

2014

ON SYSTEM

INTRUSION DETECTION SYSTEM

[illegible]

Raghav Bisht
Root-X
4/28/2014

Root-X

4/28/2014

CAPSTONE PROJECT REPORT

(Project Semester January- May, 2014)

INTRUSION DETECTION SYSTEM

Submitted by

RAGHAV BISHT

PROJECT GROUP NUMBER:

Under the Guidance of

Declaration

We hereby declare that the project work entitled (Intrusion Detection System) is an authentic record of our own work carried out as requirements of Capstone Project for the award of degree of B.Tech in Information & Technology from Lovely Professional University, Phagwara, under the guidance of MS. Cherry Khosla, during January to May, 2014.

Student : Raghav Bisht

Registration Number: 00000000

(Signature of Student 1)

Root-X Raghav

Certificate

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. The Capstone Project Proposal based on the technology (Intrusion Detection System) learnt is fit for the submission and partial fulfillment of the conditions for the award of B.Tech in Information & Technology from (_____ University.)

Name:

U.ID:

Designation:

Signature of Faculty Mentor

Contents

S.NO	TITLE	PAGE NO.
1.	Introduction	7 - 8
2.	Profile Of The Problem	9 - 10
3.	Existing System <ul style="list-style-type: none">• Introduction• Existing Software• DFD For Present System	11 - 14
4.	Problem Analysis <ul style="list-style-type: none">• Product definition• Feasibility Analysis• Project Plan	15 - 18
5.	Software Requirement Analysis <ul style="list-style-type: none">• Specific Requirements	19 - 21
6.	Design <ul style="list-style-type: none">• Detail Design• Flowcharts• Pseudo Code	22 - 60
7.	Testing <ul style="list-style-type: none">• Functional Testing• Structural Testing• Penetration Testing	61 - 74
8.	Implementation <ul style="list-style-type: none">• Implementation Of The Project	

	<ul style="list-style-type: none"> • Post-Implementation and Software Maintenance 	75
9.	Project Legacy <ul style="list-style-type: none"> • Current Status Of The Project • Remaining Areas Of Concern • Technical and Managerial Lessons Learnt 	76
10.	User Manual	77 - 81
11.	Snapshots	82 - 89
12.	Bibliography	90

Root-X Raghna

Introduction

1.1 Intrusion Detection Systems

Intrusion Detection Systems help information systems prepare for, and deal with attacks. They accomplish this by collecting information from a variety of systems and network sources, and then analyzing the information for possible security problems. An intrusion detection system (IDS) examines system or network activity to find possible intrusions or attacks. Intrusion detection systems are either network-based or host-based. Network based intrusion detection systems are most common, and examine passing network traffic for signs of intrusion. Host-based systems look at user and process activity on the local machine for signs of intrusion. Since each type has specific strengths and weaknesses.

There are three main components to the Intrusion Detection System

1. Network Node Intrusion Detection System (NNIDS) performs the analysis of the traffic that is passed from the network to a specific host. The difference between NIDS and NNIDS is that the traffic is monitored on the single host only and not for the entire subnet.

Example of the NNIDS would be, installing it on a VPN device, to examine the traffic once it was decrypted. This way you can see if someone is trying to break into your VPN device.

- Host Intrusion Detection System (HIDS) – takes a snap shot of your existing system files and matches it to the previous snap shot. If the critical system files were modified or deleted, the alert is sent to the administrator to investigate.

Example of the HIDS can be seen on the mission critical machines that are not expected to change their configuration.

1.2 What Intrusion Detection System CAN And CAN NOT Provide

The IDS however is not an answer to all your Security related problems. You have to know what you CAN, and CAN NOT expect of your IDS. In the following subsections I will try to show a few examples of what an Intrusion Detection Systems are capable of, but each network environment varies and each system needs to be tailored to meet your enterprise environment needs.

The IDS CAN provide the following:

- CAN add a greater degree of integrity to the rest of your infrastructure

- CAN trace user activity from point of entry to point of impact
- CAN recognize and report alterations to data
- CAN automate a task of monitoring the Internet searching for the latest attacks
- CAN detect when your system is under attack
- CAN detect errors in your system configuration
- CAN guide system administrator in the vital step of establishing a policy for your computing assets
- CAN make the security management of your system possible by non-expert staff

The IDS CAN NOT provide:

- CAN NOT compensate for a weak identification and authentication mechanisms
- CAN NOT conduct investigations of attacks without human intervention
- CAN NOT compensate for weaknesses in network protocols
- CAN NOT compensate for problems in the quality or integrity of information the system provides.
- CAN NOT analyze all the traffic on a busy network
- CAN NOT always deal with problems involving packet-level attacks
- CAN NOT deal with some of the modern network hardware and features.

2. Profile Of The Problem

Information Systems and Networks are subject to electronic attacks. Attempts to breach information security are rising every day, along with the availability of the Vulnerability Assessment tools that widely available on the Internet, for free, as well as for a commercial use. Tools such as Subs even, BackOrifice, Nmap, L0ftCrack, can all be used to scan, identify, probe, and penetrate your systems. Firewalls are put in place to prevent unauthorized access to the Enterprise Networks.

Let's, however, ask ourselves: Are the firewalls enough?

An example.

Imagine that you have just purchased a state of the art Home Theatre System. Everyone who knows anything about electronics, have an idea of how much it may cost. After installing it, you decided that you might need to install new locks on all the doors in your house, because the old ones do not use the up to date secure mechanisms. You call the locksmith, and in about 2 month (if you are lucky) you have a new locks on your doors, and you are the only one who have the keys (well, maybe you mother have another pair). With that in mind you pack your things, and with whatever money you got left from you recent purchases, you go on vacation.

As you came back a week later, you find that the Entertainment room looks different. After careful examination, you realize that your Home Theater System, that you were dwelling over for the last year, is missing. What worse is that your wife told you that the window in the kitchen is broken, and there is boot stains on the carpet, all over the house. That led you to believe that someone broke into your house, stole, and vandalized a lot of your prized possessions. After you wipe the tears from your eyes, you suddenly begin to vaguely remember the brochure that you got, about a burglar alarm installation in your neighborhood. You threw it away just a week before. The installation and monitoring would have cost you 19.95 a month with this promotional offer. Neglecting to install the system, is a secret that you would have to leave with for the rest of your life could you have prevented it from happening, were you to install an alarm? May be not completely, but the damage would be much less.

The real life example above is the exact same analogy of what might happen to your network. What's worth is that the thief may be on your network for a long time, and you might not even know it. Firewalls are doing a good job guarding your front doors, but they do not have

a possibility to alert you in case there is a backdoor or a hole in the infrastructure. Script kiddies are constantly scanning the Internet for known bugs in the system, including constant scans by subnets. More experienced crackers may be hired by your competitors, to target your network specifically, in order to gain competitive advantage. The list of threats can go on.

Root-X Raghav

3. Existing Software

In this section we will discuss some of the current generation intrusion detection system.

3.1 Introduction

Network IDS

The network IDS usually has two logical components: the sensor and the management station. The sensor sits on a network segment, monitoring it for suspicious traffic. The management station receives alarms from the sensor(s) and displays them to an operator.

The sensors are usually dedicated systems that exist only to monitor the network. They have a network interface in promiscuous mode, which means they receive all network traffic not just that destined for their IP address and they capture passing network traffic for analysis. If they detect something that looks unusual, they pass it back to the analysis station.

Host IDS

The host-based IDS looks for signs of intrusion on the local host system. These frequently use the host system's audit and logging mechanism as a source of information for analysis.

They look for unusual activity that is confined to the local host such as logins, improper file access, unapproved privilege escalation, or alterations on system privileges. This IDS architecture generally uses rule-based engines for analyzing activity; an example of such a rule might be, "super user privilege can only be attained through the su command." Therefore successive login attempts to the root account might be considered an attack.

3.2 Existing software

3.2.1 SNORT

Snort's open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching, and content matching. These basic services have many purposes including application-aware triggered quality of service, to deprioritize bulk traffic when latency-sensitive applications are in use.

The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, common gateway interface, buffer overflows,

server message block probes, and stealth port scans. Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console. In packet logger mode, the program will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyze it against a rule set defined by the user. The program will then perform a specific action based on what has been identified.

3.2.1.1 Snort Performance

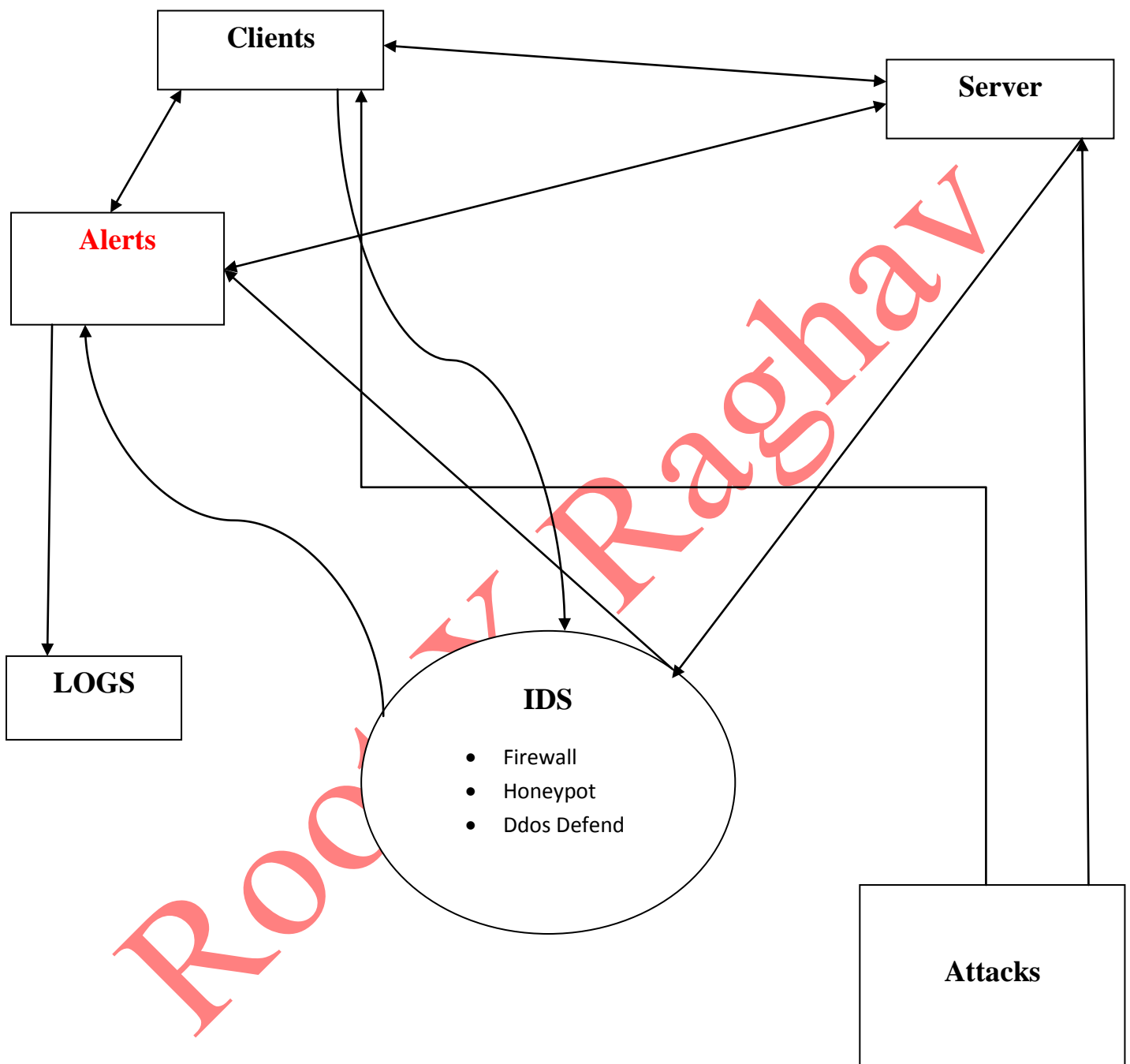
We obtained a profiling result of Snort (ver 2.6.1.5) on Intel 2.0 GHz processor on DebianLinux 2.6.18 using a GNU gprof [11] profiler (ver 2.16). Snort was configured with five preprocessing components (Stream4, frag2, HTTP Inspect, Telnet decode and sfportscan) and 6565 rules. The network traces used to obtain the profiling result was obtained from MIT Lincoln Lab's 1998 DARPA Intrusion Detection Evaluation project file (tcpdump file format)[12]. Table 1 shows the percentage of the total execution time used by each component (Profiling results depend on Snort configuration and test data).

3.2.2 Tiger

Tiger is a security tool that can be used both as a security audit and intrusion detection system. It supports multiple UNIX platforms and it is *free* and provided under a GPL license. Unlike other tools, **Tiger** needs only of POSIX tools and is written entirely in shell language.

Tiger has some interesting features that merit its resurrection, including a modular design that is easy to expand, and its double edge, it can be used as an audit tool and a host intrusion detection system tool. Free Software intrusion detection is currently going many ways, from network IDS (with Snort), to the kernel (LIDS, or SNARE for Linux and Systrace for OpenBSD, for example), not mentioning file integrity checkers (many of these: aide, integrity samhain, tripwire...) and logcheckers (even more of these, check the Log Analysis pages). But few of them focus on the host-side of intrusion detection fully. Tiger complements this tools and also provides a framework in which all of them can work together. Tiger it is not a logchecker, nor it focused in integrity analysis. It does "the other stuff", it checks the system configuration and status. Read the manpage for a full description of checks implemented in Tiger.

3.3 DFD for present system



3.4 What's new in the system

As we know IDS system are intrusion detection system and it can only detect the attack at real time so, as compare to 3rd party IDS software like SNORT & TIGER some new features are added to our IDS like :-

- **Trojans Scanner** : It check the active connection and find the possible Trojans plus report the administrator about it.
- **Shell Finder** : The program find the shell and back-connect backdoors in apache server and report it to the administrator.
- **PSAD** : well known as "Port Scan Attack Detector". The concept of the PSAD is from well known book called "The Art Of War" By Sun Tzu where he states "**If you know the enemy and know yourself, you need not fear the result of a hundred battles**". Similar port scanning is the first step of the hacking to know your enemy so, PSAD detect and alert the administrator.
- **Fake Access Point** : The fake access point is a honeypot which attract attacker and administrator can easily knows about attacker and its capabilities.
- **ADS Blocker** : The program block ads and spam for user.

4. Problem Analysis

In this phase all the problems associated with the IDS development process will be discussed and new uncovered problem will be discovered.

4.1 Product Definition

There are various factors and benchmarks which are to be kept in mind whilst we design and develop a IDS . This phase of the IDS project development process decides the actual outcome of the IDS . The environment and the IDS objects in the system must support the storyline and should be rendered in such way that they are a treat to the eyes of the hackers. At the same time we'll have to make sure that the IDS is light weight so as to run at an optimal pace with minimum system requirements. Hence it is important to know the problems in hand and also to grade them in the best possible so that it would help us in deciding up on the factors that is to be given a higher priority while designing and developing the IDS.

4.2 Feasibility Analysis

It is an analysis of our idea related to the system and give a validity and make our idea important. It takes an effort and necessity of thinking ability about the system feasibility of a problem occurred. Feasibility is the study of a significant or strong influence, what occurs at the time of system development. The influence can be either positive or negative. The system is considered feasible when positivity entitles negativity. Feasibility study can be performed in various ways related to various fields, we are describing four important way of performing feasibility study which are described following:-

4.2.1 Technical Feasibility

Technical feasibility of a system defines the compatibility, comfort, ability to achieve using current existing technology. It takes into attention weather the required technology is available or not and it also check for available resources like equipment's and software tools for development of the system.

We can say that our developed system is technically feasible because we are not getting any difficulty related to resource of development and maintenance of this project. Whatever software tools related to the development of system are commonly available and easy to get from internet and any other way like shop, friends etc.

4.2.2 Economical Feasibility

This system is highly economic feasible because it is not taking any extra tools other than our required tools for development which are easily available and free to download and use for development of projects. We need not to spend more money for the development of the system. It is making an environment for the development with an effective manner. If we do as it than we can see the maximum usability of the related resources of system. After development of this system, we need not to be attentive for this system. Therefore we can say that, this system is economically feasible.

4.2.3 Schedule Feasibility

It is defined as the state of being probable and completed within scheduled time. Our Project can be fail when it takes too long to be completed before it is used. It means estimating the project with respect of time that how long this system will take to develop. Schedule feasibility is a measurement of is timetable for project is reasonable. We discuss with our team and decide is the project deadline reasonable? Our project is initiated with specific deadline. We have determined whether the deadlines are mandatory or desirable.

4.2.4 Operational Feasibility

It is related to the measurement of performance of system for which purpose it is developed. It relates to all the functions and features related to the system and look for speed of execution of requests came from users and effectiveness of response in well manner. It provides an advantage of the opportunities introduced at time of scope definition and its satisfaction of requirement identified. It also provide satisfaction for phase of system development.

It ensures desired operational outputs which is the part of design and development. It includes design parameters like reliability, maintainability, supportability, usability of system for users. There all parameters are required for consideration of stages of design.

A design and development of system requires appropriate and timely application software for development and provide efforts to meet the mentioned parameters which are defined previously. A system performs its planned purpose most effectively when its technical and

operational characteristics are identified into the design. So, we can say that operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

4.3 Project Plan






























		Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors
1			 feasibility study	5 days	Wed 15-01-14	Tue 21-01-14	
2			technical feasibility	2 days	Wed 15-01-14	Thu 16-01-14	
3			economical feasibility	2 days	Fri 17-01-14	Mon 20-01-14	
4			behavioural feasibility	1 day	Tue 21-01-14	Tue 21-01-14	
5			 requirement analysis	8 days	Wed 22-01-14	Fri 31-01-14	
6			requirement gathering	3 days	Wed 22-01-14	Fri 24-01-14	
7			group interaction	3 days	Sat 25-01-14	Tue 28-01-14	
8			analysis	2 days	Wed 29-01-14	Thu 30-01-14	
9			 UI design	20 days	Mon 03-02-14	Fri 28-02-14	
10			firewall	20 days	Mon 03-02-14	Fri 28-02-14	
11			honeypot	20 days	Mon 03-02-14	Fri 28-02-14	
12			Ddos	20 days	Mon 03-02-14	Fri 28-02-14	
13			log management	20 days	Mon 03-02-14	Fri 28-02-14	
14			 Coding	21 days	Mon 03-03-14	Mon 31-03-14	
15			firewall	21 days	Mon 03-03-14	Mon 31-03-14	
16			honeypot	21 days	Mon 03-03-14	Mon 31-03-14	
17			Ddos	21 days	Mon 03-03-14	Mon 31-03-14	
18			log management	21 days	Mon 03-03-14	Mon 31-03-14	
19			 testing	14 days	Tue 01-04-14	Fri 18-04-14	
20			unit testing	6 days	Tue 01-04-14	Tue 08-04-14	
21			integration testing	5 days	Wed 09-04-14	Tue 15-04-14	
22			system testing	3 days	Wed 16-04-14	Fri 18-04-14	

Fig 4.3.1 : Project plan

Jan to Feb :

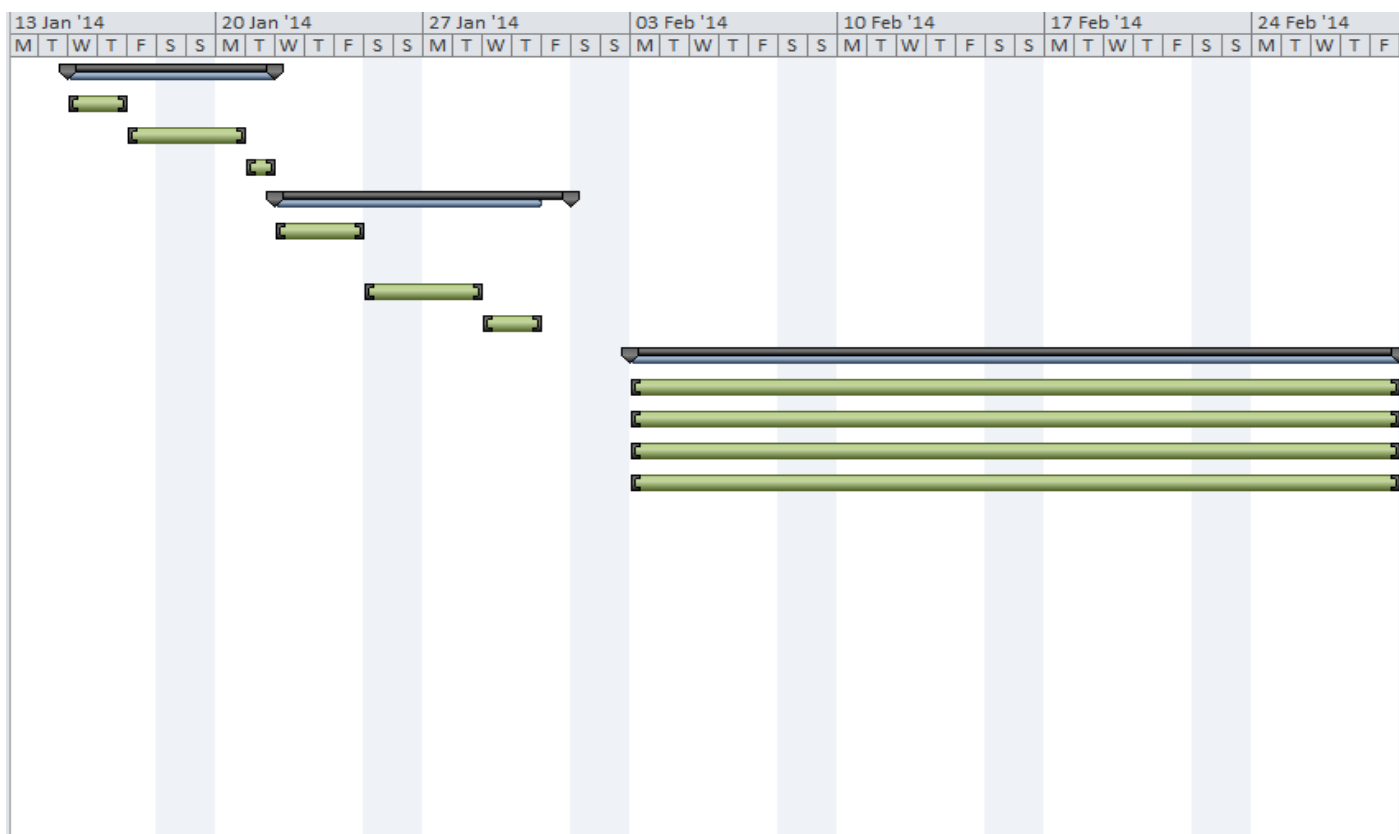


Fig 4.3.2 : Gantt Chart From Jan-Feb

March to April :

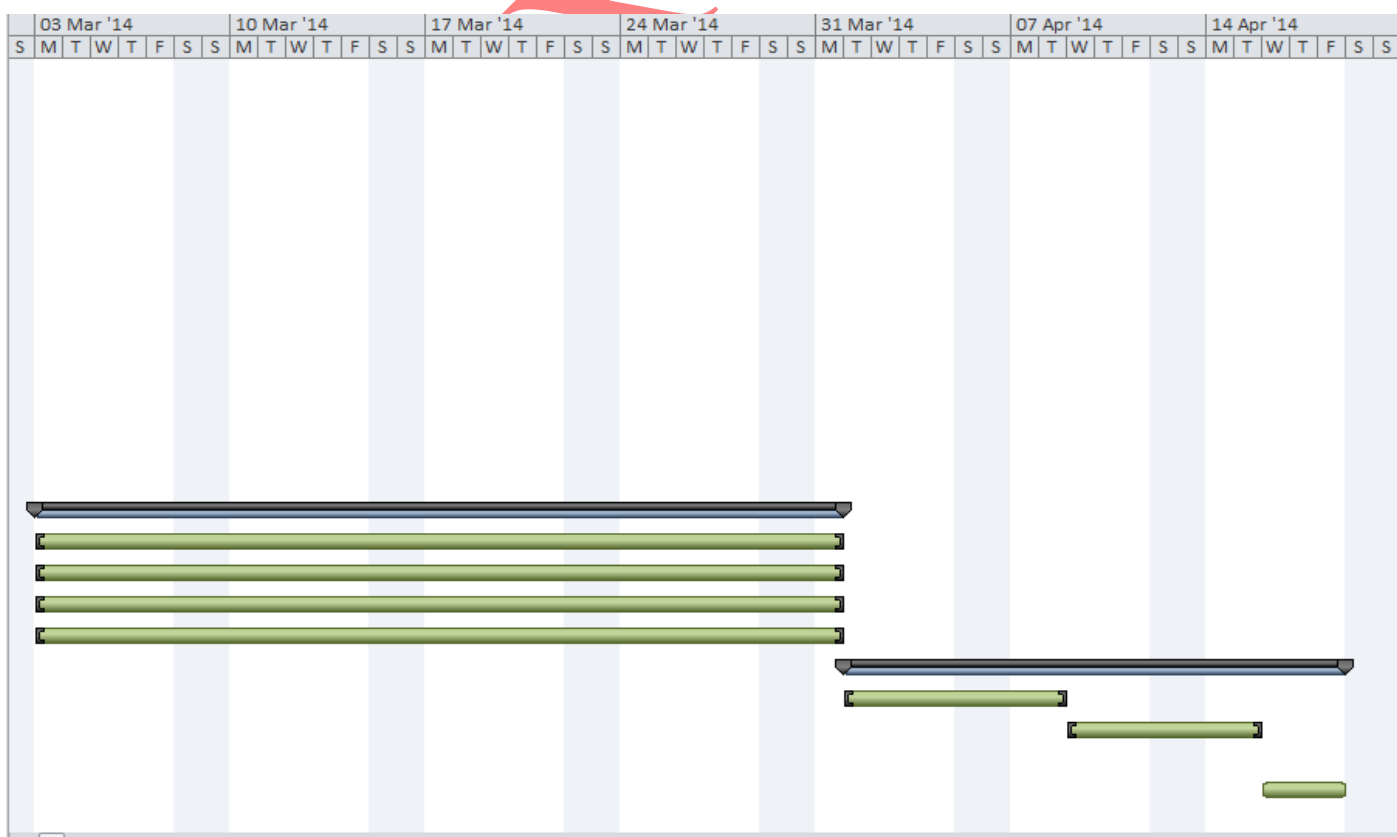


Fig 4.3.2 : Gantt Chart From March-April

5. Software Requirement Analysis

5.1. Specific Requirements

➤ System Requirements

- **Linux Operating System.**
- **Any penetration testing operating system like :**
 - Backtrack OS
 - Kali OS
 - NodeZero OS
 - BlackBox Linux
 - BlackBuntu Linux
 - Samurai Web Testing Framework
 - Knoppix STD
 - Pentoo Linux
 - Weakerth4n Linux
 - Matriux Krypton
 - DEFT
 - CAINE
 - Bugtraq
- **Linux Distribution Like :**
 - Fedora
 - Redhat
 - Ubuntu
 - OpenSuse
 - Slowloris
 - Bodhi
 - Xubuntu
 - Tails
 - Puppy Linux
 - Linux Mint
 - Google Chrome
- **Must Install Following Software's :**
 - Support Perl Programming
 - Wireshark

- Apache Server
- Tcp-Dump
- Tshark

➤ **System Architecture :**

• **System Hardware Architecture :**

- Linux compatible laptop or PC With OS architecture x86 Or x64.
- CPU : Core 2 Duo
- RAM : At least 4Gb
- HDD : 320Gb
- Virtual Memory: Max Size: 12,215 MB
- Network Card : Realtek PCIe GBE Family Controller.

• **System Software Architecture :**

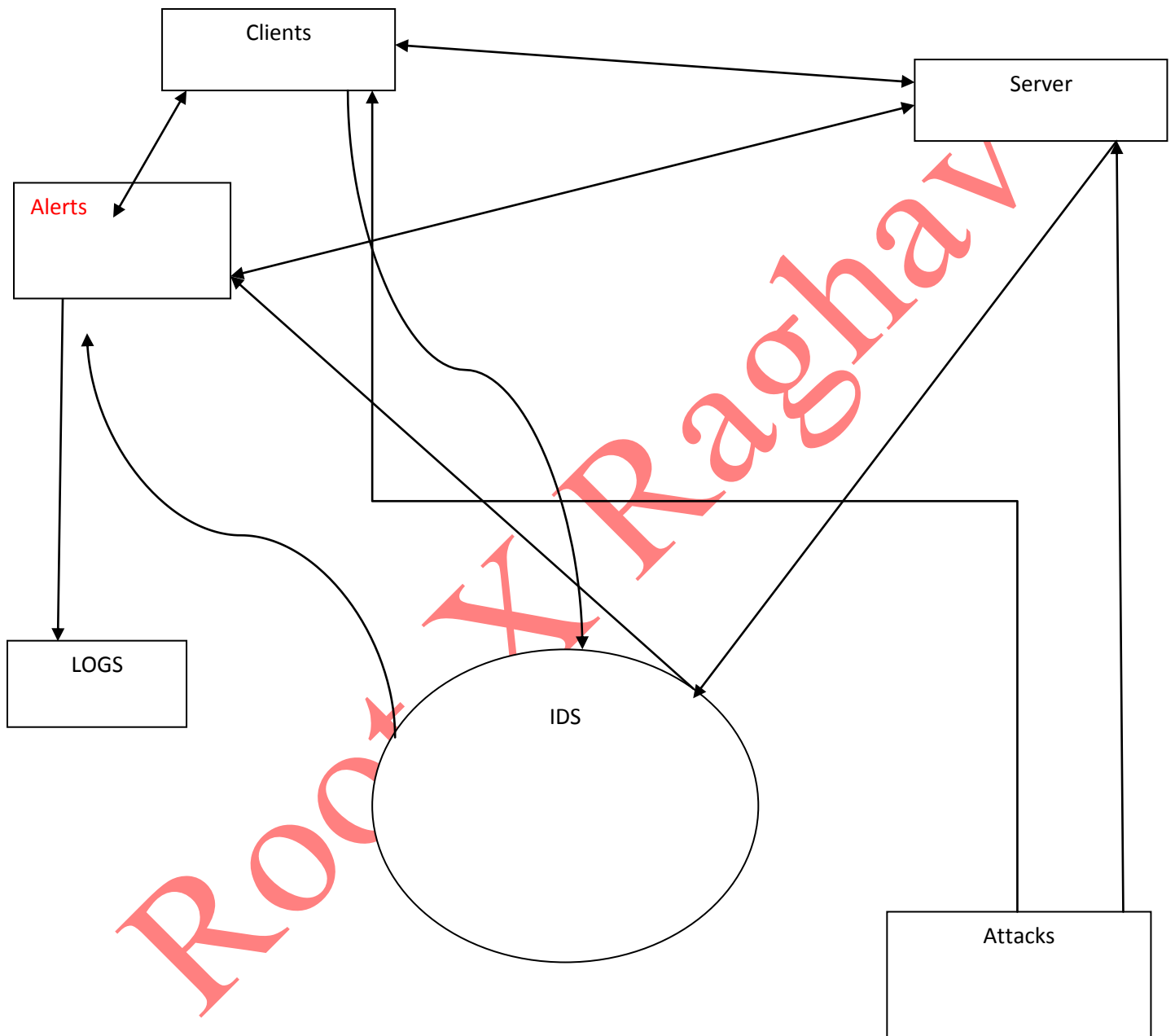
- Install Perl In UNIX Systems By Default Perl Package is installed test using " perl -v " Command.
- Install PerlX-Assert-0.900_01
- Install TCP Dump Program
- Install Wireshark Program
- Install Tshark Programm
- Install Apache Server
- Install airmon-ng Program
- Install airodump-ng Program
- Install airplay-ng Program
- Install airbase-ng Program
- Configure DHCP Server
- Install BrupSute Program
- Install Driftnet Program
- Install hamster & Ferret Program

Tools Description

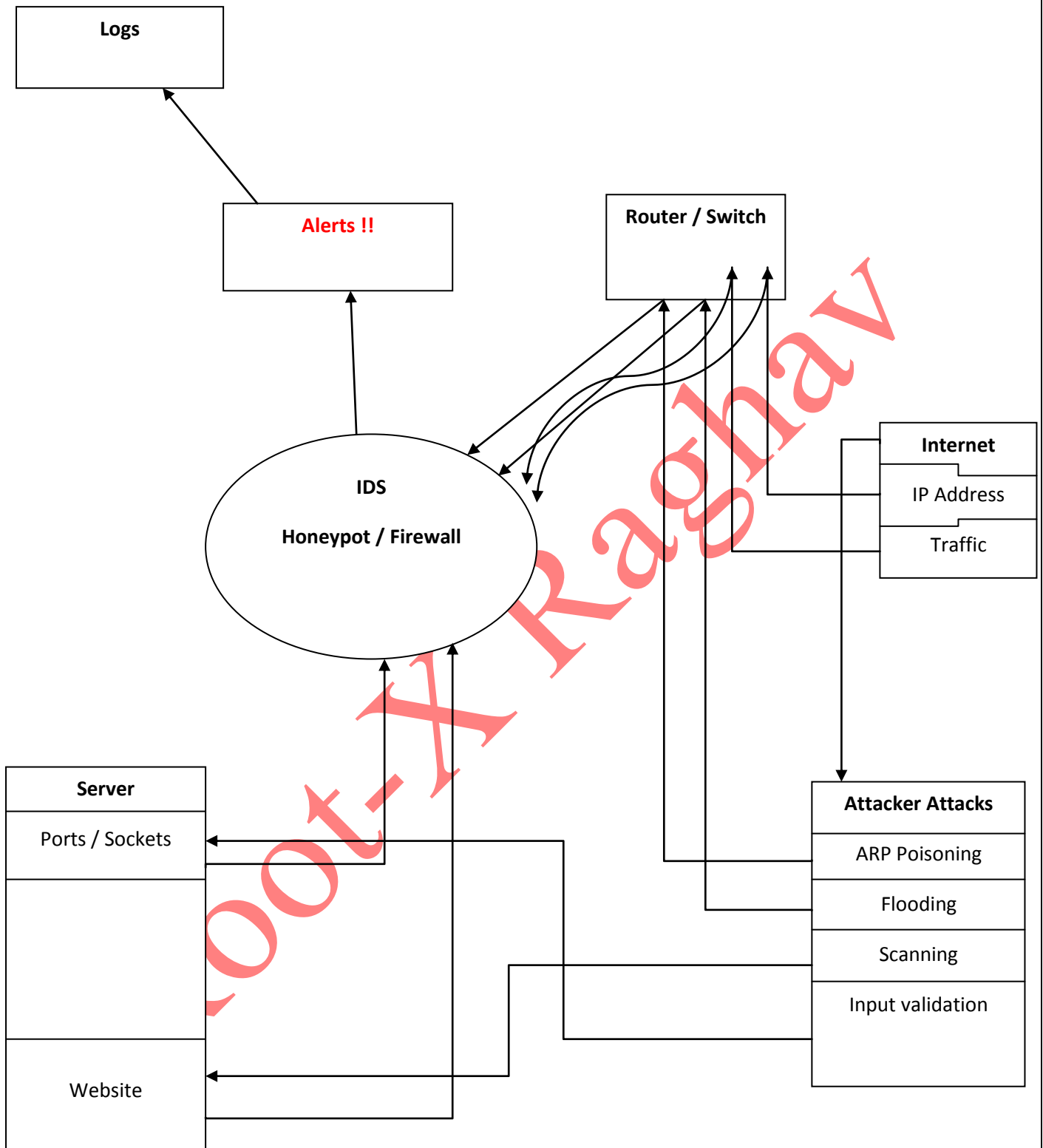
SO.NO.	TOOL	USE
1.	Perl	Programming Language
2.	Wireshark	Packet Sniffing Tool
3.	Tshark	Command Line Packet Sniffing Tool
4.	TCP DUMP	TCP Traffic Intercepting Tool
5.	Apache Server	HTTPD Server to host Websites
6.	Airmon-ng	To Put wireless network card at monitor mode.
7.	Airodump-ng	To Dump All Wireless connection detail
8.	Airplay-ng	To do ARP Poisoning Attack on Base Stations
9.	Airbase-ng	To Create a New Base Station
10.	DHCP Server	To configure Networking such as IP-Address Ranges
11.	BrupSute	To Intercept Session Of Users
12.	Driftnet	To Sniff Images form captured packets
13.	Hamster & Ferret	To Sniff Documents from captured packets

6. Detail Design

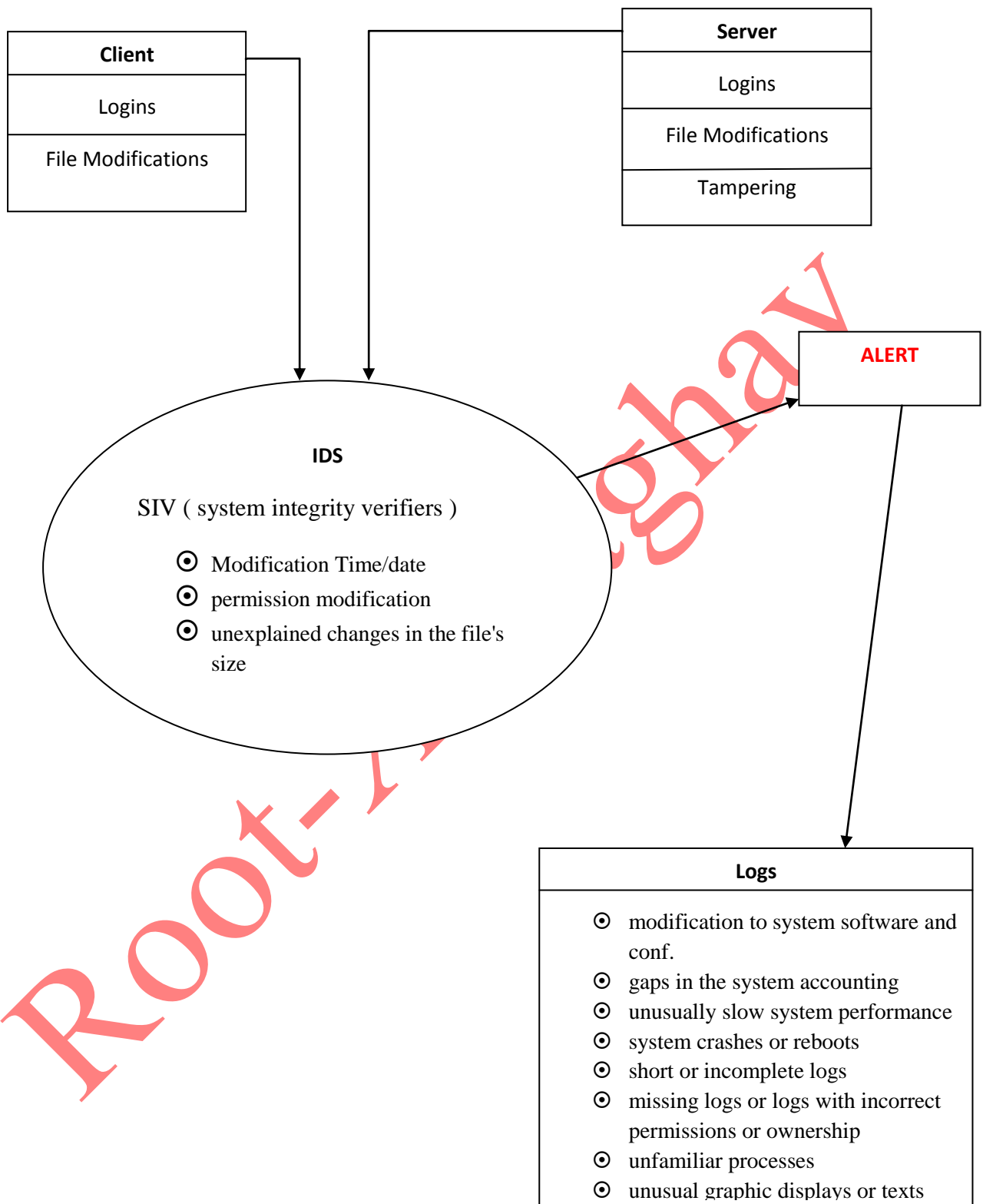
DFD For IDS Overview



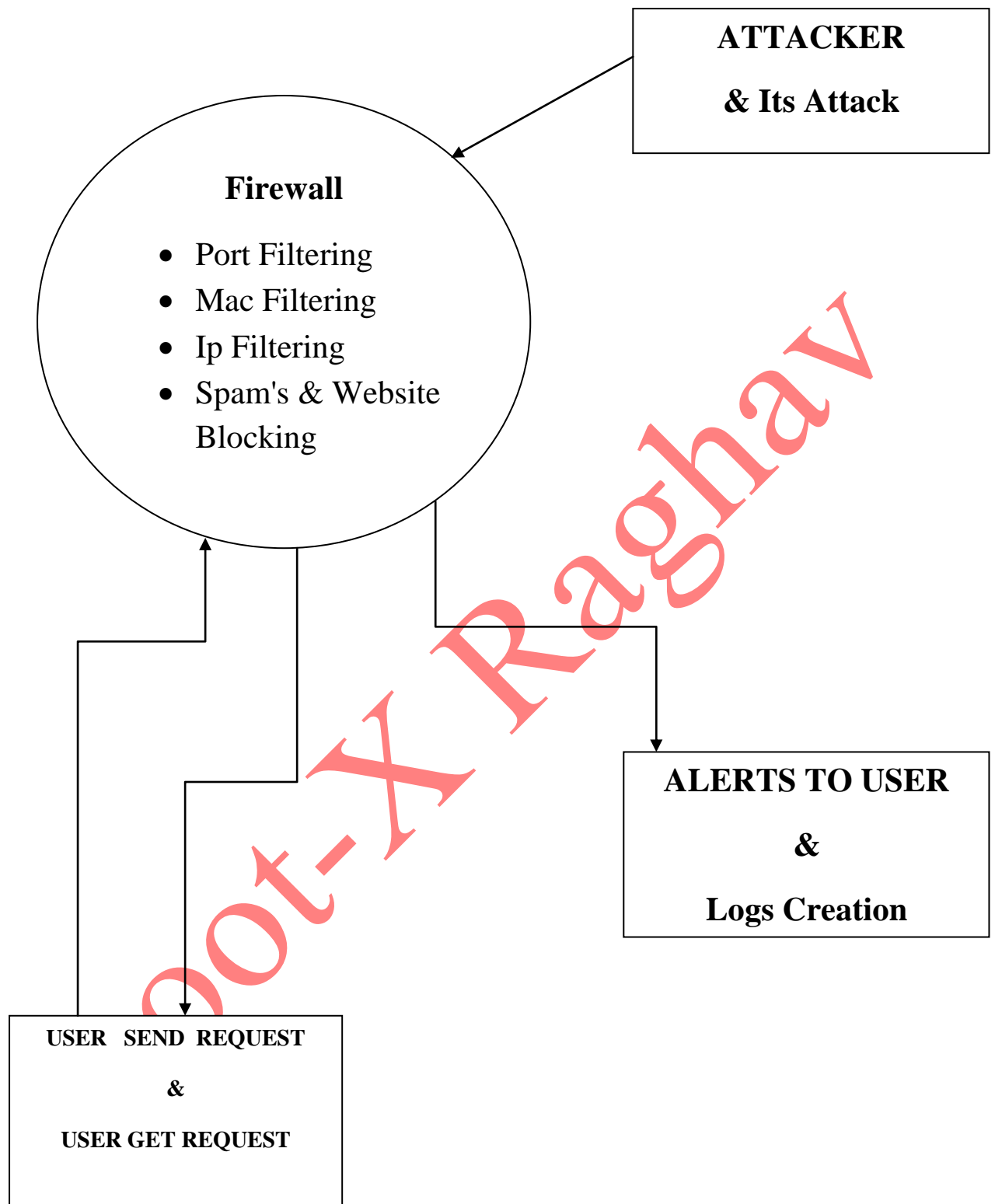
DFD For NIDS A.K.A Network-Based Intrusion Detection System



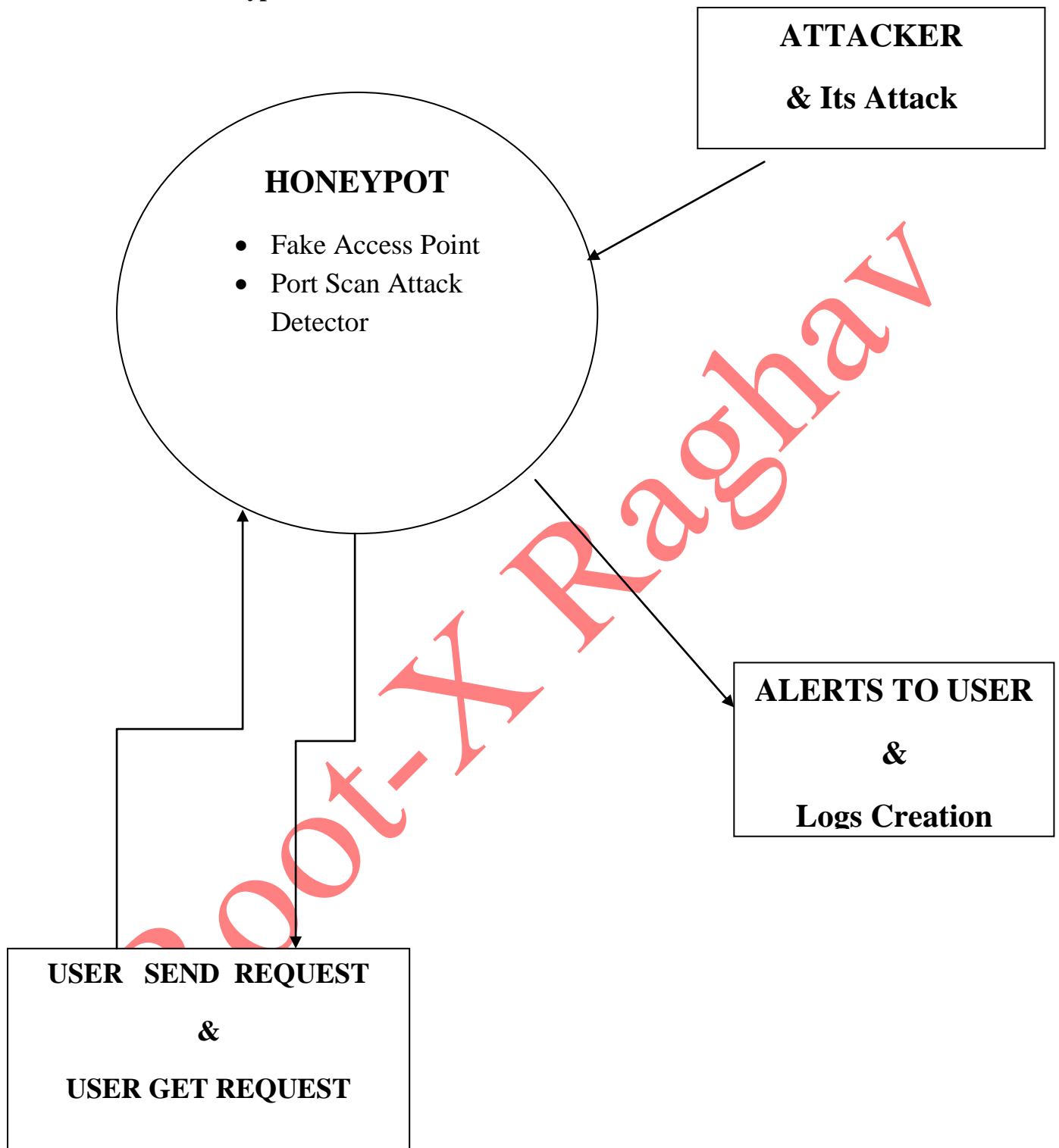
DFD For HIDS A.K.A Host-Based Intrusion Detection System



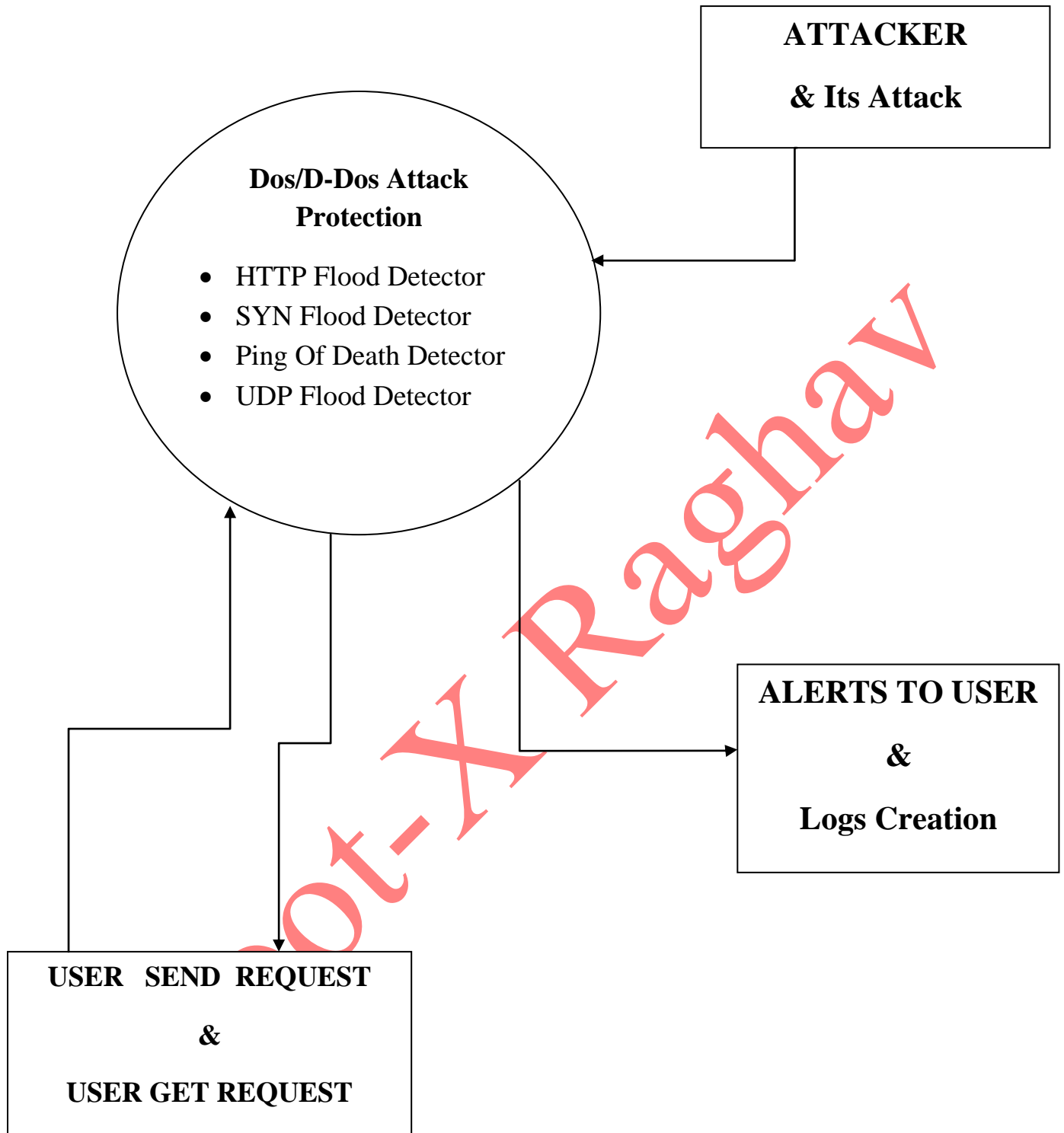
Module 1 : Firewall



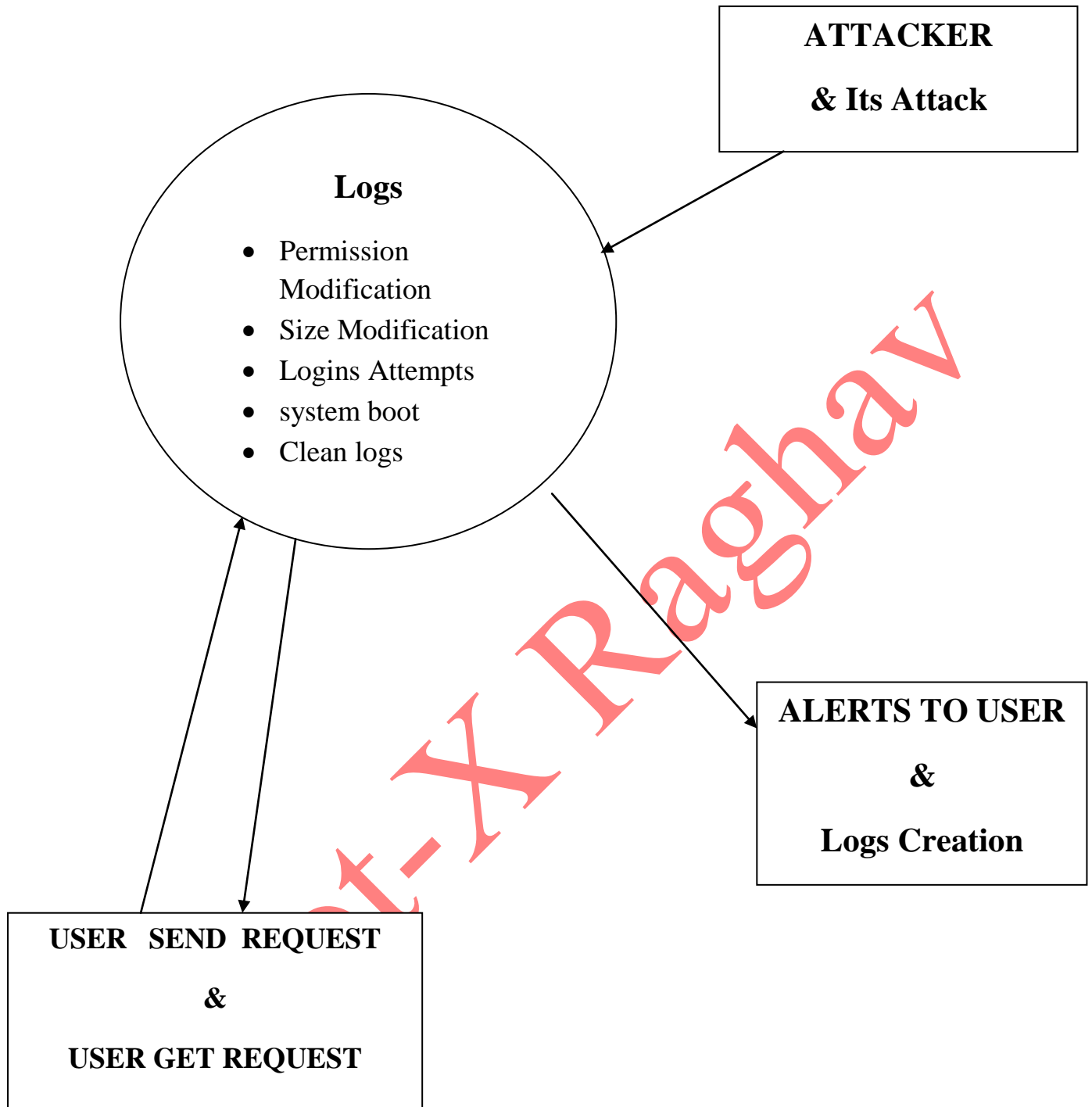
Module 2 : Honeypot



Module 3 : Dos-D-Dos Attack

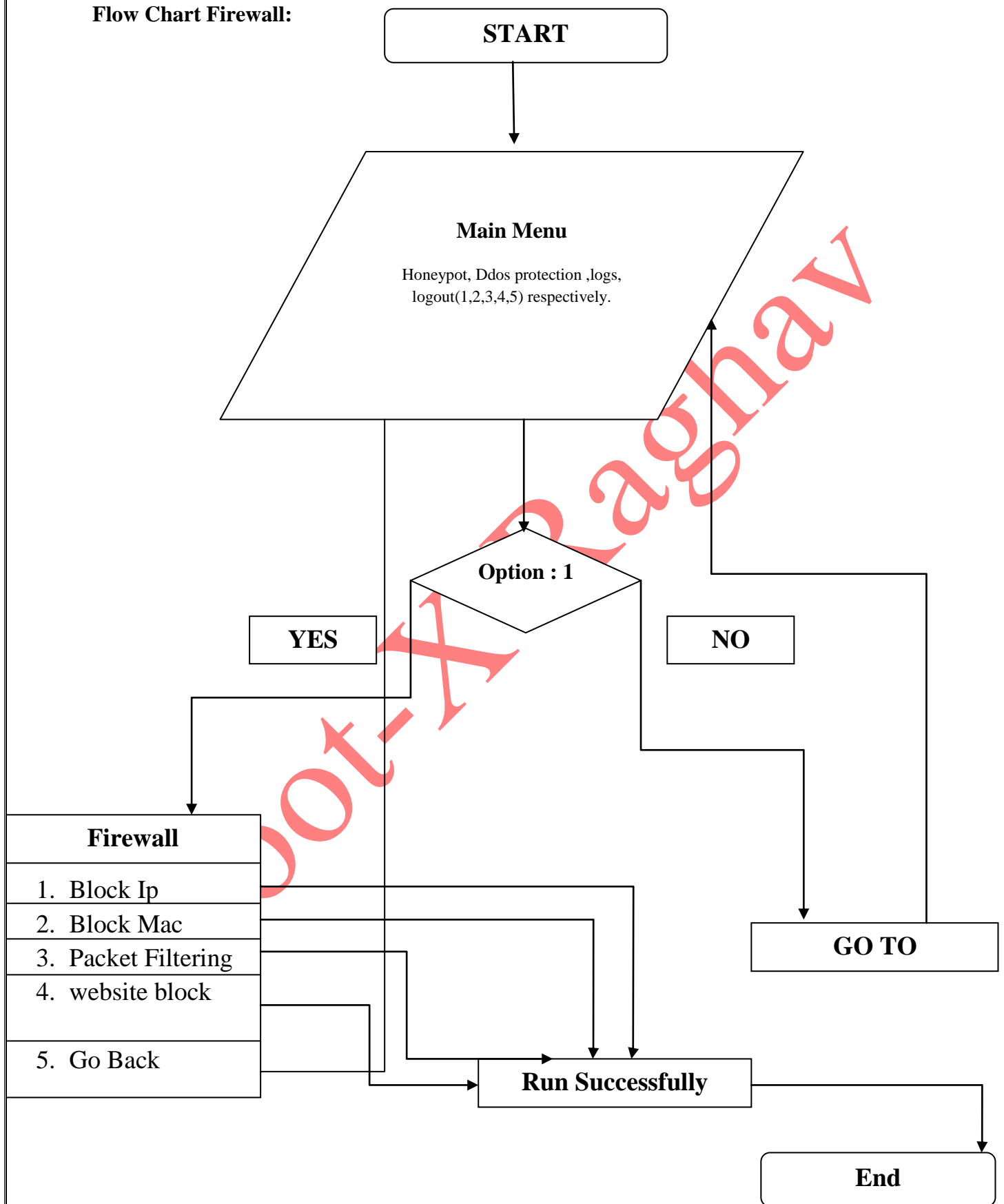


Module 3 : Logs

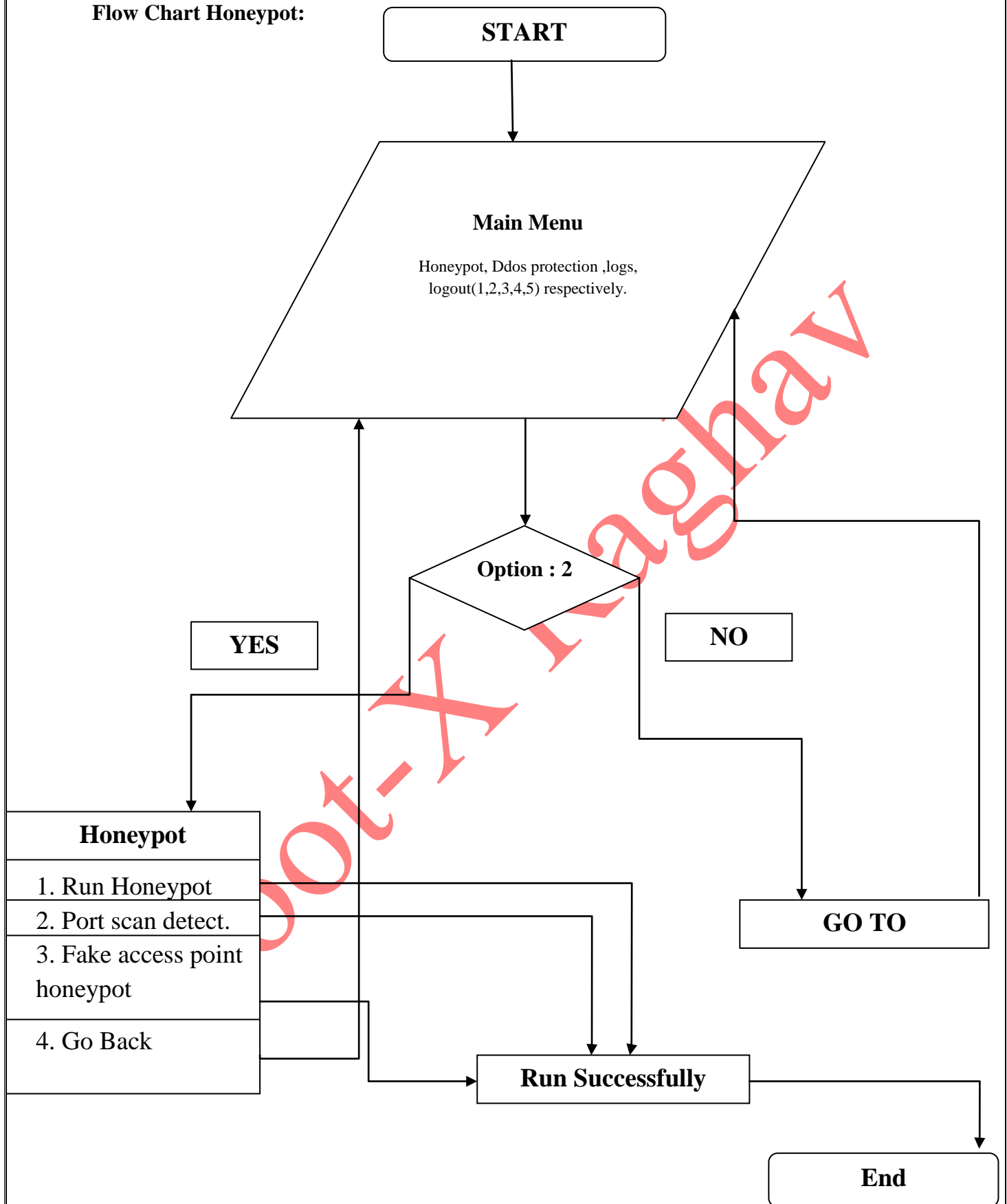


3. Flow Charts :

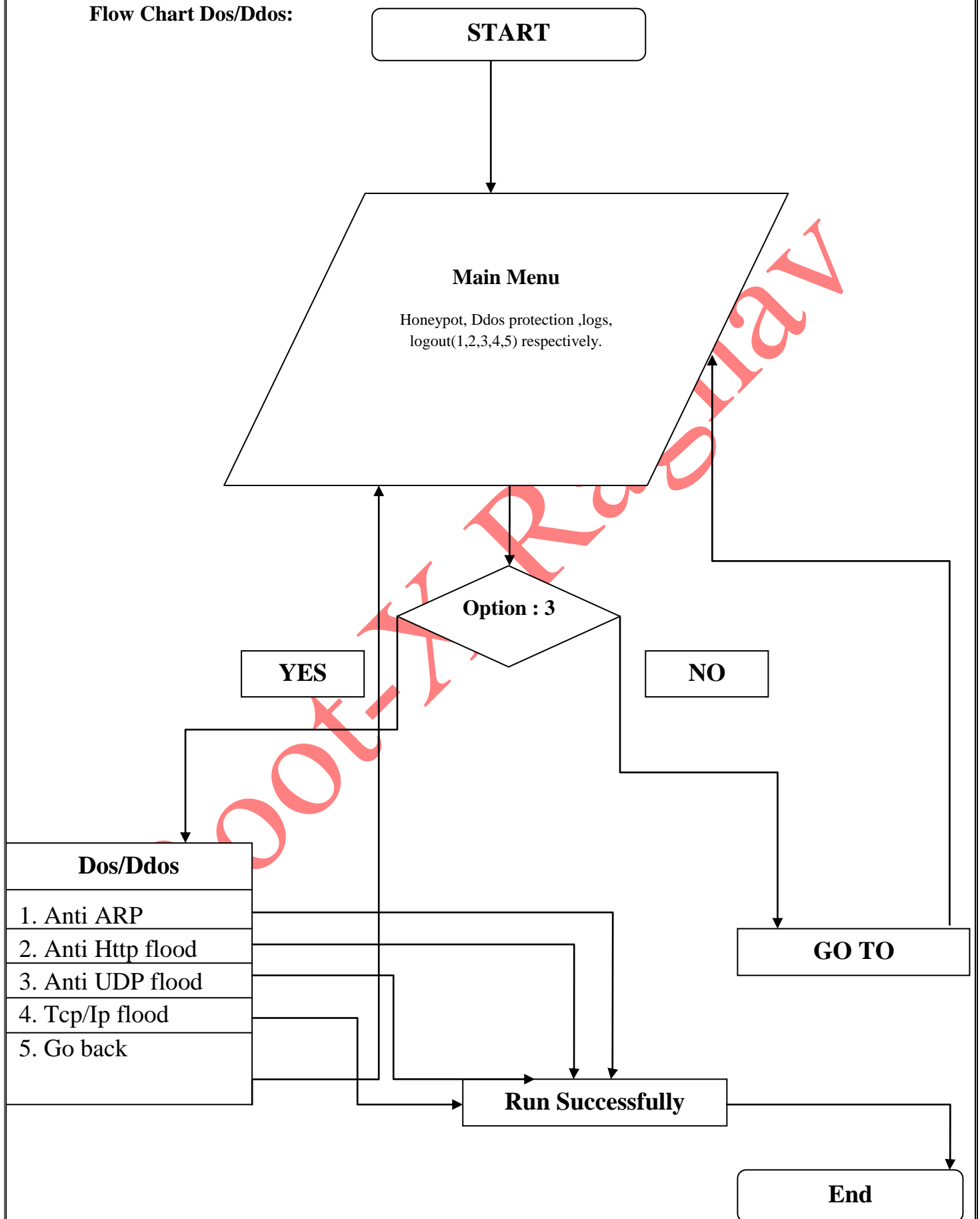
Flow Chart Firewall:



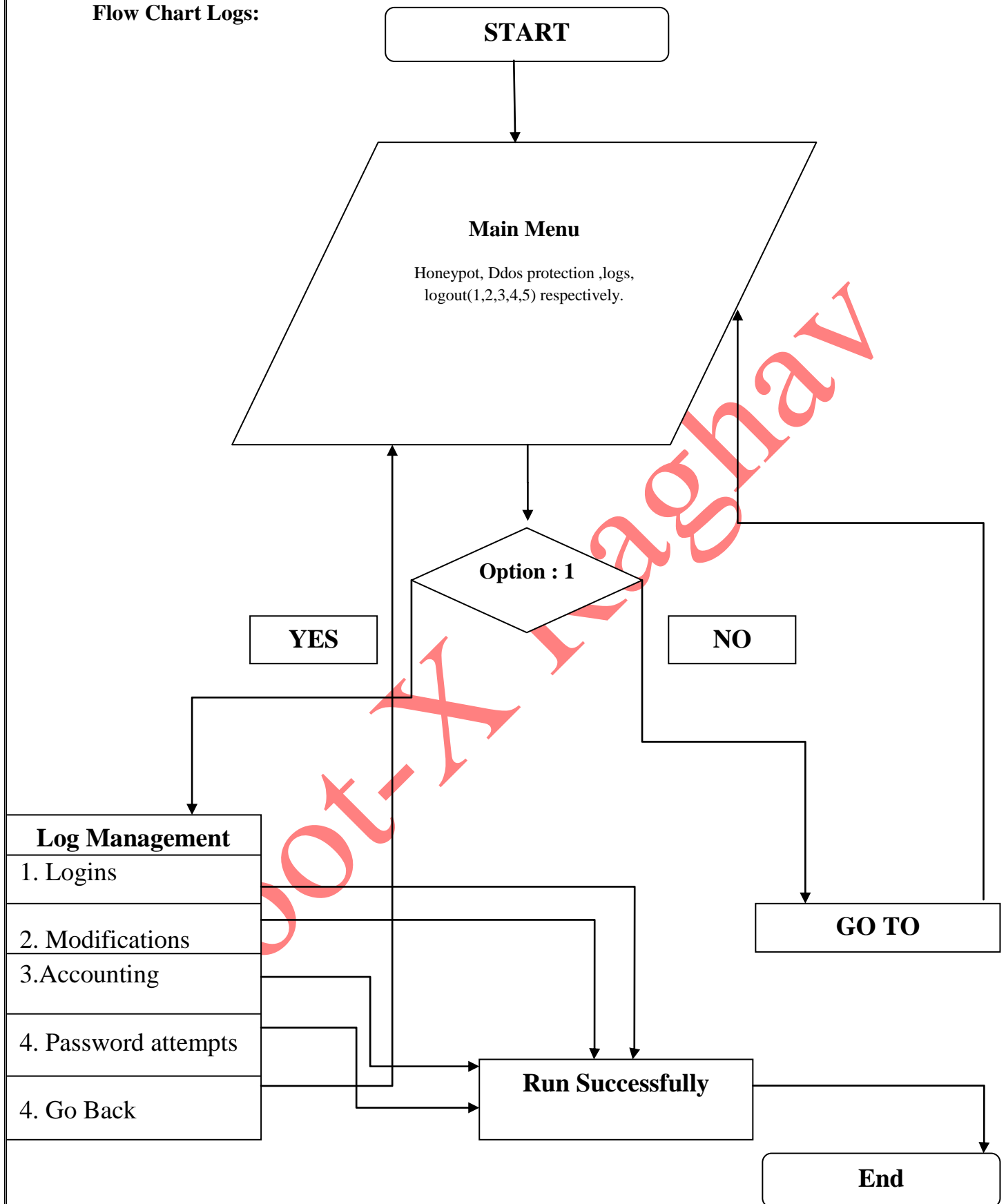
Flow Chart Honeypot:



Flow Chart Dos/Ddos:

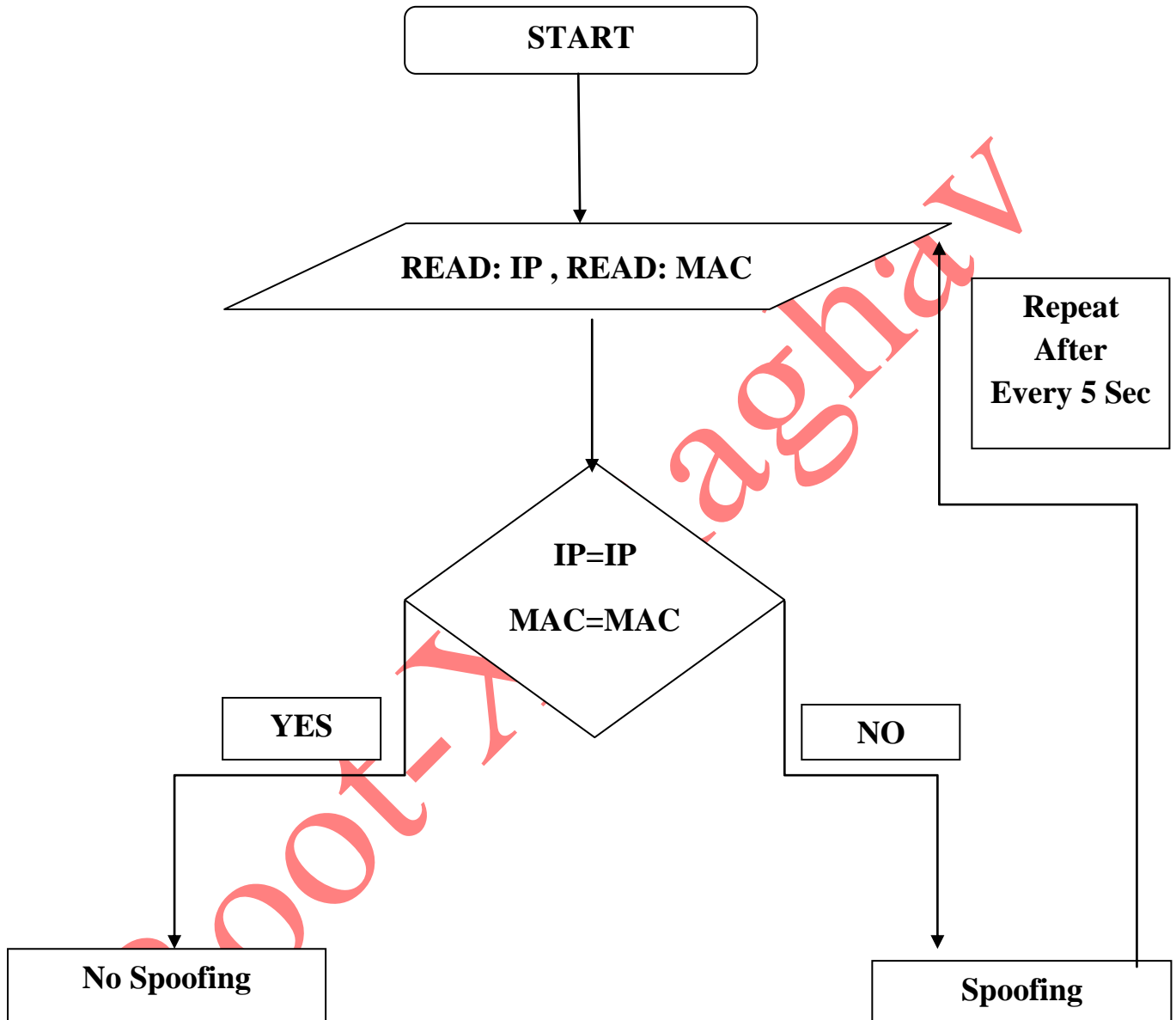


Flow Chart Logs:



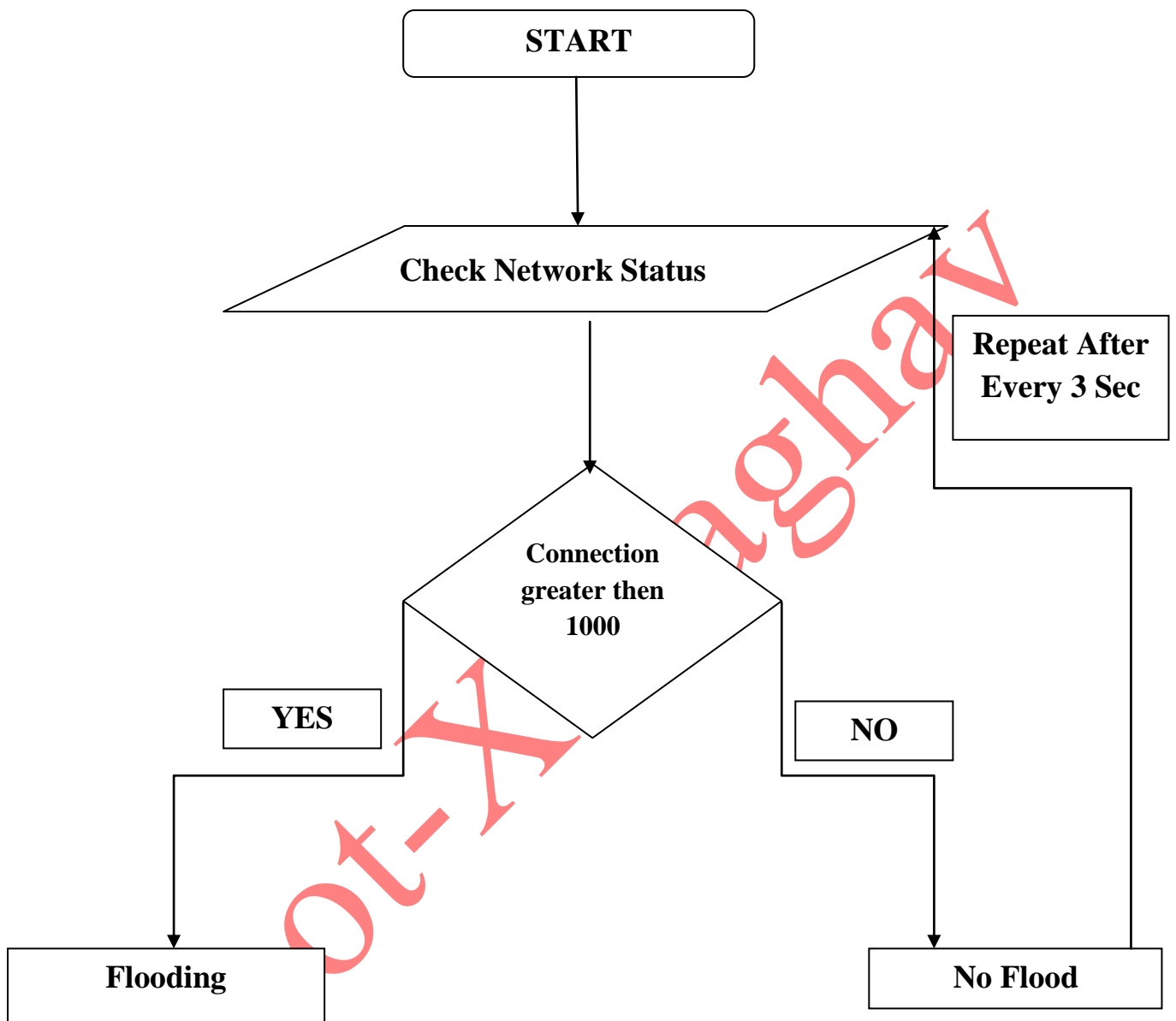
Anti ARP spoofing detection

Flow Chart :



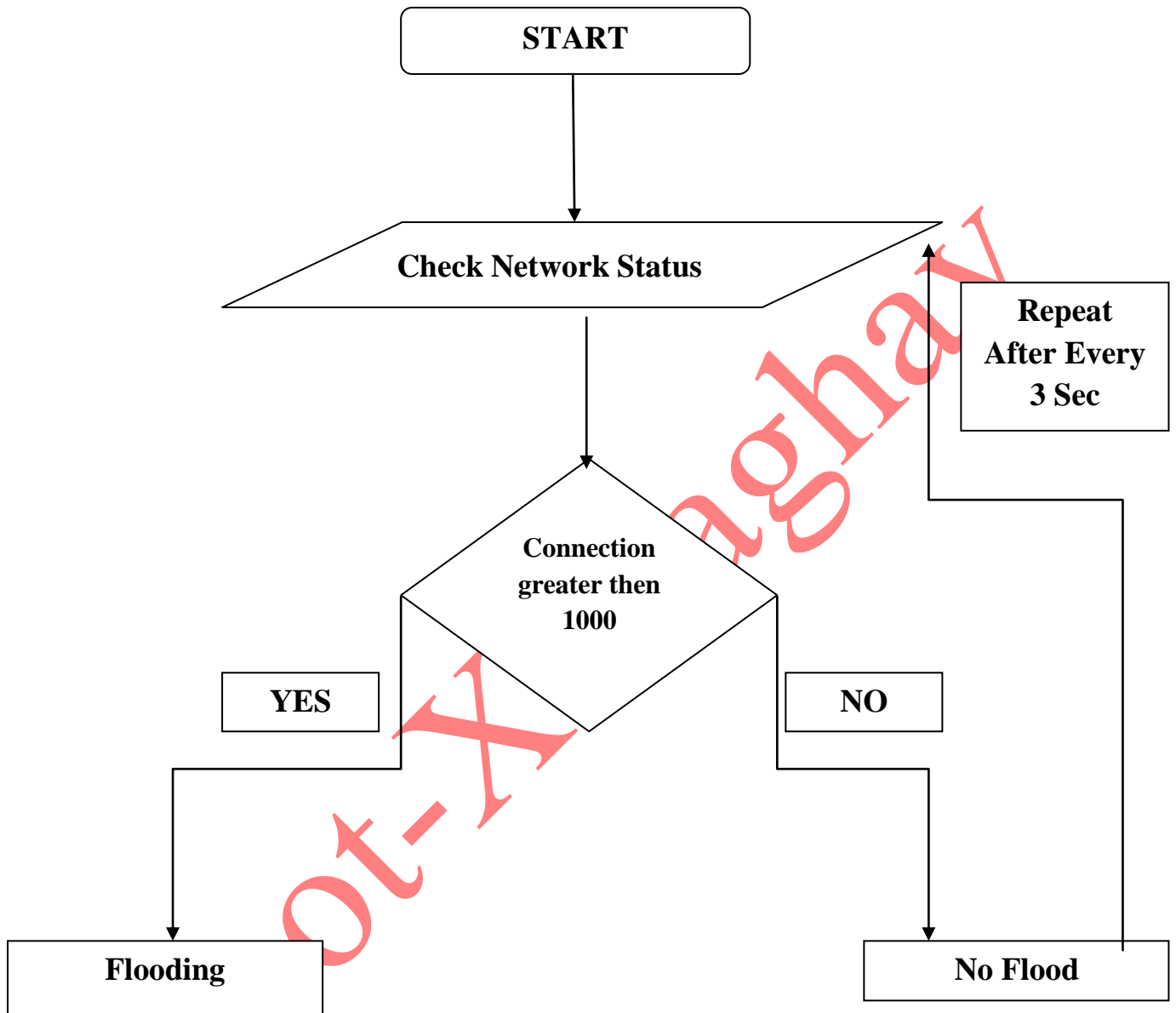
Tcp / Ip Flood detection

Flow Chart:



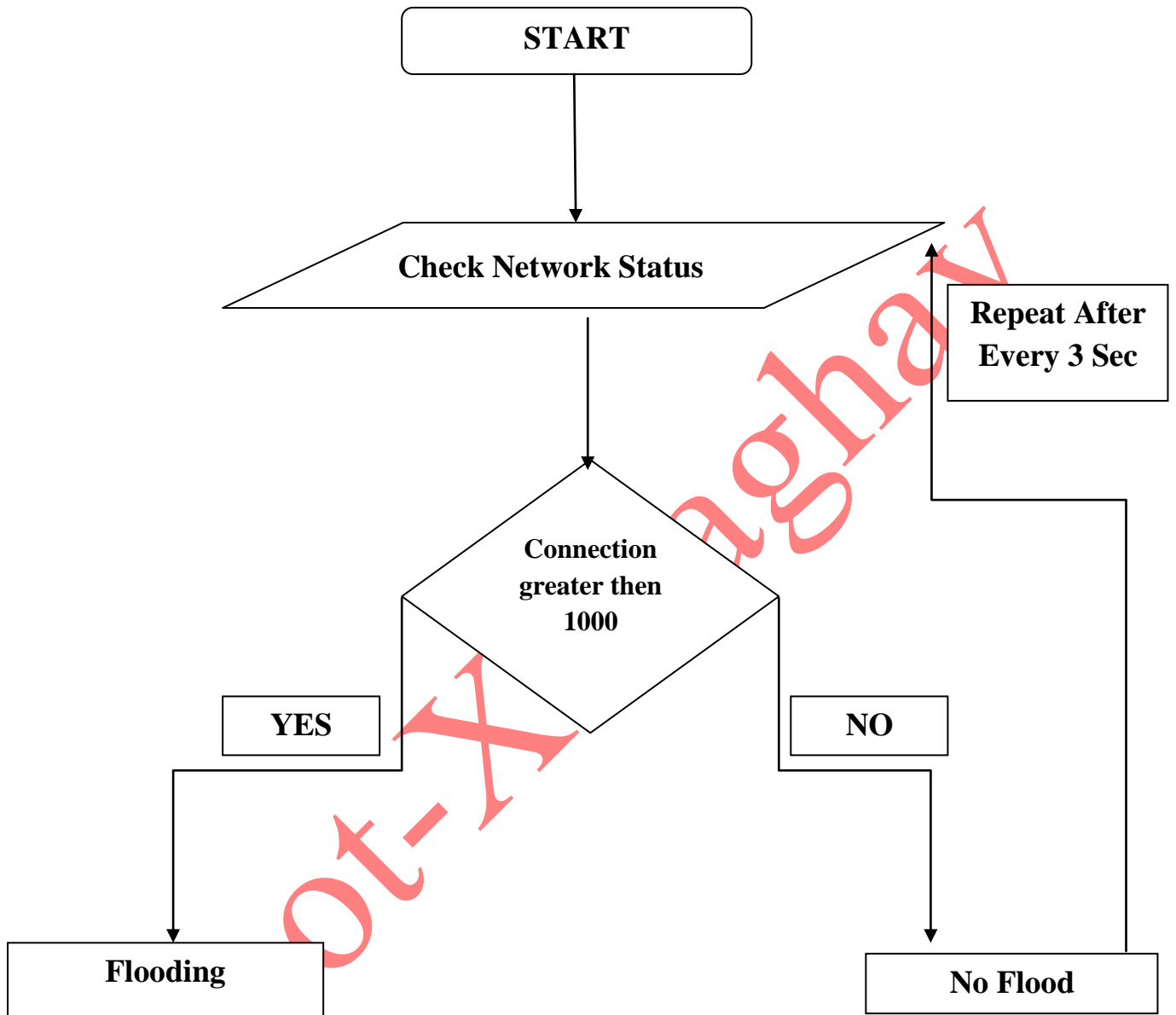
UDP Flood detection

Flow Chart:



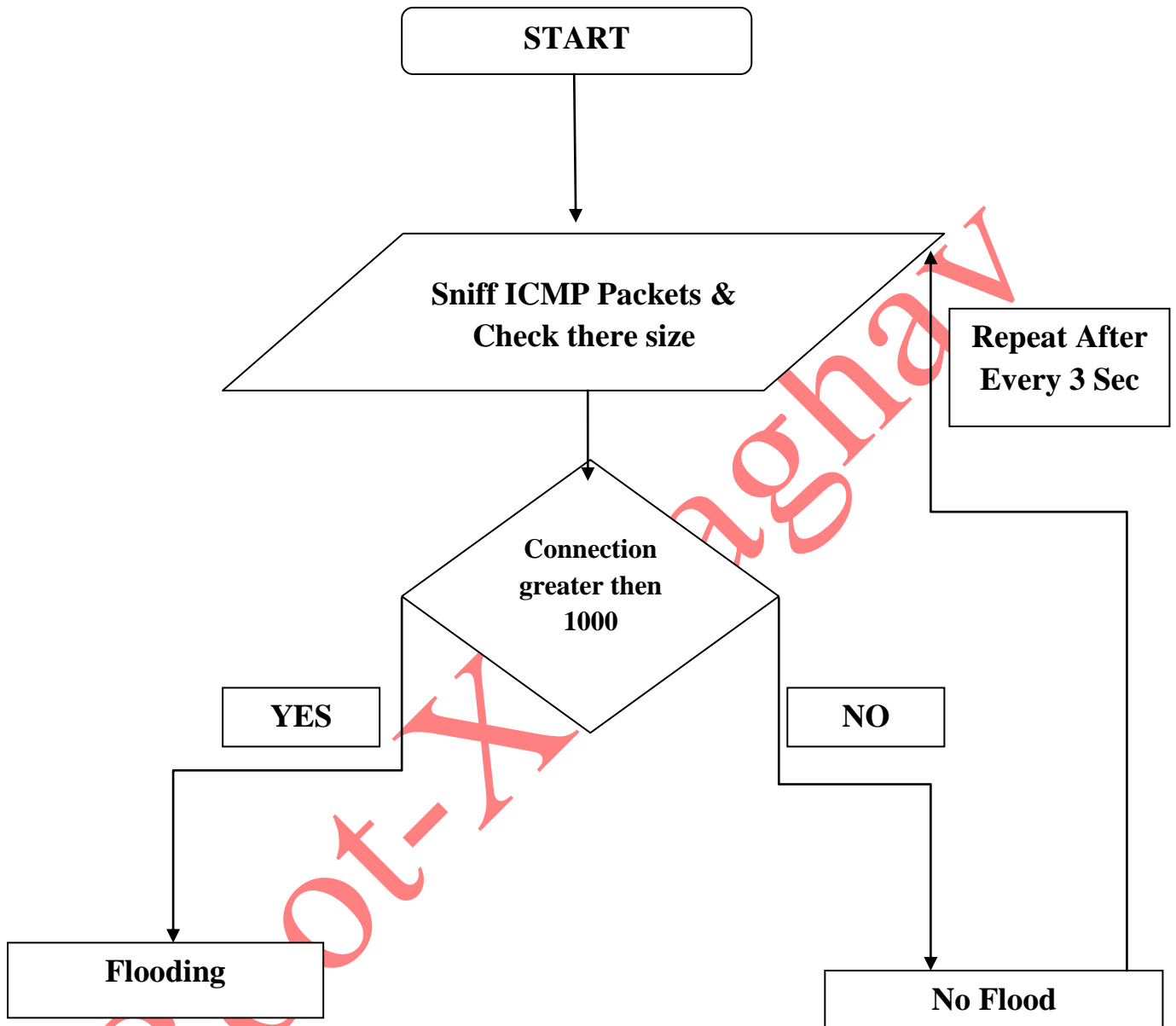
SYN Flood detection

Flow Chart:



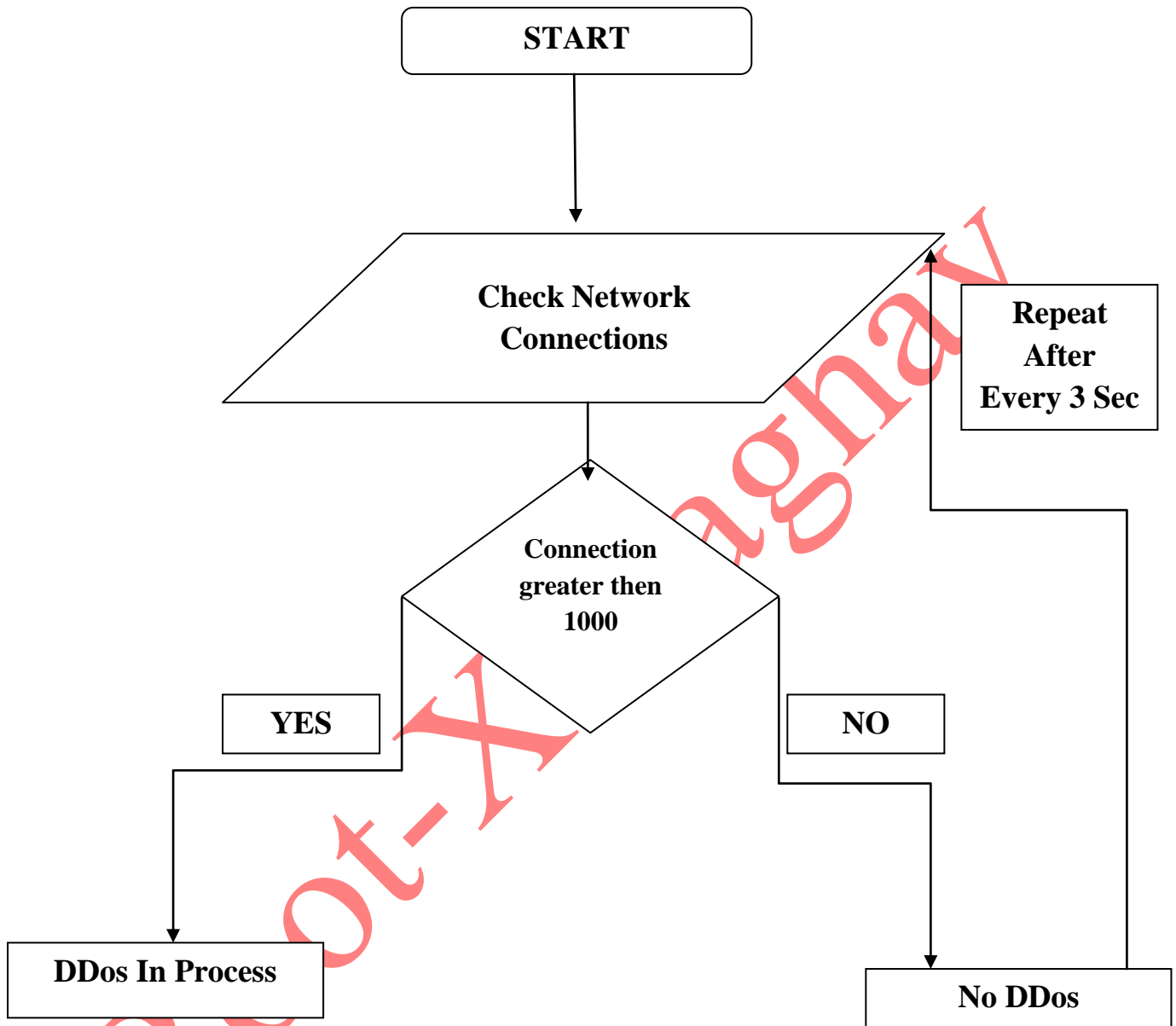
Ping of Death A.K.A ICMP Flood detection

Flow Chart:



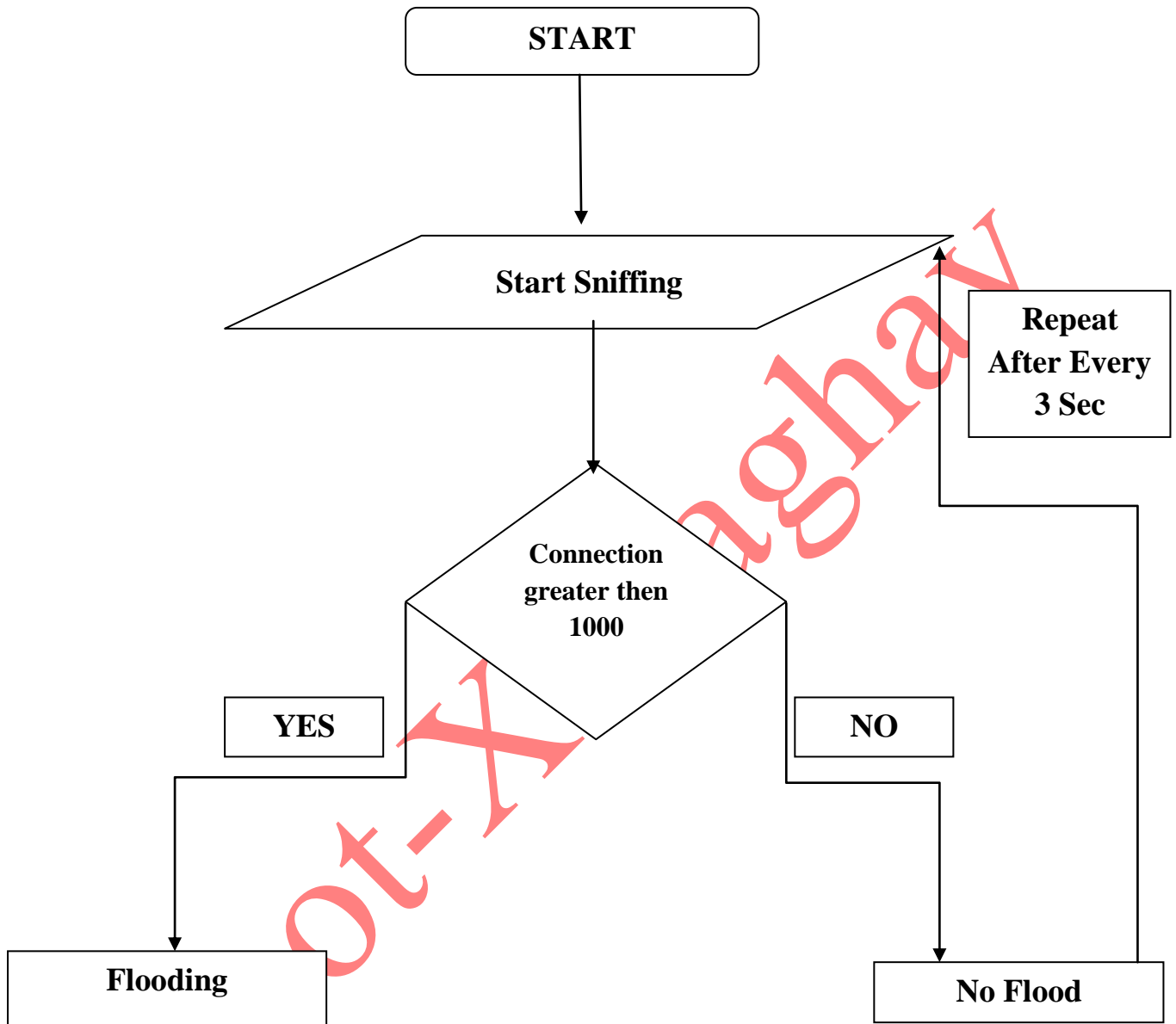
Established Connection Attack

Flow Chart:



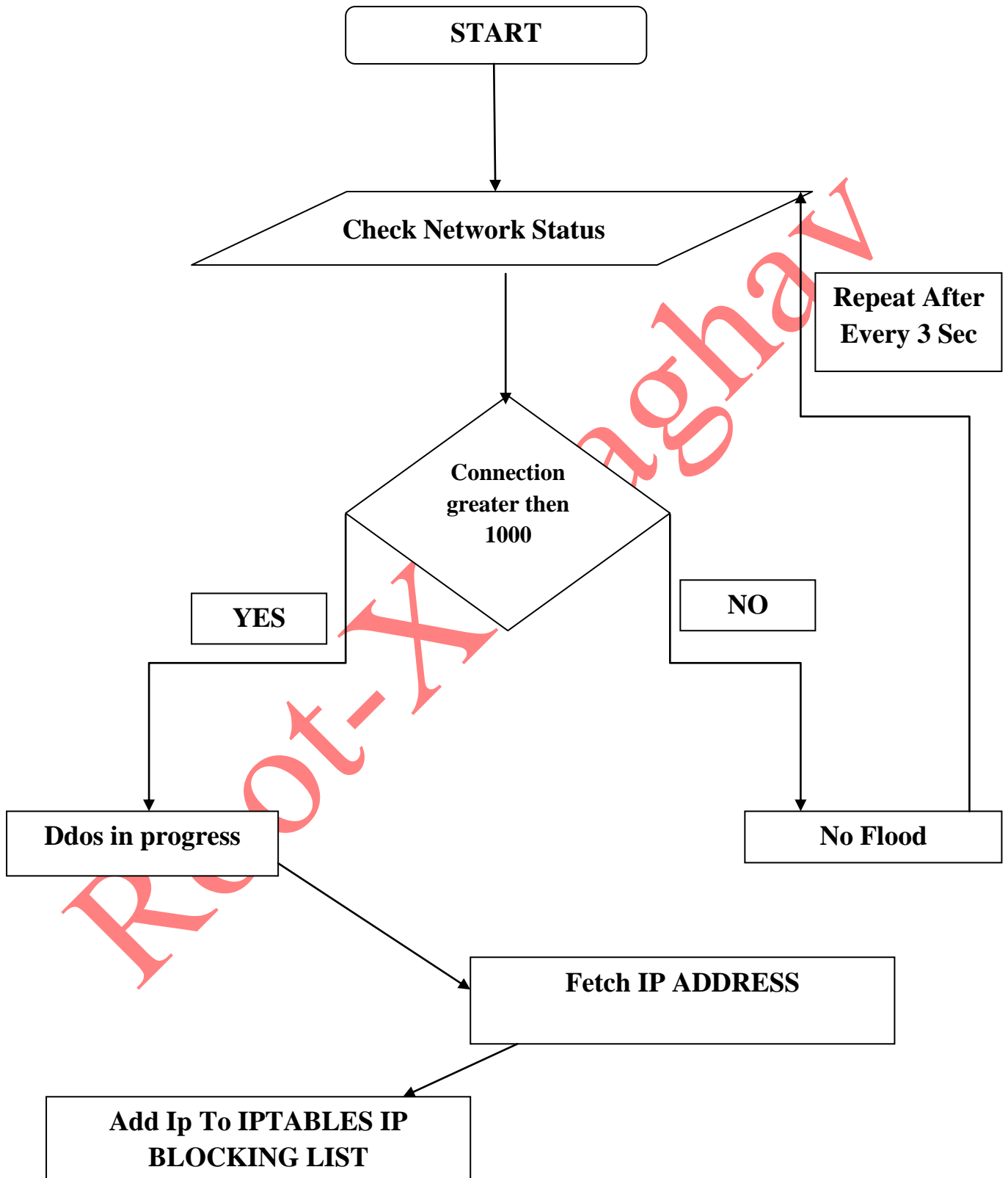
0-63055 Ports Flood detection

Flow Chart:



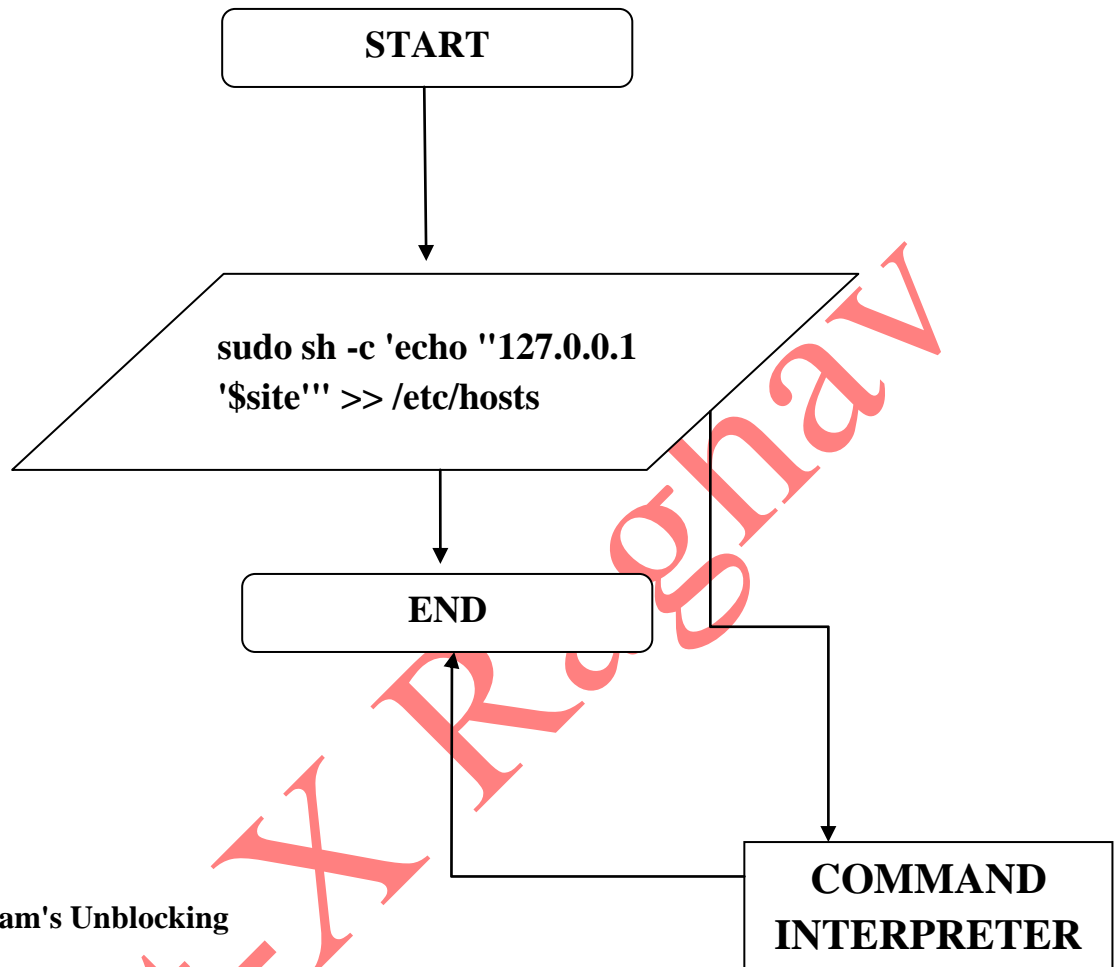
DDos Attack Blocker

Flow Chart:



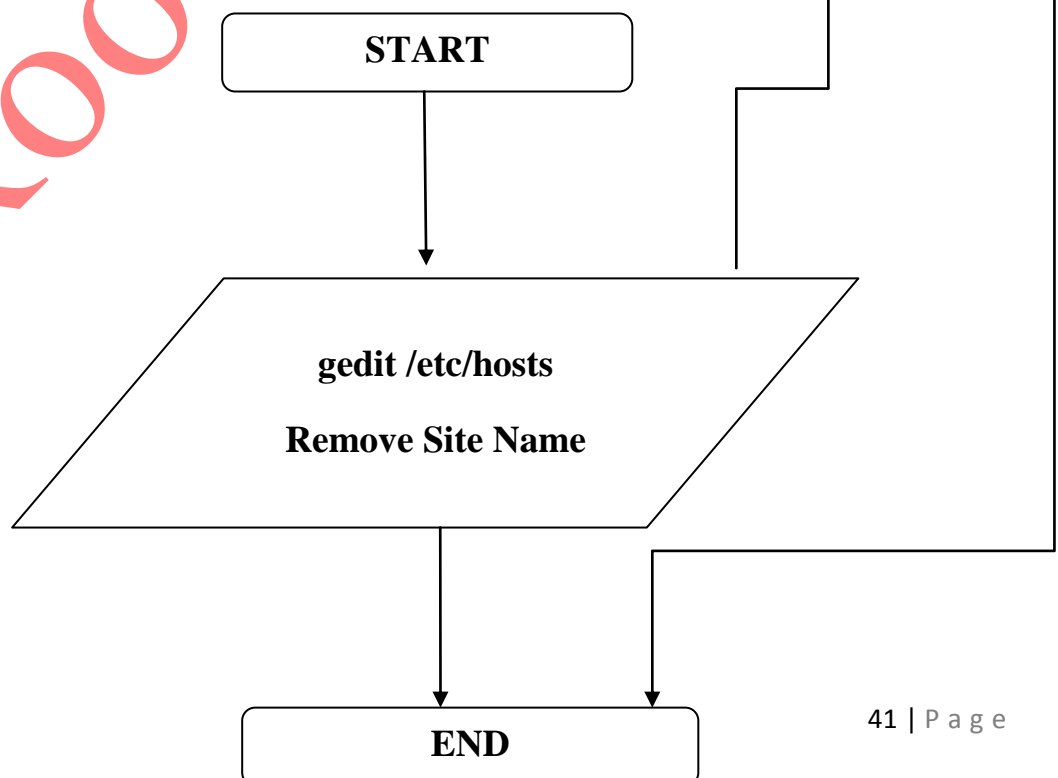
Website & Spams Blocker

Flow Chart:



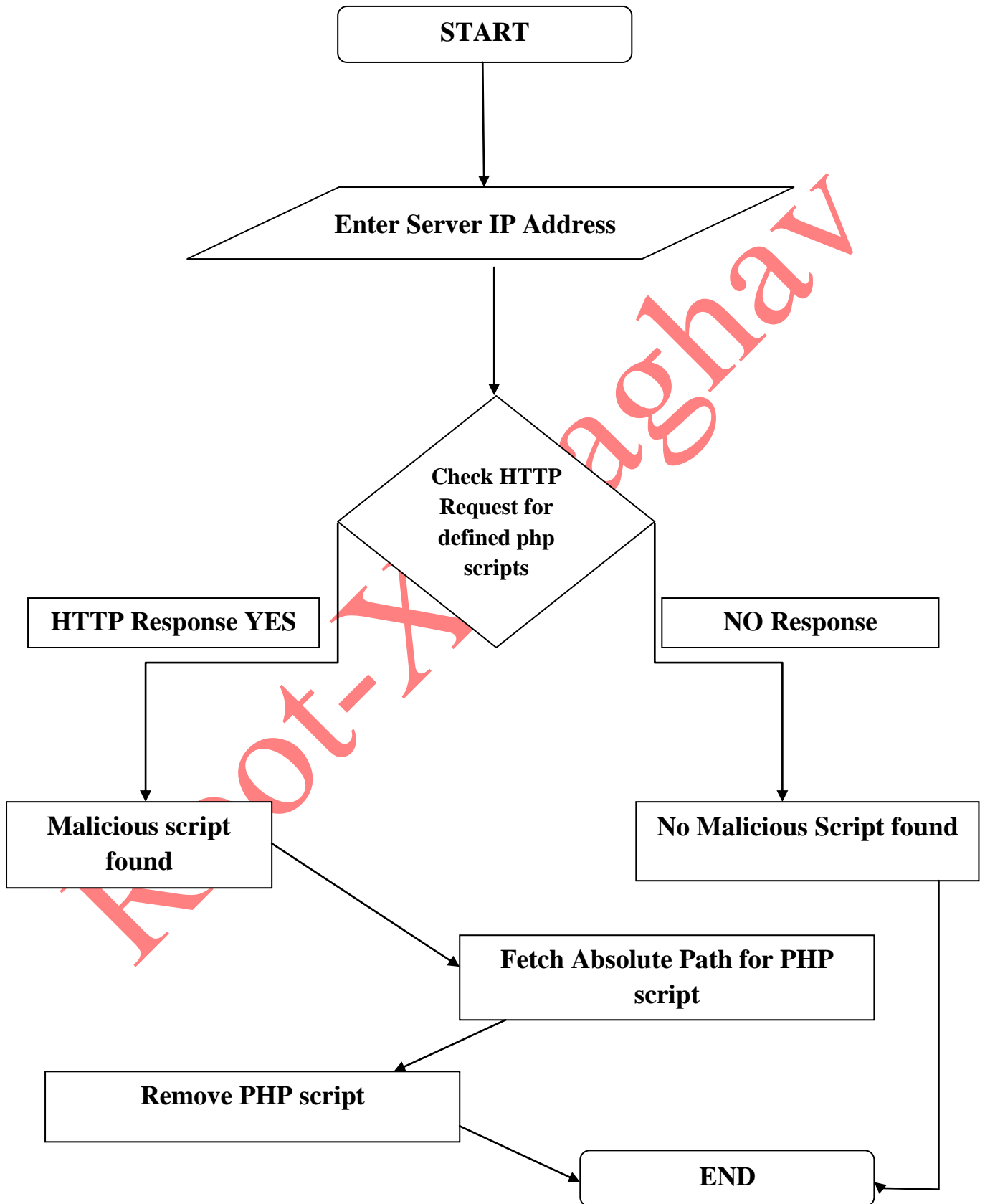
Website & Spam's Unblocking

Flow Chart:



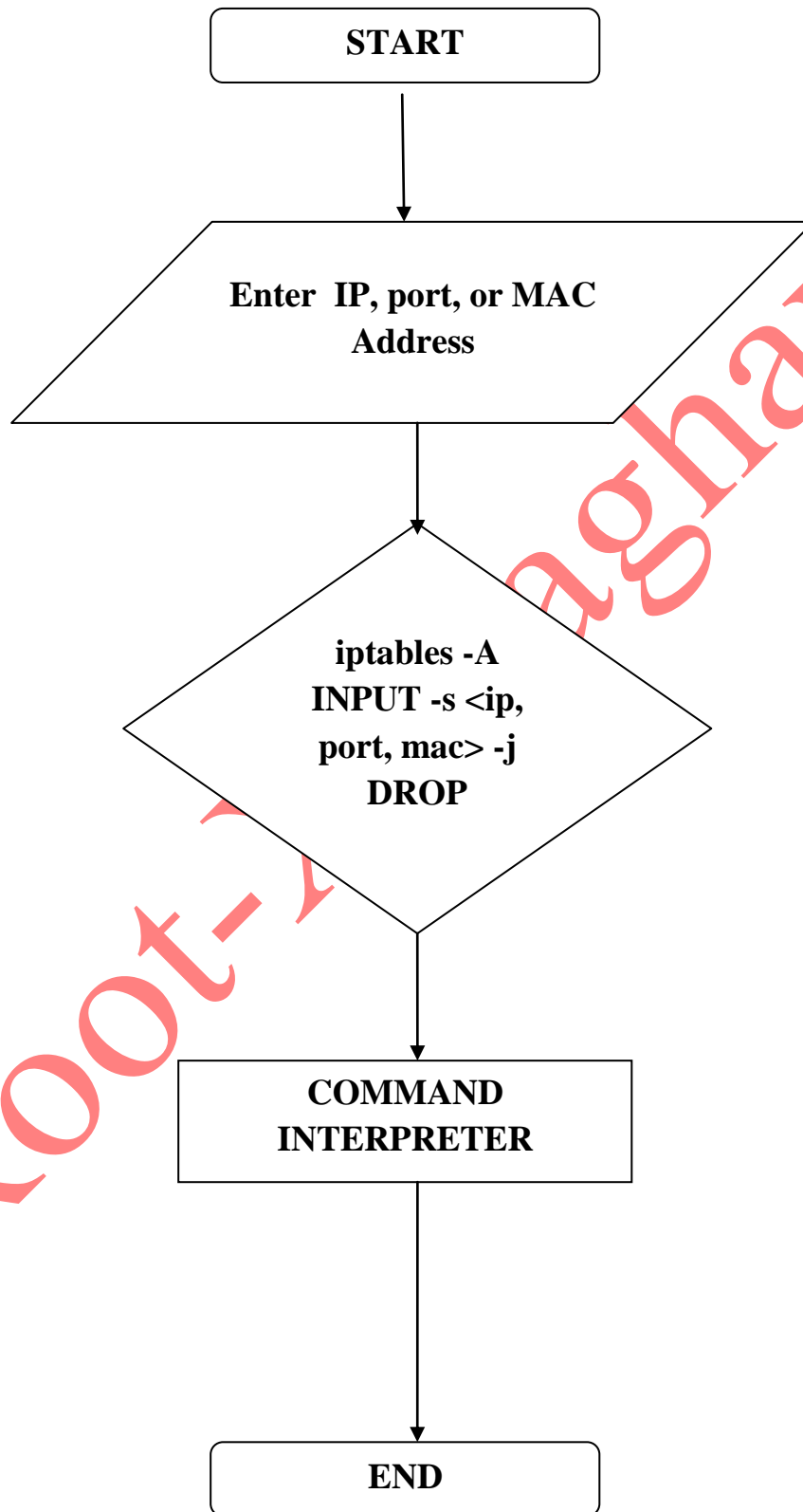
Malicious PHP Scripts Finder

Flow Chart:



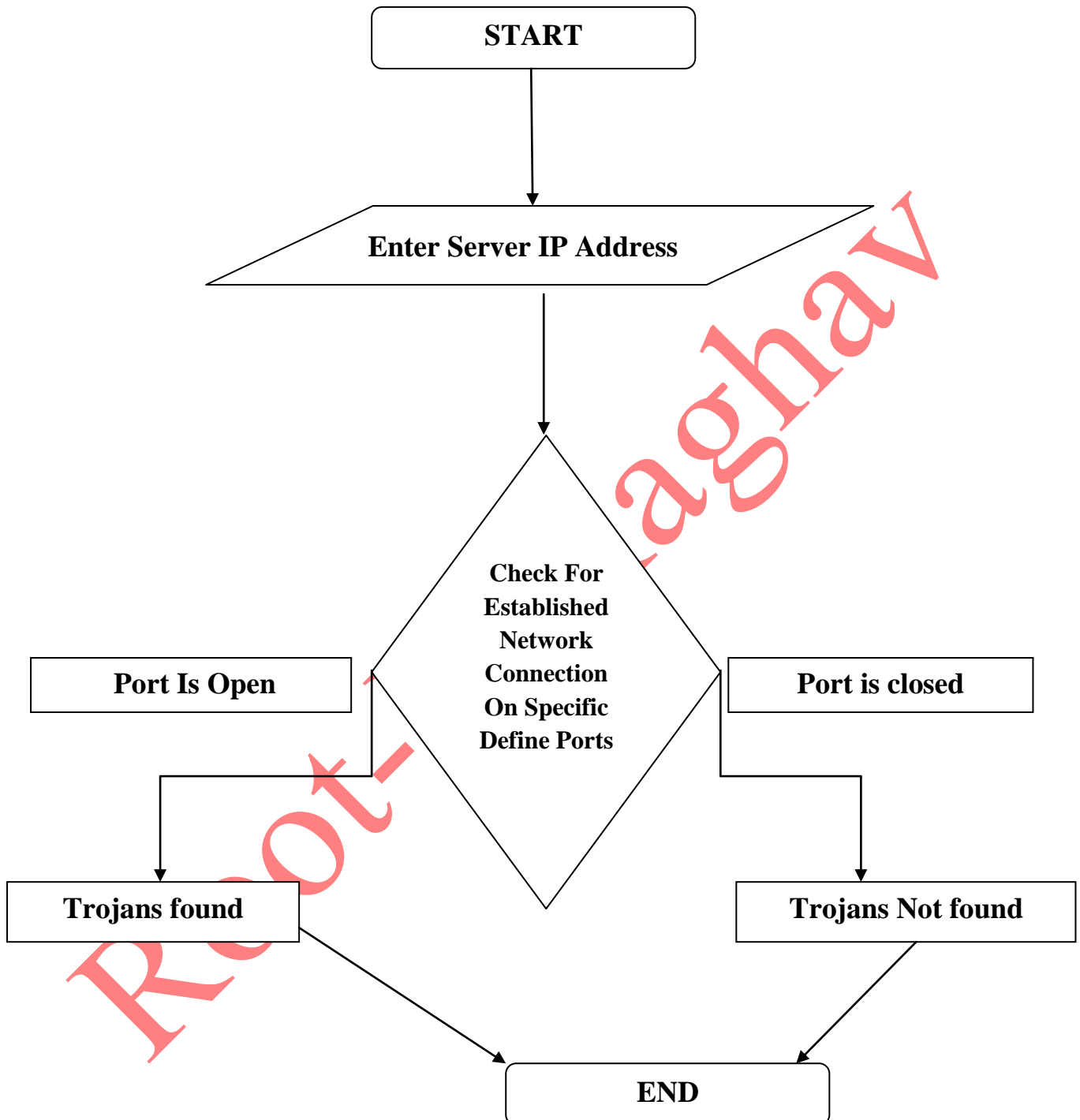
IP Address, Mac Address & Port Blocking

Flow Chart:



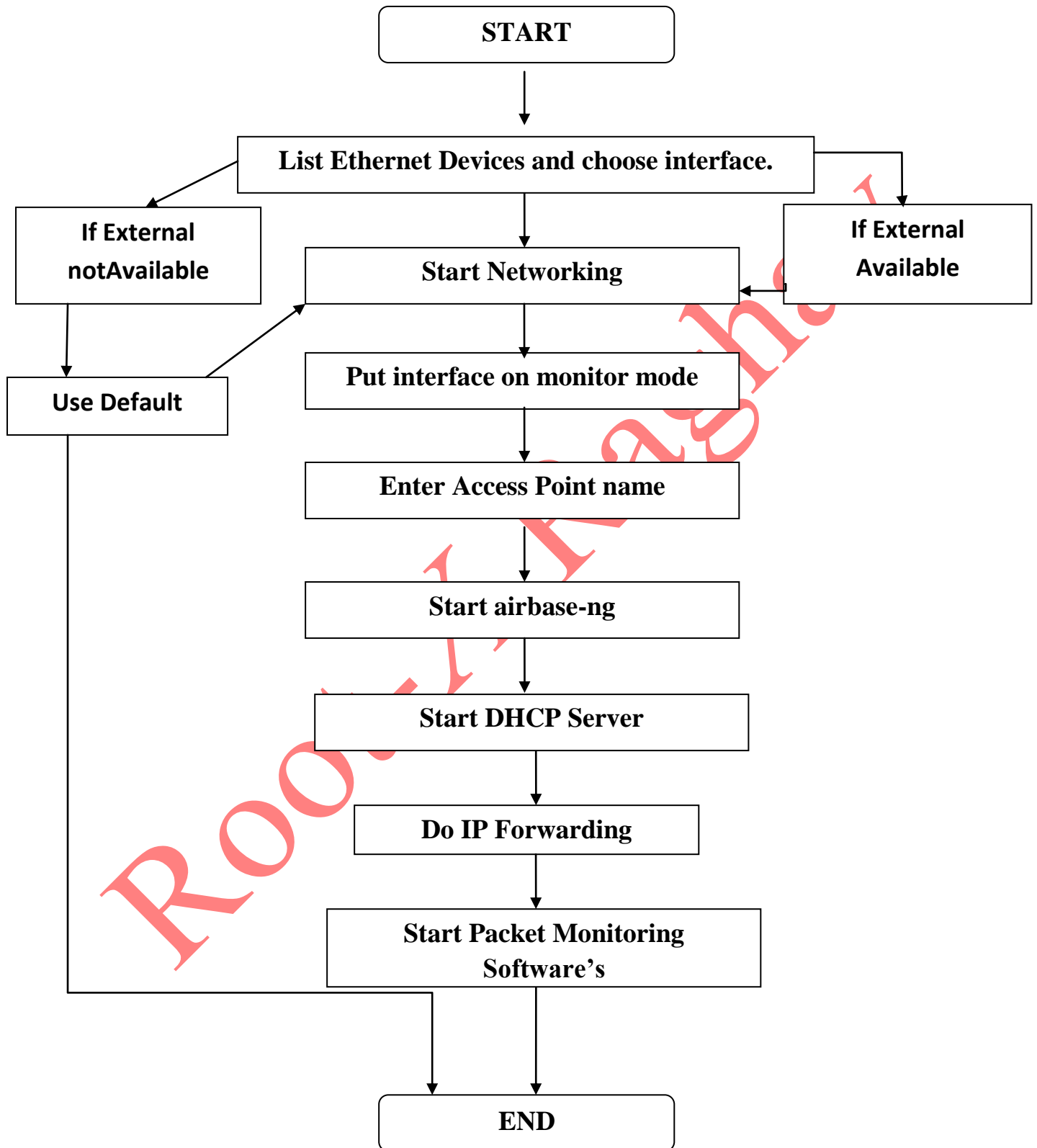
Trojans Scanner

Flow Chart:



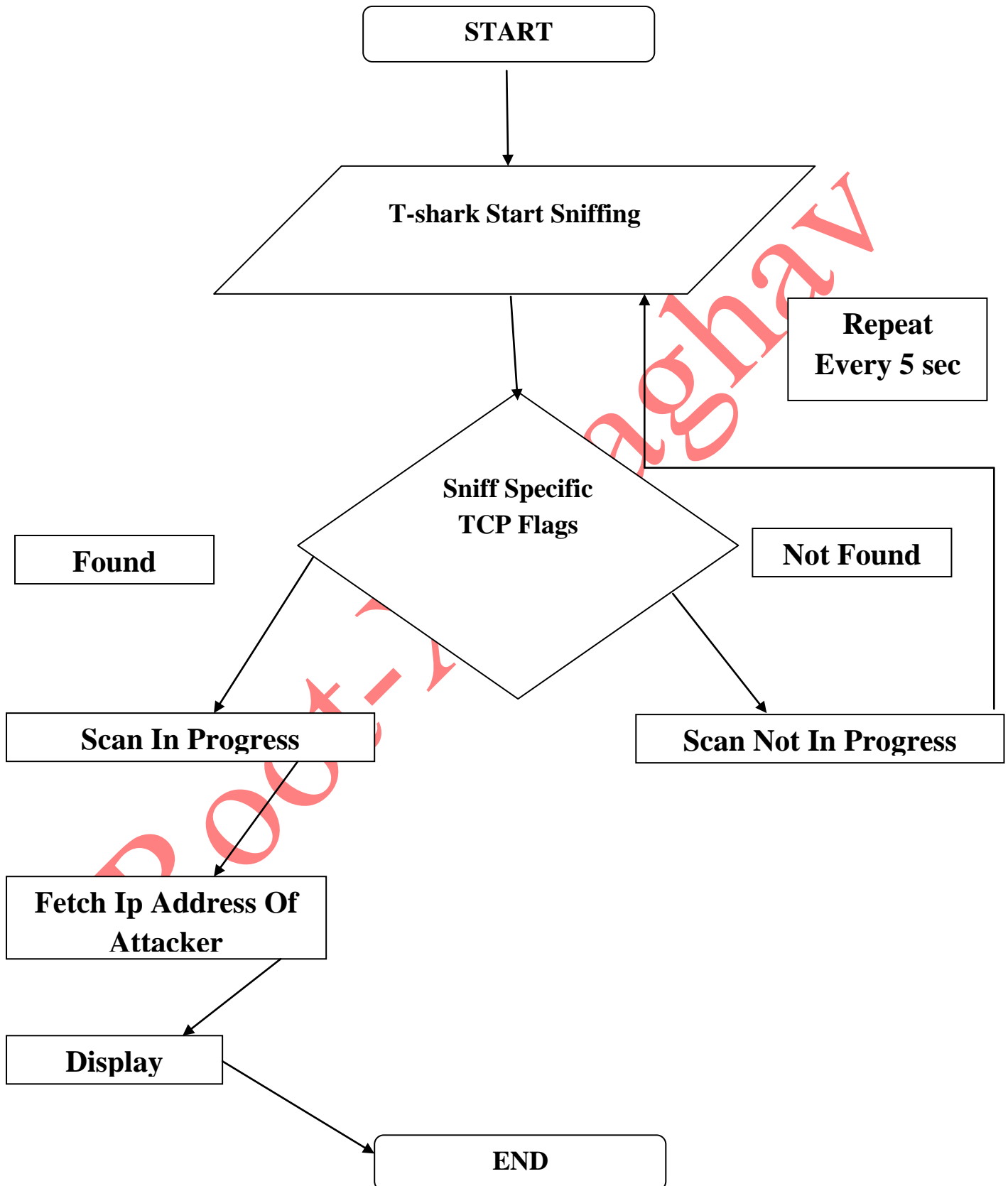
Fake Access Point

Flow Chart:



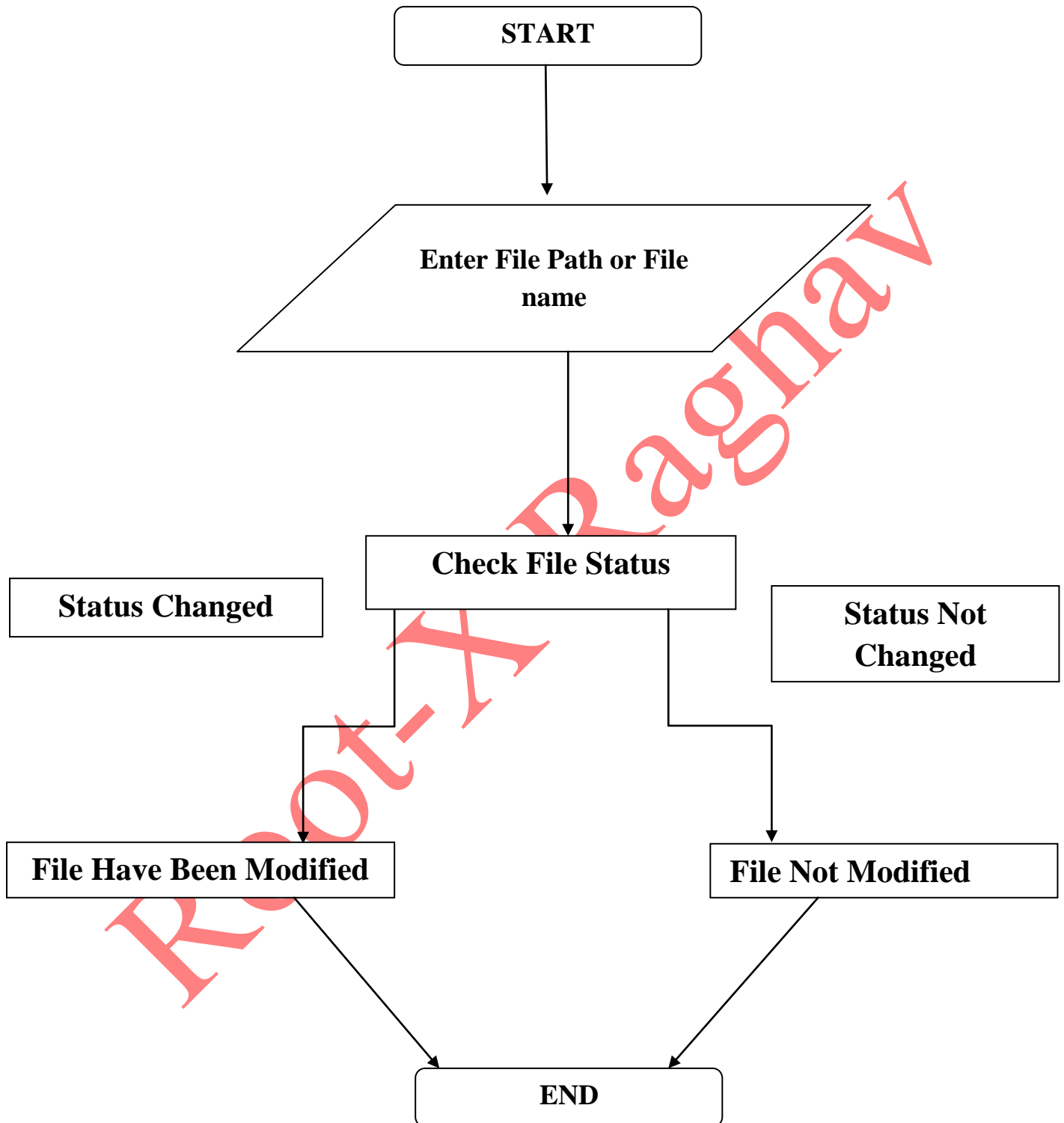
Port Scanning Attack Detector

Flow Chart:



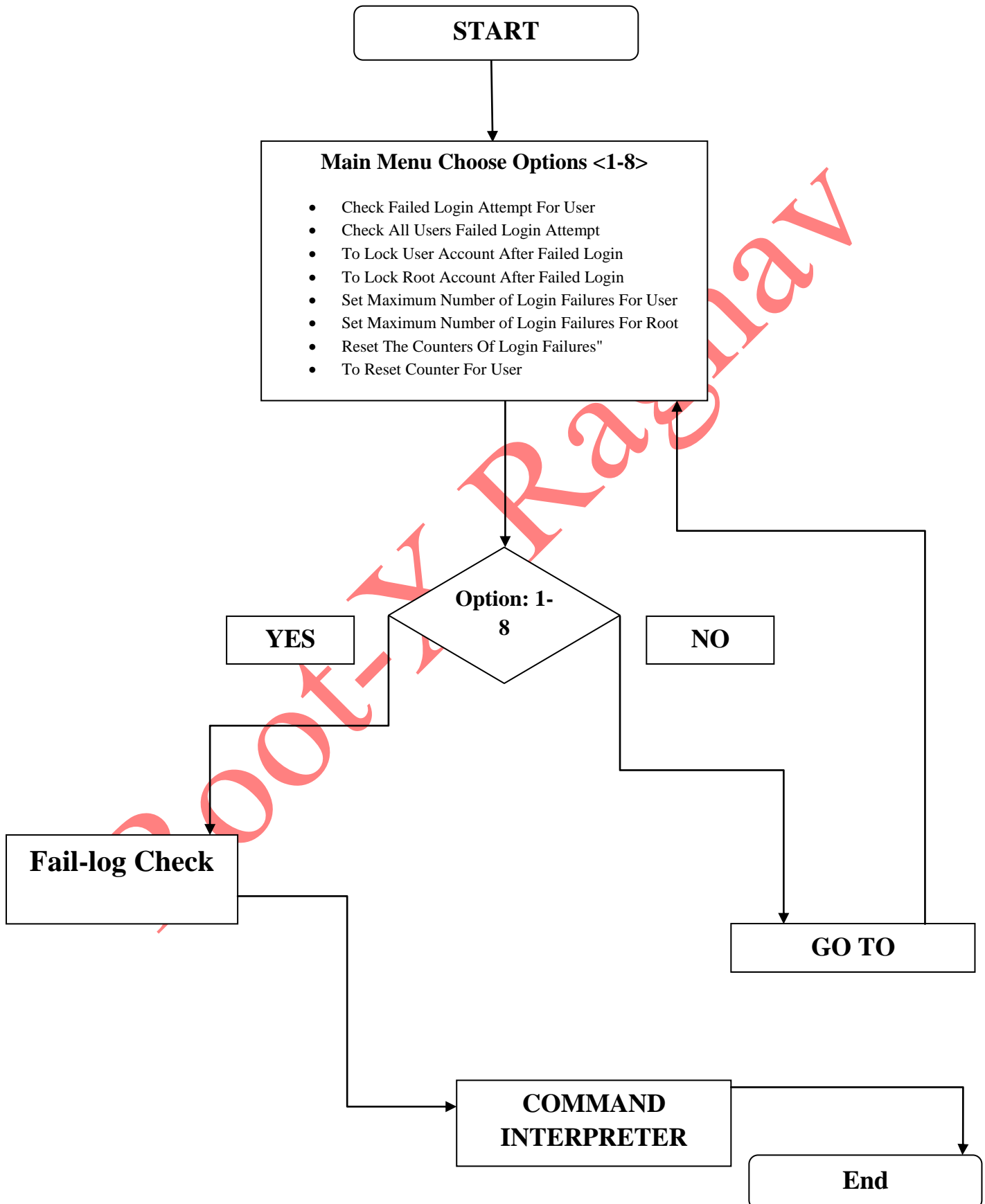
Checking Last Permission, Size, Files modification

Flow Chart:



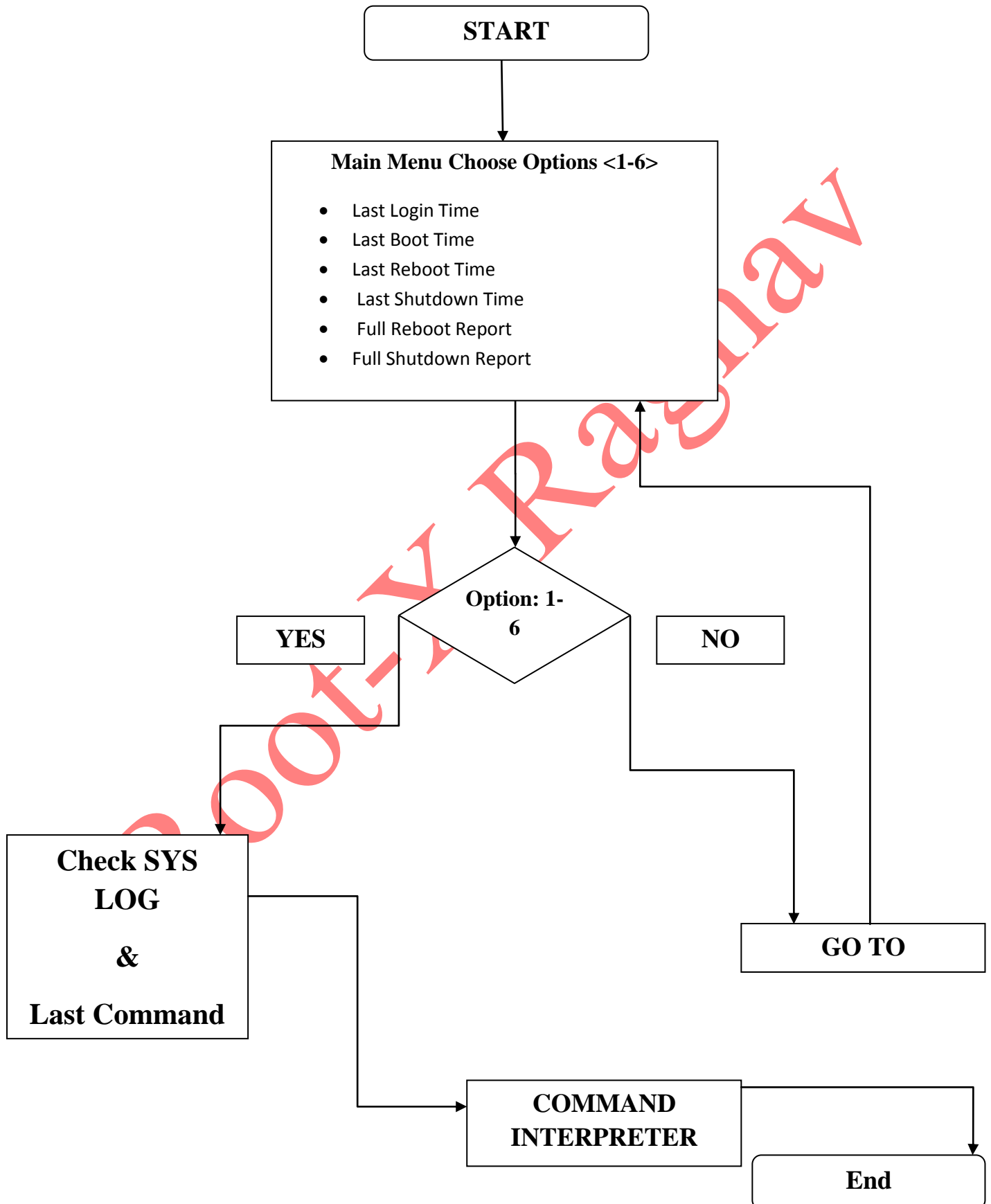
Login Attempts

Flow Chart:



System Boot logs

Flow Chart:



6.4. Pseudo codes:

➤ **Module 1 : Dos DDos**

➤ **udp.sh**

- while check,
if UDP Packets greater then 1000
print "UDP flood in Progress"
else
Print "UDP flood not in Progress"

➤ **TCP/IP.sh**

- while check,
if tcp/ip connection is greater than 1000
print "flood in progress"
else
print "flood not in progress"

➤ **SYN.sh**

- while check,
if SYN Packets greater then 1000
print "SYN flood in Progress"
else
Print "SYN flood not in Progress"

➤ **HTTP.sh**

- while check,
if HTTP Connections are greater than 1000
print "HTTP flood in Progress"
else
Print "HTTP flood not in Progress"

➤ **EstablishConnection.sh**

- while check,
if over all connection are greater than 2000
print "Establish Connection attack in Progress"

else

Print " Establish Connection attack in Progress"

➤ **PingOfDeathAttack.sh**

- start sniffer tshark on interface till count=200&
save logs,
while check,
if logs contain ICMP Packets size greater than=65500 "or" packets count > 1000
print "ping of death attack in Progress"
else
Print " ping of death attack in Progress"

➤ **CheckFlood.sh**

- Enter network interface,
Enter Port you want to check,
start TCPDUMP,
while check,
if TCPDUMP packets count > 1000
print "flooding in Progress"
else
Print " flooding not in Progress"

➤ **Check.sh**

- Init_pro=\$(ps -a | grep tcpdump --count) && while check,
Curr_pro=\$(ps -a | grep tcpdump --count)
if Init_pro equal to Curr_pro
print "Flooding in progress" else
print "Flooding not in progress"
- start xterm and run CheckFlood.sh & sane time run Check.sh

➤ **ArpPoisoningDetection.sh**

- fetch Original MAC Address : \$original_mac
fetch Default Gateway IP Address : \$ip
while check,

```

fetch Current MAC Address : $current_mac
fetch Default Gateway IP Address : $ip
if $original_mac == $current_mac\
then,
print" ARP Poisoning NOT In Progress"
else
print" ARP Poisoning In Progress"
print"Suspect MAC: $current_mac"
print"Suspect IP: $current_ip"

```

➤ **Module 2 : Firewall**

➤ **Port_Filtering.sh**

- choose option : 1-4

```

if option 1:
print "You Have Chosen Manually Blocking Of Incoming Port : "
iptables -A INPUT -p tcp --destination-port $Po -j DROP
else
print "Wrong option"
if option 2:
print "You Have chosen Manually Blocking Outgoing Port: "
iptables -A OUTPUT -p tcp --dport $out -j DROP
else
print "Wrong option"
if option 3:
print "You Have chosen Manually Allowing / Open TCP Incoming Port: "
iptables -I INPUT -p tcp --dport $qw -j ACCEPT
else
print "Wrong option"
if option 4:
print "You Have chosen Manually Allowing / Open UDP Port: "
iptables -I INPUT -p udp --dport $we -j ACCEPT
else
print "Wrong option"
end

```

➤ **Ip_Address_Blocking.sh**

- choose option : 1-2

if option 1:

print "You Have Chosen Manually Blocking Of IP: "

use ufw deny from \$ip command

else

print "Wrong option"

if option 2:

print "You Have chosen Manually Allowing Of IP :"

sudo ufw allow from \$al

else

end

➤ **Mac_Address_block.sh**

- choose option : 1-4

if option 1:

print "You Manually Blocking MAC Address: "

iptables -A INPUT -m mac --mac-source \$mac -j DROP

else print "Wrong option"

if option 2:

print "Manually Allowing MAC Address: "

iptables -I INPUT -m mac --mac-source \$al -j ACCEPT

else print "Wrong option"

if option 3:

print "You Manually Allowing SSH Access to Specific MAC Address: "

iptables -I INPUT -p tcp --dport 22 -m mac --mac-source \$qw -j ACCEPT

else print "Wrong option"

if option 4:

print "Manually Denying SSH Access to Specific MAC Address: "

iptables -I INPUT -p tcp --dport 22 -m mac --mac-source \$we -j REJECT

else print "Wrong option"

end

➤ **Spam_ip_block.sh**

- download file wget <http://www.spamhaus.org/drop/drop.lasso>
fetch IP address `cat $FILE | egrep -v '^;' | awk '{ print $1 }'`
for ip,
block ,
`iptables -A droplist -s $ipblock -j LOG --log-prefix "DROP List Block"`
`iptables -A droplist -s $ipblock -j DROP`

➤ **website_block.sh**

- choose option : 1-2
if option 1:
print "Manually Blocking Website: "
open file /etc/hosts & print "echo "127.0.0.1 '\$site'"
else print "Wrong option"
if option 2:
print "Manually Unblocking Website:"
open file /etc/host & remove 127.0.0.1 '\$site'
else
end

➤ **Evil_shell_finder.sh**

- enter server ip address
create a function=@path for all possible paths & shell
then,
check HTTP get Request
for,
given php shells path
if
HTTP::Response=success
print"full path of shell"
else
print"shell not found"
end

➤ **Trojan_Scanner.sh**

- Enter victim server name

create function =@ports for all special ports

create function =@Trojans which contain all Trojans name

for 0 to 171

check socket :

```
$socket = IO::Socket::INET->new(PeerPort => "$po", PeerAddr => $victim, Proto =>
"tcp", Timeout => $timeout)
```

```
if n=0
```

```
print "Trojan Found"
```

```
else
```

```
print "Not found "
```

```
end
```

➤ **Module 3 : Honeypot**

➤ **PortScanAttackDetector.sh**

- Enter Your Network Interface

Start capturing TCP Flags and capture logs : tshark -i \$fa -f "ip proto 6 or ip proto 17" -R

```
"tcp.flags == 16 or tcp.flags == 1 or tcp.flags == 2 or tcp.flags == 18 or tcp.flags == 41 or
tcp.flags == 16 or tcp.flags == 0 or ip.len == 28 or icmp.type == 8" >>
```

```
/root/Desktop/ids/Honeypot/HoneypotLog/Port_Attack_Scan
```

➤ **Main.sh**

- Choose Interface : cat /tmp/ethlist

Run program : xterm -hold -e sh /root/Desktop/ids/Honeypot/Port_Scan_attack_Detect.sh

```
| xterm -hold -e sh /root/Desktop/ids/Honeypot/check.sh
```

➤ **Check.sh**

- if current file size is not equal to initial file size

```
print "Scanning in progress"
```

```
else
```

```
print "Scanning not in progress"
```

```
end
```

➤ **fake_access_point.sh**

- Select a wireless interface to use for the AP : selectWirelessInterface ()
use iwconfig
print result cat /tmp/ethlist

Start Networking

networking()

if USER input Y or y : then,

service networking start

elif USER input N or n : then,

call function selectWirelessInterface()

print "Check for external connection"

if USER input Y or y : then,

ifconfig \$wlan up

Start monitor : monitor()

print "Starting monitor device putting your device in monitor mode... "

airmon-ng start \$wlan

Start fake access point : createAP ()

print "Use default ESSID [Free Public WiFi][Y/N]?"

if USER enter N then

print "What will the AP name be?"

print "Start Access Point [Y/N]?"

print "AP name will be \$nameap"

if User Enter Y : airbase-ng -P -e \$nameap \$monintme

else : airbase-ng -e \$nameap \$monintme

active external interface : ExternalAccess ()

print "Please wait network routing is on run....."

iptables -t nat -A POSTROUTING -o \$wlan -s 192.168.121.0/24 -j MASQUERADE

echo 1 > /proc/sys/net/ipv4/ip_forward

start: startRodents ()


```
/pentest/sniffers/hamster/hamster &  
/pentest/sniffers/hamster/ferret -i at0 &
```

```
capture image : captureImages ()  
driftnet -i at0 -a -d /tmp/driftimages
```

```
start brupsuite : startBurp ()  
/root/burpme.sh
```

Call all functions one by one :

```
network  
monitor  
createAP  
ExternalAccess  
dhcp  
routing  
captureImages  
startRodents  
startBurp
```

➤ **Kill_fake_access_point.sh**

- stop networking : stopNetworking ()
service networking stop

```
stop fake access point : killAP ()  
print "Killing Fake Access Point"  
basepid=`ps -ef | grep -i [a]irbase | awk '{ print $2 }`  
kill -9 $basepid
```

```
Kill all external access : killExternalAccess ()  
print "Stopping DHCP server..."  
ifconfig at0 down
```

```
Kill Routing : noRouteAPClients ()  
print "Killing routing..."
```

```
iptables -t nat -D POSTROUTING -o wlan0 -s 192.168.121.0/24 -j MASQUERADE  
echo 0 > /proc/sys/net/ipv4/ip_forward
```

Call all functions :

noRouteAPClients

stopDHCP

killExternalAccess

killAP

stopMonitor

stopNetworking

➤ **Module 4 : Logs Management**

➤ **Last_Modification_of_file.sh**

- Enter file full path
if 1 then check /dev/null
else
print "command not found"
end if
check time of last access : stat -c %x \$FILE
time of last modification : stat -c %y \$FILE
time of last change : stat -c %z \$FILE
end

➤ **Modified_size.sh**

- Enter file full path
if 1 then check /dev/null
else
print "command not found"
end if
Inode Numbe Of File : stat -c %i \$FILE
Input/Output Block Size : stat -c %o \$FILE
Total Size in Bytes : stat -c %s \$FILE
Fundamental block size (for block counts) : stat -c %S \$FILE
Number of blocks allocated : stat -c %c \$FILE
end

➤ **Modified_permission.sh**

- Enter file full path
if 1 then check /dev/null
else
print "command not found"
end if
Permission of file : ls -l \$FILE
SELinux security context string : stat -c %C \$FILE
Number of hard links : stat -c %h \$FILE
end

➤ **System_Boot_Logs.sh**

- choose option 1-7
if option is 1 :
print "Last Login is : " last login
elif option is 2 :
print "Last Boot Time is : " who -b
elif option is 3 :
print "Last Reboot Time is : " last reboot | head -1
elif option is 4 :
print "Last Shutdown Time is : " last -x | grep shutdown | head -1
elif option is 5 :
print "Full Reboot Report is : " last reboot
elif option is 6 :
print "Full Shutdown Report is : " last -x | grep shutdown
elif option is 7 :
print "wrong code"
exit
end if

➤ **System_Boot_Logs.sh**

- choose option 1-8
if option is 1 :
print "Failed Login Attempt For User : " faillog -u \$root
elif option is 2 :

```
print "All Users Failed Login Attempt :" faillog -a
elif option is 3 :
print "Lock User Account After Failed Login :" faillog -l $sec -u $usr
elif option is 4 :
print "Lock Root Account After Failed Login :" faillog -l $ro
elif option is 5 :
print "Maximum Number of Login Failures For User Accounts :" faillog -M $d -u $u
elif option is 6 :
print "Maximum Number of Login Failures For Root Accounts :" faillog -M $f
elif option is 7 :
print "Reset The Counters Of Login Failures :" faillog -r
elif option 8 :
print "To Reset Counter For User :" faillog -r -u $v
elif option is 9 :
print "wrong code"
exit
end if
```

7. TESTING

7.1 Functional Testing

In Functional testing we need check the each components are functioning as expected or not, so it is also called as Component Testing

Functional testing is to testing the functionality of the software application under test. Basically, it is to check the basic functionality mentioned in the functional specification document. Also check whether software application is meeting the user expectations. We can also say that checking the behavior of the software application against test specification.

This type of testing is mandatory and irrespective of what type of application this should be exercised.

What all need to be check in Functional Testing:

2. Is software is functioning as it should do?
3. Is software is not functioning as it should not do?
4. Is software is not doing as it not intended to do?

7.2. Structural Testing

The structural testing is the testing of the structure of the system or component. Structural testing is often referred to as 'white box' or 'glass box' or 'clear-box testing' because in structural testing we are interested in what is happening 'inside the system/application'. In structural testing the testers are required to have the knowledge of the internal implementations of the code. Here the testers require knowledge of how the software is implemented, how it works.

During structural testing the tester is concentrating on how the software does it. For example, a structural technique wants to know how loops in the software are working. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software. Structural testing can be used at all levels of testing. Developers use structural testing in component testing and component integration testing, especially where there is good tool support for code coverage. Structural testing is also used in system and acceptance testing, but the structures are different. For example, the

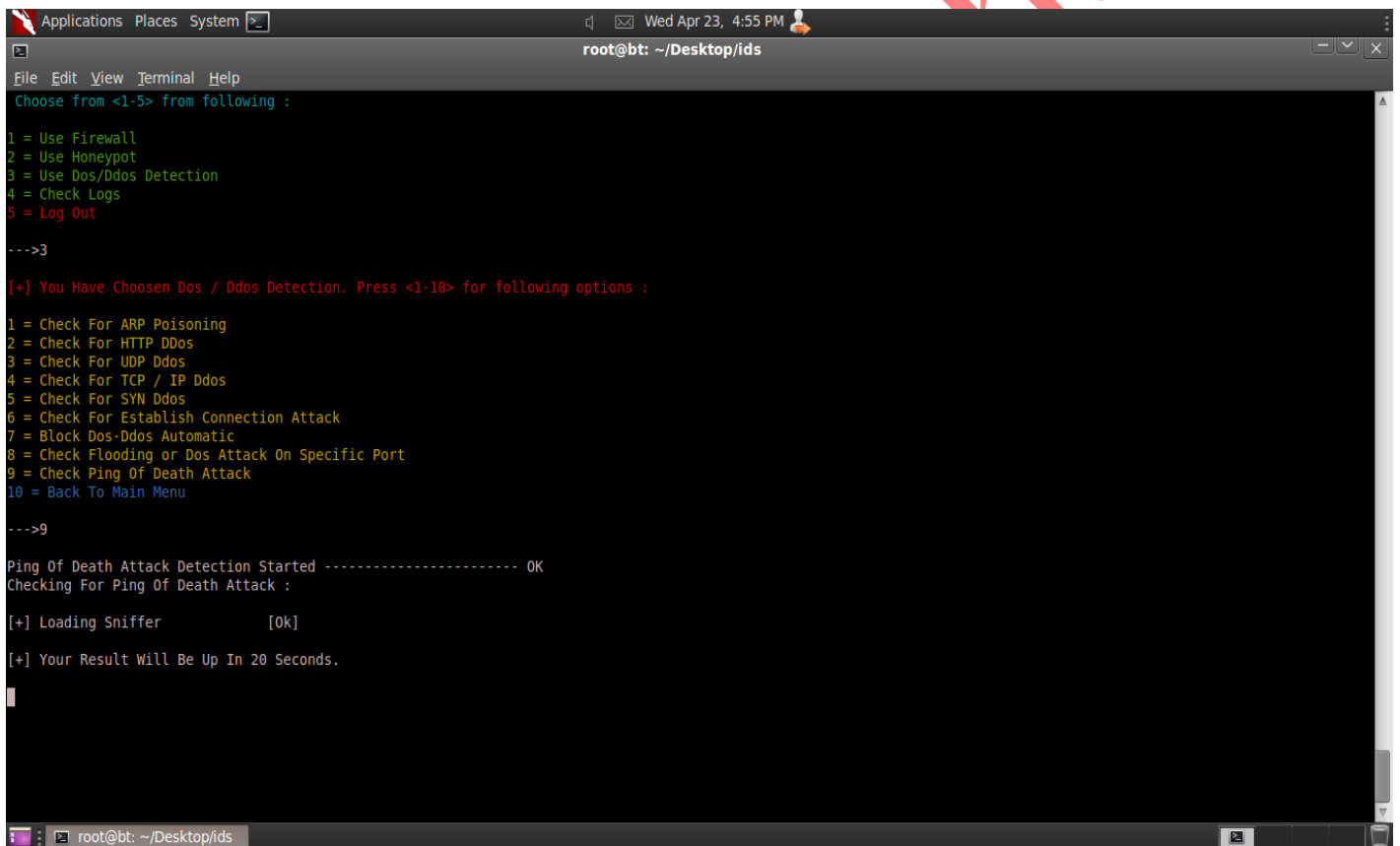
coverage of menu options or major business transactions could be the structural element in system or acceptance testing.

Here is some of the tests we performed

Test1: Ping of death attack

Expected Result: Ping of Death Attack Should Start

Actual Result: Ping of Death Attack Started



```
Applications Places System | root@bt: ~/Desktop/ids
File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use HoneyPot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->3

[+] You Have Chosen Dos / Ddos Detection. Press <1-10> for following options :

1 = Check For ARP Poisoning
2 = Check For HTTP DDos
3 = Check For UDP Ddos
4 = Check For TCP / IP Ddos
5 = Check For SYN Ddos
6 = Check For Establish Connection Attack
7 = Block Dos-Ddos Automatic
8 = Check Flooding or Dos Attack On Specific Port
9 = Check Ping Of Death Attack
10 = Back To Main Menu

--->9

Ping Of Death Attack Detection Started ----- OK
Checking For Ping Of Death Attack :

[+] Loading Sniffer          [0k]

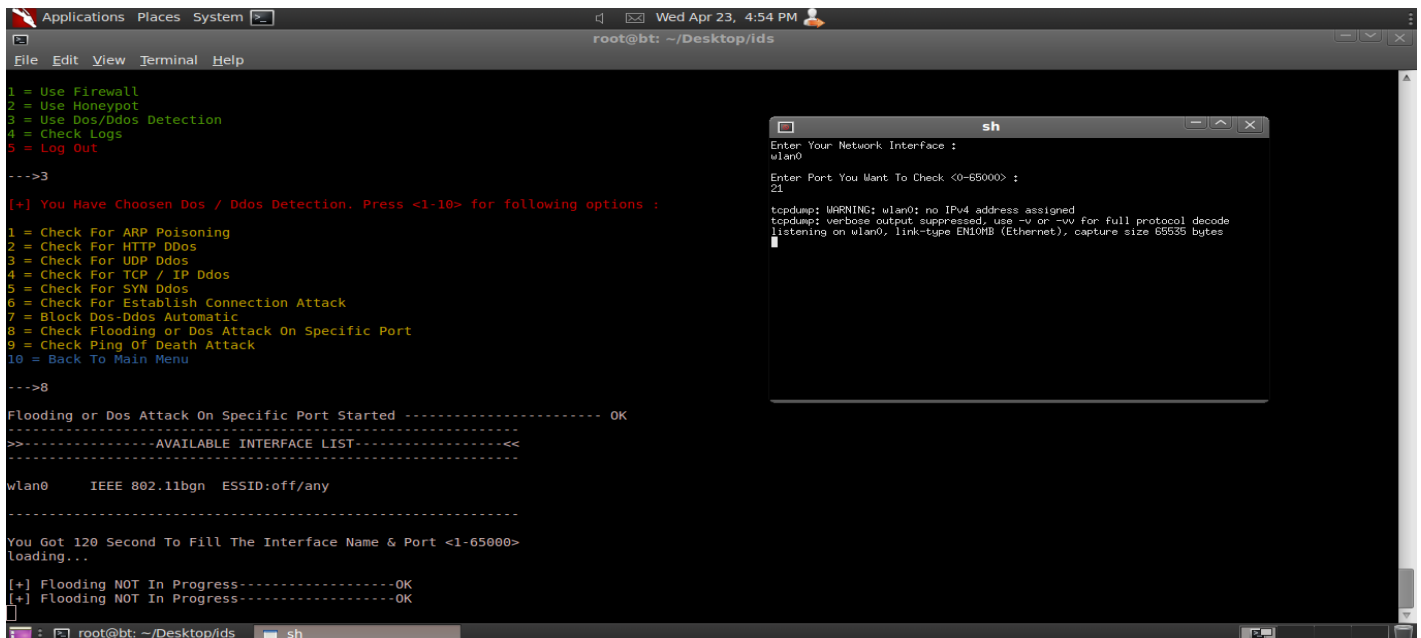
[+] Your Result Will Be Up In 20 Seconds.
```

Fig 7.1

Test 2: Check Flooding or Dos Attack

Expected Result: Flooding or Dos Attack Scanning Should Start

Actual Result: Flooding or Dos Attack Scanning Started



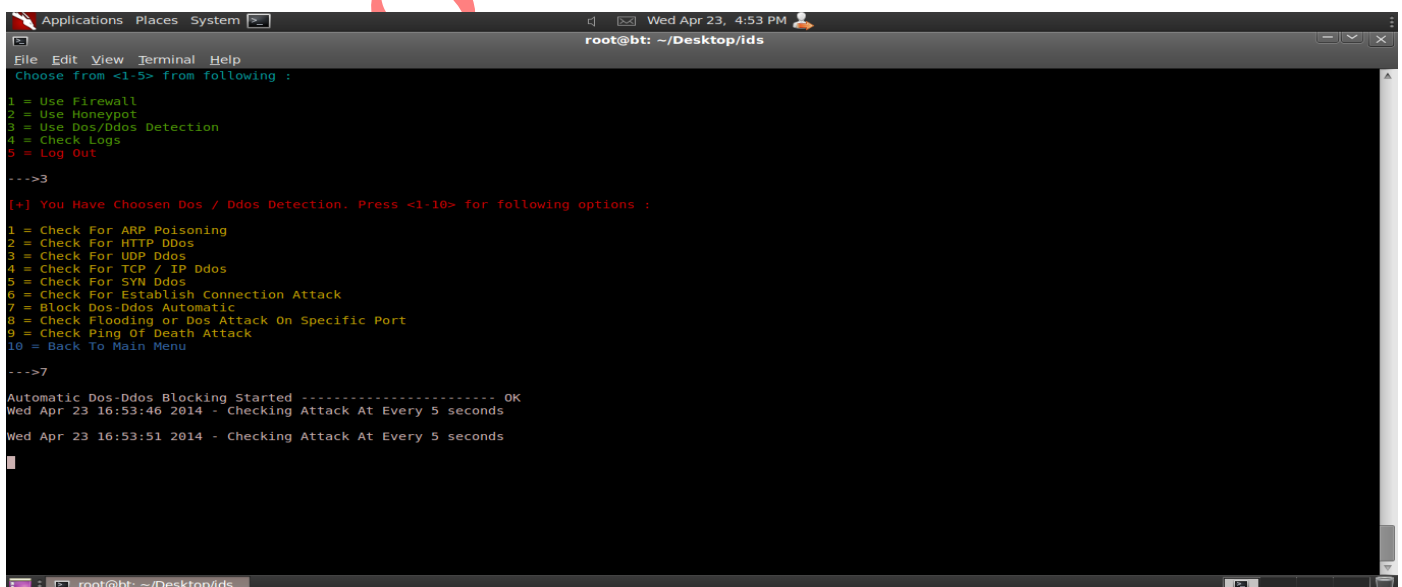
```
root@bt: ~/Desktop/ids
1 = Use Firewall
2 = Use HoneyPot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
--->3
[+] You Have Chosen Dos / Ddos Detection. Press <1-10> for following options :
1 = Check For ARP Poisoning
2 = Check For HTTP Ddos
3 = Check For UDP Ddos
4 = Check For TCP / IP Ddos
5 = Check For SYN Ddos
6 = Check For Establish Connection Attack
7 = Block Dos-Ddos Automatic
8 = Check Flooding or Dos Attack On Specific Port
9 = Check Ping Of Death Attack
10 = Back To Main Menu
--->8
Flooding or Dos Attack On Specific Port Started ----- OK
>>-----AVAILABLE INTERFACE LIST-----<<
wlan0      IEEE 802.11bgn  ESSID:off/any
-----
You Got 120 Second To Fill The Interface Name & Port <1-65000>
loading...
[+] Flooding NOT In Progress-----OK
[+] Flooding NOT In Progress-----OK
```

Fig 7.2

Test 3: Check DDos or Dos Attack

Expected Result: DDos or Dos Attack Checking Should Start

Actual Result: DDos or Dos Attack Scanning Started



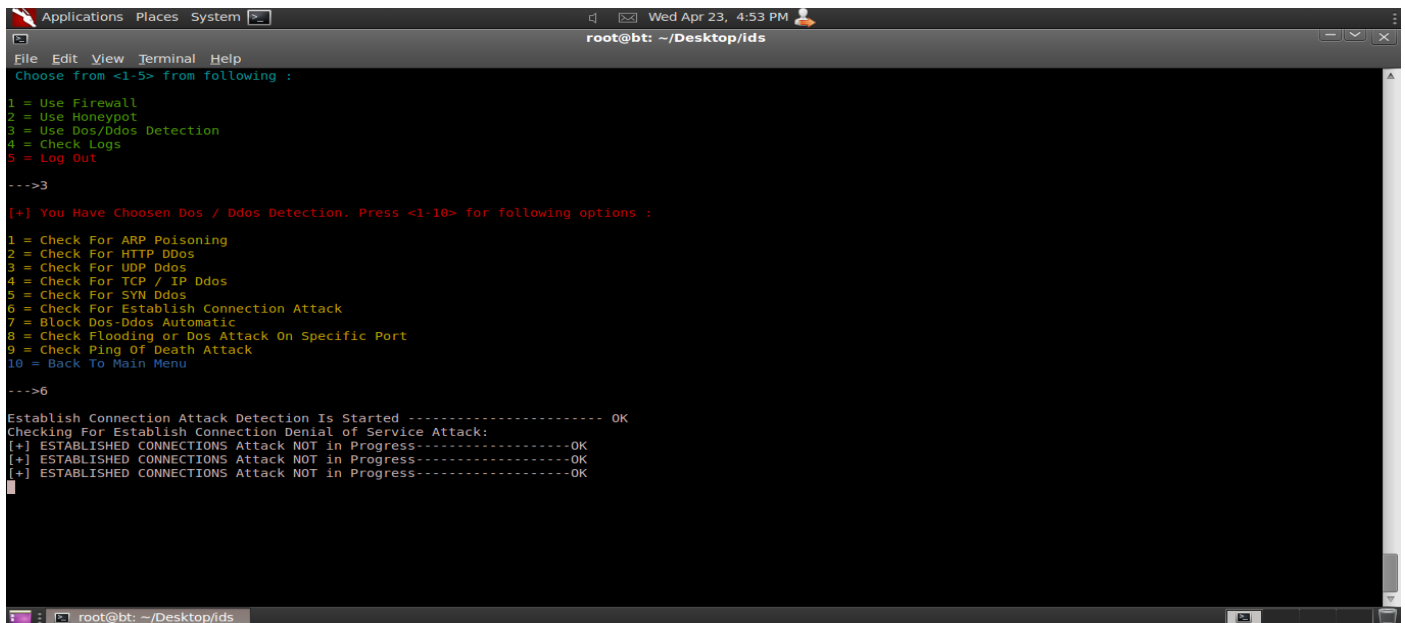
```
root@bt: ~/Desktop/ids
Choose from <1-5> from following :
1 = Use Firewall
2 = Use HoneyPot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
--->3
[+] You Have Chosen Dos / Ddos Detection. Press <1-10> for following options :
1 = Check For ARP Poisoning
2 = Check For HTTP Ddos
3 = Check For UDP Ddos
4 = Check For TCP / IP Ddos
5 = Check For SYN Ddos
6 = Check For Establish Connection Attack
7 = Block Dos-Ddos Automatic
8 = Check Flooding or Dos Attack On Specific Port
9 = Check Ping Of Death Attack
10 = Back To Main Menu
--->7
Automatic Dos-Ddos Blocking Started ----- OK
Wed Apr 23 16:53:46 2014 - Checking Attack At Every 5 seconds
Wed Apr 23 16:53:51 2014 - Checking Attack At Every 5 seconds
```

Fig 7.3

Test 4: Check Establish Connection Attack

Expected Result: Establish Connection Attack Scanning Should Start

Actual Result: Establish Connection Scanning Started



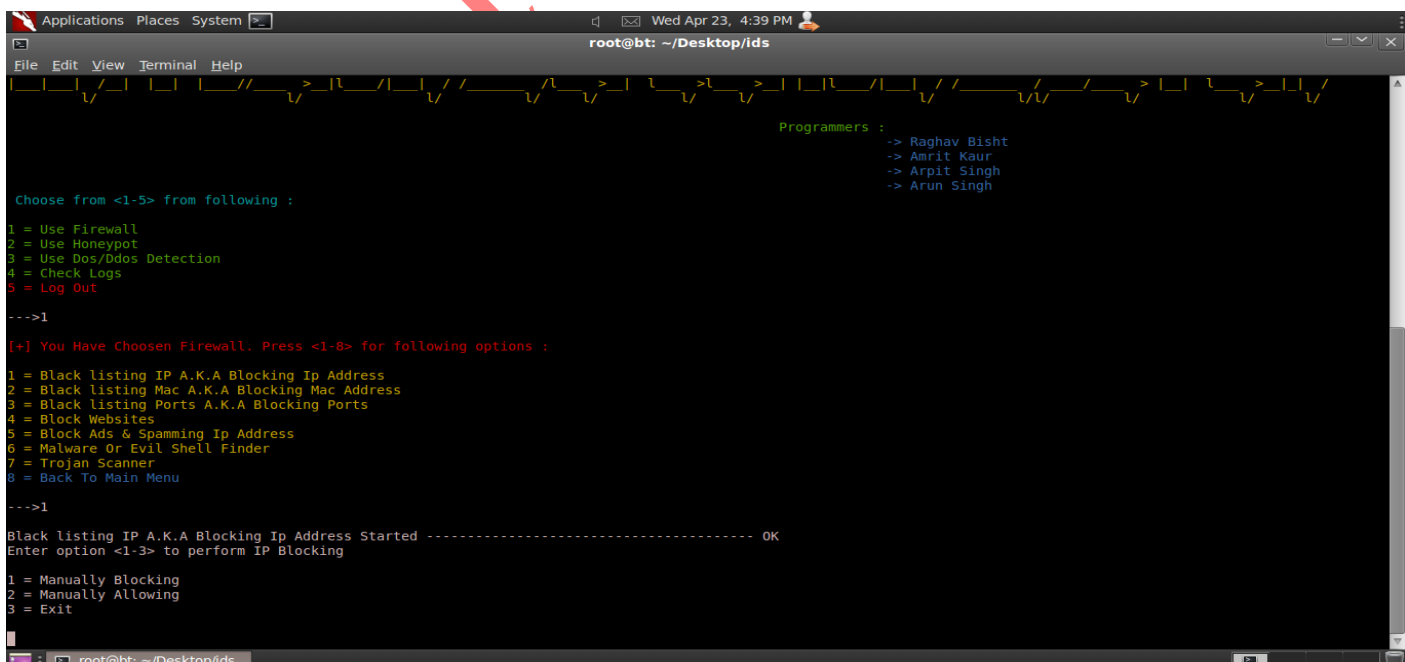
```
root@bt: ~/Desktop/ids
Choose from <1-5> from following :
1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
--->3
[+] You Have Chosen Dos / Ddos Detection. Press <1-10> for following options :
1 = Check For ARP Poisoning
2 = Check For HTTP Ddos
3 = Check For UDP Ddos
4 = Check For TCP / IP Ddos
5 = Check For SYN Ddos
6 = Check For Establish Connection Attack
7 = Block Dos-Ddos Automatic
8 = Check Flooding or Dos Attack On Specific Port
9 = Check Ping Of Death Attack
10 = Back To Main Menu
--->6
Establish Connection Attack Detection Is Started ----- OK
Checking For Establish Connection Denial of Service Attack:
[+] ESTABLISHED CONNECTIONS Attack NOT in Progress-----OK
[+] ESTABLISHED CONNECTIONS Attack NOT in Progress-----OK
[+] ESTABLISHED CONNECTIONS Attack NOT in Progress-----OK
```

Fig 7.4

Test 5:Block I.P Address

Expected Result: Interface for Blocking I.P Address Should Open

Actual Result: Interface for Blocking I.P Address is Opened



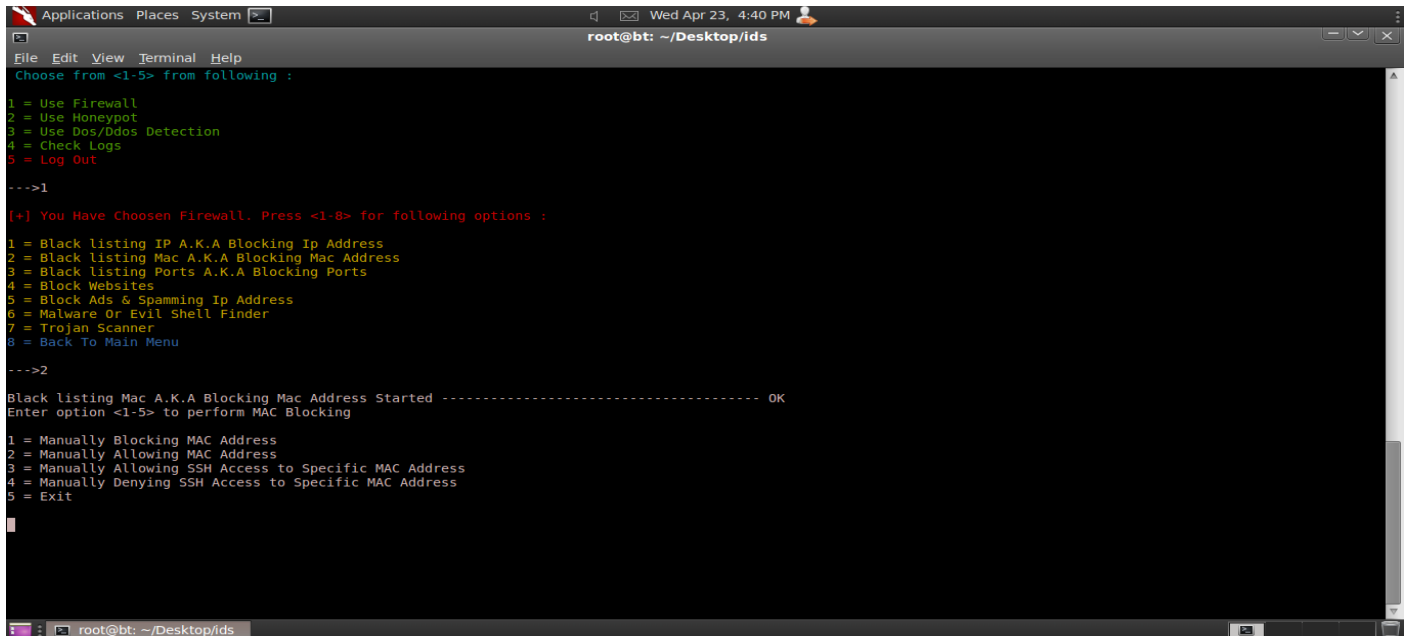
```
root@bt: ~/Desktop/ids
Choose from <1-5> from following :
1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
--->1
[+] You Have Chosen Firewall. Press <1-8> for following options :
1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu
--->1
Black listing IP A.K.A Blocking Ip Address Started ----- OK
Enter option <1-3> to perform IP Blocking
1 = Manually Blocking
2 = Manually Allowing
3 = Exit
```

Fig 7.5

Test 6: Block Mac Address

Expected Result: Interface for Blocking Mac Address Should Open

Actual Result: Interface for Blocking Mac Address is Opened



```
Applications Places System
root@bt: ~/Desktop/ids
File Edit View Terminal Help
Choose from <1-5> from following :
1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
--->1
[+] You Have Chosen Firewall. Press <1-8> for following options :
1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu
--->2
Black listing Mac A.K.A Blocking Mac Address Started ----- OK
Enter option <1-5> to perform MAC Blocking
1 = Manually Blocking MAC Address
2 = Manually Allowing MAC Address
3 = Manually Allowing SSH Access to Specific MAC Address
4 = Manually Denying SSH Access to Specific MAC Address
5 = Exit
```

Fig 7.6

Test 7:Blacklisting Ports

Expected Result: Interface for Blocking/AllowingPorts Should Open

Actual Result: Interface for Blocking/Allowing Ports is Opened



```
Applications Places System
root@bt: ~/Desktop/ids
File Edit View Terminal Help
Choose from <1-5> from following :
1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
--->1
[+] You Have Chosen Firewall. Press <1-8> for following options :
1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu
--->3
Port Blocking Started ----- OK
Enter option <1-5> to perform MAC Blocking
1 = Manually Blocking / Close Incoming Port
2 = Manually Blocking / Close Outgoing Port
3 = Manually Allowing / Open TCP Incoming Port
4 = Manually Allowing / Open UDP Port
5 = Exit
```

Fig 7.7

7.3 Penetration Testing

1. Test for Resource Exhaustion

- IDS are prone to resource exhaustion attacks
- Every IDS system has memory, CPU, or bandwidth limitations
- The IDS performance might degrade or fail if these resources are exhausted
- Test by sending large amounts of traffic to the IDS system

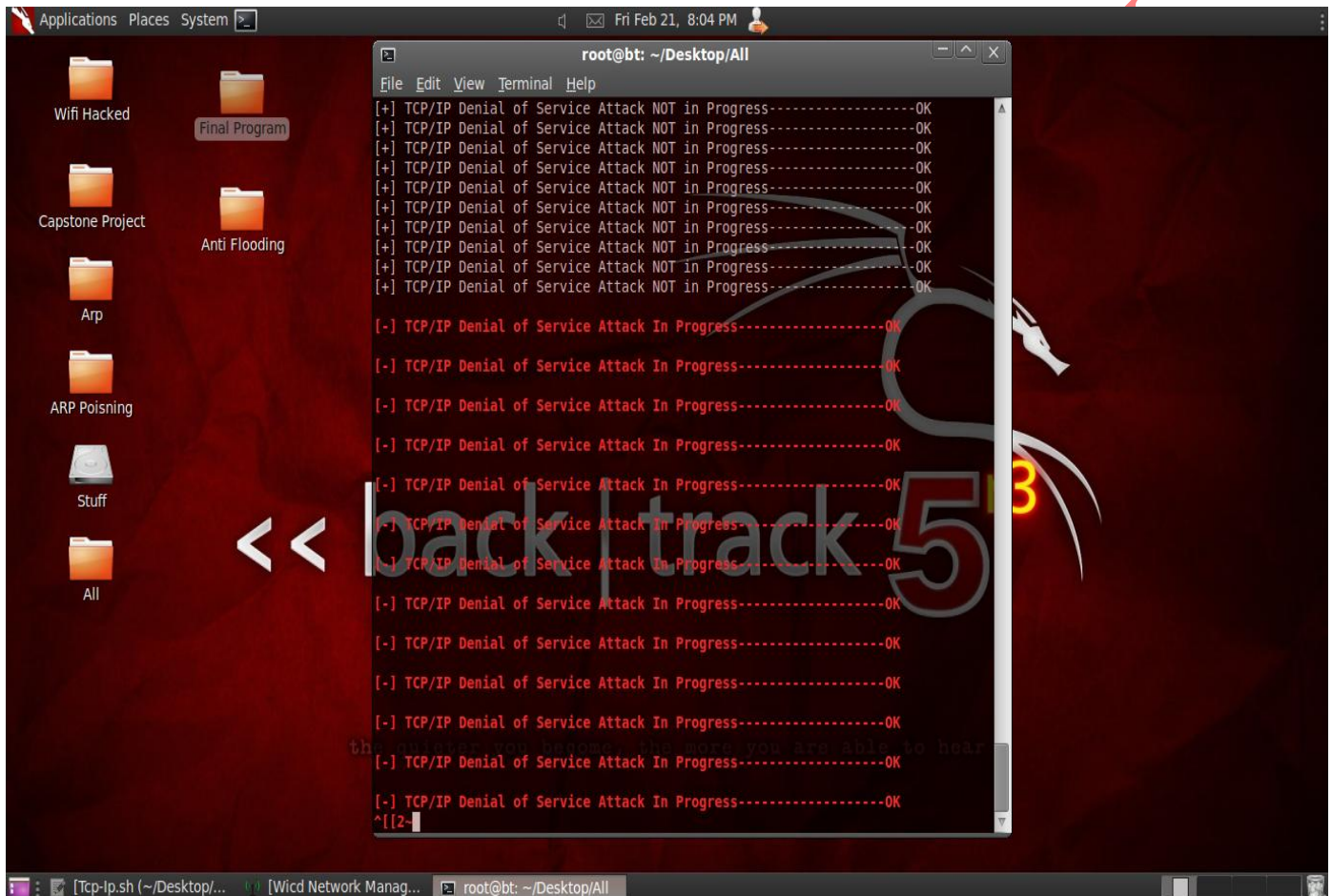


Fig 7.3.1

2. Test the IDS by Sending ARP Flood

- Flood the network by sending ARP packets.
- See the IDS response and how it reacts to this attack.

Avoid trust relationships: Organizations should develop protocols that rely on trust relationships as little as possible. It is significantly easier for attackers to run spoofing attacks when trust relationships are in place because trust relationships only use IP addresses for authentication.

Use spoofing detection software: There are many programs available that help organizations detect spoofing attacks, particularly ARP spoofing. These programs work by inspecting and certifying data before it is transmitted and blocking data that appears to be spoofed.

Use cryptographic network protocols: Transport Layer Security (TLS), Secure Shell (SSH), HTTP Secure (HTTPS), and other secure communications protocols bolster spoofing attack prevention efforts by encrypting data before it is sent and authenticating data as it is received.

5. Ping of Death Test

IP specifications prohibit the creation of packets greater than 65535 bytes in length. However, packet fragmentation permits an attacker to transmit packets exceeding this length. The "Ping of Death" attack involves transmitting a fragmented ICMP echo packet greater than 65535 bytes in length to a vulnerable system. When the victim system networking stack reassembly code reassembles the packet, the allocated buffer may not be able to accommodate the packet. This can cause the system to crash, restart, or behave in unpredictable ways.

If a Ping Of Death is issued against a system that is immune to such attacks, the reply to the ping will also be a Ping Of Death. The first of the two events points to the attacker.

This attack is not limited to ICMP and can be exploited with any protocol that uses IP.

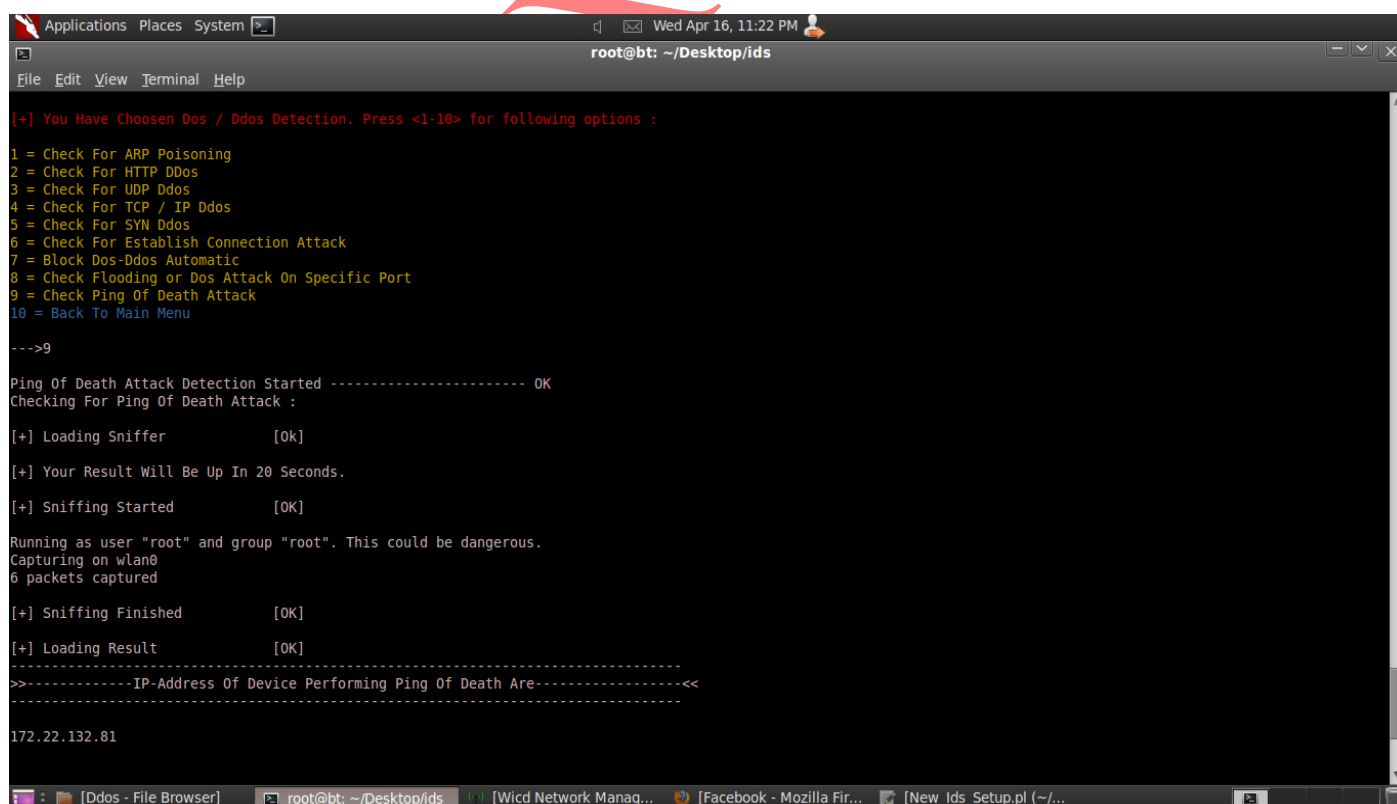
A screenshot of a Linux terminal window titled 'root@bt: ~/Desktop/ids'. The terminal shows a menu of options for detecting various attacks. Option 9, 'Check Ping Of Death Attack', is selected. The process begins with 'Ping Of Death Attack Detection Started' and 'Checking For Ping Of Death Attack :'. It then shows 'Loading Sniffer' and 'Sniffing Started'. A message indicates it is running as user 'root' and group 'root', which could be dangerous. It shows 'Capturing on wlan0' and '6 packets captured'. After 'Sniffing Finished', it proceeds to 'Loading Result'. The final output is a list of IP addresses: '172.22.132.81'. The terminal window has a standard Ubuntu-style top bar with 'Applications', 'Places', and 'System' menus, and a system clock showing 'Wed Apr 16, 11:22 PM'. The bottom of the window shows a taskbar with several open applications including '[Ddos - File Browser]', 'root@bt: ~/Desktop/ids', '[Wicd Network Manag...', '[Facebook - Mozilla Fir...', and '[New_ids_Setup.pl (~/...]'.

fig 7.3.5

6. Test for Odd Sized Packets

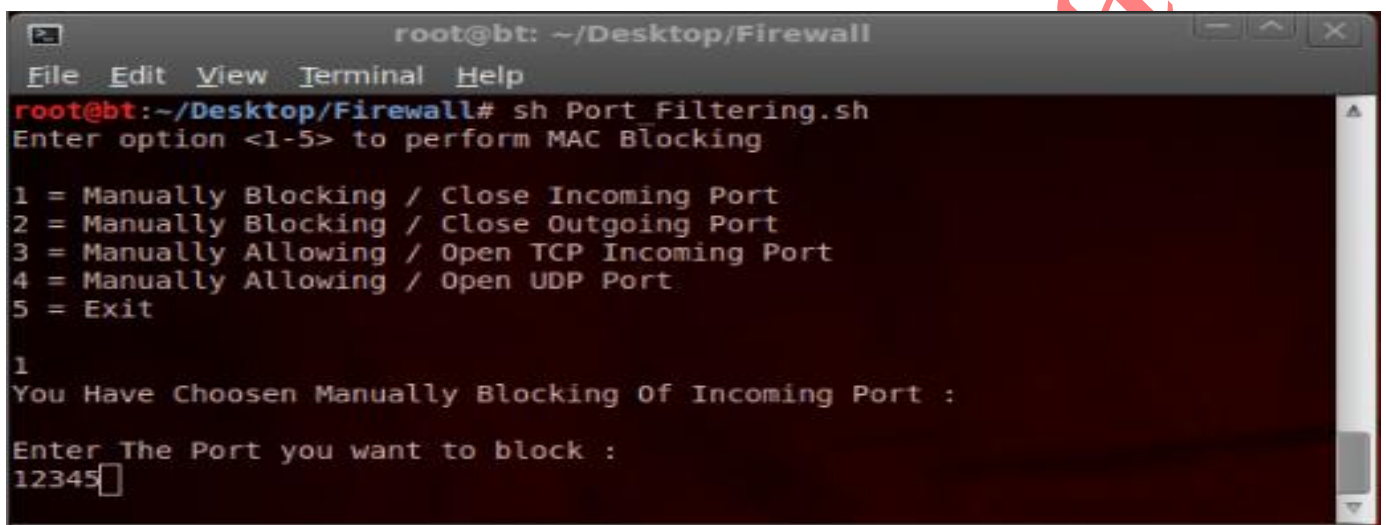
- It is highly suspicious if a fragmented packet has a length that is not an even multiple of 8, since packets are fragmented in multiples of 8.

7. Test by Sending a Packet to Port 0-65500

- Test for checking network traffic on all ports.

8. Test by Sending a Packet to Port 0

- For both TCP and UDP, port 0 traffic is considered unusual, since it is officially a reserved port and shouldn't be used for any network communications. Any port 0 traffic is probably not legitimate, since the packets are probably generated synthetically.



```
root@bt: ~/Desktop/Firewall
File Edit View Terminal Help
root@bt:~/Desktop/Firewall# sh Port Filtering.sh
Enter option <1-5> to perform MAC Blocking

1 = Manually Blocking / Close Incoming Port
2 = Manually Blocking / Close Outgoing Port
3 = Manually Allowing / Open TCP Incoming Port
4 = Manually Allowing / Open UDP Port
5 = Exit

1
You Have Chosen Manually Blocking Of Incoming Port :

Enter The Port you want to block :
12345
```

fig 7.3.8

9. Test TCP Flags.

☉ (none)

- A packet with no flags is neither Session Initiation (SYN), Midstream (ACK), nor Termination (FIN/RST). It is not part of any valid TCP transaction.

☉ SYN/FIN

- The flag combination indicates both Session Initiation (SYN) and Session Termination (FIN).

☉ SYN/RST

- The flag combination indicates both Session Initiation (SYN) and Session Termination (RST).

☉ SYN/FIN/ACK

- This flag combination indicates Session Initiation (SYN), Midstream (ACK), and Session Termination (FIN).

☉ SYN/RST/ACK

- This flag combination indicates Session Initiation (SYN), Midstream (ACK), and Session Termination (RST).

☉ All Flags

- Often called the Xmas Tree flag combination, this combines the problem of Initiation, Midstream, and Termination flags with the PSH and URG flags.

10. Test the IDS by Sending SYN Floods

- Many TCP implementations are vulnerable to a resource-exhaustion attack known as SYN flooding, in which excessive requests are made to create sessions, thus causing memory utilization to occur
- If these SYN packets are spoofed from addresses that do not exist, no response packet containing SYN/ACK will be received, and the pending connection queue will expand

11. Test the IDS by Sending HTTP Floods

- Many TCP implementations are vulnerable to a resource-exhaustion attack known as HTTP flooding, in which excessive requests are made to get server responses.

12. Test the IDS by Sending UDP Floods

- Many TCP implementations are vulnerable to a resource-exhaustion attack known as UDP flooding, in which excessive flow of packets towards victim.

13. Test the IDS by Checking TCP/Ip Connections

- Many TCP implementations are vulnerable to a resource-exhaustion attack known as TCP/IP flooding. In this huge amount of data is send to server on different protocols.

14. Test the IDS by Checking Establish Connection

- In establish connection attack we check all IP address connected to our serve is making server resource 100% consume. Eg. Ra Flooding attack for IPv6 IPs.

15. Test for Backscatter

- ☉ The term backscatter refers to the response SYN/ACK packets that a SYN-flooded host will send in response to receiving the SYN packets

- ⊙ If the source address of the original SYN packet is spoofed, the SYN/ACKs will be sent to that spoofed address, which may use all the network bandwidth for the spoofed host or network
- ⊙ Backscatter can easily be detected as a flood of SYN/ACK packets without an initial SYN being sent

16. Test the IDS With ICMP Packets

- ICMP packet spoofing can be used to create denial-of-service situations by falsely propagating error indications throughout the network

17. Test Using TCP Replay

- How does TCPReplay help test NIDS systems?
- Performance degrades as network traffic increases
- Attacks are hidden by heavily loaded traffic

18. Locate the firewall

- ⊙ Craft an SYN packet using Hping or any other packet crafter send it to the firewall
- ⊙ If you get ICMP unreachable type 13 message (which is admin prohibited packet) with a source IP address of access control device, usually this is a packet filter firewall
- ⊙ Tool
- `hping2 www.xsecurity.com -c2 -S -p23 -n`
- ICMP Unreachable type 13 from 10.10.2.3

19. Port Scan the Firewall

- ⊙ Most firewall implementations have default ports in use for remote management purposes
- ⊙ Example: user authentication, management, VPN connections etc
- ⊙ Tool:
- `#nmap -n -vv -P0 -p256, 1080 <www.xsecurity.com>`

20. Create Custom Packets and Look for Firewall Responses

- ⊙ Creating custom packets that are sent towards the firewall can elicit unique responses from the firewall
- ⊙ This can also be used to determine the type of firewall
- ⊙ Example:

`hping 10.0.0.5 -c 2 -S -p 23 -n`

HPING 10.0.0.5 (eth0 10.0.0.5): S set, 40 data bytes

60 bytes from 10.0.0.5 : flags=RA seq=0 ttl=59 id=0 win=0 time=0.4 ms

21. Test Access Control Enumeration

☉ Use Nmap to enumerate the firewall access control list

☉ Nmap shows three states of ports

1. Open – port is listening
2. Filtered – port is blocked by an access control device (Router/Firewall)
3. Unfiltered – traffic is passing from access control devices (Firewall/Router) but the port is not open

☉ Example:

```
#nmap -sA 192.168.0.1
```

Interesting ports on 192.168.0.1:

(The 65530 ports scanned but not shown below are in state: filtered)

PORT STATE SERVICE

110/tcp Unfiltered pop-3

13701/tcp Unfiltered VeritasNetbackup

13711/tcp Unfiltered VeritasNetbackup

13721/tcp Unfiltered VeritasNetbackup

13782/tcp Unfiltered VeritasNetbackup

Nmap run completed -- 1 IP address (1 host up) scanned in 12205.371 seconds

22. Test to Identify Firewall Architecture

☉ Hping2 is a tool for custom packet crafting

☉ Use hping2 to identify:

- Open
- Blocked
- Dropped
- Rejected packets

23. Test Firewall Using Fire-walking Tool

- ⊙ Firewalk can be used to discover open ports behind a firewall and it can be used for access control list discovery
- ⊙ Helps determine open ports on a firewall (packet filter)
- ⊙ Firewalk determines if a given port is allowed through a F/W
- ⊙ Traceroute to any machine behind the firewall or the router before the firewall
- ⊙ Once the hop count of the router is known, we can change our TTL value for our IP packet to be 1 more than the hop count of the router & perform a port scan on the firewall
- ⊙ Thus if "TTL exceeded error" comes back then port on the firewall is open

24. Test Covert Channels

- ⊙ Install a backdoor on a victim machine inside the network
- ⊙ Reverse connect to a machine outside the firewall
- ⊙ Tools: WWW Reverse Shell

25. Overview Report

- ⊙ 56 Tests With Comparison to "Other Top 10 IDS "






























SO.NO.	TESTS	OTHER	IDS
1	Test for Resource Exhaustion	Y	Y
2	IDS by Sending ARP Flood	Y	Y
3	Test the IDS by MAC Spoofing	Y	Y
4	IP Spoofing	Y	Y
5	Test by Sending a Packet to the Broadcast Address	Y	N
6	Inconsistent Packets	Y	N
7	Test IP Packet Fragmentation	Y	N
8	Duplicate Fragments	Y	N
9	Test for Overlapping Fragments	N	N
10	Ping of Death	Y	Y
11	Test for Odd Sized Packets	Y	Y
12	TTL Evasion	Y	N
13	Test by Sending a Packet to Port 0	Y	Y
14	UDP Checksum	N	N
15	Test for TCP Retransmissions	N	N
16	TCP Flag Manipulation	Y	N
17	Test TCP Flags	Y	Y
18	Test the IDS by Sending SYN Floods	Y	Y
19	Sequence Number Prediction	N	N
20	Test for Backscatter	Y	Y

21	Test the IDS With ICMP Packets	Y	Y
22	IDS Using Covert Channels	N	N
23	Test Using TCPReplay	N	N
24	Test Using TCPOpera	N	N
25	Test Using Method Matching	N	N
26	Test the IDS Using URL Encoding	N	N
27	Test the IDS Using Double Slashes	N	N
28	Test the IDS for Reverse Traversal	N	N
29	Test for Self Reference Directories	N	N
30	Test for Premature Request Ending	N	N
31	Test for IDS Parameter Hiding	N	N
32	Test for HTTP-Misformatting	N	N
33	Test for Long URLs	N	N
34	Test for Dos/Win Directory Syntax	N	N
35	Test for Null Method Processing	N	N
36	Test for Case Sensitivity	N	N
37	Test Session Splicing	N	N
38	Test heavy loads on server	Y	Y
39	Check for DoS vulnerable systems	Y	Y
40	Run SYN attack on server	Y	Y
41	Run Ping of Death	Y	Y
42	Run e-mail bomber on e-mail servers	N	N
43	Flood the website forms and guestbook with bogus entries	N	N
44	Place huge orders on e-commerce gateways and cancel before reaching the credit card screen	N	N
45	Locate the firewall	Y	Y
46	Traceroute to identify the network range	Y	N
47	Port scan the router	Y	Y
48	Grab the banner	Y	N
49	Create custom packets and look for firewall responses	Y	Y
50	Test Access Control Enumeration	Y	Y
51	Test to identify firewall architecture	Y	N
52	Test firewall using firewalking tool	Y	N
53	Test for Port Redirection	N	N
54	Test Covert channels	N	Y
55	Test HTTP Tunneling	N	N
56	Test Firewall specific vulnerabilities	N	N

8. Implementation

8.1 Implementation of the project

Planning :

		Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾	Predecessors
1			 feasibility study	5 days	Wed 15-01-14	Tue 21-01-14	
2			technical feasibility	2 days	Wed 15-01-14	Thu 16-01-14	
3			economical feasibility	2 days	Fri 17-01-14	Mon 20-01-14	
4			behavioural feasibility	1 day	Tue 21-01-14	Tue 21-01-14	
5			 requirement analysis	8 days	Wed 22-01-14	Fri 31-01-14	
6			requirement gathering	3 days	Wed 22-01-14	Fri 24-01-14	
7			group interaction	3 days	Sat 25-01-14	Tue 28-01-14	
8			analysis	2 days	Wed 29-01-14	Thu 30-01-14	
9			 UI design	20 days	Mon 03-02-14	Fri 28-02-14	
10			firewall	20 days	Mon 03-02-14	Fri 28-02-14	
11			honeypot	20 days	Mon 03-02-14	Fri 28-02-14	
12			Ddos	20 days	Mon 03-02-14	Fri 28-02-14	
13			log management	20 days	Mon 03-02-14	Fri 28-02-14	
14			 Coding	21 days	Mon 03-03-14	Mon 31-03-14	
15			firewall	21 days	Mon 03-03-14	Mon 31-03-14	
16			honeypot	21 days	Mon 03-03-14	Mon 31-03-14	
17			Ddos	21 days	Mon 03-03-14	Mon 31-03-14	
18			log management	21 days	Mon 03-03-14	Mon 31-03-14	
19			 testing	14 days	Tue 01-04-14	Fri 18-04-14	
20			unit testing	6 days	Tue 01-04-14	Tue 08-04-14	
21			integration testing	5 days	Wed 09-04-14	Tue 15-04-14	
22			system testing	3 days	Wed 16-04-14	Fri 18-04-14	

8.2 Post-Implementation and Software Maintenance

- From January 2014 to April 2014 - 5 versions of software is launched.
- Updates like Scanners & Honeypot are introduce in the software.
- Update like IPS Intrusion preventing system is introduce in the system.

9. Project Legacy

9.1 Current Status of the project

SO. NO	Project	Status
1.	Planning	Complete
2.	Information Gathering	Complete
3.	Design	Complete
4.	Source Code	Complete
5.	Testing	Complete
6.	Working	Complete

9.1 Remaining Areas of concern

- Firewall specific vulnerabilities
- HTTP Tunneling
- Covert channels
- Port Redirection
- Grab the banner
- Traceroute to identify the network range
- Sequence Number Prediction
- TCP Retransmissions
- UDP Checksum
- Overlapping Fragments

9.1 Technical and Managerial lessons learnt

- Create plans
- Justify Project
- Investigate options
- Plan Security
- Plan Access
- Training

10. User Manual

10.1 Introduction

Ever wonder someone spoofing around or accessing your data illegally or hacking your website? If answer to any of the above questions is a yes then you have come to the right place.

Intrusion Detection Systems help information systems prepare for, and deal with attacks. They accomplish this by collecting information from a variety of systems and network sources, and then analyzing the information for possible security problems. An intrusion detection system (IDS) examines system or network activity to find possible intrusions or attacks. Intrusion detection systems are either network-based or host-based.

Network based intrusion detection systems are most common, and examine passing network traffic for signs of intrusion.

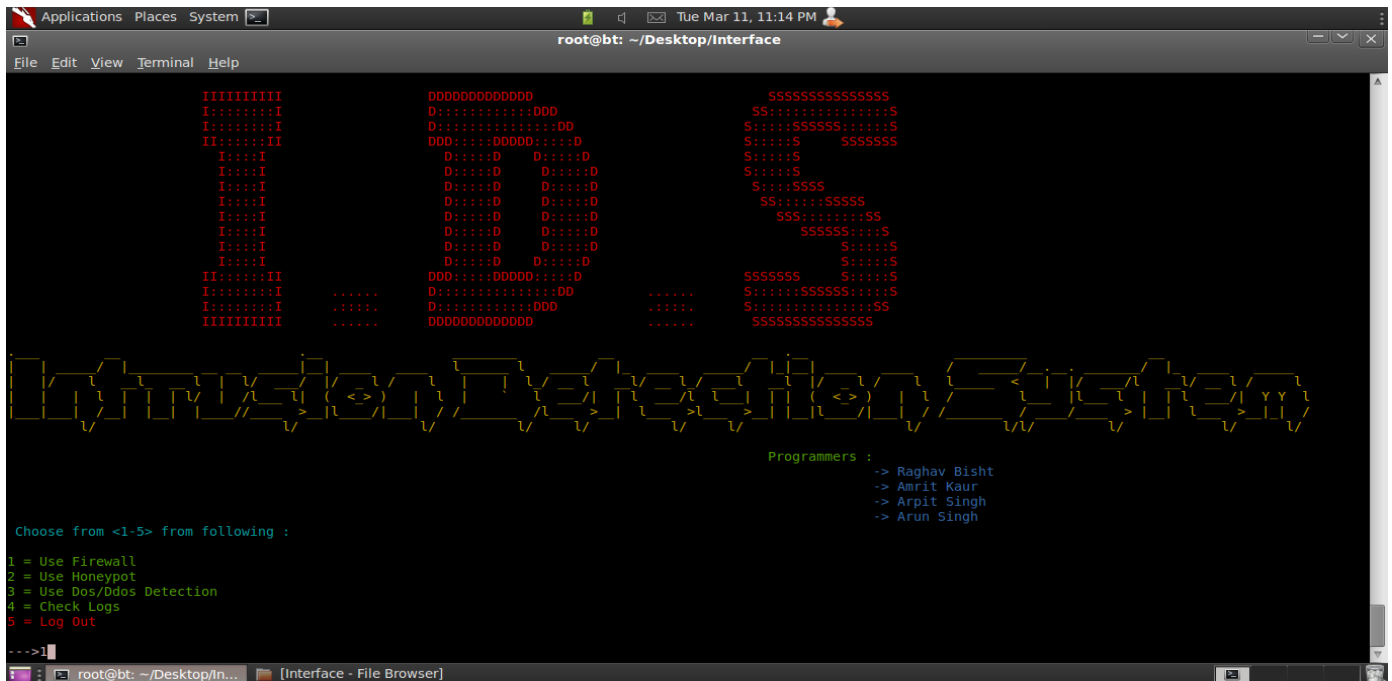
Host-based systems look at user and process activity on the local machine for signs of intrusion. Since each type has specific strengths and weaknesses.

10.2 Minimum requirements

Before installing the IDS make sure to check this section of the guide to check whether your PC is equipped with the required hardware to run the IDS efficiently.

Hardware	Minimum Specification	Recommend Specification
PROCESSOR	Intel core2duo or higher	Intel i3
MEMORY	512 MB DDR3	1 GB DDR3
DISK SPACE	50 MB	100 MB

10.3 The IDS Menu



```
Applications Places System root@bt: ~/Desktop/Interface
File Edit View Terminal Help

IIIIIIIIII DDDDDDDDDDD SSSSSSSSSSSSS
I:~::~:I D:~::~:DDD S:~::~:SSSSSSSSSS
I:~::~:I D:~::~:DD S:~::~:SSSSSSSSSS
II:~::~:II DDD:~::~:DDD S:~::~:S SSSSSS
I:~::~:I D:~::~:D D:~::~:D S:~::~:S
I:~::~:I D:~::~:D D:~::~:D S:~::~:S
I:~::~:I D:~::~:D D:~::~:D S:~::~:SSSS
I:~::~:I D:~::~:D D:~::~:D SS:~::~:SSSS
I:~::~:I D:~::~:D D:~::~:D SSS:~::~:SS
I:~::~:I D:~::~:D D:~::~:D SSSSS:~::~:S
I:~::~:I D:~::~:D D:~::~:D S:~::~:S
I:~::~:I D:~::~:D D:~::~:D S:~::~:S
II:~::~:II DDD:~::~:DDD SSSSSS S:~::~:S
I:~::~:I D:~::~:DDD S:~::~:SSSSSSSSSS
I:~::~:I D:~::~:DDD S:~::~:SSSSSSSSSS
IIIIIIIIII DDDDDDDDDDD SSSSSSSSSSSSS

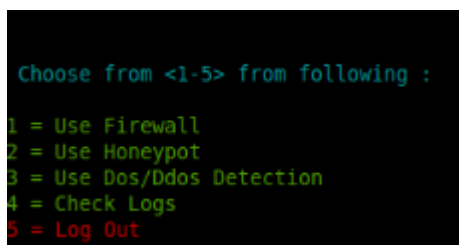
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

Programmers :
-> Raghav Bisht
-> Amrit Kaur
-> Arpit Singh
-> Arun Singh
```

1. Main menu

The main menu will have the following options



```
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out
```

1.1 Use Firewall

By selecting use firewall option i.e. selecting first option will enable the firewall option.

1.2. Use Honeypot

By selecting use honeypot option i.e. selecting second option will enable the firewall option

1.3. Use Dos/Ddos Detection

By selecting use Dos/Ddos option i.e. selecting third option will enable the firewall option.

1.4. Check logs

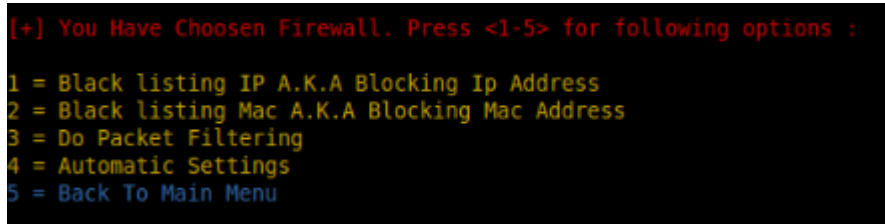
By selecting use Check logs option i.e. selecting first option will enable the firewall option.

1.5. Log out

By selecting log out option i.e. selecting the fifth option you will exit the IDS.

1.1. Use Firewall option

After selecting the use fire wall options you will have the following options shown in the figure

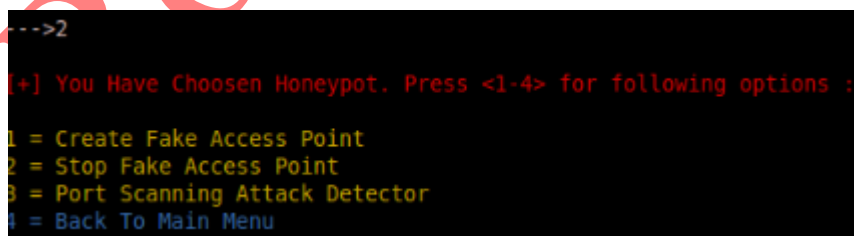


```
[+] You Have Chooosen Firewall. Press <1-5> for following options :  
1 = Black listing IP A.K.A Blocking Ip Address  
2 = Black listing Mac A.K.A Blocking Mac Address  
3 = Do Packet Filtering  
4 = Automatic Settings  
5 = Back To Main Menu
```

1. By selecting 1. Blocking IP address option will be enable.
This option enable the user to block the desired IP.
2. By selecting 2. Blocking mac address will be enable.
This option enable the user to block the desired mac address.
3. By selecting 3. Blocking port will be enable.
The option enable the user to block the desired ports.
4. By selecting 4. Block website option will be enable.
5. The option enable the user to block the desired website.
6. By selecting 5. Block Ads & Spamming IP Address will be enable
7. By selecting 6. Malware Or Evil Shell Finder will be enable
8. By selecting 7. Trojan scanner will be enable It will scan for any Trojan.

1.2. Use Honeypot option

After selecting the use honeypot options you will have the following options shown in the figure



```
--->2  
[+] You Have Chooosen Honeypot. Press <1-4> for following options :  
1 = Create Fake Access Point  
2 = Stop Fake Access Point  
3 = Port Scanning Attack Detector  
4 = Back To Main Menu
```

1. By selecting 1.Create fake accessing point will be enable..
This option enable the IDS to create fake access point.
2. By selecting 2.Stop fake accessing point will be enable
This option disable the fake access point created earlier.
3. By selecting 3. Port Scanning Attack Detector will be enable.

The option enable the user to block the desired ports.

4. By selecting 4. Back to main menu.

This option will redirect user to the main menu.

1.3 Use Dos/Ddos

After selecting the Use Dos/Ddos Detection options you will have the following options shown in the figure.

```
[+] You Have Chosen Dos / Ddos Detection. Press <1-10> for following options :  
1 = Check For ARP Poisoning  
2 = Check For HTTP Ddos  
3 = Check For UDP Ddos  
4 = Check For TCP / IP Ddos  
5 = Check For SYN Ddos  
6 = Check For Establish Connection Attack  
7 = Block Dos-Ddos Automatic  
8 = Check Flooding or Dos Attack On Specific Port  
9 = Check Ping Of Death Attack  
10 = Back To Main Menu
```

1. By selecting 1. Check For ARP Poisoning option will be enable

This option enable the user to block the desired IP.

2. By selecting 2. Check for HTTP Ddos will be enable.

This option enable the user to block the desired mac address.

3. By selecting 3. Check for UDP Ddos will be enable.

The option enable the user to block the desired ports.

4. By selecting 4. Check for TCP / IP Ddos will be enable.

5. By selecting 5. Check for SYN Ddos will be enable.

6. By selecting 6. Check for Establish Connection Attack will be enable.

7. By selecting 7. Block Dos-Ddos Automatic will be enable.

8. By selecting 8. Check Flooding or Dos Attack on Specific Port will be enable.

9. By selecting 9. Check Ping of Death Attack will be enable.

10. By selecting 10. Back to main menu.

This option will redirect user to the main menu.

1.4 UseLogs

After selecting Check logs options you will have the following options shown in the figure.

```
-->4

[+] You Have Chooosen Log File Monitoring. Press <1-7> for following options :

1 = Check Modification Time/date For Files
2 = Check Permission Modification
3 = Check Unexplained Changes In The File's Size
4 = Modification To System boot logs
5 = Clean System logs
6 = Check & Give Threshold Password To Accounts
7 = Back To Main Menu
```

1. By selecting 1. Check Modification Time/date For Files option will be enable
2. By selecting 2. Check Permission Modification will be enable.
3. By selecting 3. Check Unexplained Changes in the File's Size will be enable.
4. By selecting 4. Modification to System boot logs will be enable.
5. By selecting 5Clean System logs will be enable.
6. By selecting 6. Check & Give Threshold Password to Accounts will be enable.
7. By selecting 7. Back to main menu.

This option will redirect user to the main menu.

Snapshots

Here are some of the snapshots of the Intrusion Detection System

Firewall



Fig 11.1

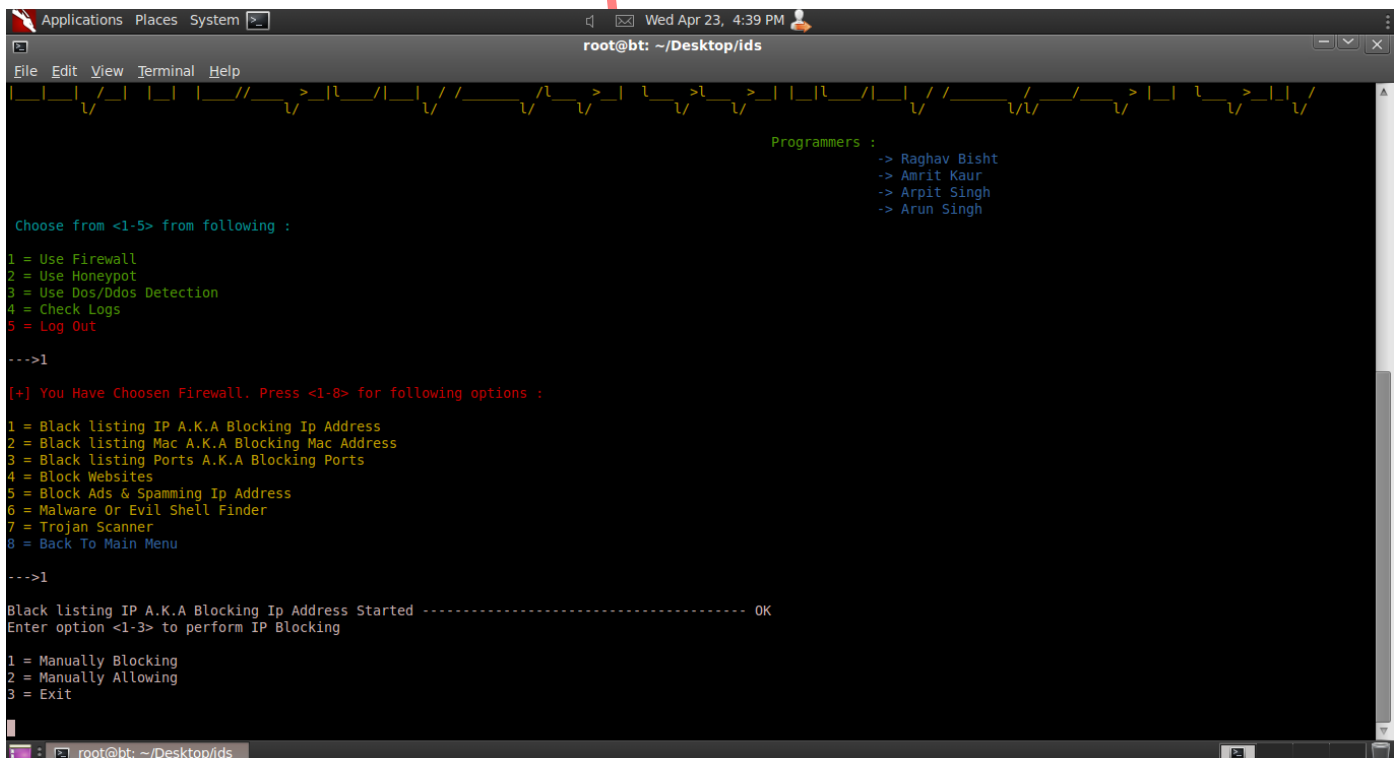


Fig 11.2

```
Applications Places System [x]
root@bt: ~/Desktop/ids

File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use HoneyPot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->1

[+] You Have Chooosen Firewall. Press <1-8> for following options :

1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu

--->2

Black listing Mac A.K.A Blocking Mac Address Started ----- OK
Enter option <1-5> to perform MAC Blocking

1 = Manually Blocking MAC Address
2 = Manually Allowing MAC Address
3 = Manually Allowing SSH Access to Specific MAC Address
4 = Manually Denying SSH Access to Specific MAC Address
5 = Exit

[
```

Fig 11.3

```
Applications Places System [x]
root@bt: ~/Desktop/ids

File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use HoneyPot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->1

[+] You Have Chooosen Firewall. Press <1-8> for following options :

1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu

--->3

Port Blocking Started ----- OK
Enter option <1-5> to perform MAC Blocking

1 = Manually Blocking / Close Incoming Port
2 = Manually Blocking / Close Outgoing Port
3 = Manually Allowing / Open TCP Incoming Port
4 = Manually Allowing / Open UDP Port
5 = Exit

[
```

Fig 11.4

```
Applications Places System
root@bt: ~/Desktop/ids

File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->1

[+] You Have Chooosen Firewall. Press <1-8> for following options :

1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu

--->4

Website Blocking Started ----- OK
Enter option <1-3> to perform to Block Website

1 = Manually Blocking Website
2 = Manually Unblocking Website
3 = Exit
Enter :
http:// facebook.com
```

Fig 11.5

```
Applications Places System
root@bt: ~/Desktop/ids

File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->1

[+] You Have Chooosen Firewall. Press <1-8> for following options :

1 = Black listing IP A.K.A Blocking Ip Address
2 = Black listing Mac A.K.A Blocking Mac Address
3 = Black listing Ports A.K.A Blocking Ports
4 = Block Websites
5 = Block Ads & Spamming Ip Address
6 = Malware Or Evil Shell Finder
7 = Trojan Scanner
8 = Back To Main Menu

--->6
```

Fig 11.6

Honey Pot

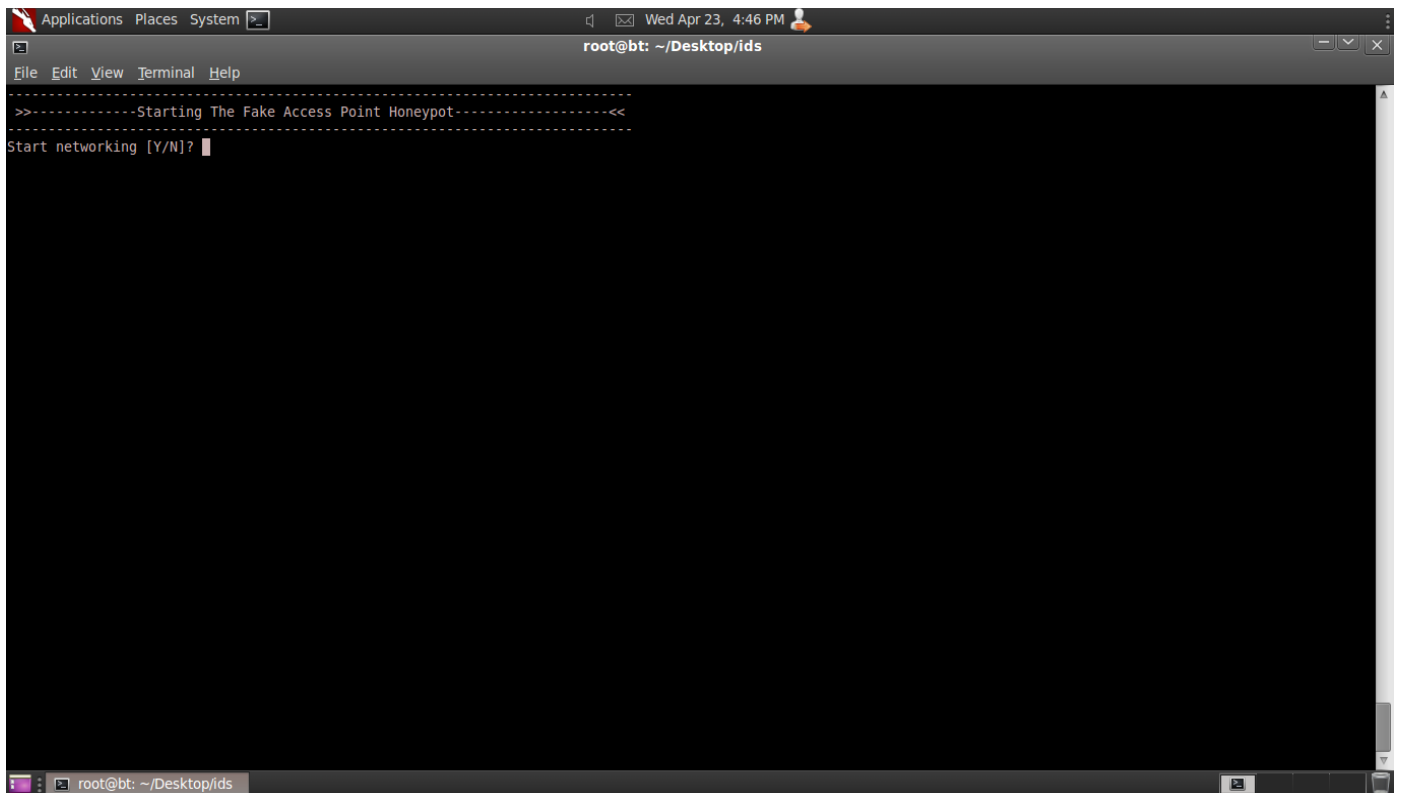


Fig11.9

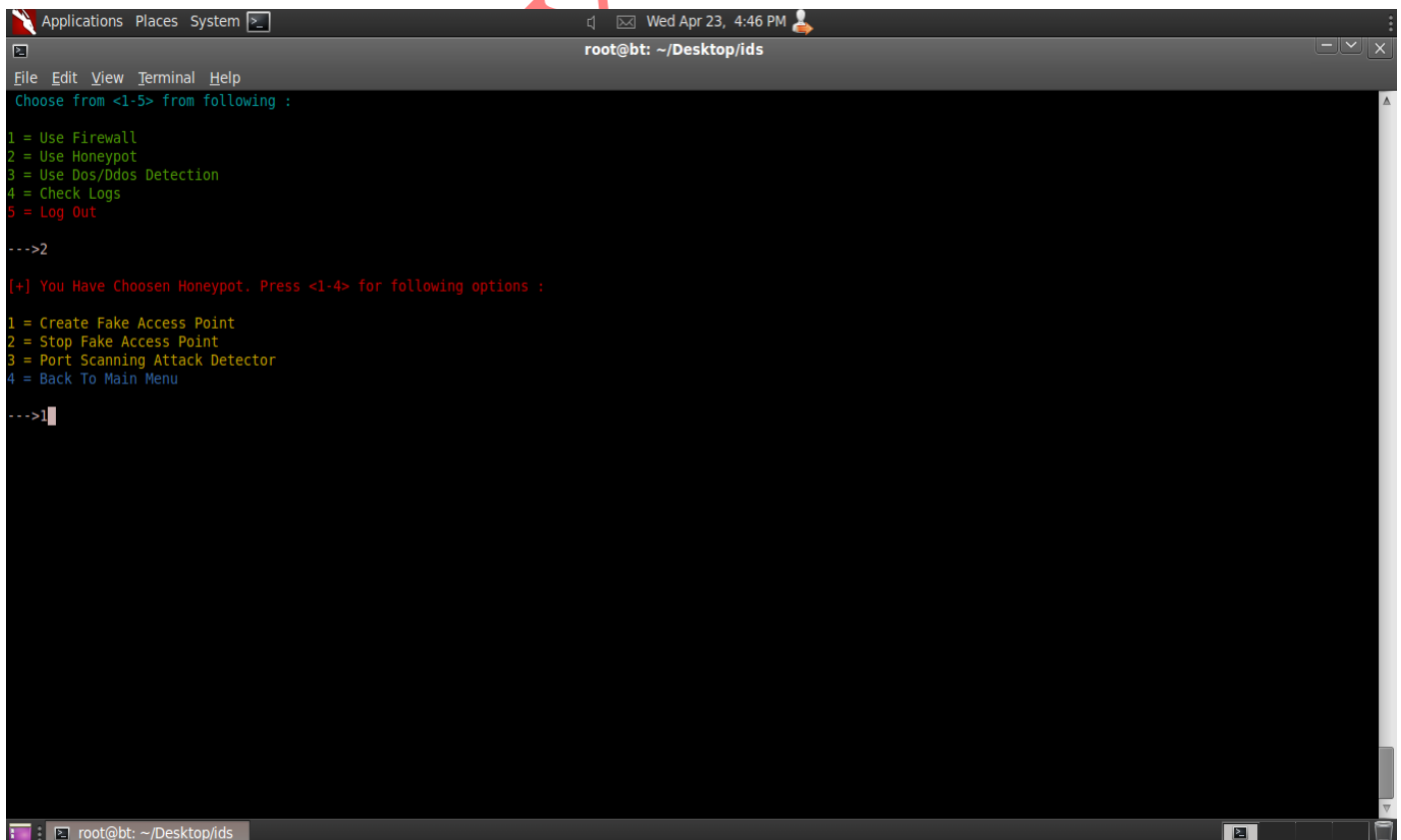


Fig 11.9

```
Applications Places System [x]
root@bt: ~/Desktop/ids

File Edit View Terminal Help
Clean Killing Everything...
-----
Killing routing...          DONE
Stopping DHCP server...    DONE
Killing external access...  DONE
Shutting down the fake AP... DONE
Stopping monitor device...  DONE
Killing logs...            DONE
-----

Shutdown complete!

[+] You Have Chooosen Honeypot. Press <1-4> for following options :

1 = Create Fake Access Point
2 = Stop Fake Access Point
3 = Port Scanning Attack Detector
4 = Back To Main Menu

--->|
```

Fig 11.10

```
Applications Places System [x]
root@bt: ~/Desktop/ids

File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->2

[+] You Have Chooosen Honeypot. Press <1-4> for following options :

1 = Create Fake Access Point
2 = Stop Fake Access Point
3 = Port Scanning Attack Detector
4 = Back To Main Menu

--->2|
```

Fig 11.11

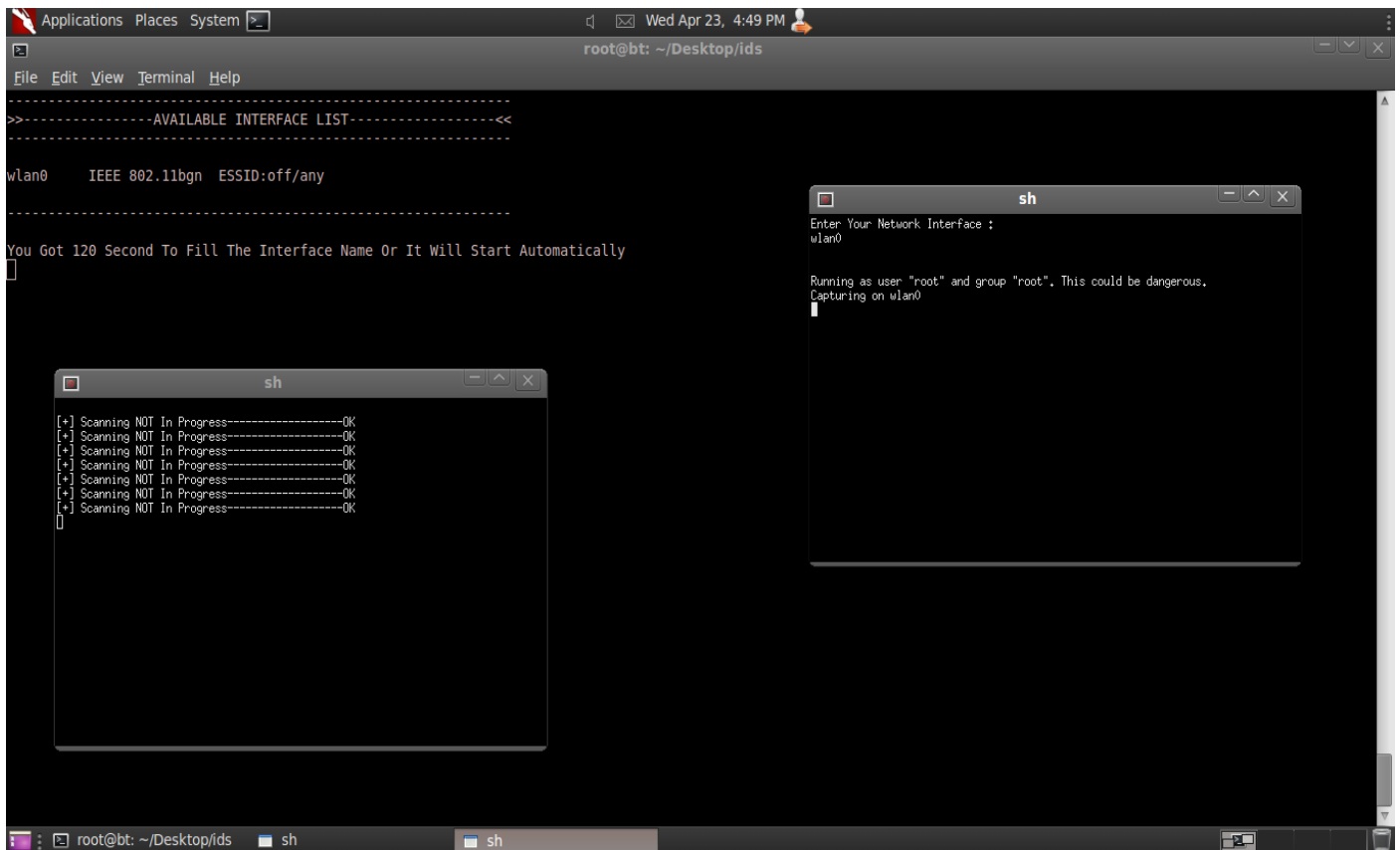


Fig 11.12

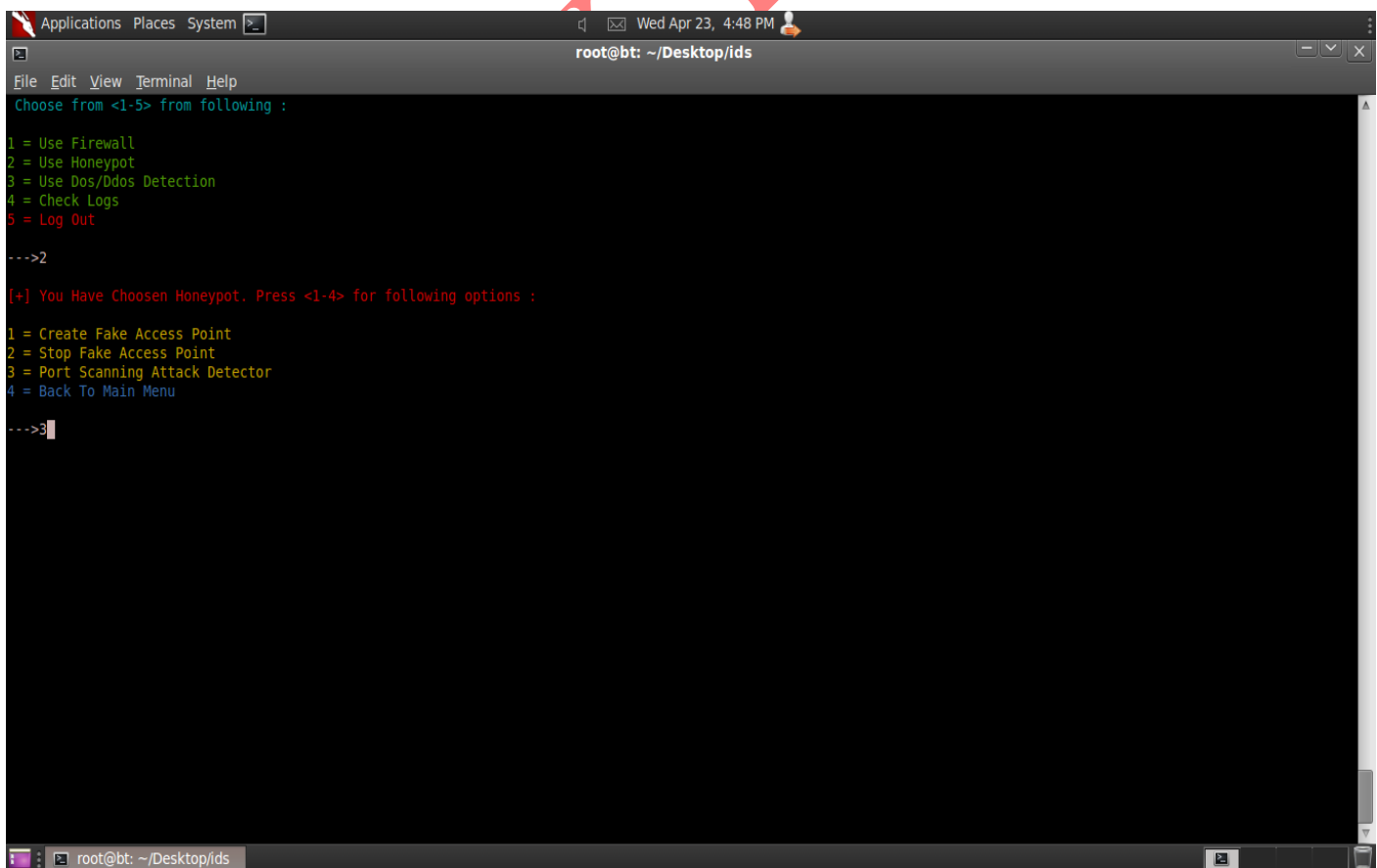
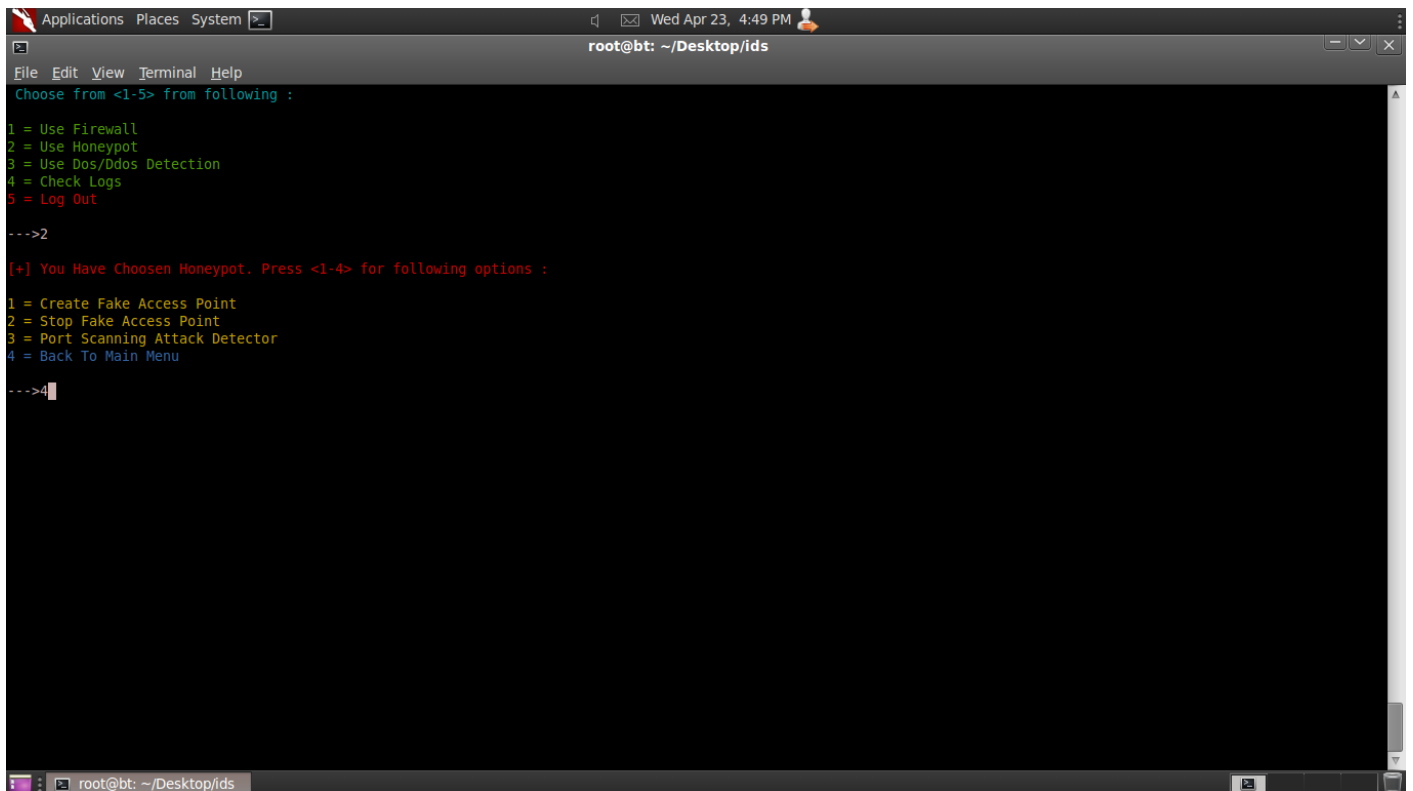


Fig 11.13



```
Applications Places System
root@bt: ~/Desktop/ids
File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

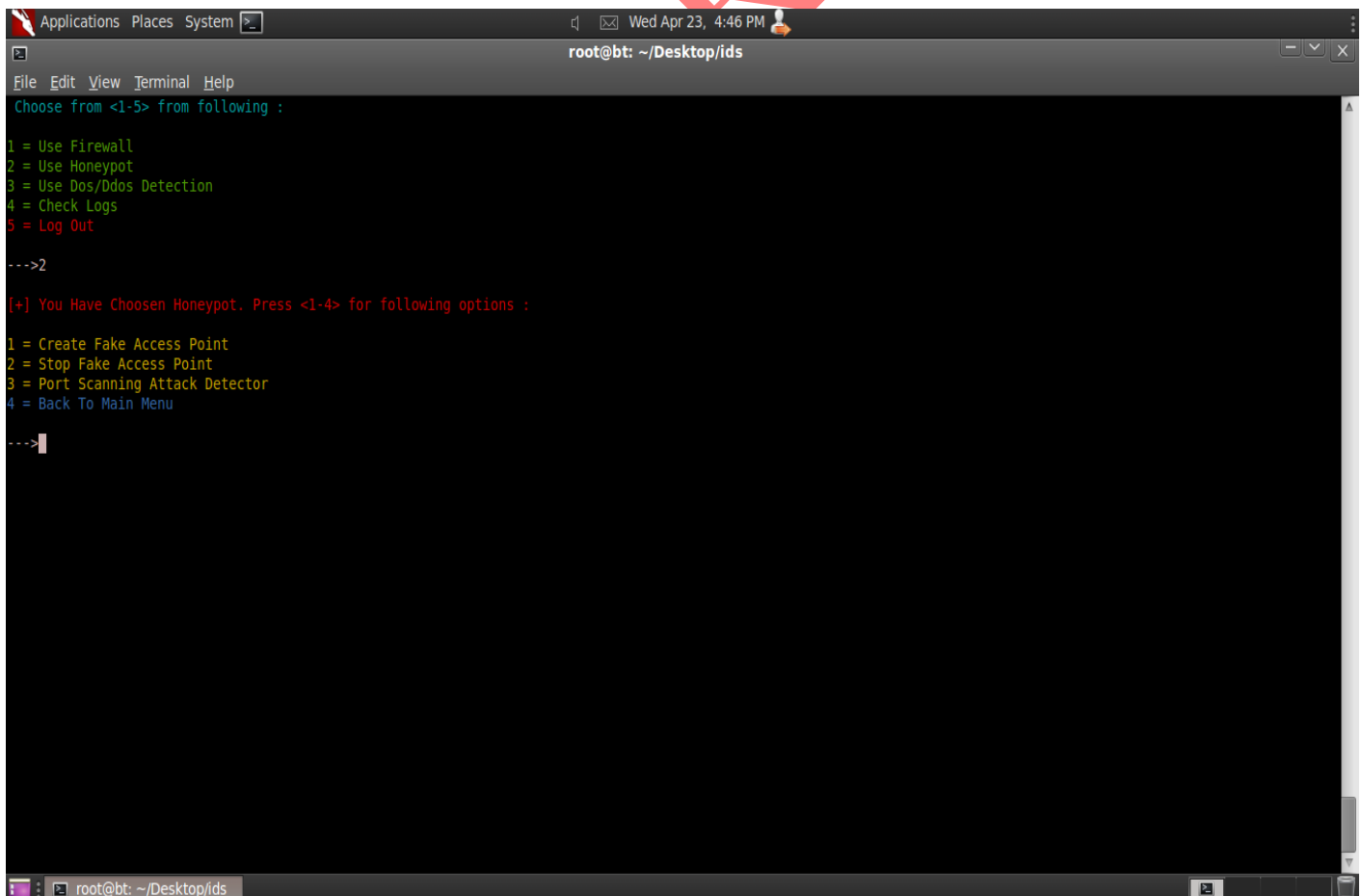
--->2

[+] You Have Chosen Honeypot. Press <1-4> for following options :

1 = Create Fake Access Point
2 = Stop Fake Access Point
3 = Port Scanning Attack Detector
4 = Back To Main Menu

--->4
```

Fig 11.14



```
Applications Places System
root@bt: ~/Desktop/ids
File Edit View Terminal Help
Choose from <1-5> from following :

1 = Use Firewall
2 = Use Honeypot
3 = Use Dos/Ddos Detection
4 = Check Logs
5 = Log Out

--->2

[+] You Have Chosen Honeypot. Press <1-4> for following options :

1 = Create Fake Access Point
2 = Stop Fake Access Point
3 = Port Scanning Attack Detector
4 = Back To Main Menu

--->
```

Fig 11.15

12. REFERENCES

http://en.wikipedia.org/wiki/White-box_testing

http://en.wikipedia.org/wiki/Functional_testing

Root-X Raghav