Group Behavior Metrics for P2P Botnet Detection

John Felix, Charles Joseph[1], and Ali A. Ghorbani[2]

University of New Brunswick,
Fredericton NB, Canada
johnfelixc@gmail.com, ghorbani@unb.ca

Abstract. Botnet is becoming the biggest threat to the integrity of Internet and its resources. The advent of P2P botnets has made detection and prevention of botnets very difficult. In this paper, we propose a set of metrics for efficient botnet detection. The proposed metrics captures the unique group behavior that is inherent in bot communications. Our premise for proposing group behavior metrics for botnet detection is that, group behavior observed in botnets are unique and this unique group behavior property is inherent in the botnet architecture. The proposed group behavior metrics uses three standard network traffic characteristics, namely, topological properties, traffic pattern statistics and protocol sequence and usage to derive the proposed metrics. We derive six group behavior metrics and illustrate the efficiency of botnet detection using these metrics. It was observed that, group behavior metrics offers a promising solution for botnet detection.

1 Introduction

Malicious botnets has become a major security threat to the integrity of Internet [19]. A bot is an autonomous software agent which is programmed to perform some designated tasks automatically. A network formed by a set of bots residing in different hosts is referred to as a botnet. Though the concept of botnet was initially designed for benign purposes, its current usage in Internet serves for more malicious causes [3].

Peer to Peer (P2P) botnets [16] is new generation botnets which have replaced the old centralized IRC/HTTP based botnets [8]. P2P botnets are more stealthy and hard to detect. Due to the distributed and autonomous network structure of P2P systems, it is almost impossible to shutdown a botnet [6]. Attackers have become aware to strengths of P2P botnets and there has been steady increase in bot malwares that use P2P protocol for malicious botnets.

In this paper, we propose a set of metrics that capture group behavior among hosts to detect botnets. Our premise for proposing group behavior metrics for botnet detection is that, group behavior observed in botnets are unique and this unique group behavior property is inherent in the botnet architecture. As bots are software agents and follow a fixed protocol, their communication patterns are similar. In our work, we exploit this property of bot behavior to detect them. The proposed work uses topological properties, traffic pattern statistics and protocol based signatures for identifying hosts which have

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 93–104, 2012.

similar communication patterns. In evaluating the group behavior metrics for botnet detection, we found that the metrics deliver accurate and precise detection of bots in the traffic.

The paper is organized as follows. The succeeding section discusses the relevant work with respect to group behavior based botnet detection. Section 3 gives an overview on the inherent group behavior of botnets. The derivation of the proposed group behavior metrics is presented in section 4. The group behavior metrics is evaluated for accuracy on detecting P2P botnets in section 5. Section 6 concludes the paper by summarizing the contribution of the work and brief comments on future work.

2 Related Work

Botnet detection is a non-trivial problem [5]. Experience with respect to centralized IRC/HTTP based botnet detection and mitigation will prove that[19]. Now, the challenge of botnet detection has become harder due to the advent of P2P botnets. Research is yet to provide standard and efficient system for botnet detection [12]. Existing botnets detection methodologies suffer from the tactics used by attackers to thwart detection. Just like software, the bot malware is constantly updated and new revised versions are released, periodically. With P2P technology, this update process is distributed and autonomous, which make botnet resilient to detection and mitigation.

Existing work on group behavior based detection of botnets are very few [2,1,18]. Group behavior is often looked due to the intuitive belief that presence of groups of bots within the same subnet is highly unlikely. However, a look at the traffic through an ISP gateway will prove otherwise. Due to bot propagation mechanism, it is highly likely that more than few bots exist in the traffic of a subnet.

Choi.H et al [2] proposed BotGAD, a framework for capturing group activity in network traffic for botnet detection. They provided a comprehensive overview of current climate in botnet detection research and the usefulness of group behavior as a measure for botnet. Chang.S and Daniels [1] proposed a set of schemes to detect C&C channel of P2P botnets. In this work, the authors characterize a host behavior by jointly considering the spatial and temporal correlations within the traffic. These correlations essentially capture the group behavior of hosts within the network traffic. Hosseinpour and Borazjani [18] proposed a botnet detection framework that uses Artificial Immune System (AIS) to detect common network behavior in the traffic. This approach primarily focuses on detecting spam messages and port scan activity of infected hosts. Spamming and port scanning activities exhibits strong group behaviour among the infected hosts and this property of the malicious behaviour is used to detect them.

3 Group Behavior in P2P Botnets

The generic development cycle of a malicious botnet consists of three primary stages [13] , as shown in figure 1. First, the malicious bot is made to install on an end-user machine by various techniques such as, social engineering, spamming, etc. This process is referred to as *bot infection or initial infection*. In the second stage of botnet development cycle, the bot searches and connects to bots that reside in other infected hosts.

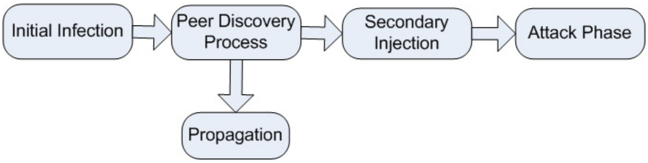

Fig. 1. Bot Development Cycle

Thus, the malicious botnet is formed. This stage also establishes a command and control (C&C) channel for the botmaster (attacker) to control the bot. Additionally, bots try to propagate itself to hosts in the infected host's network neighborhood. Bots are generally equipped with propagation mechanisms which will spread the bot malware to hosts which are connected to the infected host. At the third stage, the bot downloads infection vectors through C&C channel which will program the bot for future malicious tasks. This process is referred to as *secondary injection*. After the three stages, the bots and botnet is ready for malicious attacks controlled by the botmaster.

As mentioned earlier, group behavior among bots is inherent due to the botnet architecture. After the initial infection, each phase in the development cycle of the malicious botnet adds strong group behavior properties to the bot behavior.

Once a host is being infected with the malicious bot code, the bot tries to propagate itself to other hosts that are connected to the infected host. Through this process, the bot infects other neighboring hosts with the same bot code. As the hosts that are infected using the propagation mechanism are infected with the same bot code, the bots' network behavior is completely identical. However, this identical network behavior is often difficult to notice, as this behavior is hidden within the hosts' network traffic generated by other benign applications within the infected host. The proposed group metrics in this paper aim to capture the bots' identical network behavior that is hidden within the infected hosts' benign traffic.

In the second stage of botnet development cycle, the malicious bot installed in the infected host tries to connect to other bots (peers). This process in P2P terminology is referred to as peer discovery process [11]. This peer discovery process causes a bot to exhibit strong group behavior with respect to common network connectivity. In most bots, the peer discovery process starts by trying to connect to a set of peers whose IP addresses are hard-coded within the bot code. This property causes the bots to have high common connectivity, as bots infected with the same malware will connect to the same list of peers. Even between different versions of bot malware, large number of peers in the hard-coded peer list remains unchanged. After the peers in the hard-coded peer list are connected, the bot downloads a list of active peers in the botnet through the successfully connected peers. This downloaded peer list is almost the same for different bots in the botnet. This further strengthens the common network connectivity among the bots. Thus, the similarity between the network topology among bots is inherent. If the attacker tries to hide this similarity by randomizing and sub-grouping the peer list, thus formed botnet will be disconnected and hard to manage for the attacker. Thus, similarity in network topology is key feature for detecting group behavior in bots.

During the attack phase of a botnet, bots exhibit strong group behavior. This is primary due to the fact that attacks are coordinated using a set of bots. For example, Denial of Service (DoS) attacks using bots are usually coordinated using a set of bots. Hence, bots tend to behave in the same fashion. Additionally, the command and control channel is not unique to a bot. Due to the propagation mechanism of P2P protocols, there is a high probability of bots in the same subnet to receive the same attack commands. In P2P networks, such as eDonkey, local peers are preferred over distant peers for propagation. This property of the botnet system make the bots to exhibit strong group behavior in terms of network connectivity, traffic pattern and protocol sequence and usage.

4 Group Behavior Metrics

The group behavior metrics for hosts in the network are derived using three network traffic characteristics, namely, topology, traffic pattern and protocol usage. The process of deriving the group behavior metrics from the network traffic is illustrated in figure 2. The process comprises of five stages. In the succeeding sections, we discuss each stage of deriving the group behavior metrics.

For each of three network traffic characteristics, we use features that capture group behavior in network behavior. Common connectivity among hosts is derived from the topological properties of the network and is used for capturing the group connectivity. Similarity in packet sizes and frequency is used to measure the group behavior in traffic patterns. Similarity in protocol sequence exhibited by hosts in their network traffic is used to measure the group behavior in protocol usage. Thus, the process uses the three primary characteristics of network traffic to derive the group behavior metrics.

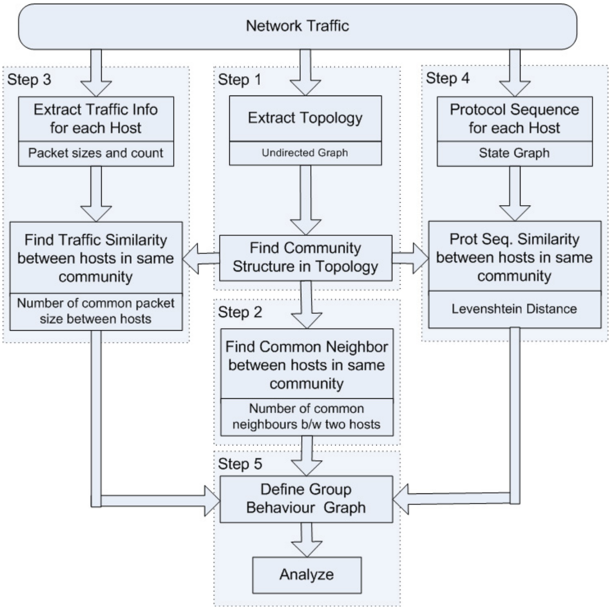
Network Traffic
Step 3
Extract Traffic Info for each Host
Packet sizes and count
Step 1
Extract Topology
Undirected Graph
Step 4
Protocol Sequence for each Host
State Graph
Find Traffic Similarity between hosts in same community
Number of common packet size between hosts
Find Community Structure in Topology
Prot Seq. Similarity between hosts in same community
Levenshtein Distance
Step 2
Find Common Neighbor between hosts in same community
Number of common neighbours b/w two hosts
Step 5
Define Group Behaviour Graph
Analyze

Fig. 2. Group Behavior Metrics

4.1 Topological Properties

In the first step, the topology of the network that is represented in the network is extracted, as shown in figure 2. A directional graph $G_{IP}(V, E)$ is used to represent the topology, with vertices (V) being the set of unique host IPs found in the network traffic and edges (E) represent the communication links between source IP and destination IP found in the packet header. Similarly, a directional graph $G_{IP/PORT}(V, E)$ is used to represent the topology that considers both IP address and port numbers. The vertices in graph $G_{IP/PORT}(V, E)$ is a set of unique host IP address and port number pairs found in the network traffic and edges (E) represent the link between source IP / source port and destination IP / destination port.

After the topology graphs $G_{IP}(V, E)$ and $G_{IP/PORT}(V, E)$ are defined, each topology is divided into groups based on the connectivity between the hosts. Sub-graphs are defined for each graphs $G_{IP}(V, E)$ and $G_{IP/PORT}(V, E)$. These sub-graphs are formed by considering the connectivity of hosts in the graph, such that strongly connected hosts are formed into sub-graphs. The process of deriving this sub-graphs is discussed later in this section.

The purpose of dividing the topology into sub-graphs is to reduce the complexity of deriving the group behavior metrics. Without grouping, deriving the group behavior metric for each host with all remaining hosts in the topology is almost impossible due to computational complexity. In this case, for deriving the group behavior metric, each host's network behavior has to be compared with the remaining hosts in the network. Such a process has computational complexity in the order of $O(N^2)$, where N is the number of hosts in the network. By dividing the topology into groups, the group behavior is evaluated for hosts only within the sub-graphs of the topology. This reduces the complexity of the group behavior metric computation, significantly.

To find the groups of strongly connected hosts in the topology, we use community detection algorithm [4]. The fundamental idea behind community detection algorithms is that, the nodes of a network can be formed into groups based on the connectivity between them. Newman [10], in his seminal work proposed the notion of modularity which is used as a measure to group nodes in a network. Modularity is a benefit function which quantifies the quality of grouping a certain set of nodes in the network based on connectivity. Modularity [10] is high for set of nodes which have a high degree of connectivity between them but less to nodes with few connections. Hence, in other words, modularity aims to maximize the number links within the group and minimize the links between the groups. We use the community detection algorithm proposed by Schuetz and Caflisch [14].

Community detection algorithm aim to group strongly connected nodes. The outcome from the community detection algorithm is a group index to each host in the topology. The nature of strong connectivity among bots will be preserved by the community structure detection algorithm. In community detection terminology, groups or clusters are referred to as communities. Therefore, hereinafter, the terms communities and groups will be used interchangeably.

At stage two, the group behavior within the topological properties of the network is evaluated. For each graph, $G_{IP}(V, E)$ and $G_{IP/PORT}(V, E)$, the common connectivity [9] of nodes is derived by computing the number of common neighbors between two hosts within the community. The common neighbor for every node pairs in the subgraphs is computed. The property of bot to have high common connectivity is captured using this metric.

4.2 Traffic Pattern Statistics

As discussed in section III, bots exhibit a common traffic pattern. However, this traffic pattern similarity is often hidden within the network traffic generated by the benign applications in the infected host. Typical features of network traffic statistics [15] include,

- Aggregated number of incoming packets
- Aggregated number of incoming bytes
- Aggregated number of outgoing packets
- Aggregated number of outgoing bytes

The above four features are most common used traffic statistics features in existing detection systems. It should be noted here that, the four features represent only the incoming and outgoing bandwidth of a specific host communication. Traffic patterns, however, cannot be perceived using the above four features. Furthermore, the aggregation of packet and byte count for host traffic allows the bot traffic properties to be hidden within the benign traffic. Thus, the conventional features of traffic statistics are inadequate for deriving group behavior.

For representing the traffic pattern, we primarily use packet size feature of the network traffic. At stage three, we extract the traffic information for each host in the network. For each host, we record different packet sizes that are observed within the host's network traffic. Additionally, the frequency of packet sizes within the host's communication is also extracted. Hence, for i^{th} host in the network, the traffic information is represented as $\left(P^I_E, F^I_E\right)$, where P^I is the set of packet sizes observed for the i^{th} host, F^I is the set of frequency of corresponding packet size and S is the number of unique packet sizes observed within the host's network traffic.

After the traffic pattern is extracted, the common traffic pattern is evaluated for every two hosts within every group in the topology. Similarity in traffic pattern using the packet size representation is computed as follows:

$$P^{I,J}_{Common} = P^I \cap P^J \;\; \forall I \in C \;\; and \;\; \forall J \in C \tag{1}$$

$$F^{I,J}_{Common} = \sum_{K=P_{Common}} min\left(F^I_K, F^I_K\right) \tag{2}$$

Equation 1 finds common packet sizes observed between i^{th} and j^{th} host in the topology community C. The number of packets $\left(F^{I,J}_{Common}\right)$ between i^{th} and j^{th} host that have similar packet size is computed in equation 2. The two features namely, $F^{I,J}_{Common}$ and number of elements in $P^{I,J}_{Common}$ are used to represent the group behavior in traffic pattern between the i^{th} and j^{th} host. Similarly, traffic pattern similarity is computed for all hosts pairs within the different topology communities.

4.3 Protocol Sequence Signature

Protocol usage is an important property for botnet detection. This feature is more important in group behavior metrics, as the protocol among the bots is fixed and the protocol sequence exhibited by bot traffic is highly similar. This phenomenon in bots is further illustrated in the results section of the paper, where we discuss the efficiency of this particular metric. Protocol sequence of a host represents the different protocols and protocol states of the host network behavior and also the various protocol state transitions exhibited by the host's network communications.

At stage four, the protocol usage and sequence is extracted from the network traffic. First, we identify the protocol of each packet in the network traffic by using wireshark's protocol dissectors. The wireshark's protocol dissectors identifies the protocol and also the type of protocol message, for example, HTTP get request, eDonkey protocol's kademlia hello request/response, etc. The wireshark dissectors searches the packet payload for keywords to identify protocol message type of the packet. For each packet, the protocol state is defined in the following format: *<Network Layer Protocol>* . *<Transport Layer Protocol>* . *<Application Layer Protocol>* . *<Application Layer Message Type>* . *<Application Layer Message sub-type>*. Hence, a protocol state definition looks like "$ip.tcp.edonkey.helloreq$".

After protocol analysis, for each host, the sequence of protocol communication is captured in a state graph. For every i^{th} host in the network, a state graph (S^I) represents the protocol sequence of host's network traffic. In the state graph, the different unique protocol states observed in the host's traffic is defined. The sequence of protocol usage is represented as state transitions in the state graph.

With the protocol sequence state graphs defined for all hosts in the network, we then compute the similarity in protocol sequence between every two hosts in a network topology community. The similarity between state graphs of two hosts is used to compute the common behavior in protocol sequence and usage. For measuring the similarity between two state graphs, we use two different similarity measures, namely, Levenshtein distance [17] and Jaccard similarity [7]. Levenshtein distance is most commonly used approach for comparing DNA sequences in bio-sciences. The distance measure computes the number of minimum steps necessary to change a graph/sequence A to another graph/sequence B. The computed number of steps represents the Levenshtein distance between graph A and B. The Jaccard similarity is a more generic similarity measure that computes the ratio of number of common elements between graph A and B with the total number of unique elements in A and B.

$$PS^{I,J}_{Levenshtein} = L\left(S^I, S^J\right), \forall I \in C \;\; and \;\; \forall J \in C \tag{3}$$

$$PS^{I,J}_{Jaccard} = \frac{\left(S^I \cup S^J\right) - \left(S^I \cap S^J\right)}{S^I \cup S^J} \forall I \in C \;\; and \;\; \forall J \in C \tag{4}$$

Equation 3 computes the Levenshtein distance between protocol sequence state graphs of every i^{th} and j^{th} host in the topology community C. The algorithm for computing the Levenshtein distance function $L()$ can be found here [17]. Similarly, equation (4) computes the Jacobian similarity measure between state graphs and of every i^{th} and j^{th} host in the topology community, respectively.

4.4 Group Behavior Graph

The metrics derived from the above three group behavior properties is used to define a group behavior graph. The features used include,

- $CN_{IP}^{I,J}$– Number of Common Neighbors between two hosts in graph $G_{IP}(V, E)$
- $CN_{IP/PORT}^{I,J}$– Number of Common Neighbors between two hosts in graph $G_{IP/PORT}(V, E)$
- $P_{Common}^{I,J}$– Number of Similar Packet Size
- $F_{Common}^{I,J}$– Frequency of Similar Packet Size
- $PS_{Levenshtein}^{I,J}$– - Levenshtein Distance between protocol sequence
- $PS_{Jaccard}^{I,J}$– Jaccard Similarity measure between protocol sequence of two hosts.

Host pairs, which have non-zero values for all the above six group behavior features are added to group behavior graph. That is, if the derived six group behavior metrics are non-zero for I^{th} and J^{th} host, then the host I and J are added to the group behavior graph as vertices and the added vertices are connected using an edge. Thus, all hosts which exhibit strong common behavior are captured in the group behavior graph.

In order to filter hosts which exhibit benign group behavior, we define a threshold (T) for each of the above group behavior features. Hence, hosts which has group behavior feature values below the threshold are removed from the group behavior graph. We propose to train the threshold value for the six group behavior features using known bot group behavior. In the next section, we illustrate that finding the threshold is not difficult and can be statically defined.

After the threshold based filtering, the group behavior graph consists only of infected hosts and the botnet topology is represented by this graph.

5 Results and Evaluation

In this section, the proposed group behavior metrics is evaluated for accuracy in detection of botnets. Furthermore, the properties of the observed group behavior with respect to the three network traffic characteristics, namely, topology, traffic pattern and protocol usage are discussed.

The results using the group behavior metrics is summarized in Table 1. The threshold for filtering hosts in group behavior graph is trained as a simple Bayesian classifier. The trained threshold is listed in Table 1.

5.1 Experimental Setup

Among our research community, real botnet traffic is a scarce resource. Due to the sensitive nature of the content in network traffic traces, ISPs are reluctant to share their traffic captures. It is even more difficult to obtain network traffic traces that contain few bots in the traffic. To evaluate the efficiency of group metrics, we needed a traffic data that has few bots in the traffic. Thus, we had to build our own network traffic data.

Table 1. Detection Accuracy

<table>
<tr><td colspan="3">Number of Hosts in Benign Traffic</td><td colspan="3">42456</td></tr>
<tr><td colspan="3">Number of Hosts in Benign Traffic with Group Behavior</td><td colspan="3">8556</td></tr>
<tr><td colspan="6"><b>Threshold for Group Behavior Graph</b></td></tr>
<tr><td>$CN^{I,J}_{IP}$</td><td>$CN^{I,J}_{IP/PORT}$</td><td>$P^{I,J}_{Common}$</td><td>$F^{I,J}_{Common}$</td><td>$PS^{I,J}_{Levenshtein}$</td><td>$PS^{I,J}_{Jaccard}$</td></tr>
<tr><td>143</td><td>59</td><td>15</td><td>2964</td><td>62</td><td>0.7</td></tr>
<tr><td colspan="3">Bots used for Training Threshold</td><td colspan="3">6</td></tr>
<tr><td colspan="3">Bots used for Testing</td><td colspan="3">4</td></tr>
<tr><td colspan="3">Detected Bots</td><td colspan="3">4</td></tr>
<tr><td colspan="3">Detection Accuracy</td><td colspan="3">100%</td></tr>
</table>

Initially, we collected from various sources, bot traffic captures from different versions of the same bot malware. We were able to collect network traffic generated by 10 different versions of Stormbot [13]. The malware network traffic is captured using honeypots which ran different versions of the Stormbot.

To create the network setup, we use traffic captured from an ISP's gateway. The captured network traffic is real-world traffic data, thus, gives a realistic network setup. We select the 10 IPs from the ISP traffic data and map the IP addresses to the 10 different Stormbot attack traffic data. Once the IP address in the attack traces are modified, we merge and synchronize the 10 attack traffic data with the ISP traffic dataset. Now, the merged network traffic data comprises of 10 bots that run different versions of Stormbot. Using this network data, we evaluate the efficiency of botnet detection using group behavior metrics.

For testing the detection accuracy of the proposed group behavior metrics, two network traffic datasets are built using the above technique. The first traffic dataset comprises of 6 bots within traffic, that is merged using 6 bots and ISP traffic data. ISP traffic data acts as the background traffic for the network setup. This dataset is used for training the threshold (T) that is used to filter the group behavior graph. Similarly, the second dataset, comprises of 4 bots within traffic. This dataset is used for testing the detection accuracy of the proposed group behavior metrics.

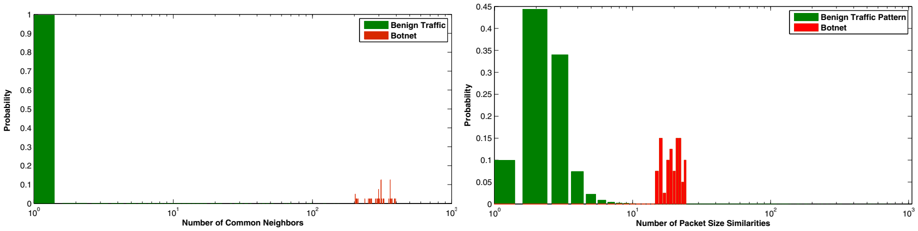

Fig. 3. (a) Group Behavior in Topology (b) Group Behavior in Traffic Pattern

5.2 Group Behavior in Topology

In this section, we discuss the topology properties that were observed in normal topology and P2P botnet topology. Analyzing the structure of groups identified by the community detection algorithm, it was found that the topological properties within and between groups of normal and P2P botnet topology differs significantly.

In normal topology, communities are evenly and sparsely connected, whereas, in botnet topology, communities are strongly connected with many intra community links. Density of links within communities is relatively high in P2P botnet topology, whereas, in normal P2P, hosts and links are evenly distributed within communities.

The most notable observation in the group structure is that, in normal topology, the hosts are connected in one-to-many connectivity configuration (tree structure) within the community. In most cases, communities have one or two central host to which all other hosts in the community are connected. Due to the above intra community structure, most of the hosts within the community have one common neighbor. On the other hand, in P2P botnet communities, the hosts are connected in many-to-many connectivity configuration. Botnet communities are very strongly connected. Due to many-to-many connectivity, infected hosts have many common neighbors (mostly > 150 and < 400). This is shown in Figure 3a. It can be clearly observed from Figure 3a, that number of common neighbors observed is different between normal topology and P2P botnet topology. In normal topology, 99.56% of hosts in the network have less than two common neighbors. Whereas, in botnets, number of common neighbors ranges between 236 and 396.

Thus, the number of common neighbors between hosts found within the community is efficient to be used for botnet detection and it is key feature describing the group behavior of hosts.

5.3 Group Behavior in Traffic Pattern

Packet size and frequency of packet size within traffic of a specific host is used in our approach to represent the traffic pattern. This is a unique way of representing the traffic pattern and these features truly captures the traffic pattern of network behavior. In this section, we discuss the efficiency of using similarities between packet size and frequency of packet size to compute group behavior of hosts for botnet detection.

Figure 3b shows the unique packet size similarities observed between hosts which exhibit group behavior. It can be observed that, in benign traffic, number of similar packet size ranges between 0 and 9. Whereas, the infected hosts in the botnet show high similarity in packet size which range between 16 and 24. In other words, the number of similar packet size between bots is between 16 and 24. Therefore, there is a clear distinction between packet size similarities between benign hosts and infected hosts of the botnet. This distinction is captured in this metric and used for botnet detection. Similar distinction in similarity was also observed over the second group behavior metric for traffic pattern - frequency of packet size.

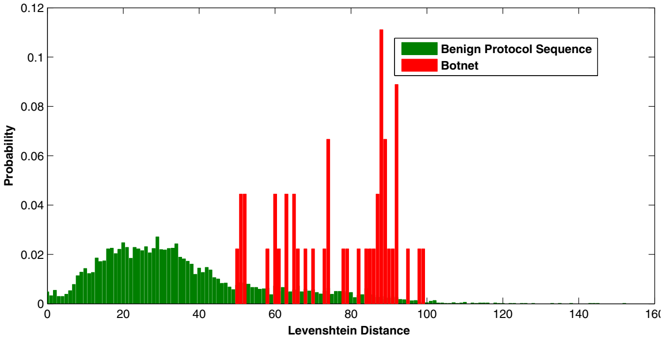

Fig. 4. Group Behavior in Protocol Sequence

5.4 Group Behavior in Protocol Sequence

Using protocol sequence as a feature for botnet detection is an unprecedented approach. As mentioned earlier in the paper, the Levenshtein distance and Jaccard distance is used to represent the protocol sequence similarity between hosts in the network community. Figure 4 shows the Levenshtein distance observed between protocol sequence of benign hosts and botnets.

Similarity in protocol sequence between bots and benign hosts are not completely different. The probability distribution of Levenshtein distance observed from figure 4 shows that the two distributions overlap. Hence, protocol sequence similarity is not a distinct metric as observed in group behavior metrics derived using topology and traffic pattern characteristics. However, though distributions overlap in Figure 4, the distributions are not completely similar. The center of distributions lies far apart. Protocol sequence similarity measure is still efficient for botnet group behavior detection.

Inferring from Figure 4, Levenshtein distance observed for protocol sequence between infected hosts range between 49 and 98. Within this range, 22% of the hosts that exhibit group behavior in benign traffic are observed. As 21% of the uninfected hosts in the network exhibit group behavior, the false positives using only this metric for botnet detection is $\tilde{4}$ % and the true positives is 100%. This illustrates the strength of group behavior metrics for botnet detection. With better protocol analysis techniques, the false positives can still reduce further.

6 Conclusion

In this paper, we have presented a set of group behavior metrics which are efficient for botnet detection. The property of bots to exhibit similar communication patterns is exploited to derive these metrics. Three network properties, namely, topological characteristics, traffic statistics and protocol usage sequence is used to derive the group behavior for each host in the network. It is observed that, group behavior of bots is distinctly captured by these metrics.

References

1. Chang, S., Daniels, T.E.: P2p botnet detection using behavior clustering & statistical tests. In: Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence, pp. 23–30. ACM (2009)
2. Choi, H., Lee, H., Kim, H.: Botgad: detecting botnets by capturing group activities in network traffic. In: Proceedings of the Fourth International ICST Conference on Communication System Software and Middleware, pp. 2:1–2:8. ACM (2009)
3. Dagon, D., Gu, G., Lee, C.: A taxonomy of botnet structures. In: Botnet Detection, vol. 36, pp. 143–164. Springer US (2008)
4. Fortunato, S., Castellano, C.: Community structure in graphs, pp. 1141–1163 (2009)
5. Grizzard, J.B., Sharma, V., Nunnery, C., Kang, B.B., Dagon, D.: Peer-to-peer botnets: overview and case study. In: Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets, p. 1. USENIX Association (2007)
6. Ha, D.T., Yan, G., Eidenbenz, S., Ngo, H.Q.: On the effectiveness of structural detection and defense against p2p-based botnets. In: IEEE/IFIP International Conference on Dependable Systems Networks, pp. 297–306 (2009)
7. Honov, S.A., Ivchenko, G.I.: On the jaccard similarity test. Journal of Mathematical Sciences 88(6), 789–794 (1998)
8. Kang, B., Nunnery, C.: Decentralized peer-to-peer botnet architectures. Advances in Information and Intelligent Systems 251, 251–264 (2009)
9. Choi, S., Kang, Y.: Common Neighborhood Sub-graph Density as a Similarity Measure for Community Detection. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009, Part I. LNCS, vol. 5863, pp. 175–184. Springer, Heidelberg (2009)
10. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 69(62), 066133-1–066133-5 (2004)
11. Rossi, D., Sottile, E., Veglia, P.: Black-box analysis of internet p2p applications. In: Peer-to-Peer Networking and Applications, pp. 1–19 (2010)
12. Van Ruitenbeek, E., Sanders, W.H.: Modeling peer-to-peer botnets. In: Proceedings of the 2008 Fifth International Conference on Quantitative Evaluation of Systems, pp. 307–316. IEEE Computer Society (2008)
13. Stover, J.H.S., Dittrich, D., Dietrich, S.: Analysis of the storm and nugache trojans: P2p is here (2007)
14. Caflisch, A., Schuetz, P.: Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 77(4) (2008)
15. Strayer, W., Lapsely, D., Walsh, R., Livadas, C.: Botnet detection based on network behavior. In: Botnet Detection, vol. 36, pp. 1–24. Springer US (2008)
16. Wang, P., Wu, L., Aslam, B., Zou, C.C.: A systematic study on peer-to-peer botnets. In: International Conference on Computer Communications and Networks, pp. 1–8 (2009)
17. Bo, L., Yujian, L.: A normalized levenshtein distance metric. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 1091–1095 (2007)
18. Borazjani, P.N., Zeidanloo, H.R., Hosseinpour, F.: Botnet detection based on common network behaviors by utilizing artificial immune system(ais) 1, V121–V125 (2010)
19. Kadobayashi, Y., Zhang, Z.: A holistic perspective on understanding and breaking botnets: Challenges and countermeasures. Journal of the National Institute of Information and Communications Technology 55(2-3), 43–59 (2008)