NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO:

You are asked to review a utility method written for a web application. The method, `setDate`, changes the date of the server. It takes a `String` as the new date (“dd-mm-yyyy” format), attempts to change the date of the server, and returns `true` if it succeeded, and false otherwise. Consider the snippet of code below (assuming the code runs on a Windows operating system) and answer the following questions, assuming that the code has all required permissions to execute.

```
01   // OMITTED: Import whatever is needed
02   public final class SystemUtils {
03     public static boolean setDate (String date)
04         throws Exception {
05       return run("DATE " + date);
06     }
07
08     private static boolean run (String cmd)
09         throws Exception {
10       Process process = Runtime.getRuntime().exec("CMD /C " + cmd);
11       int exit = process.waitFor();
12
13       if (exit == 0)
14         return true;
15       else
16         return false;
17     }
18   }
```

Questions:

1. What will the `setDate` method do when executed?

2. If a program calls the `setDate` method with an arbitrary `String` value as the new date, which one statement is correct?

a. If the given `String` value does not conform to the “dd-mm-yyyy” format, an exception is thrown.
b. The `setDate` method cannot change the date.
c. The method might do more than change the date.
d. The return value of the `waitFor` method is not interpreted correctly (lines 14~17).
e. The web application will crash.

[Other statistical questions will be imported here while creating survey.]

NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.

ANSWER:

c

The method above lets the argument be almost anything which makes it vulnerable to command injection attack. For example calling the setDate method with "`10-12-2015 && shutdown /s`" as the argument changes the date and then turns off the server. Either the argument has to be sanitized or its type should be Java `Date` class.

NOTE: THE REST OF THIS DOCUMENT CONTAINS EXTRA INFORMATION FOR THE PROJECT RESEARCHERS. IT IS NOT TO BE SHOWN TO THE PUZZLE TAKERS.

TAGS:

java, runtime.exec, command-injection, input-verification, input-sanitization

CATEGORIES:

Blindspot - YES
Type - Command Injection
Number of distinct functions - 4
Number of total functions - 4
Blindspot function - `Runtime.exec()`
Function call omitted - NO
Blindspot type - Incorrect usage
Number of Parameters in the blindspot function - 1 parameter
Cyclomatic complexity - 3

NAME:

Runtime class, exec method - Executes the specified string command in a separate process.

DESCRIPTION:

Every Java application has a single instance of class `Runtime` that allows the application to interface with the environment in which the application is running. The current runtime can be obtained from the `getRuntime` method. The `exec` method executes the specified command and arguments in a separate process.

BLINDSPOT:

The `exec` method is highly vulnerable to command injection attack and programmers should avoid using it, if possible. If there is no way to avoid, the data passed to the method should be carefully verified and sanitized.

CORRECT USE EXAMPLE:

#N/A

MORE INFORMATION:
#N/A

REFERENCES:

1. Runtime
2. OWASP: Command Injection