Highlights of Cybersecurity

Drew Dean
SRI International

How to speak cyber in 30 minutes or less

- These slides will be necessarily incomplete
- Feel free to bring up other topics in discussion
- Please ask questions as they come up
- I agree with almost all of Chapter 4, so I'm not going to brief it directly; please ask questions

Language

- Inside the Beltway, people say "cyber" or "cybersecurity"
- Outside, hardly anyone does: it's computer security, information security, or infosec
- Government terms: CNA, CND, CNE, CNO, IA

Cyber is where the $ is

- Everyone believes doing more of their thing will solve the problem, whatever their thing is:
  - IC
  - DoD
  - Research community
  - Private industry

The Money Quote

There are no intrinsic “laws of nature” for cyber-security as there are, for example, in physics, chemistry or biology. Cyber-security is essentially an applied science that is informed by the mathematical constructs of computer science such as theory of automata, complexity, and mathematical logic.

— *Science of Cyber-Security*, JSR-10-102, http://fas.org/irp/agency/dod/jason/cyber.pdf

In plain English, much of Computer Science, which is a mathematical science, not a physical science (emphasis on proof vs. experimentation)

Most software stinks

- The vast majority of vulnerabilities are caused by poor quality software
- Why?
  - Many reasons, but misaligned economic incentives explain a lot
  - Usability concerns are also real, but more manageable
- Improving software quality is necessary but not sufficient (to use a term of art from mathematics)

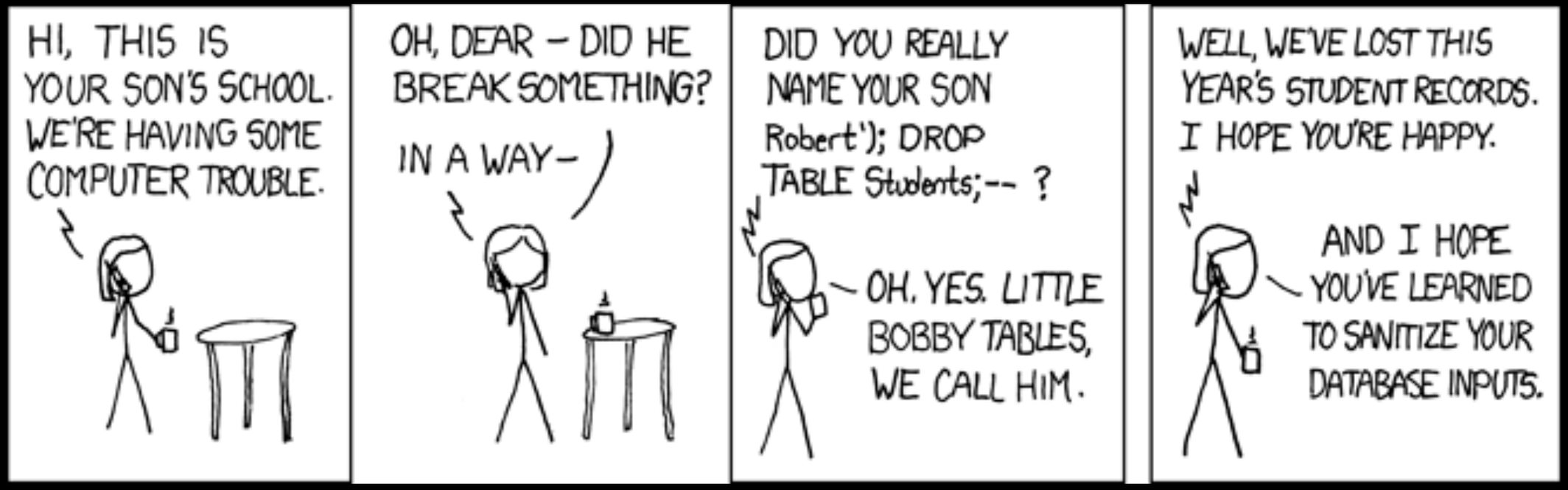
Defining software security

- Many definitions, but let's keep it simple
- Secure software does only what it's supposed to do, *and nothing else*
- The nothing else is the hard part
- I've never seen a requirements document that says "insert security vulnerability here"

Many issues but few causes

- Almost all vulnerabilities fall into one or more of the following categories
  - Memory corruption
  - Confusion of code & data
  - Poor or non-existent cryptography
  - Unexpected feature interactions
  - Lack of input validation
  - Missing authorization checks

Internet meme: Bobby Tables from XKCD



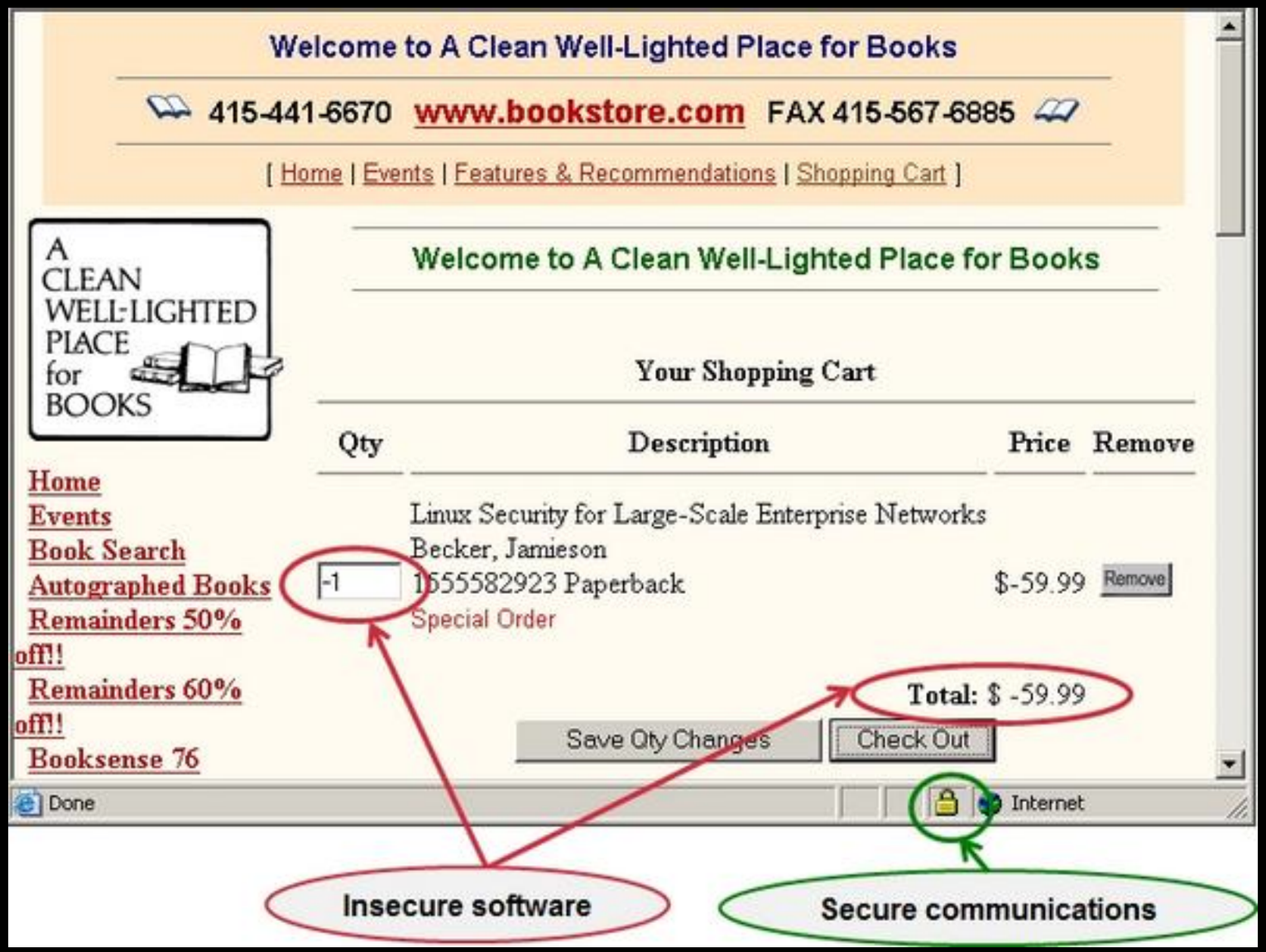
http://xkcd.com/327/

The canonical example of "SQL injection"

What went wrong

- Program built a database query by concatenating strings
- Special characters (in this case, the single quote) have meaning (causing code/data confusion)
- Inadequate input validation let special characters through

What if?



https://twitter.com/ericbaize/status/492777221225213952

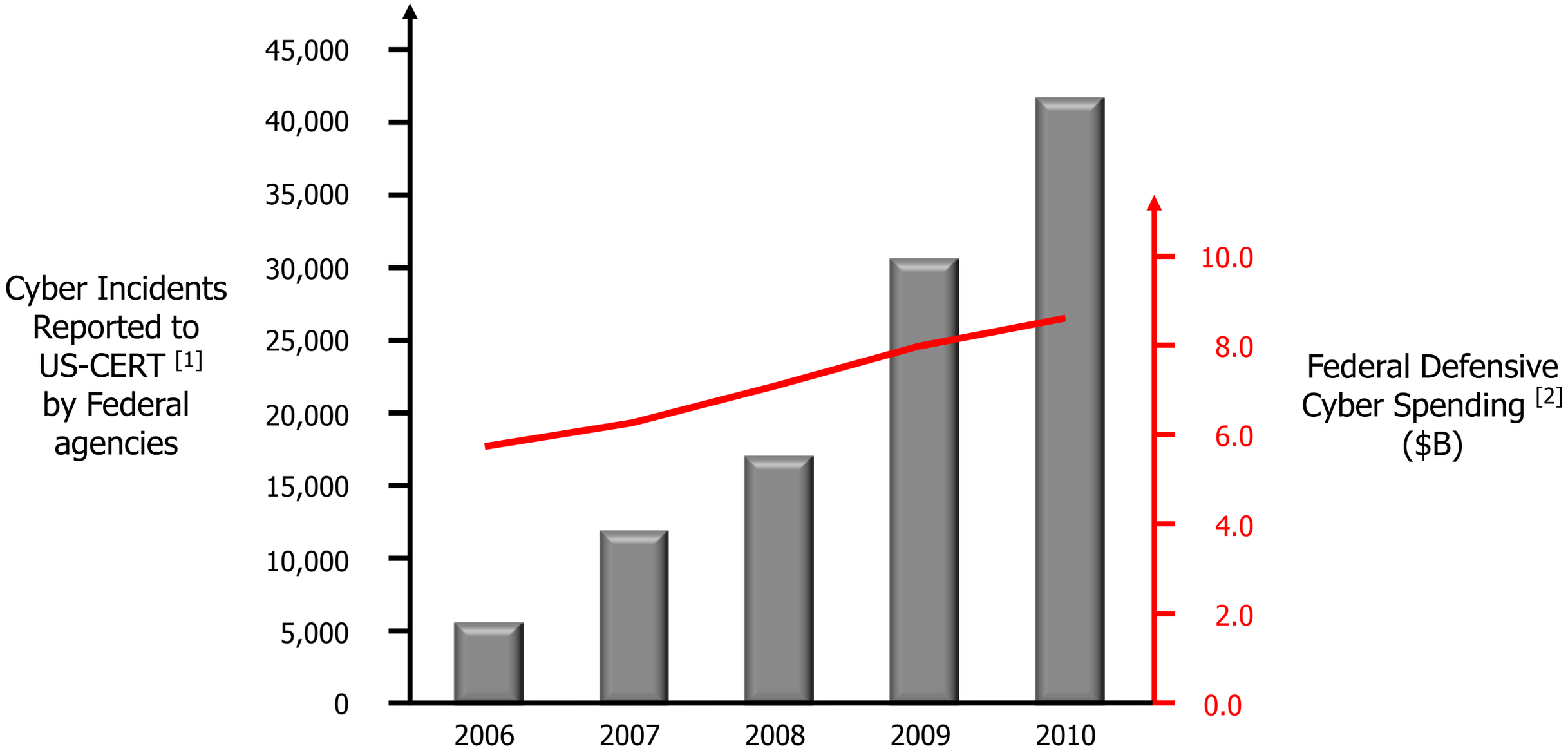
What went wrong

- Inadequate (if not entirely missing) input validation on quantity: no one ever considered a negative quantity
- At one point, Amazon had the same problem: http://www.businessinsider.com/when-amazon-launched-a-bug-allowed-users-to-get-paid-by-the-company-2011-10

Why?

- I'm going to steal some slides from DARPA's Cyber Analytic Framework that I helped produce
- It's from 2010-2011, so slightly dated, but still highly relevant
- See http://www.darpa.mil/Cyber_Colloquium_Presentations.aspx for more

Ground truth…

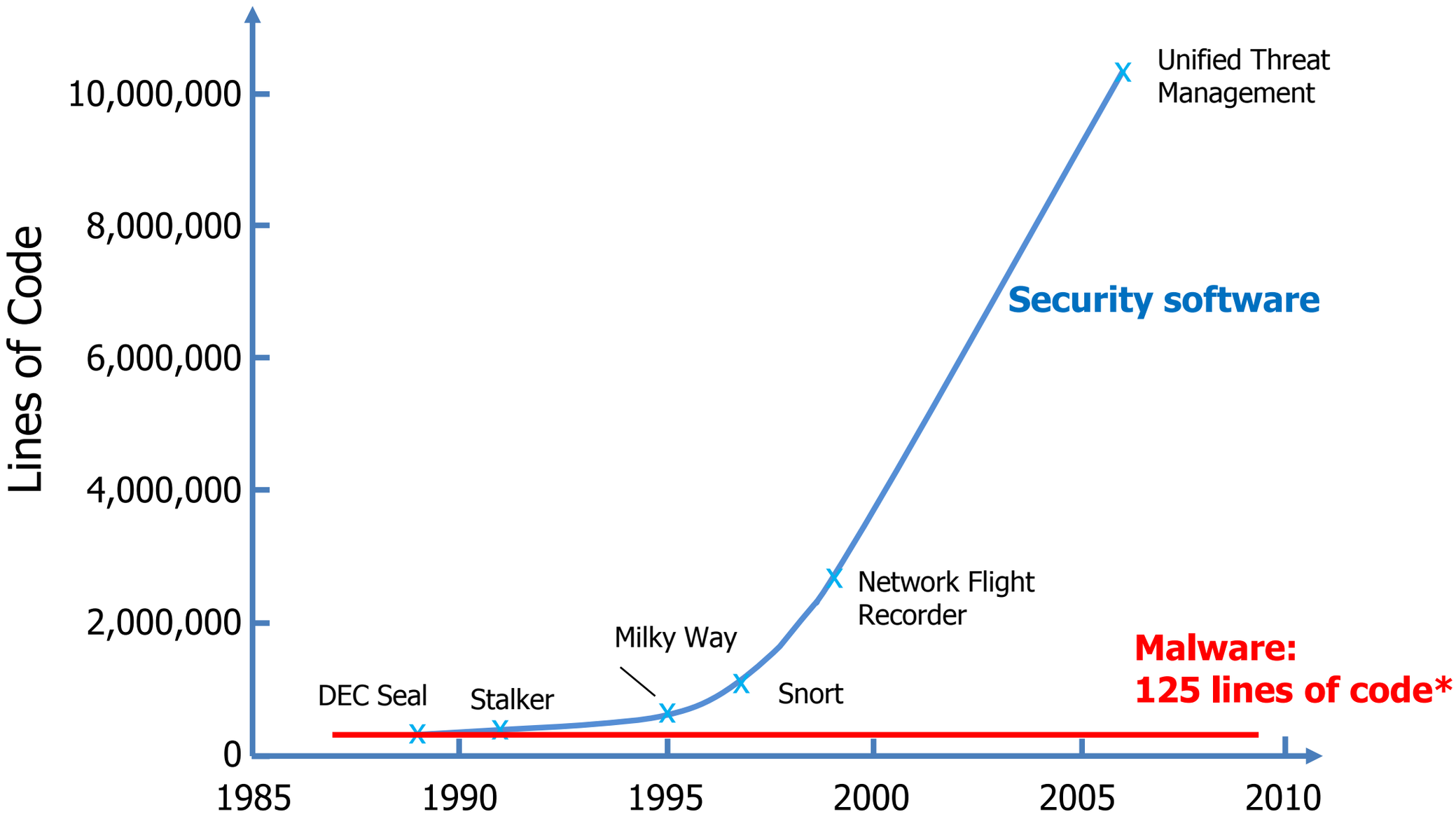


Federal Cyber Incidents and Defensive Cyber Spending
fiscal years 2006 – 2010

[1] GAO analysis of US-CERT data.
GAO-12-137 Information Security: Weaknesses Continue Amid New Federal Efforts to Implement Requirements

[2] INPUT reports 2006 – 2010

We are divergent with the threat…



* Public sources of malware averaged over 9,000 samples
(collection of exploits, worms, botnets, viruses, DoS tools)

Additional security layers often create vulnerabilities…

October 2010 vulnerability watchlist

Vulnerability Title	Fix Avail?	Date Added
XXXXXXXXXXXX XXXXXXXXXXXX Local Privilege Escalation Vulnerability	No	8/25/2010
XXXXXXXXXXXX XXXXXXXXXXXX Denial of Service Vulnerability	Yes	8/24/2010
XXXXXXXXXXXX XXXXXXXXXXXX Buffer Overflow Vulnerability	No	8/20/2010
XXXXXXXXXXXX XXXXXXXXXXXX Sanitization Bypass Weakness	No	8/18/2010
XXXXXXXXXXXX XXXXXXXXXXXX Security Bypass Vulnerability	No	8/17/2010
XXXXXXXXXXXX XXXXXXXXXXXX Multiple Security Vulnerabilities	Yes	8/16/2010
XXXXXXXXXXXX XXXXXXXXXXXX Remote Code Execution Vulnerability	No	8/16/2010
XXXXXXXXXXXX XXXXXXXXXXXX Use-After-Free Memory Corruption Vulnerability	No	8/12/2010
XXXXXXXXXXXX XXXXXXXXXXXX Remote Code Execution Vulnerability	No	8/10/2010
XXXXXXXXXXXX XXXXXXXXXXXX Multiple Buffer Overflow Vulnerabilities	No	8
XXXXXXXXXXXX XXXXXXXXXXXX Stack Buffer Overflow Vulnerability	Yes	8
XXXXXXXXXXXX XXXXXXXXXXXX Security-Bypass Vulnerability	No	8
XXXXXXXXXXXX XXXXXXXXXXXX Multiple Security Vulnerabilities	No	8
XXXXXXXXXXXX XXXXXXXXXXXX Buffer Overflow Vulnerability	No	7/29/2010
XXXXXXXXXXXX XXXXXXXXXXXX Remote Privilege Escalation Vulnerability	No	7/28/2010
XXXXXXXXXXXX XXXXXXXXXXXX Cross Site Request Forgery Vulnerability	No	7/26/2010
XXXXXXXXXXXX XXXXXXXXXXXX Multiple Denial Of Service Vulnerabilities	No	7/22/2010

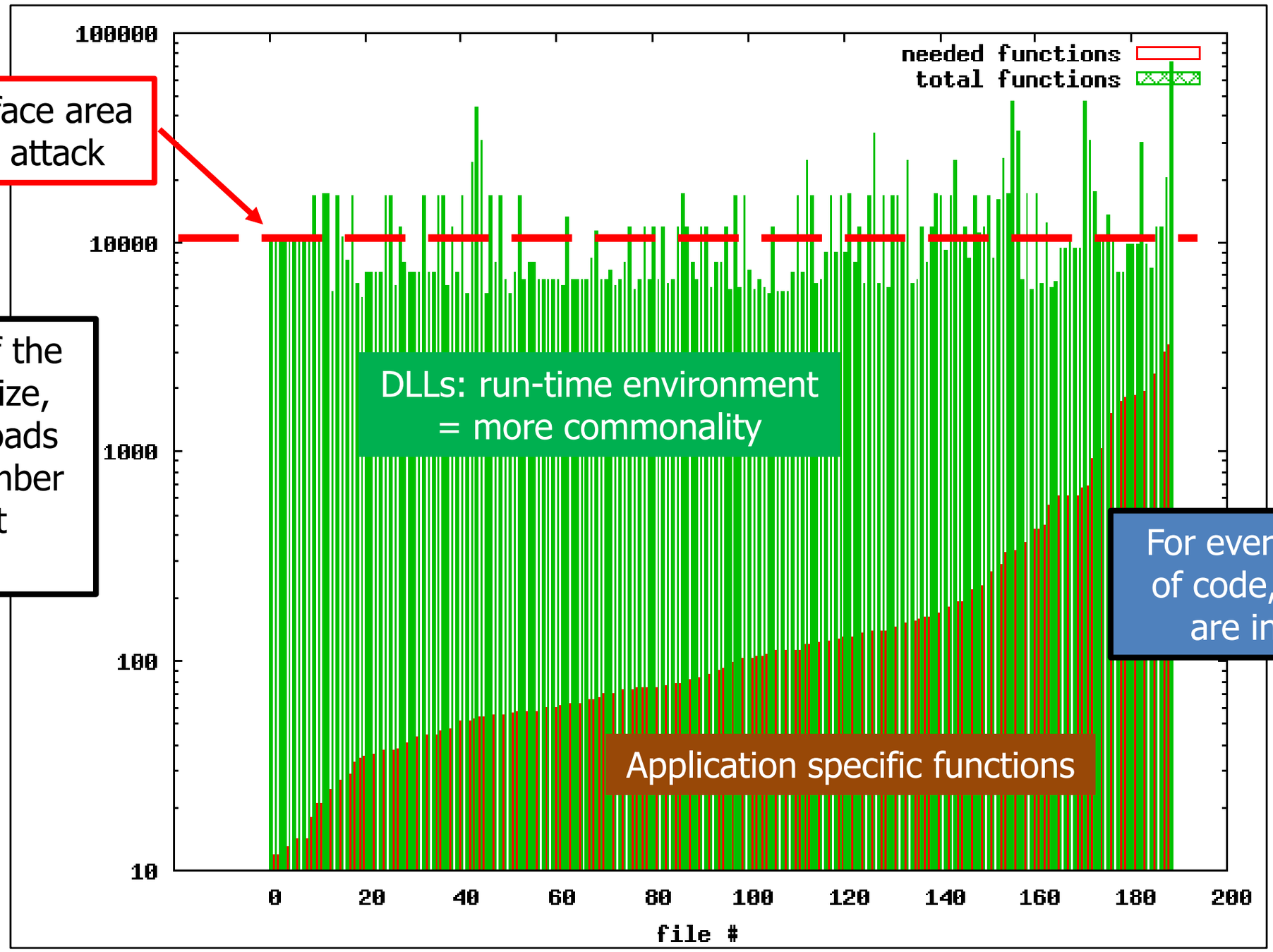
6 of the vulnerabilities are in security software

Color Code Key: Vendor Replied – Fix in development | Awaiting Vendor Reply/Confirmation | Awaiting CC/S/A use validation

These layers increase the attack surface…

Constant surface area available to attack

Regardless of the application size, the system loads the same number of support functions.

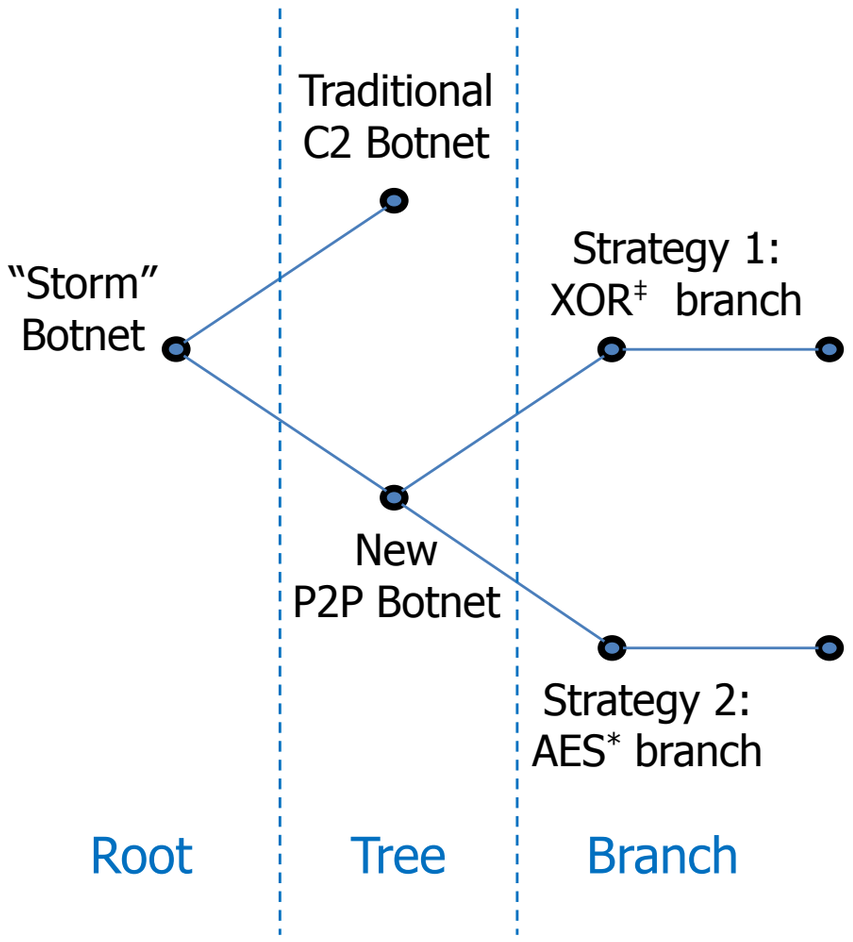


For every 1,000 lines of code, 1 to 5 bugs are introduced.

Business incentives matter…

Understanding them in the context of 'game theory' reveals the problem.

Bot Herder strategy example:



Bot Herder Cost	Bot Herder Return		Antivirus Cost	Antivirus Return
	Short	Long		
Small	High	High	Low	High
Small	High	0	High	Low

The security layering strategy and antitrust has created cross incentives that contribute to divergence.

‡ = "exclusive or" logical operation

∗ = Advanced Encryption Standard

Cost of being late to market with software

- Direct costs are surprisingly large: 3–6 month delay can cost 20%-40% of total lifetime profit
  - See https://www.initialstate.com/LateCalc
- Also loss of first mover advantage, etc.
- So time is even more important than money to improve software quality and hence security

Security costs are externalized

- … at least until they get too large (c.f. Microsoft)
- Removing security from OS vendor is bad policy
  - See Windows Vista, Storm botnet example above
- Current poster child / whipping boy: Adobe
  - It seems like every other time I need to use Flash, it needs updating to patch a vulnerability…
  - Same is true of Acrobat Reader

So…

- Current software quality is economically rational
- Unclear how to change the large scale incentive structure
- Probably some combination of market forces, regulation, and liability

On security metrics…

- Metrics for cybersecurity remain a really hard problem
- They've been an open problem for a long time
- Discontinuity (nearly trivial difference being the difference between being secure and totally insecure) tends to defy human intuition
- Lack of metrics contributes to information asymmetry, a classic contributor to poor decision making

Thank you

- Questions?