1-64. Secure Coding ( ) ?

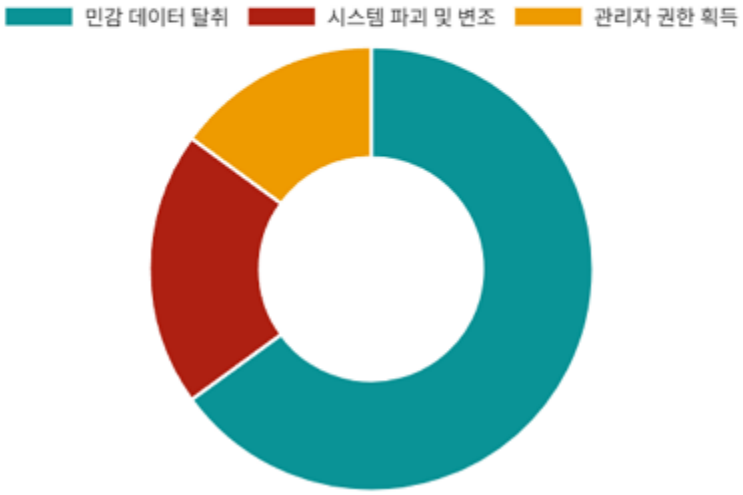
1. Secure Coding ( ) ?
    a.

2. SQL Injection ?
    a.

, .

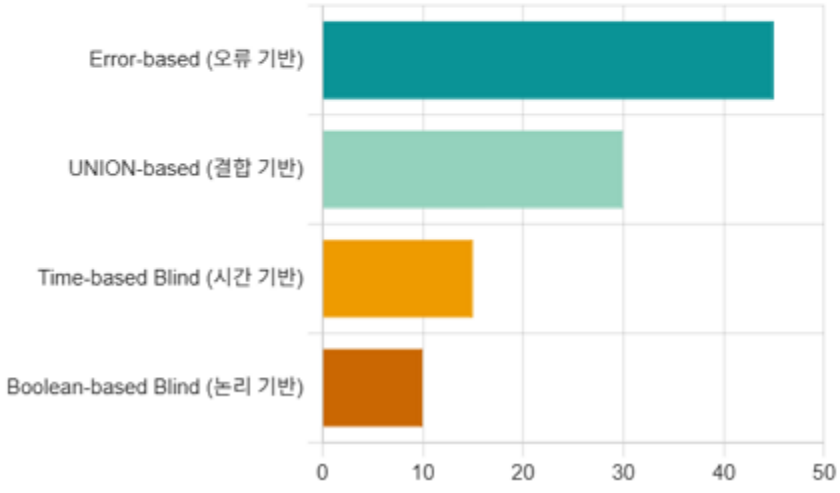
SQLI 공격의 주된 목표

SQL Injection 공격은 단순한 장난이 아닙니다. 공격자들은 명확한 목적을 가지고 시스템의 가장 민감한 부분인 데이터베이스를 노립니다. 주된 공격 목표는 다음과 같습니다.



가장 흔한 공격 벡터

공격자들은 다양한 기법을 사용하여 SQL Injection을 시도합니다. 그 중에서도 데이터베이스의 오류 메시지를 활용하거나, 여러 쿼리 결과를 합치는 방식이 가장 빈번하게 사용됩니다.

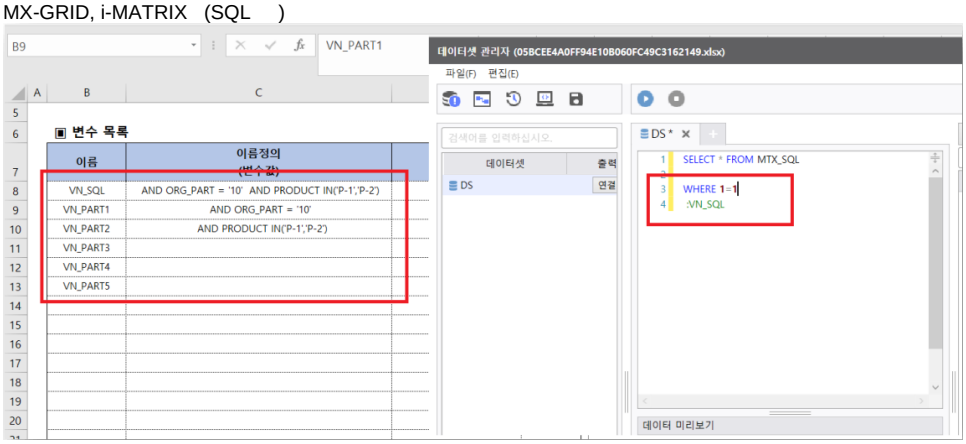


b. SQL Injection

i. SQL Injection SQL

c. SQL Injection

i. MX-GRID, i-MATRIX (SQL)

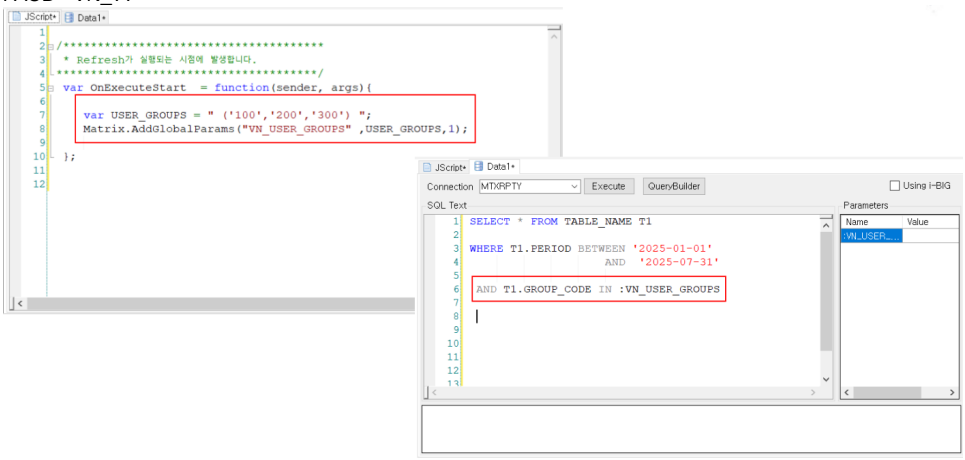


ii. MATRIX VBA SQL
SQL Injection & (vba .)
VBA SQL SQL .

(일반)

```
'-------------------------
' 조건 만들기
'-------------------------
Sub make_filter_sql()

    Dim sql As String
    Dim vsCode As String
    Dim vnFilter As String

    ' 이름정의에서 값 읽기
    vsCode = Range("VS_CODE").Text   ' 예: 사용자 입력값
    vnFilter = Range("VN_FILTER").Text   ' 예: 필터 조건

    sql = "WHERE 1=1"

    ' VS_CODE 조건
    If vsCode <> "" Then
        sql = sql & " AND T1.VS_CODE = '" & vsCode & "'"
    End If

    ' VN_FILTER 조건
    If vnFilter <> "" Then
        sql = sql & " AND T1.FILTER_COLUMN = '" & vnFilter & "'"
    End If

    ' 최종 SQL을 FILTER_SQL 셀에 출력
    Range("FILTER_SQL").Value = sql

End Sub
```

iii. i-AUD VN_??



iv. (.)

3. SQL Injection
   a. VN_
      i. VS_ SQL ‘’
      ii. VN_ .
   b. **Dynamic SQL**
      i. ,
      ii. .
   c. .
      i. Client(PC) .
      ii. .
   d. **SSL**
      i.

      : https://audp.bimatrix.co.kr/pages/viewpage.action?pageId=78708798
   e. AUD
      i.

      https://audp.bimatrix.co.kr/pages/viewpage.action?pageId=108396697

()	i-AUD Server Script / Dynamic SQL	SQL (i-META)	i-AUD JScript	()	
IP_ADDR	session.getAttribute ("IP_ADDR") <%=IP_ADDR$%>	:VS_IP_ADDR$	Matrix. GetUserInfo(). IPAddress	192.168.xxx. xxx	
USER_CODE	session.getAttribute ("USER_CODE") <%=USER_CODE$%>	: VS_USER_CO DE$	Matrix. GetUserInfo(). UserCode	matrix	
USER_NAME	session.getAttribute ("USER_NAME") <%=USER_NAME$%>	: VS_USER_NA ME$	Matrix. GetUserInfo(). UserName		
ORG_CODE	session.getAttribute ("ORG_CODE")	: VS_ORG_COD	Matrix. GetUserInfo().	B060000	

	<%=ORG_CODE$%>	E$	DeptCode		
ORG_NAME	session.getAttribute ("ORG_NAME") <%=ORG_NAME$%>	: VS_ORG_NAM E$	-		
DeptCode	session.getAttribute ("DeptCode") <%=DeptCode$%>	:VS_DeptCode$	Matrix. GetUserInfo(). DeptPath	'B060000', DEFAULT,-1	
USER_DEP T_PATH	session.getAttribute ("USER_DEPT_PATH") <% =USER_DEPT_PATH$%>	: VS_USER_DEP T_PATH$	-	|B060000|DE FAULT|-1	
USER_ROLE	session.getAttribute ("USER_ROLE") <%=USER_ROLE$%>	: VS_USER_ROL E$	Matrix. GetUserInfo(). UserRole	SU;3;V0;SV; ST;M0;1;	
LANG_CODE	session.getAttribute ("LANG_CODE") <%=DeptCode$%>	: VS_LANG_COD E$	Matrix. GetUserInfo(). LangCode	ko	(Admin PORTAL)
LANG_IDX	session.getAttribute ("LANG_IDX") <%=DeptCode$%>	: VS_LANG_IDX$	-	1	(MTX_LANG > LANG_NAME_COLUMN 'COMMENTS')

4. SQL Injection (Dynamic SQL)
    a. AUD Platform Dynamic SQL
    https://audp.bimatrix.co.kr/display/rndmanual/%5Bi-AUD%5D+AUD+Platform+Dynamic+SQL

b. AUD Platform Dynamic
https://audp.bimatrix.co.kr/pages/viewpage.action?pageId=92078261

```
<%
var req = Matrix.getRequest();
var util = Matrix.getUtility();
var VS_KEYWORD = req.getParam("VS_KEYWORD");
var VS_TIME_CODE = req.getParam("VS_TIME_CODE");
var sqls = [];
if(VS_KEYWORD){
    VS_KEYWORD = util.Replace(VS_KEYWORD, "'", "''");

    sqls.push(" AND ( T1.국가별 LIKE '%"+VS_KEYWORD+"%' ");
    sqls.push("     OR  T1.시점 LIKE '%"+VS_KEYWORD+"%' ");
    sqls.push(" )");
}
if(VS_TIME_CODE == "Y-1"){
    sqls.push(" AND T2.시점 = '2025' ");
}else if(VS_TIME_CODE == "Y-2"){
    sqls.push(" AND T2.시점 = '2024' ");
}
%>
-------------------------------------------
-- 국가별 연구 현황 조회
-------------------------------------------
SELECT   T1.시점 -- 시점(년도별_현황)
    , T1.국가별 -- 국가별(년도별_현황)
    , T2.시점 -- 시점(연구주제별)
    , T2.국가별 -- 국가별(연구주제별)
    , SUM(T1.연구원수) -- 연구원수
    , SUM(T2.정부) -- 정부
FROM mex_rnd1 T1 -- 1.년도별_국가별_연구자_수
LEFT OUTER JOIN mex_rnd3 T2 -- 연구주체별_연구개발비
    ON (
        T1.국가별 = T2.국가별
        AND T1.시점 = T2.시점
        )
WHERE 1 = 1

   <%=sqls.join("\n")%>

GROUP BY T1.시점,
    T1.국가별,
    T2.시점,
    T2.국가별
```