

PROACTIVE WORM DETECTION IN PEER TO PEER NETWORK USING DYNAMIC QUARANTINE WITH FEEDBACK

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

INTEGRATED DUAL DEGREE

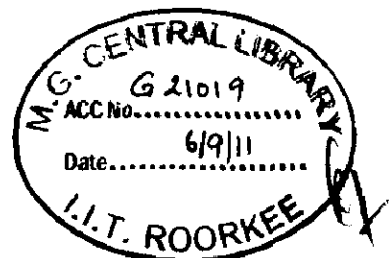
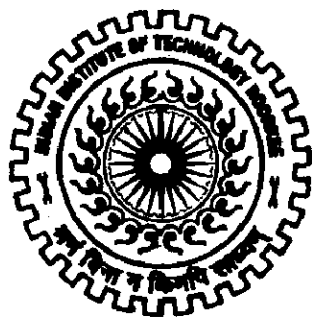
in

COMPUTER SCIENCE AND ENGINEERING

(With Specialization in Information Technology)

By

KAIVALYA YADAV



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2011**

CANDIDATE'S DECLARATION

I hereby declare that the work is being presented in the dissertation work entitled "**Proactive Worm Detection in P2P network using Dynamic Quarantine with feedback**" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May, 2010 to May, 2011 under the guidance and provision of **Dr. A K Sarje, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: May 2011

Place: Roorkee

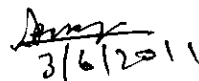

(KAIVALYA YADAV)

CERTIFICATE

This to certify that the work contained in the dissertation entitled "**Proactive Worm Detection in P2P network using Dynamic Quarantine with feedback**" by Kaivalya Yadav of Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology), has not been submitted elsewhere for a degree or diploma to the best of my knowledge.

Date: May,2011

Place: Roorkee


3/6/2011
Dr. A K Sarje
Professor,
E&CE Department
IIT Roorkee, India

ABSTRACT

Peer to Peer networks are responsible for large amount of internet traffic today. They started with the motive of content distribution and have now entered mission critical applications like Skype. But with the increase in usage they have also become vulnerable and an attractive target for attacks by internet worms.

In this dissertation we discuss several propagation models, attack techniques, detection methods and control strategies of internet worms. In most of the methods it is not possible to timely detect and put a halt on worm spread. Also many methods suffer from false alarms and require human countermeasure to remove infected host from a Peer to Peer Network. In this dissertation we use a dynamic quarantine model which is based on the principle “assume guilty before proven innocent”. To reduce false alarms we introduce a system for getting feedback in the dynamic quarantine model, which works by collecting behavioral information from the neighbors of an abnormally behaving node and comparing weighted Euclidean differences of behaviors with a threshold.

Simulation results show that our model is better than most of the pre existing models in terms of reducing the maximum number of infected hosts and reducing false alarms.

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. A K Sarje**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertation under him.

I also extend my sincere thanks to **Dr. S.N. Sinha**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I would like to thank all my friends who supported and encouraged me to finish this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

KAIVALYA YADAV

Contents

Candidate Declaration.....	i
Abstract.....	iii
Acknowledgment.....	v
Contents.....	vii
List of Figures and Tables.....	x
Chapter 1 Introduction.....	1
1.1 Overview.....	1
1.2 Motivation.....	3
1.3 Research Challenges.....	3
1.4 Problem Statement.....	5
1.5 Dissertation Organization.....	5
Chapter 2 Background and Literature Survey.....	7
2.1 Traditional worm propagation models.....	7
2.1.1 Simple Epidemic Model.....	8
2.1.2 General Epidemic Model: Kermack- Mckendrick epidemic model.....	8
2.2 Two-factor Worm Model.....	8
2.3 Dynamic Quarantine Model.....	9
2.4 Worm Attack Strategies.....	10
2.4.1 Worm Propagation via email.....	11
2.4.2 Network Worms.....	11
2.4.3 Web Searching Worms.....	11
2.4.4 Online Social Networking Worms.....	11
2.4.5 Peer to Peer Worms	13
2.5 Worm Detection Technique.....	13
2.5.1 Intelligent Connection Failure Algorithm.....	13
2.5.2 Distributed Framework of P2P network for worm detection.....	14
2.5.3 Dynamic quarantine model for P2P system.....	16
2.6 Shortcomings and Research Gaps.....	19

Chapter 3 Proposed Protocol for Proactive Worm Detection in P2P Network.....	20
3.1 Algorithm	20
3.2 Quarantine Protocol	22
3.3 Message Formats	23
Chapter 4 Implementation and Simulation Details	25
4.1 Implementation.....	25
4.2 Assumption.....	25
4.3 Worm Detection System.....	25
4.4 Simulation.....	26
4.4.1 Algorithm for simulation	27
Chapter 5 Results and Discussion	31
5.1 Simulation Parameters and State Transition Trends of Hosts	31
5.2 Comparison with no quarantine and no defense strategy.....	33
5.3 Effect of Degree of Node.....	34
5.4 Effect of Quarantine Time.....	35
Chapter 6 Conclusion and Future Work.....	36
References.....	37
Appendix.....	40

LIST OF FIGURES AND TABLES

Figure 2.1	Simple Epidemic Model.
Figure 2.2	Kermack-Mckendrick Epidemic Model.
Figure 2.3	Dynamic quarantine worm Model.
Figure 2.4	Koobaface Worm Propagation.
Figure 2.5	Intelligent Failure Connection Algorithm
Figure 2.6	The format of I message
Figure 2.7	The format of R message
Figure 3.1	Proposed Algorithm for Worm Detection on a host.
Figure 3.2	Format of proposed I message.
Figure 3.3	Format of R message.
Figure 3.4	Format of S message
Figure 3.5	Format of J message
Figure 4.1	Host stages at different times during simulation
Figure 4.2	Stage transitions of hosts
Figure 5.1	Plot of state transitions of hosts vs time
Figure 5.2	A comparison of different worm detection strategies
Figure 5.3	Effect of node degree on number of infectious hosts wrt time
Figure 5.4	Effect of quarantine time on number of infectious hosts wrt time
Table 4.1	Simulation Parameters used
Table 4.2	Various Notations used in the model
Table 5.1	Simulation Parameter and their values used for model simulation

INTRODUCTION

1.1 Overview

Millions of users employ Peer to Peer network for content distribution making P2P networks responsible for large amount of internet traffic. Due to the high scalability and reliability properties many mission-critical applications like Skype have also started using peer-to-peer protocols. The widely-deployed P2P systems used by end users, however, have strong security implications. They can be a potential vehicle for the attacker to achieve rapid worm propagation in the Internet.

Internet worm is an autonomous intrusion agent that is capable of infecting one computer system and using it, in an automated fashion, to infect another system. This cycle is then repeated and the population of worm infected hosts grows exponentially. The term 'worm' is derived from the word tapeworm, which is a parasitic organism that lives inside a host and saps its resources to maintain itself[1]. Most internet worms are malicious. They are designed to take control of computers they land on either to steal confidential user information or to convert them into remote-controlled 'zombie' [2]. Networks of these - 'zombies' - are then rented out by organized crime for sending spam email or attacking business and government computer systems.

Unlike viruses, which are computer programs that are designed to spread themselves from one file to another on a single computer, internet worms are insidious because they rely less (or not at all) on human behavior in order to spread themselves from one computer to another over a network. A virus might rapidly infect every application file on an individual computer, or slowly infect the documents on that computer, but it does not intentionally try to spread itself from that computer to other computers. In most cases, that's where humans come in. This means that computer worms spread much more rapidly than computer viruses. Furthermore, almost all viruses are attached to an executable file which means the virus may exist on computer but it actually cannot infect your computer unless you run or open the malicious program. Worms often infect computers by exploiting bugs in legitimate software. Typically, a high profile,

trusted web page may be tampered with so it transmits (often invisibly) a carefully corrupted document file to the user when the page is viewed. The corrupted file causes the viewer program to crash, opening a door for the injection of a malicious program. To help hide the infection, the malicious program is usually a 'downloader' - a very small program that later connects to a remote computer over the internet to download a more substantial piece of malicious software. Many Internet worms have also been Visual Basic script viruses[3] which replicate on Windows computers by interacting with the user's email program to send themselves to many (often all) of the addresses in the address book. Once on a new machine, they repeat the process with the new user's address book, quickly expanding the number of people reached.

A peer-to-peer (P2P) worm is a malicious code that makes use of a P2P network to spread from one machine another (Khat *et al.*, 2006)[5]. In principle, we cast all P2P worms into three categories:

1. passive P2P worms,
2. reactive P2P worms,
3. proactive P2P worms,

each different in the pattern of propagation.

Passive P2P worms copy themselves into the share folder of the P2P client and allure other users to download these copies and then complete propagation by running them in the peers' terminals (Thommes and Coates, 2006)[6]. Apparently, passive P2P worms cannot infect others without users' intervention. On the contrary, reactive and proactive P2P worms automatically propagate through common vulnerabilities of P2P clients (Chen and Gray, 2006)[7]. Reactive P2P worms infect only peers which are requesting files at that time while proactive P2P worms aim at infecting all vulnerable nodes as quickly as possible leveraging the cached neighbors' information. (Chen and Gray, 2006; Li *et al.*, 2009)[7]. Put simply, a proactive P2P worm is a more severe and fatal threat than passive and reactive P2P worms.

1.2 Motivation

The ability of attackers to rapidly gain control of vast numbers of Internet hosts through internet worms poses an immense risk to the overall security of the Internet. Once subverted, these hosts can not only be used to launch massive denial of service floods, but also to steal or corrupt great quantities of sensitive information, and confuse and disrupt use of the network as discussed below:

1. Distributed Denial of Service (DDOS) attacks could readily bring down e-commerce sites, news outlets, command and coordination infrastructure, specific routers, or the root name servers.
2. Illegal access to any sensitive information present on any of the million machines—passwords, credit card numbers, address books, archived email, patterns of user activity, etc can be misused for bank frauds, illegal money transfer or in the service of terrorism.
3. Not only access to this information, but attacker can sow confusion and disruption by corrupting the information, or sending out false or confidential information directly from a user's desktop. In short, if one could control a million Internet hosts, the potential damage is truly immense: on a scale where such an attack could play a significant role in warfare between nations or in the service of terrorism.

The severity of worm threat has rapidly grown with the increasing degree to which the Internet has become part of a nation's critical infrastructure, and the recent introduction of very large, very rapidly spreading Internet worms, such that this technique is likely to be particularly current in the minds of attackers.

1.3 Research Challenges

Worms continue to be challenging for the following four main reasons:

1. **Ease.** Automation cannot be beaten. Although the overhead associated with writing worm software is somewhat significant, worm continues to work while the developers are away. Due to its nature of propagation, growth is exponential as well.
2. **Penetration.** Due to the speed and aggressiveness of most worms, infection in some of the more difficult to penetrate networks can be achieved. An example of this would be an

affected laptop being brought inside a corporate network, exposing systems normally behind a firewall and protected from such threats. This usually happens through serendipity, but could, with some work, be programmed into the worm system.

3. **Persistence.** While it is easy to think that once the attack vectors of a worm are known and patches for the vulnerabilities are available, networks would immunize themselves against the worm, this has been proven otherwise [8]. Independent sources have shown that aggressive worms such as Code Red and Nimda have been persistent for longer than 8 months since their introduction date, despite well-known patches being available since the rise of these worms.
4. **Coverage.** Because worms act in a continual and aggressive fashion, they seek out and attack the weakest hosts on a network. As they spread through nearly all networks, they find nearly all of the weakest hosts accessible and begin their lifecycle anew on these systems. This then gives worms a broad base of installation from which to act, enabling their persistence on the Internet because they will have a continued base from which to attack for many months or even years.

P2P systems generally entail a large set of computers all running the same software. Each node in the P2P network is both a client and a server. Accordingly, the problem of finding a pair of exploits to infect both client and server for a worm is reduced to the problem of finding a *single* exploit, significantly less work for the attacker. P2P systems also have several other properties that make them well suited to worms. Some of them are following:

1. The P2P protocols are generally not viewed as mainstream and hence receive less attention in terms of monitoring by worm detection systems and analysis of implementation vulnerabilities.
2. The programs often execute on user's desktops rather than servers, and hence are more likely to have access to sensitive files such as passwords, credit card numbers, address books.
3. The use of the P2P network often entails the transfer of "grey" content (e.g., pirated music and videos), arguably making the P2P users less inclined to draw attention to any unusual behavior of the system that they perceive.

4. The final property of P2P networks responsible for making them worm target is their potentially immense size.

1.4 Problem Statement

The main objective of the present research work can be described by the statement of the problem expressed as follows:

“To develop a strategy to control proactive worm break out on a peer-to-peer (P2P) network using dynamic quarantine of suspicious host and infected hosts in a P2P system”

Proactive worm makes use of the information stored in the routing table of various hosts in a peer to peer network to identify its targets and are more dangerous than active and passive worms. We are required to devise a strategy based on dynamic quarantine model to decrease the number of infected hosts in a P2P network under proactive worm attack and as well as achieve decline in the worm propagation speed.

1.5 Dissertation Organization

Remaining dissertation is organized as follows:

Chapter 2 details the fundamentals and provides a literature review of the worm propagation models, detection techniques and control strategies including traditional epidemic models, two factor model and dynamic quarantine model. We also review anomaly detection based schemes to understand worm detection. Research gaps and shortcomings are identified and described.

Chapter 3 provides details about our proposed work which is a dynamic quarantine based algorithm to detect proactive worm. A host based intrusion detection system is used to collect data from sources internal to the computer at the OS level. The chapter details on the set of differential equations that can be derived from our model.

Chapter 4 discusses the assumptions made and implementation and simulation details of the proposed quarantine protocol. The chapter details on state transitions in our model.

In Chapter 5 we provides the results of our simulation and discuss state transition trends in hosts, effect of node degree on worm propagation, effect of quarantine interval on infected hosts and compare our model with other defense method.

Chapter 6 concludes the thesis with a summary of the contribution towards efficient worm detection in P2P networks. Possible horizon for future work is also discussed.

BACKGROUND AND LITERATURE SURVEY

A study of Internet worm propagation models gives insight into worm behavior, identifies the weakness in the worm spreading chain and provides accurate prediction for the purpose of damage assessment for a new worm threat .Since Morris Worm attack in 1988 [8], internet worm has been an active field of research and researchers have proposed different modeling, analysis and control strategies of worm propagation from time to time.

Many behavior-based anomaly detection methods have been developed for worm detection in P2P systems which have been potential points for attackers to achieve rapid worm propagation in the Internet. This is mainly because of two reasons: firstly there are a large number of homogenous users and secondly in a P2P network worm propagation accelerates because worms are very efficient in searching for targets using neighbor's information.

Traditional worm propagation models do not take the P2P network into account. However, the bulk of the models are based on these traditional models. Some of the Internet worm propagation models discussed in this dissertation are: the classical simple epidemic model (SEM)[9], Kermack-McKendrick (KM) model[10], two factor worm model[11], and a dynamic quarantine model before the quarantine model by Wei[12] which is introduced later.

2.1 Traditional worm propagation models

In epidemiological modeling, the hosts that are vulnerable to infection are *susceptible hosts*, hosts that are infected and can infect other hosts are *infectious hosts* and the hosts that are immune or dead such that they can not be infected are called *removed hosts* no matter they have been infected before or not.

2.1.1 Simple Epidemic Model

Simple epidemic model[9] assumes that population is large and each host stays in one of the two states: susceptible and infectious. Thus this model assumes that once a host is infected it stays in infectious state forever. Also it can be very easily seen that there exists only one possible state transition:

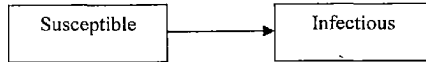


Figure 2.1 Simple Epidemic Model

2.1.2 General Epidemic Model: Kermack-Mckendrick epidemic model

Kermack-Mckendrick model[10] considers removal of infectious host. Thus in this model at any time t , the host stays in any one of the three states: *susceptible state*, *infectious state* or *removed state*.

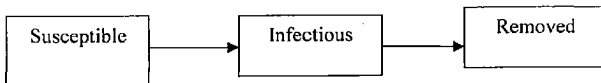


Figure 2.2 Kermack-Mckendrick Epidemic Model

The hosts are in *removed state* after they recover or die from the disease.

The Kermack-Mckendrick model is an improvement over the classical simple epidemic model because it considers that some infectious hosts either recover or die after some time. However, it is still not suitable for internet propagation because first in internet cleaning, patching and filtering counter measures against worms will remove both susceptible and infectious hosts from circulation and secondly it assumes that the infection rate is constant which isn't true for rampantly spreading worm.

2.2 Two-factor Worm Model

Zou Gong Towsley *et al* [11] derived a general Internet worm model based on the classical epidemic Kermack-Mckendrick model and called it to be *two-factor worm model*. In this model worms that are continuously activated and without topology constraint were considered. A

continuously activated worm is a worm on an infectious host that continuously tries to find and infect other susceptible hosts. Topology constraint means that an infectious host may not be able to directly reach and infect an arbitrary susceptible host – it needs to infect several hosts on the route to target before it can reach the target.

The following two factors were found that were not considered in traditional epidemic model

- Human countermeasures result in removing both susceptible and infectious computers from circulation
- Decreased infection rate $B(t)$ and not a constant rate B --- the large scale worm propagation have caused congestion and troubles to some internet routers ,thus slowing down the scanning process.

As the bandwidth of Internet connections increased it was observed that worms require very less time to finish the infection task. For such fast spreading worms human's manual countermeasures can not catch up with the worm's propagation speed. This was the major limitation of this model.

2.3 Dynamic Quarantine Model

Zou et al [14] presented a soft dynamic quarantine method based on the principle “*assume guilty before proven innocent.*” where every host of the system can be quarantined individually when the worm anomaly detection program raises alarm for this host; the alarmed host is released after a quarantine time T , even if the host has not been inspected by the security staff yet. Once the quarantine on a host is released, this host can be quarantined again if the anomaly detection program raises alarm for this host again some time later.

If the service port with suspicious activities can be determined then the quarantine will block the traffic on this port without interrupting normal connections on other ports. Thus this method had two advantages: first a falsely quarantined healthy host will only be quarantined for a short time, thus its normal activities will not be interfered too much; second, it was now possible to tolerate

higher false alarm rate than normal permanent quarantine, and the worm anomaly detection program can be set to be more sensitive to a worm's activity. All hosts in this model belong to one of the following states: susceptible, infected, quarantine, or removed as shown in Fig.2.3

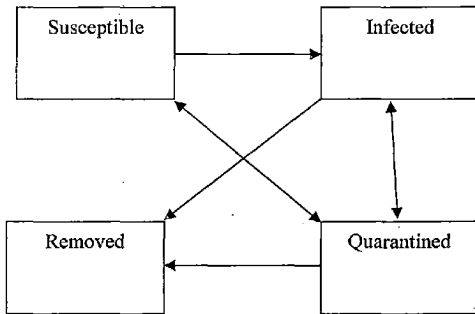


Figure 2.3 Dynamic quarantine worm model

Dynamic quarantine method can thus alleviate the negative impact of false alarms and is very effective for controlling worm.

2.4 Worm Attack Strategies

Elias Levy et al [15] has discussed different generic attack strategies used by internet worms. The defining task of worm is to locate new targets to attack. Viruses search for files in a computer system to which to attach, whereas worms search for new targets to which to transmit themselves. Over a period of time attackers have developed different strategies for worm propagation

2.4.1 Worm Propagation via email

These types of worms are very successful in propagating themselves by using the victim's e mail address book or searching through mailbox to find next potential targets to attack. The addresses in a user's mailbox are almost certain to be valid, and thus user's social web is hijacked and his trust relationship is exploited by worm. In most cases the worm will craft its own message to send to target, but some will wait for the user to send a message and attach themselves to it.

2.4.2 Network Worms

Network worms function to attack network services and they must determine their next victim's IP address by randomly generating IP addresses. Their techniques of propagation are however not very efficient because of two reasons:-

1. IPv4 address space is reasonably large
2. IP address space is sparsely populated, with host clustering in some areas, while the rest remain unpopulated.

Later, authors of network worms started using several strategies to effectively determine local addresses, for instance Code Red II displayed bias towards local addresses i.e. those near the current computer in the IP address space—which usually implies closeness in the network topology. The reasoning behind this is that there's likely to be other machines near the infected machine, and furthermore, they're likely to be running similar software.

2.4.3 Web Searching Worms

Worms can make use of a search engine to quickly and easily send malicious code to new targets. But the malware should also develop some way to randomize the order of results because the results returned by a search engine are usually static for a given query in a short time period; otherwise malware will attack the same targets in the same order, thereby wasting resources.

2.4.4 Online Social Networking Worm

Xu Zhang and Zhu et al [16] suggested online social networking (OSN) websites like Facebook have become an attractive target for worms (hereinafter referred to as *OSN worms*). Social networks are small-world networks, which means they have the properties of small average shortest path length and high clustering. The small average shortest path length property can reduce the propagation time from one user account to another user account. Meanwhile, the high clustering property suggests that users are tightly connected together, which facilitates the explosion of OSN worms. Second, online social networks are also scale-free networks, where high-degree nodes tend to connect with other high-degree nodes. When an OSN worm infects the

account of a popular user (i.e., user with a large number of friends), this scale-free property suggests that the worm can infect another popular account shortly. As a result, the worm can achieve exponential growth by propagating to the large friends set of these popular users. Moreover, OSN worms also leverage social engineering to increase the authenticity of worm messages.

Figure 2.4 illustrates the propagation flow of Koobface in Facebook. User *A* receives a worm message from one of her friends (step 1) after this friend was infected by Koobface. Within this worm message, there is a link to a video clip hosted on a fake YouTube website. When user *A* clicks that link, the browser is redirected to the fake YouTube webpage (step 2), where the user is prompted by a request to install an update for Adobe Flash player" plugin, which is actually a malware. After user *A* installs the claimed browser plugin (step 3), Koobface infects user *A*'s Facebook account and iterate its infection cycle by sending similar worm messages to all the friends in user *A*'s profile (step 4). Actually, besides sending messages, Koobface also sends invitations or composes posts, both of which contain similar worm content.

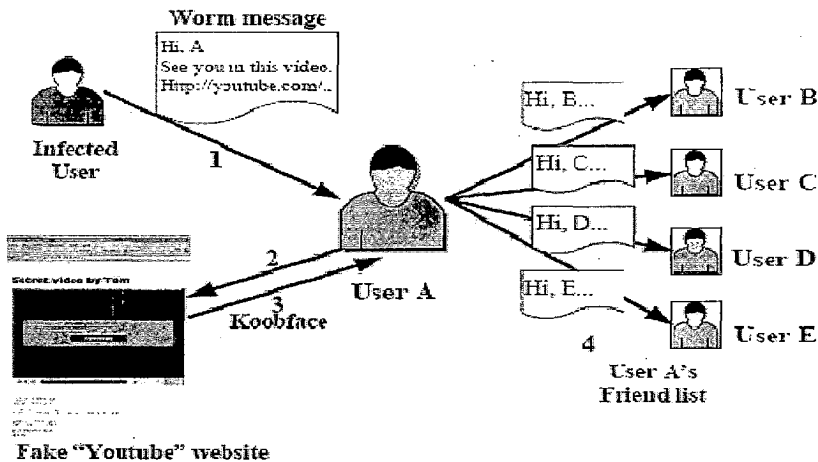


Figure 2.4 Koobface Worm Propagation

2.4.5 Peer to Peer Worms

Peer-to-peer worms are faster than other types of worms. This is because of three main factors. Firstly, the users of a P2P application must run the peer program on their terminal and practical experience shows that most of the users run the same software. So, a vulnerability in this software could potentially allow a worm to infect millions of hosts. Secondly, by making use of the overlay topology of the P2P system, P2P worms do not waste time probing unused IP addresses. Thirdly, they do not generate high rates of failed connections and they can blend into the normal traffic patterns of the P2P network. According to its propagation strategy, a P2P worm can belong to one of the following two classes:

Topological scan based P2P worms: in this case, the worm uses information about victim's neighbors in the P2P system to spread through it by creating a hit list.

Passive P2P worms: In this case, the worm does not search for targets, but waits for them. The worm resides in the shared folder of the infected machine, under several names. When another peer downloads one of those files, the worm spreads to this host, and when the user runs the file, the worm duplicates itself under several attractive names in the shared folder of the new victim, and waits for other victims, and so on.

2.5 Worm Detection Techniques

In order to detect an unknown worm, a straightforward way is to use various threshold-based anomaly detection methods to detect the presence of a worm. We can directly use some well-studied methods established in the anomaly intrusion detection area.

2.5.1 Intelligent Connection Failure Algorithm

Rasheed et al[17] proposed an intelligent early system detection mechanism for detecting internet worm. The average of failure connections by using Artificial Immune System (AIS) was the main factor on which the technique depended on in detecting the worm. Intelligent Failure Connection Algorithm (IFCA) is based upon the difference between regular connection and worm connection. The worm scans different IP addresses every second. IFCA depends on the TCP failure and ICMP unreachable connection on different random addresses. Therefore, there will be a large number of failures connections if the computer has worm. If normal connection is

received, i.e. TCP SYN/ACK, “counter” will be decreased. Only the first failed connection sent from the forged source IP address to different destination IP address is recorded. Normal network activities are considered to decrease the counter value. IFCA will remove the “counter” every three days. The packet should be ignored when the destination IP is recorded into the counter table. Once detecting the first failed connection packets, the algorithm then extracts (the source address, source port, destination address, destination port) from the packet and creates the record. Whenever the counter value does not exceed the threshold during time cumulative computation phrase, the traffic sent from the corresponding IP address would be forwarded as normal activity (see figure 2.5).

2.5.2 Distributed Framework of P2P network for worm detection

Adeel et al[18] proposed a distributed framework for passive worm detection and throttling in P2P network. The framework is aimed at sustaining the distributed behavior of the network as all the nodes act together to detect the threat as early as possible. In his model nodes would perform a purpose-specific functionality, analyzing the traffic at first instance. The phases of implementation was divided into detection, analysis & confirmation, patch selection and finally the patch propagation.

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 analyze the traffic patterns and to identify any malicious behavior.

B. Analysis & Confirmation of Threat

In this phase if a worm threat is confirmed by guardian node by looking at various worms definitions, it will generate an alert to the entire P2P network. This alert will have different meanings for the peers and other guardian nodes in the network. The guardian nodes will get the patch ready and they can simply push the patch to other devices or wait for this patch to be pulled by the devices.

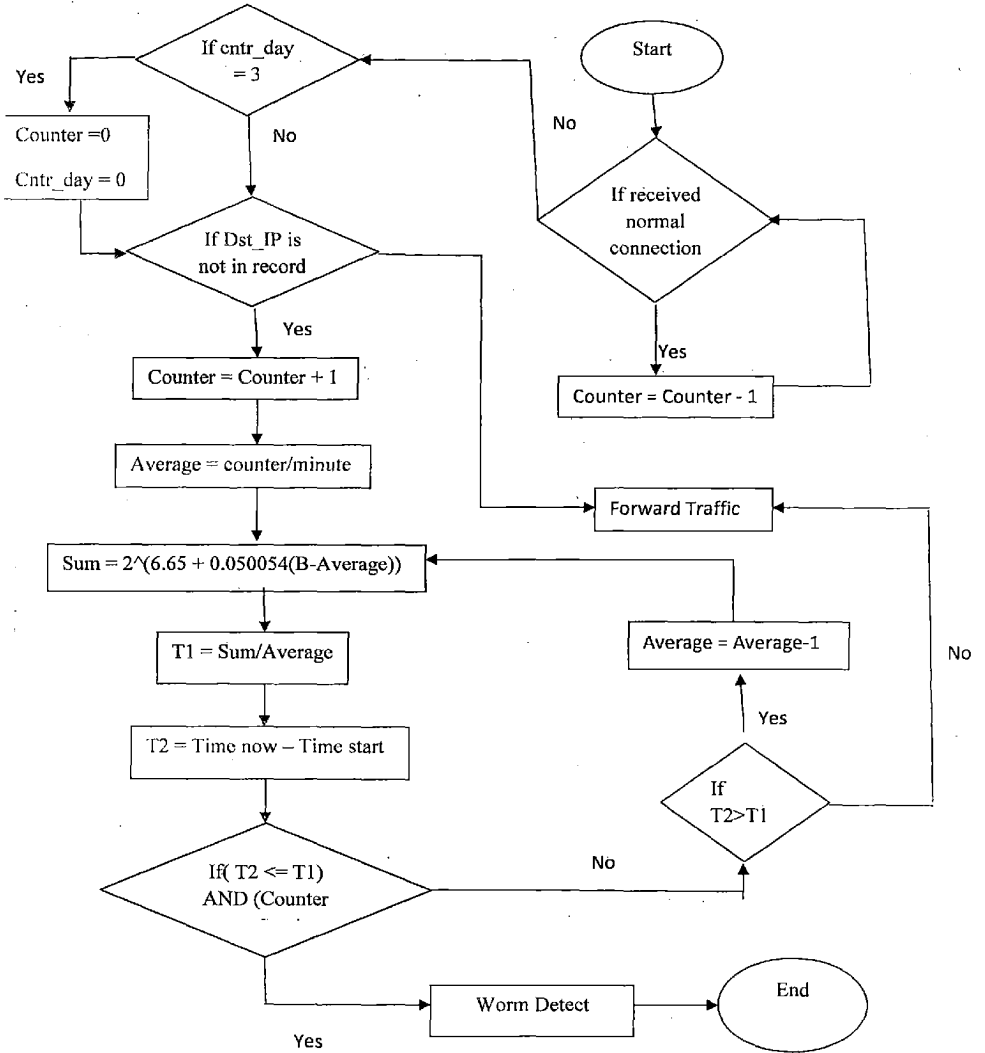


Figure 2.5 Intelligent Failure Connection Algorithm

C. Patch Selection

Selection of a proper patch from the patch reservoir is a key task when for the worm throttling process.

D. Patch Propagation

When the patch is ready, it can either be propagated straightaway to the peers or the guardian node will 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 will also announce the identity of the worm so that the peers that may already have the patch can start taking care of the worm. The guardian nodes receiving the alert will make the patch available in their shared folders or even reactively flood the patch into the network.

2.5.3 Dynamic quarantine model for P2P system

Various worm detection methods have been proposed, two major variants of which are namely host based and network based approaches. Network based approach monitor packet by setting network interface to promiscuous mode and analyze network traffic. But they suffer from scalability problems in case of high network load and have problems with encrypted communication. Host based system collect high quality data directly at the OS level. For a P2P system, adjacent hosts can not only be far away from each other but need not be in the same network as well. So the host based worm detection program is more suitable for dynamic quarantine defense.

By Quarantine we mean that we need to block traffic only on the suspicious port without interfering connections on other port. But the P2P worms use P2P application to propagate themselves and thus the P2P service is also blocked once the port is blocked. So the quarantined hosts need to inform their neighbors about their leave from the network. The neighbors also need to identify the quarantined host and emend their routing tables.

Wei et al[12] discusses a quarantine protocol for P2P networks based on hosts based detection method. The quarantine protocol is as follows:

Quarantine Protocol

1. node i be a host in the P2P system and its neighbors' set be N
2. **If** node i is detected presenting a suspicious behavior
 While N is not empty **do**
 Select a neighbor set V from N
 The host sends an I message to the neighbor and waits for an R message
 If gets an R message
 $N=N-V$
 end while
 do Quarantine operation
3. **If** node i gets an I message from its neighbor node j to inform it about a suspicious behavior
 node i sends an R message
 $N=N-Node\ j$
 While N is not empty **do**
 Select a neighbor set V from N
 The host sends an I message to the neighbor and waits for an R message
 If gets an R message
 $N=N-V$
 end while
 do Quarantine operation
4. **If** node i has got an I message that informs it that neighbor node j is detected by the worm
 detection presenting a suspicious behavior
 node i sends an R message
 The host sends an I message to the node j and
 Waits for an R message
 do Emend Routing Table operation
5. **If** the neighbor of node i has got an I message that informs it that the neighbor of node j is
 detected by the worm detection presenting a suspicious behavior
 node i sends an R message
 do Emend Routing Table operation

The formats of the messages are following

1. I message

SID	DID	QID	P2PID	START	QTIME
-----	-----	-----	-------	-------	-------

Figure 2.6. The format of I message

I message is used to inform a host that the host has the suspicious behavior. The format of *I* message is shown in Figure. 2.6. **SID (Source node ID)** field is used in the P2P system to identify the host sending the message. **DID (Destination node ID)** field is used in the P2P system to identify the host receiving the message. **QID** field is used to identify the host who has the suspicious behavior. When QID field equals SID, it means the host with SID needs to be isolated. Otherwise, when QID field equals DID, it means that the host with DID needs to be quarantined. **P2PID** field is a port number which is used to identify P2P service which is blocked. **START** field is a timestamp to record the start time of quarantine. **QTIME** field is used to record the quarantine time.

2. R message

R message is response to a host. It is an acknowledgment message. The format of *R* message is shown in Fig.2.7

SID	DID	ACK
-----	-----	-----

Figure.2.7. The format of R message

SID field and DID field is the same meaning as in the *I* message. ACK field means that the host in the P2P system has got the *I* message.

2.6 Shortcomings and research gaps

1. Most of the worm detection strategies make use of various threshold-based anomaly detection methods to detect the presence of a worm. However, many threshold based anomaly detections have the trouble to deal with their high false alarm rate.
2. Traditional ways based on firewalls, IDS, honey pots and so on, are limited to one host or a small group of hosts, while worms' propagation in P2P networks can cover a wide range of hosts in short time
3. The dynamic quarantine protocol proposed by Wei et al considers the dynamic process of peer joining and leaving. But it does not consider any transition from quarantine state to susceptible state. Thus it suffers from high false alarm rate.

In this dissertation we do not use threshold-based anomaly detection methods alone. Instead, we fully exploit a worm's simple behavior based on well-studied dynamic quarantine models. We also take advantage of the distributed feature of P2P network to make a better analysis of dubious network traffic. If the adjacent hosts show the same abnormal behaviors, there is greater probability of worm propagation. We can take into account the similarity of the abnormal behaviors between adjacent hosts to judge whether a worm is actually propagating or not.

PROPOSED PROTOCOL FOR PROACTIVE WORM DETECTION IN P2P NETWORK

3.1 Algorithm

Dynamic quarantine method based on the principle “assume guilty before proven innocent” is used to find and control worm break out. In addition to it a host based detection system is used to collect data from sources internal to the computer at the OS level. We know that worm in an infectious computer in a P2P network tries to infect adjacent nodes. The peers whose original address is contiguous especially those in the same LAN constitute a peer group. A ‘super node’ is in charge of the analysis of suspicious information exchanged in the peer group.

Super node collects abnormal behavior vector within its peer group and gives notification about worm attack based on the principle “If adjacent hosts show the same abnormal behavior there is a greater possibility of worm propagation”. The abnormal behavior vector is given by equation 1. We take the similarity of abnormal behaviors between adjacent hosts into consideration to reduce false alarms.

The abnormal behavior can be detected using the vector given below:

$$(B_{pm}[1], B_{pm}[2], B_{pm}[3], \dots, B_{pm}[n]), \text{ --- (1)}$$

Where,

$B[i]$ denotes a parameter i such as the increase rate of source addresses etc,

m denotes a host in peer group,

P denotes a peer group in P2P network,.

The similarity of vectors can be calculated using Euclidean distance formula

$$d(P_m, P_{m+1}) = (w_1|B_{pm}[1] - B_{pm+1}[1]|^2 + \dots + w_n|B_{pm}[n] - B_{pm+1}[n]|^2)^{1/2} \text{ --- (2)}$$

where,

m_i 's denote a host in peer group

P denotes a peer group

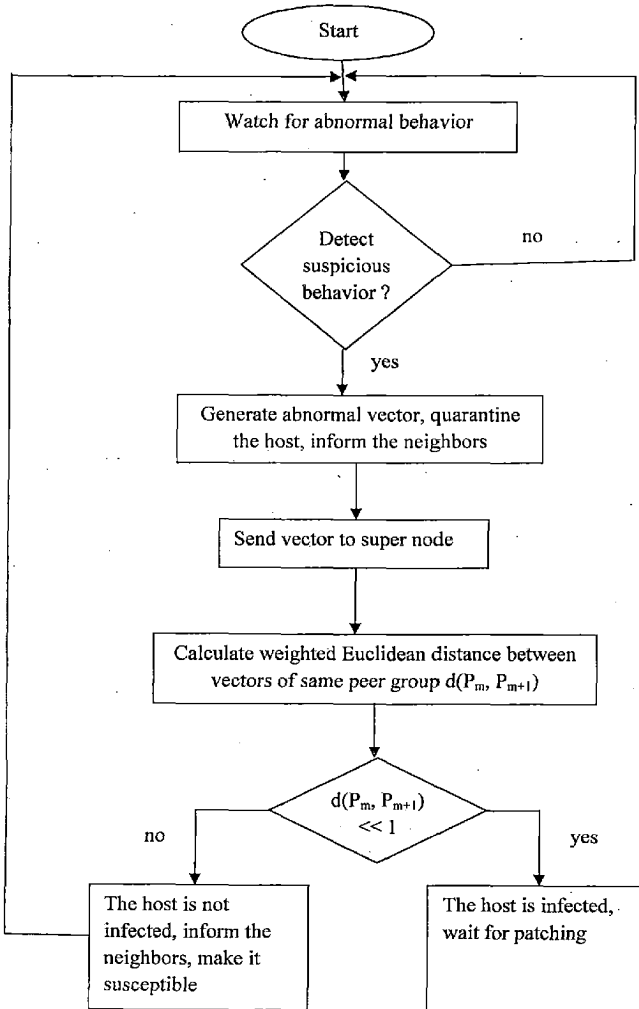


Figure 3.1 Proposed Algorithm for Worm Detection on a host

if $d(P_m, P_{m+1}) \ll 1$ super node gives alarm about some suspicious activity. Once super node gives alarm, we block traffic on the suspicious port i.e. quarantine the suspicious hosts in a peer group without interfering normal connections on other ports. Figure 3.1 shows the algorithm. Since the P2P worm uses P2P application to propagate themselves, P2P service is also blocked once the port is blocked. Thus, the network also needs to identify the quarantine hosts and inform the neighbors.

3.2 Quarantine Protocol

Now we will discuss the proposed Quarantine Protocol, which is an application layer protocol and uses UDP to send data. The Quarantine protocol is composed of four basic mechanism:

- Inform Hosts
- Inform Super Node
- Emend Routing Table
- Quarantine the service port

The proposed protocol is as follows:

1. Once a host is detected presenting a suspicious behavior, the host will send an **I** message to all its neighbors and do Quarantine operation and will send an **S** message to the super node containing abnormal vector.
2. Once a host has got the message from its neighbor to inform it of a suspicious behavior. The host will send an **R** message to its neighbor and also send **I** messages to the other neighbors and do Quarantine operation i.e. emend routing table.
3. Once a host has got the message that informs it that its neighbor is detected presenting a suspicious behavior. The host will send an **R** message and also send **I** messages to the neighbor. Finally the host will emend the routing table of P2P systems making the neighbor's state block.
4. Once the neighbor of a node has got the message that informs it that the neighbor of the other node is detected presenting a suspicious behavior. Then the node just emends the routing table of P2P systems.

5. Once a super node has got an **S** message from a host, it checks the weighted Euclidean difference of abnormal vectors from hosts belonging to the same peer group and compares it to a threshold value. The super node then sends a **J** message to its neighbors.
6. Once the neighbor of a node has got a **J** message, it checks the flag bit to decide whether the quarantined host is actually infected or not and emends the routing table of P2P systems accordingly.

3.3 Message Formats

4 kinds of message are designed for different purposes. The formats of the messages are as follows:

1. **I** message

I message is used to inform a host that the host has suspicious behavior. The format of **I** message is shown in Figure 3.2

SID	DID	QID	P2PID	START	QTIME	SNID
-----	-----	-----	-------	-------	-------	------

Figure 3.2 Format of proposed **I** message

SID (Source node ID) field is used in the P2P system to identify the host sending the message. **DID (Destination node ID)** field is used in the P2P system to identify the host receiving the message. **QID** field is used to identify the host who has the suspicious behavior. When **QID** field equals **SID**, it means the host with **SID** needs to be isolated. Otherwise, when **QID** field equals **DID**, it means that the host with **DID** needs to be quarantined. **P2PID** field is a port number which is used to identify P2P service which is blocked. **START** field is a timestamp to record the start time of quarantine. **QTIME** field is used to record the quarantine time. **SNID** field is used to identify the super node in peer group.

2. R message

R message is response to a host. It is an acknowledgment message. The format of **R** message is shown in Figure 3.3.

SID	DID	ACK
-----	-----	-----

Figure.3.3 Format of R message

SID field and DID field is the same meaning as in the **I** message. ACK field means that the host in the P2P system has got the **I** message.

3. S message

S message is used to send the abnormal vector to super node. The format of S message is shown in Figure 3.4

SID	SNID	QID	P2PID	START	QTIME	PID
Abnormal Vector						

Figure 3.4 Format of S message

SID, SNID, QID, P2PID, START and QTIME field have the same meaning as in the **I** message. PID is peer group ID. Abnormal vector field contains the abnormal vector that is being sent to the super node.

4. J message

J message is the response of super node. The format of J message is shown in Figure 3.5

SID	DID	QID	Flag bit (1/0)
-----	-----	-----	----------------

Figure 3.5 Format of J message

SID, DID, and QID field have the same meaning as in the **I** message. The flag bit is used to indicate whether the raised alarm is true or not. If flag bit is 0, it means that QID is safe and not infected, if flag bit is 1, it means QID is infected.

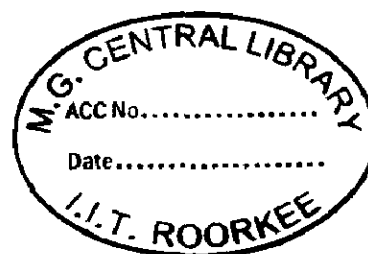
IMPLEMENTATION AND SIMULATION DETAILS

4.1 Implementation

The proposed Algorithm is simulated in C language for the consideration of simulation speed. C is very efficient and widely used for Unix network programming as well. Its modularity and object oriented nature gives the freedom to individually build and test modules. For simulation I have not considered packet-level events. I have recorded the simulation results and used MS Excel to draw plots.

4.2 Assumptions

1. Once a host is infected , it tries to infect all its neighbors
2. Effect of worm size is ignored
3. Effect of network distance is ignored
4. Effect of network bandwidth is ignored
5. Time duration to infect other hosts is set to one unit time
6. P2P worm propagation is a discreet event process
7. Congestion by network traffic is ignored.



4.3 Worm detection system

The following are the characteristics of Worm Detection System running on every host:-

- It is a daemon process which is responsible for monitoring the computer on which it runs and collecting any information related to probable security attack.
- It detects any recently identified security attack to the computer it is running on.
- At regular time intervals, it records the number of hits (h_i^n) the node received over the past interval. It calculates and transmit percentage p_i^n by which it differs from average hits in k intervals

- $p_t^n = (h_t^n - \sum h_i^n) / \sum h_i^n$
where I varies from t-k to t-1
- If $p_t^n \gg 1 \Rightarrow$ node n is under security attack

P_t^n is used to construct abnormal behavior vector of the host.

4.4 Simulation

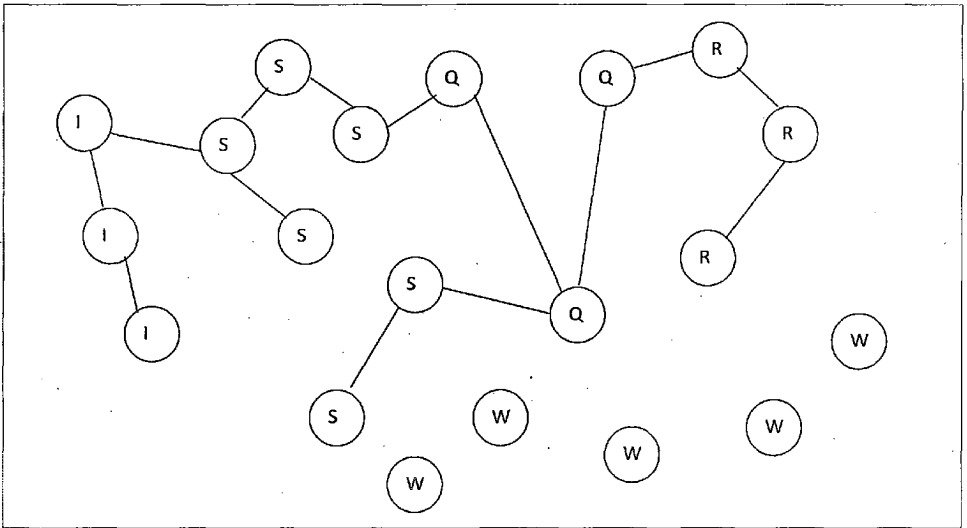


Figure 4.1 Host stages at different times during simulation

All hosts belong to a closed space represented by an undirected graph $\langle V, E \rangle$ with nodes and links between the nodes. The undirected graph is represented by an adjacency matrix. The vertex set $V = \{V_1, V_2, V_3, \dots, V_n\}$ denotes a host in the P2P system. The total number of hosts is denoted by N , which is a constant independent of time.

The simulation parameters used are shown in table 4.1

Table 4.1

Simulation Parameters	
Total Number of Nodes, N	Probability of false alarm, X_0
Initial number of Susceptible Nodes, S_0	Node degree, d
Initial number of Infected Nodes, I_0	Probability of host joining P2P network, l
Initial number of Quarantined Nodes, Q_0	Probability of host patching, P_p
Quarantine Time, Q_t	Probability of host leaving, P_l

4.4.1 Algorithm for simulation:

We maintain separate linked list for holding hosts in different state. Initially all the active nodes are in susceptible state and inactive hosts are in waiting state. A host in waiting state can make transition only to susceptible state with a probability of 0.5. One host is randomly chosen and is considered an Infected node and is inserted into a linked list for newly infected nodes (`newly_infected_list`). There is a small probability of host obtaining patching or leaving P2P network thus making transition to removed state and waiting state respectively which is provided as simulation parameter. The transition from susceptible to quarantine host takes place with a very small probability which is also provided as a simulation parameter and is possible only when at least one of the adjacent neighbor is infected. The Simulation works as follows:-

At each step do the following

- Randomly select a node and find its current state
- Use the nodes present in `new_infected_list` Linked List to infect closest neighbors equal to the user specified degree of P2P network(3 in our case) .
- Enter the nodes from `new_infected_list` Linked List into `infect_list` Linked List. The `new_infected_list` contains the newly infected nodes during the present step.

The randomly chosen host can potentially be in any of the following stages at any time t :

(1) S (susceptible): All hosts in this stage are vulnerable to P2P worm infection, and can acquire the worm infection when contacting with an infected host.

(2) I (infected): All hosts in this stage has been infected by malicious P2P worms and can propagate the infection to other hosts.

(3) Q (quarantined): All hosts in this stage have exhibited suspicious behavior and consequently, have been quarantined.

(4) R (recovery): All hosts in this stage have been gained immunity and are no longer infectious.

(5) W (waiting): All hosts in this stage have been left the P2P systems or waiting to join the P2P system for the next time tick.

These hosts may change their stages according to their own characteristics as shown in Fig. 4.2

The waiting host can only go into stage S.

The stages of susceptible hosts may transform in the following cases:-

- 1) If the host can obtain immunity via patching and it will go directly into stage R
- 2) If the host exhibits suspicious behavior it will go into stage Q with a very small probability only when at least one of the neighbor is infected.
- 3) If the host communicates with the infected host it will go into stage I
- 4) If the host leaves the P2P system, it will go into stage W.

The stages of quarantine hosts may transform in the following cases:-

- 1) If the host can obtain immunity via patching it will go directly into stage R.
- 2) If the host leaves the P2P system it will go into stage W.
- 3) If the flag bit is not set in the J message then the host will go into stage S which in our simulation is established using a small probability for false alarm.

The stages of infected hosts may transform in the following cases:-

- 1) If the host can obtain immunity via patching it will go directly into stage R
- 2) If the host leaves the P2P system it will go into stage W.
- 3) Else it will be quarantined and go into stage Q

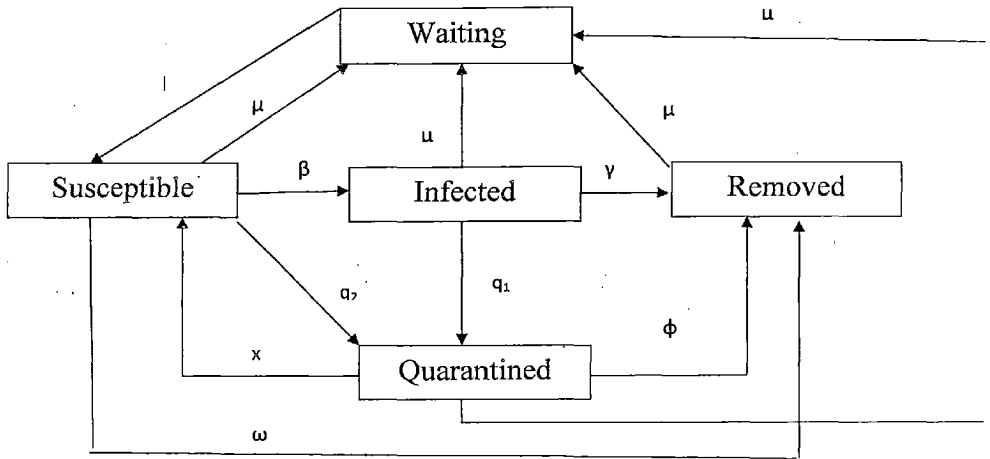


Figure.4.2 State transitions of hosts

Table 4.2 shows various notations used in the model:-

Table 4.2 Various Notations used in the model

$S(t)$	The number of susceptible hosts at time t	ϵ	Mortality rate due to P2P worm
$I(t)$	The number of infected hosts at time t	l	Probability of hosts joining P2P network
$Q(t)$	The number of quarantine hosts at time t	d	Average degree of hosts
$R(t)$	The number of removed hosts at time t	x	Recovery probability of quarantine host susceptible host.
$W(t)$	The number of wait hosts at time t	λ_1	The quarantine rate of infected peers
N	The total number of hosts	λ_2	The quarantine rate of susceptible peers
β	Number of hosts which can be attacked by p2p worm at the same time	q_2	Effective quarantine probability of susceptible
γ	Recovery probability of infected host by deleting worm	ϕ	Recovery probability of quarantine host patch
ω	The recovery rate at which susceptible hosts are treated	q_1	Effective quarantine probability of infected h
μ	The probability of host leaving P2P network	$E(t)$	The number of newly infected hosts from the susceptible stage to the infected stage at time $E(t)$

Our model can be represented by equations given below, the proof is similar to as given by Weif[12] and is briefly discussed in Appendix A :-

$$5. \quad dS/dt = -E(t) + IW(t) + xQ(t) - (\omega + \mu + q_2)S(t) \quad \text{----- (3)}$$

$$6. \quad dI/dt = E(t) - (\mu + q_1 + \gamma + \varepsilon) I(t) \quad \text{----- (4)}$$

$$7. \quad dR/dt = \omega S(t) + \phi Q(t) + \gamma I(t) - \mu R(t) \quad \text{----- (5)}$$

$$8. \quad dQ/dt = q_2 S(t) + q_1 I(t) - (\mu + \phi + x)Q(t) \quad \text{----- (6)}$$

$$9. \quad dW/dt = \mu(S(t) + I(t) + R(t) + Q(t)) + \varepsilon I(t) - IW(t) \quad \text{----- (7)}$$

$$10. \quad E(t) = S(t)[1 - (1 - 1/(N - W(t)))^{dI(t)}] \quad \text{----- (8)}$$

$$11. \quad S(t) + I(t) + R(t) + Q(t) + W(t) = N \quad \text{----- (9)}$$

The change in the number of infected hosts $I(t)$ from time t to time $t + \Delta t$ in time is represented by equation (8):

$$I(t + \Delta t) - I(t) = S(t)[1 - (1 - 1/(N - W(t)))^{dI(t)}] - (\mu + q_1 + \gamma + \varepsilon) I(t) \quad \text{----- (10)}$$

Hence

$$dI(t) / dt = S(t)[1 - (1/(N - W(t)))^{dI(t)}] - (\mu + q_1 + \gamma + \varepsilon)I(t) \quad \text{----- (11)}$$

RESULTS AND DISCUSSION

The following section contains the performance of proposed algorithm.

5.1 Simulation Parameters and State transition trends of hosts

Simulation results show that the worm propagation in our model is depressive as shown in Figure 5.1. The Simulation Parameters used are shown in Table 5.1

Table 5.1

Simulation Parameters and values	
Total Number of Nodes	1500
Initial number of Susceptible Nodes, S_0	1450
Initial number of Infected Nodes, I_0	1
Initial number of Quarantined Nodes, Q_0	0
Initial number of Waiting Nodes, W_0	50
Node degree, d	3
Probability of host joining P2P network	0.5
Probability of host patching	0.00002
Probability of host leaving	0.0001

Initially all the connected nodes are in susceptible state as shown in Figure 5.1. In the beginning the number of infected nodes $I(t)$ increases exponentially as most of the nodes are neither patched nor quarantined and therefore can be immediately infected by neighbor. The susceptible nodes thus become infected quickly and very few make transition to quarantine state directly. There are also small possibilities for a susceptible host to leave the P2P network or get patched and thus make transition to waiting or removed state respectively. This explains the shape of susceptible curve, which shows a sharp decrease in number of hosts in the beginning, and a very slow change in the number of hosts later. But unlike PWPQ model[14] where at the end of simulation, number of susceptible nodes was negligible (almost zero) in comparison to beginning because there was no transition to susceptible state other than that from waiting state, in our model since there is a possibility of transition to susceptible from quarantine state as well, we are

left with few susceptible hosts at the end. Thus in the end the hosts are in susceptible, quarantined or removed state but not in infected state.

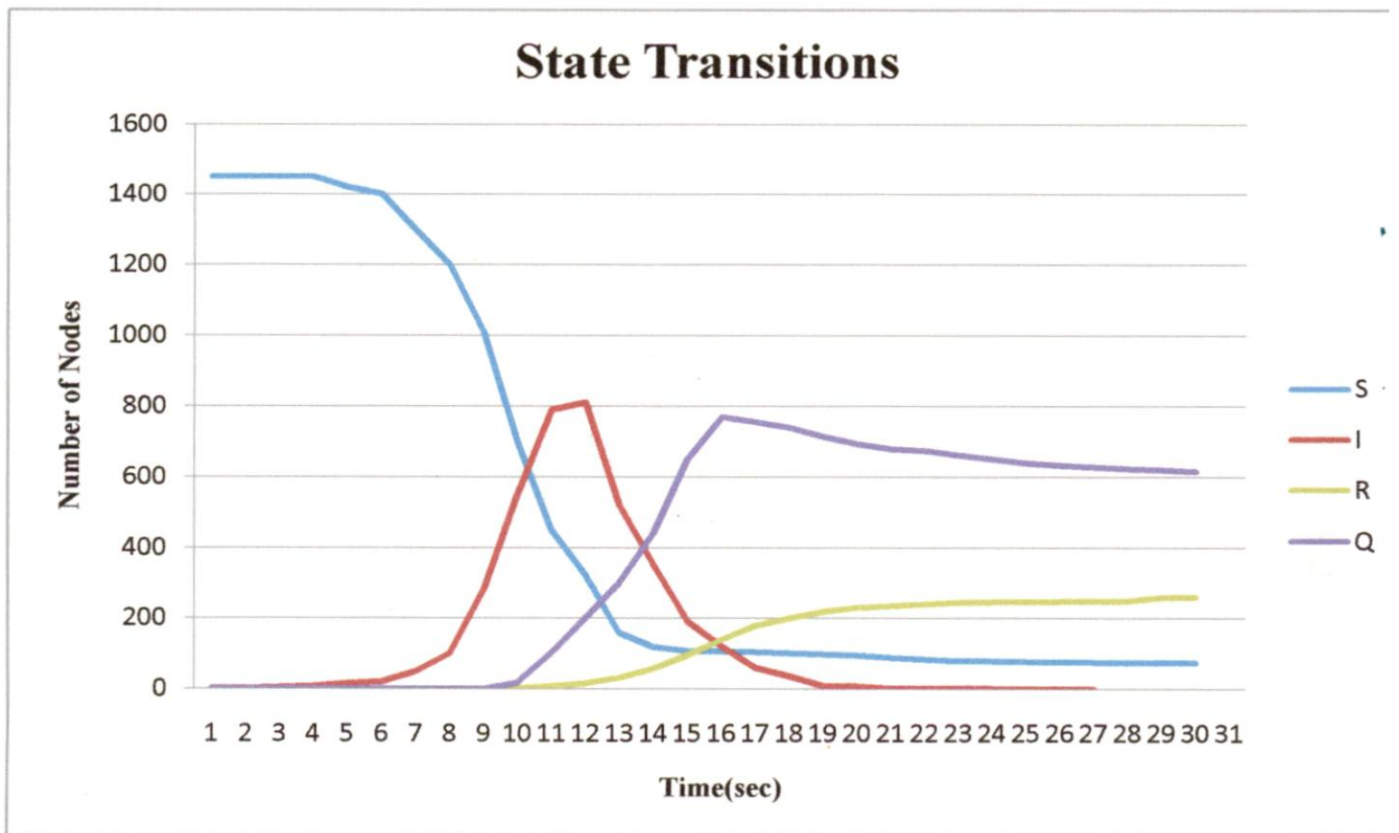


Figure 5.1 Plot of state transitions of hosts vs time

At $t = 0$, number of infected nodes = 1, the growth shows an exponential increase in the beginning, but after some time the infected nodes are picked quickly, get quarantined, and henceforth either wait for patching or leaving the P2P network and therefore decrease. This also explains the shape of the quarantined hosts curve. We can notice that when the number of infected hosts starts decreasing, the number of quarantined hosts starts increasing and when the number of quarantined hosts is approximately equal to the maximum number of infected hosts, at that time the number of infected nodes is almost zero. There is also a gradual and slow increase in the number of hosts that get patched and make transition to the removed state. This explains the shape of the removed hosts curve.

5.2 Comparison with no quarantine strategy and no defense strategy

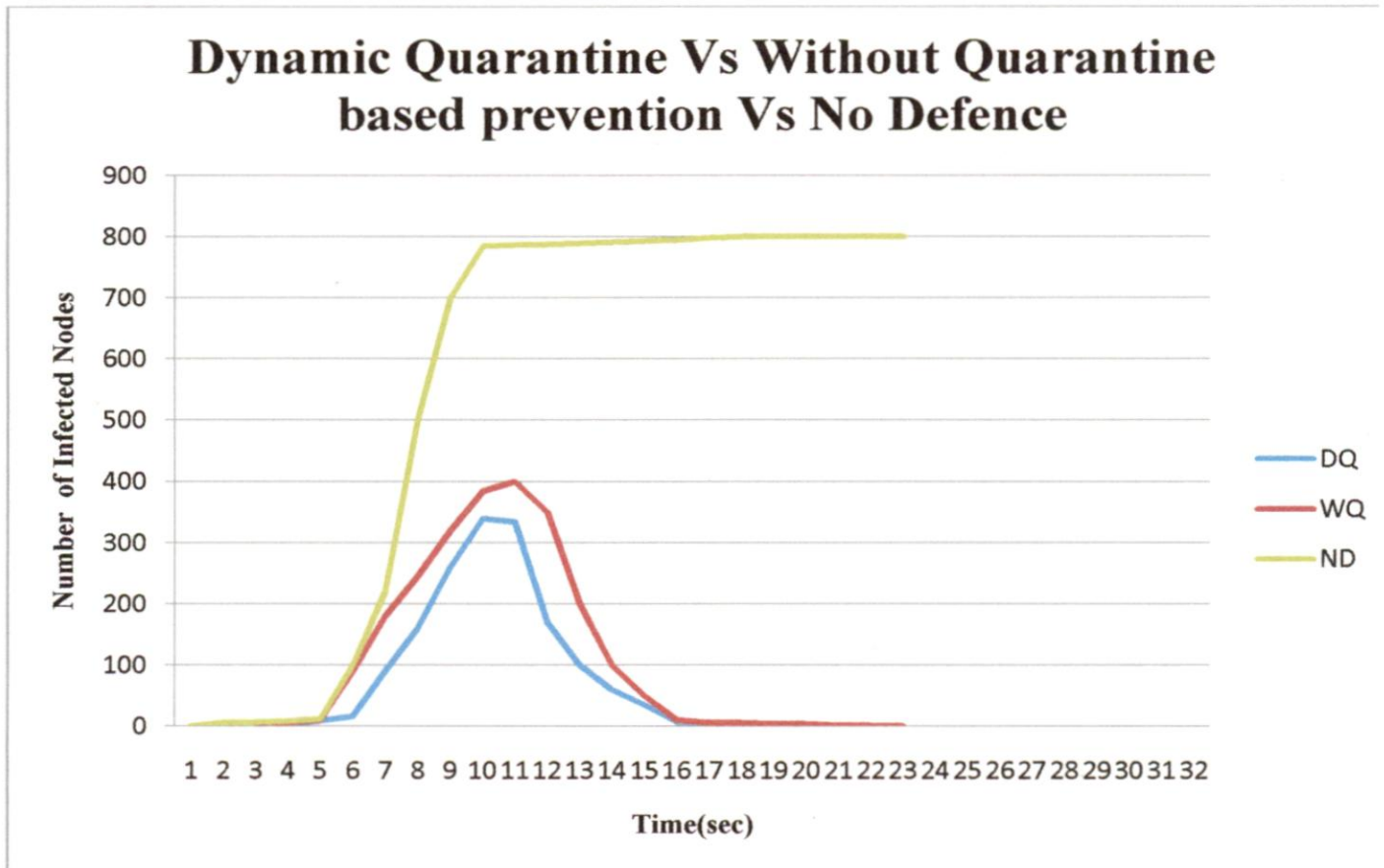


Figure 5.2 A comparison of different worm detection strategies

The dynamic quarantine curve (DQ) can be analyzed in the same way as in Figure 5.1. The Without quarantine curve(WQ) and No Defense curve (ND) have been plotted. It can be seen from Figure 5.2 that in our model the time taken by DQ curve to reach its peak is prolonged and there is a decrease in the maximum number of infected nodes in comparison to the case where no quarantine strategy or no defense strategy has been employed. This is because unlike without quarantine or no defense technique we quarantine i.e. “assume guilty before proven innocent” an abnormal host as soon as possible instead waiting for human counter measure on detecting an alarm for abnormal behavior.

5.3 Effect of degree of Node

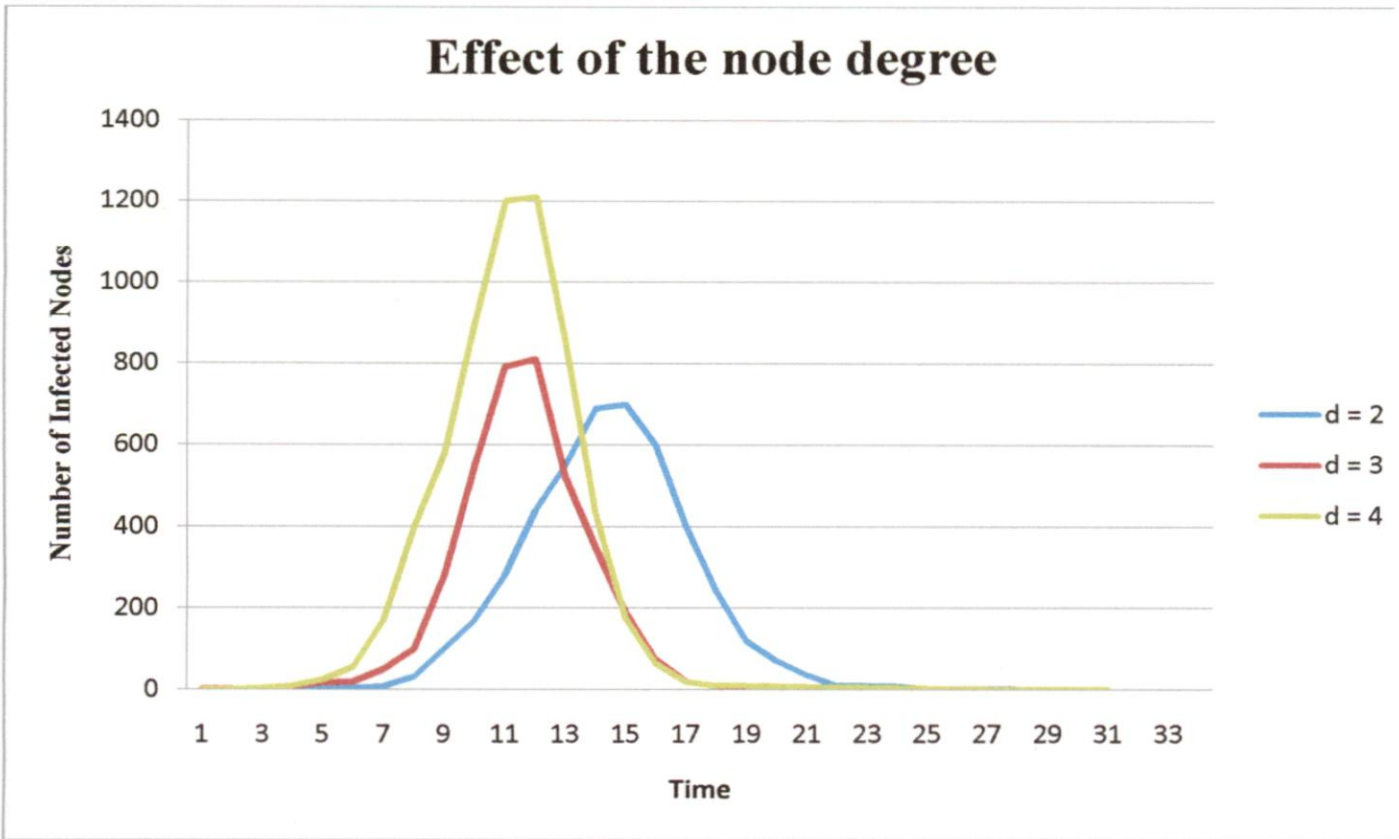


Figure 5.3 Effect of node degree on number of infectious hosts

The worm propagation speed is directly proportional to the peer degree. Higher is the peer degree, higher will be the number of newly infected nodes. The degree of node also determine the topology of peer to peer network, thus we can conclude that topology structure of P2P network is a critical factor that determines proactive worm propagation. As evident from Figure.5.3 higher degree makes the curve of number of infected hosts more steeper and the number of maximum infected nodes has also increased.

5.4 Effect of Quarantine Time

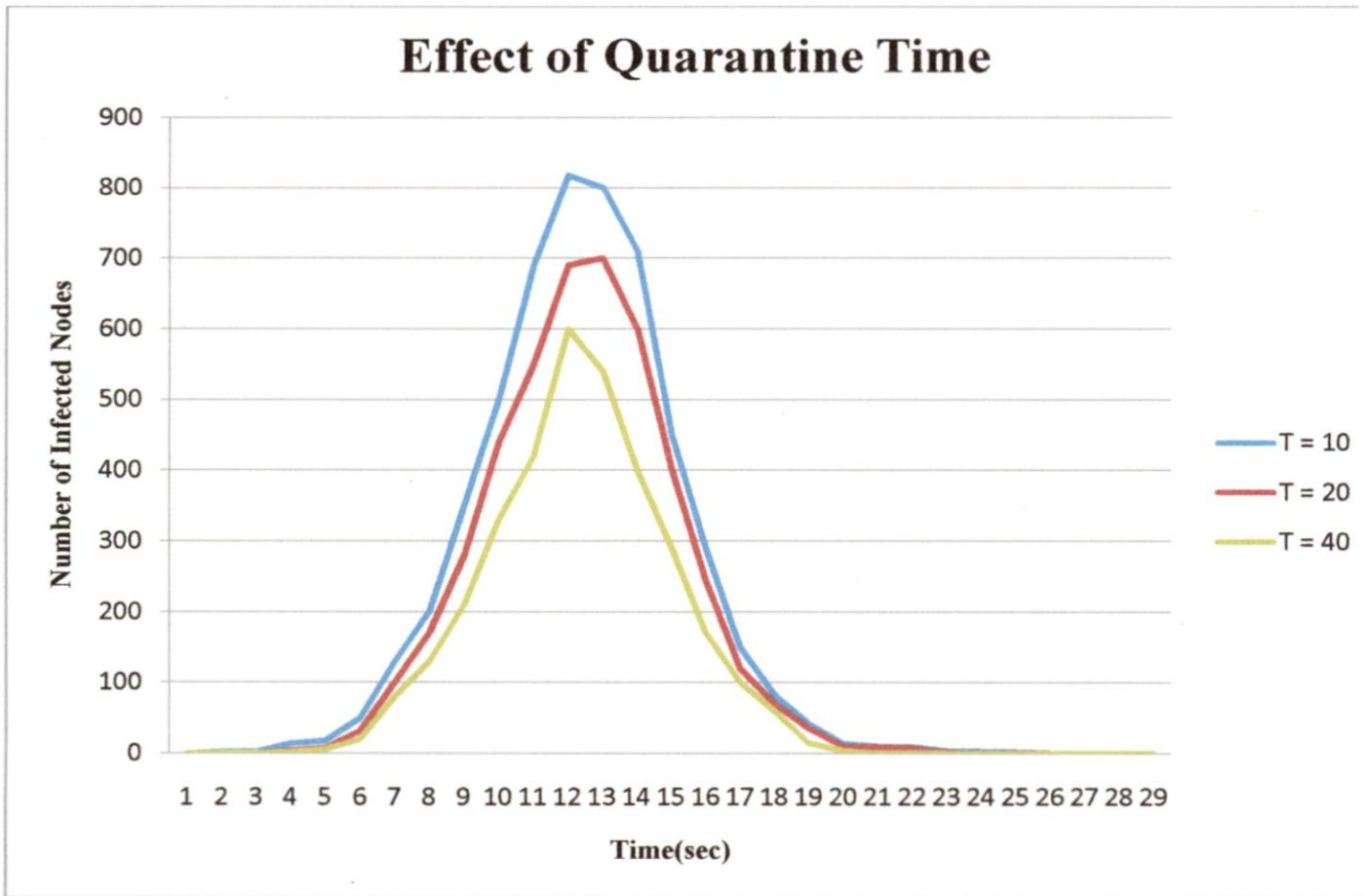


Figure 5.4 Effect of quarantine time

As evident from Figure 5.4 The larger the quarantine time, slower is the worm propagation rate and smaller is the number of infected hosts. This is because larger the time for which infected host remain quarantined, larger will be the time during which it can not infect its neighbors and larger will be time available for human countermeasures to do patching.

CONCLUSION AND FUTURE WORK

Proactive worms are a potential threat to P2P networks. A good control strategy for dealing with worm break out in a P2P network can be developed by taking advantage of their inherent property of distributed structure. A study of Internet worm propagation models can give a very good insight into worm behavior, identify the weakness in the worm spreading chain and provide accurate predictions for the purpose of damage assessment for a new worm threat in a P2P network.

Dynamic Quarantine method based on the principle “assume guilty before proven innocent” is a very efficient method and is based on quarantining a host whenever its behavior looks suspicious by blocking traffic only on anomaly port. Principles of dynamic quarantine model when combined with a host based worm detection system decreases the number of infected hosts and decline the propagation speed of worm. The false alarm rate in such a system is reduced by presence of some feedback technique based on principle “if adjacent hosts show the same abnormal behavior there is a greater possibility of worm propagation” which can be implemented by using a ‘super node’ for every peer group.

P2P’s topology structure and quarantine time are significant factors for deciding worm propagation. A larger quarantine time means lesser propagation speed and lesser number of infected nodes. A higher node degree means higher propagation speed and higher number of infected nodes.

Future work will involve more advanced dynamic quarantine system in which quarantine time and detection threshold can be changed dynamically during a worm’s propagation. Like epidemic disease control in the real world, if an internet worm is more infectious and poses more damage to our networks, the dynamic quarantine defense can be made more aggressive—the anomaly detection can become more sensitive to the worm’s activities, and the quarantine time can become longer to further constrain quarantined infectious hosts.

REFERENCES

- [1] "P2P Communication Explained- for geeks only" Available at <http://www.skype.com/intl/en-us/support/user-guides/p2pexplained/>
- [2] N, Weaver "A Brief History of the Worm" Available at <http://www.symantec.com/connect/articles/brief-history-worm>
- [3] Mike Barwise, "What is an Internet Worm" Available at <http://www.bbc.co.uk/webwise/guides/internet-worms>
- [4] J Thomsson "Internet Worm" http://www.livinginternet.com/i/is_vir_first.htm
- [5] Khiat, N., Charlinet, Y., Agoulmine, N., "The Emerging Threat of Peer-to-Peer Worms." *Proc. 1st IEEE Workshop on Monitoring, Attack Detection and Mitigation*, 2006. p.1-3.
- [6] Thommes, R., Coates, M., "Epidemiological Modeling of Peer-to-Peer Viruses and Pollution." *Proc. 25th IEEE Int. Conf. on Computer Communications*, 2006. p.181-192.
- [7] Chen, G., Gray, R.S., "Simulating Non-Scanning Worms on Peer-to-Peer Networks". *Proc. 1st Int. Conf. on Scalable Information Systems*, 2006. p.29-41. [doi:10.1145/1146847.1146876]
- [8] Moore, D., "The Spread of the Code-Red Worm (crv2)," 2001. Available at http://www.caida.org/analysis/security/code-red/coderev2_analysis.xml
- [9] H. Andersson, T. Britton. "Stochastic Epidemic Models and Their Statistical Analysis". Springer-Verlag, New York, 2000. 154-159
- [10] Mollison, D and HE Daniels 'The simple deterministic epidemic unmasked', *Math Biosciences* 117, 1993,147-153.

- [11] Zou C, Gong W, Towsley I. "Code Red Worm Propagation Modeling and Analysis" *Proc of ACM Conference on Computer and Communications Security*. Washington D C: ACM Press, 2002:138-147
- [12] Wei Yang, Gui-ran Chang, Yu Yao and Xiao-meng Shen "Stability Analysis of P2P Worm Propagation Model with Dynamic Quarantine Defense", *Journal Of Networks*, Vol. 6, No. 1, January 2011, doi:10.4304/jnw.6.1.153-162
- [13] D. Seeley. "A tour of the worm". In *Proceedings of the Winter Usenix Conference*, San Diego, CA, 1989.
- [14] Zou C, Gong W, Towsley D. "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense" *Proc of 2003 ACM Workshop on Rapid Malcode*. Washington D C: ACM Press, 2003:51-60
- [15] Elias Levy, "Worm Propagation and Generic Attacks," *IEEE Security and Privacy*, vol. 3, no. 2, pp. 63-65, March/April, 2005.
- [16] Wei, Fangfang Zhang Xu, Sencun Zhu: "Toward worm detection in online social networks." *ACSAC 2010: 11-20*
- [17] Mohammad M. Rasheed, Osman Ghazali, Norita Md Norwawi and Mohammed M. Kadhum, "Intelligent Failure Connection Algorithm for Detecting Internet Worms", *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.5, May 2009
- [18] Muhammad Adeel, Laurissa Tokarchuk, Laurie Cuthbert, Chao-sheng Feng, Zhi-guang Qin "A Distributed Framework for Passive Worm Detection and Throttling in P2P Networks" *CCNC'09 Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference* ISBN: 978-1-4244-2308-8
- [19] N. Weaver S. Staniford, V. Paxson. "How to Own Internet in your Spare Time." In *IEEE Security & Privacy*, 2004.

[20] Kai-Gui Wu, Yong Feng: "Proactive Worm Prevention Based on P2P Networks", *IJCSNS International Journal of Computer Science and Network Security*, Vol.6 No.3B,page 300-306, March 2006.

APPENDIX

$$I(t + \Delta t) - I(t) = S(t)[1 - (1 - 1/N - W(t))^{d(t)}] - (\mu + q_1 + \gamma + \varepsilon) I(t)$$

$$dI(t) / dt = S(t)[1 - (1 - 1/N - W(t))^{d(t)}] - (\mu + q_1 + \gamma + \varepsilon) I(t)$$

Proof:

Since the newly infected hosts will attack all their neighbors immediately. We assume that the infected hosts can generate k ($k > 0$), times attacks in time t in the overall P2P systems where $k = \Sigma \min(\beta, d_i)$. For any host in the P2P system, the probability of being attacked by one attack is $1/(N - W(t))$ and thus the probability of not being attacked is $1 - 1/(N - W(t))$. Then for any host, the probability of not being attacked by k times attacks is $(1 - 1/(N - W(t)))^k$. So the probability of being attacked by at least one of k attacks is $1 - (1 - 1/(N - W(t)))^k$.

There are total $S(t)$ hosts belonging to the susceptible population. So for k attacks, the newly added infected hosts from the susceptible stage to the infected stage can be derived by $E(t/k) = S(t)[1 - (1 - 1/(N - W(t)))^k]$.

When $k=1$ (one attack), there are $S(t)$ susceptible hosts and total $N - W(t)$ hosts at time t in the P2P system. One attack adds $S(t)/(N - W(t))$ newly infected hosts. Therefore for k attacks, we assume that the newly added infected hosts from the susceptible stage to the infected stage can be derived by $S(t)[1 - (1 - 1/(N - W(t)))^k]$.

Then, the $k+1$ th attack can be divided into two steps: the first k attacks and the last attack. For the last attack, there are two possibilities: adding a newly infected host and not adding a newly infected host. For convenience, a variable Y is introduced. If the last attack hit a susceptible host, let $Y = 1$. Otherwise, let $Y = 0$. Obviously event $Y = 1$ and event $Y = 0$ are mutually exclusive.

Let the probability P defined on events $Y = 1$ to be p that is $P(Y=1) = p$, then $P(Y=0) = 1 - p$.

The number of hosts transitioned from the susceptible stage to the infected stage at time t is given as follows:

$$\begin{aligned} E(t/k+1) &= (E(t/k) + 1)P(Y=1) + E(t/k)P(Y=0) \\ &= (E(t/k) + 1)p + E(t/k)(1-p) \end{aligned}$$

$$\begin{aligned}
&= E(t/k) + p \\
&= S(t)[1 - (1 - 1/(N - W(t)))^{k+1}]
\end{aligned}$$

The number of infected hosts at time t is $I(t)$ and the degree of infected host j is d_j . For infected host j can launch $k = \min(\beta, d_j)$ times attack at time t . The average degree of host is d and there for total infected hosts can launch $k = \sum \min(\beta, d_j) = dI(t)$ attacks.

Also, the number of hosts that transition from the infected stage to the recovery stage, wait stage and

quarantined stage are $\gamma I(t)$, $(\mu + \epsilon)I(t)$ and $q I(t)$, respectively. Hence we can derive equation :

$$dI(t) / dt = S(t)[1 - (1/(N - W(t)))^{dI(t)}] - (\mu + q_1 + \gamma + \epsilon)I(t)$$