Analysis of Malware Propagation in Twitter

Ameya Sanzgiri, Andrew Hughes and Shambhu Upadhyaya
Computer Science and Engineering, University at Buffalo, Buffalo, New York 14260
{ams76,ahughes6,shambhu}@buffalo.edu

***Abstract*—Malware propagation in social networks is a potential risk that has not been well-studied yet as there are no formal threat models for social networks. In this paper we investigate the vulnerability and cost of spreading malware via Twitter. Towards this end we present three specific attack scenarios targeted for Twitter and systematically analyze the cost of staging each of these attacks. Our analysis presents the first step for understanding the threats on the security of a class of social networks. We identify the attack related parameters and verify these parameters by testing the attack on a NetLogo-based simulator. Our analysis indicates that the cost of staging attacks to infect users of Twitter is low and that the proposed attack scenarios are plausible. Further, even with a low degree of connectivity and a low probability of clicking links, Twitter and its structure can be exploited by such attacks to infect many users with malware.**

I. Introduction

Malware and its propagation is a difficult problem to solve. In the past, spammers used traditional "social-networks" such as emails and newsgroups to entice unsuspecting users to install and then propagate worms. The advent of Pay-Per-Install (PPI), which help "miscreants to outsource the global dissemination of their malware" [1] has led to a diversification of malware propagation attempts. The PPIs are institutions that aim at spreading malicious software for financial gain. One such target of these attempts is the on-line social network such as Facebook or Twitter. Online social networks are Internet based schemes that could provide a convenient way for malware propagation since there are clearly defined paths already set up.

Facebook showcased its vulnerability when it was targeted by *koobface* [2] and $clickjacking$ worms [3]. Twitter has also been targeted by spammers in the past, who targeted overloading Twitter's servers [3]. Further, early attempts have also been made on Twitter (as detailed in Sec. II-C) which prove that Twitter is being targeted to spread malware. Malware propagation via Twitter could be devastating, even with a low probability of infection and low degree of connectivity as described in [4]. The rise of such attacks, targeting online social networks leverages the facts that these technologies have not fully matured and its users are not completely educated on the risks.

However, the risk of exploiting these technologies to spread malware, by the groups which promote either Pay-Per-Install or Pay-per-Click (PPC) is high since the motivation is financial gain. These groups not only cooperate with each other but also share their resources with each other [1]. Towards this end, they have the resources (both manpower and expertise) to launch sophisticated attacks that leverage complementary technologies such as *short-URLs* and target advertisement. Thus, the threat of these institutions tricking unwitting customers into downloading and propagating malware also becomes high.

In this paper we investigate and build a formal threat model for Twitter that aims at malware propagation. We present three attack scenarios and mathematically analyze the costs and impacts of the attacks using probabilistic techniques. Our results indicate that a miscreant (an attacker who aims at spreading malware) can launch low cost attacks on Twitter and can still infect a large number of users. The contributions of this paper are:

1. Modeling Twitter for the understanding of the various vulnerabilities in its user-interactions.
3. Creating several real-world attack scenarios that exploit these vulnerabilities by considering the specifics of Twitter such as its structure and inherent relationships.
4. Identifying the attack related parameters and verifying the same via simulation.
5. Mathematically analyzing the impact and cost of such attacks and hence the feasibility.

The rest of the paper is organized as follows: The preliminaries and background appear in Section II. A common attack and a complex indirect attack are introduced in sections III and IV respectively. Extension of this attack to the # - tag model of Twitter appears in Section V. After identifying the attack-related factors and validating it via simulations, we present a cost analysis of launching such attacks in Section VI. We also present the results of our detailed analysis in this section. Finally, we compare our work with related literature in Section VII and give our conclusions in Section VIII.

II. Background

A. Twitter User Model

The structure of Twitter can be visualized as two distinct entities: a User $\rightarrow$ Follower model and a # - tag model. The user $\rightarrow$ follower entity abstracts the dissemination model of information from a user to her followers, other users who "follow" a user. Information dissemination occurs through *tweets* which are Twitter specific messages. The tweets, which are a broadcast to the world, have a limit of 140 characters. By using the string *@username* at specific positions, a tweet is classified either as a direct message or a "mention" thus bringing the specific message to the attention of the user. The tweets of a user are available only to her followers and cannot be accessed by anyone else. However, if a follower of the original user

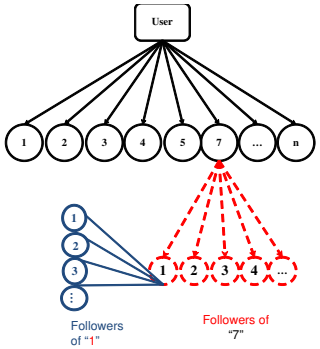

Fig. 1. Twitter structure

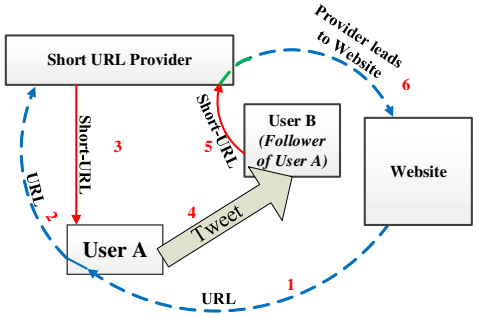

Fig. 2. How short-URLs work in Twitter (numbers depict sequence of operation)

"retweets" the tweet, the followers of the said follower gain access to the tweet. Figure 1 depicts this model and process, where a user's tweets are accessible to her followers only. When a tweet is retweeted by follower number seven, her followers (indicated in red dashed circles) gain access to the user's tweet. On being retweeted by a follower (numbered 1), users who are not followers of the original user also gain access to the tweet. As we can see, the User $\rightarrow$ Follower model of Twitter has a tree structure, where the information flow occurs down the tree. It is important to also note that, unlike other social networks, the relationship between a user and her follower can be asymmetrical. Specifically, when a user gains a follower, they both do not automatically follow each other, thus a user does not necessarily gain access to *all* the tweets of their followers.

Tweets can also be used to broadcast information about specific topics by appending "#-tags" to it. These #-tags are used in determining "trending topics list," which describes the topics that are generating most interest (in a geographic location). These # -tags have been extensively used in market research, disseminating political opinions and obtaining current news. Many Twitter users actively use and follow these #-tags for communication and networking. The #-tag entity comprises of such users. It is important to note that any user of Twitter falls either into the User $\rightarrow$ Follower model and/or a # - tag model. This also includes users who are not following anyone or any *trending* topic.

The User $\rightarrow$ Follower and # - tag entities make Twitter a unique model from a security/privacy perspective, since they provides two avenues for a miscreant to leverage. Specifically, the #-tag model provides a miscreant an opportunity to attack a network of user-follower entities which is spatially isolated from other networks of user-follower entities.

B. Twitter Vulnerabilities

Twitter like various other online social networks, inherently possesses the risks of some miscreants using the medium to share malicious ideas, executables and worms. Personal information can be gleaned through Twitter conversations, that a malicious user can leverage into social engineering kind of attacks [5]. The threat of attacks on Twitter can be realistic since these attacks can misuse the trust between users and combine it with other vulnerabilities such as the strict character limit for tweets. Due to the severe limitation on tweets lengths, users use short-Universal Resource Locators in tweets instead of standard URLs. Short-URLs are normal URLs that are encoded into URLs with fewer characters, and can thus be used in tweets. However, short-URLs have some inherent issues of concern. First, some services encode the same input URL into different (unique) short-URLs for different users. Second, unlike traditional systems, a user cannot follow the target of the short-URL (by hovering their mouse over the URL). The short-URL providers such as bit.ly [6] or tiny URL [7] services are required to decode them. Figure 2 shows the process of using an URL in a tweet. The dashed blue arrows in the figure depict the use of the normal URL, whereas the solid red arrows depict the use of the short-URL. As can be seen from the figure, a user has a very limited knowledge of the target of the short-URLs. The encoding of URLs is a method of obfuscating information, which can be exploited into tricking unwilling users to download/spread malicious software without their knowledge. In the following section we show how a miscreant can leverage this information to stage a variety of attacks on Twitter. Our goal is not to investigate the attacks targeted on Twitter (its infrastructure or availability) but rather the attacks on the users of Twitter. This precludes attacks such as spamming of tweets that aims at overloading Twitter's servers. As explained in Section I, the emergence of PPI could make Twitter users an attractive target for spreading malware.

C. Attacks on Twitter

Twitter has been under various attacks ever since its conception. The attacks launched on Twitter and its users have not only become more complex, but due to the permeance of Twitter in the social and cultural context of society, have also been able to target many users. Figure 3 shows a timeline of attacks on Twitter and its users from various media reports. While the details of the attacks are not clear/available, it is obvious that miscreants view Twitter as a viable avenue for attacks. It is important to note that the attackers on Twitter have leveraged not only on the specifics of Twitter but also on

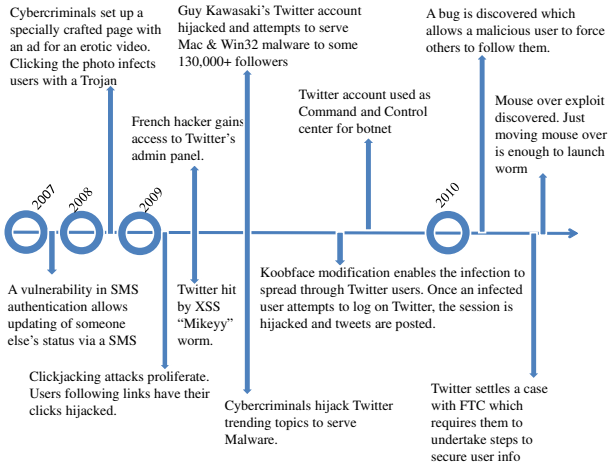

Fig. 3. Timeline of attacks on Twitter

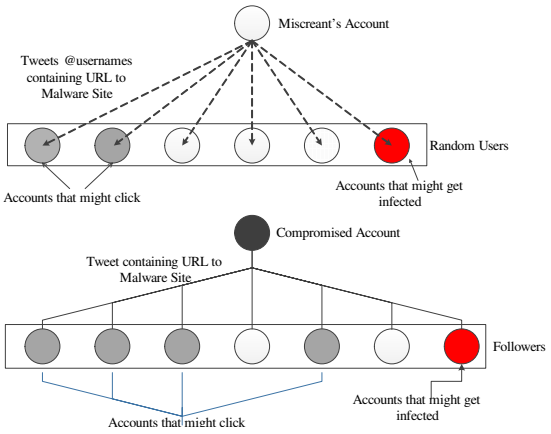

Fig. 4. Attack scenarios depicting a simple attack

the vulnerabilities of the Internet. Figure 3 showcases how vulnerable Twitter is as an attack avenue for malicious activities such as propagation of malware, especially in the case of zero-day attacks. Next, we develop three conceivable attacks aimed at malware propagation and present their analyses. We start with a simple and common attack methodology, progressing into more complex attacks.

III. A Common Attack Methodology

In this section we begin with the conceptualization and analysis of a simple attack that follows a more direct approach to spread malware via Twitter and then proceed to more complex attacks where the attacks leverage the Twitter structure and use its dissemination mechanics.

A. A Simple Attack

To spread malware using Twitter and its users, any miscreant would have to first encode the malware site as a short-URL. Now to disseminate this information she would have two approaches – a) Use as many *@username* in her tweets and hope some users click on the link or b) Compromise and control a user account and then post the tweet to her followers. Figure 4 depicts the two common attack methodologies. The upper part of the figure, depicts the methodology of sending directed messages to users while the lower part of the figure shows the process of using a compromised account to send tweets to followers. We will briefly analyze the pros and cons of both these methods shortly. In its simplest form, this attack would be similar to the *koobface botnet* attack which misled users into going to a malware site and then forced them to download the malware under the pretence of updating their flash player or other software. However, this attack depends on the probability of the infection of the malware and the probability of a user clicking on a link.

The attack introduced above is a simplistic one that has the potential of infecting many users at the same time. However, from a practical consideration, there are certain aspects we need to consider for this attack.

Analysis: Of the two attack scenarios presented, the attack in the first scenario while being low cost in terms of resources, has two important obstacles. First, Twitter has strict spamming standards; a user is termed as a spammer if either a user follows too many users or (and) posts many *@username* posts. Even if a miscreant is able to get around this obstacle, the other issue is that the probability of a user clicking on a @username from an unknown user (and in an unknown context) will be low. In the second attack scenario, a miscreant will have to spend considerable time and resources to compromise and control an account. However, once in control of a user account, the miscreant can then proceed with the attack as mentioned. The probability of other users (followers of the user) clicking on a link in this case is higher than the case described earlier, simply because the trust between a follower and a user is high due to their interactions over time. Another aspect to consider is that the propagation/installation of malware depends on the degree of infectivity of the malware, which can be affected by devices used by the victims, operating systems, frequency of patching operating systems, etc. The attacks described in this, may not infect many users due to the reasons explained above. Further, this attack does not really consider or leverage the User $\rightarrow$ Follower model, which would allow a miscreant to reach deeper into the network. In the following section we present an advanced version of this attack which follows the principle of the common attack but also leverages the Twitter structure and present the analyses on this advanced attack.

B. An Advanced Self-Propagating Attack

Based on the premises of the above simple attack one could conceive an advanced attack that leverages the User $\rightarrow$ Follower model of Twitter. The advanced self-propagating attack uses the *clickjacking* technology. There are two important considerations to take into account while designing this attack. First, the attack needs to exploit the inherent trust between a user and a follower. As explained above a link is more likely to be clicked by a follower of a user than by another non-following user. Second, the attacker needs to consider that in

Twitter, information (in this specific case the malicious short-URL) can only propagate down the tree if it is retweeted by the followers. Thus, the advanced attack would need to involve *clickjacking* such that the tweet retweets itself whenever a user of Twitter (follower) clicks on the link. The clickjacking attack can also exploit the weakness of short-URL providers which encodes a new short-URL for different users. This attack has the benefits of propagating down the Twitter tree, with the additional benefits of making it difficult for Twitter to analyze the different short-URLs due to the amount of information generated and traversing through the network.

Analysis: Let us assume a Twitter tree structure as shown in Fig. 1, with a depth $d+1$ and the average number of followers for each user as η. Let each follower with a probability ρ_{click} clicks on a link in a tweet and with a probability $\rho_{retweet}$, retweets a link. For the User $\rightarrow$ Follower model we can assume that $\rho_{click} \geq \rho_{retweet}$ since not all followers might retweet it. Now, the equation for the total number of users (N_1) in the tree of depth $d+1$, who would have see a benign link in the tweet would be

$$N_1 = \sum_{i=1}^{d} \left({\rho_{click}}^i \times {\rho_{retweet}}^{i-1} \times \eta^i \right) + 1 \qquad (1)$$

where the term "1" is for the main user who starts propagating/tweeting the particular tweet. Similarly, considering the advanced attack, where retweeting happens automatically, the total number of people who can see the link would be:

$$N_2 = \sum_{i=0}^{\mathrm{d}} \left({\rho_{click}}^i \times \eta^i \right) \qquad (2)$$

where N_2 describes the number of users who see the malicious link. Here, $N_2 > N_1$ since retweeting happens automatically in the advanced attack and would thus go further down the Twitter tree structure. Similarly, the calculation for N_2 starts from 0 since, there is a chance that no one sees the tweet, unlike in the calculation of N_1 where at least "1" user has to start the tweet. It is important to notice that as the depth of the tree increases (i.e., $\mathrm{d} \rightarrow \infty$), the number of users who can see the malicious link will also increase and $N_2 \rightarrow N$, where N represents all the users of Twitter. Thus, the attack would theoretically encompass all the users of Twitter; which is unlikely. Another consideration is that even in this case, N_2 or N, does not denote *the number of infections*, rather it is a measure of the number of users who are susceptible to the malware. The total number of infections depends on the probability of infection of the malware. Simply put,

$$\mathrm{N_{infection}} = \rho_{malware} \times \mathrm{N} \qquad (3)$$

where $N_{infection}$ is the total number of infections and $\rho_{malware}$ represents the probability of infection of a malware.

IV. A Complex Indirect Attack

In our analysis and attacks we have so far assumed that a miscreant either randomly tweets to other users or compromises and controls a genuine user account and then engages in the attack. The success of the attack depends on the ability of a miscreant to effectively and efficiently compromise user accounts which is a cost intensive process. Further, while Twitter may be a new medium, the authentication mechanisms of usernames and passwords is not, which means that if users use strong passwords, it might be difficult to crack. This means that the cost is going to rise exponentially, when the miscreant tries to take over more than one account to ensure a high probability of success. Further, a miscreant may get diminishing returns for her investment due to a number of dependencies (probabilistic) for successful infection. However, this does not imply a low level of risk for attack on Twitter. An important aspect for consideration is that the compromise of an account is not limited to the purview of Twitter; the clickjacking attack in principle can be modified to take place even outside the purview of Twitter. Let us assume a more plausible attack scenario, where a Twitter user is surfing public websites which also allow users to enter links to other websites, such as blogs and news sites. Such sites provide an avenue for a miscreant to insert malicious short-URLs, and clicking on the links would instigate the advanced attack if the victim is also a Twitter user. Given this scenario, we assume that the miscreant is inserting links in between conversations people are having in the comments section of a popular news article. Assuming that any user of the website will randomly click on a malicious link (given that a user will click on a link on the website) the following factors affect the probability of the user clicking on the malicious link:

1. A posting strategy by miscreants in relation to the number of posts at a given time such that they control the majority of the posts.
2. The probability of a miscreant posting the malicious links at any given time frame.
3. The user's probability of clicking on a link.

The first factor represents the number of posts that are occurring by other users. In relation to this, a miscreant also needs to input her links so that the malicious links can be seen by users and thus have a *chance* of being clicked on. The second factor represents the probability of posting links to help miscreants circumvent detection techniques implemented against spammers. Further, there could be many such spammers who also are attacking such websites; a miscreant with the aim of spreading malware may also be competing with other PPIs or PPCs. This probability helps the miscreant to post links based on her discretion (if done manually). Finally, these strategies do not make any sense, if no users click on any of the links. We make the following assumptions for all cases:

1. There are some users who are reading/clicking/entering links (benign or malicious) on a certain website. The total number of such users is denoted by $\eta_{website}$ and the probability of a user clicking on a link is ρ_{web_click}.
2. The number of posts being entered by users other than the miscreant is some function of the number of users who are on a particular website, i.e., $\Phi(\eta_{website})$.
3. Similarly, the number of posts (links) by the miscreant is also based on the number of users who are on a particular site, i.e., $\Psi(\eta_{website})$.

In the following subsection, we analyze the complex attack scenarios for the two possible posting strategies 1) posting malicious links all the time, and 2) posting links based on some probability.

Analysis of the Complex Attack Scenario

1) Posting malicious links all the time: In this scenario a miscreant posts a malicious link (mal-link) every time another innocent/malicious post is made. Given that a link was clicked by a user, the probability of the link being malicious is

$$\Pr(\text{click a mal\_link}|\text{clicked a link}) = \frac{\Psi\left(\eta_{website}\right)}{\Psi\left(\eta_{website}\right) + \Phi\left(\eta_{website}\right)} \tag{4}$$

If we assume that $\Psi(\eta_{website}) = \Phi(\eta_{website}) + \varepsilon$, such that $0 < \varepsilon < \eta_{website}$, then we can rewrite the above equation as

$$\rho_{link} \approx \frac{1}{2} + \varepsilon' \tag{5}$$

where ρ_{link} is the conditional probability $\Pr(\text{click a mal\_link}|\text{clicked a link})$ and $0 < \varepsilon' < \frac{1}{2}$. Now the probability of the malicious link being clicked is $1 -$ Probability that the malicious link is not clicked; i.e., a user clicked on a link, but the link was not the malicious link or the user did not click at all. If the attacks were repeated over x trials, then we could rewrite the equation as,

$$\Pr(\text{click}_{mal\_link}) = 1 - \Big[\Pr(\text{click})\times (1 - \Pr(\text{click mal\_link}|\text{clicked a link})) + (1 - \Pr(\text{click}))\Big]^x \tag{6}$$

Using the definitions from Eq. (4) and Eq. (5), the Probability of clicking on a malicious link simplifies to,

$$\Pr(\text{click}_{mal\_link}) = 1 - \Big[\rho_{web\_click} \times (1 - \rho_{link}) + (1 - \rho_{web\_click})\Big]^x \tag{7}$$

For one instance of the attack (i.e., $x = 1$), the probability of clicking a malicious link would simply be,

$$\Pr(\text{click}_{mal\_link}) = \rho_{web\_click} \times \rho_{link} \tag{8}$$

While the probability of some user clicking on the malicious link increases with more number of trials, a miscreant might not want to post links all the time due to the non-zero cost (in terms of both time and resources) involved in posting the link. Further, there is always the risk of being termed as a spammer. To avoid these situations, a miscreant might need to post the link based on a probability (denoted as q). The analysis of this scenario follows next.

2) The probabilistic scenario: A miscreant can either post a link or not post a link, in a probabilistic sense. Then the only aspect that changes from the previous scenarios, will be the conditional probability in Eq. (5). Thus, the probability now becomes:

$$\rho_{link} = \left(\frac{1}{2} + \varepsilon'\right) \times q \tag{9}$$

where q is the probability with which the miscreant posts her links. This new probability can be replaced in Eq. (7) to get the probability of a user clicking on a malicious link.

The analysis so far presents the probability of a user clicking on a malicious link using different malicious link insertion strategies. The result of clicking the malicious link is that the followers of the users (who we now call the *root*) become susceptible to the malware. Thus, the number of users susceptible to malware by clicking on the malicious link, by any user is a function of:

1. The probability of a user clicking on the root, $\Pr(\text{click mal\_link})$, which we will now denote as $P_{\text{mal\_link}}$.
2. The number of people who are active on the website, the miscreant is targeting.
3. The depth of the User $\rightarrow$ Follower model, i.e., the total number of followers of the said user and their followers.

Hence, we can write the number of susceptible users to propagate the malware as:

$$N_{\text{susceptible}} = P_{\text{mal\_link}} \times \eta_{\text{website}} \times \underbrace{\sum_{i=0}^{d} (\rho_{\text{click\_Twitter}} \times \eta_{\text{Twitter}})^i}_{\text{A}} \tag{10}$$

where the term A is a form of Eq. (2) from Section III-B; the terms $\rho_{\text{click\_Twitter}}$ and $\eta_{Twitter}$ represent the probability of clicking links in tweets (different from clicking links in websites, i.e., $\rho_{\text{click\_Twitter}} \neq \rho_{web\_click}$) and the average number of followers per user in the User $\rightarrow$ Follower model, respectively. Similarly, the number of infected users can be written as

$$\text{N}_{\text{infection}} = \rho_{malware} \times \text{N}_{\text{susceptible}}$$

At this point, we would like to clarify that based on the attack and its analysis we have presented, a miscreant can have potentially more strategies (at a higher or lower cost) by tweaking the parameters. However, the overall strategy of probabilistically posting malicious links still remains the same. Further details will be provided in Section VI.

V. Extension to hash tags

The analysis presented so far is based on attacks that primarily target the User $\rightarrow$ Follower model of Twitter. However, one aspect of Twitter that is unique is the # - tag model which provides a miscreant the ability to infect users that are not connected in any way to an infected network. This model can be exploited to propagate malware deeper and to newer networks. The attack to target this particular model can be constructed in conjunction with the attack that targets the User $\rightarrow$ Follower model by simply appending an #-tag to the tweets. This makes the tweet visible to those users of Twitter, who follow only #-tags. The attack then behaves as described earlier, targeting the followers of this particular user who belong to a different network. The analysis of this attack is also similar to the analysis of miscreants inserting links into websites as discussed in Section IV. The strategies of posting the tweets also remain largely the same with the exception of the following two choices:

1. Appending a #-tag that is already trending.

2. Appending a new #-tag that the miscreant creates.

Both these choices directly affect the probability of the malicious link being clicked (which is now encoded in a tweet) and thus the number of susceptible users.

Analysis of Attack in the #-Tag Model

In the #-tag model, the factors that affect the probability of a malicious link being clicked in Eq. (4) and Eq. (7) are:

- The number of users following a particular #-tag ($\eta_{\text{\#-tag}}$) and their probabilities of clicking on a link in a tweet with #-tags ($\rho_{\text{click \#-tag}}$, where $\rho_{\text{click \#-tag}} \leq \rho_{web\_click}$).
- The number of posts that are being generated by other users that are appended with #-tags ($\Phi_{\text{\#-tag}}$).

Miscreant enters trending #-Tag: If a miscreant starts using an already trending #-tag, the factors that affect the probability of the malicious link being clicked are the number of people who are following the trend and the rate with which the posts are made. Further, there exists the cost of analyzing topics that would maximize the chance of people clicking on links. Similarly, the miscreant also has to evaluate the duration that a topic may remain trending. For example, a trending topic that is related to local news will have a smaller group of people following as compared to national level topics or a global level topic. At the same time, a local topic might have a higher chance of remaining a trending topic for a longer duration than a national or global topic. Thus, we can write the probability of a malicious link being clicked as

$$P_{\text{mal\_link \#-tag}} = 1 - \left[(\rho_{\text{click\_\#-tag}} \times (1 - \rho_{\text{link\_\#-tag}}) + (1 - \rho_{\text{click \#-tag}}) \right]^{x} \quad (11)$$

where x is the number of retries and $\rho_{\text{link\_\#-tag}} = \frac{1}{2} + \varepsilon'$ if the miscreant is appending the #-tag all the time or $\rho_{\text{link\_\#-tag}} = \left(\frac{1}{2} + \varepsilon'\right) \times q$ if the miscreant is appending the #-tag based on a probability q.

Miscreant creates her own trending #-tag: Similar to Section V, if the miscreant decides to create her own #-tag and appends it to all the tweets, the first factor she will have to consider is the probability of any user being interested in this topic and clicking on the link. However, a greater consideration will be the time it takes for this topic to actually become a trending topic. Simply put, the term $\rho_{link\_\#-tag}$ which is a function of the number of users following a given #-tag will now have to account for *all* the users of Twitter, since the #-tag will be competing at a global scale. This means that the number of posts that need to be generated will simply be too large for the miscreant to even have a chance for a user to click on which will also increase her cost by a large amount.

Thus, the case of a miscreant creating her own #-tag is too costly in terms of both time and resources to be considered by a miscreant, although other variations of the attack may still exist. However, those variations are beyond the scope of this paper.

VI. Cost Analysis, Simulation and Discussion

So far, we have conceptualized and analyzed attacks on Twitter using probabilistic methods for different strategies, which give us a probabilistic measure of the degree of success and penetration into the Twitter network. While all these attacks have been mathematically modeled we still need to identify what aspects/parameters can be controlled by the miscreant (to increase the chance of her success) and thus analyze the feasibility of these attacks. In the following section we first identify the parameters of our model that a miscreant will have to consider/control for an attack and validate the identification by empirically inserting values in the equations and analyzing the results. We then present results from a simulator we built using NetLogo [8], to simulate the propagation of attack on Twitter. Finally, we perform computational analysis to assess the amount of work for a miscreant to launch an attack on Twitter using our attack scenarios as well as the feasibility of attack on Twitter.

A. Parameters of Interest to a Miscreant

The models of attack albeit probabilistic, describe the relationships between the various parameters that an attacker has to consider. In the best case scenario, an attacker may execute her attack in a manner that gets the probability $P_{\text{mal\_link}}$ to be as close to 1 as possible, targeting a polynomial number of infections. However, the time taken and the cost of inserting links may make the attack moot and infeasible. Conversely, an attacker may choose a low degree of infection and a low probability of a user clicking on her link, to save cost and detection and repeat the attacks over longer time. Thus, at least at a high level, there is the inevitable tradeoff between the cost of an attack (in terms of time, resource and implementation of attack) and maximizing the number of susceptible users.

Analyzing the various equations for the different strategies of attack, we have the following factors that can be controlled by a miscreant:

1. The number x of trials.
2. The probability ρ_{link} of the malicious link being clicked, given that a link was clicked by a legitimate user. We assume this to be $\frac{1}{2} + \varepsilon'$. However, this depends on the number of posts by both legitimate users (Φ_{website}) and the miscreant (Ψ_{website}).
3. The probability $P_{\text{mal\_link}}$ that a malicious link was clicked.

However, not all factors can be controlled at the same time; a miscreant has to insert a number of posts depending on the number of posts by legitimate users to stay in contention. To try and maximize $P_{\text{mal\_link}}$(wanting), a miscreant can only do so after a certain number of trials x which is dependent on the probabilities of people clicking the links. The miscreant can also reduce the number of posts per trial to save the cost while lowering the degree of infection. The resulting "amount of work" and "number of susceptible" users under such conditions can only be probabilistically measured.

It is obvious that the success of the attack of a miscreant depends on the legitimate users' probability of clicking links (both malicious and benign). Even with low probabilities it is still quite possible for successful attacks. In Figure 5, under the assumption of a small probability for users clicking, we plot the probabilities of users clicking on a malicious link

for different number of trials and different probabilities (q) of miscreants posting links. It can be seen that a miscreant can achieve a high probability of $P_{\text{mal\_link}}$ by increasing the number of trials. Further, for even a small probability of posting links a miscreant can still guarantee a good probability of users clicking on the malicious link which would lower the chance of the miscreant being detected as a spammer.

Similarly, the number of susceptible users depends directly on the depth of the User $\rightarrow$ Follower model and the average number of followers at each depth (η_{Twitter}). Figure 6 shows the change in the number of susceptible users for an increasing number of followers and depths. The probabilities of users clicking links (ρ_{link}) and the number of trials of attack are fixed. The plot shows that even at small depth the attack can still affect a large number of users. Also, the cost of such an attack would be low for a miscreant, since the fixed parameters have been chosen conservatively.

Figure 7 provides more insight into the strategies a miscreant can leverage to increase the chance for a successful attack for different depths as well as different probabilities of users clicking links. Here we note that if the attacker can shape the attack by enticing users to click on links even by a small probability of 0.01, the number of susceptible users increases drastically. Further, by targeting users who have a lower average number of followers but a higher probability of clicking links, a miscreant can save on her cost of attack while still maintaining her target of susceptible users. This also validates the assumptions that successful attacks might be possible even when users do not have a high probability of clicking random links.

The figures above corroborate the identification of the parameters that a miscreant can potentially manipulate to increase her chance of success. In the following section we first validate the factors we have identified via simulations and then present the analysis of the amount of work (cost) required for a successful attack.

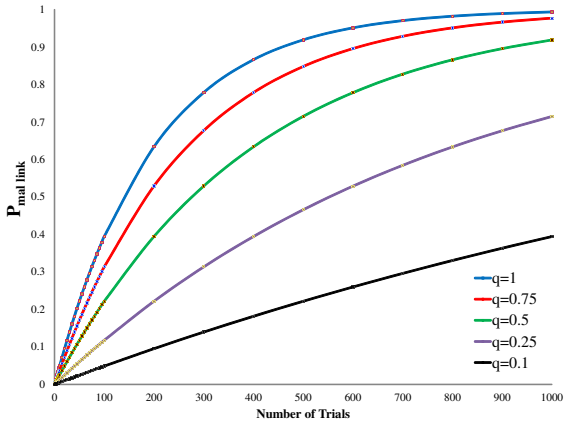

Fig. 5. The effect of the no. of trials on probability of malicious link being clicked

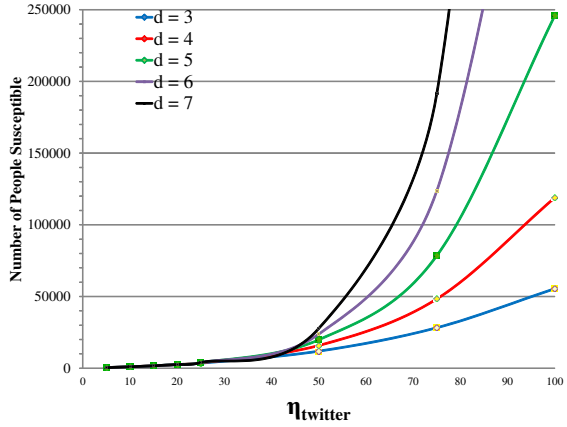

Fig. 6. The probabilistic estimate of no. of susceptible for different depths

B. Simulation Results

In the previous section, we identify three factors an attacker may be interested in, based on our attack scenarios. These factors were derived quantitatively from our probabilistic model, where some aspects such as the same number of followers for every user, the same probability of a follower to click on links, etc., might not hold true in the real world. To validate whether these factors would still hold true, we played our attack scenarios in a simulated platform using the agent-based simulation tool NetLogo. The Twitter simulator is built using the NetLogo programming language. The simulator creates a Twitter like structure, for a user specific number of nodes, and configures them into the User $\rightarrow$ Follower model. This configuration is done probabilistically based on a preferential attachment model [9], where the nodes are likely to follow some user with more followers or could randomly connect themselves to other nodes.

Experimental Set Up and Procedure: The NetLogo simulator allows configuration of parameters such as number of users, maximum probability of clicking on a link, probability of retweeting, max probability of getting infected, max probability of tweeting new content as well as list of tweets that are viewable for each node. Similarly, one can also configure parameters such as probability of inserting comments on a blog and clicking of links on a blog for *each* node. The latter parameters are used in simulating the complex attack scenario. A separate node (not part of the Twitter structure), termed as a miscreant is used to insert malicious links into the "blog" based on a probability (q). Figure 8 shows a screenshot of the simulator along with some set up parameters such as type of network topology, number of users, etc. The circles in the screenshot depict the various users and the connections between them based on the User $\rightarrow$ Follower model. On starting the experiment, the simulator randomly cycles through the nodes and based on the probability takes one of the following actions – either creates and tweets new content, simply views a tweet or retweets a tweet (if any are viewable). If a retweet occurs, the simulator will update the viewable tweet list of all of the followers. We also configure the simulator to insert links (based on a probability) if it tweeted new content. Similarly, each node based on a probability, will either insert a comment on the blog, click on a blog entry if

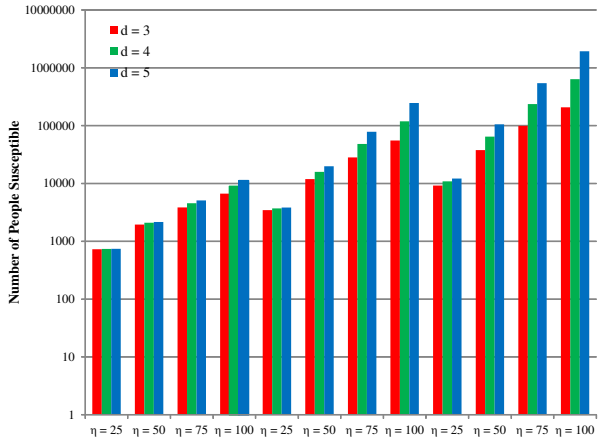

Fig. 7. The effect of varying the no. of followers and probability of clicking on links on the number of susceptible users

it could view it or do nothing related to the blog. To model a real world scenario, we set the probability of acting on a blog to be less than that of Twitter, unless a certain number of comments had been inserted in the blog in consecutive "ticks" of simulation.

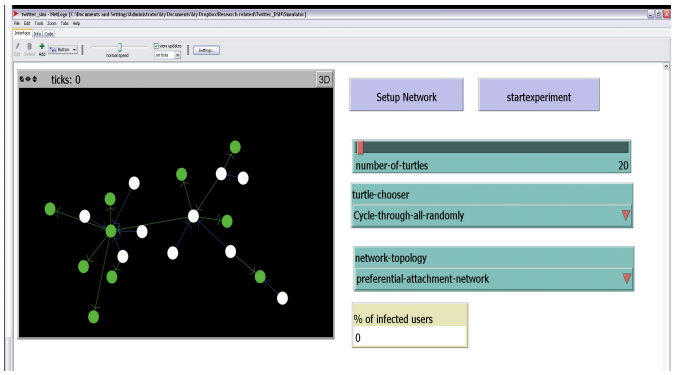

Fig. 8. Screenshot of Twitter simulator for illustration purposes only

Infection and Susceptible: If a node clicks on a link from the miscreant, the simulator randomly generates a number and compares it to the node's probability of infection. If the number generated is higher, then the node is termed to be infected. Subsequently, all further contents from the infected node are deemed to contain malicious links, if they have links inserted. The simulator also ascertains in a similar manner, if a follower node of the infected user has been infected. All simulations are run until 90% of the nodes are infected. All results presented in the following section are averaged over 2000 runs.

Simulation Results: Figure 9 shows the comparison of the theoretical and simulated probabilities of a malicious link for different number of trials when a miscreant has different probabilities of inserting the links in a website or blog. As can be seen from the figure, the simulated values are very close to theoretical calculations with a maximum error of 10%. The simulated and theoretical probabilities of the malicious link are equal when the probability of inserting links is 0.75, thus causing an overlap of lines in the plot. This shows that the chances of a user clicking on a malicious link is affected not only by the probability of inserting malicious links but also by the number of trials the miscreant inserts links.

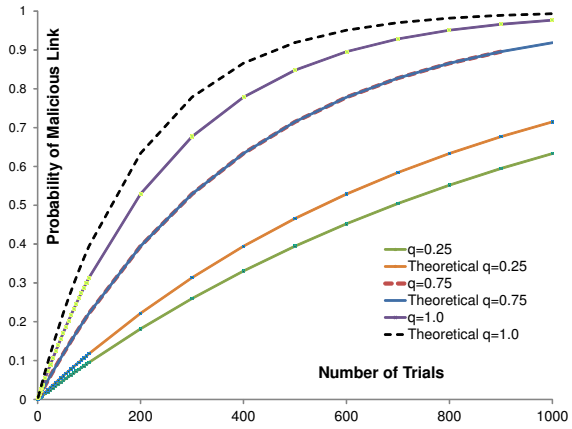

Fig. 9. Comparison of theoretical and simulated probability of malicious links for various number of trials

Figure 10 shows the plot of the number of infected users for different click probabilities. From the plot, we can infer that once the top level users are infected from the blog there is a significant increase in the number of followers getting infected in a short span of time. This follows our intuition that once the top level users are infected, the malicious link propagates faster through the Twitter tree structure. An interesting point to observe is that, even an increase by 0.01 in the clicking probabilities causes a significant increase in the number of infected users. Further, we can observe that if the click probability is small (0.01), it takes significant time to infect the top level users, thus delaying the infection down the tree.

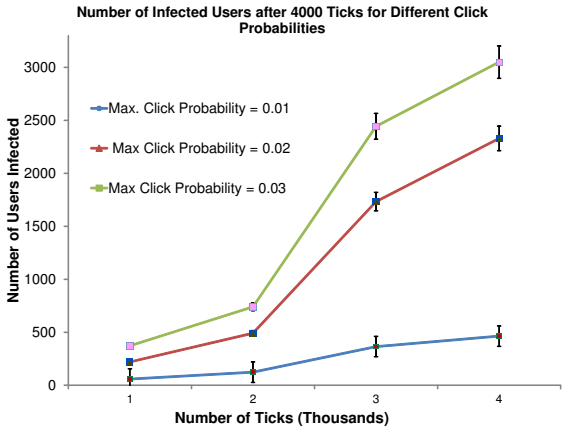

Fig. 10. Number of infected users for different click probabilities

C. Cost Analysis

In this section we perform the cost analysis of launching an attack on Twitter and its users by a miscreant and summarize it in the form of a table. Table I gives the analysis for the amount of work that occurs for different $P_{\text{mal\_link}}$, assuming different

TABLE I
COST ANALYSIS TARGETING

$\Phi(\eta_{web})$ (Assume)	$P_{\text{mal\_link}}$ (Aim)	$\Psi(\eta_{web})$ (Forced)	**No. of Trials** x	**No. of Susceptible users**	**Amount of Work (Cost)**
$\log(\eta_{web})$	≈ 1	$\log(\eta_{web})$	poly	$lin(\eta_{web})$	poly-$\log(\eta_{web})$
$lin(\eta_{web})$	≈ 1	$lin(\eta_{web})$	poly	$lin(\eta_{web})$	poly(η_{web})
$lin(\eta_{web})$	$\approx \frac{log(\eta_{web})}{lin(\eta_{web})}$	$\log(\eta_{web})$	poly	$\log(\eta_{web})$	poly-$\log(\eta_{web})$

TABLE II
COST ANALYSIS TARGETING NUMBER OF SUSCEPTIBLE

No. of of Susceptible (Aiming)	Ψ_{website} (Assumed)	**No. Trials** (Forced)	**Amount of Work**
exp(η_{web})	$log(\eta_{web})$	exp	Quasi-Poly $(2^{[log(\eta_{web})]^k})$
exp(η_{web})	$lin(\eta_{web})$	exp	exp $(2^{(\eta_{web})^k})$
poly(η_{web})	$log(\eta_{web})$	poly	poly-log(η_{web})
poly(η_{web})	$lin(\eta_{web})$	poly	poly(η_{web})

user posting activity (Φ_{website}) and with the miscreant aiming for different probabilities of users clicking the malicious links. The amount of work a miscreant has to perform is the product of the work that she is *forced* to do ($\Psi(\eta_{web})$), along with the work she needs to do to increase her chance of success – the number of trials x. For instance, from the first entry of Table I, if we assume that the activity of legitimate users on a website is $log(\eta_{web})$ and the miscreant aims at the probability of any user clicking on her malicious link ($P_{\text{mal\_link}}$) to be close to 1, then by Eq. (4), we can see that the miscreant is forced to at least match the activity so that her link can be seen, i.e., she is also forced to do $log(\eta_{web})$ amount of work. This is because her chance of a successful attack is directly dependent on the activity (or response) from legitimate users. Since it is a factor she cannot control, the only way $P_{\text{mal\_link}}$ will be close to 1 is, if she repeats the number of trials (x) to a polynomial number of times. Thus, the total amount of work in theoretical terms is poly-log(η_{web}). It is important to note that under these conditions, the number of susceptible users will be a linear function of the legitimate users and this number cannot be controlled by the miscreant. Further, the miscreant cannot aim at a specific number of susceptible users.

Similarly, if the miscreant is to aim for or target specific number of susceptible users, while assuming that there is certain activity from legitimate users, she is forced to increase the number of trials. In this particular scenario, the probability of users clicking on malicious links does not really factor in, since she is targeting a "specified number" of website users. This means that the miscreant has to match the posting activity of the legitimate users, i.e., $\Psi_{\text{website}} \approx \Phi_{\text{website}}$. Table II shows a summary of the resulting amount of work required from a miscreant, when she targets for a specific number of susceptibles while matching the legitimate users' posting activities. It is important to note that the legitimate users' posting activities will never go beyond some fraction of followers (η_{web}) which is a linear factor of η_{web}. It can be seen that if the relative user activity is less (say $\log(\eta_{web})$), a miscreant can still target a high number of susceptible users by repeating the attack an exponential number of times at a quasi-poly amount of work. Similarly, by targeting for a lower number of legitimate users she can save cost. Overall, the tables show that the work required by the miscreant while being large, is not impossible. Thus, probabilistically at least, the attacks on Twitter are definitely feasible for a determined miscreant.

D. Discussion

From the tables we infer that the amount of work for a miscreant when targeting for a certain number of susceptibles is not prohibitively large under our attack scenarios. However, the cost analysis in the paper has not considered certain factors or accounted for events that could affect the attack. In the following paragraphs, we first present some factors that have not been considered as well as the reason behind and then present the reasons for lack of real world experiments.

1) Factors not considered: One of the most important factors we have not considered in this paper is *time*. While we have made certain observations regarding time, they are mostly limited to factors which can be quantified. This is due to the reason that it is not probabilistically or deterministically possible to model the duration of an attack or other durations such as the time taken to insert links, activity of other users, etc.

Another aspect that has not been considered in this work is the diversity of devices involved. The emergence of smartphones and other hand-held devices has resulted in new methods of accessing and interacting with the Internet resources which might accelerate or decelerate malware propagation.

The factors that work against a miscreant also requires close attention. First, the attack proposed in this paper to a large extent requires human expertise in activities such as choosing blogs, articles and semantically constructing the correct sentences. The effort that goes into this activity has been abstracted in this work. Second, mechanisms of banning the miscreant from making more posts by other users could adversely affect the attack. If such an event occurs in the middle of an attack, the attack could possibly get voided completely. The bigger repercussion however would also be the loss of the miscreant's account; the creation and maintenance of which adds to the cost of an attack, which has been abstracted. Further, by tweaking the parameters of attacks and changing

the posting strategies can also affect the attacks and the cost. A careful consideration and analysis are however required to understand the results of the variations and this part is outside the scope of the paper.

2) Lack of real-world experiments: Finally, the cost analysis in this paper is based on a probabilistic model which provides the best-case scenario for an attack and accounting for real world factors. Verifying this model requires real experiments and user study to ensure the validity of the model. However, it may be noted that the experiments may present scenarios that are not captured by our model, since the factors such as activities on websites, clicking on links are dependent on individuals and their personality. Similarly, the results from these experiments may not be completely reproducible. These details are beyond the scope of this paper.

VII. Comparison with Related Work

Malware propagation has been a long studied topic in network security. Malware propagation in unconventional networks such as scale-free networks [10], [11], wireless sensor networks [12], [13], cellular networks (using Multimedia Messaging Service (MMS) and Bluetooth) [2] as well as traditional networks has been studied in [3], [14]. Worm and spammer based attacks on social networks have recently led researchers to focus on the security of online social networks using simulated topologies and user activities such as in [15]–[17]. A recent focus of researchers has been the understanding of how information flows in social networks such as Facebook and Twitter [18]–[20] who have used this information to detect spammers in online social networks. Our work differs from these works in three aspects. First, we present a formal model that captures the specifics of Twitter, which also models the spread of malware using Twitter. Second, we identify and validate via simulations the attack-parameters based on our model and attack scenarios. Finally, we present a comprehensive analysis in terms of cost that reflects the different methodologies of attack aimed at malware propagation via Twitter.

VIII. Conclusion

In this paper, we have presented an attack model and analyzed Twitter as a malware propagation medium. Our attacks leverage Twitter's inherent models, obfuscation of information by short-URLs and clickjacking methods that are common methodologies of web-based attacks in the real world. Our attacks also present strategies that model user behaviors and considered other avenues/entry-points of attacks.We have theoretically identified different factors that an attacker needs to consider to be successful and have also validated these attack-parameters using our NetLogo simulator. We have also shown via our probabilistic model that such attacks are feasible and that even with a low degree of connectivity an attacker can infect many users. Since the success of these attacks depends on the personal choices of people, it is difficult to obtain real world data regarding such attacks. This makes formulating effective mitigation techniques challenging. In the future, we intend to improve the model to capture more information and perform extensive simulations, which would aid in our final goal of creating effective mitigation techniques.

References

[1] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring Pay-per-Install: The Commoditization of Malware Distribution," in *Proc. of the the 20th USENIX Security Symposium*, San Francisco, CA, Aug. 2011.

[2] C. Shin-Ming, A. Weng Chon, C. Pin-Yu, and C. Kwang-Cheng, "On Modeling Malware Propagation in Generalized Social Networks," *Communications Letters, IEEE*, vol. 15, no. 1, pp. 25–27, 2011.

[3] M. Mannan and P. C. van Oorschot, "On Instant Messaging Worms, Analysis and Countermeasures," in *Proc. of the 2005 ACM Workshop on Rapid Malcode-WORM'05*. New York, NY, USA: ACM, 2005, pp. 2–11.

[4] A. Sanzgiri, J. Joyce, and S. Upadhyaya, "The Early (tweet-ing) Bird Spreads the Worm: An Assessment of Twitter for Malware Propagation," *Procedia Computer Science*, vol. 10, pp. 705–712, 2012.

[5] D. Siegel, "On the new threats of social engineering exploiting social networks," Bachelor's Thesis, Technische Universität München, Aug. 2009.

[6] "Bitly— Do More With Your Links." [Online]. Available: https://bitly.com/

[7] "TinyURL.com - Shorten That Long URL in a Tiny URL." [Online]. Available: http://tinyurl.com/

[8] S. Tisue and U. Wilensky, "NetLogo: A Simple Environment for Modeling Complexity," in *in International Conference on Complex Systems*, 2004, pp. 16–21.

[9] A. D. Flaxman, A. M. Frieze, and J. Vera, "A Geometric Preferential Attachment Model of Networks," in *Algorithms and Models for the Web-Graph: Third International Workshop, WAW 2004*. Springer, 2004, pp. 44–55.

[10] L. Briesemeister, P. Lincoln, and P. Porras, "Epidemic Profiles and Defense of Scale-Free Networks," in *Proc. of the 2003 ACM workshop on Rapid malcode – WORM '03*. New York, NY, USA: ACM, 2003, pp. 67–75.

[11] C. Griffin and R. Brooks, "A Note on the Spread of Worms in Scale-Free Networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 1, pp. 198–202, 2006.

[12] P. De, L. Yonghe, and S. K. Das, "An Epidemic Theoretic Framework for Vulnerability Analysis of Broadcast Protocols in Wireless Sensor Networks," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 3, pp. 413–425, 2009.

[13] R. Di Pietro and N. V. Verde, "Introducing Epidemic Models for Data Survivability in Unattended Wireless Sensor Networks," in *In Proc. of WoWMoM*, 20-24 June 2011, pp. 1–6.

[14] C. C. Zou, D. Towsley, and G. Weibo, "Email Worm Modeling and Defense," in *Computer Communications and Networks, 2004. ICCCN 2004. Proc. 13th International Conference on*, 11-13 Oct. 2004, pp. 409–414.

[15] M. R. Faghani and H. Saidi, "Malware propagation in Online Social Networks," in *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on*, 13-14 Oct. 2009, pp. 8–14.

[16] G. Yan, G. Chen, S. Eidenbenz, and N. Li, "Malware Propagation In Online Social Networks: Nature, Dynamics, and Defense Implications," in *Proc. of the 6th ACM Symposium on Information, Computer and Communications Security*. Hong Kong, China: ACM, 2011, pp. 196–206.

[17] W. Xu, F. Zhang, and S. Zhu, "Toward Worm Detection in Online Social Networks," in *Proc. of the 26th Annual Computer Security Applications Conference - ACSAC '10*. New York, NY, USA: ACM, 2010, pp. 11–20.

[18] K. Lerman and R. Ghosh, "Information Contagion: An Empirical Study of the Spread of News on Digg and Twitter Social Networks," in *Proc. of 4th International Conference on Weblogs and Social Media (ICWSM)*, 2010.

[19] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting Spammers on Twitter," in *Proc. of the Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, Washington, DC, USA, 2010.

[20] K. Beck, "Analyzing Tweets to Identify Malicious Messages," in *2011 IEEE International Conference on Electro/Information Technology (EIT 2011)*, 15-17 May 2011.