

FTP 프로토콜 연결 및 자동화

모의해킹 업무, 취약점 분석, 컨설팅 업무!!!

ISMS-P 인증 - 물리, 관리, 기술 (인프라진단, 모의해킹), 보안관제 업무(침해대응)

1번째, vmware 워크스테이션을 다운로드 받아 설치 (재시작 필요할 수 있음)

회원 가입 후에 → 다시 아래 링크에서 다운로드 시작

My Downloads > Free Software Downloads available [HERE](#) > VMware Workstation Pro 선택 > 이후 설치는 아래쪽

<https://support.broadcom.com/group/ecx/downloads>

support.broadcom.com

<https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Workstation%20Pr...>

2번째, 반디집 설치 (압축/해제 프로그램)

B Bandisoft 반디집 공식 홈페이지 · **ALZ, EGG, 7Z, RAR** 지원 무료 압축 프로그램

3번째, Metasploitable v2 다운로드 (2개 링크 중 하나 선택 다운로드) > vmwares 폴더 만들고 보관

<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/metasploitable-linux-2.0.0.zip/download>

<https://drive.google.com/file/d/1QM5SFoGZBSaWko-RJO0v7ZTGNUv4DMMm/view?usp=sharing>





가상환경 이미지 압축 해제 후



vmware에서 Open 해서 해당 압축 해제 했던 것 vmx 파일을 불러온다.



파이썬 FTP 자동화

ftplib 라이브러리

- **ftplib** 모듈은 FTP 클라이언트 세션을 생성하고, 관리하는 기능을 제공
- 이 모듈을 사용하여 파일을 업로드, 다운로드하고, 디렉토리를 조작하고, 서버의 파일 리스트를 조회 가능

- FTP 라이브러리 ftplib 임포트

```
import ftplib
```

- FTP 서버 연결 시도
- hostname은 FTP 서버 도메인 이름 또는 아이피 주소

```
ftp = ftplib.FTP(hostname)
```

- FTP 서버 로그인

```
ftp.login('msfadmin','msfadmin')
```

- 디렉터리 리스트 가져오기

```
ftp.retrlines('LIST')
```

- 종료

```
ftp.quit()
```

```
import ftplib hostname = "169.254.231.101" ftp = ftplib.FTP(hostname) ftp.  
login('msfadmin','msfadmin') ftp.retrlines('LIST') ftp.quit()
```

주요 FTP 명령어

1. LIST

- 디렉토리 내용과 파일 정보를 상세하게 나열합니다. 출력은 서버에 따라 다르며 일반적으로 파일 권한, 소유자, 크기, 최종 수정 날짜 등이 포함됩니다.
- 예: `ftp.retrlines('LIST')`

2. NLST

- 현재 디렉토리 또는 지정된 디렉토리의 파일 및 서브디렉토리 목록만을 간단히 나열합니다. `LIST` 보다 더 간결한 정보를 제공합니다.
- 예: `ftp.nlst()`

3. RETR

- 서버에서 파일을 검색하여 다운로드합니다. 바이너리 파일에 주로 사용되지만, 텍스트 파일에도 사용할 수 있습니다.
- 예: `ftp.retrbinary('RETR example.txt', open('example.txt', 'wb').write)`

4. STOR

- 로컬 시스템에서 파일을 가져와 서버에 저장합니다. 파일 업로드 시 사용됩니다.
- 예: `ftp.storbinary('STOR newfile.txt', open('newfile.txt', 'rb'))`

5. DELE

- 서버에서 특정 파일을 삭제합니다.
- 예: `ftp.delete('obsolete.txt')`

6. MKD

- 서버에 새 디렉토리를 만듭니다.
- 예: `ftp.mkd('new_directory')`

7. RMD

- 서버에서 디렉토리를 삭제합니다.
- 예: `ftp.rmd('old_directory')`

8. PWD

- 현재 작업 디렉토리의 경로를 표시합니다.
- 예: `ftp.pwd()` - `print(ftp.pwd())`

9. CWD

- 현재 작업 디렉토리를 변경합니다.
- 예: `ftp.cwd('new_directory')`

아래에서 처음에는 mkd → rmd로 테스트로..

```
import ftplib hostname = "192.168.81.130" ftp = ftplib.FTP(hostname)
ftp.login('msfadmin','msfadmin') print(ftp.pwd()) ftp.retrlines('LIST')
print("=====") ftp.rmd("new_folder") ftp.retrlines('LIST')
print("=====") ftp.cwd("vulnerable") ftp.retrlines('LIST')
ftp.quit()
```

주요 FTP 메서드

retrlines 메서드

- FTP 서버로부터 텍스트 데이터를 읽어오는데 사용
- 서버에 텍스트 기반의 명령(예: `LIST`, `NLST`)을 보내고, 서버로부터 받은 응답을 라인 별로 처리

storbinary 메서드

- 로컬의 바이너리 파일을 FTP 서버에 업로드할 때 사용
- 'STOR filename' 형식으로 업로드할 파일의 이름을 지정

```
import ftplib def upload_file(ftp, filename): with open(filename, 'rb') as f:
ftp.storbinary(f'STOR {filename}', f) hostname = "192.168.81.130" ftp =
ftplib.FTP(hostname) ftp.login('msfadmin','msfadmin') upload_file(ftp, 'ma
lwares1.xlsx') ftp.quit()
```

retrbinary 메서드

- FTP 서버에서 바이너리 파일을 다운로드할 때 사용

- 'RETR filename' 형식으로 파일을 지정

```
import ftplib
def download_file(ftp, filename):
    with open(filename, 'wb') as f:
        ftp.retrbinary(f'RETR {filename}', f.write)
    ftp = ftplib.FTP("169.254.231.101")
    ftp.login('msfadmin', 'msfadmin')
    download_file(ftp, 'example.txt')
    ftp.quit()
```

- 다른 이름으로 저장

```
import ftplib
def download_file(ftp, filename):
    with open(f'new_{filename}', 'wb') as f:
        ftp.retrbinary('RETR ' + filename, f.write)
    ftp = ftplib.FTP("169.254.231.101")
    ftp.login('msfadmin', 'msfadmin')
    download_file(ftp, 'example.txt')
    ftp.quit()
```

정의 함수로 만든 사례

```
import ftplib
def upload_file(ftp, filename):
    with open(filename, 'rb') as file:
        ftp.retrbinary(f'RETR {filename}', f.write)
def main():
    hostname = "192.168.81.142"
    ftp = ftplib.FTP(hostname)
    ftp.login('msfadmin', 'msfadmin')
    upload_file(ftp, 'result.xlsx') #리스트를 확인
    ftp.retrlines('LIST')
    ftp.quit()
if __name__ == "__main__":
    main()
```

압축

- 윈도우 폴더 및 파일을 압축해서 FTP 서버로 업로드(백업 시스템)

```
import zipfile
```

ZIP 파일 생성 및 파일 추가

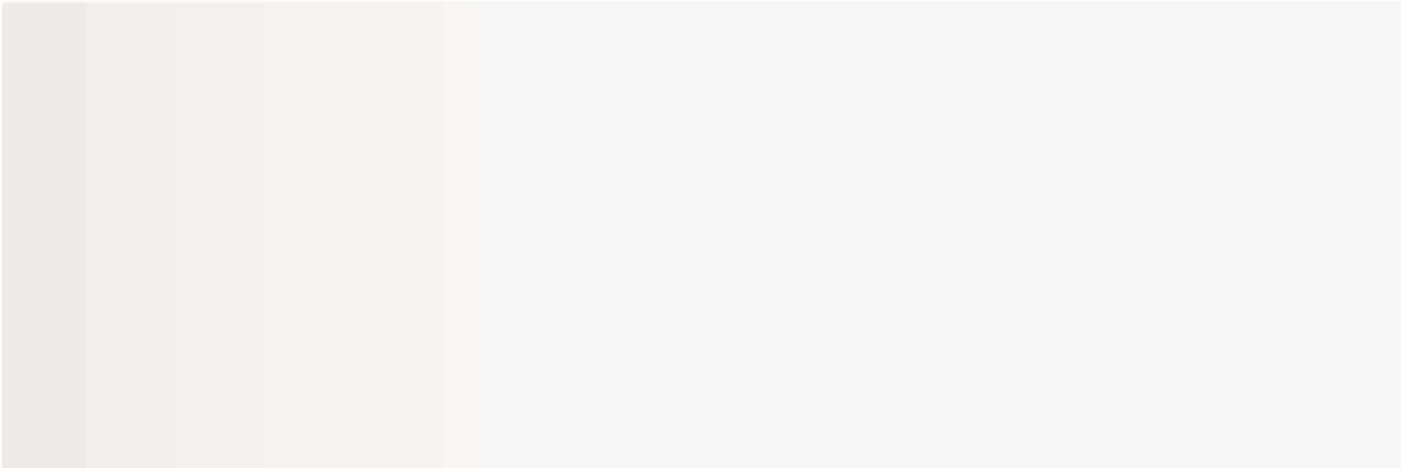
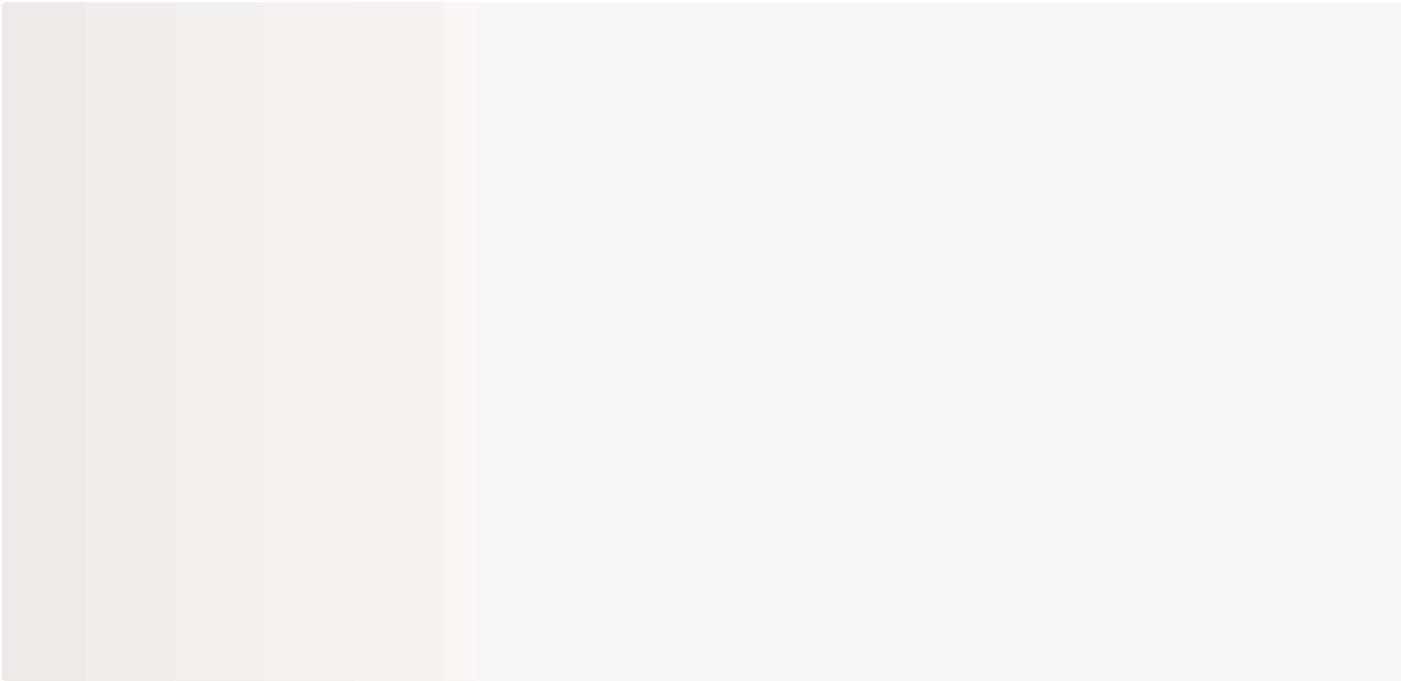
- `zipfile.ZipFile` 클래스를 사용하여 ZIP 파일 생성 가능
- `zip_file.write()` 메서드는 지정된 파일을 ZIP 파일에 추가

```
import zipfile import os RESULT_DIR = 'rss_results' zip_file =
zipfile.ZipFile('results.zip', 'w') for root, dirs, files in
os.walk(RESULT_DIR): for file in files: zip_file.write(os.path.join(root,
file)) zip_file.close()
```

4시 50분까지 보고!!!! 진행률보면서 갑시다!!!

퀴즈 1. RSS 서비스에서 `list.txt` 파일의 정보를 수집한다. (주의 : 데이터가 매칭이 안되면 제외) 2. 수집된 정보를 `xlsx` 파일에 리스트별로 저장한다. 3. `uploads` 폴더에 3개의 `xlsx` 파일이 저장될 것이다. 4. 결과 값 디렉터리를 압축을 한다. 5. `zip` 파일을 FTP에서 전송한다. 6. (후에 추가) 메일로도 전송을 한다. `list.txt` 파일 내용 <https://www.dailysecu.com/rss/allArticle.xml>
<https://www.dailysecu.com/rss/S1N2.xml>
<https://www.dailysecu.com/rss/clickTop.xml>

결과 사진 사례



```

import feedparser from openpyxl import Workbook import os import zipfile
import ftplib ##RSS 서비스 대상으로 정보 수집 및 엑셀 파일 저장 RESULT_DIR =
"results" if not os.path.exists(RESULT_DIR): os.makedirs(RESULT_DIR) with
open('list.txt', 'r', encoding='UTF-8') as file: rss_urls =
file.readlines() for index, url in enumerate(rss_urls): feed =
feedparser.parse(url) titles = [] links = [] descriptions = [] authors =
[] pubDates = [] for entry in feed.entries: titles.append(entry.title)
links.append(entry.link) descriptions.append(entry.description)
authors.append(entry.author) pubDates.append(entry.published) # 워크북과 워
크시트를 생성합니다. wb = Workbook() ws = wb.active ws.title = f"{index+1}번
째 Data" # 첫 번째 행에 헤더를 작성합니다. headers = ['Title', 'Link',
'Description', 'Author', 'Published'] ws.append(headers) # 데이터를 행 단위
로 작성합니다. for i in range(len(titles)): ws.append([titles[i], links[i],
descriptions[i], authors[i], pubDates[i]]) # 엑셀 파일로 저장합니다.
file_path = os.path.join(RESULT_DIR, f'{index+1}_result.xlsx')
wb.save(file_path) ##결과 값을 result.zip 파일로 저장 zip_file =
zipfile.ZipFile("result.zip", "w") for root, dirs, files in
os.walk(RESULT_DIR): for file in files: zip_file.write(os.path.join(root,
file)) zip_file.close() ##FTP 서버에 결과 값을 보낸다. hostname =
"192.168.81.130" ftp = ftplib.FTP(hostname)
ftp.login('msfadmin', 'msfadmin') ftp.retrlines('LIST') with
open("result.zip", "rb") as f: ftp.storbinary(f"STOR result.zip", f)
print(f"현재작업디렉터리 : {ftp.pwd()}") ftp.retrlines('LIST') ftp.quit()

```

만약 pandas로 한다면.. 데이터 수집까지!!