International Journal of Control and Automation
Vol. 8, No. 4 (2015), pp. 83-88
http://dx.doi.org/10.14257/ijca.2015.8.4.10

# The Endpoint Application Execution Control Scheme Based on the Whitelist

Chang-Hong Kim[1], Jeong-Hyun Yi[2] and Jong-Bae Kim[3*]

[1]*Department of IT Policy and Management, Graduate School of Soongsil University, Seoul 156-743, Korea*
[2]*Department of Computer Science, Soongsil University, Seoul 156-743, Korea*
[3*]*Graduate School of Software, Soongsil University, Seoul 156-743, Korea*
[1] *chkim@nextup.kr,*[2] *jhyi@ssu.ac.kr,*[3*]*kjb123@ssu.ac.kr*

***Abstract***

*Under the situation that existing information protection systems adopt an approach that tackles malicious codes based on already known signatures or analyzed behavior/feature, they are limited in detecting and identifying the unknown and the deformation of the malicious code. The present study, as a means of overcoming such a shortcoming, proposes a way of endpoint application control capable of more securely protecting endpoint devices (PCs) from intrusion of malicious codes and attacks through exploitation of an application or operating system vulnerability, by implementing a hybrid of technology such as whitelist-based application execution control via authentication of integrity, media access control, prevention of modification of important files, and control over IP/port attempting for process access or reverse link.*

***Keywords:*** *Malicious code, Execution control, Media access control, Endpoint, Information protection*

## 1. Introduction

The cyber threat landscape is recently increasing by keeping abreast with the advancement of technology, causing damages from cyber incidents to become bigger due to evolving malware [1]. Persistent attacks, targeting terminal PCs ("Endpoints") where malicious code can be hidden and running covertly, by evading traditional network security and signature-based AV systems for information protection and disseminating malicious codes have brought about a vast size of damage, such as personal information leakage and system failure secured by national agencies, banking institutions, shopping malls or telecommunication companies.

Nevertheless, the currently applied information protection system reveals limitations to combat malicious codes targeted at endpoint devices; that is, as they respond to attacks by taking advantage of already secured signatures or information analyzed before, there exist limits for detection and identification even when encountering the deformation of malicious code, as well as exploits [2] by malicious code with its signature unknown, attacking the susceptibility of operating system and application program.

In order to identify and block unknown malicious codes, a sophisticated combination of analysis and detection technology is required by using the static information analysis methodology, such as authentication of integrity based on hash information of executable file, attribute and route information of files and authentication of e-signature of files, and the dynamic information analysis methodology, such as detection of abnormal program execution after its running [3].

* Corresponding author. Tel. : +82-10-9027-3148.
Email address: kjb123@ssu.ac.kr(Jong-Bae Kim).

ISSN: 2005-4297 IJCA

In this regard, the current study proposes an option for endpoint application control capable of more securely protecting endpoint devices from intrusion of malicious codes and attacks through exploitation of an application or operating system vulnerability, by implementing a hybrid of technology such as whitelist-based application control via authentication of integrity, media access control, prevention of modification of important files, and control of IP/port from reverse link.

## 2. Related Works

The method blacklisting is used for the purpose of quickly responding to a detected attempt of attack, before it is more widely disseminated, by analyzing its behavior/feature characteristics.

On the contrary, the method whitelisting maintains robust security capable of blocking new malicious code with its hazard level not proved, since it allows only input values proven as safe [4].

Efforts to shut off unapproved attempts of access by resorting to whitelisting-based configuration are seen in the research of [5, 6, 7], etc.

However, as most of them are focused on blocking access through network, they are not adequate for a method to prevent installation or execution of unapproved malicious code.

## 3. Proposed Execution Control System

In this study a method of controlling endpoint application execution is proposed so that terminal PCs are more securely protected from intrusion of malicious codes and exploit attacks to vulnerable OS and applications.

The system of endpoint application control as defined herein is comprised of three sections : Agent Section that prevents programs unapproved (i.e. not registered to the whitelist) and unidentified executable files not to be executed before they run; Event Server Section that transmits a command of application execution or a command of application running stop; and Monitoring Console Section that offers functions of policy operation management, registration to whitelist and application management.

### 3.1. Agent Section of the System

The section Agent installed in a terminal PC plays the role of the authenticating process with respect to integrity based on whitelist prior to running of an application program and based on static information analysis. When an event of requesting for execution of application occurs by an operating system, it is designed that an executable file is allowed to run if it has been registered to the whitelist by a medium permitted by execution control policy, but any program unapproved due to non-registration to the whitelist, unidentifiable application program, or executable file are blocked from running.

At this point, the Agent transmits application execution history to the event server, and then the event server analyzes the transmitted history to perform the modification of the whitelist and execution control policy.

In addition, the Agent installed to a terminal PC authenticates flow of process execution in accordance with the execution control policy after an application program runs.

 

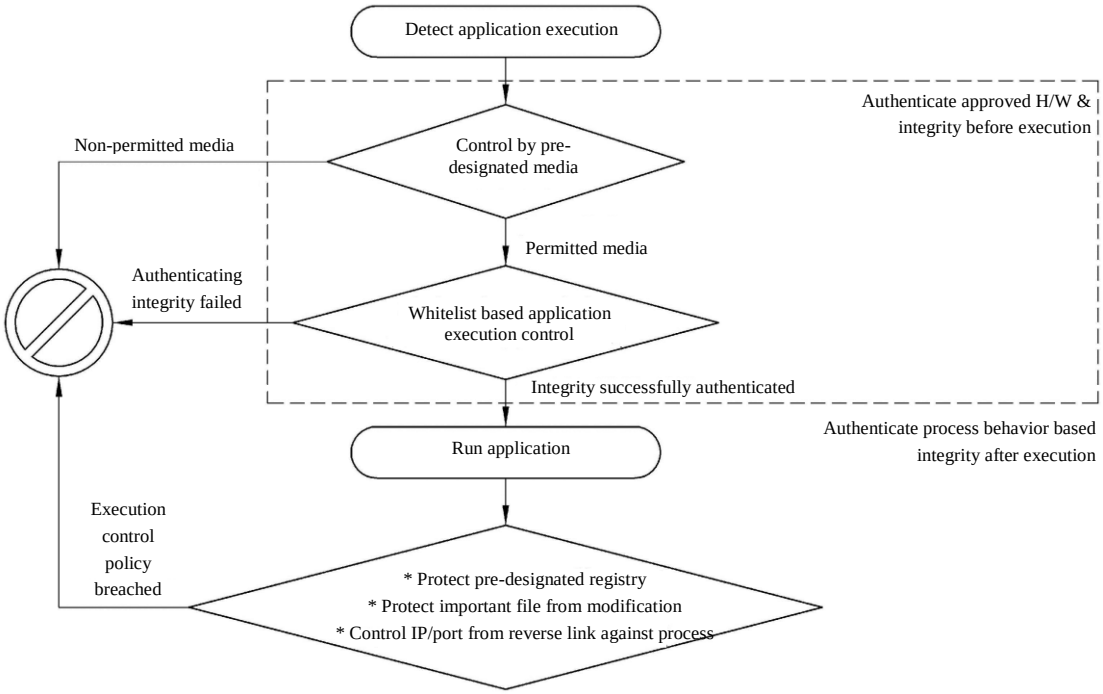


**Figure 1. Sequence of Controlling Endpoint Application Execution**

As shown in Figure 1, the sequence for controlling endpoint application execution is made in such a way that the Agent installed in an endpoint device detects an event of application execution from the operating system and primarily identifies whether the detected execution event is being executed in a medium whose route is permitted.

If so, then it performs authenticating integrity of the application program through whitelist based application execution control loop by using hash value of the execution file, file attributes and route information.

Lastly, it performs process behavior based authentication of the application, once executed, and based on execution control policy with regards to prevention of modification of important files and control of process reversing IP/port, all of which are pre-designated in the policy.

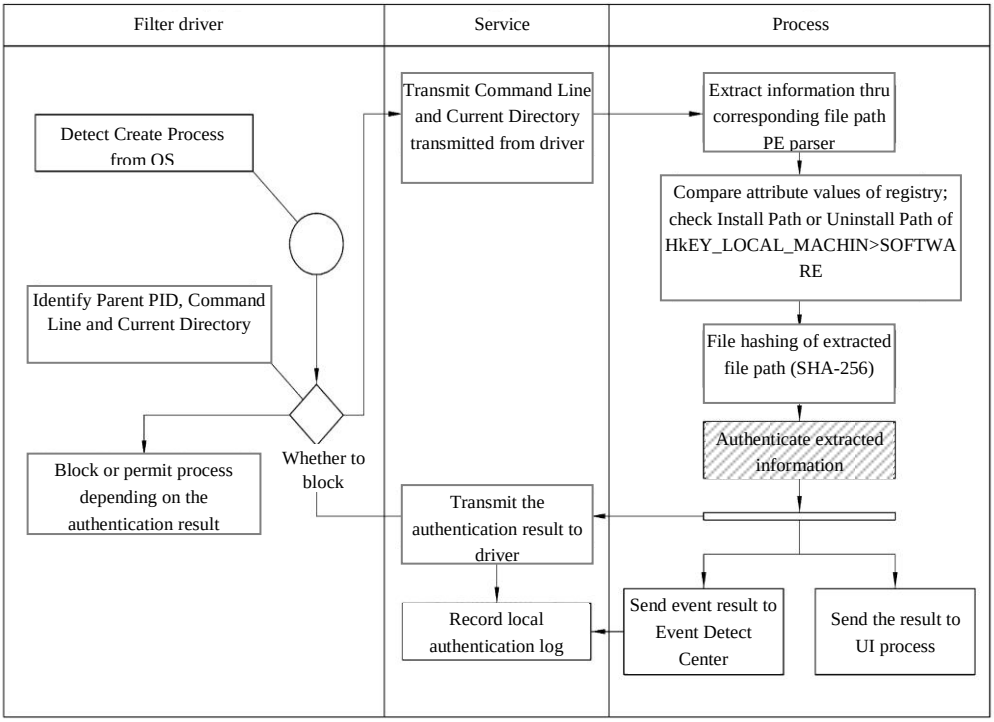


**Figure 2. Process of Authenticating Integrity based on Pre-designated MAC and Whitelist**

Figure 2 accounts for the processing of integrity authentication based on pre-designated media control and whitelist. In the case of media control pre-designated by policy, any attempt of indirect or unapproved accessing to any port (e.g. USB port) of an endpoint device that connects with peripheral device is blocked.

### 3.2. Event Server Section of the System

As shown in Figure 3, the Event Server acts to manage logs of the history of execution of application programs and blocking of their execution being transmitted from the Agent.

Also, it determines whether an application program requested by the Agent for registration to the whitelist has been normally approved, and then if so makes the whitelist to be modified as appropriate and sends the execution control policy to the Agent.

 

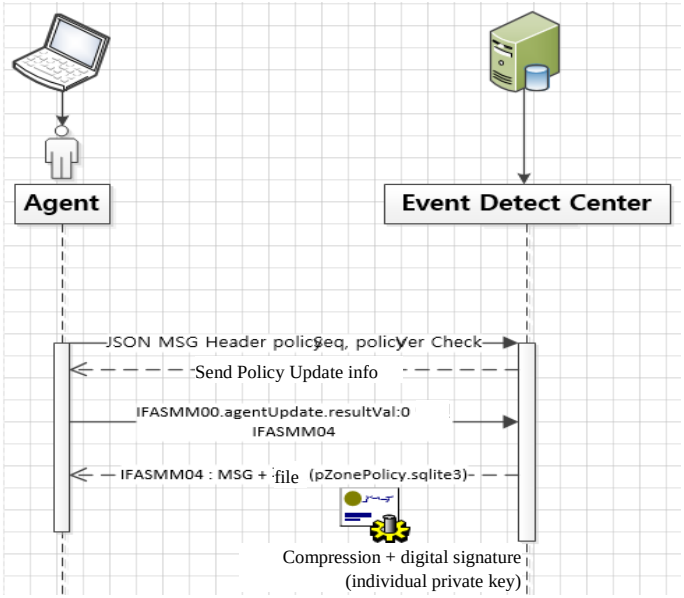


**Figure 3. Process of Sending Policy info between Agent and Event Server**

### 3.3. Monitoring Console Section of the System

The Monitoring Console is in charge of support tasks such as notice, management of application programs registered to the whitelist, management of policies, analysis on the status of services and functions involving system operation.

In addition, it provides various charts and statistical reports associated with the status and tendency of data identified through monitoring, and carries out creation and registration of policies, registration of updated modules, fault/failure management, reporting on frequency of using application programs and various log related information, retrieval of user login history, etc.

Aside from that, it allows the administrator to distribute execution control policies from the Event Server to the Agent, through the Monitoring Console, by ways of designation of files as important, designation of object as MAC and designation of process subject to control of reversing link.

## 4. Conclusion

The system of endpoint application execution control proposed by this study adopts combined technology relating to whitelist based application execution control, prevention of modification of critical files, pre-designated MAC and control of IP/port from reverse link with process, which, in a narrow sense, sets a foundation of controlling activity using malicious codes, rather than malicious codes themselves, through prevention of unapproved executable application from running, prevention of important files from forgery and blocking of indirect access to ports for movable devices, thereby in a more secured fashion protecting endpoints from zero-day attacks via unknown malicious codes and exploit attacks by exploiting the security vulnerability in the operating system or application program.

By making combined use of authenticating the application's integrity before its running and control over executing flow of application after its running, this study provides a means of basically blocking unknown malicious codes, which is not achievable by the conventional information protection system that relies on information of signatures of blacklisted malicious codes.

## References

[1] J. K. Park, "A Realtime Malware Detection Technique Using Multiple Filter", JKSCI, vol. 19, no. 7, **(2014)**, pp.77-85.
[2] M. J. Kim and J. H. Yoo, "Estimating Economic Loss by S/W Vulnerability", JSEBS, vol. 19, no. 4, **(2014)**, pp.31-43.
[3] C. H. Kim, D. Y. Choi, J. H. Yi and J. B. Kim, "A Study of Program Execution Control based on Whitelist", Conference on Information and Communication Engineering, KIICE, **(2014)**.
[4] D. S. Lee, "Threats according to the Type of Software Updates and White-List Construction Scheme for Advanced Security", JKIICE, vol. 18, no. 6, **(2014)**, pp.1369-1374.
[5] J. W. Kim, J. T. Ryu, K. Y. Ryu and B. H. Roh, "A Countermeasure Scheme Based on Whitelist using Bloom Filter against SIP DDoS Attacks", KICS, vol. 36, no. 11, **(2011)**, pp.1297-1304.
[6] D. H. Lee and K. H. Choi, "A Study of an Anomalous Event Detection using White-List on Control Networks", JISKIASQ, vol. 12, no. 4, **(2012)**, pp.77-84.
[7] K. Y. Bae, K. S. Chae and Y. B. Kim, "SPIT Prevention Framework using Expanded White List", JIEIE, vol. 47, no. 2, **(2010)**, pp. 95-102

## Authors

**Chang Hong Kim**, he received his bachelor`s degree in Public Administration from Soongsil University, Seoul, Korea, (1993). He worked in the IT field as a Information Security Architect over 25 years. Now He is President of Nextup Technoplogy Co., LTD. since 2011.

**Jeong Hyun Yi** is an Associate Professor in the School of Computer Science and Engineering at Soongsil University, Seoul, Korea. He received the B.S. and M.S. degrees in computer science from Soongsil University, Seoul, Korea, in 1993 and 1995, respectively, and the Ph.D. degree in information and computer science from the University of California, Irvine, in 2005. He was a Principal Researcher at Samsung Advanced Institute of Technology, Korea, from 2005 to 2008, and a member of research staff at Electronics and Telecommunications Research Institute (ETRI), Korea, from 1995 to 2001. Between 2000 and 2001, he was a guest researcher at National Institute of Standards and Technology (NIST), Maryland, U.S. His research interests include mobile security and privacy, IoT security, and applied cryptography.

**Jong-Bae Kim**, he received his bachelor's degree of Business Administration in University of Seoul, Seoul(1995) and master's degree(2002), doctor's degree of Computer Science in Soongsil University, Seoul(2006). Now he is a professor in the Graduate School of Software, Soongsil University, Seoul, Korea. His research interests focus on Software Engineering, and Open Source Software.

 