



Software Engineering Institute

Carnegie Mellon University

Spotlight On: Programmers as Malicious Insiders—Updated and Revised

Matthew Collins
Dawn M. Cappelli
Tom Caron
Randall F. Trzeciak
Andrew P. Moore

December 2013

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE DEC 2013		2. REPORT TYPE		3. DATES COVERED 00-00-2013 to 00-00-2013	
4. TITLE AND SUBTITLE Spotlight On: Programmers as Malicious Insiders-Updated and Revised				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Copyright 2013 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense 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.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon[®] and CERT[®] are registered marks of Carnegie Mellon University.

DM-0000612

Table of Contents

Acknowledgements	iii
A Snapshot of Malicious Insiders Who Used Programming Techniques	1
Introduction	1
Definitions	1
Who They Are (Updated).....	2
What They Strike (Updated)	3
When and Where They Strike (Updated)	5
Why They Strike	6
How They Strike (Updated)	7
How They Were Detected (New).....	8
Case Examples (Updated)	10
Case 1: Terminated insider remotely deletes source code	10
Case 2: Student steals personally identifiable information to commit fraud.....	10
Case 3: Insiders steal confidential press releases and make stock trades.....	11
Case 4: Three insiders work together to commit sabotage	11
Case 5: Insider steals source code to create competing organization.....	11
Case 6: Former employee alters customer database	12
Case 7: Former system administrator and member of the internet underground attacks victim organization.....	12
Case 8: Insider sets logic bomb after decreased bonus	13
Case 9: Insider and outsider conspire to plant virus	13
Case 10: Disgruntled insider deploys logic bomb to take down a manufacturing plant..	14
Conclusion	15
About the Insider Threat Team	16

List of Figures

Figure 1: Insider Employment Type	2
Figure 2: Insider Employment Status	2
Figure 3: Age of Insider at Time of Attack.....	3
Figure 4: Type of System Compromise.....	3
Figure 5: Impact of Insider Attack to Organization in 33 Cases Where Impact Was Known..	4
Figure 6: Programming Attacks by Industry Sector.....	4
Figure 7: Time of Attack (When Known)	5
Figure 8: Location of Attack (When Known).....	5
Figure 9: Type of Crime.....	6
Figure 10: Insider's Motives.....	6
Figure 11: Technical Methods Used by Insiders	7
Figure 12: Programming Attacks by User Account	8
Figure 13: Method Used to Detect Insider.....	8
Figure 14: Role of Logs in Insider Detection	9

Acknowledgements

Special thanks to Paul Ruggiero and all of the Software Engineering Institute's CERT® Division.

A Snapshot of Malicious Insiders Who Used Programming Techniques

Introduction

This white paper updates the 2008 article “Spotlight On: Programming Techniques Used as an Insider Attack Tool.” The white paper begins with a discussion of the who, what, when, where, and how of insider attacks and covers case examples of malicious insiders who attacked using programming techniques. This paper highlights technical malicious insiders who use their skills to create scripts or programs that harm their organizations. The insiders in these attacks were able to modify source code, set logic bombs to destroy data, and write programs to capture user credentials.

Insiders who use programming techniques to attack most often commit sabotage and fraud. Their motives are most commonly revenge and financial gain. The insiders in these cases most commonly use their own information technology (IT) account and have authorized access to the source code or systems that they attack. The insiders described in this paper span all age ranges, work in all industry sectors, and attack both while on-site and from remote locations. Though these insiders were highly technical, all of the attacks in this paper could have been detected earlier or prevented by following the recommendations in the CERT® Insider Threat Center’s *Common Sense Guide to Mitigating Insider Threats, 4th Edition*.¹

Definitions

Below are some useful definitions that show how we define a malicious insider threat as well as the types of insider crime: sabotage, fraud, and theft of intellectual property.

The CERT Insider Threat Center, part of Carnegie Mellon University’s Software Engineering Institute, defines a malicious insider threat as a current or former employee, contractor, or business partner who has or had authorized access to an organization’s network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization’s information or information systems.

We define insider IT sabotage crimes as those in which an insider uses IT to direct specific harm at an organization or an individual. Insider fraud cases are those in which an insider uses IT for the unauthorized modification, addition, or deletion of an organization’s data for personal gain or

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

¹ Silowash, George; Cappelli, Dawn; Moore, Andrew; Trzeciak, Randall; Shimeall, Timothy; & Flynn, Lori. *Common Sense Guide to Mitigating Insider Threats, 4th Edition* (CMU/SEI-2012-TR-012). Software Engineering Institute, Carnegie Mellon University, 2012.
<http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=34017>

uses IT to steal information that leads to identity crime. Insider theft of intellectual property (IP) is an insider's use of IT to steal IP from the organization.

Who They Are (Updated)

Since the original "Spotlight On: Programming Techniques Used as an Insider Attack Tool" was published in 2008, nearly 500 cases have been added to the CERT insider threat database. Of the more than 700 total cases, 49 involved insiders using programming tactics. The majority of the insiders in these cases were full-time, current employees of the victim organization at the time of attack (Figure 1 and Figure 2). The ages of the insiders ranged from younger than 20 years old to older than 50 years (Figure 3).

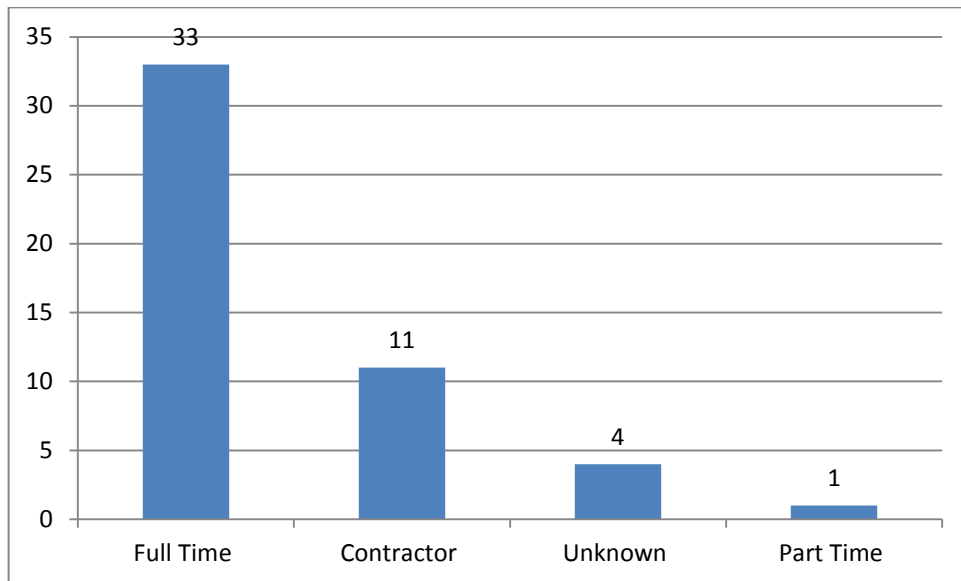


Figure 1: Insider Employment Type

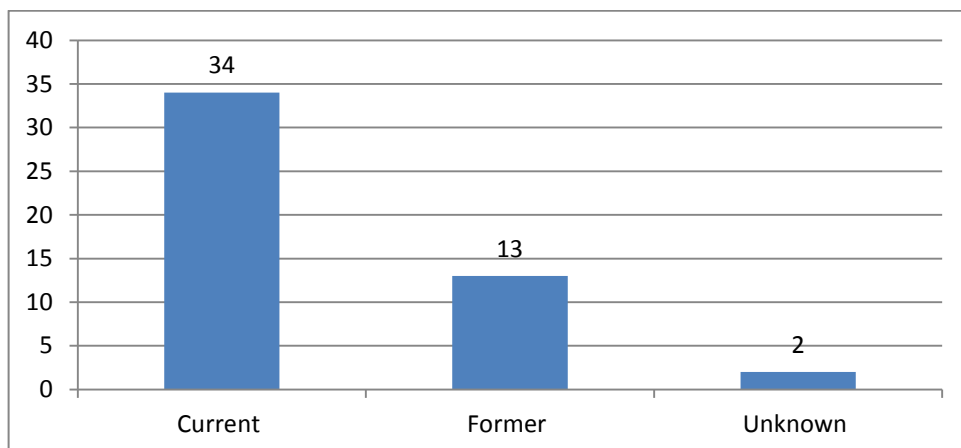


Figure 2: Insider Employment Status

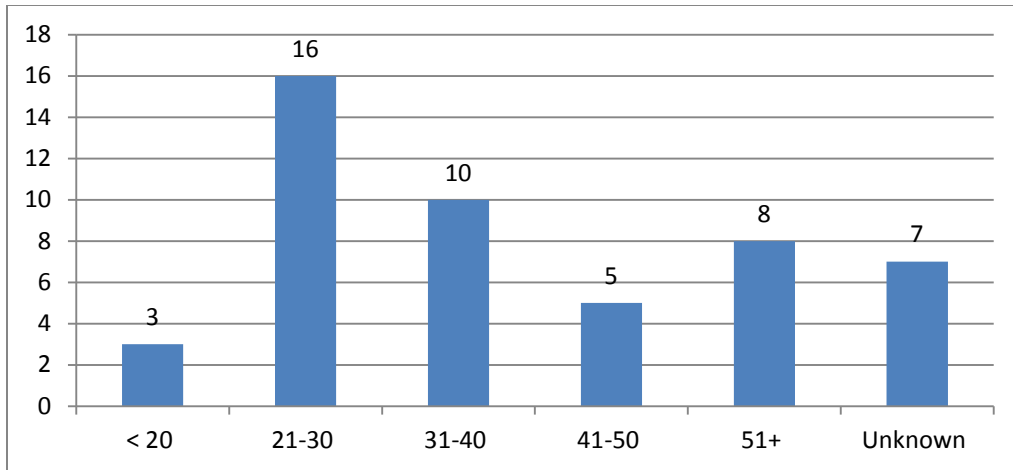


Figure 3: Age of Insider at Time of Attack

What They Strike (Updated)

Insiders used programming tactics to modify or compromise the integrity, availability, and confidentiality (Figure 4) of systems in nearly all industries. The financial impact of programming attacks was wide ranging, but in some cases they cost the victim organization more than \$1 million (Figure 5).

The organizations most often victimized by insiders using programming methods were related to banking and finance as well as information technology (Figure 6). Cases involving insiders who used programming methods also occurred in at least 11 additional industry sectors.

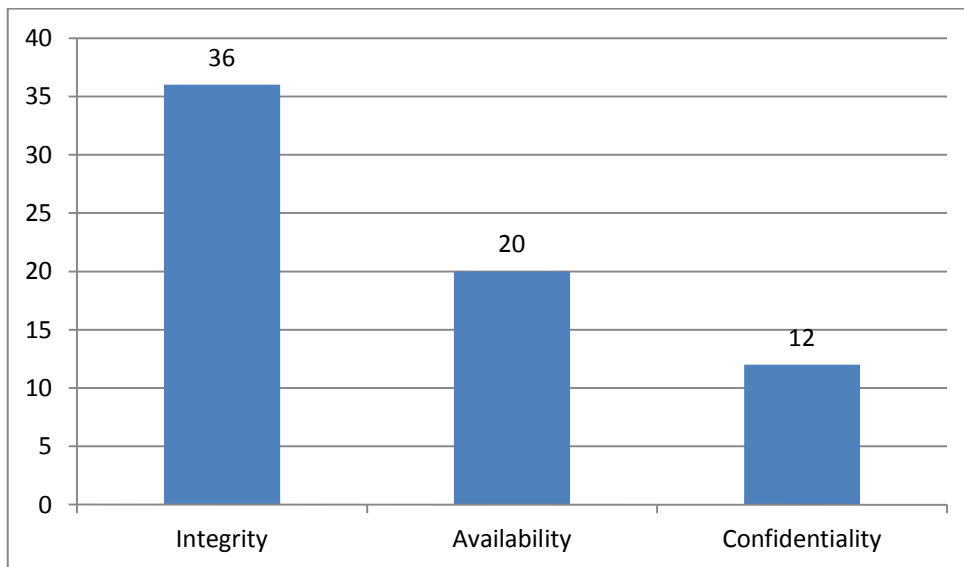


Figure 4: Type of System Compromise

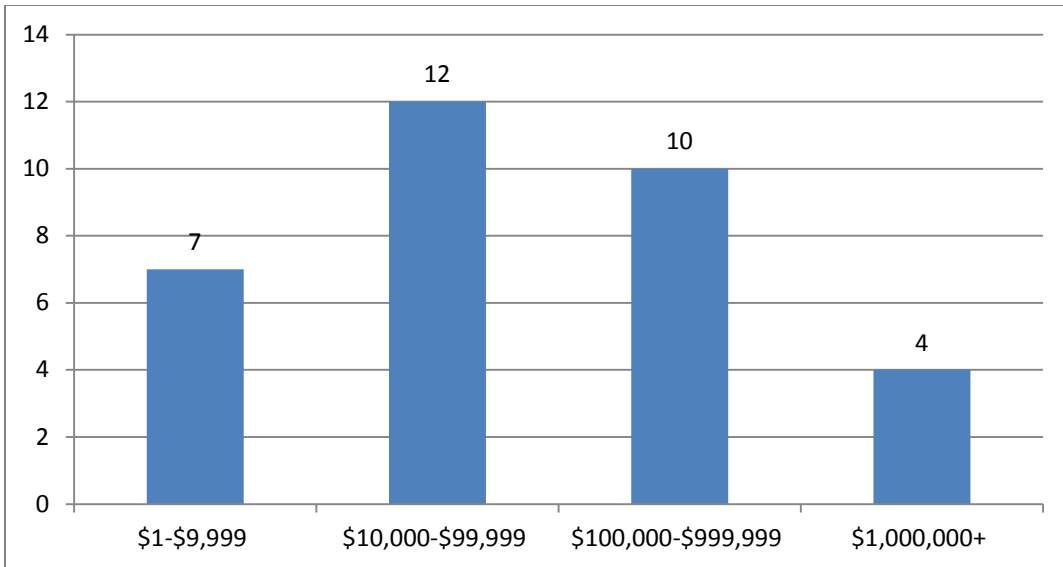


Figure 5: Impact of Insider Attack to Organization in 33 Cases Where Impact Was Known

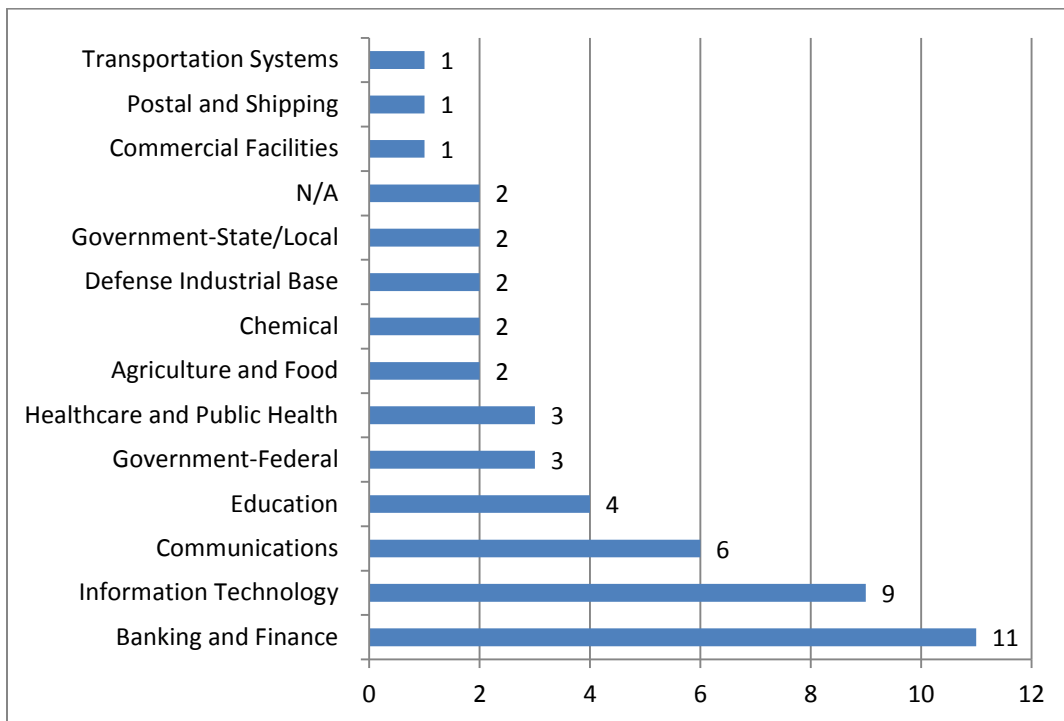


Figure 6: Programming Attacks by Industry Sector

When and Where They Strike (Updated)

In the 30 cases in which the time of attack was known, more than half of the insiders attacked during work hours (Figure 7). The location of attack was known in 38 of the 49 cases: on-site attacks slightly outweighed remote attacks, and just one insider carried out the attack both on-site and remotely (Figure 8). In four of the cases, the time of attack relative to the insider taking a new position outside the victim organization was known: three of the insiders attacked after taking a new position, and one insider attacked the organization both before and after taking a new position.

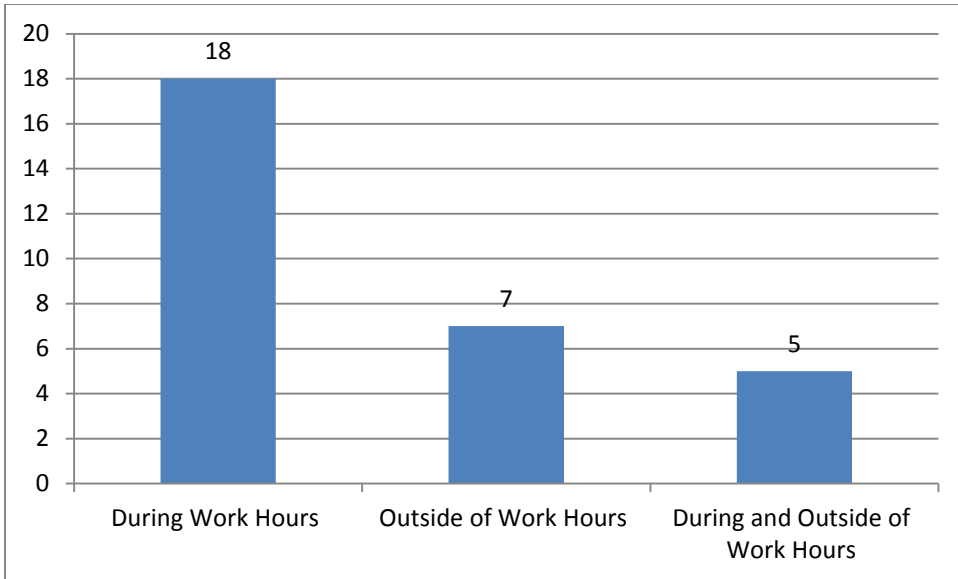


Figure 7: Time of Attack (When Known)

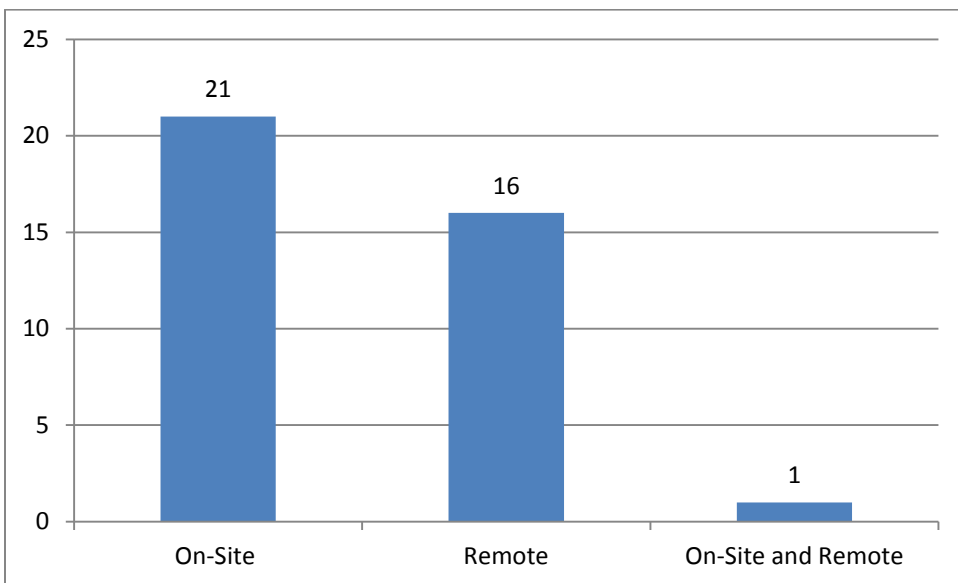


Figure 8: Location of Attack (When Known)

Why They Strike

The insiders who attacked using programming techniques most often committed sabotage or fraud. In some cases, the insider felt personal ownership over the victim organization's intellectual property and stole it upon leaving the organization (Figure 9). Malicious insiders who used programming techniques to attack did so most commonly for revenge or financial gain (Figure 10).

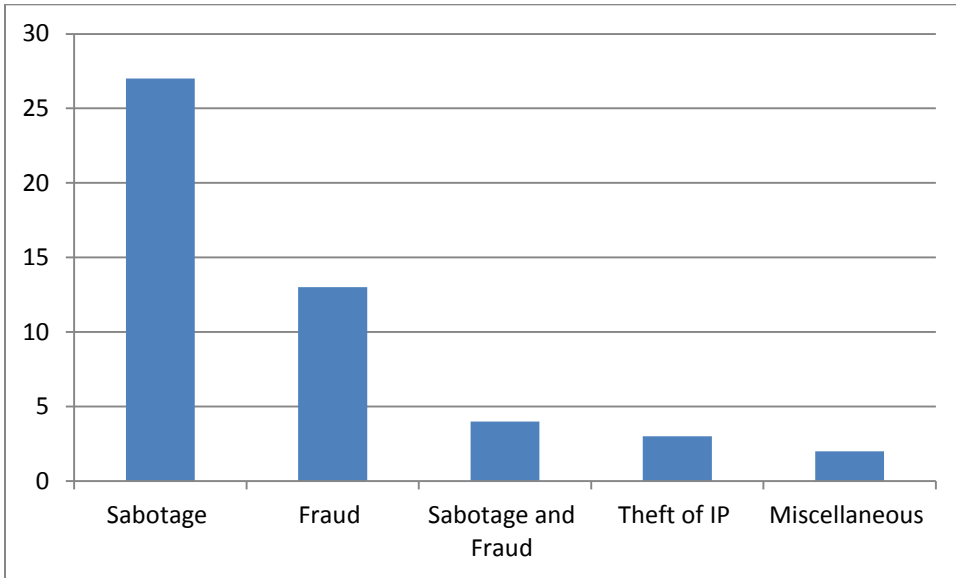


Figure 9: Type of Crime

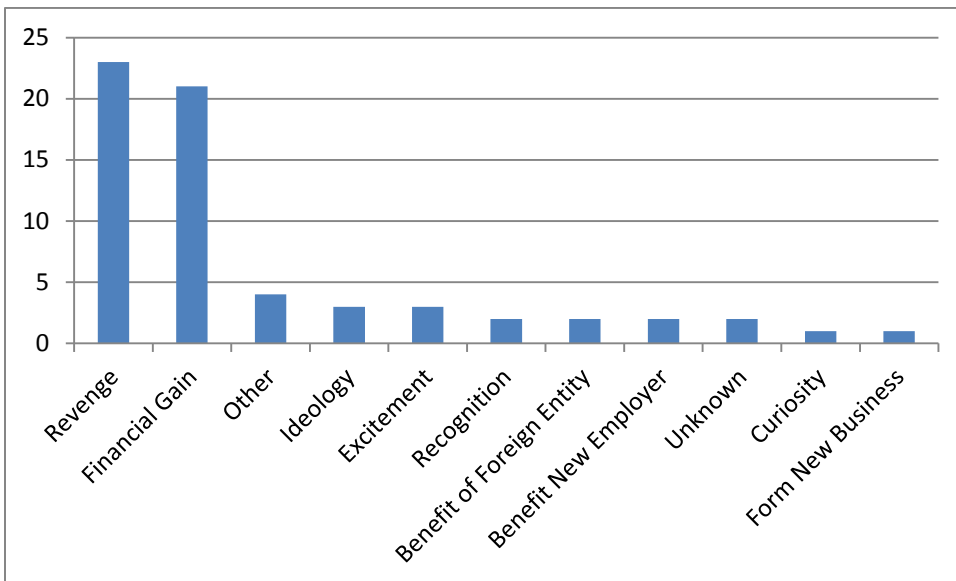


Figure 10: Insider's Motives

How They Strike (Updated)

Insiders used some type of programming method to carry out their attacks in all of the cases considered in this article. Programming methods include techniques such as modifying source code, creating a logic bomb, writing malicious code, using a hack tool, deploying a virus or worm, and using keystroke loggers. The most popular attack method was authorized use (Figure 11), in which attackers used the access granted to them by the organization to access the organization's systems and to carry out their attacks. Of the cases involving use of a system account, insiders most often used their own accounts when carrying out a programming attack (see Figure 12).

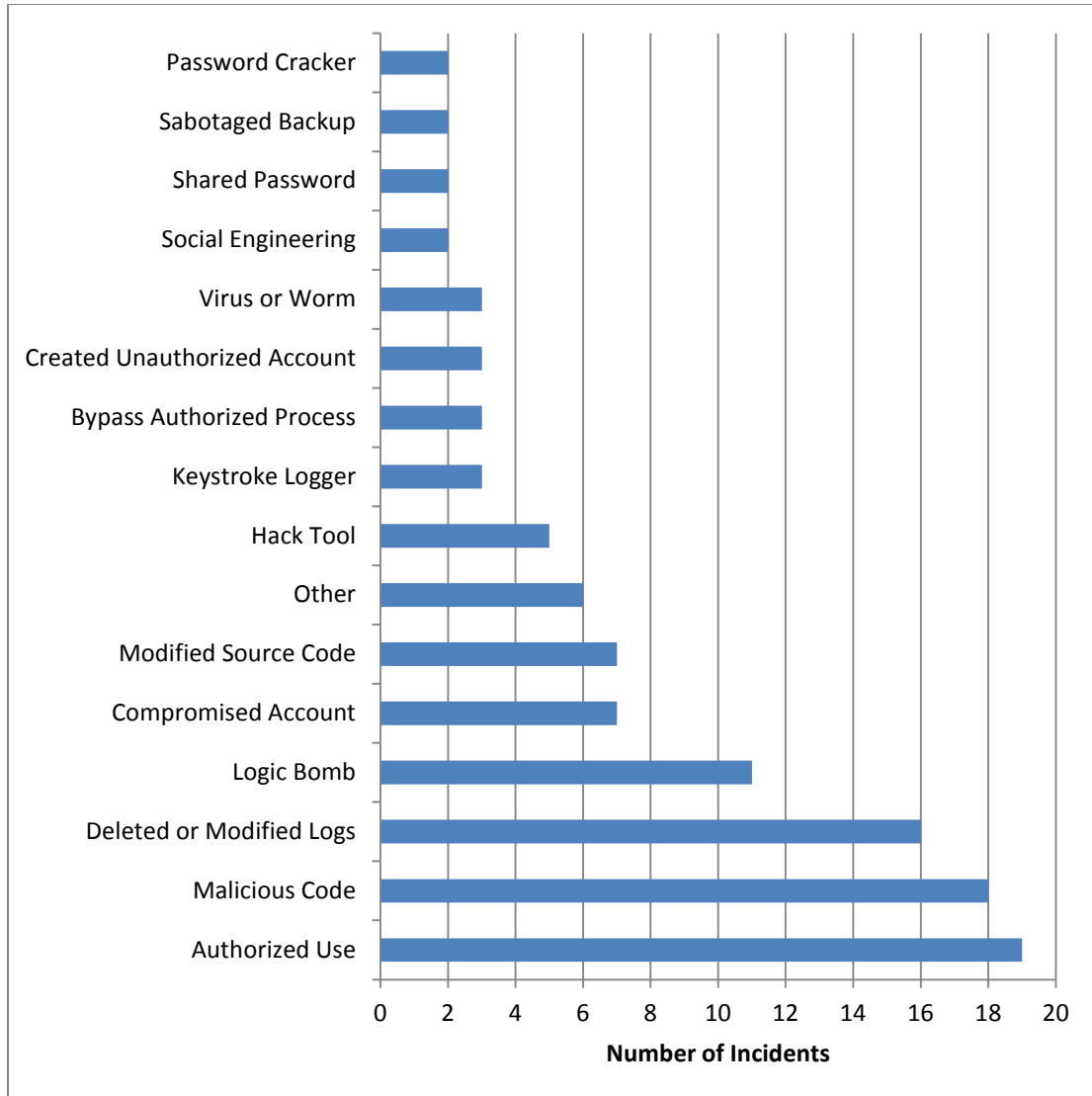


Figure 11: Technical Methods Used by Insiders

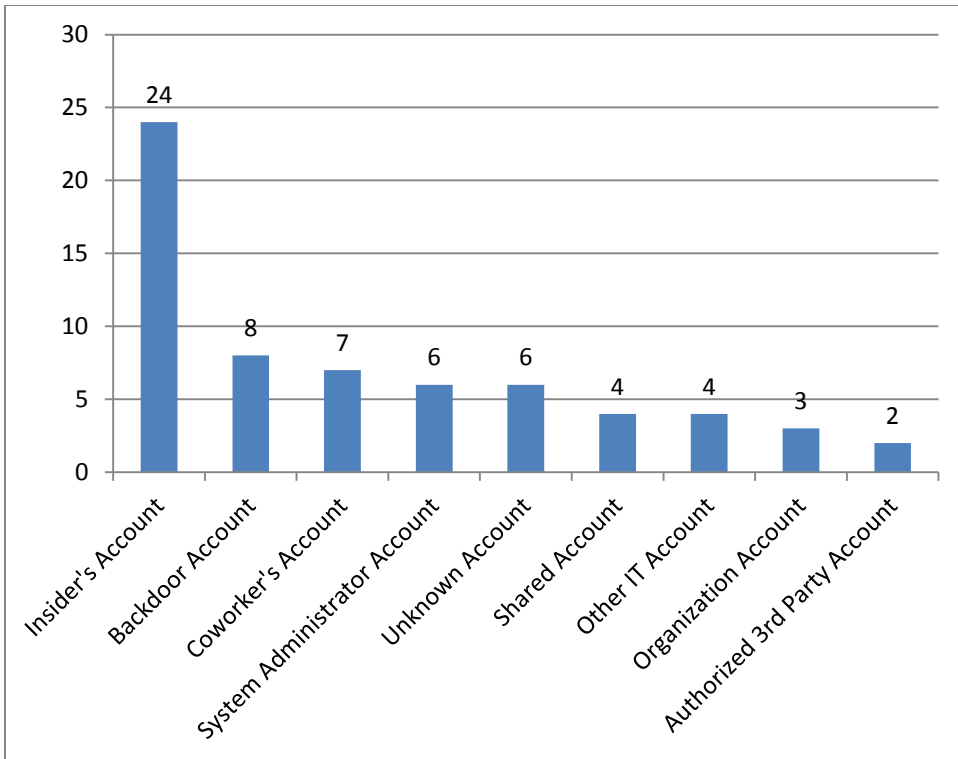


Figure 12: Programming Attacks by User Account

How They Were Detected (New)

Insiders were most often detected and reported by other employees. IT staff reported a large number of cases. System failure accounted for 20% of detection in the insider threat cases. Logs were valuable and were used in a large majority of the cases to determine the extent of the insider's attack. In some insider cases, multiple methods were used for detection.

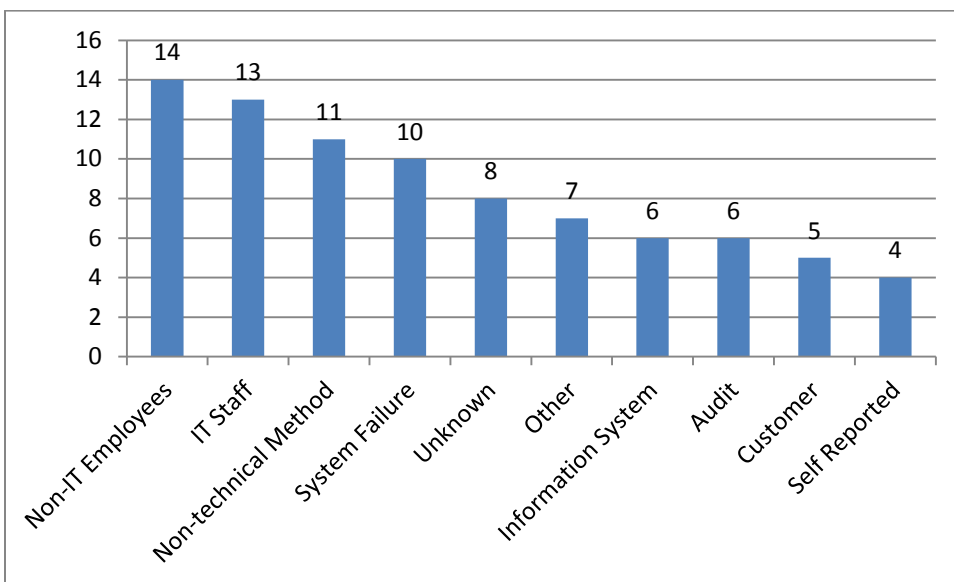


Figure 13: Method Used to Detect Insider

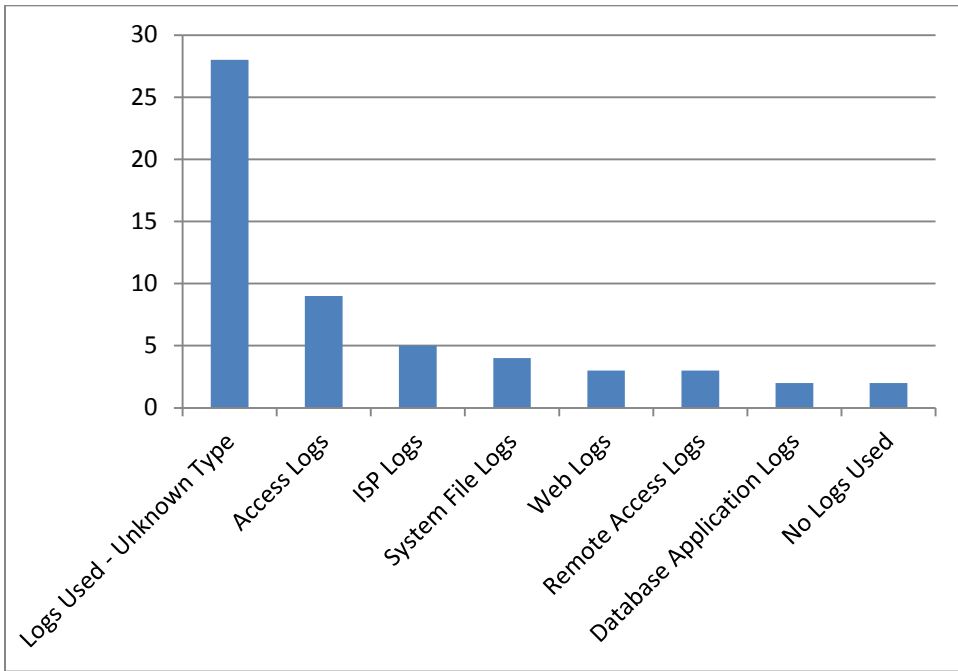


Figure 14: Role of Logs in Insider Detection

Case Examples (Updated)

Detailed descriptions of new cases involving programming are provided below to illustrate the complexity of such attacks, the contextual factors surrounding the incidents, and the steps taken by insiders to carry out their attacks.

As in all of the CERT Division's insider threat research, the names of both the insiders and victims have been omitted.

Case 1: Terminated insider remotely deletes source code

The insider was originally employed as an e-commerce software developer for the victim organization, a producer of manufacturing equipment for computer chips. The insider chose to move to a new state, and the organization agreed to keep the insider employed. For legal reasons, the insider's employment status had to change from full-time employee to contractor consultant. As a consultant, the insider was given remote access to the organization's servers during normal working hours.

Over time, the insider's relationship with the victim organization deteriorated because the insider felt the reduced benefits received as a contractor were inadequate compared to the benefits previously received as a full-time employee. The victim organization then gave the insider one month's notice of termination. Upon receiving notice, the insider remotely logged into the server during working hours and deleted multiple pieces of software in development. The insider then changed the root password, modified system logs, and reported problems logging into the server. The insider resigned at the end of the day.

The insider was detected when the organization noticed the lost data. Forensic audits revealed that the server had been accessed by the insider's internet service provider's (ISP's) domain. The victim organization spent more than \$20,000 to recover the lost data. The insider was arrested, convicted, ordered to pay restitution, and sentenced to three years of probation.

Case 2: Student steals personally identifiable information to commit fraud

The insider was a student and computer science major at the victim organization, a college. The insider installed keystroke-monitoring software on more than 100 of the organization's computers and collected more than 4,000 records of students' personally identifiable information (PII). The PII included credit card numbers, Social Security numbers, and students' passwords for computer and building access.

Using the stolen PII, the insider re-coded his student ID card, which was also a debit card, to gain access to campus buildings and to make fraudulent purchases. The incident took place over six months, until the organization detected the illicit use of ID cards at the bookstore where the insider had made purchases with others' accounts.

Ultimately, the insider was accused of interception of wire communications, unauthorized access to computer systems, larceny, and identity fraud. The insider was found guilty, ordered to pay

restitution, and sentenced to five years of electronically monitored probation. The insider was also required to receive counseling.

Case 3: Insiders steal confidential press releases and make stock trades

The insiders were employed as contractors for part of the investment services team of a foreign financial institution. The primary insider was a day trader with a computer programming background. The secondary insider was the head of the trading department. The insiders' employer was a trusted business partner to the victim organization, a commercial news distribution service. One of the victim organization's services was to proofread and release news from other organizations. The two insiders created a spider program to access and record the confidential information contained in the press releases. The insiders then made trades on this information that had not yet been released to the public.

The insiders obtained more than 350 confidential press releases from more than 200 organizations and made in excess of \$7 million in related trades. The insiders were detected when the victim organization's technical team noticed unusual trading the day before a merger announcement. The insiders were arrested, convicted, and ordered to pay financial penalties. The insiders' employer was also required to pay more than \$500,000 in fines.

Case 4: Three insiders work together to commit sabotage

Three insiders were employed by the victim organization, a provider of data related to insurance. The insiders were responsible for the operation and functionality of the victim organization's computer systems, networks, and programs. For more than eight months, the insiders sabotaged the organization's computer network. The insiders acted both on-site and remotely. Some of the acts of sabotage included deleting critical information and restricting access to machines. The insiders programmed the organization's computers to erase evidence of their attacks.

The victim organization discovered the attack when the insiders demanded more than \$150,000 each to settle a discrimination claim. The three insiders were arrested, and two were convicted of the crimes. The cost to the victim organization is estimated at more than \$600,000.

Case 5: Insider steals source code to create competing organization

The insider was employed as a programmer and product support engineer by a small networking organization. The company was bought out by a competitor, the victim organization. The insider had worked at the original organization for only 10 days prior to the buyout and received \$100,000 worth of stock options. The insider signed a proprietary inventions agreement with the victim organization at the time of the buyout. After the buyout, the insider recruited two outsiders to help him develop a product to compete with the one he was developing for the victim organization. The insider provided the outsiders with the design and source code for the victim organization's product. The insider created a competitor company, the beneficiary organization. On the insider's last day of work, nine months after the buyout, the insider used his work computer to create back-up tape containing the victim organization's source code and copied it to his home computer. The insider used the source code to develop a competitor product. Eleven days later, the insider posted a message to a Usenet group, which indicated he had a DOS version

of his competitor product running and asked for help programming the code on an embedded chip. The two outsiders dropped out of the scheme. The victim organization discovered the insider's competitor product at a trade show and accused the insider of stealing its source code. The insider claimed that his competitor product was different from the victim organization's original product and disputed the victim organization's ownership of the intellectual property. The insider was arrested after authorities discovered copies of the victim organization's source code and software on his personal computer. The insider was charged and found guilty.

Case 6: Former employee alters customer database

The insider was formerly employed by the victim organization, a court document subscription service. The insider became disgruntled when a new CEO was hired and refused to honor a verbal agreement between the former CEO and the insider regarding compensation and vacation time. The insider resigned and took a series of malicious actions intended to deny customers access to the database unless they called the organization's help desk. The insider was able to bypass system front-ends to obtain unauthenticated access to a customer database. The insider remotely accessed the database, outside of work hours, and made malicious changes to customer information, including changing user names by a single character and changing customers' access once they logged in. The insider made complex queries intended to reduce system performance for all logged-in customers. The insider also updated the source code of web pages by making small changes to the database queries, including commenting out code or changing the query to use a slower method.

Though the changes were relatively minor and did not cause a large financial impact to the organization, the organization had to handle multiple customer complaints and had to troubleshoot each problem individually. The insider was detected when the organization recognized that an apparent intruder had changed some website-related files. To identify the insider, the organization created duplicates of its servers and routed calls from the attacker into the duplicate servers. The organization worked with the ISP, who managed the source IP of the attacks, to tie them back to the insider's home computer. The incident took place over a week. The insider was arrested, convicted, and sentenced to two years of unsupervised probation.

Case 7: Former system administrator and member of the internet underground attacks victim organization

The insider was originally employed as a system administrator by the victim organization, a telecommunications company. The insider resigned without providing any advance notice to the organization. The insider refused to provide the system administrator passwords to the organization until he received payment for his last two days of work. The insider then used remote access, during working hours, to attack the organization's network for a month. The insider remotely accessed the organization's key files and email. The insider also modified systems to prevent the organization from performing administrative functions.

The insider remotely accessed the Domain Name System (DNS) server and changed the name resolution settings to point to a malicious DNS name. The next day, the organization finally received passwords from the insider and promptly changed them for all administrative functions. The organization contacted law enforcement for assistance. The insider executed several

additional attacks, including running a sniffer on the network for several hours, running port scans from the organization's systems, and downloading internal files to his home computer. The insider also used the organization's systems to scan government systems.

While on the company network, the insider, who was associated with the internet underground, chatted with other hackers, bragged about the damage he could inflict on the organization, and claimed that he installed a password on the organization's hardware that prevented others from editing system settings. The insider had a history of psychological and psychiatric problems. The insider also had an extensive criminal history, including burglary, theft, credit card fraud, and weapons violations. A search of the insider's home revealed bomb-making materials, terrorist manuals, and child pornography stored on his home computer. The insider was arrested, convicted, ordered to pay a \$3,000 fine, and sentenced to two years of supervised probation. Three years later, the insider committed another act of insider sabotage against a different former employer.

Case 8: Insider sets logic bomb after decreased bonus

The insider was employed as a systems administrator by the victim organization, a financial services firm. The insider became disgruntled when the organization announced to employees that bonuses would be half of the normal amount. The insider complained about the lower bonus to his supervisor. The insider built and distributed a logic bomb on the organization's Unix-based network, which took down nearly 2,000 servers in the head office and 370 servers at branch offices around the country. Prior to the logic bomb's detonation, the insider purchased put options on the company, expecting the subsequent detonation of the logic bomb to drive down the firm's stock price. The insider quit when the organization suspected the insider had committed malicious activity. Although the stock price did not drop, the logic bomb cost the victim organization \$3.1 million in repairs and recovery time. A forensics investigation connected the insider to the incident by examining VPN connections, accesses, and code snippets sent between his home computer and the organization's network. The insider was arrested, convicted, and sentenced to 97 months of imprisonment.

Case 9: Insider and outsider conspire to plant virus

The insider was employed as a technical manager by the victim organization, a computer manufacturer. The insider was working on a new product line and actively sabotaged portions of the project to skew performance results. For five months, the insider sabotaged tests on the project's server by reformatting disks, cutting cables, sending reset commands to the project's server, and falsifying logs. Management at the organization suspected that the project had been sabotaged when, despite multiple changes, they were unable to stabilize the product. The organization moved product testing to a new facility and restricted access to a smaller group of individuals, which did not include the insider. During the controlled testing, the project appeared to have no problems.

When the insider was granted remote access to the system, the problems returned to the project. Monitoring revealed a link between the insider and the sabotage efforts. The insider was caught stealing seven years' worth of a colleague's email records and was sent home. The insider remotely accessed the organization's network, connected to computers to which he had no access

privileges, transferred confidential information outside of the company, and attempted to destroy logs of his actions. The organization spent over \$1 million to fix the damages, but its lost profits and reputation damages were estimated to be more than \$80 million. The organization filed a civil suit against the insider, and the suit was settled for \$200,000.

Case 10: Disgruntled insider deploys logic bomb to take down a manufacturing plant

The insider was employed as a network administrator by the victim organization, a manufacturer of measurement and control devices. The insider became disgruntled when the organization went through a major expansion. Prior to the expansion the insider had been promoted to management and subsequently demoted after being reprimanded for inappropriate behavior. The insider behaved aggressively and abusively toward his coworkers by purposely bumping into people and downplaying their achievements, bragging about his own abilities, taking credit for others' work, bottlenecking projects, and loading faulty programs to make others look bad. The insider also stole the organization's equipment for personal use and ran a side business.

Prior to his termination, the insider interviewed with competing organizations. The insider then systematically centralized the critical manufacturing programs for one of the organization's manufacturing plants to prepare for the release of a logic bomb. The insider tested the logic bomb on the system three times and set the logic bomb to detonate three weeks after the insider's termination. The logic bomb, designed to execute at first login, used an unauthorized account to delete many crucial programs that the plant relied on for its manufacturing process. Although the malicious software was never found, reformatted backup tapes and malicious programs were found in the insider's possession. The organization's damages were estimated at more than \$10 million. The insider was arrested, convicted, and sentenced to more than three years of imprisonment.

Conclusion

Technical employees are not the only insiders who can use programming methods to attack their organizations; employees with any job role within the organization can do so. Programming methods are most often used to commit sabotage, and the attacks occur in all industry sectors. Revenge and financial gain are insiders' most common motives. Programming attacks are carried out both during and outside of normal working hours and occur both on-site and from remote locations. In our sample, the majority of insiders were full-time employees. The majority of the insiders were also currently working for the organization. The insiders ranged in age from younger than 20 years to older than 50.

With the wide range of ages, job roles, attack locations, and attack times, there is no single, clear picture of a malicious insider who uses programming methods to harm a victim organization. The majority of these insiders, however, did have similar motives and goals for their attacks. In all cases, the victim organizations were impacted by the material and immaterial costs of the attack. At least four of the cases caused losses to the victim organization that exceeded \$1 million, and one attack led the victim organization to declare bankruptcy. All of the organizations had to deal with the negative publicity that comes with an insider case and the time and resources to prosecute the insider.

While many of these attacks were successful, in some cases the insider attack was detected and mitigated by the organization before the insider was able to cause a significant financial impact. Although there is no one profile that can be used to detect malicious insiders who use programming methods in their attacks, countermeasures can be implemented that would stop many of these attacks. The CERT Insider Threat Center has studied these and similar cases to develop the countermeasures presented in the *Common Sense Guide to Mitigating Insider Threats*, 4th Edition. Following these mitigation strategies would have prevented many, if not all, of the attacks covered in this article.

About the Insider Threat Team

The CERT Insider Threat Center is part of the Enterprise Threat and Vulnerability Management (ETVM) team in the CERT Division of the Software Engineering Institute at Carnegie Mellon University. The ETVM team helps organizations improve their security posture and incident response capability by researching technical threat areas; developing information security assessment methods and techniques; and providing information, solutions, and training for preventing, detecting, and responding to illicit insider activity. ETVM team members are domain experts in insider threat and incident response, and team capabilities include threat analysis and modeling; development of security metrics and assessment methodologies; and creation and delivery of training, courses, and workshops. Our insider threat database allows us to examine broad and specific trends.

For additional information regarding the content of this white paper or other research conducted at the CERT Insider Threat Center, please contact insider-threat-feedback@cert.org.