ISSN:2321-1156

International Journal of Innovative Research in Technology & Science(IJIRTS)

NETWORK-BASED EXAMINATION TARGET EXTRACTION TECHNOLOGY TO DETECT APT ATTACKS EFFECTIVELY

Byungik Kim, KISA; Haeryong Park, KISA; Kyungho Chung, KISA

Abstract

Recently, long-term, advanced cyber-attacks have been occurring again, targeting a specific enterprise or organization. These attacks occur over a long period and bypass detection by security systems unlike the existing attack pattern. For such reason, they create problems such as delayed real-time response and detection after the damages have already been incurred. This paper introduces the design of technology that applies network traffic real-time monitoring to detect unknown functional cyber-attack at the network. The algorithm was verified and evaluated in terms of performance in an actual commercial environment. Cyber-attack detection performance is expected to be improved by enhancing the algorithm and processing a large volume of traffic.

I. Introduction

Cyber-attacks of late usually target a specific enterprise or organization. Such cyber-attacks go beyond just system hacking or web server operation, causing economic and social problems.

Such attack against a specific target is called advanced persistent threat (APT). It has the characteristics of being prepared over a long period and of bypassing the existing intrusion detection system [1],[2],[3],[4].

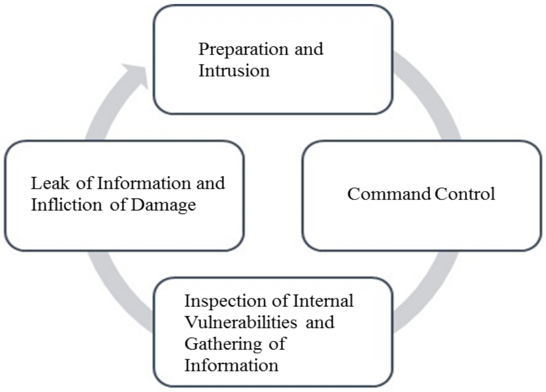

Preparation and Intrusion
Command Control
Inspection of Internal Vulnerabilities and Gathering of Information
Leak of Information and Infliction of Damage

Figure 1. APT Attack Phases

An APT attack is divided into four steps (see Figure 1): "Preparation and Intrusion"; "Command Control"; "Inspection of Internal Vulnerability and Gathering of Information," and; "Leak of Information and Infliction of Damage." Such steps are carried out over a long period.

To intrude an attack target, the attacker passes a URL that can infect the member of the target organization with the malicious code and installs the delivered malicious code to position itself inside the target organization. The attacker transmits other malicious codes to understand other vulnerabilities and the network structure of the target organization. The transmitted malicious code then checks the internal vulnerable server and gathers information concerning the access method and privilege of the server at the command of the hacker. The malicious code also gathers important documents in the vulnerable server and sends them to the attacker at the attacker's command. Lastly, the attacker leaks the information or destroys the major servers of the target enterprise or organization [4].

Such attack process occurs over a long period and minimizes the traffic volume to bypass the existing detection system. Moreover, detecting it takes a long time since it uses unknown or customized malicious code.

Table 1. APT Attack vs. Other Cyber-Attacks

	APT	BotNet	Malware
Target	Targeted	Unspecific	Unspecific
Frequency	Long-term	One-time	One-time
Instrument	New Malware	URL Downloaded Bot	Malicious Program

Table 1 shows the comparison of APT attacks and other cyber-attacks.

This paper seeks to introduce the network traffic data analysis algorithm to detect the APT attack that bypasses the existing detection system. The rest of this paper is organized as follows: Chapter II describes the existing detection technologies; Chapter III suggests the network data analysis algorithm to detect APT; Chapter IV describes the result of verifications of the suggested algorithm in the commercial environment; Lastly, Chapter V presents the conclusion by introducing the future research direction.

II. Related Research

Chapter II describes the existing technologies to detect cyber-attack.

A. Signature-Based Malicious Code Detection Technology

The first step in responding to a cyber-attack is to collect and analyze the malicious code used for the cyber-attack. The malicious code is a file that performs malicious behavior based on the attacker’s intention and an essential tool used for cyber-attack. Therefore, it is most important to collect such malicious code and analyze how the attack was executed and how the damage was incurred.

The signature-based malicious code detection technology can analyze such malicious code in real time. As the leading signature-based malicious code detection technology, the vaccine stores the unique values of malicious codes of past cyber-attacks and checks if the same malicious code exists inside the system [5],[6].

For that, highly skilled analysts analyze the malicious codes and extract the malicious behavior data from the analysis result. If files behaving similarly to the extracted malicious code data exist, maliciousness is determined using a simple file analysis/comparison technique. Figure 2 illustrates how a vaccine detects the malicious code.

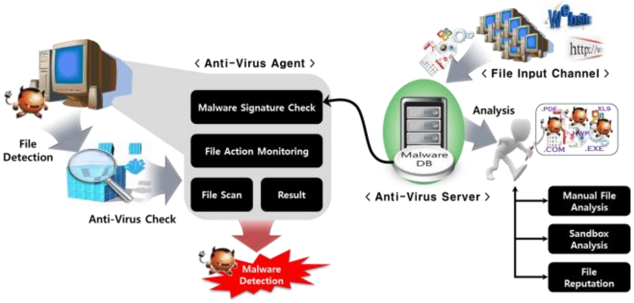


Figure 2. Operating Steps of Antivirus Products

Such signature-based malicious code detection technology can quickly detect the known malicious code or similar malicious code. Moreover, it can detect new malicious codes by simply exchanging the file’s unique data [6],[7].

As its weakness, however, it is difficult to collect highly advanced malicious codes such as APT attack. Moreover, the damage cannot be prevented since the data are collected after the cyber-attack has occurred.

B. Network Anomaly Detection System

In addition to the technology for detecting malicious code as introduced in Chapter II. Section A, this section describes another technology to detect and prevent cyber-attack by grasping the network anomaly.

The network-based cyber-attack detection technology detects cyber-attack by using the difference between the normal network traffic and anomalous network traffic [10],[11]. For that, it analyzes the characteristics of normal network traffic and identifies the traffic deviating from the normal traffic pattern.

This technology extracts the normal traffic characteristics using the analysis of total traffic volume, internal/external connection condition, and long-term traffic distribution trend analysis during a specific period. The extracted traffic characteristics are turned into a traffic pattern, with more forms of traffic patterns created given longer normal traffic period [7],[9].

Unlike normal traffic, the cyber-attack will show unknown traffic characteristics that are different from the existing traffic patterns. Many malicious code-infected PCs periodically access the unknown outside IP to get the command of the attacker. Moreover, they show abnormal traffic behaviors such as downloading the malicious code or increased traffic volume to leak the collected information. Separately, it can detect and block specific external IP, URL addresses, and files to prevent the attack.

Nonetheless, it has the weakness of not being able to detect attacks such as APT, which occurs over a long period of time and has a pattern similar to normal traffic unless they are analyzed [1],[4].

This paper suggests a total network monitoring-based APT attack detection algorithm to overcome such weakness.

III. APT Attack Detection Algorithm

As described in Chapter I, APT attacks occur secretly and over a long period of time, targeting specific organizations and enterprises; thus, the existing signature-based malicious code detection technology and network anomaly detection system cannot detect them in real time. Chapter III suggests an APT attack detection algorithm based on the real-time analysis of network traffic.

A. Suggested Algorithm

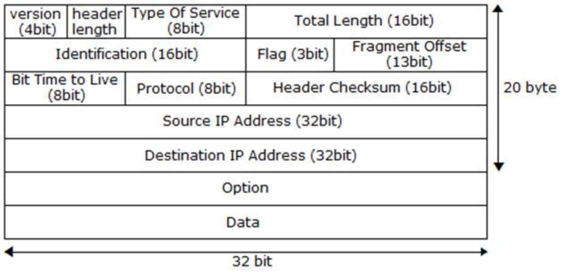
An APT attack generally begins with a URL that can download the malicious code. Therefore, this paper introduces the algorithm to detect such malicious code-downloading URLs.

This algorithm monitors all URLs penetrating into the organizations across the network and identifies a series of URL sets that transferred files. It then inspects some of the identified URL sets (URL sets to be inspected) to form the URL sets subject to visit inspection so that all identified URL sets can be inspected.

The URL sets subject to visit inspection enable detecting the APT attack preparation step, such as malicious code distribution, malicious code passage/source checking, checking of infected internal PC, etc.

B. Principle and Description of Algorithm

To apply the algorithm, all network traffic transmitted over the internal/external network must be collectable. The headers of the collected traffic will be extracted through network packet analysis. The extracted header data becomes the Source IP/port and destination IP/port information. It also extracts the Windows executable file (.exe) transmitted to the traffic to detect the APT attack.



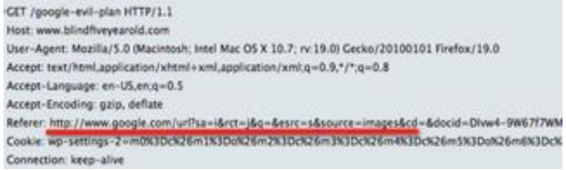
version (4bit)	header length	Type Of Service (8bit)	Total Length (16bit)	
Identification (16bit)		Flag (3bit)	Fragment Offset (13bit)	
Bit Time to Live (8bit)	Protocol (8bit)	Header Checksum (16bit)		
Source IP Address (32bit)				
Destination IP Address (32bit)				
Option				
Data				

Figure 3. IP Header Structure

The extracted data are sorted into Source IP/port and destination IP/port pairs, and the additional URL data are extracted. The extracted URLs can be separated into the Request URLs and Referrer URL that generates the traffic.

All web communications contain the Referrer URL and Request URL, and the suggested algorithm can analyze the relation of these URLs to detect the APT attack preparation step.

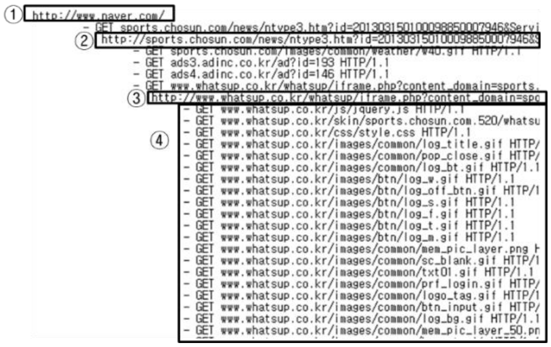
When a web communication session is activated, its Referrer URL has a Null value, and the Request URL has the destination data of the packet. For example, when a user attempts to access google.com/index.html with an Internet browser, the Referrer URL is empty, and the Request URL will have the value of google.com/index.html. Figure 4 shows an example of a traffic header that contains the data.



```
GET /google-evil-plan HTTP/1.1
Host: www.blindfiveyearold.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&docid=Dlvw4-9W67f7WM
Cookie: wp-settings-2=m0%3Dc%26m1%3Do%26m2%3Dc%26m3%3Dc%26m4%3Dc%26m5%3Do%26m6%3Dc%
Connection: keep-alive
```

Figure 4. TCP Traffic Header and Referrer URL

When a user access an Internet page, the Web page generally calls URLs of which the user is not aware. Such URLs are extracted using the link data inserted in the web page and are mostly related to the referred websites or web banners. For example, if a user wants to move to Yahoo.com from Google.com, Yahoo.com becomes the Request URL. In such case, the generated packet will contain Google.com as the Referrer URL and Yahoo.com as the Request URL. If the user is not aware of it in such case, however, the URL addresses of various contents of Yahoo.com -- such as news article links, advertising windows, and YouTube pages -- become the Request URL. Such automatically created Request URLs are the traffic generated by Yahoo.com and are called without users being aware. Figure 5 shows such data.



```
① http://www.naver.com/
        - GET sports.chosun.com/news/ntype3.htm?id=201303150100098850007946&Servi
      ② http://sports.chosun.com/news/ntype3.htm?id=201303150100098850007946&S
           - GET sports.chosun.com/images/common/weather/w40.gif HTTP/1.1
           - GET ads3.adinc.co.kr/ad?id=193 HTTP/1.1
           - GET ads4.adinc.co.kr/ad?id=146 HTTP/1.1
           - GET www.whatsup.co.kr/whatsup/iframe.php?content_domain=sports.
         ③ http://www.whatsup.co.kr/whatsup/iframe.php?content_domain=spo
              - GET www.whatsup.co.kr/js/jquery.js HTTP/1.1
              - GET www.whatsup.co.kr/skin/sports.chosun.com.520/whatsu
              - GET www.whatsup.co.kr/css/style.css HTTP/1.1
          ④   - GET www.whatsup.co.kr/images/common/log_title.gif HTTP/
              - GET www.whatsup.co.kr/images/common/pop_close.gif HTTP/
              - GET www.whatsup.co.kr/images/common/log_bt.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/btn/log_w.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/btn/log_off_btn.gif HTTP/1
              - GET www.whatsup.co.kr/images/btn/log_s.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/btn/log_f.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/btn/log_t.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/btn/log_m.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/common/mem_pic_layer.png H
              - GET www.whatsup.co.kr/images/common/sc_blank.gif HTTP/1
              - GET www.whatsup.co.kr/images/common/txt01.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/common/prf_login.gif HTTP/
              - GET www.whatsup.co.kr/images/common/logo_tag.gif HTTP/1
              - GET www.whatsup.co.kr/images/common/btn_input.gif HTTP/
              - GET www.whatsup.co.kr/images/common/log_bg.gif HTTP/1.1
              - GET www.whatsup.co.kr/images/common/mem_pic_layer_50.pn
```

Figure 5. Request URL vs. Referrer URL

Although users visit URL ①, traffic-accessing URLs ②~④ are created. URL ④ is the URL automatically called by the web page without users being aware, and its Referrer URL becomes URL ③. As such, when a user visits URL ③, the user effectively visits URL ④ automatically.

Using the information, the suggested algorithm collects the referrer URL and Request URL data from the time the traffic begins to the time it ends and analyzes their navigation sequence. Moreover, it uses the characteristics of web traffic such as URL ③ to create the minimum visit inspection URL set for detecting the URL's downloading of the malicious code. Figure 6 shows the sequence of the suggested algorithm.

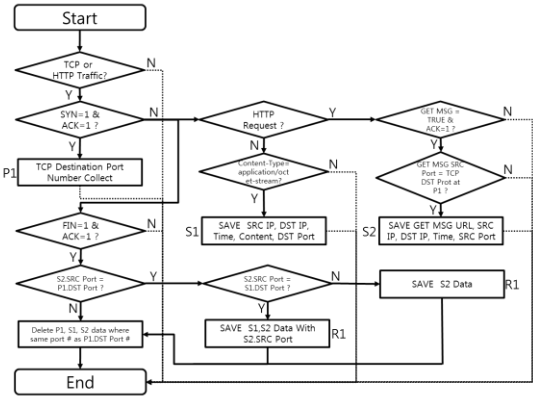

Start
TCP or HTTP Traffic?
N
Y
SYN=1 & ACK=1 ?
N
Y
P1
TCP Destination Port Number Collect
FIN=1 & ACK=1 ?
N
Y
S2.SRC Port = P1.DST Port ?
Y
N
Delete P1, S1, S2 data where same port # as P1.DST Port #
End
HTTP Request ?
Y
N
Content-Type= application/octet-stream?
N
Y
S1
SAVE SRC IP, DST IP, Time, Content, DST Port
S2.SRC Port = S1.DST Port ?
N
Y
SAVE S1,S2 Data With S2.SRC Port
R1
GET MSG = TRUE & ACK=1 ?
N
Y
GET MSG SRC Port = TCP DST Prot at P1 ?
N
Y
S2
SAVE GET MSG URL, SRC IP, DST IP, Time, SRC Port
SAVE S2 Data
R1

Figure 6. Suggested Algorithm

Since the existing network data-based analysis system inspects all generated URLs, it consumes unnecessary system resources and takes too much time to deduce the result of inspection. Note, however, that the suggested algorithm can analyze the malicious behavior of many URLs in a short period. Chapter IV introduces the result of performance verification of the suggested algorithm.

IV. Verification of the Suggested Algorithm

The verification of the suggested algorithm in an actual commercial environment indicated 5.996 times higher detection performance than the existing malicious behavior detection method.

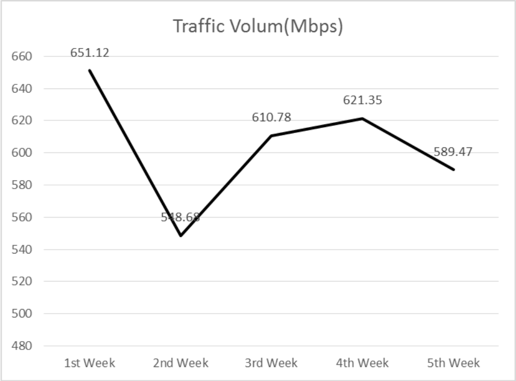


Figure 7. Collected Network Traffic Volume

For the whole month of February 2014, a business site using KT's Internet in Korea was used to test the performance of this algorithm. The daily average traffic volume was around 600Mbps, and 231,258 URLs to be inspected were extracted per minute. The server used for algorithm verification was a mid-size server. Figure 7 presents the graph of bandwidth inspected for the month of February, and Figure 8 shows the number of URLs collected per minute.

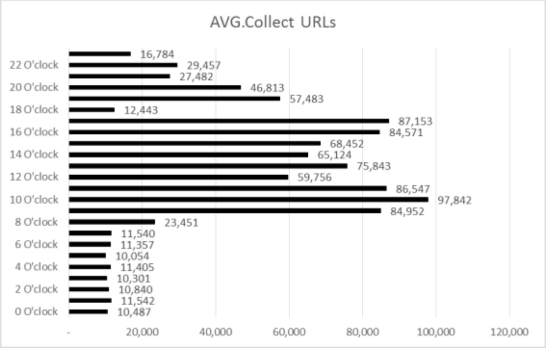


Figure 8. Average Number of Collected URLs

When the suggested algorithm was applied to the collected URLs, 38,657 visit inspection subject URL sets, or 1/6 of the total URLs, were extracted; this is equivalent to 5.996 times higher detection performance efficiency. Moreover, 147 URLs related to file creation were detected among the Referrer URLs extracted by the suggested algorithm. Among them, 2 URLs were found to be related to the actual penetration of malicious code.

Table 2 shows the difference in performance among the suggested algorithm and existing malicious website inspection technologies.

TABLE 2. Test Result of the Suggested Algorithm

	Wepawet [8]	Network Filtering	Suggested 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

V. Conclusion and Future Research

The suggested algorithm can detect the "Preparation and Intrusion" step of APT. It partly supplements the weakness of existing security systems by detecting the APT attacks with unknown pattern. Moreover, it solved the problem of network traffic analysis, i.e., inspecting all URLs and moving files, by extracting the visit inspection subject URLs so that the malicious behaviors penetrating in the enterprises can be effectively detected.

Nonetheless, the algorithm needs to be verified as to whether it can adequately inspect larger network bandwidth or users. A real-time APT attack detection monitoring technology applying the suggested algorithm – targeting multiple enterprises instead of one – must be developed.

Moreover, the in-depth automatic analysis technology of the collected URLs must be developed to profile the APT attacks. Based on such profiling, the malicious code, hacker data, exploit kit data used, and malicious code source/passage data should be developed into a DB so that that similar cyber-attack can be detected early.

Acknowledgments

This work was supported by the ICT R&D program of MSIP/IITP. [10044938, The Development of Cyber Attacks Detection Technology based on Mass Security Events Analysing and Malicious code Profiling]

References

[1] Ajay K. Sood, "Modern Malware and APT: What You May be Missing and Why'" AtlSecCon, March 2012.

[2] Giura.P, Wei Wang, "A Context-Based Detection Framework for Advanced Persistent Threats," Cyber Security 2012 International Conference, pp. 69-74, 2012.

[3] Mandiant, the Advanced Persistent Threat, M.Trends, 2010.

[4] Michael K. Daly, "The Adavnced Persistent Threat," LISA `09.

[5] Feng Xue, "Attacking the Antivirus," Black Hat Europe Conference, 2008.

[6] Wei Yan, Erik Wu, "Toward Automatic Discovery of Malware Signature for Anti-virus Cloud Computing," Complex Sciences, 2009, pp. 724-728.

[7] Peter Mell, Karen Kent, Joseph Nusbaum, "Guide to Malware Incident Prevention and Handing," Computer Security, NIST, 2005.

[8] Wepawet, http://wepawet.iseclab.org, UCSB.

[9] Radoslav Bodo, Michal Kostenec, "Experiences with IDS and Honeypots-Best Practice Document," GEANT, 2012.

[10] Robert Drum, "ISD and IPS placement for network protection," CISSP, 2006.

[11] Suchita Patil, Pallavi Kulkarni, Pradnya Rane, B.B.Meshram, "IDS vs IPS," IRACST, 2012, pp.86-90.

Biographies

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 (KISA) in Korea.

Haeryong Park was born in 1974 in GwangJuKorea. He received M.S degree in cryptography from Seoul National University in and Ph.D. degree in 2001 cryptograpy from Chonnam National University in 2006 respectively. His research interests are cryptography and malware analysis. He is currently working in Korea Internet & Security Agency (KISA) in Korea as a Manager.

Kyungho Chung received B.S degree fron Hanyang University and Seoul National University. And He received M.S degree from Seoul National University. He worked at ETRI in 18 years and received Ph.D. degree in Virginia Polytechnic Institute in 2002. He is currently working in Korea Internet & Security Agency (KISA) in Korea as a Vice President.