A STUDY ON SECURE WEBSITES USING MALWARE DETECTION SYSTEM

Yong-joon Lee¹, Ki-hwan Kim^{2*} and Hwa-sung Chae³

¹*Defense Security Institute, 89-10, Seosomun-ro, Jung-gu, Seoul 03742,
Republic of Korea*

²*F1 Security, 952AB Ho, H-1, KIST, Hwarang-no 14-gil 5, Seongbuk-gu, Seoul,
Republic of Korea*

³*IT Nomads Co., Ltd, 603 Daeryung Post Tower 2 Cha, 306, Digital-ro,
Guro-gu, Seoul 08378, Republic of Korea*

¹jlee4279@gmail.com, ²itconsult@f1security.co.kr, ³mars@itnomads.com

Abstract— Trends show that there has been a huge increase in damages from malware that is hidden on websites by hackers to infect the PCs of visitors to the site. Since websites with a lot of visitors are an easy way to spread malware, attackers use these sites to spread their malware as far as possible in order to steal financial information, or to gather information about a specific user as a pre-attack step. In this research, we intend to quickly detect malware and scientifically analyze the hidden purpose of attackers through a malware detection system that uses a website sensor network.

Keywords— Secure Websites, Detection of Malware on Websites, Detection of Forged Update Files

1. INTRODUCTION

Recently, malware related with cybercrimes has been increasingly spreading through websites. In the process of distributing malware through a website, the malicious script hidden in the web source runs when a user's PC that is vulnerable to it accesses the site, infecting the PC with malware [1]. In particular, if an ActiveX distribution server is hacked and malware is spread through the forging of ActiveX, all the PCs of users that connect will be infected with malware, resulting in much damage [2].

The spread of malicious code through websites is causing security threats such as personal information leakage, financial information capture, DDoS attacks using botnets, and intelligent APT attacks [3]. Analysts are constantly researching effective methods of malware detection in order to deal with these threats.

Research on pattern and technology trends of distribution of malicious codes. Through this process, it presents trends of changes in patterns of distribution of malicious code and predicted trends in the future through analysis of patterns of distribution of new malicious codes. Based on collected distribution patterns, it analyzes vulnerability types and patterns and develops detection algorithms and similarity comparison algorithms by distribution patterns of malicious codes. It is going to develop automated modules that analyze patterns of distribution of malicious codes by applying developed algorithms.

In this study, we tested our website sensor network for 12 months using a detection system that integrated three detection techniques: static analysis, dynamic analysis, and

Received: June 4, 2019

Reviewed: August 12, 2019

Accepted: September 5, 2019

* Corresponding Author

update file forgery checks. The purpose of this study is to introduce a system for detecting malware distributed through websites. It is expected that this system will detect malware quickly and scientifically, and grasp the hidden purpose of the attackers[12].

2. SPREADING MALWARE THROUGH A WEB SENSOR NETWORK

2.1. ATTACKS BY HIDING MALWARE ON WEBSITES

Figure 1 is The attacker hacked websites, emails, and update files by using malware through the web sensor network to spread malware and exploit victims[15]. Significantly, websites are becoming a major attack target because they tend to be vulnerable in the development process. Furthermore, because many users access websites, malware can be spread in large quantities [4].

- Step 1 : An attacker attacks a website's vulnerability and attempts to hack it. The attacker conceals a malicious script in the source code of the hacked website, which automatically downloads malware to the user's PC.
- Step 2 : When a user accesses a website, if the user's PC is vulnerable to a malicious script that is concealed on the site by an attacker, the malware is downloaded to the PC without the user's knowledge.
- Step 3 : An attacker steals personal and financial information from a user's PC infected with malware, and exploits it to carry out cyber crimes such as personal information sales and financial fraud [5].

Fig. 1 Attack techniques to hide malware on websites.

When attackers acquire personal information from a user PC infected with malware, additional damage can follow from financial fraud such as voice phishing and pharming. In recent years, as the financial supervisory agencies have strengthened their measures to prevent voice phishing, pharming has been the main method used. Pharming is cybercrime in which malware is used to modify the DNS address of the host file on the user's PC and lead the user to a forged financial website in order to steal the user's financial information. The malicious pharming code first transmits the user's financial information to the attacker. Then, the attacker tampers with the host file of the user's PC to forge a portal site. When the user attempts to access the portal, the forged portal

prompts them to input their financial information through the pop-up window, sometimes even by impersonating an investigation agency [10].

2.2. ATTACKS THROUGH FORGING THE UPDATE FILES ON WEBSITES

To use Internet services such as file hosting services, financial transactions, etc. on the website, it is often necessary to download update files. In one type of attack, an attacker can intensively attack and hack the distribution server of update files and turn the update files into malware[13]. When an update file is tampered with, the user's PC recognizes the malware as a normal file and automatically updates it, resulting in all of the connected PCs being infected with the malware. Even if security patches are used to remove security vulnerabilities from users' PCs, they are infected with malware. For this reason, this type of attack is characterized by a large scale of damage [6].

Fig. 2 is an example of an attack that was carried out through the forgery of update files for file hosting services in the 6.25 Cyber Attack by North Korea in 2013.

- Step 1 : The attacker intensively hacked two file hosting service sites with many visitors, and forged the sites by inserting malware into the file hosting service update files.
- Step 2 : The falsified update files were automatically downloaded to the user's PCs connected to the file hosting service and infected a large number of PCs
- Step 3 : The attacker executed a DNS DDoS attack on a malware-infected PC connected to a DNS server of the National Integrated Computing Center, thereby blocking access to a number of government websites.

Fig. 2 Forgery of website update files

The attacker hacked file hosting service sites with many visitors, and automatically infected all the visitors with malware by forging the update files. It was a mutated cyber attack, in which infected PCs performed DoDOS attacks on a DNS server of a government agency [11].

3. MALICIOUS CODE DETECTION TECHNIQUES THAT USE THE WEBSITE SENSOR NETWORK

3.1.DETECTING MALWARE THROUGH THE WEBSITE SENSOR NETWORK

Static analysis of the website sensor network involves collecting a website to be checked by using a crawler, which is a collection tool, and checking the website source code for malicious scripts using anomaly detection patterns[16]. The anomaly detection patterns check the website sources for two types of anomalies, which are website links for spreading malware and malicious scripts for attacking a PC's vulnerabilities. The detection patterns were derived from two sources: websites that spread malware by using the information provided by security companies, and malicious scripts that we analyzed for our own research. Attackers typically have a high level of technical skill, and can partially modify and distribute existing malware to bypass security products such as vaccines and firewalls. For this reason, it is important to collect and analyze as many cases of malware as possible and register them as detection patterns in order to detect them when they are concealed in the website sensor network [7]. The static analysis method detects patterns by comparing character strings. The advantage of this method is that it can detect patterns quickly, but the downside is that it cannot detect new, unregistered malware[14].

Dynamic checks of the website sensor network is a method of detecting abnormal changes in a PC through a virtualization method when the user's PC accesses a website [8, 9]. The dynamic checks automatically virtualize various environments of operating systems, and automatically connect to the target website. If malware is hidden on the connected website, the registry of the virtual PC changes and downloads the malware file, and then attempts to access a specific IP. Dynamic checks have the advantage of being able to detect variants and new malware; however, it is a slow process compared with static checks [10, 11].

Fig. 3 Architecture of the website sensor network for detecting malware

Therefore, in this study, an analysis was performed by using both the static and dynamic checks to detect hidden malware through inspection on the website sensor network. The architecture of the website sensor network for detecting malware is as shown in Fig. 3. The websites to be checked are registered in the target domain DB and checked through both static and dynamic analysis, as follows:

In the static analysis (Signature Based Detection), collect the websites with a web crawler and check them with 200,000 detection patterns in the Malware Signature DB. In the dynamic analysis (Behavior Based Detection), access the website with a virtual PC, and check for any malware downloaded, changes in the registry, and network access requests in order to detect malfunctions. Classify malware into variant and new by using dynamic analysis, and register the cases with the list of detection patterns for use in the static analysis. Store the information of the websites that spread malware in the Malware Website DB through static and dynamic checks, and request the website operator to delete the malware.

3.2. WEBSITE SENSOR NETWORK CHECKING FOR UPDATE FILE TO DETECT FORGERY

If a website is hacked and the update file is forged, all user PCs that download the file will be infected with malware. For this reason, real-time checking is essential for forgery detection.

Fig. 4 Architecture to check for update file forgery

As shown in Figure 4, the website sensor network detects update file forgery in the following manner: The website operator registers the update file on the inspection server before distributing it to the users. It is important for the website operator to register the update file, and then distribute it to the users. Forgery detection of update files is performed in real time by comparing the hash value of the update file on the website with the registered hash value.

3.3. FEASIBILITY STUDY ON TECHNIQUES TO DETECT HIDDEN MALWARE THROUGH THE WEBSITE SENSOR NETWORK

Table I suggests a countermeasure to the spreading of malware that can occur in the website sensor network as follows: Malicious code hidden on a website can be detected quickly by performing both static and dynamic analysis in parallel. In addition, forgery in the update file can be detected through real-time comparison of the file to be distributed with the pre-registered file. Detection of the website sensor network by using these three methods enables multi-level detection of malware spread by attackers.

Table I. Multi-level Detection Techniques of the Website Sensor Network

Targets of detection	Detection techniques	Main functions	Limitations
Website With Hidden Malicious code	Static analysis	<ul style="list-style-type: none"> Collect website sources through Crawling to check for malware with the detection patterns. High-speed detection can be performed by comparing strings with the detection patterns. 	<ul style="list-style-type: none"> New and variant malware not registered in the list of detected patterns cannot be recognized
	Dynamic analysis	<ul style="list-style-type: none"> Detect unusual activity inside the PC by connecting a virtual PC to the website Any new and variant malware can be detected by checking the abnormalities in the PC without using the detection patterns. 	<ul style="list-style-type: none"> Due to the nature of the virtualized environment and monitoring of abnormal behavior, the detection time is increased
Web services Update file Forgery	Hash value Comparison	<ul style="list-style-type: none"> Website operator can check for forgery by comparing the hash value of the update file with the registered hash value in the inspection server before distributing it. Real-time inspection allows rapid detection 	<ul style="list-style-type: none"> This method requires a procedure through which the website operator registers the update file with the inspection server before distributing it.

4. EXPERIMENT TO DETECT HIDDEN MALWARE THROUGH THE WEBSITE SENSOR NETWORK

4.1. EXPERIMENTAL ENVIRONMENT

As shown in Table II, we used the experimental environment to detect malware hidden on the proposed website sensor network as follows: The targets we monitored for 12 months in 2017 include: 2,500,000 websites for static analysis, 500,000 websites for dynamic analysis, and 50 web security services for forgery of update files.

Table II. Experimental Environment

Targets of detection	Detection techniques	Targets to check	Check cycle
Website Hidden malware	Static analysis	• 2,500,000 websites	• Used the check frequency of once a day
	Dynamic analysis	• 500,000 websites with high user access	• Check once a day with a virtual PC
Update file Forgery	Hash comparison	• 50 file hosting services	• Five-minute check

4.2. Experiment Results

As shown in Table III and Figure 5, the total number of cases of malware spread on the website in 2017 was 13,347; these originated from 1,197 websites and were spread via 12,150 intermediate websites. In June and July, when a large number of cases were detected, many intermediate malware sites occurred due to the bulletin board of a specific site, and in December many of the past intermediate sites were redetected again. In particular, in the third quarter (July to September), a lot of spreading sites were detected due to an increased occurrence of Ransomware.

Table III. Results of checks on malware hidden on websites

Classification	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Total
Intermediate sites	115	801	418	360	584	4,754	1,850	667	283	237	245	1,836	12,150
Spreading sites	71	48	116	77	70	52	62	170	298	136	57	40	1,197

Fig. 5 Results of Checks on Malware Hidden on Websites

Of the malicious scripts hidden in the malware spreading sites, 130 cases were automated by attack toolkits, as shown in Table IV. A typical toolkit used for attacks is CKVip. These automated attack toolkits are used for web-based attacks that exploit vulnerabilities in web browsers and plug-ins, and are used to hide malware on websites. Attack toolkits, which mainly originate from China and Russia, can easily create malicious scripts that attack vulnerabilities of various programs such as Java applets, Flash players, and Internet Explorer.

Table IV. Hacker's Attack Toolkits used to Spread Malware to Websites

Classification	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Total
Gongda								1					1
CK Vip	14	2	24	23	7	12	9	6	5	4	22	1	129

Table V and Figure 6 show the S/W vulnerabilities used for malware attacks in 2017. The severity of the vulnerabilities can be ranked in the following order: Adobe Flash Player, Java Applet, and MS IE. They attack the vulnerabilities of various programs instead of individual ones by using an automated attack toolkit to increase the success rate of malware infection on a user's PC. Direct downloads, other than attacks on S/W vulnerability, represented 37%.

Table V. PC Vulnerabilities used to Spread Malware to Websites

Classification	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Total
Adobe Flash Player	48		48	44	11	24	24	8	16	16	11	4	254
Java Applet	36		36	33	9	18	18	6	12	12	9	3	192
MS IE	23	3	32	19	4	12	12	4	8	8	6	2	133
MS Edge	9		12	11	2	6	6	2	4	4	3	1	60
MS OLE	1	1			1	1					1		5
MS Silverlight					1								1
Direct download	11	9	42	29	23	24	30	56	84	33	12	22	375
Total	128	13	170	136	51	85	90	76	124	73	42	32	1,020

Fig. 6 PC Vulnerabilities used to Spread Malware on Websites

Table VI and Figure 7 shows the malware incidents by type detected in 2017. Types of malware attacks can be ranked in the following order: information leakage (financial information) 33%, ransomware 18%, downloader 10%, dropper 8%, information leakage (account information) 7%, backdoor 7% and virtual currency mining 5%. Information leakage (financial information), which is a combination of financial information leakage and phishing, declined in the second half of the year. Ransomware was continuously detected in 2016 and 2017, and virtual currency mining malware newly appeared with the value of virtual currency increasing.

Table VI. Website Malware Types

Classification	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Total
Information leakage (financial information)	18	2	40	19	9	16	10	12	14	5	2	1	148
Ransomware	1		1	1	2	5	3	24	29	14	2	1	83
Downloader		3	6	6	2	2	4	2	5	4	5	6	45
Dropper		2	2	5	4	2	3	4	8	3	3	1	37
Information leakage (account information, etc.)			1	1	1		3	4	8	2		11	31
Backdoor			1	2	4	3	2	3	11	2		2	30
Virtual currency mining				1			5	2	5	6	3	1	23
DDoS			5		1		3	3	3				15
Remote control	2	2	1	3	2		1						11
Key logging		1		1				4	4	1			11
Information leakage (mobile device)	1	2	1		1								5
Other	1			2		3	2		1		1		10
Total	23	12	58	41	26	31	36	58	88	37	16	23	449

Fig. 7 Malicious Code Type of Websites

Distributing malware directly by forging file distribution software updates or free software update files is an effective way of spreading malware in a short time, and security management of the distribution site is particularly important. Table VII shows the status of update file forgery of file distribution software updates and free software.

Table VII. Status of Update File Forgery Checks

Classification	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Total
File distribution service	6	-	-	2	-	6	4	-	2	7	2	1	30
Free software	-	-	-	-	1	-	-	-	-	-	-	-	1
Total	6	-	-	2	1	6	4	-	2	7	2	1	31

4.3. ANALYSIS OF RESULTS OF EXPERIMENT

We can confirm the following characteristics through efforts to detect malware spreading through the website sensor network that were carried out for 12 months. First, attackers are making “watering hole attacks” on websites where they can access a certain user type. They also aim at websites with a lot of visitors or web hosting servers that manage multiple websites in order to effectively spread malware to many users. The purpose of spreading malware is to steal financial information and use it for cyber crimes.

- Watering hole attack

Watering hole attacks try to identify the characteristics of users accessing websites and hide malware with a clear goal. It has been confirmed that the attackers hide malware on the websites of medical associations, labor unions, and other organizations to plunder personal information from medical workers, members of labor unions and others.

- Websites with a lot of visitors

It is considered that attackers intend to spread as much malware as possible by hacking websites with high user access, such as weather forecasts, news media, online petitions, advertisement events, etc.

- Web hosting service

The attackers made focused attacks on small to medium-sized web hosting servers with weak security. The reason for this approach is that by hacking a web hosting server, they can hide their malware on all the internal websites hosted there and use them as an effective path.

- Malicious code for seizing financial information

148 pieces of malware detected in this experiment were designed to steal financial information, accounting for 33% of all 449 malware cases. Significantly, malware related to Ransomware and cryptography mining newly appeared.

5. CONCLUSION

Attacks on website sensor networks are increasing, as they can be used as a route to infect users' PCs with malware. The attackers hack the website to hide malicious scripts on the website, avoiding the administrator's notice, or to forge the update file. In this way, they spread malware to the PCs of users connected to the website.

In this study, a website was tested for 12 months with a detection system that integrated three detection techniques: static analysis, dynamic analysis, and update file forgery check. Experiments have shown that attackers mainly use attack toolkits and spread malware to steal financial information. In addition, attackers use a watering hole attack to target a specific group in order to effectively spread malware. Their major victims also include websites with a lot of visitors and web hosting servers where a lot of websites are concentrated, in order to spread as much malware as possible. The purpose of this study is to introduce a system for detecting malware distributed through websites. It is expected that this system will detect malware quickly and scientifically, and grasp the hidden purpose of the attackers.

REFERENCES

- [1] K. Thomas, and D. M. Nicol, "The Koobface Botnet and the Rice of Social Malware", IEEE Int. Conf. Malicious and Unwanted Software, (2010): 63-70.
- [2] Boo Joong Kang, Kyoung Soo Han, Eul Gyu Im, "Malware Current Status and Detection Technology", Communications of the Korea Information Science Society, vol. 30, no. 1. (2012): 44-53.

- [3] K. H. Lee and J. P. Park, "A Software Vulnerability Analysis System using Learning for Source Code Weakness History", Journal of The Korea Academia Industrial cooperation Society, 18 (2017), 46-52. <https://doi.org/10.5762/KAIS.2017.18.11.46>.
- [4] JooBeom Yun, YoungJoo Shin, "MiGuard: Detecting and Guarding against Malicious Iframe through API Hooking", IEICE Electronics Express. (2011): 460-465.
- [5] M. Fredrikson, S. Jha, M. Christodorescu, R. Sailer, and X. Yan, "Synthesizing Near-Optimal Malware Specifications from Suspicious Behaviors", Proceedings of the 2010 IEEE Symposium on Security and Privacy. (2010): 45-60.
- [6] Q. Miao, Y. Wang, Y. Cao, X. Zhang, and Z. Liu, "APICapture - a Tool for Monitoring the Behavior of Malware", Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering. (2010): 390-394.
- [7] S. Y. Min, C. S. Jung, K. H. Lee, E. S. Cho, T. B. Yoon and S. H. You, "Design of Comprehensive Security Vulnerability Analysis System through Efficient Inspection Method according to Necessity of Upgrading System Vulnerability", Journal of The Korea Academia Industrial cooperation Society, 18 (2017):1-8. <https://doi.org/10.5762/KAIS.2017.18.7.1>.
- [8] J. Lee, K. Jeong, and H. Lee, "Detecting Metamorphic Malwares using Code Graphs", Proceedings of the 2010 ACM Symposium on Applied Computing. (2010): 1970-1977.
- [9] K. Kim, and B. Moon, "Malware Detection based on Dependency Graph using Hybrid Genetic Algorithm", Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. (2010): 1211-1218.
- [10] S. H. Oh, T. E. Kim and H. K. Kim, "Technology Analysis on Automatic Detection and Defense of SW Vulnerabilities", Journal of The Korea Academia Industrial cooperation Society, 18 (2017), 94-103. <https://doi.org/10.5762/KAIS.2017.18.11.94>.
- [11] Korea Information & Security Agency(KISA), October trends: Cyber Security Issue, (2013): 31-42.
- [12] Korea Information & Security Agency(KISA), Recent Cyber Attack Trend, Information Sharing Seminar Report, (2014):10-18.
- [13] Korea Information & Security Agency(KISA), Large-scale malicious code distribution trend analysis report, (2014):1-15.
- [14] Asish Vardhan and M. Murali Krishna, "DGD: An Intrusion Detection System for Providing Security to Web Applications" Journal of Statistical Computing and Algorithm 1.1(2017): 27-32. <http://dx.doi.org/10.21742/JSCA.2017.1.1.03>.
- [15] G Sai Asritha, "Security for files in Operating System" International Journal of Security Technology for Smart Device 3.2(2016): 1-6. <http://dx.doi.org/10.21742/IJSTSD.2016.3.2.01>.
- [16] Lalit Mohan Joshi, and Dr. Rajendra Bharti, "A Research Paper on Online Security Issue with Data Tempering" International Journal of Security Technology for Smart Device 4.1(2017): 1-8. <http://dx.doi.org/10.21742/IJSTSD.2017.4.1.01>.
- [17] Vempati Sudheshna, "Adhoc Network Infrastructure Security" International Journal of Wireless and Mobile Communication for Industrial Systems 3.1(2016): 21-34. <http://dx.doi.org/10.21742/IJWMCIS.2016.3.1.03>.
- [18] SeongMuk, Choi, JoongHyo, Bok, HyungTaek, Lee, Jong Bae, Kim and Gwang Yong, Gim, "An Empirical Study on Crawler-based Security Control Systems" International Journal of Reliable Information and Assurance 5.2(2017): 7-12. <http://dx.doi.org/10.21742/IJRIA.2017.5.2.02>.