Air Force Institute of Technology

Integrity - Service - Excellence

Log Analysis for Insider Threat Detection

Michael R. Grimaila, PhD, CISM, CISSP, IAM/IEM
Department of Systems and Engineering Management
Center for Cyberspace Research
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433-7765

Cyber Security Expo
University of Memphis
Memphis, TN
15 October 2010

Disclaimer

The views expressed in this presentation are my own and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government

Overview

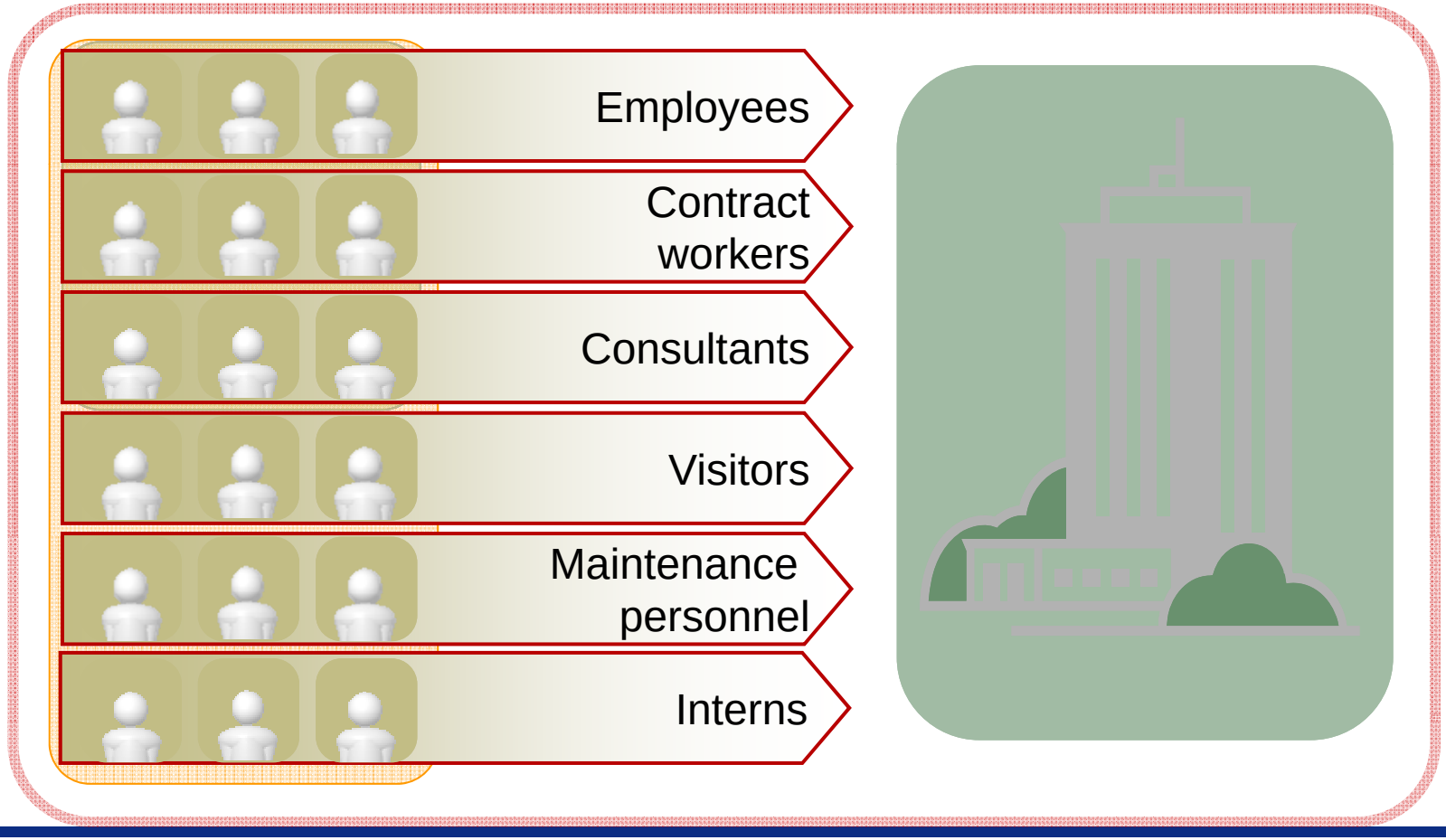
- What is an Insider?
- Observables
- How to Detect Insiders?
- Problem Statement
- Scenario-Based Detection
- Detection Criteria
- Total Cost of Ownership
- Conclusions

What is an Insider?

- An insider is a person in possession of information and/or access not generally available to the public
- An insider has: *Access*, *Authority*, *Intent*, *Trust*
- Trust concepts include *Benevolence*, *Competence*, and *Integrity*
- Insiders can easily exfiltrate information without being detected by employing stealth, deception, or other clandestine means
- DoD considers malicious code as an insider

Who Are The Insiders?

Who has (or in the past has had) physical or logical access to network resources?



Why Insider Threats?

Ignorance	**Insiders do not understand or are not familiar with the existing security policies**
Unintentional	**Using a system outside of its normal modes of operation**
Carelessness	**Insiders do not think about how their actions will break the rules, resulting in a breach of security**
Disregard of security policies	**Insiders will act in ways that make their lives easier, even if it involves going against security policies**
Maliciousness	**Insiders will purposely damage, destroy or compromise information – for financial gain, personal satisfaction, or adversarial advantage**

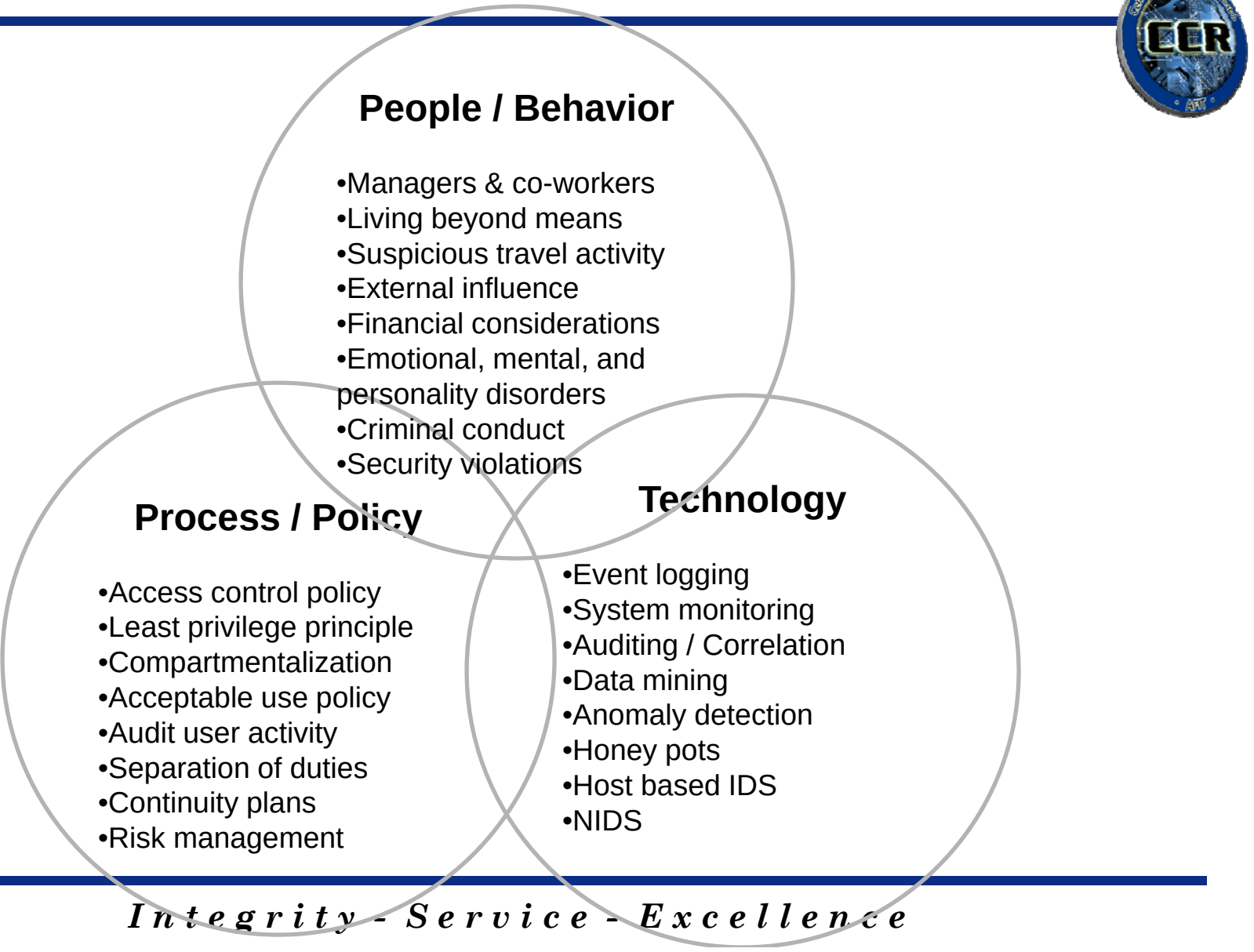
A Significant Problem

- Marie Cooley deleted $2.5 Million worth of architectural files
- Jameel Ahed (RobotFX) $280 Million government contract accused of stealing trade secrets from former employer iRobot
- Hanjuan Jin: Random search stops $600 Million in trade secrets bound for China
- Robert Hanssen: 27 year FBI veteran, spied for USSR/Russia for 15 years – Held TS/SCI clearance

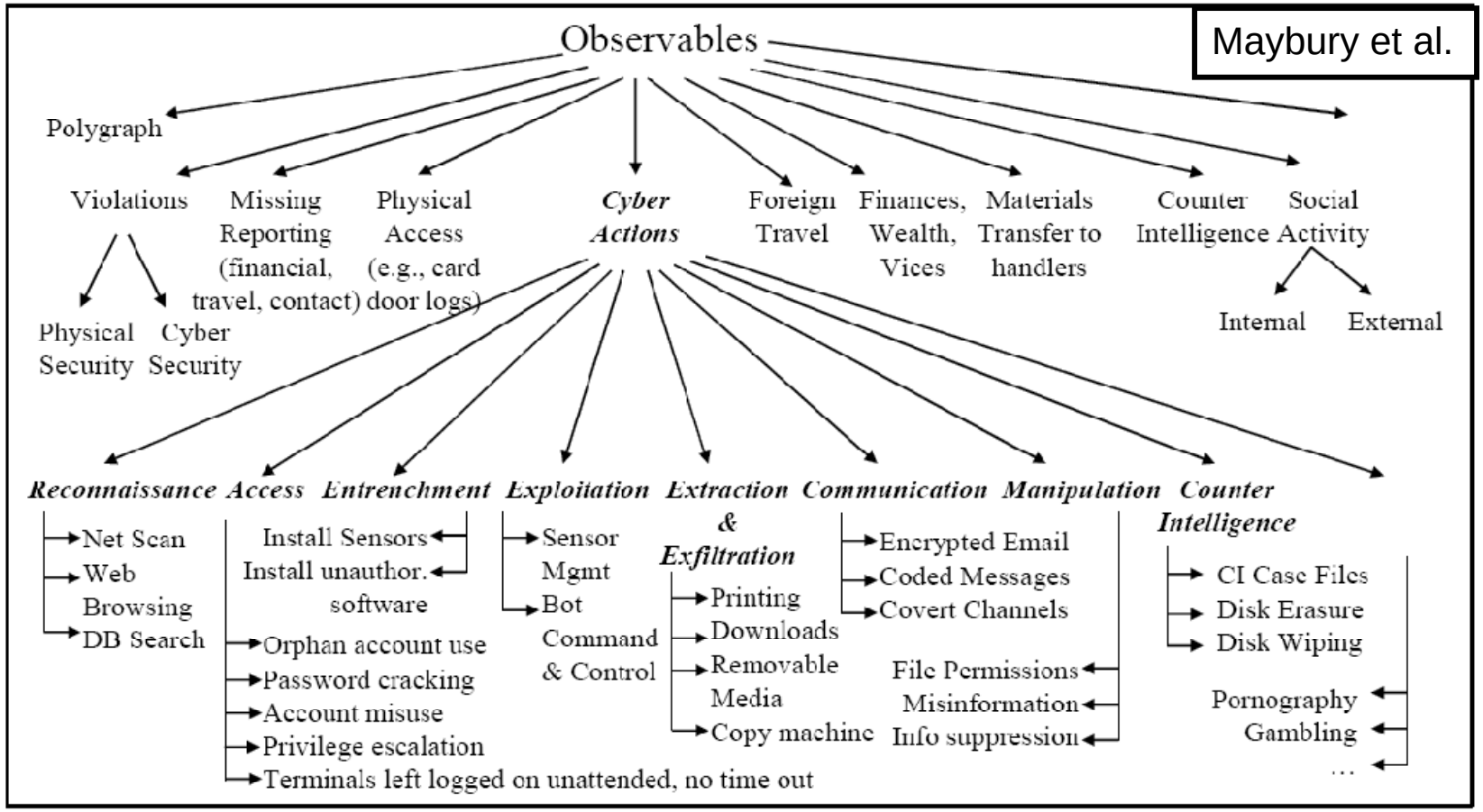
Insider Characteristics

- Technically competent to highly-skilled
- Attempts to cover up and/or destroy evidence
- Employs sophisticated search / query techniques
- Repeated violations of “need to know” access
- Downloads large amounts of data
- Copying data to portable media
- High volume printing
- Encrypted communications
- Changes system logs to hide activity
- Looks for signs that they are under suspicion
- Unusual changes in behavioral patterns

A Multidisciplinary Problem



Insider Threat Observables

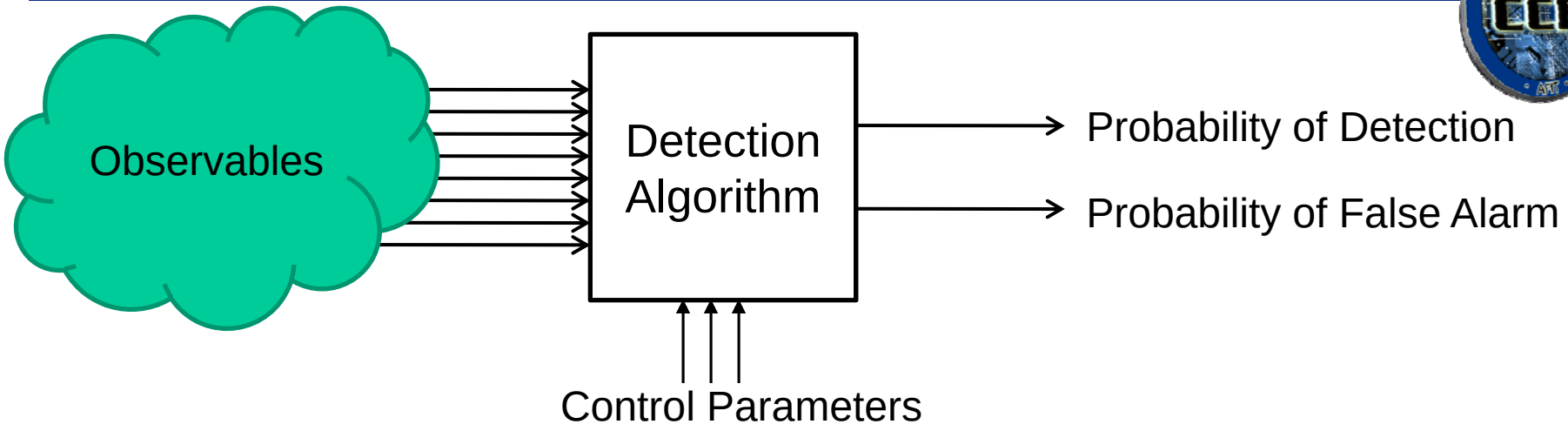


Not everything that counts can be counted, and not everything that can be counted counts – *Albert Einstein*

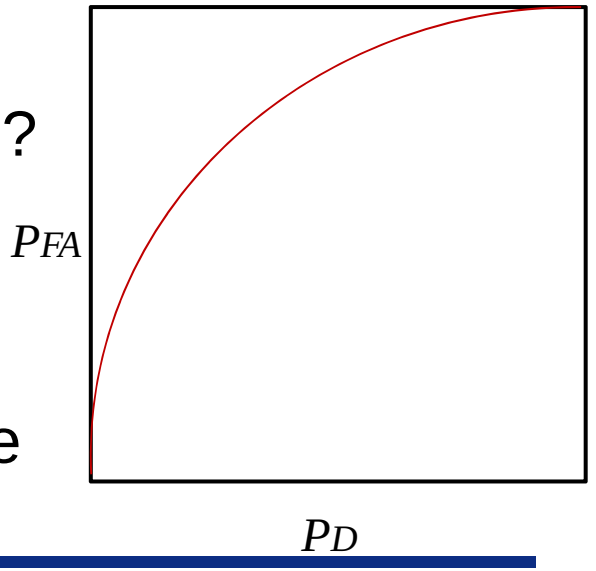
Insider Detection Approaches

- Staged:
  - Detect anomalies in user behavior to assess the risk of malicious insider behavior
- Multi-Perspective:
  - Detect anomalies in user behavior with respect to *user-to-user, user-to-content, user-to-resource* relationships
- Multi-Disciplinary:
  - Social Network Analysis (SNA) - Apply concepts from SNA to detect anomalies in social behavior [user-to-user]
  - Semantic Analysis (SA)- Leverage Natural Language Processing (NLP) and machine learning techniques to analyze the textual data associated with insiders at a semantic (conceptual) level [user-to-content]
  - Composite, Role-based Monitoring (CRBM) – Analyze insider activity based on the organizational, application and operating system roles [user-to-resource]

How to Detect Insiders?



- Which observables should I use?
- Which detection algorithm(s) should I use?
- How should I adjust control parameters?
- How can I evaluate the effectiveness of different detection algorithms?
- What is the Total Cost of Ownership of the insider detection solution?



A Challenging Problem

- Access: Increasing requirements to share information
- Interdisciplinary Nature: Spans across the behavioral, cognitive, social, and technical domains
- Malicious Behavior: “Low and Slow” nature
- Risk Analysis: Low-probability, high-impact threat
- Internal System / Process Design: Assumes benevolence of “trusted” entities
- Immature Tools: Require significant learning curve to become efficient at “tuning”, operating, and maintaining
- Cost: Resources (e.g., time, effort, personnel) to conduct screening, analysis, auditing, and investigations
- Volume: Huge amounts of information accesses

Practical Solutions

- Ideal solutions are often impractical
- Observable collection across all domains is EXPENSIVE!
- Creation and evaluation of detection algorithms across multiple domains is difficult
- How good can we do if we focus only upon information technology related events?
- Within a Single System:
  - Operating System Logs
  - Web Server Logs
  - Application Logs
- Across Multiple Systems and Infrastructure Elements

Operating System Logs

- Operating systems have configurable system and security logging capability
- Logging provides information about:
  - Application: Records events logged by applications and user programs
  - System: Records events logged by system components such as system processes and device drivers
  - Security: Records security events such as changes in user's privileges, changes in the audit policy, file and directory access, and system logons and logoffs

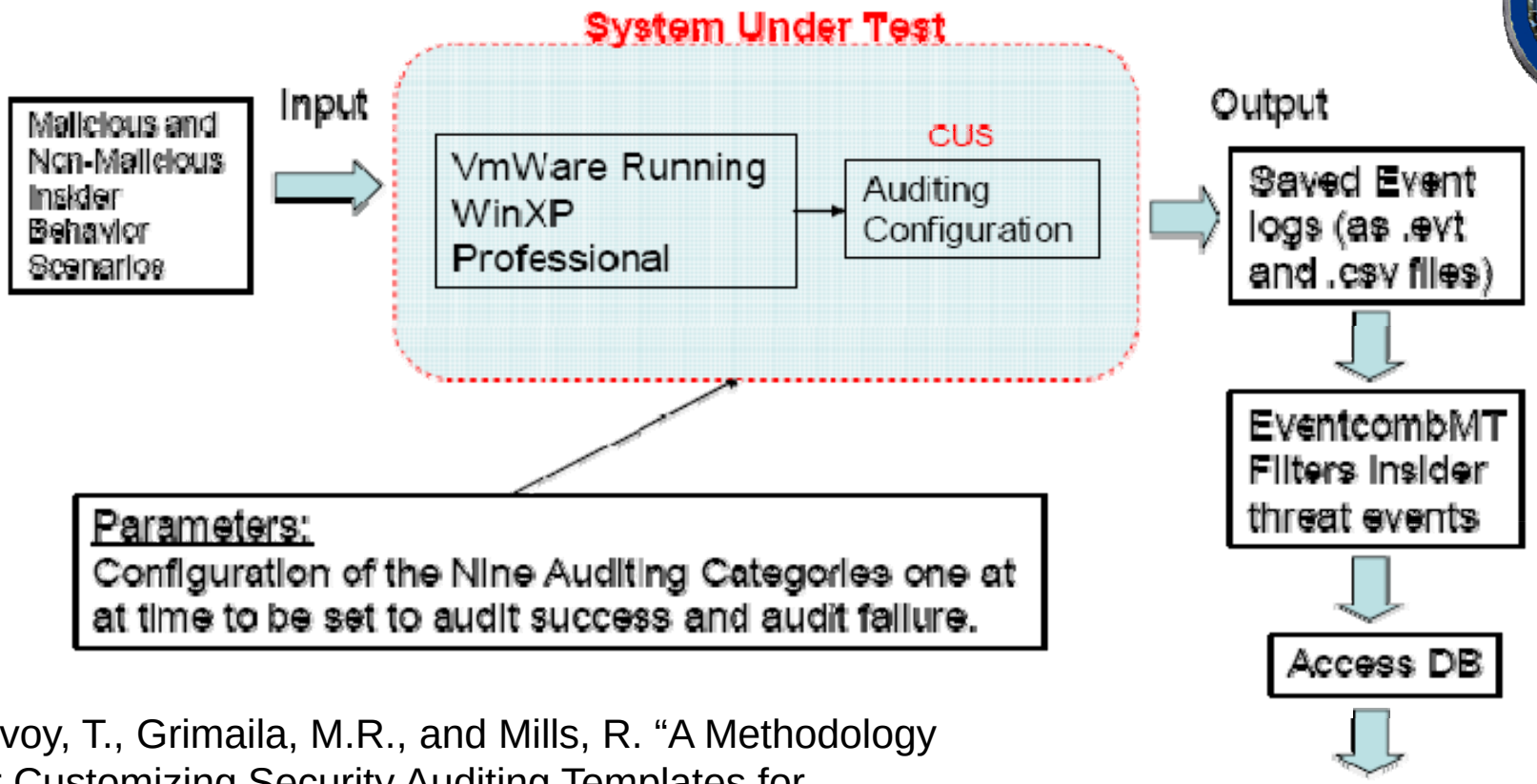
Application Server Logs

- Application programs often log status messages
- Logged events will typically include:
  - Unhandled application exceptions
  - Application errors
  - Loader problems (references to classes that are not available)
  - System error messages passed to application
  - Other implementation dependent items
  - Varies greatly depending upon the applications involved… but can provide valuable information

Web Server Logs

- Network accessible resource
- Commonly logged information includes:
  - Host information
  - Authentication information
  - Time / Access information
  - Status information
  - Resources accessed
  - Success / Failure
- Apache and Microsoft IIS web servers log similar parameters

Example: Windows XP Logging



Levoy, T., Grimaila, M.R., and Mills, R. "A Methodology for Customizing Security Auditing Templates for Malicious Insider Detection," Proceedings of the 8th International Symposium on System and Information Security (ISSIS 2006); Sao Jose dos Campos, Sao Paulo, Brazil; Nov. 08-10, 2006.

Figures of Merit

Win XP Security Log Categories

Category	Switch	Audit Events
Account Login	S1	Account Login Success
	F1	Account Login Failure
Account Management	S2	Account Management Success
	F2	Account Management Failure
Directory Service Access	S3	Directory Service Access Success
	F3	Directory Service Access Failure
Login	S4	Login Success
	F4	Login Failure
Object Access	S5	Object Access Success
	F5	Object Access Failure
Policy Change	S6	Policy Change Success
	F6	Policy Change Failure
Privilege Use	S7	Privilege Use Success
	F7	Privilege Use Failure
Process Tracking	S8	Process Tracking Success
	F8	Process Tracking Failure
System Events	S9	System Events Success
	F9	System Events Failure

Win XP Security Logging Policy Templates

Audit Setting	Windows XP® Out of the Box	*NIST EC* Template	*NIST SSLF* Template	*NSA SNAC* Template
Audit account logon events	No Auditing	Success	Success, Failure	Success, Failure
Audit account management	No Auditing	Success	Success, Failure	Success, Failure
Audit directory service access	No Auditing	No Auditing	No Auditing	No Auditing
Audit logon events	No Auditing	Success	Success, Failure	Success, Failure
Audit object access	No Auditing	No Auditing	Failure	Failure
Audit policy change	No Auditing	Success	Success	Success, Failure
Audit privilege use	No Auditing	No Auditing	Failure	Failure
Audit process tracking	No Auditing	No Auditing	No Auditing	No Auditing
Audit system events	No Auditing	Success	Success	Success, Failure

Malicious Insider Scenarios

Scenario	Description
1a	Attempt to Circumvent Auditing (clear Audit Logs) as Admin
2a	Attempt to access user's folder as Admin (User's Folder has no *SACL*, but file inside does. Admin has access to folder and file)
3a	Attempt to access a user's file successfully as admin with *SACL* (Admin has access to file, auditing is enabled in *SACL*)
4a	Attempt to access a user's file unsuccessfully as admin with *SACL* (Admin Does NOT have access to file, auditing is enabled in *SACL*)
5a	Attempt to access a user's file as admin with *SACL* by changing user's password and logging in as the user
6a	Attempt to access a user's file as admin by creating an admin account to use to mask attempts
7a	Attempt to access a user's file as admin by changing a user account to admin to use to mask attempts
8a	Attempt to access a user's file by changing a user account to admin by placing into admin group to use to mask attempts
9a	Attempt to access a user's file as admin by attempting to create a hard link
10a	Attempt to access a user's file as admin with an *SACL* by attempting to create a hard link
11u	Attempt to access another user's folder as user (User's Folder has no *SACL*, but file inside does. Admin has access to folder and file)
12u	Attempt to access another user's file as user successfully with *SACL* (User has access to file, Auditing is enabled in *SACL*)
13u	Attempt to access another user's file as user unsuccessfully with *SACL* (User does not have access to file, Auditing is enabled in *SACL*)
14a	Attempt to access a user's file as admin with *SACL* by changing ownership of the file
15a	Attempt to access a user's file as admin by attempting to Change Security Policy
16u	Attempt to guess Admin password by rebooting into safe mode and guessing admin password
17u	Access a user's file by booting with a boot CD and deleting password
18u	Run a root kit

Preliminary Definitions

An auditing configuration, AC_X, consists of a set of M binary switches specific to configuration X:

$$AC_X = \{S_1, S_2, ..., S_M\}$$

The normal (non-malicious) scenario set, NS_Y, consists of Y unique scenarios each containing normal user activity:

$$NS_Y = \{NS_1, NS_2, ..., NS_Y\}$$

The malicious scenario set, MS_Z, consists of a set of Z malicious scenarios each containing activities that are malicious:

$$MS_Z = \{MS_1, MS_2, ..., MS_Z\}$$

Metrics

The detection coverage metric, *DC*, is calculated as the sum of detected scenarios, MS_D, divided by the total number of malicious scenarios, MS_Z, in the malicious scenario set for a given audit configuration, AC_X:

$$DC\big|_{AC_X} = \frac{|\,MS_D\,|}{|\,MS_Z\,|}$$

The Logging Cost metric, *LC*, is the total cost for a given audit configuration, AC_X, and calculated as the sum of all events in the security log for that configuration after both the normal and malicious scenarios have been executed:

$$LC\big|_{AC_X} = E_N\,\big|_{AC_X} + E_M\,\big|_{AC_X}$$

The False Positive Cost metric, FPC, for a given audit configuration, AC_X, is calculated as the sum of critical events in the security log file after only non-malicious scenarios are executed.

$$FPC\big|_{AC_X} = \sum E_{CRITICAL}\,\big|_{MS_N}$$

Figure of Merit

The *FOM* consists of three summed weighted terms: The first term contains the detection coverage. The second term contains the ratio of the logging cost for the given audit configuration and the total events when all auditing switches are enabled. The third term contains the ratio of the false positive count for the given audit configuration and the logging cost for the given audit configuration. Organizations can assign values for *W1, W2, and W3* based on their assessment of the importance of detection coverage, logging cost, and false positives.

$$FOM\Big|_{AC_x} = W_1 \text{ x } DC - W_2 \text{ x } \left[\frac{LC|_{AC_x}}{E_T}\right] - W_3 \text{ x } \left[\frac{FPC|_{AC_x}}{LC|_{AC_x}}\right]$$

Non-Malicious Scenario Event Count by Switch

The table below shows the event cost, scenario detection, and false positive counts for the normal (non-malicious) scenario set by switch. The event totals were derived by repeating the events generated during the 15-minute activity contained in the normal scenario 160 times. The NFPC row shows a count by switch that shows the number of critical events that were detected but were false positives.

Of particular interest in this data is that switch S8 has the highest cost of 13,760 events. This can be explained by the auditing of program starts and terminations found in the normal user activities. Several switches have no cost (F1, S2, F2, S6, F6, F7, S9, and F9) because the normal scenarios did not contain activities that would result in an audited event.

	S1	F1	S2	F2	S3	F3	S4	F4	S5	F5	S6	F6	S7	F7	S8	F8	S9	F9
Events	160	0	0	0	480	160	480	160	480	1120	0	0	160	0	13760	320	0	0
NFPC	0	0	0	0	0	0	320	160	160	1120	0	0	160	0	6080	0	0	0

Malicious Scenario Event Count by Switch

	S1	F1	S2	F2	S3	F3	S4	F4	S5	F5	S6	F6	S7	F7	S8	F8	S9	F9
1a	**1**	0	**1**	0	0	0	**1**	0	**1**	0	**1**	0	**4**	0	**2**	0	**2**	0
2a	0	0	0	0	0	0	0	0	**15**	0	0	0	0	0	2	0	0	0
3a	0	0	0	0	0	0	0	0	**62**	0	0	0	0	0	2	0	0	0
4a	0	0	0	0	0	0	0	0	0	**108**	0	0	0	0	0	0	0	0
5a	1	7	**2**	0	0	0	6	6	86	0	0	0	1	0	35	0	0	0
6a	3	7	**6**	0	0	0	11	7	21	0	0	0	11	0	88	0	0	0
7a	2	8	**1**	0	0	0	7	8	33	0	0	0	10	0	47	0	0	0
8a	0	1	**1**	0	0	0	9	2	17	0	0	0	10	0	46	0	0	0
9a	3	2	0	0	0	0	2	1	**6**	0	0	0	5	0	12	0	0	0
10a	0	1	0	0	0	0	2	0	0	**1**	0	0	0	0	12	0	0	0
11u	1	0	0	0	0	0	2	1	0	0	0	0	1	0	10	0	0	0
12u	0	0	0	0	0	0	0	0	**54**	0	0	0	0	0	3	0	0	0
13u	0	0	0	0	0	0	2	0	0	**108**	0	0	0	0	9	0	0	0
14a	0	1	0	0	0	0	4	1	**76**	78	0	0	13	0	22	0	0	0
15a	**2**	1	**1**	0	0	0	**3**	1	**5**	0	**1**	0	**6**	0	**13**	0	**1**	0
16u	1	4	0	0	0	0	**9**	**4**	12	3	1	0	39	0	40	0	28	0
17u	2	0	0	0	0	0	35	0	**73**	0	12	0	22	0	36	7	**16**	0
18u	1	1	0	0	0	0	2	1	0	0	11	0	8	0	30	0	0	0
Events	17	33	12	0	0	0	95	32	461	298	26	0	130	0	409	7	47	0
MFPC	0	0	2	0	0	0	30	28	27	10	0	0	7	0	216	0	0	0

Comments on Malicious Scenario Event Counts

- S5 detected the largest number of the scenarios. This is mainly because S5 audits for successful object access and most of the malicious scenarios were based on the premise of attempting to access other user's files.
- Switch S2 detected the next largest number mainly because it detected abuse of administrator privileges in which accounts are created, destroyed, or modified.
- Two scenarios, 1a and 15a, were detected by several switches.
- One scenario, 11u, was not detected at all due to the fact that if the folder object does not have a SACL defined, logging of a failed access attempt of the object will not occur even if the appropriate switch is enabled.

Pros of Scenario Approach

- Low false positive rate due to explicit definition of malicious behavior
- Extremely customizable… organizations can define specific scenarios of interest
- Development and refinement of a library of malicious behavior scenarios over time
- Accounts for the costs and benefits involved when collecting security events in audit logs
- Objectively evaluation of different insider threat detection systems
- Educates security personnel on capabilities and limitations of the auditing process

Cons of Scenario Approach

- Requires organization to define the malicious behavior it deems important
- Requires explicit definition of insider behavior
- Requires a substantial investment in time and effort to develop a realistic set of scenarios that accurately characterize malicious user behavior
- Must periodically maintain and update scenarios
- Certain scenarios are organization specific and cannot easily be shared across different organization units

Conclusions

- It is impossible to determine how well you're doing with a detection system unless you:
  - Clearly define the set of scenarios of interest
  - Quantify the costs / benefits of the detection system
  - Quantify the importance of the system(s) being monitored
  - Evaluate the TCO for the detection system
- Challenging part is developing and maintaining the set of realistic scenarios
- Log analysis is a discovery process, not just a turn key solution… it needs to be resourced.

Questions

Michael R. Grimaila, PhD, CISM, CISSP, IAM/IEM
Department of System and Engineering Management
Center for Cyberspace Research
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
Michael.Grimaila@afit.edu