第5章 Web服务器攻防实训

攻击者在确定了攻击对象后，只有在实施攻击前全面掌握被攻击对象的详细配置信息，才能从中发现可利用的安全漏洞，进而确定具体的攻击方法，并实施渗透攻击。因此，Web服务器的安全防范十分重要。本章主要针对 Web 服务器的安全问题，通过对几个典型的攻防实验操作，使读者对 Web 服务器的安全性有一个直观的认识。

扫一扫

视频讲解

5.1 主机扫描：路由信息的收集

5.1.1 预备知识：路由信息

路由路径跟踪是实现网络拓扑探测的主要手段，Linux 操作系统中的 traceroute 和 Windows 环境中的 tracert 程序分别提供了不同平台上的路由路径跟踪功能，两者的实现原理相同，都是用 TTL(time to live，生存时间)字段和 ICMP 错误消息确定从一个主机到网络上其他主机的路由，进而确定 IP 数据包访问目标 IP 所采取的路径。在对目标网络中的不同主机进行相同的路由跟踪后，攻击者就可以综合这些路径信息，绘制出目标网络的拓扑结构，并确定关键设备在网络拓扑中的具体位置信息。

路由器(router)是一种网络通信设备，它工作在网络层，可以将应用层的报文划分成一个个分组后独立地发送到目的地(目的网络)，这个过程被称为路由。在网络拓扑组成中，路由器就是连接两个以上网络的互联设备。目前，路由器是互联网中连接不同局域网、广域网的网络互联设备。路由器根据不同的算法(路由协议)自动选择和设定路由，以最佳路径将原网络中的分组逐个发送到目的网络。路由器是互联网络的枢纽，可以将其理解为现代交通路网中的“交通警察”。

5.1.2 实验目的和条件

1. 实验目的

在掌握 TCP/IP 体系结构，特别是掌握网络层路由协议和路由器工作原理的基础上，通过学习 Linux 操作系统环境下相关路由跟踪工具的使用方法，掌握路由信息的探测和信息收集与分析方法。尤其是通过对几个工具(主要有 traceroute、dmitry、itrace、tcptraceroute、tctrace)应用功能的对比分析，可以发现不同工具的应用优势。

2. 实验条件

为使实验与实际应用有机结合，本实验使用一台运行 Kali Linux 2021 操作系统的计算机作为实验环境进行实验。

5.1.3 实验过程

步骤 1：正确登录 Kali Linux 系统，选择菜单栏上的 Terminal 选项，打开终端操作窗口。

步骤 2：traceroute(跟踪路由)是路由跟踪程序，用于确定 IP 数据包到目标主机所经过的路径。traceroute 命令可以用 IP 生存时间 (TTL) 字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由，其格式为“traceroute [参数] [主机]”。如果使用 traceroute

工具来追踪 www. baidu. com(36. 152. 44. 95 是百度的 IP 地址,也可以直接使用域名 www. baidu. com),追踪成功后将显示如图 5-1 所示的结果。由于 traceroute 默认使用 udp 方式,大部分应用服务器就不处理 udp 包,虽然图 5-1 中最后全是"*",实际上已经是到达目标主机了。

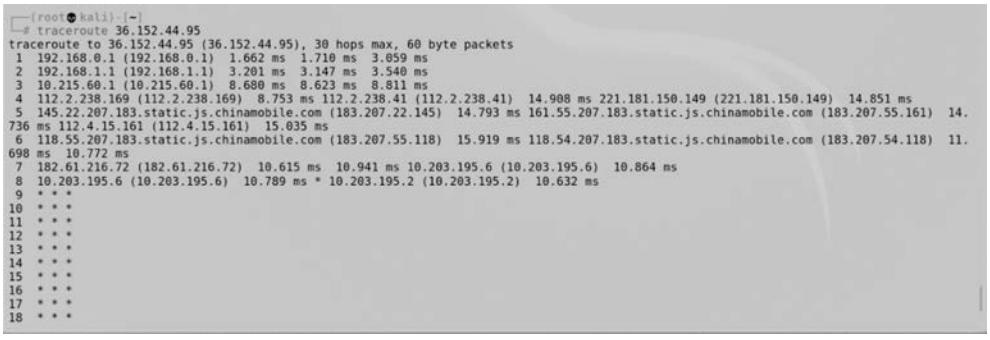


图 5-1 使用 traceroute 工具成功追踪 www. baidu. com 的显示结果

需要注意的是,如果 Kali Linux 安装在虚拟机环境下,采用 traceroute 进行外网路由跟踪时,应将网络连接设置为非 NAT 模式(如桥接模式),因为 NAT 是有网段隔离的,返回的 ICMP 包无法到达虚拟机。

步骤 3:DMitry 是黑客渗透流程中进行深度信息收集的利器。其常用功能如下。

(1) 进行 TCP 端口扫描,收集端口相关状态或其他信息。

(2) 从 Netcraft. com 获取主机信息、子域名、域名中包含的邮件地址。

(3) 收集 Whois 主机 IP 和域名等信息。

上述功能的具体使用方法可以通过 dmitry -h 命令查看,如图 5-2 所示。

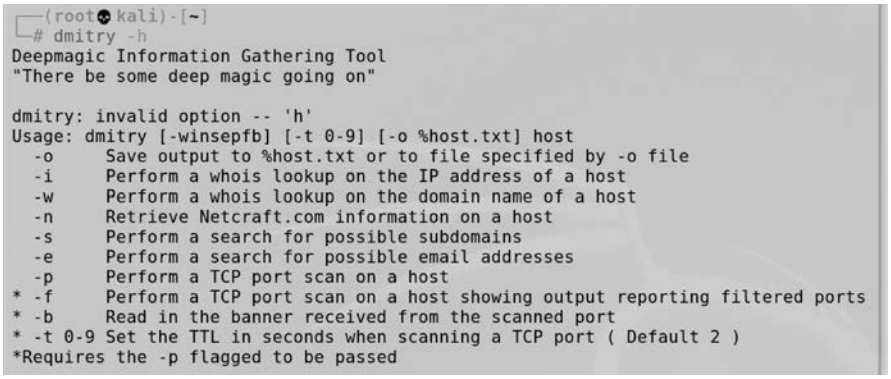


图 5-2 dmitry 支持的使用参数

输入 dmitry -wnp www. baidu. com 命令,扫描百度网站的 Whois 域名注册信息、对应服务器主机打开的端口信息,操作过程和显示结果分别如图 5-3、图 5-4 所示,读者会发现该主机开放了 SMTP 的 25 端口。

步骤 4:itrace 工具的应用。itrace 与 traceroute 使用-I 参数功能类似,使用 ICMP 反射请求跟踪路由。

```
┌──(root💀kali)-[~]
└─# dmitry -wnp www.baidu.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:180.101.49.11
HostName:www.baidu.com

Gathered Inic-whois information for baidu.com
---------------------------------
   Domain Name: BAIDU.COM
   Registry Domain ID: 11181110_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2020-12-09T04:04:41Z
   Creation Date: 1999-10-11T11:05:17Z
   Registry Expiry Date: 2026-10-11T11:05:17Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
eteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
```

图 5-3　百度网站的 Whois 域名注册信息

```
Gathered TCP Port information for 180.101.49.11
---------------------------------

 Port           State

25/tcp          open
80/tcp          open
110/tcp         open

Portscan Finished: Scanned 150 ports, 0 ports were in state closed


All scans completed, exiting
```

图 5-4　百度网站服务器打开的端口信息

执行 itrace -i eth0 -d www. baidu. com 命令，可以看到如图 5-5 所示的回复信息，说明已经进行了成功追踪。

```
┌──(root💀kali)-[~]
└─# itrace -i eth0 -d www.baidu.com
1(1)   [192.168.0.1]
2(1)   [192.168.1.1]
3(1)   [10.215.60.1]
4(2)   [221.181.146.25]
5(1)   [183.207.22.149]
6(1)   [183.207.55.118]
7(1)   [182.61.216.72]
8(1)   [10.203.195.6]
9(1)   [36.152.44.95] (reply)
```

图 5-5　使用 itrace 工具成功追踪到 www. baidu. com 后的显示信息

需要说明的是，百度 www. baidu. com 主机前面肯定是存在防火墙等安全设备的。这时，如果再使用 traceroute 工具追踪，将会得到如图 5-6 所示的结果，显示无法正常追踪，这充分说明了 itrace 工具的应用优势。

步骤 5：tcptraceroute 工具的应用。tcptraceroute 工具通过向目标主机发送 TCP SYN 数据包来追踪路由。与 traceroute 使用-T 参数功能类似。

执行 tcptraceroute www. baidu. com 命令，可追踪到百度服务器的路由路径。如图 5-7

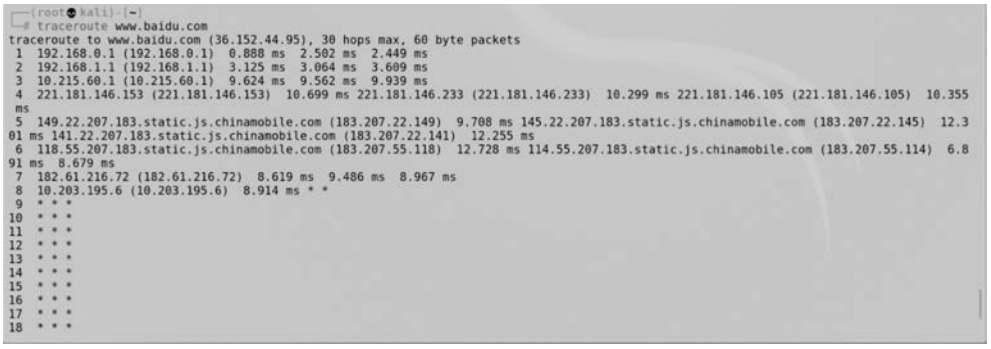


图 5-6 使用 traceroute 工具无法追踪到 www. baidu. com 主机的信息

所示，追踪到的最后一跳主机的 IP 地址为 36. 152. 44. 95，而该地址为百度服务器的地址，说明已经进行了成功追踪。

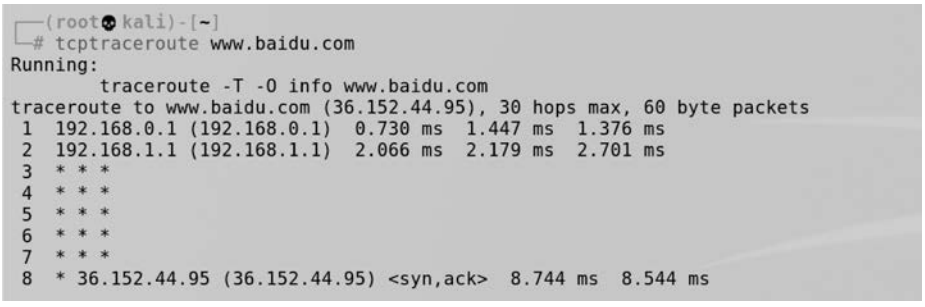


图 5-7 tcptraceroute 成功追踪 www. baidu. com 主机后的显示信息

步骤 6：与 tcptraceroute 相比，如果使用不带参数的 traceroute 进行追踪，则默认使用的是 udp 数据包，网络中一些路由器或防火墙可能会封掉 ICMP 返回的信息，该条信息会以“*”显示。同时大部分应用服务器不提供 udp 服务，因此最后没有收到主机不可达的信息，不停增大 TTL 值进行追踪，如图 5-8 所示。

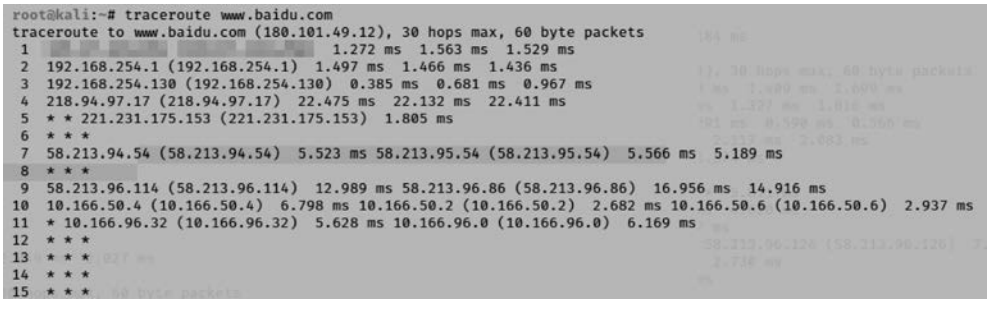


图 5-8 使用 traceroute 发送 upd 追踪 www. baidu. com 后显示的结果

traceroute 使用参数-I 发送 ICMP 请求，追踪结果如图 5-9 所示，网络中大部分路由不响应 ICMP 请求。

traceroute 使用参数-q，设定发送数据包的个数，如图 5-10 所示，会发现一些能够响应 ICMP 请求的路由器。

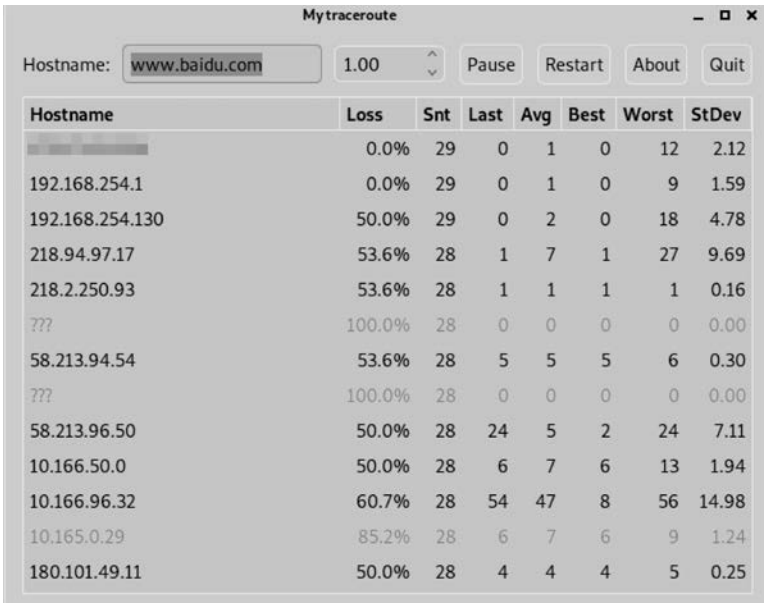
```
root@kali:~# traceroute -I www.baidu.com
traceroute to www.baidu.com (180.101.49.11), 30 hops max, 60 byte packets
 1  [redacted]  1.257 ms  1.566 ms  1.545 ms
 2  192.168.254.1 (192.168.254.1)  1.517 ms  1.495 ms  1.468 ms
 3  192.168.254.130 (192.168.254.130)  0.273 ms  0.572 ms  0.546 ms
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * 180.101.49.11 (180.101.49.11)  4.533 ms
```

图 5-9　使用 traceroute -I 发送 ICMP 追踪 www. baidu. com 后显示的结果

```
root@kali:~# traceroute -I www.baidu.com -q 10
traceroute to www.baidu.com (180.101.49.11), 30 hops max, 60
 1  [redacted]  1.742 ms  2.015 ms  1.98
 2  192.168.254.1 (192.168.254.1)  2.170 ms  2.146 ms  2.121
 3  192.168.254.130 (192.168.254.130)  0.667 ms  0.644 ms  0.
 4  * * * * * * * * * *
 5  * * * * * * * * * *
 6  * * * * * * * * * *
 7  * * * * * * * * * *
 8  * * * * * * * * * *
 9  * * * * * * * * * *
10  * * * * 10.166.50.0 (10.166.50.0)  6.698 ms  6.896 ms  6.
11  10.166.96.32 (10.166.96.32)  7.429 ms * * * * * * * * *
12  * 10.165.0.29 (10.165.0.29)  7.654 ms * * * * * * * *
13  180.101.49.11 (180.101.49.11)  4.558 ms  4.877 ms  4.854
```

图 5-10　使用 traceroute -q 发送 ICMP 追踪 www. baidu. com 后显示的结果

步骤 7：my trace route 工具的应用。在命令行终端中输入 mtr www. baidu. com 命令后，会出现如图 5-11 所示的路由追踪界面，该界面定时动态刷新结果。



Hostname	Loss	Snt	Last	Avg	Best	Worst	StDev
[redacted]	0.0%	29	0	1	0	12	2.12
192.168.254.1	0.0%	29	0	1	0	9	1.59
192.168.254.130	50.0%	29	0	2	0	18	4.78
218.94.97.17	53.6%	28	1	7	1	27	9.69
218.2.250.93	53.6%	28	1	1	1	1	0.16
???	100.0%	28	0	0	0	0	0.00
58.213.94.54	53.6%	28	5	5	5	6	0.30
???	100.0%	28	0	0	0	0	0.00
58.213.96.50	50.0%	28	24	5	2	24	7.11
10.166.50.0	50.0%	28	6	7	6	13	1.94
10.166.96.32	60.7%	28	54	47	8	56	14.98
10.165.0.29	85.2%	28	6	7	6	9	1.24
180.101.49.11	50.0%	28	4	4	4	5	0.25

图 5-11　使用 mtr 追踪 www. baidu. com 的显示信息

步骤 8：tctrace 工具的应用。tctrace 工具命令格式与 itrace 类似，功能上与 traceroute 使用-T 参数类似。运行 tctrace -i eth0 -d www.baidu.com 命令，可追踪到百度服务器的路由路径。如图 5-12 所示，追踪的最后一跳显示 reached；open 信息，说明已经追踪成功。

```
┌──(root💀kali)-[/usr/local/bin]
└─# tctrace -i eth0 -d www.baidu.com
 1(1)   [192.168.0.1]
 2(1)   [192.168.1.1]
 3(1)   [10.215.60.1]
4(all)  Timeout
5(all)  Timeout
6(all)  Timeout
7(all)  Timeout
 8(1)   [36.152.44.95] (reached; open)
```

图 5-12　使用 tctrace 工具成功追踪 www.baidu.com 主机后的显示信息

5.1.4　任务与思考

通过本实验，虽然读者已经掌握了路由信息的获得方式，但为了能够适应复杂网络环境下的攻防要求，还需要进一步对路由器的相关功能进行学习。路由器的主要功能包括以下几方面。

（1）在不同的网络间接收分组，然后根据分组中的目的 IP 地址来查询路由表，再通过合适的接口将分组转发出去。

（2）选择最合理的路由，引导通信。为了实现这一功能，路由器要按照某种路由通信协议（典型协议有 RIP、OSPF、BGP 等）查找路由表。网络中的每个路由器按照这一规则动态地更新它所保存的路由表，以便维护有效的路由信息。

（3）在转发分组的过程中，为了便于在网络间传送分组，路由器需要按照预定的规则把大的数据包（应用层的报文）分解成适合在不同网络之间自由传输的小数据包，到达目的地后再把分解的数据包重组成原有形式（应用层的报文）。

（4）多协议的路由器可以连接使用不同通信协议的网络段，作为不同通信协议网络段间通信连接的平台。

（5）路由器的主要任务是把通信引导到目的地网络（局域网），然后根据分组中的目的 IP 地址转发给指定的主机。后一个功能是通过网络地址解析协议（address resolution protocol，ARP）完成的。

（6）动态限速。动态限速路由器能够实时地计算每位用户所需要的带宽，精确分析用户上网类型，并合理分配带宽。

（7）缺乏源地址认证。路由器接收到一个分组时，正常情况下只会查看其目的 IP 地址，将以目的 IP 地址为查询路由表并转发分组，但其不会对源 IP 地址进行认证。即路由器对接收到的分组，只需要考虑它到哪里去，而不考虑它从哪里来。这一机制带来了网络安全问题，许多针对网络的攻击都是利用了这一机制，通过设置虚假的源 IP 地址来欺骗目的主机，再利用一些协议的工作机制（如 TCP 的三次握手）进行攻击。

扫一扫

视频讲解

5.2 主机扫描：主机探测

5.2.1 预备知识：主机扫描方法

主机扫描(host scan)是指通过对目标网络(一般为一个或多个 IP 网段)中主机 IP 地址的扫描确定目标网络中有哪些主机处于运行状态的技术。主机扫描的实现一般需借助 ICMP、TCP、UDP 等协议的工作机制，来探测并确定某一主机当前的运行状态和可被利用的资源(如打开的进程、开放的端口等)。

1. 基于 ICMP 协议的扫描方法

Internet 控制报文协议(internet control message protocol，ICMP)是 TCP/IP 协议栈的网际层提供的一个为主机或路由器报告差错或异常情况的协议。分组网间探测(Packet Internet Groper，PING)是 ICMP 的一个重要的应用功能，它是应用层直接调用网际层 ICMP 的一个特殊应用，通过使用 ICMP 回送请求与回送应答报文来探测两台主机之间网络的连通性。

2. 基于 TCP 的主机扫描方法

传输控制协议(transmission control protocol，TCP)是一种面向连接的、可靠的、基于字节流的传输层通信协议。任意两个节点间每一个 TCP 通信的建立都需要有连接建立、数据传输和连接释放这 3 个过程(即 TCP 三次握手)，其目的是让通信的双方都知道彼此的存在，并通过双方协商来确定具体的通信参数(如缓存大小、连接表中的项目、最大窗口值等)。

3. 基于 UDP 的主机扫描方法

用户数据报协议(user datagram protocol，UDP)是一个无连接(没有提供三次握手过程)的、尽最大努力交付(不可靠)的、面向报文(保留了报文的边界)的传输层通信协议。与 TCP 相比，UDP 最大的优点是占用资源少、效率高，最大的缺点是不可靠。

5.2.2 实验目的和条件

1. 实验目的

主机探测是主机扫描过程的重要组成部分。通过主机探测，可以在确定的范围(一般为一个或多个 IP 地址段)发现正在运行(存活)的主机，为下一步攻击(端口扫描和操作系统类型确定)奠定基础。在网络攻击过程中，每一个实现步骤之间都是相互关联和相互影响的，前一个环节的操作成果是后一个环节的基础。主机探测的主要目的是确定被攻击对象，只有对象的确定是准确无误的，那么后续的工作开展才会有价值和意义。

通过本实验，读者需要掌握 Kali Linux 环境下 ping、arping、fping 和 nbtscan 工具的使用方法，以及不同工具的应用特点和功能区别。

2. 实验条件

本实验需要在网络环境中进行。建议实验在一个局域网内部进行，这样可以通过实验发现本局域网中有哪些主机处于运行状态。例如，一个局域网中有 50 台主机，为了实验，可

以让其中的 10 台(随意确定)运行,其他主机处于关闭状态。通过本实验,将实验结果与实际情况进行对比分析,以验证实验结果的正确性和可信性。

本实验中使用的攻击主机仍然是运行 Kali Linux 2021 系统的计算机。

5.2.3 实验过程

步骤 1:正确登录 Kali Linux 系统。如果进入的是命令行模式,为方便实验进行,可输入 startx 命令切换到图形界面。选择菜单栏上的 Terminal 选项,打开终端操作窗口。

步骤 2:ping 工具的应用。ping 是非常著名的用来检查主机是否在线的工具。该工具的工作原理基于发送 ICMP ECHO Request 包到目标主机,如果目标主机在线并且不对 ping 请求数据包进行阻止时,将回复一个 ICMP ECHO Reply 数据包。ping 命令的选项较多,最常用的有以下几个。

-c count:ECHO_Request 包发送数量。

-i interface address:源地址网络接口,该参数可以是 IP 地址或网卡名称。

-s packetsize:指定要发送的数据字节数,默认值是 56 字节,然后再与 8 字节的 ICMP 头数据组成 64B 的 ICMP 数据包。

在实验中,如果要检查目标 IP 地址 192.168.244.132(被攻击对象的 IP 地址),且发送两个大小为 1000 字节的包,其命令为 ping -c 2 -s 1000 192.168.244.132,如图 5-13 所示。

```
┌──(root㉿kali)-[~]
└─# ping -c 2 -s 1000 192.168.244.132                                    1 ×
ping: socket: Address family not supported by protocol
PING 192.168.244.132 (192.168.244.132) 1000(1028) bytes of data.
1008 bytes from 192.168.244.132: icmp_seq=1 ttl=128 time=0.349 ms
1008 bytes from 192.168.244.132: icmp_seq=2 ttl=128 time=0.322 ms

--- 192.168.244.132 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1030ms
rtt min/avg/max/mdev = 0.322/0.335/0.349/0.013 ms
```

图 5-13 ping -c 2 -s 1000 192.168.244.132 的运行过程和结果

步骤 3:arping 工具的应用。arping 是一个在局域网中利用 ARP 来探测目标主机连通性的工具。arping 工具在测试特定 IP 地址在网络中是否使用时非常有用。该命令只能运行在本地局域网内,无法跨越路由器和网关。在终端窗口中,可以输入 arping 命令,按 Enter 键获取该命令的所有选项及其使用方法介绍。常用的选项为 apring -c。例如,输入 arping -c 5 192.168.244.132 命令,其运行过程和结果如图 5-14 所示(可与图 5-13 中 ping 命令的运行过程和结果进行对比)。

步骤 4:fping 工具的应用。fping 工具可以同时向多个目标主机(主机列表)发送 ping (ICMP ECHO)请求包。主机列表可以在命令行中指定也可以通过包含目标主机的文件指定。默认模式下,fping 通过监视目标主机的回复判断主机是否可用。如果目标主机返回应答,其信息将会从目标记录清单中删除;如果主机在一段时间内不响应(超时或超过尝试次数),该主机将会被标记为不可达。默认情况下,fping 将尝试向每个目标发送 3 个 ICMP ECHO 数据包。

在终端中执行 fping -h 命令,可以查看该命令的帮助文档,如图 5-15 所示。

fping 可以识别多个主机,例如,通过 fping 192.168.244.129 192.168.244.132 命令

```
┌──(root💀kali)-[~]
└─# arping
ARPing 2.21, by Thomas Habets <thomas@habets.se>
usage: arping [ -0aAbdDeFpQrRuUv ] [ -w <sec> ] [ -W <sec> ] [ -S <host/ip> ]
              [ -T <host/ip ] [ -s <MAC> ] [ -t <MAC> ] [ -c <count> ]
              [ -C <count> ] [ -i <interface> ] [ -m <type> ] [ -g <group> ]
              [ -V <vlan> ] [ -Q <priority> ] <host/ip/MAC | -B>
For complete usage info, use --help or check the manpage.

┌──(root💀kali)-[~]
└─# arping -c 5 192.168.244.132                                          1 ×
ARPING 192.168.244.132
60 bytes from 00:0c:29:59:16:06 (192.168.244.132): index=0 time=341.404 usec
60 bytes from 00:0c:29:59:16:06 (192.168.244.132): index=1 time=209.794 usec
60 bytes from 00:0c:29:59:16:06 (192.168.244.132): index=2 time=244.448 usec
60 bytes from 00:0c:29:59:16:06 (192.168.244.132): index=3 time=192.691 usec
60 bytes from 00:0c:29:59:16:06 (192.168.244.132): index=4 time=435.251 usec

--- 192.168.244.132 statistics ---
5 packets transmitted, 5 packets received,   0% unanswered (0 extra)
rtt min/avg/max/std-dev = 0.193/0.285/0.435/0.091 ms
```

图 5-14　arping -c 5 192.168.244.132 的运行过程和结果

```
┌──(root💀kali)-[~]
└─# fping -h
Usage: fping [options] [targets...]

Probing options:
   -4, --ipv4         only ping IPv4 addresses
   -6, --ipv6         only ping IPv6 addresses
   -b, --size=BYTES   amount of ping data to send, in bytes (default: 56)
   -B, --backoff=N    set exponential backoff factor to N (default: 1.5)
   -c, --count=N      count mode: send N pings to each target
   -f, --file=FILE    read list of targets from a file ( - means stdin)
   -g, --generate     generate target list (only if no -f specified)
                      (give start and end IP in the target list, or a CIDR address)
                      (ex. fping -g 192.168.1.0 192.168.1.255 or fping -g 192.168.1.0/24)
   -H, --ttl=N        set the IP TTL value (Time To Live hops)
   -I, --iface=IFACE  bind to a particular interface
   -l, --loop         loop mode: send pings forever
```

图 5-15　fping 命令的帮助文档

(不同的 IP 地址之间用一个空格隔开)，可以查看当前这两台主机是否处于运行状态，运行过程和结果如图 5-16 所示。

```
┌──(root💀kali)-[~]
└─# fping 192.168.244.129 192.168.244.132                                1 ×
192.168.244.129 is alive
192.168.244.132 is alive
```

图 5-16　使用 fping 192.168.244.129 192.168.244.132 命令同时查看两台主机是否在线

另外，如果需要查看多个目标主机的统计结果，可以使用 fping -s 命令加一个或多个域名(或 IP 地址)命令，如图 5-17 所示。

输入 fping -g -a -q 192.168.138.0/24 命令，可以生成目标网络中可用主机清单，如图 5-18 所示。

步骤 5：NBTScan 工具的应用。NBTScan 工具可用于扫描网络上 NetBIOS 名称信息，该工具对给出范围内的每一个地址发送 NetBIOS 状态查询，对于每个响应的主机来说，NBTScan 列出它的 IP 地址、NetBIOS 计算机名、登录用户名和 MAC 地址，但其只能用于局域网。直接执行 nbtscan 命令，会显示该命令的帮助文档，如图 5-19 所示。

```
┌──(root㉿kali)-[~]
└─# fping -s 192.168.244.129 192.168.244.132 192.168.244.130 www.baidu.com
192.168.244.129 is alive
192.168.244.132 is alive
www.baidu.com is alive
ICMP Host Unreachable from 192.168.244.128 for ICMP Echo sent to 192.168.244.130
ICMP Host Unreachable from 192.168.244.128 for ICMP Echo sent to 192.168.244.130
ICMP Host Unreachable from 192.168.244.128 for ICMP Echo sent to 192.168.244.130
ICMP Host Unreachable from 192.168.244.128 for ICMP Echo sent to 192.168.244.130
192.168.244.130 is unreachable

       4 targets
       3 alive
       1 unreachable
       0 unknown addresses

       4 timeouts (waiting for response)
       7 ICMP Echos sent
       3 ICMP Echo Replies received
       4 other ICMP received

 0.256 ms (min round trip time)
 2.17 ms (avg round trip time)
 5.96 ms (max round trip time)
        4.096 sec (elapsed real time)
```

图 5-17　使用 fping 命令同时查看多个目标主机

```
root@kali:~#
root@kali:~# fping -g -a -q 192.168.138.0/24
192.168.138.2
192.168.138.131
192.168.138.132
root@kali:~#
root@kali:~#
root@kali:~#
```

图 5-18　使用 fping 命令生成可用主机清单

```
┌──(root㉿kali)-[/]
└─# nbtscan

NBTscan version 1.6.
This is a free software and it comes with absolutely no warranty.
You can use, distribute and modify it under terms of GNU GPL 2+.

Usage:
nbtscan [-v] [-d] [-e] [-l] [-t timeout] [-b bandwidth] [-r] [-q] [-s separator] [-m ret
ransmits] (-f filename)|(<scan_range>)
        -v              verbose output. Print all names received
                        from each host
        -d              dump packets. Print whole packet contents.
        -e              Format output in /etc/hosts format.
        -l              Format output in lmhosts format.
                        Cannot be used with -v, -s or -h options.
        -t timeout      wait timeout milliseconds for response.
                        Default 1000.
        -b bandwidth    Output throttling. Slow down output
                        so that it uses no more that bandwidth bps.
                        Useful on slow links, so that ougoing queries
                        don't get dropped.
        -r              use local port 137 for scans. Win95 boxes
                        respond to this only.
```

图 5-19　nbtscan 工具的帮助文档信息

例如，如果要显示 192.168.224.0/24 网段中的可用主机，可以使用 nbtscan 192.168.244.0/24 命令，如图 5-20 所示。

```
┌──(root㉿kali)-[/]
└─# nbtscan 192.168.244.0/24
Doing NBT name scan for addresses from 192.168.244.0/24

IP address       NetBIOS Name     Server    User             MAC address
-------------------------------------------------------------------------------
192.168.244.0    Sendto failed: Permission denied
192.168.244.1    DESKTOP-UVD6SM1  <server>  <unknown>        00:50:56:c0:00:08
192.168.244.129  DESKTOP-3E04EKE  <server>  <unknown>        00:0c:29:d0:eb:e3
192.168.244.132  SMITON-59E654AC  <server>  <unknown>        00:0c:29:59:16:06
192.168.244.255 Sendto failed: Permission denied
```

图 5-20　使用 nbtscan 工具同时显示 192.168.244.0/24 网段中的可用主机

5.2.4　任务与思考

通过本实验的练习，读者还需要继续学习和掌握基于 TCP 协议的主机扫描方法。

基于 TCP 的主机扫描方法的关键是 TCP 的三次握手过程。其中，ACK 表示 Server（服务器）对 Client（客户端）请求建立的确认，但是，如果 Client 根本没有进行 SYN 请求（第一次握手），而是直接进行确认（第三次握手），此时，Server 就会认为出现了一个重要的错误，会向 Client 发送一个头部“复位”（RST）字段为 1 的报文，告诉 Client 必须释放本次连接，再重新建立 TCP 连接。根据该工作机制，如果攻击者向目标主机发送一个只有 ACK 的报文，当接收到目标主机一个 RST 反馈报文时，就可以确认目标主机的存在。

另一种是利用 TCP 三次握手过程针对主机的 SYN 扫描。如果目标主机处于运行状态，但主机上的服务器进程没有打开，则目标主机将返回一个 RST 报文；如果目标主机上的服务器进程处于“监听”（listen）状态，则会返回一个第二次握手的 ACK/SYN 报文。不管返回哪一种报文，都可以从中判断目标主机的当前状态。

以上探测方法需要读者在继续深入学习 TCP 及相关协议工作原理的基础上，借助相关的工具软件通过具体的实验来学习和掌握。

扫一扫

视频讲解

5.3　端口扫描：Zenmap 工具的应用

5.3.1　预备知识：端口扫描

端口扫描（port scan）是对正处于运行状态的主机使用的 TCP/UDP 端口进行探测的技术。端口是用于标识计算机应用层中的各个进程在与传输层交互时的层间接口地址，两台计算机间的进程在通信时，不仅仅要知道对方的 IP 地址，还要知道对方的端口号。为此，可以将端口理解为进入计算机应用进程的窗口，在 TCP 和 UDP 中端口用 16b 字段表示，其值为 0～65 535。传输层的端口分为服务器端使用的端口号和客户端使用的端口号两大类。其中，服务器端使用的端口号又分为两类：一类为熟知端口号（well known ports）或系统端口号，其值为 0～1023，可以在 http://www.iana.org 网站上查到；另一类为登记端口号，其值为 1024～49 151，使用这类端口时需要在 IANA（the internet assigned numbers authority，互联网数字分配机构）进行登记。客户端使用的端口号被称为短暂端口号，其值为 49 152～65 535，仅在客户进程运行时临时使用，通信结束后即会被收回。

由于 TCP 和 UDP 可以使用相同的端口号(如 DNS 同时使用了 TCP 53 和 UDP 53 两个端口号),因此端口扫描需要分别针对 TCP 和 UDP 的端口号进行。由于 TCP 要比 UDP 复杂,因此针对 TCP 端口的扫描也要比 UDP 端口扫描复杂。TCP 端口扫描包括连接(connect)扫描、SYN 扫描、TCP 窗口扫描、FIN 扫描、ACK 扫描等。

1. 连接扫描

攻击者(扫描主机)通过调用系统的 connect()函数,可以与目标主机的每个端口尝试通过三次握手建立 TCP 连接,在攻击者发起连接请求(第一次握手)后,如果目标主机上对应的端口已被打开,则返回一个第二次握手的 ACK/SYN 报文,connect()调用将再发送一个 ACK 确认报文以完成第三次握手。如果目标端口是关闭的,那么目标主机将会直接返回一个 RST 报文。基于此工作原理,通过分析不同目标端口的返回报文信息,攻击者就可以判断哪些端口是开放或关闭的。该方法实现简单,但目标主机上会记录相关的尝试连接信息,容易被系统管理员或安全检测软件发现。

2. SYN 扫描

SYN 扫描也被称为半开连接扫描,是对连接扫描的一种改进。在连接扫描方法中,当被扫描端口打开时,目标主机会返回一个 SYN/ACK 报文。当攻击者收到第二次握手的 SYN/ACK 报文时,其实不需要进行第三次 ACK 握手,就已经可以判断出被扫描端口当前是否处于打开状态。不过,当目标主机(server)向 TCP 连接请求者(client)返回 SYN/ACK 报文后,其将处于"半开连接"状态,等待请求者的 ACK 确认,以便完成第三次握手过程。此时,攻击者并没有向目标主机返回 ACK 确认报文,而是构造了一个 RST 报文,让目标主机自行释放该"半开连接"。

由于各类操作系统一般不会记录"半开连接"信息,因此 SYN 扫描的安全性要比连接扫描好。

3. UDP 端口扫描

UDP 端口扫描用于探测目标主机上已打开的 UDP 端口和网络服务。UDP 端口扫描的实现原理是:首先构造并向目标主机发送一个特殊的 UDP 报文,如果被扫描的 UDP 端口关闭,将返回一个基于 ICMP 的"端口不可达"差错报文;如果被扫描的 UDP 端口处于打开状态,处于"监听"状态的 UDP 网络服务将响应特殊定制的数据报文,从而返回 UDP 数据。

UDP 端口扫描的实现原理简单,效率较高。但是,如果被探测的网络服务是一个未知的应用时,就可能无法返回 UDP 数据。

5.3.2 实验目的和条件

1. 实验目的

在学习 TCP 三次握手、TCP/UDP 端口、服务进程等网络基本知识的基础上,通过对 Zenmap 和 nmap 工具使用方法的练习,进一步掌握端口扫描的实现方法。

2. 实验条件

为了取得更好的实验效果,建议本实验在局域网中进行。本实验中攻击者采用运行

Windows 10 操作系统的计算机，需要在该主机上安装 nmap 程序，目前最新版本为 7.9.2，官方下载地址为：https://nmap.org/dist/nmap-7.92-setup.exe，该程序同时支持控制台和图形界面两种运行模式。

5.3.3 实验过程

步骤 1：运行程序 Nmap - Zenmap GUI，启动图形化的 Zenmap 工具。

Zenmap 默认提供 10 种可供选择的扫描方式，可以通过单击“配置”菜单项来选择。在选择了具体的扫描方式后，就可以看见相应扫描方式所采用的命令，所执行的命令显示在“命令”框中，如图 5-21 所示。

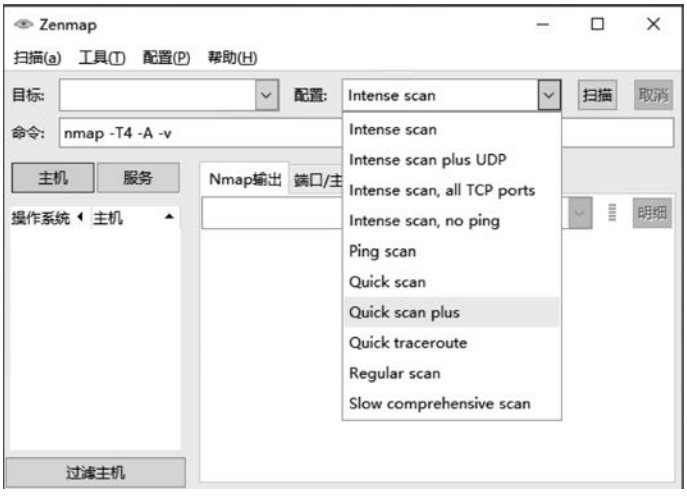


图 5-21 Zenmap 默认提供的扫描方式

步骤 2：如果扫描方式不符合攻击者的当前要求，可以创建一个新的扫描方式（新的配置或命令），或者在已有扫描方式的基础上进行编辑（编辑选中配置），如图 5-22 所示。

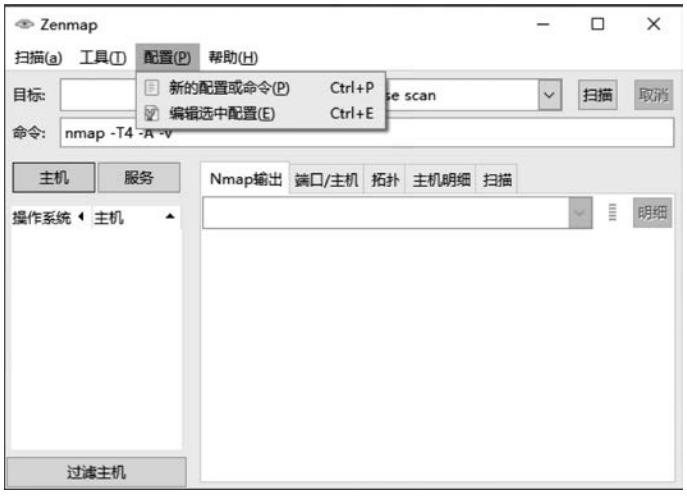


图 5-22 创建一个新扫描方式

步骤 3：在本实验中，可以选择“新的配置或命令”选项，打开如图 5-23 所示对话框，在该对话框中出现了“配置”“扫描”、Ping、“脚本”“目标”“源”“其他”“定时”等选项卡。读者可根据具体的攻击实验需要进行选择和配置。

在“配置”选项卡中“配置文件名”文本框中输入一个标识本次扫描操作的名称 SYNPortScan。

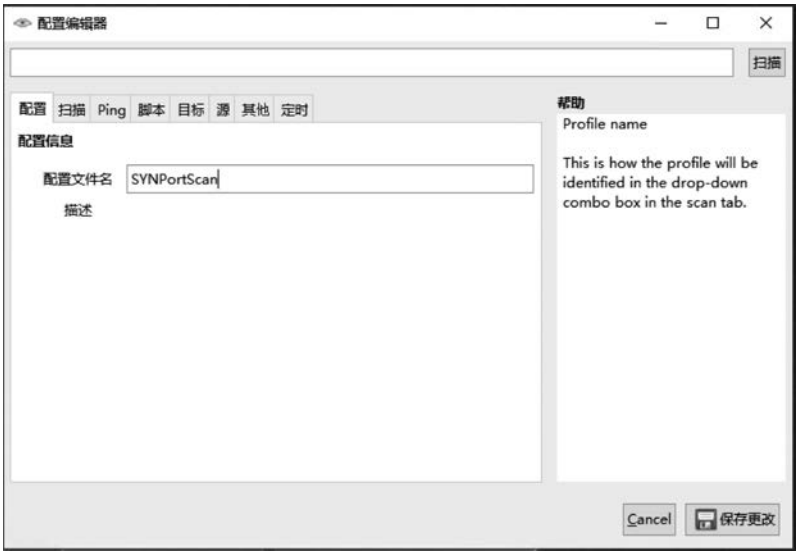


图 5-23　设置新扫描配置的文件名

步骤 4：选择“扫描”选项卡后，出现如图 5-24 所示的对话框。本实验将对 192.168.244.0/24 这个 C 网段所有主机进行端口扫描，因此在“目标(可选)”文本框中输入：192.168.244.0/24。由于扫描方式选择半连接扫描，因此“TCP 扫描”处选择“TCP SYN 扫描(-sS)”，其他选项设置为无。所有信息输入结束后，单击“保存更改”按钮进行确认。

图 5-24　对扫描对象及相关参数的配置

需要说明的是，在具体实验过程中，读者可结合不同情况，通过选取不同的选项，学习扫描过程，并对扫描结果进行分析，以便对工具的应用和功能有更加全面的认识，同时对知识的系统掌握也会有所帮助。

步骤 5：选择刚才新建的扫描配置 SYNPortScan，“命令”处将显示该配置的命令，如本实验中的 nmap -sS 192.168.244.0/24，如图 5-25 所示。

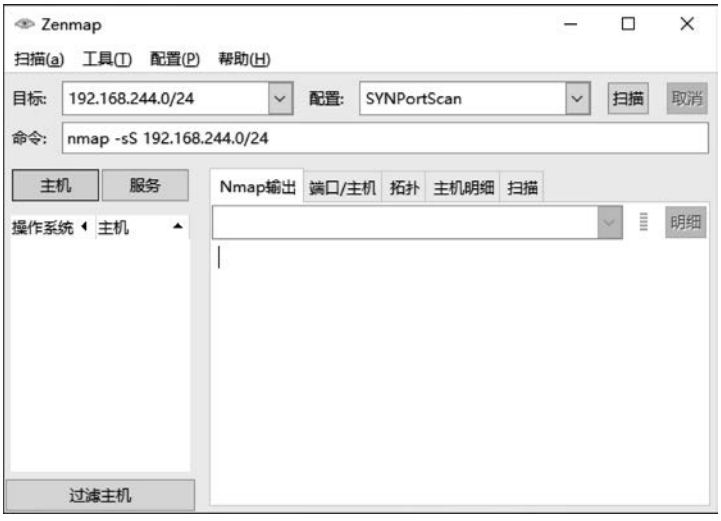


图 5-25　选择新建的扫描配置

步骤 6：单击“扫描”按钮，开始对 192.168.244.0/24 网段存在的主机进行端口扫描，扫描结果如图 5-26 所示。其中，在“主机”列表中显示了当前处于运行状态的所有主机，在右侧列表中显示了其中一台主机当前已打开的端口信息，包括端口号、当前状态、服务进程的名称及主机网卡的 MAC 地址等内容，收集到的信息非常全面。

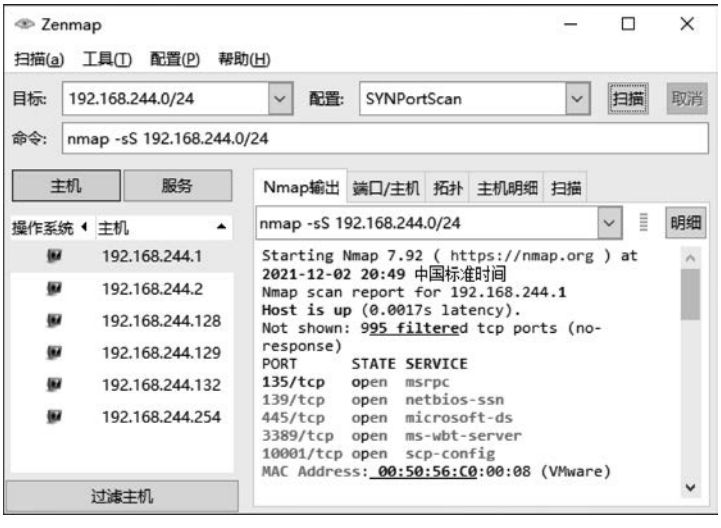


图 5-26　显示扫描结果

步骤 7：单击“拓扑”按钮，可以查看此次扫描发现的网络拓扑结构，如图 5-27 所示。

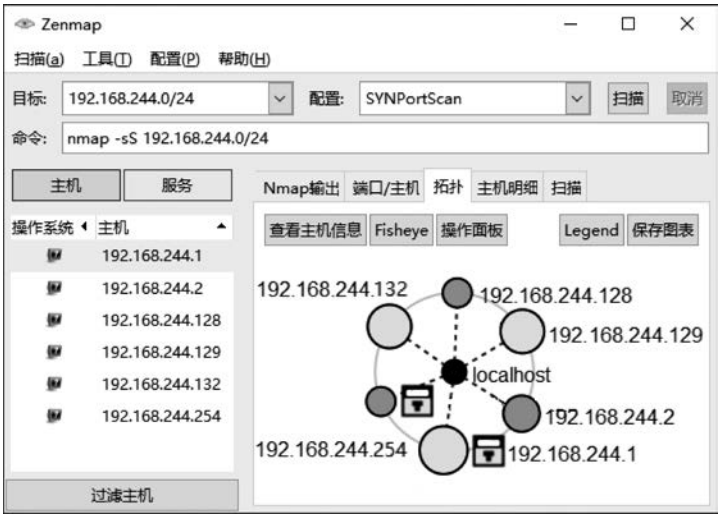


图 5-27　显示扫描结果的拓扑结构

步骤 8：如果要保存 Zenmap 的扫描结果，可以从“扫描”菜单中选择“保存扫描”选项，进行保存，默认保存格式为 XML，如图 5-28 所示。

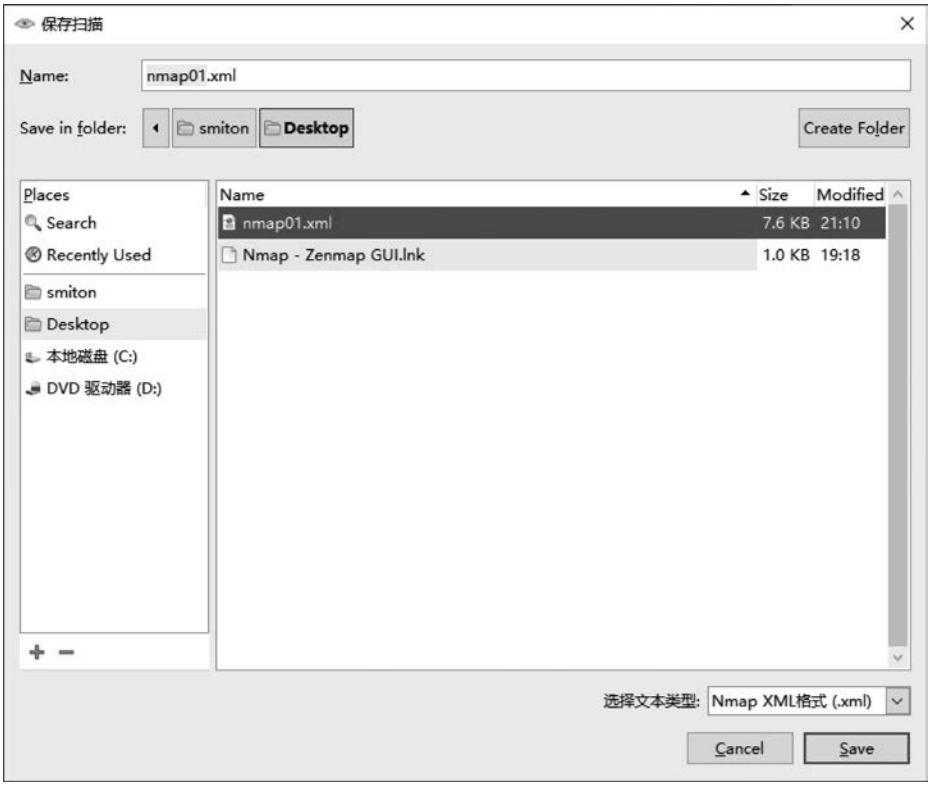


图 5-28　保存当前的扫描结果

步骤 9：在实验中，可以对同一对象范围或同一对象范围中的不同部分进行多次扫描，并对扫描结果进行比较，看是否存在不同。在本实验中，保存了第一次扫描结果后，接着修改扫描目标（如 192.168.244.1-118）（缩小了扫描范围），然后可以进行第二次扫描，并保存扫描结果，可将文件名确定为 nmap02.xml。

然后，在“工具”菜单中选择“结果比对”选项，对两次扫描结果进行比较。其中，在扫描 A 下拉列表中选择 nmap01.xml 选项，在扫描 B 下拉列表中选择 nmap02.xml 选项，比较结果如图 5-29 所示。

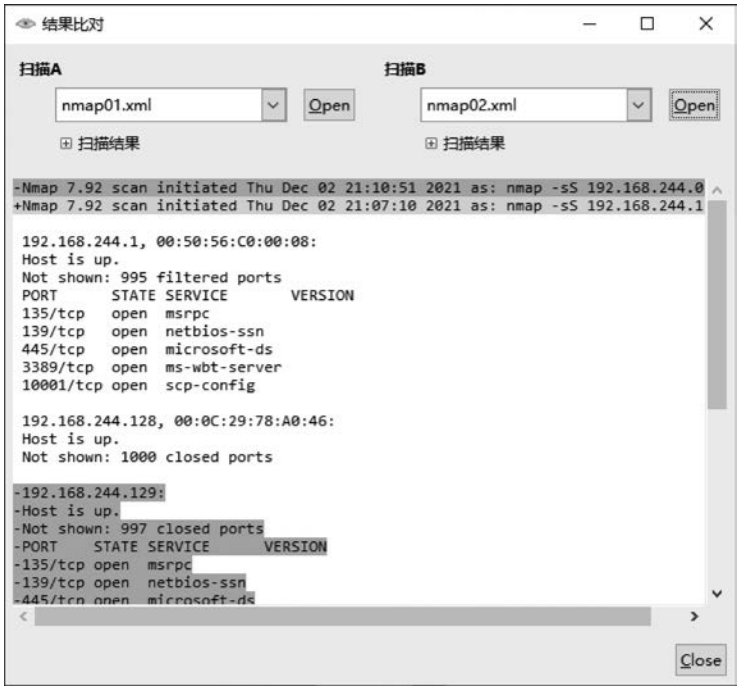


图 5-29　对两次扫描结果进行比较分析

其中，字符“-”（红色背景）表明扫描 B 中没有该行结果，相对地，字符“+”（绿色背景）表明扫描 B 中增加了该扫描结果。由图 5-29 可以看出，两次扫描的 IP 范围不同。

步骤 10：控制台式 nmap。在 Windows 10 操作系统中，选择“开始”→“运行”，在出现的对话框中输入 cmd 命令，进入命令提示符操作窗口。直接执行 nmap 命令，会显示该命令的帮助文档，如图 5-30 所示。

步骤 11：使用 namp -sS 192.168.244.0/24 命令，可实现与步骤 6 一样的功能，如图 5-31 所示。

5.3.4　任务与思考

通过本实验，使读者可以对 nmap 和 Zenmap 两个典型工具的功能特点和使用方法有一个较为全面的认识，通过比较分析，读者也可以掌握更多有关端口扫描的知识。与 nmap 相比，Zenmap 的优势表现在以下几方面。

（1）提供良好的交互性。Zenmap 可以更直观的方式输出结果，甚至能绘制其已发现网

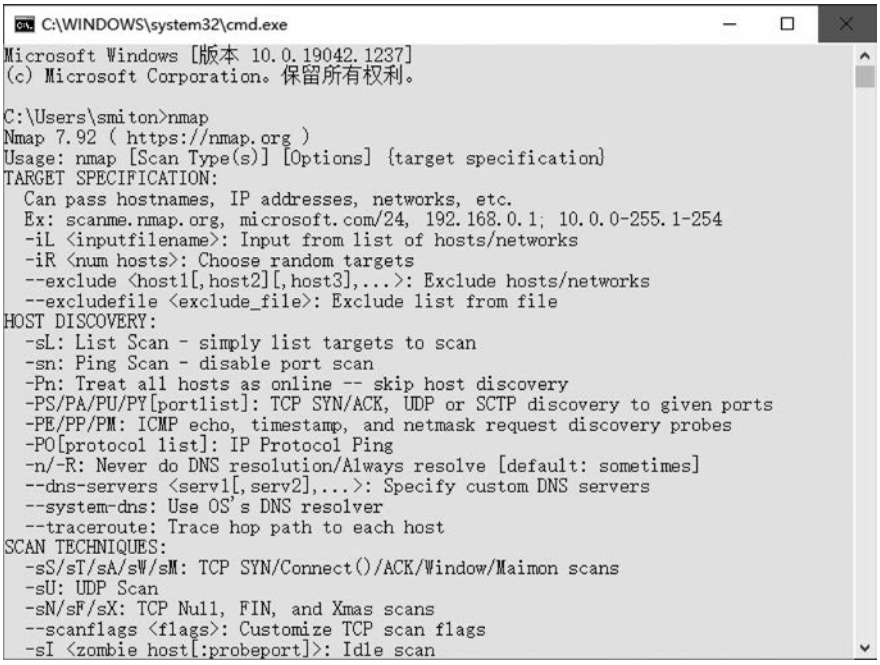


图 5-30　nmap 命令的帮助文档

图 5-31　命令行式 nmap 扫描

络的拓扑图。

（2）可以在两个扫描结果之间进行比较。

（3）能够对扫描结果进行跟踪。

（4）能够帮助渗透测试人员多次运行相同配置的扫描。

（5）显示所执行的命令，以方便渗透测试人员检查。

扫一扫

视频讲解

5.4 系统类型探测：主机系统识别

5.4.1 预备知识：主机探测

通过主机扫描和端口扫描，可以确定被攻击目标使用的 IP 地址及已开放的端口。在此基础上，还需要对被攻击主机所使用的操作系统类型和具体的版本号及提供的网络服务进行探测，为攻击者下一步选择具体的攻击方法并确定具体的攻击步骤做好准备。系统类型探测分为操作系统类型探测和网络服务类型探测两种。

1. 操作系统类型探测

操作系统类型探测（OS identification）是采取一定的技术手段，通过网络远程探测目标主机上安装的操作系统类型及其版本号的方法。在确定了操作系统的类型和具体版本号后，可以为进一步发现安全漏洞和渗透攻击提供条件。

协议栈指纹分析（stack fingerprinting）是一种主流的操作系统类型探测手段，其实现原理在于不同类型和版本的操作系统中，网络协议栈的实现方法存在着一些细微的区别，这些细微区别就构成了该版本操作系统的指纹信息。通过创建完整的操作系统协议栈指纹信息库，可以将探测或网络嗅探所得到的指纹信息在数据库中进行比对，精确地确定其操作系统的类型和版本号。

2. 网络服务类型探测

网络服务类型探测（service identification）的目的是确定目标主机上打开的端口，以及该端口上绑定的网络应用服务类型及版本号。通过网络服务类型探测，可以进一步确定目标主机上运行的网络服务及服务进程对应的端口。

操作系统类型探测主要依赖于 TCP/IP 协议栈的指纹信息，它涉及网络层、传输层、应用层等各层的信息；而网络服务类型探测主要依赖网络服务在应用层协议实现所包含的特殊指纹信息。例如，同样是在应用层提供 HTTP 服务的 Apache 和 IIS，两者在实现 HTTP 规范时的具体细节上存在一些差异，根据这些差异可以辨别出目标主机的 TCP 80 端口上运行的 HTTP 服务是通过 Apache 还是通过 IIS 实现的。

5.4.2 实验目的和条件

1. 实验目的

通过本实验，使读者对网络服务进程有更深入的学习，同时通过对主机系统识别工具使用方法的练习，掌握主机识别的主要方法和途径。

2. 实验条件

为便于实验的进行，本实验采用如表 5-1 所示的实验清单。

表 5-1 主机识别实验清单

类 型	序 号	软硬件要求
攻击机	1	数量：1 台
	2	操作系统版本：Kali Linux 2021
	3	软件版本：p0f，xprobe2，nmap
靶机	1	数量：1 台
	2	操作系统版本：Windows XP
	3	软件版本：无

5.4.3 实验过程

步骤 1：运行攻击机。正确登录 Kali Linux 系统，如果进入的是命令行模式，为方便实验进行，可输入 startx 命令进入图形界面。选择菜单栏上的 Terminal 选项，打开终端操作窗口。

步骤 2：运行 p0f 工具。p0f 工具利用 SYN 数据包实现操作系统被动检测技术，和其他扫描软件不同，它不向目标系统发送任何的数据，只是被动地接受来自目标系统的数据进行分析，进而实现目标系统类型的识别。因此，p0f 一个很大的优点是：几乎无法被检测到，而且 p0f 是专门的系统识别工具，其指纹数据库非常详尽，更新也比较快，特别适合安装在网关中。p0f 可以工作在连接到本地的机器、本地连接到的机器、不能连接到的机器、可以浏览其社区的机器等几种场景下。

执行 p0f -h 命令，系统将显示其所有参数及帮助说明（若该工具不存在，可以通过 apt install p0f 命令安装），如图 5-32 所示。

```
┌──(root💀kali)-[~]
└─# p0f -h
--- p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ---

p0f: invalid option -- 'h'
Usage: p0f [ ...options... ] [ 'filter rule' ]

Network interface options:

  -i iface  - listen on the specified network interface
  -r file   - read offline pcap data from a given file
  -p        - put the listening interface in promiscuous mode
  -L        - list all available interfaces

Operating mode and output settings:

  -f file   - read fingerprint database from 'file' (/etc/p0f/p0f.fp)
  -o file   - write information to the specified log file
  -s name   - answer to API queries at a named unix socket
  -u user   - switch to the specified unprivileged account and chroot
  -d        - fork into background (requires -o or -s)

Performance-related options:

  -S limit  - limit number of parallel API connections (20)
  -t c,h    - set connection / host cache age limits (30s,120m)
  -m c,h    - cap the number of active connections / hosts (1000,10000)

Optional filter expressions (man tcpdump) can be specified in the command
line to prevent p0f from looking at incidental network traffic.

Problems? You can reach the author at <lcamtuf@coredump.cx>.
```

图 5-32 p0f 的帮助信息

步骤 3：打开该程序后，首先输入 p0f -o p0f. log 命令，该命令会将登录信息保存到 p0f. log

文件中，如图 5-33 所示。此时，攻击机监听默认网口的 SYN 数据包，当目标主机与攻击机建立 TCP 连接，即可猜测目标主机的操作系统了。

```
┌──(root㉿kali)-[~]
└─# p0f -o p0f.log
--- p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ---

[+] Closed 1 file descriptor.
[+] Loaded 322 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on default interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file 'p0f.log' opened for writing.
[+] Entered main event loop.
```

图 5-33　输入 p0f -o p0f.log 命令将探测信息保存到 p0f.log 文件中

步骤 4：打开 Windows XP 并正常登录，运行靶机。

靶机正常启动后，需要产生一些网络活动以触发 TCP 连接。例如，可以 telnet 到一台主机（根据实验环境而定），接着就会识别系统类型。这里 telnet 虽然没有成功，但不影响实验效果，如图 5-34 所示。

图 5-34　通过 telnet 命令触发 TCP 连接

步骤 5：攻击机的命令行终端中会回显探测到目标主机的操作系统信息，如图 5-35 所示。从回显信息我们可以发现，p0f 工具已经分析出 IP 地址为 192.168.244.132 的目标主机的操作系统类型为 Windows NT，精确度不是很高。

```
┌──(root㉿kali)-[~]
└─# p0f -o p0f.log                                                    1 ×
--- p0f 3.09b by Michal Zalewski <lcamtuf@coredump.cx> ---

[+] Closed 1 file descriptor.
[+] Loaded 322 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on default interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file 'p0f.log' opened for writing.
[+] Entered main event loop.

.-[ 192.168.244.132/1041 → 192.168.244.128/23 (syn) ]-
|
|  client   = 192.168.244.132/1041
|  os       = Windows NT kernel
|  dist     = 0
|  params   = generic
|  raw_sig  = 4:128+0:0:1460:mss*44,0:mss,nop,nop,sok:df,id+:0
|
`----

.-[ 192.168.244.132/1041 → 192.168.244.128/23 (mtu) ]-
|
|  client   = 192.168.244.132/1041
|  link     = Ethernet or modem
|  raw_mtu  = 1500
```

图 5-35　分析探测到的主机类型

步骤 6：在攻击机上查看 p0f.log 文件。通过 cat p0f.log 命令，查看通过 p0f 工具被动

分析目标主机操作系统类型信息的历史记录，如图 5-36 所示。

```
┌──(root㉿kali)-[~]
└─# cat p0f.log
[2021/12/02 19:51:36] mod=syn|cli=192.168.244.132/1035|srv=192.168.244.128/23|subj=cli|os
=Windows NT kernel|dist=0|params=generic|raw_sig=4:128+0:0:1460:mss*44,0:mss,nop,nop,sok:
df,id+:0
[2021/12/02 19:51:36] mod=mtu|cli=192.168.244.132/1035|srv=192.168.244.128/23|subj=cli|li
nk=Ethernet or modem|raw_mtu=1500
[2021/12/02 19:51:36] mod=syn|cli=192.168.244.132/1035|srv=192.168.244.128/23|subj=cli|os
=Windows NT kernel|dist=0|params=generic|raw_sig=4:128+0:0:1460:mss*44,0:mss,nop,nop,sok:
df,id+:0
[2021/12/02 19:51:36] mod=mtu|cli=192.168.244.132/1035|srv=192.168.244.128/23|subj=cli|li
nk=Ethernet or modem|raw_mtu=1500
[2021/12/02 19:51:37] mod=syn|cli=192.168.244.132/1035|srv=192.168.244.128/23|subj=cli|os
=Windows NT kernel|dist=0|params=generic|raw_sig=4:128+0:0:1460:mss*44,0:mss,nop,nop,sok:
df,id+:0
```

图 5-36　p0f 工具分析的历史记录

步骤 7：使用 nmap 工具进行主机类型的探测。nmap 工具最著名的功能之一是用 TCP/IP 协议栈 fingerprinting 进行远程操作系统探测。与 p0f 不同，nmap 会主动发送一系列 TCP 和 UDP 报文到远程主机，检查响应中的每一个 bit，把结果和数据库 nmap-os-fingerprints 中超过 1500 个已知的操作系统的 fingerprints 进行比较，如果有匹配，就打印出操作系统的详细信息。在命令行终端中执行 nmap -O -PN -sV 192.168.244.132 命令，探测结果如图 5-37 所示。

```
┌──(root㉿kali)-[~]
└─# nmap -O -PN -sV 192.168.244.132
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-04 07:43 EST
Nmap scan report for 192.168.244.132
Host is up (0.00027s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:3F:23:BC (VMware)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.84 seconds
```

图 5-37　nmap 操作系统探测结果

从图中可以发现，主机 192.168.244.132 打开了 135、139 等 TCP 端口，其 MAC 地址为 00:0C:29:3F:23:BC，其操作系统类型为 Microsoft Windows XP SP2 or SP3，探测结果准确。

步骤 8：使用 xprobe2 工具进行主机类型的探测。xprobe2 是一个主动的操作系统识别工具，通过模糊签名匹配、可能性猜测、同时多匹配和签名数据库来识别操作系统。因为该工具使用原始套接字，所以其必须运行在 root 权限下。在命令行终端中直接执行 xprobe2 命令，系统将显示其所有参数及帮助说明（若该工具不存在，可以通过 apt install xprobe 命令安装），如图 5-38 所示。

步骤 9：对远程主机探测，可以直接通过 xprobe2，并指定远程主机 IP 地址或主机名。例如，通过 xprobe2 www.sina.com.cn 命令将会对该主机进行远程探测，显示结果如图 5-39 所示，可以看出远程主机为 FreeBSD 操作系统（注意：如果在虚拟机环境下采用 xprobe 进行操作系统类型探测，需将网络连接模式设置为非 NAT 模式，否则探测结果可能为乱码）。

```
┌──(root💀kali)-[~]
└─# xprobe2

Xprobe2 v.0.3 Copyright (c) 2002-2005 fyodor@o0o.nu, ofir@sys-security.com, meder@o0o.nu

usage: xprobe2 [options] target
Options:
          -v                      Be verbose
          -r                      Show route to target(traceroute)
          -p <proto:portnum:state> Specify portnumber, protocol and state.
                                  Example: tcp:23:open, UDP:53:CLOSED
          -c <configfile>         Specify config file to use.
          -h                      Print this help.
          -o <fname>              Use logfile to log everything.
          -t <time_sec>           Set initial receive timeout or roundtrip time.
          -s <send_delay>         Set packsending delay (milseconds).
          -d <debuglv>            Specify debugging level.
          -D <modnum>             Disable module number <modnum>.
          -M <modnum>             Enable module number <modnum>.
          -L                      Display modules.
          -m <numofmatches>       Specify number of matches to print.
          -T <portspec>           Enable TCP portscan for specified port(s).
                                  Example: -T21-23,53,110
          -U <portspec>           Enable UDP portscan for specified port(s).
          -f                      force fixed round-trip time (-t opt).
          -F                      Generate signature (use -o to save to a file).
          -X                      Generate XML output and save it to logfile specified with -o.
          -B                      Options forces TCP handshake module to try to guess open TCP port
          -A                      Perform analysis of sample packets gathered during portscan in
                                  order to detect suspicious traffic (i.e. transparent proxies,
                                  firewalls/NIDSs resetting connections). Use with -T.
```

图 5-38　显示 xprobe2 的帮助信息

```
┌──(root💀kali)-[~]
└─# xprobe2 www.sina.com.cn

Xprobe2 v.0.3 Copyright (c) 2002-2005 fyodor@o0o.nu, ofir@sys-security.com, meder@o0o.nu

[+] Target is www.sina.com.cn
[+] Loading modules.
[+] Following modules are loaded:
[x] [1] ping:icmp_ping  -  ICMP echo discovery module
[x] [2] ping:tcp_ping  -  TCP-based ping discovery module
[x] [3] ping:udp_ping  -  UDP-based ping discovery module
[x] [4] infogather:ttl_calc  -  TCP and UDP based TTL distance calculation
[x] [5] infogather:portscan  -  TCP and UDP PortScanner
[x] [6] fingerprint:icmp_echo  -  ICMP Echo request fingerprinting module
[x] [7] fingerprint:icmp_tstamp  -  ICMP Timestamp request fingerprinting module
[x] [8] fingerprint:icmp_amask  -  ICMP Address mask request fingerprinting module
[x] [9] fingerprint:icmp_port_unreach  -  ICMP port unreachable fingerprinting module
[x] [10] fingerprint:tcp_hshake  -  TCP Handshake fingerprinting module
[x] [11] fingerprint:tcp_rst  -  TCP RST fingerprinting module
[x] [12] fingerprint:smb  -  SMB fingerprinting module
[x] [13] fingerprint:snmp  -  SNMPv2c fingerprinting module
[+] 13 modules registered
[+] Initializing scan engine
[+] Running scan engine
[-] ping:tcp_ping module: no closed/open TCP ports known on 112.25.53.216. Module test failed
[-] ping:udp_ping module: no closed/open UDP ports known on 112.25.53.216. Module test failed
[-] No distance calculation. 112.25.53.216 appears to be dead or no ports known
[+] Host: 112.25.53.216 is up (Guess probability: 50%)
[+] Target: 112.25.53.216 is alive. Round-Trip Time: 0.50508 sec
[+] Selected safe Round-Trip Time value is: 1.01015 sec
[-] fingerprint:tcp_hshake Module execution aborted (no open TCP ports known)
[-] fingerprint:smb need either TCP port 139 or 445 to run
[-] fingerprint:snmp: need UDP port 161 open
[+] Primary guess:
[+] Host 112.25.53.216 Running OS: "FreeBSD 4.9" (Guess probability: 100%)
[+] Other guesses:
[+] Host 112.25.53.216 Running OS: =FQV (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: =FQV (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: =FQV (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: =FQV (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: =FQV (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: =FQV (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: "FreeBSD 5.4" (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: "FreeBSD 5.3" (Guess probability: 100%)
[+] Host 112.25.53.216 Running OS: "FreeBSD 5.2.1" (Guess probability: 100%)
[+] Cleaning up scan engine
[+] Modules deinitialized
[+] Execution completed.
```

图 5-39　显示探测到的远程主机操作系统类型

5.4.4 任务与思考

通过本实验，读者将对 Kali Linux 的操作方法有更全面的掌握。同时，还有一个问题需要读者思考：获取到系统版本后能做什么？

其实，回答了这个问题，也就为后面的实验提前做了准备。获得系统版本之后可以继续探测该系统存在哪些漏洞，然后利用漏洞找到对应的攻击工具，进一步获取攻击过程中所需要的信息。

5.5 漏洞扫描：Web 安全漏洞扫描及审计

5.5.1 预备知识：Web 漏洞的获取方法与 w3af

1. 漏洞扫描

漏洞扫描除可用于网络攻击外，还可用于对网络的安全防御。系统管理员能够通过对网络漏洞的系统扫描，全面地了解网络的安全状态，并对发现的安全漏洞及时安装补丁程序，提升网络防范攻击的能力。

漏洞扫描技术的工作原理是基于目标对象（操作系统、网络服务、应用程序等）的特征码来实现的。例如，对于同一个类型和版本号的操作系统来说，针对某一安全漏洞，对于某些网络请求的应答，安装安全补丁前后会存在一些细微的差异，这些差异便构成了针对特定安全漏洞的特征码（指纹信息）。漏洞扫描技术正是利用了这些特征码来识别目标对象是否存在特定的安全漏洞。

2. 漏洞扫描器

网络漏洞扫描器对目标系统进行漏洞检测时，首先探测目标网络中的存活主机，对存活主机进行端口扫描，确定系统已打开的端口，同时根据协议栈指纹技术识别出主机的操作系统类型。然后，扫描器对开放的端口进行网络服务类型的识别，确定其提供的网络服务。漏洞扫描器根据目标系统的操作系统平台和提供的网络服务，调用漏洞资料库（一般该资料库需要与业界标准的 CVE 保持兼容）中已知的各种漏洞进行逐一检测，通过对探测响应数据包的分析判断漏洞是否存在。

3. w3af

Web 应用攻击与审计架构（web application attack and audit framework，w3af）是一个 Web 应用安全的攻击、审计平台，其通过增加插件的方式来对功能进行扩展。w3af 是一款用 Python 语言编写的工具，同时支持 GUI 和命令行模式。

w3af 目前已经集成了大量的功能丰富的各类攻击和审计插件，为便于使用，其对插件进行了分类，而且有些插件还提供了实用工具，并支持多种加/解密算法。下面介绍几类典型的 w3af 插件。

（1）Crawl 类插件。Crawl（爬取）类插件的功能是通过爬取网站站点获得新的 URL 地

址。如果用户启用了 Crawl 类的插件，则其将会产生一个循环操作：A 插件在第一次运行时发现了一个新的 URL，w3af 会将其发送到插件 B；如果插件 B 发现了一个新的 URL，则会将其发送到插件 A。这个过程持续进行，直到所有插件都已运行且无法找到更多的新信息为止。

（2）Audit 类插件。Audit（审计）类插件会向 Crawl 插件爬取出的注入点发送特制的探测信息，以确认是否存在漏洞。

（3）Attack 类插件。如果 Audit 插件发现了漏洞，Attack（攻击）类插件将会利用该漏洞进行攻击，通常会在远程服务器上返回一个操作界面，或进行 SQL 注入以获取数据库中的数据。

（4）Infrastructure 类插件。Infrastructure（基础设施）类插件用来探测有关目标系统的信息，如目标系统是否安装了 Web 应用程序防火墙（web application firewall，WAF）和目标系统上运行的操作系统 HTTP 守护进程等。

（5）Grep 类插件。Grep（检索）类插件会分析其他插件发送的 HTTP 请求和应用信息，并识别存在的漏洞。

（6）Output 类插件。Output（输出）类插件会将插件的数据以文本、XML 或 HTML 形式保存，供分析使用。另外，如果启用了 text_file 和 xml_file 两个 Output 插件，就会记录有关 Audit 类插件发现的任何漏洞。

另外，Mangle 类插件允许用户修改基于正则表达式的请求和响应；Broutforce 类插件在爬取阶段可以对系统进行暴力登录；Evasion 类插件通过修改由其他插件生成的 HTTP 请求来绕过简单的入侵检测规则。

5.5.2 实验目的和条件

1. 实验目的

在进行本实验之前，读者需要对漏洞的产生、安全威胁及管理方法有所掌握。在此基础上，通过对 w3af 工具使用方法的学习，使读者能够掌握服务器安全漏洞的扫描和审计方法。

2. 实验条件

本实验中使用的软硬件清单如表 5-2 所示。

表 5-2 Web 安全漏洞扫描及审计实验清单

类　型	序　号	软硬件要求
攻击机	1	数量：1 台
	2	操作系统版本：Kali Linux 2020
	3	软件版本：w3af 2019.1.2
靶机	1	数量：1 台
	2	操作系统版本：Windows XP
	3	软件版本：XAMPP，DVWA

5.5.3 实验过程

步骤 1：运行攻击机，正确登录 Kali Linux 2020 系统，选择菜单栏上的 Terminal 选项，

打开终端操作窗口。

步骤 2：运行 cd w3af-master 命令，切换到 w3af 工作目录，然后使用 ls 命令查看当前目录下的文件，如图 5-40 所示。

```
root@kali:~# cd w3af-master/
root@kali:~/w3af-master# ls
circle.yml  extras    README.md  scripts  w3af       w3af_console
doc         profiles  result     tools    w3af_api   w3af_gui
root@kali:~/w3af-master#
```

图 5-40　显示 w3af-master 目录下的内容

步骤 3：使用./w3af_console 命令启用 w3af，并转到个性化的控制台模式(w3af >>>)，如图 5-41 所示。虽然该工具同时提供了 GUI 版本，但考虑到控制的灵活性和自定义配置的方便，在具体应用中建议使用控制台版本。

```
root@kali:~# cd w3af-master/
root@kali:~/w3af-master# ls
circle.yml  extras    README.md  scripts  w3af       w3af_console
doc         profiles  result     tools    w3af_api   w3af_gui
root@kali:~/w3af-master# ./w3af_console
/usr/local/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2
is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and
will be removed in a future release.
  from cryptography import x509
w3af>>>
```

图 5-41　w3af 的个性化控制台模式

为便于实验的进行，建议读者使用 help 帮助命令，查看相关的命令介绍。

步骤 4：首先使用 plugins 命令配置相关的插件，然后用 help 命令查看操作命令的帮助信息，如图 5-42 所示。

```
w3af>>> plugins
w3af/plugins>>> help
┌──────────────────┬──────────────────────────────────────────────────────────┐
│ list             │ List available plugins.                                  │
├──────────────────┼──────────────────────────────────────────────────────────┤
│ back             │ Go to the previous menu.                                 │
│ exit             │ Exit w3af.                                               │
├──────────────────┼──────────────────────────────────────────────────────────┤
│ infrastructure   │ View, configure and enable infrastructure plugins        │
│ evasion          │ View, configure and enable evasion plugins               │
│ crawl            │ View, configure and enable crawl plugins                 │
│ output           │ View, configure and enable output plugins                │
│ audit            │ View, configure and enable audit plugins                 │
│ grep             │ View, configure and enable grep plugins                  │
│ mangle           │ View, configure and enable mangle plugins                │
│ auth             │ View, configure and enable auth plugins                  │
│ bruteforce       │ View, configure and enable bruteforce plugins            │
└──────────────────┴──────────────────────────────────────────────────────────┘
w3af/plugins>>>
```

图 5-42　进入 plugins 目录并显示帮助信息

步骤 5：输入 output html_file 命令启用 html_file 格式的输出，然后输入 output 命令查看 output 类插件，将显示如图 5-43 所示的信息。

```
w3af/plugins>>> output html_file
w3af/plugins>>> output
| Plugin name          | Status   | Conf  | Description
| console              | Enabled  | Yes   | Print messages to the console.
| csv_file             |          | Yes   | Export identified vulnerabilities to a CSV file.
| email_report         |          | Yes   | Email report to specified addresses.
| export_requests      |          | Yes   | Export the fuzzable requests found during crawl to
| html_file            | Enabled  | Yes   | Generate HTML report with identified vulnerabiliti
| json_file            |          | Yes   | Export identified vulnerabilities to a JSON file.
| system_log           |          | Yes   | Write log entries to Linux's syslog
| text_file            |          | Yes   | Prints all messages to a text file.
| xml_file             |          | Yes   | Print all messages to a xml file.
w3af/plugins>>>
```

图 5-43　输入 output 命令后显示的信息

步骤 6：输入 output config html_file 命令对 html_file 格式的输出进行配置。接着使用 view 命令列出可利用的选项和值，如图 5-44 所示。

```
w3af/plugins>>> output config html_file
w3af/plugins/output/config:html_file>>> view
| Setting      | Value                                              | Modified | Description      |
| output_file  | /root/report.html                                  |          | File name where  |
|              |                                                    |          | this plugin      |
|              |                                                    |          | will write to    |
| verbose      | False                                              |          | True if debug    |
|              |                                                    |          | information      |
|              |                                                    |          | will be          |
|              |                                                    |          | appended to the  |
|              |                                                    |          | report.          |
| template     | w3af/plugins/output/html_file/templates/complete.html |       | The path to the  |
|              |                                                    |          | HTML template    |
|              |                                                    |          | used to render   |
|              |                                                    |          | the report.      |
w3af/plugins/output/config:html_file>>>
```

图 5-44　使用 view 命令列出可利用的选项和值

步骤 7：使用 set output_file /root/testreport.html 命令，设置输出的扫描报告文件为 testreport.html，如图 5-45 所示。

```
w3af/plugins/output/config:html_file>>> set output_file /root/testreport.html
w3af/plugins/output/config:html_file>>> view
| Setting      | Value                                              | Modified | Description      |
| output_file  | /root/testreport.html                              | Yes      | File name where  |
|              |                                                    |          | this plugin      |
|              |                                                    |          | will write to    |
| verbose      | False                                              |          | True if debug    |
|              |                                                    |          | information      |
|              |                                                    |          | will be          |
|              |                                                    |          | appended to the  |
|              |                                                    |          | report.          |
| template     | w3af/plugins/output/html_file/templates/complete.html |       | The path to the  |
|              |                                                    |          | HTML template    |
|              |                                                    |          | used to render   |
|              |                                                    |          | the report.      |
w3af/plugins/output/config:html_file>>>
```

图 5-45　设置输出的扫描报告文件 testreport.html

步骤 8：使用 back 命令返回 plugins，再使用 crawl 命令查看爬取类插件。同时使用 crawl web_spider 命令开启网页蜘蛛，如图 5-46 所示。

```
| url_fuzzer                  |          | Yes | Try to find backups,
| urllist_txt                 |          |     | Analyze the urllist.
| user_dir                    |          |     | Identify user direct
| web_diff                    |          | Yes | Compare a local dire
| web_spider                  | Enabled  | Yes | Crawl the web applic
| wordnet                     |          | Yes | Use the wordnet lexi
| wordpress_enumerate_users   |          |     | Finds users in a Wor
| wordpress_fingerprint       |          |     | Finds the version of
| wordpress_fullpathdisclosure|          |     | Try to find the path
| wsdl_finder                 |          |     | Find web service def
+-----------------------------+----------+-----+---------------------
w3af/plugins>>> crawl web_spider
w3af/plugins>>>
w3af/plugins>>>
```

图 5-46　开启网页蜘蛛

步骤 9：使用 audit all 命令开启所有审计插件，输入 audit 命令查看，如图 5-47 所示。

```
w3af/plugins>>> audit all
w3af/plugins>>> audit
+-----------------------+----------+------+-----------------------
| Plugin name           | Status   | Conf | Description
+-----------------------+----------+------+-----------------------
| blind_sqli            | Enabled  | Yes  | Identify blind SQL injection vulnera
| buffer_overflow       | Enabled  |      | Find buffer overflow vulnerabilities
| cors_origin           | Enabled  | Yes  | Inspect if application checks that t
|                       |          |      | the value of the remote IP address/H
| csrf                  | Enabled  |      | Identify Cross-Site Request Forgery
| dav                   | Enabled  |      | Verify if the WebDAV module is prope
| deserialization       | Enabled  |      | Identify deserialization vulnerabili
| eval                  | Enabled  | Yes  | Find insecure eval() usage.
| file_upload           | Enabled  | Yes  | Uploads a file and then searches for
| format_string         | Enabled  |      | Find format string vulnerabilities.
| frontpage             | Enabled  |      | Tries to upload a file using frontpa
| generic               | Enabled  | Yes  | Find all kind of bugs without using
| global_redirect       | Enabled  |      | Find scripts that redirect the brows
| htaccess_methods      | Enabled  |      | Find misconfigurations in Apache's "
```

图 5-47　开启所有审计插件

步骤 10：输入 grep all 命令开启所有检索类插件，如图 5-48 所示。

```
w3af/plugins>>> grep all
w3af/plugins>>> grep
+-----------------------+----------+------+-----------------------
| Plugin name           | Status   | Conf | Description
+-----------------------+----------+------+-----------------------
| analyze_cookies       | Enabled  |      | Grep every response for se
| blank_body            | Enabled  |      | Find responses with empty
| cache_control         | Enabled  |      | Grep every page for Pragma
| cdn_providers         | Enabled  |      | Check CDN (Content Deliver
| clamav                | Enabled  | Yes  | Uses ClamAV to identify ma
| click_jacking         | Enabled  |      | Grep every page for missin
| code_disclosure       | Enabled  |      | Grep every page for code d
| content_sniffing      | Enabled  |      | Check if all responses hav
| credit_cards          | Enabled  |      | This plugin detects the oc
| cross_domain_js       | Enabled  | Yes  | Find script tags with src
| csp                   | Enabled  |      | Identifies incorrect or to
| directory_indexing    | Enabled  |      | Grep every response for di
| dom_xss               | Enabled  |      | Grep every page for traces
| dot_net_event_validation | Enabled |    | Grep every page and identi
|                       |          |      | validation.
```

图 5-48　显示漏洞类型

步骤 11：使用 back 命令返回，接着使用 target 命令进入 target 目录，如图 5-49 所示。

步骤 12：启动靶机上的服务器，确认网站可以访问，如图 5-50 所示。

```
w3af>>>
w3af>>> target
w3af/config:target>>> help
┌──────────────────────────────────────────────────────────────────────────────┐
│ view        | List the available options and their values.                   │
│ set         | Set a parameter value.                                         │
│ save        | Save the configured settings.                                  │
├──────────────────────────────────────────────────────────────────────────────┤
│ back        | Go to the previous menu.                                       │
│ exit        | Exit w3af.                                                     │
└──────────────────────────────────────────────────────────────────────────────┘
w3af/config:target>>> view
┌──────────────────────────────────────────────────────────────────────────────┐
│ Setting           | Value   | Modified | Description                         │
├──────────────────────────────────────────────────────────────────────────────┤
│ target_framework  | unknown |          | Target programming framework (unknown/php/asp/asp.net/java/jsp/
│ target            |         |          | A comma separated list of URLs
│ target_os         | unknown |          | Target operating system (unknown/unix/windows)
└──────────────────────────────────────────────────────────────────────────────┘
w3af/config:target>>>
```

图 5-49　使用 target 命令进入 target 目录

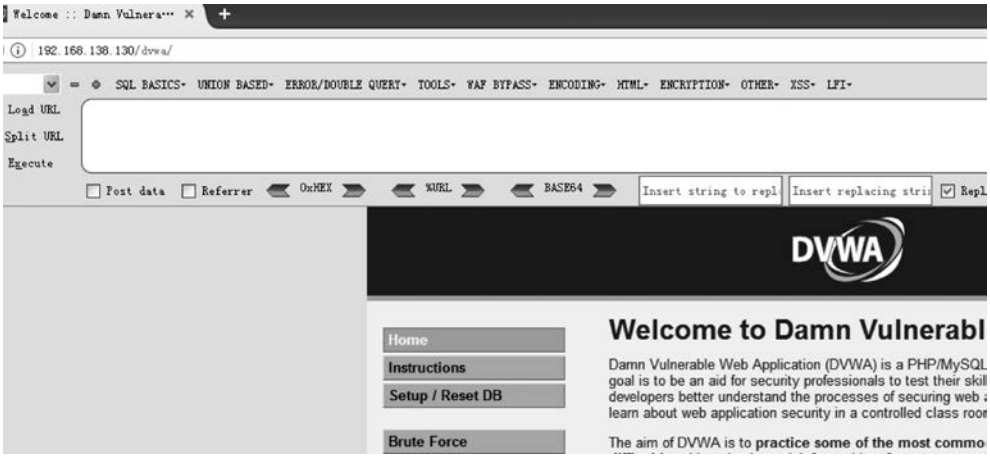


图 5-50　靶机网站服务运行

步骤 13：设置目标地址(target)为 http://192.168.138.130/dvwa/index.php，为下一步扫描进行准备，如图 5-51 所示。

```
w3af/config:target>>>
w3af/config:target>>>
w3af/config:target>>> set target http://192.168.138.130/dvwa/index.php
w3af/config:target>>>
w3af/config:target>>> view
┌──────────────────────────────────────────────────────────────────────────────┐
│ Setting          | Value                                  | Modified | Description
├──────────────────────────────────────────────────────────────────────────────┤
│ target_framework | unknown                                |          | Target programmi
│                  |                                        |          | (unknown/php/asp/
│ target           | http://192.168.138.130/dvwa/index.php  | Yes      | A comma separate
│ target_os        | unknown                                |          | Target operating
└──────────────────────────────────────────────────────────────────────────────┘
w3af/config:target>>>
```

图 5-51　设置目标地址(target)

步骤 14：使用 back 命令返回主目录 w3af >>>，然后使用 start 命令开始扫描，如图 5-52 所示。

步骤 15：扫描结束后，使用 exit 命令退出 w3af。然后切换到 root 目录下，使用

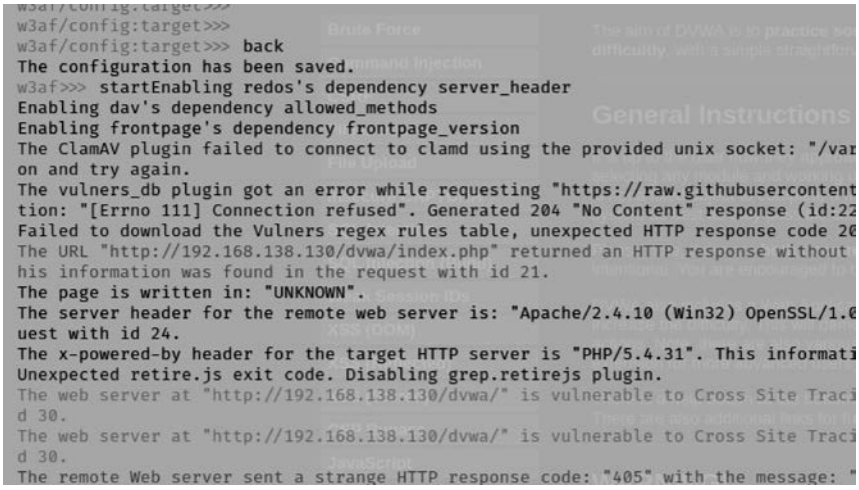


图 5-52　使用 start 命令开始扫描

ls -l testreport.html 命令查看是否生成扫描报告 testreport.html 文件，如图 5-53 所示(存在该文件)。

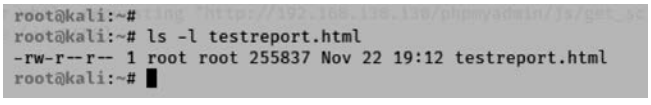


图 5-53　退出扫描并查看创建的文件是否存在

步骤 16：利用浏览器(本实验为 Firefox)打开 testreport.html 文件，其将显示如图 5-54 所示的信息。在该页面中详细记录了前面实验中扫描得到的信息。

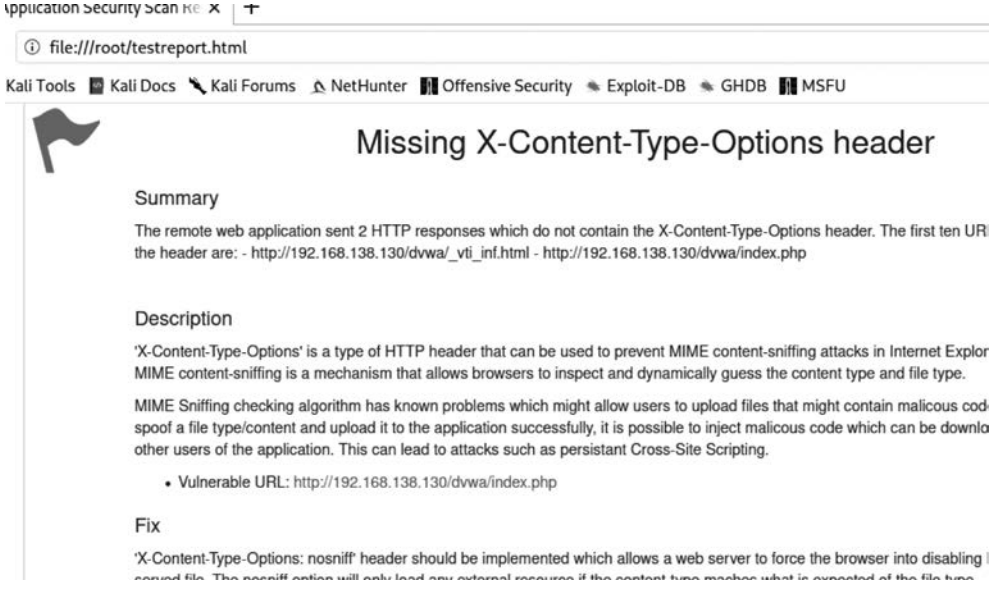


图 5-54　显示 testreport.html 文件的内容

5.5.4 任务与思考

在具体应用中，主要使用漏洞扫描器进行漏洞的扫描和发现。下面介绍漏洞扫描器的组成和主要功能。

1. 安全漏洞数据库

安全漏洞数据库一般与通用漏洞披露目录（common vulnerabilities and exposures，CVE）保持兼容，主要包含安全漏洞的具体信息、漏洞扫描评估的脚本、安全漏洞危害评分（一般采用 CVSS 标准）等信息，新的安全漏洞被公开后，数据库需要及时更新。其中，通用漏洞评价体系（common vulnerability scoring system，CVSS）是一个开放的并且能够被产品厂商免费采用的标准。

2. 扫描引擎模块

作为漏洞扫描器的核心部件，扫描引擎模块可以根据用户在配置控制台上设定的扫描目标和扫描方法，对用来扫描网络的请求数据包进行配置与发送，并将从目标主机接收到的应答包与漏洞数据库中的漏洞特征码进行比对，以判断目标主机上是否存在这些安全漏洞。为了提高效率，扫描引擎模块一般提供了主机扫描、端口扫描、操作系统扫描、网络服务探测等功能，供具体扫描时选用。

3. 用户配置控制台

用户配置控制台是供用户进行扫描设置的操作窗口，需要扫描的目标系统、检测的具体漏洞等信息都可以通过配置控制台设置。

4. 扫描进程控制模块

在针对漏洞的具体扫描过程中，攻击者不仅需要知道扫描结果，许多时候还要实时了解扫描过程中显示的内容，以便通过一些细节获取有价值的信息。扫描进程控制模块提供了这些功能。

扫一扫

视频讲解

5.6 XSS 跨站脚本攻击

5.6.1 预备知识：关于 DVWA

DVWA（damn vulnerable web application）是基于 PHP＋MySQL 的一套用于常规 Web 漏洞教学和检测 Web 脆弱性的程序，可以为安全专业人员测试自己的专业技能和工具提供所需要的环境，帮助 Web 开发者更好地掌握 Web 应用安全防范的过程。

DVWA 提供了以下 10 个功能模块。

（1）Brute Force（暴力破解）。

（2）Command Injection（命令行注入）。

（3）CSRF（跨站请求伪造）。

（4）File Inclusion（文件包含）。

（5）File Upload（文件上传）。

(6) Insecure CAPTCHA(不安全的验证码)。

(7) SQL Injection(SQL 注入)。

(8) SQL Injection(Blind)(SQL 盲注)。

(9) XSS(Reflected)(反射型跨站脚本)。

(10) XSS(Stored)(存储型跨站脚本)。

需要注意的是,DVWA 的代码被分为 4 种安全级别:Low、Medium、High 和 Impossible。初学者可以通过比较 4 种级别的代码,接触一些 PHP 代码审查的内容。

5.6.2 实验目的和条件

1. 实验目的

通过本实验,读者可掌握以下内容。

(1) 了解 XSS 漏洞的攻击原理及相关知识。

(2) 能够进行简单的攻击分析。

2. 实验条件

由于 DVWA 环境是基于 PHP/MySQL 的,因此需要先安装 DVWA 环境。为了便于操作,建议直接使用 XAMPP 集成软件来搭建。本实验在 Windows 环境下安装,并使用与 XAMPP 集成的 DVWA,具体操作步骤如下。

步骤 1:安装 XAMPP。从 http://www.xampps.com/官网下载和安装,只需注意选择 Windows 环境,其他的按照系统提示进行即可。

步骤 2:下载 DVWA 压缩包。从 http://www.dvwa.co.uk/官网下载 DVWA 压缩包,并将压缩包解压到 dvwa,再将其复制到 XAMPP 安装目录下的\xampp\htdocs 目录。

步骤 3:通过 XAMPP 的控制台启动 XAMPP 的 Apache 和 MySQL 服务,如图 5-55 所示。

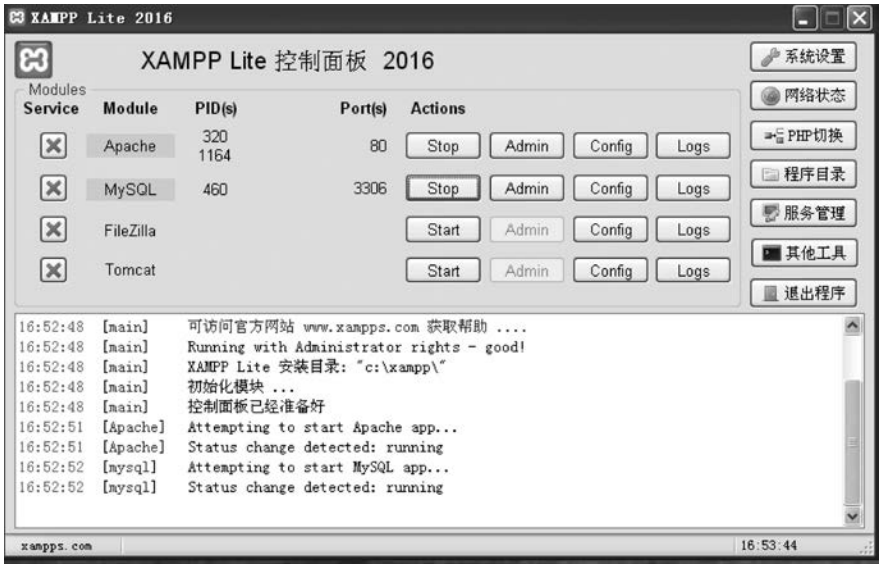


图 5-55 通过 XAMPP 的控制台启动 XAMPP 的 Apache 和 MySQL 服务

步骤 4：修改\xampp\htdocs\dvwa\config 下的 config.php 配置文件，在该配置文件中包含了连接 MySQL 数据库的密码（XAMPP 集成环境下面 MySQL 的默认登录账号为 root，密码为空）。

步骤 5：在浏览器中输入 http://127.0.0.1/DVWA/setup.php，就可以访问 DVWA 的配置页面，如图 5-56 所示。单击该页面中的 Create/Reset Database 按钮，直接建立 DVWA 的数据库。

图 5-56　访问 DVWA 配置页面并通过 Create/Reset Database 建立数据库

另外，在 XAMPP 环境下，也可以通过如图 5-57 所示的操作界面来创建 DVWA 数据库。只有 Setup Check 全部显示为绿色，而没有出现红色时，才能表示完全安装成功。

图 5-57　在 XAMPP 环境下创建 DVWA 数据库

步骤 6：创建好 DVWA 数据库后，系统将自动跳转到 DVWA 的登录首页，如图 5-58 所示，系统默认的登录账号名称和密码为 admin/password。

图 5-58　DVWA 登录页面

5.6.3　实验过程

步骤 1：进入实验场景，依次选择“开始”→“所有程序”→XAMPP→XAMPP Control Panel 选项打开 XAMPP 控制台，在如图 5-55 所示的界面中开启 Apache HTTP 服务和 MySQL 服务。

步骤 2：打开 DVWA 网站。在浏览器中输入 http://127.0.0.1/dvwa，正确输入账号名称和密码(系统默认为 admin/password)后登录，如图 5-59 所示。

图 5-59　DVWA 成功登录后的主页面

步骤 3：选择 XSS(Reflected)选项后，打开存在漏洞的网站(本实验为 http://127.0.0.1/dvwa/vulnerabilities/xss_r/)，将该 URL 复制到浏览器的地址栏，进入如图 5-60 所示的界面。

步骤 4：选择 DVWA Security 的安全级别，此处选择 low(低)选项，如图 5-61 所示。

图 5-60　存在反射型 XSS 漏洞页面

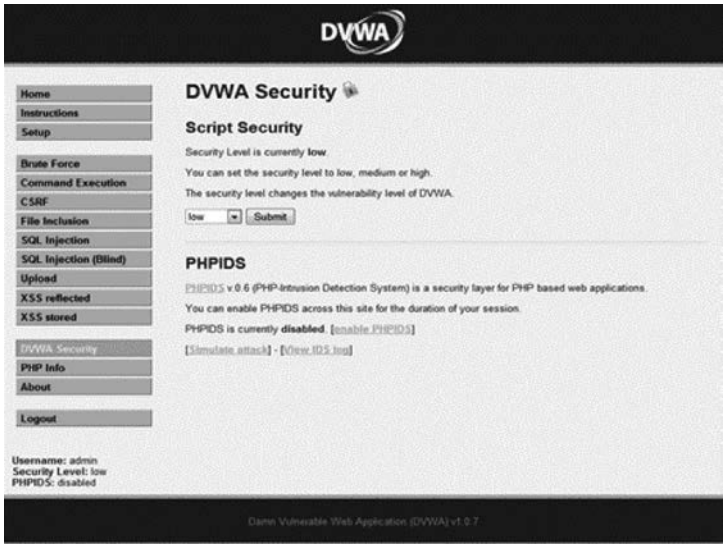


图 5-61　DVWA Security 的安全级别

步骤 5：查看正常输入输出。在输入框中输入 test，单击 Submit 按钮，可以看到页面上的正常返回结果，如图 5-62 所示，说明这个页面的功能是将用户输入的信息直接发送给用户。

步骤 6：查看 PHP 源码。单击右下角的 View Source 按钮可以看到页面的 PHP 源码，如图 5-63 所示。从源码中可以看出，页面直接将用户输入的信息返回给用户。

步骤 7：进行攻击测试。在输入框中输入< script > alert(/XSS/)</script >，可以看到非正常返回结果页面，如图 5-64 所示。同时在地址栏中可以看到输入内容的 URL 编码，如图 5-65 所示。由此说明，Web 应用将未经验证的数据通过请求发送给客户端。

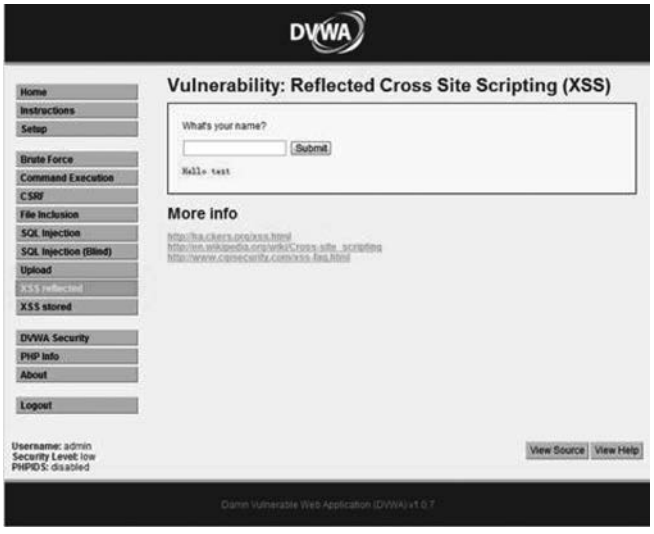


图 5-62　页面正常返回效果

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET
['name'] == NULL || $_GET['name'] == ''){

 $isempty = true;

} else {

 echo '<pre>';
 echo 'Hello ' . $_GET['name'];
 echo '</pre>';

}

?>
```

图 5-63　页面 PHP 源码

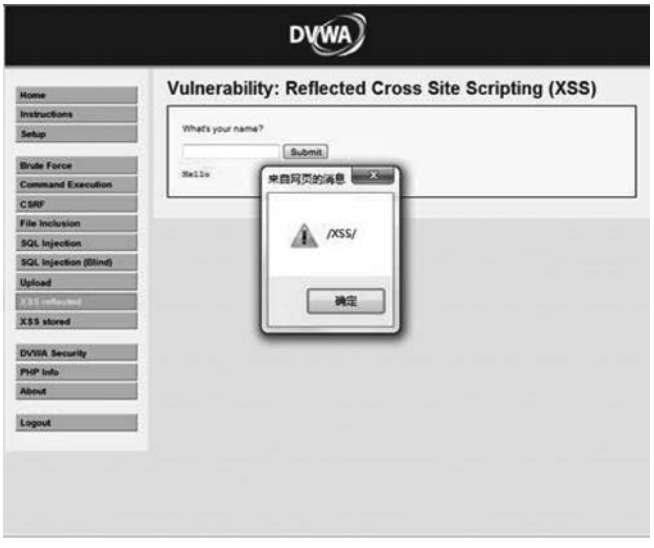


图 5-64　页面攻击效果

图 5-65　被攻击页面此时的地址栏内容

步骤 8：验证此类漏洞的非持久性。重新访问 DVWA 页面，单击刷新按钮或再次选择左侧 XSS(Reflected)选项，可以看到页面恢复正常，由此说明之前输入的信息未保存，是非持久性跨站脚本漏洞。

步骤 9：存储型 XSS 攻击。打开存在漏洞的网站 http://localhost/dvwa/vulnerabilities/xss_s/，在 DVWA 页面中选择左侧的 XSS(Stored)选项，出现如图 5-66 所示的界面。

图 5-66　存在存储型 XSS 漏洞的页面

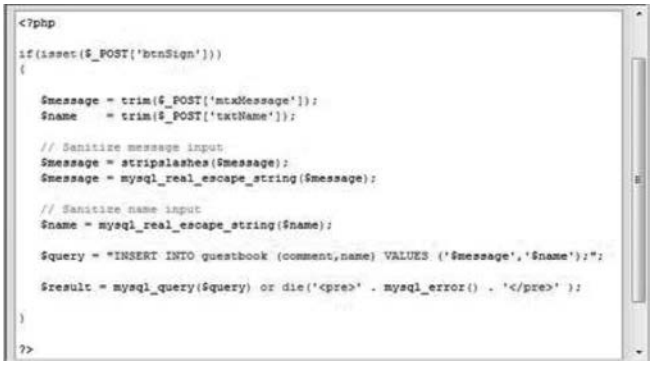
步骤 10：查看正常输入输出。在 Name 输入框中输入 test，在 Message 输入框中输入 This is a test comment。单击 Sign Guestbook 按钮，可以得到正常的返回结果，如图 5-67 所示，说明该网页是为用户发表署名和评论的。

步骤 11：查看 PHP 源码。单击右下角的 View Source 按钮，可以看到页面的 PHP 源码，如图 5-68 所示。从源码中可以看出，页面允许用户存储未正确过滤的信息。

步骤 12：进行攻击测试。在 Name 输入框中输入 Test，在 Message 输入框中输入 <script>alert(/XSS/)</script>，单击 Sign Guestbook 按钮，再次访问页面就可以看到如图 5-69 所示的对话框。

步骤 13：验证漏洞的存储性。重新访问 DVWA 页面，单击刷新按钮或再次选择左侧的 XSS(Stored)选项，可以看到页面仍然为如图 5-69 所示之前的对话框，说明之前输入的信息已被保存。

图 5-67　正常返回效果



```
<?php

if(isset($_POST['btnSign']))
{

   $message = trim($_POST['mtxMessage']);
   $name    = trim($_POST['txtName']);

   // Sanitize message input
   $message = stripslashes($message);
   $message = mysql_real_escape_string($message);

   // Sanitize name input
   $name = mysql_real_escape_string($name);

   $query = "INSERT INTO questbook (comment,name) VALUES ('$message','$name');";

   $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>' );

}

?>
```

图 5-68　页面 PHP 源码

图 5-69　被攻击的页面

5.6.4 任务与思考

可通过以下方法来防范 XSS 攻击。

1. XSS 过滤

虽然 XSS 攻击的对象是客户端，但 XSS 的本质是 Web 应用服务的漏洞，所以必须同时对 Web 服务器和客户端进行安全加固才能避免攻击的发生。XSS 过滤需要在客户端和服务器端同时进行。

2. 输入验证

输入验证就是对用户提交的信息进行有效性验证，仅接受有效的信息，阻止或忽略无效的用户输入信息。在对用户提交的信息进行有效性验证时，不仅要验证数据的类型，还要验证其格式、长度、范围和内容。

3. 输出编码

由于大多数 Web 应用程序都会把用户输入的信息完整地输出到页面中，从而导致 XSS 漏洞的存在。为解决这一问题，当需要将一个字符串输出到 Web 网页，但又无法确定这个字符串是否包含 XSS 特殊字符时，为了确保输出内容的完整性和正确性，可以使用 HTML 编码（HTML encode）进行处理。

扫一扫

视频讲解

5.7 针对 MS SQL 的提权操作

5.7.1 预备知识：MS SQL 提权

在很多时候，当攻击者入侵一个系统后，需要得到的是这个系统的管理员权限。但是，一般情况下获取到的往往是普通用户账户信息，拥有的权限相对较小。这时就必须采取提权方式，将普通用户的权限提升到管理员的权限。提权是指操作者提高自己在系统中的操作权限，主要用于网站入侵过程，在攻击者入侵某一网站时，往往需要通过各种漏洞提升 Web Shell 权限以夺得该服务器的控制权。

MS SQL 是指微软的 SQL Server 数据库服务器，它是一个数据库平台，提供从服务器到终端的完整数据库解决方案，其中的数据库服务器部分是一个数据库管理系统，用于建立、使用和维护数据库。

MS SQL 提权是专门针对 MS SQL 数据库用户账户管理权限的一种攻击方式，通过提升普通用户账户的权限，获取对 MS SQL 数据库系统的管制权限。

5.7.2 实验目的和条件

1. 实验目的

在熟悉系统提权攻击基本方法的基础上，以 MS SQL 数据库系统为操作对象，掌握针对 MS SQL 提权的实现方法。

2. 实验条件。

本实验中使用的软硬件清单如表 5-3 所示。

表 5-3 MS SQL 提权实验清单

类　　型	序　　号	软硬件要求
攻击机	1	数量：1 台
	2	操作系统版本：Windows XP 以上
	3	软件版本：X-Scan 扫描器，SQL Tools 工具，SQL 查询分析器
靶机	1	数量：1 台
	2	操作系统版本：Windows Server 2003
	3	软件版本：无

5.7.3 实验过程

步骤 1：进入实验环境，分别运行攻击机 Windows XP 和靶机 Windows Server 2003 的操作系统。

步骤 2：查看靶机的 IP 地址并确认 MS SQL 服务已经正常启动，如图 5-70 和图 5-71 所示。

图 5-70 使用 ipconfig 命令查看靶机的 IP 地址

图 5-71 查看 SQL Server 是否已经正常启动

步骤 3：在攻击机上运行 X-Scan 扫描器并进行设置，如图 5-72 所示。

图 5-72　设置 X-Scan 扫描器

步骤 4：扫描的目标 IP 地址设置为靶机的 IP 地址(参照图 5-70 中显示的 IP 地址)，如图 5-73 所示。根据需要，在确定攻击对象的 IP 地址范围但是无法确定具体 IP 地址的前提下，可以在“指定 IP 范围”文本框中输入需要扫描的 IP 地址或地址段。

图 5-73　输入扫描的 IP 地址或地址段

步骤 5：选择了扫描参数中的“扫描模块”选项后，在中间的列表框内选中“SQL-Server

弱口令”复选框，如图 5-74 所示。

图 5-74　设置扫描模块

步骤 6：在选择了“其他设置”选项后，在打开的如图 5-75 所示的对话框中可以根据需要选择相应的功能项。如选中“显示详细进度”复选框后可以实时查看扫描过程的进展情况。

图 5-75　设置“其他设置”功能项

步骤 7：选择“插件设置”→“字典文件设置”选项，在打开的如图 5-76 所示的对话框中设置扫描过程中需要使用的字典(也可以使用默认字典)。

步骤 8：单击“确定”按钮开始进行扫描，扫描结果如图 5-77 所示，得到了 MS SQL 数据库使用的弱口令(sa/123456)。

步骤 9：使用第三方工具 SQL Tools 连接到 MS SQL 数据库(也可以使用 MS SQL 自身提供的 SQL-Server 工具进行连接)，如图 5-78 所示。

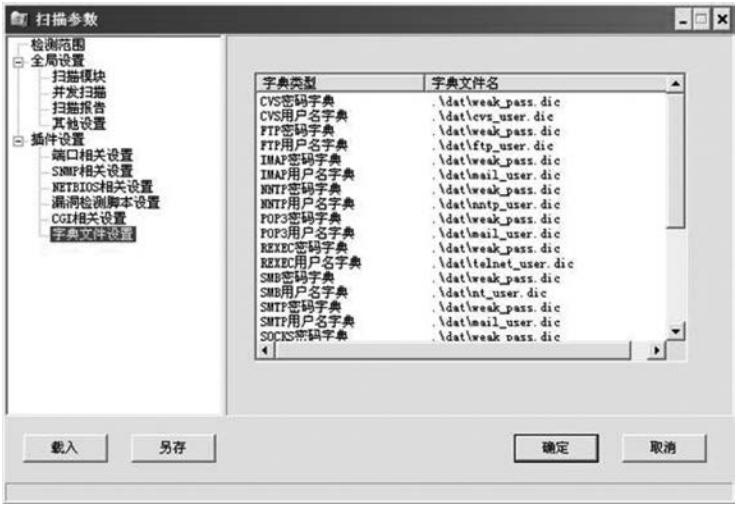


图 5-76　设置扫描过程中使用的字典

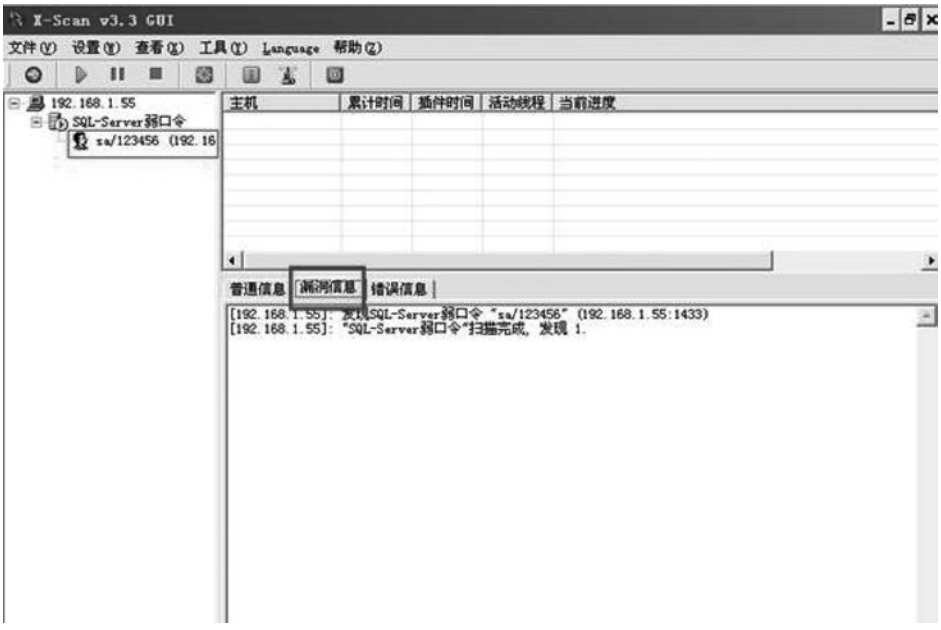


图 5-77　显示扫描结果

图 5-78　使用 SQL Tools 工具连接到 MS SQL 数据库

步骤 10：连接成功后进入如图 5-79 所示的操作界面。

图 5-79　连接成功后的操作界面

步骤 11：选择“利用目录”→“执行 DOS 命令”选项，在如图 5-80 所示的文本框中输入要执行的命令，如输入 whoami 命令查看在线的用户。

图 5-80　执行 DOS 命令界面

步骤 12：使用查询分析器连接 MS SQL 数据库，如图 5-81 所示。

步骤 13：在查询分析器中执行如下代码（执行过程和结果如图 5-82 所示。

```
;EXEC sp_configure 'show advanced options', 1 --
;RECONFIGURE WITH OVERRIDE --
;EXEC sp_configure 'xp_cmdshell', 1 --
;RECONFIGURE WITH OVERRIDE --
;EXEC sp_configure 'show advanced options', 0 --
```

图 5-81　使用查询分析器连接 MS SQL 数据库

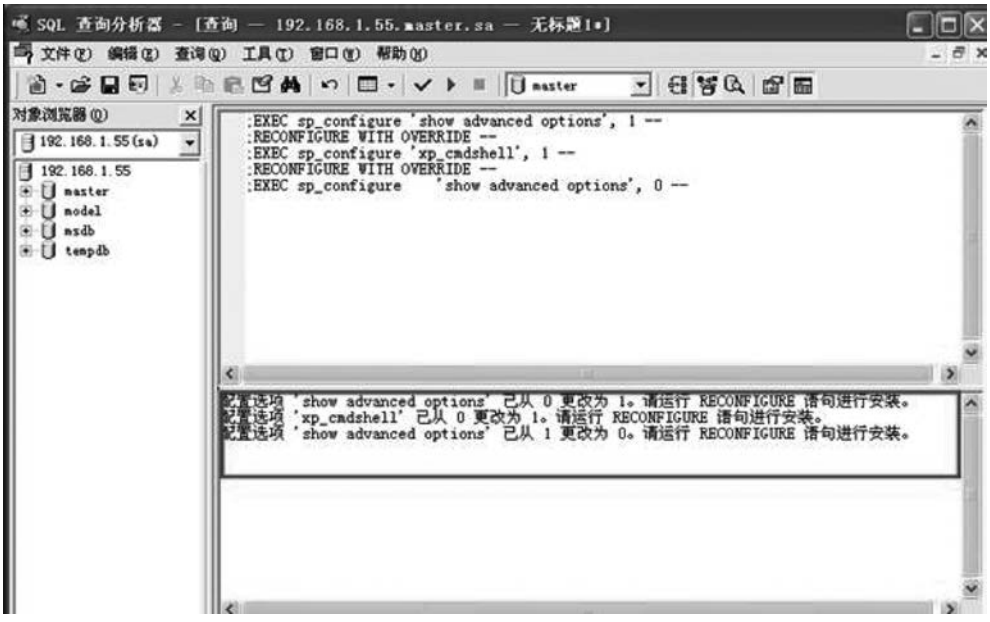


图 5-82　在查询分析器中执行相关代码

步骤 14：再次使用 SQL Tools 执行 whoami 命令时，如图 5-83 所示，从返回的信息可以看出，已经获得了最高权限(system)，权限提升过程结束。

5.7.4　任务与思考

本实验介绍了针对 MS SQL 数据库系统的提权实现方法，通过实验读者会发现提权攻

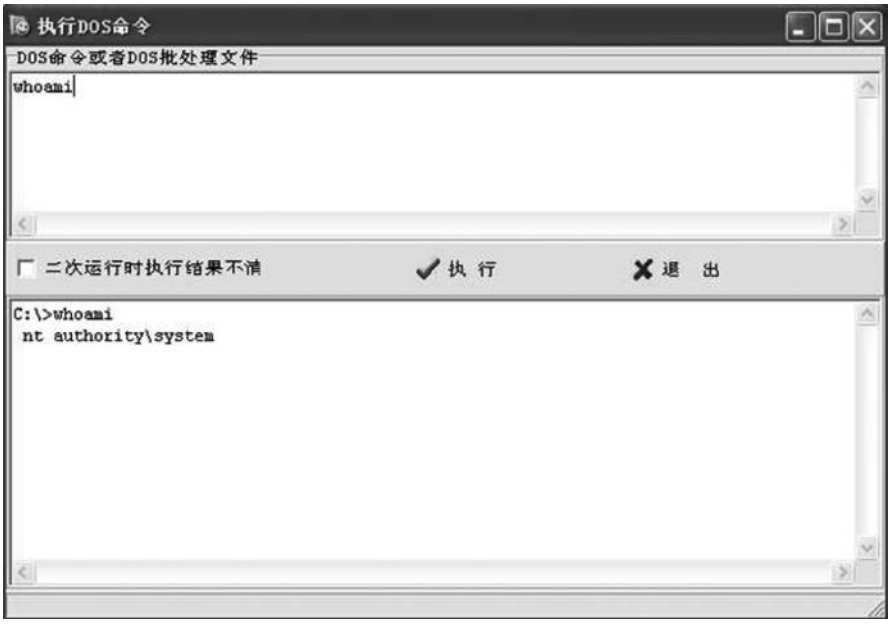


图 5-83 已经实现提权操作

击存在的危害性。通过提权，普通用户将会拥有管理员的权限。在拥有了管理员权限后，攻击者可以像控制本地计算机一样来操控被攻击对象。

在大多数情况下，MS SQL 服务器将被安装在一个混合模式下，它的默认用户是 sa，很多时候默认用户只会设置一个简单的密码，这意味着攻击者容易使用字典文件进行暴力破解得到密码。为此，针对 MS SQL 提权攻击，最简单和有效的办法还是为系统管理员账户设置复杂的密码并定期或不定期更换，同时提供完整的系统日志，并及时对日志记录进行分析，当发现攻击迹象时尽快找到攻击源，并进行必要的封堵。另外，还需要及时为 MS SQL 数据库系统安装补丁程序。