ISSN:0975-9646

Improving Trust Management for Effective Collaborative Intrusion Detection Network

P.Ayesha Barvin[1], G.Bakkiyaraj[2]

[1]*Dhanalakshmi Srinivasan College of Engineering, Perambalur, Tamil Nadu, India.*
[1]*Asst.Prof, Department of CSE,* [2]*P.G Student,*
[2] *Roever EngineeringCollege,*

Abstract:An effectiveness of detecting intrusion among a Collaborative Network Systems, by authorizing IDS to collaborate and share their knowledge to enhance the overall accuracy and detecting new intrusion. To evaluate intrusion we need of trust and trustworthiness. To improve the Trust we propose a cluster based hierarchical trust management protocol adopts Hop distance method. For trustworthiness of HIDS collaboration system, We focused on acquaintance management where each HIDS selects and maintain a list of Collaborators from which they can consult about intrusions. It evaluates both false positive (FP) rate and false negative(FN) of its neighbouring HIDS. An Acquaintance management of probation list algorithm allows each HIDS to effectively select a set of Collaborators and to detect intrusion. We evaluate our approach based on simulated CIDN, demonstrating its improved efficiency, scalability for Collaborative intrusion detection in comparing with other existing models

***Index terms*: Collaborative Intrusion Detection Network(CIDN), Host based Intrusion Detection System(HIDS),Intrusion Detection System(IDS),Trust Management.**

1. INTRODUCTION:

The purpose of intrusion detection is to help computer systems prepare for and deal with attacks. Intrusion detection systems collect information from a variety of sources within computer systems and networks. For most systems, this information is then compared to predefined patterns of misuse to recognize attacks and vulnerabilities. In a Network Intrusion Detection System (NIDS) and Host Intrusion Detection System(HIDS) are an

(CIDN) is an IDS network intended to overcome this weakness by having each peer IDS benefit from collective information, knowledge and experience shared by other peers. An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewalls simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host.

Since the HIDS are distributed and may have different expertise levels in detection intrusions. A HIDS may also turn malicious due to runtime bugs, having been updated with the faulty new configuration, or having been deliberately made malicious by its owner. Intrusion can have several forms such as worms, sparmware, viruses, Deniel of service attacks (DOS), malicious logins, etc. The potential damage of these intrusions can be significant if they are not detected promptly. An example is the Code Red Worm [1] that infected more than 350000 systems in less than 14 hours in 2001 with the damage cost of more than two billion dollars this is explained in the paper of Code-Red Worm. There are various attacks like Betrayal attack, Sybil attack, New Comer attack etc. Large scale peer-to-peer systems face security threats from faulty or hostile remote computing elements. One approach to preventing these “Sybil attacks” is to have a trusted agency certify identities [9].

The Conflicker worm which appeared first in 2008-2009 and very short period it spread in the world with the damage cost estimated around 9.1 billion dollars [2].

Host based system is a piece of software is loaded onto a system to detect intrusion. The software uses log files or system auditing agents, which look at communication traffic. The software then checks the integrity of system files. Agents are installed on publicly accessible servers such as corporate mail servers or application servers. The agents then report events to a central console that is protected by agent software [12].

Log file analyzers -analyzes log files for patterns that indicate intrusion. File system monitor -monitors the system to check for integrity of files and directories. Connection analyzers to monitor connection attempts. Kernel based analyzer-to detect malicious activity on a kernel, the hosts within the private network will have an intrusion detection system that will send alerts to the agent console from where they are analyzed.

A trust management is the solution for evaluating trustworthiness of Host based Intrusion Detection Systems in Collaborative Intrusion Detection Networks. Acquaintance management of probation list is proposed so measure the uncertainty in estimating the likely future behaviour of HIDSes[3]. The measured uncertainty allows trust management to employ an adaptive message exchange, resulting in good scalability. Acquaintance management algorithm is also demonstrated to have the properties of fairness and convergence and to provide incentive for collaboration.

In order to achieve the goal of identifying the best route to optimize and for trustworthiness uses the trust aggregation and trust formation. It would generate mathematical model to derive "ground truth and can obtain objective trust against which subjective trust obtained as a result of executing the trust management. It can be compared and validated [4].

This work proposes two capabilities, (a) early warning when pre attack activities are detected. (b) detecting and isolating compromised nodes by trust mechanism and voting based peer-level protocol.

2. PROPOSED SYSTEM ARCHITECTURE

In a proposed system, the same procedure is followed which is acquaintance management algorithm but to increase the trustworthiness and trust management , we adapts Acquaintance management table also used here to maintain the information about each node The root finds the length of the longest prefix of the ID it shares. Then it sends a multicast message that reaches all existing nodes sharing the same prefix. These nodes then add the new node to their routing tables. The next once may take over being the root for some of the roots objects. The nodes will contact the new node to provide a temporary neighbourhood list. The new node then performs an iterative nearest neighbour search to fill all levels in its routing table. The nodes are collected from wireless link which is a wireless local area network (WLAN) links two or more devices over a short distance using a wireless distribution method, usually providing a connection through an access point for Internet access.

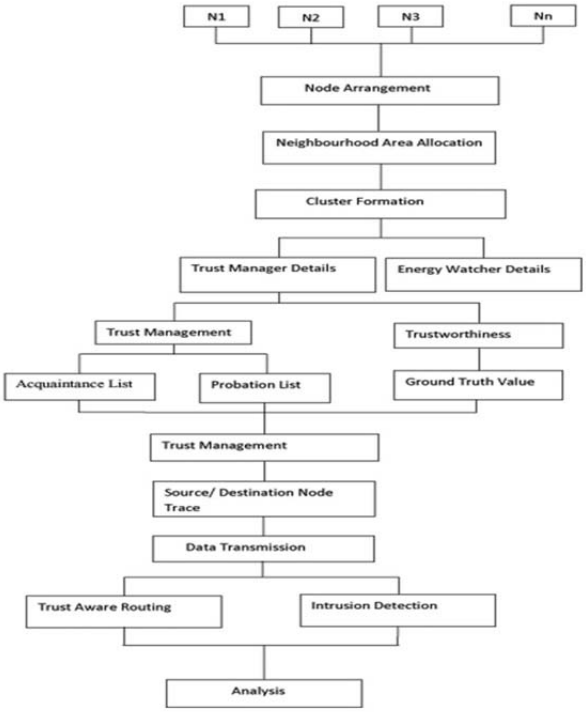


Fig. 1. CIDN architecture Design

Clusters are usually deployed to improve performance and availability over that of a single computer, while typically being much more cost effective than single computers of comparable speed or availability. The far as is reasonable, as well as the resources again forecast by time period into the future as far as is reasonable. The head is assigned and the particular node should be a management of transmission. There are various attacks such as Betrayal attack, It is when a trusted peer suddenly turns in to malicious one and send false feedback. A trust management system can be degraded dramatically because of this type of attacks. Sybil attack is one when a malicious peer in the system creates a large amount of pseudonyms. Such a malicious peer uses fake identifies to gain larger influence over the fake alert ranking on others in the network. Newcomer attack is the process, when a malicious peer can easily register as a new user. Such a malicious peer creates a new ID for the purpose of erasing its bad history with other peer in the network and creates immediate damages. The simulated CIDN can be evaluated by result analysis and the route optimization can be analysed using the graph by the variation in routing level such as Trust Management without attack and Trust management with attack and for trustworthiness uses the trust aggregation and trust formation. It would generate mathematical model to derive "ground truth and can obtain objective trust against which subjective trust obtained as a result of executing the trust management. This would evaluate the route evaluation time and packet transmission time and network status also can be viewed. Thus the statistical performance of trustworthiness of each node can be analysed. System implementation Modularity is a desirable property of a system that partitions it into modules that are solvable and modifiable separately. Modularity helps in system debugging and maintenance. A modular system can be easily built by putting its modules together. The following are the modules used for implementing the system.

Algorithm 1 Acquaintance Selection (A*c, Lemin, Lemax*)
Require: A set of acquaintance candidates *Ac*
Ensure: A set of selected acquaintances *Aq* with minimum length *Lemin* and max length *Lemax* which brings the minimum overall cost
1: *Quit = false* //quit the loop if *Quit = true*
2: $Ac \Leftarrow \emptyset$
3: $U = min(\pi 0Cfp, \pi 1Cfn)$ //initialize the overall cost while there is no acquaintance. $min(\pi 0Cfp, \pi 1Cfn)$ is the cost when a node makes a decision without feedback from collaborators
4: **while** *Quit = false* **do**
5: //select the node that reduces cost most in each iteration
6: $Rmax = -MAXNUM$ //initialize the maximum cost reduction to the lowest possible
7: **for all** $e \in Aq$ **do**
8: $Aq = Aq \cup e$
9: **if** $U - R(Aq) - M(Aq) > Rmax$ //see Equation (9) and Equation (10) for $R(Aq)$ and $M(Aq)$ **then**
10: $Rmax = U - R(Aq) - M(Aq)$
11: *emax = e*
12: **end if**
13: $Aq = A\ q \setminus e$ //remove *e* from *Ac*
14: **end for**
15: **if** ($Rmax > 0$ and $|Aq| < Lemax$) or $|Aq| < Lemin$ **then**

16: *A q= Aq ∪emax*
17: *C = C \ emax* //remove *emax* from *C*
18: *U = U − Rmax*
19: **else**
20: *Quit = true*
21: **end if**
22: **end while**

Algorithm 2 Managing Acquaintance & Probation Lists
1: **Initialization** :
2: *Al ⇐Ø* //Acquaintance list.
3: *Pl ⇐Ø* //Probation list.
4: *lp = lini* //initial Probation length
5: //Fill *P* with randomly selected nodes
6: **while** |*Pl*| < *lp* **do**
7: *n* ⇐ select a random node
8: *Pl ⇐Pl ∪n*
9: **end while**
10: **set** new timer event(*tu*, "**SpUpdate**")
11: **Periodic Maintenance:**
12: **at timer event** ev of type "**SpUpdate**" **do**
13: //Merge the first mature node into the acquaintance list.
14: *n ⇐ selectOldestNode*(*Pl*)
15: *Ac ⇐Aq* //*Ac* is the temporary candidate list
16: **if** *te* > *tp* //*te* is the age of node *e* in the probation list **then**
17: *Pl ⇐Pl \ n*
18: **if** *Te* > *Tmin* and *Fe* < *Fmax* //*Te* and *Fe* are the true positive rate and false positive rate of the node *e* **then**
19: *Ac⇐ Ac ∪n*
20: **end if**
21: **end if**
22: //Consensus protocol
23: *As* =Acquaintance Selection(*C, lmin,* max(*lmin, q q*+1 *lmax*))
24: //Send requests for collaboration and receive responses
25: *Saccp ⇐ RequestandReceiveCollaboration*(*As, ttimeout*)
26: *Aq ⇐ Saccp* //Only nodes that accept the collaboration invitations are moved into the acquaintance list
27: //Refill *P* with randomly selected nodes
28: **while** |*Pl*|< *max*(*q*|*Aq*|, *lmin*) **do**
29: *e* ⇐ Select a random node not in *A*
30: *Pl ⇐Pl ∪e*
31: **end while**
32: **set** new timer event(*tu*, "**SpUpdate**")
33: **end timer event**

3.MODULES DESCRIPTION:

3.1 DYNAMIC NODES

NODE INSERTION

The new node becomes then root for its node ID. The root finds the length of the longest prefix of the ID it shares. Then it sends a multicast message that reaches all existing nodes sharing the same prefix. These nodes then add the new node to their routing tables. The new node may take over being the root for some of the roots objects. The node will contact the new node to provide a temporary neighbourhood list. The new node then performs an iterative nearest neighbour search to l in the fill all levels in its routing table.

NODE DEPARTURE

To leave the network, a node broadcasts its intention of leaving and transmits the replacement node for each level in the routing tables of the other nodes. Objects at the leaving node are redistributed or replenished from redundant copies.

NODE FAILURE

Unexpected node failure is handled through redundancy in the network and backup pointers to re-establish damaged link.

3.2WIRELESS LINK

A Wireless local area network(WLAN) links two or more devices over a short distance using a wireless distribution method,usually providing a connection through an access point for Internet access.The use of spread spectrum or OFDM technologies may allow users to move around within a local coverage area, and still remain connected to the network.

3.3CLUSTER AREA

Cluster consists of a set of loosely connected computers that work together so that in many respects they can be viewed as a single system. The components of a cluster are usually connected to each other through fast local area networks, each node running its own instance of an operating system. Computer clusters emerged as a result of convergence of a number of computing trends including the availability of low cost microprocessors, high speed networks, and software for high performance distributed computing. Clusters are usually deployed to improve performance and availability over that of a single computer, while typically being much more cost effective than single computers of comparable speed or availability .Computer cluster have a wide range of applicability and deployment, ranging from small business clusters with a handful of nodes to some of the fastest supercomputers

3.4RESOURCE MANAGEMENT

In resource management is the efficient and effective deployment of organizations resources when they are needed. One resource management technique is resource levelling. It aims at smoothing the stock of resources on hand, reducing both excess inventories and shortages. The required data are the demands for various resources, forecast by time period into the future as far as is reasonable, as well as the resources configurations required in those demeans ,and the supply of the resources, again forecast by time period into the future as far as is reasonable. The goal is to achieve 100% utilization but that is very unlikely, when weighted by important metrics and subject to constraints, for examples, meeting a minimum service level, but otherwise minimizing cost. In this design the node is allocated as resource management one who has higher transaction rate of transmission[5].

3.5TRUST MANAGEMENT

SATISFACTION MAPPING

In this model, a HIDS sends requests to its peer and evaluates the satisfaction level of received feedback. Note

that the request can be a test message or a real request. The true answer of a test message is known beforehand while that of a real request is verified by administrators after some delay through the observed impact of the corresponding alert

ATTACKS

Betrayal attack: It is when a trusted peer suddenly turns in to malicious one and set false feedbacks. A trust management system can be degraded dramatically because of this type of attacks. Sybil attack is one when a malicious peer in the system creates a large amount of pseudonyms. Such a malicious peer uses fake identifies to gain larger influence over the false alert ranking on others in the network. Newcomer attack is the process, when a malicious peer can easily register as a new user. Such a malicious peer creates a new ID for the purpose of erasing its bad history with other peer in the network and creates immediate damages.

4.SIMULATION AND RESULTS:

NS-2 is an event driven packet level network simulator developed as a part of the VINT task (Virtual Internet Test bed).Version 1 of NS was established in 1995 and with version 2 in 1996 Ns-2 with C++/OTCL integration article. Version 2 comprised a scripting language called Object oriented Tcl (OTCl). It is an open source software platform available for both Windows 32 and Linux platforms.

Software tools used with ns-2:

In the simulation, there are the two implements are used NAM(Network Animator) and x-Graph.

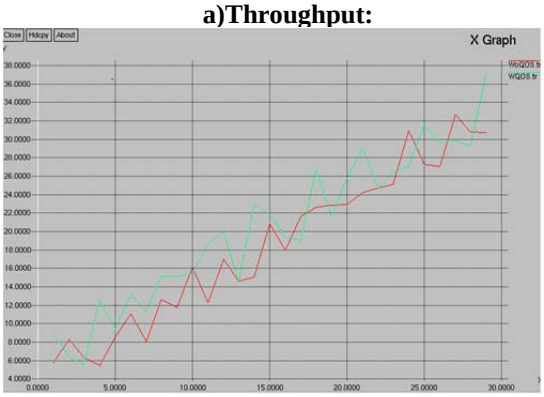
NAM (Network Animator):

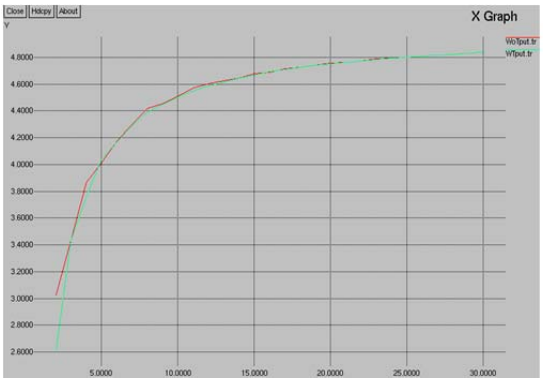
NAM provides a visual interpretation of the link topology created. The application was developed as measure of the VINT scheme.

X Graph:

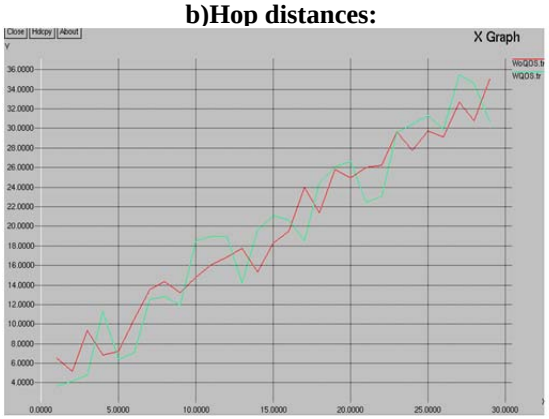
X- Graph is a X-Window application that includes:
Interactive plotting and graphing Animated and derivatives to use Graph in NS-2 the executable can be called within a TCL characters. Before capacity a graph displaying the information visually displaying the information of the file produced from the simulation. The output is a graph of size 800 x 500 displaying information on the traffic flow and time

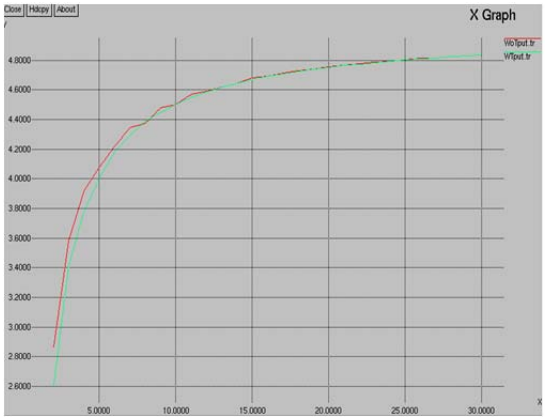
a)Throughput:



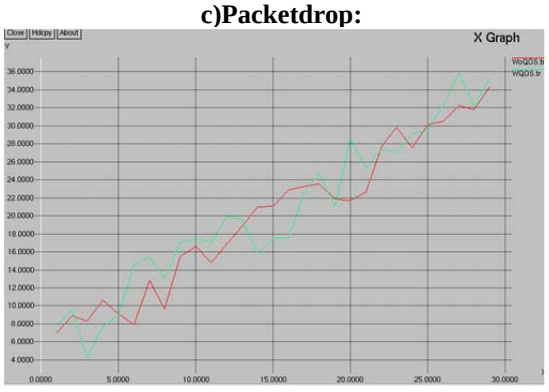


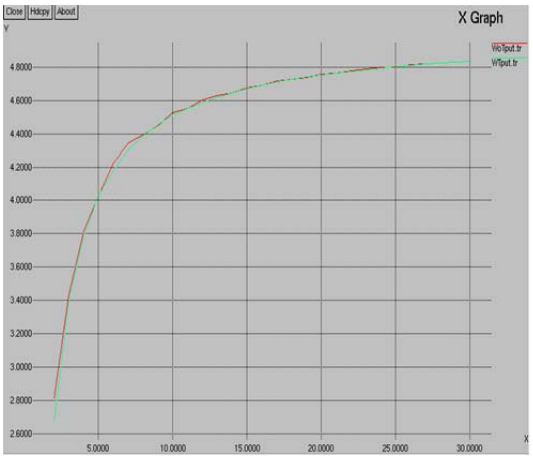
b)Hop distances:





c)Packetdrop:





5. CONCLUSION

A trust management is the solution for evaluating trustworthiness of Host based Intrusion Detection System in Collaborative Intrusion Detection Networks. In this project, Trust Management adapts acquaintance management with probation list and ground truth value to measure the uncertainty in estimating the likely future behaviour of HIDSes. The measured uncertainty allows trust management to employ an adaptive message exchange, resulting in good scalability.

Intrusion can be detected by analysing various attacks and trusted route can be optimized using geography routing and also demonstrated to have the properties of fairness and convergence and to provide incentive for collaboration.

FUTURE ENHANCEMENTS:

Other possible direction for future work is to improve the efficiency of trust management in various methods using image for more secure.KDD dataset to trust the host and other management, to detect the trust management, all host can generator key to make sure the performance of trust management. In this case the importance issue of scalability and collusion attacks should be addressed and can also to deploy a real CIDN using existing intrusion detection system.

REFERENCES

1. D.Moore, C.Shannon, and k claffy,"Code-red: a case study on the spread and victims of an Internet worm," in proc.2nd ACM SIGCOMM workshop Internet Measurement,2002.
2. "ZDnet."Available:http://www.zdnet.com/blog/security/a lo t of overhead which limits their scalability. Another confickers-estimated-economic-cost-91-billion/3207 [Last accessed Oct. 15, 2010].important concern is that IDSes can be easily compromised
3. R. Vogt, J. Aycock, and M. Jacobson, "Army of botnets," in Netw.andecome deceptive when reporting the trustworthiness of Distributed Syst. Security Symp
4. D.Dagon ,X.Qin, G.Gu, W.Lee, J.Levine, and H.Owen , "Honeystat : local worm detection using honeypots"Recent Advances in Intrusion Detection . Springer,2004.
5. Q.zhu,C.Fung, R.Boutaba, and T.Basar ,"A game-theoritical approach to incentive design in collabrative intrusion detection networks,"in proc. International Symp Game theory netw.May2009
6. P.Resnick, K.Kuwabara, R.Zeckhauser and E.Friedman "Reputation systems", Commun.ACM,vol.43, no.12,pp. 45-48,2000
7 R. Vogt, J. Aycock, and M. Jacobson, "Army of botnets," in Netw.and ecome deceptive when reporting the trustworthiness of Distributed Syst. Security Symp
8. D.Dagon ,X.Qin, G.Gu, W.Lee, J.Levine, and H.Owen , "Honeystat : local worm detection using honeypots"Recent Advances in Intrusion Detection . Springer,2004.
9 .P.Resnick, K.Kuwabara, R.Zeckhauser and E.Friedman "Reputation systems", Commun.ACM,vol.43, no.12,pp. 45-48,2000
10. S.Russell and P.Norvig, Artificial Intelligence: A Morden Approach,2nd edition. Prentice Hall, [2002]
11. C.Fung, O.Baysal, J.Zhang, I.Aib, and R.Boutaba,"Trust management for host based collaborative intrusion detection" , in 19th IFIP/IEEE International Workshop Distributed Syst.,2008.