A DISTRIBUTED FRAMEWORK FOR PASSIVE WORM DETECTION AND THROTTLING IN P2P NETWORKS

Muhammad Adeel[1], Laurissa Tokarchuk [1], Laurie Cuthbert[1], Chao-sheng Feng[2], Zhi-guang Qin[2]
[1]Queen Mary, University of London, School of Electronic Engineering and Computer Science, London E1 4NS
{muhammd.adeel, laurissa.tokarchuk, laurie.cuthbert}@elec.qmul.ac.uk
[2]School of Computer Science & Engineering, University of Electronic Science and Technology of China,
Chengdu Sichuan 610054,China
csfenggy@126.com, quinz@uestc.edu.cn

We analyse different worm and patch propagation models along with the ones we have developed and evaluated as a part of our ongoing passive P2P worm & patch modelling project. This is followed by a brief discussion on worm detection mechanisms proposed by various authors. Towards the very end of this article, we propose a distributed framework for passive worm throttling in P2P networks and discuss its feasibility and efficiency keeping in view different design considerations.

***Index Terms*—Modelling and Security.**

I. INTRODUCTION

Peer to Peer (P2P) applications as we know them today in form of Gnutella [1], Napster [2], Freenet [3], Kazaa [4], Pastry [26], CAN [4] and Chord [27] contribute the major chunk of the Internet traffic. Being technically categorized as unstructured and structured, the P2P networks have diversified applications like file sharing (e.g. Kazaa and BitTorrent), collaborations (e.g. ICQ and Skype), process sharing (e.g. Distributed.net and Adhoc Networks) and distributed computing (e.g. Seti@home and Folding@home).

Decentralized nature of P2P networks benefits through the properties like scalability, reliability, fault tolerance and load balancing, while in presence of no centralized authority, these networks are prone to many security threats in respect to breaches of confidentiality, integrity, authentication, access control and non-repudiation [5]. While authors in [15] have discussed different threats to the cyber space, this work is confined to the passive worms in unstructured P2P networks.

Over the years, worms have emerged as a main source of trouble in P2P networks. Worms can be categorized mainly as scanning and non-scanning. Scanning worms always keep on probing addresses for new victims. They do waste time in probing unused addresses and may potentially have a high rate of failed connections. Moreover, they do not blend with the normal P2P traffic [6]. Due to the circumstances discussed, the non-scanning worm could sometimes be more dangerous than the scanning ones as they chose the vulnerable nodes through the neighbor lists and are hence more successful in acquiring precise and fast knowledge of their prey.

In this paper we mainly focus on passive worms that hide themselves in popular P2P resources by embedding malicious code in executable files. This strategy of selecting the targets has made passive worms unpopular & less attended in history because most of the files shared in the early P2P networks were non-executable files like MP3 or some other media files. However, more recent popular P2P systems, like Bit Torrent, Kazaa, eDonkey2000 & others provide the users with much easier access to executable files, and make passive worms become a major threat yet again to the safety of the P2P networks [7].

The passive worms operate in a purely epidemic manner to spread in the network. Firstly they embed themselves in the popular executable files in the P2P network and make a few copies in the sharing folder of the infected user. Once another user downloads the files and executes them, the worms duplicate themselves and create a few new copies in the sharing folder, which increases their possibility of being downloaded by the other vulnerable users. Since the user can only be infected after the file is executed, the downloading of the passive worms are, most of the time, treated as legitimate P2P network behavior and this actually makes it quite difficult to detect [7]. Some researchers define the passive worms as the ones that attach to files and propagate with user activities as viruses [8]. In the discussion to follow, we would use terms "worms" and "viruses" alternatively for the passive worms.

This paper is organized as follows. In section 2, we initially discuss the contributions of other authors in the field of passive worms modeling and then briefly analyze our work on worm modeling as a part of our ongoing project. In the second half of the same section, we take an overview of the literature in the field of patch modeling and then describe our work on the passive worm modeling. In section 3 we study different techniques in the literature to detect the worms in P2P networks and in section 4 we propose and analyze a framework for detection of worms and measures in the P2P network in presence of malicious activity.

II. RELATED WORK ON MODELLING WORMS & PATCHES

Researchers often try to reverse engineer the worm

propagation to understand its effects and to design the remedies. There have been lot of efforts to study propagation of P2P active worms and defences against them but a little has been done in regards to passive worms [9]. Although such worms may propagate in a slower passion, the P2P networks are themselves the vehicles for fast passive worm propagation. As discussed in section 1, the P2P worms propagate as a part of legitimate network activity and hence are difficult to detect than scanning worms. Many studies are underway to analysing the patterns of virus propagation in P2P networks to better understand worm behaviour. For this article, we mainly focused on unstructured file sharing P2P networks such as Kazaa and BitTorrent because most of the existing P2P worms target these kinds of systems.

A. Worm Propagation Modelling

For P2P networks being complex systems, it may not be feasible to use an analytical approach to model worm propagations without making overly simplified assumptions [8], hence based on various assumptions regarding file downloading patterns, different authors have come up with various models on P2P worm propagation.

Guanling Chen et al. in [8] have modelled the propagation of all types of P2P non-scanning worms. In case of passive worms, more the number of infected files a worm can generate with popular file names, the more likely other users will download these files and become infected and hence prevalence of worms is increased in P2P networks. Similarly for reactive worms, mixed infection strategy in which connection establishment for requesting a file or the other way round, spreads the worm at much higher rates as predicted by previous works i.e. [10]. For the reactive worms although a few technical issues produced some varying results, still it was proved that the larger cache is proportional to the worm propagation.

Jie Ma et al. in [11] observed that worms reach better attack performance with increase in vulnerable nodes. While enlarging the networks with a certain probability, it was concluded that enlarging P2P system can bring more vulnerable nodes in P2P system and hence can exponentially increase the passive worm propagation.

Thommes & Coates modelled the virus spreading assuming the networks like eDonkey 2000, Kazaa and Gnutella. They use epidemiological modelling and progression for all peers in their model. They conclude that the probability of a peer downloading an infected file is proportional to the prevalence of infected files in the network [12].

Authors in [13] give analytical results about the influence of different parameters on worm propagation in both unstructured and structured P2P systems. Similarly Weaver et al. have emulated worm propagation on a scaled-down version of Internet [14]. Although the work of authors on hit-list worm, flash worm and routing worms, summarized in [9], is of abstract level but still very considerable.

1) Worm Propagation Modelling in this Project

As a part of our ongoing project, we have developed an epidemiological spreading model for passive worms in P2P networks. Although a part of this work is published in [9] and could be referred for further details, here we would like to summarize whatever the findings of this phase of our work are.

Based on the worm propagation patterns and specially the node status during the data collection process, we have tested four distinct models. In SI model a susceptible node gets infected when it downloads a virus file while in SIS model a susceptible node gets infected and can get back to susceptible mode after taking care of the worm through an assumed mechanism present at the node itself. In SIR model a susceptible node gets infected and besides changing to susceptible status again, a proportion of such nodes could change to the immunized state and so when an infected peer is removed, it is assumed that all infected files on the peer are deleted and the peer can be infected no more after that. Finally in SIRE model that is an extension of SIR model, some infected nodes may get back to susceptible mode while some could get immunized.

Figure 1 summarizes the results of these four models. The number of infected peers in the SI model increases fastest. Compared with the SI model, the number of infected peers in SIS model has a slower increasing rate and for some time units the number keeps invariable i.e. the infection reaches the steady state. The curve of the SIR model and the curve of the SIRE model have common features. Both curves go up first and then after reaching the peak prevalence, begin to go down. At any given time unit, the number of infected peers in the SIRE is less than the ones in the SIR model. It is so because in the SIRE model, a proportion of infected peers return to be susceptible, while in the SIR model, infected peers do not return to the susceptible state.

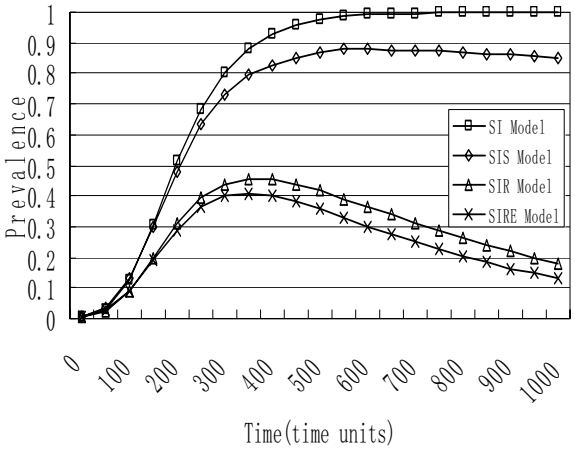

Fig. 1. Worm Propagation

B. Patch Propagation Modelling

The point of patching is to effectively halt the propagation of worms by fixing the holes in the application that allows them to do so [16]. Reverse engineering the patch propagation in the network is as important as worm propagation modelling. After better understanding the behaviour of worms and their propagation patterns now we take a look at anti-worm propagation and its effect on the polluted P2P network. With the ever-increasing efficiency of worms, there has always been a need of patch propagation modelling to keep the competition between worms and anti-worms balanced.

There have been lot of efforts going on to develop better patch propagation model and quite considerable in this regard is the work by Srinivas et al. presented in [16]. The author argues that the start of patch dissemination is

important in early stages of threat detection as at that time the population of infected hosts would be very less. Authors have also studied host-based and server-based patch dissemination and concluded that even a small rate of patching by the peers of a P2P network has far better impact than an enormous rate of a fixed number of patch servers. Progressive Susceptible Infectious Detected Removed (PSIDR) model analyses the behaviour of patch dissemination in the P2P network. The results conclude that greater the value of signature delay (π), greater will be the population of viruses in the network. It is considerable that the π represents the time at which the patch is introduced in the network [17].

Michael and David, besides other worm throttling mechanism, have also evaluated Patching Counter-Worm mechanism where the anti-worm uses the same propagation strategy as used by the worm itself. An effective response requires a combination of low response time and a sufficiently large initial population [18]. Similarly the work of Milan et al. [19] and Frank et al [20] is also considerably important in this regard.

1) Patch Propagation Modelling in this Project

After developing a worm propagation model for unstructured P2P networks and having analysed the background work in the field, we were all set to model the patch dissemination patterns for P2P networks. A part of this work has already been published as [7] and could be referred for further details. Here in this section, we analyse the conclusions drawn at the end of this phase of the project. Five different models were developed naming Patch Mode 1 through Patch Mode 5 were implemented and evaluated on Peersim simulator [22]. The SI model (described in worm modelling section of our project) was selected to dispatch the patches and hence the deducted results were analysed. Figure 2 gives the performance of all the scenarios we modelled. For Patch Mode 1, the initial patch file population in the network is 500 while every patch file has to compete with other popular files for network resources. The results show a rather exponential rise in number of infected nodes in the network out of 10,000 nodes. For Patch Mode 2, there is a separate P2P file sharing system for the patch files and hence it increases the containment of worms in the network.

In case of Patch Mode 3, we increase the number of neighbours to probe when a node decides to download a patch file and hence it is more sort of a worm behaviour demonstrated by the path file. For Patch Mode 4, we have assumed that the node would not probe its N neighbours to download the patch file, instead, the recovered node that already has the patch file sends the patch file to its N neighbours before it uses P2P network to download a sharing file. Patch Propagation Mode 5 outperformed the other patch modes. In this mode we have combined Patch Mode 3 & Patch Mode 4 to get the much-improved performance in perspective of infected nodes in the network.

III. Detection of P2P Worms

After having the worm and patch dissemination patterns analysed in previous section we now proceed to the worm detection which is bound to be the next phase of this project. From the work of different authors including [16 and 18], it is evident that earlier detection of worms is the key towards low virus population in a P2P network. It is also observed that earlier injection of anti-virus in the network eases the rescue process [17]. Some other researchers focused on the defence against the Internet worms. J. Sandin introduced a worm detection method by using P2P network [23] while M. Costa et al. also focused on the same by monitoring the abnormal behaviour in the network traffic. Most of the worm detection methods stated and summarized in [7] are defence methods against scanning worms in which the key towards detection is their abnormal network behaviour.

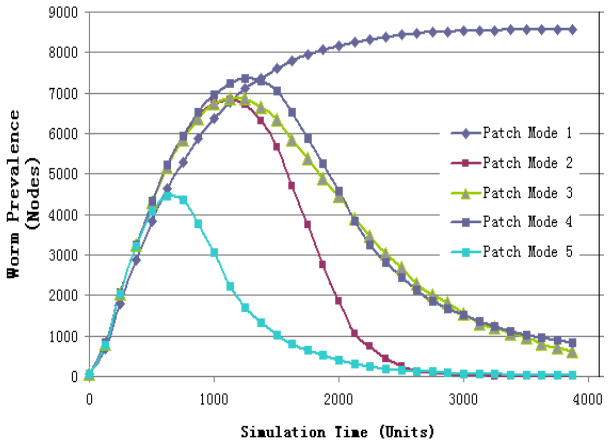

Fig. 2. Patch Propagation

Most of the worm detection techniques focused random scanning worms. In such worms, it is achievable through capturing the scans spread into unused IP space, by detecting exponential scan increases and probe failures, or by hypothesis testing on fast port scans [8]. These methods are not effective for non-scanning P2P worms. Hence we are left with the limited options like host-based detection methods, such as Tripwire that have potential to detect P2P worms. An observation regarding such systems is that it is difficult to deploy such systems onto wide-area P2P nodes [24]. So the network-based worm detection methods are required to cope up with scalability of P2P networks and are efficient as well.

Authors in [16] have discussed the mechanism in which some collaborative firewalls upon detection of worms spread implicit and explicit alerts in the network. However, this scheme is prone to generation of high rate of false alarms in the network. There are some important observations about detection of passive worm from their behaviour. Authors in [8] have highlighted some general characteristics regarding behaviour of nodes during a possible worm attack and we elaborate this behaviour in the lines to follow. Passive worms will create and share popular files on the victim hosts and hence a considerable increase in the popularity of these nodes could be observed. Moreover the victim node will see an increased number of file requests by other peers and hence a disproportional or unusual increase in number of inbound and outbound connections could also be noticed.

IV. Proposed Distributed Security Framework

Based on all the considerations from our modelling work and in-depth analysis of worm detection techniques, in this section, we propose a framework for P2P networks to elegantly handle the worm attacks from their launch till the remedy. The framework is based on the concept of guardian nodes that is already there in literature [6]. The guardian

nodes would perform a purpose-specific functionality, in our case analysing the traffic at first instance. The phases of implementation could be divided as into detection, analysis & confirmation, patch selection and finally the patch propagation. Figure 3 gives a pictorial view of the framework. A light-weight communication framework was required to be built on top of existing P2P technologies that makes every peer share the threat with guardian node and similarly, the guardian node shares its intelegence with all the peers and other guardian nodes in the network

A. Detection Phase

As an integral part of the framework, the guardian node is equipped with observation software like Intrusion Detection System (IDS) and/or firewalls to analyse the traffic patterns and to identify any malicious behaviour. Some authors have discussed the case where such detection devices (specially the firewalls) may also be present on individual peers and detect the threat [25]. Besides detection of attacks, the positioning of IDS in P2P network is vital. Hence locating the nodes responsible for intrusion detection on the key spots in the networks is important to make this activity rather efficient in identifying the threats.

There may be different kinds of intrusions into a network and hence the IDS should be capable of performing varied sort of detections including misuse and anomaly detection. Deployed as either a network-based system or a host-based system, the IDS could be used to detect misuse by a host or network, or anomaly at host or network. Norbik et al. in [28] give a detailed overview of the Artificial Intelligence techniques used in intrusion detection. Such techniques would be utilized in the framework for both anomaly and misuse detection.

In the event of worm detection by the nodes, the nodes would explicitly ask the guardian nodes to confirm its malicious behaviour and rest is taken care of by the guardian node. Coming back to our case in which the guardian node detects some malicious code, it would request the worm definition database to look for the worm definition and confirm it. Besides the content, the threat could also be detected through the behaviour of the network or traffic suppose by an alarmingly increased number of connections. This activity may be traced by the firewalls and reported to the guardian nodes for the remedy.

B. Analysis & Confirmation of Threat

In this phase, if the guardian node, by looking at the virus definitions confirms the threat, it would generate the alert to the entire P2P network. This alert generation would have different meanings for the peers and other guardian nodes in the network. The guardian nodes would get the patch ready and they could simply push the patch to other devices or wait for this patch to be pulled by the devices.

C. Patch Selection

Selection of a proper patch from the patch reservoir is a key task when we look at the worm throttling process. Prompt and proper patch availability could let the network recover quickly from the attack. While the definitions for some worms are not there, techniques used by Frank et al. in [20] could be deployed to convert the worm into anti-worm. Failure to which could require a human intervention.

D. Patch Propagation

A better strategy is required to be deployed to make the patch dissemination process fast to an extent that it could take over the worms in the network. As described by [7], the speed of epidemiological behaviour of worms has always been a hard question. Hence when the patch is ready, it could either be propagated straightaway to the peers or the guardian node would wait for the peers to download it in response to the alert. An important phase in this regard is the communication between guardian nodes upon receiving the patch. When a guardian node detects a threat directly or through any peer, in an alert message, it is assumed that it would also announce the identity of the worm so that the peers that may already have the patch could start taking care of the worm. The guardian nodes receiving the alert would make the patch available in their shared folders or even reactively flood the patch into the network.

The framework is aimed at sustaining the distributed behaviour of the network as all the nodes act together to detect the threat as early as possible suggested by the authors like [17]. Guardian nodes may share the patch update information between them and make sure that an updated information lies at every guardian node ready to be downloaded by a peers. Besides the generalized worm threats, some minor Denial of Service attacks like TCP flood or UDP flood attacks that are proposed by [25], could also be detected and taken care of by applying their logic on top of the proposed protocol.

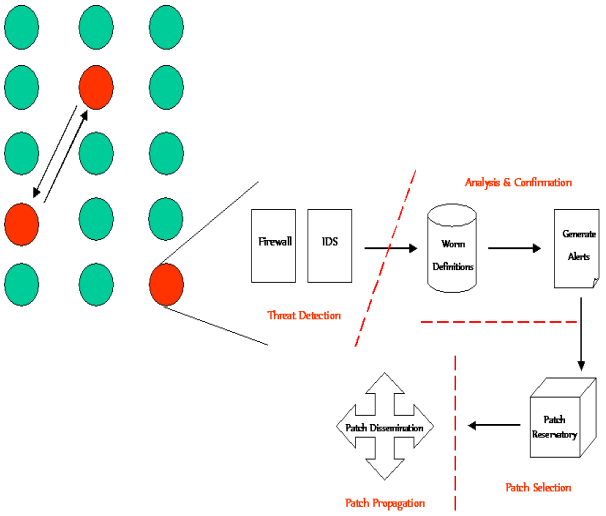

Fig. 3. Proposed Framework

V. Discussion

Due to the limitations on scope of this article, we do not proceed with the discussion on types of packets that would make the communication between framework entities possible. However, an important consideration regarding selection of guardian node as discussed earlier is that the selection could be performed through the election process defined for Adhoc Networks [21].

We have adopted a multi-tier policy in which initially an active throttling approach automatically contains the damage caused by fast spreading viruses. Rather than attempting to prevent a machine becoming infected which is the role of most anti-virus softwares, the throttle instead

prevents the further propagation of the virus from that infected machine [17]. Hence the addresses from which the worm attack is being generated could be blocked for some duration to at least contain this epidemic while the recovery process would be underway in parallel. The alert messages could be made more effective if they also carry the information that could result in probing all the peers to block the traffic from some particular addresses. Doing so, these alerts could play a vital part in worm containment process. Meanwhile the devices like firewalls and IDSs could take charge of the major recovery process through patch propagation and worm scans on the individual peers.

VI. Conclusions

After briefly analysing the worm and patch modelling work and a considerable review of worm detection mechanisms, we conclude that worm detection could be very effective if done in a distributed manner. We argue that for the scalable P2P networks, the distributed or technically hybrid detection mechanisms could prove even more effective than conventional centralized detection. Towards the later part of the article, we proposed a distributed threat detection and worm throttling framework and deducing from the previous work in the field we could safely say that the performance of this framework would depend on the prompt and intelligent threat detection, efficiency in sharing the threat information with the entities that matter, and a very strong recovery strategy.

Acknowledgments

The author would like to thank the anonymous reviewers for their valuable comments and suggestions that improve the presentation of this paper. This work is supported by the National Natural Science Foundation of China under Grant No.60473090 and a joint research project funded by the Royal Society in the UK and by the National Natural Science Foundation of China (NSFC) under Grant No.60711130232.

References

[1] Napster homepage, http://www.napster.com/
[2] Gnutella homepage, http://www.gnutella.com/
[3] Eric Chien, "Malicious Threats of Peer-to-Peer Networking", Symantec White Paper, 2003.
[4] www.cim.mcgill.ca/~sveta/COMP102/P2P.pdf
[5] William Stallings, "Cryptography and Network Security, Principles and Practice", Second Edition, Prentice Hall Publishing, ISBN-13: 9780130914293, 2001.
[6] Lidong Zhou et al., "A First Look at Peer-to-Peer Worms: Threats and Defenses", Book Chapter, *Peer-to-Peer Systems IV,* Springer Publishing, 2005.
[7] Bo Zhan et al., "Defense against Passive Worms in P2P Networks", Proceedings of Networking & Electronic Commerce Research Conference (NAEC 2008), 2008.
[8] Guanling Chen et al., "Simulating Non-Scanning Worms on Peer-to-Peer Networks", Proceedings of INFOSCALE '06, Hong Kong, 2006.
[9] Zhiguang Qin , "Propagation Models of Passive Worms in P2P Networks", IEEE International Conference on Machine Learning and Cybernetics (ICMLC), 2008
[10] S. Staniford et al., "How to Own the Internet in Your Spare Time", Proceedings of the 11th USENIX Security Symposium, San Francisco, 2002.
[11] Jie Ma et al., "Modeling Passive Worm Propagation in Peer-to-Peer System", International Conference on Computational Intelligence and Security, 2006.
[12] R.W. Thommes et al., "Modeling Virus Propagation in Peer-to-Peer Networks", Proceedings of IGCICS 2005.
[13] W. Yu et al., "Analyzing impacts of peer-to-peer systems on propagation of active worm attacks", Technical report, Department of Computer Science & Engineering, Ohio-State University, 2004.
[14] N. Weaver et al., "Preliminary results using scale-down to explore worm dynamics", Proceedings of the 2nd ACM workshop on Rapid Malcode, Washington, 2004.
[15] M. Adeel et al., "Classification of Cyber Crimes and Pertinent Legislation in Pakistan", Proceedings of the Conference on Cyber Technology Issues, Challenges and Development, Pakistan, 2007.
[16] Srinivas Shakkottai et al., "Peer to Peer Networks for Defense Against Internet Worms", Proceedings of Inter-Perf'06, Italy, 2006.
[17] Matthew M. Williamson et al., "An epidemiological model of virus spread and cleanup", HP Technical Report, February 2003.
[18] Michael Liljenstam et al., "Comparing Passive and Active Worm Defenses", Proceedings of QEST'04, 2004.
[19] Milan Vojnovic et al., "On the Effectiveness of Automatic Patching", Proceedings of WORM'05, Fairfax, Virginia, USA, 2005.
[20] Frank Castaneda et al., "WORM vs. WORM: Preliminary Study of an Active Counter-Attack Mechanism", Proceedings of WORM'04, Washington, 2004.
[21] Johann Van Der Merwe et al., "A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks", ACM Computing Surveys, Vol. 39, No. 1, Article 1, April 2007.
[22] peersim.sourceforge.net
[23] J. Sandin, "P2P systems for worm detection," Proceedings of DIMACS Workshop on large scale attacks, Piscataway, USA, 2003.
[24] G. H. Kim et al., "The design and implementation of tripwire: a file system integrity checker", Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, USA, 1994.
[25] Jamie Twycross, "Implementing and Testing a Virus Throttle" Proceedings of the 12th USENIX Security Symposium, Washington DC, USA, 2003.
[26] Al Sukkar, G. Afifi, H. Senouci, S. M. "Party: Pastry-Like Multi-hop Routing Protocol for Wireless Self-Organizing Networks", Proceedings of the First Mobile Computing and Wireless Communication International Conference, 2006. MCWC 2006. 17-20 Sept. 2006
[27] Chord Protocol, www.inf.ed.ac.uk/teaching/courses/ip/chord-desc.html
[28] Norbik Bashah and Idris Bharanidharan Shanmugam et al., "Hybrid Intelligent Intrusion Detection System", Proceedings of World Academy of Science, Engineering and Technology, Vol. 6, Junes 2005