```
        // Forward to handler
      }
    }
```

Nel seguente snippet viene effettuato un controllo che impedisce l'injection:

```
// ... beginning of LDAPInjection.searchRecord()...
sc.setSearchScope(SearchControls.SUBTREE_SCOPE);
String base = "dc=example,dc=com";
if (!userSN.matches("[\\w\\s]*") || !userPassword.matches("[\\w]*")) {
      throw new IllegalArgumentException("Invalid input");
}
String filter = "(&(sn = " + userSN + ")(userPassword=" + userPassword + "))";
// ... remainder of LDAPInjection.searchRecord()...
```

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').

7.2.6 Resource Injection

Come riconoscerla

Si verifica quando l'applicazione ha la necessità di far aprire un socket da parte dell'utente. Un malintenzionato potrebbe aprire una backdoor che permette di connettersi direttamente al server, facendo escalation dei privilegi fino a prendere il controllo della macchina. Tramite questa vulnerabilità il malintenzionato potrebbe utilizzare eventuali connessioni aperte dall'utente, nel caso non fossero gestite adeguatamente.

Come difendersi

Non si deve in alcun caso consentire a un utente di definire i parametri relativi ai sockets di rete.
Validare l'input raffrontandolo con una white list di valori possibili ammessi.

Esempio:
La situazione iniziale:

```
public class ResourceInjection {
      public static void main(String[] args) {
               Scanner userInputScanner = new Scanner(System.in);
               System.out.print("\nEnter port number: ");
               int portNumber = Integer.parseInt(userInputScanner.nextLine());
               try {
                        ServerSocket serverSocket = new ServerSocket(portNumber);
               } catch (Exception e) {
                        System.err.println("Caught Exception: " + e.getMessage());
               }
      }
}
```

Questa vulnerabilità viene risolta limitando le possibilità a poche scelte (white list):

```
public class ResourceInjectionFixed {
      public static void main(String[] args) {
               Scanner userInputScanner = new Scanner(System.in);
               System.out.print("\nEnter port name: ");
               String portName = userInputScanner.nextLine();
               int portNum;
               switch (portName) {
               case "ftps":
                        portNum = 989;
                        break;
               case "ftp":
                        portNum = 20;
```