Poster: Web Change Analysis Approach for Malicious Website Detection

SangYong Choi
Cyber Security Research Center
KAIST
Daejeon, Korea
csyong95@gmail.com

HoMook Cho
Cyber Security Research Center
KAIST
Daejeon, Korea
chmook79@kaist.ac.kr

KiMoon Han
Cyber Security Research Center
KAIST
Daejeon, Korea
linuzen@kaist.ac.kr

JaeKyung Park
Cyber Security Research Center
KAIST
Seoul, Korea
wildcur@kaist.ac.kr

I. MOTIVATION

Recently, the growth of the information communication technology based on internet have evolved as important means in various fields such as finance, tele-communication, and electronic customer. However this growth bring up new threats. Specially, malware distribution that exploits a vulnerability in the website known at the Drive-by download attack that is a classification to one of very critical threats as over [1]. In Drive-by download attack, attackers compromise legitimate websites with web attack methods such as SQL Injection and make them malicious websites. They manipulate the websites to covertly redirect visitors to malware distribution networks by adding related JavaScript or HTML tags. To maintain the stealthiest of the attack, attackers use hidden frame (or iframe) and compromised advertisement banner or 3rd party widget [2]. In the attack, once the victim visits a compromised malicious website (landing page), he or she is automatically redirected to a malware distribution page via some hopping pages. In this case, we defined landing pages, hopping pages and malware distribution pages as malicious websites. Various methods to detection of malicious website have been proposed. First, static analysis approach is the method to analyze the contents or metadata which is widely used for distributing malware such as JavaScript functions, character strings, URL of web pages, IP address and location of hosting server [2-4]. Second, dynamic analysis approach is the method to basically utilize the virtual machine environment or web browser emulator [4-5]. In dynamic analysis, they visit websites and check whether any malicious changes occur in the system. However, both detection methods have some limitations. Static analysis approach is more faster then dynamic analysis, but they cannot precisely analyze obfuscated contents. In dynamic analysis, malwares can detect and evade analysis environment. The reason of limitations is both detection methods are to analyze the website already changed by malicious.

In general, The Website changes are classified into 4 type [6].

- ***Structural Changes***: It occurs when some HTML tags have been added or deleted in the web pages.
- ***Content Changes***: It occurs when the content or information of the web page such as Metadata, Flow, Heading, etc. has been added, deleted or updated.
- ***Presentation Changes***: It occurs when the design or appearances of the web page have been changed but the content or the information have not been changed.
- ***Behavioral Changes***: It occurs when the active components such as applets, scripts, etc. have been changed

It is closely related to the *Structural Changes* and *Behavioral Changes* that the legitimate website was changed to the malicious website. Because Drive-by download attackers insert (or change) some objects such as HTML tags, JavaScript or applet into the legitimate website. The common characteristic of website change detection methods is that they use comparison approach of two webpages (earlier downloaded webpage and next downloaded webpage for same website). And they do not analysis the context of webpages. In this case, the difference of various change detection methods is the method of signature generation to detection [7]. Thus, by the common characteristic of website change detection methods is facilitate identify whether change even if the attackers using attack code in a new way. However, suggested website change detection techniques regard dynamic object as one of tags included website such as JavaScript, applet, thus they cannot detect change of the sub-website that have automatically redirected from current website like the case of the Drive-by download attack.

In this paper, we suggest new approach that combines the advantages of dynamic analysis and website change detection for detection of the malicious website

II. DESIGN OF WEB CHANGE ANALYSIS BASED APPROACH

There are two issues for detection of malicious website. First, it must be able to response to the new concealment technique used by attackers such as obfuscation, encryption, etc. dully redirected from the current webpage.

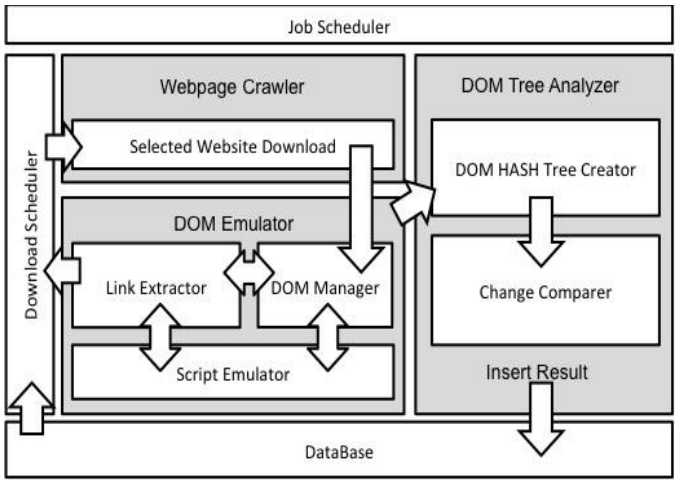

Fig. 1. Architecture of Malicious Website Analyzer.

To solve the first issue, we approach from the surface of the website change detection, thus we can provide that an administrators are able to check unauthorized change without reference to any form of malicious contents that was hidden by attackers. To solve the second issue, we are keep track of the links that are automatically connected using the emulator such as DOM emulator, script emulator.

We have designed the basic approach for possible to solve the two issue. The architecture of new approach based on web change analysis and used dynamic analysis and that include three modules such as crawler, DOM emulator, DOM tree analyzer. The architecture is shown in Fig.1. Detection procedure of malicious website are as follows:

Step1. *The crawler module downloads the contents that is external linked website extracted from monitoring target website or stored in the database by DOM emulator.*

Step2. *The DOM emulator creates a DOM tree from downloaded website and extracts links that automatically connected to other website such as hidden frame (iframe), JavaScript. The DOM emulator emulates the JavaScript. If there exists other link in JavaScript, then extract that and updates the DOM tree.*

Step3. *Insert the extracted links into the download scheduler*

Step4. *Repeat execution from Step1 to Step3 until no longer automatically connected links are exist (link trace)*

Step5. *The DOM tree analyzer creates the HASH tree of each node as equation (1) and compare the current webpage and the previously downloaded webpage.*

$$T_n = \coprod\{N_n, D_n, PN_n, sig0_n, Flag_n, Sig1_n, TN_n, TN_n, TA_n, TC_n\}, n \in N \quad (1)$$

- *N: Node number*
- *D: Node depth*
- *PN: Parent node number*
- *Sig0: Hash of N*
- *Sig1: Hash of N include leaf node*
- *TN: Tag name, such as <HTML>, <Table>, etc.*
- *TA: Tag attribute*
- *TC: Tag contents excepting the text between Tag*

The comparison for detection of changed website is performed two-stage. First stage, the analyzer compare the *sig1* value of two tree. If there exist node that value of *sig1* is same, insert value "S (same)" into flag of the each matched node and the leaf node. Next, insert value "M (modify)" into flag of other nodes. Second stage, among the node that flag is recorded on "M". If there exist the leaf node that flag is recorded on "S", then compare *Sig0* of node in same depth. And if there exist leaf node that value of *Sig0* is same, then current node is not modified but some of leaf node only modified. Thus the analyzer modify flag value of current node as "S". Finally, the nodes that flag is "M" are the changed nodes.

The advantage of this approach is able to response to the new concealed method used by attackers such as obfuscation, encryption because this approach using only HASH value to detection and identification. For the same reason, this approach is not necessary to manage of signature update.

III. Open Questions

In this section, we describe open challenges that will need to be studied in detail to improve the performance of our approach.

- ***The completeness of the analysis environment:*** Recently, among malicious code in malicious website, we found some code that applied the ability to bypass the analysis environment. Thus, we need some advice to make the analysis environment to more elaborate like as the real machine.
- ***The improvement of comparison algorithm:*** The heart of the detection of website change is the analysis algorithm. Thus, requires ongoing study of more accurate and error-free analysis algorithm.

IV. Preliminary Test and Result

We have implemented prototype using DOM emulator based on Spider Monkey [8]. We tested to the malicious websites that was applied the *Structural Changes* and the *Behavioral Changes*. Result of the test, we were able to confirm that the proposed approach is accurately identify the tracking of the link and the parts of changed portion. We are currently performing extensive prototype on various websites for more reliable verification.

V. Conclusion

In this paper, we suggested a new approach for detection of malicious websites. The suggested approach is combined the advantages of two approach. First approach is the dynamic analysis that used to detection of malware distribution site. Second approach is the website change analysis. Through a simple test, we were able to confirm that the suggested approach is effective in the detection of malicious websites. Thus, we expect that the web-administrator can use this approach for monitor of the security of own website. As an ongoing work, we are planning to study for the improvement of analysis algorithm and emulator performance.

References

[1] European Union Agency for Network and Information Security (ENISA): ENISA threat landscape 2014, http://www.enisa.europa.eu-/activities/risk-management/evolving-threat-environment/enisa-threat-landscape/enisa-threat-landscape-2014, Dec 2014

[2] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., & Modadugu, N. "The ghost in the browser analysis of web-based malware," In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 4-4, April 2007

[3] Ma, J., Saul, L. K., Savage, S., & Voelker, G. M., "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," In*Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* , pp. 1245-1254, June 2009

[4] Cova, M., Kruegel, C., & Vigna, G., "Detection and analysis of drive-by-download attacks and malicious JavaScript code," In *Proceedings of the 19th international conference on World wide web*, pp. 281-290, April 2010.

[5] Mavrommatis, N. P. P., & Monrose, M. A. R. F., "All your iframes point to us," In *USENIX Security Symposium*, pp. 1-16, July 2008

[6] Varshney, Naveen Kumar, and Dilip Kumar Sharma., "An enhanced architecture and algorithm for web page change detection," *Information Systems and Computer Networks (ISCON), 2013 International Conference on*. IEEE, pp. 151-154, 2013.

[7] Shobhna, ManojChaudhary"A Survey on Web Page Change Detection System Using Different Approaches"International Journal of Computer Science and Mobile Computing", Vol. 2, Issue. 6, pp. 294 – 299. June 2013

[8] SpiderMonkey, https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey