- 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: