The International Journal Of Engineering And Science (IJES)
|| Volume || 3 || Issue || 6 || Pages || 23-26 || 2014 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Detection of local intrusion to avoid damage of global effects

N. Alekhya[1], Dr.N.Chandra SekharReddy[2], S.Srinivas[3]
[1,] *Student M-Tech IT Dept,* [2,] *Professor CSE Dept,* [3,] *Asst.Prof IT Dept*
Institute of Aeronautical Engineering, Hyderabad, Andhrapradesh, India-500043.

----------**ABSTRACT**----------

The present security technologies have some drawbacks with the security performance and cost. The paper presents an automatic security chain to accomplish security protection. It uses the concept called swarm intelligence as a reference to the "intrusion detection system" but it has a difficulty to find the exact local actions which are performed at the host & nodes in the networks. By this intruders may damage the local actions which are performed at the host and nodes of the networks.

The paper implements an algorithm that helps to find the correct local actions at host and nodes in the networks and also reduces the global effects damaged by the intruders.

Keywords: *DLAD, Swarm Intelligence, Intrusion Detection, AODV, DSDV.*

Date of Submission: 07 May 2014 Date of Publication: 10 June 2014

I. INTRODUCTION

The Internet changing and possibilities and opportunities are limitless. The important factor about security mechanisms, that they should be designed to avoid non authorized access of system resources and also data. But, it is not possible to prevent completely which is unrealistic. We can perform some attempts to detect this intrusion by that action may be taken and later the damage will be repaired. This field of research is called Intrusion detection system. Traditionally, host based intrusion detection focuses mainly on the changes made to the system and also attempts to identify changes to system configuration files, selected executables and also key files.

II. LITERATURE SURVEY

Bonabeau, E., Dorigo, M., Theraulaz, G developed a "swarm intelligence" as a reference to build an "intrusion detection system" that is simple in both the structure and detection algorithms but can also detect "complicated intrusion" [1].

Pitcher, T.J., Partridge, B.L., Wardle paper presents an automatic security chain including preparation, detection and response. The components in the chain cooperate closely to accomplish security protection [2].

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm is based on the concept of the classical Bellman-Ford Routing Algorithm for certain improvements to Ad hoc mobile network. In this every mobile station contains a routing table that consists all available destinations [5] and each entry contains sequence numbers.

Disadvantages: This DSDV algorithm must frequently update its routing table, which must have battery power and also some bandwidth even when the network is idle.

The Ad Hoc on-Demand Distance Vector Routing (AODV) uses traditional routing tables, one entry per destination. This is contrast to DSR, i.e., for each entries of destination [5] may have multiple route cache. It establishes a route to the destination only on demand so that it is a reactive protocol and avoids the counting-to-infinity problem by using sequence numbers.

Advantage: This protocol will establishes a route on demandand the sequence numbers of destinations are used to find the latest route to destination. The setup of connection delay is lower. **Disadvantage:** AODV takes much time to establish a connection and initial communication is heavier than other approaches.

III. PROBLEMS IN EXISTING METHOD

When we were implementing intrusion detection on our systems, the best known product for this host based intrusion detection was Tripwire. The commercial product is too expensive and also open source version lacked the ability to automatically store and analyze the database on a remote machine of the tracked machine. Tripwire primarily analyzes files on disk and does not track processes in memory. And also by using Swarm Intelligence [1] [4] in Firm Computer Network Securities to detect the Intrusion detection, there is a problem in finding the local actions matching the global effects. The methods DSDV, AODV are used today for finding the local actions are proved to be ineffective.

IV. PROPOSED METHOD

The paper presents an algorithm for building an automatic security chain including detection, preparation and response. The components in this chain cooperate closely to accomplish security protection. By using swarm intelligence into Firm Computer Network Security techniques to detect "complicated intrusion", there is a difficulty to find the correct local actions matching the necessary global effects .So, intruders can damage the local actions performed at Host & Nodes in the Networks. So, we have implemented "Distributed Local Action Detection" (DLAD) Algorithm which helps to find the correct local actions at Nodes in the Networks.

Intrusions detection is a task for good computer security strategy an efficient and effective plan for recovery is a necessity. Maintaining well-administered and up-to-date systems will minimize the occurrence of intrusions, but they will inevitably happen, so it is critical that recovery be efficient and thorough. In conventional systems, intrusion recovery is difficult and time-consuming, in terms of both system down-time and administrator time. It requires a significant amount of the administrator's time because there are many, error-prone steps involved in returning a compromised computer system to a safe state.

Pseudo code:
/* Reading the nodes and Setting the path*/
Input: Give the name of file containing the number of nodes and coordinates as the command line argument.

Process Steps:
Step 1:Loading of number of nodes and Coordinates into array.
Step 2:Set path to all nodes by calculating using the statement dist[pos][i]=Math.sqrt(Math.pow(c oords[pos][0] - coords[i][0], 2) + Math.pow (coords[pos][1] - coords[i][1], 2))
If the path is less than the radius then set path to -1
Step 3:Initialize the variables allThread Created =false and
intialRREQ=false
Step 4:Create the threads, no of threads created is equal to no of nodes present in the input file
Step 5:Start all the threads by using the statement (new Thread (new
DSR (i))).start ();

/* Sending Packets*/
Step 1:Assign all the variables i.e. start node, port number, and time Delay
Step 2:Using sleep () method, make some delay for the thread using the statement
Thread.sleep((int)(delay * 50));
Step 3:Create a new object sendPacket for DatagramPacket class by passing the data,
length of data, ip address, and port no's as the parameters.
Step 4:Send the packet using the statement socket.send(sendPacket);
/* receive packets*/
Step1: Create new object receivePacket for DatagramPacket class by passing the data
and data length as parameters
Step2:Receive the data using the statement socket.receive (receivePacket)
Step3: declare a variable named message [numnodes+2] as String type.
Step 4:Assign the values of the variables messageLength=0 and shouldsend=true.
Step 5:Retrive the data using getData() method and divide it into number of tokens using
String Tokeniser StringTokenizer st =new StringTokenizer(new String(
receivePacket.getData(), 0, receivePacket.getLength()) , ":")
Step 6: Using the method hasMoreTokens() and nextToken() present
in the String Class assign all tokens present in the data to the
message Variable. message[messageLength++]
=st.nextToken();
Step 7:The value of the variable shouldsend is changed to false when
the condition -1>=Integer.ParseInt (Message[i]) is sarisfied.
Step 8:From the 1st position to the last position in the variable
Message, concatenate each of the character present in variable
to the variable showMessage.
Step 9:Repeat above step except for position 2nd to the variable
DisplayMessage
Step 10:If(message [2].charAt(0)= =names [pos]) then append the

Display message display.append("\n"+ "RREP " + message[1]
+ " complete: " + displayMessage)

Step 11: If the above condition not satisfies then send packet to the previous node and append the message.(new Thread (new SendPacket (sendMessage, 800 0 + j, dist[pos][j]))).start()display.append ("\n" + "Sending R REP " + message[1] + ": " + displayMessage)

V. EXPERIMENTAL RESULTS AND ANALYSIS

The performance of the algorithm depends on some metrics such as, Packet delivery, Packet dropped and Throughput have been considered as analytical observation.

Packet delivery: This mainly illustrates the level of delivered data to destination by the ratio of number of delivered packet to destination.

Packet Dropped: This illustrates packets dropped while the simulation. If the numbers of lost packets are less it means the better performance of the protocol.

Throughput: This illustrates average rate of message delivery which is successfull over a communication channel. This throughput data will pass through a certain network node.

1. Packet Received vs. Node:

Form the figure we can notice that the received packets for DLAD are higher than DSDV and AODV. This packets received have been calculated by changing the nodes number with the fixed simulation time. Between DSDV and AODV, AODV can have more successful transfer than the DSDV.

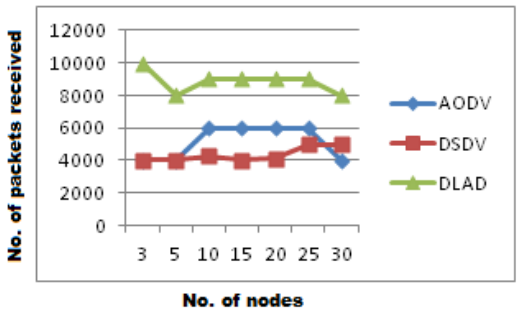

Fig: Comparision of AODV,DSDV,DLAD with packet received Vs. Node.

2. Throughput vs. Node:

Figure shows the comparison of throughput here DLAD shows higher throughput than the DSDV and AODV because its routing overhead is less than others. Also the rate of packet received for DSDV is less than the AODV.

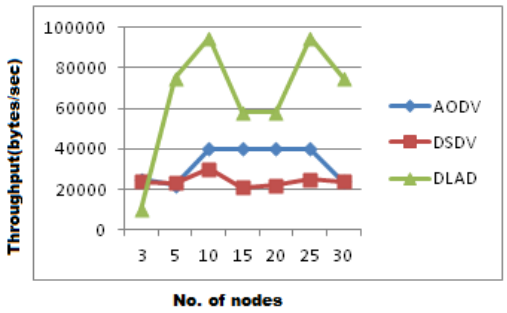

Fig: Comparision of AODV, DSDV, DLAD with Throughput Vs. Node.

3. Packet Delivery fraction vs. Pause Time:

Packet delivery fraction is the ratio between the number of packets originated by the application layer sources and the number of packets received at the final destination. It will also describes the loss rate by the transport protocols, which affects the maximum throughput that the network can Support. This simulation chooses 0, 100, 200, 300, 400, 500, 600,700 , 800 and 900seconds pause time. This simulation generates 50 nodes.

Figure above shown at pause time 0 Seconds (high mobility) environment, DLAD and AODV outperforms DSDV in high mobility.

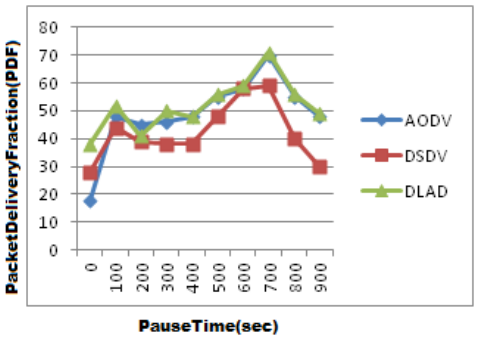

Fig: Comparision of AODV,DSDV,DLAD with Packet Delivery Vs Pause Time.

VI. CONCLUSION

In this paper, by using swarm intelligence into Firm Computer Network Security techniques; i.e., the emergent collective intelligence of group of small agents, but there is difficult to find local actions matching the global effects. So, we have implemented an Algorithm which helps to find the correct local actions at Nodes in the Networks.

VII. REFERENCES

[1] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press,New York, NY (1999).

[2] Pitcher, T.J., Partridge, B.L., Wardle paper presents an automatic security chain., C.S.:Ablind fish can school. Science194(4268) (1976) 963–965. Available from: http://www.sciencemag.org/cgi/content/abst ract/194/4268/963.

[3] A. Wespi, H. Debar, and M. Dacier, ``An Intrusion-Detection System Based on the Teiresias Pattern-Discovery Algorithm," Eicar '99, Aalborg, Denmark, Feb. 27 - Mar. 2, 1999.

[4] Meyer, K.D., Nasut, S.J., Bishop, M.: Stochastic diffusion search: Partial function evaluation in swarm intelligence dynamic optimisation. In braham, A., Grosan.

[5] [Md. Anisur Rahman, Md. Shohidul Islam, Alex Talevski, Performance Measurement of Various Routing Protocols in Ad-hoc Network , IMECS 2009, March 18 - 20, 2009, HongKong.