雙 使用WinPcap编程

2013年10月06日 19:47:03 阅读数:2731

创建一个使用 wpcap.dll 的应用程序

用 Microsoft Visual C++ 创建一个使用 wpcap.dll 的应用程序,需要按一下步骤:

- 在每一个使用了库的源程序中,将 pcap.h 头文件包含(include)进来。
- 如果你在程序中使用了WinPcap中提供给Win32平台的特有的函数, 记得在预处理中加入 WPCAP 的定义。
- 如果你的程序使用了WinPcap的远程捕获功能,那么在预处理定义中加入 HAVE_REMOTE 。 不要 直接把remote-ext.h直接加入到你的源文件中去。
- 设置VC++的链接器(Linker),把 wpcap.lib 库文件包含进来。 wpcap.lib 可以在WinPcap中找到。
- 设置VC++的链接器(Linker),把 ws2_32.lib 库文件包含进来。这个文件分布于C的编译器,并且包含了Windows的一些socket函数。本教程中的一些范例程序,会需要它。

记住以下几点:

- 要添加一个预处理定义,你需要打开 Project 菜单,选择 Settings ,然后选择 C/C++ 选项卡,在 General 类下,你必须在 Preprocessor Definitions 下的文本框中添加定义。
- 要在一个VC++6.0工程中,添加一个新的库,你必须打开 Project 菜单,选择 Settings ,然后选择 Link 选项卡,然后把新库的名字添加到 Object/Library modules 下的文本框中
- 要向VC++6.0中添加一个新的库所在的路径,你必须打开 Tool 菜单,选择 Options ,然后选择 Directories 选项卡,在 Show directories 下拉框中选择 Library files ,并且将新的路径添加到 Directories 中去
- 要向VC++6.0中添加一个新的包含文件所在的路径,你必须打开 Tool 菜单,选择 Options ,然后选择 Directories 选项卡,在 Show directories 下拉框中选择 Include files ,并且将新的路径添加到 Directories 中去

范例程序

我们一共了一些范例程序来显示WinPcap API的用法。这些程序的源代码,以及编译运行这些代码所需的所有文件,都可以在 Developer's Pack找到。作为教程,在这里,我们提供了浏览器式的代码:这样,在每个函数和变量之间的跳转会比较方便。更多完整的范例程序,请参阅 WinPcap 教程.

Packet Dump

这个程序会依据命令行参数,从网络适配器,或是从文件来读取数据包。如果没有提供源,那么程序会显示出所有可用的适配器,你可以选其中一个。当捕获过程开始,程序会打印数据包的时间戳,长度,原始内容。一旦被编译了,那么它将能运行于所有的Win32平台,当然,它也可以被编译成Unix平台的程序。

```
[cpp] 📳 📑
      * Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
2.
       * Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
3.
      * All rights reserved.
4.
6.
      * Redistribution and use in source and binary forms, with or without
       * modification, are permitted provided that the following conditions
8.
      * 1. Redistributions of source code must retain the above copyright
10.
11.
       * notice, this list of conditions and the following disclaimer.
12.
      * 2. Redistributions in binary form must reproduce the above copyright
       * notice, this list of conditions and the following disclaimer in the
13.
      * documentation and/or other materials provided with the distribution.
14.
       * 3. Neither the name of the Politecnico di Torino, CACE Technologies
15.
      \ \ \ast nor the names of its contributors may be used to endorse or promote
16.
17.
       * products derived from this software without specific prior written
      * permission.
18.
19.
      * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
20.
21.
       ^{st} "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
22.
      * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
23.
       * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
      * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
24.
25.
       * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
      * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
26.
       * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
27.
28.
      * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
29.
       * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
30.
31.
32.
33.
34.
35.
      #include <stdlib.h>
36.
      #include <stdio.h>
37.
```

```
38.
 39.
       // NOTE: remember to include WPCAP and HAVE_REMOTE among your
 40.
       // preprocessor definitions.
 41.
       11
 42.
 43.
       #include <pcap.h>
 44.
 45.
       #define LINE LEN 16
 46.
 47.
       main(int argc, char **argv)
 48.
       pcap_if_t *alldevs, *d;
 49.
 50.
       pcap_t *fp;
 51.
       u_int inum, i=0;
 52.
       char errbuf[PCAP_ERRBUF_SIZE];
 53.
       int res;
 54.
       struct pcap_pkthdr *header;
 55.
       const u char *pkt data;
 56.
 57.
           printf("pktdump ex: prints the packets of the network using WinPcap.\n");
       printf(" Usage: pktdump_ex [-s source]\n\n"
 58.
                      Examples:\n"
 59.
                      pktdump_ex -s file://c:/temp/file.acp\n"
 60.
 61.
                         pktdump\_ex -s \ rpcap://\Device\\NPF_{C8736017-F3C3-4373-94AC-9A34B7DAD998}\\n\n");
 62.
 63.
           if(argc < 3)
 64.
       {
 65.
               printf("\nNo adapter selected: printing the device list:\n");
 66.
 67.
               /* The user didn't provide a packet source: Retrieve the local device list */
 68.
               if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf) == -1)
 69.
 70.
                   fprintf(stderr,"Error in pcap_findalldevs_ex: %s\n", errbuf);
 71.
                   return -1:
 72.
 73.
 74.
               /* Print the list */
 75.
               for(d=alldevs; d; d=d->next)
 76.
 77.
                   printf("%d. %s\n
                                     ", ++i, d->name);
 78.
 79.
                   if (d->description)
 80.
                      printf(" (%s)\n", d->description);
 81.
 82.
                      printf(" (No description available)\n");
 83.
               }
 84.
               if (i==0)
 85.
 86.
               {
 87.
                   fprintf(stderr, "No interfaces found! Exiting.\n");
 88.
                   return -1;
 89.
 90.
 91.
               printf("Enter the interface number (1-%d):",i);
 92.
               scanf("%d", &inum);
 93.
 94.
               if (inum < 1 || inum > i)
 95.
               {
 96.
                   printf("\nInterface number out of range.\n");
 97.
 98.
                   /* Free the device list */
 99.
                   pcap freealldevs(alldevs);
100.
                   return -1;
101.
               }
102.
103.
               /* Jump to the selected adapter */
104.
               for (d=alldevs, i=0; i< inum-1; d=d->next, i++);
105.
106.
               /* Open the device */
107.
               if ( (fp= pcap_open(d->name,
108.
                                   100 /*snaplen*/,
109.
                                   PCAP_OPENFLAG_PROMISCUOUS /*flags*/,
                                   20 /*read timeout*/,
110.
111.
                                   NULL /* remote authentication */,
                                   errbuf)
112.
113.
                                   ) == NULL)
114.
115.
                   fprintf(stderr, "\nError opening adapter\n");
116.
                   return -1;
117.
               }
118.
       }
119.
           else
120.
121.
               // Do not check for the switch type ('-s')
122.
               if ( (fp= pcap_open(argv[2],
123.
                                   100 /*snaplen*/,
124.
                                   PCAP_OPENFLAG_PROMISCUOUS /*flags*/,
125.
                                   20 /*read timeout*/,
126.
                                   NULL /* remote authentication */,
127.
                                   errbuf)
                                   ) == NULL)
128.
```

```
130.
                   fprintf(stderr,"\nError opening source: %s\n", errbuf);
131.
                    return -1;
132.
133.
134.
135.
            /* Read the packets */
136.
           while((res = pcap_next_ex( fp, &header, &pkt_data)) >= 0)
137.
138.
139.
                if(res == 0)
                   /* Timeout elapsed */
140.
141.
                    continue:
142.
143.
                /* print pkt timestamp and pkt len */
144.
                printf("\$ld:\$ld\ (\$ld)\n",\ header->ts.tv\_sec,\ header->ts.tv\_usec,\ header->len)
145
146.
                /* Print the packet */
147.
                for (i=1; (i < header->caplen + 1 ) ; i++)
148.
149.
                    printf("%.2x ", pkt_data[i-1]);
                    if ( (i % LINE_LEN) == 0) printf("\n");
150.
151.
152.
153.
                printf("\n\n");
154.
155.
       if(res == -1)
156.
157.
158.
                fprintf(stderr, \ "Error \ reading \ the \ packets: \ \$s\n", \ pcap\_geterr(fp));
159.
                return -1;
160.
161.
162.
           return 0;
163. }
```

数据包过滤器

这是一个更加完整的使用libpcap的范例程序,它显示了如何创建和设置过滤器,如何把捕获保存到磁盘。这个程序在Win32和Unix平台下都能编译。Pcap_filter(pf.exe) 是一个通用的数据包过滤程序:它的输入参数有数据包的源(可以是物理接口,或是一个文件),过滤器和一个输出文件。它会从源获取数据包,并对它们进行过滤,如 果它们符合过滤器的要求,就把它们保存到输出文件,直到按下Ctrl+C,或者整个文件处理完毕。Pcap_filter不但可以根据一个特定的过滤器,来堆处理网络中的数据 ,而且可以从已经保存过的文件中提取数据包。输入和输出文件的格式都是libpcap兼容的格式,比如,WinDump,tcpdump和其他许多网络工具。

```
1.
2.
      * Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
 3.
       * Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
      * All rights reserved.
4.
      * Redistribution and use in source and binary forms, with or without
 6.
       * modification, are permitted provided that the following conditions
      * are met:
8.
9.
      * 1. Redistributions of source code must retain the above copyright
10.
       * notice, this list of conditions and the following disclaimer.
11.
      * 2. Redistributions in binary form must reproduce the above copyright
12.
13.
       * notice, this list of conditions and the following disclaimer in the
14.
      ^{st} documentation and/or other materials provided with the distribution.
15.
       * 3. Neither the name of the Politecnico di Torino, CACE Technologies
16.
      * nor the names of its contributors may be used to endorse or promote
17.
       \ ^{*} products derived from this software without specific prior written
18.
      * permission.
19.
20.
      * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
       * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
21.
      * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
22.
23.
       * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
       * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
24.
       * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
25.
      * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
26.
       * DATA. OR PROFITS: OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
27.
      * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
28.
        * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
29.
      * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
30.
31.
32.
33.
34.
      #include <stdlib.h>
35.
36.
      #include <stdio.h>
37.
38.
      #include <pcap.h>
39.
      #define MAX PRINT 80
40.
41.
      #define MAX LINE 16
42.
43.
44.
      void usage();
```

```
46.
 47.
        void main(int argc, char **argv)
 48.
       {
 49.
       pcap t *fp;
       char errbuf[PCAP ERRBUF SIZE];
 50.
       char *source=NULL;
 51.
       char *ofilename=NULL;
 52.
       char *filter=NULL;
 53.
 54.
       int i:
 55.
       pcap dumper t *dumpfile;
 56.
       struct bpf_program fcode;
 57.
       bpf_u_int32 NetMask;
 58.
       int res;
 59.
        struct pcap_pkthdr *header;
 60.
       const u_char *pkt_data;
 61.
 62.
       if (argc == 1)
 63.
           {
 64.
                usage():
 65.
                return;
 66.
 67.
       for(i=1;i < argc; i+= 2)</pre>
 68.
 69.
 70.
 71.
                switch (argv[i] [1])
 72.
 73.
                    case 's':
 74.
                    {
 75.
                       source=argv[i+1];
 76.
 77.
                    break;
 78.
 79.
                    case 'o':
 80.
                    {
                       ofilename=argv[i+1]:
 81.
                    }:
 82.
 83.
                    break:
 84.
                    case 'f':
 85.
 86.
 87.
                       filter=argv[i+1];
 88.
 89.
                    break;
 90.
 91.
 92.
            // open a capture from the network
 93.
 94.
        if (source != NULL)
 95.
                if ( (fp= pcap_open(source,
 96.
 97.
                                    1514 /*snaplen*/.
                                    PCAP_OPENFLAG_PROMISCUOUS /*flags*/
 98.
 99.
                                    20 /*read timeout*/,
                                    NULL /* remote authentication */,
100.
101.
                                    errbuf)
102.
                                    ) == NULL)
103.
104.
                    fprintf(stderr, "\nUnable to open the adapter.\n");
105.
                    return;
106.
107.
           }
108.
109.
           else usage();
110.
111.
            if (filter != NULL)
112.
113.
                // We should loop through the adapters returned by the pcap_findalldevs_ex() \,
114.
                // in order to locate the correct one.
115.
               // Let's do things simpler: we suppose to be in a C class network ;-)
116.
117.
                NetMask=0xffffff;
118.
                //compile the filter
119.
120.
               if(pcap_compile(fp, &fcode, filter, 1, NetMask) < 0)</pre>
121.
                {
122.
                    fprintf(stderr,"\nError compiling filter: wrong syntax.\n");
                    return;
123.
124.
125.
                //set the filter
126.
                if(pcap_setfilter(fp, &fcode)<0)</pre>
127.
128
129.
                    fprintf(stderr,"\nError setting the filter\n");
130.
                   return;
131.
                }
132.
133.
134.
135.
            //open the dump file
```

```
130.
          1T (OTILENAME != NULL)
137.
              dumpfile= pcap_dump_open(fp, ofilename);
138.
139.
140.
              if (dumpfile == NULL)
141.
142.
                 fprintf(stderr,"\nError opening output file\n");
143.
144.
145.
146.
       else usage();
147.
148.
       //start the capture
149.
          while((res = pcap_next_ex( fp, &header, &pkt_data)) >= 0)
150.
       {
151.
152.
             if(res == 0)
              /* Timeout elapsed */
153.
154.
            continue;
155.
156.
            //save the packet on the dump file
157.
              pcap_dump((unsigned char *) dumpfile, header, pkt_data);
158.
159.
160.
      }
161.
162.
       void usage()
163.
164.
      {
165.
       printf("\npf - Generic Packet Filter.\n");
166.
167.
          printf("\nUsage:\npf -s source -o output\_file\_name [-f filter\_string]\n\n");
168.
          exit(0);
169.
     }
原文地址: http://www.ferrisxu.com/WinPcap/html/index.html
文章标签: winpcap 应用程序 源代码 visual c++ 编程
个人分类: 纯编程 计算机网络
```

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com