计算机网络

Computer Network

3

侦听并分析帧和报文

实验课程

厦門大學
XIAMEN UNIVERSITY

信息学院
（国家示范性软件学院）
School of Informatics

黄　炜
博士，副教授
Dr. Wei Huang

实验目的

- 捕获并分析以太网的帧，获取目标与源网卡的MAC和IP地址
- 获取本机地址
  - IPCONFIG.EXE
  - 通过WinSock的GetAddress命令
- 获取远端MAC地址
  - ARP
  - WinPCAP

内容纲要

包含文件头和库

```
#include <iphlpapi.h>
#include <windows.h>
#pragma comment(lib,“iphlpapi")
#pragma comment(lib,"WS2_32")
using namespace std;
```

包含文件头和库

- 1、申请内存

```
::GetAdaptersInfo(pAdapterInfo, &ulLen);
pAdapterInfo=(PIP_ADAPTER_INFO)::malloc(ulLen);
```

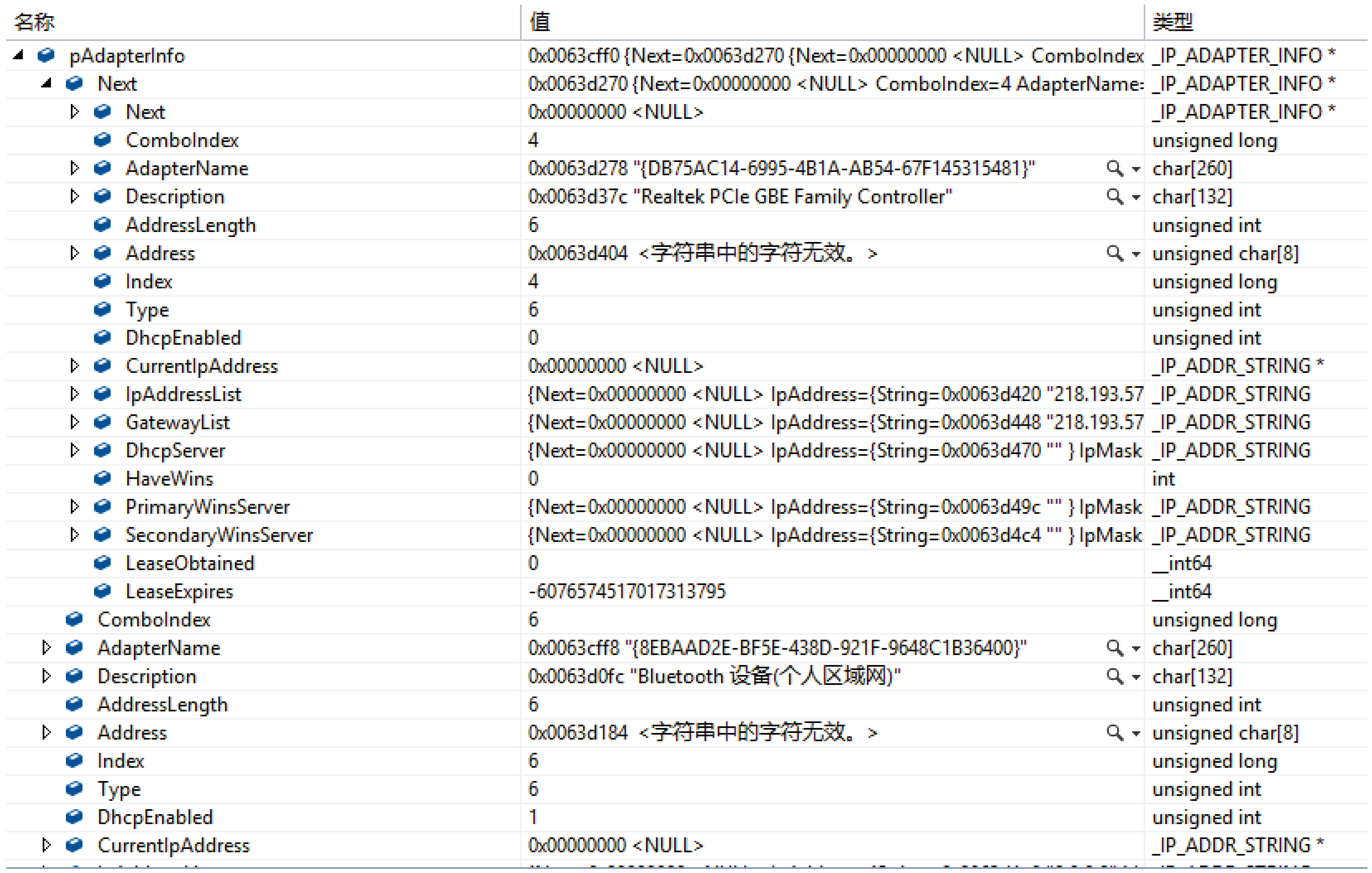
  – 此时，ulLen为网卡个数（禁用除外）乘以640。

- 2、获取本地适配器结构信息

```
::GetAdaptersInfo(pAdapterInfo, &ulLen);
```

- 3、malloc新建的变量都要free

适配器信息



名称	值	类型
◢ pAdapterInfo	0x0063cff0 {Next=0x0063d270 {Next=0x00000000 <NULL> ComboIndex	_IP_ADAPTER_INFO *
◢ Next	0x0063d270 {Next=0x00000000 <NULL> ComboIndex=4 AdapterName:	_IP_ADAPTER_INFO *
▷ Next	0x00000000 <NULL>	_IP_ADAPTER_INFO *
ComboIndex	4	unsigned long
▷ AdapterName	0x0063d278 "{DB75AC14-6995-4B1A-AB54-67F145315481}"	char[260]
▷ Description	0x0063d37c "Realtek PCIe GBE Family Controller"	char[132]
AddressLength	6	unsigned int
▷ Address	0x0063d404 <字符串中的字符无效。>	unsigned char[8]
Index	4	unsigned long
Type	6	unsigned int
DhcpEnabled	0	unsigned int
▷ CurrentIpAddress	0x00000000 <NULL>	_IP_ADDR_STRING *
▷ IpAddressList	{Next=0x00000000 <NULL> IpAddress={String=0x0063d420 "218.193.57	_IP_ADDR_STRING
▷ GatewayList	{Next=0x00000000 <NULL> IpAddress={String=0x0063d448 "218.193.57	_IP_ADDR_STRING
▷ DhcpServer	{Next=0x00000000 <NULL> IpAddress={String=0x0063d470 "" } IpMask	_IP_ADDR_STRING
HaveWins	0	int
▷ PrimaryWinsServer	{Next=0x00000000 <NULL> IpAddress={String=0x0063d49c "" } IpMask	_IP_ADDR_STRING
▷ SecondaryWinsServer	{Next=0x00000000 <NULL> IpAddress={String=0x0063d4c4 "" } IpMask	_IP_ADDR_STRING
LeaseObtained	0	__int64
LeaseExpires	-6076574517017313795	__int64
ComboIndex	6	unsigned long
▷ AdapterName	0x0063cff8 "{8EBAAD2E-BF5E-438D-921F-9648C1B36400}"	char[260]
▷ Description	0x0063d0fc "Bluetooth 设备(个人区域网)"	char[132]
AddressLength	6	unsigned int
▷ Address	0x0063d184 <字符串中的字符无效。>	unsigned char[8]
Index	6	unsigned long
Type	6	unsigned int
DhcpEnabled	1	unsigned int
▷ CurrentIpAddress	0x00000000 <NULL>	_IP_ADDR_STRING *

运行结果

```
NIC 1:
    IP: 0.0.0.0; Mask: 0.0.0.0; Gateway: 0.0.0.0
    MAC: C4D987******
NIC 2:
    IP: 218.193.57.***; Mask: 255.255.255.***; Gateway:
218.193.57.***
    MAC: F8B156******
```

代码示例

```
PIP_ADAPTER_INFO pAdapterInfo = NULL;
ULONG ulLen = 0;
::GetAdaptersInfo(pAdapterInfo, &ulLen);
pAdapterInfo = (PIP_ADAPTER_INFO)::malloc(ulLen);
::GetAdaptersInfo(pAdapterInfo, &ulLen);
int count = 0;
while (pAdapterInfo) {
       printf("NIC %d: \n", ++count);
       printf("\tIP: %s; Mask: %s; Gateway: %s\n", pAdapterInfo-
>IpAddressList.IpAddress.String, pAdapterInfo-
>IpAddressList.IpMask.String, pAdapterInfo-
>GatewayList.IpAddress.String);
       printf("\tName: %s; Desc: %s\n", pAdapterInfo->AdapterName,
pAdapterInfo->Description);
```

代码示例

```
        printf("\tMAC: ");
        for (size_t i = 0; i < pAdapterInfo->AddressLength; i++) {
                printf("%02X", pAdapterInfo->Address[i]);
        }
        printf("\n");
        pAdapterInfo = pAdapterInfo->Next;
}
system("pause");
if (pAdapterInfo) {
        free(pAdapterInfo);
}
```

内容纲要

事前准备

- 安装WinPCAP
  - WinPcap_4_1_3.exe
- 解压缩WpdPack将文件夹拷出备用
  - 4.1.1-WpdPack.zip
  - Include、Lib
- 正确解析MAC和IP地址是本节课第一要务，也是基本功

包含头文件和库

```
#define HAVE_REMOTE
#include <pcap.h>
#include <Packet32.h>
#include <ntddndis.h>
#pragma comment(lib, "Packet")
#pragma comment(lib, "wpcap")
#pragma comment(lib, "WS2_32")
```

数据结构定义

IP帧格式

```
typedef struct ip_header {
        u_char  ver_ihl;                // Version (4 bits) + Internet 
header length (4 bits)
        u_char  tos;                    // Type of service
        u_short tlen;                   // Total length
        u_short identification; // Identification
        u_short flags_fo;               // Flags (3 bits) + Fragment 
offset (13 bits)
        u_char  ttl;                    // Time to live
        u_char  proto;                  // Protocol
        u_short crc;                    // Header checksum
        u_char saddr[4];                // Source address
        u_char daddr[4];                // Destination address
        u_int   op_pad;                 // Option + Padding
} ip_header;
```

数据结构定义

```
typedef struct mac_header {
        u_char dest_addr[6];
        u_char src_addr[6];
        u_char type[2];
} mac_header;
```

用回调方法捕获数据包

- packet_handler指向一个可以接收数据包的函数。这个函数会在收到每个新的数据包并收到一个通用状态时被libpcap所调用。

```
/* prototype of the packet handler */
void packet_handler(u_char *param, const struct pcap_pkthdr *header,
const u_char *pkt_data);
```

函数主体

- 1、获取本地适配器结构信息，打开适配器
  - pcap_findalldevs_ex; pcap_freealldevs
- 2、循环编译过滤器
  - pcap_compile
  - pcap_setfilter
- 3、开始循环捕获
  - pcap_loop

获得设备列表

```
/* Retrieve the device list */
if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf)
== -1) {
        fprintf(stderr, "Error in pcap_findalldevs: %s\n", errbuf);
        exit(1);
}
/* Print the list */
for (d = alldevs; d; d = d->next) {
        printf("%d. %s", ++i, d->name);
        if (d->description)
                printf(" (%s)\n", d->description);
        else
                printf(" (No description available)\n");
}
```

选择设备

```
if (i == 0) {
        printf("\nNo interfaces found! Make sure WinPcap is 
installed.\n");
        return -1;
}
printf("Enter the interface number (1-%d):", i);
scanf_s("%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++);
```

选择设备

```
/* Open the adapter *
if ((adhandle = pcap_open(d->name, 65536, PCAP_OPENFLAG_PROMISCUOUS,
        1000, NULL, errbuf)) == NULL) {
      fprintf(stderr, "\nUnable to open the adapter. %s is not
supported by WinPcap\n");
      pcap_freealldevs(alldevs);
      return -1;
}
```

设备名

捕获包，混杂模式
65536为所有包

混杂模式

时延

远程验证

错误缓冲

释放设备列表，最后一步都要释放

预处理

```
if (pcap_datalink(adhandle) != DLT_EN10MB) {
        fprintf(stderr, "\nThis program works only on Ethernet 
networks.\n");
        pcap_freealldevs(alldevs);
        return -1;
}
if (d->addresses != NULL)
        netmask = ((struct sockaddr_in *)(d->addresses->netmask))-
>sin_addr.S_un.S_addr;
else
        netmask = 0xffffff;
```

检查链路层。只简单支持以太网。

检索接口的第一个地址的掩码

如果接口没有地址，假设在一个C类网络

编译和设置过滤器

```
//compile the filter
if (pcap_compile(adhandle, &fcode, packet_filter, 1, netmask) <0) {
        fprintf(stderr, "\nUnable to com[illegible]le the packet filter. Check 
the syntax.\n");
        pcap_freealldevs(alldevs);
        return -1;
}
//set the filter
if (pcap_setfilter(adhandle, &fcode)<0)     {
        fprintf(stderr, "\nError setting the filter.\n");
        pcap_freealldevs(alldevs);
        return -1;
}
```

编译过滤器

```
char packet_filter[] = "ip and udp";
```

设置过滤器

开始捕获

```
printf("\nlistening on %s...\n", d->description);


/* At this point, we don't need any more the device list. Free it */
pcap_freealldevs(alldevs);


/* start the capture */
pcap_loop(adhandle, 0, packet_handler, NULL);
```

开始捕获

开始捕获

```
void packet_handler(u_char *param, const struct pcap_pkthdr *header,
const u_char *pkt_data)
{
...
       mac_header *mh;
       ip_header *ih;
...
       int length = sizeof(mac_header)+sizeof(ip_header);
       for (int i = 0; i<length; i++) {
              printf("%02X ", pkt_data[i]);
              if ((i & 0xF) == 0xF)
                     printf("\n");
       }
       printf("\n");
}
```

通过libpcap的每一个传入的数据包调用回调函数

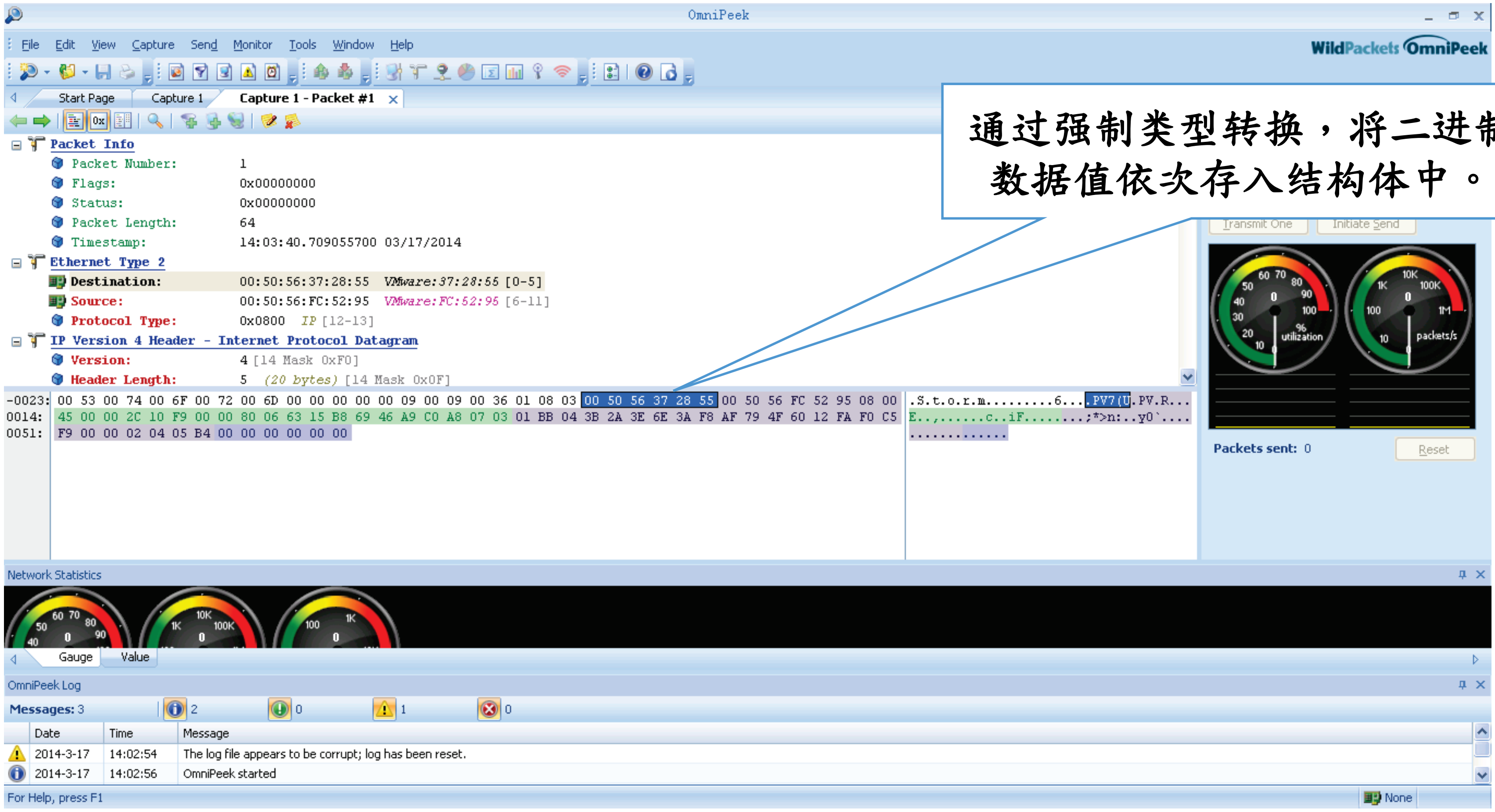
按二进制输出数据

捕获后的处理

```
mh = (mac_header*)pkt_data;
printf("mac_header:\n");
printf("\tdest_addr: ");
for (int i = 0; i<6; i++) {
        printf("%02X ", mh->dest_addr[i]);
}
printf("\n");
printf("\tsrc_addr: ");
for (int i = 0; i<6; i++) {
        printf("%02X ", mh->src_addr[i]);
}
printf("\n");
printf("\ttype: %04X", ntohs(mh->type));
printf("\n");
```

通过强制类型转换，将二进制数据值依次存入结构体中。

编程时注意对照Omnipeek工具



```
Packet Info
    Packet Number:        1
    Flags:                0x00000000
    Status:               0x00000000
    Packet Length:        64
    Timestamp:            14:03:40.709055700 03/17/2014
Ethernet Type 2
    Destination:          00:50:56:37:28:55  VMware:37:28:55 [0-5]
    Source:               00:50:56:FC:52:95  VMware:FC:52:95 [6-11]
    Protocol Type:        0x0800  IP [12-13]
IP Version 4 Header - Internet Protocol Datagram
    Version:              4 [14 Mask 0xF0]
    Header Length:        5  (20 bytes) [14 Mask 0x0F]

-0023: 00 53 00 74 00 6F 00 72 00 6D 00 00 00 00 00 09 00 09 00 36 01 08 03 00 50 56 37 28 55 00 50 56 FC 52 95 08 00  .S.t.o.r.m.........6....PV7(U.PV.R...
0014:  45 00 00 2C 10 F9 00 00 80 06 63 15 B8 69 46 A9 C0 A8 07 03 01 BB 04 3B 2A 3E 6E 3A F8 AF 79 4F 60 12 FA F0 C5  E..,......c..iF.......;*>n:..yO`....
0051:  F9 00 00 02 04 05 B4 00 00 00 00 00 00                                                                     ............
```

Packets sent: 0

Messages: 3 | 2 | 0 | 1 | 0

Date	Time	Message
2014-3-17	14:02:54	The log file appears to be corrupt; log has been reset.
2014-3-17	14:02:56	OmniPeek started

For Help, press F1

通过强制类型转换，将二进制数据值依次存入结构体中。

厦門大學 XIAMEN UNIVERSITY 信息学院（国家示范性软件学院）School of Informatics 黄 炜 博士·副教授 Dr. Wei Huang

捕获后的处理

```
/* retireve the position of the ip header */
ih = (ip_header *)(pkt_data + sizeof(mac_header)); //length of
ethernet header

printf("ip_header\n");
printf("\t%-10s: %02X\n", "ver_ihl", ih->ver_ihl);
printf("\t%-10s: %02X\n", "tos", ih->tos);
printf("\t%-10s: %04X\n", "tlen", ntohs(ih->tlen));
printf("\t%-10s: %04X\n", "identification", ntohs(ih>identification));
printf("\t%-10s: %04X\n", "flags_fo", ntohs(ih->flags_fo));
printf("\t%-10s: %02X\n", "ttl", ih->ttl);
printf("\t%-10s: %02X\n", "proto", ih->proto);
printf("\t%-10s: %04X\n", "crc", ntohs(ih->crc));
printf("\t%-10s: %08X\n", "op_pad", ntohs(ih->op_pad));
printf("\t%-10s: ", "saddr:");
```

通过强制类型转换，将二进制数据值依次存入结构体中。

网络端序转为主机端序ntohs

捕获后的处理

```
for (int i = 0; i<4; i++) {
       printf("%02X ", ih->saddr[i]);
}
printf("  ");
for (int i = 0; i<4; i++) {
       printf("%d.", ih->saddr[i]);
}
printf("\n");
printf("\t%-10s: ", "daddr");
for (int i = 0; i<4; i++) {
       printf("%02X ", ih->daddr[i]);
}
printf("  ");
for (int i = 0; i<4; i++) {
       printf("%d.", ih->daddr[i]);
}
printf("\n");
```

运行结果

```
14:30:38.511184 len:339
00 0C 29 73 69 8A 00 50 56 FC 52 95 08 00 45 00
01 45 1D FB 00 00 80 11 8C 56 C0 A8 07 02 C0 A8
07 04 00 35 CB 42
mac_header:
        dest_addr : 00 0C 29 73 69 8A
        src_addr  : 00 50 56 FC 52 95
        type      : 0800
ip_header
        ver_ihl    : 45
        tos        : 00
        tlen       : 0145
        identification: 1DFB
        flags_fo   : 0000
        ttl        : 80
        proto      : 11
        crc        : 8C56
        op_pad     : 0035CB42
        saddr      : C0 A8 07 02   192.168.7.2
        daddr      : C0 A8 07 04   192.168.7.4
```

有时候在这里还有PPPoE头

注意：MAC地址合理。

注意：IP Ver应为4

注意：IP地址合理。

内容纲要

包含头文件

```c
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <netinet/if_ether.h>
#include <linux/sockios.h>
#include <stdio.h>
#include <string.h>
```

强制类型转换

```
int main()
{
...
        eh = (struct ethhdr *)ep;

        fd = socket(AF_INET, SOCK_PACKET, htons(0x0003));
        strcpy(ifr.ifr_name, "eth0");
        i = ioctl(fd, SIOCGIFFLAGS, &ifr);
        ifr.ifr_flags |= IFF_PROMISC;
        i = ioctl(fd, SIOCSIFFLAGS, &ifr);
...
```

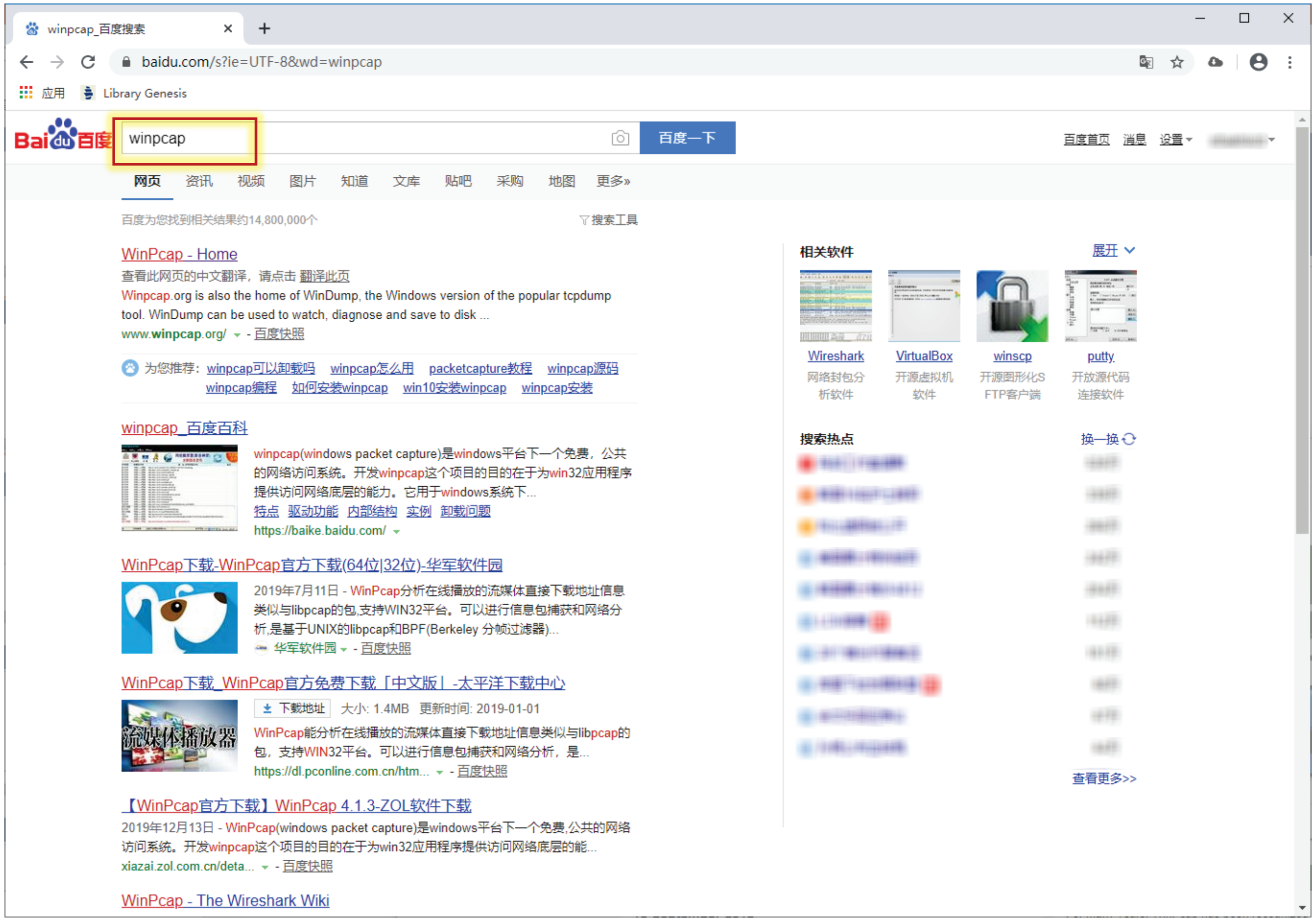
输出源地址和目的地址

```
        while (1) {
                fl = read(fd, ep, sizeof(ep));
                if (fl > ETH_HLEN) {
                        printf("Packet is from ");
                        for (i = 0; i < 6; i++)
                                printf("%x-", eh->h_source[i]);
                        printf(" to ");
                        for (i = 0; i < 6; i++)
                                printf("%x-", eh->h_dest[i]);
                        printf("\n");
                } // End of if
        } // End of While
}
```

内容纲要

搜索开发支撑软件

- 搜索关键字WinPCAP，在官网上下载



winpcap_百度搜索

baidu.com/s?ie=UTF-8&wd=winpcap

应用 Library Genesis

winpcap 百度一下

百度首页 消息 设置 [redacted]

网页 资讯 视频 图片 知道 文库 贴吧 采购 地图 更多»

百度为您找到相关结果约14,800,000个 搜索工具

WinPcap - Home
查看此网页的中文翻译，请点击 翻译此页
Winpcap.org is also the home of WinDump, the Windows version of the popular tcpdump tool. WinDump can be used to watch, diagnose and save to disk ...
www.winpcap.org/ - 百度快照

为您推荐：winpcap可以卸载吗 winpcap怎么用 packetcapture教程 winpcap源码 winpcap编程 如何安装winpcap win10安装winpcap winpcap安装

winpcap_百度百科
winpcap(windows packet capture)是windows平台下一个免费，公共的网络访问系统。开发winpcap这个项目的目的在于为win32应用程序提供访问网络底层的能力。它用于windows系统下...
特点 驱动功能 内部结构 实例 卸载问题
https://baike.baidu.com/

WinPcap下载-WinPcap官方下载(64位|32位)-华军软件园
2019年7月11日 - WinPcap分析在线播放的流媒体直接下载地址信息类似与libpcap的包,支持WIN32平台。可以进行信息包捕获和网络分析,是基于UNIX的libpcap和BPF(Berkeley 分帧过滤器)...
华军软件园 - 百度快照

WinPcap下载_WinPcap官方免费下载「中文版」-太平洋下载中心
下载地址 大小: 1.4MB 更新时间: 2019-01-01
WinPcap能分析在线播放的流媒体直接下载地址信息类似与libpcap的包，支持WIN32平台。可以进行信息包捕获和网络分析，是...
https://dl.pconline.com.cn/htm... - 百度快照

【WinPcap官方下载】WinPcap 4.1.3-ZOL软件下载
2019年12月13日 - WinPcap(windows packet capture)是windows平台下一个免费,公共的网络访问系统。开发winpcap这个项目的目的在于为win32应用程序提供访问网络底层的能...
xiazai.zol.com.cn/deta... - 百度快照

WinPcap - The Wireshark Wiki

相关软件 展开

Wireshark	VirtualBox	winscp	putty
网络封包分析软件	开源虚拟机软件	开源图形化SFTP客户端	开放源代码连接软件

搜索热点 换一换

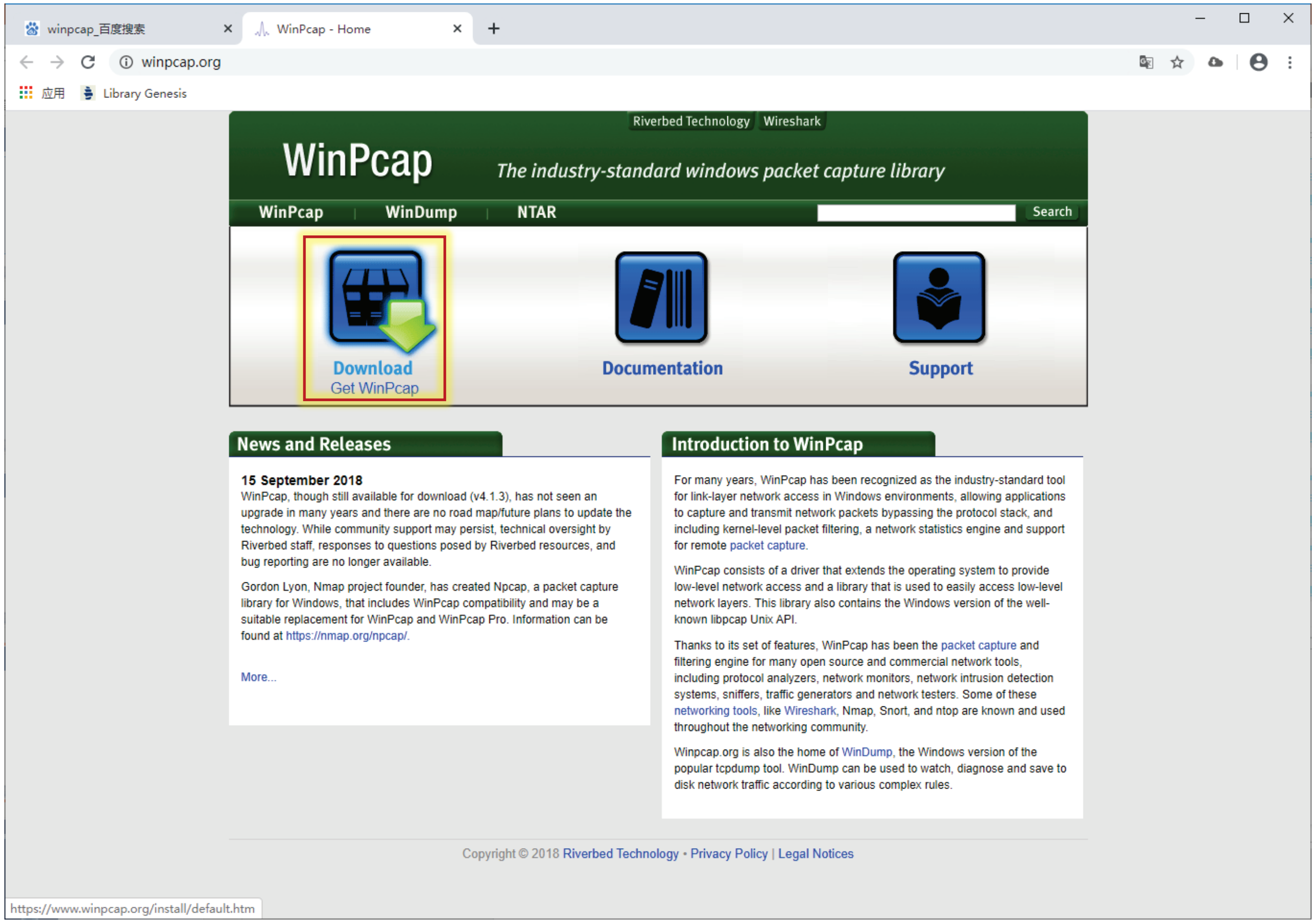
[redacted]

查看更多>>

厦門大學 XIAMEN UNIVERSITY 信息学院（国家示范性软件学院）School of Informatics 黄炜 博士·副教授 Dr. Wei Huang

下载开发支撑软件

- 在官网上找到下载链接



News and Releases

15 September 2018
WinPcap, though still available for download (v4.1.3), has not seen an upgrade in many years and there are no road map/future plans to update the technology. While community support may persist, technical oversight by Riverbed staff, responses to questions posed by Riverbed resources, and bug reporting are no longer available.

Gordon Lyon, Nmap project founder, has created Npcap, a packet capture library for Windows, that includes WinPcap compatibility and may be a suitable replacement for WinPcap and WinPcap Pro. Information can be found at https://nmap.org/npcap/.

More...

Introduction to WinPcap

For many years, WinPcap has been recognized as the industry-standard tool for link-layer network access in Windows environments, allowing applications to capture and transmit network packets bypassing the protocol stack, and including kernel-level packet filtering, a network statistics engine and support for remote packet capture.

WinPcap consists of a driver that extends the operating system to provide low-level network access and a library that is used to easily access low-level network layers. This library also contains the Windows version of the well-known libpcap Unix API.

Thanks to its set of features, WinPcap has been the packet capture and filtering engine for many open source and commercial network tools, including protocol analyzers, network monitors, network intrusion detection systems, sniffers, traffic generators and network testers. Some of these networking tools, like Wireshark, Nmap, Snort, and ntop are known and used throughout the networking community.

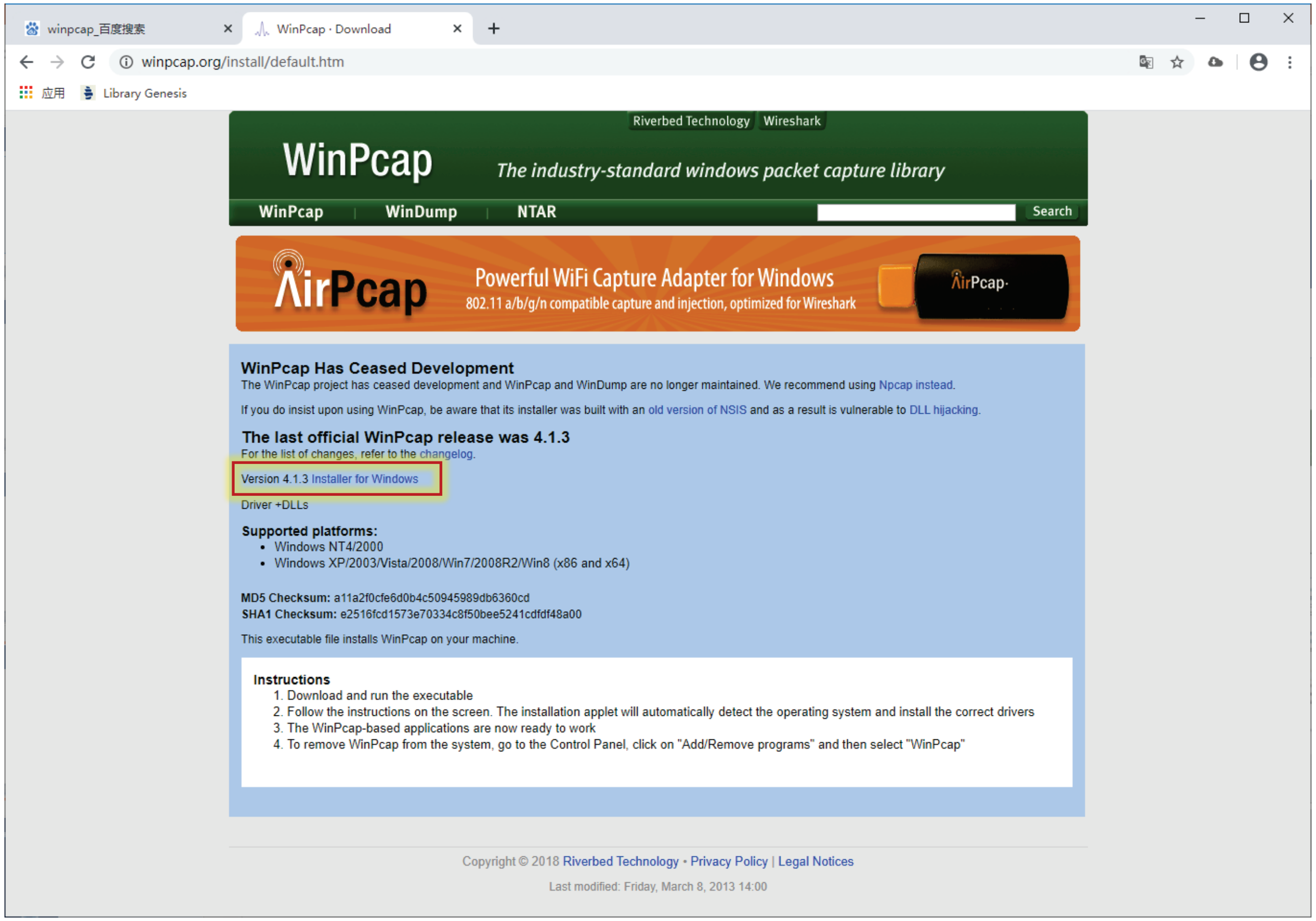
Winpcap.org is also the home of WinDump, the Windows version of the popular tcpdump tool. WinDump can be used to watch, diagnose and save to disk network traffic according to various complex rules.

Copyright © 2018 Riverbed Technology • Privacy Policy | Legal Notices

https://www.winpcap.org/install/default.htm

下载开发支撑软件

- 判定下载链接位置


winpcap_百度搜索
WinPcap · Download
winpcap.org/install/default.htm
应用
Library Genesis
Riverbed Technology
Wireshark
WinPcap
The industry-standard windows packet capture library
WinPcap
WinDump
NTAR
Search
AirPcap
Powerful WiFi Capture Adapter for Windows
802.11 a/b/g/n compatible capture and injection, optimized for Wireshark

WinPcap Has Ceased Development

The WinPcap project has ceased development and WinPcap and WinDump are no longer maintained. We recommend using Npcap instead.

If you do insist upon using WinPcap, be aware that its installer was built with an old version of NSIS and as a result is vulnerable to DLL hijacking.

The last official WinPcap release was 4.1.3

For the list of changes, refer to the changelog.

Version 4.1.3 Installer for Windows

Driver +DLLs

Supported platforms:
- Windows NT4/2000
- Windows XP/2003/Vista/2008/Win7/2008R2/Win8 (x86 and x64)

MD5 Checksum: a11a2f0cfe6d0b4c50945989db6360cd
SHA1 Checksum: e2516fcd1573e70334c8f50bee5241cdfdf48a00

This executable file installs WinPcap on your machine.

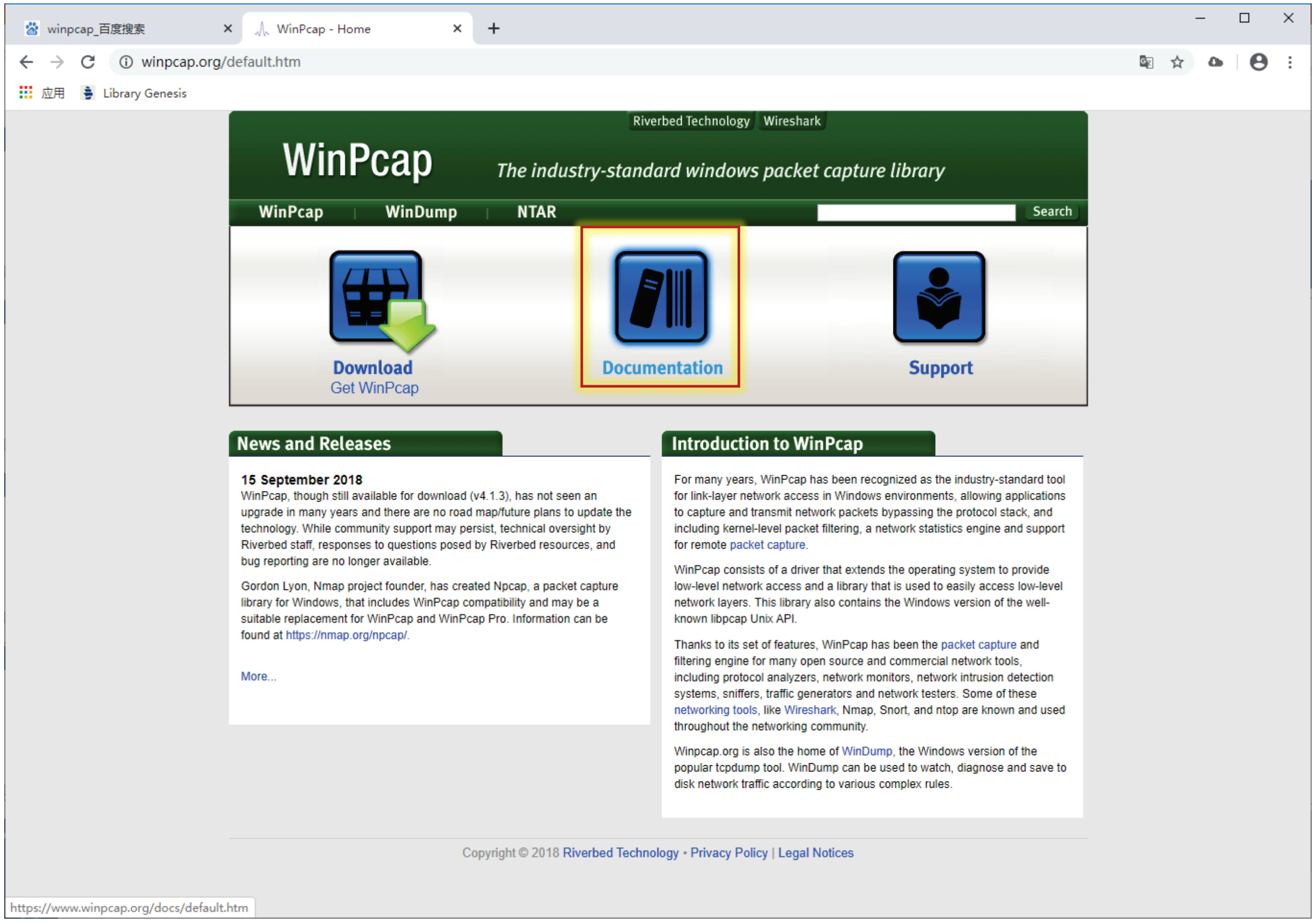
Instructions
1. Download and run the executable
2. Follow the instructions on the screen. The installation applet will automatically detect the operating system and install the correct drivers
3. The WinPcap-based applications are now ready to work
4. To remove WinPcap from the system, go to the Control Panel, click on "Add/Remove programs" and then select "WinPcap"

Copyright © 2018 Riverbed Technology • Privacy Policy | Legal Notices

Last modified: Friday, March 8, 2013 14:00

查找官方示例代码

- 找到示例代码的下载位置



News and Releases

15 September 2018
WinPcap, though still available for download (v4.1.3), has not seen an upgrade in many years and there are no road map/future plans to update the technology. While community support may persist, technical oversight by Riverbed staff, responses to questions posed by Riverbed resources, and bug reporting are no longer available.

Gordon Lyon, Nmap project founder, has created Npcap, a packet capture library for Windows, that includes WinPcap compatibility and may be a suitable replacement for WinPcap and WinPcap Pro. Information can be found at https://nmap.org/npcap/.

More...

Introduction to WinPcap

For many years, WinPcap has been recognized as the industry-standard tool for link-layer network access in Windows environments, allowing applications to capture and transmit network packets bypassing the protocol stack, and including kernel-level packet filtering, a network statistics engine and support for remote packet capture.

WinPcap consists of a driver that extends the operating system to provide low-level network access and a library that is used to easily access low-level network layers. This library also contains the Windows version of the well-known libpcap Unix API.

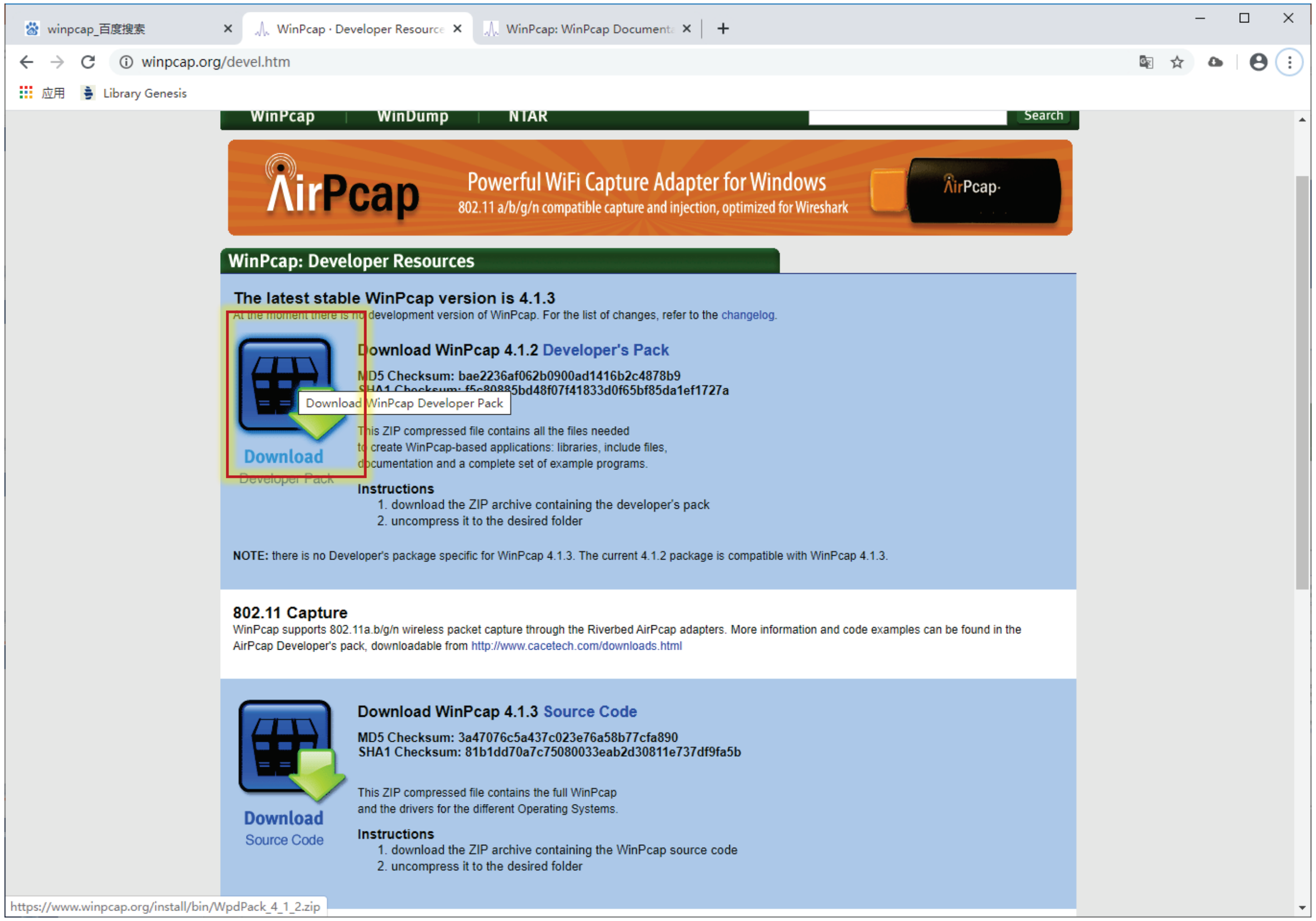
Thanks to its set of features, WinPcap has been the packet capture and filtering engine for many open source and commercial network tools, including protocol analyzers, network monitors, network intrusion detection systems, sniffers, traffic generators and network testers. Some of these networking tools, like Wireshark, Nmap, Snort, and ntop are known and used throughout the networking community.

Winpcap.org is also the home of WinDump, the Windows version of the popular tcpdump tool. WinDump can be used to watch, diagnose and save to disk network traffic according to various complex rules.

Copyright © 2018 Riverbed Technology • Privacy Policy | Legal Notices

查找官方示例代码

- 找到示例代码的下载位置



winpcap_百度搜索 | WinPcap · Developer Resource | WinPcap: WinPcap Document

winpcap.org/devel.htm

应用 Library Genesis

WinPcap WinDump NTAR Search

AirPcap
Powerful WiFi Capture Adapter for Windows
802.11 a/b/g/n compatible capture and injection, optimized for Wireshark

WinPcap: Developer Resources

The latest stable WinPcap version is 4.1.3

At the moment there is no development version of WinPcap. For the list of changes, refer to the changelog.

Download WinPcap 4.1.2 Developer's Pack

MD5 Checksum: bae2236af062b0900ad1416b2c4878b9
SHA1 Checksum: f5c80885bd48f07f41833d0f65bf85da1ef1727a

Download WinPcap Developer Pack

This ZIP compressed file contains all the files needed to create WinPcap-based applications: libraries, include files, documentation and a complete set of example programs.

Instructions
1. download the ZIP archive containing the developer's pack
2. uncompress it to the desired folder

NOTE: there is no Developer's package specific for WinPcap 4.1.3. The current 4.1.2 package is compatible with WinPcap 4.1.3.

Download Developer Pack

802.11 Capture

WinPcap supports 802.11a.b/g/n wireless packet capture through the Riverbed AirPcap adapters. More information and code examples can be found in the AirPcap Developer's pack, downloadable from http://www.cacetech.com/downloads.html

Download WinPcap 4.1.3 Source Code

MD5 Checksum: 3a47076c5a437c023e76a58b77cfa890
SHA1 Checksum: 81b1dd70a7c75080033eab2d30811e737df9fa5b

This ZIP compressed file contains the full WinPcap and the drivers for the different Operating Systems.

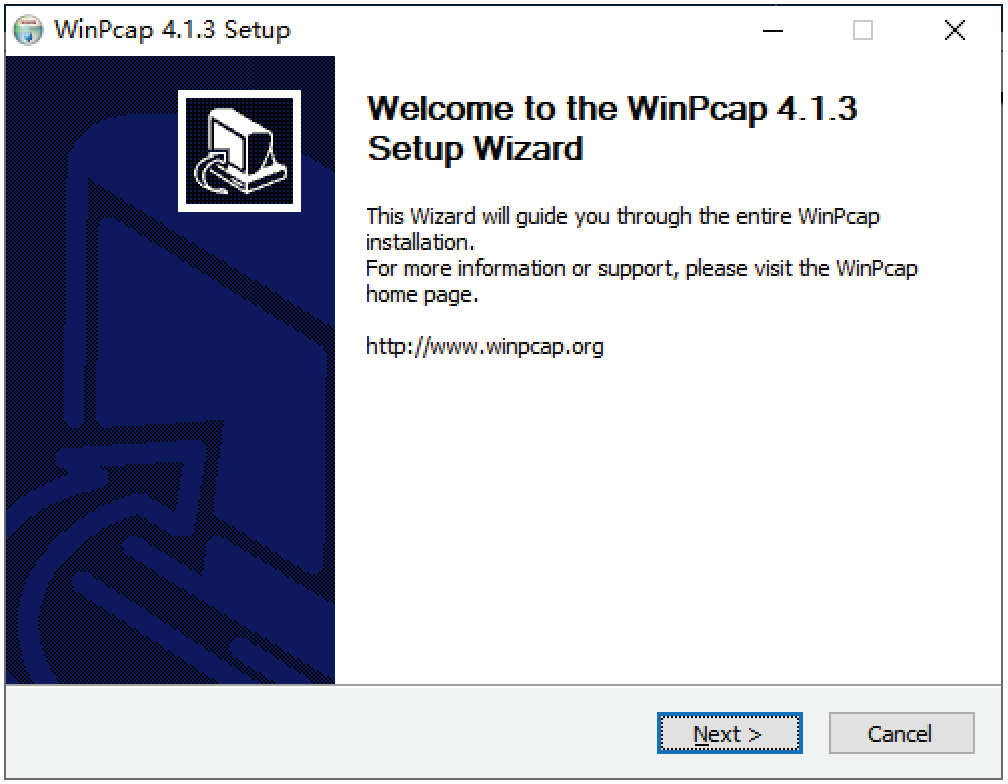
Instructions
1. download the ZIP archive containing the WinPcap source code
2. uncompress it to the desired folder

Download Source Code

https://www.winpcap.org/install/bin/WpdPack_4_1_2.zip

安装支撑软件

- 双击并运行WinPCAP支撑软件WinPcap_4_1_3.exe
  - 有的驱动软件需要重启生效


WinPcap 4.1.3 Setup

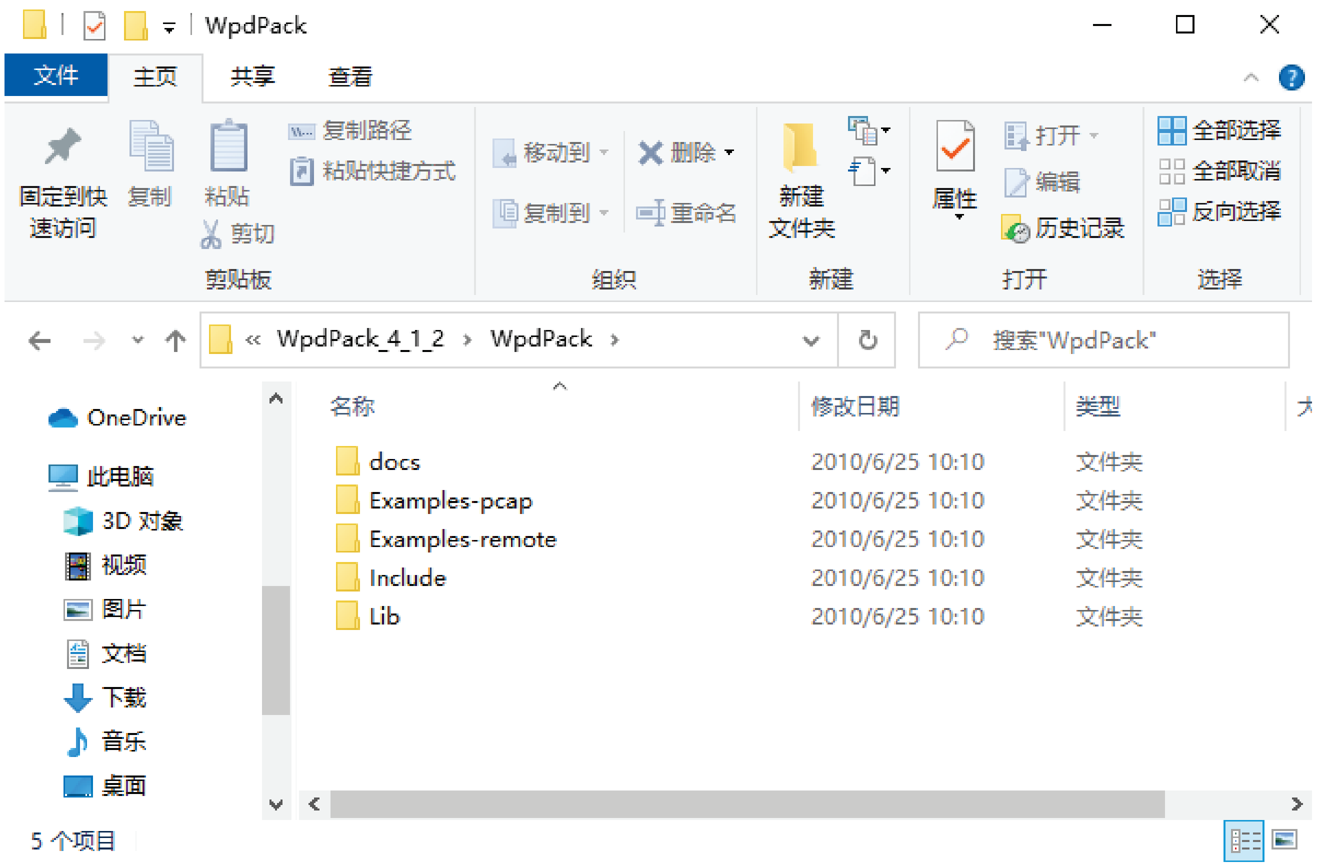
Welcome to the WinPcap 4.1.3 Setup Wizard

This Wizard will guide you through the entire WinPcap installation.
For more information or support, please visit the WinPcap home page.

http://www.winpcap.org

查看示例代码

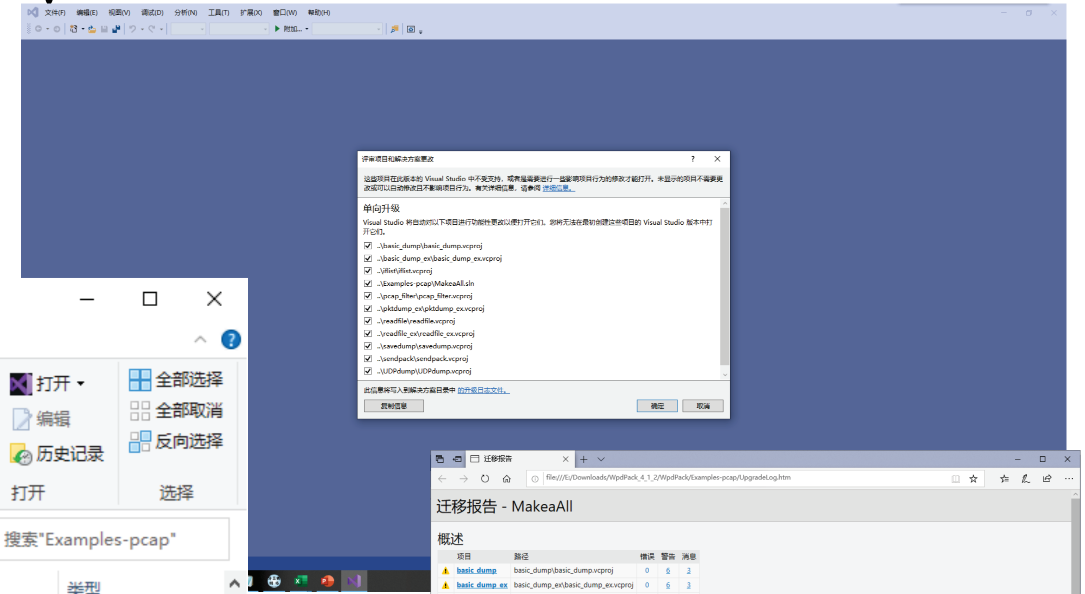
- 解压示例代码包，认清结构，找到示例程序
  - 帮助文档
  - 示例代码
  - 头文件
  - 库文件



名称	修改日期	类型
docs	2010/6/25 10:10	文件夹
Examples-pcap	2010/6/25 10:10	文件夹
Examples-remote	2010/6/25 10:10	文件夹
Include	2010/6/25 10:10	文件夹
Lib	2010/6/25 10:10	文件夹

运行示例代码

- 双击.sln文件，打开工程


评审项目和解决方案更改

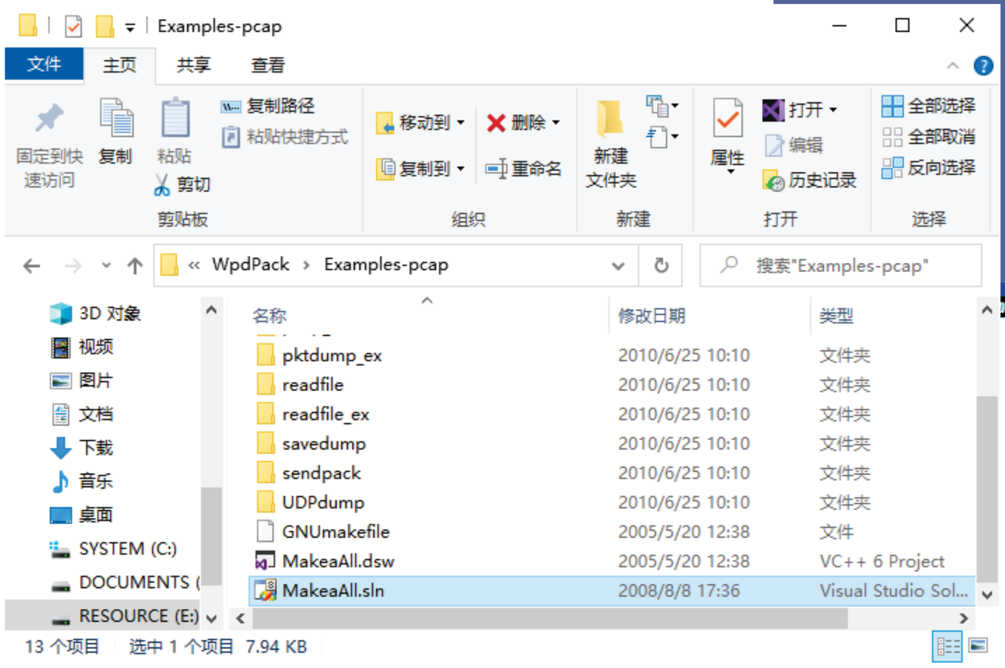
这些项目在此版本的 Visual Studio 中不受支持，或者是需要进行一些影响项目行为的修改才能打开。未显示的项目不需要更改或可以自动修改且不影响项目行为。有关详细信息，请参阅 详细信息。

单向升级

Visual Studio 将自动对以下项目进行功能性更改以便打开它们。您将无法在最初创建这些项目的 Visual Studio 版本中打开它们。

- ✓ ..\basic_dump\basic_dump.vcproj
- ✓ ..\basic_dump_ex\basic_dump_ex.vcproj
- ✓ ..\iflist\iflist.vcproj
- ✓ ..\Examples-pcap\MakeaAll.sln
- ✓ ..\pcap_filter\pcap_filter.vcproj
- ✓ ..\pktdump_ex\pktdump_ex.vcproj
- ✓ ..\readfile\readfile.vcproj
- ✓ ..\readfile_ex\readfile_ex.vcproj
- ✓ ..\savedump\savedump.vcproj
- ✓ ..\sendpack\sendpack.vcproj
- ✓ ..\UDPdump\UDPdump.vcproj

此信息将写入到解决方案目录中 的升级日志文件。

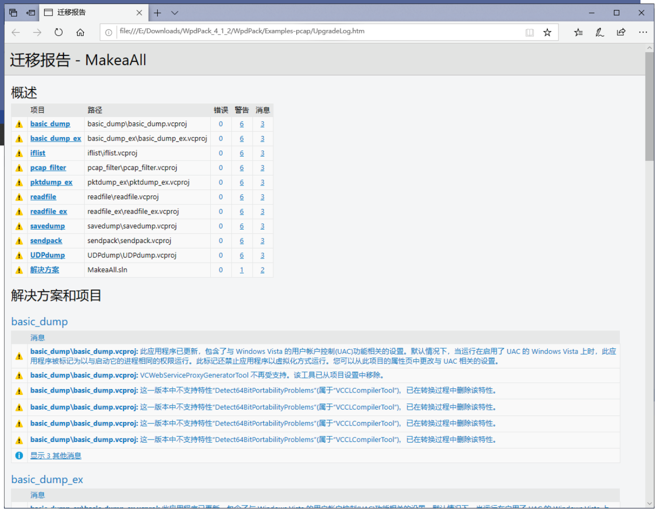

Examples-pcap

« WpdPack › Examples-pcap

搜索"Examples-pcap"

名称	修改日期	类型
pktdump_ex	2010/6/25 10:10	文件夹
readfile	2010/6/25 10:10	文件夹
readfile_ex	2010/6/25 10:10	文件夹
savedump	2010/6/25 10:10	文件夹
sendpack	2010/6/25 10:10	文件夹
UDPdump	2010/6/25 10:10	文件夹
GNUmakefile	2005/5/20 12:38	文件
MakeaAll.dsw	2005/5/20 12:38	VC++ 6 Project
MakeaAll.sln	2008/8/8 17:36	Visual Studio Sol...

13 个项目　选中 1 个项目 7.94 KB


迁移报告

file:///E:/Downloads/WpdPack_4_1_2/WpdPack/Examples-pcap/UpgradeLog.htm

迁移报告 - MakeaAll

概述

	项目	路径	错误	警告	消息
⚠	basic_dump	basic_dump\basic_dump.vcproj	0	6	3
⚠	basic_dump_ex	basic_dump_ex\basic_dump_ex.vcproj	0	6	3
⚠	iflist	iflist\iflist.vcproj	0	6	3
⚠	pcap_filter	pcap_filter\pcap_filter.vcproj	0	6	3
⚠	pktdump_ex	pktdump_ex\pktdump_ex.vcproj	0	6	3
⚠	readfile	readfile\readfile.vcproj	0	6	3
⚠	readfile_ex	readfile_ex\readfile_ex.vcproj	0	6	3
⚠	savedump	savedump\savedump.vcproj	0	6	3
⚠	sendpack	sendpack\sendpack.vcproj	0	6	3
⚠	UDPdump	UDPdump\UDPdump.vcproj	0	6	3
⚠	解决方案	MakeaAll.sln	0	1	2

解决方案和项目

basic_dump

	消息
⚠	**basic_dump\basic_dump.vcproj:** 此应用程序已更新，包含了与 Windows Vista 的用户帐户控制(UAC)功能相关的设置。默认情况下，当运行在启用了 UAC 的 Windows Vista 上时，此应用程序被标记为以与启动它的进程相同的权限运行。此标记还禁止应用程序以虚拟化方式运行。您可以从此项目的属性页中更改与 UAC 相关的设置。
⚠	**basic_dump\basic_dump.vcproj:** VCWebServiceProxyGeneratorTool 不再受支持。该工具已从项目设置中移除。
⚠	**basic_dump\basic_dump.vcproj:** 这一版本中不支持特性"Detect64BitPortabilityProblems"(属于"VCCLCompilerTool")，已在转换过程中删除该特性。
⚠	**basic_dump\basic_dump.vcproj:** 这一版本中不支持特性"Detect64BitPortabilityProblems"(属于"VCCLCompilerTool")，已在转换过程中删除该特性。
⚠	**basic_dump\basic_dump.vcproj:** 这一版本中不支持特性"Detect64BitPortabilityProblems"(属于"VCCLCompilerTool")，已在转换过程中删除该特性。
⚠	**basic_dump\basic_dump.vcproj:** 这一版本中不支持特性"Detect64BitPortabilityProblems"(属于"VCCLCompilerTool")，已在转换过程中删除该特性。
ℹ	显示 3 其他消息

basic_dump_ex

消息

内容纲要

计 算 机 网 络
Computer Network

谢谢观看
Thank you

3

侦听并分析帧和报文

实验教程

厦門大學
XIAMEN UNIVERSITY

信息学院
（国家示范性软件学院）
School of Informatics

黄 炜
博士，副教授
Dr. Wei Huang