浅谈操作系统指纹识别

无锡商业职业技术学院　蔡岚岚

[摘　要]本文介绍了操作系统指纹的概念、TCP数据包结构以及识别操作系统指纹的方法。

[关键词]操作系统指纹　指纹识别　TCP数据包

作为网络操作系统，系统的指纹实际上来源于TCP/IP协议栈。TCP/IP协议栈技术只是在RFC文档中描述，并没有一个统一的行业标准，各个公司在编写应用于自己的操作系统的TCP/IP协议栈时，对RFC文档做出了不尽相同的诠释，造成了各个操作系统在TCP/IP协议的实现上有所不同。好比人类的指纹，每个普通人都有指纹，但是没有两个人的指纹是一模一样的。通过比较不同的操作系统的TCP/IP协议栈的细微差异，就可以判定操作系统类型及版本，这种方式也称为“指纹方法学”。

TCP/IP协议栈指纹以独特的方式解决了操作系统辨识的问题，再加上一般的网络管理员不会有意识地修改操作系统的网络堆栈参数，所以很多探测程序就充分组合这些不同的参数，判断出操作系统的类型与版本。例如nmap能够可靠地区分出Solaris 2.4、Solaris 2.5–2.5.1和Solaris 2.6，也能区分2.0.30、2.0.31–34或2.0.35版本的Linux内核。

1.TCP数据包结构

TCP/IP协议栈是操作系统指纹识别的基础，下面介绍一下TCP数据包结构。

TCP数据被封装在一个IP数据包中，如图1所示：

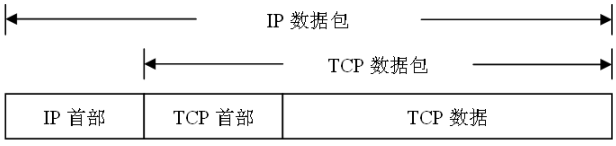

图1 TCP数据在IP数据包中的封装

TCP首部的数据格式。如果不计任选字段，它通常是20个字节。如图2所示：

16位源端口号									16位目的端口号
32位序号									
32位确认号									
4位首部长度	保留6位		URG	ACK	APSH	RST	SYN	FIN	16位窗口大小
16位校验和									16位紧急指针
选项									
数据									

图2 TCP包首部

在TCP首部中有6个标志比特。它们的意义分别为：

SYN：标志位用来建立连接，让连接双方同步序列号。如果SYN=1而ACK=0，则表示该数据包为连接请求，如果SYN=1而ACK=1则表示接受连接。

FIN：表示发送端已经没有数据要求传输了，希望释放连接。

RST：用来复位一个连接。RST标志置位的数据包称为复位包。一般情况下，如果TCP收到的一个分段明显不是属于该主机上的任何一个连接，则向远端发送一个复位包。

URG：为紧急数据标志。如果它为1，表示本数据包中包含紧急数据。此时紧急数据指针有效。

ACK：为确认标志位。如果为1，表示包中的确认号是有效的；否则，包中的确认号无效。

PSH：如果置位，接收端应尽快把数据传送给应用层。

TCP的流量控制由连接的每一端通过声明的窗口大小来提供。窗口大小为字节数，起始于确认序号字段指明的值，这个值是接收端正期望接收的字节。窗口大小是一个16bit字段，因而窗口大小最大为65535字节。

检验和覆盖了整个的TCP报文段：TCP首部和TCP数据。这是一个强制性的字段，一定是由发端计算和存储，并由收端进行验证。TCP检验和的计算和UDP检验和的计算相似，使用一个伪首部。

只有当URG标志置1时紧急指针才有效。紧急指针是一个正的偏移量，和序号字段中的值相加表示紧急数据最后一个字节的序号。TCP的紧急方式是发送端向另一端发送紧急数据的一种方式。

最常见的可选字段是最长报文大小，又称为MSS（Maximum Segment Size）。每个连接方通常都在通信的第一个报文段（为建立连接而设置SYN标志的那个段）中指明这个选项。它指明本端所能接收的最大程度的报文段。

2.指纹识别

网络协议栈指纹识别分为两类：一类是主动识别，另一类是被动识别。

（1）主动协议栈指纹识别

FIN探测：通过发送一个FIN数据包到一个打开的端口，并等待响应。

BOGUS标志探查：它原理是在一个SYN数据包TCP头中设置未定义的TCP“标记”。

TCP ISN取样：其原理是通过在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。

“无碎片”标记位：许多操作系统逐渐开始在它们发送的数据包中设置IP“不分片（无碎片）”位。这对于提高传输性能有好处。但并不是所有操作系统都有这个设置或许并不总是使用这个设置，因此通过留意这个标记位的设置可以收集到关于目标主机操作系统的更多有用信息。

TCP初始化“窗口”：就是检查返回数据包的“窗口”大小。有些新的探测器会记录确切的窗口值，因为该窗口随操作系统类型有较为稳定的数值。

ACK值：操作系统在ACK域值的实现也有所不同。

ICMP错误信息查询：有些操作系统根据RFC 1812的建议对某些类型的错误信息发送频率做了限制。

ICMP信息引用：RFC定义了一些ICMP错误信息格式。

ICMP错误信息回显完整性：机器必须根据接收到的数据包返回“端口不可到达”数据包。有些操作系统会在初始化处理过程中弄乱了请求头，这样当你接收到这种数据包时会出现不正常。

服务类型（TOS）：对于ICMP的“端口不可到达”信息，经过对返回包的服务类型（TOS）值的检查，几乎所有的操作系统使用的都是ICMP错误类型0，而Linux使用的值是0xC0。

片段（碎片）处理：不同操作系统在处理IP片段重叠时采用了不同的方式，从而能帮助确定操作系统类型。

（2）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识别相似，但是它不主动发送数据包，只是被动地捕获远程主机返回的包来分析其操作系统类型、版本。下面是4个常用的被动签名：

1）TTL：操作系统对出站信息包设置的存活时间；

2）Windows SIZE：操作系统设置的窗口大小；

3）DF：是否设置了不准分片位；

4）TOS：设置的服务类型。

在捕捉到一个数据包后，通过综合上述4个因素的分析，就能基本确定一个操作系统的类型。例如获得了一个局域网内数据包，它具有如下几个特征，即TTL为64；Windows Size为0x7D78；DF为The Don't Fragment bit is set；TOS为0x0。

将以上数据对照指纹数据库进行分析时，首先，发现TTL值为64，因为它是局域网内主机发过来的数据包，所以它是经过了0个路由器到达当前的主机，初始的TTL值为64。基于这个TTL值，查看数据库，发现有3种操作系统的TTL值为64，因此暂时还无法确定是哪一种操作系统。

然后比较窗口大小，获得的数据为0x7D78（十进制为32120），而在数据库中，发现这一窗口大小正是一个Linux系统所使用的，这时即可确定收到的包是从一个内核版本为2.2.x的Linux系统中发出的。

由于大多数系统都设置了DF位，因此这个签名提供的信息非常有限，然而它也能够使我们很容易地鉴别少数没有使用DF标识的系统，如SCO或OpenBSD。与DF类似，TOS提供的信息也同样很有限，通常是与上面几项结合使用。

因此，通过分析数据包头部这几个信息，基本上就能够确定操作系统的类型。

3.总结

本文从TCP/IP协议栈入手讨论了操作系统指纹的识别，以期对网络安全的研究提供一点帮助。