International Journal of Advanced Smart Convergence Vol.8 No.2 77-87 (2019)
http://dx.doi.org/10.7236/IJASC.2019.8.2.77

IJASC 19-2-9

Piosk : A Practical Kiosk To Prevent Information Leakage

Suchul Lee[1], Sungil Lee[2], Hayoung Oh[3], and Seokmin Han[1]

[1]Dept. Computer Science and Information Engineering, Korea National University of Transportation, Korea.
[2]National Security Research Institute, Daejon, Korea.
[3]Ajou University, Suwon, Korea.
sclee@ut.ac.kr, silee@nsr.re.kr, hyoh79@gmail.com, seokmin.han@ut.ac.kr

Abstract

One of important concerns in information security is to control information flow. It is whether to protect confidential information from being leaked, or to protect trusted information from being tainted. In this paper, we present `Piosk` *(**P**hysical blockage of **I**nformation flow Ki**osk**) that addresses both the problems practically.* `Piosk` *can forestall and prevent the leakage of information, and defend inner tangible assets against a variety of malwares as well. When a visitor who carries a re-writable portable storage device, must insert the device into* `Piosk` *installed next to the security gate. Then,* `Piosk` *scans the device at the very moment, and detects & repairs malicious codes that might be exist. After that,* `Piosk` *writes the contents (including sanitized ones) on a new read-only portable device such as a compact disk. By doing so, the leakage of internal information through both insiders and outsiders can be prevented physically. We have designed and prototyped* `Piosk`*. The experimental verification of the* `Piosk` *prototype implementation reveals that,* `Piosk` *can accurately detect every malware at the same detection level as Virus Total and effectively prevent the leakage of internal information. In addition, we compare* `Piosk` *with the state-of-the-art methods and describe the special advantages of* `Piosk` *over existing methods.*

Keywords: *Information flow control, Information leakage, Malwares detection, Kiosk, Physical security*

1. Introduction

A fundamental concern in information security, especially for institutions, e.g., universities, research institutes, and governmental organizations like Korean electric power corporation (KEPCO) [1] and companies, is to control information flow. It is whether to protect confidential information from being leaked, or to protect trusted information from being tainted. They are also responsible for securing their tangible assets such as Enterprise Resource Planning (ERP) system, servers, networked devices, etc. Due to the recent advancement of Information Communication Technology (ICT), the protection of intangible assets such as intellectual property right (IPR) has become particularly important.

Various techniques that address this problem have been proposed. For example, Digital Right Management (DRM) [2] and/or security papers [3] are widely used as countermeasures against the leakage of internal information. There are many more commercial solutions for control the information flow in reverse direction.

Manuscript Received: Apr. 2, 2019 / Revised: Apr. 9, 2019 / Accepted: Apr. 17, 2019
Corresponding Author: seokmin.han@ut.ac.kr
Tel: +82-31-460-0587
Dept. of Computer Science and Information Engineering, Korea National University of Transportation, Korea

Internal assets should be protected from external malwares such as Ransomwares [4], Advanced Persistent Threats (APTs) [5], Trojan horses [6], etc. In general, when internal assets are connected to the Internet, it can be defended through commercially available anti-virus programs such as Avast [7] and V3 [8]. Moreover, there are many in-line malicious network-based packet detectors. They include Intrusion Prevention/Detection System (IPS/IDS), firewalls [9], unified threat managements (UTMs) [10]. Typically, they are used in conjunction with aforementioned anti-virus programs which mainly focus on private security.

In physical security, security gates [22] and security papers [3] are already commercially available because they are quite practical and cost effective. The security gate checks whether the data storage device such as a USB flash disk is carried in or out by the personnel manually entering and leaving the company. One can naturally claim that clever insiders who are aware of the vulnerability of their own security gate may intentionally bypass the gates carrying a USB flash disk intentionally, resulting in the leakage of internal information. Another example of information leakage is as follows. External system administrators, e.g., employees of a system integration company, S/W outsourcing, etc., might carry their lap tops or USB flash disks to do their jobs. There is a possibility that internal data is leaked through that USB flash disk. Even, the USB flash disk might been infected with a zero-day malware previously (whether the owner of a USB is aware of the infection in priori). The owner of the USB flash disk may enter the company, which is an intrinsic potential risk of infection from malwares and/or the leakage of information.

In this paper, we present `Piosk` (**P**hysical blockage of **I**nformation flow Ki**osk**) that can forestall and prevent the leakage of information. `Piosk` can also defend inner tangible assets against a variety of malwares. A key idea for `Piosk` is simple. `Piosk` is located next to the security gate (precisely, outside the gate), and plays a role. Its core functions are; (i) when a USB flash disk is connected to it, it scans the USB flash disk at the very moment to detect and repair malicious codes that might be exist. (ii) it burns a new read-only CD that contains the copy of (sanitized) contents of the USB flash disk, e.g., source code, S/W programs, documents, and provides the CD to whom passing through the security gate. By doing so, organizations which install `Piosk` can fundamentally prevent the leakage of internal information through both insiders and outsiders. This is because the CD has a read-only property. `Piosk` also can be a basis for enacting an enhanced policy on internal information leakage.

The main contributions of this paper are the following:

- We present a simple idea that can fundamentally prevent the leakage of internal information. To the best of our knowledge, burning a new read-only CD that contains the copy of USB contents to prevent the leakage of internal information is the first work.
- We design and prototype `Piosk`. Specifically, the implementation of `Piosk` including H/W, S/W (+ GUI) is done, as we will show in Section 5. This work is supported by National Security Research Institutes (NSR), and we are now commercializing `Piosk`.
- We perform the verification of `Piosk` by our real `Piosk` implementation. The verification results confirm that `Piosk` can accurately detect every malware at the same detection level as Virus Total [11] while running in off-line mode, as we will show in Section 3. In addition, we compare `Piosk` with the state-of-the-art methods and describe the special advantages of `Piosk` versus existing methods in Section 4.2. The results also confirm that `Piosk` can effectively prevent the leakage of internal information, in the view of physical security.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the design of `Piosk`. Section 4 provides some practical use cases of `Piosk` and compares `Piosk` with the state of the art methods. Section 5 details the implementation of `Piosk`. Section 6 discusses various practical issues. Finally, Section 7 concludes this paper.

2. Related work

There have been many attempts to control information flow. Theoretically, information flow in both directions has been mathematically modeled and analyzed in [13-15]. In [16], Side-channel Vulnerability Factor (SVF) was proposed as a metric for measuring information leakage. SVF can answer to this question, "how much vulnerable is a certain vulnerability?" where the vulnerability is exploited in side channel attacks. As emerging cloud-based IT technologies became popular, the studies for addressing the problem of information leakage are re-visited. There have been some practical proposals [12, 17] to this problem in the domain of information security for cloud-based systems. The control of information flow in reverse direction, boils down to the acquisition of a method that efficiently detects (and cures) malwares. It is well-known that the polymorphism and meta-morphism of malwares have eventually made this problem difficult [18]. Dynamic malware detection techniques have been proposed in [19, 21]. In [19, 21], a sandbox technique is used. The sandbox is typically made up of virtual machine(s). Securing one's assets in the network level was also a popular area of security research. Modern IPS/IDSes typically employ a rule- or signature-based technology to detect various attacks. An IPS/IDS rule that contains a proper signature for malicious traffic is generally called well-written. This rule is typically written by security administrators, e.g., Computer Emergency Response Team (CERT), however, [20] attempted to automate this cumbersome task using Latent Dirichlet Allocation (LDA).

In physical security, security gates [22] and security papers [3] are already commercially available. Perhaps the most relevant to our study is [23, 24]. Specifically, MetaDefender [23] is a cyber security platform for preventing and detecting cyber security threats on multiple data channels. MetaDefender scans a portable storage device through a kiosk to block malicious code coming into the internal network, or to define, manage and control normal files to be imported. MetaDefender's core function is to scan files in portable devices using a variety of on-line malware detection engines. In addition to the core function of MetaDefender, it can perform file blocking, conversion, isolation, etc., through the file filtering function. However our proposed system, namely Piosk, has several additional advantages. The most important advantage is that, in addition to the basic function of a kiosk system designed for information security, e.g., the protection of internal resources of a company or an organization against malwares, Piosk can physically prevent internal information leakage.

3. Piosk Design

In this section, we present the proposed method, Piosk. We first outline the overall framework of Piosk, and then describe each module of Piosk in more detail.

3.1 Overview of Piosk

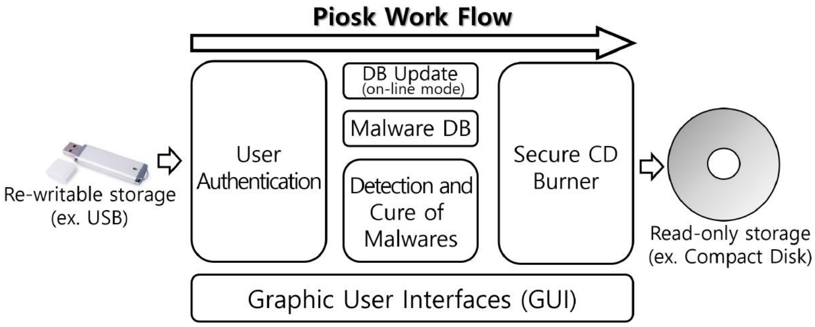


Figure 1. Piosk's overall architecture.

As shown in Fig. 1, Piosk consists of six parts; (i) input, (ii) user authentication, (iii) detection & cure of malwares, (iv) output, (v) secure read-only CD generator, and (vi) output. One who needs to pass through the security gate with re-writable storage devices, e.g., USB flash disks, re-writable CD, etc., must insert the storage device into Piosk. Then, Piosk scans the storage device through the database of malwares incorporated in Piosk in advance. This database of malwares must be up-to-date, which in general can be achieved highly credible web-based malware database such as Virus Total. Of course, this connection between the Internet and Piosk should be on-demand manner. Piosk cures the content previously stored in the device, and the (sanitized) secure data is written to read-only devices, e.g., CD. Now, he/she can go inside with only CD, with an approval from the security system. All of these Piosk usages are written in the form of a log trace.

3.2 User Authentication

The user authentication (UA) module consists of three parts; the authentication of (i) new users, (ii) existing users, and (iii) the registered user management. The UA module can be regarded as a front-line safety device for Piosk. Although there are many cases in which internal information leakage occurs by people who know vulnerabilities intensely, the UA module can be a preemptive safety module for outsiders.

UA for a new user: A user who firstly uses Piosk requests the administrator to create an ID Piosk through the "new user authentication" function. If requested, administrators will (or not) permit the user registration remotely through the administrative system which is always connected to Piosk. Then, user information is stored in the internal DB.

UA for an existing user: A user who has already been registered is authenticated through the "authentication" function of Piosk. This is normally done the user's ID and its password. Additionally, this also can be done through Radio Frequency Identification (RFID) and/or the fingerprint recognition.

Registered user management: Administrators can manage (modify, delete) information of pre-registered users through the administrative system.

3.3 Detection & Cure of Malwares

On-line mode: Malwares exist in various forms. Therefore, customized detection and defense strategies should be used for each type of malware. For example, if malwares that are spreading over a network, they should be detected and defended through network-based detectors such as IPSes. Encryption and/or mutation of malwares make detection difficult significantly. Piosk can leverage Virus Total, one of the most reputable web-based malware DBs, in which case the Piosk must be connected to the Internet. We refer to this operating mode as "on-line" mode. To minimize malicious code infection through Internet connection, Piosk operates in this mode only to update inner malware DB.

File type classification: All files in a portable storage device are classified according to the policy set by the administrator and given priority to the user. This is because different malware (file) detection and defense policies must be applied as described above. A Piosk administrator can preset the files for each type with the "Policy Setting" function through the administrative system. A Piosk user can also classify each file in the device directly, which requires the permission of the administrator. Malware scanning through various methods: The files stored on the storage device are subjected to the file verification according to the type as shown below. Each verification method is as follows:

Basic file scanning: Basically, all files on a portable storage device are subjected to anti-virus scanning. Vaccine software programs, e.g., V3 [8] and Avast [7] can be used, where each software program should run in its own virtual machine, and should not be conflict with each other. Additionally, Piosk uploads each file in the device to Virus Total [11], then Virus Total checks hash values of all files whether to check the files are tampered with or not. This link between Piosk and Virus Total requires a network connection, which is performed on-demand manner.

Integrity verification: `Piosk` performs advanced integrity verification for some types of files. For example, firmware files are a typical example that is necessary for advanced integrity verification. This verification can be performed through the reference data set (RDS). RDS is a repository that holds hash values of normal files, SCADA manufacturers disclose the hash value of the firmware patch file through their web-sites. These new hash values are maintained securely and up-to-date in the inner `Piosk` DB. Operating system (installed in `Piosk`) patch files should be verified as well.

Detection of malicious scripts in document files: There are known and unknown, a.k.a., zero-day, vulnerabilities in document files such as PDF [25], HWP [26], and MS office (DOC and PPT) files [27]. `Piosk` checks to see if any hidden scripts or macros exist in the document file.

Dynamic executable file verification: Lastly, `Piosk` supports to analyze files by observing the behavior of executable files. This function is implemented by a technique called a sandbox [28], which runs (possibly malicious) executable files, e.g., *.exe and *.dll, in a virtual machine that is not connected to the Internet.

3.4 Secure contents delivery

`Piosk` writes (sanitized) contents to a read-only device so that they can be delivered into the inner network. There are a typical example: Assume that `Piosk` receives file α, file β, and file γ as verification targets and file α and file β pass the verification. Then, only file α and file β are written to the read-only device, and file γ is excluded typically. If file γ can be sanitized by one of the vaccines incorporated in `Piosk`, sanitized file $\overline{\gamma}$ can be written as well. This depends on the `Piosk` operation policy.

4. Evaluation

4.1 Case study

Since the vulnerability of the installed operating system (by default, MS Windows 10) in `Piosk` is widely known, `Piosk` basically operates in "internal off-line mode" which is basically used without internet connection, as shown in Fig. 2. This connection between `Piosk` and the Internet is activated only while updating the malware DB of `Piosk` (described as "temporal network link").

Consider the following situation. Company A uses the ERP solution developed by system integration company B. We also assume that this ERP system operates without the connection to the Internet. An occasional failure of this ERP system may occur, or the ERP system may need to be updated. Then, engineers (referred to as a visitor later in this section) who work for company B should visit company A to fix the ERP-related problems.

Step 1: UA is in progress before inserting the re-writable portable storage device (e.g., USB) into `Piosk`, as described in Subsection 3.2. The verification level for the files in the USB can be flexibly adjusted according to the result of UA. The visitor now can insert the re-writable portable storage device into `Piosk`. If the device is inserted, `Piosk` first classifies the files stored in the device according to the file type, then verifies each file through various methods corresponding to the file type, as described in Subsection 3.3.

Step 2: After the insertion of the device, he/she can pass through the security gate. In general, security gates check whether storage devices such as lap-tops and other storage devices (except the device inserted into `Piosk`) are brought or not, which is however out of scope of this paper.

Step 3: Now, only verified and sanitized files are recorded on a read-only CD. A visitor who has passed the gate can receive that CD and enter inside. The files in the CD are secure (at the same level of `Piosk`'s scanning programs) and will not infect the internal system when connected to the internal network. Also, since it is a read-only device, internal information cannot be exported through that device fundamentally.

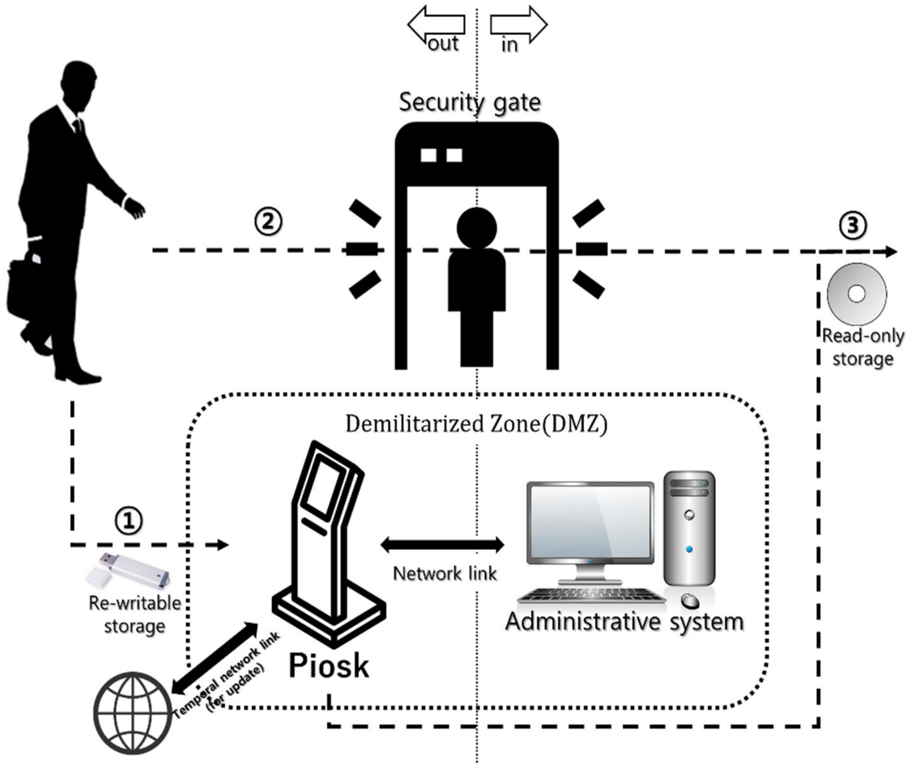


Figure 2. Piosk use case

4.2 Comparison of Piosk with the state of the art methods

This section describes the advantages of the proposed system through comparison with other commercially available security kiosks [23–25]. As described in Table 1, our special function of transferring data to the internal network using a device that cannot be re-written such as a CD is unique. With this function, a portable storage device entering the inside can copy data to the internal network and/or system, however inner data cannot be brought through the de- vice because it cannot be writable. It is the key for the function of preventing the leakage of internal information, which exists only in Piosk.

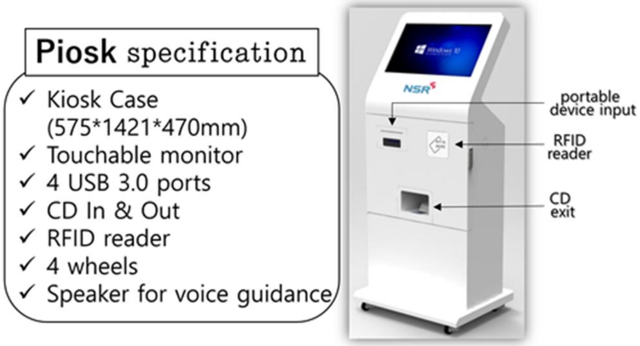
One of the other features of a security kiosk is the functionality to detect and sanitize malwares contained on portable storage devices. Generally, two methods can be applied for malware detection. The first is static analysis and the second is dynamic analysis. Static analysis is a technique for detecting malware through the presence of specific bit patterns in malware, typically referred as to signatures or rules. IPSes and computer virus vaccines are typical examples. Typically, the performance of these malware detectors is determined by how big/comprehensive the malware signature DB is. Security kiosks including Piosk incorporate as many malware signature DBs as possible to enhance malware detection performance. Perhaps, the most reputable and widely used malware signature DB is Virus Total, which is adopted only in Piosk. Dynamic analysis is a way to actually run an executable file and detect it as malware, if it performs malicious activity. Obviously, this execution should not be occurred on a kiosk. Therefore, Piosk adopts virtual environment such as a sandbox, which is only supported in by Piosk.

Finally, one of the important functions is whether or not the security method is applied to the kiosk system itself. The key of this function is whether the kiosk is physically connected to the Internet during operation. MetaDefender [23] and ODIX kiosk [25] support on-line mode operation, which is also available in Piosk. Piosk is designed to be free of OS-dependency at the design stage, as it will be discussed in Section 6, the default OS installed in Piosk is MS Windows, but a secure OS such as Gooroom can be installed as well, and we have verified the Piosk operation in Gooroom [34]. ODIX kiosk [25] has no HDD/SSD disk. This means that ODIX kiosk is not likely to be infected by malwares. However, `Piosk` has a SSD disk that is required for virtual environment, which is used in dynamic analysis as described above.

Table 1 Comparison of `Piosk` with other security kiosks

Functionality	`Piosk`	MetaDefender [23]	California Kiosk [24]	ODIX [25]
Prevention of information leakage	O	X	X	X
Data sanitization	O	partially	partially	partially
- Static analysis	O	O	O	O
- Dynamic analysis (implemented via VMs)	O	X	X	X
Self-defending strategy	O	partially	X	O
- Off-line mode operation	O	partially	X	O
- Secure OS (default OS)	O (Windows)	X (Windows)	X (Windows)	X (Linux)
- No HDD system	X			O
- Dedicated management system support	O	X	X	O

5. Implementation of `Piosk`



Piosk specification

- ✓ Kiosk Case (575*1421*470mm)
- ✓ Touchable monitor
- ✓ 4 USB 3.0 ports
- ✓ CD In & Out
- ✓ RFID reader
- ✓ 4 wheels
- ✓ Speaker for voice guidance

Figure 3. `Piosk` prototype implementation and its detailed specification.

We have developed the `Piosk` prototype over two years in NSR. We have implemented `Piosk` and its administrative system by using commercial computer parts such as CPU, Memory, M/B, etc. Some of them are customized as appropriate for `Piosk`. We developed both the `Piosk` and its administrative applications, which basically run on Windows 10. The implementation de- tails and the picture of `Piosk` are shown in Fig 3. We have developed an administrator software and `Piosk`'s user friendly graphical user interface (GUI). Major functions of the administrative system are as follow:

① A task list shows the list of TODO.

② A file list shows the list of file information corresponding to each task (shown in Fig. 4-(c)).

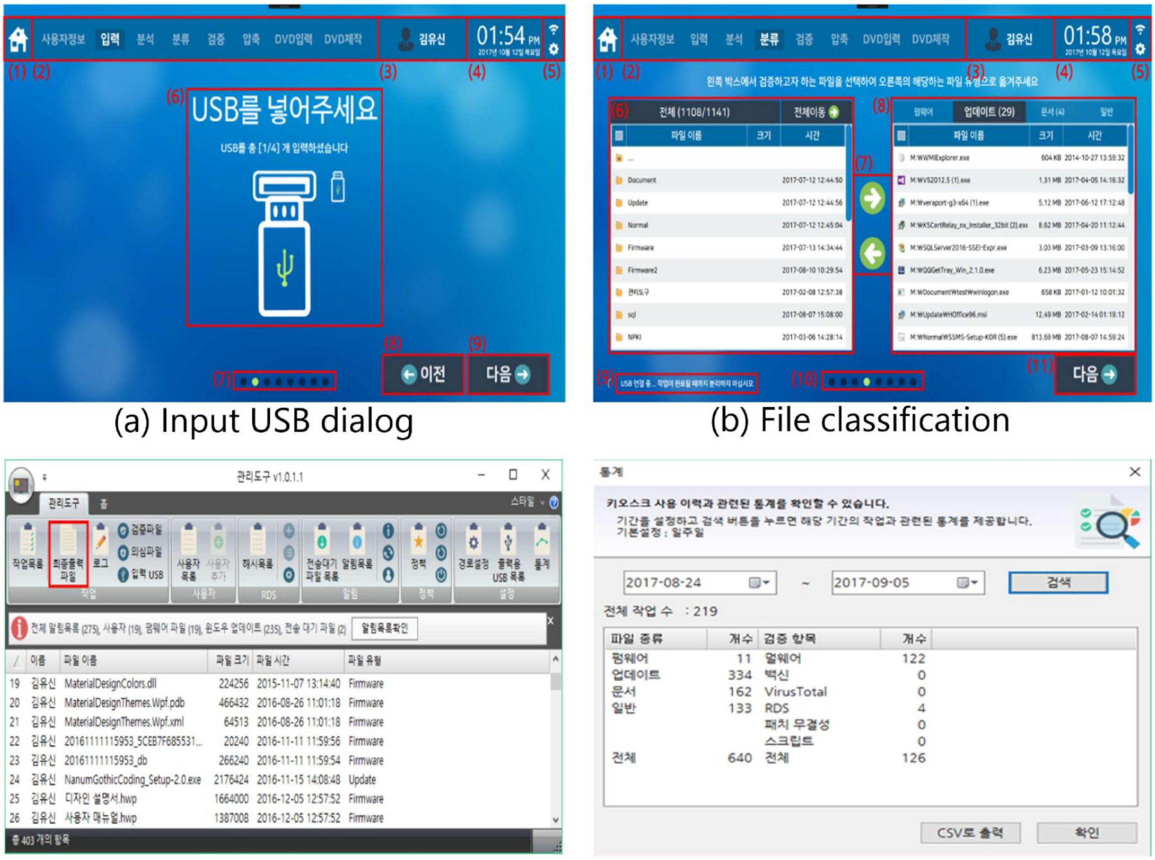
③ A log system summarizes every usage of `Piosk`.

④ A user list shows the list of registered users. There is another dialog for adding a new user.

⑤ Malware DB: RDS hash values, Virus Total hash values (by default, off-line), etc.

⑥ Important events are noticed so that an administrator can easily locate a problem. Events of these type include UA regarding errors, malware detection, policy modification, and so on.

⑦ Finally, there is a dialog to set/modify policies applied in `Piosk`



USB를 넣어주세요

USB를 총 [1/4] 개 입력하셨습니다

(a) Input USB dialog

왼쪽 박스에서 검증하고자 하는 파일을 선택하여 오른쪽의 해당하는 파일 유형으로 옮겨주세요

파일 이름	크기	시간
..		
Document		2017-07-12 12:44:50
Update		2017-07-12 12:44:56
Normal		2017-07-12 12:45:04
Firmware		2017-07-13 14:34:44
Firmware2		2017-08-10 10:29:54
관리도구		2017-02-08 12:57:38
sql		2017-08-07 15:08:00
NPKI		2017-03-06 14:28:14

파일 이름	크기	시간
M:₩WMIExplorer.exe	604 KB	2014-10-27 13:59:32
M:₩VS2012.5 (1).exe	1.31 MB	2017-04-05 14:16:32
M:₩veraport-g3-x64 (1).exe	5.12 MB	2017-06-12 17:12:48
M:₩KSCertRelay_nx_Installer_32bit (2).exe	8.62 MB	2017-04-20 11:12:44
M:₩SQLServer2016-SSEI-Expr.exe	3.03 MB	2017-03-09 13:16:00
M:₩QQGetTray_Win_2.1.0.exe	6.23 MB	2017-05-23 15:14:52
M:₩Document₩test₩winlogon.exe	658 KB	2017-01-12 10:01:32
M:₩Update₩HOffice96.msi	12.49 MB	2017-02-14 01:19:12
M:₩Normal₩SSMS-Setup-KOR (5).exe	813.69 MB	2017-08-07 14:59:24

USB 연결 중... 작업이 완료될 때까지 분리하지 마십시오

(b) File classification

전체 알림목록 (275), 사용자 (19), 펌웨어 파일 (19), 윈도우 업데이트 (235), 전송 대기 파일 (2)

	이름	파일 이름	파일 크기	파일 시간	파일 유형
19	김유신	MaterialDesignColors.dll	224256	2015-11-07 13:14:40	Firmware
20	김유신	MaterialDesignThemes.Wpf.pdb	466432	2016-08-26 11:01:18	Firmware
21	김유신	MaterialDesignThemes.Wpf.xml	64513	2016-08-26 11:01:18	Firmware
22	김유신	20161111115953_5CEB7F685531...	20240	2016-11-11 11:59:56	Firmware
23	김유신	20161111115953_db	266240	2016-11-11 11:59:54	Firmware
24	김유신	NanumGothicCoding_Setup-2.0.exe	2176424	2016-11-15 14:08:48	Update
25	김유신	디자인 설명서.hwp	1664000	2016-12-05 12:57:52	Firmware
26	김유신	사용자 매뉴얼.hwp	1387008	2016-12-05 12:57:52	Firmware

총 403 개의 항목

키오스크 사용 이력과 관련된 통계를 확인할 수 있습니다.

기간을 설정하고 검색 버튼을 누르면 해당 기간의 작업과 관련된 통계를 제공합니다.
기본설정 : 일주일

2017-08-24 ~ 2017-09-05

전체 작업 수 : 219

파일 종류	개수	검증 항목	개수
펌웨어	11	멀웨어	122
업데이트	334	백신	0
문서	162	VirusTotal	0
일반	133	RDS	4
		패치 무결성	0
		스크립트	0
전체	640	전체	126

Figure 4. Four exemplary GUI screenshots for `Piosk` and its administrative system.

Fig. 4 shows some representative GUI screens. The GUI is basically implemented to support Korean language, however we are now commercializing `Piosk`, and the commercialized `Piosk` product will support English language. Fig. 4-(a) shows a `Piosk` screen that guides a visitor to insert USB flash disks. The visitors can insert up to four USB flash disks at the same time. These USBs will be scanned and the verified. The verified files (including sanitized ones) are written to the read-only CD.

Fig. 4-(b) shows a `Piosk` screen that classifies files in the USB flash disks. Firstly, `Piosk` automatically classifies each file based on pre-defined classification rules. Then, several additional classification or modification of existing classification can be done manually by a user, which should be confirmed through the administrative system, as shown in Fig. 4-(c).

Fig. 4-(d) shows one of the statistical information for `Piosk` jobs where this statistic is summarized from logging data stored in `Piosk`.

6. Discussion

We have described the proposed physical blockage of information flow kiosk, `Piosk`, and have confirmed, by real implementation, that the control of information flow in both directions performs at the same level of reputable web-based malware DB (Virus Total), commercial vaccines (e.g., V3 and Avast), etc. In this section, we discuss several practical but unavoidable security issues while using `Piosk`.

6.1 Off-line mode operation for Virus Total

It is generally considered impossible to run an Internet- based malware scanner such as Virus Total in off-line mode. Nevertheless, there have been many efforts to make this possible [30–32]. Ultimately, this problem comes down to the problem of building an internal mal- ware database like a small Virus Total. If Piosk is connected to the Internet all the time, various vulnerabilities (including zero-days, unfortunately) can be a target. Or we can make Piosk to be connected to the Inter- net, only when internal database update is in progress. On the other hand, there are many advantages of off- line mode operation. If a security problem of the link between Piosk and the internal network can be addressed, it is not necessary to transfer the contents through a read-only device such as a CD. A simple way to solve this problem is to install a uni-directional switch between Piosk and the internal network.

6.2 Vulnerability within MS Windows

There are known and unknown vulnerabilities in Microsoft Windows 10 [33]. Security patches are being announced from time to time. This is not because Microsoft Windows is very vulnerable. We claim that the most reasonable reason is that, hackers generally target MS Windows because it is the most popular operating system. Unfortunately, there is no perfect S/W in the view of security. Another alternative is to use a different operating system that is more secure. We have also developed the Piosk prototype that runs on Gooroom OS [34].

6.3 Advantages of Piosk for companies

By installing Piosk next to the security gate, the company can apply enhanced policies as well as physically block the leakage of internal critical information. For example, the company that installed Piosk can enact a rule that cannot import a USB flash disk internally. This is because digital information from outside sources can be inspected and exported through a CD. In addition, it is also possible to prohibit the insertion of a general USB disk, except a secure USB [35], in the internal network. Of course, the leakage of internal information through the network should be monitored by other security solutions in parallel with Piosk, which is out of scope for this paper.

7. Conclusion

We proposed Piosk, a practical kiosk that can forestall and prevent the leakage of information. We first presented a simple idea that could fundamentally prevent the leakage of internal information. And we designed and prototyped Piosk at the commercial product level. We also developed a user-friendly GUI that might facilitate the usage of Piosk for both users and system administrators. We performed the verification of Piosk by our real Piosk implementation. The verification results confirmed that Piosk can accurately detect every malware at the same detection level as Virus Total. The results also confirmed that Piosk can effectively pre- vent the leakage of internal information, in the view of physical security. We are now commercializing Piosk.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2017R1C1B5017028) and by Korea National University of Transportation in 2019.

References

[1] Korea Electric Power Corporation (KEPCO). http://home.kepco.co.kr

[2] Talukder, K. Asoke, and M. Chaitanya. Architecting secure software systems. Auerbach publications, 2008.

[3] Security Paper Limited. http://www.security-papers.com/

[4] Ransomware. https://en.wikipedia.org/wiki/ Ransomware

[5] Advanced persistent threat https://en.wikipedia.org/ wiki/Advanced_persistent_threat

[6] Trojan horse. https://en.wikipedia.org/wiki/Trojan_ Horse

[7] Free antivirus protection that never quits. https://www.avast.com

[8] V3 Internet Security, Greater Business comes with Greater Security. http://global.ahnlab.com

[9] A. Wool, "A quantitative study of firewall configuration errors," Computer, vol. 37, no. 6, pp. 62–67, 2004. DOI: https://doi.org/10.1109/MC.2004.2

[10] Y. Qi, B. Yang, B. Xu, and J. Li, "Towards system-level optimization for high performance unified threat management", IEEE 3rd International Conference on Networking and Services 2007. DOI: https://doi.org/10.1109/ICNS.2007.126

[11] Virus Total. https://www.virustotal.com/

[12] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds", ACM conference on Computer and communications security 2009. DOI: https://doi.org/10.1145/1653662.1653687

[13] J. McLean, "Security models and information flow", IEEE Symposium on Security and Privacy, 1990. DOI: https://doi.org/10.1109/RISP.1990.63849

[14] B. Ko¨pf and D. Basin, "An information-theoretic model for adaptive side-channel attacks", 14th ACM Conference on Computer and Communications Security. DOI: https://doi.org/10.1145/1315245.1315282

[15] S. Mario, C. Kostas, P. Catuscia, and S. Geoffrey, "Measuring Information Leakage Using Generalized Gain Functions", IEEE 25th Computer Security Foundations Symposium 2012. DOI: https://doi.org/10.1109/CSF.2012.26

[16] J. Demme, R. Martin, A. Waksman, and S. Sethumadhavan, "Side-channel vulnerability factor: A metric for measuring information leakage", ACM SIGARCH Computer Architecture News 2012, 40(3), 106-117. DOI: https://doi.org/10.1109/ISCA.2012.6237010

[17] H. Takabi, J. B. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments" IEEE Symposium on Security and Privacy 2010. DOI: https://doi.org/10.1109/MSP.2010.186

[18] A. Sharma and K. S. Sanjay, "Evolution and detection of polymorphic and metamorphic malwares: A survey." arXiv preprint arXiv: 1406.7061 (2014).

[19] C. Willems, H. Thorsten, and F. Felix, "Toward automated dynamic malware analysis using cwsandbox", IEEE Symposium on Security and Privacy 2007. DOI: https://doi.org/10.1109/MSP.2007.45

[20] S. Lee, S. Kim, S. Lee, H. Yoon, D. Lee, J. Choi, and J. Lee, "LARGen: automatic signature generation for Mal wares using latent Dirichlet allocation", IEEE Transactions on Dependable and Secure Computing Vol.15 No.5 2018. DOI: https://doi.org/10.1109/TDSC.2016.2609907

[21] A. Dinaburg, P. Royal, M. Sharif, and W. Lee, "Ether: malware analysis via hardware virtualization extensions", 15th ACM conference on Computer and communications security 2008. DOI: https://doi.org/10.1145/1455770.1455779

[22] R. E. Knoedler, T. B. Freese, R. M. Parker, and J. E. Janicke, "Security gate with walk through feature", U.S. Patent No. 5,272,840 (1993). Washington, DC: U.S. Patent and Trademark Office.

[23] MetaDefender, Opswat, https://www.opswat.com/products/metadefender

[24] California Cyber Security Kiosk, https://www.olea.com/product/california-cyber-security-kiosk/

[25] ODIX kiosk, File sanitization system, https://odix.com/odix-kiosk/

[26] D. Maiorca, G. Giacinto, and C. Igino, "Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection", ACM SIGSAC symposium on Information, computer and communications security 2013.

DOI: https://doi.org/10.1145/2484313.2484327

[27] Z. Wang, Z. Tang, K. Zhou, R. Zhang, Z. Qi, and H. Guan, "DsVD: an effective low-overhead dynamic soft- ware vulnerability discoverer", IEEE International Symposium on Autonomous Decentralized Systems 2011. DOI: https://doi.org/10.1109/ISADS.2011.56

[28] T. Schreck, B. Stefan, and G. Jan, "BISSAM: Automatic vulnerability identification of office documents", Springer International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment 2012. DOI: https://doi.org/10.1007/978-3-642-37300-8_12

[29] L. Gong, M. Mueller, H. Prafullchandra, and R. Schemers, "Going beyond the sandbox: An overview of the new security architecture in the Java development kit 1.2", USENIX Symposium on Internet Technologies and Systems 1997.

[30] Sigcheck v2.60. Microsoft, https://docs.microsoft.com/ko-kr/sysinternals/downloads/sigcheck

[31] Scan Virus Total offline to preserve privacy, GitHub. https://github.com/teeknofil/Virus-Total-Never-Analyzed

[32] VirusTotal offline analysis, GitHub, https://github.com/cuckoosandbox/cuckoo/issues/2052

[33] CVE Details, The ultimate security vulnerability datasource. https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-32238/Microsoft-Windows-10.html

[34] Gooroom OS. https://www.gooroom.kr/

[35] S. F. Fruhauf and T. Jerome, "Secure universal serial bus (USB) storage device and method." U.S. Patent No. 8,528,096. 3 Sep. 2013.