SECURING CLOUD AVAILABILITY against DDoS ATTACK

Project Report Submitted in partial fulfillment of the Degree of

Master of Technology

In

Computer Science and Engineering

Under the Supervision of

Prof. S.P Ghrera Brig(Retd.)

&

Dr. Pradeep Kumar Gupta

By

Anamika Rangra (122215)

To

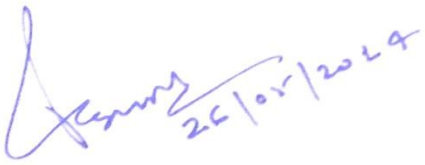
Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Table of Content

CERTIFICATE

This is to certify that the work titled **SECURING CLOUD AVAILABILITY against DDoS ATTACK** submitted by **Anamika Rangra (122215)** in partial fulfillment for the award of degree of **Master of Technology** of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.



Signature of Guide:

Name of Guide: **Dr. Pardeep Kumar Gupta**

Designation: **ASSISTANT PROFESSOR(CSED)**

Name of Co-Guide: **Prof. S.P Ghrera Brig(Retd)**

Designation: **HOD(CSED)**

Date: **26 May,2014**

ACKNOWLEDGEMENT

I sincerely express my gratitude to my advisor, **H.O.D Prof Brig. S.P. Gherera and Dr. Pradeep Kumar Gupta**, for consistent guidance in this research work. They are always there to tell me how to go about my project in a systematic manner and who always took out time to help me with my technical and non-technical doubts at various stages of the project.

Last but not the least; it was my fellow students who came to me rescue whenever I got stuck in any piece of code or otherwise.

Signature of the Student

Name of the Student **Anamika Rangra**

CHAPTER 1

Introduction

1.1 Introduction to Cloud Computing

Cloud computing is a recently developing paradigm of distributed computing. Though, it is not a new idea that emerged just recently. In 1969 L. Klein Rock anticipated, "As of now, computer networks are still in their infancy. But as they grow up and become more sophisticated, we will probably see the spread of 'computer utilities' which, like present electric and telephone utilities, will service individual homes and offices across the country." His vision was the true indication of today's utility based computing paradigm. One of the giant steps towards this world was taken in mid 1990s when grid computing was first coined to allow consumers to obtain computing power on demand. The origin of cloud computing can be seen as an evolution of grid computing technologies. The term Cloud computing was given prominence first by Google's CEO Eric Schmidt in late 2006 (May be he coined the term). So, the birth of Cloud computing is very recent phenomena although its root belongs to some old ideas with new business, technical and social perspectives. From the architectural point of view cloud is naturally build on an existing grid based architecture and uses the grid services and adds some technologies like virtualization and some business models.

A widely accepted definition of cloud computing stems from early work done by the National Institute of Standards and Technology (NIST), a U.S. Department of Commerce agency that promotes innovation and industrial competitiveness via measurement science, standards and technology. **"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction."**

In other words, Cloud computing allows organizations to provide their staff with access to the applications, infrastructure or platforms they need to do their jobs- all via a simple front-end interface, such as a web browser. They might need access to these resources for a few

minutes at a time or for a longer period. Depending on the deployment model in use, organizations can pay on a utility basis only for what they use.

In brief, Cloud is essentially a bunch of commodity computers networked together in same or different geographical locations, operating together to serve a number of customers with different need and workload on demand basis with the help of virtualization. Cloud services are provided to the cloud users as utility services like water, electricity, telephone using pay-as-you-use business model. These utility services are generally described as XaaS (X as a Service) where X can be Software or Platform or Infrastructure etc. Cloud users use these services provided by the cloud providers and build their applications in the Internet and thus deliver them to their end users. So, the cloud users don't have to worry about installing, maintaining hardware and software needed. And they also can afford these services as they have to pay as much they use. So, the cloud users can reduce their expenditure and effort in the field of IT using cloud services instead of establishing IT infrastructure themselves.

Cloud is essentially provided by large distributed data centers. These data centers are often organized as grid and the cloud is built on top of the grid services. Cloud users are provided with virtual images of the physical machines in the data centers. This virtualization is one of the key concept of cloud computing as it essentially builds the abstraction over the physical system. Many cloud applications are gaining popularity day by day for their availability, reliability, scalability and utility model. These applications made distributed computing easy as the critical aspects are handled by the cloud provider itself.

Cloud computing is growing now-a-days in the interest of technical and business organizations but this can also be beneficial for solving social issues. In the recent time E-Governance is being implemented in developing countries to improve efficiency and effectiveness of governance. This approach can be improved much by using Cloud computing instead of traditional ICT. In India, economy is agriculture based and most of the citizens live in rural areas. The standard of living, agricultural productivity etc. can be enhancing by utilizing cloud computing in a proper way. Both of these applications of cloud computing have technological as well as social challenges to overcome.

In this report we would try to clarify some of the ideas why is cloud computing a buzzword today? i.e., what are the benefits the provider and the users get using cloud? Though its idea has come long back in 1990 but what situation made it indispensable today? How is cloud built? What differentiates it from similar terms like Grid computing and utility computing? What are the different services are provided by the cloud providers? Though cloud computing now-a-days talks about business enterprises not the non-profit organizations; how can this new paradigm is used in the services like e-governance and in social development issues of rural India?

1.2 Cloud Computing Basics

Cloud computing provides the customers on-demand, utility based computing services. It can provide more reliable, available and updated services to its clients in turn. It consists of physical machines in the data centers of cloud providers. Virtualization is provided on top of these physical machines. These virtual machines are provided to the cloud users. Different cloud provider provides cloud services of different abstraction level. e.g., Amazon EC2 enables the users to handle very low level details where Google App Engine provides a development platform for the developers to develop their applications. So, the cloud services are divided into many types like Software-as-a-Service, Platform-as-a-Service or Infrastructure-as-a-Service. These services are available over the Internet in the whole world where the cloud acts as the single point of access for serving all customers. Cloud computing architecture addresses difficulties of large scale data processing.

1.3 Types of Cloud

Cloud can be of three types :

- **Private Cloud:** This type of cloud is maintained within an organization and used solely for their internal purpose. So, the utility model is not a big term in this scenario. Many companies are moving towards this setting and experts consider this is the first step for an organization to move into cloud. Security, network bandwidth are not critical issues for private cloud.
- **Public Cloud:** In this type an organization rents cloud services from cloud

providers' on-demand basis. Services provided to the users using utility computing model.

- **Hybrid Cloud:** This type of cloud is composed of multiple internal or external clouds. This is the scenario when an organization moves to public cloud computing domain from its internal private cloud.

1.4 Cloud Architecture

The cloud providers actually have the physical data centers to provide virtualized services to their users through Internet. The cloud providers often provide separation between application and data. The underlying physical machines are generally organized in grids and they are usually geographically distributed. Virtualization plays an important role in the cloud scenario. The data center hosts provide the physical hardware on which virtual machines resides. User potentially can use any Operating System supported by the virtual machines used.

Operating systems are designed for specific hardware and software. It results in the lack of portability of operating system and software from one machine to another machine which uses different instruction set architecture. The concept of virtual machine solves this problem by acting as an interface between the hardware and the operating system called as system Virtual Machine .

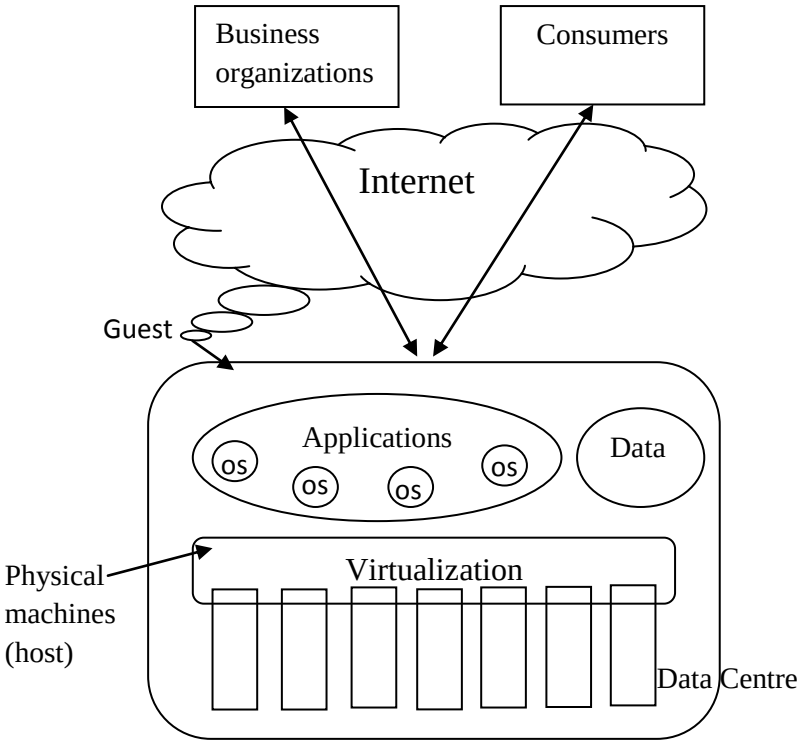


Figure 1: Basic Cloud Computing Architecture

CHAPTER 2

2.1 Background of DDoS

In DDoS attacks, attackers generate a huge amount of requests to victims through compromised computers (zombies), with the aim of denying normal service or degrading of the quality of services. It has been a major threat to the Internet since year 2000, and a recent survey on the largest 70 Internet operators in the world demonstrated that DDoS attacks are increasing dramatically, and individual attacks are more strong and sophisticated. Furthermore, the survey also found that the peak of 40 gigabit DDoS attacks nearly doubled in 2008 compared with the previous year.

While DDoS attacks have received increasing attention from both network operators and journalists in the past half-dozen years, the basic network vulnerabilities that make attacks possible have been recognized since the early days of the commercial web. Practical Unix and Internet Security, the "bible" for many system administrators of the early commercial web, offers a chapter on denial of service attacks. **Carnegie Mellon's Computer Emergency Response Team (CERT)** published its first bulletin on SYN flooding (a popular technique for overwhelming a target system) in September 1996, 12 and a more thorough bulletin on denial of service in October 1997,13 suggesting that denial of service was beginning to emerge as a priority for network administrators.

DDoS attacks are targeted at exhausting the victim's resources, such as network bandwidth, computing power, and operating system data structures. To launch a DDoS attack, the attacker(s) first establishes a network of computers that will be used to generate the huge volume of traffic needed to deny services to legitimate users of the victim. To create this attack network, attackers discover vulnerable hosts on the network. Vulnerable hosts are those that are either running no antivirus or out-of-date antivirus software, or those that have not been properly patched. These are exploited by the attackers who use their vulnerability to gain access to these hosts. The next step for the attacker is to install new programs (known as attack tools) on the compromised hosts of the attack network allowing them to control all these burgled machines to launch coordinated attacks on victim sites.

So the perpetrator starts by breaking into weakly-secured computers, using well-known defects in standard network service programs, and common weak configurations in operating systems. On each system, once they break in, they perform some additional steps. First, they install software to conceal the fact of the break-in, and to hide the traces of their subsequent activity. For example, the standard commands for displaying running processes are replaced with versions that fail to display the attacker's processes. These replacement tools are collectively called a ``rootkit", since they are installed once you have ``broken root", taken over system administrator privileges, to keep other ``root users" from being able to find you. Then they install a special process, used to remote-control the burgled machine. This process accepts commands from over the Internet, and in response to those commands it launches an attack over the Internet against some designated victim site. And finally, they make a note of the address of the machine they've taken over. All these steps are highly automated. A cautious intruder will begin by breaking into just a few sites, then using them to break into some more, and repeating this cycle for several steps, to reduce the chance they are caught during this, the riskiest part of the operation. By the time they are ready to mount the kind of attacks we've seen recently (gigabytes per second of traffic dumped on Yahoo, according to reports in SANS) they have taken over thousands of machines and assembled them into a DDoS network; this just means they all have the attack software installed on them, and the attacker knows all their addresses (stored in a file on their control system).

Now comes time for the attack. The attacker runs a single command, which sends command packets to all the captured machines, instructing them to launch a particular attack (from a menu of different varieties of flooding attacks) against a specific victim. When the attacker decides to stop the attack, they send another single command.

While there are variations, they generally take a common form. The controlled machines being used to mount the attacks send a stream of packets. For most of the attacks, these packets are directed at the victim machine. For one variant (called “smurf”, named after the first circulated program to perform this attack) the packets are aimed at other networks, where they provoke multiple echoes all aimed at the victim. To go into further detail, some background description of the Internet is in order.

The Internet consists of hundreds of thousands or millions of small networks (called Local Area Networks, or LANs), all interconnected; attached to these LANs are many millions of separate computers. Any of these computers can communicate with any other computer. This works by assigning every computer an address. The addresses are structured (organized into groups) so that special-purpose traffic-handling computers, called routers, can direct them in the right direction to reach their intended destination. A typical connection today may require 15 or more hops, crossing from one LAN to another, before it reaches its final destination. But most of these ``LANs'' are actually special-purpose links within and between network transport companies. These backbone providers handle the hard problems of routing traffic.

Looking a little closer, when one computer wants to send a message to another, it divides it into fixed-size pieces, called ``packets''. Each of these packets is handled separately by the Internet, then the message (if it is larger than a single packet) is reassembled at the remote computer. So the traffic passing between machines consists entirely of packets of data. Each of these packets has a pair of addresses in it, called the Source and Destination IP (for Internet Protocol) addresses. These are the addresses of the originating machine, and the recipient. They are quite analogous to the address and return address on an envelope, in traditional mail.

When such a packet is sent over the Internet, it is passed first to the nearest router; commonly this router is at the point where the local network connects to the Internet. This router is often called a border router. In larger organizations the story may be more complex; a large organization often assembles its own collection of LANs, interconnected into an in-house internet, cross-connected at one or more points (often with firewalls) with the Internet that we all know and love. But returning to our tale, when a packet leaves a computer, it is passed to a border router. This router passes it upstream to a core router, which interconnects with many other core routers all over the Internet; they pass the packet on until it reaches its destination. The source address is normally ignored by routers; it normally only tells the final destination machine where the request is coming from. That's an essential part of the problem we face today.

The packets used in today's DDoS attacks use forged source addresses; they are lying about where the packet comes from. The very first router to receive the packet can very easily catch the lie; it has to know what addresses lie on every network attached to it, so that it can correctly route packets to them. If a packet arrives, and the source address doesn't match the network it's

coming from, the router should discard the packet. This style of packet checking is called variously Ingress or Egress filtering, depending on the point of view; it is Egress from the customer network, or Ingress to the heart of the Internet. If the packet is allowed past the border, catching the lie is nearly impossible. Returning to our analogy, if you hand a letter to a letter-carrier who delivers to your home, there's a good chance he could notice if the return address is not your own. If you deposit a letter in the corner letter-box, the mail gets handled in sacks, and routed via high-volume automated sorters; it will never again get the close and individual attention required to make any intelligent judgments about the accuracy of the return address. Likewise with forged source addresses on internet packets: let them past the first border router, and they are unlikely to be detected.

Now let's look at the situation from the victim's point of view. The first thing you know, the first sign that you may have a problem, is when thousands of compromised systems all over the world commence to flood you with traffic, all at once. The first symptom is likely to be a router crash, or to look a lot like one; traffic simply stops flowing between you and the Internet. When you look more closely you may discover that one or more targeted servers are being overloaded by the small fraction of the traffic that actually gets delivered, but the failures extend much further back.

So you try and find out what's going wrong. After the first few quick checks don't solve the problem, you look at the traffic flowing through your network, and about then you realize you are a victim of a major denial of service attack. So you capture a sample of the packets flying over your net, as many as you can. What does each packet tell you? Well, it will have your address as its destination address, and it will have some random number as a source address. There's no trace of the compromised host that is busy attacking you now. All that's there is a low-level, hardware address of the last router that forwarded the packet; these low-level addresses are used to handle distribution of packets within a LAN. So you can see what router passed the packet to you, but nothing else. Identifying that router may identify the Internet carrier that passed the traffic to you, if you don't have a complex internet of your own, within your own organization. But either way, the next step is to capture another packet on the other side of the forwarding router, and see where that packet came from. Each step of the trace requires starting over, collecting fresh evidence.

Every time the back-trace crosses an administrative boundary, between you and your Internet provider, between them and the next backbone provider on the path, all the way back to the compromised machine, you have to enlist the aid of another team of administrators to collect fresh evidence and carry the trace further back. Now remember that you have to do this in thousands of directions, to each of the thousands of compromised machines that are participating in this attack.

Today there's no possibility of performing more than a few back-traces at most, in as little as a few hours. Even that would require some luck to favor your efforts. So as long as the attacker turns their attack off after at most a few hours, you are unlikely to find more than a few of the thousands of machines used to launch the attack; the remainder will remain available for further attacks. And the compromised machines that are found will contain no evidence that can be used to locate the original attacker; your trace will stop with them.

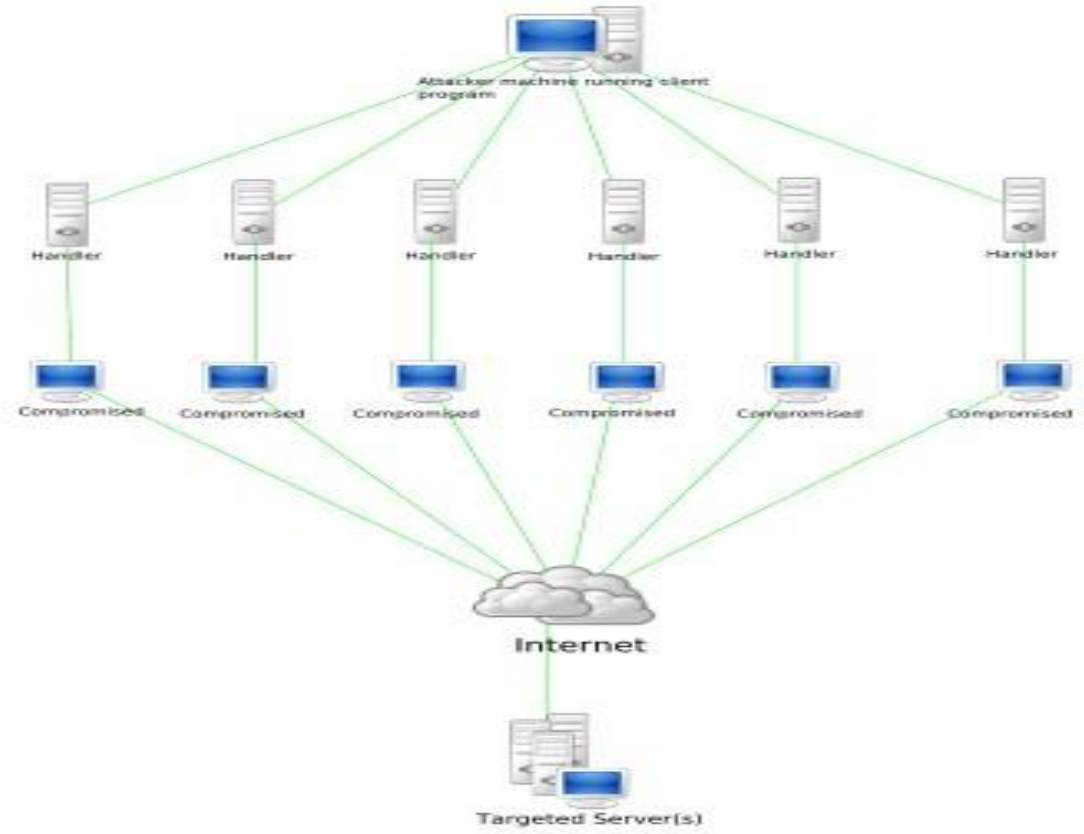


Figure 2: General DDoS scenario

DDoS attacks are targeted at exhausting the victim's resources, such as network bandwidth, computing power, and operating system data structures. To launch a DDoS attack, the attacker(s) first establishes a network of computers that will be used to generate the huge volume of traffic needed to deny services to legitimate users of the victim. To create this attack network, attackers discover vulnerable hosts on the network. Vulnerable hosts are those that are either running no antivirus or out-of-date antivirus software, or those that have not been properly patched. These are exploited by the attackers who use the vulnerability to gain access to these hosts. The next step for the attacker is to install new programs (known as attack tools) on the compromised hosts of the attack network. The hosts running these attack tools are known as zombies, and they can be used to carry out any attack under the control of the attacker. Numerous zombies together form an army or botnet.

2.2 Symptoms and manifestations

The United States Computer Emergency Readiness Team (US-CERT) defines symptoms of denial-of-service attacks to include:

- Unusually slow network performance (opening files or accessing web sites)
- Unavailability of a particular web site
- Inability to access any web site
- Dramatic increase in the number of spam emails received—(this type of DoS attack is considered an e-mail bomb)

Denial-of-service attacks can also lead to problems in the network 'branches' around the actual computer being attacked. For example, the bandwidth of a router between the Internet and a LAN may be consumed by an attack, compromising not only the intended computer, but also the entire network. If the attack is conducted on a sufficiently large scale, entire geographical regions of Internet connectivity can be compromised without the attacker's knowledge or intent by incorrectly configured or flimsy network infrastructure equipment.

2.3 History of Distributed Denial of Service Attacks Types

- **Genesis**

It is always difficult to evaluate the very beginning of such a phenomenon. However some critical attacks have left traces and some techniques are so old that they are part of the History.

- **The Morris Worm**

On November 2 1988 the first logical bomb was launched on the electronic world. That very day Robert Morris Jr. let his proof-of-concept worm invade the Internet. As a result about 15% (about 6.000) of the systems connected to the network were infected and stopped running.

The most surprising part of the story is the fact that this worm was not intended to block systems. Its only purpose was to propagate as far and as fast as possible. However, as it did not include any routine to avoid propagating locally and on an already infected system, it propagated thousands of times on each system, until resources were exhausted, leading to one of the largest breakdowns the Internet has ever known to that point.

- **SYN Floods**

SYN Floods have existed since TCP has existed. They are a direct consequence of TCP specifications. It is therefore possible to say that SYN Floods are part of TCP just as spoofing is part of UDP.

Easy to implement, effective and hard to trace back to the real source, Denial of Service attacks are still appreciated by malicious Internet abusers, managing to launch hundreds of megabits, sometimes more than one gigabit, of SYNs targeted to a single service.

- **The Ping of Death**

In 1995 the first popular and do-it-yourself anomaly-based Denial of Service appeared. Built on the incapacity of most TCP/IP stacks to properly handle ICMP packets longer than 65.635 bytes this attack showed the world that DoS was a reality that anybody could reach from the command line:

Denial of Service Attacks 3

C:> ping <target> -l 65.511

This attack gave rise to a race for more and more anomalies that TCP/IP stacks were not able to manage. Fragmented UDP packets, weird TCP flag combinations, packets with same layer 3 and

4 sources and destinations and the like quickly followed. One can say that most possible anomalies have been explored during this period of time, and tools such as bo(i)nk, pepsi, land and teardrop are still in memories.

- **Smurfing**

Once anomalies were patched, it became difficult to crash a system with a single packet. In addition Internet links at home relied on slow PSTN connections (14.400 to 33.600 bps) and the power of personal computers was far below that of servers. It became necessary to find leveraging factors in order to increase the effect of DoS attacks.

In 1998 a solution was found and the smurfing attack started to impact networks. This ancestor of reflection techniques relied on loosely protected networks (which were very common at the time) to relay their ICMP Echo traffic to the target. This target, flooded with ICMP Echo Replies from every system in the relay network, could reach very high CPU usage trying to handle each received ICMP packet and see if they match any entry in its tables. As a side effect, Internet links of the target IT infrastructure could be filled up with those ICMP replies.

- **DDoS**

The DDoS onslaught on February 7th and 8th 2000 is a case study. This is the typical result of a static defense facing an active and moving attack line. About one year before the launch some proof-of-concept codes of DDoS elements started to become available. In October 1999 a bugtraq post provided links to the source code and even the CERT/CC issued an advisory in December the same year. Behind their firewalls IT infrastructures felt safe. However, in just a few minutes yahoo, amazon, e-bay as well as other world major web sites were down on their knees and remained unreachable for as long as the attack lasted. They had been suddenly hit by thousands and thousands of legitimate connections, either crashing the components of the server or filling up the ISP uplink.

- **Maturity**

The end of the 20th century was a period of exploration. A lot of new techniques emerged and proved their efficiency. But it was time to stop experimentation and move to the industrialization process.

- **CodeRed and Slammer**

CodeRed was nothing new. It used blocks of technologies already known and widely exploited. But CodeRed was the first to put them all together and to target a popular application such as Microsoft IIS.

Then, exploiting the trivial UNICODE vulnerability CodeRed not only affected hundreds of thousands of systems but also loaded an agent whose purpose was to behave like an automated DDoS agent which was supposed to target the white house web site on August 2001. This automation and use of a worm to propagate a DDoS agent were quite new at the time.

The scale and speed of the infection were also unknown, including to the author of the worm. When he thought it would take about 3 months to propagate a single week was enough. Then the worm could be captured and reverse engineered. In the mean time IIS users were urged to apply a patch that had been made available for more than 6 months previously. This was the bright side of CodeRed: having people feel the need for massive and quick patch deployment solutions.

Slammer remains in memories as the second worm that heavily impacted the Internet infrastructure on January 25th 2003. This worm propagated by sending hundreds of thousands of small UDP packets. This lead to network congestion and router collapse. The effect found a huge leveraging factor in the choice of date and time of the attack: 6 AM GMT on a Saturday.

- **Broadband access and WiFi**

In the early years of the Internet, home-users as well as SOHO relied on dynamic and slow Internet connections. Botnets then needed to be quite huge and were pretty unreliable. Cable and ADSL technologies changed the situation. Low prices, high bandwidth and permanent connection have propagated broadband access almost everywhere. A compromised university network is no longer necessary to launch a huge DoS attack.

Moreover the power of personal computers has also increased and made it possible to exploit the full capacity of bandwidth offered by broadband connections. Actual computers are able to generate more than 150.000 packets per second. For small packets, similar to the ones used for SYNFloods, this means that they can create around 80 Mbps of traffic.

Last, WiFi popularity lead to a huge number of unprotected or loosely protected access points that can be used to anonymously launch attacks on behalf of the legitimate owner of the Internet connection.

2.4 Types of attack

(a) ICMP flood

A **smurf attack** is one particular variant of a flooding DoS attack on the public Internet. It relies on misconfigured network devices that allow packets to be sent to all computer hosts on a particular network via the broadcast address of the network, rather than a specific machine. The network then serves as a smurf amplifier. In such an attack, the perpetrators will send large numbers of IP packets with the source address faked to appear to be the address of the victim. The network's bandwidth is quickly used up, preventing legitimate packets from getting through to their destination.

To combat Denial of Service attacks on the Internet, services like the Smurf Amplifier Registry have given network service providers the ability to identify misconfigured networks and to take appropriate action such as filtering.

Smurf attacks are one of the most devastating DoS attacks. In this attack, the attacker sends an ICMP echo request (ping) to a broadcast address. The source address of the echo request is the IP address of the victim (uses the IP address of the victim as the return address). After receiving the echo request, all the machines in the broadcast domain send echo replies (responses) to the victim’s IP address.

Victim will be crash or freeze when receiving larger-sized packet flood from many machines. Smurf attack uses bandwidth consumption to disable a victim system's network resources. It accomplishes the consumption using amplification of the attacker's bandwidth. If the amplifying network has 100 machines, the signal can be amplified 100 times, so the attacker with relatively low bandwidth (such as the 56K modem) can flood and disable a victim system with much higher bandwidth (such as the T1 connection).

The Fraggle (UDP Packet Magnification) attack is the cousin of Smurf attack. Fraggle attack uses UDP echo packets in the same fashion as the ICMP echo packets in Smurf attack. Fraggle usually achieves a smaller amplification factor than Smurf, and UDP echo is a less important service in most network than ICMP echo, so Fraggle is much less popular than Smurf.

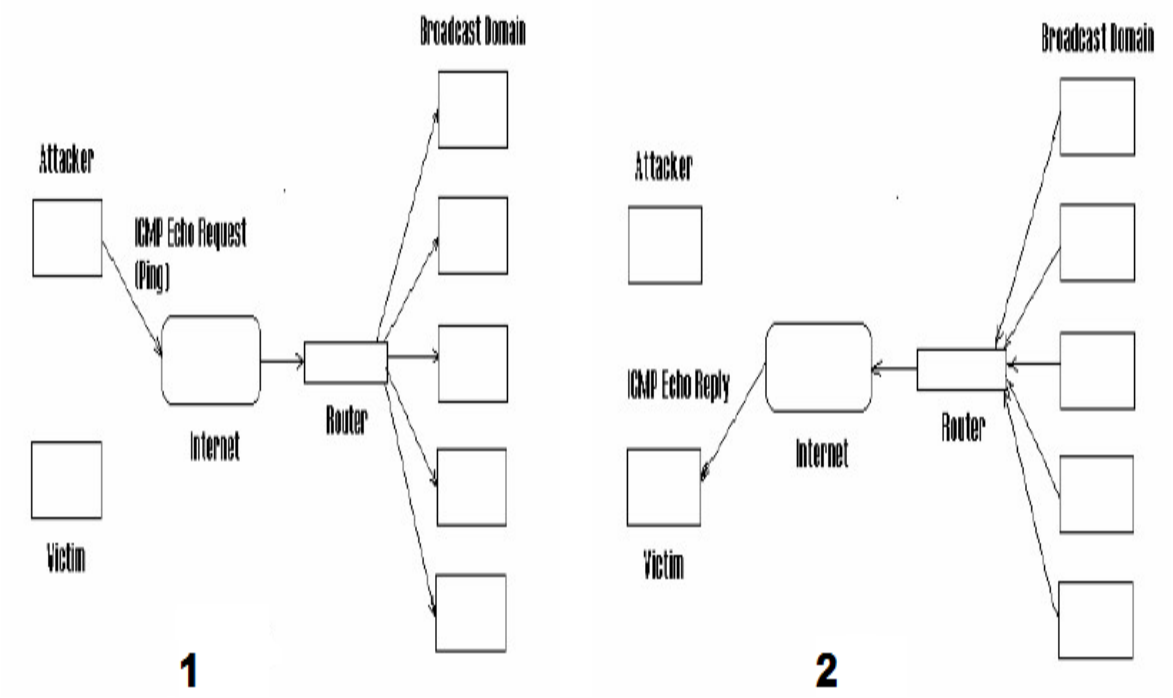


Figure 3: ICMP Flood attack

(b) SYN Flood

The SYN flood attack was considered to be the most devastating DoS attack method before the Smurf was discovered. This method uses resource starvation to achieve the DoS attack. See the figure on below, during a normal TCP handshake, a client sends a SYN request to the server; then the server responds with a ACK/SYN to the client, finally the client sends a final ACK back to the server.

But in a SYN flood attack, the attacker sends multiple SYN requests to the victim server with spoofed source addresses for the return address. The spoofed addresses are nonexistent on network.

The victim server then responds with an ACK/ SYN back to the nonexistent address. Because no address receives this ACK/SYN, the victim server just waits for the ACK from the client. The ACK never arrives, and the victim server eventually times out. If the attacker sends SYN requests often enough, the victim server's available resources for setting up a connection will be consumed waiting for these bogus ACKs. These resources are usually low in number, so relatively few bogus SYN requests can create a DoS event.

A SYN flood occurs when a host sends a flood of TCP/SYN packets, often with a forged sender address. Each of these packets is handled like a connection request, causing the server to spawn a half-open connection, by sending back a TCP/SYN-ACK packet, and waiting for a packet in response from the sender address. However, because the sender address is forged, the response never comes. These half-open connections saturate the number of available connections the server is able to make, keeping it from responding to legitimate requests until after the attack ends.

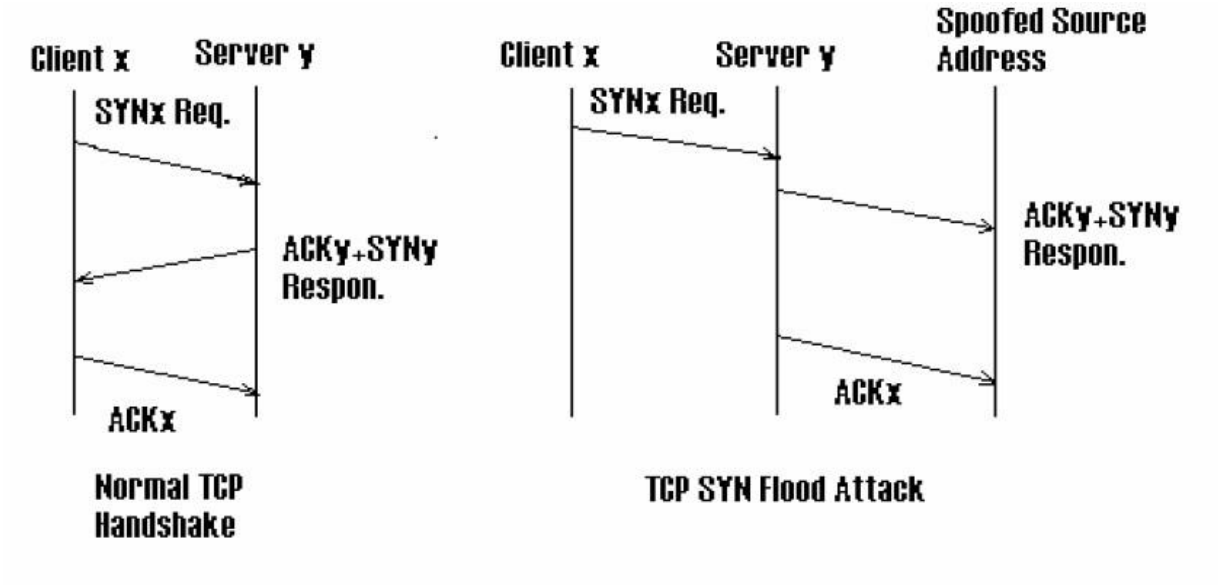


Figure 4: TCP SYN Flood attack

(c) Trinoo

Trinoo is essentially a master/slave (called Masters and Daemons) programs that coordinate with each other to launch a UDP DoS flood against a victim machine.

See the figure, in a typical scenario, the following steps take place as the Trinoo DDoS network is set up:

Step 1 The attacker, using a compromised host, compiles a list of machines that can be compromised. Most of this process is done automatically from the compromised host, because the host stores a mount of information including how to find other hosts to compromise.

Step 2 As soon as the list of machines that can be compromised has been compiled, scripts are run to compromise them and convert them into the Trinoo Masters or Daemons. One Master can control multiple Daemons. The Daemons are the compromised hosts that launch the actual UDP floods against the victim machine.

Step 3 The DDoS attack is launched when the attacker issues a command on the Master hosts. The Masters instruct every Daemon to start a DoS attack against the IP address specified in the command, many DoSs compromise the DDoS attack.

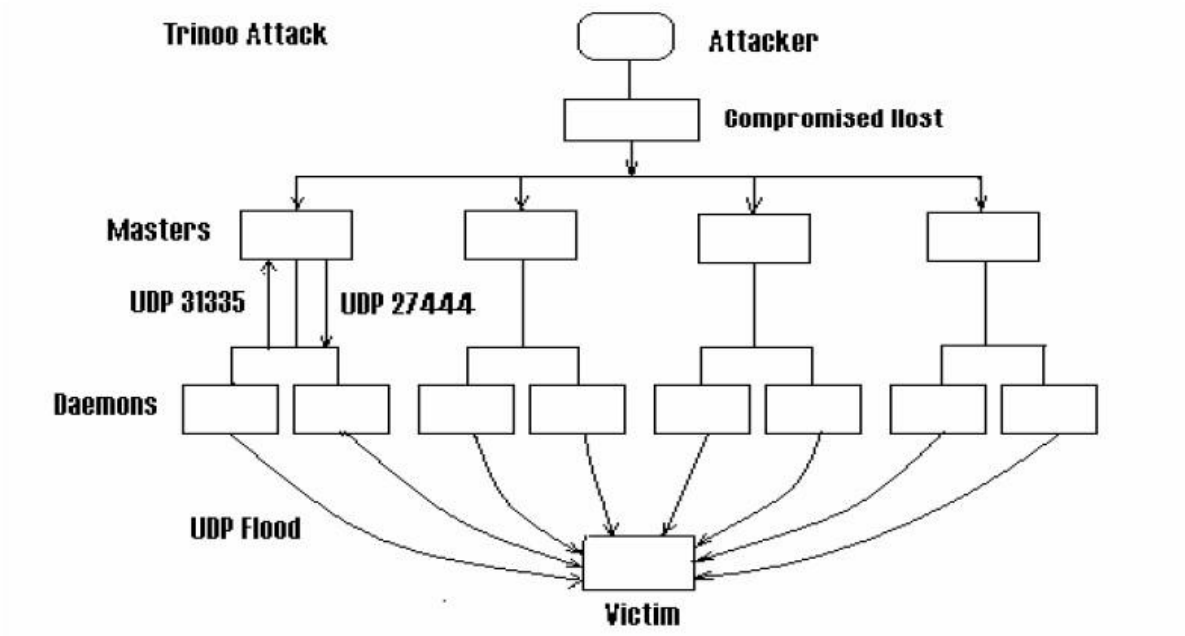


Figure 5: Trinoo attack

(d) TFN/TFN2K

TFN (Tribal Flood Network), like Trinoo, is essentially a master/slave (called Clients and Daemons) programs that coordinate with each other to launch a SYN flood against a victim machine, see the figure. The TFN Daemons, however, are capable of a larger variety of attacks, including ICMP flooding, SYN flooding, and Smurf attacks, so TFN attack is more complicated than the Trinoo attack.

TFN2K introduces some enhancements to the original TFN tool. TFN2K attacks are launched using spoofed IP addresses, making detecting the source of the attacks more difficult. TFN2K attacks are not just simple floods like those in TFN. They also include attacks exploiting the operating system's vulnerabilities to malformed or invalid packets, which can cause the victim machines to crash.

The TFN2K attackers no longer need to execute commands by logging into the Client machine, they can execute these commands remotely. The communication between the Clients and the Daemons is no longer limited to simply ICMP echo replies, it can take place over a larger variety of mediums, such as TCP and UDP. So TFN2K attacks are more dangerous and also more difficult to detect.

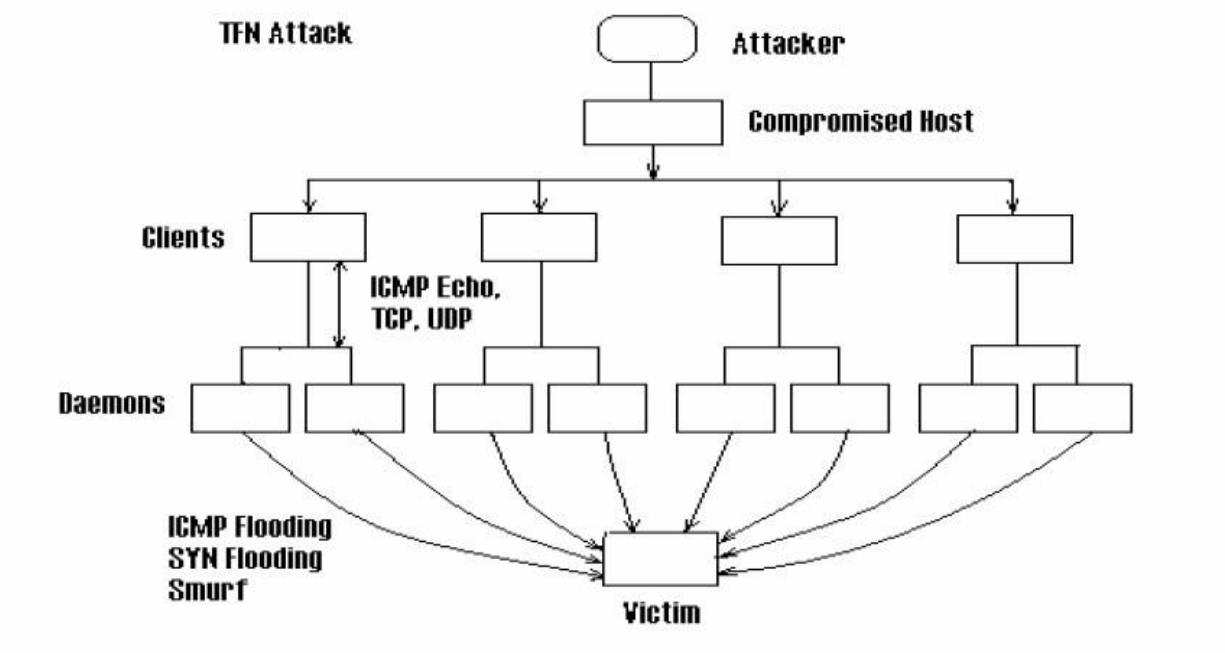


Figure 6: TFN attack

(e) Stacheldraht

Stacheldraht code is very similar to the Trinoo and TFN, but Stacheldraht allows the communication between the attacker and the Masters (called Handlers, see the figure) to be encrypted; the Agents can upgrade their code automatically; can launch different types of attacks such as ICMP floods, UDP floods and SYN floods.

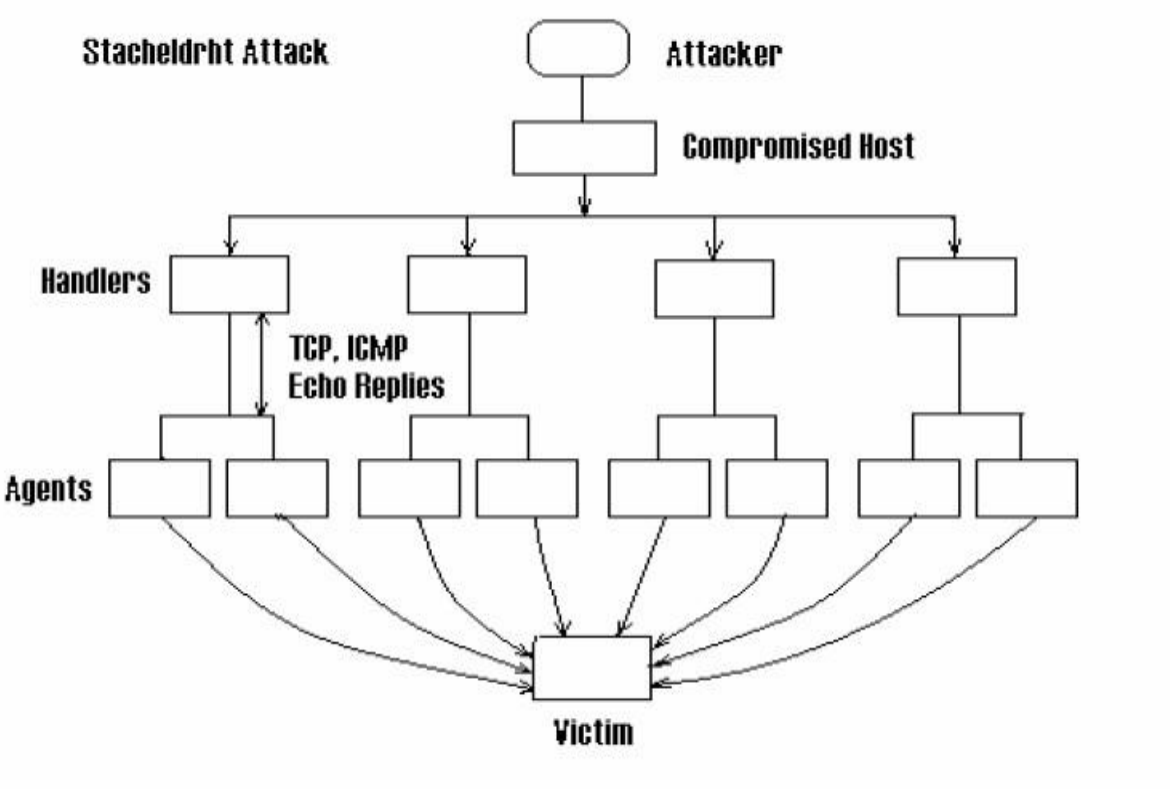


Figure 7: Stacheldraht attack

(f) UDP Flood

UDP is also naturally bound to be a vector of Denial of Service attacks. As it is specified, a server receiving a UDP packet on a closed port sends back an ICMP Port Unreachable packet to the source. The data part of the ICMP packet is filled with at least the first 64 bytes of the original UDP packet. As no limit or quota is specified as a standard, it is then possible to send huge amount of packets on closed ports. At very high load, operations necessary to generate ICMP error packets consume a lot of CPU, eventually leading to CPU resource exhaustion.

Generating such attacks is once again possible from a simple command line.

hping3 --rand-source -–udp <target IP> --flood

Once again spoofing can be used so that ICMP packets don't lower the bandwidth of the attacker.

(g) Teardrop attacks

A Teardrop attack involves sending mangled IP fragments with overlapping, over-sized payloads to the target machine. This can crash various operating systems due to a bug in their TCP/IP fragmentation re-assembly code.[6] Windows 3.1x, Windows 95 and Windows NT operating systems, as well as versions of Linux prior to versions 2.0.32 and 2.1.63 are vulnerable to this attack.

Around September 2009, a vulnerability in Windows Vista was referred to as a "teardrop attack", but the attack targeted SMB2 which is a higher layer than the TCP packets that teardrop us.

(h) Low-rate Denial-of-Service attacks

Recently a new kind of DoS attack, Low-rate Denial-of-Service (LDoS) attack, has been proposed that exploits TCP's retransmission timeout mechanism to reduce TCP throughput without being detected. Compared to traditional flooding based Denial-of-Service attacks, the low-rate DoS attack does not employ a "sledge-hammer" approach of high-rate transmission of packets, and consequently eludes detection. These kinds of attacks are also called shrew attacks, Pulsing DoS (PDoS) attacks, and Reduction of Quality (RoQ) attacks. Recent Publications in low-rate Denial-of-Service (DoS) attacks

The Low-rate DoS (LDoS) attack exploits TCP's slow-time-scale dynamics of retransmission time-out (RTO) mechanisms to reduce TCP throughput. Basically, an attacker can cause a TCP flow to repeatedly enter a RTO state by sending high-rate, but short-duration bursts, and repeating periodically at slower RTO time-scales. The TCP throughput at the attacked node will be significantly reduced while the attacker will have low average rate making it difficult to be detected.

(i) Peer-to-peer attacks

Attackers have found a way to exploit a number of bugs in peer-to-peer servers to initiate DDoS attacks. The most aggressive of these peer-to-peer-DDoS attacks exploits DC++. Peer-to-peer attacks are different from regular botnet-based attacks. With peer-to-peer there is no botnet and the attacker does not have to communicate with the clients it subverts. Instead, the attacker acts as a "puppet master," instructing clients of large peer-to-peer file sharing hubs to disconnect from their peer-to-peer network and to connect to the victim's website instead. As a result, several

thousand computers may aggressively try to connect to a target website. While a typical web server can handle a few hundred connections per second before performance begins to degrade, most web servers fail almost instantly under five or six thousand connections per second. With a moderately large peer-to-peer attack, a site could potentially be hit with up to 750,000 connections in short order. The targeted web server will be plugged up by the incoming connections.

While peer-to-peer attacks are easy to identify with signatures, the large number of IP addresses that need to be blocked (often over 250,000 during the course of a large-scale attack) means that this type of attack can overwhelm mitigation defenses. Even if a mitigation device can keep blocking IP addresses, there are other problems to consider. For instance, there is a brief moment where the connection is opened on the server side before the signature itself comes through. Only once the connection is opened to the server can the identifying signature be sent and detected, and the connection torn down. Even tearing down connections takes server resources and can harm the server. This method of attack can be prevented by specifying in the peer-to-peer protocol which ports are allowed or not. If port 80 is not allowed, the possibilities for attack on websites can be very limited.

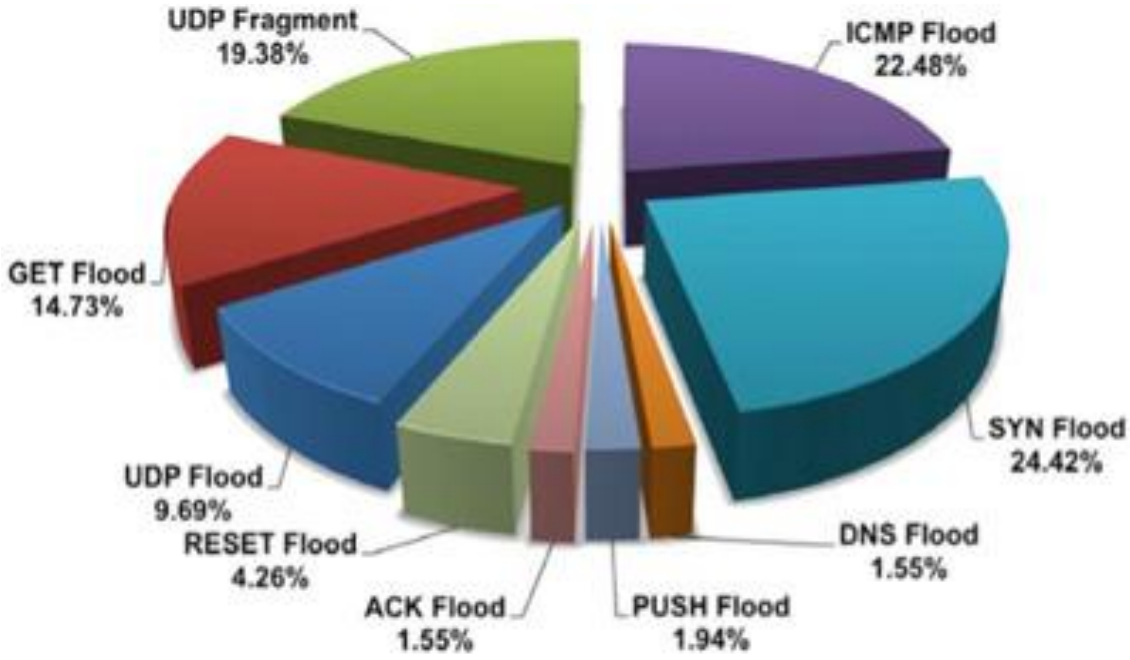


Figure 8: Comparison of attacks

- Network layer (Layer 3) attacks were the most common, making up 83% of total attacks with application layer attacks (Layer 7) accounting for the remaining 17%.
- Average attack duration was 1.4 days and the average speed of traffic mitigated was 1.5 Gbps.
- The highest volume of attacks occurred during the period of August 19-25 and August was the month with the highest number of attacks overall.
- The top three countries from which attacks originated were China, India, and Turkey with China-based IP addresses accounting for 55% of attacks.
- Online gambling was the most heavily targeted industry with an average traffic speed of 1.3 Gbps and average attack duration of 1.2 days.

2.5 Denial of Service Today

Techniques

The time for innovation in Denial of Service techniques is over and we are far from the discoveries of the last decade. However, broadband access, automation and increased power of today's home computers don't make any research necessary. This becomes particularly obvious when we find that old attacks such as land, which sends UDP packet with similar source and target IP addresses and ports, and disappeared in late 90's are back. The only improvement provided is the possibility to launch parallel tasks thus increasing the power of the attack in a way that was impossible to a simple 486 processor.

Another interesting point to take into consideration is the fact that IP stacks don't seem to have been properly patched. Computers don't crash with a single packet anymore; however, CPU operations remain high in order to handle such packets. As packet generation was limited at the time the patches went out, it was not that easy to realize. Maybe technology improved too fast. Whatever the reason, those old and obsolete attacks are now back and terribly efficient.

2.6 Use of Denial of Service

Denial of Service attacks were first used to "have fun", get some kind of revenge from system operators or make complex attacks possible, such as blind spoofing on r services. IRC servers were also often targeted after one got insulted on a channel. At this time networks and Internet uses were "confidential", and those attacks had very limited impact.

With time and as the Internet gets more and more used as a communication channel, hacking becomes more and more popular. Geopolitical situations, wars, religious concerns, ecology, any motive is then good to launch attacks on companies, political organization or even national IT infrastructures.

A more recent use of Denial of Service is linked to online gaming. Many servers have been victims of such attacks, generated by unhappy gamers who lost lives or their favorite weapon during game.

But the very use of Denial of Service today is definitely extortion. More and more enterprises rely on their IT infrastructure. Mail, critical data and even phone are handled by the network. Very few companies can survive without their main communication channel. Furthermore the Internet is also a production tool. Search engines and gambling web sites, as an example rely entirely on their connectivity to the network.

2.7 Incidents

- The first major attack involving DNS servers as reflectors occurred in January 2001. The target was Register.com. This attack, which forged requests for the MX records of AOL.com (to amplify the attack) lasted about a week before it could be traced back to all attacking hosts and shut off. It used a list of tens of thousands of DNS records that were a year old at the time of the attack.
- In February 2001, the Irish Government's Department of Finance server was hit by a denial of service attack carried out as part of a student campaign from NUI Maynooth. The Department officially complained to the University authorities and a number of students were disciplined.
- In July 2002, the Honeynet Project Reverse Challenge was issued. The binary that was analyzed turned out to be yet another DDoS agent, which implemented several DNS related attacks, including an optimized form of a reflection attack.
- On two occasions to date, attackers have performed DNS Backbone DDoS Attacks on the DNS root servers. Since these machines are intended to provide service to all Internet users, these two denial of service attacks might be classified as attempts to take down the entire Internet, though it is unclear what the attackers' true motivations were. The first occurred in October 2002 and disrupted service at 9 of the 13 root servers. The second occurred in February 2007 and caused disruptions at two of the root servers.

- In February 2007, more than 10,000 online game servers in games such as Return to Castle Wolfenstein, Halo, Counter-Strike and many others were attacked by the hacker group RUS. The DDoS attack was made from more than a thousand computer units located in the republics of the former Soviet Union, mostly from Russia, Uzbekistan and Belarus. Minor attacks are still continuing to be made today.
- In the weeks leading up to the five-day 2008 South Ossetia war, a DDoS attack directed at Georgian government sites containing the message: "win+love+in+Rusia" effectively overloaded and shut down multiple Georgian servers. Websites targeted included the Web site of the Georgian president, Mikhail Saakashvili, rendered inoperable for 24 hours, and the National Bank of Georgia. While heavy suspicion was placed on Russia for orchestrating the attack through a proxy, the St. Petersburg-based criminal gang known as the Russian Business Network, or R.B.N, the Russian government denied the allegations, stating that it was possible that individuals in Russia or elsewhere had taken it upon themselves to start the attacks.
- During the 2009 Iranian election protests, foreign activists seeking to help the opposition engaged in DDoS attacks against Iran's government. The official website of the Iranian government (ahmedinejad.ir) was rendered inaccessible on several occasions.Critics claimed that the DDoS attacks also cut off internet access for protesters inside Iran; activists countered that, while this may have been true, the attacks still hindered President Mahmoud Ahmadinejad's government enough to aid the opposition.
- On June 25, 2009, the day Michael Jackson died, the spike in searches related to Michael Jackson was so big that Google News initially mistook it for an automated attack. As a result, for about 25 minutes, when some people searched Google News they saw a "We're sorry" page before finding the articles they were looking for.
- June 2009 the P2P site The Pirate Bay was rendered inaccessible due to a DDoS attack. This was most likely provoked by the recent sellout to Global Gaming Factory X AB, which was seen as a "take the money and run" solution to the website's legal issues. In the end, due to the buyers' financial troubles, the site was not sold.
- Multiple waves of July 2009 cyber attacks targeted a number of major websites in South Korea and the United States. The attacker used botnet and file update through internet is known to assist its spread. As it turns out, a computer trojan was coded to scan for

existing MyDoom bots. MyDoom was a worm in 2004, and in July around 20,000-50,000 were present. MyDoom has a backdoor, which the DDoS bot could exploit. Since then, the DDoS bot removed itself, and completely formatted the hard drives. Most of the bots originated from China, and North Korea.

- On August 6, 2009 several social networking sites, including Twitter, Facebook, Livejournal, and Google blogging pages were hit by DDoS attacks, apparently aimed at Georgian blogger "Cyxymu". Although Google came through with only minor set-backs, these attacks left Twitter crippled for hours and Facebook did eventually restore service although some users still experienced trouble. Twitter's Site latency has continued to improve, however some web requests continue to fail.
- In July and August 2010, the Irish Central Applications Office server was hit by a denial of service attack on four separate occasions, causing difficulties for thousands of Second Level students who are required to use the CAO to apply for University and College places. The attack is currently subject to a Garda investigation.
- On November 28, 2010, whistle blower site wikileaks.org experienced a DDoS attack. This was presumably related to the pending release of many thousands of secret diplomatic cables.

On December 8, 2010, a group calling themselves "Anonymous" launched orchestrated DDoS attacks on organizations such as Mastercard.com, PayPal, Visa.com and PostFinance; as part of the ongoing "Operation Payback" campaign, which originally targeted anti-piracy organizations, in support of the Whistleblowing site Wikileaks and its founder, Julian Assange. The attack brought down the Mastercard, PostFinance, and Visa websites successfully by deploying 3 versions of the Denial of Service tool. PostFinance, the bank that had frozen Julian Assange's account was brought down for more than 16 hours due to the attacks.

2.8 Prevention and response

FIREWALLS

Firewalls have simple rules such as to allow or deny protocols, ports or IP addresses. Some DoS attacks are too complex for today's firewalls, e.g. if there is an attack on port 80 (web service), firewalls cannot prevent that attack because they cannot distinguish good traffic from DoS attack traffic. Additionally, firewalls are too deep in the network hierarchy. Routers may be affected even before the firewall gets the traffic. Nonetheless, firewalls can effectively prevent users from launching simple flooding type attacks from machines behind the firewall.

Some stateful firewalls, like OpenBSD's pf(4) packet filter, can act as a proxy for connections: the handshake is validated (with the client) instead of simply forwarding the packet to the destination. It is available for other BSDs as well. In that context, it is called "synproxy".

Switches

Most switches have some rate-limiting and ACL capability. Some switches provide automatic and/or system-wide rate limiting, traffic shaping, delayed binding (TCP splicing), deep packet inspection and Bogon filtering (bogus IP filtering) to detect and remediate denial of service attacks through automatic rate filtering and WAN Link failover and balancing.[citation needed]

These schemes will work as long as the DoS attacks are something that can be prevented by using them. For example SYN flood can be prevented using delayed binding or TCP splicing. Similarly content based DoS can be prevented using deep packet inspection. Attacks originating from dark addresses or going to dark addresses can be prevented using Bogon filtering. Automatic rate filtering can work as long as you have set rate-thresholds correctly and granularly. Wan-link failover will work as long as both links have DoS/DDoS prevention mechanism.

Router Similar to switches, routers have some rate-limiting and ACL capability. They, too, are manually set. Most routers can be easily overwhelmed under DoS attack. If you add rules to take

flow statistics out of the router during the DoS attacks, they further slow down and complicate the matter. Cisco IOS has features that prevent flooding.

Application front end hardware

Application front end hardware is intelligent hardware placed on the network before traffic reaches the servers. It can be used on networks in conjunction with routers and switches. Application front end hardware analyzes data packets as they enter the system, and then identifies them as priority, regular, or dangerous. There are more than 25 bandwidth management vendors. Hardware acceleration is key to bandwidth management.

- ***IPS based prevention*** Intrusion-prevention systems (IPS) are effective if the attacks have signatures associated with them. However, the trend among the attacks is to have legitimate content but bad intent. Intrusion-prevention systems which work on content recognition cannot block behavior-based DoS attacks.

An ASIC based IPS can detect and block denial of service attacks because they have the processing power and the granularity to analyze the attacks and act like a circuit breaker in an automated way.

A rate-based IPS (RBIPS) must analyze traffic granularly and continuously monitor the traffic pattern and determine if there is traffic anomaly. It must let the legitimate traffic flow while blocking the DoS attack traffic.

- ***DDS based defense*** More focused on the problem than IPS, a DDoS Defense System (DDS) is able to block connection-based DDoS attacks and those with legitimate content but bad intent. A DDS can also address both protocol attacks (such as Teardrop and Ping of death) and rate-based attacks (such as ICMP floods and SYN floods).

Like IPS, a purpose-built system, such as the well-known Top Layer IPS products, can detect and block denial of service attacks at much nearer line speed than a software based system.

Prevention via proactive testing

Test platforms such as Mu Dynamics' Service Analyzer are available to perform simulated denial-of-service attacks that can be used to evaluate defensive mechanisms such IPS, RBIPS, as well as the popular denial-of-service mitigation products from Arbor Networks. An example of proactive testing of denial-of-service throttling capabilities in a switch was performed in 2008: The Juniper EX 4200 switch with integrated denial-of-service throttling was tested by Network Test and the resulting review was published in Network World.

Blackholing and sinkholing

With blackholing, all the traffic to the attacked DNS or IP address is sent to a "black hole" (null interface, non-existent server, ...). To be more efficient and avoid affecting your network connectivity, it can be managed by the ISP.

Sinkholing routes to a valid IP address which analyzes traffic and rejects bad ones. Sinkholing is not efficient for most severe attacks.

2.9 Literature Survey

- **Yang Xiang et al.[9]** propose using two new information metrics such as the generalized entropy metric and the information distance metric to detect low-rate DDoS attacks by measuring the difference between legitimate traffic and attack traffic. The proposed IP trace back algorithm can find all attacks as well as attackers from their own local area networks (LANs) and discard attack traffic.

- **Chonka et al.[10]** recreate some of the current attacks that attackers may initiate as HTTP and XML. Then offer a solution to traceback through Cloud TraceBack (CTB) to find the source of these attacks, and introduce the use of a back propagation neutral network, called Cloud Protector, which was trained to detect and filter such attack traffic.

Later on , they present a new system called ENDER. ENDER addresses the problem of HXDoS attacks against Cloud Web Services.

- **Muhammad Zakarya [11]** proposed a DDoS detection and prevention mechanism that has the attractiveness of being easy to adapt and more trustworthy than existing counterparts. He introduced entropy based detection mechanism for DDoS attack detection.

- **Priyanka Negi et al. [12]** propose a modification to the confidence Based Filtering method (CBF) which is investigated for cloud computing environment based on correlation pattern to mitigate DDoS attacks on Cloud. The modification introduces nominal additional bandwidth and tries to increase the processing speed of the victim initiated server.
- **Savage et al.** [13] first introduced the probability-based packet marking method, node appending, which appends each node's address to the end of the packet as it travels from the attack source to the victim. The authors proposed the node sampling algorithm, which records the router address to the packet with probability, p, on the routers of the attack path. Then, the probability of a packet marked by a router d that hops away from the victim is $p(1-p)^{d-1}$. Based on the number of marked packets, we can reconstruct the attack path. However, it requires large number of packets to improve the accuracy of the attack path reconstruction.
- **Shui Yu et al.** [14] proposed method needs no marking on packets and therefore avoids the inherent shortcomings of packet marking mechanisms. It employs the features that are out of the control of hackers to conduct IP traceback. Once a DDoS attack has been identified by the victim via detection algorithms, the victim then initiates the pushback tracing procedure. The traceback algorithm first identifies its upstream routers where the attack flows came from, and then submits the traceback requests to the related upstream routers .This procedure continues until the most far away zombies are identified**.**

CHAPTER 3

System Implementation and Results

3.1 Proposed Work

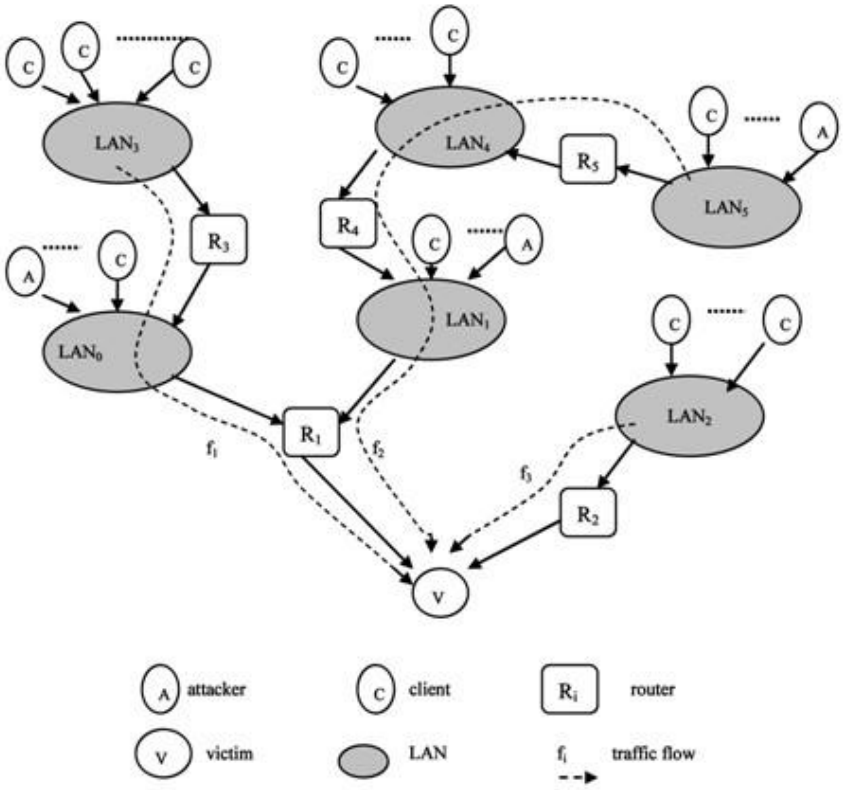


Figure 9: Sample DDoS attack network

In a DDoS attack scenario, as shown in above figure, the flows with destination as the victim include legitimate flows, such as f3, and a combination of attack flows and legitimate flows, such as f1 and f2. Compared with non attack cases, the volumes of some flows increase significantly in a very short time period in DDoS attack cases. Observers at routers R1, R4, R5, and V will notice the dramatic changes; however, the routers who are not in the attack paths, such as R2 and R3, will not be able to sense the variations. Therefore, once the victim realizes an ongoing attack, it can pushback to the LANs, which caused the changes based on the information of flow entropy variations, and therefore, we can identify the locations of attackers.

3.2 Proposed System

In this research, I have taken a novel mechanism for IP traceback using information theoretical parameters, and there is no packet marking in the proposed strategy. The packets that are passing through a router are categorized into flows, which are defined by the upstream router where a packet came from, and the destination address of the packet. During non-attack periods, routers are required to observe and record entropy variations of local flows. I have used flow entropy variation or entropy variation interchangeably. Once a DDoS attack has been identified, the victim initiates the following pushback process to identify the locations of zombies: the victim first identifies which of its upstream routers are in the attack tree based on the flow entropy variations it has accumulated, and then submits requests to the related immediate upstream routers. The upstream routers identify where the attack flows came from based on their local entropy variations that they have monitored. Once the immediate upstream routers have identified the attack flows, they will forward the requests to their immediate upstream routers, respectively, to identify the attacker sources further; this procedure is repeated in a parallel and distributed fashion until it reaches the attack source(s) or the discrimination limit between attack flows and legitimate flows is satisfied.

A novel mechanism for IP trace-back uses information theoretical parameters, and there is no packet marking in the proposed strategy. The packets categorized in such a way that are passing through a router into flows, which are defined by the upstream router where a packet came from, and the destination address of the packet. During non-attack periods, routers are required to observe and record entropy variations of local flows. Flow entropy variation or entropy variation interchangeably are used.

The proposed strategy is fundamentally different from the existing PPM or DPM traceback mechanisms, and it outperforms the available PPM and DPM methods. Because of this essential change, the proposed strategy overcomes the inherited drawbacks of packet marking methods, such as limited scalability, huge demands on storage space, and vulnerability to packet pollutions.

The implementation of the proposed method brings no modifications on current routing software. Both PPM and DPM require update on the existing routing software, which is extremely hard to achieve on the Internet. On the other hand, this method can work independently as an additional

module on routers for monitoring and recording flow information, and communicating with its upstream and downstream routers when the pushback procedure is carried out.

The proposed method will be effective for future packet flooding DDoS attacks because it is independent of traffic patterns. Some previous works [23] depend heavily on traffic patterns to conduct their traceback. For example, they expected that traffic patterns obey Poisson distribution or Normal distribution. However, traffic patterns have no impact on the proposed scheme; therefore, this method can deal with any complicated attack patterns, even legitimate traffic pattern mimicking attacks.

3.3 Analysis of Existing system and proposed system

Existing system

A number of IP traceback approaches have been suggested to identify attackers and there are two major methods for IP traceback, the probabilistic packet marking (PPM) and the deterministic packet marking (DPM).

Both of these strategies require routers to inject marks into individual packets.

The DPM strategy requires all the Internet routers to be updated for packet marking. Moreover, the DPM mechanism poses an extraordinary challenge on storage for packet logging for routers. Further, both PPM and DPM are vulnerable to hacking, which is referred to as packet pollution.

Disadvantages

- PPM strategy can only operate in a local range of the Internet (ISP network), where the defender has the authority to manage. ISP networks are generally quite small, and cannot traceback to the attack sources located out of the ISP network.
- Because of the vulnerability of the original design of the Internet, we may not be able to find the actual hackers at present.
- Possibility of packet pollution attacks.

Proposed System

The proposed method can archive real-time traceback to attackers. Once the short-term flow information is in place at routers, and the victim notices that it is under attack, it will start the traceback procedure. The workload of traceback is distributed, and the overall traceback time mainly depends on the network delays between the victim and the attackers.

Advantages

- Effective and efficient
- Outperforms the available PPM and DPM methods
- The implementation of the proposed method brings no modifications on current routing software.

3.4 Algorithm

\

Local Flow Monitoring Algorithm

Step 1: Initialize local threshold parameter mean c, standard variation δ and time interval T

Step 2: Identify flows f1, f2, f3…fn and set count number for each flows=0 ,$x_1=x_2=\ldots=x_n=0$

Step 3: When time interval for file transfer is over, then calculate the entropy

$$H(F) = -\sum_{i=1}^{n} p_i \log\ p_i$$

Step 4: save x_1, x_2 ,.., x_n and H(F)

Step 5: If there is no randomness or dramatic change in entropy then the flow is monitored as normal, else compare H(F) with δ, then inform the upstream router about the variation

Step 6: goto Step 2

IP Traceback Algorithm

Step 1: Initialize a set A= ϕ ; local threshold parameter c, δ ;

Step 2: Let U={u_i}, i Є I, be the set of upstream routers; D = {d_i}, i Є I be the destination of the packet to send and V be the victim (I is the set of integers)

Step 3: Attack flow f_i= <uj, v>,i=1,2,…,n, u_j Є U and sort the attack flows in the descending order we have flows namely f1, f2, f3…fn

Step 4: For i= 1…n, calculate flow variation if it is greater than δ then append the corresponding upstream router to set A

Step 5: Submit the traceback request to routers in set A and inform the confirmed attacker IP to router in set A.

3.5 System Design

Data Flow diagram

Level: 0

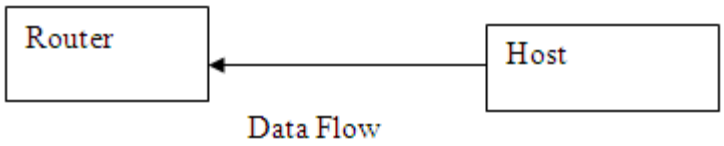


Figure 11: 0 level DFD

In this Level-0 DFD I try to show the data flow between the two main entities i.e host and the router. Host is the entities where data is created and prepared for delievery whereas router is the entitie through which packet is routed to the receiving host.

Level: 1

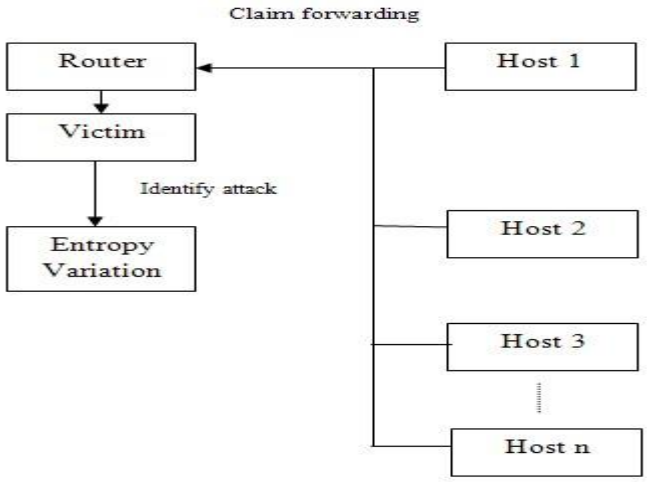


Figure 12: 1 level DFD

In this Level 1 DFD receiving host and vitim after detecting the attack sends the request packet to the router for detecting the source of DDOS attack.After detecting its source router sends the source information to the vitim host.

UML Diagram

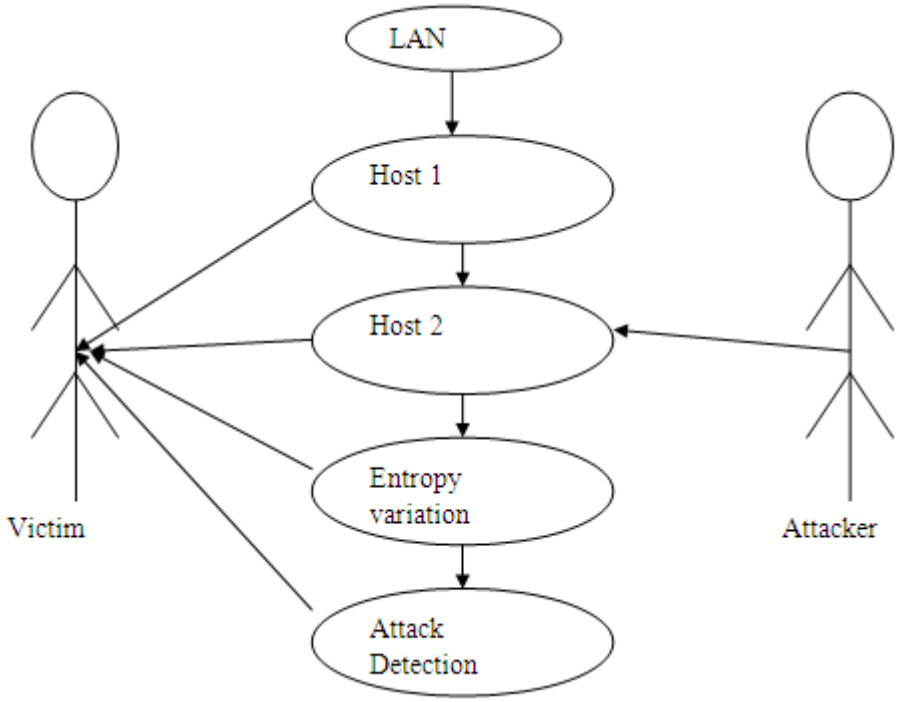


Figure 13: Use case Diagram

Sequence Diagram

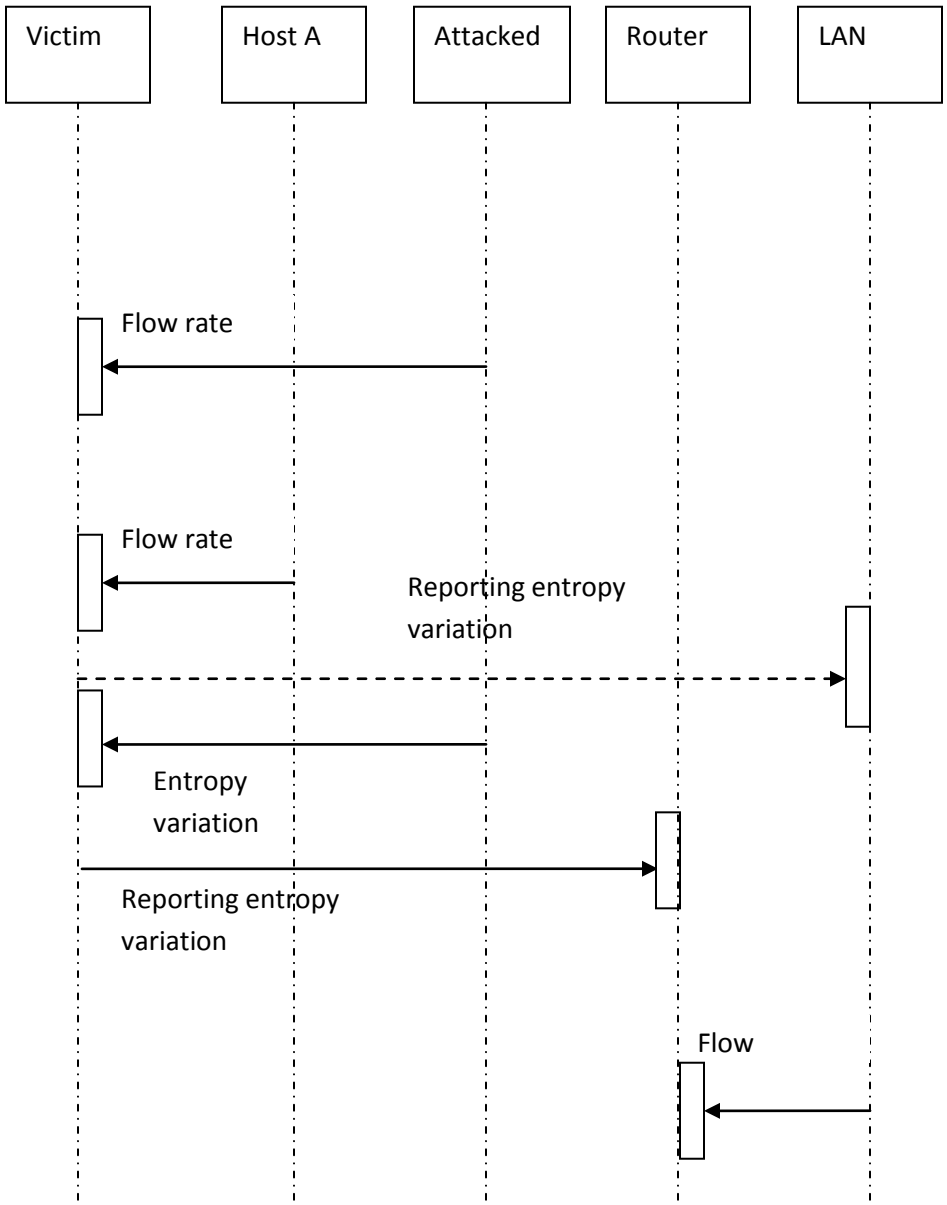


Figure 14: Sequence diagram

Activity Diagram

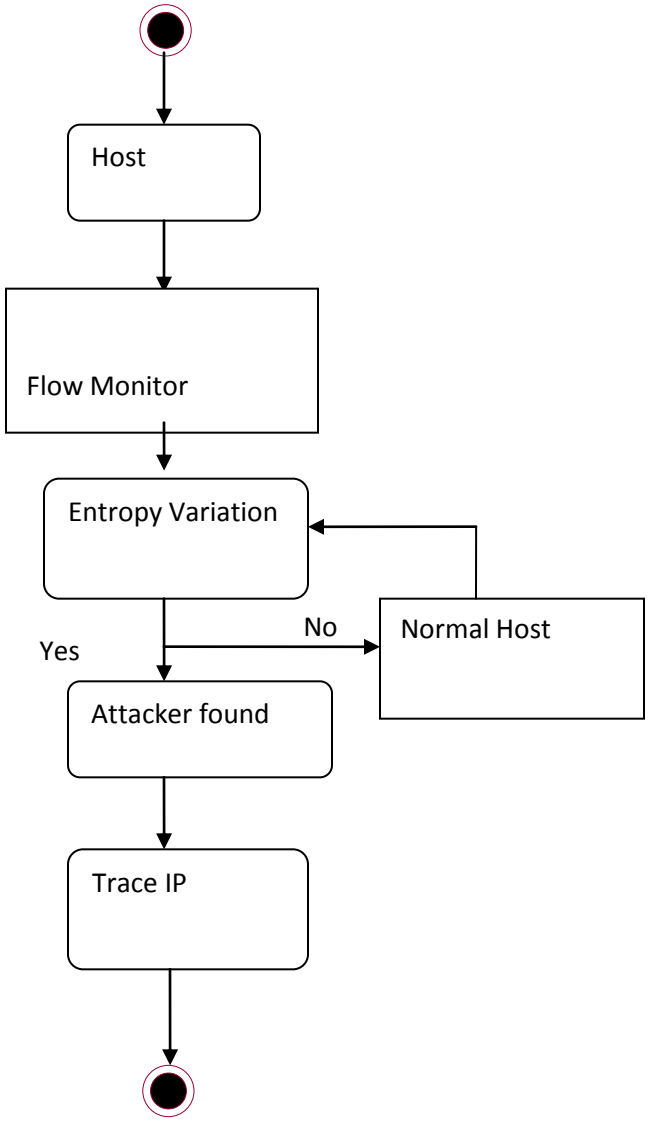


Figure 15: Activity Diagram

Collaboration Diagram:

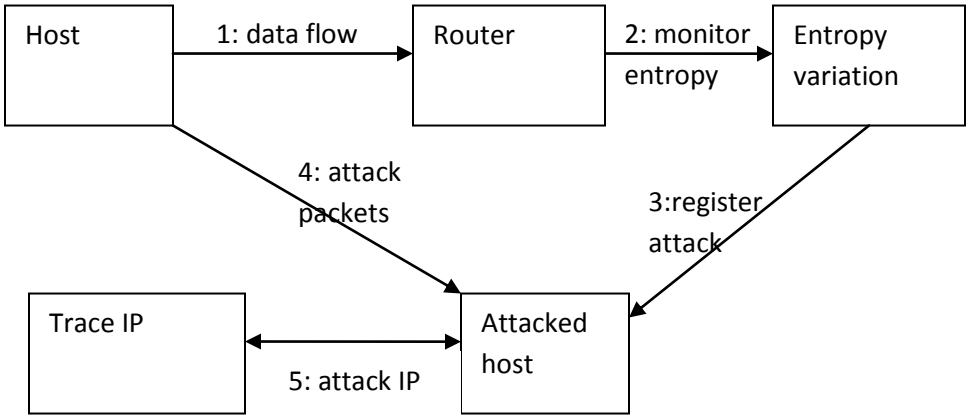


Figure 16: Collaboration diagram

CHAPTER 4

System Implementation and Results

4.1 Implementation Plan

Firstly I have shown the implementation of ICMP Smurf flood attack which I have implemented on linux fedora operating system using gcc compiler. The program takes three command line arguments. The first is the source IP address to use, the second is the destination address, and the third argument is the number of packets to be sent. If you specify 0 packets it will send an infinite amount of packets. This program uses a raw socket to spoof the source IP address and therefore must be run as root.

Then, I try to find out the source of DDOS attack in the network. The network is designed in such a way that it contains router, client, and victim. I use Java swing for GUI design, to enhance look and feel of the GUI, we are using substance-lite.jar. Router, client and victim are design as windows application using swing.

Router is executed first, then client and victim. Socket connection will be established between router and client, victim. Client can send normal data to the destination, in the router; we can monitor the information such as, IP from which the packets are coming and the number of packets.

The attacker is also considered as a client, from the client's GUI, we can send attack packet to the destination. When we pass the data from client to victim, the data will be sent twice instead once. Data redundancy found in the victim and thus the victim identifies the flow and informs the upstream router.

In router, we can able see the attacker IP, number of packet received in victim side. Thus the source of DDoS attack can be found in the router.

4.2 Module Description

4.2.1 Flow monitoring

The packets that are passing through a router are categorized into flows. A flow is defined by a pair the upstream router where the packet came from, and the destination address of the packet. In the LAN each every router is continuously monitored for flow rate from upstream flow. From the flow, the flow rate can be monitored continuously.

4..2.2 Entropy variation

Entropy is an information theoretic concept, which is a measure of randomness. Entropy variation is employed to measure changes of randomness of flows at a router for a given time interval. Entropy variation is only one of the possible metrics. Entropy variation is identified as Standard variation or high-order moments of flows. Entropy variation chosen rather than others because of the low computing workload for entropy variations

4.2.3 IP traceback

IP traceback means the capability of identifying the actual source of any packet sent across the network. Because of the vulnerability of the original design of the Internet, we may not be able to find the actual hackers at present. The victim first identifies which of its upstream routers are in the attack tree based on the flow entropy variations it has accumulated, and then submits requests to the related immediate upstream routers. The upstream routers identify where the attack flows came from based on their local entropy variations that they have monitored. Once the immediate upstream routers have identified the attack flows, they will forward the requests to their immediate upstream routers, respectively, to identify the attacker sources further; this procedure is repeated in a parallel and distributed fashion until it reaches the attack source(s) or the discrimination limit between attack flows and legitimate flows is satisfied.

4.3 Results

4.3.1 ICMP smurf flood attack

Snapshots

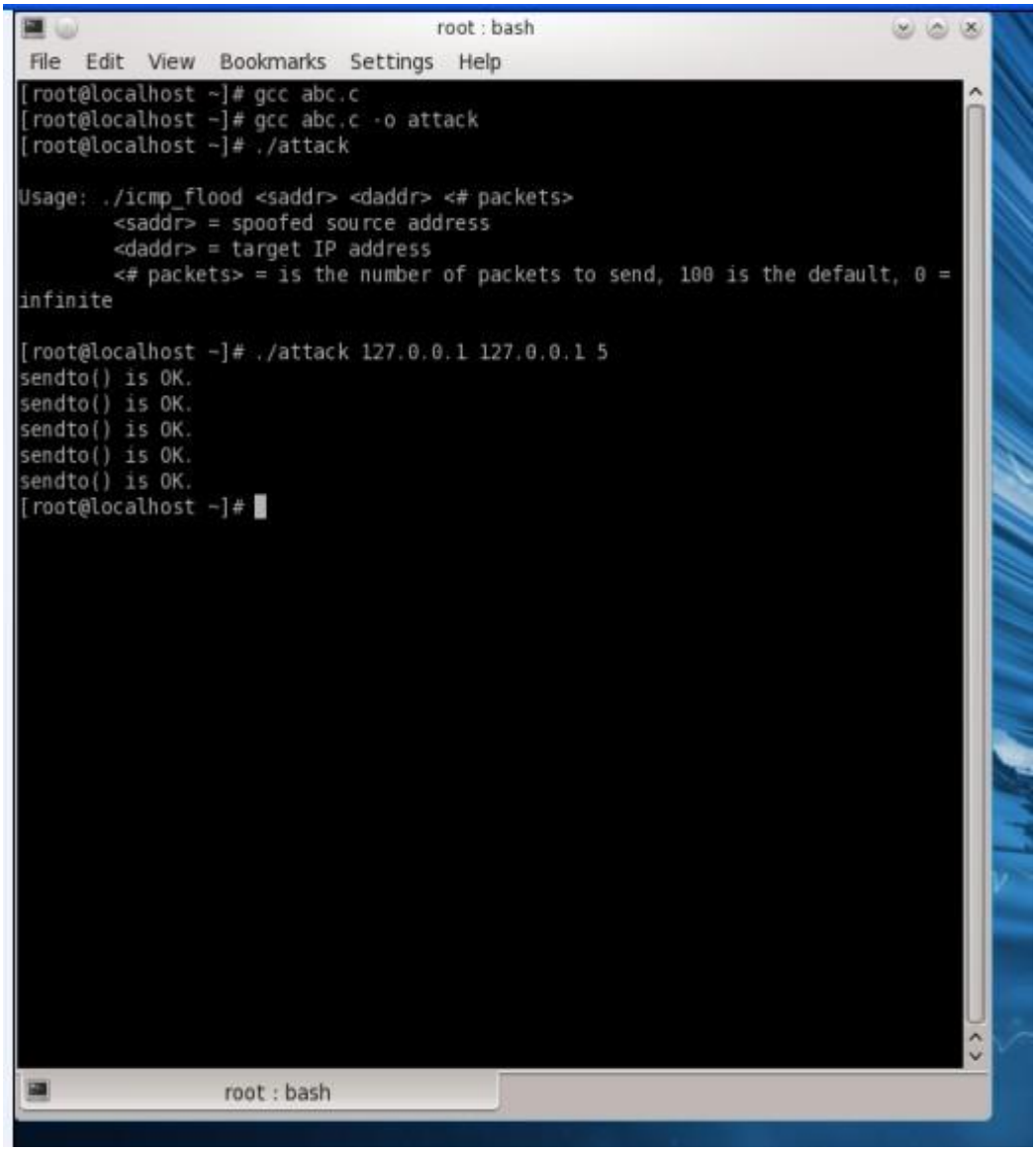


Figure 17: Compilation and run of smurf application

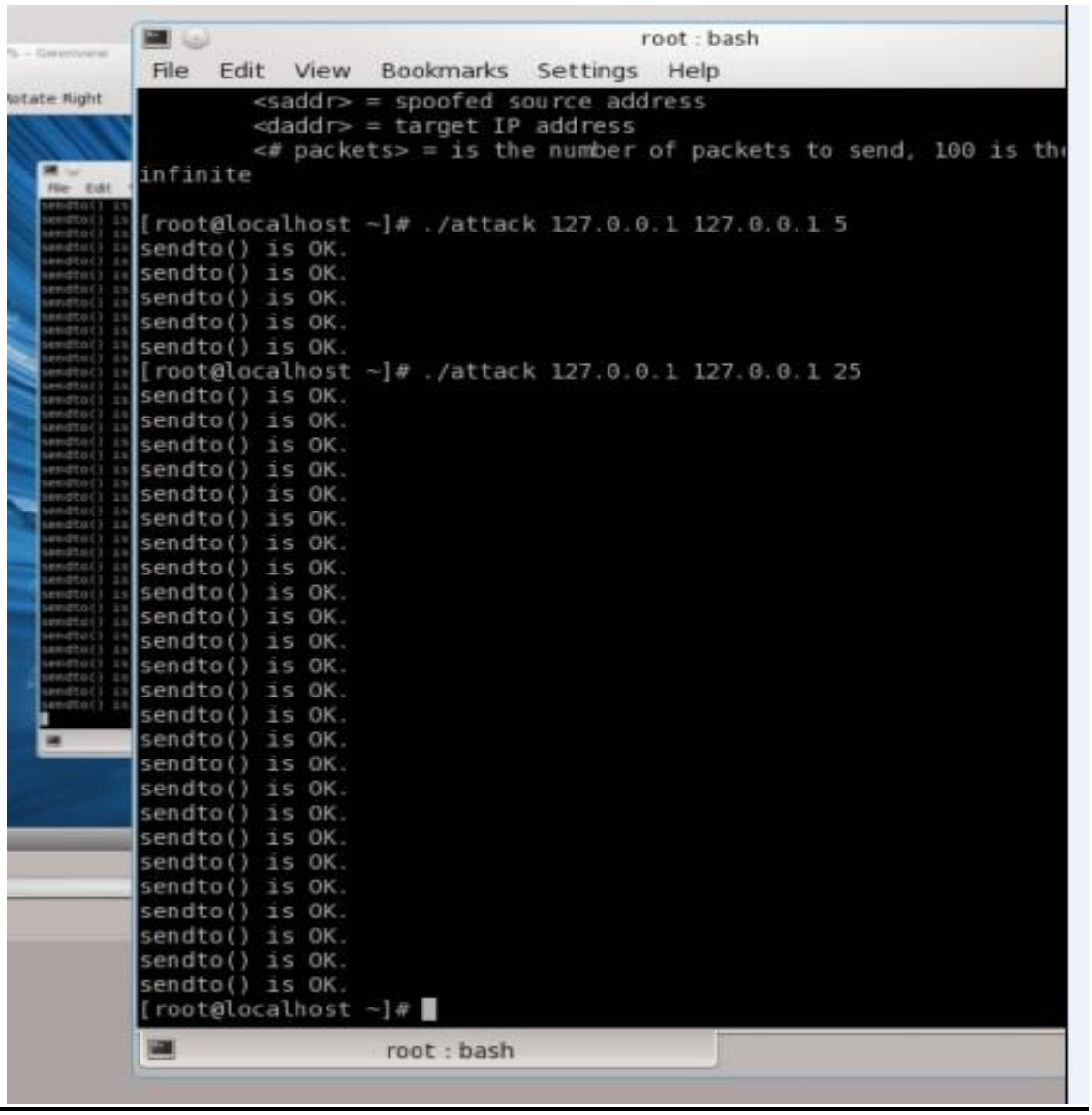


Figure 18: Flood 25

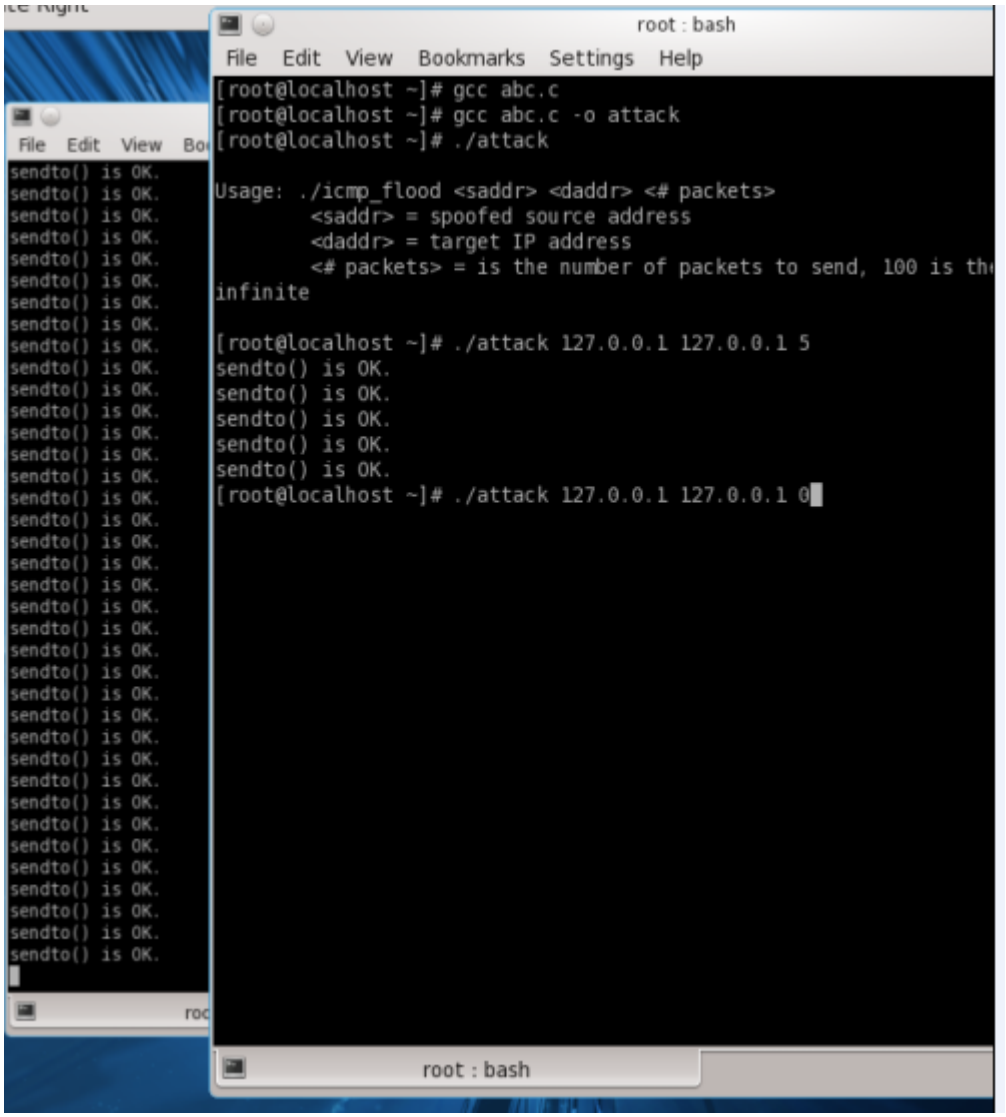


Figure 19: Flood attack by entering zero

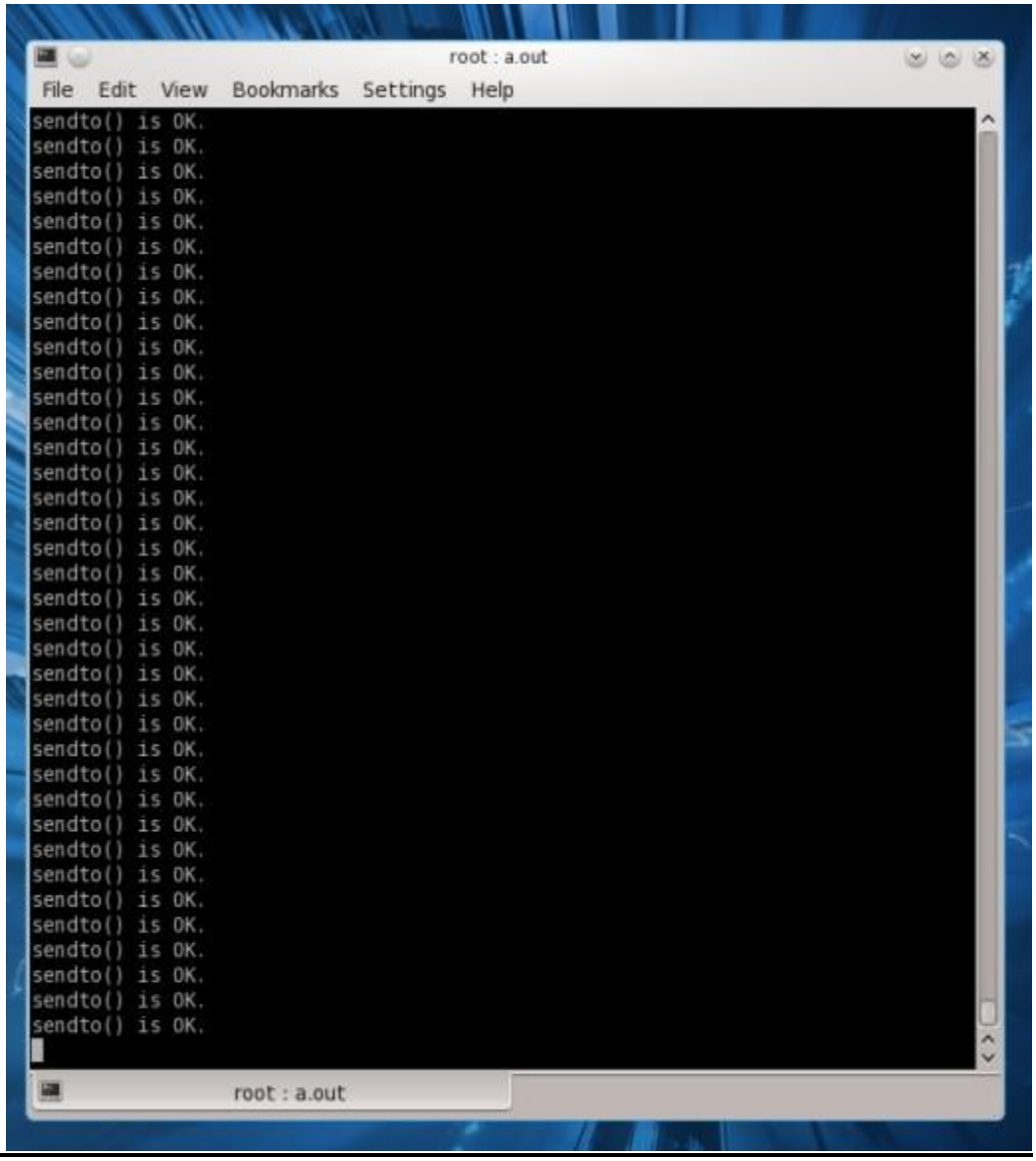


Figure 20 : Attack performed “hanging of application”due to 0

4..3.2 Detection of source of flood attack

Snapshots

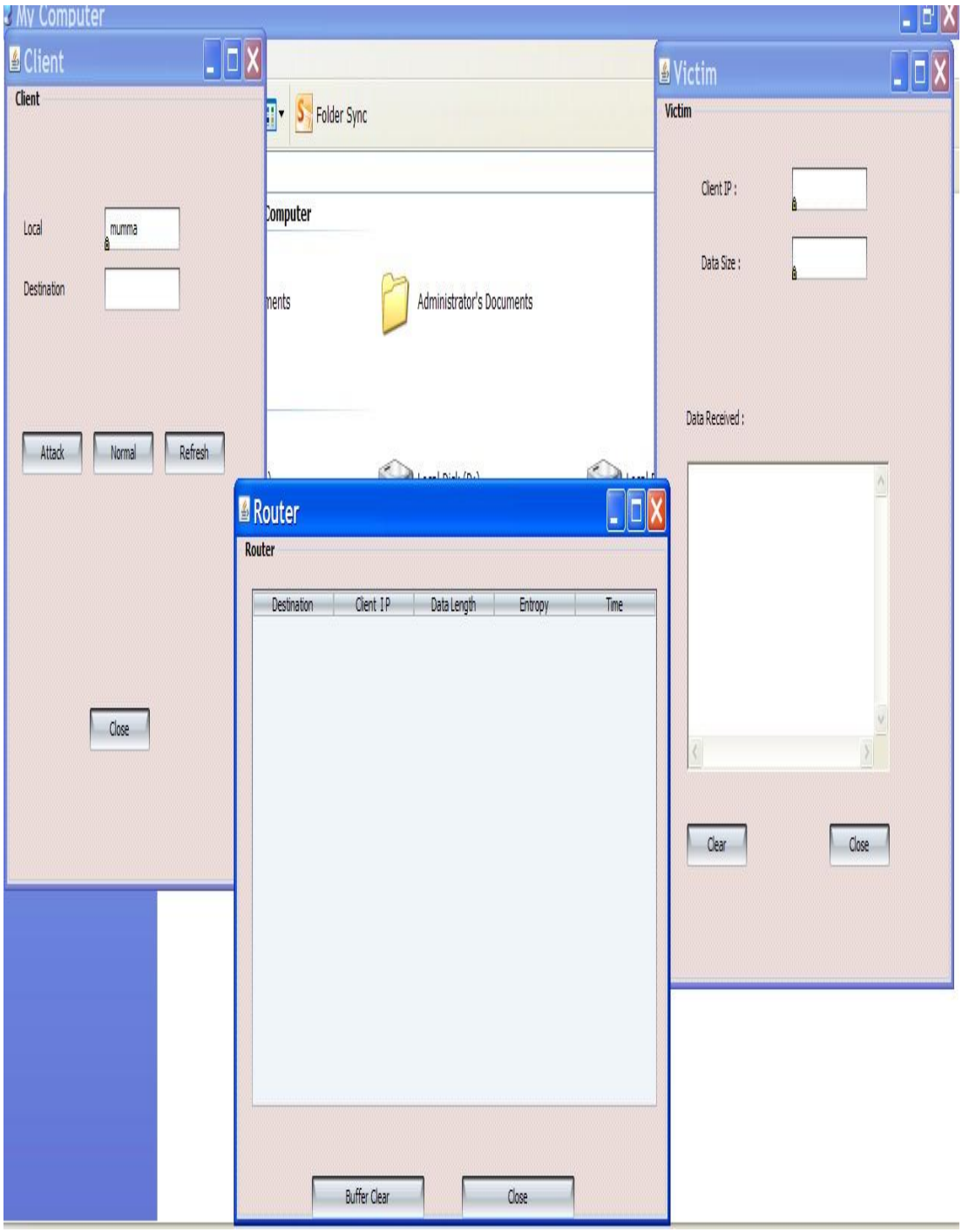


Figure 21 : Initial stages of client, router and victim

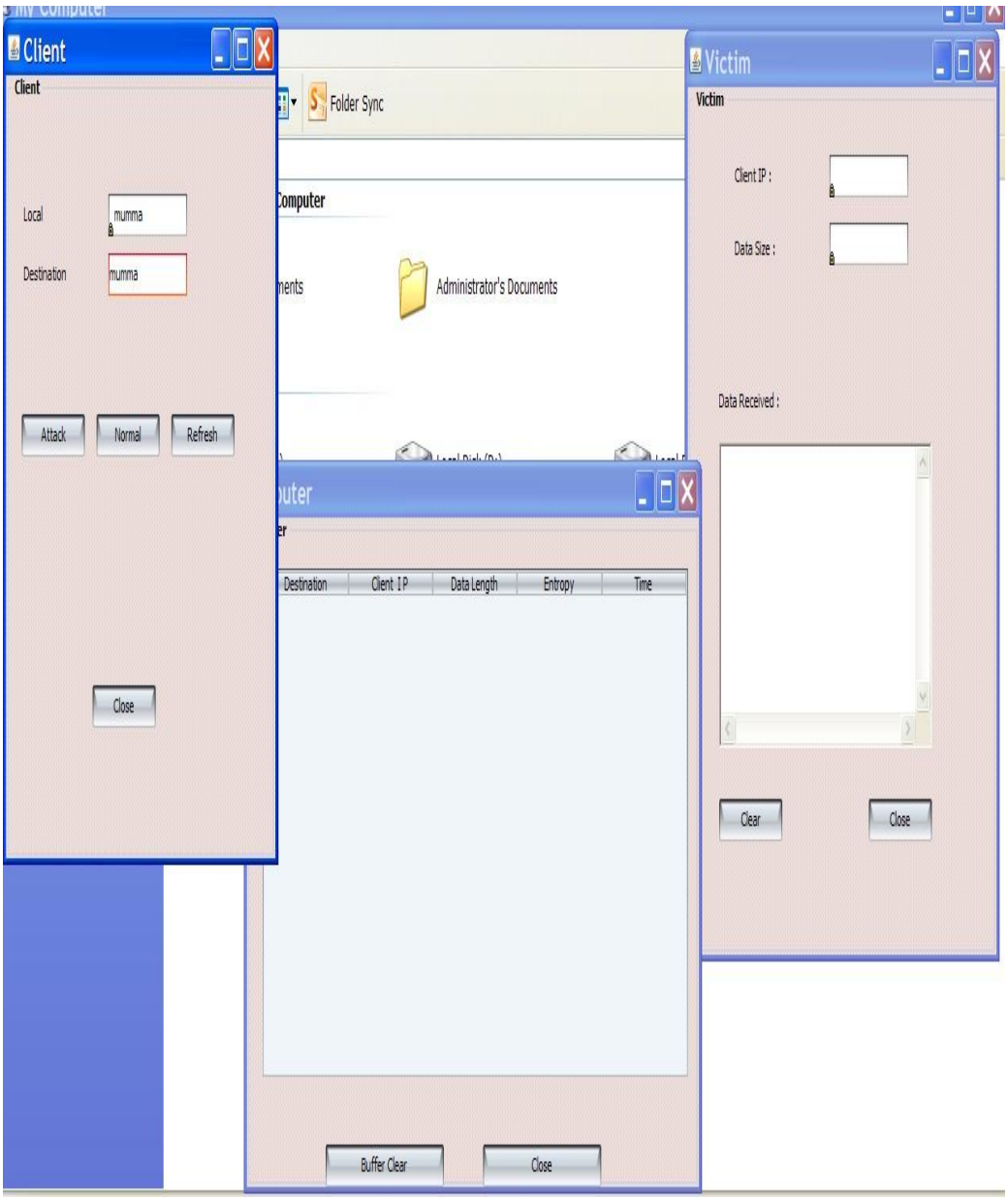


Figure 22 : Client enters the destination system name

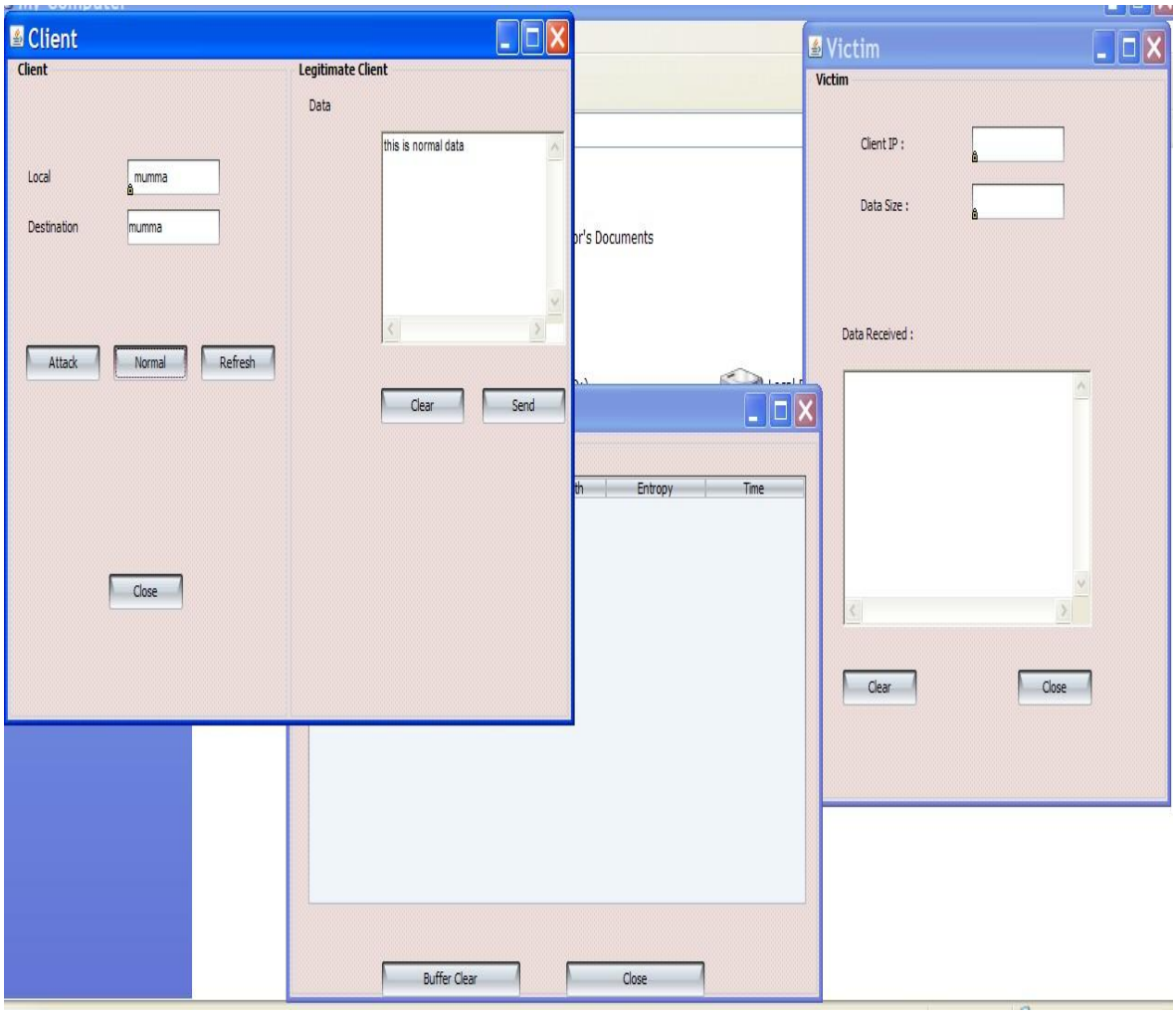


Figure 23 : Client behave as normal client and enters data to send

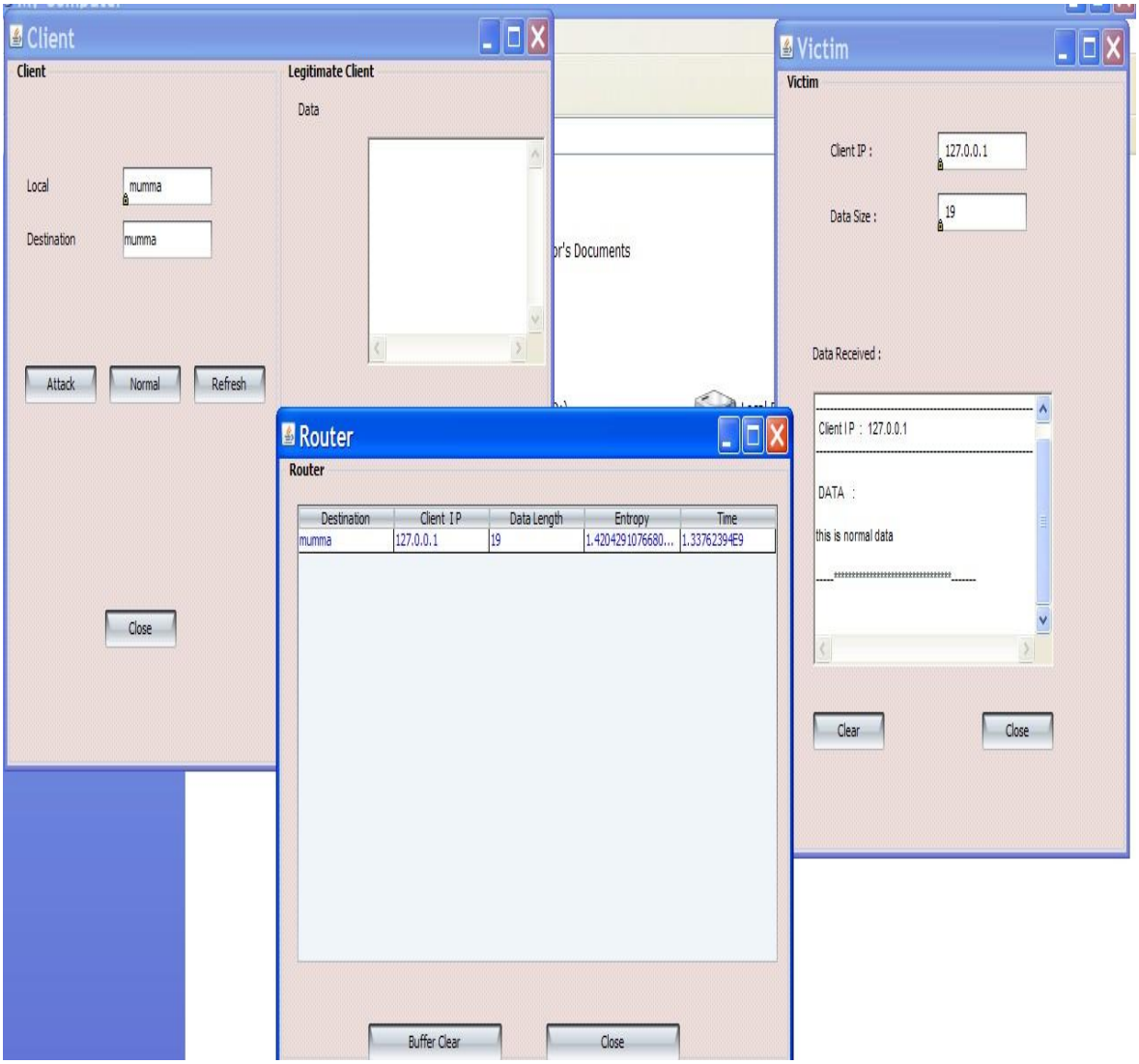


Figure 24 : Client send data , router add enteries to its table and victim receive data

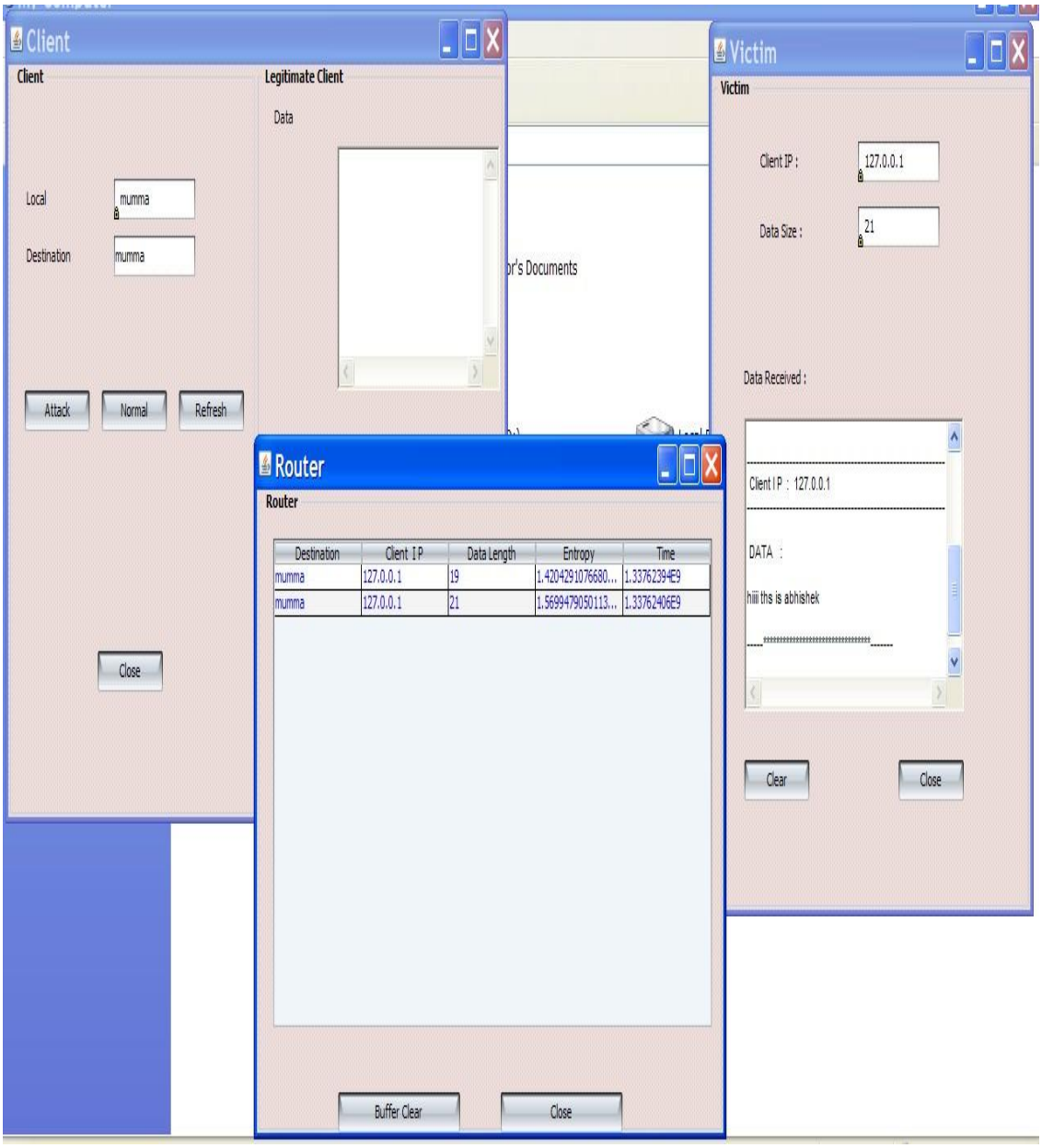


Figure 25 : Again client sends normal data

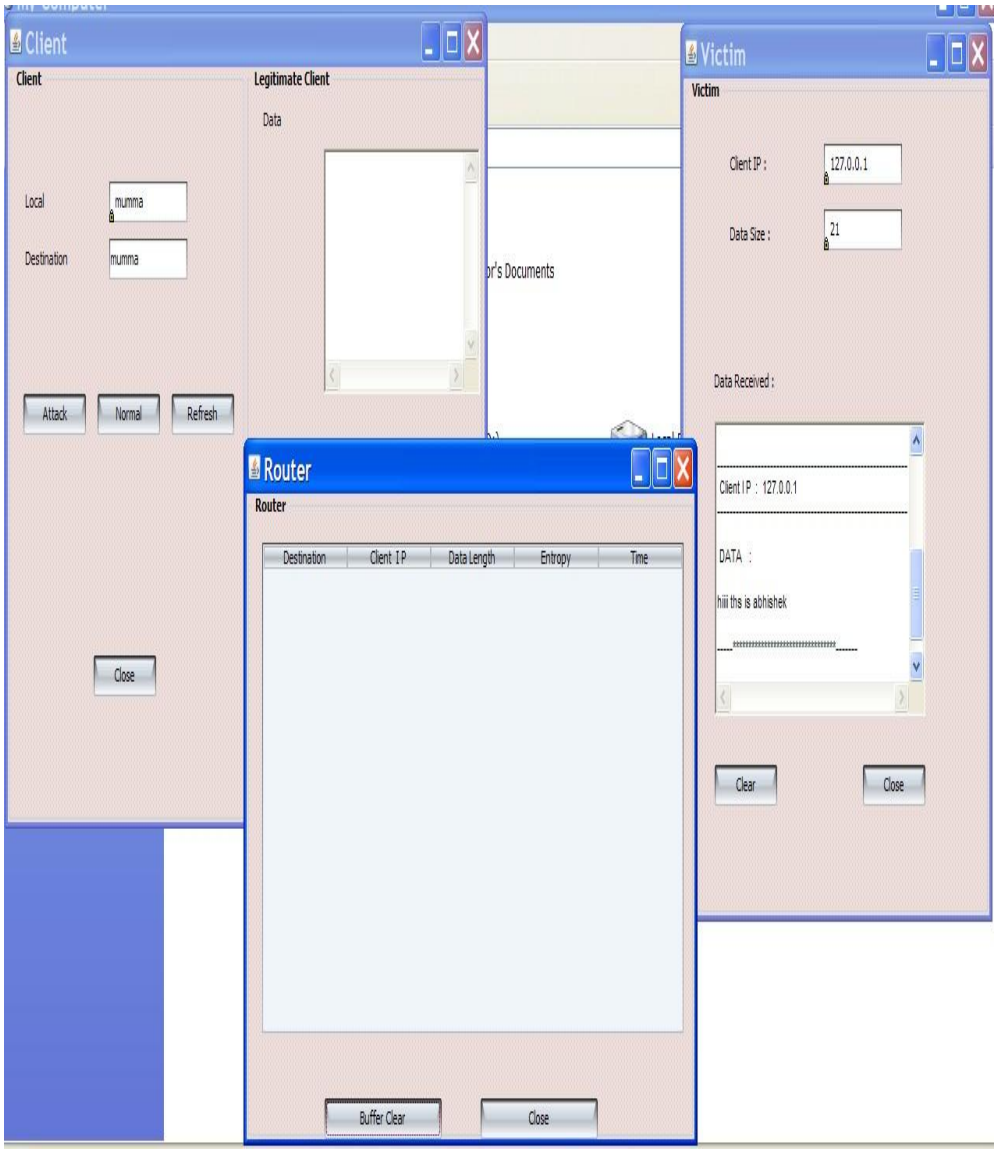


Figure 26 : Buffer clear operation at router

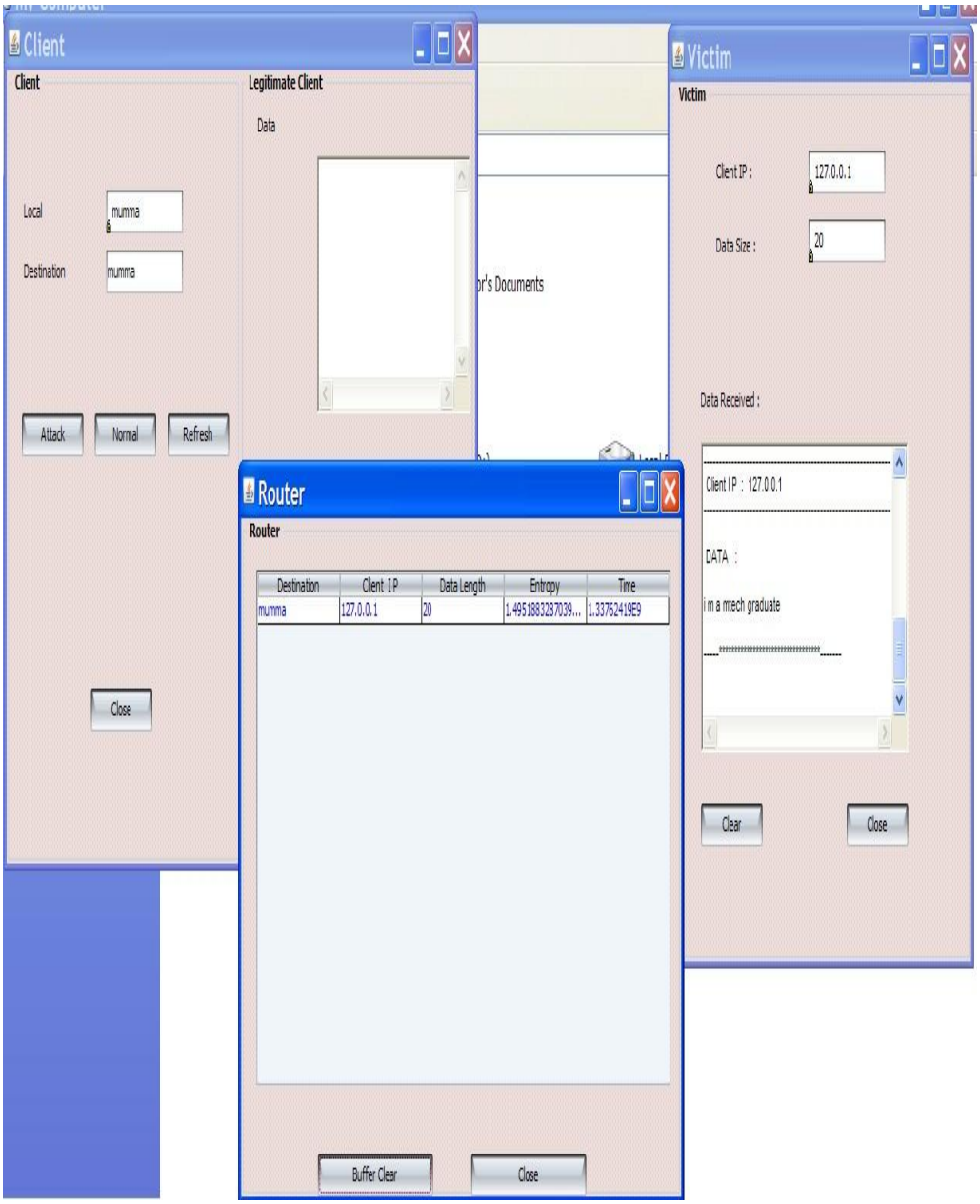


Figure 27 : Again client sends normal data and router updates its table

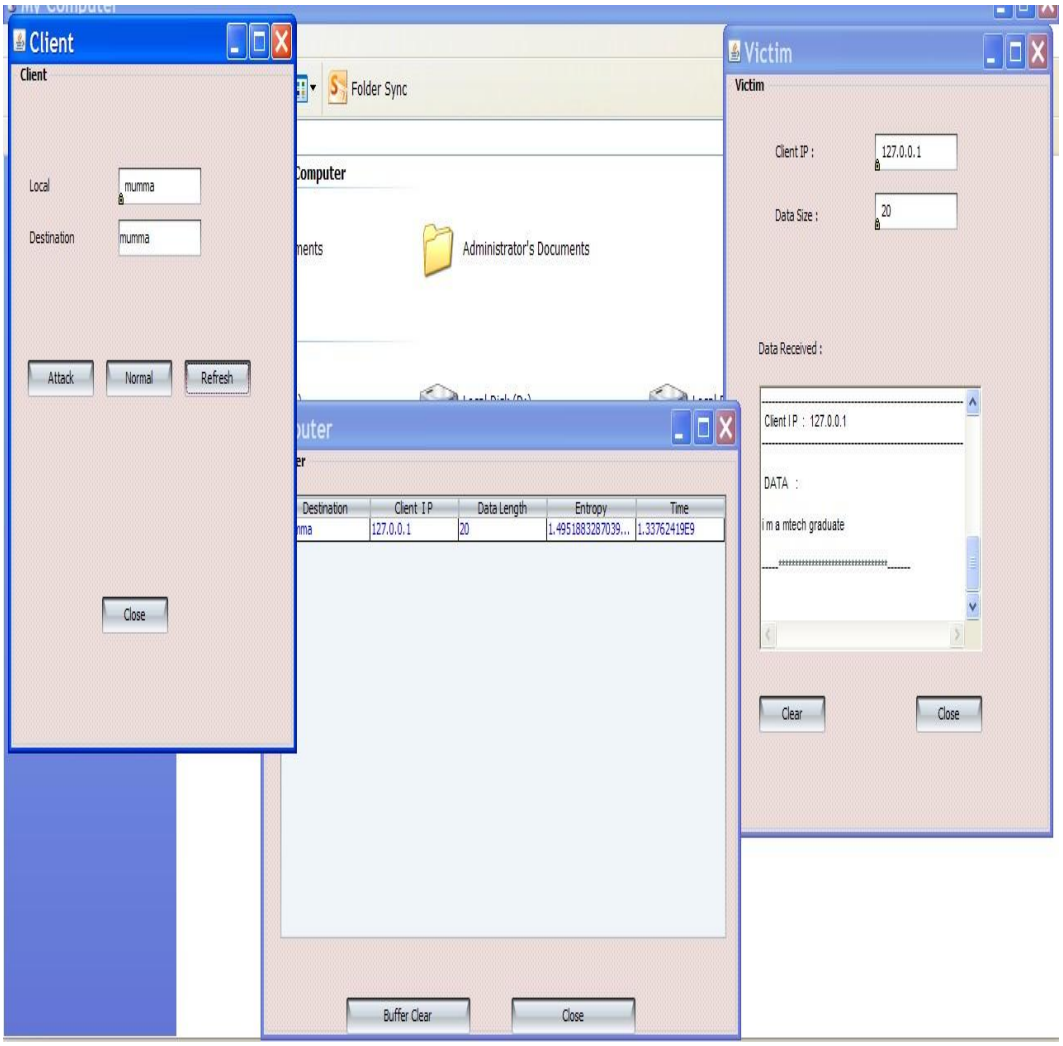


Figure 28 : On click of refresh, client wants to choose attack option

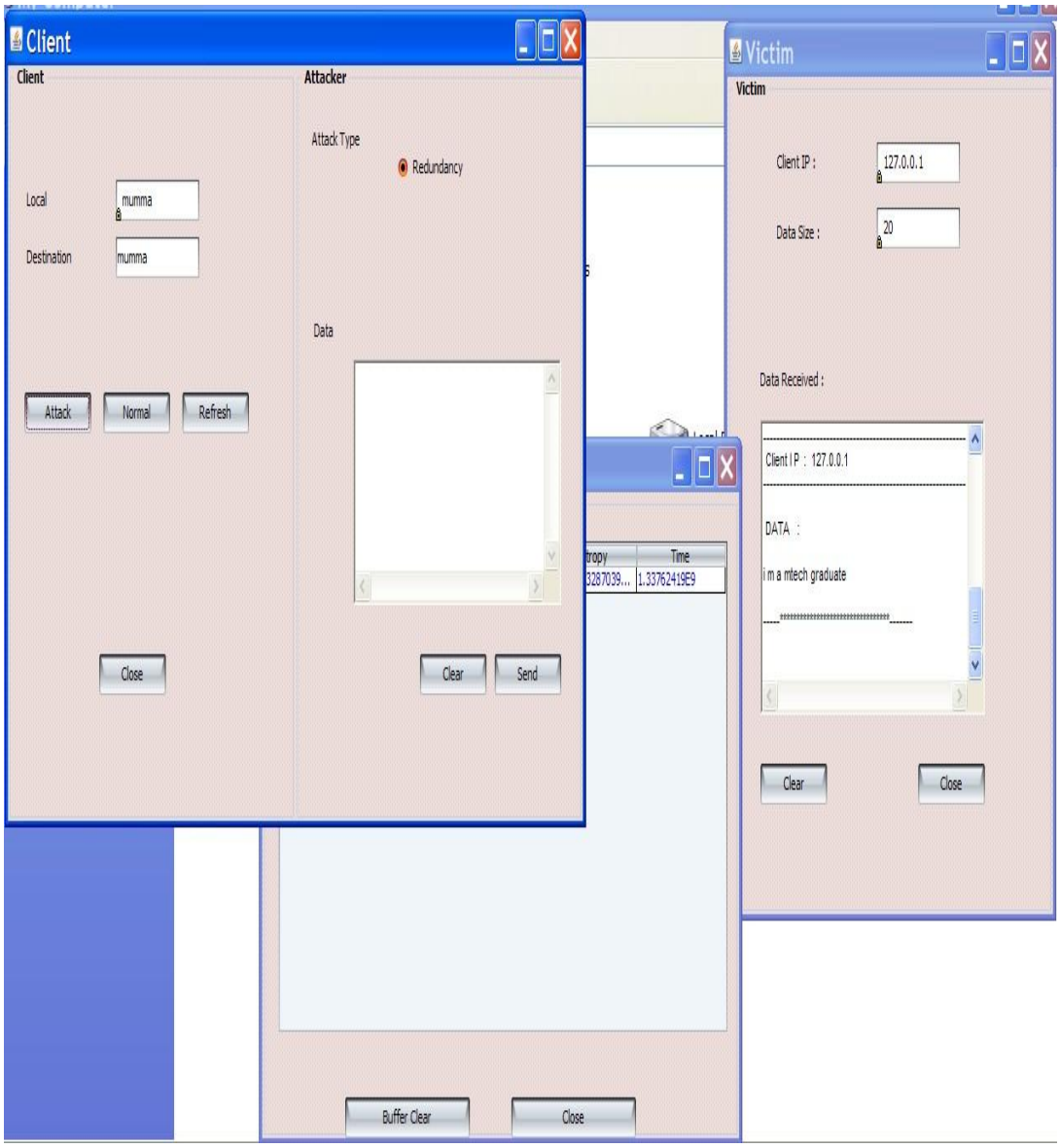


Figure 29 : Client chooses attack option and now behave as an attacker

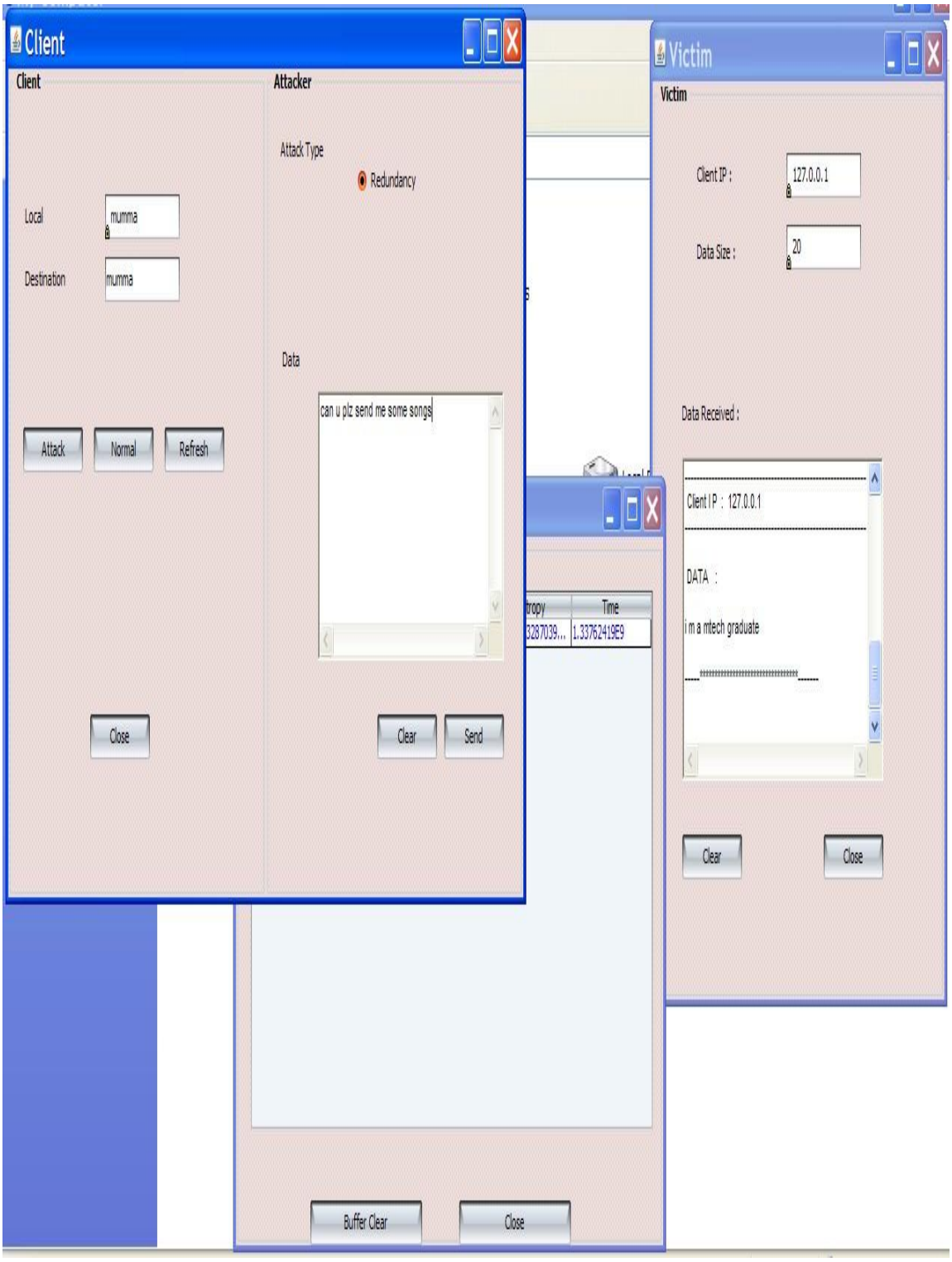


Figure 30 : Attacker client enters data and ready to send redundant data to victim

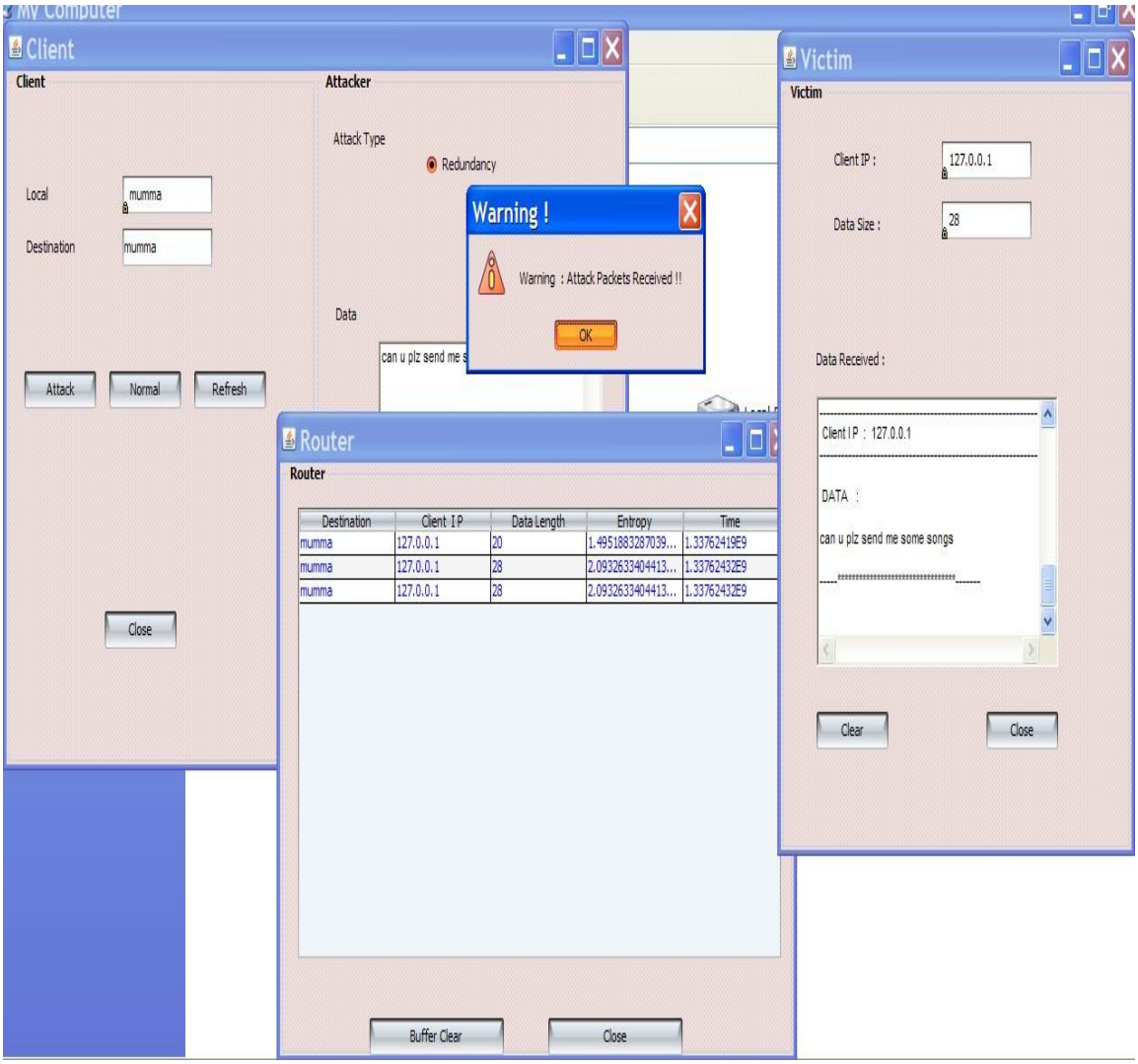


Figure 31: Attacker client sends data, router updates its table entering two times the redundant data, victim detect that this is a attack so warning generated

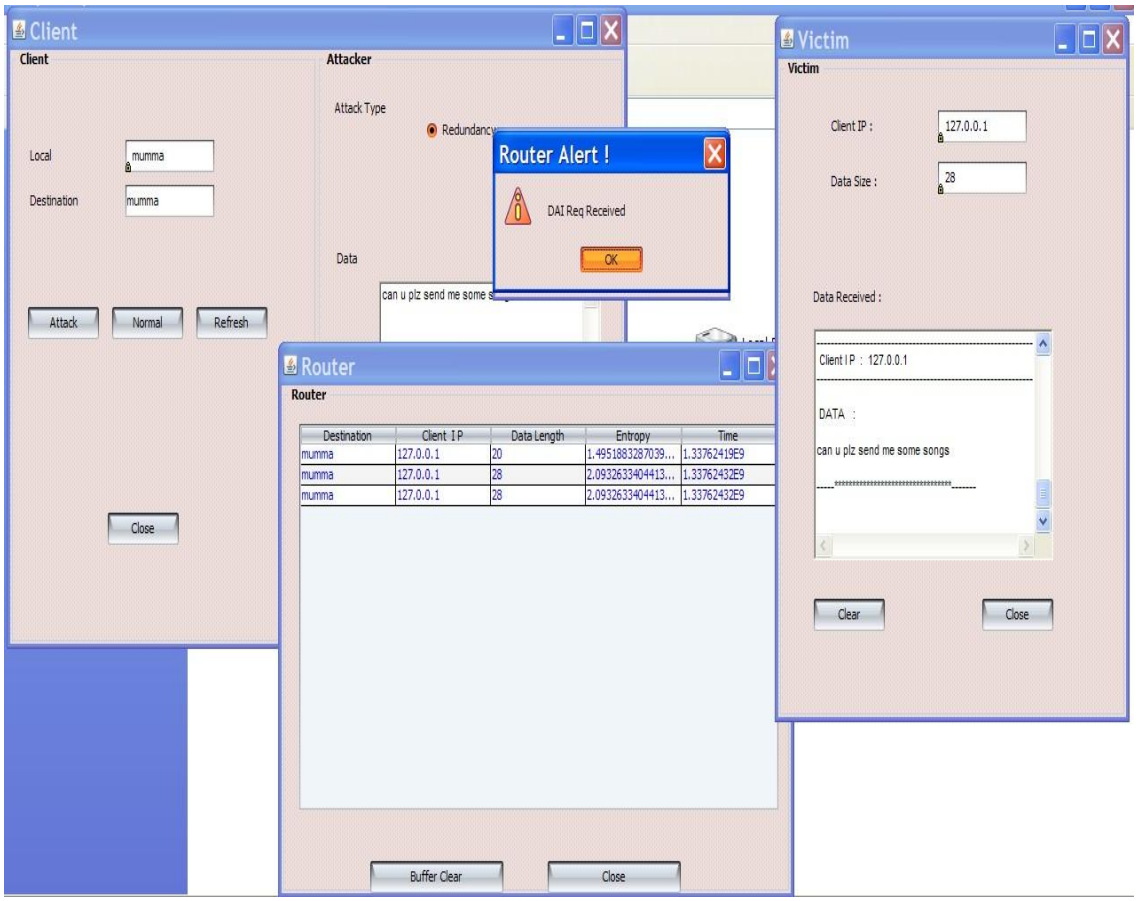


Figure 32 : Victim send the trace back request to router and router receives the trace back request to detect the source of this attack

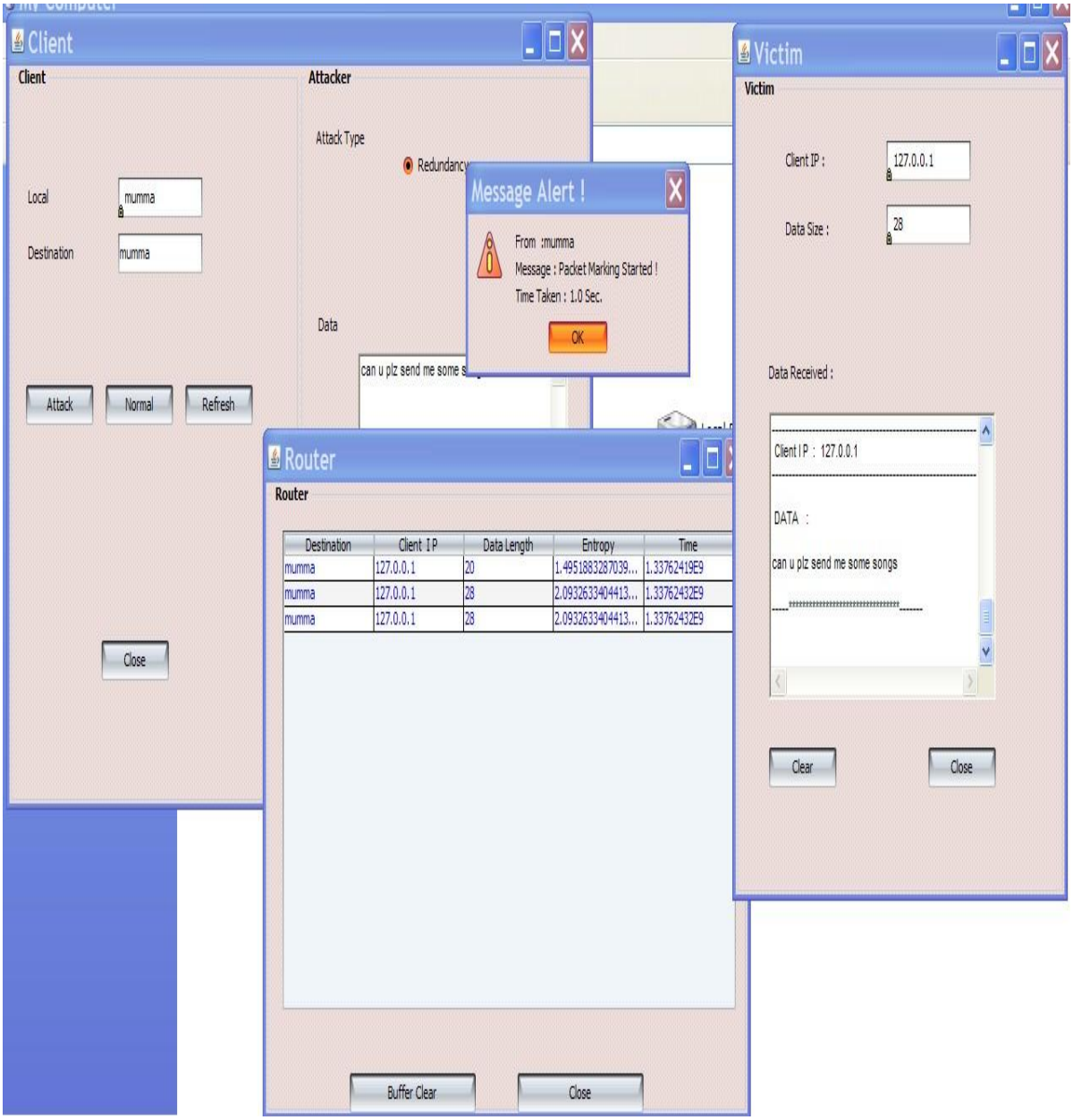


Figure 33 : Router after initiating the trace back algorithm came to know the actual source of this attack and display the name of attacker client

CHAPTER 5

Conclusion And Future work

Conclusion

Although we know that as the cloud goes bigger and bigger threat to its availability also goes higher due to DDoS attack nature. So, there is a need to combat this notorious threat before moving our in-house data towards cloud. Although many work has already be done on mitigating this effect but the research is still going on because no network is completed secure if it is connected to internet.

The traceback algorithm first identifies its upstream routers where the attack flows came from, and then submits the traceback requests to the related upstream routers. This procedure continues until the most far away zombies are identified or when it reaches the discrimination limitation of DDoS attack flows. The results demonstrate that the proposed mechanism works very well in terms of effectiveness and efficiency. Compared with previous works, the proposed strategy can traceback fast in larger scale attack networks.

It can traceback to the most far away zombies within 25 seconds in the worst case under the condition of thousands of zombies. Moreover, the proposed model can work as an independent software module with current routing software. This makes it a feasible and easy to be implemented solution for the current Internet.

Challenges and Future work

The problem in detection of DDoS using IP traceback algorithm and all other methods is that they require continuous access or reach to the path through which the packet flow is coming and in the absence of any path or lan these kind of methods do not work. For example if while traceback algorithm, suppose any of the router or lan is crashed like R4 or Lan0 than the path to the attacker is lost which make impossible to the algorithm to reach to the attacker.

IDS and other devices are vulnerable to DDoS attack

Intrusion detection systems (IDS) are effective if the attacks have signatures associated with them. However, the trend among the attacks is to have legitimate content but bad intent. Intrusion-prevention systems which work on content recognition cannot block behavior-based DDoS attacks.

Since firewalls and IDS are stateful devices, they have to maintain the state of each communication that passes through them. Attackers know this and routinely target this vulnerability by launching connection-layer attacks (i.e. TCP SYN flood) which are designed to flood state tables in firewalls and IDS, ultimately causing these systems to fail and potentially stop all traffic traversing through them, thus completing the DDoS attack for the attacker. Additionally, most DDoS attacks utilize legitimate HTTP (TCP port 80) or HTTPS (TCP port 443) traffic. By default, this traffic is allowed to pass by firewalls and IDS.

IDS devices depend on signature-based detection of known threats; they usually miss a new threat because the signature has yet to be developed. Network-based IDS devices also use anomaly-based detection, which is not effective in detecting and stopping DDoS attacks. Lastly, because IDS devices introduced latency during packet floods attack while performing analysis across multiple links which is unacceptable.

Future Enhancement

Future work could be carried out in the following promising directions:

1. The metric for DDoS attack flows could be further explored. The proposed method deals with the packet flooding type of attacks perfectly. However, for the attacks with small number attack packet rates, e.g., if the attack strength is less than seven times of the strength of non-attack flows, then the current metric cannot discriminate it. Therefore, a metric of finer granularity is required to deal with such situations.
2. Location estimation of attackers with partial information. When the attack strength is less than seven times of the normal flow packet rate, the proposed method cannot succeed at the moment. However, we can detect the attack with the information that have been accumulated so far using

traditional methods, e.g., the hidden Markov chain model, or recently developed tools, e.g., the network tomography.

3. Differentiation of the DDoS attacks and flash crowds. In this research this issue is not considered—the proposed method may treat flash crowd as a DDoS attack, and therefore, resulting in false results.

So in future I will be working to resolve these problems using some other method or framework.

References

[1] Leonard Kleinrock. "An Internet Vision: the invisible global infrastructure. AdHoc Networks", 1(1):3–11, 2003.

[2] F.M. Aymerich, G. Fenu and S. Surcis. "An approach to a cloud computing network. Applications of Digital Information and Web Technologies", 2008. ICADIWT 2008, pages 113 –118, August 2008.

[3] Peter Mell and Timothy Grance, NIST Special Publication 800-145 (September 2011).National Institute of Standards and Technology, U.S. Department of Commerce.

[4] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. "A taxonomy and survey of cloud computing systems. Networked Computing and Advanced Information Management", International Conference on, 0:44–51, 2009.

[5] .E. Smith and R. Nair. "An overview of virtual machine architectures". Pages 1–20, October 2001. http://www.ece.wisc.edu/~jes/902/papers/intro.pdf.

[6] I. Foster, Yong Zhao, I. Raicu, and S. Lu. "Cloud computing and grid computing 360-degree compared." Technical report. Grid Computing Environments Workshop, 2008.

[7] http://www.cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

[8] Guidelines on Security and Privacy in Public Cloud Computing, NIST Special Publication 800-144 December 2011.

[9] Yang Xiang , *Ke Li , and Wanlei Zhou* , Low-Rate DDoS Attacks Detection and Traceback by using new information metrics, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 6, NO. 2, JUNE 2011 page427.

[10] A.Chonka, Y.Xiang, W. Zhou, and A. Bonti, , Cloud Security Defence to Protect Cloud Computing against HTTP-DoS and XML DoS Attacks, *Journal of Network and Computer Applications, Elsevier,* vol. 3, no. 4, pages 1097-1107, 2011.

[11] Muhammad Zakarya , DDoS Verification and Attack Packet Dropping Algorithm in Cloud Computing, World Applied Sciences Journal 23 (11): 1418-1424, 2013 ISSN 1818-4952 © IDOSI Publications, 2013.

[12] Priyanka Negi, Anupama Mishra and B. B. Gupta, Enhanced CBF Packet Filtering Method to Detect DDoS Attack in Cloud Computing Environment, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013 pages 142-146.

[13] S. Savage, "Network Support for IP Traceback," IEEE/ACM Trans. Networking, vol. 9, no. 3, pp. 226-237 June 2001.

[14] Esraa Alomari, Selvakumar Manickam, B. B. Gupta, Shankar Karuppayah, Rafeef Alfaris, "Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art," in International Journal of Computer Applications, (IJCA), Vol. 49, no. 7, pp. 24-32, 2012

[15] A. Srivastava, B.B. Gupta, A. Tyagi, A. Sharma, A. Mishra, et. al., "A Recent Survey on DDoS Attacks and Defense Mechanisms," Book on Advances in Parallel, Distributed Computing, Communications in Computer and Information Science (CCIS), LNCS, Springer-Verlag Berlin Heidelberg, CCIS 203, pp. 570-580, 2011.

[16] Qi Chen, Wenmin Lin, Wanchun Dou, Shui Yu, "CBF: A Packet Filtering Method for DDoS Attack Defense in Cloud Environment", in Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, 978-0-7695-4612-4/11, 2011.

[17] P. Porras and P. Neumann. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proceedings of the Nineteenth National Computer Security Conference*, October 1997.

[18] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *Proceedings of ACM SIGCOMM 2000,* August 2000.

[19] N. Security. NFR Network Intrusion Detection. http://www.nfr.com/products/NID/.

[20] I. S. Systems. Intrusion Detection Security Products. http://www.iss.net/securing_e-business/security products/ intrusion detection/index.php.

[21] MIT Lincoln Laboratory Data Sets [Online]. Available:http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/2000/LLS_DDOS_0.2.2.html

[22] CAIDA, 2010 [Online]. Available: http://data.caida.org/datasets/security/ddos-20070804

[23] D. Moore *et al.*, "Inferring Internet denial-of-service activity," *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, 2006.

[24] T. K. T. Law, J. C. S. Lui, and D. K. Y. Yau, "You can run, but you can't hide: An effective statistical methodology to trace back DDoSattackers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 9, pp.799–813, Sep. 2005.

[25] L. Feinstein *et al.*, "Statistical approaches to DDoS attack detection and response," in *Proc.DARPA Information Survivability Conf. Exposition*, 2003, pp. 303–314.

[26] S. Yu and W. Zhou, "Entropy-Based collaborative detection of DDoS attacks on community networks," in *Proc. 6th IEEE Int. Conf. Pervasive Computing and Communications (PerCom 2008)*, 2008, pp.566–571.

[27] Chong, F; Carraro, G & Wolter, R (2006), 'Multi-Tenant Data Architecture' - available at *http://msdn.microsoft.com/en-us/library/aa479086.aspx*

[28] Leung, AW; Pasupathy, S; Goodson, G & Miller, EL (2008), 'Measurement and analysis of large-scale network file system workloads', ATC'08: USENIX 2008 Annual Technical Conference on Annual Technical Conference, USENIX Association, p. 213-226

[29] Next Generation GRIDs Expert Group (2006), 'Future for European Grids: GRIDs and Service Oriented Knowledge Utilities - Next Generation GRIDs Expert Group Report 3', available at *ftp://ftp.cordis.lu/pub/ist/docs/grids/ngg3_eg_final.pdf*

[30] Fan, X; Weber, WD & Barroso, LA (2007), 'Power Provisioning for a Warehouse-sized Computer'. Proceedings of the 34th International Symposium on Computer Architecture in San Diego, CA. Association for Computing Machinery, ISCA '07 – available at *http://labs. google.com/papers/power_provisioning.pdf.*

[31] Clarke, G (2005), 'Open source taking over Europe - We just don't know it' - available at *http://www.theregister.co.uk/2005/10/21/opensource_government/*

[32] Truffle Capital (2007), 'Truffle Capital: European Commission Recognises the Need for a "European Strategy for Software" - Commenting on the 2007 Truffle 100 Europe, Viviane Reding Calls on Europe to Develop a Leadership Position in Software' - available at *http://www.businesswire.com/news/google/20071122005070/en*

[33] DG Information Society and Media - Directorate for Converged Networks and Service (2009),'Towards A European Software Strategy - Report Of An Industry Expert Group' - available at *http://www.nessieurope.com/Nessi/LinkClick.aspx?fileticket=7teEO5hzywY%3D&tabid=304&mid=1571*

[33] DG Information Society and Media - Directorate for Converged Networks and Service (2009),'Towards A European Software Strategy - Report Of An Industry Expert Group' - available at *http://www.nessieurope.com/Nessi/LinkClick.aspx?fileticket=7teEO5hzywY%3D&tabid=304&mid=1571*

[34] Webhosting Unleashed (2008), 'Cloud-Computing Services Comparison Guide' - available at *http://www.webhostingunleashed.com/whitepaper/cloud-computing-comparison/*

[35] Wayner, P (2008), 'Cloud versus cloud: A guided tour of Amazon, Google, AppNexus, and GoGrid' - available at *http://www.infoworld.com/d/cloud-computing/cloud-versus-cloudguided-tour-amazon-google-appnexus-and-gogrid-122?page=0,0*

[36] Golden, B (2009), 'The Cloud as Innovation Platform: Early Examples' - available at *http://www.nytimes.com/external/idg/2009/06/18/18idg-the-cloud-as-innovation-platformearly-examples-24294.html*