一种基于 TCP/IP 协议栈的操作系统识别技术

沙 超[1], 陈云芳[1,2]
(1. 南京邮电大学 计算机学院, 江苏 南京 210003;
2. 苏州大学 计算机学院, 江苏 苏州 215006)

摘 要: 远程主机的操作系统识别是获取系统基本特征的重要方法, 也是网络攻击与攻击防范的前奏。文中重点阐述了各种扫描方式以及它们在指纹识别中所起的作用, 讨论了通过使用 TCP/IP 协议栈的指纹特征来识别远程主机的操作系统的方法并给出了一个设计实例。文中描述了系统设计的具体结构、工作流程, 以及其模块的功能划分等。

关键词: 操作系统识别; TCP/IP 协议栈; 堆栈指纹
中图分类号: TP393.08 文献标识码: A 文章编号: 1673-629X(2006)10-0125-03

Research and Implementation of Detection of OS Based on TCP/IP Protocol Stack

SHA Chao[1], CHEN Yun-fang[1,2]
(1. Computer College of Nanjing University of Post and Telecommunication, Nanjing 210003, China;
2. Computer College, Soochow University, Suzhou 215006, China)

Abstract: OS detection of remote host is a very important method to get the basic feature of the system. In this paper, illustrate various modes of scanning and their impact in fingerprint identification. Discuss a technology on OS detection of remote system via TCP/IP protocol stack, and implement an instance based on it. Also discribe the ensemble structure of system and the procedure of its working and of course including the module demarcating.

Key words: OS detection; TCP/IP stack; stack fingerprinting

0 引 言

网络技术的飞速发展带来了巨大的安全隐患[1]。Internet 互连所带来的安全性问题日益受到重视。入侵者攻击某一网络系统, 几乎都是通过挖掘操作系统和应用服务程序的弱点或缺陷来实现。因此, 从入侵检测的角度出发, 对系统的安全扫描必不可少。如果说防火墙和网络监控系统是被动的防御手段, 那么安全扫描就是一种主动的防范措施, 可以有效避免黑客攻击行为, 做到防患于未然[2]。

一般而言, 对系统的攻击是从操作系统的识别开始[3]。只有了解了对方操作系统的具体类型, 黑客才能采取相应的攻击手段。如果在系统识别环节就出错, 那么攻击很可能就无法进行。因此无论对于入侵者还是网络管理者, 操作系统的识别都尤为重要。

文中研究的是基于 TCP/IP 协议栈的操作系统指纹识别技术。首先通过网络数据包的收发来扫描目标主机, 获得判断目标机器操作系统类型的"指纹"依据, 然后根据获得的指纹特征进行比较, 从而识别目标主机的操作系统类型。

1 指纹识别技术原理

1.1 识别远程主机 OS 的一般方法

随着网际互联技术的发展, 远程主机 OS 的识别方式越来越多, 也更加隐蔽。一般说来有以下几种识别目标主机操作系统的方法:

1) 根据 ICMP 数据包的 TTL 值。
2) 通过获取应用程序旗标。
3) 利用 TCP/IP 协议栈指纹鉴别。
4) WEB 查点+综合分析[3]。

各个方法的原理与优缺点如表 1 所示。

1.2 协议栈指纹识别技术

生物学上的指纹自动识别系统利用人的指纹特征, 通过特殊的光电扫描和计算机图像处理技术, 对活体指纹进行采集、分析和对比, 自动、迅速、准确地鉴别出个人身份。基于 TCP/IP 协议栈的操作系统指纹识别与之类似。由于 TCP/IP 协议技术只是在 RFC[4~6] 文档中描述, 并没有一个统一的行业标准, 造成了各个操作系统在 TCP/IP 协议实现上的不同。正是这些差别产生了区分操作系统的

收稿日期: 2005-12-14
作者简介: 沙 超(1983-), 男, 江苏南京人, 硕士研究生, 研究方向为计算机网络安全、传感器网络等。

依据。通过对不同的操作系统的 TCP/IP 协议栈存在的细微差别来鉴别操作系统的技术就是操作系统的指纹识别。因此,要想正确识别操作系统的类型,必须做到至少两点:首先,要获得可靠的系统指纹,以构建比较有效的指纹数据库。因此,前期必须获得尽可能准确的,特别是关于操作系统和指纹对应关系的信息;其次,要能够对这些指纹进行分析,获得一个与相应操作系统的对应关系。

表 1 识别目标主机操作系统的常用方法

方法	原理	优点	不足
TTL 值	向目标主机发送 ICMP 报文,根据返回报文中的 TTL 值识别系统类型	简单、易实行	不精确且不可靠。TTL 值可手动配置
应用程序旗标	直接对主机进行 telnet 或者 ftp 连接,缺省将得到远程主机返回的 Banner 信息,便于辅助确认远程 OS	方便、快捷	旗标信息可方便地被修改,可靠性不高
TCP/IP 协议栈指纹	通过数据包的收发,对不同的操作系统的 TCP/IP 协议栈存在的细微差别来鉴别操作系统。即文中提到的方法	准确性和可靠性高	主动识别方式易被远程系统察觉,指纹库需更新
WEB脚本	若目标主机是一台 WEB 主机,可直接通过 WWW 访问。通过在 WEB 页面上搜集的信息也可以粗略判断 OS	不需专门操作,只需搜集站点信息	可靠性和精确度都不高

协议栈指纹识别技术分两类:一类是主动识别,另一类则是被动识别[7]。主动协议栈指纹识别需要主动向目标主机发送数据包,但这些数据包在网络流量中比较惹人注目,因为正常的网络传输不会出现这样顺序的包,因此比较容易被发现,为了隐秘地识别远程 OS,就需要使用被动协议栈指纹识别。被动协议栈指纹识别在原理上和主动协议栈指纹识别相似,但不是向目标主机发送数据包,而是被动监测网络通信,从正常的网络数据包中确定目标主机所用的操作系统类型。如根据 TCP/IP 会话中的几个属性:TTL,窗口大小,DF,TOS 等与属性库比较以判断远程 OS 类型。文中主要讨论的是主动识别技术。

最为经典的栈指纹辨识技术由 Fyodor 提出[3],通过构造并发送不同类型的数据包,来收集并分析远端操作系统的响应,从而区别不同的 TCP/IP 协议栈。这一期间,还出现了基于初始序列号(ISN)采样的指纹辨识技术[3]。

1.3 系统扫描方式

如上所述,不同的操作系统有不同的指纹特征,而这些指纹特征的来源,就是它们针对各种扫描方式的不同反应。对远程主机的扫描方式主要有:

● 全连接扫描:TCP 端口扫描的基础。有 TCP connect 和 TCP 反向 ident 扫描等。

最大优点是不需要任何权限并且速度快[8]。

● FIN 探测:发送 FIN 包到打开端口,等待回应。RFC793[3] 定义的标准行为是不响应,但 Windows,CISCO,HP/UX 和 IRIX 等会回应一个 RESET 包。

● 隐蔽扫描:很少会在目标机上留下记录,3 次握手的过程从来都不会完全实现。

● 标记位探测:在 SYN 包 TCP 头中设置未定义的 TCP 标记(64 或 128)。低于 2.0.35 版本的 Linux 会在回应包中保持此标记,其它 OS 几乎没有这个问题。

● TCP ISN 取样:从操作系统对连接请求的回应中寻找 TCP 连接初始化序列号的特征。目前可区分的有传统的 64k、随机增加、真正随机。

● TCP 初始窗口:检查返回数据包的窗口大小。某些系统使用比较特殊的窗口值(如 AIX 使用 0x3F25,而 Microsoft 使用的窗口值总是 0x402E)。

● ACK 值:向一个关闭的 TCP 端口发送一个 FIN|PSH|URG 包,许多 OS 会将 ACK 值设置为 ISN 值,但 Windows 会设置为 seq+1。

● 服务类型 TOS:对于 ICMP 的"端口不可到达"信息,经过对返回包的 TOS 值的检查,几乎所有 OS 使用的是 ICMP 错误类型 0,而 Linux 使用的值是 0xC0。

2 系统设计实例

2.1 系统的开发环境

系统主要是基于 TCP/IP 协议族进行网络数据包的收发,所以需要对底层网络硬件进行控制,文中使用 VC++进行开发。Microsoft 制定了一套 Windows 下的网络编程接口,即 Windows Sockets 规范,它是一套开放的、支持多种协议的 Windows 下的网络编程接口。在 VC++中定义了 Winsock 类,如 CAsyncSocket 类和派生于 CAsyncSocket 的 CSocket 类,它们简单易用,可以使用这些类来实现自己的网络程序,而本系统采用了 WinPcap 提供的底层的 API 函数实现简单的 Winsock 网络应用程式设计的方法。

使用 WinPcap 函数库实现底层网络的数据包收发工作。WinPcap 是 UNIX 下的 LibPcap 移植到 Windows 下的产物,是一个免费开源项目。主要功能就是捕获原始数据包(包括在共享网络上各主机发送和接收的以及相互之间交换的数据包),以及在数据包发往应用程序之前,按照自定义的规则将某些特殊的数据包过滤掉。它能够为 Win32 应用程序提供访问网络底层的能力,并能以很高的效率进行网络操作。

WinPcap 提供了包括数据包捕获、过滤、发送和导入导出文件等一系列强大的功能,它主要由 3 个部分组成:

* NPF(Netgroup Packet Filter):一个虚拟设备驱动程序文件。它的功能是过滤数据包,并把这些数据包原封不动地传给用户态模块。

* packet.dll:为 Win32 平台提供了一个公共的接口。

* wpcap.dll:是不依赖于操作系统的。它提供了更加高层、抽象的函数。

2.2 总体设计

为了实现远程主机的 OS 识别,系统需要实现于一个 Windows 操作系统的 TCP/IP 网络环境中,其总体结构如图 1 所示,其中运行 Windows 的工作站作为发起扫描的主机(称为扫描主机),在其上运行扫描模块、捕获模块和控制平台,并建有操作系统识别模块。扫描模块直接从扫描主机上通过网络以其他机器为对象(称为目标主机,其上运行的操作系统可以是 UNIX,Linux,Windows 2000/

NT 等) 进行扫描, 按照用户的要求对用户指定的 IP 和端口发送各种数据包。捕获模块则同时捕获被扫描主机回送给系统的数据包, 并取出其中的相应字段。识别模块负责数据包的解析与识别, 通过回送数据包的各个首部字段, 查找相应的操作系统指纹, 给出判断结果。控制平台则提供一个人机交互的界面。

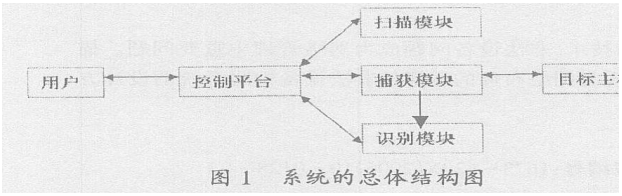

图 1　系统的总体结构图

2.3　系统流程及模块划分

指纹识别系统内部传递的主要就是捕获到的数据包的各个首部字段, 以及发送和捕获的相关数据。根据对系统输入输出信息的分析, 系统流程大致如图 2 所示。

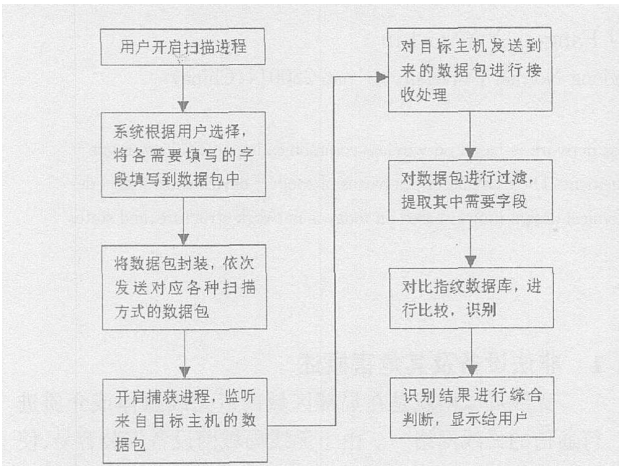

图 2　系统的工作流程

根据系统的实现原理以及工作流程分析, 设计了如下几个模块:

1) 主模块。

作为用户界面, 用于接受用户的输入信息, 同时将扫描过程和操作系统识别结果显示给用户。它负责调度处理其下的所有模块。包括发送模块、捕获模块和识别模块。

2) 发送模块。

本模块接受来自主模块的调用, 负责数据包的发送。主模块将所有与发送有关的信息传递给发送模块, 然后开始发送。一般而言, 发送模块与捕获模块是相对独立的, 但是发送数据包的正确与否与效率的高低将会直接影响到捕获的质量。

3) 捕获模块。

该模块和发送模块几乎是并行开始。因为网络内(尤其在局域网的实测环境下), 主机之间的相互通信非常迅速, 所以, 本系统的设计思路便是主模块在接到用户的"开始扫描"的指令以后, 同时调用捕获和识别模块。当然, 主模块同样也需要将所有的相关信息传递给捕获模块。

4) 识别模块。

本模块负责最终的操作系统的识别, 它是在捕获完成以后开始运行的。所以捕获模块需要将捕获并经过解析后的数据包首部的相应字段传递给本模块。然后通过堆栈指纹特征进行分析、比较和识别, 最终将识别结果送给主模块并显示给用户。

系统最终的运行结果如图 3 所示。

3　结束语

文中主要介绍了判断远程主机操作系统类型的指纹识别技术原理、方法及实现。并在此基础上初步设计了一个指纹识别系统。对于操作系统的类型判断, 提供了理论上的依据, 同时为网络体系的安全性提供了一个有效的检测与保障方法。随着操作系统版本的不断更新和系统漏洞的不断出现, 系统识别技术必将向着多样化、智能化的方向发展。如何增加更多的识别方法, 如何有效综合各种识别手段, 以及主被动识别的综合利用, 都是本课题今后的研究内容。

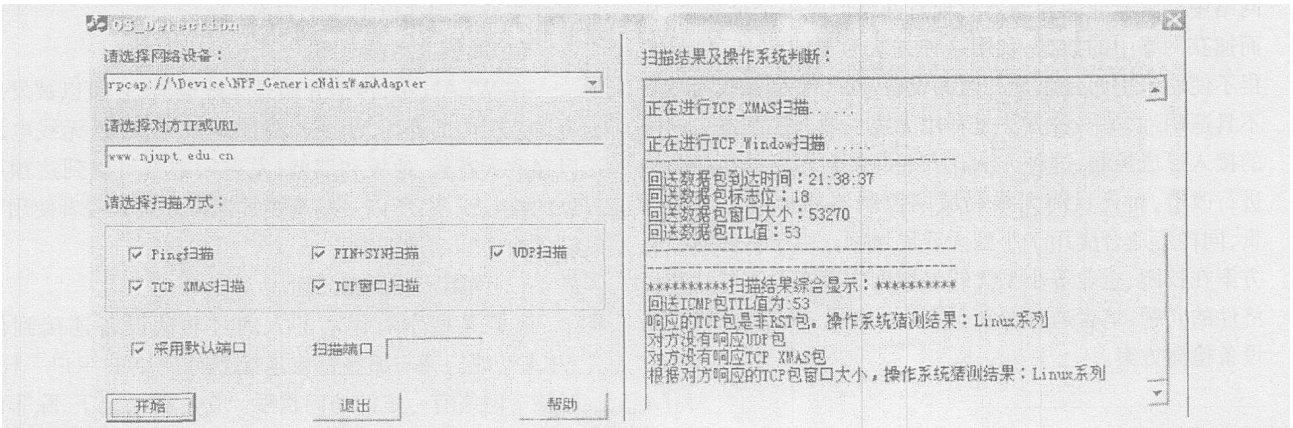

图 3　系统的运行结果图

参考文献:

[1]　沈金龙. 计算机通信与网络[M] . 北京: 北京邮电大学出版社, 2002.

[2]　谢希仁. 计算机网络[M] . 大连: 大连理工大学出版社, 2000.

(下转第 130 页)

无线网络接口卡有两种工作模式:一种是正常模式;一种是射频监听模式。在正常模式下,一个合法的网络接口卡只响应这样的两种数据帧:

(1)帧的目的地址与本网络接口卡 MAC 地址相同。

(2)帧的目的地址与广播地址相同。

按照这种方式,每一台机器都只能正常工作:发送和接收属于自己的数据。

然而如果改变网络接口卡的工作模式,使之工作在射频监听模式,那么它就能接收网络上的所有数据包[4]。我们就是利用这种摸式,捕获设备所在的基本服务集(Basic Service Set, BSS)中的所有数据包。并通过上层分析器对捕获到的数据进行分析,从而检测非法设备。使用的是 3Com 公司的 3CRWE771－el 型号无线网卡,它是基于 Prism2 的,所以可借助 libpcap 函数库实现数据包捕获。其程序流程如图 2 所示。

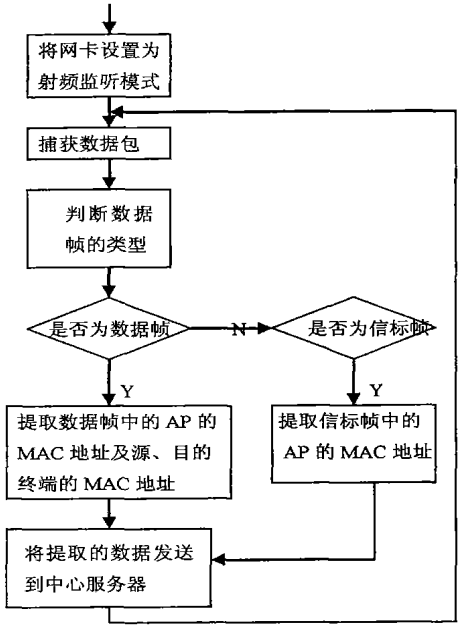

图 2 客户端数据流程图

其中在 Linux 命令行下把无线网卡置为射频监听模式的命令是:

ifconfig wlan0 promisc up

wlanctl－ng wlan0 lnxreq－wlansniff channel＝1 enable＝true prism－header＝true

3.2 中心服务器的检测

为了能够检测非法设备,建立一个较大的数组,将网络中合法的 AP 和终端设备的 MAC 地址通过 Hash 函数映射到数组的一定位置中。为此构建了一个数组 MAC[MAX_NUM]。在数组中查找客户端发送来的 MAC 地址,如果 AP 和终端设备的 MAC 在数组中存在,说明是合法的设备,否则是非法的并发出相应的警告提示。

4 实验分析

按如上设计描述,利用 C 语言在 Linux 系统下实现了该方案[5]。首先在一台 PC 机上安装具探测功能的 3Com 公司的 3CRWE771－el 型号无线网卡,并将其置为射频监听模式,然后利用一个非法 PDA 在无线网络中与其它设备进行通信。实验发现,在室内当 PDA 与探测器在约 70m 以内时,能够准确检测出该非法设备的存在。通过实验分析可知,通过增加探测器的数量,能够减少漏检,又由于一般小型无线局域网覆盖范围有限,所以并不需要太多的探测器。

5 总 结

针对目前非法设备检测工具不适用于小型无线局域网,利用无线网卡价格低廉而且能够捕获数据包的功能,设计了这样一种方案,并且通过在实验室无线局域网中的应用得出该方案能够达到预计的目的。该方案主要有以下优点:采用分布式的结构,便于数据采集;不需改变原有的网络拓扑结构;可以充分利用现有的设备,经济成本较低;可扩展性好。由于其便于实现,故具有一定的推广价值。随着对网络安全要求的不断提高,今后的研究方向是如何进一步提高该系统的性能。

参考文献:

[1] Mobile & Wireless Technology Review WLAN Security Monitors Watching the Waves[EB/OL]. http://www.nwc.com/story/singlePageFormat.jhtml?articleID＝18200309, 2004.

[2] 刘乃安. 无线局域网(WLAN)原理、技术与应用[M]. 西安:西安电子科技大学出版社, 2004.

[3] IEEE Standard for Information technology－telecommunications and Information Exchange Between Systems－local and Metropolitan Networks Specific Requirements－part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHZ Band[S]. Copyright 2001 Wildpackets, Inc. 2001.

[4] 贺龙涛,方滨兴,云晓春. 网络监听与反监听[J]. 计算机工程与应用, 2001(18): 20－21.

[5] 张 威. Linux 网络编程教程[M]. 北京:北京希望电子出版社, 2002.

(上接第 127 页)

[3] Fyodor. Remote OS detection via TCP/IP Stack FingerPrinting[EB/OL]. http://www.insecure.org/nmap/nmap－fingerprinting－article.html., 1998－10－18.

[4] Postel J. RFC 792[S]. 1981.

[5] del Rey M. RFC 793[S]. 1981.

[6] Braden R. RFC 1122[S]. 1989.

[7] Spangler R. Analysis of Remote Active Operating System Fingerprinting Tools[Z]. University of Wisconsin, 2003.

[8] 王轶俊,薛 质. 基于 TCP/IP 协议栈指纹辩识的远程操作系统探测[J]. 计算机工程, 2004, 30(18): 7－9.