Insider Threats in the SDLC

Lessons Learned From Actual Incidents of Fraud, Theft of Sensitive Information, and IT Sabotage

Dawn M. Cappelli
Andrew P. Moore
Randy F. Trzeciak

© 2006 Carnegie Mellon University

Financial Institution Discovers $691 Million in Losses…

End User Evades Auditors for 5 Years by Modifying Source Code

Customers Report Strange Disruptions in Telecommunications Firm's Operations...

Malicious Code Planted One Year Ago by Former Employee Modified Company's Communications Protocol

Purpose of This Presentation

Our goal here is to use lessons learned from actual cases to motivate you to ask yourself:

Could something like this happen to me?

© 2007 Carnegie Mellon University

Overview of Talk

Introduction

Evolution of CERT's insider threat research

Instructional case on insider threats in the software development life cycle (SDLC)

Summary of SDLC-related insider threat issues

Introduction

What is CERT?

Center of Internet security expertise

Established in 1988 by the US Department of Defense on the heels of the Morris worm that created havoc on the ARPANET, the precursor to what is the Internet today

Located in the Software Engineering Institute (SEI)

- Federally Funded Research & Development Center (FFRDC)
- Operated by Carnegie Mellon University (Pittsburgh, Pennsylvania)

Evolution of our Insider Threat Research

Insider threat case studies

- DOD Personnel Security Research Center (PERSEREC)
- CERT/U.S. Secret Service *Insider Threat Study*

Best practices

- CyLab *Common Sense Guide to Prevention and Detection of Insider Threats*

System dynamics modeling

- CyLab – *Management and Education on the Risk of Insider Threat (MERIT)*
- PERSEREC - *Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis*

Education & awareness

- CyLab – workshop
- CyLab – MERIT$_{IA}$

CERT/USSS *Insider Threat Study*

Definition of insider:

Current or former employees or contractors who

- *intentionally exceeded or misused an authorized level of access to networks, systems or data in a manner that*
- *targeted a specific individual or affected the security of the organization's data, systems and/or daily business operations*

Insider Threats During the SDLC

Scope

We defined cases of insider threats during the software development life cycle (SDLC) as:

- Insider threats enabled when security requirements were overlooked or insufficiently addressed
- Software/architecture design flaws facilitated insider threats
- Defects introduced during the software implementation process were later used by insiders for illicit activity
- Oversights when software systems were deployed from development to production enabled insiders to commit unauthorized acts
- Vulnerabilities introduced or exploited during software maintenance

Examples of Impacts

Company went out of business

Fraud losses up to $691 Million

Disruption of telecommunications services by telecom firm

Company website defaced

Virus planted on customers' systems

Source code wiped out for production, mission critical system

Instructional Case of Insider Threats in the SDLC

Insider Fraud at InsureACure, Inc.

We will hand out a case of insider fraud that involves SDLC issues

Please take a few minutes to review the case description

We will then conduct an interactive discussion of this case

Overview of Case Timeline

1. 1993: InsureACure established
2. 1994: Fischburn hired
3. 1998: Fischburn started side business
4. Early 12/2000: Rourke first calls Fischburn; MSP addresses sent
5. Mid 12/2000: Rourke requests address change; scheme set up
6. 12/20/2000: Fischburn obtains password
7. Holidays/2000: Fischburn executes scheme
8. 1/2001: Co-worker notices DB access; questions Fischburn
9. 1-7/2001: Fischburn continues scheme
10. 7/9/2001: Customer Support receives call identifying discrepancy; Fischburn overhears and starts cover up
11. 7/13/2001: Customer Support reports problem to O'Neil; investigation launched
12. 7/16-20/2001: O'Neil investigates problem with IT
13. 7/23-30/2001: O'Neil tries to work with Davidson to diagnose problem
14. 7/31/2001: O'Neil discovers full scope of problem; goes to HR with plea for help

Questions & Discussion

Questions about Case

What problems at InsureACure facilitated the fraud?

What should InsureACure do now (at end of case)?

What should InsureACure do in the future?

What problems at InsureACure facilitated the fraud?

- Technical
  - Flaws in SDLC
- Managerial
  - Organization conditions
  - Manager behavior

What did InsureACure do right?

What should InsureACure do now (at end of case)?

What should InsureACure do in the future to prevent, detect, and respond to such incidents more effectively?

- Technical Issues
- Management Processes

Summary of SDLC-Related Insider Threat Issues

Requirements Definition Oversights

Neglecting to define **authentication** and **role-based access control** requirements simplified insider attacks.

Neglecting to define **security requirements/separation of duties** for **automated business processes** provided an easy method for insider attack.

Neglecting to define requirements for **automated data integrity checks** gave insiders the security of knowing their actions would not be detected.

System Design Oversights

Insufficient attention to security details in **automated workflow processes** enabled insiders to commit malicious activity.

Insufficient **separation of duties** facilitated insider crimes.

- not designed at all
- no one to “check the checker”

Neglecting to consider security vulnerabilities posed by “**authorized system overrides**” resulted in an easy method for insiders to ”get around the rules”.

System Implementation Exploits

Lack of **code reviews** allowed insertion of "backdoors" into source code.

Inability to **attribute actions** to a single user enabled a project leader to sabotage his own team's development project.

System Deployment Oversights

Lack of enforcement of **documentation practices** and **backup procedures** prohibited recovery efforts when an insider deleted the only copy of source code for a production system.

Use of the same **password file** for development and the operational system enabled insiders to access and steal sensitive data from the operational system.

Unrestricted access to all customers' systems enabled a computer technician to plant a virus directly on customer networks.

Lack of **configuration control** and well-defined **business processes** enabled libelous material to be published to organization's website.

System Maintenance Issues

Lack of **code reviews** facilitated insertion of malicious code.

Ineffective **configuration control** practices enabled release of unauthorized code into production.

Ineffective or lack of **backup processes** amplified the impact of mass deletion of data.

End-user access to source code for systems they used enabled modification of security measures built into the source code.

Ignoring known **system vulnerabilities** provided an easy exploit method.

Summary – Most Prevalent SDLC Issues

IT Sabotage:

- Software architecture that allows for efficient recovery or sustains the organization during disasters
- Configuration and access control of source code
- Formal code review/inspection to prevent malicious code from being inserted into production applications

Fraud:

- Existence and enforcement of authorization/approval steps in automated work flow to ensure proper approvals for critical business functions

Theft of Sensitive or Confidential Information:

- Configuration and access control of source code

Takeaways?

Questions?

Points of Contact

Insider Threat Team Lead:
Dawn M. Cappelli
Senior Member of the Technical Staff
CERT Program
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890
+1 412 268-9136 – Phone
dmc@cert.org – Email

Business Development:
Joseph McLeod
Business Manager
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890
+1 412 268-6674 – Phone
+1 412-291-3054 – FAX
+1 412-478-3075 – Mobile
jmcleod@sei.cmu.edu – Email

http://www.cert.org/insider_threat/