A First Step Towards Characterizing Stealthy Botnets

Justin Leonard, Shouhuai Xu and Ravi Sandhu
Department of Computer Science, University of Texas at San Antonio
{jleonard,shxu}@cs.utsa.edu, ravi.sandhu@utsa.edu

Abstract

Botnets have become a top cyber threat. Existing studies on botnets have mainly focused on showing how to exploit certain characteristics of existing botnets to detect them. However, such detection mechanisms could be defeated by stealthy botnets that are designed to evade them. Therefore, it is important to understand the power of stealthy botnets so as to answer questions such as: What kinds of stealth techniques can survive what kinds of detection mechanisms? Towards the ultimate goal, this paper makes a first step with the aim to build fundamental understandings of stealthy botnet Command and Control (C&C).

Keywords: Stealthy botnets, botnet stealth management, botnet modeling, botnet characteristics, botnet C&C

1. Introduction

A bot is a compromised computer that can carry out the commands of its master; a botnet is a network of bots. Botnets have become a significant and growing threat on the Internet [2], [3], [5], [15], [16], [24], [21]. The botnet research community has vigorously pursued various *ad hoc* detection technologies that may deal with the variations in attack behaviors (DDoS, port scan, remote exploits, phishing, spam, spyware, identity theft), the variations in Command and Control (C&C) topologies (centralized, distributed, P2P, random scan), the variations in rally mechanisms (hard-coded IP, dynamic DNS, distributed DNS, Fast Flux), and the variations in communication protocols (Internet Relay Chat (IRC), HTTP, Instant Messaging (IM), Peer-to-Peer (P2P)) [14], [3], [11]. However, such detection mechanisms could be defeated by stealthy botnets that are designed to evade them. Thus, it is important to understand and characterize stealthy botnets.

Our contributions. We propose a graph-based model for botnet C&C mechanisms. The model captures an important aspect of C&C mechanisms — who knows whom and who can send or forward C&C messages to whom. The model also accommodates that the botnet topology itself might have already "embedded" some craftiness of the botnet master (i.e., botnet topologies are carefully chosen by the masters). Moreover, the model can accommodate the master's attack power or sophistication as well as the defender's detection capability.

The model considers two botnet stealth measures called *detection ratio* and *resilience*. The former aims to capture the detection of bots due to the conducting of C&C activities, and the latter aims to capture the tracing of bots based on botnet topology. Simulation study allows us to draw useful insights on factors contributing to botnet stealth, such as the following. First, in order to achieve the same degree of security, countering a more sophisticated attack requires a corresponding improvement in the defender's detection capability. Second, in- and out-degree regular graphs, in which each vertex has the same in-degree as well as out-degree, as botnet topology exhibit best observed stealth. In particular, such graphs exhibit an "all or nothing" detection effect, meaning (1) either all or no bots are detected and (2) the defender cannot benefit from tracing bots according to the botnet topology. Moreover, a botnet master who is able to maintain such a botnet topology does not, in contrast to a recent rule of thumb, necessarily gain stealth by splitting a large botnet into smaller ones.

Related work. The most closely related work is perhaps Dagon et al. [4], who investigate botnet *connectivity* while bots may be destroyed, but not necessarily stealth of botnet C&C. Dagon et al. [5] provide a model for capturing the propagation of botnets, but not necessarily C&C activities.

Other studies on botnets have mainly focused on exploiting certain characteristics to detect botnets. There are mainly two approaches.

- The first approach, either host-based or network-based, aims to detect botnets *without infiltrating* them. Host-based detection includes signature-based IRC botnet detection systems such as [7], and behavior-based bot detection systems such as [18], which focuses on the way bots respond to data received over the network. Network-based detection aims to detect and possibly track botnets based on, for example, DNS lookup information [15], [16], flow data across large ISP [12] or local networks [20], [19], [13], network-level conversations within centralized botnets as visible from sampled traffic flows [17], email traces whereby the defender can map botnet membership by looking for multiple bots participating in the same spam email campaign [23], the correlation of IDS-driven dialog according to a user-defined "vertical" bot infection model (accommodating inbound scanning, exploit usage, egg downloading, outbound bot coordination dialog, and

outbound attack propagation) [9], the correlation of "horizontal" (spatial-temporal) network anomalies of the hosts within a network (e.g., based on the observation that the bots should exhibit the same network behaviors) [10], [8], or the aggregation of centralized C&C traffic [22].

- The second approach aims to capture and analyze bot samples and then *infiltrate* into botnets. This approach has successfully tracked IRC-based botnets [6], [1], [9], and has very recently been extended to deal with a class of P2P-based botnets (which use *unauthenticated* content-based publish/subscribe communications for C&C). In general, this approach consists of three steps: (1) Capture and analyze a bot so as to extract information such as IP addresses of initial peers, service ports, and application-specific connection information. (2) Infiltrate into the botnet so as to receive botnet commands and even identify, for example, the central IRC server. (3) Mitigate botnets by, for example, taking IRC server offline.

While these mechanisms have been effective in countering past and (possibly) current botnets, future stealthy botnets, which are devised while bearing the defenses in mind, could deploy strategies such as extreme delays to render the defenses ineffective. Therefore, it is important to conceive a general framework to understand stealthy botnets while accommodating defenses that could detect, trace and remove portions of bots. We believe that this paper presents a first step towards this ultimate goal.

Outline of the paper. The rest of the paper is organized as follows. In Section 2, graph-based botnet C&C model is specified. Section 3 reports our simulation study. We conclude the paper in Section 4 with notes on future work.

2. A C&C Model and Stealth Measures

Model. We model a botnet as a *directed* graph $G = (V, E)$, where V is the bot set and E is the arc set. The rationale for adopting directed graphs (rather than undirected graphs) is that a crafty master would mitigate the damage caused by the detection of bots, which can be naturally accommodated by directed graphs such that detection of one bot may only cause the detection of its descendant. Moreover, directed graphs allow us to accommodate anonymous channels that may be used for conducting botnet C&C. As usual, the degree of a bot u, denoted by $\deg(u)$, in a directed graph is the sum of its in-degree and out-degree.

There are two possible definitions of the arc set E. The first is based on the "knows" relation, namely that $(u, v) \in E$ if and only if bot u knows, for example, the IP address or pseudonym of bot v. The second is based on the "communication" relation, namely that $(u, v) \in E$ if and only if bot u sends some C&C message to bot v. These two relations can be different; for simplicity, we assume that the two relations are the same, meaning that C&C messages are sent in a (pruned) flooding fashion according to E.

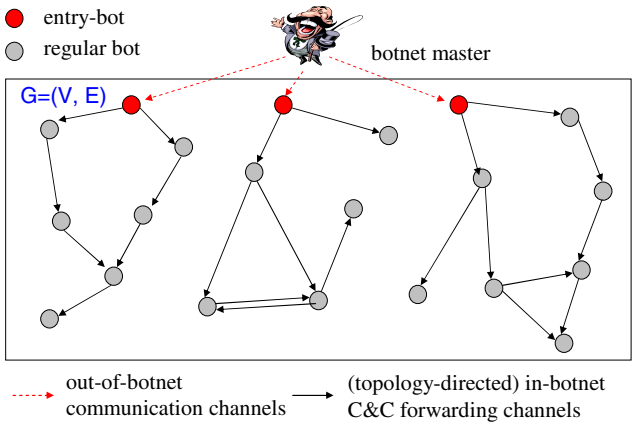


Figure 1. An illustrative botnet example

Figure 1 shows an illustrative botnet topology. We assume the master $\mathcal{A}$ of a botnet $G = (V, E)$ does not belong to V, instead $\mathcal{A}$ may only communicate with some "entry bots," which forward C&C messages through the botnet topology (i.e., each bot forwards C&C messages along its out-going arcs). Note that letting $\mathcal{A} \notin V$ is not really a restriction because the botnet master might often use some out-of-botnet (anonymous) channels (i.e., the red-color dashed arcs) to send C&C messages. Moreover, it is straightforward to extend our model to accommodate $\mathcal{A} \in V$. Note also that a C&C mechanism can be captured by a set of messages $\{m_{(u,v)}\}_{(u,v)\in E}$ sent through E.

Model parameters. We consider the following parameters: *attack sophistication* that is captured by $(\alpha_{(u,v)}, \beta_{(u,v)})$ for $(u, v) \in E$, and *detection threshold* $k \in [0, 1]$. For any $(u, v) \in E$, we define $\alpha_{(u,v)}$ to be the probability that u is exposed because of sending a message to v, and define $\beta_{(u,v)}$ to be the probability that v is exposed because of receiving a message from u. Note that for $(u, v) \in E$, "u uses a sender-anonymous channel to send a C&C message to v" effectively means $\alpha_{(u,v)} = 0$ or $\alpha_{(u,v)} \approx 0$, and "u uses a receiver-anonymous channel to send a C&C message to v" effectively means $\beta_{(u,v)} = 0$ or $\beta_{(u,v)} \approx 0$, dependent upon the anonymity assurance of the anonymous channels. For any $(u, v) \notin E$, we define $\alpha_{(u,v)} = 0$ and $\beta_{(u,v)} = 0$ because no C&C messages will be sent from u to v. We assume that bot exposure events, which are caused by the sending/receiving of C&C messages, are independent.

The detection threshold k captures the master's estimation of the defender's capabilities for detecting bots. A smaller k means a better detection capability. A crafty botnet master may have its own risk management system such that a bot may be abandoned once its exposure is about to exceed the detection threshold k so as to prevent the bot from being detected.

Botnet stealth measures. We propose measuring botnet stealth through the following attributes: *detection ratio* and *resilience*. Let $\mathbb{V}$ be the set of all possible bots, $\mathbb{C}$ the set of all possible C&C mechanisms, $\mathbb{N}$ the set of positive integers, and $\mathbb{G}$ be the set of all possible botnet topologies.

Definition 1: (Detection ratio) Let $\mathcal{E}'_u$ be the probability that a bot $u \in V$ is already exposed before conducting any further C&C activities, $\mathcal{E}_u(C)$ be the probability that a bot $u \in V$ is exposed to the defender due to the conducting of some C&C activities. Then, the probability $\mathcal{E}_u(C)$ that u is exposed after sending $m_{(u,v)}$ C&C messages over $(u,v) \in E$ and receiving $m_{(w,u)}$ C&C messages over $(w,u) \in E$ is

$$\begin{aligned}\mathcal{E}_u(C) &= 1-(1-\mathcal{E}'_u)\cdot \prod_{(u,v)\in E}(1-\alpha_{(u,v)})^{m_{(u,v)}} \\ &\quad \cdot \prod_{(w,u)\in E}(1-\beta_{(w,u)})^{m_{(w,u)}}.\end{aligned}$$

We define the *detection ratio* as $|V'|/|V|$, where $V' = \{u : \mathcal{E}_u(C) > k\}$ for some detection threshold k. Note that the use of anonymous channels, e.g., $\alpha_{(u,v)} = 0$ or $\beta_{(u,v)} = 0$, will affect detection ratio.

To define resilience, we need some notations so as to accommodate the possible use of anonymous channels. Given $G = (V,E) \in \mathbb{G}$, a botnet topology known only to the master, we note that $(u,v) \in E$ does not necessarily mean that the detection of u will lead the defender to trace to v because, for example, u may use some receiver-anonymous channel to send C&C messages to v (e.g., u may, without knowing v's IP address, broadcast an encrypted message that can only be decrypted by v). We say v is traceable from u if and only if $(u,v) \in E$ and $\beta_{(u,v)} > 0$ (meaning that the channel for u to send C&C messages to v is not receiver-anonymous). Note that the "traceable" relation effectively imposes a new topology $\bar{G} = (V, \bar{E})$, where $(u,v) \in \bar{E}$ if and only if $(u,v) \in E$ and $\beta_{(u,v)} > 0$.

Definition 2: (Resilience) Suppose $G = (V,E)$ is a botnet, and $\bar{G} = (V,\bar{E})$ be the topology after imposing the "traceable" relation to G. Let V' be defined as above. The resilience measure, denoted by $\mathcal{R}(\cdot,\cdot) : \mathbb{G} \times \mathbb{V} \to [0,1]$, can be defined as

$$\mathcal{R}(G,V') = 1 - \frac{|V' \cup \left(\cup_{u\in V'}\{v : u \rightsquigarrow v \in \bar{E}\}\right)|}{|V|},$$

where "$u \rightsquigarrow v \in \bar{E}$" if there is a path (consisting of one or more directed arcs) from u to v in $\bar{G}$. Note that resilience captures the consequence after a subset $V' \subseteq V$ of bots are detected. Note that the use of receiver-anonymous channels, i.e., $\beta_{(u,v)} = 0$, affects botnet resilience.

3. Simulation Study

Our model suggests that large-degree bots would be relatively easily detected, unless anonymous channels are extensively utilized. Therefore, we consider the following kinds of directed graphs in our simulation study (while excluding, for example, powerlaw graphs in which there are some large-degree vertices). (i) In-degree random graph: In such a graph, on average, each vertex or bot has the same in-degree. (ii) Out-degree random graph: In such a graph, on average, each bot has the same out-degree. (iii) In-degree regular graph: In such a graph, every bot has the same in-degree. (iv) Out-degree regular graph: In such a graph, each bot has the same out-degree. (v) In- and out-degree regular graph: In such a graph, each bot has the same degree (i.e., in-degree + out-degree). We generate, for each type of the topologies mentioned above, 10 instances of graphs of $|V| = 9000$. To make them comparable, the (average) bot degree in each instance is 10 (e.g., in the case of in-degree random graph, the average in-degree is 5; in the case of in-degree regular graph, each bot has in-degree is 5). Throughout this section, we will omit the subscriptions of α and β when we assume $\alpha = \alpha_{(u,v)}$ and $\beta = \beta_{(u,v)}$ for all $(u,v) \in E$. We define $\tau = k/\alpha$, where k is the detection threshold mentioned above.

3.1. When Should a Bot Be Abandoned?

The crafty master of a botnet $G = (V,E)$ might abandon a bot after sending some number m^* of C&C messages so as to prevent the botnet from being detected. How can the botnet master compute m^*, the largest number of C&C messages before u is detected by defender? For simplicity we assume that $\mathcal{E}'_u = 0$ for all $u \in V$and $\alpha = \alpha_{(u,v)} = \beta_{(u,v)}$ for all $(u,v) \in E$; it would be easy to extend to more general cases. According to Definition 1, the master can compute m^* as follows.

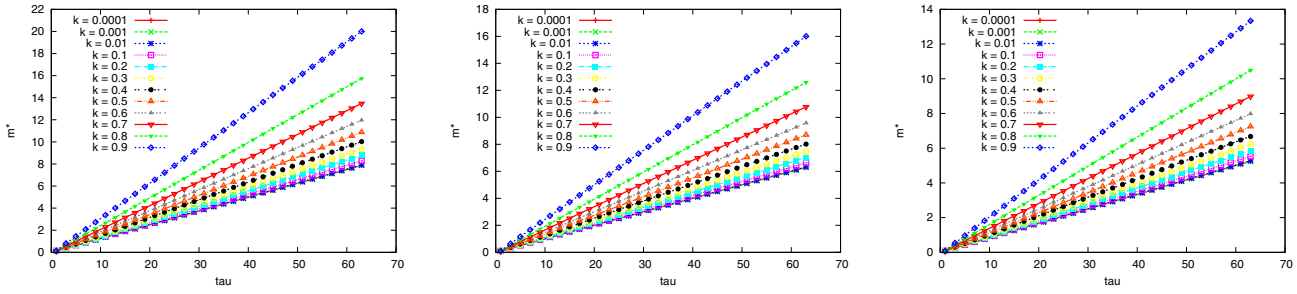
$$m^* = \left\lfloor \frac{\log(1-k)}{\deg(u) * \log(1-\alpha)} \right\rfloor,$$

where $\deg(u)$ is the degree of u in G.

To draw insights into "what is the key factor that dominates m^*," we consider example scenarios of $\deg(u) = 8, 10, 12$. Figure 2 plots m^* in the respective cases. It shows that in all of the three cases m^* is linear to τ. It also shows that for fixed τ and k, the larger $\deg(u)$ implies a smaller m^*, and that for fixed τ, a larger k (i.e., the detection system is not so good) implies a larger m^*. On the other hand, a larger τ means that more C&C messages can be sent or bots can have larger degrees without jeopardizing botnet stealth. Note that we did not consider the case that $k \approx 1$, in which case there is almost no defense, and m^* can be extremely large.

3.2. Impact of Topology on Botnet Stealth

To capture the effect of C&C activities, we set $\mathcal{E}'_u = 0$ for each $u \in V$ in each simulation — meaning that initially the



(a) $\deg(u) = 8$ (b) $\deg(u) = 10$ (c) $\deg(u) = 12$

Figure 2. Linear scale ratio of m^* vs. $\tau = k/\alpha$

probability that a bot is exposed is zero, and set $m_{(u,v)} = 1$ for any $(u, v) \in E$. The latter is motivated to capture the "clean" effect of C&C activities because, without enforcing this, a bot u may have been detected after forwarding a single C&C message (dependent upon k). This would be avoided because a crafty master might abandon a bot after sending some C&C messages. Moreover, setting $m = 1$ is not really a restriction because (1) we will consider a spectrum of k and (2) it accommodates a crafty master who commands the bots to prune broadcast C&C messages (i.e., one message flows through an arc exactly once).

We compare their stealth via the averages over the 10 graphs instances with parameters $\alpha_{(u,v)} = \beta_{(u,v)} = 0.00005$ for all $(u, v) \in E$, $k \in [0, 0.002]$. Figure 3(a) compares the detection ratios, from which we draw the following observations. First, there is no significant difference between the in-degree and the out-degree random graphs, and between the in-degree and out-degree regular graph topologies. This is because the botnet degree distributions in the in-degree random graphs are similar to their counterparts in the out-degree random graphs. The same applies to the case of in-degree and out-degree regular graphs. Second, for very small detection threshold k (e.g., very good botnet detection systems), many bots will be detected after even conducting light-weight C&C activities; for large detection threshold (i.e., not so good botnet detection systems), light-weight C&C mechanisms do not cause the detection of bots. Third, there exists a phase transition, meaning that when k is below a threshold the number of bots that will be detected abruptly increases, even if the C&C activities are light-weight. Moreover, in- and out-degree regular graphs exhibit an "all or nothing" phenomenon because every bot has the same degree.

Figure 3(b) compares botnet resilience, from which we draw the following observations. First, there is no significant difference between the in-degree and the out-degree random graphs. The same applies to the case of the in-degree and the out-degree regular graphs. Second, in all of the five types of botnet topologies, botnet resilience exhibits an almost "all or nothing" phenomenon, namely that either almost no bots are detected, or almost all bots are traced once some bots are detected. This is because detecting a small number of bots, which have some reasonable out-degrees, can lead the defender to track down almost the whole botnet.

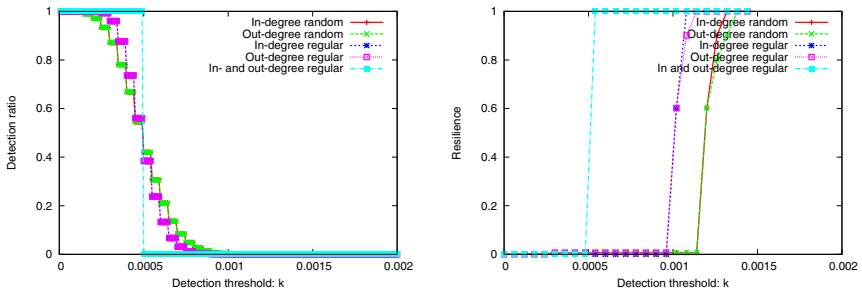
Figures 3(a) and 3(b) show that it is most difficult to detect and trace botnets with in- and out-degree regular graph topology, and that it is difficult to trace botnets of in-degree and out-degree regular graph topology when compared with botnets of in-degree and out-degree random graphs.

3.3. Impact of Fragmentation on Stealth

Recently there is a rule of thumb [21] that splitting a large botnet into smaller components would make botnets more dangerous. Therefore, it is interesting to know whether one large connected botnet or a forest of smaller botnets would be more stealthy. To gain insight into this, we conduct a case study based on f fragments, where $f \in \{1, 2, 3, 4, 8, 9, 16, 27\}$ as examples. For each f, we create 10 graphs with each graph composed of f components of size $|V|/f$, where $|V| = 9000$. As before, we always set the (average) bot degree to be 10, the (average) bot in-degree to be 5, $\alpha_{(u,v)} = \beta_{(u,v)} = 0.00005$ for all $(u, v) \in E$, and $k \in [0, 0.002]$. In the case that f does not divide $|V|$ there is a small difference between the resulting sizes. Nevertheless, by drawing the bot degree histograms, we are confirmed that, for a specific type of topology, the difference in degree distributions remains insignificant for any of the above f's.

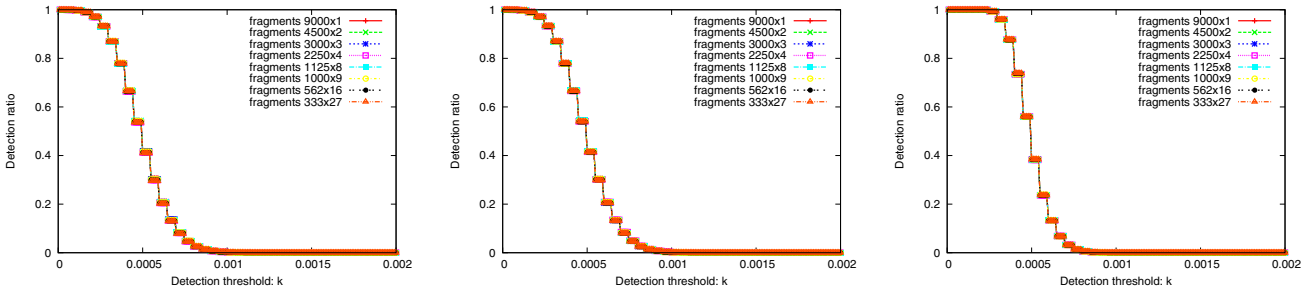
Figure 4 plots the impact of fragmentation on detection ratio. It shows that fragmentation does not change the detection ratio. This is caused by the fact that the bot degrees follow the same degree distribution, or more specifically that each bot still has (average) degree 10.

Figure 5 shows the impact of fragmentation on botnet resilience. It shows that in the in- and out-degree regular graph case (Figure 5(e)), fragmentation does not affect botnet resilience at all. However, in all of the other cases, fragmentation does improve resilience in the sense that, in order to detect and trace the same number of bots,

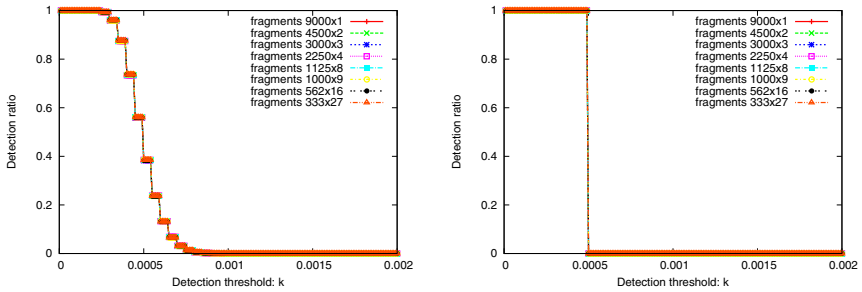


(a) Detection ratio (b) Resilience

Figure 3. Impact of botnet degree distribution ($k \in [0, 0.002]$ corresponding to $\tau \in [0, 40]$)



(a) In-degree random graph case (b) Out-degree random graph case (c) In-degree regular graph case



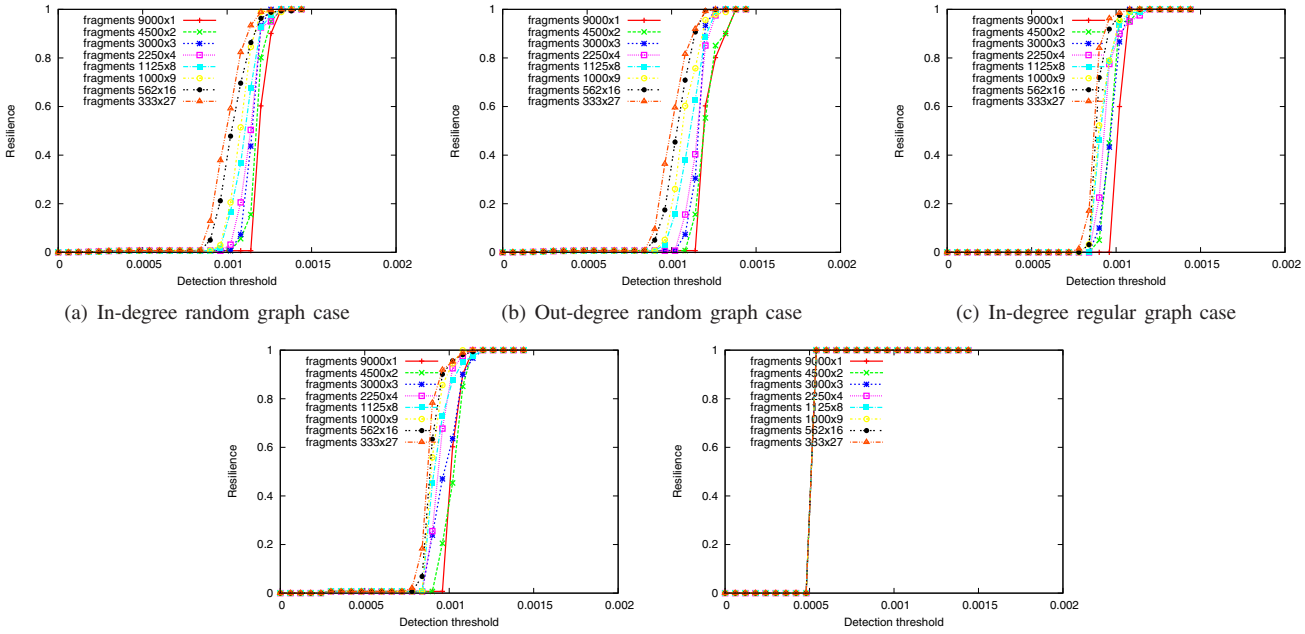
(d) Out-degree regular graph case (e) In- and out-degree regular graph case

Figure 4. Impact on fragmentation on detection ratio ($k \in [0, 0.002]$ corresponding to $\tau \in [0, 40]$)

more fragments will require a smaller detection threshold k (i.e., better detection systems). Nevertheless, there is no significant difference between the in-degree random graphs and the out-degree random graphs, and between the in-degree regular graphs and the out-degree regular graphs. The conclusion is two-sided. On one hand, if the master is able to maintain specific botnet topology such as in- and out-degree regular graphs, it does not necessarily improve botnet resilience by fragmenting a large botnet into smaller ones. Actually there is a side-effect of fragmentation because more fragments means more entry bots, which would force the botnet master to use more anonymous channels to communicate with the entry bots; otherwise, the master may have a higher chance of being traced. On the other hand, if the master is not crafty or powerful enough (e.g., the master cannot maintain in- and out-degree regular graph topology), splitting a large botnet into many smaller botnets does improve botnet resilience. This confirms the recent rule of thumb mentioned above. The caveat is again that fragmentation will impose more entry bots that may increase the probability that the master is traced.

3.4. Impact of Attack Sophistication

Since the above study suggests that there is no significant difference between in-degree random graphs and out-degree random graphs, and between in-degree regular graphs and out-degree regular graphs, in what follows we will only consider out-degree random graphs, out-degree regular graphs. We omit in- and out-degree regular graphs due to space limitation. For each of these topologies, we consider three scenarios: (a) $\alpha = \beta > 0$; (b) $\alpha > \beta > 0$; (c) $0 < \alpha < \beta$.



(a) In-degree random graph case

(b) Out-degree random graph case

(c) In-degree regular graph case

(d) Out-degree regular graph case

(e) In- and out-degree regular graph case

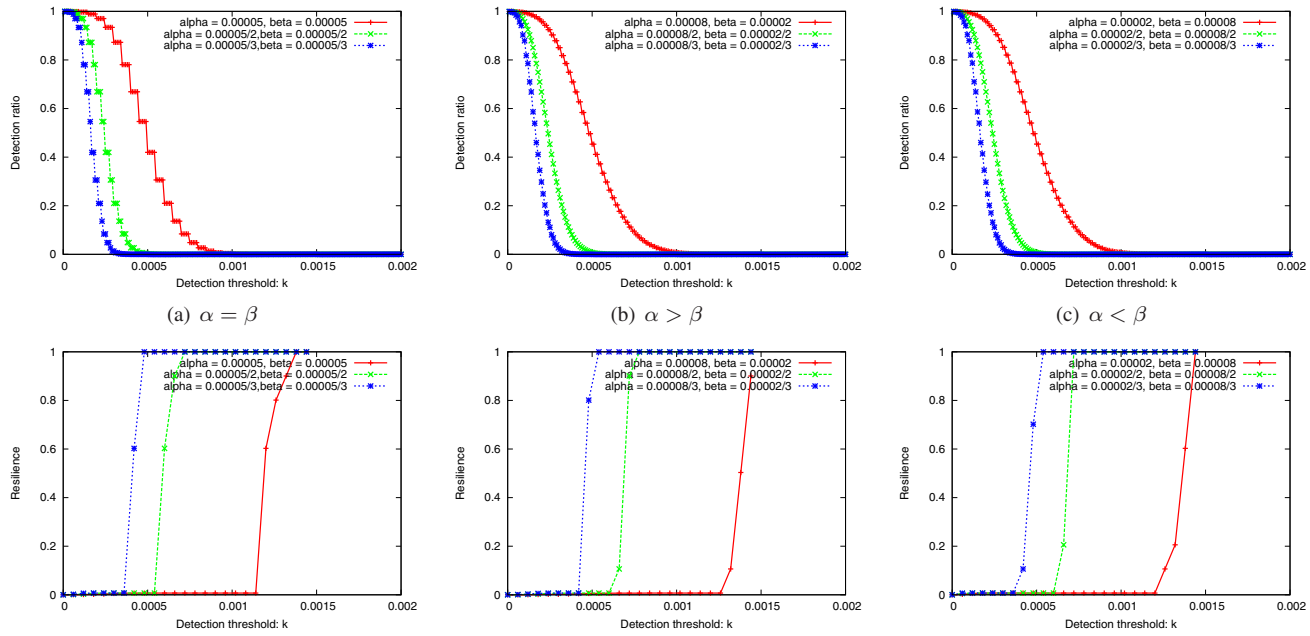
Figure 5. Impact of fragmentation on botnet resilience ($k \in [0, 0.002]$ corresponding to $\tau \in [0, 40]$)

The case of out-degree random graphs. Figures 6(a) - 6(c) plot the impact of attack sophistication on detection ratio, from which we draw the following observations. First, for a fixed detection threshold k, the more sophisticated the attack (i.e., the smaller the α and β), the fewer bots will be detected. In other words, for a fixed detection ratio, more sophisticated attacks will require more advanced detection systems with a proportional improvement. For example, in the case of $\alpha = \beta$ with detection ratio 42%, the required detection threshold for $\alpha = 0.00005/2$ is about half of the detection threshold for $\alpha = 0.0005$. Second, Figures 6(b) and 6(c) show that there is a symmetry between α and β. For example, the curve corresponding to $\alpha = 0.00008$ in Figure 6(b) almost mirrors the curve corresponding to $\alpha = 0.00002$ in Figure 6(c). This means that it has an equal effect to protect the sending bots or to protect the receiving bots, provided that the bot in-degree and out-degree distributions are about the same.

Figures 6(d) - 6(f) plot the impact of attack sophistication on botnet resilience, from which we draw the following observations. First, for a fixed resilience, when we reduce the parameters α and β (improving attack resilience) by some proportion, the required detection threshold shifts left by approximately the same proportion. This means that increasing attack sophistication by a proportion always require the defender's capability be elevated with a proper extent. Second, Figures 6(e) and 6(f) show that there is a symmetry between α and β. For example, the curve corresponding to $\alpha = 0.00008$ in Figure 6(e) almost mirrors the curve corresponding to $\alpha = 0.00002$ in Figure 6(f). This means that it has an equal effect to protect the sending bots or to protect the receiving bot, provided that the bot in-degree and out-degree distributions are about the same.

By comparing the impact of attack sophistication on detection ratio and its impact on resilience, we observe the value of being able to trace botnets. Consider the case of $\alpha = 0.00008$ and $\beta = 0.00002$ as an example. In order to detect all bots, it is required that $k \approx 0.0001$ as shown in Figure 6(b); in order to detect and trace all bots, it is only required that $k \approx 0.0014$ as shown in Figure 6(e). This shows the power of tracing because it may be very costly or expensive to improve k from approximately 0.0014 to 0.0001.

The case of out-degree regular graphs. Figures 7(a) - 7(c) plot the impact of attack sophistication on detection ratio in the case of out-degree regular graphs mentioned above. We draw the following observations. First, for a fixed detection threshold k, the more sophisticated the attack (i.e., the smaller the α and β), the fewer bots will be detected. In other words, for a fixed detection ratio, more sophisticated attacks will require more advanced detection systems with a proportional improvement. For example, in the case of $\alpha = \beta$ with detection ratio 40%, the required detection threshold for $\alpha = 0.00005/2$ is about half of the detection threshold for $\alpha = 0.00005$. Second, Figures 7(b) and 7(c) show that there is no symmetry between α and β. This is because the bot in-degree distribution is significantly different from the bot out-degree distribution (i.e., every bot



(a) $\alpha = \beta$ (b) $\alpha > \beta$ (c) $\alpha < \beta$

(d) $\alpha = \beta$ (e) $\alpha > \beta$ (f) $\alpha < \beta$

Figure 6. Out-degree random graphs ($\alpha = \beta = 0.00005$ and $k \in [0, 0.002]$ corresponding to $\tau \in [0, 40]$)

has the same out-degree but in-degree could vary). This means that it has different effect to protect the sending bots or to protect the receiving bots in this case.

Figures 7(d) - 7(f) plot the impact of attack sophistication on botnet resilience, from which we draw the following observations. First, for a fixed resilience, when we reduce the parameters α and β by some proportion, the required detection threshold shifts left by approximately the same proportion. This means that increasing attack sophistication always require the defender's capability be elevated with a proper extent. Second, Figures 7(e) and 7(f) show that there is no symmetry between α and β. This means that it has different effect to protect the sending bots or to protect the receiving bots in the case of out-degree regular graphs.

By comparing the impact of attack sophistication on detection ratio and its impact on resilience, we observe the value of being able to trace botnets. Consider the case of $\alpha = 0.00008$ and $\beta = 0.00002$ as an example. In order to detect all bots, it is required that $k = 0.0004$ as shown in Figure 7(b); in order to detect and trace all bots, it is only required that $k \approx 0.00075$ as shown in Figure 7(e). This shows the power of tracing because it may be much more costly or expensive to improve k from approximately 0.00075 to 0.0004.

4. Conclusion and Future Work

We presented a first study for fundamentally understanding and characterizing stealthy botnet C&C through a model that captures both defender's and attacker's capabilities. We emphasized on addressing a specific question — how would a crafty botnet master make a botnet as stealthy as possible? Simulation study is then leveraged to draw useful insights.

We hope that this work will inspire more studies on the C&C mechanisms of stealthy botnets and, in particular, a full-fledged framework for understanding, characterizing, and defeating stealthy botnets. Our future work includes: How can we build a holistic analytic framework so as to accommodate both C&C and attack activities of stealthy botnets? How can we extend the model so as to accommodate the attack-defense interaction that, for example, bots may suicide once them realize that they are detected so as to prevent the defender from tracing other bots? What are the other good models? What are the other stealth measures? How should we validate the models in real-world testbeds?

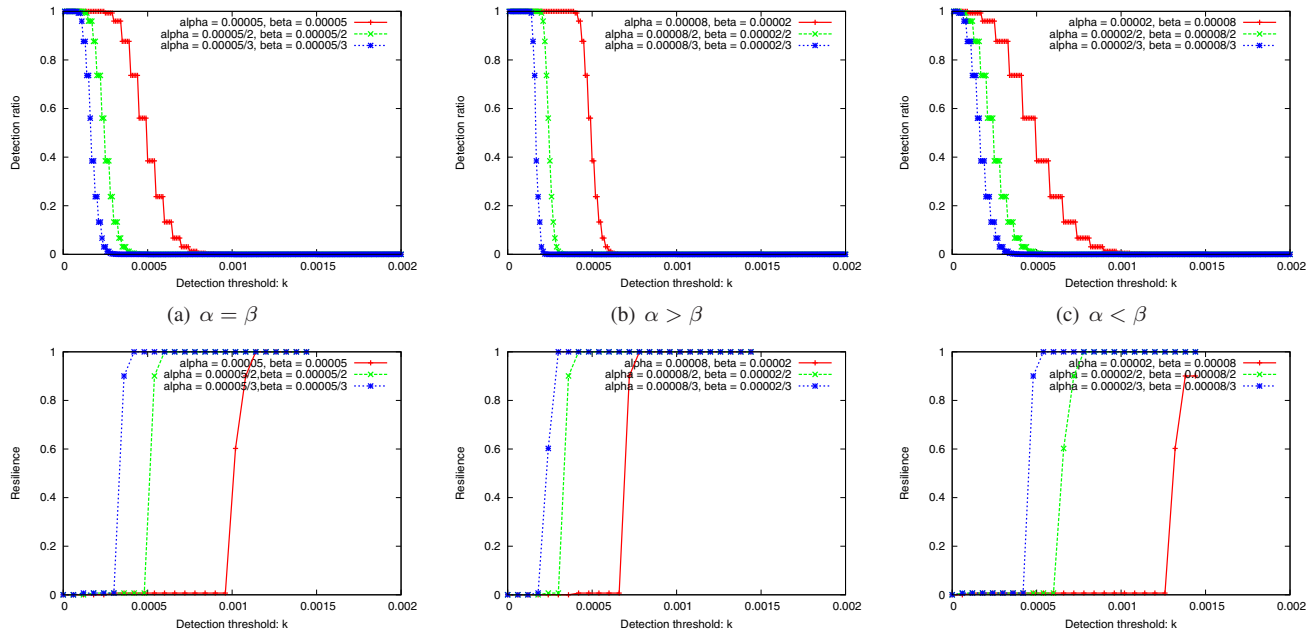
Acknowledgment. The authors are partially supported by a grant from the State of Texas Emerging Technology Fund.

References

[1] P. Bacher, T. Holz, M. Kotter, and G. Wicherski. Know your enemy: Tracking botnets. http://www.honeynet.org/papers/bots/, dated on 13 March 2005.

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

[3] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets. In *SRUTI'05*.



(a) $\alpha = \beta$ (b) $\alpha > \beta$ (c) $\alpha < \beta$

(d) $\alpha = \beta$ (e) $\alpha > \beta$ (f) $\alpha < \beta$

Figure 7. Out-degree regular graphs ($\alpha = \beta = 0.00005$ and $k \in [0, 0.002]$ corresponding to $\tau \in [0, 40]$)

[4] D. Dagon, G. Gu, C. Lee, and W. Lee. A taxonomy of botnet structures. In *ACSAC'07*, 2007.

[5] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Proc. NDSS'06*, 2006.

[6] F. Freiling, T. Holz, and G. Wicherski. Botnet tracking: Exploring a root-cause methodology to prevent denial-of-service attaks. In *ESORICS'05*.

[7] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation. In *First Workshop on Hot Topics in Understanding Botnets*, 2007.

[8] G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *USENIX Security'08*.

[9] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *USENIX Security'07*.

[10] G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic. In *NDSS'08*.

[11] C. Kanich, K. Levchenko, B. Enright, G. Voelker, and S. Savage. The heisenbot uncertainty problem: Challenges in separating bots from chaff. In *LEET'08*.

[12] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. *HotBot'07*.

[13] C. Livadas, R. Walsh, D. Lapsley, and W. Strayer. Using machine learning techniques to identify botnet traffic. In *WNS'06*.

[14] Trend Micro. Taxonomy of botnet threats (white paper), November 2006.

[15] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *IMC'06*.

[16] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. In *SRUTI'06*.

[17] A. Ramachandran, S. Seetharaman, N. Feamster, and A. Lakhina. Monitoring stealthy network conversations with sampled traffic. Georgia Tech Technical Report, 2006.

[18] E. Stinson and J. Mitchell. Characterizing bots' remote control behavior. In *DIMVA'07*.

[19] W. Strayer, D. Lapsley, R. Walsh, and C. Livadas. *Botnet Detection: Countering the Largest Security Threat*, In *Advances in Information Security*, 2008.

[20] W. Strayer, R. Walsh, C. Livadas, and D. Lapsley. Detecting Botnets with Tight Command and Control. In *LCN'06*.

[21] R. Vogt, J. Aycock, and M. Jacobson. Army of botnets. In *NDSS'07*.

[22] T. Yen and M. Reiter. Traffic aggregation for malware detection. In *DIMVA'08*.

[23] L. Zhuang, J. Dunagan, D. Simon, H. Wang, I. Osipkov, G. Hulten, and J. Tygar. Characterizing botnets from email spam records. In *LEET'08*.

[24] C. Zou and R. Cunningham. Honeypot-aware advanced botnet construction and maintenance. *DSN'06*.