

- Validare tutti gli input, indipendentemente dalla loro provenienza. Per la validazione, si consiglia l'approccio white list (sono accettati solo i dati che adottano una struttura specificata nella white list, scartando quelli che non la rispettano). In generale, è necessario controllare il tipo del dato, la sua dimensione, l'intervallo di validità (range), il formato ed eventuali valori attesi (white list).
- Evitare di costruire dinamicamente stringhe di connessione. Se è necessario creare dinamicamente una stringa di connessione evitare di includere l'input dell'utente. In ogni caso, utilizzare utilità basate sulla piattaforma, come SqlConnectionStringBuilder di .NET, o almeno codificare l'input validato come il più idoneo per la piattaforma utilizzata.
- Le stringhe di connessione possono essere custodite nel file web.config. Si tratta di una scelta migliore rispetto a comporle a runtime con l'input dell'utente. Si separa così l'applicazione dai metadati. Il file di configurazione in questione deve essere messo in sicurezza attivando la modalità "protected configuration", che permette di memorizzare le stringhe di connessione in forma crittografata (encrypted).

Per ulteriori informazioni si veda: <http://cwe.mitre.org/data/definitions/99.html>,  
CWE-99: Improper Control of Resource Identifiers ('Resource Injection').

#### 7.7.5 LDAP Injection

##### Come riconoscerla

La LDAP Injection è un tipo di attacco cui sono vulnerabili le applicazioni e che utilizzano l'input, senza verificarlo adeguatamente, per costruire query LDAP (Lightweight Directory Access Protocol).

Se coronato da successo, l'LDAP injection potrebbe consentire un furto di informazioni, un'elevazione dei privilegi e l'autenticazione con un'identità altrui (spoofing).

Per comunicare con la directory delle utenze (ad esempio Active Directory), l'applicazione costruisce dinamicamente delle query. Se utilizza l'input utente senza verificarlo, un malintenzionato può inserire comandi modificati ad arte per carpire informazioni non dovute.

##### Come difendersi

Validare tutti gli input, indipendentemente dalla loro provenienza. Per la validazione, si consiglia l'approccio white list (sono accettati solo i dati che adottano una struttura specificata nella white list, 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).

Per ulteriori informazioni: <http://cwe.mitre.org/data/definitions/90.html>,  
CWE-90: Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection').

#### 7.7.6 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.