Research Paper Engineering

A Self-Organizing Trust Worthy Model for Peer-To-Peer Systems Environment

Nirmal Sam | Asst.professor CSE SRM university Chennai.

Ramya Sathish Kumar | ME CSE SRM university Chennai

ABSTRACT

The objective of the work is to prevent peer-to-peer systems exposes them to malicious activity. Building trust relationships among peers can mitigate attacks of malicious peers. This paper presents distributed algorithms that enable a peer to reason about trustworthiness of other peers based on past interactions and recommendations. Peers create their own trust network in their proximity by using local information available and do not try to learn global trust information. Two contexts of trust, service, and recommendation contexts, are defined to measure trustworthiness in providing services and giving recommendations. Interactions and recommendations are evaluated based on importance, recentness, and peer satisfaction parameters. Additionally, recommender's trustworthiness and confidence about a recommendation are considered while evaluating recommendations. Simulation experiments on a file sharing application show that the proposed model can mitigate attacks on 16 different malicious behavior models. In the experiments, good peers were able to form trust relationships in their proximity and isolate malicious peers.

KEYWORDS

1 INTRODUCTION

PEER-TO-PEER (P2P) systems rely on collaboration of peers to accomplish tasks. Ease of performing malicious activity is a threat for security of P2P systems. Creating long-term trust relationships among peers can provide a more secure environment by reducing risk and uncertainty in future P2P interactions. However, establishing trust in an unknown entity is difficult in such a malicious environ-ment. Furthermore, trust is a social concept and hard to measure with numerical values. Metrics are needed to represent trust in computational models. Classifying peers as either trustworthy or untrustworthy is not sufficient in most cases. Metrics should have precision so peers can be ranked according to trustworthiness. Interactions and feedbacks of peers provide information to measure trust among peers. Interactions with a peer provide certain information about the peer but feedbacks might contain deceptive information. This makes assessment of trust-worthiness a challenge.

In the presence of an authority, a central server is a preferred way to store and manage trust information, e.g., eBay. The central server securely stores trust information and defines trust metrics. Since there is no central server in most P2P systems, peers organize themselves to store and manage trust information about each other [1], [2]. Management of trust information is dependent to the structure of P2P network. In distributed hash table (DHT)-based approaches, each peer becomes a trust holder by storing feedbacks about other peers [1], [3], [4]. Global trust information stored by trust holders can be accessed through DHT efficiently. In unstructured networks, each peer stores trust information about peers in its neighbor-hood or peers interacted in the past [2], [5], [6]. A peer sends trust queries to learn trust information of other peers. A trust query is either flooded to the network or sent to neighborhood of the query initiator. Generally, calcu-lated trust information is not global and does not reflect opinions of all peers.

We propose a Self-ORganizing Trust model (SORT) that aims to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity. No a priori information or a trusted peer is used to leverage trust establishment. Peers do not try to collect trust informa-tion from all peers. Each peer develops its own local view of trust about the peers interacted in the past. In this way, good peers form dynamic trust groups in their proximity and can isolate malicious peers. Since peers generally tend to interact with a small set of peers [7], forming trust relations in proximity of peers helps to mitigate attacks in a P2P system.

In SORT, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. Using a service of a peer is an interaction, which is evaluated based on weight (importance) and recentness of the interaction, and satisfaction of the requester. An acquaintance's feedback about a peer, recom-mendation, is evaluated based on recommender's trust-worthiness. It contains the recommender's own experience about the peer, information collected from the recommen-der's acquaintances, and the recommender's level of con-fidence 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.

and defines two contexts of trust: service and recommendation contexts. Information about past interactions and recom-mendations are stored in separate histories to assess competence and integrity of acquaintances in these contexts.

SORT defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases. Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations. When calculating the reputation metric, recommendations are evaluated based on the recommendation trust metric.

We implemented a P2P file sharing simulation tool and conducted experiments to understand impact of SORT in mitigating attacks. Parameters related to peer capabilities (bandwidth, number of shared files), peer behavior (online/ offline

periods, waiting time for sessions), and resource distribution (file sizes, popularity of files) are approximated to several empirical results [8], [9], [10]. This enabled us to make more realistic observations on evolution of trust relationships. We studied 16 types of malicious peer behaviors, which perform both service and recommenda-tion-based attacks. SORT mitigated service-based attacks in all cases. Recommendation-based attacks were contained except when malicious peers are in large numbers, e.g., 50 percent of all peers. Experiments on SORT show that good peers can defend themselves against malicious peers without having global trust information. SORT's trust metrics let a peer assess trustworthiness of other peers based on local information. Service and recommendation contexts enable better measurement of trustworthiness in providing services and giving recommendations.

Outline of the paper is as follows: Section 2 discusses the related research. Section 3 explains the computational model of SORT. Section 4 presents the simulation experi-ments and results. Section 5 summarizes the results and possible future work directions.

2 THE COMPUTATIONAL MODEL OF SORT

We make the following assumptions. Peers are equal in computational power and responsibility. There are no privileged, centralized, or trusted peers to manage trust relationships. Peers occasionally leave and join the network. A peer provides services and uses services of others. For simplicity of discussion, one type of interaction is con-sidered in the service context, i.e., file download.

2.1 Preliminary Notations

p_i denotes the ith peer. When p_i uses a service of another peer, it is an interaction for p_i. Interactions are unidirectional. For example, if p_i downloads a file from p_j, it is an interaction for p_i and no information is stored on p_j.

If p_i had at least one interaction with p_j, p_j is an acquaintance of p_i. Otherwise, p_j is a stranger to p_i. A_j denotes p_j's set of acquaintances. A peer stores a separate history of interactions for each acquaintance. SH_{ij} denotes p_i's service history with p_j where sh_{ij} denotes the current size of the history. sh_{max} denotes the upper bound for service history size. Since new interactions are appended to the history, SH_{ij} is a time ordered list.

Parameters of an interaction. After finishing an interac-tion, p_i evaluates quality of service and assigns a satisfaction value for the interaction. Let $0 \_ s^k_{ij} \_ 1$ denote p_i's satisfaction about kth interaction with p_j. If an interaction is not completed, $s^k_{ij} = 0$. An interaction's importance is measured with a weight value. Let $0 \_ w^k_{ij} \_ 1$ denote the weight of kth interaction of p_i with p_j.

TABLE 1
Notations on the Trust Metrics

Notation	Description
s^k_{ij}	p_i's satisfaction about k^{th} interaction with p_j
w^k_{ij}	weight of p_i's k^{th} interaction with p_j
f^k_{ij}	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_{ik}	p_i's recommendation trust about p_k
sh_{ij}	size of p_i's service history with p_j

file, average download speed, average delay, retransmis-sion rate of packets and online/offline periods of the service provider might be some parameters to calculate s^k_{ij}. Size and popularity of a file might be some parameters to calculate w^k_{ij}. In Section 4, we suggest equations to calculate these values in a file sharing application.

Importance of an old interaction should decrease as new interactions happen. The fading effect parameter addresses this issue and forces peers to stay consistent in future interactions. Old interactions lose their importance so a peer cannot easily misbehave by relying on its good history. Let $0 \_ f^k_{ij} \_ 1$ denote the fading effect of kth interaction of p_i with p_j. It is calculated as follows:

$$f^k_{ij} = \frac{k}{sh_{ij}}; \quad 1 \_ k \_ sh_{ij}: \qquad (1)$$

After adding (or deleting) an interaction to SH_{ij}, p_i recalculates f^k_{ij} values. The fading effect can be defined as a function of time but it has to be recalculated whenever its value is needed. Furthermore, interactions continually lose value with time causing all peers to lose reputation even though no bad interaction happens.

Let $SH_{ij} = f\_^1_{ij}; \_^2_{ij}; \ldots; \_^{shij}_{ij}$ g, where $\_^k_{ij} = (s^k_{ij}; w^k_{ij})$ is a

tuple representing the information about kth interaction. When adding a new interaction, $\_^1_{ij}$ is deleted if $sh_{ij} = sh_{max}$. An interaction is deleted from the history after an expiration

REFERENCES

[1] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," Proc. 10th Int'l Conf. Information and Knowl-edge Management (CIKM), 2001. | | [2] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," Proc. 11th World Wide Web Conf. (WWW), 2002. | | [3] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The (Eigentrust) Algorithm for Reputation Management in P2P Networks," Proc. 12th World Wide Web Conf. (WWW), 2003. | | [4] 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, pp. 843-857, July 2004. | | [5] A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks," Proc. IEEE/ACM Fourth | | [6] Int'l Symp. Cluster Computing and the Grid (CCGRID), 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, pp. 1282-1295, Sept. 2008. | | [8] J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective," Proc. 32nd ACM Symp. Theory of Computing, 2000. | | [9] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," Proc. Multimedia Computing and Networking, 2002. | | [10] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," IEEE Internet Computing, vol. 6, no. 1, pp. 50-57, Jan. 2002. | | [11] S. Saroiu, K. Gummadi, R. Dunn, S.D. Gribble, and H.M. Levy, "An Analysis of Internet Content Delivery Systems," Proc. Fifth USENIX Symp. Operating Systems Design and Implementation (OSDI), 2002. | |