7.5.6 Resource Injection

Come riconoscerla

Un utente malintenzionato potrebbe aprire una backdoor per connettersi direttamente al server, aggirando tutti le procedure di autenticazione e autorizzazione.

Come difendersi

Non consentire a un utente di definire i parametri relativi ai sockets di rete.

Esempio:
Forma non corretta – L'applicazione apre una socket di rete utilizzando un nome host immesso dall'utente:

```
from sys import stdin
import socket
import sys
userInput = stdin.readline()
HOST = userInput
PORT = 8888 # Arbitrary non-privileged port

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'Socket created'
#Bind socket to local host and port
try:
  s.bind((HOST, PORT))
except socket.error as msg:
  print 'Bind failed. Error Code : ' + str(msg[0]) + ' Message ' + msg[1]
  sys.exit()
print 'Socket bind completè
```

Forma corretta - L'applicazione indica uno o piu' indirizzi host codificati in una white-list tra i quali l'utente può scegliere.

```
import socket
import sys
HOST = ''  # Symbolic name, meaning all available interfaces
PORT = 8888 # Arbitrary non-privileged port
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'Socket created'
#Bind socket to local host and port
try:
  s.bind((HOST, PORT))
except socket.error as msg:
  print 'Bind failed. Error Code : ' + str(msg[0]) + ' Message ' + msg[1]
  sys.exit()
print 'Socket bind completè
```

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

7.5.7 SQL Injection

Come riconoscerla

Se l'applicazione compone le query SQL per interrogare il database con l'input dell'utente, un malintenzionato potrebbe introdurre satringhe alterate ad arte per accedere indebitamente ai dati del sistema, rubare qualsiasi informazione riservata memorizzata (ad esempio i dati personali dell'utente o le carte di credito) ed eventualmente modificare o cancellare i dati esistenti.
L'applicazione comunica con il suo database inviando una query SQL in formato testo. Se l'applicazione crea la query semplicemente concatenando le stringhe provenienti dall'input dell'utente, non verificandone la validità, il pericolo che venga sferrato un attacco di SQL injection è molto concreto.

Come difendersi