TCPdump抓包命令

参数列表

-A 以ASCII码方式显示每一个数据包(不会显示数据包中链路层头部信息). 在抓取包含网页数据的数据包时, 可方便查看数据(nt: 即Handy for capturing web pages).

-c count

tcpdump将在接受到count个数据包后退出.

-C file-size (nt: 此选项用于配合-w file 选项使用)

该选项使得tcpdump 在把原始数据包直接保存到文件中之前, 检查此文件大小是否超过file-size. 如果超过了, 将关闭此文件,另创一个文件继续用于原始数据包的记录. 新创建的文件名与-w 选项指定的文件名一致, 但文件名后多了一个数字.该数字会从1开始随着新创建文件的增多而增加. file-size的单位是百万字节(nt: 这里指1,000,000个字节,并非1,048,576个字节, 后者是以1024字节为1k, 1024k字节为1M计算所得, 即1M=1024 ＊ 1024 ＝ 1,048,576)

-d 以容易阅读的形式,在标准输出上打印出编排过的包匹配码, 随后tcpdump停止.(nt | rt: human readable, 容易阅读的,通常是指以ascii码来打印一些信息. compiled, 编排过的. packet-matching code, 包匹配码,含义未知, 需补充)

-dd 以C语言的形式打印出包匹配码.

-ddd 以十进制数的形式打印出包匹配码(会在包匹配码之前有一个附加的'count'前缀).

-D 打印系统中所有tcpdump可以在其上进行抓包的网络接口. 每一个接口会打印出数字编号, 相应的接口名字, 以及可能的一个网络接口描述. 其中网络接口名字和数字编号可以用在tcpdump 的-i flag 选项(nt: 把名字或数字代替flag), 来指定要在其上抓包的网络接口.

此选项在不支持接口列表命令的系统上很有用(nt: 比如, Windows 系统, 或缺乏 ifconfig -a 的UNIX系统); 接口的数字编号在windows 2000 或其后的系统中很有用, 因为这些系统上的接口名字比较复杂, 而不易使用.

如果tcpdump编译时所依赖的libpcap库太老,-D 选项不会被支持, 因为其中缺乏 pcap\_findalldevs()函数.

-e 每行的打印输出中将包括数据包的数据链路层头部信息

-E spi@ipaddr algo:secret,...

可通过spi@ipaddr algo:secret 来解密IPsec ESP包(nt | rt:IPsec Encapsulating Security Payload,IPsec 封装安全负载, IPsec可理解为, 一整套对ip数据包的加密协议, ESP 为整个IP 数据包或其中上层协议部分被加密后的数据,前者的工作模式称为隧道模式; 后者的工作模式称为传输模式 . 工作原理, 另需补充).

需要注意的是, 在终端启动tcpdump 时, 可以为IPv4 ESP packets 设置密钥(secret）.

可用于加密的算法包括des-cbc, 3des-cbc, blowfish-cbc, rc3-cbc, cast128-cbc, 或者没有(none).默认的是des-cbc(nt: des, Data Encryption Standard, 数据加密标准, 加密算法未知, 另需补充).secret 为用于ESP 的密钥, 使用ASCII 字符串方式表达. 如果以 0x 开头, 该密钥将以16进制方式读入.

该选项中ESP 的定义遵循RFC2406, 而不是 RFC1827. 并且, 此选项只是用来调试的, 不推荐以真实密钥(secret)来使用该选项, 因为这样不安全: 在命令行中输入的secret 可以被其他人通过ps 等命令查看到.

除了以上的语法格式(nt: 指spi@ipaddr algo:secret), 还可以在后面添加一个语法输入文件名字供tcpdump 使用(nt：即把spi@ipaddr algo:secret,... 中...换成一个语法文件名). 此文件在接受到第一个ESP　包时会打开此文件, 所以最好此时把赋予tcpdump 的一些特权取消(nt: 可理解为, 这样防范之后, 当该文件为恶意编写时,不至于造成过大损害).

-f 显示外部的IPv4 地址时(nt: foreign IPv4 addresses, 可理解为, 非本机ip地址), 采用数字方式而不是名字.(此选项是用来对付Sun公司的NIS服务器的缺陷(nt: NIS, 网络信息服务, tcpdump 显示外部地址的名字时会用到她提供的名称服务): 此NIS服务器在查询非本地地址名字时,常常会陷入无尽的查询循环).

由于对外部(foreign)IPv4地址的测试需要用到本地网络接口(nt: tcpdump 抓包时用到的接口)及其IPv4 地址和网络掩码. 如果此地址或网络掩码不可用, 或者此接口根本就没有设置相应网络地址和网络掩码(nt: linux 下的 'any' 网络接口就不需要设置地址和掩码, 不过此'any'接口可以收到系统中所有接口的数据包), 该选项不能正常工作.

-F file

使用file 文件作为过滤条件表达式的输入, 此时命令行上的输入将被忽略.

-i interface

指定tcpdump 需要监听的接口. 如果没有指定, tcpdump 会从系统接口列表中搜寻编号最小的已配置好的接口(不包括 loopback 接口).一但找到第一个符合条件的接口, 搜寻马上结束.

在采用2.2版本或之后版本内核的Linux 操作系统上, 'any' 这个虚拟网络接口可被用来接收所有网络接口上的数据包(nt: 这会包括目的是该网络接口的, 也包括目的不是该网络接口的). 需要注意的是如果真实网络接口不能工作在'混杂'模式(promiscuous)下,则无法在'any'这个虚拟的网络接口上抓取其数据包.

如果 -D 标志被指定, tcpdump会打印系统中的接口编号，而该编号就可用于此处的interface 参数.

-l 对标准输出进行行缓冲(nt: 使标准输出设备遇到一个换行符就马上把这行的内容打印出来).在需要同时观察抓包打印以及保存抓包记录的时候很有用. 比如, 可通过以下命令组合来达到此目的:

``tcpdump -l | tee dat'' 或者 ``tcpdump -l > dat & tail -f dat''.(nt: 前者使用tee来把tcpdump 的输出同时放到文件dat和标准输出中, 而后者通过重定向操作'>', 把tcpdump的输出放到dat 文件中, 同时通过tail把dat文件中的内容放到标准输出中)

-L 列出指定网络接口所支持的数据链路层的类型后退出.(nt: 指定接口通过-i 来指定)

-m module

通过module 指定的file 装载SMI MIB 模块(nt: SMI，Structure of Management Information, 管理信息结构MIB, Management Information Base, 管理信息库. 可理解为, 这两者用于SNMP(Simple Network Management Protoco)协议数据包的抓取. 具体SNMP 的工作原理未知, 另需补充).

此选项可多次使用, 从而为tcpdump 装载不同的MIB 模块.

-M secret 如果TCP 数据包(TCP segments)有TCP-MD5选项(在RFC 2385有相关描述), 则为其摘要的验证指定一个公共的密钥secret.

-n 不对地址(比如, 主机地址, 端口号)进行数字表示到名字表示的转换.

-N 不打印出host 的域名部分. 比如, 如果设置了此选现, tcpdump 将会打印'nic' 而不是 'nic.ddn.mil'.

-O 不启用进行包匹配时所用的优化代码. 当怀疑某些bug是由优化代码引起的, 此选项将很有用.

-p 一般情况下, 把网络接口设置为非'混杂'模式. 但必须注意 , 在特殊情况下此网络接口还是会以'混杂'模式来工作； 从而, '-p' 的设与不设, 不能当做以下选现的代名词:'ether host {local-hw-add}' 或 'ether broadcast'(nt: 前者表示只匹配以太网地址为host 的包, 后者表示匹配以太网地址为广播地址的数据包).

-q 快速(也许用'安静'更好?)打印输出. 即打印很少的协议相关信息, 从而输出行都比较简短.

-R 设定tcpdump 对 ESP/AH 数据包的解析按照 RFC1825而不是RFC1829(nt: AH, 认证头, ESP， 安全负载封装, 这两者会用在IP包的安全传输机制中). 如果此选项被设置, tcpdump 将不会打印出'禁止中继'域(nt: relay prevention field). 另外,由于ESP/AH规范中没有规定ESP/AH数据包必须拥有协议版本号域,所以tcpdump不能从收到的ESP/AH数据包中推导出协议版本号.

-r file

从文件file 中读取包数据. 如果file 字段为 '-' 符号, 则tcpdump 会从标准输入中读取包数据.

-S 打印TCP 数据包的顺序号时, 使用绝对的顺序号, 而不是相对的顺序号.(nt: 相对顺序号可理解为, 相对第一个TCP 包顺序号的差距,比如, 接受方收到第一个数据包的绝对顺序号为232323, 对于后来接收到的第2个,第3个数据包, tcpdump会打印其序列号为1, 2分别表示与第一个数据包的差距为1 和 2. 而如果此时-S 选项被设置, 对于后来接收到的第2个, 第3个数据包会打印出其绝对顺序号:232324, 232325).

-s snaplen

设置tcpdump的数据包抓取长度为snaplen, 如果不设置默认将会是68字节(而支持网络接口分接头(nt: NIT, 上文已有描述,可搜索'网络接口分接头'关键字找到那里)的SunOS系列操作系统中默认的也是最小值是96).68字节对于IP, ICMP(nt: Internet Control Message Protocol,因特网控制报文协议), TCP 以及 UDP 协议的报文已足够, 但对于名称服务(nt: 可理解为dns, nis等服务), NFS服务相关的数据包会产生包截短. 如果产生包截短这种情况, tcpdump的相应打印输出行中会出现''[|proto]''的标志（proto 实际会显示为被截短的数据包的相关协议层次). 需要注意的是, 采用长的抓取长度(nt: snaplen比较大), 会增加包的处理时间, 并且会减少tcpdump 可缓存的数据包的数量， 从而会导致数据包的丢失. 所以, 在能抓取我们想要的包的前提下, 抓取长度越小越好.把snaplen 设置为0 意味着让tcpdump自动选择合适的长度来抓取数据包.

-T type

强制tcpdump按type指定的协议所描述的包结构来分析收到的数据包. 目前已知的type 可取的协议为:

aodv (Ad-hoc On-demand Distance Vector protocol, 按需距离向量路由协议, 在Ad hoc(点对点模式)网络中使用),

cnfp (Cisco NetFlow protocol), rpc(Remote Procedure Call), rtp (Real-Time Applications protocol),

rtcp (Real-Time Applications con-trol protocol), snmp (Simple Network Management Protocol),

tftp (Trivial File Transfer Protocol, 碎文件协议), vat (Visual Audio Tool, 可用于在internet 上进行电

视电话会议的应用层协议), 以及wb (distributed White Board, 可用于网络会议的应用层协议).

-t 在每行输出中不打印时间戳

-tt 不对每行输出的时间进行格式处理(nt: 这种格式一眼可能看不出其含义, 如时间戳打印成1261798315)

-ttt tcpdump 输出时, 每两行打印之间会延迟一个段时间(以毫秒为单位)

-tttt 在每行打印的时间戳之前添加日期的打印

-u 打印出未加密的NFS 句柄(nt: handle可理解为NFS 中使用的文件句柄, 这将包括文件夹和文件夹中的文件)

-U 使得当tcpdump在使用-w 选项时, 其文件写入与包的保存同步.(nt: 即, 当每个数据包被保存时, 它将及时被写入文件中,而不是等文件的输出缓冲已满时才真正写入此文件)

-U 标志在老版本的libcap库(nt: tcpdump 所依赖的报文捕获库)上不起作用, 因为其中缺乏pcap\_cump\_flush()函数.

-v 当分析和打印的时候, 产生详细的输出. 比如, 包的生存时间, 标识, 总长度以及IP包的一些选项. 这也会打开一些附加的包完整性检测, 比如对IP或ICMP包头部的校验和.

-vv 产生比-v更详细的输出. 比如, NFS回应包中的附加域将会被打印, SMB数据包也会被完全解码.

-vvv 产生比-vv更详细的输出. 比如, telent 时所使用的SB, SE 选项将会被打印, 如果telnet同时使用的是图形界面,

其相应的图形选项将会以16进制的方式打印出来(nt: telnet 的SB,SE选项含义未知, 另需补充).

-w 把包数据直接写入文件而不进行分析和打印输出. 这些包数据可在随后通过-r 选项来重新读入并进行分析和打印.

-W filecount

此选项与-C 选项配合使用, 这将限制可打开的文件数目, 并且当文件数据超过这里设置的限制时, 依次循环替代之前的文件, 这相当于一个拥有filecount 个文件的文件缓冲池. 同时, 该选项会使得每个文件名的开头会出现足够多并用来占位的0, 这可以方便这些文件被正确的排序.

-x 当分析和打印时, tcpdump 会打印每个包的头部数据, 同时会以16进制打印出每个包的数据(但不包括连接层的头部).总共打印的数据大小不会超过整个数据包的大小与snaplen 中的最小值. 必须要注意的是, 如果高层协议数据没有snaplen 这么长,并且数据链路层(比如, Ethernet层)有填充数据, 则这些填充数据也会被打印.(nt: so for link layers that pad, 未能衔接理解和翻译, 需补充 )

-xx tcpdump 会打印每个包的头部数据, 同时会以16进制打印出每个包的数据, 其中包括数据链路层的头部.

-X 当分析和打印时, tcpdump 会打印每个包的头部数据, 同时会以16进制和ASCII码形式打印出每个包的数据(但不包括连接层的头部).这对于分析一些新协议的数据包很方便.

-XX 当分析和打印时, tcpdump 会打印每个包的头部数据, 同时会以16进制和ASCII码形式打印出每个包的数据, 其中包括数据链路层的头部.这对于分析一些新协议的数据包很方便.

-y datalinktype

设置tcpdump 只捕获数据链路层协议类型是datalinktype的数据包

-Z user

使tcpdump 放弃自己的超级权限(如果以root用户启动tcpdump, tcpdump将会有超级用户权限), 并把当前tcpdump的用户ID设置为user, 组ID设置为user首要所属组的ID(nt: tcpdump 此处可理解为tcpdump 运行之后对应的进程)

此选项也可在编译的时候被设置为默认打开.(nt: 此时user 的取值未知, 需补充)

Tcpdump返回数据的结构

以下分析tcpdume输出的数据，取以下这次输出作为样例：

13:37:15.996335 IP lsyp1002.xxx.xxxx.com.58939 > 10.101.10.7.15001: P 1:53(52) ack 1 win 46 <nop,nop,timestamp 870149289 3676470709>  
    0x0000:  4510 0068 df89 4000 4006 e0dd cbd1 fe80   E..h..@.@.......  
    0x0010:  3cbf 7307 e63b 3a99 be10 94c3 6225 198c   <.s..;:.....b%..  
    0x0020:  8018 002e 296c 0000 0101 080a 33dd 6ca9   ....)l......3.l.  
    0x0030:  db22 7db5 2457 322c 3533 2c31 3030 2c32   ."}.$W2,53,100,2  
    0x0040:  3030 2c30 3537 3138 3139 3332 3630 322c   00,057181932602,  
    0x0050:  3133 3635 3636 3435 3536 332c 3835 3032   13656645563,8502  
    0x0060:  3230 3838 2c23 0d0a                       2088,#..

红色部分，为IP数据报的首部，共20字节。   
蓝色部分，为TCP数据报的首部，共8\*32bit=32个字节。   
灰色部分，为TCP的DATA部分。

TCP的首部，没有DATA的长度信息，因为IP的首部已经包含了整个IP包的共长度，可以由下面的减法求出DATA的长度：

**length(TCP.DATA) = IP包的总长度 - IP首部长度 - TCP首部长度**