A HOST REFERENCE MONITOR APPROACH TO THE PROBLEM OF HUMAN AND PROGRAMMATIC INSIDER THREAT TO COMPUTER INFORMATION SYSTEMS

Craig Chamberlain, Donato Buccella, Daniel Geer, Sc.D.

Verdasys, Inc.

950 Winter St. Suite 2600

Waltham, MA 02451

craig@verdasys.com , dbuccella@verdasys.com , geer@verdasys.com

ABSTRACT

Insider threat is a significant problem area in information security due to the complexity of the problem and the lack of effective technical controls designed to address the insider threat problem in the commercial software sector. The security reference monitor in its strictest definition has never been fully implemented in COTS operating systems; such an implementation may or may not be practical or desirable at the time of this writing. We propose the reference monitor concept can be applied to the problem of insider threat mitigation in commercial software environments with greater effectiveness than conventional technical controls. We present Digital Guardian, a technology developed by Verdasys, Inc., as mechanism for detection and interdiction of insider misuse of information systems, including information leakage or theft, using a reference monitor model adapted for implementation in commercial operating systems. Four insider threat scenarios are detailed to illustrate how technical controls and countermeasures could be implemented using the reference monitor model.

INTRODUCTION

The problem of insider threat presents significant challenges to the security and integrity of computer information systems. Commercial information systems are often designed with the expectation that authorized users are trustworthy; these systems consequently are vulnerable to misuse. Many organizations employ administrative controls in the form of security policies and legal agreements but effective technical controls have been slow to appear. Traditional security measures such as access controls and firewalls are largely designed around threats by external or unauthorized users and are ineffective against the insider threat. Intrusion detection systems are similarly designed with attack by outsiders in mind and have the disadvantage of relying on signature databases which makes them inherently reactive in nature. VPNs and encryption technologies protect information in transit but are partial insider threat defenses at best and can actually make the problem more difficult by blinding network-layer defenses. Many applications and operating systems have technical controls against misuse in the form of internal auditing and logging capabilities; application level auditing is limited to the application instance(s) and provides little or no visibility once data leaves the application. Operating system auditing can provide visibility into data movement but in many cases suffers from poor tamper resistance and usability. A technically sophisticated insider with detailed knowledge of auditing and logging

controls can often bypass or render auditing ineffective (particularly when it has little or no tamper-resistance). Digital rights management also faces usability and scalability questions and is inherently document-centric; information which travels outside a protected document is still at risk. Role-based access control and the two-person rule can help mitigate collusion or intentional sabotage by trusted users of an information system but are largely ineffective against trusted insiders. New approaches utilizing strong technical defense, prevention and detection technologies are needed. We propose using the reference monitor concept for constructing these control mechanisms as it has strong applications in the insider threat space.

THE REFERENCE MONITOR

The concept of the reference monitor, first introduced by James Anderson's 1972 *Computer Security Technology Planning Study*, provides for mandatory enforcement of a security policy against the actions of subjects (users or programs) and objects (programs or data). [AND72]

The reference monitor model can be used to construct technical controls against insider misuse by subjecting all transactions taking place within an information system to mandatory scrutiny and comparison against a security policy which describes which types of transactions tend to increase risk. Transactions are audited and approved or denied depending on whether they tend to violate security policy. We propose defining a transaction as an attempt by a subject to reference an object or objects in order to process data or in support of processing data. Transactions include all operations that involve or support the processing of data including process executions, file creation and modification, file movement within and between hosts, network connections and file transfers, print operations and application data exchange (e.g. copy and paste) operations. Transactions may also constitute access to or manipulation of individual records or fields within a database or a database driven application.

Transactions with the potential to exfiltrate data from a host are of particular interest when considering insider misuse as these constitute the major pathways for data to exit an information system and pass beyond the control of its owners. The movement of data to uncontrolled systems such as removable media or foreign networks, for example, are the two major pathways to information loss within organizations. This type of data movement may or may not be desirable depending on contextual factors such as the destination, the privilege level of the user and the user's intentions. With a reference monitor in place providing transaction level scrutiny a sufficient level of detail and control becomes available to detect and/or prevent the inappropriate movement of information. With the availability of rich detail about what users are actually doing it becomes possible to automate contextually appropriate and intelligent policy enforcement decisions without preventing legitimate users from performing their duties in many types of scenarios. In situations where automated policy enforcement is impractical or undesirable detection and response are still available.

DIGITAL GUARDIAN: AN IMPLEMENTATION OF THE REFERENCE MONITOR CONCEPT FOR WINDOWS

We have developed a commercially relevant approximation of the reference monitor within the limits of COTS operating systems in order to apply the reference

monitor concept to commercial information systems. Digital Guardian (DG), a software product currently available for the Windows platform, is a technology consisting of client agents controlled by a central server. The agents perform kernel-level transaction auditing and policy enforcement and report audit data to a collection server for analysis.

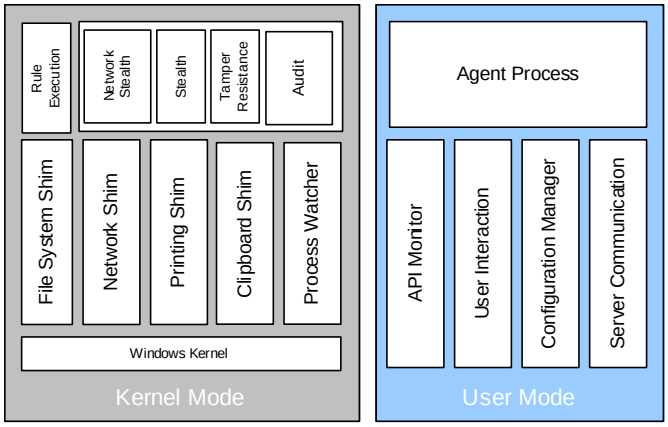

Figure 1. The Digital Guardian Agent Functional Components

Policy enforcement is defined and managed at the server. The agent is orthogonal to Windows and operates without regard for user privilege levels which provides for mandatory policy enforcement. Tamper resistance functionality prevents the agent from being disabled or circumvented.

The Digital Guardian (DG) Agent

The agent is the client side software component of the DG system. The DG agent consists of kernel level operating system sensors or shims and a user mode process. The shims intercept input / output (I/O) transactions which involve or support the processing of data on their way to the kernel, record them for the audit trail, and permit or deny them to take place after verifying they do not violate any defined policies. System I/O operations are collected by the shims and reported to the user mode agent process for aggregation and delivery to the server. There is an increase in processing cycles per transaction required by the shims but the performance penalty is negligible on relatively recent Intel based hardware and does not interfere with or adversely affect normal operation.

The user mode agent receives transaction audit data from the shims, communicates with the server to report audit data and processes configuration instructions or security policies. The agent is autonomous and fully functional while server communication is unavailable; all policy and configuration information is locally cached. The collected audit data is organized by system operation and time period and is stored in an encrypted and signed XML message format. The agent transmits the audit data to a collection server using a secure proprietary protocol tunneled over HTTP using two-way authentication, payload encryption and digital signatures. The communication subsystem implements a fault tolerant transaction queuing and caching mechanism with transmission retries. The user mode agent component is also responsible for event consolidation using an aggregation algorithm. For example, low level operations such as file reads and writes can be summarized as file edits to make the audit reports more concise and human-readable.

Operating System Shims

The kernel level operating system shims are implemented using the Microsoft Windows filter driver interface. The file system shim is the software middleman between the operating system's file system drivers (NTFS, FAT, CDFS, etc.) and user applications. When a file I/O request is made by an application the file system shim intercepts and retrieves the parameters of the request such as the path of the file being accessed, number of bytes, username, etc. and then asks the rule execution subsystem whether to prevent the file operation or let it

continue. The network system shim similarly intercepts all user application requests to the network communication layer, records them, and checks to see if they violate security policy. Monitoring of writes to CDs and DVDs is implemented using a specialized subsystem shim. This CD / DVD shim is also capable of blocking CD/DVD burning operations.

Process Watcher

The process watcher is a kernel level module. This module intercepts system process level OS functionality such as creation and termination of processes and records the process operation parameters. A kernel side process watcher works in tandem with the API monitor to intercept calls and parameters being requested by user applications.

Rule Execution Engine

The Rule execution engine is responsible for performing real-time policy enforcement on the transactions taking place in the I/O subsystems. The rule engine compares events observed by the shims against a security policy; this security policy consists of a set of rules in a custom language defining which types of events are eligible for policy enforcement and should be prevented from taking place and/or flagged for urgent attention. The rule execution system also determines which events are defined as "noise" or unwanted events with no significance from a security standpoint and discards them from the audit trail. The rule execution engine implements a sophisticated decision tree that optimizes the checking of rules against system operations in a per user basis.

Tamper Resistance and Stealth

The agent prevents its components from being disabled or terminated using its own policy enforcement capabilities. It intercepts and blocks attempts to modify or delete its files or registry entries. It prevents itself from appearing in the output of process lists such as the Windows task manager. It blocks attempts to stop the agent processes and filter drivers. An optional "stealth" mode further obscures the presence of the agent by using API hooking and file system driver output redaction to conceal the presence of the agent files and registry keys.

The Digital Guardian Server

The Digital Guardian server is implemented as a set of database driven ASP .NET web applications and several Windows services. The server component provides three major functions:

- Reporting and analysis of the event audit data reported by the agents
- Management of policy enforcement
- Installation, management and configuration of the agent population

One of the web applications receives audit data from the agents and the other, the Digital Guardian Management Console (DGMC), provides the server's user interface. The DGMC provides an interface for querying and viewing the audit data collected by the agent population. It provides several reports with different levels of detail and summarization. The reporting interface has the ability to summarize and detail the movements of data and data-processing transactions performed by users on agent protected systems. For example, file, network, print, clipboard and process launch operations can be summarized and analyzed in detail. The audit and reporting

capabilities are essentially a near real-time forensic tool for Windows systems which provides broad and deep visibility into the movement of data and the use or misuse of information systems.

The DGMC is also where security policies are defined and assigned to users. Policies are collections of rules in a custom language which reflect security policy and describe specific kinds of transactions that take place in the I/O subsystems and present risk of information loss or tend to violate security policy. The advantages of enforcing policy at the kernel layer include the availability of detailed information about how data is flowing and the smaller number of possible transactional scenarios to consider as opposed to the limitless possibilities present in the application vulnerability space. The syntax of this XML language follows a specific XML schema and uses standard naming conventions for subsystems, properties, and variables. The rules also define what action should be taken by the agent when such a transaction is attempted. A variety of actions is can be taken (which are not mutually exclusive and can be combined):

- The agent can permit or deny the transaction.
- The agent can warn the user or prompt them for input.
- The agent can silently message a security analyst that a user has performed some noteworthy action.

Policies can be applied to individual users or groups and specific rules can be disabled on and user or group basis which provides for policy reusability and implementation flexibility. Policy assignments can be made using Active Directory user hierarchies, local Windows user accounts, or arbitrary groups of users or computers.

APPLICATIONS FOR THE REFERENCE MONITOR MODEL TO INSIDER THREAT

Insider Threat Modeling

As the study of insider threat continues to advance more numerous and sophisticated models of insider threats to information systems are needed. The data collected by the reference monitor can provide technical and behavioral profiling which can be mined in support of efforts to construct such profiles.

Insider Scenario One: Novice Insiders; Populations With Role-Limited Access and Privileges

Neumann [NEW99], Anderson [AND00] and Caloyannides & Landwehr [CAL00] advocate approaching the insider misuse problem with more sophisticated and granular technical controls. Organizations where policy follows the principle of least privilege are particularly well-suited to the implementation of more precise technical controls using a reference monitor security model. Using such a model, technical controls against insider theft can be constructed by blocking potential avenues for information to exit controlled systems. Such environments typically feature a semi-trusted population that requires access to sensitive information but whose responsibilities do not require unlimited levels of access or privileges. Examples of such environments include call centers, outsourcing providers, military and/or government facing organizations, contractor workspaces or any workplace featuring a semi-trusted population with access to sensitive or regulated data.

A least privilege environment makes the job of the security policy analyst considerably easier. Rather than anticipating all possible ways an information system can be misused, the analyst simply has to define how it may be used with an eye toward closing potential pathways of information leakage. For example, the following types of policies could be employed in order to prevent theft or misuse or data by insiders:

- Sensitive files, or perhaps any file, cannot be written to removable media. This is probably the major exit route for data in many organizations.
- Files residing on sensitive servers cannot be copied to other disks or file systems.
- Sensitive files, or possibly any files, cannot be uploaded to uncontrolled servers on foreign networks.
- Sensitive files can be read only by trusted processes such as client applications, backup agents and system maintenance utilities.
- Trusted processes enabled to read sensitive files cannot write files to locations other than the designated location(s) for sensitive data. This prevents a trusted process from exporting sensitive data.
- Trusted client processes can copy and paste data within their own process memory space but cannot copy and paste data into another process's memory space (or a file opened by another process).
- Only trusted processes may connect to sensitive servers, reducing the attack surface by blocking connections from untrusted or rogue processes including tools that could be used to exploit vulnerabilities and compromise servers.
- Connections to foreign database, application, file and print servers may not be made. This reduces avenues for information to move to uncontrolled networks.
- Only trusted processes can make outbound connections to foreign networks, particularly on ports 80 and 443. This provides protection against a Trojan horse or remote control application exploiting access to the world wide web in order to communicate information to foreign networks.
- Only trusted VPNs may be used.
- Untrusted Internet facing applications cannot write or rename executable files on the file system or modify system paths; this reduces exposure to Trojan horse, rootkit and spyware programs attempting to infiltrate a host by exploiting a vulnerable Internet facing application (or a naive user).

Additional policy enforcement can be designed as necessary to address additional avenues of information loss as they are discovered. In the reference monitor model, all transactions can be subjected to auditing regardless of whether they violate policy so if a new method of information leakage becomes available against which existing controls are ineffective sufficient data should be available to determine the nature of the data exit point and construct new technical controls.

Insider Scenario Two: Programmatic Insiders; Trojan Horse Detection And Interdiction

Trojan Horses are essentially entry level insiders in programmatic form. We classify them as entry level because their intelligence and capabilities are limited by virtue of the fact that they are computer programs and cannot exceed the sum of their instruction set. Trojan horse programs may, however, use ingenious and creative methods to escape detection. For example, network avenues of exfiltration are increasingly limited to ports 80 and 443 in an effort to stem the tide of outgoing information. In

most organizations, incoming network connections are blocked completely, making the conventional remote access Trojan (RAT) useless. Ports 80 and 443, necessary to provide access to the world wide web, are the Achilles heel of modern network security and are commonly exploited by Trojan Horse programs to tunnel information over what appear to be innocent HTTP and SSL sessions. Detection of this at the network layer is confounded by the fact that the TCP/IP and HTTP protocols have no provision for identifying the process which made an HTTP request (The HTTP protocol does provide the user agent string but this is unauthenticated). Detection is further frustrated by the fact that encryption makes any meaningful real-time inspection of the traffic impossible. This problem can be solved with a host reference monitor enforcing a mandatory whitelist policy that permits legitimate applications and web services to make outbound connections on ports 80 and 443 but denies requests from unauthorized or unauthenticated processes. Applications can be positively identified using hashes of the program files. This kind of policy is better implemented with a reference monitor model than a host firewall as it needs to have both mandatory enforcement and tamper-resistance to be effective against user manipulation and hostile code.

At some point as the Trojan horse arms race escalates and detection becomes increasingly expensive it becomes necessary to move to a prevention model. Trojan horse programs that use stealthy methods such as DLL injection to hide their presence and make their actions appear to be those of a legitimate process present significant challenges in detection. Such programs generally enter a host through an Internet facing application such as an email client, web browser or instant messaging application. A policy preventing such Internet facing applications from writing to system paths or writing (and / or renaming) executable files on the file system would help to close many of these avenues of infection and make the host more resistant to infection by hostile or invasive programs.

Insider Scenario Three: Intermediate Level Insiders

We define intermediate level insiders as those having medium to high privilege levels with unpredictable levels of access to information systems and broad responsibilities. This combination may make them unsuitable for a least privilege security model and the restrictive policy enforcement scenarios described in the previous sections. Such automated policy enforcement models, while effective at preventing information loss, may present unacceptable levels of restriction to the intermediate user or may get underfoot and interfere with the performance of their duties.

Automated policy enforcement can often still be performed when such users are working with sensitive data. However, there may be many cases where an intermediate level user requires the ability to access and work with sensitive data without restriction. Schneier [SCH03] points out that prevention is sometimes impractical or undesirable due to costs and other trade-offs; he advocates designing security systems to feature detection and response capabilities in addition to prevention in order to maximize their effectiveness. The reference monitor model facilitates such a defense-in-depth model by providing for detection and response when prevention is ineffective or impractical. We provide for detection and response with an audit trail of transactions and data movement which becomes actionable in the event information is misused. Rapid detection can be provided for by messaging a security administrator

when a transaction occurs which the reference monitor recognizes as potential misuse or exposing sensitive information to unacceptable risk. With such a forensic-quality audit trail investigation and incident response is considerably more effective and likely to succeed. Anderson [AND00] proposes the use of dynamic warning banners in combination with highly granular access controls to create a deterrent effect. This is also possible in our implementation by presenting a dialog box whenever a user takes action that presents unnecessary or unacceptable risk of information loss. Such a banner can be interactive, presenting information on security policy or allowing the user to input a response to be included in the audit data.

An additional challenge of intermediate users is that they may have above average technical sophistication. Randazzo [RAN04] found that 23% of insiders held technical roles in their organizations and 17% had high privilege levels such as administrator or root access; however, Randazzo also found that 87% of the insider attacks they studied neither required nor employed technical sophistication. The question becomes whether technically sophisticated insiders actually exist in smaller numbers or their rate of detection is low. Magklaras [MAG] noted technical sophistication and the potential to commit insider attacks on information systems are correlated in several studies. At least two approaches to this problem exist; the use of very stealthy monitoring and the use of very obvious monitoring.

Covert vs. Overt Monitoring

There are at least two schools of thought with respect to audit and monitoring. One is to make audit policy and require that all systems be monitored. Overt universal monitoring makes prevention of information theft less urgent if it creates a deterrent effect through the perceived increased risk of detection.

Overt monitoring, optionally combined with dynamic bannering, could possibly help mitigate what Anderson et. al. [AND04] call the “dynamic trigger hypothesis” of insider threat. In this model, organizations experience the “detection trap”, the “trust trap” and the “unobserved emboldening” syndromes. The “detection trap” is an interesting phenomenon where a lack of detection capability resulting in a low rate of incident detection leads to a generalized disinterest in improving security capabilities and onset of the “trust trap”. The “trust trap” is a self-reinforcing delusion where organizations with poor incident detection capabilities ignore questions of security and exist in a state of considerable vulnerability after mistakenly concluding, in the absence of detection data, that no incidents have taken place and the organization’s members are highly trustworthy. This in turn gives rise to “unobserved emboldening” where the lack of detection capability and the state of vulnerability creates perception of low risk on the part of potential insiders and creates a “reinforcing cycle of emboldening” to which an organization may be more or less blind until or unless a significant incident occurs. [AND04] Melara, Gonzalez and Cooke [MEL03] describe a similar phenomenon they call the “dynamic hypothesis” which predicts increased frequency of insider misuse when high levels of vulnerability are perceived to exist within information systems.

Another approach is to conceal the presence of monitoring systems in order to covertly detect data leakage or theft and then address the problem through investigation and incident response. If the problem is detected in time this method may be successful in identifying insiders before irreparable damage is done. As described

previously, detection can be accelerated by messaging a security administrator when a transaction occurs that violates policy or presents high risk of information leakage. For example, the unnecessary movement of sensitive data to removable media or foreign networks could be flagged for urgent attention.

Early Detection Through Behavioral Profiling

Schultz [SCH02], Magklaras [MAG02], Anderson et. al. [AND04], Melara, Gonzalez and Cooke [MEL03], Wood [WO00] and Randazzo [RAN04] describe the use of anomalous and/or preparatory behavior detection as a method of early identification of potential insiders. In a study of insider incidents in the financial services arena, Randazzo [RAN04] found that preparatory behavior was present in 35% of incidents. An insider may engage in unusual or anomalous activity either to test for the presence of technical controls or detection mechanisms or to make preparations for an actual attack. Wood [WO00] describes a model of predictable insider behavior where attacks are prefaced by target identification and reconnaissance. Examples of reconnaissance activities could include profiling and mapping servers and networks, operating a network sniffer, attempting to access password files or use password cracking tools, probing for exploitable vulnerabilities and performing trial runs of information exfiltration. An insider may use these methods to try and determine an organization's ability to detect suspicious behavior before proceeding to attempt the exfiltration of the actual target information. Schultz [SCH02] notes that mistakes made during such trial runs present detection opportunities and proposes using the presence of such "meaningful errors" as an indicator of insider activities. Schultz also proposes using what he calls "deliberate markers" as a method of insider detection; these are artifacts or events intentionally created by the insider with the intention of communicating a message of some sort. An example of a deliberate marker Schultz gives is a hostile email message whose origin is ambiguous.

When the target is an application or database server, the insider may perform anomalous modifications or interact with it in an unusual way in order to create opportunities to compromise or subvert it. Examples of such preparations could be detected in the form of unexplained modification or replacement of program files or accessing servers or applications using low-level diagnostic or debugging tools when no maintenance has been scheduled or authorized.

Insider Scenario Four: Advanced Insiders

The advanced insider is one of the most challenging scenarios for the security analyst and potentially the most costly type of security incident an organization can experience. Advanced insiders may be employed by an organization as part of a competitive intelligence effort; if so they are likely focused on a specific information target of interest and will ignore targets of opportunity. They may be highly skilled both technically and socially and be capable of maintaining high levels of access and trust within an organization. Advanced insiders may have nearly unlimited access to systems and can very likely escape detection in many cases by subverting or bypassing conventional auditing and monitoring mechanisms. They may be undetectable by psychological profiling methods and display none of the insider personality traits described by Shaw [SHA98] such as introversion, social frustration, computer

dependence, ethical flexibility, reduced loyalty and attitudes of entitlement. Randazzo [RAN04] found that only 15% of insiders had been thought of as difficult to manage, 19% as disgruntled, 4% as untrustworthy and only 27% of insiders had engaged in behavior that previously called attention to them. Neumann [NEU99] theorizes that a sophisticated insider could evade anomaly detection mechanisms by taking care to display behavior that generally shows no statistically significant deviation from the norm. Application and system level hardening and auditing may be effective in many cases where advanced insiders are involved; Anderson [AND00] proposes monitoring and restricting application behavior to reduce exposure to insider misuse. Additional approaches include stealth monitoring and deception technologies.

Stealth Monitoring

The covert reference monitor is an option for detection of advanced insiders. In this scenario the investigative goals may be expanded to identification of the insider's targets and which, if any, organization they are acting on behalf of. These kinds of advanced investigations may be supporting litigation or competitive intelligence efforts. In some cases it may not be desirable to place a highly sophisticated advanced insider under direct technical surveillance if strong risk exists that they will be capable of determining this. If they learn their usage is being monitored they may not engage in the activity an advanced investigation needs to observe in order to make progress.

Instrumented Deception Technologies

Anderson [AND99] proposes using "deception technologies" to identify insiders and their targets of choice. Such technologies typically take the form of a honeypot. Spitzner [SPI03] proposes using honeypots, unadvertised decoy servers with fictitious sensitive data which appears genuine, to identify insiders. Such a server has no legitimate users; anyone accessing or removing data from it is very likely an insider (provided they're not an actual network intruder). Remote detection of a covert surveillance engine in the form of a stealthy reference monitor is non-trivial, even for the technically sophisticated user, and requires specialized tools. If the server appears sufficiently genuine it may not even inspire suspicion on the part of the insider. In any event, performing detailed system analysis in support of remote detection of a stealthy kernel level reference monitor presents the risk of attracting attention. In An Insider Threat Model for Adversary Simulation, Wood [WO00] notes, "the insider is very risk-averse. Their ultimate defeat is to be discovered before they have mounted a successful attack".

Spitzner [SPI03] also describes the use of honeytokens, which are false digital objects such as fabricated sensitive files, to identify insiders. This method could be used as an alternative to honeypots by sprinkling false documents which appear to be highly sensitive throughout an information system or server population.

With the instrumented server's audit trail of access to the false resources in hand all that is left is to determine where the trail leads and to whom. Corroborating with other forms of auditing such as building access logs and surveillance cameras where possible can be helpful in making a positive identification.

It may be desirable in some cases to place the false information just out of reach, perhaps with file permission settings that deny access to the suspect population; if the false information is then taken forcefully you have almost certainly identified an

insider and not a user simply on a careless exploration.

Counterintelligence

Once the advanced insider has been identified, the investigators have a choice: they can begin incident response or they can begin a counterintelligence operation. Wood [WO00] advocates counterintelligence as a countermeasure to insider threat. Counterintelligence is the traditional response to insider threat in the world of the military and intelligence services where extensive experience with such things exists. Depending on the stakes and the nature of the adversary it may be advantageous to feed the insider disinformation in order to frustrate their competitive intelligence effort. A counterintelligence operation may yield valuable information about the nature of the information targeted for exfiltration and possibly the identity of the organization or individuals running the competitive intelligence effort. A limiting factor to this option in the non-government sector is that many commercial organizations may not have the experiential base or requisite knowledge to mount an effective counterintelligence or disinformation campaign. An effective disinformation campaign would also have significant costs associated with it and presents the risk of disinformation being re-introduced into the organization. Where this risk is present, disinformation may need to be of a nature such that re-introducing it would present risk of exposure to the insider(s).

A counterintelligence campaign that succeeds in preventing an information loss incident with a very high single-loss expectancy, however, could very well be cost-effective. Depending on the size of the potential loss and associated economic and political consequences it may be possible to seek and receive the benefit of experienced help from the government sector or other sources in these kinds of operations.

Logical Insiders

Neumann [NEU99] observed that the attacker who makes it past an organization's perimeter defenses is a logical insider and may pose technical challenges similar to those presented by the conventional insider. We believe the techniques described above in concert with traditional prevention and detection capabilities are suitable for application to the logical insider problem.

CONCLUSION

We have re-examined the value of a reference monitor in the world of the Internet, COTS products and a security situation where data protection must be host-based if it is to address insider and outsider attacks with equal effectiveness. We have provided motivating use cases and have further related these use cases to a representative sample of both the classic and the current literature. We would not be writing this were it not possible to reduce this theorizing to practice, which we have done thus confirming the theoretic material to which this paper is itself confined.

REFERENCES

[AND04] David F. Andersen, Dawn Cappelli, Jose J. Gonzalez, Mohammad Mojtahedzadeh, Andrew Moore, Eliot Rich, Jose Maria Sarriegui, Timothy J. Shimeall, Jeffrey M. Stanton, Elise Weaver, and Aldo Zagonel. "Preliminary System Dynamics Maps of the Insider Cyber-threat Problem", *Proceedings of the 22nd International Conference of the System Dynamics Society*, July 20-24, 2004 at Oxford, UK.

[AND72] J.P. Anderson. “Computer Security Technology Planning Study,” ESD-TR-73-51, Air Force Electronic Systems Division, Hanscom AFB, Bedford, MA.

[AND99] Robert H Anderson. “Research and Development Initiatives Focused on Preventing, Detecting, and Responding to Insider Misuse of Critical Defense Information Systems: Results of a Three-Day Workshop,” Rand National Defense Research Institute, Santa Monica, CA, 1999.

[AND00] Robert H. Anderson, Thomas Bozek, Tom Longstaff, Wayne Meitzler, Michael Skroch, Ken Van Wyk. “Research on Mitigating the Insider Threat to Information Systems - #2: Proceedings of a Workshop Held August, 2000,” CF-163-DARPA, 2000. Rand National Defense Research Institute. Santa Monica, CA. August 2000

[CAL00] Michael Caloyannides and Carl Landwehr, Mitretek Systems. “Can Technology Reduce the Insider Threat?” Published in *Research on Mitigating the Insider Threat to Information Systems - #2: Proceedings of a Workshop Held August, 2000*, CF-163-DARPA, 2000. Rand National Defense Research Institute. Santa Monica, CA. August 2000.

[HEU00] Richard J. Heuer, Jr. ”The Insider Espionage Threat” Published in *Research on Mitigating the Insider Threat to Information Systems - #2: Proceedings of a Workshop Held August, 2000.* CF-163-DARPA, 2000. Rand National Defense Research Institute. Santa Monica, CA. August 2000.

[MAG] G.B. Magklaras and S.M. Furnell. “A Preliminary Model Of End User Sophistication For Insider Threat Prediction In IT Systems” *Computers & Security*, In Press. Available online http://www.sciencedirect.com

[MAG02] G.B. Magklaras and S.M. Furnell. “Insider Threat Prediction Tool: Evaluating The Probability Of IT Misuse” *Computers & Security*, Volume 21, January 2002.

[MEL03] Carlos Melara, Jose Maria Sarriegui, Jose J. Gonzalez, Agata Sawicka, and David L. Cooke. “A System Dynamics Model Of An Insider Attack On An Information System” Paper read at the Proceedings of the 21st International Conference of the System Dynamics Society July 20-24, 2003, New York, NY. http://www.systemdynamics.org/conf2003/proceed/PAPERS/294.pdf

[NEU99] Peter G. Neumann. “The Challenges of Insider Misuse” Prepared for the Workshop on Preventing, Detecting, and Responding to Malicious Insider Misuse 16-18 August 1999, at RAND, Santa Monica, CA http://www.csl.sri.com/users/neumann/pgn-misuse.html

[RAN04] Marisa Reddy Randazzo, Ph.D., Dawn Cappelli, Michelle Keeney, Ph.D., Andrew Moore, Eileen Kowalski. “Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector,” CERT® Coordination Center, United States Secret Service and Carnegie Mellon University, 2004.

[SCH03] Bruce Schneier. *Beyond Fear: Thinking Sensibly About Security in an Uncertain World.* 2003. New York, NY: Copernicus Books.

[SCH02] E. Eugene Schultz. “A Framework For Understanding And Predicting Insider Attacks,” *Computers & Security*, Volume 21, June 2002.

[SHA98] Eric D. Shaw, Ph.D., Keven G. Ruby, M.A. and Jerrold M. Post, M.D. ”The Insider Threat To Information Systems,” *Security Awareness Bulletin No. 2-98*, published by Department of Defense Security Institute, September 1998.

[SPI03] Lance Spitzner. “Honeypots: Catching the Insider Threat,” 19th Annual Computer Security Applications Conference December 2003 Las Vegas, Nevada.

[WO00] Bradley J. Wood. “An Insider Threat Model for Adversary Simulation,” Published in *Research on Mitigating the Insider Threat to Information Systems - #2: Proceedings of a Workshop Held August, 2000*, CF-163-DARPA, 2000. Rand National Defense Research Institute. Santa Monica, CA. August 2000.