文章编号：1006-2475(2003)11-0021-03

操作系统类型识别方法

林天峰

（温州职业技术学院计算机系，浙江 温州 325000）

摘要：回顾了各种传统的操作系统识别方法，并详细介绍了一种新出现的基于超时分析的识别技术。

关键词：操作系统；指纹识别；时间分析；协议

中图分类号：TP316　　文献标识码：A

Method of Detecting Operating System

LIN Tian-feng

(Dept. of Computer, Wenzhou Professional Technology College, Wenzhou 325000, China)

Abstract: All kinds of traditional methods of detecting host operating systems are reviewed and a new detection technique is introduced in detail.

Key words: operating system; fingerprinting; temporal analysis; protocol

0 引　言

随着计算机网络的普及，通过网络完成的各种业务越来越多，网络安全问题也越来越引起人们的重视。黑客在攻击网络前，总要事先收集有关信息，判断网络服务器的各种特征，从而找出漏洞，再实施攻击。其中，提供各种服务的主机所运行的操作系统类型及版本是最基本的、也是黑客们最希望得到的信息。因为很多的已知漏洞就是由操作系统直接造成的，而且知道了操作系统的类型，还可以大致判断这台主机所提供的服务的特点。因此，网络管理员应了解各种操作系统类型识别技术，从而有效地配置路由器、网关等安全设备及 IDS（入侵检测系统），以保证网络的安全运行。

1 传统的操作系统类型识别方法

按照获取信息的方式来分，操作系统识别方法可分为主动式和被动式两种。主动式是指客户机主动向远程主机发送信息，远程主机一般要对这些信息作出反应，回复一些信息，发送者再对这些返回的信息进行分析，就有可能得知远程主机的操作系统类型。这些信息可以是通过正常的网络程序如 Telnet、FTP 等与主机交互的信息，也可以是一些经过精心构造、正常的或残缺的数据报文。主动式识别方法还可分为标识攫取和 TCP/IP 栈探询两种方式。被动式是指通过 Sniffer 等网络嗅探工具收集流经网络的数据报文，再从这些数据报文中得到主机操作系统信息，它的分析方法和主动式基本上是一样的。

标识攫取方式是指用户通过客户端程序访问服务器，在和服务器正常的交互过程中根据服务器返回的提示或一些正常的操作判别操作系统类型。例如，通过以下 Telnet 程序的交互过程，用户可以了解主机操作系统的类型是 Red Hat Linux 7.2。

```
C:\> telnet noname.nowhere
Red Hat Linux release 7.2 (Enigma)
Kernel 2.4.7-10 on an i686
login:
```

还有一种标识攫取方式是从主机上得到一个二进制可执行文件，再用 Unix 的 file 命令测试它的类型，有时也可以从提示中得到操作系统的信息。如：

```
Unix: ~ $ ftp target anonymous ... get /bin/ls...
Unix: ~ $ file ./ls
./ls: executable (RISC System/6000 V3.1) or obj module
```

标识攫取方式是通过正常的交互过程来获取信息的，因此不会受到防火墙的干扰及被 IDS 系统察觉，但它一般要以手工的方式进行，因此效率较低，而且有可能被网络管理员通过修改提示信息所蒙蔽。

TCP/IP 栈探询方式利用不同的操作系统在实现 TCP/IP 协议时的细微差别来识别操作系统类型，它能够以很高的效率迅速确定远程操作系统的类型。按

收稿日期：2003-02-17

作者简介：林天峰（1970-），男，浙江温州人，温州职业技术学院计算机系讲师，硕士研究生，研究方向：计算机网络安全。

道理讲,各个厂商在操作系统中实现 TCP/IP 协议时都要遵循相应的 RFC 标准,它们之间应该是一样的,但由于各种原因,不同类型、不同版本的操作系统其 TCP/IP 实现还是有细微的差别,这些原因包括:

(1)有些 RFC 标准存在可选部分,以致有些操作系统支持这些可选部分,而有些不支持;

(2)RFC 标准经常更新,使得有些操作系统实现的是新标准,有些实现的还是旧标准;

(3)有些提示信息内容 RFC 未做规定,由厂商自己决定;

(4)有些厂商对 RFC 标准的理解存在偏差;

(5)有些厂商故意加一些自己特有的标志;

(6)有些 TCP/IP 的实现代码本身出错。

这些不同操作系统在实现 TCP/IP 时的具体差别也称为指纹,传统上,可用于识别操作系统的指纹主要有以下一些。

当向服务器的一个开放端口发送一个 FIN 报文时,按照 RFC793 规范,操作系统应不予响应,但有部分操作系统却要响应一个 RESET 报文。

TCP 头部有 6 个未定义的位,一般要置为 0,但客户端发送 SYN 报文时如果其中两位(左数第 7,8 位)置为 1,则低于 2.0.35 版本的 Linux 内核会在回应报文中保持这个标记,而其它操作系统则没有这个问题。

每个 TCP 报文都要包含一个序号,不同操作系统产生初始报文序号的方法是不一样的,因此,根据初始序号的特征可以判别操作系统的种类。

TCP 头部包含一个 16 位窗口大小的域,该域的值会随操作系统类型有较为稳定的数值,客户端通过记录这些值可以得到有用的信息。

虽然在一般情况下,ACK 的值都是很标准的,但在某些特定情况下,从不同操作系统返回的报文其 ACK 的值还是有特征的,如向关闭的端口发送 FIN|PUSH|URG 报文,或向打开的端口发送 SYN|FIN|PUSH|URG 报文时。

TCP 协议还提供了很多的选项,并不是所有的操作系统都实现这些选项,通过探测各种操作系统对各种选项的反应,可以搜集很多有效的信息。

所有 ICMP 差错报文回复时都要包含引起差错的源报文的 IP 首部及一定长度的数据,根据 RFC792 标准,数据的长度是 8 个字节,而新的 RFC1122 标准允许达到 576 字节。但实际上,不同的操作系统实现时这个长度差异很大。另外,有些操作系统还会在处理过程中不正确地设置了 IP 首部。

有些操作系统根据 RFC1812 的建议对某些类型的错误信息发送频率作了限制,通过向高端的随机 UDP 端口发送成批的报文,并计算接收到"目标不可到达"报文的数量,可以判别某些类型的操作系统。

按照标准,ICMP 请求回显报文的代码域应该为 0,如果故意设成非 0 值,则有些操作系统回显时要改为 0,而有些操作系统不做修改。

以上指纹都是和 TCP 或 ICMP 协议有关。另外,通过分析 ICMP 回复报文的 IP 首部,也可以得到很多有价值的指纹信息,具体内容不再一一介绍,可参见文献[2~3]。

上述的指纹信息被广泛地应用于 NMAP、QueSo 等具有识别操作系统类型功能的安全工具中,为了提高正确识别率,这些工具需要综合使用上述指纹,但采用不同的指纹时对远程主机的要求是不一样的,采用有些指纹要求能访问主机开放的 TCP 端口,有些要求能访问关闭的 TCP 或 UDP 端口,有些要求主机能回复 ICMP 协议查询报文,如果主机外围存在防火墙,则上述条件不一定都能满足,这样就会影响正确率。另外,上述很多指纹在使用时经常要向主机发送特殊的报文,这样就很容易被 IDS 系统检测到。因此,除了充分利用现有的指纹外,人们还在不断地寻找高效的新指纹,以满足精确、不受防火墙和 IDS 系统干扰、对网络环境影响小、容易扩展等要求。下面再详细介绍一种具有这样一些特点的新方法。

2 基于超时分析的操作系统类型识别方法及实现

2.1 基本原理

TCP 是一种面向连接的协议,在通信双方传送数据前,要先经过三路握手协议建立连接。具体过程如图 1 所示,连接发起方(一般是客户端)先向对方(一般是服务器)的某一端口发送一个 SYN 报文,服务端操作系统收到该报文后,如果报文中指明的端口是开放的,则向客户端回复一个 SYN+ACK 报文,客户端收到回复报文后,再向服务端发送一个 ACK 报文,这样,连接的过程就完成了,双方就可以开始交换数据。在这个过程中,如果任何一个报文在转输的过程中丢失或未按期到达,都会引起发送方的重输。至于引起重输的超时值,RFC793 标准只给出了建议值,RFC2988 标准给出了一个与 RTT 有关的算法,但没有给出重发次数,由于连接刚发起时 RTT 的值无法确定,因此 RFC2988 给出了一个建议初始值 3 秒。然而各种操作系统的具体实现却是各种各样的,不仅初始值、重发次数不一样,有些操作系统最后还要发送一个 RST 报文。如果把各种操作系统重发 SYN+ACK 报文的时间、次数、是否要发 RST 报文等事先记录下来,做成一个特征数据库,再把实测值与特征数据库进行比较,则可判定操作系统的类型及版本。

2.2 实现方法

如图 2 所示,采用上述原理识别操作系统的程序由三部分组成:报文发送器、报文过滤器和报文监听器。报文发送器用于构造一个 SYN 报文发送给目标服务器,报文过滤器用于阻挡来自目标服务器的响应报文,否则的话,客户端操作系统在接收到响应报文后会自动向目标服务器发送 ACK 或 RST 报文,这样就无法让目标服务器重发报文。报文监听器通过报

文过滤器而不是操作系统直接接收目标服务器的报文, 然后计算收到这些报文的时间差。在 Unix 系统下, 用 Libpcap 和 Libdnet 库实现上述三个组件的功能是一个不错的选择。

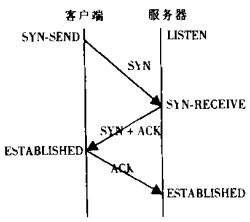

图 1　TCP 连接的建立

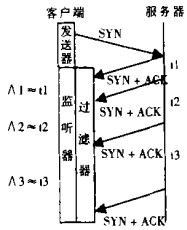

图 2　超时分析工作原理

由于报文可能要经过一些不稳定的网段, 在客户端测到时间间隔 Δ1, Δ2, …与发送方发送的时间间隔 t1, t2, …会有一些偏差, 因此这个时间间隔在与特征数据进行比较时应该允许有一定的误差, 也就是要取与实测值最接近的特征值所对应的操作系统。为了避免由于报文在 Internet 这样的环境中传输时引起的偏差过大导致判断失误, 可以在发送第一个 SYN 报文时采用时间戳选项, 使服务端在响应的报文中加上发送时间, 这样可以更精确地计算重发时间间隔。但是, 这种方法并不一定能奏效, 因为有些操作系统所注明的时间其增量可能过大。

在具体实现中, 为了提高效率及准确度, 可以对收到的报文再做一些互补性较强的传统指纹识别, 如根据 SYN+ACK 报文的 ACK、初始序号、窗口大小等值的特征及对一些 TCP 选项的回应, 可以更加准确有效地判断远程主机的操作系统类型。

2.3　特　点

基于超时分析的方法的主要优点是它只需用到一个开放的 TCP 端口。对于一台为 Internet 提供服务的主机来说, 它一般至少要开放一个 TCP 端口让外界访问, 为了提高安全性, 防火墙一般只允许与开放端口有关的报文通过, 其它协议及端口的报文一律予以过滤, 在这种情况下, 要求使用 ICMP、UDP 协议及要求使用关闭的 TCP 或 UDP 端口的传统识别方法就无法工作了, 而基于超时分析的方法能适应这种情况。另外, 采用基于超时分析的方法识别操作系统时, 客户端与服务器的交互完全是处于正常的模式内, 也不会产生残缺的报文, 因此它很难会被 IDS 发现及受到干扰。当然, 这种方法也有缺点, 如所需时间较长, 如果防火墙配置了 SYN 中继功能则无法工作等等, 因此也需要与其它方法相结合, 以便取长补短。

3　结束语

基于超时是一种较新的远程主机操作系统识别方法, 它实现简单, 是对传统方法的一种重要补充, 并且在传统方法不能奏效时, 使用它有时也能得到满意的结果。利用这种方法识别操作系统的工具包也已出现, 如开源项目 RING, 它可以独立运行, 也可以做为 NMAP 的一个补丁。

参考文献:

[1] W Richard Stevens. Unix 网络编程(第 1 卷)[M]. 北京: 清华大学出版社, 1999.

[2] Fyodor Yarochkin. Remote OS Detection Via TCP/IP Stack FingerPrinting[DB/OL]. http://www.insecure.org/nmap/nmap-fingerprinting-article.html, 2002-06-11.

[3] Ofir Arkin. A Remote Active OS Fingerprinting Tool Using ICMP[DB/OL]. http://www.sys-security.com/html/projects/X.html, 2001-08-14.

[4] Franck Veysset, Olivier Courtay, Olivier Heen. New Tool and Technique for Remote Operating System Fingerprinting[EB/OL]. http://www.intranode.com/site/techno/techno_articles.htm, 2002-03-01.

(上接第 15 页)

6　更好地完成审阅工作

充分有效地完成技术文档的审阅工作不仅会让外部的用户, 也会让内部的用户从中受益。但是, 经常会有技术人员认为做这样的工作是没有多大意义的, 那么, 作为管理者就面对着这样一种挑战, 就是要在整个的审阅过程中设置好优先次序从而保证为开发工作所做出的努力获得成功。下面是审阅工作的几个注意事项: (1)不要让他们从头至尾地审阅技术文档。(2)和技术文档的编写者一起确定哪些部分必须让开发设计人员进行审阅。(3)与他们一起利用大段的完整时间来审阅技术文档。

如果技术文档的审阅者时间表安排得很紧, 那么就给他提供一个具体的列表, 在其中明确哪些部分需要他审阅。并且保证让小组内的其他成员完成剩余部分的审阅工作。技术文档中与审阅者专业技术领域直接相关的部分绝对是需要他进行仔细审阅的。

7　结束语

软件产品的技术文档需要有一位专职的技术文档编写者来创作, 同时注意以上提到的五种方法, 并且切实把这些方法和技巧都应用到技术文档的编写中去, 这样才能够为公司和公司软件产品的用户们编写出随时可用并且易于阅读的优秀技术文档, 提升公司的信誉和品牌, 给公司带来丰厚的利润。

参考文献:

[1] 张海藩. 软件工程[M]. 北京: 清华大学出版社, 1998.

[2] 孙涌, 等. 现代软件工程[M]. 北京: 北京希望电子出版社, 2002.