

# Real-time Visualization of a Distributed Peer-to-Peer Botnet

Alejandro Soler Gayoso<sup>1</sup> and Rushdi Abualhaija<sup>2</sup>

<sup>1</sup>Worcester Polytechnic Institute, asolergayoso@wpi.edu

<sup>2</sup>Worcester Polytechnic Institute, rmabualhaija@wpi.edu

## Abstract

*Botnets are a major persistent threat to modern technological society. Finding ways to detect and manage these botnets is an important task. One method to help deal with these otherwise invisible threats is to create tools to visualize and perform operations on the nodes in them easily. To accomplish this task we review many of the available studies in this field. After our review we determined that creating a visualization with the idea of contextual network navigation at its core would produce a valuable tool for understanding and controlling botnets. From these findings we developed such a tool.*

## 1 Introduction

Botnets stand out as one of the largest security threats to the average user. Composed sometimes by tens of thousands of nodes, they can collectively perform massive coordinated attacks, such as Distributed Denial of Service. Traditionally, bots in the network receive their commands from Command and Control (C&C) servers, which produced malicious commands camouflaged in regular network traffic.

It is no easy task to bring down one of these nefarious networks, especially as the size of the botnet increases. Taking down one individual node will not serve any purpose, as there could be thousands of others still at work. However, if the C&C server were to be compromised, it could lead to the downfall of the entire botnet. However, the rise of a new type of botnet built on a distributed peer-to-peer (P2P) architecture are resilient to this approach since commands are not distributed from a single C&C server, but rather from each node to its neighbors and so on.

Effectively controlling these type of networks can be a rather overwhelming task. In a stealthy P2P botnet, traffic should always take different routes, and commands should come from different origins, mak-

ing the management quite a tedious and leading to inefficient routes, and partial command propagation. For that reason, we are proposing a visualization of a P2P botnet on real-time, that would optimize the its management by providing an overall view of the topology, as well as route statistics, for better propagation.

## 2 Background

Botnets are extremely prevalent today and have already caused untold damage with their DDoS attacks and ability to silently infect and perform actions from within countless hosts. For example in 2008 the Kraken botnet was found to be present within 10% of all fortune 500 companies and was capable of sending over 600,000 emails a day. Another aspect that makes botnets risky is the range of threats from malicious actors to bored kids such as in the case of the 2016 Mirai botnet which infected over 600,000 machines and used to DDoS attacks to make the internet inaccessible to most users on the east coast. This attack was put together by a group of kids looking to have an advantage in a videogame[3]. Even more recently botnet attack on Electrum bitcoin infected more than 150,000 user and stole \$4.6 Million in value from users wallets. Due to the prevalent and pervasive threat that botnets pose, it is important to effectively control them.

In this section we will discuss what a botnet is, visualization tools that exist to help deal with their threat, and one particular technique that we will be addressing in this paper, Contextual Network Navigation.

### 2.1 Botnet

Since our paper deals with botnets we should discuss what exactly a botnet is. A botnet is a network of private computers that have been infected by some

malicious malware. There are several ways a particular target machine could get infected by the bot malware, an attacker with physical access to machine could plant it, the user could make an error such as clicking a malicious link that downloads it, or by a phishing attack. The malicious malware can have up to two goals. 1) Perform malicious actions on the target's machine e.g. send spam emails, steal information, perform a DDoS attack, etc. 2) Propagate themselves to other targets to grow the botnet. It is important to also note that once a particular machine is infected it becomes easier to infect machines near the infected one. For example a bot on an infected machine could potentially send phishing emails from the victims email account to colleagues and friends of that user. Since the email is coming from a trusted source from the new victims' perspective they will be much more likely to open a link or attachment and thus infect themselves.

## 2.2 Existing Visualization Tools

There are several existing visualizations concepts and tools that are useful in the space of visualizing botnets. In their paper "Countering security information over-load through alert and packet visualization", Conti et. al developed a framework for designing security visualizations. They additionally created two systems with this tool and surveyed security experts on the efficacy of the tools they created. Their surveys showed that this type of interactive visualization presented a significant advantage to security analysts over traditional text based analysis tools. [2]

In their paper "Real-time and forensic network data analysis using animated and coordinated visualization, Krasser et. al wrote about the limitations of just having human analysis of network traffic to detect anomalies in that the volume of traffic is too great for a human to analyze. Additionally they covered how automatic techniques generate an unacceptable number of false positives and false negatives. Krasser et. al developed a tool to allow human and automatic processes to achieve better detection margins, and they showed that this approach in fact works.[6] Both of these above studies show the efficacy of and need for the kind of visualization tools we attempt to develop in this paper.

In their paper "Gridmapper: a tool for visualizing the behavior of large-scale distributed systems", Allcock et. al developed GridMapper, a tool for registering data to physical locations and then automatically laying it out on a map. It also provides tools for animating these data points and resources and how they change based on their activities. This is similar to

how we want to visualize the distributed peer-to-peer network and the flow of network traffic across it.[1]

## 2.3 Contextual Network Navigation

In their paper "Contextual network navigation to provide situational awareness for network administrators", Gray et. al proposes to use contextual navigation to provide situational awareness for the user. Instead of plotting all the nodes and link on the same relative location, a context-based navigation would present a different location for every location, representing the "best" route from the location to all the others.[4] This paper is useful as it lays out an interesting method of representing a network which in our case would be a botnet. The contextual navigation would allow a user to explore the botnet in an intuitive way. Additionally, by adding more data such as ip addresses, node status, and visualizing network traffic along the links, it could allow a user, such as a network administrator to much more easily see anomalies in the network. This contextual network navigation is the main idea we will explore and implement in this paper.

## 3 Method

P2P architectures run on top of existing networks [1]. Their decentralized fabric allows every node to act as a client and a server, and thus eliminating the need for a dedicated server and making them cheaper and easier to set up. When used as a botnet, it can be a tremendously difficult task to trace the attack back to the origin due to the distributed nature of the network. Similarly, as the size of such a network increases, it becomes harder administer it. For that reason, as well as having worked with P2P networks in the past, it seemed a good area to improve through the power of data visualization. In particular, we focus on the building a visualization capable of displaying close to real-time data about the nodes and the links of the network, as well as providing additional context-based information for every node.

### 3.1 Generating Real Data

Since real-time data was a necessity for this project, we built and implemented a real P2P botnet with 15 bots in a controlled environment. This botnet was the responsible for generating realistic traffic that would then be utilized for the visualizations. Although the scale was not as large as typical malicious botnet, we were able to replicate the P2P distribution, avoiding



### 3.3 Visual Representations

Similarly to [4] all the possibilities we considered to visualize the topology of the network were in the realm of node-link and adjacency diagrams. In particular, one of the questions that we asked at the beginning was whether a P2P botnet carries some sort of intrinsic hierarchy that could be represented through hierarchical force layouts. Among the potential advantages of using hierarchies, we find the ability to mimic real path selection algorithms[4]. Nonetheless, P2P networks and especially botnets are highly dynamic networks [1] that are constantly adapting to new changes in the environment to ensure some level of robustness. Hence, hindering the natural formation of hierarchical structures, and therefore establishing the master-slave relation as the sole difference. Consequently, when visualizing the nodes, we avoided depicting any hierarchy, by representing all the nodes with the same shape and size.

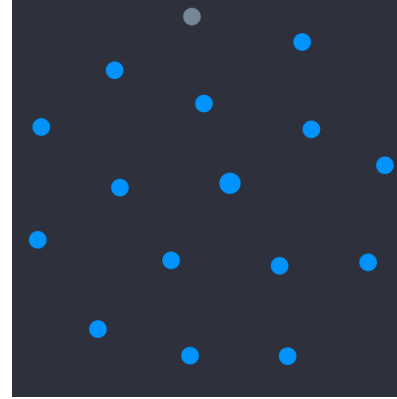
Additionally, we pondered whether all the links between the nodes should be included in the visualization, as it poses a trade-off between structural information of all the links and gains in readability. This question ties back to the previous one, as the ever changing topology and lack of hierarchy makes the distinction between links' relevance rather ambiguous. As a result, considering the reduced size of our botnet, we decided to include all links in the visual representation. This decision however came at some cost, namely the added overhead that it takes when parsing all the neighbors of a given node.

#### 3.3.1 State of the Network

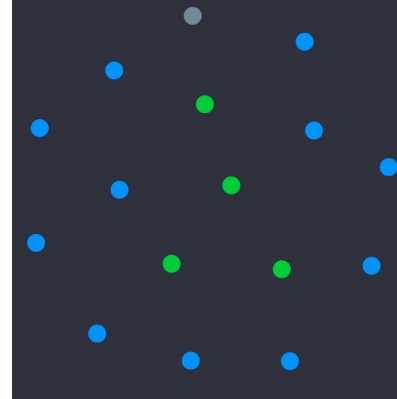
After taking into account the different considerations mentioned before, we proceeded to create a visualization that accurately represented the state of the network in real-time. As shown in Figure 6a, the master can query the state of the network at any point in time, however, the text output is quite impractical to quickly understand the topology and the state of the nodes. For that reason, we a dynamic network graph using a force layout, that draws all the nodes as circles on the screen and connects them to each other, based on each of their "neighbors" fields. Additionally, each node is color filled with a different color depending on its state, blue being "Asleep", green "Ready", and grey "Unreachable". Figure 3 how the boot sequence of the botnet can be visualized in real-time. At first, all nodes are initialized but unreachable, depicted as disconnected grey circles, see Figure 3a. In the next step of the sequence, the nodes have been recognized by the master with the



(a) Nodes have been initialized but remain unreachable.



(b) Nodes are infected by master in the 'Asleep' state.



(c) Some of the nodes have been awoken.

Figure 3: Real-time view of the botnet boot sequence.

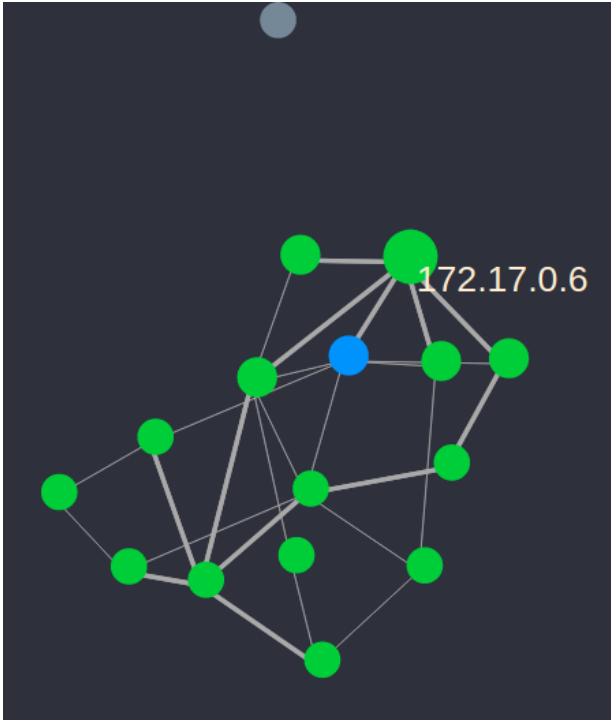


Figure 4: Visualization of the state of the network.

"infect" signal and given the 'Asleep' state, therefore their color changes to blue, see Figure 3b. At the instant captured in Figure 3c, some nodes had been awoken by master, but the links had not been drawn in the screen. Finally, Figure 4 shows a representation of the botnet at a given time after initialization, when one of the nodes was "Asleep" (blue), another node was "Unreachable" (grey and disconnected from the botnet), and the rest are "Ready" (green). Any changes applied on the botnet will be immediately reflected on the visualization.

Previously, we concluded that discriminating between links based on relevance was quite ambiguous, especially due to the dynamic nature of the network. However, it is possible to visualize links differently based on the amount of traffic each link carries. In Figure 3b certain links are thicker than others, as they have recently passed the most recent command. Of course, at this scale and with this amount of traffic this link representation changes all the time and is not very practical, but on a larger scale, it could help the administrator identify certain common routes between the nodes.

### 3.3.2 Node-specific Context

The user can also directly interact with the visualization by using the mouse pointer. As seen in Figure

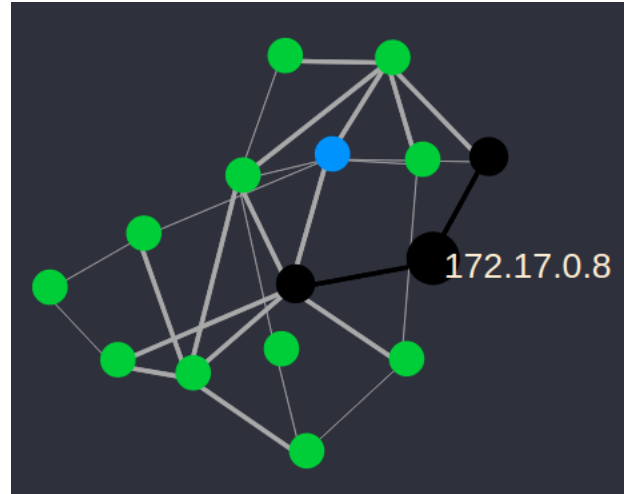


Figure 5: Selecting a node displays all the direct path of the node.

4, by placing the cursor on a node, the IP address of that node will be displayed, followed by an increase in the size of the node, as well as all its direct links. Moreover, by clicking on the specific node, the context will focus to the selected node by changing its color, as well as the color of its links and directly connected nodes to black, see Figure 5.

Although implementation for this part was not completed, including coordinated multiple views with an additional visualization would enhance the contextual information given to a selected node. In particular, we had thought of displaying a table with all the previously executed commands, as well as a line chart with the throughput of the node against time.

## 4 Evaluation

This section, we will evaluate our visualization using use cases and performance measures. This evaluation procedure was inspired by [1].

### 4.1 Performance measures

Both the botnet and the visualization were run simultaneously on a single laptop, running 64-bit Ubuntu 16.04 LTS with an Intel Core i7-4700MQ CPU and 8GB of RAM. Since the different components of the botnet and the visualization would not be run on the same host in a real setting, we discarded running CPU performance statistics. On the other hand, we focused on network statistics in particular the amount of incoming bytes on the hosts' docker interface, in other words, the bytes received by the master. We believe

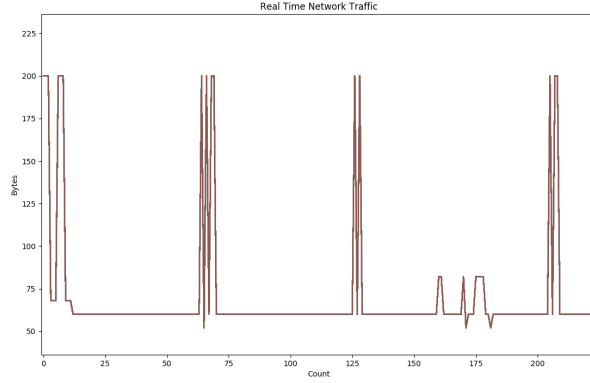


Figure 6: Traffic Flow of the botnet.

that this metrics provided more useful data to identify potential bottlenecks, and recognize patterns in network flow.

Figure 7 shows the byte flow received by the docker network interface over a period of time. There are four different peaks that reach the 200 bytes which correspond to a new node awakening and joining the rest of the botnet. Moreover, between the third and fourth peaks, we see a small fluctuations that barely reach 80 bytes. These corresponds to commands being sent by the master to a node. Finally, there seems to be a base load at 60 bytes. This corresponds to the continuous background update between the master and the nodes, vital to the accuracy and responsiveness of the visualization.

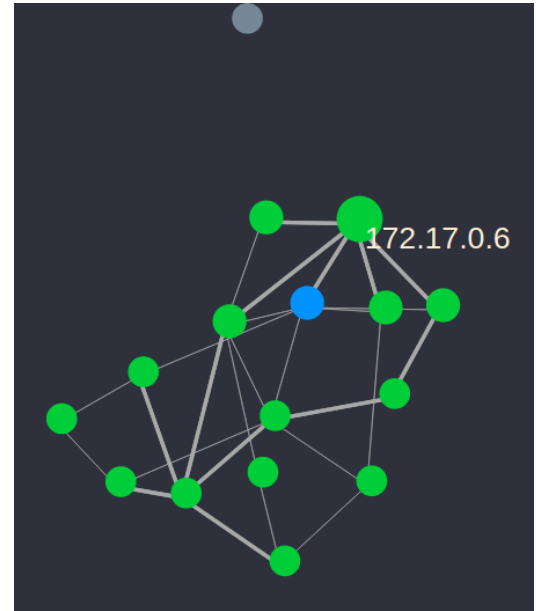
## 4.2 Use cases

The underlying purpose of this project was to create an accurate and responsive visualization of a distribute P2P network that would greatly simplify the cumbersome task of managing it. As seen in Figure 7a, the botnet terminal provided the state of the nodes and the topology of the botnet in a text format. Even with 15 nodes, which a relatively small botnet, it was already hard to get a clear picture of the topology from the terminal. Scale up to a 1000-node network and the topology in text format becomes useless.

From our experience, this tool makes the process of reacting to changes and having a clear and update topology much easier. In Figure 7, we contrast the difference in reading topology. There is clear added value in using tool on top of an existing P2P network. Additionally, it is proven to be valuable for non-experts in P2P network, as it provides a fairly intuitive picture of the state of the network.

```
(Cmd) show bots
##### LIST OF BOTS #####
##### Botnet Topology #####
172.17.0.14 -----> Ready
172.17.0.3 -----> Ready
172.17.0.2 -----> Ready
172.17.0.7 -----> Ready
172.17.0.13 -----> Ready
172.17.0.6 -----> Ready
172.17.0.4 -----> Ready
172.17.0.10 -----> Ready
172.17.0.11 -----> Ready
172.17.0.12 -----> Ready
172.17.0.9 -----> Ready
172.17.0.8 -----> Ready
172.17.0.10 -----> Ready
172.17.0.5 -----> Asleep
172.17.0.15 -----> Ready
(Cmd) show topo
##### Botnet Topology #####
172.17.0.14 is connected to ["172.17.0.8", "172.17.0.2", "172.17.0.15"]
172.17.0.3 is connected to ["172.17.0.12", "172.17.0.10", "172.17.0.8", "172.17.0.2", "172.17.0.11", "172.17.0.9", "172.17.0.6", "172.17.0.4"]
172.17.0.2 is connected to ["172.17.0.4", "172.17.0.3", "172.17.0.14", "172.17.0.7", "172.17.0.8", "172.17.0.5"]
172.17.0.7 is connected to ["172.17.0.16", "172.17.0.2", "172.17.0.10"]
172.17.0.13 is connected to ["172.17.0.12", "172.17.0.10"]
172.17.0.6 is connected to ["172.17.0.5", "172.17.0.15", "172.17.0.3", "172.17.0.7", "172.17.0.10"]
172.17.0.4 is connected to ["172.17.0.2", "172.17.0.3", "172.17.0.16", "172.17.0.9"]
172.17.0.10 is connected to ["172.17.0.4", "172.17.0.7"]
172.17.0.11 is connected to ["172.17.0.3", "172.17.0.12", "172.17.0.5"]
172.17.0.12 is connected to ["172.17.0.3", "172.17.0.11", "172.17.0.13"]
172.17.0.9 is connected to ["172.17.0.4", "172.17.0.3"]
172.17.0.8 is connected to ["172.17.0.14", "172.17.0.2", "172.17.0.3"]
172.17.0.16 is connected to ["172.17.0.3", "172.17.0.15", "172.17.0.6"]
```

(a) State of the network seen on terminal.



(b) State of the network seen on Visualization.

Figure 7: Dramatic improvement of real-time state of the network understanding.

## 5 Discussion

Overall the method we implemented of visualizing a Peer-to-peer botnet using contextual based navigation has potential to make understanding and detecting botnets easier. The ability to visualize which the status of the nodes, the connections they have to other nodes, and other node data such as ip address is very powerful for detecting patterns and developing solutions for defeating botnets. For example finding key nodes in the network with the most connections could aid in slowing down the spread of a botnet within an organization. In the future more richness could be added to the visualization to increase its efficacy further. Additionally, experiments to quantify how much more useful this visualization is over a command line interface would be valuable.

## 6 Conclusion

The threat presented by botnets in a technology driven society is great. In this paper we discuss the difficulty of both detecting and controlling the spread of such botnets. Studies have shown that visualizing these networks provides concrete advantage over command line or automatic tools. One particularly useful concept to creating an efficient tool for visualizing botnets is the idea of contextual based navigation which we used when developing a peer-to-peer botnet visualization tool. While already presenting many valuable improvements over the command line equivalent, in the future this tool could be developed further to provide even more powerful tools and visualizations.

## References

- [1] W. Allcock, J. Bester, J. Bresnahan, I. Foster, J. Gawor, J. A. Insley, J. M. Link, and M. E. Papka. Gridmapper: a tool for visualizing the behavior of large-scale distributed systems. In *Proceedings 11th IEEE International Symposium on High Performance Distributed Computing*, pages 179–187, July 2002.
- [2] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. A. Copeland, M. Ahamad, H. L. Owen, and C. Lee. Countering security information overload through alert and packet visualization. *IEEE Computer Graphics and Applications*, 26(2):60–70, March 2006.
- [3] B. Fink. 9 of history’s notable botnets.
- [4] C. C. Gray, P. D. Ritsos, and J. C. Roberts. Contextual network navigation to provide situational awareness for network administrators. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8, Oct 2015.
- [5] K. Junemann and J. Dinger. Ovlvis: Visualization of peer-to-peer networks in simulation and testbed environments. *IEEE Conference for Communication and Network Security*, 2008.
- [6] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, pages 42–49, June 2005.
- [7] M. Kumar. Rapidly growing electrum botnet infects over 152,000 users; steals \$4.6 million, 2019.
- [8] O. Tsigkas, O. Thonnard, and D. Tzovaras. Visual spam campaigns analysis using abstract graphs representation. *IEEE Symposium on Visualization for Cyber Security*, 2012.
- [9] R. Vamosi. Security experts visualize botnets with an eye toward defense, 2009.