NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO:

Consider the following snippet of Python code that searches for a particular file in a file system, for which a path is entered by the user. The code uses the `os.path.exists(path)` function, which returns `true` if the string passed as the parameter refers to an existing path, and `false` otherwise. Consider the snippet of code below and answer the following questions, assuming that the code has all required permissions to execute.

```
01   import os
02
03   print "For which pathname do you want to search?"
04   # The user enters a file pathname to search
05   path = input('> ')
06   if os.path.exists(str(path)):
07     print path, "pathname exists."
08   else:
09     print path, "pathname does not exist."
```

Questions:

1. What will the program do when executed?

2. What will happen if the user enters the string "`os.system("date")`" when prompted for the file pathname?

a. The current date will be displayed on the terminal.
b. The program will print "`pathname does not exist.`"
c. The program will crash with error message "`invalid input`".
d. The program will crash with no error message.
e. None of the above.

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

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

ANSWER:

1. The program reads a line from the standard input and evaluates it as a Python expression. Then it prints "`0 pathname exists,`" if there is a file named "`0`" in the current directory.

Otherwise, it prints "`0 pathname does not exist.`" (Note: the 0 value is the return value of the `os.system` function when executed, which generally is system-dependent, meaning it might return another value instead of 0.) By importing the `os` package, the user can virtually execute any system command using `os.system` function.

2. a
Having `print` as a statement (not as a function) implies the code is written in Python version 2.x. As such, the `input` function is equivalent to `eval(raw_input(prompt))`, meaning it reads some string from the standard input and then parses and evaluates the given string as a Python expression. Thus the string "`os.system("date")`" will be evaluated as a Python expression and prints the current system date. The return value of the whole expression, an integer value, is stored in the `path` variable.

TAGS:
python, code-evaluation, code-injection

CATEGORIES:
Blindspot - YES
Type - Injection
Number of distinct functions - 3
Number of total functions - 3
Blindspot function - `input()`
Function call omitted - NO
Blindspot type - Validation missing
Number of parameters in the blindspot function - 1 parameter
Cyclomatic complexity - 2

NAME:
input([prompt])

DESCRIPTION:
Equivalent to `eval(raw_input(prompt))`. In the `eval(expr)` function the `expr` argument is parsed and evaluated as a Python expression. The `raw_input(prompt)` function writes the `prompt` argument, if it is present, to standard output without a trailing newline. Then it reads a line from input, converts it to a string (stripping a trailing newline), and returns it.

BLINDSPOT:
The name of the `input` function might be misleading, especially in the presence of another function with the similar name `raw_input`. The `input` function is not meant to read data from standard input. It is a shortcut to read string from standard input and evaluate it as a Python expression equivalent to `eval(raw_input(prompt))`.

CORRECT USE EXAMPLE:

```
import os
print "For which pathnames do you want to search?"
# The user enters a list of file pathnames to search
path = raw_input('list> ')
if os.path.exists(path):
  print path, "pathname exists."
else:
  print path, "pathname does not exist."
```

MORE INFORMATION:
#N/A.

REFERENCES:
1. https://stackoverflow.com/questions/33946678/