ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Detecting Threads in Secure Communication using RCEE

T.Mathupriya

PG Scholar,Computer Science and Engineering, V.S.B. Engineering College,Karur, TamilNadu, India[1]

ABSTRACT—Security and privacy issues are important with the multi-agent systems such as pervasive computing, grid computing and P2P networks which are open, anonymous and dynamic in nature .These characteristics introduces vulnerabilities and threats to providing secured communication. To minimize these threats and to manage altering behavior of malicious agent, it evaluates the trust and reputation of the interacting agents; a dynamic trust computation model called 'Secured Trust'. It uses different factors to evaluate the trust of an agent and proposes a novel load balancing algorithm and effectively manage with strategic behavioral change of malicious agents and distribute workload among the service providing agents under stable condition by finding the satisfaction level .These are implemented in client server architecture to self defense the application and to detect the threat from the server application itself and runtime code execution behavior monitoring is implemented. At runtime, a monitor compares the behavior of the variants at certain synchronization points and raises an alarm when a difference is detected.

KEYWORDS—Multi-agent system, security, trust management, reputation model, load balancing, malicious, runtime code execution, self defense.

I. INTRODUCTION

In a multi-agent system, agents interact with each other to achieve a definite goal that they cannot achieve alone [1] and such systems include P2P [2][3][4][5], grid computing [6], the semantic web [7], pervasive computing [8] and MANETs. Multi-agent Systems (MASs) are increasingly becoming popular in carrying valuable and secured data over the network. Nevertheless, the open and dynamic nature of MAS has made it a challenge for researchers to operate MAS in a secured environment for information transaction. Malicious agents are always seeking ways of exploiting any existing weakness in the network. This is where trust and reputation play a critical role in ensuring effective interactions among the participating agents [9][10]. Researchers have long been utilizing trust theory from social network to construct trust models for effectively suppressing malicious behaviors of participating agents. Trust issues have become more and more popular since traditional network security approaches such as the use of fire-wall, access control and authorized certification cannot predict agent behavior from a 'trust' viewpoint.

A reputation based trust model [11] [12] [13] [14] collects, distributes, and aggregates feedback about participants' past behavior. Trustworthy behavior, and discourage participation by agents who are dishonest. Reputation based trust models are basically divided into two category based on the way information is aggregated from an evaluator's perspective [15][16]. They are "Direct/Local experience model" and" Indirect/Global reputation model" where direct experience is derived from direct encounters or observations and indirect reputation is derived from inferences based on information gathered indirectly (second- hand evidence such as by word-of-mouth). So, in case of global reputation models [17] an agent aggregates feedback from all the agents who have ever interacted with the target agent i.e., an agent has a view of the network which is wider than its own experience, thus enabling it to quickly converge to a better decision. However, global reputation models are much more complex to manage than local experience models as malicious agents have the opportunity to provide false feedbacks.

Most of the existing global reputation models can successfully isolate malicious agents when the agents behave in a predictable way. However, these models suffer greatly when agents start to show dynamic personality i.e., when they start to behave in a way that benefits them. These models also fail to adapt to the abrupt change in agents' behavior and as a result suffer when agents alter their activities strategically. Moreover, some of the models show dishonest or unfair rating and collusion. Another aspect which is slowly becoming critical for the proper maintenance a little effect

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

in dealing with more complex attacks such as nonce of service quality is the appropriate distribution of workload among the trusted service providers. Without a proper load balancing scheme the load at highly reputable service providers will be immense which will eventually cause a bottleneck in the system's service quality. To the best of our knowledge none of the existing trust models consider load balancing among service providers.

With these research problems in mind, RCEE has been proposed in which can effectively detect sudden strategic alteration in malicious behavior with the additional feature of balancing workload among service providers. This model considers variety of factors in determining the trust of an agent such as satisfaction, similarity, feedback credibility, and recent trust, and historical trust, sudden deviation of trust and decay of trust and implemented in client server architecture. We have used a novel policy of utilizing exponential averaging function to reduce storage overhead in computing the trust of agents. We have also proposed a new load balancing algorithm based on approximate computation of workload present at different service providers.

II. RELATED WORK

The main objective of this paper is to provide a dynamic trust computation model for effectively evaluating the trust of agents even in the presence of highly oscillating malicious behavior. Our model also provides an effective load balancing scheme for proper distribution of workload among the service providing agents. A number of parameters have been considered in our trust model for computing the trust of an agent. Now, some of these parameters have been previously but none of these models can fully cope with the strategic adaptations made by malicious agents. The mathematical and logical definitions used for these parameters also cannot reflect the true scenarios faced in real life. In the following sections we redefine the mathematical expressions used for some of the parameters and at the same time define some new ones and then finally combine all of the parameters to present our new dynamic trust model.

A. Satisfaction

Satisfaction function measures the degree of satisfaction an agent has about a given service provider. In other words, it keeps record of the satisfaction level of all the transactions an agent makes with another agent. However, instead of storing all of the transaction history we have defined an exponential averaging update function to store the value of satisfaction. This greatly reduces the storage overhead and at the same time assigns time relative weight to the transactions.

Let, $Sat^t(p, q)$ represent the amount of satisfaction agent p has upon agent q based on its service up to n transactions in the t-th time interval .The satisfaction update function is defined as follows-

$$Sat^t(p, q) = \alpha \times Satcur + (1 - \alpha) \times Sat^t n-1 (p, q) \quad (1)$$

Here, Sat_{cur} represents the satisfaction value for the most recent transaction has used a feedback based system where an agent rates other agent's service quality according to the following function.

$$Sat_{cur} = \begin{cases} 0, & \text{if transaction is fully unsatisfactory} \\ 1, & \text{if transaction is fully satisfactory} \\ \in (0,1), & \text{otherwise} \end{cases} \quad (2)$$

The weight α change based on the accumulated deviation $\xi^t_n(p,q)$.

$$\alpha = threshold + c \times \frac{\delta^t_n(p,q)}{1 + \xi^t_n(p,q)} \quad (3)$$

$$\delta^t_n(p,q) = |Sat^t_{n-1}(p,q) - Sat_{cur}| \quad (4)$$

$$\xi^t_n(p,q) = c \times \delta^t_n(p,q) + (1-c) \times \xi^t_{n-1}(p,q) \quad (5)$$

$\xi^t{}_0(p,q)=\xi^{t-1}{}_{last}(p,q)$ and $\xi^0{}_0(p,q)=0$.Here c is some user defined constant factor which controls to what extent we

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

will react to the recent error (δ^t (p, q)). So, if we increase the value of c then we give more significance to the recent deviation than the accumulated deviation and vice versa. Again we can see that as recent error (δ^t (p, q)) increases so does α which means that recent satisfaction is given higher weight than accumulated satisfaction (i.e., higher significance is given to the recent service feedback). The threshold represents a threshold which is used to prevent α from saturating to a fixed value. Initial value of α is set to 1 (we consider only the current transaction at the very beginning as we have no knowledge of any previous transaction) and threshold is set to 0.25.

B. Similarity

Similarity metric defines to what extent two agents are alike. We have computed similarity by determining the personalized difference in satisfaction rating over the common set of interacted agents and have then used the computed difference rating to define the degree of similarity. I S(p) represent the set of agents with whom agent p has made interaction Credibility is a direct logarithmic function of similarity for its slow rise to the highest attainable value. This implies that agents with higher similarity with respect to the evaluating agent have higher feedback credibility. Using this trust can be evaluated between the agents.

III. LOAD BALANCING AMONG AGENTS

In this section we propose an algorithm for balancing loads among the trusted agents. For selective scenario, we first compute the trust of agents who respond to a transaction request and then we select the agent with the highest trust value. However, in this scenario the agent with the highest trust value will have immense workload while other capable agents with slightly lower reputation will have considerably less workload. The problem that will arise from this disproportionate allocation of workload is that the quality of service will fall greatly due to the heavy workload So a load balancing algorithm is required for the sustainability of good service quality.

In our load balancing algorithm, calculate either a heuristic value of workload or choose the agent with the smallest load.

Pseudo-code of the load balancing algorithm is given in Algorithm:

Input: Evaluating agent p and the set of agents
responding to a service request S
Output: Service providing agent q
for each $x \in G$ do
if $Trust(p, x) > \gamma$ then
$G \leftarrow G \cup \{x\}$
else
$U \leftarrow U \cup \{x\}$
end if
end for
if $G = \emptyset$ then
for each $x \in G$ do
compute load N (p, x)
end for
sort G in increasing order of load N
return agent q with the smallest load N
else
$Total\_trust \leftarrow 0$
for each $x \in U$ do
$Total\_trust \leftarrow Total\_trust + Trust(p, x)$
end for
if $Total\_trust > 0$ then
for each $x \in U$ do compute Prob(p, x)
end for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

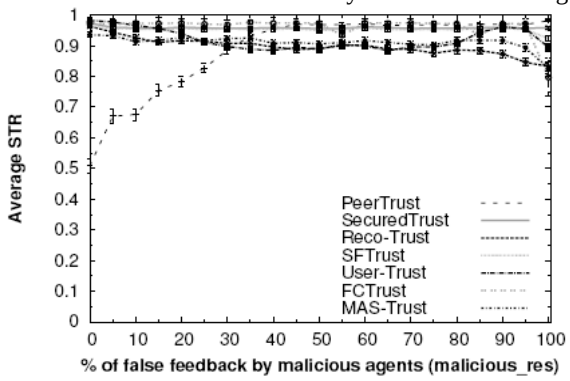
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

return agent q with probability P rob(p, q)
else
return any agent q randomly
end if
end if

We then first seek to choose an agent from G by computing an approximate value (heuristic value) of load present at each responder in G. Sorting the responders in increasing order of load we take the responder with the smallest workload. In case of no responders being present in the class G we select an agent from U either probabilistically based on its trust value or randomly.

IV. COMPARISON WITH OTHER TRUST MODELS

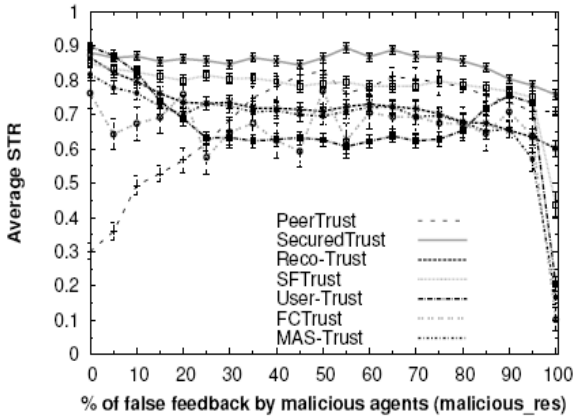
In this set of experiments we will demonstrate the efficiency of Secured Trust against other existing trust models



(a)

Fig. 1. Comparing SecuredTrust with other trust models in terms of average STR (with 95% confidence interval) against malicious res (a) 40% malicious agents (b) 60% malicious agents.

In these experiments an agent first computes and compares the trust values of the responding agents and chooses the agent with the highest trust value for interaction. A transaction is successful if the participating agent is cooperative i.e., if it is a good agent and this model must be implemented with balancing weights.



(b)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

V. CLIENT SERVER ARCHITECTURE

This model does not kernel privileges and it is implemented to provide security with the client server architecture and to detect the threat from the server application itself .To implement these concepts the following methods are implemented in the client server architecture. Self Defense Concept

The server application is activated and task can be assigned and the client node's ID, IP address and system name are keyed in and stored in the database table [18]. The server self defense activity is made such that the files in the server shared folder is not affected by the client nodes. If any of the file is about to changing, then the new file is again replaced by the old file which is located in non-shared folder path. The node is selected, an executable file is copied to the node's share folder (which is located in root folder of the client application) and a number of parameters are saved in database table. When the client application executes the task, this executable file is invoked and uses the parameter values from the database table. The node details are viewed using the data grid view control. The details include node id, IP Address, and system name. The attack details executed are viewed using the data grid view control [19]. The details include AttackTime, AttackType, FileName, OldFileName details.

B. Client Behavior and Activities

The task details are collected from 'Tasks' table and are saved in System. Collections. Generics. List class's object. Then the executable files are executed by using System. Diagnostics. Process class. the task details are collected from 'Tasks' table and are saved in System.Collections.Generics.List class's object. When the executable files are executed, instead of working as mentioned, the code works differently with additional values generated randomly. It seems that instead of server version code, the code is different and so the result produced should be suspicious.

The server's Self Defense Resources shared folder is accessed and ten files are created with random names. The details are shown in log list box. Likewise, the folder's files are collected and any one file is renamed. In addition, any one file is deleted. Note that, the self defense module if activated in server, the renamed file is deleted and file is restored from original backup path. Likewise, if a file is being deleted, the file restored from original backup path. folder content if required.

VI.CONCLUSION

RCEE approach is that it enables us to detect and prevent a wide range of threats, including "zero-day" attacks. Multivariate execution is effective even against sophisticated polymorphic and metamorphic viruses and worms. Many everyday applications are mostly sequential in nature.

At the same time, automatic parallelization techniques are not yet effective enough on such workloads. The statistical analysis of code injection attacks data if prepared can be used for further project development number of software can be found out easily where the injections are found out. Once code affected part are send the mail to particular client that intruders came to affect the software. The multimedia files attacks can also be detected.Multi tasking can also be performed and time taken to complete the task is minimized. The efficiency is improved by improving the coding efficiency.

REFERENCES

[1] N. R. Jennings, "An agent- based approach for building complex software systems," Communications of t he ACM ,vol. 44, no. 4, pp.

[2] R. Steinmetz and K. Wehrle, Peer-to-Peer Systems and Applications. Springer-Verlag New York, Inc., 2005.

[3] I. Foster, C. Kesselman, and S. Tuecke, " The anatomy of the grid: enabling scalable virtual organizations," International Journal of High Performance Computing Applications, vol.15, no. 3, pp. 00– 222, 2001.

[4] T. Berners-Lee, J.Hendler, and O.Lassila,"The semantic Web, "Scientific American, pp. 35–43, May 2001.

[5] D. Saha and A. Mukherjee, "Pervasive computing: A paradigm for the 21st century," Computer, vol. 36, no. 3, pp. 25–31, 2003.

[6] S. D. Ramchurn, D. Huynh, and N. R. Jennings, "Trust in multi- agent systems," The Knowledge Engineering Review,vol. 19, no. 1, pp. 1–25, 2004.

[7] P. Dasgupta, "Trust as a commodity," Trust: Making andBreaking Cooperative Relations, pp. 49–72, 2000.

[8] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," Communications of the ACM, 12, pp.45–48, 2000.

[9] A. A. Selcuk, E.Uzun, and M. R.Pariente, "A reputation- based trust management system for P2P networks," in Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid(CCGRID), 2004, pp.

[10] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video (NOSSDAV). ACM, 2003, pp. 144–152.

[11] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in Proceedings of the tenth international conference on

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Information and knowledge management (CIKM). ACM, 2001, pp. 310–317.

[12] L. Mui, M. Mohtashemi, and A. Halberstadt, "A computational model of trust and reputation for e-businesses," in Proceedings of the35th Annual Hawaii International Conference on System Sciences(HICSS'02), 2002, pp. 2431 – 2439.

[13] L. Mui, "Computational models of trust and reputation: agents, evolutionary games, and social networks," Ph.D. Thesis, Massachusetts Institute of Technology(MIT), 2002

[14] F. Cornelli, E. Damiani, S. D. Capitani, S. Paraboschi, and P. Samarati, "Choosing reputable servents in a P2P network," in Proceedings of the 11th ACM World Wide Web Conference (WWW), May 2002, pp. 376–386.

[15] R. I. Benjamin, J. F. Rockart, M. S. S. Morton, and J. Wyman, "Information technology: A strategic opportunity", Sloan Manage. Rev., vol. 25, no. 3, pp.3 -10 1984.

[16] Mantri,T.,Ayu,M.A.;Borovac,A.;Zay,A.Z.Z.Multimedia Computing and Systems(ICMCS),2012 International Conference on,2012.