Certificate

It is hereby certify that work which is being presented in this thesis **"Ant Colony Optimization(ACO) based Intrusion Detection System"** , in partial fulfillment of the requirement for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala is an authentic record of my own work carried out under the supervision of **Dr.Maninder Singh** and refers other researcher's work which are duly listed in the reference section. The matter presented in thesis has not been submitted for the award of other degree of this or any other university.

(Tikka Singh)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr.Maninder Singh)

Associate Professor

Computer Science and Engineering Department

Thapar Unversity

Patiala

Countersigned by

(Dr. Rajesh Bhatia) 19/07/10

Head

Computer Science & Engg. Dept

Thapar University,

Patiala

(Dr.R.K.Sharma) 23/7/10

Dean(Academic Affairs)

Thapar University,

Patiala

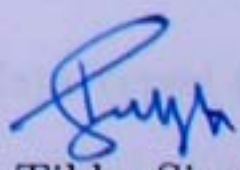
Acknowledgement

I would like to express sincerest thanks to my thesis supervisor Dr.Maninder Singh, Associate Professor, Computer Science and Engineering Department for his inspiration, guidance, simulating suggestions, immense help and support throughout the period of this research work. He has provided me with all the necessary resources including motivation and research environment without which it would not have been possible to complete this work. It was a great opportunity for me to do this work under his supervision.

I would like to thank Dr.Rajesh Bhatia (Assistant Professor and Head), Computer Science and Engineering Department for his moral support and the research he had facilitated for this work.

I would also like to thank all my teachers for their simulating discussions and invaluable support I received during this period of research. I am thankful to the authors whose work I have consulted and quoted in this work.

Finally, I wish to thank my dearest family and friends especially Yogesh,Vandana, Chinki, Shirin, Anoop, Rashmi, Sushilla for all their immense love, enthusiastic encouragement and support throughout my life without which it would not have been possible to complete this work. Last but not the least I would like to thank God who has always been with me in my good and bad times.



Tikka Singh

(800832025)

Ant Colony Optimzation(ACO) based Intrusion Detection System

Thesis submitted in the partial fulfillment of the requirements for the award of degree of

Master of Engineering

In

Computer Science & Engineering

by

Tikka Singh

Roll No 800832025

under the supervision of

Dr. Maninder Singh

(Associate Professor)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

Patiala-147004

July-2010

Certificate

It is hereby certify that work which is being presented in this thesis **"Ant Colony Optimization(ACO) based Intrusion Detection System"** , in partial fulfillment of the requirement for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala is an authentic record of my own work carried out under the supervision of **Dr.Maninder Singh** and refers other researcher's work which are duly listed in the reference section. The matter presented in thesis has not been submitted for the award of other degree of this or any other university.

(Tikka Singh)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr.Maninder Singh)
Associate Professor
Computer Science and Engineering Department
Thapar Unversity
Patiala

Countersigned by

(Dr. Rajesh Bhatia)
Head
Computer Science & Engg. Dept
Thapar University,
Patiala

(Dr.R.K.Sharma)
Dean(Academic Affairs)
Thapar University,
Patiala

Acknowledgement

I would like to express sincerest thanks to my thesis supervisor Dr.Maninder Singh, Associate Professor, Computer Science and Engineering Department for his inspiration, guidance, simulating suggestions, immense help and support throughout the period of this research work. He has provided me with all the necessary resources including motivation and research environment without which it would not have been possible to complete this work. It was a great opportunity for me to do this work under his supervision.

I would like to thank Dr.Rajesh Bhatia (Assistant Professor and Head), Computer Science and Engineering Department for his moral support and the research he had facilitated for this work.

I would also like to thank all my teachers for their simulating discussions and invaluable support I received during this period of research. I am thankful to the authors whose work I have consulted and quoted in this work.

Finally, I wish to thank my dearest family and friends especially Yogesh,Vandana, Chinki, Shirin, Anoop, Rashmi, Sushilla for all their immense love, enthusiastic encouragement and support throughout my life without which it would not have been possible to complete this work. Last but not the least I would like to thank God who has always been with me in my good and bad times.

Tikka Singh

(800832025)

Abstract

Security is a big issue for all networks in todays enterprise environment. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Many methods have been developed to secure the network infrastructure and communication over the Internet, among them the use of firewalls, encryption, and virtual private networks. Intrusion detection is a relatively new addition to such techniques. Intrusion detection methods started appearing in the last few years. Using intrusion detection methods, you can collect and use information from known types of attacks and find out if someone is trying to attack your network or particular hosts. The information collected this way can be used to harden your network security, as well as for legal purposes. Both commercial and open source products are now available for this purpose. Many vulnerability assessment tools are also available in the market that can be used to assess different types of security holes present in your network. There are different techniques to detect Intrusions.

This work explain working of IDS specifically Bro and also suggests the use of ACO for Intrusion Detection. The major emphasis is on the design and development of the policy scripts to detect various network intrusions. These scripts are written using the scripting language of Bro, which supports various special data types to support network level activities. It also has signature-matching features to make threat signatures to match against various attacks and detect them later.

Contents

List of Figures

Chapter 1

Introduction

1.1 Network Security

The protection of a computer network and its services from unauthorized modification, destruction, or disclosure. A computer network consists of many different parts, including networking devices such as firewalls, routers and switches that are responsible for the transmission of data into and out of the network. Networking devices are critical to the way information is stored and exchanged online today, and any threats to the security of these devices must be taken seriously. Network security threats and unauthorized access can come in many forms. One of the most common is hackers, who often exploit networking devices for personal gain or profit. Gaining access to the large amount of secret information that is inside the network such as social security numbers, credit card numbers and other personally identifiable information can be used for identity theft and other personal gains. Network security is critical because threats to computer networks can unsecure communication channels, as well as cost time, money and resources. Many methods have been developed to secure the infrastructure and communication over the internet, among

them the use of firewall, encryption and virtual private networks. Intrusion detection is a relatively new addition to such techniques. Intrusion detection methods started appearing in the last few years. Using intrusion detection methods, you can collect and use information from known types of attacks and find out if someone is trying to attack your network or particular hosts.

Network Security is becoming an important issue for all the organizations, and with the increase in knowledge of hackers and intruders they have made many successful attempts to bring down high-profile company networks and web services. With the recent advances in the field of network security a technique called Intrusion Detection System are develop to further enhance and make your network secure. It is a way by which we can protect our internal network from outside attack, and can take appropriate action if needed. Using intrusion detection methods, information can be collected from known types of attack and can be used to detect if someone is trying to attack the network. Many techniques are there to detect intrusion in a network like signature matching, anomaly based and others. So there is a need to protect these networks from viruses, hackers, unauthorized attack and other network vulnerabilities that are available.

1.2 Types of Attack in a Network

1.2.1 SYN Flooding

The SYN flood attack is, simply, to send a large number of SYN packets and never acknowledge any of the replies. A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system. The attacker sends several packets but does not send the "ACK" back to the server. The connections are hence half-opened and consuming server resources, a legitimate

user, tries to connect but the server refuses to open a connection[20].

1.2.2 Buffer Overflow Attacks

A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information, which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them. Although it may occur accidentally through programming error, buffer overflow is an increasingly common type of security attack on data integrity. In buffer overflow attacks, the extra data may contain codes designed to trigger specific actions, in effect sending new instructions to the attacked computer that could, for example, damage the user's files, change data, or disclose confidential information[20].

1.2.3 Packet Sniffing

A packet sniffer, sometimes referred to as a network monitor or network analyzer, can be used legitimately by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the Packet sniffer, an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission. In its simple form a packet sniffer simply captures all packets of data that pass through a given network interface. Typically, the packet sniffer would only capture packets that were intended for the machine in question. However, if placed into promiscuous mode, the packet sniffer is also capable of capturing all packets traversing the network regardless of destination. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyze all of the network traffic. Within a

given network, username and password information is generally transmitted in clear text which means that the information would be viewable by analyzing the packets being transmitted[30].

1.2.4 Viruses

A small piece of software that replicates itself on real programs and runs every time a program runs .Most can reproduce and attack other programs. The following are the most common types of viruses[20]:

- E-mail viruses: Moves around in e-mail messages, usually replicates itself by automatically mailing itself to dozens of people in the victim's e-mail address book.
- Worms: A small piece of software that uses computer networks and security holes to replicate itself. Worms can expand very rapidly scans a network for another machine that has a specific security hole and copies itself to the machine.
- Trojan horses: A computer program that contains hidden functions that can exploit the privileges of the user running the program. Can erase the disk, send your credit card numbers, and password to the hackers.

1.2.5 Spyware

Spyware is a technology that gathering information about a person or organization without their knowledge. On the Internet spyware is programming that is put in someone's computer to secretly gather information about the user and relay it. Spyware can get in a computer as a software virus or as the result of installing a new program. There are three classes of Spyware[29]:

- This will change the default homepage of your browser to some target ads, pop up etc.

- Information collection: This class of spyware is generally interested in collecting some kind of useful information about you, the sites you visited most, and so that third party can send you targeted popup and ads.

- Malicious: This class includes full logging and collecting information along with sending private and confidential information to the server

1.2.6 Denial-of-Service Attack

A denial-of-service attack or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its actual users. The means to carry out, motives for, and targets of a DoS attack may vary; it generally consists of the concerted efforts of a person or people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root nameservers. The term is generally used with regards to computer networks but is not limited to this field, for example, it is also used in reference to CPU resource management[28].

1.2.7 Flood Attack

The earliest form of denial of service attack was the flood attack. The attacker simply sends more traffic than the victim could handle. This requires the attacker to have a faster network connection than the victim. This is the lowest-level of the denial of service attacks, and also the most difficult to completely prevent, for example a UDP flood attack is a denial of service attack (DoS) attack using User

Datagram protocol, a session less/connectionless computer networking protocol. An UDP protocol attack can be initiated by sending large number of UDP protocol packets to random ports on a remote host. As a result the random host will[28]:

- Check for application listening on that host.
- Sees that no application listens on that port.
- Reply with an ICMP destination unreachable packet.

1.2.8 Social Engineering

Social engineering is the name given to a category of security attacks in which someone manipulates others into revealing information that can be used to steal, data, access to systems, access to cellular phones, money or even your own identity. Such attacks can be very simple or very complex. Gaining access to information over the phone or through web sites that you visit has added a new dimension to the role of the social engineer. A skilled social engineer will often try to exploit this weakness before spending time and effort on other methods to crack passwords. The goal of social engineering is to trick someone into providing valuable information or access to that information.

1.2.9 Phishing

It's a kind of important spam; in it some links are provided to the user in the email and the user asked to follow the instructions and condition connect to certain bank, organization, E-commerce web sites or company to verify their account no., username, passwords and then hacker/attacker can exploits your account. These kinds of links and shortcuts of web sites are known as phishing attacks[32].

1.3 Network Security Wheel

The network security wheel is a methodology how the network security of an enterprise is maintained. The notion of 'wheel' here says that network security is a continuous process. In other words, in order to keep the wheel rolling, network manager in an enterprise should always maintain four steps[31]:

- Secure
- Monitor
- Test
- Improve

The policy is at the centre of everything. It means that all these four steps are regulated and executed according to the policy. Because of this, the presence of a good security policy is critical.

- Secure: In this part of security wheel main thing is that to secure the networks. This is the step where implement our security solutions in the organization. Firewalls, authentication, encryption are included in this step.
- Monitor Security solutions implemented in the previous step. So in this moniter part the security manager can monitoring the traffic if a security breach exists. The IDS solution implemented in this stage. This step can also be used to validate our security solutions.
- Test Where the security network manager /engineers try to break their own security solutions. We can think of this step as penetration tester's kind of job. However sometimes we can always test the security from the business point-of-view.

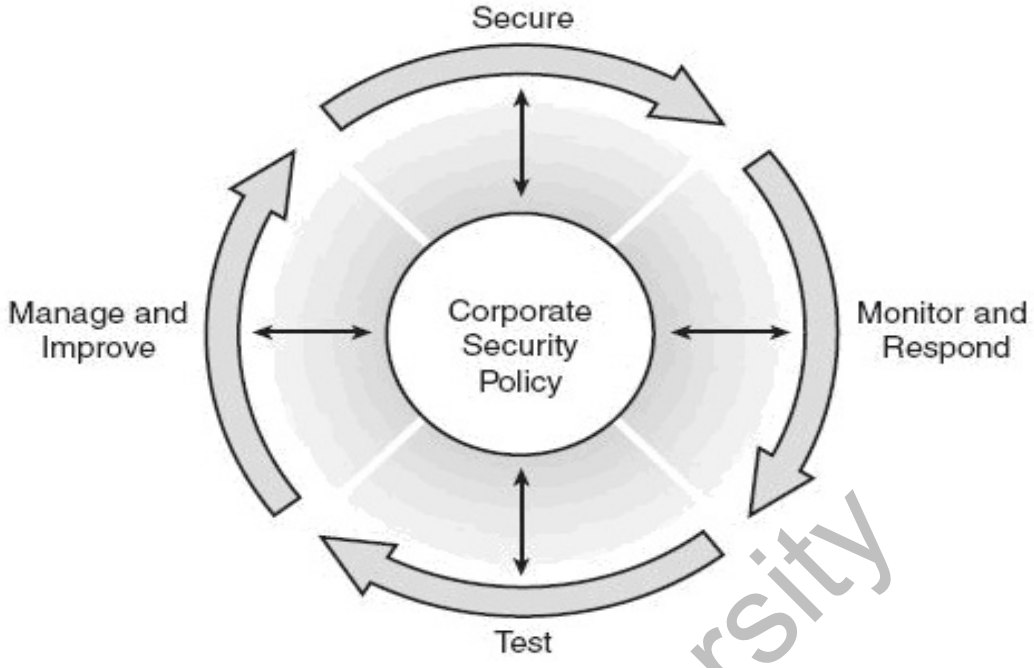


Figure 1.1 Security Wheel [31]

- Improve This step as the continuation of the previous step. Once a breach find or something that hinders employee's productivity, then improve it there. This may also be a good place to change our security policy.

1.4 Security Tools and Techniques

1.4.1 Firewall

A firewall is the combination of both hardware and software used to prevent unauthorized programs or internet users from accessing a private network and a single host computer. Firewall and intrusion detection systems both are software's that protect your network or computer from various intrusions and attackers. The firewall can only block the external viruses and unauthorized access in the network. Looking for complete solution to protection against anomaly based attacks the only

way to go is the combination of firewall and intrusion detection system. There are two types of firewalls, hardware based and software based. At the same time you can use both kind of firewalls, but still may not give you enough protection[18].

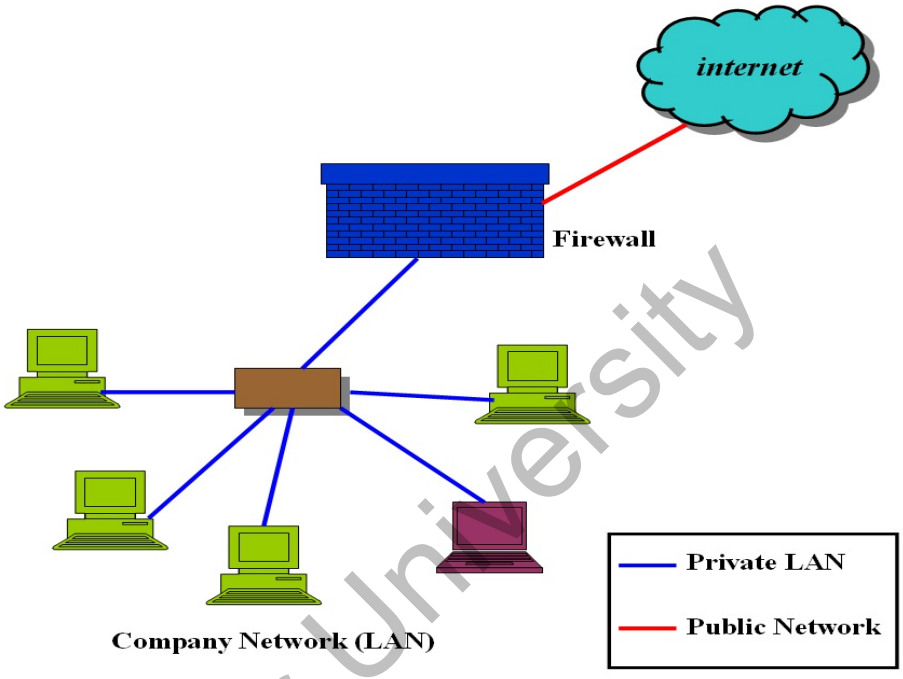


Figure 1.2 Firewall [18]

Firewalls blocking the port communication between computer system and the network in which you are connected. So port scan describes some kinds of weak links in the communication. The firewalls can block some kinds of attacks but there is also a danger of hacker guessing that computer system is having a firewall in between into the network. By using the right techniques, the hacker might even able to exact the name of the firewall; this is helpful for the hacker to exploit your system. Implementing firewall you can set rules to allow/disallow traffic through ports and programs. But firewalls have some limitations. They can only protect inbound and outbound internet connections. If somebody or hacker already gained access to your system by installing a malicious program on your computer, then firewall not able

to do anything. Firewall is a set that protects and separates inner private network from Internet, it can filter packets according to information of source address, destination address, port and packet state that contained in IP packets transmitted on network, and control whether the packets pass or not. Firewall becomes a popular method to control visit to network. This is a usually used and effective method to keep away hackers. Consider some advantages of adopting firewall for the protection of a network or system.

- Ensure safe visit to host and applications
- Ensure security of several clients and server
- Protect pivotal department from inner or outer attacking, provide safe channels for employee, client and provider that pass Internet and visit long-distance computer.

1.4.2 Intrusion Detection System

The act of detecting actions that attempts to compromise the confidentiality, integrity or availability of a resource. Intrusion is the act of violating the security policy or legal protections that pertain to an information system. An intruder is somebody ("hacker" or "Cracker") attempting to break into or misuse your system. Many methods have been developed to secure your network infrastructure and communication over the internet, among them the firewall, encryption and virtual private network. Using intrusion detection methods, one can collect and use information from known types of attacks and find out if someone is trying to attack your network. Many vulnerability assessment tools are also available in the market that can be used to assess different types of security holies present in your network. Security system consists of multiple tools, including[3][8]:

- Firewalls are used to block unwanted incoming as well as outgoing traffic of data. There is a range of firewall products available in the market both in open source and commercial products.

- Intrusion detection system (IDS) that are used to find out internal and external intrusion from the network.

- Vulnerability assessment tools that are used to find and plug security holes present in the network. Information collected from vulnerability assessment tools is used to set rules on firewalls so that these security holes are safeguarded from malicious internet users. Several vulnerability tools are available in market such as Nmap and Nessus.

Firewalls and other simple boundary devices lack some degree of intelligence when it comes to observing, recognizing, and identifying attack signatures that may be present in the traffic they monitor and the log files they collect. Without sounding critical of such other systems' capabilities, this deficiency explains why intrusion detection systems (IDS) are becoming increasingly important in helping to maintain proper network security. Whereas other boundary devices may collect all the information necessary to detect attacks that may be getting started or already underway, they haven't been programmed to inspect for and detect the kinds of traffic or network behavior patterns that match known attack signatures or that suggest potential unrecognized attacks may be incipient or in progress.

The simplest way to define an IDS might be to describe it as a specialized tool that knows how to read and interpret the contents of log files from routers, firewalls, servers, and other network devices. Furthermore, an IDS often stores a database of known attack signatures and can compare patterns of activity, traffic, or behavior it sees in the logs it's monitoring against those signatures to recognize when a close

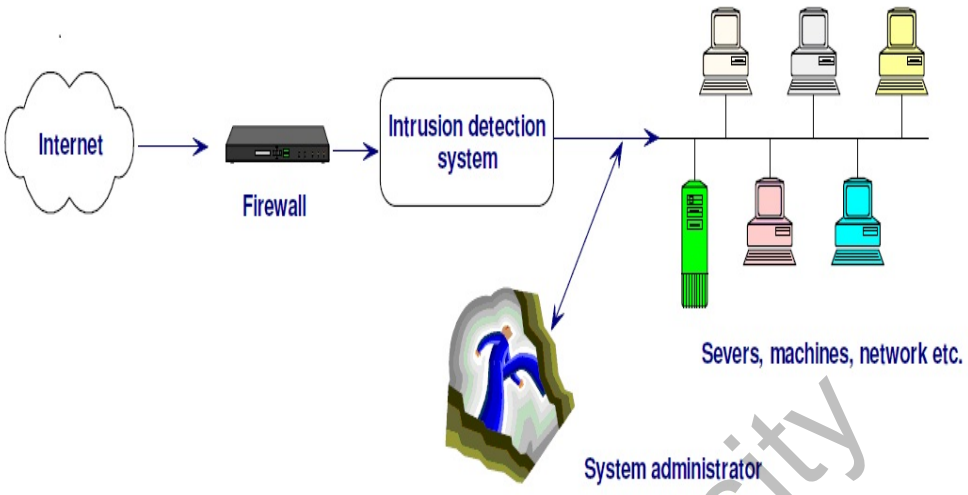


Figure 1.3 Simple IDS environment [8]

match between a signature and current or recent behavior occurs. At that point, the IDS can issue alarms or alerts, take various kinds of automatic action shutting down Internet links or specific servers to launching backtraces, and make other active attempts to identify attackers. Some main points that describe the need of an IDS. These are:

- Provide a greater degree of integrity to infrastructure of the organization.
- Able to trace user activity from entry point to point of impact.
- Will record alteration of data and give a report.
- IDS also helps in monitoring the Internet for the latest attacks
- Notify us when the system will be under attack.
- Analysis of abnormal activity patterns

Intruder's Definition

An Intruder is a person who attempts to gain unauthorized access to a system, to damage that system, or to disturb data on that system. In summary, this person attempts to violate Security by interfering with system Availability, data Integrity or data Confidentiality. Another category of intruders includes persons within a system misusing privileges to gain access to information and systems they are unauthorized to use.

Type of Intruders

There are basically two types of intruders: external and internal. External intruders do not have any authorized access to the system they attack. Internal intruders on the other hand have some access rights, but they seek to gain additional capability.

Intrusion detecting Techniques

Intrusion detection system mostly used two types of intrusion detecting techniques, these are: Anomaly detection and Signature Recognition or Signature based.

- Anomaly based detection Anomaly Intrusion Detection Systems are designed to detect general misuse and attacks for which no signature exists. This attempts to construct a model according to the statistical knowledge about the normal activity of the computer system like CPU utilization, user logins, disk activity, file activity etc.

- Signature Recognition or pattern matching detection uses signatures or rules that describe undesirable events. This approach allows the detection of intrusions which the system has learned their signatures perfectly. IDSs contain databases with strings based patterns for known hacker techniques. The IDSs

examine the network traffic, looking for well-known patterns of attack.

Architecture of Intrusion Detection System

When we are discussing about the architecture or structure of an IDS, we find out following components, these are: Sensors, Monitors, Resolver, and Controller[9].

- Sensors: These are the data gathering components of IDS. These are similar monitoring processes on networked hosts those extract information from the hosts' event logs, audit information, application logs etc. Sensors may also be dedicated network monitors connected to an observation point on a network segment. A network monitor inspect all visible network traffic, these system filter the event logs, generating summaries that are used by IDS Monitors.

- Monitors: These are the processing unit or segment of IDS. The event summaries or database which taken by sensors are inspected for suspicious activity and suspicious reports are generated. These suspicious reports are forwarded to next higher level of monitor or to resolver units.

- Resolver: These resolvers receives suspicious contests or reports from monitors and responses like reporting to an administrator, changing the behavior of lower level components or reconfiguring other security mechanism such as firewalls.

- Controller: In distributed IDS, where configuration of components is possible via centralized controllers. Controllers segments or components specify administration and allow administrative personnel to reconfigure IDS component.

The DIDS architecture combines distributed monitoring and data reduction with centralized data analysis. DIDS architecture consists of DIDS director, a single host

monitor per host and a single LAN monitor for each broadcast LAN segment in the network which is monitored. In DIDS, the host and LAN monitors report events, which possibly lead to intrusive activity, to a centrally-located DIDS director. The director employs an expert system to detect the possible intrusion attacks. This architecture provides accountability by trying the users with their actions.

The host and LAN monitors are responsible for the collection of evidence of suspicious activity and DIDS director is responsible for its evaluation. Reports are sent

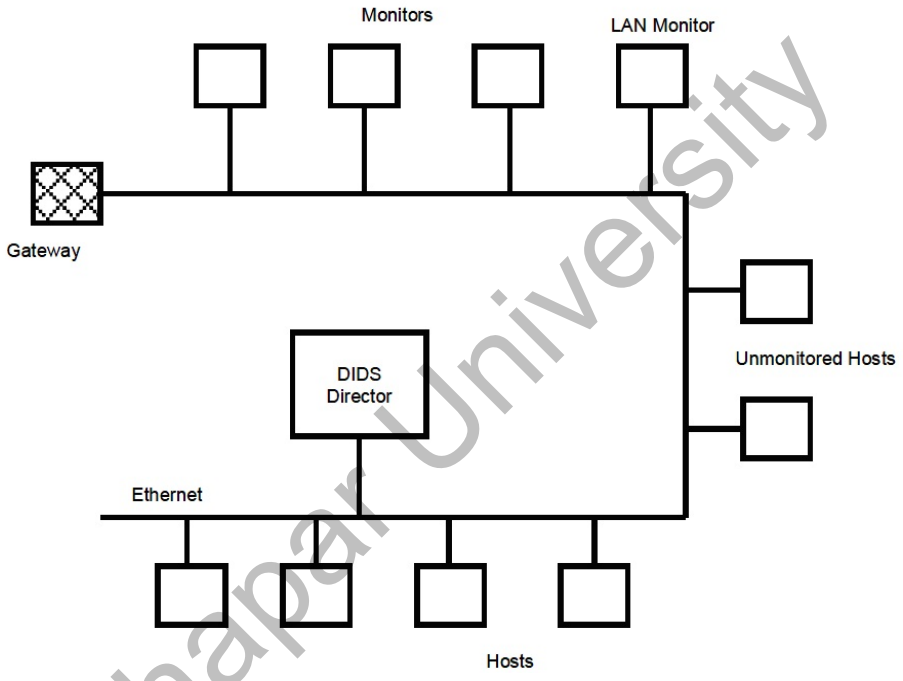


Figure 1.4 Distributed IDS environment [3]

independently and asynchronously from the host and LAN monitors to the DIDS director through a communications architecture shown in figure 1.2. For high level communication protocols between the components are based on Common Management Information Protocol (CMIP) recommendations. The architecture provides a bidirectional communication between the DIDS director and any monitor in the configuration. The architecture provides bidirectional communication between the DIDS director and any monitor in the configuration and the communication con-

sists of notable events and anomaly reports. The director makes requests for more detailed information from the distributed monitors. The host monitor consists of host event generator and host agent. The agent generator collects and analysis audit records from the host operating system, in which, the audit records are scanned for notable events. The notable events are sent to the director of the next analysis. The LAN monitor consists of a LAN event generator and a LAN agent. The LAN event generator is a subset of NSM (Network Status Monitor) and is responsible to observe all the traffic on its segment of the LAN, in order to monitor host-to-host connections, services used and volume of traffic. The DIDS director consists of three major components namely a communication manager, an expert system and a user interface. The communication manager is used to transfer data between the director and it accepts the notable event records from each host and LAN monitors and sends them to the expert system. It also sends request to the host and LAN monitors for information regarding a particular user. The expert system is responsible for evaluating and reporting the security state of the monitored system and it receives the reports from the hosts and the LAN monitors. This makes inferences about the security of each individual host and the expert system is having simple learning capabilities[3].

There are several algorithms used to perform both static and adaptive or dynamic routing in to the network. The routing is the process of selecting paths in a network along which to send network traffic. Routing is performed for many kinds of networks, including the telephone network, electronic data networks (such as the Internet), and transportation networks. The routing process usually directs forwarding on the basis of routing tables which maintain a record of the routes to various network destinations. Most routing algorithms use only one network path at a time,

but multipath routing techniques enable the use of multiple alternative paths. So when any Firewall or IDS implemented, they follows some rules and mathematical equations approach for transmission from source to destination node with in a network. From the several algorithms Ant Colony Optimization algorithm is one of most popular these days by virtue its fast speed for finding best results.

1.5 Ant Colony Optimization

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs[16][17][7]. This algorithm is a member of ant colony algorithms family, in

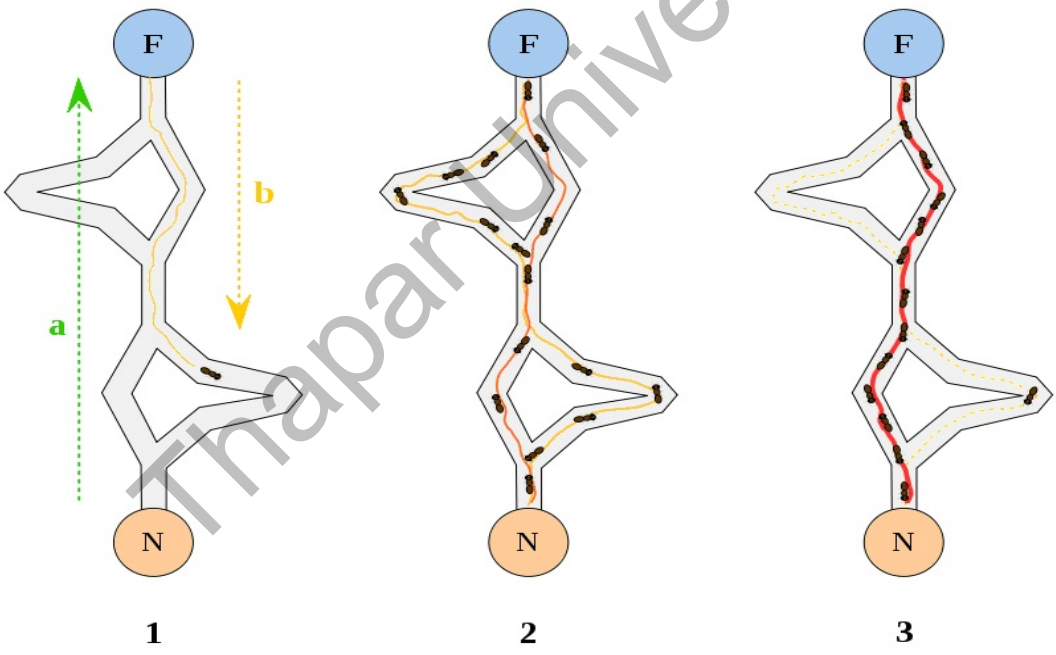


Figure 1.5 Ant Colony Optimization [33]

routing methods, the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. Ant as a single individual has a very limited effectiveness. But as

a part of a well organized colony, it becomes one powerful agent, working for the development of the colony. The ant lives or the colony and exists only as a part of it. Each ant is able to communicate, learn, cooperate, and all together they are capable of develop themselves and colonize a large area.

They manage such great successes by increasing the number of individuals and being exceptionally well organized. The self organizing principles they are using allow a highly coordinated behavior of the colony. The original idea comes from observing the exploitation of food resources among ants, in which ants' individually limited cognitive abilities have collectively been able to find the shortest path between a food source and the nest.

1. The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b).

2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route.

3. Ants take the shortest route; long portions of other ways lose their trail pheromones.

In a series of experiments on a colony of ants with a choice between two unequal length paths leading to a source of food, biologists have observed that ants tended to use the shortest route. A model explaining this behavior is as follows:

- An ant runs more or less at random around the colony.

- If it discovers a food source, it returns more or less directly to the nest, leaving in its path a trail of pheromone.

- These pheromones are attractive, nearby ants will be inclined to follow, more or less directly, the track.

- Returning to the colony, these ants will strengthen the route.
- If two routes are possible to reach the same food source, the shorter one will be, in the same time, traveled by more ants than the long route will.
- The short route will be increasingly enhanced, and therefore become more attractive.
- The long route will eventually disappear, pheromones are volatile.
- All the ants have determined and therefore chosen the shortest route.

Chapter 2

Literarture Survey

Intrusion detection is a set of techniques and methods that are used to detect suspicious activity both at the network and host level. Intrusion detection systems fall into two basic categories: signature-based intrusion detection systems and anomaly detection systems. Intruders have signatures, like computer viruses, that can be detected using software. You try to find data packets that contain any known intrusion-related signatures or anomalies related to Internet protocols. Based upon a set of signatures and rules, the detection system is able to find and log suspicious activity and generate alerts.

Anomaly-based intrusion detection usually depends on packet anomalies present in protocol header parts. In some cases these methods produce better results compared to signature-based IDS. Usually an intrusion detection system captures data from the network and applies its rules to that data or detects anomalies in it. Intrusion Detection System or IDS is software, hardware or combination of both used to detect intruder activity. An IDS may have different capabilities depending upon how complex and sophisticated the components are. IDS appliances that are a combination of hardware and soft-

ware are available from many companies. As mentioned earlier, an IDS may use signatures, anomaly-based techniques or both[8][9].

2.1 Classification of Intrusion Detection System

Intrusion Detection Systems can be classified in several categories. These are:

- Deployment based
- Scope based
- Techniques based

2.1.1 Deployment based IDS

Into the Deployment based classification we have two main types of IDS. These are:

- Inline IDS: Inline IDSs are placed in the line of packet transfer. IDS check all packets passing through it for intrusion and detected, alert the network manager. Inline IDS take place some methods transmitting a copy of packets and identifier from the network to an intrusion detection system and analyzing the copy of packet by the IDS to check whether the packet includes a signature attack and communicating a reply message from intrusion detection system to the network gateway.
- Sniffer IDS :Sniffer IDSs that snoop and examine packets within a network. Sniffer IDSs do not come in the way of packet transfer. They

passively receive all the packets from the network and perform intrusion analysis on these packets. Where the flows of packets are analyzes the sniffer Ids placed there. The working of a sniffer to capture the packets for analyzing the network traffic.

2.1.2 Scope based IDS

When we considering the source of data used for intrusion detection, another classification of intrusion detection systems can be used in term of the scope based IDSs. These are:

- Honeypot IDS
- Network Based IDS
- Host Based IDS
- Hybrid IDS
- GrIDS (Graph based IDS)

Honeypot Intrusion Detection System (IDS)

Security Honeypots are closely monitored network, serving several purposes: they can more valuable machines on a network, they can act as an early warning system for new attack and exploitation trends, and they allow in-depth examination of adversaries during and after the exploitation of a honeypot[27]. Honeypots are a highly flexible security tool with different application for security. They have multiple uses, such as information gathering, detection and prevention. Honeypots all share the same concept: a security resource that should not have any production or authorized activity. In other words,

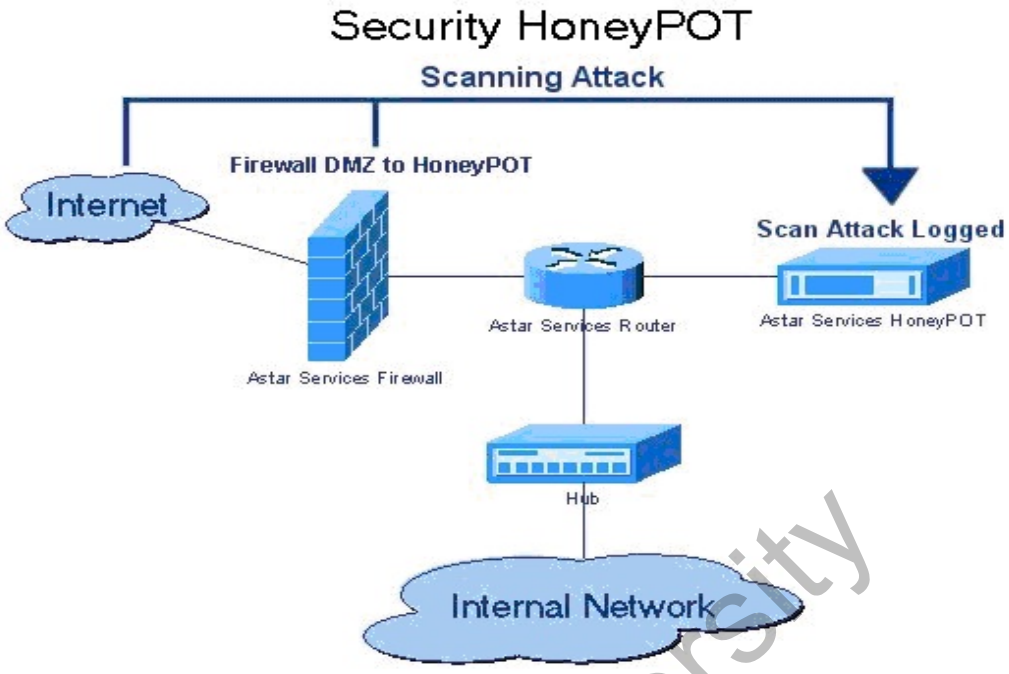


Figure 2.1 Honey Pot [27]

the deployment of honeypots in a network should not affect critical network services and applications. Once a hacker a honeypot , it is more likely that the hacker will stick around for some time. During this time hacker activities can be logged to find out hacker's actions and techniques. Some organizations are concerned with detecting the earliest signs of widespread incidents, such as major new worms, that they deploy deceptive measures such as honeypots so that they can collect better data on these threats. Honeypots are hosts that have no authorized users other than the honeypot administrators because they serve no business function; all activity directed at them is considered suspicious. Attackers will scan and attack honeypots, giving administrators data on new trends and attack tools, particularly malware etc.

Network Based Intrusion Detection System (NIDS)

NIDS are intrusion detection systems that capture data packets traveling on the network media (cables, wireless) and match them to a database of signatures. Depending upon whether a packet is matched with an intruder signature, an alert is generated or the packet is logged to a file or database. Network Based System (NIDS) the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to overlook a firewall's simplistic filtering rules. NIDS are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network[3]. Strengths of Network-Based Intrusion Detection Systems

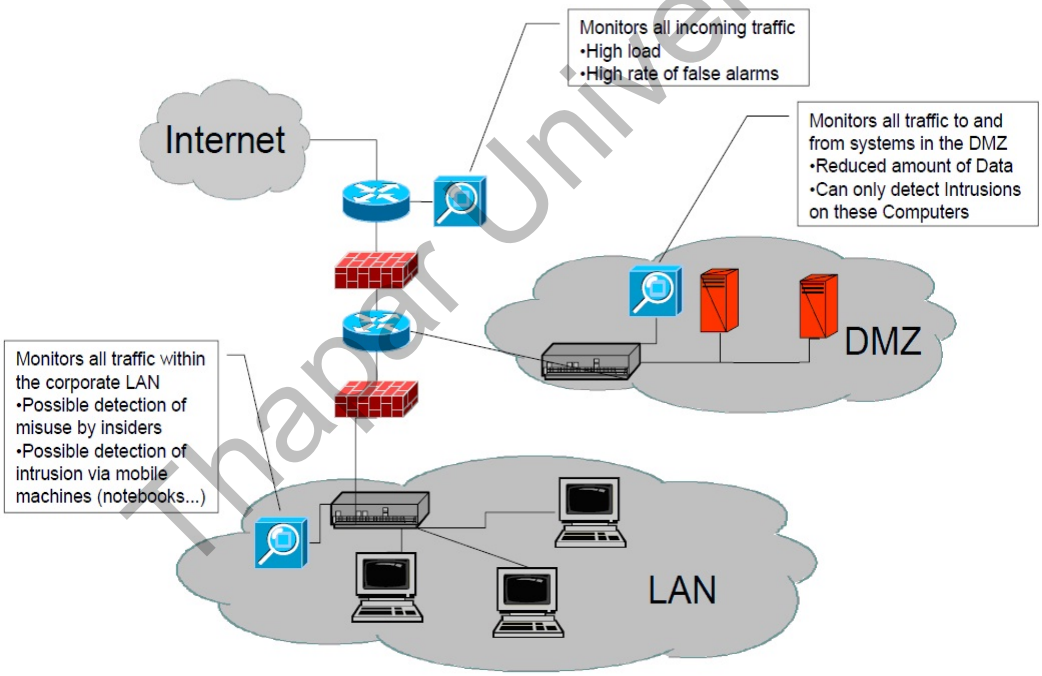


Figure 2.2 Network based IDS [3]

Network-based IDS have much strength that cannot easily be offered by host-based intrusion detection alone. Many customers, in fact, deploy network-based intrusion detection when using IDS for the first time due to its low cost

of ownership and rapid response times. Below are major reasons that make network-based intrusion detection a critical component of sound security policy implementation.

- **Lowers cost of ownership** - network-based IDS allow strategic deployment at critical access points for viewing network traffic destined to multiple systems. As a result, network-based systems do not require software to be loaded and managed on a variety of hosts. Since fewer detection points are required, the cost of ownership is lower for an enterprise environment.
- **Detects attacks that host-based systems miss -** network-based IDS examine all packet headers for signs of malicious and suspicious activity. Host-based IDS do not see packet headers, so they cannot detect these types of attacks. For example, many IP-based denial-of-service (DOS) and fragmented packet (Teardrop) attacks can only be identified by looking at the packet headers as they travel across a network. This type of attack can be quickly identified by a network-based system looking at the packet stream in real-time. Network-based IDS can investigate the content of the payload, looking for commands or syntax used in specific attacks. For example, an attacker probing for the new Back Orifice exploit on systems not yet infected with the Back Orifice software can be detected by examining the packet payload. As above, host-based systems do not see the payload, and not be able to recognize embedded payload attacks.
- **More difficult for an attacker to remove evidence -** network-based IDS use live network traffic for real-time attack detection. Therefore, an attacker cannot remove the evidence. Captured data includes not only

the method of attack, but information that may help lead to identification and prosecution. Since many hackers understand audit logs, they know how to manipulate these files to cover their tracks, frustrating host-based systems that need this information to detect an intrusion.

– **Real-time detection and response** - network-based IDS detect malicious and suspicious attacks as they occur, and so provide faster notification and response. For example, a hacker initiating a network based denial of service (DOS) based on TCP can be stopped by having a network-based IDS send a TCP reset to terminate the attack before it crashes or damages a targeted host. Host-based systems usually do not recognize an attack or take action until after a suspicious log entry has been written. By this time, critical systems may already be compromised, or the system running the host-based IDS may have crashed. Real-time notification allows rapid reaction according to predefined parameters. These responses range from allowing the penetration in surveillance mode in order to gather information to immediate termination of the attack.

– **Detects unsuccessful attacks and malicious intent** - network-based IDS add valuable data for determining malicious intent. Network-based IDS placed outside of a firewall can detect attacks intended for resources behind the firewall, even though the firewall may be rejecting these attempts. Host-based systems do not see rejected attacks that never hit a host inside the firewall. This lost information can be critical in evaluating and refining security policies.

– **Operating system independence** - network-based IDS are not dependent on host operating systems as detection sources. By way of comparison, host-based systems require specific operating systems to function

properly without having been compromised to generate meaningful results.

Host Based Intrusion Detection System Host Based Systems (HIDS) the IDS examines all the activity on each individual computer or host. HIDS consists of an agent on a host which identifies intrusions by analyzing system calls, application logs, file-system modifications and other host activities. Host-based intrusion detection systems or HIDS are installed as agents on a host[3]. These intrusion detection systems can look into system and applica-

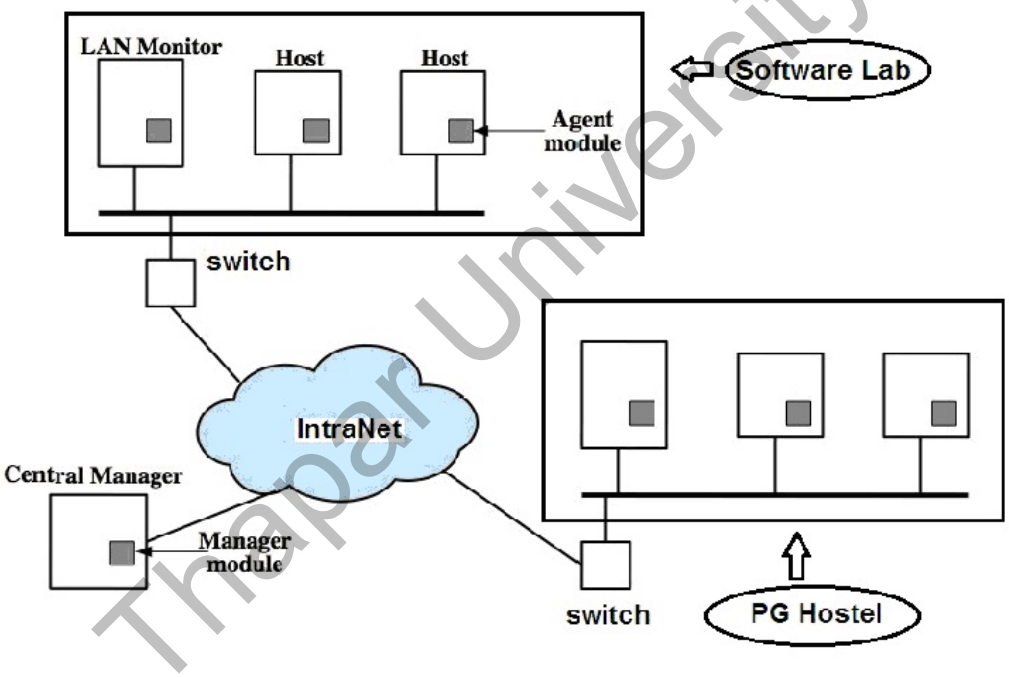


Figure 2.3 Host based IDS [3]

tion log files to detect any intruder activity. Some of these systems are reactive, meaning that they inform you only when something has happened. Some HIDS are proactive; they can sniff the network traffic coming to a particular host on which the HIDS is installed and alert you in real time. Strengths of Host-Based Intrusion Detection Systems While host-based intrusion detection

systems are not as fast as their network counterparts, they do offer advantages that the network-based systems cannot match. These strengths include stronger forensic analysis, a close focus on host-specific event data and lower entry-level costs. Host-based intrusion detection[9]:

- **Verifies success or failure of an attack** - Since host-based IDS use logs containing events that have actually occurred they can measure whether an attack was successful or not with greater accuracy and fewer false positives can network-based systems. In this respect, host based IDS make an excellent complement to network-based intrusion detection, with the network component providing early warning and the host component providing verification of whether an attack was successful or not.
- **Monitors specific system activities** - host-based IDS monitor user and file access activity, including file accesses, changes to file permissions, attempts to install new executables and/or attempts to access privileged services. For example, host-based IDS can monitor all user logon and logoff activity, as well as what each user does while connected to the network. It is very difficult for a network-based system to provide this level of event detail. Host-based technology can also monitor activities that are normally executed only by an administrator. Operating systems log any event where user accounts are added, deleted, or modified. The host-based IDS can detect an improper change as soon as it is executed. Host-based IDS can also audit policy changes that affect what systems track in their logs. Finally, host-based systems can monitor changes to key system files and executables. Attempts to overwrite vital system files, or to install trojan horses or backdoors, can be detected and stopped. Network-based systems sometimes miss this kind of activity.

– **Detects attacks that network-based systems miss** - Host-based systems can detect attacks hat cannot be seen by network-based products. For example, attacks from the keyboard of critical server do not cross the network, and so cannot be seen by a network-based intrusion detection system.

– **Well-suited for encrypted and switched environments** - Since host-based systems reside on various hosts throughout an enterprise, they can overcome some of the deployment challenges faced by network-based intrusion detection in switched and encrypted environments. Switches allow large networks to be managed as many smaller network segments. As a result, it can be difficult to identify the best locations for deploying network-based IDS to achieve sufficient network coverage. Traffic mirroring and administrative ports on switches can help, but these techniques are not always appropriate. Host-based intrusion detection provides greater visibility in a switched environment by residing on as many critical hosts as needed. Certain types of encryption also present challenges to network-based intrusion detection. Depending where the encryption resides within the protocol stack, it may leave a network based system blind to certain attacks. Host-based IDS do not have this limitation. By the time an operating system, and therefore the host-based system, sees incoming traffic, the data stream has already been de-encrypted.

– **Near-real-time detection and response** - Although host-based intrusion detection does not offer true real-time response, it can come extremely close if implemented correctly. Unlike older systems, which use a process to check the status and content of log files at predefined intervals, many current host-based systems receive an interrupt from the operating

system when there is a new log file entry. This new entry can be processed immediately, significantly reducing the time between attack recognition and response. There remains a delay between when the operating system records the event and the host-based system recognizes it, but in many cases an intruder can be detected and stopped before damage is done.

- **Requires no additional hardware** - Host-based intrusion detection resides on existing network infrastructure, including file servers, Web servers, and other shared resources. This efficiency can make host-based systems very cost effective because they do not require another box on the network that requires addressing, maintenance, and management.
- **Lower cost of entry -** While network-based intrusion detection systems can offer wide coverage for little effort, they are often expensive. Deploying a single intrusion detection system can cost more than $10,000$. Host-based intrusion detection systems, on the other hand, are often priced in the hundreds of dollars for a single agent and can be deployed by a customer with limited initial capital outlay.

Both network- and host-based IDS solutions have unique strengths and benefits that complement each other. Next-generation IDS, therefore, must include tightly integrated host and network components.

Combining these two technologies will greatly improve network resistance to attacks and misuse, enhance the enforcement of security policy and introduce greater flexibility in deployment options. The graphic below illustrates how network- and host-based intrusion detection techniques interact to create a more powerful network defense. Some events are detectable by network means only. Others those are detectable only at the host. Several require both types

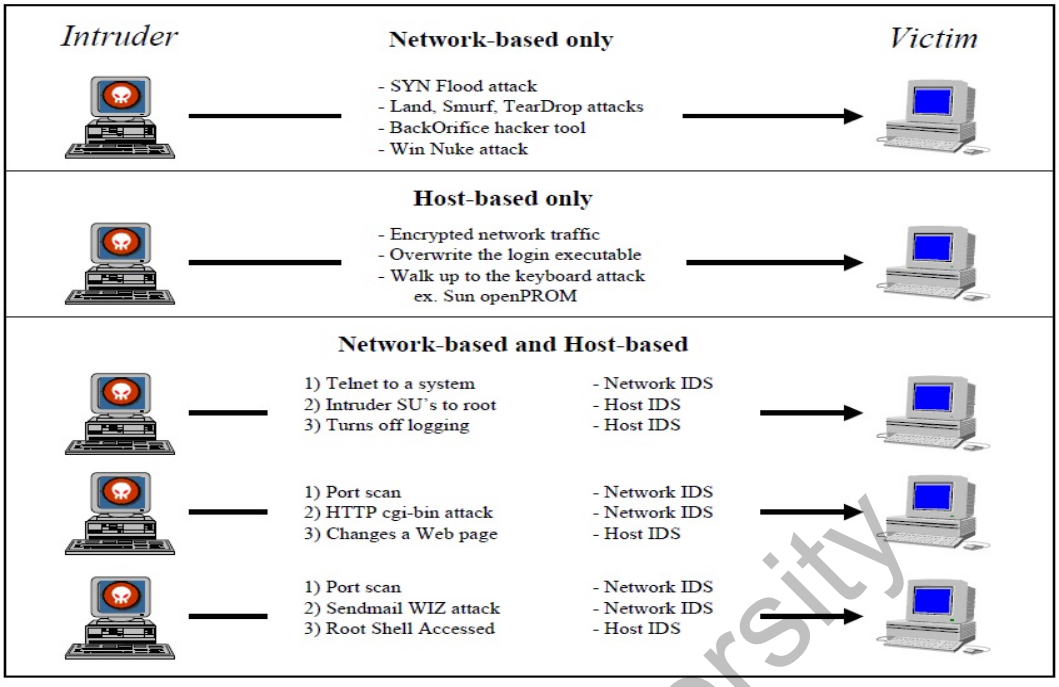


Figure 2.4 Need of HIDS and NIDS

of intrusion detection to function properly.

Hybrid Intrusion Detection System Hybrid Intrusion Detection System means that both the combination of network based (NIDS) and host based intrusion system (HIDS)[22]. other host activities. Host-based intrusion detection systems or HIDS are installed as agents on a host. They are based on the model which brings multiple agents of multiple types such as, Simple Network Message Protocol (SNMP), packet sniffer, and filtering software or tool. The purpose of this technology is to bring more flexibility, expandability and most important to crosscheck anomalies against other system accuracy of alerts and reduce False positive. Both NIDS and HIDS are each patrolling its own area of the network for unwanted and illegal network traffic. They, however, complement each other. Both bring to the security of the network their own strengths and weaknesses that nicely complement and augment the security of the network. Hybrids are new and need a great deal of support to

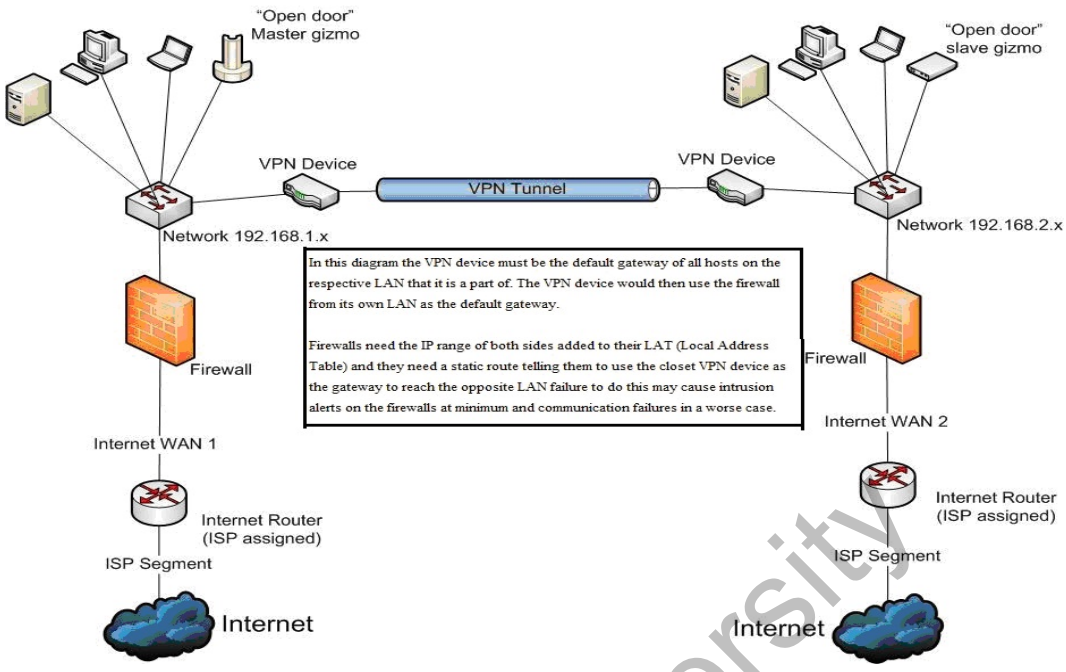


Figure 2.5 Hybrid IDS [22]

gain on their two cousins. However, their success will depend to a great extent on how well the interface receives and distributes the incidents and integrates the reporting structure between the different types of sensors in the HIDS and NIDS spheres. Also the interface should be able to smartly and intelligently gather and report data from the network or systems being monitored.

Graph based Intrusion Detection System (GrIDS)

Graph based intrusion detection system (GrIDS) is built to detect distributed attacks into a large network. It constructs an activity graph to represent connections between hosts (domain) and every edge in the graph represents the network traffic between the nodes. GrIDS uses network sniffer and also host based intrusion detection systems as its data source. Collected data will be the input of the graph engine, which is responsible for creating activity

graphs. A well defined syntax for reporting to GrIDS is available to GrIDS users who wish to write their own filters. The GrIDS mechanisms make no specific assumptions about the nature of attributes, and GrIDS is able to import externally written correlation functions[26].

2.1.3 Technologies based classification

Intrusion can be detected by several techniques. Some of these are:

Anomaly detection

Anomaly based detection approach are those that can be used to detect attacks that falls within a certain environment or certain state. The idea behind this approach is to measure a baseline of such states as CPU utilization, disk activity, user logins, and file activity. Then, the IDS can trigger when there is a deviation from this baseline. The benefit of this approach is that it can detect the anomalies without understanding the underlying cause behind the anomalies. An automatically developed profile is created by software that collects and processes characteristics of system behavior over time and forms a statistically valid sample of such behavior. Note that an unexpected behavior is not necessarily an attack; it may represent new, legitimate behavior that needs to be added to the category of expected behavior. Events in an anomaly detection engine are caused by any behaviors that fall outside the predefined or accepted model of behavior.

A disadvantage of anomaly-detection engines is its complexity to define rules. Each protocol being analyzed must be defined, implemented and tested for accuracy. The rule development process is also compounded by differences

in implementations of the various protocols. Custom protocols traversing the network cannot be analyzed without great effort. On the other hand, once a protocol has been built and a behavior defined, the engine can scale more quickly and easily than the signature-based model because a new signature does not have to be created for every attack and potential variant. Another pitfall of anomaly detection is that malicious activity that falls within normal usage patterns is not detected.

Signature detection

Signature or Misuse detection systems essentially contain attack descriptions or signatures and match them against the audit database or data stream, looking for evidence of known attacks. A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to IDS. The main advantage of signature detection systems is that they focus analysis on the audit data and typically produce few false positives. The main disadvantage of signature detection systems is that they can detect only known attacks for which they have a defined signature. As new attacks are discovered, developers must model and add them to the signature database. The block diagram shows that the signature based NIDS[26].

In these the whole initial information about the IDS given, so IDS can be used into the network for protection from unwanted intruders and viruses.

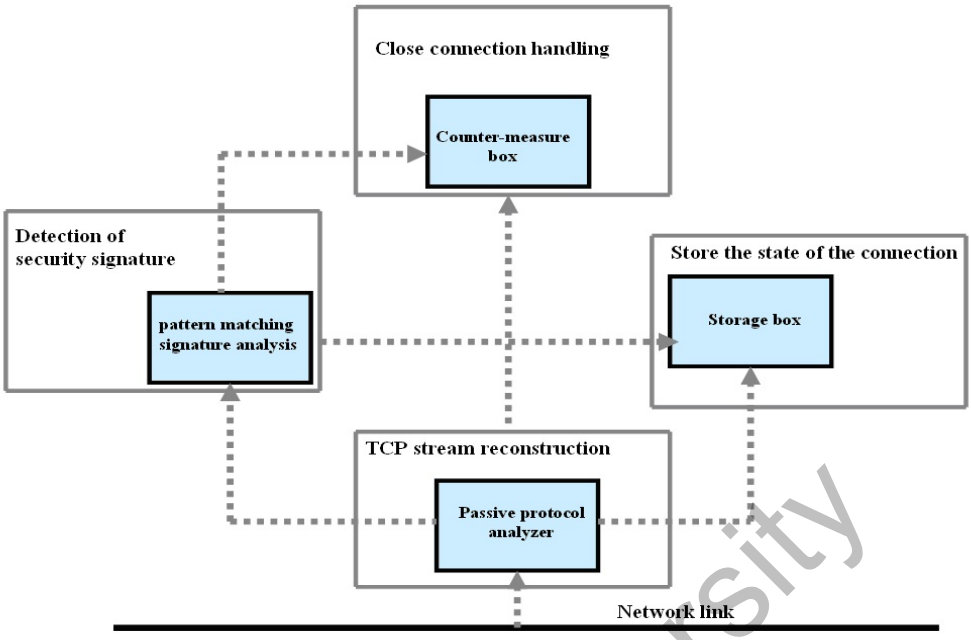


Figure 2.6 Block schematic of the components in a signature based NIDS [26]

The several IDS systems are used to prevent security and important privileges or accessibility into a particular network. So all the IDS systems used several techniques and routing algorithm for implementation and fixed the intrusion or vulnerabilities. So important here to know about these techniques or routing algorithms.

2.1.4 Routing Algorithms

Routing is at the core of any network control system, strongly affecting the overall network performance. Routing can be characterized in the following general way. Let the network be represented in terms of a directed weighted graph G = (V,E), where each node in the set V represents a processing and forwarding unit and each edge in E is a transmission system with some capac-

ity/bandwidth and propagation characteristics. Data traffic originates from one node (endpoint) and can be directed to another node (unicast traffic), to a set of other nodes (multicast traffic) and/or to all the other nodes (broadcast traffic). The node from where the traffic flow originates is also called source, or starting end-point, while the nodes to which traffic is directed are the final end-points, or destinations. The nodes in between that forward traffic from sources to destinations are called intermediate, or relay, nodes. The general routing problem is the problem of defining path flows to forward incoming data traffic such that the overall network performance is maximized. At each node data is forwarded according to a decision policy parameterized by a local data structure called routing table. In this sense, a routing system can be properly seen as a distributed decision system. The routing problem is composed of two parts: (i) the communication structure, which in a sense defines the constraints, and (ii) the traffic patterns that make use of this structure[2].

Clearly, different strategies implement each of the operations of the algorithm in a possibly different way according to the specificities of the problem and of the design choices. At each network node:

- Acquisition and organization of up-to-date information concerning the local state, that is, information on the local traffic flows and on the status of the locally available resources.
- Build up a view of the global network state, possibly by some form of exchanging of the local state information.
- Use of the global view to set up the values of the local routing table and, consequently, to define the local routing policy with the perspective of

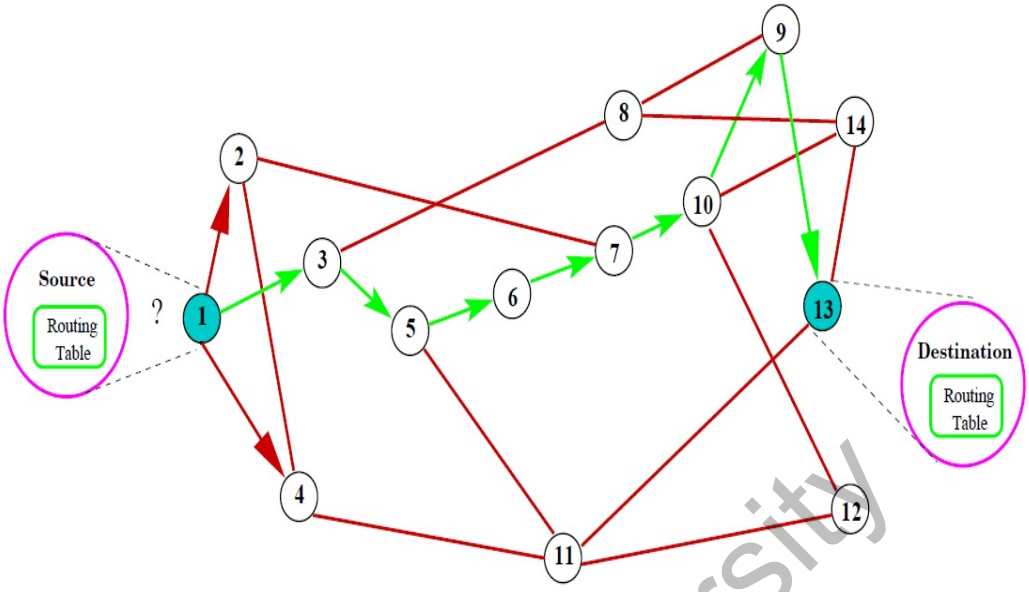


Figure 2.7 Routing in a Network [2]

optimizing some measure of network performance.

- Forward of the user traffic according to the defined routing policy.
- Asynchronously and concurrently with the other nodes repeat the previous activities over time.

Classification of Routing Algorithms

Routing algorithms are usually designed in relationship to the type of both the network and the services delivered by the network. There are several types of routing algorithms classified according to few characteristics[2].

- Control Architecture Level
- Routing Tables based
- Load distribution level

– Optimization level based

Control Architecture Level: centralized vs. distributed

In centralized algorithms, a main controller is responsible for the updating of all the node routing tables and/or for every routing decision. Centralized algorithms can be used only in particular cases and for small networks. In general, the controller has to gather information about the global network status and has to transmit all the decisions/updates.

In distributed routing systems, every node autonomously decides about local data forwarding. At each node a local routing table is maintained in order to implement the local routing policy. The distributed paradigm is currently used in the majority of network systems.

Routing Tables based: static vs. dynamic

Routing tables can be statically assigned or dynamically built and updated. In both cases routing tables are built in order to possibly optimize some network-wide criteria which are made depending in turn on costs associated to network elements. In static routing systems, the path to forward traffic between pairs of nodes is determined without regard to the current network state. The paths are usually chosen as the result of the offline optimization of some selected costs. Once defined the paths to be used for each source-destination pair, data are always forwarded along these paths. Routing tables can be also assigned on the basis of some a priori knowledge about the expected input traffic. For instance, traffic statistics can be periodically recorded, and if some regularity can be spot, these can be used in turn to model the incoming traf-

fic and assign the routing tables as the result of optimal routing calculations. Dynamic (or adaptive) routing goes beyond static routing by admitting the possibility of building / changing the routing tables online according to the current traffic events. It is useful to distinguish between the ability of adapting to the changing traffic conditions and to topological modifications. The problems with adaptive routing are well captured by the following sentence, slightly changed from the original situation; Link arrival rates depend on routing, which in turn depends on arrival rates via routing selected paths, with a feedback effect resulting.

Load distribution level: single vs. multiple paths

Data traffic toward the same destination d can be forwarded along always the same link or it can be spread along multiple paths. when routing tables are updated being adaptive to traffic patterns, the resulting effect can be that of actually spreading the data packets toward the same destination over multiple paths at the same time, if the updating interval is shorter than or comparable to the inter-arrival time of the packets directed to destination d With multipath routing is the situation in which multiple next hop entries for the same destination are maintained in the routing table and used to forward data according to some scheme. On the other hand, alternate routing is the situation in which information about multiple paths is maintained in the routing table. Multipath routing strategies can provide optimized utilization of network resources utilization, automatic load balancing,

increased robustness to failures, easier recovery from sudden congestion, and better throughput and end-to-end delay performance under heavy traffic loads. On the other hand, algorithm design is in general more complex and in par-

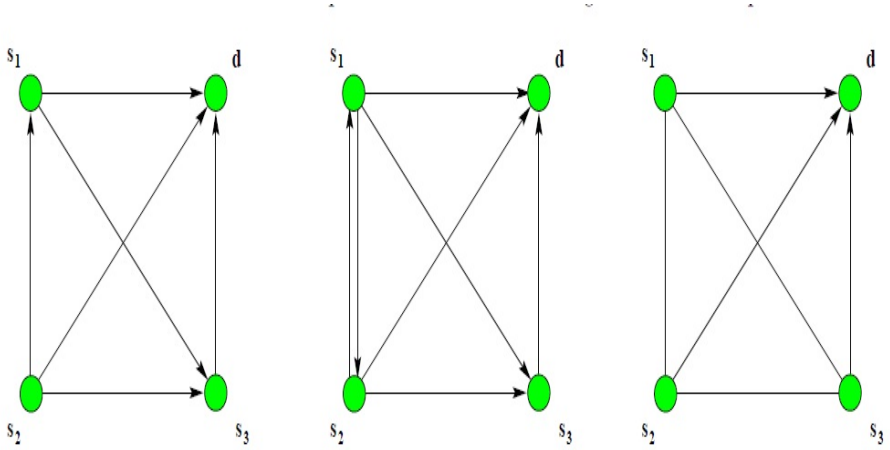


Figure 2.8 Example of multipath routing from sources si, i = 1, 2, 3 to destination d [2].

ticular loops can easily result from piecewise combinations of paths during transient phases, such that they have to be either avoided or guaranteed to be short-lived. The directed links show the possible routing decisions that are available at nodes for a packet bound for d according to their routing tables. The links are assumed to have all the same unit cost. The leftmost graph shows a routing policy which is globally loop-free independently from the specific policy adopted to locally spread the data along the different links. That is, the combination of the routing policies of all the nodes defines a directed acyclic graph rooted in d. The middle graph shows an assignment of the routing tables which can give rise to packet looping between s1 and s2, depending on the specific utilization of the local multiple alternatives as a function, for instance, of the distance to the destination. If the distances/costs are calculated in a wrong way, possibly because of traffic fluctuations, is easy to incur in packet looping in this case. The rightmost graph shows the assignment of the routing tables resulting from a single-path shortest path calculation. Multipath routing strategies can provide optimized utilization of network resources

utilization, automatic load balancing, and increased robustness to failures, easier recovery from sudden congestion, and better throughput and end-to-end delay performance under heavy traffic loads. On the other hand, algorithm design is in general more complex and in particular loops can easily result from piecewise combinations of paths during transient phases, such that they have to be either avoided or guaranteed to be short-lived[2][16].

In shortest path algorithms, at each node s, the local link which is on the minimum cost path to the destination d, for all the possible destinations d in the network is identified and used to forward the data traffic directed to d. The minimum cost path is calculated without taking into account the paths for the other destinations. That is, the path for each destination is treated as an entity independent from the paths (i.e., the paths flows) for all the other destinations. This is in contrast with the optimal routing approach that allocates each flow minimizing a joint function of all the flows in the network.

Optimization level based: optimal vs. shortest paths

Shortest path routing is the routing paradigm most in use in real networks. In shortest path routing the optimizing strategy for path flows consists in using the minimum cost paths connecting all the node pairs in the network, where the paths are calculated independently for each pair. That is, shortest path routing adopts a per pair perspective. Optimal routing which is the other main reference paradigm (at least from a theoretical point of view), has a network-wide perspective, since the path flows are calculated considering all the incoming traffic sessions. Clearly, in order to adopt such a global strategy, optimal routing requires the prior knowledge of the statistical characteristics of all the incoming flows, a requirement which is usually quite hard to satisfy.

Both optimal and pure shortest path routing implement minimal routers[16]. At each network node: General behavior of shortest path routing algorithms

1. Assign a cost to each one of the out links. The cost can be either static or adaptive; in the following it is assumed the most general case of adaptive link cos
2. Periodically, and without the need for inter-node synchronization, transmit to the neighbors either estimates about cost and status (on/off) of the attached links, or some other information related to the estimated distance/delay from the node to the other known nodes in the network.
3. Upon receiving fresh information from a neighbor, update the local routing table and local information database (i.e., the local view of the global network status). The routing tables are updated in order to associate to each destination the out link that satisfies the conditions of minimum cost path. That is, for each network destination d, the out link belonging to the minimum cost path to reach d will be used to route data traffic bounded for d. The computation of the minimum cost paths is executed on the basis of the locally available information only.
4. The received information packet, or the updated routing information, can be in turn also forwarded to the neighbors, which might further forward it.
5. Asynchronously and concurrently with the other nodes repeat the previous activities over time. According to the network routing point of view there is main type of routing algorithm shortest path routing; it has also sub divide into several types of algorithms, some these are:
   - Distance-vector algorithms

- Link-state algorithms
- Dijkstra's Algorithm

Distance-vector algorithms

In distance-vector algorithms, each node n maintains a matrix (i) of distance estimates for each possible network destination d and for each possible choice of next node i, where i (n), the set of neighbor nodes of n. These distance estimates are used to build up the vector S of the shortest distances to d, which, in turn, is used to implement routing decisions. The stored topological information is represented by the list of the known nodes identifiers. The average memory occupation per node is of order O(Nn), where N is the number of nodes in the network and n is the average connectivity degree (i.e., the average number of neighbor nodes considered over all the nodes). Distance-vector algorithms forward a packet with destination d along the local link belonging to the path associated with the shortest estimated distance S to d. Central to the whole scheme is the matrix D of the distances, containing the estimated distance (cost) to reach each possible destination in the network for each one of the local out links. The algorithm maintains also a vector of distances to each one of the neighbors. A one-step Bellman equation is used to build the vector SD of the shortest distances to each possible destination d. This vector is used in turn to implement the routing policy[2][16][15]. The central component of the algorithm is the distributed computation of such minimum cost paths using the locally available topological description of the network, the costs to go received from the neighbors, and the local distance to the neighbors. The equation in order to compute/estimate the traveling distance to each one of the possible destinations d in the network for each

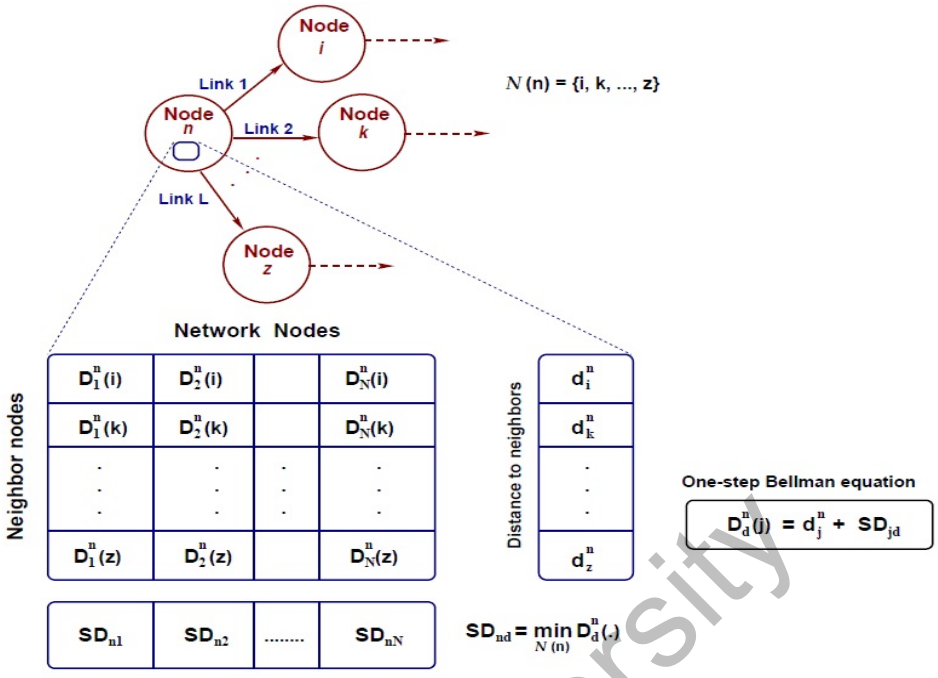


Figure 2.9 Data structures and basic equations used in distance-vector algorithms [2]

one of the local next hops i: $D_d^n(i) = D_d^n + SD_{id}$ Once the entries of the matrix D are set up, the vector SD of the shortest distances from n is set up accordingly: $SD_{nd} \models \min_{j \epsilon N(n)} [d_n^j + SD_{jd}]$ The routing table is defined at the same time as the vector SD: for each destination d the chosen next hop node is the one minimizing the expression used to compute SD. Each node n, in order to compute the matrix of the estimates D, in addition to the locally estimated value needs to know the values from all its neighbors . This is the critical part of the distributed algorithm. At the beginning of the operations, the matrix D and the vector SD are initialized all over the network nodes with the same arbitrary values. Then, at each node n, when either the local cost estimates are updated, or an updated value of SD is received from one of the neighbors are re-computed, the routing table is updated, and the possibly new value of is sent, in turn, to all its neighbors. Iterating this distributed asynchronous

behavior over the time, after a transitory phase, the distance estimations at each node converge to the correct minimum values with respect to the used cost metric.

Link-state algorithm

Link-state algorithms make use of routing tables containing much more information than that used in distance-vector algorithms; at the core of link-state algorithms there is a distributed and replicated database. This database is essentially a topological map, usually built in a dynamic way, of the whole network. The map contains comprehensive information concerning the link-state of the links of all the network nodes, that is, information about their cost value, node endpoints, and operational status. Each node on the same hierarchical level maintains a complete description of the link states (costs, interconnections, operational status) of all the network nodes that belong to the same hierarchical level. In order to avoid this local database to grow up linearly with the size of the whole network, as well as for other general administrative and logistic reasons, networks usually have a hierarchical organization. Therefore, the link-state maps need to contain only information related to the network nodes on the same hierarchical level. Hierarchical organization is an important aspect of the whole network management. For example, in the Internet, nodes are organized in hierarchical Autonomous Systems and multiple routing areas inside each Autonomous System. Roughly speaking, sub-networks are seen as single host nodes connected to interface nodes called gateways. Gateways perform fairly sophisticated network layer tasks, including routing. Groups of gateways, connected by topology, define logical areas. Inside each area, all the gateways are at the same hierarchical level and flat

routing is performed among them. Areas communicate only by means of area border gateways. In this way, the computational complexity of the routing problem itself, as seen by each gateway, is much reduced (i.e. areas in the internet typically group from 10 to 300 gateways), while the complexity of the design and management of the routing protocol is much increased. These considerations are in general not to limited to link-state algorithms, but are valid for any type of routing scheme[2][15].

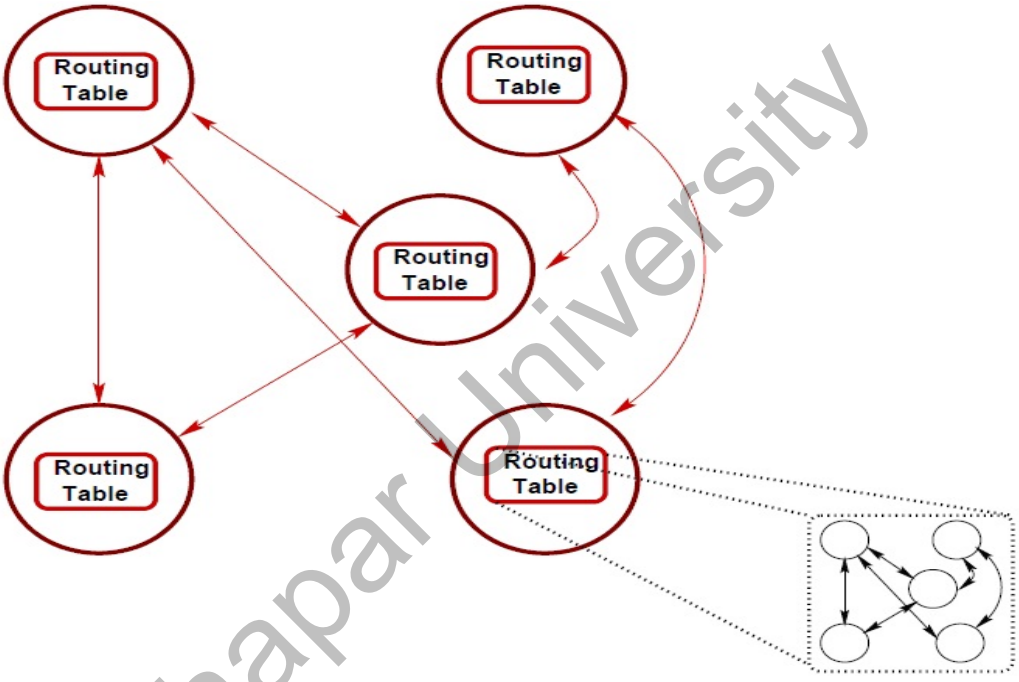


Figure 2.10 Linkstate Routing [2]

Distance-vector algorithms carry out the computation in a distributed way involving all the network nodes, while in link-state algorithms every single node carries out an independent centralized computation of the shortest paths. Therefore, link-state algorithms require locally both more computational power and more memory space, but can provide also more robust routing. Complete topological description of the whole network is available, link-state schemes can be used for source routing. When a packet is generated,

the whole path from the source to the destination node can be computed at the source. Therefore, this information can be added to the packet data, such that the packet becomes self-contained: it carries both the data and all the necessary routing directives. Most critical activity in link-state algorithms is that of the link-state advertising (LSA). Each node acts autonomously and monitors the state of the local links. According to significant changes, it must somehow broadcast ("advertise") the information concerning the link state information to all the other nodes on the same hierarchical level. On receiving the LSA, each neighbor will process the associated information, update the corresponding link states in its topological map, re-compute all the shortest paths and, in turn, will forward the received LSA packet to its neighbors. The LSA packets throughout the network in order to minimize the number of multiple transmissions of the same packet to the same node, while, at the same time, to ensure the quick propagation of the packet to all the network nodes. The link-vector algorithm has been proposed to overcome the significant overhead due to topology broadcast in typical link-state algorithms by using link-states in conjunction with distance-vector style of information propagation. Each router updates its neighbors with the state of the links it uses to reach a destination and also informs them of the links that it stops using to reach destinations. The LSA packets are then propagated incrementally in the same way distance information propagates in the distance-vector case.

Dijkstra's Algorithm

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest

path tree. This algorithm is often used in routing. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. Let G= V, E be a directed weighted graph with V having the set of vertices. The special vertex s in V, where s is the source and let for any edge e in E, Edge Cost(e) be the length of edge e. All the weights in the graph should be non-negative. Before going in depth about Dijkstra's algorithm let's talk in detail about directed-weighted graph. Directed graph can be defined as an ordered pair G: = (V,E) with V is a set, whose elements are called vertices or nodes and E is a set of ordered pairs of vertices, called directed edges, arcs, or arrows. Directed graphs are also known as digraph[2]. Directed-weighted graph is a directed graph with weight attached to each of

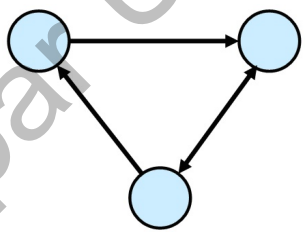


Figure 2.11 Dijakstra Digraph [2]

the edge of the graph. So there are several routing algorithms, discussed that can describe the shortest path finding in a network. Let discuss about the most popular routing algorithm which is used mostly these days known as Ant Colony Optimization Algorithm.

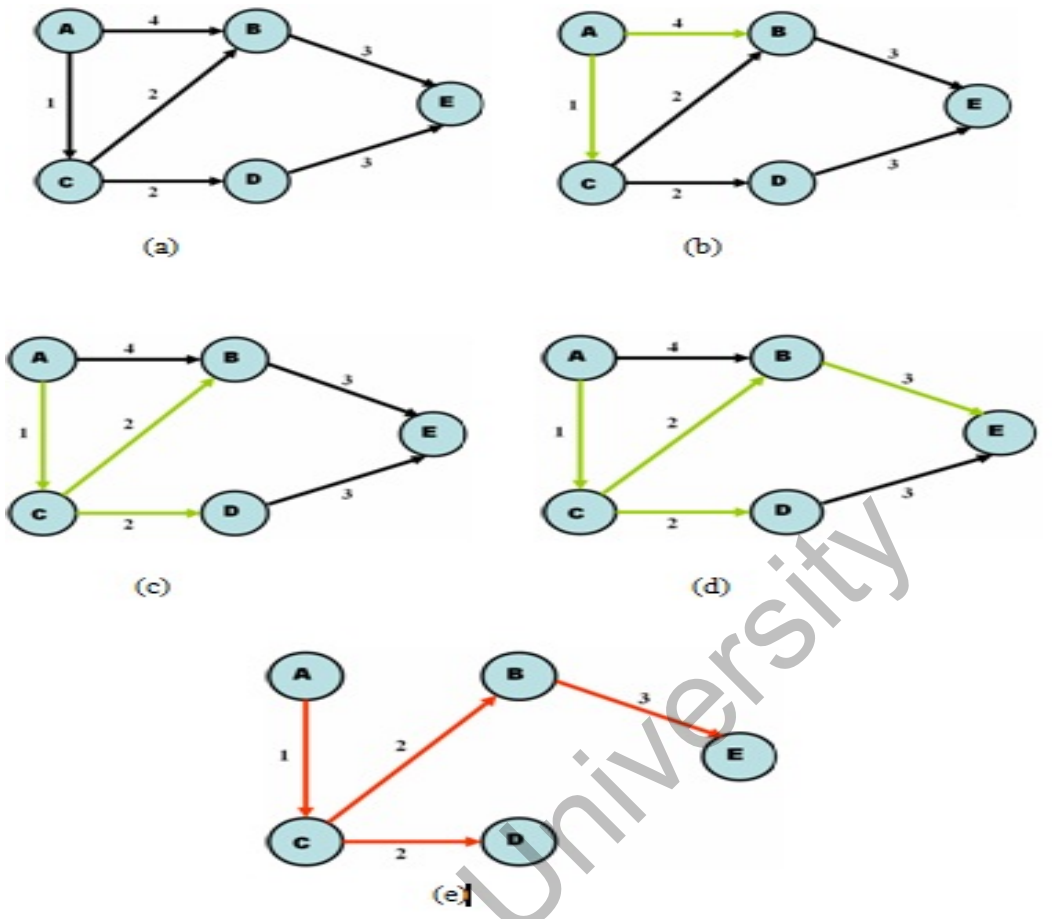


Figure 2.12 Weighted graph [2]

2.2 Ant Colony Optimization

The ACO algorithm was first introduced by Colorni et al. (1991) and the first ant system (AS) was proposed by Dorigo (1992) in his Ph.D. thesis. Ant Colony Optimization (ACO) is a family of optimization algorithms based on real ants' behavior. ACO is inspired by the foraging behavior of ant colonies, where in they are able to find shortest path to food source. It has been observed that of available routes, ants find shortest route to food source. In real life, ants deposit some kind of chemical substances to mark the path that they used. Then on their way back they choose the path with the highest

pheromones which becomes the shortest path. Real ants have the following key characteristics[2][26].

- Real ants prefer to choose the trail with a higher intensity of pheromone.
- The shorter the distance of a path, the more the pheromone left on it.
- Real ants conduct indirect communication via pheromone.

The above behavior of self-organizing real ants for finding the shortest path fostered the development of the ACO. Artificial ants cooperate to come up with the solution by exchanging information via depositing pheromone on paths. Four main algorithms for routing in telecommunication networks (AntNet, AntNet-FA, AntNet+SELA and AntHocNet), a general framework for the design of routing algorithms (Ant Colony Routing) ant-inspired routing algorithms.

2.2.1 Main ACO Algorithms

Ant System.

Ant Colony System (ACS).

MAX-MIN Ant System (MMAS).

Ant system (AS) was the first ACO algorithm to be proposed in the literature. Its main characteristic is that the pheromone values are updated by all the ants that have completed the tour. Ant Colony System (ACS) differs from Ant System (AS) in three main aspects:

- The state transition rule provides a direct way to achieve balance between exploration of new edges and exploitation of a prior and accumulated knowledge on the problem.

- The global updating rule is applied only to edges which belong to the best tour.
- While ants construct a solution, a local pheromone updating rule (local updating rule, for short) is applied to shuffle the tours. In this way ants will make better use of pheromone information. These three modifications enable ACS to solve large problems effectively. With about 160 nodes for ants to visit, ACS algorithm has been chosen to deal with this optimization problem.

MIN-MAX Ant System (MMAS) differs from AS in that (I) only the best ant adds pheromone trails, and (II) the minimum and maximum values of the pheromone are explicitly limited (in AS and ACS these values are limited implicitly, that is, the value of the limits is a result of the algorithm working rather than a value set explicitly by the algorithm designer).It improves the original algorithm by providing dynamically evolving bounds on the pheromone trails such that the heuristic is always within a limit comparing with that of the best path.

2.3 AntNET

AntNet is a routing protocol for packet switched networks, invented by M. Dorigo and G.Di Caro. It is an alternative routing algorithm for the well-known OSPF (Open Shortest Path First) protocol, based on Ant Colony Optimization (ACO). Open Shortest Path First, a distance-vector routing protocol, based on the Dijkstra algorithm. ACO studies the behaviour of ants in a colony and mimics this behavior in software. The problem to be solved, is represented by a graph. Artificial agents, i.e. software ants, gradually construct

paths in this graph. This phase is repeated until an optimal (or in some cases a sub-optimal) solution is found. ACO has been applied to many domains, e.g. the Traveling Salesmen Problem, manufacturing control systems, etc. ACO itself is a metaheuristic. When combined with an actual problem area, it can lead to several heuristics. AntNet is a result of the application of ACO on the problem of Internet routing. Intelligent agents, ants for short, are sent over the network. They communicate indirectly by information they leave behind in the routers on their path. Over time, this information leads to optimal routing paths between the routers in the network. The operation of AntNet is based on two types of agents:

- Forward Ants who gather information about the state of the network, and
- Backward Ants who use the collected information to adapt the routing tables of routers on their path.

An AntNet router contains a special routing table where each destination is associated to all interfaces and each interface has a certain probability. This probability indicates whether or not it is interesting to follow that link in the current circumstances. The router also contains a statistical model to store the mean and variance values of the trip times to all destinations in the routing table. These are used as reference values. On a regular time base, every router sends a Forward Ant with random destination over the network. The task of the Forward Ants is collecting information about the state of the network. In each router they pass, the elapsed time since the start is stored on an internal stack together with the identifier of the router. Then the next hop is determined. Normally this is based on the probabilities in the routing table.

There is however a small chance (exploration probability) that the next hop is randomly chosen. This is necessary to constantly explore the network and to be able to react fast to network changes like link failures or congestion.

When the Forward Ant reaches its final destination again the elapsed time since the start and the identifier of the router are stored on the stack of the ant. The Forward Ant is transformed into a Backward Ant. This Backward Ant will follow exactly the same path as the Forward Ant but in the opposite direction. The Backward Ants use the information collected by the Forward Ants to update the different data structures in each router along their path. The time information on the stack is compared with the model in the router and based on this comparison, the probabilities in the routing table are updated. When the Backward Ant arrives in the start router, it dies. Backward Ants have a higher priority than data packets, so that they are processed as fast as possible making the algorithm more adaptive. Forward Ants have the same priority as data packets, to suffer the same delays so that the algorithm can react to network congestion

AntNet and AntNet-FA are two ACO algorithms for adaptive best-effort routing in wired datagram networks. Ant Colony Routing (ACR) is a general framework of reference for the design of autonomic routing systems. The aim of providing a meta-architecture of reference for the design and implementation of fully adaptive and distributed routing systems for a wide range of network scenarios (e.g., wired and wireless, best-effort and QoS, static and mobile). In the same way ACO has been define as an ant-inspired meta-heuristic for generic combinatorial problems. Both AntNet and AntNet-FA can be seen as specific instances of ACR for the case of best-effort routing in wired datagram networks. ACR,s ideas can find their application, as well as in order to

introduce a new routing algorithm for each one of the most important and popular network scenarios, we briefly describe two additional routing algorithms, AntNet+SELA and AntHocNet. *AntNetSELA* is a model to deliver QoS routing in ATM networks, while AntHocNet is intended for routing in mobile ad hoc networks. AntNet was specifically designed to address the problem of adaptive best-effort routing in wired datagram networks (*e.g.Internet*). AntNet-FA has brought some major improvements into the AntNet design and made it also suitable for a possible use in connection-oriented and QoS networks. Ants are mobile agents that migrate from one node to an adjacent one searching for feasible paths between source and destination nodes. ACO's solution components correspond to network nodes, and, accordingly, routing tables correspond to pheromone tables in which each pheromone variable holds the estimated goodness of selecting k's neighbor n to forward a data packet toward d. The immediate relationship between ACO and network routing is likely one of the main reasons behind the popularity of the application of ACO to routing problems, as well as behind the usually good performance and the strong similarities showed by the different implementations[1][2].

In particular, the adopted pheromone model is in practice the same for all the implementations: a pheromone variable is always associated to a pair of nodes, which are the "natural" solution components for routing problems. The relationship between ACO (as well as its biological context of inspiration) and networks is particularly evident for the case of datagram protocols. In fact, in this case each node builds and holds its own routing table and an independent routing decision is taken for each single data packet on the sole basis of the contents of the local routing table. The direct relationship between ACO and datagram models is likely one of the main reasons behind the fact that most

of the works on ant-based routing have focused so far on best-effort datagram networks

- provide traffic-adaptive and multipath routing,
- rely on both passive and active information monitoring and gathering,
- set up paths in a less selfish way than in pure shortest path schemes favoring load balancing,
- Show limited sensitivity to parameter settings.

2.4 AntNet: Traffic-adaptive multipath routing for best-effort IP networks

AntNet is an ACO algorithm for distributed and traffic-adaptive multipath routing in wired best effort IP networks. AntNet's design is based on ACO's general ideas, which was a first application of algorithms inspired by the foraging behavior of ant colonies to routing tasks (in telephone networks). AntNet behavior is based on the use of mobile agents, the ACO's ants that realize a pheromone-driven Monte Carlo sampling and updating of the paths connecting sources and destination nodes. A detailed description and discussion of all AntNet's components is provided[2][16]:

- From each network node s mobile agents are launched towards specific destination nodes d at regular intervals and concurrently with the data traffic. The agent generation processes happen concurrently and without any form of synchronization among the nodes. These agents moving from their source to destination nodes are called forward ants and are indicated

with where i is the ant identifier.

- Each forward ant is a random experiment aimed at collecting and gathering at the nodes non-local information about paths and traffic patterns. Forward ants simulate data packets moving hop-by-hop towards their destination. They make use of the same priority queues used by data packets. The characteristics of each experiment can be tuned by assigning different values to the agent's parameters (e.g., the destination node).
- Ants, once generated, are fully autonomous agents. They act concurrently, independently and asynchronously. They communicate in an indirect, stigmergic way, through the information they locally read from and write to the nodes.
- The specific task of each forward ant is to search for a minimum delay path connecting its source and destination nodes.
- The forward ant migrates from a node to an adjacent one towards its destination. At each intermediate node, a stochastic decision policy is applied to select the next node to move to. The parameters of the local policy are: (i) the local pheromone variables, (ii) the status of the local link queues (playing the role of heuristic variables), and (iii) the information carried into the ant memory (to avoid cycles). The decision is the results of some tradeoff among all these components.
- While moving, the forward ant collects information about the traveling time and the node identifiers along the followed path.
- Once arrived at destination, the forward ant becomes a backward ant and goes back to its source node by moving along the same path = [s, v1, v2 d] as before but in the opposite direction. For its return trip the ant

makes use of queues of priority higher than those used by data packets, in order to quickly retrace the path.

– At each visited node and arriving from neighbor the backward ant updates the local routing information related to each node in the path followed by the forward ant from to d, and related to the choice of as next hop to reach each . In particular, the following data structures are updated: a statistical model of the expected end-to-end delays, the pheromone table used by the ants, and the data routing table used to route data packets. Both the pheromone and the routing tables are updated on the basis of the evaluation of the goodness of the path that was followed by the forward ant from that node toward the destination. The evaluation is done by comparing the experienced traveling time with the expected traveling time estimated according to the local delay model.

– Once they have returned to their source node, the agent is removed from the network.

– Data packets are routed according to a stochastic decision policy based on the information contained in the data-routing tables. These tables are derived from the pheromone tables used to route the ants: only the best next hops are in practice retained in the data-routing tables. In this way data traffic is concurrently spread over the best available multiple paths, possibly obtaining an optimized utilization of network resources and load balancing. Node data structures used the ant agents in AntNet for the case of a node with L neighbors and network with N nodes. For simplicity the identifiers of the neighbors are supposed to be 1, 2,L. Both the ant-routing and data-routing tables are organized as in distance-vector algorithms, but the entries are not distances but probabilities indicating

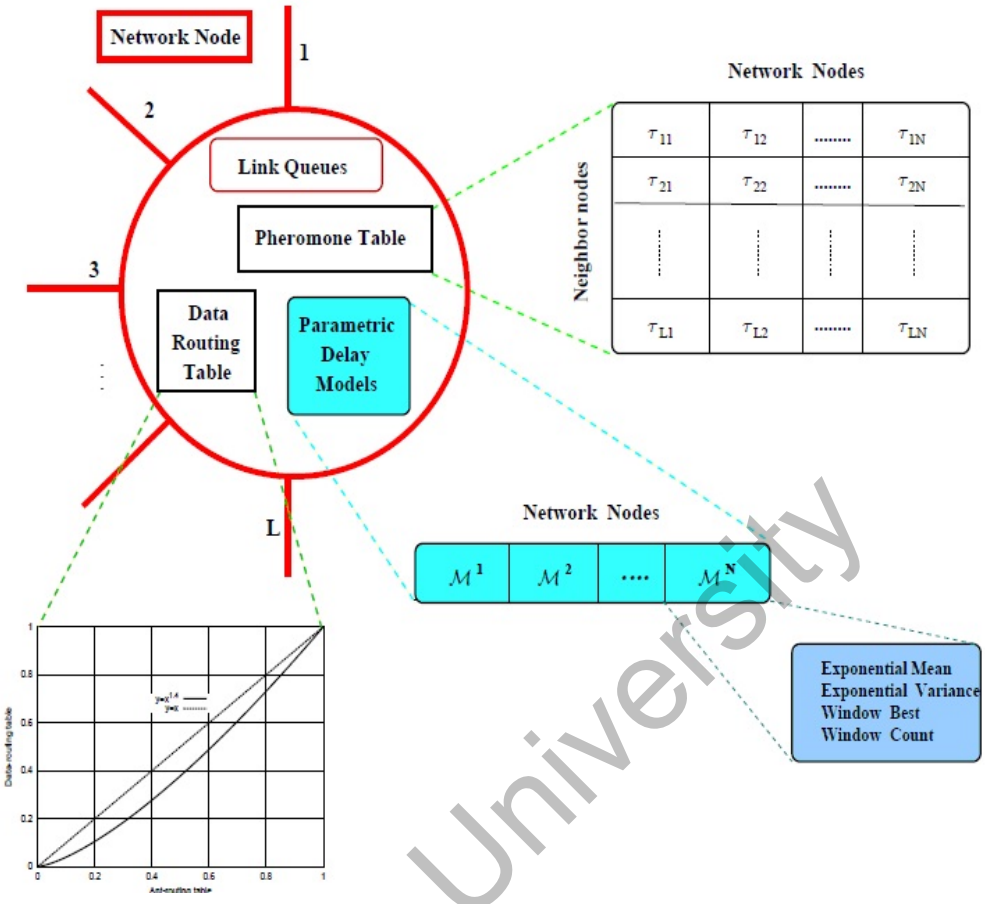


Figure 2.13 Data Structure maintained at the nodes [2]

the goodness of each next hop choice. The data-routing table is obtained by the ant-routing table by means of an exponential transformation as the one showed in the graph in the lower-left part of the figure. The entries of the vector of delay models are data structures representing parametric models for the expected traveling times to reach each possible destination from the current node. Also the current status of the link queues (in terms of bits waiting to be transmitted) is used by AntNet, and it is represented in the upper part of the node diagram.

Chapter 3

Problem Statement

Intrusion detection is used at layer 7 to filter the content according to rules and regulations postulated by network policy. Industry standard IDS like snort and bro exist in the market and have good success rate with traditional Dbase based detections. We wish to explore and implement Ant Colony Based Optimization for IDS for better success rate against anomaly-based attacks.

- To study and explore various IDS and their respective pros and cons.
- Design and Develop ACO based IDS wrapper.
- Demonstrate the use using Isolated University Network.

Chapter 4

Implementation and Results

4.1 BRO Intrusion Detection System using ACO

Bro is a research tool being developed by the Lawrence Livermore National Laboratory. Bro is a Unix-based Network Intrusion Detection System (NIDS).Bro was designed and developed by Vern Paxson of ICSI's Centre for Internet (ICIR), Berkeley. He started the project in the year 1995 and bro is under active development since then. Bro is operationally deployed at the University of California, Berkley and at LBL .Bro fundamental design goal is to separate policy and mechanism. Bro is neither fundamentally anomaly based nor misuse-based intrusion detection system because bro is by policy neutral itself, the network activity is abstracted into events which are further passed to policy layer. On the policy layer the administrator defines its own environmental constraints by writing its own custom scripting in a scripting language. Bro is very customizable, and there are several ways to modify Bro

to suit your environment. You can write your own policy analyzers using the Bro language. Most sites will likely just want to do minor customizations, such as changing the level of an alert from "notice" to "alarm", or turning on or off particular analyzers.

Bro provide a real time network monitoring. It comes with a predefined set of policy scripts which should be tuned by the user to the specific network in which it shall run. Deploying and understanding Bro takes more time than other IDSs. The Bro language is harder to learn than e.g. Snort's language for building custom rules. The coding of Bro is also not very well polished and the documentation often lies behind the recent version. Bro comes also with a module for converting Snort rules. In this way a user can download Snort rules and can use these in Bro.

4.1.1 Architecture of Bro

Below shown the basic architecture of Bro Intrusion Detection System using ACO policy event that consists of policy layer, event engine and packet.

Packet Capture

The packet capture layer utilizes libpcap to capture packets from the network. Libpcap also includes a filter which reduces the amount of packets to be analyzed. The filter is built during run-time according to the policy scripts given by the user. As an example, if no policy script handles http traffic, the filter will then discard all http traffic at the packet capture layer and no http traffic is sent up to the upper layers. This also allows a significant fraction of the

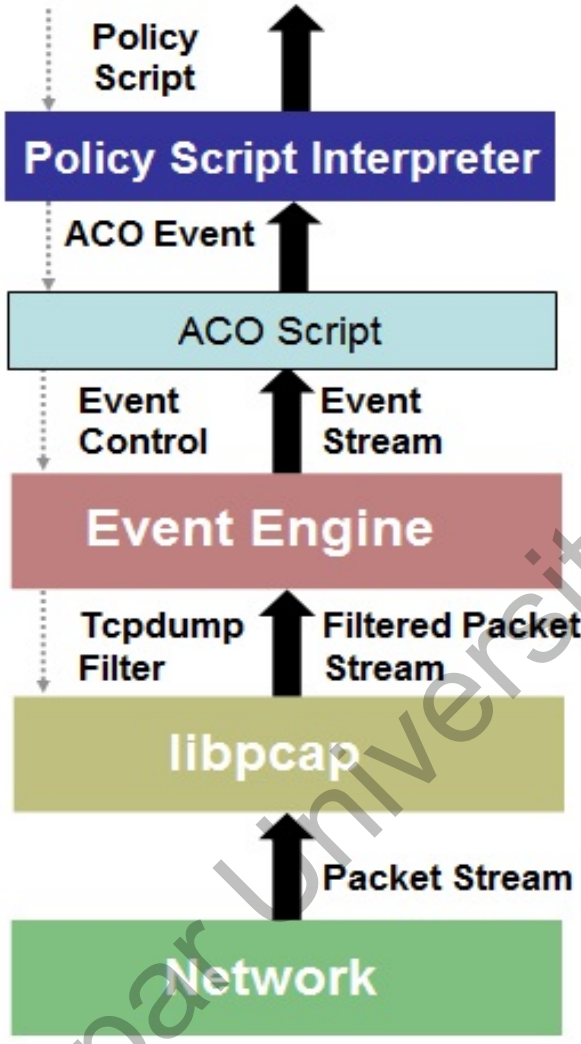


Figure 4.1 Bro architecture including ACO policy Event

packets entering the network to be rejected at a low level. Thus libpcap will capture all packets associated with the application protocols (e.g., finger, ftp, telnet) of which Bro is aware.

Event Engine:

At event engine layer the low-level analysis of network packets takes part. The event layer performs integrity checks on packet headers first, single IP packets are collected and then TCP data streams are reassembled. At the transport level TCP, UDP and ICMP packets are analyzed. At the application level

HTTP, SMTP, DNS, FTP and many more are analyzed. If an analyzer finds anything interesting it generates an event which is sent up to the policy layer for processing. Events are generated from this process and placed on a queue to be interrogated by the policy script interpreter which resides in the third layer.

Policy Layer:

At this layer, policy scripts written in the proprietary Bro-language by the user handle events generated by the event engine. An event can be processed by several handlers successively. When Bro starts, it looks through the enabled event handlers to find which event engine analyzers to start. To add new capability to Bro, one needs to identify the events associated with the protocols of the application, and write corresponding event handlers to extend the functionality of the policy script interpreter which improves Bro's extensibility.

4.2 Implementation Setup

In this part the policy scripts of Bro to detect various network intrusions are discussed. Various kind of traffic is captured by using Bro and analyzed offline. Some of the scripts are experimented on the live traffic also.

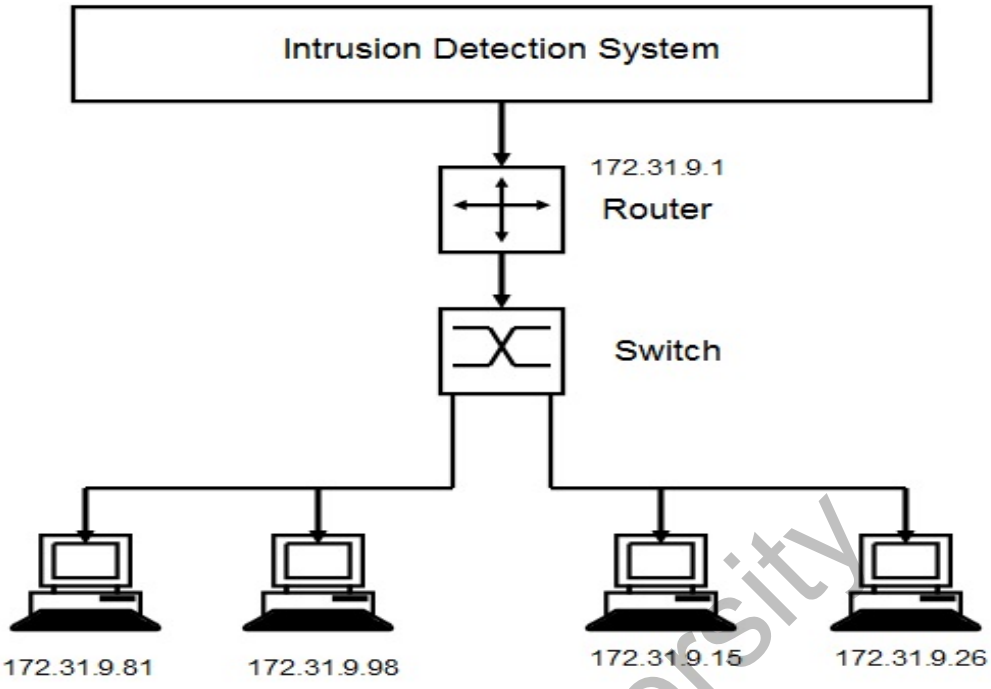


Figure 4.2 Implementation Setup

4.2.1 Capturing Packet using pcap sniffer component augmented using customized ACO pre processor policy

Following analysis of "Slammer.pcap" file describes several points which are helpful in Intrusion Detection.

- The captured file display that intrusion can be found in an Isolated network, because IP addresses are local.
- The local IP of host systems 172.31.9.81 and 172.31.9.1 are source and destination respectively.
- The captured file shown that the bit number 27, 28, 29 and 30 are AC,1F,09,51 hex value of Source machine has the IP address 172.31.9.81. like that bit number 31,32,33,34 are AC,1F,09,01 hex value of destination machine have the IP address 172.31.9.1.

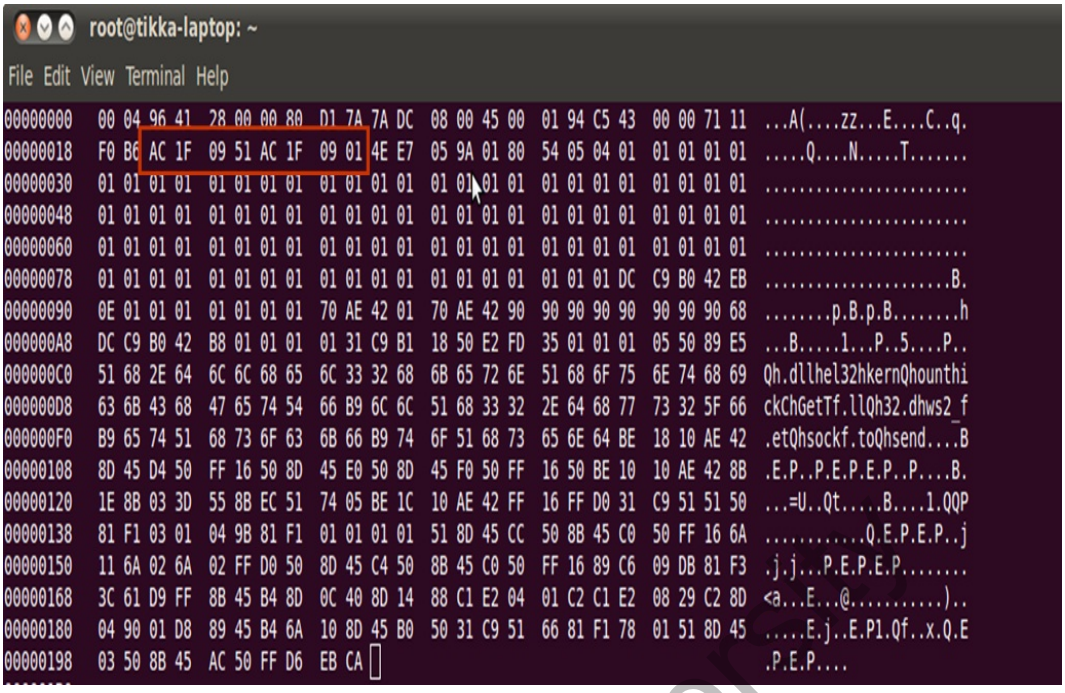


Figure 4.3 IP Capture Source and Destination IP addresses

– The MAC address of source machine is 00:80:D1:7A:7A:DC and destination MAC address is 00:04:96:41:28:00.

This report describe the intrusion in a captured file with intrusion type source and destination address and host machine location. By using the all details we process Intrusion Detection process proceed towards Anomaly Detection using following scripts.

HTTP URLs reporting in the traffic with traditional approach

In this milestone all the URLs which are visited by a particular machine are reported by the Bro.

Script: s_http-header.bro

Script to report all the HTTP URLs in the traffic.

@load weird

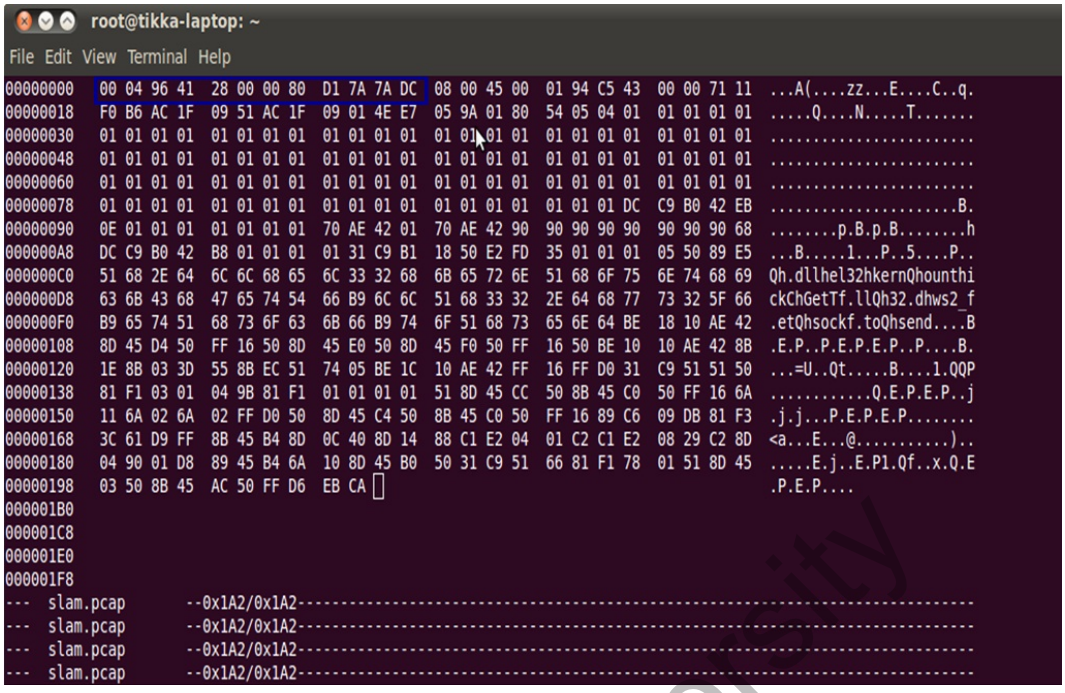


Figure 4.4 MAC Capture, 12-bits represents Source and Destination MAC addresses

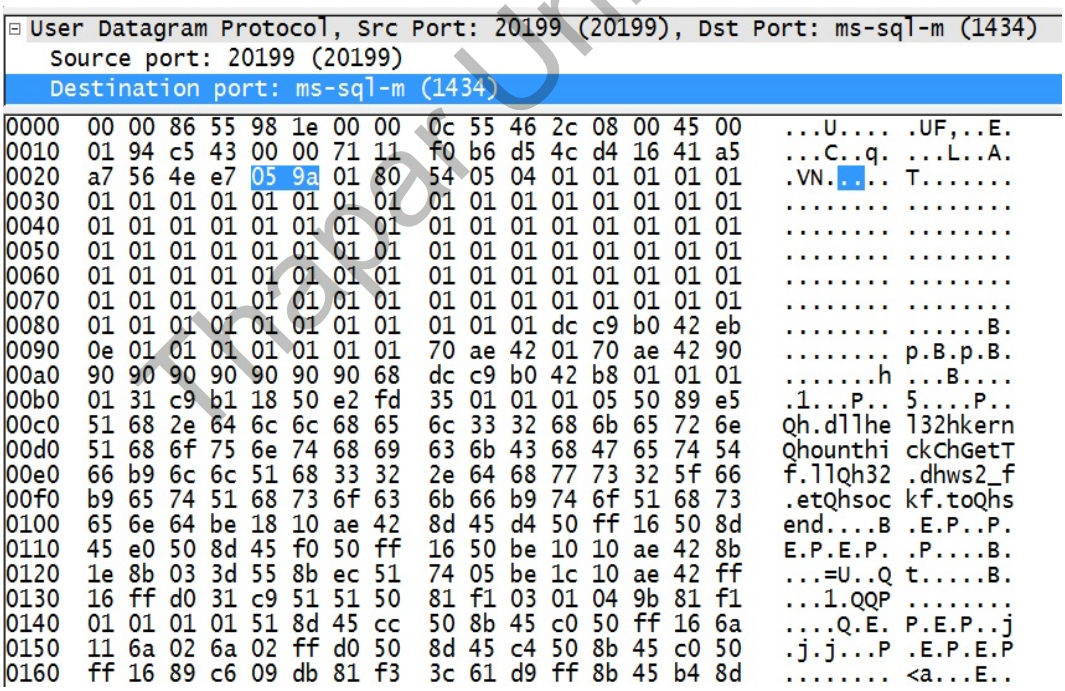


Figure 4.5 Sql Destination port

```
@load alarm
@load http
global path: string;
redef ignore_checksums = T;
event http_request(c: connection, method: string, original_URI:
string,
unescaped_URI: string, version: string)
{
path = original_URI;
{
event http_header(c: connection, is_orig: bool, name: string, value:
string)
{
if(name == "HOST" )
{
print fmt("URL: %s%s",value,path );
}
}
```

Trace File: trace.out

This is the captured traffic file against which s_http-header.bro policy script will run.

The following is the command to see the results:

bro -r trace.out s_http-header.bro

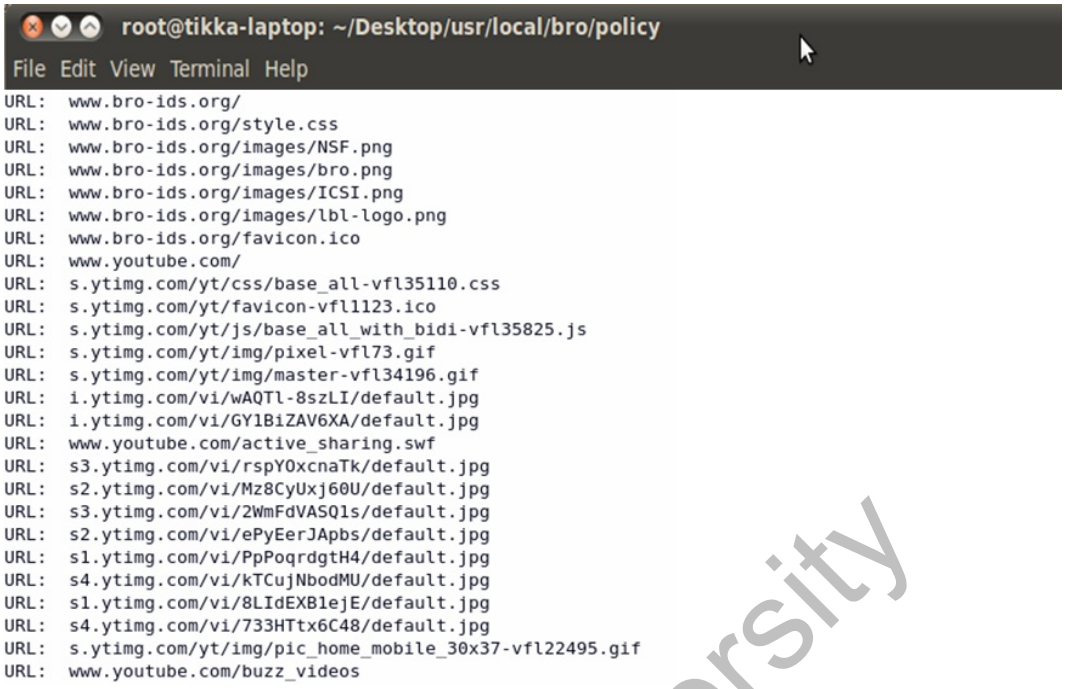


Figure 4.6 Reporting HTTP URLs visited by a host in traditional approach using Bro IDS

All the connections which are accessing *www.youtube.com* using HTTP with ACO algorithm approach using Bro IDS

Script: s_http-header1.bro

Bro script to report all the connections that are accessing www.youtube.com

using HTTP

@load weird

@load alarm

@load http

global path: string;

redef ignore_checksums = T;

```
event http_request(c: connection, method: string, original_URI:
string,
unescaped_URI: string, version: string)
{
path = original_URI;
}
event http_header(c: connection, is_orig: bool, name: string, value:
string)
{
if(name == "HOST" )
{
local v = value;
if("www.youtube.com" in v )
{
print fmt("CONNECTION IS: %s:%s>%s:%s%s%s",
c$id$orig_h,c$id$orig_p,c$id$resp_h,c$id$resp_p,v,path);
}
}
}
```

Trace file: trace.out

The above is the captured traffic file against which s_ http-header1.bro policy script will run.

The following is the command to see the results:

bro -r trace.out s_http-header1.bro

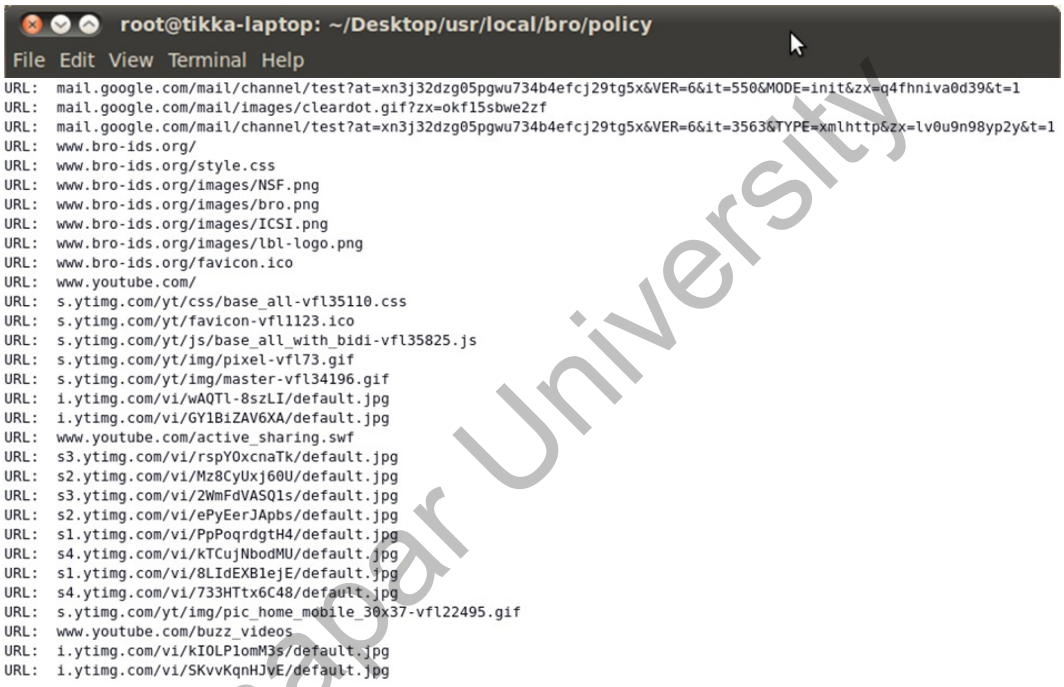


Figure 4.7 HTTP URLs reporting visited by a host with ACO algorithm using Bro IDS

Chapter 5

Conclusions and Future Scope

In this thesis work we have explored and designed the policy scripts of Bro using ACO policy script to detect various kinds of traffic like web traffic, mail traffic, web mail traffic etc. The scripts are experimented against captured traffic as well as live traffic. Results obtains show the results using ACO is comparatively better and more elaborative, with this technique system is able to raise alerts based on anomalous HTTP packets. This anomalous behavior can be augmented with the help of pre processing scripts. However no IDS can detect all the intrusions. So we need a combination of anomaly and signature based techniques.

Future Scope In future work can be extended to include other protocol anomalous behavior e.g FTP, SMTP. Complete API(Application Programming Interface) can be design for purposed ACO-IDS architecture.

References

[1] AntNet: ACO routing algorithm in practice Vincent Verstraete, Matthias Strobbe, Erik Van Breusegem, Jan Coppens, Mario Pickavet, Piet Demeester Ghent University - IBBT - IMEC, Department of Information Technology Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium (1998).

[2] Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks Gianni Di Caro Bruxelles, September 2004.

[3] ACO based Distributed Intrusion Detection System S. Janakiraman, V. Vasudevan International Journal of Digital Content Technology and its Applications ,Volume 3, Number 1, March 2009.

[4] B.M.T. Lin C.Y. Lu S.J. Shyu C.Y. Tsaic Development of new features of ant colony optimization for flowshop scheduling International Journal Production Economics 742-755 (2008).

[5] Ant Colony Optimization: A Modified, Version Shuchita Upad-

hyaya, Richa Setiya Int. J. Advance. Soft Comput. Appl., Vol. 1, No. 2, November 2009 ISSN 2074-8523; Copyright ICSRS Publication, 2009 www.i-csrs.org

[6] The Ant Colony Optimization (ACO) Metaheuristic: a Swarm Intelligence Framework for Complex Optimization Tasks by Gianni Di Caro gianni@idsia.ch IDSIA, USI/SUPSI, Lugano (CH)

[7] Ant Colony Optimization Vittorio Maniezzo, Luca Maria Gambardella, Fabio de Luigi

[8] Rules definition for anomaly based intrusion detection by Lubomir Nistor vol 1.1, 2002-2003

[9] Intrusion Detection Techniques and Approaches Theuns Verwoerd and Ray Hunt Computer Communications, Volume 25, Issue 15, 15 September 2002, Pages 1356-1365

[10] L. Spitzner, Honeypots: Tracking Hackers. Addison-Wesley, 2003.[Online]. Available: http://www.tracking-hackers.com/book/

[11] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Computer Networks (Amsterdam, Netherlands: 1999), vol. 31, no. 23-24, pp. 2435-2463, 1998. [Online]. Available: http://citeseer.nj.nec.com/article/paxson98bro.html

[12] Krugel Christopher, Toth Thomas: A Survey on Intrusion Detection Systems. TU Vienna, Austria (2000) 7, 22-33

[13] A Honeypot Architecture for Detecting and Analyzing Unknown Network Attacks P. Diebold, A. Hess, G. Schafer 14th conf. kommunikation in verteilten system 2005 (kiVS05), kaiserslautern, Germany, February 2005.

[14] Bro Quick Start Guide, Vern Paxson, Jim Rothfuss, Brian Tierney, version 0.9, DRAFT, 11-15-2004.

[15] An improved Ant System algorithm for the Vehicle Routing Problem, Bernd Bullnheimer, Richard F. Hartl and Christine Strauss, Annals of Operations Research 89(1999) 319-328.

[16] M. Dorigo, Optimization, Learning, Natural Algorithms (in Italian), PhD thesis, Dipartimento di Elettronica, Politecnicodi Milano, Italy, 1992.

[17] M. Dorigo, T. Stuzle, Ant Colony Optimization, MIT Press, 2004.

[18] Firewall Architecture Understanding the purpose of a firewall when connecting to ADSL network services. A Nextep Broadband White Paper June 2001.

[19] Intrusion Detection in Wireless Ad-Hoc Networks Foong Heng Wai,Yin Nwe Aye, Ng Hian James International Conference on Mobile Computing and Networking, 6th annual international conference on Mobile computing and networking Pages: 275 - 283 (2000).

[20] Anomaly Detection of Web-based Attacks Christopher Kruegel, Giovanni Vigna, Conference on Computer and Communications Security 10th ACM conference on Computer and communications security, Washington D.C., USA, Pages: 251 - 261 (2003).

[21] BRO - an IDS IMT5151 Intrusion Detection and Prevention -Vidar Evenrud Seeberg. December 2005.

[22] A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks , K.Q. Yan, S.C. Wang, C.W. Liu, International MultiConference of Engineers and Computer Scientists 2009 Vol , IIMECS 2009, March 18 - 20, 2009, Hong Kong.

[23] Design and implementation of policy- based intrusion detection system-Generic Intrusion Model for a distributed network by Akhil Narayan Karkera, a thesis presented to the graduate school of the University of Florida (2002).

[24] An Ant Colony Optimization Algorithm for Network Vulnerability Analysis by M. Abadi, S.Jalili , IRANIAN JOURNAL OF ELECTRICAL ELECTRONIC ENGINEERING, Volume 2, Num-

ber 3 (2006), Pages 106 - 120.

[25] Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks by Gianni Di Caro PROMOTEUR: PROF. MARCO DORIGO IRIDIA, Institut de Recherches Interdisciplinaire set the Development Intelligence Article (2003-2004).

[26] Guide to Intrusion Detection and Prevention Systems, Karen Scarfone, Peter Mell Special Publication 800-94, (February 2007).

[27] http://www.astarservices.com/index.php?
page=Security-HoneyPots

[28] http://en.wikipedia.org/wiki/Denial-of-service$_attack$

[29] http://www.spywareguide.com/term$_show.php?id = 21$

[30] http://netsecurity.about.com/cs/hackertools/a/aa121403.htm

[31]
http://fadils.wordpress.com/2008/06/03/network-security-wheel/

[32] http://www.janisian.com/forum/showthread.php?6530-Buffer-Overflowp=87907

[33] http://en.wikipedia.org/wiki/Ant$_c$*olony*$_o$*ptimization*

List of Publications

"Ant Colony Optimization(ACO) based Intrusion Detection Sytem", Tikka Singh, Maninder Singh, "Journal of Network and Computer Applications", Communicated June, 2010.