

Security Assurance: An Example

Paddy Krishnan

(Joint work with Andreas and Sergej) Centre for Software Assurance

Bond University, Gold Coast, Australia

Email: pkrishna@staff.bond.edu.au

Background

- US-DHS forecast a shortage of security professionals
- They provide a wide variety of sources that can be used in teaching
- Security: Network/Infrastructure, Application
- This project arose from investigating projects for the teaching of application level security leading to exploration of research level problems

Plan

- Simple Assurance Example
- Various choice of Guidelines
- Applying Various QA Techniques

Key Issues

What process to follow to assurance an application for a mobile platform?

How can this be taught to senior students?

Can we identify automatable techniques (still open)?

Prior Information

- Number of SDLC processes
- Number of vulnerability databases (OWASP, CWE)
- Taxonomy of threats: both general and specific
- Number of programming suggestions
 - General language: Oracle (Java),
 - Platform/Application Domain Specific: Android
- Build Security in Software Assurance Initiative provides general guidelines

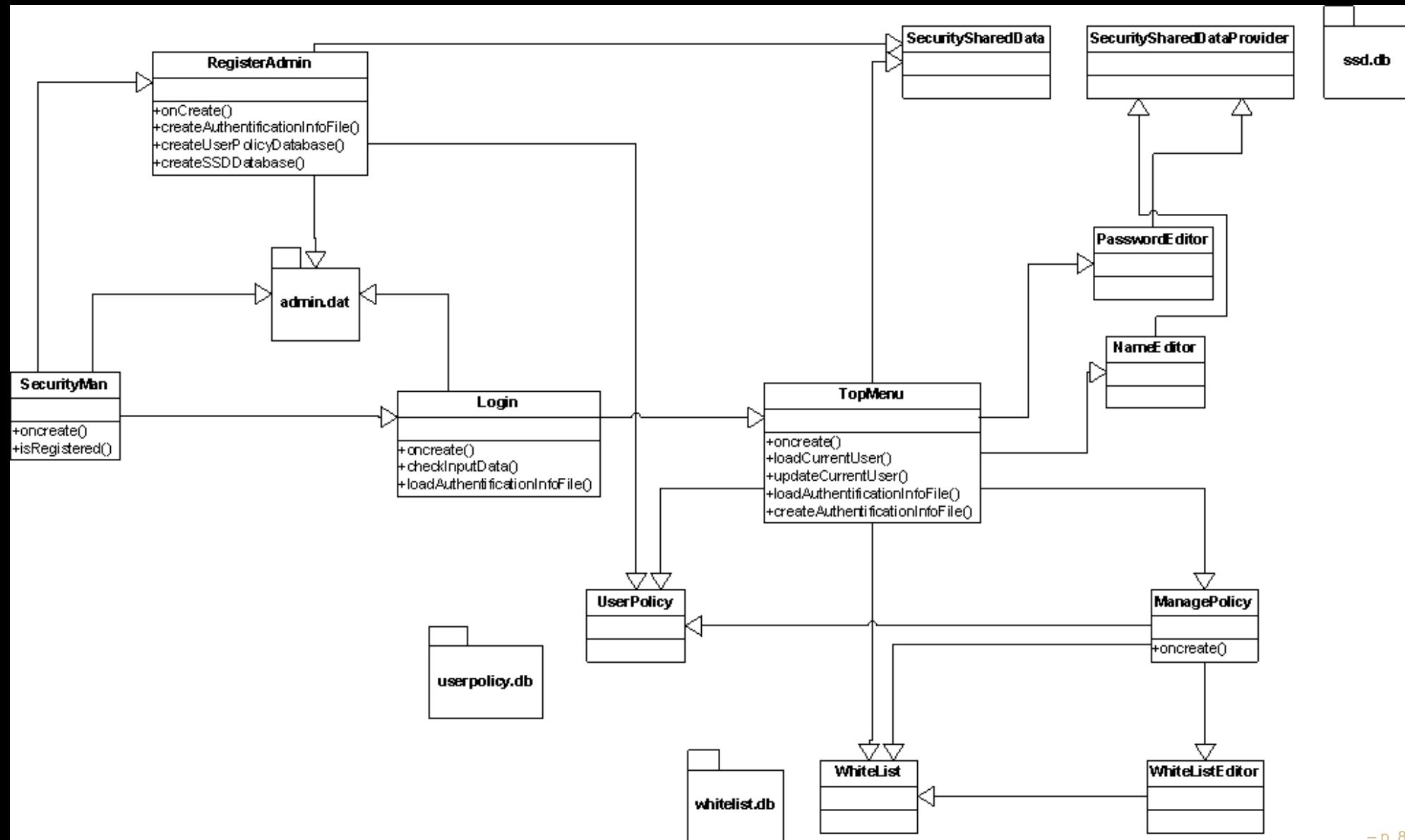
Application

- For the Android Environment
- Change the standard dialler
- White-lists for sending, receiving calls/messages
- A security manager to manage the changes to the white-lists
- Application not developed by security assessor

Particulars

- Trusted certification vs User approval
- Language specifics: Java vs C
- Operating System guidelines
- General guidelines

Security Manager



Key Points

- Initial class diagram obtained from code
- Various Data-Bases
- Read/Updates based on user input
- Passwords: Editing/Usage

Process

- Use the design to understand the system and assurance process
- Identify the key functionality and the resources necessary in the implementation
- Identify the attack surface based on the above step (but we do not attach any numeric measure to this). Using the published guidelines we map attacks to the identified surface.
- Validate (via testing, verification or code-inspection) to determine if the system is resilient to the attacks.

Behaviours

- Any user can attempt to register or login as administrator (Login).
- Only an authenticated administrator can create or delete an account (Create/Delete).
- The administrator can change a user's password (Change Password)
- Only an authenticated administrator can change the policy setting of any user (Change Policy)
- Only an authenticated administrator can launch the secure dialer via the security manager.

Resource Summary

Fnc	Class(es)	Resource
Login	SecurityManager.java Login.java	admin.dat
Create/Delete	UserPolicy.java SecuritySharedData.java	userpolicy.db ssd.db
Password	PasswordEditor.java	ssd.db admin.dat
Change Policy	ManagePolicy.java SecuritySharedData.java	whitelist.db ssd.db

Attack Surface

Guideline	Brief Description	Functionality
CWE-805	Buffer Overflow	All functions
Cert Java IDS00-J	Validate Input	All functions
CWE-311	Missing Encryption of data	Create User Change Password Change Policy
CWE-732 Cert Java ENVO3-J	Permissions of resources	Change Password
Cert Java MSC01-J	Weak Crypto	All functions
Cert Java EXP03-J	Comparing Strings	Login
CAPEC-112	Brute-Force	Login

General Guidelines

Guideline	Brief Description
CWE-129	Validation of Array Index
Cert Java EXP07-J	Short circuit of conditionals
Cert Java MET03-J	Security check: private or final
Cert Java MET04-J	Use of overridable methods

Model-Based Testing

- Simple SAL model: generic execution, application behaviours
- Reachability analysis using SAL
- Generate test sequences using SAL-ATG
- Test sequences hard to map to implementation
- So model-based testing rejected

Variety of Techniques

- Code inspection
 - Use of encryption algorithm: library call
 - Setting of file permissions
- Black box Testing : Three failed logins: locked
- Unit testing: readability of contents of files
- Static analyser: FindBugs: conditionals having side-effect

Summary

- A report on the analysis is produced
- 100 hours to understand guidelines: can be reused
- 100 hours to actually carry out assurance
- Totally 270 validation items: many of them are very small

Lessons/Open Issues

- Can teach application security assurance to senior UG and coursework masters students
- The case study chosen was large enough to have different issues but small enough to be tackled by a group of students.
- Generalisability is thus an issue
- Need to identify automatable techniques

Thank You

Questions??