Introduction

김영지

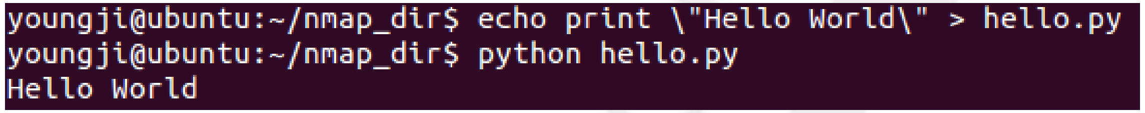
환경 구축

- Python 언어 설치
  Mac OS X, Linux는 이미 설치되어 있음
  Windows는 따로 설치

- 라이브러리 설치
  wget , easy-install , pip
  nmap (포트 스캐닝 툴)
  pygeoip (ip)
  mechanize (웹)
  BeautifulSoup4 (HTML)

Interpreted vs. Interactive

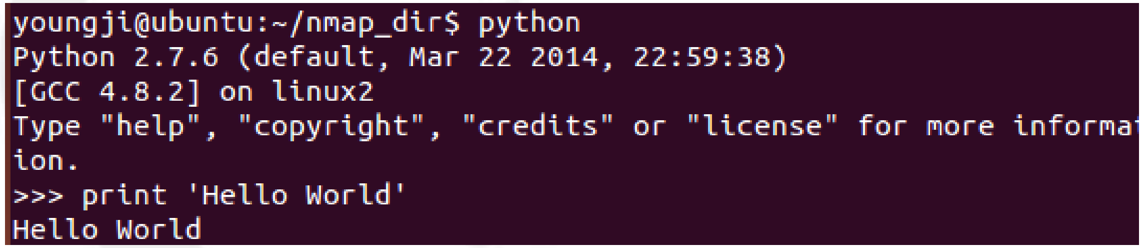
Interpreted Python



```
youngji@ubuntu:~/nmap_dir$ echo print \"Hello World\" > hello.py
youngji@ubuntu:~/nmap_dir$ python hello.py
Hello World
```

런타임 시 코드를 읽고 실행시킨다.

Interactive Python



```
youngji@ubuntu:~/nmap_dir$ python
Python 2.7.6 (default, Mar 22 2014, 22:59:38)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more informat
ion.
>>> print 'Hello World'
Hello World
```

프로그래머가 인터프리터를 불러 직접적으로 상호작용할 수 있다.

명령프롬프트에 python 치면 "〉〉〉" 나타남.

Python 언어

①

Python 언어

- 변수(Variables)
  프로그래머가 변수 타입을 선언할 필요 없다.
  -〉 인터프리터가 결정함

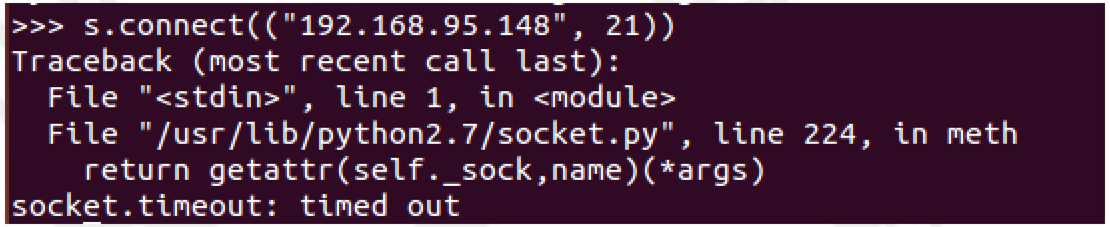
- 문자열(Strings)
  upper(), lower(), replace(), find()

- 리스트(Lists)
  appending, inserting, removing, popping, indexing, counting, sorting, reversing lists

- 사전(Dictionaries)
  해시테이블 제공. Dict = [‘키’ : 값, …]
  Dict.keys(), Dict.items(), Dict[‘키’]

Python 언어

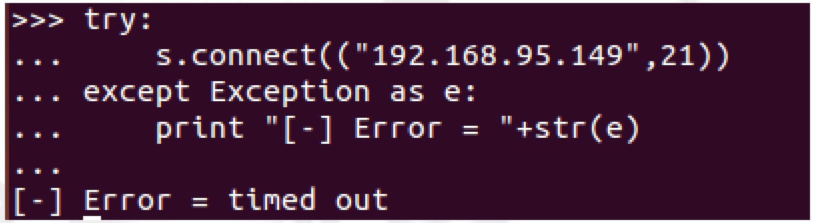
- 네트워킹(Networking)
  socket 모듈 사용



```
>>> s.connect(("192.168.95.148", 21))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/python2.7/socket.py", line 224, in meth
    return getattr(self._sock,name)(*args)
socket.timeout: timed out
```

- 조건문(Selection)
  If/elif/else
  :로 시작, 들여쓰기 중요함

- 예외처리(Exception handling)
  try:/except:
  변수 e 사용



```
>>> try:
...     s.connect(("192.168.95.149",21))
... except Exception as e:
...     print "[-] Error = "+str(e)
...
[-] Error = timed out
```

Python 언어

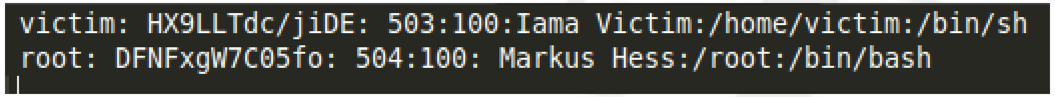
- 함수(Functions)
  앞에 def 선언
- 반복문(Iteration)
  for x in range(a, b)
  for x in list
- 파일입출력(File I/O)
  open, readlines, strip
- 시스템모듈(Sys Module)
  플래그, 버전, integer 최대 크기, 가능한 모듈 등
  인터프리터에 의해 사용되거나 유지되는 object들으로의 접근 제공
- OS모듈(OS Module)
  OS 환경, 파일시스템, 사용자 DB, 권한 등

Python 프로그램

②

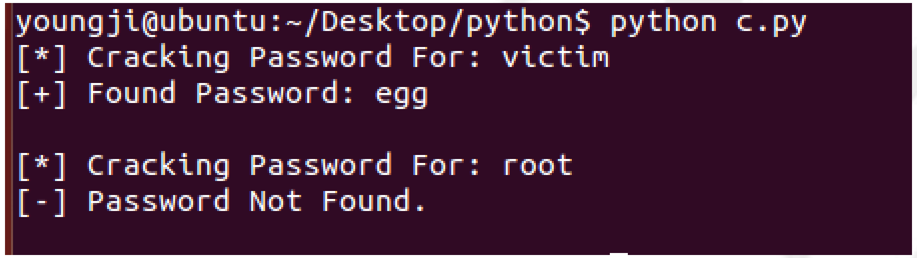
Unix Password Cracker

- Import crypt
- Crypt(word, salt) -> string

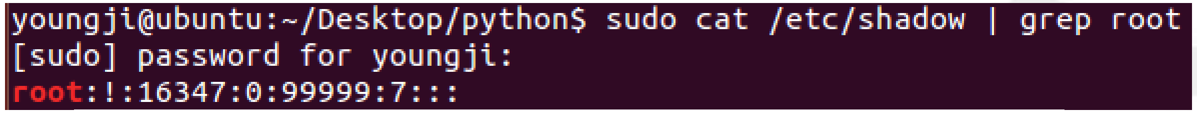


```
victim: HX9LLTdc/jiDE: 503:100:Iama Victim:/home/victim:/bin/sh
root: DFNFxgW7C05fo: 504:100: Markus Hess:/root:/bin/bash
```

Passwords.txt



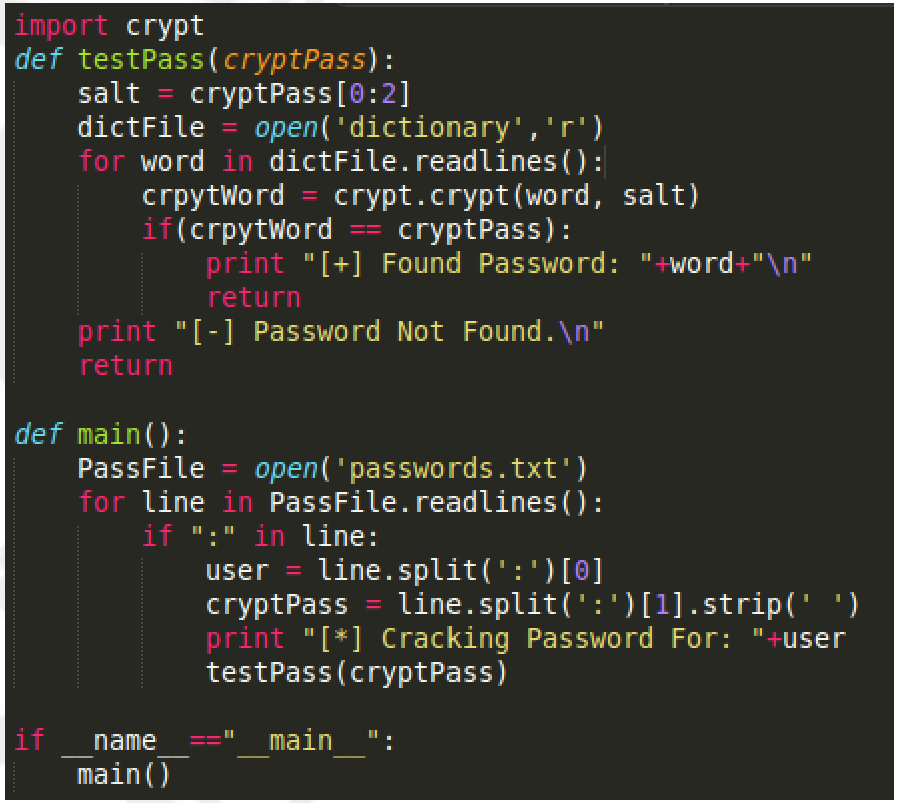
```
youngji@ubuntu:~/Desktop/python$ python c.py
[*] Cracking Password For: victim
[+] Found Password: egg

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



```
youngji@ubuntu:~/Desktop/python$ sudo cat /etc/shadow | grep root
[sudo] password for youngji:
root:!:16347:0:99999:7:::
```

```
cat /etc/shadow | grep root
root:$6$ms32yIGN$NyXj0YofkK14MpRwFHvXQW0yvUid.slJtgxHE2EuQqgD74S/
  GaGGs5VCnqeC.bSOMzTf/EFS3uspQMNeepIAc.:15503:0:99999:7:::
```

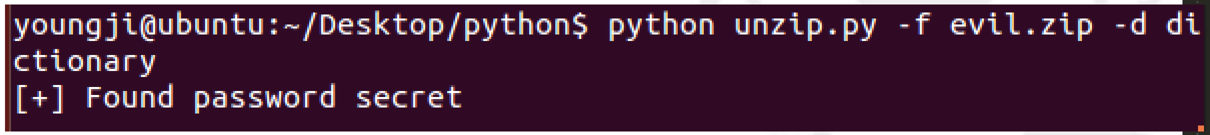
<SHA-512>



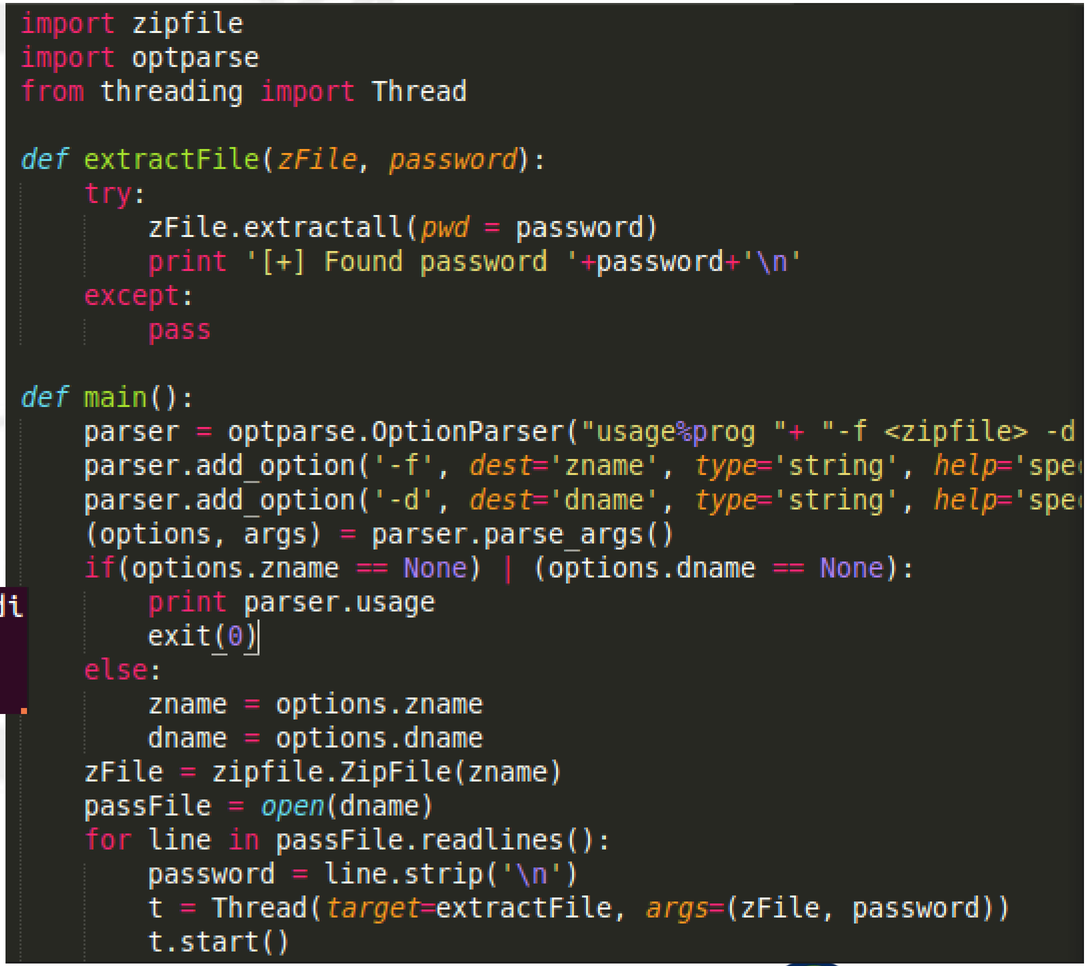
```
import crypt
def testPass(cryptPass):
    salt = cryptPass[0:2]
    dictFile = open('dictionary','r')
    for word in dictFile.readlines():
        crpytWord = crypt.crypt(word, salt)
        if(crpytWord == cryptPass):
            print "[+] Found Password: "+word+"\n"
            return
    print "[-] Password Not Found.\n"
    return

def main():
    PassFile = open('passwords.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()
```

Zip-File Password Cracker

- Import zipfile
- Extractall(password=pwd)
- Import optparse



```
youngji@ubuntu:~/Desktop/python$ python unzip.py -f evil.zip -d di
ctionary
[+] Found password secret
```



```
import zipfile
import optparse
from threading import Thread

def extractFile(zFile, password):
    try:
        zFile.extractall(pwd = password)
        print '[+] Found password '+password+'\n'
    except:
        pass

def main():
    parser = optparse.OptionParser("usage%prog "+ "-f <zipfile> -d
    parser.add_option('-f', dest='zname', type='string', help='spe
    parser.add_option('-d', dest='dname', type='string', help='spe
    (options, args) = parser.parse_args()
    if(options.zname == None) | (options.dname == None):
        print parser.usage
        exit(0)
    else:
        zname = options.zname
        dname = options.dname
    zFile = zipfile.ZipFile(zname)
    passFile = open(dname)
    for line in passFile.readlines():
        password = line.strip('\n')
        t = Thread(target=extractFile, args=(zFile, password))
        t.start()
```