Securing Structured P2P Overlay Networks

Zied Trifa, Maher Khemakhem
Department of Computer Science, University of Sfax
Sfax, Tunisia
trifa.zied@gmail.com, maher.khemakhem@fsegs.rnu.tn

Abstract—Distributed systems, such as structured P2P overlay networks, are vulnerable to malicious peers. This is due to their flexibility, which allows to any given single participant to appear as many. This will lead, unfortunately, to the possibility of allowing such participant to control much more peers than it can actually do. This is commonly known as Sybil attack. On the other hand, the voluntary arrival and departure of thousands or millions of peers creates a collective effect called Churn. Consequently, an attacker could exploit this way by generating peers which joining and leaving the network very quickly in order to corrupt the best function of the network. Moreover, an attacker could generate a pollution attack leading to corrupt or delete data stored in the system. In such situation, these attacks enable the malicious participant to mediate and monitor most overlay traffics, enabling arbitrarily the denial of service or censorship attacks. Unfortunately, most of existing security solutions are incapable or ineffective to detect malicious peers in dynamic and decentralized systems such as structured P2P systems. Hence, we propose in this paper a Monitoring Framework, which attempts to secure structured P2P networks by monitoring, analyzing, detecting, and mitigating misbehavior peers.

Keywords—structured P2P systems; security; monitoring; mitigation

I. INTRODUCTION

Structured P2P overlay networks are supposed to be able to provide cooperative, stable, and robust mechanisms for storing and retrieving useful arbitrary contents. This is mainly due to the fact that such an overlay network assigns keys to data items (contents) and organizes its peers into a graph that maps each data key to its corresponding peer. Deployments of networks such as Chord [1], CAN [2], Pastry [3], Tapestry [4], and Kademlia [5] can reach very large scales, where millions of users can share contents over global networks.

Unfortunately, the open nature of such systems allows for example to a single malicious user to create multiple fake identities and pretend being multiple and having distinct physical peers in the system. Such attack is known as Sybil attack [6]. In this case, malicious peer can compromise the network by generating and controlling large numbers of fake identities. It can attack several protocols such as distributed storage to defeat replication and fragmentation mechanisms, and routing protocol to defeat routing algorithms. In an overlay network, each peer maintains links to a relatively small set of peers called neighbors. All communication within the overlay occurs on these links. The overlay’s integrity depends on the ability of peers to communicate with each other over a sequence of overlay links. This characteristic makes such systems attractive to thousands or millions of users but at the same time vulnerable to the phenomena of Churn. The random arrival and departure of thousands or millions of peers creates a collective effect called Churn [7]. An attacker could exploit this property by generation peer joining and leaving the network very quickly to corrupt the best function of the network. Also, an attacker could attempt to generate a pollution attack, to corrupt or delete data stored in the system. 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 [8]. In this way, attacker corrupts the content of shared file, making 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 between polluted and unpolluted files.

The increasing complexity of structured P2P overlay networks is behind the big difficulty and challenges faced in securing existing systems. In such systems, trust nodes and high levels of security are often essential. However, the destructive and malicious intent of misbehaving peer is often overlooked, despite being one of the most difficult problems faced in such systems. Most security solutions are incapable or ineffective when detecting peer. This poses more concerns over the security and integrity of future and emerging systems. This paper presents a novel solution called Secure Structured P2P Networks Monitoring Framework which is able in monitoring and mitigating malicious peers in structured P2P systems. This is achieved by using a peer profile following its periodic monitoring by using some key metrics during predefined timeslots given the dynamic nature of such peer. Using the honeypots principles of and controllers, a preset peer status is approached and maintained. This will provide detailed information of peer behavior.

In this study, we are interested in detecting and deterring the use of mass malicious identities by malicious users. We observe that the main goal of these malicious peers is to increase the power of the attacker by amassing links to honest users. There is therefore the need to explore other mitigation strategies, which can be used in combination with disinfection efforts to attenuate the threat of the malicious peers. In this paper, we present our finding pertaining to a feasibility study we conducted on the effectiveness of the monitoring framework as a mitigation strategy against malicious peers. We argue that through a deep analysis of the malicious peers behavior, it is possible to gather additional information about their motives, when they attack the overlay, and their actions after compromising a system. Therefore, it is also possible to learn about new ways of attacking a network in order to prevent such attacks.
The remainder of this paper is organized as follows. In the next section, we provide some background information about the structured P2P overlay networks security concerns and outline how a practical malicious attack can be mounted on these systems. Section III reviews the related works. Section IV describes our proposed solution and its underlying ideas and methodologies. In Section V, we give details about the simulation setup and the performance measures we used to assess the effectiveness of our framework and mitigation strategy and Section VI concludes.

II. BACKGROUND

Structured P2P overlay networks were based on providing efficient search of data items, robust wide area routing architecture, redundant storage, scalability, and fault tolerance. These characteristics can be used to build more complex systems. Several structured P2P overlay networks emerged such as Chord, Pastry, Tapestry, and Kademlia. Moreover, these systems use the Distributed Hash Table (DHT) as a substrate, in which data object location information is placed at the peers with identifiers corresponding to the data object unique key [9].

Early work in structured P2P overlay networks security attacks was based on providing an overview of individual attacks and attempted to remove them. Unfortunately, this task was in most cases unending since more attacks always seemed to appear. In our previous work [10], we have described, detailed and provided a classification of actual attacks threaten such systems. Our goals were more ambitious than previous works; we seek indeed to provide an understandable organization of security attacks that have occurred and help who have the attention to build a safer network. From this classification we found that there is a strong link and correlation between various attacks. To improve the efficiency of these systems we must study several factors such as the assignment of identifiers, the routing table and the communication between peers.

The classes of P2P networks we consider are those based on trust. Peers trust each other to behave appropriately. In such systems, the communication between peers are not random, each peer maintains links to relatively a small set of nodes called neighbors. All communication within the overlay occurs on these links. The overlay integrity depends on the ability of safe peers to communicate with each other. Thus the peer behavior has crucial consequence on the way a system will work, in particular on its efficiency. The term malicious behavior is interpreted as any behavior, message or operation exhibited by a peer, which may negatively impact on the other peers or the system as a whole. This means that monitoring such systems is required, as malicious behavior cannot be effectively controlled. The unusual nature of structured P2P overlay networks such as being dynamic, unpredictable and decentralized means that the security monitoring and analyzing is the most efficient method to ensure the security of a system.

A. Security concerns

Before introducing our monitoring framework of malicious behavior peers, we must understand how they work. We present an overview of the most damages attacks, which provide the source of the others attacks. Please note that this description is a summary of the behavior we observed when studying the malicious behavior of certain peers for a period of time. The results from this section describe several important aspects. We try to generalize our finding as much as possible.

Malicious attacks are based first on the weakness of assigning identities to the users of the structured P2P network. Before joining a network, every participant must generate a user identifier. These identities uniquely identify peers in a network. For this reason, the proper assignment and use of identities are essential for the good operation of the network. This attack is one of the most challenging and difficult problems to solve in structured P2P networks. A misbehaving peer generates a huge number of node identifiers and possibly chooses some of them in order to disrupt availability or integrity in the P2P network. From all these identifiers, the attacker can pick a specific subset, in order to execute other attacks.

However, when a malicious user's comprise a large fraction of the peers in the system, that one user is able to out vote the honest peers in collaborative tasks such as Byzantine failure defenses. Another class of P2P applications exists that even a limited number of malicious peers can wreak havoc such as reputation systems. Besides, in such attack the attackers obtain many identities with much fewer physical entities. By this way they can increase the power or resources of a single user, influence within the target community and mediate the network communication.

Moreover, an individual peer could forward lookup to an incorrect or non-existent peer. Since the malicious peer would be participating in the routing update system, it would appear to be alive, and would not be removed from the routing tables of other peers. Thus it will fail to lookup correct peers by forwarding requests to malicious peers and do not provide any information or provide bugs results.

Since each peer in a lookup system builds its routing table by consulting other peers, a malicious peer could corrupt the routing tables of other peers by sending them incorrect updates. The effect of these updates would be to cause innocent nodes to misdirect queries to inappropriate peers. Also, remove all requests received from it neighboring peers thus disrupting the routing table and information retrieval. Thereby disrupting the network message routing for lookup process as well as isolating some part of the network.

A malicious peer can send requests several times for content from a particular legitimate peer, thus preventing it from rewarding valid requests and flooding the network traffic in that particular route. It can also send false responses to the queries it receives to propagate its own propaganda or send malicious files to honest nodes. It can strategically spread the damaged files across the whole network. Finally, malicious peers as a group can collaborate to disconnect a part of the network from the rest to launch a denial of service attack over chosen nodes or a part of the network.

B. Impact of malicious behavior peers

From these observations, we carry out a probabilistic analysis to demonstrate the impact of malicious peers on the
structured P2P overlay networks. First we introduce terminology and notation used in this analysis:

- The network consists of n peers,
- A peer \( n_i \) is the entity of a structured P2P network. The set \( N = \{n_1, n_2, ..., n_n\} \) with \( |N| = n \) form the number of participated peers,
- The peer identifier of peer \( n_i \) is denoted \( id_i \). The identifiers are selected from an identifier space \( K \),
- The source peer distributes its objects with m peers. It applies a hash function in order to obtain the appropriate key for each object (key = hash (object)),
- The request is distributed to \( r \) peers. Each peer which perceive an object must apply the same hash function already applied previously to obtain the search key (key = hash (object) key = hash (Object)),
- Among the \( n \) peers, there is a proportion \( x \) of peers that are honest;
- The probability of finding the object is denoted \( P(t) \).

Next, we will calculate the probability of finding the object in two cases: in the first case we assume that the network contains only honest peers and in the second case we assume that the network contains a proportion \( x \) of honest peers and the others are considered as malicious. The probability of finding the object (1):

\[
P(t) = 1 - P(\neg t) \quad (1)
\]

In the first case we assume that the network contains only honest peers so the probability of finding the object (2):

\[
P(t) = 1 - \frac{(n - m)! (n - r)!}{n! (n - m - r)!} \quad ; \quad n \geq m + r \quad (2)
\]

In the second case we assume that the network contains a proportion \( x \) of honest peers so the probability of finding the object (3):

\[
P(t) = 1 - \frac{(n - mx)! (n - r)!}{n! (n - mx - r)!} \quad ; \quad n \geq m + r \quad (3)
\]

To evaluate the performance of our probabilistic analysis we used a network of 1000 peers.

We analyze the impact of malicious peers on the performance of the system, more specifically on the probability of finding an object. Fig.1 represents five curves; each curve represents the probability of finding an object in function of the number of physical peers, which represent the number of replication object and the percentage of honest peers. This figure shows that the probability of finding an object decreases with the percentage of malicious peers, whereas this probability increases with the number of replication in the system.

Fig.1 shows also that the probability of finding an object increases quickly when m and n increases. For example, the red curve shows that the probability moves from 0,2 when m=n=15 to reach 0,9 when m=r=75 nodes. However, the probability of finding an object decreases quickly when the percentage of malicious peers increase. For example, the probability moves from 0,3 when the network does not contain malicious peers to reach 0,07 when the network contains 80 % of malicious peers.

III. RELATED WORK

Various papers have addressed monitoring and measurement of P2P systems. Current research efforts are predominantly focused on monitoring and measurement of P2P overlays and protocols, thus ignoring peer activity. There have been a lot of works done in this field. We rapidly describe them in this section.

J.L. Guillaume and al [12] suggested using a modified server on eDonkey system to passively collect data. The modified server may log the queries it manages, capturing the traffic and decoding it. D. Zeinalipour-yazti and T. Foliis [13] proposed a modified client on Gnutella network. The modified client may observe the traffic going through it. The proposed solution monitors the network by being attached to a large number of peers. S. B. Handurukande and al [14] designed a client that sends queries in the system and records the obtained answers. This solution has been done in Napster and eDonkey with success. M. Steiner and al [15], developed a traffic-monitoring tool for Kad, called Mistral, which captures traffic by placing a large number of peers into a zone. In the study they place around 65,000 monitoring peers into an 8-bit zone. Memon and al [16], developed Monta, a traffic monitoring tool for Kad, they add a smaller number of monitors to cover the same area of zones. The proposed solution does not significantly disrupt the overlay structure of the existing peers, it provides richer data compared to Mistral. X. Jin and S. H. G. Chan [17] proposed a method based on collaboration. Each peer is monitored by its neighbors and observations are collected to compute a reputation score for the particular node. A. Visan and all [18] suggested the use of decentralized trust management to rank participant and moderate interactions between them. Graffi and all [19] proposed a monitoring and management framework, the objective is to ensure quality not security. The proposed solution use a statistical representation of the entire network, if
it falls below a defined quality interval, a process is initiated to automatically adapt the system configuration.

IV. Secure Structured P2P Overlay Networks Monitoring Framework

In this section, we describe and detail a new framework to monitor, analyze, detect, and mitigate malicious peers in structured P2P overlay networks. After an overview of the different vulnerabilities, we try to solve these malicious peers by detecting their malicious behavior. We model our solution on a structured network, which employs a Kademlia protocol.

Our vision of monitoring and mitigation of structured P2P systems is based on the idea that it is possible to detect misbehavior peer by monitoring for abnormalities in operations used for common actions in a system and changes in system security. Therefore, knowledge of these operations and security changes would provide baseline of what could be considered as normal behavior. Hence, we used first a principle of honeypots and controllers, injected in such system, which involves the monitoring and logging certain key metrics at predefined intervals for a period of time. This will provide a profile of peer behavior, which contains preset intervals for a set of metrics. The same process will maintain this profile regularly.

Second, our work aims to monitor suspicious behavior in the system level. We set up a honeypot, which had two purposes. The first purpose is identifying as many as possible suspicious peer and the second is detecting the current status of the metrics of suspicious peer in a running system. In case that an analysis process detects a deviation from the normal behavior threshold, a mitigation process is initiated automatically to adapt the system and reject these malicious peers. The idea is depicted in Fig.2.

![Fig. 2. Secure Structured P2P Networks Monitoring Framework.](image)

A. Monitoring at system level

A common approach for capturing traffic in a distributed hash table is to place randomly a few instrumented peers within the network. The goal of the monitoring is to gather a wide set of metrics on the peer behavior. The metrics describes a measure of the under test peer. For example: the number of initiated queries, the number of initiated operations, number of messages received, the number of messages submitted, etc.

In structured P2P systems, the communication between peers are not random, each peer maintains link to a relatively small set of peers called neighbors. All communication within the overlay occurs on these links. The overlay integrity depends on the ability of honest peers to communicate with each other over a sequence of overlay links and messages. Likewise, the peer behaviors has crucial consequence on the way a system will work, in particular on its efficiency. We propose here a study of peer’s behavior. To achieve this, we analyze a large trace of messages and queries in the overlay, which gives an insight on the proprieties and peer behaviors. Our measurement infrastructure consists in a set of fake peer called Honeypots doted with specific controllers. These honeypots are connected to different zones in the overlay and controlled by a coordinator.

The first function of the coordinator is to launch the honeypots. It specifies for each of them a zone to connect to. Several strategies make sense for assigning honeypots to zones. One may typically choose a different zones for each honeypot, in order to obtain a more global view. A zone is specified by x higher order bits of the ID space that is common among all peers in that zone. Therefore, by monitoring all peers in the entire zone, we capture all operations and messages for that zone.

The second function of the coordinator is to specify the set of operations and messages to generate. Each honeypot constructs a log file of all queries and messages it receives. We use a trace of the queries made to each honeypot injected in the system. We focus on the exchanges between peers in different levels of the system (Network, Transport, DHT, Application). Therefore, we will mainly consider messages and operations. We record the answers to such messages in the following form: [T | MT | Prot | PQ (IP@, UD | Node ID) | HD | P1, P2,...,Pn]. Where T is a timestamp represented in seconds from the beginning of the trace, MT is the message type, Prot represents the protocol used (TCP, http, UDP), PQ is the peer which has emitted the query, HD is the hash code of the data queried by PQ and P1…Pn is a list of peers which has registered the data.

Finally, the coordinator gathers periodically the data collected by honeypots and does basic management based on statistical representation to create a profile of peer behavior. This process is repeated periodically to maintain up to date the profile of peer since the distributed nature and the large size of structured P2P systems.

B. Honeypot detector

To detect suspicious behavior in structured P2P systems, we focus on Kademlia system and its security problems. We choose Kademlia systems because they are the most cited and deployed as structured P2P overlay. These systems have long since been known to have inherent security issues. Effective lack of protection on these systems implies that is very easy for an attacker to monitor large fraction of traffic on the system or even hijack the system. We base these claims on three forms of attacks: Sybil attack, Churn attack, and Pollution attack. We argue in our previous work [10] that Sybil attacks are one of the most prevalent and practical attacks against these systems. Therefore, we will focus on these attacks to understand the methodology of these malicious peers to mount such practical attacks. We studied and analyzed the different messages and
Peer misbehavior is very difficult to detect, particularly in dynamic environment. Therefore, there is not always a defined behavior pattern or signature to observe and there is no centralized administration. This challenging environment means that the only feasible method for detecting misbehavior is with capable security monitoring. As mentioned earlier, large number of monitors, while necessary to collect a complete view of the system, to create a profile of peer behavior.

In order to investigate the suspicious peers detected by our honeypots detector, we place a few instrumented peers within these suspicious peers. The key idea in this is to make monitors near to the suspicious IDs. The monitoring peer of the suspicious peer captures these messages in the following three steps. First, The coordinator initiates the monitoring peer and places near to the suspicious peer in the ID space by setting its ID, as the ID of monitoring peer equals ID of suspicious peer XOR 1 since Kademlia uses XOR based metric topology to enhance the robustness of the system. The distance between any two peers, object, or a peer and object is defined as the output of the XOR function between their IDs. After that, the peer monitoring add itself to the suspicious peer routing table by exchanging PING messages. Second, when suspicious peer receives a request from the requester peer, with a destination in the ID space, suspicious peer replies with peer monitoring address because according to the suspicious peer routing table, peer monitoring is one of the nearest peer to the requested ID. Finally, when the requester peer learn about the peer monitoring, it sends the same request to this peer. Thus, peer monitoring receives a copy of all queries for the address space attributed to suspicious peer. The peer monitoring constructs a log file of all queries and messages it receives, we use the same trace used in the first functionality. Finally, the monitoring peer report the result to the coordinator.

D. Analysis process

The monitoring process is a constantly running process, first to maintain a peer profile up to date and second to detect malicious behavior based on preset metrics gained by the first process. This module contain the novel decision algorithm which compares the profile of peer behavior, obtained by the monitoring module at system level, and the statistical information obtained by the monitoring process at peer level. Each metric that is monitored on the peer has a different influence on the behavior of others. It is often possible to identify problem by examining metrics that are highly influential on each other. As our decision algorithm takes all metrics into account, it is important to take into account the relationship between them. Weightings are numerical representation of how influential metrics are on each other and how reliable they would be as an indicator to a problem. However, the current behavior metrics need to be analyzed and compared to the profile of peer behavior. If any deviations are detected, then these are reported to the mitigation process.

E. Mitigation process

In this process, the monitoring information is analyzed in terms of matching the profile of peer behavior. In case where the analysis step detects a violation of the interval behavior bounds, a mitigation process is initiated in order to determine
how to reject the malicious peers or how to adjust the overlay of structured P2P system under concern.

The mitigation of malicious peers must attack the control infrastructure to be effective, i.e. the communication method. We argue that structured P2P systems style communication has weaknesses, which can be generally exploited. Structured P2P systems are distributed hash table based. They scale to large networks because each peer stores routing table that scale linearly with the network size. Using this limited state, peers cooperatively map keys to physical network peer using a multihop lookup mechanism called key based routing. However, in such system, there is no way to send information directly. Note that since we consider DHT systems, any member and object is identified by an ID. This is the key idea of our mitigation strategy. We can use our injected honeypots to try to either inject commands into the malicious peers detected, or disrupt the communication channel. In general, disruption is possible since we can flood the malicious peers with a large number of requests. However, in order to inject commands, we need to understand the communication process in details and then corrupt the communication channel.

V. EVALUATION OF THE FRAMEWORK

For the evaluation of the different component of our framework we performed several experiments in which we simulated the functioning of the Kademlia protocol and the behavior of malicious peers. All experiments are done in the P2P simulator PeerfactSim.Kom [11]. In the simulator, we have divided the time into intervals. We considered a network with 2500 nodes and 5 simulated hours. In the first time interval \([0, 60m]\), 2000 peers join the network. Besides joining, peers start to execute some overlay operations. They publish their data using FIND_NODE and STORE messages. After that, they perform a number of lookup, search operations starting at 90m. Once the joining process is done, the coordinator starts the injection process of eight honeypots with the different controllers in the overlay. Each honeypot perform a set of operations, such as publishing fake objects or searching random objects, during the interval \([90m..300m]\). Each of them construct a log file of all operations and messages it receives in the following from: \([T \mid MT \mid Prot \mid PQ \mid HD \mid P1, P2,...,Pn]\). Also we added 500 peers during the time interval \([90m..180m]\) to simulate the network evolution. After that we create a significant number of malicious peer’s to simulate the malicious behavior peers. During this process the group of honeypots record the answers and the exchanges between peers in different level of the system. In the second experiment, we based our simulation on the first one. However, we activated our honeypot detector, the analysis and the mitigation process after the bootstrapping process is done.

In Fig.3 we show the number of peers in the network and the corresponding number of monitored peers. As we can notice, especially at \(t=210m\), the monitoring is good. However, the monitoring has also some variations, especially in the first hour of the monitoring process, the number of peers observed is very low. This is caused by the initialization phase of this measurement as described above.

![Fig. 3. Evolution of the number of peers observed during the measurement.](image)

We portray in Fig.4 the estimated number of suspicious peers and the evolution of the number of suspicious peers detected after the activation of the honeypot detector. However, we display in Fig.5 the evolution of the number of suspicious peers tracked after the activation of the analysis and the mitigation process.

![Fig. 4. Evolution of the Estimated number of suspicious peers VS the evolution of the number of suspicious peers detected during the measurement.](image)

![Fig. 5. Evolution of the detected number of suspicious peers VS the evolution of the number of suspicious peers tracked during the measurement.](image)

As we can observe in Fig.4, the detection of suspicious peers has a lot of variations; this is due to the honeypots detectors and the variation of the malicious behavior peers. After the creation of the peer profile, the honeypots detectors are not well introduced in the different zones of the system. Besides, the dynamic nature of malicious peers that caused a high changing in the structure of the network. However, in Fig. 5, the mitigation of detected suspicious peers is accurate since we based our mitigation process on the identity of suspicious peer.
VI. CONCLUSION

The lack of a useful and powerful tool able to secure properly structured P2P networks constitutes the main concern and the basic of this paper. Consequently, we proposed a novel framework for the monitoring leading to the mitigation of malicious peers in structured P2P networks. Our proposed framework aims at providing to the interested community a way to make a good profile for every peer in a given systems in order to understand its behavior which can be malicious. In such a case, our framework gives also the way to mitigate it and consequently makes safer the functioning of such networks.

For the future we expect to test our solution on some real overlays such as Chord, CAN, Pastry, Tapestry and try to refine both the framework and the corresponding features in order to go further towards reaching our expectation.

REFERENCES