Computer Standards & Interfaces 44 (2016) 94–101

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Patching by automatically tending to hub nodes based on social trust

Xin Liu, Yao Wang, Dehai Zhao, Weishan Zhang, Leyi Shi

College of Computer and Communication Engineering, China University of Petroleum (East China), Qingdao, China

ARTICLE INFO

Article history:
Received 31 January 2015
Received in revised form 29 June 2015
Accepted 3 August 2015
Available online 8 August 2015

Keywords:
Social trust
Social computing
Automatic patching
Hub nodes
Vulnerability

ABSTRACT

Malicious code can propagate rapidly via software vulnerabilities. In order to prevent the explosion of malicious codes on the Internet, a distributed patching mechanism is proposed in which the patch can tend to hub nodes automatically based on social computing in social networks. A server in social network generates automatic patches and then selects those nodes with maximum degree to push automatic patches to. Those hub nodes then send the patch to their buddies according to their degree in social network. Automatic patches propagate rapidly through hub nodes and patch nodes in social network, which will improve the security of the whole social network. Those receivers accept the patch according to trust value to the sender, which can avoid some malicious codes exploit our scheme to propagate themselves. Experiments show this mechanism is more efficient than other patching mechanisms.

1. Introduction

According to the "2013 National Information Security, Computer and Mobile Terminal Virus Epidemic Survey Analysis Report," published by National Computer Virus Emergency Response Center, 76.4% network security incidents were caused primarily by malicious codes, and vulnerabilities in system and network without patching are still serious [1].

Malicious codes generally propagate by three kinds of methods: software vulnerability, users, or a combination of the two methods. Some malicious codes can start automatically without user involvement, such as worm, embedded script, etc. In order to propagate more effectively, most worms do not destroy their host during the propagation. The host may not realize it has been infected in general. For those host systems and applications with vulnerabilities, patching the system or the application is very effective. That is to say, those hosts which are vulnerable or have already been infected can download and install corresponding patches against the worms.

Here is an example of vulnerability.

Vulnerability MS06-014 is a logical vulnerability [2]. There is a vulnerability in RDS.Dataspace ActiveX, which binds with Microsoft Data Access Components (MDAC), leading to remote code execution vulnerability because it may not guarantee security interaction in certain conditions. An attacker who successfully exploits this vulnerability can take complete control of the host system.

Exploiting vulnerability MS06-014 will not occupy a lot of system memory and will not lead to browser crash and implement anti-antivirus easily, which makes it is one of the most influential vulnerability. There are lots of web trojan generators exploiting this vulnerability with a high success rate.

Exploit.MS06014.c is a script virus, which propagate itself by exploiting the vulnerability MS06-014. It generally propagates by web malicious codes. If the patch corresponding to MS06-014 has not been installed in a host, the host will be infected by this code when the user browses the webpage containing this malicious codes. Then the user host is under remote control by an attacker in all probably.

According to statistics from China National Vulnerability Database of Information Security (CNNVD), the number of vulnerabilities is always on the uptrend. Installing the security patches in OS and application program is an effective implement to patch network hosts. However, many users are unwilling install patches, or they forget or ignore to install the patches. Thus, the automatic patching mechanism is very necessary to protect user hosts. In this paper, we proposed a patching scheme in which automatic patch propagates in social network and patch the vulnerable hosts on the Internet to improve the security of hosts and the Internet.

The remainder of the paper is organized as follows. Section 2 presents related work. In Section 3, we explicate the motivation for automatic patching. The patching mechanism automatically tending to hub nodes based on social computing is explicated in Section 4. We presents security analyses and experimental evaluation in Sections 5 and 6. We conclude and present some future work in Section 7.

2. Related work

2.1. Automatic patching mechanism

Vojnovic and Ganesh researched the validity of automatic patching mechanism [3], taking the dissemination speed of the patches utilized in worm containment into quantization. An automatic patching system should be able to detect worms, generate patches, disseminate patches, verify and install patches. The author verified that using filter together with patch can improve effectiveness of worm containment radically.

http://dx.doi.org/10.1016/j.csi.2015.08.001
0920-5489/© 2015 Elsevier B.V. All rights reserved.

Xie and Zhu utilized existing P2P overlay network structure to disseminate security patches to susceptible hosts automatically [4]. It took two measures. One was based on segmentation, utilizing immune hosts to prevent the propagation of worms in overlay network in advance. The other one was based on connected dominating set (CDS), utilizing a group of dominating nodes in the overlay network to improve the dissemination speed of patches.

Shakkottai and Srikant researched two different dissemination strategy for defense against worms propagation [5]. First, they defended against worms incomparably with a fixed number of patching server. However, it generally took no time for worms to infect a large number of hosts, so that patching server with a fixed number could not deal with worm propagation. Second, they utilized PULL mechanism to disseminate patches to P2P nodes. Each P2P node randomly connected to another node and then inquired whether there is the patch or not. If there was, then the node downloaded, verify, and installed it. It could restrain the worm propagation effectively by using the exponential patching dissemination speed in P2P network.

Friedman proposed a method in which users scan friends' machines to make them defend users' local host [6]. Although the third-party scanning was a kind of intrusion to the local host, the scanning results were valuable and trustworthy. If there was a copy of the patch in a security host, then this host could scan its neighbors so that the susceptible neighbors made fences to propagate the patch to all hosts in which vulnerabilities exist. That is, a node which had installed patches could patch its neighbors. The system had two modes such as critical mode and casual mode. The critical mode was based on the new patches, embedding release date in records of the patches. The casual mode was to patch the minority hosts which had not been patched.

Zhu proposed a strategy that can contain phone worms in early stage of propagation [7]. It could establish a social relationship graph, a presentation of the most probable propagation path of phone worms, by the analysis of network flow. It used two algorithms to segment the social relationship graph and then chose the phone user who can infect the most phone users as the best phone set that should be patched first so that it could slow down the phone worm propagation speed and narrow the propagation range.

Stelios Sidiroglou and Angelos D. Keromytis proposed an architecture for automatically repairing software flaws that are exploited by zero-day worms [8]. This approach relies on source code transformations to quickly apply automatically created patches to vulnerable segments of the targeted application.

2.2. Social trust

Lazer et al. described that a computational social science is emerging that leverages the capacity to collect and analyze data with an unprecedented breadth and depth and scale [9]. The use of social trust relationships is both practical and necessary as the Web evolves [10].

Social computing is used in multiple fields, such as recommender systems [11][12], spam filtering [13][14], limiting free-riding in P2P networks [15], P2P routing [16], defending against Sybil attacks [17], defending against malicious Web pages [18].

When users share and exchange information with other people in the OSN, they have to manage the risk involved with the transactions without previous experience and knowledge about other users reputation. One way to solve this problem is to establish the system that can help users assign trust values to unknown users. Trust is the foundation of all interactions among society members [20].

Golbeck verified that the most correct information is from the most credible node in her doctoral dissertation by experiments [19]. She proposed TidalTrust, which is an algorithm for inferring trust, considering the trust values to be numbers in a continuous range [0,1]. The author showed that trust values inferred through shortest path may be more accurate. This is a classical algorithm with high citation rate, which is compared with by many algorithms later to prove efficiency.

We proposed a method of forming secure P2P network using benign worms against malicious worms [21]. The benign worm with a hitlist is generated to patch and clean the corresponding malicious worm for peers in the P2P network, which is based on the largest distance list. The benign worm propagates along the hitlist. The spread of the benign worm is also a distributed patching process. The patch can be disseminated more quickly, and the network congestion is much less than centralized patching scheme in P2P networks.

SocialTrust [22] provided community users with dynamic trust values by distinguishing relationship quality from trust, incorporating a personalized feedback mechanism for adapting as the community evolves and tracking user behavior.

Xin et al.[23] proposed a dynamic trust conference algorithm for social network. When there is only one shortest path from the source node to the destination node, the trust value calculated by the algorithm of Golbeck [19] will be the same as the trust value of the node, which is the last but one in the chain to the destination node. The result is not rational because a user usually has less trust in a stranger than a friend. In such case, the trust value to the destination node is calculated as the product of the trust values of the nodes adjacent to each other in the trust train. Then the authors give the mechanism for calculating dynamic trust value according to users' behavior in the network.

Zhuo et al.[24] introduced time-based dynamic trust model (TDTM), a model for time-based dynamic trust. Every node in the distributed environment is endowed with a trust vector, which figures the trust value between this node to the others. The trust value is dynamic due to the time and the inter-operation between two nodes. This change is quantified based on the mind of pheromone in the ant colony algorithm.

Yan et al. [25] proposed an algorithm for trust cluster head election based on ant colony systems. The cluster head plays an important role in clustering wireless sensor networks (WSNs). In [26], the authors presented a trust model for WSNs, called BTRM-WSN, based on ant colony systems. This model uses the mind of the ant colony system to find the most trustworthy route to the node that provides the requested service. The node in the route is selected according to the pheromone on the edge. The pheromone of a trace is regarded as the amount of trust on the edge and is used for the process of cluster head election. The goal of ACO algorithm in the model is to find the most trustworthy path to a request node. ACO is not used for trust calculation, which is not mentioned in the paper.

Bedi et al. [27] proposed a trust-based recommender system in which using ant colony for trust computation. The process of the trust computation is not the same as that in this paper. The thinking of the ant colony is used to select best neighborhood for the active user in the recommender system. The best neighborhood of the active user is a specific number of most trustworthy nodes that is linked to it. Along with those neighbors' ratings about some items, the active user can make better decisions.

3. Motivation

User nodes with various operation systems and applications in social network scatter around the Internet. Each software has its own particular vulnerability, which results in the special security problems of nodes in social networks. In order to improve the defending ability against malicious codes, nodes with vulnerabilities need to be patched.

Nodes on the Internet usually belong to some overlay networks such as P2P network, Instant Messaging network (IM network), and social network. Some special vulnerability exists in particular client software. For P2P network, many P2P worms propagate automatically exploiting vulnerabilities that exist in P2P software itself. It results in most peers in the P2P network are vulnerable, which makes some methods cannot carry out. For example, it is difficult to utilize the P2P network topology to contain the worms because most peers are vulnerable [28]. How to make nodes on the Internet be healthy?