Codio Activity

Exploring a simple Python shell

In this session, you will create a command shell in Python, and then run it and answer questions about it. Review the blogs at Praka (2018) and Szabo (n.d.) and then create a CLI/ shell that implements the following:

- When you enter the command LIST it lists the contents of the current directory
- The ADD command will add the following two numbers together and provide the result
- The HELP command provides a list of commands available
- The EXIT command exits the shell

Add suitable comments to your code and add the program to your e-portfolio. Run the shell you have created, try a few commands, and then answer the questions below.

Generated Python code

```
import os

class SampleCLI():

    def list_contents(self):
        """Lists contents of the current directory"""
        try:
            for item in os.scandir(path='.'):
                size = os.stat(item.name).st_size
                print(f"Directory element: {item.name.ljust(60)}   {size} bytes")

        except Exception:
            print("Unable to list contents of the current directory")


    def add_number(self, x, y):
        """Adds two numbers together and displays the end results"""
        try:
            res= x + y
            print(f"Result of {x} and {y}= {res}")
        except Exception:
            print("Error occurred attempting to add two numbers")


    def show_commands(self):
        """Lists all the commands available for selection by an end user"""

        print ("""
            LIST - Shows all commands
            ADD - computes the sum of two numbers
            HELP - shows all available commands
```

```
            EXIT - termintes the application
            """)


def main():
    cli = SampleCLI()

    while True:
        inp = input("> ")
        if inp == "LIST":
            cli.list_contents()
        elif inp == "ADD":
            x = int(input(" 1st Number: "))
            y = int(input(" 2nd Number: "))
            cli.add_number(x, y)
        elif inp == "HELP":
            cli.show_commands()
        elif inp == "EXIT":
            break


if __name__ == "__main__":
    main()
```

Q: What are the two main security vulnerabilities with your shell?

1. The LIST command allows the user to list contents of a directory without performing any access level checks.

2. The ADD command may fail if the user inputs any value that is not convertible to an integer. This results in the program abruptly terminating

```
> ADD
 1st Number: 1.20394

---------------------------------------------------------------------------
ValueError                                Traceback (most recent cal
<ipython-input-1-d1f7bd73bacb> in <module>()
     52
     53 if __name__ == "__main__":
---> 54     main()

<ipython-input-1-d1f7bd73bacb> in main()
     42             cli.list_contents()
     43         elif inp == "ADD":
---> 44             x = int(input(" 1st Number: "))
     45             y = int(input(" 2nd Number: "))
     46             cli.add_number(x, y)

ValueError: invalid literal for int() with base 10: '1.20394'
```

Q: What is one recommendation you would make to increase the security of the shell?

One recommendation for inputs is to *sanitise inputs* received via the **`input`**() function, for example by using regular expressions on the input for X and Y to ensure that each input is indeed within the valid range of an integer value; or, even by using the **`isinstance`**(xxx, int)

One recommendation for **`os`** module routines is to test the current user's access by calling the **`os.access`**() function.

Add a section to your e-portfolio that provides a (pseudo)code example of changes you would make to the shell to improve its security.

```
PROCEDURE main

   SHOW COMMAND PROMPT
   LOOP
      ACCEPT COMMAND
      IF COMMAND EQ "LIST" THEN
         CHECK USER'S OS ACCESS FOR CURRENT DIRECTORY
         IF Access Level IS NOT PERMITTED
            SHOW ERROR MESSAGE
         OTHERWISE
            LIST CURRENT DIRECTORY CONTENTS
      OTHERWISE
      ... other command processing goes here
   UNTIL Exit = TRUE

END PROCEDURE
```

References

Praka, D. (2018) Write a shell in Python.

Szabo, G. (2018) Create your own interactive shell with cmd in Python