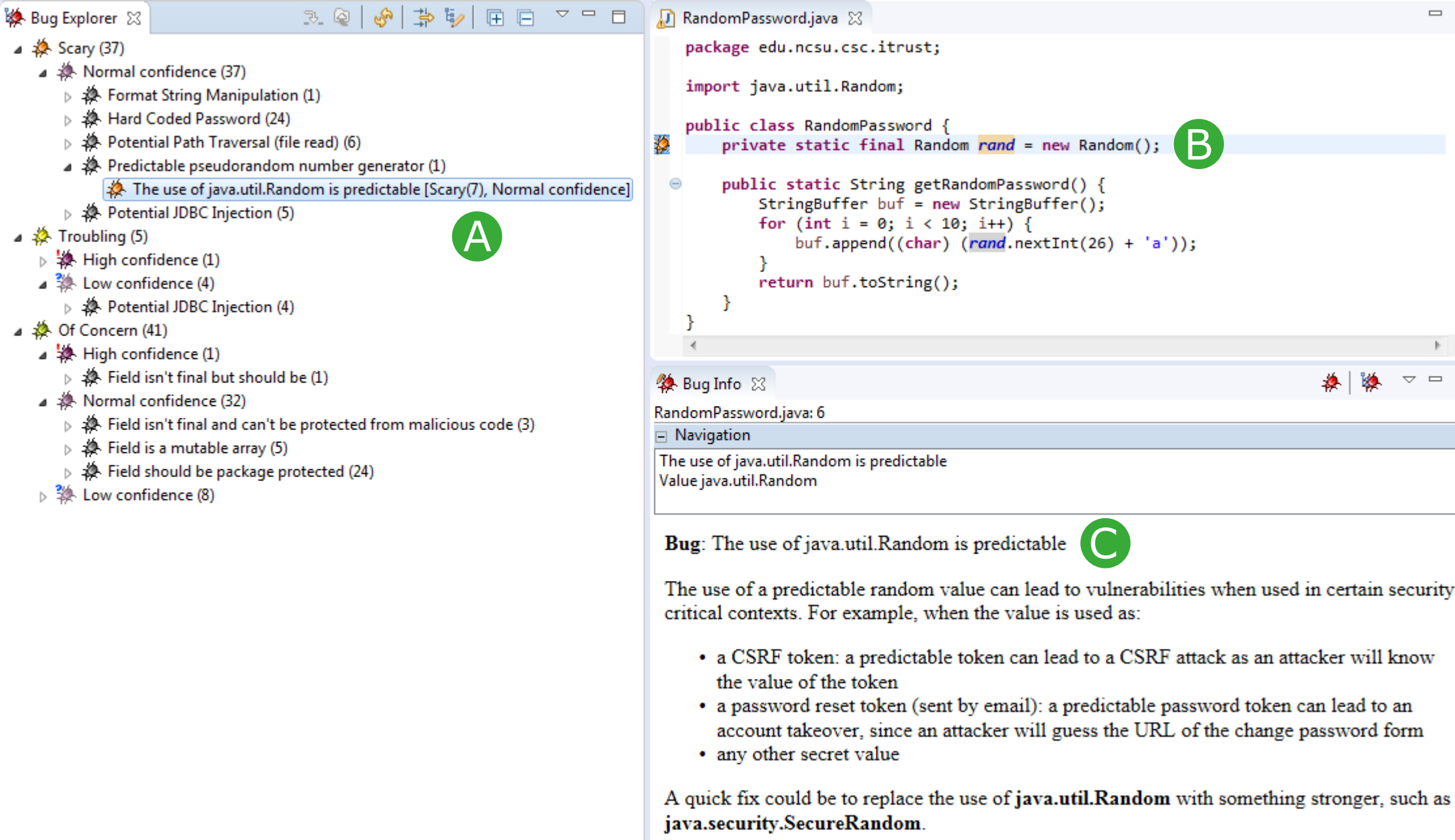
Bug Explorer
Scary (37)
Normal confidence (37)
Format String Manipulation (1)
Hard Coded Password (24)
Potential Path Traversal (file read) (6)
Predictable pseudorandom number generator (1)
The use of java.util.Random is predictable [Scary(7), Normal confidence]
Potential JDBC Injection (5)
Troubling (5)
High confidence (1)
Low confidence (4)
Potential JDBC Injection (4)
Of Concern (41)
High confidence (1)
Field isn't final but should be (1)
Normal confidence (32)
Field isn't final and can't be protected from malicious code (3)
Field is a mutable array (5)
Field should be package protected (24)
Low confidence (8)
A
RandomPassword.java
package edu.ncsu.csc.itrust;
import java.util.Random;
public class RandomPassword {
private static final Random rand = new Random();
public static String getRandomPassword() {
StringBuffer buf = new StringBuffer();
for (int i = 0; i < 10; i++) {
buf.append((char) (rand.nextInt(26) + 'a'));
}
return buf.toString();
}
}
B
Bug Info
RandomPassword.java: 6
Navigation
The use of java.util.Random is predictable
Value java.util.Random
C
Bug: The use of java.util.Random is predictable
The use of a predictable random value can lead to vulnerabilities when used in certain security critical contexts. For example, when the value is used as:
a CSRF token: a predictable token can lead to a CSRF attack as an attacker will know the value of the token
a password reset token (sent by email): a predictable password token can lead to an account takeover, since an attacker will guess the URL of the change password form
any other secret value
A quick fix could be to replace the use of java.util.Random with something stronger, such as java.security.SecureRandom.