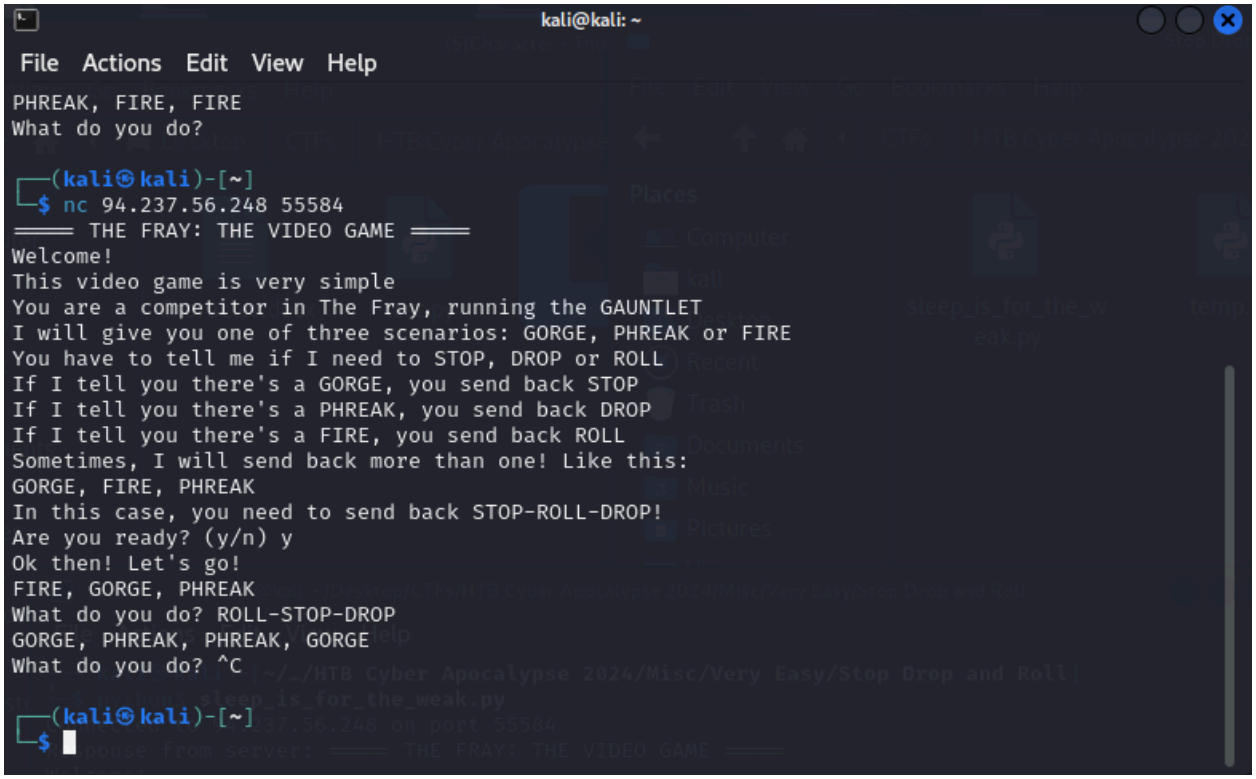
Main Solution:

When you initially netcat to the ip provided, you would notice that it is a game where you need to respond with the correct thing to do given either 1 or multiple scenarios.



So now to automate this process…
The finalized script that would not be terminated by the client is to have a time.sleep(.53), the script I used “beacons” every 2 seconds but have noticed that .53 seconds works as well

1.1 Finalized working script (BEST WAY IN MY OPINION)

The creator could have easily make you miss the flag by continuing with the program execution even after the flag was generated as such since we know the flag start with HTB, if at any point the response contains HTB, i would output it to a file named test.txt with the response.

```
import socket
import time
"""
        Usage: python3 temp.py

===== THE FRAY: THE VIDEO GAME =====
Welcome!
```

This video game is very simple
You are a competitor in The Fray, running the GAUNTLET
I will give you one of three scenarios: GORGE, PHREAK or FIRE
You have to tell me if I need to STOP, DROP or ROLL
If I tell you there's a GORGE, you send back STOP
If I tell you there's a PHREAK, you send back DROP
If I tell you there's a FIRE, you send back ROLL
Sometimes, I will send back more than one! Like this:
GORGE, FIRE, PHREAK
In this case, you need to send back STOP-ROLL-DROP!

"""

```
def main():
        target_ip = "94.237.56.248"
        target_port = 55584

   # Create a socket object
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        try:
      # Connect to the target
                client_socket.connect((target_ip, target_port))
                print("Connected to", target_ip, "on port", target_port)

      # Start sending and receiving data based on server's response
                while True:
                        # Receive response from the server
                        response = client_socket.recv(4096).decode()

                        if "Are you ready?" in response:
                                print("Response from server:", response)
                                data = input("Enter data to send (type 'exit' to quit): ") + "\n" #the \n
or sendline() from pwntools denotes the end of the response
                                client_socket.sendall(data.encode())
                        elif "What do you do?" in response:
                                formatted_response = response
                                formatted_response = formatted_response.replace('Ok then!
Let\'s go!', '').replace('What do you do?', '')
                                if "," in formatted_response:
                                        tosend = []
                                        formattedText = ""
                                        temps = response.split(',')
                                        for i in range(len(temps)):
                                                if "GORGE" in temps[i]:
```

```
                                        tosend.append("STOP")
                                if "PHREAK" in temps[i]:
                                        tosend.append("DROP")
                                if "FIRE" in temps[i]:
                                        tosend.append("ROLL")
                        for l in range(len(tosend)):
                                if l == (len(tosend)-1):
                                        formattedText += tosend[l] + "\n"
                                else:
                                        formattedText += tosend[l] + "-"
                        client_socket.sendall(formattedText.encode())
                        print("Multi Response from server:", response)
                        time.sleep(.53)
                        print("Multi Sent to server:", formattedText)
                elif "GORGE" in formatted_response or "PHREAK" in
formatted_response or "FIRE" in formatted_response:
                        singleTxt = ""
                        if "GORGE" in formatted_response:
                                singleTxt = "STOP" + "\n"
                        if "PHREAK" in formatted_response:
                                singleTxt = "DROP" + "\n"
                        if "FIRE" in formatted_response:
                                singleTxt = "ROLL" + "\n"
                        client_socket.sendall(singleTxt.encode())
                        print("Single Response from server:", response)
                        time.sleep(.53)
                        print("Single Sent to server:", singleTxt)
            elif "HTB" in response:
                print("Whatever else:", response)
                f = open("test.txt", "a")
                f.write(response)
                f.close()
        except ConnectionRefusedError:
                print("Connection refused. Make sure the server is running and the address/port
are correct.")
        except Exception as e:
                print("An error occurred:", e)
        finally:
    # Close the connection
                client_socket.close()

if __name__ == "__main__":
    main()
```

1.2 Python Script that I Used:

The difference between this and the above is that for the below i only print out the response that i need while in 1.1 i would save the formatted response in formatted_response and work from there, also the sleep duration for the below is longer and even if the response contains HTB (the flag) it would only print it and not output to a file which is dangerous as mentioned in 1.1 that the creator could have made their code in a way that even after generating the flag it would continue with the game

```
import socket
import time
"""
        Usage: python3 temp.py

===== THE FRAY: THE VIDEO GAME =====
Welcome!
This video game is very simple
You are a competitor in The Fray, running the GAUNTLET
I will give you one of three scenarios: GORGE, PHREAK or FIRE
You have to tell me if I need to STOP, DROP or ROLL
If I tell you there's a GORGE, you send back STOP
If I tell you there's a PHREAK, you send back DROP
If I tell you there's a FIRE, you send back ROLL
Sometimes, I will send back more than one! Like this:
GORGE, FIRE, PHREAK
In this case, you need to send back STOP-ROLL-DROP!

"""

def main():
        target_ip = "94.237.56.248"
        target_port = 55584

  # Create a socket object
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        try:
    # Connect to the target
                client_socket.connect((target_ip, target_port))
                print("Connected to", target_ip, "on port", target_port)

    # Start sending and receiving data based on server's response
                while True:
                        # Receive response from the server
                        response = client_socket.recv(4096).decode()
```

```
                if "Are you ready?" in response:
                        print("Response from server:", response)
                        data = input("Enter data to send (type 'exit' to quit): ") + "\n" #the \n
or sendline() from pwntools denotes the end of the response
                        client_socket.sendall(data.encode())
                elif "What do you do?" in response:
                        formatted_response = response
                        formatted_response = formatted_response.replace('Ok then! Let\'s
go!', '').replace('What do you do?', '')
                        if "," in formatted_response:
                                tosend = []
                                formattedText = ""
                                temps = response.split(',')
                                for i in range(len(temps)):
                                        if "GORGE" in temps[i]:
                                                tosend.append("STOP")
                                        if "PHREAK" in temps[i]:
                                                tosend.append("DROP")
                                        if "FIRE" in temps[i]:
                                                tosend.append("ROLL")
                                for l in range(len(tosend)):
                                        if l == (len(tosend)-1):
                                                formattedText += tosend[l] + "\n"
                                        else:
                                                formattedText += tosend[l] + "-"
                                client_socket.sendall(formattedText.encode())
                                print("Multi Response from server:", response)
                                time.sleep(2)
                                print("Multi Sent to server:", formattedText)
                        elif "GORGE" in formatted_response or "PHREAK" in
formatted_response or "FIRE" in formatted_response:
                                singleTxt = ""
                                if "GORGE" in formatted_response:
                                        singleTxt = "STOP" + "\n"
                                if "PHREAK" in formatted_response:
                                        singleTxt = "DROP" + "\n"
                                if "FIRE" in formatted_response:
                                        singleTxt = "ROLL" + "\n"
                                client_socket.sendall(singleTxt.encode())
                                print("Single Response from server:", response)
                                time.sleep(2)
                                print("Single Sent to server:", singleTxt)
                elif "HTB" in response:
```

```
                                        print("Whatever else:", response)
        except ConnectionRefusedError:
                print("Connection refused. Make sure the server is running and the address/port
are correct.")
        except Exception as e:
                print("An error occurred:", e)
        finally:
    # Close the connection
                client_socket.close()

if __name__ == "__main__":
    main()
```

Final Output

I think you can guess from that scrollbar that indeed this took waaaaaay tooooooo long