仿 airdrop 实现

AirDrop（隔空投送）作为苹果公司 iOS，iPadOS 和 macOS 系统下特有的功能，十分轻松就能使得多台设备之间进行文件的分享，因此，我们小组仿 AirDrop，实现了一个可在相同局域网下进行文件互传、剪贴板复制粘贴的程序。

一、关键功能

我们小组实现的“局域网传输文件”项目的服务端运行在 Windows 平台，服务端运行后，服务端本身和与服务端平台处于同一局域网下的设备可以通过访问服务端的 5000 号端口进入服务页面。

实现的主要功能如下：

1. **文件互传：**在不同设备上访问服务页面，都可选择本地的文件进行上传，上传至服务端主机上；不同设备都可以看到已上传的文件并选择下载。
2. **共享剪贴板内容：**服务端剪切板的内容可以实时共享，不同设备在访问服务页面时都可以看到其内容。

二、关键代码

（一）实现思路及方案：

基于 Python，使用 tkinter 实现服务端的 GUI 程序，服务页面采用 Flask 框架，threading 创建线程运行服务器。

文件互传的实现：用户选择上传的文件会保存在服务端的 tmp_files 文件夹下，在页面中会显示这些文件。当用户点击某一个文件名时，这个文件名链向形式如 http://10.31.23.3:5000/tmp_file/xxx.zip 的链接，即访问服务端的 5000 端口对应文件夹下 tmp_files 文件夹中的 xxx.zip。

剪切板共享的实现：每次访问服务页面时，服务端都会运行 Tk().clipboard_get()函数，获取服务端剪切板的内容，并在 flask 的 return render_template 时将剪切板内容作为一个参数 clip，在服务页面中通过访问 clip 显示其内容。

（二）关键代码

1.服务端的 GUI

```
if __name__ == '__main__':
    root = Tk()
    winWidth = 700
    winHeight = 500

    # 获取屏幕分辨率
    screenWidth = root.winfo_screenwidth()
    screenHeight = root.winfo_screenheight()
    x = int((screenWidth - winWidth) / 2)
    y = int((screenHeight - winHeight) / 2)
    # 设置主窗口标题
```

```
    root.title("局域网传输文件")
    # 设置窗口初始位置在屏幕居中
    root.geometry("%sx%s+%s+%s" % (winWidth, winHeight, x, y))
    # 设置窗口图标
    root.iconbitmap("./img/logo.ico")
    # 设置窗口宽高固定
    root.resizable(0, 0)
    # 添加菜单栏
    f = tkinter.Menu(root)
    root['menu'] = f

    f.add_command(label='源码', command=source)
    # 增加背景图片
    photo = tk.PhotoImage(file="./img/bg.png")
    theLabel = tk.Label(root, text="", justify=tk.LEFT, image=photo, compound=tk.CENTER)
    #theLabel.place(relx=0.8, rely=0.63, anchor=CENTER)
    theLabel.place(relx=0.5, rely=0.5, anchor=CENTER)

    count = 0
    chiose = 0
    ip_list, len_list = getIP()
    Button(root, text='开启服务', command=start).place(relx=0.9, rely=0.05, anchor=CENTER)
    Button(root, text='更换网址', command=show_qrc).place(relx=0.9, rely=0.15, anchor=CENTER)
    Button(root, text='文件所在', command=open_dir).place(relx=0.9, rely=0.25, anchor=CENTER)
    Button(root, text='关闭服务', command=delete_dir).place(relx=0.9, rely=0.35,anchor=CENTER)
#删除 tmpfiles 里的文件


    l = Label(root)
    l.place(relx=0.3, rely=0.5, anchor=CENTER) #二维码位置

    l1 = Label(root)
    l1.place(relx=0.3, rely=0.1, anchor=CENTER) #地址显示位置

    tips = Label(root)
    tips.place(relx=0.3, rely=0.9, anchor=CENTER) #提示信息位置


    root.mainloop()
```

2.上传文件

```
@app.route('/up_video', methods=['post'])
def up_video():
    try:
        filepath = os.path.join(os.path.dirname(os.path.abspath('__file__')), 'tmp_file')
        if not os.path.exists(filepath):
            os.makedirs(filepath)
        for f in request.files.getlist('file'):
            if f and '/' in f.filename:
                # print('这是文件夹')
                temp_path = filepath + os.sep + f.filename.split('/')[0]
                if not os.path.exists(temp_path):
                    os.makedirs(temp_path)
                filename = f.filename.split('/')[1]
                upload_path = os.path.join(temp_path, filename)
                f.save(upload_path)
            elif f:
                # print('这是多文件')
                filename = f.filename
                upload_path = os.path.join(filepath, filename)
                f.save(upload_path)
            else:
                continue
        return render_template('up_video_ok.html')
    except Exception as e:
        print(e)
        return json.dumps({'code': "502"}, ensure_ascii=False)
```

3.下载文件

```
@app.route('/tmp_file/<file_name>', methods=['GET'])
def tmp_file(file_name):
    try:
        filepath = os.path.join(os.path.dirname(os.path.abspath('__file__')), 'tmp_file')
        if not os.path.exists(filepath):
            os.makedirs(filepath)
        filename = file_name
        # #print(file_name)
        file = os.path.join(filepath, filename)
        return send_file(file)
    except Exception as e:
        return json.dumps({'code': "502"}, ensure_ascii=False)
```

4.剪切板共享

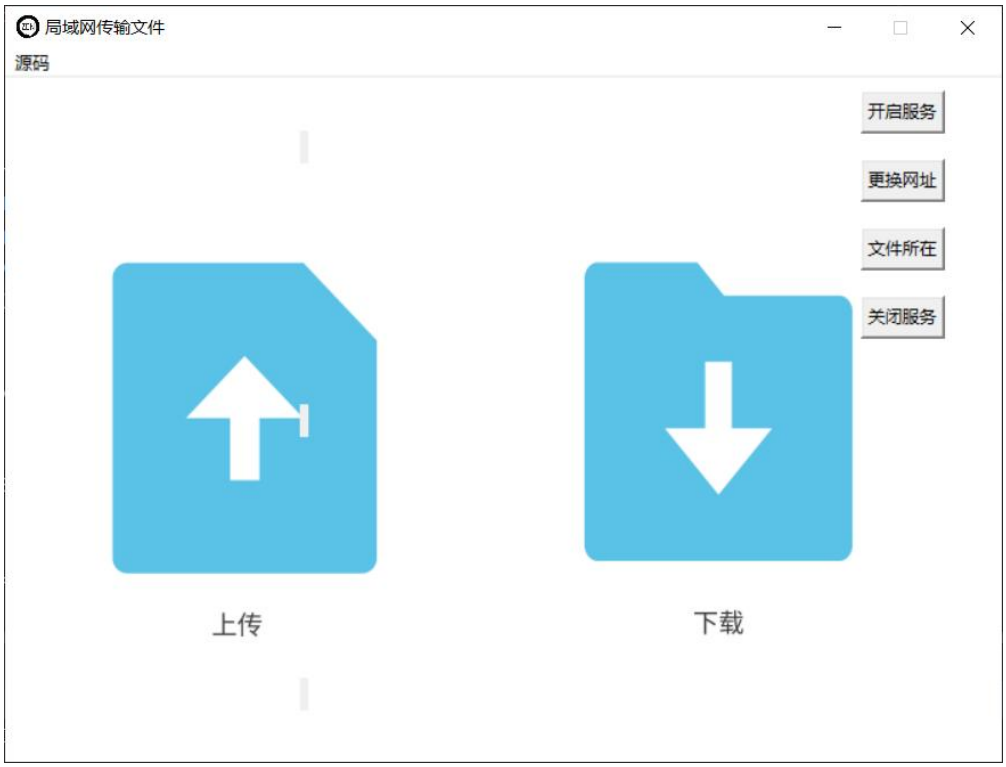
```
try:
    clip = Tk().clipboard_get()
except BaseException:
    clip=''
#如果没有复制内容，会出错，所以如果没有复制内容就设为剪切板内容为''
```

```
return render_template('up_video.html', clip=clip)
```

三、单元测试

四、使用教程

1.在服务端，运行 se2.py，运行成功就会弹出如下窗口。



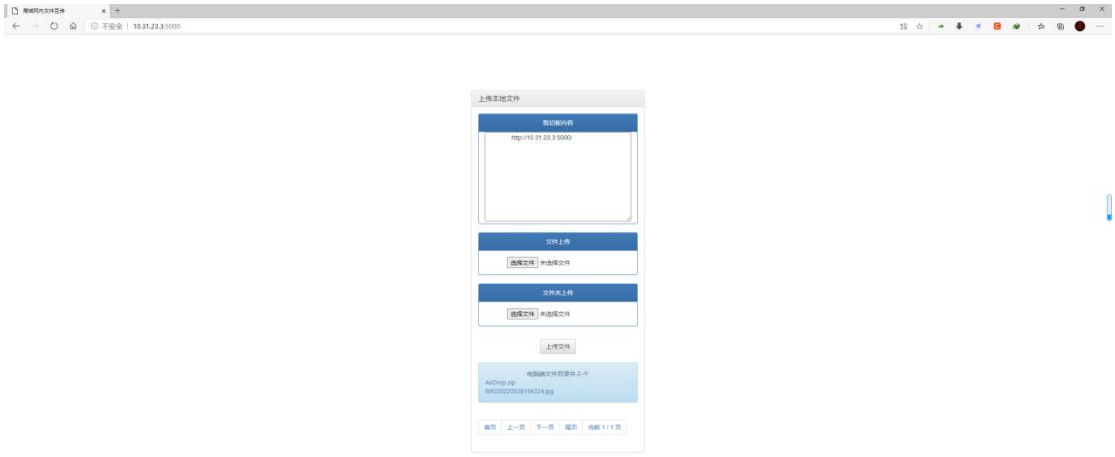
2.点击“开启服务”，窗口就会出现一个网址和二维码，与服务端处于同一局域网下的设备都可以通过访问网址进入服务页面。二维码可以方便移动端设备快速访问。

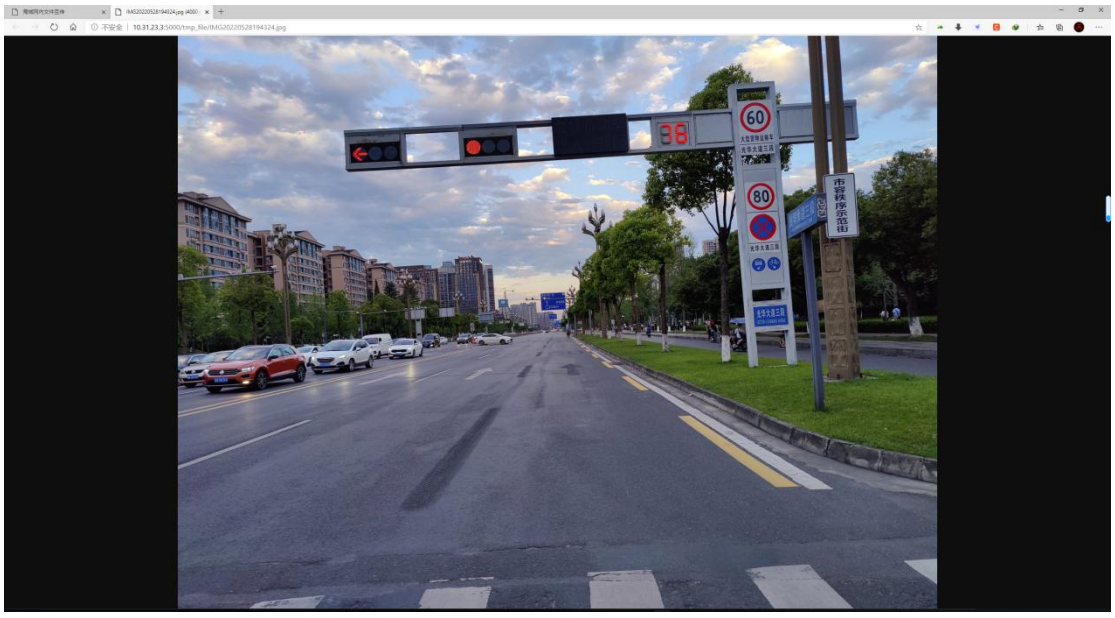
3.这里我们用手机访问该网址，可以看到已共享的文件有 AirDrop.zip，这是先前在服务端选择上传的文件。

点击即可下载

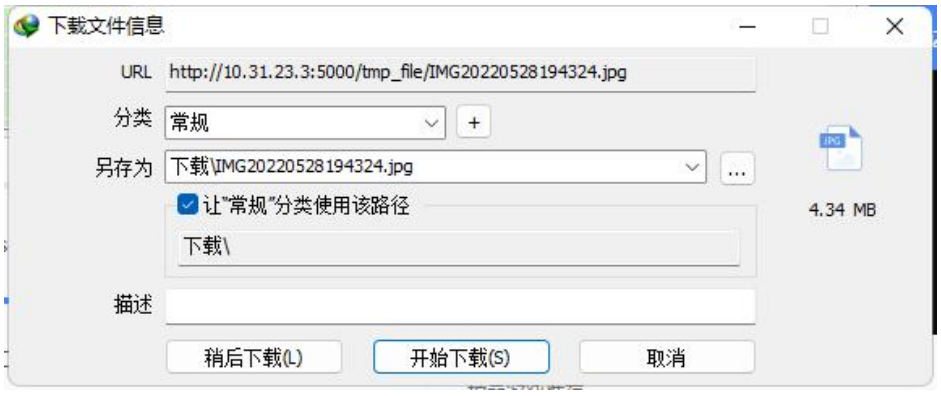
4.同样我们也可以在手机端上传一张照片测试

5.我们此时在 PC 端访问服务页面，可以看到那张照片

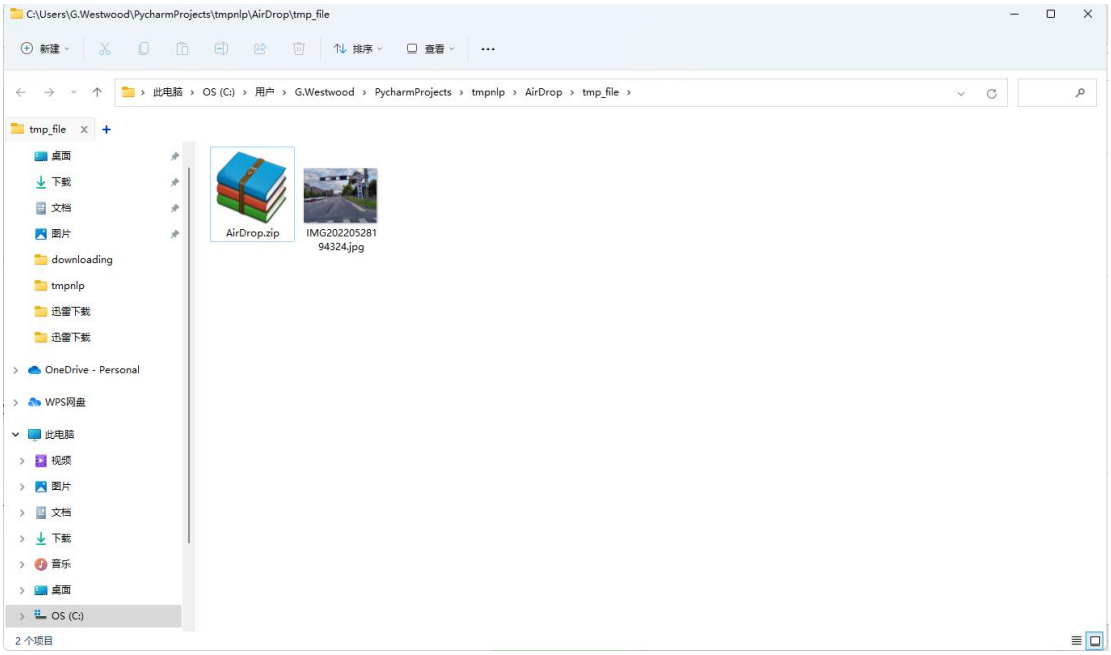




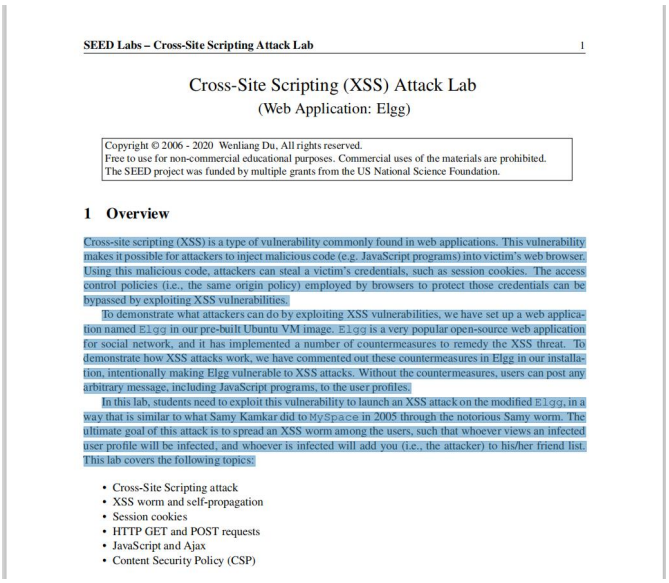
也可以选择复制其链接选择下载



6.在服务端，我们点击“文件所在”，即可查看 tmp_file 文件夹，查看所有上传的文件。



7.我们在服务端复制一段文字



SEED Labs – Cross-Site Scripting Attack Lab 1

Cross-Site Scripting (XSS) Attack Lab
(Web Application: Elgg)

1 Overview

Cross-site scripting (XSS) is a type of vulnerability commonly found in web applications. This vulnerability makes it possible for attackers to inject malicious code (e.g. JavaScript programs) into victim's web browser. Using this malicious code, attackers can steal a victim's credentials, such as session cookies. The access control policies (i.e., the same origin policy) employed by browsers to protect those credentials can be bypassed by exploiting XSS vulnerabilities.

To demonstrate what attackers can do by exploiting XSS vulnerabilities, we have set up a web application named `Elgg` in our pre-built Ubuntu VM image. `Elgg` is a very popular open-source web application for social network, and it has implemented a number of countermeasures to remedy the XSS threat. To demonstrate how XSS attacks work, we have commented out these countermeasures in Elgg in our installation, intentionally making Elgg vulnerable to XSS attacks. Without the countermeasures, users can post any arbitrary message, including JavaScript programs, to the user profiles.

In this lab, students need to exploit this vulnerability to launch an XSS attack on the modified `Elgg`, in a way that is similar to what Samy Kamkar did to `MySpace` in 2005 through the notorious Samy worm. The ultimate goal of this attack is to spread an XSS worm among the users, such that whoever views an infected user profile will be infected, and whoever is infected will add you (i.e., the attacker) to his/her friend list. This lab covers the following topics:

- Cross-Site Scripting attack
- XSS worm and self-propagation
- Session cookies
- HTTP GET and POST requests
- JavaScript and Ajax
- Content Security Policy (CSP)

在其它设备上即可查看