SOCELLBOT: A NEW BOTNET DESIGN TO INFECT SMARTPHONES VIA ONLINE SOCIAL NETWORKING

Mohammad Reza Faghani and Uyen Trang Nguyen

Department of Computer Science and Engineering
York University, Toronto, Ontario, Canada, M3J 1P3
{faghani, utn}@cse.yorku.ca

ABSTRACT

Given the popularity of both smartphones and online social networking, it is only a matter of time before attackers exploit both to launch new types of attacks. In this paper, we propose a new cellular botnet named SoCellBot that exploits online social networks (OSNs) to recruit bots and uses OSN messaging systems as communication channels between bots. Our proposed botnet is the first that uses the OSN platform as a means to control cellular bots. The structure and characteristics of OSNs make this botnet harder to detect, more resilient to bot failures and more cost-effective to cellular bots. Our objective is to raise awareness of new mobile botnets that exploit OSNs to recruit bots so that preventive measures can be implemented to deter this kind of attack in the future. We also analyze the behaviors of the proposed botnet via simulation to offer a better understanding of this new type of botnet.

Index Terms— Network security, mobile botnet, cellular botnet, malware, online social networking.

I. INTRODUCTION

In recent years, cellular phones have been revolutionized from basic voice and text phones to IP-enabled smartphones, capable of browsing the Internet and running various applications. One of the most popular activities among mobile device users is online social networking. Online social networks (OSNs) such as Facebook, Twitter and MySpace have attracted hundreds of millions of people worldwide. The ubiquitous nature of smartphone services and the popularity of online social networking can be a lethal combination that spreads malware in a quick and efficient manner to a large number of OSN users, which will in turn infect their own computers and local networks.

A mobile botnet is a group of compromised cellular phones that are controlled by one or more botmasters. OSNs are a preferred medium for botnets to carry out such an attack for the following reasons. First, most cellular network providers offer OSN access to their clients free of charge. This makes OSN messaging systems a cost-effective solution for cellular bots to send and receive commands and control messages. Second, messages exchanged in OSNs are usually encrypted, making it hard for cellular network providers to identify and block botnet messages. Third, the topology of an OSN-based botnet is more resilient to bot failures or unavailability (compared with commonly seen botnets using on short message services (SMS) [20], [28]) thanks to the highly clustered structure of the social network graph.

Given the popularity of both smartphone usage and online social networking, it is a matter of time before attackers exploit both to launch new types of attacks. In this paper, we present the design of a new cellular botnet named SoCellBot that exploits social networks to recruit bots and uses messaging systems of OSNs as communication channels between bots. Our objective is to raise awareness of new mobile botnets that exploit OSNs to recruit bots so that preventive measures can be implemented to deter this kind of attack in the future. We also analyze the behaviors of the proposed botnet via simulation to offer a better understanding of this new type of botnet.

Although there exist several cellular botnet designs in the literature [20], [24], [23], [28], [26], our proposed botnet is the first that uses the OSN platform as a means to control cellular bots. The structure and characteristics of OSNs make this botnet harder to detect, more robust and more cost-effective to cellular bots.

The remainder of this paper is organized as follows. We discuss related work in Section II, and describe the design of SoCelBot botnet in Section III. The simulation model and parameters are presented in Section IV. We analyze the simulation results in Section V. In Section VI, we discuss analytical models related to the proposed botnet and directions for our future work. Section VII concludes the paper.

II. RELATED WORK

Botnets have been actively researched in recent years, especially PC-based botnets [12], [1]. Research on cellular botnets have also appeared recently [20], [24], [23], [28], [26]. Traynor et al. [24] theorize the existence of cellular botnets. They conclude that the rigid hierarchical structure of cellular networks make them more vulnerable than other types of networks to a simple threat such as denial-of-service attacks. They also show that a relatively small number of infected phones can easily shut down the core network. Singh et al. [23] study the feasibility of using Bluetooth as the command and control (C&C) channel of a botnet. Mulliner et al. [20] propose a SMS-HTTP command and control system in which commands created by the botmaster are sent to bots via SMS. The commands are then uploaded to designated websites in an encrypted file. Each bot will download and decrypt the file, and send out the commands to other bots via SMS. Zeng et al. [28] design a SMS-P2P hybrid botnet which uses SMS as the C&C channel, and the peer-to-peer network as the underlying structure. In this botnet, no IP connection is involved. Bots search and obtain commands in a P2P fashion by sending and receiving SMS messages. This approach easily leads to detection since it imposes significant monetary costs on the victims by sending SMS messages to get the commands via the P2P system. Many cellular network providers charge a fee for using SMS. Andbot [26] eliminates the weakness of a single point of failure in HTTP-based C&C schemes by taking advantage of URL fluxes. This makes the botnet more resilient to different types of attacks such as DNS sinkhole and IP black listing.

Our proposed SoCellBot botnet is the first that exploits OSNs to recruit bots. Using OSN messaging systems as the C&C channel makes the botnet more difficult to be detected and more robust against bot failures or unavailability. Unlike SMS-based botnets, our SoCellBot incurs only very small monetary costs, as discussed next.

III. THE PROPOSED SOCELLBOT

The objective of a SoCellBot botnet is to infect as many smartphones as possible with malware. The medium to spread the infection is messaging systems of OSNs, which is more cost-effective to bots than SMS messages. The design of a botnet consists of three major components: propagation mechanism, command and control channel, and botnet topology maintenance.

III-A. Propagation Mechanism

Mobile devices are recruited into a botnet by running malicious software. This can be achieved in two ways: one is to exploit vulnerabilities of the operating systems (OS); the three major mobile phone operating systems, IOS, Android and Symbian, have been shown to be vulnerable to malware attacks [11]. The other method is to use social engineering techniques to trick users into running the software (e.g., clicking an eye-catching web link leading to the malicious content). A SoCellBot botnet exploits both attacking vectors. That is, either the user will follow the malicious web link and execute the malware, or the smartphone OS is vulnerable to a specific attack that will run the malicious code without any user intervention.

To initiate a botnet, an attacker can compromise a part of social graph by infiltrating. Infiltration can be started by a number of fake profiles that will try to get connected to real users. After the first connections, they will try to get connected to friends of those users, and so on. Infiltration has been shown to be effective for starting a botnet in an OSN such as Facebook [4].

III-B. Command and Control Channel

In many countries, users have to pay for sending and receiving SMS messages. Our proposed botnet tries to minimize the use of SMS to avoid being detected by users or cellular network providers. Therefore, each bot will forward the command through an online social network messaging system (OSNMS). (The botmaster can send out the initial commands to a small number of bots through SMS though.) As more and more cellular network providers offer access to OSNs free of charge, forwarding the commands using an OSNMS overcomes the cost challenge existing in current SMS-based botnets [14]. The commands can be disguised to look like normal messages using an algorithm such as the one proposed by Zeng et al. [28].

Sending a message to a random user in Facebook is generally possible. However, some users may deactivate this feature for non-friend users. These users will not take part in the initiation phase, but they can be infected by their infected friends in the future.

III-C. SoCellBot Botnet Topology

The SoCellBot botnet topology is ensured to be connected thanks to the high clustering characteristic of OSNs [8], [13], [27], which refers to the fact that users tend to create tightly knit groups characterized by a relatively high density of ties (friendships). As a result, if some bots become idle or are disabled, there are still some other ways to reach the neighbors of the disconnected bot. A SoCellBot botnet is thus resilient to bot failures and unavailability.

IV. SIMULATION MODEL AND PARAMETERS

In this section, we review the characteristics of online social networks, and describe the network graph model and malware propagation model used in our simulations.

IV-A. OSN Model and Graphs

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 (friendship) between the two respective persons. Our simulations were carried out on synthesized graphs that possess all the characteristics of real-life OSNs. The characteristics of online social networks, which are studied in [8], [13], [27], can be summarized as follows:

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 [21].

There exist a few algorithms that can generate social network graphs with the above characteristics [8], [7], [13]. For the simulations reported in this paper, we used the algorithm proposed by Holme and Beom [13], because it can be fine tuned to generate a social network graph with the required clustering coefficient and power law distribution of node degrees. We used the algorithm by Holme and Beom to generate three OSNs of sizes 5,000 nodes, 10,000 nodes and 15,000 nodes. The parameters and characteristics of these three OSN graphs are listed in Table I. In all the three graphs, the node degrees are power-law distributed with $\alpha = 3$.

We also created three equivalent random graphs (ERG), each corresponding to one of the above three OSN graphs. These random graphs are generated using the above three OSN graphs and the algorithm proposed by Viger and Latapy [17]. Each of these random graph has the same node degree distribution as the equivalent OSN graph. However, the other parameters may be different. For instance, an ERG usually has a lower clustering coefficient and

Parameter	Value	Value	Value
Number of vertices (people)	5,000	10,000	15,000
Number of edges	14991	29991	44991
Average clustering coefficient	0.142	0.157	0.123
Average shortest path length	4.73	5.8	5.12
Network diameter	10	14	10
Maximum node degree	204	182	366
Average node degree d	5.99	5.99	5.99
$\log(s)/\log(d)$	4.76	5.2	5.37

Table I. Three OSN graphs used in our simulations

Parameter	Value	Value	Value
Number of vertices (people)	5,000	10,000	15,000
Number of edges	14991	29991	29991
Average clustering coefficient	0.006	0.003	0.002
Average shortest path length	4.16	4.39	4.5
Network diameter	8	8	8
Maximum node degree	204	182	366
Average node degree d	5.99	5.99	5.99
$\log(s)/\log(d)$	4.76	5.2	5.37

Table II. Equivalent random graphs generated from the above OSN graphs

network diameter than the original OSN graph. The parameters of these three equivalent random graphs are listed in Table II. In all the three equivalent random graphs, the node degrees are power-law distributed with $\alpha = 3$.

We now explain the reason for studying equivalent random graphs in addition to the original OSN graphs. An attacker may be able to obtain the graph of an OSN using a tool such as R [15] or Pajek [2]. He/she may then create ERGs using an algorithm such as the one by Viger and Latapy [17]. As our simulation results will show, an ERG helps a malware to propagate faster than the original OSN graph, but requires more messages to infect the same number of victims. Our goal is to determine whether ERGs help or hinders the malware propagation in order to predict attack strategies.

IV-B. Malware Propagation Model

In the first step of each experiment, a node (user) in the social network graph is chosen randomly as a seed for infiltration. When the user executes the command (e.g., clicks on a web link), the user's smartphone sends out a message to his/her friends (adjacent vertices in the social network graph), directing them to the malicious content. Upon receiving the message, each friend will execute the malware with a probability p. (Some people may be more cautious and do not follow the web link.)

Each command to execute the malware has unique sequence number (SN) and a time-to-live (TTL) field. After receiving a message carrying a command, a node checks the SN to see if it has seen the message before. If the message is new, it decreases the TTL by one, and forwards the message to its one-hop neighbors (adjacent vertices) in the OSN graph. If the message is a duplicate, the node simply discards it. Sequence numbers help to minimize the number of duplicate messages, and thus the total number of messages, sent by the bots in order to avoid detection. The TTL limits the lifetime of a message, again to minimize the number of messages sent by the bots. A good estimate for the TTL is the diameter of the OSN graph, which was also used in our simulations.

V. SIMULATION RESULTS

The simulation was done in MATLAB based on discrete-event simulation. Each data point in the graphs is averaged over 100 runs, each of which started with a different node (user) selected randomly.

We conducted two sets of experiments. In the first set, we measured the total number of infected smartphones T over time. A virtual time unit is defined as the time the malicious command takes to traverse one hop in the OSN to reach all the neighbors of the current sender of the command, which is a newly infected node. We assume that all newly infected nodes forward the command within one virtual time unit. As a result, time can be represented by the number of hops in the OSN graph, where a hop is equivalent to a virtual time unit. In addition to the total number of infected smartphones T over time, we also measured the number of newly infected smartphones N at every virtual time unit t. That is, $T(t+1) = T(t) + N(t+1)$. Metric N shows us the point in time when the propagation achieves its peak performance, i.e., infecting the most number of new phones.

In the second set of experiments, we recorded the total number of messages (carrying the malicious command) M sent by all the infected phones via the OSN over time. Again, time is represented by the number of hops in the OSN graph as explained above. Metric M reflects the amount of network bandwidth and resources consumed by the botnet. Attackers would want to keep M as low as possible to avoid alerting the network administrator to the attack. According to the malware propagation model described in Section IV-B, a node will reject (discard) duplicate messages received from its neighbors. We thus also recorded the total number of non-duplicate messages accepted by nodes in the OSN over time. Following are the results we obtained from these two sets of experiments.

V-A. The First Set: Number of Infected Smartphones

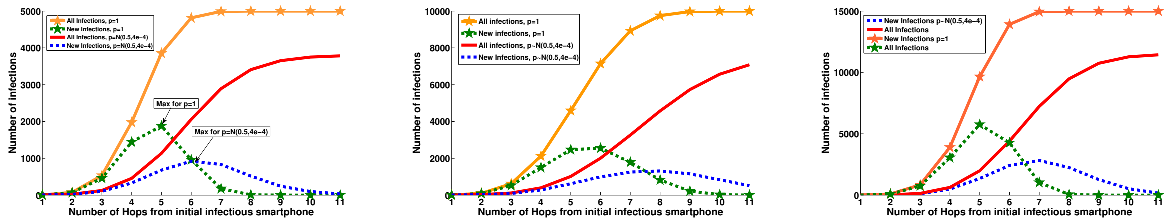
This set of experiments consists of three scenarios.

V-A1. Scenario 1

The graphs in Fig. 1 show the total number of infected smartphones and the number of new infected smartphones over time using the three OSNs defined in Table I for two different values of p, the probability that a user will execute the malicious command. We considered a constant value of $p = 1$ (i.e., a user always executes the malicious command), and a normal distribution of p with mean $\mu = 0.5$ and variance $\sigma^2 = 0.02^2$.

In all three OSNs, as p increases from the mean value 0.5 to 1, more users will execute the malicious command, making the malware propagate faster (i.e., requires less hops to infect the same number of victims), as we would expect. For instance, in the 5000-node OSN, it takes four hops when $p = 1$ and six hops when $p = 0.5$ to infect a total of 2,000 phones.

We also observe that when $p = 1$, metric N, the number of newly infected phones, reaches the maximum values when $t = 5$ (Fig. 1(a) and Fig. 1(c)) or $t = 6$ (Fig. 1(b)). The result is consistent with a phenomenon called *six degree of separation*, which refers to the idea that on average any two persons on earth could be connected through at most five acquaintances [16]. (According to study of 5.2 billion relationships on Twitter by the social media monitoring firm Sysomos, the most common friendship distance is five steps, and the average distance is 4.67 steps [5].)

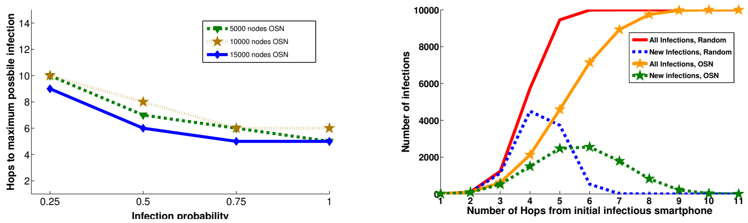


(a) 5,000 nodes

(b) 10,000 nodes

(c) 15,000 nodes

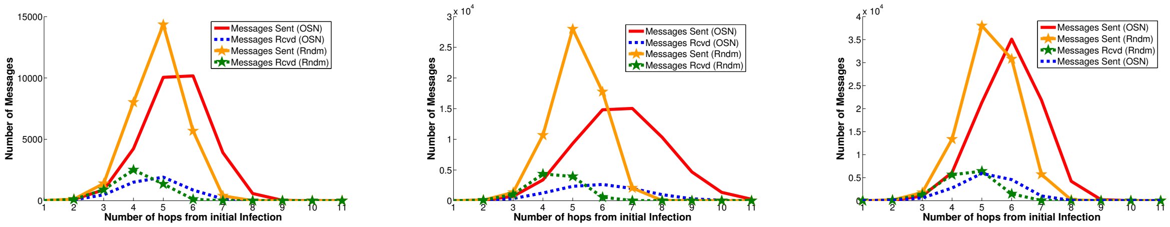
Fig. 1. The first set of experiments - Scenario 1



(a) Varying probability p

(b) OSN vs. random graphs

Fig. 2. The first set of experiments - Scenarios 2 and 3



(a) 5,000 nodes

(b) 10,000 nodes

(c) 15,000 nodes

Fig. 3. The second set of experiments

V-A2. Scenario 2

To further study the effect of the command executing probability p on metric N, we ran experiments on the three OSNs defined in TableI using p values of 0.25, 0.5, 0.75 and 1. The goal is to determine the point in time where the malware reaches the maximum number of potential victims. The results, shown in Fig.2(a), shows that as p increases, the malware propagates faster (i.e., requires less hops to infect the same number of new victims). However, as p gets larger, above 0.5, its impact on metric N becomes negligible. The malware can reach most of the uninfected users in the OSN within five or six hops of the first victim. Our results once again are consistent with the six degree of separation phenomenon in OSNs.

V-A3. Scenario 3

In this experiment, we compare the 10000-node OSN graph (see TableI) with its equivalent random graph (see TableII). The result in Fig2(b) shows that metric N, the number of newly infected nodes, reaches the maximum value when $t = 6$ (at the 6th hop) in the original OSN graph, and when $t = 4$ (at the 4th hop) in the ERG. At time $t = 4$, the total number of infected smartphones in the OSN graph is 1501 versus 4497 in the random graph. This indicates that the malware propagates faster in the random graph. The reason is that the ERG has a lower clustering coefficient than the original OSN graph, 0.003 vs. 0.157. A higher clustering coefficient implies that a message will circulate for a while in a community among friends before reaching to other parts of the OSN, slowing down the malware propagation.

The above result implies that an attacker may prefer to construct an equivalent random graph from an OSN graph, and use it to propagate the malicious command instead of the original OSN graph in order to speed up the malware propagation. (However, the botnets in the random graph generated much more messages than those in the original OSN graph as will be discussed next, which may raise a red flag in the network.)

V-B. The Second Set: Number of Messages

In this set of experiments, we also used the network graphs and malware propagation model described in SectionIV. The result graphs in Fig.3 show the total number of messages M sent by all the infected phones via the OSN over time and the total number of non-duplicate messages accepted by nodes in the OSN over time for each of the OSN graphs (defined in Table I) and their equivalent random graphs (defined in Table II).

We can see that the value M obtained from an ERG is significantly higher than that from the original OSN graph. For example, in the 5000-node networks, when $t = 4$, the ERG gives $M = 8021$ messages, while the value M given by the OSN is 4234 messages. We observe a similar trend on the larger networks. The results demonstrate that the bots in an ERG send out much more messages than those in the original OSN. This could alert the network administrator to the presence of the botnet. Although a random graph helps a malicious command propagate faster as discussed in the previous section, it also incurs the risk of making the botnet more vulnerable to detection. Therefore, equivalent random graphs are not a good choice for SoCellBot attacks.

VI. DISCUSSION AND FUTURE WORK

For our future work, we will propose an analytical model characterizing the propagation of the malware spread by a SoCellBot botnet. In this section, we discuss existing models of malware propagation and direction for our SoCellBot analytical model.

There exists research in the field of epidemiology that models the behavior of contagious diseases in society [19], [3], [22]. These models have been applied to modeling of malware propagation in computer networks [29], [10], [6], [18] and OSNs [19], [10].

Let $P(k)$ be the probability that a node in the network graph has degree k. The average degree of the network is thus $E[k] = \Sigma_k kP(k)$. 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 infection rate for nodes with degree k is given by the following differential equation [19]:

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

where

$$\Theta(t) = \frac{\sum_n nP(n)i_n(t)}{\sum_n nP(n)} = \frac{\sum_n nP(n)i_n(t)}{E[k]} \qquad (2)$$

The factor $\Theta(t)$ is the probability that an edge is connected to an infectious user (smartphone). Variable n takes values in the range $[d_{min}, d_{max}]$, where d_{min} and d_{max} are the minimum and maximum node degree in the OSN graph, respectively.

The above model does not consider the high clustering characteristic of OSNs, which helps to slow down the propagation of a malware in an OSN[9]. To characterize the impact of the clustering coefficient C on the propagation speed of a malware, Wu et al.[25] add a factor $f(C)$ to Eq. (1) and (2), as follows:

$$\frac{di_k(t)}{dt} = \lambda k[1 - i_k(t)]f(C)\Theta(t) \qquad (3)$$

Our simulation results presented above suggest that the probability p of executing the malware plays an important role in the malware propagation (Fig.1 and 2(a)). We thus suggest a factor called $g(p)$ that characterizes the effect of probability p on the malware propagation. The suggested analytical model for our smartphone malware is as follows:

$$\frac{di_k(t)}{dt} = \lambda k[1 - i_k(t)]f(C)g(P)\Theta(t) \qquad (4)$$

The factors $f(C)$ and $g(p)$ will accurately model the effects of the clustering coefficient and user behaviors, respectively, on the propagation of the smartphone malware. Solutions to $f(C)$ and $g(p)$ are left as future work.

VII. CONCLUSION

As smartphones are getting more powerful and capable of Internet connectivity, they become potential targets of malware attacks. In this paper, we present the design of a new mobile botnet that utilizes OSNs to transmit commands and control messages. Our simulation results indicate that OSNs are more suitable for mobile botnet communications than the traditional SMS in terms traffic load, propagation speed, reachability, robustness, detectability and monetary cost. Most cellular network providers offer free OSN access to their subscribers. The highly clustered structure of OSNs make the botnet immune from random node failures. Messages sent in OSNs are mostly encrypted, making it difficult for cellular network providers to block these botnet messages. Based on the simulation results, we also observe that equivalent random graphs allow a malware to propagate faster than the original OSN graphs. However, the botnet in an ERG floods the network with a much higher volume of messages than in the original OSN graph, which may alert the OSN administrator to the attack. To the best of our knowledge, our cellular botnet design is the first that exploits OSNs to transmit commands and control messages, and considers the characteristics of real social networks (i.e., low average network distance, high clustering, and power-law distributed node degrees).

VIII. REFERENCES

[1] P. Barford and V. Yegneswaran. ”An inside look at botnets”. In *Proc. Special Workshop on Malware Detection, Advances in Information Security*, Springer Verlag, 2006.

[2] V. Batagelj, A. Mrvar, ”Pajek a Program for large network analysis.”, In *Connections*, 21: 47-57, 1998

[3] M. Boguna, R. Pastor-Satorras, A. Vespignani, ”Epidemic spreading in complex networks with degree correlations.” *Lecture Notes in Physics: Statistical Mechanics of Complex Networks*, 625, 127-147., 2003

[4] Y. Boshmaf , I. Muslukhov, K. Beznosov, M. Ripeanu, ”The Socialbot Network: When Bots Socialize for Fame and Money”, In *Proc. of the 27th ACSAC*, Orlando, USA, 2011

[5] A. Cheng, “Six Degrees of Separation, Twitter Style”, April 2010, www.sysomos.com/insidetwitter/sixdegrees/

[6] S. M. Cheng, et al. ”On modeling malware propagation in generalized social networks,” *IEEE Comm. Lett.*, vol. 15. no. 1, pp. 25-27, Jan. 2011.

[7] J. Davidsen, H. Ebel, S. Bornholdt, ”Emergence of a Small World from Local Interactions: Modeling Acquaintance Networks”, In *Physical Review Letters*, vol. 88(12), 128701-1:4.

[8] A. H. Dekker, ”Realistic Social Networks for Simulation using Network Rewiring”, In *Proc. of International Congress on Modelling and Simulation*, 2008

[9] M. R. Faghani, H. Saidi. Malware propagation in online social networks, In *Proc. 4th Malicious and Unwanted programs (MALWARE09)*, pp 8-14, 2009

[10] M. R. Faghani, H. Saidi, Social networks XSS worms, In *Proc. CSE09*, pp. 1137-1141., 2009

[11] A.P. Felt, M. Finifter, E. Chin, S. Hanna, D. Wagner, ”A Survey of Mobile Malware in the Wild”. In *Proc. of the ACM SPSM11*, 2011

[12] J. B. Grizzard , V. Sharma , C. Nunnery , B. ByungHoon Kang , D. Dagon, ”Peer-to-peer botnets: overview and case study”, In *Proc. of First Workshop on Hot Topics in Understanding Botnets*, 2007

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

[14] J. Hua and K. Sakurai, ”A sms-based mobile botnet using flooding algorithm”, In *Proc. 5th WISTP*, 2011.

[15] R. Ihaka, R. Gentleman,”R: a language for data analysis and graphics.”, In *J. Comput. Graph. Stat.*, vol. 5(3), 299314.

[16] J. Kleinfeld, ”Could it be a Big World After All? The Six Degrees of Separation Myth”. *Society*, April 2002

[17] F. Viger, F. Latapy,”Efficient and Simple Generation of Random Simple Connected Graphs with Prescribed Degree Sequence”, *Lecture notes in computer science*, vol. 3595

[18] M. Mannan and P. C. van Oorschot. ”On instant messaging worms, analysis and countermeasures,” In *Proc. of ACM WORM*, pp. 2-11., 2005

[19] Y. Moreno, J. B. Gomez, and A. F. Pacheco, ”Epidemic incidence in correlated complex networks,” *Physical Review E*, vol. 68, pp. 035103, 2003.

[20] C. Mulliner, J.P. Seifert,”Rise of the iBots: 0wning a telco network”. In *Proc. MALWARE 2010*, France 2010

[21] M. Newman, ”Power laws, Pareto distributions and Zipf law”, In *Contemporary Physics*, vol. 46(5), pp. 323-351.

[22] R. Pastor-Satorras and A. Vespignani ”Epidemic spreading in scale-free networks”, *Phys. Rev. Lett.*, vol. 86, p.3200 , 2001.

[23] K. Singh, S. Sangal, N. Jain, P. Traynor, W. Lee, ”Evaluating Bluetooth as a Medium for Botnet Command and Control”. In *Proc. DIMVA 2010*, 2010

[24] P. Traynor, M. Lin, M. Ongtang, et al., ”On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core”. In *Proc. CCS 2009*, Chicago, USA 2009

[25] X. Wu, Z. Liu, ”How community structure influences epidemic spread in social networks“, *Phys. A: Stat. Mech. App.*, vol 387, pp. 623-630, January, 2008.

[26] C. Xiang, F. Binxing, Y. Lihua et al, ”Andbot: Towards Advanced Mobile Botnets,” *4th Usenix Workshop on Large-scale Exploits and Emergent Threats* 2011

[27] A. Yong-Yeol, Seungyeop, H. Kaok, S. Moon, S., H. Jeong, ”Analysis of topological characteristics of huge online social networking services”, In *Proc. of the 16th International conference on World Wide Web*, 2007.

[28] Y. Zeng, X. Hu, K.G. Shin, ”Design of SMS Commanded-and-Controlled and P2P-Structured Mobile Botnets”. *University of Michigan Technical Report CSETR- 562-10*, 2010

[29] C. Zou, D. Towsley, W. Gong, ” Code red worm propagation modeling and analysis”, In *Proc. of ACM Conference on Computer and Communications Security*, pp. 138-147, 2002.