

[Pharming]

[PHARMING을 통해 얻은 IP, 공인 인증서 데이터 파싱]

[담당 : 최원영 멘토님]

개요

여러분은 어쩌다보니 수사관이 되었습니다
그러다가 팀장님이 파밍사건 지시를 합니다

- 사건 경위

피의자는 한국인(해외거주민 포함)을 상대로 공인인증서를 탈취하는 악성코드를 유포하였습니다.
악성코드에 감염된 피해자의 PC에 저장된 공인인증서는 [IP주소].zip 파일의 형태로 해외 서버(<http://107.174.85.141/cert>)에 업로드되어 있습니다
다행히 디렉토리리스팅 취약점으로 인하여 공인인증서 들은 노출되어 있습니다
빠른 시일내에 폐기처분을 해야합니다

- 팀장 지시사항

1. 피해자의 일련번호 - 피해시각(서버에 피해자의 공인인증서가 업로드된 시간) - 피해자의 이름 - 은행명 - 계좌번호 - IP주소 - 피해자의 현재 소재지(국가만) 를 데이터베이스(mysql)에 저장하시오.
2. 은행별 유출된 공인인증서의 갯수를 계산하시오

- 제출방법

2018. 8. 31. 23:59:59 이메일(fl0kfl0ck@hotmail.com)로 제출할 것

각 지시사항을 보고서 형태로 제출하고 분석과정을 세세히 기록할 것(첨부해야할 파일 : 보고서, 사용한 소스코드, 데이터베이스에서 csv 형태로 익스포트한 파일)

제목을 “[BoB7기][트랙명][이름][실습과제]”로 할 것, 시간엄수하세요

!!) 파이썬 등 사용하지 않아도 됩니다. 근성 인정하나 이왕이면 파이썬 이용하세요

개발환경

- Windows 10
- Python 3.6
- MS office Excel
- Mysql Workbench

프로젝트 구현

1. Pharming Server zip파일 다운로드

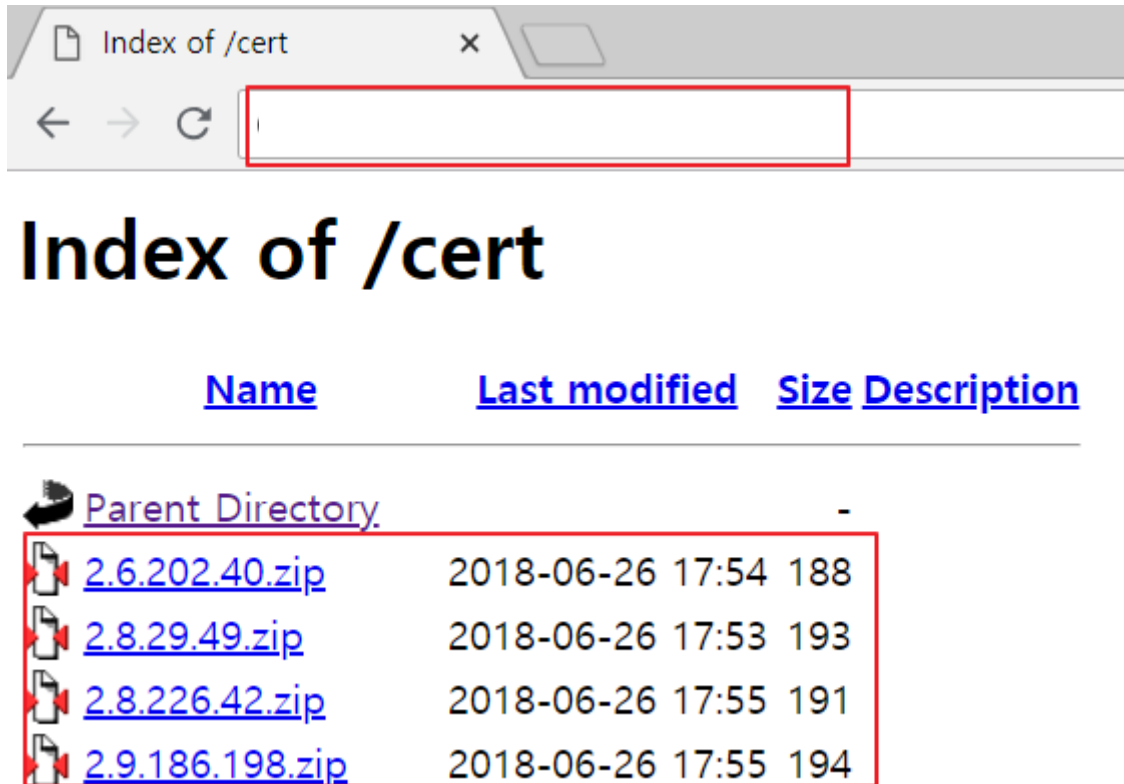


Figure 1 공인 인증서 서버 파일

공인 인증서가 저장되어 있는 (아이피).zip 파일이 저장되어 있는 서버를 확인할 수 있다. 해당 파일을 selenium을 사용해서 36833개의 인증서를 다운로드 하였다.

이름	수정된 날짜	유형	크기
2.6.202.40.zip	2018-08-27 오후 9:58	압축(ZIP) 파일	1KB
2.8.29.49.zip	2018-08-27 오후 9:58	압축(ZIP) 파일	1KB
2.8.226.42.zip	2018-08-27 오후 9:58	압축(ZIP) 파일	1KB
2.9.186.198.zip	2018-08-27 오후 9:58	압축(ZIP) 파일	1KB
2.11.166.3.zip	2018-08-27 오후 9:59	압축(ZIP) 파일	1KB
2.11.225.47.zip	2018-08-27 오후 9:59	압축(ZIP) 파일	1KB
2.12.35.126.zip	2018-08-27 오후 9:59	압축(ZIP) 파일	1KB
2.12.128.193.zip	2018-08-27 오후 9:59	압축(ZIP) 파일	1KB

Figure 2 파일 다운로드

Selenium을 이용해서 다운로드 한결과 10시간이 걸렸다. 직접 클릭하듯 해당 URL에 접속하여 다운로드 하였다. 시간적으로 비효율적인 방법으로 생각되었다. 다음부터는 대용량의 파일을 다운로드 받는 경우 urllib, urlopen과 같은 방법으로 사용하려 한다.

```
def seleniumDownload():
    driver = webdriver.Chrome()
    driver.get("http://107.174.85.141/cert/")

    i = 4
    while True:
        try:
            download_url = '/html/body/table/tbody/tr[' + str(
                i) + ']/td[2]/a'
            driver.find_element_by_xpath(download_url).click()
            i = i+1
        except:
            print("end")
```

Figure 3 파일 다운로드

Selenium을 사용하여 (아이피).zip 파일을 다운로드 한다.

2. 다운로드 파일 압축 해제 및 DB 업로드

```
def search(dirname):
    sql_insertlist = []
    count = 0
    try:
        filenames = os.listdir(dirname)
        #print (len(filenames))
        #zzzexit(1)
        for filename in filenames:
            full_filename = os.path.join(dirname, filename)
            if os.path.isdir(full_filename):
                search(full_filename)
            else:
                ext = os.path.splitext(full_filename)[-1]
                if ext == '.zip':
                    #unzip_file
                    cert = unZip(full_filename)
                    sql_insertlist.append(cert)
                    count = count + 1
                if len(sql_insertlist) >= 2000 or count >= len(filenames):
                    # print (sql_insertlist)
                    parming.dbquery.inputDB(sql_insertlist)
                    sql_insertlist = []
```

Figure 4 압축해제 및 DB 업로드

다운로드 한 파일의 압축을 풀고 입력된 IP, 은행명, 업로드 날짜 정보를 DB에 입력한다.

```
def inputDB(sql_list): #insert log file into mysql database
    conn = pymysql.connect(host='localhost', user='root', password='1234',
                           db='cert_info', charset='utf8')

    curs = conn.cursor()
    sql = "insert into cert(date, name, account_number, bank_info, country, IP) values (%s, %s, %s, %s, %s, %s)"

    for i in range(0, len(sql_list)):
        #date..bank_info..accountnumber..ip..country
        curs.execute(sql, (sql_list[i][7][:-2], sql_list[i][0], sql_list[i][1], sql_list[i][2], sql_list[i][5], sql_list[i][6]))
        #curs.execute(sql, ('2018-08-26', 'name', 'kkr', 'ip'))
    print("dataBase Insert")
    conn.commit()
```

Figure 5 DB 파일 저장

DB에 signCert.txt 파일의 내용을 파싱해서 저장한다.

```

#공인인증서 정보 추출
def find_people_info(text):
    #print(text)
    reg_info = '^cn=([가-힣]+\(\)\([0-z]+\),ou=([a-zA-Z]+\),ou=([a-zA-Z]+\),o=([a-zA-Z]+\),c=([a-zA-Z]+)'
    cert_value = re.findall(reg_info, text)

    ...

    def findIP_time(fileName(ip정보))
    return type
    ip data
    date data
    ...

#Name_계좌번호
cert_list=[]

for i in range(0, len(cert_value[0])):
    cert_list.append(cert_value[0][i])

# print (cert_list)
return cert_list

```

Figure 6 signCert.txt 파싱

DB에 signCert.txt 파일을 파싱한다.

seq	date	name	account_number	bank_info	country	IP
22001	2018-06-26 17:05:00	홍익현	26722557442828053766	shinhan	kr	100.100.13.11
22002	2018-06-26 17:05:00	남궁진일	76513505351718136427	ieonbuk	kr	100.100.195.172
22003	2018-06-26 17:05:00	남궁경주	87038123328852058123	daegu	kr	100.101.120.45
22004	2018-06-26 17:05:00	남궁진희	88304108312607183285	awanciu	kr	100.101.30.242
22005	2018-06-26 17:05:00	권만세	33554028560838647378	kookmin	kr	100.101.43.251
22006	2018-06-26 17:05:00	권수미	28332136276645034331	ieonbuk	kr	100.102.98.32

Result Grid
count(*)
36833

Figure 7 저장된 DB

DB에 입력된 signCert.txt 정보를 확인하였다.

3. IP별 국가 조회

```
def getCountry(ip_info):
    api_key = '2018083111331754935052'

    url = 'http://whois.kisa.or.kr/openapi/ipascc.jsp?query=' + ip_info + '&key=' + api_key + '&answer=json'
    # url = 'http://whois.kisa.or.kr/openapi/ipascc.jsp?query=211.101.23.45&key=2018083111331754935052&answer=json'
    req = requests.get(url)
    text = req.text

    data = json.loads(text)
    return data["whois"]["countryCode"]
```

Figure 8 whois api로 국가코드 조회

내 Gmail 계정으로 로그인을 한다. 수신한 메일중 "f10ckf10ck@hotmail.com"으로부터 수신된 메일의 정보가 있는지 파악

```
# 단축 URL 가져오기
def find_shortUrl(self, part):
    try:
        text = ''
        #body = part.get_payload()
        #text += body
        text += part.get_payload()
        re_x = 'https?:\/\/(?:[-\w.]|(?:%[\da-fA-F]{2}))+\/?.{4,7}?'
        urls = re.findall('https?:\/\/(?:[-\w.]|(?:%[\da-fA-F]{2}))+\/?.{4,7}', text)
        for url in urls:
            for short_url in short_url_list:
                if short_url in url:
                    in_url = short_url + url[len(short_url):len(short_url) + short_url_list[short_url]]
                    print("connect url info : ", in_url)
                    # 이미지 다운로드
                    self.checkUrl(in_url)
    except UnicodeDecodeError:
        print("From : UnicodeDecodeError Next\n")

short_url_list = {
    "https://hoy.kr/": 4,
    "http://hoy.kr/": 4,
    "https://bit.ly/": 7,
    "http://bit.ly/": 7,
    "https://bitly.kr/": 4,
    "http://bitly.kr/": 4,
    "https://goo.gl/": 6,
    "http://goo.gl/": 6
}
```

Figure 9 whois API 사용

Whois API를 이용하여 국가코드 검색

```
def retIPList(ipTime):
    iplist = []
    country = getCountry(ipTime.split('\t')[0][:4])

    iplist.append(ipTime.split('\t')[0][:4])
    iplist.append(ipTime.split('\t')[1][:1])
    iplist.append(country)
    return iplist

def inputIPDB():
    fp = open('ipinfo.txt', 'r')

    conn = pymysql.connect(host='localhost', user='root', password='1234',
                           db='cert_info', charset='utf8')
    curs = conn.cursor()
    input_IPdbList = []
    count = 0
    while (True):
        ipTime = fp.readline()

        count = count + 1

        if (len(input_IPdbList) >= 3000 or ipTime==''):
            for i in range(0, len(input_IPdbList)):
                curs.execute(sql, (input_IPdbList[i][0], input_IPdbList[i][1], input_IPdbList[i][2]))

            input_IPdbList = []
            print('input db')
            conn.commit()
            if (ipTime==''):
                break
        iplist = retIPList(ipTime)
        input_IPdbList.append(iplist)
```

Figure 10 IP의 국가 정보 DB 입력

Whois API로 확보한 국가코드 정보를 DB에 입력한다.

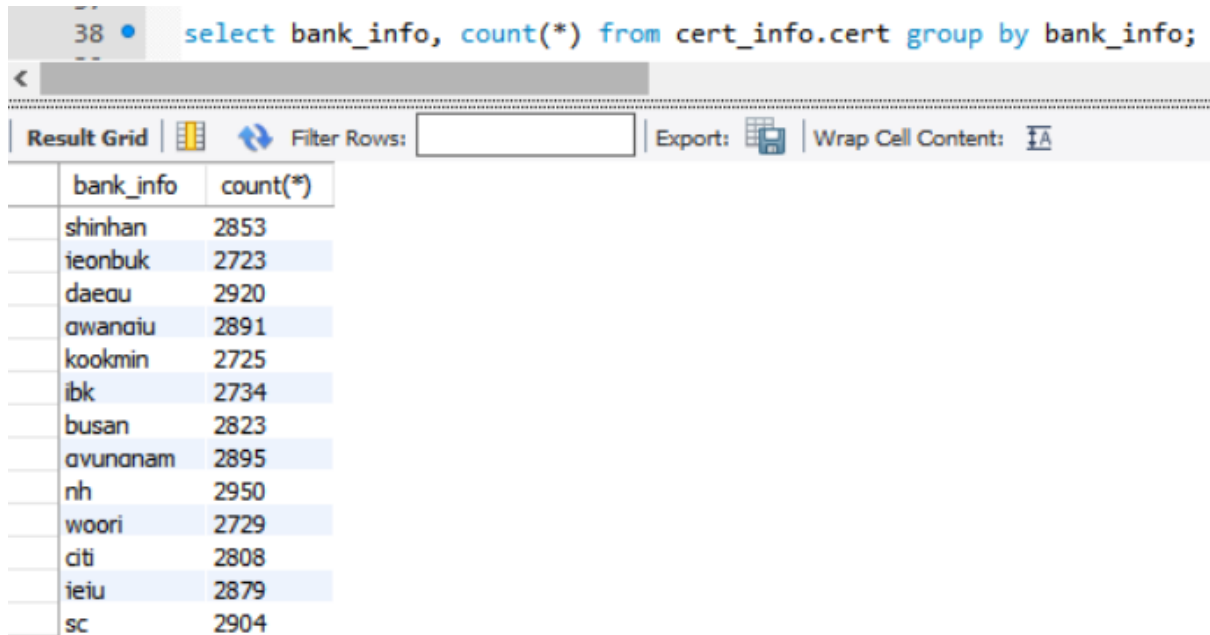
seq	ip_addr	date	country
1	2.6.202.40	2018-06-26 17:54:00	FR
2	2.8.29.49	2018-06-26 17:53:00	FR
3	2.8.226.42	2018-06-26 17:55:00	FR
4	2.9.186.198	2018-06-26 17:55:00	FR
5	2.11.166.3	2018-06-26 17:52:00	FR
6	2.11.225.47	2018-06-26 17:55:00	FR
7	2.12.35.126	2018-06-26 17:52:00	FR

count(*)
36834

Figure 11 IP당 국가코드

인증서별 입력된 국가코드와 수를 확인 할 수 있다.

4. 유출국가 및 유출 은행수



```
38 • select bank_info, count(*) from cert_info.cert group by bank_info;
```

bank_info	count(*)
shinhan	2853
jeonbuk	2723
daegu	2920
awantui	2891
kookmin	2725
ibk	2734
busan	2823
avunnam	2895
nh	2950
woori	2729
citi	2808
iei	2879
sc	2904

Figure 12 은행별 유출 건수

은행별 유출된 공인인증서 수가 다음과같이 DB로 확인 되었다.

41 • `select cert.*, ip_info.country from cert join ip_info on cert.IP = ip_info.ip_addr;`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

seq	date	name	account_number	bank_info	country	IP	country
22019	2018-06-26 17:05:00	문선용	17802241101544211721	nh	kr	100.124.22.63	none
22020	2018-06-26 17:05:00	한종철	10083425254784674272	ietu	kr	100.126.97.213	none
22021	2018-06-26 17:05:00	권선영	83146201554651817386	kookmin	kr	100.127.178.173	none
22022	2018-06-26 17:05:00	채재섭	75831716708730578745	ibk	kr	100.128.39.40	US
22023	2018-06-26 17:05:00	강경희	14665002668163372172	citi	kr	100.129.134.69	US
22024	2018-06-26 17:05:00	이준현	33440171642037282783	ietu	kr	100.129.202.125	US
22025	2018-06-26 17:05:00	소현	58602747708531058821	sc	kr	100.13.70.208	US
22026	2018-06-26 17:05:00	장규리	35264530673358715657	nh	kr	100.132.116.118	US
22027	2018-06-26 17:05:00	방순영	22251501026652507126	shinhan	kr	100.132.46.186	US
22028	2018-06-26 17:05:00	홍은민	55736562786722606841	citi	kr	100.134.56.220	US
22029	2018-06-26 17:05:00	박연주	81342221408333035376	awangiu	kr	100.137.150.238	US
22030	2018-06-26 17:05:00	백영진	40801544352531165872	ietu	kr	100.143.174.21	US
22031	2018-06-26 17:05:00	기제성	35437260066563616707	sc	kr	100.144.181.62	US
22032	2018-06-26 17:05:00	조하진	43843220065001278047	ietu	kr	100.145.94.100	US
22033	2018-06-26 17:05:00	권유정	23872055617550625028	sc	kr	100.152.232.87	US
22034	2018-06-26 17:05:00	탁수지	34436226685881620871	nh	kr	100.156.110.180	US
22035	2018-06-26 17:05:00	강정석	45611876873001518024	teonbuk	kr	100.156.249.166	US

Figure 13 IP별 국가정보

IP별 국가정보를 조회한 결과를 확인 하였다.

40 • `select country, count(*) from ip_info group by country;`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

country	count(*)
AU	399
TH	77
CA	594
PR	10
AR	175
TW	322
PH	51
NZ	56
NL	394
SY	4
PS	9
HR	20
CZ	95
CY	12
SK	27
BA	9

Figure 14 인증서가 유출된 국가

인증서가 유출된 국가정보를 확인 할 수 있다.

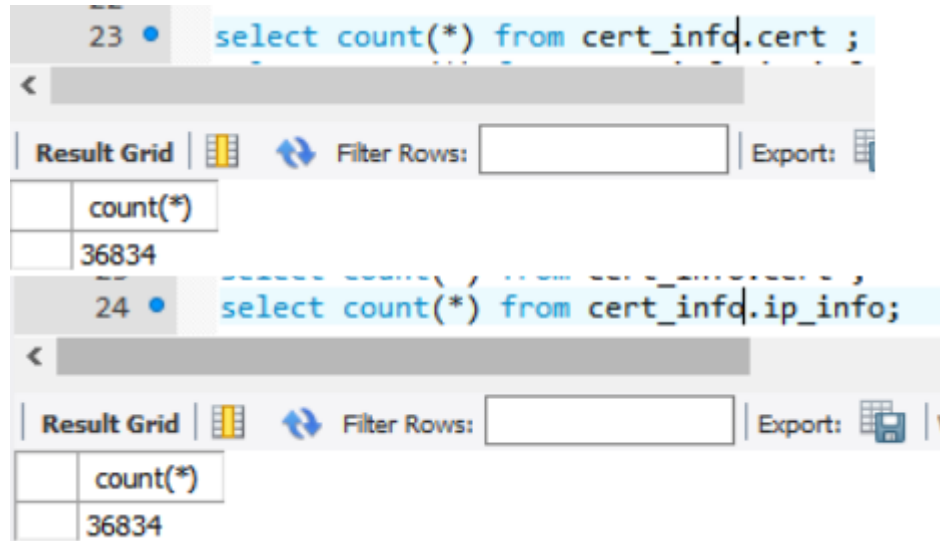


Figure 15 다운로드 받은 수 확인

저장한 파일수를 36834로 맞추었다. 처음 다운로드 받았던 파일을 실수로 종료하고 실행하는 경우가 발생하여 IP하나를 찾지 못하여 IP검색을 통한 1개의 IP를 찾아 내어 다시 입력하였다.

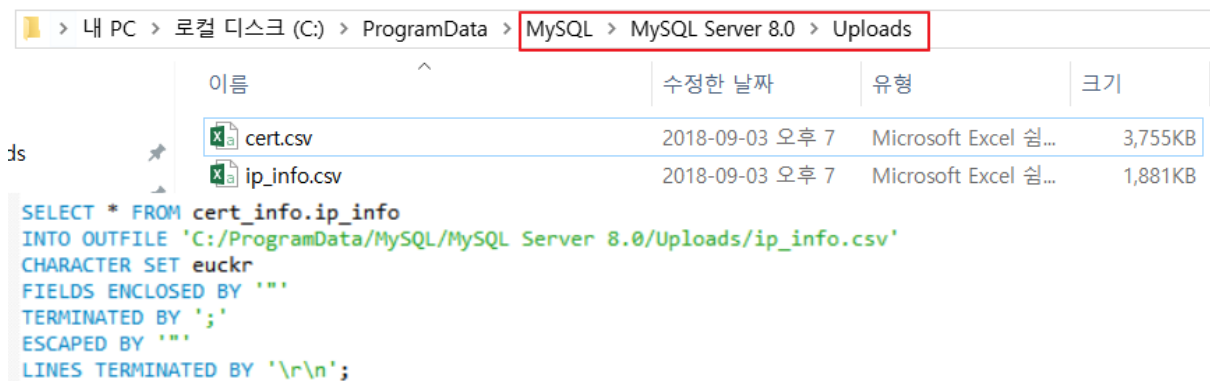


Figure 16 mysql to csv 생성

`SHOW VARIABLES LIKE 'secure_file%';` 명령어로 얻은 경로에서 Mysql DB의 내용을 csv 파일로 생성하여 저장

프로젝트 결과 및 정리

1. 완성도 및 결과

이번 프로젝트의 완성도는 80% 정도로 생각 된다.

- 1) 파일 다운로드 방법 : Selenium을 사용한 .zip 파일 다운로드
- 2) Zip파일 압축 해제후 DB 업로드
- 3) WHOIS 국가 코드 조회 및 DB 업로드
- 4) DB 쿼리를 통한 유출 국가 정보 확인

2. 프로젝트를 통한 느낀점

이번 프로젝트를 통해서 지금까지 학습했던 , DB 쿼리 사용법, 파일을 반복적으로 입 출력하는 방법과, 새로운 WHOIS API 사용법을 학습 할 수 있었다. 앞의 과정을 통해서 학습한 내용덕에 간단하게 구현 할 수 있었다.