**Weak XML Schema: Lax Processing Development Mitigation SOP**

A schema that does not enforce strict input validation can allow arbitrary elements or attributes to be validated against it. This opens the door for an attacker to supply a malicious document to the system. This can happen with values for processContents of “skip” and “lax”.

**Defense Against Weak XML Schema: Lax Processing**

Ensure that processContents is set to strict. This will ensure that validation is performed on wildcards.

**Example**

Imagine you are using the following two schemas for validation.  
  
**Schema 1:**

<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >  
 <xs:element name="cart" >  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="itemID" maxOccurs="1" />  
 <xs:any namespace="http://itemInfo" processContents="lax" minOccurs="1" maxOccurs="5"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

**Schema 2:**

<?xml version="1.0"?>  
<xsd:schema targetNamespace="http://itemInfo" xmlns:xsd="http://www.w3.org/2001/XMLSchema" >  
 <xsd:element name="name" type="xsd:string"/>  
 <xsd:element name="description" type="xsd:string"/>  
</xsd:schema>

Schema 1 permits arbitrary tags in the http://itemInfo namespace to be validated against it, thus allowing the following XML to pass validation.

<?xml version=\"1.0\" ?>  
<cart xmlns:itemInfo="http://itemInfo">  
 <itemID>123</itemID>  
 <itemInfo:name>Shoes</itemInfo:name>  
 <itemInfo:description>Black Shoes</itemInfo:description>  
 <itemInfo:price>1.00</itemInfo:price>  
</cart>

The above XML contains an itemInfo:price tag which is not defined in Schema 2. As a result, it might be possible to trick the consumer of this XML document into setting a price of $1 for Shoes.

**Recommendation**

Replacing Schema 1 with the following will ensure that validation fails when it encounters the undefined itemInfo:price element.

<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >

<xs:element name="cart" >

<xs:complexType>

<xs:sequence>

<xs:element name="itemID" maxOccurs="1" />

<xs:any namespace="http://itemInfo" processContents="strict" minOccurs="1" maxOccurs="5"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**References**

1. [HP Enterprise Security - Weak XML Schema: Lax Processing Mitigation SOP](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/xml/weak_xml_schema_lax_processing.html)