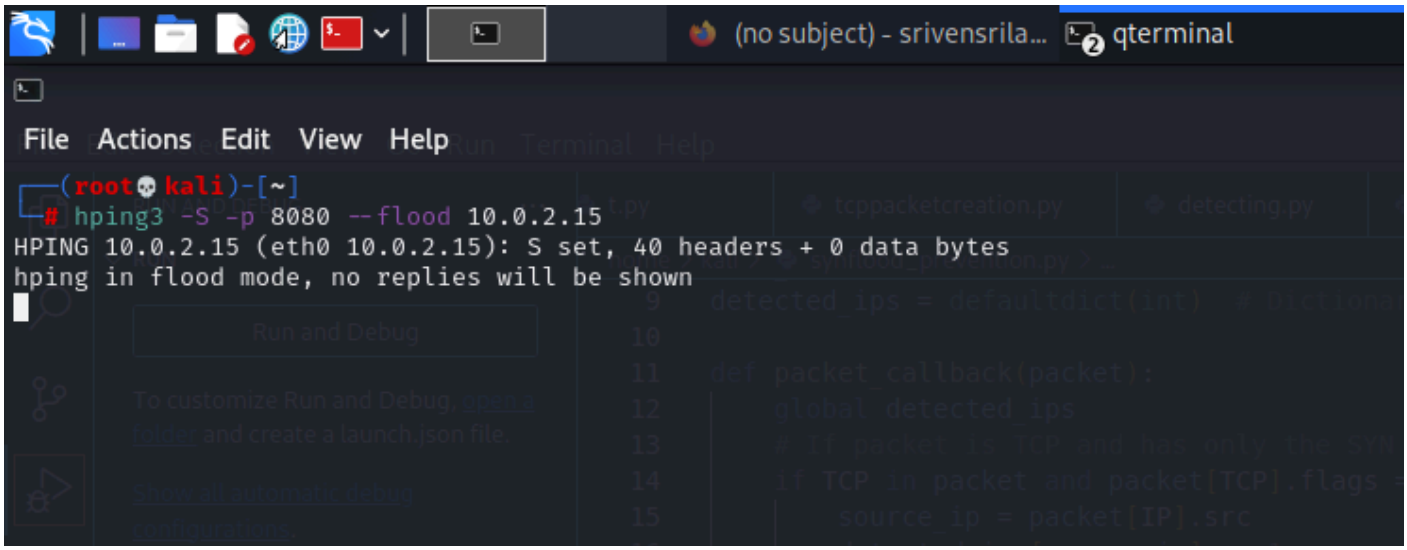
SYN Flood Attack Prevention

To prevent SYN FLOOD ATTACK, create syn flood through this hping3 code on kali linux

hping3 -S -p 8080 --flood 10.0.2.15



-S: Sets the SYN flag.
-p <port>: Specifies the port to target.
--flood: Sends packets as fast as possible.
<target_ip>: The IP address of the target system.

```python
from scapy.all import sniff, TCP, IP
from collections import defaultdict
import threading
import time

# Constants
SYN_THRESHOLD = 100  # Number of SYN packets to trigger the alarm.
TIME_WINDOW = 60     # Time window in seconds to check for SYN_THRESHOLD
detected_ips = defaultdict(int)  # Dictionary to hold IP addresses and their SYN counts

def packet_callback(packet):
    global detected_ips
    # If packet is TCP and has only the SYN flag set.
    if TCP in packet and packet[TCP].flags == 'S':
        source_ip = packet[IP].src
        detected_ips[source_ip] += 1

def syn_flood_detector():
    # Sniff TCP packets and pass them to the callback function.
    sniff(prn=packet_callback, filter="tcp", store=0)

if __name__ == "__main__":
    # Start the detector in a separate thread
    detector_thread = threading.Thread(target=syn_flood_detector)
    detector_thread.start()

    try:
        while True:
            time.sleep(TIME_WINDOW)

            # Check the counts and trigger alert for potential SYN flood
            for ip, count in detected_ips.items():
                if count > SYN_THRESHOLD:
                    print(f"[ALERT] Possible SYN flood attack from {ip}. Detected {count} SYN packets in the last {TIME_WINDOW} seconds.")

            # Reset the counts after checking
            detected_ips.clear()

    except KeyboardInterrupt:
        print("Stopping the SYN flood detector.")
```

Preventing SYN flood attacks involves several strategies to manage the half-open connections that characterize these attacks. Here are some effective methods:

1. **SYN Cookies**: Implement SYN cookies to avoid allocating resources for half-open connections until the handshake is completed. This technique encodes the connection parameters into the initial sequence number.
2. **Firewall Rules**: Use firewalls to limit the number of SYN packets per second from a single IP address. Rate limiting can help mitigate the impact of an attack.
3. **Connection Limiting**: Configure your server to limit the maximum number of half-open connections. This can help control resource usage during an attack.
4. **TCP Intercept**: Some routers and firewalls support TCP intercept, which can monitor SYN packets and help validate the legitimacy of the connection attempts.

5. **Intrusion Detection Systems (IDS)**: Deploy IDS to monitor for unusual traffic patterns that may indicate an ongoing SYN flood attack, allowing for timely responses.
6. **Load Balancing**: Use load balancers to distribute incoming connections across multiple servers, which can help absorb the impact of an attack.
7. **Increasing Backlog Queue Size**: Adjust the backlog queue size for pending connections, though this alone may not be sufficient against large-scale attacks.
8. **CAPTCHA Challenges**: Implementing CAPTCHA challenges on web applications can help differentiate between legitimate users and automated attack traffic.
9. **Regular Updates and Patching**: Ensure that your operating system and applications are up to date with the latest security patches to minimize vulnerabilities that could be exploited.