Malware Propagation in Online Social Networks

Mohammad Reza Faghani
APA Professional Center[*]
Department of Electrical and Computer Engineering
Isfahan University of Technology
Isfahan, Iran
faghani@ieee.org

Hossein Saidi
Department of Electrical and Computer Engineering
Isfahan University of Technology
Isfahan, Iran
hsaidi@cc.iut.ac.ir

***Abstract*— Online Social Networks are communities of people who share common interests. These types of networks are used by millions of people around the world. The massive adoption of this service among users has made it a popular mean for malicious activities. The aim of this paper is to identify the parameters which are related to malware propagation in online social networks. To do this, we first construct a sample network based on the features of online social networks and then we examine the effect of parameters that could affect the speed of malware propagation.**

Keywords- Social networks; XSS worms; Koobface.

I. INTRODUCTION

In recent years, Internet users have experienced different types of worms and hence lots of research is done in modeling and simulation of spreading worms [2-7]. Although the population of potential victims of Web based worms is much larger than other types of worms, only a few works are related to Web based worms [8-16]. In addition to the popularity of World Wide Web, one reason for having larger population of victims is that Web based worms are not banned through Web proxies and NAT processes. Due to this ease of access, the malware writers aimed to the Web users as a potential target.

Online social network sites are one of the Web 2.0 services that are among the most visited sites globally. The main type of Social networks means to connect friends with their self description pages or profiles. These types of networks are heavily connected among friends. Therefore malware writers may exploit the trust among the social network users to propagate their malicious codes.

The first active worm that hit online social networks was MySpace Samy worm in 2005 [9-12]. Starting with a single person, after 20 hours, the infectious profiles exceeded one million as it is shown in Fig. 1. After two days MySpace was forced to shut down its site to fix the problem [17,18]. Samy exploited a security flaw in the MySpace Web application program which was a cross site scripting vulnerability. Cross site scripting or XSS, is a security flaw that most of the Web applications are vulnerable to it [10]. While XSS is a common vulnerability in Web applications, its threat becomes more noticeable due to the combination of HTML and AJAX technology. AJAX allows browsers to issue HTTP requests on behalf of the user. Thus there is no need for the attacker to deceive the victim to click on a special crafted link. This technique provides facility for malware writers to write active worms.

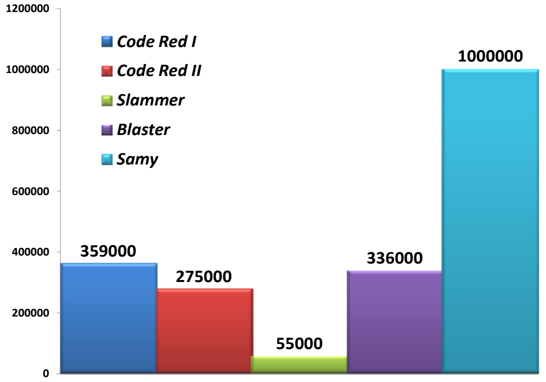

Figure 1. Total number of infections after 20 hours [11]

As Fig. 1 shows, the Samy worm propagation was considerably higher than other actively propagated worms in the same time span of 20 hours [11]. One reason is that the population of potential victims in Web based worms is much larger than other type of worms, thus the total number of infected users is potentially higher than other types of worms.

In addition to MySpace, other social networks such as Orkut, Gaia, hi5 and Twitter have been under attack by XSS active worms [19].

Although online social network providers have tried to fortify their applications to prevent cross site scripting, a new type of worms is now heading into online social networks such as Facebook and MySpace. Koobface which was originally propagated in 2008 has hit MySpace and Facebook users [20]. Koobface spreads by delivering spam messages to people who are 'friends' of someone on the social network whose computer has already been infected. The messages contain deceptive subject headers that are created using social engineering techniques.

The message directs the recipients to a third-party website unaffiliated with Facebook where they are prompted to download what is claimed to be an update of the Flash player. If they download and execute the file, they will infect their computer with Koobface.

[*] APA Professional Center is an academic Computer Security Incident Response Team (CSIRT)

In this paper we are going to analyze and simulate active worm propagation in online social networks. The rest of this paper is organized as follows. Section II gives a brief description of cross site scripting vulnerability. In section III, we give a brief overview of the characteristics of social networks. In section IV, we discuss the simulation experiments and section V concludes the paper with some discussions.

II. Cross Site Scripting Worms

Cross Site Scripting (XSS) is one of the most common Web application layer attacks that hackers use to hide their malicious codes into the applications [1]. There are two distinct types of XSS attacks, the persistent attack and the non-persistent attack. The persistent also known as stored attack is the type in which the injected code is permanently stored on the target servers as an HTML text, such as in a database, in a comment field, messages posted on forums, etc. The visitor then accesses the malicious code from the server when it retrieves the stored information via the browser. The non-persistent also known as reflective attack is the most common type of XSS attacks. In this type the injected code is sent back to the visitor off the server, such as in an error message, search results, or any other response that includes some or all of the input sent to the server as part of the request.

Researches show that, about 80% of the Web applications are vulnerable to cross site scripting attacks. This is due to the fact that the users are permitted to enter tags in the input forms. This increases the threat to the Web application by allowing the hackers to inject malicious codes such as worms in the Web applications through the permitted tags.

There are several factors that cause XSS prevalence in Web applications. First, the system requirements for attack are very simple: XSS exploits Web applications that display input before inspecting them. Second, most Web application programming languages have no filtering option for sending the untrusted inputs to the client. Typically, copying the untrusted input directly and without inspecting the output page is the simplest way of displaying such data.

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. A Cross Site Scripting worm is a combination of one of the oldest security issues, i.e. virus with the new age vulnerability in the Web application, i.e. Cross site scripting. Web application worm is a variant of stored XSS attack. This type of worms has the ability to copy itself in other part of Web pages using the existing XSS vulnerability of the Web applications.

Since online social networks serve lots of users around the world, they are perfect targets for Web based worms. Cross site scripting worms, infect members of social networks in two steps. Initially the worm creator adds a malicious payload to his profile. Subsequently any person who visits the infected profile gets infected and the malicious payload would be added to the visiting person's profile making it a source of infection too.

XSS worms are platform independent due to the fact that they exist in Web application vulnerabilities. It should be pointed out here that this type of malware has its own limitations. Conventional viruses could have access to the file system and memory or even arbitrarily communicate with others. But the XSS worms must run in the Web browser environment.

Whilst this limitation exists, XSS worms are still able to do malicious activities. For example, the payload could deliver a DDoS attack, display spam or contain browser exploits.

Lots of efforts is done on prevention of cross site scripting [14,18] but there is not a unique solution to prevent malwares from distributing among people.

III. Characteristics of Social Networks

Since we want to analyze and simulate the propagation of XSS worms in online social networks, we need to have a better understanding of the topology characteristics of social networks. If we assume each person is a single node and acquaintances make a link between themselves, we may have the equivalent graph of the social network.

Real-world social networks are highly clustered small world networks with a degree distribution often following a power law distribution. The characteristics of the social networks are studied in [21-23]:

1- A low average network distance, approximately equal to $\frac{\log n}{\log d}$ where n is the number of people, and d is the average degree of the equivalent graph.

2- Social networks typically show a high clustering, or local transitivity which means if person A knows B and C, then B and C are likely to know each other. Indeed, the clustering coefficient of vertex v is $C(v)$ which is the number of acquaintance triangles of vertex v, divided by $k(k$-1)/2, the number of all possible triangles of vertex v, where k is the degree of v. The Clustering coefficient of the graph is the average of clustering coefficients of all of its vertices. In real social networks, the clustering coefficient is roughly about 0.1 to 0.7.

3- An approximately power-law distribution of node degrees. The node degree of a power law topology is a right skewed distribution with the power law CCDF of $F(k) \propto k^{-\alpha}$, which is linear on logarithmic scale. Power law distribution states that the probability for node v to have a degree of k is proportional to $p(k) \propto k^{-\alpha}$ where α is the power law exponent [28].

There are few algorithms that generate a social network graph with the above characteristics [22-25]. We used [23] which has some advantages in describing network growth in time.

A. Generating a social network graph

In this section we will generate a sample social network using the algorithm described in [23]. This algorithm generates a graph with the characteristics of social networks with the power law exponent of $\alpha = 3$. We used the following parameters to generate the graph: n=10000, m=m_0=3 and m_t=1.8. The parameters of the generated graph are shown in Table I.

TABLE I. GRAPH CHARACTERISTICS

Graph Parameter	Value
Number of Vertices	10000
Number of Edges	29990
Clustering Coefficent	0.1409392
Average Shortest Path	5.133096
Maximum Degree	190
Longest Path (Diameter)	10
Degrees Average (d)	5.998
$\frac{\log n}{\log d}$	5.1413

The degree distribution of the graph is shown in fig. 2. As it is shown, the degree distribution of the graph follows a right skewed power law distribution.

Thus, the generated graph satisfies all the required characteristics. Because the average shortest path of the graph is less than $\frac{\log n}{\log d}$. The Clustering coefficient is moderate and the degree of vertices follows a power law distribution.

Since we want to simulate the effect of topology parameters such as clustering coefficient on worm propagation, we need a similar random graph. To do so, we use the algorithm in [26] that generates a random graph with the given degree sequence of vertices. This random graph has the same degree distribution with a different value of clustering coefficient.

Table II shows the parameters of the equivalent random graph.

TABLE II. GRAPH CHARACTERISTICS

Graph Parameter	Value
Number of Vertices	10000
Number of Edges	29990
Clustering Coefficent	0.003581474
Average Shortest Path	4.407071
Maximum Degree	190
Longest Path (Diameter)	8
Degrees Average (d)	5.998
$\frac{\log n}{\log d}$	5.1413

As it is shown in table II, the clustering coefficient of the social network graph is 40 times greater than that of the random graph. This reflects the characteristics of the highly clustered social networks. The average shortest path and clustering coefficient of the random equivalent graph is less than the social network graph. This is due to the small world phenomenon described in [27].

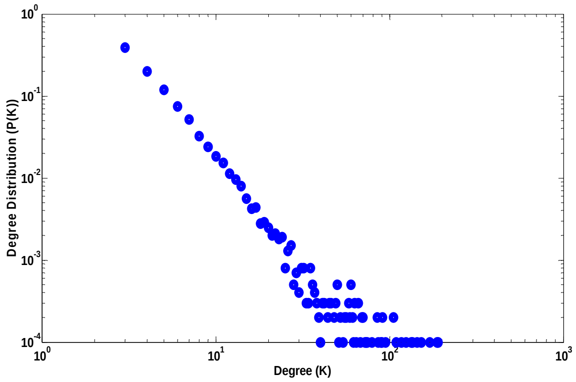

Figure 2. Degree distribution of the generated graph

IV. SIMULATION OF ACTIVE WORMS PROPAGATION

In this section we simulate the propagation of XSS and Koobface-like worms in online social networks. The simulation is of discrete event type consisting of discrete time slots. Simulation results have been averaged over 100 trials.

A. *Description of the simulation model*

As mentioned in section II, in the XSS worm propagation trend, a vulnerable user has to visit an infectious profile to get infected. User's vulnerability is determined by whether the user's browser is able to execute the malicious script or not, because some of the users may have disabled the script execution by using some add-ons like NoScript for Firefox.

To simulate the trend of XSS worm propagation, we assume that a single visit on the social network site is an event. Therefore if the visitor user is vulnerable and visits an infected profile, then the visitor user gets infected. The visited person (the one who the visitor is going to visit) has two states: one is that the visited profile is one of the visitor's friends and the other one is that the visited profile belongs to none of the visitor's friends. Here we assume user i visits his/her friends group with probability q_i.

If the user is infected with a koobface-like worm, the worm sends out a spam email to the infected user's friends. This message contains the links that may direct the users to a counterfeit video-hosting site like Youtube, upon clicking on the link. Then the fake video prompts the user to install what appears to be the Flash player update. The installed software turns out to be a malware which propagates itself by sending similar messages to friends of the user whose machine has been newly infected.

To simulate the trend of koobface-like worm propagation, we also assume that a single visit on the social network site is an event. Therefore if the visitor user is user i and has already received the spam email and is going to execute it with probability p_i, then with probability p_i he gets infected.

We simulate the worm propagation on social network graph of section IV and also we assume all people in the network are vulnerable.

B. *Simulation Expertiment of XSS Worm*

First we simulate XSS worm propagation in the social network. To do this, in each event, one person is chosen with an equal probability. Then the chosen person visits one of his/her friends with probability of q and the others with probability of 1-q. Figure 3 shows the propagation trend for different values of q. Here we assumed that all people visit their friends with the same probability of q.

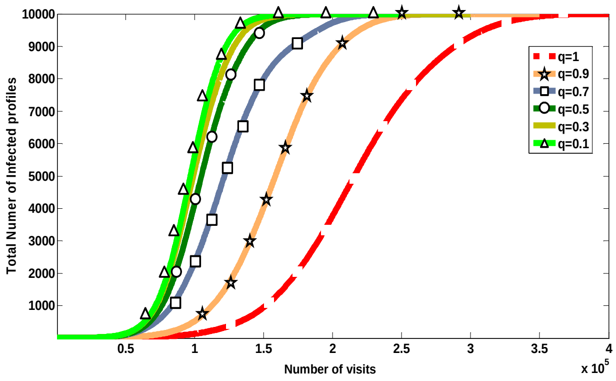

Figure 3. XSS worm propagation trend for different values of q

As it is shown in figure 3, if people visit their friends more often, the propagation would be slower. This is due to the fact that when people visit their friends more than other people, the infected population would be contained among those friends and the infection would reach other parts of the network with more delay.

Next we want to see the effect of clustering coefficient on the XSS worm propagation in social networks. To do this we simulate the propagation of the XSS worm in both social network and the equivalent random graph with two different probabilities: q=0.1 and q=0.9. The results are shown in Figure 4 and Figure 5.

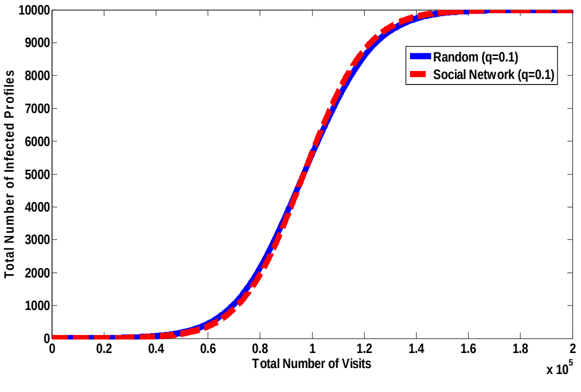

Figure 4. Random versus Social network for q=0.1

Although both graphs have the same degree distribution, they reflect different worm propagation behaviors for higher visiting friend probability.

One reason is that the graph topology is meaningful only for higher probabilities of visiting friends. Low visiting friend probability means that people randomly select someone to visit, independent of the graph topology which helps the worm to propagate faster. One of the differences between the social network graph and its equivalent random graph is that the social network graph is highly clustered. This feature slows down the worm propagation in higher visiting friend probabilities.

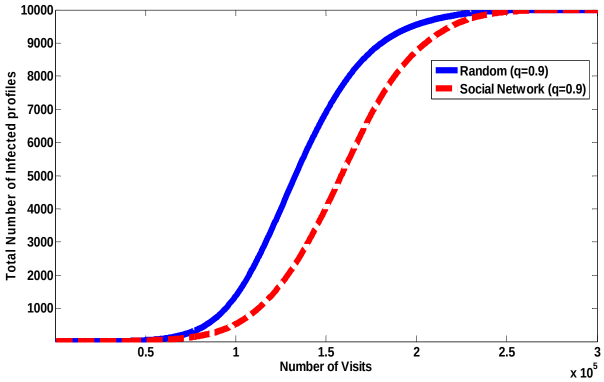

Figure 5. Random versus Social network for q=0.9

In a highly clustered graph, if the probability of visiting friends is high, we can assume two cluster cases. One case is when the cluster has at least one infected member. In this case, members of the cluster get infected soon and this infection is contained in their group and thus causes a delay in propagation to other parts of the network. The other state is that no one is infected in the cluster. If any member from this cluster is selected to make a visit, he may have a visit to other non-infected members with high probability and thus no infection occurs. Thus the highly clustered topology helps slowing down the propagation.

Another experiment is the effect of the number of initial infected profiles on XSS worm propagation. In this part we assumed that we have 1,10,100,1000 and 5000 initial infected profiles or iip in each simulation with q=0.9. The result is shown in figure 6. As it is shown in Figure 6, increasing the number of initial infected profiles in an XSS worm propagation, dramatically makes the propagation faster.

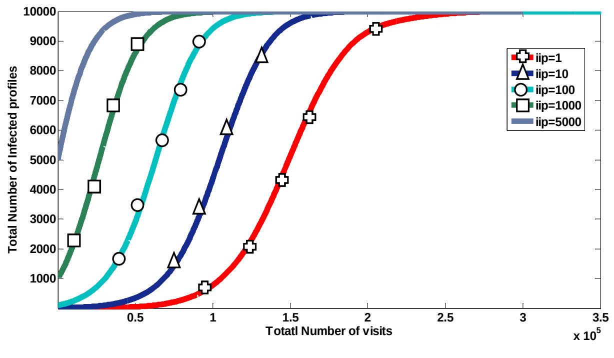

Figure 6. Different values of iip for q=0.9

To have a better view, we calculated the number of visits that the infection reaches 90% of the members for each case. Figure 7 shows the time required to reach 90% of the infection with respect to the iip value.

Note that the X axis is normalized over its maximum value and the Y axis is normalized over the total number of profiles.

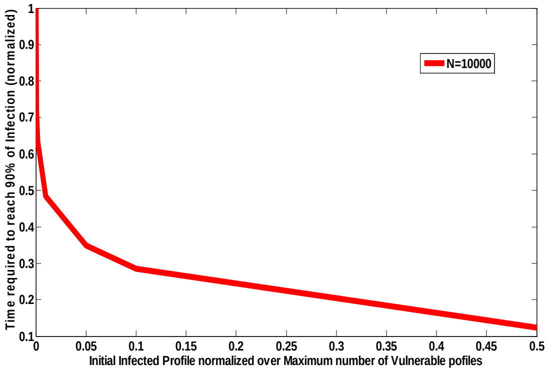

Figure 7. Effect of iip on propagation speed

Since in the XSS worm propagation, the users only have two states: vulnerable and infected, the infection can be modeled as simple epidemic [2] which is stated with the following equation:

$$i(t) = \frac{i_0 n_s}{i_0 + (n_s - i_0) e^{\frac{-\beta n_s t}{n}}} \tag{1}$$

Where $i(t)$ is the number of infected at time t, i_0 is the number of initial infected persons, n_s is the number of susceptible persons, n is the total number of people and β is the propagation parameter [2].

To calculate the time by which the total number of infected reaches k percent of all people, we have the following equation:

$$\frac{i_0 n_s}{i_0 + (n_s - i_0) e^{\frac{-\beta n_s t}{n}}} \geq k n_s \tag{2}$$

This yields to:

$$t \geq ln\left(\frac{(n_s - i_0)}{i_0\left(\frac{1-k}{k}\right)}\right) \times \frac{n}{\beta n_s} \tag{3}$$

Figure 8 shows the equation (3) with k=0.9 and n=n_s=10000 for an arbitrary infection rate of β=1/n.

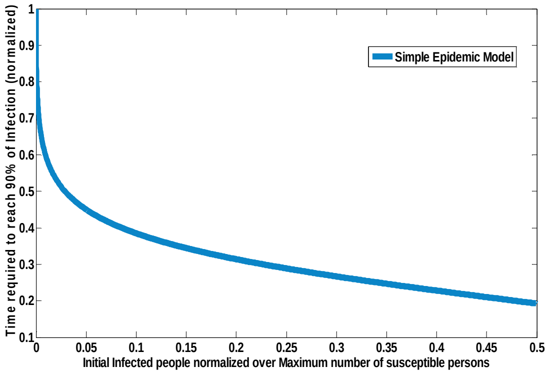

Figure 8. Simple Epidemic model for different iip valie

C. *Simulation experiment of Koobface-like worms*

In the following parts, we simulate the propagation of Koobface-like worms in online social networks and analyze the effect of some important parameters on infection of these types of worms.

As it is stated earlier, in each event, the selected user checks if it has a spam message. With probability of p, the user follows the message and executes the malware. Once the user gets infected, he sends a spam message containing the malicious link to all of his/her friends. Here we assumed that all people execute the malcode with the same probability of p.

Figure 9 shows the propagation trend for different values of p. The word "data" is plural, not singular.

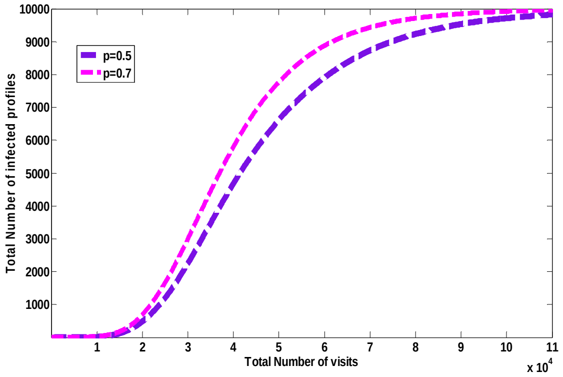

Figure 9. Koobface worm propagation for p=0.5 and p=0.7

It is obvious that for higher infection probability, the worm propagates faster. As figure 9 shows, the propagation speed of Koobface-like worms are much higher than that of XSS worms. To have a better understanding of this issue, suppose a tiny social network like the one shown in figure 10.

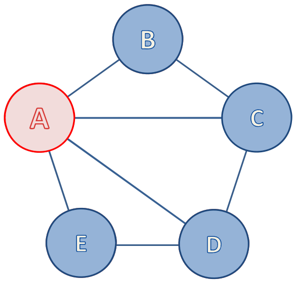

Figure 10. A tiny Social network

If person A is infected with a Koobface worm and has already sent the spam message, and if others are still vulnerable, then in the next visit with the probability of $\frac{4}{5}$, one of the uninfected users would be selected. Assuming he will execute the message with probability of 1, then one of the vulnerable profiles gets infected with probability of $\frac{4}{5}$. But if the person A is infected with an XSS worm, then in the next visit, with the probability of $\frac{4}{5}$ one of the uninfected user might be selected and with the probability of $\frac{1}{4}$ he may visits the infected person A. Therefore, in the next visit, with the probability of $\frac{4}{5} \times \frac{1}{4}$ one of the vulnerable profiles gets infected. Thus the koobface worm propagates much faster than XSS worms.

If we run the simulation for different values of initial infected profiles e.g. 1,10,100,1000 and 5000 and then normalize each axis to its maximum, we will see that the effect of increasing iip for koobface is no more noticeable. Figure 11 shows the result with respect to the XSS worm propagation.

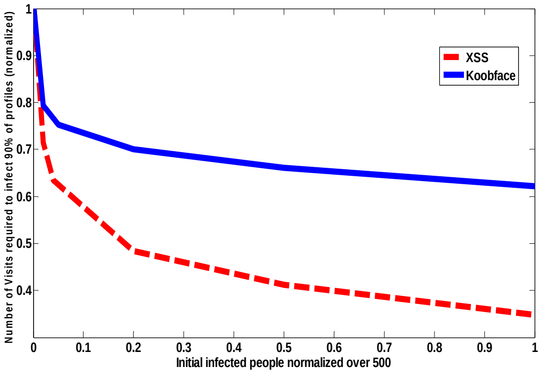

Figure 11. XSS versus Koobface for different values of iip

Although increasing the iip leads to a faster propagation, but the effect of this action on Koobface worm is not as dramatic like the XSS worm. In Koobface worm propagation, assume that each person follows the malicious link and executes the malcode with probability of 1 and also there are initially n infected persons, thus on the average, there are almost $n \times \overline{\deg(n)}$ persons that are potential target for infection, where $\overline{\deg(n)}$ is the average degree of those n infected persons, because these people just need to be selected in each event, therefore by increasing the iip value it is possible to make all members to be a potential target for infection and thus increasing n more may not have any more effects.

Suppose profile visiting of the people follows a Poisson process with the average of k in one minute, thus the interval between each visits follows an exponential distribution with an average of $\frac{1}{k}$. Figure 12 shows the result for k=10 and p=0.5,0.7. As it is shown in figure 12, the Koobface worm propagates faster than XSS worms due to the fact that was described earlier in this section. We have assumed q=0.5 for XSS worm propagation.

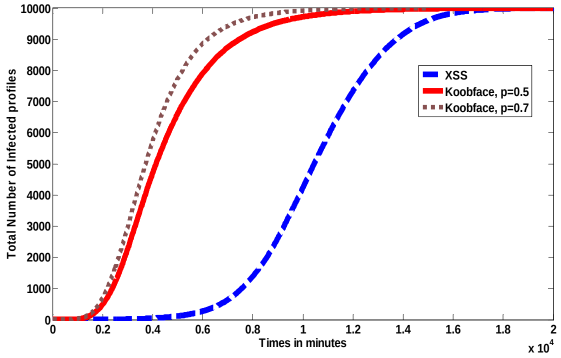

Figure 12. XSS and Koobface time trends

V. USING THE TEMPLATE

In this paper we simulated and analyzed the malware propagation in online social networks. We simulated two types of social network active worms, namely; XSS and Koobface-like worms. We found that the propagation of XSS worms depends on visiting behavior of the social network members. If members mostly visit their friends rather than strangers the worm propagates slower. The highly clustered feature of social networks also helps slowing down the propagation. Increasing the initial infected profiles in the early stages of XSS worm propagation leads to an impressively faster propagation.

Koobface-like worms propagate faster than XSS worms in social networks because of the special way they propagate. Increasing the initial infected profile in early stages of koobface-like worm propagation, does not have considerable result on the propagation speed.

Due to the slow start of the XSS worm propagation; someone may suggest a prevention mechanism like a honeypot to detect the propagation of XSS worms in their early stages.

Since the Koobface-like worms propagate faster in the early stage, users should be warned against unsolicited messages that may make them a zombie.

Of course, the model could be modified to include more details. More complex model is needed to consider other behaviors of the users such as leaving the group. Also, considering the friendship relation among the different clusters and its effect on the infection rate need further investigation.

REFERENCES

[1] Martin Casado and Michael Freedman. Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification. In *Proceedings of the 4th Networked Systems Design and Implementation*, April 2007.

[2] Cliff, C. Z., Weibo, G., Don, T., “Code red worm propagation modeling and analysis”, *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 138-147, 2002.

[3] Rohloff, K., Bacar, T., “Deterministic and stochastic models for the detection of random constant scanning worms”, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, Vol. 18, Issue. 2, No. 8, 2008

[4] Moore, D., Shannon, C., Brown, J., “Code-Red: A case study on the spread and victims of an Internet worm”, *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pp. 273-284, 2002

[5] Nicol, D., “The impact of stochastic variance on worm propagation and detection.”, *Proceedings of the 4th ACM workshop on Recurring malcode*, pp. 57 – 64, 2006

[6] Cliff, C. Z., Weibo G., Don, T., “Email worm modeling and defense”, *Proceedings of 13th International Conference Computer Communications and Networks*, pp. 409–414, 2004.

[7] Wei, Y., Sriram, C., “Peer-to-peer system-based active worm attacks: Modeling, analysis and defense”, *ACM journal Computer Communication*, Vol. 31, No. 17, pp. 4005-4017 , 2008

[8] Niels, P., Dean, M., Panayiotis M, Wang K., “The ghost in the browser analysis of Web-based malware”, *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*. 2008

[9] Shanmugam, J. Ponnavaikko, M. , “XSS Application Worms: New Internet Infestation and Optimized Protective Measures”, *Proceeding of 8th Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing Conference*, pp 1164-1169, 2007

[10] Shanmugam, J. Ponnavaikko, M. , “Cross Site Scripting-Latest developments and solutions: A survey”, Int. J*. Open Problems Compt. Math*., Vol. 1, No. 2, pp 101-121, 2008

[11] Grossman, J., “Cross-site scripting worms and viruses: the impending threat and the best defense”, Available from: http://www.whitehatsec.com/downloads/WHXSSThreats.pdf, 2006.

[12] Athanasopoulos, E., Makridakis, A, Antonatos, S., “Antisocial Networks: Turning a Social Network into a Botnet.”, *Proceeding of Information Security Conference / Workshop (ISC/ISW)* 2008, pp. 146-160, 2008

[13] Lam, V. T., Antonatos, S., Akritidis, P., Anagnostakis, K. G., “Puppetnets: misusing Web browsers as a distributed attack infrastructure”, *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 221 – 234, 2006

[14] Jones, M., Winter, J., “Protecting the Intranet Against JavaScript Malware and Related Attacks”, *Proceeding of ACM Detection of Intrusions and Malware, and Vulnerability Assessment*, Vol. 4579 (2007), pp. 40-59., 2007

[15] Johns, M., “On JavaScript Malware and related threats : WWWWeb page based attacks revisited”, *Springer Journal in Computer Virology*, Vol. 4, No. 3, pp. 161-178, 2008

[16] Trevor, J., “Defeating Script Injection Attacks with Browser Enforced Embedded Policies”, *Proceeding of 16th ACM International world wide Web conference*, pp. 601-610, 2007

[17] White Hat Securtiy, “Cross Site Scripting worms and viruses, The Impending Threat and the Best Defense”, April 2006. http://net-security.org/dl/articles/WHXSSThreats.pdf.

[18] Gary Wassermann, Zhendong Su, “Static detection of cross-site scripting vulnerabilities”, *Proceedings of the 30th international conference on Software engineering*, ICSE '08, Leipzig, Germany. 2008

[19] XSSing- Cross Site Scripting, CSRF Web Security, XSS Worms [page on internet], Available from: http://www.xssing.com/index.php?x=6

[20] McAfee Avert® Labs Threat Library, Available from: http://vil.nai.com/

[21] Yong-Yeol, A., Seungyeop, Kaok, H., Moon, S., Jeong, H., “Analysis of topological characteristics of huge online social networking services”, *Proceeding of the 16th International conference on World Wide Web*, pp. 835–844, 2007.

[22] Dekker, A.H., “Realistic Social Networks for Simulation using Network Rewiring”, *Proceeding of International Congress on Modelling and Simulation*, pp. 677-683, 2008

[23] Holme, P., Beom J., “Growing scale-free networks with tunable clustering”, *Phys. Rev. E 65*, pp. 026107-1:4, 2002

[24] Davidsen, J., Ebel, H., and Bornholdt, S., “Emergence of a Small World from Local Interactions: Modeling Acquaintance Networks,” *Physical Review Letters*, Vol. 88, No. 12, pp. 128701-1:4.

[25] Kawachi, Y., Murata, K., Yoshi, S., Kakazu, Y., "The structural phase transition among fixed cardinal networks.", *Proceeding of 7th Conference on Complex Systems Asia-Pacific*, pp 247-255, 2004.

[26] Viger, F., Latapy, F., “Efficient and Simple Generation of Random Simple Connected Graphs with Prescribed Degree Sequence”, *Lecture notes in computer science*, Vol. 3595, pp. 440-449.

[27] Watts, D.J., “Networks, Dynamics, and the Small-World Phenomenon” *American Journal of Sociology*, Vol. 105, No. 2, pp 493–527, 1999.

[28] [Newman, M., "Power laws, Pareto distributions and Zipf’s law", Contemporary Physics, vol. 46, Issue 5, pp. 323-351, 2005.