Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 32 (2014) 1135 – 1140

Procedia Computer Science

International Workshop on Secure Peer-to-Peer Intelligent Networks & Systems (SPINS-2014)

Sybil Nodes as a Mitigation Strategy against Sybil Attack

Zied Trifa[a,*], Maher Khemakhem[b]

[a]Department of Computer Science, University of Sfax, Tunisia
[b]College of Computing and Information Technology, University of King Abdulaziz, Saudi Arabia

Abstract

Sybil attack is considered one of the most damaged attack that menace structured p2p overlay networks. It's the most sophisticated node active, used for a variety of illicit activities. A key requirement for these activities is the ability of such malicious user to generate a huge number of node identifiers and possibly choose some of them in order to disrupt availability and integrity in such systems. This paper highlights the problem of Sybil attack and presents a Sybil tracking process to deal with such problem.

Selection and Peer-review under responsibility of the Program Chairs.

Keywords: Structured p2p overlay networks; Chord; Sybil Attack; Monitoring; Tracking;

1. Introduction

Structured p2p overlay networks have recently emerged as good candidate infrastructure for building novel large-scale and robust Internet applications in which participating peers share-computing resources as equals. In the past, various systems have been proposed such as Chord, Kademlia, Pastry, and probably more are to come [1].

These overlay networks are distributed systems without any centralized control or hierarchical organization, where the software running at each node is equivalent in functionality. A review of the features of recent p2p applications yields a long list: selection of nearby peers, redundant storage, efficient search/location of data items, data permanence or guarantees, hierarchical naming, trust, authentication, and anonymity. However, these systems are vulnerable to malicious nodes called Sybil.

* Corresponding author. Tel.: +216-25-052-732 ; fax: +216-74-620-469
E-mail address: trifa.zied@gmail.com

Selection and Peer-review under responsibility of the Program Chairs.
doi:10.1016/j.procs.2014.05.544

In such systems, before joining the network, every peer must usually generate a user identifier. This identifier uniquely identify node in the overlay. However, the assignment of IDs is usually not controlled enough since there is no barriers to join the system. Users are free to join or leave the network at any time. This allows malicious users to perform different types of attacks against the overlay such as Sybil attack. In this attack, a single user create multiple fake identities and pretends to be multiple, distinct physical node in the system.

Sybil attack is considered between the most difficult and challenging attacks that plague current structured p2p overlay networks. A malicious node may try to break the routing system, or block access to information by impeding queries, or partition the network.

In this paper, we are interested in tracking and mitigating the use of mass Sybil identities by malicious users. We argue that the main goal of these malicious nodes is to increase the power of the attacker by amassing links to honest users, thus integrating their identifiers into the routing table of other peers. There is therefore the need to explore other mitigation strategies that can be used in combination with disinfection efforts to attenuate the threat of the Sybil nodes.

The rest of the paper is organized as follows. Section II presents the related works. In Section III, we describe our proposed defense called Sybil tracking process. In Section VI, we give detail about the simulation setup and the performance measures we used to assess the effectiveness of our mitigation strategy. Section V concludes.

2. Related Work

Structured p2p systems have shown to be notoriously difficult to protect against Sybil attacks. Various reports have been published that discuss and describes the various proposed defenses. In this section, we present an overview of techniques reported in the literature for making DHT-based systems resistant to Sybil attack.

Haribabu and Hota [2] have proposed a technique based on resource testing. The goal is to attempt to determine if a number of identities possess fewer resources than would be expected if they were independent. It utilizes puzzle methods that exploit communication, storage or computational resource constraints of the participating. In these puzzles, the verifier sends a large random value to every other identity it wants to verify. These identities must then compute the solution within a constrained amount of time. If an entity has more than one identity it will fail to compute the solution within this time.

Bazzi and Konjevod [3] have proposed a solution wherein every identity is issued a geometric certificate by a set of beacons nodes in the network. When a node needs to join the network, it has to present the required certificates from beacons. Dinger and Hartenstein [4] proposed an ID based identity registration procedure called self-registration where an entity calculates its ID by applying a hash function on its IP address and port number. Finally, it registers the IP with ‘r’ nodes in the system.

In SybilGuard [5], authors have proposed a distributed algorithm to limit the entry of Sybil identities into a social network, exploiting the fact that there are very few trust edges between an honest and a Sybil group in a social network. They have designed a protocol in which the verification of a new entry into the network is done by intersection of random routes. SybilInfer [6] offers a decentralized protocol to guard the network against Sybil attacks exploiting the fact that a Sybil attack would interfere with the fast mixing property of social networks.

Yothi and Janakiram [7] have proposed a Sybil monitor associated with every network node to oversee each and every transaction of a node. The given SyMon prevents a Sybil node from targeting honest nodes by moderating the transactions involving the concerned node.

3. Sybil Tracking

Sybil tracking is the process of detecting Sybil attacks performed by malicious nodes, notifying those, and also isolate them. In this section, we mainly concentrate on three steps: Sybil detection, Sybil notification, and Sybil node isolation.

3.1. Sybil Detection

The main idea of the Sybil detection is to introduce monitoring peers within the overlay, the monitors, which are all controlled by one entity, the coordinator. Positioned in a strategic way, the monitors allow us to gain full control over a zone of the overlay. The monitor can supervise the traffic of suspicious nodes and its neighbors. We use the Sybil attack to infiltrate the overlay and observe the communication between peers to get a better understanding of it.

3.1.1. Detection Suspicious nodes

The aim here is to infiltrate the overlay with few number of monitor nodes, which seek to detect suspicious nodes by inserting themselves in the neighbors of honest user. For this, we need to introduce monitors and make them known, such that their presence is reflected in the routing table of other peers.

The coordinator is able to create thousands of monitors on one single physical machine. We have developed an implementation of such coordinator that is able to create detectors and specify for each of them a specific zone to connect to. We divided the overlay into zones to achieve accuracy and obtain a more global view. A zone is specified by x higher order bits (prefix) of the Id space that is common among all peers. We introduce 2^n detectors into the network; the first n bits are different (prefix of each zone) and the following bits are fixed, they are the signatures of our detectors.

To infiltrate the network and detect suspicious peers the monitor M is implemented in the following steps. First, M sends hello message to the neighbor peers in order to poison their routing tables with entries that point our monitors. The peer that receive hello message will add the detector to their routing table. Second, M sends FIND_NODE or FIND_VALUE message to locate some random Ids in the monitored zone. We must ensure that random node Ids or random content ids does not exist in the Id space. The normal behavior is to reply with the nearest nodes to the queried Id. However, the attacker puts its Id in the response and claims he is the owner of the queried Id. By checking who privileges the ownership of those non-existent Ids, we can identify suspicious nodes. Any node coming to us with those content Ids will be marked suspicious. Finally, M gathers the Ids of all suspicious nodes detected and report results to the coordinator.

3.1.2. Local monitoring

This module detects various Sybil attacks against structured p2p overlay networks and verdicts malicious nodes involved in such attacks. Local monitoring starts immediately after the infiltration process and the completion of neighbor discovery process. The infiltrated node monitors the messages going in and out of suspicious nodes and its neighbors.

3.1.2.1. Infiltration process

In order to explore the suspicious node detected by the detection process, we place a monitor peer within. The key idea in this is to make monitors locally to the target ID of suspicious node and its neighbors. This enables us to overhear all the communication. At the start of the infiltration process, the monitor node introduces itself in the overlay in the following two steps:

Step1: The coordinator initiates and places the monitor node next to the target node (suspicious node) in the ID space.

Step2: Neighbor discovery: A neighbor of a node, M, is any node that lies within the transmission range of M. As soon as a monitor node M is infiltrated, it sends a hello message. Any node that receives the message and sends a reply back to M within a predefined time out will be added to its neighbor list.

3.1.2.2. Monitoring process

For a node M to be able to monitor a node S. M must be a neighbor of both S and the neighbors of S. In such case, M monitors all the communication of S and its neighbors. It captures information for each message sent and received from Ns to S in the following two steps:

Step1: When suspicious peer S receives a request from the requester peer, it replies with monitor peer address because according to suspicious peer routing table, M is one of the closest peers to the requested ID.

Step2: When the requester peer learns about the monitor peer, it sends the same request. Thus, the monitor receives a copy of all messages for the address space attributed to the suspicious peer S.

3.1.2.3. Detecting process

In general, the activities underlying a large set of Sybil attacks in structured p2p overlay networks are comprised of the following actions performed by Sybil identities. First, Sybil nodes can drop all the messages received from its neighboring nodes, thereby disrupting the network message routing for lookup process as well as isolating some part of the network. Second, they can delay all the messages by forwarding lookup to incorrect or non-existent peer. Thus it will fail to lookup correct peer by forwarding requests to malicious peers. Finally, they can send false responses to the messages it receives in order to propagate its own propaganda or send malicious files. Thus it will strategically spread the target files in the whole network.

After the description of different kind of Sybil attacks under consideration, a monitor peer can invoke the verification protocol to determine the Sybil nodes:

1. M saves information from each message going over the link X to S in the following form [T; PQ; HD; PD; P1…Pn].
2. M time stamps the information with the deadline d.
3. M overhears every message going out of the receiver S.
4. For all messages that S claims has come from X do:

```
M lookups the information in the database
If (the information is found) then
        M verifies the packet information (HD).
        If (the packet information does not match with the packet information in the database)
                M signals that S is a Sybil node.
          Else (the packet does match)
                M drops that information from the database since the message has been correctly forwarded.
        End If
Else (the information is not found)
        M signals that S is a Sybil node. (Drop)
End If
If (the information stays in the database beyond d)
        M signals that S is a Sybil node. (Delay)
End If
```

Fig. 1. The verification protocol.

3.2. Sybil Notifications and Isolation

Detection process is only the first step towards protecting the structured p2p overlay networks against Sybil attacks. The Sybil notification is used to propagate the notification of detected Sybil to the neighbors and takes the appropriate action to isolate them from the overlay. To achieve the notification and isolation process a monitor node M executes the following actions. First, M sends to each neighbor of S an authenticated alert message in the following form: (IDm, IDs, HM, M's public key). Second, each neighbor of S who receives the alert message achieve this three actions: it verifies the authentication of the alert message, marks S as a Sybil node and stores the message in an alert buffer to prevent other nodes to accept or forward any message from and to S until S will be removed from the overlay. Finally, M proceed to the isolation process. It redirect all messages coming to S to other nodes, drops all messages forwarded by S, and removes S from its neighbor list.

4. Evaluation

For the evaluation, we use the PeerfactSim.Kom [8] simulator to simulate the chord protocol, individually without any protection, and with the approach proposed in the last section. We use a network with a fixed size (N=500 nodes) and fixed rate of Sybil nodes (m=0,2%). We use two scenarios; in the first we use a network without any protection. However, in the second we activate the Sybil detection process. We distribute the nodes over a varied number of zones 0, 2, 4 and 8 to determine the number of monitors to detect and mitigate Sybil nodes. Each node generates and publishes their data using an exponential random distribution each 60 seconds during the first interval rate. Besides, in the second interval time each node performs a lookup each 60 seconds after the stabilization process is over. When a Sybil nodes infiltrate the network and intercept a request, it can drop, delay or send false response to the requester peer. Fig. 2 shows the time event of the attack without Sybil detection process and the attack with Sybil detection process respectively.

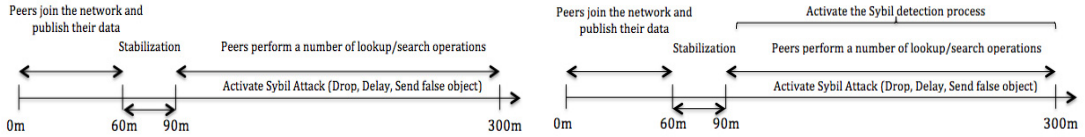

Fig. 2. (a) Attack without Sybil detection process; (b) Attack with Sybil detection process

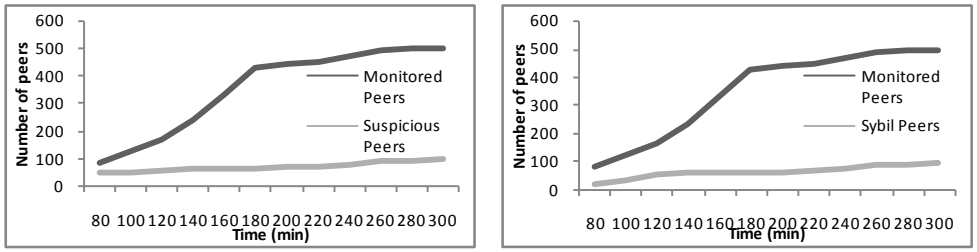

Fig. 3. (a) Evolution of the number of monitored peers VS the evolution of the number of detected suspicious peers; (a) Evolution of the number of monitored peers VS the evolution of the number of detected Sybil peers

Fig. 3.a depicts the evolution of the number of monitored peers vs the evolution of the number of suspicious peers for a 4 zones. However, Figure 3.b depicts the evolution of the number of monitored peers vs the evolution of the number of detected Sybil peers. We can notice that the number of monitored peers grows rapidly over the duration of the experiment, which due to the fact that the number of connected peers to our monitor peers increases with the simulation time. Also, the number of detected suspicious peers and Sybil peers increases with the Sybil tracking process. The high level of participating in the network, make such peers detected and tracked by our tracking process. Fig. 3.b demonstrates that approximately 94% of Sybil nodes are detected.

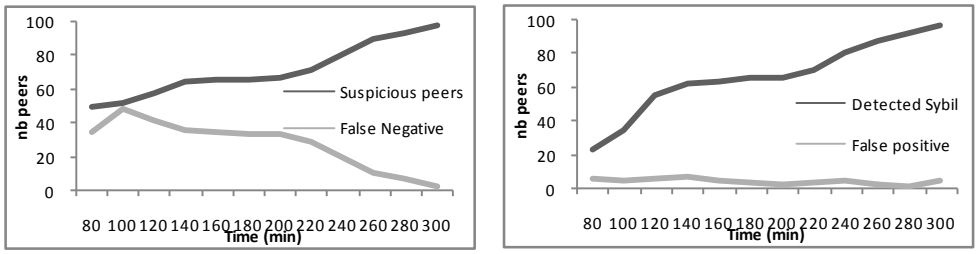

Fig. 4. (a) False Negative; (b) False Positive

We present next on Fig. 4.a the evolution of the number of false negative en relation with the evolution of number of suspicious peers. It refers to a failure to detect Sybil peers that are present on a system. Fig. 4.b demonstrates the evolution of the number of false positive en relation with the evolution of number of detected peers. It occurs when the detector mistakenly flags an honest peer as being infected.

In summary, the validation experiments show that our tracking process captures close to 98% of requests, 96% of operations and approximately 94% of Sybil nodes.

5. Conclusion

In this paper, we have introduced a Sybil attack. We have proved that this attack is one of the most dangerous attacks that plague current structured p2p overlay networks. It is employed to target honest peers and hence subvert the system. It can drop all the messages received from its neighboring node, thereby disrupting the network message routing for lookup process as well as isolating some part of the network. It can delay all the messages by forward ing lookup to an incorrect or non-existent peer. Finally, it can send false responses to the requests it receives to propagate its own propaganda or send malicious file.

Also, we have proposed a Sybil tracking process, which based on three processes to detect and attenuate Sybil nodes. The Sybil detection unit is responsible for detecting any kind of Sybil attack described above. The notification unit handles how to notify these attacks to its neighbors. Finally, the isolation process is responsible for isolating them from the overlay.

References

1. P. Maymounkov and D. Mazieres. “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” Proc. IPTPS, Cambridge, MA, USA, Feb. 2002, pp. 53–65.
2. K. Haribabu, Chittaranjan Hota. Detecting Sybils in Peer-to-Peer File Replication Systems, in Proc. International Conference on Information Security and Digital Forensics, City University, London, Sept 2009, pp. 152-164, Springer, 2009.
3. Bazzi, R. A. AND Konjevod, G. On the Establishment of Distinct Identities in Overlay Networks. In Proc. 24th Symposium on Principles of Distributed Computing. ACM Press, New York, NY, 2007, 312–320.
4. J. Dinger, H. Hartenstein. “Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self-Registration” in: Proc of the First International Conference on Availability, Reliability and Security (ARES 2006), pp. 756 - 763, IEEE Computer Society (2006)
5. H.Yu, M. Kaminsky, P.B. Gibbons, A. Flaxman, “SybilGuard: Defending against Sybil attacks via social networks” in: Proc the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 267 – 278. ACM Press New York 2006.
6. G. Danezis and P. Mittal. SybilInfer: Detecting Sybil Nodes using Social Networks. In Proc, San Diego, CA, Feb 2009.
7. Yothi, B. S., Dharanipragada Janakiram : SyMon: Defending Large Structured P2P Systems Against Sybil Attack , In: Proceedings of International Conference on Peer-to-Peer Computing, Seattle Washington, USA, Sept 9-11, 2009, 21-30
8. K. Graffi: PeerfactSim.KOM – A Peer-to-Peer System Simulator: Experiences and Lessons Learned, In Proc. of IEEE International Conference on Peer-to-Peer Computing, 2011