```
 char *nomeUtente = argv[2];

 // Codice passibile di SQL Injection
 char query[1000] = {0};
 sprintf(query, "SELECT USER_ID FROM UTENTI where nome = \"%s\"", nomeUtente);
 executeSql(query);

 // Codice "sanificato"
 char nomeUtenteSql[1000] = {0};
 encodeSqlString(nomeUtenteSql, 1000, nomeUtente);
 char querySanificata[1000] = {0};
 sprintf(querySanificata, "SELECT USER_ID FROM UTENTI where nome = \"%s\"",
nomeUtenteSql);
 executeSql(querySanificata);
}
```

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').

7.1.6 LDAP Injection

Come riconoscerla

Come in tutti i casi d'injection, anche in questo caso a essere sfruttato per l'attacco è l'input dell'utente, nel momento in cui viene utilizzato, senza subire alcun controllo o filtro, per comporre una query LDAP.
Il pericolo è che venga inquinata la directory LDAP, che contiene una base dati relativa a delle utenze.
Con un attacco LDAP injection è possibile leggere dati riservati, come è anche possibile modificarli, cancellarli o inserire utenze che poi possono essere utilizzate per successivi attacchi.

Esempio:
Il seguente codice riceve un parametro in input per comporre una query LDAP.

```
fgets(nomeUtente, sizeof(nomeUtente), socket);
snprintf(queryLDAP, sizeof(queryLDAP, "(cn=%s)", nomeUtente);
```

Se nomeUtente è "Mario Rossi", la query restituirà i dati relativi all'utente in questione, ma se viene fornito il carattere "*", verrà restituito l'intera directory di utenze.

Come difendersi

Occorre mettere in pratica le misure che seguono. Come in altri tipi d'injection, sono fondamentali il controllo e l'encoding dell'input, per costruire filtri e query verso server LDAP.
L'encoding deve filtrare i seguenti caratteri: `\ # + < > , ; " =`
Altri caratteri scpeciali sono utilizzati all'interno delle query LDAP e quindi non possono essere eliminati in automatico: * ( ) . & - _ [ ] ` ~ | @ $ % ^ ? : { } ! '
Il controllo applicativo, dipendente dal contesto, assume un'importanza fondamentale.
Anche ridurre al minimo i privilegi assegnati all'utenza con la quale il server LDAP è avviato è una misura utile a minimizzare le conseguenze di un attacco.

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

7.1.7 Process control

Come riconoscerla

Le vulnerabilità del controllo di processo si verificano quando nell'applicazione vengono importati dati provenienti da un'origine non attendibile. Tali dati vengono successivamente caricati utilizzando il metodo Load-Library. Controllando il nome o il percorso della libreria, un utente malintenzionato può sostituire la