NOTE: SCENARIO IS WHAT PUZZLE TAKER SEES.

SCENARIO:

You are developing a system which deals with lots of numbers. You have a utility class which helps with formatting of numbers. The `format` method takes a `String input`, and inserts a record into the `log` file, if the `input` is not a number. Consider the snippet of code below 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 NumberUtils {
03     public static void format (String input)
04         throws Exception {
05       try{
06           int value = Integer.parseInt(input);
07       }
08       catch(NumberFormatException e){
09           log.info("Failed to parse input: " + input);
10       }
11
12       // OMITTED: Further processing.
13       // OMITTED: Close all resources properly.
14     }
15   }
16
17
```

Questions:
1. What will the `format` method do when executed?

2. If the `format` method gets called, which one statement is correct?

a. It parses all the inputs successfully to an integer value and does not logs anything.
b. It only logs the inputs for which the `parseInt` method throws `NumberFormatException`.
c. Depending on the value of the input, the method may log some other messages as well.
d. It log at least one message for every execution.
e. None of the above.

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

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

ANSWER:

c
If the input is passed as `twenty-one%0a%0aINFO:+User+logged+out%3dadmin`, then the method logs the following:

```
INFO: Failed to parse val=twenty-one
INFO: User logged out=admin
```

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

TAGS:
java, log-injection, input-verification, input-sanitization

CATEGORIES:
Blindspot - YES
Type - Log injection
Number of distinct functions - 2
Number of total functions - 2
Blindspot function - `info`
Function call omitted - No
Blindspot type - Input validation
Number of Parameters in the blindspot function - 1
Cyclomatic complexity - 3

NAME:
Logger class is available as a Java API as well as an implementation through other Java libraries.

DESCRIPTION:
Log messages are rarely verified. Although most of the inputs to the logging methods do not come from an untrusted source, there can still be a few inputs which might contain malicious content. Thus, it is necessary to verify the inputs to the logging methods if they are from an untrusted source.

BLINDSPOT:
Log injection can be performed when the input to the methods are not verified and can cause a blindspot. The info method in this case just adds the input string to the log file. However, the input string can contain another log message, which would be appended in the log file.

CORRECT USE EXAMPLE:
#NA

MORE INFORMATION:
#N/A

REFERENCES:
1. https://www.owasp.org/index.php/Log_Injection