XXE漏洞（上）

原创

errorr0 已于 2022-04-10 17:22:44 修改 158 收藏
分类专栏： XXE 文章标签： xml web安全 安全
于 2022-04-09 23:33:54 首次发布

XXE漏洞的成因是源于**XML**语言，所以我们先简单学习**XML**是什么，如何分析，怎么用。

XML介绍与学习

XML 被设计用来传输和存储数据。是和**HTML**很像的一种标记语言。**XML**不是替代**HTML**的，而是**HTML**的扩展与补充。

XML 不会做任何事情。**XML** 被设计用来结构化、存储以及传输信息。大概意思就是说**XML**和其它编程语言不同的是它不会对数据做任何事，唯一的作用就是用来简化、规律化一串数据。如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>myfriends</to>
  <from>errorr0</from>
  <heading>Reminder</heading>
  <body>Don't forget me forever!</body>
</note>

输出：

Reminder

to: myfriends
from: errorr0

Don't forget me forever!
```

实际上并不是这样输出的，只是为了便于解释以及理解！！

XML的组成

XML总共由3个部分组成，分别是：

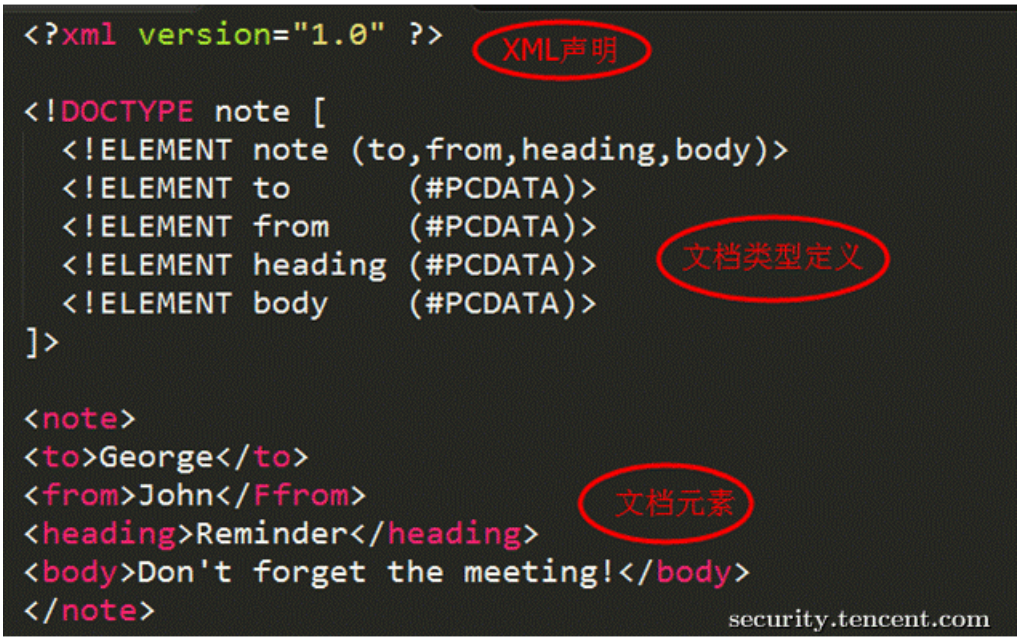
文档类型定义（Document Type Definition，简称**DTD**）即XML的布局语言；

可扩展的样式语言（Extensible Style Language，**XSL**）即XML的样式表语言；

可扩展链接语言（Extensible Link Language，**XLL**）。

XML格式

XML用于标记电子文件使其具有结构性的标记语言，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。**XML**文档结构包括**XML声明**、**DTD文档类型定义**、**文档元素**。




```
<?xml version="1.0" ?>

<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to       (#PCDATA)>
  <!ELEMENT from     (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body     (#PCDATA)>
]>

<note>
<to>George</to>
<from>John</Ffrom>
<heading>Reminder</heading>
<body>Don't forget the meeting!</body>
</note>
```

文档类型定义（DTD）是这几个之中最为重要的，也是**XXE**的关键，同样是我们要学习的重点！！DTD可以在**内部声明**也能在**外部声明**。一起来看看吧！

0x01 内部声明DTD

```
<!DOCTYPE 根元素 [元素声明]>
```

这其实是一个固定的**DTD**外部的格式，没有什么太大的作用，感觉根元素的命名和编程语言的变量命名规则差不多，规则为：不能以数字开头命名、不能为纯数字、不能有空格。

0x02 DTD实体

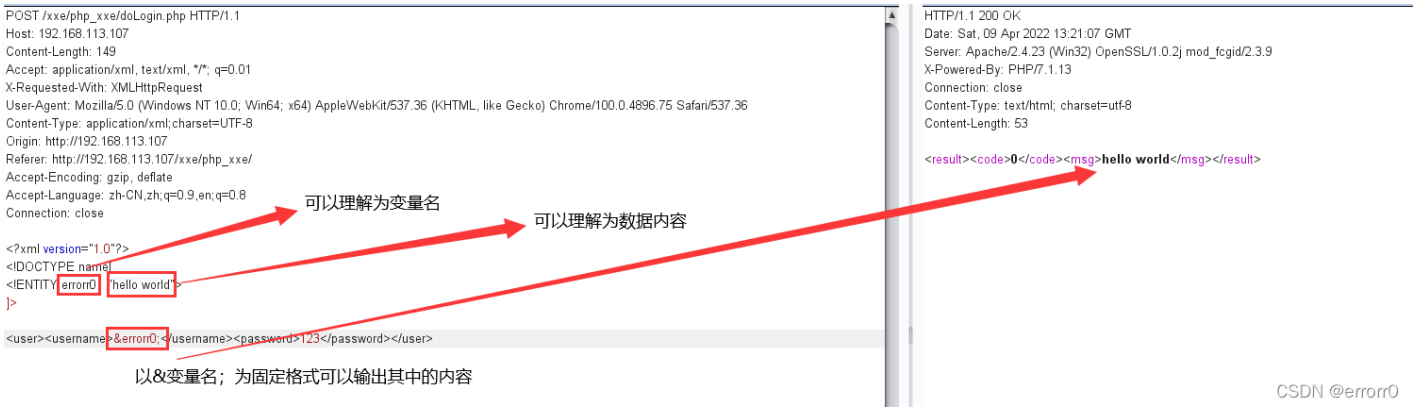
DTD实体也就是能起到作用的东西，也就是上述**DTD**内部声明中的元素声明的内容。

①内部声明实体

```
<!ENTITY 实体名称 "实体的值">
```

实体名称相当于变量名，实体的值相当于数据内容。

```
<?xml version="1.0"?>
<!DOCTYPE name[        //DTD内部声明
<!ENTITY errorr0   "hello world">    //DTD实体
]>

<user><username>&errorr0;</username><password>123</password></user>
```



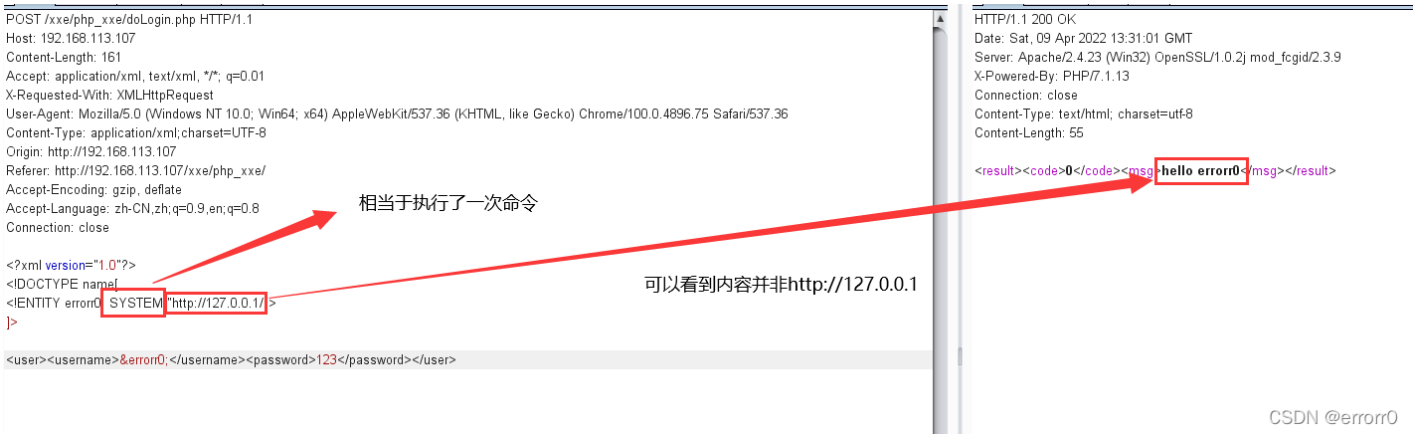
不用管这个题目是什么哈，后面讲完**XML**我们会拿出来再讲的！！

②引用外部实体

> <!ENTITY 实体名称 执行值 "外部的值">
>
> 这个执行值可以是：SYSTEM、PUBLIC

这个外部实体的执行值可以理解为一些函数的作用！！外部实体是通过一些协议调用到了外部的数据。

```
<?xml version="1.0"?>
<!DOCTYPE name[        //DTD内部声明
<!ENTITY errorr0  SYSTEM "http://127.0.0.1">    //DTD外部实体
]>

<user><username>&errorr0;</username><password>123</password></user>
```



③参数实体

无论是**外部声明实体**还是**内部实体**，我们都需要在文档元素中才能调用到这些”变量“，那么能不能在**DTD**中就直接调用呢？参数实体就是解决这一问题的。

> <!ENTITY % 实体名称 "实体的值">
>
> %实体名称 ； ----- 相当于调用

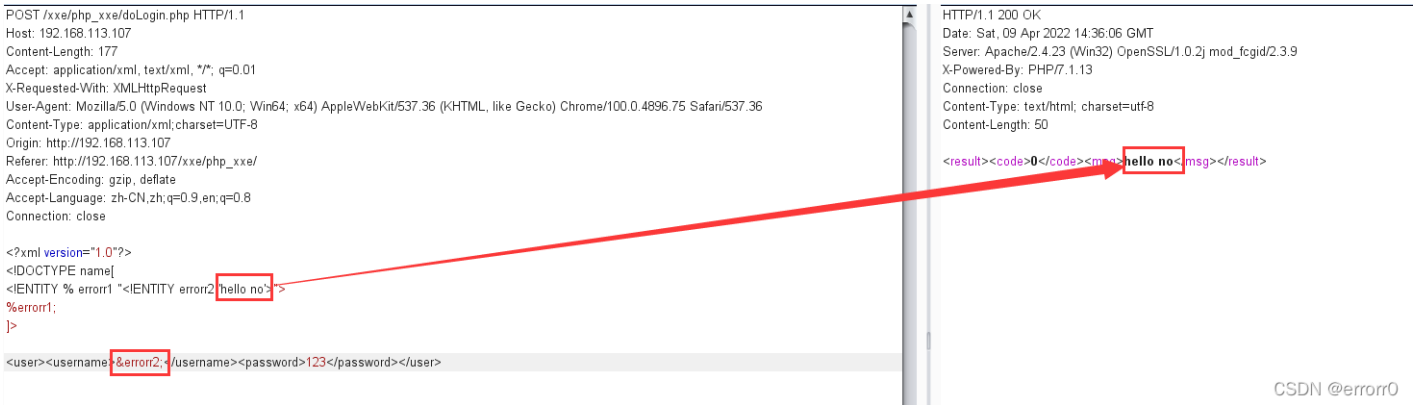
```
<!DOCTYPE name[
<!ENTITY % a   "hello">
<!ENTITY % b  "&a; errorr0">
%b;
]>


最后输出的答案为：hello errorr0
```

%b掉用"&a; errorr0"，&a;再调用一次%a 所以最后得到的结果为：hello errorr0。

再举个例子：

```
<?xml version="1.0"?>
<!DOCTYPE name[
<!ENTITY % errorr1 "<!ENTITY errorr2 'hello no'>">
%errorr1;
]>

<user><username>&errorr2;</username><password>123</password></user>
```

先内部参数实体调用，然后数据中又是一个实体，再在文档元素中调用&errorr2; 。



可以看到，我们成功了！！

④预定义的实体

但凡学过一点编程语言就应该知道被单引号包裹的数据中不能出现单引号，如果实在需要就得用上转义符号 \ ，那在**XML**中又怎么解决这个问题呢？这里就要讲到实体引用字符，这个其实和**HTML**大同小异。

<	<
>	>
&	&
"	" (双引号)

&apos;	' （单引号）

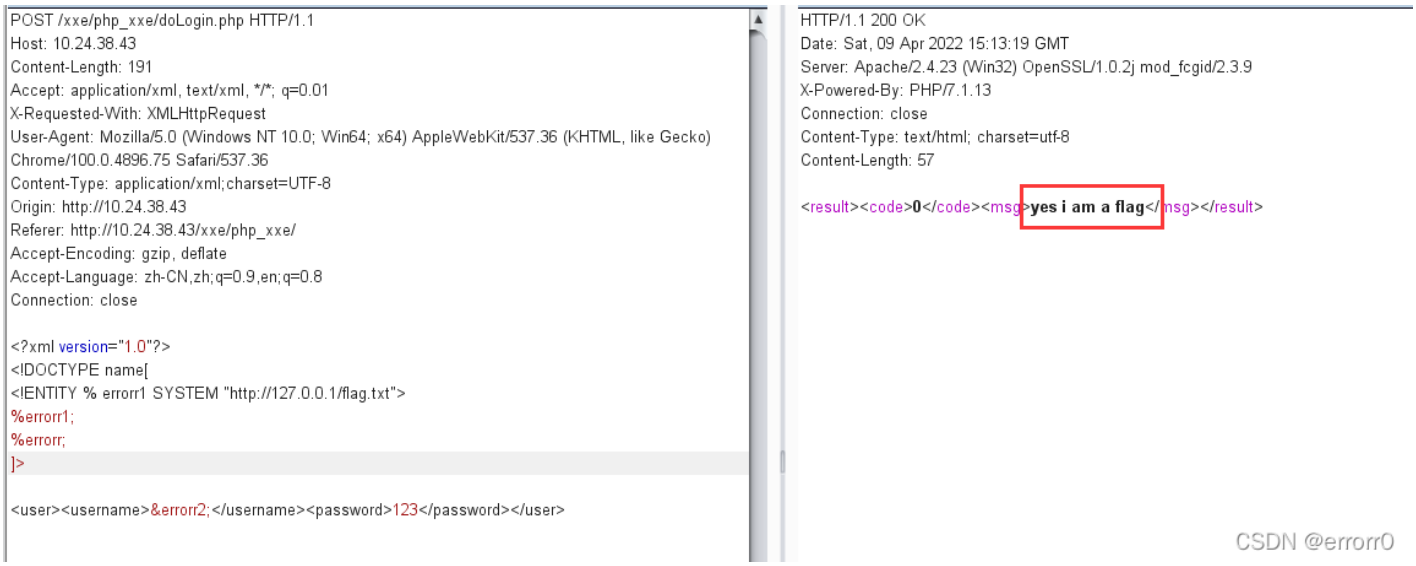
当然除了这些，还有一种方式代替一些符号比如百分号%，参数实体的百分号%也不能出现在实体值中，这个时候我们可以用**Unicode编码**，%=% 也可以写做16进制，%=%=%

⑤外部实体+参数实体的二重调用

这其实就是像套娃一样，一个套一个最后得到结果。

```
<?xml version="1.0"?>
<!DOCTYPE name[
<!ENTITY % errorr1 SYSTEM "http://127.0.0.1/flag.txt">
%errorr1;
%errorr;
]>

<user><username>&errorr2;</username><password>123</password></user>



flag.txt中内容如下：

<!ENTITY % errorr "<!ENTITY errorr2  'yes i am a flag'>">
```

其实不难，就是个套娃一级一级的来，先是**errorr1**访问后，执行http服务访问**flag.txt**，再者参数实体**errorr**访问**flag.txt**中的参数实体，然后套中套，有个**errorr2**，最后在外部实体中访问**errorr2**即可成功。



除此之外，其实还可以利用这种套娃的手段进行一些url的拼接，比如一段为IP，一段为?带的参数payload一系列的。这个还得依照碰到题目的情况而定。

总结

对**XML**感兴趣的可以去菜鸟教程看看，讲的挺详细的。因为这个只是为后面的**XXE漏洞**学习打基础，所以我并没有细讲，不过DTD的这些东西可以好好看看，挺有意思的。

参考：XXE—实体注入、XXE-lab-master(php_xxe WriteUp)_凌晨三点-的博客-CSDN博客_php_xxe

xxe-labs - A2rcher_zjh - 博客园