**INFORMATION SECURITY AND ASSURANCE**

**FINAL PROJECT REPORT**

**HTTP FLOODING:**

Language: python

HTTP flood is a type of [**Distributed Denial of Service**](https://www.incapsula.com/ddos/ddos-attacks/denial-of-service.html) **(**[**DDoS**](https://www.incapsula.com/ddos/ddos-attacks/denial-of-service.html)**)** attack in which the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application.

**Attack Description**:

When an HTTP client like a web browser “talks” to an application or server, it sends an HTTP request - generally one of two types of requests: GET or POST. A GET request is used to retrieve standard, static content like images while POST requests are used to access dynamically generated resources. The victims machine is continuously set to a busy state by sending continuous requests from different networks using a Distributed denial of service(DDos), until all resources for incoming connections on the server (the victim) are used up, hence making any further (including legitimate) connections impossible until all data has been sent.

**Attack and Defense Setup**

I have used Ubuntu for both attacker and Victim for http flooding.

I have installed a virtual Box an installed two Ubuntu and connected them to a host network. So, the IP addresses of the machines are like 192.168.230.100 and 192.168.230.101.

I have installed HTTP server to run the local host in the victim machine. Also installed Snort to Defend the attack.

**SYSTEM CONFIGURATIONS:**

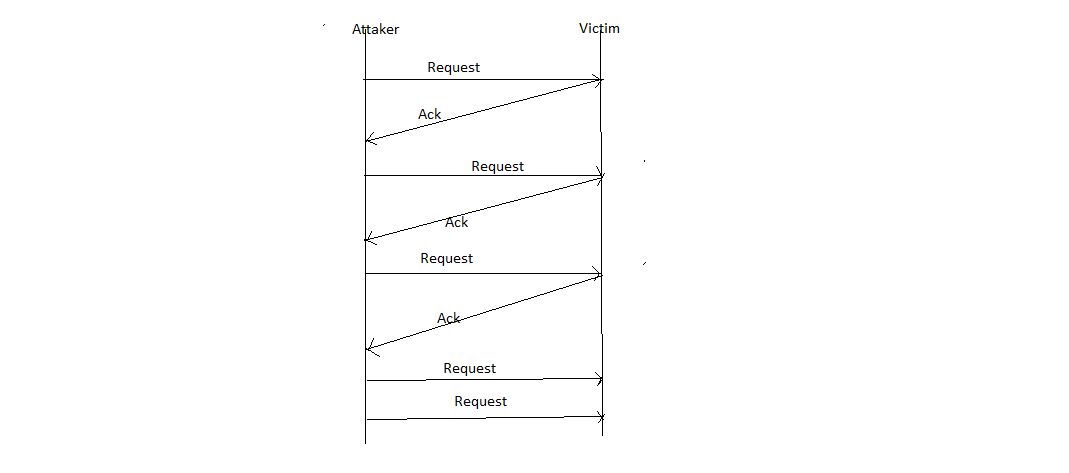
Attacker: Ubuntu 10.10, IPAddress 192.168.230.100

Defense: Ubuntu 16.04.1, IPAddress 192.168.230.101

Attacker: Ubuntu 10.10

Defense: Ubuntu 16.04.1

**ATTACK SEQUENCE:**



**Protocols**

I have used tcp protocol for http flooding.

**Attack Flow:**

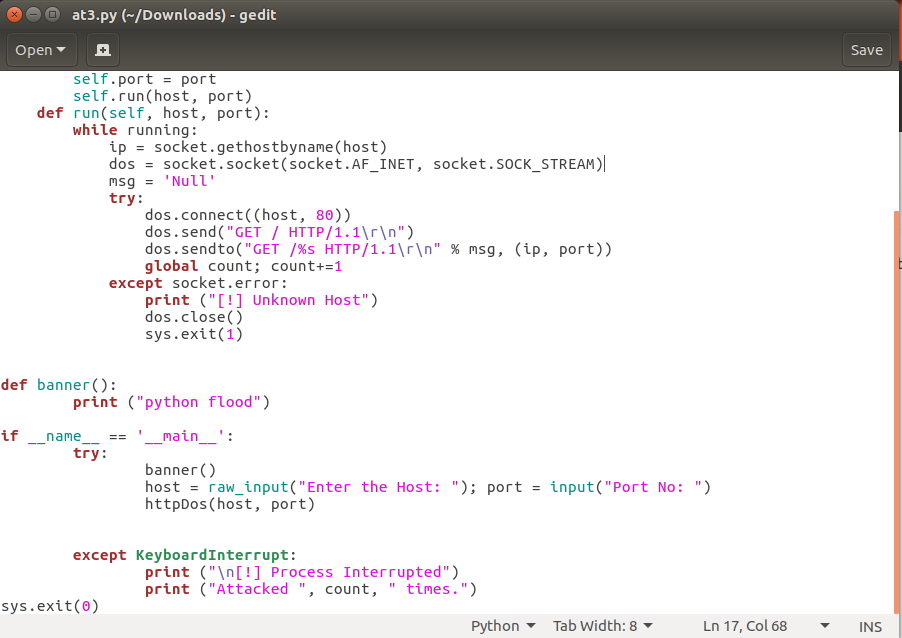
I have installed http server in the victim machine and then started the http server using the command

🡺 sudo systemctl start apache2.service

After the http server is started you have to check the localhost in the web browser, if the default web page is displayed the http server is working.

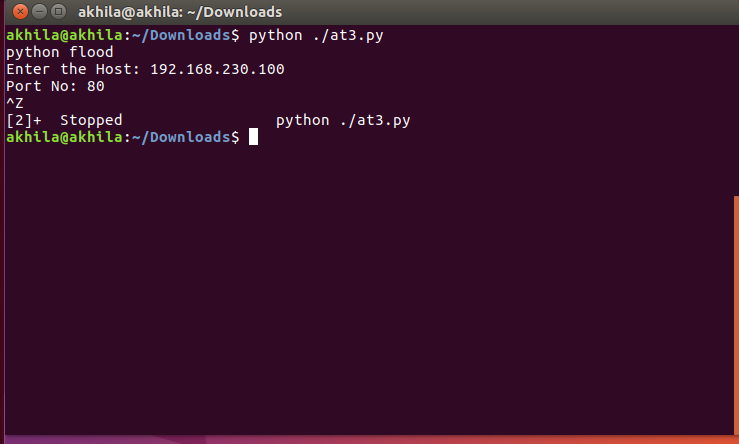
Now on the attacker side I have run the following python script.





After running the code I have checked the victim machine if I can open the local host on the web browser , I couldn’t open it .

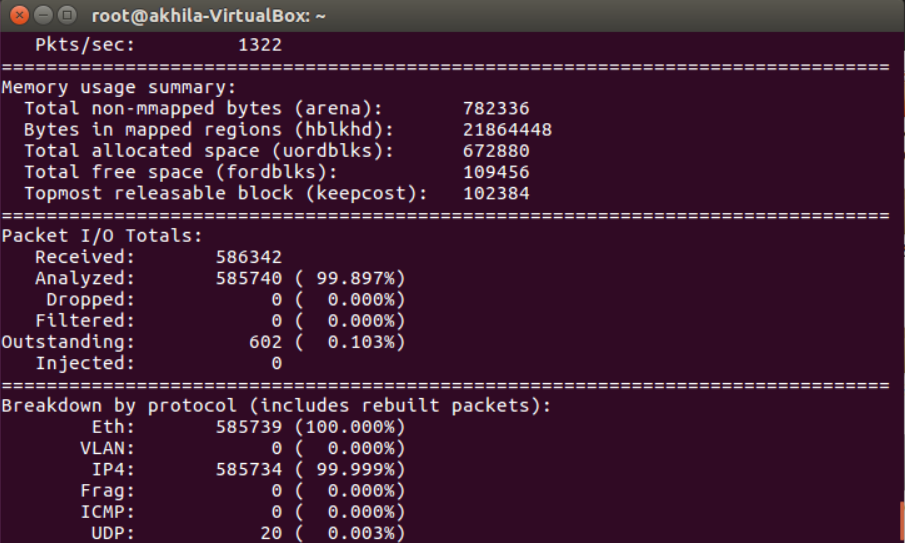
Now I have installed snort IDS in the victim machine to defend the attack. I have added my snort rule in the snort.conf file.

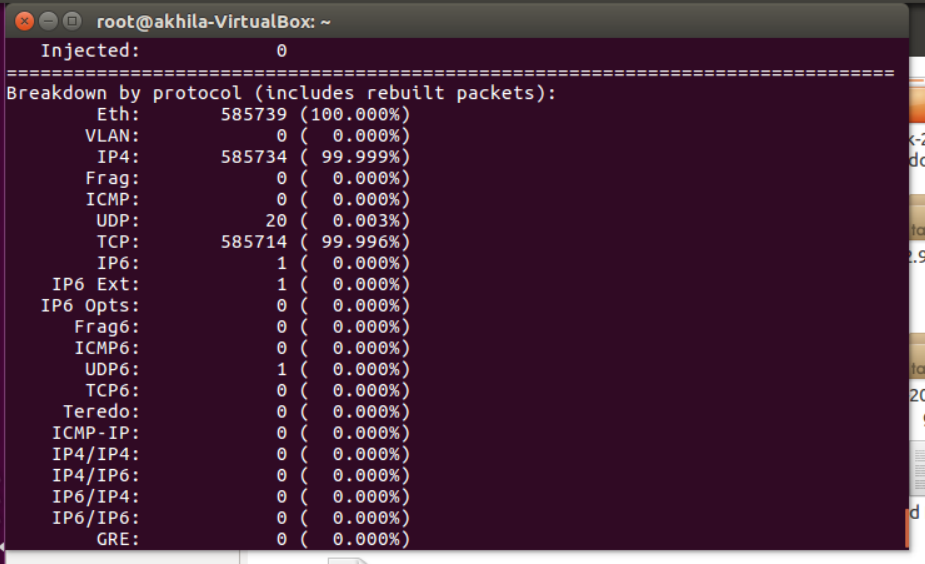


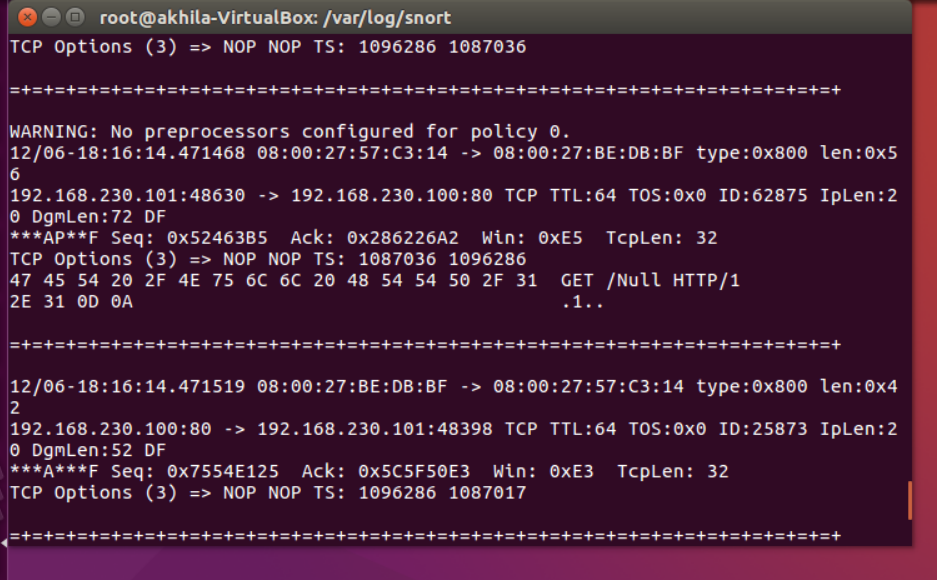
And my Snort rule is :

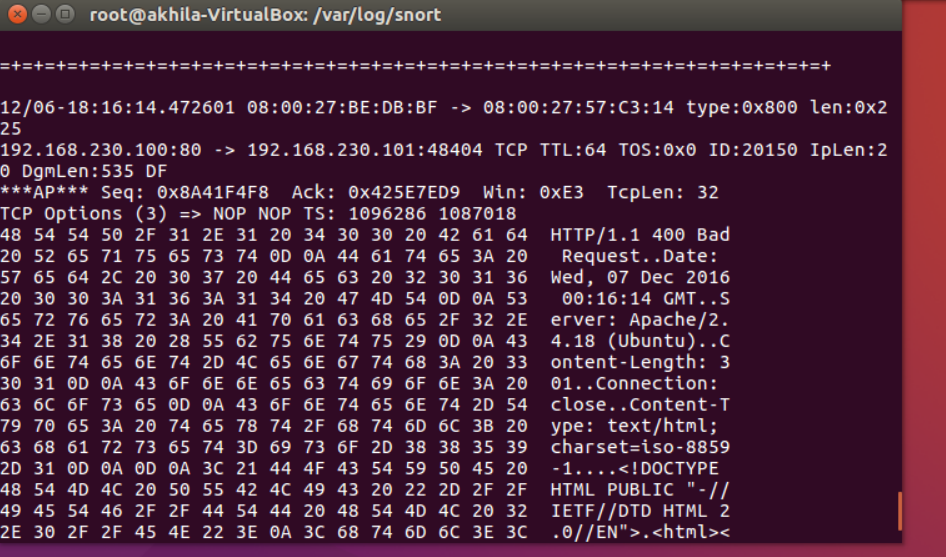
alert tcp any any -> 192.168.230.101 80 (msg: "Flood attempt using GET request!!!"; flow: to\_server , established; content: "GET"; detection\_filter: track by\_src, count 60, seconds 30; )

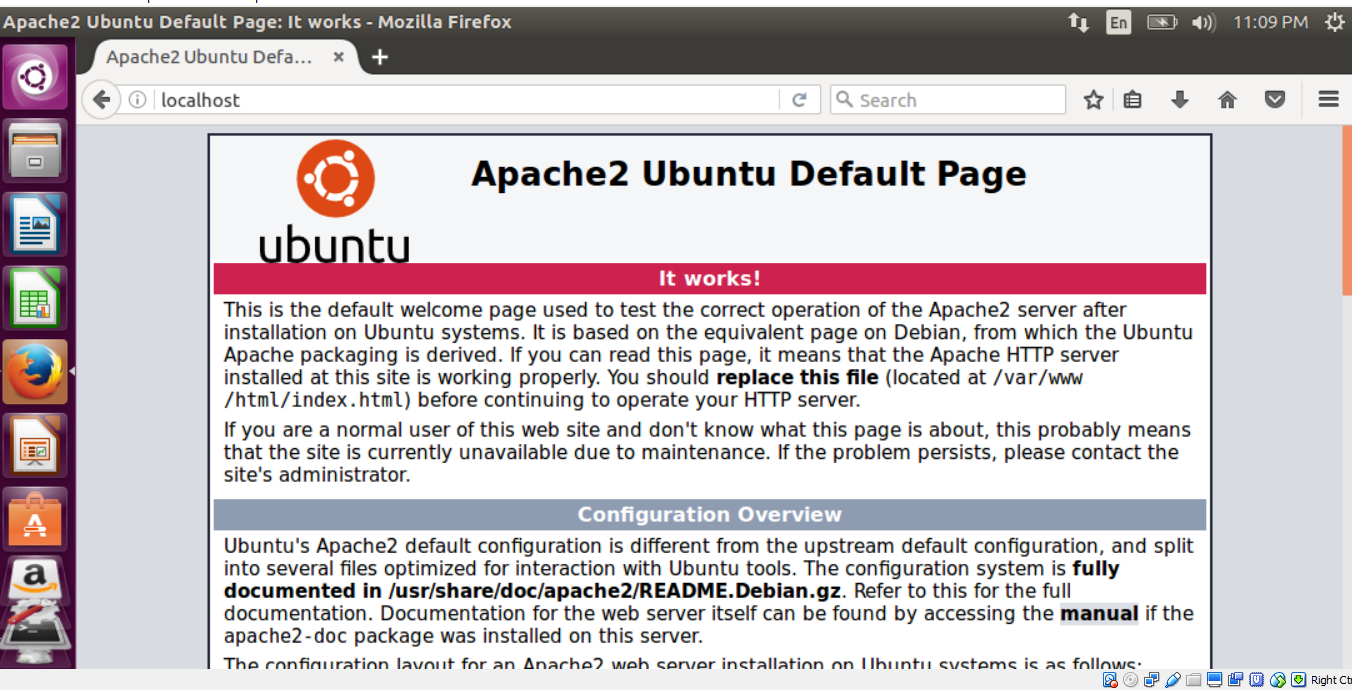
Alerting a tcp connection from any ipaddress and any port to victims ip address for port 80(http) by sending a message “Flood attempt using GET request!!! “ and specifying the flow to\_server and if the tcp session is established and if the more than 60 packets is transmitted in a time span of 30 Seconds from a particular source.











**CONCLUSION:**

When we are performing the attack, connecting the ubuntus to a host network we cannot access web.

The victim machine should keep the http server on all the time which is not all the possible in the real world.

For Defending the attack the snort configure file shows errors even if the changes are made.