ATTACK DEFENSE

by PentesterAcademy

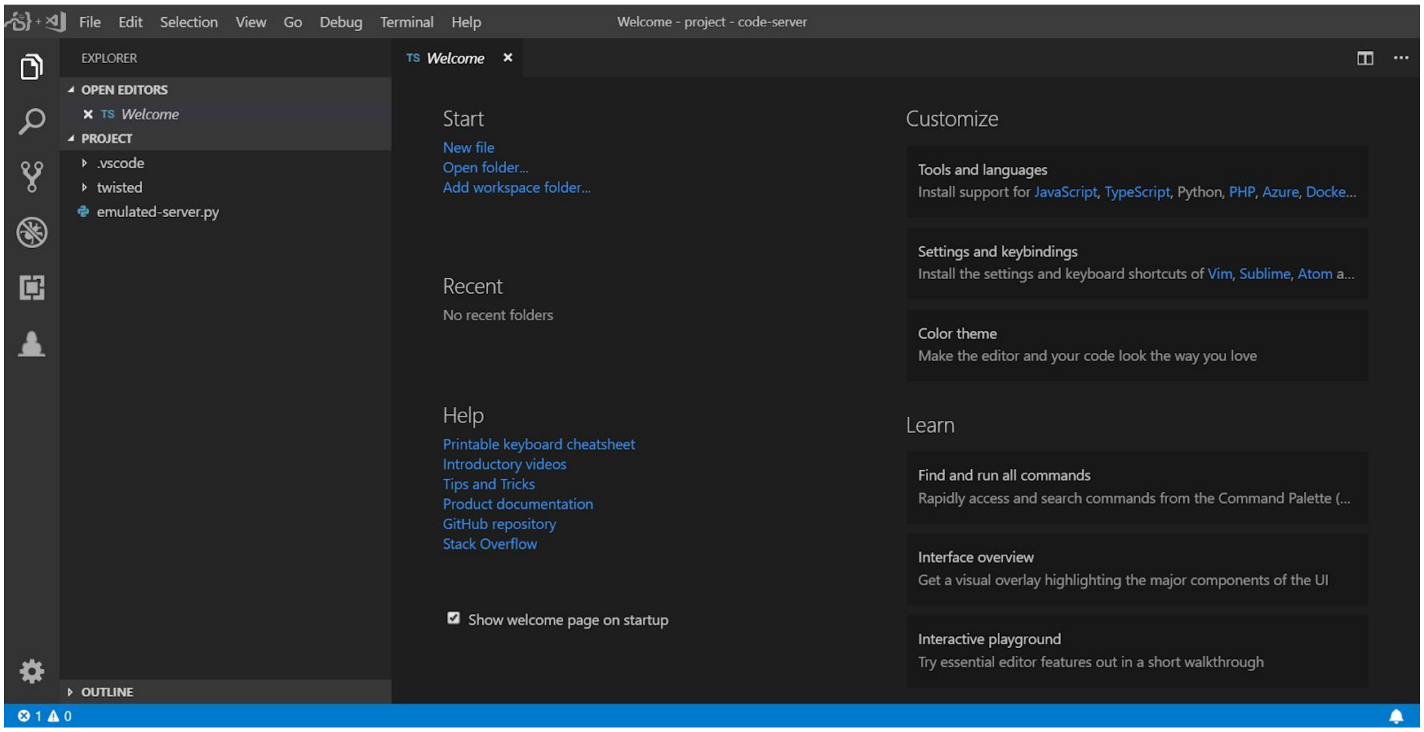
Name	Server Emulation Lab Warmup
URL	https://attackdefense.com/challengedetails?cid=1212
Type	Offensive Python : Server Emulation

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.

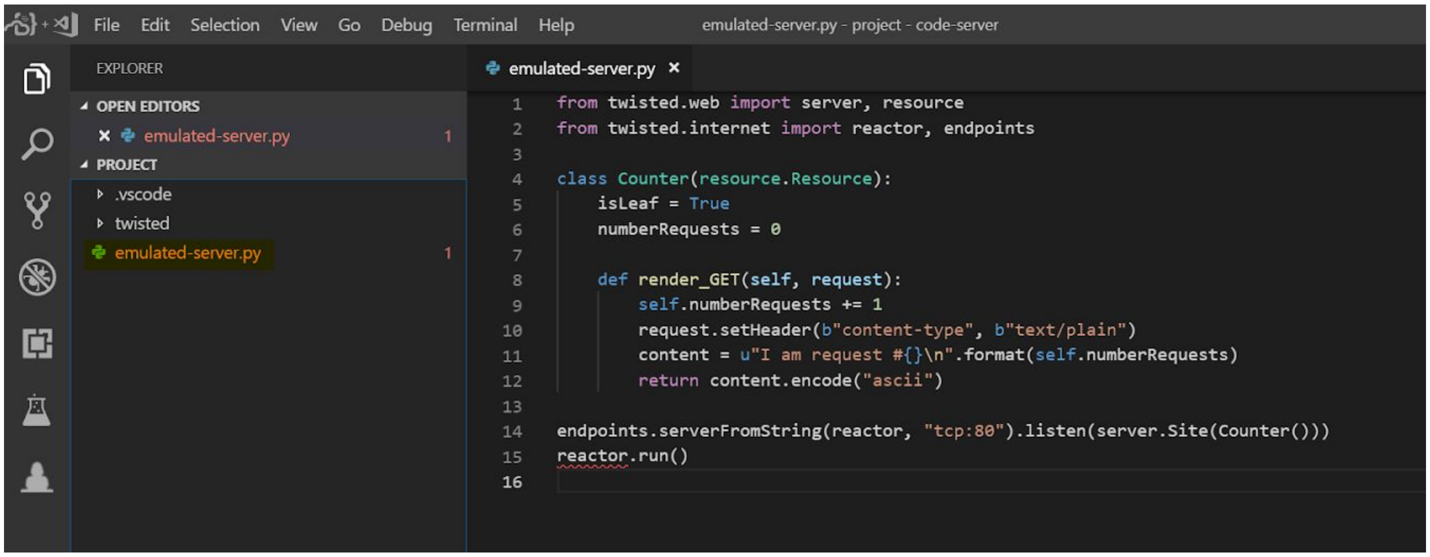
Objective: Modify the code, launch the server and use Kali Linux to interact/attack it.

Solution:

Landing Page:


File Edit Selection View Go Debug Terminal Help
Welcome - project - code-server
EXPLORER
OPEN EDITORS
Welcome
PROJECT
.vscode
twisted
emulated-server.py
OUTLINE
Start
New file
Open folder...
Add workspace folder...
Recent
No recent folders
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Show welcome page on startup
Customize
Tools and languages
Install support for JavaScript, TypeScript, Python, PHP, Azure, Docke...
Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Sublime, Atom a...
Color theme
Make the editor and your code look the way you love
Learn
Find and run all commands
Rapidly access and search commands from the Command Palette (...
Interface overview
Get a visual overlay highlighting the major components of the UI
Interactive playground
Try essential editor features out in a short walkthrough

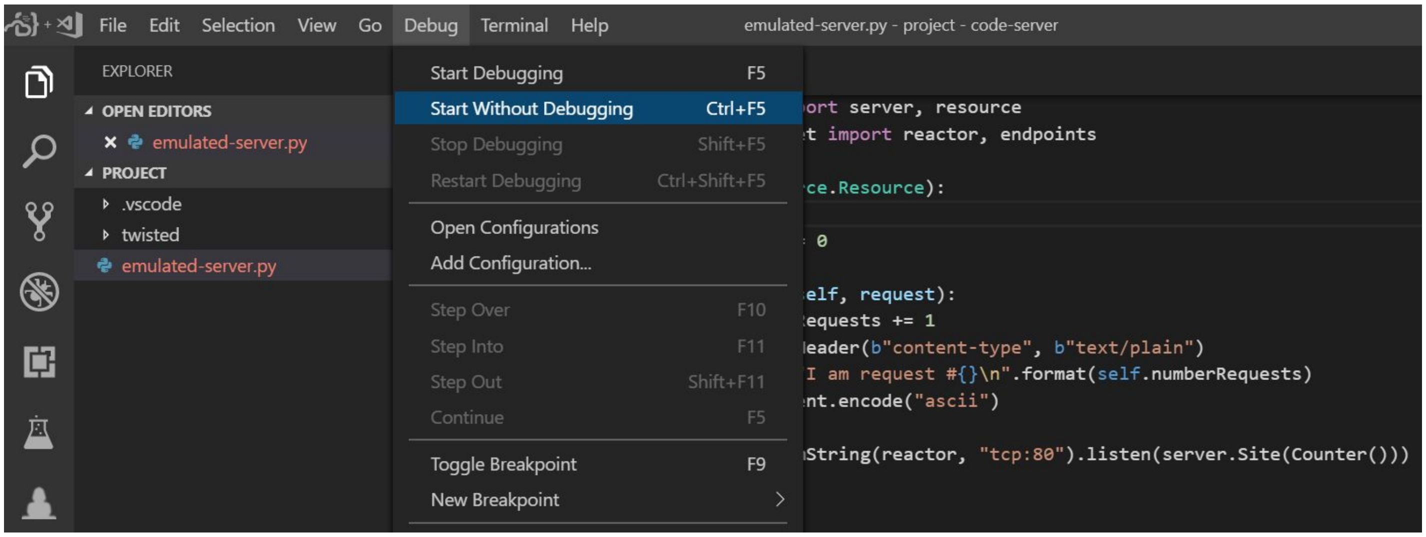
Step 1: Select “emulated-server.py” from the Project Explorer.


File Edit Selection View Go Debug Terminal Help
emulated-server.py - project - code-server
EXPLORER
OPEN EDITORS
emulated-server.py
PROJECT
.vscode
twisted
emulated-server.py

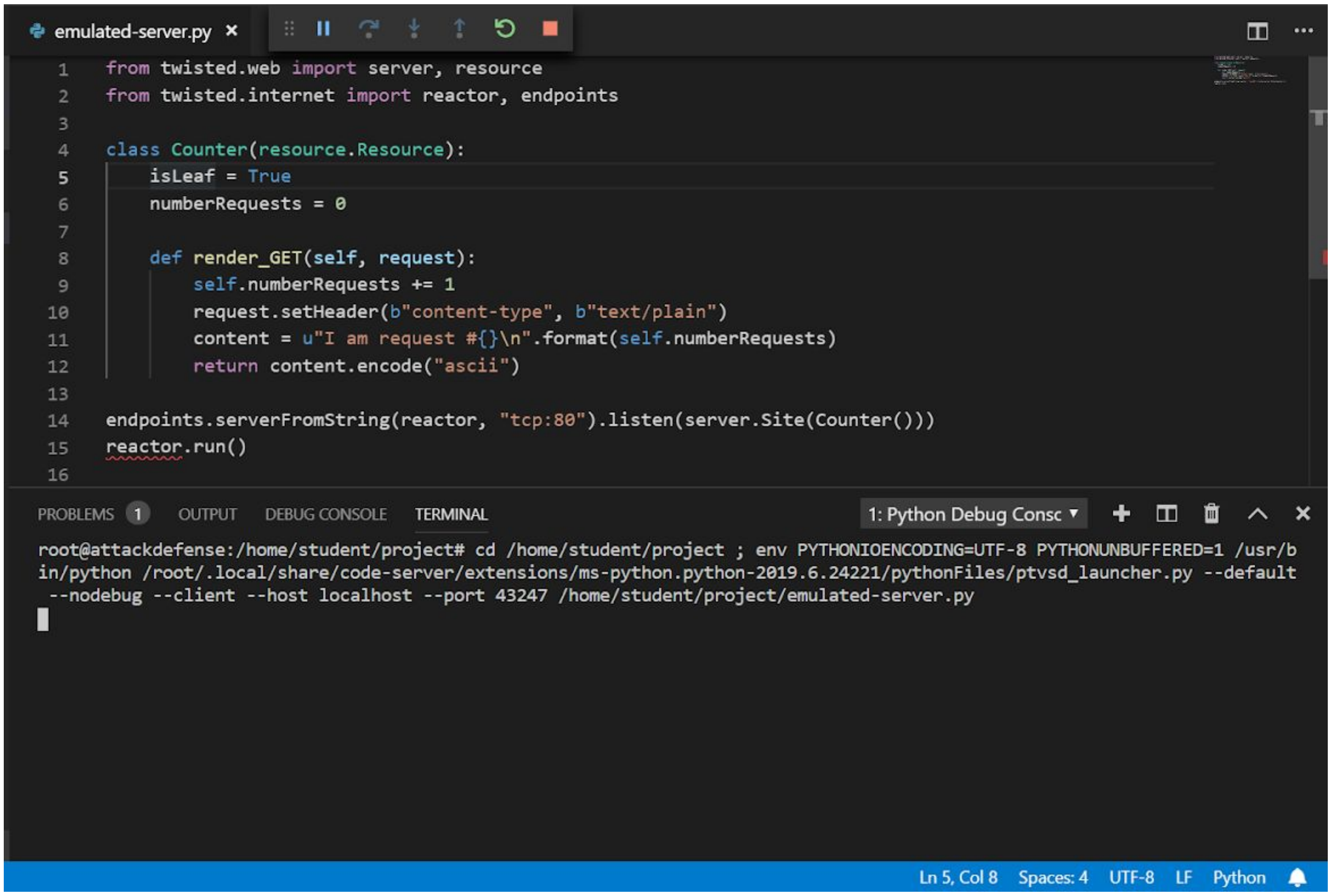
```
from twisted.web import server, resource
from twisted.internet import reactor, endpoints

class Counter(resource.Resource):
    isLeaf = True
    numberRequests = 0

    def render_GET(self, request):
        self.numberRequests += 1
        request.setHeader(b"content-type", b"text/plain")
        content = u"I am request #{}\n".format(self.numberRequests)
        return content.encode("ascii")

endpoints.serverFromString(reactor, "tcp:80").listen(server.Site(Counter()))
reactor.run()
```

The python script starts an HTTP Server on port 80. The server will process HTTP GET request and respond with the content “I am request #” followed by the number of requests made to the web server.

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



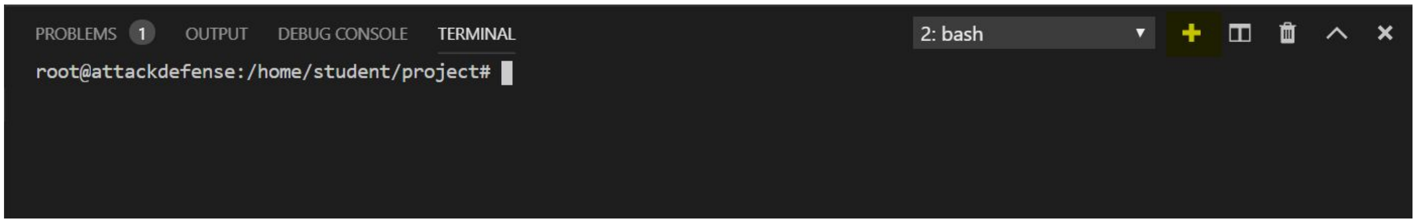
The python script will be executed and the output will be displayed on the integrated terminal.



```python
from twisted.web import server, resource
from twisted.internet import reactor, endpoints

class Counter(resource.Resource):
    isLeaf = True
    numberRequests = 0

    def render_GET(self, request):
        self.numberRequests += 1
        request.setHeader(b"content-type", b"text/plain")
        content = u"I am request #{}\n".format(self.numberRequests)
        return content.encode("ascii")

endpoints.serverFromString(reactor, "tcp:80").listen(server.Site(Counter()))
reactor.run()
```

```
root@attackdefense:/home/student/project# cd /home/student/project ; env PYTHONIOENCODING=UTF-8 PYTHONUNBUFFERED=1 /usr/b
in/python /root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/pythonFiles/ptvsd_launcher.py --default
 --nodebug --client --host localhost --port 43247 /home/student/project/emulated-server.py
```

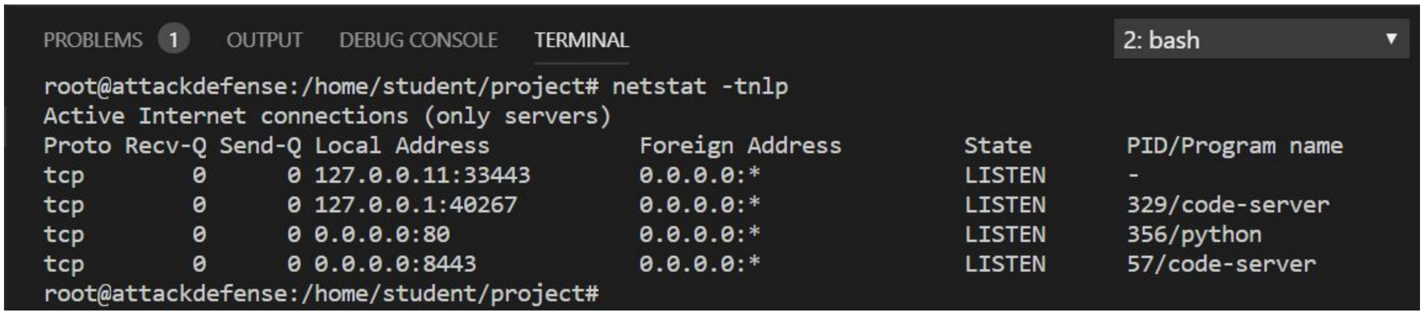
Step 3: Click on the "New Terminal" button to spawn a bash shell on the machine.



```
root@attackdefense:/home/student/project#
```

Step 4: List the open TCP ports on the machine. Check whether port 80 is open.

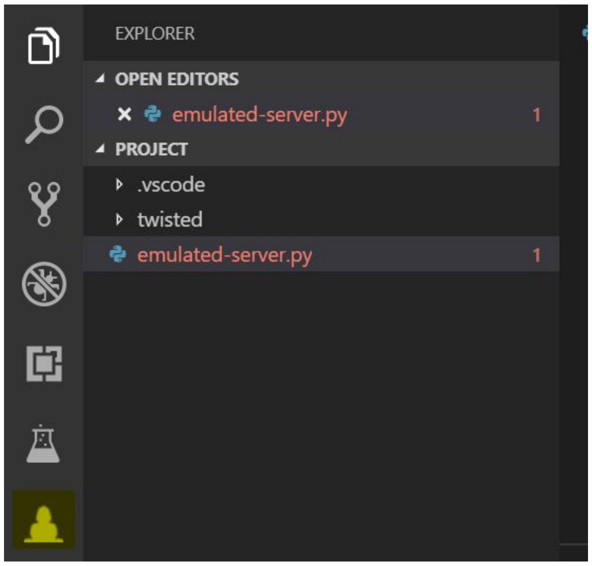
Command: netstat -tnlp



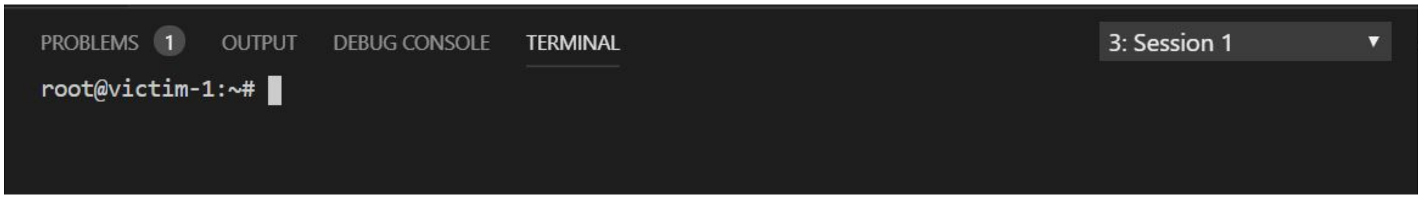
```
root@attackdefense:/home/student/project# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.11:33443        0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:40267         0.0.0.0:*               LISTEN      329/code-server
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      356/python
tcp        0      0 0.0.0.0:8443            0.0.0.0:*               LISTEN      57/code-server
root@attackdefense:/home/student/project#
```

The python program is listening on port 80.

Step 5: Click on the “Sessions” icon on the activity bar to gain access to Kali machine.



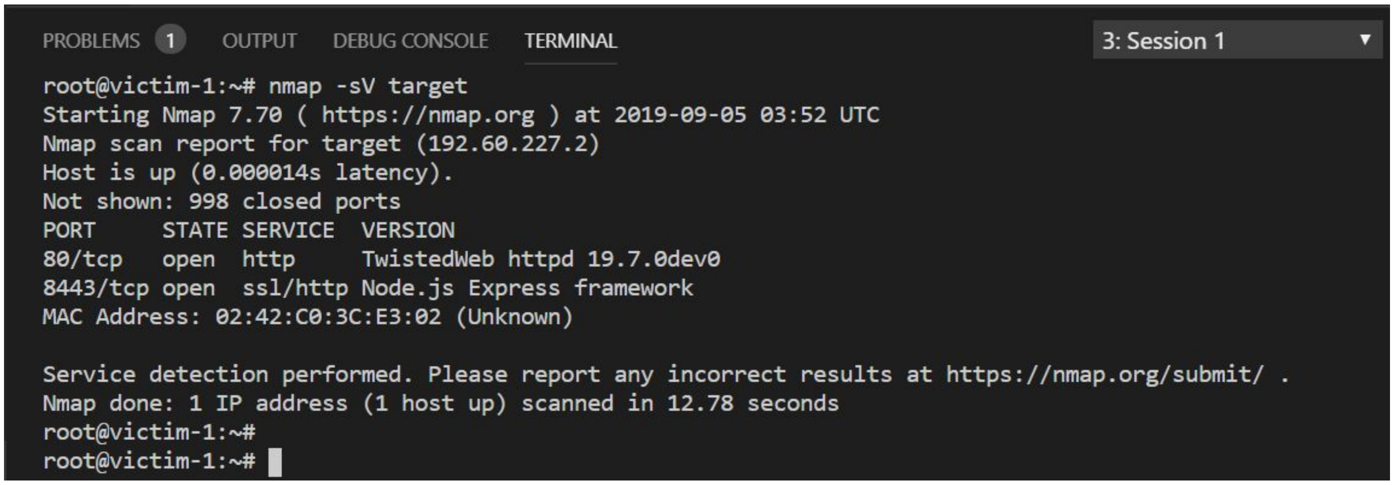
This will spawn a new Terminal “Session 1” which will provide a bash shell on a remote Kali machine.



```
root@victim-1:~#
```

Step 6: Perform Nmap scan from the Kali machine. Identify the services running on the machine on which the IDE (and code) is running. This 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

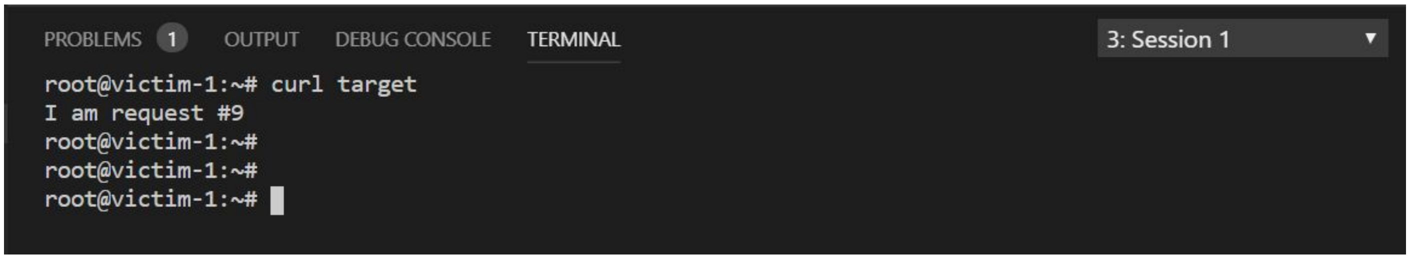


```
root@victim-1:~# nmap -sV target
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-05 03:52 UTC
Nmap scan report for target (192.60.227.2)
Host is up (0.000014s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE  VERSION
80/tcp   open  http     TwistedWeb httpd 19.7.0dev0
8443/tcp open  ssl/http Node.js Express framework
MAC Address: 02:42:C0:3C:E3:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.78 seconds
root@victim-1:~#
root@victim-1:~#
```

TwistedWeb Httpd server is running on port 80 on the target machine.

Step 7: Send an HTTP GET request to the web server and check the received response.

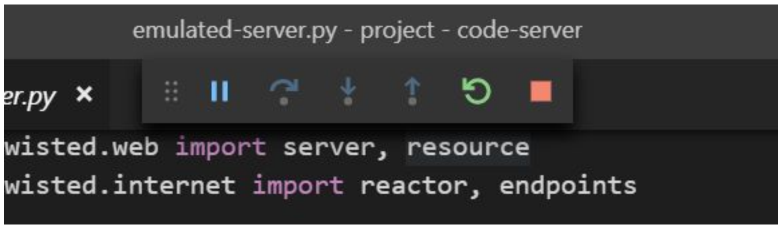
Command: curl target



```
root@victim-1:~# curl target
I am request #9
root@victim-1:~#
root@victim-1:~#
root@victim-1:~#
```

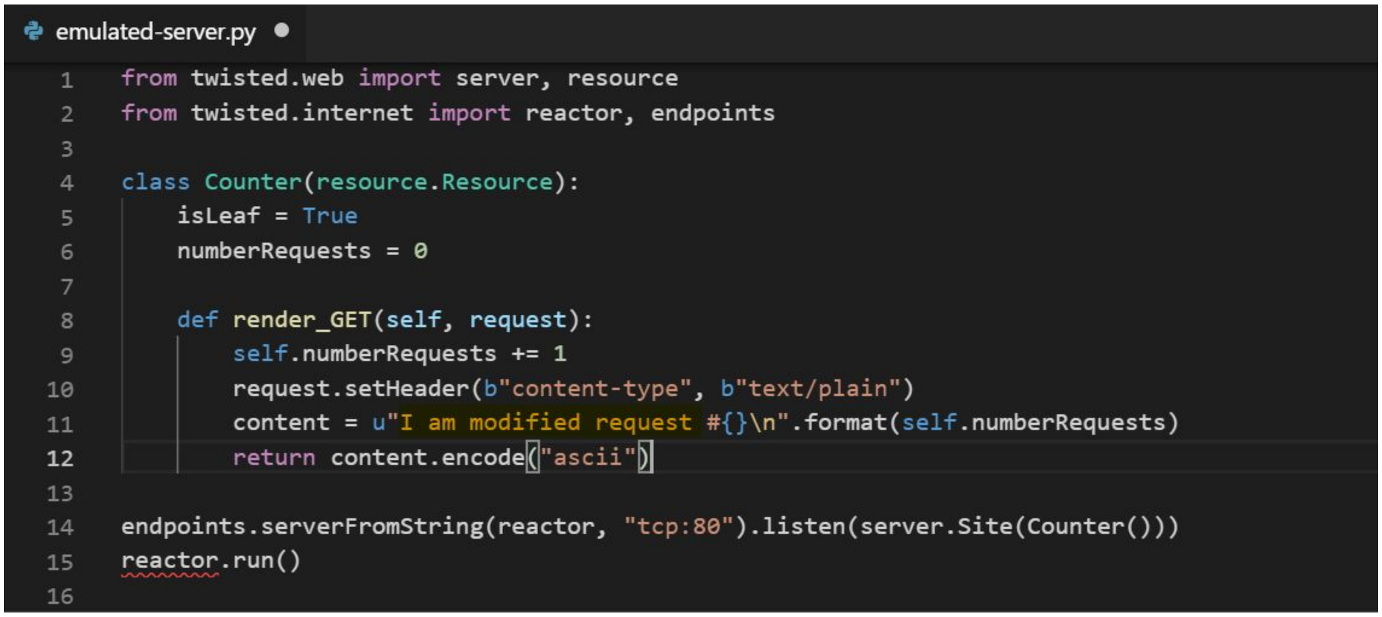
The received response was “I am request #”, followed by the number of requests made to the web server.

Step 8: Stop the python program by selecting "Stop" debug action.



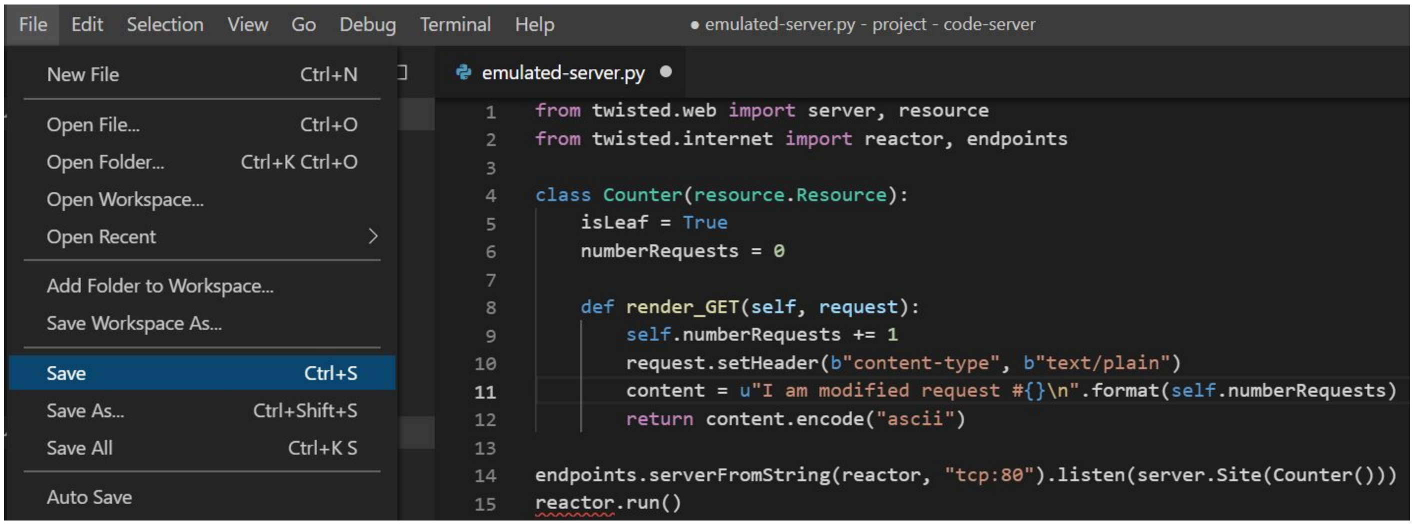
```
wisted.web import server, resource
wisted.internet import reactor, endpoints
```

Step 9: Modify the program and update the content to be sent as response from "I am request" to "I am modified request".



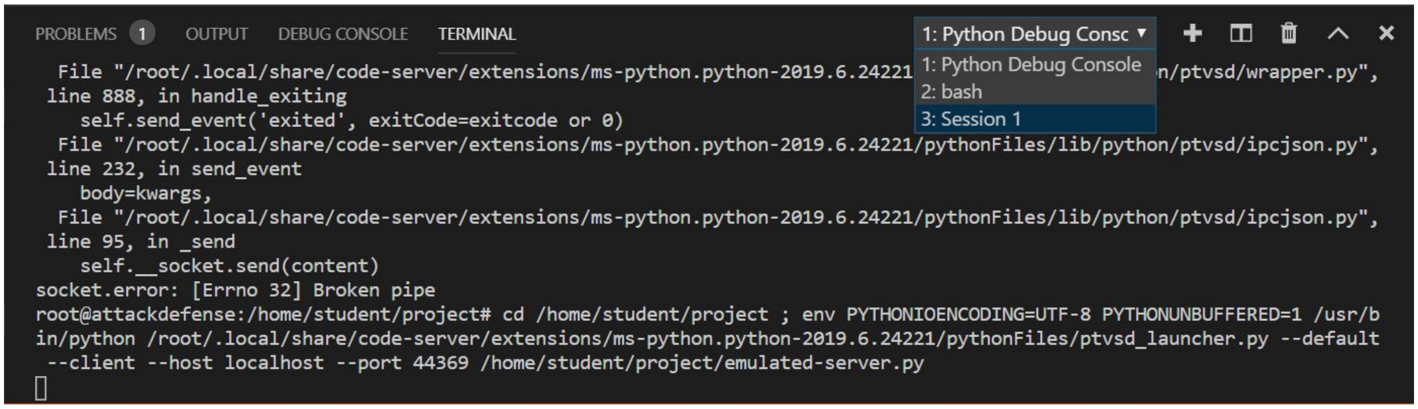
```python
from twisted.web import server, resource
from twisted.internet import reactor, endpoints

class Counter(resource.Resource):
    isLeaf = True
    numberRequests = 0

    def render_GET(self, request):
        self.numberRequests += 1
        request.setHeader(b"content-type", b"text/plain")
        content = u"I am modified request #{}\n".format(self.numberRequests)
        return content.encode("ascii")

endpoints.serverFromString(reactor, "tcp:80").listen(server.Site(Counter()))
reactor.run()

```

Step 10: Navigate to File and click on "Save" to save the file



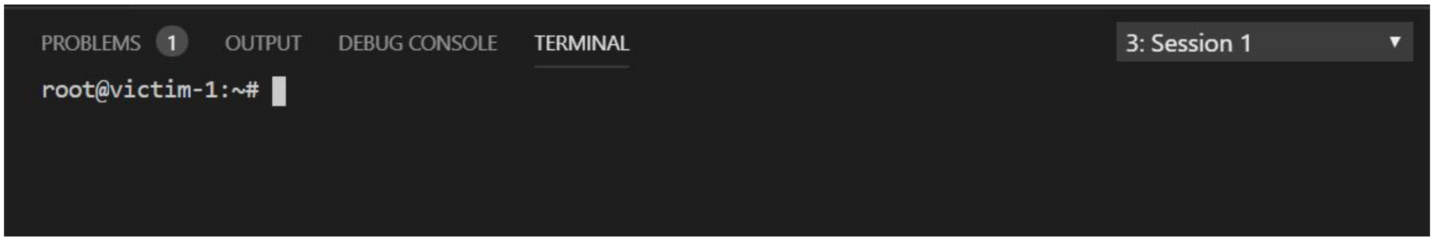
```python
from twisted.web import server, resource
from twisted.internet import reactor, endpoints

class Counter(resource.Resource):
    isLeaf = True
    numberRequests = 0

    def render_GET(self, request):
        self.numberRequests += 1
        request.setHeader(b"content-type", b"text/plain")
        content = u"I am modified request #{}\n".format(self.numberRequests)
        return content.encode("ascii")

endpoints.serverFromString(reactor, "tcp:80").listen(server.Site(Counter()))
reactor.run()
```

Step 11: Start the program without debugging. Click on the Terminal drop down list and select the "Session 1" Terminal to access the Kali machine.



```
  File "/root/.local/share/code-server/extensions/ms-python.python-2019.6.24221 n/ptvsd/wrapper.py",
 line 888, in handle_exiting
    self.send_event('exited', exitCode=exitcode or 0)
  File "/root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/pythonFiles/lib/python/ptvsd/ipcjson.py",
 line 232, in send_event
    body=kwargs,
  File "/root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/pythonFiles/lib/python/ptvsd/ipcjson.py",
 line 95, in _send
    self.__socket.send(content)
socket.error: [Errno 32] Broken pipe
root@attackdefense:/home/student/project# cd /home/student/project ; env PYTHONIOENCODING=UTF-8 PYTHONUNBUFFERED=1 /usr/b
in/python /root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/pythonFiles/ptvsd_launcher.py --default
 --client --host localhost --port 44369 /home/student/project/emulated-server.py
```

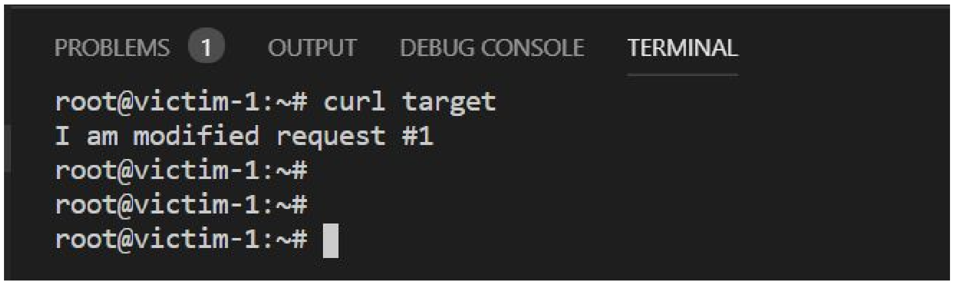
Kali Machine Terminal:



```
root@victim-1:~#
```

Step 12: Send an HTTP GET request to the server and check the received response.

Command: curl target



```
root@victim-1:~# curl target
I am modified request #1
root@victim-1:~#
root@victim-1:~#
root@victim-1:~#
```

The modification of the string in python code was reflected in the response received.

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/)