Information Encryption and Intrusion Detection in Mobile Agent Based Ad-Hoc Networks

Dissertation

Submitted in partial fulfillment of requirement for award of degree of

Master of Technology
in
Computer Science and Applications

Submitted By

Jasleen Kaur

(Roll No. 601203010)

Under

Supervision of

Dr. Sharad Saxena

Assistant Professor
SMCA, Thapar University, Patiala

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS
THAPAR UNIVERSITY
PATIALA – 147004

July 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Information Encryption and Intrusion Detection in Mobile Agent Based Ad-Hoc Networks"**, in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Application submitted in School of Mathematics and Computer Applications (SMCA), Thapar University, Patiala, is an authentic record of my own work carried-out under the supervision of **Dr. Sharad Saxena, Assistant Professor, SMCA, Thapar University, Patiala** and refers other researchers' work which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for award of any other degree of this or any other University.

(Jasleen Kaur)
601203010

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

17/7/2014

(Dr. Sharad Saxena)

Assistant Professor
SMCA, Thapar University
Patiala

Countersigned by:

(Dr. Rajesh Kumar)

Head SMCA
Thapar University
Patiala

(Dr. S. K. Mohapatra)

Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor Dr. Sharad Saxena. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to Dr. Rajesh Kumar, Associate Professor and Head, School of Mathematics and Computer application and Dr. Singara Singh, P.G. Coordinator for their constant support, motivation and inspiration that triggered me for the thesis work.

I am also thankful to the entire faculty and staff members of SMCA for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my parents for their wonderful love and encouragement, without their blessings none of this would have been possible. I would also like to thank my close friends for their constant support.

Jasleen Kaur
(601203010)

ABSTRACT

A Mobile Ad hoc network (MANET) is a network that offers better speed, and formulates a fully symmetric distributed network. As topology of network changes vigorously in an erratic manner, we can use mobile agents, which travel autonomously within the network, execute solely in agent's environment, collect required facts/data and make conclusion according to the program carried by it. The execution environment is provided to mobile agent by host. These agents' have a very flexible nature as it can travel from host to host. As agent travels in ad-hoc environment which is highly prone to various threats

Our work focuses on providing communication for transferring information from one node to another, further we want to provide security to mobile agent in ad-hoc environment by implementing Cryptographic Algorithm thus preventing information from being cracked by any intruder and protecting transport layer and Intrusion detection and prevention to prevent agents from malicious attacks by writing a rule set consisting of threats signature using SNORT. SNORT includes ips with itself which protect application layer from being attacked by blocking or dropping packets. The agents are developed by using technologies like JADE and Intrusion detection is done using SNORT.

LIST OF CONTENT

LIST OF ACRONYMS

MANET	:Mobile Ad-Hoc Network
WLAN	:Wireless Local Network
RPC	:Remote Procedure Call
REV	:Remote Evaluation
DoS	:Denial of Service
OSI	:Open Systems Interconnections
SYN	:Synchronization
IDS	:Intrusion DetectionSystem
MAC	:Media Access Control
NIDS	:Network Intrusion Detection System
HIDS	:Host Intrusion Detection System
IDPS	:Intrusion Detection and Prevention System
ASDK	:Aglet Software Development Kit
ABE	:Aglet Building Environment
JADE	:Java Agent Development Framework
FIPA	:Foundation for Intelligent Physical Agents
RMI	:Remote Method Invocation
CRT	:Cathode Ray Tube
ATDSR	:Agent-Based Trusted Dynamic Source Routing
DASR	:Distributed Anonymous Secure Routing
SNMP	:Simple Network Management Protocol
RAS	:Routing Agent System
MAS	:Monitoring Agent System
MOA	:Monitoring Agent
ROA	:Routing Agent
TREQ	:Trust Request
TREP	:Trust Reply
RTS	:Ready to Send

TTP	:Trusted third party
SNMP	:Simple Network Management Protocol
LIDS	:Local Intrusion Detection System
UDP	:User Datagram Protocol
SIM	:Secure-Image Mechanism
SENSE	:Self-Executing Security Examination
PARC	:Partial Result Authentication Codes
PKI	:Public Key Infrastructure
AMS	:Agent Management System
DF	:Directory Facilitator
ACC	:Agent Communication Channel
GUID	:Globally Unique Identifier
XML	:Extensible Markup Language
SMB	:Sending Server Message Block
TCP	:Transmission Control Protocol
DAQ	:Data Acquisition
NFQ	:NetFilter Queue
IPS	:Intrusion Prevention System
RSA	:Rivest Shamir Adleman
HTTP	:Hyper Text Transfer Protocol

LIST OF FIGURE

LIST OF TABLES

CHAPTER 1
INTRODUCTION

An ad-hoc wireless network is a network; that consist of mobile computable devices that uses wireless broadcasting for communication, and which have portable and adjustable framework (primary management like access points of WLAN(wireless local area network) or base stations of cellular wireless network). Mobile ad-hoc network (MANET) encompasses set of mobile hosts. The mobile hosts don't need assistance of base stations for communication with each-other.

MANET has self-configuration and self-maintenance capability, also because of which received tremendous attention in recent years. Early research was focused on primarily providing cooperative and secure hostile environment. Even though security is a recent active topic for research in wired network, MANET has some nontrivial challenges that need a secure design. We cannot apply directly the already existing security protocol to MANET domain.

MANET is a collection of mobile devices which can form a temporary network if there is no fixed infrastructure provided. Each node communicates via wireless interface. The routing in mobile ad-hoc networks have two phases: route detection and route maintenance. Route detection tells which route source node should follow to destination who receives packets. Route Maintenance let node S to detect whether one or more links along the route have failed or not [1]. The attacker can hamper the route discovery by impersonating the destination, by responding with corrupted routing information. To provide compendious security, both phase of MANET communication must be safe guarded [2].

Wireless transmission has a limited range for mobile devices these devices also serve as routers, that is, before reaching to the final destination a number of devices are required to route or relay a packet. Ad-hoc wireless networks should be organized rapidly everywhere and at any moment of time as it reduces the difficulty of setup of infrastructure. This type of network has

application in a number of different areas. Some of the network some times consists of communication for military purpose by using wireless devices.

1.1 MOBILE AGENT COMMUNICATION MODEL

Mobile agent is an advance programming beliefs, which has strained tremendous amount of attention of researchers. Mobile agent technology has made distributed computing over the internet more secure and safe [1].

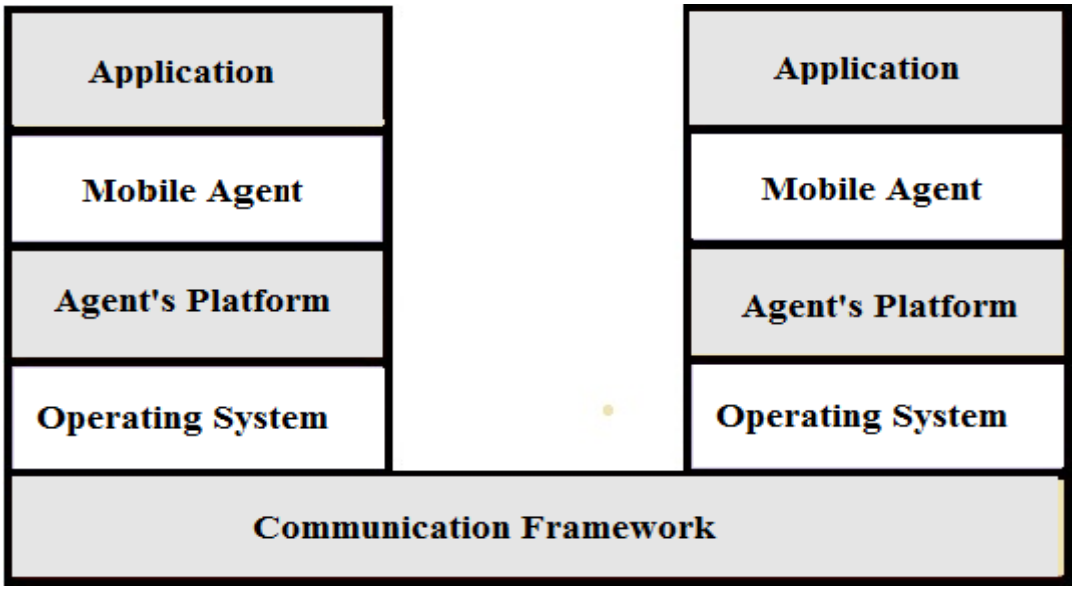


Figure 1.1: Mobile Agent Architecture

Figure 1.1 shows mobile agent architecture. A mobile agent autonomously travels within the network, execute solely in agent's environment, collect required facts/data and make conclusion according to the program carried by it [4]. Many researchers gave definitions of the mobile agent according to their different perspectives. Agents are defined focusing on their autonomy property by Pattie Maess [5]; Michael Coen concentrated on mediation and correlative of transferring information by agents [6]; Stan Frankline and Art Graesser's concentrated on the fundamental nature and behavior of agent, that is don't even related directly to s/w mobile agent [7]. For our work, we have considered, mobile agent as a software program that perform definite task and migrate from one node to another.

The basic property that any agent should consist of are:

- **Autonomy**

Agents have the intelligence of doing tasks. Autonomy has made agent to think. They have their intelligence and this make performance of agent's task efficient. Autonomy for agent can be programmed in advance [4].

- **Mobility**

Agent should have the ability to carryout task over distributed environment. Hence agent can defer the execution from the first node and transmit itself to the next node and restart its execution there. Agent's path on which agent is going to migrate can be pre-established or made at the execution time of agent [4].

- **Reactivity**

Interaction of agent with other agents inside the environment is also one of the key features. The agent should also give an apt response and in an appropriate manner [4].

1.1.1 MOBILE AGENT TECHNOLOGY

Mobile Agent is a software program that moves from one machine to another with its own code and data. After reaching to destination the programs are loaded and the task is executed in the way they are programmed. Mobile Agent on overall network interacts with other agents while moving with in the environment. Mobile agent platform is responsible to send, receive, form, execute, transfer and demolish mobile agents hence acts as a distributed middleware. A number of models are their which describes Agent's system; here, for security problems we will use a simple one, which will mainly consist of mainly two components: Agents, and Agent's Platform [2]. For our work we will consider, Agent as a collection of program, facts and control information which is carried during execution on hosts they visit/reside. Agent platform is the environment where the agent resides and executes. The platform where an Agent is initialized is known as Home Platform and it's the most reliable from all.

Mobile agents grant some benefits of creating distributed applications which will include reduction in network load, will overcome network latency, will have faster interaction and will

also provide disconnected operations[3]. Mobile Agents are useful for applications that require distributed environment for information retrieval as they move to different locations for execution. Software mobile agents are used by people to overcome their tedious repetitive job and time consuming activities.

Data Communication can be said as a suitable application for mobile agents. In e-commerce also Mobile agent technology can be used to automate several stages that consume a lot of time during buying and selling procedure. Figure 1.2 shows migration of agent from one platform to another.

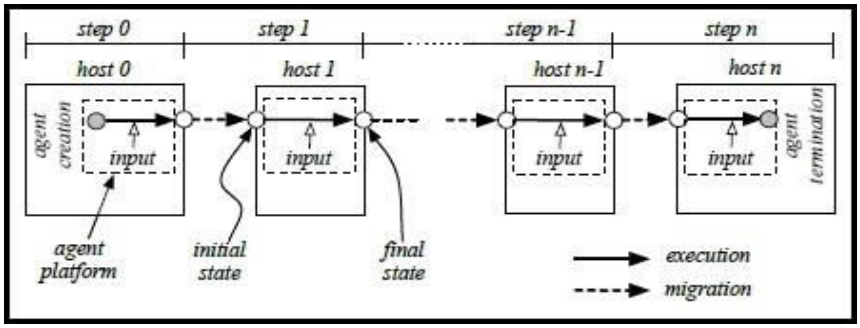


Figure 1.2: Migration of Agent [4]

Mobile agents are autonomous and customized compared to other "traditional" software. Mobile agent migrates with in the network and can search for the product specified by the user across every shop. When mobile agent moves to different sellers, the exchange of information is done in a local manner and the exchange is not over network, which saves latencies and overload with in network. Mobile agent technique can be used for direct execution of client specific queries at shop's sites. The inbuilt qualities of Mobile agents are favorable for optimizing the whole buying and selling skill and this will also revolutionize communication over the net.

1.1.2 CHARACTERISTICS OF MOBILE AGENT

Following are the common characteristics of Mobile agent [8]:

- **Autonomous**: Mobile agents should have the ability to act and taking decision without direct external intervention. They should be able to control their information and states. They should have power of decision making, *i.e.,* what to do, where to go and when to go.

- **Mobility**: It is the essential key characteristic of mobile agent who is able to drift itself from one host to another in a diverse network.
- **Proxy**: For benefit of some unit mobile agents sometime act as somebody else.
- **Proactive**: Mobile agent should respond to the network and thus should be goal-oriented.
- **Intelligent**: Through regress learning mobile agents are able to take decision.
- **Learning**: Ability of gathering knowledge from network helps in taking decision.
- **Cooperative**: Mobile agent should be able to synchronize with other agents to attain common goal.
- **Disconnected operation**: In the absence of a network connection also mobile agent should be able to work.

Use of Mobile agent reduces network traffic which is useful for huge volume of data; Instead of transferring Information over network the data is manipulated locally. Due to mobile and adaptive property it offers real-time response and overcomes network latency.

According to behavior subgroup are made for agents, these behaviors helps the agent to migrate from one place to another by some means. Finally, event's set are defined that mainly concern to the agent during its lifetime which makes model of mobile agent complete. The set of event varies from model to model; following is the most commonly used events [9]:

- **Creation** - Corresponds to the constructor of an object. Event handler should initialize state and plan the agent for further more instructions that are going to be carried out.
- **Dumping** - Analogous to the destructor of an object. Here event handler should free all the resources that the agent is using.
- **Transmit** – Send signals to the agent to plan for departure to a latest location. This event can be initiated explicitly by the agent itself upon making a request to migrate, or it can be triggered by some other agent that wants this agent to move.

- **Arrival** – Agent sends signals that it has successfully arrived at its new node and it has started performing its duties.
- **Communication** – This event sends notification of handling incoming messages from other agents and primarily means inter-agent connection.

Below Figure 1.3 shows the life cycle of mobile agent also tells from where agent starts, how agent moves from one state to another and how its life ends.

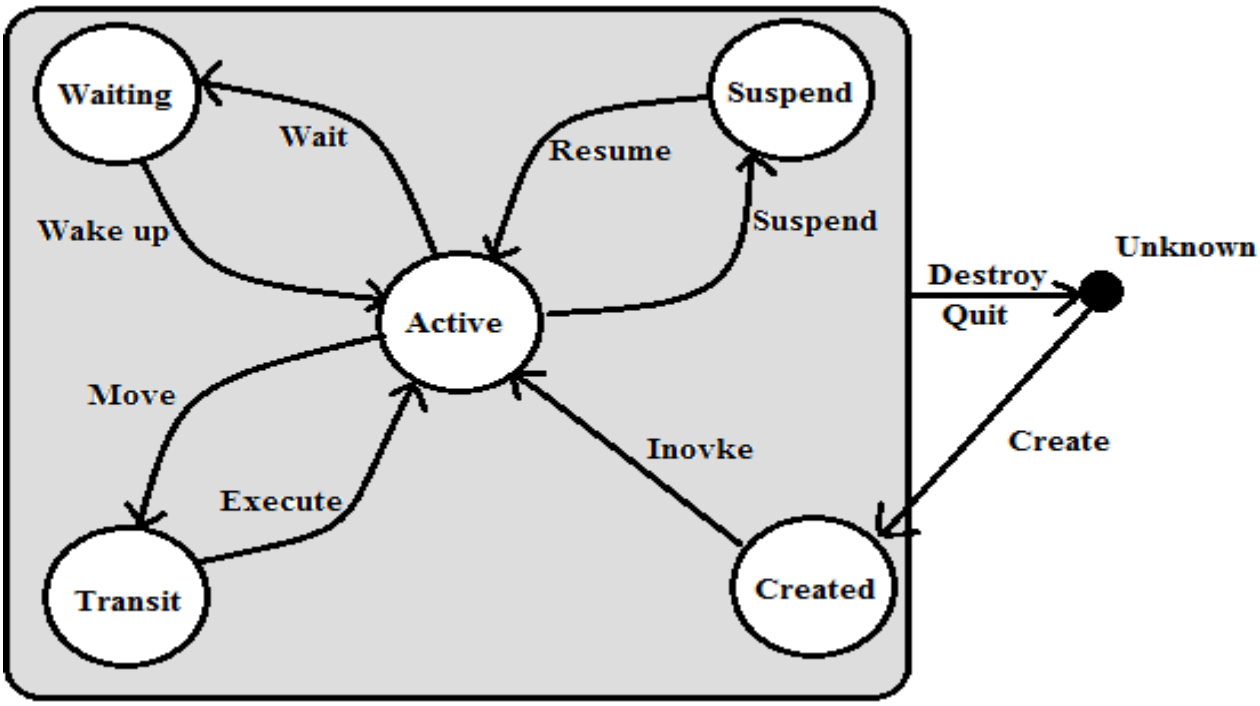


Figure 1.3: Mobile agent Life Cycle [9].

1.2 CLIENT SERVER COMMUNICATION MODEL

Every process or computer is either a client or a server on the network then the network architecture is called as client-server [10]. Figure 1.4 shows client-server architecture. Client server model consist of a client that sends request which is managed by the server on its behalf. The server as required performs request/response. Earlier communication between client and server is achieved through one of these: Message passing, Remote Procedure Call (RPC) and Remote Evaluation (REV). . In RPC, the procedures are invoked by agent through procedure

call. In RPC client act as caller and sever as callee. Server respond by sending outcome of executed procedure to the client. In REV, server directly downloads method for the client.

A computing environment is defined by Client Server architecture where network based communication is categorized as two processes: that run over multiple processors Back end and Front end. Front end and Back end interact with each other in such a way that applications are completed as combined one task.

Client Server architecture doesn't serve as a vital platform for many of the applications some of the disadvantages is as follows [10]:

1. It's a primary part hence major portion of application runs on server and therefore resources will be in continuous use by clients. That can be a source for resource congestion.
2. Due to lack of compatible development tools the back end and front end procedures needs to have composite method.
3. Lack of scalability as some network operating system is not very scalable.
4. Dependability is very high and so when server goes down operation also gets decreases

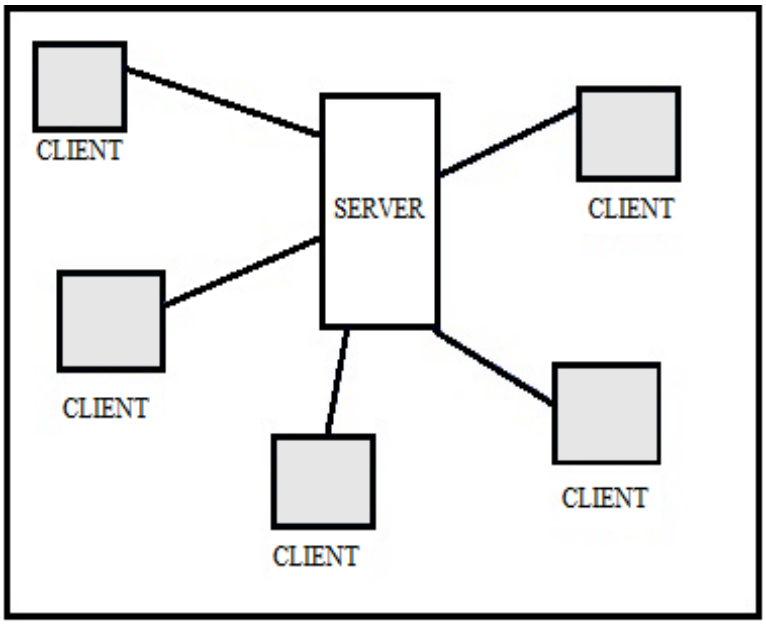


Figure 1.4: Client-Server Architecture

1.3 COMPARISION OF CLIENT SERVER AND MOBILE AGENT MODEL OF COMMUNICATION

Mobile Agent Technology imparts a new means of communication where the agents migrate with their code and data from one node to another. Also limitations of client/server model have overcome .In RPC, the procedures are invoked by agent through procedure call. In RPC client act as caller and sever as callee. Server respond by sending outcome of executed procedure to the client. In REV, server directly downloads method for the client [11].

Where as, mobile agent technology agent carry out the whole execution and overcome the limitations of above mentioned traditional approaches. Mobile agent reduces network bandwidth, it requires low power requirements and also support for mobile units, provide low latency interaction.

From Figure 1.5 we can clearly depict mobile agent architecture is better that client server architecture. For further clarification some of the advantages of mobile agent is discussed below.

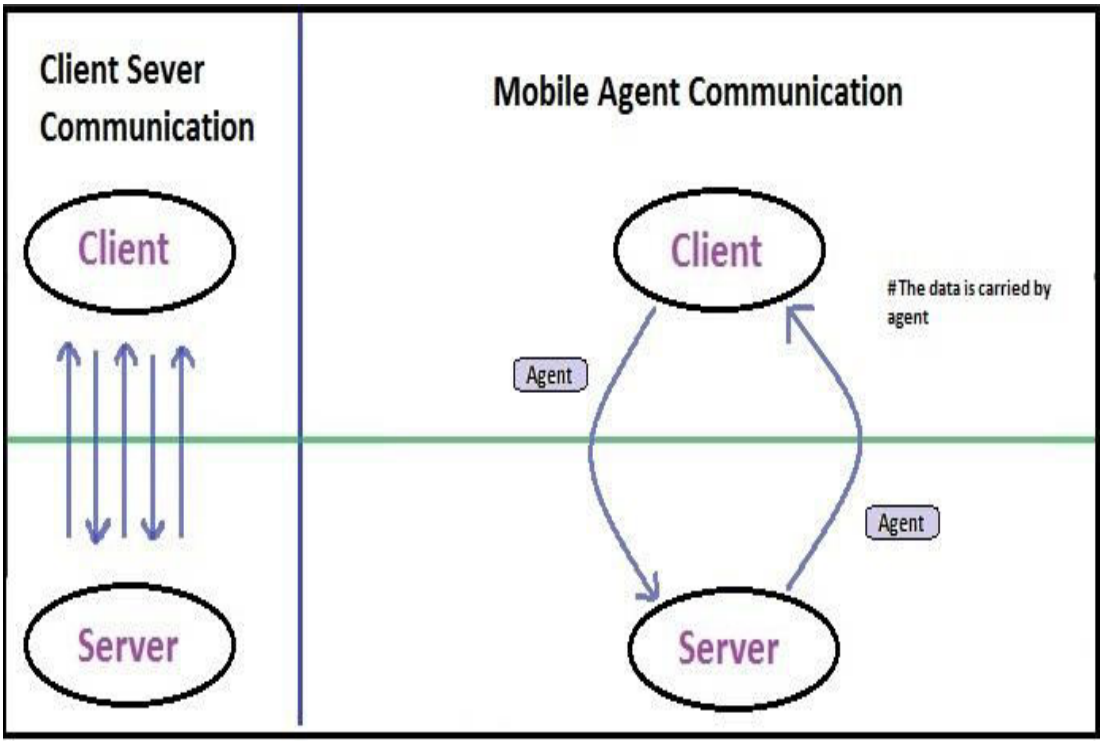


Figure 1.5: Comparative study of two different Communications Architecture

Advantages of Mobile Agents are listed below [12]:

- **Efficiency**: It consumes less network resource as movement is from computation to data rather than data to computation, which in turn increases efficiency.
- **Less bandwidth**: Bandwidth is used only when agent moves hence result in less network traffic.
- **Robustness and Fault Tolerance**: Dynamic reaction of agent to adverse situation increases fault tolerance in complex distributed system.
- **Support heterogeneous environment**: Mobile agent developed in Java is independent of computer and network as it runs on any system consisting JVM.
- **Support Data Communication**: e-marketing can be done using agents.
- **Easier Development**: Mobile agents are congenitally dispersed in nature; hence building a distributed system is easier.

1.4 THREATS TO MOBILE AGENT'S INFORMATION

Attacks is the infraction that consist of particular aim such as erasing or modifying something without knowing what is actually being erased or modified. Attack on agents can be done by following [13]:

- **Agents attacking host**: Malicious agents steals and modifies information on the host. These attacks are caused due to lack of adequate authentication and access control mechanisms. They can commit attacks like Denial of service (DoS) by draining computational resources and rejecting platform services.
- **Malicious host attacking agent**: Attack on agent can be because of malicious host, this attack can be by stealing or modifying agent's data, corrupting or altering its code, refuse requested services, return fake system call values, reinitialize agent or even lapse it completely. It can also bluff the agent by retarding the agent until the task is no more applicable. The Host may also evaluate and invalidate the agent.

- **Malicious agent attacking another agent**: Public methods of another agent are some times invoked by malicious agent to hinder with its work.

In intense cases whole agent will be deleted or tempered. From Figure 1.6 we can see that attacks mainly fall in two categories [14]:

1). Purposeful Attack

2). Frivolous Attack

1. **Purposeful Attack**: Attacks that are precisely designed so that attacker can take benefit from the attack. These types of attackers know what they are doing, why they are doing, and what will be the significance [14].

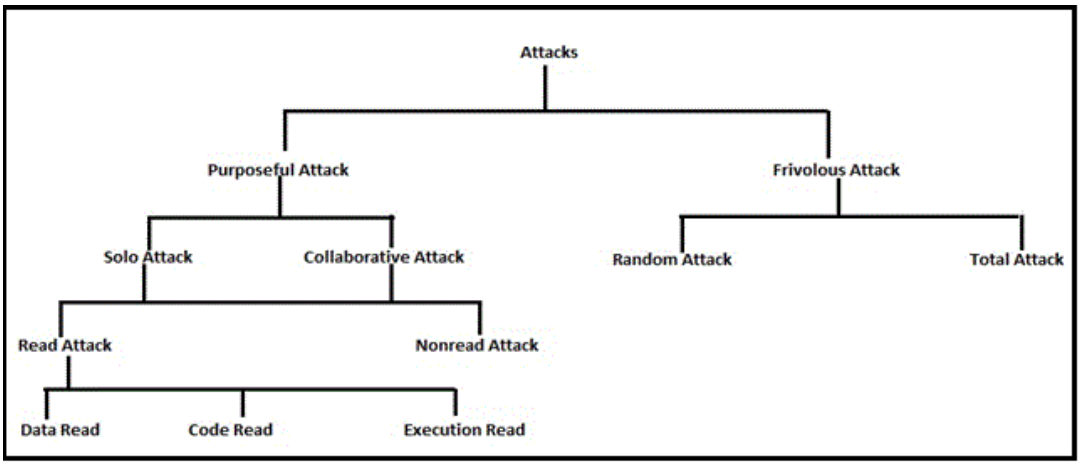


Figure 1.6: Different types of attacks[14]

Further according to the number of attackers we can have two kinds of attacks:

- **Solo Attack**: Single attacker attack and encompasses read, non-read and combination of both attacks.

- **Collaborative Attack**: Several attacker merge collectively to attain an attacking goal.

These attackers can attack in reading and non-reading pattern. Several hosts may associate together to track mobile agent in order to get information.

The nature of solo and collaborative attack is of two types [14]:

1) **Read attack**: These attackers read or duplicate the confidential information, rules or even the flow of control of a mobile agent. They leave no trace that might be spotted.

- **Data Reading**: It is simple and straight way to read out the private information.
- **Code Reading**: This leads to get knowledge about execution tactic of mobile agent.
- **Execution Reading**: With the arrangement of rules and data information attacker can infer more information about the status of the agent.

2) **Non-Read attack**: In this attacker vigorously make some action and thus leave some traces.

2. **Frivolous Attack**: Attackers that don't carefully plan and design attack plans hence attacker may or may not take the gain benefit from the attack. These attackers may alter or delete some bits without knowing what they have actually altered or deleted. They don't know the result for such exploits.

- **Random attack**: Arbitrarily change or delete some parts of agent's code or data. Sometimes just because of curiosity some attacks get happen.
- **Total attack**: Completely destroy the agent so that they cannot finish their task.

Primary goals for security in ad-hoc network can be provided by authentication, confidentiality, integrity, non repudiation, etc. Complete protection spanning should be provided to network. No single mechanism can provide all security service in MANET. The most common techniques are described below [15] [16]:

- **Confidentiality**: It guarantees that unauthorized person will never be disclosing certain information.
- **Authentication**: By enabling a node it makes sure of identity of the communicating node. To overcome authentication the attacker would imitate a node, and will gain unauthorized access to perceptive information and resources.

- **Availability**: Regardless of Denial of Service (DoS) attacks it guarantees the survivability. Attacker uses congestion techniques on physical and media access control layer to hamper physical channel of communication. The routing protocol can be interrupted by attacker on network layer. The attacker can get down the higher layers, high level services like key managing check.

- **Integrity**: The message transmitted in the network is never corrupted.

- **Non-impersonation**: The attackers uses fabrication that include fabrication message which is a fake routing message generated by attacker. No body else can act as if to be another authorized element to study any useful data. These attacks are hard to sense.

- **Non-repudiation**: The sent message by source can't be refused is ensured.

Following are some of the attacks on ad-hoc network:

- **Black Hole**: A malicious node is injected or inserted because of which bogus route requests' replies is received for the request sent, advertising it as having the shortest path to a destination [17]. Redirection of network traffic through malicious node is made up by false replies and is for eaves dropping. Denial of service which is a type of black hole attack attracts all traffic to it in order to execute DoS by dropping packets that are received.

- **Location disclosure**: The confidential necessities of an ad hoc network are attacker's target [18]. By using traffic analysis techniques, or by simple probing or monitoring, an attacker can find out the position of a node, or even the formation of the whole network it is known as location disclosure.

- **Blackmail**: For the recognition of malicious nodes the routing protocol that uses a method and messages are transmitted that try to blacklist the criminal this type of attack is seen [19]. The attack floods the network with fraud or forged route because of malicious node and packets are formed in order to use participating nodes' resources and disorder the genuine routes that are established. The sleep deficiency agony attack aims at the use of attacker that may formulate reporting message and try to separate nodes that are genuine from network.

Non-repudiation security feature can be helpful in such cases as it combines node for the generated message, and hence is also known as blackmail attack.

- **Wormhole**: One of the most commanding attacks is name as wormhole attack, since two malicious nodes that take part in the network; collaboration is involved between them [20]. Capturing routing traffic at one point of the network is done by one attacker, can be like node 1 and channeled to another node of the network; node 2, if they share a personal communication link with node 1. Then tunneled traffic back into the network is selectively introduced by node 2. The connectedness of the nodes that have created routes over the wormhole link is completely under the control of the two conspiring attackers. To get secured from wormholes we can use packet leashes.

- **Replay**: An attacker that performs a repeated attack into the network routing passage that has been previously captured. This attack usually targets the originality of routes.

- **Denial of Service**: Denial of service attacks is one of the most powerful attacks in ad-hoc network which aims at absolute interruption of the routing function and therefore the ad-hoc network perform intact operation [21]. Denial of service attacks' specific cases comprise of the overflow of routing table and the sleep deficiency agony. Overflow attack in a routing table overflow the malicious node of the network by formation of packets with fake route in order to consume participating nodes' resources and the establishment of genuine routes is interrupted. By continuously keeping node engaged in routing decisions the utilization of batteries of a specific node also increases and is known as sleep deprivation torture attack.

- **Routing Table Poisoning**: The table that holds information regarding routes of the network is preserved using routing protocols. The malicious nodes in poisoning attacks creates and sends fabricated signaling traffic, or some time alter messages that are genuine from supplementary nodes, for creating fake entries of the participating nodes in the tables [19]. For example, throw routing updates an attacker can attack and these updates do not correspond to real changes of the ad hoc network topology. The attack results in the formation of routing loops, making choice of non-optimal routes, bottlenecks and even partitioning of some network parts.

Two approaches can be followed for protecting mobile agent in ad-hoc network [22]:

a) **Reactive Approach**: First notice the threats and then respond properly.
b) **Proactive Approach**: Beware and try to prevent the information from being attacked by using some cryptographic technique.

Various types of attacks on different layers of OSI is summarized in Table 1.1[23].

Table 1.1: Each layer Security Attack on MANET

Layer	**Attacks**
Application Layer	Data Duplicity, Repudiation
Transport Layer	SYN Flooding ,Session Hijacking, Traffic Analysis, Monitoring, Disruption, Jamming, Interceptions, Eavesdropping
Network Layer	Resource Utilization, Wormhole, Black Hole
Data Link Layer	Examining ,Traffic Analysis, Interruption
Physical Layer	Jamming, Eavesdropping

Many security solutions are exists which resolves several security issues and are presented in Table 1.2[23].

Table 1.2: Security Solution for MANET

Layer	**Security Issues**	**Solutions**
Application Layer	Identifying and Intercepting Worms, Viruses, Application Abuses, Malicious Codes	Adequate Security Solution Firewalls **, IDS**, etc.
Transport Layer	Securing Point to Point Communication and Authenticating it by Encrypting	Adequate Security Solution Using Public Key

	Data.	**Cryptography**.
Network Layer	Preserving Ad-Hoc Network Routing and Protocol Forwarding.	For Source Authentication and Integrity of Message There is no Effective Mechanism but Some Routing Protocols can be used to Overcome Black Hole, Packet Leashes, etc.
Data Link Layer	The Wireless MAC protocol is Protected and Security is Provided to Support Link Layer.	For Preventing Traffic Analysis and Monitoring Layer There is no Efficient Mechanism.
Physical Layer	Signal Jamming, Denial of Service Attacks, are Prevented.	Using Spread Spectrum Mechanism.

The ongoing and very popular research work area for information safety is distributed intrusion detection system, but mostly Intrusion detection models made of single host and network examiner are used distributed IDS which also consist of centralized controller component also. The intrusion records are thrown to the centralized controller module by individual monitor that performs investigation on the received information from the other monitors. The primary concern of this kind of presented centralized systems is [24]:

- The intrusion detection and reaction real-time is not good.
- Single host duplicate with the composed facts, hence the supervised network is restricted. The network can get overloaded due to lots of data collection.
- Adding new hosts can overload the centralized controller and can significantly increase, hence IDS is non-scalable
- The system lacks dynamic configuration ability and also flexibility of the system is not up to mark.

- The cooperation between different IDS lacks. The combination of Network IDS(NIDS) and Host IDS(HIDS) is needed to used.

To detect and to prevent system and network from security threats Intrusion detection and prevention system(IDPS) is used which make computer network and system more secure and safe. For keeping information systems secure a valuable tool is Intrusion detection and prevention systems (IDPS) [25]. By using security tool like Intrusion detection and prevention systems (IDPS) we can examine, analyze, and act in response to violations in possible security against computer and network systems. Because of unauthorized external intruders the result of break in attempts are trying to compromise the system or miss-using the authority of internal privileged users. IDPS continue to develop and make new systems.

Detection methodologies that can be utilized by newer systems create some confusion when we try to better understand it. Previous and existing area of work primarily focuses on improvising one or more methodologies. Assessment of one method offers some work besides of the proposed methodology [25].

1.5 OBJECTIVE OF DISSERTATION

The primary concern of our dissertation is to provide secure communication and information transfer using mobile agent. This can be done either by preventing information carried by agent or by tracing packets.

Following are the objective needed to be achieved:

- Provide communication using mobile agent technology.
- Provide secure environment for information transfer.
- Prevent agents' from being attacked in hostile environment.
- Preventing information from being hacked using Intrusion detection methodology.
- Securing information data using ciphering techniques.

1.6 RESEARCH METHODOLOGY

1. Literature Survey with survey papers.

2. Installation and configuration.

3. Setup of simulation environment.

4. Designing mobile agent communication model.

5. Ciphering of agents' information using suitable algorithm.

6. Intrusion Detection on data packets in ad-hoc network for wireless agents.

7. Intrusion Detection on mobile agents.

8. Result Analysis and conclusion.

1.7 ORGANIZATION OF DISSERTATION

Chapter 1 gives a brief introduction about MANET, Mobile agent and its working, Client server, comparison between mobile agent and client server architecture, various threats to mobile agent in ad-hoc network. We have also discussed threats depending upon OSI layers, and some solution to them. A brief description of Intrusion detection system and security solution to mobile agent's information is primary concern for our work.

Chapter 2 provides the survey of various techniques used for detection and protection of agent and its information. This chapter gives information of some techniques, system and tools that have been used for our work.

Chapter 3 describes the experimental setup, installation and implementation details to achieve desirable objectives like, agents' information ciphering, Intrusion detection and packet dropping.

Chapter 4 describes the usage of encryption and SNORT technology and provides the results pertaining to data encryption, intrusion detection and dropping based on signatures.

Chapter 5 concludes the work and provides future scope.

CHAPTER 2

LITERATURE SURVEY

Mobile agent technology have provided several distributed applications that got fundamental changes over traditional communicational approaches like RMI, RPC, etc. that induce infrastructure of network to alter it. Security is the biggest concern which darkens the advantageous side of mobile agent infrastructure. As soon as the mobile agent migrates from home platform, it goes out of the control of its owner that gives the opportunity to attacker (eavesdropper, network sniffer, execution environment, etc.).

2.1MOBILE AGENT COMMUNICATION PARADIGM

1. **IBM-Aglet or Aglet Software Development Kit (ASDK)**

IBM-Aglet or Aglet Software Development Kit is one of the environments for developing JAVA based mobile agents [52]. These toolkits are open source freely available; Aglet 2.5 alpha is the latest version.

Good Graphical user interface is provided for development of agent. It primarily consists of two packages- Workbench of Aglet and Aglet Building Environment (ABE).

2. **Voyager**

An agent development tool developed by ObjectSpace named as Voyager, in mid- 1996 [53]. Voyager is one of the commercial products of ObjectSpace that has taken over by Recursion Software Inc. since 2001.

3. **Anchor**

Lawerence Berkeley National Laboratory, U.S.A. developed Anchor agent toolkit that facilitates secure management and transmission of mobile agents in a heterogeneous distributed environments [54]. This toolkit is available in BSD style license.

4. **Zeus**

Advanced Applications & Technology Department of British Telecommunication labs developed Zeus, an integrated environment for the rapid development of collaborative agent applications [55, 56]. This toolkit is open source and freely available. It is compatible with most hardware platforms as it is purely implemented in Java. It can be compiled with FIPA (Foundation for Intelligent Physical Agents) standards.

5. **JADE**

Tilab developed JADE (Java Agent Development Framework) for multi agent applications for peer to peer communication architecture. It's a software framework that is fully designed and implemented in Java language [52, 57, 58]. Multi-agent systems simplification has been done by the use of a middle-ware that complies with the latest Foundation for intelligent physical agents (FIPA) 2000 specifications. For debugging and deployment phases of agent development a set of graphical tools is provided which is supported by it.

Table 2.1: Comparison of Various Toolkits.

Comparison of Agent Development tool kits / **Features**	**Aglet**	**Voyager**	**JADE**	**Anchor**	**Zeus**
Production nature	Open Source and freely available	Commercial	Open Source and freely available	BSD license Available	Open Source and freely available
Implementation Standards	MASIF	- - - - - -	FIPA	X. 509	FIPA
Technique for Communication	Both Asynchronous and Synchronous	Every method	Asynchronous	Asynchronous	Asynchronous

Security Technique	Poor	Weak	Good	Very Good	Good
Mobility of Agent	Weak	Weak	Good	Weak	Don't Support
Agent Migration	Socket	RMI	RMI	Socket	Null

In the case, if the execution environment is malicious, it can perform various attacks on mobile agent. New security technology implementation is needed to be introduced that can prevent hampering of mobile agent's data/code. Most of the threats that a mobile agent can have because of un-trusted execution environment are:

- Analysis of code to change its execution behavior.
- Modification of code.
- Analysis of data collected during execution of agent's itinerary.
- Modification or deletion of data collected.

Various the approaches used to protect an agent can be broadly classified into two main mechanisms:

1. Detection mechanism attempts to detect unauthorized modification of code, state or execution of mobile agent.

2. Prevention mechanisms try to make it impossible to access or modify code, state or data of mobile agent.

2.2THREATS DETECTION MECHANISM ON MOBILE AGENT

Table 2.2: Intrusion Detection Techniques

Author	Year	Approach
Sinha and Chaki [26]	2014	CRT based Routing topology for MANET
Bhati *et al.,*[27]	2011	Agent Based AODV Routing protocol in MANET
Halim and Tharwat [28]	2010	Agent-Based Trusted Dynamic Source Routing (ATDSR)
Bindhu [1]	2010	Mobile Agent Based Routing Protocol with Security for MANET
Dang *et al.,*[29]	2010	DASR (Distributed Anonymous Secure Routing)

Narjes *et al.,*[30]	2009	Sedentary Agent Approach
Mitrokotsa *et al.,*[31]	2006	Intrusion Detection System with multiple local IDSs agents
Oscar *et al.,*[32]	2003	Traceability Techniques: Watermarking and Fingerprinting
Kachirski and Guha [33]	2003	Distributed Multi-Sensor Intrusion Detection System
Albers *et al.,*[34]	2002	Collaborative work using Simple Network Management Protocol SNMP
Smith [35]	2001	An Architecture that consist to two parts: IDS and Secure database
Jansen [36]	2000	Agent's itinerary is recorded and tracked

Sinha and Chaki (2014) has proposed AESCRT , *i.e.,* Agent enabled secure CRT based routing topology for MANET. This methodology has three types of agents named fants, bants and trust agent and these agents are used to detect secure routes amongst set of available routes. The final route is detected using a WANT agent. This routing protocol consists of three phases:

(i) Trust Value Assignment
(ii) Probable Route Detection
(iii)Final Agent Based Secure Route Detection Scheme

During Trust Value Assignment phase each node is assigned a trust value by the monitor. The initial value is set to zero. Successful transmission of message increments the trust value, if its trust value remains same, monitor declares the node as malicious and informs other neighbors about the status of malicious node.

In Probable Route Detection phase the source node broadcasts RREQ message to all its neighbors. The message is forwarded and it continues till destination node is found, when destination is found ACK message is sent back.

Finally in last phase for secure transmission encryption and CRT schemes are used. The algorithm combines the mobile agents' concept with that of CRT for providing secure environment in MANETs, combination of RSA and CRT schemes for key generation.

Bhati *et al.,* (2011) proposed an efficient Agent-Based AODV Routing Protocol in MANET that tries to remove the existence of misbehaving nodes that may slows down the routing

operation in MANET. This increases the efficiency of a network. Efficiency can be calculated by the parameters or factors such as transmission capacity, battery power and scalability. Here the most crucial factor named as transmission capacity of a node is considered. In MANET, as the network size increases complexity of a network also increases. This proposed protocol provides the most efficient and reliable route which may or may not be minimum hop count. The proposed scheme uses mobile agent that can move in the ad-hoc network to discover the network topology and collects or updates the routing table. Also it will evaluate transmission capacity value for each node. This would be performing by the fusing of mobile agent at every node which computes routing information. The static agent that runs in a host node supplies required information to the visiting mobile agent. Therefore the proposed protocol uses a Monitoring Agent System (MAS) and Routing Agent System (RAS) to achieve the task.

Halim and Tharwat (2010) proposed a Protocol for MANETs named Agent-Based Trusted Dynamic Source Routing (ATDSR) which was designed to manage trust information locally with minimal overhead in terms of extra messages and time delay. This objective is achieved through installing in each participating node in the network a multi-agent system (MAS). MAS consist of two types of agents: monitoring agent (MOA) and routing agent (ROA). A new mathematical and more realistic objective model for measuring the trust value is introduced in this protocol. This model is weighted by both number and size of routed packets to reflect the "selective forwarding" behavior of a node.

Bindhu (2010) proposed Mobile Agent Based Routing Protocol with Security for MANET which is a routing algorithm with multiple constraints proposed based on mobile agents. It uses mobile agents to collect information of all mobile nodes. The algorithm has stronger routing stability and lower probability of link failure because it selects links with large link expiration time.

Dang *et al.,* (2010) introduced a protocol named DASR (Distributed Anonymous Secure Routing) which included good scalability for Mobile Ad Hoc Networks. A new efficient distributed anonymous secure routing protocol (DASR) achieved routing and data transmission using dynamic identity pseudonyms of the nodes instead of their real identities. Trust Based

Secure Routing in AODV Routing Protocol [16] proposes modified AODV routing protocol with node trust value. It required the following modification in the existing AODV[17] protocol;

(i) Two new control packets TREQ (Trust Request), TREP (Trust Reply),
(ii) Modified extended routing table with four new fields; positive events, negative events, route status, opinion.

Using this approach, secure route can be established by calculating trust value of each node which is participating in the route establishment process from source to destination.

Narjes *et al.,* (2009) proposed clone agent protocol which is somehow related to sedentary agent approach, but here the clone agent executes on the same execution host before critical code execution. Thus, instead of exposing the mobile agent to the malicious host, it first sends a clone to investigate the behavior of the execution environment. In CAP, once an attack is detected, the mobile agent puts the responsible host in its malicious host's list and changes its destination accordingly. In this approach, original task of agent is performed after the execution of clone agent which verifies the authenticity of execution environment. The overall process creates unnecessary delay in the execution of original task.

Mitrokotsa *et al.,* (2006) proposed a distributed model. The proposed intrusion detection system is composed of multiple local IDSs agents. Each IDS agent is responsible for detecting possible intrusions locally. The collection of all the independent IDS agents forms the IDS system for the mobile wireless ad hoc network. Each local IDS agent is composed of the following components:

Data Collector: Responsible for selecting local audit data and activity logs.
Detection Engine: Responsible for detecting local anomalies using local audit data. The local anomaly detection is performed using the eSOM classification algorithm.
The procedure that is followed in the local detection engine is the one described below:

Select labeled audit data and perform the appropriate transformations. Compute the classifier using training data and the eSOM algorithm. Apply the classifier to test local audit data in order to classify it as Normal or Abnormal. Response Engine: If an intrusion is detected by the

Detection Engine then the Response Engine is activated. The Response Engine is responsible for sending a local and a global alarm in order to notify the nodes of the mobile ad hoc network about the incident of intrusion. Special attention should be paid on the function of the Response Engine in order to avoid possible flooding caused by the notification messages of intrusion. Thus, the broadcasted notification of intrusion is restricted to a few hops away from the node where the anomaly has been detected since the neighboring nodes run the greatest risk of possible intrusion. When the Response Engine is activated, the node fires a fake RTS (Ready to Send) message destined to the suspicious node. If the suspicious node replies to that packet then the node is classified as malicious. Otherwise, the node fires an AODV ERROR message as the suspicious node is indicated as moved. After the discovery of the adversary the local IDS agent fires an ALERT message notifying its one hop neighbors. Alternatively, the local IDS agent could send ALERT messages to all potentially traffic generators that exist in its routing table, thus achieving a global response to all nodes that are directly influenced by the malicious node.

Oscar *et al.,* (2003) have proposed two traceability techniques that are watermarking and fingerprinting. In these techniques a mark is embedded into agent and the agent's execution creates marked results. When an agent returns to its origin host, these results are examined. If the mark has changed or has disappeared, this means that the executing host has modified the agent. In agent's watermarking scheme, the mark is embedded into mobile agent's code because all executing hosts in the agent's itinerary must run the same marked code. But in agent's fingerprinting scheme, the embedded mark is different for each host because mark is embedded into agent's data and data is usually different for each host.

The main advantages are that these techniques are used to detect manipulation attacks performed during agent's execution and also trace the malicious host responsible for the manipulation attacks. The main advantage of mobile agent's fingerprinting technique over watermarking technique is that it avoids collusion attacks performed by a group of dishonest users.

The main drawbacks of these techniques are increase in its code and data size because embedding a mark always means that some overhead is added to the mobile agent. Moreover, a TTP (Trusted third party) is needed in order to punish malicious behavior.

Kachirski and Guha (2003) proposed algorithm of distributed multi-sensor intrusion detection system based on mobile agent technology. In this algorithm the system has main three modules, each has their own work of mobile agent with some functions that include: monitoring, decision making or initiating a response. These functional tasks divide into categories and assigning different agent. Monitoring agent: Network monitoring and Host monitoring are done by the agents of this class. A host-based monitor agent hosting system-level sensors and user-activity sensors is run on every node to monitor within the node.

A monitor agent with a network monitoring sensors run only on some selected nodes to monitor at packet-level to capture packets going through the network within its radio ranges.

Action agent: Every node also hosts this action agent. Since every node hosts a host-based monitoring agent, it can determine if there is any suspicious or unusual activities on the host node based on anomaly detection. When there is strong evidence supporting the anomaly detected, this action agent can initiate a response, such as terminating the process or blocking a user from the network.

Decision agent: The decision agent is run only on those nodes only which run network monitoring agents. These nodes collect all packets within its radio range and analyze them to determine whether the network is under attack. Moreover, if the local detection agent not able to make a decision on its own due to unsatisfactory evidence, it reports to the decision agent for investigate further. This is done by using packet-monitoring results that comes from the locally running network monitoring sensor. If the decision agent concludes that the node is malicious, the action module of the agent running on that node will carry out the response. The network is logically divided into clusters with a single cluster-head for each cluster. This cluster head will monitor the packets within the cluster whose originators are in the same cluster are captured and investigated. This means that the network monitoring agent and the decision agent run on the cluster-head. In this mechanism, the decision agent performs the decision making based on its own collected information from its network-monitoring sensor; thus, other nodes have no influence on its decision. This way, spoofing attacks and false accusations can be prevented.

Albers *et al.,* (2002) proposed a system uses the collaborative work of mobile agents running

on different nodes to make up for the complete intrusion picture. The architecture depends on the advantages offered by the Simple Network Management Protocol SNMP. Data used are those stored in the Management Information Base MIB of SNMP. Since SNMP uses UDP for communication, mobile agents are used to send requests to remote hosts to overcome the unreliability of UDP.

Various collecting agents work together in LIDS in following manner:

- Local LIDS Agent: does local intrusion detection (misuse or anomaly) and response, and reacts to intrusion alerts by other nodes. As soon as a local LIDS detects an intrusion, it updates the other nodes of the network.

- Mobile Agents: transport SNMP requests to remote hosts to overcome the unreliability of SNMP message transfer over UDP. LIDS can hand over a specific task to a mobile agent that it will achieve in an autonomous manner without any help from its LIDS.

Smith (2001) proposed an architecture that consists of two parts: mobile IDS agents which run on every node, and a stationary secure database that contains global signatures of known misuse attacks and stores patterns of each user's normal activity in a non-hostile environment. The IDS mobile agent's responsibility is to detect intrusions based on local audit data and participate in cooperative algorithms with other IDS agents to decide on attacks.

Each agent consists of five parts: a local audit trail to collect audit data; local intrusion database to warehouse the necessary information for the IDS agent; secure communication module to enable different IDSs communication; anomaly detection modules to detect different types of anomaly; and misuse detection modules to detect different types of signatures. On the other hand, the stationary secure database acts as a secure trusted repository for the mobile nodes. Mobile nodes use this database to obtain information about the latest misuse signatures and find the latest patterns of normal user activity.

Jansen (2000) has proposed a general technique that allows an agent's itinerary to be recorded and tracked by another cooperating agent and vice-versa, in a mutually supportive arrangement. When moving between agent platforms, an agent conveys the last platform, current platform, and

next platform information to the cooperating peer through an authenticated channel. The peer maintains a record of the itinerary and takes appropriate action when inconsistencies are noted. Attention is paid in this scheme so that an agent avoids platforms already visited by its peer.

The main advantages of this technique are that by dividing up the operations of the application between two agents, certain malicious behavior of an agent platform can be detected. Moreover, this scheme can be incorporated into any appropriate application.

The main drawback of this technique includes the cost of setting up the authenticated channel and the inability of the peer to determine which of the two platforms is responsible if the agent is killed.

2.3THREATS PROTECTION MECHANISM ON MOBILE AGENT

Table 2.3: Various Threats Protection Technique

Author	Year	Approach
Srivastava and Nandi [37]	2013	Self reliant protocol.
Armogum and Caully [38]	2011	Code confidentiality using Obfuscation Technique.
Venkatesan *et al.,* [39]	2010	Encryption of hash value using Digital Signature.
Esfandi and Rahimabadi [40]	2009	Multi-Agent based Cryptographic protocol and data divided into different parts.
Ahmed [41]	2009	Secure-Image Mechanism (SIM)
Zaslavsky *et al.,*[44]	2007	Self-Executing Security Examination (SENSE)
Ametller *et al.,*[45]	2004	Agent Driven approach, Security verification by agent itself.
Zaslavsky *et al.,*[46]	2003	Self-Reliant Security approach.
Karnik and Tripathi [47]	2000	Security Manager that controls agent's access.
Yee [48]	1999	Partial Result Authentication Codes (PRAC) which do checksum with the help of secret key.
Riodan and Schneier [49]	1988	Environmental Key Generation.
Sander and Tschudin [50]	1998	Encryption Function.
Hohl [51]	1998	Blackbox Security

Srivastava and Nandi (2013) introduced a security protocol for the protection of mobile agent from various types of attacks. This protocol is built on the foundation of self protection approach

based on agent driven security and integrity based confidentiality of mobile agents. The self-protection is instituted to make mobile agent less interactive during its execution. The idea of symmetric key's component distribution was adopted in which a key component is distributed in secure manner while the other key component is derived from ensuring integrity of data collected at run time. However, the concept of self protection of agent may not be useful in a situation where an application is agents' interaction specific. That is, an application environment where the interaction of mobile agents is the ultimate for specific problem solving.

An agent code obfuscator is used by Armogum and Caully (2011) along with three different obfuscation techniques combined to provide code confidentiality. Java obfuscation only provides execution privacy or confidentiality to the mobile code but it cannot be ensured that the particular obfuscated code is de-obfuscatable or not. In this case, if execution platform has sufficient computational capacity, it can de-obfuscate the code before executing the code. This technique is a preventive measure and its security is dependent upon the computational capacity of execution environment. Code confidentiality is an important security requirement for software agent applications involving mobility , *i.e.,* where agent has to migrate to other servers (assumed un-trusted) on the network to be able to achieve its goal; typically in e-commerce application where agent may be equipped with a decision making logic and/or payment information. Thus code obfuscation may provide code confidentiality for some time, which in most cases may be enough for the agent to compute on server and move on.

According to Venkatesan *et al.,* (2010) when mobile agent executes in a malicious host environment then the behaviour of agent turns malicious which may be dangerous for next host platform and suppose this platform detects that this is malicious agent then it will think the originator of this agent is responsible for this while it is not true. For the protection of agent code, Venkatesan proposed root canal algorithm in which there are two modules. One is for agent owner and another one is for remote platforms contained in the itinerary of agent. This approach is like digital signature approach and following steps are taken to perform it:

- First agent owner generates the hash value of agent byte code. Encrypt the hash value with its own private key (Digitally Signed).

- Append digitally signed hash value of agent code with original agent code.
- At the remote site, remote host get the agent' code with encrypted hashed value.
- Remote hosts decrypt hashed code with the help of public key of agent owner.
- Remote host calculate hash value of agent of agent code and compare it with appended digitally signed hashed code and identify the alteration if it occurs.

According to this approach, if succeeding host finds any malicious change then it claims the preceding host for alteration. In the case, when malicious host changes the code and claims that this change is performed by proceeding legitimate host then in this case, root canal algorithm is not suitable.

Esfandi and Rahimabadi (2009) proposed multi-agent based cryptographic protocol wherein encryption key and data are broken into different parts carried by different mobile agent so that it is necessary for an intruder to compromise all agents to crack the security. This protocol seems to provide security but at the cost of communication and computational overhead which comes in terms of encryption, authentication as well as communication overhead. In case of 'n' different mobile agents, 'n' time security checks is applied at each host to provide verifiable security which is not feasible in case of light weighted network infrastructure.

Beside overhead problem, this protocol does not provide fault tolerance capability to the entire communication process due to the multi-agent based security. In any occurrence of fault, if any of the agents do not reach at the destination, the whole communication process fails due to availability factor.

Ahmed (2009) has introduced a new mechanism called Secure-Image Mechanism (SIM) to protect the mobile agent from malicious hosts. Figure 2.1 shows generation of secure image. In this technique, secure image controller creates the image of mobile agent and sends this to unknown host for execution. Host executes this image and sends it back to the secure image controller where verification of agent's integrity takes place by comparing it with original copy of the agent. If it finds any malicious modification, SIM presumes the host to be malicious. This

mechanism provides security but takes much time and creates unnecessary communication overhead in network.

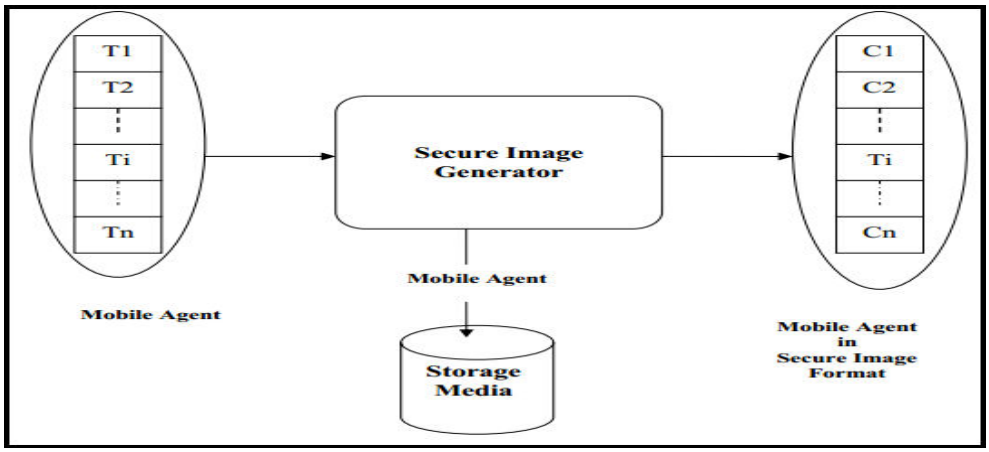


Figure 2.1: Generation of secure image [41]

Public key infrastructure based model by Ismail (2007); Saxena (2005) where most of the cryptographic based model is supported by PKI based model. In public key infrastructure there is a certification authority who registers the objects' public key and issues digital signature which includes identity and public key of objects therefore, each agent platform has a public private key pair and agent platform registers its public key and obtains a digital certificate from certification authority after getting authentication with certification authority Self-reliant security approach.

Zaslavsky *et al.,* (2004) proposed an approach of self-executing security examination (SENSE) for the protection of mobile agent from malicious execution environment. In this approach, agent is self-reliant with respect to its security function and executes in an unknown environment without having dependency on an external support for security. Agent carries scan algorithm which scans the agent's code and verifies the integrity of agent at random intervals. Continuous execution of scan algorithm degrades the performance of the approach. In addition, this approach did not give the focus on execution privacy issue of mobile agent.

Ametller *et al.,* (2004) gave a breakthrough idea for securing mobile agent to build self protected mobile agent. In their scheme, agents carry their fully protected encapsulated

protection mechanism to protect their code and data. After this innovative breakthrough in the area of agent security, various researchers used it as foundation and gave their idea for agent based application. This technique promotes agent driven approach, *i.e.,* security verification is performed by agent itself and it also uses a cryptographic interface between agent and execution platform so that platform could not directly interact with the agent. The need for agent to gain access to the private key of the execution platform is one point which makes this approach sometime inferior.

According to this scheme, agents are internally structured as follows:

- All executable agents' code wrapped up as data and encrypted with platform's public key.
- Explicit controller code C transports executable code. Thus mobile agent is structured as a combination of (C,D) where C acts as a controller code and D wrapped as data.
- Now, first agent owner generates public private key pair (P_a,S_a) and digitally sign the executable code d_j and create signature $S_j=E_{Sa}(H(d_j))$.
- After that, agent owner encrypts explicit executable code d_j and its signature S_j using pubic key of jth platform and creates encapsulated agent $D_j=E_{Kj}(d_j,E_{Sa}(h(d_j))),H(C)$.
- At the receiver end, using platform public decryption function, receiver extracts (d_j, S_j) from D_j. Before executing d_j first the receiver verifies the authenticity and integrity of d_j with the help of signature S_j then executes the code d_j.

The major drawbacks of this technique are its high computational complexity. Each receiver (agent platform) uses platform decryption function as cryptographic service acting as an inter-face between the interaction of agent and agent platform. Decryption of mobile agent using platform's private key is the main downside of this approach. So to overcome the problem of security and computational complexity, we develop a new self-protection protocol with novel features of integrity based encryption as well as secure symmetric key's component distribution.

Zaslavsky *et al.,* (2003) had brought forward a detailed concept of self-reliant security features. In their approach, agents carry only an indication of the actual message that resides on

the secure server instead of any sensitive data. On receiving the indicator digest, destination host (agent platform) sets up a secure communication channel for retrieving complete message from the secure server. In this technique, each destination host (during itinerary of mobile agent) needs to setup a secure communication channel to retrieve the original message which creates huge amount of security overhead.

Although, indicator digests does not contain whole message, but it should be secure in communication channel as well as on execution environment as it locates the original message and the address of secure server. In this scenario, malicious entity may perform denial of service (DoS) by capturing the indicator digest to analyze and modify its context so that destination host (execution agent platform) will not be able to extract original message.

Karnik and Tripathi (2000) proposed a security manager which is an extension to Java's RMI security Manager, whose function is to control agents' access to system resources, using access control list and the access is granted on the basis of identity of agent's owner. The security manager grants access permission to agent only if it's an authorized one. This system uses an approach for protecting the mobile agent. The approach consists of three mechanisms, the first is to allow the programmer to declare parts of the mobile agent state as Read-Only and if any modification occurs to these parts, the mobile agents' owners can detect it by using the digital signature mechanism. The second mechanism is let the mobile agent create append-only data states container where the data stored in this container cannot be deleted or altered without detection by the mobile agent's owner. The third mechanism is to let programmers to define data states to specific hosts and no other hosts can deal with these data states. These mechanisms use the encryption, the decryption and the digital signature. This system uses an approach for protecting the mobile agent. The approach consists of three mechanisms, the first is to allow the programmer to declare parts of the mobile agent state as Read-Only and if any modification occurs to these parts, the mobile agents' owners can detect it by using the digital signature mechanism. The second mechanism is let the mobile agent create append-only data states container where the data stored in this container cannot be deleted or altered without detection by the mobile agent's owner. The third mechanism is to let programmers to define data states to

specific hosts and no other hosts can deal with these data states. These mechanisms use the encryption, the decryption and the digital signature.

Yee (1999) proposed Partial Result Authentication Codes (PRAC) where an agent can protect its information by using Partial Result Authentication Codes, which are cryptographic checksums formed using secret key cryptography (*i.e.,* message authentication codes). This technique requires the agent and its originator to maintain or incrementally generate a list of secret keys used in the PRAC computation. Once a key is applied to encapsulate the information collected, the agent destroys it before moving onto the next platform, guaranteeing forward integrity. The forward integrity property ensures that if one of the servers visited is malicious, the previous set of partial results remains valid. However, only the originator can verify the results, since no other copies of the secret key remain. As an alternative, public key cryptography and digital signatures can be used in lieu of secret key techniques. One benefit is that authentication of the results can be made a publicly verifiable process at any platform along the way, while maintaining forward integrity.

The PRAC technique has a number of limitations. The most serious occurs when a malicious platform retains copies of the original keys or key generating functions of an agent. If the agent revisits the platform or visits another platform conspiring with it, a previous partial result entry or series of entries could be modified without the possibility of detection. Since the PRAC is oriented toward integrity and not confidentiality, the accumulated set of partial results can also be viewed by any platform visited, although this is easily resolved by applying sliding key or other forms of encryption.

Riodan and Schneier (1998) proposed Environmental Key Generation that describes a scheme for allowing an agent to take predefined action when some environmental condition is true. The approach centers on constructing agents in such a way that upon encountering an environmental condition (*e.g.,* via a matched search string), a key is generated, which is used to unlock some executable code cryptographically. The environmental condition is hidden through either a one-way hash or public key encryption of the environmental trigger. The technique ensures that a platform or an observer of the agent cannot uncover the triggering message or response action by directly reading the agent's code. The procedure is somewhat akin to the way in which modern

operating systems apply passwords to determine whether login attempts are valid (*i.e.,* the password is used as a cryptographic key). One weakness of this approach is that a platform, which completely controls the agent, could simply modify the agent to print out the unlocked executable code upon receipt of the trigger, instead of executing it. Another drawback is that an agent platform typically limits the capability of an agent to execute code created dynamically, since it is considered an unsafe operation. An author of an agent can apply the technique, however, in conjunction with other protection mechanisms for specific applications on appropriate platforms.

Sander and Tschudin (1998) proposed Encryption Function whose goal is to determine a method whereby mobile code can safely compute cryptographic primitives, such as a digital signature, even though the code is executed in un-trusted computing environments and operates autonomously without interactions with the home platform. The approach is to have the agent platform execute a program embodying an enciphered function without being able to discern the original function; the approach requires differentiation between a function and a program that implements the function.

For example, Alice has an algorithm to compute a function *f*. Bob has input *x* and wants to compute *f .x.* for Alice, but she doesn't want Bob to learn anything about *f*. If *f* can be encrypted in a way that results in another function *E. f.* then Alice can create a program *P.E. f ..*; which implements *E. f .*; and send it to Bob, embedded within her agent. Bob then runs the agent, which executes *P.E. f .* on *x*, and returns the result to Alice who decrypts it to obtain *f .x.*: If *f* is a signature algorithm with an embedded key, the agent has an effective means to sign information without the platform discovering the key.

Similarly, if it is an encryption algorithm containing an embedded key, the agent has an effective means to encrypt information at the platform. Although the idea is straightforward, the trick is to find appropriate encryption schemes that can transform arbitrary functions as intended. The technique has been shown to be useful to encrypt rationale functions, and whether more general functions can be encrypted in a similar fashion remains to be seen. The technique, while very powerful, does not prevent denial of service, replay, experimental extraction, and other forms of attack against the agent.

One of the major drawback of Encrypted function and environmental key generation is the agent owner limits the capability to execute a mobile agent code that iterated dynamically; hence it is considered as unsafe operation.

Hohl (1998) proposed a preventive technique, in which mobile code is scrambled by some obfuscation mechanism so that no host is able to analyze its functions easily. This feature is known as black box security. Obfuscation is a mechanism which provides security to mobile code so that it cannot be analyzed and modified by malicious host.

He gives a detailed overview of the threats stemming from an agent encountering a malicious host as motivation for Blackbox Security. The strategy behind this technique is simple-scramble the code in such a way that no one is able to gain a complete understanding of its function to modify the resulting code without detection. A serious problem with the general technique is that there is no known algorithm or approach for providing Blackbox protection.

However, he shows Computing with Encrypted Functions as an example of an approach that falls into the Blackbox category, with certain reservations concerning the limited range of input specifications that apply. A time limited variation of Blackbox protection is introduced as a reasonable alternative, whereby the strength of the scrambling does not necessarily imply encryption as with the unqualified one, but relies mainly on obfuscation algorithms. Since an agent can become invalid before completing its computation,

Obfuscated Code is suitable for applications that do not convey information intended for long-lived concealment. One promising method relies on a trusted host to trigger the execution of an agent's code segment. It is not strictly speaking a pure obfuscation algorithm, however, since code is redistributed to a trusted host, which is not part of the originally proposed scheme.

The method does suggest a possible relationship between Environmental Key Generation and Obfuscated Code techniques. A serious drawback to this technique is the lack of an approach for quantifying the protection interval provided by the obfuscation algorithm, thus making it difficult to apply in practice. Furthermore, no techniques are currently known for establishing the lower bounds on the complexity for an attacker to reverse engineer an agent's code.

One of the Major drawback of this methodology is no efficient algorithm is available for reverse engineering. It is also possible to send the virus code in obfuscated form, which is not able to detect.

2.4 ISSUES INDENTIFIED IN LITERATURE SURVEY

List of various issues that we have seen above in our survey are as follows:

- Mobile agent travels in ad-hoc environment where attack ratio is very high.
- No reliable secure environment available for agent to migrate.
- Single level security limits security level.
- Black-box security is one of the finest securities available but it doesn't have efficient reverse engineering methodology.
- Lack of prior knowledge due to various hardware platforms, running different operating system.
- Intrusion detection methodology lacks efficiency.
- Intrusion detection has limited upgradability of detection techniques.

2.5 CONCLUSION

We came across through literature survey about many limitation and security issues that make information carried by mobile agent unsecure. In our dissertation work we will try to detect and prevent our data from being attacked.

For our work we will try to give protection at both; transport layer as well as application layer, to provide system better and quality security that will overcome the limitations of previous methodology.

CHAPTER 3

PROPOSED SOLUTION

Development of agents as well as their wide usage requires good underlying infrastructure. Literature survey indicates scarcity of agent development tools in initial years of research which limited the exploitation of this beneficial technology. However, today a wide variety of tools are available, for developing robust infrastructure.

3.1PROPOSED AGENT BASED COMMUNICATION MODEL

We have proposed a secure communication model for mobile agent data and information transfer in Figure 3.1. In our dissertation work double layered security is provided. This can be done by securing information of agent using encryption technique thus preventing transport layer and further providing intrusion detection so that by matching signature malicious attacks can be first detected and then prevented. Detailed discussion about methodology is seen in rest part of chapter.

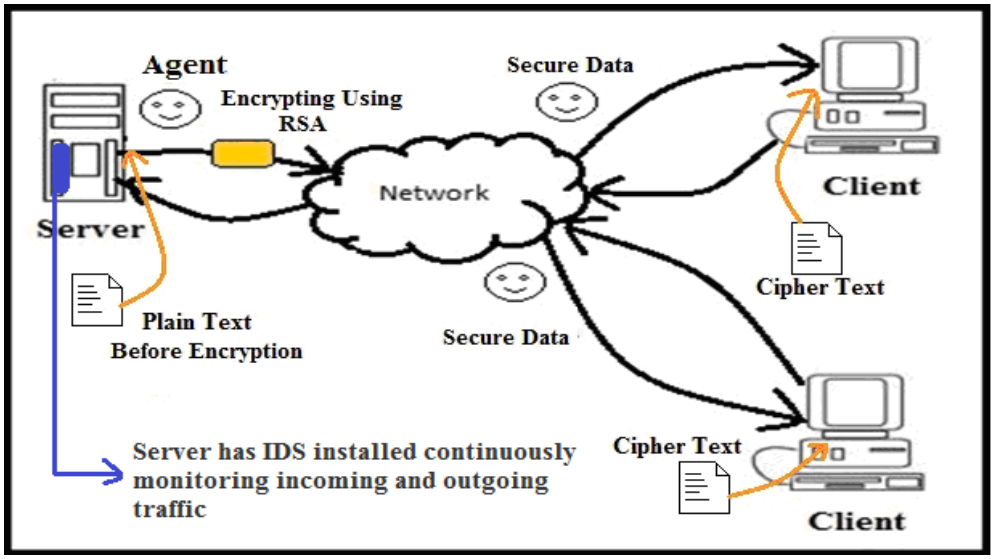


Figure 3.1: Agent Communication with Security Protocol and IDS

Above figure shows that information of agent is firstly encrypted by using ciphering technique and then that encrypted data migrates from one place to another and each packet is detected for

intrusion detection, and if malicious activity or packet found then the packet is dropped using ips. We are using JADE platform for agents' communication and SNORT for intrusion detection.

3.2 AGENT DEVELOPMENT IN JADE TECHNOLOGY

The middleware that facilitates the evolution of multi-agent systems is called JADE. The JADE agents exist within a runtime environment, to develop agents the programmers are provided with a library of classes that they can use, monitoring and administrating is done by use of a suite of graphical tools that allows management and running of agents. Sometimes artificial intelligence is also involved by agents to make them more intelligent. JADE defines an agent platform with three mandatory agent services [59]. These services include autonomy, mobility and reactivity. All agents in JADE are design and kept in a repository called container. Host address and source address both are contained within agent, so that agent can migrate from one host to another. Agent has buffer to store data information and has unique identity for naming convention. Agents in JADE exist within a runtime environment Figure 3.2 below shows JADE architecture.

JADE offers a list of features to the agent programmer [59]:

- JADE Agent Platform, includes the AMS (Agent Management System), the DF (Directory Facilitator), and the ACC (Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up. Management of agent is done by DF and AMS.

- Distributed agent platform that can split on several hosts. For single Java application, single Java Virtual Machine, is executed on each host. Agents are implemented as one Java thread and Java events are used for effective and light-weight communication between agents on the same host. JADE does scheduling for these tasks in a more efficient compared JVM for threads.

- A number of DFs (Directory Facilitator) can be started at run time in order to implement multi-domain applications.

- Transport mechanism and interface to send/receive messages to/from other agents.
- Consist of IIOP protocol to connect different agent platforms.
- Automatic registration of agents with the AMS.
- The start-up agents obtain their GUID (Globally Unique Identifier) from the platform.

Graphical user interface is used to manage several agents and agent's platform from the same agent. The activity of each platform can be monitored and logged. Following Figure 3.2 shows detailed architecture and functionality of JADE.

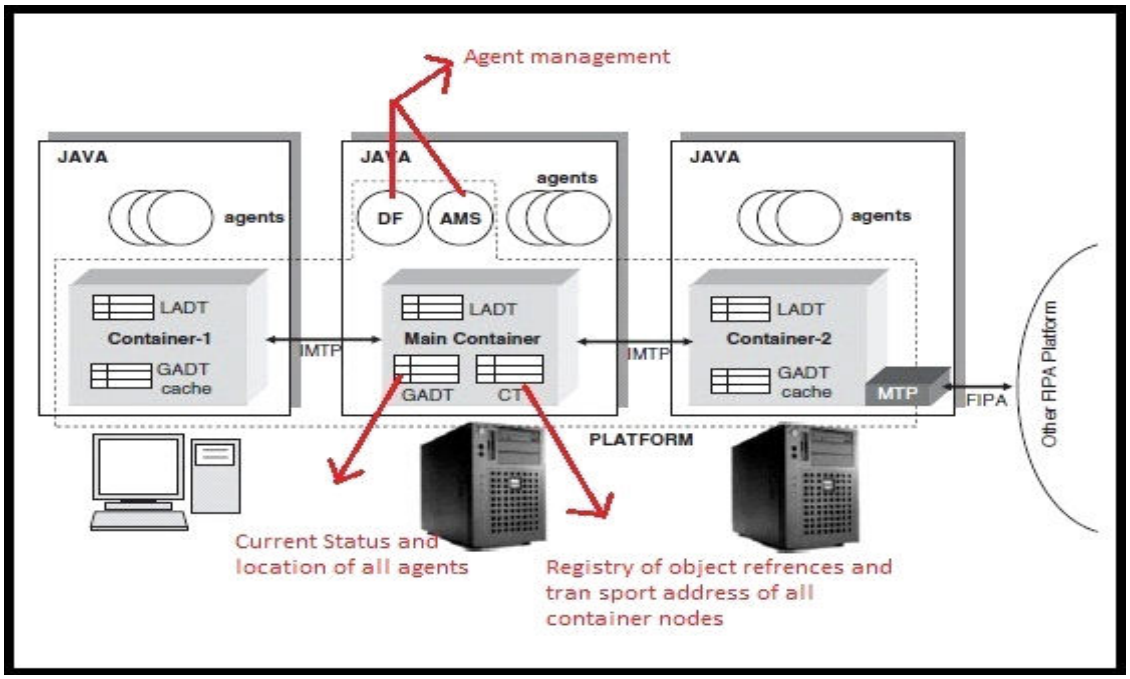


Figure 3.2: Jade Architecture [59].

3.3 INTRUSION DETECTION SYSTEM (SNORT)

The ongoing and very popular research work area for information safety is distributed intrusion detection system, but mostly Intrusion Detection prototypes that are single host and network examiner are used for distributed IDS which also consist of centralized controller component. The intrusion records are thrown to the centralized controller module by individual monitor that performs investigation on the received information from the other monitors. Basic

configurations needed for IDS/IPS are Data Acquisition API, libdnet, Snort, Snort rules. Some of the following dependencies are built for Intrusion Detection and Prevention System (IDPS):

1. `daq-0.5.tar.gz`
2. `libdnet-1.11.tar.gz`
3. `libnetfilter_queue-1.0.0.tar`
4. `libnfnetlink-1.0.0.tar`
5. `libpcap-1.1.1.tar.gz`
6. `pcre-8.12.zip`
7. `snort-2.9.0.5.tar.gz`
8. `snortrules-snapshot-2903.tar.gz`

The primary concern of this kind of presented centralized systems is [60]:

- The intrusion detection and reaction real-time is not good.
- Single host duplicate with the composed facts, hence the supervised network is restricted. The network can get overloaded due to lots of data collection.
- Adding new hosts can overload the centralized controller and can significantly increase, hence IDS is non-scalable

An open source IDPS to prevent system from attacks is SNORT. SNORT utilizes rule-based language, benefits of signature combined, with rules and irregularity based examination methods. One of the most widely organized Intrusion Detection and Prevention methodology is SNORT. Mainly for examining network traffic SNORT is used. SNORT generates alerts when threats are detected. In our dissertation work we are using SNORT inline mode. The main quality of inline mode is forwarding and routing of network traffic is configured and deployed on a server by SNORT.

The rules of SNORT for only “alert generation” are changed into “dropping rules”. Here we are also including iptables firewall application along with SNORT inline mode that interacts with iptables to obtain packets and route traffic of network.

SNORT is logically divided into multiple components [60]. These components work together to detect particular attacks and to generate output in a required format from the detection system. A SNORT based IDS consists of the following major components:

• Packet Decoder

• Preprocessors

• Detection Engine

• Logging and Alerting System

• Output Modules

Figure 3.3 shows how these components are arranged. Any data packet coming from the Internet enters the packet decoder. On its way towards the output modules, it is either dropped, logged or an alert is generated [60].

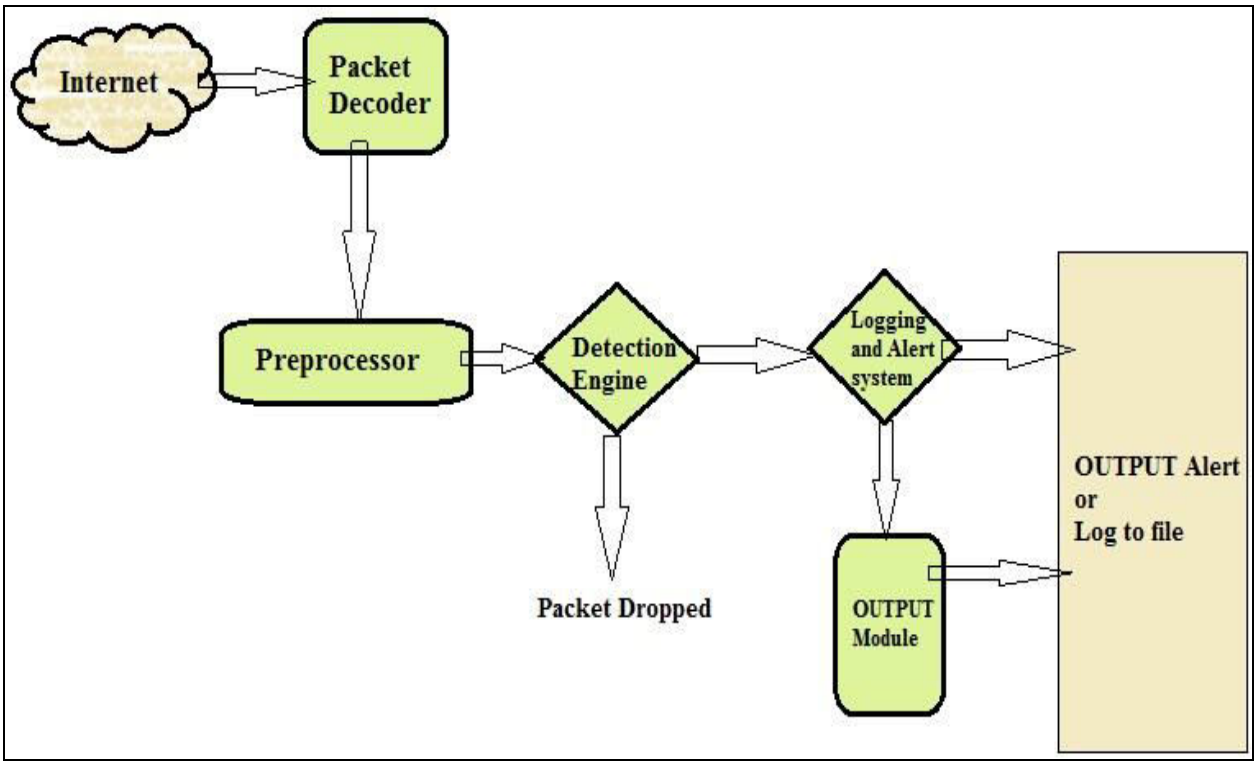


Figure 3.3: Components of SNORT [60].

1. **Packet Decoder:** The packet decoder takes packets from different types of network interfaces and prepares the packets to be preprocessed or to be sent to the detection engine. The interfaces may be Ethernet, SLIP, PPP and so on.

2. **Preprocessors:** Preprocessors are components or plug-ins that can be used with SNORT to arrange or modify data packets before the detection engine does some operation to find out if the packet is being used by an intruder. Some preprocessors also perform detection by finding anomalies in packet headers and generating alerts.

3. **Detection Engine:** The detection engine is the most important part of SNORT. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs SNORT rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets. If a packet matches any rule, appropriate action is taken; otherwise the packet is dropped. Appropriate actions may be logging the packet or generating alerts.

4. **Logging and Alerting System:** Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files, tcpdump-style files or some other form. All of the log files are stored under /var/log/snort folder by default. You can use –l command line options to modify the location of generating logs and alerts. Many command line options discussed in the next chapter can modify the type and detail of information that is logged by the logging and alerting system.

5. **Output Modules:** Output modules or plug-ins can do different operations depending on how you want to save output generated by the logging and alerting system of SNORT. Basically these modules control the type of output generated by the logging and alerting system. Depending on the configuration, output modules do things in following manner:

- Simply logging to /var/log/snort/alerts file or some other file
- Sending SNMP traps
- Sending messages to syslog facility

- Logging to a database like MySQL or Oracle. You will learn more about using MySQL later in this book
- Generating Extensible Markup Language (XML) output
- Modifying configuration on routers and firewalls.
- Sending Server Message Block (SMB) messages to Microsoft Windows-based machines.

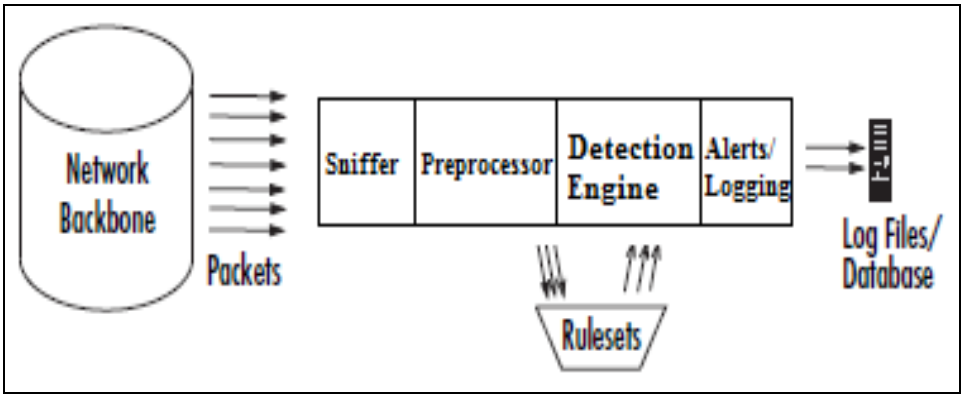


Figure 3.4: SNORT Architecture

Figure 3.4 shows detailed SNORT Architecture. This shows that the packets coming from network are preprocessed with the help of rule set written in database and if any signature matches from database it send an alert to Detection engine after which alerts are generated. In our work we are using SNORT 2.9.0 which includes Data Acquisition API (daq-0.5), for packet I/O. The DAQ replaces direct calls to libpcap functions with an abstraction layer that facilitates operation on a variety of hardware and software interfaces without requiring changes to SNORT. It is possible to select the DAQ mode and its' type while invoking SNORT to perform inline operation. After DAQ installation libdnet is installed, for installation of DAQ, lidnet, SNORT see Appendix B. SNORT-2.9.0.5 installation and working on it is done in Ubuntu 12.0.4.

SNORT have some ruleset that can be downloaded for IDS/IPS named as snortrules-snapshot-2903.tar.gz. The downloaded ruleset can be installed and configured for IDS/IPS. These rules consist of some white list and black list rules that can also be defined and configured. Various signatures like local rules can be added to snort ruleset for IDS and IPS.

3.4 SECURITY PROTOCOL IMPLEMENTATION

As we discussed in Chapter 2 many researchers have provided different solutions for protecting agent's information but its still an area of concern for researchers. In our dissertation work we have introduced security protocol for providing protection to mobile agent against different threats/attacks. Migration property of mobile agent that helps agent to move to different remote location makes this communication architecture superior than others. Our research work is providing mobile agent security during communication or execution thus targeting confidentiality. We have first provided transport layer protection using cryptographic algorithm.

3.4.1 AGENT CREATION AND COMMUNICATION

Firstly agent's are created inside container for our work we are taking an example of book trading. In which the server agent is buyer agent named "buyr". It is created inside main container which is the default container for agent creation we are using JADE platform. As soon as agent is created it sends a request. Here request sent consist of the name of the book the buyr is looking for named as "Programming in C". In Figure 3.5 screen shot shows buyr agent creation.

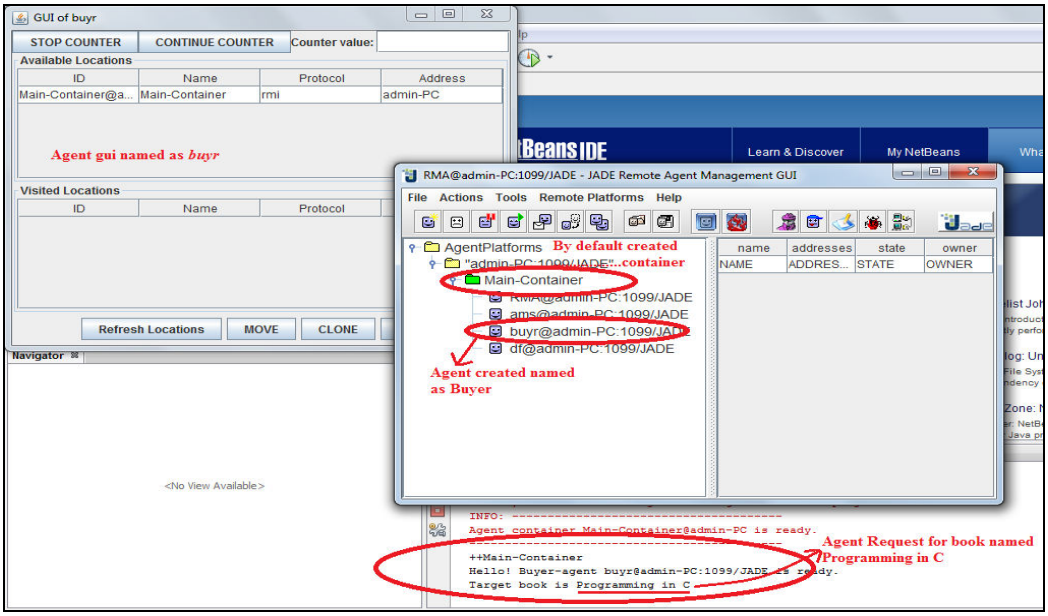


Figure 3.5: Creation of Container and Agent.

Now different client agent's can coat price according to themselves. In different containers different agent's are created these agents are sellers and to distinguish them we have given them unique names. Each container consists of one seller agent. The seller enters the price as soon as agent is created. Following Figure 3.6 and Figure 3.7 shows Seller3 agent creation that entered price as "130" for book named "Programming in C". These values get stored in buffer.

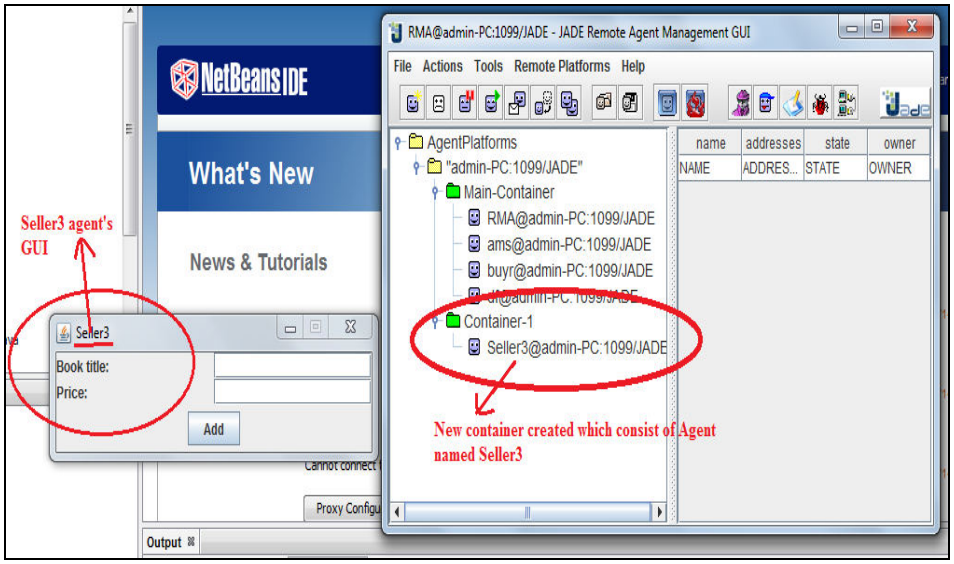


Figure 3.6: Creation Of Seller Agent And Container.

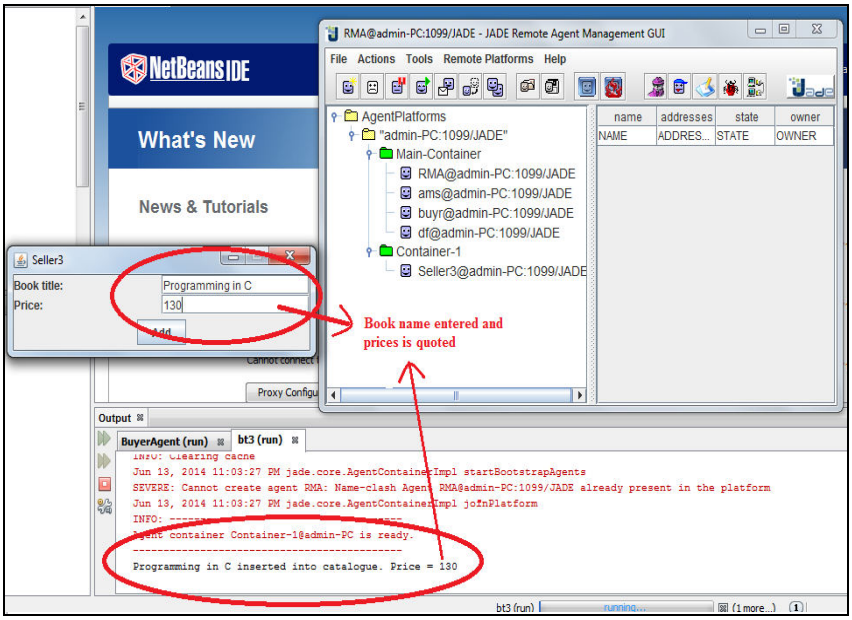


Figure 3.7: Seller3 Entered Price Stored in Buffer

Now one more agent is created named Seller2 in Figure 3.8. Seller2 has entered price = 200 for Book “Programming in C”.

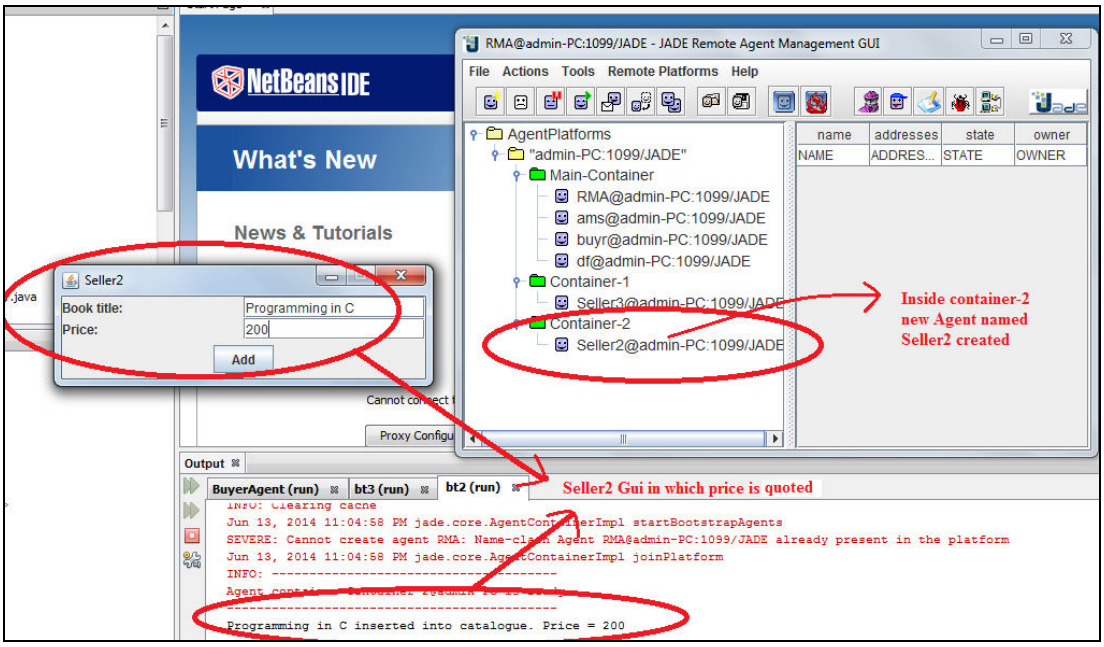


Figure 3.8: Seller Agent Creation and Quotation Stored in Buffer.

Finally on more seller agent is created inside container-3, named seller. The price entered for “Programming in C” is 120 in Figure 3.9.

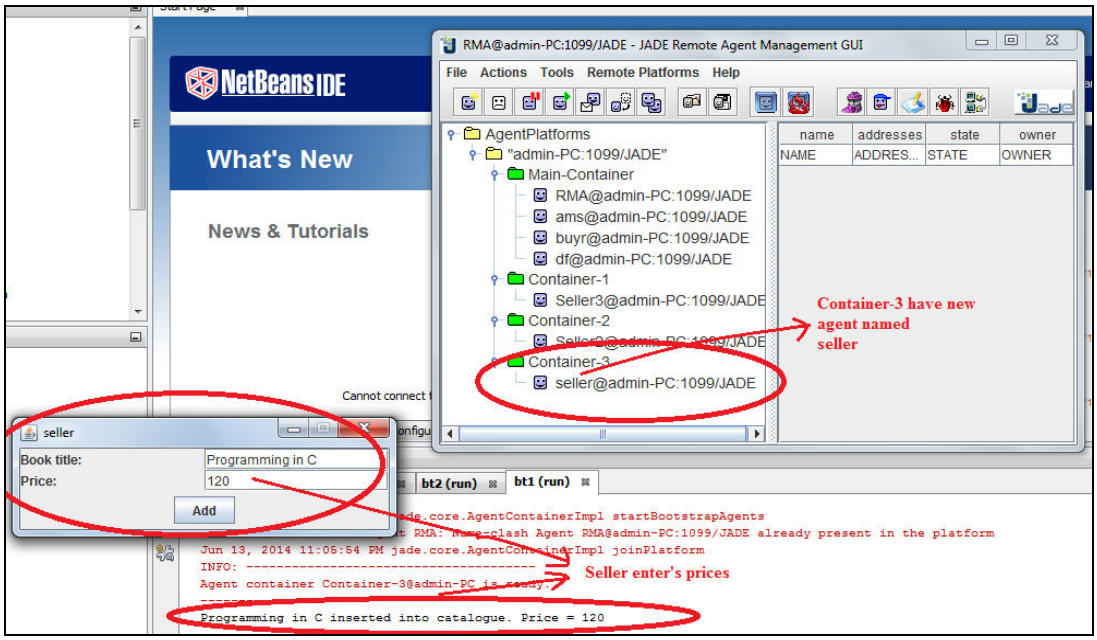


Figure 3.9: Seller Agent Creation and Price of Book Quoted.

When all sellers have entered their price then calculation is done and the least quoted price book is bought by buyer. In Figure 3.10 calculations is under process.

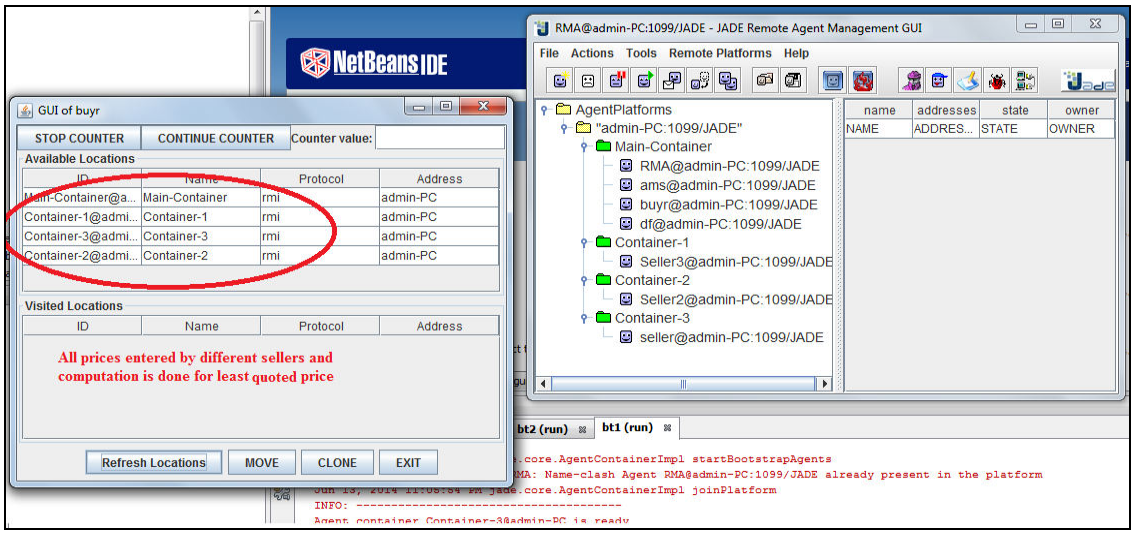


Figure 3.10: Calculation for Least Price.

In Figure 3.11 the least quoted price book is bought.

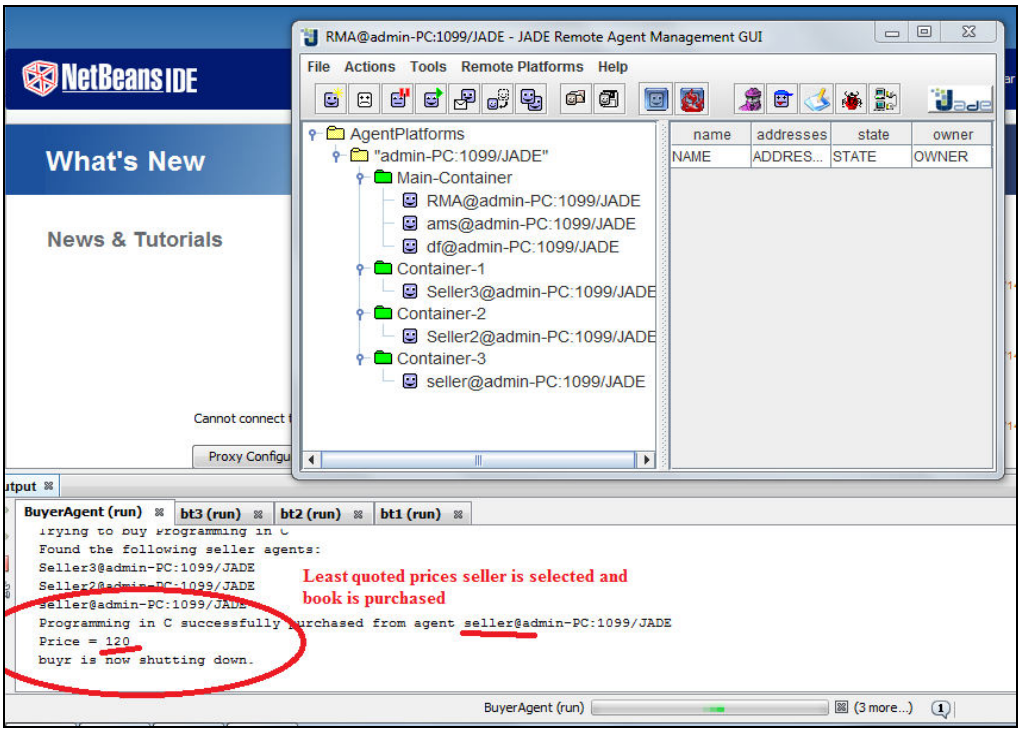


Figure 3.11: Buyer Bought Book from the Seller with Least Quoted Price.

Here the data is passed in an encrypted manner hence, by securing transport layer by using cryptographic algorithm.

3.4.2 NETWORK PROTECTION USING INTRUSION DETECTION AND PREVENTION

Now for securing application layer Intrusion Detection and Prevention System (IDPS) is used we have achieved this by using SNORT, and including signatures that help us in detecting and preventing our system from malicious agents' attacks. IDPS system keeps track of the incoming and outgoing data if any packets signature matches with the signature present in the list that packet is blocked or dropped as per command triggered. Thus it first detects and then prevent by blocking it or dropping it.

IDS are classified as either signature-based or anomaly based. Signature-based IDS aims to distinguish events that violate system network policy. Signature-based schemes seek defined patterns, or signatures, within the analyzed data. For this purpose, a signature database with correspondence to known attacks is stated as priority. Signature-based IDS schemes provide worthy detection results for specified, well-known attacks. There are different definitions of attack signatures. In our dissertation work, the main discussion focuses on content signatures, which represent a string of characters that appear in the payload of attack packets. No knowledge of normal traffic is required, but a signature database is needed for this category of detection systems. One of the main challenges faced by signature-based IDS is that every signature requires an entry in the database, and so a complete database might contain hundreds or even thousands of entries. Each packet is to be compared with all the entries in the database.

Anomaly-based IDSs attempt to analyze abnormal activities and flag these activities as attacks. Anomaly detectors detect behaviors on a computer or computer network that are not normal. Anomaly-based IDSs detect abnormal behaviors and generate alarms based on the abnormal patterns in network traffic or application behaviors. Typical anomalous behaviors captured include [32]:

- Misuse of network protocols such as overlapped IP fragments and running a standard protocol on a stealthy port.

- Uncharacteristic traffic patterns, such as more User Datagram Protocol (UDP) packets compared to Transmission Control Protocol (TCP).
- Suspicious patterns in application payload.

For prototype implementation, we choose SNORT as the signature-based IDS platform. The SNORT is installed and then during training period the system gathers information about existing threats to the network. SNORT stores its signatures in rule files referenced in the SNORT's configuration file. Once training is over, we look at the alert logs in database. Depending on the criteria and parameters that can be defined by the user, we identify the most frequent attacks based on the 1) minimum number of occurrences of a signature, 2) the age of the alert in the database, 3) and the maximum number of signatures that we would like to keep in the all the IDS.

The experiments were performed choosing two different hardware platforms to simulate attacks and run IDS, one more powerful than the other. The objective was to weaken and stop DoS attacks on agent.

For intrusion prevention IPS is used which have ips.conf file that drops the packet that have the same signature as of defined in SNORT rules. With IPS enabled SNORT we will be able to drop the incoming packet.

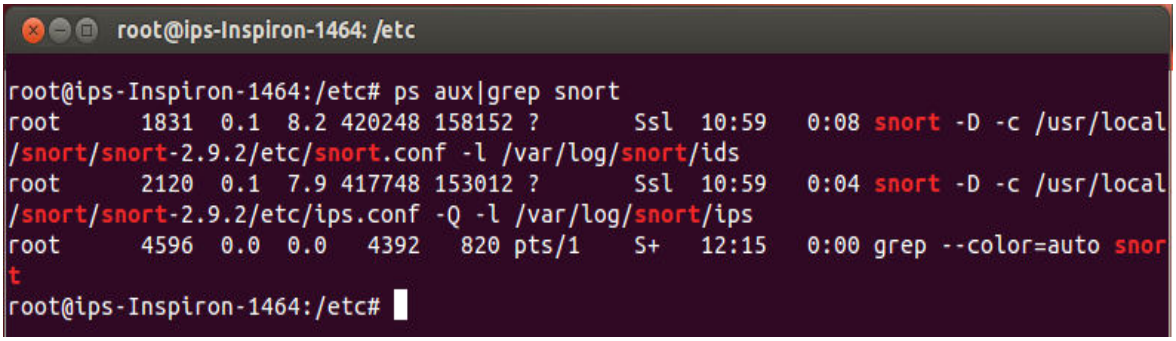


Figure 3.12: Calling ips with SNORT.

Figure 3.12 tells how snort initiates ips and execution of ips can set as default using above given command.

There is a snort_full file which consists of alerts of packet dropped and logged in case of IPS whereas in case of IDS malicious packets are detected and logged, not dropped.

The firewall used for IDS/IPS is snortalog, which is a log analyzer that uses open source libraries that categorize log files in term of distributions of attacks such port-wise distributions, top attacks statistics, pie-charts and bar charts. It is a useful tool to generate the reports of attacks by processing the IDS/IPS, it is freely available and open source software.

SNORT introduces DAQ or Data Acquisition library, for packet I/O. Figure 3.13 shows the configuration details of daq. The DAQ replaces direct calls to libpcap functions with an abstraction layer that facilitates operation on a variety of hardware and software interfaces without requiring changes to SNORT.

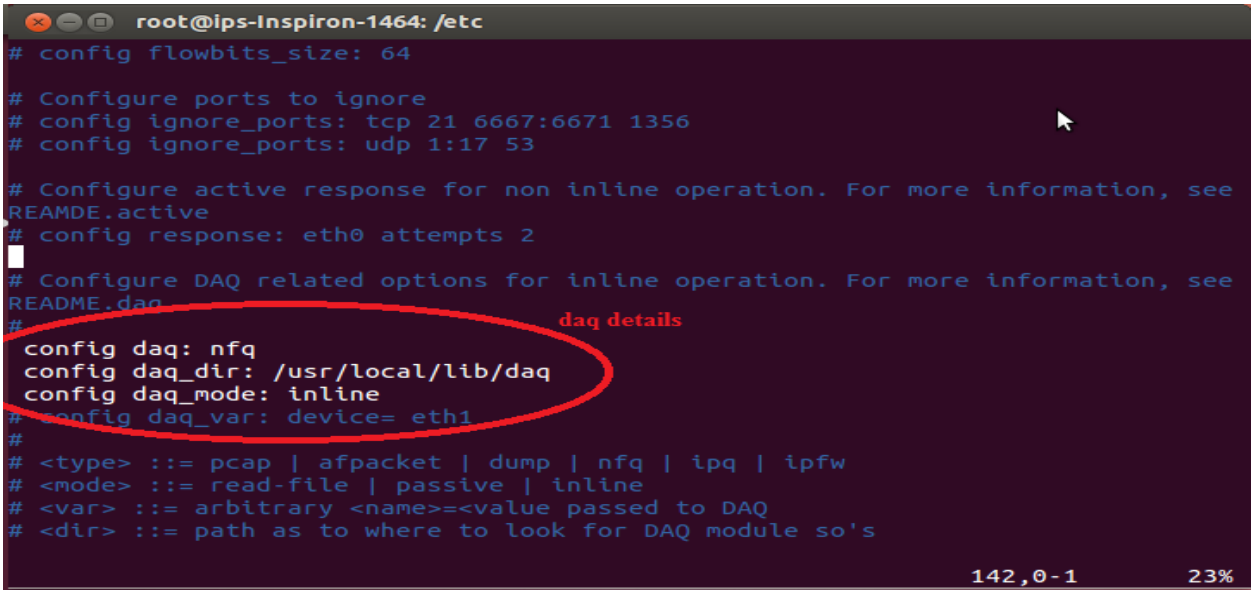


Figure 3.13: daq Configuration Details.

It is possible to select the DAQ type and mode when invoking Snort to perform pcap readback or inline operation, etc.

```
config daq: <type>
config daq_dir: <dir>
config daq_var: <var>
config daq_mode: <mode>

<type> ::= pcap | afpacket | dump | nfq | ipq | ipfw
<mode> ::= read-file | passive | inline
```

```
<var> ::= arbitrary <name>=<value> passed to DAQ
<dir> ::= path where to look for DAQ module
```

The DAQ type, mode, variable, and directory may be specified either via the command line or in the conf file. We can include as many variables and directories as needed by repeating the arg.config. DAQ type may be specified at most once in the conf and once on the command line; if configured in both places, the command line overrides the conf. NFQ is new and improved way to process iptables packets.

When Snort is in Inline mode, it acts as an IPS allowing drop rules to trigger. Snort can be configured to run in inline mode using the command line argument -Q and snort config optionpolicy_mode as follows:

```
snort -Q
config policy_mode:inline
```

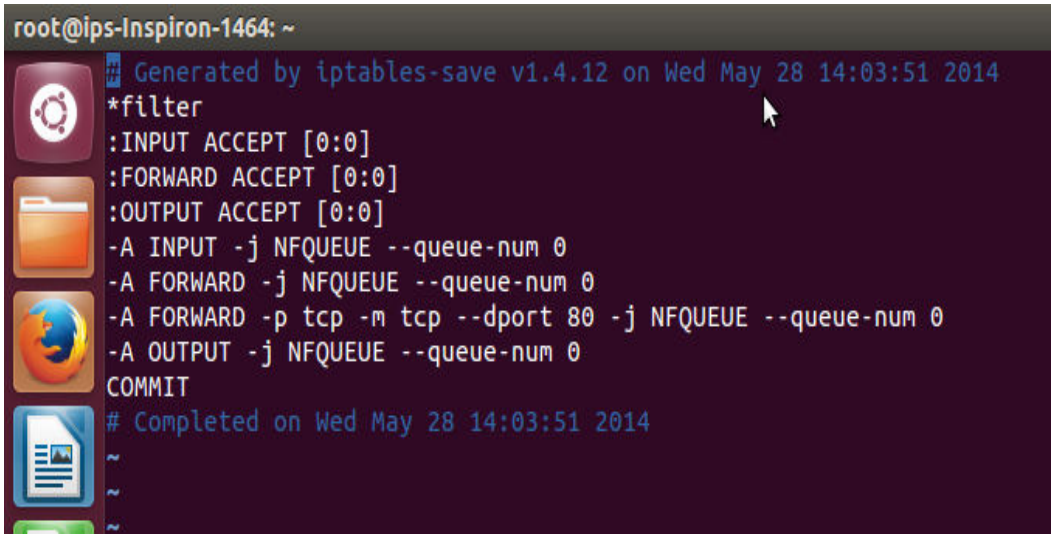


Figure 3.14: iptables Configuration.

Figure 3.14 shows the configuration of iptables, which is allowing all input and output to come with full permission. NFQUEUE is NetFilter Queue module is needed to send traffic to snort inline. Iptables sends traffic to snort inline using NFQUEUE target.

Command : **`iptables -A FORWARD -j NFQUEUE`**

The above command is used to send all traffic to NFQUEUE target by sending packets from kernel space to user space, *i.e.,* snort. Further snort decides whether to drop the packet or not. All the packets get blocked if snort is not running.

3.5 CONCLUSION

In this chapter we came across with various platform out of which JADE is one of the latest technologies. We are using JADE technology for mobile agent communication on windows platform, and for providing protection to agent against intrusion detection we studied SNORT technique.

Here we discussed the creation of agent and how they communicate and transfer data with each other. It also explains installation of snort and its libraries needed for intrusion detection and prevention system and gives a clear idea of its working.

CHAPTER 4

RESULT AND DISCUSSION

Mobile agent migrates from one place to another in ad-hoc environment, which is prone to various threats. Thus, information carried by agent is not secure. To provide security to information we are using encryption technique.

We have taken a case for information transfer which is an example of book-buying which includes two entities book_name and book_price. An agent named buyer (data initiator) is created in the main container and other agents named seller, seller 2, seller 3 who respond to the query fired by buyer.

The agent lies in ad-hoc network hence, the intruder can put all the efforts to get the information from the agent. But if client itself will encrypt the data by using some encryption method and then forward the information to agent, cracking key will be a difficult task.

The private key will be with the buyer/server after getting the information by using private key he would be able to decode it. And by this no other client would be able to other agents' information. The data obtained by seller is stored in buffer and RSA cryptographic technique is used to encrypt it [61].

4.1 TEXT CIPHERING

The idea is making agent's information safe and secure. The encryption scheme have mainly three categories :

- **Setup:** Which generates public and private key pair.
- **Encrypt:** Public key is applied to any arbitary string so that it would not be human redable.
- **Decrypt:** Decrypts message/string using corresponding private master key.

A brief RSA Algorithm is depicted in Algorithm 1 below:

Algorithm 4.1: RSA Cryptography [61]

1. Choose two large primes **a** and **b** such that **a** ≠ **b.**
2. **m← a x b**
3. **¢(m)← (a-1) x (b-1)**
4. Choose **e** such that **1 < cp < ¢(m)** and **e** is co-prime to **¢(m)**
5. **k←cp^{-1}** mod **¢(m) k** is inverse of **cp modulo ¢(m)**
6. Public_key **← (cp,m) //**To be announced publicly
7. Private_key**← k** **//**Kept Private
8. Return Private_key and Public_key

For encryption we have taken RSA parameters as: a = 3, b = 11, m = a * b = 3 * 11 = 33, now ¢ (m) = (a - 1) * (b - 1) = 2 * 10 = 20 and value of cp is taken between 1 and ¢ (m), *i.e.,* cp=7. The alphabets' numerical value will be a=0,b=1,c=2… y=24,z=25. Further steps of computation are:-

- Calculate value for k such that (k * cp) % ¢ (m) = 1. One of the answers to this statement is : K → 3 [(3 * 7) % 20 = 1]; Now
- Public key is (cp, m) → (7, 33)
- Private key is (k, m) → (3, 33)
- Message given is 'Program in C and 170'
- The encryption of character 'p' in message is p = 15, c = 15^7 % 33 = 27; for character 'r' is 17, c = 17^7%33 =8; for character 'o' = 14, c= 14^7%33= 20… Similarly all the rest characters will be encrypted using public key e.

- The decryption of message is for c = 27, m = 27^3 % 33 = 15 (15='p'); for c =8, m= 8^3%33=17 (17='r'); for c=20, m= 20^3%33=14 (14 ='o')… Similarly rest characters will be decrypted using private key d.

Any seller who wants to tell his price will encrypt his data using RSA public key. By this no other seller would be able to know what price the other one has mentioned. Hence no unauthenticated person would be able to manipulate, delete or read that data. When the buyer will get the price list of all the sellers he will encrypt the information using private key. Below in Figure 4.1 we had taken an example having book name "program in c" and whose price is "170". This information is stored in buffer in encrypted form only the buyer can decrypt it by using its private master key.

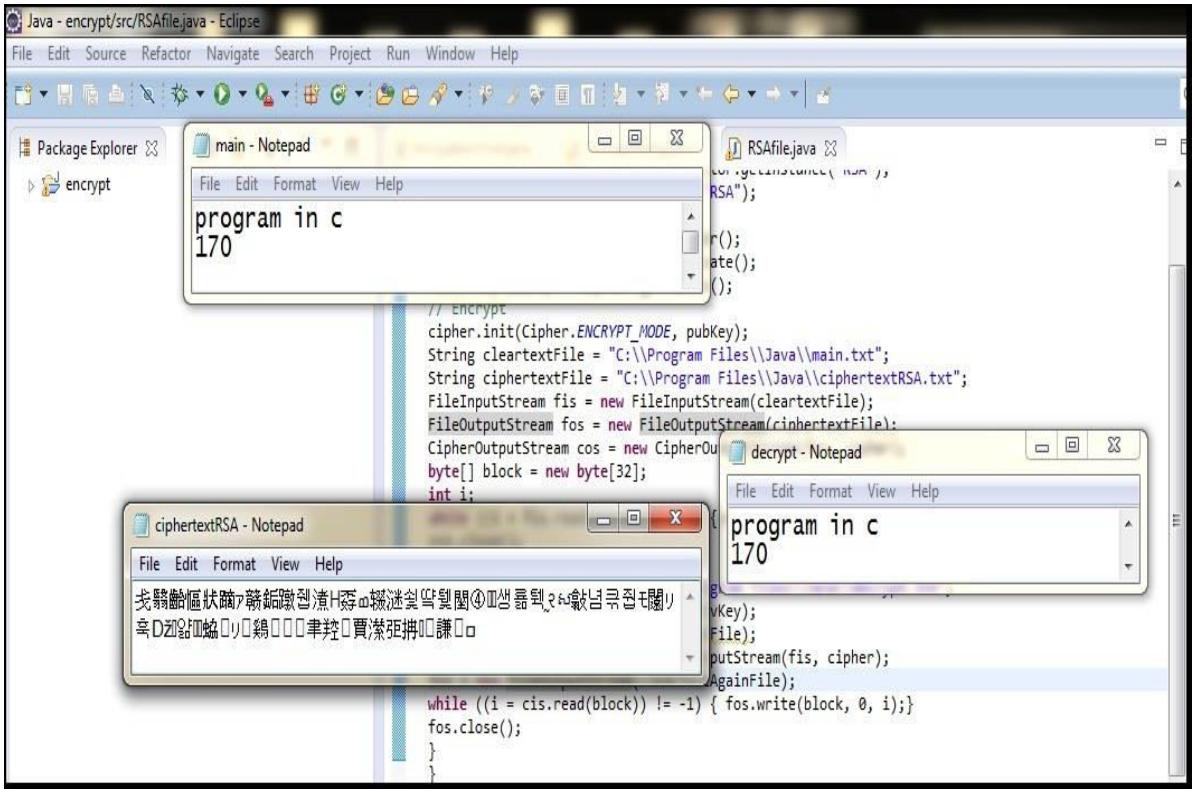


Figure 4.1: Source Data (at Source), Encrypted Data (at Internet) and Decrypted Data (at Destination)

Before migrating from one place to another the data stored by agent is encrypted so that no other agent would be able to know what information agent is carrying.

4.2 INTRUSION DETECTION AND PREVENTION

For making it more secure we also need to have application layer security, which will be possible with the help of Intrusion detection and Prevention system. In our work we are using snort and iptables for IDPS. Snort has some local rules in its database and according to which for each incoming and outgoing packets are first processed and checked. Figure 4.2 shows Local rules written. If signature is found in database then the packet is dropped. Finding or detecting packet is Intrusion Detection part and dropping or blocking that packet is Prevention part.

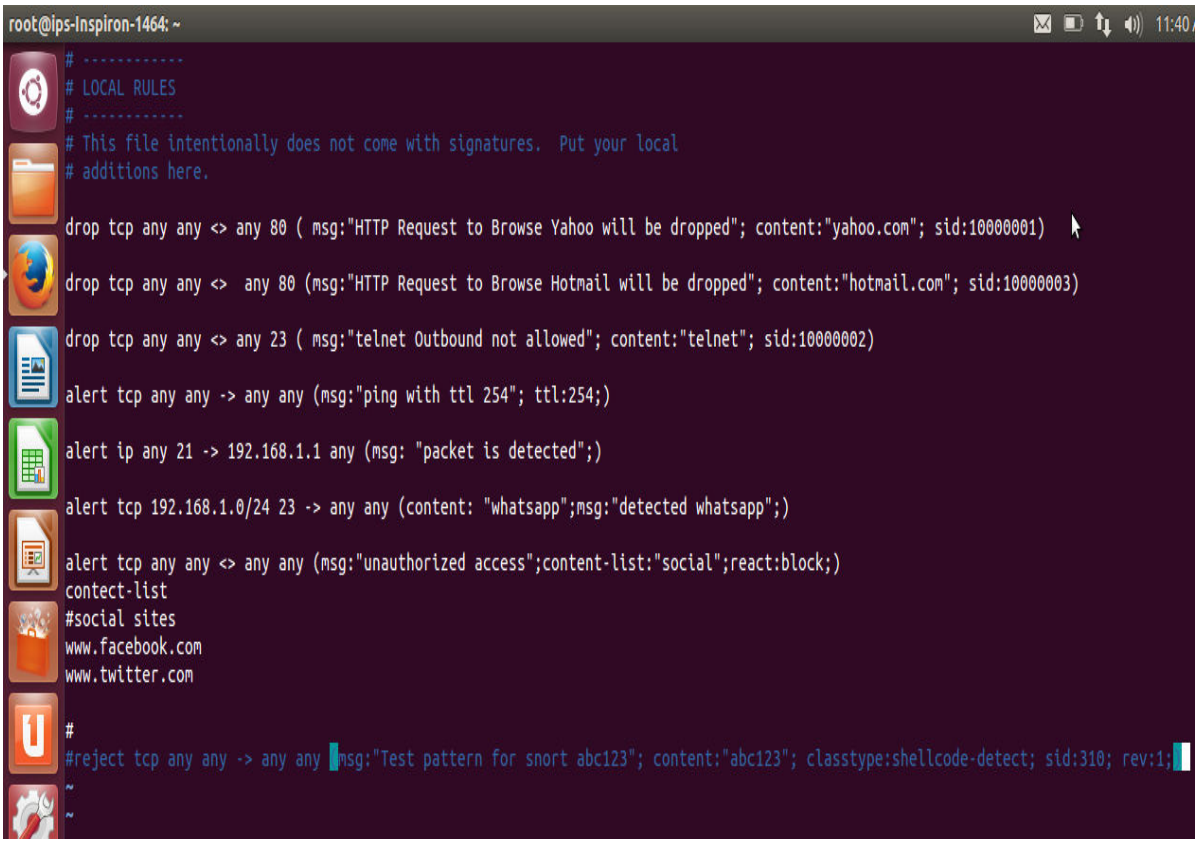


Figure 4.2: Local Rules for SNORT

Local rules written and signatures added in snort database. Below Figure 4.3 shows that if google.com is called the connection is established very easily, as there is signature found in database similar to the rules written.

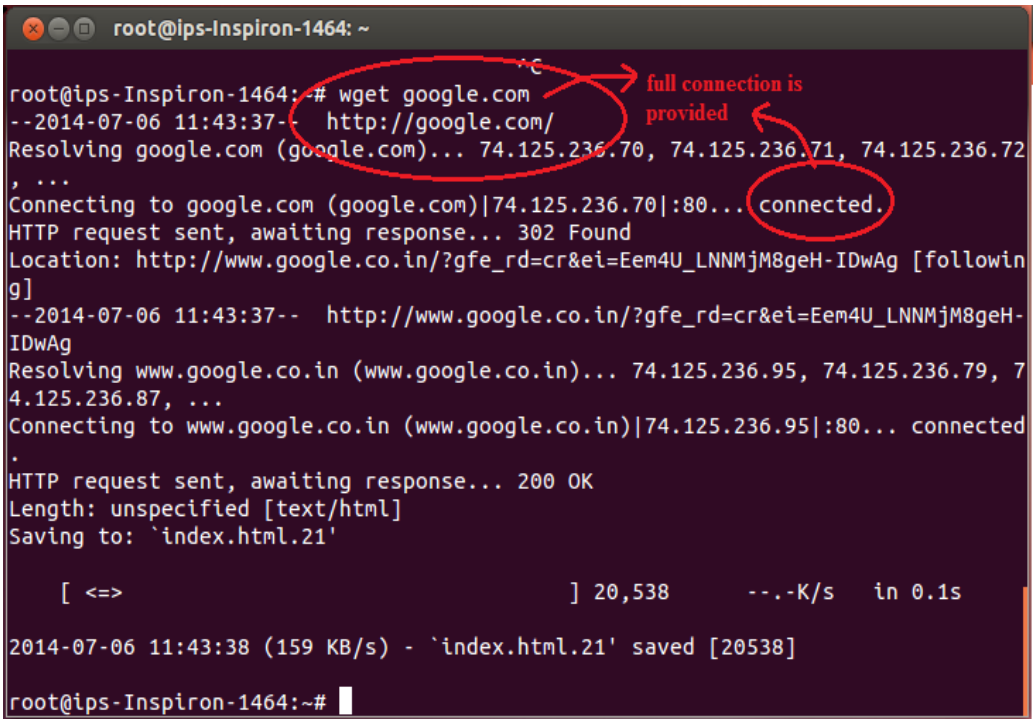


Figure 4.3: SNORT Action after Calling Google.com.

In below Figure 4.4 when yahoo.com was called snort detected the signature and ips blocked it. Hence, preventing this packet to enter and blocking it.

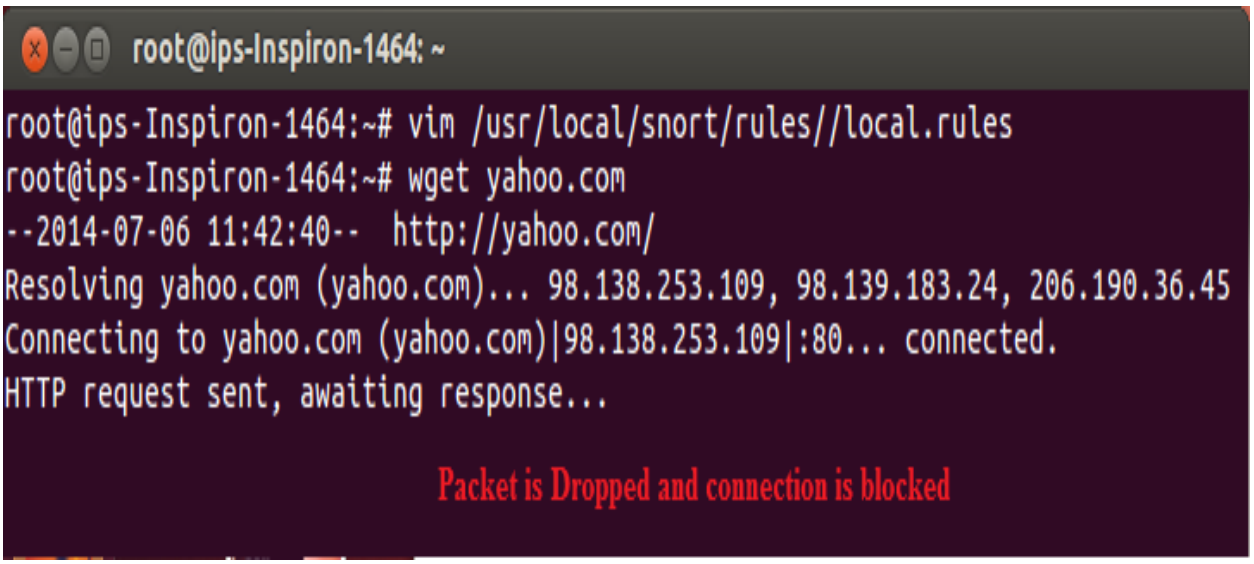


Figure 4.4: Output when yahoo.com is Called

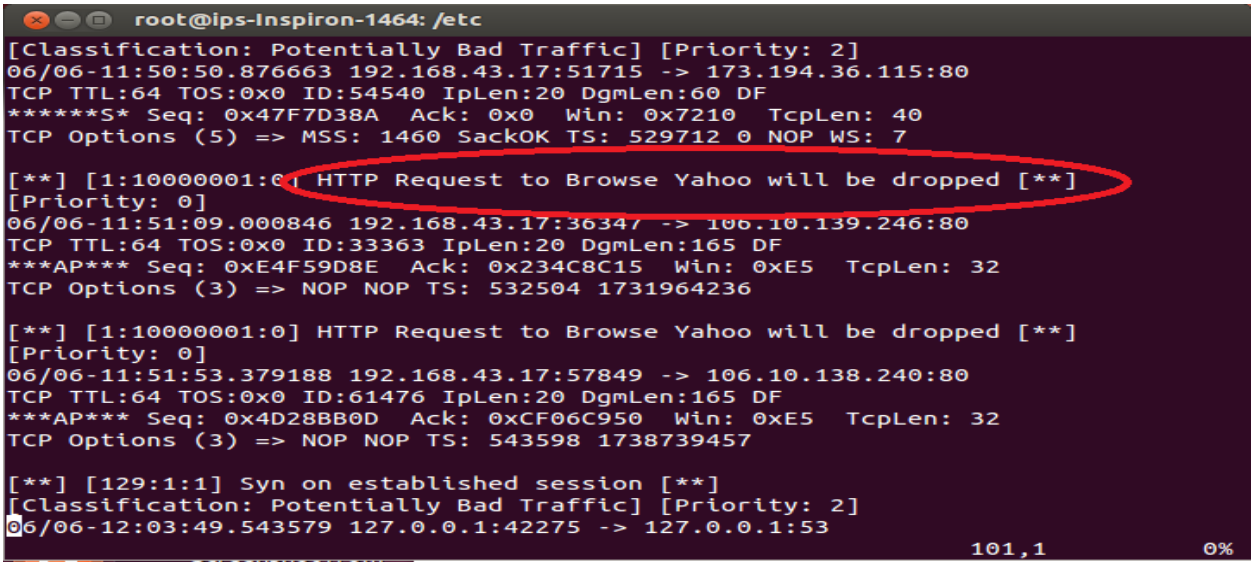


Figure 4.5: Snort_full Result

Figure 4.5 shows output of snort_full. Snort_full contains the alerts of packets dropped and logged in case of IPS whereas in case of IDS malicious packets are detected and logged, not dropped.

Below figure shows that when a HTTP request was sent for browser named yahoo.com snort blocked the packet as it got matched with the local rules specified. Hence, request for browser is dropped.After intrusion detection for approx. 2 days report is generated. Generated report is as follows:

The log begins at :	Jun 06 11:50:05	**Domains File :**	conf/domains
The log ends at :	Jun 08 13:25:21	**Number of domains :**	267
Total of Lines in log file :	4168	**Rules File :**	conf/rules
Total of Logs Dropped :	4 (0.10%)	**Number of referenced rules :**	2067
Total events in table :	711		
Source IP recorded :	17		
Destination IP recorded :	68		
Host logger recorded :	1 with 1 interface(s)		
Signatures recorded :	15		
Classification recorded :	7		
Severity recorded :	2		
Portscan detected :	0		
Host logger recorded :	1 with 1 interface(s)		

Figure 4.6: Packet Tracing Recorded Statistics for 3 Days

Figure 4.6 shows packet tracing that begins from 6th June and ends at 8th June, snort first detects total no. of log files, records source and destination IP address. All the signatures are identified and recorded.

Figure 4.7 shows that in every graph RED color show dangerous connection, ORANGE color show warning for less dangerous connection that needs investigation and BLACK color tells about the alerts that are not that dangerous. Around 83% warning connection are found that need investigation.

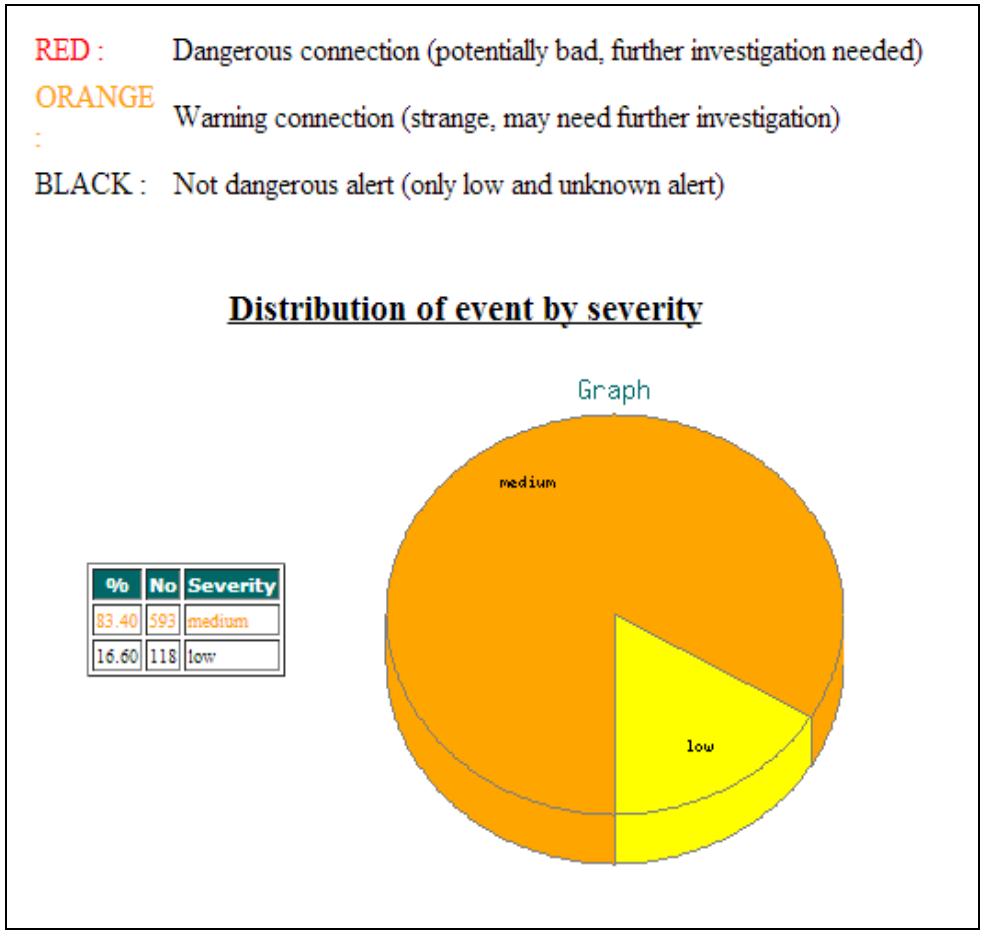


Figure 4.7: Distribution of Event by Severity

Figure 4.8 tells distribution of attack by hour, the maximum attack was at 13hour and minimum was at 11hours. The graph also tells the no. of events that took place by hours.

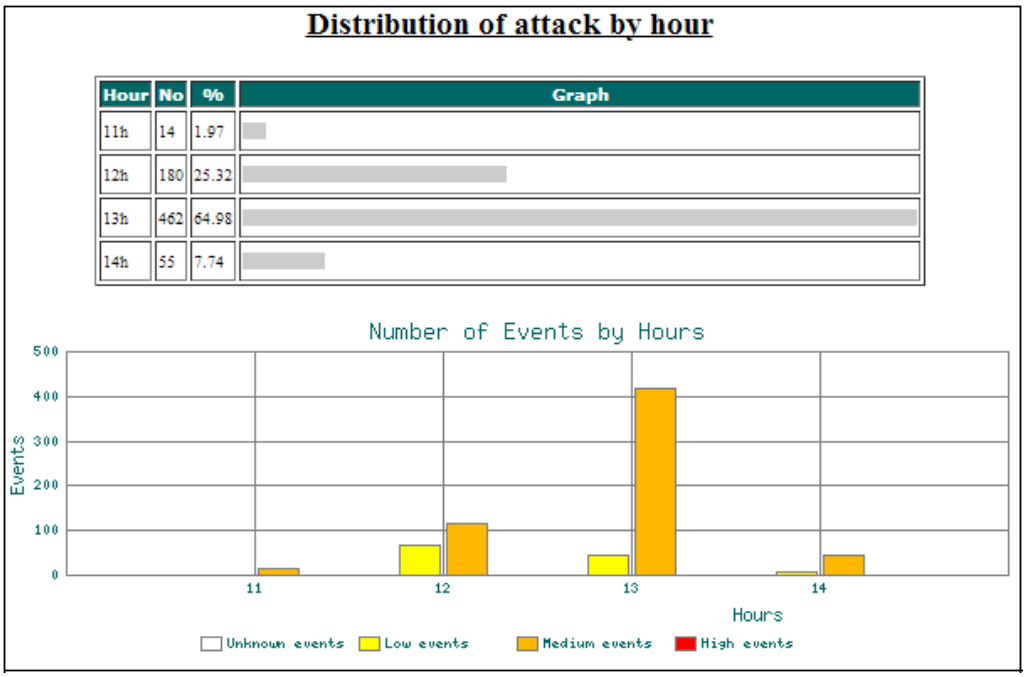


Figure 4.8: Distribution of Attack by Hour

Figure 4.9 shows distribution of events by different destination port. The most used port by event is port 80 and is approx. 82% and rest are less than 10%.

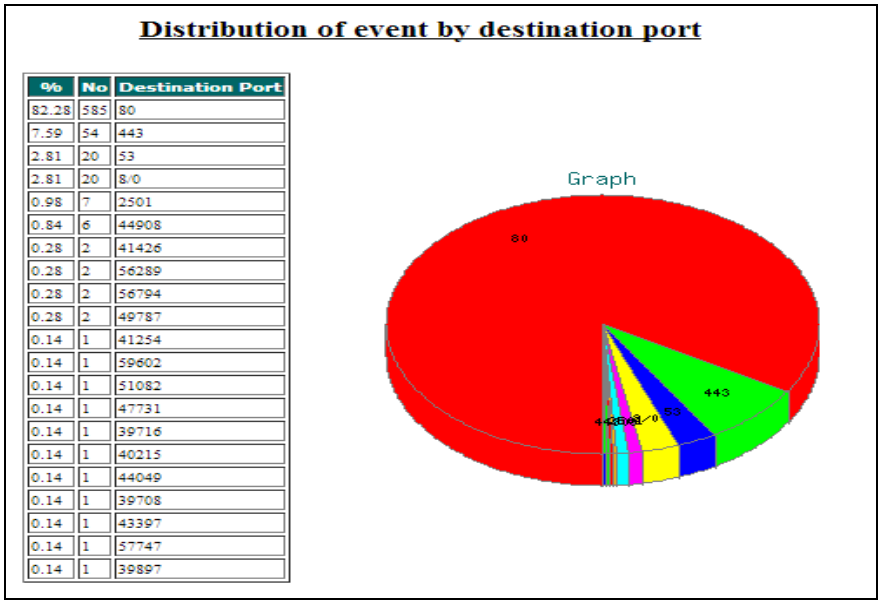
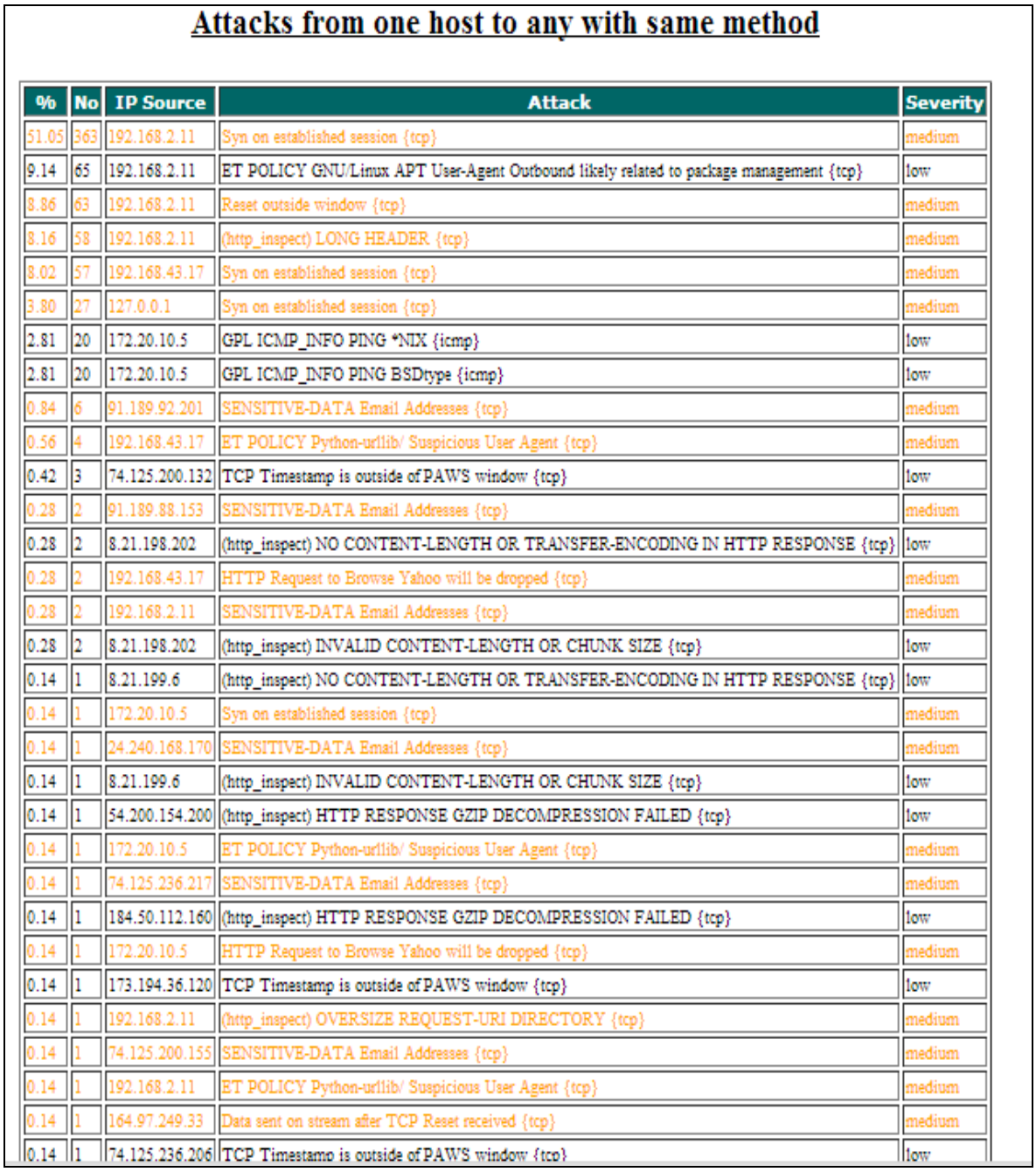


Figure 4.9: Distribution of Event by Destination Port

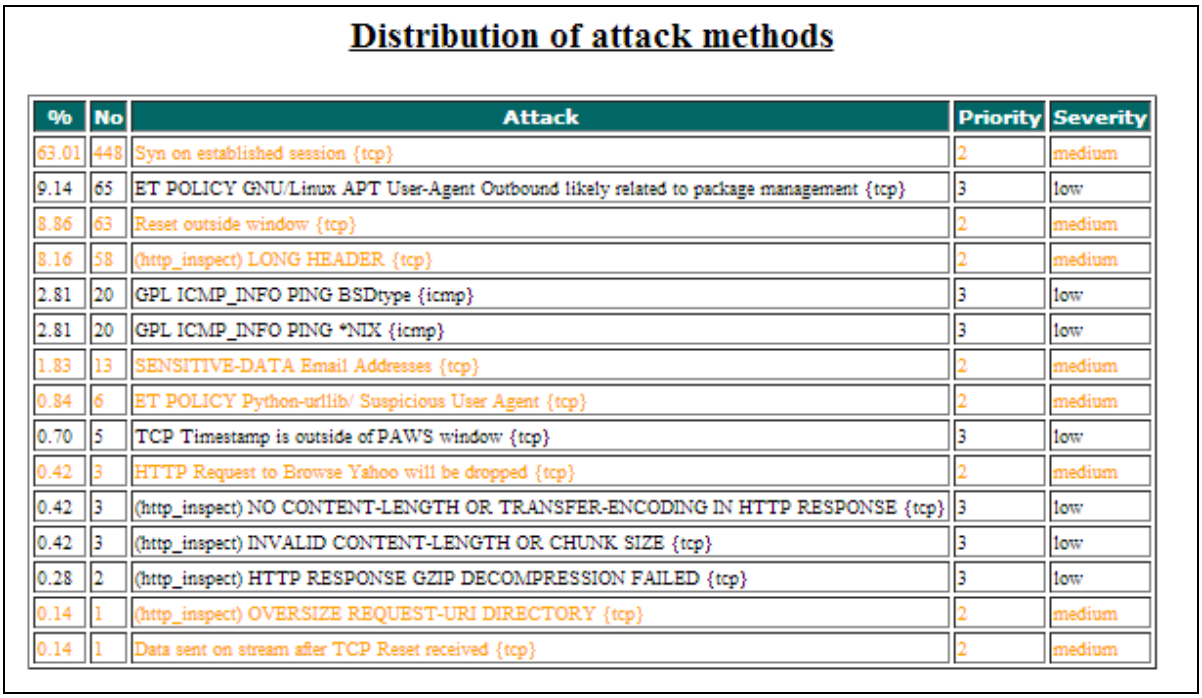


Attacks from one host to any with same method

%	No	IP Source	Attack	Severity
51.05	363	192.168.2.11	Syn on established session {tcp}	medium
9.14	65	192.168.2.11	ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management {tcp}	low
8.86	63	192.168.2.11	Reset outside window {tcp}	medium
8.16	58	192.168.2.11	(http_inspect) LONG HEADER {tcp}	medium
8.02	57	192.168.43.17	Syn on established session {tcp}	medium
3.80	27	127.0.0.1	Syn on established session {tcp}	medium
2.81	20	172.20.10.5	GPL ICMP_INFO PING *NIX {icmp}	low
2.81	20	172.20.10.5	GPL ICMP_INFO PING BSDtype {icmp}	low
0.84	6	91.189.92.201	SENSITIVE-DATA Email Addresses {tcp}	medium
0.56	4	192.168.43.17	ET POLICY Python-urllib/ Suspicious User Agent {tcp}	medium
0.42	3	74.125.200.132	TCP Timestamp is outside of PAWS window {tcp}	low
0.28	2	91.189.88.153	SENSITIVE-DATA Email Addresses {tcp}	medium
0.28	2	8.21.198.202	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE {tcp}	low
0.28	2	192.168.43.17	HTTP Request to Browse Yahoo will be dropped {tcp}	medium
0.28	2	192.168.2.11	SENSITIVE-DATA Email Addresses {tcp}	medium
0.28	2	8.21.198.202	(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE {tcp}	low
0.14	1	8.21.199.6	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE {tcp}	low
0.14	1	172.20.10.5	Syn on established session {tcp}	medium
0.14	1	24.240.168.170	SENSITIVE-DATA Email Addresses {tcp}	medium
0.14	1	8.21.199.6	(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE {tcp}	low
0.14	1	54.200.154.200	(http_inspect) HTTP RESPONSE GZIP DECOMPRESSION FAILED {tcp}	low
0.14	1	172.20.10.5	ET POLICY Python-urllib/ Suspicious User Agent {tcp}	medium
0.14	1	74.125.236.217	SENSITIVE-DATA Email Addresses {tcp}	medium
0.14	1	184.50.112.160	(http_inspect) HTTP RESPONSE GZIP DECOMPRESSION FAILED {tcp}	low
0.14	1	172.20.10.5	HTTP Request to Browse Yahoo will be dropped {tcp}	medium
0.14	1	173.194.36.120	TCP Timestamp is outside of PAWS window {tcp}	low
0.14	1	192.168.2.11	(http_inspect) OVERSIZE REQUEST-URI DIRECTORY {tcp}	medium
0.14	1	74.125.200.155	SENSITIVE-DATA Email Addresses {tcp}	medium
0.14	1	192.168.2.11	ET POLICY Python-urllib/ Suspicious User Agent {tcp}	medium
0.14	1	164.97.249.33	Data sent on stream after TCP Reset received {tcp}	medium
0.14	1	74.125.236.206	TCP Timestamp is outside of PAWS window {tcp}	low

Figure 4.10: Percentage of Attack from one Host with Same Method

Figure 4.10 gives the percentage of same method of attack from the same host. The orange colored attacks are severe attack and their severity is medium and black colored attacks have low severity.



Distribution of attack methods

%	No	Attack	Priority	Severity
63.01	448	Syn on established session {tcp}	2	medium
9.14	65	ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management {tcp}	3	low
8.86	63	Reset outside window {tcp}	2	medium
8.16	58	(http_inspect) LONG HEADER {tcp}	2	medium
2.81	20	GPL ICMP_INFO PING BSDtype {icmp}	3	low
2.81	20	GPL ICMP_INFO PING *NIX {icmp}	3	low
1.83	13	SENSITIVE-DATA Email Addresses {tcp}	2	medium
0.84	6	ET POLICY Python-urllib/ Suspicious User Agent {tcp}	2	medium
0.70	5	TCP Timestamp is outside of PAWS window {tcp}	3	low
0.42	3	HTTP Request to Browse Yahoo will be dropped {tcp}	2	medium
0.42	3	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE {tcp}	3	low
0.42	3	(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE {tcp}	3	low
0.28	2	(http_inspect) HTTP RESPONSE GZIP DECOMPRESSION FAILED {tcp}	3	low
0.14	1	(http_inspect) OVERSIZE REQUEST-URI DIRECTORY {tcp}	2	medium
0.14	1	Data sent on stream after TCP Reset received {tcp}	2	medium

Figure 4.11: Distribution of Attack Method

Figure 4.11 tells various distribution attack method, their severity and priority. Maximum attacks happened when synchronization established during session with tcp and is approx. 63%.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1CONCLUSION

In this dissertation we have discussed information communication between two nodes in ad-hoc way by using client-server and mobile agent based communication model. We have compared the two communication strategies and found that mobile agent based communication is more relevant and fruitful for ad-hoc communication. Since the communication channel is wireless and information is carried by Mobiles Agents form one station to another station, information is not secure. In other word information is highly prone to attacks and possible threats. Therefore, in this work we have proposed and discussed a method for secure mobile agent information communication between two nodes and provided safe secure communication using encryption and intrusion detection system.

For agent development and transformation we have used JADE development environment and for intrusion detection we have used SNORT technology. For the purpose of ciphering RSA algorithm has been used. As a result of dissertation we are able to design a mobile agent based secure communication model for Ad-Hoc networks, implemented RSA algorithm for encryption and finally used SNORT for packet tracing, dropping and for intrusion detection. The key research findings are as follows:

- Comparison of Client Server and Mobile Agent based communication models.
- Proposed a Mobile Agent based secure communication model
- Provide security to the information using RSA algorithm
- Intrusion detection, attacking a node
- Detection and dropping of malicious packet identified by SNORT method

5.2FUTURE SCOPE

Although the work is a start up of mobile agent based communication that we have furnished in this dissertation, we have planned to expand the work for multi agent based communication and for multiple platforms. Further, more intrusion, worms and Trojans detection mechanism need to be investigated. To provide more security to agents' data, a double encryption technique is under development.

REFERENCES

[1] R. Bindhu, "Mobile Agent Based Routing Protocol with Security for MANET", Intrenational Journal of Applied Engineering Research, Dindigul, vol 1, no l, pp. 92-101, 2010.

[2] H Yang, Haiyun Luo and L Zhang, "Security in Mobile Ad Hoc networks: Challenges and Solution", IEEE Wireless Communications,vol 11, no.1 pp. 38-47, Feburary 2004.

[3] Wayne Jansen and Tom Karygiannis, "Mobile Agent Security", NIST Special Publication 800-19, 1999.

[4] Bing Wu, Jianmin Chen and Jie Wu, "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks" in Wireless Mobile Network Security Signals and Communication tTechnology, Springer, pp. 103-135, 2007.

[5] Pattie Maes, "On Software Agents: Humanizing the Global Computer", IEEE Internet Computing, vol. 1, no. 4, pp. 10-19, August 1997.

[6] M. H. Coen, "SodaBot: A Software Agent Environment and Construction System" MIT AI Lab Technical Report 1493, 1994.

[7] Stan Franklin and Art Graesser, "Is It An Agent, or Just a Program?: A Taxonomy for Autonomous Agents", in Proc. Third International Workshop on Agent Theories, Architectures, And Languages, Springer-Verlag, pp.21-35, 1996.

[8] Tarig Mohamed Ahmed, "Increasing Mobile Agent Performance",Vol.6,No.4, June, 2007.

[9] FIPA Agent Management Support for Mobility Specification Available: http://www.fipa.org/specs/fipa00087/DC00087C.html, 2010.

[10] Berson Ales, "Client-Server architecture", McGraw-Hill, IEEE-802, 1992.

[11] Gurdeep Singh Hura, "Client-Server Computing Architecture: An efficient paradigm for project management", in proc. Engineering Management Conference IEEE, pp. 146-152, June 1995.

[12] Asha Nagesh, "Distributed Network Forensics using JADE Mobile agent Framework", Student Trade Show and Project Reports, Arizona State University, 2006.

[13] Parul Ahuja and Vivek Sharma, "A review on Mobile Agent Security", International Journal of Recent Technology and Engineering, vol. 1, no. 2, pp. 83-88, June 2012.

[14] Mo Chun Man and Victor K. Wei, "A Taxonomy for Attacks on Mobile Agent", EUROCON'2001, Trends in Communications, vol. 2, pp. 385-388, July 2001.

[15] Danny B.Lange and Mitsuru Oshima, "Programming and Developing Java Mobile Agents with Aglets", Addison-Wesley publication, 1998.

[16] Wu Bing, Jianmin Chen, Jie Wu and Mihaela Cardei. "A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks" in Wireless Network Security, Springer US, pp. 103-135, 2007.

[17] Ali Ghaffari, "Vulnerability and Security of Mobile Ad hoc Networks" in Proc. 6th WSEAS International Conference on Simulation, Modeling and Optimization, pp. 124-129, September, 2006.

[18] Xiaoxin Wu and Bharat Bhargava, "AO2P: Ad Hoc On-Demand Position-Based Private Routing Protocol",in Mobile Computing, vol. 4, no.4, pp. 335-348, August 2005.

[19] Iilyas Mahammad, "The Hand Books of Ad-Hoc Wireless Networks-II Series" CRC Press LLC, USA, 2003.

[20] Lidong Zhou and Zygmunt J. Haas, "Securing Ad Hoc Networks" Special Issue on Network Security, vol. 13, no. 6, pp. 24-30, December, 1999.

[21] C.Siva Ram Murthy and B.S.Manoj, “Adhoc Wireless Networks-II Edition”, Technology and Engineering, Pearson Education of India, May 2004.

[22] R. Arunkumar and A Annalakshmi, “A Recent Analysis of Intrusion Detection and Prevention System for Protectiong Range of Attack using Data Gathering Technique in MANET” , in International Journal of Computer Application, vol 85, no 8, pp. 9-15, January 2014.

[23] E. Gajendra and S. Vijayan , “A literature Survey on Security Challenges in Mobile Ad-hoc Networks”, International Journal of Computer Application, vol 84, no 1, pp. 1-5, December 2013.

[24] Jianxiao Liu and Lijuan Li, “A Distributed Intrusion Detection System Based on Agents”, Computational Intelligence and Industrial Application, Pacific-Asia Workshop, vol. 1, pp. 553-557, December 2008.

[25] David Mudzingwa and Rajeev Agrawal “A Study of Methodologies Used in Intrusion Detection and Prevention Systems (IDPS)” in Proc. Southeastcon IEEE, pp. 1-6, March 2012.

[26] Ditipriya Sinha and Rituparna Chaki “AESCRT: Agent enabled secure CRT based routing topology for MANET”, Applications and Innovations in Mobile Computing (AIMoC), IEEE, pp. 172-178, February 2014.

[27] Preeti Bhati, Rinki Chauhan and Ritu Khurana, “An Efficient Agent-Based AODV Routing Protocol in MANET” International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397 Vol. 3 No. 7 July 2011.

[28] Halim, Islam Tharwat A “Agent-Based Trusted On-Demand Routing Protocol for Mobile Ad Hoc Networks Network”, 4th International Conference onNetwork and System Security, pp. 255-262, September 2010.

[29] Lanjun Dang,Jie Xu,Hui Li,Nan Dang "DASR: Distributed Anonymous Secure Routing with GoodScalability for Mobile Ad Hoc Networks", Services Computing Conference (APSCC), pp. 454-461, December 2010.

[30] Narjes Bouchemal, Maamri Ramdane and Sahnoun Zaidi, "CAP: Clone Agent Protocol to Protect Mobile Agents" MASAUM Journal of Basic and Applied Sciences, 2009.

[31] Aikaterini Mitrokotsa, Rosa Mavropodi, Christos Douligeris, "Intrusion Detection of Packet Dropping Attacks in Mobile Ad Hoc Network" International Conference on Intelligent Systems and Computing, pp. 111-118, July 2006.

[32] Esparza Oscar, Marcel Femandez and Miguel Soriano, "Protecting Mobile Agents by Using Traceability Techniques", IEEE International Conference on Information Technology, pp.264-268, August 2003.

[33] Oleg Kachirski, Ratan Guha, "Effective Intrusion Detection Using Multiple Sensors in Wireless Ad-Hoc Networks"in Proc.of 36th Annual Hawaii International Conference on System Sciences (HICSS'03), pp. 1-8, January 2003.

[34] Patrick Albers, Olivier camp and Jean-Marc Percher , "Security in Ad-Hoc Networks: A General Intrusion Detection Architecture Enhancing Trust Based Approaches", in Wireless Information Systems, pp. 1-12, April 2002.

[35] Andrew. B. Smith, "An Examination of an Intrusion Detection Architecture for Wireless Ad Hoc Networks", in Proc. of the 5th National Colloquium for Information System Security Education, vol. 7, pp. 44-60, May 2001.

[36] Wayne A. Jansen "Countermeasures for Mobile Agent Security", Computer Communications, vol. 23, no. 17, pp. 1-14, 2000.

[37] Srivastava Shashank and G. C. Nandi "Self-reliant mobile code: a new direction of agent security", Journal of Network and Computer Applications, vol 37, pp.62-75, January 2013.

[38] Sandhya Armoogum and Asvin Caully "Obfuscation Techniques for Mobile Agent Code Confidentiality" Journal of Information & Systems Management, vol. 1, no. 1, 83-94, 2011.

[39] S. Venkatesan, C. Chellappan and T. Vengattaraman "Advanced Mobile Agent Security Models for Code Integrity and Malicious Availability Check" Journal of Network and Computer Applications, vol 33, pp. 661–671, 2010.

[40] Esfandi A, Rahimabadi AM "Mobile Agent Security in Multi Agent Environments using a Multi Agent-Multi Key Approach" in Proc. of 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009, pp. 438-442, August 2009.

[41] Tarig Mohamed Ahmed, "Using Secure-Image Mechanism to Protect Mobile Agent Against Malicious Hosts", World Academy of Science, vol 3,no. 11, pp. 375-380, 2009.

[42] Ismail Leila, "Evaluation of Authentication Mechanisms for Mobile Agents on Top of Java", in Proceedings of 6th IEEE/ACIS International Conference on Computer and Information Science, pp. 663–668, Octber 2007.

[43] Saxena Amitabh, Soh Ben "Authenticating Mobile Agent Platforms Using Signature Chaining Without Trusted Third Parties" in Proc. IEEE International Conference on E-Technology, E-Commerce and E-Service (EEE-05),pp. 282-285, April 2005.

[44] John Page, Arkady Zaslavsky, and Maria Indrawan "Countering Security Vulnerabilities in Agent Execution Using a Self Executing Security Examination" in Proc. of Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol. 3, pp. 1486-1487, July 2004.

[45] Joan Ametller, Sergi Robles and Ortega-Ruiz Jose A. "Self-Protected Mobile Agents" in Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol 1, pp. 362-367,July 2004.

[46] John Page, Arkady Zaslavsky, and Maria Indrawan "Security Aspects of Software Agents in Pervasive Information Systems", in Proc. 14th Australasian Conference on Information Systems (ACIS2003), November 2003.

[47] Neeran M. Karnik and Anand R. Tripathi "A Security Architecture for Mobile Agents in Ajanta" in Proc. Distributed Computing Systems 20th International Conference, pp. 402-409, 2000.

[48] Bennet S. Yee, "A Sanctuary for Mobile Agents" in Secure Internet Programming, Springer Berlin Heidelberg, pp. 261-273, 1999.

[49] J. Riodan and B. Schneier, "Environmental Key Generation Towards Clueless Agents" in Mobile Agents and Security, Springer Berlin Heidelberg, pp. 15-24, 1998.

[50] T. Sander, C. Tschudin, "Protecting Mobile Agents Against Malicious Hosts" in Mobile Agents and Security, Springer Berlin Heidelberg, pp. 44-60, 1998.

[51] Fritz Hohl, "Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts." in Mobile Agents and Security, Springer Berlin Heidelberg, pp. 92-113, 1998.

[52] G. Nguyen, T.T.Dang, L.Hluchy, M.Laclavik, Z.Balogh and I.Budinska, "Agent Platform Evaluation and Comparison", Published by Institute of Informatics, Slovak Academy of Sciences, Pellucid 5FP IST -2001-34519, June 2002.

[53] D.Horvat, D.Cvetkovic, V. Milutinovic, P. Kocovic and V. Kovacevic "Mobile Agents and Java Mobile Agents Toolkits", in Proc. of 33rd Hawaii IEEE International Conference on System Sciences (HICSS)-2000, Maui, Hawaii, USA, January 2000.

[54] S.S.Mudumbai, J. William and E.Abdelliah, "Anchor Toolkit- A Secure Mobile Agent System", Lawrence Berkeley National Laboratory, Published in eScholarship, Available at http://escholarship.org/uc/item/2594j56c , June 2008.

[55] D. Camacho, R. Aler, C. Castro and M.J. Molina, "Performance Evaluation of Zeus, Jade and Skeleton Agent Frameworks" in Systems, Man and Cybernetics,vol. 2, October 2002.

[56] Zeus, Available: www.labs.bt.com/projects/agents/zeus.

[57] F.Bellifemine, G.Caire, A.Poggi and G. Rimassa, "JADE: A White Paper", Available: http://exp.telecomitalialab.com , vol. 3, No. 3, September 2003.

[58] S.Vitabile, V.Conti, C.Militello and F.Sorbello, "An Extended JADE-S Based Framework for Developing Secure Multi-Agent Systems", Published in Computer Standards & Interfaces, Vol. 31, pp. 913–930, 2009.

[59] TILAB, JADE Specification, Available: http://jade.tilab.com/doc/programmersguide.pdf, September 2013.

[60] G.D. Kurundkar, N.A. Naik and S.D. Khamitkar, " Network Intrusion Detection Using SNORT", International Journal of Engineering Researchand Application, vol. 2, no. 2, pp. 1288-1296, March 2012.

[61] Wang Suli and Liu Ganlai, "File Encryption and Decryption System Based on RSA Algorithm", in International Conference Computational and Information Science, pp.797-800, October 2011.

APPENDIX A

LIST OF PUBLICATIONS

[1] **Jasleen Kaur**, Sharad Saxena and Anu Jain, "**Mobile Agents' Information Security in Ad-Hoc Network**", in IEEE International Conference on Computing of Power, Energy & Communication (ICCPEIC-2014),Chennai, pp.839-842 , 15-16, April 2014.

[2] **Jasleen Kaur** and Sharad Saxena, "**Securing Mobile Agents' Information in Ad-Hoc Network**", accepted in CONFLUENCE 2014,Noida, 25-26, September 2014.

Mobile Agent Information Security in Ad-Hoc Network

Jasleen Kaur
Research Scholar, SMCA
Thapar University
Patiala, Punjab, India
jassthapar@gmail.com

Sharad Saxena
Assistant Professor, SMCA
Thapar University
Patiala, Punjab, India
Sharad.saxena@thapar.edu

Anu Jain
Research Scholar, SMCA
Thapar University
Patiala, Punjab, India
Annujain.thapar@gmail.com

Abstract—**Mobile Ad hoc network (MANET), provides better speed, and make fully symmetric distributed network. As network topology changes dynamically in an unpredictable manner, we can use mobile agents, which is a program code that transfer itself along with data from one host to another and execute themselves accordingly. The host provides execution environment to mobile agent. These agents much flexible as it can travel from host to host, and are more prone to security threats. This paper focuses on providing security to mobile agents in ad-hoc environment and agents are developed by using technologies like JADE and implementing RSA algorithm to it.**

Keywords- Ad-hoc networks; mobile agents; JADE; Cryptography

I. INTRODUCTION

Mobile ad-hoc network (MANET) comprise set of mobile hosts. The mobile hosts don't need assistance of base stations for communication with each other. MANET has self-configuration and self-maintenance capability due to which MANET have received tremendous attention in recent years. Early research was focused on primarily providing cooperative and secure hostile environment. Although security is a long active research topic in wire-line network, MANET has some nontrivial challenges that need a security design. We cannot apply directly the already existing security protocol to MANET domain.

MANET is a collection of mobile devices which can form a temporary network if there is no fixed infrastructure provided. Each node communicates via wireless interface. The routing in Mobile Ad-hoc networks have two phases: route discovery and route maintenance. Route Discovery tells the route which source node S should follow to send packet to destination D. Route Maintenance let node S to detect whether one or more links along the route have failed or not.[manet1] The attacker can hamper the route discovery by impersonating the destination, by responding with corrupted routing information. To provide compendious security, both phase of MANET communication must be safe guarded [3].

Mobile Agents are the advance programming ideology, in which the program is a Software Agent. Mobile Agents are entities that are able to migrate from one hostile environment to another. After reaching to destination they are loaded and execute whole task in the way they are programmed. A number of archetypes exists which describes the Agents system; here, for discussing security issues we can use a simple one, which will contain only two main components: the Agents, and the Agent Platform [17]. Here, an Agent is a collection of code, data and control information with which it carries execution on the host they visit/reside. The Agent platform is the environment where the agent lives and operates. The platform where an Agent instantiates is known as Home Platform and it's the most trusted one for an Agent.

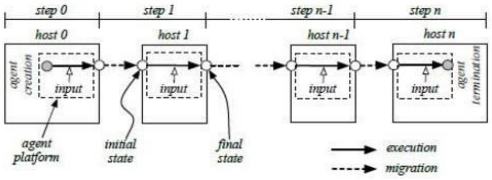


Figure 1. Agent Execution Model [2]

MANET can easily pray to attacks due to absence of centralized control, weak sanctuary of nodes and lack of well-defined boundary and thus firewalls are not reliable Quality of Service. The attacks are attempt made by attacker to detour security control to modify, expunge or peruse the data. Attacks can be categorized as:
First, Passive Attacks: These are of types in which the attacker only peruses the data without tampering it [13].
Second, Active Attacks: These are that type of attacker that can take malicious actions like tampering and deleting of data and even distorting network services [13].
The commonly identified threats to Agent over a network are categorized:

1) Agent attacking host
2) Malicious host attacking Agent
3) Malicious Agent attacking another Agent

Two approaches can be followed for protecting mobile agent in ad-hoc network:

a) Reactive Approach: First detect the threats and then react suitably [16].
b) Proactive Approach: Beware and try to prevent the data from being attacked by using some cryptographic techniques [16].

Various types of attacks on different layers of OSI is summarized in Table 1[14].

978-1-4799-3826-1/14/$31.00©2014 IEEE

TABLE I. SECURITY ATTACKS ON EACH LAYER IN MANET

Layer	Attacks
Application Layer	Repudiation, Data corruption
Transport Layer	Session hijacking, SYN flooding, Traffic analysis, monitoring, disruption, Jamming, interceptions, eavesdropping
Network Layer	Wormhole, black hole, resource consumption
Data Link Layer	Traffic Analysis, Monitoring , disruption
Physical Layer	Jamming, eavesdropping

Many Security Solutions are available which can resolve security issue. In this paper we focus on providing Transport Layer Security, which can be done by using Cryptographic Protocol.

The organization of paper is as, in Section 1 includes introduction to MANET and network attacks, Section 2 gives the related research work had been done. Section 3 describes agent development in JADE and data encryption using RSA. Finally Section 4 concludes the paper.

II. RELATED WORK

Different researchers have proposed various security and agent development approaches. Many cryptographic algorithms are used to provide confidentiality, authentication, integrity and non-repudiation. Security Modules such as tokens and smart card can be used to protect against physical attacks [2]. Zygmunt J. Haas has proposed a protocol for Secure Message transmission (SMT), which was designed to safeguard data transmission against malicious behavior of other nodes [1]. It explores the all paths of multi-path routing and makes operation more efficient and effective. Dorothy E. Denning implemented Intrusion Detection System (IDS) for mobile agent, by this we can trace the attacker, respond the target which will prevent attacker from establishing a better foothold and respond to source by restricting attacker's action [11].

DoS attacks can be limited by preventing the attacker from inserting routing loop. Some of the routing protocol that can prevent agent and its data from attacker is SECTOR mechanism which detects wormhole attack [6]. Security aware protocol can be used to defend against black-hole attack. It is based on on-demand protocols, like DSR or AODV [12]. In this security metric is added into RREQ packet, and different route discovery procedure is used. ARAN can be used to defend against spoofing and repudiation attack [8]. ARAN provides cryptographic certificates for end to end authentication.

Use of public key encryption is one of the most favorable and effective technique for providing basic security. These services include authentication, digital signature and encryption [15]. L. Zhou introduced a cryptographic technique for Ad-hoc network based mobile agents [15]. In their provision, a group of severs collectively with a master public-private key pair are established by CA (Certificate Authority). Every server has a master private key and key pair of all nodes. The master private key is generated by threshold cryptography.

Kong introduced network share CA functionality by making all nodes in network shareable [15]. A client gets the certification service by contacting to other node. If the node is neighborhood then the client can easily get certification service by one-hop broadcast request.

III. AGENT DEVELOPMENT AND DATA ENCRYPTION

JADE is a middleware that facilitates the development of multi-agent systems. It includes runtime environment where JADE agents exists, a library of classes that programmers have to use to develop their agents, and a suite of graphical tools that allows administration and monitoring the activity of running agents. Agents some time involve artificial intelligence algorithms to make them more intelligent. JADE is a Foundation for Intelligent Physical Agents (FIPA) complaint and defines an agent platform with three mandatory agent services [10]. All agents in JADE are design and kept in a repository called container. Agent has host address and the source address, so that is can migrate from one host to another. Agent has buffer to store data information and has unique identity for naming convention.

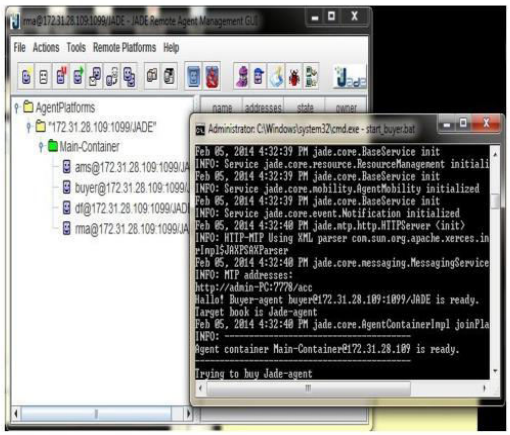


Figure 2. Creating Containers for Agents

Figure 2 above shows buyer agent creation snapshot in JADE platform.

In Figure 3 the seller agent is created and he is telling the price of various books. We are going to apply security protocol on the data buffer in which information is stored.

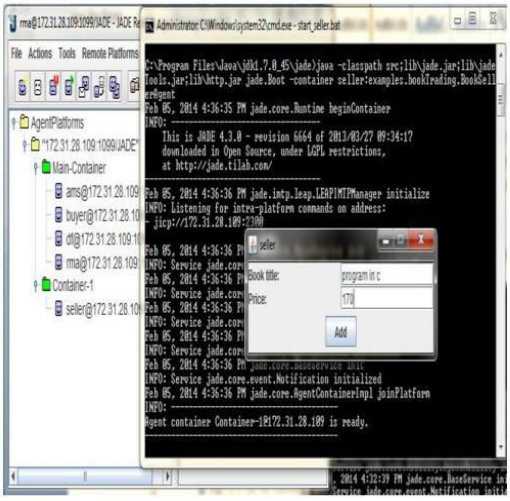


Figure 3. Creating Containers for Agents

Figure 4 below shows secure communication architecture for mobile agent.

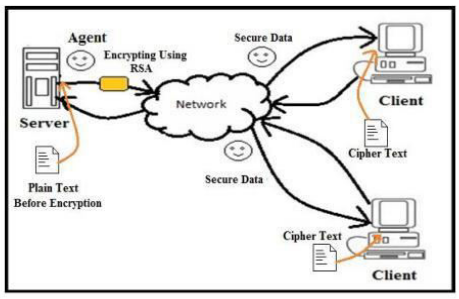


Figure 4. Agent Communication with Security Protocol

An agent named buyer (data initiator) is created in the container and another agent seller to respond to buyer with required data. The data transferred is the book name and price. As the agent is in ad-hoc network so the intruder can put all the efforts to get the information carried by the agent. But if the client itself will encrypt the data by using some public key and then will give that information to agent then cracking that key will be a difficult task. The private key will be with the buyer/server after getting the information by using private key he would be able to decode it. And by this no other client would be able to know what price marking had been done by others. The data obtained by seller is stored in buffer and RSA [9] cryptographic technique is used to encrypt it.

The idea is making agent's information safe and secure. The encryption scheme have mainly three categories :

- **Setup:** Which generates public and private key pair.
- **Encrypt:** Public key is applied to any arbitary string so that it would not be human redable.
- **Decrypt:** Decrypts message/string using corresponding private master key.

A brief RSA Algorithm is depicted in Algorithm 1 below:

ALGORITHM 1: RSA CRYPTOGRAPHY [9]

1. Select two large primes **p** and **q** such that $\mathbf{p \neq q}$.
2. $\mathbf{n \leftarrow p \times q}$
3. $\mathbf{\phi(n) \leftarrow (p-1) \times (q-1)}$
4. Select **e** such that $\mathbf{1 < e < \phi(n)}$ and **e** is coprime to $\mathbf{\phi(n)}$
5. $\mathbf{d \leftarrow e^{-1}}$ mod $\mathbf{\phi(n)}$ **//d** is inverse of **e modulo** $\mathbf{\phi(n)}$
6. Public_key ← **(e,n) //**To be announced publicly
7. Private_key← **d //**Kept Private
8. Return Public_key and Private_key

For encryption we have taken RSA parameters as: p = 3, q = 11, n = p * q = 3 * 11 = 33, now ϕ (n) = (p - 1) * (q - 1) = 2 * 10 = 20 and value of e is taken between 1 and ϕ (n), i.e. e=7. Further steps of computation are:-

- Compute a value for d such that (d * e) % ϕ (n) = 1. One solution is d = 3 [(3 * 7) % 20 = 1]
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of m = 2 is $c = 2^7 \% 33 = 29$
- The decryption of c = 29 is $m = 29^3 \% 33 = 2$

Any seller who wants to tell his price will encrypt his data using RSA public key. By this no other seller would be able to know what price the other one has mentioned. Hence no unauthenticated person would be able to manipulate, delete or read that data. When the buyer will get the price list of all the sellers he will encrypt the information using private key.

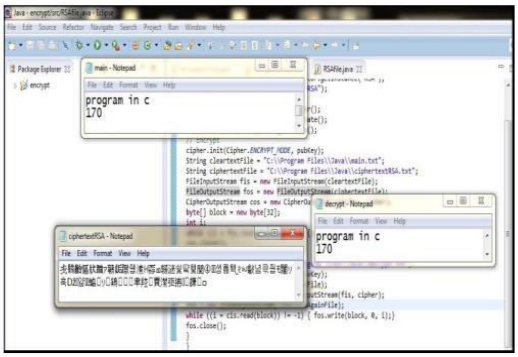


Figure 5. Source data (at Source), Encrypted data (at Internet) and Decrypted data (at Destination)

Before migrating from one place to another the data stored by agent is encrypted so that no other agent would be able to know what information agent is carrying.

IV. CONCLUSION AND FUTURE WORK

In this paper we have discussed about mobile agent and some possible threats to them. Security threats are due to their migration from one place to another for data forwarding. We have shown a brief description of mobile agent development in JADE technology and shown a secure method of agent based communication between two platforms. This paper is a start to implement security at transport layer to agents' data using RSA method between two hosts. As an extension we have planned to implement security concept to multi-agent and multi-platforms. Further we have planned in our next work to secure application layer by detecting intrusion to optimize agent based communication in terms of security and fast accessing.

ACKNOWLEDGMENT

This research paper is made possible through the immense help and support from everyone including college, friends and parent. And our deepest gratitude goes to our Guide for their expert guidance.

REFERENCES

[1] Panagiotis Papadimitratos and Zygmunt J. Haas, "Secure Data Transmission in Mobile Ad Hoc Network", in ACM workshop on Wireless Security , Sep 19, 2003.

[2] Bing Wu, Jianmin Chen and Jie Wu, "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks in Wireless/ Mobile Network Security" by Y.Xiao, X.Shen, Springer,2006.

[3] H Yang, Haiyun Luo and L Zhang,"Security in Mobile Ad Hoc networks: Challenges and Solution", IEEE Wireless Communications, Feb, 2004.

[4] Bindhu. R.,"Mobile Agent Based Routing Protocol with Security for MANET", Intrenational Journal of Applied Engineering Research,Dindigul, Vol 1, No1, 2010.

[5] Y. Hu, A Perrig, and D. Johnson, "Packet Leashes :A Defense Against Wormhole Attacks inWireless Ad Hoc Networks." Proc. of IEEE INFORCOM, 2002.

[6] S. Capkun, L. Buttyan, and J. Hubaux, "Sector: Secure Tracking of Node Encounters in Multi-hop Wireless Networks." Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[7] S. Yi, P. Naldurg, and R. Kravets, Security-Aware Ad-hoc Routing for Wireless Networks. Report No.UIUCDCS-R-2002-2290, UIUC, 2002.

[8] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, A Secure Routing Protocol for Ad Hoc Networks. Proc. of IEEE International Conference on Network Protocols (ICNP), pp. 78-87, 2002

[9] Wang Suli and Liu Ganlai, File encryption and decryption system based on RSA algorithm, in Int. Conf Computational and Information Science, pp.797-800, Oct.21-23 2011.

[10] TILAB, JADE Specification, http://jade.tilab.com/doc/programmersguide.pdf, Sep 14, 2013.

[11] Dorothy E. Denning, "An Intrusion Detection Model," IEEE Transactions on Software Engineering, 13(2), pp.222-232, Feb 1987.

[12] M. Zapata, Secure Ad Hoc On-Demand Distance Vector (SAODV). Internet draft, draft-guerrero-manet-saodv-01.txt, 2002

[13] Rupali Pathak and Durgesh Mishra "Distributed Intrusion Detection Scheme for Wireless Ad-Hoc Network : A Review", 6th International Conference on Software Engineering (CONSEG), Sept 5-7, 2012.

[14] E. Gajendra and S. Vijayan , "A literature Survey on security Challenges in Mobile Ad-hoc Networks", International Journalof Computer Application, Vol 84 – No 1, Dec 2013.

[15] Zhang Yi and Zhu Lina, "Key Management and Authentication in Ad-Hoc network based on Mobile Agent",Journal of Networks, Vol 4. No 6, Aug 2009.

[16] R. Arunkumar and A Annalakshmi, "A Recent Analysis of Intrusion Detection and Prevention System for Protectiong Range of Attack using Data Gathering Technique in MANET" , International Journalof Computer Application, Vol 85- No 8. Jan 2014.

[17] Wayne Jansen and Tom Karygiannis, "Mobile Agent Security", NIST Special Publication 800-19.

Thomas Gamer, Marcus Scholler and Roland Bless, "An Extensible and Flexible System for Network Anomaly Detection", Automatic Networking vol 4195, pp 7-108, 2006.

Jasleen Kaur recieved B.tech in Information Technology from Chhattisgarh Swami Vivekanand Technical University,Bhilai, Chhattisgarh, India in 2011.Now she is pursuing M.tech in Computer Science and Application from Thapar University, Patiala, Punjab, India. Her Research interest focusses on Mobile agents' security in ad-hoc network.

Sharad Saxena, did Ph. D. (CSE) 2012 and M. Tech. (CSE) in 2009. He has published 19 International Papers in the area of Ad-hoc and sensor networks in journal of repute. He has guided 6 M. Tech. Dissertations and one Ph.D. till date. His research interest includes Wireless Sensor Networks, Distributed Systems, and Mobile Computing, with a focus on Mobile Ad-Hoc Networks. Presently he is working as Assistant Professor in the department of Computer Application at Thapar University, Patiala, Punjab, India.

Anu Jain recieved B.tech in Computer Science from M.D. University,Panipat, Haryana, India in 2011.Now she is pursuing M.tech in Computer Science and Application from Thapar University, Patiala, Punjab, India. Her Research interest focusses on Security in Ad-Hoc Network

Securing Mobile Agent's Information in Ad-Hoc Network

Jasleen Kaur
Research Scholar, SMCA
Thapar University
Patiala, Punjab, India
jassthapar@gmail.com

Sharad Saxena
Assistant Professor, SMCA
Thapar University
Patiala, Punjab, India
sharad.saxena@thapar.edu

Mohd Abuzar Sayeed
Research Scholar, SMCA
Thapar University
Patiala, Punjab, India
abuzar.sayeed@gmail.com

Abstract- **A Mobile Ad hoc network (MANET) is a network that offers better speed, and formulates a fully symmetric distributed network. As topology of network changes vigorously in an erratic manner, we can use mobile agents, which travel autonomously within the network, execute solely in agent's environment, collect required facts/data and make conclusion according to the program carried by it. The execution environment is provided to mobile agent by host. Agents' have a very flexible nature as it can travel from host to host, and because of this are more prone to security threats. This paper focuses on providing security to mobile agent in ad-hoc environment by implementing cryptographic algorithm and Intrusion detection to prevent agents from attack. The agents are developed by using technologies like JADE and Intrusion detection is done using snort.**

Keywords- Ad-hoc networks; mobile agents; JADE; Cryptography; snort; intrusion detection

I. INTRODUCTION

Mobile ad-hoc network (MANET) encompasses set of mobile hosts. The mobile hosts don't need assistance of base stations for communication with each other. MANET has self-configuration and self-maintenance capability due to which they have received tremendous attention in recent years. Early research was focused on primarily providing cooperative and secure hostile environment. Even though security is a recent active topic for research in wired network, MANET has some nontrivial challenges that need a secure design. We cannot apply directly the already existing security protocol to MANET domain.

MANET is a collection of mobile devices which can form a temporary network if there is no fixed infrastructure provided. Each node communicates via wireless interface. The routing in mobile ad-hoc networks have two phases: route detection and route maintenance. Route detection tells which route source node should follow to destination who receives packets. Route Maintenance let node S to detect whether one or more links along the route have failed or not [1]. The attacker can hamper the route detection by spoofing target, by replying with corrupted routing facts [8]. To provide compendious security, MANET communication phases must be safe guarded [2].

Wireless transmission has a limited range for mobile devices these devices also serve as routers, that is, before reaching to the final destination a number of devices are required to route or relay a packet. Ad-hoc wireless networks should be organized rapidly everywhere and at any moment of time as it reduces the difficulty of setup of infrastructure. This type of network has application in a number of different areas. Some of the network some times consists of communication for military purpose by using wireless devices.

Mobile agent is an advance programming beliefs, which has strained tremendous amount of attention of researchers. Mobile agent technology has made distributed computing over the internet more secure and safe [1].

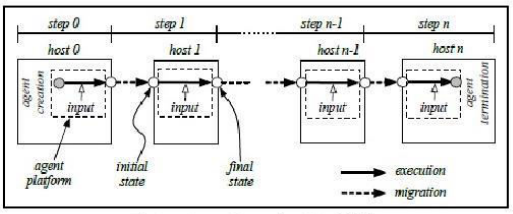


Fig 1. Agent Execution Model [4]

MANET can be easily attacked due to absence of centralized control and lack of well-defined boundary thus firewalls are not trustworthy for Quality Service. The attacker attacks many times and tries to detour security control to modify, expunge or peruse the data. Attacks can be categorized as:

a) Passive Attacks: These are of types in which without tampering the data attacker only examine it [5].

b) Active Attacks: Malicious actions like tampering and deleting of data and even distorting network services are done by this type of attacker [5].

The commonly identified threats to Agent over a network are categorized:

1) Agent attacking host: Malicious agents steals and modifies information on the host. These attacks are caused due to lack of adequate authentication and access control mechanisms. They can commit attacks like Denial

of service(DoS) by draining computational resources and rejecting platform services.

2) Malicious host attacking Agent: Attack on agent can be because of malicious host, this attack can be by stealing or modifying agent's data, corrupting or altering its code, refuse requested services, return fake system call values, reinitialize agent or even lapse it completely.

3) Malicious Agent attacking another Agent: Public methods of another agent are some times invoked by malicious agent to hinder with its work.

Two approaches can be followed for protecting mobile agent in ad-hoc network:

a) Reactive Approach: First detect the threats and then respond suitably [6].

b) Proactive Approach: Beware and try to prevent the data from being attacked by using some cryptographic techniques [6].

Various types of attacks on different layers of OSI is summarized in Table 1[7].

Table I. SECURITY ATTACKS ON EACH LAYER IN MANET

Layer	Attacks
Application Layer	Data duplicity, Repudiation
Transport Layer	SYN flooding ,Session hijacking, Traffic analysis, monitoring, disruption, Jamming, interceptions, eavesdropping
Network Layer	Resource utilization, Wormhole, black hole
Data Link Layer	Examining ,Traffic Analysis, interruption
Physical Layer	Jamming, eavesdropping

Various Security Solutions are available that can resolve security issue. In this paper we focus on providing Transport Layer Security, which can be done by using Cryptographic Protocol and Application Layer Security, by Intrusion Detection and Intrusion Protection System (IDIPS).

Table II. SECURITY SOLUTION FOR MANETS

Layer	Security Issues	Solutions
Application Layer	Identifying and intercepting worms, viruses, application abuses, malicious codes	Adequate security solution firewalls , **IDS**, etc.
Transport Layer	Securing point to point communication and authenticating it by encrypting data	Adequate security solution using **Public Key Cryptography**.
Network Layer	Preserving ad-hoc network routing and protocol forwarding	For source authentication and integrity of message there is no effective mechanism but some routing protocols can be used to overcome black hole, packet leashes, etc.
Data Link Layer	The wireless MAC protocol is protected and security is provided to support link layer	For preventing traffic analysis and monitoring layer there is no efficient mechanism
Physical Layer	Signal jamming, Denial of service attacks, are prevented	Using Spread spectrum mechanism

The organization of paper is as, in Section I includes introduction to MANET and network attacks, Section II gives the related research work had been done. Section III describes agent development in JADE and data encryption and Intrusion Detection using SNORT. Finally Section IV concludes the paper.

II. RELATED WORK

Different researchers have proposed various security and agent development approaches. Many cryptographic algorithms are used to provide confidentiality, authentication, integrity and non-repudiation. Security Modules such as tokens and smart card can be used to protect against physical attacks [4]. Zygmunt J. Haas has proposed a protocol for Secure Message transmission (SMT), which was designed to safeguard from malicious behavior of other nodes during data transmission [8]. It explores the all path of multi-path routing and makes operation more efficient and effective. Dorothy E. Denning implemented Intrusion Detection System (IDS) for mobile agent, by this we can trace the attacker, respond the target which will prevent attacker from establishing a better foothold and respond to source by restricting attacker's action [9].

Attacks like DoS can be restricted by intercepting the attacker from entering routing loop. Some of the routing protocol that can prevent agent and its data from attacker is SECTOR mechanism which detects wormhole attack [10]. Security aware protocol can be used to defend against black-hole attack. It is based on on-demand protocols, like DSR or AODV [11]. Security metric is appended into packet named RREQ; more over various route detection procedures are used. For defending against spoofing and denial attack ARAN is used [12]. ARAN provides cryptographic certificates for end to end authentication.

Use of public key encryption is one of the most favorable and effective technique for basic security. These services include authentication, encryption and digital signature [13]. L. Zhou introduced a cryptographic technique for mobile agents on Ad-hoc network [13]. By using CA (Certificate Authority) a master public-private key pair is established collectively for a group of severs. Master Private Key and key pair of all nodes is provided to every server. Threshold cryptography is used for generation of master private key.

Kong introduced network share CA functionality by making all nodes in network shareable [13]. A client gets

the certification service by contacting to other node. If the node is neighborhood then the client can easily get certification service by one-hop broadcast request.

III. AGENT DEVELOPMENT, DATA ENCRYPTION AND PREVENTION

The middleware that facilitates the evolution of multi-agent systems is called JADE. The JADE agents exist with in a runtime environment, to develop agents the programmers are provided with a library of classes that they can use, monitoring and administrating is done by use of a suite of graphical tools that allows management and running of agents. Some times artificial intellegence is also involved by agents to make them more intelligent. JADE is a Foundation for Intelligent Physical Agents (FIPA) complaint and defines an agent platform with three mandatory agent services [14]. These services include autonomy, mobility and reactivity. All agents in JADE are design and kept in a repository called container. Host address and source address both are contained with in agent, so that agent can migrate from one host to another. Agent has buffer to store data information and has unique identity for naming convention

The ongoing and very popular research work area for information safety is distributed intrusion detection system, but mostly Intrusion detection prototypes made of single host and network examiner are used distributed IDS which also consist of centralized controller component also. The intrusion records are thrown to the centralized controller module by individual monitor that performs investigation on the received information from the other monitors [16].

The primary concern of this kind of presented centralized systems is [16]:

1. The intrusion detection and reaction real-time is not good.
2. Single host duplicate with the composed facts, hence the supervised network is restricted. The network can get overloaded due to lots of data collection.
3. Adding new hosts can overload the centralized controller and can significantly increase, hence IDS is non-scalable
4. The system lacks dynamic configuration ability and also flexibility of the system is not up to mark.

The cooperation between different IDS lacks. The combination of Network IDS (NIDS) and Host IDS (HIDS) is needed to use.

To detect and to prevent system and network from security threats Intrusion detection and Intrusion protection system(IDIPS) is used which make computer network and system more secure and safe [17]. For keeping information systems secure a valuable tool is Intrusion detection and Intrusion protection system (IDIPS) [17]. By using security tool like Intrusion detection and Intrusion protection system (IDIPS) we can examine, analyze, and act in response to violations in possible security against computer and network systems. Because of unauthorized external intruders the result of break in attempts are trying to cooperate the system or exploitation of authority of internal privileged users. IDIPS continue to develop and make new systems. Recognition methodologies that can be utilized by newer systems create some confusion when we try to better understand it [17]. Previous and existing area of work primarily focuses on improvising or explaining one or more methodologies. Assessment of one method offers some work besides of the proposed methodology.

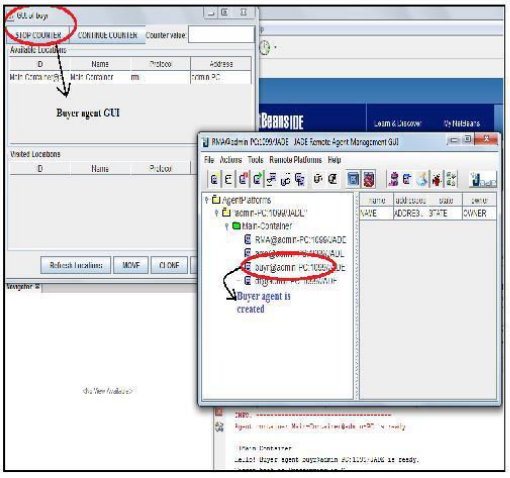


Fig 2. Creation of Container and Agent

In Fig 2 above shows creation of buyer agent in JADE platform snapshot.

In Fig 3 an agent named seller agent is created and who tells the price of various books. Security protocol is going to be applied on the buffered data in which information is stored.

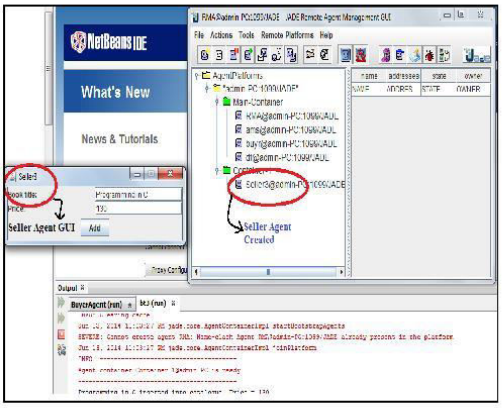


Fig 3. Creation of Container and Seller agent

In Fig 4 below shows the selected seller with least coatted price.

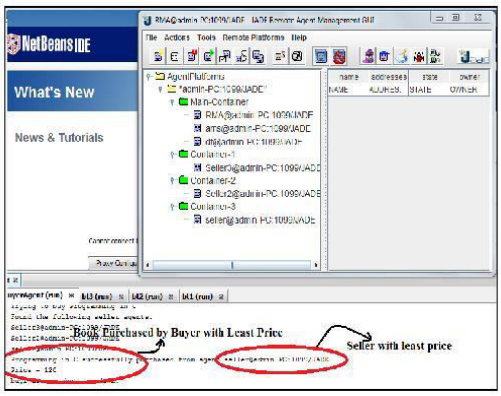


Fig 4. Buy from the Seller agent with least coatted price

In Fig 5 below shows secure communication architecture for mobile agent.

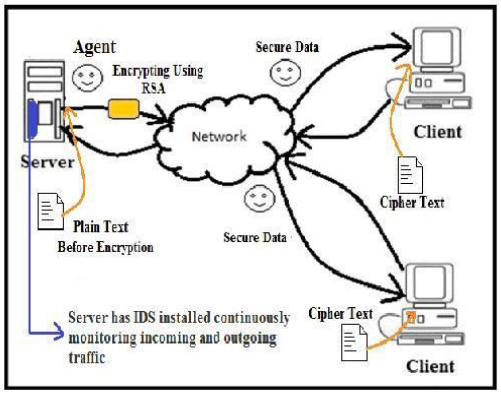


Fig 5. Agent Communication with Security Protocol and IDS

In our case the information transfer is an example of book name and price. An agent named buyer (data initiator) is created in the main container and other agents named seller, seller 2, seller 3 who respond to buyer fired query. As the agent is in ad-hoc network so the intruder can put all the efforts to get the information carried by the agent. But if the client itself will encrypt the data by using some public key and then will give that information to agent then cracking that key will be a difficult task. The private key will be with the buyer/server after getting the information by using private key he would be able to decode it. And by this no other client would be able to know what price marking had been done by others. The data obtained by seller is stored in buffer and RSA [15] cryptographic technique is used to encrypt it.

The idea is making agent's information safe and secure. The encryption scheme have mainly three categories :

- **Setup:** Which generates public and private key pair.
- **Encrypt:** Public key is applied to any arbitary string so that it would not be human redable.
- **Decrypt:** Decrypts message/string using corresponding private master key.

A brief RSA Algorithm is depicted in Algorithm 1 below:

ALGORITHM 1: RSA CRYPTOGRAPHY [15]

1. Choose two large primes **a** and **b** such that **a ≠ b.**
2. **m← a x b**
3. **ȼ(m)← (a-1) x (b-1)**
4. Choose **e** such that **1 < cp < ȼ(m)** and **e** is co-prime to **ȼ(m)**
5. **k←cp^{-1}** mod **ȼ(m)** **k** is inverse of **cp modulo ȼ(m)**
6. Public_key **←** **(cp,m)** //To be announced publicly
7. Private_key**←** **k** //Kept Private
8. Return Private_key and Public_key

For encryption we have taken RSA parameters as: a = 3, b = 11, m = a * b = 3 * 11 = 33, now ȼ (m) = (a - 1) * (b - 1) = 2 * 10 = 20 and value of cp is taken between 1 and ȼ (m), i.e. cp=7. The alphabets' numerical value will be a=0,b=1,c=2… y=24,z=25. Further steps of computation are:-

- Calculate value for k such that
  (k * cp) % ȼ (m) = 1.
  One of the answers to above statement is
  K → 3 [(3 * 7) % 20 = 1]; Now
- Public key is (cp, m) → (7, 33)
- Private key is (k, m) → (3, 33)
- Message given is 'Program in C and 170'
- The encryption of character 'p' in message is p = 15, c = 15^7 % 33 = 27; for character 'r' is 17, c = 17^7%33 =8; for character 'o' = 14, c= 14^7%33= 20… Similarly all the rest characters will be encrypted using public key e.
- The decryption of message is for c = 27, m = 27^3 % 33 = 15 (15='p'); for c =8, m= 8^3%33=17 (17='r'); for c=20, m= 20^3%33=14 (14 ='o')… Similarly rest characters will be decrypted using private key d.

Any seller who wants to tell his price will encrypt his data using RSA public key. By this no other seller would be able to know what price the other one has mentioned. Hence no unauthenticated person would be able to manipulate, delete or read that data. When the buyer will get the price list of all the sellers he will encrypt the information using private key. Below in Figure 5 we had taken an example having book name "program in c" and whose price is "170". This information is stored in buffer

in encrypted form only the buyer can decrypt it by using its private master key.

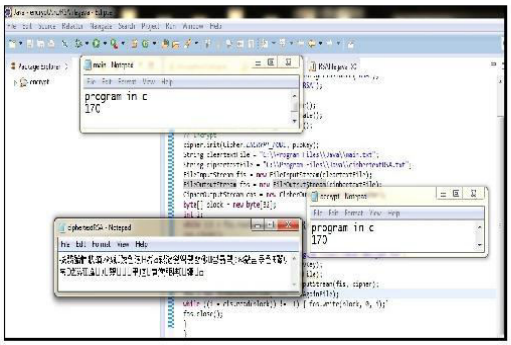


Fig 6. Source data (at Source), Encrypted data (at Internet) and Decrypted data (at Destination)

Before migrating from one place to another the data stored by agent is encrypted so that no other agent would be able to know what information agent is carrying.

An open source intrusion detection and intrusion protection system is used to prevent system from attacks. The used system is snort.

The network intrusion detection and protection system which also is an open source is snort that utilizes rule-based language, the benefits of signature is also combined, along with rules and irregularity based examination methods [18]. One of the most widely organized intrusion detection and intrusion protectiion methodology is snort. Mainly to unreceptively examine network traffic snort is used. Also snort generate alerts when threats are detected. In our paper we are using snort Inline mode which is used to drop packets and interrupt network traffic [18]. The main quality of Inline mode are:

- Forwarding and routing of network traffic is configured and deployed on a server by snort in contrast to only sniff network traffic
- The rules of Snort for only "alert generation" are changed into "dropping rules". Here we are also including iptables firewall application along with Snort Inline mode who interacts with iptables to obtain and route traffic of network.

IV. Conclusion and Future Work

In this paper we have discussed about mobile agent and some possible threats to them. Security threats are due to their migration from one place to another for data forwarding. We have shown a brief description of mobile agent development in JADE technology and shown a secure method of agent based communication between two platforms. This paper is a start to implement security at transport layer to agents' data using RSA method between two hosts. Further we secured application layer by detecting intrusion using snort. As an extension we have planned to implement security concept to multi-agent and multi-platforms. Further we have planned in our next work to optimize agent based communication in terms of security and fast accessing.

Acknowledgment

This research paper is made possible through the immense help and support from everyone including college, friends and parent. And our deepest gratitude goes to my Guide for his expert guidance.

References

[1] Bindhu. R.,"Mobile Agent Based Routing Protocol with Security for MANET", Intrenational Journal of Applied Engineering Research,Dindigul, Vol 1, Nol, 2010.

[2] H Yang, Haiyun Luo and L Zhang,"Security in Mobile Ad Hoc networks: Challenges and Solution", IEEE Wireless Communications, Feb, 2004.

[3] Wayne Jansen and Tom Karygiannis, "Mobile Agent Security", NIST Special Publication 800-19,1998.

[4] Bing Wu, Jianmin Chen and Jie Wu, "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks in Wireless/ Mobile Network Security" by Y.Xiao, X.Shen, Springer,2006.

[5] Rupali Pathak and Durgesh Mishra "Distributed Intrusion Detection Scheme for Wireless Ad-Hoc Network : A Review", 6th International Conference on Software Engineering CONSEG, Sept 5, 2012.

[6] R. Arunkumar and A Annalakshmi, "A Recent Analysis of Intrusion Detection and Prevention System for Protectiong Range of Attack using Data Gathering Technique in MANET" , International Journalof Computer Application, Vol 85, No 8. Jan 2014.

[7] E. Gajendra and S. Vijayan , "A literature Survey on security Challenges in Mobile Ad-hoc Networks", International Journal of Computer Application, Vol 84 , No.1, Dec 2013.

[8] Panagiotis Papadimitratos and Zygmunt J. Haas, "Secure Data Transmission in Mobile Ad Hoc Network", in ACM workshop on Wireless Security , Sep 19, 2003.

[9] Dorothy E. Denning, "An Intrusion Detection Model," IEEE Transactions on Software Engineering,Vol 13, No.2, pp. 222-232, Feb 1987.

[10] S. Capkun, L. Buttyan, and J. Hubaux, "Sector: Secure Tracking of Node Encounters in Multi-hop Wireless Networks." Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[11] M. Zapata, Secure Ad Hoc On-Demand Distance Vector (SAODV). Internet draft, draft-guerrero-manet-saodv-01.txt, 2002.

[12] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, A Secure Routing Protocol for Ad Hoc Networks. Proc. of IEEE International Conference on Network Protocols (ICNP), pp. 78-87, 2002.

[13] Zhang Yi and Zhu Lina, "Key Management and Authentication in Ad-Hoc network based on Mobile Agent",Journal of Networks, Vol 4. No 6, Aug 2009.

[14] TILAB, JADE Specification, http://jade.tilab.com/doc/programmersguide.pdf, Sep 14, 2013.

[15] Wang Suli and Liu Ganlai, File encryption and decryption system based on RSA algorithm, in Int. Conf Computational and Information Science, pp.797-800, Oct.21-23 2011.

[16] Liu, Jianxiao, and Li Lijuan. "A Distributed Intrusion Detection System Based on Agents." Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on. Vol. 1. IEEE, 2008

[17] Mudzingwa, David, and Rajeev Agrawal. "A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS)." Southeastcon, 2012 Proceedings of IEEE. IEEE, 2012.

[18] Valli, Craig. "Wireless Snort-A WIDS in progress." In Australian Computer, Network & Information Forensics Conference, pp. 112-116. 2004.

APPENDIX B

COMMANDS FOR INSTALLATION AND CONFIGURATION

- Installation of DAQ can be done using following command:

```
sudo tar zxvf daq-0.5.tar.gz
cd daq-0.5
```

- After DAQ installation libdnet is installed by following commands:

```
sudo tar zxvf libdnet-1.11.tgz
cd libdnet-1.11/
sudo ./configure
sudo make
sudo make install
sudo ln -s /usr/local/lib/libdnet.1.0.1 /usr/lib/libdnet.1
```

- Installation of SNORT

```
sudo tar zxvf snort-2.9.3.tar.gz
cd snort-2.9.3
sudo ./configure --prefix=/usr/local/snort --enable-sourcefire
sudo make
sudo make install
sudo mkdir /var/log/snort
sudo mkdir /var/snort
sudo groupadd snort
sudo useradd -g snort snort
sudo chown snort:snort /var/log/snort
```

- SNORT ruleset for IDS and IPS configuration is done in following manner:

```
sudo tar zxvf snortrules-snapshot-2930.tar.gz -C /usr/local/snort

sudo mkdir /usr/local/snort/lib/snort_dynamicrules

sudo cp /usr/local/snort/so_rules/precompiled/Ubuntu-12-4/i386/2.9.2.0/

/usr/local/snort/lib/snort_dynamicrules

sudo touch /usr/local/snort/rules/white_list.rules

sudo touch /usr/local/snort/rules/black_list.rules

sudo ldconfig
```