**Potential Security Objectives:**

If credit card numbers, Visa PCI

If financially significant info, Sarbanes-Oxley

Policy Holder info such as SSN, Health (HIPA), and competitive information

Agency/Agent info such as who and what is insured, competitive information, reputation

Insurance Company info such as competitive information, reputation, etc.

**Identified Assets:**

Information:

Policies

Policy datastore

Authentication data

Authentication datastore

Authorization data

Authorization datastore

Configuration information

Configuration datastore

Application authentication data

Application authentication datastore

Session data

Session datastore

Candidate username, password, and policyIDs

Backups

Archivals

Source Code

Binaries

Project resources such as images, html, other content

Temp files

Log files

Error/debug files

Audit files and records

Third party software

Cookies

Hidden fields

SessionID

Certificates, keys, and other crypto artifacts

Functional:

Authentication Service

Authorization Service

Policy Viewer Service

Policy Datastore service

Authentication Datastore service

Authorization Datastore service

Configuration Datastore service

Session Datastore service

Auditing Service

Logging Service

Error Handling Services

Server Side Interpreters – SQL Engines, XSLT Transformers, XPath Evaluators, XQuery Evaluators, etc.

XML Parsing – Web Services, AJAX frameworks, Component Configuration, etc.

Client Side Interpreters – JavaScript, Flash, etc

Infrastructure such as OS, web server, databases, network

**Trust Boundaries (examples):**

Web Container(WC)/HTTP

WC/Auth Service

WC/Policy Datastore

WC/Configuration Datastore

WC/File System

**Solution to code reviews:**

Authentication

Brute Force

Dictionary

Cookie replay

Weak Passwords

Did not use "standard" challenge mechanism

Did not use SSL (or at least there is no checking to see if used)

Exception Handling

Sensitive System details

Different types of response based on whether a password or a username

Denial of Service

Poor handling in general, no consistent framework

Input Data

Mainly for username, password, and policy number

Buffer usage (no constraint checking)

Cross-site scripting (potential)

SQL injection (potential)

Cookie manipulation

Sensitive Data

In storage

File distributed with app in WAR file

Cookie content

DiagnoseHTTP

Session Management

Session hijacking

Lifespan of session

No explicit logout option for user

Auditing and Logging

Exploit without a trace

Sensitive data in logs

Configuration

Source code being deployed

Unused resource (PolicyViewer.java) being deployed

Default error page from Tomcat

Version being revealed by Tomcat

Authorization

There is none when it is apparent there should be something in place.