OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Making Trust Relationship For Peer To Peer System With Secure Protocol

Miss. I. Jancy[1], Mr. S. Balamurugan[2]
[1, 2] *(PG Student, Assistant Professor, Sri ManakulaVinayagar Engineering College, Pondicherry-605106)*

Abstract: *In the 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 trust worthiness 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. So, neighbouring node will give the recommendation to peer. Based on the recommendation only Peer decides whether the node is good (or) malicious. Find the node is malicious node means peer will not interact with malicious node. Isolate the malicious node from the network. Find the node is good means peer interact with good peer. Peer stores a separate history of interactions for each Acquaintance. This paper also discuss the malicious threats, privacy concerns, and security risks of three commonpeer-to-peer network systems that are gaining popularity today. The malicious threats discussed willinclude how malicious threats can harness existing peer-to-peer networks, and how peer-to-peernetworking provides an additional (potentially unprotected) vector of delivery for malicious code.*

Index Terms: *Peer-to-peer systems, trust management, reputation, security.*

I. Introduction

Open nature of 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 trust worthiness 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. 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.

And Therefore, classic peer-to-peer unaware viruses could inadvertently be transmitted via a peer-to-peer network. Viruses could also take advantage of the regular use of a peer-to-peer network. For example, viruses could specifically attempt to copy themselves to or infect files within the shared peer-to-peer space.

A peer sharing files is called an uploader. A peer downloading a file is called a downloader. The set of peers who downloaded a file from a peer are called downloaders of the peer. An ongoing download/ upload operation is called a session. Simulation parameters aregenerated based on results of several empirical studies [6], [7] to make observations realistic. A file search request reaches up to 40 percent of the network and returns online uploaders only. A file is downloaded from one uploader to simplify integrity checking. All peers are assumed to have antivirus software so they can detect infected files Four different cases are studied to understand effects of trust calculation methods under attack conditions:

No trust. Trust information is not used for uploader selection. An uploader is selected according to its bandwidth. This method is the base case to understand if trust is helpful to mitigate attacks.

No reputation query. An uploader is selected based on trust information but peers do not request recommendations from other peers. Trust calculation is done based on SORT equations but reputation (r) value is always zero for a peer. This method will help us to assess if recommendations are helpful.

SORT. 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.

Flood reputation query.SORT equations are used but a reputation query is flooded to the whole network. This method will help us to understand if getting more recommendations is helpful to mitigate attacks.

In SORT, to evaluate interactions and recommendations better, importance, recentness, and peer satisfaction parameters are considered. Recommender‟s trustworthiness and confidence about recommendation are considered when evaluating recommendations. Additionally, service and recommendation contexts are separated. This enabled us to measure trustworthiness in a wide variety of attack scenarios. Most trust models do not consider how interactions are rated and assume that a rating mechanism exists. In this study, we suggest an interaction rating mechanism on a file sharing application and consider many real-life parameters to make simulations more realistic. A good peer uploads authentic files and gives fair recommendations. A malicious peer (attacker) performs both service and recommendation-based attacks. Four different attack behaviors are studied for malicious peers: naive, discriminatory, hypocritical, and oscillatory behaviours. A non-malicious network consists of only good peers. A malicious network contains both good and malicious peers.

***And Therefore, classic peer-to-peer unaware viruses could inadvertently be transmitted via a peer-to-peer network. Viruses could also take advantage of the regular use of a peer-to-peer network. For example, viruses could specifically attempt to copy themselves to or infect files within the shared peer-to-peer space.

The systems discussed include the Napster, Gnutella, and Freenet protocols. These protocols will be examined due to their popularity and different methods of achieving peer-to-peer networking.

Many other peer-to-peer networking systems exist (for example, Microsoft Networking), and while not explicitly discussed, conclusions can be applied to these systems as well.

II. Virus Protection

Viruses could actually harness the existing peer-to-peer network infrastructure to propagate themselves. For example, a worm could set up a servent on an infected system. The user with the infected system does not have to initially be part of the peer-to-peer network. Then, this servent could return the exact matches for incoming search queries, and those downloading and executing the file will in turn become infected. An example of such a worm is W32.Gnuman.

Since peer-to-peer malicious threats still need to reside on the system's current desktop, a scanninginfrastructure can provide protection against infection. However, desktop protection may not proveto be the best method in the future.

Should peer-to-peer networking become standard in home and corporate computing infrastructures,network scanning may become more desirable. Such scanning is not trivial since, by design,peer-to-peer transfer of data does not pass through a centralized server, such as an email server.

Systems such as network-based IDS may prove useful, as well as gateway/proxy scanning to preventmalicious threats from using peer-to-peer connections that pass inside and outside of organizations.

However, peer-to-peer networking models such as Freenet will render networking scanning uselesssince all data is encrypted. You will not be able to scan data that resides in the DataStore on a system.

Detection of threats passed via Freenet type models will only be scanned on the unencrypted file atthe desktop just prior to execution. The issue of encryption reinforces the necessity fordesktop-based, antivirus scanning.

III. 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 considered in the service context, i.e., file download.

3.1 Preliminary Notations

pi denotes the ith peer. When pi uses a service of another peer, it is an interaction for pi. Interactions are unidirectional. For example, if pi downloads a file from pj, it is an interaction for pi and no information is stored on pj. If pi had at least one interaction with pj, pj is an acquaintance of pi. Otherwise, pj is a stranger to pi. Ai denotes pi's set of acquaintances. A peer stores a separate history of interactions for each acquaintance. SHij denotes pi's service history with pj where shij denotes the current size of the history. shmax denotes the

upper bound for service history size. Since new interactions are appended to the history, SHij is a time ordered list.

Parameters of an interaction. After finishing an interaction, pi evaluates quality of service and assigns a satisfaction value for the interaction. Let 0 _ skij _ 1 denote pi's satisfaction about kth interaction with pj. If an interaction is not completed, skij ¼ 0. An interaction's importance is measured with a weight value. Let 0 _ wkij _ 1 denote the weight of kth interaction of pi with pj. Semantics to calculate skij and wkij values depend on the application. In a file sharing application, authenticity of a 16 IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 10, NO. 1, JANUARY/FEBRUARY 2013 file, average download speed, average delay, retransmission

Background Protocols

- **GNUTELLA**

Gnutella does not utilize a centralized server. Each computer is a client as well as a server, hereinafter called a servent. Such a true peer-to-peer networking model decreases reliability, speed, and search capabilities, and increases network traffic. Figure 1 illustrates the standard communication process involved in obtaining a file.

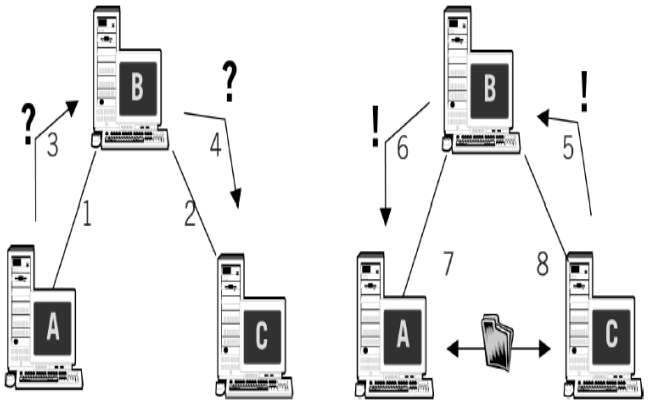

Fig.1 Standard Communication Process: Obtaining a File

- **NAPSTER**

The Napster peer-to-peer networking model involves a centralized directory server. Clients primarily communicate with a directory server that passes messages among, and maintains particular states of, clients. Figure 2 illustrates the standard communication process involved in downloading a file in the Napster protocol.Figure 2 illustrates the standard communication process involved in downloading a file in the Napster protocol.

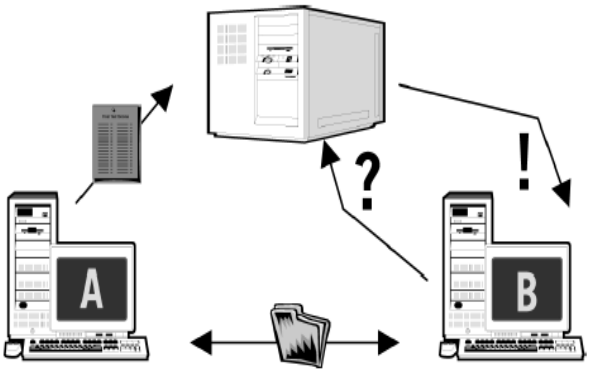

Fig.2 Standard Communication Process: Downloading a File in Napster Protocol

- **FREENET**

The Freenet model of exchange is similar to Gnutella, being a true peer-to-peer model. However, users do not have control over what content is held in their shared space, known as a DataStore. A user inserts a file into the Freenet network, where it is encrypted and propagated along the network to an appropriate node determined by a unique key, which identifies the file.

This allows data with close keys to be sorted to the same nodes on the network, which causes the clustering of close key data. This allows a fast response to search queries. Since all data is encrypted, users only have control of the amount of space they wish to make available on their systems, not the content that resides on their systems. Figure 3 illustrates the standard communication process involved in obtaining a file in the Freenet protocol.

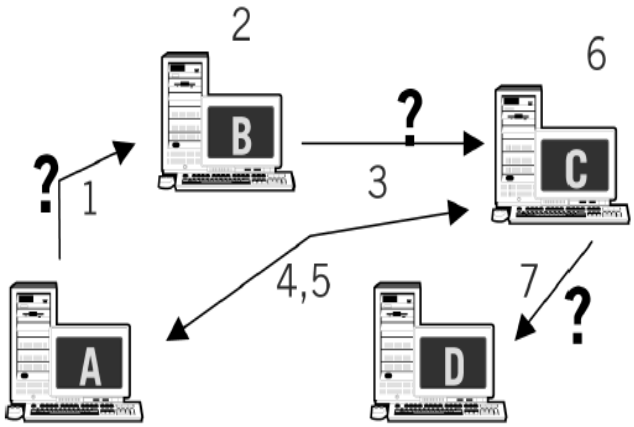

Fig.3 Standard Communication Process: Downloading a File in GNUTELLA Protocol

3.2 Service Trust Metric*(stij)*

This section describes the calculation of service trust metric. A peer first calculates competence and integrity belief values using the information about service interactions. *Competence belief* is based on how well an acquaintance satisfied the needs of interactions .*cbij*denotes the competence belief of *pi* about *pj* in the service context. Average behavior in the past interactions can be a measure of competence belief.

$$cbij = \overline{cb}(eijk \text{ ¢ } wijk \text{ ¢ } fijk) \qquad (2)$$

1*shij*

=1

X*k*

IV. PerformanceAnalysis&Design

Downloading a file is an interaction. A peer sharing files is called an uploader. A peer downloading a file is called adownloader. The set of peers who downloaded a file from a peer are called ownloaders of the peer. An ongoingdownload/ upload operation is called a session.

4.1 (a) Existing System

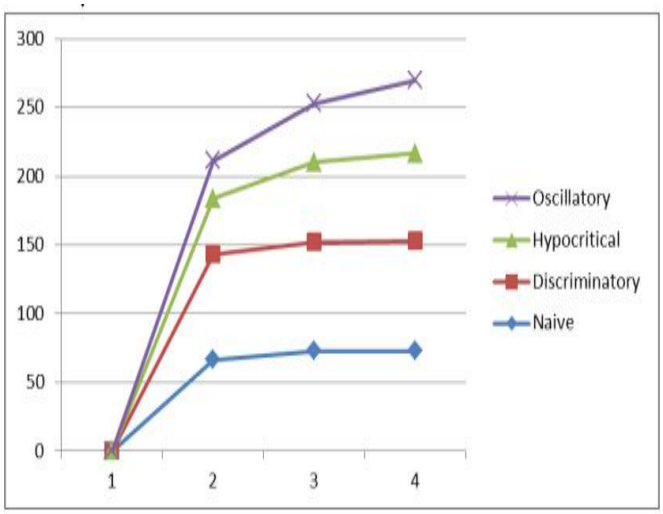

Fig.4Existing System (15% malicious)

(b) Proposed System

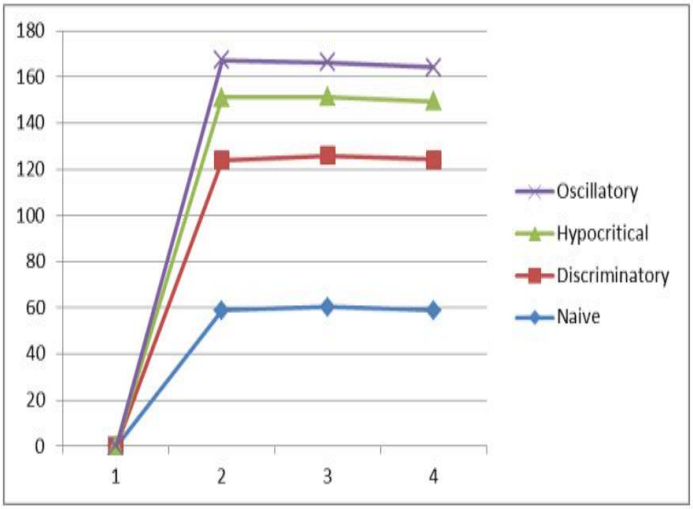

Fig.5Proposed System (50% malicious)

V. Conclusion

Peer-to-peer networks obviously pose a danger as an additional vector of delivery. Their impact on security will depend on the adoption of peer-to-peer networks in standard computing environments. If systems use peer-to-peer networks as email is used today, then they will be significant methods of delivery of malicious code. The use of two-way network communication also exposes the system to potential remote control.

More importantly, the usage of a peer-to-peer network creates a hole in a firewall and can lead to the exporting of private and confidential information. Therefore, administrators should begin analyzing their networks for peer-to-peer network usage and configure firewalls and systems accordingly to limit or prevent their usage.

REFERENCES

[1] AhmetBurakCan and Bharat(2013),"A Self-Organizing Trust Model for Peer-to-Peer Systems"IEEETrans. Dependable and Secure Computing,vol 10,No.1.

[2] Aberer.K and Despotovic.Z(2001), „Managing Trust in a Peer-2-Peer Information System" Proc. 10th Intl Conf. Information and Knowledge Management (CIKM).

[3] Kamvar.S, Schlosser.M, and Garcia-Molina.H,(2003) „The (Eigentrust) Algorithm for Reputation Management in P2P Networks" Proc. 12th World Wide Web Conf. (WWW).

[4] SelcukA.A ,Uzun.E, and Pariente.M.R(2004), „A Reputation-Based Trust Management System for P2P Networks" Proc. IEEE/ACM Fourth Int"lSymp. Cluster Computing and the Grid (CCGRID).

[5] Zhou. R, Hwang. K, and Cai. M(2008), „Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks" IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9.

[6] Abdul-Rahman. A and Hailes.S(2008), „Supporting Trust in Virtual Communities" Proc. 33rd Hawaii Int"l Conf. System Sciences (HICSS).