2014

ISSN : 0974 - 7435 Volume 10 Issue 15

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(15), 2014 [8431-8440]

The applications of P2P botnet node-based detection

Changrong Liu[1], Huabei Nie[2], Jianqiao Shen[2]
[1]Huaian College of Information Technology, Jiangsu huaian 223003, (CHINA)
[2]City College of DongGuan Univercity of Techonoly, Dong Guan 523106, (CHINA)

ABSTRACT

Botnets can not only be implemented using existing well known applications, but also be constructed by unknown or creative applications. P2P botnets with low resource requirements have developed rapidly. In this paper, a novel P2P node-based detection is proposed, which exploits the node profile of the novel behaviors. Our experimental results show that it not only successfully detects known P2P botnets with a high detection rate, but also detects some unknown P2P malware.

KEYWORDS

P2P botnet node; Detection; Feature selection; True positive rate.

INTRODUCTION

Bot is an automated process that interacts with other network services. Bots often automate tasks and provide information or services that would otherwise be conducted by human beings. A typical use of bots is to gather information (such as web crawlers), or to interact automatically with web interfaces, such as instant messaging (IM), Internet Relay Chat (IRC), and others. They may also be used to interact dynamically with websites. Bots can be used for either good or malicious intent. A malicious bot is self-propagating malware designed to infect a host and connect back to a central server or servers that act as a command and control (C&C) center for an entire network of compromised devices, or "botnet." With a botnet, attackers can launch broad-based, "remote-control," flood-type attacks against their target (s). In addition to the worm-like ability to self-propagate, bots can include the ability to log keystrokes, gather passwords, capture and analyze packets, gather financial information, launch DoS attacks, relay spam, and open back doors on the infected host. Bots have all the advantages of worms, but are generally much more versatile in their infection vector, and are often modified within hours of publication of a new exploit. They have been known to exploit back doors opened by worms and viruses, which allows them to access networks that have good perimeter control. Bots rarely announce their presence with high scan rates, which damage network infrastructure; instead they infect networks in a way that escapes immediate notice. Nowadays, Botnet is the most serious threat of advanced malware.

The approach of bot detection using a signature-based technique has been widely addressed[1-7], and it has been found that this approach is effective to find some known bots, for example, Phatbot. Kolbisch et al.[8] proposed a signature-based malware detection system which used special graphs to determine bots. The method needs to be trained before, and its detection rate is only 64%, although it is possible to detect various kinds of bots. Besides, the signature-based method is not capable to detect unknown bots and a variant of known bots. Therefore, with the increasing number of new bot variants, its detection rate may decrease significantly. Flow-based techniques for bot detection increase the detection rate. The mechanism of the flow-based techniques was proposed to represent more general bot behaviors than the signature techniques[9,10]. The relevant available research on bot detection has been focusing on flow-based techniques. Livadas et al.[11] developed a system to detect C&C traffic of botnets based on flow. This system contains two stages: one is extracting several per-flow traffic features including flow duration, maximum initial congestion window, and average byte counts per packet; another is using a Bayesian network classifier to train a model to detect bots. However, its false positive rate is still high (close to 15.04%). Choi et al.[12] proposed a botnet detection mechanism solely based on monitoring of DNS traffic in the connection stage of bots. However, the botnet can easily evade this mechanism, if it rarely uses DNS at its initialization and will never use DNS. Wang et al.[13-15] presented a detection approach of P2P botnets by observing the stability of control flows in initial time intervals of 10 minutes. The usage of the protocol of a bot differs from that of a normal user, which may fluctuate greatly with user behaviors. Kang and Song[16] proposed a novel real-time detecting model named the Multi-Stream Fused Model, in which they deal with different types of packets in different methods. However, this model could not reach a desirable detecting precision when operated in a large-scale network environment. Besides, it could also generate extra harms to the Internet. Liu et al.[17] presented a general P2P botnet detection model based on macroscopic features of the network streams by utilizing cluster techniques. However, the proposed method was unreliable or non-functional if only a single infected machine is present on the network.

According to our knowledge, there has been no research published regarding the application of the node-based bot detection. The node-based bot detection is an effective and high-efficiency method in finding bots. It is of a higher level than both flow-based and packet-based detections. We expect that the node-based detection can result in better performance. Meanwhile the node-based detection has broader adaptability, since it is sensitive to new behaviors from bots implementing highly varied protocols.

In this paper, we proposed a novel node-based P2P detection. Comparing to traditional server-client botnet on the Internet, the P2P (peer-to-peer) botnet has capabilities to realize highly scalable, extensible and efficient distributed applications. The node-based P2P detection exploits the node profile generated from the novel behaviors as well as the degradation of the amount of traffic processed with sampling. It is expected to increase the detection rate. The details of the novel node-based detection technique are described and experiments to evaluate the performance of the node-based detection technique are conducted in this paper. Finally comparisons are maken between the novel technique and previous detection ones.

THE CHARACTERS OF P2P BOTNET NODE-BASED DETECTION

Different from other Internet malware, Botnet has its own unique characteristic, namely its control communication network. Usually, a “Botnet” consists of a network of compromised computers controlled by a bot-master and has a large scale on the Internet. The disadvantage of C&C server (centralized server) is that it can be easily shut down or blocked by firewall once it has been aware by its victim. Therefore, botmasters design a new mechanism to the botnet system, so that it does not depend on the central server anymore. It depends on any computer of the system (P2P). Each computer can be act as a client or server to any other computer in P2P network.

If P2P bot program uses a fixed port, a bot can be detected by detecting specific features. But most of the current bots change ports dynamically. Besides, some bots could use the normal ports such as port 80 to communicate, cheating the IDS. Thus, the bot detection based on ports is infeasible.

The bot program may use length-fixed packets (whose length ranges in a particular interval). This feature can be used to detect bots. However, some normal applications may have the same packet length. Thus, the bot detection based on the packet length could cause misjudgement.

In addition, different bots usually have different payloads. The unique sequence in a bot's payload can be extracted as the feature sequence of the bot. However, this is only useful to the bots that are known.

Although the signature-based detection has a high detection rate, it also has a lower generality. To resolve the problem, the concept of flow is introduced, which is the set of data packets with the same attributions. The same attributions usually satisfy a tetrad property, which means they have the same source address and port, the same destination address and port. Some researchers thought it should be a quintuple, including the protocol. But the quintuple is not suitable because of the inability to identify an unknown flow protocol. Flow-based P2P bot detection is heuristic and intelligent. It has the capacity of detecting unknown botnets.

The features extracted from flows are more general than those extracted from packets. Besides the general properties of packets, the features of flows also include the number of the packets, the order of the packet arrives, the order of the interval between packets, the flow speed and the flow lasting time. Since processing a flow has less time than processing every packet in the flow, this makes that the detection based on flows has a higher efficiency than detection based on packets does. With the chosen properties, data mining classification methods can be adapted to extract features and classify. Thus, this method can be used to detect unknown bots.

Detection on P2P Botnet is difficult as it has no central point (the C&C server). Any host connected to P2P Network can act as a C&C server. Once the botmaster obtains a list of host connected to P2P network, he can control every host as he wish. Although some computers are blocked by the firewall, once a bot is connected to at least one bot in another computer, it can receive any command indirectly from the botmaster through another computer.

The protection concept of detecting potential threat for the large scale of malicious software would be of strategic significance since such threats are serious and threatening. Botnets, networks of malware-infected machines (bots) controlled by botmasters, usually carry out their nefarious tasks, such as sending spam, launching denial of service attacks, and even stealing personal data[18]. Thus, how to detect botnets and remove them has become an interesting and important problem in network security. Botnets also have a variety of types, including P2P botnets, IRC botnets, and HTTP botnets, and so on[19-22]. P2P bot is distributed. If it can be detected in a reasonable time, the network security can be improved significantly[23].

Signature-based P2P bot detection is traditional and deterministic. It belongs to low level packet-based detection. Under a background of large data communication, detection based on packets generally has low process efficiency and a bad real time attribution, since it needs to process every packet in the flow. Furthermore, because the information contained in every packet is limited, an unknown bot cannot be recognized by this method.

The current methodology of signature-based detection mainly focuses on detecting a specific feature, for example, a specific port or a specific feature sequence in the payload. If the feature exists, the related source and destination addresses are stored into the bot dataset. However, different bot program has different communication protocols, different packet length, and different flow rate. Some of the bots even have their encryption mechanism to protect themselves. For a single packet, the features it can provide mainly include its source address and port, its destination address and port, the packet length and payload. As a result, it can only be able to detect one bot at a time.

NODE-BASED BOT DETECTION

A communication process can be considered as the interaction between connected nodes, in which one node corresponds to one IP address. That is, there are at least two flows in one communication process. Features that can be extracted in a connected node include the success rate of the node connecting, the distribution of communication protocols, the number of communication processes of a node, and the communication volume of a node. These features are very useful for our node-based detection.

According to the features of the detecting node, we can have the following two initiative detecting strategies: If a node uses both UDP and TCP protocols in one communication process within a certain time interval, this may belong to a P2P bot; For a P2P communication, one node often communicates with many nodes at the same time in order to maintain its distributed communication. As a result, this node holds several similar communication processes simultaneously. At the beginning of a communication, a P2P bot sends connecting requests to other bot nodes according to its peer list. It is obvious that a certain amount of requests fail, because some peers are shutdown or not infected. However, the success rate is usually high when normal applications send connecting requests. Thus, the success rate of connecting requests can be a criterion for bot detection. If the success rate of a node connecting is below 50%, it tends to be a bot node.

Although the node-based detection has to process one node at a time, it is still more efficient than the packet-based detection. The node-based detection needs to pre-store all the flow information on the current node. Thus, it needs a large storage space. This makes it infeasible for online active detection. Thus, it is very important to improve the efficiency of the node-based detection. Sampling is a popular technique in statistics. We apply it in our node-based detection.

Sampling

As we know from the section above, each packet will be processed one by one in either flow-based detection or node-based detection. Again, one packet by one packet processing is unsuitable for real time detection in high-speed network. It produces a high packet loss rate. The packet loss rates are obtained under different capturing speeds of different traffic networks, which are shown in TABLE 1.

In order to solve this issue, sampling[24-25] is introduced to decrease the number of packets to process, while keeping a higher detection rate.

TABLE 1: The packet loss rates under different capturing speeds of different traffic networks

Length of packet	Traffic rate	The number of packet	Packet loss rate
100 Bytes	8.93M bps	10922 /s	0
	29.75M bps	37021 /s	20.91%
	68.05M bps	85610 /s	60.33%
512 Bytes	32.23M bps	8013 /s	5.19%
	69.24M bps	16734 /s	34.98%
	90.71M bps	22330 /s	57.43%
1514 Bytes	58.47M bps	4860 /s	5.73%
	80.78M bps	6664 /s	32.66%
	96.12M bps	7991 /s	50.64%

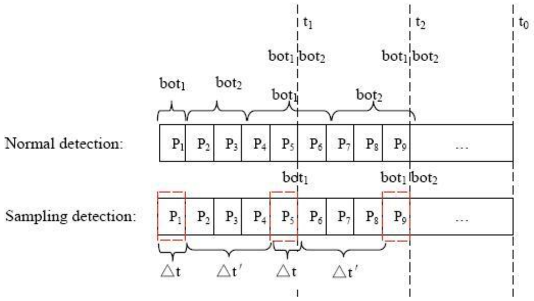

Figure 1: An example of sampling detection model

The effect of sampling on bot detection is shown in Figure 4. Two kinds of different detection approaches, i.e., normal detection and sampling detection, were analyzed in Figure 4 respectively. Figure 1 shows that the normal detection can possibly detect more bots than sampling detection at a certain moment. For example, at around t1, the normal detection detects two bots. Instead, the sampling detection detects one bot. However, with time increment, the two methods detect the same number of bots. For example, at around t2, both the normal detection and the sampling detection detect two bots. The asymptotic same result on detection at a certain moment is due to the cycle limit of the found bots from the real world.

Feature selection based on node

A feature represents a characteristic of a node in a given time window T, which could have a numeric or nominal value. TABLE 3 lists seven features we have selected for the purposes of our evaluation. Among the seven features, some features, such as the source and destination IP addresses, are extracted directly from the TCP/UDP headers, while others, such as the number of protocols used in the time interval, require additional processing and computation.

TABLE 2 : Behaviors of several P2P bot

	Host Behavior	Network Behavior	Remark
Phatbot	1. Modify the registry 2. Add startup item 3. Modify a file[1] 4. Terminate the thread of anti-virus	1. Start the IRC thread[2] 2. Start the P2P Server thread[3] 3. Start the P2P Client thread	1. Modify a file named host in system directory 2. Start the thread of IRC Client, and connect to IRC Server. 3.In order to improve the communication of p2p, start both client thread and server thread
Zhelatin .zy	1.Modify the registry 2. Add a startup item 3. Copy file[1]	1.Connect to SMTP server[2] 2. UDP connection[3]	1.In order to a bot’s propagation, copy the bot itself to the shared directory 2. Connect to SMTP Server by SMTP thread 3. A lot of UDP connections with both the same source port and the random target port
Sinit		1. UDP protocol 2. A high ICMP traffic 3. Sending packets to port 53[1]	1.Sending special discovery packets to port 53 of random IP addresses on the internet.
Nugache	1. Modify the registry[1]	1. Open TCP port 8[2] 2. encrypted data transmission[3]	1. Modify the registry and install the list with hosts into Windows’s registry. 2. Has a static list of IP addresses (20 initial peers) to which it will try to connect on TCP port 8. 3. The exchanged data could be encrypted, because it is not readable.

TABLE 3 : Selected Node features

Feature	Description	Type
Node	Computer address for transmitting information	string
NP	Number of protocols used for time interval	integer
NF	Number of flows used for time interval	integer
NPS	Number of packets sent for time interval	integer
RNP	Ratio of number of packets sent to number of packets received for time interval	real
ALPS	Average length of packets sent	real
RLP	Ratio of average sending packets length to average receiving packets length for time interval	real

To extract meaning features, we need to know the characteristics of P2P bots after we understand them. By the Vmware technology, a controlled environment is set up to analysis the behavior of some bots. In our research, four kinds of P2P bots are available. The behaviors of these botnets are shown in TABLE 2.

We selected the seven features based on well known protocols as well as the behaviors of the four botnets in TABLE 2. Please note that unlike normal peer to peer usage, P2P bot communication exhibits a more uniform behavior whereupon the bot queries for updates or instructions on the network continuously, and results in many continuous uniform small packets.

Decision tree

Many machine learning (ML) classification techniques attempt to cluster and classify data based on feature sets. Also lots of mathematical model can improve the accuracy for target detection[26-27]. In this paper, we select decision tree from popular classification techniques, because of its effectiveness and efficiency. Decision tree supports real time detection with high detection accuracy. Other efficient and effective classification algorithms can also be applied.

Evaluation indexes

In order to evaluate the performance of a botnet detection technique, we need to introduce a quantitative measurement. In our detection technique, we basically classify the network traffic data into normal or anomalous/suspicious groups. Any deviation from the normal traffic pattern is considered as suspicious. Hence we need to define true positive (TP), true negative (TN), false positive (FP) and false negative (FN) to determine true positive rate (TPR) and false positive rate (FPR). The TABLE 4 defines TP, FP, TN and FN.

TABLE 4 : Definitions of TP, FP, TN and FN

	Actual Group	Predicted Group
True Positive (TP)	Anomalous	Anomalous
False Positive (FP)	Normal	Anomalous
True Negative (TN)	Normal	Normal
False Negative (FN)	Anomalous	Normal

Now, the true positive rate (TPR) which is also known as sensitivity and the false positive rate (FPR) can be calculated using the following equations.

$$DR = TPR = \frac{TP}{TP + FN} \quad \textbf{(1)}$$

$$FPR = \frac{FP}{FP + TN} \quad \textbf{(2)}$$

$$\Pr ecision = \frac{TP}{TP + FP} \quad \textbf{(3)}$$

The true positive rate (TPR) evaluates the performance of a botnet detection technique in terms of the probability of a suspicious data reported correctly as anomalous. In other words it evaluates how well the model detected anomalous packets. On the other hand the false positive rate (FPR) evaluates the performance of botnet detection technique in terms of the probability of a normal traffic reported as suspicious generating false alarms.

Some related research on detection performance uses precision as the performance measurement. However, there is no research on the correlation between the detection rate (DR) and the precision. It can be seen from the following proof that the trend of FPR (i.e. DR) can be reached by precision.

$$precision \rightarrow 1 \Leftrightarrow \frac{TP}{TP+FP} \rightarrow 1$$

$$\Rightarrow FP \rightarrow 0$$

$$\Rightarrow \frac{FP}{FP+TN} \rightarrow 0 \Leftrightarrow FPR = 0$$

Besides, both the detection rate and the precision have the equivalent importance in the detection system. Thus, we proposed a combination of the two measurements, called Comprehensive Evaluation Index (CEI), which has a strategic significance for the evaluation of detection performance.

$$CEI = DR * 50\% + \Pr ecision * 50\% \tag{4}$$

EXPERIMENTS

In this section, we will investigate the performance of our node-based detection technique. First, we construct our experimental dataset and evaluate the performance of the node-based detection technique. Then we further compare it with the flow-based detection and a detection tool Bothunter[28].

Experimental dataset

We construct our experimental dataset by combining two separate datasets, which contain malicious traffic from the French chapter of the honeynet project[29] involving the Storm and Waledac botnets respectively. Waledac is currently one of the most prevalent P2P botnets and is widely considered as the successor of the Storm botnet with a more decentralized communication protocol. Unlike Storm using overnet as a communication channel, Waledac utilizes HTTP communication and a fast-flux based DNS network exclusively. To represent non-malicious everyday usage traffic, we further incorporated two non-malicious datasets into our experimental dataset. One of the two non-malicious dataset is from the Traffic Lab at Ericsson Research in Hungary[30], and the other is from the Lawrence Berkeley National Lab (LBNL). The Ericsson Lab dataset contains a large number of general traffic from a variety of applications, including HTTP web browsing behaviors, World of Warcraft gaming packets, and packets from popular bittorrent clients such as Azureus. The LBNL trace data provides additional non-malicious background traffics. The LBNL is a research institute with a medium-sized enterprise network. The dataset contains trace data for a variety of network activities spanning from web and email to backup and streaming media. This variety of traffic serves as a good example of day-to-day use of enterprise networks.

Experimental results

We implemented our method in Java and utilized the popular Weka machine learning framework and libraries for our classification algorithm - decision tree. Our program extracts all node information from a given pcap file, and then parses the nodes into relevant features for use in classification.

The detection effectiveness at different periods is shown in TABLE 5. When the time window is 10s, the detection rate increases very slowly with the increment of the amount of training data, and reaches the higher value 0.667 for 50, while precision reaches 0.545. However, with the further increasing amount of training data, both the detection rate and precision decreased very quickly and reach 0.333 and 0.333, respectively. When the amount of training data is 10, both detection rate and precision are momently higher, which can be due to the fact that training data is close to bot behavior. Furthermore, the less data may also result in the increased detection rate and precision.

TABLE 5 : The detection performance at different time window sizes

Time Window	10	20	30	60	180
FN rate	0.0024	0.0022	0.0026	0.0018	0.0002
FP rate	0.0172	0.0182	0.0158	0.008	0
Precision	0.9976	0.9978	0.9976	0.9982	0.9998

For node-based detection on a period of 60 seconds, the maximum detection performance has been achieved, i.e., detection rate is for 1 and precision is for 1. When the amount of training data is 10, detection rate has been the maximum, while the detection rate decreased very quickly with increasing the amount of training data and reached the lower value 0.306 at 50. When the amount of training data is 10, both detection rate and precision are momently higher, which can also be due to the fact that training data is close to bot behavior. Furthermore, the less data may also result in the increased detection rate and precision. But at 60, the maximum detection performance is reached again, presumably due to the period of 60 seconds is similar to real bot cycle.

For the period 180s, with increasing the amount of training data, detection rate always kept the maximum 1 and false positive decreased to the minimum 0, while precision reached the maximum 1 too. These results showed that the right time window was obtained for bot detection.

TABLE 6 : The detection performance of the sampling detection at different time intervals

Time interval(s)	Time window	FN rate	FP rate	Precision
0	10	0.003	0.026	0.997
	20	0.002	0.024	0.998
	30	0.002	0.007	0.997
	60	0.001	0	0.999
	180	0	0	1
10	10	0.002	0.004	0.998
	20	0.002	0	0.997
	30	0.001	0	0.998
	60	0.005	0	0.995
	180	0.004	0	0.996
20	10	0.002	0.005	0.998
	20	0.003	0.035	0.996
	30	0.001	0.015	0.998
	60	0.003	0	0.996
	180	0	0	1
30	10	0.002	0.016	0.998
	20	0.002	0.007	0.998
	30	0.003	0.053	0.997
	60	0.001	0	0.998
	180	0	0	1
60	10	0.002	0.053	0.997
	20	0.001	0	0.998
	30	0.003	0	0.997
	60	0.001	0.01	0.998
	180	0.001	0	0.998
180	10	0	0	1
	20	0	0	1
	30	0	0	1
	60	0.001	0.026	0.999
	180	0	0	1

In our experiments, when the time window is 10s, 20s, 30s, 60s, and 180s, the sampling detection experiments are conducted with the time interval of 0s, 10s, 20s, 30s, 60, and 180s, respectively. The experimental results are shown in TABLE 6. From TABLE 6, we can conclude that when the time window is 180s, the increment of the time interval does not make any impact on the detection performance.

When the time window is 10s and the time interval is 10s, the impact of the sampling detection on the number of bots was found. The number of bots found and the amount of data processing are shown in Figure 2. These results are generally in agreement with those analyzed in the previous section. The proposed approach can reduce more than 60% input raw packet traces and achieve a high detection rate (about 99%) and a low false positive rates (0-2%).

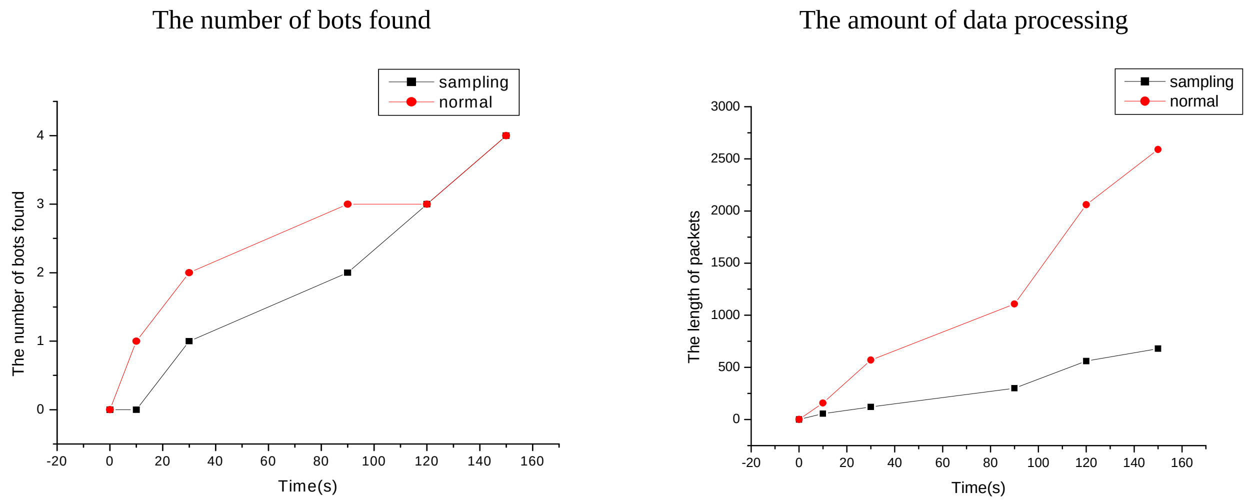

Figure 2: The number of bots found and the amount of data processing under sampling measurement

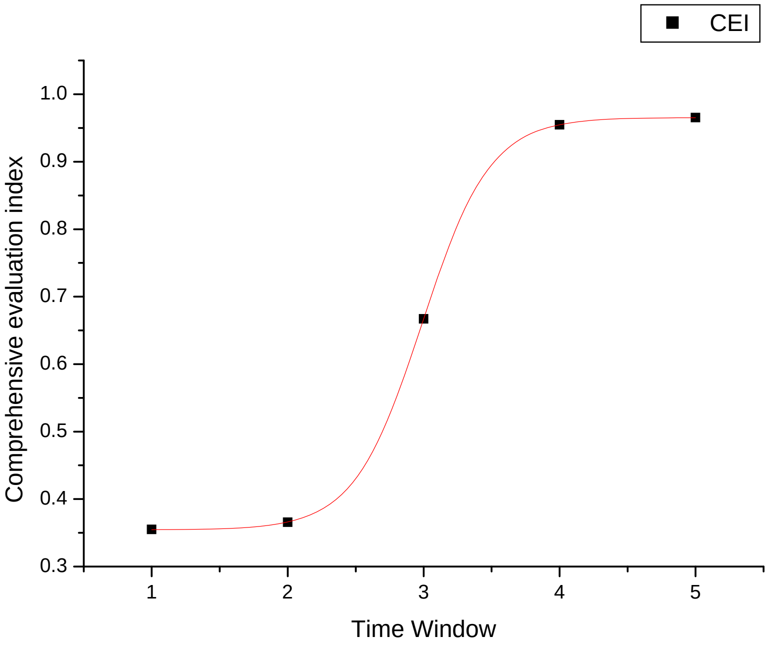

Figure 3: The CEI of P2P bot detection algorithm as a function of time window (measure time by the second).

Using proposed evaluation index, the effectiveness of the time window in term of the CEI of the node-based P2P bot detection is shown in Figure 3. Various sizes of the time window were set for evaluating our detection method on the dataset. Figure 3 shows that when the time window is below a certain level, for example, around 180s, the CEI increases very quickly with the increment of the size of the time window. However, a further increment of the size of the time window has only a small impact on the CEI. The asymptotic upper limit on CEI with the size of the time window is due to the operating cycle time of a real bot. A real bot has unique characteristics.

Comparison with flow-based detection

It is expected that node-based bot detection method performs better than flow-based one, since the node-based detection has broader adaptability than the flow-based one. The flow-based detection is sensitive to new behaviors from bots implementing highly varied protocols. To verify this expectation, we implement the flow-based detection. First, we extract 12 features from the flow traffics as the flow features, shown in TABLE 7.

TABLE 7 : 12 Features selected for flow-based detection

Attribute	Description
SrcIp	Flow source IP address
SrcPort	Flow source port address
DstIp	Flow destination IP address
DstPort	Flow destination port address
Protocol	Transport layer protocol or ‘mixed’
APL	Average payload packet length for time interval.
PV	Variance of payload packet length for time interval.
PX	Number of packets exchanged for time interval.
PPS	Number of packets exchanged per second in time interval T
FPS	The size of the first packet in the flow.
TBP	The average time between packets in time interval.
NR	The number of reconnects for a flow
FPH	Number of flows from this address over the total number of flows generated per hour.

TABLE 8 : Comparison between node-based detection and flow-based detection

	Flow based	Node based
True positive	98.3%	100%
False positive	0.01%	0.00%

TABLE 8 shows that our node-based detection outperforms the flow-based one. It can achive very high detection rates with a very low false positive rate. Thus, we can conclude that between the two methods, the node based method was more accurate.

Comparison with bothunter

BotHunter is one of the few botnet detection tools relevant to our work that is openly available. BotHunter mainly consist of a correlation engine that ties together alerts generated by Snort. It includes two custom plugins (called SLADE and SCADE) for snorting. The SLADE plugin mainly detects payload anomalies, while the SCADE plugin detects in/out bound scanning of the network. Besides, it includes a rule set that is specifically designed to detect malicious traffic related to botnet activities, such as egg downloads and C&C traffic. The correlation engine ties all the alerts together and generates a report for infections if any.

After running BotHunter on our dataset, the generated alerts indicated that there is a spambot in the dataset. More specifically, three alerts with “Priority 1” report the presence of botnet traffic. The three alerts all pointed to the same IP address. This IP address corresponds to a machine that was infected with the Waledac botnet. However, BotHunter failed to detect the other machine that was infected with the Storm botnet. Furthermore, among the 97,043 unique malicious flows in the system, BotHunter was able to detect only 56 flows (a very small number of malicious flows).

CONCLUSIONS

This paper first comparatively analyzed the generality and detection rate of different detection methods. In summary, flow-based detection generalizes the commonality features of flows via the analysis of many known botnets. With these commonality features, flow-based techniques can institute rules for multiple botnets detection, as well as for some unknown botnets. The disadvantage of this method is that some legal applications may share the same flow features. This could result in a high false positive rate. Compared with flow analysis, node-based detection extracts more general features of a botnet. One node represents one bot machine. This technique detects botnets from a macroscopic angle.

In this paper, we proposed a P2P botnet process with node-based sampling, which resulted in the increment of the detection rate, due to the node profile of the novel behaviors as well as the degradation of the amount of traffic processed with sampling. When the size of the time window is relatively proper, for example, about 180s, the detection rate is more than 90%. In the sampling process, the false positive and the amount of traffic processed can be decreased by 30% and 50-60% respectively. Precision could be significantly increased at a proper time window.

It is very important and necessary to design a system that can evaluate the performance of the detection online, instead of offline. It is also important to train the detection system online, instead of an offline training process. Such a system is ideal for identifying new threats.

REFERENCES

[1] D.Xiaomei, et al; “A Novel Bot Detection Algorithm based on API Call Correlation,” Proceedings of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery, 1157-1162, **(2010)**.

[2] Z.Wang, et al; “The Detection of IRC Botnet Based on Abnormal Behavior,” Proceedings of the Second International Conference on Multimedia and Information Technology, 146-149, **(2010)**.

[3] S.Wang, et al; “Method of Choosing Optimal Characters for Network Intrusion Detection System,” Computer Engineering, **36(15)**, 140–144 **(2010)**.

[4] B.Al-Duwairi, L.Al-Ebbini; “BotDigger: A Fuzzy Inference System for Botnet Detection,” Proceedings of the Fifth International Conference on Internet Monitoring and Protection, 16-21 **(2010)**.

[5] Y.Al-Hammadi, U.Aickelin; “Detecting Bots Based on Keylogging Activities,” Proceedings of the 3rd International Conference on Availability, Reliability and Security, 896-902, **(2008)**.

[6] M.Crotti, et al; “A statistical approach to IP-level classification of network traffic,” Istanbul, **1**, 170–176 **(2006)**.

[7] X.L.Wang, et al; “Research of automatically generating signatures for botnets,” Journal of Beijing University of Posts and Telecommunications, **34(4)**, 109–112, **(2011)**.

[8] C.Kolbitsch, P.M.Comparetti, C.Kruegel, E.Kirda, X.Zhou, X.Wang; “Effective and efficient malware detection at the end host,” Proceedings of 18th USENIX Security Symposium, 351–366, **(2009)**.

[9] K.C.Wang, et al; “A fuzzy pattern-based filtering algorithm for botnet detection,” The International Journal of Computer and Telecommunications Networking, **55(15)**, 3275–3286 **(2011)**.

[10] K.Jian, et al; “Accurate Detection of Peer-to-Peer Botnet using Multi-Stream Fused Scheme,” Journal of Networks, **6(5)**, 807-814 **(2011)**.

[11] C.Livadas, R.Walsh, D.Lapsley, W.T.Strayer; “Using Machine Learning Techniques to Identify Botnet Traffic,” Proceedings of the 31st IEEE Conference on Local Computer Networks, 967-974, **(2006)**.

[12] H.Choi, H.Lee, H.Lee, H.Kim; “Botnet detection by monitoring group activities in DNS traffic,” Proceedings of the 7th IEEE International Conference on Computer and Information Technology, 715-720 **(2007)**.

[13] Binbin Wang, Zhitang Li, Hao Tu, Jie Ma; “Measuring Peer-to-Peer Botnets Using Control Flow Stability,” Proceedings of the International Conference on Availability, Reliability and Security, 663-669 **(2009)**.

[14] Yi Wang, Cheng Gong, Baoku Su, Yunji Wang; “Delay-dependent robust stability of uncertain TS fuzzy systems with time-varing delay,” International Journal of Innovative Computing, Information and Control, **5(9)**, 2665-2674 **(2009)**.

[15] Wenting Zha, Junyong Zhai, Shumin Fei, Yunji Wang; “Finite-time stabilization for a class of stochastic nonlinear systems via output feedback,” ISA Transactions, **53(3)**, 709-716.

[16] Jian Kang, Yuan-Zhang Song; “Accurate Detection of Peer-to-Peer Botnet using Multi-Stream Fused Scheme,” Journal of Networks, **6(5)**, 807–814 **(2011)**.

[17] Dan Liu, Yichao Li, Yue Hu, Zongwen Liang; “A P2P-Botnet Detection Model and Algorithms Based on Network Streams Analysis,” Proceedings of the International Conference on Future Information Technology and Management Engineering, 55-58, **(2010)**.

[18] B.Stone-Gross, et al; “Analysis of a Botnet Takeover,” IEEE Security & Privacy, **9(1)**, 64–72 **(2011)**.

[19] X.B.Ma, et al; “A Novel IRC Botnet Detection Method Based on Packet Size Sequence,” Proceedings of the International Conference on Communications, 1-5, **(2010)**.

[20] L.Wen-Hwa, C.Chia-Ching; “Peer to Peer Botnet Detection Using Data Mining Scheme,” Proceedings of the International Conference on Internet Technology and Applications, 1-4,**(2010)**.

[21] Yang Zhao; Study on Predictive Control for Trajectory Tracking of Robotic Manipulator, Journal of Engineering Science and Technology Review, **7(1)**, 45-51 **(2014)**.

[22] D.A.L.Romana. et al; “Detection of Bot Worm-Infected PC Terminals,” Information-an International Interdisciplinary Journal, **10(5)**, 673–686 **(2007)**.

[23] P.Wang, et al; “An Advanced Hybrid Peer-to-Peer Botnet,” IEEE Transactions on Dependable and Secure Computing, **7(2)**, 113–127 **(2010)**.

[24] Braun, Lothar, G.Munz, Georg Carle; "Packet sampling for worm and botnet detection in TCP connections." Network Operations and Management Symposium (NOMS), 264-271 **(2010)**.

[25] Yang Zhao; Robust Predictive Control of Input Constraints and Interference Suppression. International Journal of Control and Automation, **7(7)**, 371-382 **(2014)**.

[26] Yunji Wang, Hai-Chao Han, Y.Jack Yang, L.Merry Lindsey, Yufang Jin; “A conceptual cellular interaction model of left ventricular remodelling post-MI: dynamic network with exit-entry competition strategy,” BMC Systems Biology, Suppl 1, S5, **(2010)**.

[27] Yunji Wang, Philip Chen, Yufang Jin; “Trajectory planning for an unmanned ground vehicle group using augmented particle swarm optimization in a dynamic environment,” IEEE International Conference on System, Man and Cybernetic, 4341-4346, **(2009)**.

[28] Gu, Guofei, et al; "BotHunter: Deteting Malware Infection Through IDS-Driven Dialog Correlation, " Proceedings of the 16th USENIX Security Symposium, 167-182, **(2007)**.