Prasit Saebae 0510451

Assignment 1 – Building the sandbox

Turing Completeness Sandbox

This sandbox is based on python programming language. So it uses same syntax as python 2.7.x. The below slow some example about python code. The tutorial for python is http://docs.python.org/2/tutorial/

```
for w in words:
    print w, len(w)
```

```
while True:
    pass
```

```
for w in words[:]:
    if len(w) > 6:
        words.insert(0, w)
```

However, this is a sandbox so it has been implemented some limitation for security reason

- Don't allow to write file, import library, gain privilege on file system
- It prevents DOS by limit memory usage(200 MB) for program and lower CPU priority for execution
- Sandbox can terminate themselves if untrust-code run over limit of time 60 s
- There are forbidden word that don't allow in code as `".os", ".write", "open", "close", "execfile", "compile", "reload", "eval", "input", "apply", "exit", "quit", "raw_iput", "dir", "globals", "locals", "vars", "import", "_"`
- Environment: Ubuntu 12.04.3, Python 2.7.3

Example program

- Example of program is in https://github.com/prasitsb/sandbox/blob/master/TestSandBox.py.
  - I have implement a program that counts from 10 to 1 and calculate Fibonacci program .
- Run sandbox
  - python <path sandbox.py> <path smallsandbox.py or untrustcode.py>
- Run sandbox in sandbox
  - <path sandbox.py> <path smallsandbox.py or untrustcode.py>
    - User have to do hard code for loading TestCode in smallsandbox.py by update `"TestSandBox.py"`

Mitigated Threat

- Prevent DOS by control Memory, CPU time
- Can terminate themselves
- Block write file

File format and operation support

- Sandbox also support many function and use same code syntax as python such as
  - define function by "**def** function_name()**:**"
  - Print or output by "**print** something"
  - Define variable, array, list, …
  - Operation: +, -,*, /, and, or, etc.
  - Etc. as python program

Source code for example program

```
def count(): # define function
    print '-' * 40 # print command
    print 'Test program that counts from 10 to 1 and print it'
    for i in range(10): # for loop
        print 10 - i
    print '-' * 40

def fib(n):
    print 'Test Fiboncci program'
    result = [] # define list
    a, b = 0, 1 # set value to variable
    for i in range(10):
        result.append(a) # append list
        a, b = b, a + b
    return result # return value

count() # call function
print fib(10) # print return value from function
```