NMAP/ZENMAP 使用指南

2011-10-17

VERSION 1.0

目　录

一、Nmap/Zenmap 简介

——重点是功能介绍

Nmap （“Network Mapper（网络映射器）”） 是一款开放源代码的网络探测和安全审核的工具，支持在 Windows、Unix/Linux 以及 MAC OS 平台下运行。它的设计目标是快速地扫描大型网络，当然用它扫描单个主机也没有问题。Nmap 以新颖的方式使用原始 IP 报文来发现网络上有哪些主机，那些主机提供什么服务（应用程序名和版本），那些服务运行在什么操作系统（包括版本信息），它们使用什么类型的报文过滤器/防火墙，以及一堆其它功能。虽然 Nmap 通常用于安全审核，许多系统管理员和网络管理员也用它来做一些日常的工作，比如查看整个网络的信息，管理服务升级计划，以及监视主机和服务的运行。

Zenmap 是官方推出的一款基于 nmap 的安全扫描 GUI（图形用户界面），它是一款多平台（Windows、Unix/Linux、MAC OS 等）免费开源的应用程序，它的推出使得 Nmap 对于新手来说更容易上手同事也对有经验的用户提供了更高级的特性。经常被用到的扫描被保存成了预配置项，这使得重复利用变得极其容易。当然也允许互动的创建命令行。扫描的结果能够被保存下来供后续分析，可以将不同的保存下来的扫描结果进行对比以找出差异。最近的扫描结果会被保存在一个可搜索的数据库中。有了这款工具就让我们可以更直观的使用 Nmap 的强大功能。

二、端口扫描技术

——基础及进阶

1、端口扫描基础

虽然 Nmap 这些年来功能越来越多，它也是从一个高效的端口扫描器开始的，

并且那仍然是它的核心功能。nmap target 这个简单的命令扫描目标主机上的超过 1660 个 TCP 端口。许多传统的端口扫描器只列出所有端口是开放还是关闭的，Nmap 的信息粒度比它们要细得多。它把端口分成六个状态: open（开放的），closed（关闭的），filtered（被过滤的），unfiltered（未被过滤的），open|filtered（开放或者被过滤的），或者 closed|filtered（关闭或者未被过滤的）。当然这些状态并非端口本身的性质，而是描述 Nmap 怎样看待它们。例如：对于同样的目标机器的 135/tcp 端口，从同网络扫描显示它是开放的，而跨网络作完全相同的扫描则可能显示它是 filtered（被过滤的）。

Nmap 所识别的六个端口状态：

Open（开放的）：应用程序正在该端口接收 TCP 连接或者 UDP 报文。发现这一点常常是端口扫描的主要目标。安全意识强的人们知道每个开放的端口都是攻击的入口。攻击者或者入侵测试者想要发现开放的端口。而管理员则试图关闭它们或者用防火墙保护它们以免妨碍了合法用户。非安全扫描可能对开放的端口也感兴趣，因为它们显示了网络上哪些服务可供使用。

Closed（关闭的）：关闭的端口对于 Nmap 也是可访问的（它接受 Nmap 的探测报文并作出响应），但没有应用程序在其上监听。它们可以显示出该 IP 地址上（主机发现，或者 ping 扫描）的主机正在运行，也对部分操作系统探测有所帮助。因为关闭的关口是可访问的，也许过会儿值得再扫描一下，可能一些又开放了。系统管理员可能会考虑用防火墙封锁这样的端口。那样他们就会被显示为被过滤的状态，下面讨论。

Filtered（被过滤的）：由于包过滤阻止探测报文到达端口，Nmap 无法确定该端口是否开放。过滤可能来自专业的防火墙设备，路由器规则或者主机上的软件防火墙。这样的端口让攻击者感觉很挫折，因为它们几乎不提供任何信息。有时候它们响应 ICMP 错误消息如类型 3 代码 13（无法到达目标: 通信被管理员禁止），但更普遍的是过滤器只是丢弃探测帧，不做任何响应。这迫使 Nmap 重试，这使得扫描速度明显变慢。

Unfiltered（未被过滤的）：未被过滤状态意味着端口可访问，但 Nmap 不能确定它是开放还是关闭。只有用于映射防火墙规则集的 ACK 扫描才会把端口分类到这种状态。用其它类型的扫描如窗口扫描，SYN 扫描，或者 FIN 扫描来扫描未被

过滤的端口可以帮助确定端口是否开放。

Open|Filtered（开放或者被过滤的）：当无法确定端口是开放还是被过滤的，Namp 就把该端口划分成这种状态。开放的端口不响应就是一个例子。没有响应也可能意味着报文过滤器丢弃了探测报文或者它引发的任何响应。因此 Nmap 无法确定该端口是开放的还是被过滤的。UDP，IP 协议，FIN，Null，和 Xmas 扫描可能把端口归入此类。

Closed|Filtered（关闭或者被过滤的）：该状态用于 Nmap 不能确定端口是关闭的还是被过滤的。它只可能出现在 IPID Idle 扫描中。

2、Nmap 指令语法

nmap [<扫描类型> ...] [<选项>] { <扫描目标说明> }

其中扫描类型和选项参考后边的端口扫描技术部分，此处将详细解释一下扫描目标说明部分。

有时候您希望扫描整个网络的相邻主机。为此，Nmap 支持 CIDR 风格的地址。您可以附加一个/<numbit>在一个 IP 地址或主机名后面，Nmap 将会扫描所有和该参考 IP 地址具有 <numbit>相同比特的所有 IP 地址或主机。例如，192.168.10.0/24 将会扫描 192.168.10.0 (二进制格式: 11000000 10101000 00001010 00000000) 和 192.168.10.255 (二进制格式: 11000000 10101000 00001010 11111111)之间的 256 台主机。192.168.10.40/24 将会做同样的事情。假设主机 scanme.nmap.org 的 IP 地址是 205.217.153.62，scanme.nmap.org/16 将扫描 205.217.0.0 和 205.217.255.255 之间的 65,536 个 IP 地址。所允许的最小值是/1，这将会扫描半个互联网。最大值是/32，这将会扫描该主机或 IP 地址，因为所有的比特都固定了。

CIDR 标志位很简洁但有时候不够灵活。例如，您也许想要扫描 192.168.0.0/16，但略过任何以.0 或者.255 结束的 IP 地址，因为它们通常是广播地址。Nmap 通过八位字节地址范围支持这样的扫描您可以用逗号分开的数字或范围列表为 IP 地址的每个八位字节指定它的范围。例如，192.168.0-255.1-254 将略过在该范围内以.0 和.255 结束的地址。范围不必限于最后的 8 位：0-255.0-255.13.37 将在整个互联网范围内扫描所有以 13.37 结束的地址。这种大

范围的扫描对互联网调查研究也许有用。

IPv6 地址只能用规范的 IPv6 地址或主机名指定。CIDR 和八位字节范围不支持 IPv6，因为它们对于 IPv6 几乎没什么用。

Nmap 命令行接受多个主机说明，它们不必是相同类型。例如命令 nmap scanme.nmap.org 192.168.0.0/8 10.0.0，1，3-7.0-255 将和您预期的一样执行。

虽然目标通常在命令行指定，下列选项也可用来控制目标的选择：

-iL <inputfilename> (从列表中输入)

从 <inputfilename>中读取目标说明。在命令行输入 一堆主机名显得很笨拙，然而经常需要这样。例如，您的 DHCP 服务器可能导出 10,000 个当前租约的列表，而您希望对它们进行 扫描。如果您不是使用未授权的静态 IP 来定位主机，或许您想要扫描所有 IP 地址。只要生成要扫描的主机的列表，用-iL 把文件名作为选项传给 Nmap。列表中的项可以是 Nmap 在 命令行上接受的任何格式(IP 地址，主机名，CIDR，IPv6，或者八位字节范围)。每一项必须以一个或多个空格，制表符或换行符分开。如果您希望 Nmap 从标准输入而不是实际文件读取列表，您可以用一个连字符(-)作为文件名。

-iR <hostnum> (随机选择目标)

对于互联网范围内的调查和研究，您也许想随机地选择目标。<hostnum> 选项告诉 Nmap 生成多少个 IP。不合需要的 IP 如特定的私有，组播或者未分配的地址自动 略过。选项 0 意味着永无休止的扫描。记住，一些网管对于未授权的扫描可能会很感冒并加以抱怨。使用该选项的后果自负！如果在某个雨天的下午，您觉得实在无聊，试试这个命令 nmap -sS -PS80 -iR 0 -p 80 随机地找一些网站浏览。

--exclude <host1[，host2][，host3]，...> (排除主机/网络)

如果在您指定的扫描范围有一些主机或网络不是您的目标，那就用该选项加上以逗号分隔的列表排除它们。该列表用正常的 Nmap 语法，因此它可以包括主机名，CIDR，八位字节范围等等。当您希望扫描的网络包含执行关键任务的服务器，已知的对端口扫描反应强烈的 系统或者被其它人看管的子网时，这也许有用。

--excludefile <excludefile> (排除文件中的列表)

这和--exclude 选项的功能一样，只是所排除的目标是用以 换行符，空格，或者制表符分隔的 <excludefile>提供的，而不是在命令行上输入的。

3、端口扫描技术

既然 Nmap 是免费的，那么掌握端口扫描的唯一障碍就是知识。没有经验的用户和刚入门者总是用默认的 SYN 扫描解决每个问题，而专家则总能够选择最合适的一种扫描技术来完成给定的任务，这就是专家嘴里说的“打的扫描技术”。

虽然 Nmap 努力产生正确的结果，但请记住所有结果都是基于目标机器（或者它们前面的防火墙）返回的报文的。这些主机也许是不值得信任的，它们可能响应以迷惑或误导 Nmap 的报文。更普遍的是非 RFC 兼容的主机以不正确的方式响应 Nmap 探测。FIN， Null 和 Xmas 扫描 特别容易遇到这个问题。不过这几种都是特定扫描类型的问题，会在个别扫描类型中讨论到。

大部分扫描类型只对特权用户可用，这是因为他们发送接收原始报文在 Unix 系统下需要 root 权限，在 Windows 上推荐使用 administrator 账户。但是当 WinPcap 已经被加载到操作系统时，非特权用户也可以正常使用 Nmap。特权选项让 Nmap 强大得多也灵活得多。

Nmap 支持的扫描类型大概有十几种，一般一次只使用一种，不过 UDP 扫描（-sU）倒是可以和任何一种 TCP 扫描类型结合使用。端口扫描类型的选项格式是-s<*C*>，<*C*>通常是扫描类型的首字符（不过也有例外）。

默认情况下，Nmap 执行一个 SYN 扫描，但是如果用户没有权限发送原始报文（在 UNIX 上需要 root 权限）或者如果指定的是 IPv6 目标，Nmap 调用 connect()。以下列出的扫描类型中，非特权用户仅能执行 connect()和 ftp bounce()扫描。

-sS（TCP SYN 扫描/半开扫描）：SYN 扫描作为默认的也是最受欢迎的扫描选项，是有充分理由的。它执行得很快，在一个没有入侵防火墙的快速网络上，每秒钟可以扫描数千个 端口。SYN 扫描相对来说不张扬，不易被注意到，因为它从来不完成 TCP 连接。它不依赖于任何的特定平台，可以应对任何兼容的 TCP 协议栈。它之所以被称为“半开扫描”是因为它从来不打开一个完整的 TCP 连接。它的方式就是先发送一个 SYN 报文（就像真是要建立一个 TCP 连接一样），然后等待响应：

（1）如果收到 SYN/ACK 则表示端口在监听（也就是 Open 状态），Nmap 将直接回复 RST 结束 TCP 连接握手转入下一个端口。

（2）如果收到 RST（复位）则表示没有监听者（这代表该端口 Closed 状态）。

（3)如果经过多次重发之后仍然没有响应，该端口将被标记为被过滤（Filtered）。

（4）如果收到 ICMP 不可达错误（类型 3，代码 1、2、3、9、10、13），该端口也将被标记为被过滤（Filtered）。

-sT（TCP connect()扫描）：当 SYN 扫描不能用时，CP Connect()扫描就是默认的 TCP 扫描。当用户没有权限发送原始报文或者扫描 IPv6 网络时，就是这种情况。Nmap 通过创建 connect()系统调用要求操作系统和目标机以及端口建立连接，而不像其它扫描类型直接发送原始报文。这就和一般的网络应用程序（如 Web 浏览器、P2P 客户端）建立的连接一样属于高层系统调用。它是 Berkeley Sockets API 编程接口的一部分，Nmap 就是使用此 API 获得每个连接尝试的状态信息，而不是读取响应的原始报文。

但是一般来说 SYN 扫描（可用的话）是更好地选择。原因如下：

（1）connect()扫描效率较低（因为它需要建立完整的 TCP 连接）

（2）connect()扫描容易在目标机上留下记录。许多普通的 UNIX 系统上的服务会在 syslog 留下记录。如果管理员在日志里面看到来自同一系统的一堆尝试连接，他就应该知道他的系统被扫描了。

-sU（UDP 扫描）：虽然互联网上很多流行的服务都是基于 TCP 连接的，但是基于 UDP 的服务也不少（如 DNS、SNMP、DHCP，注册端口对应为 53、161/162、67/68，这是最常见的三种）。不过由于 UDP 扫描一般来说不较慢，比 TCP 要困难，所以一般的安全审核人员都容易忽略这些端口。其实这是一个错误的想法，因为可探测的 UDP 服务相当普遍，攻击者不会忽略整个协议，所幸的是 Nmap 可以帮助记录并报告 UDP 端口。

UDP 扫描通过-sU 激活，它可以与各种 TCP 扫描结合使用而同时检查两种协议。

UDP 扫描发送空的（没有数据）UDP 报头到每个目标端口。如果返回 ICMP 端口不可到达错误（类型 3，代码 3），则该端口是 closed（关闭的）。其它 ICMP 不可到达错误（类型 3，代码 1，2，9，10，或者 13）表明该端口是 filtered（被

过滤的）。偶尔地，某服务会响应一个 UDP 报文，证明该端口是 open（开放的）。如果几次重试后还没有响应，该端口就被认为是 open|filtered（开放|被过滤的）。这意味着该端口可能是开放的，也可能包过滤器正在封锁通信。可以用版本扫描(-sV)帮助区分真正的开放端口和被过滤的端口。

注意：慎用 UDP 扫描，因为它的扫描速度受到操作系统的限制非常低。如果确实要使用可以采用优先扫描主要端口、并发扫描更多主机、从防火墙后面扫描、使用—host—timeout 来跳过慢速的主机等手段。

-sN、-sF、-sX（TCP Null、FIN、Xmas 扫描）：这三种扫描方法其实是利用了 TCP RFC 中发掘出的一个微妙的方法来区分 Open（开放的）和 Closed（关闭的）端口（*RFC 中的描述为“如果目标端口状态是关闭的……进入的不含 RST 的报文导致一个 RST 的响应”，“不设置 SYN、RST、ACK 位的报文发送到开放端口，理论上这不应该发生，如果确实收到了，丢弃该报文，返回。”*）。

如果扫描系统遵循该 RFC 则当端口关闭时，任何不包含 SYN、RST、ACK 位的报文都会导致一个 RST 返回，而当端口开放时，应该没有任何响应。

Null 扫描（-sN）不设置任何标志位（TCP 标志头是 0）

FIN 扫描（-sF）只设置 TCP FIN 标志位

Xmas 扫描（-sX）设置 FIN、PSH、URG 标志位，就像点亮圣诞树上所有的灯一样，所以常称为圣诞树扫描。

除了探测报文标志位不同，这三种扫描的行为完全一样。

（1）如果收到一个 RST 报文则说明该端口是 Closed（关闭的）

（2）如果没有响应则意味着端口是 Open|Filtered（开放的|被过滤的）

（3）如果收到 ICMP 不可达错误（类型 3，代号 1、2、3、9、10 或者 13）该端口就会被标记为 Filtered（被过滤的）。

这三种扫描的优势就在于能够躲过一些无状态防火墙和报文过滤路由器，甚至比 SYN 扫描还要隐蔽一些，但是也不能依赖它们，现代的 IDS 产品可以发现它们。

这三种扫描的不足之处就在于并不是所有的系统都严格遵循 RFC793，许多系统（例如 Microsoft Windows、Cisco、BSDI、IBM OS/400 等）端口不管是开放还是关闭的都响应 RST。这就会导致所有的端口都被标记为 Closed（关闭的）。

不过这种扫描对大部分的 UNIX 操作系统都能很好的工作。再一个不足就是它们不能辨别 Open 端口和一些特定的 Filtered 端口，从而返回 Open|Filtered。

-sA（TCP ACK 扫描）：这种扫描与前面几种扫描相比的区别在于它无法确定 Open（开放的）或者 Open|Filtered（开放的|被过滤的）端口。这个扫描的目的在于发现防火墙规则，确定它们是有状态的还是无状态的，哪些端口是被过滤的。

ACK 扫描探测报文只设置 ACK 标志位。当扫描未被过滤的系统时，Open 和 Closed 端口都会返回 RST 报文，Nmap 把它们标记为 Unfiltered（未被过滤的），意思是 ACK 报文不能到达，但是他们是 Open 还是 Closed 无法确定。不响应的端口或者发送特定的 ICMP 错误消息（类型 3，代号 1、2、3、9、10 或者 13）的端口，标记为 Filtered。

-sW（TCP 窗口扫描）：这种扫描与 ACK 扫描的唯一却别在于它利用了特定系统的实现细节来区分端口是 Open 还是 Closed，而不全是打印成 Unfiltered。它通过检查返回到 RST 报文的 TCP 窗口域做到了这一点。在某些特定系统上，开放端口用正数表示窗口大小，关闭端口则用 0 表示窗口大小。因此，当收到 RST 时可以根据 TCP 窗口值是正数还是 0 分别把窗口状态标记为 Open 和 Closed。

不过由于这个扫描基于特定系统的实现细节，所以也不能完全相信它。不支持它的系统通常返回所有端口都是 Closed（不过一台机器所有端口都关闭也是有可能的）。如果大部分端口都是 Closed，而一些常见端口（22、25、53）是 Filtered，该系统就很可疑了。偶尔甚至会出现恰恰相反的行为，如果扫描显示 1000 个端口开放而只有 3 个端口是关闭的或者被过滤的，那么着三个端口很可能也是开放的端口。

以下几种扫描方式简单介绍，详细可以参考官方网站：

http://nmap.org/man/zh/man-port-scanning-techniques.html

-sM（TCP Maimon 扫描）：这个扫描技术除了探测报文采用 FIN/ACK 之外与 Null、FIN、Xmas 扫描完全一样。根据 RFC793（TCP），无论端口是开放还是关闭的，都应该对这样的探测响应 RST 报文。但是 Uriel Maimon 注意到如果端口开放，许多基于 BSD 的系统只是丢弃该探测报文。

--scanflags（定制的 TCP 扫描）：这是真正的 Nmap 高手使用的，他不受这些现成的扫描类型束缚，可以定义任意 TCP 标志位来设计自己的扫描。

-sI（Idel 扫描）：这种高级扫描方法允许进行真正的 TCP 端口盲扫描。由于它不适用真是 IP 进行扫描，所以它是一种极端隐蔽的扫描方法。

-sO（IP 协议扫描）：用于确认目标机器支持那些 IP 协议（TCP、ICMP、IGMP 等）。从技术上讲，这不是端口扫描。

-b（FTP 弹跳扫描）：通过 FTP 服务器做代理对其它目标主机进行扫描的技术。这是绕过防火墙的好方法。其实这算是服务器的一个弱点，当前大部分的服务器都被修补了，本弱点利用就困难了。不过倒是可以使用这个选项来判断服务器是否脆弱。

4、端口说明和扫描顺序

默认情况下，Nmap 用指定的协议对端口 1-1024 以及 namp-services 文件中列出的更高的端口进行扫描。

-p<port range>（只扫描指定的端口）：port range 使用单个端口和用连字符表示的端口范围都可以。范围的开始/或者结束值都可以被忽略（表示取 1 和 65535），所以可以指定-p 从端口 1 扫描到 65535。如果特别指定，也可以扫描端口 0。对于 IP 协议扫描（-sO），该选项指定希望扫描的协议号（0-255）。

当既扫描 TCP 端口又扫描 UDP 端口时，可以通过在端口号上添加 T:或者 U:指定协议。这个协议指定符一直有效到下一个指定符。例如：-p U:53,111,137,T:21-25,80,139,8080 将扫描 UDP 端口 53、111、137，同时扫描列出的 TCP 端口。

注意：如果既扫描 UDP 又扫描 TCP，必须指定-sU，以及至少一个 TCP 扫描类型（如：-sS、-sF 或者-sT）。如果没有给定协议限制符，端口号会加到所有的协议列表。

-F（快速（有限的端口）扫描）：在 Nmap 的 nmap-services 文件（对于-sO，是协议文件）中指定想要扫描的端口，这比扫描所有的 65535 个端口要快得多。因为这个列表包含如此多的 TCP 端口（1200 多），这和默认的 TCP 扫描（约 1600 端口）速度差别不是很大，如果使用—datadir 选项指定自己的小小的 nmap-services 文件，则差别就会很惊人了。

-r（不要按随机顺序扫描端口）：默认情况下，Nmap 按随机顺序扫描端口（除了

处于效率的考虑部分端口前移）。这种随机化通常是不错的，但是我们可以指定 -r 来顺序端口扫描。

三、Nmap 的拓展功能

——服务版本探测及操作系统探测

1、服务版本探测

把 Nmap 指向一个远程机器，它可能告诉您端口 25/tcp，80/tcp，和 53/udp 是开放的。使用 nmap-services 数据库（包含大约 2,200 个著名的服务），Nmap 可以报告那些端口可能分别对应于一个邮件服务器 (SMTP)，web 服务器(HTTP)，和域名服务器(DNS)。这种查询通常是正确的——事实上，绝大多数在 TCP 端口 25 监听的守护进程是邮件服务器。然而，我们不应该把赌注押在这上面！人们完全可以在一些奇怪的端口上运行服务。

即使 Nmap 是对的，假设运行服务的确实是 SMTP，HTTP 和 DNS，那也不是特别多的信息。当为您的公司或者客户作安全评估(或者甚至简单的网络明细清单)时，您确实想知道正在运行什么邮件和域名服务器以及它们的版本。有一个精确的版本号对了解服务器有什么漏洞有巨大帮助。版本探测可以帮您获得该信息。

在用某种其它类型的扫描方法发现 TCP 和/或者 UDP 端口后，版本探测会询问这些端口，确定到底什么服务正在运行。nmap-service-probes 数据库包含查询不同服务的探测报文 和解析识别响应的匹配表达式。Nmap 试图确定服务协议 (如 ftp，ssh，telnet，http)，应用程序名(如 ISC Bind，Apache httpd，Solaris telnetd)，版本号，主机名，设备类型(如 打印机，路由器)，操作系统家族 (如 Windows，Linux)以及其它的细节。当然，并非所有服务都提供所有这些信息。如果 Nmap 被编译成支持 OpenSSL，它将连接到 SSL 服务器，推测什么服务在加密层后面监听。当发现 RPC 服务时，Nmap RPC grinder (-sR)会自动被用于确定 RPC 程序和它的版本号。如果在扫描某个 UDP 端口后仍然无法确定该端口是开放的还是被过滤的，那么该端口状态就 被标记为 open|filtered。版本探测将试图从这些端口

引发一个响应(就像它对开放端口做的一样)，如果成功，就把状态改为开放。open|filtered TCP 端口用同样的方法对待。注意 Nmap -A 选项在其它情况下打开版本探测。详细可以探测：http://www.insecure.org/nmap/vscan/

用下列的选项打开和控制版本探测：

-sV（版本探测）：打开版本探测，也可以用-A 同时打开操作系统探测和版本探测。

--allports（不为版本探测排除任何端口）：默认情况下，Nmap 版本探测会跳过 9100 TCP 端口，因为一些打印机简单地打印送到该端口的任何数据，这会导致数十页 HTTP get 请求、二进制 SSL 会话请求等被打印出来。这一行为可以通过修改或删除 nmap-service-probes 中的 Exclude 指示符改变，您也可以不理会任何 Exclude 指示符，指定--allports 扫描所有端口。

--version-intensity <intensity> （设置版本扫描强度）：当进行版本扫描(-sV)时，nmap 发送一系列探测报文，每个报文都被赋予一个 1 到 9 之间的值。被赋予较低值的探测报文对大范围的常见服务有效，而被赋予较高值的报文一般没什么用。强度水平说明了应该使用哪些探测报文。数值越高，服务越有可能被正确识别。然而，高强度扫描花更多时间。强度值必须在 0 和 9 之间，默认是 7。当探测报文通过 nmap-service-probes ports 指示符注册到目标端口时，无论什么强度水平，探测报文都会被尝试。这保证了 DNS 探测将永远在任何开放的 53 端口尝试，SSL 探测将在 443 端口尝试等等。

--version-light （打开轻量级模式）：这是--version-intensity 2 的一个别名。这使得版本扫描快很多，但它识别服务的可能性也略微小一点。

--version-all （尝试每个探测）：这是--version-intensity 9 的一个别名。保证对每个端口尝试每个测试报文。

--version-trace（(跟踪版本扫描活动）：是会让 Nmap 打印出详细的关于正在运行的扫描的调试信息。它是使用—packet-trace 所得到的信息的子集。

-sR（RPC 扫描）：这种方法和许多端口扫描方法联合使用。它对所有被发现开放的 TCP/UDP 端口执行 SunRPC 程序 NULL 命令，来试图确定它们是否 RPC 端口，如果是，是什么程序和版本号。这作为版本扫描(-sV)的一部分自动打开。由于版本探测包括它并且全面得多，-sR 很少被用到。

2、操作系统探测

Nmap 最著名的功能之一就是用 TCP/IP 协议栈 fingerprinting 进行远程操作系统探测。Nmap 发送一系列 TCP 和 UDP 报文到远程主机，检查响应中的每一个比特。在进行一打测试（如 TCP ISN 采样，TCP 选项支持和排序，IPID 采样，初始窗口大小检查）之后，Nmap 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。每个 fingerprint 包括一个自由格式的关于 OS 的描述文本和一个分类信息，它提供供应商名称(如 Sun)和其下的操作系统(如 Solaris)、OS 版本(如 10)以及设备类型(通用设备，路由器，交换机，游戏控制台等)。

采用下列选项启用和控制操作系统检测：

-O（启用操作系统探测）：也可以使用-A 来同时启用操作系统和版本探测。

--osscan-limit（针对指定的目标进行操作系统检测）：如果发现一个打开和关闭的 TCP 端口时，操作系统检测会更有效。采用这个选项，Nmap 只对满足这个条件的主机进行操作系统检测，这样可以节约时间，特别在使用-P0 扫描多个主机时。这个选项仅在使用-O 或-A 进行操作系统检测时起作用。

--osscan-guess; --fuzzy （推测操作系统检测结果）：当 Nmap 无法确定所检测的操作系统时，会尽可能地提供最相近的匹配，Nmap 默认进行这种匹配，使用上述任一个选项使得 Nmap 的推测更加有效。

3、其他选项

-6（启用 IPv6 扫描）：ping 扫描（TCP-only）、连接扫描以及版本检测都支持 IPv6。除增加-6 选项外，其它命令语法相同。

-A（激烈模式扫描）：这个选项启用额外的高级和高强度选项，目前这个选项启用了操作系统探测（-O）和版本扫描（-sV）。目的是启用一个全面的扫描选项集合，不需要用户记忆大量的选项。这个选项仅仅启用功能，不包含可能用到的所需要的时间选项（如-T4）或细节选项。

--datadir <directoryname> (说明用户 Nmap 数据文件位置)：

--send-eth (使用原以太网帧发送)：

--send-ip (在原 IP 层发送)：

--privileged (假定用户具有全部权限)：

--interactive (在交互模式中启动)：

-V;--version（打印版本信息）：打印 Nmap 版本号并退出

-h;--help(打印帮助摘要)：打印一个短的帮助屏幕，列出大部分常用的命令选项，这个功能与不带参数运行 Nmap 是相同的。

四、CLI 模式（命令行界面）

——介绍 Nmap 的安装配置，验证

很多 Nmap 的老手都喜欢在 CLI 下直接跑 nmap 命令，所以根本就不去安装 GUI 界面（Zenmap），那么如果是这样的话我们应该如何安装呢？最常用的系统平台就是 Windows 和 Unix/Linux，对于 MAC OS 等其他系统下的安装可以参考官网页面（http://nmap.org/download.html）

1、Windows 下安装配置验证

a) 下载最新的命令行压缩文件包（http://nmap.org/download.html）nmap-5.61TEST1-win32.zip 或者 nmap-5.51-win32.zip
b) 将 zip 压缩包解压到相应目录下（可以放置常用目录下，如 D:\nmap）
c) 因为 Nmap 的相关协议解析需要使用到 winpcap 和 Visual C++运行是环境，所以要先将该目录下的两个文件执行安装（winpcap-nmap-verNum.exe 和 vcredist.exe）（默认安装即可）。
d) 将该目录路径添加到环境变量中（建议添加到系统级环境变量中）
e) 打开 CMD（windows vista/7 下建议使用管理员身份运行），输入 nmap 进行验证。

2、Unix/Linux 下安装验证

官网上提供的离线安装文件时 rpm 包格式（其他一些站点也提供 deb 包，

可以根据系统选择一下），不过如果在网速较好的情况下建议直接使用命令进行在线安装。

f) RPM 包的安装方式

参考官网说明进行（http://nmap.org/download.html 页面的 Linux RPM Source and Binaries 节）

g) 在线安装的方式(不能确保是最新版本的 nmap),仅以 Ubuntu 平台为例，其他 unix-like 系统类似

i. 打开终端，执行 sudo apt-get install nmap （根据提示输入超级用户密码）

ii. 根据提示进行即可（可能需要几次输入 y 进行确认）。

iii. 待出现安装完成提示后，输入 sudo nmap 进行测试(sudo 是由于 nmap 部分功能需要使用到超级用户权限)。

3、Nmap 的使用。

参考前面的扫描基础和扫描技术两部分。常用的一些扫描方法参考后续 GUI 模式下的说明。

五、GUI 模式（图形用户界面）/Zenmap

——介绍 GUI 界面方式下的 Nmap 使用方式（Zenmap 安装使用）

带有 GUI 的 Zenmap 使得新接触 Nmap 的用户更容易上手，同时也使得很多高级功能不用特别的记住复杂的配置选项。和 nmap 一样，Zenmap 也支持各种主流操作系统平台。下面也仅针对 Windows 和 Unix/Linux 平台下的安装进行说明。

注意：默认安装 Zenmap 就包含了其运行的所有条件，所以不用提前安装 nmap 或者 winpacap 等。

1、Windows 平台下的安装配置和验证

a) 到 nmap 官网下载最新自安装版本，nmap-5.61TEST1-setup.exe 或者 nmap-5.51-setup.exe

b) 双击安装，除安装位置可以自选外，其他都可以采用默认设置。（安装中可能会根据系统情况提示你安装 winpcap、visual C++运行时环境，均答复同意）

c) 安装完毕后执行 Zenmap 可以进行测试验证。

2、Unix/Linux 平台下的安装验证

以 Ubuntu 下的在线安装为例，离线 RPM/DEB 包的安装方式可以参考官网说明或者借助搜索引擎

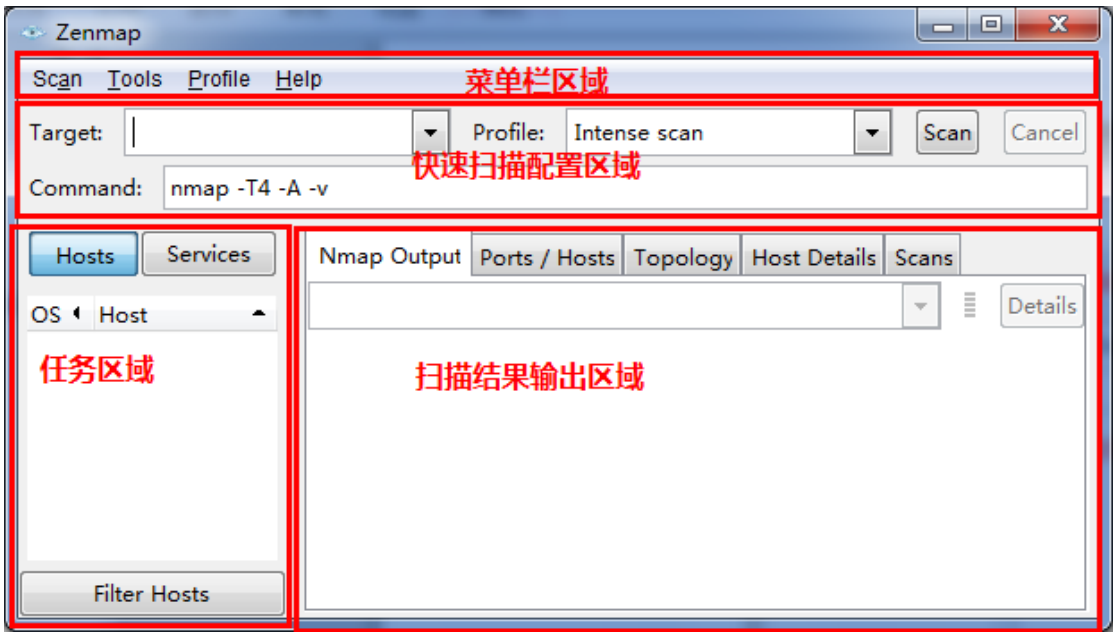
a) 打开终端，执行 sudo apt-get install zenmap （根据提示输入超级用户密码）

b) 安装过程中可能需要根据提示输入 y 进行确认。

c) 安装完毕后，在终端提示符下输入 sudo zenmap 执行（根据提示输入超级用户密码）验证。

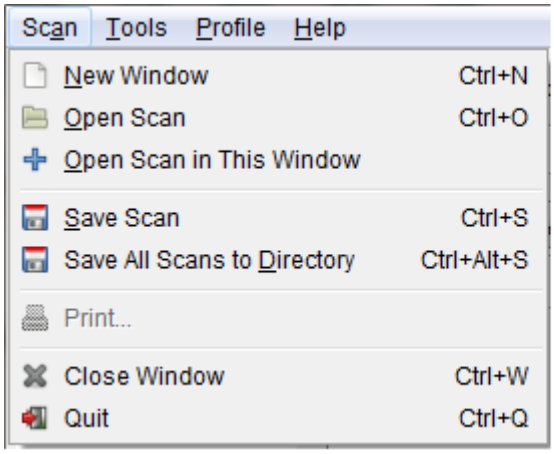
3、Zenmap 的常用使用方法

a) Zenmap 的预览及各区域简介



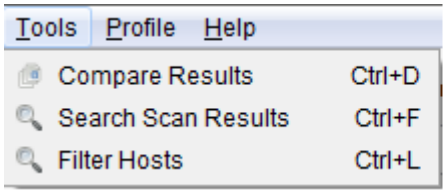
菜单栏区域：汇集 Zenmap 的各种功能的快捷连接.详细如下列图示

Scan 菜单：



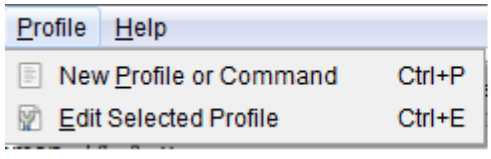
新开窗口
打开扫描文件
在本窗口中打开扫描文件
保存扫描结果
保存所有扫描结果到文件夹
打印
关闭窗口
退出

Tools 菜单：



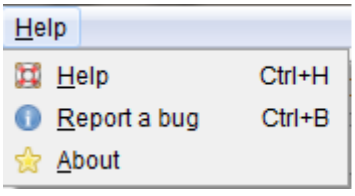
扫描结果对比
搜索扫描结果
过滤主机

Profile 菜单：



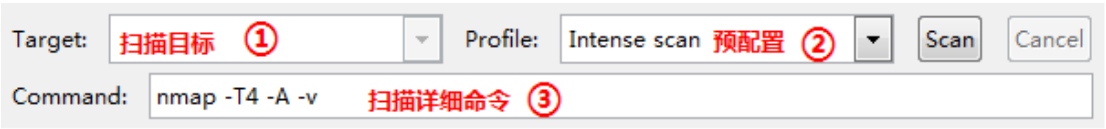
新配置文件或命令
编辑选定的预配置

Help 菜单：



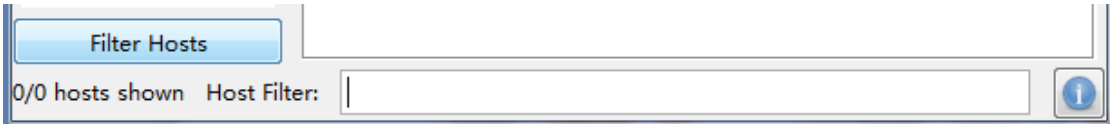
帮助
报告一个bug
关于

快速扫描配置区域：用于日常扫描的快速配置



任务区域：任务列表（含主机和服务）

此处列出所执行的任务列表，通过此处对任务进行切换查看。底部的“Filter Hosts”单击可以出现筛选列表用于在繁多的任务列表中定位要关注的主机。



扫描结果输出区域：扫描结果显示区域，包含最常用到的扫描结果显示（Nmap Output）、扫描的端口和主机（Ports/Hosts）、扫描主机拓扑（Topology）、扫描主机的详细信息（Host Details）以及扫描命令的详细说明（Scans）

b) 简单扫描流程

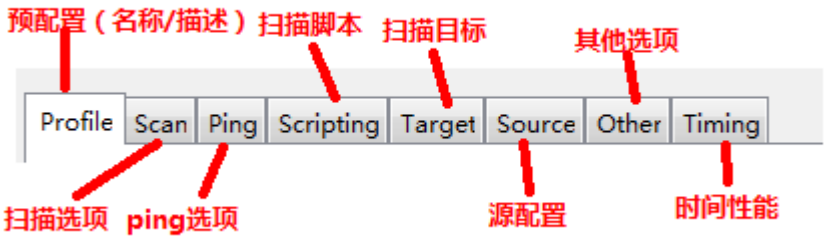
方法一：填写扫描目标→选择扫描预配置类型→根据需要修改扫描详细命令→执行扫描

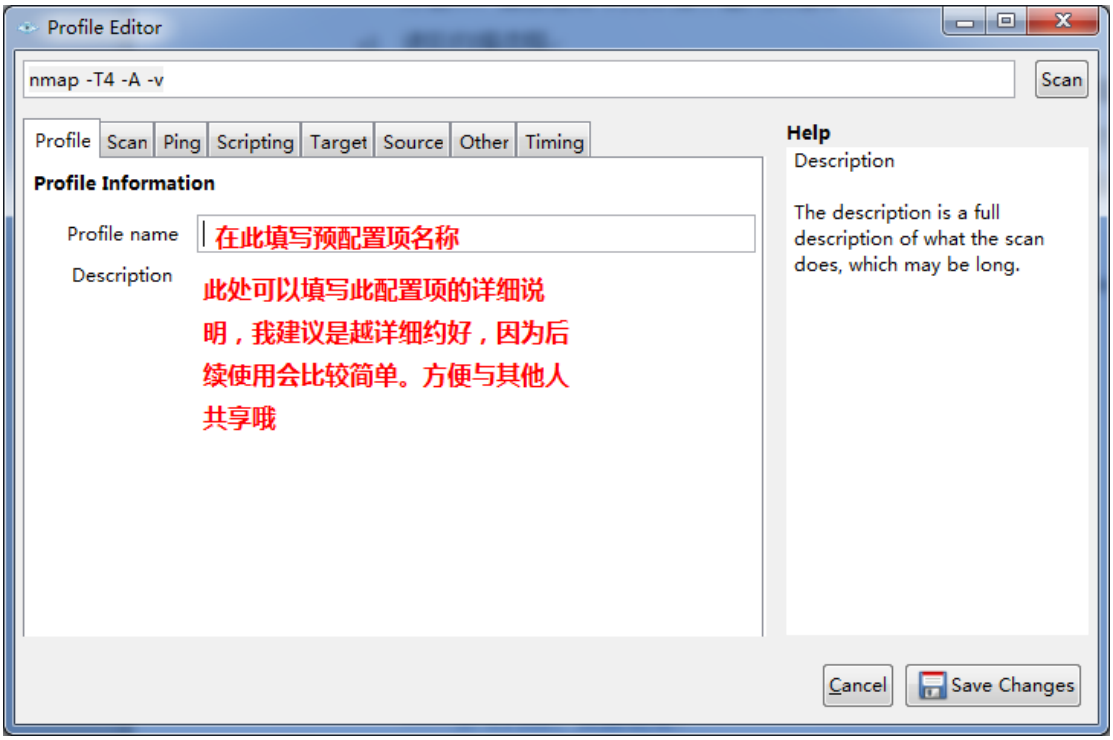
方法二：直接在命令文本框中输入扫描命令，执行扫描

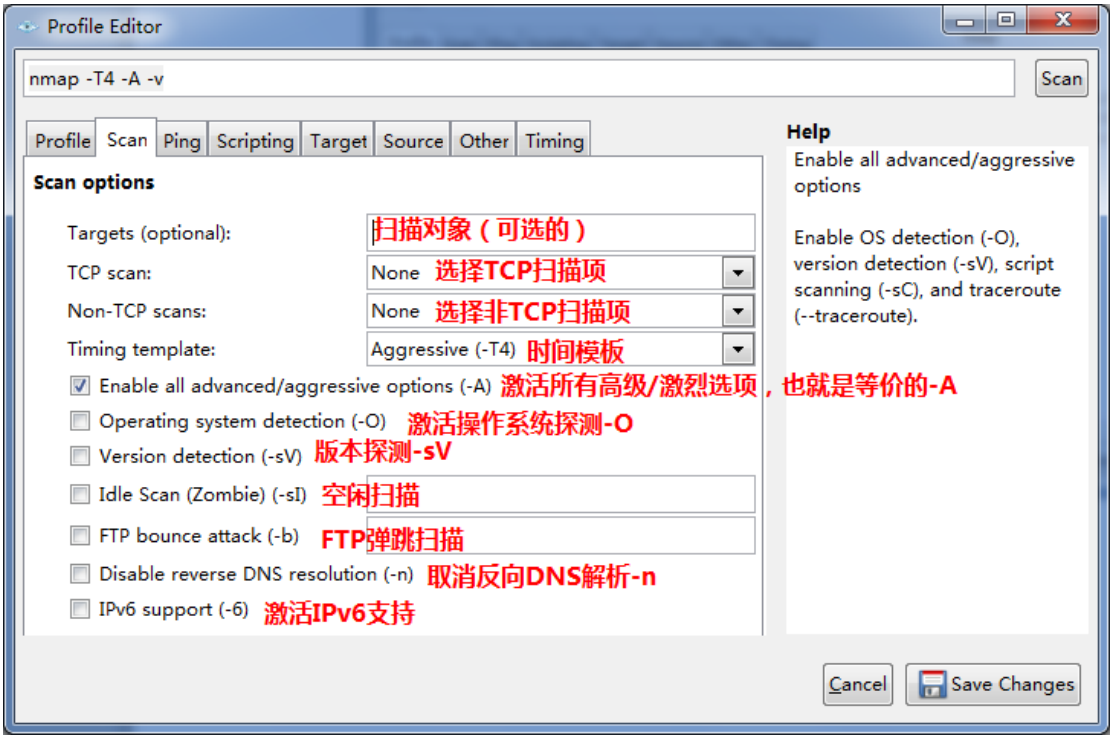
c) 进阶扫描流程

如果希望自己自定义的扫描可以重复利用，那么需要将自己编写的扫描指令存储为预配置，其自然就会出现在预配置下拉列表中，后续就可以很方便的使用了。当然也可以对系统自带的预配置项进行修改以更适应自己的需要。下面说明一下预配置制作的方法。

创建自己的预配置项：[Profile]→[New Profile or Command]打开 Profile 编辑器，后续就可以在该编辑器中创建预配置项，此处将主要的几个 TAB 配置也作简要说明，其他可参考选项右侧的说明。







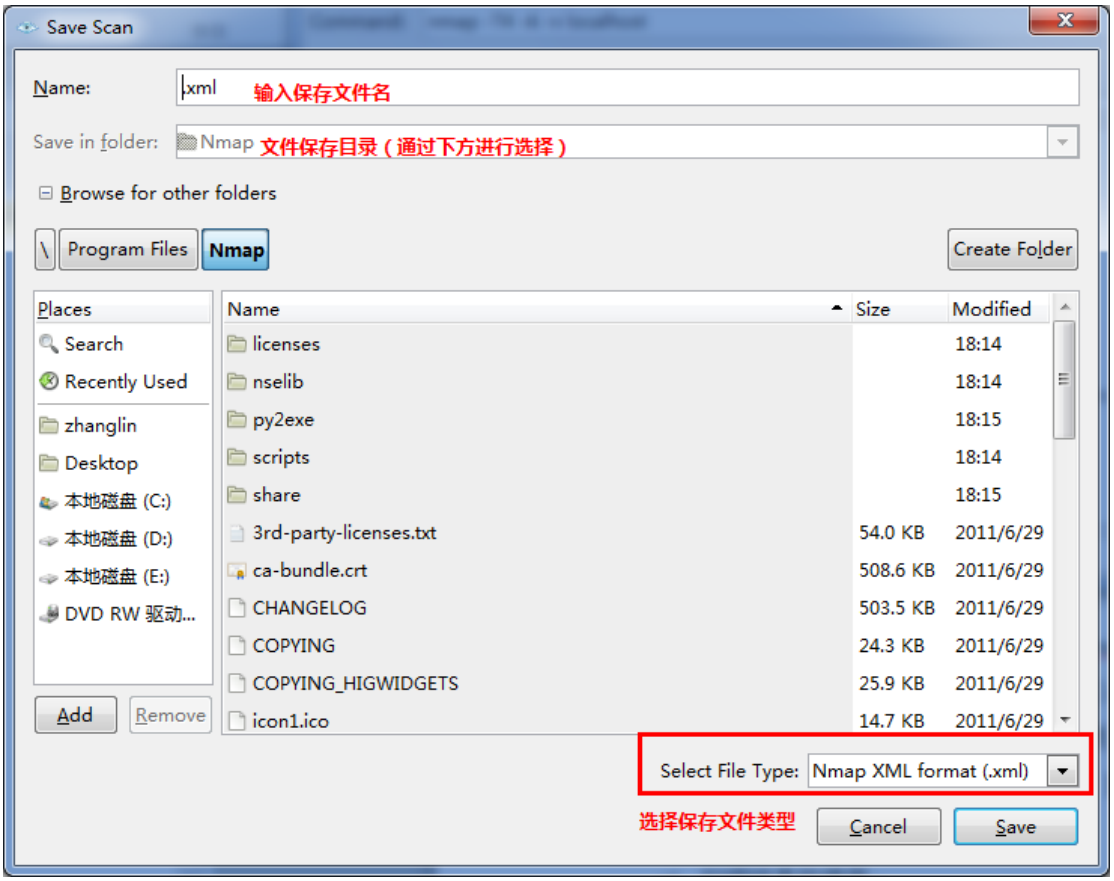
配置过程中可以在顶部的命令栏中看到详细的指令内容，等所有选项配置完毕后，点击右下角的保存按钮[Save Changes]即可。然后就可以在预配置下拉列表中看到自定义的预配置项。

修改已有的预配置项：在 Profile 下拉列表中选择待修改的选项，然后点击[Profile]→[Edit Selected Command]，剩下的就和新建没有啥区别了。

d) 扫描结果的保存

扫描任务完成后可以将扫描结果保存下来工延后分析使用，也可以用来对扫描任务的结果进行对比使用。Zenmap 提供的扫描结果保存的格式预定义的有两种（XML 格式和 nmap 格式）。其中 nmap 格式是纯文本格式，可以直接使用文本编辑器打开查看。而 XML 格式能够存储更多的信息，后续可以在 Zenmap 中打开还原使用。建议保存格式选择 XML。

[Scan]→[Save Scan]



当然如果有多个任务需要保存，可以直接选择[Scan]→[Save All Scans to Directory]。

当然后续回顾的时候直接选择[Scan]→[Open Scan]/[Open Scan in This Window]就可以打开了。

e) 扫描结果的使用/解读

此处以 nmap 官方提供的一个扫描地址“scanme.nmap.org”的扫描结果来解释，采用的是默认的激烈扫描模式“nmap –T4 –A –v scanme.nmap.org”）

注意：

1、 默认情况下 Zenmap 上显示的内容是不能直接拷贝粘贴出去的，所以你可以通过保存为 nmap

格式，然后用记事本打开来实现。

扫描结果在 Zenmap 上展示在五个 Tab 页上，但是基本上所有内容都体现在【Nmap Output】页面中，其他页面可以算作是对此页面的可视化的解释说明。所以只要将此页面解释清楚就基本上能够明白了。

Starting Nmap 5.59BETA1 (http://nmap.org) at 2011-09-22 18:34 中国标准时间【开始扫描】

NSE: Loaded 63 scripts for scanning. 【Nmap 脚本引擎：完成 63 个扫描脚本的载入】

NSE: Script Pre-scanning. 【Nmap 脚本引擎：脚本的预扫描】

Initiating Ping Scan at 18:34【初始化 ping 扫描】

Scanning sanme.nmap.org (74.207.254.18) [4 ports] 【扫描目标机】

Completed Ping Scan at 18:34, 0.90s elapsed (1 total hosts) 【完成 ping 扫描】

Initiating Parallel DNS resolution of 1 host. at 18:34【初始化反向 DNS 解析】

Completed Parallel DNS resolution of 1 host. at 18:34, 0.48s elapsed【完成反向 DNS 解析】

Initiating SYN Stealth Scan at 18:34【初始化 SYN 隐蔽扫描】

Scanning sanme.nmap.org (74.207.254.18) [1000 ports] 【扫描……】

Discovered open port 22/tcp on 74.207.254.18【发现了开放端口 22】

Discovered open port 80/tcp on 74.207.254.18【发现了开放端口 80】

SYN Stealth Scan Timing: About 41.27% done; ETC: 18:35 (0:00:44 remaining) 【SYN 隐蔽扫描时间】

Completed SYN Stealth Scan at 18:35, 43.23s elapsed (1000 total ports) 【完成 SYN 隐蔽扫描】

Initiating Service scan at 18:35【初始化服务扫描】

Scanning 2 services on sanme.nmap.org (74.207.254.18) 【在目标主机上扫描*个服务】

Completed Service scan at 18:35, 6.49s elapsed (2 services on 1 host) 【完成服务扫描】

Initiating OS detection (try #1) against sanme.nmap.org (74.207.254.18) 【初始化操作系统探测】

Retrying OS detection (try #2) against sanme.nmap.org (74.207.254.18) 【操作系统探测重试】

Initiating Traceroute at 18:35【初始化路由追踪】

Completed Traceroute at 18:35, 3.04s elapsed【完成路由追踪】

Initiating Parallel DNS resolution of 21 hosts. at 18:35【初始化*个反向 DNS 解析】

Completed Parallel DNS resolution of 21 hosts. at 18:35, 0.66s elapsed【完成*个反向 DNS 解析】

NSE: Script scanning 74.207.254.18. 【Nmap 脚本引擎：脚本扫描】

Initiating NSE at 18:35【初始化 Nmap 脚本引擎】

Completed NSE at 18:35, 10.11s elapsed【完成 Nmap 脚本引擎】

Nmap scan report for sanme.nmap.org (74.207.254.18) 【目标主机的 Nmap 扫描报告】

Host is up (0.22s latency). 【主机是活跃的】

rDNS record for 74.207.254.18: web.insecure.org【目标主机的反向 DNS 记录是……】

Not shown: 996 filtered ports【不显示……个被过滤的端口】

【下面这部分列出的是开放的或者关闭的端口，对应到 Ports/Hosts 页面】

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 4.3 (protocol 2.0)

| ssh-hostkey: 1024 81:35:70:a0:5b:9a:d2:b6:ab:04:c6:36:e6:d4:12:49 (DSA)

|_2048 c0:2c:03:87:68:e6:be:d1:6e:20:0c:48:cf:a9:74:d5 (RSA)

80/tcp open http Apache httpd 2.2.3 ((CentOS))

|_http-methods: No Allow or Public header in OPTIONS response (status code 301)

| http-title: Nmap - Free Security Scanner For Network Exploration & Securit...

|_Requested resource was http://nmap.org/

113/tcp closed auth

31337/tcp closed Elite

【下面这部分显示的是操作系统探测的一个过程，与 OS fingerprint 的匹配程度以用来确认目标主机的操作系统类型和版本，对应到 Host Details 页面】

Aggressive OS guesses: Netgear DG834G WAP (90%), Linux 2.6.19 - 2.6.36 (90%), Linux 2.6.24 - 2.6.35 (90%), Linux 2.6.34 (90%), Linux 2.6.18 (Slackware 11.0) (90%), OpenWrt (Linux 2.4.32) (89%), Linux 2.6.22 (89%), Linux 2.6.31 (88%), Linksys WRV54G WAP (87%), OpenWrt White Russian 0.9 (Linux 2.4.30) (87%)

No exact OS matches for host (test conditions non-ideal). 【没有完全匹配的操作系统类型】

Uptime guess: 14.083 days (since Thu Sep 08 16:36:11 2011) 【猜测操作系统上次的启动时间】

Network Distance: 23 hops【网络距离*跳】

TCP Sequence Prediction: Difficulty=217 (Good luck!) 【TCP 序列预测：难度是*】

IP ID Sequence Generation: All zeros

【下面是目标主机的路由探测过程，对应到 Topology 页面】

TRACEROUTE (using port 113/tcp)

HOP RTT ADDRESS

1 3.00 ms 117.130.126.129

2 ... 3

4 5.00 ms 117.128.8.193

5 5.00 ms 221.130.39.9

6 14.00 ms 211.136.94.153

7 18.00 ms 221.179.171.45

8 20.00 ms 221.179.171.5

9 56.00 ms 211.136.7.173

10 36.00 ms 211.136.2.26

11 76.00 ms 221.176.18.114

12 39.00 ms 221.176.24.130

13 42.00 ms 211.136.1.105

14 40.00 ms CME-0001.asianetcom.net (203.192.178.241)

15 41.00 ms gi3-0-0.cr3.hkg3.asianetcom.net (203.192.134.65)

16 46.00 ms te0-0-2-0.wr1.hkg0.asianetcom.net (61.14.157.101)

17 103.00 ms te0-2-0-0.wr1.osa0.asianetcom.net (61.14.157.70)

18 201.00 ms gi6-0-0.gw1.sjc1.asianetcom.net (61.14.157.98)

19 213.00 ms po2-0-0.gw2.sjc1.asianetcom.net (202.147.50.130)

20 205.00 ms 10gigabitethernet2-2.core1.sjc2.he.net (216.218.192.233)

21 240.00 ms 10gigabitethernet1-1.core1.fmt1.he.net (72.52.92.109)

22 208.00 ms linode-llc.10gigabitethernet2-3.core1.fmt1.he.net (64.62.250.6)

23 203.00 ms web.insecure.org (74.207.254.18)

NSE: Script Post-scanning. 【Nmap 脚本引擎：脚本快速扫描】

Initiating NSE at 18:35【初始化 Nmap 脚本引擎】

Completed NSE at 18:35, 0.00s elapsed【完成 Nmap 脚本引擎】

Read data files from: D:\Program Files\Nmap【从 Nmap 安装目录读取数据文件】

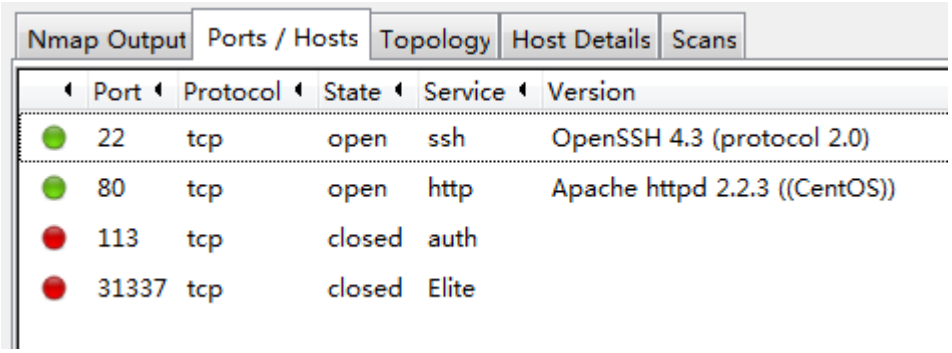
OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .【如果在执行完操作系统和服务探测后发现任何错误请提交报告到……】

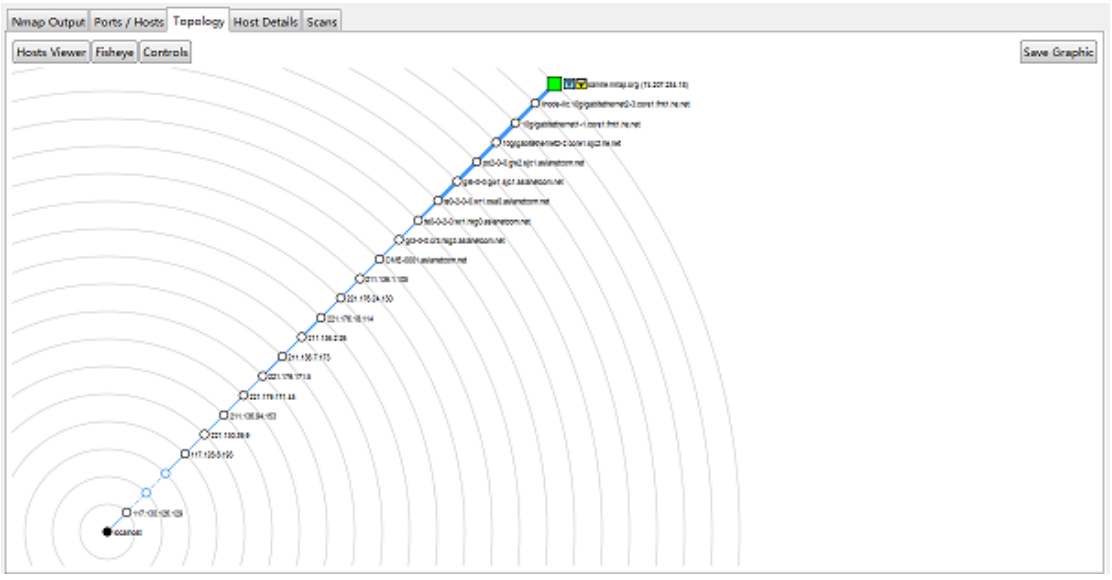
Nmap done: 1 IP address (1 host up) scanned in 76.59 seconds

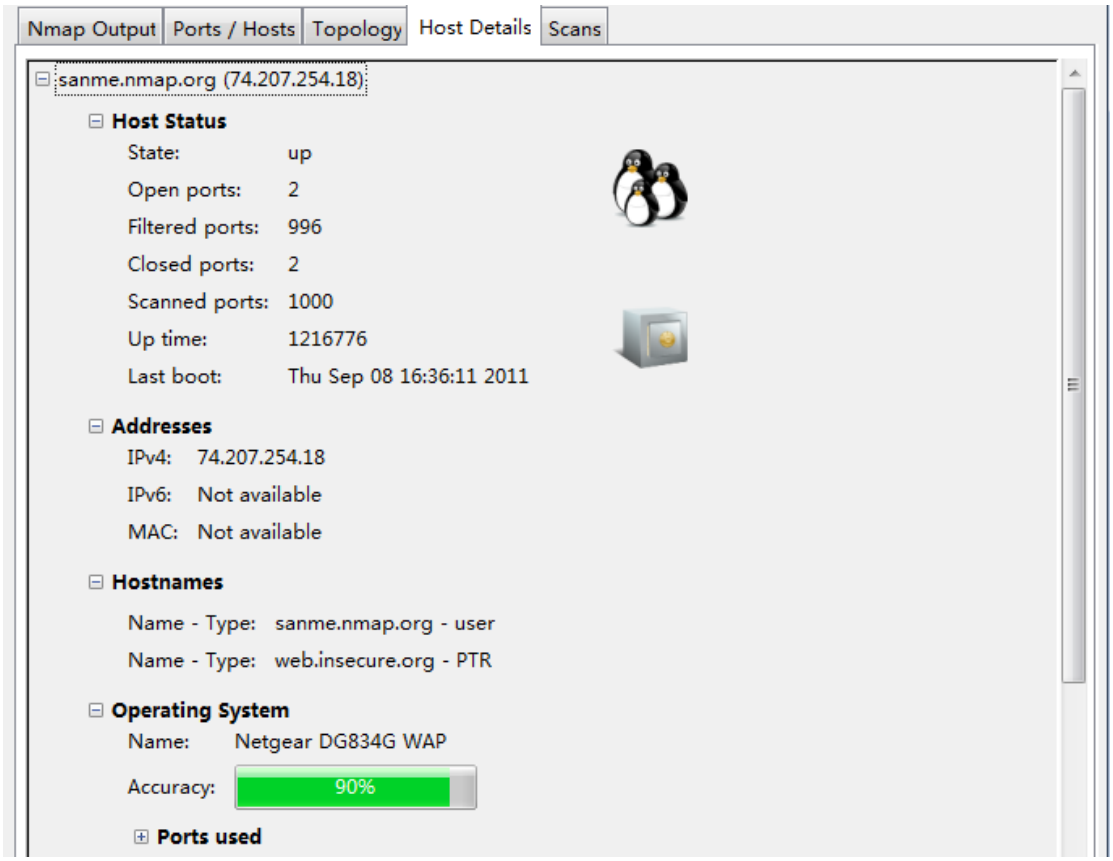
Raw packets sent: 3105 (139.324KB) | Rcvd: 103 (16.664KB)

【Nmap 结束】

下面截几张图来看看：


Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Port
Protocol
State
Service
Version
22 tcp open ssh OpenSSH 4.3 (protocol 2.0)
80 tcp open http Apache httpd 2.2.3 ((CentOS))
113 tcp closed auth
31337 tcp closed Elite

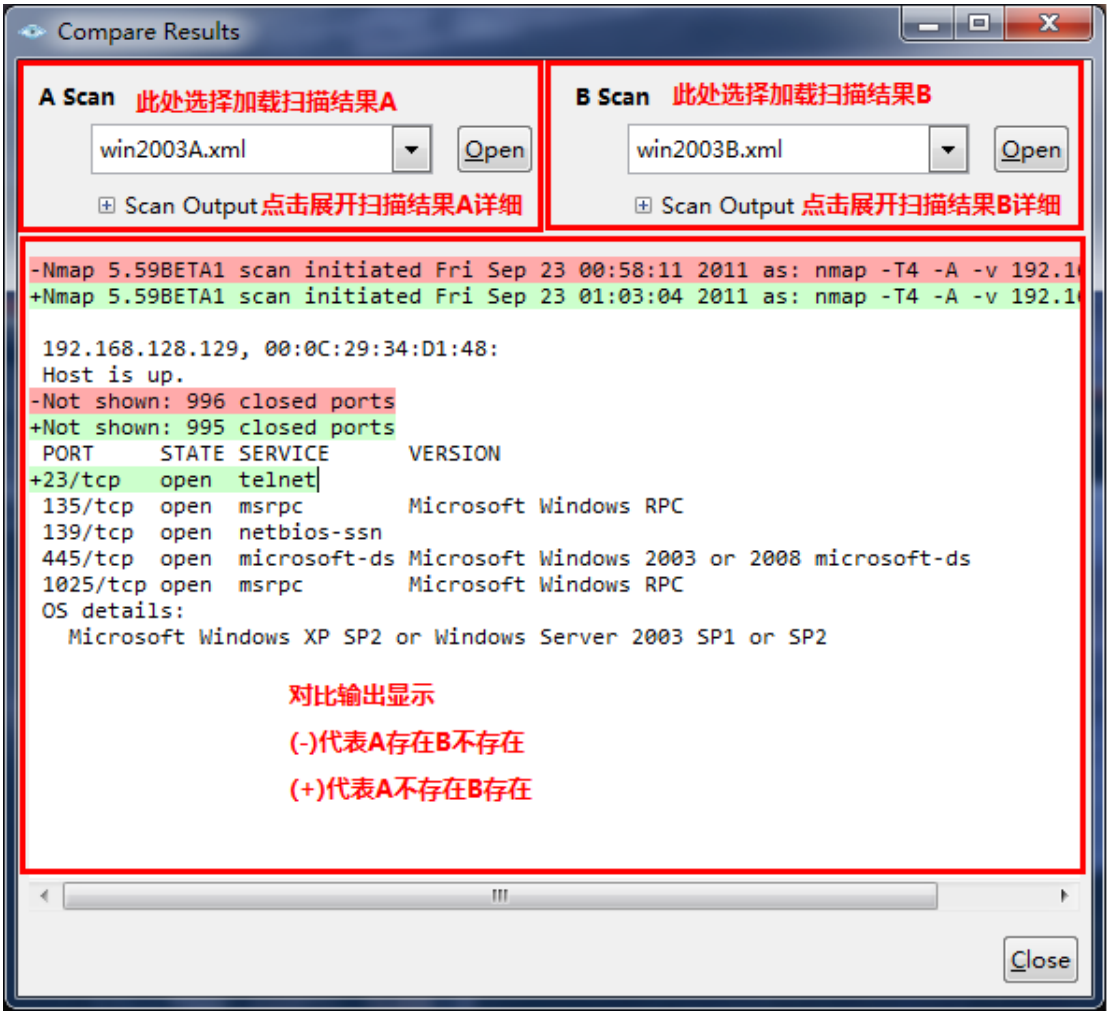

Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Hosts Viewer
Fisheye
Controls
Save Graphic



f) 其他功能的使用

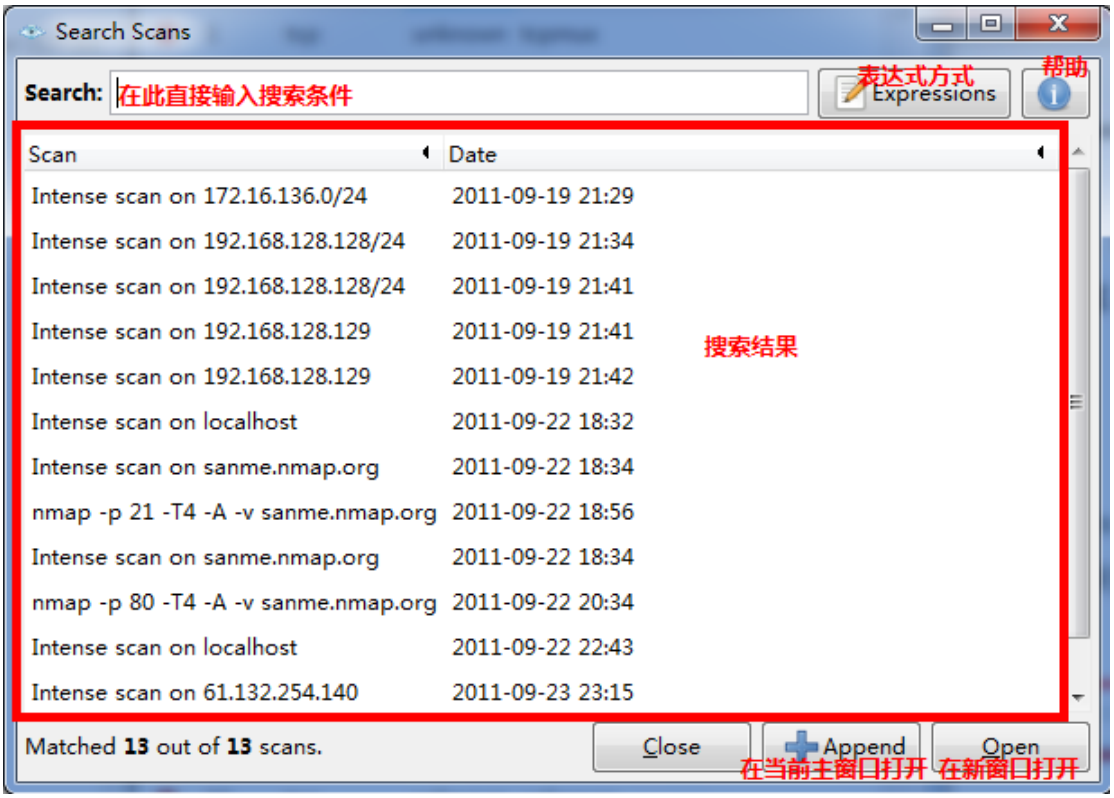
①扫描结果对比

[Tools]→[Compare Results]打开扫描结果对比窗口，如下：



②搜索扫描结果

[Tools]→[Search Scan Results]之后打开扫描结果查询窗口，该窗口用来对扫进行过的历史扫描任务进行查询。查询支持项非常丰富，支持条件组合查询。



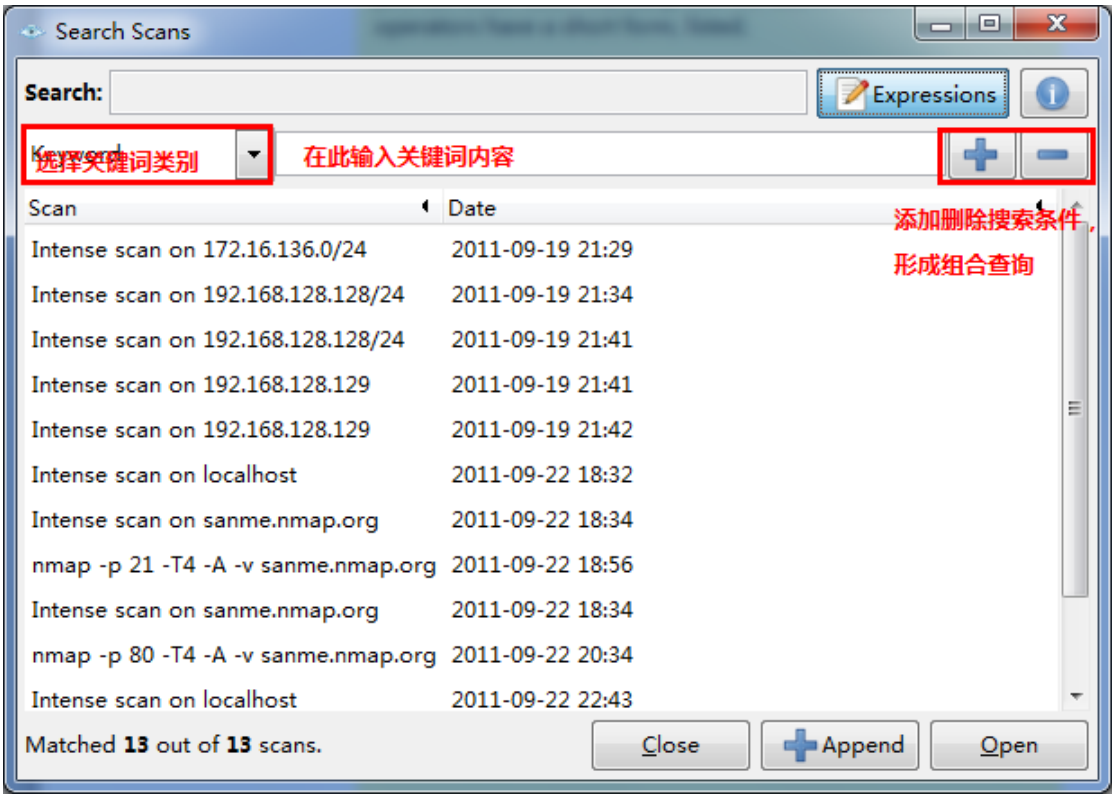
点击右上角的**蓝色叹号按钮**即可打开支持的搜索语句说明

Entering the text into the search performs a **keyword search** - the search string is matched against the entire output of each scan.

To refine the search, you can use **operators** to search only within a specific part of a scan. Operators can be added to the search interactively if you click on the **Expressions** button, or you can enter them manually into the search field. Most operators have a short form, listed.

profile: (pr:) - Profile used.
target: (t:) - User-supplied target, or a rDNS result.
option: (o:) - Scan options.
date: (d:) - The date when scan was performed. Fuzzy matching is possible using the "~" suffix. Each "~" broadens the search by one day on "each side" of the date. In addition, it is possible to use the "date:-n" notation which means "n days ago".
after: (a:) - Matches scans made after the supplied date (*YYYY-MM-DD* or *-n*).
before (b:) - Matches scans made before the supplied date (*YYYY-MM-DD* or *-n*).
os: - All OS-related fields.
scanned: (sp:) - Matches a port if it was among those scanned.
open: (op:) - Open ports discovered in a scan.
closed: (cp:) - Closed ports discovered in a scan.
filtered: (fp:) - Filtered ports discovered in scan.
unfiltered: (ufp:) - Unfiltered ports found in a scan (using, for example, an ACK scan).
open|filtered: (ofp:) - Ports in the "open|filtered" state.
closed|filtered: (cfp:) - Ports in the "closed|filtered" state.
service: (s:) - All service-related fields.
inroute: (ir:) - Matches a router in the scan's traceroute output.

如果对这种语句式输入感觉不够直观的话，可以打开帮助按钮前面的**[Expressions]**按钮。



③主机过滤

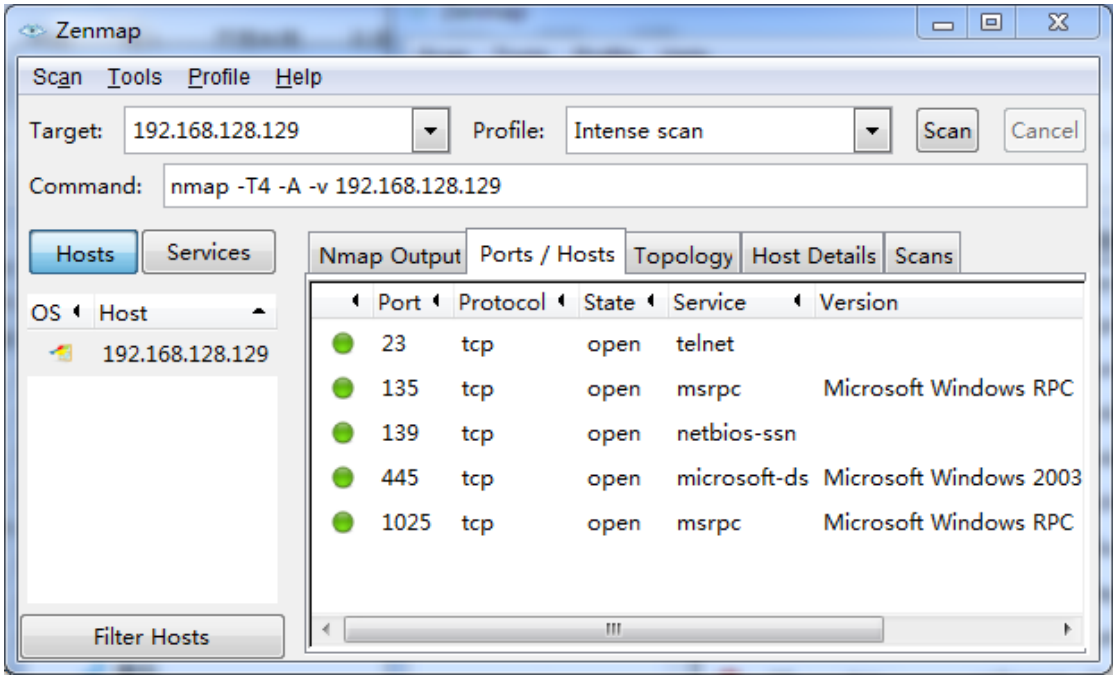
[Tools]→[Filter Hosts]或者直接点击窗口左下角”Filter Hosts 按钮”,在出现的注解过来输入框中输入需要查看的主机 IP 即可过滤出希望重点查看的主机（一般在有较多扫描任务时候使用）。

六、应用实例

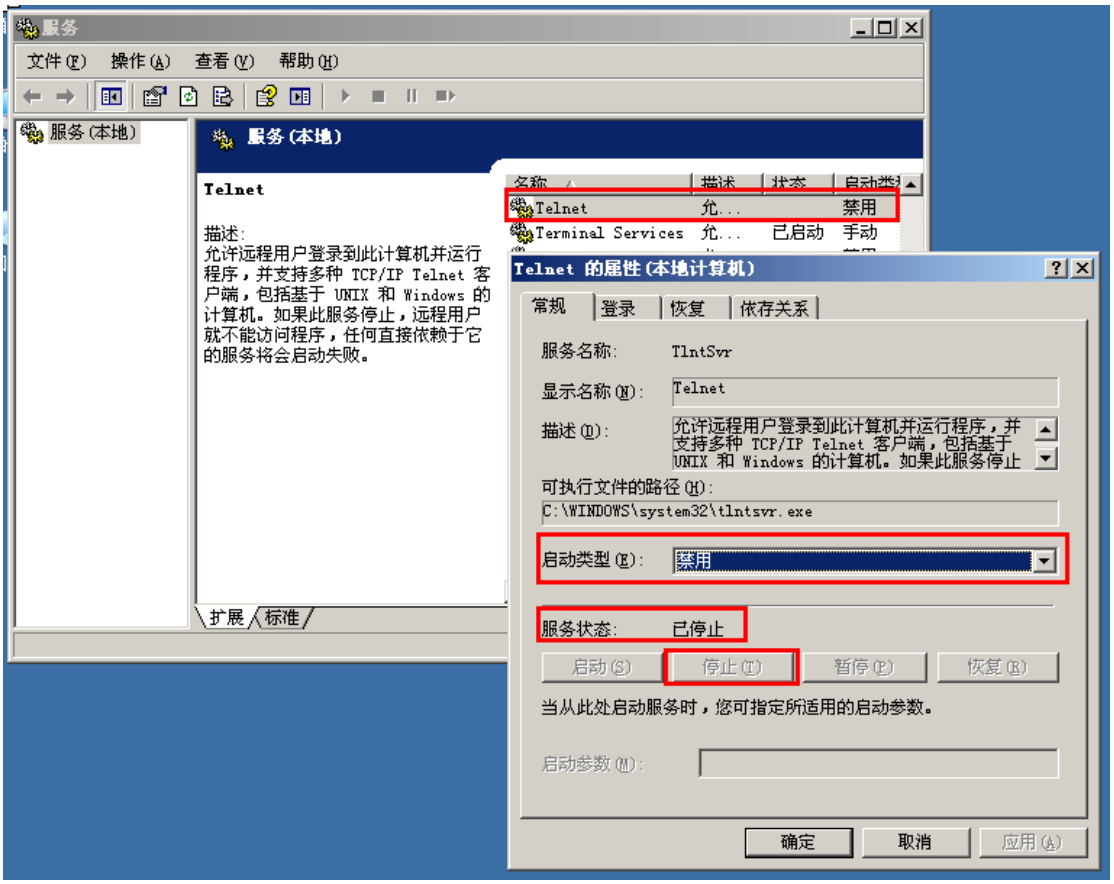
——简单举例，可以有命令行模式和 GUI 模式

就 Nmap/Zenmap 如果针对我们的工作来使用，主要用来确认相应主机开放的端口及服务，基于服务最小需求开放最大安全的原则将不需要的服务和端口进行关闭。

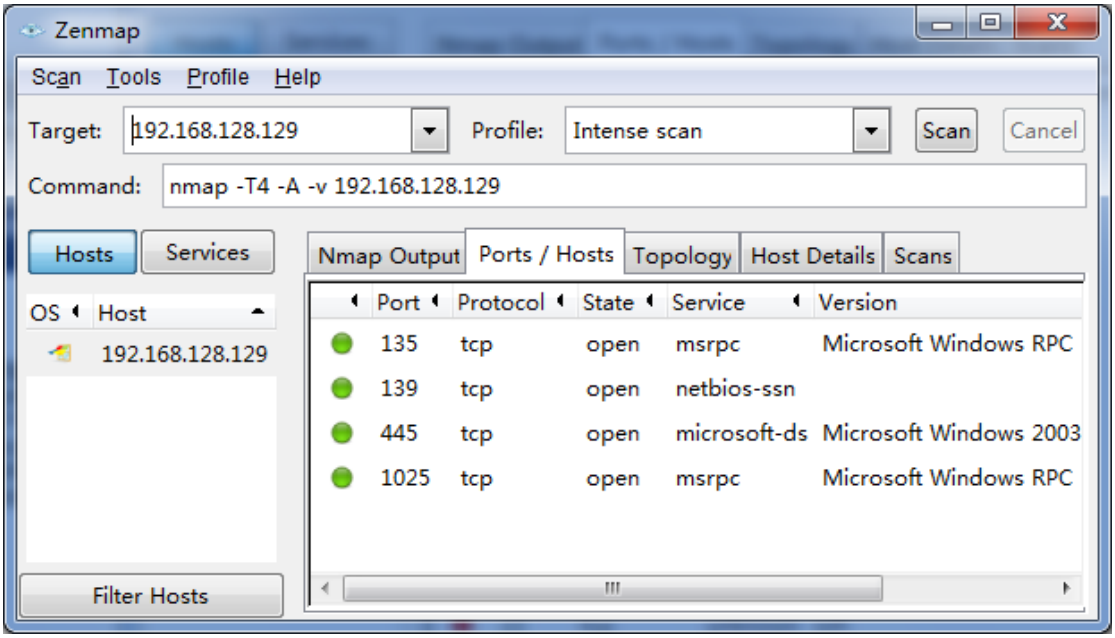
假如我们有一台服务器，通过 Nmap/Zenmap 扫描结果如下：



但是我们并不希望该服务器对外提供任何 Telnet 服务，那么我们可以将该服务适时的关闭掉以保证服务器的安全。



关闭后可以通过重新扫描进行验证服务器配置是否成功。



从上图可以看出，Telnet 服务及相关端口已被成功关闭掉。

七、附录

——参考内容、资源地址

1、 Nmap 官网主页（http://nmap.org）（http://insecure.org）

2、 Nmap/Zenmap 下载地址（http://nmap.org/download.html）

3、 Nmap/Zenmap 安装指南（http://nmap.org/book/install.html）

4、 Nmap 中文参考文档（http://nmap.org/man/zh/）