

International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 4, April 2017, pg.180 – 194

MALWARE PROPAGATION IN LARGE SCALE NETWORKS

Kavitha.U, S.Shanmugapriya

Student, Senior Assistant Professor

M.Tech, Dept. of CSE, New Horizon College of Engineering

Abstract- Malware is pervasive in networks, and poses a critical threat to network security. However, we have very limited understanding of malware behavior in networks to date. In this paper, we investigate how malware propagates in networks from a global perspective. We formulate the problem, and establish a rigorous two layer epidemic model for malware propagation from network to network. Based on the proposed model, our analysis indicates that the distribution of a given malware follows exponential distribution, power law distribution with a short exponential tail, and power law distribution at its early, late and final stages, respectively.

Extensive experiments have been performed through two real-world global scale malware data sets, and the results confirm our theoretical findings.

Keywords— Malware, propagation, modeling, power law

I. INTRODUCTION

MALWARE are malicious software programs deployed by cyber attackers to compromise computer systems by exploiting their security vulnerabilities. Motivated by extraordinary financial or political rewards, malware owners are exhausting their energy to compromise as many networked computers as they can in order to achieve their malicious goals. A compromised computer is called a bot, and all bots compromised by a malware form a botnet. Botnets have become the attack engine of cyber attackers, and they pose critical challenges to

cyber defenders. In order to fight against cyber criminals, it is important for defenders to understand malware behavior, such as propagation or membership recruitment patterns, the size of botnets, and distribution of bots[1].

To date, we do not have a solid understanding about the size and distribution of malware or botnets. Researchers have employed various methods to measure the size of botnets, such as botnet infiltration DNS redirection external information[1]. These efforts indicate that the size of botnets varies from millions to a few thousand. There are no dominant principles to explain these variations. As a result, researchers desperately desire effective models and explanations for the chaos. Dagon et al. revealed that time zone has an obvious impact on the number of available bots[2]. Mieghem et al. indicated that network topology has an important impact on malware spreading through their rigorous mathematical analysis.

In this paper, we study the distribution of malware in terms of networks (e.g., autonomous systems (AS), ISP domains, abstract networks of smartphones who share the same vulnerabilities) at large scales[3]. In this kind of setting, we have a sufficient volume of data at a large enough scale to meet the requirements of the SI model. Different from the traditional epidemic models, we break our model into two layers[3]. First of all, for a given time since the breakout of a malware, we calculate how many networks have been compromised based on the SI model. Second, for a compromised network, we calculate how many hosts have been compromised since the time that the network was compromised. With this two layer model in place, we can determine the total number of compromised hosts and their distribution in terms of networks. Through our rigorous analysis, we find that the distribution of a given malware follows an exponential distribution at its early stage, and obeys a power law distribution with a short exponential tail at its late stage, and finally converges to a power law distribution[3].

The proposed two layer epidemic model and the findings are the first work in the field.

Our contributions are summarized as follows[3].

1. We propose a two layer malware propagation model to describe the development of a given malware at the Internet level. Compared with the existing single layer epidemic

models, the proposed model represents malware propagation better in large-scale networks.

2. We find the malware distribution in terms of networks varies from exponential to power law with a short exponential tail, and to power law distribution at its early, late, and final stage, respectively[3]. These findings are first theoretically proved based on the proposed model, and then confirmed by the experiments through the two large-scale real-world data sets.

II. RELATED WORK

The basic story of malware is as follows. A malware programmer writes a program, called bot or agent, and then installs the bots at compromised computers on the Internet using various network virus-like techniques. All of his bots form a botnet, which is controlled by its owners to commit illegal tasks, such as launching DDoS attacks, sending spam emails, performing phishing activities, and collecting sensitive information[3]. There is a command and control (C&C) server(s) to communicate with the bots and collect data from bots. In order to disguise himself from legal forces, the botmaster changes the url of his C&C frequently, e.g., weekly[4]. An excellent explanation about this can be found in [4]. With the significant growing of smartphones, we have witnessed an increasing number of mobile malware. Malware writers have developed many mobile malware in recent years[5].

In this paper, we use two large scale malware data sets for our experiments. Conficker is a well-known and one of the most recently widespread malware. Shin et al. collected a data set about 25 million Conficker victims from all over the world at different levels. At the same time, malware targeting on Android based mobile systems are developing quickly in recent years. Zhou and Jiang collected a large data set of Android based malware.

Some of the previous works related to analysis of malware:

1) Information-Theoretic View Of Network Aware Malware Attacks

Smartphones are pervasively used in society, and have been both the target and victim of malware writers. Motivated by the significant threat that presents to legitimate users, we survey the current smartphone malware status and their propagation models[6]. The content of this paper is presented in two parts. In the first part, we review the short history of mobile malware evolution since 2004, and then list the classes of mobile malware and their infection vectors[6]. At the end of the first part, we enumerate the possible damage caused by smartphone malware. In the second part, we focus on smartphone malware propagation modeling[6]. In order to understand the propagation behavior of smartphone malware, we recall generic epidemic models as a foundation for further exploration. We then extensively survey the smartphone malware propagation models[6].

Disadvantage

- It only discusses the behavior of malwares.

2) Modeling and Automated Containment Of Worms

Self-propagating codes, called worms, such as Code Red, Nimda, and Slammer, have drawn significant attention due to their enormously adverse impact on the Internet. Thus, there is great interest in the research community in modeling the spread of worms and in providing adequate defense mechanisms against them. In this paper, we present a (stochastic) branching process model for characterizing the propagation of Internet worms[7]. The model is developed for uniform scanning worms and then extended to preference scanning worms. This model leads to the development of an containment strategy that prevents the spread of a worm beyond its early stage[7]. 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. We then extend our results to contain preference scanning worms[7]. 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 nonintrusive[7].

Disadvantage:

- It is not possible to prevent undesired messages. No matter user who propose them.

3) An Epidemic Theoretic Framework For Vulnerability Analysis Of Broadcast Protocols In Wireless Sensor Networks

While multi-hop broadcast protocols, such as Trickle, Deluge and MNP, have gained tremendous popularity as a means for fast and convenient propagation of data/code in large scale wireless sensor networks, they can, unfortunately, serve as potential platforms for virus spreading if the security is breached. To understand the vulnerability of such protocols and design defense mechanisms against piggy-backed virus attacks, it is critical to investigate the propagation process of these protocols in terms of their speed and reachability. In this paper, we propose a general framework based on the principles of epidemic theory, for vulnerability analysis of current broadcast protocols in wireless sensor networks[8]. In particular, we develop a common mathematical model for the propagation that incorporates important parameters derived from the communication patterns of the protocol under test. Based on this model, we analyze the propagation rate and the extent of spread of a malware over typical broadcast protocols proposed in the literature[8]. The overall result is an approximate but convenient tool to characterize a broadcast protocol in terms of its vulnerability to malware propagation.

Disadvantage:

- It uses the access control techniques to block Malware.

4) A large-scale empirical study of conficker

Conficker is the most recent widespread, well-known worm/bot. According to several reports, it has infected about 7 million to 15 million hosts and the victims are still increasing even now. In this paper, we analyze Conficker infections at a large scale, about 25 million victims, and study

various interesting aspects about this state-of-the-art malware. By analyzing Conficker, we intend to understand current and new trends in malware propagation, which could be very helpful in predicting future malware trends and providing insights for future malware defense. We observe that Conficker has some very different victim distribution patterns compared to many previous generation worms/botnets, suggesting that new malware spreading models and defense strategies are likely needed. We measure the potential power of Conficker to estimate its effects on the networks/hosts when it performs malicious operations[9]. Furthermore, we intend to determine how well a reputation-based blacklisting approach can perform when faced with new malware threats such as Conficker. We cross-check several DNS blacklists and IP/AS reputation data from Dshield and FIRE and our evaluation shows that unlike a previous study which shows that a blacklist-based approach can detect most bots, these reputation-based approaches did relatively poorly for Conficker[9]. This raises a question of how we can improve and complement existing reputation-based techniques to prepare for future malware defense? Based on this, we look into some insights for defenders. We show that neighborhood watch is a surprisingly effective approach in the case of Conficker.

Disadvantage:

- Providing this service is not only a matter of using previously defined web content mining techniques for a different application, rather it requires to design ad-hoc classification strategies.

Highlights of Related Work:

- The epidemic theory plays a leading role in malware propagation modeling. The current models for malware spread fall in two categories: the epidemiology model and the control theoretic model.
- The control system theory based models try to detect and contain the spread of malware. The epidemiology models are more focused on the number of compromised hosts and their distributions, and they have been explored extensively in the computer science community.

- Zou et al. used a susceptible-infected (SI) model to predict the growth of Internet worms at the early stage.
- Gao and Liu recently employed a susceptible-infected-recovered (SIR) model to describe mobile virus propagation.

III. SYSTEM ARCHITECTURE

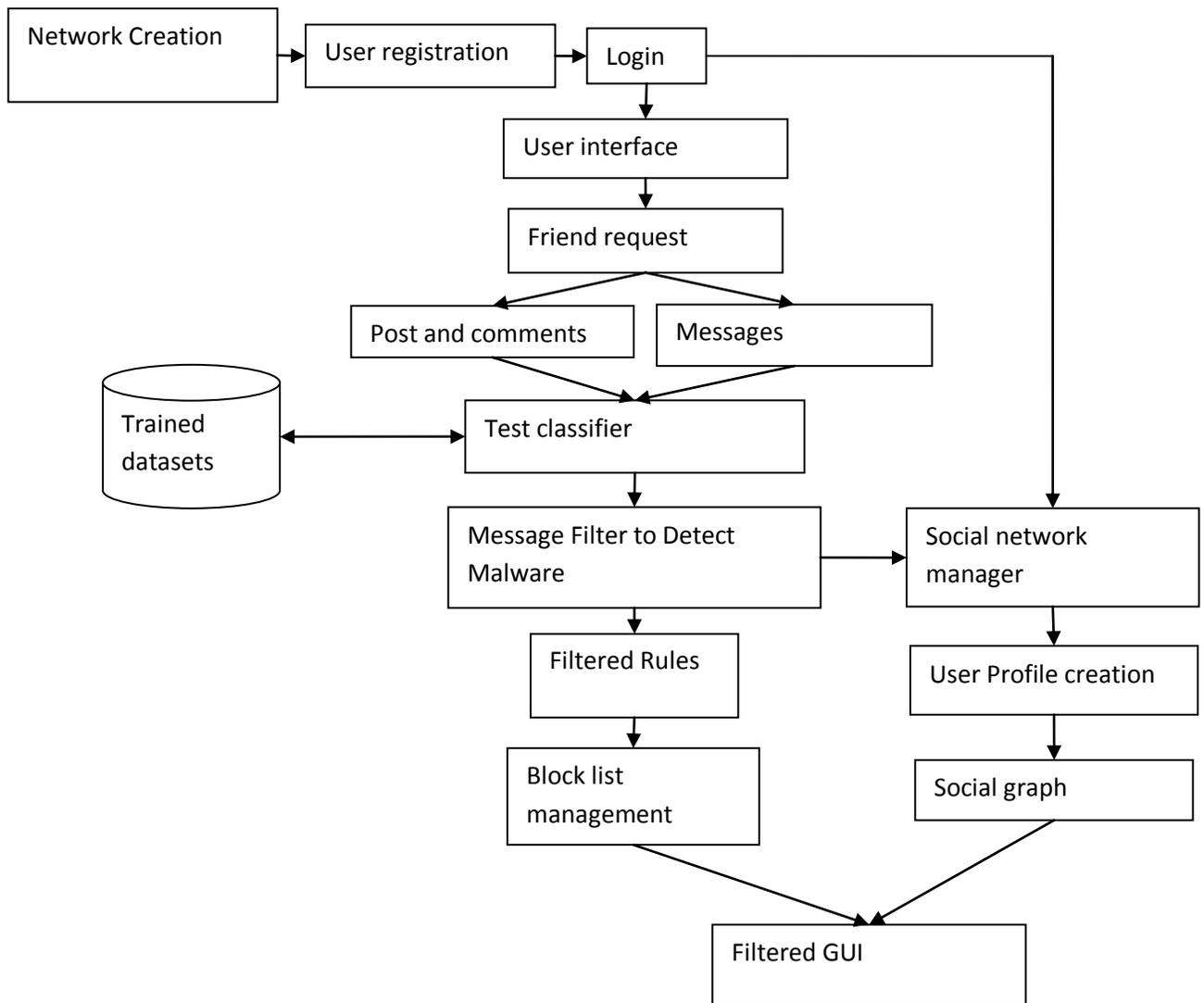
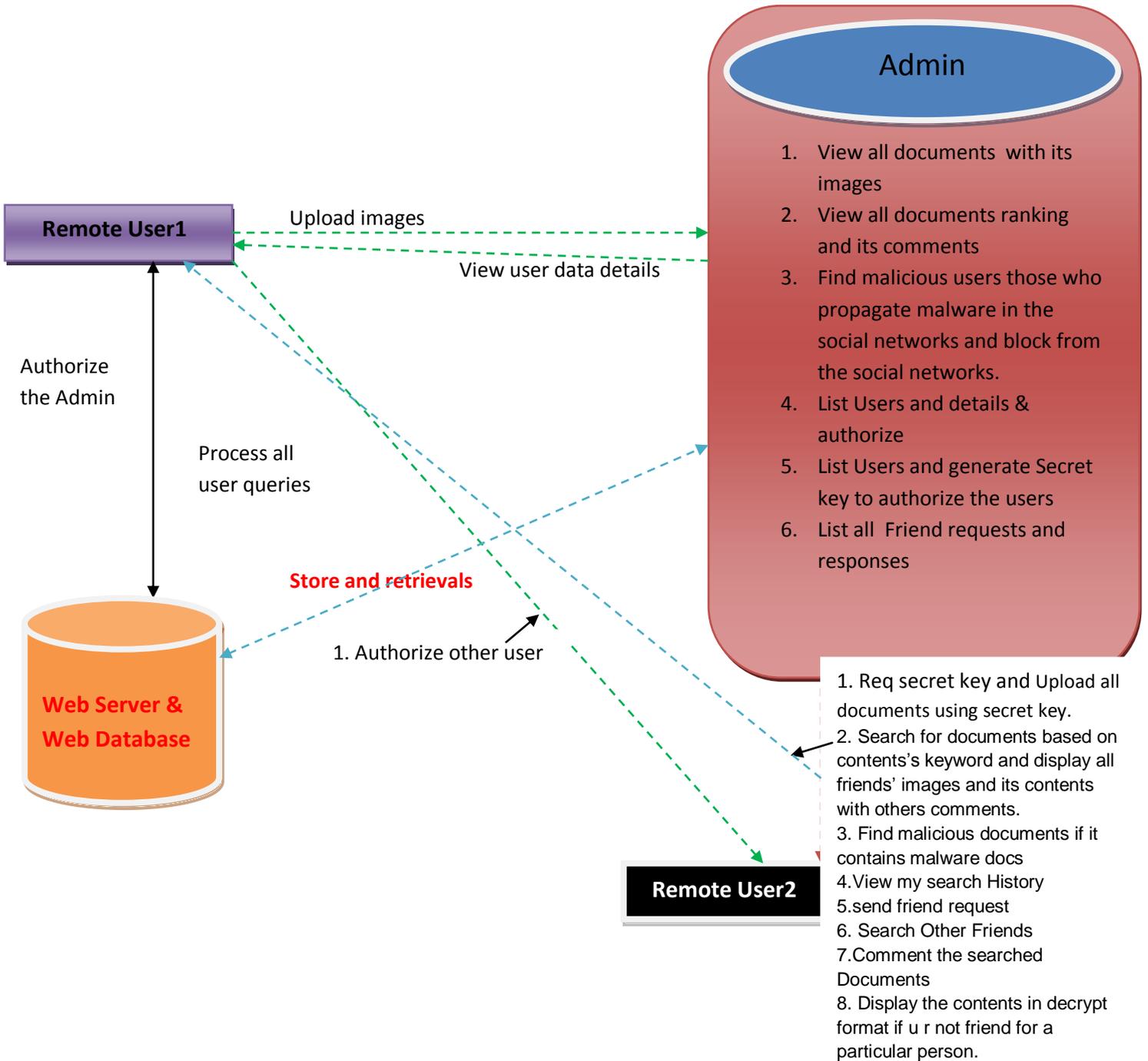


Fig 3.1 System architecture of Malware Propagation in Large scale Network

HIGH LEVEL ARCHITECTURE DIAGRAM:



NOTE :: The documents has to be searched only for friend users. Others can't retrieve the documents

Fig 3.2 A High Level Diagram of Malware propagation in Large Scale Network.

IV.PRELIMINARY INVESTIGATION

The first and foremost strategy for development of a project starts from the thought of designing a mail enabled platform for a small firm in which it is easy and convenient of sending and receiving messages, there is a search engine ,address book and also including some entertaining games. When it is approved by the organization and our project guide the first activity, i.e. preliminary investigation begins. The activity has three parts:

a) Request Clarification

b) Feasibility Study

c) Request Approval

REQUEST CLARIFICATION

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant for users within the company whose systems can be interconnected by the Local Area Network(LAN). In today's busy schedule man need everything should be provided in a readymade manner. So taking into consideration of the vastly use of the net in day to day life, the corresponding development of the portal came into existence.

FEASIBILITY STUDY

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are :

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

OPERATIONAL FEASIBILITY

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This type of automation will lead to reduce the time and energy that was previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

ECONOMIC FEASIBILITY

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

TECHNICAL FEASIBILITY

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

REQUEST APPROVAL

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, its cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

V. SYSTEM DESIGN AND DEVELOPMENT

INPUT DESIGN:

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible[10]. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations[11].

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made[11]. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format[11]. The goal of the input design is to make the data entry logical and free from errors. The error in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with an option to select an appropriate input from various alternatives related to the field in certain cases[12].

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

OUTPUT DESIGN:

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to

each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer is used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

VI. IMPLEMENTATION

The Implementation is elaborated with description of each module that play a major role in the project. They are as mentioned below:

i. Data Provider

In this module, the Service Provider browses the required file and uploads to the Social network to share with their friends. The data provider also perform the following operations such as show in the below :

❖ Add Document

In this module, the data provider can add the document. If user wants to add the new document, then he will enter document name, enter a document title, and so on then submit and that data will stored in data base.

❖ View Documents

In this module, the data provider can view the document details i.e., document name, document title, document image and document content, related images.

ii. **Admin Server(Web Server)**

The Admin server is responsible for performing some operations like to analyzing documents and contents to check whether the document contains malware. If documents are malware related then those documents will be scanned and Finds malicious users those who propagate malware in the social networks and block from the social networks and they will be keeping in the block list of the social networks.

iii. **Malware Files**

Proximity malware is a malicious program that disrupts the social network normal function and has a chance of duplicating itself to other user's documents or files in the social network.

iv. **Malware Distribution**

We find the malware distribution in terms of networks varies from exponential to power law with a short exponential tail, and to power law distribution at its early, late, and final stage, respectively[3]. These findings are first theoretically proved based on the proposed model, and then confirmed by the experiments through the two large-scale real-world data sets.

v. **End User**

In this module, there are n numbers of users are present. User should register before doing some operations. And register user details are stored in user module. After registration successful he has to login by using authorized user name and password[13]. After login successful, he will do some

operations like view or search users, send friend request, view messages, send messages and share social network data among number of users.

VII. CONCLUSION

In this paper, we thoroughly explore the problem of malware distribution at large-scale networks. The solution to this problem is desperately desired by cyber defenders as the network security community does not yet have solid answers. Different from previous modeling methods, we propose a two layer epidemic model: the upper layer focuses on networks of a large scale networks, for example, domains of the Internet; the lower layer focuses on the hosts of a given network[3]. This two layer model improves the accuracy compared with the available single layer epidemic models in malware modeling. Moreover, the proposed two layer model offers us the distribution of malware in terms of the low layer networks.

In regards to future work, we will first further investigate the dynamics of the late stage. More details of the findings are expected to be further studied, such as the length of the exponential tail of a power law distribution at the late stage. Second, defenders may care more about their own network, e.g., the distribution of a given malware at their ISP domains, where the conditions for the two layer model may not hold. We need to seek appropriate models to address this problem. Finally, we are interested in studying the distribution of multiple malware on large-scale networks as we only focus on one malware in this paper. We believe it is not a simple linear relationship in the multiple malware case compared to the single malware one[3].

REFERENCES

- [1] <http://omnetsimulation.com/malware-propagation-in-large-scale-networksmalware-propagation-in-large-scale-networksomnet-simulation/>
- [2] Fool Me If You Can: Mimicking Attacks and Anti-Attacks in Cyberspace, Article · Jan 2015 · IEEE Transactions on Computers
- [3] S. Yu, G. Gu, A. Barnawi, S. Guo and I. Stojmenovic, "Malware Propagation in Large-Scale Networks," in IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 1, pp. 170-179, Jan. 1 2015.
- [4] IJISSET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 9, September 2015. www.ijiset.com ISSN 2348 – 7968 An Efficient Detection Technique: Malware Spreading in Peer-to-Peer Networks

- [5] The Future of Mobile Malware By: Laura O'Brien
- [6] S. Peng, S. Yu and A. Yang, "Smartphone Malware and Its Propagation Modeling: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 925-941, Second Quarter 2014.
- [7] S. H. Sellke, N.B. Shroff, and S. Bagchi, "Modeling and automated containment of worms," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 2, pp. 71–86, Apr.–Jun. 2008.
- [8]. P. De, Y. Liu, and S. K. Das, "An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 3, pp. 413–425, Mar. 2009
- [9] S. Shin, G. Gu, N. Reddy and C. P. Lee, "A Large-Scale Empirical Study of Conficker," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 676-690, April 2012.
- [10] Cloud Based Protection For Multimedia Content, [IJIT-V2I5P6]:Deepak N S V, Md.Shareef Basha, Karamala Suresh.
- [11] <http://www.essay.uk.com/essays/computer-science/essay-cloud-storage/>
- [12] Cooperative Caching for Efficient Data Access in Disruption Tolerant Networks, http://www.academia.edu/28880430/Fresh_new_project
- [13] <http://www.ijcsmc.com/docs/papers/May2015/V4I5201552.pdf>