第5章 网络扫描

网络扫描是一项基于网络远程服务发现系统脆弱点的关键技术，也是构建网络攻击的重要一环。攻击者往往需要采用主动或被动的网络扫描方式来建立对网络的了解，如通过扫描攻击目标IP地址网段的多台主机，获悉主机开放端口、操作系统等关键信息；利用路由追踪工具了解数据包访问目标地址所采用的路径信息；采取模拟攻击的形式逐项检查目标潜在的安全漏洞等。网络扫描为后续的攻击步骤打下基础。

本章关注路由追踪技术和网络扫描技术，就其实现原理进行图例结合的细致讲解，引入常用的路由追踪工具traceroute与网络扫描工具Nmap设置紧扣理论知识的实验任务，就网络侦测的话题开展深入讨论。

5.1 路由追踪

5.1.1 实验目的

了解路由追踪技术的具体原理与应用场景，掌握其分别基于UDP数据包与ICMP数据包实现的理论知识依据，通过实验练习，熟悉路由追踪工具traceroute的具体使用方法。

5.1.2 实验内容

熟悉路由追踪的工作机理，掌握traceroute工具的参数意义与具体使用方法。在搭建的网络拓扑中进行路由追踪实践，结合Wireshark工具分析、验证UDP机制与ICMP机制的实现原理。

5.1.3 实验原理

路由追踪工具可以用于获取一个网络数据包从源地址到目的地址的路径信息。在类UNIX系统（如Linux、macOS等）中，实现路由追踪的命令为traceroute；在Windows系统中，命令为tracert，它们的功能相同。路由追踪功能依赖于发出数据包的存活时间（Time to Live，TTL）字段，路由器在路由时会将数据包的TTL值减1，丢弃TTL值变为0的数据包，并返回ICMP超时错误消息。

因此，路由追踪工具将发送TTL值逐渐增加的数据包，初始值为1。同时，同样TTL值的包会发送3次。对于TTL值为1的数据包，第1台路由器将其TTL值减小为0，因此丢弃该数据包，并回送ICMP超时消息给源地址。对于TTL值为2的数据包，第1台路由器将其TTL值减小为1，并转发该数据包给第2台路由器；第2台路由器将其TTL值减小为0，因此丢弃该数据包，并回送ICMP超时消息。之后的数据包同样以上述方式被转发、

处理。路由追踪工具使用返回的 ICMP 超时消息来构建数据包遍历的路由器列表，直至到达目的地址。基于路由追踪工具实现方式的不同，数据包在到达目的地址后，返回不同类型的 ICMP 信息。

traceroute 命令默认基于 UDP 实现。源地址发送的数据包是通过 UDP 协议来传输的，使用一个大于 30000 的端口号。目的地址在收到这个数据包的时候会返回一个端口不可达的 ICMP 错误信息。源地址通过判断收到的错误信息是 TTL 超时还是端口不可达来判断数据包是否到达目的地址。基于 UDP 实现的路由追踪过程如图 5-1 所示。

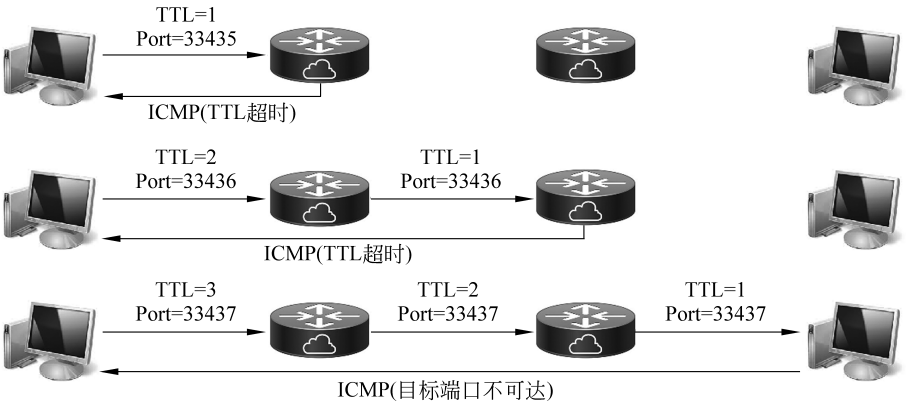

图 5-1　基于 UDP 实现的路由追踪

然而，目前大多数主机都关闭了 UDP 服务，因此这些主机会将 UDP 数据包丢弃而不返回任何信息。为了解决该问题，可以使用基于 ICMP 的实现。tracert 命令使用这种实现方式，也可以使用 traceroute 命令添加特定参数指定这种实现方式。在这种实现方式中，源地址发送的 ICMP 回显请求(echo request)数据包，目的地址在收到回显请求的时候会向源地址发送一个 ICMP 回显应答(echo reply)。同样地，源地址通过判断收到的信息是 TTL 超时错误还是 ICMP 回显应答来判断数据包是否到达目的地址。基于 ICMP 实现的路由追踪过程如图 5-2 所示。

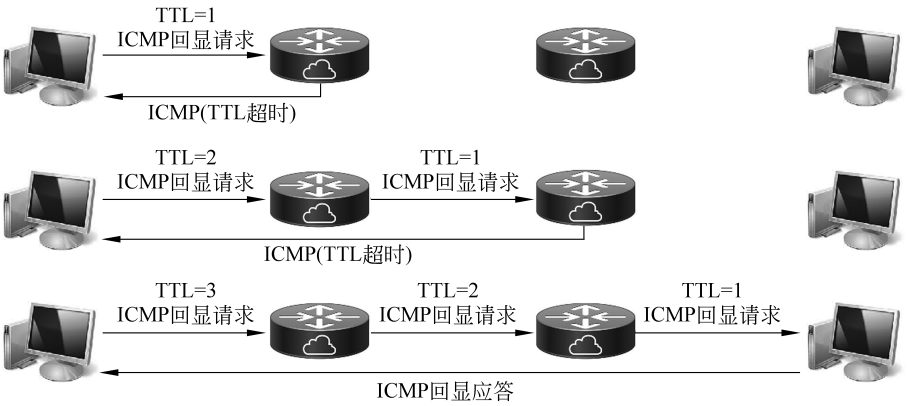

图 5-2　基于 ICMP 实现的路由追踪

5.1.4 实验步骤

本次实验在配置好的虚拟环境中操作。在开机前,为虚拟机配置两块网卡,一个是NAT网络模式,另一个是内部网络模式。开机后,先安装实验所需工具:

```
$sudo apt-get install traceroute nmap
// 安装 zenmap 必需的 Python GTK
$wget http://archive.ubuntu.com/ubuntu/pool/universe/\
      p/pygtk/python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
$sudo apt install ./python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
// 安装 zenmap
$wget http://archive.ubuntu.com/ubuntu/pool/universe/\
      n/nmap/zenmap_7.60-1ubuntu5_all.deb
$sudo apt install ./zenmap_7.60-1ubuntu5_all.deb
```

其中,traceroute是路由追踪工具,nmap与zenmap是网络扫描工具。

由于实验用到的网络扫描工具功能强大,为了不影响宿主机或者外网主机,需要在Ubuntu桌面右上角的网络连接中,对于NAT网络模式网卡建立的网络连接,单击“**Turn off**”按钮,取消该连接,并配置另一个内部网络模式网卡的IP,该网卡IP地址为192.168.1.56,掩码为255.255.255.0,网关为192.168.1.1。

本实验已经提前准备好4台虚拟服务器供读者完成实验。考虑到读者的主机资源各不相同,可能出现资源有限等问题,提供的虚拟服务器是Server版本,即只有命令行、没有图形界面。

导入的4台虚拟服务器分别为lab5-server1/2/3/4。登录这些虚拟机的账户名与口令如表5-1所示。

表5-1 虚拟服务器登录账户与口令

虚拟服务器	账　户	口　令
lab5-server1	server1	1server
lab5-server2	server2	2server
lab5-server3	server3	3server
lab5-server4	server4	4server

对于不同的虚拟服务器,安装对应服务,运行命令如下:

```
对于服务器 1:
// 安装必备的网络工具
$sudo apt-get install net-tools ifupdown

对于服务器 2:
// 安装 SSH 服务和必备的网络工具
$sudo apt-get install ssh net-tools ifupdown

对于服务器 3:
```

```
// 安装 Apache 服务和必备的网络工具
$sudo apt-get install apache2 net-tools ifupdown

对于服务器 4:
// 安装 Telnet 服务和必备的网络工具
$sudo apt-get install telnetd net-tools ifupdown
```

同样地，安装完所有服务后，停止所有虚拟服务器的 NAT 网卡连接，其操作如下：

```
对于所有虚拟服务器:
$ifconfig
// 发现只有本地回路和 NAT 网卡，NAT 网卡名记为[interface_nat]
$sudo ifconfig [interface_nat] down
```

为不同虚拟服务器的内部网络连接模式网卡配置 IP 地址，其详细配置信息如表 5-2 所示。

表 5-2　虚拟服务器各网卡 IP 地址配置

虚拟服务器	[ip_addr]	[netmask]	[gw_addr]
lab5-server1	192.168.1.1	255.255.255.0	不设置
	192.168.2.1	255.255.255.0	不设置
	192.168.3.1	255.255.255.0	不设置
lab5-server2	192.168.2.56	255.255.255.0	192.168.2.1
lab5-server3	192.168.3.57	255.255.255.0	192.168.3.1
lab5-server4	192.168.3.58	255.255.255.0	192.168.3.1

由于没有图形界面，当修改各网卡的配置文件时，可使用 nano 或者 pico 命令编辑文件。两个命令的使用方法请参阅参考文献[1]和[2]，下文也会给出具体用例。以下介绍修改网卡 IP 地址的方法，具体实践操作流程可参考本书附带的微课视频。


如何配置网卡

```
对于所有虚拟服务器:
// 显示所有网卡
$ifconfig -a
// 对于没有 IP 地址的网卡，其网卡名下文以[interface_int]指代
// 修改这些网卡的配置，添加静态 IP 地址
$sudo nano /etc/network/interfaces
// 在文件末尾加入:
   auto [interface_int]                    // 指定使用的网络接口
   iface [interface_int] inet static       // 该接口使用静态 IP 设置
   address [ip_addr]                       // 设置 IP 地址
   netmask [netmask]                       // 设置子网掩码
   gateway [gw_addr]                       // 设置网关地址，服务器 1 不需要设置该项
// 按 Ctrl+O 键保存文件，nano 将提示选择一个文件名，按 Enter 键使用默认文件名，保存好之
// 后，按 Ctrl＋X 键退出编辑
```

```
// 刷新配置
$sudo ip addr flush dev [interface_int]
// 启动网卡
$sudo ifup [interface_int]
// 查看网络连接服务状态
$sudo systemctl status networking.service
```

在服务器 1 上打开 IPv4 路由转发：

```
对于服务器 1:
$sudo nano /etc/sysctl.conf
 net.ipv4.ip_forward=1
$sudo sysctl -p
```

同时检查服务器 2/3/4 的路由表,确定默认路由的网关为表 5-2 所示。

```
对于服务器 2/3/4:
$route
```

以服务器 2 为例,其余服务器以此类推。首先,通过在服务器 2 上执行 route 命令,获悉具体的路由表,具体内容如图 5-3 所示。

```
server2@lab5-server2:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    0      0        0 enp0s8
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
```

图 5-3　服务器 2 路由表

接着,对于每个服务器,配置不同的服务以供网络扫描：

```
对于服务器 2:
// 修改 SSH 连接端口为 23,开启 SSH 服务
$sudo nano /etc/ssh/sshd_config
Port 23
$sudo service sshd restart
// 检查 SSH 服务状态
$sudo service sshd status
// 利用防火墙关闭所有端口的 UDP 连接
$sudo ufw enable
$sudo ufw deny proto udp to any
// 检查防火墙规则
$sudo ufw status

对于服务器 3:
// 开启 HTTP 服务
$sudo service apache2 restart
// 检查 HTTP 服务状态
$sudo service apache2 status
```

对于服务器 4：

```
// Telnet 服务默认开启，检查其是否开启
$netstat -a | grep telnet
```

服务器 2 的防火墙规则如图 5-4 所示。

```
server2@nmap-server2:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Anywhere/udp               DENY        Anywhere/udp
Anywhere/udp (v6)          DENY        Anywhere/udp (v6)
```

图 5-4　服务器 2 防火墙规则

通过以上的配置，实验环境已经搭建完成，请在虚拟机上利用 ping 命令检查与各个虚拟服务器的连接。

任务 5.1　在虚拟机上打开 Wireshark，指定**内部网络网卡**抓包，运行以下命令：

```
$traceroute 192.168.3.58
```

(1) 观察并截图记录命令行输出结果和 Wireshark 截获的数据包。

(2) 简述 traceroute 命令是如何实现路由追踪的。要求：结合原理，具体分析数据包的内容，指明哪些字段在路由追踪过程中起怎样的作用。

注：在 Wireshark 中选中一个数据包，如果这个数据包是在一个会话内，Wireshark 会在界面上标注其他包，如图 5-5 所示，序号为 3 和序号为 10 的包在一个会话内，结合具体包内容，可以说 10 号包是 3 号包的回复报文。另外，还可以使用 Conversation Filter 功能在界面上只显示该会话的所有数据包。

3	1.571999171	192.168.1.56	192.168.3.58	UDP	74 33948 → 33434 …
4	1.572026576	192.168.1.56	192.168.3.58	UDP	74 46818 → 33435 …
5	1.572038264	192.168.1.56	192.168.3.58	UDP	74 59474 → 33436 …
6	1.572076774	192.168.1.56	192.168.3.58	UDP	74 57819 → 33437 …
7	1.572090564	192.168.1.56	192.168.3.58	UDP	74 35215 → 33438 …
8	1.572135354	192.168.1.56	192.168.3.58	UDP	74 34747 → 33439 …
9	1.572149696	192.168.1.56	192.168.3.58	UDP	74 54840 → 33440 …
10	1.572186757	192.168.1.1	192.168.1.56	ICMP	102 Time-to-live e…

图 5-5　Wireshark 判断会话内的包

任务 5.2　在虚拟机上，Wireshark 重新开始抓包，用同样的命令对 192.168.2.56 进行路由追踪。

(1) 观察并截图记录命令行输出结果和 Wireshark 截获的数据包结果。

(2) 解释出现该现象的原因。

任务 5.3　对于任务 2 出现的问题，traceroute 提供了另一种基于 ICMP 报文的实现方式。请使用 man 命令，或者查阅参考文献[3]，查找使用 ICMP 实现的参数，完成 192.168.2.56 的路由追踪，并利用 Wireshark 抓包。

(1) 观察并截图记录命令行输出结果和 Wireshark 截获的数据包。

(2) 简述此时 traceroute 命令是如何实现路由追踪的。要求：结合原理，具体分析数据包的内容，指明哪些字段在路由追踪过程中起怎样的作用。

5.2 网络扫描

5.2.1 实验目的

了解网络扫描的具体含义,熟悉其在主机扫描、端口扫描、系统扫描、服务扫描等不同应用场景中的实施目的与实现机理。掌握网络扫描工具 Nmap 的使用方法及其不同参数的含义,熟悉网络扫描图形化工具 Zenmap 的操作面板与基本功能。

5.2.2 实验内容

掌握主机扫描、端口扫描、系统扫描与服务扫描的实现机理与具体使用场景;掌握网络扫描工具 Nmap 的使用方法,了解其不同参数的含义并对输出结果进行解释;熟悉网络扫描图形化工具 Zenmap 的使用方法,了解其在真实拓扑下的具体应用。

5.2.3 实验原理

1. 主机扫描

主机扫描是网络扫描的基础。攻击者可以利用主机扫描获得网络中活动的主机,然后以它们为目标进行后续的攻击。

主机扫描的思想是向目标主机发送特定数据包,若目标主机有回应,则认为该主机是活动的。常见的主机扫描技术包括 ping 扫描以及对局域网内的 ARP 扫描等。

最简单的 ping 扫描的方式是将 ICMP 回显请求(echo request)数据包发送至多个主机。当主机接收到 ICMP 回显请求后将予以响应,而扫描方收到响应则代表对应的主机为活动状态。然而,接收不到 ICMP ping 回复报文并不能充分地说明对应的主机不处于活动状态。这是因为随着人们安全意识的不断增强,ICMP ping 报文正在被越来越多的防火墙过滤以避免来自外部的对站点或主机的嗅探。

除了 ICMP 报文,TCP SYN 报文也可被用来进行主机发现与扫描。在 TCP 建立连接的三步握手中,扫描方向远程主机发送带有 SYN 标志的 TCP 包以请求建立新的连接。远程主机在收到该报文后返回带有 SYN 和 ACK 标志的数据包进行响应,以表示其收到了扫描方发来的数据,并准备继续执行创建新连接的后续步骤。扫描方收到该响应后即可确认远程主机处于活动状态,随后便可以发送带有 RST 标志的 TCP 报文中断连接过程。类似地,基于 TCP ACK 报文的 ping 扫描向远程主机发送 ACK 报文,通过接收返回的带有 RST 标志的数据包来发现网络中的主机。

地址解析协议(Address Resolution Protocol, ARP)是根据 IP 地址获取物理地址的一个 TCP/IP。ARP 扫描是扫描方利用 ARP 能够获悉给定 IP 地址的远程主机 MAC 地址的特性,进一步判断网段内部活跃主机的攻击方式。攻击者在该扫描初始阶段向局域网内的所有主机广播包含目标 IP 地址的 ARP 请求报文,如果该 IP 地址主机成功接收该报文且为活动状态,即会响应并发回自身 MAC 地址给发送端。相同的方法可以用于检测整个网段的活动主机。然而由于 ARP 只能用于同一物理网段(即同一局域网)下,因此 ARP 扫描也

只适用于局域网内的主机扫描。

2. 端口扫描

经过主机扫描得到活动主机后，需要对目标主机进行端口扫描，获取其开放端口信息。

TCP Connect 扫描是最简单的端口扫描技术，通过实现 TCP 三次握手发现活动的端口，但是该扫描方式不隐蔽，大量的连接请求会被记录到系统服务日志中，很容易被入侵检测系统发现，或者被防火墙屏蔽。因此，端口扫描技术还有 TCP SYN 扫描、TCP ACK 扫描、TCP FIN 扫描、X-mas 扫描、Null 扫描等。

TCP SYN 扫描是常用的扫描选项。它执行速度快，在一个没有入侵防火墙的快速网络上，每秒可以扫描数千个端口。TCP SYN 扫描程序只发送三次握手的第一次 SYN 报文段，若其收到 SYN-ACK 数据包响应则表示目标端口开放；反之，若收到 RST 数据包则目标端口关闭。若目标端口确认开放，扫描程序以 RST 数据包作为响应在握手完成之前关闭连接。其工作原理如图 5-6 所示。

(a) 目标端口开放　　(b) 目标端口关闭

图 5-6　TCP SYN 扫描工作原理

TCP ACK 扫描不能确定端口的开放状态，因为目标端口若未配置防火墙规则对 ACK 数据包进行过滤，无论其状态是开放还是关闭都将返回 RST 数据包作为响应。然而，TCP ACK 扫描可以通过观测该端口的连接是否被过滤来探测防火墙规则与配置：若返回 RST 数据包则 ACK 数据包未被过滤，目标端口不存在防火墙配置；反之，若不响应或收到特定的 ICMP 错误消息，则目标端口存在防火墙过滤配置，其工作原理如图 5-7 所示。

(a) 不存在防火墙规则　　(b) 存在防火墙规则

图 5-7　TCP ACK 扫描工作原理

由于上述扫描机制在扫描过程中并未建立完整的 TCP 连接，也被称为半开放扫描(half-open scanning)。正因如此，属于该类别的扫描机制都具备速度快，不容易被防火墙记录进日志的优势。

3. 系统扫描

得到了目标主机的开放端口后，需要确定其使用的操作系统，这决定了攻击者后续采取的攻击方式。例如，对于 Windows 系统的主机发送 PowerShell 恶意代码，而对于 Linux 系统的主机则执行 Bash 脚本；缓冲区溢出漏洞的利用也取决于系统的硬件(CPU 型号)以及软件(32 位系统或 64 位系统)。此外，特定的操作系统版本可能存在已经披露的漏洞可被攻击者利用，而其他版本的操作系统可能已经修补了该漏洞，因此攻击者只能寻找其他方式

展开攻击。需要指出的是,系统扫描并不仅仅可以提供关于操作系统的信息。除了运行Linux或Windows系统的服务器,交换机、打印机、路由器、调制解调器等其他众多设备都拥有自己的操作系统。如果能获知这些操作系统与设备的对应关系,那么硬件设备的具体型号也可以通过系统扫描披露出来。

攻击者甚至可以利用系统扫描发起社会工程学攻击:在获知详细的系统与硬件型号后,他们可以伪装成设备或系统的供应商或维护人员向网络管理者申请系统维护,并在此过程中植入木马。从安全维护人员的角度考虑,定期的系统扫描可以及时发现网络中的脆弱主机并提供帮助。

常用的系统扫描技术实际上是基于TCP/IP协议栈的指纹识别。扫描程序将一系列TCP和UDP数据包发送给目标主机,并将收到的回应数据包与数据库内已知系统指纹比对,匹配出目标主机的系统。这是因为尽管RFC定义了TCP/IP协议栈的标准,它并没有规定系统如何响应这些包。因此,不同系统对于这些包的响应不同,构成这些系统的指纹。

4. 服务扫描

得到了目标主机的开放端口后,还需要确定该端口提供的服务。因为目标主机可能将常见服务部署在非标准的端口上,利用服务扫描技术可以正确识别这些端口上的服务。

服务扫描程序根据不同服务的特点发送特定的数据包以获取对应服务在远程主机上的存在性,例如,发送"GET /"数据包以检测到Web服务等。此外,服务扫描程序也可以利用常用的服务与端口号的映射表来基于端口扫描的结果报告主机上运行的服务。如,TCP 80号端口通常对应HTTP服务(即,目标主机是一台Web服务器),TCP 25号端口的开放则表明简单邮件传输协议(Simple Mail Transfer Protocol, SMTP)的存在性(即,目标主机是一台邮件服务器),而开放了TCP 22号端口的目标主机则是一台部署了SSH服务的SSH服务器。然而,这种方式不一定是准确的,因为远程主机可以为各种服务分别指定有别于默认值的特定端口。

5. 网络扫描工具

网络映射器(Network Mapper, Nmap)是一个非常强大的网络扫描工具,其命令行使用方式如下所示:

```
$nmap [scan type(s)] [options] [target]
```

这里的[target]表示要扫描的对象,可以是IP地址、网段、主机名等,同时nmap命令还支持自定义扫描模式、参数等,部分参数需要sudo权限,表5-3介绍常用的几种参数,其余参数以及更详尽的用法请使用man nmap查看。

表5-3 nmap常用扫描参数介绍

参　数	含　义
-sN	ping扫描,主机发现
-sT	TCP连接扫描
-sS	TCP SYN扫描
-sA	TCP ACK扫描
-sU	UDP连接扫描

续表

参　数	含　义
-O	操作系统扫描
-sV	开放端口的系统服务以及版本扫描

例如，如果想要扫描 10.2.0.15 和 10.2.1.0/24 网段中所有的 UDP 端口，可以运行以下命令：

```
$nmap -sU 10.2.0.15 10.2.1.0/24
```

Zenmap 是 Nmap 的 GUI，由 Nmap 官方提供，能够在 Windows，Linux，macOS 等不同系统上运行，为用户提供更简单的操作方式。图 5-8 展示了该软件的主界面。

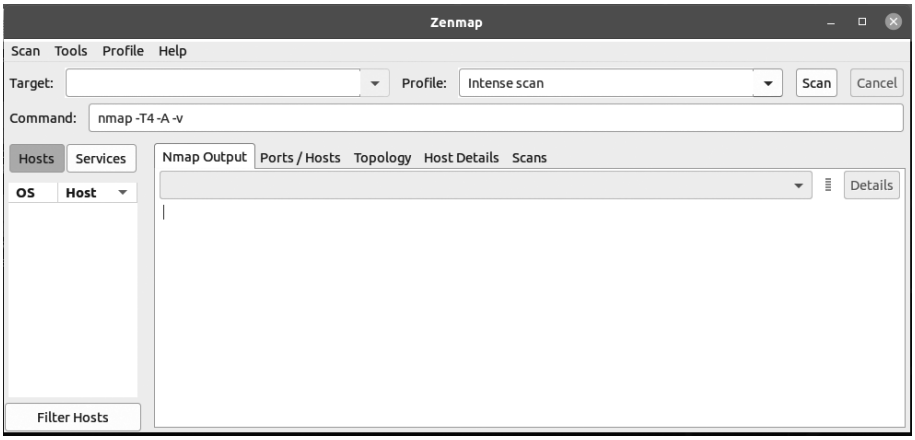

图 5-8　Zenmap 主界面

5.2.4　实验步骤

1. Nmap 工具实验

本实验在实验 5.1 配置好的环境（包括配置好的 4 台虚拟服务器）中操作。首先，服务器 2 需要关闭防火墙服务：

```
对于服务器 2:
$sudo ufw disable
```

任务 5.4　运用 nmap 命令及其参数，对 192.168.1.0/24，192.168.2.0/24，192.168.3.0/24 网段进行主机发现。

（1）截图记录操作与结果。

（2）基于结果回答问题：这些网段中存在哪些主机？

任务 5.5　针对完成任务 5.4 时发现的每台主机，回答下列问题，并截图记录操作过程与结果。其中，任务(1)、(2)、(3)需要使用 nmap 命令，并选用合适的参数执行扫描操作。

（1）这些主机的操作系统分别是什么？

（2）这些主机分别运行什么服务？