Information Society Technologies

S3MS Security of Software and Services for Mobile Systems

# 3.2.2(b)- Prototype Description Report Proof Generator / Verifier

Andreas Lundblad

| | |
|---|---|
| **Document Number** | 3.2.2(b) |
| **Document Title** | Prototype Description Report Proof Generator / Verifier |
| **Version** | 1.1 |
| **Status** | Final |
| **Work Package** | WP 3.2 |
| **Deliverable Type** | Prototype |
| **Contractual Date of Delivery** | 29/02/2008 |
| **Actual Date of Delivery** | 14/03/2008 |
| **Responsible Unit** | KTH |
| **Contributors** | KTH |
| **Keyword List** | Run-time Monitoring, Proof-Carrying Code |
| **Dissemination level** | PU |

# Change History

| Version | Date | Status | Author (Unit) | Description |
|---|---|---|---|---|
| 1.0 | 27/09/2007 | Draft | Andreas Lundblad (KTH) | First version |
| 1.1 | 14/03/2008 | Final | Andreas Lundblad (KTH) | Final version |
| | | | | |
| | | | | |

# Executive Summary

This report documents the proof generator/verifier prototype tool. It describes the two components and gives requirements and instructions for how to to run them.

# Contents

# 1 Introduction

Our approach is based on an automated static analysis. The proof generator analyzes a given inlined program and its contract specification and generates on the basis of this a contract adherence proof. The proof is embedded in the software package which is distributed to the consumers. As a consumer receives such a package it simply checks the validity of the proof before executing the program.

# 2 The Tool Components

The tool is divided into two parts: a proof generator ("producer" side) and a proof recognizer ("consumer" side). The producer tool automates the proof generation and packaging. The consumer tool parses and verifies the proofs.

## 2.1 Producer side

The producer side of the tool is a Java-based desktop application. The components with corresponding input and output are illustrated in figure 1.

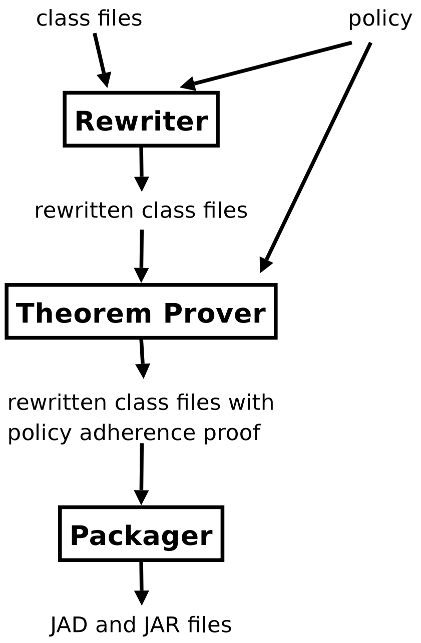


Figure 1: The components of the producer part of the tool.

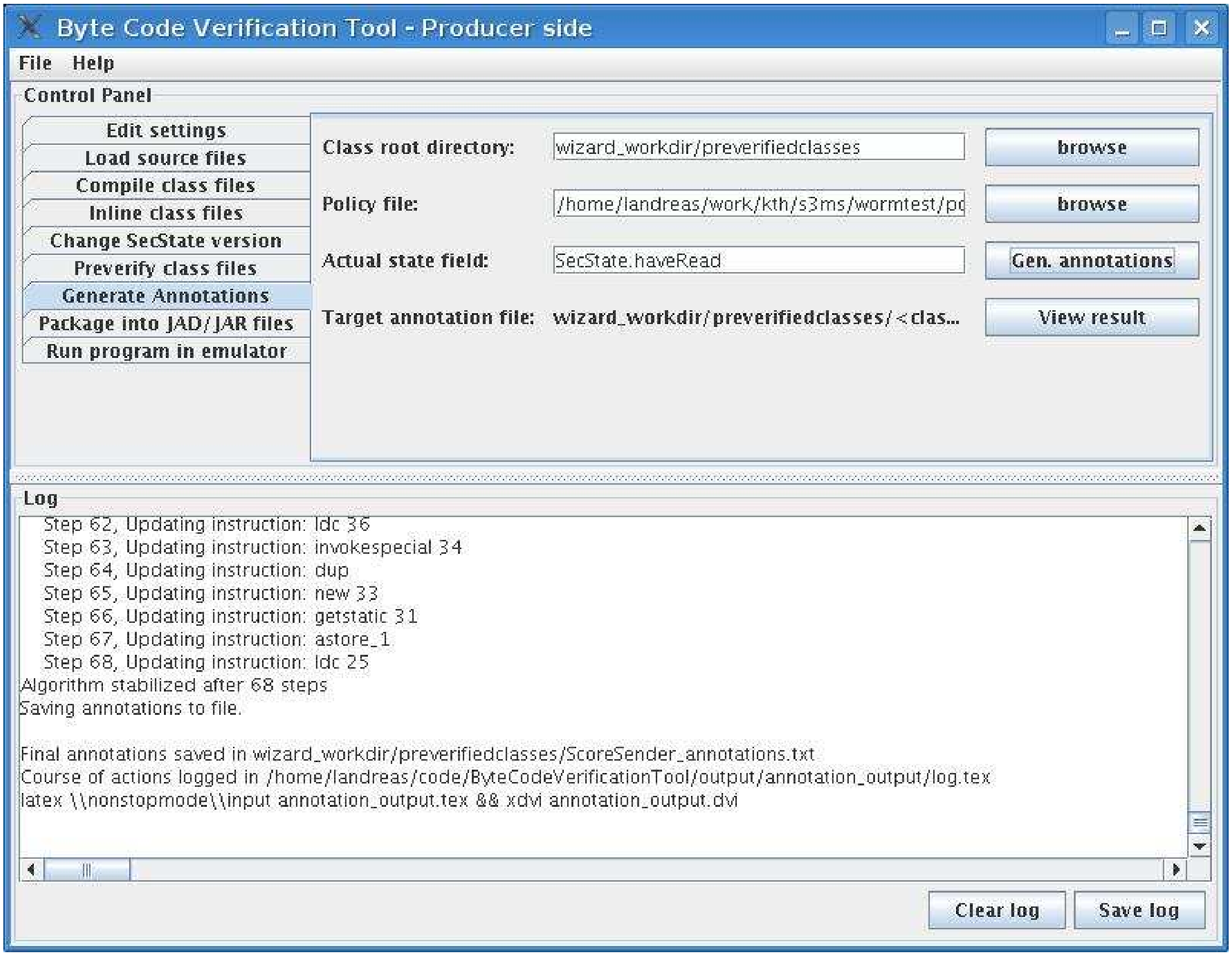
The tool can be downloaded from the demo page: `http://www.csc.kth.se/~landreas/web_demo`
A screen shot of the user interface is shown in figure 2.

## 2.2 Consumer side

The consumer part of the tool is a Java MIDlet application that runs on a regular CLDC-1.1 enabled mobile device.

The components with corresponding input and output are illustrated in figure 3.

The tool can be downloaded from the demo page: `http://www.csc.kth.se/~landreas/web_demo`



Class root directory: wizard_workdir/preverifiedclasses

Policy file: /home/landreas/work/kth/s3ms/wormtest/pc

Actual state field: SecState.haveRead

Target annotation file: wizard_workdir/preverifiedclasses/<clas...

```
    Step 62, Updating instruction: ldc 36
    Step 63, Updating instruction: invokespecial 34
    Step 64, Updating instruction: dup
    Step 65, Updating instruction: new 33
    Step 66, Updating instruction: getstatic 31
    Step 67, Updating instruction: astore_1
    Step 68, Updating instruction: ldc 25
Algorithm stabilized after 68 steps
Saving annotations to file.

Final annotations saved in wizard_workdir/preverifiedclasses/ScoreSender_annotations.txt
Course of actions logged in /home/landreas/code/ByteCodeVerificationTool/output/annotation_output/log.tex
latex \\nonstopmode\\input annotation_output.tex && xdvi annotation_output.dvi
```

Figure 2: A screen shot of the producer tool.

# 3 Requirements

## 3.1 Producer Tool

- Sun JDK (or JRE) 1.5 or later

  One step of the producer-tool involves compiling the source files. If source is already compiled this step can be skipped. In this case you will not need the full JDK. However you will still need the Java run-time library since the program itself is based on Java.

  Download url: `http://java.sun.com/javase/downloads/index.jsp`

- Sun Java Wireless Toolkit for CLDC

  If you plan to work with mobile applications such as MIDlets the following steps require this software: Compiling, inlining, preverification, running the emulator.

  Download url: `http://java.sun.com/products/sjwtoolkit/download-2_5_1.html`

- The inliner tool

  This is required for the inlining step but not required for the rest of the application to run properly.

  Download url: `http://www.csc.kth.se/~irem/s3ms/inliner`

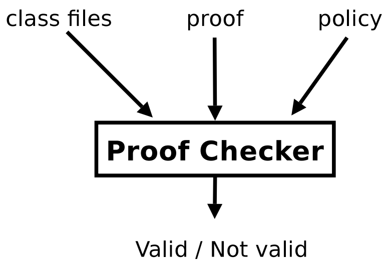


Figure 3: The components of the consumer part of the tool.

### 3.2 Consumer Tool

The consumer tool is a self contained MIDlet and does not require any other software other than the MIDP/CLDC runtimes.

## 4 Running the tool

### 4.1 Producer-side

To start the tool you type the following:

```
java -cp AnnotationGenerator.jar se.kth.s3ms.annotations.gui.Gui
```

### 4.2 Consumer-side

To start the tool you either download it to your cell phone or you could run it in the WTK emulator. To start it in the emulator you type the following:

```
path/to/wtk2.5.1/bin/emulator -Xdescriptor BCAV.jad
```