CSc 131 – Computer Software Engineering

Security and software engineering

Acknowledgements

- Slides from Susan J Lincke, PhD , Univ. of Wisconsin-Parkside
- Slides from Ninghui Li, CSC 526 course – Information Security and Assurance, Purdue Univ
- Microsoft – Security Development Lifecycle (SDL)

  http://www.microsoft.com/en-us/sdl/
- Slides from Dimitry Averin, CS996 – Information Security Management, Polytechnic Institute of New York University

Security and software engineering

- Part 1: Security, security awareness, terminologies, and some background materials
- Part 2: Security Development Lifecycle (SDL)

More In the News

Vodafone says hackers broke into nearly 2,000 customer accounts this week

Russian software virus ' Tyupkin' forces ATMs into maintenance mode and spew cash

Health care orgs fall short on software security

Don’t Ignore Software Update; May Lead to Hack, Security Flaw

Daily news?

Why Do Computer Attacks Occur?

Who are the attackers?

criminals, organized crime organizations, rogue states, industrial espionage, angry employees, …

Why they do it?

fun,

fame,

profit, …

computer systems are where the moneys are

Computer Security Issues

Computer viruses

- Virus - code that copies itself into other programs.
- A “Bacteria” replicates until it fills all disk space, or CPU cycles.

Trojan horses

- Instructions in an otherwise good program that cause bad things to happen (sending your data or password to an attacker over the net).

Computer worms

- A program that replicates itself across the network (usually riding on email messages or attached documents (e.g., macro viruses).

Distributed denial of service attacks

Computer break-ins

Email spams

- E.g., Nigerian scam, stock recommendations

More Computer Security Issues

Identity theft

Zero-day attacks

Botnets

Serious security flaws in many important systems
    electronic voting machines, ATM systems

Spywares

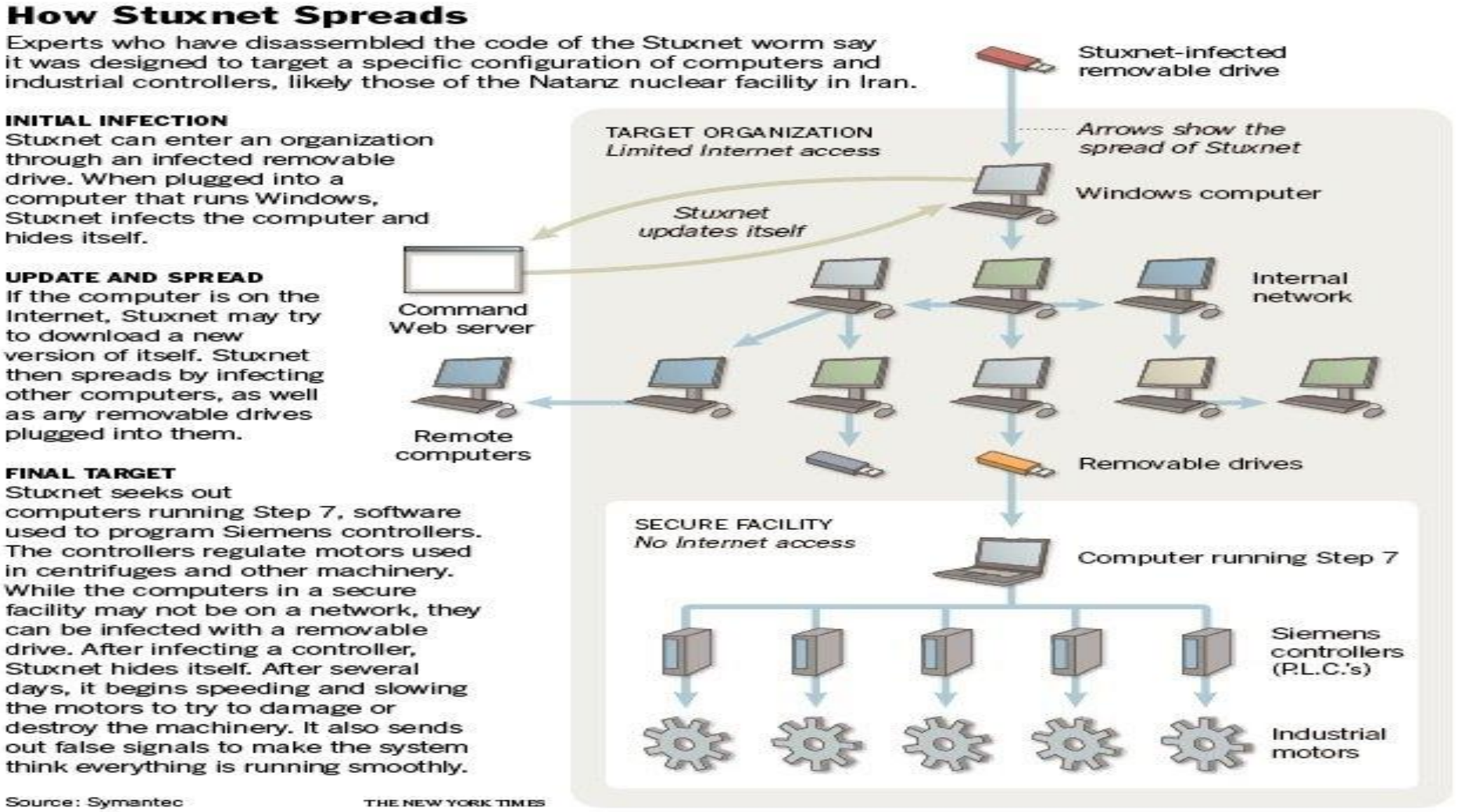
Driveby downloads

Social engineering attacks

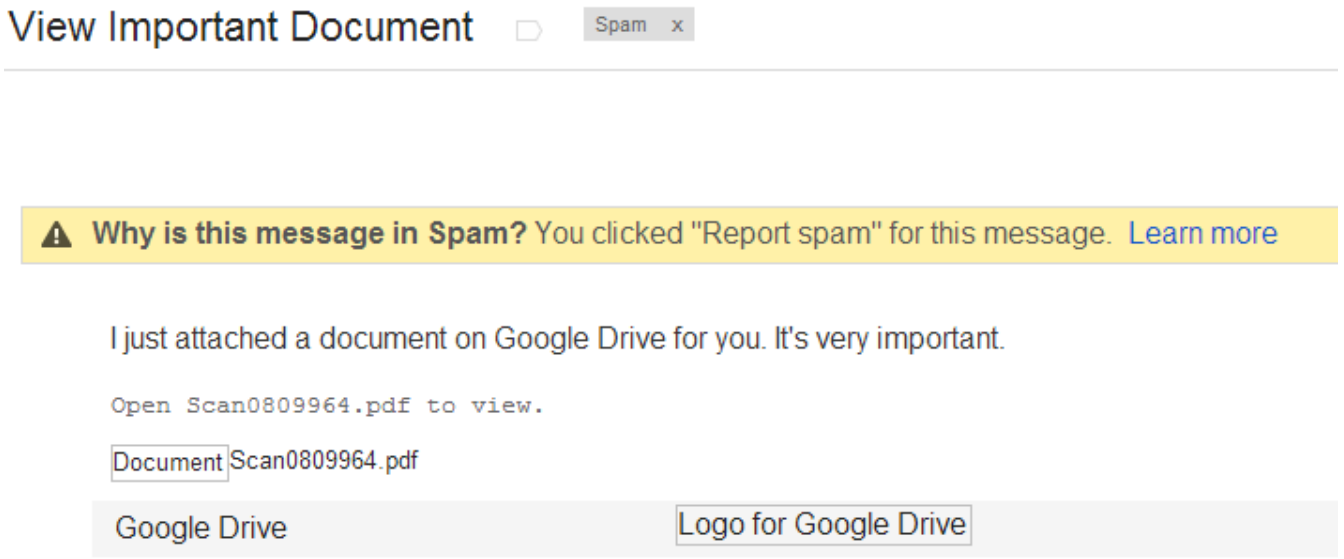
Stuxnet (2010)

- Stuxnet: Windows-based Worm
  - Worm: self-propagating malicious software (malware)
- Attack Siemens software that control industrial control systems (ICS) and these systems
  - Used in factories, chemical plants, and nuclear power plants
- First reported in June 2010, the general public aware of it only in July 2010
- Seems to be a digital weapon created by a nation-state
  - 60% (more than 62 thousand) of infected computers in Iran
  - Iran confirmed that nuclear program damaged by Stuxnet
  - Sophisticated design, special targets, expensive to develop

Stuxnet Virus



Email Spoofing



Why do these attacks happen?

Software/computer systems are buggy

Users make mistakes

Technological factors

- Von Neumann architecture: stored programs
- Unsafe program languages
- Software are complex, dynamic, and increasingly so making things secure are hard
- Security may make things harder to use

Why does this happen?

Economical factors

- Lack of incentives for secure software
- Security is difficult, expensive and takes time

Human factors

- Lack of security training for software engineers
- Largely uneducated population

Security is not Absolute

Is your car secure?

What does “secure” mean?

Are you secure when you drive your car?

Security is relative

to the kinds of loss one consider

security objectives/properties need to be stated

to the threats/adversaries under consideration.

security is always under certain assumptions

Information Security is Challenging

Defense is almost always harder than attack.

In which ways information security is more difficult than physical security?

- adversaries can come from anywhere
- computers enable large-scale automation
- adversaries can be difficult to identify
- adversaries can be difficult to punish
- potential payoff can be much higher

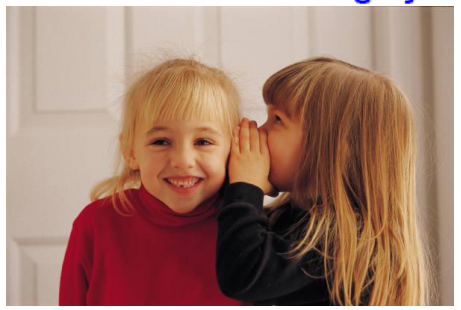
In which ways information security is easier than physical security?

Security Assures … CIA

Confidentiality: Limits access of authorized users and prevents access to unauthorized users

Integrity: The reliability of information resources and data have not been changed inappropriately

Availability: When something needs to be accessed by the user, it is available



Security Vocabulary

Asset: Diamonds

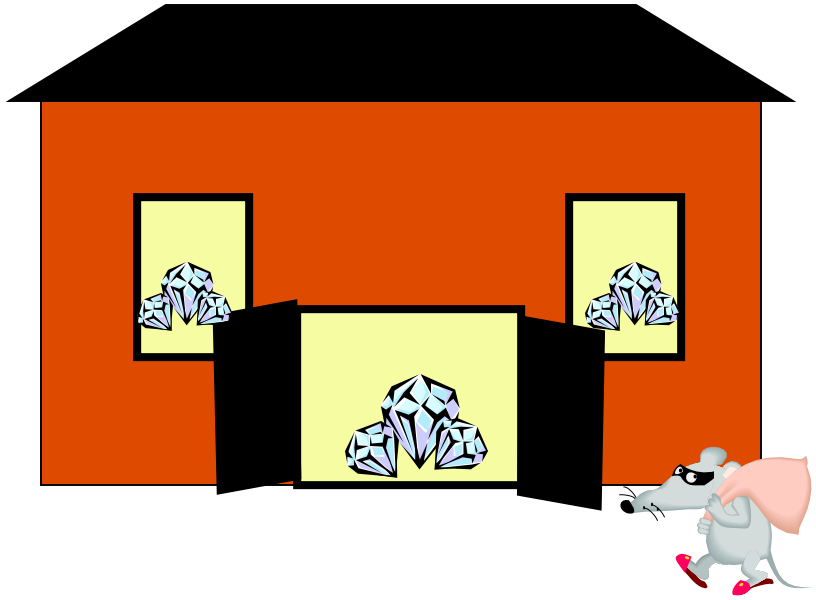
Threat: Theft

Vulnerability: Open door or windows

Threat agent: Burglar

Owner: Those accountable or who value the asset

Risk: Danger to assets



CSc 131 – Computer Software Engineering

Software Engineering & Security

Definitions

Software Engineering: Concept of creating and maintaining software applications by applying technologies and practices from computer science and project management fields [www.wikipedia.org]

Secure Software Engineering

“Current”/Traditional Software Engineering

Over 30 years of software development experience created a well defined application software development lifecycle

- There are many software development methodologies (ex. XP, waterfall, etc) they all have these basic steps
- Capability Maturity Model for Software (SW-CMM), is used to measure quality of methodologies employed

Motivation

This application development process in its essence fails to address security issues

Consequently, security flaws are identified only at the later stages of the application lifecycle. And thus

- Much greater cost to fix
- High maintenance cost
- …

Nearly every company/organization utilizes network security infrastructure (e.g. Firewalls, IDS, etc)

But very small number of them invest in application security strategy, design, and code review services

So

For the software industry, the key to meeting demand for improved security, is to implement repeatable processes that reliably deliver measurably improved security

Thus, there must be a transition to a more stringent software development process that greatly focuses on security

Goal: minimize the number of security vulnerabilities in design, implementation, and documentation

Identify and remove vulnerabilities in the development lifecycle ***as early as possible!!!***

Building Secure Software

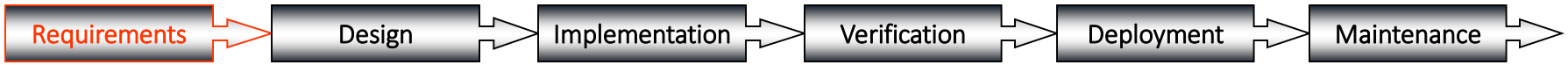
Three essential components

- Repeatable process
- Engineer Education
- Metrics and Accountability

SDL – Secure Development Lifecycle

- Used along with traditional/current software development lifecycle/techniques in order to introduce security at every stage of software development

SDL – Requirements Phase



Development of requirements

- Gather information about application [costumer/experience/survey]

Analysis of requirements

- Are all the security issues addressed
- CIA – [Confidentiality, Integrity, Availability]

Verification of requirements

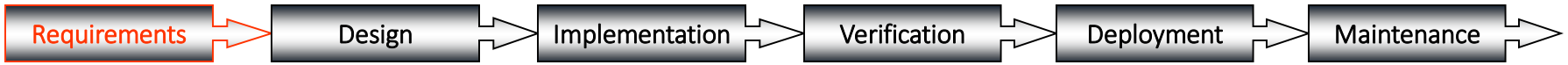
- Are there are any inconsistencies / system interface / correctness
- Documentation!!!

Feasibility of requirements

[repeat]

The bottom line: Planning at this stage offers the best opportunity to build secure software in the most efficient manner [cost, time, etc]

SDL – Requirements Phase



Develop Security Requirements

- Security Requirements of a system/application must be developed along with any other requirements (e.g. functional, legal, user, etc)

Risk analysis

- Identify all the assets at risk
- Identify all the threats

Develop security policies

- Used as guidelines for requirements

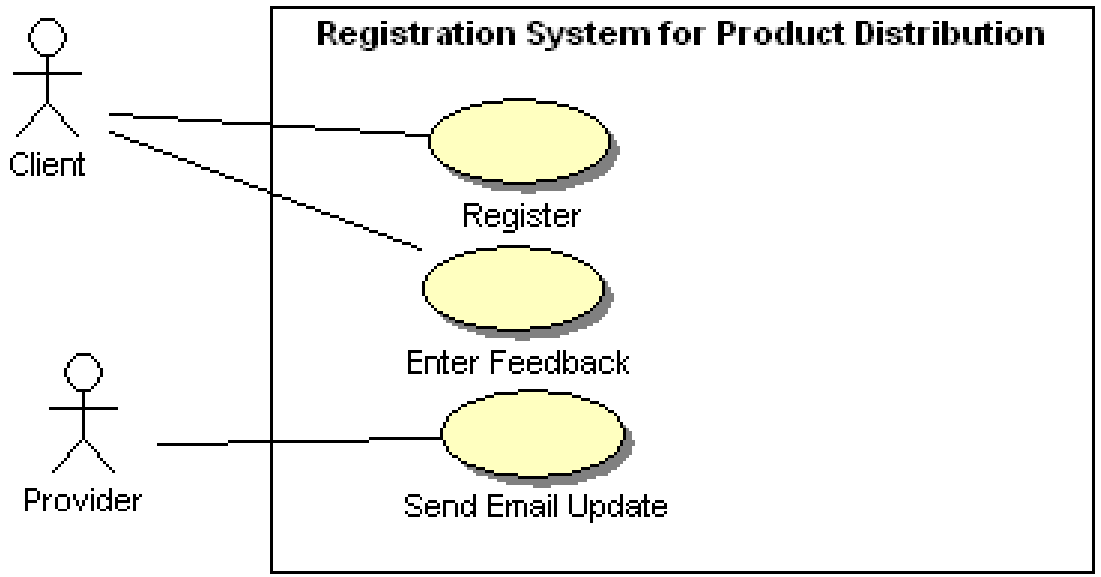
Develop security metrics

- For examples: % weak password, % user training with security, # of policy violations

Example: Registration System Use Case

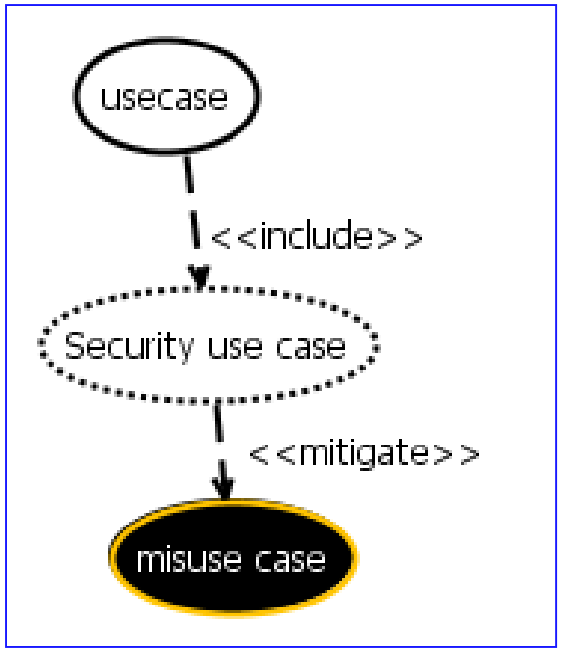
Register: Clients register to obtain documentation by providing name, email, job function

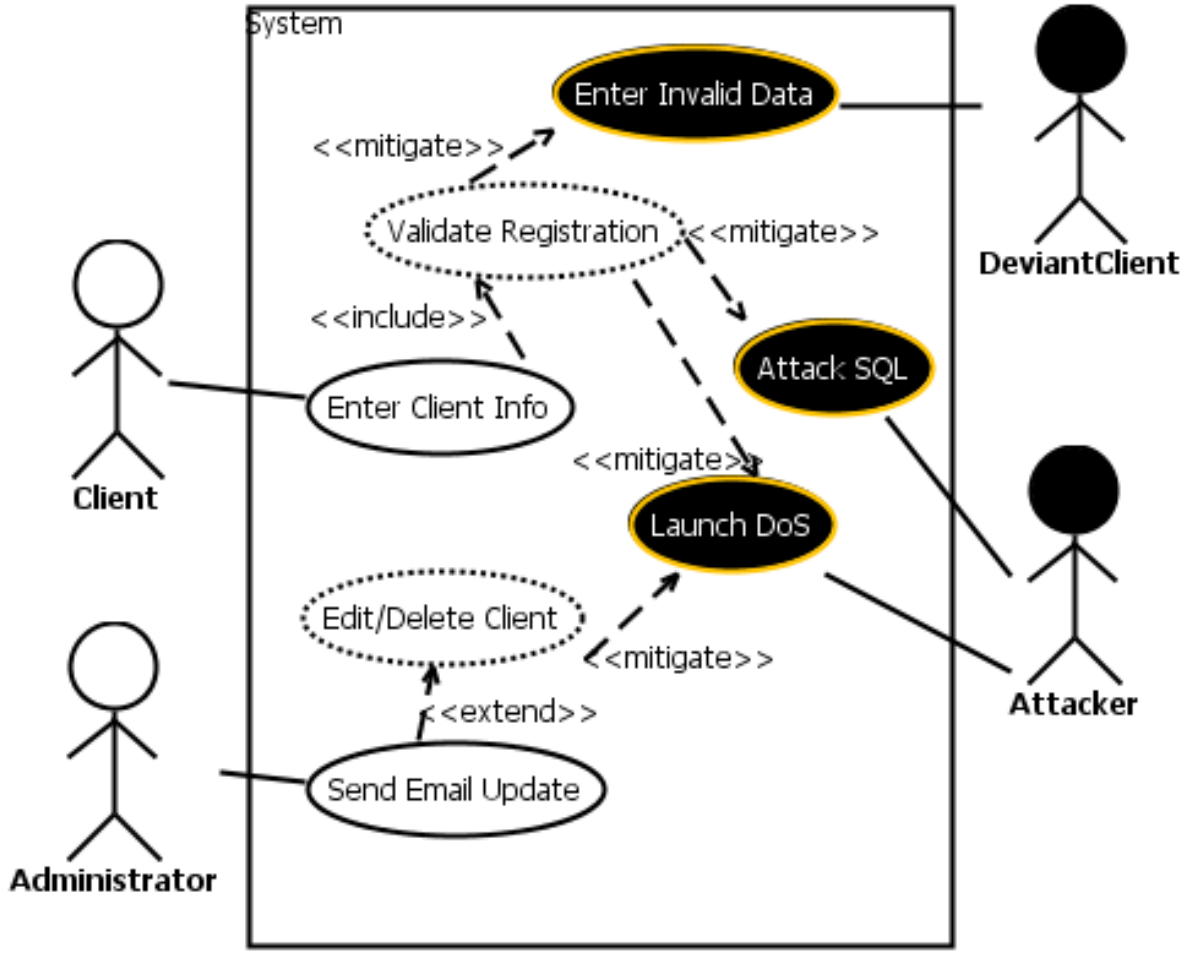
Provider: Send periodic updates to Clients to indicate changes in materials



Define Security Requirements

Definitions





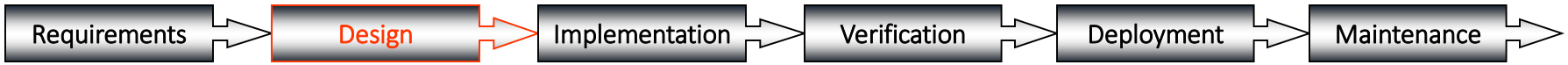
Define Security Requirements
Modify Register Use Case Desc.

Use Case: Register
Summary: Client registers to obtain access to download materials.
Preconditions: Client is at Welcome Web Page
Basic Path: 1. The client selects the Obtain Materials link. 2. The system asks the client for name, email address, job function, and CAPTCHA. 3. The client enters all three required information. *4. Include (Validate Registration)* 5. The system displays the URL for the download materials.
Alternative Path: AP1. If an attack is detected, no URL is displayed.
Postcondition: The client has access to the download materials. The database contains the client contact information.

Define Security Requirements: Validate Registration Security Use Case

Use Case: Validate Registration
Summary: This include validates a registration.
Precondition: A name, email, job function, and Captcha are provided.
Basic Path: 1. The user enters a name, email, and job function in Step 3 of Register 2. Do until valid CAPTCHA. 3. Rerequest form with new CAPTCHA 4. The system checks for valid characters, to prevent SQL injection. 5. The system checks for valid name, email and job function 6. If email is unique in database 7. Save record to database 8. The system returns success.
Postconditions: The input has been checked for bot attempt, SQL attempt, and validity.

SDL – Design Phase



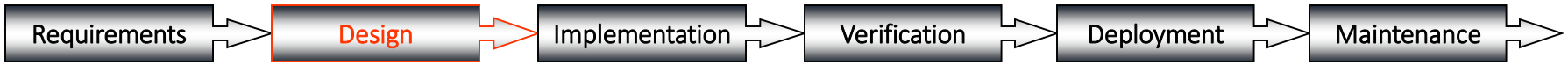
At this stage all design decisions are made, about

- Software Architecture
- Software components
- Programming languages
- Interfaces
- …

Develop documentation

Confirm that all requirements are followed and met

SDL – Design Phase



Treat Models

Input Data Types

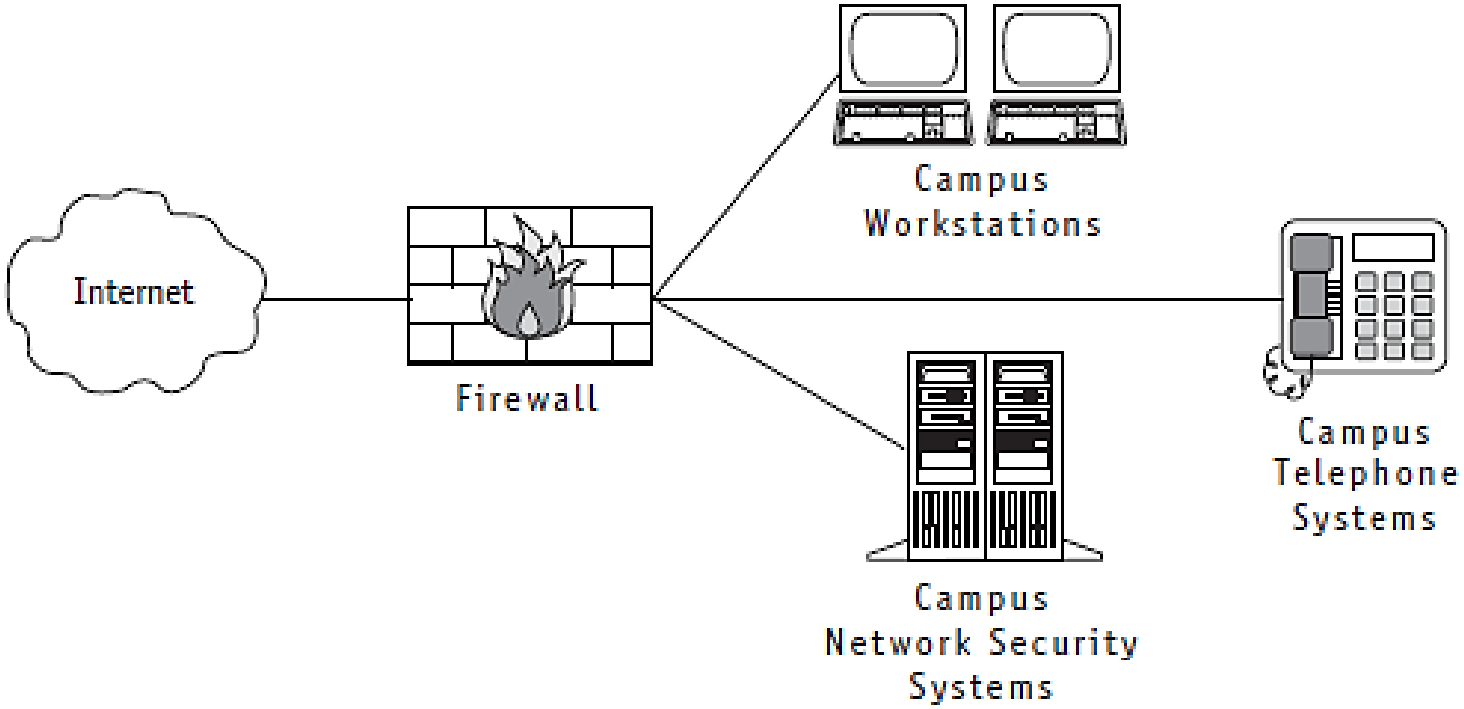
Security Use Cases

Security Architecture

For example: web traffic must pass through a proxy device that is capable of scanning HTTP, HTTPS, FTP, POP3, and IMAP for malicious content

Defense in Layers / Separate Components / Least Privilege

Example Firewalls



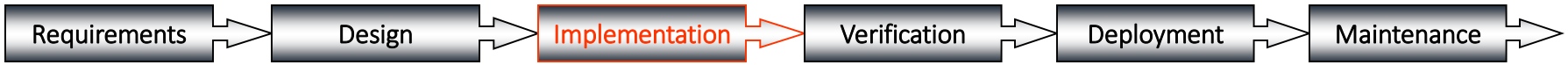
(figure 11.4)

In this example of a university system, the firewall sits between the campus networks and the Internet, filtering requests for access.

Firewalls (cont'd.)

- Typical firewall tasks
  - Log activities accessing Internet
  - Maintain access control (in deciding which packet to forward or block)
    - *Based on senders' or receivers' IP addresses*
    - *Based on services requested (i.e Email ok but block Http request)*
  - Hide internal network from unauthorized users
  - Verify virus protection installed and enforced
  - Perform authentication
    - *Based on source of a request from the Internet*
- Proxy server
  - Hides important network information from outsiders
    - *Network server invisible*
  - Determines validity of network access request

SDL – Implementation Phase



This is the stage where coding is done.

To produce secure software

- Coding Standards
- Centralized Security Modules
- Secure builds and configurations
  - *Known security vulnerabilities - use good programming practices. Be aware of*
    - Race conditions
    - Buffer overflow
    - Format string
    - Malicious logic
    - …

Follow Design & Develop Documentation [further]

Security Vulnerability

Buffer Overflow: Can long input affect service?

Script Injection: Can input with scripts execute?

Numeric Overflow: Can a large number become a negative or small number?

Race Condition: Can multiple threads cause errors?

Configuration Issues: Can software be installed improperly, causing abuse?

Programmer Backdoors: Have programmers left hooks providing entry or information?

SQL : SQL injection

Security Vulnerability

Demo:

SQL injection:

https://www.youtube.com/watch?v=FwIUkAwKzG8

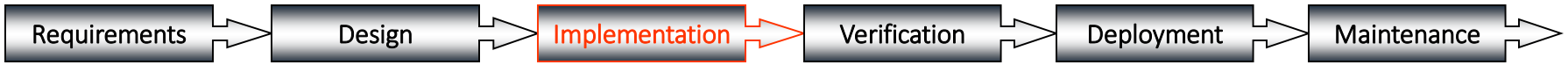
Cross-site scripting:

https://www.youtube.com/watch?v=i38LMZyKIqI

Buffer overflow:

https://www.youtube.com/watch?v=iZTilLGAcFQ

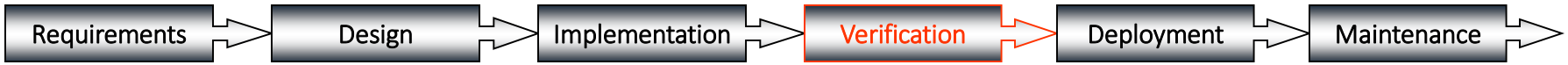
SDL – Implementation Phase



Design and implementation of security features.
From the Requirements

Robust Programming Practices
Good design and coding practices

SDL – Verification Phase



Testing of the code developed in the previous stage

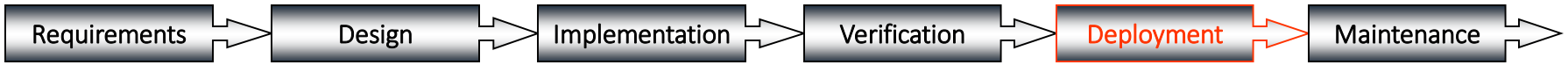
Cleared security tests

Security vulnerability tracking

Code Reviews

Documentation

SDL – Release Phase



Secure Management Procedures

Monitoring Requirements

Defense in Layers / Separate Components / Least Privilege

Security Upgrade Procedures

Vulnerability Testing

Buffer Overflow: Can long input affect service?

Script Injection: Can input with scripts execute?

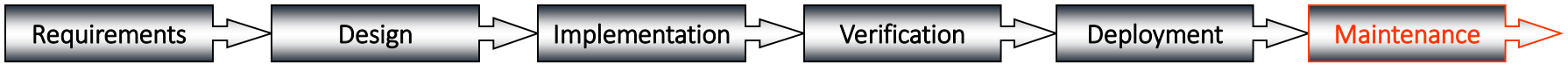
Numeric Overflow: Can a large number become a negative or small number?

Race Condition: Can multiple threads cause errors?

Configuration Issues: Can software be installed improperly, causing abuse?

Programmer Backdoors: Have programmers left hooks providing entry or information?

SDL – Response Phase



Causes:

- Costumer feedback
- Security incident details and vulnerability reports
- …

Types of maintenance

- Need to introduce new functionality
- Need to upgrade to keep up with technology
- Discovered vulnerability

Facts:

Every security vulnerability / flaw overlooked in an earlier phase will end-up at later phase[s]

Resulting into greater

- Cost
- Time
- of the software development and/or maintenance

SDL @ Microsoft