Validare tutti gli input, indipendentemente dalla loro provenienza. La validazione dovrebbe essere basata su una white list: dovrebbero essere accettati cioè solo i dati conformi a una struttura specificata, scartando quelli che non la rispettano. Occorre controllare il tipo del dato, la sua dimensione, l'intervallo di validità (range), il formato ed eventuali valori attesi (white list).

Invece di concatenare le stringhe si consiglia di:

- Utilizzare componenti di database sicuri come le stored procedures, le query parametrizzate e le associazioni degli oggetti (per comandi e parametri);
- Una soluzione consigliabile è l'adozione di una libreria ORM, come EntityFramework, Hibernate o iBatis.
- Occorre inoltre limitare l'accesso agli oggetti e alle funzionalità di database, in base al "Principle of Least Privilege" (non fornire agli utenti permessi più elevati di quelli strettamente necessari).

Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/89.html,
CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection').

Esempio:
Il codice seguente adotta una query parametrizzata, una difesa contro la SQL injection:

```
cursor = connection.cursor(prepared=True)
stringaSQLInserimento = """ INSERT INTO dipendenti
          (id, Nome, DataAssunzione, Importo_Annuo) VALUES (%s,%s,%s,%s)"""

tupla_inserimento_1 = (progressivo, input_name, datetime.datetime.now(),
input_salario)
cursor.execute(stringaSQLInserimento, tupla_inserimento_1)

connection.commit()
print("record inserito")
```

7.5.8 XPath Injection

Come riconoscerla

Gli attacchi XPath Injection sono possibili quando un sito Web utilizza le informazioni fornite dall'utente per costruire una query XPath per i dati XML. Inviando informazioni intenzionalmente malformate nel sito Web, un utente malintenzionato può scoprire come sono strutturati i dati XML o accedere a dati a cui normalmente non avrebbe accesso. Potrebbe persino essere in grado di elevare i suoi privilegi sul sito Web se i dati XML vengono utilizzati per l'autenticazione (come un file utente basato su XML).

Come difendersi

Evitare che la costruzione della query XPath dipenda dalle informazioni inserite dall'utente. Possibilmente mapparla con i parametri utente mantenendo la separazione tra dati e codice. Nel caso fosse necessario includere l'input dell'utente nella query, questo dovrà essere precedentemente validato.
Validare tutti gli input, indipendentemente dalla loro provenienza. La validazione dovrebbe essere basata su una white list (si dovrebbero accettare solo i dati che adattano a una struttura specificata, scartando quelli che non la rispettano). Bisogna controllare il tipo del dato, la sua dimensione, l'intervallo di validità (range), il formato ed eventuali valori attesi (white list).

Esempio:

Forma non corretta: l'applicazione utilizza una stringa inserita dall'utente per costruire una query XPath:

```
from sys import stdin
import XPath
print 'Insert item number: '
userInput = stdin.readline()
XPath.find('//item' + userInput, doc)
```