Engineering Secure Software

# INSIDER THREAT

# Lottery Story

# A Threat We Can't Ignore

- Documented incidents are prevalent
  - Carnegie Melon's SEI has studied over 700 cybercrimes originating from insider threat since 2000
- Many more occurring
  - In 2007, the Secret Service et al. conducted a survey of law enforcement officials & security execs
    - 31% of electronic crimes involved an insider
    - 49% of respondents experienced insider threat in the past year
- Wikileaks, anyone?

# What is insider threat?

- Actors
  - Current employees
  - Former employees (esp. “recently former”)
  - Contractors
- Intentionally exceeded or misused an *authorized* level of access
- Affected the security of the organization
  - Data
  - Intellectual property
  - Daily business operations

# Double Threat to SE

- Insider threat affects SE in two ways
  - Insider *users* for the system that we release
    - e.g. hospital administrators
  - Insiders *developers* to our own software development company
    - e.g. disgruntled developers
- Liability considerations
  - Will our software facilitate insider threat?
  - Bring this up in your requirements elicitation meeting
    - Audit mechanisms
    - Deployment mechanisms
  - For everything else: hire some lawyers for a sneaky EULA

# Types of Insiders

- Pure insider

  e.g. system administrator, developer

- Insider associate

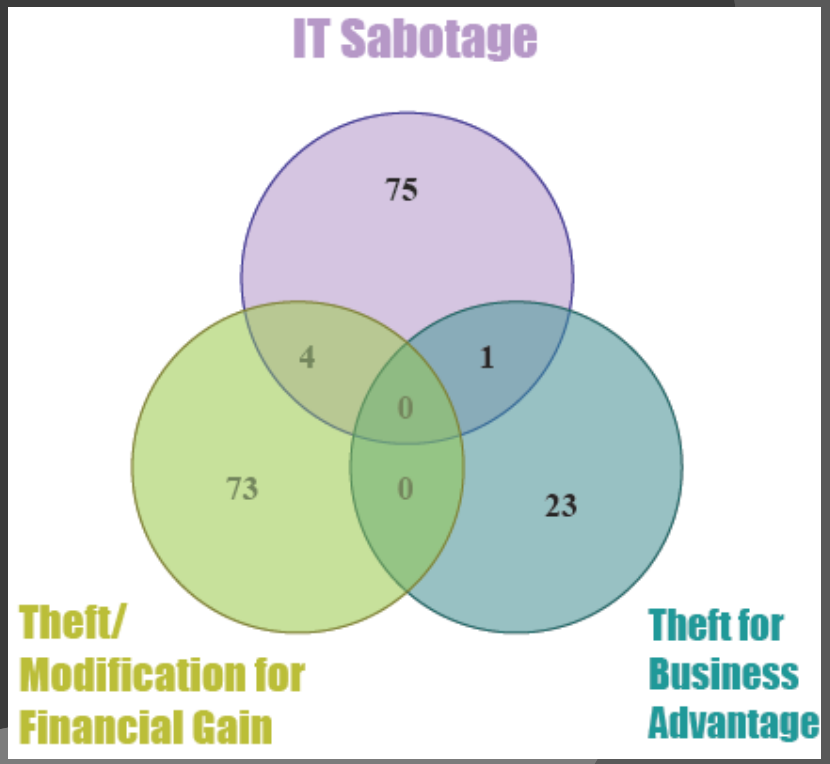
  e.g. developer, but on a different project

- Outside affiliate

  e.g. outsourced contractor

# Classes of Threats

- IT sabotage
- Personal financial gain
- Business advantage
  (e.g. industrial espionage)
- Miscellaneous



# Some considerations

- Majority of the attacks required significant planning ahead of time
- Majority of insider attacks took place physically on the premise
- Majority of insider attacks faced criminal charges
  - And in most cases, the insiders were aware that they would face charges

# Prevention vs. Detection

- Prevention is extraordinarily hard
  - Work environment
  - Predicting human nature
  - Deterrents are only somewhat effective
- Detection is much more feasible
  - Usually by someone using common sense
  - Audits of access logs
  - In most cases, live network detection was not involved
  - Drawback: reactive

# Developer Insiders

- “Security through obscurity alone” is really not an option
  - Insider would know what servers to go to
  - Insider knows the attack surface
- Access to production servers should be limited
  - Non-release changes to production need to be documented
  - Forces you to document your deployment process anyway
- On introducing backdoors
  - Very rarely introduced in the development phase
  - Most often in the maintenance phase

# General Suggestions

- Be aware of the threat
  - Keep up with the latest stories
  - Apply those situations to yours
- “Buddy” system

  Nobody should be left physically alone with important resources
- Logging and auditing
  - Everything is logged
  - Audits should actually happen

# More Suggestions

- Job termination policies
  - Have one.
  - Be prepared to disable accounts quickly
- Archives & offsite backups

  Mitigate tampering and destruction of backups
- Rotate duties
  - Better detection of anomalies
  - Better knowledge transfer anyway
- Holistic approach
  - People, data, technology, procedures, policies

# Some Resources

- SEI's CERT Insider Threat group
  - Definitive resource
  - http://www.cert.org/insider_threat/
  - http://www.cert.org/archive/pdf/ecrimesummary07.pdf
- *The Insider Threat: Combatting the Enemy Within, by Clive Blackwell*
  - ISBN 9781849280112
  - Available via RIT library electronically for free

# We More Need Stories

- …so that's today's activity
  - 5 groups
  - Assigned sectors for CERT case studies
  - Make a 5 minute presentation
    - Tell us stories of insider threat
    - Statistics
    - Some lessons learned