Androidアプリの脆弱性の学習・点検ツール
「AnCole（アンコール）」
利用マニュアル

独立行政法人情報処理推進機構

2014. 4

目次

1. AnCoLe(アンコール)とは

Android アプリの脆弱性の学習・点検ツール「AnCoLe(アンコール)」(以降、本ツール)は、Android アプリの開発者を対象とした、脆弱性が作り込まれてしまう原因や対策について実習形式で学べるツールです。
利用者は本ツールを利用して、以下を行うことができます。

1. Android アプリの脆弱性に関する学習(学習機能)
   本ツールが提供するシナリオを通して、アプリ開発時に注意すべき脆弱性の情報や、脆弱性の対策方法について学習できます。

2. プロジェクトに対する脆弱性の点検(点検機能)
   自分が開発したアプリのプロジェクト(ソースコードや設定ファイル)に対して、脆弱性となり得る問題箇所がないか点検することができます。

2. 利用者マニュアルについて

利用者マニュアル(以降、本文書)は、本ツールの使い方について説明するものです。
本ツールを使用する際の事前準備やインストール方法、本ツールが提供する各機能の基本的な使い方について説明しています。

3. 動作環境

本ツールは以下の環境で動作するように設計されています。
事前に必要な条件が整っている事を確認した上で本ツールのインストールを行ってください。

・ 動作対象 OS
  - Windows Vista (32 bit / 64 bit)
  - Windows 7 (32 bit / 64 bit)
  - Windows 8 (32 bit / 64 bit)
  - Windows 8.1 (32 bit / 64 bit)

・ ハードウェアスペック
  - OS の動作に支障がないこと
  - Eclipse を使用して Android アプリのビルドが支障なく行えること
  - メモリ4GB 以上を推奨

・ Eclipse
  本ツールは、Eclipse Foundation サイトで配布されている Eclipse、および Android Developers サイトで配布されている ADT バンドル版の Eclipse の両方で動作します。
  - Eclipse Foundation サイトで配布されている版
    - Juno Packages (v 4.2.0) 以降
  - Android Developers で配布されている版(ADT バンドル)版
    - Release 4.2.0 以降

・ 必要な SDK、ツール等

本ツールは Google が提供している、Android アプリ開発用のツールや SDK を内部的に使用しています。本ツールを使用する前に以下をインストールしておく事が必要です。

- JDK（Java Development Kit） 7 以上
- ADT（Android Development Tools） 22.3 以上
- Android SDK（Software Development Kit）

なお、これらのインストール手順については、「4 インストール」の中で説明します。

4. インストール

本ツールをインストールするために必要となる作業について説明します。

4.1. インストール手順の概要

本ツールをインストールする手順は以下のとおりです。

1. 本ツールのダウンロードとインストール
2. Eclipse の起動・インストールの確認
3. 環境設定

4.2. ツールのダウンロードとインストール

1. ツールをインストールする前に Eclipse が起動しているかどうか確認し、起動している場合は、終了します。
2. 提供サイトより zip ファイル（ancole.zip）をダウンロードし、PC の任意の場所に展開します。
3. 展開したファイルの中の利用規約を読み、同意できる場合は、同フォルダにある名称が jp.go.ipa.android.security.learning_から始まる jar ファイルを、「dropins」フォルダの中に格納します。「dropins」フォルダは、Eclipse 本体と同じ階層にあります。

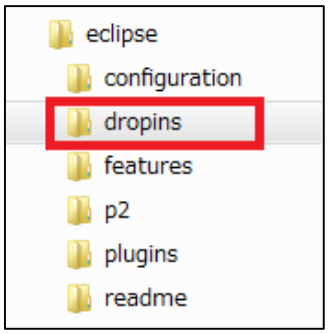

図 1 dropins フォルダ

4.2.1. Eclipse の起動・インストール確認

プラグインファイルを「dropins」フォルダに配置した後、Eclipse を起動します。
インストールに成功するとEclipse 起動時にWelcome 画面が表示され、「AnCoLe（アンコール）」が表示されます。

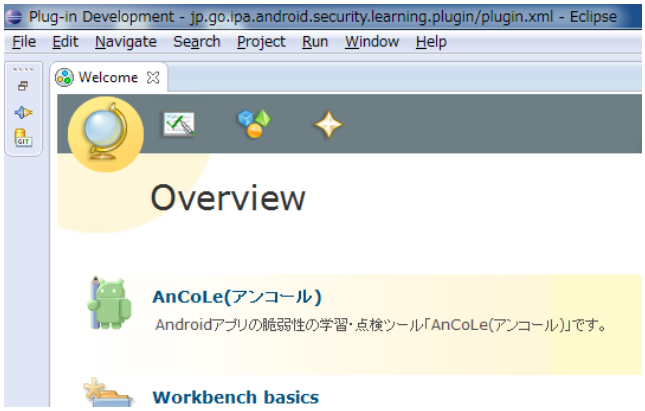

図 2 Welcome 画面

なお、Google が配布している ADT バンドルバージョンの Eclipse では Welcome 画面にはプラグインの情報は表示されません。また、Java パースペクティブを開いた時に Eclipse のツールバー上に「鉛筆」と「虫眼鏡」のボタンが追加されています。
これらのボタンが表示されていればインストールは成功です。

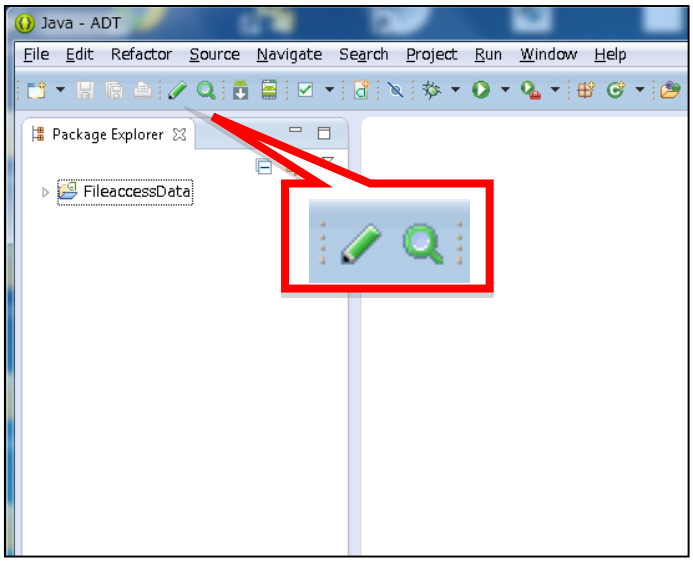

図 3 ツールバーのボタン

4.3. 環境設定

本ツールが提供する学習コンテンツでは、脆弱性を体験する目的でサンプルの Android アプリをインストールします。本ツールからサンプルアプリのインストール、ビルドを行えるように、下記の環境設定・構築を行ってください。

4.3.1. JDK のインストール

Oracle のサイトから JDK（Java Development Kit）をダウンロードし、インストールします。
http://www.oracle.com/technetwork/java/javase/downloads/index.html
Android SDK を使用するには JRE では不十分です。必ず JDK をインストールしてください。

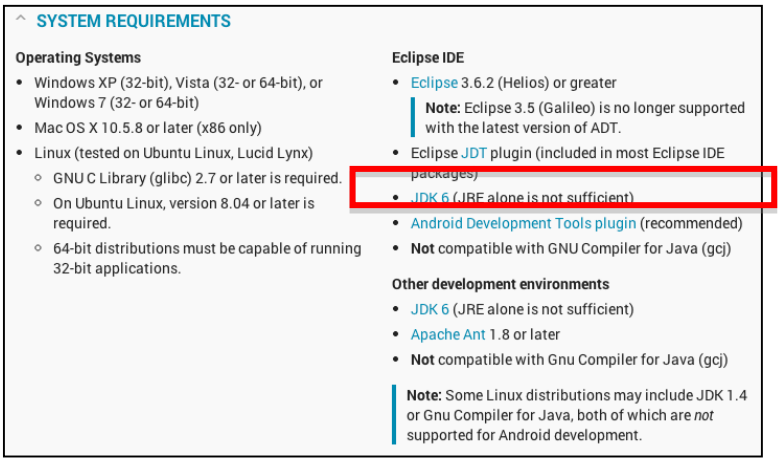

図 4 動作環境の記述

参考：http://developer.android.com/sdk/index.html

4.3.2. ADT（Android Development Tools）のインストール

Android Developers サイトで配布されている ADT バンドル版の Eclipse を使用している場合は、この手順は必要ありません。
Eclipse Foundation サイトで配布されている Eclipse を使用している場合は、Android Developer サイトで配布されている ADT をダウンロードしてインストールしてください。
インストールの手順については以下を参照してください。

［参考］Installing the Eclipse Plugin
http://developer.android.com/sdk/installing/installing-adt.html

4.3.3. Android SDK（Software Development Kit）のインストール

Eclipse ツールバーの SDK ボタンをクリックし、Android SDK Manager を起動します。

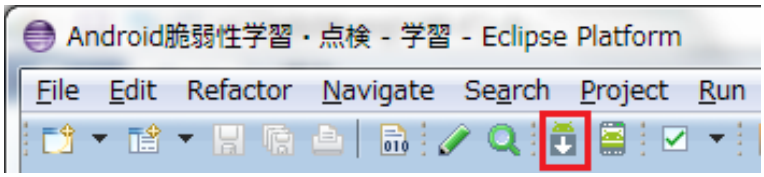

図 5 SDK マネージャの起動

インストール可能な SDK の一覧が表示されます。
サンプルアプリは、Android 2.2(API 8)、と Android 2.3.3(API 10)を使用します。必要となる API レベルにチェックを入れ、「Install n package...」ボタンをクリックしてください。

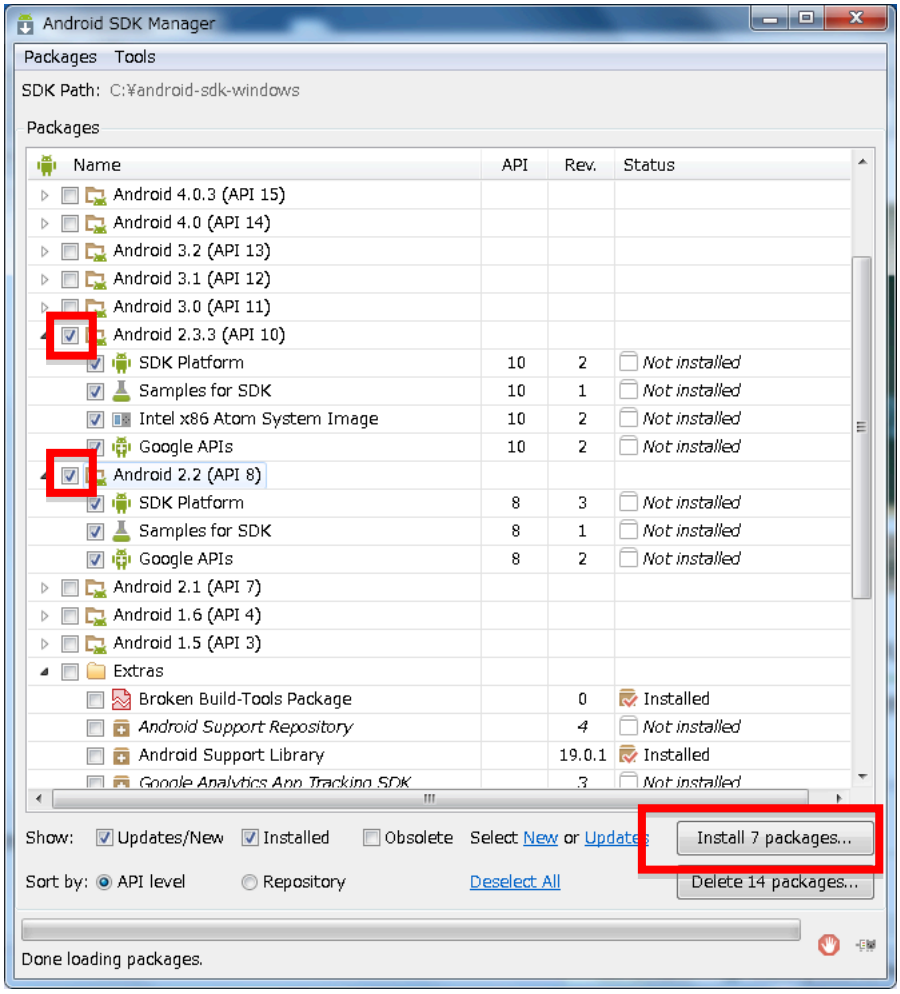

図 6 SDK(API レベル)の選択

ライセンスを読んで頂き、同意する場合「Accept License」にチェックを入れ、「Install」ボタンをクリックしてください。

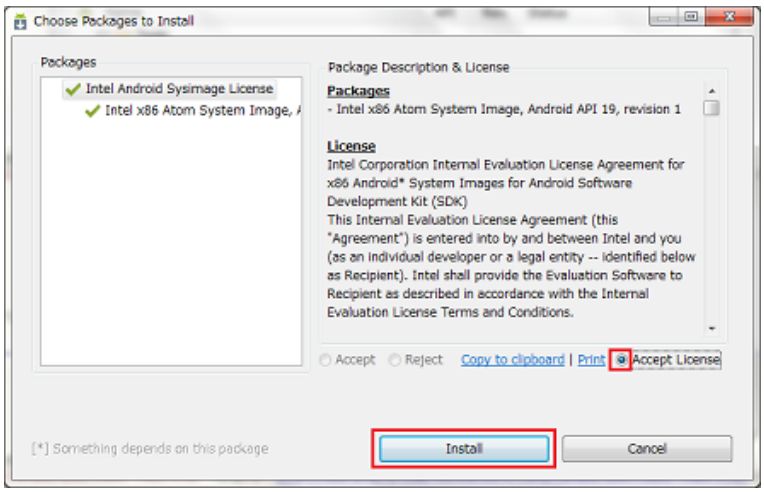

図 7 SDK のインストール

インストールが開始され、インストール状況がプログレスバーにて表示されます。

※インストールには時間がかかる場合があります。

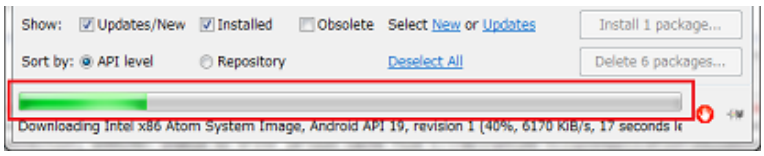

図 8 インストール中の状態

「Done loading packages.」と表示されればインストールは完了です。

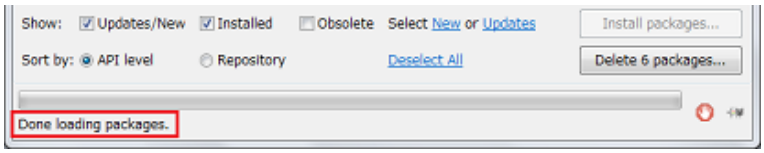

図 9 インストール完了時の状態

4.3.4. adb コマンドの path 設定

Windows の「スタート」をクリック → 「コンピュータ」を右クリック → 「プロパティ」を選択 → 「システムの詳細設定」をクリック → 「環境変数」ボタンをクリックし、環境変数の設定画面を表示します。
ユーザー環境変数」の項目の中から「Path」を選択して、「編集」ボタンをクリックします。
(「Path」が存在しない場合は「新規」ボタンをクリックします。
「変数値」の末尾に Android SDK を保存しているフォルダの platform-tools フォルダまでのパスを追加します。
例:Android SDK を C ドライブのルートに保存した場合
C:¥android-sdk-windows¥platform-tools
(追加するパスの前に「;」が無い場合はそれも追加するようにしてください。)

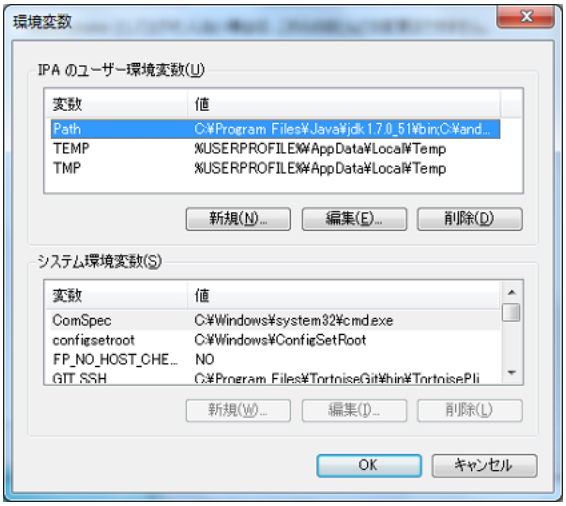

図 10 環境変数の設定画面

5. 本ツールの起動方法

本ツールを起動する方法は2つあります。

1. ツールバーのボタンから起動する方法

ツールバーに表示されている、「鉛筆」ボタン、または「虫眼鏡」ボタンをクリックする事で本ツールを起動することができます。

・学習機能を起動する場合

図 11 ツールバーからの学習機能の起動

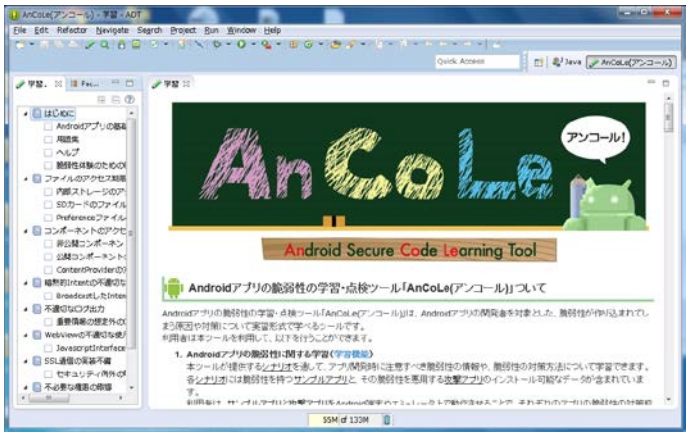

図 12 学習機能先頭ページ

・点検機能を起動する場合

図 13 ツールバーからの点検機能の起動

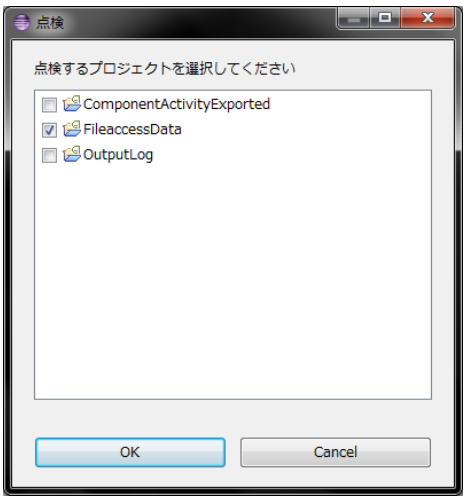

図 14 点検機能(プロジェクト選択画面)

2. Eclipse の Windows メニューから起動する方法

Windows メニューのメニューアイテムから起動することもできます。

学習機能を起動する場合

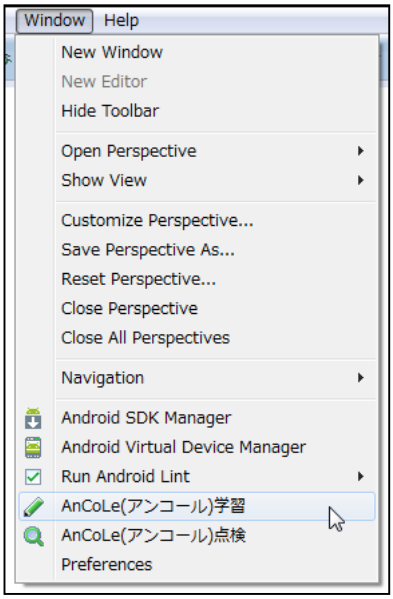

図 15 メニューからの
学習機能の起動

点検機能を起動する場合

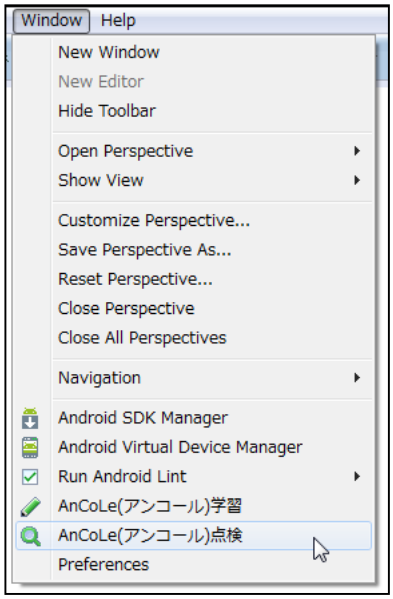

図 16 メニューからの
点検機能の起動

6. 本ツールが提供する機能

本ツールは、大きく2つの機能、「学習機能」と「点検機能」を提供します。また、利用者が円滑に学習を進められるように支援する機能も用意しています。

6.1. 学習機能とは

本ツールが提供するシナリオを通して、アプリ開発時に注意すべき脆弱性の情報や、脆弱性の対策方法について学習できます。 各シナリオには脆弱性を持つサンプルアプリと、その脆弱性を悪用する攻撃アプリのインストール可能なデータが含まれています。

利用者は、サンプルアプリと攻撃アプリを Android 端末やエミュレータ上で動作させることで、それぞれのアプリの脆弱性の対策前と対策後の挙動を体験できます。また、利用者はシナリオに従ってサンプルアプリを修正し、脆弱性の対策方法について学習できます。

図 17 学習機能の画面例

6.2. 点検機能とは

自分が開発したアプリのプロジェクト(ソースコードや設定ファイル)に対して、脆弱性となり得る問題箇所がないか点検することができます。点検結果を基に脆弱性対策の学習をしたり、Android アプリのソースコードを呼び出して修正したりすることができます。

また、修正したプロジェクトを再点検して問題が解決されたことを確認することもできます。

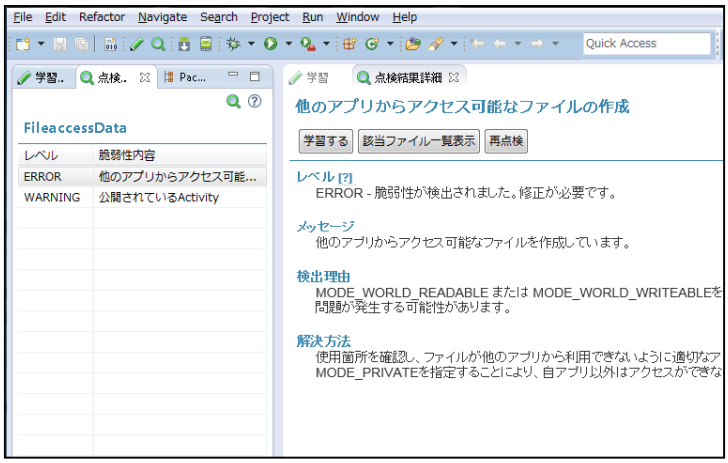

図 18 点検機能の画面例

6.3. 利用者を支援する機能

1. Android アプリの基礎知識・用語集

Android アプリの開発経験が十分でない開発者に向けて、Android OS、および Android アプリの特徴、アプリを開発する上での基本的な知識や、学習中に参照できる用語集を用意しています。

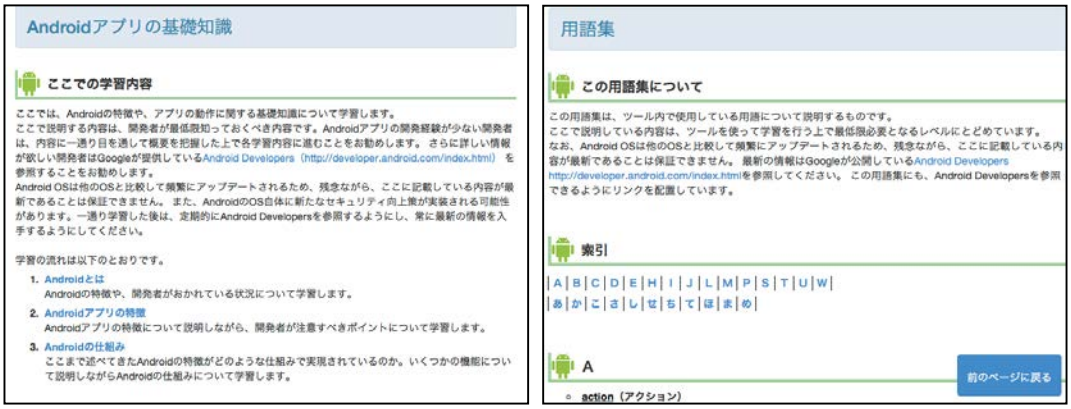

図 19 Android アプリの基礎知識・用語集

2. ヘルプ

本ツールを利用する際に適宜参照できる、ヘルプ情報を提供しています。

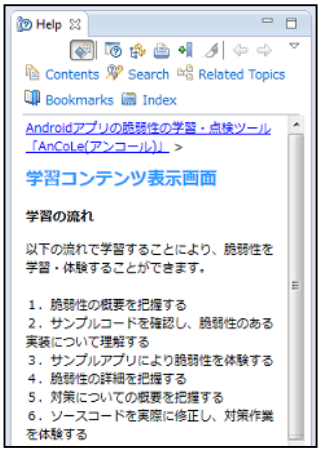

図 20 ヘルプ表示の例

7. 学習機能の使い方

7.1. 学習機能の起動

ツールバー、またはメニューから学習機能を起動します。

7.2. 学習コンテンツの選択

学習機能を使うためには、まず、学習コンテンツ一覧表示画面で学習したいコンテンツを選択します。

ツールの左側のペインに一覧表示されている学習コンテンツから、学習したい内容を選択します。

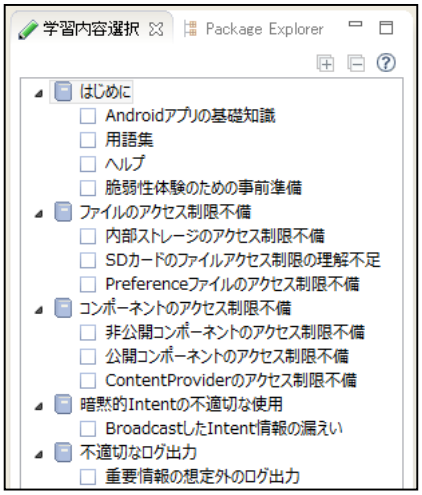

図 21 学習コンテンツ一覧表示画面

選択された学習コンテンツは、学習コンテンツ表示画面に表示されます。

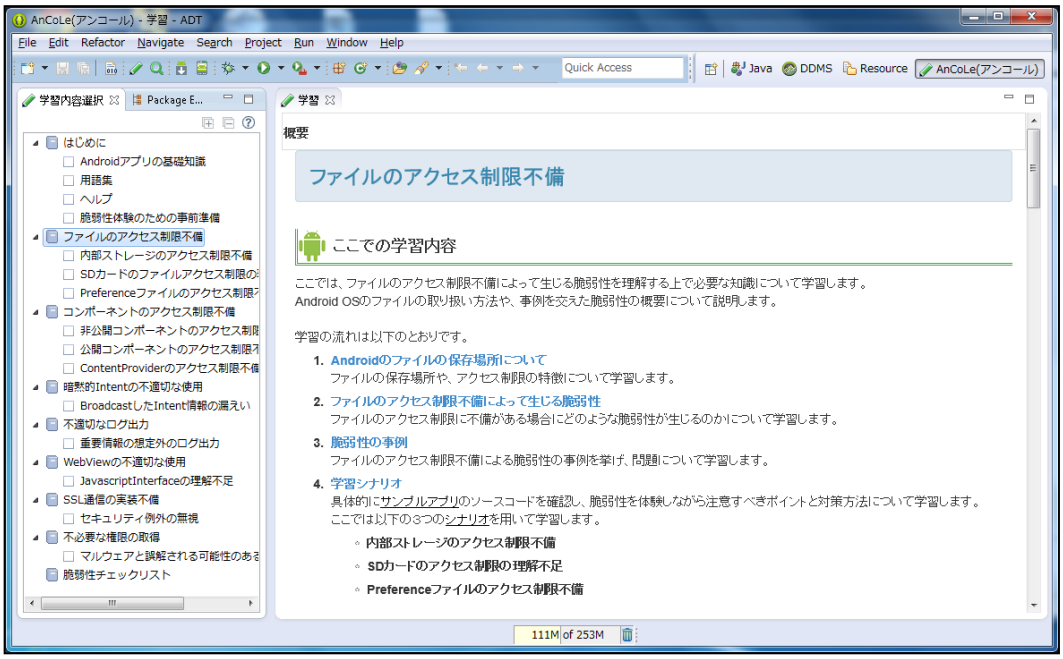

図 22 学習コンテンツ表示画面の例

なお、本ツールでは以下の脆弱性について学習するコンテンツを用意しています。

1. ファイルのアクセス制限不備
2. コンポーネントのアクセス制限不備
3. 暗黙的 Intent の不適切な使用
4. 不適切なログ出力
5. WebView の不適切な使用
6. SSL 通信の実装不備
7. 不必要な権限の取得

7.3. 学習の進め方

コンテンツに記載されている内容を読み進めながら学習して行きます。
学習の流れは以下のとおりです。

1. 脆弱性の概要説明
   学習対象の脆弱性がどのようなものなのか、どのような被害につながるのか学習します。

2. サンプルコードの解説
   どのような実装が脆弱性につながるのか、ソースコードを見ながら学習します。

3. 脆弱性体験
   脆弱性を持っているサンプルアプリをビルドして実行し、それを、攻撃アプリを使って攻撃します。
   実際にアプリを動作させて。脆弱性のあるアプリによってどのような被害に遭うのかを体験します。

4. サンプルアプリの脆弱性説明
   サンプルアプリのソースコードを確認し、脆弱性の原因について理解します。

5. 脆弱性対策方法の説明
   脆弱性の対策方法について学習し、実際にサンプルアプリのソースコードを修正します。

6. 脆弱性対策の体験
   修正されたサンプルアプリを実行し、それを、攻撃アプリを使って攻撃します。
   対策されたことによって、攻撃が成立しないことを確認します。

7.4. コンテンツ内の画面移動

学習コンテンツの内容は複数の画面に分かれています。利用者は、画面上部のリンクをクリックする事でコンテンツ内の各ページへ自由に移動できます。また、画面上部および下部に配置されたボタンを操作する事で、前後のページへ移動できます。

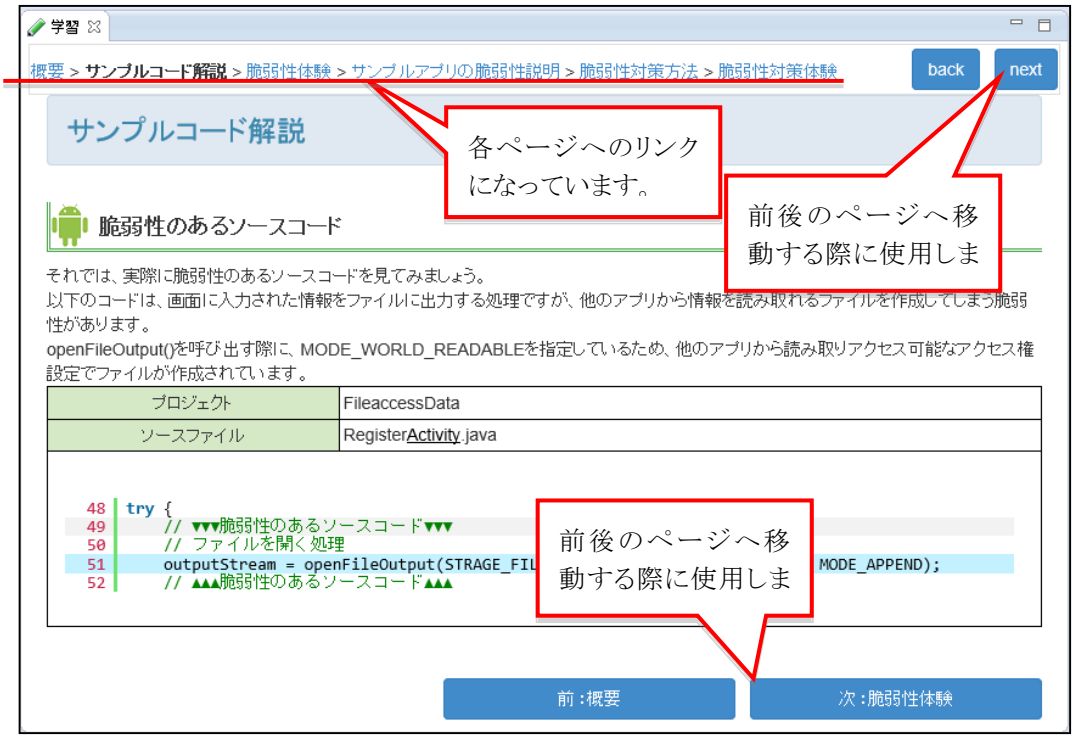

図 23 ページ移動に使用するボタン

7.5. サンプルアプリのインポート

本ツールのシナリオでは、脆弱性を持つサンプルアプリを実際にビルドして実行します。
このサンプルアプリの脆弱性を攻撃するアプリも提供していますので、利用者は脆弱性のあるアプリによって実際にどのような被害に遭うのかについて体験できます。
学習コンテンツ内にある「サンプルアプリをインポート」ボタンをクリックする事で、Eclipse のワークスペースにサンプルアプリがインポートされます。

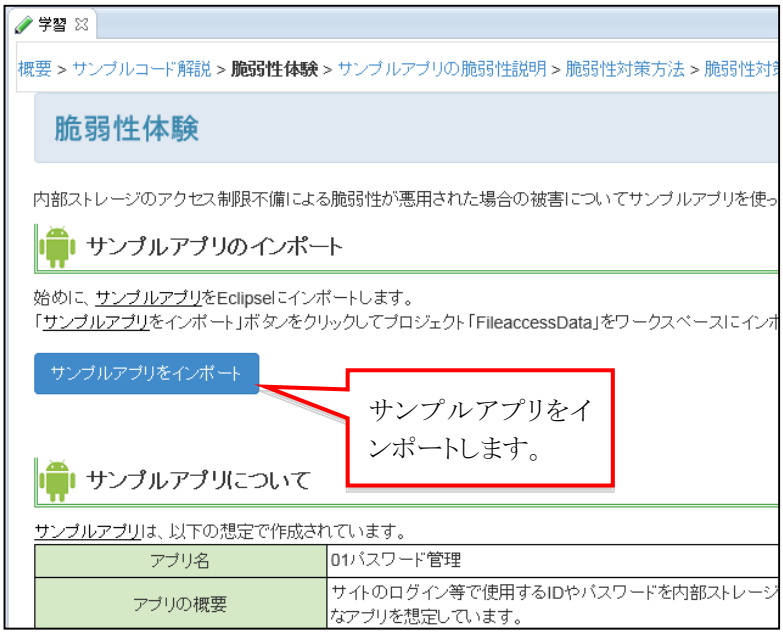

図 24 サンプルアプリのインポートボタン

7.6. 攻撃アプリのダウンロード

同様に「攻撃アプリをダウンロード」ボタンをクリックする事で、攻撃アプリの APK ファイルとインストールを実行するバッチファイル（.bat ファイル）がダウンロードされます。

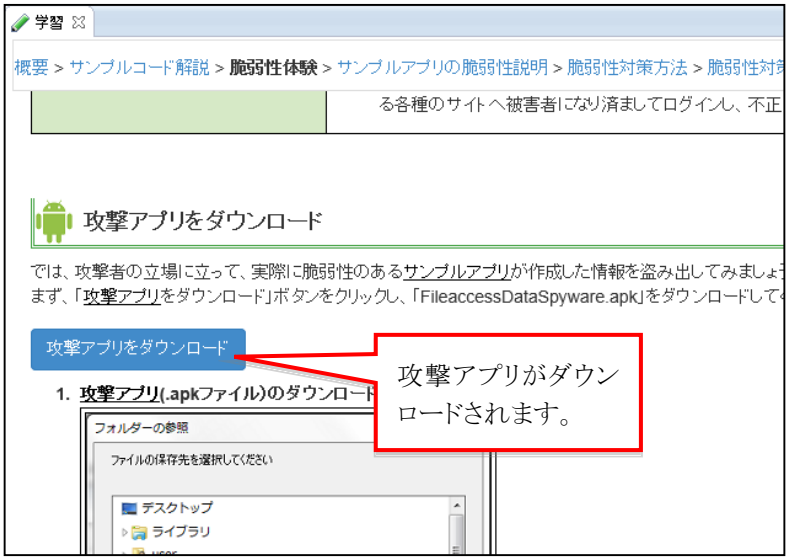

図 25 「攻撃アプリをダウンロード」ボタン

7.7. 対策の体験

サンプルアプリに対して脆弱性対策を実際に行ってみることができます。

「ソースコードを開く」ボタンをクリックする事で、サンプルアプリ内の修正対象ソースコードを開くことができます。

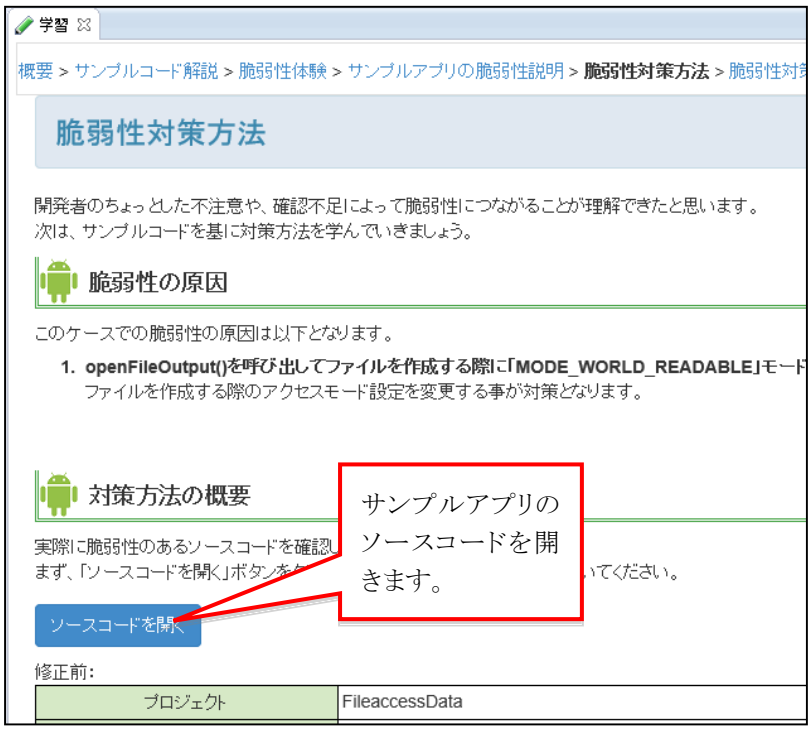

図 26 「ソースコードを開く」ボタン

7.8. 用語集・ヘルプの呼び出し

本ツールには学習コンテンツを読み進めて行く中で、知らない用語があったり、操作方法が分からなくなったりした場合に利用できる、用語集とヘルプが用意されています。

1. 用語集

説明が必要な用語は本文中で下線によってマークされており、用語集へのリンクとなっています。クリックする事で用語集を参照できます。

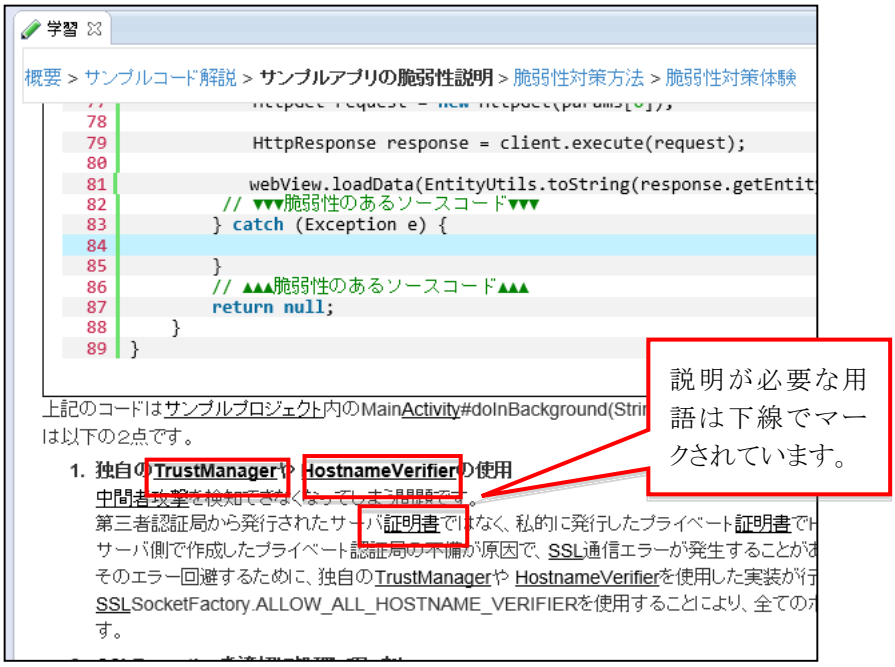

図 27 本文中で用語集へリンクしている例

2. ヘルプ

本ツールの各画面には、画面項目と操作方法について説明しているヘルプが用意されています。

各画面に用意されている「？」ボタンをクリックする、または「F1 キー」を押すことでヘルプを呼び出すことができます。

図 28 ヘルプ呼び出しボタンの例

8. 点検機能の使い方

8.1. 点検機能の起動

ツールバー、またはメニューから点検機能を起動します。

8.2. 点検対象プロジェクトの選択

点検機能を起動すると、プロジェクト選択画面が表示されます。
まず、点検を行う対象となる Android アプリのプロジェクトを選択します

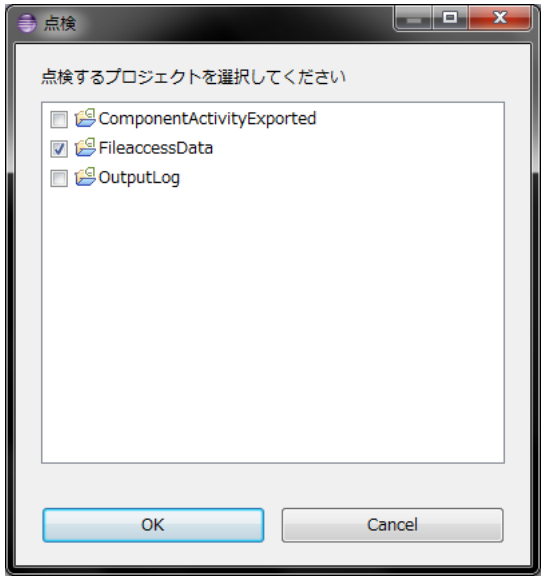

図 29 プロジェクト選択画面

プロジェクトを選択して「OK」ボタンをクリックすると点検が開始され、結果が「点検結果一覧画面」に表示されます。

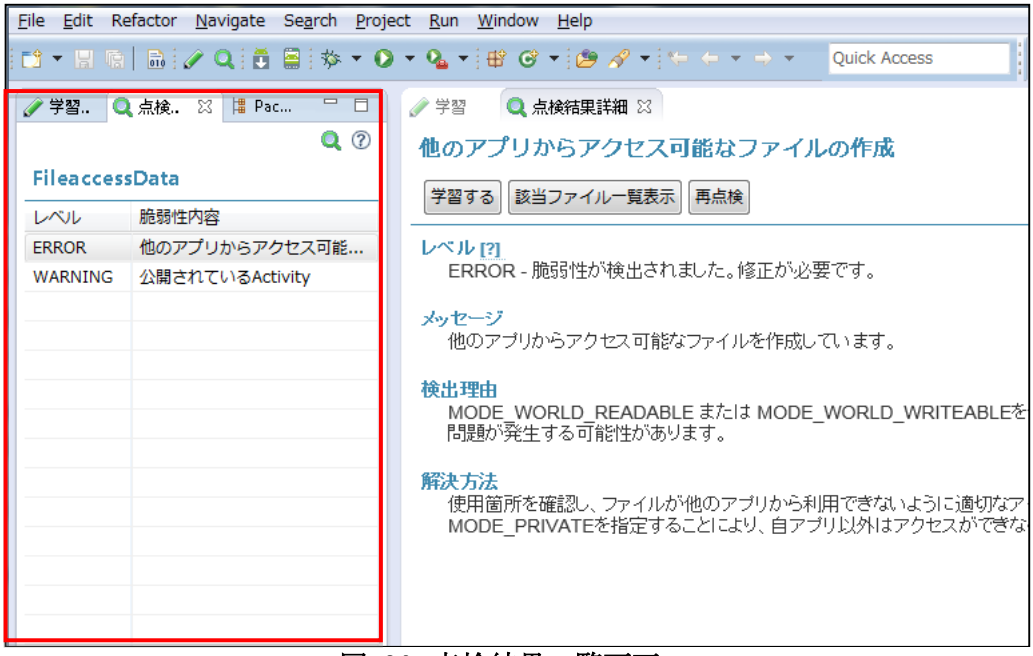

図 30 点検結果一覧画面

8.3. 点検結果の詳細画面

点検結果一覧画面で検出された脆弱性をクリックすると、「点検結果詳細画面」に検出された問題の詳細が表示されます。

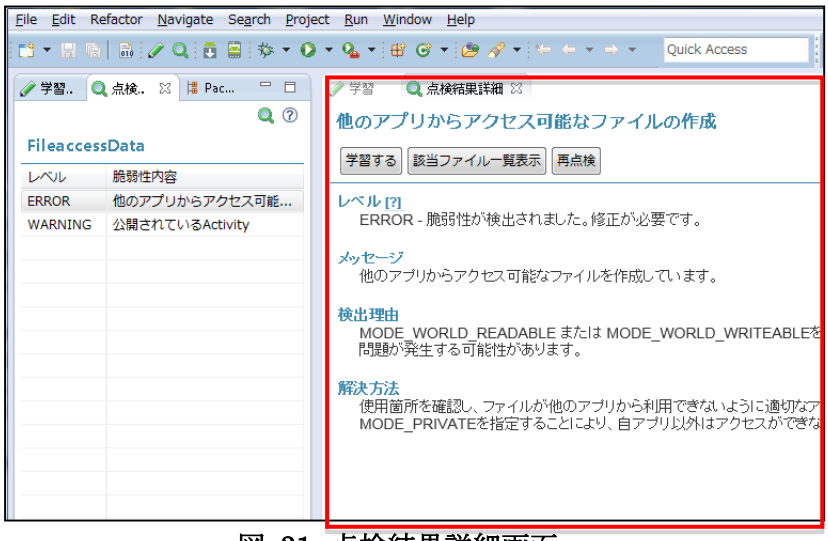

図 31 点検結果詳細画面

利用者はこの画面から、検出された問題の詳細について学ぶために学習コンテンツに移動することができます。また、問題が検出されたソースコードを呼び出して、問題を修正することもできます。

8.4. 学習コンテンツへの移動

点検結果詳細画面に配置されている「学習する」ボタンをクリックする事で、検出された脆弱性に関連する学習項目へ移動することができます。

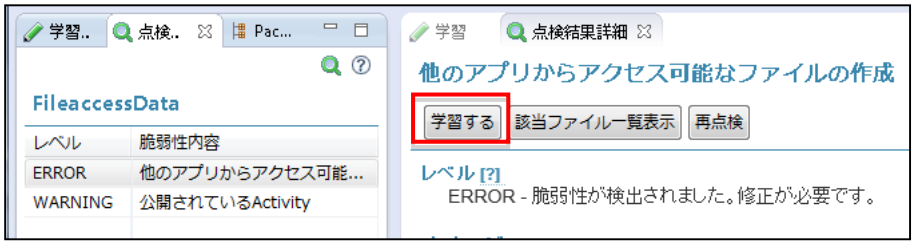

図 32 「学習する」ボタン

8.5. 該当するソースコードの一覧表示

点検結果詳細画面に配置されている「該当ファイル一覧表示」ボタンをクリックする事で、脆弱性が検出されたソースコード一覧が表示されます。

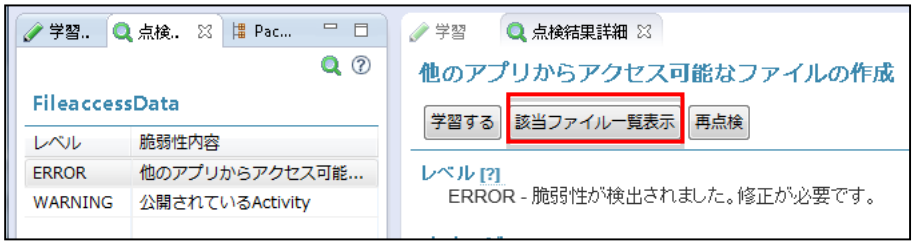

図 33 「該当ファイル一覧表示」ボタン

ファイル一覧には、該当するソースコードのファイル名と、検出された行番号と行の内容のサマリが表示されます。

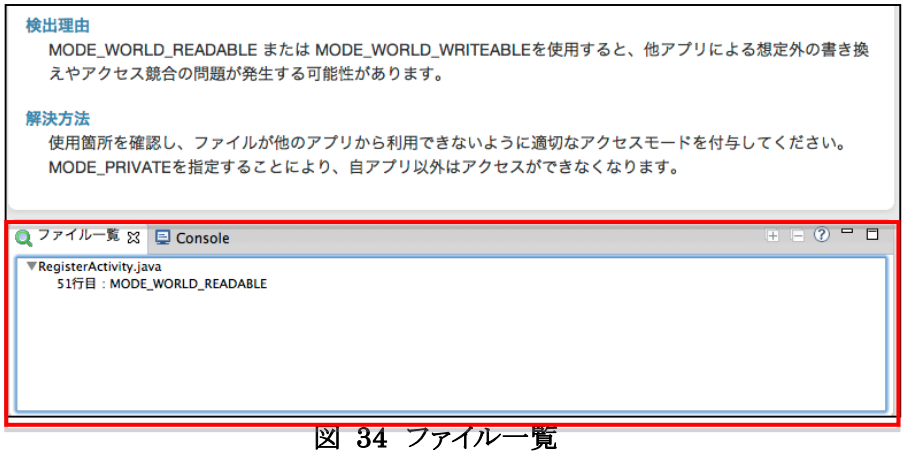

図 34 ファイル一覧

一覧表示されているファイル名、または行番号をクリックすると、該当するソースコードを編集できます。

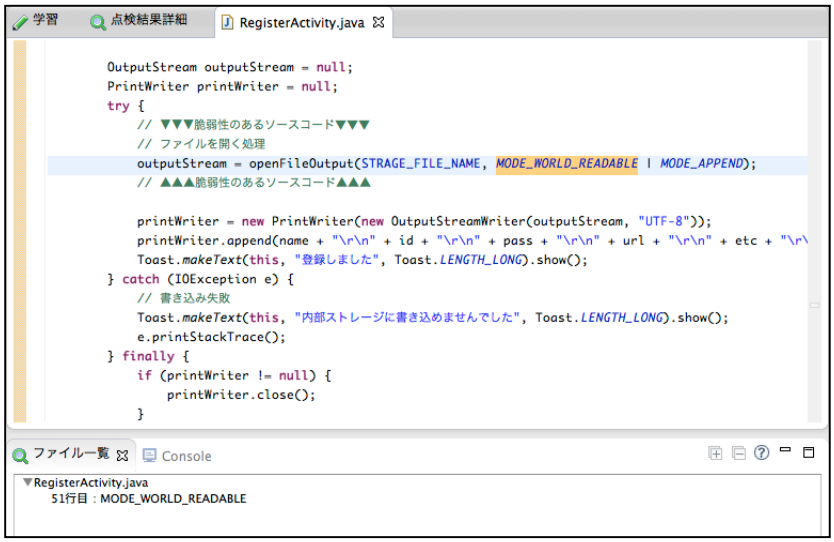

図 35 ファイル一覧からソースコードを開く

この画面でソースコードを編集して脆弱性の対策を行う事ができます。

8.6. 再点検

8.3 点検結果詳細画面から、点検対象のプロジェクトに対して再点検を行うことができます。

図 36 「再点検」ボタン

別のプロジェクトに対して点検を行う場合は、左側ペインの点検タブに配置されている「虫眼鏡」のボタンをクリックします。

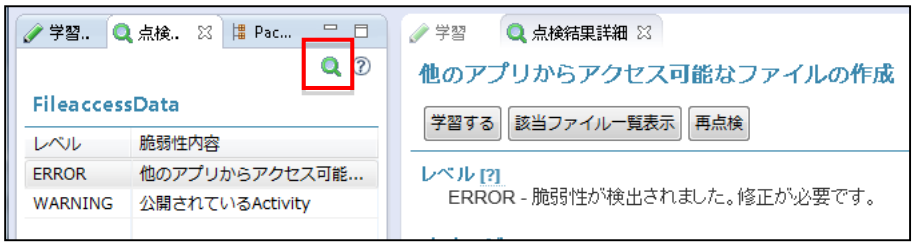

図 37 他のプロジェクトを点検する

9. アンインストール

9.1. アンインストール手順

9.1.1. AnCoLe（アンコール）のアンインストール

Eclipse が起動している場合は終了します。

Eclipse インストールフォルダの直下の「dropins」フォルダ内に配置されているプラグインファイルを削除します。

9.1.2. アンインストールの確認

プラグインファイルを削除した後、Eclipse を再度起動します。

アンインストールに成功すると、Eclipse を開いた際にツールバー上の「鉛筆」「虫眼鏡」のボタンが表示されなくなります。

9.1.3. パースペクティブに関するエラーへの対処

Eclipse 上で本ツールのパースペクティブ「AnCoLe（アンコール）」開かれている状態で、アンインストールを行った場合、本ツールのパースペクティブが Eclipse 上に残ったままになり、エラーを表示する場合があります。

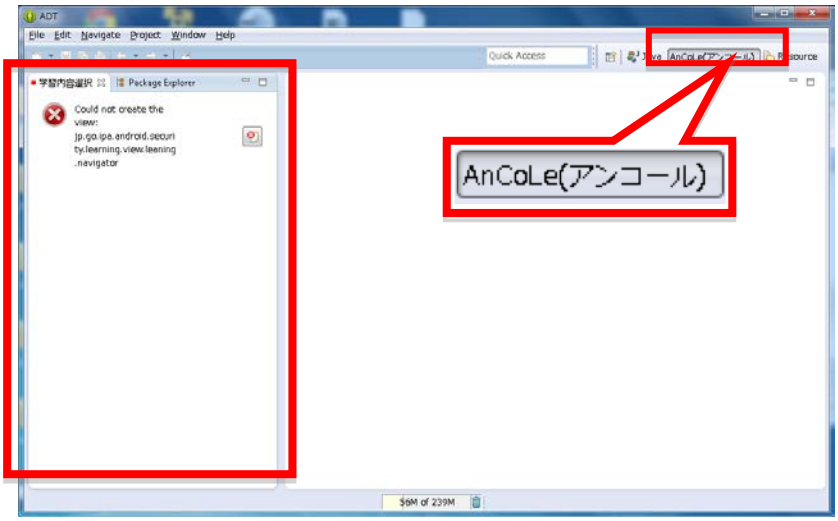

図 38 パースペクティブのエラー表示

この場合は、パースペクティブ上で「右クリック」→「close」で閉じてください。Eclipse の次回起動からはエラーメッセージは表示されなくなります。

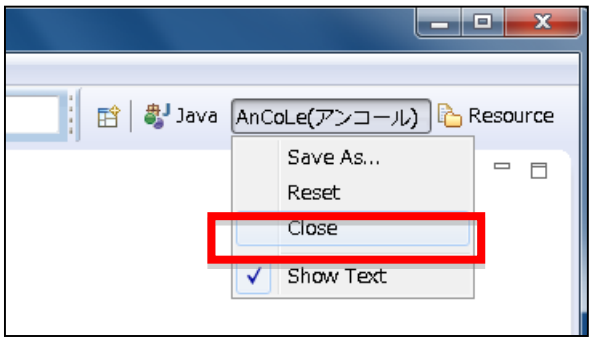

図 39 パースペクティブを閉じる

9.1.4. 本ツールが保存するファイルについて

本ツールでは学習状態を設定ファイルに保持しています。この設定ファイルは本ツールをアンインストールした後でも削除されずに PC 内に残ります。
このファイルはプラグイン毎の設定ファイルなので、情報が残っていても他のプラグインの動作に影響を与えることはありませんが、手動で削除することも可能です。
手動で削除する場合、本ツールをアンインストール後、ワークスペース配下の
「.metadata/plugins/org.eclipse.core.runtime/.settings」フォルダ内にある、
「jp.go.ipa.android.security.learning.prefs」ファイルを削除してください。

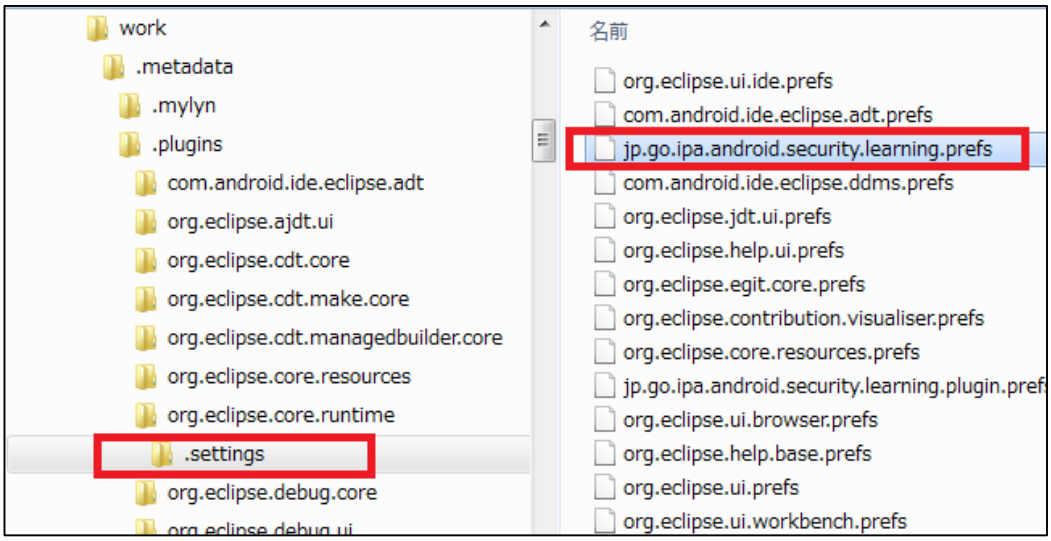

図 40 学習状態設定ファイル

以上