```
public class CodeInjectionFixed {
	static void main(String[] args){
		String fileName = null;
		switch(args[0]){
		case "First":
			fileName="First.txt";
			break;
		case "Second":
			fileName="Second.txt";
			break;
		case "Third":
			fileName="Third.txt";
			break;
		default :
			fileName="none.txt";
		}
		System.load(fileName);
	}
}
```

Si veda: http://cwe.mitre.org/data/definitions/94.html,
CWE-94: Improper Control of Generation of Code ('Code Injection').

7.2.3 Command injection

Come riconoscerla

Accade quando l'applicazione esegue comandi di sistema operativo sul server che la ospita. Un attaccante potrebbe utilizzare questa caratteristica per eseguire comandi dannosi.
Si realizza nel momento in cui un'applicazione prevede un'istruzione che lancia comandi sul sistema operativo utilizzando un input non verificato. Comandi arbitrari potrebbero:

- Alterare i permessi su file e directory del file system, (read / create / modify / delete).
- Permettere delle connessioni di rete non autorizzate verso il server da parte dell'attaccante.
- Avviare e fermare servizi di sistema.
- Consentire all'attaccante il controllo completo del server da parte dell'attaccante.

Attraverso questa vulnerabilità l'applicazione viene indotta ad eseguire dei comandi voluti dall'utente malintenzionato. L'operazione spesso viene effettuata concatenando stringhe di input dell'utente a codice dannoso. Potrebbero così essere eseguiti direttamente sul server comandi anche molto pericolosi per il sistema o per la sicurezza dei dati.

Come difendersi

- Scrivere il codice in modo che non esegua nessuna shell dei comandi. Utilizzare a questo scopo le API messe a disposizione delle librerie Java;
- Se dovessero permanere shell dirette, fare in modo che siano stringhe statiche che non utilizzino l'input dell'utente;
- In ogni caso occorre validare l'input, filtrando i caratteri pericolosi, attraverso una struttura definita per l'input, o – meglio ancora – imponendo una white list di valori ammessi.

Esempio:
Caso in cui si potrebbe avere command injection:

```
public class CommandInjection {
    public static void main(String[] args) throws IOException {
        Runtime runtime = Runtime.getRuntime();
        Process proc = runtime.exec("fileNumber" + args[0] + ".exe");
    }
}
```