**SQL Injection**

SQL Injection의 종류로는 크게 In-Band SQL Injection과 Inferential SQL Injection로 구분할 수 있습니다.

- In Band SQL Injectioin

: 결과값을 얻을 수 있는 환경에서 진행되는 공격(classic SQL Injection)

eg) Union Based SQL Injection, Error Based SQL Injection

- Inferential SQL Injection

: 결과값을 얻을 수 없는 blind 환경에서 진행되는 공격

eg) Boolean Based SQL Injection, Time/Sleep Based SQL Injection

Union Based SQL Injection의 경우 앞의 결과와 뒤의 union으로 연결한 결과의 column의 수가 같지 않으면 오류가 발생합니다. 따라서 해당 column의 수를 미리 알 필요가 있는데 이때 하나씩 늘려가며 확인하는 방법도 있지만 order by를 사용할 수 있을 경우 해당 방법을 이용한 방법도 있습니다.

확인하는 방법의 경우 query 뒤에 order by [숫자]를 입력하였을 경우 오류가 발생하지 않으면 그 숫자 이상 혹은 이하에 해당하는 column이 사용된다는 것을 알 수 있습니다. 더 빠른 검색을 위해서는 binary search를 응용할 수 있을 것으로 보입니다.

MySQL의 경우 information\_schema DB의 schemata table에서 schema\_name을 조회하면 존재하는 DB의 이름들을 확인할 수 있습니다. schema\_name을 조건으로 하여 table\_name을 조회하면 해당 DB에 존재하는 table 이름들을 확인할 수 있습니다.

서로 다른 결과를 한 줄로 출력할 때는 group\_concat(항목) 함수를 통해서 한 줄로 표현할 수 있습니다.

Error Based SQL Injection의 경우 다양한 방법이 있는 것으로 보이지만 그 중 해당 강의에서 자주 사용하는 방법은 round(rand(0))과 having min(0)의 조합입니다. 오류가 발생하는 쿼리는 다음과 같습니다.

Group by round(rand(0)) having min(0)

이때 오류의 경우 Duplicate entry ‘1’ for key ‘group\_key’라고 출력되는데 ‘1’ 부분을 공격자가 원하는 내용을 출력되도록 변경할 수 있습니다. 만약 version 정보를 출력하고 싶은 경우 group by에서 concat(version(), round(rand(0)))과 같이 사용할 경우 ‘version 정보1’과 같이 출력됩니다.

BurpSuite를 사용할 경우 semi-auto SQL Injection이 가능합니다.