PROACTIVE INSIDER-THREAT DETECTION
Against Confidentiality in Sensitive Pervasive Applications

Joon S. Park, Jaeho Yim and Jason Hallahan
The Laboratory for Applied Information Security Technology (LAIST), School of Information Studies (iSchool)
Syracuse University, Syracuse, NY 13244-4100, U.S.A.

Keywords: Insider threats, Monitoring.

Abstract: The primary objective of this research is to mitigate insider threats against sensitive information stored in an organization's computer system, using dynamic forensic mechanisms to detect insiders' malicious activities. Among various types of insider threats, which may break confidentiality, integrity, or availability, this research is focused on the violations of confidentiality with privilege misuse or escalation in sensitive applications. We identify insider-threat scenarios and then describe how to detect each threat scenario by analyzing the primitive user activities, we implement our detection mechanisms by extending the capabilities of existing software packages. Since our approach can proactively detect the insider's malicious behaviors before the malicious action is finished, we can prevent the possible damage proactively. In this particular paper the primary sources for our implementation are from the Windows file system activities, the Windows Registry, the Windows Clipboard system, and printer event logs and reports. However, we believe our approaches for countering insider threats can be also applied to other computing environments.

1 INTRODUCTION

One of the most serious modern-day threats against sensitive computer systems, networks, and data is the insider threat (Brackney et al., 2004; CSI, 2007; Hayden, 1999; Neumann, 1999; Moore et al., 2008; Keeney et al., 2005). An insider is an individual who possess a certain level of access, privilege and trust within an organization due to their position, role, or task within that organization. Whilst an outsider must gain access and privilege to a system using social engineering or some other method in order to damage that system, an insider generally inherits those capabilities by default. At this point the only thing that separates an insider employee from an outsider threat is their actions and intentions. Modern-day computer defenses range from firewalls to intrusion detection systems (IDS) to access control lists (ACL) but their primary focus of mitigating the outsider threat remains the same. Efforts to incorporate these same defenses against insiders have thus far been fruitless (Anderson, 1999; Bishop, 2005; Chinchani et al., 2005; Apap et al., 2001; Park et al., 2004; Pramanik, 2004; Renesse, 2003). A great need still exists for a real-time, lightweight detection and mitigation system for insider misuse.

An insider threat can damage an organization in various ways, and often that damage in dollars and reputation is permanent, such as when an attacker exposes a bank database of credit card numbers. Traditional forensics technologies, which help companies identify and prosecute a criminal offender after the fact, is often of little consolation. Applied digital forensics, which monitors and audits computer systems in realtime can be used to strike against insider misuse. However, applying digital forensics in real-time is a daunting task, since there are so many files and processes to monitor, and the state of an average computer system or network is changing hundreds and even thousands of times per minute.

Before any real-time digital forensics can be applied to a system, there must be a clear determination of internal security controls, normal system behavior, as well as files, processes, and behaviors that deserve the highest scrutiny. For instance, file deletion can be a benign act, but could also signal misuse, and should be monitored. System registries are often modified by software programs and system processes, but user modification of these

S. Park J., Yim J. and Hallahan J. (2009).
PROACTIVE INSIDER-THREAT DETECTION - Against Confidentiality in Sensitive Pervasive Applications.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 393-398
DOI: 10.5220/0002004203930398

files can signal suspicious behavior, such as the concealment of malicious activity. State changes of files with the attributes hidden or readonly, as well as the creation of these files, can also be considered suspicious depending on the context. The creation or modification of alternate data streams can also signal misuse.

The primary objective of the research is to mitigate insider threats against sensitive information stored in an organization's computer system, using dynamic forensic mechanisms to detect insiders' malicious activities. Among various types of insider threats, which may break confidentiality, integrity, or availability, this research is focused on the violations of confidentiality with privilege misuse or escalation in sensitive applications. In particular, we identify five generic threat-scenarios against confidentiality. We then describe how to detect each threat scenario by analyzing the insider's activities in terms of Copy, Rename, Print, and Paste. Finally, we implement our detection mechanisms by extending the capabilities of existing software packages in Windows environments. Since our approach can proactively detect insider threats before the malicious action is finished, we can prevent the damage proactively, while most of existing approaches detect the malicious action after the damage.

2 RELATED WORK

In this section we describe the related works that we use to implement our proposed ideas. We could develop a brand new system based on our approach, but we decided to use existing packages with extension by considering the cost-effectiveness, reusability, compatibility, and extensibility. The details about how we use these existing approaches are described in the following sections.

2.1 Windows Registry

The Windows Registry (Honeycutt, 2002) is a hierarchical database that stores system parameters, security information, program configuration settings and user profiles. The Windows operating system and applications query the values of specific registry keys, dictating system operations as well as user environments. Registry keys and values are added to the database when new hardware, applications, users, and information are added to the system. The Windows Registry was introduced in its current form in Window 9x/ME, and has been used in all derivations and iterations of Microsoft Windows operating systems release since then, including the most recent release, Windows Vista. There are five root keys that cover different aspects of system operation, including HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, HKEY_USERS, HKEY_CURRENT_CONFIG. These components include the name of the system process querying the registry, the type of query, the actual registry key being accessed, the status of the query, and the resultant value, if any.

2.2 Process Monitor

Microsoft Process Monitor (Windows Sysinternals, 2008) is a system-monitoring tool to show real-time file system, Registry and process/thread activities for Microsoft Windows operating systems with NT Kernel 5.0 and above such as Windows 2000 (both workstation and server), Windows XP (both 32 and 64 bit), Windows Server 2003 (both 32 and 64 bit) and Windows Vista (both 32 and 64 bit). Process Monitor consists of three monitoring modules; file system, registry, and process/thread. File system monitoring displays file system activities for all Windows file system, including local storage and remote file systems. It also automatically detects the arrival of new file system devices and monitors them. Registry monitoring logs all registry operations and displays Registry path using conventional abbreviations for Registry root keys. The process monitoring tracks all process and thread creation and exit operations as well as DLL and device load operations. The software is currently provided by Windows Sysinternals (Windows Sysinternals, 2008), which was acquired by Microsoft in 2006.

2.3 Windows Clipboard Systems

The Windows The Windows Clipboard (Windows Clipboard, 2008) is a method or a set of functions and that enable applications to transfer data within the Windows environment. The Clipboard system is often confused with the Windows Clipboard Viewer (clipbrd.exe located in the %SystemRoot%\System32\), which is just an application included in Windows-NT architecture operating systems (i.e. Windows 2000, Windows XP, Windows Server 2003 and Windows Vista). The clipboard viewer displays the current content of the Clipboard system one at a time, which means it displays only the most recent one. The viewer supports only the standard formats; CF_BITMAP, CF_TEXT, CF_METAFILEPICT, and CF_ENHMETAFILE.).

2.4 Printer-Monitoring Tools

Printing (either hardcopy or softcopy) is one of the most common methods of unauthorized information leaking. Therefore, analyzing printer logs or reviewing printer event history is an effective measure to protect the confidentiality of the organization's sensitive information. However, there are no built-in such tools in any Windows operating systems. Therefore, in our research, we use a third-party printer-monitoring tool, SoftPerfect's Print Inspector (Print Inspector, 2008). This software is a print management and auditing tool especially designed for networked systems so that administrators can manage print jobs queued at any shared printer. In addition to the primary purpose of the software, it provides an auditing function for the printed document properties along with print job date and time, number of pages, name of the user who created the job, name of the computer from which the job was sent to the printer, and etc. All the collected data can be stored in a database system so that it can be used later for the statistical purpose.

3 INSIDER THREAT SCENARIOS

Although there are various kinds of insider threats and damages, according to the basic security properties, we can classify insider threats into the violations of confidentiality, integrity, and availability. Each property can be broken by insiders with their privilege misuse or privilege escalation. Table 1 summarizes the insider threats and their goals based on our classification.

Table 1: Insider Threats and Their Goals.

Security Property Violated	Privilege Misuse	Privilege Escalation
Confidentiality	Leaking sensitive information	Obtain ability to leak information
Integrity	Changing security level of files	Obtain ability to change integrity
Availability	Perform DOS attacks	Obtain ability to stop service

Unfortunately, we cannot simply assume that all the insiders will use their privilege in a legitimate way. Furthermore, technically, some operational environments have a built-in function (e.g., "RunAs" in Windows), which allows an installer to run with elevated privileges, Administrator. By exploiting this function, for instance, a regular user may run the installation process with the credentials of a system administrator. Actually, a malicious insider or attacker can exploit this vulnerability for privilege escalation.

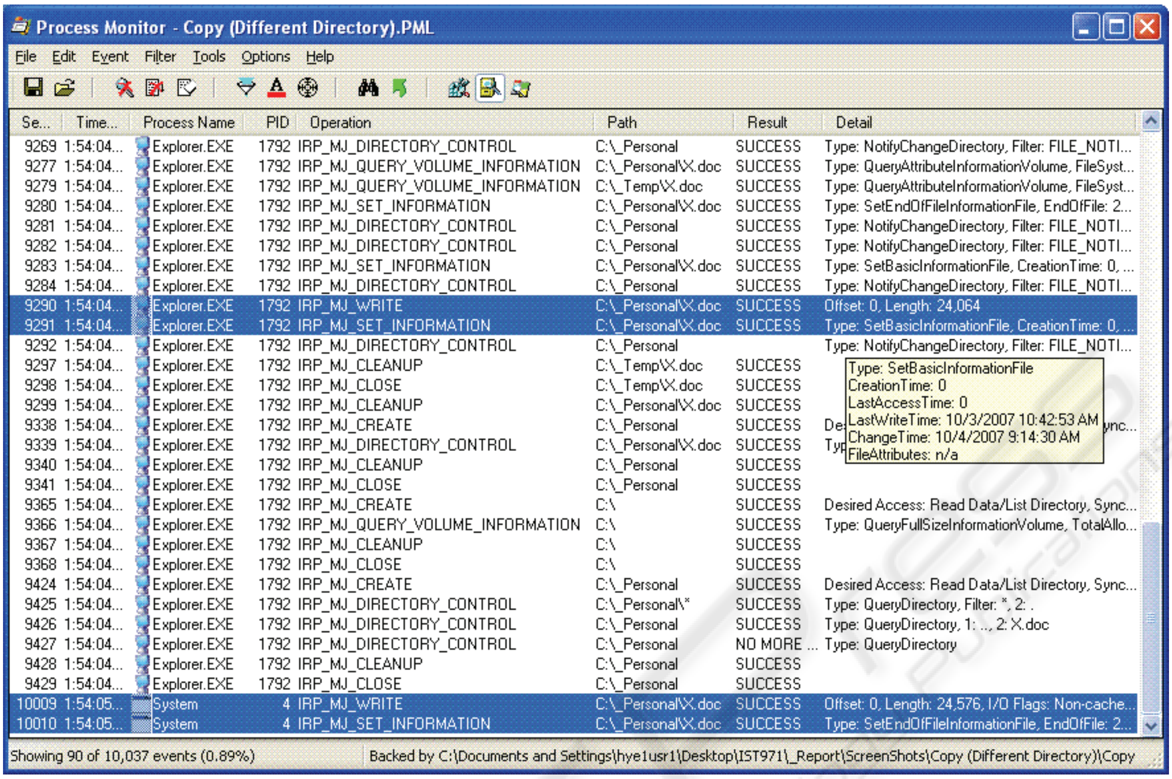
In this particular paper we focus on the violations of the confidentiality by the privilege misuse against organization's sensitive information. Even in this category, to a malicious insider, there are various ways to compromise the confidentiality of the protected resources. However, we hypothesize that all the malicious activities against confidentiality can be detected by analyzing the primitive user activities such as Copy, Rename, Print, Paste, and so on. We assume that direct file transfer to an outside machine (e.g., via FTP, HTTP, email attachments, etc.) can be detected and foiled by existing security mechanisms such as firewalls or IDS. In the followings sections we describe how we can detect each insider threat scenario and the implementation results based on our solutions. Since our approach can proactively detect the insider's malicious behaviors before the malicious action is finished, we can prevent the actual damages.

4 INSIDER THREAT DETECTION

4.1 Senenario#1

Copying a Sensitive File To an Unapproved Location. There are two common methods of making a file copy in Windows; using the Windows Explorer (this is different from Internet Explorer) and the Windows Command Prompt. In order to detect the insider's malicious behavior based on Secentrio#1 in Section 3, we can use the process monitor described in Section 2. Figure 1 is the screen shot of Process Monitor that shows the malicious insider makes a copy of the sensitive file labeled as "X.doc" in the "C:_Temp\" directory into "C:_Personal" directory using the Windows Explorer interface. We can run the same monitor using the Windows Command Prompt interface.

The highlighted information in the figure shows that the insider is tying to copy the sensitive file, X.doc, to another location. Basically, the core part of the insider's unauthorized file copy action is the same for both cases. The only difference is which process is handling the file copy action (either


Process Monitor - Copy (Different Directory).PML
File Edit Event Filter Tools Options Help

Se...	Time...	Process Name	PID	Operation	Path	Result	Detail
9269	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal	SUCCESS	Type: NotifyChangeDirectory, Filter: FILE_NOTI...
9277	1:54:04...	Explorer.EXE	1792	IRP_MJ_QUERY_VOLUME_INFORMATION	C:_Personal\X.doc	SUCCESS	Type: QueryAttributeInformationVolume, FileSyst...
9279	1:54:04...	Explorer.EXE	1792	IRP_MJ_QUERY_VOLUME_INFORMATION	C:_Temp\X.doc	SUCCESS	Type: QueryAttributeInformationVolume, FileSyst...
9280	1:54:04...	Explorer.EXE	1792	IRP_MJ_SET_INFORMATION	C:_Personal\X.doc	SUCCESS	Type: SetEndOfFileInformationFile, EndOfFile: 2...
9281	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal	SUCCESS	Type: NotifyChangeDirectory, Filter: FILE_NOTI...
9282	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal	SUCCESS	Type: NotifyChangeDirectory, Filter: FILE_NOTI...
9283	1:54:04...	Explorer.EXE	1792	IRP_MJ_SET_INFORMATION	C:_Personal\X.doc	SUCCESS	Type: SetBasicInformationFile, CreationTime: 0, ...
9284	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal	SUCCESS	Type: NotifyChangeDirectory, Filter: FILE_NOTI...
9290	1:54:04...	Explorer.EXE	1792	IRP_MJ_WRITE	C:_Personal\X.doc	SUCCESS	Offset: 0, Length: 24,064
9291	1:54:04...	Explorer.EXE	1792	IRP_MJ_SET_INFORMATION	C:_Personal\X.doc	SUCCESS	Type: SetBasicInformationFile, CreationTime: 0, ...
9292	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal		Type: NotifyChangeDirectory, Filter: FILE_NOTI...
9297	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLEANUP	C:_Temp\X.doc	SUCCESS	
9298	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLOSE	C:_Temp\X.doc	SUCCESS	
9299	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLEANUP	C:_Personal\X.doc	SUCCESS	
9338	1:54:04...	Explorer.EXE	1792	IRP_MJ_CREATE	C:_Personal	SUCCESS	De ync...
9339	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal\X.doc	SUCCESS	Typ
9340	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLEANUP	C:_Personal	SUCCESS	
9341	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLOSE	C:_Personal	SUCCESS	
9365	1:54:04...	Explorer.EXE	1792	IRP_MJ_CREATE	C:\	SUCCESS	Desired Access: Read Data/List Directory, Sync...
9366	1:54:04...	Explorer.EXE	1792	IRP_MJ_QUERY_VOLUME_INFORMATION	C:\	SUCCESS	Type: QueryFullSizeInformationVolume, TotalAllo...
9367	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLEANUP	C:\	SUCCESS	
9368	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLOSE	C:\	SUCCESS	
9424	1:54:04...	Explorer.EXE	1792	IRP_MJ_CREATE	C:_Personal	SUCCESS	Desired Access: Read Data/List Directory, Sync...
9425	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal*	SUCCESS	Type: QueryDirectory, Filter: *, 2: .
9426	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal	SUCCESS	Type: QueryDirectory, 1: .., 2: X.doc
9427	1:54:04...	Explorer.EXE	1792	IRP_MJ_DIRECTORY_CONTROL	C:_Personal	NO MORE ...	Type: QueryDirectory
9428	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLEANUP	C:_Personal	SUCCESS	
9429	1:54:04...	Explorer.EXE	1792	IRP_MJ_CLOSE	C:_Personal	SUCCESS	
10009	1:54:05...	System	4	IRP_MJ_WRITE	C:_Personal\X.doc	SUCCESS	Offset: 0, Length: 24,576, I/O Flags: Non-cache...
10010	1:54:05...	System	4	IRP_MJ_SET_INFORMATION	C:_Personal\X.doc	SUCCESS	Type: SetEndOfFileInformationFile, EndOfFile: 2...

Type: SetBasicInformationFile
CreationTime: 0
LastAccessTime: 0
LastWriteTime: 10/3/2007 10:42:53 AM
ChangeTime: 10/4/2007 9:14:30 AM
FileAttributes: n/a

Showing 90 of 10,037 events (0.89%) Backed by C:\Documents and Settings\hye1usr1\Desktop\IST971_Report\ScreenShots\Copy (Different Directory)\Copy

Figure 1: Process Monitor Example.

Explorer.EXE or cmd.exe). By using the results from the process monitor, we can detect not only the insider's unauthorized file-copy action described in Scenario 1 but also the properties of the sensitive file such as file size, file allocation size on disk, creation time, access time, write time, modification time, file attributes, and etc.

4.2 Senenario#2

Copying a Sensitive File with a Different File Name. Scenario #2 is very similar to Scenario #1, but the file name and the location of the copied file are different. Both the original file "X.doc" and the new file "Y.doc" will appear in the registry. In addition, "Y.doc" will appear to have been created after "X.doc." Also, two files will now exist on the system with the same size and same extension. This situation may be common for system files such as .dll files, but is uncommon for two supposedly different documents to be of the same size. This same anomaly appears if a user duplicates a sensitive file under a different name without using the "Save As" function as part of the file handler. In all these instances, the MAC (Modified, Access, Created) times of both the original file and the copy will be changed.

An insider might wish to copy, drag or move a file to a different location or folder. In this case the path of the sensitive file will change, as well as the MAC time. If the file is dragged or moved to an external drive such as a USB thumb drive, then the pointer to the sensitive file will disappear in the MFT (Master File Table) since the file, in this case, will no longer exist on the system.

Our approach detects that the malicious insider makes a copy of the sensitive file labeled as "X.doc" in the "C:_Temp\" directory with the different file name as "Copy of X.doc" into the current directory. In addition to that, it also provides the properties of the sensitive file such as file size, file allocation size on disk, creation time, access time, write time, modification time, file attributes, and etc. Later, the insider may change the name of the copied file, Copy of X.doc, to another name, say, Y.doc.

4.3 Senenario#3

Saving a Sensitive File as a New Name (Rename). Since we assume that the malicious insider already has the privilege to access the sensitive file, X.doc, his opening the file with the associated application, WINWORD.EXE (Microsoft Office Word) is typically not abnormal. However, as we describe in

Scenario#3, if he tries to open X.doc and saves it as a new name, such as Z.doc, we should detect such a suspicious behavior in order to prevent possible compromise of confidentiality on the sensitive file. In particular, for Microsoft Office products including Word, Excel, PowerPoint, the Windows Registry (discussed in Section 2) is another resource of monitoring certain file operating activities such as Open and Save As. In order to detect the threat Scenario#3, we can use the process monitor described in Section 2.

We also use the Windows Registry to detect the threat Scenario#3. The Windows Registry key contains the values of the "Save As" history for the Microsoft Office Word application. It also provides the information about the application version the insider used; Microsoft Office Word 2003 for this case.

4.4 Senenario#4

Contents Copy From a Sensitive File. Since the contents-copy action is not related with the file operating actions and events, a process monitor is not an appropriate tool for monitoring the insider's unauthorized action against the threat Scenario#4. To our best knowledge, unfortunately, there are no monitoring tools available yet for this type of user actions, we develop a command-line style tool named as Clipboard Control (cbcontrol.exe). With this tool we can monitor the contents-transfer from a sensitive file, which was registered in the monitoring list.

For some windows applications, when the copy/paste function is called, the clipboard owner information in the Windows Clipboard system is retrieved from a delegate window handle created from the real owner windows. Unfortunately, the delegate handle does not contain the information where the copied/paste content is come from. Therefore back-tracking is almost impossible due to the lack of the information transfer between the real owner window and the delegate window handle in the Windows Clipboard system. This limitation is from the built-in method to use the delegate handle with the current Windows Clipboard system. If we develop a new clipboard system instead of the currently existing one, which is embedded in most of Windows systems, we can simply overcome this limitation by retrieving the information about the real owner of the contents, the origin of the contents in the current clipboard.

4.5 Senenario#5

Printing Sensitive Contents: A malicious insider might choose to break the confidentiality of information by printing out the contents of a file (X.doc) for the purpose of disseminating the information contained therein or use the information for personal or financial gain, such as industrial espionage. Even though printing a file is one type of file-handling events, Microsoft Process Monitor does not provide enough information for monitoring insider's printing activities. Furthermore, there are no built-in printer-monitoring tools in current Windows operating systems. Therefore, we use a third-party tool, such as SoftPerfect's Print Inspector (http://www.softperfect.com/), to detect the threat Scenario#5. In our implementation we use Print Inspector because it is very light-weighted and easy to use with a simple user interface.

5 SUMMARIES AND FUTURE WORK

In this paper we identified insider-threat scenarios against confidentiality and then described how to detect each threat scenario by analyzing the primitive user activities such as Copy, Rename, Print, Paste, and so on. Finally, we implemented our detection mechanisms by extending the capabilities of existing software packages. Since our approach can proactively detect the insider's malicious behaviors before the malicious action is finished, we can prevent the possible damage proactively. In this particular paper the primary sources for our implementation are from the Windows file system activities, the Windows Registry, the Windows Clipboard system, and printer event logs and reports. However, we believe our approaches for countering insider threats can be also applied to other computing environments.

In our future work we are planning to apply the insider-threat detection mechanisms to other platforms by extending their functionalities of log files and monitoring mechanisms. Furthermore, we will develop new insider-threat detection mechanisms against integrity and availability. Ultimately, we will integrate all the detection mechanisms and apply them to real systems in sensitive organizations.

REFERENCES

Anderson, R. H., 1999. Research and development initiatives focused on preventing, detecting, and responding to insider misuse of critical defense information systems. In *Proceedings of Workshop at RAND*. Santa Monica, CA.

Apap, F., Honig, A., Hershkop, S., Eskin, E., Stolfo, S., 2001. Detecting Malicious Software by Monitoring Anomalous Windows Registry Accesses*. CUCS Technical Report*.

Brackney, R. C., Anderson, R. H., 2004. Understanding the insider threat. In *Proceedings of ARDA (The Advanced Research and Development Activity) Workshop*.

Bishop, M., 2005. The insider problem revisited. In *Proceedings of the 2005 Workshop on New Security Paradigms* (Lake Arrowhead, California, September 20 - 23, 2005). NSPW '05. ACM, New York, NY.

Chinchani, R., Iyer, A., Ngo, H., Upadhyaya, S., 2005. Towards A Theory Of Insider Threat Assessment. In *Proceedings of the International Conference on Dependable Systems and Networks*.

CSI Computer Crime and Security Survey, 2007.

Hayden, M. V., 1999. The insider threat to U.S. government information systems. Tech. rep., *National Security Telecommunications and Information Systems Security Committee (NSTISSAM)*, INFOSEC 1-99.

Honeycutt, J., 2002. *Microsoft Windows XP Registry Guide.* Microsoft Press.

Keeney, M., Kowalski, E., Cappelli, D., Moore, A., Shimeall, T., and Rogers, S., 2005. *Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors.* National Threat Assessment Center, U.S. Secret Service, and CERT® Coordination Center/Software Engineering Institute, Carnegie Mellon.

Moore, A.P., Cappelli, D.M., Trzeciak, R.F., 2008. *The "Big Picture" of Insider IT Sabotage Across U.S. Critical Infrastructure.* CERT Tech. Report, CMU/SEI-2008-TR-009.

Neumann, P. G., 1999. Inside risks: risks of insiders. *Commun. ACM* 42, 12.

Park, J. S., Ho, S. M., 2004. Composite role-based monitoring (CRBM) for countering insider threats. In *Proceedings of Symposium on Intelligence and Security Informatics (ISI)*. Tucson, AZ.

Pramanik, S., Sankaranarayanan, V., Upadhyaya, S., 2004. Security policies to mitigate insider threat in the document control domain. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC),* 2004.

Print Inspector (http://www.softperfect.com/), 2008.

Renesse, R., Birman, K., Vogels, W., 2003. Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining*. ACM Transactions on Computer Systems,* Vol. 21, No. 2, Pages 164–206.

Windows Clipboard (http://msdn2.microsoft.com/en-us/library/ms648709.aspx), 2008.

Windows Sysinternals (www.sysinternals.com), 2008.