52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Deployment and Exploitation of Deceptive Honeybots in Social Networks

Quanyan Zhu, Andrew Clark, Radha Poovendran and Tamer Başar

Abstract— **As social networking sites such as Facebook and Twitter are becoming increasingly popular, a growing number of malicious attacks, such as phishing and malware, are exploiting them. Among these attacks, social botnets have sophisticated infrastructure that leverages compromised user accounts, known as *bots*, to automate the creation of new social networking accounts for spamming and malware propagation. Traditional defense mechanisms are often passive and reactive to non-zero-day attacks. In this paper, we adopt a proactive approach for enhancing security in social networks by infiltrating botnets with honeybots. We propose an integrated system named SODEXO which can be interfaced with social networking sites for creating deceptive honeybots and leveraging them for gaining information from botnets. We establish a Stackelberg game framework to capture strategic interactions between honeybots and botnets, and use quantitative methods to understand the tradeoffs of honeybots for their deployment and exploitation in social networks. We design a protection and alert system that integrates both microscopic and macroscopic models of honeybots and optimally determines the security strategies for honeybots. We corroborate the proposed mechanism with extensive simulations and comparisons with passive defenses.**

I. Introduction

Online social networks such as Facebook and Twitter are employed daily by hundreds of millions of users to communicate with acquaintances, follow news events, and exchange information. The growing popularity of OSNs has led to a corresponding increase in spam, phishing, and malware on social networking sites. The fact that a user is likely to click on a web link that appears in a friend's Facebook message or Twitter feed can be leveraged by attackers who compromise or impersonate that individual.

An important class of malware attacks on social networks is social botnets [1], [2]. In a social botnet, an infected user's device and social networking account are both compromised by installed malware. The compromised account is then used to send spam messages to the user's contacts, containing links to websites with the malware executable. As a result, compromising a single well-connected user could lead to hundreds or thousands of additional users being targeted for spam, many of whom will also become members of the botnet and further propagate the malware. The most prominent example of a social botnet to date is Koobface, which at its peak had infected 600,000 hosts [1].

A promising approach to defending against social botnets is through deception mechanisms. In a deceptive defense, the defender generates fake social network profiles that appear similar to real profiles and waits to receive a link to malware. The defender follows the link to the malware site, downloads the malware executable, and runs it in a quarantined environment. By posing as an infected node and interacting with the owner of the botnet, the defender gathers links and reports them to the blacklist, reducing the detection time and increasing the success rate. Currently, however, there is no systematic approach to modeling social botnets and the effectiveness of deception, as well as designing an effective strategy for infiltrating the botnet and gathering information.

In this paper, we introduce an analytical framework for SOcial network Deception and EXploitation through hOneybots (SODEXO). Our contributions can be summarized as follows: (a) developing a game-theoretic model for quantitatively understanding the tradeoffs faced by honyeybots and analyzing their strategic behaviors in order to exploit the botnet and gain information, and (b) creating a system framework to relate the population dynamics of infected nodes and honeybots with microscopic strategic behaviors of honeybots for developing a honeybot deployment mechanism.

We model the exploitation by the honeybots as a Stackelberg game between the botmaster and the honeybots. In the game, the botmaster allocates tasks, such as spam message delivery, among multiple bots based on their trustworthiness and capabilities. The honeybots face a trade-off between obtaining more information by following the commands of the botmaster, and the impact of those commands on other network users. We derive closed forms for the optimal strategies of both the botmaster and honeybots using *Stackelberg equilibrium* as a solution concept. We then incorporate the utility of the honeybot owner under the Stackelberg equilibrium in order to select an optimal deployment strategy.

For the deployment component, we first develop a dynamical model describing the population of a social botnet over time. We derive the relevant steady-state equilibrium of our model and prove its stability. We then formulate the problem of selecting the optimal number of honeybots in order to maximize the information gathered from the botnet as a convex optimization problem. Our results are extended to include networks with heterogeneous node degrees.

The paper is organized as follows. Related work is reviewed in Section II. In Section III, we describe the architecture of our proposed framework for deceptive defense.

The research was partially supported by the AFOSR MURI Grant FA9550-10-1-0573, ARO MURI Grant W911NF-07-1-0287, and also by an NSA Grant through the Information Trust Institute at the University of Illinois.

Q. Zhu and T. Başar are with the Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. Email: {zhu31, basar1}@illinois.edu

Andrew Clark and Radha Poovendran are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA. Email: {awclark, rp3}@uw.edu

978-1-4673-5716-6/13/$31.00 ©2013 IEEE

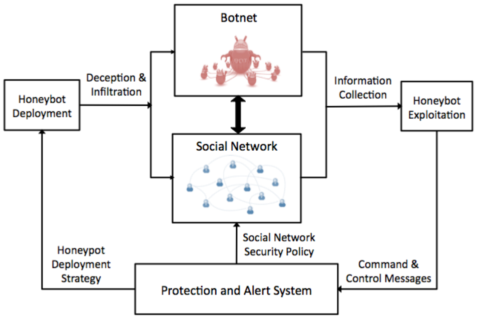
Botnet
Deception & Infiltration
Honeybot Deployment
Information Collection
Honeybot Exploitation
Social Network
Honeypot Deployment Strategy
Social Network Security Policy
Command & Control Messages
Protection and Alert System

Fig. 1. System architecture of honeybot deceptive mechanism in social networks: HD component creates and deploys deceptive honeybots in social network for infiltrating botnets; HE component exploits the honeybot infrastructure and collects information from botnets; PAS designs security policies and coordinates between HD and HE.

In Section IV, we model the exploitation phase of the botnet, in which the honeybot gathers the maximum possible information while avoiding detection by the botmaster. In Section V, we model the deployment and population dynamics of the infected nodes and honeybots. Section VI describes the Protection and Alert System (PAS), which provides a unifying framework for controlling deployment and exploitation. Section VII presents our simulation results. Section VIII concludes the paper.

II. Related Work

Social botnets are a serious threat to network users and managers, as they possess sophisticated infrastructure that leverages compromised users' accounts, known as *bots*, to automate the creation of new social networking accounts for spamming and malware propagation [2]. In [3], a honeypot-based approach is used to uncover spammers in online social networks. In [4], a zombie emulator is used to infiltrate the Koobface botnet to discover the identities of fraudulent and compromised accounts.

Game- and system-theoretic approaches have become pivotal for modeling and designing security mechanisms in a quantitative way [5]. The following works are related to our model. In [6], an optimal control approach to modeling the maximum impact of a malware attack on a communication network was presented. Stochastic models for the propagation of multiple competing malware strains were presented in [7]. In [8], the authors have proposed an architecture for a collaborative intrusion detection network and have adopted a game-theoretic approach for designing a reciprocal incentive compatible resource allocation component of the system against free-rider and insider attacks.

III. System Architecture

In this section, we introduce our honeybot-based defense system named SODEXO for protecting social networks against malicious attacks. A *honeybot* is a fake account on an online social network maintained by a specially quarantined device under the control of a network defender, which is capable of impersonating an infected node. Fig. 1 illustrates the architecture of SODEXO. Our framework consists of two components, namely, honeybot deployment (HD) and honeybot exploitation (HE). The behaviors of the two blocks are coordinated by a Protection and Alert System (PAS), which uses the gathered information to generate real-time signatures and alerts for the social network.

The SODEXO architecture resembles a feedback control system. The HE component behaves as a security sensor of the social network; PAS can be seen as a controller which takes the "measurements" from HE and yields a honeybot deployment strategy; and HD acts as an actuator that updates the honeybot policy designed by PAS. In the following subsections, we discuss each component in detail.

A. Honeybot Deployment (HD)

A honeybot is deployed by first creating an account on a social networking site. The account profile is designed to imitate a real user, as in [3]. Once deployed, the honeybot sends a set of friend requests to a set of randomly chosen other users. The honeybot continues sending friend requests to random users until the desired number of neighbors, denoted d, has been reached. The honeybot monitors the message traffic of its neighbors, which may include personal messages, wall posts, or Twitter feeds, and follows any posted link. If the link points to a site containing a known malware executable for the social botnet and has not been blacklisted, then the honeybot becomes a member of the social botnet and proceeds to the exploitation stage.

B. Honeybot Exploitation (HE)

The HE component of SODEXO takes advantage of the successfully infiltrated honeybots to gain as much information as possible from the botnet. The information is obtained in the form of command and control messages. The honeybots need to gain an appropriate level of trust from the bots and respond to the C&C messages while minimizing harm to the legitimate social network users and avoiding legal liability. Honeybots work collaboratively to achieve this goal. In the case where honeybots are commanded to send spam or malware to network users, they can send them to each other to remain active in the botnet. Depending on the sophistication of the botnet, honeybots can sometimes be detected using mechanisms described in [9]. In this case, a higher growth rate of honeybot population will be needed to replace the detected honeybots. Hence, the performance of HE heavily depends on the effectiveness of HD, and in turn, HD should change its policy based on the sophistication of botnets and the amount of information learned in HE.

C. Protection and Alert System (PAS)

The major role of PAS is to provide security policies for HD based on the information learned from HE. The first step of PAS is to process the messages and logs gained from honeybots. Using data mining and machine learning techniques, it is possible that the structure of botnets can be inferred from network traffic information [10]. This information can be used by the network administrator to detect the location of botmasters and remove them from the network.

The second important task of HD is to generate signatures for detecting malware and spam, which are then used to update the libraries of intrusion detection systems, blacklists of spam filters, and user alerts or recommendations. The process of reconfiguration of IDSs and spam filters can be done either offline or real-time as in [11].

IV. System Model for Honeybot Exploitation

In this section, we introduce a system model for hierarchical botnets and employ a Stackelberg game framework to model the interactions between the botnet and infiltrating honeybots. The proofs in the following sections are omitted due to space constraints and can be bound in [12].

A. Theoretical Framework

Consider a botmaster B that sends requests to a set of C&C bots $\mathscr{M} = \{1,2,\cdots,m\}$ with $m = |\mathscr{M}|$. Each C&C bot $i \in \mathscr{M}$ sends commands to a set of compromised bot nodes $\mathscr{N}_i$ with $n_i = |\mathscr{N}_i|$. We assume that the botnet is a three-level tree architecture and, without loss of generality, we can assume that the sets $\mathscr{N}_i$ are pairwise disjoint, i.e., $\cap_{i\in\mathscr{M}}\mathscr{N}_i = \emptyset$ since a single bot controlled by multiple independent C&C bots can be modeled using multiple duplicate bots. Let H be a honeybot that communicates with node $i \in \mathscr{M}$, i.e., $H \in \mathscr{N}_i$. We assume that all honeybots work together as a team, and hence one honeybot node H under one C&C subtree can conceptually represent a group of collaborative honeybots who have succeeded in infiltrating the same botnet.

We let $p_{ij} \in \mathbb{R}_+$ be the number of messages or commands (in bytes) per second sent from C&C bot i to bot node $j \in \mathscr{N}_i$. Likewise, p_{ji} denotes the number of response messages per second to C&C node $i \in \mathscr{M}$ from node $j \in \mathscr{N}_i$.

Each C&C node i maintains a trust value $T_{ij} \in [0,1]$ associated with a bot or honeybot node $j \in \mathscr{N}_i$. The trust values indicate the quality of response and performance of bot nodes. The trust values also inherently model the detection mechanisms in botnets, which have been discussed in [9]. For botnets with such mechanisms, low trust values indicate the inefficiency of a bot or a high likelihood of being a honeybot. For those without such mechanisms, we can take $T_{ij} = 1$, for all $j \in \mathscr{N}_i$, i.e., equivalently see all bots to be equally trusted.

One C&C bot needs to send commands to a large population of bot nodes. Hence, the goal of C&C bot $i \in \mathscr{M}$ is to allocate its communication resources $\mathbf{p}_i := [p_{ij}, j \in \mathscr{N}_i]$ to maximize the utility of its subtree network $U_i : \mathbb{R}_+^{n_i} \to \mathbb{R}$, which is the sum of utilities obtained from each bot j, i.e.,

$$U_i(\mathbf{p}_i) = \sum_{j\in\mathscr{N}_i} U_{ij}(p_{ij}), \tag{1}$$

where $U_{ij} : \mathbb{R}_+ \to \mathbb{R}$ is the individual utility of C&C bot i from bot $j \in \mathscr{N}_i$, which is chosen to be

$$U_{ij}(p_{ij}) := T_{ij} p_{ji} \ln(\alpha_i p_{ij} + 1). \tag{2}$$

The choice of logarithmic function in (2) indicates that the marginal utility of the C&C bot diminishes as the number of messages increases. It captures the fact that the bots have limited resources to respond to commands, and a larger volume of commands can overwhelm the bots, which leads to diminishing marginal utility of node i. $\alpha_i \in \mathbb{R}_{++}$ is a positive system parameter that determines marginal utility.

The utility of C&C bot i is also proportional to the number of messages or responses per second from bot j, indicated by $p_{ji} \in \mathbb{R}_+$. The number of response messages from bot j indicates the level of activity of a bot. We can see that when $p_{ji} = 0$ or $T_{ij} = 0$ in (2), then bot i is believed to be either inactive or fake, and it is equivalently removed from the subtree of C&C node j in terms of the total utility (1). Note that T_{ij} in (2) evaluates the quality of the responses while p_{ji} evaluates the quantity. The product of T_{ij} and p_{ji} captures the fact that the botnet values highly active and trusted bots.

We consider the following C&C bot optimization problem (BOP) of every node $i \in \mathscr{M}$:

$$\text{(BOP)} \max_{\mathbf{p}_i\in\mathbb{R}_+^{n_i}} \quad U_i := \textstyle\sum_{j\in\mathscr{N}_i} T_{ij} p_{ji} \ln(\alpha_i p_{ij} + 1)$$
$$\text{s.t.} \qquad \textstyle\sum_{j\in\mathscr{N}_i} c_{ij} p_{ij} \le C_i. \tag{3}$$

The constraint (3) in (BOP) is a capacity constraint on the communications using C&C channel, where C_i is the total capacity of the channel. The cost $c_{ij} \in \mathbb{R}_{++}$ is the cost on sending commands to bots. The cost is also dependent on the size of messages from C&C bot i to its controlled bots. It has been found in [4] that Twitter has larger volume of spam messages than Facebook. This is due to the fact that Twitter messages are often shorter than Facebook messages, and hence the cost for commanding bots spamming with Twitter messages is relatively less than the one for Facebook.

Let $\mathscr{F}_i := \{\mathbf{p}_i \in \mathbb{R}_+^{n_i} : \sum_{j\in\mathscr{N}_i} c_{ij} p_{ij} \le C_i\}$ be the feasible set of (BOP). We let $\mathscr{L}_i : \mathbb{R}_+^{n_i} \times \mathbb{R} \to \mathbb{R}$ be the associated Lagrangian defined as follows:

$$\mathscr{L}_i(\mathbf{p}_i, \lambda_i) = \sum_{j\in\mathscr{N}_i} T_{ij} p_{ji} \ln(\alpha_i p_{ij} + 1) + \lambda_i \left(\sum_{j\in\mathscr{N}_i} c_{ij} p_{ij} - C_i \right).$$

Since the feasible set is nonempty and convex, and the objective function is convex in $\mathbf{p}_i$, it is clear that (BOP) is a convex program, and hence we can use the first-order optimality condition to characterize the solution to (BOP):

$$\frac{\partial \mathscr{L}_i}{\partial p_{ij}} = \frac{\alpha_i T_{ij} p_{ji}}{\alpha_i p_{ij} + 1} + \lambda_i c_{ij} = 0, \tag{4}$$

which leads to $p_{ij} = -\frac{T_{ij} p_{ji}}{\lambda_i c_{ij}} - \frac{1}{\alpha_i}$. Due to the monotonicity of logarithmic functions in (2), the optimal solution is found on the Pareto boundary of the feasible set. Hence by letting $\sum_{j\in\mathscr{N}_i} c_{ij} p_{ij} = C_i$, we solve for the Lagrangian multiplier λ_i as follows:

$$\lambda_i = -\frac{\sum_{j\in\mathscr{N}_i} T_{ij} p_{ji}}{C_i + \frac{1}{\alpha_i} \sum_{j\in\mathscr{N}_i} c_{ij}}. \tag{5}$$

We make following assumptions before stating Theorem 1.

(A1) The product $T_{ij} p_{ji} \neq 0$ for all $j \in \mathscr{N}_i, i \in \mathscr{M}$.

Assumption (A1) states that all bots controlled by C&C bot i are both active and trusted by C&C bot i. This assumption is valid because a controlled bot j that is either inactive ($p_{ij} = 0$) or untrusted ($T_{ij} = 0$) can be viewed as the one excluded from the set $\mathscr{N}_i$. Hence Assumption (A0) is equivalent to the statement that $\mathscr{N}_i$ contains all active and trusted bots.

Theorem 1: Under Assumption (A1) and with α_i's sufficiently large, , (BOP) admits a unique non-negative solution:

$$p_{ij} = \left(\frac{T_{ij}p_{ji}}{\sum_{j\in\mathscr{N}_i} T_{ij}p_{ji}}\right)\left(\frac{C_i + \frac{1}{\alpha_i}\sum_{j\in\mathscr{N}_i} c_{ij}}{c_{ij}}\right) - \frac{1}{\alpha_i}. \quad (6)$$

B. Stackelberg Game

The interactions between honeybots and C&C nodes possess an inherent leader-and-follower architecture. The deceptive mechanism of honeybots suggests that they need to follow a C&C protocol and poll information from the botnet [1], [4]. They proactively initiate requests and the bots respond to them. Hence honeybots behave as leaders who can learn the behaviors of C&C bots, and choose optimal strategies to collect the maximum amount of information by responding to the commands from C&C bots subject to cost constraints.

In this section, we formulate a two-stage Stackelberg game between honeybots and C&C nodes. The goal of honeybots is to collect as much information as possible from the botmaster. We consider the following game between honeybots and a C&C bot. The honeybot node H first chooses a response rate p_{Hi} to the commands from C&C bot i, and then C&C bot i observes the response and chooses an optimal rate to send information to honeybot H according to (BOP). We make the following assumption on the real bots in the network.

- **(A2)** The real bots do not strategically interact with the C&C bot i, i.e., they follow a prescribed botnet protocol and send messages to bot i at a rate $p_{ij}, j \neq H, j \in \mathscr{N}_i$.

The above assumption is reasonable because bots are non-human driven, pre-programmed to perform the same routine logic and communications as coordinated by the same botmaster [1], [13]. Under Assumption (A2), strategic interactions exist only between honeybots and C&C nodes.

The honeybot node H has a certain cost when it responds to the botnet. This can be either because of the potential harm that it can cause on the system or due to the cost of implementing commands from the botmaster. We consider the following honeybot optimization problem (HOP), where node H aims to maximize its utility function $U_H : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$ as follows:

$$\text{(HOP)} \max_{p_{Hi}\in\mathscr{F}_H} \quad U_H(p_{iH}, p_{Hi}) := \ln(p_{iH} + \xi_H) - \beta_i^H p_{Hi}, \ (7)$$

where $\xi_H \in \mathbb{R}_{++}$ is a positive system parameter; β_i^H is the cost of honeybot H responding to the bot node i; p_{iH} is the message sending rate from honeybot node H to C&C bot i and p_{Hi} is the rate of C&C bot i sending commands to H.

$\mathscr{F}_H$ denotes the feasible set of the honeybot node H. We let $\mathscr{F}_H = \{p_{Hi}, 0 \leq p_{Hi} \leq p_{Hi,\max}\}$, where $p_{Hi,\max} \in \mathbb{R}_{++}$ is a positive parameter that can be chosen to be sufficiently large. The logarithmic part of the utility function (7) is used to model the property of diminishing returns of an information source. The value of receiving an additional piece of information from the C&C bot decreases as the total number of messages received by the honeybot increases.

The interactions between honeybot H and C&C node i can be captured by the Stackelberg game model $\Xi_S := \langle (i,H), (U_i, U_H), (\mathscr{F}_i, \mathscr{F}_H)\rangle$, and Stackelberg equilibrium can be used as a solution concept to characterize the outcome.

Definition 1 (Stackelberg Equilibrium): Let $\pi_{iH}(\cdot) : \mathbb{R}_+^{n_i} \to \mathbb{R}_+$ be the unique best response of the C&C bots to the response rate p_{Hi} of the honeybots. An action profile $(\mathbf{p}_i^*, \ p_{Hi}^*) \in \mathscr{F}_i \times \mathscr{F}_H$ is a *Stackelberg equilibrium* if $\mathbf{p}_i^* = \pi_{iH}(p_{Hi}^*)$, and for all $p_{Hi} \in \mathscr{F}_H$ the inequality $U_H(\pi_{iH}(p_{Hi}^*), p_{Hi}^*) \geq U_H(\pi_{iH}(p_{Hi}), p_{Hi})$ holds.

Theorem 2: Under Assumption (A1), the nonzero-sum continuous-kernel Stackelberg game Ξ_S admits a Stackelberg equilibrium.

Under Assumption (A1), the unique best response $\pi_{iH}(\cdot)$ can be obtained from (6) for sufficiently large α_i as follows:

$$p_{iH} = \pi_{iH}(p_{Hi}) = C_H\left(\frac{T_{iH}p_{Hi}}{T_{iH}p_{Hi} + I_{-H}}\right) - \frac{1}{\alpha_i}, \quad (8)$$

where $I_{-H} = \sum_{j\neq H, j\in\mathscr{N}_i} T_{ij}p_{ji}$ is the number of responses from real bots weighted by their trust values and $C_H := \frac{C_i + \frac{1}{\alpha_i}\sum_{j\in\mathscr{N}_i} c_{ij}}{c_{iH}}$.

Letting $\xi_H = (1/\alpha_i) + \bar{\xi}_H$ and substituting (8) in (HOP), we arrive at the following optimization problem faced by the honeybot node H:

$$\max_{p_{Hi}\in\mathscr{F}_H} U_H(\pi_{iH}(p_{Hi}), p_{Hi}) :=$$
$$\ln\left(C_H\left(\frac{T_{iH}p_{Hi}}{T_{iH}p_{Hi} + I_{-H}}\right) + \bar{\xi}_H\right) - \beta_i^H p_{Hi}. \quad (9)$$

Theorem 3: Under Assumptions (A1) and (A2), the Stackelberg equilibrium solution $(\mathbf{p}_i^*, \ p_{Hi}^*)$ of the game Ξ_S is unique and can be found as follows:

$$p_{Hi}^* \quad = \quad \frac{C_H I_{-H}}{2T_{iH}(C_H + \xi_H)}\left(\sqrt{1 + 4\frac{T_{iH}(C_H + \bar{\xi}_H)}{I_{-H}C_H\beta_i^H}} - 1\right)$$
$$+ \ \frac{I_{-H}\xi_H}{T_{iH}(C_H + \xi_H)}, \quad (10)$$

and $p_{iH}^* = \pi_{iH}(p_{Hi}^*)$ and $p_{ij}^* = \pi_{ij}(p_{ij})$ for $j \neq H, j \in \mathscr{N}_i$.

In order to provide insight into the solution obtained in (10), we make the following assumptions based on common structures of the botnets.

- **(A3)** The real bots controlled by C&C bot i have identical features, i.e., $c_{ij} = \bar{c}_i$, $p_{ij} = \bar{p}_i$ and $T_{ij} = \bar{T}_i$ for all $j \neq H, j \in \mathscr{N}_i$.
- **(A4)** The size of the real bots controlled by C&C bot i is much larger than the size of honeybots.
- **(A5)** We let $\bar{\xi}_H = 0$.

Assumption (A5) is valid due to the freedom of choosing parameter ξ_H in (HOP). Without loss of generality, we can let $\xi_H = \frac{1}{\alpha_i}$ and hence $\bar{\xi}_H = 0$. Assumption (A3) holds if real bots controlled by C&C bot i are of the same type, for example, Windows non-expert Facebook users. This type of users are commonly the target of botnets. Under (A3), we can simplify the expressions in (10) and obtain $C_H = \frac{C_i}{c_i} + \frac{n_i}{\alpha_i}$, $I_{-H} = n_i^B \bar{T}_i \bar{p}_i$.

Assumption (A4) is built upon the fact that one C&C node in botnets often controls thousands of bots and the size of honeybots are often comparably small due to their implementation costs [14].

Corollary 1: Under Assumptions (A1), (A2) and (A4), the Stackelberg equilibrium solution $(\mathbf{p}_i^*,\ p_{Hi}^*)$ of the game Ξ_S is given by

$$p_{Hi}^* = \left(\frac{1}{\beta_i^H} - \frac{I_{-H}\xi_H}{C_H T_{iH} + T_{iH}\xi_H} \right)^+, \tag{11}$$

where $(\cdot)^+ = \max\{0,\cdot\}$; $p_{iH}^* = \pi_{iH}(p_{Hi}^*)$ and $p_{ij}^* = \pi_{ij}(p_{ij})$ for $j \neq H, j \in \mathscr{N}_i$.

The ensuing result immediately follows from Corollary 1 using (A3) and (A5).

Corollary 2: Let the size of real bots under C&C be n_i^B and the size of the honeybots represented by super node H n_i^H. Note that $n_i = n_i^B + n_i^H$. Under Assumptions (A1) - (A5), the Stackelberg equilibrium of the game Ξ_S is given by

$$p_{Hi}^* = \frac{1}{\beta_i^H}, \quad p_{ij}^* = \pi_{ij}(p_{ij}), \tag{12}$$

for $j \neq H, j \in \mathscr{N}_i$, and the equilibrium solution of C&C node i is composed of two terms given by $p_{iH}^* = p_{iH,S}^* + p_{iH,N}^*$, with the first term independent of n_i^H,

$$p_{iH,S}^* = \frac{T_{iH}}{T_{iH} + \beta_i^H n_i^B \bar{T}_i \bar{p}_i} \left(\frac{C_i}{c_i} + \frac{n_i^B}{\alpha_i} \right) - \frac{1}{\alpha_i}, \tag{13}$$

and the second term dependent on n_i^H,

$$p_{iH,S}^* = \frac{n_i^H T_{iH}}{T_{iH} + \beta_i^H n_i^B \bar{T}_i \bar{p}_i}. \tag{14}$$

Remark 1: From Corollary 2, we can see that under Assumption (A1), the equilibrium response strategy is inversely proportional to the unit cost β_i^H. We can see that the number of command and control messages harvested from the botnet is affine in the number of successfully infiltrated honeybots. The growth rate of the number of messages is given by

$$r_{iH}^* := \frac{\partial p_{iH}^*}{\partial n_i^H} = \frac{T_{iH}}{\beta_i^H n_i^B \bar{T}_i \bar{p}_i + T_{iH}}. \tag{15}$$

The growth rate is dependent of the trust value T_{iH}. Honeybots can harvest more information from the botnet if they are more trusted. The growth rate is also dependent on the number of the real bots controlled by C&C bot i. As $n_i^B \to \infty$, the growth rate $r_{iH}^* \to 0$, i.e., size of honeybots will not affect the number of messages received by the network.

V. Model of Honeybot Deployment and Botnet Growth

In what follows, a macroscopic model of the dynamics of the number of bots at time t, denoted $x_1(t)$, and the number of honeybots, denoted $x_2(t)$, is presented. We present our model for both idealized networks where all nodes have the same degree, as well as networks with heterogeneous degrees.

A. Botnet and honeybot growth models

The bots are assumed to send spam messages, containing links to malware, with rate r. Each message is sent to the d neighbors of the bot, where d is the average node degree. Hence in each time interval dt, $rd\ dt$ spam messages are sent. Since the number of valid nodes is $N - x_1(t)$, the number of messages reaching valid nodes is equal to $rd\frac{N-x_1(t)}{N}\ dt$.

The number of nodes that become bots depends on the valid users' behavior and the number of links that have been blacklisted. Each user clicks on a spam link with probability q. If the link has been blacklisted, then the user will be blocked from visiting the infected site; otherwise, the user's account is compromised and becomes part of the botnet.

To determine the probability that a link has been blacklisted, we assume that each bot is independently given a set of k malicious links, out of M links total. The probability that a link has been given to a specific honeybot is therefore $\frac{k}{M}$. Hence the probability that a link has not been blacklisted is the probability that that link has not been given to any honeybot, which is equal to $\left(1 - \frac{k}{M}\right)^{x_2}$. We assume that:

(A6) The number of links given to each honeybot, k, satisfies $k \ll M$.

Under (A6), $\left(1 - \frac{k}{M}\right)^{x_2}$ can be approximated by $\left(1 - \frac{kx_2}{M}\right)$.

Finally, we assume that the infected devices are discovered and cleaned with rate μ_1. This leads to dynamics

$$\dot{x}_1(t) = rdqx_1 \left(1 - \frac{kx_2}{M}\right) \frac{N - x_1}{N} - \mu_1 x_1. \tag{16}$$

Honeybot nodes are inducted into the botnet in a similar fashion. We make the following assumptions regarding the honeybot population:

(A7) The number of honeybots that are not part of the botnet, denoted z, is constant.

(A8) The number of honeybots is small compared to the total number of users, so that $\frac{z}{z+N} \approx \frac{z}{N}$.

Assumption (A7) can be guaranteed by creating new, uninfected honeybots when existing honeybots infiltrate the botnet. Since blacklisted links are automatically deleted by the social network owner [4], honeybot nodes cannot follow such links; however, unlike real users, honeybot nodes will attempt to follow any non-blacklisted link with probability 1. The botmaster detects and removes honeybots with rate μ_2. The honeybot population is therefore defined by

$$\dot{x}_2(t) = rdx_1 \left(1 - \frac{kx_2}{M}\right) \frac{z}{N} - \mu_2 x_2. \tag{17}$$

Proposition 1: Under assumptions (A6)–(A8), the dynamics defined by (16) and (17) have two equilibria, given by $(x_1, x_2) = (0, 0)$ and

$$x_1^* = \frac{N\mu_2 M(rdq - \mu_1)}{rdq\mu_2 M + rdkz\mu_1}, \quad x_2^* = \frac{\left(rd - \frac{\mu_1}{q}\right) z}{\frac{rdkz}{M} + \mu_2}. \tag{18}$$

The quantity rdq corresponds to the rate at which new nodes are inducted into the botnet, while μ_1 is the rate at which nodes are cleaned and exit the botnet. Thus if $rdq < \mu_1$, then the number of bots converges to zero, while $rdq > \mu_1$ implies that the number of bots converges to a nonzero steady-state value.

Since network security policies are typically updated intermittently, while the dynamics of (16) and (17) converge rapidly, we base our subsequent analysis on the steady-state values of x_1 and x_2, and derive the optimal number of honeybots to introduce into the system in steady-state. In order to prove that this problem is well defined, we first examine the stability properties of each equilibrium in the following theorem.

Theorem 4: Assume that (A6)–(A8) hold. If $\mu_1 > rdq$, then $(x_1, x_2) = (0, 0)$ is asymptotically stable. If $\mu_1 < rdq$, then (x_1^*, x_2^*) is asymptotically stable in the limit as $M \to \infty$, $N \to \infty$.

B. Computation of system parameters

The parameter μ_2 determines the rate at which honeybot nodes are discovered and removed by the botmaster, and hence can be calculated by observing the lifetime of deployed honeybots (see Section VI). Similarly, the number of received messages p and the cost τ can be estimated by averaging over the set of deployed honeybots over time. The fraction of malicious links $\frac{k}{M}$ that are given to a single bot or honeybot is estimated by using the assumption that links are distributed independently and uniformly at random by the botmaster, so that the probability that a given link has been received by a honeybot is $\left(1 - \frac{k}{M}\right)^{x_2}$. This probability can be estimated by analyzing the set of malicious links received by new honeybots, which combined with knowledge of x_2 enables computation of $\frac{k}{M}$. The rate at which spam messages are sent by bots, denoted r, is estimated by the number of instruction messages received by the honeybots.

The parameters μ_1 and q, equal to the rate at which bots are removed from the botnet, and the fraction of malicious links that are followed by users, depend on user behavior. These parameters can be estimated using data sets of user behavior [15]. Furthermore, to obtain an upper bound on the effectiveness of the botnet, the parameter q can be set equal to 1, implying that a valid user always clicks any link to the malware executable (the worst case). The average node degree, d, is estimated based on existing analyses of the degree distribution of social networks [16].

VI. Modeling of Protection and Alert System

PAS is a coordination system that strategically deploys honeybots and designs security policies for social networks. In this section, we focus on optimal reconfiguration of honeybots. We introduce a mathematical framework for finding honeybot deployment strategies based on system models described in Sections IV and V.

A. Relations between HD and HE

We have adopted a divide-and-conquer approach in Sections IV and V, and have modeled the behavior of each system independently. However, the interdependencies between HD and HE are essential for PAS to make optimal security policies for the social network. The HE model in Section IV describes strategic operations of honeybots at a microscopic level while the HD model in Section V provides a macroscopic description of the population dynamics of bots and honeybots. These two models are interrelated through their parameters together with the feedback from PAS.

The interactions between bots and honeybots in the HE model occur on a time scale of seconds. The analysis of Stackelberg equilibrium in Section IV captures the steady-state equilibrium after a repeated or learning process of the game. Hence the equilibrium can be reached on a time scale of minutes. On the other hand, the population dynamics in HD model evolve on a larger time scale (for example, days). Hence, we can assume that the Stackelberg game has reached its equilibrium when the populations evolve at a macroscopic level. Decisions made at PAS are on a longer time scale (for example, weeks) because the processing of collected information, learning of bots and honeybots in social networks, and high-level decision on security policy in reality demand considerable amount of human resources for coordination and supervision.

1) Trust Values and Detection Rate: The trust values T_{ij} used in HE model are related to the macroscopic detection and removal rate μ_2 in HD model. As we have pointed out earlier, zero trust values are equivalent to the removal of honeybots from the botnet. Hence we can adopt a simple dynamic model to describe the change of T_{ij} over a longer time period (say, days). We let T_{ij}^0 be the initial condition of the trust value. The evolution of T_{ij} over the macroscopic time scale can be modeled using the following ODE:

$$T_{ij}(t) = -\mu_2 T_{ij}(t), \quad T_{ij}(t_{ij}^0) = T_{ij}^0. \tag{19}$$

Note that honeybots have different initial times t_{ij}^0. Hence from (19), we obtain

$$T_{ij}(t) = T_{ij}^0 e^{-\mu_2 (t - t_{ij}^0)}, \quad t \geq t_{ij}^0, \tag{20}$$

i.e., the trust values exponentially decay with respect to the removal rate. From (20), we can obtain the mean life time of a honeybot to be $1/\mu_2$. Macroscopic parameter μ_2 can be estimated by the rate of change of working honeybots in the botnet, which is known to the system, while T_{ij} is a microscopic parameter and is often unknown directly to honeybots. With the ODE model in (19), we can use μ_2 to estimate T_{ij}.

2) Honeybot and Bot Populations: In Section V, the populations of bots and infiltrating honeybots are denoted by x_1 and x_2, respectively, whereas in Section IV, the bot size under C&C bot is n_i^B. Under a hierarchical structure of botnet, the total bot and honeybot populations x_1, x_2 are $x_1 = \sum_{i=1}^m n_i^B$ and $x_2 = \sum_{i=1}^m n_i^H$. If all C&C bots are identical, i.e., $n_i^B = \bar{n}^B, i \in \mathscr{M}, n_i^H = \bar{n}^H, i \in \mathscr{M}$; then $x_i = m\bar{n}^B$, and $x_i = m\bar{n}^H$, $i = 1, 2$.

3) Activity Level of Bots: The rate $\bar{p}_i$ in (15) indicates the activity level of bots when they respond to or poll information from C&C node i. This level of activity is often correlated with parameter r, the rate of sending out spamming messages to the social network. Assume that all C&C bots are assumed to be identical, i.e., $\bar{p}_i = \bar{p}, i \in \mathscr{M}$, then we can let $\bar{p} = \eta r$, where $\bar{p}$ is in messages/sec, r is in messages/sec and $\eta \in \mathbb{R}_{++}$ is a unitless positive parameter.

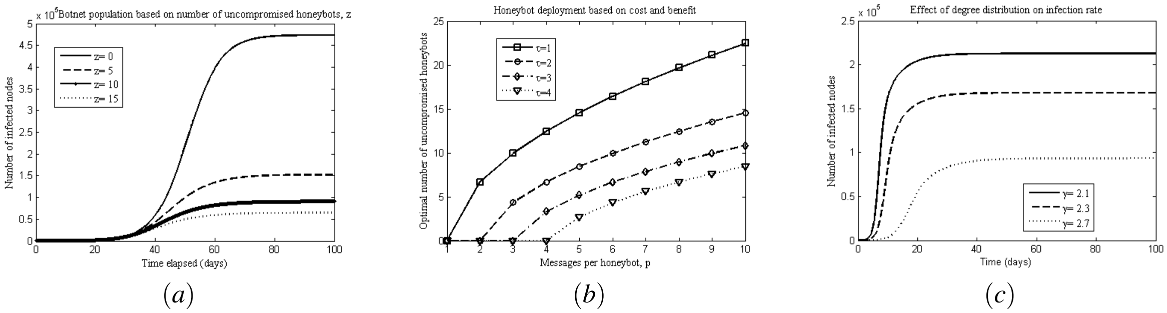

Fig. 2. Simulation of our framework for a network of $N = 10^6$ users, where each user has probability $q = 0.01$ of following a malicious link, messages are sent at a rate of 0.4 messages per bot per day, and infected nodes are cleaned after 5 days on average. (a) Effect of increasing the number of honeybot nodes on the botnet population. Deployment of a small number of honeybots can greatly reduce the number of bots present. (b) The optimum number of bots based on (22) for different costs τ and benefits ρ. The total number of honeybots remains small for each case. (c) Effect of degree distribution on the botnet population for number of honeybots $z = 5$. Each network is scale-free, with exponent γ varying between networks.

B. Optimal Honeybot Deployment

In what follows, we first derive the optimal honeybot deployment for the homogeneous degree model when the benefit from each honeybot is measurable. We then combine the analysis of Sections IV and V to determine the optimal honeybot deployment, taking into account the behavior of the honeybots during the exploitation phase.

The goal of the honeybot operator is to maximize the number of blacklisted links that are reported to the social network. Based on the analysis of Corollary 2, we assume that the number of blacklisted links is proportional to the number of honeybot nodes in the botnet in steady-state, $\mathbf{x}_2^*$. The variable is the number of honeybot nodes that have not yet been inducted into the botnet, z. This leads to a utility function given by $V_H(z) = \rho x_2^*(z) - \tau(x_2^* + z)$, where ρ and τ represent the benefit (information gathered) and cost of maintaining a single honeybot node. Substituting (18) yields

$$\begin{aligned} V_H(z) &= \rho \frac{\left(rd - \frac{\mu_1}{q}\right) z}{\frac{rdkz}{M} + \mu_2} - \tau \left(\frac{\left(rd - \frac{\mu_1}{q}\right) z}{\frac{rdkz}{M} + \mu_2} + z \right) \\ &= \frac{(\rho - \tau)\left(rd - \frac{\mu_1}{q}\right) z}{\frac{rdkz}{M} + \mu_2} - \tau z. \end{aligned} \tag{21}$$

The value of z that maximizes (21) is given by the following proposition.

Proposition 2: Assuming (A6)–(A8) and $\rho > \tau$, the optimum value of z that maximizes (21) is given by

$$z^* = M \left(\frac{-\mu_2 + \sqrt{(\rho - \tau)(rd - \frac{\mu_1}{q})\mu_2 / \tau}}{rdk} \right). \tag{22}$$

Remark 2: Eq. (22) has several implications for the design of honeybot systems. First, for malware that propagates rapidly (corresponding to a large rd value), fewer honeybots are needed, since the malware will quickly spread to the deployed honeybot. Second, if μ_2 is large, then honeybots are rapidly detected and removed by the botmaster, and hence the cost of deploying honeybots outweighs the benefits.

C. Optimal Deployment and Exploitation

The utility function (21) can be augmented by incorporating the impact on the exploitation phase. In particular, (15) implies that $\rho = \frac{1}{1 + \frac{\beta_i^H x_1^* \overline{T}_i r}{T_{iH}}}$, which we write as $\rho = \frac{1}{1 + \zeta x_1^*} \approx \frac{1}{\zeta x_1^*}$ when the number of bots is sufficiently large. The utility function V_H can then be written as

$$V_H = \left(\frac{1}{\zeta x_1^*} - \tau \right) x_2^* - \tau z \tag{23}$$

An efficient algorithm for maximizing (23) can be derived using the following theorem.

Theorem 5: Under assumptions (A6)–(A8), the problem of selecting z to maximize V_H in (23) is equivalent to the following convex program

$$\begin{aligned} \max_{\theta, \phi, x_2^*, z} \frac{1}{\zeta} & \left(-\frac{rdq\theta^2}{Nrdq\phi - \mu_1} + \frac{M/4k}{N(rdq\left(1 - \frac{kx_2^*}{M}\right) - \mu_1)} \right) \\ & -\tau x_2^* - \tau z, \end{aligned} \tag{24}$$

$$\text{s.t.} \qquad \theta = x_2^* - \frac{M}{2k}, \qquad \phi = 1 - \frac{kx_2^*}{M},$$

$$x_2^* \leq \frac{\left(rd - \frac{\mu_1}{q}\right) z}{\frac{rdkz}{M} + \mu_2}, \quad \frac{1}{\zeta} \frac{rdq\mu_2 M + rdkz\mu_1}{(rdq - \mu_1)\mu_2 M} \geq \tau, \tag{25}$$

$$z \geq 0, \quad 0 \leq x_2^* \leq N. \tag{26}$$

The convex optimization approach presented in Theorem 5 is used to select a honeybot deployment strategy in order to maximize the level of infiltration into the botnet and the amount of data gathered during the exploitation phase. Once inducted into the botnet, the honeybots follow the Stackelberg equilibrium strategy of Section IV and use the collected data to generate malware signatures and create URL blacklists. The parameters of (23) are updated in response to changes in botnet behavior observed during the exploitation phase.

VII. Simulation Study

We evaluated our proposed method using Matlab simulation study, described as follows. A network consisting of $N = 10^6$ nodes was generated, with degree $d = 100$ (consistent with observations of the average degree of social networks [16]). The rate at which malware messages are sent is given by $r = 0.4$ messages per bot day, and the rate at which nodes are disinfected and removed from the botnet is $\mu_1 = 0.2$, yielding an average lifetime for each bot of 5 days. These statistics are based on the empirical observations of [4]. Based on [15], we estimate that the probability of a user clicking on a spam link is given by $q = 0.01$. It is assumed that the fraction of malware links given to each bot is equal to $k/M = 0.01$. The rate at which honeybots are detected and removed is equal to $\mu_2 = 0.5$. In each case, we assume that there are 50 infected nodes and 0 honeybots present in the network initially.

The population dynamics of the bots, described by (16) and (17), are shown in Fig. 2(a). Each curve represents the number of infected users over time for a different level of honeybot activity, as described by the parameter z. In each case, the number of bots converges to its equilibrium value. The top curve (solid line) assumes $z = 0$, i.e. no deception takes place and malicious links are detected through blacklists only. Employing deception through honeybots significantly reduces the botnet population, even when the number of honeybots is small relative to the population size. As additional honeybots are added, the botnet population continues to decline. However, the marginal benefit of adding a honeybot decreases as the number of honeybots grows.

The optimum number of honeybots depends on the cost of introducing and maintaining honeybots, denoted τ, as well as the benefit ρ from each honeybot, as described in (22). The optimum number of honeybots is given in Fig. 2(b). As the cost of introducing new honeybots is reduced, the optimal number of honeybots increases. In each case, the optimum number of honeybots remains small, at around 25 nodes, relative to the total network population of 10^6 nodes.

The effect of a heterogeneous degree distribution is shown in Fig. 2(c). The degree distribution was chosen to be scale-free, so that the probability that a node has degree d was proportional to $d^{-\gamma}$. Hence a higher value of γ corresponds to a less-connected network. The parameter γ had a significant impact on the rate of propagation of the botnet, even through for the chosen values of γ the average degrees of the three networks were similar.

VIII. Conclusion

In this paper, we studied deception-based defenses against social botnets. We considered a defense mechanism in which fake honeybot accounts are deployed and infiltrate the botnet, impersonating infected users. The infiltrating honeybots gather information from command and control messages, which are used to form malware signatures or add spam links to URL blacklists. We introduced a framework for SOcial network Deception and EXploitation through hOneybots (SODEXO), which provides an analytical approach to modeling and designing social honeybot defenses. We decomposed SODEXO into deployment and exploitation components.

In the deployment component, we modeled the population dynamics of the infected users and honeybots, and showed how the infected population is affected by the number of honeybots introduced. We derived the steady-state populations of infected users and honeybots and proved the stability of the equilibrium point. In the exploitation component, we formulated a Stackelberg game between the botmaster and the honeybots, and determined the amount of information gathered by the honeybot in equilibrium. The two components were combined in the Protection and Alert System (PAS), which chose an optimal deployment strategy based on the information gathered by the honeybots. We presented simulation studies supporting our results, which show that a small number of honeybots significantly decrease the infected population of a large social network.

References

[1] J. Baltazar, J. Costoya, and R. Flores, "The real face of koobface: The largest web 2.0 botnet explained," *Trend Micro Research*, vol. 5, no. 9, p. 10, 2009.

[2] G. Keizer, "Worm spreads on facebook, hijacks users' clicks," *Computerworld*, December 2008.

[3] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots + machine learning," in *Proc. of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 435–442, 2010.

[4] K. Thomas and D. Nicol, "The Koobface botnet and the rise of social malware," in *5th International Conference on Malicious and Unwanted Software (MALWARE)*, 2010.

[5] M. Manshaei, Q. Zhu, T. Alpcan, T. Başar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Survey*, vol. 45, pp. 25:1 – 25:39, June 2013.

[6] M. H. R. Khouzani, S. Sarkar, and E. Altman, "Maximum damage malware attack in mobile wireless networks," in *Proc. of the 29th Conference on Information Communications (INFOCOM'10), San Diego, California, USA*, 2010.

[7] S. Xu, W. Lu, and Z. Zhan, "A stochastic model of multivirus dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 30 –45, 2012.

[8] Q. Zhu, C. Fung, R. Boutaba, and T. Başar, "A game-theoretic approach to knowledge sharing in distributed collaborative intrusion detection networks: Fairness, incentives and security," in *Proc. of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 243–250, 2011.

[9] P. Wang, L. Wu, R. Cunningham, and C. C. Zou, "Honeypot detection in advanced botnet attacks," *International Journal of Information and Computer Security*, vol. 4, no. 1, pp. 30–51, 2010.

[10] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic," in *Proc. of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, February 2008.

[11] Q. Zhu and T. Başar, "Dynamic policy-based IDS configuration," in *Proc. of the 48th IEEE Conference on Decision and Control (CDC)*, pp. 8600 –8605, 2009.

[12] Q. Zhu, A. Clark, R. Poovendran, and T. Başar, "SODEXO: A system framework for deployment and exploitation of deceptive honeybots in social networks," in *Arxiv preprint 1207.5844*, 2012.

[13] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. of the 17th USENIX Security Symposium (Security'08)*, 2008.

[14] N. Provos and T. Holz, *Virtual honeypots: from botnet tracking to intrusion detection*. Addison-Wesley Professional, first ed., 2007.

[15] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: the underground on 140 characters or less," in *Proc. of the 17th ACM Conference on Computer and Communications Security*, pp. 27–37, 2010.

[16] M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou, "Practical recommendations on crawling online social networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 29, pp. 1872–1892, October 2011.