IS 2620

Developing Secure Systems

Introduction

Aug 27, 2014

James Joshi,

Associate Professor

Contact

- James Joshi
- 706A, IS Building
- Phone: 412-624-9982
- E-mail: jjoshi@mail.sis.pitt.edu
- Web: http://www.sis.pitt.edu/~jjoshi/courses/IS2620/Fall14/
  Office Hours: By appointments
- GSA: Lei Jin

Course Objectives

- To learn about how to design/implement secure and high assurance information systems
  - Understand and analyze code for vulnerabilities
  - Secure programming (e.g., C, C++, Java)
  - Secure architectures & security assurance
- Understand the principles and practice towards designing secure information systems
  - Life cycle models/ security engineering principles
  - Usability issues
- To learn about the tools and techniques towards assurance (validation/verification/testing)
  - Use of tools to detect coding/design flaws;
  - architectural risk analysis

Course Coverage

- Secure programming
  - Coding practices, issues and guidelines
    - Code analysis;
      - Buffer overflows Race conditions
      - Input validation SQL injection
      - Cross-site scripting Mobile Code Safe Languages
- Secure software development & Assurance process
  - Security Engineering/Lifecycle models
    - E.g. Capability Maturity Models and Extensions Building security In
  - Secure Design/Implementation Principles
    - Systems / software &Formal methods and testing
      - UMLSec, Model Checking (code, protocols)
- Secure Supply Chain environments
- Verification / model checking
- Reverse engineering
- Trusted computing modules/environments

Pre-requisite

- IS 2150/TEL 2810 Information Security & Privacy
    - OR background in security
- Following courses are preferred but not required:
    - IS 2170/TEL 2820 Cryptography; TEL 2821 Network Security
- Talk to me if you are not sure of the background
- Course Reference: Check website

Grading (Tentative)

- Assignments/Presentation/Exam: 50%
  - Read/Review and/or present research papers or articles
  - Assignments/quizzes
  - Lab exercises
- Exams and Project : 50%
  - Two exams
  - One project

Course Policy

- Your work MUST be your own
  - Zero tolerance for cheating/plagiarism
  - You get an F for the course if you cheat in anything however small – NO DISCUSSION
  - Discussing the problem is encouraged

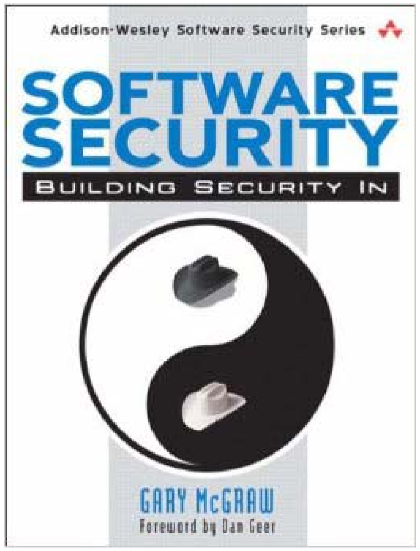
- Homework
  - Penalty for late assignments (15% each day)
  - Ensure clarity in your answers – no credit will be given for vague answers
  - Homework is primarily the GSA’s responsibility

- Check webpage for everything!
  - You are responsible for checking the webpage for updates

Why Secure Software/System Development?

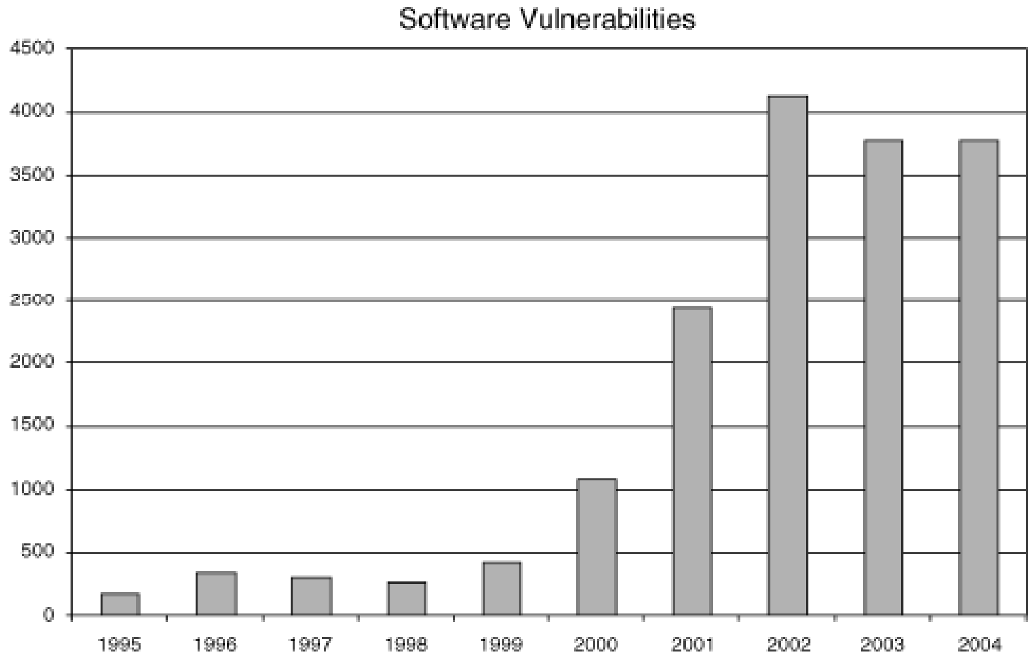
Software/Systems Security

- Renewed ---- interest & importance
  - “*idea of engineering software so that it continues to function correctly under malicious attack*”
  - Existing software is riddled with design flaws and implementation bugs
    - ~70% related to design flaws*
  - “any program, no matter how innocuous it seems, can harbor security holes”



***http://www.securitymanagement.com/archive/library/atstake_tech0502.pdf**

Software Problem

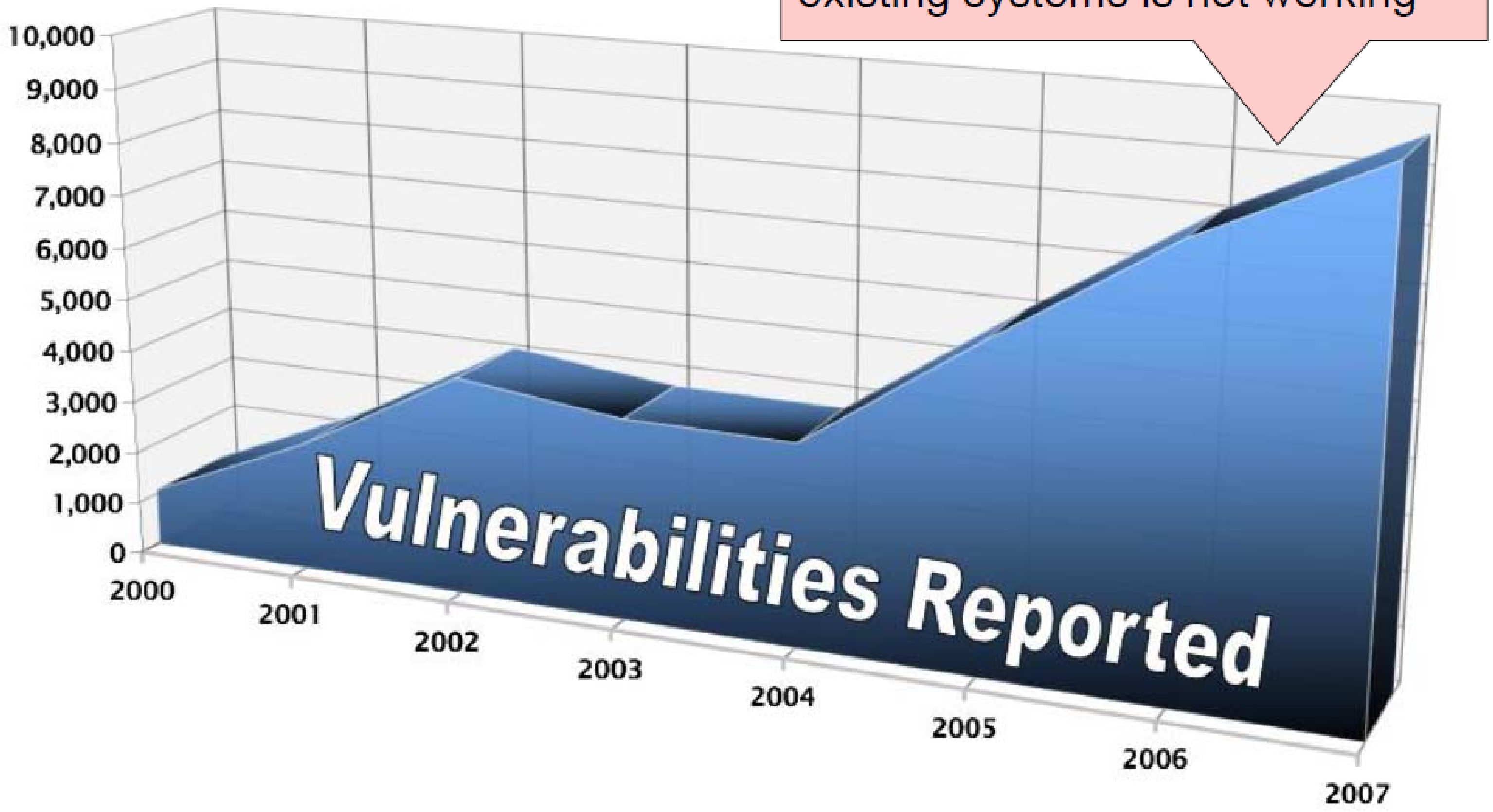


# vulnerabilities Reported by CERT/CC

- More than half of the vulnerabilities are due to buffer overruns
- Others such as race conditions, design flaws are equally prevalent

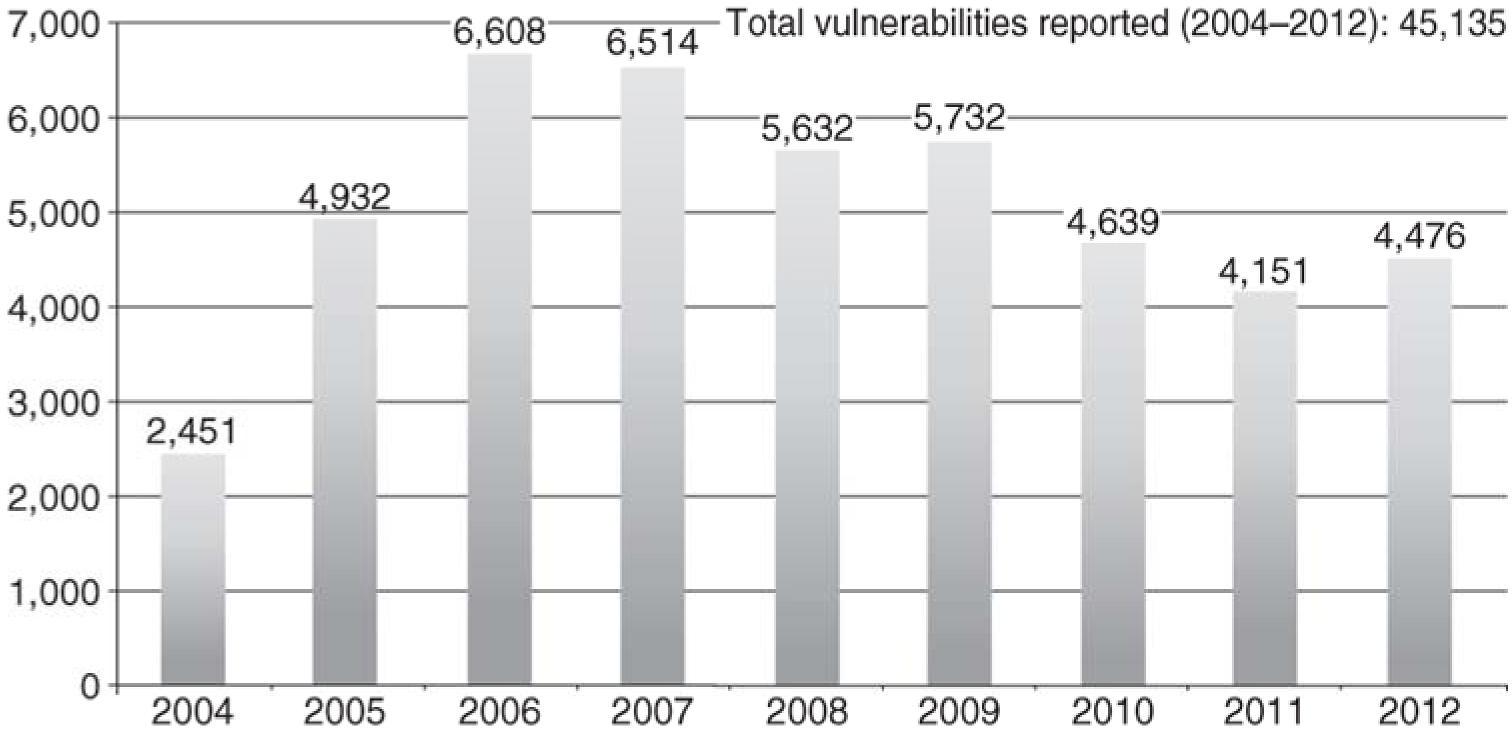
CERT Vulnerability

Reacting to vulnerabilities in existing systems is not working



Source: Seacord's Webinar on Secure Coding on C and C++

NVD statistics



Software security

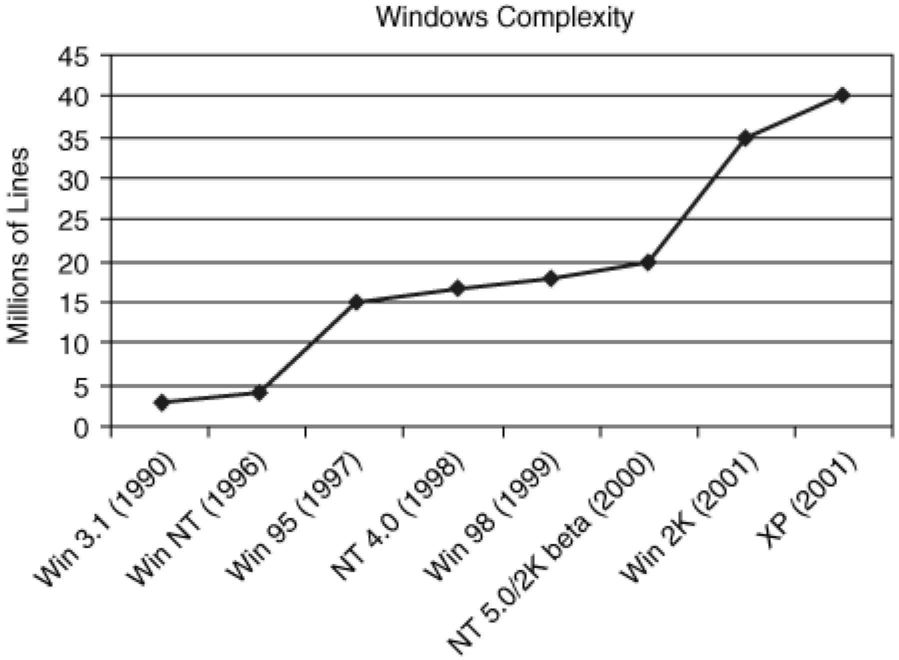
- It is about
  - Understanding software-induced security risks and how to manage them
  - Leveraging software engineering practice,
  - thinking security early in the software lifecyle
  - Knowing and understanding common problems
  - Designing for security
  - Subjecting all software artifacts to thorough objective risk analyses and testing
- It is a knowledge intensive field

Trinity of trouble

- Three trends
  - Connectivity
    - Inter networked
    - Include SCADA (supervisory control and data acquisition systems)
    - Automated attacks, botnets
  - Extensibility
    - Mobile code – functionality evolves incrementally
    - Web/Os Extensibility
  - Complexity
    - XP is at least 40 M lines of code
    - Add to that use of unsafe languages (C/C++)

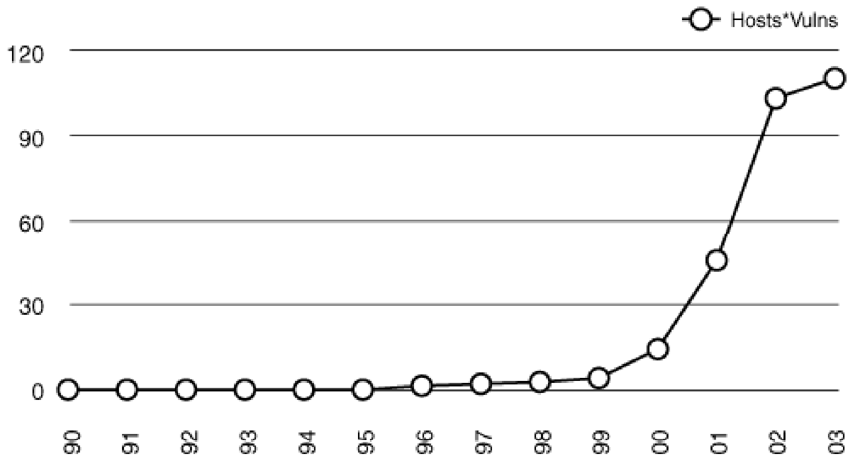
Bigger problem today

.. And growing



INFOGRAPHICS Link:
http://h.fastcompany.net/multisite_files/fastcompany/imagecache/inline-large/inline/2013/11/3021256-inline-800linesofcode5.jpg

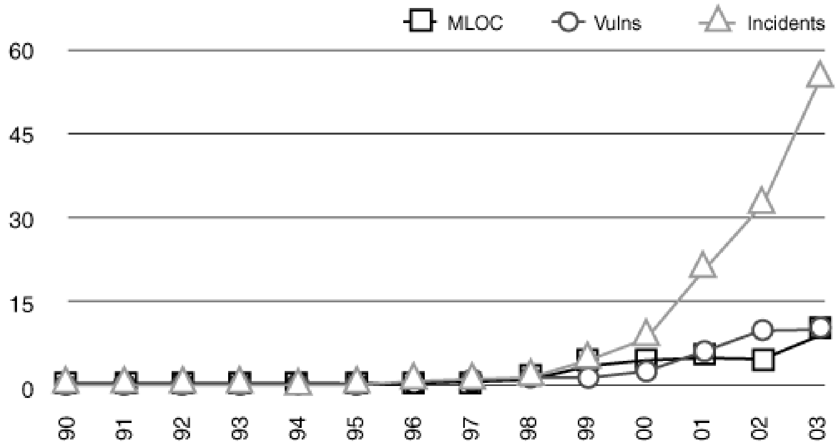
It boils down to …

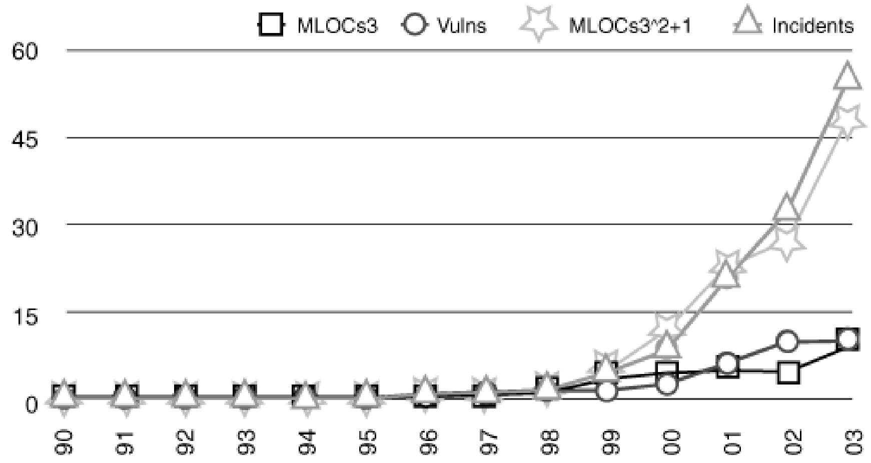

Opportunity (normalized)

more code,

more bugs,

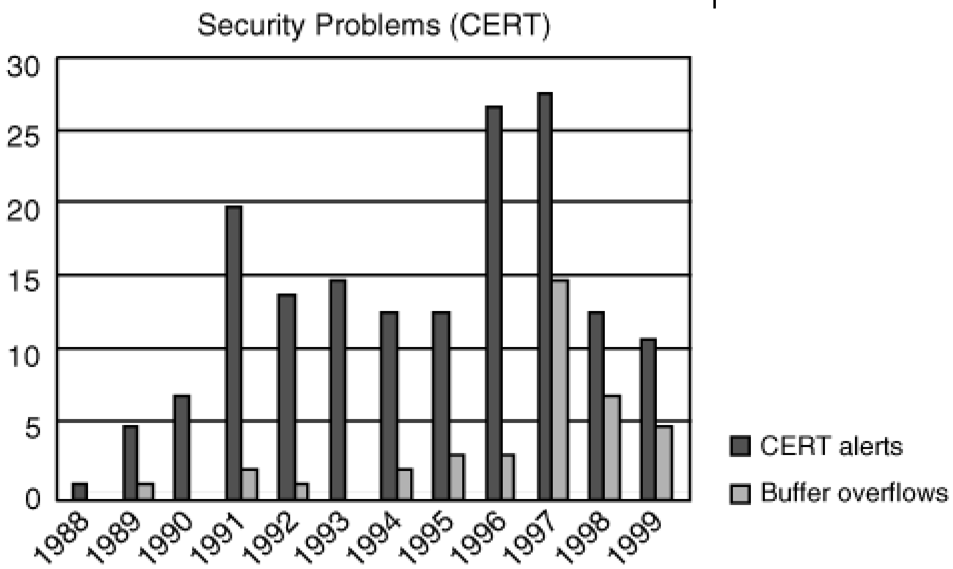
more security problems


Normalized (median, 2-year lag)


Drivers

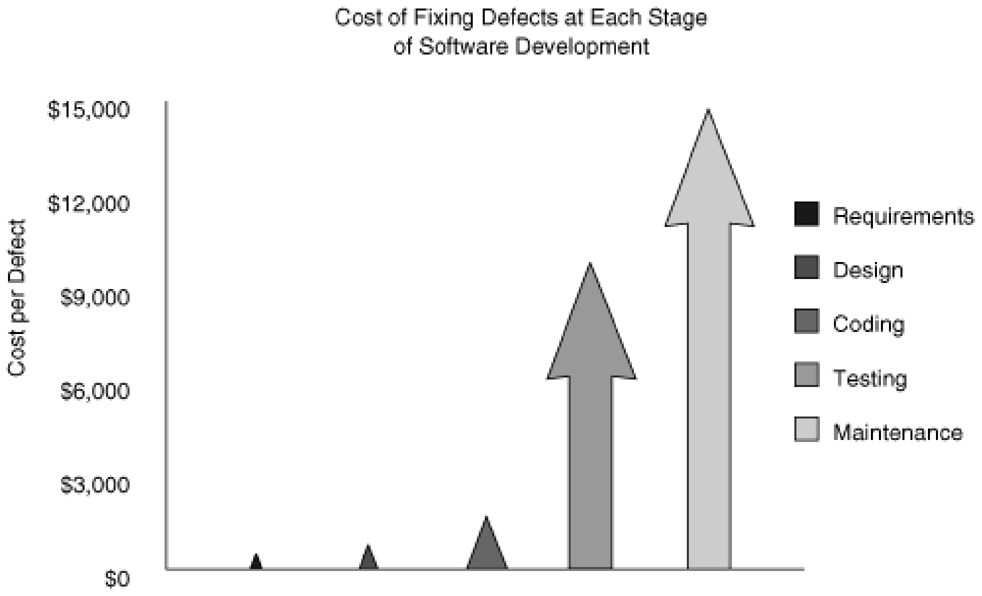
Security problems in software

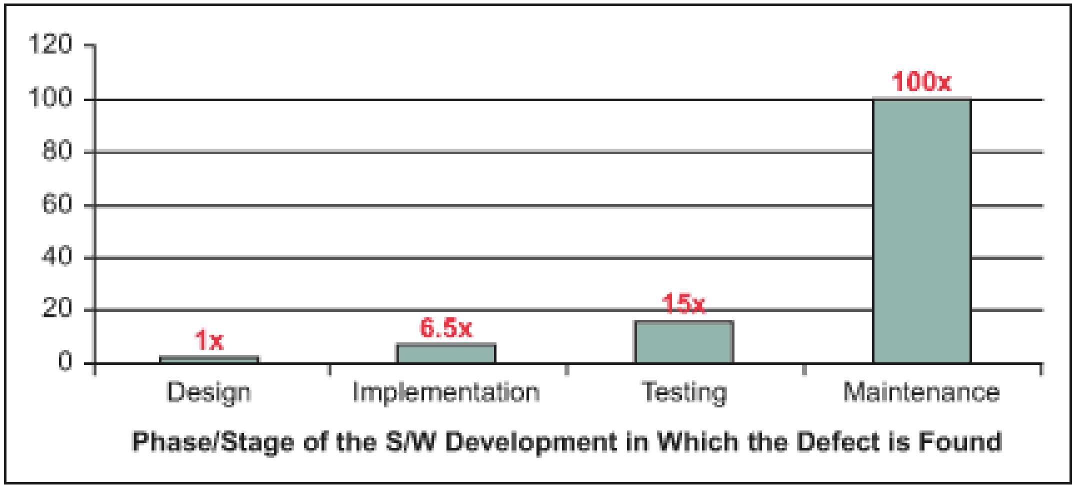
- Defect
  - implementation and design vulnerabilities
  - Can remain dormant
- Bug
  - An implementation level software problem
- Flaw
  - A problem at a deeper level
- Bugs + Flaws
  - leads to Risk



Bug	Flaw
Buffer overflow: stack smashing Buffer overflow: one-stage attacks Buffer overflow: string format attacks Race conditions: TOCTOU Unsafe environment variables Unsafe system calls (fork(), exec(), system()) Incorrect input validation (black list vs. white list	Method over-riding problems (subclass issues) Compartmentalization problems in design Privileged block protection failure (DoPrivilege()) Error-handling problems (fails open) Type safety confusion error Insecure audit log design Broken or illogical access control (role-based access control [RBAC] over tiers) Signing too much code

Cost of fixing





Relative Costs to Fix Software Defects (Source: IBM Systems Sciences Institute)

OWASP Top Ten Vulnerabilities (for 2013)

- A1-Injection
  - SQL, OS, LDAP – input validation problem
- A2-Broken Authentication and Session Management
  - Incorrect implementation (compromise passwords, keys, implementation flaws
- A3-Cross-Site Scripting (XSS)
  - Improper validation
- A4-Insecure Direct Object References
  - Improper exposure of internal implementation
- A5-Security Misconfiguration
- A6-Sensitive Data Exposure

OWASP
The Open Web Application Security Project

OWASP Top Ten Vulnerabilities (for 2013)

- A7-Missing Function Level Access Control
  - Web applications UI and server need to enforce consistent access control enforcement
- A8-Cross-Site Request Forgery (CSRF)
  - Forged HTTP requests and compromise of victim's session cookie
  - Victim's browser is forced to generate requests to the vulnerable application
- A9-Using Components with Known Vulnerabilities
  - Components could run with full privileges – vulnerable program could be exploited
  - Components could be libraries or software modules and frameworks
- A10-Unvalidated Redirects and Forwards
  - Improper validation issue
  - Web apps can redirect victims to phishing or malware sites.

Comparison: **http://www.port80software.com/support/articles/2013-owasp-top-10**

Recent incidents ..

- HeartBleed (CVE-2014-0160)
  - A serious threat in OpenSSL
  - Estimated to have made 2/3 of Internet vulnerable
  - Essentially a buffer overflow issue (overreads)
  - Improper input validation – allows access to more data
    - Automated software testing did not catch !!
    - Static analysis did not catch it ! And dynamic/hybrid not designed for such vulnerability
  - Some approaches that would have helped
    - Negative testing/Fuzzing with special checks
    - Better Source code analysis; safer language (it was in C)
    - Formal methods

Source: "Preventing Heartbleed" by David Wheeler, IEEE Computer
Also Check out: http://www.kb.cert.org/vuls/id/720951

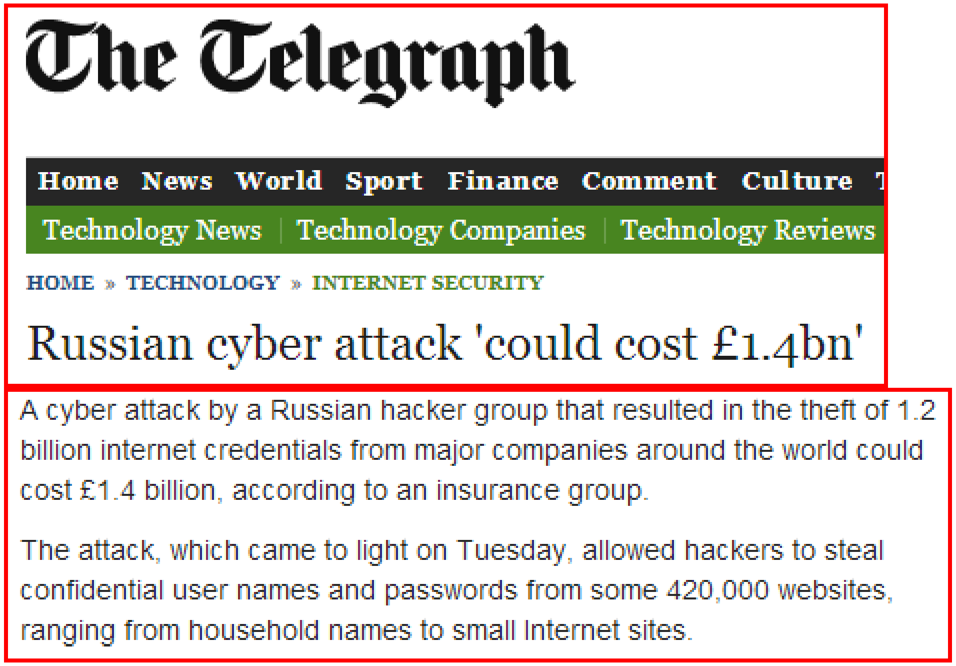
Recent incidents ..

- Stuxnet
  - Affected several ICSs; Includes
    - exploit of the LNK files – shortcut file in windows as a start (other exploits possible)
    - exploit some unpatched version of Win XP
- Target data breach*
    - Financial and personal info af ~110M customers
    - Payment card system flaw – malware installed in POS terminals (RAM Scraping attack)
    - Network access from third party (PA HVAC) which was weak in security – allowed to gain foothold in Target's network

***http://docs.ismgcorp.com/files/external/Target_Kill_Chain_Analysis_FINAL.pdf**

Recent incidents ..

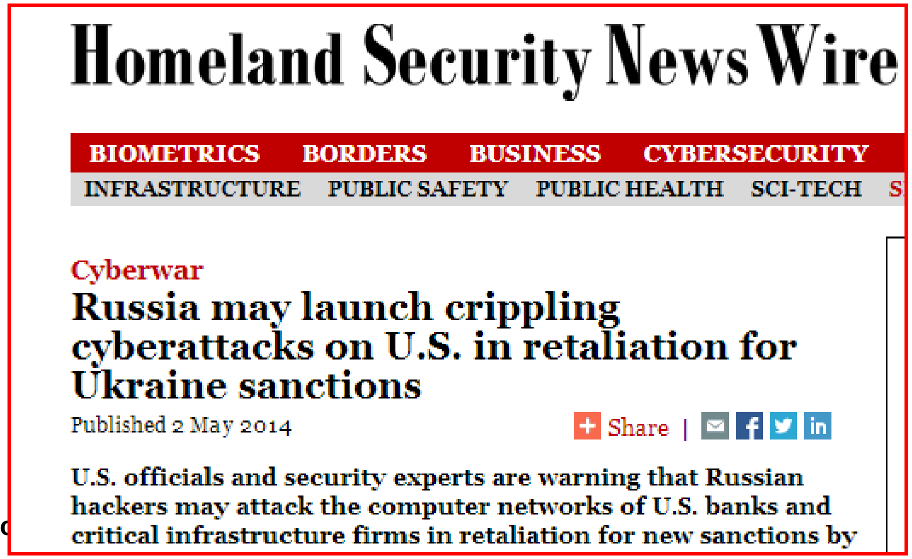
- Russian hackers
  - Targets: Oil, Gas, Energy security – industrial espionage
  - Also target seizing control of ICS


The Telegraph
Home News World Sport Finance Comment Culture T
Technology News | Technology Companies | Technology Reviews
HOME » TECHNOLOGY » INTERNET SECURITY

Russian cyber attack 'could cost £1.4bn'

A cyber attack by a Russian hacker group that resulted in the theft of 1.2 billion internet credentials from major companies around the world could cost £1.4 billion, according to an insurance group.

The attack, which came to light on Tuesday, allowed hackers to steal confidential user names and passwords from some 420,000 websites, ranging from household names to small Internet sites.


Homeland Security News Wire
BIOMETRICS BORDERS BUSINESS CYBERSECURITY
INFRASTRUCTURE PUBLIC SAFETY PUBLIC HEALTH SCI-TECH S
Share |

Cyberwar

Russia may launch crippling cyberattacks on U.S. in retaliation for Ukraine sanctions

Published 2 May 2014

U.S. officials and security experts are warning that Russian hackers may attack the computer networks of U.S. banks and critical infrastructure firms in retaliation for new sanctions by

http://www.nytimes.com/2014/07/01/technology/energy-sector-fac

Hence we need …

- Robust and Secure Software Design and Secure Systems Engineering practice
  - Secure development life-cycle/methodologies
  - Secure process models to support large scale team management
  - Fix flaw early in the life-cycle – LOW COST !!
- Secure Design principles
- Proper Testing and Verification/Validation
- Effective Tools and Techniques
- Security Engineering education
- Etc..