python的lxml库的xxe防御

niexinming / 2018-07-13 15:13:27 / 浏览数 4232

我最近在审计内部产品的时候发现一处有趣的代码

```
import lxml.objectify
jioc={}
ioco = lxml.objectify.parse(filename)
root = ioco.getroot()
jioc['short_description'] = root.short_description.__str__()
```

这段代码有xxe漏洞，而且

```
from lxml import etree
tree=etree.parse(filename)
root= tree.getroot()
for i in root.getroottree().getiterator('modelVersion'):print i.text
```

还有

```
xml='''
   <!DOCTYPE ent [
   <!ENTITY ent SYSTEM "file:///etc/passwd">
   ]>
   <b>&ent;</b>
   '''
   a=objectify.fromstring(xml)
   print a
```

都会造成xxe漏洞，关于xxe的问题，在官方网站中有提到https://bugs.launchpad.net/lxml/+bug/1742885，而解决方法在https://mikeknoop.com/lxml-xxe-exploit/提

解决方案

只要把resolve_entities设置成False就可以了

```
from lxml import etree
parser = etree.XMLParser(resolve_entities=False)
```

但是这样修改之后，就会出现一些莫名奇妙的bug，于是我就按照建议把代码做了一些修改：

```
import lxml.objectify
from lxml import etree
filename="test.ioc"
ioco = lxml.objectify.parse(filename,etree.XMLParser(resolve_entities=False))
jioc = {'rule': '', 'member': {}, 'description': '', 'short_description': '','level':''}

root={}
for elt in ioco.getroot():
       root[etree.QName(elt.tag).localname]=elt.text

jioc['short_description'] = root['short_description']
print jioc
definition = root['definition']
```

这样就可以防止xxe漏洞

我的同事qinghua做了一些更不错的修改，他使用python自带的xml库，用底层语言实现功能，方便控制
他的代码：

```
import sys
import os
reload(sys)
sys.setdefaultencoding("utf-8")
from xml.parsers import expat

class Element(object):
   '''analyze a element'''
```

```
    def __init__(self, name, attributes):
        #record tag and attribute dictionary
        self.name = name
        self.attributes = attributes
        #clear the element cdata and its children
        self.cdata = ''
        self.children = [ ]

    def addChild(self, element):
        self.children.append(element)

    def getAttribute(self, key):
        return self.attributes.get(key)

    def getData(self):
        return self.cdata

    def getElements(self, name = ''):
        if name:
            return [ c for c in self.children if c.name == name ]
        else:
            return list(self.children)

class Xml2Obj(object):
    '''transform XML to Object'''
    def __init__(self):
        self.root = None
        self.nodeStack = [ ]
    def StartElement(self, name, attributes):
        'Expat start element event handler'
        #make instance of class
        element = Element(name.encode(), attributes)
        #put the element into stack and make it become child_element
        if self.nodeStack:
            parent = self.nodeStack[-1]
            parent.addChild(element)
        else:
            self.root = element
        self.nodeStack.append(element)

    def EndElement(self, name):
        'Expat end element event handler'
        self.nodeStack.pop()

    def CharacterData(self, data):
        '''Expat character data event handler'''
        if data.strip():
            data = data.encode()
            element = self.nodeStack[-1]
            element.cdata += data

    def Parse(self, filename):
        #create Expat analyzer
        Parser = expat.ParserCreate()
        #Set the Expat event handlers to our methods
        Parser.StartElementHandler = self.StartElement
        Parser.EndElementHandler = self.EndElement
        Parser.CharacterDataHandler = self.CharacterData
        #analyz XML file
        ParserStatus = Parser.Parse(open(filename).read(), 1)
        return self.root

if __name__ == '__main__':
    filename='test_xml.xml'
    parser = Xml2Obj()
    root_element = parser.Parse(filename)
    print root_element.getElements()[0].cdata
    ch=root_element.getElements('properties')[0].children
    print ch[0].cdata
```

```
ch=root_element.getElements('dependencies')[0].children
dependency_ch= ch[0].children
print dependency_ch[0].cdata
```

点击收藏 | 0 关注 | 1

上一篇：Gitea 1.4.0未授权远程代... 下一篇：PHP索引数组+unset使用不当...

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板