Nel codice seguente, invece, l'injection non sarebbe possibile, poiché l'input è un numero e non una stringa:

```
public class CommandInjectionFixed {
    public static void main(String[] args) throws IOException {
            int num = Integer.parseInt(args[0]);
            // Controlli sul numero immesso
            Runtime runtime = Runtime.getRuntime();
            Process proc = runtime.exec("fileNumber" + Integer.toString(num)  + ".exe");
    }
}
```

Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/77.html,
CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection').

7.2.4 Connection string injection

Come riconoscerla

La stringa di connessione è un insieme di coppie chiave/valore separate da un punto e virgola. Consentono alle applicazioni Web di connettersi al database o ad altro server (per esempio Active Directory). Se un'applicazione Web crea una stringa di connessione utilizzando la concatenazione di stringhe dinamiche, per connettersi al database in base all'input fornito dagli utenti, tale applicazione Web è vulnerabile all'attacco di iniezione della stringa di connessione.
Come in tutti i casi di injection, anche qui parametri di input non verificati possono essere utilizzati per

Come difendersi

La validazione dell'input, avvalendosi di una white list, filtrando i caratteri pericolosi, è sempre la soluzione corretta per questo tipo di vulnerabilità. In questo caso i parametri non dovrebbero includere segni speciali come il punto e virgola, separatore delle varie coppie chiave/valore.

Esempio:
Il seguente codice:

```
public class ConnectionStringInjection {
       public static void main(String[] args) throws SQLException {
                Scanner userInputScanner = new Scanner(System.in);
                System.out.print("\nEnter url name: ");
                String connURL = userInputScanner.nextLine();
                Connection con = DriverManager.getConnection(connURL, "username",
"password");
       }
}
```

Andrebbe corretto come segue:

```
public class ConnectionStringInjectionFixed {
       public static void main(String[] args) throws SQLException {
       HashMap<String, String> sanitize = new HashMap<String, String>();
                sanitize.put("DB_url_1", "DB_url_1");
                sanitize.put("DB_url_2", "DB_url_2");
                sanitize.put("DB_url_3", "DB_url_3");
                Scanner userInputScanner = new Scanner(System.in);
                System.out.print("\nEnter url name: ");
                String connURL = userInputScanner.nextLine();
                Connection con = DriverManager.getConnection(sanitize.get(connURL),
"username", "password");
       }
}
```

Il valore è valido se è uno di quelli memorizzati nell'hashmap `sanitize`.

Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/99.html,