Presentation

With Each Mistake,
There Must Surely Be Learning

SDLC:
Security in the Software Development Life Cycle

Instructor: Dan Kuykendall
Director of Engineering
NT OBJECTives, Inc.

Common Ground

- Massive hookup of databases with confidential data to Internet is new phenomenon in last decade
- Development groups had not planned for potential security problems
- Result is typically thousands of vulnerabilities across enterprises exposing data
- Cost of finding and fixing once software leaves development is exponentially higher than proper planning and execution
- Hackers are getting more sophisticated and "Script Kiddies" more dangerous

Common Ground

- There is no "vendor" to turn to for a patch
- New Laws and Large Dollar amounts are making this important at all levels of business.
- Enterprises need to rethink the development process and add security as a core component
- This will dramatically improve security posture

Business Impact

- Access to unauthorized data could lead to serious problems
  - Loss of important trade secrets (source code)
  - Customer Lists and data
  - Loss of customer trust
- Access to private customer data could have legal ramifications
  - Credit cards
  - Medical records
- Data loss can cause disruption of normal business
  - Backups available?
  - Active data loss may cause confusion and embarrassment

Security Cannot Be An Afterthought

- Traditional “Penetrate and Patch” is a nightmare, and doesn't work
- A formal approach is essential
- Software without security designed in, will eventually become unmaintainable and too costly

Web Application Vulnerabilities

- Traditional Security problems have been at the network and system layers and quickly become “Known Vulns”
- Due to limited number of platforms and versions

- Web Application vulnerabilities are “Unknown Vulns”
- The application is custom
- The vulnerabilities are custom
- There is no vendor to get a patch from

Types of Attacks - Server

- Operating System
  - Vulnerabilities
  - Viruses
  - Worms
- Core Services
- Web Service(s)
- Default Settings
- Backup Files

- All are generally solved with simple installation of vendor patches and hardening procedures

Types of Attacks - Application

- Unvalidated Input
- Broken Access Control
- Broken Authentication and Session Management
- Cross Site Scripting (XSS)
- SQL Injection Flaws
- Buffer Overflows
- Improper Error Handling
- Insecure Configuration
- Denial Of Service (DOS)

- Solutions must be developed and tested internally

Types of Attacks - Database

- SQL Service Ports
- SQL Injection
  - Gain Access to unauthorized data
  - Destroy or Modify data
- Blind SQL Injection

SDLC – Software Development Life Cycle

- Buzzword that means many things to many people
- Encompasses all of the steps that an organization follows when it develops software tools or applications
- Security must be included into each stage of the process
- **Includes many players**
- Breaks the process into stages

The Major Players

- Business Customer
- Development
- Database Administrators
- QA
- Security Team
- Server Administrators (IT Department)

Business Customer/Requirements

- Focus is on functionality
- Rarely consider security
- Generally demand quick turn around
- Do not often appreciate the difference between a well designed solution, and a quick hack to solve the need.

Developers

- Rarely have strong security experience or training

- Web Application Developers tend to fall into these categories
  - Converts from in-house development of GUI or Mainframe style applications
    - Great development skills
    - Not used to the wild west lifestyle of the Internet

  - New developers, who are learning at the 3rd Generation level
    - Not experienced at “Engineering” applications
    - Not familiar with issues such as encoding types and memory management

Database Administrators

- Grant access to the databases and tables within
- Understand the data, and databases
- Understand SQL Language, and Stored Procedures
- Unfortunately they are generally cut out of the picture once development starts

QA Team

- Primary focus is on functionality
- Do not often have the tools or know-how to test security issues
- By the time QA gets involved, it's often too late

Security Teams

- Generally part of the Network group
- Have very little influence or interaction with Development teams
- If brought into the Development Process at all, its usually during the QA Process

Server Administrator

- Responsible for maintaining the server(s)

- Responsible for many "Known Attacks", such as
  - Attacks against the Operating System
  - Attacks against the Core services
  - Attacks against the web server
  - Attacks against default installed CGI scripts

- Knows little or nothing about "Unknown Attacks"
- First man on the scene when an attack takes place

- Unfortunately does not generally interact with the Development team

SDLC – Software Development Life Cycle

- Buzzword that means many things to many people
- Encompasses all of the steps that an organization follows when it develops software tools or applications
- Includes many players
- **Breaks the process into stages**

SDLC – Software Development Life Cycle

- Can be broken down in many ways

- At the highest level we see four major stages
  - Planning
  - Development
  - Implementation
  - Maintenance

SDLC – Software Development Life Cycle

- The Life Cycle happens whether we plan it out or not
- Without a plan, the cycle will be painful, costly and will likely be filled with Security problems
- A solid and well considered plan can generate more robust and secure software

Illustration

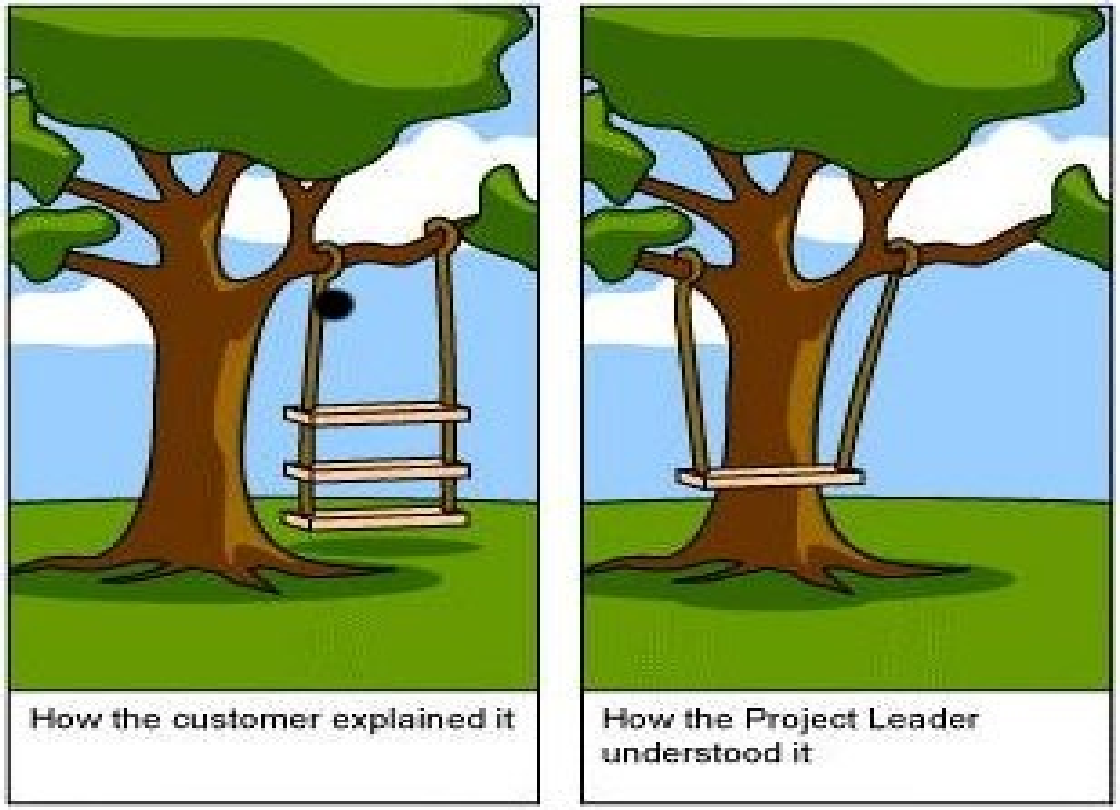

Illustration from ScaryIdeas.com

NT OBJECTIVES, INCORPORATED

Illustration

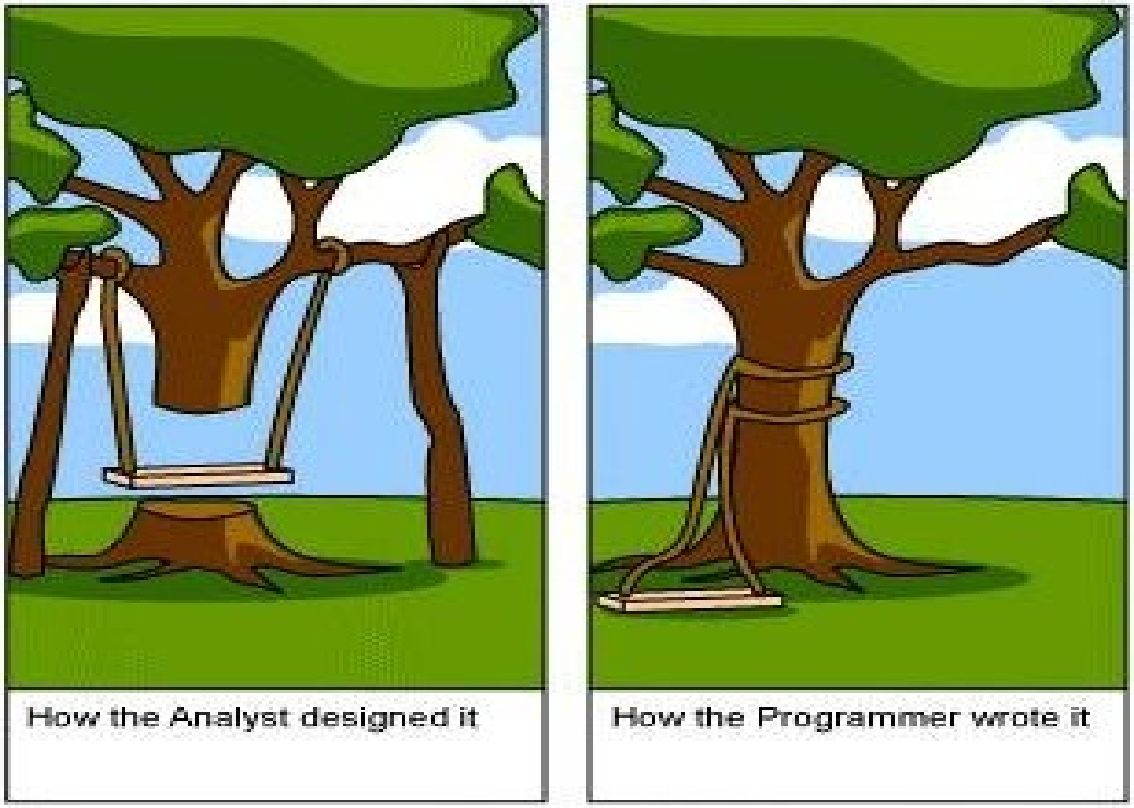

Illustration from ScaryIdeas.com

NT OBJECTIVES, INCORPORATED

Illustration

Illustration from ScaryIdeas.com

NT OBJECTIVES, INCORPORATED

Illustration

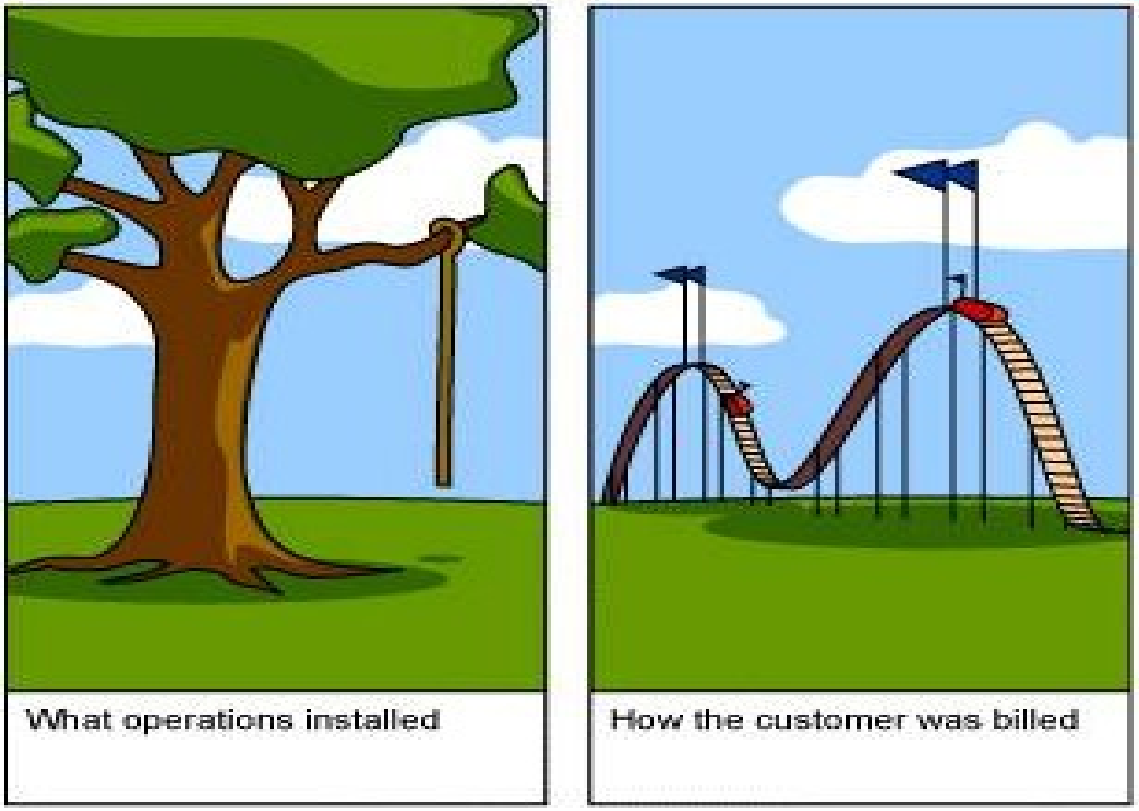

Illustration from ScaryIdeas.com

NT OBJECTIVES, INCORPORATED

Illustration

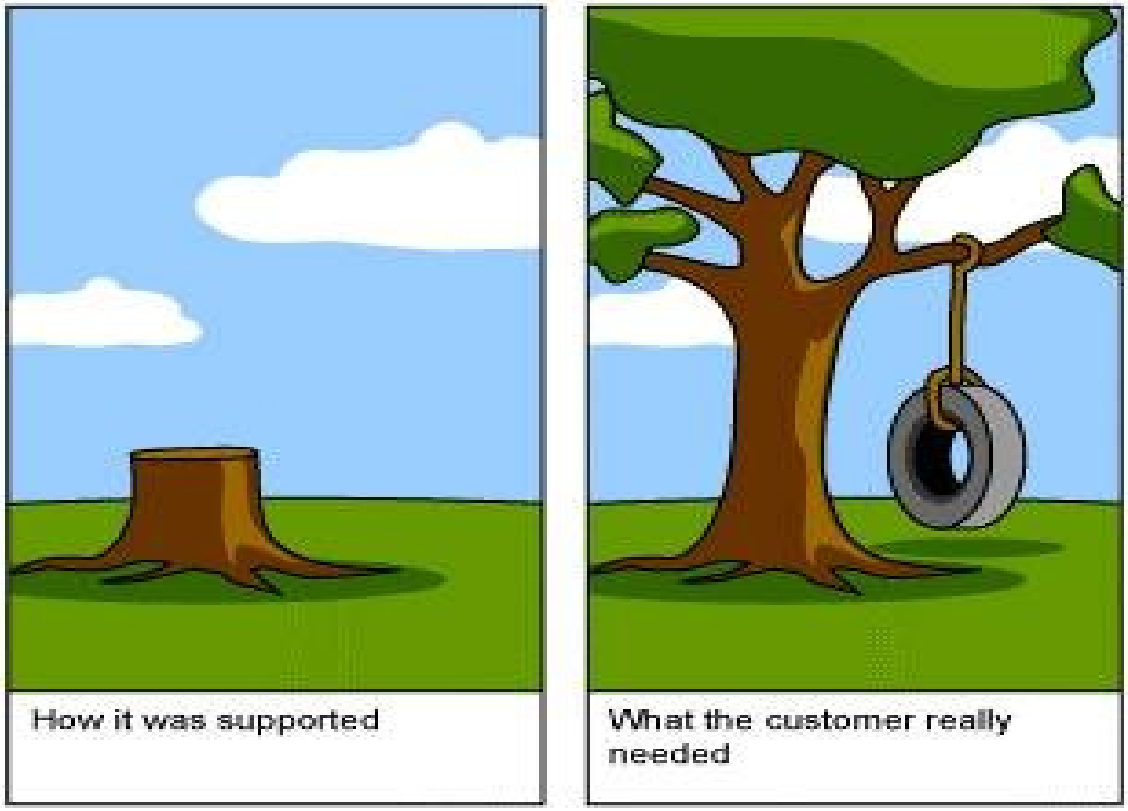

Illustration from ScaryIdeas.com

NT OBJECTIVES, INCORPORATED

SDLC – (Lack of) Planning

- Conceptual definition
  - User determines that software is needed to handle a business task
- Functional Requirements and Specifications
  - User defines what functionality the software must include

---

- Too often this stage is too brief or in worst case, completely verbal
- Rarely includes discussion of Security
- Does the organization have a formal security policy to follow?

SDLC – Proper Planning

- Conceptual definition
  - User determines that software is needed to handle a business task
- Functional Requirements and Specifications
  - User defines what functionality the software must include
- **Technical Specifications**
  - Detailed description of the technical requirements and specifications
  - Definition of small discrete components/objects for each task
- Architecture
  - Platform, Technologies and Languages that will be used

SDLC – Proper Planning

- Development Process
  - Classic
    - Developers given tasks, work largely independently
    - Should include peer review
  - Agile
    - Extreme Programming
    - Pragmatic Programming
    - Scrum
    - Crystal
    - Lean Manufacturing
- Project Management
  - Without a good Project Manager the software will go off course
  - Make sure schedule is realistic and provides for QA process

SDLC – Proper Planning

- Security Issues
  - Organizational Standards and Policies
  - Legislation
    - Sarbanes-Oxley (Everyone)
    - Gramm-Leach-Bliley (Financial Institutions)
    - HIPPA (Medical)
    - COPPA and FISMA (Child Protection)
    - CAN-SPAM (Spam, and Marketing)
    - Credit Card Processing rules
  - Platform and Technologies to be used
  - Database structures

  - **Input Validation** is the most common problem and the one that will be the hardest to go back and fix

SDLC – (Messy) Development

- Developers take requirements and start coding
- Specifications? What Specifications?
- Security handled differently by each developer. No centralized solution.

- This stage can easily be the a complete mess

SDLC - Development

- Stick to the Specifications!
  - Deviating from the specifications should be avoided
  - Change the specifications and get approval by all parties
- Make full use of the chosen Development Model
- Engineer the application
- Peer Review – Avoid individual code territorial issues

SDLC - Development

- Build Security in from the ground level
  - Security Training for developers
  - Make use of security best practices
    - Standardized and Centralized Input Validation
    - Deny all, Allow as necessary (Least and Necessary Privileges)
    - Protect your database - Avoid having the web application generate SQL Statements
  - General development best practices improve security
  - Authentication and Session Management

Deny All, Allow as Necessary

- Not just for the Network layer
- Do not "clean" bad inputs
- Define the expected data, and deny everything else
- If input validation is implemented and forced the security problems will be drastically reduced

Avoid SQL Statement Generation

- Web Application Developer and SQL Administrators
  Must work together
- Too often the web application code is responsible for generating all SQL Statements based on user input
- In the event of SQL Injection vulnerability, the database is fully exposed
- Make use of SQL Database permissions, and Stored Procedures

SDLC – Development - QA

- Testing can not be an afterthought or something that can be cut back when the schedule gets tight
- Provide QA team with security training and tools
  - Training to understand security issues
  - Tools to automate security testing
- Include Security Team into QA process
  - Don't wait till application is in production before doing an audit
- Performance Testing – Avoid simple DOS attacks
- Data migration testing

SDLC – Implementation - QA

- Involve Server Administrators (IT Department) in the QA process

- Too often the Development and QA environments do not exactly match the Production environment
  - System Locked Down
  - Various patches and service packs that may be installed
  - Load balancing solutions
  - Error handling

SDLC - Implementation

- Data Conversion
- Documentation
- Training
- Application Deployment
  - Final Stage of releasing the application
  - Not the Final Stage of the application
- Feedback and Support solution should be in place
- Have Developers and QA teams on hand

SDLC - Maintenance

- Ongoing Process
- Continuing Support of End Users
- Bug Fixing
- Security Monitoring and Auditing
- Upgrades

Doing It Right The First Time

- Will take longer up front
- Will cost more up front
- Will meet the requirements
- Will be more secure
- Is more likely going to be delivered on time
- Will be more maintainable
- Will cost less over its lifetime

Review

- Security Problems are becoming a serious bottom line issue
- Web Application Vulnerabilities need to be addressed in new ways
- Web Application Security must be integrated into the SDLC
- If integrated into the SDLC you can achieve secure software
- If integrated into the SDLC you can cut down on costs over the lifetime of the project

Presentation

With Each Mistake,
There Must Surely Be Learning

For a copy of this presentation
sales@ntobjectives.com

Instructor: Dan Kuykendall
Director of Engineering
NT OBJECTives, Inc.