Software Engineering Institute

Insider Threat Control: Using Centralized Logging to Detect Data Exfiltration Near Insider Termination

Michael Hanley
Joji Montelibano

October 2011

TECHNICAL NOTE
CMU/SEI-2011-TN-024

CERT® Program

http://www.sei.cmu.edu

Carnegie Mellon

This material is based upon work funded and supported by the United States Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

Table of Contents

List of Figures

Acknowledgments

Special thanks to our sponsors at the U.S. Department of Homeland Security, National Cyber Security Division, Federal Network Security branch for supporting this work.

Also, special thanks to Will Schroeder, Tyler Dean, and Matthew Houy, graduate students from Carnegie Mellon University, for their assistance with developing the technical content in this work.

Abstract

Since 2001, the CERT® Insider Threat Center has built an extensive library and comprehensive database containing more than 600 cases of crimes committed against organizations by insiders. A significant class of insider crimes, insider theft of intellectual property, involves highly damaging attacks against organizations that result in significant tangible losses in the form of stolen business plans, customer lists, and other proprietary information. The Insider Threat Center's behavioral modeling of insiders who steal intellectual property shows that many insiders who stole their organization's intellectual property stole at least some of it within 30 days of their termination. This technical note presents an example of an insider threat pattern based on this insight. It then presents an example implementation of this pattern on an enterprise-class system using the centralized log storage and indexing engine Splunk to detect malicious insider behavior on a network.

1 Introduction

The CERT® Insider Threat Center, part of Carnegie Mellon University's Software Engineering Institute, maintains a database of more than 600 insider threat cases. Using the 86 cases of insider theft of an organization's intellectual property (IP), the staff of the CERT Insider Threat Center found that many insiders who stole their organization's information stole at least some of the information within 30 days before their termination. This technical note presents an example of an insider threat pattern developed based on that insight. Using the centralized log storage and indexing engine Splunk,[1] we show an example implementation of this pattern on an enterprise-class system.

Specific interest in this topic has led to a separate effort to develop a pattern language for expressing the insider threat problem and appropriate countermeasures. A CERT report is forthcoming on this topic.[2] This technical note contains an example rule, and the discussion of its implementation was derived from an early version of the forthcoming work.

® CERT is a registered mark owned by Carnegie Mellon University.

1 http://www.splunk.com. The authors of this paper do not endorse Splunk. The platform's use in this research reflects its availability to the authors.

2 Moore, A.; Hanley, M.; Mundie, D. A Pattern for Insider Threat Risk Mitigation (forthcoming). Software Engineering Institute, Carnegie Mellon University.

2 CERT Insider Threat Database

The CERT Insider Threat Center database currently contains more than 600 cases of actual malicious insider crimes. Our research has revealed that most crimes fit one of four categories:

- IT sabotage
- theft of intellectual property (IP)
- fraud
- espionage

Figure 1 breaks down the number of cases by category.

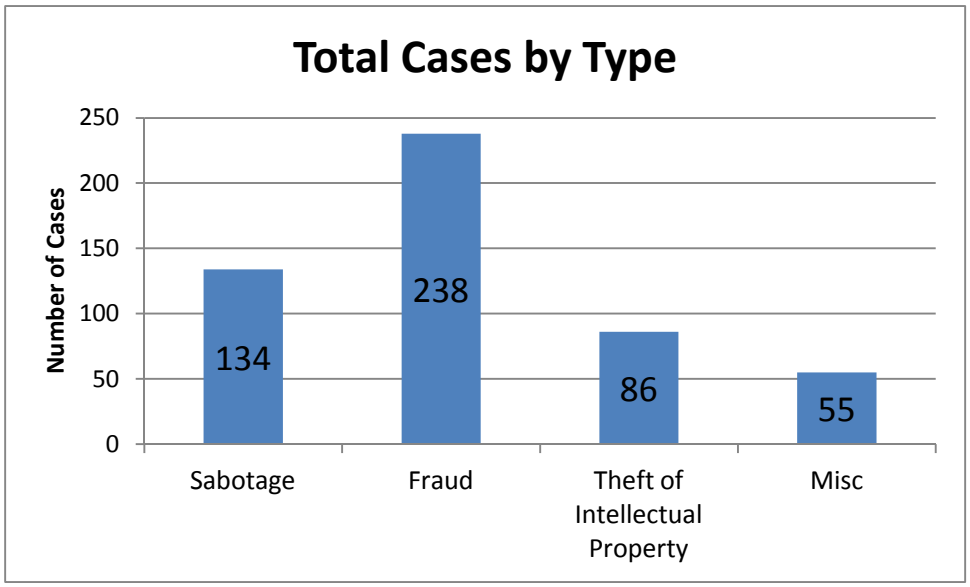

Figure 1: Number of Cases in the CERT Insider Threat Database by High-Level Category (Excluding National Security Espionage Cases)

We define a malicious insider as a current or former employee, contractor, or business partner who

- has or had authorized access to an organization's network, system, or data
- intentionally exceeded or misused that access
- negatively affected the confidentiality, integrity, or availability of the organization's information or information systems

This technical note focuses on a subset of the cases categorized as theft of IP. Insider theft of IP is defined as an insider's use of information technology to steal IP from the organization. This includes industrial espionage. In our database, each case entry contains general details about the case, including a timeline of the incident. The specific codebook items we focus on for this pattern, which were derived from our analysis of the database, are the primary method of data exfiltration (email), attack time (i.e., within 30 days before termination), and the intended recipients of the organization's stolen IP.

The purpose of this analysis is to develop a rule that could be applied to a log indexing application to help analysts detect malicious behavior.

2.1 A Note on Signature Application

Technical signatures developed by the CERT Insider Threat Center are generally designed to be applied toward a particular user or group of users. These signatures are not intended to be applied to all users across the enterprise because doing so will generate a large number of false positives.

Prior to applying this signature, the organization should facilitate proper communication and coordination between relevant departments across the enterprise, especially information technology, information security, human resources, physical security, and legal. This cooperation is necessary to ensure that any measures taken by the organization to combat insider threat comply with all organizational, local, and national laws and regulations. First the organization must identify users who warrant targeted monitoring via this signature (which in this case includes employees within 30 days before termination). The organization will then be able to determine the appropriate user names, account names, host names, and host addresses to enter into the signature to make the alert volume more meaningful and manageable.

3 Monitoring Considerations Surrounding Termination

3.1 Case Pool

In a recent report on insider theft of IP [Moore 2011], CERT Insider Threat Center staff noted a strong finding that many insiders who steal IP do so within 30 days prior to their termination. This finding and the findings from a focused study of technical concerns in cases of theft of IP [Hanley 2010] indicate that organizations should be especially concerned with exfiltration of data over the organization's network. The primary vehicle for data exfiltration over the network is corporate email systems or web-based personal email services. Considerations of this issue should include the following:

- monitoring for misuse of personal web email services. While this is no less a threat than other exfiltration methods, it is beyond the scope of this technical note. We expect to address this challenge in a future report.
- monitoring for email to the organization's competitors or the insider's personal account. Corporate email accounts running on enterprise-class servers have a wealth of auditing and logging functionality available for use in an investigation or, in this case, a query to detect suspicious behavior. For example, in the case of mail server transaction logs, if an organization enumerates (but does not blacklist) suspicious transactions, such as data transfers to competitors, those email transactions are recorded in a form that is easily consumed by a log indexing engine. If an organization uses these logs to find messages of large size (potentially indicating an attachment or large amount of text in the message body) that are sent to suspicious domains within the 30 days before an employee's departure, the organization would have the basic criteria for building a query rule.

From this discussion, we can move toward an implementation strategy for these conditions on a logging engine. Consider the following implementation outline:

> ***if*** *the mail is from the departing insider*
> ***and*** *the message was sent in the last 30 days*
> ***and*** *the recipient is not in the organization's domain*
> ***and*** *the total bytes summed by day are more than a specified threshold*
> ***then*** *send an alert to the security operator*

This solution first keys upon the population of departing insiders, and then it sets a time window of 30 days, representing the window prior to the insider's termination, in which to search for suspicious mail traffic. Because the 30-day window is the finding from behavioral modeling work that we find most interesting, this serves as the root of the query. Possible data sources that could be used to instantiate this attribute in a live query include an Active Directory or other Lightweight Directory Access Protocol (LDAP) directory service, partial human resources records that are consumable by an indexing engine, or other proxies for employee status such as physical badge access status. Human resources systems do not always provide security staff with a simple indicator that an employee is leaving the organization. Instead, suitable proxies (preset account expiration, date the account is disabled, and so on) can be used to bound the 30-day window for targeted monitoring. Assuming employers actively disable accounts at termination, the

appropriately generated alert can be queried for the associated event ID or known text (as in the example implementation in Section 4) to find all employees leaving the organization. Depending on the structure of the remainder of the query, particularly if the only initial result is a unique identifier (UID) string, some customization here to convert a UID string to a user's email address may be useful. The example in Section 4 simply concatenates the UID to the local domain name.

Once the query has identified all mail traffic from departing users in their 30-day window, the next search criterion identifies mail traffic that has left the local domain namespace of the organization. Another logical boundary may be necessary for large federations of disparate namespaces or a wide trust zone with other namespaces. This boundary identifies potential data exfiltration via email by identifying messages whose intended recipient resides in an untrusted zone or in a namespace the organization otherwise has no control over. Specific intelligence or threat information may allow for significant paring of this portion of the query, perhaps even down to very specific sets of unwanted recipient addresses by country code top-level domains (ccTLD), known bad domain names, or similar criteria.

Because not all mail servers indicate an attachment's presence uniformly, the query next narrows by byte count to indicate data exfiltration. Setting a reasonable per-day byte threshold, starting between 20 and 50 kilobytes, should allow the organization to detect when several attachments, or large volumes of text pasted into the bodies of email messages, leave the network on any given day. This variable provides an excellent point at which to squelch the query as a whole.

4 An Example Implementation Using Splunk

Organizations using Splunk for centralized log indexing and interrogation can configure it to raise an alert when it observes the behaviors discussed above. The following is a Splunk rule that organizations could adjust to their particular circumstances. The example implementation and discussion presented in this section are adapted from an early version of a forthcoming CERT report.[3] The rule uses a sample internal namespace to illustrate the implementation. We assume a generic internal namespace of corp.merit.lab, with two servers of interest: MAILHOST, an Exchange server, and DC, an Active Directory Domain Controller.

The characteristics of the attack include remote access to the organization's information systems outside normal working hours. With these characteristics, we developed the following signature:

```
Terms: 'host=MAILHOST
        [search host="DC.corp.merit.lab"
             Message="A user account was disabled. *"
             | eval Account_Name=mvindex(Account_Name, -1)
             | fields Account_Name
             | strcat Account_Name "@corp.merit.lab" sender_address
             | fields - Account_Name]
         total_bytes > 50000 AND recipient_address!="*corp.merit.lab"
         startdaysago=30
         | fields client_ip, sender_address, recipient_address,
         message_subject, total_bytes'
```

The following sections break this query into manageable segments to show how it tries to address all the items of concern from this pattern.

4.1 Mail from the Departing Insider: `'host=MAILHOST []`

This query is actually a nested query. In this demonstration, the outermost bracket refers to a mail server, MAILHOST, and looks for a set of information first pulled from DC, a domain controller in the sample domain. Because the log query tool seeks employees leaving within the 30-day activity window, the logical place to start looking for employee information is the local directory service.

If human resources systems do not provide security staff with a simple indicator of employee departure, suitable proxies (preset account expiration, date the account is disabled, and the like) can be used to bound the 30-day window for targeted monitoring. If the organization actively disables employee accounts at termination, the appropriately generated alert can be queried for the associated event ID or known text (as in this demonstration) to find all employees leaving the organization. The query then concatenates the account name associated with the disable event to a string that ends with the organization's DNS suffix (`@corp.merit.lab` in this demonstration) to form a string that represents the email sender's address. This ends the first component of the query and provides the potentially malicious insider's email address.

[3] Moore, A.; Hanley, M.; Mundie, D. A Pattern for Insider Threat Risk Mitigation (forthcoming). Software Engineering Institute, Carnegie Mellon University.

4.2 Total Bytes Summed by Day More Than Specified Threshold: `total_bytes > 50000`

Not all mail servers provide a readily accessible attribute indicating that an email message included an attachment. Thus, the mail server is configured to filter first for all messages `of size`, which might indicate an attachment or a large volume of text in the message body. This part of the query can be tuned as needed; 50,000 bytes is a somewhat arbitrary starting value.

4.3 Recipient Not in Organization's Domain: `recipient_address!="*corp.merit.lab"`

This portion of the query instructs Splunk to find only transactions in which the email was sent to a recipient not in the organization's namespace. This is a vague query term that could generate many unwanted results, but it does provide an example of filtering based on destination. Clearly, not all messages leaving the domain are malicious, and an organization can filter based on more specific criteria such as specific country codes, known bad domain names, and so on.

4.4 Message Sent in the Last 30 Days: `startdaysago=30`

This sets the query time frame to 30 days prior to the date of the account disable alert message. This can be adjusted as needed, though the data on insider theft of IP exhibits the 30-day pattern discussed previously.

4.5 Final Section: `fields client_ip, sender_address, recipient_address, message_subject, total_bytes'`

The final section of the query creates a table with relevant information for a security operator's review. The operator receives a comma-separated values (CSV) file showing the sender, recipient, message subject line, total byte count, and client IP address that sent the message. This information, along with a finite number of messages that match these criteria, should provide sufficient information for further investigation.

5 Advanced Targeting and Automation

Originally, this control required manual identification of users of interest to populate the query with targets. In fact, we find that there are ways to go a step further and use simple tools to identify all users who have accounts set to expire within a 30-day window, and possibly feed this information directly into Splunk via a command line tool.

This method first requires an assumption that the organization sets the accounts of most insiders who have submitted advance notice of their resignation to automatically expire on the insider's last day of employment. Clearly, this will not be universally true. But again, it provides a possible data point where available.

In Microsoft Active Directory, we can quickly identify the users who have expiring accounts in the next 30 days by using the PowerShell AD administration tools. For example, a simple one-line query can extract all expiring user accounts relatively quickly. Running the example query below requires the PowerShell AD modules and, depending on privilege delegation in the environment, may require a privileged user in the directory to run the command.

```
PS C:\Users\ffishbeck_sec> Search-ADAccount -AccountExpiring -TimeSpan
30.00:00:00


AccountExpirationDate : 7/9/2011 12:00:00 AM
DistinguishedName     : CN=Brian
Smith,OU=Employees,DC=corp,DC=merit,DC=lab
Enabled               : True
LastLogonDate         : 7/1/2011 18:40:03 AM
LockedOut             : False
Name                  : Brian Smith
ObjectClass           : user
ObjectGUID            : a6ed88a4-fab3-494d-9f45-4d9ad11e1069
PasswordExpired       : True
PasswordNeverExpires  : False
SamAccountName        : Brian Smith
SID                   : S-1-5-21-2581603451-735610124-1584908375-1108
UserPrincipalName     : bsmith@corp.merit.lab

AccountExpirationDate : 7/23/2011 12:00:00 AM
DistinguishedName     : CN=Jennifer
Burns,OU=Employees,DC=corp,DC=merit,DC=lab
Enabled               : True
LastLogonDate         : 6/29/2011 12:18:00 PM
LockedOut             : False
Name                  : Jennifer Burns
ObjectClass           : user
ObjectGUID            : fdd0b06f-c929-4da9-9f89-4c9415e3d756
PasswordExpired       : True
PasswordNeverExpires  : False
SamAccountName        : Jennifer Burns
SID                   : S-1-5-21-2581603451-735610124-1584908375-1110
UserPrincipalName     : jburns@corp.merit.lab

AccountExpirationDate : 7/2/2011 12:00:00 AM
DistinguishedName     : CN=Megan
Jordan,OU=Employees,DC=corp,DC=merit,DC=lab
Enabled               : True
LastLogonDate         : 6/30/2011 4:30:28 AM
LockedOut             : False
Name                  : Megan Jordan
ObjectClass           : user
ObjectGUID            : 4f11a5f4-7e49-4ec7-a34b-882fb643e5a3
PasswordExpired       : True
PasswordNeverExpires  : False
SamAccountName        : Megan Jordan
SID                   : S-1-5-21-2581603451-735610124-1584908375-1117
UserPrincipalName     : mjordan@corp.merit.lab
```

Once we know which user accounts are expiring in the near future, we can either manually populate the Splunk query with these LDAP user names, or we can experiment with piping them into a command line Splunk query. There are open source projects, including splunk-powershell, that would support this type of activity with a very simple script.[4] While splunk-powershell does not appear to work with the newest release of PowerShell 2, it does work with the original PowerShell binaries and will successfully query a current v4.1.x Splunk installation.

[4] http://code.google.com/p/splunk-powershell/

6 Conclusion

Organizations must carefully consider employee communications during the time frame immediately preceding termination. Many insiders have stolen information within the 30 days prior to departure. Many of these thefts occurred via corporate email servers. A well-constructed rule set could be placed on a centralized logging application to identify suspicious mail traffic originating from soon-to-be-departing employees. These well-crafted rules, based on trends observed from actual cases, should reduce analyst workload by presenting them with behaviors that are potentially malicious and worthy of further investigation.

Bibliography

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

[Hanley 2010]
Hanley, M. "Candidate Technical Controls and Indicators of Insider Attack from Socio-Technical Models and Data." *Proceedings of the 2010 NSA CAE Workshop on Insider Threat*, St. Louis, MO, November 2010.

[Hanley 2011]
Hanley, M.; Dean, T.; Schroeder, W.; Houy, M.; Trzeciak, R. F.; & Montelibano, J. *An Analysis of Technical Observations in Insider Theft of Intellectual Property Cases* (CMU/SEI-2011-TN-006). Software Engineering Institute, Carnegie Mellon University, 2011. http://www.sei.cmu.edu/library/abstracts/reports/11tn006.cfm

[Moore 2011]
Moore, A.; Cappelli, D.; Caron, T.; Shaw, E.; Spooner, D.; & Trzeciak, R. *A Preliminary Model of Insider Theft of Intellectual Property* (CMU/SEI-2011-TN-013). Software Engineering Institute, Carnegie Mellon University, 2011. http://www.sei.cmu.edu/library/abstracts/reports/11tn013.cfm

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	October 2011	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Insider Threat Control: Using Centralized Logging to Detect Data Exfiltration Near Insider Termination	FA8721-05-C-0003

6. AUTHOR(S)

Michael Hanley, Joji Montelibano

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2011-TN-024

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Since 2001, the CERT® Insider Threat Center has built an extensive library and comprehensive database containing more than 600 cases of crimes committed against organizations by insiders. A significant class of insider crimes, insider theft of intellectual property, involves highly damaging attacks against organizations that result in significant tangible losses in the form of stolen business plans, customer lists, and other proprietary information. The Insider Threat Center's behavioral modeling of insiders who steal intellectual property shows that many insiders who stole their organization's intellectual property stole at least some of it within 30 days of their termination. This technical note presents an example of an insider threat pattern based on this insight. It then presents an example implementation of this pattern on an enterprise-class system using the centralized log storage and indexing engine Splunk to detect malicious insider behavior on a network.

14. SUBJECT TERMS	15. NUMBER OF PAGES
insider threat, information security, system dynamics, behavioral modeling, security controls, security metrics	23

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL