AN AUTOMATED TRUST MODEL FOR CLIENT-SERVER AND PEER-TO-PEER SYSTEM

[1]LANKE PRAVIN, [2]SACHIN MALVE

[1]Student, IOKCOE, [2]Professor, IOKCOE
Email: [1]lankepd@gmail.com

Abstract—To maintain trust in peer-to-peer system is to difficult as there is no central server in between peers. To create trust in between peers can avoid attacks from malicious peers. This paper presents an automated trust model which takes into consideration peers past interaction and recommendations to choose trustworthy peer. While evaluating trustworthiness parameters like importance, recentness and peer satisfaction is taken into consideration. Recommenders trustworthiness and confidence about a recommendation are also considered while evaluating recommendations. We have implemented access control technology in the P2P file sharing system and for that we have used symmentric encryption with shared secrete key. Proposed automated trust model can matigate attacks on different malicious behaviour models. Our experiments help to detect malicious peers and form a group of good peers.

Keywords—An Automated Trust Model, Reputation, Security and Protection, Peer-to-peer system.

I. INTRODUCTION

ON e-commerce website like Olx, Quikr and Flipkart before deciding to buy any product visitors usually look for customer reviews. In the above example centralized mechanism is used for storing and manipulating reputation data. In our paper we have explored posibilities for trust management in completely decentralized environment, where no central database is used i.e. peer-to-peer system.

Reputation must be associated with self-maintaned trust model rather than global trust model hence an automated trust model is used at each peer. To form trust relationship in peers proximity can provide more security and also provide reduced risk and uncertainty in future P2P interactions. In computational model metrics are used to represent trust. Peers are classified as trustworthy or untrustworthy and also ranked according to their trustworthiness. Trust among peers can be measured using interactions and feedbacks of peers. Interation gives certain information about the peer but feedback might information,. Reputation must be associated Reputation must be associated with self-maintaned trust model rather than .

Reputation must be associated with self-maintaned trust model rather than global trust model hence an automated trust model is used at each peer. To form trust relationship in peers proximity can provide more security and also provide reduced risk and uncertainty in future P2P interactions. In computational model metrics are used to represent trust. Peers are classified as trustworthy or untrustworthy and also ranked according to their trustworthiness. Trust among peers can be measured using interactions and feedbacks of peers. Interation gives certain information about the peer but feedback might give deceptive.

We propose an automated trust model that aims to improve security in P2P system by establishing trust relations among peers in their proximity. Each peer develops its own view of trust about the peers with whoom he interacted in the past. In this way good peers form dynamic trust groups and can isolate malicious peers. At the beginning, peers are assumed to be strangers to each other and becomes an acquitance of another after providing a service, e.g. download a file. If peer has no interaction in the past, it chooses to trust strangers.

Using a service of a peer is an interaction, which is evaluated based on weight (importance), recentness of the interaction and satisfaction of the requester. An acquaintances feedback about a peer, recommendation is evaluated based on recommenders trustworthiness. It contains the recommenders own experience about the peer, information collected from the recommenders acquaintances, and the recommenders level of confidence in the recommendation. If the level of confidence is low, the recommendation has a low value in evaluation and affects less the trustworthiness of the recommender. An automated trust model defines two primary metrics to calculate trustworthiness among peers: service trust metric and recommendation trust metric. Service trust metric is used for selecting service provider and recommendation trust metric is used for requesting recommendations from other peers. Reputation metric is a secondary metric calculated using recommendation trust metric. In experiment we have studied 4 types of malicious peer behaviors, which perform both service and recommendation-based attacks. An automated trust model mitigates service-based and recommendation-based attacks.

Our experiment shows that good peers defend themselves against malicious peers and assess trustworthiness of other peers based on the information available with it.

II. LITERATURE SURVEY

Guha et al. show that expressing trust or distrust per peer allows hito predict between any two people in

the network with high accuracy. Result of their experiment shows that distrust is helpful to measure trustworthiness accurately. J. Douceur explained 'the sysbil attack' to reputation system are vulnerable to sybil attack, where malicious peers gives bogus feedbacks by creating multiple fake entities. To overcome Sybil attack, Yu et al. as well as Tran et al. propose system which is based on the observation that fack entities and have many trust relationships among each other but they rarely have relationships with real users.

Decentralized network have more challenges comparing to centralized platform. Due to lack of central authority malicious peers have more attack opportunities in P2P sytem. Attacks like self promoting, white-washing, slandering, orchestrated and denial of service attacks in P2P trust model are discussed by Hoffman et al.
In network peer is assumed as trustworthy unless there are complaints against it. In Aberer and Despotovic's trust model, peer reports their complaints using P-Grid. Eigentrust uses transitivity of trust to calculate global trust values stored on content addressable network i.e. CAN. L. Xiong and L. Liu's peer trust defines transaction and community context parameters to make trust calculations adaptive on PGrid.
Both Eigentrust and Peertrust evaluate a recommendation based on trustworthiness of the recommender.

Can and Bhargava defines a self-ORganizing trust model for P2P system. Instead of considering a particular trust holders feedback as authentic, public opinion from all acquaintances is considered as a more credible information. Instead of considering global trust information, local trust information is used to take decisions as peers develop their own trust networks. For efficient aggregation of trust values gossip trust defines a randamized gossiping protocol. This experiment shows that gossiping reduces reputation query traffic. We send reputation queries only to those peers with whom we have interacted in the past, which reduces network traffic.

III. IMPLEMENTATION DETAILS

A. Mathematical Model
Model developed in this paper is built on such an environment where reciprocity norms are expected. In our experiment suppose peer pj is evaluating pi's reputation for being cooperative. We define embedded social network of pj as the set of all the peers that pj asks for this evaluation. So by the way, the reputaion of a peer pi is relative to the particular embedded social network in which pi is being evaluated. For the simplicity, we are not adding any new peer to the system. We reinforcing relationships among the three concepts they are reciprocity, trust and reputation. For an peer pj with a embedded social network A: increase pj's reputation in A should also increase the trust from other peers for pj and the increase in pi's trust of pj should also increase the likehood that pi will reciprocate positively to pj 's action; since pj 's reciprocation action to others in A increased, its reputation in

A should also be increased.
Reciprocity is defined as mutual exchange of deeds. Two types of reciprocity are considered in this model: direct reciprocity refers to interchange between two concerned peers while indirect reciprocity refers to interchange between two concerned peers interceded by mediating peers in between. The model defines reputation as perception that an peer creates through past actions about its intentions and norms. Mathematically, let _ji(c) represent pi's reputation in an embedded social network of concern to pj for a context c. This value is subjective to every other peer since the embedded
social network difference when p_i connects to different p_j. In this way $\theta_{ji}(c)$ measures the likelihood that p_i reciprocates p_j's actions.

In this model, trust is defined as a subjective expectation an peer has about anothers future behavior based on the history of their encounters. Thus to evaluate the trustworthiness of p_i, let D_{ji}(c) represents history of encounters that p_j has with p_i within the context c. Moreover, we should take note that trust is a subjective quantity calculated based on the two peers concerned in a dyadic encounter. So we can model trust using $T(c) = E[\theta_{ji}(c)|D_{ji}(c)]$. The higher the trust level for peer p_i, the higher the expectation that p_i will reciprocate peer p_j's action.

We describe the computational model in detail with the following scenario:

We assume the notations used for this scenario are:
θ_{ab}: b's reputation in the eyes of a
X_{ab}(i): The ith encounter between a and b

$$X_{ab}(i) = \begin{cases} 1 & \text{if } b\text{'s action is complete} \\ 0 & \text{otherwise} \end{cases}$$

D_{ab}: history; The set of n previous encounters between a and b.
$D_{ab} = \{Xab(1).X_{ab}(2).....X_{ab}(n)\}$
Let p be the cooperative actions by agent b towards a in the n previous encounters, b's reputation θ_{ab} could be modeled by a simple proportion function of p cooperative actions over n encounters. In statistics, a proportion random variable can be modeled as a Beta distribution:$p(\hat{\theta})$ =Beta$(c1, c2)$ where $\hat{\theta}$ represents an estimator for θ.

If peers a and b are complete strangers, when they first meet, their estimate for each other's reputation is assumed to be uniformly distributed across the reputation's domain:

$$p_\theta(i) = \begin{cases} 1 & 0 \leq \hat{\theta} \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

In this model, the beta distribution will be uniform when $c_1 = c_2 = 1$.
Now we have a simple estimator for θ_{ab} which is the proportion of cooperation in n finite encounters:$\theta_{ab} = p/n$. Assuming that each encounter's cooperation probability is independent of other encounters between a and b, the likelihood of p cooperations and $(n - p)$ defections can be modeled as$L(D_{ab}|\hat{\theta}) = \theta^p(1-\theta)^{1-p}$. Combining the prior and the likelihood, the posterior estimate for $\hat{\theta}$ becomes: $p(\hat{\theta}|D)$ =Beta$(c_1 + p, c_2$+n−p). As we mentioned previously, trust toward b from a is the conditional expectation of reputation $\hat{\theta}$ so it can be computed by
$T_{ab} = p(X_{ab}(n+1) = 1|D) = E[\hat{\theta}|D] = \frac{c_1+p}{c_1+c_2+p}$

B. Proposed system

The proposed P2P file sharing system is a windows based program that allows you to host secure P2P file sharing. Users just need to install client software on each peer side. Key features of our model are listed below:

Represent trust in computational model Studies service and recommendation-based attacks Rank peers according to their trustworthiness Create trust network by only using local information Shared secrete key for symmetric encryption

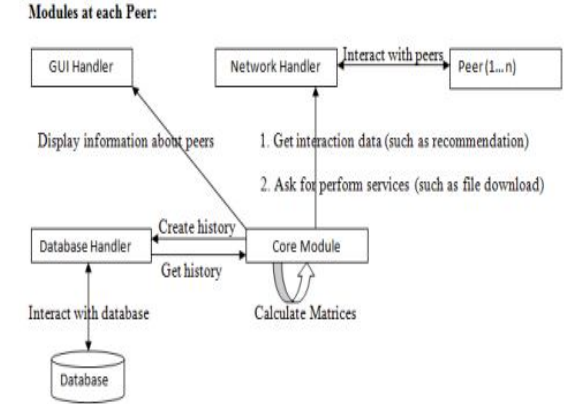
Access control facility to every peer

1) System Architecture:

In P2P network system architecture consist of four compo-
nents

Description:

Fig. 1. System Architecture



1) Network Handler

Search for peers.

Interact with peers (getting or providing services to peer).

Get recommendations from peers.

2) Core Module

Calculate matrices:

a) Service Trust Metric

b) Reputation Metric

c) Recommendation Trust Metric

Create interaction History.

3) Database Handler

interact with database:

Store interaction history.

Fetch interaction history.

Update interaction history.

4) GUI Handler

Display information about peers (such as IP ad- dress, trustworthiness etc).

Display information about recommenders.

Display information about past interaction with respective peers.

C. Computational Model:

In our experiment we have assumed that all the peers are of same computational power and responsibility. We dont have any centralized or trusted peer to manage trust relationship.

Notations :

Notations on the trust metrices are shown below:

Notation	Description
p_i	i^{th} peer
A_i	p_i's set of acquintances
SH_{ij}	p_i's service history with p_j
sh_{ij}	Current size of history
sh_{max}	upper bound for service history size
s_{ij}^k	p_i's satisfaction about k^{th} interaction with p_j
w_{ij}^k	weight of p_i's k^{th} interaction with p_j
f_{ij}^k	fading effect of p_i's k^{th} interaction with p_j
r_{ij}	p_i's reputation value about p_j
st_{ij}	p_i's service trust value about p_j
rt_{ij}	p_i's recommendation trust about p_k

Table 1: Notations used

To improve importance of new interactions fading effect parameter is calculated which forces peers to stay consistant in future interactions. It is calculated as follows:

$$f_{ij}^k = \frac{k}{sh_{ij}},\ 1 \le k \le sh_{ij}$$

Before starting downloading or uploading peers in a net- work make bandwidth agreement. The ratio of average band- width(AveBw) and agreed bandwidth (AgrBw) is measure of reliability of a peer in terms of bandwidth. Ratio of online and offline period of peer represent availability of an peer. The satisfaction parameter is calculated based on above variables:

$$S_{ij}^k = \begin{cases} \left(\frac{AveBw}{AgeBw} + \frac{OnP}{OnP+OffP}\right)/2 & \text{if AveBw} < \text{AgrBw} \\ \left(1 + \frac{OnP}{OnP+OffP}\right)/2 & \text{otherwise} \end{cases}$$

Lets assume that $Uploader_{max}$ be the number of uploaders of the most popular file. *size* and $\#Uploaders$ denote the file size and the number of uploaders, respectively. p_i calculates the weight parameter of *kth* interaction with p_j as follows:

$$w_{ij}^k = \begin{cases} \left(\frac{size}{100MB} + \frac{\#Uploaders}{Uploaders_{max}}\right)/2 & \text{if size} < 100\ \text{MB} \\ \left(1 + \frac{\#Uploaders}{Uploaders_{max}}\right)/2 & \text{otherwise} \end{cases}$$

1) Service Trust Metric (st_{ij}):

Competence belief (cb_{ij}) and *integrity belief* (ib_{ij}) parameters are used to calculate Service trust metric.*Competence belief* represent how well an acquaintance satisfied the needs of past interaction. p_i calculates cb_{ij} as follows:

$$cb_{ij} = \frac{1}{\beta_{cb}} \sum_{k=1}^{sh_{ij}} (s_{ij}^k . w_{ij}^k . f_{ij}^k)$$

β_{cb} is a normalization effect and it is given by $\beta_{cb} = \sum_{k=1}^{sh_{ij}} (w_{ij}^k . f_{ij}^k)$.*Competence belief* ($cb_{ij}$) always take value between 0 and 1.

Consistency is as important as competence. Level of confidence in predictability of future interactions is called

integrity belief.

$$ib_{ij} = \sqrt{\frac{1}{sh_{ij}} \sum_{k=1}^{sh_{ij}} (s_{ij}^k . w_{ij}^\mu . f_{ij}^\mu - cb_{ij})^2}$$

w_{ij}^μ and f_{ij}^μ are the mean of w_{ij}^k and f_{ij}^k values in SH_{ij}, respectively. We can approximate f_{ij}^μ as follows:

$$f_{ij}^\mu = \frac{1}{sh_{ij}} \sum_{sh_{ij}}^{k=1} f_{ij}^k = \frac{sh_{ij}+1}{2sh_{ij}} \approx \frac{1}{2}$$

p_i may calculate st_{ij} as follows:

$$st_{ij} = cb_{ij} - ib_{ij}/2$$

We have not considered p_j's reputation in above equation so the equation is not complete. In the early phases of trust relationship reputation is very important. When there is no any interation with acquaintance, a peer needs to depend on reputation metric only. There for p_i calculates st_ij as follows:

$$st_{ij} = \frac{sh_{ij}}{sh_{max}}(cb_{ij} - ib_{ij}/2) + (1 - \frac{sh_{ij}}{sh_{max}}) r_{ij}$$

When p_j is stranger to p_i value taken for $sh_{ij} = 0$ and $st_{ij} = r_{ij}$

2) Reputation Metric (r_{ij}):

Reputation metric is used to calculate stranger's trustworthiness based on recommendations. In our experiment we have assumed that p_j is stranger to p_i and p_k is acquaintance of p_i. To calculate r_{ij}, p_i sends reputation query to its acquaintances. Below algorithm shows how peer p_i chooses trustworthy acquaintances and request their recommendations.η_{max} represent the maximum number of recommendation collected through reputation query and $|S|$ denotes the size of set S.p_i sets high threshold value for recommendation trust values and request recommendation from highly trusted acquaintances first. It repeats the same operation untill η_{max} reach or theshold drops under $(\mu_{rt} - \sigma_{rt})$

Algorithm 1. getRecommendation(p_j)

$\mu_{rt} \Leftarrow \frac{1}{A_i} \sum_{p_k \in A_i} rt_{ik}$

$\sigma_{rt} \Leftarrow \frac{1}{A_i} \sqrt{\sum_{p_k \in A_i} (rt_{ik} - \mu_{rt})}$

$th_{high} \Leftarrow 1$

$th_{low} \Leftarrow \mu_{rt} + \sigma_{rt}$

$rset \Leftarrow \emptyset$

while $\mu_{rt} - \sigma_{rt} \leq th_{low} and |rset| < \eta_{max}$ **do**

for all $p_k \in A_i$ **do**

if $th_{low} \leq rt_{ik} \leq th_{high}$ **then**

$rec \Leftarrow RequestRecommendation(p_k, p_j)$

$rset \Leftarrow rset \bigcup \{rec\}$

end if

end for

$th_{high} \Leftarrow th_{low}$

$th_{low} \Leftarrow th_{low} - \sigma_{rt}/2$

end while

return $rset$

Let er_{ij} denote p_i's estimation about reputation of p_j. In this calculation, r_{kj} values are considered with respect to η_{kj} as shown:

$$er_{ij} = \frac{1}{\beta_{er}} \sum_{p_k \in T_i} (rt_{ik} . \eta_{kj} . r_{kj}).$$

$\beta_{er} = \sum_{p_k \in T_i} (rt_{ik} . \eta_{kj})$ is the normalization coefficient.

Then p_i calculates estimation about competence and integrity belief of p_j denoted by ecb_{ij} and eib_ij, respectively.

$$ecb_{ij} = \frac{1}{\beta_{ecb}} \sum_{p_k \in T_i} (rt_{ik} . sh_{kj} . cb_{kj}),$$

$$eib_{ij} = \frac{1}{\beta_{ecb}} \sum_{p_k \in T_i} (rt_{ik} . sh_{kj} . ib_{kj}).$$

Both these equations represent own experience of p_i's acquaintances with p_j, where $\beta_{ecb} = \sum_{p_k \in T_i} (rt_{ik} . sh_{kj})$ is the normalization coefficient.

p_i calculates the average of history sizes in all recommendations, $\mu_{sh} = \sum_{p_k \in T_i} (sh_{kj})/t_i$ value.With these observations, r_{ij} is calculated as follows:

$$r_{ij} = \frac{\lfloor \mu_{sh} \rfloor}{sh_{max}} (ecb_{ij} - eib_{ij}/2) + (1 - \frac{\lfloor \mu_{sh} \rfloor}{sh_{max}}) er_{ij}$$

3) Recommendation Trust Metric (rt_ik): According to p_k's recommendation p_i updates recommendation trust metrics(rt_{ik}) value.

Three parameters are calculated the satisfaction (rs_{ik}^z),, weight (rw_{ik}^z), and fading effect (rf_{ik}^z) of p_i's z^{th} recommendation from p_k, respectively. Then the calculation of rs_{ik}^z is as follows:

$$rs_{ik}^z = ((1 - \frac{|r_{kj} - er_{ij}|}{er_{ij}}) + (1 - \frac{|cb_{kj} - ecb_{ij}|}{ecb_{ij}} + (1 \frac{|ib_{kj} - eib_{ij}|}{eib_{ij}}))/3$$

The weight of the recommendation should be calculated with respect to sh_{kj}, η_{kj}, $\lfloor \mu_{sh} \rfloor$ values. p_i calculates rw_{ik}^z as follows:

$$rw_{ik}^z = \frac{\lfloor \mu_{sh} \rfloor}{sh_{max}} \frac{sh_{kj}}{sh_{max}} + (1 - frac \lfloor \mu_{sh} \rfloor sh_{max}) \frac{\eta_{kj}}{\eta_{max}} .$$

p_i calculates rt_{ik} in similar way to st_{ik}

$$rcb_{ik} = \frac{1}{\beta_{rcb}} \sum_{z=1}^{rh_{ik}} (rs_{ik}^z . rw_{ik}^z . f_{ik}^z)$$

$$rib_{ik} = \sqrt{\frac{1}{rh_{ik}} \sum_{z=1}^{rh_{ik}} (rs_{ik}^z . rw_{ik}^\mu . rf_{ik}^\mu - rcb_{ik})^2}$$

$$rt_{ik} = \frac{rh_{ik}}{rh_{max}} (rcb_{ik} - rib_{ik}/2) + \frac{rh_{max} - rh_{ik}}{rh_{max}} r_{ik}.$$

D. Experimental Setup:

In our experiment we have used 7 P2P client setup and 5 upload peers with 3 download peers. Hardware configuration for all the peers is same to maintain consistancy. Hardware configuration for each peer is described in below table:

Hardware Configuration	
Processor	Pentium IV
Speed	1.1 Ghz
RAM	256 MB(min.)
Hard Disk	20 GB

Table 2: Hardware Configuration

We have implemented module on java platform; software configuration is shown in below table:

Software Configuration	
Operating System	Windows XP
Programming Language	Java
DATABASE	SQL Server 2005
Devlopment Tool	Eclipse

Table 3: Software Configuration

1) Screen Shots:

Server:

In P2P file sharing Application a server is dedicated to provide services to the requesting peers.

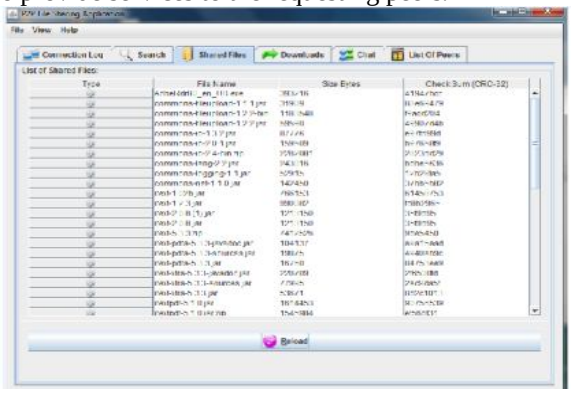


Fig 2:Server Side setup

Client

In P2P file sharing application client is one who select uploader with highest service trust value.

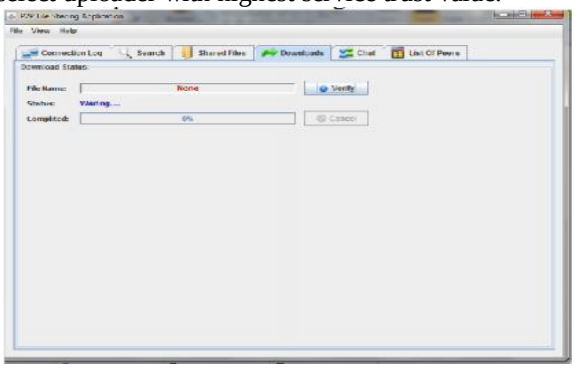


Fig 3:Client Side setup

File Sharing

File sharing refers to the uploading and downloading of file over a P2P network.

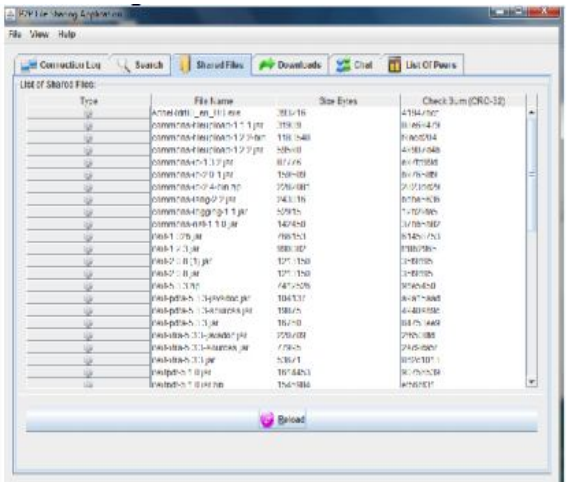


Fig 4:File sharing Application windows

IV. RESULT

In our experiment we have studied both service and recommendation-based attacks. A peer is called good one who uploads authentic file and also gives fair recommendation; at the other side a malicios peer perform service as well as recommendation-based attacks.

for the malicios peer we have studied 4 different attack behaviors, they are as follows:

1) Naive: In this case malicious peer uploads inauthentic file and also gives unfairly low recommendation about other peers.

2) Descriminatory: Here malicious peer targets group of peers and always upload inauthentic files to them.It also gives unfairly low recommendations about the victims but with other peers it behaves as a good peer.

3) Hypocritical: Here with x percent probability malicious peer uploads inauthentic file and gives unfairly low recommendations, but at other time it behaves as a good peer.

4) Oscillatory: By being good for long time period peer gets high reputation. Then it behaves as a naive attacker for a short period of time and again after some time period it becomes good peer.

We have studied all these attack behaviors one by one against our automated trust model and found that it mitigates all the attacks. We have also used shared secrete key for communication which helped us to maintain authorization in P2P network.

A. Analysis of attacks

We have studied service and recommendation-based attacks individually.

1) Service-based attacks: When any malicious peer uploads inauthentic file, then it is recorded as servic-based attack.

Percentage of attacks prevented				
	Attack type	SORT	FloodRQ	trust model
Malicious peers	Naive	65.3	64.7	68.3
	Descriminatory	72.2	72.7	75.3
	Hypocritical	43.0	47.5	46.6
	Oscillatory	37.0	44.0	44.2

Table 4:Service-based attacks

Our experiments expected results are compared with SORT model [1]. Above table shows the percentage of service-based attacks preventad by each trust model. While calculating per- centage of success we have considered base case to understand how many attacks can happen without using any trust model.

All the detected attacks for each trust model is compared with the base case to determine the percentage of attacks prevented.

In the above table we have shown result for Self-Organizing trust model (SORT) and an automated trust model.

Result shows that both the method are able to find more than sixty percent of malicious peer. Naive case shows high percentage of success because in naive case good peer identi- fies a naive attacker after having the first interaction. In all the other attacks good peer is not able to find an attacker in the first interaction.Trust model interacts less with the strangers as its set of acquitances grows and it helps to decrease service-based attacks with time.

2) Recommendation-based attacks: When any peer gives misleading recommendation then it is treated as a recommendation-based attack.

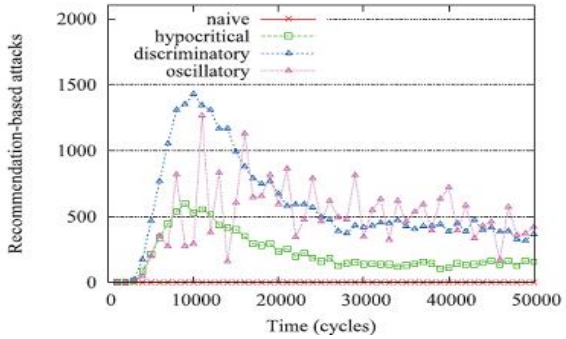


Fig 5:Recommendation-based attacks

Above figure shows percentage of recommendation-based attacks in malicious network. With our implemented trust model peers form their own trust network and do not re- quest any recommendation from any untrustworthy peers. An automated trust model collects recommendations from limited trustworthy peers and avoid overloading. It also check peers authenticity by using shared secrete key. Our experi- ment on recommendation-based attacks shows that collecting recommendations from acquaintances provide more accurate information.

CONCLUSION

An automated trust model implemented for P2P system, in which a peer can develop its own trust network in its proximity. A peer can distinguish between trustworthy and untrustworthy peers. Peer develop trust relationship with good peers. Each peer develpop its own local view of trust about the peer with whom he interacted in the past using two metrics, service trust and recommendation trust. Parameters like satisfaction, weight and fading effect are used while evaluating interactions and recommendations. While calculating recommendation, recom- mender's own experience, information from its acquaintances and level of confidance in the recommendation are considered. Interaction gives certain information about peer which is used to calculate service trust.

FUTURE SCOPE

It is difficult for a peer to determine a recommendation- based attack as a recommender might be cheated by attackers, so their is no any proof to prove that the recommendation is intentionally given. A selection algorithm can be used for load balancing to utilize all resources of peers. Thus this issue nees more investigation in future work. Trust information is not enough to solve all security related problem in P2P system but it can improve security and effectiveness of this system. Our experiment deals with both srvice and recommendation based attacks; more work is neccessory to deal with the recommendation base problems.

ACKNOWLEDGMENT

I would like to thank all the teachers of Department of Computer Engineering that patiently responded to the many queries that I had taken to them. In particular, I would like to thank my guide Prof. Sachin Malve for supporting my work. I would also like to thank Dr. Milindkumar Sarode for their helpful comments on this project.

REFERENCES

[1] Ahmet Burak Can, Bharat Bhargava,"Sort:A self Organizing Trust Model for Peer-to-peer System, " IEEE Trans., Vol. 10, NO.1 Feb 2013.

[2] S. Kamvar, M. Schlosser, H. Garcia-Molina, "The (Eigentrust) Algo- rithm for Reputation Management in P2P Networks, " Proc. 12th World Wide Web Conf, 2003.

[3] F. Cornelli, E. Damiani, P. Samarati, "Choosing Reputable Servents in a P2P Network, " Proc. 11th WWW Conf, 2002.

[4] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System, " Proc. 10th Int. Conf. Information and Knowledge Management, 2001.

[5] A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks, " Proc. IEEE/ACM Fourth Intl Symp. Cluster Computing and the Grid, 2004.

[6] L. Xiong and L. Liu, "Peertrust: Supporting Reputation-Based Trust for Peer-to-Peer Ecommerce Communities, " IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, Pg.No.843 to Pg.No.857, 2004.

[7] R. Zhou, K. Hwang, and M. Cai, "Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks, " IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9, Pg.No.1282 to Pg.No.1295, Sept. 2008.

[8] R. Guha, R. Kumar, P. Raghavan, A. Tomkins, "Propagation of Trust and Distrust, " Proc. 13th Int. Conf. World Wide Web (WWW), 2004.

[9] J. Douceur, The Sybil Attack," Proc. First Intl Workshop Peer-to- Peer Systems (IPTPS), " 2002.

[10] H. Yu, M. Kaminsky, P.B. Gibbons, A. Flaxman, "Sybilguard: De- fending against Sybil Attacks via Social Networks, " ACM SIGCOMM Computer Comm. Rev., vol. 36, no. 4, Pg.No.267 to Pg.No.278, 2006.

[11] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-Resilient Online Content Voting, " Proc. 6thth USENIX Symp. Networked Systems Design and Implementation (NSDI), 2009.

[12] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A Survey of Attack and Defense Techniques for Reputation Systems, " ACM Computing Surveys, vol. 42, no. 1, Pg.No.1 to Pg.No.31, 2009.

[13] K. Aberer, A. Datta, and M. Hauswirth, "P-Grid: Dynamics of Self- Organization Processes in Structured P2P Systems, " Peer-to-Peer Systems and Applications, vol. 3845, 2005.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network, " ACM SIGCOMM Computer Comm. Rev., vol. 31, no. 4, Pg.No.161 to Pg.No.172, 2001.

★★★