Proceedings of the 2001 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY, 5–6 June 2001

Packet Sniffing for Automated Chat Room Monitoring and Evidence Preservation

A. Meehan, G. Manes, L. Davis, J. Hale, S. Shenoi*

Abstract— **Packet sniffers are designed to intercept network traffic in shared communication channels. This is accomplished by re-configuring network interface cards to permit device drivers to process all network traffic, including packets that are not addressed to the host computer. Packet sniffing is primarily used in intrusion detection, network management, wiretapping and hacking.**

This paper describes a novel application of packet sniffing to monitor chat room conversations for criminal activity. Current manual monitoring techniques must scrutinize massive amounts of conversations for potential criminal activity. The packet sniffer described in this paper permits the automated monitoring and filtering of chat room conversations. Moreover, it records and preserves packet-based evidence, enabling the complete reconstruction of illicit chat room activity for purposes of prosecution.

Keywords— **Packet sniffing, filtering, chat room monitoring, computer crime**

I. Introduction

Packet sniffing is the act of intercepting and interpreting network traffic transmitted across shared communication channels [1,6,10]. The network interface card (NIC) in a networked computer receives all shared traffic sent across a physical link. Ordinarily, the network device driver only processes incoming traffic to the local host and broadcast packets meant for computers in the network [4]. To perform packet sniffing, it is necessary to re-configure the NIC to operate in a "promiscuous" mode where the network device driver processes all traffic transmitted across the network, regardless of whether or not packets are addressed to the host computer [10].

Packet sniffing is primarily used in intrusion detection, network management, wiretapping and hacking. Intrusion detection systems use sniffing to identify packets and packet sequences that signal potential attacks [6,10]. Network management tools employ packet sniffers to quantitatively measure network traffic and identify bottlenecks [10]. Wiretapping applications of packet sniffers are exemplified by the FBI's Carnivore system [1]. Hackers utilize packet sniffing to eavesdrop on network traffic and steal private information [3,6,10,11].

This paper describes a novel application of a packet sniffer to monitor chat room conversations for criminal activity, e.g., sex crimes investigations that monitor sexually explicit chat rooms for "travellers" – pedophiles who seek to arrange liaisons with minors across state lines. Two problems exist with manual monitoring methods [5]. First, an overwhelming amount of conversations have to be actively scrutinized and filtered for potential criminal activity. Second, it is difficult to record and preserve evidence of chat room conversations [12-14]. The packet sniffer permits automated monitoring and filtering, as well as evidence preservation. Moreover, it is possible to completely reconstruct chat room activity for purposes of prosecution.

II. Chat Room Monitor

Current chat room applications are based on the client-server model [15]. A schematic diagram of the chat room monitor is shown in Figure 1. Chat room clients A and B, a "suspicious client" and a monitoring officer communicate using a chat room server. In typical Internet chat rooms (e.g., Yahoo, MSN, AOL and IRC), as many as 100 clients communicate constantly. During an investigation, a monitoring officer must continuously monitor conversations, possibly interacting with other clients and suspects, to obtain sufficient evidence of criminal activity, including conspiracy to commit a crime [8,12,13].

The chat room monitor resides on a physically separated locked computer to remove any possibility of evidence corruption by the monitoring officer. The system intercepts all chat room communications through a "tap" located within the monitoring officer's local area network. It automatically preserves all chat room traffic in a sealed container for evidentiary purposes. Moreover, it facilitates real-time conversation filtering to alert the monitoring officer of suspicious activity. This frees the officer from tedious manual review of chat room conversations, and enables the monitoring of multiple chat room servers.

III. System Design and Implementation

This section describes the network perspective and system architecture of the chat room monitor. The filtering and evidence preservation processes are also detailed.

A. Network Perspective

The network perspective of the chat room monitor is shown in Figure 2. Three chat room servers (A, B and M) and an instant message peer (bottom of Figure 2) use Internet connections to support communication between various clients (top of Figure 2).

* To whom correspondence should be addressed (email: sujeet@utulsa.edu).

ISBN 0-7803-9814-9/$10.00 ©2001 IEEE

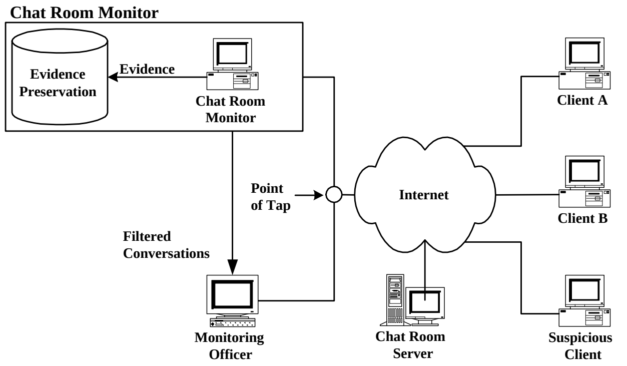

Figure 1. Chat room monitor schematic.

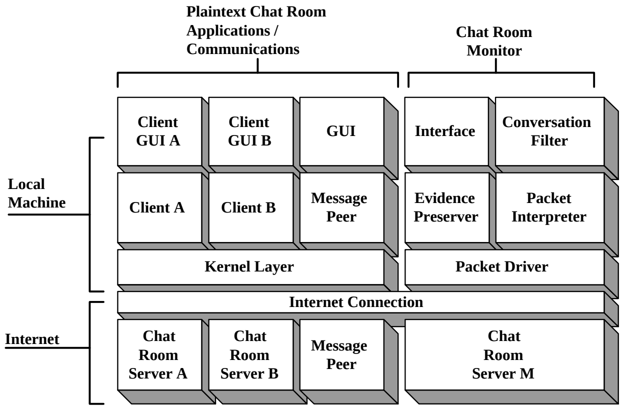

Figure 2. Network perspective.

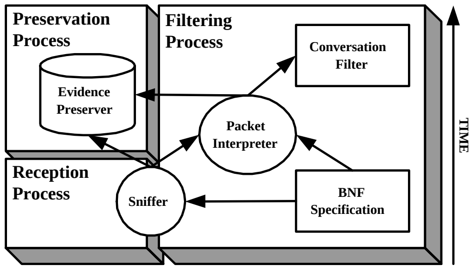

Figure 3. System architecture.

The chat room servers A and B provide services to clients A and B, respectively. The instant message peer in Figure 2 enables clients (peers) to communicate directly over Internet without the use of a server [15]. Communications from the chat room server enter the client's local machine in the form of raw plaintext packets. The kernel layer network protocols translate the raw data to the application level [4]. The chat room client receives the translated packets and interprets the commands as either actions or conversations occurring within the corresponding chat room. The graphical user interfaces (GUIs in Figure 2) present interpreted chat room data in windows corresponding to their chat room sources.

When clients participate in chat rooms, their actions and conversations are encoded as packets at the client application level and sent to the kernel for further packaging at the transport and network levels [4,15]. Next, the packets are sent across the Internet connection on the local machine to the chat room server (Figure 2). The server then presents the corresponding information to the appropriate clients in the chat room.

The chat room monitor sniffs traffic on the Internet connection using a packet driver located at the kernel layer (right half of Figure 2). The packet driver carries a unique network protocol stack for the encoding and decoding of packets. The chat room monitor can either sniff communications between chat room server A and client A or connect to server M for the automated monitoring of chat room conversations. Raw packets are copied off the network interface card of the local machine, translated in the packet driver, and sent to the evidence preserver for storage and the packet interpreter for further translation. The translated chat room packets are filtered and analyzed in real-time, and the interface immediately alerts monitoring officers to suspicious content.

B. System Architecture

The architecture of the chat room monitor is presented in Figure 3. It implements three main processes: reception, filtering and evidence preservation. The design employs a sniffer, a packet interpreter that relies on a BNF specification of packets, a conversation filter and an evidence preserver.

The sniffer conducts the reception process (bottom left of Figure 3), providing an efficient and scalable means to interface with all plaintext chat room services. The prevalence of different operating system kernels and chat room interfaces creates interoperability problems for automated chat room monitoring. By parsing chat room data at the packet level rather than the presentation level, packets are processed before they reach the kernel. This avoids interoperability problems.

The sniffer serves two purposes: the reception and delivery of chat room packets. Reception takes place in real-time, allowing for the filtering of conversations and the preservation of evidence. The delivery aspect to the sniffer component acts as a chat room client emulator, initiating and sustaining communications with servers.

Filtering (right half of Figure 3) begins at the same time as the reception process. The process screens two types of information: packets and conversations. To increase the efficiency of the overall system, the filtering process dictates what packets are accepted (by the sniffer). The conversation filter examines all communications between clients within the chat room, scrutinizing them for possible suspicious activity.

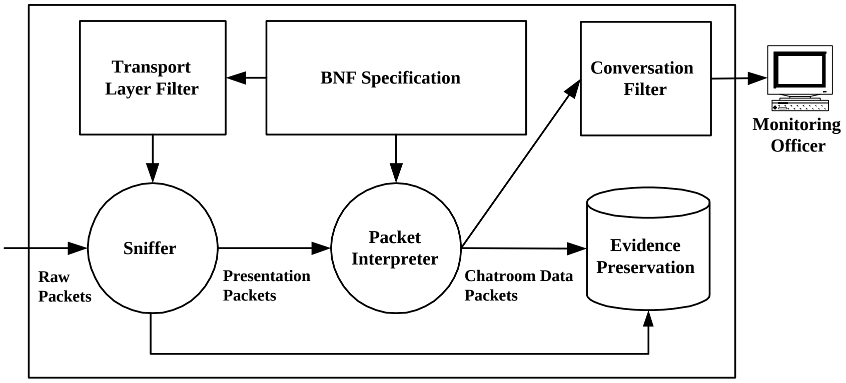

Figure 4. Filtering and evidence preservation.

The preservation process (top left of Figure 3) is initiated after the sniffer has accepted chat room packets. The evidence preserver stores all data pertaining to chat room conversations. It incorporates public key encryption (PGP) [7] and hashing techniques (CRC-MD5) [9] to prevent evidence tampering. Packets and accompanying data collected in the evidence preserver can be used to re-construct chat room conversations in their entirety.

C. Filtering and Evidence Preservation

Details of the filtering and evidence preservation processes are presented in Figure 4. The chat room monitor contains two filters: a packet filter and a chat room conversation filter. The packet filter consists of a transport layer filter (within the sniffer) and the packet interpreter. The transport layer filter screens incoming raw packets received by the sniffer. These packets are then copied and sent to the packet interpeter and the evidence preserver.

The BNF specification object controls the operation of the packet filter. It supplies the transport layer filter with rules for accepting raw packets, e.g., port numbers and IP addresses. The packet interpeter processes the accepted data after the sniffer has converted raw packets (byte representation) into plaintext (presentation level). The interpreter uses the BNF specification to translate chat room packet data into a common language that the conversation filter can understand.

The evidence preserver accepts data at two points during the filtering process: after the transport layer filter and before the conversation filter. Accepted raw packets copied off the wire are sent to the preserver as a preliminary record of the chat room conversations. After the packet interpreter has translated the presentation packets, timestamped copies of the translated packets are sent to the conversation filter and to the evidence preserver. The evidence preserver therefore contains two versions of each packet: raw and translated. Raw packets stored in the device can be re-played through the filter process and validated against the timestamped translated packets.

The conversation filter uses an intelligent keyword rule set to parse all conversations from various chat room clients. The rule set can be updated and maintained by the monitoring officer from a remote interface terminal. Once the conversation filter has detected a suspicious client, it notifies the monitoring officer and backtracks through the collected logs to locate other clients who communicated with the suspect.

IV. Applications

The chat room monitoring system currently aids in investigations of sex crimes. However, it can be applied to actively monitor chat rooms populated by hate groups, hackers, child pornographers and drug traffickers. Applications for packet sniffing based systems also exist in corporate environments.

The chat room monitor can analyze copious amounts of profile information pertaining to participants. The sniffer, acting as a packet constructor, can use a BNF specification to request participant profile information stored on chat room servers. The conversation filter can then parse the profiles and target suspicious clients. Currently, profile information is manually captured and analyzed by law enforcement agents.

The chat room monitor can also serve as an alarm system for potential attacks. The monitor residing in an IRC hacking channel can openly (through a client connection) or surreptitiously (through a server resident application) filter conversations that discuss attacks.

To address liability issues, Internet service providers and chat room hosts can use the monitoring system to filter all conversations and record illegal activity when detected. Corporations can also use the system to monitor the release of sensitive information, e.g., company secrets and stock information. Because the system is based on packet sniffing, corporations can monitor all channels of communication within their networks (ICQ, email and chat rooms). Alerts detailing the source and content of sensitive information can then be sent to appropriate corporate administrators.

If necessary, the monitor can be augmented to deny service to the source of a leak, immediately preventing any further release of sensitive information.

V. Conclusions

Current practices for chat room monitoring provide no principled or efficient means to log chat conversations for evidentiary purposes. Manual monitoring is tedious, and evidence collection techniques may not meet the stringent demands of criminal and civil proceedings in the information age.

The chat room monitoring application described in this paper provides automatic logging and evidence preservation. The separation of the human agent from the logging process promotes data integrity and the admissibility of evidence. Furthermore, real-time conversation filtering frees law enforcement agents from the tedium involved in manual monitoring, and enables them to monitor multiple chat room servers. Packet sniffer based monitoring and analysis have applications in other areas, including detecting network intrusions and attacks, filtering indecent content and preventing the electronic dissemination of sensitive corporate information.

References

[1] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, New York, 2001.

[2] A. Bacard, *The Computer Privacy Handbook*, Peachpit Press, Berkeley, California, 1998.

[3] D. Barrett, *Bandits on the Information Superhighway*, O'Reilly & Associates, Cambridge, Massachusetts, 1996.

[4] D. Comer (Ed.), *Internetworking with TCP/IP: Principles, Protocols and Architecture*, Prentice-Hall, Upper Saddle River, New Jersey, 1991.

[5] E. Deck (Ed.), *Acquisition of New Technology: A Best Practices Guide*, vol. 1(1), International Association of Chiefs of Police, Alexandria, Virginia, 2000.

[6] D. Denning, *Information Warfare and Security*, Addison-Wesley, Reading, Massachusetts, 1999.

[7] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly & Associates, Cambridge, Massachusetts, 1994.

[8] D. Icove *et al.*, *Computer Crime: A Crimefighter's Handbook*, O'Reilly & Associates, Cambridge, Massachusetts, 1995.

[9] A. Menezes *et al.*, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Florida, 1996.

[10] S. Northcutt, *Network Intrusion Detection: An Analyst's Handbook*, New Riders, Indianapolis, Indiana, 2000.

[11] D. Parker, *Fighting Computer Crime*, John Wiley & Sons, New York, 1998.

[12] E. Sinrod, *et al.*, "Cyber-crimes: A practical approach to the application of federal computer crime laws," *Santa Clara Computer and High Technology Law Journal*, vol. 16(2), Santa Clara University School of Law, Santa Clara, California, 2000.

[13] P. Stephenson, *Investigating Computer-Related Crime*, CRC Press, Boca Raton, Florida, 1999.

[14] U.S. Department of Justice, *Federal Guidelines for Searching and Seizing Computers – Evidence.*

[15] D. Vaskevitch, *Client/Server Strategies*, IDG Books, New York, 1993.

ISBN 0-7803-9814-9/$10.00 ©2001 IEEE