August Tan

Application Security

9/22/15

"potentiallyhackablesandbox.py" | https://seattle.poly.edu/attachment/wiki/EducationalAssignments/SecureTuringCompleteSandboxAttack/potentiallyhackablesandbox.py

This sandbox is written in python. It reads in a file which it then searches for blacklisted strings. It also utilizes a limited namespace run it executes the user generated code.

"easytocode.py" | https://seattle.poly.edu/attachment/wiki/EducationalAssignments/SecureTuringCompleteSandboxAttack/easytocode.py

This sandbox is written in python. It has an array of strings not allowed to be run on the program and runs with a namespace when it executes the user inputted code.

"a-sandbox.py" | https://seattle.poly.edu/attachment/wiki/EducationalAssignments/SecureTuringCompleteSandboxAttack/a-sandbox.py

This sandbox is written in python. It holds a global variable of allowed characters that it checks with the python program when it runs. All of the code is executed within a string namespace and the character A0 is used as a replacement for a 'while' loop.

"Application Security" | URL: https://github.com/ceinfo/ApplicationSecurity

This sandbox is written in Java. It limits the memory to allow 2001 integers inside. It will then compute read from a file instructions which are limited to "ADD, SET, DISP, HALT AND MUL. "

"Turing_Complete_Sandbox" | URL: https://github.com/mramdass/Turing_Complete_Sandbox

This sandbox utilizes a namespace that is applied when user inputted code is executed. It also applies a list of blacklisted strings that cannot be utilized.

"appsec1" | URL: https://github.com/fjm266/appSec1

This sandbox utilizes a restrictive namespace that is applied when the user inputted code is executed. This is the only method of security that this particular sandbox employs. This was written in python and runs python files but can also have non python data passed into it to be displayed.

“Secure_Turing_Complete_Sandbox” | URL:
https://github.com/kellender/Secure_Turing_Complete_Sandbox-

This sandbox limits the amount of memory that can be used by the sandbox. It utilizes piping to receive output from the child process. This is written in C and takes in C or C++ binary files.

“App-Sec” | URL : https://github.com/crimsonBeard/App-Sec

This sandbox is written in python. It has all of allowed functions defined by the sanbox.py file under a function called “command()”. If any of the strings entered are not allowed or are not one of the defined commands, then the program will crash.

“python-sandbox” | URL: https://github.com/PankajMoolrajani/python-sandbox

This sandbox is written in python. It utilizes a file size checker and has a blacklist of keywords not allowed. It also checks for malicious keywords when it runs and limits the available memory.