World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:7, No:12, 2013

Study on Network-Based Technology for Detecting Potentially Malicious Websites

Byung-Ik Kim, Hong-Koo Kang, Tae-Jin Lee, Hae-Ryong Park

Open Science Index, Computer and Systems Engineering Vol:7, No:12, 2013 publications.waset.org/9996682/pdf

***Abstract*—**Cyber terrors against specific enterprises or countries have been increasing recently. Such attacks against specific targets are called advanced persistent threat (APT), and they are giving rise to serious social problems. The malicious behaviors of APT attacks mostly affect websites and penetrate enterprise networks to perform malevolent acts. Although many enterprises invest heavily in security to defend against such APT threats, they recognize the APT attacks only after the latter are already in action. This paper discusses the characteristics of APT attacks at each step as well as the strengths and weaknesses of existing malicious code detection technologies to check their suitability for detecting APT attacks. It then proposes a network-based malicious behavior detection algorithm to protect the enterprise or national networks.

***Keywords*—**Advanced Persistent Threat, Malware, Network Security, Network Packet, Exploit Kits.

I. Introduction

CYBER-ATTACKS have been more prominent recently. In the past, hackers created and distributed malicious code for malicious purposes or to show off their capabilities. Note, however, that they have now gone beyond mere curiosity or showing off, evolving into a form of cyber terror inflicting economic or physical damage on a country or a specific organization. Unlike the past cyber-attacks, today's attacks move intelligently, persistently, and secretly against a specific target. Such attacks are called APT (Advanced Persistent Threat) [2]-[4]. APT attacks are not easily detected with the current security technologies. Although security technologies can easily detect known cyber-attacks, they cannot detect attacks such as APT, which is customized for the attack target. Moreover, these attacks bypass the existing security solutions (antivirus, firewall, etc.) to make them even more difficult to detect. These APT attacks target specific enterprises or countries and occur more and more frequently [1].

Byung-Ik Kim is with the Korea Internet & Security Agency, IT Venture Tower, Jungdaero 135, Songpa, Seoul, Korea (phone: +82-2-405-5253; fax: +82-2-405-5249; e-mail: kbi1983@kisa.or.kr).

Hong-Koo Kang is with the Korea Internet & Security Agency, IT Venture Tower, Jungdaero 135, Songpa, Seoul, Korea (phone: +82-2-405-5346; fax: +82-2-405-5249; e-mail: redball@kisa.or.kr).

Tae-Jin Leeis with the Korea Internet & Security Agency, IT Venture Tower, Jungdaero 135, Songpa, Seoul, Korea (phone: +82-2-405-4828; fax: +82-2-405-5249; e-mail: tjlee@kisa.or.kr).

Hae-Ryong Park is with the Korea Internet & Security Agency, IT Venture Tower, Jungdaero 135, Songpa, Seoul, Korea (phone: +82-2-405-5245; fax: +82-2-405-5249; e-mail: hrpark@kisa.or.kr).

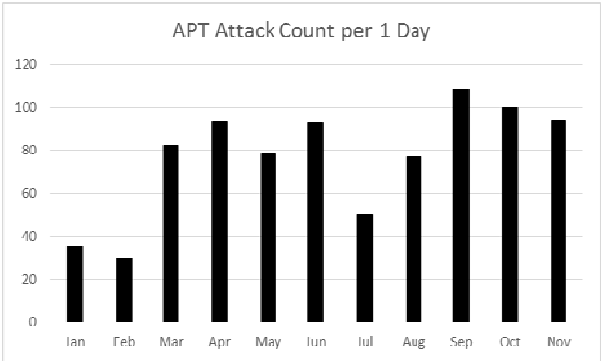


Fig. 1 Average APT Attack Count per day (Symantec, 2011)

An APT attack occurs in steps such as "Preparation and Penetration," "DataGathering," "DataLeakor System Destruction," etc [1].

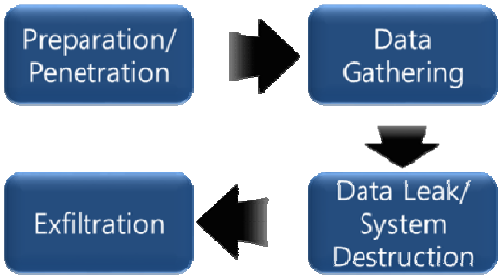


Fig. 2 APT Attack Lifecycle

The first step, "Preparation/Penetration," involves investigating the vulnerabilities to penetrate the target network. It acquires the internal user data of the target enterprise mostly through SNS (Social Networking Service) search. It then sends false e-mail or web link related to the user's interest or hobby using the collected information as well as the URL data of the website distributing the malicious code [1], [4], and [7]. When the user clicks the link, it will create the path through which the attack can penetrate the enterprise.

Once the attack penetrates an enterprise, it executes the "Data Gathering" step using various malicious codes. It checks the enterprise network structure, main server data, system administrator data, etc., and performs malicious behaviors such as acquiring confidential information or financial information of the enterprise over a long period [1].

In the "Data Leak or System Destruction" step, the attack paralyzes the target enterprise network or destroys the main system servers after receiving the confidential data collected in the previous step.

Open Science Index, Computer and Systems Engineering Vol:7, No:12, 2013 publications.waset.org/9996682/pdf

Since the current security systems are designed to detect or block known attacks, they are vulnerable to attacks with unknown attack pattern, like APT; hence the increasing need for technologies to detect unknown attack patterns or APT attacks.

This paper proposes a new algorithm that supplements the existing security systems to detect APT attacks quickly.

The rest of this paper is organized as follows: Chapter I describes the background; Chapter II presents the existing cyber-attack detection methods; Chapter III introduces the proposed key technology; finally, the last chapter suggests the future direction of the study.

II. Related Studies

Most hackers spend the longest time on the "Preparation and Penetration" step of the APT attack lifecycle. This step includes malicious code generation, development of malicious code-distributing website, website hacking, and collection of target enterprise and employee data. The website developed to distribute the malicious code is called the malicious code-landing/distributing site.

This chapter introduces the technology for detecting malicious code-distributing sites as the starting points of APT attacks. It also compares the strengths and weaknesses of existing security technologies. The new algorithm to supplement their weaknesses is described in the next chapter.

A. Preparation of Malicious Code-Landing/Distributing Site

As mentioned above, a website vulnerable to distributing the malicious code is called the malicious code spread point. The website leading Internet users to the malicious code-distributing site is called the malicious code-landing site.

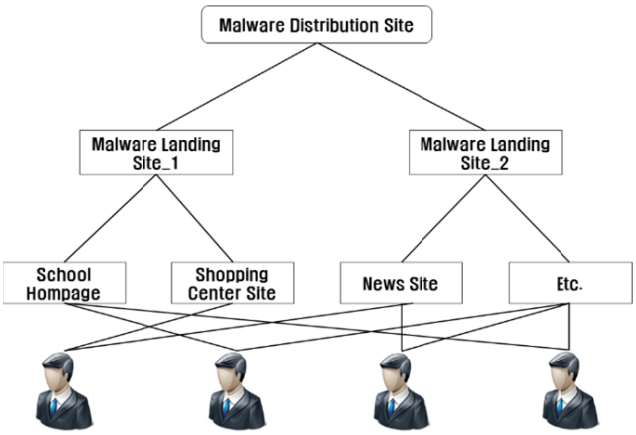


Fig. 3 Malware-Landing/Distributing Site

Hackers hack popular websites frequented by many users to link their malicious code-landing sites. Users innocently accessing a well-known website are led to the malicious code-landing site connected by the hacker. This malicious code-landing site then leads the user to the malicious code-distributing site so that the malicious code will be distributed [5].

As the use of the Internet rapidly increases, infection of malicious code through such websites is also on the rise. Therefore, most security systems are designed to detect such malicious code-distributing sites or malicious code penetrating inside from the network.

B. Automated Attack Preparation Tools

Hackers obtain the malicious code-landing/distributing sites and subsequently distribute and execute the malicious code to the users in specific ways. A typical way is to ask the users whether to download a file from a website to the user PC. Such method is not desirable to the party distributing the malicious code because users have the option not to receive the malicious code upon recognizing such.

"Drive-by Download" is an attack technique of downloading the malicious code to the user PC and executing it by bypassing such manual acceptance step [6], [8], [9], [13].

The "Drive-by Download" attack can be easily deployed using various automation tools. After securing the landing/distribution site, a hacker can easily create the attack code using the automation tool.

Generally called the "Exploit Kits," the automation tool has various functions [16], [17]. Most of the steps needed for the actual attack – such as creation of malicious code, creation of attack code to be inserted into the malicious code-distributing site, transmission of the malicious code-distributing site that is not detected by the security system, etc. -- can be created with a few mouse clicks.

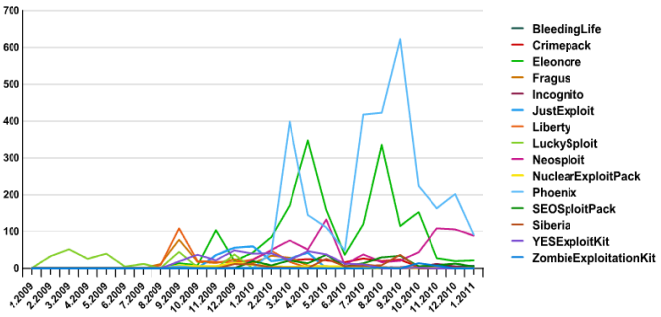


Fig. 4 Example of Exploit Kit (2011, Kaspersky Lab)

Using the exploit kit, a hacker can perform various malicious actions such as distribution of malicious code to the target enterprise, attacking the target after creating the zombie PC in the target and transmitting the information-gathering malicious code, etc.

C. Existing Malicious Website Detection Technologies

As mentioned above, exploit kits are often used in APT attacks and cyber terror snow a days. Black-hole exploit kit and Red kit are popular automation tools. There are many technologies for detecting the malicious code distributed by such automation tools.

HoneyPot is a leading technology of detecting websites distributing malicious code. A HoneyPot technology is mainly divided into the low-interaction clientHoneyPot and high-interaction clientHoneyPot [9]-[12].

The low-interaction client HoneyPot receives the list of

Open Science Index, Computer and Systems Engineering Vol:7, No:12, 2013 publications.waset.org/9996682/pdf

websites to be inspected and downloads their contents. It checks the distribution of malicious code by inspecting whether the downloaded data contain known malicious behavior signature. This method enables checking many websites for malicious code distribution quickly at a relatively low system development cost. As its weakness, however, it cannot detect anything if there is no known malicious behavior signature.

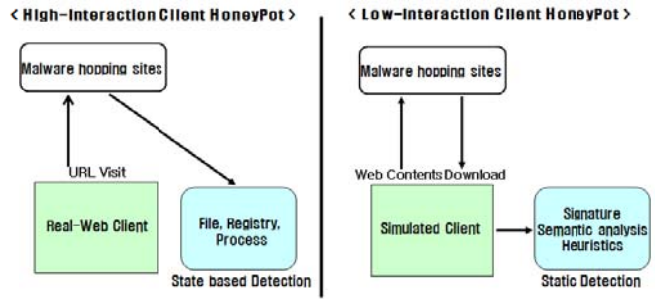


Fig. 5 Low-interaction vs. High-interaction Client HoneyPot

Developed to overcome such weakness, the high-interaction client HoneyPot technology uses virtualization to visit the sites to be inspected for malicious behaviors. Its strength is that it can detect even unknown malicious behaviors, which is the weakness of the low-interaction client HoneyPot. The most recent malicious behavior-detecting web services -- such as Wepawet [14] and MonkeyWrench [15]--use this technology to inspect malicious behaviors in the websites.

Note, however, that high-interaction client HoneyPot [10] has a weakness, i.e., taking too long to analyze each website, not being able to detect the malicious behavior of a website depending on the virtualization environment.

In addition, the IDS/IPS technology detects malicious behavior on the network layer. It identifies whether the known malicious behavior exists in the data transmitted into the enterprise network. It can detect malicious behavior in real-time, but not unknown attacks.

The next chapter introduces a new algorithm to supplement the weaknesses of the security technologies.

III. Proposed Technology

As mentioned in Chapter II, there are active ongoing studies on malicious code-distributing sites and malicious code detection technologies. Note, however, that they are still inferior in terms of detecting APT attacks with unknown patterns.

This chapter introduces an algorithm that supplements such weakness and detects the distribution site data of malicious code flowing into the enterprise network, detects the malicious code, and secures the inspection basis of unknown attacks.

A. Features of the Proposed Algorithm

For the proposed algorithm to be effective, the traffic data of the point where the enterprise network and outside network are connected must be secured. It can check the specific data of the network packet to detect unknown malicious behavior or known malicious behavior and malicious code.

This algorithm monitors all URLs transmitted into the enterprise through the network and identifies a series of URL sets generating the file transfer. Some of the selected URL sets (URL sets to be inspected) are formed into the visit inspection target URL set to check all selected URL sets.

Using the visit inspection target URL set, the steps in preparing against APT attacks -- such as checking of malicious code distribution in the enterprise, checking of malicious code-landing/distributing site, and checking of internal zombie PC -- can be detected in real-time.

B. Principle and Description of the Algorithm

The algorithm gathers the transmitted network traffic in real-time. It checks the packet header data of all network traffic in real-time and extracts TCP; in particular, the Web packets are extracted into the collection target. It then checks the header information of the extracted packet to extract the source IP/port and destination IP/port as well as the session data.

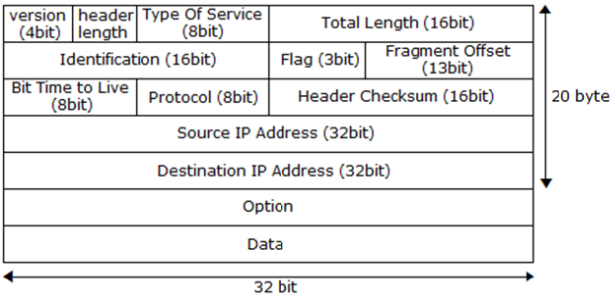


Fig. 6 TCP/IP Header

Using the collected session data, it continuously monitors the sessions of each IP/port. After a session is terminated, its IP/port communication data are restored using the session data. The URL data are extracted in the same sequence as the sessions.

There are two types of extracted URLs: Referrer URL, which contains the packet source data, and get/post URI as the packet destination and host URI. For example, when a user visits google.com/index.html through an Internet browser, the referrer URI is empty and the host URI and get URI are set to google.com and /index.html, respectively. To navigate from google.com/index.html to yahoo.com/search, the referrer URL becomes google.com/index.html, whereas the host URI and get URI become yahoo.com and /search, respectively.

```
GET /zz/combo?nn/lib/metro/g/breakingnews/breakingnews_0.0.51.css HTTP/1.1
Accept: */*
Referer: http://www.yahoo.com/
Accept-Language: ko-KR
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0;IPMS/CD0310AC-1529EE3CACA-000
Accept-Encoding: gzip, deflate
Host: l.yimg.com
Connection: Keep-Alive
```

Fig. 7 Sample of Referrer URL at yahoo.com

Using such data, the algorithm gathers the referrer URL, get/post URI, and host URI data from the time packet session begins to the time it ends. Using the collected URL data, their

Open Science Index, Computer and Systems Engineering Vol:7, No:12, 2013 publications.waset.org/9996682/pdf

sequence can be checked. URLs whose sequences are checked become part of the inspection target URL set.

In the existing malicious network detection systems, the inspection target URLs are compared with the malicious behavior pattern one by one to confirm the malicious behavior. In such case, the number of malicious behavior patterns compared is the same as the number of web pages creating the malicious behavior.

Note, however, that the proposed algorithm can visit only some of the inspection target URL set and extract the URLs (visit inspection target URL set), as if the whole set is inspected.

Visiting a website can be divided into a case of visiting with full recognition by the user and a case of automatic visit without the user being aware of it. The above inspection target URL set includes both cases.

A web page visited by a user with full recognition has a clear destination. For example, when a user navigates from yahoo.com to google.com, google.com becomes the destination. In such case, the generated packet will have yahoo.com as referrer URL and google.com as get URI. Note, however, that there are other websites visited, although the user is not aware of it. While navigating from yahoo.com to google.com, the user can check many contents such as news article link, advertising, and YouTube page. Such content pages are automatically visited and displayed through the Web browser even though the user did not specifically request such. When yahoo.com is set as the referrer URL, each content page will become the destination get/post URL.

The existing malicious network detection technologies must inspect all such URLs, but this algorithm only has to visit some yahoo.com content pages and google.com to inspect all URLs.

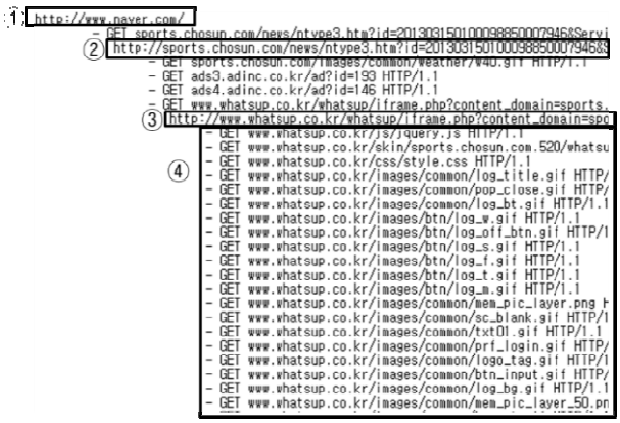


Fig. 8 Sample Structure for Collected URLs and Referrer URLs

Fig. 8 shows the result of actual traffic. It is an example of extracting URLs after analyzing the actual network traffic data and subsequently detecting all URLs with minimum visits. As a user visits URL ①, URLs ②~④are also visited without the user being aware of it. URL ④ is visited without the user being aware of it, and it will have URL ③, the previous Web page, as the referrer URL for the visit.

This algorithm visits only URLs ①, ②, ③, and ④ to have the same effect as visiting all URLs.

The algorithm can show better performance by separately extracting the URLs involved in file creation from the visit inspection target URL set extracted by the algorithm and inspecting them. Note, however, that such method will not be able to detect the malicious behaviors directly attacking the user PCs without creating the file in the exploit kit.

Therefore, this algorithm only extracts the visit inspection target URL set to detect all suspicious malicious behaviors that can be generated by the network traffic.

The proposed algorithm is depicted in Fig. 9 below.

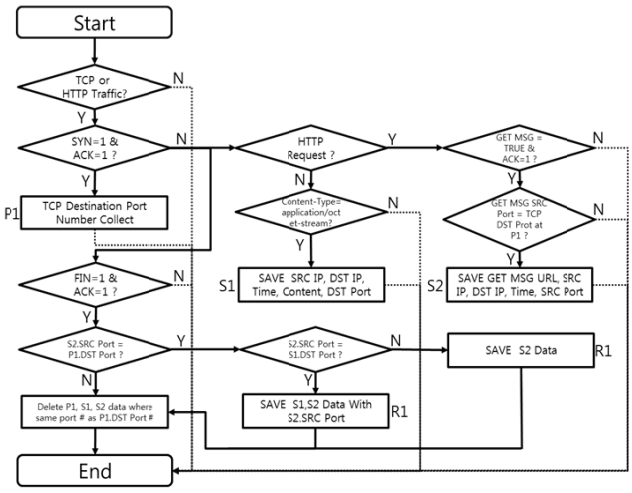


Fig. 9 Proposed Algorithm

C. Algorithm Verification Method and Result

The verification of the proposed algorithm in an actual environment shows that its performance in terms of detecting malicious behaviors was at least 5.996 times the existing methods.

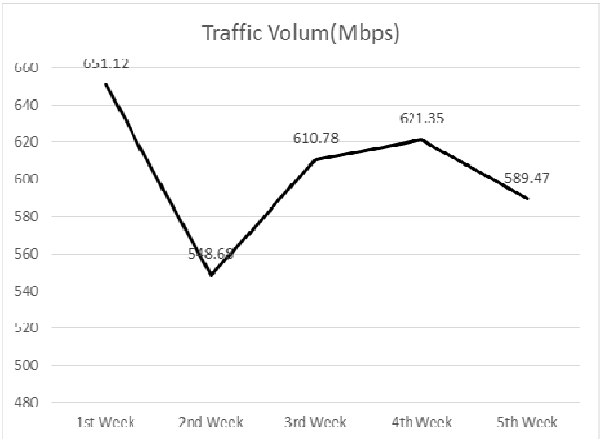


Fig. 10 Average Internet Traffic Volume at a Real Networking

The algorithm's performance was tested in a business site using the KT Internet network in Korea for the whole month of August 2013. Average daily traffic volume was around 600Mbps, and 231,258 inspection target URLs were extracted

per minute. A typical midsized server was used for the verification of the algorithm. Fig. 10 shows the bandwidth graph tested for the whole month of August, and Fig. 11 shows the average number of URLs collected per minute.

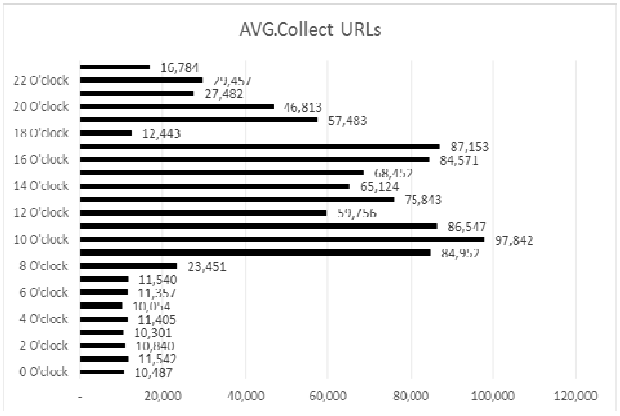


Fig. 11 Average of Collected URLs in one Minute

When the proposed algorithm was applied to the collected URLs, the visit inspection target URL set contained 38,657 or 1/6 of the total number of URLs; the algorithm improved detection performance to 5.996 times the existing technology.

A total of 147 URLs were found to be related to the file creation of all referrer URLs extracted by the proposed algorithm. Of these, two URLs were detected to be actually related to the penetration of malicious code.

The performance differences between the proposed algorithm and existing malicious website inspection technologies are shown in Table I below.

TABLE I
TEST RESULT OF PROPOSED ALGORITHM

	Wepawet [14]	Network Filtering	Suggesting Algorithm
Avg. Number of Target URLs per Min.	231,258	231,258	38,657
Detection Ratio	312	1	0.167
Avg. Analysis Period per URL	5m 35s	0.38s	0.38s

IV. CONCLUSION

The proposed algorithm could detect the "Preparation/Penetration" step among the APT attack steps. It also supplemented some of the weaknesses of the existing security systems. It could detect APT attacks with unknown pattern. Moreover, it could identify more effectively malicious behaviors transmitted into enterprises by extracting all URLs identified to be the problem by the network traffic analysis and some visit inspection target URLs concerning file inspection.

Nonetheless, the algorithm should be verified if it can inspect the enlarged network band or the increased number of internal users. There is also a need to develop the technology to monitor the detection of APT attacks by applying the proposed algorithm for multiple enterprises.

The automatic in-depth URL analysis technology should be developed to enable profiling of APT attacks. Based on such profiling, malicious code data, hacker data, exploit kit data, and malicious code-distributing/landing site data should be recorded in a DB so that any similar penetration incident can be detected early.

ACKNOWLEDGMENT

This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

REFERENCES

[1] Michael K. Daly, "The Advanced Persistent Threat," LISA `09
[2] Mandiant, the Advanced Persistent Threat, M. Trends, 2010
[3] Giura.P, Wei Wang, "A Context-Based Detection Framework for Advanced Persistent Threats," Cyber Security 2012 International Conference, pp. 69-74, 2012
[4] Ajay K. Sood, "Modern Malware and APT: What You May be Missing and Why'" AtlSecCon, March 2012
[5] Gang Wang, Jack W. Storkes, Cormac Herley, DividFelstead, "Detecting Malicious Landing Pages in Malware Distribution Networks," 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp.1-11, June 2013
[6] J. Zhang, C. Seifert, J. W. Stokes, and W. Lee., "ARROW: Generating signatures to detect drive-by downloads," In Proceedings of the 20th Annual World Wide Web Conference (WWW), pp. 187-196, Hyderabad, India, March 28 - Apr. 1, 2011
[7] BITS, "Malware Risk and Mitigation Report," BITS, June 2011
[8] Niels Provos Panayiotis Mavrommatis Moheeb Abu Rajab Fabian Monrose, "All Your iFRAMEs Point to Us," Google Technical Report provos-2008
[9] Y. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen and S. King, "Automated Web Patrol With Strider Honeymonkeys: Finding Web Sites That Exploit Browser Vulnerabilities", in 13th Annual Network and Distributed System Security Symposium. San Die: Internet Society, 2006
[10] Christian Seifert, Ian Welch and Peter Komisarczuk, "Application of divide-and-conquer algorithm paradigm to improve the detection speed of high interaction client honeypot", SAC'08, pp. 1426-1432, March 2008
[11] Ali Ikinci, Thorsten Holz, and Felix Freiling. (2008). "Monkey-Spider: Detecting Malicious WebSites with Low-Interaction Honeyclients," In Proceeding of Sicherheit, Schutz and Zuverl.
[12] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy, "A crawlerbased study of spyware on the web," in Proc. NDSS, 2006.
[13] Long Lu, Vinod Yegneswaran, Phillip Porras, Wenke Lee "BLADE: An Attack-Agnostic Approach for Preventing Drive-By Malware Infection", CCS'10, 10.2010.
[14] Wepawet, http://wepawet.iseclab.org, UCSB
[15] Monkey Wrench, http://monkeywrench.de, G Data Software AG
[16] Jon Oliver, Sandra Cheng, Lala Manly, Joey Zhu, Roland Dela Paz, Sabrina Sioting, and Jonathan Leopando "Blackhole Exploit Kit: A Spam Campaign, Not a Series of Individual Spam Runs," Trend Micro Incorporated Research Paper, 2012
[17] Rebecca Wynn, "Exploit Kits – Cybercrime Made Easy," Hakin9 IT Security Magazine, pp. 18-25, Jun 2011



Byung-Ik Kim was born in 1983 in GyeongJu, Korea. He received the B.S. degree in information and computer science from Ajou University, in February 2010.His research interests include computer security, malware analysis and .network security

He is currently working in Korea Internet & Security Agency in Korea.



Hong-Koo Kang was born in 1979 in Uijeongbu, Korea. He received M.S degree and Ph.D. degree in computer information & communication engineering from Konkuk University in Korea in2004 and 2009 respectively.His research interests are malware analysis and profiling.

He is currently working in Korea Internet & Security Agency in Korea.

Open Science Index, Computer and Systems Engineering Vol:7, No:12, 2013 publications.waset.org/9996682/pdf



Tae-Jin Lee was born in 1976 in GangNeung, Korea. He received M.S degree and Ph.D. degree in computer information & communication engineering from Postech University and Yonsei University in Korea in2004 and 2009 respectively.His research interests are malware analysis and network security.

He is currently working in Korea Internet & Security Agency in Korea.



Hae-Ryong Park was born in 1974 in GwangJuKorea. He received M.S degree in cryptography from Seoul National University in and Ph.D. degree in 2001cryptograpy from Chonnam National University in2006respectively.His research interests are cryptography and malware analysis.

He is currently working in Korea Internet & Security Agency in Korea as a Manager.

Open Science Index, Computer and Systems Engineering Vol:7, No:12, 2013 publications.waset.org/9996682/pdf