ADSS – User Manual

ASPIRE Decision Support System

Politecnico di Torino

December 7, 2015

Contents

Disclaimer

This manual documents the ASPIRE Decision Support System as released in date December 7, 2015 and its content it is likely to change as new versions are released.

1 Installing the ADSS

The simplest way to start using the ADSS is to mount and run the ASPIRE virtual image, however the ADSS can also be installed on another machine.

Before actually installing it, the host system that must met some prerequisites.

The ADSS requires the following components to be correctly installed:

- Java Runtime Environment 7.x or superior;
- SWI-Prolog 7.x or superior (http://www.swi-prolog.org);
- CodeSurfer 2.x or superior (with a proper builder license);
- the ASPIRE Annotation Extractor;
- the ACTC;
- at least one ASPIRE software protection tool.

Under a Debian/Ubuntu you can easily install the JRE and SWI-Prolog via the following command lines:

```
aptitude install openjdk-7-jre openjdk-7-jre-headless
aptitude install swi-prolog swi-prolog-java swi-prolog-nox
```

The other components installations requires however some manual effort.

Once your system has been prepared, you can start the actual ADSS installation in two different ways:

- copy the ADSS RCP package anywhere and decompress it;
- download any flavor of Eclipse 3.7 Indigo you like from http://www.eclipse.org/indigo/ and use the update site something to install the ADSS plug-ins into your IDE. Fix it

2 Overview

The ADSS main task is to automatically (or semi-automatically with some hints from the user) protect a software application.

This job is performed by following a sequence of phases that are:

1. *initialization* (automatic): the AKB is loaded, any previously saved results are restored and all the ADSS internal components are prepared;
2. *application setup* (manual): the user provides the source files that must be protected. They must be adequately annotated;
3. *code parsing* (semi-automatic): the source files are parsed (using the CodeSurfer and Annotation Extractor tools) and their structure is inferred. The user can eventually fine tune the acquired data;
4. *attacks deduction* (automatic): all the attacks that can threaten the application are inferred;
5. *protection deduction* (semi-automatic): the protection techniques that can be used to block the attacks are inferred. The user can choose to override some decisions, if needed;
6. *golden solution search* (automatic): the golden (best) solution that can be used to protect the application is found. This is the most time-consuming phase;
7. *golden solution deployment* (automatic): the golden solution is effectively applied to the application source code files (via the ACTC).

Noy yet available

Noy yet available

Noy yet available

The user can choose to stop anywhere in between the aforementioned steps, save the results (in the AKB) and resume later.

3

Tutorial: getting started

In this chapter, a tutorial describing how to protect a software application using a mostly automatic approach via the ADSS is presented. A more advanced approach is described in Chapter 4.

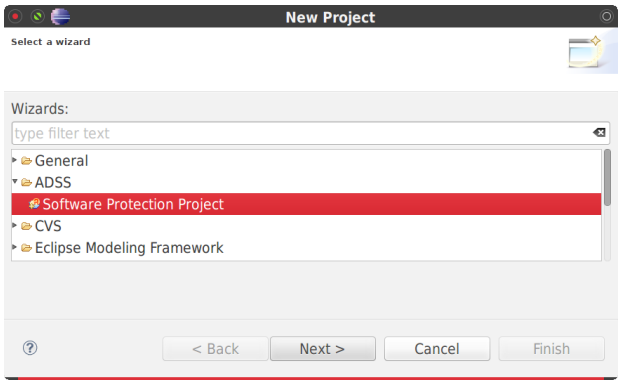
Note that the ADSS is a project-based tool, that is for each application to protect the user must create a new project.

> Note: for the time being, the ADSS contains only an hard-coded application (an OTP generator).

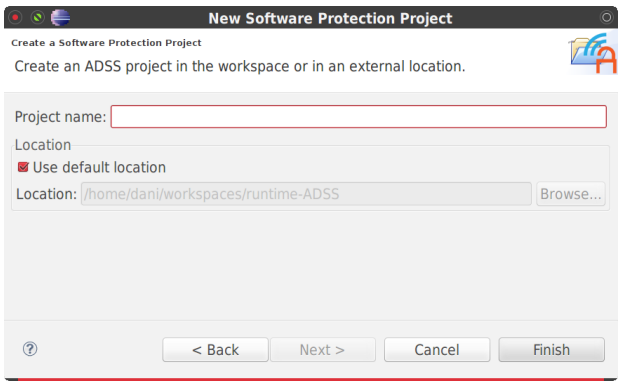
To start the protection process, perform the following steps:

1. launch your Eclipse IDE with the ADSS plug-ins installed;

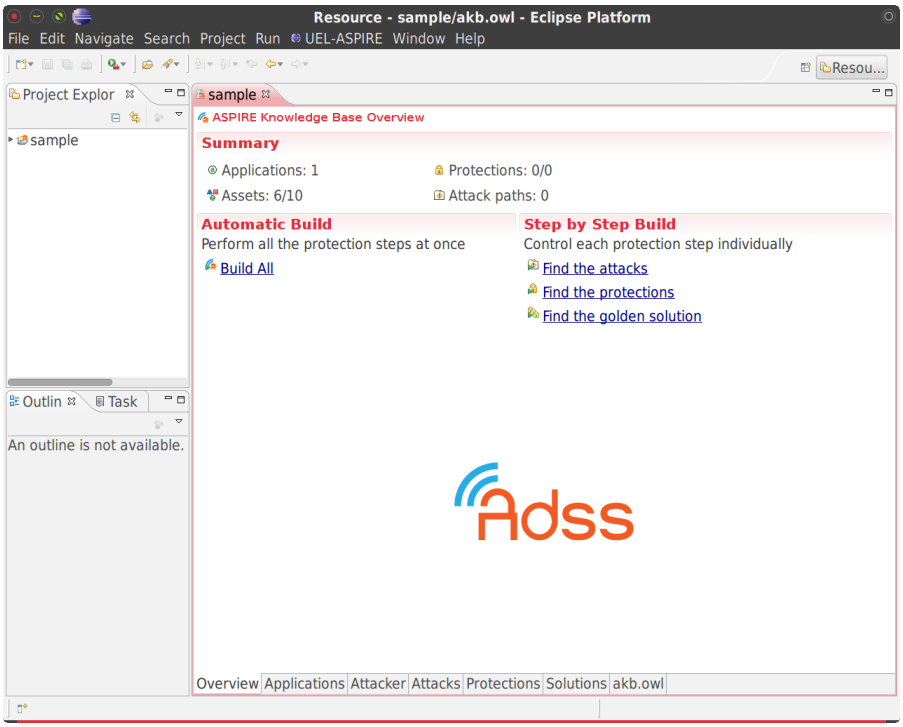
2. create a new ADSS project by choosing `File > New > Project...`. In the `New Project` wizard choose `ADSS > Software Protection Project` and click on `Next >`;



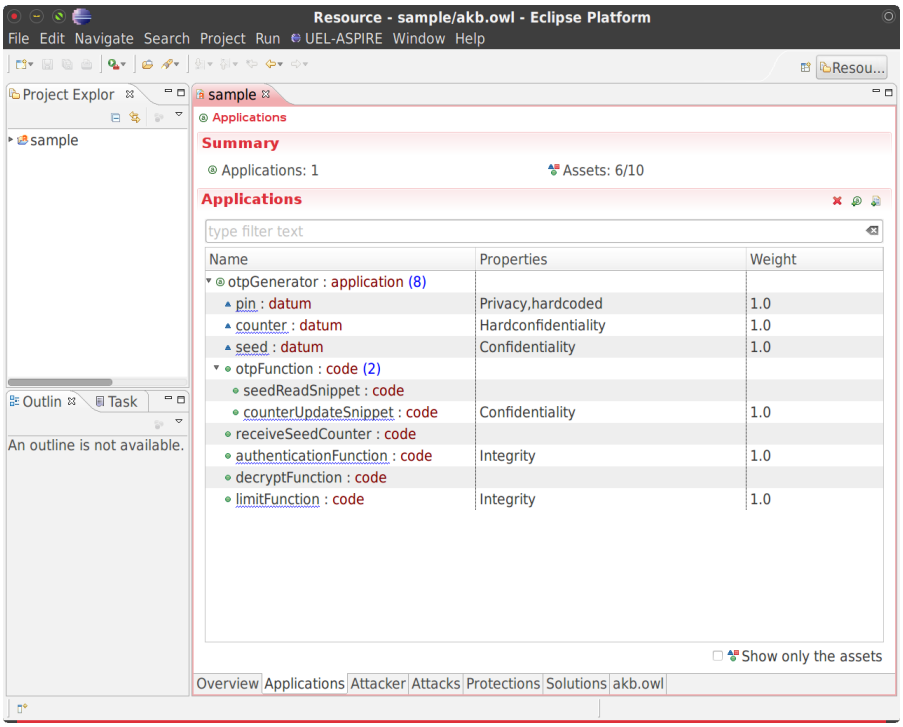
3. in the `New Software Protection Project` wizard choose a project name, a location directory (if different from the default one) and click on `Finish`. If you leave empty the name field, the ADSS will generate a random UUID for you;



4. the AKB editor should appear, opened on the `Overview` page;

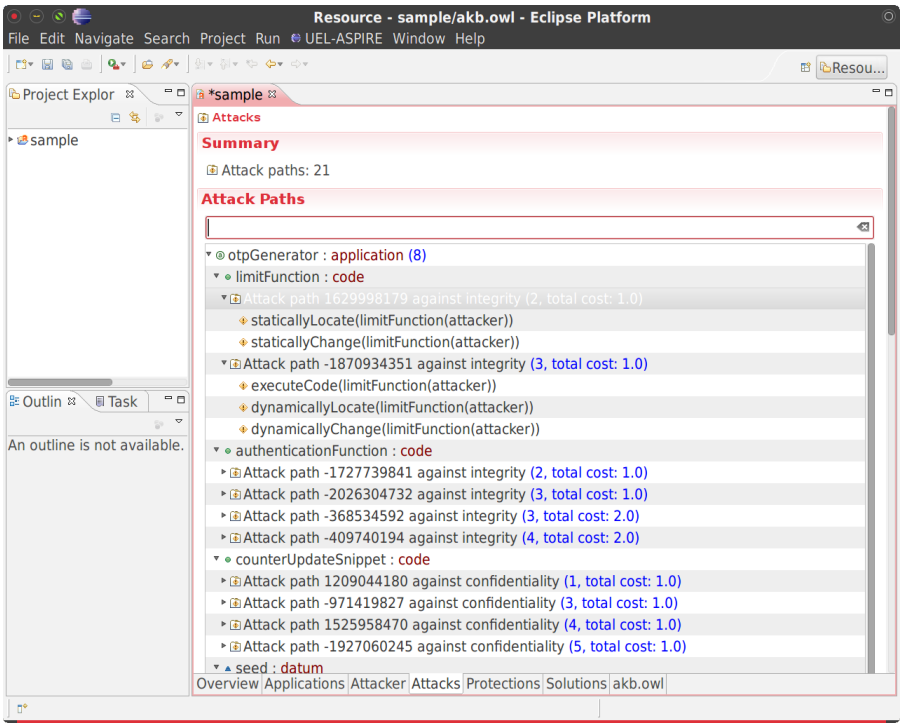


5. (optional) click on the `Applications` page to modify the assets and their properties. You can click on the `Properties` and `Weight` cells to change the application parts security requirements and weight;

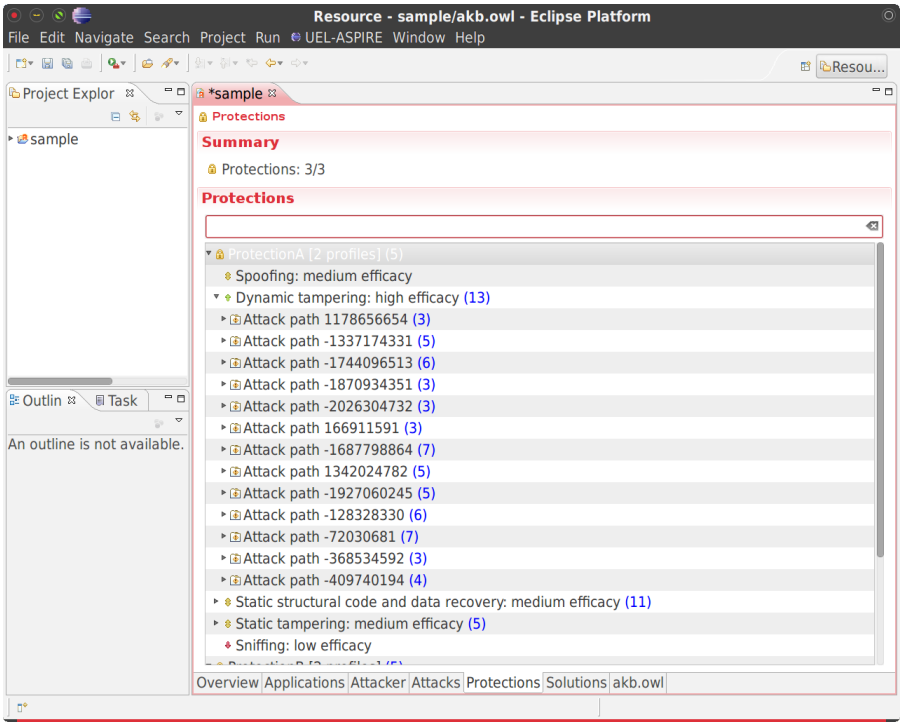


6. return to the `Overview` page, if needed, click on `Build All` and wait until the ADSS finishes its job;

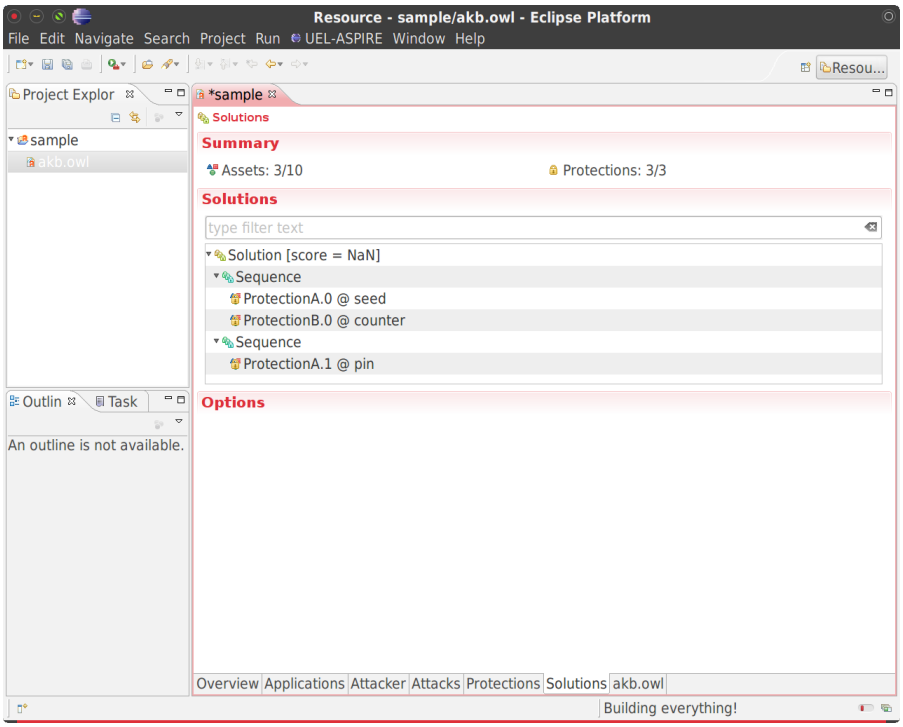
7. in the `Attacks` page you will find all the discovered attack paths and steps;



8. in the `Protections` page you will find all suitable protections found to block the attacks;



9. in the `Solutions` page you will find the best solutions that the ADSS found to protect the application, including the golden one (the first one in the list);


Resource - sample/akb.owl - Eclipse Platform
File Edit Navigate Search Project Run UEL-ASPIRE Window Help
Project Explor
sample
akb.owl
Outlin
Task
An outline is not available.
*sample
Solutions
Summary
Assets: 3/10
Protections: 3/3
Solutions
type filter text
Solution [score = NaN]
Sequence
ProtectionA.0 @ seed
ProtectionB.0 @ counter
Sequence
ProtectionA.1 @ pin
Options
Overview Applications Attacker Attacks Protections Solutions akb.owl
Building everything!

Tutorial: advanced use of the ADSS

4

I'll do it! One day...