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

L'applicazione utilizza una stringa inserita dall'utente per costruire una query XPath:

```
$user = $_GET["user"];
$pass = $_GET["pass"];

$doc = new DOMDocument();
$doc->load("test.xml");
$xpath = new DOMXPath($doc);

$expression = "/users/user[@name='" . $user . "' and @pass='" . $pass . "']";
$xpath->evaluate($expression); // Non sicuro
```

La stringa inserita dall'utente viene sottoposta a encoding prima dell'uso nella query XPath:

```
$user = $_GET["user"];
$pass = $_GET["pass"];

$doc = new DOMDocument();
$doc->load("test.xml");
$xpath = new DOMXPath($doc);

$user = str_replace("'", "&apos;", $user);
$pass = str_replace("'", "&apos;", $pass);

$expression = "/users/user[@name='" . $user . "' and @pass='" . $pass . "']";
$xpath->evaluate($expression);
```

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

7.9.11 XML External Entity (XXE) injection

Come riconoscerla

Si verifica quando un'applicazione fa il parsing e incorpora in automatico i riferimenti di entità DTD, all'interno di un documento XML. Se un attaccante predispone un documento XML manipolato, può essere in grado di leggere arbitrariamente qualsiasi file del server.
Potrebbe inserire, ad esempio, <! ENTITY xxe SYSTEM "file: /// c: /boot.ini">.
Dovrebbe poi aggiungere un riferimento che faccia riferimento alla definizione di tale entità, ad es. & xxe; . Se il documento XML analizzato viene quindi restituito all'utente, il risultato includerà il contenuto sensibile del file di sistema.
Ciò è causato dal parser XML, che è configurato per analizzare automaticamente le dichiarazioni DTD e risolvere i riferimenti alle entità, invece di disabilitare sia i riferimenti DTD che quelli esterni.
Esempio:

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

Come difendersi

La soluzione migliore, ovviamente, sarebbe quella di evitare di elaborare direttamente l'input dell'utente, ove possibile.