PAPER Special Section on Advanced Log Processing and Office Information Systems

Multi-Environment Analysis System for Evaluating the Impact of Malicious Web Sites Changing Their Behavior*

Yoshiaki SHIRAISHI[†a)], ***Senior Member***, **Masaki KAMIZONO**[††], **Masanori HIROTOMO**[†††], ***Members***, ***and*** **Masami MOHRI**[††††], ***Senior Member***

SUMMARY In the case of drive-by download attacks, most malicious web sites identify the software environment of the clients and change their behavior. Then we cannot always obtain sufficient information appropriate to the client organization by automatic dynamic analysis in open services. It is required to prepare for expected incidents caused by re-accessing same malicious web sites from the other client in the organization. To authors' knowledge, there is no study of utilizing analysis results of malicious web sites for digital forensic on the incident and hedging the risk of expected incident in the organization. In this paper, we propose a system for evaluating the impact of accessing malicious web sites by using the results of multi-environment analysis. Furthermore, we report the results of evaluating malicious web sites by the multi-environment analysis system, and show how to utilize analysis results for forensic analysis and risk hedge based on actual cases of analyzing malicious web sites.

key words: *drive-by download attack, web site analysis, multi-environment analysis, forensic, risk hedge*

1. Introduction

If malware infects some machines in companies, it causes them divulge assets such as sensitive information, account identity information, patent and so on. Recently, malware infection spreads by exploiting vulnerabilities of web browsers and their plugins instead of the operating system (OS) vulnerabilities, since the current OS has some functions to prevent vulnerable codes. Drive-by download (DBD) attacks are widely used for malware infection, and forces client users to download malware when they visit malicious web sites. So many researchers investigate the DBD attacks and develop countermeasure against the attacks.

For the protection of web users at client side in real time, malicious URLs provided by Google and Microsoft can be used as blacklist to prevent web users accessing malicious web sites. However, malicious codes redirecting web users to malware distribution sites are often inserted into valid web sites and the URLs of malicious web sites are frequently changed in short time [1]. Thus, malware prevention techniques using the blacklist are not always effective to protect web users. When an incident caused by malicious web sites occurred in organizations such as companies, they shall investigate what environment the incidents occurred in, what client was redirected, and what vulnerability was exploited in order to understand its cause and damage.

Some services provide results of analyzing web sites by automatic dynamic analysis. These services would analyze web sites in single environment because they do not provide results for several client environments. Most malicious web sites identify the software environment of the clients and change their behavior [2]. Thus, these services do not always give sufficient information appropriate to the organizations. It is required to prepare for expected incidents caused by re-accessing same malicious web sites from the other client in the organization. In order to minimize the risk, it is important to understand the behavior of malicious web sites under not only the environment at the incident but also all client environment of the organization.

Lu et al. [3] showed effectiveness of a technique for analyzing malicious web sites using multiple environment. Wang et al. [4] presented analysis technique of using honeypots, which runs multiple analysis environments. We also presented a similar study [5]. However, to authors' knowledge, there is no study of utilizing analysis results of malicious web sites for digital forensic on the incident and hedging the risk of expected incident in the organization.

In this paper, we propose a multi-environment analysis system for evaluating the impact of malicious web sites that identify the software environment of clients and change their behavior. The system focuses on an effectiveness for specific environments in the organization. Further, we present the results of evaluating the impact of accessing malicious web sites by the multi-environment analysis system, and show how to utilize analysis results for forensic analysis and risk hedge based on actual cases of analyzing malicious web sites.

This paper is organized as follows. In Sect. 2, we describe malicious web sites changing their behavior. In Sect. 3, we propose a system for evaluating the impact of malicious web sites by multi-environment analysis. In Sect. 4, we show some results of analyzing malicious web sites by the proposed system. In Sect. 5, we compare related works with the proposed method. In Sect. 6, we conclude this paper.

Manuscript received November 15, 2016.
Manuscript revised April 6, 2017.
Manuscript publicized July 21, 2017.

[†]The author is with the Graduate School of Engineering, Kobe University, Kobe-shi, 657–8501 Japan.
[††]The author is with PwC Cyber Services LLC., Tokyo, 104–0061 Japan.
[†††]The author is with the Graduate School of Engineering, Saga University, Saga-shi, 840–8502 Japan.
[††††]The author is with the Faculty of Engineering, Gifu University, Gifu-shi, 501–1193 Japan.

*This is a paper on system development.

a) E-mail: zenmei@port.kobe-u.ac.jp

DOI: 10.1587/transinf.2016OFK0001

2. Malicious Web Sites Changing Their Behavior

The mainstream of attacks in the Internet changes active attack to passive attack. Thus countermeasures of passive attacks grow increasingly important. Attacks using malicious web sites are classified as passive attack.

The literatures [1], [6] and the report of a security vendor [7] provide factual investigation into malicious web sites. Provos et al. [1] subjected over 60 million URLs for in-depth processing through their verification system during a period of ten months (Jan. 2007–Oct. 2007). Overall, they detected more than 3 million malicious URLs hosted on more than 180 thousand landing sites. The total of suspicious URLs by the end of Mar 2013 is more than 64.3 million. Suspicious URLs detected newly in the first quarter of 2013 is 2.6 million. The 94% of suspicious URLs include malware and exploit code [7].

Malicious web sites are often created by exploit kits. Exploit kits hosted in web server enable to create malicious web sites. Recently, the exploit-as-a-service (EaaS) providing exploit kits for attackers appears and enables attackers to construct the attack environments [6]. Figure 1 shows an example of attacks by malicious web site created by the exploit kit. The sites target browser vulnerabilities (e.g., Internet Explorer and Firefox) and plug-in vulnerabilities (e.g., PDF viewers, Flash, and Java) to force users to install malware. To increase the success rate of installing malware, some sites target several versions of applications and distinguish appropriate attacks to the types and versions of browser and plug-in in the client organization [2]. Additionally, the sites have a function to redirect to benign web pages when the version of applications in client machine is different from the targeted one in order to avoid detection [8].

Lu et al. [3] showed effectiveness of a technique for analyzing malicious web sites using multiple environment. Wang et al. [4] presented an analysis technique using honeypots runs multiple environments. We present similar study [5]. In this paper, we address to evaluate the impact of malicious web sites on victim organization by analyzing based on software environment of machine and network in the organization.

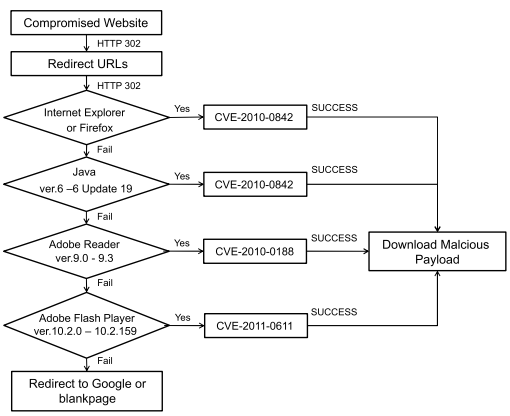

Fig. 1 Example of attack by malicious web site.

3. Impact Evaluation of Malicious Web Sites by Multi-Environment Analysis

3.1 Concept of Multi-Environment Analysis

The procedure of our multi-environment analysis is the same to web client honeypots. The web client honeypots collect the information of malicious web sites by actively accessing web sites. The honeypot is classified as low interactive honeypot (LIH) and high interactive honeypot (HIH). The LIH emulates various applications. The low interactive web client honeypot emulates web browsers, especially. As an advantage, the LIH performs speedily, but it does not contain actual vulnerabilities. On the other hand, the HIH is often composed of actual applications. Because the HIH has the vulnerabilities of the applications, it obtains more information than LIH. As a disadvantage, the HIH performs slowly and it needs to reset the virtual machine after exploiting the actual vulnerabilities in any malicious event. In this paper, we utilize a multi-environment analysis with HIH to collect the information on malicious web sites under several software environments in the client organization and the information on zero-day vulnerabilities. The configuration of multi-environment analysis is equipped with several types and versions of web browsers and plugins on the virtual machine. As shown in Fig. 2, the multi-environment analysis grasps the behavior of malicious web sites under not only the software environments at the incident but also several software environments in the victim organization.

3.2 Impact Evaluation System for Malicious Web Sites

Even if we collect the information on URLs redirecting the clients and the contents of web pages by the multi-environment analysis, we understand the event occurred in only the software environment. That is to say, we cannot understand the impact of any software environment in the client organization. In this section, we propose a system combining the following three methods for utilizing the information on multi-environment analysis in terms of digital forensic and hedging risk.

[Traffic Log Analysis] We identify URLs redirecting the clients for each software environment of the organization. For forensic analysis, we compare these URLs with the connection logs of firewall in the organization and browser in the machines. It helps us to understand attack progresses such as redirecting URLs, exploiting vulnerabilities and downloading malware. Meanwhile, for risk hedge, by accessing these web sites with other machines in the organization, we can estimate risks that the organization has not found.

[Content Analysis] We identify exploited vulnerabilities by reconstructing content files (e.g., HTML files, Jar files, PDF files, and JavaScript files) from the traffic

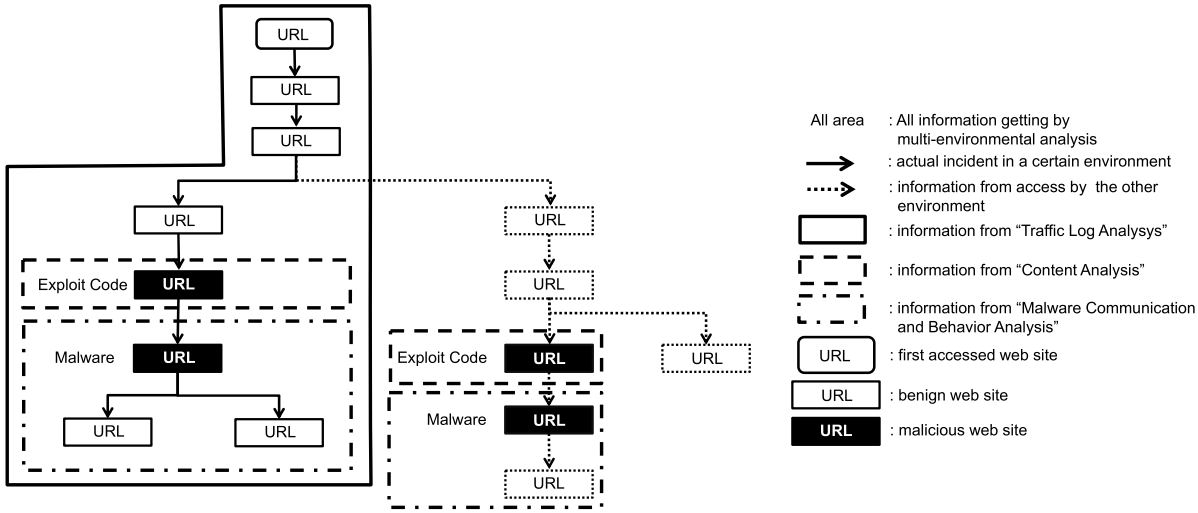

Fig. 2 Outline of the system output.

data under the same environment as the incident happened. Malicious scripts are often obfuscated to prevent us from analyzing, so it is difficult to collect the information which URL the client redirects. Therefore, we get the information on redirecting URLs and exploited vulnerabilities by using JavaScript analysis services [9], [10]. Additionally, we can identify CVE number from the information of malicious script detected by some anti-virus software. For forensic analysis, we compare the analysis results with connection logs of firewall in the organization and browser in the machines. It helps us to understand vulnerabilities inducing the incidents. Meanwhile, for risk hedge, by assessing the potential risk at other machines in the organization, we can predict incidents that the organization has not found.

[Malware Communication and Behavior Analysis]

When we find machines infected by malware, we have to investigate which machine has infected, what information has leaked out and where it has leaked out. The malware downloads its own body data from malware distribution sites and send the information stolen from the machines and organization. By uploading the malware or suspicious files restructured from the communication data to the service which dynamically analyzes the files as samples in the sandbox (e.g., Anubis [11] and ThreatExpert [12]), we can get the information whether the files are malicious and which URLs is visited by the client. For forensic analysis, comparing these URLs with the connection logs enables us to understand whether the malware actually communicates. Analyzing sent packets and received packets enables us to understand the behavior of malware (e.g., downloading the body data, directing from attackers, and sending the stolen information to attacker's server). Meanwhile, for risk hedge, by investigating the possibility of infecting other machines in the organization, we can predict incidents that the organization has not found.

The system flow which consists of the three methods is as follows:

1. A client accesses a target URL, then logs of firewall and browser are stored.
2. Traffic Log Analysis is applied to the logs, then the output is a URL transition diagram.
3. Content Analysis is applied to reconstructed files which are related to the URLs described as the nodes appeared in the diagram. If malicious contents are detected, the analysis results are recorded to the nodes.
4. Malware Communication and Behavior Analysis is recursively applied to reconstructed files which are related to the URLs described as the nodes appeared in the diagram. If malicious contents are detected, the analysis results are recorded to the nodes. If links to other URLs are detected, the diagram is updated.
5. If the organization has still other environments, repeat from 1. for a client selected from the remains of them.

The output information from the system is as shown in Fig. 2. Analyzing the results of multi-environment analysis with the above methods enables us to deal with not only actual incidents (indicated with solid arrow in Fig. 2), but also expected incidents that can happen in the organization (indicated with dotted arrow in Fig. 2).

When an incident occurs in the organization, we have to evaluate the impact of the current incident. As the evaluation, we firstly analyze the traffic logs in general. After we find exploiting vulnerabilities and downloading malware from the results of traffic log analysis, we reconstruct suspicious contents and analyze them. Then we evaluate the impact of the current incident from the results of content analysis. If we find malware in these contents, we analyze the malware communication and behavior. Then we evaluate the impact of the current incident from the results

of malware communication and behavior analysis. In order to estimate the risk of the organization from another aspect of the system, we evaluate the impact of incident which is expected to occur in the other environment. By accessing URLs found in the results of traffic analysis from the other environment, we can get several logs. We apply the traffic log analysis to the obtained logs, then apply content analysis and malware communication and behavior analysis to the results of traffic log analysis. By integrating these results, we can evaluate the impact of expected incident. Therefore, it is necessary that we execute three methods to evaluate the impact of current and expected incidents for forensic analysis and risk hedge. Figure 2 contains the results of the above analyses. It should be noted that the system is useful for specific environments in the organization.

For example, based on the evaluation results like Fig. 2, we can prevent the clients from accessing found malicious web sites at firewalls and proxy servers. Furthermore, we can apply patches and stop applications by identifying vulnerabilities of target applications. The actions would avoid the risk of potential threats.

4. Implementation and Experimental Results

4.1 Implementation for Experiment

A multiple analysis environment with the software shown in Table 1 is implemented. To analyze malicious web sites effectively, we prepared the environment with the oldest vulnerable browsers and plug-ins. The construction of analysis environment corresponding victim organization is available to changing the type and version of software. The analysis environment has a function of accessing URLs automatically by Selenium Webdriver [13] which automates browser operations of testing web applications. The system is executed on a Windows7 machine with Intel Core i7-2540M 2.8 GHz and 8 GB RAM.

4.2 Results

The malicious web sites posted in Malware Domain List [14] are visited as malicious sites by the proposed system. First, by using browsers shown in Table 1, we accessed 198 malicious web sites which exploit vulnerabilities and posted in Malware Domain List over a period of two months (7th Aug. 2013–9th Oct. 2013). Second, we picked up 30 malicious web sites changing their behavior by analyzing the communication data with Wireshark [15]. We classify them into 3 patterns in terms of different behavior against the browsers. Finally, for each patterns, we show how we evaluate the impact of malicious web sites by utilizing the multi-environment analysis for forensic analysis and risk hedge. We assume here that traffic log analysis is made available for logs of the victim organization at actual incidents. In what follows, we used firewall logs that is set to the implement environment to resemble logs of the organization actual incidents occurred. The running time of the proposed system is dominated by starting Guest OS, and the memory of the proposed system is almost used for running Guest OS.

Table 1 Implement environment of multi-environment analysis.

Virtualized Machine	VMWare Workstation 9.0.2
Guest OS	Windows 7 Professional
Host OS	Windows 7 Professional
Web browsers	Environment 1 : Internet Explorer 8 Environment 2 : Firefox 5.0 Environment 3 : Google Chrome 18.0.1017.2
Plug-ins	Java 6 Update 16 Adobe Reader 9.2 Adobe Flash Player 10 QuickTime Player 7.6.4 Shockwave Player 7.03.015

4.2.1 Case 1

We show the results of analyzing a malicious web site. The site led clients to suspicious web sites when the clients visit by using Internet Explorer and Firefox, and showed a benign behavior when the clients access by using Google Chrome.

We analyzed (A) www.******.com.br/wp-enter.php?xIKVC3UCMRU05WH6C by our multiple analysis environments. The result is shown in Fig. 3. We found that this site changes its behavior by the kind of web browsers. Visiting by Google Chrome, the clients were redirected to benign web sites (me******.com). On the other hand, visiting by Internet Explorer or Firefox, the clients were redirected to (B) http://78.**.**.**/closest/i9jfuhioejskveohnuojfir.php. Then, visiting by Firefox, the clients were redirected to http://78.**.**.**/closest//Main. Its HTTP status code was "404 Not Found". Visiting by Internet Explorer, the clients were redirected to (E) that has a lot of URLs (e.g., http://db******.com/ and http://ca******.com/) via (C) http://78.**.**.**/closest/i9jfuhioejskveohnuojfir.php?1aa10bb101bb00=b11.....,
(D) http://78.**.**.**/closest/i9jfuhioejskveohnuojfir.php?bbbbb00abab0bab1a1a=73.....

We consider how these results are utilized for forensic analysis and risk hedge. First, we analyze the connection logs to identify how the attack progressed. We set IP addresses of (A)-(E) recoded in firewall logs to the analysis environment. From the revealed result containing their IP addresses in the logs, we confirm that it actually accesses to (A)-(E). In this way, we can identify how the attack progressed.

Next, we analyze web contents of (B)-(E). We guess that (B) redirected to (C) because the JavaScript contained in HTML files of (B) has no other redirect code. By analyzing content-type of HTTP header contained in packets between the clients and (C), we identify that the client downloaded a Java archive file from (C). We reconstructed the Java file from the communication data and analyzed it by VirusTotal [16], 24 out of 48 antivirus software judged it to be malicious. We found the CVE number CVE-2013-0422 in the results. By checking the number in Japan Vulnerabil-

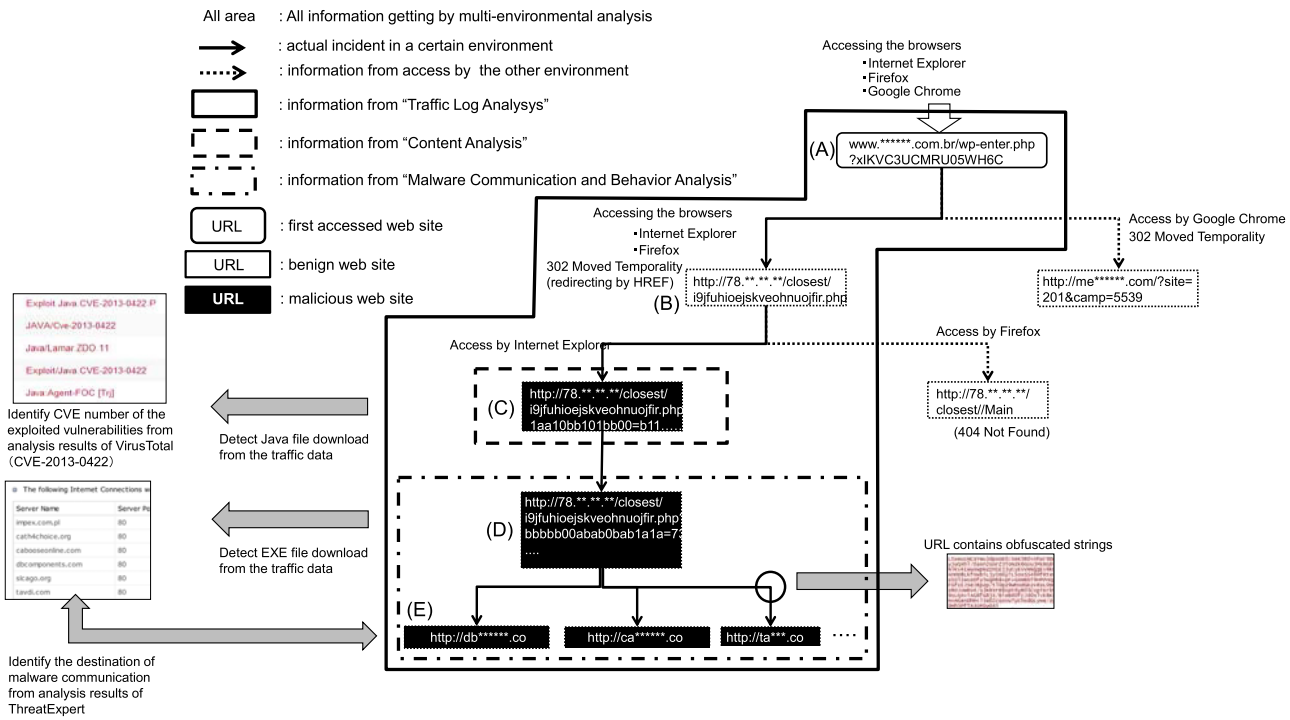

Fig. 3 Multi-environment analysis results of case 1.

ity Notes DataBase (JVNDB), it is prove to be Java vulnerability. We guess that some malicious files were downloaded from (D) by visiting to (C). Then, we checked the communication data between the clients and (D), and it was prove to be downloaded executable files. We checked the files downloaded from (E) by VirusTotal after restructuring from the communication data between the clients to (E). Then, 25 out of 48 antivirus software judged it to be malicious.

In the above results, the client was infected by exploiting Java application when it visited the malicious web site by Internet Explorer with a vulnerable Java application CVE-2013-0422. The Java application in the client is not updated to the latest version. Updating it reduces the risk for attacks by similar malicious web sites. Additionally, we can reduce vulnerabilities of the organization by ensuring the other machines from the organization.

Finally, we analyze the communication and behavior of malware. We analyzed the executable file reconstructed from the communication data by ThreatExpert, and found that the executable file accessed a lot of URLs. It implies that the executable file attempted to access (E) because these URLs correspond to (E). There are URLs consisted of unspecified number of domains in (E). By analyzing the communication data of the malware with Wireshark, it sent packets of 500-600 bytes to different URLs in (E). When we checked the sent packet, its body contained each different illegible character string every URLs in (E). The malware acquired the data of corresponding Web pages if HTTP status code is 200, whereas it received no data. By analyzing packets that the malware sent and received, it helps us to know what information the malware send to extra servers and whether the malware receives its own body data by receiving packets contained executable programs. If we acquire its behavior of referring and changing files and registry by using malware analysis services (e.g., Anubis and ThreatExpert), we can collect the information on the stolen data and its size by comparing the result with the logs of file and registry operations. Additionally, we assess the possibility that the similar malware infects the other machines in the organization by comparing its behavior of referring and changing files and registries with the logs in the other machines. If victim machine is left on the Internet, there is a risk for expansion of damage. Thus, we can prevent from leaking more information by stopping to use the machine, and attempt to remove the malware from the machine with antivirus software. Furthermore, by checking whether the machines in the organization accessed to (A)-(D) after tracing the past logs from the point of the incident, we can identify the existence of the other machines previously accessed to (A) or (B). We assess the possibility of exploiting by malicious web sites which are frequently visited by people in the organization if there are several times of accessing the web sites. In order to prevent other incidents, we can support to clarify preventive measures of regulating visit to (A) and stopping to use the target application in order not to occur the similar incidents.

4.2.2 Case 2

We show the results of analyzing (A) o******/wp-enter.php. The result is shown in Fig. 4. Visiting by Internet Explorer, the clients were redirected to (B) http://b******.s****.net:12601/post/chart/module.php?down=82 with iframe tags. Visiting by Firefox, the clients were redirected

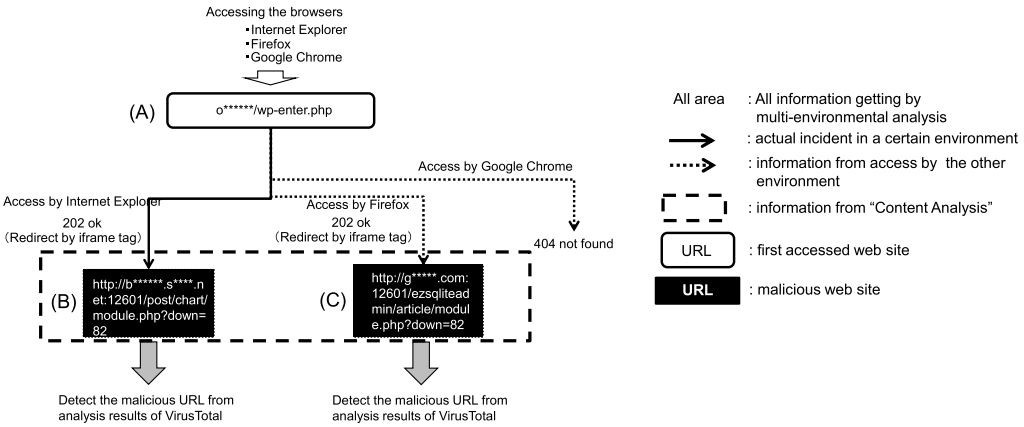

Fig. 4 Multi-environment analysis results of case 2.

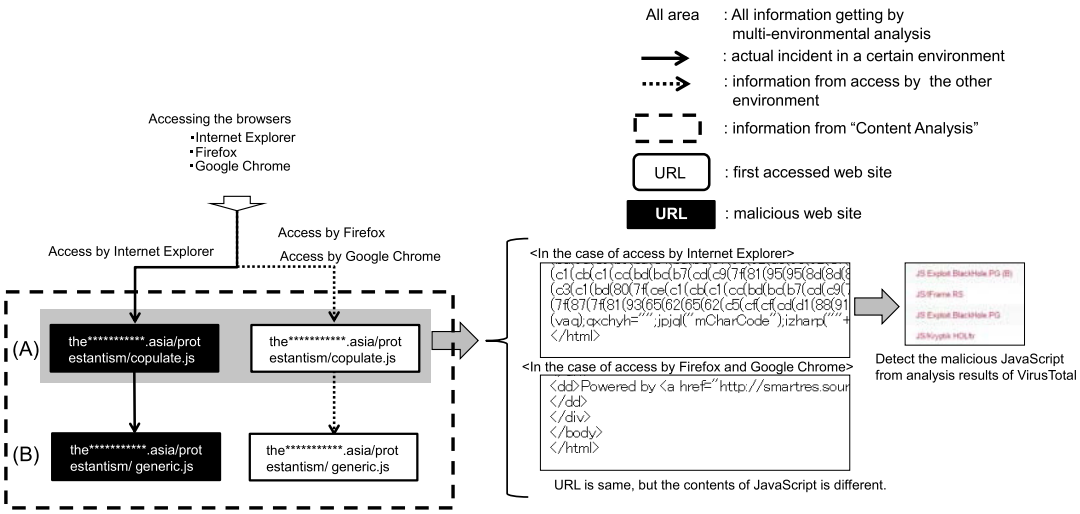

Fig. 5 Multi-environment analysis results of case 3.

to (C) http://g*****.com:12601/ezsqliteadmin/article/module.php?down=82 with iframe tags. Then, there was no response of request for URLs redirecting the clients in (B) and (C). At the visiting by Google Chrome, there was no response from the above web sites (A). This HTTP status code was 404 Not Found.

In this case, (B) and (C) have already been removed. Almost two months passed after the URL is registered to the blacklist, so the web sites have already stopped or disappeared.

As a result of analyzing URLs of (B) and (C) by VirusTotal [16], (B) is judged to be malicious from 4 out of 48 antivirus software, and (C) is judged to be malicious from 4 out of 48 antivirus software. Based on the facts, the client should be redirected malicious URLs and be attacked if (B) and (C) exist. In this case, we can understand how the attack progress in the organization is revealed to run URLs (A), (B) and (C) by comparing the result with the communication logs. In addition, we can find the quiet incident by identifying the existence of the other machines previously accessed (A) or (C) based on retrieving the logs in the organization, if there is an access the same sites in Firefox.

For example, even if the organization handles an incident occurred by accessing (A) via Internet Explorer, there is a risk of other incidents by accessing (A) via Firefox on the other machines. We can get the information on possible incidents and the impact in the organization after analyzing the flow of redirection and the downloaded contents by the results of multi-environment analysis. It is concluded that we can support to clarify preventive measures (e.g., preventing machines from accessing (A), updating targeted application and stopping its use) in order to avoid the risk of expected incidents.

4.2.3 Case 3

We show the results of analyzing the other malicious web site. The site led clients to malicious web sites when the client visits by Internet Explorer, and the site shows a benign behavior when the client visits by Firefox or Google Chrome. We analyzed (A) the**********.asia/protestantism/copulate.js. The result is shown in Fig. 5. Visiting by any browser, the client was redirected to (B) the**********.asia/protestantism/generic.js.

First, we analyze each web page by the content analysis. Nevertheless the same URL is visited by every browser, the downloaded content copulate.js has different body of html file. An obfuscated JavaScript code was embedded in copulate.js accessed by Internet Explorer. On the other hands, no obfuscated JavaScript code was embedded in copulate.js that visited by Firefox and Google Chrome. Next, when we analyzed these scripts with VirusTotal, 23 out of 48 antivirus software judged that the JavaScript code collected by Internet Explorer is malicious. Another JavaScript code was not judged to be malicious by every antivirus software. Similar to the cases 1 and 2, we can understand how the attack progresses in the organization by comparing the result with the communication logs in the organization. In addition, it is found that machines in the organization visited (A) and (B), but we cannot understand whether downloaded script is malicious to just check the URL by comparing the result with the communication logs in the victim organization. Using our multi-environment analysis, we can presume that the URL is probably malicious when malicious script is downloaded in some analysis environment. We can grasp the incident that antivirus software did not detect from the result of getting different files. In this case, only the JavaScript code downloaded by Internet Explorer was judged to be malicious by VirusTotal. It implies that we should preferentially deal with the machine that used to visit the site by Internet Explorer.

Next, we analyze the JavaScript code collected by Internet Explorer with Wepawet and Jsunpack. However, we failed to decrypt the obfuscation of JavaScript, so we did not identify redirected URLs and exploited vulnerability. In order to evaluate the impact in detail, it is desired to develop techniques for deobfuscating JavaScript codes or analyzing codes directly. Furthermore, it is important to study techniques for estimating Exploit Kit from features of character strings of URLs progresses.

5. Related Works

There are some studies of malicious web site analysis in multi-environment. Lu et al. [3] provided a high interactive web client honeypot whose detecting method avoids relying on specific attacks. Also, they investigated which application should be combined to analyze the malicious web sites. On the other hand, in this paper, we have shown how the results of multi-environment analysis are utilized for forensic analysis and risk hedge, and we have reported the actual cases of analyzing malicious web sites.

URL Blacklist is one of techniques for detecting malicious URLs. However, malicious URLs disappear and change in a short period of time, so it is difficult to detect malicious web sites by the blacklist. Therefore, effective methods to discover unknown malicious URLs were provided. Stokes et al. [17] provided methods for discovering landing pages that begin at attack by tracing hyperlink of web sites finally distributing malware in DBD attacks. Invernizzi et al. [18] provided methods for discovering unknown malicious URLs by taking advantage of the information from hyperlink, URL structure, SEO, domain registered data, and DNS query data. On the other hand, in this paper, we aim at impact evaluations of accessing malicious web sites in the organization by analyzing not only benign and malicious judgment of URL but also HTML files and contents of the corresponding web pages.

There are studies of analyzing script that redirect client to malicious web sites and exploit vulnerability. Some methods of JavaScript analysis were provided [19]–[21]. Kamizono et al. [19] focused attention on malicious polymorphic obfuscated JavaScript that is automatically generated and provide methods of treating abstract syntax trees as characteristic points. Gregory et al. [20] provided methods for classifying the obfuscation into malicious obfuscation and benign obfuscation by learning frequently-appearing tree structures with abstract syntax trees. Rieck et al. [21] provided automatic detection and protection system against DBD attacks. This system analyzes malicious patterns of JavaScript with machine learning and blocks delivering malicious JavaScript codes. The purpose of the above methods is detecting and protecting malicious JavaScript to judge whether one is malicious. For example, by using analysis tree and learning data generated by methods [19], [20] as signatures, our multi-environment analysis system can detect and classify JavaScript if it is obfuscated by same algorithm. It implies that these methods are utilized for forensic analysis and risk hedge. One of the purposes in this paper is for identifying redirected URLs and exploited vulnerabilities by analyzing with multi-environment analysis.

6. Conclusion

In this paper, we have proposed the multi-environment analysis system for evaluating the impact of malicious web sites. In our multi-environment analysis, browsers and plug-ins or their versions are changed in order to respond to malicious web sites changing their behavior. We have evaluated the impact of malicious web sites by traffic log analysis, content analysis and malware communication and behavior analysis for characteristic three cases of behavior of web sites posted on Malware Domain List. As the results, the impact evaluation can support for forensic analysis and risk hedge. In a sense, the evaluation results are local because the seeds of the case studies are obtained from Malware Domain List. If the seeds are changed in accordance with the actual situation of the organization, we can get different results.

To future work, we will utilize techniques such as decrypting obfuscated JavaScript codes to analyze more elaborate impact or developing technique of structuring dynamically analysis environment to make malicious web sites analysis fitted actual environment in the organization easy.

References

[1] N. Provos, P. Mavrommatis, M.A. Rajab, and F. Monrose, “All Your iFRAMEs Point to Us,” Proc. 17th USENIX Security Symposium,

pp.1–15, 2008.
[2] Sophos, "Exploring the Blackhole exploit kit," http://nakedsecurity.sophos.com/exploring-the-blackhole-exploit-kit/, 2012.
[3] L. Lu, V. Yegneswaran, P. Porras, and W. Lee, "BLADE: An Attack-Agnostic Approach for Preventing Drive-By Malware Infections," Proc. the 17th ACM Conference (ACM CCS'2010), pp.440–450, 2010.
[4] Y.M. 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," Proc. Network and Distributed Systems Security Symposium, pp.35–49, 2006.
[5] T. Yoshinori, M. Kamizono, M. Hirotomo, M. Mohri, and Y. Shiraishi, "Multi-Environment Analysis for Detecting Malicious Web Sites Changing Their Behavior," Computer Security Symposium 2013 (CSS2013), 2B2-2, 2013 (in Japanese).
[6] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, N. Provos, M.Z. Rafique, M.A. Rajab, C. Rossow, K. Thomas, V. Paxson, S. Savage, and G.M. Voelker, "Manufacturing Compromise: The Emergence of Exploit-as-a-Service," Proc. the 19th ACM Conference on Computer and Communications Security, pp.821–832, 2012.
[7] McAfee, "McAfee Lab Threats Report: First Quarter 2013," http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2013.pdf, 2013.
[8] Dell SonicWALL, "Blackhole Exploit Kit: Rise & Evolution," http://software.sonicwall.com/gav/Blackhole%20Exploit%20Kit%20-%20Rise%20&%20Evolution.pdf, 2011.
[9] The University of California, "Wepawet," http://wepawet.iseclab.org/, accessed Jan. 11, 2014.
[10] Google Project Hosting, "Jsunpack," http://jsunpack.jeek.org/, accessed Jan. 11, 2014.
[11] International Secure Systems Lab, "Anubis," http://anubis.iseclab.org/, accessed Jan. 11, 2014.
[12] ThreatExpert, "ThreatExpert," http://www.threatexpert.com/, accessed Jan. 11, 2014.
[13] Open QA, "SeleniumHQ Browser Automation," http://docs.seleniumhq.org/about/, accessed Jan. 11, 2014.
[14] Malware Domain List, "Malware Domain List," http://www.malwaredomainlist.com/, accessed Jan. 11, 2014.
[15] Riverbed Technology, "Wireshark," http://www.wireshark.org/, accessed Jan. 11, 2014.
[16] VirusTotal, "VirusTotal," https://www.virustotal.com, accessed Jan. 11, 2014.
[17] J.W. Stokes, R. Andersen, C. Seifert, and K. Chellapilla, "WebCop: Locating Neighborhoods of Malware on the Web," Proc. 4th USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2010.
[18] L. Invernizzi, P.M. Comparetti, S. Benvenuti, C. Kruegel, M. Cova, and G. Vigna, "EvilSeed: A Guided Approach to Finding Malicious Web Pages," Proc. IEEE Symposium on Security and Privacy, 2012.
[19] M. Kamizono, M. Nishida, E. Kojima, and Y. Hoshizawa, "Categorizing Hostile JavaScript Using Abstract Syntax Tree Analysis," IPSJ Journal, vol.54, no.1, pp.349–356, 2013 (in Japanese).
[20] G. Blanc, M. Akiyama, D. Miyamoto, and Y. Kadobayashi, "Identifying characteristic syntactic structures in obfuscated scripts by subtree matching," MWS2011, 2A4-3, 2011 (in Japanese).
[21] K. Rieck, T. Krueger, and A. Dewald, "Cujo: Efficient Detection and Prevention of Drive-by-Download Attacks," Proc. 26th Annual Computer Security Applications Conference (ACSAC' 2010), pp.31–39, 2010.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D. degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kindai University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IPSJ.

Masaki Kamizono received his B.E. and M.E. degrees in Computer Engineering from the University of Tokushima in 2003 and 2005, respectively. He is currently a researcher at PwC Cyber Services LLC, Japan. His research interests include malware dynamic analysis, malware static analysis, and malicious web site detection and analysis technology. He received the Best Paper Award at the 2010, 2011 anti-Malware engineering WorkShop (MWS 2010, 2011). He has also conducted research presentations on security technology at international conferences such as AVAR.

Masanori Hirotomo received the B.E., M.E. and Ph.D. degrees from the University of Tokushima, Japan, in 2000, 2002, and 2006 respectively. From 2005 to 2006 he was a Research Associate at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. From 2006 to 2008 he was a Researcher at the Hyogo Institute of Information Education Foundation, Japan. From 2008 to 2011 he was an Assistant Professor at the Graduate School of Engineering, Kobe University, Japan. From 2011 to 2013 he was an Assistant Professor at the Computer and Network Center, Saga University, Japan. Since 2013, he has been an Associate Professor at the Graduate School of Science and Engineering, Saga University, Japan. His research interests are in coding theory and information security. He is a member of the IEEE.

Masami Mohri received the B.E. degree in Information Science and the M.E. degree in Information Science from Ehime University, Ehime, Japan, in 1993 and 1995 respectively. And she received the Ph.D. degree in Information Science and Intelligent Systems from the University of Tokushima, Tokushima, Japan, in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa junior college, Japan. From 1998 to 2002 she was a research associate of the Department of Intelligent Systems and Information Science, Faculty of Engineering the University of Tokushima, Japan. From 2003 to 2008 she was a lecturer of the Department of Intelligent Systems and Information Science, at the University of Tokushima, Japan. From 2008 to 2017, she was an associate professor at the Information and Multimedia Center, Gifu University, Japan. Since 2017, she has been an associate professor at the Department of Electrical, Electronic and Computer Engineering, Gifu University, Japan. Her research interests are in coding theory, cryptography and network security. She is a member of the IEEE.