ISSN:0975-9646

Development of a Scalable System for Stealthy P2P Botnet Detection

Navya Balla[1], P.V. Siva Kumar[2]

[1]*M.Tech Student (SE), VNR VignanaJyothi Institute of Engineering and Technology, Hyderabad, India*
[2]*Associate Professor (CSE), VNR VignanaJyothi Institute of Engineering and Technology, Hyderabad, India*

Abstract-Peer-to-peer (P2P) botnets are the modern and most resilient bot structure which might be more difficult to take down and stealthier to notice their malicious activity, in view that of which these are adopted by means of the various recent botmasters. In this paper, we recommend a novel botnet detection process which is ready to identify resilient P2P botnets. Our method at the beginning identifies the p2p communications present within the network. It then derives p2p traffic and further distinguishes between the botnet generated traffic and legitimate traffic. The parallelized computation makes scalability a default function of our process. Excessive detection accuracy and prodigious scalability are the extra features of our proposed process.

***Keywords*—Botnet, network security, P2P, intrusion detection.**

I. INTRODUCTION

A BOTNET is a gaggle of conceded hosts that are called as bots, which are managed by means of command and control channel (C&C) with the aid of an attacker. Botnets aids as infrastructure to many of the cyber-crimes, corresponding to distributed denial-of-service (DDoS) attacks, spamming, click on frauds, identity theft and so on.The C&C channel is the major factor in botnet since botmasters depends on this channel to ship commands to their bots and to receive information from the compromised hosts or machines. The C&C channels used in botnets are of specific forms.In centralized architecture, all of the bots in a botnet will be controlled or contact through one or few C&C servers which can be underneath botmaster. Conversely, a principal disadvantage of this structure is that they have single point of failure. With a view to overcome this challenge, botmasters have started to construct P2P botnets by using more resilient C&C structure.Bots belonging to P2P botnet form a network such that any node can be utilized by botmaster to send instructions or acquire knowledge from different peers .One of the most examples of P2P botnets are Waledac [1], Strom [2], Nugache [3], and even confiker are some interested botnets on the grounds that they have got used P2P C&C architecture because the major methods to set up and spread across network. These botnets are more difficult and more costly to control in comparison with centralized botnets. P2P botnets are extra stealthy against take down efforts (by using law and enforcement) so despite the fact that giant quantity of bots in P2P are detected and taken down still the remaining bots may be competent to keep in touch with each different and to botmaster.

So far, a couple of systems were proposed capable of detecting P2P botnets [4]-[6].However, these methods can't solve all the above mentioned challenges. For example, In BotMiner if the bots share equivalent communication patterns belongs to same botnet and performs identical malicious pursuits like spamming, exploiting, scanning and so on. Then it identifies as botnets. Unfortunately, malicious activities of bots could also be resilient there by non-observable and making BotMiner ineffective. Moreover the scalability of BotMiner is restrained [7]. Yen et al. [5] has proposed an algorithm which differentiates legitimate P2P application and P2P bot. However, this algorithm did not consider the fact that same host can behave both as legitimate host and bot rendering the algorithm ineffective.BotGrep[6] collects network flows over more than one giant networks(e.G., ISP community) ,and check out to become aware of P2P botnets by using examining the communication graph formed by overlay network. Despite the fact that BorGrep don't rely on malicious movements for detection, it requires prior detection results to bootstrap the detection and likewise wants a global view of internet traffic. Nevertheless, it is very intricate to accumulate such prior information in practice.

II. RELATED WORK

In this paper, we provide a novel detection procedure to determine the resilient (stealthy) P2P botnets. We refer to a resilient P2P botnets to these whose malicious activities are probably not observable in network traffic. Mainly, Our system main aims is to observe Resilient P2P botnets event if botnets generated traffic is overlapped with legitimate P2P applications(e.G. Skype) running on same compromised host and to also ambitions to gain high scalability. Our system, despite how bots participate in malicious movements in line with botmasters instructions, It identifies P2P bots from monitored network with the aid of C&C communication patterns that represent P2P botnets. Statistically, it derives statistical fingerprints from all P2P applications and leverage them to differentiate between hosts that are part of legitimate P2P networks (ex: File sharing networks) and P2P bots. To summarize, our work makes the following contributions:

1) Identifying hosts that interact in P2P communication by using a new flow-clustering based analysis.
2) Estimating active time of various P2P applications by using an efficient algorithm for P2P traffic profiling, where we build a statistical fingerprints.
3) A P2P botnet detection system that notices stealthy P2P bots even though the P2P botnet traffic is overlapped with traffic generated through professional

P2P purposes (e.g., Skype) running on same compromised machine.

4) A scalable design based on parallelized computation and efficient detection algorithm.
5) A prototype system, which has demonstrated great scalability and high detection accuracy.

Now we have made these enhancements in current procedure compare to a few different P2P detection methods published earlier. First, we removed coarsgrained evaluation factor(two level P2P client detection factor) to simplify the design and changed with single stage P2P client detection to scale back storage cost by 60% when you consider that it eradicates the necessity of preserving failed connections as shown in figure1filteing In-active P2P Client. Second, decreasing the processing time by at least 50% by redesigning the clustering based P2P client detection algorithm. Third, parallelizing our process to raise its efficiency and scalability are few major enhancements we made in current approach compared to older exiting techniques.

The Design objectives of our technique are distinctive when compared to present systems of detecting P2P botnets: 1) our method does not require any previous botnet understanding to make detection, not like [6]; 2) our process does no longer anticipate that malicious activities that botnets performs are observable, in contrast to [4]; 3) our process should be competent to notice compromised hosts that run each legitimate P2P application and P2P bot at equal time, in contrast to[5]: 4) distinct from [4]-[6], our approach is having high scalability as built-in characteristic. Other approaches [9]-[11] use machine learning for detection, which require prior or labeled P2P botnet data to train statistical classifier. Lamentably, obtaining such understanding is difficult and would possibly not feasible, thereby drastically limiting the realistic use of these methods.

To obtain the abovementioned design objectives, our system has a couple of components. The first one is flow clustering –based analysis approach to identify hosts which are running P2P applications. In divergence to current methods of opting for hosts running P2P applications[12]-[16] our technique differs in following ways: 1) our technique does no longer depend on constant source port used by [12],[14], which can be simply violated by using P2P applications; 2) not like [13], our procedure does not need content signature since encryption makes content signature vain; 3) our method don't have training data set to construct a machine learning based model as used in[15], seeing that it is rather difficult to get such information earlier than they are detected; 4) in divergence to [16] choosing a detailed P2P application, or procedure can discover and profile quite a lot of P2P applications; 5) our process can estimate active time of a P2P application which may be very principal element in botnet detection, active time of a bot will have to be comparable with the active time of the underlying compromised system. If this was not the case, the botnet overlay network would danger degenerating into a number of disconnected sub networks.

.

III. SYSTEM DESIGN

System Overview: P2P bots exhibits some network traffic patterns which might be customary to other P2P client applications due to the fact that a P2P botnet relies on P2P protocol to set up a channel with botmaster via command and control C&C channel. For that reason, we divide our system in two phases. Within the first phase, we purpose to realize all hosts in our monitored community that are engaged in P2P communication as shown in fig. 1, we analyze raw traffic accrued at edges of monitored network and follow pre-filtering to discard the network flows that are unlikely to be generated by non P2P applications.We then compare and mine a number of statistical features to determine flows generated by p2p clients from remaining traffic. Within the second section, our process will differentiate from legitimate (good) P2P clients to P2P bots by inspecting traffic generated by using P2P clients. Principally, we determine as a candidate P2P bot whether it is constantly active on the underlying host by considering the active time of p2P clients. We additionally analyze the overlap of peers contacted by two candidate P2P bots to finalize detection.

Table I

Notations and Descriptions

Notation	Description
Tp2p	The active time of P2P application
NO-DNS peers	The percentage of flows associated with no domain names
Nclust	The number of clusters left by enforcing Θbgp and Θp2p
Nbgp	The largest number of unique bgp prefixes in one cluster
Tp2p	The estimated active time for p2p application

A. *IdentifyingP2PClients*

*1) Filtering Traffic:*The Filtering Traffic component aims at filtering out network traffic that's unlikely to be related to P2P communications. This is entire by means of passivelyinspecting DNS traffic, and identifying network flows whose destination IP addresses have been beforehand resolved in DNS responses. Specially, we leverage the following features: P2P clients on the whole contact their peers immediately with the aid of looking up IPs from a routing table for the overlay network, as analternative than resolving a domain name. This option is supported by using table II (No-DNS peers), which illustrates that the sizeable majority of flows generated by P2P applications do not have destination IPs resolved from domain names. Theremaining small fraction of flows are akin to a viable exception that a peer bootstraps into a P2P network by looking up domain names that resolve to stable super-nodes)when you consider the most non-P2P applications (e.g., browsers, email clients etc.)Often connect with a destination address resulting from domain name resolution, this simple filter can eradicate very huge fraction of non-P2P traffic, while recollecting the vast majority of P2P communication

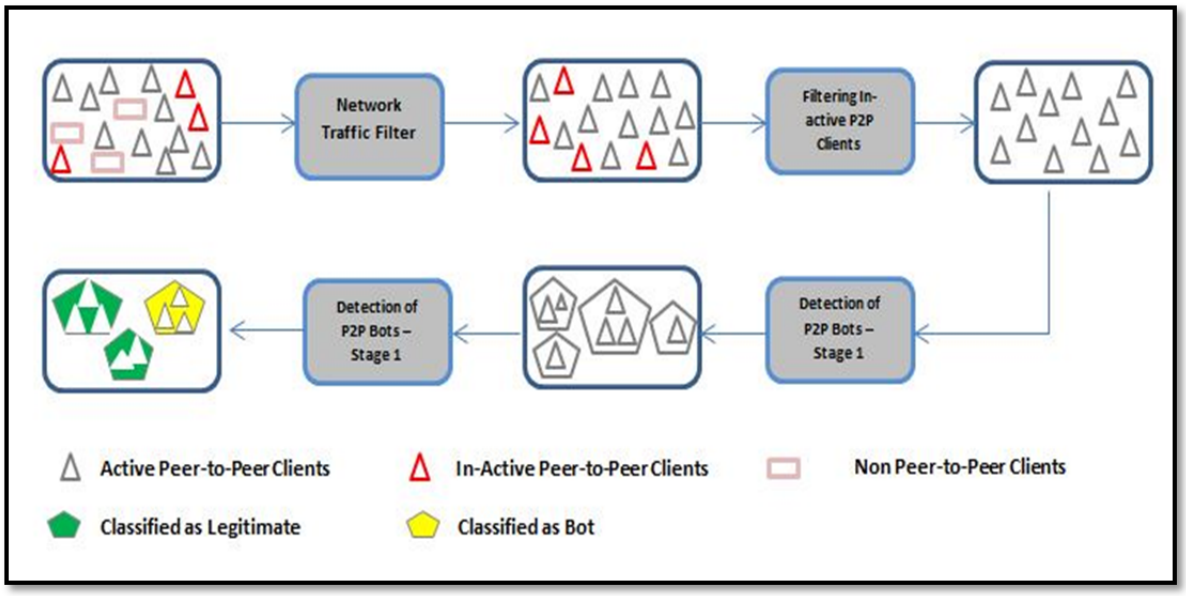

Fig. 1 System Overview

*2) Active P2P Clients Identification:*This component is liable for detecting P2P clients by inspecting the rest of the network flows after the Filtering Traffic component. For each host h within the monitored network we determine two flow sets, denoted as Stcp(h) and Sudp(h)which incorporate the flows regarding successful outgoing TCP and UDP connection, respectively. We consider as successful these TCP connections with an accomplished SYN, SYN/ACK, ACK handshake, and those UDP (virtual) connections for which there used to be at least one "request" packet and a consequent response packet.

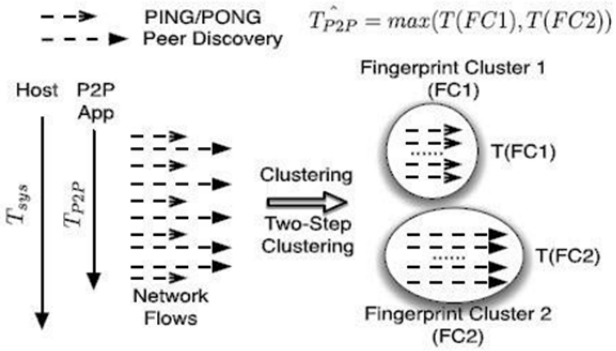

Fig. 2Example of flow clustering to identify P2P hosts

So as to identify P2P clients, we first ponder the fact that each P2P client regularly exchanges control messages (eg. ping/pong messages) with other peers. Besides, we become aware of that the characteristics of these messages, such as the size and frequency of the exchanged packets, are an identical for nodes in the identical P2P network, and vary depending on the P2P protocol and network in use. To identify flows corresponding to P2P control messages, we first observe a flow clustering process meant to group together equivalent flows for every candidate P2P node h. Given sets of flows Stcp(h) and S(h), we characterize each flow using a vector of statistical features v(h) =[Pkts, Pktr, Bytes, Byter] ,in which Pkts and Pktr represent the number of packets sent (Pkts)and received(Pktr), and Byte and Byter represent the number of bytes sent and received, respectively. The distance between two flows is subsequently defined as the Euclidean distance of their two corresponding vectors. We then apply a clustering algorithm to partition the set of flows into a number of clusters.

B. Identifying P2P Bots

*1) Detection of P2P Bots – Stage 1:*Since bots are malicious programs used to perform profitable malicious activities; they represent valuable assets for the botmaster, who will intuitively try to maximize utilization of bots. This is particularly true for P2P bots because in order to have a functional overlay network (the botnet), an enough number of peers needs to be always online. In other words, the active time of a bot should be comparable with the active time of the underlying compromised system. If this was not the case, the botnet overlay network would risk degenerating into a number of disconnected sub networks due to the short life time of each single node. Hence, this component aims at identifying P2P clients that are active for a time TP2Pclose to the active time Tsys of the underlying system they are running on. While this behavior is not unique to P2P bots and may be representative of other P2P applications (e.g., Skype clients that run for as long as a machine is on), identifying persistent P2P clients takes us one step closer to identifying P2P bots.

2) Detection of P2P Bots – Stage 2: This component is used to identify P2P bots from all persistent P2P clients. We leverage one feature: the overlap of peers contacted by using two P2P bots belonging to the same P2P botnet is far higher than that contacted via two clients within the same legitimate P2P network.Assume that two hosts in the monitored network, say hA and hB, are running the same legitimate P2P file-sharing application (e.g., Emule).).

Users of these two P2P clients will most likely have uncorrelated usage patterns. It is realistic to assume that in the general case the two users will search for and download different content (e.g., different media files or documents) from the P2P network.This translates into a deviation between the set of IP addresses contacted by hosts hA and hB. The reason is that the two P2P

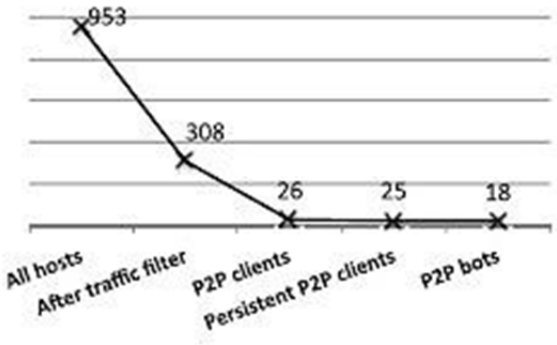

Fig. 3 Number of hosts identified by each component

Clients will tend to exchange P2P control messages (e.g., ping/pong and search requests) with different sets of peers which "own" the content requested by their users, or peers that are along the path towards the content. On the contrary, if hA and hB are compromised with P2P bots, one very common characteristic of bots is that they need to periodically search for commands published by the botmaster. This typically translates into a convergence between the set of IPs contacted by hA and hB

IV. SYSTEM IMPLEMENTATION

The implementation goal is to integrate high scalability as a constructed-in feature into our procedure. To this end, wefirst identify the performance bottleneck of our approach after which mitigate it utilizing complexity reduction and Parallelizing of system.

A. Performance Bottleneck

Out of four components in our system, "Traffic Filter" And "First Level Detection of P2P Bots" have linear Complexity seeing that they have got to scan flows only once toidentify flows with destination addresses resolved from DNS queries or calculate the active time. Other two components, "Detection of P2P clients" and "second-level Detection of P2P Bots", require pairwise assessment for distance calculation. Above all, if we denote the number of flows generated through a bunch as n and the number of hosts as S, the time complexity of Detection of P2P clients approximates O(S∗n2). Comparably, if we denote the numberof persistent P2P clients as *l*, the time complexity of second-level Bot Detection approximates O(l2).

B. Two-Step Flow Clustering

We use a dual clustering process to scale back the time complexity of "P2P client Detection". For the first-step clustering, we use an effective clustering algorithm to aggregate network flows into k sub-clusters, and each sub cluster contains flows which might be very similar to each and every other. For the second-step clustering, we examine the global distribution of sub-clusters and extra workforce an identical sub-clusters into clusters.

In the current design, we employ *K*-means as the first stepclustering. For the second-step clustering, we use *hierarchical clustering* with *DavisBouldin* validation [10] to group sub-clustersinto clusters.

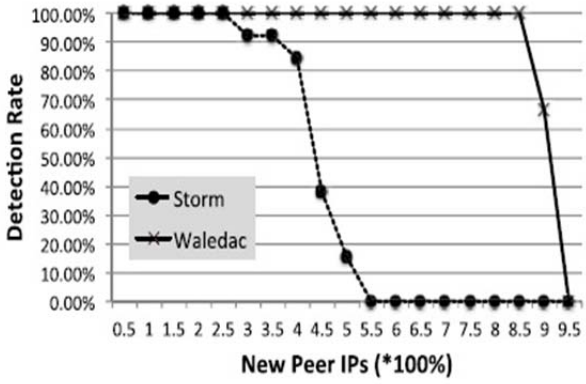

Fig. 4Challenges for attackers to instruct bots to contact different peers to peer

V. CONCLUSION

In this paper, we provided a novel botnet detection process that's able to identify resilient P2P botnets, whose malicious activities may not be identifiable. To achieve this challenge, we derive statistical fingerprints of the P2P communications to first discover P2P clients and further distinguish between those that are part of legitimate P2P networks (e.g., file sharing networks) and P2P bots. We also establish the efficiency bottleneck of our method and optimize its scalability. The analysis results established that the proposed approach accomplishes high accuracy on detecting stealthy P2P bots and satisfactory scalability.

REFERENCES

[1] G. Sinclair, C. Nunnery, and B. B. Kang, "The waledac protocol: The how and why," in Proc. 4th Int. Conf. Malicious Unwanted Softw., Oct. 2009, pp. 69–77.

[2] P. Porras, H. Saidi, and V. Yegneswaran, "A multi-perspective analysis of the storm (peacomm) worm," Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Tech. Rep., 2007.

[3] R. Lemos. (2006). Bot Software Looks to Improve Peerage [Online]. Available: http://www.securityfocus.com/news/11390

[4] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in Proc. USENIX Security, 2008, pp. 139–154.

[5] T.-F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart," in Proc. ICDCS, Jun. 2010, pp. 241–252.

[6] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in Proc. USENIXSecurity, 2010, pp. 1–16.

[7] J. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster, "Boosting the scalability of botnet detection using adaptive traffic sampling," in Proc. 6th ACM Symp. Inf., Comput. Commun. Security, 2011, pp. 124–134.

[8] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," J. Intell. Inf. Syst., vol. 17, nos. 2–3, pp. 107–145, 2001.

[9] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, et al., "Detecting P2P botnets through network behavior analysis and machine learning," in Proc. 9th Annu. Int. Conf. PST, Jul. 2011, pp. 174–180.

[10] D. Liu, Y. Li, Y. Hu, and Z. Liang, "A P2P-botnet detection model and algorithms based on network streams analysis," in Proc. IEEE FITME, Oct. 2010, pp. 55–58.
[11] W. Liao and C. Chang, "Peer to peer botnet detection using data mining scheme," in Proc. IEEE Int. Conf. ITA, Aug. 2010, pp. 1–4.
[12] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in Proc. ACM SIGCOMM, 2005, pp. 229–240.
[13] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in Proc. 13th ACM Int. Conf. WWW, 2004, pp. 512–521.
[14] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," in Proc. 4th ACM SIGCOMM Conf. IMC, 2004, pp. 121–134.
[15] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," in Proc. ACM SIGMETRICS, 2005,pp. 50–60.
[16] M. P. Collins and M. K. Reiter, "Finding peer-to-peer file sharingusing coarse network behaviors," in Proc. 11th ESORICS, 2006,pp. 1–17.