MSDS 7349

Data and Network Security

Homework Introduction to Security

Due Week 5

Name: Austin Kelly

Collaborators: Tom Elkins

What to Submit

Your final submission shall be a single pdf document that includes this document, screen captures of your exercises plus your answers to each of the written questions (if any). Note that you are expected to clearly label each section so as to make it clear to the instructor what files and data belong to which exercise/question.

Collaboration is expected and encouraged; however, each student must hand in their own homework assignment. To the greatest extent possible, answers should not be copied but, instead, should be written in your own words. Copying answers from anywhere is plagiarism, this includes copying text directly from the textbook. Do not copy answers. Always use your own words. For each question list all persons with whom you collaborated and list all resources used in arriving at your answer. Resources include but are not limited to the textbook used for this course, papers read on the topic, and Google search results. Note that 'Google' is not a resource. Don't forget to place your name on the document.

Exercise 1: UNIX Password Cracker

The goal of this exercise is to write a password cracker for the UNIX file system. UNIX stores all passwords in the file/etc/passwd. Well, it doesn't store the password itself. Instead, it stores a signature of the password by using the passwordto encrypt a block of zero bits prepended by a salt value with a one-way function called crypt( ).

The result of the crypt() function is stored in the /etc/passwd file. For example, for a password of "egg" and salt equal to "HX", the function crypt('egg','HX') returns HX9LLTdc/jiDE.

When you try to log in, the program /bin/login takes the password that you typed, uses crypt( ) to encrypt a block of zero bits, and compares the result of this function with the value stored in the /etc/passwd file.

The security of this approach rests on both the strength of the crypt( ) function and the difficulty in guessing a user's password. The crypt( ) algorithm has proven to be highly resistant to attacks. Conversely, the user's choices for passwords have been found to be relatively easy to guess, with many passwords being words contained in the dictionary.

To write our UNIX password cracker, we will need to use the crypt( ) algorithm that hashes UNIX passwords. Fortunately, the crypt library already exists in the Python standard library. To calculate the encrypted UNIX password signature, we simply call the function crypt.cypt( ) and pass it the password and salt as parameters. This function returns the signature as a string.

A simple dictionary attack involves computing the possible signatures generated for each word in the dictionary with a range of salt values.

As I am running jupyter on a windows machine, I consistently received errors when attempting to import and run crypt. After some research I learned crypt will not run on windows as it is solely meant for UNIX. This prompted a hefty Google search.

During my adventure, I stumbled upon https://pythonhosted.org/passlib/lib/passlib.hash.des_crypt.html#algorithm (https://pythonhosted.org/passlib/lib/passlib.hash.des_crypt.html#algorithm) which led me through how to work around crypt with the use of **passlib.hash**

Here is where we learn to use the library:

In [1]:
```
# import the handler class
from passlib.hash import sha256_crypt

# hash a password using the default settings:
hash = sha256_crypt.encrypt("password")
hash
#'$5$rounds=40000$HIo6SCnVL9zqF8TK$y2sUnu13gp4cv0YgLQMW56PfQjWaTyiHjVbXTgleYG9'

# note that each call to encrypt() generates a new salt,
# and thus the contents of the hash will differ, despite using the same password:
sha256_crypt.encrypt("password")
#'$5$rounds=40000$1JfxoiYM5Pxokyh8$ez8uV8jjXW7SjpaTg2vHJmx3Qn36uyZpjhyC9AfBi7B'

# if the hash supports a variable number of iterations (which sha256_crypt does),
# you can override the default value via the 'rounds' keyword:
sha256_crypt.encrypt("password", rounds=12345)
#'$5$rounds=12345$UeVpHaN2YFDwBoeJ$NJN8DwVZ4UfQw6.ijJZNWoZtk1Ivi5YfKCDsI2HzSq2'
#           ^^^^^

# on the other end of things, the verify() method takes care of
# checking if a password matches an existing hash string:
sha256_crypt.verify("password", hash)
#True
#sha256_crypt.verify("letmeinplz", hash)
#False
```

Out[1]:
```
True
```

In [2]:
```
# here I used conda to install passlib, which was then imported here:
from passlib import hash
from passlib.hash import des_crypt
from passlib.hash import sha256_crypt

# since I am learning how to use this, let's try an example!

# time to use a test encryption
test = sha256_crypt.encrypt("password", salt='tP')
test
# print(test) # From here we see that 'tP' is used for the salt.

#Let's encrypt something else
sha256_crypt.encrypt("password")
#print(newPass)

# this library came with a built-in verify function. Time to test it!
sha256_crypt.verify('password', test)
# When run, the output here is True

sha256_crypt.verify('DrEngels',test)
#The output here is False when Run. 'verify' works!
```

Out[2]:
```
False
```

Running the code above, we see the **passlib** library is effective when it comes to the DES encryption standard. Now we will begin to work on the first question

Let's create our first password cracker using a dictionary attack

1) Create a file called cracker.py. Start your program by reading in the HW1-passwords.txt file and, for each password found in the file, iterate through each dictionary word found in the HW1-dictionary.txt file and appropriate salt value. Report out the password found, if any, for each user. If no password is found, indicate that no password was found.

In [3]:
```
#importing passlib is not required as it was loaded in the previous cells

# open and read the entire dictionary
webster = open('MSDS7349Homework1/HW1dictionary_2_2.txt')
merrWeb = webster.read().splitlines()

# open and read the entire password file
keyA = open('MSDS7349Homework1/HW1passwords_2_2.txt')
openDoor = keyA.read().splitlines()

# Both files are open and read. Time to loop
for passage in openDoor:
    access = passage.split(":")
    easyAccess = access[1].strip()
    theSalt = easyAccess[:2]
    gainEntry = False

    print "\n '%s'; '%s'" % (easyAccess, theSalt)

    for passWord in merrWeb:
        newHash = des_crypt.encrypt(passWord, salt=theSalt)

        if (newHash == easyAccess):
            gainEntry = True
            print "'%s''s password is '%s'" % (access[4],passWord)

    if not gainEntry:
        print "There is no password here for %s" % (access[4])

print "cracker.py has completed the search"
```

```
 'HX9LLTdc/jiDE'; 'HX'
'Iama Victim''s password is 'egg'

 'DFNFxgW7C05fo'; 'DF'
There is no password here for  Markus Hess
cracker.py has completed the search
```

2) Using literature review, identify from where you can retrieve the salt value used in generating the signature.

A salt value is a value much like a nonce which helps to obfuscate the password at hand. The salt value is randomly chosen from both capital and lowercase letters of the english alphabet as well as the common symbols period (.) and slash (/). Counting all of these symbols together gives us 64 possible choices with a total of 4096 permutations of the initial password.

Exercise 2 : Zip File Password Cracker

The goal of this exercise is to write a zip file extractor and password cracker. For this exercise, we will use the zipfile library. You may view information about the zipfile library by issuing the command help('zipfile') to learn more about the library. Pay close attention to the extractall( ) method. You may use this method to extract the contents from a zip file.

Let's begin the process of writing a zip file password cracker.

1) Write a quick script to test the use of the zipfile library. After importing the library, instantiate a new ZipFile class by specifying the filename of the password-protected zip file (evil.zip). utilize the extractall( ) method and specify the optional parameter for the password (secret). Execute your script and turn in the code and output.

In [5]:
```
help('zipfile') # this helped a lot. (no pun intended)
```

```
...
```

2.1) Time to test the zip's library:

In [14]:
```
# Let's import the library so we can play with it.
import zipfile
import os.path # We need this so we can query the file system.

# Let's create an instance of the zip file so that we can open it!
zipper = zipfile.ZipFile("evil_2_2.zip")

# Extract what you can from the zip file
zipper.extractall(pwd = 'secret') # <-- The only reason I know this is because
                                  # I kinda went through the passwords list and entered the

# Let's keep track of the number of things in there
theStuff = 1

# Let's look at them one by one:
for filename in zipper.namelist():

    # What is the file?
    whatsThat = zipper.getinfo(filename)

    # Let's see if anything was extracted or not
    if os.path.isfile(filename):
        foundIt = "<-- We've got something!"
    else:
        foundIt = "<-- There's nothing here..."

    # What was in it?
    print 'Item %d: %-30s (%5d bytes) %s' % (theStuff, whatsThat.filename, whatsThat.file_s
                                            foundIt)

    # Update the counter from line 12
    theStuff += 1
```

```
Item 1: evil/note_to_adam.txt          (  171 bytes) <-- We've got something!
Item 2: evil/evil.jpg                  (38540 bytes) <-- We've got something!
```

Exception Handler for wrong Passwords

2) Use the except Exception exception handler to catch exceptions and print them out when an incorrect password is used. Execute your script with an incorrect password and exception handler and turn in the code and output.

In [25]:
```
# Let's make a function which acts on the object itself and password
#    string and tries to extract the contents
def extract(zipFile,password):

    try:
        # Once again, create the instance of the zipfile object and open it
        zipper = zipfile.ZipFile(zipFile)

        # Try to open it with the password supplied
        zipper.extractall(pwd=password)

    except:
        # Are we sure the file is in there?
        if os.path.isfile(zipFile):
            print 'Incorrect password "{0}" for this zip file [{1}].'.format(password,zipFi
        else:
            print "The file specified here [{0}] doesn't exist!".format(zipFile)

    else:
        # Reaching this point would indicate some progress was made and the contents were a
        print 'That password "{0}" was accepted for the file [{1}]'.format(password,zipFile

        # Let's count them again
        theStuff = 1

        # Let's loop some more
        for filename in zipper.namelist():

            # What's in this thing!?
            fileInfo = zipper.getinfo(filename)

            # Does this file exist in the local system?
            if os.path.isfile(filename):
                thingFinder = 'we found something!'
            else:
                thingFinder = 'There seems to be nothing here...'

            # Let's see what we get
            print 'Item %d: %30s (%5d bytes) %s' % (theStuff, fileInfo.filename, fileInfo.f

            # Don't forget to update the counter!
            theStuff += 1
    print
```

In [26]:
```
# Let's see if we can break this new shiny thing
extract('evil_2_2.zip', 'banana')
extract('evil_2_2.zip', 'secret')
extract('evil_4_3.zip', 'Banana')
```

```
Incorrect password "banana" for this zip file [evil_2_2.zip].

That password "secret" was accepted for the file [evil_2_2.zip]
Item 1:          evil/note_to_adam.txt (  171 bytes) we found something!
Item 2:                  evil/evil.jpg (38540 bytes) we found something!

The file specified here [evil_4_3.zip] doesn't exist!
```

Since that's done, now we can work on...

The Dictionary Attack

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. Be sure to provide user feedback on exceptions thrown.

In [29]:
```
# Create something which iterates through all the words in the file (like I 'kinda' did ear
def bookThrow(zipFile):

    # Let's load the dictionary
    webster = open('MSDS7349Homework1/HW1dictionary_2_2.txt','r')

    # Let's split those potential passwords into an array
    merrWeb = webster.read().splitlines()

    # Iterate through the possible passwords!
    for password in merrWeb:

        # See what happens when you use each word and see which passes!
        extract(zipFile,password)
```

In [31]:
```
# Test it out! Let's see what we get
bookThrow('evil_2_2.zip')
```

```
Incorrect password "apple" for this zip file [evil_2_2.zip].

Incorrect password "orange" for this zip file [evil_2_2.zip].

Incorrect password "egg" for this zip file [evil_2_2.zip].

Incorrect password "lemon" for this zip file [evil_2_2.zip].

Incorrect password "grapes" for this zip file [evil_2_2.zip].

That password "secret" was accepted for the file [evil_2_2.zip]
Item 1:          evil/note_to_adam.txt (  171 bytes) we found something!
Item 2:                 evil/evil.jpg (38540 bytes) we found something!

Incorrect password "strawberry" for this zip file [evil_2_2.zip].

Incorrect password "password" for this zip file [evil_2_2.zip].
```

Exercise 3 : Port Scanner

The goal of this exercise is to write a simple port scanner for networked systems. Using the socket library, you will 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. At least ports corresponding to telnet, ftp SSH, smtp, http, imap, and https services should be scanned and identified.

In [40]:
```
#  Import what was specified in the question itself.
import socket

#  Set the first three general octets
generalAddress = "192.168.0"

#  After googling and a substantial amount of help from Mr. Elkins, this is what was found:
thePorts = (7,20,21,22,23,25,43,53,67,68,80,102,110,119,123,135,137,138,139,143,161,162,443
            1433,1434,1725,1755,1863,2049,3306,3389,3689,3690,5000,8080,8200,8888,11371)

theNames = {7:"Echo",20:"FTP",21:"FTP",22:"SSH",23:"Telnet",25:"SMTP",43:"WHOIS",53:"DNS",6
            102:"MS Exchange",110:"POP3",119:"NNTP",123:"NTP",135:"Microsoft RPC",137:"NetB
            143:"IMAP",161:"SNMP",162:"SNMP",443:"HTTPS",465:"SMTPS",513:"rlogin",587:"SMTP
            691:"MS Exchange",1194:"OpenVPN",1433:"Microsoft SQL",1434:"Microsoft SQL",1725
            1863:"MSN",2049:"NFS",3306:"MySQL",3389:"Terminal Server",3689:"iTunes",3690:"S
            8080:"HTTP Proxy",8200:"VMWare Server",8888:"LocalHost",11371:"OpenPGP"}

#  Loop through the remaining dotDecimals
for dotDecimal in range(0,255):

    #  Let's start the number of ports at 0 (gotta catch 'em all!)
    numOpen = 0

    #  Piece together what we have to complete the IPv4 address
    testedIP = "{0}.{1}".format(generalAddress,dotDecimal)

    #  Our friend, the for loop to run through the ports in question
    for targetPort in thePorts:

        #  Make a socket object so we can use it!
        mySocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        mySocket.settimeout(0.1)

        try:
            # Check connection by trying to connect
            connStatus = mySocket.connect_ex((testedIP,targetPort))

            #  Houston, do we have a connection?
            if connStatus == 0:
                if numOpen == 0:
                    print "-" * 60

                #  Move up one each time
                numOpen += 1

                # Signal the ports which are open
                print "{0}:{1} ({2}) - OPEN".format(testedIP,targetPort,theNames[targetPort

        except socket.error:
            print "No connection to the server at {0}".format(testedIP)

    #  Did we find any open ports, though?
    if numOpen == 0:
        print "We couldn't find anything at this address: {0}".format(testedIP)
    else:
        print "-" * 60

print "Scan complete"
```

```
We couldn't find anything at this address: 192.168.0.0
-------------------------------------------------------------
192.168.0.1:80 (HTTP) - OPEN
-------------------------------------------------------------
We couldn't find anything at this address: 192.168.0.2
We couldn't find anything at this address: 192.168.0.3
We couldn't find anything at this address: 192.168.0.4
-------------------------------------------------------------
192.168.0.5:135 (Microsoft RPC) - OPEN
192.168.0.5:139 (NetBIOS) - OPEN
-------------------------------------------------------------
-------------------------------------------------------------
192.168.0.6:135 (Microsoft RPC) - OPEN
192.168.0.6:139 (NetBIOS) - OPEN
-------------------------------------------------------------
We couldn't find anything at this address: 192.168.0.7
We couldn't find anything at this address: 192.168.0.8
We couldn't find anything at this address: 192.168.0.9
We couldn't find anything at this address: 192.168.0.10
We couldn't find anything at this address: 192.168.0.11
We couldn't find anything at this address: 192.168.0.12
We couldn't find anything at this address: 192.168.0.13
We couldn't find anything at this address: 192.168.0.14
We couldn't find anything at this address: 192.168.0.15
We couldn't find anything at this address: 192.168.0.16
We couldn't find anything at this address: 192.168.0.17
We couldn't find anything at this address: 192.168.0.18
We couldn't find anything at this address: 192.168.0.19
We couldn't find anything at this address: 192.168.0.20
We couldn't find anything at this address: 192.168.0.21
We couldn't find anything at this address: 192.168.0.22
We couldn't find anything at this address: 192.168.0.23
We couldn't find anything at this address: 192.168.0.24
We couldn't find anything at this address: 192.168.0.25
We couldn't find anything at this address: 192.168.0.26
We couldn't find anything at this address: 192.168.0.27
We couldn't find anything at this address: 192.168.0.28
We couldn't find anything at this address: 192.168.0.29
We couldn't find anything at this address: 192.168.0.30
We couldn't find anything at this address: 192.168.0.31
We couldn't find anything at this address: 192.168.0.32
We couldn't find anything at this address: 192.168.0.33
We couldn't find anything at this address: 192.168.0.34
We couldn't find anything at this address: 192.168.0.35
We couldn't find anything at this address: 192.168.0.36
We couldn't find anything at this address: 192.168.0.37
We couldn't find anything at this address: 192.168.0.38
We couldn't find anything at this address: 192.168.0.39
We couldn't find anything at this address: 192.168.0.40
We couldn't find anything at this address: 192.168.0.41
We couldn't find anything at this address: 192.168.0.42
We couldn't find anything at this address: 192.168.0.43
We couldn't find anything at this address: 192.168.0.44
We couldn't find anything at this address: 192.168.0.45
We couldn't find anything at this address: 192.168.0.46
We couldn't find anything at this address: 192.168.0.47
We couldn't find anything at this address: 192.168.0.48
We couldn't find anything at this address: 192.168.0.49
We couldn't find anything at this address: 192.168.0.50
We couldn't find anything at this address: 192.168.0.51
We couldn't find anything at this address: 192.168.0.52
We couldn't find anything at this address: 192.168.0.53
```

We couldn't find anything at this address: 192.168.0.54
We couldn't find anything at this address: 192.168.0.55
We couldn't find anything at this address: 192.168.0.56
We couldn't find anything at this address: 192.168.0.57
We couldn't find anything at this address: 192.168.0.58
We couldn't find anything at this address: 192.168.0.59
We couldn't find anything at this address: 192.168.0.60
We couldn't find anything at this address: 192.168.0.61
We couldn't find anything at this address: 192.168.0.62
We couldn't find anything at this address: 192.168.0.63
We couldn't find anything at this address: 192.168.0.64
We couldn't find anything at this address: 192.168.0.65
We couldn't find anything at this address: 192.168.0.66
We couldn't find anything at this address: 192.168.0.67
We couldn't find anything at this address: 192.168.0.68
We couldn't find anything at this address: 192.168.0.69
We couldn't find anything at this address: 192.168.0.70
We couldn't find anything at this address: 192.168.0.71
We couldn't find anything at this address: 192.168.0.72
We couldn't find anything at this address: 192.168.0.73
We couldn't find anything at this address: 192.168.0.74
We couldn't find anything at this address: 192.168.0.75
We couldn't find anything at this address: 192.168.0.76
We couldn't find anything at this address: 192.168.0.77
We couldn't find anything at this address: 192.168.0.78
We couldn't find anything at this address: 192.168.0.79
We couldn't find anything at this address: 192.168.0.80
We couldn't find anything at this address: 192.168.0.81
We couldn't find anything at this address: 192.168.0.82
We couldn't find anything at this address: 192.168.0.83
We couldn't find anything at this address: 192.168.0.84
We couldn't find anything at this address: 192.168.0.85
We couldn't find anything at this address: 192.168.0.86
We couldn't find anything at this address: 192.168.0.87
We couldn't find anything at this address: 192.168.0.88
We couldn't find anything at this address: 192.168.0.89
We couldn't find anything at this address: 192.168.0.90
We couldn't find anything at this address: 192.168.0.91
We couldn't find anything at this address: 192.168.0.92
We couldn't find anything at this address: 192.168.0.93
We couldn't find anything at this address: 192.168.0.94
We couldn't find anything at this address: 192.168.0.95
We couldn't find anything at this address: 192.168.0.96
We couldn't find anything at this address: 192.168.0.97
We couldn't find anything at this address: 192.168.0.98
We couldn't find anything at this address: 192.168.0.99
We couldn't find anything at this address: 192.168.0.100
We couldn't find anything at this address: 192.168.0.101
We couldn't find anything at this address: 192.168.0.102
We couldn't find anything at this address: 192.168.0.103
We couldn't find anything at this address: 192.168.0.104
We couldn't find anything at this address: 192.168.0.105
We couldn't find anything at this address: 192.168.0.106
We couldn't find anything at this address: 192.168.0.107
We couldn't find anything at this address: 192.168.0.108
We couldn't find anything at this address: 192.168.0.109
We couldn't find anything at this address: 192.168.0.110
We couldn't find anything at this address: 192.168.0.111
We couldn't find anything at this address: 192.168.0.112
We couldn't find anything at this address: 192.168.0.113
We couldn't find anything at this address: 192.168.0.114
We couldn't find anything at this address: 192.168.0.115

We couldn't find anything at this address: 192.168.0.116
We couldn't find anything at this address: 192.168.0.117
We couldn't find anything at this address: 192.168.0.118
We couldn't find anything at this address: 192.168.0.119
We couldn't find anything at this address: 192.168.0.120
We couldn't find anything at this address: 192.168.0.121
We couldn't find anything at this address: 192.168.0.122
We couldn't find anything at this address: 192.168.0.123
We couldn't find anything at this address: 192.168.0.124
We couldn't find anything at this address: 192.168.0.125
We couldn't find anything at this address: 192.168.0.126
We couldn't find anything at this address: 192.168.0.127
We couldn't find anything at this address: 192.168.0.128
We couldn't find anything at this address: 192.168.0.129
We couldn't find anything at this address: 192.168.0.130
We couldn't find anything at this address: 192.168.0.131
We couldn't find anything at this address: 192.168.0.132
We couldn't find anything at this address: 192.168.0.133
We couldn't find anything at this address: 192.168.0.134
We couldn't find anything at this address: 192.168.0.135
We couldn't find anything at this address: 192.168.0.136
We couldn't find anything at this address: 192.168.0.137
We couldn't find anything at this address: 192.168.0.138
We couldn't find anything at this address: 192.168.0.139
We couldn't find anything at this address: 192.168.0.140
We couldn't find anything at this address: 192.168.0.141
We couldn't find anything at this address: 192.168.0.142
We couldn't find anything at this address: 192.168.0.143
We couldn't find anything at this address: 192.168.0.144
We couldn't find anything at this address: 192.168.0.145
We couldn't find anything at this address: 192.168.0.146
We couldn't find anything at this address: 192.168.0.147
We couldn't find anything at this address: 192.168.0.148
We couldn't find anything at this address: 192.168.0.149
We couldn't find anything at this address: 192.168.0.150
We couldn't find anything at this address: 192.168.0.151
We couldn't find anything at this address: 192.168.0.152
We couldn't find anything at this address: 192.168.0.153
We couldn't find anything at this address: 192.168.0.154
We couldn't find anything at this address: 192.168.0.155
We couldn't find anything at this address: 192.168.0.156
We couldn't find anything at this address: 192.168.0.157
We couldn't find anything at this address: 192.168.0.158
We couldn't find anything at this address: 192.168.0.159
We couldn't find anything at this address: 192.168.0.160
We couldn't find anything at this address: 192.168.0.161
We couldn't find anything at this address: 192.168.0.162
We couldn't find anything at this address: 192.168.0.163
We couldn't find anything at this address: 192.168.0.164
We couldn't find anything at this address: 192.168.0.165
We couldn't find anything at this address: 192.168.0.166
We couldn't find anything at this address: 192.168.0.167
We couldn't find anything at this address: 192.168.0.168
We couldn't find anything at this address: 192.168.0.169
We couldn't find anything at this address: 192.168.0.170
We couldn't find anything at this address: 192.168.0.171
We couldn't find anything at this address: 192.168.0.172
We couldn't find anything at this address: 192.168.0.173
We couldn't find anything at this address: 192.168.0.174
We couldn't find anything at this address: 192.168.0.175
We couldn't find anything at this address: 192.168.0.176
We couldn't find anything at this address: 192.168.0.177

We couldn't find anything at this address: 192.168.0.178
We couldn't find anything at this address: 192.168.0.179
We couldn't find anything at this address: 192.168.0.180
We couldn't find anything at this address: 192.168.0.181
We couldn't find anything at this address: 192.168.0.182
We couldn't find anything at this address: 192.168.0.183
We couldn't find anything at this address: 192.168.0.184
We couldn't find anything at this address: 192.168.0.185
We couldn't find anything at this address: 192.168.0.186
We couldn't find anything at this address: 192.168.0.187
We couldn't find anything at this address: 192.168.0.188
We couldn't find anything at this address: 192.168.0.189
We couldn't find anything at this address: 192.168.0.190
We couldn't find anything at this address: 192.168.0.191
We couldn't find anything at this address: 192.168.0.192
We couldn't find anything at this address: 192.168.0.193
We couldn't find anything at this address: 192.168.0.194
We couldn't find anything at this address: 192.168.0.195
We couldn't find anything at this address: 192.168.0.196
We couldn't find anything at this address: 192.168.0.197
We couldn't find anything at this address: 192.168.0.198
We couldn't find anything at this address: 192.168.0.199
We couldn't find anything at this address: 192.168.0.200
We couldn't find anything at this address: 192.168.0.201
We couldn't find anything at this address: 192.168.0.202
We couldn't find anything at this address: 192.168.0.203
We couldn't find anything at this address: 192.168.0.204
We couldn't find anything at this address: 192.168.0.205
We couldn't find anything at this address: 192.168.0.206
We couldn't find anything at this address: 192.168.0.207
We couldn't find anything at this address: 192.168.0.208
We couldn't find anything at this address: 192.168.0.209
We couldn't find anything at this address: 192.168.0.210
We couldn't find anything at this address: 192.168.0.211
We couldn't find anything at this address: 192.168.0.212
We couldn't find anything at this address: 192.168.0.213
We couldn't find anything at this address: 192.168.0.214
We couldn't find anything at this address: 192.168.0.215
We couldn't find anything at this address: 192.168.0.216
We couldn't find anything at this address: 192.168.0.217
We couldn't find anything at this address: 192.168.0.218
We couldn't find anything at this address: 192.168.0.219
We couldn't find anything at this address: 192.168.0.220
We couldn't find anything at this address: 192.168.0.221
We couldn't find anything at this address: 192.168.0.222
We couldn't find anything at this address: 192.168.0.223
We couldn't find anything at this address: 192.168.0.224
We couldn't find anything at this address: 192.168.0.225
We couldn't find anything at this address: 192.168.0.226
We couldn't find anything at this address: 192.168.0.227
We couldn't find anything at this address: 192.168.0.228
We couldn't find anything at this address: 192.168.0.229
We couldn't find anything at this address: 192.168.0.230
We couldn't find anything at this address: 192.168.0.231
We couldn't find anything at this address: 192.168.0.232
We couldn't find anything at this address: 192.168.0.233
We couldn't find anything at this address: 192.168.0.234
We couldn't find anything at this address: 192.168.0.235
We couldn't find anything at this address: 192.168.0.236
We couldn't find anything at this address: 192.168.0.237
We couldn't find anything at this address: 192.168.0.238
We couldn't find anything at this address: 192.168.0.239

```
We couldn't find anything at this address: 192.168.0.240
We couldn't find anything at this address: 192.168.0.241
We couldn't find anything at this address: 192.168.0.242
We couldn't find anything at this address: 192.168.0.243
We couldn't find anything at this address: 192.168.0.244
We couldn't find anything at this address: 192.168.0.245
We couldn't find anything at this address: 192.168.0.246
We couldn't find anything at this address: 192.168.0.247
We couldn't find anything at this address: 192.168.0.248
We couldn't find anything at this address: 192.168.0.249
We couldn't find anything at this address: 192.168.0.250
We couldn't find anything at this address: 192.168.0.251
We couldn't find anything at this address: 192.168.0.252
We couldn't find anything at this address: 192.168.0.253
We couldn't find anything at this address: 192.168.0.254
Scan complete
```

References

http://www.pythonforbeginners.com/code-snippets-source-code/port-scanner-in-python (http://www.pythonforbeginners.com/code-snippets-source-code/port-scanner-in-python)