사이버범죄 관련
서버 정보 수집 도구 제작

Track	Digital Forensic
Category	Tech_06
Nick Name	L3ad0xFF

목 차

1. 개발 목적 1

2. 개발 SCRIPT 1

2.1 Main function - Initiate 1

2.2 Definition Function filecreate() 2

2.3 Definition Function setCsv() 2

2.4 Background 동작 (start.py) 4

2.5 참고 내용 5

3. LOOKUP THE NAMESERVICE (도메인 IP 주소 조회 GUI) 6

4. 첨부파일 8

5. 참고문헌 8

1. 개발 목적

- 특정 도메인에 연결된 서버의 IP주소를 수집하는 목적이다.
- IP주소 회신을 위해 특정 도메인을 질의할 DNS 서버를 직접 지정한다.
  질의할 DNS 서버는 국내가 아닌 국외 서버를 지정하여 이용한다.
- 특정 주기(1시간에 1번씩)를 가지고 자동으로 정보를 수집한 후, CSV 형식으로 저장하여 추적 관리한다.
- 설정한 주 DNS 서버가 작동을 중단되는 경우 수집이 불가능하므로 보조 DNS 서버를 설정하여 주 DNS 서버 기능 중단 시, 보조 DNS 서버를 이용한다.
- 해당 프로그램은 세션의 터미널 연결이 종료되더라도 프로세스가 백그라운드에서 동작한다.

2. 개발 Script

2.1 Main function - Initiate

```
count = 0
while True :
    start = time.time()
    if os.path.exists("fl0ckfl0ck.csv") == False :
        filecreate()
        count = setCsv(count)
        end = time.time() -start
        sleepTime = 3600 -end
        time.sleep(sleepTime)
    else :
        count = setCsv(count)
        end = time.time() - start
        sleepTime = 3600 - end
        #print (sleepTime)
        time.sleep(sleepTime)
```

- 수집하는 횟수를 기록하기 위해 count 이름의 변수를 선언 후, 0으로 초기화 한다.
- While 문 사용 : 사용자의 종료의사가 있기 전까지, 프로그램이 계속 동작하도록 한다.
- Start, end : while 문을 통해 해당 코드가 한번 수행 시, 동작 시간을 측정하고, 주어진 1시간의 조건에 최대한 오차를 줄이기 위해, while문 시작 시간에서 수행 시간 만큼을 제외하고, 다음에 수행할 whilie 문의 시간을 설정한다.
  ex) 22:00:00에 시작, 소요 시간 1초의 경우, 한번의 while문이 종료된 시간은 22:00:01.
  그대로 3600초(=1시간)후 동작 시, 23:00:01에 질의를 하기 때문에 3600-1=3559만큼만 간격을 제공하여, 23:00:00에 시작할 수 있도록 설정
- If os.path.exist : 수집한 IP 주소를 저장할 파일의 유무를 파악한다.
  1. 파일 없을 시 : filecreate() 함수 수행 후 setCsv()로 IP 주소 획득
  2. 파일 있을 시 : 존재한 파일에 바로 접근하여 setCsv()로 IP 주소 획득

2.2 Definition Function filecreate()

```
def filecreate() :
    target_file = open("fl0ckfl0ck.csv", "w")
    column_list = ["Num","    date", "          time", "          domain name", "       ip_addr","    DNS Server IP"]
    csvWriter = csv.writer(target_file)
    csvWriter.writerow(column_list)
    target_file.close()
    print ("\nfile create Success!!")
```

- 획득한 IP 주소를 저장할 때, 저장할 파일이 존재하지 않을 경우 해당 함수를 실행하여 'fl0ckfl0ck.csv'라는 이름의 파일을 생성한다.
  ★ 파일명을 사용자가 질의할 [domain name].csv로 설정할 수 있으나, 본 과제에서 요구한 domain name이 'fl0ckfl0ck.info'로 명시되어 있기 때문에, 'fl0ckfl0ck.csv'로 바로 설정
- python에서 제공하는 csv module을 사용하여, csv 파일에 쓰기를 할 수 있도록 하며, 앞으로 저장될 내용에 대한 간단한 column명 설정한다.

2.3 Definition Function setCsv()

```
def setCsv(num) :
    pid_num = list()
    count = num + 1
    csv_path = str(os.getcwd()) + str("/fl0ckfl0ck.csv")
    print (csv_path)
    print ("\n")
    target_File = open(csv_path, 'a', newline = '')
    qurey_list = list()
    now_date = time.strftime("%Y-%m-%d", localtime())
    now_time = time.strftime("%H:%M:%S", localtime())
    #gmt_date = time.strftime("%Y-%m-%d", gmtime())
    #gmt_time = time.strftime("%H:%M:%S", gmtime())
```

- 저장할 파일이 있기 때문에, 해당 함수로 진입을 하였고, 존재하는 파일의 경로명을 사용자 편의성을 제공하기 위해 출력을 한다. (Backgraoud로 실행 시, 출력 하지 않음)
- DNS Query 전송을 이용하여 IP를 획득하기 위해, 질의 시간을 localtime으로 획득한다.

```
domain = "fl0ckfl0ck.info"
resolver = dns.resolver.Resolver()
resolver.nameservers = ['8.8.8.8']

try :
    ip_query = resolver.query(domain, 'A')
    for rdata in ip_query :
        ip_addr = rdata.address
except :
    resolver.nameservers = ['64.6.64.6']
    ip_query = resolver.query(domain, 'A')
    for rdata in ip_query :
        ip_addr = rdata.address

qurey_list = [str(" ") + str(count), str("  ") + now_date, str("  ") + now_time + str(" (GMT+9)"),
              str("  ") + domain, str("  ")+ ip_addr, str("     ") + str(resolver.nameservers[0])]
print ("  ".join(qurey_list) +"\n")
csvWriter = csv.writer(target_File)
csvWriter.writerow(qurey_list)
target_File.close()
p = sub.Popen(['pidof', 'python3', 'DNS_Query.py'],stdout=sub.PIPE,stderr=sub.PIPE)
output, errors = p.communicate()
pid_num.append(output.decode('utf-8'))
pid_num = pid_num[0].split(" ")
print ("IF you want quit, Input the command 'kill -9 " + str(pid_num[0]) + "'")
return count
```

- 질의할 도메인 주소 설정 : domain = fl0ckfl0ck.info
- 질의 대상 DNS 서버 설정 : resolver.nameservers = [8.8.8.8]
- 대체 DNS 서버 설정 : except 구문 실행 시, resolver.nameservers = [64.6.64.6]
- Csv에 저장할 형식 설정 : list에 질의 횟수(=연번), 질의 날짜, 질의 시간(GMT+9), 질의대상 도메인, 도메인 IP 주소, 질의 대상 DNS 서버 IP 주소 순으로 저장한다.
- Foreground에서 동작 시, 실행 중인 프로그램의 pid를 획득 하고 사용자에게 프로그램 종료를 원할 경우, kill -9 [pid number]를 제공한다.
- 현재까지 질의한 횟수를 return하여, 다음 실행 시, 기록할 연번의 변수로 사용한다.

2.3.1 주 DNS와 보조 DNS

```
try :
    ip_query = resolver.query(domain, 'A')
    for rdata in ip_query :
        ip_addr = rdata.address
except :
    resolver.nameservers = ['64.6.64.6']
    ip_query = resolver.query(domain, 'A')
    for rdata in ip_query :
        ip_addr = rdata.address
```

- 앞서 설정한 DNS 서버에 domain 변수에 저장된 도메인 이름(=fl0ckfl0ck.info)의 IPv4에 해당되는 레코드(='A')만 요청한다.
- 수집한 A 레코드에 해당되는 모든 값들 중 IPv4에 해당되는 주소만 획득한다.
- 설정된 주 DNS 주소(=8.8.8.8)의 작동이 중지 되면 except 구문에서 DNS 서버를 '64.6.64.6'으로 설정(보조 DNS)하고 재 질의를 수행한다. 이 후 과정은 위와 동일하다.

```
root@kali:~/Desktop/BoB6_Forensic/tech/04_DNS# python3 DNS_Query.py

file create Success!!
/root/Desktop/BoB6_Forensic/tech/04_DNS/fl0ckfl0ck.csv


 1    2018-01-24    01:59:04 (GMT+9)    fl0ckfl0ck.info    120.50.131.112      64.6.64.6
```

```
root@kali:~/Desktop/BoB6_Forensic/tech/04_DNS# cat fl0ckfl0ck.csv
Num,    date,          time,          domain name,       ip_addr,    DNS Server IP
 1,  2018-01-24,  01:59:04 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    64.6.64.6
```

2.3.2 획득한 IP 주소 및 추가 정보 csv에 저장

```
qurey_list = [str(" ") + str(count), str("  ") + now_date, str("  ") + now_time + str(" (GMT+9)"),
              str("  ") + domain, str("  ")+ ip_addr, str("    ") + str(resolver.nameservers[0])]
print ("  ".join(qurey_list) +"\n")
csvWriter = csv.writer(target_File)
csvWriter.writerow(qurey_list)
target_File.close()

p = sub.Popen(['pidof', 'python3', 'DNS_Query.py'],stdout=sub.PIPE,stderr=sub.PIPE)
output, errors = p.communicate()
pid_num.append(output.decode('utf-8'))
pid_num = pid_num[0].split(" ")
print ("IF you want quit, Input the command 'kill -9 " + str(pid_num[0]) + "'")
return count
```

- List 형태로 획득 한 정보를 생성하고, csv module을 이용하여, 파일에 추가한다.
- 표준 입출력을 이용하여, 현재 실행 중인 본 코드의 pid를 수집한 후, 사용자가 종료를 원할 경우에 대한 행동방안을 제시한다.

2.3.3 주기 설정

```
count = 0
while True :
    start = time.time()
    if os.path.exists("fl0ckfl0ck.csv") == False :
        filecreate()
        count = setCsv(count)
        end = time.time() -start
        sleepTime = 3600 -end
        time.sleep(sleepTime)
    else :
        count = setCsv(count)
        end = time.time() - start
        sleepTime = 3600 - end
        #print (sleepTime)
        time.sleep(sleepTime)
```

- 1시간에 1번 씩, 정보를 수집하도록 프로그램에 일시정지 시간을 부여한다.
- 수집 시간의 오차를 최소화하기 위해, 앞서 언급했듯 '3600 – 코드 수행시간'을 sleep의 시간으로 설정한다.

2.4 Background 동작 (start.py)

```
import os
import sys

try :
    os.system("nohup python3 DNS_Query.py &")
except :
    print ("Can't run DNS_Query.py")
```

- 도메인 이름을 질의하여 IP 주소를 획득하는 python code의 파일명을 DNS_Query.py라 명명하였다.

- 수집 목적의 코드를 Background에서 실행하여, 터미널 연결이 종료 되어도 실행 할 수 있도록 한다.

- Start.py를 실행하면, 수집 목적의 DNS_Query.py가 background로 동작한다.

2.5 참고 내용

- Primary DNS Server 8.8.8.8 : Google DNS Server

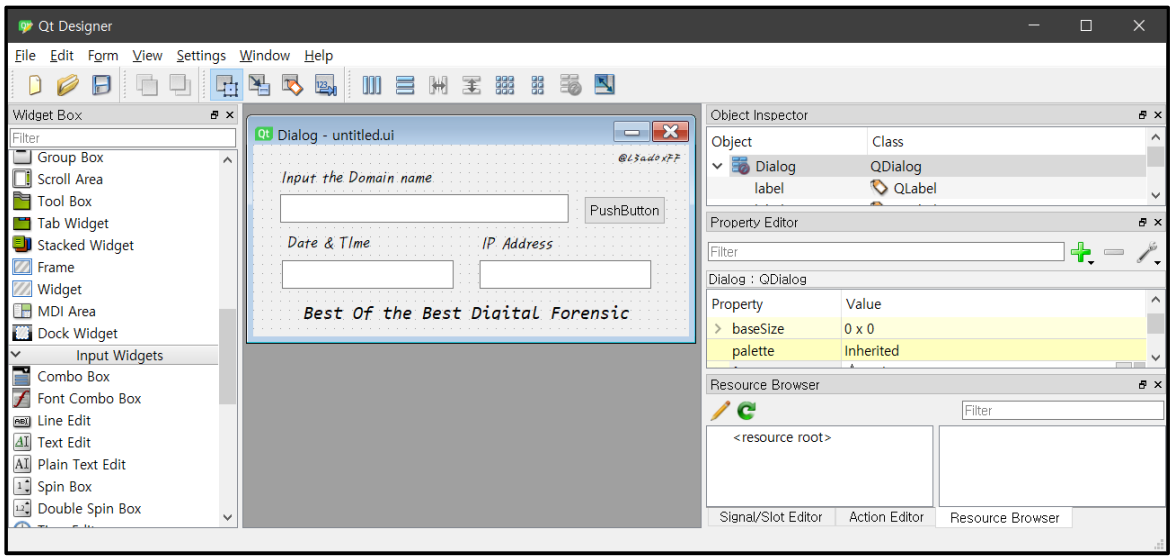
- Secondary DNS Server 64.6.64.6 : Verisign DNS Server

- 2018. 01. 15 02:14 진행상황

```
root@kali:~/Desktop/BoB6_Forensic/tech/04_DNS# cat fl0ckfl0ck.csv
Num,    date,           time,           domain name,       ip_addr,    DNS Server IP
 1,  2018-01-24,  02:04:17 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 2,  2018-01-24,  03:04:17 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 3,  2018-01-24,  04:04:17 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 4,  2018-01-24,  05:04:17 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 5,  2018-01-24,  06:04:17 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 6,  2018-01-24,  07:04:17 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 7,  2018-01-24,  08:04:18 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 8,  2018-01-24,  09:04:18 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 9,  2018-01-24,  10:04:18 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 10,  2018-01-24,  11:04:18 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 11,  2018-01-24,  12:04:18 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 12,  2018-01-24,  13:04:20 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 13,  2018-01-24,  14:04:20 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 14,  2018-01-24,  15:04:20 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 15,  2018-01-24,  16:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 16,  2018-01-24,  17:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 17,  2018-01-24,  18:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 18,  2018-01-24,  19:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 19,  2018-01-24,  20:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 20,  2018-01-24,  21:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 21,  2018-01-24,  22:04:45 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 22,  2018-01-24,  23:04:46 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 23,  2018-01-25,  00:04:46 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 24,  2018-01-25,  01:04:46 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
 25,  2018-01-25,  02:04:46 (GMT+9),  fl0ckfl0ck.info,  120.50.131.112,    8.8.8.8
```

3. Lookup the NameService (도메인 IP 주소 조회 GUI)

3.1 QT Designer



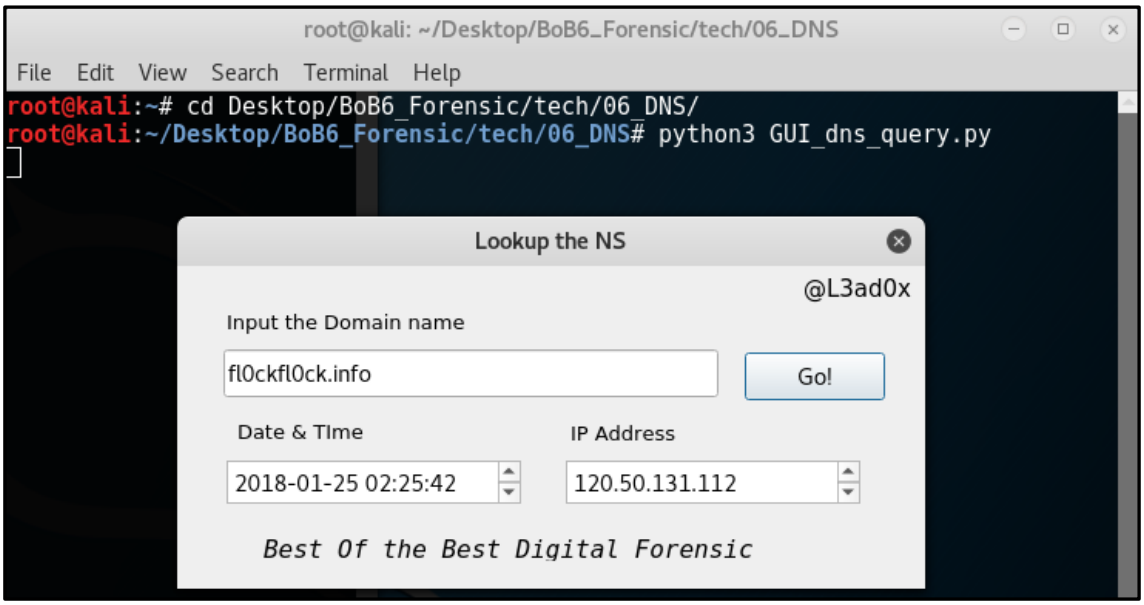
- QT Designer를 이용하여 기본적인 틀을 생성 후 파일명.ui로 저장한다.

3.2 Convert to py code

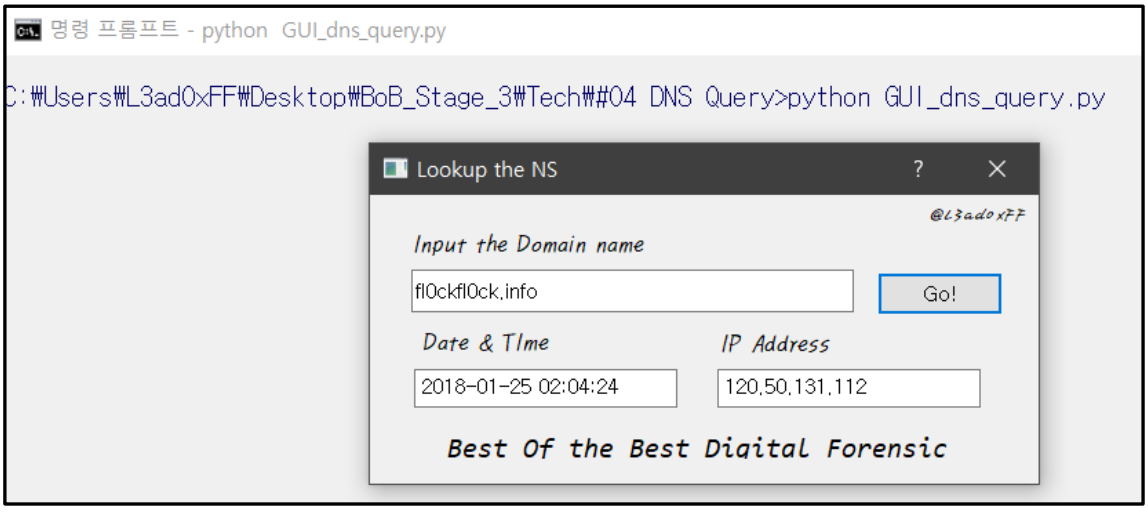
```
        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)
        self.pushButton.clicked.connect(self.query)

    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Lookup the NS"))
        self.pushButton.setText(_translate("Dialog", "Go!"))
        self.label.setText(_translate("Dialog", "Input the Domain name"))
        self.label_2.setText(_translate("Dialog", "Date & TIme"))
        self.label_3.setText(_translate("Dialog", "IP Address"))
        self.label_4.setText(_translate("Dialog", "Best Of the Best Digital Forensic"))
        self.label_5.setText(_translate("Dialog", "@L3ad0xFF"))

    def query(self):
        qurey_list = list()
        domain = self.lineEdit.text()
        now_date = time.strftime("%Y-%m-%d", localtime())
        now_time = time.strftime("%H:%M:%S", localtime())
        #gmt_date = time.strftime("%Y-%m-%d", gmtime())
        #gmt_time = time.strftime("%H:%M:%S", gmtime())
        the_time = str(now_date) + str(" ") +str(now_time)
        self.textBrowser.setText(the_time)
        resolver = dns.resolver.Resolver()
        resolver.nameservers = ['8.8.8.8']

        try :
            ip_query = resolver.query(domain, 'A')
            for rdata in ip_query :
                ip_addr = rdata.address

        except :
            resolver.nameservers = ['64.6.64.6']
            ip_query = resolver.query(domain, 'A')
            for rdata in ip_query :
                ip_addr = rdata.address

        self.textBrowser_2.setText(str(ip_addr))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Dialog = QtWidgets.QDialog()
    ui = Ui_Dialog()
```

- 버튼을 누르면 선언한 함수가 동작하도록 설정한다.
- .선언한 함수 query에는 사용자가 입력한 도메인을 함수의 parameter로 받아서, DNS 서버에 질의를 하여 IP 주소를 획득한다.
- 질의 날짜 및 시간과 획득한 IP를 창에 출력한다.

3.3 출력 화면



[At linux]



[At window]

4. 첨부파일

- fl0ckfl0ck.csv : 2018. 01. 24 02:55:17 ~ 2018. 01. 28 01:58:47 까지 flockflock.info 도메인에 대한 IP 주소 수집 기록

5. 참고문헌

- [Python] DNS Request : https://blog.naver.com/aka_handa/10188051988
- Python dns query : http://apollo89.com/wordpress/?p=3680
- DNS Server IP list : https://www.lifewire.com/free-and-public-dns-servers-2626062.