Checker Framework Tutorial

Previous Validating User Input

Finding a Security Error

This example uses the Tainting Checker to verify that user input does not contain SQL statements, thus preventing SQL injection. (If you have not already done so, download the tutorial sourcefiles.)

Outline

1. Run the Tainting Checker — 1 error found
2. Correct the error
3. Re-run the Tainting Checker — a new error is found
4. Correct the new error
5. Re-run the Tainting Checker — no errors

1. Run the Tainting Checker — 1 error found

Run the buildfile:
(The Ant buildfile makes use of the Checker Framework support for Ant.)

```
$ cd personalblog-demo
$ ant
Buildfile: .../personalblog-demo/build.xml

clean:

check-tainting:
    [mkdir] Created dir: .../personalblog-demo/bin
[jsr308.javac] Compiling 2 source files to .../personalblog-demo/bin
[jsr308.javac] javac 1.8.0-jsr308-3.5.0
[jsr308.javac]
.../personalblog-demo/src/net/eyde/personalblog/service/PersonalBlogService.java:17
5: error: incompatible types in argument.
[jsr308.javac]                      "where post.category like '%", category,
[jsr308.javac]                                                    ^
[jsr308.javac]   found   : @Tainted String
[jsr308.javac]   required: @Untainted String
[jsr308.javac] 1 error

BUILD FAILED
.../personalblog-demo/build.xml:35: Compile failed; see the compiler error output
for details.

Total time: 2 seconds
```

The checker issues an error for `getPostsByCategory()` because the possibly-tainted string `category` is used in the query construction. This string could contain SQL statements that could taint the database. The programmer must ensure that `category` does not contain malicious SQL code.

2. Correct the Error

To correct this error, **add `@Untainted`** to the type of the `category` parameter.

```
     public List<?> getPostsByCategory(@Untainted String category) throws
ServiceException {
```

This forces clients to pass an `@Untainted` value, which was the intention of the designer of the `getPostsByCategory` method.

3. Re-run the Tainting Checker — a new error is found

Run the Tainting Checker again.

```
$ ant
Buildfile: .../personalblog-demo/build.xml

clean:
   [delete] Deleting directory .../personalblog-demo/bin

check-tainting:
    [mkdir] Created dir: .../personalblog-demo/bin
[jsr308.javac] Compiling 2 source files to .../personalblog-demo/bin
[jsr308.javac] javac 1.8.0-jsr308-3.5.0
[jsr308.javac]
.../personalblog-demo/src/net/eyde/personalblog/struts/action/ReadAction.java:58:
error: incompatible types in argument.
[jsr308.javac]                          pblog.getPostsByCategory(reqCategory));
[jsr308.javac]                                                    ^
[jsr308.javac]   found   : @Tainted String
[jsr308.javac]   required: @Untainted String
[jsr308.javac] 1 error

BUILD FAILED
.../personalblog-demo/build.xml:35: Compile failed; see the compiler error output
for details.

Total time: 2 seconds
```

There is an error in `ReadAction.executeSub()`, which is a client of `getPostsByCategory`. The `reqCategory` is accepted from the user (from request object) without validation.

4. Correct the New Error

To correct, **use the `validate` method** as shown below.

```
    String reqCategory = validate(cleanNull(request.getParameter("cat")));
```

5. Re-run the Tainting Checker — no errors

There should be no errors.

```
$ ant
Buildfile: .../personalblog-demo/build.xml

clean:
   [delete] Deleting directory .../personalblog-demo/bin
```

```
check-tainting:
    [mkdir] Created dir: .../personalblog-demo/bin
[jsr308.javac] Compiling 2 source files to .../personalblog-demo/bin
[jsr308.javac] javac 1.8.0-jsr308-3.5.0

BUILD SUCCESSFUL
Total time: 2 seconds
```

You are done with the personalblog-demo project, so return to the parent directory

```
$ cd ..
```

For a complete discussion of how to use the Tainting Checker, please read the Tainting Checker chapter in the Checker Framework manual.

Next, try Writing an Encryption Checker or return to the main page of the Tutorial.