- Possibilità dell'uso del version-control e backup
- Maggiore protezione del codice sorgente, difficile da sovrascrivere
- Nei packages del DB:
- Maggior efficenza del codice
- Accesso al codice tramite la tabella USER_SOURCE
- Integrazione con alcuni IDE

7.3.4.2 Tipologie di procedure vulnerabili

L'utilizzo di differenti strumenti di manipolazione dei dati che il PL/SQL mette a disposizione degli sviluppatori, determina la modalità con cui il codice viene scritto, ed in ultima istanza determina la tipologia di risorsa che il codice andrà a comporre. Esistono in PL/SQL i seguenti tipi di "risorse":

- embedded SQL
- cursori (ovvero i recordset del PL/SQL)
- EXECUTE IMMEDIATE (ovvero PL/SQL dinamico)
- Packages
- Triggers

Per tutte queste differenti tipologie di risorse, comunque, la casistica in cui il PL/SQL risulta vulnerabile può essere ridotta a due tipologie di codice:

- Blocco di PL/SQL anonimo, ovvero un blocco di codice racchiuso da BEGIN ed END, utilizzato per eseguire query multiple.
  Esempio:
  ```
  EXECUTE IMMEDIATE
  'BEGIN INSERT INTO TABELLA (COLONNA1) VALUES (''' || PARAM || ''');
  END;';
  ```
- Blocco di PL/SQLa singola riga, ovvero quel codice che non è dichiarato con BEGIN ed END, e non permette l'utilizzo del carattere ";" per l'iniezione di query multiple.
  Esempio:
  ```
  OPEN cur_cust FOR 'select name from customers where id = ''' || p_idtofind ||
  ''''';
  ```

7.3.4.3 Filtraggio dei tipi di input iniettabile

Quando si utilizzano le stored procedures, è necessario porre opportuna attenzione al filtro dei seguenti tipi di input:

- UNIONI: possono essere utilizzate per includere query ulteriori rispetto a quelle effettuate dalla stored procedure.
- SUBSELECTS
- Comandi DDL/DML (INSERT, UPDATE, DELETE etc.)
- Nomi dei packages

7.3.4.4 Filtro dei caratteri potenzialmente dannosi

- È necessario che i caratteri " (ASCII 34), ' (ASCII 39), in tutte le loro possibili codifiche (hex, ascii, utf-8, etc.), siano filtrati e/o opportunamente sanitizzati mediante escaping.
- È inoltre necessario che i caratteri # (ASCII 35), -- (ASCII 4545), % (ASCII 37), ; (ASCII 59), in tutte le loro possibili codifiche (hex, ascii, utf-8, etc.) siano filtrati e/o opportunamente sanitizzati mediante escaping.

7.3.4.5 Direttive per Oracle

Si elencano di seguito le direttive di configurazione del database Oracle alle quali è necessario attenersi – nei limiti posti dalle esigenze applicative – per raggiungere un elevato livello di sicurezza delle applicazioni sviluppate con questa tecnologia. Si tratta di azioni che devono essere eseguite per garantire una certa sicurezza.

Account:

- cambiare la password all'utente SYS;
- disabilitare gli account di default del database;

Ruoli:

- revocare il ruolo RESOURCE dagli utenti;
- revocare il ruolo CONNECT da tutti gli utenti;

Permessi:

- revocare il permesso pubblico di esecuzione su utl_file (vedi par. "prevenire l'upload remoto di file") ;
- revocare il permesso pubblico di esecuzione su utl_http (vedi par. "prevenire la redirezione dell'output") ;
- revocare il permesso pubblico di esecuzione su utl_tcp ;
- revocare il permesso pubblico di esecuzione su utl_smtp ;
- controllare il permesso pubblico di esecuzione sui packages e le viste di cui gli utenti sys e dba sono proprietari;
- revocare il permesso pubblico su dbms_random;
- revocare il permesso pubblico su dbms_lob ;
- revocare ogni tipo di permesso su dbms_sql e dbms_sys_sql granted;
- utilizzare i permessi dell'utente chiamante per ogni tipo di procedura;
- controllare e opportunamente dispensare il permesso "BECOME USER";
- controllare e opportunamente dispensare il permesso "CREATE ANY DIRECTORY";
- controllare e opportunamente dispensare il permesso "CREATE JOB";
- controllare e opportunamente dispensare il permesso "CREATE LIBRARY" ;
- revocare ogni permesso di esecuzione su sys.initjvmaux;
- revocare il permesso pubblico di esecuzione su dbms_job;
- revocare il permesso pubblico di esecuzione su dbms_scheduler ;
- revocare il permesso pubblico di esecuzione su owa_util;
- negare l'accesso all'esecuzione di "SELECT ANY TABLE";
- controllare ed opportunamente disporre i permessi di accesso al package dbms_backup_restore;
- revocare il permesso di creazione degli oggetti a tutti gli utenti eccetto quelli proprietari dello schema;
- controllare l'accesso agli oggetti ed assicurarsi che gli utenti possano interagire unicamente con gli oggetti che sono loro necessari;
- impedire al dba di leggere le tabelle di sistema;
- impedire al dba di leggere i dati dell'applicazione.

Inoltre:

- controllare ed opportunamente sanitizzare il parametro utl_file_dir;
- controllare l'accesso di Java al sistema operativo;
- controllare e regolare opportunamente la maniera in cui Java e Oracle interagiscono;
- rendere extproc sicuro;
- settare il parametro _trace_files_public a FALSE ;
- controllare e rendere sicuro il package statspack.

Altre misure per proteggere il codice PL/SQL consistono nei seguenti punti:

- Offuscamento del codice con WRAP: l'utility "wrap" (utilizzabile nella forma: wrap iname=input_file [oname=output_file]), deve essere utilizzato per offuscare i files SQL ove le procedure sono memorizzate. È necessario ricordare che l'utility wrap è in grado di offuscare il codice rendendo di difficile lettura il sorgente (e quindi l'algoritmo), ma non è in grado di proteggere eventuali stringhe di testo memorizzate staticamente nel codice, come nomi di tabelle e passwords.
- Prevenire la redirezione dell'output; è sempre necessario filtrare l'accesso al package UTL_HTTP che può essere utilizzato per la redirezione dell'output nelle query.