SocialImpact: Systematic Analysis of Underground Social Dynamics *

Ziming Zhao, Gail-Joon Ahn, Hongxin Hu and Deepinder Mahi

Laboratory of Security Engineering for Future Computing (SEFCOM)
Arizona State University, Tempe, AZ 85281, USA
{zmzhao, gahn, hxhu, dmahi}@asu.edu

Abstract. Existing research on net-centric attacks has focused on the detection of attack events on network side and the removal of rogue programs from client side. However, such approaches largely overlook the way on how attack tools and unwanted programs are developed and distributed. Recent studies in underground economy reveal that suspicious attackers heavily utilize online social networks to form special interest groups and distribute malicious code. Consequently, examining social dynamics, as a novel way to complement existing research efforts, is imperative to systematically identify attackers and tactically cope with net-centric threats. In this paper, we seek a way to understand and analyze social dynamics relevant to net-centric attacks and propose a suite of measures called SocialImpact for systematically discovering and mining adversarial evidence. We also demonstrate the feasibility and applicability of our approach by implementing a proof-of-concept prototype *Cassandra* with a case study on real-world data archived from the Internet.

1 Introduction

Today's malware-infected computers are deliberately grouped as large scale destructive botnets to steal sensitive information and attack critical net-centric production systems [1]. The situation keeps getting worse when botnets make use of legitimate social media, such as Facebook and Twitter, to launch botnet attacks [2]. Previous research efforts on countering botnet attacks could be classified into four categories: (i) capturing malware samples [3], (ii) collecting and correlating network and host behaviors of malware [27], (iii) understanding the logic of malware [4], and (iv) infiltrating and taking over botnets [5].

Notably, most studies in the area of countering malware and botnets have been focused on detecting bot deployment, capturing and controlling bot behaviors. However, there is little research on examining how these malicious programs are created, rented and sold by adversaries. Even though preventive solutions

* This work was partially supported by the grants from National Science Foundation (NSF-IIS-0900970 and NSF-CNS-0831360). All correspondence should be addressed to Dr. Gail-Joon Ahn, gahn@asu.edu.

against thousands of known bots have been deployed on networked systems, and some botnets were even taken down by law enforcement agencies [6], the majority of adversaries are still at large and keep threatening the Internet by developing more bots and launching more net-centric attacks. The major reason for this phenomenon is that previous malware-related activities–such as developing, renting and selling bots–occurred mostly offline, which were way beyond the scope of security analysts.

In recent years, the pursuit of more profit in underground communities leads to the requirement for global collaboration among adversaries, which tremendously changed the division of labor and means of communication among them [8]. (Un)fortunately, adversaries started to communicate with each other, distribute and improve attack tools with the help of the Internet, which leaves security analysts new clues for evidence acquisition and investigation on unwanted program development and trade. Before the widespread use of online social networks (OSNs), adversaries would communicate via electronic bulletin board systems (BBS), forums, and Email systems [10].

Content-rich Web 2.0, ubiquitous computing equipments, and newly emerging online social networks provide an even bigger arena for adversaries. In particular, the value of OSNs for adversaries is the capability to cooperate with destructive botnets. The role of OSNs in botnet attacks is twofold: first, OSNs are the platforms to form online black markets, release bots, and coordinate attacks [3, 9]; second, OSN user accounts act as bots to perform malicious actions [7] or C&C server nodes coordinates other networked bots [2]. Although our efforts in this paper are mainly concerned about the former case, our proposed model for online underground social dynamics and corresponding social metrics can be also utilized to identify compromised and suspicious OSN profiles.

Given the great amount of valuable information in online social dynamics, the investigation of the relationships between online underground social communities and network attack events are imperative to tactically cope with net-centric threats. In this paper, we propose a novel solution using social dynamics analysis to counter malware and botnet attacks as a complement to existing research investments.

The major contributions of this paper are summarized as follows:

- We formulate an online underground social dynamics considering both social relationships and user-generated contents.
- We propose a suite of measures named SocialImpact to systematically quantify social impacts of individuals and groups along with their online conversations which facilitate adversarial evidence acquisition and investigation.
- We implement a proof-of-concept system based on our proposed model and measures, and evaluate our solution with real-world data archived from the Internet. Our results clearly demonstrate the effectiveness of our approach for understanding, discovering, and mining adversarial behaviors.

The rest of this paper is organized as follows. Section 2 presents our online underground social dynamics model and addresses SocialImpact, which is a

systematic ranking analysis suite for mining adversarial evidence based on the model. In Section 3, we discuss the design and implementation of our proof-of-concept system *Cassandra*. Section 4 presents the evaluation of our approach followed by the related work in Section 5. Section 6 concludes this paper.

2 SocialImpact: Bring Order to Online Underground Social Dynamics

In this section, we first address the modeling approach we utilized to represent online underground social dynamics (OUSDs). Unlike existing OSN models [11] which emphasize on user profile, friendship link, and user group, our model gives attention to user-generated contents due to the fact that a wealth of information resides in online conversations. We also elaborate the design principles of social metrics to identify adversarial behaviors in OUSDs. Then, we present SocialImpact, which consists of nine indices, to bring order to underground social dynamics based on our OUSD model.

2.1 Online Underground Social Dynamics Model

As shown in Figure 1, an OUSD can be represented by six fundamental entities and five basic types of unidirectional relationships between them.

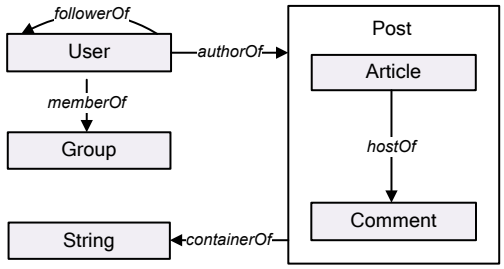


Fig. 1. *OUSD* Model: Entities and Relationships

Users are those who have profiles in the network and have the rights to join groups, post articles, and give comments to others. *Groups* are those to which users can belong. In an OUSD, groups are mainly formed based on common interests. *Articles* are posted by users who want to share them with the society. In an OUSD, articles might introduce the latest technologies, analyze recent vulnerabilities, call for participation of network attacks, and trade newly developed and deployed botnets. In terms of the form of articles, they do not have to be literary. They could also contain multimedia contents, such as photos and melodies. *Comments* are the subsequent posts to articles. *Posts* are the union of articles and comments. *Strings* are the elementary components of articles and comments. Strings are not necessarily meaningful words. They could be names, URLs, and underground slangs. A user has a relationship *authorOf* with each post s/he creates. A user has a relationship *followerOf* with each user

s/he follows. A user has a relationship *memberOf* with each group s/he joins. An article has a relationship *hostOf* with each comment it receives. A post has a relationship *containerOf* with each string it consists of.

The following formal description summarizes the above-mentioned entities and relationships.

Definition 2.1 ***(Online Underground Social Dynamics).*** *An* ***OUSD*** *is modeled with the following components:*

- U *is a set of users;*
- G *is a set of user groups;*
- A *is a set of articles;*
- C *is a set of comments;*
- P *is a set of posts.* $P = A \cup C$*;*
- S *is a set of strings;*
- $UP = \{(u,p)|\ u \in U, p \in P$ *and* u *has an* authorOf *relationship with* $p\}$ *is a one-to-many user-to-post relation denoting a user and her posts;*
- $FL = \{(u,y)|\ u \in U, y \in U$ *and* u *has a* followerOf *relationship with* $y\}$ *is a many-to-many user-to-user follow relation;*
- $MB = \{(u,g)|\ u \in U, g \in G$ *and* u *has a* memberOf *relationship with* $g\}$ *is a many-to-many user-to-group membership relation;*
- $AC = \{(a,c)|\ a \in A, c \in C$ *and* a *has a* hostOf *relationship with* $c\}$ *is a one-to-many article-to-comment relation denoting an article and its following comments; and*
- $PS = \{(p,s)|\ p \in P, s \in S$ *and* p *has a* containerOf *relationship with* $s\}$ *is a many-to-many post-to-string relation.*

We focus on the main structure and activities in online underground society and overlook some sophisticated features & functionalities, such as online chatting, provided by specific OSNs and BBS. Hence, our OUSD model is generic and can be a reference model for most real-world OSNs and BBS. As a result, security analysts could easily map real-world social dynamics data archived from any OSNs and BBS to our model for further analysis and investigation.

2.2 Principles of Metric Design and Definitions

We also address the following critical issues related to evidence mining in underground society: How can we identify adversaries among a crowd of social users? Given the additional evidence acquired from other sources, how can we correlate them with underground social dynamics? How can we measure the evolution in underground community? To answer these questions, we articulate several *principles* that the measures for underground social dynamics analysis should follow: 1) The measures should support identifications of interesting adversaries and groups based on both their social relationships and online conversations; 2) The measures should be able to take external evidence into account and support interactions with security analysts; and 3) The measures should support temporal analysis for the better understanding of the evolution in adversarial groups.

To this end, we introduce several feature vectors to achieve aforementioned goals. For the mathematical notations, we use lower case bold roman letters such as $\mathbf{x}$ to denote vectors, and uppercase bold roman letters such as $\mathbf{V}$ to denote matrices. We assume all vectors to be column vectors and a superscript T to denote the transposition of a matrix or vector. We also define $max()$ as a function to return the maximum value of a set.

Definition 2.2 ***(Article Influence Vector).*** *Given an article $a \in A$, the article influence vector of a is defined as $\mathbf{v}_a^T = (v_1, v_2, v_3)$, where v_1 is the length of the article, $v_2 = |\{c \mid c \in C \text{ and } (a, c) \in AC\}|$ is the number of comments received by a, and v_3 is the number of outlinks it has.*

When stacking all articles' influence vector together, we get the **article influence matrix V**. We assess an article's influence by its activity generation, novelty and eloquence [12].

Definition 2.3 ***(Article Relevance Factor).*** *Given a set of strings $\mathbf{s} = \{s_1, s_2, ..., s_n\} \subseteq S$ and an article $a \in A$, article relevance factor, denoted as $r(a, \mathbf{s})$, is defined as the number of occurrence of strings $\mathbf{s}$ in the article a.*

The strings $\mathbf{s}$ could represent an external evidence that security analysts acquired from other sources and query keywords in which security analysts are interested.

Definition 2.4 ***(User Activeness Vector).*** *The user activeness vector of u is defined as $\mathbf{z}_u^T = (z_1, z_2, z_3)$, where $z_1 = |\{p \mid p \in P \text{ and } (u, p) \in UP\}|$ is the number of articles and comments u posted, $z_2 = |\{y \mid y \in U \text{ and } (u, y) \in FL\}|$ is the number of users u follows, and $z_3 = |\{g \mid g \in G \text{ and } (u, g) \in MB\}|$ is the number of groups u joins.*

We measure a user's activeness by the number of posts s/he sends, users s/he follows, and groups s/he joins. By aggregating all users' $\mathbf{z}_u$, we get **user activeness matrix Z**.

Definition 2.5 ***(Social Matrix).*** *Social matrix, denoted as $\mathbf{Q}$, is defined as a $|U| \times |U|$ square matrix with rows and columns corresponding to users. Let v be a user and N_v be the number of users v follows. $\mathbf{Q}_{u,v} = 1/N_v$, if $(v, u) \in FL$ and $\mathbf{Q}_{u,v} = 0$, otherwise.*

Social matrix is similar to transition matrix for hyperlinked webpages in PageRank. The sum of each column in social matrix is either 1 or 0, which depends on whether the vth column user follows any other user.

Definition 2.6 ***(δ-n Selection Vector).*** *A δ-n selection vector, denoted as $\mathbf{y}_\delta^n$, is defined as a boolean vector with n components and $\|\mathbf{y}_\delta^n\|_1 = \delta$.*

A δ-n selection vector is used to select a portion of elements for one set. For example, the top 10 influential articles of a user a could be represented by a selection vector $\mathbf{y}_{10}^{|A|}$ over the article set A. By stacking all users' δ-n selection vectors over the same set together, we get the **δ-n selection matrix $\mathbf{Y}_\delta^n$**.

2.3 Ranking Metrics

As shown in Figure 2, SocialImpact consists of nine indices, which are classified into three categories: string & post indices, user indices, and group indices. Each index in upper categories is computed by the indices from lower categories.

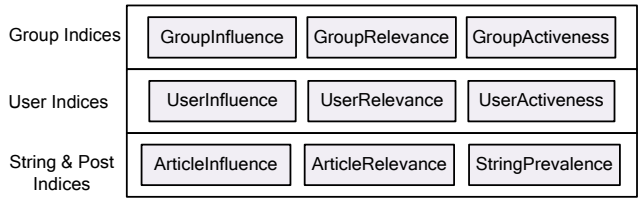


Fig. 2. SocialImpact: Systematic Ranking Indices

To fulfill *Principle 1*, user and group indices are devised to identify influential, active, and relevant users and groups. We devise personalized PageRank models [13] to calculate UserInfluence and UserRelevance, since it could capture the characteristics of both user-to-user relationships and user-generated contents in social dynamics. To accommodate *Principle 2*, ArticleRelevance, UserRelevance and GroupRelevance are designed to take external strings as inputs, combine them with existing data in social dynamics, and generate more comprehensive results. To fulfill *Principle 3*, all feature vectors and indices could be calculated for a given time window and StringPrevalence could indicate the topic evolution in the society. Moreover, we believe the combination of UserActiveness and UserInfluence could also be used to identify suspicious spam profiles in online social networks.

We consider a weighted additive model [14] when there exist several independent factors to determine one index. To reduce the bias introduced by different size of sets, we use δ-n selection vector to choose a portion of data in calculation. The followings are the detailed descriptions of indices.

ArticleInfluence, *denoted as* $x_1(a)$*, represents the influence of article* a*.* $x_1(a)$ *is computed as* $\mathbf{v}_a^T\mathbf{w_1}$*, where* $\mathbf{w_1}$ *denotes the weight vector.*

By normalizing $x_1(a)$ to $[0, 1]$ and stacking $x_1(a)$ from all articles together, we get a vector $\mathbf{x_1}$.

$$\mathbf{x_1} = \frac{\mathbf{V}^T\mathbf{w_1}}{max_{b\in A}(x_1(b))} \tag{1}$$

ArticleRelevance, *denoted as* $x_2(a, \mathbf{s})$*, represents the relevance of the article* a *to given strings* $\mathbf{s}$*.* $x_2(a, \mathbf{s})$ *is proportional to the occurrence of the given strings in the article and the influence of the article.*

$$x_2(a, \mathbf{s}) = \frac{r(a,\mathbf{s})x_1(a)}{max_{b\in A}(r(b,\mathbf{s})x_1(b))} \tag{2}$$

By stacking $x_2(a, \mathbf{s})$ from all users together, we get a vector $\mathbf{x_2}(\mathbf{s})$ denoting all articles' relevance to $\mathbf{s}$.

UserInfluence, *denoted as* x_3*, represents the influence of a user.* x_3 *can be measured by two parts. One is the impact of the user's opinions, which is modeled by* ArticleInfluence*. The other is the user's social relationships, which is modeled by* $\mathbf{Q}$*.* x_3 *is devised as a personalized PageRank function to capture both parts.*

By stacking x_3 from all users together, we get a vector $\mathbf{x_3}$.

$$\mathbf{x_3} = d_3\mathbf{Q}\mathbf{x_3} + (1 - d_3)\mathbf{Y}_\alpha^{|A|}\mathbf{x_1} \tag{3}$$

Where $d_3 \in (0,1)$ is the decay factor which makes the linear system stable and convergent. $\mathbf{Y}_{\alpha}^{|A|}$ is the $\delta - n$ selection matrix corresponding to all users's top α influential articles.

UserRelevance, *denoted as $x_4(\mathbf{s})$, represents the relevance of a user to strings* $\mathbf{s}$.

By stacking $x_4(\mathbf{s})$ from all users together, we get a vector $\mathbf{x_4}$.

$$\mathbf{x_4}(\mathbf{s}) = d_4 \mathbf{Q} \mathbf{x_4}(\mathbf{s}) + (1 - d_4)(\mathbf{Y}_{\alpha}^{|A|} \mathbf{x_2}(\mathbf{s})) \tag{4}$$

Where $d_4 \in (0,1)$ is the decay factor. $\mathbf{Y}_{\alpha}^{|A|}$ is a $\delta - n$ selection matrix corresponding to all users's top α relevant articles to $\mathbf{s}$.

UserActiveness, *denoted as x_5, represents the activeness of a user.*

$$\mathbf{x_5} = \mathbf{Z}^T \mathbf{w_5} \tag{5}$$

We use the addition of a group's top α members' influence, relevance, and activeness to model its influence, relevance, and activeness, respectively. As mentioned before, this model can reduce the bias caused by the number of members.

GroupInfluence, *denoted as x_6, represents the influence of a group.*

By stacking all x_6 together, we get $\mathbf{x_6}$.

$$\mathbf{x_6} = \mathbf{Y}_{\alpha}^{|U|} \mathbf{x_3} \tag{6}$$

Where $\mathbf{Y}_{\alpha}^{|U|}$ is the δ-n selection matrix corresponding to all groups' top α influential users.

GroupRelevance, *denoted as x_7, represents the relevance of a group to strings* $\mathbf{s}$.

By stacking all x_7 together, we get $\mathbf{x_7}$.

$$\mathbf{x_7} = \mathbf{Y}_{\alpha}^{|U|} \mathbf{x_4} \tag{7}$$

Where $\mathbf{Y}_{\alpha}^{|U|}$ is the δ-n selection matrix corresponding to all groups' top α relevant users.

GroupActiveness, *denoted as x_8, represents the activeness of a group.*

By stacking all x_8 together, we get $\mathbf{x_8}$.

$$\mathbf{x_8} = \mathbf{Y}_{\alpha}^{|U|} \mathbf{x_5} \tag{8}$$

Where $\mathbf{Y}_{\alpha}^{|U|}$ is the δ-n selection matrix corresponding to all groups' top α active users.

StringPrevalence, *denoted as $x_9(s)$, represents the popularity of a string s.*

$$x_9(s) = \sum_{p_j \in P} ti_{s,p_j} \tag{9}$$

where ti_{s,p_j} is the term frequency-inverse document frequency [15] of a string s in post p_j.

The computations for UserInfluence and UserRelevance are proven to be convergent [16]. And the corresponding time complexity is $O(|H|log(1/\epsilon))$, where $|H|$ is the number of $followerOf$ relationships in the social dynamics and ϵ is a given degree of precision [16]. The time complexity for calculating StringPrevalence is $O(|P||S|)$, where $|P|$ is the number of posts and $|S|$ is the size of string set. The complexities for all other indices are linear if the underlying indices are calculated.

3 *Cassandra*: System Design and Implementation

In this section, we describe the challenges in analyzing real-world underground social dynamics data. We address our efforts to cope with these challenges and present the design and implementation of our proof-of-concept system *Cassandra*.

3.1 Challenges from Real-world Data

The first challenge of real-world data is its multilingual contents. The most effective way of coping with this challenge is to take advantage of machine translation systems. *Cassandra* utilizes Google Translate[1] to detect the language of the contents and translate them into English. However, machine translation systems may fail to generate meaningful English interpretations for the following cases: i) adversaries may use cryptolanguages that no machine translation system could understand. For instance, *Fenya*, a Russian cant language that is usually used in prisons, is identified in online underground society [17]; and ii) both intentional and accidental misspellings are common in online underground society [18]. In order to cope with this challenge, *Cassandra* maintains a dictionary of known jargons, such as *c4n* as *can* and *sUm1* as *someone*.

Another challenge is that the social dynamics data may not be in a consistent format. Different OSNs use different styles in web page design. Even in one OSN, in order to make the web page more personalized, the OSN allows users to customize the format of their posts. Since HTML is not designed to be machine-understandable in the first place, extracting structural information from HTML is a tedious and heavy-labor work. To address this problem, we first cluster data, and then devise an HTML parser for each cluster. We also design a light-weight semi-structure language to store the information extracted from HTML.

Since one major component in social dynamics is the relationships between entities, storing and manipulating social dynamics data in a relational database become relatively time-consuming. We choose a graph database [19] which employs the concepts from graph theory, such as node, property, and edge, to realize faster operations for associative data sets.

3.2 System Architecture and Implementation

Figure 3 shows a high level architecture of *Cassandra*. The upper level of *Cassandra* includes several visualization modules and provides query control for security an-

[1] http://code.google.com/apis/language/translate/overview.html

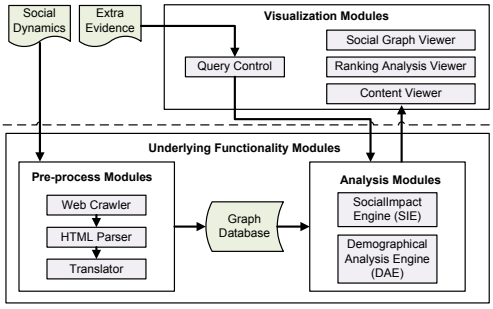
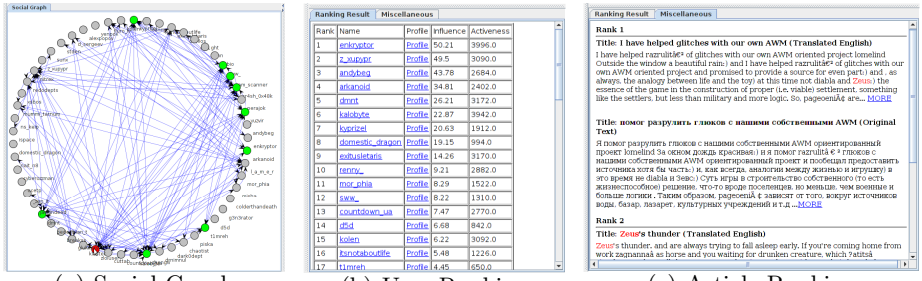


Fig. 3. System Architecture of *Cassandra*

alysts to provide the additional evidence. In reality, these evidences could be in the format of text, picture, video, audio or any other forms. Yet, representing multimedia contents like pictures and videos in a machine-understandable way is still a difficult challenge. *Cassandra* acts like a modern web search engine in response to keyword queries. Social graph viewer is designed to show social relationships among users and groups. Ranking analysis viewer is used to list the ranking results based on security analysts' queries. Content viewer can show both original and translated English web resources.

The lower level of the architecture realizes underlying functionalities addressed in our framework. After underground community data is crawled from the Internet, the HTML parser module extracts meaningful information from it. If the content is not in English, our translator takes over and generates English translation. All extracted information is stored in a graph database for the efficient retrieval. Analysis modules have two working modes: offline and online. The offline mode generates demographical information with demographical analysis engine (DAE) and intelligence, such as user influence and activeness, with SocialImpact engine (SIE). When security analysts provide the additional evidence, SocialImpact engine switches to online mode and generates analysis results, such as user relevance, based on data in the graph database and additional evidence provided by security analysts.

Cassandra was implemented in Java programming language. We took advantage of Java swing and JUNG to realize graphical user interfaces and graph visualization. As we mentioned before, *Cassandra* uses Google Translate API to translate texts. In most cases, Google Translate could output acceptable translations from original texts. *Cassandra* stores user profiles, user-generated contents, and social relationships among users in a Neo4j[2] graph database. For each group, user, article, and comment, *Cassandra* creates a node in the database, stores associated data–such as the birthday of user and the content of article–in each node's properties, and assigns the relationships among nodes.



(a) Social Graph (b) User Ranking (c) Article Ranking

Fig. 4. Screenshots of *Cassandra*

3.3 Visualization Interfaces of *Cassandra*

Figure 4 depicts interfaces of *Cassandra*. As illustrated in Figure 4(a), all users in a social group are displayed by a circle. And their *followerOf* relationships are displayed with curved arrows. It is clear to view that some users have lots of followers while others do not. By clicking any user in the group, *Cassandra* has the ability to highlight this user in red and all his followers in green. In this way, *Cassandra* helps analysts understand the social impact of any specific user. Another window as shown in Figure 4(b) displays the ranking results. Analysts can specify the ranking metric, such as UserInfluence and UserActiveness, to reorder the displayed rank. Clicking a user's name which is the second column in Figure 4(b) would bring the analysts to the list of all articles posted by the user in descending order of ArticleInfluence. Clicking the user's profile link which is the third column in Figure 4(b) would bring the analysts to the webpage of the user's profile archived from the Internet. Analysts could also specify some keywords in query control and *Cassandra* would display the results in descending order of ArticleRelevance. As shown in Figure 4(c), *Cassandra* displays both the original and translated texts and highlights the input keywords in red.

4 A Case Study on Real-world Online Underground Social Dynamics

In this section, we present our evaluation on real-world social dynamics. We evaluated *Cassandra* on 4GB of data crawled from *Livejournal.com* which is a popular online social network especially in the Russian-speaking countries. We anonymized the group names and user names in this OSN for preserving privacy.

All webpages in this OSN could be roughly divided into two categories in terms of content: i) profile and ii) article. A profile webpage contains basic information of a user or a group, which includes name, biography, location, birthday, friends, and members. Every article has title, author, posted time, content, and several comments by other users. The webpages are mainly *.html* files, along

[2] http://neo4j.org/

with some *.jpeg*, *.gif*, *.css*, and *.js* files. Our solution only considers text data from *.html* files.

We started to crawl group profiles from six famous underground groups in this OSN [3]. Then we crawled all members' profiles and articles of these six groups. We also collected one-hop friends' articles of these members. Therefore, we ended up with 29,614 articles posted by 6,364 users which are from 4,220 groups. Based on the information in user profiles, we noticed that about 32.7% and 52.7% users were born in early and mid-late 80's. This clearly illustrates the age distribution of active users in this community.

4.1 Post, User and Group Analysis

Cassandra calculated all articles' ArticleInfluence and identified top 50 articles over a time window of 48 months. Since not all of these articles are related to computer security, we checked these articles in descending order of their influences and picked five articles that are highly related to malware. We could observe some popular words related to malware, such as PE (the target and vehicle for Windows software attacks), exploits (a piece of code to trigger system vulnerabilities), hook (a technique to hijack legitimate control flow) and so on.

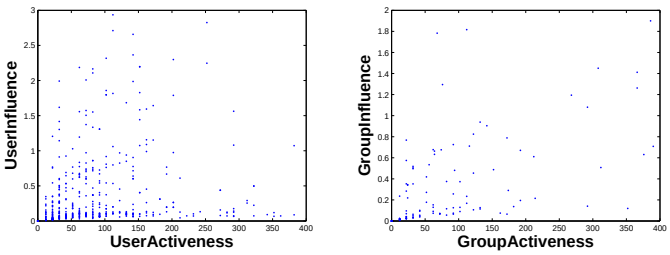
Top Five Influential Users		Top Five Active Users		Top Five Influential Groups		Top Five Active Groups	
User	UserInfluence	User	UserActiveness	Group	GroupInfluence	Group	GroupActivenss
z_xx_ur	49.5020	xsbxx_ur	4024	b_gp	344.4807	b_gp	57798
andxx_ur	43.7800	enkxx_ur	3942	c_gp	79.7781	d_gp	28644
arkxx_ur	34.8074	kalxx_ur	3936	d_gp	45.5222	demxx_gp	20846
_moxx_ur	26.7700	exixx_ur	3170	murxx_gp	26.2094	beaxx_gp	20290
kyp_ur	20.6292	kolxx_ur	3092	chrxx_gp	18.6487	_hoxx_gp	19486

Table 1. Top Five Influential/Active Users/Groups

Cassandra also generated each user's UserInfluence and UserActiveness and group's GroupInfluence and GroupActiveness over a time window of 48 months. And, Table 1 shows the top five influential/active users/groups for the entire period of our observation. We can notice that there is no overlap between the top five *influential* users and the top five *active* users, while there exists similarity for the top five *influential* groups and the top five *active* groups.

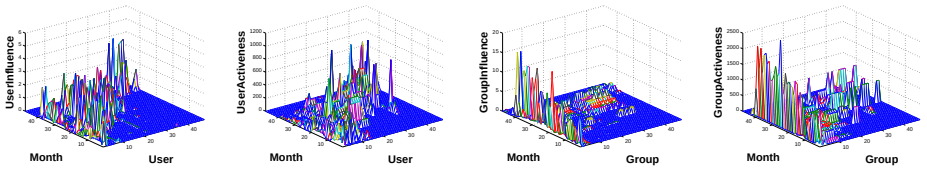
We calculated the correlation coefficient ($corrcoef$) for the pairs of UserInfluence and UserActivenss, GroupInfluence and GroupActivenss based on the results generated from *Cassandra*. Similar to the phenomenon we identified in Table 1, in Figure 5(a) we observed that the correlation coefficient between UserInfluence and UserActivenss is around 0.52 (the maximum value for correlation coefficient is 1 indicating a perfect positive correlation between two variables), which means one user's influence is not highly correlated to her/his activeness. This phenomenon indicates that talking more does not make a user more influential in a community. On the other hand, as shown in Figure 5(b) we observed that the correlation coefficient between GroupInfluence and GroupActivenss is around 0.90,

[3] These targeted groups are indicated by law enforcement agency who sponsored this project.



(a) $corrcoef = 0.5204$ (b) $corrcoef = 0.9094$

Fig. 5. Correlation Coefficient of UserActiveness & UserInfluence and GroupActiveness & GroupInfluence



(a) UserInfluence over 48 months (b) UserActiveness over 48 months (c) GroupInfluence over 48 months (d) GroupActiveness over 48 months

Fig. 6. Temporal Pattern Analysis

which indicates a very strong positive correlation between the influence and the activeness of a group. The application of influence and activeness indices is not limited to identify such a social phenomenon. We could also leverage the high UserActivenss and the low UserInfluence as indicators for the analysis of social spammers in any OSN.

The temporal patterns of the influential/active users/groups could be observed in Figure 6, where x-axis denotes the users/groups who were identified as the most influential/active ones for each month. For example, $x = 1$ denotes the most influential/active user/group of the first month in our time window and $x = 48$ denotes the most influential/active user/group of the last month in our time window; y-axis denotes the entire 48 months in the time window; and z-axis denotes user/group's influence/activeness value. As shown in Figure 6(a), some users maintain their influence status for several months. The large plain area in the right part of this figure indicates most users come as the most influential ones suddenly. This observation implies that a user does not need to be a veteran to be an influential one in the community. On the other side, we can see from Figure 6(b) that most active users remain active before they became the most active ones. The plain area in the left portion of Figure 6(b) implies that most users do not always keep active. Normally they keep active for 15 - 30 months, then get relatively silent. While the smaller plain area in the left part of Figure 6(a) shows once a user becomes influential, s/he keeps the status for a long period of time. Figure 6(c) shows that there are 2 or 3 groups who maintain the status of influence during the whole 48 months and get even more influential

(a) Results for Botnet

Keywords	Relevant Articles #
spam	490
botnet	44
zeus	9
rustock	1
mega-d	0

(b) Results for Identity Theft and Credit Card Fraud

Keywords	Relevant Articles #
pin	129
credit card	93
carding	1
credit card sale	0
ssn	0

(c) Results for Vulnerability Discovery and Malicious Code Development

Keywords	Relevant Articles #
vulnerability	418
shellcode	169
polymorphic	12
zero-day	11
cve	2

Table 2. Results from *Cassandra* for Queries

as time goes on. While, other groups only keep influential for a relatively short period of time and just fade out. Figure 6(d) shows the similar phenomenon.

4.2 Evidence Mining by Correlating Social Dynamics with Adversarial Events

We present our finding with keyword queries on the same dataset in *Cassandra*. For each query, *Cassandra* returns the lists of articles, users, and groups in descending order of ArticleRelevance, UserRelevance and GroupRelevance, respectively. The results we present in this section are with regard to three major adversarial activities: i) botnet; ii) identity theft and credit card fraud; and iii) vulnerability analysis and malicious code development.

Botnet As we mentioned before, botnet is a serious threat to all networked computers. In order to identify adversaries and their conversations in our dataset related to botnet, we queried the keywords shown in Table 2(a) in *Cassandra*. *Cassandra* was able to identify 490 articles related to 'spam', 44 articles related to 'botnet', 9 articles related to 'zeus' and 1 article about 'rustock'.

Then, we checked the results returned by *Cassandra* carefully and Table 3 shows several interesting articles and their information including the number of comments they received, ArticleRelevance of each article, and authors of these articles. We first noticed one article titled 'Rustock.C' with very high ArticleRevelance and ArticleInfluence. This article presented an original analysis of the C variant of Rustock that once accounted for 40% of the spam emails in the world.

Translated Article Title	# Comments Received	x_2 [1]	Author
Rustock.C	13	135.3	swx_ur
On startup failure to sign the drivers in Vista x64	5	59.8	crx_ur
video	3	35.6	zlx_ur
sleepy	3	32.3	crx_ur
FireEye Joins Internet2	2	27.8	eax_ur

[1] ArticleRelevance

Table 3. Selected Top Relevant Articles

Another article titled 'On startup failure to sign the drivers in Vista x64' returned by *Cassandra* as a top relevant article to 'botnet' attracting our attention

as well. In this article, the author crx_ur discussed about how to load unsigned driver to Windows Vista x64 by modifying PE file header. The corresponding author claimed that malware vendors would use this technique to build bots and infect thousands of computers. A further investigation on this user shown in Table 4 reveals that s/he authored several security-related articles. Her/his profile indicated that s/he was very active in malicious code development and interested in several cybercrime topics, such as rootkit, exploits, and shellcode.

Translated Article Title	# Comments Received	x_1 [1]	Translated Interests
The old tale about security	7	79.6	malware, ring0, rootkit, botnets, asm, exploits, cyber terrorism, shellcode, viruses, underground, Kaspersky, paintball
Malcode statistics	6	68.9	
Cold boot attacks on encryption keys	2	37.6	
Wanted Cisco security agent	2	28.1	
Antirootkits bypass	1	18.7	
Syser debugger	0	8.9	
Termorektalny cryptanalysis	0	7.8	

[1] ArticleInfluence

Table 4. Selected Articles by crx_ur and Her/His Information

Identity Theft and Credit Card Fraud Identity theft and credit card fraud are both serious issues in Internet transactions. Online identity theft includes stealing usernames, passwords, social security numbers (SSNs), personal identification numbers (PINs), account numbers, and other credentials. Credit card fraud also consists of phishing (a process to steal credit card information), carding (a process to verify whether a stolen credit card is still valid), and selling verified credit card information.

Translated Interests	carding, banking, shells, hacking, freebie, web hack, credit card fraud, security policy, system administrators, live in computer bugs
# Articles Posted	1295
# Comments Posted	7294
# Comments Received	2693

Table 5. Information about dx_ur

Table 2(b) shows results that *Cassandra* returned when these keywords are queried. *Cassandra* identified one article that was authored by a user dx_ur related to 'carding' in the dataset. A further investigation on this user revealed that s/he was a member of a carding interest group, which had more than 20 members around the world. Table 5 shows some basic information of dx_ur. Compared to crx_ur, it is obvious that dx_ur has more interests in financial security issues, such as credit card fraud, web hack, and banking. We could also notice that dx_ur was very active in posting articles and replying others' posts.

Vulnerability Analysis and Malicious Code Development We analyzed several keywords related to vulnerability analysis and malicious code development, such as polymorphism (a technique widely used in malware to change

the appearance of code, but keep the semantics), CVE (a reference-method for publicly-known computer vulnerabilities), shellcode (small piece of code used as the payload in the exploitation of software vulnerabilities), and zero-day (previously-unknown computer vulnerabilities, viruses and other malware).

As shown in Table 2(c), the community is very active in these topics. More than 400 articles related to vulnerabilities were found. However, we noticed most of these articles have low-ArticleInfluence. We checked these low-ArticleInfluence articles and discovered that most of them were articles copied from other research blogs and kept the links to original webpages. Our ArticleInfluence index successfully identified these articles were not very novel, thus calculated low ArticleInfluence for them.

At the same time, as shown in Table 6, *Cassandra* also identified several high-ArticleInfluence vulnerability analysis articles. For example, the article entitled 'Blind spot' authored by arx_ur which analyzed a new Windows Internet Explorer vulnerability even attracted 79 replies.

Translated Article Title	# Comments Received	x_2[1]	Author
Blind spot	79	793.2	arx_ur
Seven thirty-four pm PCR	14	146.4	tix_ur
HeapLib and Shellcode generator under windows	1	15.6	eax_ur
Who fixes vulnerabilities faster, Microsoft or Apple?	0	5.6	bux_ur
FreeBSD OpenSSH Bugfix	0	4.2	sux_ur

[1] ArticleRelevance

Table 6. Selected Top Relevant Articles

4.3 Comparison with HITS algorithm

In order to evaluate the effectiveness of our approach, we implemented the hubs and authorities algorithm (HITS) [20] in *Cassandra* and compared the results with our SocialImpact metrics. HITS algorithm is able to calculate the authorities and hubs in a community by examining the topological structure where *authority* means the nodes that are linked by many others and *hub* means the nodes that point to many others. Note that the fundamental difference between SocialImpact and HITS is that SocialImpact takes more parameters, such as user-generated content and activity, into account, therefore ranking results are based on a more comprehensive set of social features.

Top Five Authorities		Top Five Hubs	
User	auth	User	hub
zhengxx_ur	0.506	zlo_xx_ur	0.265
crx_xx_ur	0.214	zhengxx_ur	0.237
yuz_ur	0.163	crx_xx_ur	0.234
t1mxx_ur	0.148	yuz_ur	0.205
rst_ur	0.143	t1mxx_ur	0.183

Table 7. Top Five Authorities and Hubs by HITS

Comparing the results for authorities and hubs shown in Table 7 with UserInfluence and UserActiveness (SocialImpact) in Table 1, we can observe that the authorities and hubs have much overlap with HITS algorithm when online conversations are ignored and the results generated by SocialImpact are different from HITS counterparts.

5 Related Work

Computer-aided crime analysis (CACA) utilizes the computation and visualization of modern computer to understand the structure and organization of traditional adversarial networks [21]. Although CACA is not designed for the analysis of cybercrime, its methods of relation analysis, and visualization of social network are adopted in our work. Zhou *et al.* [22] studied the organization of United State domestic extremist groups on web by analyzing their hyperlinks. Chau *et al.* [23] mined communities and their relationships in blogs for understanding hate group. Lu *et al.* [24] used four actor centrality measures (degree, betweenness, closeness, and eigenvector) to identify leaders in hacker community. Motoyama *et al.* [29] analyzed six underground forums. In contrast, our proposed solution in this paper considers both social relationships and user-generated contents in identifying interesting posts and users for cybercrime analysis.

Systematically bringing order to a dataset has plenty of applications in both social and computer science. With the development of web, ranking analysis in hyperlinked environment received much attention. Kleinberg [20] proposed HITS by calculating the eigenvectors of certain matrices associated with the link graph. Also, Page and Brin [25] developed PageRank that uses a page's backlinks' sum as its importance index. However, both HITS and PageRank only consider the topological structure of given dataset but ignore its contents [16]. Therefore, we devised a ranking system based on personalized PageRank, which is proposed to efficiently deal with ranking issues in different situations [13].

In order to provide a safer platform for net-centric business and secure the internet experience for end users, huge research efforts have been invested in defeating malware and botnets. Cho *et al.* [26] proposed to infer protocol state machines in botnet C&C protocols. Gu *et al.* analyzed botnet C&C channels for identifying malware infection and botnet organization [27]. Stone-Gross *et al.* [5] took over *Torpig* for a period of ten days and gathered rich and diverse set of data from this infamous botnet. Besides research efforts, legal actions are taken to shutdown certain botnets. *Srizbi* and *Mega-D* botnets were taken down in late 2008 and 2009 [6]. Recently, Microsoft took down *Rustock* by blocking the controller and clearing out the malware infected [28]. Our work focusing on the analysis of malware circulation is complementary to those existing efforts on countering net-centric attacks.

6 Conclusions

In this paper, we have presented a novel approach to help identify adversaries by analyzing social dynamics. We formally modeled online underground social dynamics and proposed SocialImpact as a suite of measures to highlight interesting adversaries, as well as their conversations and groups. The evaluation of our proof-of-concept system on real-world social data has shown the effectiveness of our approach. As part of future work, we would continuosly test the effectiveness and the usability of our system with subject matter experts and broader datasets.

References

1. Anselmi, D.; Kuo, J.; Santhanam, N.; and Boscovich, R., "Microsoft Security Intelligence Report Volume 9."
2. K. Thomas, "The Koobface botnet and the rise of social malware," in *Proc. of the 5th IEEE International Conference on Malicious and Unwanted Software (MALWARE)*, 2010, pp. 1–8.
3. P. Bächer, T. Holz, M. Kötter, and G. Wicherski, "Know your Enemy: Tracking Botnets–Using honeynets to learn more about Bots," 2005.
4. L. L. Chiang, K., "A case study of the rustock rootkit and spam bot," in *Proc. of Usenix Workshop on Hot Topics in Understanding Botnets*, 2007.
5. B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proc. of Computer and Communications Security (CCS)*. ACM, 2009.
6. A. Mushtaq, "Smashing the Mega-d/Ozdok botnet in 24 hours. http://blog.fireeye.com/research/2009/11/smashing-the-ozdok.html."
7. E. Athanasopoulos, A. Makridakis, S. Antonatos, D. Antoniades, S. Ioannidis, K. Anagnostakis, and E. Markatos, "Antisocial networks: Turning a social network into a botnet," in *Proc. of the 11th International Conference on Information Security (ISC)*. Springer, 2008.
8. K. Dunham and J. Melnick, *Malicious bots: an inside look into the cyber-criminal underground of the internet*. Auerbach Pub, 2008.
9. G. W. B. Holt, Thomas J. and A. M. Bossler., "Social Learning and Cyber Deviance: Examining the Importance of a Full Social Learning Model in the Virtual World," *Journal of Crime and Justice*, p. 33, 2010.
10. D. Goodin, "Online crime gangs embrace open source ethos, http://www.theregister.co.uk/2008/01/17/globalization-of-crimeware."
11. E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *Proc. of the 18th International Conference on World Wide Web (WWW)*. ACM, 2009, pp. 531–540.
12. N. Agarwal, H. Liu, L. Tang, and P. Yu, "Identifying the influential bloggers in a community," in *Proc. of the 1st International Conference on Web Search and Web Data Mining (WSDM)*. ACM, 2008.
13. S. Chakrabarti, "Dynamic personalized pagerank in entity-relation graphs," in *Proc. of World Wide Web (WWW)*, 2007.
14. R. Keeney and H. Raiffa, "Decisions with multiple objectives," *Cambridge Books*, 1993.

15. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.
16. M. Bianchini, M. Gori, and F. Scarselli, "Inside pagerank," *ACM Transactions on Internet Technology (TOIT)*, vol. 5, no. 1, pp. 92–128, 2005.
17. F. V. Yarochki, "From Russia with love.exe, http://www.seacure.it/archive/2009/stuff/Seacure2009FyodorYarochkin-FromRussiaWithLove.pdf."
18. E. Raymond, *The new hacker's dictionary.* The MIT press, 1996.
19. R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, pp. 1–39, 2008.
20. J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.
21. J. Xu and H. Chen, "CrimeNet explorer: a framework for criminal network knowledge discovery," *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 2, pp. 201–226, 2005.
22. Y. Zhou, E. Reid, J. Qin, H. Chen, and G. Lai, "US domestic extremist groups on the Web: link and content analysis," *IEEE intelligent systems*, pp. 44–51, 2005.
23. M. Chau and J. Xu, "Mining communities and their relationships in blogs: A study of online hate groups," *International Journal of Human-Computer Studies*, vol. 65, no. 1, pp. 57–70, 2007.
24. Y. Lu, M. Polgar, X. Luo, and Y. Cao, "Social Network Analysis of a Criminal Hacker Community," *Journal of Computer Information Systems*, pp. 31–42, 2010.
25. L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web." 1999.
26. C. Cho *et al.*, "Inference and analysis of formal models of botnet command and control protocols," in *Proc. of the 17th ACM conference on Computer and communications security (CCS)*. ACM, 2010, pp. 426–439.
27. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation," in *Proc. of USENIX Security Symposium*. USENIX Association, 2007.
28. B. Prince, "Microsoft takes down a botnet responsible for 39 percentage of global spam, http://www.pcmag.com/article2/0,2817,2368935,00.asp."
29. Motoyama, M. and McCoy, D. and Levchenko, K. and Savage, S. and Voelker, G.M., "An analysis of underground forums." in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM , 2011.