How to Defend the Universe from Evil-doers

A Guide for Software Developers and Security Teams

Bruce Jenkins
Managing Consultant
20 Jun 2011

bcjenkins@hp.com

About the Presenter

Bruce Jenkins (Major, USAF, Ret.) enlisted in the US Air Force in 1979 as a weapons control systems technician for the F-4 Phantom. After assignments to military bases in Denver, Colorado, and the Mojave Desert in California, he spent 10 years in Germany, with short assignments to Spain and the United Kingdom. In 1992, after completing a BS in computer science, he transferred to the computer-communications field and performed technical analysis on NORAD's Integrated Tactical Warning / Attack Assessment network in Colorado Springs. In 1995 Jenkins was commissioned a second lieutenant and then managed the Crime and Counterintelligence Terrorism Information System at the Office of Special Investigations in Washington, D.C. From 1998 to 2000, he obtained his MS in operations research (management science) in Dayton, Ohio, after which he project managed wargame simulation software development at the Wargaming Institute in Montgomery, Alabama. He then was CISO at the College for Professional Development before spending 14 months commanding a communications squadron in Kuwait. He returned to Montgomery in March 2005, where he was responsible for systems security policy and compliance. He led the Crisis Action Team following a USAF personnel system breach, and then managed an 11-month pilot program to evaluate software security products. His final project before his USAF retirement in 2007 was to design the framework and resource requirements for what is now the Application Software Assurance Center of Excellence (ASACoE). Mr. Jenkins then joined HP Fortify, where he assists organizations in developing software security assurance strategies and programs.

Agenda

- Why Software Security?
- Software (Dev) vs. Security
- How to Save the Day (Seriously)

What this presentation is based on…

- Four years of anecdotal accounts from software security consultants working with clients
- Over 300 software security assessments spanning DoD, finance, retail, utilities, ISVs, systems integrators…
- Personal involvement in over 60 professional services engagements since July 2007

What this presentation is based on…

- Four years of anecdotal accounts from software security consultants working with clients
- Over 300 software security assessments spanning DoD, finance, retail, utilities, ISVs, systems integrators…
- Personal involvement in over 60 professional services engagements since July 2007
- And… really intense conversations with “passionate” developers and security teams

So… Why Software Security?

"Enough is Enough: The Threats Have Changed"

– Michael Howard and Steve Lipner

The Security Development Lifecycle

Why Software Security?

Why Software Security?

1. Customer Demands
2. Regulatory Compliance
3. Breach / Data Loss
4. Well-informed, Proactive

   (This group has been breached and they're just not admitting it.)

Security Spending and Regulation Continue to Increase

Computer Security Act of 1987

FFIEC IT Exam Handbook

EU Data Protection Directive

HIPAA

FD 21CFR Part 11

GLBA

NIST

PIPEDA (Canada)

USA Patriot Act

EC 8th Directive

PCI

EC Data Privacy

FISMA I and II

CA SB 1386

NERC CIP

SOX

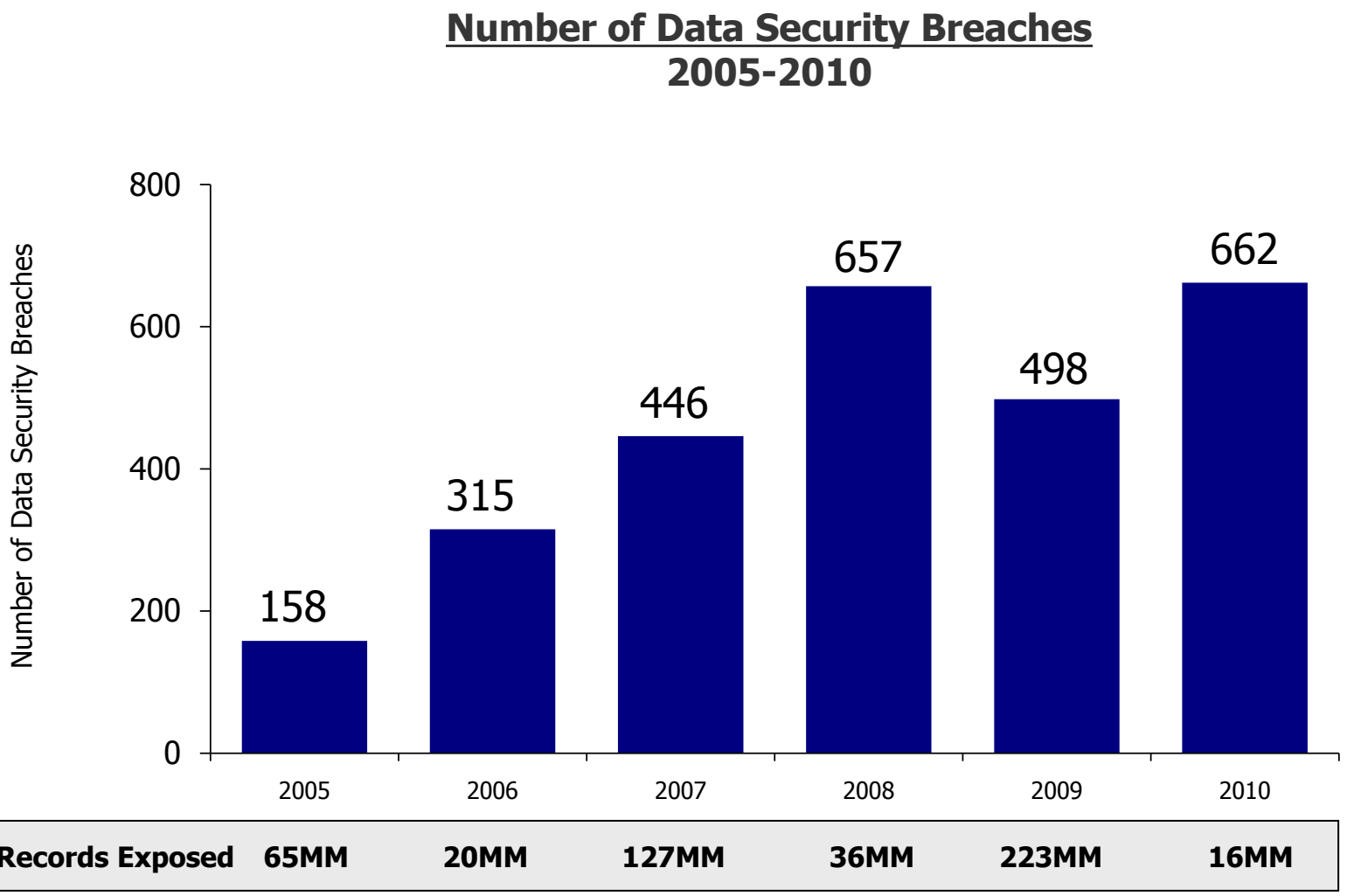
Basel II Accord

National Credit Union Administration regulations

National Defense Authorization Act

1980s **1990s** **2000s**

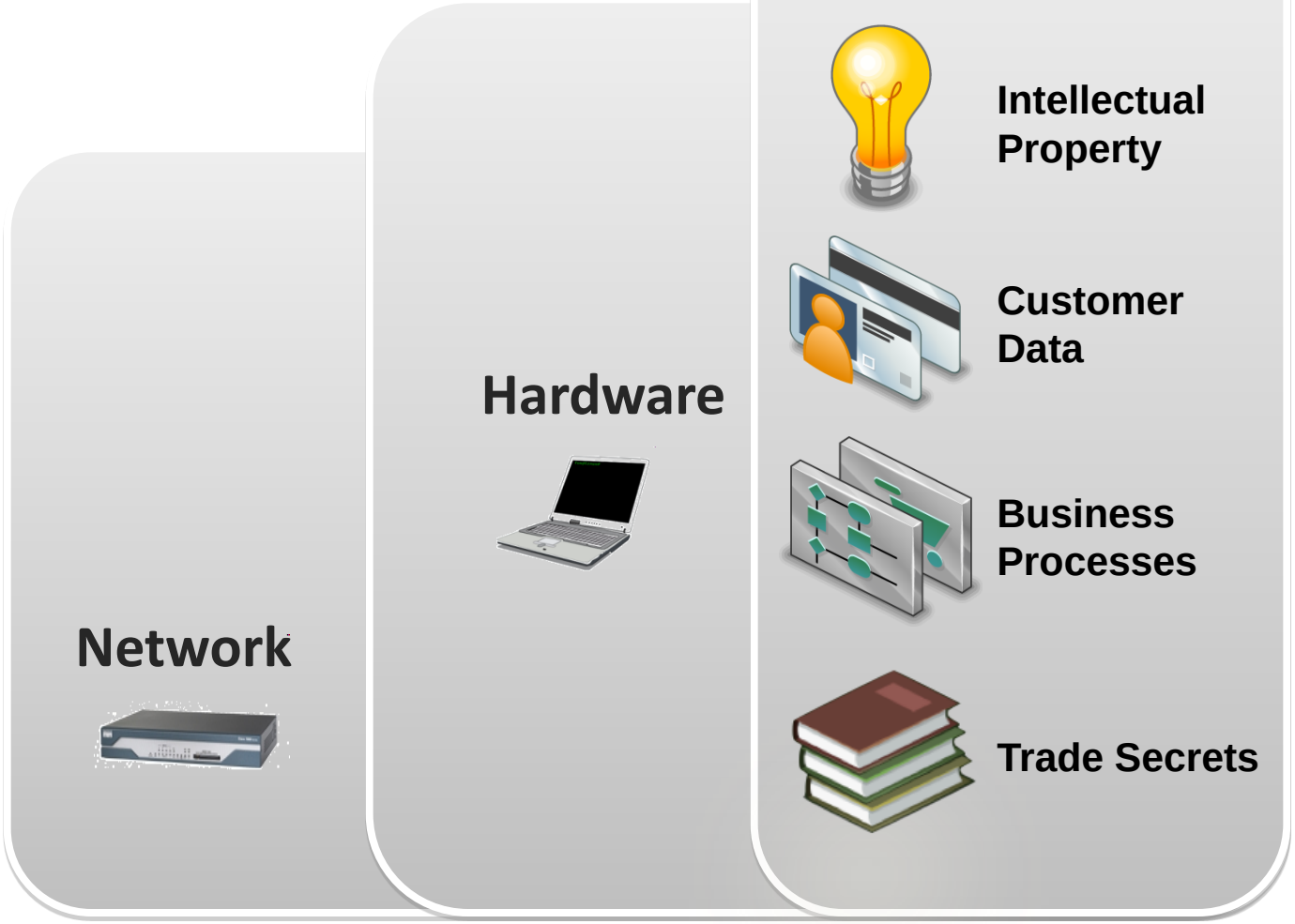
Yet, Security Breaches Continue



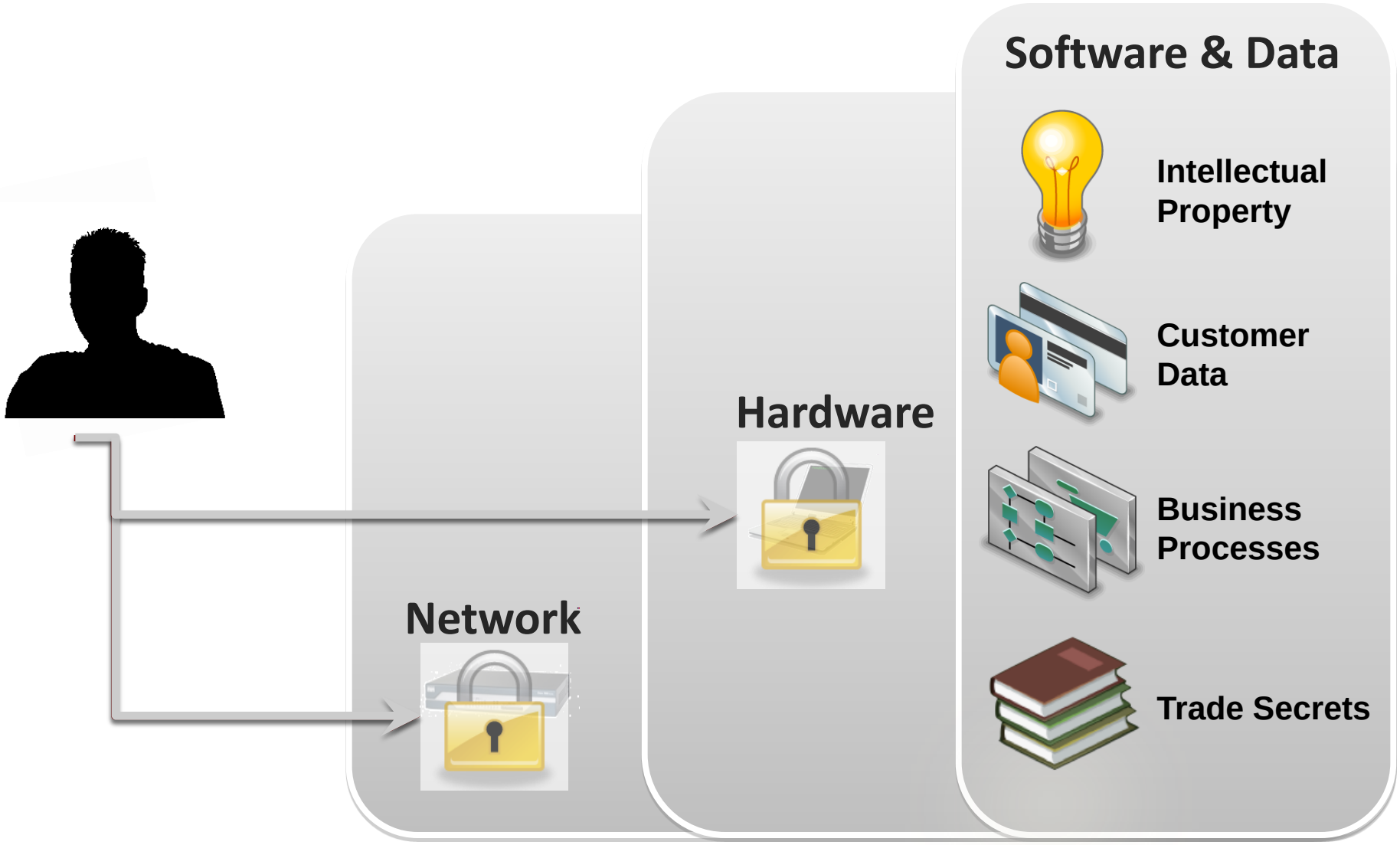
# of Records Exposed	65MM	20MM	127MM	36MM	223MM	16MM

Source: Identify Theft Resource Center

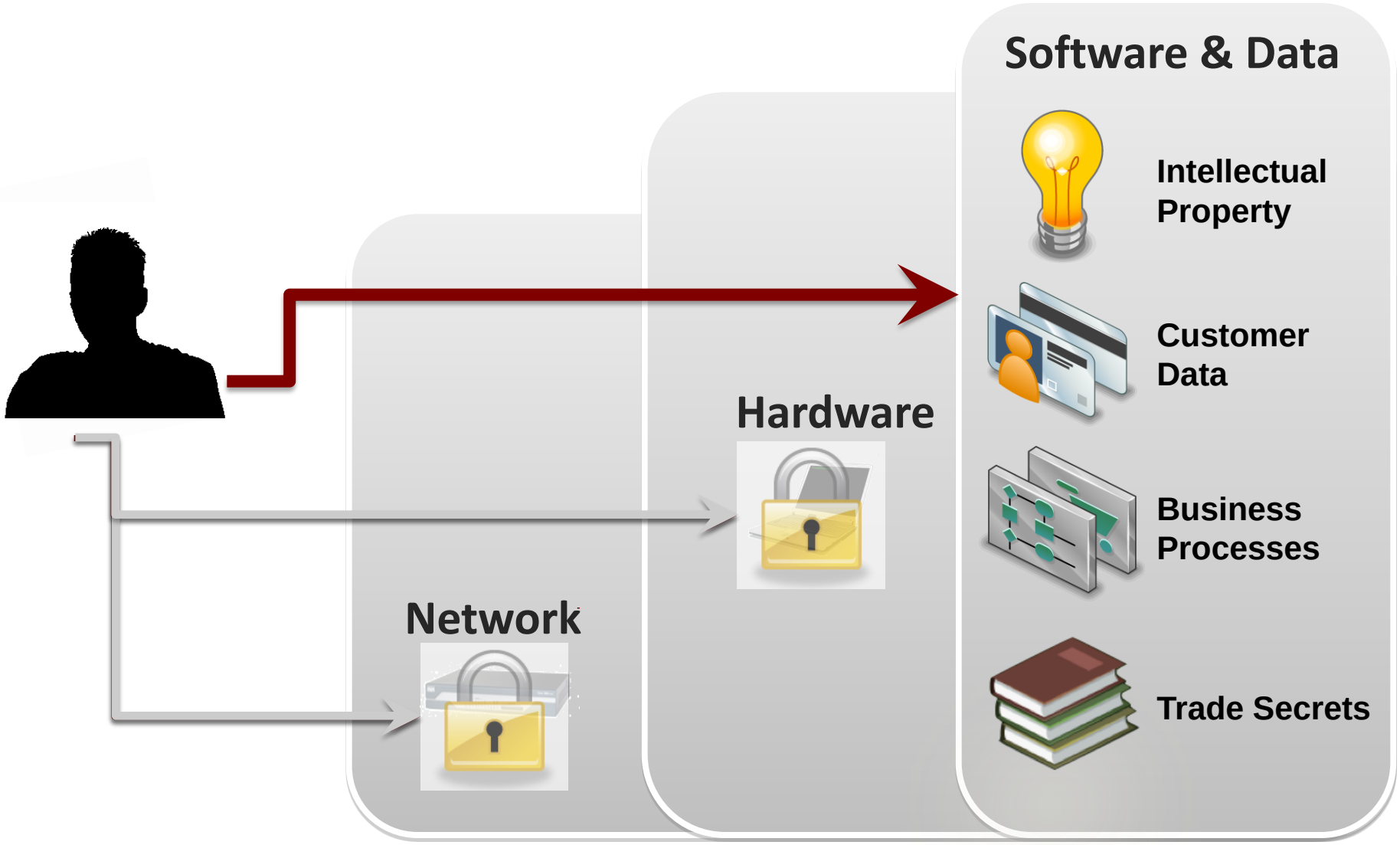
Why Software is Attacked



Why Software is Attacked



Why Software is Attacked



Exploiting Weaknesses: Path of Least Resistance

Quality Issue or Security Issue?

Quality Issue or Security Issue?

"Security is never black and white, and context matters more than technology"

– Bruce Schneier

Secrets & Lies: Digital Security in a Networked World

So… Quality Issue or Security Issue?

Quality Issue or Security Issue?

3rd largest US payment processer

The Incident

- Breach reported Jan 2009
- 94M credit records stolen
- Fines levied to banks **> $6M**
- Total cost of damages / loss **> $140M**

The Attack

- Personnel application attacked by **SQL Injection**
- Attackers inject code into data processing network
- Credit card transactions stolen

Who is Responsible for Software Security?

Who is Responsible for Software Security?

"I just want to be a coder; I'm really not interested in security."

– Anonymous

Cut the Developers Some Slack?

"Everyone knows that debugging is twice as hard as writing a program in the first place. So if you are as clever as you can be when you write it, how will you ever debug it?"

– Brian Kernighan

The Elements of Programming Style

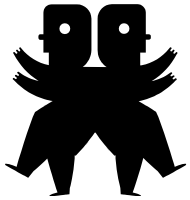
Cut the Developers Some Slack? (No way!)

"How do I get the software engineering teams to wake up and start taking software security seriously?"

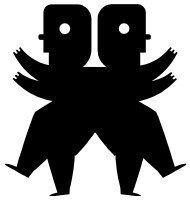
– Brad Arkin, Senior Director of Product Security and Privacy, Adobe Systems

IEEE Security & Privacy, May / June 2011

Software (Dev) vs. Security



Software (Dev) vs. Security



(Don’t worry—it’s not really that bad.)

Software (Dev) vs. Security: Four "Concerns"

- **Awareness**
- **Education, Training**
- **Issue Management**
- **Source Integrity (*this is about trust)***

Viewpoint: Software Developer

- **Awareness**
  - Don't know about the issue
  - Don't know about the *requirement*
- **Education, Training**
  - Don't know how to fix it
  - Definitely don't have *time* to get trained on how to fix it
- **Issue Management**
  - What am I going to do with 35,000+ "findings"?
  - No way these are legit—these definitely are *false positives*!
- **Source (Messenger) Integrity**
  - Those security guys don't know what they're talking about!
  - They don't understand how we write our code.

Viewpoint: Security Team

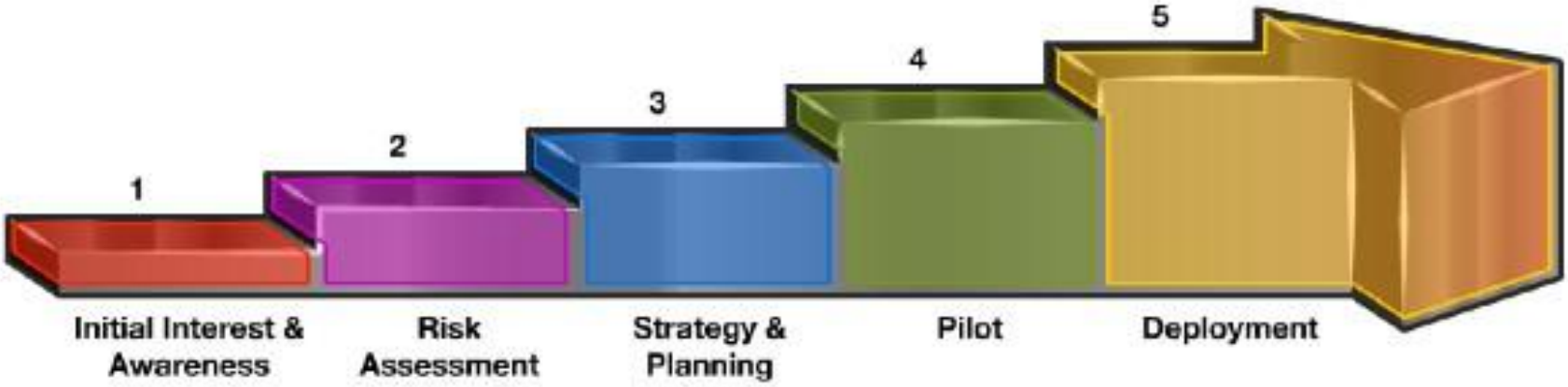
- **Awareness**
  - These issues are common knowledge.
  - The requirement for security is inherent.
- **Education, Training**
  - Don’t know how to fix it.
  - Too busy to show up for “developer training.”
- **Issue Management**
  - Why can’t these guys just fix this stuff? They have the whole list….
  - We tell them what’s important, and they tell us that it isn’t an issue.
- **Source (Code) Integrity**
  - Why is their code so messed up!?
  - Developers are sneaky; they’ll do anything to not look bad.

How to Save the Day…

1. **Obtain Executive Sponsorship**
   - Influence spans business units
   - Supports… and holds accountable
2. **Define Program Goals**
   - Associate AppSec goals to company goals
   - Consider tying to MBOs
3. **Develop a Reasoned Strategy** (with Objectives!) for supporting Program Goals
   - Keep it simple
   - Ensure Objectives are measurable and time-boxed

Strategy Example

"**Implement a five-phased approach** to raising awareness of application security, educating and training stakeholders on process changes, and building security into the SDLC."



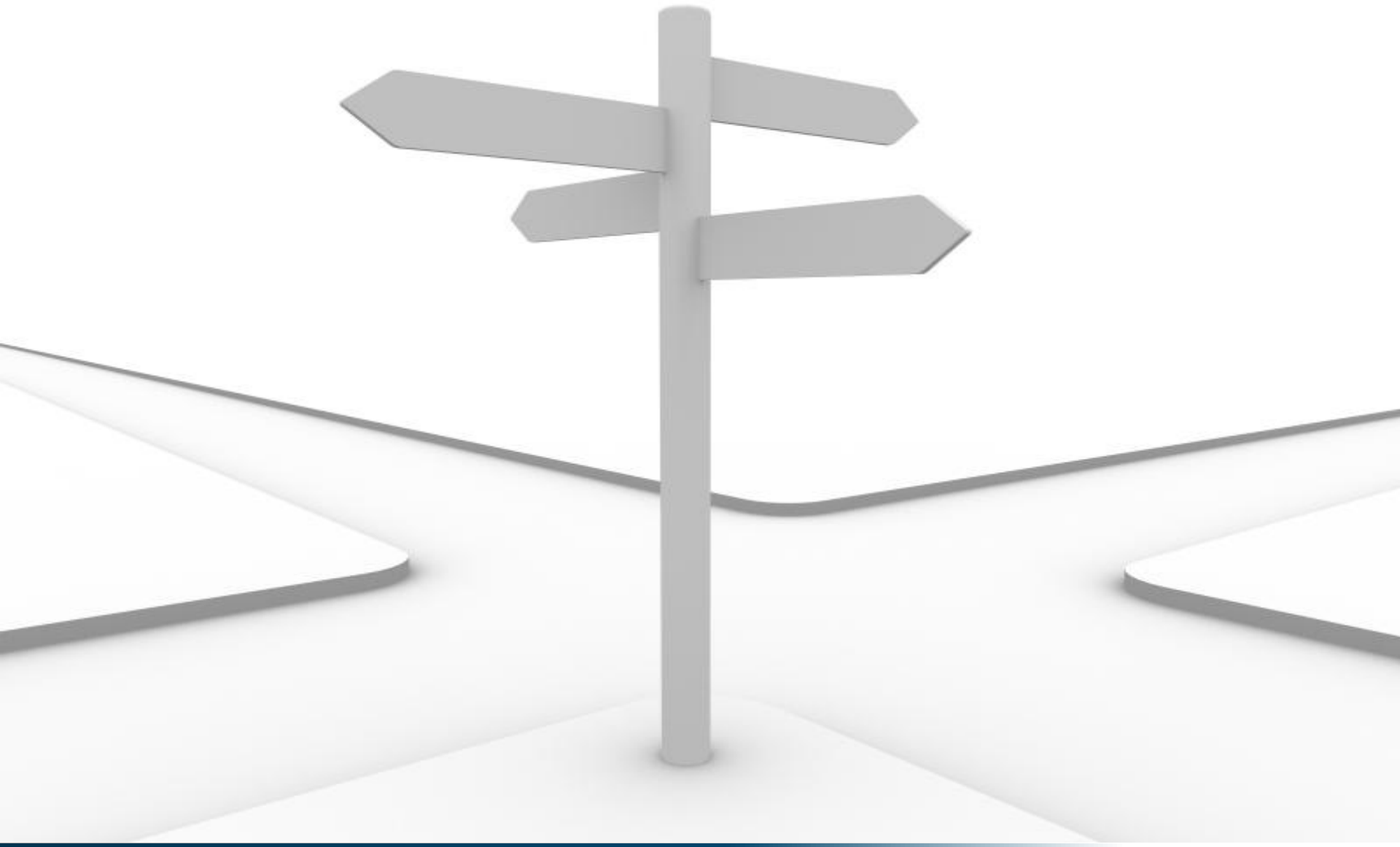
How to Save the Day… (cont'd)

4. **Communicate the Plan**
   - Who, what, when, where, why (and how)
   - Communicate again (and again) (and again)
5. **Measure Progress**
   - Collect metrics for a specific reason, not simply because you can
   - Use the right KPIs
     (search: magic numbers kpi hp owasp webcast)
6. **Report Results**
   - Agree on what will be reported, when and to whom
   - Be creative with rewards
   - Hold people accountable

How to Save the Day… (a few more tips)

- Put Experienced Developers on the Security Team
- Publish Secure Coding Standards
- Train Developers and Security Teams
- Collaborate on the "Top *n*" Security Issues for <period>
- Obtain C-level Sponsorship / Approval of Your Top *n*
- "Tune" Your Security Testing Product(s) to Support the Identification and Presentation of the Top *n* Security Issues
- Treat *All* Security Issues as You Would Any Other Software Defect (i.e., get the issues into your defect tracking system)

Where are you now?



"There is a difference between knowing the path and walking the path."

– Morpheus, *The Matrix*

Final Thought…

"Ever wonder why so many programmers are so bad at security?

Final Thought…

"Ever wonder why so many programmers are so bad at security? Part of the problem is that most of them don't know they're bad."

– Dr. Brian Chess
Founder & Chief Scientist, HP Fortify
http://blog.fortify.com/

(There is an easy fix for this. Really!)

Questions

b c j e n k i n s @ h p . c o m