

MUNIQSOFT

APEX Security

Marco Patzwahl

DOAG Regio 2018

Muniqsoft GmbH

◆ Tätigkeitsbereiche:

- ▶ Oracle IT-Consulting & Services
- ▶ Oracle Remote Support und Rufbereitschaft
- ▶ Oracle Schulungen (SQL, PL/SQL, DBA, APEX, ... - gerne auch Inhouse)
- ▶ Software-Lösungen
- ▶ Oracle Lizenzen

Dauerwerbesendung

Muniqsoft GmbH
Schulungszentrum
Grünwalder Weg 13a
82008 Unterhaching
Tel.: 089 / 679090 40

MUNIQSOFT

Muniqsoft GmbH
IT-Consulting & Support
Witneystr. 1
82008 Unterhaching
Tel.: 089 / 6228 6789 0

Marco Patzwahl

- ◆ **Studium der Elektrotechnik an der TUM**
- ◆ **Ab 1993 5 Jahre bei Oracle gearbeitet, meine erste Oracle Version war 7.0.16 (1993)**
- ◆ **htp.p Package seit 1993 im Einsatz ☺, APEX seit Version 1.6**
- ◆ **Seit 1998 Geschäftsführer der Firma Muniqsoft GmbH**
- ◆ **Schwerpunkt: Schulungen (ca. 45 Stück, davon 4 Apex Schulungen ...)**



Die sichersten Server ...

- ◆ Ziehen das Stromkabel des Servers mit einem Ruck nach oben.
- ◆ Alternativ Hauptsicherung:AUS



99% Sicherheit

Sicherheitsrelevante Fehlkonfiguration

Verfügbarkeit der Applikation

- ◆ Shared Components → Definition
- ◆ Regelung der Applikations-Verfügbarkeit im Bereich Availability

Availability	
Status	Available with Developer Toolbar <input type="button" value="⌵"/> <input type="button" value="⌶"/> <input type="button" value="ⓘ"/>
Build Status	Run and Build Application <input type="button" value="⌵"/> <input type="button" value="⌶"/> <input type="button" value="ⓘ"/>
Message for unavailable application	<input type="text" value="This application is currently unavailable at this time."/>
Restrict to comma separated user list (status must equal Restricted Access)	<input type="text"/>

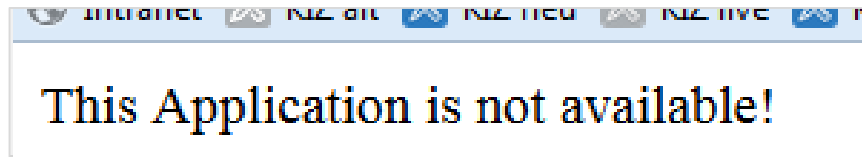
Verfügbarkeit der Applikation

◆ ab APEX 5.1: APEX_UTIL.SET_APPLICATION_STATUS

- ▶ Festlegen des Applikations-Status über SQL*PLUS

- ▶ Beispiel:

```
begin
  apex_util.set_workspace('APEX');
  apex_util.set_application_status(
    p_application_id      => 101,
    p_application_status => 'UNAVAILABLE',
    p_unavailable_value  => 'This Application
                           is not available!');
end;
/
commit;
```



Die größten Hackerangriffe der letzten Jahre

Firma	Wann	Betroffen
Ashley Madison	2015	37 Millionen Accounts
Home Depot Hack	2014	50 Mio. Kreditkartenaccounts
Ebay	2014	145 Mio. Benutzeraccounts
JPMorgan Chase	2014	ca. 80. Mio Kundendaten
LinkedIn	2016	164 Mio Benutzeraccounts
Anthem Health Care	2015	78 Mio. Kundendaten
UBER	2017	57 Mio. Kunden und Fahrerdaten
MyFitnessPal	2018	150 Mio. Benutzeraccounts
Swisscom	2018	0.8 Mio Kundendaten
T-Mobile (US)	2017	Ca. 70 Mio. Kundendaten
Paypal Tochter	2017	1.6 Mio Kundendaten

Hinweis: Facebook fehlt, weil die Daten nicht geklaut, sondern verschenkt wurden!

OWASP

- ◆ **Open Web Application Security Project**
 - ▶ **gemeinnützige, unabhängige Organisation**
 - ▶ **Ziel: Unterstützung von Unternehmen bei Entwicklung, Kauf und Wartung von sicheren Anwendungen**
 - ▶ **Angebot:**
 - **Werkzeuge und Standards für Anwendungssicherheit**
 - **Sicherheitsforschung**
 - **Publikationen, Sicherheitsmaßnahmen und –bibliotheken**
 - ▶ **Top 10: die zehn kritischsten Sicherheitsrisiken für Organisationen**

SQL Injection

Neulich bei einem Radar-Foto ...



SQL Injection

- ◆ SQL Injection ist der Versuch zusätzliche Kommandos/Filter in ein SQL Befehl einzubauen
- ◆ bei Oracle nicht möglich: mit Semikolon weiteres Kommando anzuhängen! Der DELETE hier wird nicht ausgeführt
 - ▶ `SELECT sal FROM emp`
`WHERE empno=7839; delete from emp;`
- ◆ Angriffsmöglichkeiten der Hacker
 - ▶ Hinzufügen von Filtern: OR 1=1
 - ▶ Einbauen von SQL Funktionen, die als autonome Transaktionen in der DB Änderungen durchführt

SQL Injection - Beispiele

◆ Report Query:

```
SELECT * FROM scott.emp WHERE ename='&P1_ENAME.';
```

- ▶ Filter A' or 'a'='a → zeigt alle Zeilen der Tabelle an

◆ UNION Klausel an SQL-Befehl hängen:

```
▶ SELECT * FROM emp UNION SELECT ... FROM all_tables;
```

◆ Unterabfrage anhängen

◆ Aufruf von gefährlichen Packages/Funktionen

◆ WHERE Klausel auskommentieren: **--WHERE**

SQL Injection - Beispiele

◆ Anzeige der installierten Benutzer:

- ▶ `select * from dual
where 1 = sys.dbms_metadata.openw(null, (select
listagg(username, ':') within group (order by username)
from all_users))` Textlänge: 113!
- ▶ `ORA-31600: Ungültiger Eingabewert
ANONYMOUS:APEX_050100:APEX_PUBLIC_USER:APPQOSSYS:AUDSYS:BI
:CTXSYS:DBSFUSER:DBSNMP:DIP:DVF:DVSYS:FLows_FILES:GGSYS:G
SMADMIN_INTERNAL:GSMCATUSER:GSMUSER:HR:IX:LBACSYS:MDDATA:M
DSYS:OE:OJVMSYS:OLAPSYS:ORACLE_OCM:ORDDATA:ORDPLUGINS:ORDS
YS:OUTLN:PM:REMOTE_SCHEDULER_AGENT:SCOTT:SH:SI_INFORMTN_SC
HEMA:SPATIAL_CSW_ADMIN_USR:SPATIAL_WFS_ADMIN_USR:SYS:SYS$U
MF:SYSBACKUP:SYSDG:SYSKM:SYSRAC:SYSTEM:WMSYS:XDB:XS$NULL`
- ▶ `ORA-06512: in "SYS.DBMS_SYS_ERROR", Zeile 105`

SQL Injection - Schutzmaßnahmen

◆ Bindvariablen statt Austauschvariablen!

- ▶ `SELECT * FROM emp WHERE ename=:P1_ENAME;`
- ▶ Bindvariablen nur in WHERE-Klausel verwenden (nicht bei Spalten-/Tabellennamen, Gruppenfunktionen, ORDER BY)

◆ Aktivierung der Session State Protection

◆ Einsatz von VPD (Virtual Private Database)

- ▶ Leider nur in EE verfügbar

SQL Injection - Schutzmaßnahmen

- ◆ werden Bindvariablen z.B. für Tabellennamen genutzt, verwenden Sie folgende Syntax:

```
DECLARE
s CLOB;
BEGIN
s := 'select ENAME from EMP
      where ename= '|| nvl( chr(39) || replace(
                        :P17_SUCHE, chr(39), chr(39)||chr(39) )
      || chr(39), 0);
http.p(s);
RETURN s;
END;
```

Ersetzt z.B. O'Reilly durch 'O"Reilly'

Items gegenüber Sonderzeichen schützen

- ◆ **Ab Version 5.1 kann bei Textitems im Bereich Security /Restricted Characters ein zusätzlicher Schutz aktiviert werden:**

All characters can be saved.

All characters can be saved.

Whitelist for a-Z, 0-9 and space

Blacklist HTML command characters (<>"')

Blacklist &<>"'/;,*|=%% and --

Blacklist &<>"'/;,*|=%% or -- and new line

Items gegenüber Sonderzeichen schützen (f)

◆ Prüfung, welche Items diesen Schutz besitzen:

```
◆ SELECT f.flow_step_id as page,  
f.name,f.csize,f.cmaxlength,  
f.restricted_characters,f.protection_level,  
f.escape_on_http_output,f.encrypt_session_state_yn  
FROM apex_180100.wwv_flow_step_items f  
WHERE f.flow_id=102 --AND f.flow_step_id=8  
ORDER BY 1,2;
```

PAGE	NAME	CSIZE	CMALENGTH	RESTRICTED_CHARACTERS	PROTECTION_LEVEL	ESCAPE_ON_HTTP_OUTPUT	ENCRY
0	P0_FEHLER	(null)	(null)	(null)	N	Y	N
0	P0_KURS_ID	(null)	(null)	(null)	N	Y	N
0	P0_KURSTEXT	(null)	(null)	(null)	N	Y	N
0	P0_SUCHE	(null)	(null)	(null)	N	(null)	(null)
5	P5_ANREDE_1	(null)	(null)	WEB_SAFE	N	(null)	N
5	P5_ANREDE_10	(null)	(null)	WEB_SAFE	N	(null)	N

gefährliche Module für SQL Injection



- ▶ **APEX Variablen (&MY_VAR.)**
- ▶ **EXECUTE IMMEDIATE**
- ▶ **dynamische Cursor**
 - REF Cursor
 - DBMS_SQL



- ▶ **Bindvariablen (:MY_VAR)**
- ▶ **PL/SQL Cursor**
 - CURSOR LOOP
 - OPEN FETCH CLOSE
- ▶ **in anderen Programmiersprachen (C, Java):**
 - Feldlängen begrenzen
 - Filter einbauen
 - Werte maskieren

DBMS_ASSERT

- ◆ Benutzereingaben mittels DBMS_ASSERT prüfen und ggf. Abbruch mit Fehlermeldung

- ◆ DBMS_ASSERT.SIMPLE_SQL_NAME
 - ▶ Prüfung auf nutzbare SQL-Objektnamen
 - ▶ Beispiele:
 - `SELECT DBMS_ASSERT.simple_sql_name('007Bond') FROM dual;` ✗
 - `SELECT DBMS_ASSERT.simple_sql_name('Bond007#') FROM dual;` ✓
 - `SELECT DBMS_ASSERT.simple_sql_name('"$Bond007"') FROM dual;` ✓

DBMS_ASSERT

◆ DBMS_ASSERT.ENQUOTE_LITERAL

- ▶ Prüfung, ob einfache Hochkommata falsch verwendet werden (schliessendes Hochkomma fehlt)

- ▶ Beispiele:

- ```
SELECT dbms_assert.enquote_literal(
 q' #O' 'Neill#'
FROM dual;
```



- ```
SELECT dbms_assert.enquote_literal(  
  q' #KING' or 1=1#'  
FROM dual;
```



DBMS_ASSERT

◆ Beispiel ✘

```
▶ CREATE OR REPLACE PROCEDURE doit(  
    p_table in varchar2,  
    p_valuecol in varchar2,  
    p_wherecol in varchar2,  
    p_param in varchar2 )  
  
IS  
  
    v_value varchar2(32767);  
    v_sql varchar2(32767);  
  
BEGIN  
v_sql := 'select '||p_valuecol||' from '||p_table||  
' where '||p_wherecol||' = '''||p_param||''';  
EXECUTE IMMEDIATE v_sql INTO v_value;  
END;
```

DBMS_ASSERT

◆ Beispiel ✓

```
▶ CREATE OR REPLACE PROCEDURE doit(
    p_table IN VARCHAR2,
    p_valuecol IN VARCHAR2,
    p_wherecol IN VARCHAR2,
    p_param IN VARCHAR2 )
IS
    v_value VARCHAR2(32767);
    v_sql VARCHAR2(32767);
BEGIN
    v_sql := 'select
    ||dbms_assert.enquote_name(p_valuecol)||' '||' 'from
    ||dbms_assert.enquote_name(p_table)||' '||' 'where
    ||dbms_assert.enquote_name(p_wherecol)||' =
    ||dbms_assert.enquote_literal(p_param);
EXECUTE IMMEDIATE v_sql INTO v_value;
END;
```

Reguläre Ausdrücke

- ◆ alle Zeichen, die keine Buchstaben, Zahlen oder Leerzeichen sind, werden durch ein Leerzeichen ersetzt:

- ▶

```
SELECT regexp_replace(
'Hallo 'x' ', '^[:alnum:][:blank:]', ' ')
FROM dual;
```

→ Hallo x

- ▶

```
SELECT regexp_replace(
'Hallo "x" ', '^[:alnum:][:blank:]', ' ')
FROM dual;
```

→ Hallo x

- ◆ weitere Sonderzeichen müssen nach Bedarf freigegeben werden, wie z.B. - ; : .

Fehler in Authentifizierung und Session Management

Benutzer und Passwörter

- ◆ für jeden Workspace einen Administrator mit einem Namen (nicht ADMIN) und einem langen Passwort anlegen
- ◆ strenge Passwortvorgaben einhalten und Passwörter regelmäßig ändern lassen
- ◆ Accounts sperren, wenn mehrmals ein falsches Passwort verwendet wurde

Neuer Admin für Internal Workspace

- ◆ neuen Benutzer für den Internal Workspace anlegen
- ◆ alternativ: Skript **apxchpwd.sql**
 - ▶ **ADMIN Name**
 - ▶ **ADMIN Email**
 - ▶ **ADMIN Passwort**
- ◆ Name sollte nicht ADMIN beinhalten (Besser: KING_MARCO)
- ◆ Passwort sollte schwer zu knacken sein (z. B. owgdNwdiwmPh2018)
- ◆ owgdNw,diwmPh=Oh wie gut dass Niemand weiß, dass ich wie mein Passwort heiß

Überwachung von Logins

◆ Überprüfen Sie regelmäßig die fehlerhaften Logins

▶ alternativ mit dem SELECT:

```
SELECT * FROM APEX_180100.APEX_WORKSPACE_ACCESS_LOG  
WHERE authentication_result<>'AUTH_SUCCESS';
```

→ ggf. Email schicken

Monitor Activity > Login Attempts

Q Go Actions

Login Name	Workspace	Application	Owner	Authentication Result	Authentication Method	Access Date
ADMIN	INTERNAL	4500	APEX_050000	Normal, successful authentication	Internal Authentication	12/05/2016 04:02:40 PM
ADMIN	APEX	102	SCOTT	Incorrect Password	Application Express Authentication	12/05/2016 04:02:28 PM
ADMIN	APEX	102	SCOTT	Incorrect Password	Application Express Authentication	12/05/2016 04:02:20 PM
ADMIN	APEX	4500	APEX_050000	Normal, successful authentication	Internal Authentication	12/05/2016 04:02:09 PM
ADMIN	INTERNAL	4050	APEX_050000	Normal, successful authentication	Internal Authentication	12/05/2016 04:01:34 PM

Password Policy

◆ INTERNAL WS → Manage Instance → Security → Password Policy

Password Policy

Manage password policy for Application Express users (workspace administrators, ...)

Password Hash Function: Most Secure

Minimum Password Length: 0

Minimum Password Differences: 0

Must Contain At Least One Alphabetic Character: No

Must Contain At Least One Numeric Character: No

Must Contain At Least One Punctuation Character: No

Must Not Contain: oracle:hello:welcome:guest:user:database

Alphabetic Characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Punctuation Characters: !"#\$\$%&()``*+,-/;<=>?_

Session Timeout

- ◆ Festlegung der Session Lebensdauer (z.B. max. 12 Stunden)
- ◆ unter Shared Components → Security Attributes → Session Management
 - ▶ Tipp: erstellen Sie eine öffentliche Seite mit der Info, dass die Session abgelaufen ist

Maximum Session Length in Seconds	<input type="text" value="7200"/>	^	?
	When Maximum Session Length is not set, the worksp		
Session Timeout URL	<input type="text" value="#LOGOUT_URL#"/>		
Maximum Session Idle Time in Seconds	<input type="text" value="3600"/>	^	?

Session Timeout

◆ Festlegung der Session Lebensdauer per APEX_UTIL (ab 12c):

```
▶ BEGIN  
  APEX_UTIL.SET_SESSION_LIFETIME_SECONDS (  
    p_seconds => 7200) ;  
END;
```

```
▶ BEGIN  
  APEX_UTIL.SET_SESSION_MAX_IDLE_SECONDS (  
    p_seconds => 1200) ;  
END;
```

▶ **Beispiel: Application Prozess (After Authentication)**

Cross Site Scripting

Cross Site Scripting

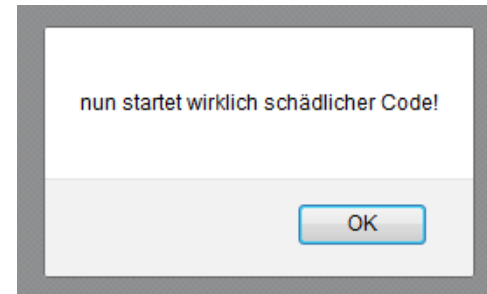
◆ **Problem:** Scriptcode wird im Browser des Benutzers ausgeführt

- ▶ Übernahme der Session
- ▶ Änderung von Seiteninhalten
- ▶ Umleitung auf böartige Seiten
- ▶ Trend 2017/2018: COINHIVE

COINHIVE

500 Millionen Website-Nutzer für Kryptomining missbraucht

Immer mehr Webseiten nutzen heimliches [Kryptomining](#). Neben dem Plugin von Coinhive gibt es jetzt weitere Anbieter. Bekannte Webseiten aus den Alexa Top 100.000 sollen gemeinsam auf eine Reichweite von einer halben Milliarde [Nutzer](#) kommen.



Cross Site Scripting

◆ XSS Beispiel:

- ▶ Tabellen Kommentare anlegen (mit HTML Tags zur Formatierung, siehe Notizen)
- ▶ Report erstellen:
`select * from kommentare`
- ▶ Anpassung der Kommentar-Spalten:
 - Escape Special Characters: NO
 - **Sicherheitslücke!!**

APEX_ESCAPE (ab 5.0)

- ◆ **Lösung:** APEX_ESCAPE Package (Groß und Kleinschreibung wird unterschieden!)

- ▶ Report Query:

- ```
select
 datum,
 username,
 apex_escape.html_whitelist(
 kommentar,
 '<h1>,</h1>,<h2>,</h2>,<h3>,</h3>,<h4>,</h4>,<p>,</p>,,,,,<i>,</i>,,,,,,,
,<hr/>,,'
) as kommentar
from kommentare
```

# APEX\_ESCAPE (ab 5.0)

## ◆ APEX\_ESCAPE.HTML:

- ▶ Escaping mithilfe eines PL/SQL Aufrufs
- ▶ Analog zur Standard-Einstellung in APEX  
→ alles wird maskiert!

## ◆ APEX\_ESCAPE.HTML\_WHITELIST

- ▶ lässt die in der Whitelist enthaltenen HTML-Tags intakt
- ▶ kann auch selbst definiert werden
- ▶ standardmäßig in der Whitelist:  
<h1>, </h1>, <h2>, </h2>, <h3>, </h3>, <h4>, </h4>, <p>, </p>, <b>, </b>, <strong>, </strong>, <i>, </i>, <ul>, </ul>, <ol>, </ol>, <li>, </li>, <br />, <hr/>

# APEX\_ESCAPE

- ◆ `SELECT apex_escape.html ('<B>Hello</B>') FROM dual;`
  - ▶ `&lt;B&gt;Hello&lt;/B&gt;`
- ◆ `SELECT apex_escape.html_whitelist ('<B>Hello</B>', '<B>,</B>') FROM dual;`
  - ▶ `<B>Hello</B>`
- ◆ `SELECT apex_escape.json (q'!O'Brien!')`  
`FROM dual;`
  - ▶ `O\u0027Brien`

# APEX\_ESCAPE

- ◆ `SELECT apex_escape.js_literal('<script>alert(1);</script>') FROM dual;`
  - ▶ `\u003Cscript\u003Ealert(1);\u003C\u002Fscript\u003E`
- ◆ `SELECT apex_escape.ldap_dn('Hans+Dieter') FROM dual;`
  - ▶ `Hans\+Dieter`
- ◆ `APEX_ESCAPE.SET_HTML_ESCAPING_MODE ( p_mode IN VARCHAR2);`
  - ▶ `p_mode = E(xtended) oder B(asic)`

# Substitution Syntax (ab 5.0)

## ◆ Vermeidung von Cross Site Scripting durch folgende Item-Referenzierung:

- ▶ `&P3_TEXT!JS.` → escape JavaScript  
(`apex_escape.js_literal`)
- ▶ `&P3_TEXT!HTML.`  
(`apex_escape.html`) → escape HTML
- ▶ `&P3_TEXT!ATTR.` → escape HTML Attribute  
(`apex_escape.html_attribute`)
- ▶ `&P3_TEXT!RAW.` → **VORSICHT:** kein Escape!



# Prüfung der Textlänge

- ◆ **Problem:** Hacker könnten die Länge von Textfeldern im HTML-Code beliebig erhöhen
- ◆ **Lösung #1:** Prüfung durch Validation, ob die Textlänge den internen maximalen Wert überschreitet (und manipuliert ist)
  - ▶ Validation (Typ: No Rows returned)
    - ```
SELECT 1 FROM apex_180100.wwv_flow_step_items
WHERE flow_id=:APP_ID
AND name='P3_NAME'
AND length(:P3_NAME)>CMAXLENGTH
```

Prüfung der Textlänge

◆ Lösung #2: Applikations-Prozess

```
▶ DECLARE
  stmt VARCHAR2(32000);
BEGIN
  FOR c IN (SELECT PAGE_ID, PAGE_NAME, ITEM_NAME,
    ITEM_ELEMENT_WIDTH, ITEM_ELEMENT_MAX_LENGTH
  FROM APEX_050100.APEX_APPLICATION_PAGE_ITEMS
  WHERE application_id=:APP_ID AND page_id=:APP_PAGE_ID
  AND ITEM_ELEMENT_MAX_LENGTH IS NOT NULL)
  LOOP
    stmt:='BEGIN IF length(v('' || c.item_name || '')) >
  ' || c.ITEM_ELEMENT_MAX_LENGTH || q'! THEN
  RAISE_APPLICATION_ERROR(-20500, 'Hacker Angriff
  entdeckt!!!'); END IF; END; !';
    EXECUTE IMMEDIATE stmt;
  END LOOP;
END;
```

Fehlerhafte Zugriffskontrolle

Letzten Monat ...

Hackers once stole a casino's high-roller database through a thermometer in the lobby fish tank



Hidden Items

◆ Arten von Hidden Items:

▶ Hidden:

- wird auf Seite nicht angezeigt, kann aber Werte aufnehmen

▶ **Hidden & Protected:**

- Prüfung bei Submit, ob der Inhalt, der beim Seitenaufbau existierte, nachträglich geändert wurde

◆ Alternative: Application Items

▶ liegen in der Applikation

- ▶ werden auf keiner Seite angezeigt (Auslesen der Werte nicht möglich)

Gefälschte Formularwerte

- ◆ Manipulation von Itemwerten direkt in der Webseite (HTML-Code) möglich
- ◆ auch Prüfsummen verhindern dies nicht!
 - ▶ Fehlermeldung kommt bei der Weiterverarbeitung
- ◆ Lösungsansatz #1:
 - ▶ After Submit Prozess
 - ▶ Prüfung, ob Werte des Items zugelassen sind, wenn nicht:
 - `apex_authentication.logout(p_app_id=>:APP_ID,p_session_id=>0);`

Gefälschte Formularwerte

◆ Lösungsansatz #2: Select List mit Hash Funktion

```
◆ SELECT distinct job,  
job||': '||dbms_crypto.hash(utl_raw.cast_to_raw(job),4) r  
FROM emp
```

◆ Validation:

▶ Function Returning Boolean:

```
IF  
dbms_crypto.hash(utl_raw.cast_to_raw(substr(:P29_JOB_SEC,1,instr(  
:P29_JOB_SEC,':')-1)),4)=  
substr(:P29_JOB_SEC,instr(:P29_JOB_SEC,':')+1) then  
RETURN true;  
ELSE  
RETURN false;  
END IF;
```

```
<select id="P29_JOB_SEC" class="selectlist" onchange="apex.submit('P29  
B_SEC');" size="1" name="p_t01">  
<option value=""></option>  
<option value="MANAGER:3A2449232D248F19B8B5CC136B0AA712">MANAGER <  
</option>
```

Gefälschte Formularwerte

- ◆ **VORSICHT!**
Validation und Prozesse (mit RAISE_APPLICATION_ERROR) brechen die weitere Verarbeitung der Seite ab, laden sie aber mit dem gefälschten Wert erneut!!
- ◆ Verbesserung bringt 5.1 mit der Seiteneinstellung Reload on Submit: Only for success
- ◆ Lösungsansatz #3:
 - ▶ Werte bereits in der Report Query ausschließen:

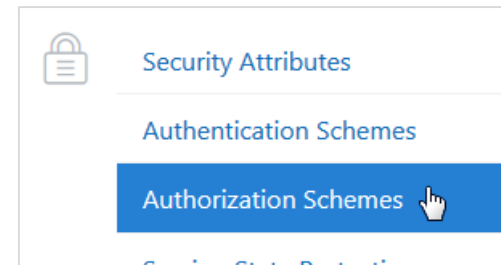
```
select ename, job, sal, deptno
from emp where deptno= :P8_DEPTNO
AND deptno IN (20,30)
```



Authorization Scheme

- ◆ **Definition von Zugriffsberechtigungen für Seiten und Komponenten**
 - ▶ u. a. Regionen, Reportspalten, Formulare, Navigationsmenü, Items
- ◆ **erstellen Sie ein Authorization Scheme für die jeweiligen Objekte**
- ◆ **Beispiel: nur der Benutzer "Admin" soll besondere Rechte bekommen**

Authorization Scheme

- ◆ unter Shared Components → Authorization Schemes
- ◆ neue Prüfung anlegen



Authorization Schemes		
Subscription by Component Utilization History		
Name 	Type	Caching
CHECK_USER	Exists SQL Query	Once per page view
LDAP_Group_Test	Exists SQL Query	Once per page view

Authorization Scheme

◆ Schema Type:
"Exists SQL Query"

◆ SQL Query:

```
SELECT 1 FROM dual
WHERE 'ADMIN'
= :APP_USER
```

The screenshot shows the configuration for an Authorization Scheme in Oracle APEX. The fields are as follows:

- Name:** CHECK_USER
- Scheme Type:** Exists SQL Query
- SQL Query:** select 1 from dual where :APP_USER = 'IBINADMIN'
- Identify error message displayed when scheme violated:** Seite für &APP_USER. gesperrt!
- Validate authorization scheme:** Once per session (selected), Once per page view, Once per component, Always (No Caching)

Authorization Scheme

◆ Beispiel #1: eigene Tabelle mit Benutzer und Berechtigungen

```
▶ CREATE TABLE scott.user_a_u_t_h (  
  username VARCHAR2(30),  
  r_name VARCHAR2(30));
```

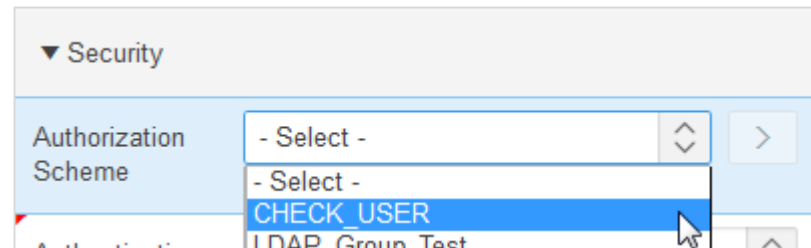
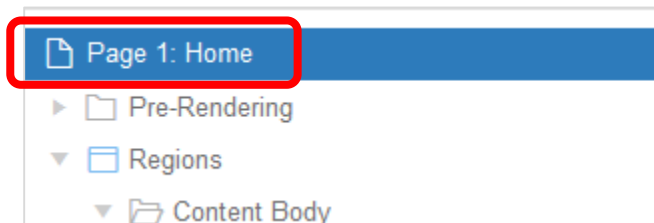
```
INSERT INTO scott.user_a_u_t_h VALUES  
( 'MARCO' , 'ADMIN' );
```

▶ Schema Type: Exists SQL Query

```
SELECT 1 FROM scott.user_a_u_t_h  
WHERE username=:APP_USER  
AND r_name='ADMIN';
```

Authorization Scheme

- ◆ verknüpfen Sie das Authorization Scheme mit der gewünschten Komponente
 - ▶ Eigenschaften → Security → Authorization Scheme



Parameter von apex_instance_admin

Parameter	Erklärung
ACCOUNT_LIFETIME_DAYS=<n>	Lebensdauer eines Account-Passworts
ALLOW_HOSTNAMES=<name>	Liste von Hostnames auf die navigiert werden darf
ALLOW_PUBLIC_FILE_UPLOAD=[Y N]	Dürfen Dateien ohne Benutzerauthentifizierung hochgeladen werden ?
ALLOW_REST=[Y N]	Y= Erlaubt Reports Region als Restservices zur Verfügung zu stellen
APPLICATION_ACTIVITY_LOGGING	[A]lways, [N]ever, [U]se application settings
APPLICATION_ID_MAX	Höchste nutzbar APP ID
APPLICATION_ID_MIN	Niedrigst nutzbare APP ID

Beispiel: `EXEC apex_instance_admin.set_parameter('ALLOW_REST'='Y');`
Die Daten stehen dann in `wwv_flow_plattform_prefs`

Weitere Parameter von apex_instance_admin

Parameter	Erklärung
AUTOEXTEND_TABLESPACES=[Y N]	Sollen Tablespace bis zur maximalen Größe erweitert werden?
CHECK_FOR_UPDATES=[Y N]	Prüfung auf akt. Updates
DELETE_UPLOADED_FILES_AFTER_DAYS=<n>	Zeitspanne nach der Uploaded Files automatisch gelöscht werden (Def.:14)
DISABLE_ADMIN_LOGIN=[Y N]	Y=Verhindert das Anmelden eines Administrators am Internal WS
DISABLE_WORKSPACE_LOGIN=[Y N]	Y=Verhindert das Anmelden am Workspace
EXPIRE_FIND_USER_ACCOUNTS=[Y N]	Y= Ablauf von Benutzeraccounts ist aktiviert
MAX_SESSION_IDLE_SEC=<N>	Maximale Idle-Zeit in Sekunden
MAX_SESSION_LENGTH_SEC	Maximale Lebensdauer einer Session in Sekunden

Weitere Parameter von apex_instance_admin

Parameter	Erklärung
PASSWORD_ALPHA_CHARACTERS	Möglich Zeichen für Password: Default: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ MNOPQRSTUVWXYZ
PASSWORD_HISTORY_DAYS	Expire Zeit für ein Passwort (Default 45 Tage)
PASSWORD_PUNCTUATION_CHARACTERS	!"#\$%&()``*+,-/!;<=>?_
REQUIRE_HTTPS	Development und Administration nur via https
REQUIRE_OUT_HTTPS	A = All = Komplette Instanz I= Dev und Admin N= https nicht zwingend notwendig
STRONG_SITE_ADMIN_PASSWORD =[Y N]	Y = Komplexe Password Regeln müssen befolgt werden, N = Kein Regeln
TRACING_ENABLED=[Y N]	Ermöglicht das Tracen von Seiten (Erzeugt u.U. viele Dateien auf dem Server

Verlust der Vertraulichkeit sensibler Daten

Verschlüsselung von Items

◆ Verschlüsselungsfunktion in der DB anlegen

- ▶ in allen DB-Versionen ohne Zusatzoptionen durch Package `dbms_crypto` verfügbar



Verschlüsselung von Items

◆ Beispiel einer einfachen Crypt-Funktion:

```
◆ CREATE OR REPLACE FUNCTION crypt (  
  text          IN VARCHAR2,  
  key           IN VARCHAR2 DEFAULT 'MuniQSoft_Key',  
  cryptmode     IN VARCHAR2 DEFAULT 'E') RETURN VARCHAR2 IS  
  p_key_raw    RAW(2048) := utl_raw.cast_to_raw(key);  
  
◆ BEGIN  
  IF substr(upper(cryptmode),1,1)='E' THEN  
    RETURN (sys.dbms_crypto.encrypt(  
      src => sys.utl_raw.cast_to_raw(text),  
      typ => sys.dbms_crypto.des_cbc_pkcs5,  
      key => p_key_raw));  
  ELSE  
    RETURN (utl_raw.cast_to_varchar2(sys.dbms_crypto.decrypt(  
      src => text,  
      typ => sys.dbms_crypto.des_cbc_pkcs5,  
      key => p_key_raw)));  
  END IF;  
END;  
/
```

APEX Security Tipps

◆ Debugging ausschalten

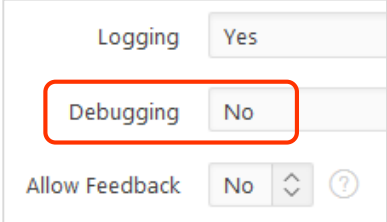
- ▶ Shared Components → Definition → Properties
- ▶ Prüfung, ob Debugging aktiviert ist:

```
SELECT name,owner, decode( f.APPLICATION_TAB_SET,
1,'Allowed',0,'Not Allowed','Allowed') debugging
FROM wwv_flows f
WHERE security_group_id<>10
```

APEX in einem eigenen Tablespace installieren!

◆ bei fertiger Applikation: Editiermöglichkeit abschalten

- ▶ Shared Components → Definition → Availability → Build Status
→ Run Application Only



The screenshot shows a portion of the APEX Properties dialog box. It contains three rows of settings:

Logging	Yes
Debugging	No
Allow Feedback	No

The 'Debugging' row is highlighted with a red rectangular border. To the right of the 'Allow Feedback' row, there are two small icons: a downward-pointing arrow and a question mark.

APEX Runtime Installation

- ◆ **in reiner Produktions- oder Testumgebung: nur Runtime Installation verwenden**
- ◆ **diese kann nachträglich installiert werden:**
 - ▶ `@?/apex/apxdevrm.sql`

Muniqsoft GmbH

◆ Tätigkeitsbereiche:

- ▶ Oracle IT-Consulting & Services
- ▶ Oracle Remote Support und Rufbereitschaft
- ▶ Oracle Schulungen (SQL, PL/SQL, DBA, APEX, ... - gerne auch Inhouse)
- ▶ Software-Lösungen
- ▶ Oracle Lizenzen

Muniqsoft GmbH
Schulungszentrum
Grünwalder Weg 13a
82008 Unterhaching
Tel.: 089 / 679090 40

MUNIQSOFT

Muniqsoft GmbH
IT-Consulting & Support
Witneystr. 1
82008 Unterhaching
Tel.: 089 / 6228 6789 0