World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:3, 2013

Attacks and Counter Measures in BST Overlay Structure of Peer-To-Peer System

Guruprasad Khataniar, Hitesh Tahbildar, and Prakriti Prava Das

***Abstract*—**There are various overlay structures that provide efficient and scalable solutions for point and range query in a peer-to-peer network. Overlay structure based on m-Binary Search Tree (BST) is one such popular technique. It deals with the division of the tree into different key intervals and then assigning the key intervals to a BST. The popularity of the BST makes this overlay structure vulnerable to different kinds of attacks. Here we present four such possible attacks namely index poisoning attack, eclipse attack, pollution attack and syn flooding attack. The functionality of BST is affected by these attacks. We also provide different security techniques that can be applied against these attacks.

***Keywords*—**BST, eclipse attack, index poisoning attack, pollution attack, syn flooding attack.

I. Introduction

PEER-TO-PEER can be defined as an autonomous, self-organized, scalable distributed system with shared resource pool without a single point of failure in which all nodes have identical capabilities and responsibilities and all communications are normally symmetric; and where main characteristics of the participants are decentralized resource usage and decentralized self-organization [1]. The notion of P2P was first established in 1969, in the first Request for Comments, RFC-1. The RFC implies a "host-to-host" connection, indiscriminate of a client-server categorization, which provides responses in the fashion of teletype (TTY) terminals [2]. However, the first true implementation of a P2P network was Usenet, developed in 1979. In Usenet, while end-user clients still access resources through servers, servers themselves peer with each other in the fashion of a P2P network, sending messages to each other on demand without a central authority. Since the late 1990s, there has been a surge of popularity in P2P network applications, mainly in the form of file sharing applications used to exchange multimedia files. Some of the most popular and high-profile file sharing protocols include Freenet [3], Napster [4], Gnutella [5] etc. The P2P networks have gained immense popularity by showing their strength in providing many services such as sharing files without the need for central servers, streaming multimedia with distributed load balancing, distributed backup systems etc as they are scalable, and resilient to node failure. So there is a tremendous growth in all types of P2P systems. In order to continue growing, P2P networks must be robust, and fault tolerant. P2P system has millions of concurrently active peers. With a huge user base and lack of any authentication, P2P networks can be leveraged by an attacker to launch a DDoS (Distributed Denial-of-Service) attack [6] against a victim machine on the Internet. The victim need not be a participant in the P2P network, and could be a web server, a mail server or even a home user's desktop.

P2P networks have indexes. An index in a P2P network is a set of mappings from keys to values. For a P2P file sharing network, the keys are file hashes and the values are locations at which the file corresponding to the file hash is present. Location is usually described by the tuple <NodeID, IP:Port>, where NodeID uniquely identifies a peer in the P2P network. Thus, an index record in a P2P file sharing network specifies which file is present at which location. An attacker can exploit a P2P file sharing network in two ways to launch DDoS attacks - index poisoning and routing table poisoning. In an index poisoning attack, the attacker plants false index records in a large number of peers. These false index records indicate that a popular file is present with the victim. When any peer tries to search for that file, it receives this false index record from the poisoned peer. Since the file is popular, there will be large number of requests for that file. On receiving the false index record, the searching peers try to connect to the victim, trying to download the file, filling up the number of allowed connections, preventing legitimate users from connecting to the victim. In the routing table poisoning attack, the attacker makes the victim, a neighbor of a large number of peers by sending them false node announcement messages. Whenever a peer receives a search query or a maintenance message, it may select the victim from its routing table and forward the message to the victim. If the attacker poisons the routing table of a large number of peers, the victim may receive a flood of search queries and maintenance messages, saturating the victim's link.

With the increasing popularity of the peer-to-peer network, the demand for securing such a distributed network is also increasing day by day. But there are few methods which describe the implementations of security techniques in a peer-to-peer network. As of today, several peer-to-peer models have been developed. Some of them relate to structured peer-to-peer systems and some relate to unstructured peer-to-peer systems. But there are few peer-to-peer models that are developed to find an efficient and scalable solution for range query to discover the contents in the presence of transient

Dr.Guruprasad Khataniar is with Computer Engineering Department, Assam Engineering Institute, Assam, India (phone: +91-9864055805; e-mail: drkhataniar@gmail.com).

Dr.Hitesh Tahbildar is with Computer Engineering Department, Assam Engineering Institute, Assam, India (phone: +91-9864018339; e-mail: tahbil@rediffmail.com).

Mrs.Prakriti Prava Das is with Computer Engineering Department, Assam Engineering Institute, Assam, India (phone: +91-9864078616; e-mail: prakritidas.das@gmail.com).

node populations. Overlay structure based on *m*-Binary Search Tree (BST) as proposed by B.K. Shrivastava et al. [7] is one such model, which is based on *m*-BST for providing an efficient and scalable solution for point as well as range query. The complete key space is divided into several key interval based on lexicographic order and each key interval is assigned to a Binary Search Tree. The BST overlay is scalable, fault tolerant and self-organizing in nature. In this paper, we present the possible type of security attacks that can affect this P2P model. We also present the countermeasures against each of those attacks.

II. Related Work

P2P file-sharing system has become the most popular Internet content delivery systems [8]. Sharing content files containing audio, video, data or anything in digital format is very common, and real time data, such as telephony traffic, is also passed using P2P networks. P2P file-sharing networks are built on a large number of peer hosts running the same software. Anonymity of these peers is the key for the popularity of the P2P network. The distinct features of P2P networks present some unique security vulnerabilities. For example, P2P client software usually caches IP addresses of recently accessed peers. Once vulnerability is discovered in the software and attacked by a malware, the attacking program is much easier to propagate since most likely all the peers in the cache have exactly the same vulnerability [9].

Naoumov et al. [10] discovered that with a large number of poisoned file indexes, an attacker can launch DDoS attacks against arbitrary host in the Internet, either inside or outside of the P2P network. Essentially, index poisoning attack turns a P2P file-sharing network into a DDoS attack platform without even altering peer software. Index poisoning attack is effective in all these networks despite their significant differences in protocol design, network structure and user population. In pseudo-distributed networks, the attacker would spoof its address to that of the victim, and then send poisoned index information to the tracker. In structured and unstructured networks, poisoned index propagate via attacker sending messages to peers/super nodes, and peers exchanging index information.

L. Wang [11] presents some security attacks in peer-to-peer environment. He shows several kinds of attacks like DoS and DDoS attacks, TCP Syn Flooding attack, Query Flooding attack, poisoning attacks etc. which affect a peer-to-peer system in general. J. Liang et al. [6] discuss the index poisoning attack in peer-to-peer file sharing systems. They show that both structured and unstructured P2P file sharing systems are highly vulnerable to index poisoning attack. They develop a novel and efficient methodology for determining index poisoning levels and pollution levels in file sharing systems. D.S. Wallach [12] has outlined several structured peer-to-peer overlays such as CAN [13], Chord [14], Pastry [15] and Tapestry [16], which are providing a self organizing substrate for large scale peer-to-peer applications. He has shown how cryptographic techniques can be applied to increase the security and trust for applications in the peer-to-peer network. M. Parashar et al. [17] propose three peer-to-peer application categories that have elements in common with most popular peer-to-peer applications: distributed file sharing, real-time communications and distributed computing. The properties of several common security enabling technologies such as public key cryptography, smart cards and steganography are measured based on real-world applications, simulation, and results of related research. Structured P2P overlay networks are widely used to deploy services. This characteristic makes such system attractive to thousands or millions of users and at the same time vulnerable to the phenomena of churn. The independent arrival and departure of thousands or millions of peers creates a collective effect called churn. An attacker could exploit this attack by generation peer joining and leaving the network fast enough to corrupt the best function of the network. To cope with churn, G. Khataniar et al. [18] pointed out that P2P networks should be designed to be able to efficiently handle the large number of peers joining the system for just a few minutes.

III. Attacks and Countermeasures

Since P2P systems inherently rely on the dependence of peers with each other, security implications arise from abusing the trust between peers. In a traditional client-server model, internal data need not be exposed to the client, but with P2P, some internals must be exposed to fellow peers in the name of distributing the workload. Attackers can leverage this in compromising P2P networks.

In the overlay structure based on *m*-BST, the unreliable nodes can frequently join and leave the system. Therefore the implementation of the security issues in such an unstable structure is quite challenging. In this section, we present different malicious activities in such a model along with their countermeasures.

A. Index Poisoning Attack

In index poisoning attack, the aim of the attacker is to make several peers believe that some popular file is present with the victim. To achieve this, the attacker *A* sends a location publish message to every crawled peer. In these messages, the attacker includes victim's IP address and port number. The attacker puts the file hash of a popular file, which is expected to receive lots of search queries. When a peer *B* receives such a publish message, it adds this file hash into its index along with the location of the victim. *B* does not verify whether the victim has the corresponding file or even that *A* or victim is a participant in the P2P network. When some peer *C* searches for that file, it may be told by some poisoned peer that victim has the file. The peer *C* then creates a TCP connection to the victim in order to download the file. The downloading peer then sends a protocol specific message, specifying the file that it wishes to download. Not understanding the message, the victim may ignore it, reply with some error message or may even terminate the connection. Unable to download the file, the downloading peer may retry after some time. Since there will be many peers searching for that popular file, victim will receive a large number of incoming connections, filling up its

TCP queue, and therefore making it deny connections to its legitimate users. Moreover, index poisoning attack is also dangerous because of its residue effects: the victim remains under attack even hours after the attacker has stopped poisoning the indexes. This is because the fake records persist in the indexes for hours, even after peers fails to download from the target host.

In index poisoning attack, the attacker sends massive number of bogus records to the *superpeer*. All bogus records point the target address of the popular files to one target victim host. When other peers want to download those files, they get bogus records from the *superpeer*. Then these peers make a TCP connection to the victim node. The other peers cannot get services from the victim node as the fooled peers have already occupied the connections. BST can also suffer from index poisoning attack. In *m*-BST overlay structure, each *superpeer* maintains the Group Routing Table (GRT). The GRT maintains the information about the key interval, load status and the *superpeer* vector of each BST. Suppose a peer *P* wants to join a BST network. The *P* knows an existing node *X* in the network. The *X* will select a group G_i for *P* and give *P* the *id* of superpeer S_{ij} of G_i. The position of the peer *P* is chosen in such a way that it makes the BST as complete as possible. Then *P* can send bogus records such as invalid key interval, invalid load status or wrong superpeer vector to the superpeer. The superpeer can enter this bogus information to the GRT. Since the same GRT is maintained by all the superpeers, so the bogus information spread all over the P2P network. In order to make the superpeer polluted, the attack peer must create a TCP connection to the superpeer and publish false key interval, false load status, false superpeer vector etc.

B. Eclipse Attack

In case of eclipse attack, the attacker controls a significant part of the network. Here a good node is surrounded by several malicious nodes and these malicious nodes work together to fool the good node.

BST overlay structure can also be vulnerable to eclipse attack. To have an eclipse attack in BST overlay structure, there must be several polluted nodes in the tree. The non-polluted nodes that are still exist in the tree are fooled by these polluted nodes. Here the attacker controls the large part of the BST and the unions of the polluted nodes try to fool the good nodes.

To have an eclipse attack, the indegree of an attacker must be higher than the average level of indegree of nodes in a peer-to-peer network. Here indegree means number of direct routes coming into a node and outdegree means number of direct routes going out of a node.

To deal with the eclipse attack, we first apply the countermeasures to the sybil attack [19]. This is because a sybil attack can be considered as a specific eclipse attack, if the attacker generates great amount of identifications to act as neighbours of a good node. We can do so by establishing a trusted certificate authority to make distinct entity has distinct identities. Then we concentrate on how to deal with the indegree and the outdegree of the attacker nodes.

C. Pollution Attack

The best way to corrupt P2P file sharing is to deposit into the file sharing system some junk pieces of data known as polluted files. In this way, attacker corrupts the content of shared file, rendering it unusable, and forwards the corrupted file to other peers. As a result, polluted files spread through the network and users become unable to distinguish polluted files from unpolluted file. To fight against polluted files, Dhungel et al. [20] propose four possible defenses: blacklisting, traffic encryption, hash verification, and chunk signing. Other mechanisms presented by Liang et al. [21]: detection without downloading, after receiving search results the mechanism attempt to determine whether the files in the results are polluted. Detection with downloading, for this class, the mechanism detects whether a file is polluted by first downloading portion of the file.

BST overlay structure can also suffer from pollution attack. Here, a specific file is corrupted by using some technique so that the corrupted file becomes unusable. Then the file is made available for sharing in large volumes. In a peer-to-peer network, there is no central server that can be used for storing the files and providing download. This means that all nodes can directly download the required files from the other peers. If the peers don't have any provision for distinguishing the polluted files from the non-polluted ones, then the peers download it into their own file sharing folders and other peers may download it from this particular peer. In this way, the corrupted file is being spread in the system.

D. Syn Flooding Attack

When a peer *P* wants to join the peer-to-peer network, it should know about an existing peer *X* in the BST. Peer *P* sends a request for joining the network to *X*. Suppose *P* wants to make the peer-to-peer network unstable. Then it sends several requests to the superpeer S_{ij} with different IP addresses. Superpeer S_{ij} then responses with its *id* and try to place the new peers in the tree such that the tree becomes complete. Since most of the requests are false, so the replies coming from the S_{ij} lost. During this time, if the number of requests is very high, then S_{ij} runs out of the resources and may crash.

One possible remedy to such an attack is that when a superpeer gets a TCP syn packet, it hashes and encrypts some security values such as client IP address and Port number to get an initial sequence value. When S_{ij} get another syn request, it also replies with the syn+ack. When S_{ij} receives an ack, it verifies the ack by hashing and then comparing with the initial sequence value. If both the values are same then the superpeer allocates data section to handle the data section.

IV. Security Mechanisms

It is very difficult to define a security mechanism that can remove all the vulnerabilities of existing peer-to-peer overlays. In this section we discuss how cryptographic solutions can be useful in developing a secure peer-to-peer network.

Cryptography is an art and science of achieving security by

encoding messages to make them non-readable. Among the various cryptographic tools, encryption and authentication are two most commonly used crypto primitives. Encryption is the conversion of plain text or data into unintelligible form by means of a reversible translation, based on a translation table or algorithm. Symmetric key encryption algorithm and asymmetric key encryption algorithms are the two types. The first one uses the same key for both encryption and decryption while the later uses two different keys for encryption and decryption. Examples of symmetric key encryption algorithms include Data Encryption Standard (DES) and Advance Encryption Standard (AES), which are standardized by National Institute of Standards. Example of asymmetric key encryption algorithm is RSA algorithm, which involve the finding of two large prime numbers.

Encryption is very much useful in BST overlay structure security. If the confidential information is encrypted, then even if the attacker peer able to get the message when it is transmitting over an insecure BST network, he cannot decrypt it without a proper decryption algorithm. Thus the security risks will be subsequently reduced.

Authentication is another security tool in computer science. It ensures that the origin of a electronic message or document have been correctly identified. It guarantees that an object is in fact who or what that object declares itself to be. It can also play positive roles in BST overlay structure security. For example, combining secure authentication of each peer with message encryption, a BST system can prevent eavesdropping attacks.

V. Conclusion

In this paper, we present the different types of security measures in the overlay structure based on m-Binary Search Tree (BST). We present four basic types of security attacks in this model namely index poisoning, eclipse, pollution and syn flooding attacks. The defending actions against these attacks have also been clarified. Then we present how cryptographic solutions such as encryption and authentication can be useful in developing a secure overlay structure based on *m*-Binary Search Tree.

References

[1] G. Khataniar and D. Goswami, *SHP: A Hierarchical Protocol to improve Performance of Peer to Peer Systems*, International Journal of Peer to Peer Networks (IJP2P) Vol.3, No5 pages 1-21, 2012.

[2] Steve Crocker, *RFC-1 Host Software*, http://www.faqs.org/ftp/rfc/rfc1.txt, 1969.

[3] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, *Freenet: A distributed anonymous information storage and retrieval system*, In Lecture Notes in Computer Science, pages 46–66, 2000.

[4] J. F. Buford, H. Yu, and E. K. Lua, *P2P Networking and Applications*, Morgan Kaufmann, 2009.

[5] M. Ripeanu, *Peer-to-peer architecture case study: Gnutella network*, In Proceedings of the IEEE 1st International Conference on Peer-to-Peer Computing, pages 99–100, Linkoping, Sweden, Aug 2001.

[6] Jian Liang, Naoum Naoumov, and Keith W. Ross, *The Index Poisoning Attack in P2P File Sharing Systems*, In IEEE Conference on Computer Communication, Barcelona, Spain, 2006.

[7] B. K. Shrivastava, G. Khataniar and D. Goswami, *Binary Search Tree: An Efficient Overlay Structure to Support Range Query*, 27th International Conference on Distributed Computing Systems Workshops, 2007.

[8] S. Saroiu , K. P. Gummadi, J. D. Richard, S. D. Gribble, and H. M. Levy, *An Analysis of Internet Content Delivery Systems*, ACM Operating System Review, Vol. 36, pp. 315- 327, 2002.

[9] L. Zhou, L. Zhang, F. McSherry, N. Immorlica, M. Costa, and S. Chien, *A First Look at Peerto-Peer Worms: Threats and Defenses*, 4th International Workshop on Peer-To-Peer Systems (IPTPS'05), Ithaca, NY, 2005.

[10] N. Naoumov and K. W. Ross, *Exploiting P2p Systems for DDoS Attacks*, International Workshop on Peer-to-Peer Information Management (keynote address), Hong Kong, 2006.

[11] Lin Wang, *Attacks Against Peer-to-peer Networks and Countermeasures*, Helsinki University of Technology, 2006.

[12] Dan S. Wallach, *A Survey of Peer-to-Peer Security Issuues* Rice University, USA.

[13] Sylvia Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A Scalable Content- Addressable Network*, in Proceedings of ACM SIGCOMM 2001.

[14] Ion Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, *Chord: A scalable peer-topeer lookup service for internet applications*, in Proceedings of SIGCOMM 2001.

[15] Antony Rowstron and P. Druschel, *Pastry: Scalable, distributed object location and routing for largescale peer-to-peer systems*, in IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pp.329-350, November 2001.

[16] Ben Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, *Tapestry: A Resilient Global-Scape Overlay for Service Deployment*, IEEE Journal on Selected Areas in Communications, vol. 22, 2004.

[17] Manish Parashar,Manish Agarwal,Steele Arbeeny,Viraj Bhat,Rangini Chowdhury , *Evaluating Security Mechanisms in Peer-to-Peer Applications*, Piscataway, USA.

[18] G. Khataniar and D. Goswami, *Avoidance of churn rate through temporal centralization in Chord*, Peer-to-Peer Networking and Applications, Volume 4, Issue 3, pp 251-258, 2011.

[19] H. Rowaihy, W. Enck, P. McDaniel, and T. La Porta, *Limiting Sybil attacks in structured P2P networks*, in INFOCOM IEEE, pp. 2596 – 2600, 2007.

[20] Dhungel P, Hei X, Ross KW, Saxena N, *The pollution attack in P2P live video streaming: measurement results and defences*, In Proc of the workshop on peer-to-peer streaming and IP-TV (P2P-TV'07). ACM, New York, pp. 323–328, 2007.

[21] J. Liang, R. Kumar, Y. Xi and K. Ross, *Pollution in P2P File Sharing Systems*, In Proc. of INFOCOM'05, 2005.

Dr.Guruprasad Khataniar has completed B.E. from Jorhat Engineering College, M.Tech from Indian Institute of Technology, Delhi and Ph.D from Indian Institute of Technology, Guwahati. His Area of interest is distributed system and P2P systems.

Dr.Hitesh Tahbildar has completed B.E. from Jorhat Engineering College, M.Tech from Indian Institute of Technology, Kharagpur and Ph.D from Gauhati University, Guwahati. His Area of interest is software testing and P2P systems.

Mrs.Prakri Prava Das has completed B.E. from National Institute of Technology, Silchar and M.Tech from Birla Institute of Technology, Pilani. Her Area of interest is distributed system and P2P systems.[1]

[1] *The work is supported by the Research Promotion scheme (RPS) project of AICTE, Govt. of India with ref.no. 8023/RID/RPS-5/(NER)/2011-12.*