**ISA Project**

**Team Details:**

Sharath Koppu

Aravind Sheri

Harish Joshi

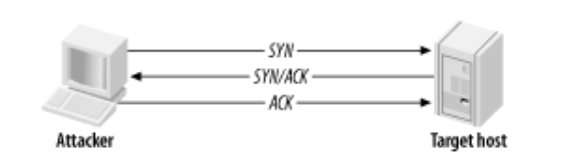
**Phase 1**

**1. TCP Port Scanning -**

A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to 10 different destination ports within a defined interval (5000 microseconds is the default). The purpose of this attack is to scan the available services in the hopes that at least one port will respond, thus identifying a service to target.

**2. Flow Diagram –**

**TCP Scan result when port is Open**



**3. Project Configurations –**

Written in **Java** language and implemented through **Eclipse Framework**

**Client** – Windows Operating Systems

**Target** – Ubuntu OS on Vmware Workstation

**Applications started in target before attacking**

Port 22 – SSH

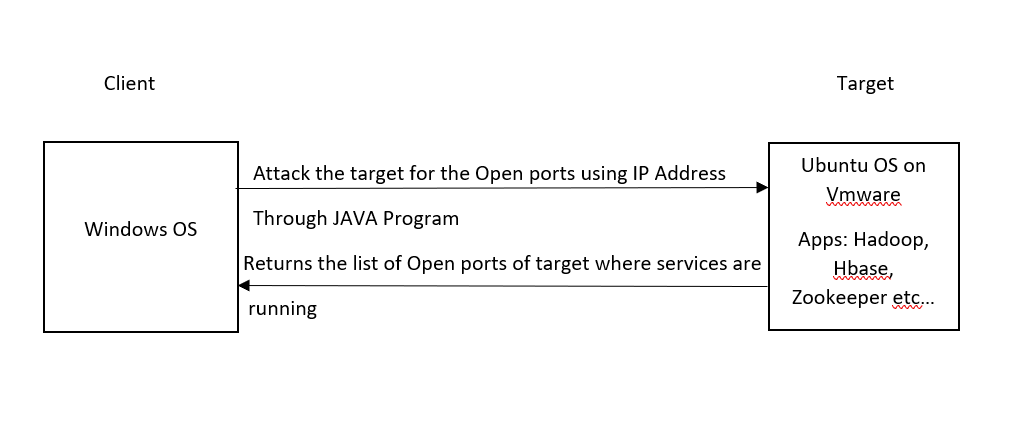
Port 2181 - Zookeeper

Port 13562 – Mapreduce

Port 16010 – Hbase

Port 50070 – Hadoop

**Project Setup**



**4. Source Code –**

import java.net.\*;

class TcpPortScanner {

public static void main(String []args) {

for (int port = 1; port <= 65000; port++) {

try {

Socket socket = new Socket();

socket.connect(new InetSocketAddress("192.168.157.131", port), 10);

socket.close();

System.out.println("Port " + port + " is open");

} catch (Exception ex) { }

}}

**Output –**

Port 22 is open

Port 2181 is open

Port 8030 is open

Port 8031 is open

Port 8032 is open

Port 8033 is open

Port 8040 is open

Port 8042 is open

Port 8088 is open

Port 13562 is open

Port 16010 is open

Port 16301 is open

Port 41437 is open

Port 50010 is open

Port 50020 is open

Port 50070 is open

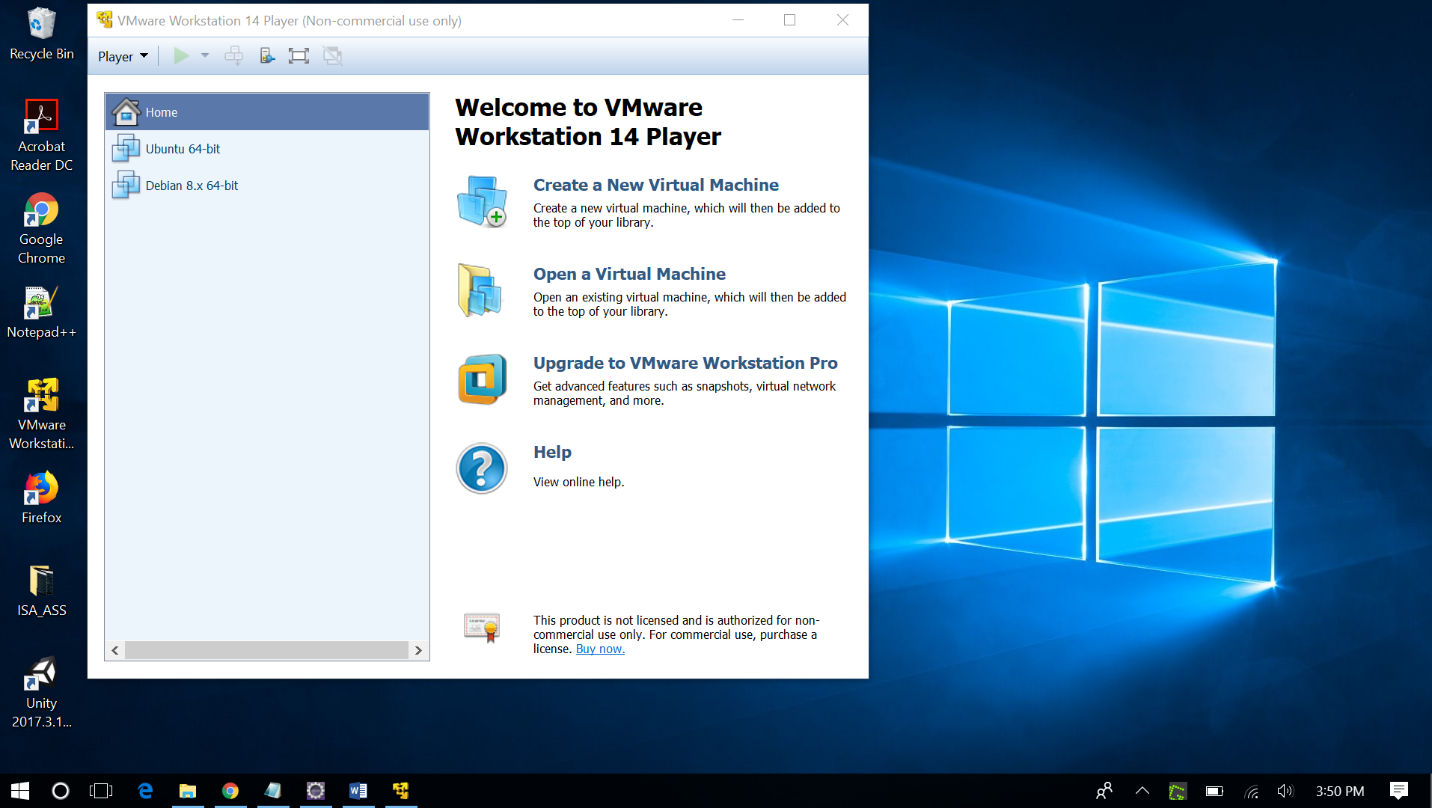
Port 50075 is open

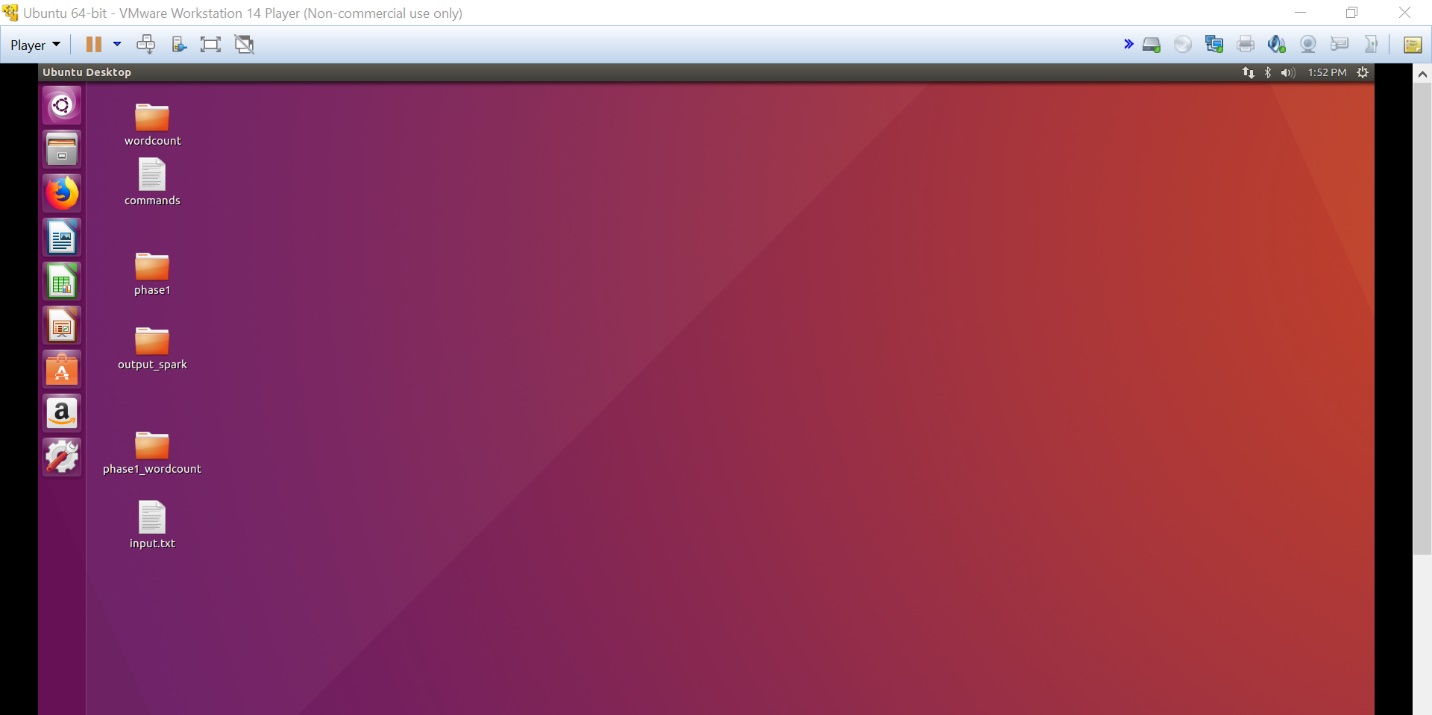
Port 50090 is open

**Execution –**

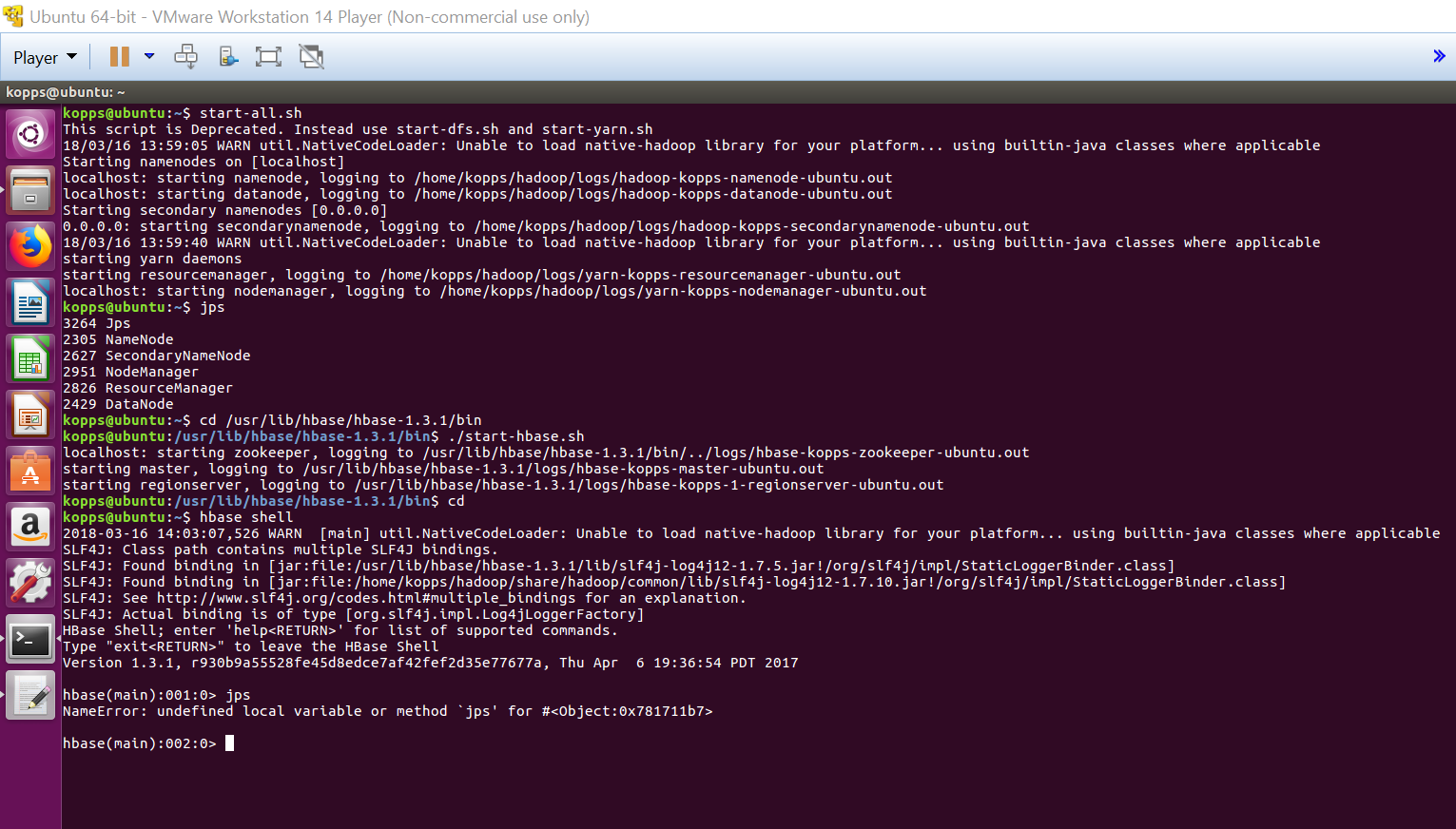
1. Save the code in a file with name as TcpPortScanner
2. Compile the program using javac TcpPortScanner.java
3. Run the program using java TcpPortScanner
4. Make sure that java is installed in the system
5. **Screenshots –**

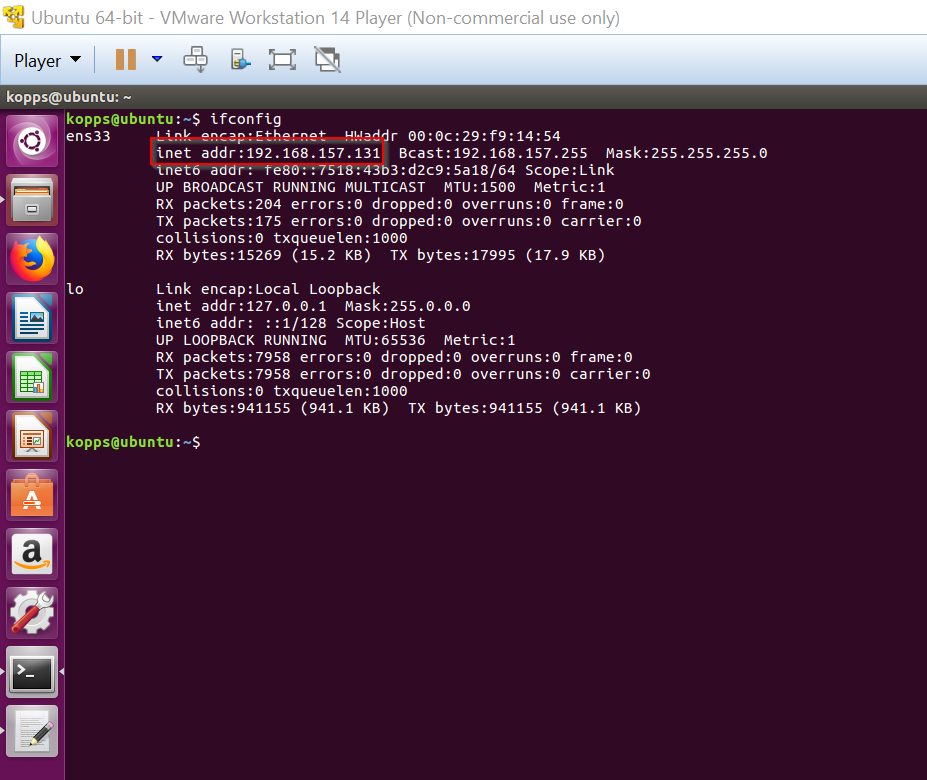
**Target System -> Vmware -> Ubuntu OS**

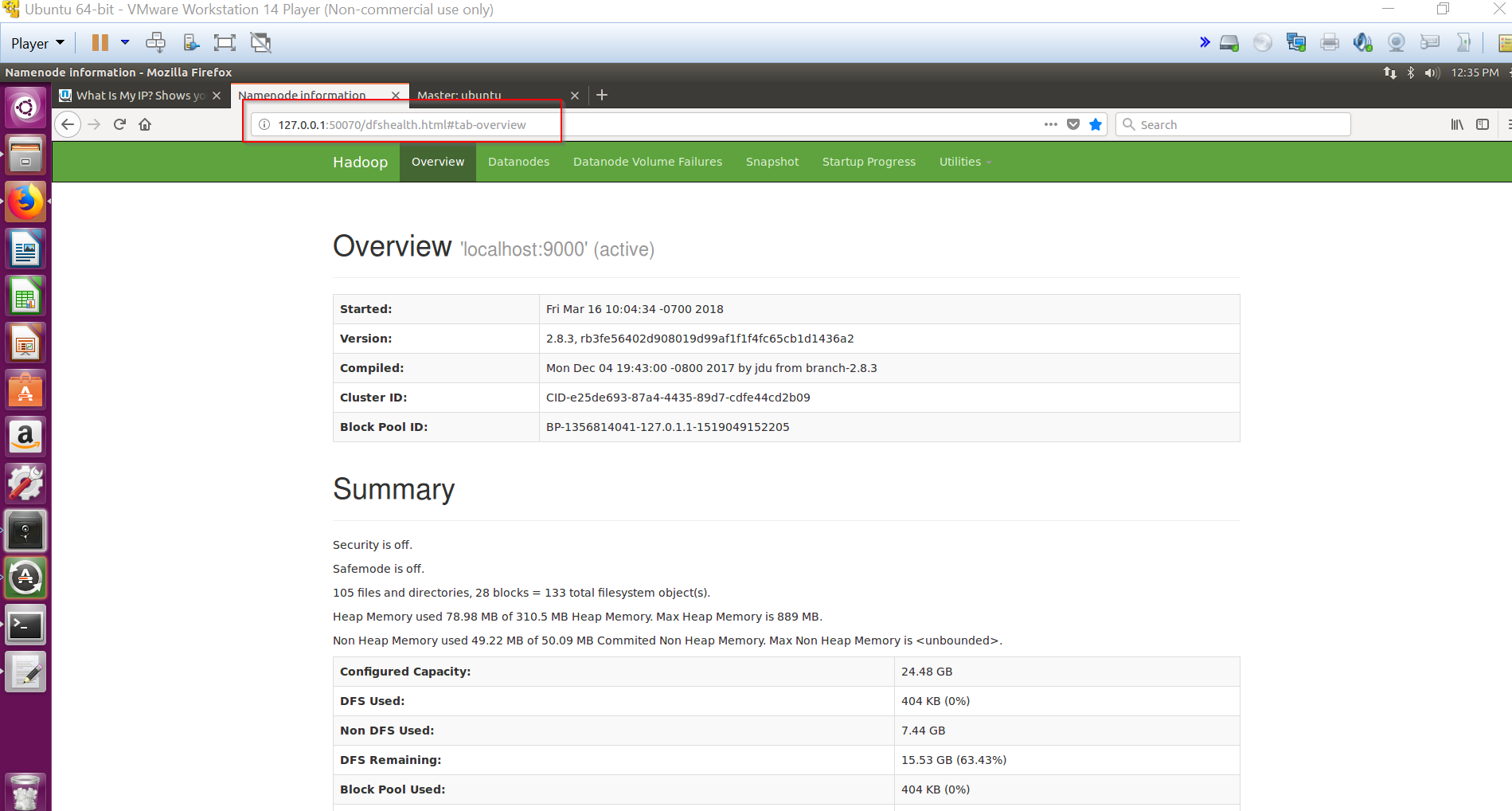


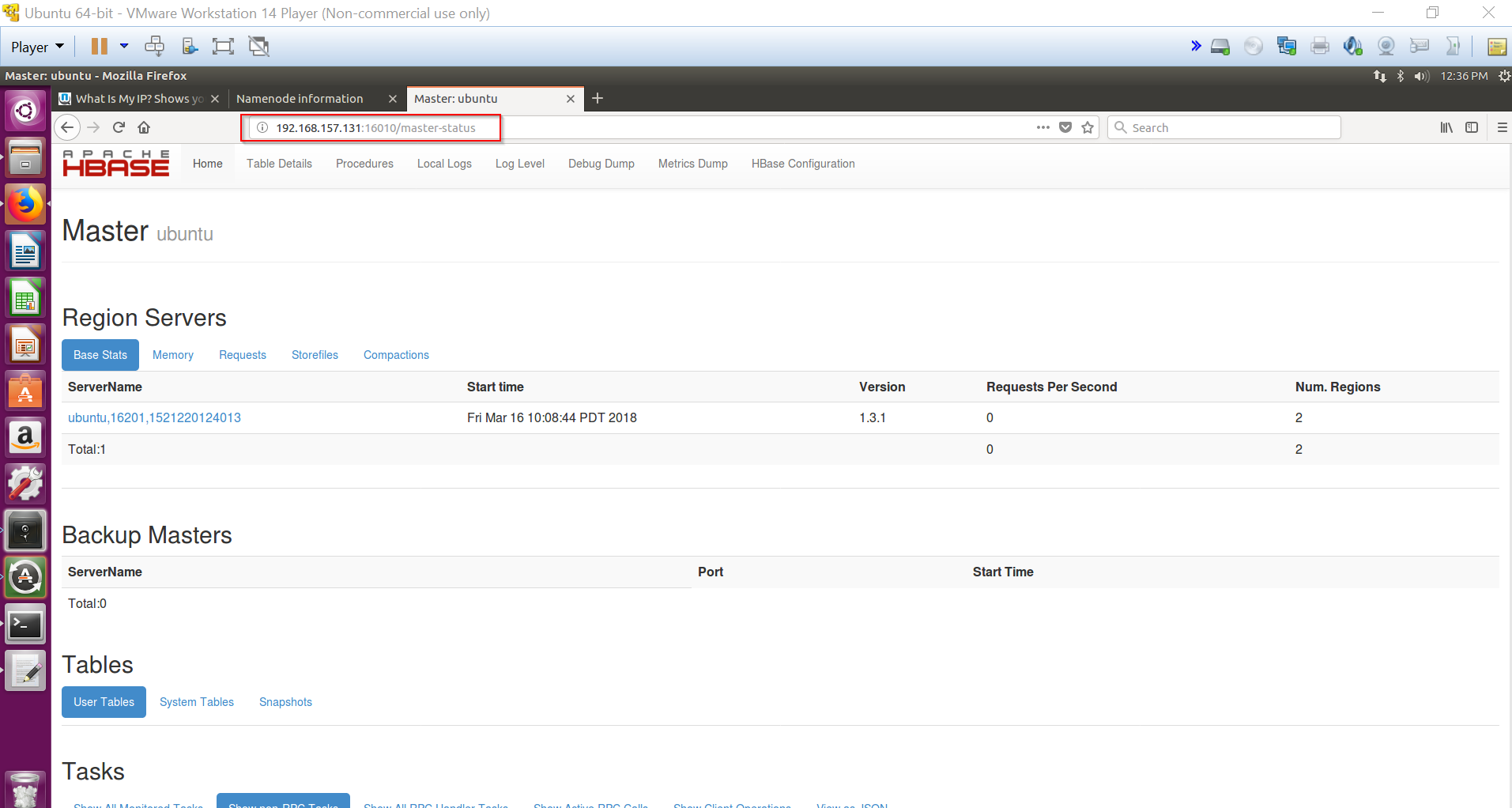


**Running the applications Hadoop, Zookeeper and Hbase in the local system**

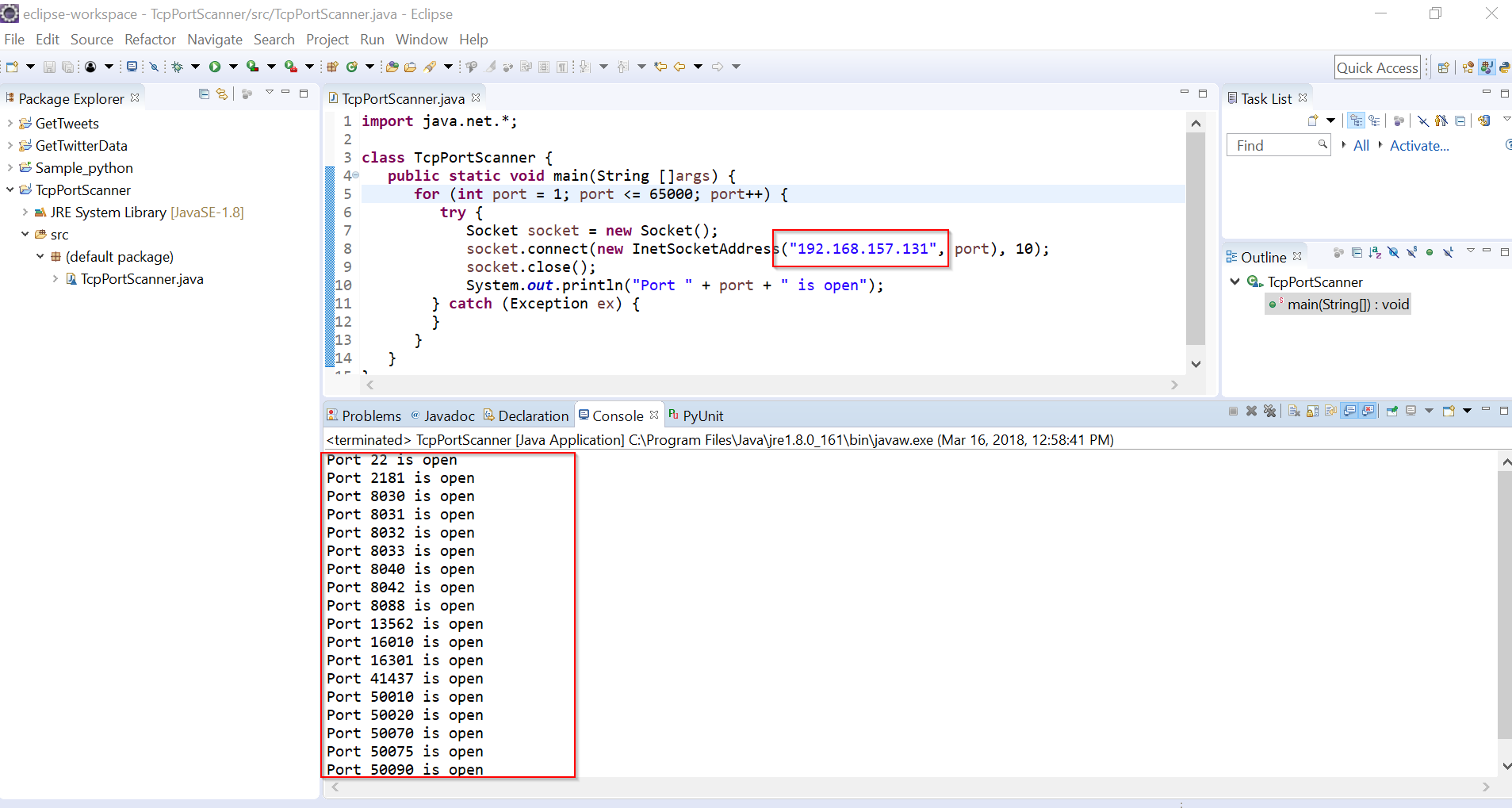








**Client System**



**6. References**

1. <http://etutorials.org/Networking/network+security+assessment/Chapter+4.+IP+Network+Scanning/4.2+TCP+Port+Scanning/>
2. <https://www.garykessler.net/library/is_tools_scan.html>
3. <https://stackoverflow.com/questions/1826977/get-application-server-name-or-ip-and-port-in-java>
4. <https://www.youtube.com/results?search_query=tcp+port+scanner>

**Phase 2**

1. **Snort rules to detect Attacker**

The snort rule that runs on the victim machine informs the victim by alarming him based on the below snort rule that is configured in local file of Snort.

**Regular Traffic:**

In regular traffic, snort identifies and prevents the packets even though packets are not coming from attacker system. The packets can come from the authorized user too. The following is the snort rule for regular traffic:

**alert tcp any any -> 192.168.1.200 any (msg:"TCP PORT SCAN";flags:S,12;sid:10000006)**

**Attack Traffic:**

In attack traffic, snort can identify and prevents the packets coming from attacker itself. This can be done by adding type threshold in rule options of rule header. The following is the snort rule for this traffic:

**Rule:**

**alert tcp any any -> 192.168.1.200 any (msg:"TCP PORT SCAN";flags:S,12;threshold:type threshold, track by\_src, count 40, seconds 60; sid:10000006)**

**The above snort on the victim machine will reject the packets coming from a particular source which send the more than 40 packets in minute**

**Rule Header Prototype: Rule Format: Rule Header (Rule Options)**

**Action Protocol Src\_IP Src\_Port Direction Dest\_IP Dest\_Port**

**Action:** It generates an alert using the selected alert method, and then log the packet. In the above alert is an action that notifies the victim about the attacker

**Protocol:** This is a protocol that Snort currently analyzes for suspicious behavior. Snort supports TCP, UDP, ICMP and IP. In the above tcp protocol is used to detect the applications/services of target running on TCP port.

**Source & Destination IP:** It specifies the IP address of the attacker from which the packets are being sent to the target victim IP address. In the above “any” is used to detect any attacker and victim IP address.

**Source & Destination Port:** It specifies Port number of the attacker from which packets are being sent to the target victim port numbers. In the above “any” is used to detect any attacker and victim port.

**Rule Options:**

Rule options form the heart of Snort’s intrusion detection engine, combining ease of use with power and flexibility. All Snort rule options are separated from each other using the semicolon (;) character. Rule option keywords are separated from their arguments with a colon (:) character.

**Msg -** The msg rule option tells the logging and alerting engine the message to print along with a packet dump or to an alert.

**Flags –** The flags keyword is used to check if specific TCP flag bits are present**.** In the above, flag S - SYN is used to Synchronize sequence numbers

**Sid -** The sid keyword is used to uniquely identify Snort rules. This information allows output plugins to identify rules easily. In the above any id number can be given in the options.

**Threshold -** These will embed threshold into the rule. For instance, a rule for detecting a too many login password attempts may require more than 5 attempts. This can be done using the ‘limit’ type of threshold. It makes sense that the threshold feature is an integral part of this rule

Format:

threshold: \

type <limit|threshold|both>, \

track <by\_src|by\_dst>, \

count <c>, seconds <s>;

**type threshold -** Type threshold alerts every m times we see this event during the time interval.

**track by src –** Here**,** Rate is tracked by source IP address This means count is maintained for each unique source IP addresses .

**Count -** Number of rule matching in s seconds that will cause event filter limit to be exceeded. c must be nonzero value**.** In the above rule count is given as 40 seconds.

**Seconds -** Timeperiod over which count is accrued. In the above rule count is given as 60 seconds.

1. **Snort commands**

Snort is configured in Network Intrusion Detection System (NIDS) mode to alert the victim and is described below:

**snort -i 2 -c c:\Snort\etc\snort.conf -A fast**

In the above,

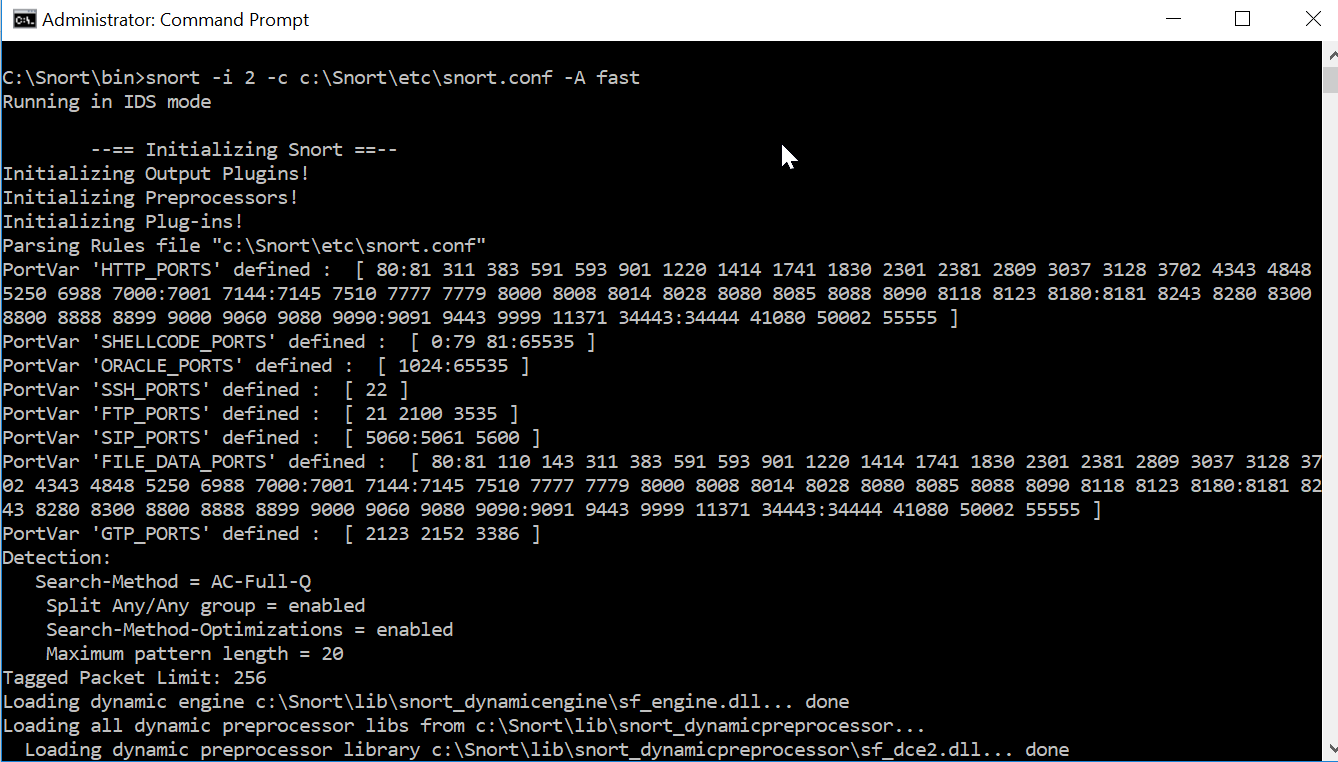
i – Specifies the Snort interface

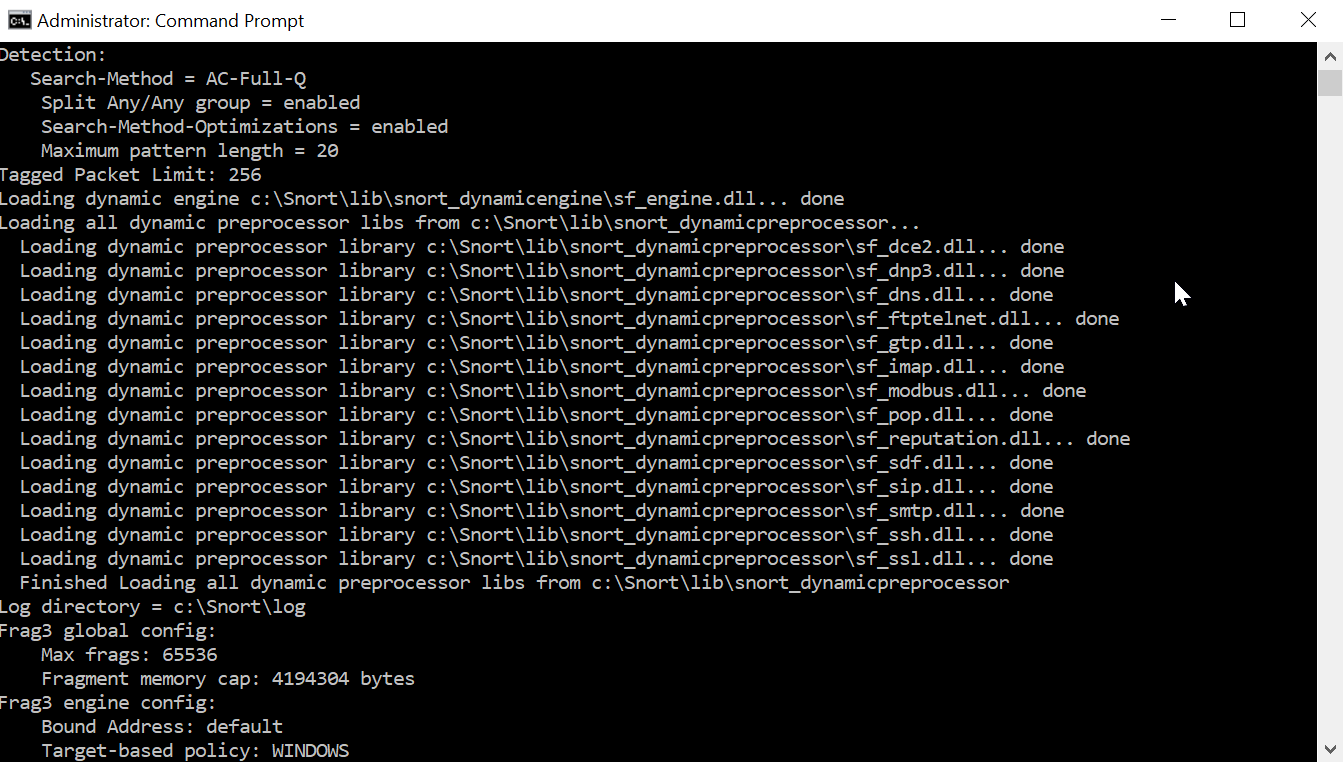
2 specifies the interface number

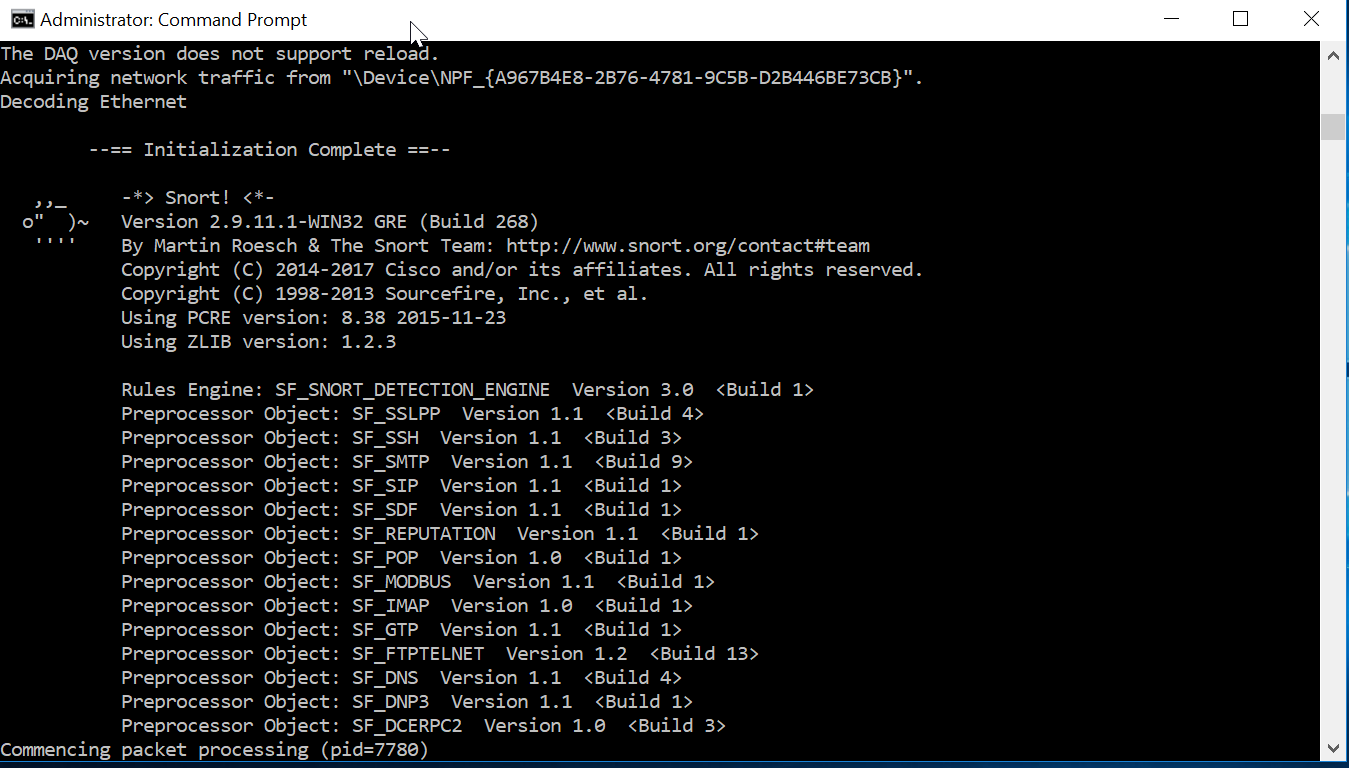
c - Describes the location of snort configuration file in the system

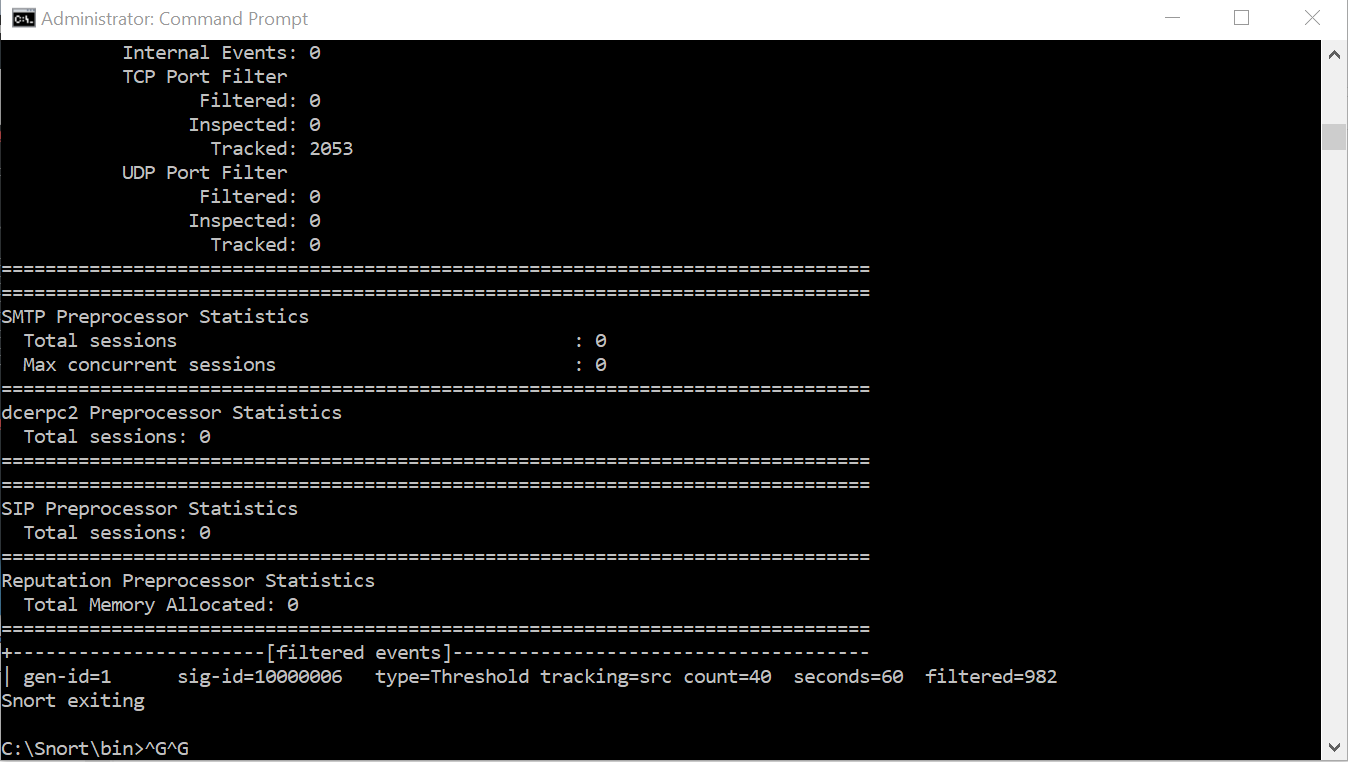
-A fast - alert mode writes the alert in a simple format with a timestamp, alert message, source and destination IPs/ports.

**Running Snort in Victim machine**









1. **Alert IDS File**

Snort throws the alerts if it matches the snort rule options in the alert file of Snort log directory. The following is the output of the IDS file:

04/10-12:00:39.358985 [\*\*] [122:1:1] (portscan) TCP Portscan [\*\*] [Classification: Attempted Information Leak] [Priority: 2] {PROTO:255} 192.168.157.138 -> 192.168.1.200

04/10-12:00:39.569359 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:5414

04/10-12:00:40.614896 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:5000

04/10-12:00:41.670859 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:2607

04/10-12:00:42.709299 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:2967

04/10-12:00:45.787269 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:2393

04/10-12:00:45.791192 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:8090

04/10-12:00:45.820604 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:2500

04/10-12:00:45.834666 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:5100

04/10-12:00:45.839997 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:5960

04/10-12:00:45.862226 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:8000

04/10-12:00:45.865562 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:3828

04/10-12:00:45.879024 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:45100

04/10-12:00:45.893904 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:1056

04/10-12:00:45.907790 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:1999

04/10-12:00:45.923538 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:1021

04/10-12:00:45.926204 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:3168

04/10-12:00:45.940710 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:12265

04/10-12:00:45.961858 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:10629

04/10-12:00:45.982206 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:3300

04/10-12:00:45.999631 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:15742

04/10-12:00:46.002014 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:7200

04/10-12:00:46.020988 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:1063

04/10-12:00:46.058811 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:64680

04/10-12:00:46.079643 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:1088

04/10-12:00:46.098852 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:12000

04/10-12:00:46.101382 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34269 -> 192.168.1.200:3995

04/10-12:00:47.238553 [\*\*] [1:10000006:1] TCP PORT SCAN [\*\*] [Priority: 0] {TCP} 192.168.157.138:34270 -> 192.168.1.200:4343

1. **Difficulties faced:**

Faced problems during running snort.conf file. Had to make modifications in snort.conf file in order to remove the errors

**Knowledge Gained:** Gained practical knowledge on how to attack the web servers in real time and at the same time how to detect the attackers to prevent attacking.

1. **References:**

* <https://www.snort.org/configurations>
* <https://blog.rapid7.com/2016/12/09/understanding-and-configuring-snort-rules/>
* <https://www.youtube.com/watch?v=RwWM0srLSg0>