Of Bees and Botnets

Vijay Sarvepalli[1]

[1] Software Engineering Institute, Carnegie Melon University, Pittsburgh, PA 15213, USA
vssarvepalli@sei.cmu.edu

Abstract. Botnets ability to grow to large sizes combined with our inability to exhaustively incapacitate them has forced us to look for more effective methods to model their growth and seek ways to curtail it. Particle swarm optimization is a stochastic computation technique based on the movement and intelligence of swarms. This study uses a simulated bee colony to model the growth of a botnet. Botnets are like swarms of bees in several ways. For both, a successful hive grows while withstanding losses. A swarm of honeybees also uses distributed decision making [1] similarly to a botnet. This study uses swarm optimization to model the growth a botnet and estimate the optimal number of "scout bees" required to make a botnet "hive" successful. The technique is applied to the Mirai botnet to simulate its growth, find ways to reduce that growth, and minimize distributed denial-of-service attacks launched from these botnets.

Keywords: particle swarm optimization, botnets, artificial bee colony, Mirai botnet

1 Introduction

In recent years, growth of botnets has provided nefarious actors the ability to create massive attacks against even large-scale services providers like Akamai and OVH [2]. The more recent massive attacks have been attributed to the Mirai botnet and its elusive yet powerful backend of Internet of Things (IoT) [3]. What can we learn from a botnet such as Mirai? The malware code of Mirai has revealed a decentralized model for growth, an almost worm-like infection behavior [4]. This type of self-propagation can sustain takedown attempts and survive attempts to curtail its power. Another factor that has contributed to Mirai's success is the large number of insecure, lightweight devices on the Internet. In this paper, I explore the swarming behavior of bee colonies, which is one of most impressive examples of decentralized decision making seen in animal groups. While there are several nuances and differences in this ecological behavior between bees and botnets, there is some commonality in the collective behavior of loosely coupled individual entities. I will show how this meta-heuristic [5] model can help describe Mirai botnet growth and hopefully motivate others to build on this model in their analytical techniques to reduce its spread.

2 Bee Behavioral Ecology

Bees in a beehive can be categorized into three broad categories:

- active foragers: visit current food sources in memory and repeat activity until exhausted
- scouts: explore new food sources in a random way (or look for a new home for the hive)
- inactive foragers: rest, recover, and swap with active foragers once well rested

Bees have settled into an optimal ratio for these roles in the hive to create a successful colony and safeguard the ongoing survival of their species [6]. As the bees carry out their roles, they effectively communicate with each other using a waggle dance [7]. The figure-eight-shaped waggle dance is very detailed and communicates specific information to the other bees. The scout bees especially can communicate location of a food source with accurate angle and distance information. Researchers such as Karaboga and Basturk have done extensive work in this area, and I use some of their results in my modeling technique [7]:

- average percentage of Active, Scout, and Inactive bees (75%, 15%, 10%)
- probability of information transfer from Scouts to Active foragers (0.9–0.95)
- probability of finding a new food source or new site (0.005–0.01)

For simplicity, in the modeling I have not introduced environment and external effects on the bees. More attributes can be added later to enhance this model to provide a richer model for ongoing research.

3 Mirai's Scanning and Propagation

Mirai botnet operation has been explored in detail by Antonakakis et al. [8], who showed that each infected machine becomes a potential scanner to recruit other machines into the botnet. The botnet Command and Control (C2) is usually engaged to accept commands for a DDOS attack. These commands from the C2 server are currently not included in this model. Mirai's scanning activity is the focus here. This scanning is performed by already compromised machines and not from a central C2 server. This is much like the self-propagation behavior of worms, in contrast with typical malware that performs tasks that have been directed from a C2 server. A scan activity can also be triggered when a C2 server requests that a scan be performed. The scanning activity is randomly selected but has some whitelisted networks that will not be part of a scan, such as the DOD's network (214.0.0.0/7) and General Electric's network (3.0.0.0/8). Essentially the scanning destination is its own pseudorandom number generator (PRNG), like a "random search" algorithm to explore new victims for the botnet [9]. The malware also has many nuances such as the use of "Loaders" for distributing the malicious code to the victims [10]. These can be added to the model to make it richer for ongoing research, as mentioned earlier.

4 Comparing Mirai and Bees

As mentioned before, the ability to perform distributed decisions making stands out as a common feature between the bees and the botnet. There are also heuristics that I have developed, shown in Table 1 and **Error! Reference source not found.**, to model botnet growth on the simulated bee colony method.

Table 1. Modeling botnets on beehives.

Entity (Beehive)	Element (Botnet)	Characteristic comparison
Active bees	An infected device is part of a botnet and ready for C2 to launch an attack.	Active bees are like robots, or even zombies, dedicated to performing what is programmed in their memory. Their primary job is foraging.
Scout bees	Scanners actively infect other machines and bring them to the botnet.	Scout bees perform random searches for food sources with little clue about where to go. Their model is to randomly search and then assess the viability of a good food source or nesting location.
Inactive bees	Compromised devices are no longer reachable from the C2 network to take action such as performing a DDOS attack.	Inactive bees, numbering about 10% of a hive, need rest and are unable to take action such as foraging. These bees can become active and even later become scouts.

Table 2. Important factors used in simulation of the bee colony.

Beehive observed statistics	Element (Botnet) Characteristic comparison
The probability of a scout bee's success in finding a new site is 0.005 to 0.01.	Active scanners can compromise a new machine with a probability of *p*. This probability is time bound, with a 6–15-second period in foraging for the scout bee.
About 15% of the bee colony is made up of scout bees.	About *x*% of the botnet is composed of active scanners that are able to sustain scanning and reach new devices for compromise.

It is important to note that the purpose of scouting in the beehive is to find a better food location or locate a new nesting site. In the modeling technique, the purpose of scouting is to increase the effective size of the botnet. This should be apparent when I highlight the heuristics below that apply this model to a botnet.

5 Simulating Colony Heuristics

Captured below is the pseudo code of the particle swarm optimization technique applied to the simulated bee colony (SBC). The SBC is used to model the growth of a botnet. Later a comparison of the statistics collected from multiple ISPs will help tune this model and develop an equation for ease of use.

```
build a hive with nb number of bees
separate the hive into three kinds of bees
loop max_scouting_operations
foreach bee in hive
  if bee is scout
   loop ns (scouting numbers) times
    do random locks and searches for new recruits
    get new recruits using probability density function
    evaluate recruits with deterministic ratio
   end loop and return new recruits
   add new recruits to hive
  if bee is active_forager
   establish and be prepared to go scout
  if bee is inactive_forager
   plan for move to active or death/demise
 implement loss/death decrease in hive size
end loop scouting
evaluate size of hive
```

The intended result is to understand the effectiveness of the botnet's scanners and find ways to reduce the botnet's growth by targeting the scanners as one would target scout bees to prevent bee population growth.

6 Running the Numbers

In collaboration with two ISPs and Akamai, I collected data to get some realistic values for this simulation. The first ISP shared scanning data from its dark space (unused IP space) that provided some raw numbers about the global Mirai scanners looking for victims in this unused network space. The second ISP shared more specific data on both internal and external devices performing scanning that can be identified as Mirai scanners. This data provided the following insights:

- About 13% of the compromised botnet machines performed repeated scanning activity (or scanning over multiple days).
- Scanning activity showed a success rate of 0.0005 to 0.001 in reaching and successfully adding another victim device to the botnet.

Using this data and the above logic in software code, I ran some simulations to understand how the botnet has grown over time. The graph in Fig. 1 shows the number or the size of the simulated botnet against the number of scanned IP addresses as observed by Akamai. The graph compares the relative growth of scanning activity to the SBC model. The data from Akamai is collected from Akamai's 2016 DDOS report, which discusses the Mirai botnet [11].

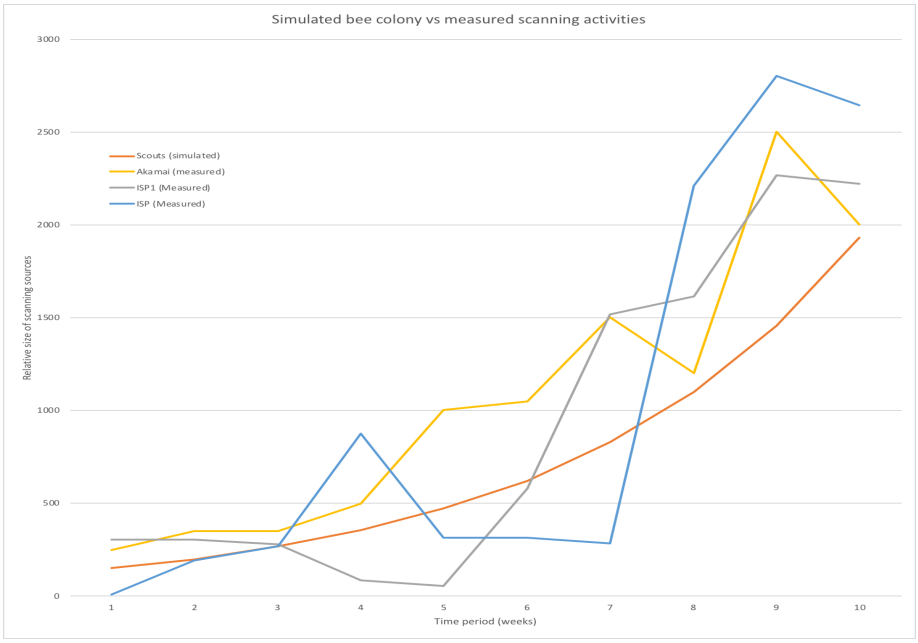

Fig. 1. Comparison of simulated bee colony to actual botnet scanning data from three providers (Akamai and two ISPs). (Data adapted from [9].)

The SBC model is represented in the formula below and can be used to predict the size of the botnet:

$$\Sigma(\tau) = N\left(1 + \frac{Pscout * Psuccess}{n}\right)^{n\tau} - \mu * N \quad (1)$$

The $\Sigma(\tau)$ is the total size of the botnet at any given time τ, where N is the current size of the botnet, Psuccess is the probability success rate of scanning, Pscout is the percent of devices that are scanners, n is the number of scan operations per time period τ, and μ represents the decrease in size of the botnet due to a simulated death or reduction in hive size. The potential power of the botnet itself is slightly less, as depicted below.

$$\Sigma_{effective} = \Sigma_{TOTAL} - Ninactive \quad (2)$$

The number of scanners (or scouts) for any given N-sized botnet can be calculated as shown in Equation 3.

$$Nscout = Pscout * N \quad (3)$$

The current simulated model from the equation shows a steady growth in the population of the botnet, which is a very simplified version of the heuristics. The attempt here is to focus on understanding botnet growth in its simplest form and learning ways to curtail that growth by focusing on the key growth factor—scout bees. The model does not take into account other complex phenomenon observed in nature, such as the external environment, at this time.

7 Controlling the Beehive or the Botnet

Beekeepers who do not wish to increase their number of active hives usually use one or more swarm control methods to effectively prevent bee population growth [12][13]. Some of these methods pursue reducing the swarming drive that exists in the bees, especially the scout bees. I propose some techniques to reduce the growth of botnets by extinguishing the scout-like activity of scanning that is initiated from the botnet. Cumulatively, these multiple techniques can reduce the botnet's growth.

In earlier efforts, large corporations like Microsoft have gone after botnets by pursuing cleanup of infected machines or by takedown of the C2 infrastructure (domain takedowns) [14]. The self-propagation techniques used by Mirai make it more resilient to such takedowns. The proposed solution here—to control a portion of the botnet (scouts or scanners) instead of the entire botnet—makes it a more effective strategy than large-scale takedown attempts. Another technique suggested by many is to fix the problem of the IoT, or all the lightweight, insecure devices present on the Internet. The need for more security in the design, manufacture, and operations of these devices cannot be ignored. However, in the meantime, my suggested methods can happen very soon, in contrast to the longer-term work of securing devices before they come to market.

I model two key factors in Equation (1) that are targets for controlling the botnet growth:

1. *P*success: probability of success of a scan from a scout/scanner
2. *P*scout: percentage of scouts/scanners active in the botnet

Many researchers have done thorough analysis of Mirai's malware and documented some of its weaknesses. The botnet characteristics listed below describe weaknesses that I have tried to map to these two variables that can impact the botnet's growth:

1. On reboot, an infected device loses its infection and therefore its position as a scanner (as it can still get re-infected). Mathematically speaking, where *D*uptime is the infected device uptime,

$$Pscout \propto Duptime$$

2. The malware's scanning activity will exclude certain portions of the Internet. The solution *Xi*, which is a possible victim's IP address, is a member of *NV* (maximum number of victims):

$$Xi \in \{i = 1,2, \ldots NV\} \,\&\, Xi \notin \{IANA - localnet, DOD - net..\}$$

3. The malware's scanning activity efficiently ignores unresponsive IP addresses, but it can be engaged or delayed by a device that responds to a scan but cannot be compromised. *D*quality is the victim device's quality. If the victim device either has a password reset by default or denies access from untrusted networks, it will disproportionately impact the probability of Mirai's success:

$$Psuccess \propto \frac{1}{Dquality}$$

4. An infected machine's IP address consistency [15] (Dynamic Host Configuration Protocol or DHCP lease time and unchanged IP address) can prevent the device from being used as a scanner. This is mainly because the loss of registration status with the C2 server introduces instability to the scanning evocation.

The graphs in Fig. 2 and Fig. 3 illustrate that the scanning activity is successfully repeated by only about 13% of the infected devices (like scout bees). These devices are really the ones that are able to successfully infect other victims

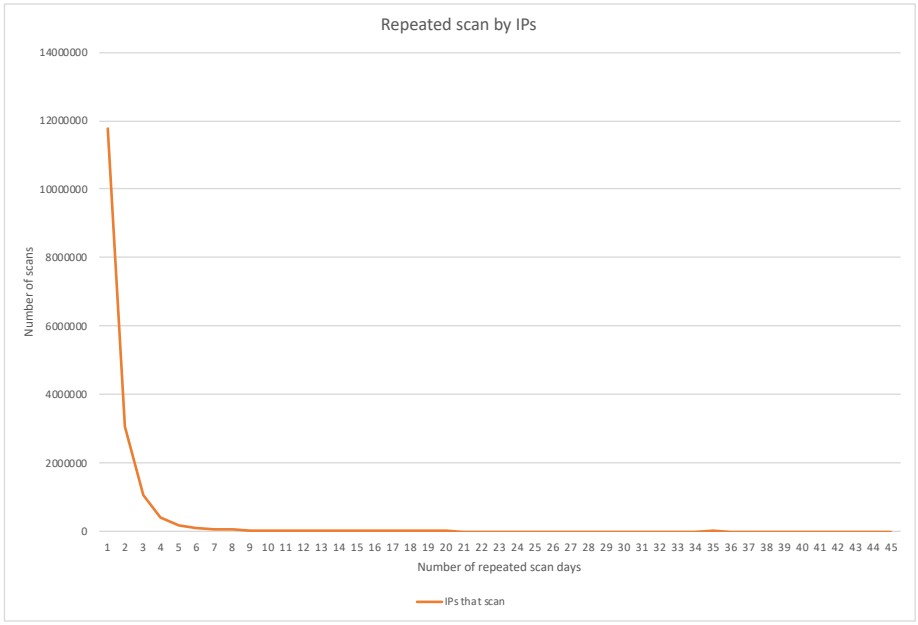

Fig. 2. Repeated scan by IPs (device IP addresses).

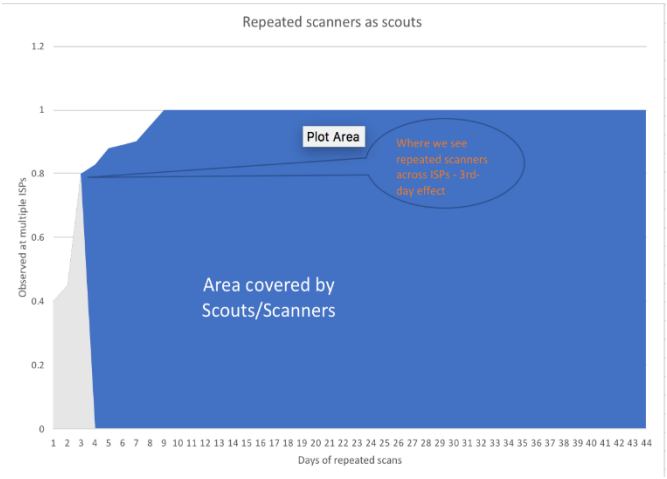

Fig. 3. The "scout" scanners cover most of the scanning area and are visible at multiple ISPs.

Just like in bee colony control, by focusing our effort on this smaller number of entities (scout bees or scanners), we can limit the growth of the botnet. The following recommendations are mostly operational activities that can enable ISPs and device vendors to concentrate their activities to control the Mirai botnet:

- ISPs can analyze their dark space for repeated scanners that appear for three or more days and target remediating these.
- The remediation work can be extended to find the manufacturers and types of devices that are likely to become effective "scouts" and pursue patching and fixing vulnerabilities for these devices.
- ISPs can monitor outgoing scanning activity and pursue modifying DHCP lease times to reduce sustained scanning activity from these devices. (Such monitoring will also not impact their customers' normal traffic in a negative way.)

8 Conclusion and Further Research

Bio-inspired models can be effective ways to analyze and understand survival techniques that mimic characteristics of a biological system. Fig. 4 shows a large amount of scanning activity over a two-year time period along with the size of the Mirai botnet. It provides some insight into the botnet's current status: it appears to be in a more steady-state phase, waiting to be triggered into another growth pattern. This is an appropriate time to prepare operational activities that can help reduce the growth of a botnet built with these types of self-propagation capabilities. Future work can expand this model to include information about environmental factors such as distance and weather [1], as well as optimization of the SBC as proposed in the Artificial Bee Colony (ABC) optimization studies [14]. These enhancements will increase the accuracy of the

model as well as the efficiency of the algorithm. In the ever-changing Internet environment where these botnets exist, these additions will be necessary for a good model that can scale and address concerns of the cybersecurity community. The enhanced model should augment Equation (1) to determine more factors to better simulate these botnets and defend against the cybersecurity threats imposed by them.

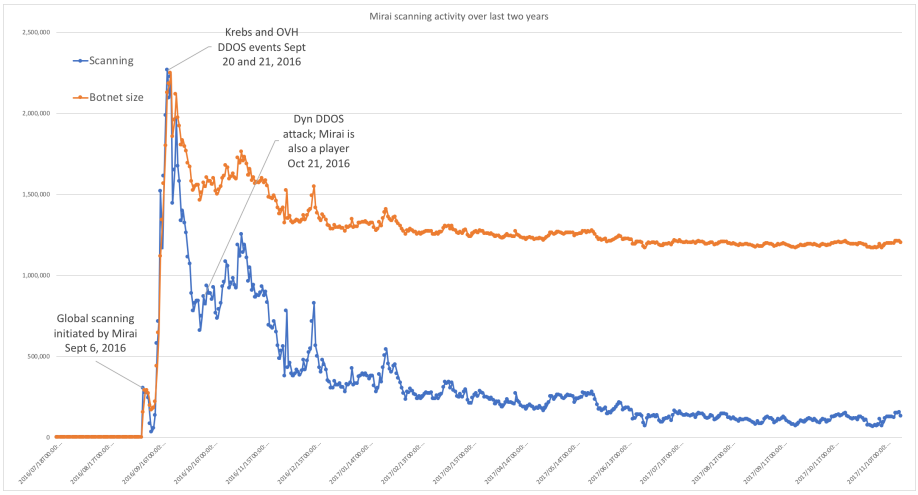

Fig. 4. Mirai scanning activity over the last two years.

9 Acknowledgment

I would like to thank Soumya Moitra (Carnegie Melon University), Angelos Stavrou (George Mason University), Constantinos Kolias (George Mason University), Ryan Goddard (Muscatine Power and Water), and Martin Mckeay (Akamai Technologies) for their input, support and feedback.

References

1. Janson, S., Middendorf, M., Beekman, M.: Searching for a new home—scouting behavior of honeybee swarms. Behav. Ecol. 18, 384–392 (2006).

2. Kolias, C., Kambourakis, G., Stavrou, A.: DDoS in the IoT: Mirai and other botnets. Computer 50.7 80–84 (2017).
3. Bertino, E., Islam, N.: Botnets and Internet of Things security. Computer 50.2, 76–79 (2017).
4. Moore, D., Shannon, C.: Code-red: a case study on the spread and victims of an Internet worm. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, pp. 273–284. ACM, New York (2002).
5. McCaffrey, J.: Simulated bee colony algorithm for the traveling salesman problem using Python." https://jamesmccaffrey.wordpress.com/2015/05/30/simulated-bee-colony-algorithm-for-the-traveling-salesman-problem-using-python/ last accessed on 2017/12/22.
6. Getz, W.M., Brückner, D., Parisian, T.R.: Kin structure and the swarming behavior of the honey bee Apis mellifera. Behav. Ecol. Sociobiol. 10, 265–270 (1982).
7. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. 8, 687–697 (2008).
8. Antonakakis, M. et al.: Understanding the Mirai botnet. In: Proceedings of the 26th USENIX Security Symposium, pp. 1093–1110. USENIX Association, Berkeley (2017).
9. Passino, K.M., Seeley, T.D., Visscher, P.K.: Swarm cognition in honey bees. Behav. Ecol. Sociobiol. 62, 401–414 (2008).
10. Riegel, M.C.: Tracking Mirai: an in-depth analysis of an IoT botnet (MS thesis). Computer Science and Engineering, The Pennsylvania State University (2017).
11. Akamai. Q4 2016 state of the Internet - connectivity report. https://www.akamai.com/kr/ko/multimedia/documents/state-of-the-internet/q4-2016-state-of-the-internet-connectivity-report.pdf last accessed on 2017/11/23.
12. Wikipedia. Swarm control methods. https://en.wikipedia.org/wiki/Swarming_(honey_bee)#Swarm_control_methods last accessed on 2017/12/22
13. Boreham, M.M., Roubik, D. W.: Population change and control of Africanized honey bees (Hymenoptera: Apidae) in the Panama Canal area. Bulletin ESA 33.1, 34–39 (1987).
14. Hiller, J.S.: Civil cyberconflict: Microsoft, cybercrime, and botnets. Santa Clara Comput. High Technol. Law J. 31, 163–214 (2014).
15. Moura, G.C.M., Gañán, C., Lone, Q., Poursaied, P., Asghari, H., van Eeten, M.: How dynamic is the ISPs address space? Towards Internet-wide DHCP churn estimation. In: IFIP Networking Conference (IFIP Networking), 2015. IEEE, New York (2015).
16. Anuar, S., Selamat, A., Sallehuddin, R.: A modified scout bee for artificial bee colony algorithm and its performance on optimization problems. J. King Saud U. Comput. Inf. Sci., 28, 395-406 (1973). https://www.sciencedirect.com/science/article/pii/S1319157816300039