

# IP TRACEBACK OF DENIAL OF SERVICE (DOS) ATTACKS USING MOBILE AGENTS - TRACEABILITY IN E-SERVICES

Ghada El-Keissi                      [gh\\_elkeissi@hotmail.com](mailto:gh_elkeissi@hotmail.com)  
Sherif El-Kassas                     [sherif@aucegypt.edu](mailto:sherif@aucegypt.edu)  
*Computer Science Department, American University in Cairo, Egypt.*

**Abstract:** A current important network threat is the launch of Denial of Service (DoS) attacks. The main problem behind such attacks is the ability of an attacker to spoof his IP address. Thus, it's very difficult to identify the actual attacker. Accordingly, tracing back an attacker to actual source became a very important step to respond to DoS attacks. This paper aims at introducing an improved technique in tracing back IP spoofed flooding attack by using mobile agent technology. The work presented is an improvement to work described in [1] and [2]. Attacker would be traced back through three different network topologies: LAN, Inter-connected Network and WAN. The agent system is made up of four different agents: Master, Manager, Sensor, and Tracer agents. Every LAN would have independent trace back system deployed on to it. Different trace back systems in different LANs would cooperate to trace back an attacker within inter-connected networks and WAN. Similar to work presented by [2], the trace back system is mainly based on the idea of using Data link-level identifier (MAC address) to identify the next hop in the trace path. This comes from the fact that it's common and easy to spoof IP address unlike MAC addresses. Experiment on real network topology has been conducted and result is described in the paper.

**Keywords:** IP traceback, Denial-of-service attacks, Mobile agents, IP spoofing.

## **1. INTRODUCTION**

DoS attack is an attack that denies the target victim the ability to offer services to legitimate users. Many sites have been subject to such destructive attacks such as yahoo, cnn, and amazon. Most of the DoS attacks tend to be flooding attacks where the attacker uses spoofed IP addresses. Thus actual source of the attack couldn't be identified. Currently, there are many techniques used to respond to such destructive attacks. In this paper an improved technique based on work presented by [1] and [2] is presented. Its main aim is tracing back an attacker who launches a flooding attack using spoofed IP address. An experiment is described in details proving the ability of this improved method to identify actual source of the attack or at least the closest point to an attacker.

The rest of the paper will be organized as follows: Section 2 and Section 3 define DoS attacks and IP spoofing respectively. Section 4 describes the two categories of DoS attacks responsive mechanisms. Section 5 identifies an important responsive method which is IP trace back. It covers some common trace back methods used. Section 6 describes mobile agent technology. Section 7 outlines some related work in the field of IP trace back using mobile agent technology and data-link level identifiers in trace process. Section 8 presents in details the improved trace back technique implemented. Section 9 lists one of the experiments conducted and results achieved. Finally, section 10 summarizes findings and describes limitations and future work to be conducted.

## **2. DOS ATTACKS**

As explained above, DoS is an attack that denies a target victim the ability to offer its services to legitimate users. It can achieve this in many ways e.g. flooding the network preventing legitimate traffic, disrupting connection between two machines preventing access to services, or disrupting service to system or user [3]. Thus the target victim resources could be fully consumed or even the system could crash making it unable to serve legitimate users. Example of the DoS attacks are UDP attacks [5], and TCP/SYN attacks [6].

## **3. IP SPOOFING**

The widespread of DoS attack has emerged mainly due to weakness of security mechanisms implemented at different sites. Sites don't maintain and

update their security patches and anti-viruses, while attack tools have become highly advanced. Yet, the major reason for widespread of DoS attack is considered the ability of an attacker to send attacks to a victim using spoofed source IP address (i.e. fake IP address) hiding his true identity. This task has become a trivial one with the increase of the Internet size and number of IP addresses that could be easily forged by an attacker. With IP spoofing the victim finds it very difficult to take countermeasures against attacks. This comes from the fact that the actual source of an attack couldn't be determined, and it became difficult to decide whether the incoming packets are attack packets or legitimate ones. Accordingly the victim cannot decide whether to block an incoming traffic or not.

## **4. DOS RESPONSE TECHNIQUES**

Many techniques have been used to mitigate the effect of DoS attacks. The techniques could be divided in to two categories: Proactive and Reactive. The proactive approach consists of techniques that deals and prevents an attack before they actually happen. Adding filters to routers, updating security patches at network hosts and using advanced intrusion detection tools makes good preventive methods. On the other hand, reactive approach tries to handle attack and mitigate its effect during and after the attack has taken place. IP traceback is one main reactive technique. This paper concentrates on reactive technique, mainly IP traceback, since it is considered more challenging and important solution to prevent DoS attacks compared to proactive techniques. This emerges from the fact that proactive methods can prevent an attack from happening at target host but don't stop an attack forever. Still attacker is there with the spoofed IP address and he is not discouraged to continue attacking other sites. Thus proactive techniques are not the best solutions, as they don't eliminate the problem as a whole.

## **5. IP TRACEBACK**

IP traceback system is not easy to implement since it requires to traceback attacks from a LAN to another until the source of the attack is reached. This would require having a portable, extensible system that can spawn heterogeneous systems. Not only this, the IP traceback system implemented on target hosts could be an interesting target for DoS attack itself. An attacker could detect the components of traceback system and conduct flooding attack targeting it. In this case, the traceback system must be attack resistant. Many papers have described various techniques to

implement traceback systems. Some of the current IP Traceback systems described at [8] are as follows:

1. Link Testing: tracing starts at router closest to the victim and is repeated recursively on upstream routers. It stops when an attacker is identified, or trace leaves border of an ISP. The main disadvantage is that traceback process must occur during an attack, not after it.
2. Logging: logs packet information at key routers and then uses data mining techniques to determine the path the packet has traversed [8]. This is considered an expensive technique, due to high resource requirement.
3. Marking Algorithm: Burch and Cheswick have suggested a way to traceback an attack by marking packets probabilistically or deterministically with address of routers they traverse. Accordingly, the victim would use information in marked packets to traceback the source of the attack to its origin. The main disadvantage is that a packet might not have enough space for marked addresses.

Most of these developed techniques have many problems associated with them. In order to overcome the problems we need to build inexpensive, portable, extensible, attack resistant system that could spawn heterogeneous systems, and be able to trace an attack during or after an attack has been completed. Mobile agent technology was introduced to develop traceback systems.

## 6. MOBILE AGENTS

Mobile agents are mainly software components that can execute certain tasks. Agents reside at agent platforms that constitute their execution environment. The main agent's feature is its mobility and portability. The agent could move from one place to another and is portable. Its execution doesn't rely on the underlying OS and could spawn heterogeneous systems. Another important feature is that agents overcome network latency problem. In time of DoS attacks the network will already be congested. Tracing back attack might require transfer of data from one location to another. The use of mobile agents has overcome the problem of transferring bulky data at time of attack. Instead of transferring data to computation place, computation, in form of agent, is transferred to data. Of course agents would be much, much smaller in size compared to data. Example of mobile agent system is IBM Aglets [4].

Tracing intruders using mobile agents is considered a very efficient technique to solve DoS attacks. Midori, Shunji and Atsushi from Waseda University has tackled this problem and presented a solution [1]. Their work

was mainly to detect and traceback LAN attacks (i.e. attacks performed by users who have access to network machines) by using mobile agents. Based on their work, this paper will present an improved technique to traceback DoS attacks using mobile agents within Inter-connected network and WAN.

## **7. RELATED WORK**

### **7.1 Tracing Intruders Using Mobile Agents**

The Information-technology Promotion Agency (IPA) in Japan has developed an Intrusion Detection Agent (IDA) System [1] that could collect and gather information about intrusions and trace attackers with the help of mobile agents. The system is implemented fully to detect and respond to LAN attacks. The IDA system as described by [1] consists of manager, sensor, bulletin board, message boards, tracing agents, and information gathering agents.

When the sensor on a target host detects an MLSI (Marks Left by Suspected Intruder), the sensor would report that to the manager. Accordingly the manager would launch a tracing agent to that target. The tracing agent would launch an information-gathering agent on the target. The information-gathering agent would start to gather information about the MLSI on the target host. Tracing agent would try to identify the source origin of the attack using information about the network connections and processes running on the system. Independent of the tracing agent the information-gathering agent would report back information to the manager. The tracing agent would move to next hop in the tracing route until either it finds the attacker origin site or cannot move elsewhere. At this point it returns back to the manager. The tracing agents use message board to avoid the overlap of tracing routes by other tracing agents. Tracing agents use the message board to determine their path and destination.

### **7.2 Distributed IP TraceBack**

A proposed architecture presented by [2] relies on hop-by-hop tracing where routers log packet information and keep this data to traceback attacks later on. The approach used goes beyond this where datalink-level identifiers such as Ethernet media access control (MAC) address, ATM virtual path channel identifier (VPI/VCI), and frame relay datalink connection identifier (DLCI) are used to identify the packet path [2]. The idea is mainly based on the fact that an attacker can easily spoof the source IP address but it will be extremely difficult for him to spoof the datalink-level identifier of the forwarding node. The forwarding node changes the packet

datalink-level identifier to match its interface identifier and by looking at the identifier in the packet the forwarding node can know the adjacent node through which the packet has passed through. The forwarding nodes, tracers, would keep information about the packet and its datalink-level identifier, in a buffer memory (also known as packet information area) [2]. Then it will identify the adjacent node by matching the datalink-level identifier of forwarded packet with that of the attack packet. The system architecture is composed of sensor, manager and tracer. It operates in a similar manner as [1] but by using distributed management approach [2] and not mobile agents.

## **8. IMPROVED IP TRACEBACK USING MOBILE AGENTS**

Traceback system developed is an improvement to work presented by [1] and [2]. The traceback would cover LAN, Inter-connected networks, and WAN. The source of DoS attack would be traced back even if attacker uses spoofed IP address and creates flooding attacks. This is achieved by using datalink-level identifier (Mac address) and packet timestamps to identify next hop in trace path. Mobile agent technology has been used to implement the traceback system, mainly IBM Java Aglets agent system. For tracing to work successfully, agent platform must be deployed on all domains in WAN. This would enable agents to be dispatched and execute successfully on different LANs. An agent would fail to be dispatched to a LAN that doesn't have the agent platform installed on to it.

### **8.1 Architecture**

The architecture is similar to that presented by [1]. The system is composed of Manager, Sensor, and Tracing agents. There is also an application implemented using Java Packet Capture (JPCap) library [7] designed to capture packets going in and out of a LAN. It works along with the sensor agent in detecting an attack. The system is based on having different domains. Each domain will have a machine acting as the manager of that domain. This manager machine is also called an Agent Station. The Agent station of each domain will have a sensor agent and Java Packet Capture tool running on it. They work together to detect an attack. All other machines in the domain will have sensor agents running and trying to detect attacks too. Only an Agent Station can create tracing agents and send messages to other Agent Stations in different domains. Each Agent Station has a file called Ether\_IP.txt. This file contains information (IP address and Mac Addresses) of the following:

- Machines in the domain.
- Routers connecting the domain to other domains in Inter-connected networks.
- Agent Stations in different domains that are connected to those routers.

Based on such information, the tracing agent could identify the next hop. Obviously, it's not feasible to keep all information (MAC Address) of routers and other Agent Stations in a WAN at each individual Agent Station. Thus to extend search in a WAN, each Agent Station keeps information about a remote WAN "controller". This controller acts as a web server, and could be located at different ISPs. The controller would have a bigger list of IP addresses of routers and Agent Stations in a WAN. WAN would contain many controllers. Different LANs could be designed to connect to different controllers. This would require high coordination and management system to update the controllers with new routers and Agent Stations IP addresses. The paper doesn't focus on details of such coordination and management system as it's considered one thing to be investigated in the future work.

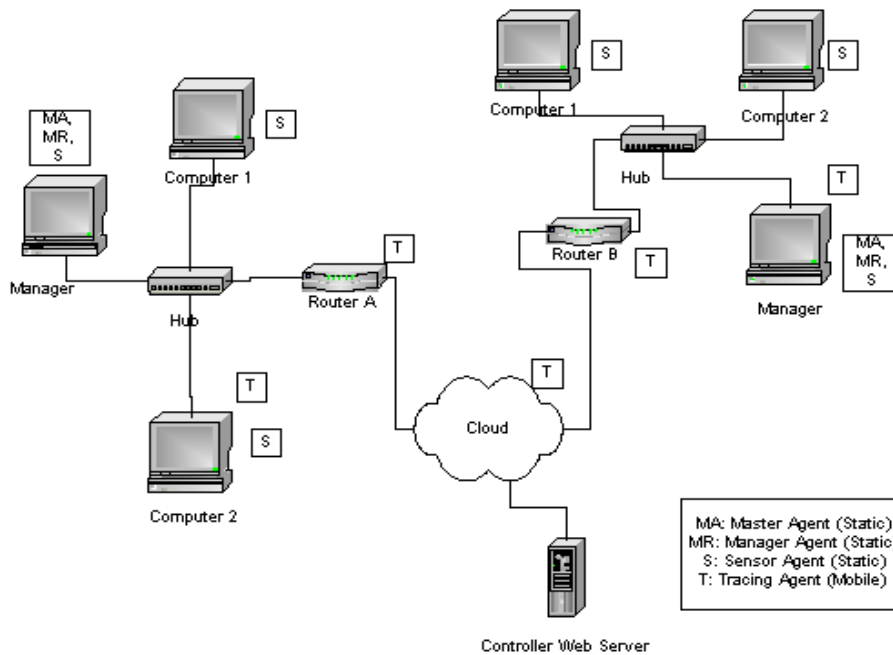


Figure 8.1 shows the traceback system architecture.

## 8.2 Tracing Technique

The Java Packet Capture tool is running on the Agent Station. It detects an attack if any of the following two situations happen:

- Both the source and destination ports of the attack packet are equal to 800. Port 800 was just chosen for testing purposes. It could have been set with any other port number.
- An attack packet is sent from same source to same victim at the Agent Station's domain for more than 20 times. The number is kept low (only 20) for testing. In real environment it should be kept higher than that.

Once an attack is detected, the Java Packet Capture tool sends a message to sensor agent running on the victim machine specifying that an attack has been detected. The sensor agent will receive the message and logs down the attack packet feature to a file. The sensor agent then sends the manager agent a message to start a new tracing process. It also sends the manager the attack packet features. The manager agent creates a new tracing agent and dispatches it to the victim. The tracing agent compares the source Mac of the attack packet against Mac addresses found at the Agent Station Ether\_IP.txt file. The result of the Mac address search is one of the following:

- Mac address is found to be of a local machine to the domain.
- Mac address is found to be of a router in an Inter-connected network and has a list of Agent Stations connected to it.
- Mac address isn't found in the list and considered to be of a router in the WAN. In this case controller must be contacted.

Accordingly, the tracing agent action would be as follows:

- If the Mac address is a local one, then the tracing agent will report to the manager the attacker is found locally and the tracing process will stop.
- If the Mac address is that of a router within the inter-connected network, the tracing agent will retrieve all Agent Stations (at different domains) connected to that router. It will visit each Agent Station, to examine log files for traces of the attack packet. The tracing agent will create list of all Agent Stations it has visited and had traces of the packets in their logs. The list will be sorted in descending order based on timestamp of the attack packets logged. The tracing agent will send this list to the manager agent of its home domain, requesting it to start new tracing process at these different domains. The manager will send a message to first Agent Station in the list to start a new tracing process. The manager will send that remote Agent Station the attack packet features too. The manager

will wait for results from the remote Agent Station. The result could be one of the following:

- Attacker is found at that agent station domain.
  - Attacker isn't found but new suspected domains are identified with new list of Agent Stations.
  - Attacker isn't identified, and there aren't any new agent stations lists.
- If the attacker is found then the overall tracing process would be stopped. If the attacker isn't found and there are new agent stations identified, then this new Agent Station list will be appended to the front of the existing list at the manager. The manager will retrieve next Agent Station in the list and will send a request for it to start a new tracing process. If the attacker isn't identified and there are no any new suspected domains, then the next Agent Station in the existing list will be retrieved. The manager will send it a message to start a new tracing process and will wait for the result.
- This tracing process is repeated until attacker is identified or the list becomes empty.
- If the Mac address isn't found in the Ether\_IP.txt file. Then it must be of a router that is considered to be linking the domain to WAN. Accordingly the tracing agent will contact main controller web server identified for that domain. The controller will have list of routers IP addresses and peer Agent Stations IP addresses. The tracing agent will send the list of all Agent Stations provided, to the manager requesting it to start new tracing process on all these remote Agent Stations domains. The manager will send tracing request to all Agent Stations along with the packet feature and will wait for the results.

## 9. EXPERIMENTS

Several tests were conducted using the American University in Cairo's network. The traceback covered different domains at the university. Tests have proved the ability of the implemented traceback system to identify actual attacker or at least closest point to the attacker. The attacker would create DoS flooding attack using spoofed IP address. Below is one of the conducted experiments and results. More experiments are described in thesis work.

## 9.1 Overview

The experiment is based on tracing back an attacker within three different domains. The attacker resides in one domain, while two victims reside on the two other domains. In this experiment, the attacker launches attack to two different domains. The two victims start a trace process until the actual attacker IP address is revealed. The experiment is divided in to two tests. The first test is conducted using an IP spoofing attack tool (check Appendix A.1). The second test is conducted using a flooding attack tool (check Appendix A.2). Time statistics is obtained for both tests. The tracing process executes while the attack is taking place.

## 9.2 Network Topology

The network topology implemented to conduct the test involves three domains.

### 9.2.1 First Domain – Attacker Domain

The domain is 172.25.3. It has two machines connected to a Hub.

- The attacker machine:

IP Address = 172.25.3.71

Mac Address = 00:08:74:e6:e1:a7

- The other machine is the Agent Station of the domain:

IP address = 172.25.3.70.

Mac Address = 00:03:47:a2:37:44

The Aglet Server is running on the Agent Station as well as the java packet capture application.

### 9.2.2 Second Domain – Victim Domain

The domain is 172.16.244. It has two machines.

- The victim machine:

IP Address = 172.16.244.21.

Mac Address = 00:03:47:29:a8:ff

- The other machine is the Agent Station of the domain:

IP Address = 172.16.244.20.

Mac Address = 00:03:47:2a:71:1c

Both machines have the Aglet Server running. The Agent Station has Java Packet Capture application running too.

### 9.2.3 Third Domain – Intermediate Domain

The domain is 172.25.5. It has one machine.

The machine is an Agent Station and victim of the domain:

IP Address = 172.25.5.201.

Mac Address = 00:03:47:a2:13:58

It has both Aglet Server and Java Packet Capture application running.

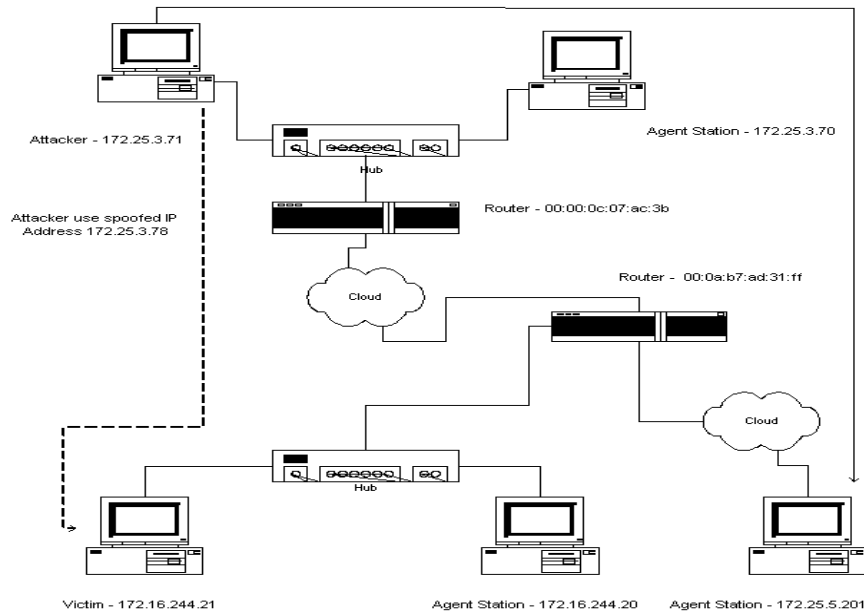


Figure 9.1: Experiment Network Topology

### 9.3 Attack/Trace Process Description

1. Attacker with IP address 172.25.3.71 starts an attack targeted to two victims with IP address 172.16.244.21 and 172.25.5.201. The attacker uses spoofed IP address 172.25.3.78. The attack packet feature is as follows:

```
1055578596:307462 172.25.3.78->172.16.244.21
protocol(6) priority(0) hop(46) offset(0)
ident(242) jpcap.EthernetPacket@50bd4d
00:0a:b7:a8:61:ff->00:03:47:29:a8:ff (2048)
```

2. Tracing process for attack towards 172.16.244.21
  - Java Packet Capture residing on the Agent Station 172.16.244.20 captures the network packets. It captures the attack.
  - The Agent station Java Packet Capture application sends an attack detection message to the sensor agent running on the victim 172.16.244.21. It also writes the packet features in a threshold file in c:\thersholds folder on the victim's machine.
  - The sensor agent sends a request to the manager agent at 172.16.244.20 to start a new trace process.
  - The manager agent creates a new tracing agent and sends it to the victim 172.16.244.21.
  - Tracing agent will read the threshold file and extract source Mac address of attack packet.
  - Tracing agent will compare the source Mac address to Mac addresses found in Ether\_IP.txt file. The Mac is found to be Mac of a router.
  - The tracing agent will extract the IP addresses of all Agent Stations linked to that router. In this test its two Agent Stations, 172.25.3.70 and 172.25.5.201.
  - The tracing agent will dispatch itself to the second domain having Agent Station 172.25.3.70.
  - The tracing agent arrives at Agent Station and checks logs created in c:\logs folder for packet traces.
  - The tracing agent finds traces for the attack packet on 172.25.3.70.
  - The tracing agent will dispatch itself to the third domain having Agent Station 172.25.5.201.
  - The tracing agent arrives at Agent Station and checks logs created in c:\logs folder for packet traces.
  - The tracing agent doesn't find traces for the attack packet on 172.25.5.201.
  - The tracing agent retracts back to its home domain Agent Station 172.16.244.20 and sends the manager agent a request to create a new trace process. It informs its manager about the remote domain where new tracing process should be created: 172.25.3.70.
  - The tracing agent stops tracing.
  - The manager at Agent Station 172.16.244.20 sends a message to manager at 172.25.3.70 requesting it to start a new trace process for the attack packet. It sends the manager at 172.25.3.70 the attack packet features.
  - The manager at 172.25.3.70 writes the packet to folders c:\PacketsVisited and creates a new tracing agent.

- The tracing agent reads the packet features from file created at the c:\Packets Visited folder. It compares the attack packet features against that found in the logs folder.
  - The packet is found in the logs saved.
  - The source Mac is retrieved from the logs and compared against those found in the Ether\_IP.txt file.
  - Tracing agent finds that Mac address corresponds to Mac address of a machine in the domain with IP address 172.25.3.71.
  - The tracing agent sends message to manager agent at 172.25.3.70 specifying that the attacker is found at IP address 172.25.3.71.
  - The manager agent at 172.25.3.70 sends a message to manager agent at 172.16.244.20 specifying that the attacker is found at IP Address 172.25.3.71
  - The tracing process stops and manager at 172.16.244.20 writes the result.
3. Tracing process for attack towards 172.25.5.201
- Java Packet Capture residing on the Agent Station 172.25.5.201 captures the network packets. It captures the attack.
  - The Agent station Java Packet Capture application sends an attack detection message to the sensor agent running on the victim. It also writes the packet features in a threshold file in c:\thersholds folder.
  - The sensor agent sends a request to the manager agent to start a new trace process.
  - The manager agent creates a new tracing agent and sends it to the victim 172.25.5.201.
  - Tracing agent will read the threshold file and extract source Mac address of attack packet.
  - Tracing agent will compare the source Mac address to Mac addresses found in Ether\_IP.txt file. The Mac is found to be Mac of a router.
  - The tracing agent will extract the IP addresses of all Agent Stations linked to the router. In this test it's only one 172.25.3.70.
  - The tracing agent will dispatch itself to the other domain having Agent Station 172.25.3.70.
  - The tracing agent arrives at Agent Station and checks logs created in c:\logs folder for packet traces.
  - The tracing agent finds traces for the attack packet on 172.25.3.70.
  - The tracing agent retracts back to its home domain Agent Station 172.25.5.201 and sends the manager agent a request to create a new trace process in other domain.
  - The tracing agent stops tracing.
  - The manager at Agent Station 172.25.5.201 sends a message to manager at 172.25.3.70 requesting it to start a new trace process for

the attack packet. It sends the manager at 172.25.3.70 the attack packet features.

- The manager at 172.25.3.70 writes the packet to folders c:\PacketsVisited and creates a new tracing agent.
- The tracing agent reads the packet features from file created at the c:\Packets Visited folder. It compares the attack packet features against those found in the logs folder.
- The packet is found in the logs saved.
- The source Mac is retrieved from the logs and compared against those found in the Ether\_IP.txt file.
- Tracing agent finds that Mac address corresponds to Mac address of a machine in the domain with IP address 172.25.3.71.
- The tracing agent sends message to manager agent at 172.25.3.70 specifying that the attacker is found at IP address 172.25.3.71.
- The manager agent at 172.25.3.70 sends a message to manager agent at 172.25.5.201 specifying that the attacker is found at IP Address 172.25.3.71.
- The tracing process stops and manager at 172.25.5.201 writes the result.

## 9.4 Time Statistics

### 9.4.1 Attacker Using IP Spoofing Tool

The network load was kept low during the IP spoofing attack. It was kept at maximum of 300 kbps.

<b>Total Trace Time at 172.16.244.20</b>	
<b>Start Time</b>	12:07:39:256
<b>End Time</b>	12:08:17:817
<b>Total Trace Time at 172.25.5.201</b>	
<b>Start Time</b>	12:06:22:205
<b>End Time</b>	12:08:00:926

### 9.4.2 Attacker Using Flooding Tool

The network load was kept high. The network load at 172.16.244.21 was 700 Kbps and at 172.25.5.201 was 1.3 Mbps.

<b>Total Trace Time at 172.16.244.20</b>	
<b>Start Time</b>	02:48:02:306
<b>End Time</b>	02:49:01:212

<b>Total Trace Time at 172.25.5.201</b>	
<b>Start Time</b>	03:48:22:660
<b>End Time</b>	03:49:07:294

## 10. CONCLUSION AND FUTURE WORK

The improved technique described in this paper has succeeded in tracing back and identifying an attacker under the following two conditions: attacker spoofs his IP address and attacker floods the network. It could traceback an attacker in LAN, Inter-connected networks, and WAN. Using mobile agents have made the system more attack resistant than distributed architecture described in [2]. If part of agent system gets corrupted, the rest can function properly. Also, during traceback process, huge amount of data must be analyzed. Mobile agents are usually of smaller size than the data to be analyzed. Thus, it's easier and more efficient to transfer agents to data during network congestion caused by flooding attack, than transferring data to computation. This would help in solving network latency problem. Another very important feature of the developed traceback system is its ability to traceback an attacker while an attack is taking place or even after it has been completed.

The implemented traceback system could spawn heterogeneous systems as it is coded using java agents. It doesn't require any special hardware. It also, doesn't require any changes in TCP/IP protocol. Only agent system and platform need to be deployed in each LAN as described above. The traceback process doesn't involve any costly router computation.

On the other hand, there are several limitations to the proposed technique that must be taken in to consideration in the future work. The limitations are as follows:

- Tests were conducted on The American University in Cairo's network. Due to limited resources and permissions, tests were conducted over network of three inter-connected LANs. In order to extend the experiments, a single LAN was used to simulate an environment of multiple LANs (i.e. a LAN was divided in to several LANs by having more than one independent Agent Station and separate traceback systems). Experiments could be extended to cover larger real network topologies across WAN.
- The traceback system must be deployed on almost every LAN in WAN. Otherwise, tracing agent will not be able to function properly.
- Timeout mechanism must be implemented at two levels: timeout of messages, and timeout of tracing agent.

- Security of mobile agents must be taken into consideration. There are several mobile agent security methods that can be used and are presented in [9].
- Having efficient packet logging system is important to make sure that data doesn't get lost. At time of a flooding attack, logs are kept in huge amount and some data may not be examined or even get lost. Efficient logging mechanism is required to make sure that the tracing agent could succeed in examining all data properly for finding actual attacker. In the above mentioned experiment, packet logging has been considered a problem. At high flooding attack, data was lost and sometimes the experiment failed in identifying the actual attacker (i.e. just stopped at closest point to attacker).
- Although MAC address spoofing isn't currently a common act, still it should be taken in to consideration for future work.
- Implement controllers' coordination and management system.

## 11. REFERENCES

[1] Asaka Midori, Shunji Okazawa, and Atsushi Taguchi(June 1999), A Method of Tracing Intruders by Use of Mobile Agents. Proc. INET 99.

[2] Baba, Tatsuya and Shigeyuki Matsuda (April 2002), Tracing Network Attacks To Their Sources. NTT Data Corporation, IEEE.

[3] Lau, Felix, Staurt Rubin, Michael Smith and Ljiljana Trajkovic. Distributed Denial of Service Attacks.

Aglets Specification 1.1 Draft. IBM Corporation, 1998.

[4] CERT Advisory(1996), UDP Port Denial-of-Service Attacks.  
<http://www.cert.org/advisories/CA-1996-01.html>

[5] CERT Advisory(1996), TCP SYN Flooding and IP Spoofing Attacks.  
<http://www.cert.org/advisories/CA-1996-21.html>

[6] JPCap(2003), Java Package for Packet Capture.  
<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>

[7] Savage, Stefan, David Wetherall, Anna Kerlin, and Tom Anderson.(2000), Practical Network Support for IP Traceback. University of Washington,.

[8] Jansen, Wayne and Tom Karygiannis(1999), Mobile Agent Security. National Institute of Technology.

[9] Libnet. Libnet Packet Assembly. [www.packetfactory.net/projects/libnet/](http://www.packetfactory.net/projects/libnet/)

[10] Misoskian Packet Builder. Attack tools and Protection.  
<http://www.angelfire.com/my/bulat/download.html>