# Distributed and anonymous way of the malware detection

Peter Kenyeres, Gabor Feher

*Abstract* — **In this paper we introduce a multi-domain architecture and novel algorithms for malicious (potential botnet) activity recognition based on NetFlow network traffic statistics. Scalability and robustness were the main principles during the design of this architecture. We demonstrate a new method is able to recognize botnet participant computers (zombies), while the algorithms provide utmost anonymity to network operators. Furthermore, we also provide an aggregation scheme to significantly reduce the number of NetFlow records. This is important to handle the current high-speed networks efficiently.**

*Index Terms* — **anonymous, botnet, distributed malware detection, netflow**

## I. INTRODUCTION

In recent years, the Internet and the services are built upon it turned into part of our everyday life. Thus, we have to face the drawbacks of the technology more and more frequently. The criminals quickly recognized the exceptional possibilities lie in the new medium. In some cases, such as spam or phishing a whole business model was built around these activities. Nowadays, numerous and different abuses may threaten the confiding and/or careless users. And the number of users concerned by these attacks is continuously growing [1]. Generally speaking, the botnets give the technical background of the largest attacks.

Actually, botnet is vast network of compromised hosts under the control of single master who possesses the ability to launch crippling denial of service attacks (DoS), send enormous quantities of unsolicited e-mail messages (spam) and infect thousands of vulnerable systems with privacy-violating spyware or serving phishing sites, performing click fraud, etc. Besides, they also have aggressive exploit activity as they rope in new vulnerable systems to increase size of the network. The detection of above mentioned attacks are a relatively easy task, there are numerous solutions in the literature, e. g. [2] [3].

Despite these solutions the elimination or paralysis of attacks' sources raises more serious challenges. Researchers have proposed many different approaches to detect botnet behaviour in the monitored network. Gu, G. et al. show correlation based approach of the botnet detection process [9]. Livadas, C. et al. [10] integrate the recent results of the machine learning technique to detect botnet activities and [11], [12] use compression methods before the classification of the network traffic. However, one of the most common problem of the currently existing solutions, almost all of them are designed to use data from one single network only.

In this paper we introduce a novel security architecture which is reliable, efficiently scalable and can be anonymous. The architecture relies on a structured peer-to-peer (P2P) network to satisfy the scalability and the global availability requirements. Considering the huge amount of traffic data NetFlow [4] is applied to reduce the storage space required for traffic logs. Furthermore, since joined peers do not have any intention of revealing their traffic properties, so data anony-mization is a key issue in the system. However, network ad-ministrators will be able to recognize new threats and they can react to the infections more efficiently by contribution of our work.

In order to measuring the risk of the botnet threat together with gathering inputs to evaluate our algorithms: during six months long test period a HoneyPot [5] was run over an unused IP domain. Table 1 summarizes the results. Roughly speaking, one suspicious attempt occurred in every five seconds averagely. According to this, we may relate that the botnets still mean serious threat to the world computer networks. And we may deduct the inference that more than 80 per cent of the captured botnet clients still use IRC protocol as a C&C channel. However, the considerable part of the attacks embittering our everyday life (spam and particularly DDoS) are successful, only if they are executed with many computers in near identical time from many distinct places. Therefore, extended and distributed protections are desirable, which can be reached by collecting data from different local networks.

Thus, the whole malware network's recon, disablement and elimination become quicker and easier. But it brings up the following problems: the sample recognition can be quite difficult in the networks because of the different structure and their unique traffic patterns. The users' contrariety may mean additional difficulty because the network operators do not want to reveal their managed networks' structure or communication processes.

Manuscript received January 9, 2011.

Peter Kenyeres is with the *Budapest University of Technology and Economics, Department of Telecommunication and Media Informatics, High-Speed Networks Laboratory* (e-mail: kenyeres@tmit.bme.hu).

Gabor Feher is with the *Budapest University of Technology and Economics, Department of Telecommunication and Media Informatics, High-Speed Networks Laboratory* (e-mail: feher@tmit.bme.hu).

TABLE I
RESULTS OF OUR HONEYPOT

| Measured value | Quantity |
|---|---|
| Attack record | 3 303 194 |
| Mean attack frequency (per day) | 18 352 |
| Logged submission | 516 838 |
| Captured infected file | 10 689 |
| Captured unique malware | 108 |
| Captured botnet clients | 56 |
| Different IP address | 907 |
| The highest attempt from single IP | 31 295 |
| Origin countries | 64 |

Currently, this area of the botnet issue is quite open. Exactly, this is where we can fill the vacuum and can prove the necessity of a distributed architecture which provides efficiency, robustness and utmost anonymity. We put steps to organizing the defense based on the separately collected network traffic data. Furthermore, the anonymity guaranteed by our algorithms helps to win the users' confidence.

The remainder of the paper is organized as follows. In Section 2 the system model is introduced including the system architecture and the different type of nodes participating in it. In Section 3 the phases of our system and realization of the design priorities are presented. In Section 4 efficiency of algorithms are evaluated. Finally, the results are summarized in Section 5.

## II. SYSTEM ARCHITECTURE

The architecture consists of four elements: agents, honeypots, data processor and distributors. This layout is depicted in Figure 1. The first three are connected via a structured P2P network which implements Distributed Hash Table (DHT). This property helps to the participants join into system easily, hence to reach a globally available and distributed malware detection system. Further, it also improves scalability and robustness of the structure.

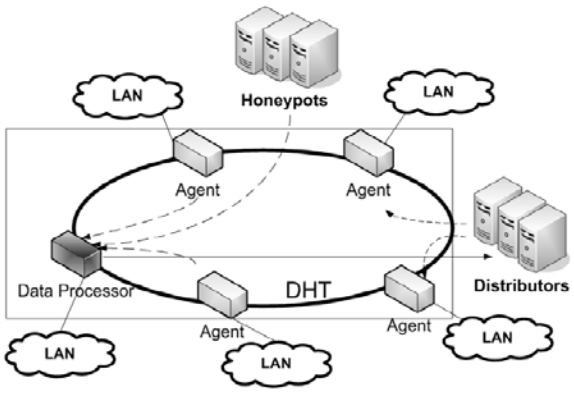


Fig. 1 The main components of the structure

The roles of the components are the following:

• Agents: These nodes are located between the border of the local area network and the Internet. They interactively dump and analyze the local network traffic to detect possible attacks, e. g. DDoS, spam, scan, etc.

• HoneyPots: These entities mark the suspicious traffic. When a new threat was detected, honeypot creates traffic traces of the malware, mark their command and control (C&C) channel and send the marked and anonymized trace to the current data processor.

• Data Processor: It is a specific agent node. Only one data processor exists in the network at same time, but this role is passed on at certain times. It collects the reports of malicious activities and the compressed and anonymized flows from agents as well as the anonymized and marked flows from honeypots. Its task is to create clusters from the data, evaluate the results. Furthermore, if malicious activities are detected it will have to send the network traces to a distributors.

• Distributors: They are responsible for sharing the collected anomalous traces with the agents. Distributors are independent from the P2P network. They accept requests from the data processors and serve the available sample updates to the agents.

## III. METHODOLOGY

In this chapter, we describe the details of the aforementioned task, such as, flow aggregation and sample creation.

### A. Flow Aggregation

NetFlow [4] logs are the inputs of this method. The most important fields of these are the source and destination IP addresses, source and destination port numbers and the transport protocol, since these define a session. In that case, if the same IP addresses are communicating with the same transport protocol, and at least one of them on the same port. Formally, if it is true for two flows A and B that

$$IP_{src}\_A=IP_{dest}\_B,\ IP_{dest}\_A=IP_{src}\_B,$$
$$Port_{src}\_A=Port_{dest}\_B\ OR\ Port_{dest}t\_A=Port_{src}\_B$$

These flows can be put into the same group, because they represent the different direction of the same connection. Hence, we can assume that the two records belong to the same session so it is unnecessary to treat them separately. This idea gives the key for the compression.

In the first step flow regrouping can be done by putting flow records with the similar connection parameters to the same group to represent each group by a single flow. Then, outlier filtering is applied for both directions to separately handle the salient data. This filtering due to all dimensions respectively is done by computing the m mean and the σ variance of the values. If the variance is relatively high (σ/m is greater than a fixed $\varepsilon_0$), a new group will be created for the most outlying flow. This step is iterated until there will be no more outliers. The last step is the aggregation of the remaining flows in each group to obtain a representant. The values for packets, octets and active time will be added up, the earliest start time and the latest end time will be selected, and 5 more values will be computed: number of flows aggregated, mean packet size, mean active time, duration (the time elapsed between the

earliest start time and the latest end time), up/down+down/up (up/down stands for the sum of octets in the flows with direction up/down, but these directions can be chosen arbitrary). This 5-tuple will represent a group. The IP addresses, port numbers and the transport protocol are omitted to get a kind of anonymity.

### B. Flow Processing and Sample Generation

The incoming aggregated NetFlow logs have to be classified to obtain flow samples belong to the botnet traffic we want to detect. Logs are sent by agents which are detected an attack. If this agent is a honeypot, the traffic logs will contain botnet traffic related flows (C&C channel communication and attack) without any legal background traffic. These flows are trusted in the sense that these are originated from a trusted entity and can be used as a sample of the botnet traffic. Therefore, these flows are referred as marked flows. The flows captured by honeypots that do not belong to the C&C channel can be marked differently or simply omitted. The clustering is applied to partition the data set that consists of marked and unmarked flows. Several previous works [7] [8] demonstrated that clustering of Internet traffic using flow statistics has the ability to group together flows according to the same traffic. The unmarked flows are used to improve the precision of the classifier. In this paper we applied the X-Means algorithm [6]. After clustering supervised learning and maximum likelihood estimation is applied to identify botnet traffic related cluster(s). It is not our purpose to identify all of the clusters, our aim is just to select those belong to botnet communication.

### C. C&C Channel Recognition

After agents have downloaded the samples they can start the C&C channel recognition procedure. First of all, all agents have to aggregate their flows to present a similar data structure like the aggregated sample. It not just decreases the size of the data set, but offers relatively fast search and preserves anonymity as well. To select all botnet related flows from the agent's flow set the clustering method discussed in Section 3.2 can be applied. Let x be a five dimensional vector from the agents aggregated flow set (described in Section 3.1). Then the following steps are required:

1. Calculate the distances of the feature vector x from the cluster center(s) in the samples:
$$d_1 = d(x, C_1), \ldots, d_r = d(x, C_r)$$
2. Select an index i, if there exists such that $d_i < \varepsilon_i$ (Note that if such an index exists, then it will be unique)

We can assume that, if those vectors are closer to the botnet releated cluster than certain di they belong to the botnet communication. Because if this vector is added to the training data set, then after the next iteration of X-Means the vector will be an element of this cluster.

## IV. EXPERIMENTAL RESULTS

We implemented our algorithms in native C and we have created a testbed network in the laboratory of the university to collect Netflow logs which contain certain malware traffic.

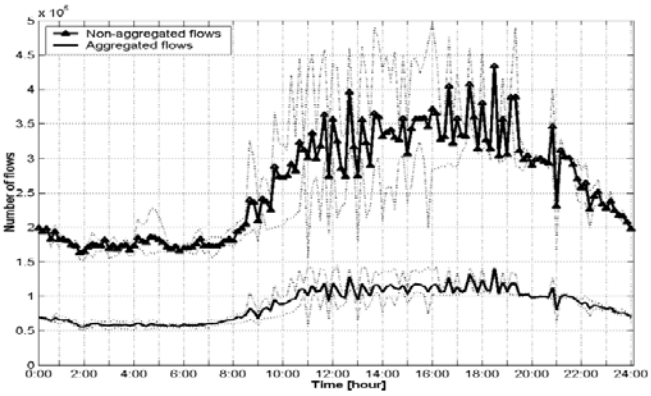


Fig. 2 Size of the original and the aggregated data sets over one day

At first, the efficiency of the aggregation scheme was analyzed separately by each campus log. Note that these traffic data came from a live network environment. And for all the 432 of 10-minute logs the compression ratio of the algorithm was between 0.3 and 0.35. Which means it reduced the size of the data set by 2/3. The average single-threaded preprocessing running time for one 10-minute log was less than 10 seconds. Figure 2 shows the size of the original and the aggregated data sets in the time period of one day.

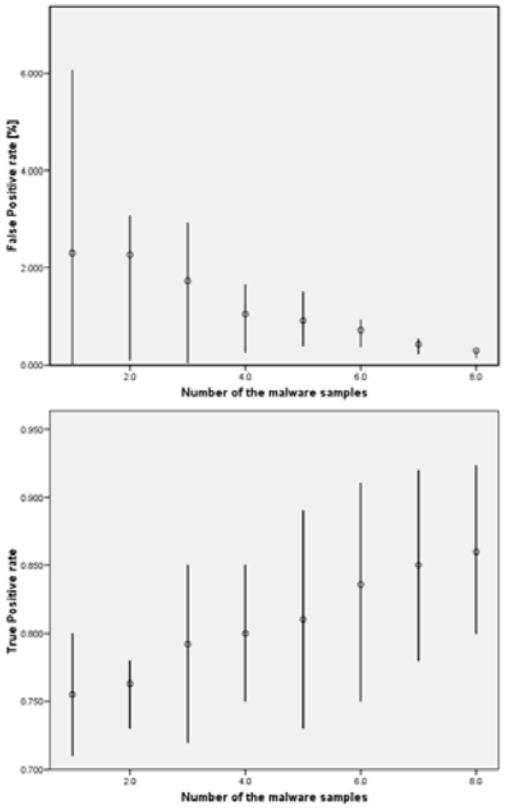


Fig. 3 and true positive rate depending on number of the samples

Further, we tested the C&C channel recognition algorithm. The data set came from campus live network and the laboratory testbed. We simulated three virtual LAN networks and in-

fected them with different botnet clients. In addition we had two more subnets: one for the victim and one for the botnet controller. Besides the legal traffic generated by the computers of the subnets, such as FTP, HTTP and e-mail, we simulated an attack against the victim directed by the botnet controller. This event triggered the sample generation process discussed in Section 3.2. Figure 3 depicts the recognition becomes more efficient (number of the false positive elements significantly decreasing while true positive ratio increasing) by increasing number of the malware samples which come from the detector agents.

## V. CONCLUSION

In this paper, we have shown architecture for anonymous and distributed malware detection. After the basics of system we presented our solution proposals to provide scalability, robustness and anonymization together with generate and distribute malware sample in multi-domain environment. In addition, we proposed two algorithms: one for detection with the help of samples which was generated in different subnets and another one for the reduction of the huge amount of network statistical data. We demonstrated the strength of the algorithms: i) the detection algorithm was able to find botnet clients using the aggregated samples. ii) the aggregation method reduced the NetFlow entries to one third in practice. We note that these samples provide anonymity in that sense they do not contain any kind of valid IP information. Consequently, each and every user can be sure that their network traffic is not revealed totally. As a result, there is no need to establish mutual and unconditional trust among all participants. This property of the architecture can facilitate to make extensive use of the system.

## REFERENCES

[1] Cloudmark International Spam Survey, http://www.cloudmark.com/en/survey-results/2010-08-02.
[2] V. Sekar, N. Duffield, O. Spatscheck, J. Van Der Merwe, H. Zhang, "LADS: Large-scale Automated DDoS detection System" USENIX ATC, pp. 171-184., 2006W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
[3] A. Garg, N. Reddy, "Mitigation of DoS attacks through QoS regulation" Microprocessors and Microsystems, vol. 28, Issue 10, pp. 521-530, Elsevier, 2004
[4] Cisco Systems NetFlow Services Export Version 9, RFC 3954, http://www.ietf.org/rfc/rfc3954.txt
[5] L. Spitzner, "Honeypots - Tracking hackers" Pearson Education, 2003
[6] D. Pelleg, A. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters" Int. Conf. on Machine Learning, pp. 727-734., Morgan Kaufmann, San Francisco, CA, 2000
[7] J. Erman, M. Arlitt, A. Mahanti, "Traffic Classification using Clustering Algorithms" SIGCOMM'06 MineNet Workshop, Pisa, Italy, 2006
[8] A. McGregor, M. Hall, P. Lorier, J. Brunskill, "Flow Clustering Using Machine Learning Techniques" PAM 2004, Antibes Juan-les-Pins, France, 2004
[9] G. Gu, P. Porras, V. Yegneswaran, M. Fong, W. Lee, "BotHunter: Detecting malware infection through ids-driven dialog correlation" Security'07, 2007
[10] C. Livadas, R. Walsh, D. Lapsley, W. T. Strayer, "Using machine learning techniques to identify botnet traffic" 2nd IEEE LCN WoNS'2006, Tampa, USA, 2006
[11] M. K. Reiter, T. F. Yen, "Traffic aggregation for malware detection" DIMVA'08, Paris, France, 2008
[12] S. Wehner, "Analyzing worms and network traffic using compression" Journal of Computer Security, pp. 303-320, vol. 15., IOS Press, 2007

**Peter Kenyeres** was born in Budapest, Hungary in 1983. He has been studying at the Budapest University of Technology and Economics since 2003. He obtained his MSc degree with major in Security of Information and Communication Systems in 2008. Currently, he is a PhD student of the Department of Telecommunication and Media Informatics in BME and a member of the High-Speed Networks Laboratory.

**Gabor Feher** graduated in 1998 as a computer engineer at the Budapest University of Technology and Economics. He received his PhD degree in 2004 in the field of resource control in IP networks. Currently he is an associate professor of the Department of Telecommunication and Media Informatics. He gives lectures in the topics of network and multimedia security. Since 1997 he is a member of the High Speed Network Laboratory of the Department and is involved in several national and international research projects.