

Study of Behavior-Based High Speed Visit/Inspection Technology to Detect Malicious Websites

Ji-Sang Kim, Hong-Koo Kang and Hyun-Cheol Jeong

Abstract While the Web provides much convenience and many people all over the world use it almost every day, it is often misused as a medium for distributing malware without users' knowledge. Special care is particularly needed with regard to Websites that are popular with users, since their infection with malware can greatly extend the scope of any damage. Damage caused by malware can be minimized by detecting malicious sites and taking the necessary countermeasures early on. As attack techniques have been evolving, including the abuse of unknown vulnerabilities and the application of detection evasion technology, the advancement of detection technology is urgently required. Leading methods of inspecting the malware concealed in websites include low interaction Web crawling detection, which is fast but dependent upon the signature, and high interaction behavior-based detection, which offers a wide detection range and enables the detection of unknown attacks, although it is somewhat slow. This paper proposes a technology that can visit and quickly inspect large websites to more accurately detect unknown attacks and detection-evading attacks.

J.-S. Kim (✉) · H.-K. Kang · H.-C. Jeong

Team of Security R&D Korea Internet and Security Agency (KISA) Seoul,
Korea IT Venture Tower, Jungdaero 135, Songpa, Seoul 138-950, Korea
e-mail: jisang@kisa.or.kr

H.-K. Kang
e-mail: redball@kisa.or.kr

H.-C. Jeong
e-mail: hcjung@kisa.or.kr

1 Introduction

The technology used to inspect the maliciousness of websites can be categorized into the low interaction method, which uses a Web crawling tool; the high interaction method, which inspects infection by enabling a dynamic visit with a Web browser; and the hybrid method, which inspects a suspicious site using a Web crawler and then visits the site.

Since inspection by a behavior-based dynamic visit does not require a signature and inspects actual infection, it is highly accurate and has a high detection rate. However, the inspection of malware concealed in a website using the behavior-based dynamic visit requires 2–3 min for each website inspection, which includes virtual machine revert and analysis.

Considering the number of websites that are active on the Internet and the number of subpages of each site, the number of URLs to be inspected in Korea alone would amount to millions or even tens of millions.

To realistically inspect so many websites using the high interaction system, the current analysis environment, which requires 2–3 min to inspect each website, would have to be dramatically improved (i.e., an acceleration of 100 times or more).

This paper describes a high-speed website visiting technology that uses the multiplex browser and multi-frame, infection-attempt identification acceleration technology using the process-file-registry correlation analysis, and distributes URI tracking technology to enable such high-speed inspection capability.

2 Related Studies

Open source groups like Honeynet.org have released HoneyPot, a behavior-based malicious website analysis client tool. However, it has the limitation of not being able to analyze multiple websites simultaneously. MS developed Honey monkey, which can inspect malicious sites by running a snapshot before visiting them and then visiting and observing changes in the sites using multiple IE processes. Although it featured a relatively fast inspection performance of 8,000 URLs per machine per day, it still required too much time to inspect large sites. As such, the Honey monkey recommended a method of increasing the detection hit rate by preselecting potentially malicious URLs such as advertising sites as the inspection targets.

As part of its Monkey-Spider project, Mannheim University in Germany developed a system for detecting malware routes and sources using a crawler by organizing a honey pot-type network of malware analysis solutions and vaccines, and then analyzing the contents downloaded through the proxy server from the target website. However, the system still had such problems as duplicated URL analysis and recursive visit error due to the limitations of the open crawler used for

website content download. KISA in Korea is operating the Web crawling-based MC-Finder and a hybrid inspection system, which has enhanced the existing Web crawling technology by enabling it to collect malicious files by dynamically visiting a suspicious URL after scrawling it first.

3 High-Speed Website Visiting Technology

3.1 Website Alive Checking

Of the domains registered in Korea, more than 40 % are reportedly inactive. As such, executing the 'Alive' check of the domain first can minimize unnecessary visits and improve the performance. Such inactive domains can be checked through DNS query transfer and TCP Syn transfer. The procedure is described below.

- ① Sending of DNS query for a high-speed check and checking of the response.
- ② After acceptance of the response to DNS, Syn is sent to TCP port 90 and Ack is checked.
- ③ Assumes that the Web service is provided to the TCP port 80 when an Ack is received.

Since such an inspection method requires fewer CPU and network resources, multi-threads can be used for inspection. The use of multiple threads enables the advance checking of a large number of URLs on the list. (In the test using 17 CPUs, 100 threads were executed to inspect 1.8 million sites in 4 h. As a result, the number of targets to be investigated was reduced from 1.8 million to 1 million.)

3.2 High-Speed Visit Using Multiplex Browser and Multi-Frames

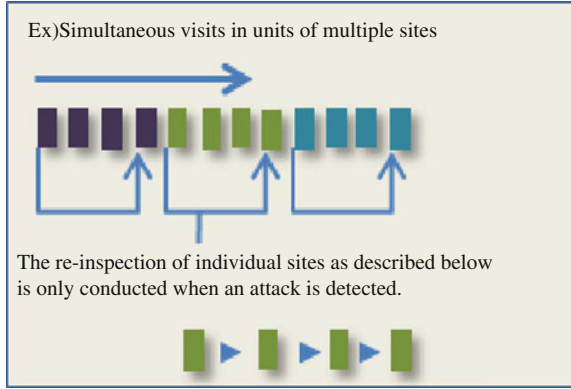
The high-speed inspection method introduced here uses the multiplex browser and multi-frames. It visits multiple websites by opening multiple Web browsers simultaneously. A main page is visited by 30 or more multiplex browsers simultaneously, while a visit to the subpages is accelerated by applying the multiplex browser and multi-frame visit techniques simultaneously.

When using 20 browsers with 5 frames, 100 sites (5×20) can be inspected simultaneously.

Multi-frames are used only to inspect the subpages.

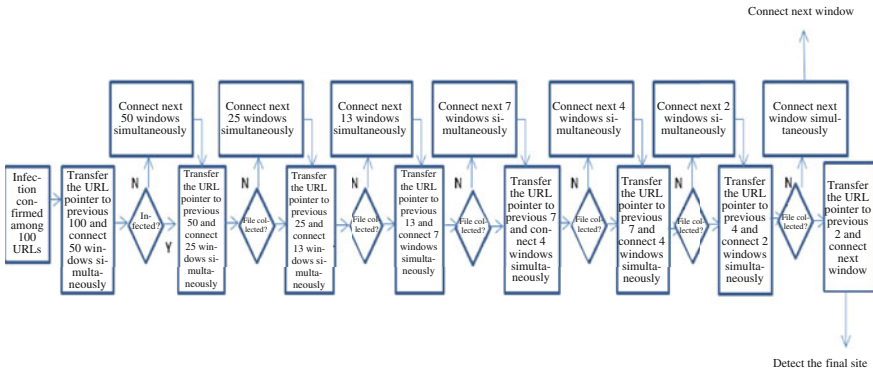
Sites are simultaneously visited using the multiplex browsers and multi-frames. If an infection attempt is not observed, then the net inspection target group is visited. If an infection attempt is confirmed, the suspicious site is tracked.

To track a suspicious site, the following tree method is used to quickly track the site with the minimum number of inspections.



If an infection attempt is detected among the 100 sites simultaneously visited, those sites are revisited in units of 50, i.e., $1/2$ of the original number of sites. If an infection attempt is detected in a unit, then those sites are revisited in units of 25, i.e., $1/2$ of the 50 sites. Inspection and re-inspection are recursively executed. Such a tree algorithm based re-inspection method can be greatly effective as the number of sites simultaneously visited increases. For example, when 100 sites are tracked, a malicious site can be identified in 7 inspections in the best case, 14 inspections in the worst case, and 10 inspections on average.

Compared to sequentially visiting one site at a time (once in the best case, 100 times in the worst case, and 50 times on average), this method can improve the performance fivefold on average.



3.3 Fast Malware Infection Attempt Identification Technology

3.3.1 Identification of an Infection Attempt by Analyzing the Correlation Pair of the Behavior Generated During the Visit

After rapidly inspecting the sites using the multiplex browsers, any vulnerability attack or malware infection attempt in the visited target site must be quickly identified.

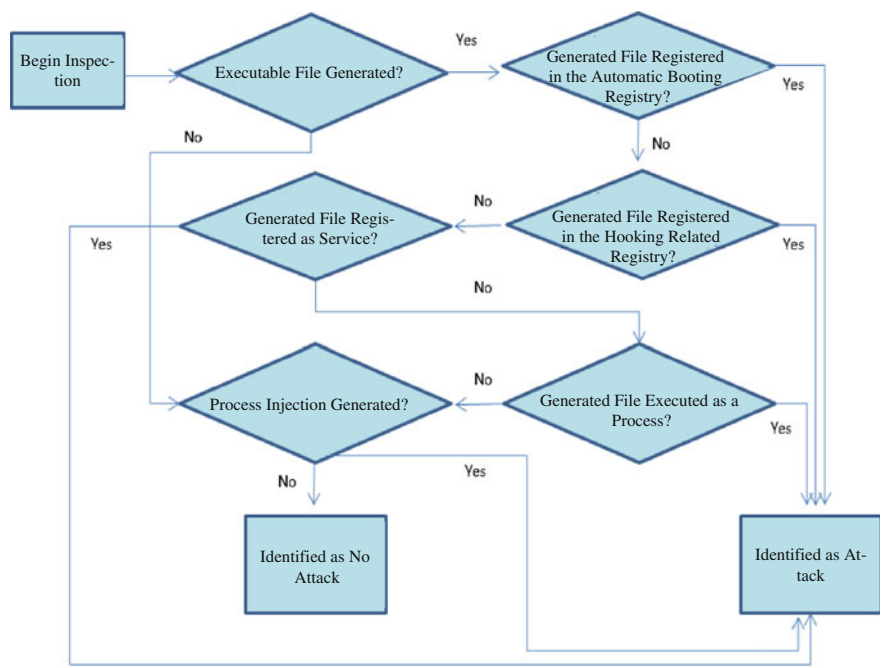
A Web browser limits the behaviors allowed after connecting a Web page to prevent security problems. The identification uses the feature to identify the infection attempt after visiting the website.

For example, one may suspect a malware infection attack if it detects executable file generation, registry registration, or process creation after a visit.

However, such behavior does not always mean malware infection has occurred since various files can be generated and processes loaded into the memory by a normal Website visit also.

Therefore, to correctly identify a malware infection attempt, correlation pair analysis is performed on the files, processes and registry registrations generated after a visit.

In other words, correlation analysis—such as the correlation of file generation and process load of the generated file, correlation of file generation and registry registration of the generated file, etc.—is used to accurately identify an infection attempt. Moreover, since the process injection can be considered as an attack on the vulnerable point, all injection generations are identified as attacks.

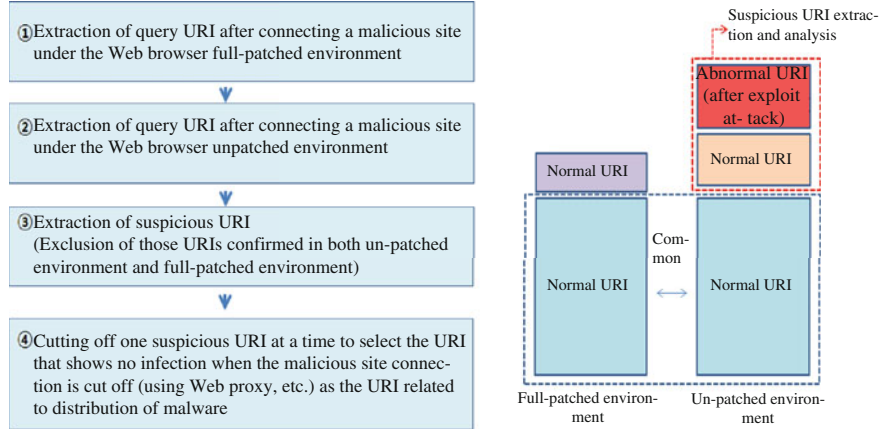


3.3.2 Malicious URI Tracking

When a malicious site is confirmed after high-speed inspection using multiplex browsers, the malicious URI within the malicious site needs to be checked.

Various codes exist in a malicious site, and it is very difficult to separate the attack codes from the normal codes. However, a malicious URI, such as malware distribution after an exploit attack, can be identified with the query session differentiation analysis of the Web browser full-patch environment and the un-patched environment, as shown below.

In the un-patched environment, an additional query such as malware download is generated after the exploit attack has been successfully executed. The detailed



procedure for tracking a URI is described below.

Of the session generated in the un-patched environment, those that cannot be observed in the full-patch environment are selected as suspicious URIs. The site is revisited after cutting off the URIs one at a time and checking the infection. If the infection is not generated after an URI has been cut off, then it is identified as the URI distributing the malware.

4 Performance Test

Tests showed that the environment described above enabled the high-speed behavior-based inspection and detection of many malicious sites. The detailed test results are shown below.

Test Performance

Condition	Performance
Tested system environment	Main page inspection
- CPU: i7v	–25,000 URL/day, 1host
- RAM : 16G	Subpage inspection
- Internet speed: 100 M	–65,000 URL/day, 1host

Domain Inspection Results

Detection system specification	Inspection target	Required period	No. of site domains inspected	No. of detected cases
1 Host - CPU: i7 - RAM: 16G - Internet speed: 100 M	1st inspection of service domains in Korea	48 h per host	More than 130,000	4 malicious site detected
	2nd inspection of service domains in Korea	48 h per host	More than 130,000	4 malicious site detected
	3rd inspection of service domains in Korea	48 h by host	More than 130,000	6 malicious site detected
	4th inspection of service domains in Korea	48 h per host	More than 130,000	0 malicious site detected
	5th inspection of service domains in Korea	48 h Per host	More than 130,000	8 malicious site detected
	6th inspection of service domains in Korea	48 h per host	More than 130,000	12 malicious site detected
	7th inspection of service domains in Korea	48 h per host	More than 130,000	4 malicious site detected

5 Conclusion

The need for high-speed, behavior-based identification technology is increasing in line with the advances made in techniques for concealing Web attacks and the ever increasing number of cases of exploitation of unknown vulnerabilities being reported. The use of multiplex browsers and high-speed identification technology is expected to help cope with malicious websites more effectively by overcoming the limitations of Web crawling to detect more malicious sites more quickly and by supplementing the existing Web crawler systems.

Acknowledgments This research was supported by the Korea Communications Commission (KCC), Korea, under the R&D program supervised by the Korea Communications Agency (KCA)”(KCA-2012-(10912-06001)).

References

1. Jamie R (2008) Server honeypot vs. client honeypot. The HoneyNet project. <http://www.honeynet.org/node/158>. Accessed Aug 2008

2. İkinci A, Holz T, Freiling F (2008) Monkey-spider: detecting malicious websites with low-interaction honeyclients. In: Proceedings of Sicherheit, Schutz und Zuverl, April 2008
3. Wang Y, Beck D, Jiang X, Roussev R, Verbowski C, Chen S, King S (2006) Automated web patrol with strider honeymonkeys: finding web sites that exploit browser vulnerabilities. In: 13th annual network and distributed system security symposium. Internet Society, San Die
4. New Zealand Honeynet Project Capture-HPC—capture—the high interaction client honeypot. <http://www.nz-honeynet.org/capture.html>
5. Kim BI, Cheong JI, Cheong HC Study of search keyword based automatic malware collection system
6. Kim BI Study of automatic collection of malware distributed through SNS. ISSN 1738-611X

IT Convergence and Security 2012

Kim, K.J.; Chung, K.-Y. (Eds.)

2013, XIX, 1244 p. In 2 volumes, not available
separately., Hardcover

ISBN: 978-94-007-5859-9