞，通过巧妙的方法注入恶意指令代码到网页，使用户加载并执行攻击者恶意制造的网页程序，其中 CSRF 存在是指攻击者构建的恶意网站被用户访问后，返回一些攻击性代码，并发出一个请求要求访问第三方站点，从而盗用用户身份，如用户名义发送邮件、虚拟货币转账等。

实验环境：一台 Ubuntu23.04 虚拟机

实验步骤：

1. 配置实验环境
   1) 本实验使用一台基于 VMware Workstation Pro 搭载 Ubuntu23.04 的虚拟机 Website 进行演示。系统信息如下图所示。

   2) 虚拟机 Website 在本实验中模拟用来搭建网站的服务器。在终端中输入命令 sudo apt update 和 sudo apt install python3-flask -y 安装 Flask 框架。
   3) 创建文件夹 XSStest，其中包含文件夹 templates 和网站运行程序 app.py，文件夹 templates 中存放网页文件 index.html。
   4) 运行 app.py，打开浏览器输入访问 localhost:5000 即可看到网站创建成功。

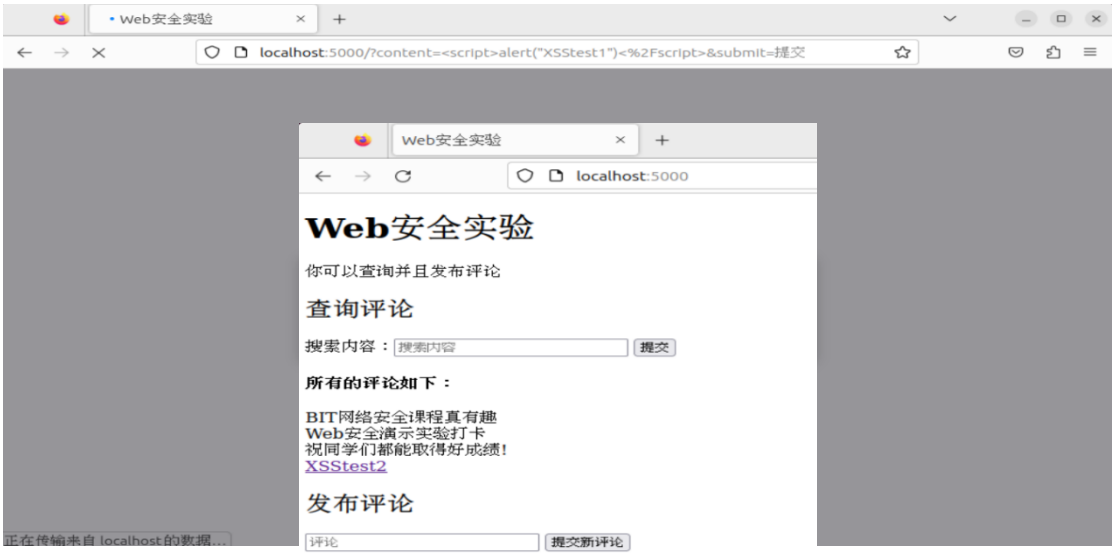
```
打开(O)                                    index.html
                                           ~/桌面/XSStest/templates
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Web安全实验</title>

</head>
<body>
<h1>Web安全实验</h1>
你可以查询并且发布评论
<br>
<h2>查询评论</h2>
<form action="" method="get">
    搜索内容: <input type="text" name="content" placeholder="搜索内容">
    <input type="submit" name="submit" value="提交">
</form>
{%if query%}
<h4>查询评论 {{query|safe}} 结果如下: </h4>
{%else%}
<h4>所有的评论如下: </h4>
{%endif%}
<!-- {{ "<script>alert('XSS')</script>" }} -->
{% for comment in comments %}
<td>{{comment|safe}}</td>
<br>
{% endfor %}
<h2>发布评论</h2>
<form action="" method="post">
    <input type="text" name="newComment" placeholder="评论">
    <input type="submit" name="submit" value="提交新评论">
</form>
</body>
</html>
```

2. 实现 XSS 攻击

1) 实现 XSS 反射型攻击。在网站“搜索内容”输入框内输入攻击脚本 <script>alert(“XSStest1”)</script>，点击提交，弹出弹窗，攻击成功，如下图所示。



2) 实现 XSS 存储型攻击。在网站“评论”输入框内输入攻击脚本 XSStest2，点击“提交新评论”，评论区增加“XSStest2”评论，点击该评论，跳转到百度首页，攻击成功，如下图所示。

3. 防御 XSS 攻击

1) 修改 app.py，从 flask 引入 Markup 类，Markup 是对 HTML 的一种安全标记，并将其转化为 str 类型以防止 XSS 攻击。该类中的 escape()方法可以将输入转义为字符串。使用 Markup 类中的 escape()方法将输入的 comment 和 query 变量进行转义。

打开(O) · app.py ~/桌面/XSStest

```
from flask import Flask, render_template, request, Markup

app = Flask(__name__)

dataset=["BIT网络安全课程真有趣","Web安全演示实验打卡","祝同学们都能取得好成绩！"]


@app.route("/", methods=["GET", "POST"])
def index():
    query = ""
    if request.method == "POST":
        if request.form.get("submit") == "提交新评论":
            comment = request.form.get("newComment").strip()
            comment = Markup.escape(comment)
            print(type(comment))
            if comment:
                dataset.append(comment)
    elif request.method == "GET":
        if request.args.get("submit") == "提交":
            query = request.args.get("content").strip()
            query = Markup.escape(query)
            if query:
                sub_dataset = [x for x in dataset if query in x]
                return render_template("index.html", query=query, comments=sub_dataset)
    return render_template("index.html", query=query, comments=dataset)


if __name__ == "__main__":
    app.run()
```

2) 重新运行网站，测试第 2 步 1)和 2)中的两种攻击脚本，发现均被转义，无法攻击，防御成功，结果如下图所示。


Web安全实验
localhost:5000
Web安全实验
你可以查询并且发布评论
查询评论
搜索内容：
搜索内容
提交
所有的评论如下：
BIT网络安全课程真有趣
Web安全演示实验打卡
祝同学们都能取得好成绩!
XSStest2
发布评论
评论
提交新评论