Name: Shanqing Gu

Due 07/30/2018

MSDS 7349-HW3

Exercise 1: UNIX Password Cracker (30)

1) Write the cracker.py program. Turn in the code and output. (20)

1.1.1 Craker.py codes

```
#!/usr/bin/python
# -*- coding: utf-8 -*-


import crypt

def testPass(cryptPass):
    salt = cryptPass[0:2]


    dictFile = open('/Users/shanqinggu/Desktop/HW3dictionary.txt', 'r')
    for word in dictFile.readlines():
        word = word.strip('\n')
        cryptWord = crypt.crypt(word, salt)
        if (cryptWord == cryptPass):
            print("[+] Found Password: "+word+"\n")
            return

    print("[-] Password Not Found. \n")
    return

def main( ):
    passFile = open('/Users/shanqinggu/Desktop/HW3passwords.txt')
    for line in passFile.readlines():
        if ":" in line:
            user = line.split(':')[0]
            cryptPass = line.split(':')[1].strip(' ')
            print ("[*] Cracking Password For: "+user)
            testPass (cryptPass)


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

1.1.2 Output

In [1]:
```
import crypt
crypt.crypt ( "egg" , "HX" )
```

Out[1]:
```
'HX9LLTdc/jiDE'
```

In [2]:
```
passFile = open('/Users/shanqinggu/Desktop/HW3dictionary.txt')
plines = passFile.readlines()
print(plines)
passFile.close()
```

```
['apple\n', 'orange\n', 'egg\n', 'lemon\n', 'grapes\n', 'secret\n', 'strawberry\n', 'password\n']
```

In [3]:
```
dictFile = open('/Users/shanqinggu/Desktop/HW3passwords.txt', 'r')
lines = dictFile.readlines()
print(lines)
dictFile.close()
```

```
['victim: HX9LLTdc/jiDE: 503:100:Iama Victim:/home/victim:/bin/sh\n', 'root: DFNFxgW7C05fo: 504:100: Markus Hess:/ro
ot:/bin/bash\n']
```

In [4]:
```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import crypt

def testPass(cryptPass):
    salt = cryptPass[0:2]


    dictFile = open('/Users/shanqinggu/Desktop/HW3dictionary.txt', 'r')
    for word in dictFile.readlines():
        word = word.strip('\n')
        cryptWord = crypt.crypt(word, salt)
        if (cryptWord == cryptPass):
             print("[+] Found Password: "+word+"\n")
             return

    print("[-] Password Not Found. \n")
    return

def main( ):
    passFile = open('/Users/shanqinggu/Desktop/HW3passwords.txt')
    for line in passFile.readlines():
        if ":" in line:
            user = line.split(':')[0]
            cryptPass = line.split(':')[1].strip(' ')
            print ("[*] Cracking Password For: "+user)
            testPass (cryptPass)


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

```
[*] Cracking Password For: victim
[+] Found Password: egg

[*] Cracking Password For: root
[-] Password Not Found.
```

2) Identify from where you retrieve the salt value used in generating the signature (10)

The Unix operating system from both locally administered and network-based systems implements passwords in the */etc/passwd* file. The */etc/passwd* file contains the username, real name, identification information and the basic account information for each user. Each line in the file contains a database record and the records are separated by a colon (:).

(Reference: Practical UNIX and Internet Security, 3rd Edition by Alan Schwartz, Gene Spafford, Simson Garfinkel. https://www.oreilly.com/library/view/practical-unix-and/0596003234/ch04s03.html)

When you change your password, the */bin/passwd* program selects a salt based on the time of day. The salt is converted into a two-character string and is stored in the */etc/passwd* file along with the encrypted "password." In this manner, when you type your password at login time, the same salt is used again. Unix stores the salt as the first two characters of the encrypted password.

The salt can (1) increase the effective length of a password, (2) protect different users shared with the same password, (3)make hardware implementation of DES difficult.

Exercise 2: Zip File Password Cracker (30)

1- Write a script to test the use of the zipfile library (10)

2.1.1 Script to unzip eveil.zip with correct password

```
# Use right password (secret) to unzip evil.zip

import zipfile
zf = zipfile.ZipFile('/Users/shanqinggu/Desktop/evil.zip')
zf.extractall('/Users/shanqinggu/Desktop',pwd=b'secret')
print("Correct password. Unzipped files are extracted into file directory")
```

2.1.2 output

```
In [1]: # Use right password (secret) to unzip evil.zip

        import zipfile

        zf = zipfile.ZipFile('/Users/shanqinggu/Desktop/evil.zip')
        zf.extractall('/Users/shanqinggu/Desktop',pwd=b'secret')
        print("Correct password. Unzipped files are extracted into file directory")

        Correct password. Unzipped files are extracted into file directory
```

2.1.3. File 1: note_to_adam.txt

Sorry, you are too late - she ate the apple.

[Image downloaded from http://farm3.staticflickr.com/2422/4424308439_7bd9e833d3_z.jpg under Creative Commons License]

2.1.4. File 2: evil.jpg

2- Use the except Exception handler to catch exceptions and print them out when an incorrect password is used. (10)

2.2.1 Script to unzip eveil.zip with correct password

```
# Use right in correct password (evilpassword) to unzip evil.zip

import zipfile
zf = zipfile.ZipFile('/Users/shanqinggu/Desktop/evil.zip')
try:
    zf.extractall('/Users/shanqinggu/Desktop', pwd=b'evilpassword')
except:
    print ('Incorrect password')
```

2.1.2 output

```
In [2]: # Use right in correct password (evilpassword) to unzip evil.zip

        import zipfile

        zf = zipfile.ZipFile('/Users/shanqinggu/Desktop/evil.zip')

        try:
            zf.extractall('/Users/shanqinggu/Desktop', pwd=b'evilpassword')

        except:
            print ('Incorrect password')

        Incorrect password
```

3- Write a script that performs a dictionary attack on the password protected zip file. Execute your script and turn in the code and output. (10)

2.3.1 Script to unzip eveil.zip with dictionary attack

```
#!/usr/bin/python

# Use dictionary attck on the password protected evil.zip

import zipfile

def CheckPwd(evil, password):
    zf = zipfile.ZipFile('/Users/shanqinggu/Desktop/evil.zip')
    try:
        zf.extractall('/Users/shanqinggu/Desktop', pwd=password.encode('utf-8')) # or use 'ascii'
        return True
    except:
        return False

def main():
    ZipFile = '/Users/shanqinggu/Desktop/evil.zip'
    DictPwd = open('/Users/shanqinggu/Desktop/HW3dictionary.txt','r')
    for word in DictPwd.readlines():
        if (CheckPwd(ZipFile,word.rstrip())):
            print('PASSWORD FOUND!  It is: ' + word.rstrip())
            return
        else:
            print ('Password is not: ' + word.rstrip())


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

2.3.2 Output

In [1]:
```
#!/usr/bin/python

# Use dictionary attck on the password protected evil.zip

import zipfile

def CheckPwd(evil, password):
    zf = zipfile.ZipFile('/Users/shanqinggu/Desktop/evil.zip')
    try:
        zf.extractall('/Users/shanqinggu/Desktop', pwd=password.encode('utf-8')) # or use 'ascii'
        return True
    except:
        return False

def main():
    ZipFile = '/Users/shanqinggu/Desktop/evil.zip'
    DictPwd = open('/Users/shanqinggu/Desktop/HW3dictionary.txt','r')
    for word in DictPwd.readlines():
        if (CheckPwd(ZipFile,word.rstrip())):
            print('PASSWORD FOUND!  It is: ' + word.rstrip())
            return
        else:
            print ('Password is not: ' + word.rstrip())


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

```
Password is not: apple
Password is not: orange
Password is not: egg
Password is not: lemon
Password is not: grapes
PASSWORD FOUND!  It is: secret
```

Exercise 3: Port Scanner (40)

1-Create a script that iterates through a range of IP addresses and, for each IP address, will identify the active ports available for that IP address (20)

3.1.1 Python script:

```
import threading
import socket

ip = socket.gethostbyname('www.google.com')
print('The IP address for Google is: ' + ip )

def portscan(port):

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(0.5)#

    try:
        con = s.connect((ip, port))
        print('Port:', port,"is open.")
        con.close()

    except:
        print('closed')

def main():

    portlist = [20, 21, 22, 23, 25, 53, 80, 143, 443]
    # ports 20/21 for ftp, 22 for SSH, 23 for telnet, 25 for SMTP, 53 for DNS, 80 for http, 143 for IMAP, 443 for https

    for port in portlist:
        print(str(port)+ '.', end=" ")
        scanlist = portscan(port)

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

3.1.2 Screenshot of output

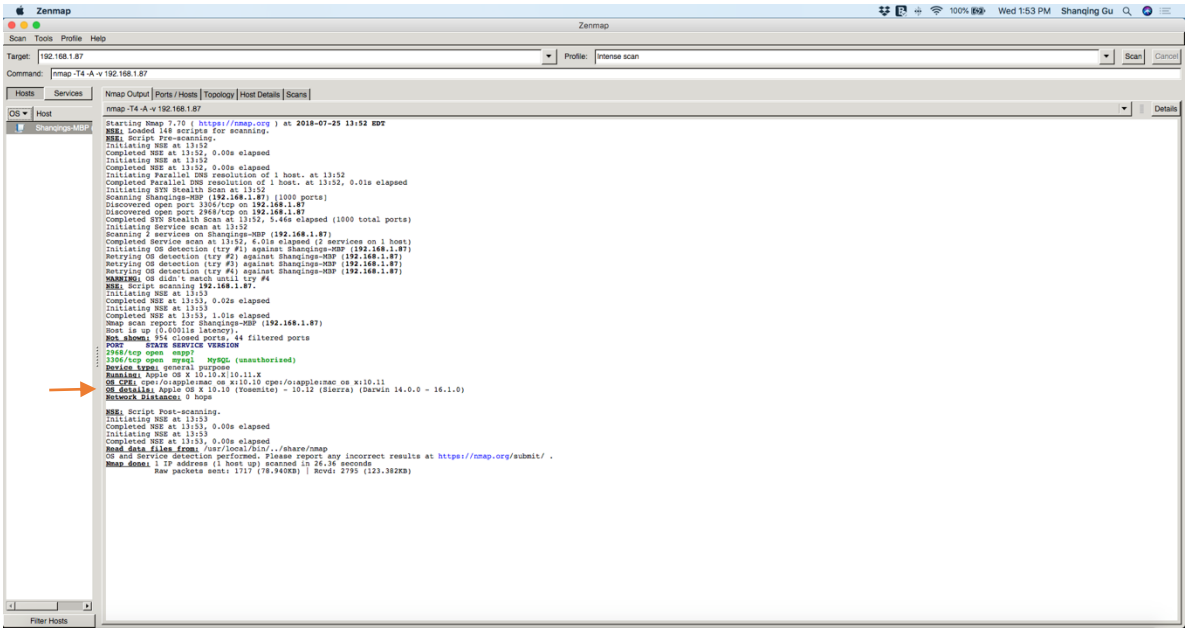
In [1]:

```
#!/usr/bin/python

# Simple Python-based port scanner using the socket library

import threading
import socket

ip = socket.gethostbyname('www.google.com')
print('The IP address for Google is: ' + ip )

def portscan(port):

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(0.5)#

    try:
        con = s.connect((ip, port))
        print('Port:', port,"is open.")
        con.close()

    except:
        print('closed')

def main():

    portlist = [20, 21, 22, 23, 25, 53, 80, 143, 443]
    # ports 20/21 for ftp, 22 for SSH, 23 for telnet, 25 for SMTP, 53 for DNS, 80 for http, 143 for IMAP, 443 for https

    for port in portlist:
        print(str(port)+ '.', end=" ")
        scanlist = portscan(port)

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

```
The IP address for Google is: 172.217.9.164
20. closed
21. closed
22. closed
23. closed
25. closed
53. closed
80. Port: 80 is open.
closed
143. closed
443. Port: 443 is open.
closed
```

2- Download and install the nmap port scanning software from nmap.org. Utilize nmap to identify the operating system and the open ports of devices on a range of IP addresses (20)

3.2.1 Utilize nmap to identify the operating system



3.2.2 Utilize nmap to open ports of devices on a range of IP addresses