Tracking Multiple C&C Botnets by Analyzing DNS Traffic

Jehyun Lee*, Jonghun Kwon*, Hyo-Jeong Shin†, Heejo Lee*
*Division of Computer and Communication Engineering, Korea University
†Korea Telecom
{arondit, signalnine, heejo}@korea.ac.kr, hshin@kt.com

***Abstract*—Botnets have been considered as a main source of Internet threats. A common feature of recent botnets is the use of one or more C&C servers with multiple domain names for the purpose of increasing flexibility and survivability. In contrast with single domain botnets, these multi domain botnets are hard to be quarantined because they change domain names regularly for connecting their C&C server(s). In this paper, we introduce a tracking method of botnets by analyzing the relationship of domain names in DNS traffic generated from botnets. By examining the DNS queries from the clients which accessed the known malicious domain names, we can find a set of unknown malicious domain names and their relationship. This method enables to track malicious domain names and clients duplicately infected by multiple bot codes which make botnets revivable against existing quarantine methods. From the experiments with one hour DNS traffic in an ISP network, we find tens of botnets, and each botnet has tens of malicious domains. In addition to botnet domains, we find a set of other domain names used for spamming or advertising servers. The proposed method can be used for quarantining recent botnets and for limiting their survivability by tracking the change of domain names.**

I. Introduction

A botnet is the network of remote controllable malwares, which is one of the most significant source of Internet threats today. Botnets perform an attack such as spamming, DDoS, spying, and whatever possible malicious activities. According to the report from MessageLabs [1], botnets are ruling the cyber security landscape with at least five million compromised computers in 2009.

One of the important features of botnets we can observe and apply to detect them is that the botnets use multiple domain names to connect remote control servers. They have used various domain names to send spam mails, to drive users to infection servers, to find a victim and so on. Necessarily, accessing a server over the Internet with a domain name means that it needs to use DNS. In many cases, we can obtain the evidences of botnet activities from DNS queries generated by botnets. Some botnets use hard-coded IP addresses to connect target servers; however, according to the behavior of reported botnets, recent botnets tend to contact their target servers using domain names or both of IP addresses and domain names. In many studies [2], [3], [4], DNS is being used to detect and analyze botnet activities with the advantage of the possibility of monitoring in a single location, and we used DNS query traffic to track activities of botnets and the their relationship among the suspicious domain names contacted by the botnets.

Recent botnets commonly use multiple domain names for locating distributed servers. A botnet can have multiple functionalities, and it is controlled and communicating with different servers for each functionality. Some botnets which connects to its C&C(Command and Control) server through a domain name have used DDNS(Dynamic DNS) for changing the IP address. But recent botnets even change the C&C domain names more than IP addresses [5], [6]. We called these kinds of botnets multi-C&C botnets.

The multi-C&C botnets attempt to connect multiple domain names so that the connection failure one server will not stop working, and it is not only for C&C servers, but also for spam template servers, information update servers, infection servers of other botnets for cross-infection, and so on. In terms of botnet quarantine, the use of multiple domain names increase the botnet survivability. The survivability is caused from the mobility and redundancy of botnet related server including C&C. Against the botnet quarantine techniques such as DNS sinkholing, the multi-C&C botnets have stronger survivability based on redundancy.

In this paper, we proposed a simple but practical tracking mechanism to defend multi-C&C botnets. By tracking DNS queries from a botnet, we monitor the flow of queried domains of the botnet. Finding the set of multiple C&C, cross-infection status, and other distributed servers is a significant condition for detecting regeneration attempts and measuring survivability of a botnet. The tracking approach from the well-known information toward unknown is one of the basic investigation approaches in the real world, and we apply it to track the relationship among the domain names starting from blacklisted domain names and infected clients.

For measuring the reputation of each domain name, we use two categories of features target domain names have, which are sequential relationship and statistical commonness with the known malicious domain. The first challenge of tracking unknown malicious domain names is to distinguish malicious domain names from the legitimate. The second one is to consider unexpected environments of bot infected clients. From the observation of known botnet activities, we obtain the statistical and sequential difference. In the part of observation and experiment, we show the result of our observation with the botnet related domain names tracked from the test data set which is captured from a real network.

978-1-4244-8914-5/10/$26.00 ©2010 IEEE

II. Related Work

DNS has been considered as a monitoring place to detect botnets. In comparing with other approaches, DNS monitoring has had advantages against encrypted protocols and change of traffic behavior. On the other hand, the DNS based approaches have shown disadvantage against P2P based botnets. With considering portion of centralized botnets on wild, the DNS based approach might take majority and can be cooperated with any other approaches.

In the study with the concept of the set of clients of a domain name, Choi *et al.* [3] observed group activities on DNS traffic distinct from legitimate users, and measure the similarity of DNS clients of each domain names with quantitative likelihood. The approach is countered by recent multi-C&C botnets which separate their domain names.

Villamar *et al.* [2] proposed a Bayesian method to detect botnet based on the DNS traffic. In their study, known botnets, mutation of the botnet, and their fluxed domain names can be detected by their similarity of DNS traffic, however, the result is affected by traffic noises from background traffic because there can be a similarity on a famous domains with high possibility. In our study, we applied the sequence of queries to the concept of the DNS traffic similarity. The concept of sequence may enhance the performance by applying the previous studies.

Guofei *et al.* [7]'s work focused on the spatial-temporal correlation and similarity of botnet activities to detect C&C servers. The approach was also based on the pre-programmed activities of bots, and the property is hard to break. Their work can be applied to the multiple C&C condition without no big change, if it takes an appropriate policy of botnet classification to multiple C&Cs and domain-flux. This approach based on spatial-temporal correlation and similarity might be able to applied to the other type of servers that botnets try to contact over C&C focused on their work.

There have been several studies applying the graph structure for detecting botnets, and in the study of Shishirat *et al.* [8], they attempts to distinguish botnet communication. The graph approach they took on to structure and represent communication relationship between hosts. Their approach shares several concepts with this study. But, because of their focus and target data, the type of botnets in their coverage is mainly P2P.

John *et al.* [9] showed valuable analysis result on their work with their own system. Though observation result might have a limitation of artificially generated botnet traffic, their approach and concrete analysis of active botnets are applicable as a basis data to proof properties of botnets.

III. Multiple Domain Tracking System

To detect unknown and mutated malicious botnet domains, we propose the multiple domain tracking system based domain reputation. The reputation of a domain is estimated from the commonly appeared DNS behavior of botnet infected clients. The clients which are infected by the same bot have the same set of C&C domain names, at least a hub domain, to connect and the same query policy. For each bot of a botnet has own C&C domain names, or rarely shared domain names, a botmaster must manage all of the bots and their C&C domain names, and it is impossible without a hub domain. If there are any shared domain names including the hub, our mechanism tracks the other domain names and the botnet members.

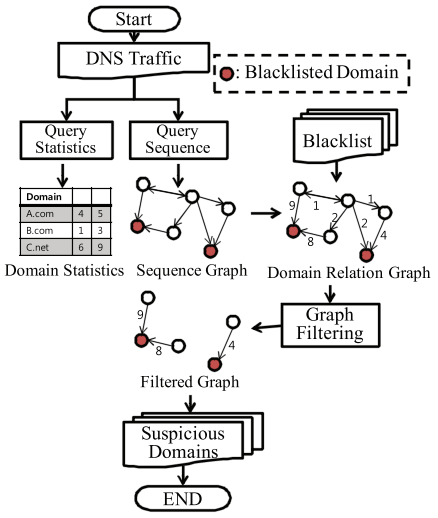

Fig. 1. The conceptual diagram of multi C&C botnet tracking system

Figure. 1 shows the conceptual diagram of the tracking system. In our system, the reputation of a domain is determined from two factors, statistical similarity with known malicious domains, and common sequential query pattern to known infected clients. Therefore, the system has two analysis module, statistical analysis and sequentiality analysis. The statistical analysis measuring how many clients a domain share with the malicious domains, and the sequentiality analysis track the sequential relationship among domains, The output of the statistical analysis is the number of shared clients and chained queries between two domains. And the output of the sequentiality analysis is sequence graphs which represent contacted domains by the clients.

By to composite two outputs of analysis, we get a set of domains which have high statistical similarity and the query behavior of the shared clients upon the domains. In Figure. 1, the composite result of two analysis is represented by a domain relation graph. Using the statistical data, the system verify the more common sequence of botnet clients and less common sequence of the clients.

A. Statistical Analysis

The first method is measuring statistics of DNS queries. Comparing with many other statistical tools for DNS traffic analysis, our system measures the statistics for estimating reputation of domain names. The statistical characteristic of some known botnet is distinctly observed because of their programmed and repeated queries. In terms of botnet recognition, each different botnet shows a different pattern with other botnets. One of the common features of known botnet

TABLE I
STATISTICAL FACTORS FOR THE ANALYSIS SYSTEM

Symbols	
q_{AB}	A sequential query of domain name B after A
Q_{AB}	Set of sequential queries q_{AB}
C_{AB}	Set of clients queries a sequential query A, and B
$\|Q_{AB}\|$	The number of sequential queries (A, B)
$\|C_{AB}\|$	The number of clients queries query (A, B)
W_{AB}	$\|Q_{AB}\|/\|C_{AB}\|$

compared with legitimate domain names are relatively high density of queries. The large amount of legitimate domain names on web service shows obviously smaller density than other services. Basic statistics factors and symbols what the system check in the traffic are the values listed in Table I.

A set of related domains showed similar values in the factors in the both of legitimate domains and botnet domains. It is caused from sharing of clients. Distinction of the legitimate domains and botnet domains can be achieved from the following condition. A client set of a large and famous legitimate domain may contain a large portion of botnet infected clients, but they have a lot of non-infected clients also, then it is hard to have the similarity on clients even though botnet infected clients queries the legitimate domain. Oppositely, if a large portion the clients of a legitimate domain are contained in a set of infected clients, the legitimate domains should be doubted. It is surly possible to some clients contact a same domain but is not that any few legitimate clients do not contact, if the legitimate domain has legitimate condition to the every clients.

B. Sequentiality Analysis

The second method is measuring temporal relationship of domain names. A bot shows programmed behavior on DNS queries. This approach is started from repetition and commonness of programmed queries. If multiple domain names are queried by a set of programs regardless whether it is malicious or not, it shows commonness on the order. We estimate the sequence of queries and the repetition of the sequential queries. W_{AB} stands for the repetition of the sequence between A and B. And $|C_{AB}|$ stands for the commonness of the sequences that queried by number of clients.

To consider the possibility of unexpected behavior between programmed queries by a human user or another programmed queries, we cumulate the sequential activities and filter the relatively rarely occurred sequences. Consequently, to make noise into the result, a significant numbers of infected clients must have same legitimate program which makes DNS queries, or an user who works like a program. Properly, programmed and regularly queries to a legitimate domain for noising is considered as a related domain because it is also one of the botnet behavior, and randomly queried legitimate domains by a botnet is omitted remaining malicious domains. In contrast with statistical result, representing sequential relationship among numbers of domain names is complex and too large in a simple list or a table. In our system, we adopt graph structure to represent and analyze this concept. Figure. 2 is a pseudo

Algorithm : Query sequence graph generation

Symbols
A: Client Set, D: Domain Set, Q: Query Set
G: Graph, N: Node, E: Edge
A = {a}, D = {d}, Q = {q | q := (a, d)}
G = {N, E}, N = {n | n := d}, E = {e | e := (d_1, d_2, w)}

INPUT : Q
OUTPUT : G

MAP<Key, Value> := < IP, Previous Domain >
N<D, I> := < Domain, Index >
E<N_1, N_2, W > := < $Node_1$, $Node_2$, Weight >

LOOP (q_t, $t = 0 \sim i$, $i = |Q|$) {
  IF (MAP.Exist($q_t.a$)) {
    d_{prev} = MAP.GetValue($q_t.a$)
    d_{new} = $q_t.d$
    n_1 = N.GetIndex(d_{prev})
    n_2 = N.Exist(d_{new})

    IF (!n_2) { n_2 = N.Add(d_{new})
    } **END IF**

    IF (w = E.Exist(n_1, n_2)) {
      $w = w + 1$
      e = (n_1, n_2, w)
      E.Modify(e)
    } **ELSE** {
      $w = 1$;
      e = (n_1, n_2, w)
      E.Add(e)
    } **END IF**
  } **ELSE** { MAP.Add($q_t.a$, $q_t.d$)
  } **END IF**
} **END LOOP**

Fig. 2. An algorithm for constructing a sequence graph from a given DNS traffic

code of the sequence graph generation.

- Graph Structure
  Sequential relationship among domain names is represented on a directed graph. Each domain name as a node is connected with links if two nodes have sequential relation. A sequential query q_{AB} links node A and B with a directed edge. At each edge has $|Q_{AB}|$, $|C_{AB}|$, and $|Q_{AB}|/|C_{AB}|$ as internal data. And each node has statistics of own domain name as node values.

- Graph Filtering
  To refine the regular behavior of clients from a domain relationship graph, we filtered the graph according to statistical values that each nodes and edges has. The filtering thresholds of node and edge values are dependent on analysis purpose and monitoring time slice. Lower threshold remains less related domains and the higher remains only a few strongly related domains only. In our experiment, we set the edge values as the average query frequency per an IP, W_{AB}, and perform graph filtering with threshold 20 to data set both of T_m and T_n.

Figure. 3 is an example of tracked botnet domains from a real traffic. At the first step, sequence graphs are constructed from the queries from clients. At the next step, for measuring repetition and commonness of sequential queries, we make a domain relation graph by cumulating the sequence graph of each client. This domain relation graph shows entire relationship of queried domains from clients. In the example case, every client contact known malicious domains, and they share unknown domains which is connected to known

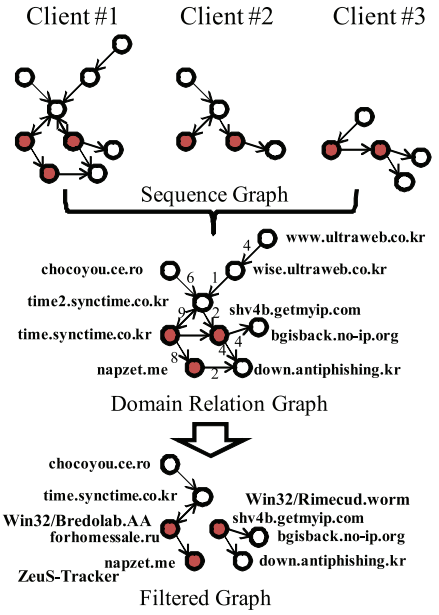

Fig. 3. Tracking example case: Three botnets and related domains

domains. At last step, the filtered graph is made from the domain relation graph by removing edges and domains with edge weight threshold. In the example case, we remove the edges which have weight less than 2. After the filtering, the filtered graph are separated to several components for each botnet or for a strongly related set of botnets.

In conclusion, the sequential analysis system distinguishes regular queries and random queries by cumulated behavior, and botnet related domains and normal domains by the reputation of their clients started from black list domains. A domain relation graph is structured by cumulating the sequence graphs of each client, and the relation graph is separated to botnet components by graph filtering. The domain relation graph shows overall relationship of domain names contacted by infect clients, and the filtered graph remains strongly connected relation which have high average query frequency and client share.

IV. Experiment and Observation Result

In this part, we discuss the experimental result of the proposed mechanism tested on the DNS traffic of the real network, analyze the topological and statistical characteristics which observed from some botnets and tracking abilities of our mechanism.

A. Experiment

1) Data Set: To show our tracking mechanism working well, we did experiment with DNS traffic captured on a real network. We had continuously captured DNS query traffic in front of a DNS server working on large sized network for two weeks. The DNS traces have near ten million of queries, one million of distinct domain names, and two hundred thousand of IP addresses per an hour in the daytime. As sample cases, we chose two sets of traces for detailed analysis, which were captured at midnight and noon. In this part, we call the two data sets as T_m, and T_n. T_m, the data set of the midnight, has just one third amount of queries as T_n has in the same duration. T_n has two hours of DNS traces, and T_m has six hours of DNS traces. Though the two data set set similar number of queries, IP addresses and domain names of each data set shows different volume.

TABLE II
Statistical similarity of botnet domains

Name	$\|C_D\|$	Disparity	$\|D\|$	W_D
$Xema(T_m)$	3K	0.1%	2	0.2K
$Xema(T_n)$	7K	1%	2	0.2K
$Tikayb(T_m)$	47∼ 50	4%	3	1.0K
$Tikayb(T_n)$	216∼223	1.3%	10	2.0K
$Pilleuz(T_m)$	40	0%	3	1.0K
$Pilleuz(T_n)$	56∼73	14%	3	2.0K
$Conficker(T_m)$	263∼307	10%	219	1.2K
$Conficker(T_n)$	422∼690	24%	245	1.2K
$Lethic(T_m)$	4	0%	9	0.6K
$Lethic(T_n)$	7	0%	9	0.3K

2) Statistical Analysis: At first, we measured statistics on T_m and T_n in terms of the client set. At the result, related domains showed similar values in every factors in the both of legitimate domains and botnet domains. Table II is statistics of known botnets in T_m and T_n. A botnet which has enough clients shows stableness on the number of clients and few disparity of client members among domains. In the case of *Conficker*, it has large gap between most and least cases, but each domain has just one or two gaps with close domains.

3) Sequentiality Analysis: The experiment of sequential analysis is also performed with data set T_m and T_n. The result of the sequential analysis is represented by a graph. Among the numerous components, we started to track the botnet related domains from a component which has a known botnet domain. We could extract near twenty components of known botnets from the three hundred of components. To visualize the graph, we used a graph tool *Pajek* [10]. Figure. 4 is visualized known botnet domains with their sequential relationship. We got a large component which connects several botnets domains and incidently linked domains without graph filtering. To remove incidental data, we set the query count filter to the graph. We will discuss about the graph filtering in the next part.

Topology of traced components showed a strict line, poly connected, or a particular shape. From this experiment, we can check the related domains what drove users to the malicious domains and where users driven after then. In some component, we can also find that there is an entrance domain what is in front of a set of domains which have complex connections. A strict sequence of queries shows a line, but in some cases, it makes a ring when the sequence is repeated from the first domain. By a repetition of the sequence, some component have a outer ring started from a domain. In a observed case, the

members of the ring was sequentially queried advertisement domains, and it was started from a botnet domain.

According to the particular features of each botnet, they are shown in different topology. For example, *Trojan:Win32/Lethic.B* shows a line topology with its multiple C&Cs. And the *Conficker* and *TrojanProxy:Win32/Tikayb.A* shows randomly queried and fully connected component with a several entrance domains. *Trojan-Downloader.Win32.Piker.jk* shows a bridge-like topology between two botnets, and has obviously high edge weight between their own domains, compared with other part of the component.

B. Observation and Analysis

From the experimental result, we could find the known botnet domains and their clients. Starting from the infected clients, we performed the botnet tracking with the experimental tools. As a result, we could make some groups of domains which had multiple C&C domains as their members, and some other groups that has some kinds of distinct botnet domains within one group. In terms of graph topology, related domain names made statistically similar links and they would be a component by graph filtering.

The most important observation results gotten from the

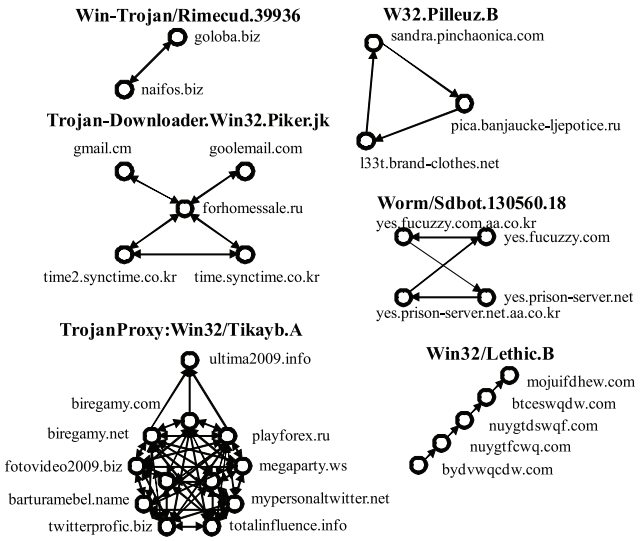

Fig. 4. The sub graphs for botnet domains from the data set T_m

component including a known botnet domain are a list of multiple C&Cs and a list of related malicious domains where drive users to the infection domain of the botnet by phishing, and also where redirect users to illegal advertisement domain or an infection domain of another botnet.

1) Known Botnets: The sequential analysis is starting from already known botnets in black lists. Even though many of them are on the black list and isolated by DNS sinkhole systems, some infected clients continuously try to connect to their C&C and generate DNS queries. From the DNS traces of known botnets, we can get the two information immediately, a set of bot-infected clients, and additional malicious domain names from the infected clients. From the basic approach of our mechanism, the sets of clients are connected with commonly queried domain names, and the set of domain names are connected by shared clients. From the test DNS traffic and currently updated black list, we could observe tens of clearly appeared known botnet domain names within both of thirty minutes and a few hours of DNS traces.

Within the list of known botnets, the statistically abnormal behavior of the domain names of *Win32/Tikayb.A* has been observed before December 2009, but they registered on the public blacklist near February 2010. Domain names of *Conficker* has also been fast-fluxed, and many of them are not in the blacklist.

2) Group of Botnet domains: A group of botnet domains are structured by mainly two reasons, multiple C&C domains and cross-infected clients. From the component of known botnets, we could recognize several group of cross infected clients which make links between two different botnet domains. In many other cases of the domain groups, the member domains were multi C&C domains of one same botnet.

The number of multiple C&C is represented as the size of graph in the sequential analysis. The graph shows a subset of entire list of C&C domains in a short traces, but it increase along with the trace time length. For instance, *Win32/Lethic.B* and *Win32/Alureon.gen!U* have tens of multiple C&C domains but not fully appeared in the short observation. In another case, *Win32/Tikayb.A* shows lots of member domains, but it is changed in each trace time. Moreover, the infected clients of the botnet infected by another botnet whose domain names is not on the blacklist.

Figure. 5 is the domain relation graph of data set T_n. The graph shows entire domain names including advertisement and spamming domains connected with botnet domains. The botnets in the graph are related to hundreds of domain names including other botnets. The width of edges on the graph shows weigh of each edge W_{AB}. In the center of the graph, botnet domains are connected to malicious domains such as phishing domains, and malware infection domains. The two sets of cycled nodes mainly observed at the lower side of the graph are mainly gambling and adult domains.

C. Discussion

From the observation results using our tracking mechanism, we could find the two evidence of botnet survivability. multiple C&C and cross-infection. Many of the distinct domains of botnets were well-known by anti-virus vendors and security agencies. But their cross-infection domains are not so well, and additional illegal domains which harass the users of the infected clients are not dealt with the botnet related malicious domains. In addition, a botnet which is regenerating new C&C domains on time, and it should be tracked continuously, not limited on onetime snapshot and blacklisting.

For the real-time analysis on the ISP level DNS, the time complexity of the mechanism is one of the challenges. The time complexity of major graph construction algorithm is

Fig. 5. The graph of malicious domains including botnets taken from T_n

based on the amount of queries within a pre-defined unit time . For N of DNS queries, N times of graph searching and modification is performed. The searching on the graph with the hashed node identifier takes constant time and their addiction and modification also takes constant time. Therefore, the time complexity with N queries is $O(N)$. The hashed map for a graph management has an advantage for time complexity but requires space complexity instead.

Another important problem is keeping the privacy of clients. The graph structure of our method enable to keep the anonymity of client activities from recording and identifying. The method recorded and analysis only the statistics and the sequences of domains to the graphs. During the analysis and tracking time, there are no recognizable relation between IP addresses and domain names. Practically, our tracking method does not be effected on the result with IP anonymized DNS logs and traffic if it keeps distinctness of clients.

V. Conclusion

Recent botnets are increasing their survivability against detection and quarantine methods. In this paper, to respond these evaluation of botnets, we proposed a tracking mechanism of botnet domains and clients focusing on the botnet related domains as a factor which increases the botnets survivability. The proposed multi-domain tracking mechanism based on domain reputation analyzes the statistical and sequential behavior of DNS clients to classify botnet related domains. In our experiment performed with a DNS trace data captured from an ISP DNS, the tracking mechanism shows the statistical and sequential similarity among the known botnet clients and unreported relationships among suspicious domain names and botnet domain names. The major contribution of this study is expansion target area to the indirectly related domains out of C&C domains that the prior botnet detection mechanisms focused on by connecting multiple domains as one component of domains and gathering the distributed activities of botnet members.

Acknowledgment

This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-(C1090-1031-0005)) and Seoul R&BD Program(WR080951).

References

[1] W. Paul, B. Dan, N. Mat, Z. Jason, J. Nicholas, L. Martin, and L. Daren, "Messagelabs intelligence:2009 annual security report," MessageLabs, Tech. Rep., 2009.

[2] R. Villamarín-Salomón and J. C. Brustoloni, "Bayesian bot detection based on dns traffic similarity," in *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*. New York, NY, USA: ACM, 2009, pp. 2035–2041.

[3] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by capturing group activities in network traffic," in *COMSWARE '09: Proceedings of the 4th International ICST Conference on COMmunication System softWAre and middlewaRE*. New York, NY, USA: ACM, 2009, pp. 1–8.

[4] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing botnet membership using dnsbl counter-intelligence," in *SRUTI'06: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet*. Berkeley, CA, USA: USENIX Association, 2006, pp. 8–8.

[5] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in *CCS'09: Proceedings of the 16th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2009, pp. 635–647.

[6] R. Perdisci, I. Corona, D. Dagon, and W. Lee, "Detecting malicious flux service networks through passive analysis of recursive dns traces," in *ACSAC '09: Proceedings of the 2009 Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 311–320.

[7] G. Guofei, Z. Junjie, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic," in *NDSS'08: Proceedings of the 15th Annual Network and Distributed System Security Symposium*. Reston, VA, USA: ISOC, 2008.

[8] N. Shishir, M. Prateek, H. Chi-Yao, C. Matthew, and B. Nikita, "BotGrep: Finding p2p bots with structured graph analysis," in *Proceedings of the 19th USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2010, pp. 95–110.

[9] J. P. John, A. Moshchuk, S. D. Gribble, and A. Krishnamurthy, "Studying spamming botnets using botlab," in *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*. Berkeley, CA, USA: USENIX Association, 2009, pp. 291–306.

[10] "Pajek," http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[11] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, and E. Gillum, "Botgraph: large scale spamming botnet detection," in *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*. Berkeley, CA, USA: USENIX Association, 2009, pp. 321–334.

[12] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in dns traffic," in *CIT '07: Proceedings of the 7th IEEE International Conference on Computer and Information Technology*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 715–720.

[13] J. A. Morales, A. Al-Bataineh., S. Xu, and R. Sandhu, "Analyzing DNS activities of bot processes," in *MALWARE'09: Proceedings of the 4th IEEE International Conference on Malicious and Unwanted Software*. IEEE, 2009, pp. 98–103.

[14] F. Leder, T. Werner, and P. Martini, "Proactive botnet countermeasures:an offensive approach," in *Proceedings of the Conference on Cyber Warfare*, 2009.

[15] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *CATCH'09: Proceedings of 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, 2009, pp. 299–304.