Spotlight On: Programming Techniques Used as an Insider Attack Tool

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

December 2008

This work was funded by

Copyright 2008 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY 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.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder. Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be directed to **permission@sei.cmu.edu.**

Spotlight On: Programming Techniques Used as an Insider Attack Tool

This report is the first in a new quarterly series, *Spotlight On*, published by the CERT insider threat team and funded by CyLab. Each report will focus on a specific area of concern and present analysis based on the hundreds of actual insider threat cases cataloged in the CERT insider threat database. For more information about CERT's insider threat work, see http://www.cert.org/insider_threat/.

In this article, we focus on persons who used programming techniques to commit malicious acts against their organizations. We begin by providing a snapshot of the cases, then detail the actions taken by the insiders in each case. A summary of issues related to the cases follows, as well as references to best practices that might have been effective in countering these incidents.

We would like to thank our colleagues in the CERT Program who reviewed and provided valuable feedback on this article: Julia Allen, Robert Seacord, and Carol Woody.

Snapshot of Malicious Insiders Who Used Programming Techniques

Who they are

Insiders used programming tactics in only 15 of the more than 200 cases cataloged in CERT's insider threat database. Of those fifteen cases, two were technical managers, five were system administrators (two of those were contractors), two were students, two were consultants, one was a salesman, one was a computer technician, and two were programmers.

What they strike

Insiders used programming tactics to target the organizations that employed them, or the clients of those organizations, harming them in various ways. The targets included the following:

- **A regional Internet service provider (ISP)**. An employee re-programmed wireless access points to deny service to the ISP's customers for three weeks.
- **A professor's university account**. A student compromised the account and stole the personal information of both the professor and fellow students. The student was also able to change his own grades.
- **A state agency**. An IT manager in the agency reprogrammed a system so it would fail to notify the security office in the event of suspicious transactions.
- **A patient-specific, drug-interaction database**. A system administrator attempted to launch a logic bomb attack against a database used by pharmacists prior to dispensing medication. Had the logic bomb successfully executed, it would have wiped out the database.

When and where they strike

Ten of the insiders were current employees when they committed their crimes, although some planted logic bombs while employed that were set to execute after their termination. The majority of the employees struck at the workplace, but many launched their attack remotely.

Why they strike

Ten of the insiders were motivated by revenge against their employer. Four were motivated by financial gain. Other motives included recognition and excitement.

How they strike

Nine of the insiders in these cases inserted malicious code with the intent of causing harm to their organization or to individuals. Six of the insiders used logic bombs to carry out their attacks. Other attacks methods included

- social engineering
- sabotaging backup tapes
- compromising accounts
- deleting and modifying log files
- unauthorized access
- intentionally deploying a virus on customer systems

Similarities across Cases

While the number of cases analyzed for this article is limited, there are similarities worth noting. The majority of these cases were IT Sabotage cases,[1] which follow the escalation patterns documented in CERT's MERIT model.[2] The MERIT model is a system dynamics model of the insider IT sabotage problem that elaborates complex interactions in the domain and unintended consequences of organizational policies, practices, technology, and culture on insider behavior.

In each of the fifteen cases, changes made by the insider may have been detected prior to the malicious code being deployed had the organization had change controls in place to detect unauthorized modifications to critical systems and software. Some of the organizations did use configuration management tools to track and log changes to critical software. However, either the tools did not prohibit software from being released without approval from a trusted second person, or the organization failed to audit the change control logs for unauthorized changes.

Types of Illicit Activity

The objectives of the malicious activity carried out via programming techniques included

- overriding security measures to conceal further illicit activity
- obtaining unauthorized authentication credentials
- disrupting service or destroying organization systems and/or data
- stealing proprietary or confidential information
- embarrassing the organization

[1] See *Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors* for more details on insider IT sabotage. http://www.cert.org/archive/pdf/insidercross051105.pdf

[2] See *The "Big Picture" of Insider IT Sabotage Across U.S. Critical Infrastructures* for a description of CERT's MERIT model of insider IT sabotage. http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tn041.pdf

The following case summaries describe how insiders modified production source code or scripts to perpetrate their attacks. The methods used to achieve these objectives suggest countermeasures that should be considered to help mitigate risks associated with these types of insider attacks; countermeasures are summarized in the Conclusions section following the case examples.

Modification of Production Source Code or Scripts

1. A manager in the IT department at a state agency was able to carry out a fraud scheme for two years after commenting out a single line of source code. The code he commented out sent a notice to the security office any time a certain system function was used. Once this code was commented out, compiled, and released into production, he was then able to change information in the system without his actions being detected. He carried out his fraud scheme for more than two years.

2. An insider employed as a consultant modified source code used by his former employer, an ISP, and disabled its communications capability for three weeks. He gained remote access to the ISP's radio-tower computer, then used administrator passwords to re-program the wireless access points of 110 of its customers, cutting off their Internet service. He reprogrammed the access points to complicate repair efforts, requiring the service provider to dispatch technicians to the premises of the subscribers who lost Internet access, an effort that extended over a three-week period. His actions also disrupted the communications of other ISPs outside the victim's network.

3. A system administrator, fearing company layoffs, embedded malicious code within other scripts on the organization's servers. The code was set to execute on his next birthday, approximately six months in the future. Had he been successful, the code would have wiped out critical data on more than 70 servers and caused widespread financial damage. It also would have caused potential health risks to the organization's customers. Even after surviving the layoffs a few days later, the insider did not remove the malicious code; in fact, he modified it one month later. The malicious code contained a programming error and failed to execute on his birthday as scheduled. However, the insider allegedly corrected the programming error six months later, setting the code to execute on his next birthday. Fortunately, a few months before the intended execution date, another system administrator investigating a system error discovered the malicious code and disabled it.

4. A contractor hired as a system administrator wrote a logic bomb to delete all of the organization's files. He placed the logic bomb in two different scripts. The first was in an operating system script that rotated log files when a volume reached a certain point; rather than rotating log files it would execute his logic bomb. He placed the second logic bomb in his supervisor's log-in script. The logic bomb was set up to display a threatening and insulting message to his supervisor during login, execute the logic bomb, and remove all traces of the logic bomb from the system, including log files.

5. A programmer on a software development team added two additional lines of code to his employer's premier product, an inter-network communication interface. When triggered, the new code randomly inserted the octal code for the letter "i" into the transmission stream during the protocol initialization. The insider checked-in the modified code using his boss's computer and username on a development platform for versioning control. Five months after the insider voluntarily left the company, the logic bomb executed

and began to randomly insert the letter “I” into the organization’s communication stream, disrupting customer services.

6. Following termination, a former application developer at a consumer data marketing firm remotely logged into the organization’s systems and modified its web site by inserting pornographic images. While this attack did not definitively use programming techniques, we chose to include it in this article due to the serious consequences.

Obtaining Unauthorized Authentication Credentials

1. An IT manager modified the password synch program that propagated password changes between the production and development systems. The insider was the only person on staff who knew this program existed. By removing a single line of code, he altered the program to store all password changes (account, old password, and new password) in a file as clear text, thereby gaining access to all account passwords that had been changed using the program.

2. A student employed multiple methods for gaining access to unauthorized authentication credentials; the most devious involved programming methods. First, he decrypted the password file on a departmental computer system and obtaining the password for his professor’s account. Using one of the passwords he obtained, he was able to gain access to the professor's personal account on Yahoo.com. The student wrote and installed a program in the professor's computer account that would run when the professor logged in, requesting the professor to enter his user ID and password for the University's administrative computer system, a separate network and computer system. That program enabled him to surreptitiously capture the professor's user ID and password for that network and computer system.

Disruption of Service and/or Theft of Information

1. A computer-science major wrote a malicious program that, on several occasions, shut down the university server that was used as a portal for enrollment services. His real intention was theft: over a two-year period, he accessed a database on the server and stole 8,000 names and Social Security numbers along with 37,000 personal records.

2. A system administrator working as a contractor planted several logic bombs on the organization’s network after it rejected his proposal to replace one of his fellow system administrators. When the organization decided to award the work to another firm, he planted logic bombs on five servers scheduled to detonate after he left. Three of the servers went offline when they executed, but the system administrator located the malicious code and prevented it from executing on the other two targeted computers.

3. A consultant hired as a software developer accessed his client’s servers remotely and removed some code needed to run the system, rendering the organization’s systems inaccessible. This action followed a year of unmet demands and threats by the consultant. The insider intended to return the code once his demands were met.

4. A salesman colluded with an outsider employed at a competing firm to steal information from his organization. The outsider sent the salesman an email with a virus attachment. The salesman downloaded the attachment, a keystroke-logging virus, to one of his organization's computers. The malicious code then collected information from the victim organization's computer system and sent that information to the outsider's computer. The salesman installed the program on several computers.

5. A system administrator, disgruntled because his yearly bonus was not a large as expected, built and deployed a logic bomb that deleted 10 billion files and took down nearly 2,000 servers in the main office and 370 servers at branch offices around the country. He was able to distribute the malicious code by using the standard software distribution methods.

6. After convincing the organization to systematically centralize the critical manufacturing programs, a disgruntled system administrator was able to develop and deploy a software time bomb to detonate after he was terminated. The individual became disgruntled when he began to lose technical and administrative control of the systems he built from the ground up. He wrote and tested the logic bomb three times on the organization's network before deploying it. In addition, he sabotaged the backups to hinder the recovery effort of the organization.

7. When the organization changed the salary structure for its employees, a computer technician downloaded a virus from the internet, included the virus in other production source code, and uploaded the malicious code to five customer sites. The insider felt the new salary program, which gave all employees a 25 percent increase in salary instead of rewarding exemplary employees with performance bonuses, failed to reward only the hard working employees. Critical systems at the customer sites were disabled when the viruses were executed.

8. A programmer and manager of a credit service firm's network placed a "time bomb" on the network that deleted and modified more than 50,000 debtor accounts and disrupted the firm's computer network. The insider had been advised of adverse employment issues and was placed on a performance improvement plan. A few weeks later, when he failed to show up at work without notice, he was terminated. However, malicious code he had previously placed on the system had been programmed to delete and modify data at the end of the month. The investigation uncovered evidence that the malicious code was written to delete the source code (self destruct), and that the insider had deleted system logs that had recorded his online activity related to the malicious code.

Conclusions

It is possible for insiders to create malicious code or modify existing code to harm an organization even if they do not develop software as part of their assigned job responsibilities. As seen from the case examples, insider programming attacks can be launched against a variety of targets. The majority of the incidents studied by CERT were current employees at the time they set up the attack, and a significant number of attacks occurred during working hours. Insiders were largely motivated by revenge. They attacked by modifying source code, gaining unauthorized credentials, denying service, and embarrassing the organization as shown in Table 1.

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12	Case 13	Case 14	Case 15	Total	% Total
Behavioral issues																	
Disgruntlement	x	x				x		x			x	x	x	x	x	9	60%
Promotion, compensation, employment status issues						x	x	x		x	x		x	x	x	8	53%
Concerning behavior or activity	x	x					x	x			x	x	x	x	x	9	60%
Social engineering or recruitment	x								x							2	13%
Previous incidents or policy violations	x				x		x					x		x		5	33%
Technical issues																	
Altered system executables	x		x	x	x	x	x	x	x		x	x	x	x	x	13	87%
Altered critical data		x	x			x				x		x				5	33%
Denied critical services		x			x			x		x	x		x	x	x	8	53%
Accessed systems after termination		x										x				2	13%
Exported critical data					x											1	7%

Table 1: Summary of Cases

The systems, software, and processes at the victim organizations typically contained multiple vulnerabilities and, in many cases, insiders were able to use different attack strategies to exploit common vulnerabilities. A separate CERT report, *The Common Sense Guide to Prevention and Detection of Insider Threats,*[3] provides a discussion of best practices that would have been effective in preventing or detecting malicious insider activity in the actual insider threat cases collected and analyzed by CERT. Practices from that report that are applicable to these particular cases are summarized below.

PRACTICE 3*: Institute periodic security awareness training for all employees.* In a few of these cases, the malicious insider enlisted the assistance of unwitting coworkers in carrying out the crime. A culture of security awareness must be instilled in the organization so that all employees understand the need for policies, procedures, and technical controls. All employees in an organization must be aware that security policies and procedures exist, that there is a good reason why they exist, that they must be enforced, and that there can be serious consequences for infractions. They also need to be aware that individuals, either inside or outside the organization, may try to co-opt them into activities counter to the organization's mission. Each employee needs to understand the organization's security policies and the process for reporting policy violations.

PRACTICE 4*: Monitor and respond to suspicious or disruptive behavior, beginning with the hiring process.* The prevalence of behavioral issues in the table above indicates that this practice is relevant to preventing the types of incidents described in this article. Organizations should closely monitor suspicious or disruptive behavior by employees before they are hired, as well as in the workplace after they are hired, including repeated policy violations that may indicate or escalate into more serious criminal activity. Signs of

[3] http://www.cylab.cmu.edu/pdfs/CommonSenseGuideInsiderThreats-V3.pdf

disgruntlement exhibited through concerning behavior or activity may allow earlier detection of increasing risk of insider attack.[4] Organizations should be particularly aware of the increased insider threat risk they face due to an insider's unmet expectations (as when passed over for promotion), perceived unfair compensation for work performed, or undesired changes in employment status.

PRACTICE 5: *Anticipate and manage negative workplace issues*. Some of the insiders in this article acted in response to negative workplace events or issues. The *Common Sense Guide* describes suggestions for organizations beginning with pre-employment issues and continuing through employment and termination issues. For example, employers need to clearly formulate employment agreements and conditions of employment. Responsibilities and constraints of the employee and consequences for violations need to be clearly communicated and consistently enforced. In addition, workplace disputes or inappropriate relationships between co-workers can serve to undermine a healthy and productive working environment. Employees should feel encouraged to discuss work-related issues with a member of management or human resources without fear of reprisal or negative consequences. Managers need to address these issues when discovered or reported, before they escalate out of control. Finally, contentious employee terminations must be handled with utmost care, as most insider IT sabotage attacks occur following termination.

PRACTICE 7*: Implement strict password and account management policies and practices.* Two of the cases in this article involved current employees who were able to gain access following termination due to inadequate password and account management. No matter how vigilant an organization is in trying to prevent insider attacks, if their computer accounts can be compromised, insiders have an opportunity to circumvent both manual and automated controls. Password and account management policies and practices should apply to employees, contractors, and business partners. They should ensure that all activity from any account is attributable to the person who performed it. An anonymous reporting mechanism should be available and used by employees to report attempts at unauthorized account access, including potential attempts at social engineering. Audits should be performed regularly to identify and disable unnecessary or expired accounts.

PRACTICE 8: *Enforce separation of duties and least privilege*. If all employees are adequately trained in security awareness, and responsibility for critical functions is divided among employees, the possibility that one individual could commit fraud or sabotage without the cooperation of another individual within the organization is limited. In the cases in this article, technical measures to enforce separation of duties for releasing changes to production source code might have prevented the crimes from being carried out. Requiring and enforcing regression testing by independent testers before releasing any modifications to critical systems can be effective in detecting both intentional and unintentional introduction of malicious code.

PRACTICE 9*: Consider insider threats in the software development life cycle*. This practice is obviously applicable to the cases detailed in this article. In addition to cases involving malicious code, other cases are also considered in this practice that resulted from other types of vulnerabilities introduced throughout the Software Development Life Cycle.

PRACTICE 10*: Use extra caution with system administrators and technical or privileged users.* System administrators and privileged users, like database administrators, have the technical ability and access to commit and conceal malicious activity. Technically adept individuals are more likely resort to technical means to exact revenge for perceived wrongs. Techniques like separation of duties or two-man rule for critical system administrator functions, non-repudiation of technical actions, encryption, and disabling accounts upon termination can limit the damage and promote the detection of malicious system administrator and privileged user actions.

[4] See "The 'Big Picture' of Insider IT Sabotage Across U.S. Critical Infrastructures," SEI Technical Report CMU/SEI-2008-TR -009, available at http://www.cert.org/archive/pdf/08tr009.pdf.

PRACTICE 11: *Implement system change controls*. A wide variety of insider compromises relied on unauthorized modifications to the organization's systems, which argues for stronger change controls as a mitigation strategy. We acknowledge it is difficult for organizations to audit and monitor every transaction that occurs, but it is imperative that mission-critical software be closely monitored. Once baseline software and hardware configurations are characterized, technical controls can be implemented to protect critical files from unauthorized access, compare current configurations, detect discrepancies, and alert managers so that modifications are examined to detect unauthorized changes.

PRACTICE 12: *Log, monitor, and audit employee online actions*. If account and password policies and procedures are enforced, an organization can associate online actions with the employee who performed them. Logging, periodic monitoring, and auditing provide an organization the opportunity to discover and investigate suspicious insider actions before more serious consequences ensue. Some of the crimes described in this article continued well after the source code modifications were released, suggesting the organization had an opportunity to detect the crime earlier to prevent it altogether or to reduce the consequences.

PRACTICE 13: *Use layered defense against remote attacks*. Some current and former employees in the cases in this article used remote access to conduct their attacks. If employees are trained and vigilant, accounts are protected from compromise, and employees know that their actions are being logged and monitored, then disgruntled insiders will think twice about attacking systems or networks at work. Insiders tend to feel more confident and less inhibited when they have little fear of scrutiny by coworkers; therefore, remote access policies and procedures must be designed and implemented very carefully. When remote access to critical systems is deemed necessary, organizations should consider offsetting the added risk by requiring connections only via organization-owned machines, closer logging, and frequent auditing of remote transactions. Disabling remote access and collection of organization equipment is particularly important for terminated employees.

PRACTICE 14: *Deactivate computer access following termination.* When an employee terminates employment, whether the circumstances were favorable or not, it is important that the organization have in place a rigorous termination procedure that disables all of the employee's access points to the organization's physical locations, networks, systems, applications, and data. Timely disabling of all access paths and revocation of all credentials available to a terminated employee requires rigorous management. This should include all employee computer system accounts, shared passwords, and card control systems.

PRACTICE 15*: Implement secure backup and recovery processes.* No organization can completely eliminate its risk of insider attack; risk is inherent in the operation of any profitable enterprise. However, with a goal of organizational resiliency, risks must be acceptable to the stakeholders. Consequently, the impacts of potential insider attacks must be minimized. Therefore, it is important for organizations to prepare for the possibility of insider attack and minimize response time by implementing secure backup and recovery processes that avoid single points of failure. These processes should be tested periodically. The impact of insider attacks can be also mitigated through redundancy of critical services.

Although insider threat is a complex problem that is difficult to manage, there is hope. In some cases, organizations detected and stopped the malicious behavior before harm was inflicted. System defenders were able to successfully use technical controls to detect malicious insider actions, as well as recognize non-technical indicators (such as those detailed in the MERIT model) to guard against insider threats.

About the Insider Threat Team

The Insider Threat team is part of the Threat and Incident Management (TAIM) team in CERT. The TAIM 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 activity. TAIM 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 management system allows us to examine broad and specific trends.

For additional information regarding the content of this article or other research conducted at The Insider Threat Center at CERT, please contact Dawn Cappelli (dmc@cert.org).