Passive Worm and Malware Detection in Peer-to-Peer Networks

Sahar Fahimian, Amirvala Movahed
Department of Information Technology
Sharif University of Technology
International Campus
Kish Island, Iran
{*fahimian,movahed*}*@ieee.org*

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
&
IPM School of Computer Science
Tehran, Iran
kharrazi@sharif.edu

***Abstract*—Today P2P networks are responsible for a large amount of traffic on the Internet, as many Internet users employ such networks for content distribution. At the same time, P2P networks are vulnerable to security threats such as Internet worms and facilitate their propagation. Internet worms and more generally malware are a major concern to the network security community. There are many different type of worms in the wild, mostly categorized based on how they find and infect their new victims (i.e. active, passive, etc.). In this paper, we investigate a new approach for detecting passive worms and malware in P2P networks based on the popularity of files in the network. As part of our investigation, we crawl the Gnutella P2P network over a 12 day period collecting file names and file popularity statistics. We are then able to extract the highly popular files and identify worm/malware files within them with high accuracy.**

***Keywords*-Peer-to-Peer, Worm, Detection**

I. Introduction

Todays peer-to-peer (P2P) systems transmit a major part of traffic on the Internet. Most of P2P users are using such networks for content distribution as it provides benefits such as scalability, reliability, fault tolerance, while using resources efficiently. In general, P2P networks are categorized as either structured or unstructured. In this paper we mainly focus on the unstructured file sharing P2P networks because of their wide usage and hence potential worm infection vulnerability. Peers in an unstructured P2P networks are laid out like a random graph, and can generally be categorized into three categories of Centralized, Pure, and Hybrid P2P systems. In a centralized P2P system, all peers connect to a central server, which is responsible for collecting information from all peers and responding to search queries made by any of the peers in the network, an example of such model would be the bittorrent network. In a Pure P2P network, there is no central server (unlike the centralized model); rather peers forward their requests to their neighbors and via them flood the network until they find the file of interest, the earlier version of Gnutella network would fit in to this category. Lastly, Hybrid P2P systems consist of a set of super nodes and leaf nodes, where a large number of leaf nodes connect to a single super node, and super nodes connect to each other, any request by a node is sent to its respected super node, and from there forwarded to other super nodes and leaf nodes connected to the super-nodes. P2P networks which are based on hybrid systems are Kazaa and the newer version of Gnutella also known as G2.

P2P systems have many benefits, nevertheless, such network also facilitate many security threats such as propagation of worms and malwares by infecting files which are downloaded by other peers, or alternatively by exploiting vulnerabilities that exist in P2P clients. The latter point is quite important given the fact that most of the P2P users run the same client (authors in [15] state that about 75 percent of P2P users run the LimeWire client), so the Internet worms can exploit vulnerabilities limited to the LimeWire client and propagate themselves into the network and infect users. In general worms are standalone programs with the main goal of propagating themselves through the network by exploiting security vulnerabilities. They could carry a payload which provides additional functionality such as participating in a distributed denial of service attacks, accessing sensitive information, or corrupting information by sending false data. Internet worms that use P2P vulnerabilities to propagate themselves in the network are called P2P worms. In fact we use the term worm and malware interchangeable in this work, as both are in the scope of our work.

Worms propagate quickly on the Internet in a short period of time. Although many different categorization of worm proposed in the literature, here we categorize worms into two general groups of scanning and non-scanning. Scanning worms probe addresses to find new victims; also traffic pattern that they create is distinguishable from the normal traffic seen on the net. They could be further divided on three sub categories of random scanning, hitlist scanning and permutation scanning worms. But non-scanning worms select vulnerable nodes from information available to them (i.e. neighbor list, hit list) and they do not waste any time in probing address space for vulnerable hosts, so their probability of success in infecting vulnerable hosts is higher than scanning worms. The reader if referred to [11] for more detail on the mentioned propagation methods.

An alternative way to divide worms is in two subgroups of active and passive worms. Active worms do not require human intervention and transfer from a computer to another automatically. One of the most well known active worms is called the *Morris* worm. Alternatively, and unlike active worms, passive worms hide themselves within other files, and propagate as the file is copied to new hosts. In the context of P2P networks, the worm copies itself with multiple file names into the share directory of the infected host, thereby increasing the chance of being downloaded by the next victim (it is now available in multiple file names). When the file is downloaded by the next victim this process is repeated. A good example of such worm is the *Benjamin* worm.

Unlike the active worms which create anomalous network traffic as they try to propagate themselves, passive worms are quite stealth and are hidden within the normal peer exchanges. After all the peer is downloading a file he/she was looking for, unaware that the file is infected, and such exchange does not look suspicious. In fact the aim of our work, is to propose a new method with which passive worms could be detected in P2P networks. In what follows we go over the related work in Section II, and then present our methodology in Section III. We then evaluate our approach in Section IV, and finally conclude in Section V.

II. Related Works

One could divide related work into two broad categories. A set of works deal with modeling the propagation of worms others work on detection mechanisms [5], [2], [12], [6], [3], [4], [14], [1], [7], [8], [9], [13]. Below we will cover work which is most related to ours. On the modeling, Feng et al. [5] present three models (i.e. SI, SIS, and SIR models) for propagation of passive worms in unstructured P2P file sharing networks like Gnutella and Edonkey via capturing network activities and topologies (file request and download). Adeel et al. [2] further improved their propagation models by adding a fourth model named SIRE to the previously proposed propagation models.

Based on these propagation models Wu et al. [12] presented a model for detection and prevention of active worms. They proposed an overlay network based on two levels: base level and super level. In the base level, peers in the same LAN constitute a peer group in which the most powerful peer in the group has the role of super peer. These super peers create a high level P2P network. Also they are responsible to analysis anomalous information in the group and exchange information with other peers to detect worm. In the case of worm detection they used of distributed feature of this overlay architecture to analysis of abnormal traffic. If hosts in the same LAN show the similar abnormal behavior there is probability of worm propagation.

In addition, Chen et al. [6] studied the effect of both active and passive worms propagation models and simulated these worms in order to explore the factors that effect their propagation. They captured file requests and download in order to understand the infection strategies used by the worms under study. Authors monitored the inbound and outbound connections for two P2P peers with different popularity rank and considered a sudden increasing in the connections as susceptible behaviors and hence worm activity. Based on this information they presented an on-line detection algorithm for passive worms in unstructured P2P file sharing networks. They considered a sensor to probe outbound connections of n peers, and considered the peer as infected if the number of outbound connections exceeded a predefined threshold value.

Authors in [2] improved the passive worm detection method by using guardian nodes, a node which has an IDS or similar functionality. They propose a distributed framework that operate in four phases which include detection, analysis/confirmation, patch selection and patch propagation. In detection phase each guardian node is equipped with an intrusion detection system (IDS) or firewalls to monitor traffic and identify malicious behavior or perform anomaly detection. After an anomaly is detected, it verifies the anomaly against a worm database. In analysis and confirmation phase when an anomalous behavior is detected, the guardian nodes broadcast an alarm into the network, so that the other guardian nodes and peers would download the patch and spread it in the network.

Unlike prior works, in this paper our goal is to detect passive worms not based on the traffic generated nor anomalous behavior of the infected host. But, our approach is based on the popularity of a given hash value in the P2P network, and it's increase or lack of over time. In what follows we will discuss our methodology in detail.

III. Proposed Methodology

As stated earlier, our main goal is the detection of passive worms in P2P networks. There are many such worm is the wild, where they use P2P clients for propagation and copy themselves with different names to the shared folder on the victims machine. Below are a few examples listed from [16]:

- *P2P-Worm.Win32.Mandragoere*: is a passive worm that propagates using the Gnutella file sharing network. It announces the file names that is being searched, but with EXE suffix.
- *P2P-Worm.W32.Nugg*: is a P2P worm that uses Gnutella clients like LimeWire as a platform to gain access to the user computer.
- *P2P-worm.win32.polip.a*: is a virus/passive P2P worm that uses Gnutella file sharing protocol or email messages for propagation.

Use of passive worms is widespread, as the worm attaches itself to the popular P2P files and given their propagation

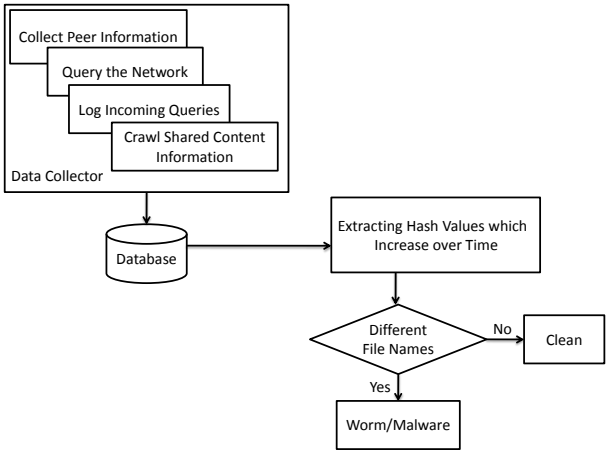

Figure 1. Worm Detector Architecture

method in which users request for these popular files, the passive worm is able to infect and propagate itself quickly. Furthermore the worm creates multiple copies of itself in the users shared directory with different names. This increases the probability of download by the next victim. In P2P systems the contents of each file is hashed in order to obtain an identifier specific to the content of the file. Hence two different versions of the same mp3 file will have different hash values even though the same song/music is present in both files. Now, even though worms copy themselves multiple times, the worm code remain constant and as such the hash value for different copies of the worm remains the same, while the name of the files is quite different from one copy to the next. Therefore, one could then look at all hash values present in the P2P network and see if a given hash value is representing a wide variety of file names this would be an indication that the content which results in such hash value is suspect and most likely a worm or malware.

But there are many files and hence unique hash values available on a given P2P network, and it would be a huge challenge to analyze the relationship between hash values and file names. Hence, based on the behavior of the worms, and their goal of propagation in the network, we conjecture that by looking at how a hash value gains popularity in the P2P network we could concentrate on potentially malicious content, although we should not ignore the fact that legitimate files may also become popular in a short period of time. Therefore after extracting hash values which have become popular in a short amount of time we are left with a small number of hash values which we could then analyze to see whether they represent a single file name hence clean, or are representing a wide variety of file names hence malicious.

As figure 1 shows the detection system has a simple architecture which is explained in more detail below:

- Data collector: The responsibility of the data collector is to log incoming queries; crawl shared content information. These information could be file names, hash values, peers IP addresses, and so on. In fact the collector monitors the IP addresses of peers and their shared folder, and records their unique host id, unique file hash value, file name, crawl time, and other information in some distinct tables in a database.
- Popularity analyzer: The popularity analyzer is simply tasked with finding hash values which increase over time.
- Worm detector: File hash values which are found to be increasing over time, are tagged and checked. If the hash represents a variety of file names, then we have a worm or malware. If the hash represents a single file name with minor variations then we have a clean file.

IV. Evaluation

We use the Gnutella P2P network to evaluate our proposed approach. Gnutella is a distributed and decentralized peer-to-peer file sharing system, where peers join the network by connecting to a few other peers in the network. Search is decentralized, so that the requester sends a query to it's neighbors, which in turn is forwarded by the neighbors to their neighbors. If the receiving peer has the requested file it would responds to the request using the same path on which the query was received on, otherwise the query is forwarded to other peers. The current version of Gnutella has improved scalability by employing a two level hierarchy. New nodes connect into the network as leaf-nodes, with no routing responsibilities. More stable and powerful nodes are elevated to an ultra-peer node which is tasked with routing messages. An overlay network is built between the ultra-peers, within which queries and other messages are routed. Each leaf-node is then connected to a few ultra-peer nodes. Other important properties of the Gnutella protocol include the QRP (Query Routing Protocol) with which the query is only forwarded by the ultra-peer to the leaf-nodes which may have a response to the query, and DQ (Dynamic Query) which is used to limit the query broadcasts and limit the number of search responses.

A. Experimental Setup

In order to collect information from the Gnutella network, such as the number of nodes and files in each node, we employed the IR-wire crawler [10]. IR-Wire is executed over Limewire [17] client, which is a popular Gnutella client. We crawled the Gnutella network for 12 days, starting from April 15th, 2010. The network was crawled 4 times, each crawl lasting for a duration of 3 days, hence we had 4 crawl data sets. The data was then loaded into mysql for processing, and a set of script were used to obtain the popularity of the hash values and analyze the change in their popularity over different collection periods. Table I present

the number of peers, and unique file hash values found in each of the collection periods.

Table I
NUMBER OF AVAILABLE PEERS, AND UNIQUE FILE HASH VALUES

Time(Days)	Number of Available Peers	Number of Hash Values
April 15th-18th	134,469	930,128
April 19th-21st	245,465	1,638,680
April 22nd-24th	258,485	1,121,566
April 25th-27th	217,208	1,641,291

B. Results

First we found the top 50 popular file hash values in the last collection period (April 25th to 27th), We then selected file hash values which were monotonically increasing up to the last collection period, excluding file hash values which remained constant over the collection periods. We were left with 34 file hash values, which are of interest and represent hash values which have become popular over our collection periods.

We next investigate the 34 file hash values and found 11 to represent a wide variety of file names, which we designate as malware. We found the remaining 23 file hash values each representing a single file name with minor variations, we hence designate them as clean. In order to verify our findings we tried downloading the 34 files, using a Limewire feature which allows the user to query for files with a given hash value, and verified their status (i.e. clean or infectious) by checking them with two anti-virus software (Kaspersky and Avira).

As presenting in Table II, out of the 11 suspected files, we were able locate and download 7 of the files and verified that these files are infectious by testing them with the noted anti-virus software. We were also able to locate and download 20 of the files we designated as clean, Table III, which we verified as being clean again by testing them with the mentioned anti-virus software.

V. CONCLUSION

The widespread use of P2P networks among computer users make them suitable for the worm propagation and also accelerates worm propagation in comparison with other networks. This research contributes to the understanding of the ways with which worms and malwares propagate in P2P networks and presents a new methodology for passive worm and malware detection. We were able to detect correctly all worms and malwares in the top 34 popular files. That shows that our approach is effective and accurate. Obviously, there are a number of issues which need to be improved on and we are actively working on them as part of our future works. For example, we are working on incorporating our approach in to an on-line system, instead of working on offline datasets as done in this work. We believe that such approach is more promising that alternate approaches which look at traffic from nodes in the network, and look for anomalous patterns.

ACKNOWLEDGMENT

The authors would like to thank Ali Ashrafi for assisting with the experimental setup, and Ali Fahimian for helpful discussions.

REFERENCES

[1] S. Shakkottai and R. Srikant, *Peer to Peer Networks for Defense Against Internet Worms*, USA: ACM International Conference Proceeding Series, Proceedings from the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer and communications sytems, 2006.

[2] M. Adeel, L. Tokarchuk, L. Cuthbert, C.Sh. Feng, and Zh.G. Qin, *A distributed framework for passive worm detection and throttling in P2P networks*, USA:Proceedings of The Fourth International Workshop on Dependable and Sustainable Peer-to-Peer Systems (DAS-P2P), 2009.

[3] Y. Zhang, Zh. Li, Zh. Hu, Q. Huang, Ch. Lu, *Evolutionary Proactive P2P Worm: Propagation Modeling and Simulation*, Hubei: Second International Conference on Genetic and Evolutionary Computing 2008 (WGEC '08), 2008.

[4] F. Wang, Y. Zhang, and J. Ma, *Modelling and Analyzing Passive Worms over Unstructured Peer-to-Peer Networks*, International Journal of Network Security, 2010.

[5] C. Feng, Z. Qin, L. Cuthbet, L. Tokarchuk, *Propagation modeling of passive worms in P2P networks*, Chengdu:IEEE International Conference on Cybernetics and Intelligent Systems (ICCIS), 2008.

[6] G. Chen, R.S. Gray, *simulating non scanning worms on P2P networks*, USA:Proceedings of the 1st international conference on Scalable information systems(ACM), 2006.

[7] K. R. Rohlo, T. Basar, *Deterministic and Stochastic Models for the Detection of Random Constant Scanning Worms*, USA:ACM Transaction Models Computer Simulation, 2008.

[8] Zh. Li, Y. Zhang, Zh. Hu, H. Lin, Ch. Lu, *Containing Proactive P2P Worm Based on its Multicast Characteristic*, nswctc:International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009.

[9] X. Chunhe, Sh. Yunping, L. Xiaojian, G. Wei, *P2P worm detection based on application identification*, china: Higher Education Press, co-published with Springer-Verlag GmbH, 2007.

[10] Sh. Sharma, L.Th. Nguyen, D. Jia, *IR-Wire: A Research Tool for P2P Information Retrieval*, USA: In SIGIR Open Source Workshop-ACM, 2006.

[11] S. Staniford, V. Paxson, N. Weaver, *How to 0wn the Internet in Your Spare Time*, USA:Proceedings of the 11th USENIX Security Symposium, 2002.

Table II
INFECTED FILE HASH VALUES FROM OUR PERSPECTIVE AND COMPARING WITH REALITY

	Infected File Hash Values From Our Perspective	Reality	Types of Infection	# of unique file hash values in 4 collection periods
1	urn:sha1:52XMEBBC4OB5U7H3WPRJTDMELYAEC2PG	Not-Found		0-361-462-825
2	urn:sha1:EXYYYD2A2XQ6LNMKER255DE7UDMZENTV	Not-Found		0-220-399-811
3	urn:sha1:LUHORUJSSAIHLFR4MLVD7LJ7JI6KPWIC	Not-Found		0 -188-371-804
4	urn:sha1:BZTVR4TE2IGQ3CJXLAL6BYERIHU7VZKZ	Infected	W32/Tracur.A.gen!Eldorado (Avira) P2P-Worm.W32.Nugg.an (Kaspersky)	0-164-343-791
5	urn:sha1:4ROTVVDGYW6OK2HDTWEAUQZKDRNPB7IO	Infected	EXP/ASF.GetCodec.Gen (Avira) TrojanDownloader.WMA.GetCodec (Kaspersky)	20-31-50-761
6	urn:sha1:3NBFXFX6QQ2EFFGV6XTG6AJH3NJJPAMW	Infected	EXP/ASF.GetCodec.Gen (Avira) TrojanDownloader.WMA.GetCodec (Kaspersky)	15-30-48-760
7	urn:sha1:HMRSJZ2G5VJKIABQUBVL4T4J3HOB45KN	Infected	EXP/ASF.GetCodec.Gen (Avira) TrojanDownloader.WMA.GetCodec (Kaspersky)	14-23-46-752
8	urn:sha1:MHK53AYOPYUA5FDWCI2CZY6IUOFH66LE	Infected	EXP/ASF.GetCodec.Gen (Avira)	12 -18-44-747
9	urn:sha1:ZWFRCWZUXWVFBR6VD54K7GAHX4MYE4O3	Not-Found		12-18-44-747
10	urn:sha1:2CTAOY6ZEJC555BUADCDNG6TVNQRSISR	Infected	TR/PSW.Magania.BGWQ (Avira) Trojan-Downloader.Win32.Agent (Kaspersky)	12-19-44-747
11	urn:sha1:BEQJUSFQPGCD7YZNRZARII2FA3UCLAWP	Infected	Gen:Trojan.Heur.FF9F6075C7 (Kaspersky)	12-18-44-747

Table III
CHECK THE RESOLUTION OF OUR METHOD FOR HEALTHY FILE HASH VALUES

	Clean file hash values from our perspective	Reality	Name of the file	# of unique file hash values in 4 collection periods
1	urn:sha1:56H2ESKS7VBDLWBKS4JTD6OZF6XJTXXF	Not Found		0-149-218-1547
2	urn:sha1:SD3P6X5CR6TDBEQVB7SGVCEHMUY72Y5Z	Clean	2Pac - Tupac-Smoke weed all day	665-607-1388-1324
3	urn:sha1:VVIMBEAUTLUHYAIHSOJXJXRAPDDRGNSY	Not Found		0-23-240-1152
4	urn:sha1:NHBOV3XKVUHUUW3YSH7Q52MAQ55QH4SD	Not Found		0-106-131-1079
5	urn:sha1:ZKWZ7LX6FR4K5RG4O47JLA4TU4EQSUH7	Clean	Lady GaGa- Bad Romance	540-971-530-1007
6	urn:sha1:M3E6PYDHQYMS5GQBFB67XJU4NPLKWWE5	Clean	Lady Gaga feat.BeyoncTelephone	465-835-632-927
7	urn:sha1:KXFXLONJKLM6EJUOUAQGAW75DJMCXX2O	Clean	Young Money - Bed Rock ft Lil Wayne	514-973-505-878
8	urn:sha1:RINPDWXHUZH6B3QCHXOGX6CMRA4P5JTI	Clean	Jay Sean - Down (feat. Lil Wayne)	455-762-465-821
9	urn:sha1:F3ZXDF5I2FQ3MJ545WJNTPZGLYG7FVUI	Clean	Kesha-TiK-ToK	534-573 -367-821
10	urn:sha1:6LH4DZ4PJO26IISP2SU5PSXZJ3VXJVVH	Clean	FROSTWIRE LAWSUIT WARNING - VISIT	0-13-141-812
11	urn:sha1:VTPUGWDHLUHPVJ4W52CDYRTAHFI5CI73	Clean	Black Eyed Peas - I'mma Be(1)	377-672-467-789
12	urn:sha1:TBY5WSR3USY6J2XBDSW3LNHUJ2IA6O3E	Clean	BoB-Nothing-On-You-Feat-Bruno-M	320 -448-505-768
13	urn:sha1:CKXLVWPSQ5PW3UPCBC2SKTOCZ7R7RXHH	Clean	Lady GaGa Paparazzi	362-718-392-754
14	urn:sha1:5HA42NB3ILWZVIWIO74WJIMSA5HFOVXG	Clean	Drake-Forever Ft. Kanye West	365-642-383-702
15	urn:sha1:PRZJX2BEVZ5WGAGYLFLQEQDYEOTFT2EP	Clean	Rihanna - Hard -Feat. Young Jeezy	364-568-358-685
16	urn:sha1:FLDSDJSIOCIKJLQY4GNPWKX4776CUWQB	Clean	Rihanna - 2009 - Rated R - 08	330-585-340-681
17	urn:sha1:JZJAV2LMBGPBOUDV7KMVFMHW3VMQL4BP	Clean	Mike Jones ft. T-pain, Twista, Lil'	185-3877-595-678
18	urn:sha1:F2DK7EAQ4GD25FAECKFE7GC3M652UE3B	Clean	Justin Bieber feat Ludacris Baby	327-543-452-662
19	urn:sha1:EGEY64G7O36PHFQYHIAOE24TC2SVCP6T	Clean	Lady Antebellum-01 Need You	241-449 -246-640
20	urn:sha1:DN2OSWYUG4OQ4PRJ3LSWL5IF5DANSPXT	Clean	01 Drake - Over	261- 532-306-639
21	urn:sha1:YQED7JCJEFWHITOXNGDRP4ORY5LJOQCY	Clean	Jason Derulo - In My Head	360-576-359-631
22	urn:sha1:CNNMT2U4653RFPNVKJRBACF7UH55LOD3	Clean	Miley Cyrus - Party In The USA	329-680-355-627
23	urn:sha1:AQZAFWTALXDH6GS74KBC5PFC24LVOXBN	Clean	Chris-Brown I-Can-Transform-Ya	409-567-327-624

[12] K.G. Wu and Y. Feng, *Proactive worm prevention based on P2P network*, Korea: International Journal of Computer Science and Network Security(IJCSNS), 2006.

[13] W. Yu, C. Boyer, D. Xuan, *On Defending Peer-to-Peer System-based Active Worm Attacks*, in Proceeding of IEEE Global Telecommunications Conference (GLOBECOM), 2005.

[14] T. Lia, Zh. Guana, X. Wu, *modeling and analyzing the spread of active worms based on P2P systems*, Computers and Security, 2007.

[15] J. Lloret, J.R. Diaz, J.M. Jimnez, M. Esteve, *The popularity parameters in unstructured P2P file sharing networks*, USA:Proceedings of the 4th WSEAS International Conference on Applied Informatics and Communications, World Scientific and Engineering Academy and Society (WSEAS), 2004.

[16] http://www.viruslist.com/en/virusesdescribed?chapter=153311928,28.06.2010

[17] LimeWire home page. http://www.limewire.com/en