Carnegie Mellon University
Software Engineering Institute

DEVELOPING INSIDER RISK METRICS FROM HOST-BASED MONITORING

Carrie Gardner
Daniel Costa
Michael Albrethsen

February 2021

DOI: 10.1184/R1/13424687

1 Introduction

Host-based monitoring serves as a key capability for the security strategy of insider threat programs. Host-based monitoring tools collect observable events on information system usage and end-user behavior for risk measurement to facilitate data-driven risk mitigation strategies. Aggregated and analyzed, recorded events provide intelligence on the activity taking place on organizations' assets.

To gather and record applicable, high-fidelity data points, many organizations turn to some form of a host-based monitoring tool for their organization's devices, such as servers, user workstations, and mobile devices. There are many associated tools, some with overlapping collection coverage, that can be used for host-based monitoring. We list examples of these tools in Table 1.

Table 1: Taxonomy of Host-Based Monitoring Tools

Tool Class	Brief Description	Example Solutions
User Activity Monitoring (UAM)	UAM tools must be able to monitor any user activities, at any time, on any organization-owned asset, and on any network (National Insider Threat Task Force, 2014). In general, UAM tools must be able to monitor keystrokes; record screen capture; record content (e.g., communications, web browsing); and monitor clipboard data, USB port activity, kernel processes, and application activity.	ForcePoint Insider Threat, Digital Guardian, Veriato
Data Loss Prevention (DLP)	DLP tools must be able to identify, monitor, and safeguard data at rest, data in motion, and data in use. DLP tools must employ deep-content analysis to identify and monitor changes to data (Brenner, 2009).	Symantec DLP, Trustwave DLP, McAfee Total Protection for DLP, Check Point DLP, Digital Guardian Endpoint DLP
Operating System Logs	This tool class logs events such as machine state changes or security events.	Windows event logs, Ubuntu system logs, macOS console logs

Tool Class	Brief Description	Example Solutions
Application Logs	Application logs are created by software applications that log events such as errors, warnings, or information events.	MySQL logs, Microsoft Office
Host-Based Intrusion Detection System (HIDS)	A HIDS is a tool that mitigates threats by monitoring system files and network interfaces.	OSSEC, Verisys, Lacework, Nessus
Antivirus	An antivirus tool detects and mitigates malicious software.	Norton Security Antivirus, McAfee AntiVirus, Kaspersky Anti-Virus
Enterprise Audit Management (EAM)	EAM tools monitor authentication events, file and object events, removable media events, privileged activity events, system state changes, and print events (Committee on National Security Systems Directive, 2013).	McAfee HBSS, ForcePoint Insider Threat, Digital Guardian, Veriato, Splunk, Arcsight

In this report, we focus on the **generic capabilities** of host-based monitoring across the span of tools listed in Table 1 using the competencies we identify in Table 2.

Table 2: Taxonomy of Host-Based Monitoring Capabilities

Capability Class	Example Observable Events
System Events	The system is powered on; the system is restarted; the system registry key is changed.
Logon Events	A user logs on; a user incorrectly enters a password.
File Events	A file is created; a file is modified; a file is deleted.
Print Events	A sensitive file is printed.
Process Activity Events	An application is started; an application is stopped.
Web Activity Events	A browser is directed to cnn.com; a file is downloaded from a website.
Electronic Communication Events	A user instant messages a co-worker; a user receives an email from an external address.
Removable Media Events	An unknown removable media device is attached.
Audit Policy Events	A user violates a security policy.

Given the extensive scope of data that can be collected, host-based data collection provides holistic insight into the operating state of the machine as well as critical insight into the activities and mind of the user. With such granularity of insight, host-based monitoring is one of the prime collection capabilities for measuring insider threat indicators, or technical or behavioral observables that are associated with a potential increased risk to an organization.

We refer to this measurement process as *building insider threat risk metrics from host-based monitoring*. Building these metrics helps identify observable precursors to undesirable events, such as information technology sabotage or a less-severe consequence of employee disgruntlement.

To comply with the White House Memorandum on *National Insider Threat Policy & Minimum Standards* and Committee on National Security Systems Directive (CNSSD) 504, insider threat programs can adopt an approved Commercial-Off-the-Shelf (COTS) solution to meet the comprehensive list of requirements (National Insider Threat Task Force, 2017). However, insider threat programs can also adopt a multi-tool approach by leveraging a set of tools that satisfy those capability standards.

In this report, we document the breadth and depth of standard host-based monitoring capabilities to illuminate the rich data-collection potential these tools offer in the aggregate. Insider threat programs can use this report to build their multi-tool approach and be in compliance with CNSSD 504.

1.1 Structure of this Report

This report is organized as follows:

Section 2 describes, in detail, the generic host-based monitoring capabilities and discusses generic attributes that should be recorded for data collection.

Section 3 enumerates various insider threat indicators that can be monitored or measured via host-based collection capabilities, and documents vendor-neutral approaches for monitoring.

Section 4 discusses the general state of the art of host-based monitoring and its challenges, as we look to the future of activity monitoring.

The **appendix** identifies Windows 10 log events that relate to each of the host-monitoring functions discussed in the report.

2 Observables from Host-Based Monitoring

In this section, we present several capability tables that document functional data-collection areas for host-based monitoring. These nine capabilities are

- system events
- logon events
- file events
- print events
- process activity events
- web activity events
- electronic communication events
- removable media events
- audit policy events

Aggregated together, these data sources can construct a rich perspective of user and machine activities. Such a perspective offers insider threat analysts context into a potential event, revealing nuance in specific events while preserving the overarching trends or patterns of behavior.

In each table, we identify associated fields for each collection record. This is a non-exhaustive list of fields; the tables distinguish only necessary attributes of the records. Some of these records might not be logged by default on all Operating System (OS) versions. Refer to the documentation for each OS to determine where each field is found. The appendix lists relevant event types for Windows 10.

2.1 System Events

System events describe events when the system state changes, such as when a system is starting up, shutting down, restarting, or sleeping.

Table 3: System Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
Operation	Starting up, shutting down, restarting, or sleeping

2.2 Logon Events

Logon events describe events when a user is logging onto or off of a system.

Table 4: Logon Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
System Identifier	Hostname or IP address of the computer accessed
Status	Logon, Logoff, or Failed logon

2.3 File Events

File events are activities associated with CRUD (Create, Read, Update, or Destroy) operations on a file. File event logs contain information about how users interact with file objects. In Table 5, we provide the associated fields for file events.

Table 5: File Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
Object Identifier	Unique identifier of the accessed file object
CRUD Operation	Create, Read, Update, or Destroy

2.4 Print Events

Print events are the observables associated with printing, scanning, or faxing documents either electronically or with a physical multi-function printer device. In Table 6, we identify standard fields associated with printing activity. These fields capture the context associated with printer activities.

Table 6: Print Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
Filename	Name of the file being printed
Printer Identifier	The printer or print server being used

2.5 Process Activity Events

Process activity events are the observables associated with creating, changing, allocating resources, or terminating a process or application. In Table 7, we identify standard fields associated with collecting process activity event data. These fields capture process activity events, such as the termination of a shell program or the start of a batch script.

Table 7: Process Activity Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
Process Name	Name of the process or application
Window Title	Title of the application window
Window Visibility	Whether the window is visible or not
Activity Event	The process activity event (e.g., start, update, stop, scheduled)

2.6 Web Activity Events

Web activity events record Internet browsing and http requests. Examples of web activity events include browsing an intranet page or connecting to an external FTP server. In Table 8, we identify standard fields associated with logging these types of events.

Table 8: Web Activity Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
URL	Name of the process or application
Title	Title of the web page window
Website Domain	Domain of website
Category	Categorical type of the website visited

2.7 Electronic Communication Events

Electronic communication events are communication records that travel to or from a user in any channel, such as instant messaging or email. Electronic communication channels are rich, unstructured data sources that can yield unique insight into the mind of the sender through affect analysis. Affect analysis broadly relates to the task of identifying and measuring emotion, sentiment, personality, mood, and interpersonal stances (Gardner, 2018). In Table 9, we present basic relevant fields from electronic communication events.

Table 9: Electronic Communication Event Fields

Name	Description
Timestamp	Time of the event
Sender	Unique identifier of the user who sent the message
Recipients	Unique identifiers of the users who the message was sent to
Subject	Subject of the message
Content	Content of the message sent
Attachments	Attachments, if any, that were included

2.8 Removable Media Events

Removable media events are the observables associated with the use of a removable media device, such as an external hard drive or thumb drive. In Table 10, we present relevant fields for removable media events.

Table 10: Removable Media Event Fields

Name	Description
Timestamp	Time of the event
Device Identifier	Unique identifier of the removable device
Interface	What interface the device is connected to
Activity	What operation was done on the device (i.e., file CRUD operations)
File Name	Name of the file
File Classification	Sensitivity classification of the file
Device Status	Whether the device was connected or disconnected
Device Identifier	Unique ID that shows the manufacturer, product ID, and version

2.9 Audit Policy Events

Audit policy events record policy violations or rule triggers. Audit policies can be locally managed on the host through specific security settings or centrally managed through a policy orchestration service. In Table 11, we present the fields associated with collecting audit policy event data.

Table 11: Audit Policy Event Fields

Name	Description
Timestamp	Time of the event
User Identifier	Unique identifier of the user
Policy	Name of the violated or triggered policy rule
Description of Activity	Description of the activity that violated the rule

3 Deriving Insider Risk Indicators from Observables

CERT researchers define an *insider threat indicator* as, “A technical or behavioral observable that is associated with a potential increased risk to an organization’s critical assets posed by insiders.*”* To develop an indicator, we first must identify the types of threats the organization is concerned about. Then, observables associated with those threats being realized are identified and implemented as an indicator.

We list indicators that can be measured with host-based monitoring in Table 12. We identified indicators that have easy-to-medium-implementation complexity. An insider threat indicator is a technical or behavioral observable that is associated with a potential increased risk to an organization’s critical assets that is caused by insiders.

In Table 12, we identify each indicator, provide a short description, list the associated observables, and note the number of actual incidents we have documented that include the indicator. We calculate the incident counts from the cases coded in the CERT Insider Threat Incident Corpus.[1]

Table 12: Example Host-Based Indicators

Short Name	Description	Associated Observables
Data Exfiltration – Email	The insider removes an organization's intellectual property via email without authorization.	A user transmitting a sensitive file to an external email address
Data Exfiltration – Cloud Storage	The insider removes an organization's intellectual property via cloud storage device(s) without authorization.	A user uploading sensitive files to an online platform
Data Exfiltration – Paper	The insider removes an organization's intellectual property via paper (i.e., printing) without authorization.	A user printing sensitive documents
Data Exfiltration – Removable Media	The insider removes an organization's intellectual property via removable media device(s) without authorization.	A user moving sensitive files to a non-company removable media device
Deleted/Modified Logs	The insider modifies or deletes software or system logs without authorization.	A user deleting Windows Security logs
Disgruntled Employee	The insider is disgruntled with the organization and desires to “get back” at it.	A user including a higher than normal number of words related to anger, disgust, or sadness in email communications
Hidden/Renamed Files or Registry	The insider has hidden or renamed files or registry entries.	A user changing a registry key value

[1] Read more about the CERT Insider Threat Incident Corpus in a September 2020 SEI blog post: https://insights.sei.cmu.edu/insider-threat/2020/09/insider-threat-incidents-assets-targeted-by-malicious-insiders.html (Miller 2020).

Short Name	Description	Associated Observables
Inserted Malicious Code into the Operational System	The insider inserts malicious code (e.g., viruses or worms) into the organization's operational system (e.g., The insider planted a logic bomb that was set to delete information).	A user installing malware
Inserted Malicious Code into the Source Code	The insider inserts malicious code into the organization's IT system's source code (e.g., The insider inserted malicious code into the organization's valuable search algorithm to direct traffic to their site).	A user programming a backdoor into a program
Attached External Hardware to the Organization Desktop	The insider attaches a removable device to an organizational desktop (e.g., The insider attached a USB drive to their computer).	A user attaching a non-company removable media device to a company machine
Use of a Compromised Account	The insider can access the organizational system using an account of another employee that had been compromised (e.g., The insider copied another employee's account and password prior to being terminated).	A user using an anomalous account

4 Implementing Analytics for Host-Based Indicators

A number of basic analytic techniques can be used to process metrics and develop indicators. These techniques range in complexity from matching a simple value in a field (e.g., looking for an important keyword in a filename) to detecting a complex anomaly. In Table 12, we summarize the types of analytics commonly implemented in insider threat monitoring systems.

Table 13: Analytics for Host-Based Indicators

Name	Short Description	Implementation Considerations
Value Matching	The ability to identify if a field value matches a known string	This analytic is good for identifying explicit rule violations, but it can lead to false positive when applied more generically.
Pattern Matching	The ability to identify if a field value matches a pattern	This analytic is good for identifying explicit rule violations, but it can lead to high false positive when applied more generically. It has additional computation power when compared to value matching.
Individual Volumetric Anomaly Detection	The ability to identify statistical anomalies relative to how often or how much a particular event occurs relative to a particular individual	How do individuals' job responsibilities change over time? Will this analytic catch long-term malicious activity?
Peer-Group Volumetric Anomaly Detection	The ability to identify statistical anomalies relative to how often or how much a particular event occurs relative to a particular set of individuals	How are peer groups defined? How do job responsibilities change over time?
Individual Temporal Anomaly Detection	The ability to identify statistical outliers relative to when a particular event occurs relative to a particular individual	How do individuals' job responsibilities change over time?
Peer Group Temporal Anomaly Detection	The ability to identify statistical anomalies relative to when a particular event or events occurs relative to a particular set of individuals	How are peer groups defined? How do job responsibilities change over time?

Anomaly detection is a useful technique for determining when something falls outside of the norm, and does not fit into a specific pattern or signature that we are explicitly looking for. However, many anomalous events are not indicative of anything bad happening. This is an important distinction that must be accounted for when integrating anomaly detection techniques into an insider threat monitoring program.

5 Conclusion

There are many challenges to overcome when implementing a host-based monitoring system. Available resources must be considered when determining data sources, analytic methods, and procedures used. This consideration includes the available network bandwidth within and between networks, storage capabilities, and processing capabilities.

As analytics become more complex and more data sources are brought into the system, the processing required increases significantly. Prioritizing assets and their threats is important in helping to determine what data sources should be collected and what analytics and indicators should be implemented. Indicators should be constantly reviewed, based on feedback from the investigations team, to determine which ones provide the most value in detecting actual negative insider activity.

We defined standard capabilities of host-based monitoring to elucidate the often-ambiguous field of host-based or user-activity-based monitoring. We described specific capability definitions, depicted generalized audit rules in a product-neutral approach, and characterized analytical techniques for interpreting data for risk measurement and analysis. This explicit explanation should enable researchers, developers, and general users to have more constructive conversations about host-based monitoring capabilities.

Mobile devices are a common blind spot in network monitoring, as they typically do not provide the same level of logging produced by servers and workstations. Many organizations allow their employees to have email access at a minimum, and many provide access to other internal resources such as file shares. This user access can provide mobile devices with a substantial amount of sensitive organizational resources. Limiting mobile device access to critical resources or increasing visibility into user actions on their mobile devices via mobile device management (MDM) software is an important consideration.

In future work, we plan to introduce a standard for describing implementation configurations for indicator monitoring. This new standard is planned to be a practical application of this work that is intended to refine the process of indicator sharing and deployment.

Appendix Windows Event Logging

In this appendix, we identify Windows 10 log events that relate to each of the host-monitoring functions we previously discussed. These events are just a sample of the related Windows Events. Please see the Ultimate IT Security website (https://www.ultimatewindowssecurity.com) for more information about Windows event logging (Ultimate IT Security, 2020).

System Events

Window Event ID	Name	Description
4608	Windows is starting up.	Identifies when Windows boots
4625	Windows is shutting down.	Identifies when Windows shuts down
4697	A service was installed in the system.	Identifies when a service extension of the Windows OS has been installed
5025	The Windows Firewall Service has been stopped.	Identifies when the firewall has been disabled
5030	The Windows Firewall Service failed to start.	Identifies when the firewall service was unable to start

Logon Events

Window Event ID	Name	Description
4624	An account was successfully logged on.	Identifies when a user logs on to a system
4625	An account failed to log on.[2]	Identifies an unsuccessful logon event and why the attempt was unsuccessful
4634	An account was logged off.	Identifies when a user logs off an account
4647	User initiated logoff.	Identifies when a user intentionally initiated account logoff (This is typically found with event 4643; however, when event 4647 is not present, then the logoff may be due to an idle network session.)
4648	A logon was attempted using explicit credentials.	Identifies when a user attempts to log on or connect to a system using alternative credentials than the ones for the current session (e.g., an administrator is logged into a system and then attempts to mount a file server using the credentials of another user)
4649	A replay attack was detected.	Identifies when a login appears to be a replay attack

[2] Failed logon event logging is not enabled by default.

Window Event ID	Name	Description
4672	Special privileges assigned to new logon.	Identifies when an account logs in with administrator or privileged access
4800	The workstation was locked.	Identifies when the workstation is locked, either due to user initiation or inactivity
4801	The workstation was unlocked.	Identifies when the user logs back into a workstation after the workstation was locked out (event 4800)

File Events

Window Event ID	Name	Description
4656	A handle to an object was requested.	Identifies when a user attempts to access (open) an object (e.g., a file or folder) that is configured to log all access attempts
4658	The handle to an object was closed.	Identifies when a user closes an object (e.g., a file or folder) that is configured to log all access activity
4663	An attempt was made to access an object.	Identifies when an activity is performed on the object that has been accessed
4664	An attempt was made to create a hard link.	Identifies when a user creates a hard link (i.e., symbolic pointer) to an object

Print Events

Window Event ID	Name	Description
307	Printing a document.[3]	Identifies the time, computer, user, and printer associated with a print job

Process Activity Events[4]

Window Event ID	Name	Description
4688	A new process has been created.	Identifies when a new process is initiated
4689	A process has been exited.	Identifies when a process is terminated
4698	A scheduled task was created.	Identifies when a user creates a new scheduled task
4699	A scheduled task was deleted.	Identifies when a user deletes a scheduled task

[3] The Windows print service operational log must be enabled to log these events.

[4] On Windows systems, process activity events are not automatically logged. However, systems process logging can be enabled by enabling ‘audit process tracking’ through group policy. When this functionality is enabled, the logs are viewable in the Windows Security log.

Window Event ID	Name	Description
4700	A scheduled task was enabled.	Identifies when a scheduled task is initiated
4701	A scheduled task was disabled.	Identifies when a scheduled task is terminated
4702	A scheduled task was updated.	Identifies when a user updates a scheduled task

Web Activity Events

See the documentation for each web browser in use to learn how to enable web activity logging.

Electronic Communication Events

See the documentation for each electronic communication application in use to learn how to enable logging.

Removable Media Events

Window Event ID	Name	Description
6416	A new external device was recognized by the system.	Identifies when Windows recognizes that a new device was attached
6423	The installation of this device is forbidden by system policy.	Identifies when a device is installed on a machine that has a policy explicitly prohibiting it
6424	The installation of this device was allowed, after having previously been forbidden by policy.	Identifies when a device is installed on a machine after it has been recently forbidden by policy

Bibliography

URLs are valid as of the publication date of this document.

Brenner, B. (2009, June 02). *Security Analyst to DLP Vendors: Watch Your Language*. Retrieved December 15, 2020, from https://www.csoonline.com/article/2124056/security-industry/security-analyst-to-dlp-vendors--watch-your-language.html

Committee on National Security Systems Directive. (2013). *Enterprise Audit Management Instruction for National Security Systems (NSS).* Retrieved December 15, 2020, from https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-1015_Enterprise_Audit_Management_Instruction_for_NSS.pdf

Gardner, C. (2018, August 10). *Foundational Research Behind Text Analytics for Insider Threat: Part 2 of 3.* Retrieved December 15, 2020, from Software Engineering Institute (SEI) Blogs: https://insights.sei.cmu.edu/insider-threat/2018/08/foundational-research-behind-text-analytics-for-insider-threat-part-2-of-3.html

Microsoft. (2017, April 19). *Basic Security Audit Policy Settings*. Retrieved May 2, 2018, from https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/basic-security-audit-policy-settings

Miller, S. (2020, September 29). *Insider Threat Incidents: Assets Targeted by Malicious Insiders.* Retrieved December 15, 2020, from Software Engineering Institute (SEI) Blogs: https://insights.sei.cmu.edu/insider-threat/2020/09/insider-threat-incidents-assets-targeted-by-malicious-insiders.html

National Insider Threat Task Force. (2014, March 14). *Clarification of Enterprise Audit Management (EAM), User Activity Monitoring (UAM), Continuous Monitoring, and Continuous Evaluation.* Retrieved December 20, 2018, from https://security.pae.com/Documents/Regulations/EAM_UAM_and_Continuous_Monitoring_Definitions-Signed.pdf

National Insider Threat Task Force. (2017). *National Insider Threat Guide: A Compendium of Best Practices to Accompany the National Insider Threat Minimum Standards.* Retrieved December 15, 2020, from https://www.dni.gov/files/NCSC/documents/nittf/NITTF-Insider-Threat-Guide-2017.pdf

Ubuntu. (2015, January 23). *LinuxLogFiles*. Retrieved May 2, 2018, from help.ubuntu.com/communityLinuxLogFiles

Ultimate IT Security. (2020, December 15). *Ultimate IT Security Website*. Retrieved December 15, 2020, from https://www.ultimatewindowssecurity.com/

Legal Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other docu-mentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

DM20-1172.

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu