Botnet Detection and Mitigation in P2P networks

Abhishek Pratap Singh
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati
abhishek.pratap@iitg.ac.in

Prosenjit Biswas
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati
prosenjit.biswas@iitg.ac.in

***Abstract*—P2P botnets are widely used for a variety of malicious activities. In this paper, we have listed a variety of approaches to detect P2P botnets. We begin by stating, how a P2P botnet is formed. After that a survey of various botnet detection techniques is given. The paper includes countermeasures that should be taken after detection of botnet.**

***Index Terms*—P2P botnet, Detection**

I. Introduction

Bots are the systems which can be remotely controlled and commanded by its owner. A botnet is a network of such compromised bots. The huge size of modern botnets combined with decentralized control allow them to carry out operations on large scales. They are used for activities such as spamming, cryptocurrency mining, Distributed Denial of Service (DDoS) attacks. [1]

Botnets may have a centralized or distributed architecture for Command and Control (C&C) communications. Centralized architecture suffer from problem of single point of failure due to single bot-master [1]. Therefore, we have seen a shift towards distributed, decentralized, P2P based architectures for botnets.

P2P botnets are resilient to dynamic churn (i.e. peers joining and leaving the system at high rates). Their communication is not disrupted on losing a number of bots. A bot in a P2P network can act as both C&C server and client. [2]

II. Botnet attack and Life cycle

A. Botnet attacks

Botnets are being used for a variety of attacks, with common ones being listed below:

- **DDos attacks:** DDos attacks involves sending requests from all the bots in the botnet. With increasing size of botnets, the number of requests could be in multimillions. This prevents the server from servicing legitimate request, thus denying access to legitimate users.
- **Spamming and Spreading Malware:** Botnets are increasingly used for spamming and spreading malware through emails. According to an estimate, around 70% to 90% of the world's spam is caused by botnets nowadays. It is also easy to distribute malware through numerous bots in a botnet.
- **Information leakage:** Systems which are infected as bots, can easily leak sensitive information. A infected system can record all sensitive keyboard inputs like usernames and passwords send those to botmaster.
- **Click Fraud:** By using botnet, perpetrators are able to install advertisement add-ons. They can use the botnets to periodically click on a specific hyperlink and increase its click through rate (CTR). CTR is a metric used used for calculating income from advertisement.
- **Identity fraud:** Botnets are being used for phishing mails.

B. Botnet life cycle

Construction of botnet can be divided into two steps. In the initial stage, the focus is on infecting as many systems as possible. The next step is forming a botnet by all of these infected systems [2]. We will discuss these steps under the following headings:

B.1 Selection of botnet candidates

In P2P network with absence of centralised authority, one cannot guarantee that the file being exchanged is not malicious. Generally attackers target existing hosts in a P2P network. When a particular peer in a network is compromised, it tries to infect others in its "hot-list". This hot-list generally includes other peers contacted before or peers who responded after a file search query.

How large a particular will go to depends on how many vulnerable hosts are present in the P2P network. Now P2P botnets don't confine themselves to existing P2P networks, but recruit members in the entire internet.

B.2 Forming a botnet

After a particular host is compromised, it needs to join the botnet. This is generally accomplished in two ways as listed below:

1) Every infected bot will have a hard coded peer list. It needs to connect to any of these peers to become part of a botnet. This list is made available in the bot binaries which is used for infecting a system.
2) By using a shared web cache. The location of the cache is put in the client code. Any new peer can refresh its neighbouring peer list by going to the web cache and fetching the latest updates.

This process of finding and joining a P2P network is usually called "bootstrap" procedure. This bootstrap procedure has also become the bane of botnets. As both methods involve querying a finite list of addresses, one can easily block them to stop the formation of botnets. Recently botnets have evolved to deal with this single point of failure vulnerability. They

have started using a procedure where where when a bot A compromised a vulnerable host B, A passes its own peer list to this newly infected host B, and B will add A into this neighbouring peer list [2].

III. Detecting P2P botnets

Some previously used and now obsolete methods for detecting P2P bots [3]:

- **Open Ports:** Earlier P2P bots used a particular port or a list of hard coded ports. One can easily monitor these port's traffic to decipher if a bot is using it or not. Bots can also use ports used by other applications, in this case we would be needing additional information to take action.
- **Connection Failures:** Just after a bot is initialised, it tries to connects to all available addresses in it's list. Therefore, there is a high rate of connection failures observed. To deal with this situation, botmasters used special hosts called supernodes or trackers. But these supernodes also act as single point of failure. Taking down a single supernode will put out many bots from the system.
- **Peer discovery:** New peers need to know which host to connect to. Some use a static list of IP addresses and some use a shared web cache. Putting these IP's under surveillance will help us know which bots tried connecting to it. We can use this information to put down a botnet.

Initial methods of detection of P2P botnets involving port-based techniques or signature-based mechanism are presently easily deceived. Botnets can easily avoid this by randomized port selection. For signature based mechanism bots have started using obfuscation techniques such as padding and encryption. Recent methods have shifted to machine learning based approaches [1].

A. Botnet detection based on P2P traffic similarity

The authors in this method [4] have used a 2-stage detection method. In the first stage, we filter P2P and non-P2P traffic. The second stage involves utilizing session characteristics to extract features which will help in identifying P2P botnet.

1) **First stage: Non-P2P traffic filtering -** This method uses non-P2P well-known port filtering mechanism, DNS query, flow counting rules to filter non-P2P traffic.
2) **Second stage: P2P botnet identification -** This stage is mainly concerned with extracting meaningful features which can be used in classification algorithm. The paper uses session-based strategy for feature extraction. For botnet the duration of the flow is generally short and fixed. Therefore, session duration is used as a feature. A difference in the distribution of traffic in the session for normal P2P network traffic is observed when compared to traffic in the session of a P2P botnet traffic. Therefore, distribution of flow in the session is also used as features.

The list of classification algorithms used:

1) Naive Bayes
2) Decision Tree
3) ANN

They used the dataset of CTU-13 [6] project to run experiments on. The detection rate achieved for Naive Bayes algorithm was 75.5%, for ANN it was 93.8%. The highest detection rate of 94.4% was achieved with Decision Tree algorithm.

B. Adaptive Multi-layer botnet detection technique

The authors of paper [5] have used a multi-layer approach. The initial layer is mainly concerned with feature extraction. To make this process less resource intensive, firstly traffic reduction is done, then feature extraction. The second layer involves classifying captured traffic into non-P2P and P2P traffic. The third layer is concerned with feature reduction. This is done by removing those features which have marginal impact on final classifier output. The final layer is where the classification for normal P2P or botnet traffic happens.

1) **First Layer: Traffic reduction -** With bots increasingly using encryption for payload and encapsulating protocols, Deep packet inspection (DPI) is not suitable for our need. DPI is also resource intensive. Therefore the authors of this paper have decided to use only Transmission Control Protocol (TCP) packets. This also reduces the volume of network traffic.
2) **Second Layer: P2P and Non-P2P traffic classification -** Port identification method will not work with random or custom ports, DPI does not recognise encrypted P2P traffic. This method uses non-P2P well-known port filtering mechanisms, DNS query, flow counting rules to filter non-P2P traffic. This is combined with fast heuristic P2P traffic identification method.
3) **Third layer: Feature extraction and Feature Reduction -** All features are obtained explicitly from the control packet header via a thorough check of the payload material of the packets. The methods also uses feature reduction which helps in reducing the overfitting. For feature reduction, a classifier is used to identify those features which are less important for final classifier outcome. Those features are removed, this also reduces the amount of data.
4) **Fourth Layer: P2P traffic classification -** From previous data flow characteristic analysis of P2P botnet, no difference in traffic characteristics of benign networks and botnets was observed. Therefore, the author's have used session-based strategy for feature extraction. This helped in improving the detection efficiency. One observation regarding duration of flow of bots was that the flow is generally short and fixed. As a result, session duration was extracted as a feature.

The datasets used are CTU-13 dataset [6] and ISOT dataset [7]. The authors have used Decision Tree classification algorithm. They achieved a detection rate of 98.7%.

C. Entelecheia: Detecting P2P botnets in their waiting stage

Entelecheia [8] adopts a different approach from previous methods where they considered 5-tuple flows. In this paper they considered 2-tuple flows. This method is able to detect botnets in the waiting stage itself. The detection strategy works on the basis of two bots' characteristics:

- P2P bots in wait stage are observed to maintain long-lived flows in anticipation of commands,
- Bots only exchange bot commands as traffic, thus have only low-volume traffic be exchanged.

The dataset used was WIDE dataset from MAWI Traffic archive. The accuracy on the dataset was 98.1%. However, one major drawback of this approach is that it won't be able to detect bots which use obfuscation bits to get comparatively high volume flows.

D. Detection scheme based on decision tree and adaptive Multi-layer NN

This method [9] works by slightly monitoring network traffic. It takes advantage of the fact that bots during their distribution phase will demonstrate regular communication behavior with their C&C servers / peers in order to find other peers and get the latest activity updates due to their pre-programmed environment
Steps involved in the proposed method:

- **Network traffic reduction:** This is an important step as it helps to control large amounts of network traffic in the event that resources are limited (for example Hard disk). DPI is statistically expensive and does not apply to the signature of an unknown payload. Here we select only TCP control packets. This filtering process is done in two steps. In the first step, TCP-related traffic traffic is filtered, and TCP control packets are released.

- **Feature Extraction:** The quality of the features determines the outcome of our dividers. Features are extracted based on the contact description as a group of switches switched between two different hosts, identified by 4-tuple (source IP address, destination IP address, source port and destination port). These features are extracted from control pack titles, no DPI required. This increases the speed as the calculator is used extensively by resources.

- **Feature Reduction:** It is used to eliminate those features that have a small effect on the final release of our separator. The Classification and Regression Tree (CART) is used as a factor reduction factor used to eliminate useless elements. Principal Component Analysis (PCA) is also used to reduce the feature. PCA reduces the first number of elements into a small number of unrelated elements, which are counted as a linear combination of the real ones.

- **Neural Network:**The method uses the neural network as a classifier. The neural network proposed in this route has 10 units in the input layer and 2 units in the output layer. To determine the number of hidden layers and the number of neurons in the hidden layers, use the method proposed in this study [10].

The authors have tested their method over two datasets: ISOT [7] and ISCX [11] dataset. They were able to achieve accuracy of 99.1% and detection rate of 99.09%.

E. Malicious Fast Flux Network Identification

The modern P2P botnets have become advanced with adoption of Fast Flux Service Networks (FFSNs). FFSNs help the botnet to be more resistant to discovery and counter measures for detection. FFSNs also improves availability and allows for swapping of IP addresses. FFSNs can be malicious as well as non-malicious.
The paper [12] tries to identify botnet by identifying malicious fast-flux networks. This is done by collecting features from TCP/UDP level network behaviour of botnets. The detection framework works in the following stages:

- **Filtering stage:** Filtering is essential so that we can get a reduced subset of domains for which we can capture data for extended period of time. For this elimination of non-malicious domains from our list of domains need to be done. For this purpose, the authors have used a rule-based detector. This detector collects several domain attributes over a period of one week. These attributes are then analysed for malicious behaviour. Any malicious domain found is then put under monitoring for extended period.
- **Data Capture Stage:** Malicious domains from previous stage are used here for data capturing. This is done by monitoring them for extended period of time. The process involves polling the DNS records and then collecting the response. This data is then used for feature extraction for our classification model in next stage.
- **Classification model:** Decision tree classifier is used for classification and clustering of data captured.

DETER testbed has been used for generation of botnet traffic and user-induced dataset contacting 5054 destinations was used. The system detected malicious nodes with 97% accuracy.

F. Reinforcement learning-based Botnet detection approach

The proposed method [13], involves passive monitoring of network traffic during the frequent communication between bots and their C&C servers during the propagation phase. The method works in following stages: network traffic capture and packet reduction, feature extraction, malicious activity detection, and bot behaviour detection using reinforcement

learning.

- **Network traffic capture and packet reduction**
  The network traffic will be sniffed based on the sliding time-window size. These techniques are similar to as mentioned in [9].
- **Feature Extractions**
  Network traffic feature extraction can occur at three levels: packet-level, flow-level, and connection-level. The author's have used a mixture of connection and packet levels. This is done in two stages. First, connection features are extracted. Then, these features serve as host features.
  - Connection level features
    At this level the focus is on features that are important for the detection of the P2P botnet. The features collected comprise control packets exchanged between network hosts and is 5 tuples (Source IP, Destination IP, source port, destination port, Protocol)
  - Feature reduction
    Feature reduction is used to decrease the 'overfitting' problem. They have used the classification and regression tree (CART) as the feature reduction technique.
  - Host feature extraction at network level
    This approach is based on following three observations. First, bot infected hosts share particular malicious behaviour, and the features differ from those of a normal host. Second, bot's behaviour during propagation repeats itself in a frequent manner since it is attempting to infect multiple hosts. Third, a software program generates the bot connection.
- **Malicious activity detection**
  Malicious activity detection includes an offline stage(training), an online detection stage, and a reinforcement-learning stage. Training involves providing classifier with a group of legitimate and bot feature vectors. Online detection stage will continuously classify the host within the network. Reinforcement learning agent simultaneously operates to extract new features that shall participate in improving the performance level of the detection agent in the future.
- **Bot detection using reinforcement learning**
  Reinforcement learning (RL) are widely used to handle problems that involve difficulty in determining the solution explicitly, provided that it is probable to generate the signals of reward. This applies to our botnet detection problem. The RL 'obstacle' is expressed in the partially observable markov decision process (POMDP) context.

The method was used on ISOT and ISCX dataset. It was able to achieve accuracy of 99.10%.

G. Peerfox: Detecting parasite P2P botnets in their waiting stage

The method proposed [14] a three tier approach for detection of parasite botnet. It is able to detect the bots in their waiting stage itself with high detection accuracy and negligible FPR. The detection framework considers two type of packets for detection: *advertisement packets*, and *search request packets*. The framework consists of three modules:

- **Pre-processing module:** This filtering module involves filtering only UDP packets and discarding all other raw data. From those UDP packets collected, only advertisement and search packets are retained.
- **Suspected Peers' Identification Module:** It creates a list of IP address of peers that have been advertising themselves continuously for an hour and marks them as suspected peers. The peers which send out search request packets consistently and with high intensity are labelled as bots.
- **Classification Module:** For classification it uses MultiBoostAB, DecisionTable and various implementations of Decision Trees.

The dataset used was the Storm botnet dataset. It was able to bots with detection accuracy above 99

IV. Countermeasures for botnet attacks

Over time, many methods have been developed by researchers to deal with P2P botnets. Some deal with stopping them, others with mitigating the impacts of it [2].

A. Detection

Detecting a botnet is the first line of defence against a botnet. If we could detect a botnet in infant stage we can prevent it from infecting multiple hosts. Various detection technique are already described in previous section.

B. Monitoring

Monitoring of P2P botnets help us in understanding their motivations, working patterns, evolution of design, etc. There are two effective ways to conduct P2P botnet monitoring:

- **Sensors:** Existing peers in a network infected by botnet can be used as sensors. Legitimate peers in a P2P network can used as sensor for botnet monitoring. For choice of sensor, generally an important node is chosen so that more information can collected about the botnet.
- **Honeypots:** Honeypot is system which acts as lure ofr botnets. After a honeypot has infiltrated the botnet C&C channel, it can be used for variety of purposes. It can collect information related to identities of other bots connected to network, various commands issued by botmasters. However, botnets have now evolved techniques to detect honeypots. Parallely, researchers have also found ways to better disguise honeypots.

C. Shutdown

Ultimately we want to shutdown botnets.

- **Physically shutting down P2P Bots**
  Bootstrapping is an essential step in botnet construction. By isolating or shutting down those bootstrap servers, we can prevent a botnet from growing. Blocking those bots which are present in the initial list of bots binary will also help in stopping the growth of botnets.
  Botnets detected need to be removed. Removing bots which are important in the botnet C&C communication is more effective in stopping botnets. Generally, two approaches are taken for removing botnets. We can take **random removal** approach where by a bot is removed whenever it is encountered. Other approach is **targeted removal**, where we remove critical bots. We develop some metric to calculate "critical" bot.

- **Shutting down botnet C&C channel**
  Shutting down individual detected bot is slow and sometimes impossible. A better approach would be to prevent bots from communicating with their C&C channel. This is assumed to be difficult for P2P botnets as there is no centralised C&C servers. However, P2P botnets that rely on publishing/subscribing mode for C&C communication are still vulnerable to this method.

  Index Poisoning Attack: Peers in a network locate required files using an index. This index can be stuffed with massive number of bogus records. The index can be updated by any peer in the network. When a particular peer searches for a file, the poisoned index will return garbage result. The peer won't be able to download the file or it could redirected to download some other file. This way index poisoning can be used to mitigate P2P botnets. Especially those P2P botnets those are based on publishing/subscribing are vulnerable to index poisoning attack due to:

  – No authentication is required for a peer to insert or rewrite records in file index.
  – The P2P botnet uses a limited number of predefined hash values for command communication and these values can be figured out by defenders.

  Sybil Attack: It works by forging identities to subvert the reputation system in P2P networks. It can be used a mitigation strategy for fighting against P2P botnets. This strategy generally involves flooding a botnet with large number of fake nodes or sybils. Sybil nodes that are inserted in the peer list of bots can disrupt botnet C&C communication by re-routing or stopping C&C traffic going through them.

  Blacklisting: For blacklisting, we can either use query blacklisting or peer blacklisting. A query blacklist will stop queries related to botnet command. Peer blacklist stops stop bots which are there in blacklist.

V. Conclusion

This paper looked at various techniques for botnet detection. We have stated some initial methods for botnet discovery such as looking at ports, connections failures, etc. However, attackers have developed effective technique to escape these method. Presently, the detection methods have mostly shifted to machine learning paradigm. They generally operate in two stages. First stage, is where data collection, feature extraction and feature reduction is done. The next stage involves using variety of classifiers to classify and cluster data. We have listed those methods above.

References

[1] Narang, P., Khurana, V., Hota, C. (2014, May) Machine-learning approaches for P2P botnet detection using signal-processing techniques. In Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems (pp. 338-341).

[2] Wang, P., Aslam, B., Zou, C. C. (2010). Peer-to-peer botnets. In Handbook of Information and Communication Security (pp. 335-350). Springer, Berlin, Heidelberg.

[3] R. Schoof and R. Koning, "Detecting peer-to-peer botnets", University of Amsterdam. 2007.

[4] Khan, R. U., Kumar, R., Alazab, M., Zhang, X. (2019, May). A hybrid technique to detect botnets, based on P2P traffic similarity. In 2019 Cybersecurity and Cyberforensics Conference (CCC) (pp. 136-142). IEEE.

[5] Khan, R. U., Zhang, X., Kumar, R., Sharif, A., Golilarz, N. A., Alazab, M. (2019). An adaptive multi-layer botnet detection technique using machine learning classifiers. Applied Sciences, 9(11), 2375.

[6] Garcia, S.; Grill, M.; Stiborek, J.; Zunino, A. An empirical comparison of botnet detection methods. Comput. Secur. 2014, 45, 100–123.

[7] Sherif, S.; Traore, I.; Ghorbani, A.A.; Sayed, B.; Zhao, D.; Lu, W.; Felix, J.; Hakimian, P. ISOT Dataset Description. In Proceedings of the 9th Annual Conference on Privacy, Security and Trust (PST2011), Montreal, QC, Canada, 19–21 July 2011; pp. 19–21.

[8] Hang, H., Wei, X., Faloutsos, M., Eliassi-Rad, T. (2013, May). Entelecheia: Detecting p2p botnets in their waiting stage. In 2013 IFIP Networking Conference (pp. 1-9). IEEE.

[9] Alauthaman, M., Aslam, N., Zhang, L., Alasem, R., Hossain, M. A. (2018). A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks. Neural Computing and Applications, 29(11), 991-1004.

[10] Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classifi- cation and regression trees. Wadsworth Inc., Belmont, California

[11] Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 31:357–374

[12] David Zhao, Issa Traore, "P2P Botnet Detection through Malicious Fast Flux Network Identification", IEEE Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2012, pp. 170-175.

[13] Alauthman, M., Aslam, N., Al-Kasassbeh, M., Khan, S., Al-Qerem, A., Choo, K. K. R. (2020). An efficient reinforcement learning-based Botnet detection approach. Journal of Network and Computer Applications, 150, 102479.

[14] Priyanka, Mayank Dave, "PeerFox: Detecting Parasite P2P Botnets in their Waiting Stage", In International Conference on Signal Processing, Computing and Control ISPCC, IEEE, 2015, pp. 350-355.