Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080530

A Hybrid Defense Technique for ISP Against the Distributed Denial of Service Attacks

Young Hoon Moon[1], *Suk Bong Choi*[1], *Huy Kang Kim*[2] *and Changsok Yoo*[3,*]

[1] Security Control Team, Cyber Security Center of Korea Telecom, Republic of Korea
[2] Graduate School of Information Security, Korea University, Seoul, Republic of Korea
[3] Department of Cultural Tourism Contents, Kyung Hee University, Seoul, Republic of Korea

Received: 2 Sep. 2013, Revised: 30 Nov. 2013, Accepted: 1 Dec. 2013
Published online: 1 Sep. 2014

Abstract: As malicious traffic from botnets now threatens the network infrastructure of Internet Service Providers (ISPs), the importance of controlling botnets is greater than ever before. However, it is not easy to handle rapidly evolving botnets efficiently because of the highly evolved detection avoidance techniques used by botnet makers. Further, nowadays, Distributed Denial of Service (DDoS) attacks can compromise not only specific target sites but also the entire network infrastructure, as high-bandwidth Internet services are now being provided. Thus, ISPs are deploying their own defense systems to prevent DDoS attacks and protect their network infrastructure. However, the new problem ISPs confront is that botnet masters also try to destroy their defense systems to make their attack successful. ISPs can mitigate DDoS through botnet-specific management by taking preemptive measures, such as the proactive reverse engineering of suspicious code and the use of honeypots. This paper illustrates an advanced DDoS defense technique for the use of ISPs with a real case study of the technique's implementation. This technique was proven very effective method for controlling botnets, and we could confirm this effectiveness in a real ISP environment.

Keywords: Security, Botnet, Internet Service Provider, Distributed Denial of Service attack

1 Introduction

With the unprecedented increase in the number of Internet users, Internet Service Providers (ISPs) are required to provide them with high-quality services as well as high bandwidth connections to remain competitive. Ironically, this enormously expanded network infrastructure is also a favorable environment for malicious attackers who wish to launch Distributed Denial of Service (DDoS) attacks. The impact of DDoS attacks could damage not only innocent victims, but also the ISP itself. Thus, the attack could lead to slowing down of Internet services due to the excessive workload on an ISP's service routers.

With ISPs providing high-bandwidth Internet services at low cost to subscribers under a competitive marketing environment, the volume of the traffic in a DDoS attack that is detected by ISPs is increasing. Hence, as time goes by, the total volume by attacks will increase. Thus, many ISPs have to deploy their own defense system to block or mitigate high-bandwidth DDoS attacks.

However, it is almost impossible to block DDoS attacks perfectly using a legacy defense system, because botnets are improving at a very fast rate, with which such a system cannot keep pace. The defense systems in ISPs also need to be renovated to be able to deal with any rapidly evolving botnet. In this paper, we introduce a very significant case study from one of the biggest ISPs in South Korea. This ISP has already built a honeypot system and constantly analyzes the suspicious code collected by it.

In this case, the honeypot system plays a significant role in detecting and responding to botnet activity. Actually, a bot-specific defense system with a honeypot and continuous proactive reverse engineering of suspicious code captured by this honeypot could be an effective countermeasure to botnets causing massive DDoS attacks. The honeypot can also be used for analyzing botnet network activities. This system requires only a small monetary investment, but its result is very effective. We can confirm that, as a result of the system, the total DDoS attack size can also be significantly reduced.

* Corresponding author e-mail: csyoo@khu.ac.kr

In this paper, we introduce recent attacker defense techniques, show how a typical ISP responds to these problems in a real-life situation, and suggest ideas pertaining to defending an ISP against a DDoS attack.

2 Related Works

The DDoS defense mechanism has been significantly diversified ever since the concept of DDoS attack was first introduced. DDoS-related survey papers have dealt with numerous concepts and research trends over the decades [7]-[9] [26] [8]. The surveys have considered different viewpoints and standards for the classification of DDoS attacks and defenses over time. In this section, several DDoS defense mechanisms are reviewed and compared for practical use in the ISP. Provos et al. introduced many types of botnet: spyware, command and control (C&C), and DDoS attack types [1]. Recent botnets include aspects of all these types [2] [5]. They receive orders from a C&C server, they can initiate a DDoS attack, and some of them take personal information from infected users PCs, all at the same time. To defend against these botnet attacks, two kinds of defense mechanism have been proposed according to the areas that they protect: the network-based and the application-based defense mechanism. The former approach consists mainly of three technical methods: congestion control [33] [34], network configuration [22] [35], and signature filters [16] [18] [36]. On the other hand, the types of application-based defense mechanisms are much more numerous: client-puzzle [37], IRC-based [23], anomaly-based [10] [27], DNS tracking [25], and attack traffic suppression [24]. In addition to these traditional defense methods, advanced defense methods, such as reverse engineering [20] and honeypot [29] [31], also exist. We divide our literature review below into three parts.

2.1 Network based Defense Mechanisms

The congestion control-based defense mechanism is a traditional approach to network- oriented DDoS attacks. Ioannidis and Bellovin proposed implementing pushback concepts in the router for defending against DDoS attacks. The authors addressed DDoS problems as network congestion. A router cannot distinguish between attack traffic and normal traffic, but only between heavy traffic and light traffic. Using this concept, each edge router that experiences congestion due to DDoS attacks communicates with other routers, informing them of the rate-limit traffic to the destination router [33]. Hu et al. also suggested applying packet filtering on network routers to limit malicious traffic by using time-windows [34]. These methods are quite effective against high bandwidth DDoS attacks and some of these concepts have already been implemented in network routers [33] [34].

The network configuration-based defense mechanism originated in methods that enhance the network availability by adding more network facilities or reconfiguring the logical network boundaries [22] [35]. Secure Overlay Service (SOS) [22] uses an overlay network to provide the server with good protection. However, the service's users experience a significant delay because transmission is routed on the overlay. Due to the additional access point (SOAP), the SOS approach originally had routing-related drawbacks.

Oikonomou et al. developed the overlay concept of defense mechanism, proposing DefCOM [35]. The authors present a defense system that is both collaborative and deployed widely by combining the advantages of end-to-end approaches, core defenses, and heterogeneous network systems. Notwithstanding the theoretical benefits of this collaborative defense, it is not easy for IPSs to cooperate with each other under circumstances where each ISP is competing fiercely for a market share [11] [21] [22] [35].

Liu et al. introduced a method for mitigating DDoS attacks using a feedback packet [16]. This method can also control network congestion and solve the spoofing problem by using cryptography techniques. It focuses on an end-to-end system, such as the sender or the receiver, rather than on routers located in the middle of the path, and uses a combination of link load and packet loss rates as attack indicators. If an attack is detected, the receiver sends a feedback packet to the sender to suppress abnormal traffic. However, there is no guarantee that the feedback packet will reach the attacker router promptly.

Liu et al. suggested filter-based denial-of-service defense systems [18]. If packet flooding is detected in a destination host, under a hypothetically ideal attack detection system, it requests permission to apply an Access Control List (ACL) to access the routers that are located at both the source and destination. Then, the requested access-router starts to apply the ACL to the specific IPs that are causing malicious activity to block the flow of traffic. However, this method is not able to remove the root of the problem. If attack commands keep changing, eventually all the IPs will be blocked, effectively cutting off the network from the Internet.

Sung and Xu suggested a traceback concept for defending against DDoS attacks based on the signature filtering method [36]. The traceback mechanism is used to identify the source of attackers according to the location information and several control messages. In their study, the authors develop the existing IP traceback technique further by adding more crucial information as 'intelligence.' By using smart filtering, DDoS traffic can be dropped by each router that is on the edge of the attack paths.

2.2 Application based Defense Mechanisms

Contrary to the network-based defense mechanism, the variety of fields in which the application-based approach can be deployed is huge. While the network-based approach can generally be applied in the battlefield, the application-based approach should be used specifically for defending against rapidly changing attacks.

Suriadi et al. offered an additional suggestion for defending web services using client puzzles [37]. The authors verified the effectiveness of integrating client puzzles in the existing web service platforms. Usually, the client puzzle method is used for distinguishing normal human beings from automated programs by asking tricky questions in order to block automated requests, such as auto logins and auto registers. The authors reported that puzzle questions were very effective for conserving the server resources in that they discard malicious traffic early in the process. With proper blocking, a puzzle-enabled web service can maintain a good performance level for the normal client even if it is under attack [37].

The Internet Relay Chat (IRC) based botnet has been studied and IRC-related botnets continue to appear. A well-known IRC-based detection approach is to sniff the traffic on common IRC ports and check the payloads [23].

Binkley et al. [27] proposed an algorithm for anomaly-based detection to mitigate the DDoS attack. The authors suggested an algorithm that examines a large number of TCP packets heading toward IRC hosts and computes the ratio of the total amount of the control packets to the total number of TCP packets. A high proportion of IRC-related packets is assumed to represent a potential attack [27].

C&C-structured botnets can be disarmed if their domain names are interpreted. Choi et al. [25] developed an algorithm for identifying botnet DNS queries [25]. Scrutinizing the DNS traffic and shutting down the suspicious URLs are the most effective ways to mitigate DDoS attacks in the early stages of a C&C botnet. However, the limitations of these methods became clear as the botnet evolved. Bot masters can avoid this detection by frequently changing their DNS names and fake URLs.

Walfish et al. proposed the 'DDoS Defense by Offense' concept for suppressing the attack traffic [24]. This approach to the defense mechanism is contrary to general concepts: slow down the bad clients. It is unique in that encourages the good clients to increase their traffic on the upload bandwidth to the target site in order to suppress the bad traffic. The idea originated from several assumptions, one of which is that, while not much upload bandwidth is available to attackers, legitimate users have sufficient bandwidth when an attack is approaching. It is obvious that if the attackers are depleting their upload bandwidth, then encouragement will not change their traffic volume. Inversely, normal users generally have much more bandwidth available for sending requests. Consequently, they will react to encouragement from the victims by increasing their traffic volume. As the normal requests increase, the enormous volumes of good traffic finally suppress the bad traffic coming from the attackers [24].

Jeong et al. proposed the detection and reputation mechanism for possible zombie PCs by analyzing spam mail traffic [38]. This idea is based on the statistics that most of Internet spam mails are coming from zombie PCs. This proposed system has applied to South Korea's spam-filtering system managed by KISA (Korea Information and Security Agency).

2.3 Reverse Engineering and Honeypots

In general, a honeypot system is used to capture and explore unknown malware. Reverse engineering techniques are used by anti-malware engineers to analyze the captured malware in order to find a countermeasure against the malware.

Honeypots are divided mainly into two different categories: low-interaction and high-interaction. As the name indicates, the low-interaction honeypot attracts limited types of attacker, usually through simple simulations of network services or operation systems [30]. This makes the honeypot easily detectable by attackers by using a combination of two or three multi-purpose messages. Thus, a low-interaction honeypot often lures only automated attacks [30]. An example of this type of honeypot is honeyd [32]. A high-interaction honeypot, on the other hand, uses real systems to interact with attackers; hence, the level of interaction is not limited [30]. However, the deployment of a high-interaction honeypot involves more risks because an attacker can gain complete control of the honeypot and abuse it [30]. Levine et al. used this honeypot method to collect and analyze rootkits manually [31].

However, recent research [20] has revealed that there is a novel way to deal with malware. Honeypot systems and reverse engineering techniques can also be used to discourage bot makers and botnet service sellers. Ormerod et al. demonstrated how honeypots and reverse engineering techniques can be used as discrediting tools in defense against botnets, not only for analysis purposes. From the economic viewpoint, there always exists a supply and demand of bot and botnet services, enticing bot suppliers and botnet buyers. Thus, according to the authors, the reason for the rapid evolution of bots and botnets is the financial profits that all bot authors and botnet makers are eager to make. Many countermeasures related to botnet problems have been suggested but they are not usually ultimate solutions, because they do not focus on botnet masters or authors. Ormerod et al. proposed a bottom-up approach to deal with this food chain system by discrediting botnet toolkits [20]. By defaming the bot suppliers and prosecuting the end-users of tractable stolen IDs that reverse engineering and honeypot techniques discovered, their harmful activities were quite effectively mitigated. In particular, the authors

suggested a method of defaming Zeus botnet toolkits using honeypots and reverse engineering. Finally, they succeeded in submitting false information through Zeus botnet services and prosecuting criminals who wanted to use stolen credit card numbers purchased from sellers.

A honeypot-based proactive defense is more effective and efficient for blocking DDoS attacks [29]. Moon et al. introduced the hybrid honeypot system, which enables the ISP to find the C&C structured botnet before it initiates an attack. According to the paper, in most botnet ecosystems, there is always a propagation period during which the bot master recruits more bots in order to stage an enormous attack. Starting from this timeframe, the suggested honeypot system collects the suspicious sample binary files of malware, utilizing the sub channels of the collectors installed in the ISP network. Then, it inputs these sample files to the virtual machine to record their behavior for dynamic analysis. While an intentionally infected virtual machine is trying to communicate with its C&C servers, the dynamic analyzer finds the botnet structural information, such as C&C IP addresses, malware distributor's URLs, and so on. Exploiting these crucial data, the ISP can block the server IP addresses or URLs by disabling the routing from their network infrastructure. After a 12-month experimental period, over 40% of malicious server IPs and URLs were identified and blocked before their bot masters commanded attacks [29].

Security companies have been playing a major role in avoiding DDoS attacks. It is true that if all users would just install anti-virus applications, the attacks would be significantly reduced, and many security applications exist that are able to detect and cure the malware and bots causing the attack. However, many users still do not install anti-virus applications, and therefore security companies focus on finding the signature of malware rather than seeking a C&C server. Botnet activities can be stopped only by eliminating C&C servers.

3 Legacy DDoS defense systems of ISPs

3.1 General Architecture

Generally, all security issues of ISPs are handled by a security team. ISPs have their own security teams to solve security problems in their network. They employ a network traffic monitor, which detects attacks based on sample traffic analysis. There are several ways to detect DDoS attacks [13] [14] [15]. One way is to measure the total traffic or the specific protocol traffic. This method relies on statistical data and human decisions. If the traffic of a specific protocol is greater than normal, it indicates to an operator who is monitoring initial network intrusion problems that there is malicious traffic.

C&C type botnets were mentioned in Section 2. Botnet masters have used IP addresses or IRC servers as C&C servers in the past. Recently, however, they started to use Dynamic DNS, which is able to change the IP address of DNS addresses dynamically to prevent the C&C addresses being blocked [10]. A system that blocks DNS addresses as well as IP addresses is therefore also needed to defend against botnets. ISPs also have a system that is able to block such Dynamic DNS C&C servers: sinkhole DNS. The principle of a sinkhole DNS is outlined below. It is located upstream of the DNS server, and checks all the packets going through it.

```
If requested DNS is in malicious DNS database
    Return 127.0.0.1 (Local loopback)
Else
    Return matched IP address
```

3.2 Standard Procedure

General ISPs respond to DDoS attacks based on their own attack response procedures.

The members of a security team can be classified as operators and analysts. First, an operator monitors network traffic and blocks or redirects malicious traffic to protect the network infrastructure, such as backbone routers and Internet Exchange routers [17] [19]. After this operation, an analyst examines the malicious traffic and extracts source IPs by means of a cross-check system that has the correlation data of the IP address and its subscriber information. The analyst then contacts subscribers whose system might be infected with malicious code. With the subscriber's consent, the analyst will examine the subscriber's PC using remote access systems. Accessing a subscriber's PC to analyze malware does not seem to be a very common practice. It is more common simply to block the source IPs that cause DDoS attacks. However, due to the competitive market environment, especially in Asia, ISPs cannot block subscriber IPs without permission. For security reasons, subscribers in Asia usually do not agree to an ISP controlling their Internet service.

Generally, an analyst finds a C&C server using Microsoft Windows system software and packet sniffing tools [4]. The analyst examines the virtual memory space looking for malware processes that will reveal the C&C server string, which would be something like xxx.9966.org. The manager then checks the side effects, such as typical services being unavailable when a DNS address is blocked.

Finally, the C&C server is determined based on analysis. The C&C server's address will then be put into the sinkhole DNS database. Then, every connection trying to contact this server will receive the loopback address 127.0.0.1 instead of the real IP address of the C&C server [12].

3.3 Discussion

One of the advantages of legacy systems and standard procedures using sinkhole routers is that they can protect the backbone of the network infrastructure securely and quickly. Massive malicious traffic can destroy network routers located on the path to the target as well as the original target itself. The domino effect of cascading DDoS attacks can damage intermediate network systems and cause countless lost connections between routers. From this point of view, the appropriate measures that sinkhole routers take to avoid these types of system failure should suffice.

Conversely, legacy systems have some drawbacks. First, legacy defense procedures are applicable and trustworthy only if the botnet is a C&C type. ISPs still cannot take action against P2P-based botnets because no effective method for dealing with them yet exists. Second, there is a high possibility that the analyst will make a mistake, thus blocking legitimate IP or DNS addresses. This could generate complaints from subscribers and content providers due to network service denials. Third, if vulnerabilities reside on a defense system that creates very serious problems, then the system can be compromised and abused by hackers. A perfect system never really exists in the security industry. For example, if bots use other DNS services such as Google DNS (8.8.8.8), the sinkhole DNS is useless, as the DNS traffic never passes through it. Moreover, botnet masters often deceive analysts by implanting fake C&C server addresses; therefore there is always the possibility that the analyst cannot discover the real C&C server address.

4 A Case Study

4.1 The DDoS attack

In April 2010, a security team detected an incoming 40-Gbps DDoS attack, and responded using its standard procedures. Having found and blocked the C&C server's DNS address, they believed that the DDoS attack had been blocked. A few months later, another DDoS attack occurred that was twice the size of the previous one, with incoming traffic reaching up to 80 Gbps. The security team panicked, because the DNS address they had blocked was again allocated on the subscriber PCs' IP address range. They realized that their legacy defense system was completely ineffectual. IP address-based access control could be a solution in this circumstance, and it is being used in many ISPs. However, it causes another problem from the viewpoint of customer satisfaction, in that the blocking operation can lock down a legitimate users' IP address range. In August 2010, there was a 200-Gbps DDoS attack, and the main backbone of the network was impaired. The damage was very severe at the Internet Exchange link. The company gradually lost control of their traffic and began to be unable to provide stable Internet services to their clients, because the incoming malicious traffic consumed all the available resources.

The security team realized that they needed additional techniques for defense against a DDoS attack from this botnet. In order to perform a dynamic analysis, the security team retrieved some copies of the malware from subscriber PCs. Unfortunately, the malware did not work outside the original PC, because it needed a specific Windows registry configuration to run, and it was a DLL file working through the service svchost.exe. This file was executed only when an original file was executed to set up the specific Windows registry, before the main file was executed. The team therefore decided to secure the original file.

First, the team reviewed the installed programs on subscriber PCs. All the infected PCs had one specific P2P program in common. The team therefore assumed that the P2P was the starting point for the malware's distribution, and examined possibly malicious files. Many contacted subscribers had downloaded adult movie files, and therefore the team searched for adult movie files in a particular format. Finally, they found a matching one on a famous P2P site.

4.2 Analysis of the malware

The malware that had caused the previous DDoS attack was named Backdoor.Mulkerv by Symantec. First, it came into existence from a self-extracting archive file that included an adult movie, and it was distributed via a P2P site. The name of the file was interesting, "hot 18 year old girl.exe" Wondrecek et al. shows that adult movies are an effective way to distribute malicious code. Only $150 is needed to make 20,000 bots [6]. The botnet master used this fact to make as many bots as possible, and was successful. The file size of the malicious file was over 100 MB. The size of the chunk of malicious code was much larger than the file size of other malicious code, as it included an actual movie file in avi format to gain the downloader's trust. Once the user downloaded the file and executed it, a self-extracting archive started to extract automatically, and a file called up.exe, which was included in the self-extracting archive, registered a background service to initiate malicious activity, and then deleted itself. It constituted a rootkit process that could only be found by using anti-rootkit software. It initiated a request for two DNS addresses, "xxx.9966.org" and "yyy.9966.org", when the install was complete, and created one TCP connection regardless of the results of the DNS queries. The infected user did not notice it, because the self-extraction looked successful, and the malicious process was hiding.

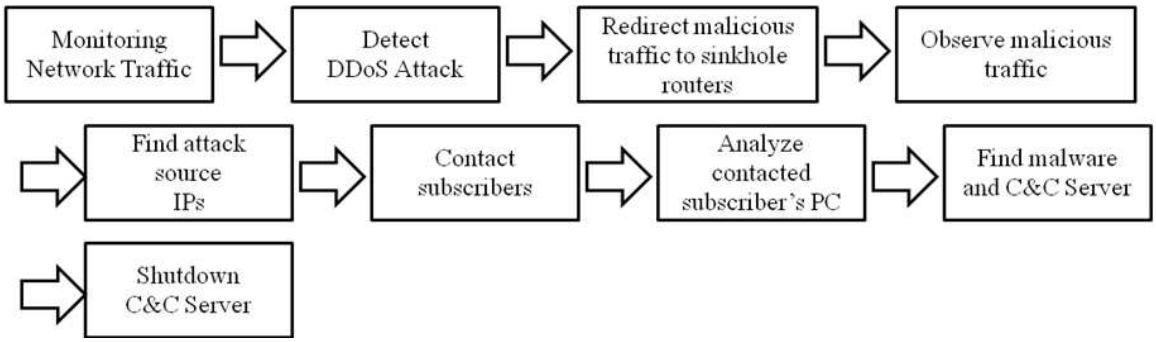

Fig. 1: Standard procedure for defending against a DDoS attack

Table 1: Security team's description of an analyzed botnet

	Backdoor.Mulkerv
File Name	inetpic#.dll (# is number)
Original File	up.exe (it was included at self-extracted file)
Species	NETBOT variety
Propagation	P2P site with fascinating file name
Main target	Servers in Internet Data Center
Anti-debug	Virtual Protect Ex, Inserting garbage codes
Other Information	Rootkit process Service process 2 DNS Queries Encrypted TCP communication After executing, dropper file is deleted. Changed Windows registry

4.3 The Mistakes

The first mistake made by the security team was that its members did not analyze the malware thoroughly. Since they needed only the C&C IP or DNS address according to their standard procedure, they ignored other aspects of the malware that could neutralize their defense system. Their second mistake was that they believed a legacy defense system would suffice to block any type of DDoS attack. Thus, they simultaneously overestimated their defense system and underestimated the threats of evolving botnets.

It was assumed that the botnet master had studied ISP defense systems for a long time. The botnet master found vulnerable points in standard ISP defense systems and neutralized or avoided each aspect of the defense system, one by one. Obviously, the first weak point the botnet master found was a sinkhole DNS, as he or she decided to use dynamic DNS for the C&C server. The weakness of the sinkhole DNS system has been mentioned in Section 3.3. The botnet master used open DNS to avoid a sinkhole DNS. Fig. 2 shows how the botnet master avoided a sinkhole DNS.

The botnet master could therefore change the IP of the C&C server dynamically without being blocked by the sinkhole DNS server. The security team discovered this fact by packet capturing on a subscriber PC. Therefore, they decided to track and block the IPs of the dynamic DNS in real time, but this was the third mistake. The botnet entered two DNS queries, xxxx.9966.org and yyyy.9966.org. These queries returned the IP address AAA.BBB.127.0 and AAA.BBB.253.192, respectively. However, the botnet only made one TCP connection at CCC.DDD.12.4. The security team concluded that one of the addresses, either AAA.BBB.127.0 or AAA.BBB.253.192, was directly related to the C&C server. They assumed that the botnet received the real address, CCC.DDD.12.4, from either AAA.BBB.127.0 or AAA.BBB.253.192, without any validations. Meanwhile, additional massive DDoS attacks occurred, and the security team realized that they were wrong again. More precise methods needed to be used, instead of only assumptions.

The security team tried to find the C&C server's IP through a network packet signature. They assumed that Backdoor.Mulkerv was a NETBOT variety of malware based on the file signature. They therefore thought that if the signature was given to a global Intrusion Detection System (IDS), they would find the C&C server's IP. Up to this point, all the C&C server IP addresses had belonged

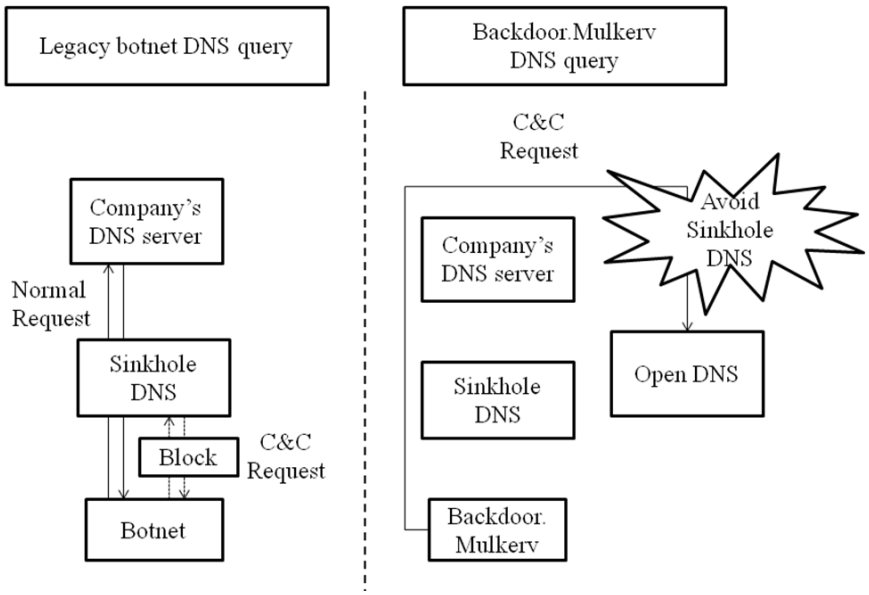

Fig. 2: How Backdoor.Mulkerv avoided the sinkhole DNS

to other countries, which was a strong argument for initiating this plan. They discovered that the packet signature in NETBOT was like a string, such as "AAABNz," which was used to communicate between the server and client.

After they finished setting up the Intrusion Detection System (IDS) in the global link area, the operators monitored events from the IDS. If they found matched signature packets, they blocked IPs based on the logs. However, relying heavily on IDS was also a problem. If the IDS was not efficient, the operators could miss a change in the C&C server.

4.4 The new approach

After some debate, the team decided to use a honeypot to examine the botnet in greater detail. They noticed that the main problem was that they could not find the network activities when the malicious code was installed on the subscriber PC. They installed Microsoft Windows XP Professional with Service Pack 2, a packet sniffer tool, and an anti-rootkit tool. In addition, they installed Microsoft TCP-View to examine C&C servers that were changing frequently. They ran the dropper file up.exe as the last step. They recorded all the network activities for one week, a procedure that had not been done previously. They focused on network activities rather than other activities, such as file activity, because their main interest was tracking and blocking C&C servers and discovering how the C&C servers ordered attack commands. After examining network packets, they reached the following conclusions.

(A) The IP addresses yielded by DNS queries for xxxx.9966.org and yyyy.9966.org were not directly related to the C&C server address.
(B) However, the relationship between the DNS query and the C&C address was indirect, based on the fact that the IPs had changed immediately before the attack was initiated. They assumed that some calculations had been performed to build a real C&C server's IP address.
(C) Attack commands were not encrypted. They were written only in plain text, such as 10.10.10.10, which was intended to create an attack toward 10.10.10.10, with TCP, UDP, and ICMP flooding, while live checking that packets were encrypted. The targets that the team found were very limited, but it seemed that attacking other targets was possible, because attack commands were in just plain text.
(D) Even if the C&C server was eliminated, the attack continued for at least 30 min. It was important to block the C&C server immediately before attack commands were given by the bot. Thus, a fast response was most important in this case.
(E) The DNS query was not performed at one specific open DNS server. It used eight different open DNS servers to obtain an exact IP address. If the result of the DNS query was 127.0.0.1, it ignored the result and queried another open DNS server. It is assumed that the botnet master had maximized the robustness of each bot. Its robustness was eight in terms of DNS addresses [31]. Since it was impossible to block eight open DNS servers, the problem in this case was more serious than in others.

```
DWORD IP_ADDRESS_1 = result of the DNS query ”xxxx.9966.org”;
DWORD IP_ADDRESS_2 = result of the DNS query ”yyyy.9966.org”;
DWORD MASKS = 0x0000FFFF; //filled with ”1” in last 16 bits

IP_ADDRESS_1 = IP_ADDRESS_1 & MASKS; //drop first 16 bits
IP_ADDRESS_2 = IP_ADDRESS_2 & MASKS; //drop first 16 bits

DWORD NEW_IP_ADDRESS = IP_ADDRESS_1 << 16 | IP_ADDRESS_2;

for(int i=0;i<5;i++)
{
    NEW_IP_ADDRESS = RotateBitsRight(NEW_IP_ADDRESS, 0xc);

    //Rotate NEW_IP_ADDRESS toward right with 12 bits

    NEW_IP_ADDRESS = NEW_IP_ADDRESS ^ 0x708FF9CD ^ 0x4ADDFBA0;

    //XOR Calculation

}
DWORD Real_Command_And_Control_Server_IP = NEW_IP_ADDRESS;

Trying to TCP connect toward ”Real_Command_And_Control_Server_IP”
```

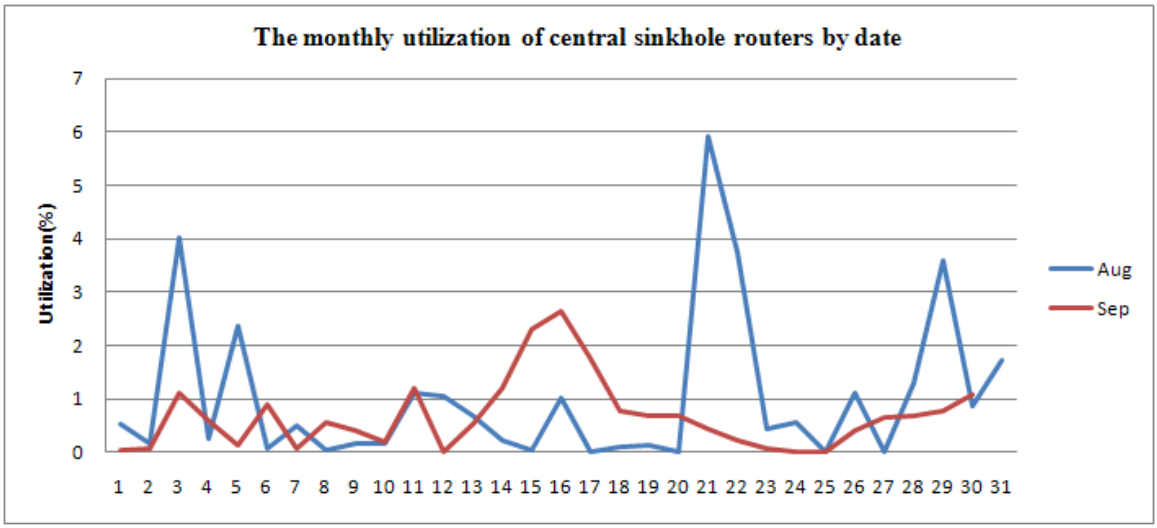

Fig. 3: The monthly utilization of central sinkhole routers by date

The honeypot was useful for checking C&C server's IP addresses, but it was also inconvenient, because the operators had to connect to the honeypot every minute in order to check whether or not the IP address had changed. Thus, it was not an automated process. Moreover, the time difference between changing the C&C server and giving attack commands was very small. The botnet continued to attack despite the fact that the C&C server based on (D) had been blocked. The security team therefore decided to analyze the malware using reverse engineering to understand the algorithm used to build a C&C address from the DNS query results. It was a time-consuming task, because a piece of anti-debugging technology called Virtual Protect Ex prevented the analyst from performing reverse engineering through various debugging check techniques, including specific register checks. After breaking the anti-debugging technology, they encountered another problem, which was an obstruction caused by the insertion of garbage code. The virus creator used an anti-debugging program to insert

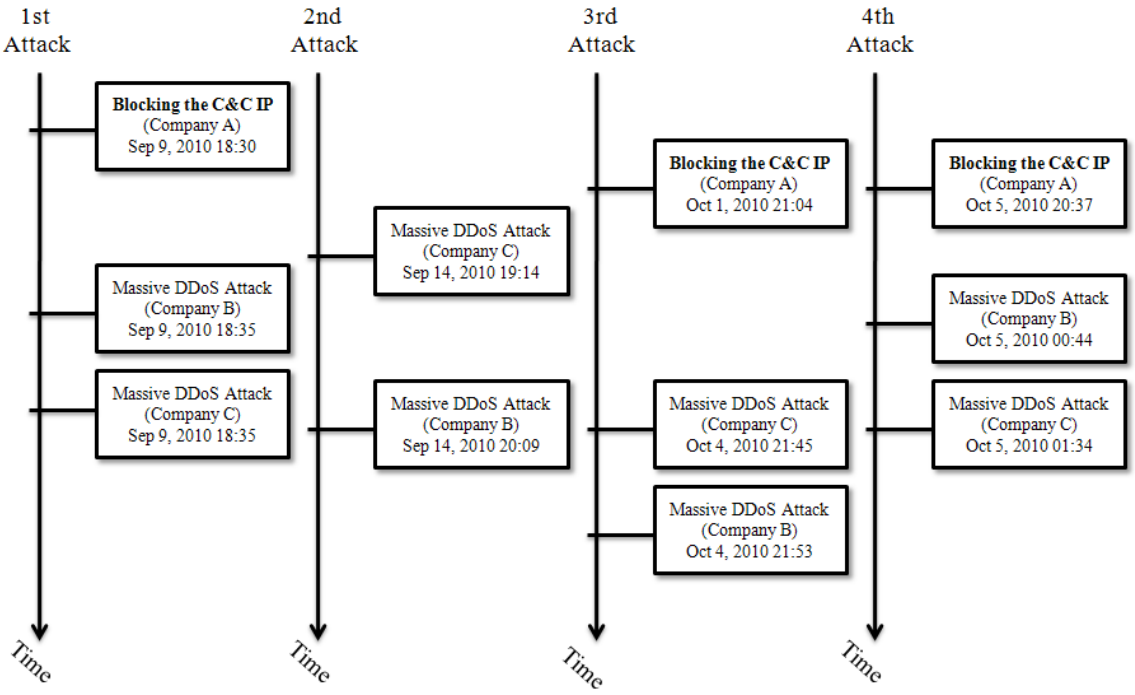

Fig. 4: Records of massive DDoS attacks that occurred at ISPs A, B and C

irrelevant execution code to block reverse engineering. This kind of program has been widely adopted by the creators of malware, including spyware, which makes reverse engineering more difficult for virus analysts. The analyst discovered over 1,000 meaningless machine instructions to execute programs, and the analyst eliminated these by checking each one. This took more time than originally expected. Finally, they found the C&C IP address-building algorithm. The pseudo code looked as shown below.

It is assumed that the botnet master tried to hide the C&C server's IP by encoding the original IP address. It seems that the botnet master had analyzed standard ISP defense systems at least as much as the ISPs had analyzed botnet network activities. The botnet master knew the structure of ISP defense systems and anti-virus systems thoroughly, and chose the best way to neutralize the defense system. In the end, the security team built a new monitoring and control system for Backdoor.Mulkerv based on analytic data, and they finally started to block the C&C server IP correctly and promptly.

4.5 A Consequence

The result of the efforts to defend Backdoor.Mulkerv was very significant. There were over 10 DDoS attacks at a volume of over 100 Gbps in August 2010 alone. In September, however, there was only one attack. Fig. 3 illustrates the method's effectiveness in detail. The figure presents empirical data according to operational activities in the real world. Sinkhole routers are used to redirect malicious traffic. Each line in Fig. 3 represents a relationship that is proportional to the total volume of malicious traffic in a month. A variable, "Utilization", is the combination of the duration of DDoS attacks and their total volume.

It can easily be seen that the peaks in September are lower than those in August in general, except on a few days, and the top utilization in September was 2.64 (%) while in August it was 5.93 (%).Thus, it can be concluded that the average volume of malicious traffic was reduced in September as compared with August as the utilization of central sinkhole routers was reduced. This reduction was derived by a new monitoring and control systems that the security team built and implemented on September 8, 2010. They actually succeeded in blocking some C&C servers before attacks occurred.There are other data that offer detailed proof that the treatment was very effective from a different point of view. Fig. 4 shows the records of massive DDoS attacks that the three major ISPs in South Korea experienced.

Thus, once a DDoS attack has occurred at company A, the attack is not only company A's problem. The attack immediately affects the neighboring ISPs. In the case study above, ISPA introduced a bot-specific defense system using a honeypot and a unique solution, but the ISP of B and C did not have this system. The Fig. 4

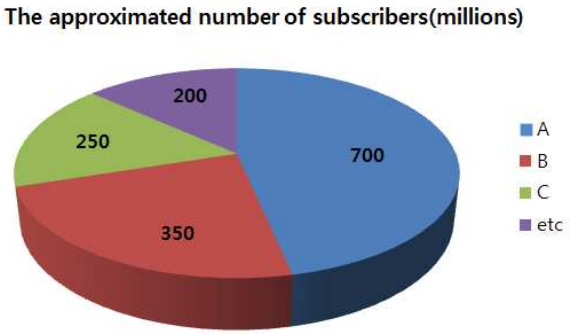

Fig. 5: The approximate number of subscribers

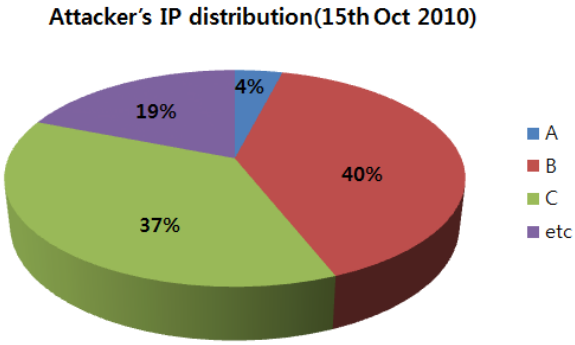

Fig. 6: The IP distribution of attackers

represents the timetable that depicts the proactive actions of the ISP A according to a new method and the DDoS attacks records of companies B and C. After ISP A blocked the C&C IP address, the DDoS attacks occurred in other companies 5 min or a few days later. No massive DDoS attacks were reported in the network of ISP A in this period, which proves these proactive actions were quite effective and efficient. It also shows that the proactive action of company A could diminish massive DDoS attacks, while the actions of companies B and C.

Performances can be estimated in several ways, such as by using the following pie graphs, Figs. 5 and 6.

Fig. 5 indicates the approximate number of subscribers of each ISP in South Korea and Fig. 6 shows how many subscribers to the ISPs were involved in DDoS attacks. This information is based on the routing logs from sinkhole routers on 15th October, 2010. It was discovered that the C&C server was a combination of two unique DNS queries. It was a massive attack also, but not as large as previous attacks. ISP A, with the largest number of subscribers, could find only a few belonging to it that were affected by this attack because of the proactive measures taken by their security team before the attack actually happened. Only 4% of attackers are from the ISP A.

The probability of being infected with a botnet is equally distributed to every user, which means that every subscriber at any time could be a bot on cyberspace, regardless of any particular ISP. However, ISP A showed the lowest portion of attackers, despite having the largest amount of subscribers. This fact shows that that the proactive measures taken by the ISP A were obviously quite effective and efficient. It also shows that honeypots, reverse engineering, and sinkhole DNS are desirable DDoS attack defense techniques for network service providers, always managing to overcome them.

5 Conclusion

The real case described above shows how an ISP can defend itself against recent DDoS attacks.

5.1 Respond proactively

As mentioned in Section 2.2, anyone could be a victim of a botnet coordinator, and DDoS attacks are a global problem. When the security team examined source IP addresses generating DDoS attacks, they found a wide range of IP addresses that did not belong to specific areas.

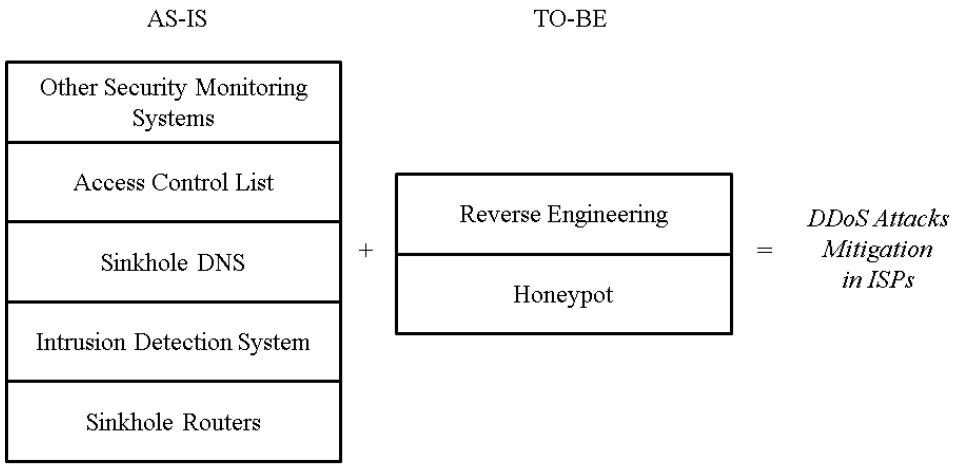

Fig. 7: Advanced DDoS mitigation framework for ISP

As discussed earlier, the botnet master uses social engineering techniques, such as exploiting adult movies, to make as many bots as possible.

In particular, ISPs should bear in mind the high probability that no anti-virus program has been installed on their subscribers' computers, since unfortunately, many subscribers are not aware of the necessity of such programs. Therefore, it is highly recommended that ISPs should not simply wait until infected subscribers debug their computers, but rather that they find and block botnet C&C servers.

5.2 Approach differently

The conventional belief is that honeypots and reverse engineering are some of the techniques intended only for use by anti-virus companies such as Kaspersky and Symantec. However, even ISPs need to analyze malware to block DDoS attacks effectively. Even though the targets of DDoS attacks are not the ISP network infrastructure, as the volume of the DDoS attack grows, an ISP's network infrastructure could incur collateral damage. The main reason for this damage is that the throughput of all intermediate network systems is limited. As a result, a massive DDoS attack is able to destroy network infrastructure as well as its original target. Failure to find the exact C&C servers is critical for ISPs. Hence, in a real battlefield, the key to solving the problem in this particular case will be honeypots and reverse engineering. It is important for ISPs to acquire at least basic analysis techniques. As mentioned in Section 2.4, honeypots and reverse engineering techniques could be used not only to analyze malware, but also to mitigate botnet activities. Prompt removal of a C&C server guarantees the stability of the network.

5.3 Apply uniquely

It is true that ISPs cannot handle high-level security threats above the network layer. Unlike an anti-virus company, they can usually block only IPs and DNS addresses that might generate malicious network activities. However, disconnecting the C&C server is also a very powerful measure. Only small modifications are needed to block evolving botnets. In the real case studies presented herein, the security team implemented only an additional monitoring and control system that was able to estimate the real C&C server, but the effect was very significant.

Building a unique defense system could also be an asset for controlling network quality. In this case study, blocking one botnet reduced the volume of garbage traffic. Basically, ISPs are companies that sell network bandwidth. High bandwidth availability indicates that they can provide higher-quality services.

5.4 Do it efficiently

Empirical data was used to prove that our suggestions are effective. Fig. 3 indicates the reduction in the total volume of malicious traffic, showing a comparison of the time before and after the adoption of the new defense system. Fig. 4 shows a massive DDoS attack that was not reported in A company taking proactive measures, while other companies that did not take preemptive measures were still being attacked. Figs. 5 and 6, which give the approximate number of subscribers and the attacker's IP distribution in terms of ISPs, respectively, show clearly the positive results of proactive measures. It is commonly thought that a large ISP needs to make a huge monetary investment to handle security problems in depth, but this may not be true. The security team used only a honeypot

PC and a cheap server for monitoring and control. A little additional effort makes a big difference in terms of network quality.

Fig. 7 shows an advanced DDoS mitigation framework for an ISP. ISPs use many expensive tools to block DDoS attacks, such as sinkhole routers and intrusion detection systems. These systems are effective for defending their infrastructure, but botnet masters are always trying to conceive new ways to avoid them, and, as time goes by, many will succeed. We suggest two new factors to compensate AS-IS systems that will result in mitigation of DDoS attacks on ISPs. We strongly believe our method will be effective and efficient in other ISPs as well.

Acknowledgement

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract.

References

[1] N. Provos and T. Holz, Virtual honeypots - from botnet tracking to intrusion detection, Addison Wesley.
[2] M. Feily and A. Shahrestani, A survey of botnet and botnet detection, Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference, 268-273 (2009).
[3] J. Leonard, S. Xu and R. Sandhu, A framework for understanding botnets, Availability, Reliability and Security, 2009. ARES '09. International Conference, 917 (2009).
[4] K. S. Han and E. Im, A study on the analysis of Netbot and design of detection framework, 2009 Joint Workshop on Information Security, 1-12 (2009).
[5] S. L. Pfleeger, Anatomy of an intrusion, IT Professional, **12**, 20-28 (2010).
[6] G. Wondracek, T. Holz, C. Platzer, E. Kirda, and C. Kruegel, Is the Internet for porn? An insight into the online adult industry, in Proc. of the 9th Workshop on Economics of Information Security, (2010).
[7] J. Mirkovic and P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, in SIGCOMM Computer Communications, 39-53 (2004).
[8] C. Douligeris and A. Mitrokotsa, DDoS attacks and defense mechanisms : Classification and state-of-the-art, in Computer Networks, 643-666 (2004).
[9] A. Asosheh, and N. Ramezani, A comprehensive taxonomy of DDOS attacks and defense mechanism applying in a smart classification, in Trans. on Computers, 281-290 (2008).
[10] R. Villamarín-Salomón and J. C. Brustoloni, Identifying botnets using anomaly detection techniques applied to DNS traffic, in Consumer Communications and Networking Conference, 476-481 (2008).
[11] M. Sachdeva, G. Singh, K. Kumar, and K. Singh, A comprehensive survey of distributed defense techniques against DDoS attacks, in International Journal of Computer Science and Network Security, **9**, 7-15 (2009).
[12] F. Freiling , T. Holz and G. Wicherski , Botnet tracking : exploring a root-cause methodology to prevent distributed denial-of-service attacks, in Proc. of the 10th European Symposium on Research in Computer Security (ESORICS '05), **3679**, 319335 (2005).
[13] T. Xu, D. K. He and Y. Zheng, Detecting DDoS attack based on one-way connection density, in 10th IEEE Singapore International Conference on Communication Systems, (2006).
[14] Y. You, M. Zulkernine and A. Haque, Detecting flooding-Based DDoS attacks, in ICC '07 IEEE International Conference on Communications, 1229-1234 (2007).
[15] J. FENG and Y. LIU, The research of DDoS Attack detecting algorithm based on the feature of the traffic, 5th International Conference on Wireless Communications, Networking and Mobile Computing, 1-4 (2009).
[16] X. Liu, X. Yang and Y. Xia, NetFence : Preventing Internet denial of service from inside out, in Proc. of the ACM SIGCOMM 2010 Conference on SIGCOMM, 255-266 (2010).
[17] D. Rajnovic, Black hole routers, TF-CSIRT Cisco.com, (2002).
[18] X. Liu, X. Yang and Y. Lu, To filter or to authorize : Network-layer DoS defense against multimillion-node botnets, in Proc. of the ACM SIGCOMM 2008 Conference on Data Communication, 195-206 (2008).
[19] Y. Afek, R. brooks, N. Fischbach, MPLS-based traffic shunt, NANOG28, (2003).
[20] T. Ormerod, L.Wang, M. Debbabi, A. Youssef, H. Binsalleeh, A. Boukhtouta, and P. Sinha, Defaming botnet toolkits: A bottom-up approach to mitigating the threat, in 2010 Fourth International Conference on Emerging Security Information, Systems and Technologies, 195-200 (2010).
[21] Y. C. Chen, K.H Hwang, Collaborative change detection of DDoS attacks on community and ISP networks, in Transactions of the IRE Professional Group, 401-410 (2006).
[22] A. Keromytis, V. Misra, and D. Rubenstein, SOS: An architecture for mitigating DDoS attacks, in the IEEE Journal on Selected Areas in Communication, **22**, 176-188 (2004).
[23] S. Racine, Analysis of Internet Relay Chat usage by DDoS zombies, In Master's thesis in the Swiss Federal Institute of Technology Zurich, (2004).
[24] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, DDoS defense by offense, in ACM Transactions on Computer Systems (TOCS), **28**, (2010).
[25] H. Choi, H. Lee, H. Lee, and H. Kim, Botnet detection by monitoring group activities in DNS traffic, in 7th IEEE International Conference on Computer and Information Technology, 715-720 (2007).
[26] D. Champagne and R. B. Lee, Scope of DDoS countermeasures : taxonomy of proposed solutions and design goals for real-world deployment, Princeton Univ. Tech. Report CE-L2005-007 (2005).
[27] J.R. Binkley and S. Singh, An algorithm for anomaly-based botnet detection, in Proc. of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), 43-48 (2006).
[28] M. Sachdeva, G. Singh, K. Kumar, and K. Singh, A comprehensive survey of distributed defense techniques against DDoS attacks, in International Journal of Computer Science and Network Security, **9**, 7-15 (2009).

[29] Y. H. Moon, E. Kim, S. M. Hur, and H. K. Kim, Detection of botnets before activation : an enhanced honeypot system for intentional infection and behavioral observation of malware, in Security Communication Networks, **5**, 1094-1101 (2012).
[30] J. Zhuge, T. Holz, X. Han, C. Song, and W. Zou, Collecting autonomous spreading malware using high-interaction honeypots, in Proc. of the 9th International Conference on Information and Communications Security, (2007).
[31] J. Levine, J. Grizzard, and H. Owen, Application of a methodology to characterize rootkits retrieved from honeynets, in Proc. of the Fifth Annual IEEE SMC, 15-21 (2004).
[32] N. Provos, A virtual honeypot framework, in Proc. of the 13th USENIX Security Symposium, (2004).
[33] J. Ioannidis and S.M. Bellovin, Implementing pushback: Router- based defense against DDoS attacks, in Proc. Network Distributed System Security Symposium, 79-86 (2004).
[34] Y. H. Hu, et al., Packet filtering for congestion control under DoS attacks, in Proc. of the 2nd IEEE International Information Assurance Workshop, 3-18 (2004).
[35] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson, A framework for a collaborative DDoS defense, in Proc. of Computer Security Applications Conference, 33-42 (2006).
[36] M. Sung, J. Xu, IP traceback-based intelligent packet filtering : a novel technique for defending against Internet DDoS attacks, in IEEE Trans on Parallel and Distributed Systems, **14**, 861-872 (2003).
[37] S. Suriadi, D. Stebila, A. Clark, L. Hua, Defending web services against Denial of Service attacks using client puzzles, in IEEE International Conference on Web Services, 25-32 (2011).
[38] H. Jeong, H.K. Kim, S. Lee, and E. Kim, Detection of Zombie PCs Based on Email Spam Analysis, in KSII Transactions on Internet & Information Systems, **6**, 1267-1478 (2012).

Young Hoon Moon received his B.S. in Telecommunication and Information Engineering from Korea Aerospace University in 1997 and his M.S. in Electrical Engineering from University of Southern California in 2004. He was an Information Security Officer in Korean Air Force for three years and he is now a network security engineer in the cyber security center for Korea Telecom.

Suk Bong Choi received the Bachelor's degree in science with great honor from Korea University in 2009. He was a member of the distributed computing labs. He is working for Korea Telecom since 2010. He is taking charge in IT Governance coordinating IT projects in terms of their architecture. His research area is in security, cloud, and platform.

Huy Kang Kim received his B.S. in Industrial Management, M.S. and Ph.D. in Industrial Engineering from KAIST (Korea Advanced Institute of Science and Technology) in 1998, 2000 and 2009, respectively. He was a technical director for security division in NCSoft and he is now an assistant professor in Graduate school of Information Security, Korea University.

Changsok Yoo received the Master's degree in Engineering from Seoul National University in 2001, and the Ph.D. degree in Energy Economics from Seoul National University in 2011. He was a chief data analyst in GameHi, and he is currently an assistant professor in Department of Cultural Tourism Content, Kyung Hee University. His current research interests are in the area of online game economics, industry analysis, and new business models.