

[illegible]

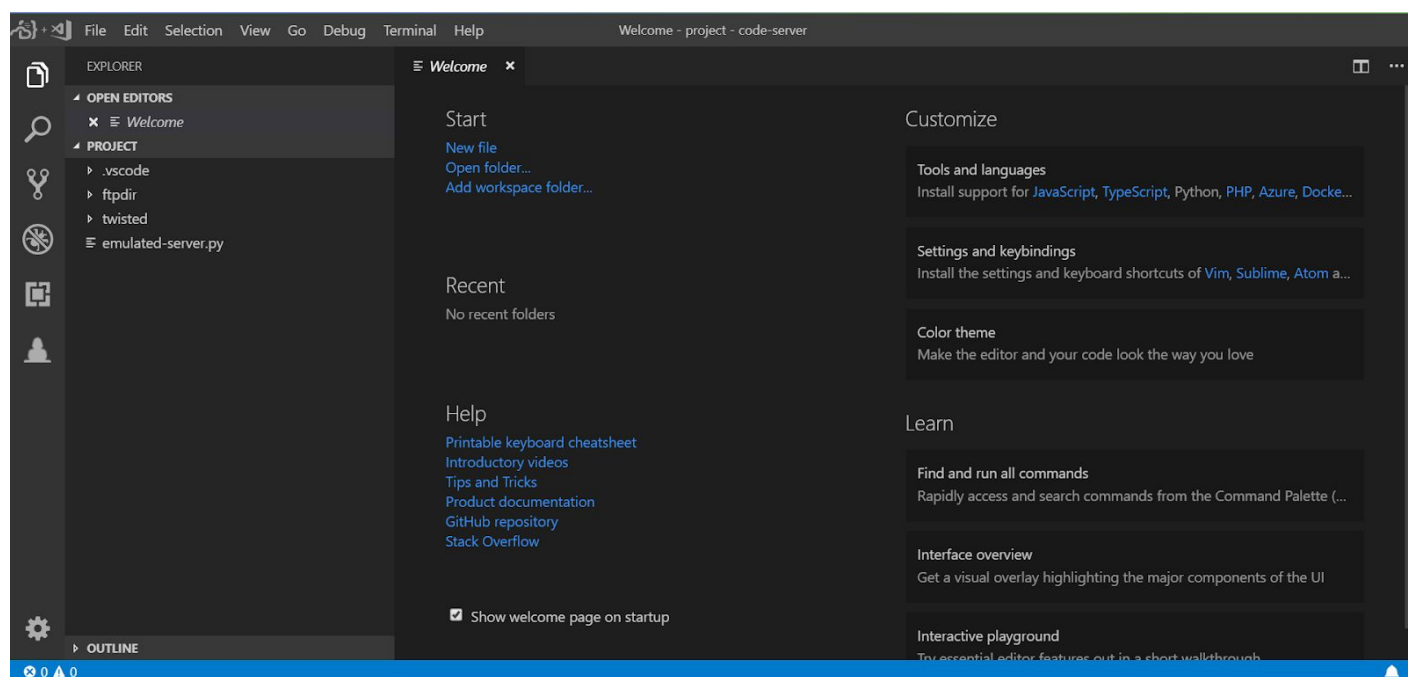
<b>Name</b>	FTP Server Emulation
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=1214">https://attackdefense.com/challengedetails?cid=1214</a>
<b>Type</b>	Offensive Python : Server Emulation

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

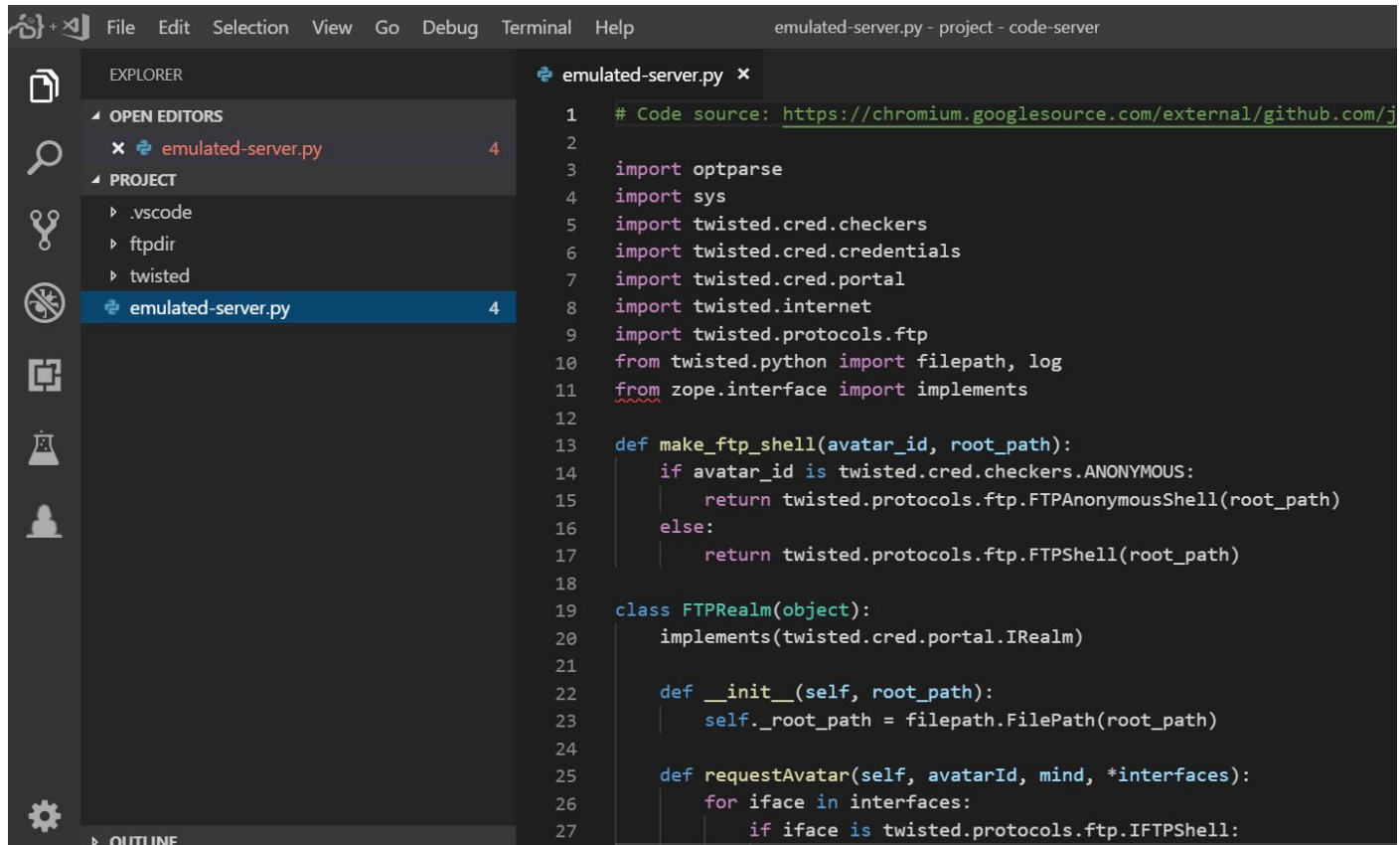
**Objective:** Modify the code, launch the server and use Kali Linux to interact/attack it.

**Solution:**

**Landing Page:**



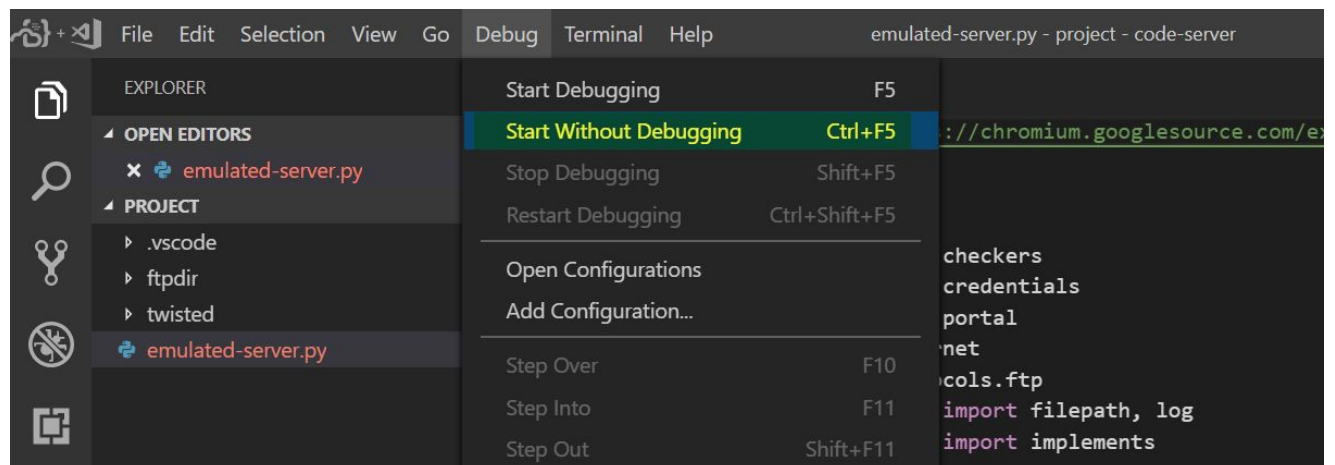
**Step 1:** Select “emulated-server.py” from the Project Explorer.



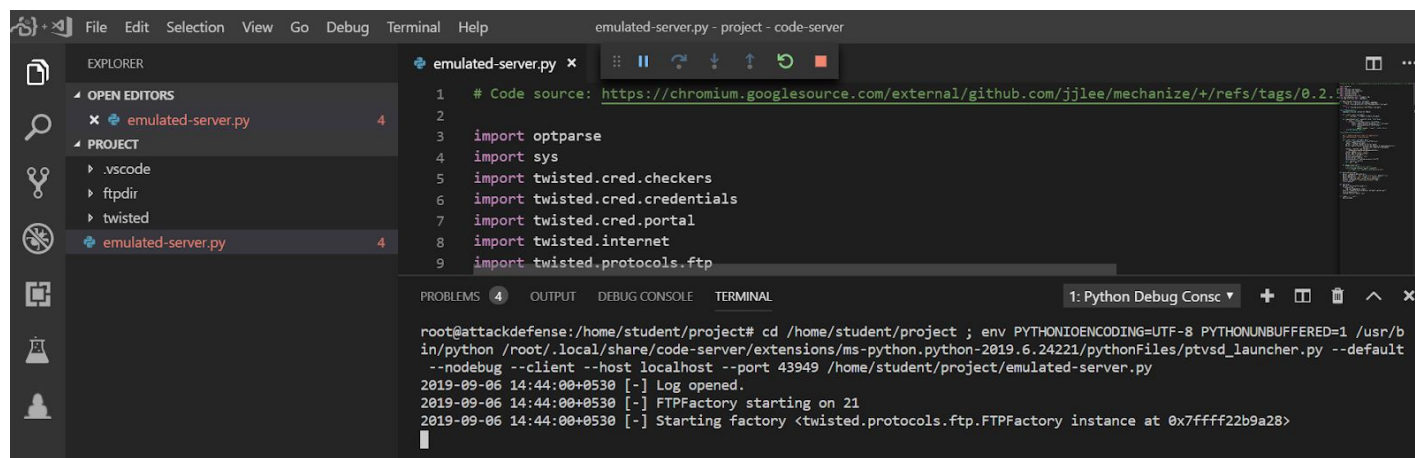
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left is open, showing the 'PROJECT' section with a tree view containing '.vscode', 'ftpdirc', 'twisted', and 'emulated-server.py'. The 'emulated-server.py' file is selected and highlighted in blue. The main editor area displays the code for 'emulated-server.py'. The code includes imports for 'optparse', 'sys', 'twisted.cred.checkers', 'twisted.cred.credentials', 'twisted.cred.portal', 'twisted.internet', and 'twisted.protocols.ftp'. It also shows a function 'make\_ftp\_shell' and a class 'FTPRealm' that implements 'twisted.cred.portal.IRealm'.

```
1 # Code source: https://chromium.googlesource.com/external/github.com/j
2
3 import optparse
4 import sys
5 import twisted.cred.checkers
6 import twisted.cred.credentials
7 import twisted.cred.portal
8 import twisted.internet
9 import twisted.protocols.ftp
10 from twisted.python import filepath, log
11 from zope.interface import implements
12
13 def make_ftp_shell(avatar_id, root_path):
14     if avatar_id is twisted.cred.checkers.ANONYMOUS:
15         return twisted.protocols.ftp.FTPAnonymousShell(root_path)
16     else:
17         return twisted.protocols.ftp.FTPShell(root_path)
18
19 class FTPRealm(object):
20     implements(twisted.cred.portal.IRealm)
21
22     def __init__(self, root_path):
23         self._root_path = filepath.FilePath(root_path)
24
25     def requestAvatar(self, avatarId, mind, *interfaces):
26         for iface in interfaces:
27             if iface is twisted.protocols.ftp.IFTPShell:
```

**Step 2:** Navigate to Debug Menu and click on “Start Without Debugging option” to run the program.



The python script will run and start an FTP Server on port 21.

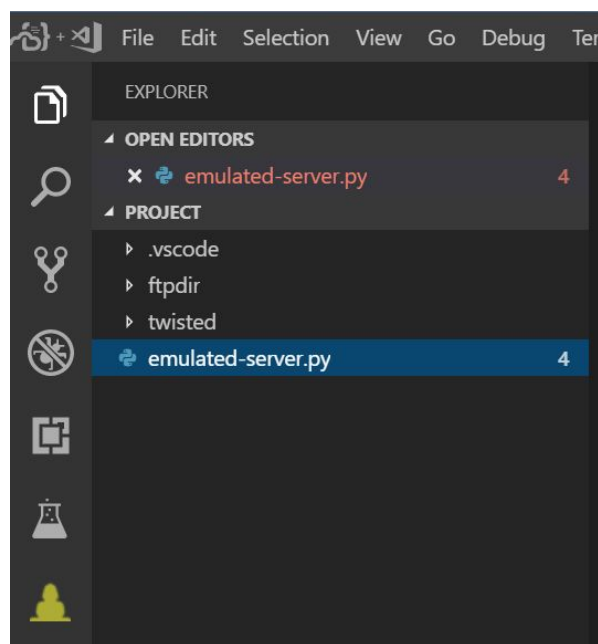


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the project structure with folders .vscode, ftpdir, and twisted, and a file emulated-server.py. The main editor displays the contents of emulated-server.py, which is a Python script using mechanize, optparse, and twisted to start an FTP server. The Terminal panel at the bottom shows the command prompt output, indicating the script is running and the FTP factory is starting on port 21.

```
1 # Code source: https://chromium.googlesource.com/external/github.com/jjlee/mechanize/+/refs/tags/0.2.3
2
3 import optparse
4 import sys
5 import twisted.cred.checkers
6 import twisted.cred.credentials
7 import twisted.cred.portal
8 import twisted.internet
9 import twisted.protocols.ftp
```

```
root@attackdefense:/home/student/project# cd /home/student/project ; env PYTHONIOENCODING=UTF-8 PYTHONUNBUFFERED=1 /usr/bin/python /root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/pythonFiles/ptvsd_launcher.py --default --nodebug --client --host localhost --port 43949 /home/student/project/emulated-server.py
2019-09-06 14:44:00+0530 [-] Log opened.
2019-09-06 14:44:00+0530 [-] FTPFactory starting on 21
2019-09-06 14:44:00+0530 [-] Starting factory <twisted.protocols.ftp.FTPFactory instance at 0x7ffff22b9a28>
```

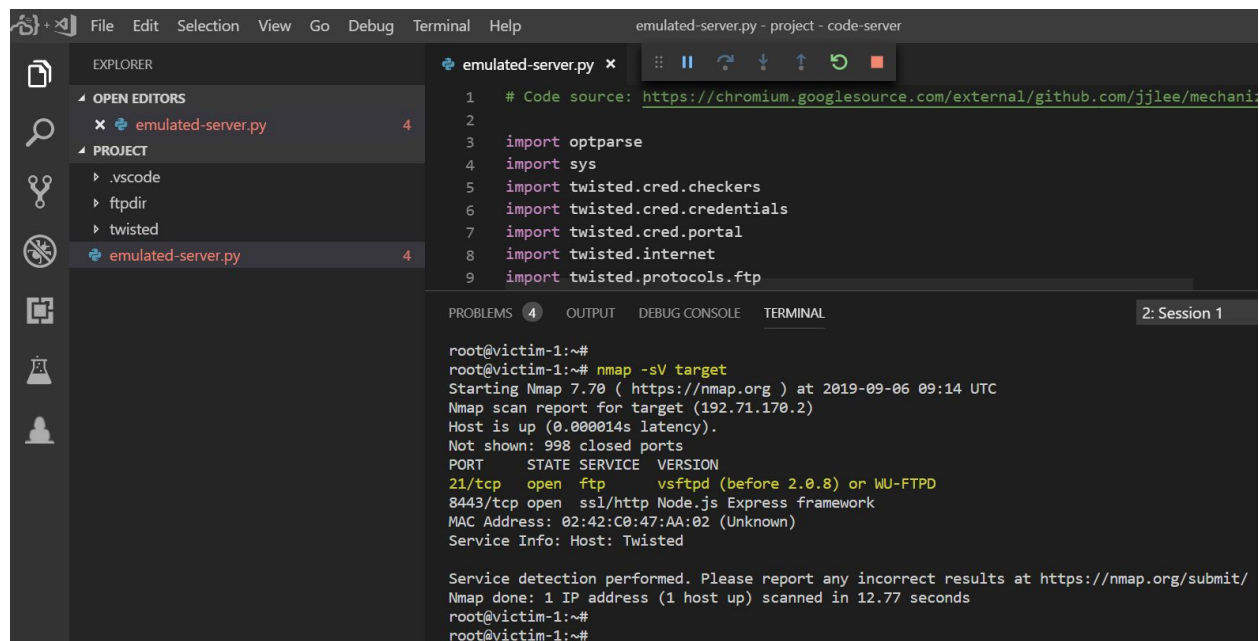
**Step 3:** Click on the “Sessions” icon on the activity bar to gain access to Kali machine.



This will spawn a new Terminal “Session 1” which will provide a bash shell on a remote Kali machine.

**Step 4:** Perform Nmap scan from the Kali machine. Identify the services running on the machine on which the IDE (and code) is running. The IDE machine is mapped to “target” hostname. So, “target” can be used while launching scans or tools on this machine. Alternatively, the IP address of the both machines can be found by running “ip addr” command on respective machine. The IDE machine should be on 192.x.y.2 and Kali machine should be on 192.x.y.3.

**Command:** nmap -sV target



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project with files like .vscode, ftpdir, twisted, and emulated-server.py. The main editor area shows the code for emulated-server.py, which imports optparse, sys, and various modules from the twisted library. The Terminal panel at the bottom shows the output of an Nmap scan performed on the target machine. The scan results indicate that port 21/tcp is open and running vsftpd (before 2.0.8) or WU-FTPD. Other open ports include 8443/tcp (Node.js Express framework) and 443/tcp (Node.js Express framework). The scan was completed in 12.77 seconds.

```
root@victim-1:~#  
root@victim-1:~# nmap -sV target  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-06 09:14 UTC  
Nmap scan report for target (192.71.170.2)  
Host is up (0.000014s latency).  
Not shown: 998 closed ports  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd (before 2.0.8) or WU-FTPD  
8443/tcp  open  ssl/http Node.js Express framework  
MAC Address: 02:42:C0:47:AA:02 (Unknown)  
Service Info: Host: Twisted  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/  
Nmap done: 1 IP address (1 host up) scanned in 12.77 seconds  
root@victim-1:~#  
root@victim-1:~#
```

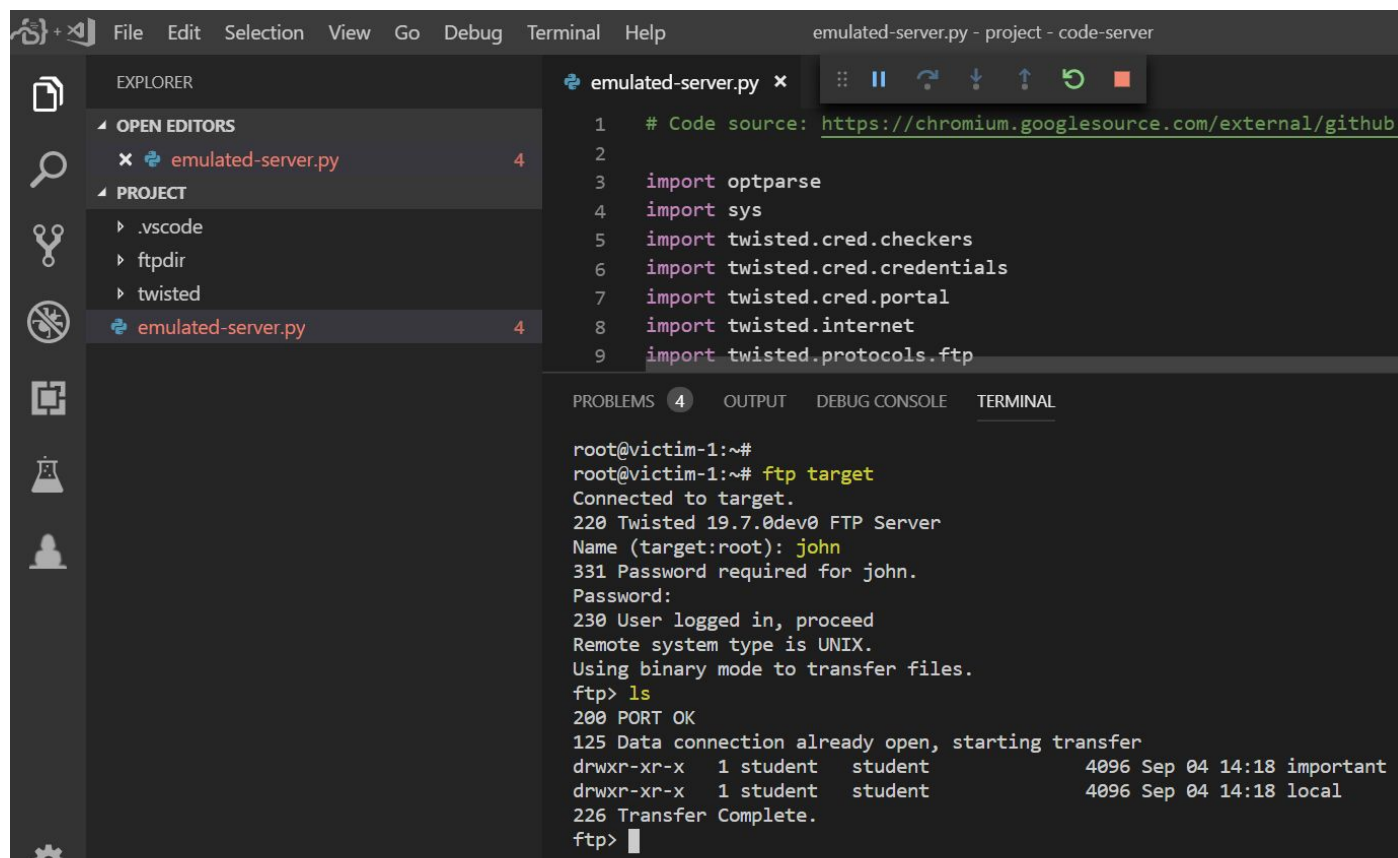
FTP server is running on port 21 on the target machine.

**Step 5:** Login to FTP server using username “john” and password “john”. Alternatively, as anonymous login is enabled on the server, one can use that.

**Command:** ftp target

After successful authentication, an FTP shell will be provided to the user.





The screenshot shows the Visual Studio Code editor with a dark theme. The Explorer sidebar on the left shows the project structure with files like .vscode, ftpdir, twisted, and emulated-server.py. The main editor window displays the code for emulated-server.py, which imports modules like optparse, sys, and twisted. Below the editor, the TERMINAL tab is active, showing a command prompt session where an FTP connection is established to a target system. The terminal output shows the user logging in as 'john' and listing files in the current directory.

```
emulated-server.py x
1 # Code source: https://chromium.googlesource.com/external/github.
2
3 import optparse
4 import sys
5 import twisted.cred.checkers
6 import twisted.cred.credentials
7 import twisted.cred.portal
8 import twisted.internet
9 import twisted.protocols.ftp

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

root@victim-1:~#
root@victim-1:~# ftp target
Connected to target.
220 Twisted 19.7.0dev0 FTP Server
Name (target:root): john
331 Password required for john.
Password:
230 User logged in, proceed
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT OK
125 Data connection already open, starting transfer
drwxr-xr-x  1 student  student      4096 Sep 04 14:18 important
drwxr-xr-x  1 student  student      4096 Sep 04 14:18 local
226 Transfer Complete.
ftp> |
```

## References:

1. Visual Studio Code (<https://code.visualstudio.com/>)
2. VS Code Basic Editing (<https://code.visualstudio.com/docs/editor/codebasics>)
3. Twisted (<https://www.twistedmatrix.com/trac/>)