转 使用WinPcap编程

2013年10月06日 19:47:03　阅读数：2733

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]

```cpp
/*
 * Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
 * Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Politecnico di Torino, CACE Technologies
 * nor the names of its contributors may be used to endorse or promote
 * products derived from this software without specific prior written
 * permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */


#include <stdlib.h>
#include <stdio.h>

```

```c
//
// NOTE: remember to include WPCAP and HAVE_REMOTE among your
// preprocessor definitions.
//

#include <pcap.h>

#define LINE_LEN 16

main(int argc, char **argv)
{
pcap_if_t *alldevs, *d;
pcap_t *fp;
u_int inum, i=0;
char errbuf[PCAP_ERRBUF_SIZE];
int res;
struct pcap_pkthdr *header;
const u_char *pkt_data;

    printf("pktdump_ex: prints the packets of the network using WinPcap.\n");
    printf("   Usage: pktdump_ex [-s source]\n\n"
           "   Examples:\n"
           "      pktdump_ex -s file://c:/temp/file.acp\n"
           "      pktdump_ex -s rpcap://\\Device\\NPF_{C8736017-F3C3-4373-94AC-9A34B7DAD998}\n\n");

    if(argc < 3)
    {

        printf("\nNo adapter selected: printing the device list:\n");
        /* The user didn't provide a packet source: Retrieve the local device list */
        if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf) == -1)
        {
            fprintf(stderr,"Error in pcap_findalldevs_ex: %s\n", errbuf);
            return -1;
        }

        /* Print the list */
        for(d=alldevs; d; d=d->next)
        {
            printf("%d. %s\n    ", ++i, d->name);

            if (d->description)
                printf(" (%s)\n", d->description);
            else
                printf(" (No description available)\n");
        }

        if (i==0)
        {
            fprintf(stderr,"No interfaces found! Exiting.\n");
            return -1;
        }

        printf("Enter the interface number (1-%d):",i);
        scanf("%d", &inum);

        if (inum < 1 || inum > i)
        {
            printf("\nInterface number out of range.\n");

            /* Free the device list */
            pcap_freealldevs(alldevs);
            return -1;
        }

        /* Jump to the selected adapter */
        for (d=alldevs, i=0; i< inum-1 ;d=d->next, i++);

        /* Open the device */
        if ( (fp= pcap_open(d->name,
                            100 /*snaplen*/,
                            PCAP_OPENFLAG_PROMISCUOUS /*flags*/,
                            20 /*read timeout*/,
                            NULL /* remote authentication */,
                            errbuf)
                            ) == NULL)
        {
            fprintf(stderr,"\nError opening adapter\n");
            return -1;
        }
    }
    else
    {
        // Do not check for the switch type ('-s')
        if ( (fp= pcap_open(argv[2],
                            100 /*snaplen*/,
                            PCAP_OPENFLAG_PROMISCUOUS /*flags*/,
                            20 /*read timeout*/,
                            NULL /* remote authentication */,
                            errbuf)
                            ) == NULL)
        {
```

```
        {
            fprintf(stderr,"\nError opening source: %s\n", errbuf);
            return -1;
        }
    }

    /* Read the packets */
    while((res = pcap_next_ex( fp, &header, &pkt_data)) >= 0)
    {

        if(res == 0)
            /* Timeout elapsed */
            continue;

        /* print pkt timestamp and pkt len */
        printf("%ld:%ld (%ld)\n", header->ts.tv_sec, header->ts.tv_usec, header->len);

        /* Print the packet */
        for (i=1; (i < header->caplen + 1 ) ; i++)
        {
            printf("%.2x ", pkt_data[i-1]);
            if ( (i % LINE_LEN) == 0) printf("\n");
        }

        printf("\n\n");
    }

    if(res == -1)
    {
        fprintf(stderr, "Error reading the packets: %s\n", pcap_geterr(fp));
        return -1;
    }

    return 0;
}
```

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。

[cpp]

```
/*
 * Copyright (c) 1999 - 2005 NetGroup, Politecnico di Torino (Italy)
 * Copyright (c) 2005 - 2006 CACE Technologies, Davis (California)
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Politecnico di Torino, CACE Technologies
 * nor the names of its contributors may be used to endorse or promote
 * products derived from this software without specific prior written
 * permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */


#include <stdlib.h>
#include <stdio.h>

#include <pcap.h>

#define MAX_PRINT 80
#define MAX_LINE 16


void usage();


void main(int argc, char **argv)
{
pcap_t *fp;
char errbuf[PCAP_ERRBUF_SIZE];
char *source=NULL;
char *ofilename=NULL;
char *filter=NULL;
int i;
pcap_dumper_t *dumpfile;
struct bpf_program fcode;
bpf_u_int32 NetMask;
int res;
struct pcap_pkthdr *header;
const u_char *pkt_data;

    if (argc == 1)
    {
        usage();
        return;
    }

    for(i=1;i < argc; i+= 2)
    {

        switch (argv[i] [1])
        {
            case 's':
            {
                source=argv[i+1];
            };
            break;

            case 'o':
            {
                ofilename=argv[i+1];
            };
            break;

            case 'f':
            {
                filter=argv[i+1];
            };
            break;
        }
    }

    // open a capture from the network
    if (source != NULL)
    {
        if ( (fp= pcap_open(source,
                            1514 /*snaplen*/,
                            PCAP_OPENFLAG_PROMISCUOUS /*flags*/,
                            20 /*read timeout*/,
                            NULL /* remote authentication */,
                            errbuf)
                            ) == NULL)
        {
            fprintf(stderr,"\nUnable to open the adapter.\n");
            return;
        }
    }

    else usage();

    if (filter != NULL)
    {
        // We should loop through the adapters returned by the pcap_findalldevs_ex()
        // in order to locate the correct one.
        //
        // Let's do things simpler: we suppose to be in a C class network ;-)
        NetMask=0xffffff;

        //compile the filter
        if(pcap_compile(fp, &fcode, filter, 1, NetMask) < 0)
        {
            fprintf(stderr,"\nError compiling filter: wrong syntax.\n");
            return;
        }

        //set the filter
        if(pcap_setfilter(fp, &fcode)<0)
        {
            fprintf(stderr,"\nError setting the filter\n");
            return;
        }

    }

    //open the dump file
    if (ofilename != NULL)
    if (ofilename != NULL)
    {
        dumpfile= pcap_dump_open(fp, ofilename);

        if (dumpfile == NULL)
        {
            fprintf(stderr,"\nError opening output file\n");
            return;
        }
    }
    else usage();

    //start the capture
    while((res = pcap_next_ex( fp, &header, &pkt_data)) >= 0)
    {

        if(res == 0)
        /* Timeout elapsed */
        continue;

        //save the packet on the dump file
        pcap_dump((unsigned char *) dumpfile, header, pkt_data);

    }
}


void usage()
{

    printf("\npf - Generic Packet Filter.\n");
    printf("\nUsage:\npf -s source -o output_file_name [-f filter_string]\n\n");
    exit(0);
}
```

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