

August Tan

Professor Cappos

Application Security

September 15, 2015

This Turing-complete sandbox was written in python. There are two main security aspects to this sandbox.

The first is a user-required log in to gain access to the sandbox. If the user does not log in then the sandbox will not start. The account creation's current implementation is meant to be a precursor for more advanced security. At the moment, there are no user privileges and the account creation is available to anyone who has access to the sandbox. The accounts are intended to store personal user information once the sandbox is expanded beyond its current basic computation functionality. The passwords and usernames are stored in separate .txt files. A python library called "passlib" is utilized for its sha256 encryption on the passwords. The hashes are stored inside the password file and the hashes are checked with the user input when a login attempt is made.

The second security aspect to this program are the amount of restrictions placed on the user's input. The sandbox is essentially a compiler that will run user programs. However, those programs are limited to basic arithmetic such as: addition, multiplication, subtraction, and division. Additionally, only integers from 0-9 are allowed. The only python functions allowed are: print, fib(num), and powersTwo(num). This restriction ensures that the user can run only code that involves computation and prevents malicious code from executing from within the sandbox.

The example programs were included with the sandbox for convenience. They can be run by calling them with fib(10) or powersTwo(10) after logging in and reaching the code entry point. The Fibonacci function is a for loop that iterates through numbers leading up to the value passed in and adds them in the same way the Fibonacci sequence does. Afterwards, the Fibonacci value is printed out. The powersTwo function squares each number leading up to the value that was passed in and prints all of them out.

This sandbox is Turing-complete because it can take any computation, assuming they only use +, -, **, *, and /. It can handle such computations regardless of their length or complexity and compile/run them.

List of commands and restrictions of sandbox.py:

- At the beginning of the program you will be prompted to login or register
- After logging in you can run code within the sandbox
- Registration/Login
 - Any characters of any length are allowed for the username and the password
- Coding Sandbox
 - All the basic math operators +, -, *, /
 - All numerical values (0-9)

- Fib(int), runs the Fibonacci sequence on a given number and returns the value of it
- powersTwo(int), prints out the values of all the powers of two for integers between 0 and the int passed in

•

Git repo url:

<https://github.com/aot221/SandboxEnvironment/commit/35812632427610687df7156f645748da62b641d0>

Git sha1 hash: 35812632427610687df7156f645748da62b641d0