Botnet Visualization using DNS Traffic *

Inhwan Kim, Hyunsang Choi and Heejo Lee

Korea University, Seoul 136-713, South KOREA
{neutrino37, realchs, heejo}@korea.ac.kr

Abstract. One of the major challenges for network security is the botnet. It is one of the major causes of network threats such as spam, DDoS(distributed denaial-of-service) attacks, and so on. To be sure, there have been studies specifically concerning botnet detection, but most of these studies can detect specific types of botnets only with an offline analysis making it hard to respond to a botnet immediately. In this paper, we describe our development of a visualization mechanism using DNS (Domain Name System) traffic. The goal of our mechanism is to provide a network administrator with meaningful visual information allowing the administrator to detect botnets intuitively. We can reveal botnets from DNS traffic. And our mechanism can afford to operate in real-time, because the DNS possesses a small amount of network traffic. The mechanism is comprised of parallel coordinates to describe a botnet in an intuitive graphical pattern. The coordinates represent three different parameters in a DNS packet. The color of a line in parallel coordinates is determined by statistical values from the cumulative series of duplicated DNS data over time. We define four patterns of graphs as a signature for the visualization system. And we have demonstrated that our mechanism shows the effectiveness in revealing many important aspects of real-world botnet patterns by experiments on /16 campus network traces.

1 Introduction

Botnets are one of the most important security problems to resolve. A botnet is a type of malicious code that is controlled by the owner of the botnet (generally called a botmaster). Botnets perform malicious activities involving spam, Internet worms, DDoS attacks, click fraud, and so on. Numerous research efforts offer suggestions on detecting the botnet. However, most approaches can detect specific types of botnets only with offline analyses, making it hard to response to the botnet immediately.

One promising approach is visualization, using a simple and intuitive method, to handle complex situations. Visualization is becoming an effective appliance to assist in network security. Visual images can be obtained from raw data. From the images, we can acquire valuable insights. It is an efficient link, representing key technology for extracting information, from the human mind to the modern computer. This visual representation is becoming more and more essential in the field of Internet security.

* This research was supported by the Ministry of Knowledge Economy, Korea, under the ITRC support program supervised by the IITA(IITA-2008-(C1090-0801-0016)), the IT R&D program of MKE/IITA(2008-S-026-01) and partially supported by Defense Acquisition Program Administration and Agency for Defense Development(2008-SW-51-IM-02)

In this paper, we describe a visualization mechanism using DNS traffic. Our goal has been to design a visualization system that makes it possible for an administrator to see, at a glance, what kinds of anomalous activities have occurred on a monitoring network while maintaining the ability to also provide detailed information on the activity. The coordinates in the visualization system represent three different parameters in a DNS packet. The color of the connected lines uses statistical values by a cumulative series of duplicated DNS data over time. From the image on the parallel coordinates, we can find abnormal patterns, including a funnel, hourglass, diamond, and pentagon that represent different types of botnets. From the abnormal patterns of graph, we can detect real-world botnets from the DNS traffic in a B class campus network.

This study provides three contributions. (1)The proposed visualization enables network administrators to discover botnets and malicious activities intuitively. (2) With the visualization mechanism, we can immediately detect botnets in the large scale of the monitoring network before the botnets can perform abnormal behaviors. (3) The visualization can find malicious domains as well as infected hosts. Therefore, the network administrator can provide adequate countermeasures to the botnet.

The rest of this paper is organized as follows. In Section 2, we review related work. In Section 3, we describe our proposed visualization mechanism and some of its benefits. In Section 4, we describe our evaluation using real network traces. We summarize our results and conclude the paper in Section 5.

2 Related Work

There are several research efforts to find botnets using network anomaly based detection. Dagon presents a botnet detection and response approach [1] while analyzing the peculiarity of botnets rallying DNS traffic (particularly, measuring the canonical DNS request rate and making a DNS density comparison). However, Dagon's approach is inefficient because his approach generates many false alarms. BotHunter [2] models the botnet infection life-cycle as sharing common steps. BotHunter then detect botnets by employing IDS-driven dialog correlation according to a bot infection life-cycle model. However, any malware not conforming to this model would seemingly go undetected when using this approach. BotSniffer [3] is designed to detect botnets using either IRC or HTTP protocols. BotSniffer uses a detection method referred to as spatial-temporal correlation. This method relies on the assumption that all botnets, unlike humans, tend to communicate in a highly synchronized fashion.

Visualization, where there are several innovative approaches, is increasingly important in the field of network security to help analyze data by using human intuition. VisFlowConnect [4] utilizes a set of parallel axes viewable on a 2D screen. Each point on an axis represents an IP address (of a machine or domain). Connections between the points on the parallel axes represent network connections and the flow of data. NVisionIP [5] presents a visual representation of an entire class-B IP network on a single screen. The overview screen has horizontal and vertical axes listing all subnets of a network along with the top axis; the hosts in each subnet are listed on the vertical axis. The Spinning Cube of Potential Doom [6] extends the grid-based visualization techniques to a three dimensional volume where the axes represent source IP, destination IP, and

port. Then, port scans and network scans can both show up as distinctive lines, but in different directions.

Researchers have rarely proposed visualization approaches to handle botnets. There are only a few methods for lighting on clues of a botnet. The IDS rainstorm [7] shows alarm activity within a network. The information is presented with the local network IP addresses plotted over multiple y-axes representing the location of alarms. Time on the x-axis is used to show the pattern of the alarms; variations in color encode the severity and amount of the alarms. DNS visualization [8] is a visualization approach addressing DNS security challenges, such as distributed denial of service (DDoS) and cache poisoning attacks. Krasser et al. [9] suggest combining the strength of link analysis using parallel coordinate plots with time-sequence animation of scatter plots. They examine a 2D and 3D coordinated displays that provide insight into both legitimate and malicious network activity.

3 Botnet Visualization using the DNS

3.1 The DNS used by the Botnet

Generally, botnets use a DNS for many situations. Centralized botnets use a DNS to perform regular DNS queries and to join in a communication channel. Candidate C&C servers can be a replacement from the broken C&C server. They use a dynamic DNS (DDNS) [10] a resolution service that automatically changes the IP address of a server, that substitutes the DNS record by frequent updates and changes, to keep the botnets mobile. Botnets frequently migrate their C&C server, either by being instructed to move to a new IRC channel/server or to download replacement software that points them to a different C&C server.

Here, we describe five cases where DNS is used in a botnet : (1) at the rallying procedure. If the host infection succeeds, the infected hosts should be gathered. They use the DNS to assemble. (2) at the update stage. The botnets usually update their code with the latest one by downloading it from their Web server for evading the newest detecting mechanism or adding new functions. (3) at the malicious behaviors of the botnet. Several types of malicious activities are accompanied with the DNS transmission. For example, in spam mailing, bots send a DNS query of mail server to the DNS server. (4) at botnet cloning and reconnecting. The botmaster uses instructions of cloning and reconnecting the botnet, since the botmaster wants to make their botnet more robust and undetectable. (5) at the C&C server migration. The botnet migrates from one to another candidate C&C server. At that moment, a DNS query is also used to find a new C&C server.

3.2 Botnet Characteristics

In order to draw a visual mechanism for a botnet, its characteristics must be considered in terms of visualization. Fortunately, the botnet DNS traffic has a characteristic which can be distinguished from a legitimate DNS [11]. Table 1 shows differences between the botnet and legitimate DNS traffic. Only infected hosts send queries to the

	Group Consistency	Periodicity	Intensity
Botnet DNS	Relatively consistent	Periodically regularly appeared	Intensively appeared
Normal DNS	Fluctuated group (anonoymous)	Intermittently appeared	Randomly moderately appeared

Table 1. Difference between botnet and normal DNS

domain name of a botnet channel server and the legitimate user never makes queries of the server. Those botnet members (infected hosts) always act as a group with the members relatively unchangeable. Here, relatively means that there could be trivial changes, including temporally deviated member(s), removed member(s), and recruited member(s). Although these changes have an effect on the botnet group, it can be ignored within a short time period based on the robustness of the botnet and botnet propagation model [12]. The dynamics of the IP addresses can affect the uniformity of the botnet group, but it also can be disregarded within a certain period. Xie et al. in [13], indicated that more than half (61.4%) of the IP addresses observed are dynamic IP addresses, but they also showed that over 30% of the IP addresses have inter-user durations ranging between 1 and 3 days and over 95% of the IP addresses have durations longer than 1 hour. Therefore, deviations from the dynamics of the IP addresses can be ignored if we monitor the network within a short time. Therefore, a group of anomalous DNS traffic can be considered as a botnet. We also find that the botnet DNS traffic, usually occurs periodically, regularly, and intensively. Therefore, these temporal botnet properties can also be used to detect a botnet.

3.3 Selected Parameters in the DNS

There are two types of DNS packets: query and response [14]. A DNS response packet includes DNS query data so we use only DNS response packets as shown in Fig. 1. We select a total of 6 parameters from the DNS packet. (1) Host IP: Destination IP in the IP header field implies a host that sends a DNS packet. (2) Target domain name: There is a NAME field in the Answer RR fields that is chosen to represent a queried domain name. (3) Target IP: We select the IP address of the domain name. In the Answer RR fields, there is a RDATA having a variable length for the RR(Resource Record). If the TYPE is A, the RDATA is a 4 octet ARPA Internet address. (4) TTL: TTL, in the Answer RR fields, specifies the time interval that the RR may be cached before the source of the information should again be consulted. It is helpful to find Dynamic DNS(DDNS) are fast flux [15] which are related to the botnet. (5) TYPE: Generally, the botnet uses DNS to resolve the C&C server, launch a DDoS attack, and send spam. Therefore, we only deal with A (A host address) and CName (Canonical name) queries according to the

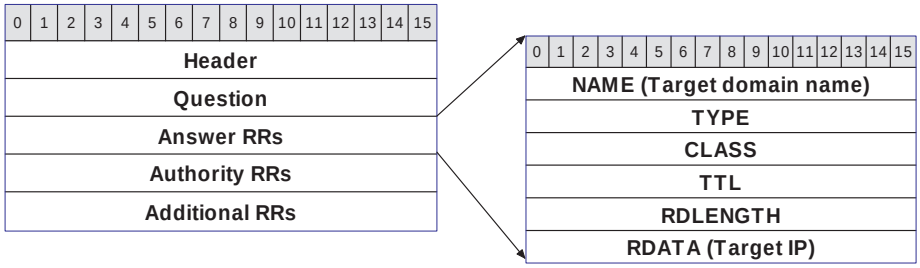

Fig. 1. DNS protocol fields

IANA assigned numbers of the DNS parameter [16]. (6) RCODE: RCODE in the DNS header field is used to check an error in the DNS response packets. The parameters can be classified into three different groups. Main parameters (1), (2), and (3) are used to draw a line on the parallel coordinate. Sub parameter (4) is to decide the color of the line. Filter parameters (5) and (6) are selected to filter out the DNS packets that we use in visualization.

3.4 Visualization System Design

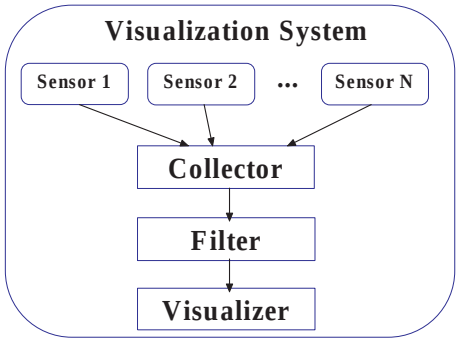

Fig. 2. System design of the visualization system

In this section, we describe our visualization system architecture. The visualization system consists of four different parts as shown in Fig. 2: (1) sensor(s), (2) a receiver, (3) a filter, (4) a visualizer. The sensor(s) collects DNS traffic data and sends the data to the receiver. The receiver stores the data into buffer(s) and delivers the data to the filter periodically. The system only deals with the A (or CName) type DNS which has no error. Finally, the visualizer draws a graph using the extracted data. In order to adequately

view the parameter relationship, we use a parallel coordinates' view [17]. With this visual representation, we plot data from an arbitrary number of axes onto a two dimensional view. This visualization uses three parallel axes with each axis corresponding to the parameters: host IP, the target domain name, and the target IP from the top to bottom. With the value of the parameter, we decide points on each coordinate that a chain of connected lines should cross.

The top axis (called the host IP) denotes the destination IP address (0.0.0.0 $\sim$ 255.255.255.255) of the observed packets. The bottom axis (called the target IP) also represents the IP address. We chose to implement these values by diminishing them so as to be the length of the axes. The middle axis denotes the target domain name. We use a hash function to estimate the location value of the domain name on the middle axis, Eq.(1)

$$H(DN) = \omega \times h(SLD + TLD) + h(3LD) \tag{1}$$

Each variable, TLD, SLD and $3LD$ represents the top, second, and the third level name of the domain DN. To draw the data in limited space efficiently, we devise a modified hash function $H(DN)$. The function distributes third level domains uniformly and closely locates a similar domain name (that has that same top and second level domain names) with the weight value ω. A hash function $h(S)$ produces a hash value from a string. For example, a hash function $hashCode()$ is provided by a String class in Java. Its equation is Eq.(2).

$$h(S) = \sum_{i=1}^{n} (S_i \times 31^{n-i}) \tag{2}$$

We use Eq.(3) to determine the color of a line using abnormal rate $R(x)$.

$$R(x) = \alpha \times T + \beta \times G + \gamma \times F + \delta \times P \tag{3}$$

α, β, γ, δ are the given weights of each variable T, G, F and P. The variable has a boolean value. T implies the dynamicity of a target domain name. The dynamicity is decided by the TTL value. If the TTL value is lower than a given threshold [18], the domain is dynamic and we set T to 1. G denotes the size of the domain group queried from many hosts. We record for each domain how many hosts send a DNS query to the domain. The number of hosts is regarded as group size G. If group size exceeds a given threshold, we set G to 1. Frequency F denotes the querying frequency of each host, that is, how many queries are sent from the host to the target domain. If the querying frequency exceeds a given threshold, F is set to 1. The periodicity, P, represents how periodically a host sends queries to the target domain. Here, periodicity can be measured to check the similarity of the DNS querying time intervals. If the similarity is smaller than the given threshold, we set P to 1. These statistical values are estimated by a cumulative series of duplicated DNS data over time. From the equation, we get the abnormal rate $R(x)$. If $R(x)$ exceeds a threshold, the color of the connected line becomes red.

The proposed visual representation is valuable because it shows prominent trends, correlations, and divergences from the raw data. It shows unexpected intensity, unexpected IP address space, unexpected interactions between IP addresses and domain

Signature	TTL, Group size, Frequency, Periodicity				Description
	Low	*	High	Regular	**Botnet C&C, DRDoS (DDNS)**
	Low	*	Low	Regular	**Botnet C&C (DDNS)**
	High	High	High	*	**DRDoS, Broken botnet C&C**
	Low	*	High	Regular	**Fast fluxed botnet C&C**
	Low	*	High	Regular	**C&C with multiple domain name**
	*	High	High	Regular	**Filtered or broken C&C**

(* : any)

Table 2. Graphical signatures of a botnet

names. It also groups similar lines together to highlight them and emphasize their patterns using different colors. This representation enables us to gain critical insight into the DNS in traffic and establish reliable intuitive hypotheses. Even if an unknown anomaly occurs, a specific image pattern can be gained and the anomaly can be detected in a timely manner.

3.5 The Graphical Signatures of a Botnet

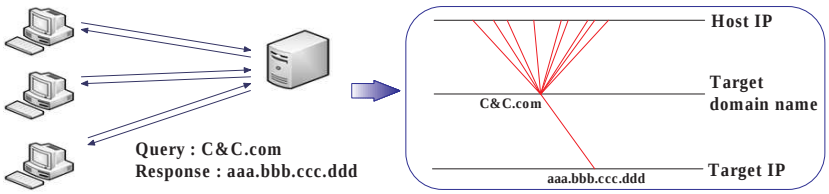

Fig. 3. An example of botnet visualization

Now we show how parallel coordinates can be used to describe a botnet in an intuitive graphical pattern. The coordinates represent three different parameters in a DNS packet. The first represents the IP address host; the second, the target domain name; the third, the target IP. These three values enable the packet to be plotted as a connected line on parallel coordinates. We determine the color of the connected lines using statistical values by a cumulative series of duplicated DNS data over time.

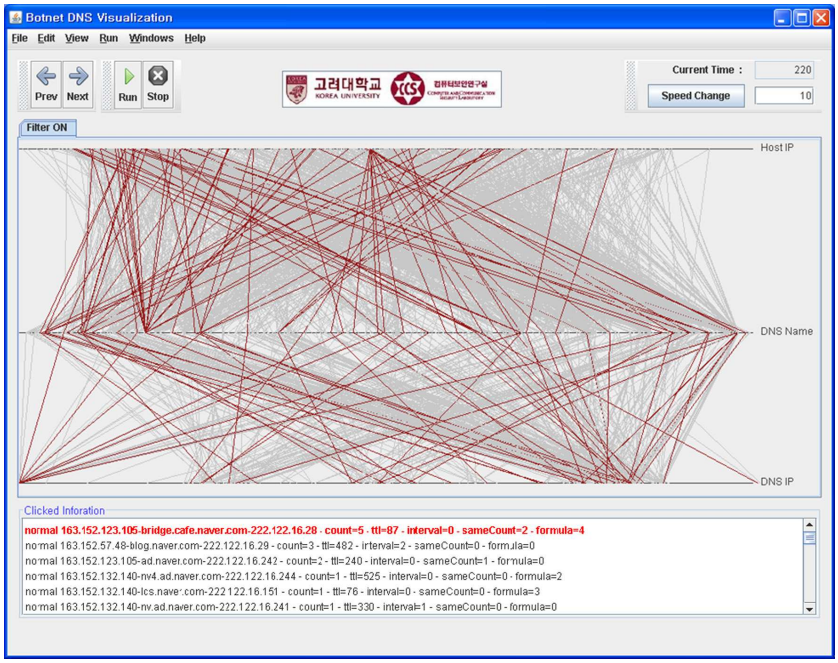

Fig. 4. Overview of the visualization

As above, we draw a graph on parallel coordinates and get valuable patterns of graphs that we use to define the graphical signatures of the botnet, shown in Table 2. All of the signatures have red colored lines due to the parameter values described in the table. There are 4 visual signatures: funnel, hourglass, diamond, and pentagon patterns possibly interpreted as six situations.

A funnel-like pattern indicates that infected hosts try to find the IP address of the C&C channel or launch a DDoS attack using DNS as shown in Fig. 3. The infected hosts send queries to the DNS server to discover a communication channel. The response from the DNS server includes parameters. If we draw a graph on parallel coordinates using the parameters, we can get a funnel-like pattern graph. The funnel-like patterns can also appear in normal cases, but we measure the abnormal rate and apply this rate to the color of the line. Therefore, only anomalies are shown as red lines. If the red lines have a pattern described in Table 2, this pattern can be considered as a botnet C&C pattern. If the TTL is short, it would then be considered a dynamic DNS domain. If the frequency is high, it can also be considered as a DRDoS attack where the host launches a DoS attack using a reflection of the DNS server [19]. We also observe a case of a broken C&C where there is a group of infected hosts, but they cannot connect to their C&C channel. Fast-flux is also usually related to a botnet [15]; often the infected hosts have numerous target IP addresses resolved to the same domain name. In this case, the graph forms the hourglass-like pattern. On the other hand, if C&C uses multiple

domains, the graph has a diamond-like pattern. Finally, if the domain name of the botnet C&C is broken or filtered, we can get a pentagon-like pattern graph.

3.6 Prototype Visualization

Fig. 4. shows an overview of the visualization. The visualization system runs on Microsoft Windows XP with Intel Pentium Core 2 Duo (2.66GHz, 2.67GHz) processor and 2 GB of RAM system. Our system provides a main view that presents an overall network using colored DNS lines. There are two states in the main view. In a total view state, we can see from all lines that normal situation lines are gray and abnormal situation lines are red. When we click a point on a coordinate, the system changes the main view from total view state to detailed view state. In a detail view state, we can see clicked lines which are related to a kind of main parameters (host IP, target domain name or target IP). And the system reports detailed information of clicked lines that not only main parameters but also sub parameter, statistical values and an abnormal rate at the clicked information view.

3.7 Experiment Result

In order to show the effectiveness of the proposed method, we get experimental results at host infection using a bot code and at live traffic in a campus network.

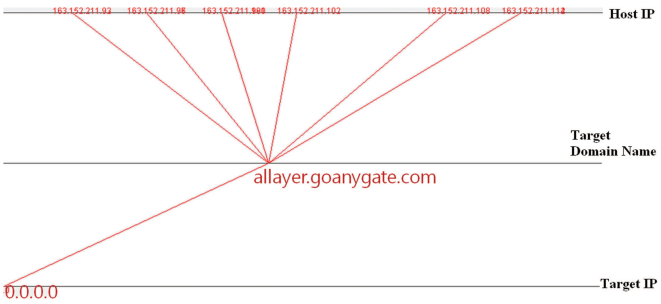

Fig. 5. The botnet built-in testbed

We created a modified Agobot code and composed a botnet with 24 infected hosts in the testbed network. We tested several commands to perform attacks such as UDP flooding and ICMP flooding. In the experiment, we used a dynamic DNS domain *'allayer.goanygate.com'*. We can get a graph such as that shown in Fig. 5. This graph is represented as a funnel-like pattern. Zombie hosts make DNS queries requlary to find their C&C server. The access pattern of zomibies can be appeared as a funnel- like pattern (Fig. 5). In many normal cases, a funnel-like pattern could also be appeared. However, DNS queries from the zombie hosts are different from normal cases. Zombies queries has 30 second of TTL value and appeared simultaneously and frequently

(24 zombie hosts send same queries every 8 seconds). Using the Eq.(3), the properties of them make a funnel-like patterns in red lines. Consequently, we can differentiate the red line patterns from numerous noisy gray lines. We extract and rescale the red lines as intuitive result of botnet (zombies) detection.

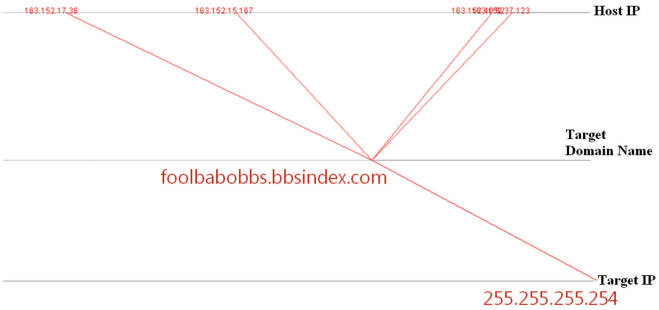

Fig. 6. The known domain of the botnet (Virus.Win32.AutoRun.fw)

We also obtained DNS traces tapped from the gateway router of the 1Gb/s campus network that has a B class addresses, on May 19th, 2008. The captured traces only included 6.28GB of the DNS traffic estimated at 0.58Mbps. 19.52 million DNS queries were captured. In this experiment, we found several pieces of evidence indicating a botnet having graphical patterns from the DNS traces.

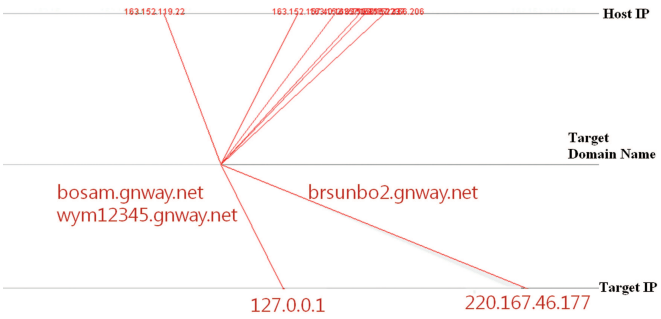

Fig. 7. The unknown domain of the botnet

We found a known botnet pattern from the DNS traces shown in Fig. 6. The domain name was *'foolbabobbs.bbsindex.com'* involved in Virus.Win32.AutoRun.fw as it was defined by Kaspersky and Infostealer defined by Symantec [20]. The IP address of the domain was 255.255.255.254. Four hosts requested the domain. The query had 54

seconds of TTL value and 15 seconds for the querying time interval. This was also represented as a funnel-like pattern. It turned out that the domain was used in a botnet from our manual inspection.

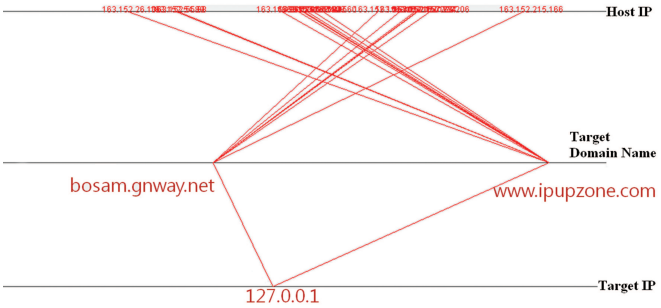

Fig. 8. The filtered domain of the botnet

'bosam.gnway.net', *'wym12345.gnway.net'* and *'drsunbo2.gnway.net'* had low TTLs and regular querying time intervals. They indicated a funnel-like pattern as shown in Fig. 7. *'bosam.gnway.net'* and *'wym12345.gnway.net'* resolved into a loopback IP address (127.0.0.1) and *'drsunbo2.gnway.net'* resolved into 220.167.46.177. Only a particular group of hosts requested these domains names. We inferred that these were unknown domain names of the botnet.

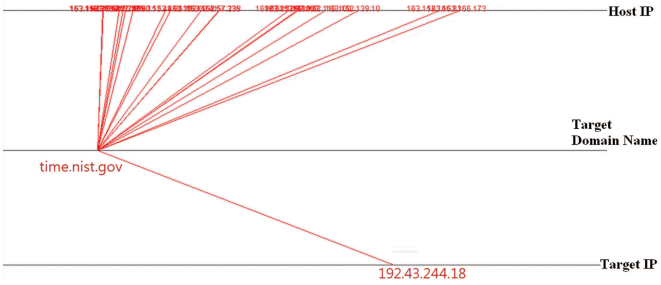

Fig. 9. Time synchronization domain

We found a pentagon-like pattern from the DNS traces as shown in Fig. 8. The resolved IP address of *'bosam.gnway.net'* and *'www.ipubzone.com'* had a loopback IP address (127.0.0.1). These domains were filtered by the blacklist of the DNS server.

We also observed a funnel-like pattern as shown in Fig. 9. However, we found that the domain (*'time.nist.gov'*) was a legitimate one providing a time synchronization service. Another domain for time synchronization (*'clock.iptime.co.kr'*) was also shown

as this pattern. Most hosts made more than three queries in every second. Therefore, this pattern can be regarded as evidence of a DRDoS attack or queries for experimental purposes.

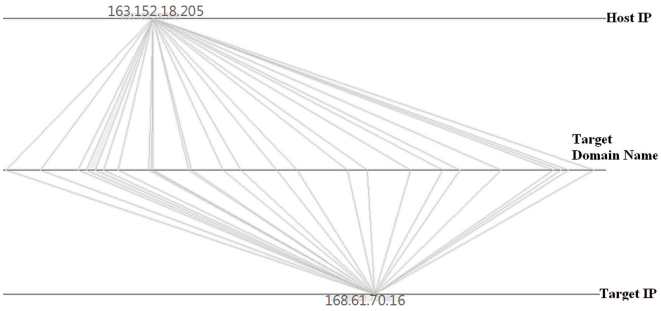

Fig. 10. Observed diamond-like pattern

Fig. 10. shows an example of the diamond-like pattern observed in the DNS traces. There were 62 different domain names that were resolved into the same IP address. The domains had the same third level domain *'mx1'* but a different top and second level domain such as *'mx1.cyber-group.com'*, *'mx1.willieswinners.com'*, *'mx1.internet33.com'* and so on. Therefore, our visualization system drew a diamond-like pattern. However, the color of the line was not red since they had high TTL, low frequency, and irregular periodicity. These queries were generated for the purpose of experimentation.

4 Conclusions

The botnet has become one of the most important security problems to resolve . While there have been studies concerned specifically with botnet detection, most approaches detect specific types of botnets, after the fact, making it hard to immediately respond to these detected botnets through proposed this mechanisms. In this paper, we described our development of a visualization mechanism using DNS traffic. The proposed visualization enables network administrators to discover botnet and malicious activities intuitively. We can immediately detect botnets through a large-scale monitoring of networks before the botnets perform anomaly actions. The visualization can find malicious domains as well as infected hosts. As a consequence of this visualization, a network administrator can provide adequate countermeasures to the botnet. The coordinates represent three different parameters in a DNS packet. The color of the connected lines demonstrates statistical values by a cumulative series of duplicated DNS data over time. From the image on parallel coordinates, we define four patterns of graphs, the funnel, hourglass, diamond, and pentagon representing different types of botnets. We find significant evidences of botnets having graphical patterns from the campus network DNS traces. It is shown that our visual approach can detect not only known botnet but also

unknown botnet. And we demonstrated that our mechanism shows the effectiveness in revealing many important aspects of botnet pattern by the experiments on real-life network trace.

References

1. Dagon., D.: Botnet detection and response. In: OARC Workshop. (2005)
2. Gu, G., P. Porras, V. Yegneswaran, M.F., Lee., W.: Bothunter: Detecting malware infection through ids-driven dialog correlation. In: USENIX Security Symposium (Security'07). (2007)
3. Gu, G., Zhang, J., Lee., W.: Botsniffer: Detecting botnet command and control channels in network traffic. In: Network and Distributed System Security Symposium (NDSS'08). (2008)
4. Yin, X., Yurcik, W., Treaster, M., Li, Y., Lakkaraju., K.: Visflowconnect: Netflow visualizations of link relationships for security situational awareness. In: ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC'04). (2004)
5. Lakkaraju, K., Yurcik, W., Lee., A.J.: Visflowconnect: Netflow visualizations of link relationships for security situational awareness. In: ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC'04). (2004)
6. Lau., S.: The spinning cube of potential doom. In: Communications of the ACM. (2004)
7. Abdullah, K., Lee, C., Conti, G., Copeland, J., Stasko, J.: Ids rainstorm: Visualizing ids alarms. In: ACM Workshop on Visualization for Computer Security (VizSEC'05). (2005)
8. Ren, P., Kristoff, J., Gooch., B.: Visualizing dns traffic. In: ACM Workshop on Visualization for Computer Security (VizSEC'06). (2006)
9. Krasser, S., Conti, G., J. Grizzard, J.G., , Owen., H.: Real-time and forensic network data analysis using animated and coordinated visualization. In: IEEE Workshop on Information Assurance. (2005)
10. Vixie, P., Thomson, S., Rekhter, Y., , Bound, J.: Dynamic updates in the domain name system (dns update) http://www.faqs.org/rfcs/rfc2136.html.
11. Choi, H., Lee, H., Lee, H., Kim, H.: Botnet detection by monitoring group activities in dns traffic. In: Computer and Information Technology (CIT'07). (2007)
12. Dagon, D., Gu, G., Lee, C., Lee, W.: A taxonomy of botnet structures. In: Annual Computer Security Applications Conference (ACSAC'07). (2007)
13. Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., Wobber, T.: How dynamic are ip addresses? In: Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'07). (2007)
14. Mockapetris, P.: Domain names - implementation and specification http://www.ietf.org/rfc/rfc1035.txt.
15. Zdrnja, B., Brownlee, N., , Wessels, D.: Passive monitoring of dns anomalies. In: Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'07). (2007)
16. Internet Assigned Numbers Authority (IANA)Cooperation : Domain Name System Parameters http://www.iana.org/assignments/dns-parameters.
17. Inselberg, A.: The plane with parallel coordinates. In: The Visual Computer. (1985)
18. Dynamic Network Services Inc.: DNS Caching http://www.dyndns.com/support/kb/dns_caching.html.
19. Courses, E., Surveys, T.: Motivation for behaviour-based dns security: A taxonomy of dns-related internet threats. In: Emerging Security Information, Systems, and Technologies (SECURWARE'07). (2007)
20. ThreatExpert Report: Virus.Win32.AutoRun.fw, Infostealer, W32/CWT.worm http://www.threatexpert.com/report.aspx?uid=4763a123-3245-4beb-b959-0b8a3274d0eb.