Computer Networks 108 (2016) 97–107

Contents lists available at ScienceDirect

Computer Networks

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

A model for malware propagation in scale-free networks based on rumor spreading process

Soodeh Hosseini[a], Mohammad Abdollahi Azgomi[b,*]

[a] Trustworthy Computing Laboratory, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[b] School of Computer Engineering, Iran University of Science and Technology, Hengam St., Resalat Sq., Tehran, Iran, 16846-13114.

ARTICLE INFO

Article history:
Received 13 February 2016
Revised 12 July 2016
Accepted 8 August 2016
Available online 9 August 2016

Keywords:
Scale-free networks (SFNs)
Malware propagation modeling
Rumor spreading process
Diversification
Vaccination
Basic reproductive ratio

ABSTRACT

In this paper, we propose a dynamic model of malware propagation in scale-free networks (SFNs) based on a rumor spreading model. The proposed model, which is called the susceptible–exposed–infectious–recovered–susceptible with a vaccination state (SEIRS-V) model, illustrates the dynamics of malware propagation with respect to time in SFNs. The model considers the impact of software diversity to halt the outbreak of malware in networks. Using the SEIRS-V model, we derive the basic reproductive ratio that governs whether or not a malware is extinct. Furthermore, we calculate the number of diverse software packages installed on computer nodes that can be introduced as a parameter to prevent malware spreading. We accomplish the systematic analysis of the model and represent the local and global stability of malware-free equilibrium. Using numerical simulations, we examine the theoretical analysis. The effects of diversification and vaccination on the model are investigated. Simulation results demonstrate that the model is more effective than other existing models of malware propagation, in terms of reducing the density of infected node.

1. Introduction

With the advent and development of Internet technology during the past decades, the threats of malware and rumor become more serious. These threats can damage network security and social network safety. Malware, such as worms and viruses, are referred to as a crucial threat to confidentiality, integrity, and availability (CIA) of computer applications on the Internet. Moreover, rumor spreading can cause important consequences such as public panic and economic losses [1]. From the security viewpoint, the rumor spreading phenomenon is similar to malware propagation, in which all the informed nodes diffuse rumor by informing their neighbor nodes [2]. The existing models of rumor spreading are mostly obtained from the models of epidemic models [3]. The more realistic models of spreading processes are observed in real-world complex networks (e.g., Internet, world wide web (WWW), citation networks, online social networks and so on). Researches have shown that these networks have power-law degree distribution and heterogeneous network topology [4]. These networks are often referred to as scale-free networks (SFNs).

Most of the network nodes are running "the same software applications", which is called monoculture [5]. The monoculture networks share similar security vulnerabilities that facilitate the outbreak of malware [6]. Thus, the network malware exploit common vulnerabilities of software application to attain fast malware spreading. Due to security concerns of common vulnerabilities in identical applications, software diversity received much attention as cyber defense mechanisms in the real-world networks. Diversification generates different variants of applications by applying automatic program transformations, which preserve the functional behavior [5]. Using software diversity, we can reduce the virulence of malware and the efficiency single attacks in SFNs. Furthermore, we apply vaccination as an active defense approach to prevent the outbreak of infectious malware. The node vaccination leads to immunized nodes against malware infection. The vaccinated node cannot become infected again. We emphasize that node vaccination and software diversity together can terminate malware propagation in SFNs. This paper describes a malware propagation model according to the rumor spreading model, which is introduced in [7]. Unlike other models, our model considers the impact of software diversity with the assignment of distinct software packages to network nodes.

Our contributions in this paper are as follows. First, based on the rumor spreading model, we present a malware propagation model that considers software diversity. We investigate the

* Corresponding author. Fax: +98 21 73021480.
E-mail addresses: so_hosseini@iust.ac.ir (S. Hosseini), azgomi@iust.ac.ir (M.A. Azgomi).

http://dx.doi.org/10.1016/j.comnet.2016.08.010
1389-1286/

effectiveness of the model through extensive simulations and study the effects of different parameters such as diversification and vaccination in reducing the outbreak of malware. Second, we analyze dynamical behaviors of the model, and obtain the malware-free equilibrium point. Also, we derive the important parameters such as the basic reproductive ratio and the critical number of diverse software packages, to control malware propagation in the network.

The rest of the paper is organized as follows. In Section 2, we briefly review related work. In Section 3, we model malware propagation according to the rumor spreading model in SFNs with considering software diversity. In Section 4, we analyze the dynamics of the model and discuss the stability of the malware-free equilibrium. In Section 5, we represent a set of numerical simulations supporting our theoretical analysis, and study the impacts of software diversity and vaccination on the infected nodes. Finally, in Section 6, we conclude the paper with some future directions.

2. Related work

During the past few years, there has been serious attention in modeling and studying the spreading dynamics in complex networks. The studies have shown that many real-world complex networks, such as Internet, WWW, social networks and biological networks demonstrate heterogeneous topological properties such as scale-free distribution of degree and small world properties (high clustering and short average path length) [8]. These kinds of networks are often referred to as SFNs, which display power-law degree distribution $p(k) \sim k^{-\gamma}$ $(2 < \gamma \leq 3)$ [4]. In these networks, a few nodes (i.e., hubs) are linked to many other nodes, and a large number of poorly connected elements [4]. Many researchers have explained how the properties of networks influence the dynamical process occurring in real-world complex networks.

One of the most dynamical processes on these networks is the epidemic propagation. Studies of epidemic propagation have been done by many researchers [9]. There are two typical models to describe the epidemic propagation. The first epidemic model for dynamic process is the two-state susceptible-infectious-susceptible (SIS) model. In the susceptible state, the nodes are vulnerable to infection. In the infectious state, the nodes are already infected and can attack other vulnerable nodes. The SIS is the model where the susceptible nodes can become infected and the infected nodes can recover and come back to the susceptible state again [10]. The second model is the three-state susceptible-infectious-recovered (SIR) model, which is called Kermack–Mckendrick (KM) model [9,11]. The SIR epidemic model explains the epidemic that the infected nodes can become recovered or deadly. This model describes a new recovered state in comparison with the SIS model. The outbreak of the epidemic on complex networks has also been investigated with the other models, such as the SIRS model [12], SEIR model [13], SEIRS model [14] and etc. The SEIR epidemic model is a variation of the SIR epidemic model including the impacts of exposed (E) nodes, which have been infected by the malware but cannot yet propagate it [13]. Kuznetsov et al. [15] presented the numerical bifurcation analysis of SIR and SEIR epidemic models with periodic contact rate. They demonstrated that the parametric portrait of the SEIR epidemic model undergoes considerable structural changes when the latent period is modified [15]. Sanatinia et al. [16] applied an epidemiological approach, combined with experimental war-driving measurements to study the speed of infections propagation with distinct population and demographics. Due to the significant similarity between malware propagation and infectious diseases spreading, epidemic models have been applied in the malware propagation modeling.

Another dynamical process is the rumor spreading. Rumor spreading is also similar in nature as epidemic propagation [2]. The majority of the existing models of rumor propagation are variants of SIR epidemic models [17]. Based on the SIR model, Daley and Kendall [18] introduced a classical rumor spreading model, which was called the DK model. In DK model, homogeneous population is categorized into three groups [19]: *ignorant, spreaders* and *stifler*. When a spreader communicates with an ignorant, the ignorant becomes a spreader and when a spreader communicates a stifler, the spreader transits into a stifler [20]. Maki and Thomson [21] proposed the MK model of rumor spreading, which is a variant of The DK model. In the MK model when a spreader contacts another spreader only the initiating spreader becomes a stifler [22]. The DK and MK models have been applied widely for rumor spreading modeling, but serious weakness of these models is that they have not taken into account the topological properties of the complex networks [23]. With considering scale-free network topology structure, we introduced a new dynamic model of rumor spreading [7]. The introduced model was a variant of epidemic models.

Diffusion of software monocultures in real-world networks is considered to be a great threat to network security. Software diversity breaks up the impacts of the software monoculture [24], and for an attacker is very difficult to be able to design a unique attack to exploit common vulnerabilities in the software applications [5]. Software diversity is to produce various types of software with identical behavior (semantics) but with different structures. Thus, software diversity can change many of the existing approaches to software systems security. It will create the communication area and computer network safer. The results in [25] showed that diversity could decrease the virulence of malware such as worms, they assigned different software packages to network nodes in order to reduce the total number of nodes an attacker could attack using a single attack.

3. A dynamical model for malware propagation

In this section, we utilize and modify the proposed rumor spreading model in [7] to model malware propagation in SFNs with considering diversification. Here, we introduce the susceptible-exposed-infectious-recovered-susceptible with a vaccination (SEIRS-V) model to describe the dynamics of malware propagation. For modeling malware propagation, we will change the proposed rumor spreading model by (1) ignoring the hibernator[1] state in the model, because forgetting and remembering mechanisms apply in rumor spreading process; (2) adding transmission from recovered state to susceptible state. Also, we assign diverse software packages to nodes on the network to reduce the outbreak of malware and prevent the exploit of software vulnerability by a malware. In the following, we describe the SEIRS-V model and formulate it.

3.1. Model description

The principle of SEIRS-V rumor spreading model for malware propagation on scale-free networks is as follows. Considering a network shown as a graph with N nodes and M links (edges) representing the nodes and their connections. At each time step, each node adopts one of five possible states:

(1) *Susceptible (S)*: The nodes in this state are vulnerable to malware infection, can become infected when connecting to an infected node. (*Susceptible*, similar to ignorant state in the rumor spreading model.)
(2) *Exposed (E)*: The nodes in this state are exposed to the malware infection but do not show any noticeable symptoms, generally; the nodes have non-activated malware codes.

[1] Represents the nodes that receive the rumor, but forget it and later can remember it again.

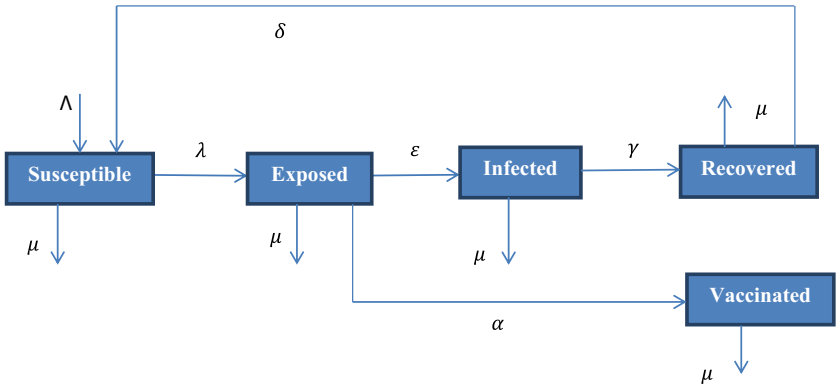

Fig. 1. The state transition diagram of the model.

(*Exposed*, similar to lurker[2] state in the rumor spreading model.)

(3) *Infectious (I)*: The nodes in this state propagate the malware infection, and transmit it to all their neighbors. (*Infectious*, similar to spreader state in the rumor spreading model.)

(4) *Recovered (R)*: The nodes in this state are recovered from the malware by anti-virus software (*Recovered*, similar to stifler1[3] state in the rumor spreading model.)

(5) *Vaccinated (V)*: The nodes in this state have been vaccinated and immunized to the malware infection. (*Vaccinated*, similar to stifler2[4] state in the rumor spreading model.)

In this model, we consider software diversity on the network to reduce the malware propagation speed. For controlling the malware spreading, *C*diverse software packages are assigned to the network nodes randomly. With the assignment of distinct software packages, the nodes acquire different binary codes in the network, i.e., nodes with the same type share the exploitable vulnerability while nodes with various types have no common vulnerability. Each diverse software package equal to a different color ($c = 1, 2, 3, \ldots, C$ for $1 \leq C \ll N$), N is denoted network size, in $C = 1$, we have no diversity, which is called monoculture.

Fig. 1 shows state transition diagram of the model.

As shown in Fig. 1, the states of susceptible, exposed, infected, recovered and vaccinated in the network follow the rules:

(1) When a susceptible node connects to an infected node of the same type, the susceptible node becomes an exposed node with probability λ, namely malware propagation rate. Generally, the malware infection can spread from every infected node to one of its susceptible neighbors of the same type.

(2) After elapsing exposure time, the exposed node can become an infected node with probability ε; or can transmit to the vaccinated state at the probability α before malware code activation with taking countermeasures, e.g., patching, intrusion-detection system (IDS), or anti-virus software [26].

(3) When an infected node is connected to another infected node the same type cannot spread infection because both of them are infectious, also the infected node can become recovered at a probability γ. In the rumor spreading process, when two spreaders (infected nodes) of the same type connect together, both receive two fragments of information, which are not inconsistent, so they stop propagation, i.e. in the malware propagation process, when the neighbors of an infected node are infected or immunized, thus it becomes isolated and cannot spread malware infection.

(4) The nodes are partially recovered and can become susceptible again with probability δ by malware within or outside the environment because of updated anti-virus without patching and loophole plugging [27]. As shown in Fig. 1, the recovered nodes can transfer into the susceptible state and become vulnerable to the malware infection. While the vaccinated nodes are immunized to the malware infection and do not get the malware infection.

Here, we consider leaving and joining nodes that are proportional to the density of nodes of degree k with the joining rate Λ and leaving rate μ, also, joins are balanced by leaves and thus the total number of nodes stays time invariant. Since join is equalized by leave, and the joining or leaving of a node or an edge only takes a small proportion in the network topology, hence, this is reasonable simplification in our model. Table 1 shows all the notations used in the malware propagation model.

Suppose that the population density of nodes at each time step is equal to $N_{k,c}(t) = S_{k,c}(t) + E_{k,c}(t) + I_{k,c}(t) + R_{k,c}(t) + V_{k,c}(t)$, each subset has $N_{k,c}(t)/C$ nodes at the time t, which is known a monoculture subgraph.

An important issue in modeling is the assumptions applied to simplify and solve the model. The model has the following assumptions:

1. Network topology is based on the Barabási–Albert (BA) with considering clustering.
2. The assignment of diverse software package (color) to each node is done randomly.
3. We consider the number of 1000 nodes in our experiments ($N = 1000$).
4. The total number of nodes remains time invariant; that is, joins are balanced by leaves, thus $\Lambda = \mu$ and $N_{k,c}(t) = S_{k,c}(t) + E_{k,c}(t) + I_{k,c}(t) + R_{k,c}(t) + V_{k,c}(t) \equiv 1$.
5. We consider the same leaving rate (μ) for each node.
6. In the beginning of the malware propagation process, all nodes are susceptible apart from a number of infected nodes (e.g. 100 nodes), acting as the "seed". The selection of initial infected nodes determines the malware propagation strategy. If the nodes are randomly selected to start the propagation process, the malware spreading strategy will be random.

[2] Represents the nodes that receive the rumor, but willing to spread the rumor because they require active effort to discern between true and false.

[3] Represents the nodes that accept the rumor but lose tendency to spread it.

[4] Represents the nodes that never accept the rumor and transmit this rumor again.

Table 1
The notations of the model.

Notation	Description
$S_{k,c}(t)$	The density of susceptible nodes of degree k and type c at the time t.
$E_{k,c}(t)$	The density of exposed nodes of degree k and type c the at the time t.
$I_{k,c}(t)$	The density of infected nodes of degree k and type c the at the time t.
$R_{k,c}(t)$	The density of recovered nodes of degree k and type c at the time t.
$V_{k,c}(t)$	The density of vaccinated nodes of degree k and type c at the time t.
C	The diversification rate; $C = 1,2,3,\ldots,c$.
λ	The malware propagation rate each infected node of type c. $0 < \lambda \leq 1$
ε	The transmission rate from exposed node to infected node of the same type. $0 < \varepsilon \leq 1$
γ	The transmission rate from infected node to recovered node of the same type. $0 < \gamma \leq 1$
α	The transmission rate from exposed node to vaccinated node of the same type. $0 < \alpha \leq 1$
δ	The transmission rate from recovered node to susceptible node of the same type. $0 < \delta \leq 1$
$\Lambda > 0$	The joining rate or logging into network.
$\mu > 0$	The leaving rate, which may occur in each of the states with the same proportion by crashing of nodes.

7. For increasing the accuracy of the simulations, every experiment is performed by 20 runs on average.

3.2. Model formulation

To investigate the dynamics of malware propagation according to the rumor spreading process, a mathematical model of the system is introduced. The model is analytically solved and simulated. In the following, we introduce the analytical model for modeling malware propagation.

We model malware propagation in SFNs with N nodes of ($C \geq 1$) types, we consider diversity with ($C > 1$). The infectivity of each node is proportional to its degree and applying C diverse software packages, thus the rate of infection propagation will be $\frac{\lambda}{C} k$. We have $I(t) = \sum_{k=m}^{\infty} I_{k,c}\ P(k)$; the parameter $I_{k,c}$ denotes the density of infected nodes of degree k and type c. m is the minimum degree in network topology and $P(k)$ denotes the probability that a node has the degree k. Since, on average, a node of degree k has k/c neighbors of the same type, $\frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\langle k\rangle/c}$ denotes the probability that a connection from a node exists to an infected node of the same type. The mean degree is $\langle k\rangle = \sum_k k\ P(k) = P(1) + 2P(2) + 3P(3) + \ldots + k_{max}P(k_{max})$, the mean degree of each subgraph with considering diversity is equal to $\langle k\rangle/c$. Diversity increases the number of monoculture sub-graphs and reduces malware propagation. At each time step, newly infected nodes of type c will be able to infect their susceptible neighbors with the same type (with the same software packages) because of the exploitability of common vulnerability by a malware, then some of them transfer into the exposed state and leave the susceptible state.

Thus, we will have:

$$\frac{\lambda}{c}\ kS_k(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\langle k\rangle/c}. \tag{1}$$

When an attack occurs, the density of the susceptible nodes transmitting to the exposed state will be decreased (Eq. (1)), also the recovered nodes can transmit into the susceptible state with probability δ. By considering the joining and the leaving rates in this state, we will have the following equation for the susceptible state:

$$\frac{dS_{k,c}(t)}{dt} = \Lambda - \frac{\lambda}{c}\ kS_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\langle k\rangle/c} + \delta R_{k,c}(t)\ -\ \mu\ S_{k,c}(t). \tag{2}$$

The density of susceptible nodes, which has been attacked them, will be added to the density of exposed nodes (Eq. (1)). The exposed nodes of type c will be infected after elapsing latency time and then will start to infect their susceptible neighbors of the same type with probability ε. Using anti-virus countermeasures, the exposed nodes can transfer into the vaccinated state with rate α. By considering the leaving rate μ in this state, the following equation for exposed state will be as follows:

$$\frac{dE_{k,c}(t)}{dt} = \frac{\lambda}{c}\ kS_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)\ I_{k,c}(t)}{\langle k\rangle/c} - \varepsilon E_{k,c}(t) - \alpha\ E_{k,c}(t) -\ \mu E_{k,c}(t). \tag{3}$$

The density of infected nodes is increased by the density of exposed nodes of the same type at the time t with probability ε. When an infected node connects to another infected node or a recovered node or a vaccinated node of the same type, the initiating infected node becomes a recovered node with probability γ, then we have the reduction of the density of infected nodes. By considering the leaving rate μ in this state, we will determine the following equation for infected state:

$$\frac{dI_{k,c}(t)}{dt} = \varepsilon E_{k,c}(t) - \gamma\ k\ I_{k,c}(t) \times \frac{\sum_{k=m}^{\infty}(k/c)P(k)\ \left[I_{k,c}(t) + R_{k,c}(t) + V_{k,c}(t)\right]}{\langle k\rangle/c} - \mu\ I_{k,c}(t). \tag{4}$$

As stated earlier, the infected nodes will be recovered at rate of γ with antivirus software, thus the density of recovered nodes will be increased. Furthermore, they will be decreased when the recovered nodes of type c transfer into the susceptible state of the same type with probability δ due to updates of virus-bases or re-installing operating system [26]. By considering the leaving rate μ in this state, the following equation for recovered state will be as follows:

$$\frac{dR_{k,c}(t)}{dt} = \gamma\ k\ I_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)\ \left[I_{k,c}(t) + R_{k,c}(t) + V_{k,c}(t)\right]}{\langle k\rangle/c} - \delta R_{k,c}(t) - \mu\ R_{k,c}(t). \tag{5}$$

The density of vaccinated nodes will be increased, when the exposed nodes transit into the vaccinated state before malware code activation. The transmission rate from exposed state to vaccinated state of the same type will be α. By considering the leaving rates μ in this state, we will have the following equation for the vaccinated state:

$$\frac{dV_{k,c}(t)}{dt} = \alpha\ E_{k,c}(t) - \mu\ V_{k,c}(t). \tag{6}$$

In summary, the differential equations of the SEIRS-V model are as follows:

$$\frac{dS_{k,c}(t)}{dt} = \Lambda - \frac{\lambda}{c}\ kS_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\langle k\rangle/c} + \delta R_{k,c}(t)\ -\ \mu\ S_{k,c}(t)$$

$$\frac{dE_{k,c}(t)}{dt} = \frac{\lambda}{c}\ kS_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)\ I_{k,c}(t)}{\langle k\rangle/c} - \varepsilon E_{k,c}(t) - \alpha\ E_{k,c}(t) -\ \mu E_{k,c}(t)$$

$$\frac{dI_{k,c}(t)}{dt} = \varepsilon E_{k,c}(t) - \gamma\ k\ I_{k,c}(t) \times \frac{\sum_{k=m}^{\infty}(k/c)P(k)\left[I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)\right]}{\langle k\rangle/c} - \mu\ I_{k,c}(t)$$

$$\frac{dR_{k,c}(t)}{dt} = \gamma\ k\ I_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)\left[I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)\right]}{\langle k\rangle/c} - \delta R_{k,c}(t) - \mu\ R_{k,c}(t)$$

$$\frac{dV_{k,c}(t)}{dt} = \alpha\ E_{k,c}(t) - \mu\ V_{k,c}(t). \quad (7)$$

The initial conditions for model of Eq. (7) are as follows: $S_{k,c} > 0,\ E_{k,c} \geq 0, I_{k,c} \geq 0, R_{k,c} \geq 0, V_{k,c} \geq 0$, for any $k = 1, 2, \ldots n,\quad c = 1, 2, \ldots C$ and $t \geq 0$.

Summing up the five equations of model of Eq. (7), we can get the following equation

$$\frac{d\ N(t)}{dt} = \Lambda - \mu\ N(t). \quad (8)$$

Hence: $S_{k,c} + E_{k,c}, +I_{k,c} + R_{k,c} + V_{k,c} \leq \frac{\Lambda}{\mu}$, where $\Lambda = \mu$.

The feasible region for model of Eq. (7) is $U = \{(S_{k,c}, E_{k,c}, I_{k,c}, R_{k,c}, V_{k,c}) \in \Gamma_{+}^{5} : S_{k,c} + E_{k,c}, +I_{k,c} + R_{k,c} + V_{k,c} \leq 1,\ k = 1, 2, \ldots n,\quad c = 1, 2, \ldots C\}$ is a positive invariant set for model of Eq. (7), and it is sufficient to study the dynamics of model of Eq. (7) in U.

4. Dynamical analysis of the model

Mathematical analysis can make suitable theoretical foundation for predicting malware spreading. In this section, we will obtain the equilibria of model of Eq. (7) and determine dynamical behaviors of the model. We calculate the basic reproductive ratio and the number of diverse software packages required to prevent malware propagation. Also, we discuss the local and global stability of the model malware-free equilibrium.

Now we find the equilibrium points of model of Eq. (7) and analyze its stability, the steady states of model of Eq. (7) are as follows:

$$\frac{dS_{k,c}(t)}{dt} = 0,\ \frac{dE_{k,c}(t)}{dt} = 0,\ \frac{dI_{k,c}(t)}{dt} = 0,\ \frac{dR_{k,c}(t)}{dt} = 0,\ \frac{dV_{k,c}(t)}{dt} = 0. \quad (9)$$

After simple calculating, we obtain equilibrium points as: $EQ_1 = (1, 0, 0, 0, 0)$ for malware-free state and $EQ_2 = (S^*, E^*, I^*, R^*, V^*)$ for endemic stage, where

$$S^* = \frac{c(\varepsilon+\alpha+\mu)}{\lambda k\theta_1^*} \times \frac{(\gamma k\theta_2^* + \mu)}{\varepsilon} I^*,$$

$$E^* = \frac{(\gamma k\theta_2^* + \mu)}{\varepsilon} I^*,$$

$$V^* = \frac{\alpha}{\mu}\frac{(\gamma k\theta_2^* + \mu)}{\varepsilon} I^*,$$

$$I^* = \frac{\delta+\mu}{\left[\frac{(\gamma k\theta_2^*\varepsilon+\beta\gamma k\theta_2^*+\mu)}{\varepsilon}\right] + (\delta+\mu)\left[1 + \frac{(\gamma k\theta_2^*+\mu)}{\varepsilon} + \frac{c(\varepsilon+\alpha+\mu)}{\lambda\ k\ \theta_1^*} \times \frac{(\gamma k\theta_2^*+\mu)}{\varepsilon} + \frac{\alpha}{\mu}\frac{(\gamma k\theta_2^*+\mu)}{\varepsilon}\right]},$$

$$R^* = 1 - S^* - E^* - I^* - V^*,$$

where $\theta_1^* = \frac{\sum_{k=m}^{\infty}(k/c)P(k)\ I^*}{\langle k\rangle/c}$, $\theta_2^* = \frac{\sum_{k=m}^{\infty}(k/c)P(k)\ [I^*+R^*+V^*]}{\langle k\rangle/c}$.

4.1. The basic reproductive ratio

In mathematical epidemiology, one of the most fundamental concepts is to determine the threshold which illustrates whether or not an infectious state can persist in the network. This threshold is called basic reproductive ratio (R_0, which is equal to the expected number of secondary infectious cases generated by a typical infected node during its entire period of infectiousness in a completely susceptible population [28]. Generally, the basic reproductive ratio investigates the global dynamics of the model. It is derived through the local stability of malware-free equilibrium and established as a threshold that governs the malware dynamics [29]. In particular, the malware infection fades out from the network if $R_0 < 1$, and otherwise, if $R_0 > 1$ the malware infection persists at the endemic level in the network [29]. The basic reproductive ratio is often obtained by the spectral radius of the next-generation operator [28].

In order to obtain the basic reproductive ratio, we rewrite the model of Eq. (7). Noticeably, the first four equations in Eq. (7) do not depend on the fifth equation, and hence, without loss of generality, this equation can be ignored. Therefore, the model of Eq. (7) becomes

$$\frac{dS_{k,c}(t)}{dt} = \Lambda - \frac{\lambda}{c}\ kS_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\langle k\rangle/c} + \delta R_{k,c}(t) - \mu\ S_{k,c}(t).$$

$$\frac{dE_{k,c}(t)}{dt} = \frac{\lambda}{c}\ kS_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)\ I_{k,c}(t)}{\langle k\rangle/c} - (\varepsilon+\alpha+\ \mu)E_{k,c}(t).$$

$$\frac{dI_{k,c}(t)}{dt} = \varepsilon E_{k,c}(t) - \gamma\ k\ I_{k,c}(t) \times \frac{\sum_{k=m}^{\infty}(k/c)P(k)\left[I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)\right]}{\langle k\rangle/c} - \mu\ I_{k,c}(t).$$

$$\frac{dR_{k,c}(t)}{dt} = \gamma k I_{k,c}(t)\frac{\sum_{k=m}^{\infty}(k/c)P(k)\left[I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)\right]}{\langle k\rangle/c} - (\delta+\mu)R_{k,c}(t). \quad (10)$$

We will investigate the dynamical behavior of the model of Eq. (10). As mentioned earlier, the model has a malware-free equilibrium, $EQ_1 = (1, 0, 0, 0)$.

Let $x = (E, I, R, S)^T$, then the model of Eq. (10) can be written as

$$x' = F(x) - Z(x), \quad (11)$$

where

$$F(x) = \begin{pmatrix} AS \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad Z(x) = \begin{pmatrix} (\varepsilon+\alpha+\mu)E \\ (\mu+D)I - \varepsilon E \\ (\delta+\mu)R - DI \\ -\delta R + (\mu+A)S \end{pmatrix}.$$

In $F(x)$; $A = \frac{\lambda}{c}\ k\frac{\sum_{k=m}^{\infty}(k/c)P(k)\ I_{k,c}(t)}{\langle k\rangle/c}$, and in $Z(x)$; $D = \gamma\ k\ \frac{\sum_{k=m}^{\infty}(k/c)P(k)\ [I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)]}{\langle k\rangle/c}$.

The Jacobian matrices of $F(x)$ and $Z(x)$ at the malware-free equilibrium EQ_1 are, respectively,

$$JF(EQ_1) = \begin{pmatrix} F_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{pmatrix}, \quad JZ(EQ_1) = \begin{pmatrix} Z_{2\times2} & 0_{2\times2} \\ Z^1_{2\times2} & Z^2_{2\times2} \end{pmatrix}$$

where

$$F_{2\times2} = \begin{pmatrix} 0 & G \\ 0 & 0 \end{pmatrix}, \quad Z_{2\times2} = \begin{pmatrix} (\varepsilon+\alpha+\mu) & 0 \\ -\varepsilon & \mu \end{pmatrix},$$

$$Z^1_{2\times2} = \begin{pmatrix} 0 & 0 \\ 0 & G \end{pmatrix}, \quad Z^2_{2\times2} = \begin{pmatrix} (\delta+\mu) & 0 \\ -\delta & \mu \end{pmatrix},$$

and

$$G = \frac{\lambda}{c\langle k\rangle} \begin{bmatrix} 1 \\ 2 \\ . \\ . \\ n \end{bmatrix} \begin{bmatrix} 1\ P(1), & 2\ P(2), & \ldots, & n\ P(n) \end{bmatrix}.$$

$F_{2\times2}\ Z^{-1}_{2\times2}$ is the next generation matrix for the model of Eq. (10). For finding R_0, we use next-generation matrix method. According to Theorem 2 in [30], the basic reproductive ratio of the model is proportional to $R_0 = \rho(F_{2\times2}\ Z^{-1}_{2\times2}) = \frac{\varepsilon\ G}{\mu(\varepsilon+\alpha+\mu)}$.

where

$$R_0 = \frac{\varepsilon}{\mu\ (\varepsilon+\alpha+\mu)} \frac{\lambda}{c\langle k\rangle} \begin{bmatrix} 1P(1) & 2P(2) & \ldots.. & nP(n) \\ 2P(2) & 2^2P(2) & \ldots.. & 2nP(n) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ nP(1) & 2nP(2) & \ldots\ldots & n^2P(n) \end{bmatrix},$$

$$\Rightarrow \quad R_0 = \frac{\varepsilon}{\mu\ (\varepsilon+\alpha+\mu)} \frac{\lambda\langle k^2\rangle}{c\langle k\rangle}. \tag{12}$$

Now, we calculate the critical number of diverse software packages to halt the malware propagation in the network. Hence, the critical number of distinct software packages needed for prevention of malware spreading is as follows:

$$R_0 = \frac{\varepsilon}{\mu(\varepsilon+\alpha+\mu))} \frac{\lambda\langle k^2\rangle}{c\langle k\rangle} < 1$$

$$\Rightarrow C_{critical} = \left[\frac{\varepsilon}{\mu(\varepsilon+\alpha+\mu)} \frac{\lambda\langle k^2\rangle}{\langle k\rangle} \right]. \tag{13}$$

4.2. Stability analysis of the malware-free equilibrium

We investigate the stability of the malware-free equilibrium of the model of Eq. (10) to study dynamical behaviors of the model. According to the mode of Eq. (10), Jacobian matrix at the malware-free equilibrium EQ_1 is

$$J(EQ_1) = \begin{bmatrix} -\mu & 0 & -G & \delta \\ 0 & -(\varepsilon+\beta+\alpha+\mu) & G & 0 \\ 0 & \varepsilon & -\mu & 0 \\ 0 & 0 & 0 & -(\delta+\mu) \end{bmatrix}. \tag{14}$$

where $G = \frac{\lambda}{c\langle k\rangle} \begin{bmatrix} 1 \\ 2 \\ . \\ . \\ n \end{bmatrix} [1\ P(1), \quad 2\ P(2), \quad \ldots, \quad n\ P(n)]$, the eigenfunction of $J(EQ_1)$ is

$$|\omega\ H - J(EQ_1)| = \begin{bmatrix} \omega+\mu & 0 & G & -\delta \\ 0 & \omega+(\varepsilon+\alpha+\mu) & -G & 0 \\ 0 & -\varepsilon & \omega+\mu & 0 \\ 0 & 0 & 0 & \omega+(\delta+\mu) \end{bmatrix}, \tag{15}$$

the matrix H is identity matrix and ω is eigenvalue.

So, the eigenfunction of $J(EQ_1)$ is equal to $f(\omega) = a_4\omega^4 + a_3\omega^3 + a_2\omega^2 + a_1\omega + a_0$, where

$$a_4 = 1;$$
$$a_3 = 4\mu + \delta + \alpha + \varepsilon;$$
$$a_2 = 6\mu^2 + 3\mu(\delta+\alpha+\varepsilon) + \delta(\alpha+\varepsilon) - G\varepsilon;$$
$$a_1 = 4\mu^3 + 3\mu^2(\delta+\alpha+\varepsilon) + 2\mu\delta(\alpha+\varepsilon) - 2G\varepsilon\mu - G\varepsilon\delta;$$
$$a_0 = \mu^4 + \mu^3(\delta+\alpha+\varepsilon) + \mu^2\delta(\alpha+\varepsilon) - G\varepsilon\mu^2 - G\varepsilon\mu\delta;$$

Now, we use the following lemma and theorem in the stability analysis of the model. (The lemma and theorem are based on the existing works on stability analysis, such as [31,32]).

Lemma 1. *The malware-free equilibrium EQ_1 is locally asymptotically stable if $R_0 < 1$, and it is unstable if $R_0 > 1$, where R_0 is calculated by Eq. (12).*

Proof. Based on the Routh–Hurwitz criterion, the Routh-Hurwitz array for malware-free equilibrium EQ_1 is as follows

$$\begin{bmatrix} a_4 & a_2 & a_0 \\ a_3 & a_1 & 0 \\ b_1 & a_0 & 0 \\ c_1 & 0 & 0 \\ a_0 & 0 & 0 \end{bmatrix}, . \tag{16}$$

where $b_1 = \frac{a_3\ a_2 - a_4a_1}{a_3}$, $\quad c_1 = \frac{b_1\ a_1 - a_3a_0}{b_1}$.

According to theorem in [33], *"The necessary condition for the system to be stable is; all the elements of the first column of dynamic-Routh's array must have positive values."* Hence, the necessary condition for the model to be stable is that $a_4 > 0$, $a_3 > 0$, $b_1 > 0$, $c_1 > 0$, and $a_0 > 0$.

For a_0, if we can show that $\mu^4 + \mu^3(\delta+\alpha+\varepsilon) + \mu^2\delta(\alpha+\varepsilon) > G\varepsilon\mu^2 + G\varepsilon\mu\delta$, then $a_0 > 0$. When $R_0 < 1$, $\mu^4 + \mu^3(\delta+\alpha+\varepsilon) + \mu^2\delta(\alpha+\varepsilon) > \frac{\lambda}{c\langle k\rangle} M\varepsilon(\mu^2 + \mu\delta)$, where $M = \begin{bmatrix} 1 \\ 2 \\ . \\ . \\ n \end{bmatrix} [1\ P(1), \quad 2\ P(2), \quad \ldots, \quad n\ P(n)]$. Also, when $R_0 < 1$, we have verified $a_1 > 0$ and $a_2 > 0$. Since $a_3 > 0$, if $a_3\,a_2 > a_4a_1$ then $b_1 > 0$. Furthermore, when $b_1\,a_1 > a_3a_0$, then $c_1 > 0$.

We got $a_4 > 0$, $a_3 > 0$, $a_2 > 0$, $a_1 > 0$, and $a_0 > 0$, so the conditions that all roots have negative real parts are $a_3\,a_2 > a_4a_1$, and $b_1\,a_1 > a_3a_0$. Thus, the Routh–Hurwitz stability conditions are satisfied, which implies that the malware-free equilibrium is locally asymptotically stable. □

Theorem 1. *When $R_0 \le 1$, the malware-free equilibrium EQ_1 is globally asymptotically stable. When $R_0 > 1$, the malware-free equilibrium EQ_1 is unstable.*

Proof. According to the Li–Muldowney linear Lyapunov function [34], we construct the following Lyapunov function for the model:

$$L(t) = E_{k,c}(t) + \frac{(\varepsilon+\alpha+\mu)}{\varepsilon} I_{k,c}(t). \tag{17}$$

Calculating the time derivative of L along the solution of model of Eq. (10), we have:

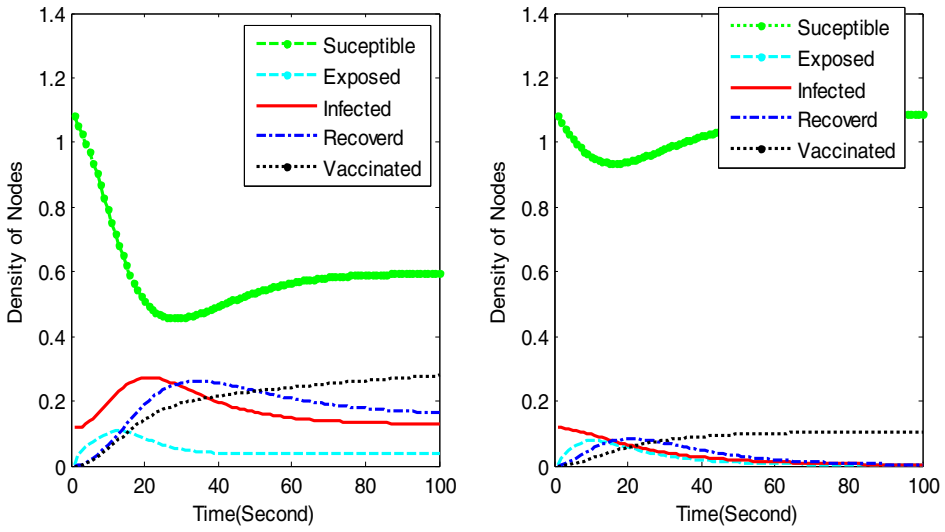

Fig. 2. Densities of five states in the malware propagation model. (a) Without assignment of diverse software packages. (b) With considering software diversity. Parameters: $\lambda = 0.3,\ \varepsilon = 0.21,\ \gamma = 0.1,\ \delta = 0.05,\ \alpha = 0.1$ and $\Lambda = \mu = 0.07$.

$$L'(t) = E'_{k,c}(t) + \frac{(\varepsilon+\alpha+\mu)}{\varepsilon} I'_{k,c}(t)$$

$$L'(t) = \left[\frac{\lambda}{c} k S_{k,c}(t) \frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\langle k\rangle/c} - (\varepsilon+\alpha+\mu)E_{k,c}(t)\right] + \frac{(\varepsilon+\alpha+\mu)}{\varepsilon}\left[\varepsilon E_{k,c}(t) - \gamma k I_{k,c}(t) \times \frac{\sum_{k=m}^{\infty}(k/c)P(k)[I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)]}{\langle k\rangle/c} - \mu I_{k,c}(t)\right];$$

$$L'(t) = \frac{\lambda}{c}\ k S_{k,c}(t) \frac{\sum_{k=m}^{\infty}(k/c)P(k)I_{k,c}(t)}{\frac{\langle k\rangle}{c}} - \frac{(\varepsilon+\alpha+\mu)}{\varepsilon}\ \gamma\ k\ I_{k,c}(t) \times \frac{\sum_{k=m}^{\infty}(k/c)P(k)\ [I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)]}{\langle k\rangle/c} - \frac{(\varepsilon+\beta+\alpha+\mu)}{\varepsilon}\ \mu\ I_{k,c}(t);$$

$$L'(t) = \frac{(\varepsilon+\alpha+\mu)\ \mu}{\varepsilon}\ (R_0 S - 1) I_{k,c}(t) - \frac{(\varepsilon+\alpha+\mu)}{\varepsilon}\ \gamma\ k\ I_{k,c}(t) \times \frac{\sum_{k=m}^{\infty}(k/c)P(k)\ [I_{k,c}(t)+R_{k,c}(t)+V_{k,c}(t)]}{\langle k\rangle/c};$$

when $R_0 \le 1$, then we have $L'(t) \le 0$. Moreover, $L'(t) = 0$ if and only if $I_{k,c}(t) = 0$.

Hence, we indicate that the malware-free equilibrium $EQ_1(1, 0, 0, 0)$ is globally asymptotically stable. When $R_0 > 1$, we analyze the stability of R_0. Using Eq. (15), the characteristic equation of the model of Eq. (10) at the malware-free equilibrium EQ_1 is as follows:

$$(\omega+\mu)(\omega+\delta+\mu)[\omega^2 + (\alpha+\varepsilon+2\mu)\omega + \alpha\mu + \varepsilon\mu + \mu^2 - G\varepsilon] = 0. \quad (18)$$

Obviously, Eq. (18) has three negative eigenvalues and one positive eigenvalue when $R_0 > 1$. Thus EQ_1 is an unstable saddle point. □

5. Numerical simulations

In this section, we perform a set of numerical simulations to verify dynamical behaviors of the malware propagation model based on Barabási–Albert (BA) in SFN and more results are reached. Using the growth and preferential attachment features of BA algorithm [35], the SFN is generated. In order to decline the malware propagation process, we consider the effect of software diversity and assign diverse software packages (C) on the SFN randomly. Moreover, we investigate the effect of vaccination in the outbreak of malware. Here, the network size is 1000 nodes, the minimum degree is 3, and the maximum degree is 149. We let $S_{k,c}(0)$, $E_{k,c}(0)$, $I_{k,c}(0)$, $R_{k,c}(0)$, and $V_{k,c}(0)$ value be 900, 0, 100, 0, 0, respectively. The numerical simulations are done by MATLAB.

5.1. The trend of malware propagation

Fig. 2 illustrates the general trends of the five states of nodes (susceptible, exposed, infected, recovered, and vaccinated) with respect to time on SFN in two cases (a) without diversification and (b) with considering diversification under the parameters setting at $\lambda = 0.3,\ \varepsilon = 0.21,\ \gamma = 0.1,\ \delta = 0.05,\ \alpha = 0.1$ *and* $\Lambda = \mu = 0.07$. When we assign diverse software packages (C) to network's nodes, the infected node cannot infect its neighbor with the different software packages because they have no common exploitable vulnerability.

In Fig. 2(a) we do not assign any diverse software packages ($C = 1$). Hence we can see that the density of infected nodes increases sharply at the beginning of the malware propagation process. With the further propagation of the malware, the density of infected nodes reaches a peak and decreases as time goes on, but they are not removed and remain in the network. The variation of the density of exposed nodes is similar to that of infected nodes, but the density of exposed nodes has much less change and the peak value of the exposed nodes smaller than that of the infected nodes. Also, we have the trend of increasing and decreasing of the density of recovered nodes. With the increment of time, the density of susceptible nodes always decreases while the density of vaccinated nodes always increases, and they achieve the balance at the end of malware propagation. Using analytical solution, and substituting the value of parameters into Eq. (13), we calculate the basic reproductive ratio $R_0 = 14.1334$ and the critical number of diverse software packages $C_{critical} = 15$. Since the value of R_0 is more than one, thus the malware infection persists in the network.

In Fig. 2(b), we assign fifteen distinct software packages ($C = 15$) to network's nodes to halt malware propagation and guarantee the basic reproductive ratio $R_0 < 1$. Substituting the value of parameters into Eq. (12), we can acquire that the basic reproductive

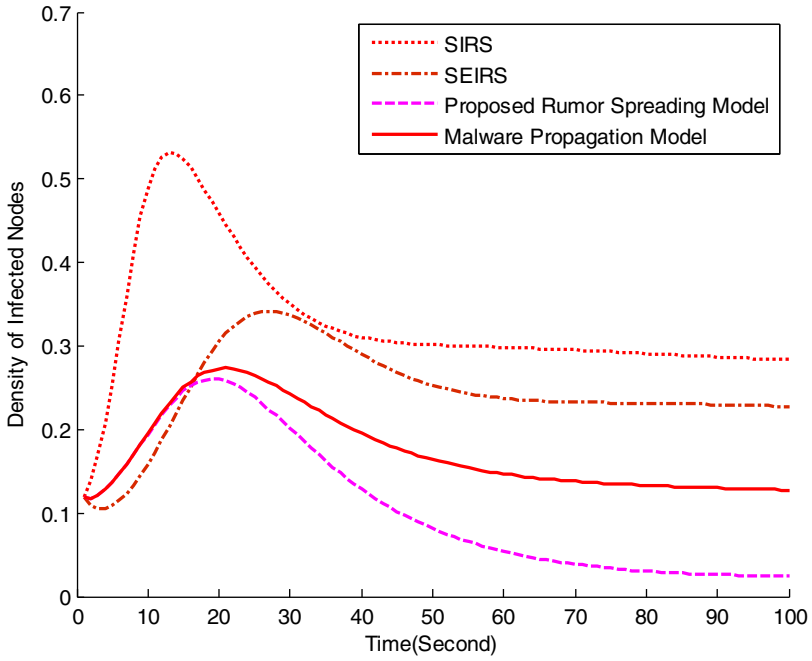

Fig. 3. Comparison between four models.

ratio R_0 is 0.9681<1. When $R_0 < 1$, the malware-free equilibrium is globally asymptotically stable, as shown in Fig. 2(b). In this figure, the malware will gradually vanish and we can obviously show that the tendency of malware propagation is depressive. Finally, the density of infected nodes is zero and malware propagation is terminated.

5.2. Comparison between four models

We compare our malware propagation model with the SIRS epidemic model [9], SEIRS epidemic model [14], and the proposed rumor spreading model [7] to verify the effectiveness of our model. The malware propagation model, which is introduced in Section 3, establishes on the proposed rumor spreading model in [7]. In Fig. 3, we first implement the SIRS epidemic model as a baseline, and then simulate the SEIRS epidemic model, and after that we implement the proposed rumor spreading model with substituting forgetting rate $\delta = 0$ and remembering rates $\eta = \zeta = 0$. Finally, we experiment our malware propagation model. All models use the same parameters without considering diversification. The initial values of parameters are the same as in Fig. 2. Also, these models are implemented on BA network.

Fig. 3 describes the density of infected nodes for each of the four spreading models. The simulation results demonstrate a considerable decrease and a reduced spreading for the infected nodes in malware propagation and rumor spreading models in comparison with the SIRS and SEIRS epidemic models. In our malware propagation model, the peak value of infected nodes is nearly equal to the rumor spreading model. But since in our model, we add a transmission from recovered state to susceptible state, thus recovered nodes can become susceptible again and propagate malware infection. As shown in Fig. 3, the trend of decreasing of the density of infected nodes in the rumor spreading model is faster than the malware propagation model. Generally, in comparison with the SIRS and SEIRS epidemic models, our model is more appropriate for simulation of malware spreading processes. Thus, it is effective.

5.3. The effect of software diversity

Fig. 4 represents the dynamical behaviors of nodes with considering diverse software packages C = 5, 8, 15 respectively. The initial values of parameters are the same as in Fig. 2. In this figure, we can see the variation trend of the densities of susceptible, exposed, infected, recovered, and vaccinated nodes over time under diversification.

As shown in Fig. 4, with assigning diverse software packages to network nodes, we observe a reduced propagation spread for the infected nodes. With increasing the number of diverse software packages, the malware propagation speed is reduced and the value of the basic reproductive ratio (R_0) goes on decreasing. Fig. 4 shows the impact of C on R_0 that is derived in Section 4.1. With substituting the values of parameters into Eq. (12) under different software packages (C = 3, C = 8 and C = 15) the value of R_0 is calculated 2.9541, 1.7515, and 0.9681. In C = 3 and C = 8, the value of R_0 is obtained more than one. While in C = 15 we have R_0 < 1. As stated earlier, when $R_0 < 1$, the density of infected nodes will gradually disappear from the network, and when $R_0 > 1$ the malware infection persists in the network. This result is observed in Fig. 4.

By comparison between the simulation curves in Fig. 4, and the numerical curves in Fig. 2 (b), under $C = 15$, it can be shown that they correspond with each other. This result indicates the correctness of the model.

5.4. The effect of vaccination

Fig. 5 represents the impact of varying the vaccinated rates (which change between 0.1 and 0.7) on malware propagation process without considering diversification. The initial values of parameters are the same as in Fig. 2. As shown in Fig. 5, the larger the vaccinated rate α is, the slower the malware propagation speed. This is due to the fact that the vaccination has a powerful effect in controlling malware propagation and reducing the density of infected nodes. But since we do not assign any software pack-

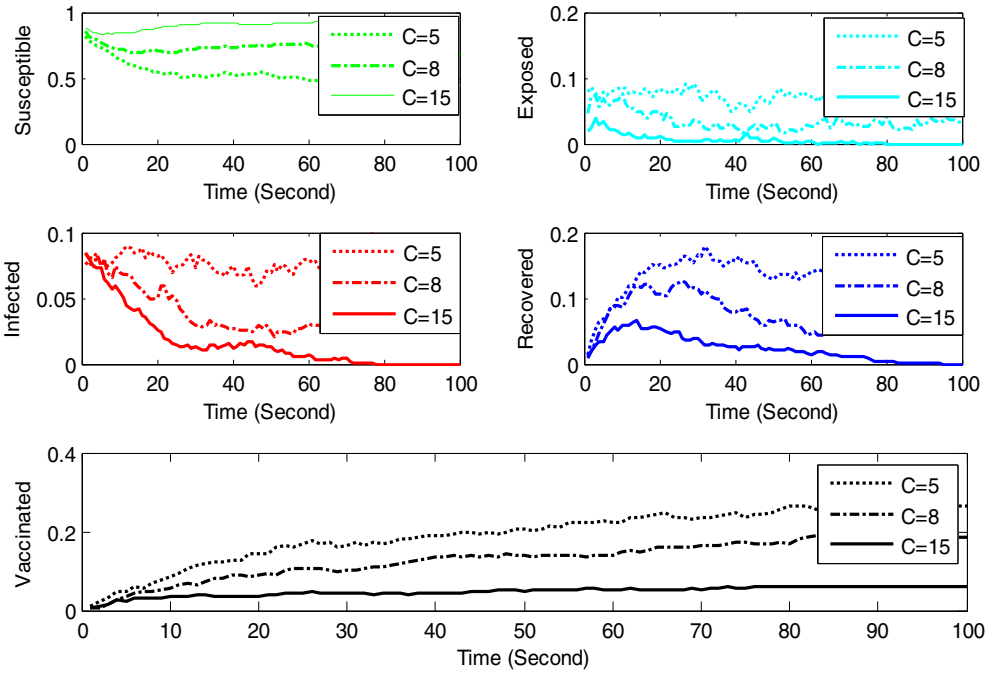

Fig. 4. Densities of susceptible, exposed, infected, recovered, and vaccinated nodes with respect to time under different software packages C = 5, 8, 15. *Parameters:* $\lambda = 0.3$, $\varepsilon = 0.21$, $\gamma = 0.1$, $\delta = 0.05$, $\alpha = 0.1$ *and* $\Lambda = \mu = 0.07$.

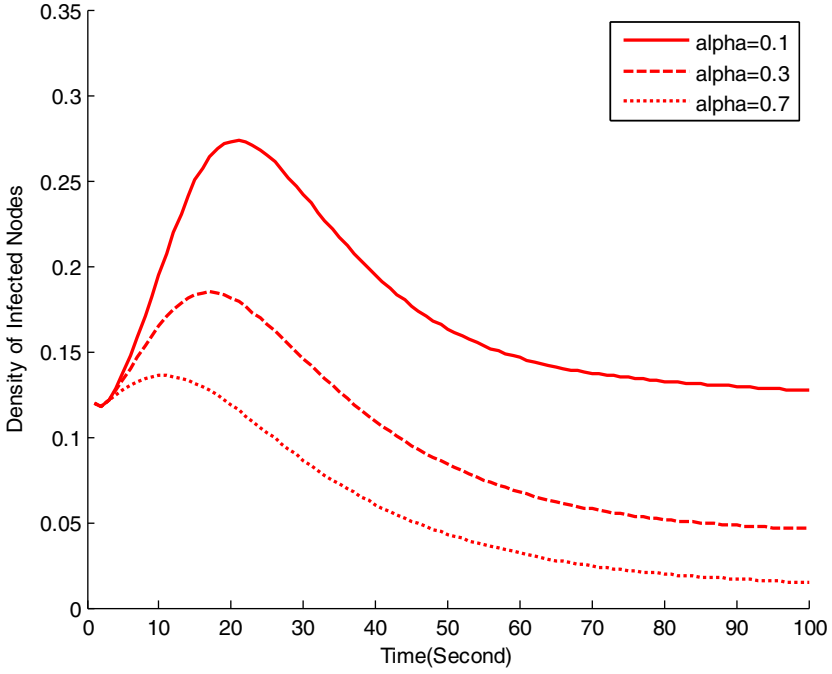

Fig. 5. Density of infected nodes over time under different vaccinated rates α. *Parameters:* $\lambda = 0.3$, $\varepsilon = 0.21$, $\gamma = 0.1$, $\delta = 0.05$ *and* $\Lambda = \mu = 0.07$.

Table 2
The values of R_0 under different vaccinated rates α in two cases $C = 1$ and $C = 8$.

	$C = 1$			$C = 8$		
	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.7$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.7$
R_0	14.3654	9.4121	5.4226	3.7121	1.2223	0.7402

ages to network nodes, thus the infected nodes exist in the network and the malware persists at an endemic equilibrium state.

Table 2 shows the effects of vaccinated rates α and diverse software packages C on R_0, which is calculated in Section 4.1. The initial values of parameters are the same as in Fig. 5. As shown in Table 2, without considering software diversity (i.e., $C = 1$) under different vaccinated rates $\alpha = 0.1$, 0.3 and 0.7 the values of R_0 are calculated more than one (with substituting the values of parameters into Eq. (12)). This result shows that although, the increasing of the vaccinated rate is effective in decreasing of the R_0 value and reducing the outbreak of malware, but it does not terminate the malware infection, which persists at the endemic level ($R_0 > 1$). This result is demonstrated in Fig. 5. With considering software diversity ($C = 8$) under different vaccinated rates, the R_0 value is decreased significantly. In $\alpha = 0.7$, the value of R_0 is calculated less than one and the malware-free equilibrium is obtained. Hence, the

combination of vaccination and software diversity is more effective in decreasing the R_0 value and reducing the malware propagation speed.

6. Conclusions

In this paper, we proposed a malware propagation model based on a rumor spreading model to study the dynamics of malware spreading in scale-free networks (SFNs). The proposed model considers the assignment of diverse software packages to network nodes to prevent malware propagation. We have used the susceptible–exposed–infectious–recovered–susceptible with a vaccination state (SEIRS-V) and analyzed the conditions for the stability of the malware-free equilibrium. We obtained the basic reproductive ratio (i.e., R_0), and determined that the dynamics of the model is completely governed by R_0. Furthermore, we derived the critical number of software packages based on R_0 to guarantee that a malware infection does not become an epidemic in SFNs. As the number of distinct software packages (i.e., C) augments gradually, the value of R_0 declines. Theoretical analysis presents that basic reproductive ratio is appreciably dependent on diversification and the network topology.

We have also conducted a series of numerical simulations to confirm the correctness of the analytical results. We have compared the proposed model with existing ones and showed that our model provides a noticeable decrease in the infected nodes compared with other models (i.e., SIRS and SEIRS models), and also a decrease in the spreading speed. Moreover, the simulation results represented that the malware propagation is governed by the number of diverse software packages and the vaccinated rate. This can be used as a guideline to control malware propagation process and devise defense strategies.

In the future, we will focus on investigating more complex malware propagation model to control malware spreading in SFNs. We will also extend the study of software diversity through automatic program transformation techniques for the assignment of diverse software packages to network nodes.

References

[1] J. Wang, L. Zhaoa, R. Huang, 2SI2R rumor spreading model in homogeneous networks, Physica A 413 (2014) 153–161.
[2] A. Singh, Y.N. Singh, Nonlinear spread of rumor and inoculation strategies in the nodes with degree dependent tie strength in complex networks, Acta Phys. Polon. B 44 (1) (2013) 5–28.
[3] J. Wang, L. Zhao, R. Huang, SIRaRu rumor spreading model in complex networks, Physica A 398 (2014) 43–55.
[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: structure and dynamics, Physics Reports 424 (2006) 175–308.
[5] A. Gherbi, R. Charpentier, Diversity-based Approaches to software systems security, Commun. Comput. Inf. Sci. 259 (2011) 228–237.
[6] A. Gherbi, R. Charpentier, M. Couture, Software diversity for future systems security, J. Defense Softw. Eng. 25 (5) (2011) 10–13.
[7] S. Hosseini, M. Abdollahi Azgomi, A. Torkaman Rahmani, Dynamics of a rumor spreading model with the diversity of configurations in scale-free networks, Int. J. Commun. Syst. Wiley 28 (2015) 2255–2274.
[8] R. Albert, A.L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (1) (2002) 1–8.
[9] Y. Moreno, R. Pastor-Satorras, A. Vespignani, Epidemic outbreaks in complex heterogeneous networks, Eur. Phys. J. B 26 (4) (2002) 521–529.
[10] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Lett. 86 (14) (2001) 3200–3203.
[11] M.E. Newman, Spread of epidemic disease on networks, Phys. Rev. E 66 (2002) 016128.
[12] C.H. Li, C.C. Tsai, S.Y. Yang, Analysis of epidemic spreading of an SIRS model in complex heterogeneous networks, Commun. Nonlin. Sci. Numer. Simul. 19 (4) (2014) 1042–1054.
[13] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, Alessandro Vespignan, Epidemic processes in complex networks, Phys. Soc. (2014) 1–61.
[14] S. Hosseini, M. Abdollahi Azgomi, A. Torkaman Rahmani, Malware propagation modeling considering software diversity and immunization, J. Comput. Sci. Elsevier 13C (2016) 49–67.
[15] Y.A. Kuznetsov, C. Piccardi, Bifurcation analysis of periodic SEIR and SIR epidemic models, J. Math. Biol. 32 (1994) 109–121.
[16] A. Sanatinia, S. Narain, G. Noubir, Wireless spreading of Wifi Aps infections using WPS flaws: an epidemiological and experimental study, in: IEEE Conference on Communications and Network Security (CNS), 2013, pp. 430–437.
[17] Y. Zhang, C. Chen, A rumor spreading model considering latent state, in: Proceedings of the Eighth International Conference on Management Science and Engineering Management, Springer, 2014, pp. 155–162.
[18] D. Daley, D.G. Kendal, Stochastic rumors, IMA J. Appl. Math. 1 (1965) 42–55.
[19] A. Singh, R. Kumar, Y.N. Singh, Rumor dynamics with acceptability factor and inoculation of nodes in scale free networks, in: Proceedings of the 8th Signal Image Technology and Internet Based Systems, IEEE, 2012, pp. 798–804.
[20] E. Lebensztayn, F.P. Machado, P.M. Rodrígue, On the behavior of a rumor process with random stifling, Environ. Model. Softw. 26 (2011) 517–522.
[21] D. Maki, M. Thomson, Mathematical Models and Applications, Prentice-Hall, Englewood Cliffs, 1973.
[22] M. Nekovee, Y. Moreno, G. Bianconi, M. Marsili, Theory of rumor spreading in complex social networks, Physica A 374 (2007) 457–470.
[23] L. Zhao, J. Wang, Y. Chen, Q. Wang, J. Cheng, H. Cui, SIHR rumor spreading model in social networks, Physica A 391 (2012) 2444–2453.
[24] A.J. O'Donnell, H. Sethu, Software diversity as a defense against viral propagation: models and simulations, in: Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation, IEEE Computer Society, 2005, pp. 247–253.
[25] A.J. O'Donnell, H. Sethu, On achieving software diversity for improved network security using distributed coloring algorithms, in: Proceedings of the 11th ACM Conference on Computer and Communications Security, ACM, 2004, pp. 121–131.
[26] L. Feng, X. Liao, Q. Han, H. Li, Dynamical analysis and control strategies on malware propagation model, Appl. Math. Model. 37 (2013) 8225–8236.
[27] M. Marsono, A. Mohammed, Analysis of internet malware propagation models and mitigation strategies, Int. J. Comput. Netw. Wireless Commun. 2 (2012) 16–20.
[28] O. Diekmann, J. Heesterbeek, J.A. Metz, On the definition and the computation of the basic reproduction ratio r0 in models for infectious diseases in heterogeneous populations, J. Math. Biol. 28 (1990) 365–382.
[29] Y. Wang, J. Cao, Global dynamics of a network epidemic model for waterborne diseases spread, Appl. Math. Comput. 237 (2014) 474–488.
[30] P. Driessche, J. Watmough, Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission, Math. Biosci. 180 (2002) 29–48.
[31] C. Jin, X. Wang, Analysis and control stratagems of flash disk virus dynamic propagation model, Secur. Commun. Netw. 5 (2) (2012) 226–235.
[32] F. Wang, Y. Zhang, C. Wang, J. Ma, S. Moon, Stability analysis of a SEIQV epidemic model for rapid spreading worms, Comput. Secur. 29 (4) (2010) 410–418.
[33] B.K. Sahu, M.M Gupta, B. Subudhi, Stability analysis of nonlinear systems using dynamic-routh's stability criterion: a new approach, in: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, 2013, pp. 1765–1769.
[34] M. Li, J. Muldowney, Global stability for the SEIR model in epidemiology, Math. Biosci. 125 (1995) 155–164.
[35] A.-L. Barabási, R. Albert, H. Jeong, Scale-free characteristics of random networks: the topology of the world-wide web, Physica A 281 (2000) 69–77.

Soodeh Hosseini received B.S. degree in computer science (2004) from Shahid Bahonar University of Kerman and M.S. degree in computer engineering (software) (2007) from Iran University of Science and Technology, where she is currently a Ph.D. candidate. Her main research interests include analytical modeling, computer simulation and software security. She has published several papers in international journals and conferences.

Mohammad Abdollahi Azgomi received B.S., M.S. and Ph.D. degrees in computer engineering (software) (1991, 1996 and 2005, respectively) from Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. His research interests include modelling and evaluation of security, privacy and trust, and dependable and secure software development. Dr. Abdollahi Azgomi is currently an associate professor at School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.