Worm Containment in Gnutella Network with Heterogeneity

Amir-Reza Niakan-Lahiji and Babak Sadeghiyan

Abstract—**Attentions to security on P2P networks have recently been increased, especially with respect to worm epidemics. Most of countermeasures that are proposed to prevent worm propagation are ineffective in P2P overlay networks. In this paper, we use heterogeneity among peers to prevent worm propagation in Gnutella network. Our solution is based on the observation that in P2P networks, there are different types of peers which have different vulnerabilities due to their different platforms. Therefore, a worm, which normally uses a specific vulnerability for propagation, can only infect a specific type of peers in the network. We modify Gnutella joining procedure to restrict worm infection to a small fraction of the network.**

Index Terms—**Heterogeneity, P2P active worm, worm propagation, Gnutella**

I. Introduction

Since the propagation of Morris worm in 1989, worm propagation is considered as one of the most important security issues in the Internet and many computer scientists try to study their behavior. According to [1], worm is "a program that self-propagates across a network exploiting security or policy flaws in widely-used services". There are different types of worms but active worms, which automatically choose their victims and propagate through the network, are mostly concerned. Staniford et al. in [2] showed that active worms can propagate through the Internet within seconds. Most of active worms in Internet are scanning worms[2]. This means once the worm has infected a host; it tries to find the next host by choosing a random IP address from IP address space. The rest of active worms are non-scanning worms in which worms collect local information from infected hosts and based on this information try to find the next target. For example, in Gnutella P2P network, each peer in the network has a list of neighbors. A non-scanning worm can use this list in infected peer and try to infect the neighbors.

Non-scanning worms are more serious security threat than scanning worms, since they have the following two features: 1) they don't waste time to attack non-existence systems, so they are faster than scanning worms 2) most prevention and detection mechanisms [3]-[5] for scanning worms like honeytrap or honeypot are not effective for non-scanning worms. From now on in this paper, by worm we mean active worm.

Manuscript received February 20, 2012; revised April 22, 2012.

The authors are with the Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran (e-mail: basadegh@aut.ac.ir; amirreza_nl@aut.ac.ir).

Many scientists in different disciplines, e.g. computational biology, computer science, tried to model epidemic propagations on different kind of populations. To model the propagation of epidemics, they pointed out the most important factors which have direct effect on the epidemics. Kermack and Mckendrick on 1927 had presented the first mathematical model for spreading epidemics. Their model show that epidemic characteristics like contact rate has crucial impact on epidemic propagation. It has already shown by numerous authors that their assumption about uniformly mixed population is not realistic and contact graph, which individuals in population constructs, has significant impact on propagation of epidemics among them. In [6] authors tried to show how different network topology has impact on epidemics. It has shown that many networks such as social networks, collaboration networks, and, World Wide Web are complex network. In [7]-[8] authors mention the effect of complex network on epidemic spreading. In [9] authors mention heterogeneity among population also impact on slowing down epidemic propagation. We can easily classify these factors in two categories: 1) Factors which are related to characteristics of epidemics like contact rate 2) Factors which are related to network structures like degree distribution. By affecting on these factors, one can control the propagation of infection.

Recent advancement in P2P networks and emergence of P2P networks like Gnutella attract a lot of users such that more than 50% of all internet traffic is belonged to P2P traffic[10]. Furthermore, scanning worm's strategy seems to be ineffective in next generation networks based on IPv6 due to large IPv6 addressing space. Hence, it seems most worms in the future would turn to non-scanning worms and try to propagate on networks like P2P networks.

Some authors have indicated the potential problems which can be raised by P2P worms[9], [11]. Such worms can infect a large population of computers in Internet just in a few minutes. They use topological information maintained in peers and choose the next victim based on this information.

In this paper, we show how heterogeneity among peers is important in propagation of P2P active worm and use it to extend Gnutella protocol so it can resist completely against propagation of active P2P worms. To achieve this goal, we first show that heterogeneity among peers in a P2P network is a common phenomenon. To show this, we had crawled the Gnutella network, which is one of the most popular P2P network, and collected information about peers in the network. We, then, launched a P2P worm simulation on the captured network. Then, we show the simulation result. After showing that our observations are valid, we use them to improve the current Gnutella protocol in the face of

active P2P worms. Then, we examine the proposed improvement through simulation.

The rest of the paper is organized as follows: we first briefly introduce different type of P2P overlay networks. After that, we briefly discuss about P2P worms and epidemic models which are proposed for them. Then we present our observation on Gnutella network and propose an extension for enhancing the network reaction in face of worm propagation.

II. P2P Networks

Although based on strictest definitions of peer-to-peer networks are considered to be totally decentralized, in practice we have seen that this property are not always hold and encounter with P2P networks with different degree of centralization. In [12] authors categorized P2P networks based on degree of centralization into three classes: 1) Purely Decentralized Architecture in which all node in the network has the same functionality 2) Partially Centralized Architectures in which there are some supernodes in the network that assume to have more important role than regular nodes. 3) Hybrid Decentralized Architectures. In these systems, there is a central server facilitating the interaction between peers by maintaining directories of metadata, describing the shared files stored by the peer nodes.

We can also categorize P2P overlay network based on network structure. By structure we mean whether network is non-deterministically grows or not. In other words, whether there is a rule on where new node and contents add to the network or not. In this regard, P2P networks can be categorized into structured and unstructured networks. Many popular P2P networks like Kazza, Morpheus, Gnutella, and BitTorrent which are used for file-sharing are unstructured P2P networks. Structured P2P networks like CAN, Tapestry, Chord, and Pastry have emerged in attempt to address scalability issues that unstructured networks were faced [12], but most of them are not used in practice.

III. Gnutella Network

Gnutella is one of the most popular P2P networks which is used by many peoples in the past decade and attracted many researchers who were interested in P2P networks. Gnutella is unstructured P2P networks. In early years, Gnutella was purely decentralized P2P network which means all nodes were connected to each other randomly and had the same functionality. After a while, for scalability issues, peers divided to ultrapeers and leaves (in Gnutella protocol version 0.6) so current Gnutella network is Partially Centralized P2P network.

When a peer wants to join a Gnutella network, it must find, at least, one peer in the network. This process is called Bootstrapping. Various methods have been used for this, including a pre-existing address list of possibly working nodes shipped with the software, using updated web caches of known nodes (called Gnutella Web Caches) [13], UDP host caches[14] and, rarely, even IRC. However, the most common method is to use Gnutella Web Cache servers.

In using Gnutella Web Caches, new peer query already known servers via http protocol and ask for a list of peers in the network. Then try to open TCP connections to these peers. If they are alive then the peer do handshake with them to join Gnutella overlay network. The exact procedure of handshake can be found in [15].

In the past, ping/pong messages are used to crawl the Gnutella network (for example in [16]). But nowadays for scalability reasons, many Gnutella peer implementations use mechanisms like pong caching which limit the number of pong messages that a peer node can get. This method would not anymore be accurate for crawling the network. However, today most Gnutella peer software reply to crawl header in a connect message in handshake phase. When a peer receive a connect message which has crawl header, the peer replies a connect message which contains all neighbors of that peer. One may use this feature and extract topology of the network.

IV. Heterogeneity in P2P networks

One of the factors in network that affect worm propagation is heterogeneity among peers in the network. In [9], the impact of heterogeneity in peers' software on worm propagation speed is shown through simulation.

We considered that heterogeneity in current open-source P2P networks, which their protocols are released under open-source licenses, is usual phenomenon and there exist many different type of peers' software in such networks. Even, in proprietary P2P networks, different version of peers exists. To examine the correctness of above statement, we crawled the Gnutella network.

To crawl the Gnutella network, we wrote a new crawler which use crawl header in Gnutella protocol. It crawled ultrapeers in Gnutella network for 12 hours in October 2010 and collected information about 9000 peers in the network. In this observation, about 53 percent of peers was LimeWire v5.*, about 23 percent of them was

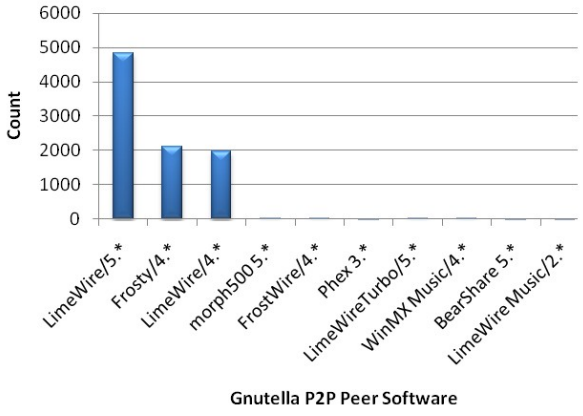

Fig. 1. Different type of gnutella software. The figure shows frequency of each peer's software in Gnutella network when we crawl the network during 12 hours.

Frosty v4.* and about 22 percent of them was LimeWire/4.*.

So usually in P2P networks we have different kind of peers. Heterogeneity among peers in a network slows down worm propagation in two ways. Firstly, it affects on

infection rate of the worm. Secondly, it may cause disjunction between vulnerable peers. In other words, it may cause some vulnerable peers are not reachable from the other vulnerable peers.

We examine these justifications in the crawled network with 9026 peers. Let assume we have a worm that can infect LimeWire/5.*, which has the most population in the network. At most, from each 4833 peers with LimeWire/5.* we can only infect 3244 peers. In other word, the biggest connected component with the same type has 3244 peers in itself.

Most models which have been proposed do not consider heterogeneity in population type and overestimate propagation speed in P2P networks. We suppose these models consider network structure, in this situation they, mostly, estimate in 20th cycle, all 3244 peers are infected by the worm.

We launched two types of worm propagation simulations on the crawled network. In the first type of simulation, we didn't consider peers' software type and in the second type of simulation we considered peer's software type. We assumed, worm infection take one cycle. In the first type of simulation, the worm infected 9026 peers in average on 62 cycles. In the second type of simulation, the worm could infect only 3244 from 4833 susceptible peers in about 59 cycles and ended. In real world, our condition is like the second simulation so worm propagation speed is lower than what most P2P worm propagation models state and can infect only a fraction of P2P networks like Gnutella.

In Fig. 2, we see that if the population is composed of different types, worm propagation speed becomes slower. Because, some of worm attempts hit to the heterogeneous peers, which are immune.

V. Modifying Gnutella Network

Our approach for preventing P2P worm propagation is to use heterogeneity among P2P nodes. Hence, we modify the bootstrapping procedure in Gnutella, in order to take advantage of peers' heterogeneity in the network.

To join a network, a new peer mostly uses a Gnutella Web Cache server and queries a list of peers which are already in the network. When a peer wants to query the Gnutella Web Cache, it sends its type to the server and server return a list of peers which is selected by a random function with a uniform distribution. To query the server, the client use "get" method in http protocol and sends its information via query string parameters. These parameters includes "client", "version", and etc. "client" parameter is filled by software type of client. By client type, we mean "client" and "version" parameters in this request.

Instead of using a random function with a uniform distribution, we use our weighted random selection function which is based on the alias method algorithm in [17]. Our Gnutella Web Cache server assigns weight to each peers in its list based on the type of requested peer. If a peer in the list has the same type as the requested peer, server assigns a weight α for that peer, otherwise it assigns β. Clearly, α must be lower than β, otherwise, with more probability, peers with the same type are returned. We set β and α to 1 and 0.1 respectively. If α / β get smaller, then with higher probability peers with the different type are chosen. However, α / β must not tend to zero. If it happens and one type is absolute majority in the peer population, then peers in minority population get high connectivity request and maybe get overloaded.

Our scheme works as follow: suppose a new peer x which has type Y, wants to get a list of peers from our Gnutella Web Cache server. Our server assigns 0.1 to each Y-typed peer in our server and 1 to others. Then the server uses the weighted random selection function, instead of conventional random selection function, and selects ten peers at random from the list of weighted peers. We choose ten because commonly Gnutella Cache Servers return in average ten peers for each query. In this way, whenever a new peer wants to get a list of peers in the Gnutella network, our server returns other types with more probability.

With our proposed approach, any peer is mostly surrounded with peers which have different type. In this way the contact graph of peers with the same type become like small distinct islands. So if a worm infects a peer in one island, it can only infect a few susceptible peers and remain captive in that area.

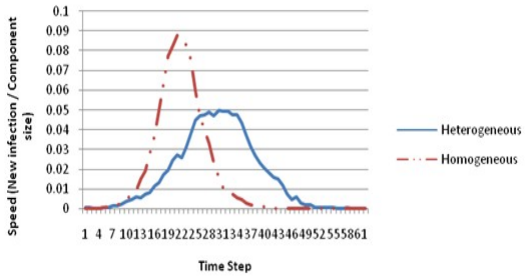

Fig. 2. Propagation speed overestimation without considering peer's type in the population.

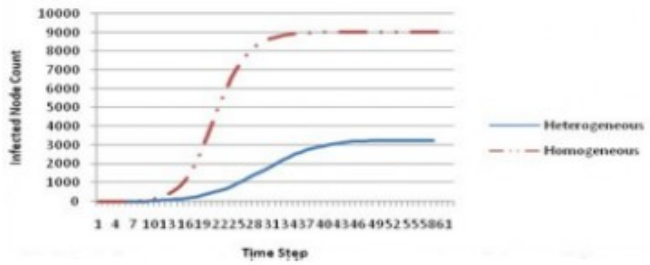

Fig. 3. Infection size overestimation without considering peer's type in the population

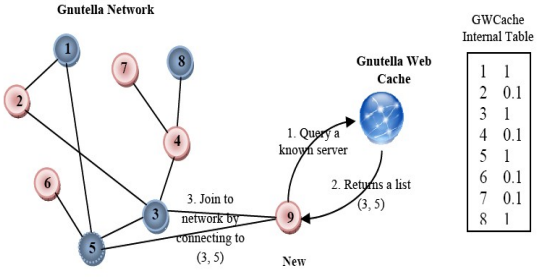

Fig. 4. Joining the network. whenever a new node wants to join a network, it queries a known Gnutella Web Cache (GWCache) server. In our scheme, server select a list of peers based on theirs weights. It returns this list to the node. Then the node tries to connect to the nodes in the list.

VI. Simulation

To evaluate our new build-in worm containment mechanism in Gnutella, we develop a simulator which simulates the formation of the modified Gnutella network. It has shown that Gnutella network has a power-law property[18]. Power-law network has the following property: in these type of networks the probability of a k degree node is proportional to 1/k^λ, where λ is usually 2 to 3. Hence, to examine our worn containment strategy in the new network, we develop a graph generator based on Barabási–Albert model [19] which resembles the formation of our new Gnutella network.

Each node in the generated graph has a type which is determined when a node created based on a probability distribution (with probability 0.4, node is vulnerable otherwise it is immune). In the initial stage, there are **m** nodes which are fully connected with each other. When a new node wants to join the graph, it attaches to at most **n** from **N** nodes in the network based on a preference function (n<=N). In this simulator the preference function is our weighted random selection function. We set m and n to 3 and 10 respectively. Based on our experiment, in this setting, the degree distribution of the generated graph is very similar to the obtained data from crawled Gnutella network which is stated in section 4.

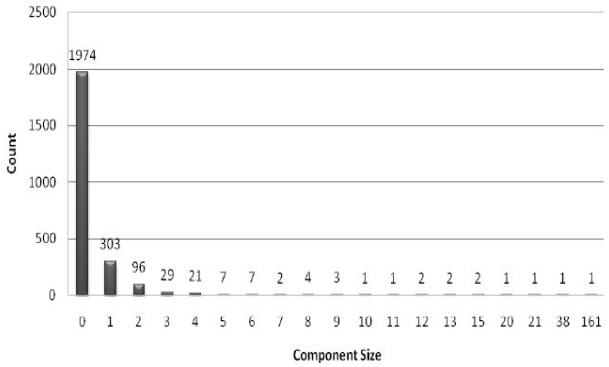

Fig. 5. Compartmentalization of the vulnerable population in the network: a generated power-law graph with 9027 nodes which 3615 of them are vulnerable. In this network, each new node tried to connect to at most 10 nodes. Component size shows: from a node how many connected nodes with the same type are reachable Therefore, zero component size means isolated node, which are surrounded only by different types of nodes.

In Fig. 5, you can see the frequency of component size for vulnerable population in a generated network with 9027 nodes. In this network about 0.4 of all nodes are vulnerable to an epidemic.

As one can see in this figure, most vulnerable nodes are isolated in this network and the biggest vulnerable component has only 161 nodes. With the chance of 0.045, one of the nodes in the biggest component initially gets infected and in this situation the infection can only infect 161 out of 3615 vulnerable nodes. Therefore, an infectious can only infect a small island in this network.

VII. Conclusion

In this paper, we showed that heterogeneity is a common phenomenon in open source P2P networks such as Gnutella. In fact, heterogeneity is a very common phenomenon not only in open source P2P networks, but also in any open source networks. Heterogeneity in any network can cause reduction in non-scanning worm propagation speed and final size of infected node population. We showed reduction in final infection size in Gnutella network through simulation. Thereafter, we used it to devise a self-worm containment capability in Gnutella network. Our proposed mechanism does not incur any performance penalty because we did not perform any kind of extra calculation in any nodes in the network. We only change the node selection procedure in Gnutella Web Cache servers to immune network against active P2P worms. Hence, our approach needs not a lot of effort to be used in practice. We believe that by using heterogeneity in any network, one can isolate any unknown non-scanning worms in a small portion of that network.

Acknowledgment

This paper has been supported by Research Center of ICT – ITRC.

References

[1] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A taxonomy of computer worms," in *Proc. of the 2003 ACM workshop on Rapid malcode*, pp. 11-18. 2003.

[2] S. Staniford, V. Paxson, and N. Weaver, "How to own the internet in your spare time," in *Proc. of the Eleventh USENIX Security Symposium,* 2002.

[3] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levine, and H. Owen, "HoneyStat: Local Worm Detection Using Honeypots," in *Recent Advances in Intrusion Detection*. vol. 3224, E. Jonsson*, et al*, Eds. ed: Springer Berlin / Heidelberg, pp. 39-58, 2004.

[4] N. Weaver, S. Staniford, and V. Paxson, "Very fast containment of scanning worms," in *Proc. of the 13th conference on USENIX Security Symposium* - Vol. 13, 2004.

[5] D Whyte, P. C. V. Oorschot, and E. Kranakis, "Detecting intra-enterprise scanning worms based on address resolution," in *Proc. of Computer Security Applications Conference*, pp. 10-380. 2005.

[6] A Ganesh, L Massoulié, and D Towsley, "The effect of network topology on the spread of epidemics," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1455-1466. 2005.

[7] Y. Gang, Z. Tao, W. Jie, F. Zhong-Qian, and W. Bing-Hong, "Epidemic spread in weighted scale-free networks," *Chinese Physics Letters,* vol. 22, pp. 510, 2005.

[8] T Zhou, Z. Q Fu, and B. H Wang, "Epidemic dynamics on complex networks," *Progress in NaturalScience,* vol. 16, pp. 452-457, 2006.

[9] L. Zhou, L. Zhang, F. McSherry, N. Immorlica, M. Costa, and S. Chien, "A first look at peer-to-peer worms: Threats and defenses," *Peer-to-Peer Systems IV,* pp. 24-35, 2005.

[10] H. Schulze and K. Mochalski. (2009, 12 November 2010). Internet *Study 2008/2009*. Available: http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009

[11] N. Khiat, Y. Charlinet, and N. Agoulmine, "The emerging threat of peer-to-peer worms," in *IEEE / IST Workshop on "Monitoring, Attack Detection and Mitigation,"* Germany, 2006.

[12] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys (CSUR),* vol. 36, pp. 335-371, 2004.

[13] P. Karbhari, M. Ammar, A. Dhamdhere, H. Raj, G. F Riley, and E Zegura, "Bootstrapping in Gnutella: a measurement study," *Passive and Active Network Measurement,* pp. 22-32, 2004.

[14] S. Langkemper, "Bootstrapping Gnutella Using UDP Host Caches," 2005.

[15] T. Klingberg and R. Manfredi. (2002, 10 December 2010). *Gnutella 0.6*. Available: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[16] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing Journal,* vol. 6, 2002.

[17] R. A. Kronmal and A. V. Peterson Jr, "On the alias method for generating random variables from a discrete distribution," *American Statistician,* vol. 33, pp. 214-218, 1979.
[18] M. Ripeanu and I. Foster, "Peer-to-peer architecture case study: Gnutella network," in *Proc. of First International Conference on Peer-to-Peer Computing*, 2001.
[19] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science,* vol. 286, pp. 509, 1999.

Amir-Reza Niakan-Lahiji was born in Tehran, Iran in 1986. He received his B.Sc. in 2008 in Software Engineering and M.Sc. in 2011 in Information Technology from Amirkabir University of Technology, Tehran, Iran.
He is currently works in APA laboratory in Amirkabir University as a senior researcher.
His research area of interest includes Network Security, Malware and Honeynet.

Babak Sadeghiyan, Ph.D., was born in 1961. He received his B.Sc. in 1985 in Electrical (Electronics) Engineering from Isfahan University of Technology, and his M.Sc. in 1989 in Electronics Engineering from Amirkabir University of Technology, Tehran, Iran.
He received his Ph.D. in 1993, in Computer Science from University College, University of New South Wales, Australia, on the design of secure hash functions. Then he joined the Computer Engineering and IT Department of Amirkabir University of Technology in 1993, where he is still continuing his academic activities, and is currently an associate professor of the department.
His research area of interest includes all aspects of Cryptology and Information Security, more specifically; he has contributed on the design and analysis cryptographic algorithms, cryptographic protocols, hardware implementation and cryptanalysis of cipher systems, network security and intrusion detection systems.
He has pioneered the academic study of intrusion detection systems in Iranian universities since 1996. He is an author to one book, 14 research journal papers and to more than 100 conference papers.