ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Volume 1, Issue 7, September 2013

Automatic Detection and Control of Malware Spread in Decentralized Peer To Peer Network

K.G.S.Venkatesan

Assistant Professor, Dept of CSE, Bharath University, Chennai, Tamilnadu, India

ABSTRACT: Formulating an analytical model to characterize the spread of malware in decentralized, Gnutella type peer-to-peer (P2P) networks and the dynamics associated with the spread of malware are used. Compartmental model is used to derive the system parameters or network conditions so that the P2P network may reach a malware free equilibrium. The model also evaluates the effect of control strategies like node quarantine on stifling the spread of malware.

I.INTRODUCTION

Peer to peer networks provide a paradigm shift from the traditional client server model of most networking applications by allowing all users to act as both clients and servers. The use of peer-to-peer (P2P) networks as a vehicle to spread malware offers some important advantages over worms that spread by scanning for vulnerable hosts. This is primarily due to the methodology employed by the peers to search for content. For instance, in decentralized P2P architectures such as Gnutella where search is done by flooding the network, a peer forwards the query to its immediate neighbors and the process is repeated until a specified threshold time-to-live, TTL, is reached. Here TTL is the threshold representing the number of overlay links that a search query travels. A relevant example here is the Mandragore worm that affected Gnutella users. Having of the networking infrastructure. Infected a host in the network, the worm cloaks itself for other Gnutella users.

Every time a Gnutella user searches for media files in the infected computer, the virus always appears as an answer to the request, leading the user to believe that it is the file the user searched for. The design of the search technique has the following implications: first, the worms can spread much faster, since they do not have to probe for susceptible hosts and second, the rate of failed connections is less. Thus, rapid proliferation of malware can pose a serious security threat to the functioning of P2P networks. Understanding the factors affecting the malware spread can help facilitate network designs that are resilient to attacks, ensuring protection of the networking infrastructure. This paper addresses this issue and develops an analytic framework for modeling the spread of malware in P2P networks while accounting for the architectural, topological, and user related factors. We also model the impact of malware control strategies like node quarantine.

II. RELATIONSHIP TO PRIOR WORK

The existing system is specialized to Bit Torrent like networks .Bit Torrent is a P2P application whose goal is to facilitate fast downloads of popular files. Here we provide a brief description of how Bit Torrent operates when a single file is downloaded by many users. Typically the number of simultaneous downloaders for popular files could be of the order of a few hundred while the total number of down loaders during the lifetime of a file could be of the order of several tens or sometimes even hundreds of thousands. The basic idea in Bit Torrent is to divide a single large file (typically a few 100 MBytes long) into pieces of size 256 KB each. The set of peers attempting to download the file do so by connecting to several other peers simultaneously and download different pieces of the file from different peers.

Bit Torrent network is a very complicated system. It ignores node dynamics such as online-offline transitions. The models do not account for the fact that once a peer is infected, any susceptible peer within a TTL hop radius becomes a likely candidate for a virus attack.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Volume 1, Issue 7, September 2013

III. CURRENT WORK

Formulating an analytic framework for modelling the spread of malware in P2P networks while accounting for the architectural, topological, and user related factors. We also model the impact of malware control strategies like node quarantine. We formulate our model as a compartmental model, with the peers divided into compartments, each signifying it's state at a time instant. This model leads to the development of an automatic worm containment strategy that prevents the spread of a worm beyond its early stage. Automatic worm containment schemes effectively contain the worms and stop its spreading.

IV.MODULES

User Interface Design

This module we have designed the user interface for all the hosts. We design the user interface to show propagation of worms in a graphical manner or GUI. By showing the output in GUI gives more attractive and understandable to everyone. Then we design the containment window to show the scanning, detection of worms. Thus we design the whole user interface in this module.

Worm Propagation Model

This module, we create a worm spreading model. This model is designed for the propagation of worms inside a network. Inside the network we spread the worms in a controlled environment. To create worm propagation model we need to form a network by using the server socket class and socket class available in Java. These two classes are used to create a connection to transfer data from a host to other host inside a network.

Scanning for worms:

Our strategy is based on limiting the number of scans to dark-address space. The limiting value is determined by our analysis. Our automatic worm containment schemes effectively contain both uniform scanning worms and local preference scanning worms, and it is validated through simulations and real trace data to be non-intrusive.

Detecting and categorizing worms:

The model is developed for uniform scanning worms and then extended to preference scanning worms. We detect these two worms and categorize it in this module.

Containment of worms:

This model leads to the development of an automatic worm containment strategy that prevents the spread of a worm beyond its early stage. Specifically, for uniform scanning worms, we are able to

1) Provide a precise condition that determines whether the worm spread will eventually stop and
2) Obtain the distribution of the total number of hosts that the worm infects.

V. SYSTEM STUDY

A. ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

B. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Volume 1, Issue 7, September 2013

VI.TESTING

A. WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

B. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

VII. RESULTS

In this paper, we developed an analytic model to understand the automatic detection and control of malware spread in P2P networks. The need for an analytic framework incorporating user characteristics (e.g., offline to online transitional behavior), In this section, we validate our model using simulations and also demonstrate its capability to illustrate the effect of various system parameters on malware dynamics. The simulations were conducted using a custom built simulator. Results are reported for a 10,000 node network with a power-law graph topology . The initial network state for all simulations consisted of 4,950 randomly selected nodes in the susceptible online state, 5,000 randomly selected nodes in the susceptible offline state, and 50 randomly selected nodes in the infected online state.

VIII.CONCLUSION

In this paper, we developed an analytic model to understand the dynamics of malware spread in P2P networks. The need for an analytic framework incorporating user characteristics (e.g., offline to online transitional behavior) and communication patterns (e.g., the average neighborhood size) was put forth by quantifying their influence on the basic reproduction ratio.It was proved analytically that a model that does not incorporate the above features runs the risk of grossly overestimating R0 and thereby falsely reporting the presence of an epidemic. Further, our simulations show that the bound on the spectral radius for the spread of malware needs to take into account, the underlying communication pattern, especially in a P2P kind of setting so as arrive at an accurate estimate.

REFERENCES

[1] Clip2, "The Gnutella Protocol Specification v0.4," http://www.clip2.com/ GnutellaProtocol04.pdf, Mar. 2001.
[2] E. Damiani, D. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to- Peer Networks," Proc. ACM Conf. Computer and Comm. Security (CCS),pp. 207-216, Nov. 2002.
[3] X. Yang and G. de Veciana, "Service Capacity in Peer-to-Peer Networks," Proc. IEEE INFOCOM '04, pp. 1-11, Mar. 2004.