

Web-chat Monitor System-Research and Implementation

Fan XIONG¹, Yong FANG², & Tao TAO²

¹ Information Security Institute, Sichuan University, Chengdu
Sichuan 610064 China

² School of Computer Science & Engineering, University of New
South Wales, Sydney NSW 2032 Australia

ABSTRACT: Nowadays, Web-chat becomes wildly used as communication tool on Internet. However, the technology has been misused in communication as crime related matters, like abduction of children and young teenagers. The ability of Web-chat monitoring is more and more demanding. This paper presents an integrated Web-chat automatic monitoring and security management solution, which can control the chat conversation according to the blocking rules. This paper contributes in maximizing the real time supervising and controlling ability on Web-chat users. Our results show that the Web-chat monitoring and controlling tasks can be efficiently automated using the Web-chat Monitor System implemented in this paper.

1 INTRODUCTION

1.1 Existing Problems

Nowadays, as the power of Internet service is growing rapidly, more and more people enjoy resource sharing and other conveniences provided by Internet. However, the technology has been misused in communication as crime related matters, like abduction of children and young teenagers via chat rooms, since these are hard to monitor. Such monitoring tools would aid in crime detection or even crime prevention. In general, there is not much research on monitoring chat room conversations for discovering criminal activities. Current monitoring techniques are basically manual [MMDHS01], which is difficult, tedious, costly, and time consuming.

1.2 Features And Pitfalls Of The Existing Products

Chat-monitor system has been wildly used in the world, and there are already several mature products. For instance, Chat Blocker [CB05], Net Nanny Chat Monitor [NNCM05], eBlaster [Ebl05], and MSN Chat Monitor [MCM05]. By comparing and analyzing these products, they can be divided into 2 kinds. The first one is for single PC, such as Chat Blocker, Net Nanny Chat Monitor and eBlaster. The other one is for the LAN, such as MSN Chat Monitor. For both kinds of Chat-monitor products, they have some pitfalls. For the first one, the host needs to install the client software, which decreases the monitoring flexibility. It can not monitor the PC which hasn't installed the client software yet. Although it can disable chat conversations from taking place, it only has the ability to allow chat conversations or not. The logs are just recorded for auditing, which can't be used to moderate and restrict the chat conversations. For the other one, it can monitor all conversations in your "Local Area Network" without the use of client software installed on the remote computer, however, the chat conversation of the remote computer can't be moderated and restricted. And it can only monitor the chat conversation in the same subnet.

1.3 Advantages Of System Implemented In This Paper

Compared with several existing Chat-monitor products mentioned above, the Web-chat Monitor System realizing in this paper has several advantages. First, bypass access technology is introduced in this project, and the architecture of network doesn't need to be changed. Second, it will monitor all conversations of your PCs without the use of client software installed on the remote computer, which increases the monitoring flexibility. Even with the brand-new computer accessing to network, the system still works well. Third, the monitor terminal connects the Hub or the mirror port of Switch, and it can capture all datagrams passing through the equipment. Not only can it monitor all conversations in your "Local Area Network", but also monitor the remote computers which locate in other subnet. Forth, and most importantly, it logs and alarms instantly, and meanwhile it blocks or allows the chat conversations according to the rules, which can be dynamically set based on the content of

conversation. It avoids the lag of audit. And last, rules can be set on-line according to the sensitive information manager cares about, and it is transparent to the end users.

This paper is organized as follows. Section 2 describes the concept of Web-chat system and related technology of Web-chat Monitor System realizing in this paper. Section 3 provides an overview of the Web-chat monitoring life cycle. Section 4 presents implementation details focused on monitoring and management functionalities. The final section presents related work, conclusions and future work.

2 BACKGROUND

2.1 Web-chat System

In today's Internet there are various chat systems in use which differ in a number of aspects. Internet Relay Chat (IRC) is one of the oldest systems still in use [DWF03, OR93]. While IRC is still a widely used protocol, a significant share of today's users appear to be using Web-chat systems. There seem to be two main motivations: ease of use and ease of access. Everyone is familiar with the user interface of a Web browser, and most portals are offering Web-chat systems. Unfortunately these systems vary widely in terms of visual appearance and technical realization as well as in terms of protocol. In sharp contrast to IRC, there are a large number of disparate Web-chat systems, each based on different protocols using a wide range of port numbers.

2.2 WinPcap Library

WinPcap [WinP05] is the industry-standard tool for link-layer network access in Windows environments: it allows applications to capture and transmit network packets bypassing the protocol stack, and has additional useful features, including kernel-level packet filtering, a network statistics engine and support for remote packet capture. WinPcap consists of a driver, which extends the operating system to provide low-level network access, and a library that is used to easily access the low-level network layers.

2.3 Bypass Access Technology

Bypass access technology is used in this project [CJR89]. Using this technology, the host can receive all the datagrams passing intranet's export regardless of the IP destinations. And the access of the host is transparent to the other PCs of intranet. The network performance isn't limited by the host's. By contraries, using series access technology, the intranet's PCs need to know the IP address of the host. It receives all the datagrams sent by PCs. Then the host transmits the datagrams according to the IP address. The network performance is mostly decided by the host's. Bypass access technology has several advantages compared with series access technology.

The advantage of using bypass access technology is that:

1. No change is necessary for existing network topological architecture.
2. No bottleneck problem will be caused by such technology.
3. It brings great flexibility and adaptability between Monitor System and Management System.

3. THE FUCTIONALITIES OF WEB-CHAT MONITORING SYSTEM

3.1 Datagram Inspect Process

The monitoring system processes data as following steps [Eln02, Ste00]:

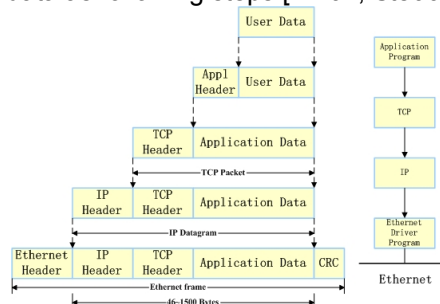


Figure 1 Network packet encapsulation format

1. Resolve the datagram. As shown in Figure 1, the datagram captured by WinPcap will be

parsed into *raw user data* according to TCP/IP standard which is supported by application layer.

2. Pre-process the data in *raw user data* format. According to the transmit protocols of Web-chat, the data will be translate from *raw user data* into *plaintext data* format.
3. Analyze *plaintext data*. According to the keyword library, analysis will be applied to the data in *plaintext data* format, an identifier will be set to label it if this TCP connection should be blocked or not.
4. Web-chat monitoring is managed by identifier. The system monitors the Web-chat by identifier created from last step. If the content of Web-chat violates the monitoring policy, a reset packet will be sent to disconnect the TCP connection between the client and server.
5. Online dynamic management of the keyword library. Keyword library can be created for different monitoring purposes. According to management's classification of granularity, different keyword libraries' threshold (Threshold is the times or frequency of keyword which appears in one paragraph.) can be set for different modifying rule.

3.2 Architecture Of Web-chat Monitor System

The Web-chat monitor system is consisted by two following components:

1. **Monitor System.** Monitor system is installed on Monitor Terminal. The Monitor Terminal connects the Hub or the mirror port of Switch, it captures all datagrams passing the equipment, resolves the packets according to protocols, analyzes the content and decides whether to block the connection.
2. **Management System.** The Management System is installed on Management Terminal. The Management Terminal connects several Monitor Terminals through Internet or intranet as long as the Monitor System can access the Management System. The Management System's monitoring policy and related configuration can be set by friendly graphic user interface, then distributed to those terminals.

As mentioned above, the centralized Management System is used to manage those distributed Monitor Systems. The architecture of Web-chat monitor system is shown in Figure 2.

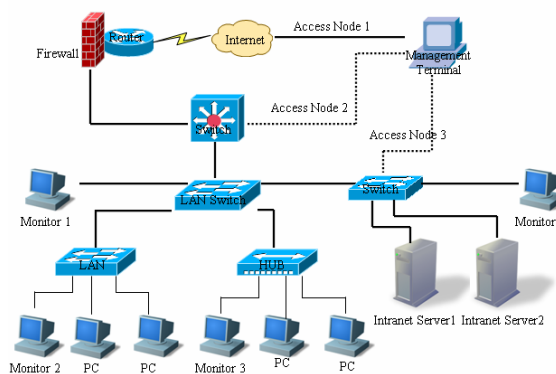


Figure 2 Distributed monitor topological architecture

4 IMPLEMENTATION DETAILS

4.1 The Functional Modules Of Management System

The architecture of management system's modules is shown in Figure 3.

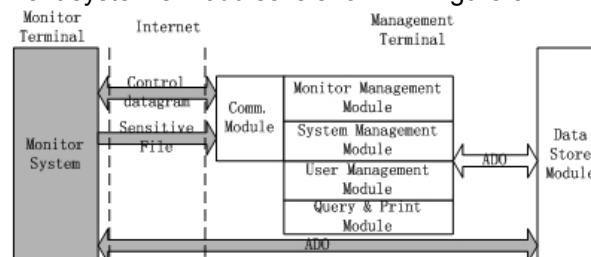


Figure 3 The architecture of management system's modules

The management terminal maintains and refreshes the keyword library dynamically and manages the monitor terminal online and distributes the keyword library.

The management terminal gets logs and alarms sent by monitor system by ADO/DAO, and it refreshes the record information in the main interface (Figure 4) instantly and makes statistical analysis with pie chart.

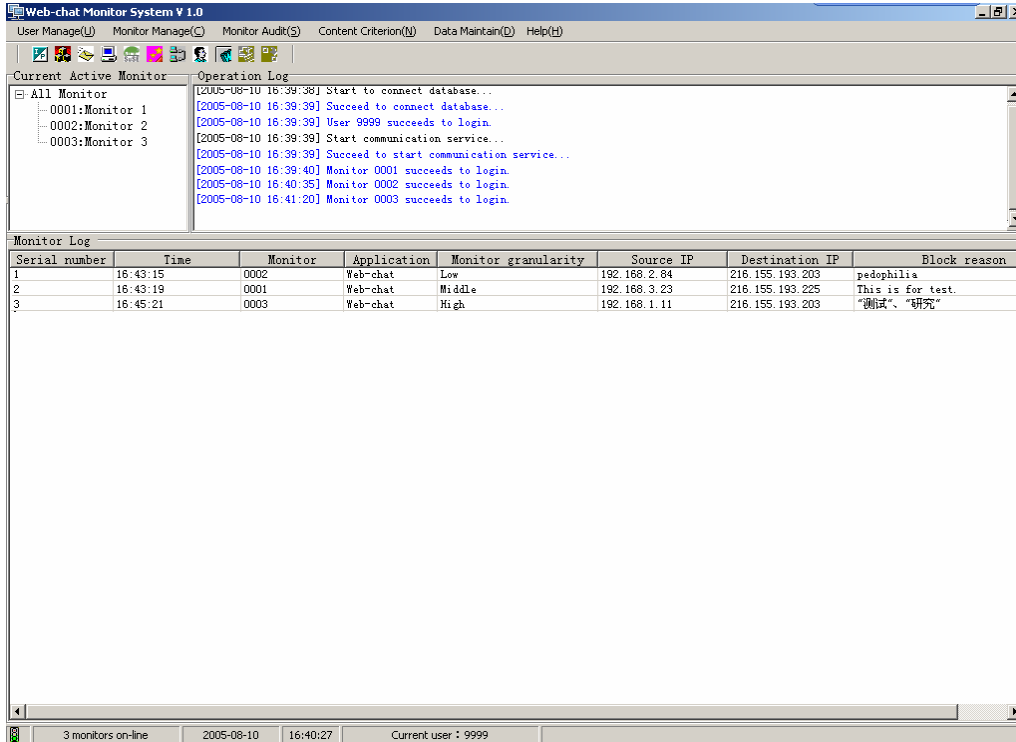


Figure 4 The main interface of Management System

4.2 The Functional Modules Of Monitor System

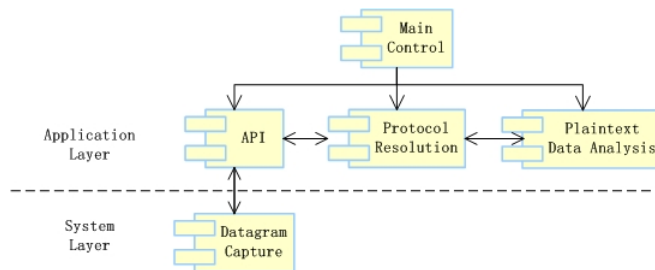


Figure 5 Module architecture of the bypass access monitor system

As shown above, the Monitor System is consisted of five modules:

1. Main Control Module. The Main Control Module has following functionalities:
 - a) To download all the information (e.g. policy and configuration) from the Management System through ADO/DAO.
 - b) To receive and dispose all kinds of commands sent from Management System.
 - c) To show the whole flow of datagram in different protocols by graph (Figure 6).

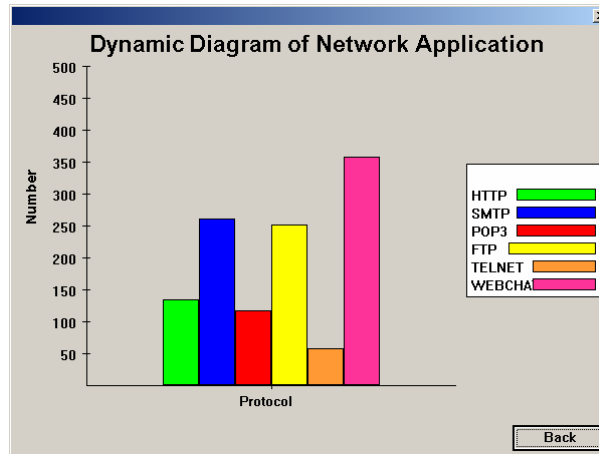


Figure 6 Flow of datagram in different protocols

2. Datagram Capture Module. It is implemented by following Network driver interface standard (NDIS). Benefited from the broadcasting characteristic of shared Ethernet, it catches the network frame directly from data link layer. If a frame is identified to be blocked, the function `pcap_sendpacket(pCap, rstBuffer, length)` [Tan02] provided by WinPcap will be called in order to send fabricated block datagram to both client and server. According to the TCP/IP protocols, the block datagram is fabricated by setting the identifiers ACK 1 and RST 1 of TCP header. A reset datagram is produced to stop current TCP connection and announce the application connection to be reset. By doing this, we are able to then filter out all non-TCP packets and transmit TCP packets to the API Module.

```
//First step: Looking for device
//it will return a character string which presents the adapter
pAdapter = pcap_lookupdev(ebuf);
//Second step: Open the adapter
pCap = pcap_open_live(pAdapter, 1800, 1, 20, ebuf);
//Third step: Check the type of data link
pcap_datalink(pCap);
//Forth step: Begin to capture network packet
pcap_loop(pCap, 0, Netmon_handler, (UCHAR *)lpParameter);
//Fifth step: Close the adapter
pcap_close(pCap);
```

Figure 7 API of Datagram Capture Module developed by WinPcap

3. API Module. It transmits the TCP packet which is captured by Datagram Capture Module to the Protocol Resolution Module. And it provides an independent interface from Operating System.
4. Protocol Resolution Module. It decodes and resolves the TCP packet partially or completely, then transmit those data to the Plaintext Data Analysis Module.
5. Plaintext Data Analysis Module. It tries to find key word in received data packet with calling a semantic analysis component named "WAnalyses" of this system. If the data is identified to be blocked, Datagram Capture Module will be called.

4.3 Case Study

Netease (www.163.com), Sina (<http://english.sina.com/index.html>) and Yahoo (www.yahoo.com) has been used for our case study. The focus will be Protocol Resolution Module because Web-chat communication protocols and port numbers are Website dependant. The Plaintext Data Analysis Module is also worth mentioning because different module needs to be created for different language (e.g. Chinese, Japanese).

Through our case study, it is found that the Web-chat service provided by the websites is adopted different communication protocols. However, HTTP standard is followed when Web-chat service connection is established.

According to the analyzed network packet, the TCP Web-chat connection port used for Netease is 11036, Sina is 80 and Yahoo is 8002. On the content transmitting format side, Netease and Sina use plaintext, while Yahoo uses UTF-8.

Analysis will be applied to the raw user data after preprocess it, the whole flow is shown as followed:

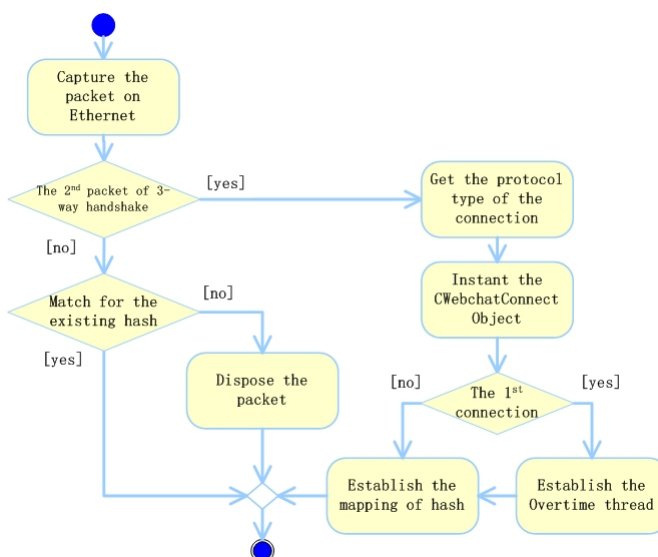


Figure 8 Flow chart of protocol resolve module

Notice that all TCP connections are established through three-way handshake [Fel00], new connection should be identified when the second packet (In the TCP header, identifier ACK 1, SYN 1) arrives in order to keep the generality. As a result, the data begins to be analyzed in the protocol resolution module after the second packet arrives.

Plaintext Data Analysis Module is language dependant. Here we take Yahoo China as a sample. Following processes will be followed to analyze the data sent between server and client:

In Yahoo case, it uses UTF-8 format [Yer03]. We need to resolve the data into plaintext data format before applying Plaintext Data analysis over it. According to the coding check list of UTF-8 to Unicode (It is shown in Table 1), the plaintext data can be gotten as the input of the next module.

Table 1 The coding check list of UTF-8 to Unicode

Char number range (hexadecimal)	UTF-8 octal sequence (binary)
0000 0000-0000 007F	0xxxxxxx
0000 0080-0000 07FF	110xxxxx 10xxxxxx
0000 0800-0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000-0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

The coding field of Chinese Unicode is from 0000 0800 to 0000 FFFF. According to this coding check list, the Web-chat data of Yahoo can be decoded. The decoding approach is shown in Figure 9.

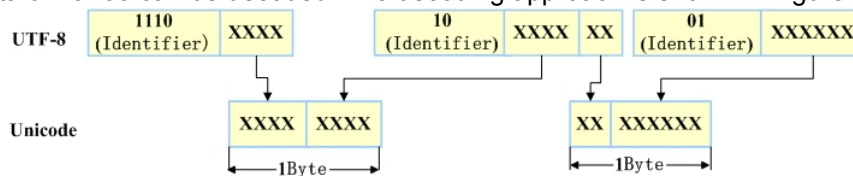


Figure 9 The decoding approach from UTF-8 to Unicode

After getting the plaintext data of Web-chat, it can be transmitted to Plaintext Data Analysis Module to get the return identifier whether it needs to block the TCP connection. Part of the C++ code is provided below (It is shown in Figure 10).

After getting the plaintext data format, it will transmit to the Plaintext Data Analysis Module. Then it returns an identifier to decide whether the chat conversation needs to be blocked. Figure 11 shows the implementing code of function SendRstTcpPacket, which is called by Datagram Capture Module.

```

case WEBCHAT_YAHOO_PORT:
if(!bWebchatLock){ // yahoo TCP port 8002
bWebchatLock = true;
//store character string after 'YCHT'
PUCHAR chatbuffer = new UCHAR[MAX_WEBCHAT_LINE];
PUCHAR temp = new UCHAR[pDataLen+1];
//receive data from 'pData' for this function
if(temp != NULL)
memcpy(temp,pData,pDataLen);
if(strstr((PCHAR)temp,"YCHT") != NULL){
char *str1, *str3;
str1 = strstr((PCHAR)temp,"YCHT");
//point to the end of the message
str3 = (PCHAR)temp+pDataLen;
PUCHAR p;
int chatlen = 0;
int i = 0;
//get the content of the message
for(i=0,p=(PUCHAR)str1+4;p<(PUCHAR)str3;p++,i++){
if(*p > 0x00){
//recount the length of the string
chatbuffer[chatlen] = *p;
chatlen++;
}
}
pRet = new CHAR[MAX_WEBCHAT_LINE];
WebchatAnalysis_yahoo(chatbuffer,chatlen,pRet,
m_IsDeny,m_Return);
if(m_Return)
goto TRUERETURN; //block
else
goto FALSERETURN; //not block
}
}
break;
}
//call the module of semantic analysis
void CWebchatConnect::WebchatAnalysis_yahoo(PUCHAR WebchatBuffer,
int WebchatLen, PCHAR pReturn, int &m_IsDeny, bool &m_return)
{
SocketKey mykey;
mykey.ulDstIP = key->ulDstIP;
mykey.ulSrcIP = key->ulSrcIP;
mykey.usDstPort = key->usDstPort;
mykey.usSrcPort = key->usSrcPort;
InputData *TextData = new InputData;
PCHAR tempdata = new CHAR[WebchatLen+2];
TextData->pData = tempdata;
PCHAR WebchatBufferDecoded = new CHAR[WebchatLen+2];
//If the character string is not decoded,
//it dose not need to encapsulate
WebchatDecode_yahoo(WebchatBuffer, WebchatLen,
WebchatBufferDecoded);
if(m_bDecoded) {
Encapsulation(TextData,mykey,WebchatBufferDecoded,
WebchatLen);
pAnalyseText(TextData,pReturn);
PCHAR IsHave;
CHAR pTempStartchar[5];
strcpy(pTempStartchar,"%");
int i = 1;
while(i<5) {
IsHave = strstr(pReturn,pTempStartchar);
if(IsHave) {
PCHAR pTempIsHave = strstr(IsHave,"#");
int k = pTempIsHave - IsHave;
if(k>3) {
//Assign the address of pMyRetStr to TempStr
Webchat_Rcd *TempStr = (Webchat_Rcd *)pMyRetStr;
strcpy(TempStr->cRemark,pReturn);
m_IsDeny = 1;
m_return = true;
delete TextData;
}
}
}
}
}
}

```

Figure 10 Implementing code of Protocol Resolution Mode

```

if (pTcpH->syn && !pTcpH->ack) {
//send RST packet to server
if( MakeRstTcpPacket(pMach, pIPH, pTcpH, FALSE, uiTcpDataLen, rstBuffer, PACKET_BUFFER_SIZE, &length))
pcap_sendpacket(pHold, rstBuffer, length);
//send RST packet to client
if( MakeRstTcpPacket(pMach, pIPH, pTcpH, TRUE, uiTcpDataLen, rstBuffer, PACKET_BUFFER_SIZE, &length))
pcap_sendpacket(pHold, rstBuffer, length);
}
}

```

Figure 11 Implementing code of SendRstTcpPacket

5 CONCLUSION AND FUTURE WORK

The paper has focuses on maximizing the real time supervising and controlling ability on Web-chat conversations. Benefited from WinPcap technology, the datagram can be captured on Ethernet for Web-chat application. It is implemented to log, alarm and block instantly in the Web-chat Monitor System. Block rules can be set on-line according to the sensitive information the manager cares about. And block does not simply disable chat conversation from taking place, it can allow or block the chat conversation according to the rules set by the manager. It is extremely important to prevent from misusing of Web-chat.

However, we found several areas in which Web-chat monitor system could be improved. By experimental evaluation, under the flow of 10 mbps, the capture rate of datagram is 100 percent, but the resolve rate is 93 percent, block rate is only 82 percent. There are two limitations in our solution. The first limitation is that the delayed time of datagram transmitted from Datagram Capture Module to Protocols Resolution Module is not short enough, compared with the flow of the access node of the Monitor Terminal. The second limitation is that accuracy of protocol resolution and plaintext analysis will decrease as long as the network flow increases. In the future, we are going to optimize the design of protocol resolution and plaintext analysis in order to have a better performance with quicker resolution and higher accuracy of analysis and block. And we are going to apply this system under the network environment of 100 mbps and implement it with hardwares to improve its performance.

ACKNOWLEDGE

We would like to thank the Information Security Institute of Sichuan University (ISISCU) for funding this

research effort. Also we would like to acknowledge numerous colleagues for helping us access and understand the network measurement and for valuable suggestions on how to improve the presentation of the material.

REFERENCES

- [CB05] Chat Blocker, 2005 <http://www.allformp3.com/chat-blocker.htm>
- [CJR89] D. D. Clark, V. Jacobson, J. Romkey, and H. Salwen, "An analysis of TCP processing overhead," *IEEE Communications Magazine*, vol. 27, June 1989
- [Eln02] Eiman M. Elnahrawy. Log-Based Chat Room Monitoring Using Text Categorization: A Comparative Study. *Proceeding of the IASTED International Conference Information and Knowledge Sharing*, pages 111-115, 2002
- [EBI05] eBlaster. 2005 <http://www.eblaster.co.uk>
- [Fel00] A. Feldmann, "Characteristics of TCP connection arrivals," in *Self-Similar Network Traffic And Performance Evaluation* (K. Park and W. Willinger, eds.), J. Wiley & Sons, Inc.2000
- [MMDHS01] A. Meehan, G. Manes, L. Davis, J. Hale, and S. Sheno, "Packet sniffing for automated chat room monitoring and evidence preservation," in *Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security*, June 2001
- [MCM05] MSN Chat Monitor. 2005 <http://www.ajivasoft.com/msn-chat-monitor.htm>
- [NNCM05] Net Nanny Chat Monitor.
<http://www.microdirect.co.uk/ProductInfo.aspx?ProductID=6466&GroupID=0>
- [OR93] J. Oikarinen and D. Reed, "Internet Relay Chat Protocol RFC 1495," 1993
- [Ste00] W. Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols. China Machine Press, April 2000
- [Tan02] Siliang Tan. Monitor and Hide-Disclosure of Network Listening and Technique of Data Protection [M]. Beijing: Posts & Telecommunications Press, 2002
- [WinP05] WinPcap. 2005 <http://www.winpcap.org>
- [Yer03] Yergeau, F. 2003. "UTF-8, a transformation format of ISO 10646," RFC 3629, November