Security and Privacy: A Brief Overview

Tadayoshi (Yoshi) Kohno
University of Washington

University of Washington
Computer Science & Engineering

Challenges to Security

Why Are Technologies Insecure?

80%

1. Oversight: Designers didn't consider security and privacy
2. Choice: Designers chose to ignore security and privacy

10%

3. Definitions: What does security mean?
4. Mistakes: "Easy" solutions, flawed implementations (including usability problems)

10%

5. Fundamental challenges: Fundamental research challenges

Numbers are estimates; for emerging technologies (not desktop applications)

Security is Non-intuitive

- Computer security can be non-intuitive at first:
  - Mentality: Bad parties can be skilled, clever, sneaky, and cunning. Not "rational" by most people's definition. Goal is to cause *intentional* failures.
  - Imbalance: Bad parties only need to find *one* way to compromise the security of your system; defender must defend against *all* realistic attack vectors
  - Unpredictability: Bad parties "*win*" by doing what the defenders don't expect. Common expression:

    > "Anyone can design a system that they themselves cannot break."
- Next few slides: Survey common themes in security

Threat Modeling

- Security is about *threat modeling*:
  - Who are the potential attackers?
  - What are their resources and capabilities?
  - What are their motives?
  - What assets are you trying to protect?
  - What might the attackers try to do to compromise those assets?

- Need to answer these questions early, before you can even begin to make any conclusions about a real system

Common Fallacy #1

- Common fallacy #1: **“A system is either secure or insecure.”**

- Security is a gradient
- No such thing as a “perfectly secure system”
  - All systems are vulnerable to attacks
  - We’re interested in the *level* of security that a system provides (recall threat model)

Common Fallacy #2

- Common Fallacy #2: **“There’s never been an attack in the past, so security is not an issue”**
  - Many variants, like: **“There’s never been an attack in the past, so there won’t be in the future”**
- Above reasoning is *intuitive* but also *incorrect*.
- Equivalent to
  - “I’ve never been robbed, so I don’t need to lock my front door.”
- Problems with this:
  - It might have happened, you just don’t know because you haven’t been worrying about it.
  - Technology changes capabilities, incentives, and context so always new things attackers might do

Common Fallacy #2

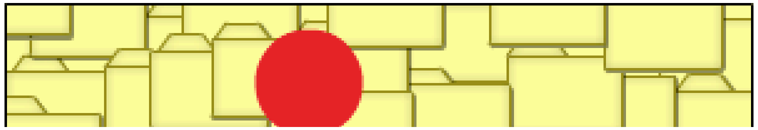
- Example: Ping-of-Death
  - When Microsoft created Windows 95, the developers thought that something "would never happen"
  - But then the Internet evolved, Windows 95 machines were hooked to the Internet ... and ... it happened!
  - Result: What's called the Ping-of-Death

Common Fallacy #3

- Common Fallacy #3: “**We use proprietary security algorithms, so the bad guys won’t know these algorithms and our system is secure.**”
- Flaw #1: Bad guys can learn these algorithms
  - Insiders, consultants, dumpster divers, corporate espionage, terrorists, ...
  - Bad guys could reverse engineer algorithms
- Flaw #2: Security through obscurity
  - Proprietary algorithms have a history of being less secure than standardized algorithms
  - Common saying “anyone can design a system they themselves cannot break”

MiFare RFID crack more extensive than previously thought

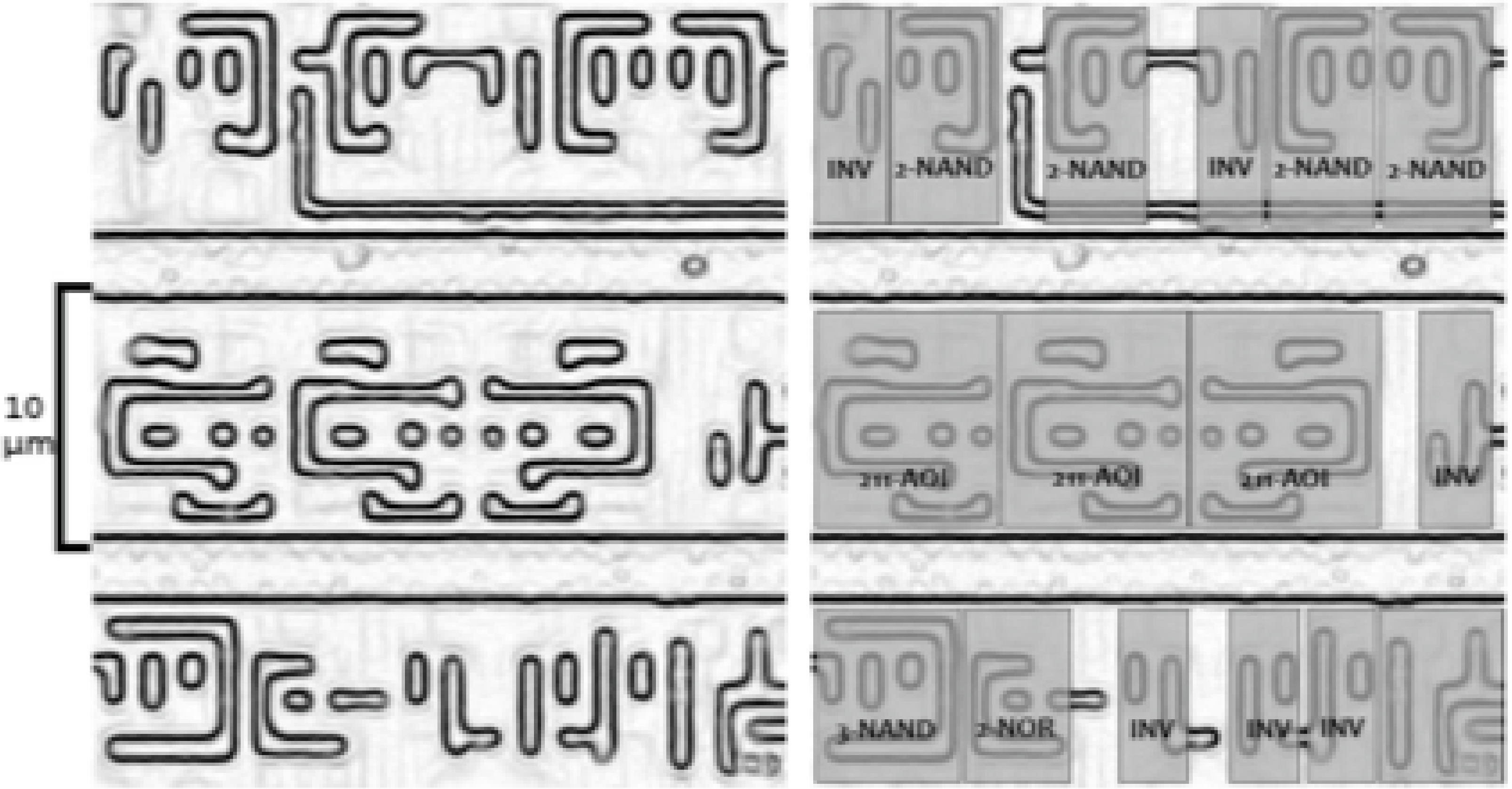
Seconds, not hours, to effect; plus version tappable too



By Geeta Dayal

April 15, 2008 (Cor
-- used daily by mi
passes and other a
thought, according
development Tues
conference in Istar

Mere seconds are
few hours, as estir
graduate student a
reverse-engineerin
takes only 12 secc
an ordinary laptop.



Common Fallacy #4

- Common Fallacy #4: “**We’re secure because we use standardized security algorithms like RSA, AES, SSL, ...**”
- Using standardized algorithms is a *good*, but *far from* sufficient
- Analogy:
  - Standardized security algorithms are like standardized locks
  - Locks themselves may be strong, but security of building depends on many other things (how you key the locks, how you attach locks to door, how door frame is mounted, whether you also lock the windows, etc)
- Many examples, e.g.,
  - Diebold Voting Machines

Common Fallacy #5

- Common Fallacy #5: “**We’ve addressed all known security concerns, so our system is now secure**”
- An example:
  - 2003: Identified security problems with the Diebold voting machine
  - 2004: Diebold introduced defenses to that specific attack; RABA re-evaluated and found that the fix *introduced a new security vulnerability*
  - 2007: Diebold introduced defenses to that new attack; we re-evaluated and found that the second fix *introduced another new security vulnerability*

Common Fallacy #6

- Common Fallacy #6: **“If we increase security, we’d be forced to decrease usability”**
- Challenging, but not impossible
- To make educated decisions and arguments we need to:
  - explore solution space,
  - gauge what’s possible, and
  - assess levels of security and usability provided by different solutions

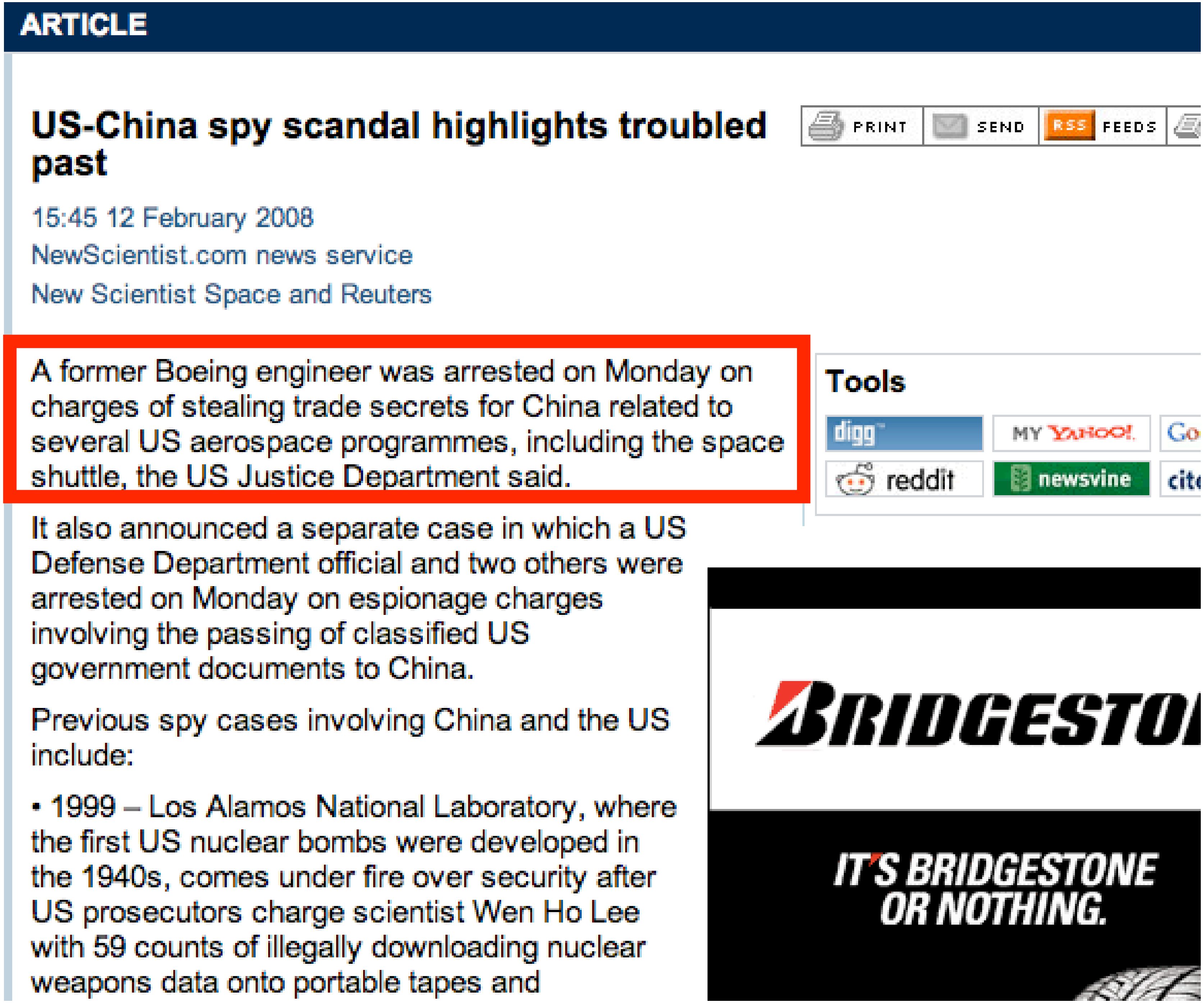
Common Fallacy #7

- Common Fallacy #7: “**Only sophisticated adversaries will be able to successfully attack our system**”
- Expression in security community:
  - “Attacks only get better, easier to mount over time”
- Some adversaries will be sophisticated (we return to this later)
- Different actors: Sophisticated bad guys create tools that less sophisticated bad guys use

Common Fallacy #8

- Common Fallacy #8: “**Insiders are not going to be adversaries**”
- Plenty of examples to the contrary (although companies don’t like to talk about it)

- Spies
- Greedy employees
- Disgruntled ex-employees
- ...



US-China spy scandal highlights troubled past

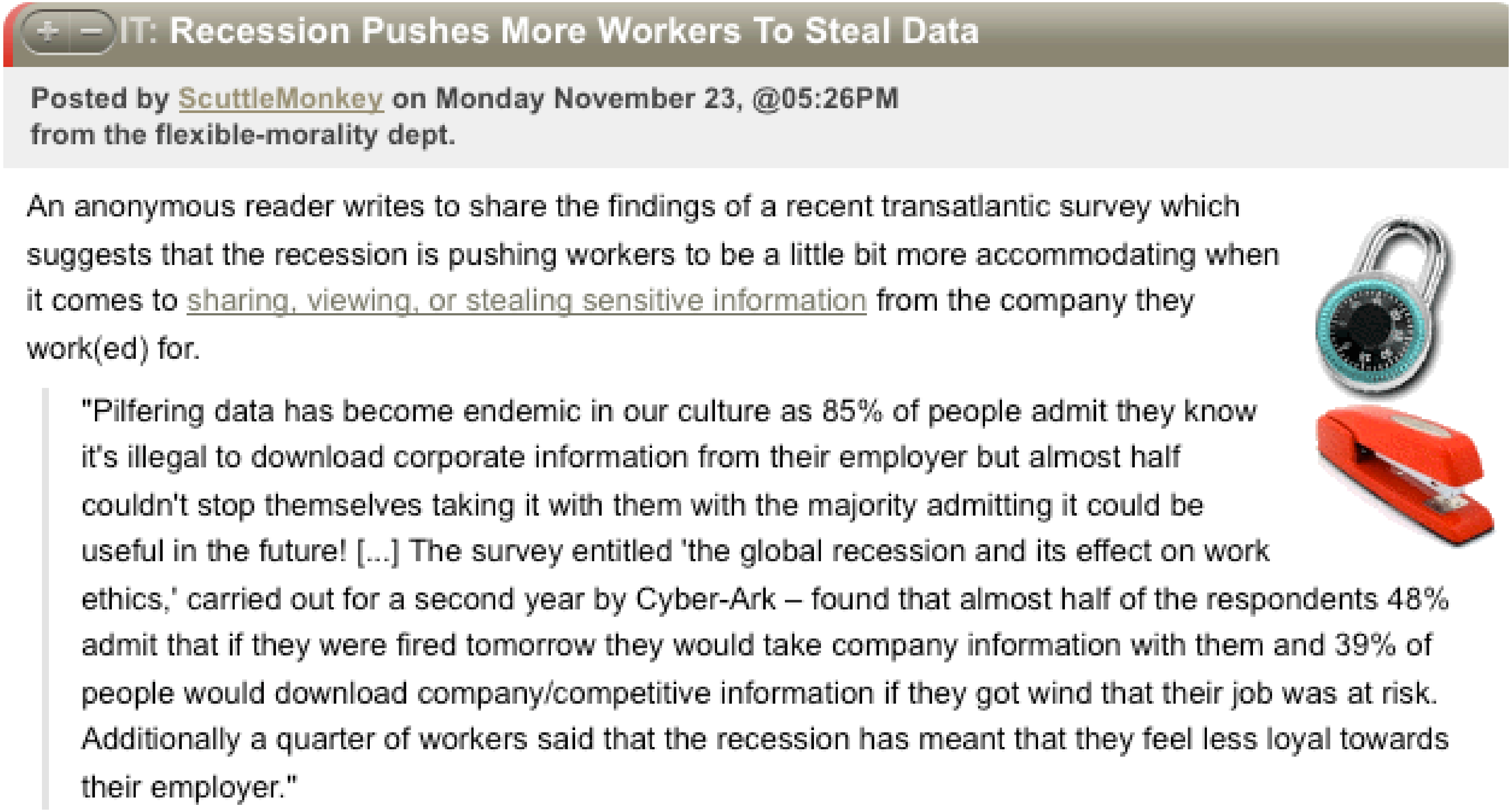
15:45 12 February 2008
NewScientist.com news service
New Scientist Space and Reuters

A former Boeing engineer was arrested on Monday on charges of stealing trade secrets for China related to several US aerospace programmes, including the space shuttle, the US Justice Department said.

It also announced a separate case in which a US Defense Department official and two others were arrested on Monday on espionage charges involving the passing of classified US government documents to China.

Previous spy cases involving China and the US include:

- 1999 – Los Alamos National Laboratory, where the first US nuclear bombs were developed in the 1940s, comes under fire over security after US prosecutors charge scientist Wen Ho Lee with 59 counts of illegally downloading nuclear weapons data onto portable tapes and



IT: Recession Pushes More Workers To Steal Data

Posted by ScuttleMonkey on Monday November 23, @05:26PM
from the flexible-morality dept.

An anonymous reader writes to share the findings of a recent transatlantic survey which suggests that the recession is pushing workers to be a little bit more accommodating when it comes to sharing, viewing, or stealing sensitive information from the company they work(ed) for.

> "Pilfering data has become endemic in our culture as 85% of people admit they know it's illegal to download corporate information from their employer but almost half couldn't stop themselves taking it with them with the majority admitting it could be useful in the future! [...] The survey entitled 'the global recession and its effect on work ethics,' carried out for a second year by Cyber-Ark – found that almost half of the respondents 48% admit that if they were fired tomorrow they would take company information with them and 39% of people would download company/competitive information if they got wind that their job was at risk. Additionally a quarter of workers said that the recession has meant that they feel less loyal towards their employer."

Common Fallacy #9

- Common Fallacy #9: “**We’ve thought of everything**”
- Doesn’t apply to computer security - can never *prove* to yourself that you’ve thought of all attackers
- Same thing applies to these slides: This list of common fallacies is not exclusive

How to Think about Security

Whole-System is Critical

- Securing a system involves a whole-system view
  - Cryptography
  - Implementation
  - People
  - Physical security
  - Everything in between

- This is because “security is only as strong as the weakest link,” and security can fail in many places
  - No reason to attack the strongest part of a system if you can walk right around it.

Analyzing the Security of a System

- First thing: Summarize the system as clearly and concisely as possible
  - Critical step. If you can't summarize the system clearly and concisely, how can you analyze it's security?

- Next steps:
  - Identify the assets: What do you wish to protect?
  - Identify the adversaries and threats
  - Identify vulnerabilities: Weaknesses in the system
  - Estimate the risks

Assets

- Need to know what you are protecting!
  - Hardware: Laptops, servers, routers, PDAs, phones, ...
  - Software: Applications, operating systems, database systems, source code, object code, ...
  - Data and information: Data for running and planning your business, design documents, data about your customers, data about your identity
  - Reputation, brand name
  - Responsiveness
- Assets should have an associated value (e.g., cost to replace hardware, cost to reputation, how important to business operation)

Adversaries

- National governments
- Terrorists
- Thieves
- Business competitors
- Your supplier
- Your consumer
- New York Times
- Your family members (parents, children)
- Your friends
- Your ex-friends
- …

Threats

- Threats are actions by adversaries who try to exploit vulnerabilities to damage assets
  - Spoofing identities: Attacker pretends to be someone else
  - Tampering with data: Change outcome of election
  - Denial of service: Attacker makes voting machines unavailable on election day
  - Elevation of privilege: Regular voter becomes admin

- Specific threats depend on environmental conditions, enforcement mechanisms, etc
  - You must have a clear, simple, accurate understanding of how the system works!

Threats

- Several ways to classify threats
  - By damage done to the assets
  - By the source of attacks
    - (Type of) insider
    - (Type of) outsider
    - Local attacker
    - Remote attacker
    - Attacker resources

- I like to think of a matrix
  - Adversaries on one axis
  - Assets on the other axis

Vulnerabilities

- Weaknesses of a system that could be exploited to cause damage
  - Accounts with system privileges where the default password has not been changed (Diebold: 1111)
  - Programs with unnecessary privileges
  - Programs with known flaws
  - Known problems with cryptography
  - Weak firewall configurations that allow access to vulnerable services
  - ...
- Sources for vulnerability updates: CERT, SANS, Bugtraq, the news(?)

Risks

- Quantitative risk management
  - Example: Risk = Asset × Threat × Vulnerability
  - Monetary value to assets
  - Threats and vulnerabilities are probabilities
  - (Yes: Difficult to assign these costs and probabilities)
- Qualitative risk management
  - Assets: Critical, very important, important, not important
  - Vulnerabilities: Has to be fixed soon, should be fixed, fix if convenient
  - Threats: Very likely, likely, unlikely, very unlikely

Let’s try it out

- Pick a system
- Identify assets
- Identify threats
- Identify potential vulnerabilities