International Journal of Grid and Distributed Computing
Vol. 11, No. 6 (2018), pp.27-36
http://dx.doi.org/10.14257/ijgdc.2018.11.6.03

A Study on the Crawler-based Security Model for Improving Modulation Monitoring of Websites

Seong Muk Choi[1], Joong Hyo Bok[2], Hyung Taek Lee[3] and Gwang Yong Gim[4]

[1,2,3]*Department of IT Policy, Management Soongsil University, Republic of Korea*
[4]*Department of Business Administration, Soongsil University, Republic of Korea*
[1]*csm0107@gmail.com*, [2]*jhbok@kisvan.co.kr*, [3]*htlee@innotium.com*,
[4]*gygim@ssu.ac.kr*

Abstract

The main purpose of this study is to propose a security control system model, which enables to strictly monitor the web for security and detect the damages caused by malicious code when using random devices for transits and disposals. This model is significantly different in comparison to other existing monitoring systems. It connects the crawler not just to the main-URL but further extends its connection to the sub-URL, enabling it to sequentially detect vicious links and the status of the website. Moreover, the model includes an algorithm that enables the detection of malicious code in an internet-downloaded file through analyzing its behavioral patterns.

To validate this model, this study has developed and implemented the core algorithm to a prototype, conducting various experiments. As a result, the new module which anticipated in this study was considered as a representative factor to monitor and detect the malfunctions of the website. Furthermore, the website's security could be effectively identified utilizing the proposed modules.

Keywords: *Crawler Security Model, Website protection, Hack detection*

1. Introduction

The recent developments of Internet-based services in the 21st century has evolved in the high-speed Internet digital Informa ionization, which conveniently utilizes online web-based services throughout the Korean society including civil affairs, banking, shopping, VOD, VOIP. However, just as there are shadows in the sun, the dysfunctions of information are also occurring in a variety of ways. The major adverse effect of information development include Cyberattacks continuously assaulting homepages.

In accordance with Article 2 of the National Cyber Safety Management Regulation (Presidential Order No. 316), national public institutions have established comprehensive cyber security control centers operated by the local and central governments. The organizations operate 24 hours every day throughout the year, continuously monitoring the web of comprehensive cyber threats. Each organization monitors at least a dozen to hundreds of web sites, and various security control measures such as a harmful site detection system, a log collection system, and a malicious code detection system have been established and operated. However, most security controls are built and operated as a network traffic-based signature detection systems, and the control of the website is being monitored by building an early version of the website's tempering/forgery detection system that checks only the web status, such as the disconnection or delay of the homepage as well as the change rate in web sources by the established cycle. This control system only conducts fragmentary status, and it is difficult to immediately detect the damage caused by web hacking attacks due to the zero-day attacks.

Received (January 5, 2018), Review Result (March 9, 2018), Accepted (March 12, 2018)

ISSN: 2005-4262 IJGDC
Copyright ⓒ 2018 SERSC Australia

In this study, we examine cyber-attacks' types and malicious codes' distribution on the website as well as design a new crawler-based security control model that is available for the security control center to confirm not only the excellence of existing model but also the proposed model in the real control environment.

2. Related Researches

2.1. Web Site Hacking Attacks

The generic term 'malicious code' indicates an executable code written for malicious purposes. It is also classified as a worm, a virus or a Trojan horse depending on its ability to duplicate itself and contamination target's presence. One of the most common malicious code propagation methods which causes multiple victims is called a drive-by-download attack, that is a circulation method through a web page. This attack spreads more easily than malicious codes which are mainly infected by clicking attached files via email. This is true because it is infected only by visiting a website and because it is difficult to confirm the path.

Drive-by download attacks are hard for discovering as well as analyzing as a network security system due to the insertion of a script that hacks a web site and directs it to an infected site. unlike the method of uploading malicious code to a website in the past. Users who access the website are infected with malicious codes due to drive-by download attacks when they use the browser or plug-in containing the vulnerability.

In some cases, malicious code or script is executed on the site simply by accessing the website, and a method of infecting the user is used by moving the malicious code infected in various stages to various stop places and the spreader. In general, the first stopover area is abused as the first stopover point for a large number of visitors, a large portal with social purpose, or a homepage of a government agency.

The first penetration route of attacks such as insertion of tempering/forgery code, malicious codes, *etc.*, is mainly caused by web-shell upload through file upload, remote access through account leaking, SQL injection, and so on. Attackers who have succeeded in initial attack and penetration perform additional attacks such as web shell, Rootkit, and Daemon tampering, and the most frequent attacks are attacking in the form of using a combination of various system commands by installing a web shell on the attack target server.

2.2. Web Site Forgery/Tampering Detection System

The website modulation and monitoring system which is being utilized in most of security control centers is a system to analyze only the simple comparison of data which is collected by crawler's fragmentarily registering only showing homepage of several tens to as many as several hundred websites and accessing domains consecutively. Website anomaly behavior detection is a forgery/tempering check function which includes simple source collection, custom keyword analysis, similarity analysis of string comparison, state analysis of error/delay/disconnect status code verification.

The similarity analysis method is a method of comparing the change of the source string at the next connection with the original after removing the HTML tag of the source of the web page accessed by the crawler, and is expressed by the modulated value according to the amount of the changed portion, and keyword analysis method is the check of string matching result contained in the source of the keyword defined by the user, and as the page break/error/delay analysis consists of a simple check of the HTTP status code and the result of HTTP request and response is composed of the type to simply check various HTTP Status Code, so It is impossible to detect malicious codes distributed by small amount of malicious links or file modification according to the change of the tampering/forgery rate on a client basis.

 Copyright ⓒ 2018 SERSC Australia

To prevent such tampering/forgery accidents and malicious programs from spreading, Shin Ji-yong (2016) suggested the similarity between the captured image of the normal website and that of the website by comparison/analysis. In order to compare the similarity of captured images, Feature Matching using OpenCV was applied, and the speed and accuracy of analysis are pointed out as the limit.

To overcome the existing research's limitation, it is essential to monitor and analyze through the subpage, dead link, external link and dynamic behavior analysis. In addition to the website's representative page, it is essential to broaden the link of the subpage which is linked to the website in accordance with the setting of administrator. Likewise, it is essential to discover an external link by contrasting a website's domain name in the web source with other domains as well as to discover a link that is not used currently by checking the detected external link's disconnection. Besides, it is essential to check whether the malicious code is distributed by analyzing all the downloaded files dynamically when the customer accesses the web page in case that whether the website is hardly a malicious code route as well as the website is not the malicious code route.

3. Security Control Model

3.1. Components of the Model

For solving the limitations of current website forgery/tampering detection system, this study proposes a crawler-based security control system which controls the safety of the homepage by using a malware analysis system which includes a forgery/tampering analysis algorithm, a multi-thread crawler system and a dynamic analysis function. The functions of each algorithm are as follows.

First, the MTCS (Multi-thread Crawler System) algorithm can broaden the link of the subpage which is linked to the website in accordance with the setting of administrator in addition to the website's representative page. The feature of the MTCS algorithm is that it uses sequential collection using an HTTP client, sub-URL extraction of pattern method, and file extraction technique of packet segmentation method.

Second, the DAS (Deface Analysis System) algorithm grafts dead link detection technology to detect links that are not currently in use through the technique of detecting external link by comparing domain name of website in web source with other domains with improvement of the technique that performs only the discontinuity of tampering/forgery, delay, error status check and source rate change of existing website.

Third, MAS (Malware Analysis System) algorithm makes the distribution of malicious code checked by analyzing all the downloaded files dynamically when the customer accesses the web page assuming the website is hardly a malicious code route with the improvement of detection types that used a simple web source.

Copyright ⓒ 2018 SERSC Australia

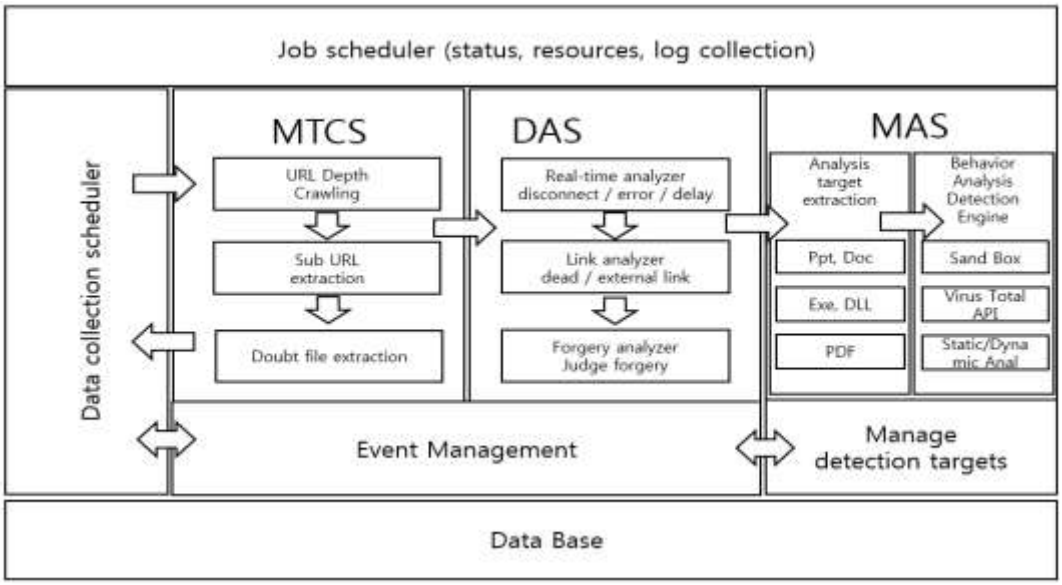

Figure 1. Architecture of Crawler-Based Security Control Model
(Source: Seongmuk Choi et al, 2017)

4. Empirical Analysis

This study proposed a crawler-based security control model as well as established a prototype for enhancing the functionality of the current simple website forgery/tampering detection system.

The first goal of the experiment through the prototype is to verify the range of web crawler collection and the detection of abnormal behavior of the MTCS algorithm. This study verifies that it can extract Sub-URL through the URL pattern comparison by extracting the link information after collecting URI information of the website registered targeting crawler monitoring and analyzing HTML tag (Tag).

The second goal of the experiment is to verify a total of eight detection functions of the DAS algorithm which check the abnormality of the website in various ways. To analyze the disconnection, delay, error, modulation rate, keyword, external link, dead link, and behavior analysis to detect malicious code distribution on the website, this study verifies the effectiveness of malicious code analysis which analyzes the file transmitted through MTCS algorithm and transmits as MAS algorithm and again DAS algorithm.

To test the performance of the algorithms, an existing model and an improved model were constructed in the same environment and 100 detection target websites were set up. The crawler collects the URL to be detected every 5 minutes and downloads pre-collected malicious code to enable behavior analysis. This study shows the analysis results using the collected information and compares the crawler functions of the existing model crawler with the models presented in this study and verifies the analysis performed.

This study also examines the effectiveness of the detection rate improvement through the verification of external links, dead links, and behavior analysis functions that are added to the suggested model.

 Copyright ⓒ 2018 SERSC Australia

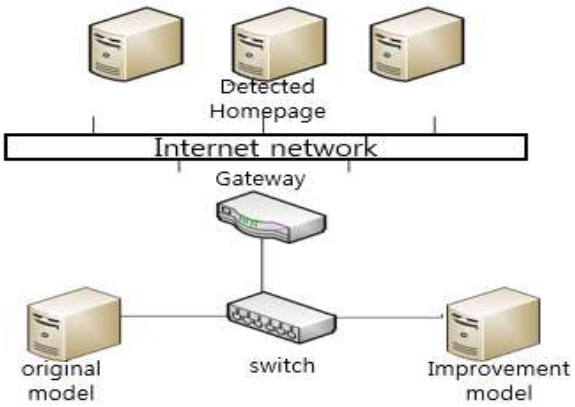

Figure 2. Configuration of the Experiment Environment
(Source: Choi Sung-muk et al, 2017)

In this study, for the Crawler-based Security Control Model's empirical analysis, we tested in the same environment as the existing model. The empirical analysis was conducted in one month, from February 1st, 2017 to February 28th, 2017.

Network (Outside: Dedicate-10M, Insude: 100M), Hardware (CPU: Xeon 2.4GHz, Memory: 64GB, HDD: 1TB SATA, OS: CentOS 7.2, WAS: Tomcat 6.0, DB: Mysql 15.1, JAVA: 1.8)

5. Experimental Results

5.1. MTCS Algorithm Performance Verification

This study tested the performance of MTCS algorithm in the same configuration and environment. In the case of existing models, sub-URL extraction is impossible because only main-URL is collected. However, this model analyzes the HTML pattern of Main-URL and extracts sub-URLs using link information. As a result, this study extracted 125,210 URLs and collected 35 sub-URLs.

As a consequence of testing 100 target domains, the proposed model in this research could extract Entity list which use the HTTP client and collect the subdomains through the Mail-URL's HTML pattern later. Such displays the ability of MTCS algorithm in performing a broader monitoring in the comparison with the existing model.

Moreover, when the crawling performance is observed, the average analysis speed for one domain is measured for almost one month, it is measured that the average crawler time for 100 Main-URL for current homepage forgery/tampering detection model was measured as 3 minutes and 27 seconds, and it was measured to be at least 2 minutes and 2 seconds and maximum 4 minutes and 3 seconds. In the same configuration and environment, the MTCS algorithm crawler performance was measured as about 125,210 (including 100 Main-URLs) and the average analysis speed for each domain was 3 minutes and 12 seconds.

5.2. DAS Algorithm Performance Verification

Disconnect detection is Reconnection 3time (15minutes) in date 1, Error detection is 1 case upon HTTP status Code!=200, Delay detection is 1 case with 30seconds delay, Keyword detection is 1 case upon matching registered keyword, Change Rate detection is 1 case when changing more than 70%, External links are 1 detection of external links

Copyright ⓒ 2018 SERSC Australia

other than the Main-URL name format in Web source, Dead link is 1 case upon detection of non-connected links in web source.

Table 1. Detection Standard by Event

Detection Events	Crawler-based Security Control Model			Existing Model
	Main-URL	Sub-URL	Total	Main-URL
Disconnection	64	9	73	64
Error	287	676	963	287
Delay	269	423	692	269
Keyword	0	0	0	0
Change Rate	20	584	604	20
External Links	10,257	27,579	37,836	Undetectable
Dead Link	1,529	2,937	3,926	Undetectable

In other words, there are many detections in the crawler-based security control model compared to the existing model in all the discontinuity, error, delay, and change rate, and 37,836 external links were detectable not in the existing model, but in the crawler-based security control model, and dead links were 3,926. The detected values show that the detection range is wider than the existing model, and this shows that discontinuity, error, delay, and rate of change are occurring even in the range that was not detected in the existing model, and that the crawler-based security control system has the effect of detecting it.

External link and dead link detection is a function that detects malicious link insertion by web hacking, which cannot be detected in existing model, and in the crawler-based security control model, 10,257 Main-URLs and 27,579 Sub-URLs were detected in external links, totaling 37,836 cases.

5.3. MAS algorithm performance Verification

At first, the MAS module analyzes the malicious nature of all the files collected and transmitted by the MTCS module. Then it transmits the analyzed behavioral data to the DAS module. The comparison of behavioral analysis in the same environment that validates the existing model and the improved function of this study model during the experimental period does not fit. From thousands to tens of thousands of all types of formatted files by website analysis of the behavior of the file does not fit the experiment to verify this function. Therefore, the analysis of the behavior was set so that only document files and PE files can be analyzed, except various image-use files on website, such as JPG, GIF, Flash. A total of 4,375 files were analyzed during the experiment and 579 malicious files were detected.

Compares the results of the existing model and the crawler-based security control model of this research model. The discovery rate of the present model was high for discontinuity, error, delay, and modulation rate of the existing models. Moreover, as a result of adding external and dead links, and malicious code detection functions, we are able to confirm that it is possible to uncover even the parts the previously existing models could not detect. Thus, the performance of the MTCT, DAS, and MAS algorithms of the crawler-based security control model has been proved. As a result, it is expected that the scope of controlling and monitoring will be widened, and homepage security can be improved by detecting malicious codes.

 Copyright ⓒ 2018 SERSC Australia

Table 2. Comparison of Verification Result

Item	Crawler-based security control model	Existing model
Main-URL	125,210	100
Average crawling time (28-day average)	192 seconds	207 seconds
Disconnect Detection	73	64
Error Detection	963	287
Delay Detection	692	269
Keyword Detection	0	0
Modulation Rate Detection	604	20
External Link Detection	37,836	-
Dead Link Detection	3,962	-
Malicious Code Detection	4,375 file analyses (579 detected)	-

6. Conclusion

6.1. Summary of Research Results

This research studied the crawler-based security control model, that is an improved model regarding the tampering/forgery monitoring system currently in operation, to monitor the safety of web sites in a 24-hour, 365-day basis operated by security control centers.

Previous studies regarding monitoring methods of web site tampering/forgery were limited only to monitoring simple tampering/forgery using images and codes. Such simple monitoring of tampering/forgery is not able to detect the spread of malicious code throughout the web.

To overcome such limitations, analysis and monitoring through external links, dead links, and dynamic behavior analysis are necessary. In addition to the main page of the website, the link to the subpage in connection to the website needs to be analyzed through the administrator ‘s setting. Furthermore, it is essential to discover an external link by contrasting the website’s domain name with other domains in the web source and to discover a link which is not used currently by examining and confirming the disconnection of the detected external link. Not stopping at the detection of the web source, it needs to be considered that the website is not just a transit for the malicious code but the origin of distribution. In such case, when a client accesses a web page, it must be able to dynamically analyze the downloaded file to check whether the malicious code has infected the system.

This research applied various functions and methods to detect the cyber-hacking threat, which intimidates the security of the website, in the target model in a situation where the website hacking attacks such as various web hacking attempts and new application vulnerabilities are increasing daily.

The first component of the crawler-based security control model presented in this study is an MTCS module that monitors the enlargement of sub-URLs, rather than the target website’s main, unlike the existing website tampering/forgery monitoring system. This is due to the importance of expanding the scope of the automatic crawler to access and monitor the entire structure of the website, as the web page is the representative page with the greatest damage due to the attack of the website. However, there are various hacking attempts are made also to the subpage.

The second component of the model is a DAS module that detects the risk of a website by various functions. It is an algorithm that detects the status or malicious link of the

Copyright ⓒ 2018 SERSC Australia

website with various data stored by sequential access to the website from the MTCS module crawler. Important factors of website security including DDoS attack and DNS forgery are modeled to examine the status of disconnect, error, and delay to monitor the availability. This study has examined an algorithm that allows the user to register and detect harmful regular expressions and irregular patterns registered in the contents of the web source. In addition to the previous algorithm, another one was examined to monitor whether the web source is modulated by the rate of change by expressing the web source as a hash value. Moreover, an additional algorithm was suggested to detect dead links for discovering traces of external links embedded in the web source due to web hacking attacks or web hacking attacks of the past which are currently not active.

The third component of the proposed model is the MAS module. It is a module that tracks malicious files by analyzing the behaviors of a file downloaded when accessing a website through a crawler access such as a client environment that a malicious code is distributed. Behavioral analysis was the source to develop the algorithm of the MAS module using open source Cuckoo Sandbox. Moreover, it designed the detection and analysis results to be transmitted to the DAS module for detection.

The experiment was conducted on 100 websites, and it was confirmed that the website's forgery/tampering hacking accidents, which previously could not be monitored by the convention and simple website forgery/tampering detection model, was detected effectively. Particularly, during the experiment period, this study broadened the range to 125,210 websites, and proved the ability of crawler in detecting the keyword, error, delay, modulation rate as well as disconnection of the website without decreasing the speed in a five-minutes cycle. All along the experiment, without losing the hacking attacks, the sub-URLs could be discovered. By going through the analysis of behavior, this study discovered the malicious code-spreading pages; besides, the proposed model's performance was confirmed as well.

6.2. Limitations of Research and Future Research Directions

Experiments conducted to analyze the performance of models presented in this study, were carried out by specific prototypes and may have limitations in applying them over long periods of time in the actual operational environment. Due to the inability to operate and experiment various systems at the same time, the model used as comparison could only be compared to specific models. Furthermore, randomness regarding the data collection period and target site selection are considered limitations of this experiment.

Therefore, it is necessary to refine this model beyond the prototype level in the near future, in which it can be applied to the actual environment. This model needs to be applied with a sufficient time period to a wider variety of system environments, to collect reasonable amounts of data and analyze it to results.

However, despite the limitations mentioned above, the model complements the limitations of the existing network-based IDS and IPS, which were illustrated by website tampering/forgery detection researches in the past that simple tampering/forgery monitoring is not enough to detect the distribution of malicious code through the web. At the same time, the proposed model is expected to be able to adapt to the various new security vulnerability patterns that increase daily in various fields and industries.

Moreover, in the academic perspective, the currently existing crawler method based on the client connection proposed in this study, shows the direction of the new research in situations where the existing studies are concentrated on the signature-based detection, which is considered to be significant.

Based on this study, additional research and studies are needed regarding the multi-stage parallel crawler with a 2-tier structure and the technique for parallel behavior analysis of all format files in a single web page.

 Copyright ⓒ 2018 SERSC Australia

References

[1] National Intelligence Service, "National Cyber Safety Management Regulations", 2 in Article 10, **(2013)**.
[2] J.-H. Kim, "A Study on the Measurement Index of Outsourcing Security Management Level of Public Institutions", Doctoral Thesis, Graduate School of Soongsil University, **(2014)**.
[3] J. Moon Yoon, "Design and Implementation of Real-Time Integrated Analysis System Based on MPSM for Malicious Code Response", Ph.D. Thesis, Graduate School of Kyonggi University, **(2014)**.
[4] J.-H. Oh, C.-T. Lim and H.-C. Jung, "Drive-by Download Technology Trends and Solutions", Journal of the Korean Information Science Society, vol. 28, no. 11, **(2010)**, pp. 112-116.
[5] J.-Y. Moon and Y.-H. Jang, "Ransomware Analysis and Minimization Plan of Damage", The Journal of the Convergence on Culture Technology (JCCT), vol. 2, no. 1, **(2016)**, pp. 79-85.
[6] J.-M. Yoon, J.-K. Jo and J.-C. Ryu, "Ransomware Attack Blocking Methodology Using File I/O Interval", Journal of the Korea Institute of Information Security, vol. 26, no. 3, **(2016)**, pp.645-653.
[7] Korea Ransomware Infringement Response Center, "2017 Ransomware Infringement Analysis Report", **(2017)**.
[8] Korea Internet & Security Agency, "'16 Ransomware Trend and '17 Outlook", **(2017)**.
[9] D. Orc Tabianto, I. Malhardiano and Y. Kim, "Analysis of Malicious code using Cuckoo Sandbox", ACOM Publishing, **(2014)**.
[10] T.-K. Kim, "A Study on Malicious Code Detection Method", Journal of Security Engineering Research, vol. 9, no. 5, **(2012)**, pp.387-400.
[11] D.-W. Seo, A. Khan and H.-J. Lee, "A Study on Detection of Malicious Code Diffusion Site", Collection of these in KIIS Fall Conference, vol. 15, no. 2, **(2008)**.
[12] H. Xu, Y. Zhang and Y. Hu, "Study on a deep web crawler in security validation mode", Jisuanji Yingyong yu Ruanjian, vol. 27, no. 5, **(2008)**, pp. 9-11.
[13] K. Kim, S. Choi, H. Park, S. Ko and J. Song, "Website Falsification Detection System Based on Image and Code Analysis for Enhanced Security Monitoring and Response", Journal of the Korean Institute of Information Security and Cryptology, vol. 25, no. 5, **(2014)**.
[14] L. Xuesong and Z. Rong, "Influences and Countermeasures of Web Crawler on Network Security", Jisuanji yu shuzi gongcheng, vol. 37, no. 12, **(2009)**, pp. 86-88.
[15] Q. Zhiqing and H. Fei, "Vulnerability Mining Tool Base on Crawler and Fuzzing", Microcomputer applications, vol. 32, no. 3, **(2016)**, pp.73-76.
[16] Y. Kawano and E. Nunohiro, "A Proposal of Distributed Autonomous Cooperative System about Exclusive Web Crawling for Cyber Security", Institute of Electrical and Electronics Engineers, **(2016)**.
[17] D. Doran and S. S. Gokhale, "An integrated method for real time and offline web robot detection", Expert systems, vol. 33, no. 6, **(2016)**, pp. 592-606.
[18] H. Shin and J.-S. Moon, "A Study on Minimizing Infection of Web-based Malware through Distributed & Dynamic Detection Method of Malicious Websites", Korea Internet & Security Agency, Korea University, **(2011)**.
[19] P. Sriram Rohit and R. Krishnaveni, "Deep Malicious Website Detection", International Journal of Computer Science and Mobile Computing, vol. 2, no. 4, **(2013)**, pp. 517-522.
[20] S. Choi, "An Empirical Study on Crawler-based Security Control Systems", Doctoral Thesis, Graduate School in Soongsil University, **(2017)**.

Copyright ⓒ 2018 SERSC Australia

Copyright ⓒ 2018 SERSC Australia