1-2013

Insider Threat Detection using Virtual Machine Introspection

M. Crawford

Gilbert L. Peterson
Air Force Institute of Technology

Robert F. Mills
Air Force Institute of Technology

Michael R. Grimaila
Air Force Institute of Technology

Recommended Citation

M. Crawford and G. Peterson, "Insider Threat Detection Using Virtual Machine Introspection," 2013 46th Hawaii International Conference on System Sciences, Wailea, HI, USA, 2013, pp. 1821-1830, doi: 10.1109/HICSS.2013.278.

Insider Threat Detection using Virtual Machine Introspection

Martin Crawford, Gilbert Peterson, Robert F. Mills, and Michael R. Grimaila
Air Force Institute of Technology
Martin.Crawford05@gmail.com, {Gilbert.Peterson, Robert.Mills, Michael.Grimaila}@afit.edu

Abstract

This paper presents a methodology for signaling potentially malicious insider behavior using virtual machine introspection (VMI). VMI provides a novel means to detect potential malicious insiders because the introspection tools remain transparent and inaccessible to the guest and are extremely difficult to subvert. This research develops a four step methodology for development and validation of malicious insider threat alerting using VMI. A malicious attacker taxonomy is used to decompose each scenario to aid identification of observables for monitoring for potentially malicious actions. The effectiveness of the identified observables is validated using two data sets. Results of the research show the developed methodology is effective in detecting the malicious insider scenarios on Windows guests.

1. Introduction

Insiders are frequently [1-7] defined as individuals who are current or former members of an organization, contractor or partner, who are trusted and have or had access or knowledge of the organization's information systems, and objectives. Malicious insiders are a subset of individuals who intentionally misuse their trusted position through a set of actions and against a target or targets that result in a violation of confidentiality, integrity and/or availability (CIA). Malicious insiders may be disgruntled employees, employees who see an opportunity for financial benefit or spies who join an organization in order to commit espionage or financial fraud [6-8].

Successful modification of the CIA affects the organization through data loss, data manipulation, destruction of information, and denial of access to data or a service, all of which negatively affect an organization's efficiency, profit, public image, and overall mission [9].

Unlike external attackers, insiders are already trusted with information on workstations and access portions of it daily. Their trusted position within an organization enables them to cause greater damage. Therefore, developing a monitoring capability to alert for potential insider threats on a workstation can greatly improve defensive potential. Although insider threat monitoring technologies currently exist [1] [11-13], they run at the same privilege level as the insider, allowing the possibility of subversion or determining its capabilities. As such, a monitoring capability that is undetectable by a user would be an improvement for mitigating a malicious insider.

To alert to a potential malicious insider threat, organizations must develop use cases that categorize possible attack techniques, such as data exfiltration via printing. From a generic use case, specific attack scenarios are developed to enumerate steps a malicious insider may perform.

In this paper, the taxonomy developed by Howard and Longstaff [10] for a network attacker is modified to focus on insider threats. Each generated scenario is broken down using this taxonomy to provide a better understanding of the attack. After each action in an attack is identified, corresponding observables are recorded which enable alerting when a specific action is performed.

After identifying observables for each action, they are implemented into the CMAT-V VMI [27] and are tested against malicious insider threat data to confirm the alerting technique for each action is successful. An alert is generated if a potentially malicious action is detected for any observable during a scenario. The alert generation techniques are also compared against two data sets not containing an insider threat. This enables confirmation that the detection techniques only alert for malicious activity and not normal user actions.

The significance of this research is that it presents a novel method for alerting to potential insider threat actions. This is accomplished through VMI allowing monitoring to remain transparent to the individual being monitored. The results of this research reveal a successful detection of all eighteen malicious insider scenarios through VMI on Windows guests.

2. Background

A common method used to mitigate insider threats is monitoring the user's workstation. Compared with network defenses, this is advantageous as monitoring is not limited to only actions that involve network access. Instead, these technologies are capable of monitoring a majority of actions on a workstation.

Current research efforts [11] present methodologies for organizations to employ to obtain better information for existing logging functionality of the system. A solution such as [11] requires minimal additional cost and little overhead to employ. This research sought to maximize the logging capabilities of Linux in order to detect insider threats earlier in their attack, rather than after the insider accomplishes their objective.

Similarly, research by [12] developed a methodology to generate a custom auditing template for the Windows XP OS. Existing Windows logging capabilities are often employed without knowledge of what the organization's actual logging requirements are. Levoy developed a methodology that allows an organization to create an insider threat logging template tailored to their requirements, thus improving the response time to detect insider threat actions.

An overlooked area for insider threat detection is logging on physical hardware. Research by [14] presented a methodology and solution to detect insider threat attacks against Cisco network devices. The solution relies only upon existing functionality of the device, meaning implementation is straightforward by an organization and it does not require any additional firmware to be installed on the device for detection to be successful. Network infrastructure devices are often overlooked by security personnel as they are often thought of as not possessing enough storage capability to hold information, but they process all traffic on a network [14].

Through examination of the Windows registry, a detailed profile of the user's activities on a computer system can be captured. Current research efforts [1] present solutions to insider threats through live monitoring of the Windows registry. This technique allows an organization to build a strong profile of a user's computer usage pattern, enabling rapid insider threat detection and mitigation. However, current detection techniques execute at user or kernel privilege level within the OS and could be disabled or manipulated by an insider.

2.1. Virtual Machine Introspection

Within the Intel x86 architecture, there are four privilege modes or rings, numbered 0 to 3 where 0 is the most privileged. On a host OS, the operating system and kernel execute at ring 0 and applications at ring 3; rings 1 and 2 are not used. The separation of privileges allows the kernel and operating system to remain secure if an application should become compromised [15]. However, it is possible that ring 0 could become compromised and therefore the entire machine would be under an attacker's control. Virtual machines can assist with mitigating this threat.

A virtual machine is an isolated guest operating system instance running on a normal host operating system instance. A host operating system is able to run multiple virtual machine instances; the only limitation is the hardware resources available to the host operating system. Virtual machines have hardware abstraction performed through the hypervisor, or virtual machine manager (VMM). This allows VMs to function the same as if they were a host OS. A VMM is designed as a small software layer to ensure isolation between virtual machines and the host system [16].

With the recent increase in virtualization, organizations have looked for new techniques to monitor the security of their systems. VMI is emerging as a feasible method for securely monitoring a guest OS. Although bridging the semantic gap is challenging, VMI enables more secure guest OS monitoring. Software running within the guest OS is vulnerable to malicious insider modification or disabling while also remaining undetected to administrators. A user with full permissions to an OS instance can easily disable any security software with enough time, an abundant resource for insiders. With VMI, even users with full permission are unaware of the monitoring capabilities of the VMI tool and are unable to compromise them. VMI allows the system administrators to continue to receive information about a VM despite the guest OS being compromised [17]. Additionally, a host-based intrusion detection system (HIDS) typically runs in the OS, meaning it can easily be compromised by malicious insiders or malware [17].

Compiled Memory Analysis Tool - Virtual (CMAT-V) [20] is capable of performing VMI. CMAT-V expands upon the Compiled Memory Analysis Tool (CMAT). CMAT analyzes memory dumps for system information such as network ports, active processes, drivers, registry keys, clipboard information and current users [20-21]. CMAT-V extends CMAT to perform live forensics upon a Windows unprivileged domain (DomU) VM. CMAT-V utilizes Xen's hypervisor management API to interact with the privileged domain (Dom0) and manage DomU virtual machines. The beneficial aspect of CMAT-V, for the purpose of this research, is its live introspection mode. This mode generates the previously mentioned data items from memory introspection of the executing virtual machine. The impact on the guest while running virt-live mode was determined to be approximately 3% to 4.5% decrease in performance [20]. CMAT-V instead of operating in introspection method instead performs full memory captures every 30 minutes. This lowers the computational requirements, and provides a means to expand upon CMAT-V's data capture and use to obtain additional information from the guest's memory. By capturing the memory, the extraction of details about running processes, TCP and UDP network connection information, currently loaded libraries, file handles, registry entries, and loaded drivers from a running virtual machine [20]. The current capabilities are expanded to also obtain all additional sensors identified in Section 3.3.

3. Methodology

The goal of this research is to determine whether insider threat detection can be performed on a Windows guest virtual machine (VM) through virtual machine introspection (VMI)

To accomplish this goal, the research methodology is decomposed into four interrelated steps. The four steps consist of: development of malicious insider taxonomy, VMI observable analysis, malicious insider detection, and data validation.

Prior to performing the four step process, six use cases in which VMI can be used to detect insider threats are identified. From these use cases, eighteen specific attack scenarios are extracted and each is processed and tested following the four step methodology.

The following assumptions are introduced to scope the research. These assumptions are applied to each use case and scenario within this research.

- Users have full access to files on the DomU system, except for those specifically restricted by the Windows OS.
- Users are unaware of the existence of CMAT-V. As a result, malicious insiders will not attempt to obfuscate their activities from CMAT-V specifically, but may attempt to hide from DomU level monitoring.
- The host OS (Dom0), virtual machine manager (VMM), and CMAT-V cannot be circumvented, disabled, infected, or modified by the user.
- Threats modeled are intentionally malicious and their actions are not the result of an accidental breach of confidentiality, integrity, and availability (CIA).
- The malicious insider is acting alone and does not utilize social engineering tactics to aid their attack.
- The insider does not modify the Windows registry to hide their actions.
- Malicious insiders will not use physical attack to access other systems in the network.
- Workarounds for Xen USB passthrough and optical discs produce similar observables as native Windows functionality.
- Clipboard and print operations are performed on pre-determined files for both malicious and non-malicious users.
- All of the actions of a single malicious insider scenario are performed within the time span of one memory capture and each action is performed in the order specified.
- Documents deemed sensitive to the organization have been identified and appropriately flagged.

3.1. Use Case Development

Use cases provide a high-level overview of actions a malicious insider could perform to achieve an unauthorized state of the system. The use cases are selected through examination of malicious insider techniques and security reports. Each use case represents a malicious insider attempting to accomplish a malevolent objective differing from the organization's mission, such as data theft or damage. Specific to each use case are several scenarios which an insider would need to perform. These scenarios provide different techniques a malicious insider may employ and allow the malicious steps to be decomposed into an attack taxonomy and observables.

3.1.1. Use Case 1: Printing Activity. Printer use is a legitimate activity performed by many computer users on a daily basis. However, a malicious insider can employ a printer as a technique to exfiltrate sensitive or classified information. In an environment without strict monitoring of employee's possessions when exiting the premises, a malicious insider could easily walk out with sensitive information. As discussed previously by [21], disgruntled employees may use a printer as their method for stealing corporate data.

The first printing activity scenario examines a malicious insider who connects a new printer to their workstation. The first advantage presented to the insider by this technique is bypassing any network monitoring tools. Network printers are connected to workstations via Ethernet, which allows administrators to easily capture all or specific traffic items, such as print jobs. Another advantage the insider obtains though this method is bypassing monitoring methods on the printer itself. For example, tools such as [22] monitor printer utilization and record job information. By directly connecting the printer to their workstation, the malicious insider is able to bypass both of these security features.

Another scenario created involves a malicious insider who attempts to exfiltrate data not related to their position within an organization by searching for keywords. Motivation for this scenario is to determine if a work scope breach and printing the resulting document can be detected. This scenario also uses a local printer; for organizations that do not allow local printers, this scenario will have a similar taxonomy and observables for network based printers.

The final printing scenario examined involves a malicious insider printing an unusually quantity of documents outside of normal work hours. Insiders may perform malicious actions outside of normal workplace hours to avoid detection by coworkers. Of the presented printing scenarios, this is likely the most damaging to an organization as the user has almost zero risk of being caught by a coworker if they are the only one in the office. Additionally, the insider has ample time to determine which documents are the most valuable and formulate a plan to avoid detection by any physical security at the building's entrance.

3.1.2. Use Case 2: Disable Defense Tools. This use case focuses on a malicious insider who has a technical background. Some malicious insiders are technically proficient and may attempt to subvert known monitoring technologies. The motivation for this use case comes from the potential ability of a malicious insider to disable monitoring that is occurring on their workstation. The use of CMAT-V allows monitoring from a higher privilege level than the user and cannot be directly attacked or disabled unless the malicious insider is able to break out of the virtual machine, an undertaking that is difficult on bare metal virtualization without guest tools [32].

The first scenario created for defensive tools examines a situation where an insider disables current antivirus software. For almost every organization, antivirus is the primary defensive tool employed on workstations against malware that is spread via email, browser exploits, or network exploitation. Newer HBSS may also employ user-level monitoring for insider threat actions.

The second scenario relating to disabling existing defensive technologies involves the malicious insider disabling Windows event logs. The Windows event logs are a valuable tool to administrators and security analysts to monitor activity on a system, such as installing software and account logons/logoffs.

The final defensive tool attempted to be defeated by a malicious insider is post-incident forensics by enabling the InPrivate browsing functionality. Private browsing is a feature in most modern browsers, including Internet Explorer, Google Chrome and Mozilla Firefox. The purpose of this functionality is to prevent history and multimedia items from being stored on the local computer. Although this does not prevent network level traffic inspection, a malicious insider could use this in combination with either HTTP Secure (HTTPS) or Secure Shell (SSH) to bypass network level defenses and possibly hinder post-incident forensics. For the purposes of this scenario, only Internet Explorer will be evaluated as most organizations do not allow users to install additional software on their workstation.

3.1.3. Use Case 3: Removable Media. Removable media is another frequently used method for stealing sensitive data from an organization. The Department of Defense (DoD) currently bans removable flash media, such as USB thumb drives, from all Unclassified but Sensitive Internet Protocol Router Network (NIPRNET) computers [23]. However, as allegedly performed by Pfc. Manning, malicious insiders will find alternate methods to exfiltrate data while still adhering to DoD policy. This use case addresses malicious insiders who use DoD approved removable media to steal sensitive information.

The first scenario involves an insider exploiting existing DoD policy on removable devices and uses a USB hard drive, since flash media is prohibited. Although flash media may be easier to conceal, a USB hard drive could easily be hidden inside a stack of papers, briefcase, or a shoe to bypass physical security inspections. This scenario examines an insider who uses a USB hard drive to steal a document contained within the insider's work scope.

The second scenario is directly motivated by Bradley Manning and replicates a scenario where an insider uses a Compact Disc-Rewritable (CD-RW) to exfiltrate data. Although these devices are not as easy to hide as an external hard drive and do not store as much information, they are still a useful storage medium for a malicious insider to exfiltrate data.

3.1.4. Use Case 4: Employee Behavior. Sudden changes in employee behavior are a precursor to malicious insider attacks against an organization. Coworkers observe visible warning signs from the insider before they perform malicious actions [30]. These scenarios address several suspicious employee behaviors that an organization could monitor to assist in mitigating insider attacks.

The first employee behavior scenario is unauthorized file access by the malicious insider [30]. This scenario is representative of an employee who is able to obtain access to a file that is not within their job description. For the purpose of this scenario, it is ignored how the access was obtained. Possible methods for obtaining access are through privilege escalation or incorrectly configured permissions.

Another malicious employee action modeled is installation of additional software on their computer to assist with data exfiltration [1]. An insider who is able to install software can use the installed to subvert existing defensive technologies employed by the organization on the network and/or workstation. This scenario specifically examines installation of TrueCrypt [28].

The third malicious behavior scenario examines an insider who uses the existing Windows FTP command to transfer files to a remote server [29]. An organization that monitors a workstation for new software installation or running processes outside of those that come with Windows would not detect insiders who use existing Windows software to perform malicious actions.

The fourth scenario modeled is similar to the third in that it relies exclusively on existing Windows commands. Unlike other scenarios, the insider is not attempting to steal property from the organization. Instead, the only goal is to sabotage productivity by destroying data and files within the organization [30]. The malicious insider uses command line commands to connect to another user's computer and delete files.

The last scenario generated examines a situation in which an administrator abuses his or her elevated privilege in an attempt to perform malicious actions under a new user account [31]. The malicious administrator creates a new user in attempt to hinder potential forensic investigation into the actual perpetrator.

3.1.5. Use Case 5: Remote Access. Another technique employed by malicious insiders is remote access. Using remote access allows the insider to perform their attack while not being distracted by coworkers or their currently assigned work task. Additionally, coworkers cannot observe any potentially malicious activity on the insider's screen and report the actions to a security manager within the organization.

The first of two scenarios are representative of a user who uses Microsoft Remote Desktop Protocol (RDP) to access their workstation remotely, such as from their personal computer at home. The malicious insider uses RDP to steal data remotely from their work computer to a personal computer at home. RDP can be configured to use transport layer security (TLS) to prevent an organization from performing a man-in-the-middle (MITM) attack to determine the user's activity, thus defeating any network level defenses.

The second scenario examines a malicious insider who employs RDP to implant malware on his or her own workstation to be leveraged in a future attack by first RDP into a server within the organization and then copying it to their workstation. Although malware was specifically examined in this scenario, this scenario would also be representative of how an organization could monitor file transfer operations between multiple computers by an insider.

3.1.6. Use Case 6: Clipboard Activity. The Windows clipboard is used frequently by users on a system for normal computer tasks. However, it can also contains valuable information regarding an insider attack and therefore examination of the Windows clipboard for post-incident investigation is extremely valuable in determining actions performed by the user [19]. Applying this principle to live introspection can significantly reduce the time between incident and detection and potentially generate real-time detection of malicious activity.

Copying and pasting between two documents is a common use of the Windows clipboard functionality. This scenario is representative of clipboard activity by a malicious insider who accesses an unauthorized document and copies and pastes the contents to a new document.

Similar to the previous scenario, copying and pasting between a document and a web form is another common use of Windows clipboard. The second scenario models an insider who uses an anonymous web form to exfiltrate information from the organization. The insider employs the Windows clipboard and Internet Explorer to perform the attack.

The final malicious clipboard scenario performed is similar to the aforementioned web form clipboard operation, with the key difference being the source application used for the text clipboard operation. Instead, the insider copies sensitive text from Outlook and submits it to a web form.

3.2. Step 1: Malicious Insider Taxonomy

To accurately model and prevent malicious insider behavior, each scenario is decomposed into individual attack actions that can be observed from beginning to end. Decomposition of attacks enables the identification of VMI observables and an effective alerting strategy to be developed. The model described by [10], shown in Figure 1, can be used to describe each malicious insider attack scenario. Each component of the taxonomy and the use of the component for application to the malicious insider threat problem is described below.

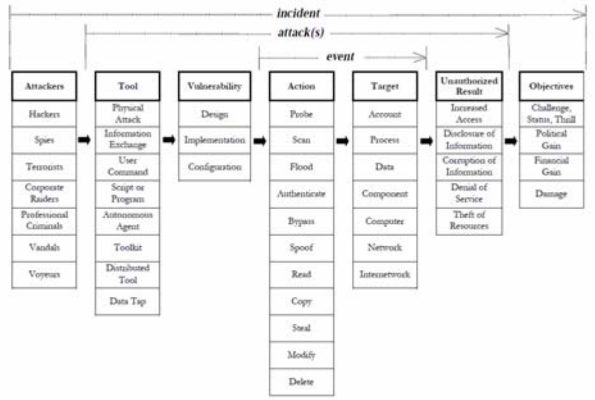

Figure 1. Computer and Network Incident Taxonomy [10].

3.2.1. Attacker. Malicious insiders are the attackers who perform actions against a company using information technology to accomplish an objective. A classification of these individuals appears in [10]. For the purpose of this research, the attacker is always a malicious insider. These individuals are already trusted users of the system and as such, attempting to classify them into the types of attackers identified in [10] only differentiates the title of the attacker and not the tool, action, and target of their action.

3.2.2. Tool. A malicious insider begins their attack by using a tool to exploit a vulnerability within the system. A tool can range from a simple and legitimate command, such as copy and paste, to an automated program or virus. A malicious insider can employ multiple tools during a single scenario.

3.2.3. Vulnerability. A vulnerability is a weakness or deficiency within an information system that can lead to unforeseen and unauthorized access [10]. A vulnerability is typically considered a bug in implementation of a software program that can lead to the development of an exploit. However, it can also be an architectural problem with the design of the system or a misconfiguration of the system. Successful exploitation of the vulnerability results in a breach of the confidentiality, integrity, and/or availability (CIA) of the organization's computer network.

3.2.4. Action. An action is a step taken by the malicious insider in order to obtain a desired effect. Actions incorporate the tool and vulnerability against the target in order to provide the desired result. Actions can include modification, deletion, disabling, moving, copying, pasting, installation, bypassing, or printing. Scenarios may include multiple actions by the insider.

3.2.5. Target. The target is the focus of the malicious insider's tool, vulnerability, and actions. A malicious insider's target is data on the system, or a running program on the system. Several example targets include sensitive corporate documents, other user's account credentials, and running programs on the system.

3.2.6. Unauthorized Result. An unauthorized result is defined as the conclusion of the malicious insider's actions that is not permitted by the organization. These results can include increased system privileges, denial of service, distribution of information, or modification of information.

3.2.7. Objective. The final item in the malicious insider taxonomy is the insider's objective. For the purpose of this research, knowledge of the objective is not relevant to successful detection, but possible objectives a malicious insider may have are enumerated. Objectives can include, but are not limited to, financial gain, damage, or espionage.

3.3. Step 2: VMI Observable Analysis

In the four-step research approach, the second step is identification of possible introspection observables. Each action in the scenario, as identified in the taxonomy, is individually analyzed and an identifier is recorded which facilitates successful observation of the performed action.

These observables consist of registry entries, hexadecimal patterns, and clipboard information. To identify potential observables, each action from the scenarios in Section 3.1 is performed within a Server 2003 and Windows 7 virtual machine running procmon.exe to identify any possible changes in running processes or registry entries. Each individual action is performed and the virtual machine is reset in between each action to ensure a specific action can be correlated with the identified observable. If no observables are identified for an action using this method, a memory capture with an action is examined using a hex editor. Memory captures are examined for any unique hexadecimal patterns that would allow identification of an action.

Table 1 includes a complete listing of all identified observables with a description and the scenario each observable assists with detection. Many observables repeat between scenarios as each scenario may have an action that overlaps with another scenario, such as opening a Word document. It is possible that a scenario may only have few or no observables through VMI and as a result, Windows event logs are employed to assist with identification of observables.

Table 1. VMI Observables.

Scenario	Description	VMI Observable
UC1.S1	Current Printers	HKLM\SYSTEM\ControlSet001\Control\Print\Environments\Windows NT x86\Drivers\Version-3
UC1.S1	Network Printers	HKLM\SYSTEM\ControlSet001\Control\Print\Monitors\Standard TCP/IP Port\Ports
UC1.S1	Current Printers	HKLM\SYSTEM\ControlSet001\Hardware Profiles\0001\System\CurrentControlSet\Control\Print\Printers
UC1.S1 UC6.S1 UC6.S2	Addresses typed in Windows Explorer	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths
UC1.S1 UC1.S3 UC4.S1	Recently mapped network drives	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU
UC1.S1 UC1.S2 UC4.S1 UC6.S1 UC6.S2	Recently accessed Word documents	HKCU\Software\Microsoft\Office\12.0\Word\File MRU
UC1.S2 UC6.S1	Queries sent to Windows search	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery
UC1.S3 UC5.S1	Current user session info (W 7)	HKCU\Volatile Environment\1
UC5.S1	Current user session info (2003/XP)	HKCU\Volatile Environment
UC1.S3	Recent documents and shortcuts	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
UC2.S1	Microsoft Security Essentials Monitoring	HKLM \SOFTWARE\Microsoft\Microsoft Antimalware\Real-Time Protection\DisableRealtimeMonitoring
UC3.S1 UC4.S5	USB Device Information	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
UC3.S1 UC4.S5	USB Device Information	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
UC3.S1	Mounted removable devices	HKLM\SYSTEM\CurrentControlSet\Enum\Storage\Volume\
UC3.S2 UC4.S2	Mounted network shares	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2
UC3.S2	CD Burning Information	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\CD Burning
UC3.S2	CD Burning Information	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\CD Burning
UC4.S2	Typed URLs in Internet Explorer	HKCU\Software\Microsoft\Internet Explorer\TypedURLs
UC4.S2	Mounted devices driver letter	HKLM\SYSTEM\MountedDevices
UC5.S1	RDP Information (Windows 7)	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\##?#Root#RDPBUS#0000# {28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001
UC5.S2	RDP Information (W2003/XP)	HKLM\SYSTEM\ControlSet001\Control\DeviceClasses\{28d78fad-5a12-11d1-ae5b-0000f803a8c2}\##?#Root#RDPDR#0000# {28d78fad-5a12-11d1-ae5b-0000f803a8c2}\#TS001
UC2.S3	InPrivate Browsing	49006E007400650072006E00650074002000450078007006C006F0072006500720020002D0020005B0049006E0050007200690076006100740065005D
UC1.S2	Print Jobs	4E005400200045004D004600200031002E003000300038000000
UC2.S3 UC4.S2	File Downloads	003A005A006F006E0065002E0049006400650065006E007400690066006900650072
UC2.S3 UC6.S2 UC6.S3	Browsing History	68007400740070003A002F002F00
UC2.S3 UC6.S2 UC6.S3	Browsing History	0063006F006D005B0031005D002E00680074006D
UC6.S3	Email Contents	3C68746D6C20786D6C6E733A763D2275726E3A736368656D61732D6D6963726F736F66742D636F6D3A766D6C2220786D6C6E733A6F3D2275726E3A736368656D61732D6D6963726F736F66742D636F6D3A6F66666963653A6F666669636522

3.4. Step 3: Malicious Insider Detection

The third step utilizes information obtained from the previous steps to generate an alerting method for each scenario. Since the research focuses on VMI, the identified observables in combination with Windows event logs are used for alerting. Observables available within the guest that could improve detection accuracy are not used for insider alerting. Several scripts are developed to assist with the extraction of VMI observables from full memory captures. These scripts also compare changes between observables between different memory captures and generate an alert if a change occurs, signaling potentially malicious behavior. Alerts indicate the difference between the previous and current memory capture for a specific observable. After extracting the data from the full memory capture, the output is analyzed for each specific step in the scenario to determine if a single step can be declared malicious. For the purposes of this research, each malicious scenario has the actions performed in the order specified, thereby causing alerts to be generated in a specific order.

3.5. Step 4: Data Validation

After developing detection techniques for each step in a scenario, the detection technique is compared against manually performed malicious and non-malicious scenarios and data collected from the Advanced Cyber Education (ACE) Hackfest [25-26]. This allows for evaluation of the developed alerting mechanisms for accuracy in identifying only insider threats. Analysis of generated alerts is expected to reveal the same observables are present, but the data sets do not contain the same sequence of malicious insider actions.

The first data collection network is created specifically for this experiment to perform malicious insider scenarios.

In addition to malicious insider scenarios, normal user behavior is performed within the malicious insider network. To generate non-malicious data, a script created by [24] is used. Some of the malicious insider scenarios performed by [24] are modified or omitted to maintain the research focus. In addition to this script, non-malicious scenarios are derived from the malicious insider scenarios. The purpose of performing actions similar to malicious insider actions is to ensure only the insider threat actions generate alerts, and not normal user actions. The second network used for data collection is one created during the ACE Hackfest. The experimental network also contained several non-malicious users that allows for collection of non-malicious data. Advanced Cyber Education (ACE) is an eight week course at Wright-Patterson Air Force Base (WPAFB) held during the summer and is open to Air Force, Army, and Navy ROTC cadets. The culmination of the course is a two day exercise focusing on CNO, where two teams attack and defend, while also performing typical user behavior, by completing an internet scavenger hunt that requires workstation interactions such as editing Word documents and sending email [27-28].

For this research, data from the ACE exercise is only used as an additional data set. Unfortunately, many actions simulated in the normal user data set are not present during the ACE Hackfest, such as USB activity, printing, or extensive file access. Additionally, document of the performed actions is not available and assumptions used for this research, such as pre-identification of sensitive files, are not present.

4. Results

Table 2 shoes that all eighteen previously identified malicious scenarios are successfully detected. The scenarios are successfully detected when accounting for the context of a user's actions and correlating additional alerts generated in a similar timeframe. Attempting to determine if a single event could be identified as malicious resulted in a substantial number of false positives, as discussed below.

Applying the contextual approach to the non-insider data sets, all except three scenarios did not appear in the data set. The ACE Hackfest could not have an outcome determined for the three scenarios associated with use case six (UC6) as the source and destination files or programs could not be determined for the clipboard operations. This is due to a limitation with CMAT-V when the data was captured. Without knowing source and destination, an analyst only has a small piece of text they could inspect for blacklisted strings.

Examining each observable individually results in a significant number of false positive alerts. That is to say, examining an alert without considering the context of any other actions the user may be performing. An alert was generated because the value of the identified observable changed significantly from the previous memory capture. In sixty-three percent of alerts, it is the result of logging on or off of a system. Therefore, a substantial number of false positives could be eliminated by enhancing the functionality of CMAT-V to recognize these events. It is important to note that one logoff event would cause all listed observables to generate a false positive as the registry keys no longer exist.

The second set of false positive alerts when examining each observable individually is generated as the result of actual user action on a workstation or server. Examination of these alerts does not indicate malicious insider behavior. These false positives compromise thirty-three percent of the total false positives. This set of false positives indicates it is not feasible to identify a potentially malicious user based on a single event as either many false positives would be reported or monitored events would be to obscure.

Table 2. Malicious Insider Scenario Detection.

Scenario	Insider	Non-Malicious	ACE Hackfest
UC1.S1	Detected	Not Present	Not Present
UC1.S2	Detected	Not Present	Not Present
UC1.S3	Detected	Not Present	Not Present
UC2.S1	Detected	Not Present	Not Present
UC2.S2	Detected	Not Present	Not Present
UC2.S3	Detected	Not Present	Not Present
UC3.S1	Detected	Not Present	Not Present
UC3.S2	Detected	Not Present	Not Present
UC4.S1	Detected	Not Present	Not Present
UC4.S2	Detected	Not Present	Not Present
UC4.S3	Detected	Not Present	Not Present
UC4.S4	Detected	Not Present	Not Present
UC4.S5	Detected	Not Present	Not Present
UC5.S1	Detected	Not Present	Not Present
UC5.S2	Detected	Not Present	Not Present
UC6.S1	Detected	Not Present	Unknown
UC6.S2	Detected	Not Present	Unknown
UC6.S3	Detected	Not Present	Unknown

The final set of false positive alerts indicates potentially malicious insider behavior and should be investigated. For example, one such alert was the result of a user mounting a new volume to their computer, but is not a true positive because no additional potentially malicious actions were performed. These false positives compromise four percent of total false positives.

Reducing false positives of the tool would not affect the success rate for alerting of the malicious insider scenarios. These alerts are the result of registry entries no longer existing after a user logs off of the system, or conversely, the creation of a registry entry when a user logs onto the system. Consequently, eliminating these false positive alerts would not affect the success of alerting for malicious insiders, but would affect the cost of investigation.

Further reduction of false positives would require the alert generation tool to have knowledge of each user's work scope and relevant documents and programs. This would require significant overhead for initial setup, but would enable faster alert generation. Additionally, behavior analysis could be incorporated into the alert generation to enhance existing functionality.

5. Conclusions and Future Work

This paper presented an alerting mechanism for six generated insider threat use cases and their corresponding scenarios. The novel approach to VMI insider threat alerting is presented and functions in a transparent manner to the individual under observation. Transparency to the user provides the insider with a potential false sense of security by not knowing of the existence of the organization's monitoring capabilities.

The paper presented a means to alert based on the presented use cases using hypervisor introspection, which prevents the user from altering the reporting mechanism. However, these are alerts to only the observables and future work needs to be conducted to correlate each alert in a manner that is robust to changes in the order that alerts to malicious insider activities are conducted and accurate.

6. Acknowledgements

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

7. References

[1] J. Hallahan, "Countering Insider Threats - Handling Insider Threats Using Dynamic, Run-Time Forensics," AFRL-RI-RS-TR-2007-213, Rome, NY, 2009.

[2] "Examining Insider Threat Risk at the U.S. Citizenship and Immigration Services (Redacted)," Department of Homeland Security, Office of Inspector General, OIG-11-33, Washington, DC, 2011, pp. 2

[3] W. Baker et al.,"2011 Data Breach Investigations Report," Verizon RISK Team, Available: www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf, pp. 1-72 2011.

[4] S. L. Pfleeger, J. B. Predd, J. Hunker, and C. Bulford, "Insiders Behaving Badly: Addressing Bad Actors and Their Actions," IEEE Transactions on Information Forensics and Security, vol. 5, no. 1, pp. 169-179, Mar. 2010.

[5] V. N. L. Franqueira, A. van Cleeff, P. Van Eck, and R. Wieringa, "External Insider Threat: a Real Security Challenge in Enterprise Value Webs," in Availability, Reliability, and Security, 2010. ARES'10 International Conference on, 2010, pp. 446–453.

[6] D. Cappelli, A. Moore, R. Trzeciak, and T. J. Shimeall, "Common Sense Guide to Prevention and Detection of

Insider Threats (version 3.1)," CERT/Software Engineering Institute, Available: www.cert.org/archive/pdf/CSG-V3.pdf

[7] M. Maybury, "Detecting Malicious Insiders in Military Networks," Military Communications Conference, 2006. MILCOM 2006, Washington, DC, pp. 1-7, 2006.

[8] E. D. Shaw, K. G. Ruby and J. M. Post. "The Insider Threat to Information Systems: The Psychology of the Dangerous Insider," Security Awareness Bulletin, 2-98, Department of Defense Security Institute, Richmond, Virginia. Sept. 1998.

[9] M. Bishop, Computer Security: Art and Science, 2003, Westford, MA: Addison Wesley Professional, 2003, pp. 4-12.

[10] J. D. Howard and T. A. Longstaff, "A Common Language for Computer Security Incidents," Technical Report SAND98-8667, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 1998.

[11] W. Bai, "Development of a Methodology for Customizing Insider Threat Auditing on a Linux Operating System", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2010.

[12] T. Levoy, "Development of a Methodology for Customizing Insider Threat Auditing on a Microsoft Windows XP® Operating System", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2006.

[13] N. Nguyen, P. Reiher, G.H. Kuenning, "Detecting insider threats by monitoring system call activity", IEEE Information Assurance Workshop, pp. 45-52, June 2003.

[14] M. Woolingham, "Detecting Insider Threats on a Cisco Router Using the Native Functionality of the Internetwork Operating System", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2011.

[15] T. Ables, P. Dhawan, and B. Chandrasekaran, "An Overview of Xen Virtualization," Dell Power Solutions, 2005, Available: www.dell.com/downloads/global/power/ps3q05-20050191-Abels.pdf [Nov 23 2011]

[16] B. D. Payne, M. Carbone, and W. Lee. "Secure and Flexible Monitoring of Virtual Machines." Proceedings of the Annual Computer Security Applications Conference, 2007.

[17] T. Garfinkel and M. Rosenblum. "A Virtual Machine Introspection Based Architecture for Intrusion Detection," In Proceedings of the 2003 Network and Distributed System Security Symposium (NDSS), 2003.

[18] J. Okolica and G. Peterson, "Windows Operating Systems Agnostic Memory Analysis," in Proceedings of the Digital Forensic Research Workshop Conference (DFRWS), 2010.

[19] J. Okolica and G. Peterson, "Extracting the Windows Clipboard from Physical Memory," Digital Investigations Journal, 2011 pp. 118-124.

[20] D. Dodge, "Cyber-Situational Awareness Using Live Hypervisor-Based Virtual Machine Introspection", M.S. thesis, ENG, AFIT, Wright-Patterson AFB, OH, 2010.

[21] CERT, (2012, March) Data Exfiltration and Output Devices – An Overlooked Threat. [Online]. https://www.cert.org/blogs/insider_threat/2011/10/data_exfiltration_and_output_
devices_-_an_overlooked_threat.html

[22] Hewlett-Packard. (2012, March). HP Access Control Printing Solutions [Online]. h71028.www7.hp.com/enterprise/us/en/ipg/access-control-printing-solutions.html

[23] United States Air Force. (2012, April). Thumb drives/flash media still prohibited on Air Force Network [Online]. http://www.af.mil/news/story.asp?id=123192400

[24] Swartzmiller, Maj., Simpson, Capt., Sievers, Capt., "Live Network Forensic Response", Final Project, CSCE 527, Air Force Institute of Technology, WPAFB, OH, Sep. 2, 2011, pp. 9-11.

[25] Center for Cyberspace Research, Advanced Cyber Education [Online]. Available: http://www.afit.edu/ccr/ace/docs/Prospective_ACE_Student_Brochure.pdf

[26] Center for Cyberspace Research, General Course Information [Online]. Available: http://www.afit.edu/ccr/ace/docs/general_course_information.pdf

[27] D. A. Dodge et al., "Simulating windows-based cyber attacks using live virtual machine introspection", in *2010 Summer Simulation Multiconference*, 2010, pp. 550–555.

[28] "TrueCrypt", Available: www.truecrypt.org, [15 Aug 2012].

[29] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, 'SIDD: A framework for detecting sensitive data exfiltration by an insider attack', in System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on, 2009, pp. 1–10.

[30] M.R. Randazzo et al., "Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector", Carnegie Mellon Univ., Software Eng. Inst., 2004; Available: www.cert.org/archive/pdf/bankfin040820.pdf

[31] Rayethon, "Best Practices for Mitigating and Investigating Insider Threats", Available: www.raytheon.com/capabilities/rtnwcm/groups/iis/documents/content/rtn_iis_whitepaper-investigati.pdf, 2009 [7 Nov 2011].

[32] P. Hoffman, K. Scarfone, and M. Souppaya. "Guide to Security for Full Virtualization Technologies," National Institute of Standards and Technology (NIST), 2011, Available: csrc.nist.gov/publications/nistpubs/800-125/SP800-125-final.pdf [Nov 23 2011]