A Study of Malware Propagation via Online Social Networking

Mohammad Reza Faghani and Uyen Trang Nguyen

Abstract The popularity of online social networks (OSNs) have attracted malware creators who would use OSNs as a platform to propagate automated worms from one user's computer to another's. On the other hand, the topic of malware propagation in OSNs has only been investigated recently. In this chapter, we discuss recent advances on the topic of malware propagation by way of online social networking. In particular, we present three malware propagation techniques in OSNs, namely cross site scripting (XSS), Trojan and clickjacking types, and their characteristics via analytical models and simulations.

1 Introduction

Online social networks (OSNs) such as Facebook, Twitter and MySpace have provided hundreds of millions of people worldwide with a means to connect and communicate with their friends, family and colleagues geographically distributed all around the world. The popularity and wide spread usage of OSNs have also attracted attackers and hackers who would use OSNs as a platform to propagate automated worms from one user's computer to another's.

The population of potential victims of web-based malware is much larger than that of other types of worms due to the popularity of the world wide web. In addition, web-based worms are not banned through web proxies and network address translation (NAT) processes. Research has shown that web-based malware can propagate much faster than traditional malware [13]. As a matter of fact, the first OSN

Mohammad Reza Faghani
York University, Department of Computer Science and Engineering, Toronto, ON, Canada, M3J 1P3 e-mail: faghani@cse.yorku.ca

Uyen Trang Nguyen
York University, Department of Computer Science and Engineering, Toronto, ON, Canada, M3J 1P3 e-mail: utn@cse.yorku.ca

worm that hit MySpace in 2005 by exploiting a cross site scripting vulnerability in a MySpace web application infected about one million victims within 24 hours [13]. As Figure 1 shows, this worm, which was named Samy, propagated much faster other traditional computer worms.

The topic of malware propagation in OSNs has only been investigated recently [9, 8, 10, 29, 23, 28]. The objective of this chapter is to discuss recent advances on this topic. In particular, we present three malware propagation techniques in OSNs, namely XSS, Trojan and clickjacking types, and their characteristics via analytical models and simulations.

The remainder of this chapter is organized as follows. In Section 2, we briefly describe three types of malware propagating via online social networking. In Section 3, we discuss the characteristics of OSNs and algorithms used for generating simulated OSN graphs. A simulation-based study of malware propagation in OSNs is presented in Section 4. In Section 5, we review analytical models characterizing the propagation of malware in OSNs. We discuss OSN malware countermeasures in Section 6 and related work in Section 7. We then summarize the chapter in Section 8.

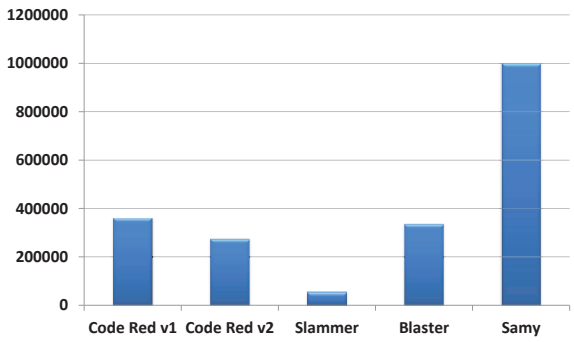


Fig. 1: Total number of infections caused by different computer worms in 20 hours

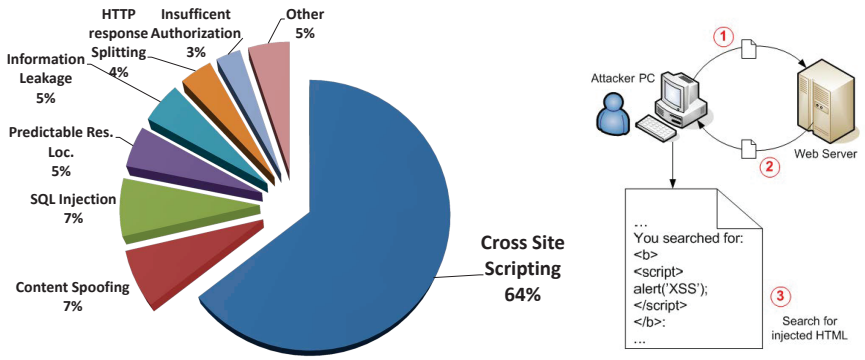
2 Different Types of Malware

In this section, we briefly discuss the main characteristics of different types of malware propagating through OSNs. There exist currently three different types of OSN malware: cross site scripting, Trojan and clickjacking worms.

2.1 Cross Site Scripting Worms

Cross site scripting (XSS) is a security flaw to which many web applications are vulnerable [13, 20]. The graph in Figure 2(a) shows the distribution of web application vulnerabilities, among which XSS is the most common threat. While XSS is a common vulnerability in web applications, its threat becomes more noticeable due to the combination of HTML and Asynchronous JavaScript and XML (AJAX) technologies. AJAX allows a browser to issue HTTP requests on behalf of the user. Thus there is no need for an attacker to trick the user into clicking a malicious link. This technique provides facility for malware writers to create XSS worms.

There are two types of XSS attacks: persistent and non-persistent [20]. In persistent attacks (also known as stored attacks), the injected code is permanently stored on a target server as HTML text in a database, a comment field, or messages posted on online forums. A victim's computer then accesses the malicious code on the server when it retrieves the stored information via the web browser. Non-persistent attacks (also known as reflective attacks) are the more common type of XSS attacks. In this case, the injected code is sent back to the visitor by the server in an error message, a search result, or any other type of response that reflects some or all of the user's input in the result (Figure 2(b)). Since the reflected response contains the malicious code, the visitor's browser will interpret the code, execute the malware, and infect his/her computer. .



(a) Distribution of Vulnerabilities in web applications (b) Reflective XSS attack [13]

Fig. 2: Cross Site Scripting can be used to create self propagating worms

An XSS worm, also known as a cross site scripting virus, is a malicious code that propagates itself automatically among visitors of a website in an attempt to progressively infect other visitors. XSS worms can be considered as a hybrid of stored and reflected XSS attack. This type of worms has the ability to copy itself

from a source to another part of the Internet using existing XSS vulnerabilities of web applications.

An XSS worms infect members of a social network in two steps. The worm creator first adds the malicious payload to his profile, e.g., in the form of a link. Subsequently, any person who visits this profile will get infected and the malicious payload will be added to the visitor's profile due to an AJAX technique and an XSS flaw. The visitor's profile then becomes an infectious profile, which allows the worm to propagate as a new infection source [9, 8].

2.2 Trojan Malware

The best known OSN Trojan worm is Koobface [15], which was first detected in 2008. It spread itself in both MySpace and Facebook by sending messages with interesting topics using social engineering techniques to deceive people into opening the messages. Such a message directed the recipients to a third-party website unaffiliated with Facebook where they were prompted to download what was claimed to be an update of the Flash player. If they downloaded and executed the file, they would infect their computers with Koobface. The infected machine turned into a zombie or a bot. Moreover, the owner of the infected profile unknowingly sent out messages to all people on his/her friend list, allowing the worm to propagate further in the network.

2.3 Clickjacking Malware

Clickjacking worms are also known as "likejacking" or "user interface (UI) redressing". A clickjacking attacker creates a website that shows a counterfeit YouTube video player, or other graphical icons, and invites the victim to click on a play button to view the video. What really happens is that the victim is clicking a Facebook "Like" button that has been hidden beneath the images using a method of coding called UI redressing. What the victim has just "Liked" is then displayed on his wall, which in turn may attract his/her friends to click on that link and become new sources of infection (Figure 3). Since Trojan and clickjacking worms operate and propagate in a similar manner, we consider them as one type in this chapter.

Clickjacking worms could be combined with Trojan malware to create a new hybrid type but, to the best of our knowledge, no such malware has been created or deployed yet.

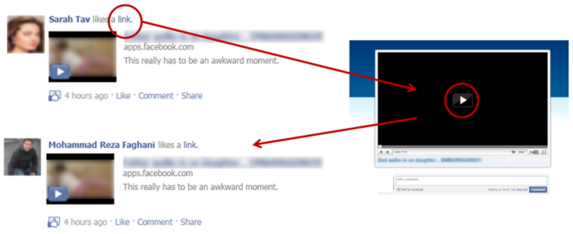


Fig. 3: Clickjacking technique used to spread spam messages that may lead to malicious software

3 Characteristics of Online Social Networks

An OSN can be represented by an equivalent graph in which each vertex (or node) represents a person and a link between two vertices indicates the existence of a relationship between the two respective persons. To simplify the discussions in this chapter, we generalize relationships between OSN users as friendship. (In some OSNs such as LinkedIn, relationships can be colleagues or business contacts). Studies have shown that real-world social networks are highly clustered small-world networks [26] with a degree distribution often following a power law distribution. The characteristics of *online* social networks can be summarized as follows [30, 7, 14]:

1. An OSN typically has a low average network distance, approximately equal to $\log(s)/\log(d)$, where s is the number of vertices (people), and d is the average vertex degree of the equivalent graph.
2. Online social networks typically show a high clustering property, or high local transitivity. That is, if person A knows B and C, then B and C are likely to know each other. Thus A, B and C form a friendship triangle. Let k denote the degree of a vertex v. Then the number of all possible triangles originated from vertex v is $k(k-1)/2$. Let f denote the number of friendship triangles of a vertex v in a social network graph. Then the clustering coefficient $C(v)$ of vertex v is defined as $C(v) = 2f/(k(k-1))$. The clustering coefficient of a graph is the average of the clustering coefficients of all of its vertices. In a real OSN, the average clustering coefficient is about 0.1 to 0.7.
3. Node degrees of a social network graph tend to be, or at least approximately, power-law distributed. The node degree of a power-law topology is a right-skewed distribution with a power-law Complementary Cumulative Density Function (CCDF) of $F(k) \propto k^{-\alpha}$, which is linear on a logarithmic scale. The power law distribution states that the probability for a node v to have a degree k is $P(k) \propto k^{-\alpha}$, where α is the power-law exponent.

There exist few algorithms that can generate social network graphs with the above characteristics [7, 14, 6]. For the simulations reported in this chapter, we used the algorithm proposed by Holme and Beom [14], due to the fact that it can be

fine tuned to generate a social network graph with a desired clustering coefficient and power law distribution of node degrees.

In one of our experiments, we evaluated the speed of malware propagation as a function of clustering coefficients. For this experiment, we would need to vary the clustering coefficient while keeping other parameters of the network graph such as the maximum and average node degrees constant. To create such similar graphs with different clustering coefficients, we would need random graph generation algorithms such as random rewiring or the algorithm by Viger and Latapy [25]. In this section, we discuss the algorithm by Holme and Beom along with these random graph generation algorithms. We call a random graph generated based on an OSN graph an *equivalent random graph* (ERG).

3.1 Holme's Social Network Graph Generation Algorithm

This algorithm is based on the algorithm proposed by Barabasi and Albert [2], which we term the BA algorithm. The objective of the BA algorithm is to create graphs with node degrees following power law distributions. These graphs have short average network distances typical of OSNs, but they may not have high clustering coefficients, between 0.1 to 0.7, to faithfully model social network graphs [14]. This motivated Holme and Beom to modify the BA algorithm to generate graphs having high clustering coefficients typical of OSNs.

The BA algorithm works as follows:

1. The initial condition: A graph consists of m_0 vertices and no edges.
2. The growth step: One new vertex v with m edges is added to the above graph at every time step. Time t is identified as the number of time steps.
3. The preferential attachment (PA) step: Each of the m edges incident on v is then attached to an existing vertex u with the probability P_u defined as follows:

$$\frac{k_u}{\sum_{i \in v} k_i} \tag{1}$$

In the above equation, k_i represents the degree of node i, and V is the set of vertices of the current graph. The growth step is iterated N times, where N is the total number of vertices (users) in the final OSN graph. Every time a vertex v with m edges is added to the network, the PA step is performed m times, once for each of the m edges incident on v. After t time steps, the BA network graph will contain $m_0 + t$ vertices.

To increase the clustering coefficient, Holme and Beom suggested a new step called triad formation (TF). If, in a PA step, an edge between u and v is formed, then a TF step will attempt to add another edge between v and an arbitrary neighbor w of u. If all neighbors of u have already been connected to v, the TF step is skipped and a new PA step will start.

In each iteration, a PA step is first performed: a vertex v with m edges is added to the existing network. Then a TF step is executed with probability P_t. The average number of the TF trials per added vertex is given by $m_t = (m-1) \times P_t$ which is a control parameter in Holme's algorithm. It has been shown that the degree distribution of any graph generated by Holme's algorithm will have node degrees following a power law distribution with $\alpha = 3$.

We used Holme's algorithm to generate a graph that has the characteristics of a social network and the following parameters: $\alpha = 3$, $N = 10000$, $m_0 = 3$, $m = 3$ and $m_t = 1.8$. The parameters of the resulting social network graph are listed in Table 1.

Table 1: Parameters of the simulated OSN and its ERG

	Graphs	
	OSN graph (Holme and Beom)	ERG (Viger and Latapy)
Parameter	**Value**	**Value**
Number of vertex (people)	10000	10000
Number of edges	29990	29990
Average clustering coefficient	0.14	0.0035
Average shortest path length	5.13	4.4
Network diameter	10	8
Maximum node degree	190	190
Average node degree d	5.99	5.99
$\log(n)/\log(d)$	5.14	5.14

As Table 1 shows, the synthesized OSN graph satisfies all the three required characteristics of an OSN. The average shortest path length of the graph is 5.13, which is less than $\frac{\log n}{\log d} = 5.14$. The clustering coefficient is moderate, approximately 0.14. The degrees of the vertices follow a power law distribution, as shown in Figure 4.

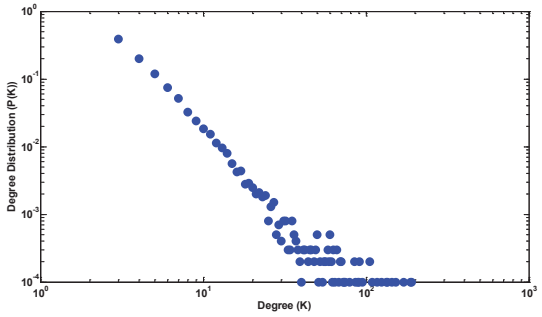


Fig. 4: The degrees of the vertices of the resulting graph follow a power law distribution.

3.2 Random Graph Generation Algorithms

To evaluate the impacts of clustering coefficients on worm propagation, we would need to keep the other parameters of a graph such as the maximum and average node degrees constant while varying the clustering coefficient. Such graphs with different clustering coefficients can be generated as equivalent random graphs (ERG). Given a network graph, an ERG with the same node degree distribution can be generated using random rewiring or the algorithm proposed by Viger and Latapy [25].

In the random rewiring scheme, we randomly select a pair of edges and "substitute" the edges as shown in Figure 5. The random selection and substitution are done until we obtain the desired clustering coefficient.

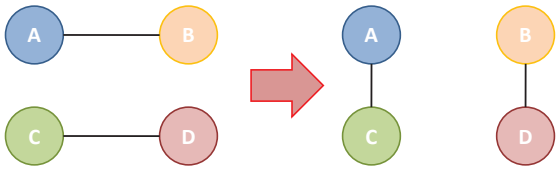


Fig. 5: Random rewiring technique to generate equivalent random graphs

The random rewiring algorithm generates ERGs that have strong correlations to the original graph. We would want random graphs that are as versatile as possible while maintaining the same node degree distribution of the original graph. Therefore, we chose the algorithm proposed by Viger and Latapy [25] to generate equivalent random graphs for our simulations. Such a random graph has the same degree distribution as the original network graph, but different characteristics such as a different clustering coefficient, average shortest path length or network diameter. The parameters of the OSN created earlier and its ERG are listed in Table 1.

Following are some observations obtained from comparing the ERG and the corresponding OSN graph (Table 1). The average clustering coefficient of the original OSN graph is about 40 times higher than that of the similar random graph, 0.14 vs. 0.0035. This reflects the high clustering characteristic of OSNs. Also, the average shortest path length of the original OSN graph is longer than that of the ERG, 5.13 vs. 4.4. The network diameter of the OSN is 10 hops compared to eight hops in the ERG. These differences are due to the small-world phenomenon described by Watts [26].

4 Simulation of Malware Propagation in Online Social Networks

In this section, we present our simulation results on malware propagation in OSNs. The simulation software is implemented in MATLAB. The simulation is of discrete-

event type, consisting of discrete time slots. In each time slot, a user (node) is chosen randomly and the user will perform an action such as visiting a profile (for XSS malware), or executing the malware (for Trojan/clickjacking malware). In all the simulations presented below, we use the synthesized OSN created using Holme's algorithm [14] and its ERG created using the algorithm by Viger and Latapy [25] whose parameters are listed in Table 1, unless otherwise stated. The performance metric is the total number of infected profiles (users) over time, assuming an initial number of infected profiles of one, unless otherwise stated. Each data point in the result graphs is the average of 100 runs, each with a different random seed. We measure the total number of infected profiles over time as functions of the following parameters:

- **Visiting-friends probability q.** The probability that a user visits his/her friends' profiles versus strangers' profiles. A friendship exists between two users u and v if there is an edge connecting nodes u and v in the network graph.
- **Graph structure.** A graph can be an original social network graph created using Holme's algorithm, or an equivalent random graph generated based on an original social network graph using the algorithm by Viger and Latapy [25].
- **Probability p of executing the malware by a user.** For Trojan or clickjacking worms, this is the probability that a user will click on a malicious link and execute the malware.
- **Node degree threshold for visiting friends vs. strangers.** Let M be the maximum node degree in the network, and K_c be a threshold factor $(0 \leq K_c \leq 1)$. Nodes with degrees less (more) than the threshold K_cM will visit their friends less (more) frequently than strangers. That is, users whose numbers of friends (node degrees) are lower than the threshold K_cM will visit their friends with probability $q \leq 0.5$ and visit strangers with probability $1-q$.
- **Initial number of infected profiles (users).**

We first present the simulation results on XSS worm propagation, followed by the results on Trojan and clickjacking worms. As mentioned earlier, Trojan and clickjacking worms spread in a similar fashion since both send the malware globally to all friends of a user. They are thus treated as one type in our simulations.

4.1 Simulation of XSS Worm Propagation

In order for an XSS worm to propagate, a vulnerable user has to visit an infectious profile (user) to get infected. The user's vulnerability is determined by whether or not the user's web browser is able to execute the malicious script. (A browser may not be able to execute the script because there is an Anti-Virus active or the user has disabled JavaScript using special add-ons such as NoScript for a Firefox browser.)

In the simulation of XSS worm propagation, an event is defined as a single visit to the social network website. If the visitor is vulnerable and visits an infected profile, then the visitor will get infected.

4.1.1 Effects of Visiting-Friends Probability q

In each time unit, an uninfected user is chosen randomly using a uniform distribution. The user then visits one of his/her friends with probability q, or picks randomly a stranger to visit with probability of $1-q$. We assume that all users have the same visiting-friends probability q. Figure 6 shows the trend of XSS worm propagation for different values of q from 0.1 to 1. The simulation results show that if people visit their friends more often than strangers, the propagation speed will be slower. This confirms the analytical model proposed by Faghani and Saidi in [9], which will be described in Section 5.2.

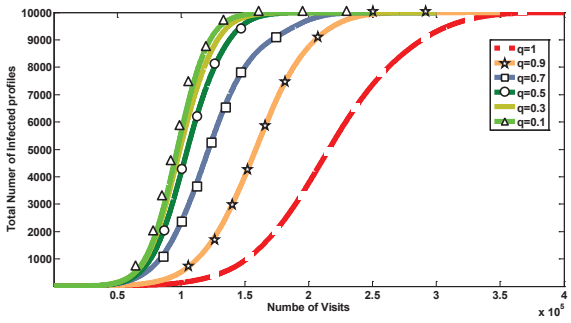


Fig. 6: XSS worm propagation for different values of probability q

4.1.2 Effects of the Network Graph Structure

In this experiment, we examine the effects of the clustering coefficient on the speed of XSS worm propagation using simulations. We use the social network graph and its ERG whose parameters are listed in Table 1. We assume a visiting-friends probability $q = 0.9$ on both network graphs. Figure 7 shows the trends of XSS worm propagation for both graphs. Although both networks share the same visiting-friends probability and other parameters (e.g., maximum and average node degrees), their results are different. The propagation is slower in the original OSN graph (i.e., the small-world graph) thanks to its highly clustered structure, which makes the malware circulate within a cluster for a while before making its way to other parts of the network. In short, the high clustering characteristic of OSNs helps slow down the propagation of malware. (However, this fact has not been considered in existing analytical models of malware propagation in OSNs.)

We repeated the above experiment, but changed the visiting-friends probability to $q = 0.1$. The results in Figure 8 show that both networks experienced the same XSS worm propagation speed in this case. This implies that the network topology is meaningful only with high probabilities of visiting friends. A low visiting-friends

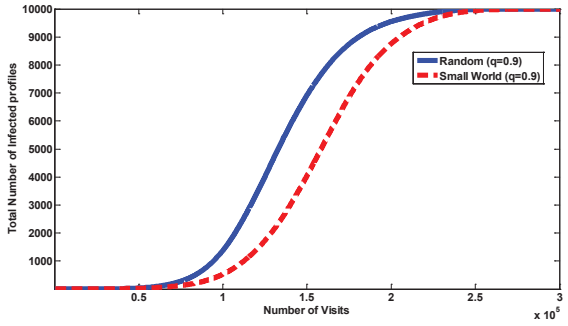


Fig. 7: XSS worm propagation in the OSN vs. its ERG with $q = 0.9$

probability means that a malware is distributed from one community to another in the network more often than being contained within that community. Therefore, the highly clustered structure of the OSN does not help in this case, and the XSS worm propagation speed is similar to that in the ERG network.

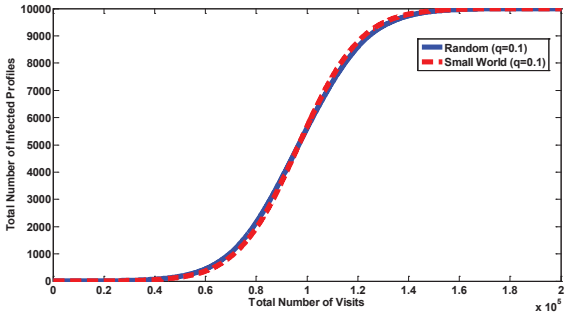


Fig. 8: XSS worm propagation in the OSN vs. its ERG with $q = 0.1$

4.1.3 Effects of Node Degree Threshold for Visiting Friends vs. Strangers

Initially when users join an online social network, they start looking for friends by visiting different profiles. Hence, they visit strangers more frequently than their friends. After establishing friendships with a set of people, a user tends to socialize with his/her friends and thus visit their profiles more often than strangers'.

In this simulation, we assume that a fraction of people with low numbers of connections will visit strangers more frequently than their friends. Let M be the maximum node degree in the network, and K_c be a threshold factor $(0 \leq K_c \leq 1)$. Nodes with degrees less (more) than the threshold K_cM will visit their friends less (more)

frequently than strangers. In this experiment, users whose numbers of friends (node degrees) are lower than the threshold K_cM will visit their friends with probability q_1, where q_1 has a normal distribution with $\mu = 0.25$ and $\sigma = 0.05$. The other users (i.e., those whose node degrees are higher than the threshold K_cM) will visit their friends with probability q_2, where q_2 has a normal distribution with $\mu = 0.75$ and $\sigma = 0.05$. The trends of worm propagation for are $K_c = 0.1$ and $K_c = 0.5$ (equivalent to a threshold of 19 friends and 95 friends, respectively, given the network in Table 1 which has a maximum node degree of 190) are shown in Figure 9.

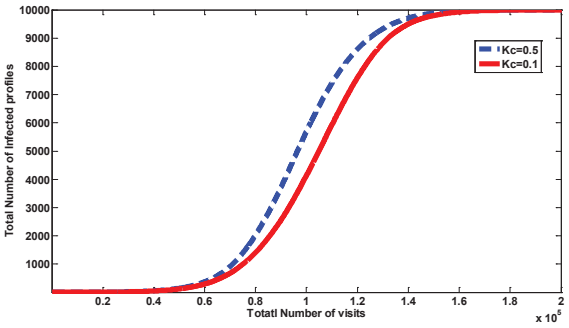


Fig. 9: XSS worm propagation given $K_c = 0.1$ and $K_c = 0.5$

The results show that the propagation is slower when $K_c = 0.1$. The reason is that when $K_c = 0.1$ there are 281 users that have more than 19 friends in contrast to 12 users having more than 95 friends when $K_c = 0.5$. When $K_c = 0.1$, more users visit their friends more frequently than when $K_c = 0.5$, which leads to slower worm propagation.

4.2 Simulation of Trojan and Clickjacking Malware Propagation

In each time step, a user is randomly selected, who will check his/her messages sent via the OSN messaging system. This action is also called a *visit* or an *event*. With probability p, the user follows the malicious link in the message and executes the malware. Once the user gets infected, the Trojan code sends a similar message containing the malicious link to all of the user's friends (or posts a message on his/her wall as done by a Facebook clickjacking worm).

4.2.1 Effects of the Probability p of Executing the Malware by a User

The trend of Trojan malware propagation is shown in Figure 10 for different distributions of probability p listed in Table 2. The graphs demonstrate that the higher the

probability of executing the malware, the faster the worm propagates. Given p with a uniform distribution having $\mu = 0.5$ from $p = 0$ to $p = 1$, the propagation is very slow because very few people open the messages and follow the malicious links.

Table 2: Probabilities p in Trojan worm propagation

Distribution	**Value**
Constant	$p = 0.5$
	$p = 0.7$
Normal distribution	$\mu = 0.5$ and $\sigma = 0.12$
	$\mu = 0.7$ and $\sigma = 0.075$
Uniform distribution	Range $[0, 1]$ with $\mu = 0.5$
	Range $[0.4, 1]$ with $\mu = 0.7$

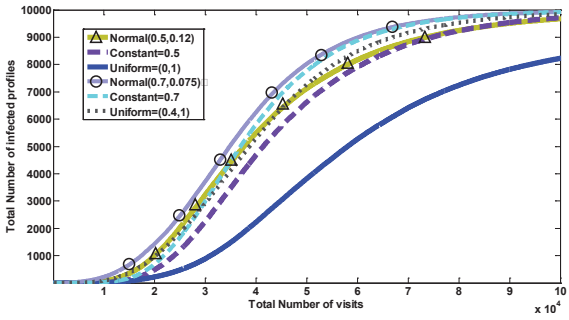


Fig. 10: Trojan worm propagation given different distributions of p

Faghani et al. [10] show that increasing the value of p will speed up the propagation *exponentially*. The authors used a synthesized OSN with 10000 nodes, 29991 edges, and a clustering coefficient of 0.16. They then measured the number of visits (events) needed to get all the users infected as a function of probability p, starting with one infected user. Their result, shown in Figure 11, indicates that the propagation speed increases exponentially with higher probability of p. The result highlights the importance of raising awareness of malware threats among OSN users.

4.2.2 Effects of the Clustering Coefficient on Trojan Propagation

Like XSS worm propagation, Trojan worm propagation is also affected by the highly clustered structure of OSNs. We examine this effect using the OSN and its ERG listed in Table 1 and assuming a malware execution probability $p = 0.9$. The result in Figure 12 show that the malware propagates more slowly in the OSN than in

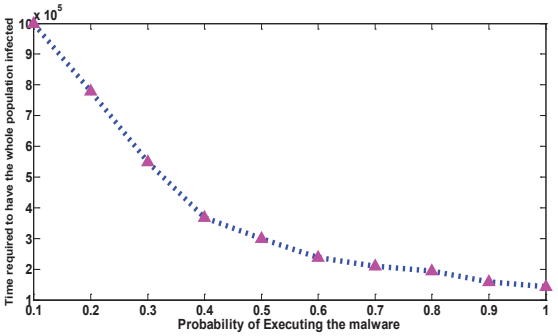


Fig. 11: Propagation speed increases exponentially with probability p [10].

the ERG network. This is consistent with the observation from the XSS malware experiment presented in Section 4.1.2.

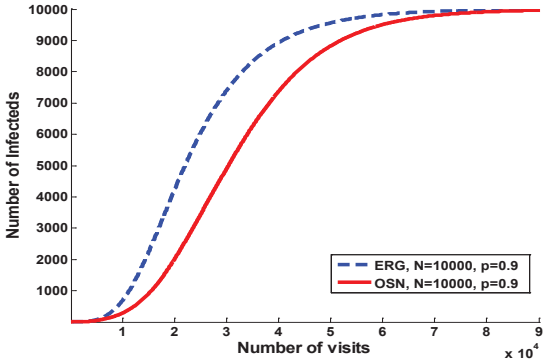


Fig. 12: Trojan worm propagation in the OSN vs. its ERG with $p = 0.9$

4.2.3 Trojan vs. XSS Propagation

In this experiment, we compare the propagation speed of Trojan and XSS worms under the same network conditions and parameters. We assume that people visit other users (profiles) following a Poisson process with an average of k times per minute. Thus the interval between visits follows an exponential distribution with an average of $\frac{1}{k}$. For the XSS malware, we assume a visiting-friends probability $q = 0.9$. For the Trojan (Koobface) worm, we consider two malware executing probabilities of $p = 0.5$ and $p = 0.7$. Given $k = 10$, Figure 13 shows the results of propagation speeds of Trojan and XSS worms in the 10000-node OSN defined in Table 1. The

results demonstrate that the propagation speed of Trojan worms is faster than that of XSS worm in OSNs.

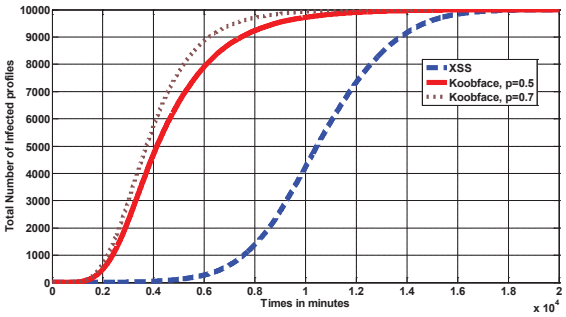


Fig. 13: Propagation speed of Trojan vs. XSS malware

To explain the above result intuitively, we consider a tiny social network having five users as depicted in Figure 14. Initially user A is infected while the others are not. If user A is infected with a Trojan malware and she has already sent the malicious message to all her friends, B, C, D and E, then in the next visit (event) an uninfected user will be selected with a probability of $\frac{4}{5}$. Assuming that this user will open the message and follow the malicious link, this means that one of the uninfected users will get infected with probability of $\frac{4}{5}$.
However, if user A is infected with an XSS worm, then in the next visit, an uninfected user will be selected with a probability of $\frac{4}{5}$ and this user will visit the infected user A with a probability of $\frac{1}{4}$. Therefore, one of the uninfected users will get infected with probability of $\frac{4}{5} \times \frac{1}{4} = \frac{1}{5}$. This explains why the Trojan worm propagated much faster than the XSS worm in the above simulation.

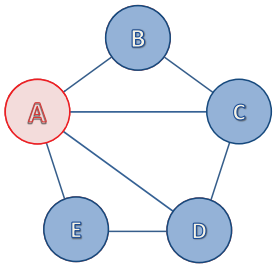


Fig. 14: A tiny social network in which user A is initially infected while the others are not.

In other words, Trojan worms are more proactive than XSS worms. They present themselves to users (in the form of messages) so as to be activated and propagated, while an XSS worm sits on an infected profile waiting for users to select and visit the profile.

4.2.4 Effects of the Initial Number of Infected Profiles

Another important parameter that should be considered in the propagation speed of Trojan and XSS worms is the initial number of infected profiles (users) i_0. Using the OSN graph listed in Table 1, we varied the initial number of infected profiles from 50 to 500, and measured the number of visits (events) required to get 90% of the population (i.e., 9000 users) infected. The obtained results were then normalized to the maximum number of visits measured for each type of worm. Figure 15 shows the results of this experiment. We can see that the effect of increasing the initial number of infected profiles for a Trojan worm such as Koobface is not noticeable. However, increasing the initial number of infected profiles for XSS leads to faster propagation.

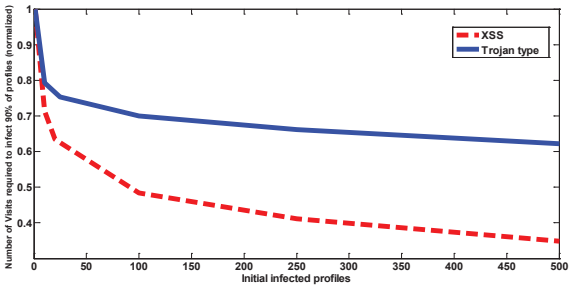


Fig. 15: XSS versus Trojan worm propagation for different values of the number of initial infected profiles

In Trojan malware propagation, assume that each person follows the malicious link and executes the malware code with a probability of 1. Suppose that there are initially n infected profiles. Thus, on average, there are $n \times \overline{deg(n)}$ potential targets for infection, where $\overline{deg(n)}$ is the average degree of the n infected nodes. Therefore by choosing a large value for the initial number of profiles infected by a Trojan malware i_0, it is possible to make almost all members become potential targets for infection. Thus if we keep increasing i_0 after that, the increase does not have significant effects on the propagation speed.

XSS malware, in contrast, requires a user to visit an infected profile to get infected. Therefore, if we increase the initial number of infected profiles, we practically speed up the propagation of the malware in the network.

In Section 5.1, we will explain why the initial number of infected profiles affects the speed of propagation logarithmically using the susceptible infected (SI) model.

5 Modeling Malware Propagation in Online Social Networks

OSN worms, like other computer worms, behave in a similar manner to biological viruses in terms of infectious disease propagation. Therefore, mathematical analyses on propagation behaviors of biological viruses can be adapted to studies of computer worms [31, 32].

In the area of epidemiology, infectious disease propagation can be modeled using either stochastic or deterministic models [1]. Stochastic models are suitable for a small-scale population, while deterministic models can be used for a large-scale population. Deterministic models should thus be used for modeling OSN worm propagation because of large sizes of OSNs. (As of December 2011, Facebook has approximately 800 million registered users around the world.)

One of the most popular differential equation models for biological worm propagation is the susceptible infected (SI) model. This model has used in several computer worm propagation models such as those in [9, 31, 22]. In this section, we discuss the SI model and existing models proposed for OSN worms.

5.1 The SI Model

The SI model is defined as follows:

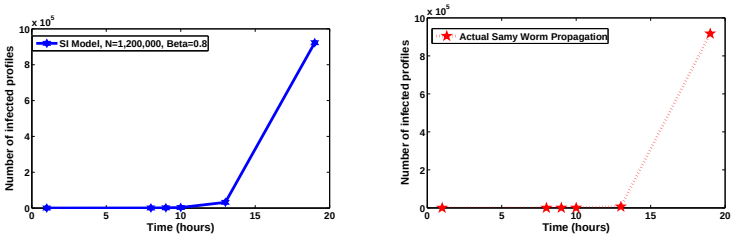
$$\frac{dI(t)}{dt} = \frac{\eta}{\Omega} I(t)[N - I(t)] \tag{2}$$

In this model, N is total number of people in the population; $I(t)$ is the number of infected hosts at time t; η is the worm infectious activity rate; and Ω is the number of possible targets that can be reached by the worm. All hosts are assumed to be either vulnerable (susceptible) or infected according to the SI model. In the field of epidemiology, susceptible hosts are defined as those vulnerable to infection by the virus. Infectious hosts are those that have been infected and can infect others. A host is considered infected at time t if it had been infected before time t. Assuming that η is not a time variant variable and the initial condition is $I(t) = i_0$, the solution to Eq. (2) is as follows:

$$I(t) = \frac{i_0 N}{i_0 + (N - i_0)e^{-\eta \frac{N}{\Omega} t}} \tag{3}$$

The SI model has been used to model XSS worm propagation in OSNs by Faghani and Saidi [9]. Figure 16(a) shows the numerical results of the SI model from Eq. (3). Figure 16(b) shows the propagation trend of a real XSS worm, Samy,

which attacked the MySpace network in 2005. These two graphs demonstrate that XSS worm propagation can be modeled using the SI model [9].



(a) SI model, numerical result (b) Actual propagation of Samy worm

Fig. 16: Samy Worm

The simulation results in Section 4.2.4 show that the initial number of infected profiles i_0 affects the speed of propagation logarithmically. We can explain those results using the SI model, as follows. Given Eq. (3), to infect k per cent of the population, or kN people, we need:

$$I(t) = \frac{i_0 N}{i_0 + (N - i_0)e^{-\eta \frac{N}{\Omega} t}} \geq kN \tag{4}$$

$$t \geq ln\left(\frac{N - i_0}{i_0 \frac{1-k}{k}}\right) \times \frac{\Omega}{\eta N} \tag{5}$$

Inequality (5) suggests that the time required to infect k per cent of the population is logarithmically proportional to the initial number of infected people i_0. The simulation results presented in Section 4.2.4 are consistent with this suggestion: increasing the initial number of infections will decrease the time required to infect 90% of the OSN population logarithmically.

5.2 Modeling XSS worms

Faghani and Saidi propose the following model for characterizing XS worm propagation in OSNs [9]:

$$\frac{dI(t)}{dt} = \beta(q)\frac{I(t)}{N}(N - I(t))^{K(q)} \tag{6}$$

In Eq. (6), $\beta(q)$ is the rate of visiting friends in an OSN, which is a function of the visiting-friends probability q. $K(q)$ represents the sensitivity of the susceptible population to q. In general, $K(q)$ is proportional to q [9]. As q increases, members

will visit their friends more often than strangers. If the number of infected friends is small, a large value of q will delay the propagation of worm in their community. Therefore, uninfected users are also sensitive to (affected by) the value of q, which $K(q)$ takes into account.

The model represented by Eq. (6) is based on the following two facts.

- First, when the visiting-friends probability q increases, the infection is more contained among friends. Less strangers would be affected by infected users. Therefore, the delay for the infection to reach other parts of the network will be longer. In other words, the rate of infection is inversely proportional to probability q.
- Second, after most of the users are being infected, the infection rate will slow down since the total number of susceptible hosts has also been decreased.

5.3 Modeling Trojan and Clickjacking Worms

Let $P(k)$ be the probability that a node in the network graph has degree k. The average degree of the network is thus $E[k] = \sum_k kP(k)$. Suppose that the fraction of infected users having degree k is $i_k(t)$. Let λ be the infection rate, which is the probability of getting infected by an infectious neighbor in a time unit. The differential equation model characterizing the infection rate of a node with degree k is as follows [18]:

$$\frac{di_k(t)}{dt} = \lambda k[1 - i_k(t)]\Theta(t) \tag{7}$$

$$\Theta(t) = \frac{\sum_n nP(n)i_n(t)}{\sum_n nP(n)} = \frac{\sum_n nP(n)i_n(t)}{E[k]} \tag{8}$$

The factor $\Theta(t)$ is the probability that a user (node) is connected by an edge to an infected user (friend) in the OSN graph. To compute $\Theta(t)$, we rely on the fact that the probability of a user having a friend with degree k is $kP(k)$ [18].

The above model is later improved in another model proposed by Boguna et al. [4], which considers the fact that the originator of an infection will not be infected again (by its children in the spanning tree of the network graph). The revised model is as follows:

$$\frac{di_k(t)}{dt} = \lambda k[1 - i_k(t)]\Theta(t) \tag{9}$$

$$\Theta(t) = \frac{\sum_n (n-1)P(n)i_n(t)}{\sum_n nP(n)} = \frac{\sum_n nP(n)i_n(t)}{E[k]} \tag{10}$$

The total number of infected nodes $I(t)$ would be:

$$I(t) = \sum_k i_k(t)P(k)N \tag{11}$$

In a recent research by Faghani et al. [10], the authors suggest an adjustment to Eq. (9) that takes into account the effects of the clustering coefficient and user behaviors as follows :

$$\frac{di_k(t)}{dt} = \lambda k[1 - i_k(t)]\Theta(t)f(c)g(p) \tag{12}$$

Functions $f(c)$ and $g(p)$ reflect the effects of the clustering coefficient and user behaviors, respectively, where c is the clustering coefficient and p is the probability that a user will click on a malicious web link. The authors leave the solutions to $f(c)$ and $g(p)$ to future work.

6 OSN Malware Countermeasures

Several malware detection and containment mechanisms for OSNs have been proposed recently. Nguyen et al. [19] propose a centralized patch distribution algorithm which monitors the number of infected users. When the fraction of infected users reaches a pre-determined threshold, the detection system raises the alarm and sends out treatment patches to influential users. Influential users of a community are those having large numbers of relationships (connections) with other communities. They are thus the best candidates to distribute the treatment patches efficiently throughout the whole network. After receiving treatment patches, a user will apply them to eliminate the worm and forward them to his/her friends.

Xu et al. [28] suggest a scheme in which by monitoring a small fraction of users, the entire network can be under surveillance. An early detection will allow for effective worm containment and elimination measures.

Stein et al. describe the Facebook immune system in [23]. Their immune system performs real-time checks on every incoming and outgoing query to network. To defend against malware, their classifier identifies infected users when they send many messages flagged by the classifier or other users.

Yan et. al. [29] suggest three different approaches for malware detection in OSNs. In the first approach, nodes with the highest degrees are selected and monitored for unusual messages or activities. In the second approach, the most active nodes (in terms of number of messages read and posted) in the network are monitored. In the third approach, the OSN is divided into small islands and every message exchanged between these islands is inspected.

The models and simulation results presented in Sections 4 and 5 suggest that one of the most resource-efficient ways to defend against malware in OSNs is to detect it early within communities. The reason is that a malware will circulate among members of a community for a while before it gets a chance to move to another community, due to the high clustering property of OSNs. This approach of malware detection and containment is still an open issue for future research.

Another optimized defending mechanism is to take into account portions of the network graph which are built based on active relationships among OSN users, since

active users are more likely to visit friends and execute malware code. Wilson et al. [27] show that the graph of interactions among active users is different from the graph of relationships (friendships) in an OSN.

7 Related Work

There exists research in the field of epidemiology that models the behavior of contagious diseases [18, 4, 21, 24, 17, 12]. These models can be and have been used to study the propagation of malware in online social networks [9].

Malware propagation in specific types of computer networks such as e-mail, instant messaging and mobile networks has also been well studied [31, 16, 5, 11].

Among the first works studying malware propagation in OSNs are those by Faghani and Saidi [9, 8], Yan et al. [29], and Xu et al. [28]. Faghani and Saidi [9] model the propagation of XSS worms using the SI model, and investigate malware propagation in OSNs using synthesized OSNs and based on user activities. Yan et al. [29] use realistic network graphs in addition to realistic user activities to confirm that user activities play an important role in malware propagation in OSNs. Xu et al. [28] propose a correlation-based scheme to slow down worm propagation in OSNs. Their scheme was designed and evaluated using a real OSN graph, the Flickr network.

There exists also research on human activities in OSNs. Benevenuto et al. [3] analyze user activities on four popular OSNs (Orkut, Hi5, Myspace and LinkedIn) and provided useful information on how users behave in OSNs.

8 Chapter Summary

We discuss the characteristics of malware propagation in online social networks using analytical models and simulation results. In general, the propagation of XSS worms depends largely on users' behaviors: If OSN users visit mostly their friends rather than strangers, the worms will propagate more slowly. The highly clustered feature of social networks also helps to slow down the propagation. Increasing the initial number of infected profiles in the early stages of XSS worm propagation leads to an impressively faster propagation. Trojan worms propagate faster than XSS worms in social networks because of their inherent aggressive propagation method. Increasing the initial number of infected profiles in the early stages of Trojan worm propagation does not have considerable effects on the propagation speed. We also identify open issues for future research. First, current analytical models do not consider the network graph structure in the propagation of malware in OSNs. Second, we should exploit the high clustering structure of OSNs to detect the propagation of XSS worms early within a community, e.g., using a honeypot detection mechanism.

Acknowledgements We would like to thank Hossein Saidi, Ashraf Matrawy, Chung-Horng Lung and our anonymous reviewers for their helpful comments and discussions.

References

1. Andersson, H., Britton, T.: Stochastic Epidemic Models and Their Statistical Analysis. Springer-Verlag, New York (2000)
2. Barabsi, A., Albert, R.: Emergence of scaling in random networks, Science, 286(5439) , 509-512. (1999)
3. Benevenuto, F., Rodrigues, T., Cha, M., Almeida, V.: Characterizing user behavior in online social networks. In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. pp. 49-62. ACM, New York (2009)
4. Boguna, M., Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in complex networks with degree correlations. In: Lecture Notes in Physics: Statistical Mechanics of Complex Networks, 625, pp. 127-147. (2003)
5. Cheng, S.-M., Ao, W. C., Chen, P.-Y., Chen, K.-C.: On modeling malware propagation in generalized social networks. In: IEEE Communication Letters, 15(1), 25-27. (2011)
6. Davidsen, J., Ebel, H., Bornholdt, S.: Emergence of a Small World from Local Interactions: Modeling Acquaintance Networks. In: Physical Review Letters, 88(12), 128701-1:4. (2002)
7. Dekker, A.H.: Realistic Social Networks for Simulation using Network Rewiring. In: Proceeding of International Congress on Modeling and Simulation, pp. 677-683 (2008)
8. Faghani, M. R., Saidi, H.: Malware propagation in online social networks. In: proceeding of the 4th IEEE International malicious and unwanted programs, pp. 8-14, IEEE Press, New York, (2009)
9. Faghani, M. R., Saidi, H.: Social networks' XSS worms. In: proceeding of the 12th IEEE International conference on computational science and engineering, pp.1137-1141, IEEE Press, New York, (2009)
10. Faghani, M. R., Matrawy A., Lung C.: A study of malware propagation in Online Social Networks. In: Proceeding of 5th IEEE IFIP International conference on New Technologies, Mobility and Security, IEEE Press, New York, (2012)
11. Faghani, M. R., Nguyen U. T.: SoCellBot: A new botnet design to infect smartphones via social networks. In: Proceeding of 25th IEEE Canadian Conference on Electrical and Computer Engineering, IEEE Press, New York, (2012)
12. Griffin, C., Brooks, R.: A note on the spread of worms in scale-free networks. In: IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 36(1), 198202 (2006)
13. Grossman, J.: Cross-site scripting worms and viruses: the impending threat and the best defense, http://www.whitehatsec.com/downloads/WHXSSThreats.pdf (2006)
14. Holme, P., Beom J.: Growing scale-free networks with tunable clustering. In: Phys. Rev. E, 65, 026107-1:4 (2002)
15. Kaspersky Lab Detects new worm attacking MySpace and Facebook, http://www.kaspersky.com/news?id=207575670 (2008)
16. Mannan, M., Van Oorschot, P. C.: On instant messaging worms, analysis and countermeasures. In: Proceedings of the 2005 ACM workshop on Rapid malcode, pp.2-11, ACM, New York, (2005)
17. Moore, C., Newman, M. E. J.: Epidemics and percolation in small-world networks. In: Physical Review E, 61(5), 56785682 (2000)
18. Moreno, Y., Gomez, J., Pacheco, A. F.: Epidemic incidence in correlated complex networks. In: Phys. Rev. E, 68, 521-529 (2003)
19. Nguyen, N.P., Ying X., Thai, M.T.: A novel method for worm containment on dynamic social networks. In: The Military Communications Conference, pp. 475-478, IEEE Press, New York, (2010)

20. Open Web Application Security Project, OWASP Top 10 Project 2010, http://www.owasp.org
21. Pastor-Satorras R., Vespignani, A.: Epidemic spreading in scale-free networks. In: Phys. Rev. Letters, 86, 3200-3203. (2001)
22. Staniford, S., Paxson, V., Weaver, N.: How to 0wn the Internet in your spare time. In: Proceedings of 11th USENIX Security Symposium, pp. 149167, USENIX Association, Berkeley, (2002)
23. Stein, T., Chen, E., Mangla, K.: Facebook Immune System. In: Proceeding of Eurosys Social Network Systems SNS, pp. 8:1-8:8, ACM, New York, (2011)
24. Telo, Nunes, A.: Epidemics in small world networks. In: The European Physical Journal B - Condensed Matter and Complex Systems, 50(1), 205208 (2006)
25. Viger, F., Latapy, F.: Efficient and Simple Generation of Random Simple Connected Graphs with Prescribed Degree Sequence. In: Proceeding of the 11th Conference of Computing and Combinatorics, pp. 440-449, Springer-Verlag, Berlin (2005)
26. Watts, D.J.: Networks, Dynamics, and the Small-World Phenomenon. In: American Journal of Sociology, 105(2), 493527 (1999)
27. Wilson, C., Boe, B., Sala, A., Puttaswamy, K. P.N., Zhao, B. Y: User interactions in social networks and their implications. In: Proceedings of the 4th ACM European conference on Computer systems, pp. 205-218, ACM, New York, (2009)
28. Xu, W., Zhang, F., Zhu, S: Toward worm detection in online social networks. In: Proceedings of the 25th Annual Computer Security Applications Conference, pp. 11-20, ACM, New York (2010)
29. Yan, G., Chen, G., Eidenbenz, S., Li, N.: Malware Propagation in Online Social Networks: Nature, Dynamics, and Defense Implications, In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, pp. 196206, ACM, New York (2011)
30. Yong-Yeol, A., Seungyeop, Kaok, H., Moon, S., Jeong, H.: Analysis of topological characteristics of huge online social networking services. In: Proceeding of the 16th International conference on World Wide Web, pp. 835844, ACM, New York (2007)
31. Zou, C. C., Towsley, D., Gong, W.: Modeling and Simulation Study of the Propagation and Defense of Internet Email Worm. In: IEEE Transactions on Dependable and Secure Computing, 4(2), 105118 (2007)
32. Zou, C., Towsley, D., Gong, W.: Code red worm propagation modeling and analysis. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 138-147, ACM, New York (2002)