基于 Python 的端口扫描技术研究

宋来建
（重庆科创职业学院，重庆 402160）

摘　要　端口扫描技术作为一种工作和学习中的重要技术，对于测试网络、构建网络布局有着十分重要的作用，基于 Python 的端口扫描更是将这种扫描技术精细化，提高了验算效率，是现今许多代码工作所倾向使用的重要技术，然而在这一技术的使用过程中有许多细节之处需要使用者注意。基于此，本文对基于 Python 的端口扫描技术进行了探究，旨在为同行业者提供有益参考。

关键词　端口扫描技术　缩进扫描　连接扫描　秘密扫描　间接扫描
中图分类号：TP3　　**文献标识码**：A　　**文章编号**：1007-0745(2022)02-0007-03

端口在计算机发展和使用过程中起着十分重要的作用，端口能够连接计算机与外界，实现两者之间信息的交互。通常情况下，客户并不关心使用的端口号，只需要保证该端口在本机上是唯一的即可。

1 端口扫描技术概述

1.1 TCP/IP 工作原理

TCP/IP 参考模型来源于 20 世纪 60 年代末美国的一个网络分组交换研究项目，是一系列网络协议的总称。这些协议的存在能够促使计算机之间的信息交换变得更加方便和容易。TCP/IP 参考模型一共分为四层，每一层都负责不同的通信功能，从上到下分别为：应用层、传输层、Internet 层、网络接入层。TCP/IP 参考模型的传输层包括 TCP 和 UDP 两种，前者是指传输控制协议，主要面向受控对象，提供可靠的、面向连接的传输任务，在传送数据前务必先要建立连接，在确认信息到达目的之后，再开始对错误的检查与修复、顺序和流量的控制等。后者是指用户数据报送协议，这一传输层主要提供不可靠、无连接的传输服务，在传送数据钱无需提前建立连接，也不需要确认信息是否送达。

1.2 端口的定义

端口专门为计算机而设计的，负责计算机通信，可以说，离开了端口计算机通信就无法进行。在计算机网络中，一般分为以下两种端口：硬件端口和软件端口。集线器、交换机、路由器等网络设备由于可以连接到其他网络设备上使用，故被称为硬件端口。软件端口通常有套接字中的端口。套接字是由 IP 地址和端口两部分组成。软件端口通常而言是逻辑概念，定义了计算机之间通过软件通信的方式。

1.3 端口扫描技术

端口扫描是一种通过连接到目标计算机的 TCP 和 UDP 端口来确定目标计算机上运行的服务方法。现阶段，端口扫描主要用于识别计算机上启用的服务和开放的端口；识别计算机上操作系统类型和系统信息；识别在线的一台计算机或多台计算机组成个网络。

1.4 端口扫描的目的

端口扫描技术的出现主要是为了满足网络管理员、网络技术人员以及普通计算机用户的需求而诞生的一种科学技术。每个用户进行端口扫描的目的也有所不同，所以不同的用户对网络的使用方式不同。普通用户使用端口扫描是为了了解某一特定的服务器是否可以提供自己需要的服务；网络技术人员进行端口扫描是为了利用已经打开的窗口来获取对自己有用的信息；而网络管理员进行端口扫描主要是为了关闭不在继续使用的端口，进而安装有漏洞的端口补丁程序。

2 通过扫描主机端口建立循环嵌套构

我们都知道，C、Python 等语言都可以利用来编写端口扫描器。（大部分互联网应用使用的都是 TCP 协议，如 HTTP 在 TCP80 端口上，SMTP 在 TCP25 端口上等等）。这种扫描方式可用来检测端口是否被过滤。TCPFIN 扫描一一请求服务的一方发送一个 FIN=1 的数据包，这种扫描方式大多用于判断操作类型。此外，还有很多种扫描方式，对应于不同的网络环境，选择不同的扫描方式，往往会达到比较理想的效果。Scan 函数引入 IP 和端口通过套接字进行连接并打印相关信息，在函数中通过发送一个数据串从而获取到使用对应端口的服务回应的 banner。再介绍下 for 循环：使用内建 socket 模块建立一个 socket 连接：准备环境 ubuntu

（ip:192.168.213.20）主机 win10（ip:192.168.213.16）开着 ssh 服务 kali（ip:192.168.213.19），笔者使用的是 vim 编辑器，上面这个例子先引用 socket 模块然后调用 connect（）函数连接 ip 与端口。它会与 tcp 连接，并且这样就不会出现错误，一行很简单的代码就可以让程序继续执行下去。接下来使用 for 循环来写一个简单的端口扫描器：使用“try/except”循环来处理当 socket 连接的时候遇到端口关闭的错误，另外打印出来连接成功的端口以及对应的服务信息。下面创建一个自己指定的端口进行扫描，接下来笔者就可以尝试一下循环嵌套构。[1-2]

比如说，笔者知道在服务器的端口上运行着相应的服务，这些服务有其版本。笔者在服务器上执行端口扫描有两个目的：第一个是确定服务器上启用了哪些服务；第二个是确定使用哪个版本的服务。这两个目的都是缩小行动范围，提高其效率和准确性。例如，如果笔者知道主机上启用了端口 80，它很可能提供 Web 服务，笔者可以根据主机用于 Web 服务的语言（如 PHP）、框架（如 WordPress）和框架版本（如 4.8.2）进一步搜索相应的漏洞；笔者经常使用 SSH 远程登录服务器，例如，默认运行在端口 22 上，如果笔者扫描服务器打开此端口，服务器很可能会打开远程连接的功能，如果笔者获取 banner 获取其服务版本的 SSH 则存在一个版本的安全漏洞，因此有一个新的测试攻击点。接触网络技术的读者大概知道端口是什么，没有接触过的读者，经过下面的简单介绍也应该能够了解端口是什么。在网络上，笔者将使用各种服务（不是所有类型的网站，当然网站提供 Web 服务），例如浏览网站、发送电子邮件、使用 FTP 下载某些资源、使用 SSH 或 Telnet 连接到远程服务器等。这些服务可能都由同一台服务器提供。同一台服务器上有这么多服务，如何区分它们？端口用于区分和唯一标识它们，每个服务使用不同的端口，就像建筑物中的不同房间一样。笔者常用的 Web 服务默认使用端口 80；用于上传和下载文件的 FTP 使用端口 21；笔者通过域名或 IP 地址找到相应的服务器，然后通过端口号找到相应的服务。[3-4]

3 基于 for 循环的缩进扫描

为了实现这一目标，笔者将引入一个新概念，即 for 循环：请注意，在上面的代码片段中，for 循环的主体是缩进的。通常人们用 2 个空格或用制表符缩进，只要在整个剧本中保持一致就没关系。要制作简单的端口扫描程序，笔者将使用创建套接字连接的代码片段替换 print 语句。下面的代码显示了如何使用内置套接字模块建立套接字连接：上面笔者导入套接字模块并调用 connect（）函数连接到给定的 IP 地址和端口号。这将建立 TCP 连接（SYN/SYN-ACK/ACK），笔者实际上使用 send（）函数将数据发送到给定服务，并使用 recv（）打印响应。现在，如果端口未打开，socket 将抛出异常：这可以通过多种方式解决。现在笔者将使用一种非常简单的方法并使用“try/except”循环并传递异常：注意没有错误。现在让笔者结合所有这些概念并制作一个快速的循环端口扫描程序：上面笔者演示了“try/except”循环的基本用法，以便在端口关闭时传递 socket 抛出的异常。笔者还展示了如何利用带有“if”的基本条件语句，如果端口响应笔者的探测器，则仅尝试打印端口打开。创建端口扫描程序的另一种方法是定义一个希望用数组扫描的端口列表，然后遍历该数组：如果笔者想要一次处理多个主机，就要利用嵌套的 for 循环。这将涉及循环通过主机的外层 for 循环和将循环通过端口的内部 for 循环。下面是如何利用嵌套 for 循环来制作稍微复杂的扫描程序的基本示例：正如在输出中看到的那样，它会循环 hosts 数组并尝试 ports 数组中的每个端口，然后继续前进到下一个主机。对于最终端口扫描程序，可能希望将 print 语句修改为仅打印已打开的端口。在一天结束时，会发现 Nmap 仍然是端口扫描的更好选择，但笔者将在后面的文章中构建这些概念，以完成一些更实际的用例。我们可以花一些时间来探索插座模块“dir（socket）”中可用的各种功能。[5]

4 深度测试 TCP 连接扫描

TCP 连接扫描使用完整的三方握手来确定服务器或端口是否可用。笔者将脚本分为几个单独的步骤，首先输入主机名和以逗号分隔的端口列表并开始扫描，然后将主机名转换为 IPv4 地址，再与每个端口建立 TCP 连接并捕获目标端口应用程序的标题信息。首先，来看看 OptParse 模块。由于 OptParse 模块主要用于将命令参数传递给脚本，因此它使用预定义的选项来解析命令行参数。需要导入 OptParser 模块，初始化它，实例化 OptionParser 对象（带或不带参数），并向命令行添加选项。其次，笔者需要定制两个函数，每个函数一个。第一个功能是创建一个 socket 对象，将测试信息发送到端口，然后接收主机返回的信息并打印；第二个功能是将从主机参数获得的目标值转换为标准 xxx.xxx.xxx。XXX 表单，主要使用 socket gethostbyname 函数将域名值转换为四点基表单。最后，笔者在 main

函数中使用 OptParse 模块创建实例对象，确定当前主机和端口是否为空，如果为空，则打印实例对象。那么，笔者如何使用 Python 进行端口识别和扫描呢？[6]

比如说，在 Python 中，有一个名为 socket 的内置模块，该模块提供网络套接字操作，并包含以下函数以支持套接字连接的实现。要使用源地址创建链接并设置连接超时，笔者可以使用此模块的 socket（）函数创建一个 socket 对象以连接到服务器的指定端口，如下代码所示：笔者使用了 socket（）套接字模块的功能是建立到IP地址 192.168.223.152 端口 22 的套接字连接，然后关闭连接。当然，无法成功连接。运行此代码甚至没有输出信息：如何获取套接字连接的输出信息？套接字对象中的方法 Recv（）可能会有所帮助：它从套接字连接返回指定大小的字节。让笔者测试这个方法：笔者在这里测试的 IP 地址是 Intranet 主机的 IP 地址。笔者使用 Metasploitable2 作为目标主机，Metasploit 是一个著名的渗透测试框架，提供了一个目标虚拟机，它是一个易受攻击的 Linux 虚拟机。该虚拟机可用于安全培训、测试安全工具和实践常见的渗透测试技术。运行代码，笔者得到套接字返回的第一个 1024 字节（通常是服务消息的标题）：从返回的标题消息中，笔者知道服务器的端口 22 是打开的，服务版本是 Ubuntu 上的 OpensSH4.7。由此可见，使用 recv（）方法，笔者可以获得有关端口服务的最基本信息，但是如果相应的端口未启用或具有其他限制，则可能会报告错误。

5 秘密扫描与间接扫描

端口扫描的第一步是将探测数据包发送到目标主机的 TCP/UDP 端口，然后确定端口是否在对方响应后打开。由于不同的网络环境以及操作系统没有响应连接请求，为确保扫描的准确性和速度，通常在端口扫描中支持多种扫描方式。由于该技术不包含标准的 TCP3 次握手协议的任何部分，因此无法记录，从而比 SYN 扫描隐蔽得多。另外，FIN 数据包能够通过只监测 SYN 包的包过滤器。秘密扫描技术使用 FIN 数据包来探听端口。当 FIN 数据包到达关闭的端口，数据包会丢掉，并且会返回一个 RST 数据包；当 FIN 数据包到达打开的端口，数据包也丢掉，但不返回 RST。Xmas 和 l Null 扫描是秘密扫描的 2 个变种。Xmas 扫描打开 FIN、URG 和 PUISH 标记，而 Null 扫描关闭所有标记。这些组合的目的是为了通过 FIN 标记监测器的过滤。跟 SYN 扫描类似，秘密扫描也需要自己构造 IP 包。间接扫描是利用第三方的 IP（欺骗主机）来隐藏真正扫描者的IP。间接扫描时，使用第三方 IP 地址（主机错误）来隐藏实际扫描仪的 IP 地址。

6 使用 Pyhton 进行端口扫描

计算机之间通过端口和 IP 地址进行通信的方式称为 SOCKET 通信，HTTP、FTP、DNS 等都是通过 SOCKET 通信方式实现的。这一通信方式中提供服务的一方被称之为 SOCKET 服务端，而被服务的一端则被称之为SOCKET客户端。Python可以直接调用SOCKET对象，通过 socket 套接字向目标主机的端口发送 TCP connect（）请求，如果目标主机上的指定端口处于侦听状态，那么便可以建立连接请求，但相反若该目标主机并未开放端口，则 connect（）操作失败，产生异常。

在进行扫描之前首先需要建立 TCP 连接，关键就是要利用 socket 对象 tcps 对目标主机进行 connect 连接，connect（）函数只能接受一个参数。如果目标主机指定的端口并未开启，则需要返回到 connect refuse 结果，并查收能 connect 操作失败的指示。在实现程序的过程中，需要定义一个连接判断函数，这一函数的主要功能是根据 IP 和 Port 参数指定的 IP 地址和端口号，连接目标主机的相应端口，若显示连接失败或者直接不显示，则表明主机的端口并未开放。

7 总结

在探测一个区域的网络布局时，端口扫描是黑客经常使用的一种辅助攻击手段，扫描主机端口并不能直接对主机直接造成危害，但是可以获取到主机的一些信息，借此来对目标主机进行分析，为进一步入侵做好准备。这是每一位计算机从业人员都需要格外注意的细节。

参考文献：

[1] 杨迎 . 基于 Python 语言的网络传输层 UDP 协议攻击性行为研究 [J]. 数字技术与应用 ,2021,39(02):196-198.
[2] 孟彬 , 智云垒 , 钟翡 . 基于 Python 的端口扫描技术研究 [J]. 网络安全技术与应用 ,2021(01):42-43.
[3] 周耀鹏 , 王晖 , 陈嘉伦 . 基于 Python 的网络空间安全扫描系统的设计与实现 [J]. 电脑知识与技术 ,2020,16(31):59-61.
[4] 侯美静 . 基于智能爬行算法的网络扫描技术研究及实现 [D]. 西安电子科技大学 ,2018.
[5] 朱欣嘉 . 利用网络回溯分析技术进行端口扫描行为分析 [J]. 网络安全技术与应用 ,2013(04):5-6.
[6] 李树军 . 反射式 TCP 端口扫描技术的研究 [J]. 网络安全技术与应用 ,2006(09):29-30,37.