International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 3, Issue 4, 2016, PP 27-32
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: http://dx.doi.org/10.20431/2349-4859.0304005
www.arcjournals.org

Unified Detection and Response Technology for Malicious Script-Based Attack

Soojin Yoon[1], Hyun-lock Choo[2], Hanchul Bae[3], Hwankuk Kim[4]

Information Security R&D Technology Sharing Center
KISA (Korea Internet & Security Agency), Seoul, Korea

Abstract: *Dynamic functions using scripts are being implemented in web browsers, and the threat exists of attacks such as DoS or information leaks that exploit such functions. Especially, DoS has many case studies, but detecting script-based attacks using existing technologies is difficult because they are executed in web browsers and have no separate execution file. In this paper, we propose an integrated response technology that combines proxy-type detection and corresponding signature generation.*

Keywords: *Web, web attack, web browser, script-based attack, HTML5*

1. INTRODUCTION

Sohu TV (2014) [1], a shared platform considered the YouTube of China, GibHub (2015) [2], and IMGUR (2015) [3], an image sharing site, have both suffered DoS attacks based on script.

Script-based attacks operate on web browsers using pure Java-script and HTML elements. Previous web attacks were based on drive by download, which downloads the execution file to the user's terminal. Script-based attacks are different from traditional attacks based on execution files, since such attacks have no execution files and are run automatically.

Due to the nature of script-based attacks, their detection with existing security devices has proven difficult. Our team has proposed several studies on script-based attacks, and taking and combining proxy-based and signature technologies, we have created a system of integrated technology proposed in this paper.

2. RELATIVE WORKS

2.1. Script-Based Attack

A script-based attack is malicious behavior on a web browser. For example, SohuTV, GibHub and IMGUR were attacked by massive traffic from user browsers. A script-based attack used functions to send traffic to user web browsers. The functions themselves were innocent and thus did not constitute malicious behavior, but when attackers replayed them, users did not know of their attacks.

The big difference is the attack tool. A drive-by-download tool is an executable file. So prevention lies in detecting a downloaded file and checking if the file is malicious. But a script-based attack runs a webpage through a user's web browser with no file downloaded. Moreover, after closing the browser, no evidence remains in a user's device except that they connected. Detecting web pages including malicious script is not unhelpful. In addition, with the emergence of HTML5 and new functions added to the web, the influence of script-based attacks will grow stronger.[4]

2.2. Previous Detection Method

Among the distinct features of a script-based attack, one detection method stands out.

Spy proxy [5] and Web Shield [6] run on proxy servers between web servers and users to protect users. Spy proxy catches a web page a user connects to and runs it though a VM worker. The VM worker detects behavior such as making a new process, manipulating a file or breaking a sandbox. Spy proxy also picks user actions on a web page before sending it to a web server. Spy proxy tests selected user actions on a VM worker and determines whether it is safe. Spy proxy can detect malicious web pages, but the criteria of detection are focused on drive-by-download attacks. Web

Shield also uses sandbox to run a web page. The major difference from Spy proxy is that a web browser in a sandbox makes a DOM structure and shares it with a user's browser instead the latter making the DOM structure from a web server's response.

J Sand [7] is specialized for third-party Java-script. It provides a platform to run third-party Java-script through the J Sand sandbox environment. It blocks the outer actions of a third-party Java-script that can cause malicious behavior or side effects, and is geared for web developers.

Our team has developed a method to detect script-based attacks. Signature-based [8], network based (proxy server) [9], website checker [10] and behavior based[11] are proposed. This paper mixed the features of signature- [8] and network-based [9] technologies.

3. OVERALL

The proposed technology can be mainly divided into three parts: network processing, detection engine and static signature generator. The first and second are upgrades of the previously proposed and developed network-based [9] technology. Third we simplified signature-based [8] technology.

4. UPGRADING NETWORK-BASED TECHNOLOGY

Network-based technology [9] is a proxy-based technology that works between the user and web servers. We created the network processing part using the existing proxy and implemented the rest.

Previously, the network processing and detection parts were separated. The result of operation after implementation showed that static analysis occurred frequently, but dynamic analysis occasionally. But the network processing and detection parts had to communicate whenever analysis took place. This is why the analysis took a lot of time.

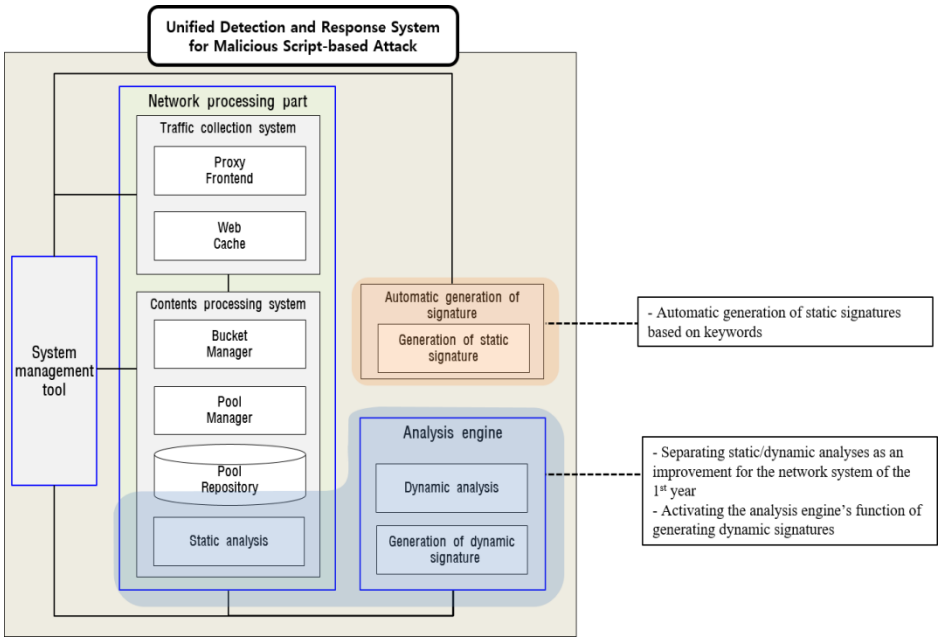

Figure1. *Unified Detection and Response System for Malicious Script-based Attack*

Different from existing technology, static analysis was installed in the network processing unit. The aim was to reduce the number of communications between the network processing and analysis parts caused by the static analysis.

In addition, signature-based [8] technology was simplified and installed. We will cover this in Chapter 5.

4.1. Network Processing Part

The network processing part consists of the traffic collection and content processing systems.

The traffic collection system was implemented using an open source proxy. It handles traffic between the user and web server, and separately stores web resources that require no processing. Because this technology targets HTML documents and scripts for detection, resources such as pictures and images are cached in the traffic collection system.

The content processing system receives HTML documents and scripts from the traffic collection system. Then it checks for external scripts referred in the HTML documents and scripts received. If external scripts are detected, the system also collects them as well. The content processing system

combines the collected scripts and performs static analysis, then judges whether dynamic analysis is necessary by observing the use of obfuscation or vulnerable Tag/API.

4.2. Detection Part

If dynamic analysis is required, scripts are passed to the analysis engine and then run using the Java-script engine. When the engine is running, no malicious actions occur because it is not a user environment. Using the Java-script engine, the API call is recorded then compared with the API call for malicious actions defined. The degree of similarity is measured, and the API call is judged to be malicious if the degree of similarity exceeds a certain threshold. For more details, refer to the previous paper [9].

5. SIMPLIFYING SIGNATURE-BASED TECHNOLOGY

Signature-based [8] technology was implemented and the occurring problems were identified to simplify the technology. Signature-based technology consists of pre-processing, clustering, generation and refinement. Among them, clustering binds the same malicious types. Signatures are generated by extracting common keywords from the clustered scripts. The problem is that as the number of clustered scripts increases, that of common keywords decreases.

To solve this problem, we attempted to reconstruct the clustering, but this triggered an additional problem in the management of previously created signatures. In the end, we decided to exclude the clustering process, and simplified the generation of signatures in the order of previous pre-processing, generation and refinement.

5.1. Signature Generation

Signature generation consists of three major stages; pre-processing, generation and refinement.

Pre-processing uses TF-IDF [12] [13] for the inputted script and determines the importance of each keyword. Here, the comparison values are keywords of other scripts of the same malicious type. TF-IDF is a method of calculating how representative a keyword is in a document by using the keyword's frequency. Previously, TF-IDF was used for clustering, but in this paper, we used it for calculating values used in keyword extraction and refinement.

Static signatures are generated by grouping the extracted keywords. These signatures have the form of a YARA [14] rule.

Refinement is a process of optimizing signatures by deleting keywords with low TF-IDF values when the keyword is larger than a predetermined value.

6. INTEGRATED TECHNOLOGY

As shown in Fig.1, integrated technology is the combination of network- and signature-based technologies.

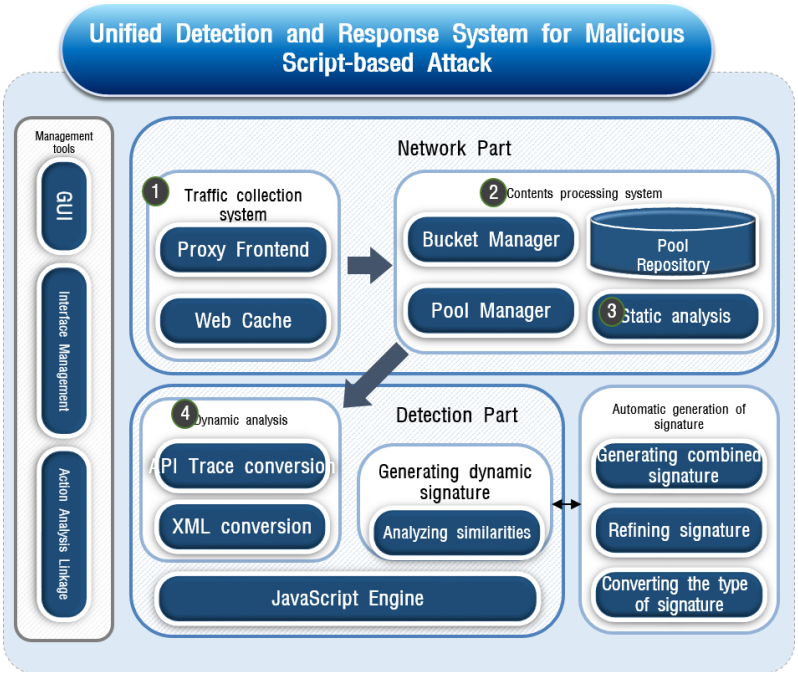

Figure2. *Detail Structure*

Figure 2 is the structure of implementing the integrated technology. In the system, an action analysis linkage indicates the linkage with the previous study, behavior-based [11] technology. This, however, is not an important element of the system and thus will not be discussed.

7. TEST

We measured the detection rate and web access delay time of the implemented system.

7.1. Detection Rate

For the subjects of the detection rate, we selected 23 possible items of writing and collecting sample web pages by aggregating HTML5 [4] and script-based attacks.

Table1.*Result of Detection*

	Malicious type	Malicious action	Related element		Result of detection
			Script	Tag	
1	DoS	Hash DoS	○		○
2	DoS	Server-Sent Event Bot	○		○
3	DoS	Worker DDoS	○		X
4	DoS	Recursion by SVG		○	X
5	DoS	Script DoS (SetInterval)	○		○
6	DoS	Script DoS(for)	○		○
7	DoS	Client DoS		○	X
8	Info Leak	Network Scan	○		○
9	Info Leak	Port Scan	○		○
10	Info Leak	Click Jacking Drag & Drop	○	○	○
11	Info Leak	Click jacking Click info		○	○
12	Info Leak	Web Socket Data Extortion	○		○
13	Info Leak	SVG Key logger		○	X
14	Info Leak	Mouse Logger	○		○
15	Info Leak	Geolocation	○		○
16	Info Leak	Web Storage Leak	○		○
17	Info Leak	Indexed DB Leak	○		○
18	Info Mod	Web Storage Modification	○		X
19	Info Mod	Indexed DB Modification	○		X
20	Info Mod	History Modification	○		○
21	Phishing	Auto complete Phishing		○	X
22	Request Forgery	Cross Site Web Socket Hijacking	○		○
23	Request Forgery	Cross Site Printing	○		○

Sixteen out of 23 attacks were detected. Among the seven undetected attacks, four were used tags that could not be detected by the script engine. Two were Web Storage Modification and Indexed DB Modification. Though they were attacks, judging their simple modification codes as malicious was difficult because they worked the same as normal functions. The final one, Worker DDoS, was not detected because Java-script called through the Worker was not recognized as external Java-script. We plan to eventually resolve these problems.

7.2. Web Access Delay Time

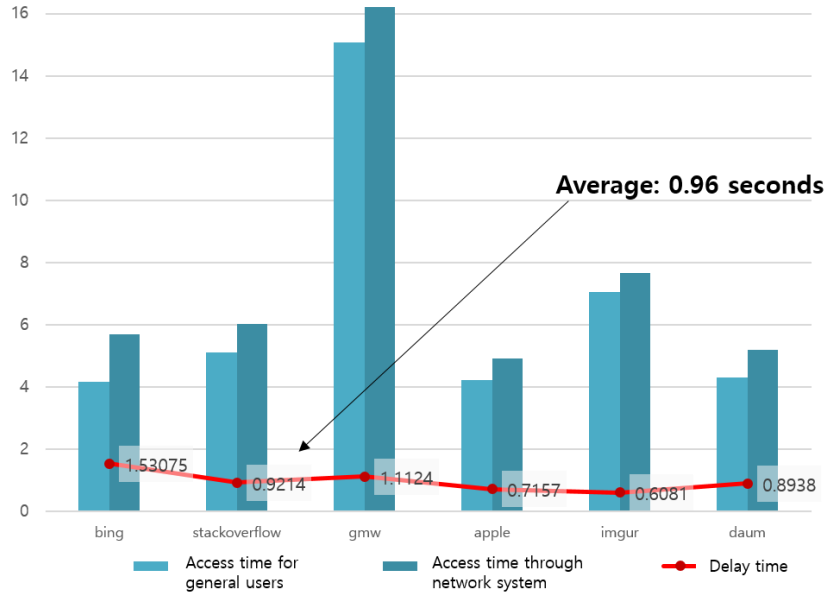

Figure3. *Web access delay time*

The developed system is installed in Korea. We measured the web access delay time for two sites, Bing and Daum, in which the servers are in Korea, and four sites, Stack overflow, Gmw, Apple and Imgur, where the servers are abroad. The method measured the web access delay time while accessing each site more than 100 times in IE11 using Http Watch. The average was calculated by excluding outlier values that were deemed too large or too small. With the developed system, the average web access delay time was 0.96 seconds.

8. CONCLUSION

Script-based attacks have features of being automatically executed in web browsers and are different from existing web attacks. So, this paper implemented and tested a specialized system. This independent system collects web information based on proxy, performs DCI (deep content inspection) and generates signatures by itself. But tests showed a flaw of being unable to cover attacks using tags. In future studies, we plan to improve the system by studying measures for attacks with tags and by boosting the system's performance.

ACKNOWLEDGEMENTS

This work was supported by the ICT R&D Program of MSIP/IITP. [B0101-15-0230, The Development of Script-based Cyber Attack Protection Technology]

REFERENCES

[1] One of World's Largest Websites Hacked: Turns Visitors into 'DDoS Zombies', INCAPSULA, https://www.incapsula.com/blog/world-largest-site-xss-ddos-zombies.html

[2] GitHub jammed by injected JavaScript, servers whacked by DDoS, The Register, http://www.theregister.co.uk/2015/03/27/github_under_fire_from_weaponized_great_firewall/

[3] Imgur suffers DDoS attack on 4chan and 8chan servers, SC magazine, http://www.scmagazine.com/imgur-suffers-ddos-attack-on-4chan-and-8chan-servers/article/440522/

[4] Soojin Yoon, JongHun Jung, and Hwan Kuk Kim. "Attacks on Web browsers with HTML5." ICITST 2015, 2015.

[5] Moshchuk, Alexander, et al. "SpyProxy: Execution-based Detection of Malicious Web Content." USENIX Security. 2007.

[6] Li, Zhichun, et al. "Web Shield: Enabling Various Web Defense Techniques without Client Side Modifications." NDSS. 2011.

[7] Agten, Pieter, et al. "JSand: complete client-side sandboxing of third-party JavaScript without browser modifications." Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012.

[8] Soojin Yoon, et al. "Automatic attack signature generation technology for malicious javascript." ICMIC, 2014.

[9] Hyunlock Choo, et al. “The Analysis Engine for Detecting the Malicious JavaScript”, ICOINI, 2014.

[10] Hanchul Bae, et al. “Study on Inspection of Website Vulnerability and Risk Assessment Method”, World IT Congress 2016, 2016.

[11] Choo, Hyun Lock, et al. "The Behavior-Based Analysis Techniques for HTML5 Malicious features.", IMIS 2015, 2015.

[12] G. Salton, "Automatic information organization and retrieval", 1968.

[13] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, Vol.28, No.1, 1972, pp.11-21.

[14] YARA, http://virustotal.github.io/yara/

AUTHORS' BIOGRAPHY

Soojin Yoon, received the B.S. in computer engineering and M.S. in Security from Korea University, Korea. She is a researcher in Cyber Security R&D at KISA since 2014. Her current research interesting includes Web Security and Mobile Security.

Hyun-lock Choo, received the B.S. degree in computer multimedia engineering from Pykyong National University, Korea, and M.S. Candidate in security from Sungkyungkwan University, Korea. He is a duputy general researcher in Cyber Security R&D at KISA since 2014. His current research interesting includes Network/Web security, Cloud Service, Big Data Analysis, IoT, Technology Valuation and etc.

Hanchul Bae, received the B.S. degree in computer science from Yonsei University, Korea, and M.S. degree in electronic engineering from Korea Polytechnic University. He had worked for developing solutions and services of mobile and web over 10 years. He is a deputy general researcher in Cyber Security R&D at KISA since 2012. His current research interesting includes Mobile Security and Android App Security.

Hwankuk Kim, received the B.S. and M.S. degrees in computer engineering from Hankuk Aviation University, Korea, in 1998 and 2001. and Ph.D. Candidate in the graduate school of Information Security at Korea University, since 2009. He was as a researcher at ETRI until 2006 and currently is a team manager in Cyber Security R&D at KISA. His current research interesting includes ISMS, IoT Vulnerability Analysis, Wireless Network Secrity and its Application.