Hide And Seek: Detecting Sensors in P2P Botnets

*Leon Böck, *§Shankar Karuppayah, *Tim Grube, *Max Mühlhäuser, †Mathias Fischer

*Telecooperation Group
TU Darmstadt / CASED, Germany
firstname.lastname@cased.de

† Networking and Security Group
International Computer Science Institute, USA
mfischer@icsi.berkeley.edu

§ National Advanced IPv6 Center,
Universiti Sains Malaysia (USM),
Malaysia

I. Introduction

Many of recent cyber crime activities such as Distributed Denial of Service (DDoS) and banking credential thefts are executed using *botnets* that consist of malware-infected computers, so-called *bots*, throughout the world. These bots receive orders from the *botmaster*, which instructs them to execute malicious activities. The high income generated by botnets renders them as important assets to the botmasters. For this reason, botmasters take precautions to safeguard their investments by deploying more and more unstructured P2P-based botnets, e.g., *P2P Zeus* [1] and *Sality* [2]. Their unique self-organizational capabilities render them more resilient against takedowns. As a result, these botnets are often under particular surveillance of researchers and law enforcement agencies. As part of this monitoring activities, they are constantly crawled [3] and infiltrated by sensor nodes [4]. Crawling allows to retrieve a graph of all routable nodes in a botnet, but falls short in identifying nodes behind NAT. Sensor nodes, in contrast, allow the enumeration of all bots in the botnet. Information gathered from such monitoring is then used to execute takedowns, e.g., via *sinkholing* attacks, on the botnets. In particular, sensor nodes are important for conducting *sinkholing* attacks as such attacks require to know every bot in the botnet in advance. In the actual attack, bots are tricked to believe that all other bots are no longer responsive, except designated sinkholing server(s), i.e., a sensor variant. If the attack is successful, the botmaster cannot communicate further with his bots.

Since monitoring threatens the ecosystem of botnets, recent P2P botnets implemented anti-monitoring countermeasures to prevent sinkholing, e.g., the P2P Zeus botnet uses a blacklisting mechanism [5] to detect and blacklist crawlers as well as to limit the information that can be obtained via crawling. To circumvent these measures, researchers often have to spend valuable time on developing new monitoring techniques [6], [3]. However, it is just a matter of time before the botmasters adapt to them with new countermeasures.

In this paper, we anticipate future botnet countermeasures to prepare the development of enhanced monitoring mechanisms. For that, we first introduce the membership management in P2P botnets and describe related work in Section II. In Section III, we describe a mechanism for detecting sensor nodes as the main contribution of this paper. To the best of our knowledge, we were the first to introduce a theoretically-sound mechanism that is able to distinguish sensors in a P2P botnet. Section IV provides evaluation results on an active botnet, called Sality [2], and Section V concludes the paper.

II. Botnet membership management

Each bot in a botnet maintains a Neighbor List (NL), that contains information of a subset of other known bots in the botnet and that is maintained according to the botnet's membership management protocol. Amongst others, the membership management specifies that a bot regularly probes all entries in its NL for responsiveness. If some of the probed bots are no longer alive, i.e., bots that went offline or that have been cleaned from the infected machine, they are removed and/or replaced with information of other responsive bots to ensure a connected botnet overlay. New bots are discovered by requesting NLs from other responsive bots. As P2P botnets experience churn, bots tend to retain more NL entries of other reliable bots, i.e., bots that were responsive in the past. As a result, a *backbone* of reliable nodes is formed. Bots within the backbone are usually very popular amongst all bots in the botnet, which is usually the same for deployed sensor nodes.

Sensors are usually popularized by frequently announcing them to as many bots as possible [7]. However, to remain popular within NLs of bots, sensors need to reliably respond to all incoming probing requests from other bots. Although sensors usually exhibit a characteristic of being popular within the botnet, unfortunately, this behavior is indistinguishable with those of the backbone nodes. Furthermore, sensor nodes usually refuse to participate in regular bot-activities like command dissemination and NL exchanges, due to legal issues. Hence, detecting sensors as bots that do not share neighbors is easy. However, sensors can easily circumvent this by returning spoofed entries to the requesters. As a result, it is often not possible to distinguish sensors from benign bots. In the next section, we propose a mechanism that uses a novel approach in detecting sensor nodes based on network-specific characteristics that can distinguish them from normal bots.

III. Sensor Detection Mechanism

Our detection mechanism exploits the observation in unstructured P2P botnets that reliable nodes, i.e., nodes that are responsive and available most of the time, often establish neighborhood relationships among themselves and form a backbone. Furthermore, as backbone nodes are very popular in the NLs of most bots in the botnet, there is a high probability that an ordinary bot has multiple backbone bots in its NL. The degree of interconnectivity of a bot's neighbors' can be represented by the *clustering coefficient (cc)* metric. The *cc* is often used to express the density of network. We use the *local clustering coefficient (lcc)* [8] to express the connectivity of a

node's neighbors by computing the degree of interconnectivity of its neighbors. Extreme values of 0.0 and 1.0 indicate that the neighbors are not connected amongst each other at all or that they are completely meshed respectively.

To detect sensors, our mechanism crawls the botnet and produces snapshots of its overlay topology. On that basis, we calculate the *directed* variant of the lcc for each bot x, $lcc^+(x)$, to analyze the interconnectivity of its neighbors by using Eq. (1). E is the set of all edges in the network and NL_x represents the NL of a bot x. Naturally, we set $lcc^+(x)=0.0$ if $|NL_x| = 0$ or 1, since the numerator will fast-evaluate to 0.

$$lcc^+(x) = \frac{|\{(u,v) \in E : u,v \in NL_x, u \neq v\}|}{|NL_x| \times (|NL_x| - 1)} \quad (1)$$

In the analysis, benign bots will exhibit a similar degree of interconnectivity in their neighborhood, because of the presence of the backbone. However, as mentioned in Section II, sensor nodes are usually not participating in bot-activities. They will not share or give away information of legitimate bots when they receive a NL request. Thus, their lcc^+ will differ from benign bots. A sensor has three possible behaviors on receiving a NL-request: i) return *no* neighbors or ignore the request. This will lead to $lcc^+(x) = 0.0$. ii) return *invalid* neighbors. As invalid neighbors will also not be interconnected, again $lcc^+(x) = 0.0$ holds. iii) return *responsive sensors*. If each of the returned sensor return each other as their neighbors, $lcc^+(x) = 1.0$ will hold since the sensors are connected as a full mesh, i.e., forming a clique. However, if the connectivity between the returned sensors is as in a directed cycle or not connected at all, it will lead to $lcc^+(x) = 0.0$. As stated earlier in this section, due to the tendency of connecting to the backbone, benign bots will not have these extreme lcc^+-values. Therefore, we classify bots that exhibit extreme lcc^+-values as potential sensors.

IV. Results

We evaluated our proposed mechanism on a Sality [2] botnet crawl dataset that consisted of continuous multi-session crawling from *03/23/2015* to *03/29/2015*. In each crawl session, our crawler sent 30 NL requests to each discovered node. We also identified and sanitized churn-affected nodes from our dataset. For our evaluation, we categorized the sanitized dataset into chunks of hourly aggregated crawl snapshots that each consists of an average of 42 crawl sessions. To increase the efficiency of our mechanism, we further picked only the chunk that has seen the least number of total nodes in a day, which was consistently the 24th chunk of the day, i.e., 23:00 CET, for the whole week. Our proposed mechanism is then executed on the selected chunks.

The result of our analysis is depicted in Figure 1 that shows the lcc^+ value in dependence on the node popularity in the network for the chunk on *03/25/2015*. As can be observed at the bottom right corner of the figure, many nodes have a value of $lcc^+ = 0.0$, although having very high popularity. These were verified as sensors deployed by researchers. To evaluate the accuracy of our detection mechanism, we repeated the analysis on all seven selected chunks. We were able to construct a list of nodes that have been marked by our mechanism for having extreme values as explained in Section III. We then manually verified each of the marked nodes by looking at the raw datasets and by analyzing their past and future behavior. The results are given in Figure 2 that shows the number of marked and verified sensors respectively for the whole week. We detected an average of 20 verified sensors daily with an average false positive rate of 0.2. 62% of all verified sensors were observed to return no neighbors when requested, while the remaining returned only responsive sensor nodes. We also looked at outliers in our analysis, i.e., $lcc^+ \neq 0.0$ and $lcc^+ \neq 1.0$, and were able to verify that they were benign bots.

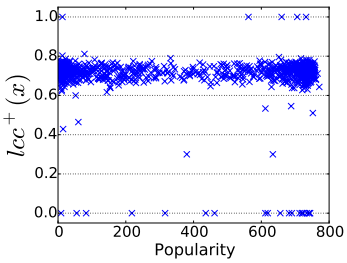

Fig. 1. lcc^+ Analysis (*03/25/2015*)

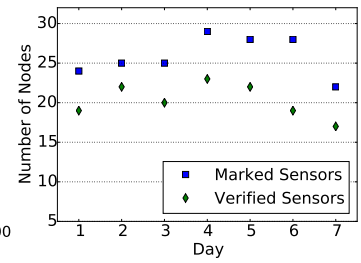

Fig. 2. Sensor Verification Analysis

V. Conclusion and Future Work

In this paper, we presented a novel sensor node detection mechanism that is able to detect sensor nodes with an average false positive rate of 0.2. From our analysis, we were able to conclude that many existing sensor implementations are susceptible to be detected by our mechanism. The findings from our work should be carefully considered by other researchers conducting monitoring activities to avoid being detected by botmasters. As future work, we are focusing on improving our mechanism to detect more sophisticated sensors that may have eluded our current proposal. Moreover, we are also looking into the direction of developing strategies for sensors to evade such a sensor detection mechanism.

References

[1] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly Resilient Peer-to-Peer Botnets Are Here: An Analysis of Gameover Zeus," in *IEEE MALWARE*, 2013.

[2] N. Falliere, "Sality: Story of a peer-to-peer viral network," Symantec Corporation, Tech. Rep., 2011.

[3] S. Karuppayah, M. Fischer, C. Rossow, and M. Muhlhauser, "On Advanced Monitoring in Resilient and Unstructured P2P Botnets," in *IEEE ICC*, 2014.

[4] B. Kang, E. Chan-Tin, and C. Lee, "Towards complete node enumeration in a peer-to-peer botnet," *ASIACCS*, 2009.

[5] C. Rossow, D. Andriesse, T. Werner, B. Stone-gross, D. Plohmann, C. J. Dietrich, and H. Bos, "P2PWNED: Modeling and Evaluating the Resilience of Peer-to-Peer Botnets," in *IEEE SNP*, 2013.

[6] S. Karuppayah, S. Roos, C. Rossow, M. Mühlhäuser, and M. Fischer, "ZeusMilker: Circumventing the P2P Zeus Neighbor List Restriction Mechanism," in *IEEE ICDCS*, 2015.

[7] J. Yan, L. Ying, Y. Yang, P. Su, Q. Li, H. Kong, and D. Feng, "Revisiting Node Injection of P2P Botnet," in *NSS*, LCNS, 2014.

[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.