NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO:

You are developing a library for filesystem operations. One of the tasks is to implement a recursive version of the `ls` command, which will recursively list all files and directories in the current directory and its subdirectories. In the following implementation of such a feature, the `os.walk` function generates file names in a given directory tree by walking the tree either top-down or bottom-up. Consider the snippet of code below and answer the following questions, assuming that the code has all required permissions to execute.

```
import os
dir = raw_input("Enter the directory: ")
for root, dirs, files in os.walk(dir, followlinks=True):
  for name in files:
    print(os.path.join(root, name))
  for name in dirs:
    print(os.path.join(root, name))
```

Questions:
1. What will the program do once it is executed?

2. What will happen if the directory given by the user exists and contains symbolic links?

a. The program recursively prints the complete path of all files present in `dir` and exits.
b. The function throws an exception with a message upon processing the symbolic links.
c. The program will crash without any message.
d. 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. Program will take as input a directory name, i.e. ‘home,’ and recursively print the complete path of all the files in the given directory.
2. d

The answer depends on to what the symbolic link is pointing. For example, If the current directory contains a link to the parent directory, then a call of the `os.walk` function with `followlinks=True` leads to infinite recursion.

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

TAGS:

Python, denial-of-service

CATEGORIES:
Blindspot - YES
Type - File
Number of distinct functions - 3
Number of total functions - 3
Blindspot function - `walk()`
Function call omitted - NO
Blindspot type - Missing verification
Number of parameters in the blindspot function - 2 parameters
Cyclomatic complexity - 3

NAME:
os.walk()

DESCRIPTION:
Generate the file names in a directory tree by walking the tree either top-down or bottom-up. For each directory in the tree rooted at directory top (including top itself), it yields a 3-tuple `(dirpath, dirnames, filenames)`.

BLINDSPOT:
Setting `followlinks` to `True` in the `os.walk` function can lead to infinite recursion if a link pointing to a parent directory is present in the current directory. The `os.walk` function does not keep track of the directories it already visited. Another possibility is that some other process adds a link to the parent directory (in dir). Then this code will recursively print filenames and cause a DoS attack on the server.

CORRECT USE EXAMPLE:
#N/A

MORE INFORMATION:

#N/A

REFERENCES:

1. https://docs.python.org/3.1/library/os.html#os.walk