Indian Journal of Science and Technology, *Vol 8(S8), 10-14, April 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
DOI: 10.17485/ijst/2015/v8iS8/71501

Security Evaluation Metric of Windows-Based Information Security Products

Kisoo Kim, Sangho Lee*, Yeowung Yun, Jaemin Choi and Hyungjin Mun
Department of Computer Science, Chungbuk National University; caram12th@naver.com, shlee@cbnu.ac.kr, ywyun@ksel.co.kr, cjmin010@naver.com, jinmun@gmail.com

Abstract

Recently, when updating the products for Windows-based agents, an integrity breaches by the malicious files or unusual behavior of the product by the user's intentional product modulation / deletion is causing a frequent accident cases. The PC based products such as anti-virus, personal firewall (integrated PC security), data loss prevention, security USB and network access control that perform the functions of the Agent based on Windows operating systems, the product itself gets attacked and becomes neutralized or has the existence of vulnerabilities that bypass security. To prevent this, we would like to give help in the development to organize the self-protection function which are required for Windows-based information security products in metrics.

Keywords: Self-Protection Function, Windows-Based Agents

1. Introduction

The PC based products such as anti-virus, personal firewall (integrated PC security), data loss prevention, security USB and network access control that perform the functions of the Agent based on Windows operating systems, the product itself gets attacked and becomes neutralized or has the existence of vulnerabilities that bypass security. To prevent this, organizing of the self-protection function which are required for the Windows-based information security products This paper is organized as follows. The 2nd chapter deals with considerations and assumptions for the self-protection function. The 3rd chapter defines metrics for realization of the self-protection and provides help to substantialize the self-protection function. The 4th chapter shows verifying procedures through metrics, and finally the 5th chapter derives a conclusion.

2. Realization of Self-Protection

The realization of self-protection function, regardless of its type of information security product and the role, must satisfy the following points.

*Author for correspondence

2.1 Mandatory Realization Points

1) The installation folder (C: \ Program Files \ Agent) for the Agent does not include the execution file that performs self-protection function. When the Agent execution file and self-protection execution file is in the same folder, a user can easily disable the self-protection function. 2) The Agent and the self-protection must operate at the service level of the Windows. When operating at a process level the Agent and self-protection can be easily neutralized. 3) The self-protection execution file is located at the folder where the user cannot easily find. 4) The file name for the self-protection execution file is named so the user cannot easily infer. 5) Also the self-protection service name, it is named so the user cannot easily infer. 6) For the integrity deletion prevention function and availability auto-recovery function, a backup folder to recover the Agent installation folder (source folder) is required. 7) A mutual monitoring should be performed so that when the Agent is stopped, it restarts using the self-pro tection function and when the self-protection functionstops it can restart through the Agent. 8) Self-protection process (service) must start before the login of the Windows manager. 9) When removed using the "program add / remove" of Windows control panel or Uninstall.exe file which are

supported by the product, in order to prevent unauthorized access a deletion password should be entered. 10) Agent deletion policy allocation through the manager or if the user inputs the deletion password to Uninstall the product, the self-protection function must be terminated. 11) When the user performs Uninstall using a deletion password the manager should know that Agent has been removed. At the Agent management menu of the Manager's web page, the Agents installed within the network for protection is managed and monitored. 12) Upon detection of the integrity damage of the Agent, in order for the manger to recognize the integrity damage of the Agent, a log alert or manage mail notification function must be available. 13) The encryption key, settings, executable files, certification, DDL file, signature (pattern) are applied with the confidentiality, integrity and availability mechanism.

2.2 Assumptions

The self-protection function of windows based Agent involves assumptions as follows. 1) The function may not be enabled in Safe Mode. However, an alarm about the integrity error is always generated when booting up in the normal mode after changing the integrity in Safe Mode.
2) When menacing agents try to get a grip on the self-protection services and additional ones, if the process is stopped at the same time, the product will be neutralized. However, users can restart the process automatically if the process is interrupted in the case that the service based Agent and the self-protection function are able to work.
3) To activate the Agent in service level of Windows, it should be stored as the plain text in the registry because the zone is used by Windows. If the menacing agents remove the registry key, the self-protection function may be neutralized. 4) When menacing agents delete the backup folder and Windows-based Agent folder with an anti-root kit software, the self-protection function is neutralized even though the process is progress. After considering all the factors, when files and folders related to the self-protection are exposed to the menacing agents, the function is not safe at all. Paper of [1] represents a secure and fast han Windows based Agent should be managed via an unique identifier aside from the self-protection function. 1) If the Agent is verified as an address or IP of MAC address of subordinate NIC, the administrator can recognize the value as if the agent newly registered it in a specific PC. 2) Although the PC with the Agent is equal as it was, the intact replacement of NIC owing to the malfunction causes changing of MAC address. In that case, administrator recognizes the agent as a new one not the one which has been managed since when it was installed.3) If it is recognized as a new Agent because of modulation attacks to MAC address and falsification of network information, beware of bypassing the existing allocation policy.4) It is because the management server can identify it as a new one and execute a default policy when the Agent which had been operated under the existing policy A is allocated with a new MAC. Although this is not a major problem in TOE operating environment where all the Agent works under an equal policy, the vulnerability to bypassing policy can be a problem in an operating environment where all the agents are operated under respective policies.5) The Agent should be managed by an unique and irreplaceable identifier which is not equipment-subordinate. The unique identifier is considered as follows. A specific string the user entered during installation of the Agent. A specific string allocated in the progress of registration after installing the Agent: registry and configuration files are stored. 6) After the product which identifies the Agent based on MAC stores MAC address of NIC into the registry or configuration file during installation, it notifies the server that the MAC address is changed when NIC is altered or MAC address is modulated, and it not only inherits the existing policy but makes allowance for a function to re-register the Agent.

2.4 Additional Implementation Details

1) Encryption key management
The encryption key cannot be included in the source code. The encryption key should be stored in the source code, encoded, or an execution file including the key and DLL file itself should be encrypted. In this case, another encryption key, such as asymmetric key, private key, and session key, which decodes the encryption key encoded in the source code, is required.
2) Manual Recovery Availability is divided into auto-recovery and manual recovery. Auto-recovery applies to the product that should be recovered to be in Safe Mode. Integrity being damaged, one way to recover it manually is to output the procedure and method on pop-ups. Or, an operation manual containing corrective measures to recover, reinstall, recover via the management server or update should be provided. In case of manual recovery, TOE should be enabled not to expose or leak out assets from error occurred point to starting point.

3. Self-Protection Security Evaluation Metric

For the security evaluation of the information security products the following metric is used to measure the confidentiality, integrity and availability.The contents summarized in the Table 1 can be used as a check list for self-protection realization level. The things to be realized for the self-protection of the Agent are summarized into a checklist

Table 1. Self-protection security evaluation metric

Security	Metric
Confidentiality	Are the encryption target registry / file encrypted?
Confidentiality	Are the self format target registries / file being protected above the encoding level?
Integrity	Status of the detection for the integrity damage during the booting after the Agent modulation at the safe mode
Integrity	During the detection of the integrity damage of the Agent (integrity protection target) is there an alert function to notify the user through a pop up window?
Integrity	Is there a hash table for checking the integrity?
Availability	Does it satisfy the self protection operation time?
Availability	Is the execution file that performs the configuration file for the Agent and self protection function is stored in a different folder?
Availability	Is the Agent process and self-protection process being operated at the Windows service level?
Availability	Can the Agent be inferring through the self protection execution file name (process name)?
Availability	Can the Agent's service name be inferred through the self protection service name?
Availability	Is the deletion of the Agent can be possible only through the manager's deletion policy allocation or by input of the password of the user?
Availability	If the Agent stops and restarted using the self protection process, when the self protection process stops, can the Agent be restarted through a mutual monitoring?
Safe management	When Uninstalled using the deletion password of the user, can the manager know that the Agent has been removed?
Bypass policy	Is there vulnerability for MAC tampering attack?
Integrity Deletion prevention	When the total TOE installation file has been deleted, is the corrective actions provided in forms of the user's operation manual to the user for the recovery (recovery through reinstallation, management server / update server)?
Integrity	'When the file has been deleted by the 'IceSword' and 'GMER' the time where the integrity inspection is performed(for example: when calling the function, at the user's request, update time, PC boot time, etc.) Can the user/manager be able to recognize this?
Integrity	When the backup folder is deleted, does the integrity damage alarm go off? (not mandatory)
Availability	When the integrity deletion protection target is deleted, is it automatically restored? Except for neutralized attacks by 'IceSword' and 'GMER' it should be able to be recovered.
Availability	When auto-recovery is blocked because the integrity deletion prevention is deleted and falsified, is manual recovery possible? Except for neutralizing attacks by IceSword and GMER, it should be able to recover.
Availability	Recovering the integrity protection target, does it use either backup folder or remote server to recover it manually and automatically? Even if the backup folder is deleted and connection with the remote server is failed, is it capable to recover?
Availability	When neither the integrity deletion prevention nor the availability auto-recovery works, does it provide the administrator with any method to notify him that?

4. Verification

The practical product is evaluated with suggested metrics above. There were differences between A and B of which implementation method are subtle after analyzing

Table 2. Comparison of the self-protection implementation range

Functions		Product A	Product B
Work	whether works at the point of booting up OS	o	o
Confidentiality	encryption of important file	o	o
Integrity	integrity alarm when integrity damage is detected	o	o
	use of hash table for checking integrity	o	o
Availability	prevention of important file deletion	o	o
	prevention of registry information deletion	o	o
	restarts when process is terminated	o	o
	self-protection implementation in service level	o	o
	Safe Mode activation	X	o
	use of backup folder to prevent deletion	o	o
	log transmission when integrity damage is detected	X	o
	notification function for administrator when integrity damage is detected	o	X
	whether checks integrity damage when booting up in normal mode after modulating agent in Safe Mode	o	o
	traceability for administrator when uninstalling Agent	o	o
	alarm of integrity damage when backup folder is deleted	o	X
	mutual monitoring between Agent and Watchdog	o	o
	nondeletable by Add or Remove Programs of Control Panel	o	o
	whether uses of Agent with password for deletion	o	X
	whether deletes Agent via delete policy of administrator	X	o
	workability of Agent when service registration is deleted	X	X
	vulnerability against MAC falsification attacks	X	o

the product with the suggested method. Leaving the analysis result aside, there exists residual vulnerability of neutralization when the service registry is deleted and the self-protection process including service and backup folder is exposed.

5. Conclusion

it has provided the methods to protect the information security product itself and supplied the metric to measure the realization and the performance indicators. Also, it has organized the self-protection function which is required by the Windows based information security products. If the metric is used according to the level of realization of development of Agent, it will be helpful to the information security product developing company or to the developers.

6. References

1. Blount J, Tauritz D, Mulder S. Adaptive rule-based malware detection employing learning classifier systems: a proof of concept. Proceedings IEEE 35th Annual Computer Software and Applications Conference Workshops; 2011 Jul; p.110–15.
2. Chess D, Palmer C, White S. Security in an autonomic computing environment. IBM Systems Journal. 2003; 42(1):107–18.
3. Elkhodary J, Whittle. A survey of approaches to adaptive application security. Workshop on Software Engineering for Adaptive and Self-Managing Systems; 2007 May; p.16.
4. Erlingsson U, Schneider F. SASI enforcement of security policies:a retrospective. DARPA Information Survivability Conference and Exposition, 2000; 2000; 2: p. 287–95.
5. English, Terzis S, Nixon P. Towards self-protecting ubiquitous systems monitoring trust-based interactions. In UbiSys '04: 2004.
6. Ganna M, Horlait E. Toward secure autonomic pervasive environments. Proceedings IEEE GLOBECOM '05; 2005 Nov; 2: p.6.
7. Liang Z, Sekar R. Fast and automated generation of attack signatures: a basis for building self-protecting servers. ACM conference on Computer and communications security; 2005; p. 213–22.
8. Lorenzoli, Mariani L, Pezze M. Towards self-protecting enterprise applications. IEEE International Symposium on Software Reliability (ISSRE '07); 2007 Nov; p.39–48.
9. Marcus L. Local and global requirements in an adaptive security infrastructure. International Workshop on Requirements for High Assurance Systems; 2003 Sep.
10. Maximilien, Singh M. Toward autonomic web services trust and selection. International Conference on Service Oriented Computing; 2004; p.212–21.
11. Nguyen Q, Sood A. Designing SCIT architecture pattern in a Cloud-based environment. International Conference on Dependable Systems and Networks Workshops (DSN-W); 2011 Jun; p.123–8.