# The Design and Implementation of Database Audit System Framework

Kehe Wu
*School of Computer Science and Technology*
*North China Electric Power University*
*Beijing , China*
epuwkh@ncepu.edu

Liang Hua, Xiaoxiang Wang, Xuewei Ding
*School of Computer Science and Technology*
*North China Electric Power University*
*Beijing , China*
hualiang19900415@163.com

*Abstract*—**For the information system, Database acts an important role in it. How to find a way to audit the operation of the database is becoming more and more important. An effective database auditing system can not only reduce the potential security risk, also make it possible to trace the source when errors happen. In this paper, we design and implement an effective audit framework. To avoid causing any database performance delays, bypass mode is especially useful for the auditing system. Compared with the traditional audit system, our approach has more advantages for example using zero-copy to acquire and reorganize data. In addition, we have a flexible policy to audit more effectively.**

***Keywords-database; auditing system; zero-copy; policy***

## I. Introduction

Database Security Audit is an effective means to ensure the security of the database when the data is modified maliciously and causing system problems for the database, in this case, DBA can trace the process of modifying data, determine the extent and scope of the data corruption, make an appropriate solutions in order to minimize the recovery period and reduce losses as possible as we can.

Currently on the market, there are a wide range products of database security audit system. Function of domestic manufacturers products is basically the same, and they all have problems and defects. The type of plug-in modules embedded in database management system is widely used in Database vendors, because it can deal with the internal processing and control it more accurate and deeply. But among those products one usually can't fit for others, therefore it can not provide a good support for multiple database environments. In this paper, we provide a system which can deal with different types of database's network traffic according to different data protocols so that it easily solves the problem of multiple database auditing. In order to avoid affecting the transmission performance of the database itself, we prefer to choose the bypass mode, so that it won't increase the burden on the database, on the other hand it is more convenient to deploy and advantageous to the system extension.

The purpose of the database audit system is to record every access to the database effectively and immediately, generate detailed analysis of audit records, so that the system can use the five elements to record every incident: who, when, where, what, how. Flexible audit system needs to be able to determine which database should be recorded or not, so it requires the audit system based on a flexible, variable and diverse strategies in order to achieve effective audit records to the database.

This article will focus on how to design a database audit framework to support multiple protocols, how to quickly capture and filter network traffic, how to design a effective way to maintain TCP sessions. In this paper we will not focus on process of analyzing for database communication protocol such as TNS, TDS, DRDA.

## II. Systems Analysis and Design

Database auditing system described in this article contains two parts:Web management platform and protocol analysis module. The second part can be further subdivided into six parts: Network Data Capture module, TCP / IP data processing module, configuration module, strategic matching module, protocol processing module, SQL statement analysis module. Fig 1 shows the database auditing system design.

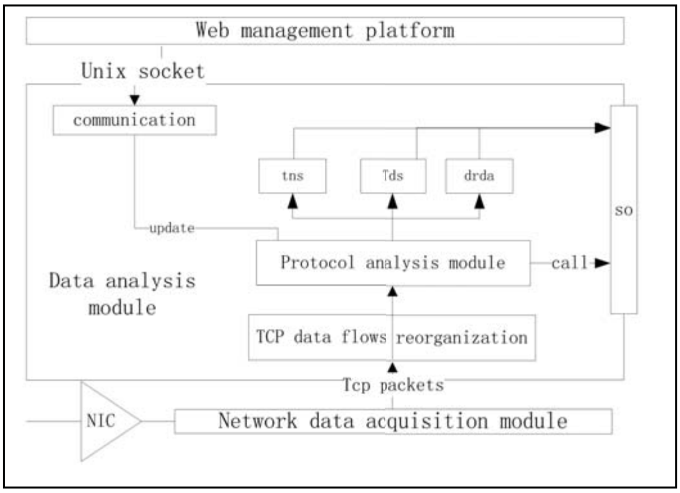

Figure 1. database audit system architecture

### A. Network data acquisition module

In traditional packet processing, network data needs to be copied twice: first it transfer from NIC to operating system's memory buffers, then if comes to the user buffers. Copy in memory spent a huge overhead, which seriously affected the

**978-1-4799-3279-5 /14/$31.00 ©2014 IEEE**

performance of packet acquisition module. After we compared different types of capture modes, it turns out to be zero-copy technology is more effective. It make possible data from NIC can be mapped directly to the application by DMA device, in the process, it saves the CPU resource and dramatically reduces the overhead due to a large number of system calls and data copies between kernel and user spaces [1]. We can Improves the processing speed of the system by this way.

There are two key points of zero copy technology:

*1) Data in transmission can reach to the address spaces which are pre-allocated by system kernel, so that it will not handled with CPU.*

*2) It can make the address spaces of the data packets mapped to the user buffers which are stored in the kernel, then we provide a way for example the API like the system to make the user application can access the memory directly to avoid memory copy of the system call.*

Zero-copy module in this system is based on the modification of E1000E NIC driver. We add the function of filtering network traffic by rules so that we can deal with the packets we want. Zero copy module can be devide into some parts below.

First we create a virtual character device which is used to apply for the space storing the data frame and get some information from the data. After creation we start the device. Then we apply for the kernel buffers and user buffers via mapping mechanism to the physical address [2]. At last, we can use it after we initialize this space. In the process of managing this spaces by ourselves application, we can transfer the data packets from the NIC to the physical address, but the most important is that user buffer can access the data directly without handled by the CPU. Fig 2 shows the simple flow chart.

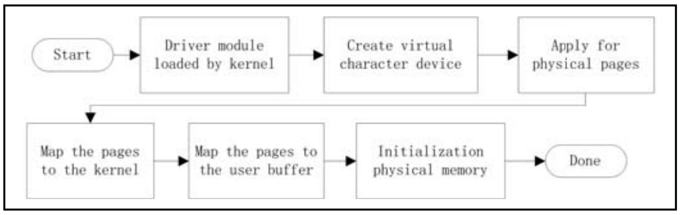
Start
Driver module loaded by kernel
Create virtual character device
Apply for physical pages
Map the pages to the kernel
Map the pages to the user buffer
Initialization physical memory
Done

Figure 2. Flow chart of the initialization of memory and NIC driver

The structure of the physical page contains the actual physical address of the page and the other information of usage. When we mapping physical pages to the kernel space, the essence of what we did is writing the physical page address to kernel page table contiguously. We can get the logical address mapped from the physical locations by page, and the location of the first piece in physical pages is represented by the logical start address in the kernel. Thus the actual physical address is connected to the logical address. The process of mapping physical memory page to user space is similar to mapping to the kernel space, the only difference is that the address written to the space is the pages of application instead of the pages of kernel itself, so that the logical address in the user domain is also connected to the physical memory pages. Obviously the three address can successfully convert to another which makes the zero copy technology possible.

### *B. TCP data flows reorganization*

Data flows reorganization is the most important part in the process of dealing with the data which is guarantee for the the implementation of programs. Packets from the network sometimes will cause packet disorder, also consider the network delay some packets needs to be sent more than once. In order to ensure the normal execution of the application, we need to restructure a complete and ordered data stream. In our system, design and implementation of TCP reassembly is based on shared memory [3].

#### *1) The management of the flow table*

When the driver application apply for memory space, it will apply the physical page first, then map the actual physical page to kernel space and user space. This space are contiguous logical addresses in both kernel buffers and user buffers. In the following discussion, memory space refers to a logical contiguous address. In the process of initialization, space is divided into five part, Fig 3 shows the memory partition:

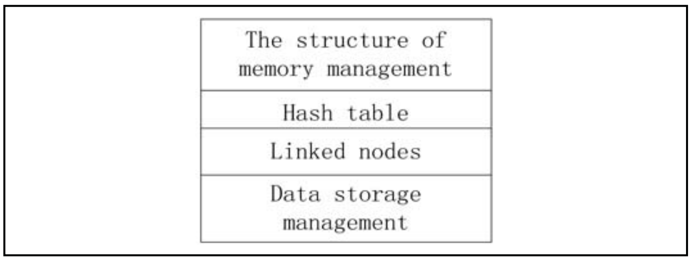

Figure 3. memory partition

*a) The structure of the memory management*

This structure record important information of the memory space which include two address information, one is the starting address for the kernel to access and the other is the starting address for the user buffer.

*b) Hash table*

We use hash table to manage the current connections so that when a new packet come, we can find and locate the packet belong to which session quickly. The hash table structure is similar to the table in linux kernel.

```
struct list_head
{
  struct list_head *next, *prev;
}
```

*c) Linked nodes*

It maintain data flow according to the characteristics like IP address and the historical information such as status of connection. It help the data flow can be submit to the upper in order.

*d) Management of data store*

The management of the spaces for storing data is very important, it is the first step for providing physical address to the DMA and logical address to the user application. Also it makes possible that network traffic can be copied to the user buffers without handled by CPU.

#### *2) The processing of reassembling*

*a) Hash search*

In this module, we will check the information of each packet and then calculate the hash index in order to find the connection in the link list. If it return nothing matched, then we regard it as a new connection and put it into the link list. Otherwise, we consider this packet is belong to an exist connection. In some case, we will drop the packet if the connection has a flag for free. Then we will deal with this packet if it is expected. When it comes false, we put the packet on the link list and waiting for others. If the packet is what we want, then we can deliver the packet to the next step.

In order to simplify the process of hash search, the packets need to be mapped to the same hash code if they belong to the same connection. Therefore, to ensure the hash algorithm for the connection between the two sides are symmetrical treatment. We can differ a TCP connection by source IP, destination IP, source port and destination port, so we can use the four member group as the keys of the hash algorithm.

*b) State Machine for TCP data flow*

In the process of TCP reassembly, each operation is related to the connection state directly, we should find a way to manage them. As we known, state Machine for TCP can describe every state in the TCP data flow [4].

We don’t need to consider two sides negotiated information because we capturing packets by bypass mode. Here we only need to focus on these TCP states: SYN, FIN, RST, ACK. Fig 4 shows the status switching figure.

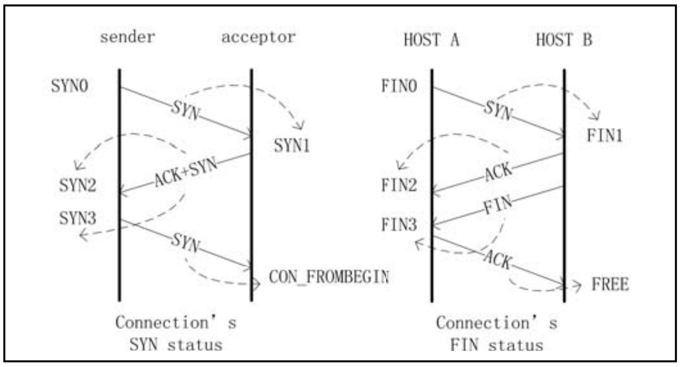

Figure 4. connection’s status switching figure

Three-way handshake of TCP has these steps:

- originator of the connection sent a packet of SYN
- receiver machines respond a packet of ACK and SYN
- originator of the connection sent back a packet of ACK

4-way handshake of TCP disconnection has those steps:

- Host A sent a packet of FIN
- Host B respond a packet of ACK
- Host B sent a packet of FIN immediately
- Host A respond a packet of ACK

*3) Byte order process*

In the data reorganization process, the sort operation is based primarily on byte order of data packet. In the TCP protocol, every packet contains two byte order, one is the first byte order of the packet named SEQ and the other is first order of the packet which we expect it come next called ACK_SEQ. The code of dealing with the byte order is as follows:

```
if SYN
exp_seq+=1;
  else if FIN
    exp_seq +=1;
  else if (ACK in FINx OR SYNx)
    do nothing;
  else
    exp_seq +=app data length;
```

Now we have finished the process of byte order, then the data packet is in order and we can transfer it to the upper module.

### C. Policy configuration management module

For the audit system, they are based on the policy. With the policy we can decide which database we should audit and which we can ignore to improve audit efficiency. After initialization of the project, NIC driver calls the database interface functions to load the information of policy which is stored in the local database. The most valuable information is the IP and the port so we can use this filtering the packets.

### D. Web management platform

Administrator do their work via the web. The main role of web management platform is to provide intuitive and accurate audit data to auditor. The second role is to provide vivid and rich charts and tables to let auditor know the status of audited database and the potentially dangerous of audited database. When a policy is finished, it will send a message to the policy configuration module by the JNI. Then the message will be received according to the unix socket. Here is the data structure of the message:

```
typedef struct __BLOCK_HDR
{
  unsigned int magic;
  unsigned int code;
  unsigned int param1;
  unsigned int cnt_length;
}BLOCK_HDR;
```

### E. Protocol analysis module

Our system can audit different type of databases according to their own protocol, such as the TNS for the oracle, the TDS for the MS SQL, and the DRDA for the DB2. Here we don’t focus on the specific parsing process. After the SQL is analyzed we can match this sentence with our policy. We will find if there are any key words in this sentence such as the operation type, Sensitive field, and the database user name.

We choose the AC algorithm as the string matching algorithms because it is very efficient. The core of the algorithm is about three tables for search: goto table, failure table, and the output table [5]. It proposed four methods, three of them are using for searching the tables and the left one is for the algorithm itself [6].

If the match is successful, we record the SQL into the local database otherwise we drop it and deal with the next process. The above is the entire audit process.

## III. SYSTEM DEPLOYMENT

System deployment will take the bypass mode. All the data flow from client to server will mirror to database audit system by the mirror port of switch. The system only record and audit the database communication packets which are mirrored by switch.

## IV. CONCLUSIONS AND FUTURE WORK

This paper designs and implements a database auditing system which can deal with different type of databases. We mainly talked about the zero copy technology and the tcp reassembly. Compared with native auditing mechanism in database, our approach has an obvious advantage that we don’t cause any delay to the database itself. But on the other hand, this system also has many shortcoming, for example when the data is encrypted by the client we can not analyze the information any more. We will study more on this point in future.

## REFERENCES

[1] Liu Tianhua, Zhu Hongfeng, Chang Guiran, et a1. "The design and implementation of zero — copy for linux" Eighth International Conference on Intelligent Systems Design and Applications. 2008: 121 —123.

[2] Welsh M., Basu A., von Eicken T.. "Incorporating memory management into user level network interfaces." Cornell University Ithaca , NY, USA: Technical Report TR9721620 , 1997

[3] Shaoqiang Wang, DongSheng Xu and ShiLiang Yan. 2010. “Analysis and application of Wireshark in TCP/IP protocol teaching”. E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on (Volume: 2 ).

[4] Fusion embedded TCP/IP stack, http://www.unicoi.com/fusion net/fusion tcpip.htm.

[5] Navaro G R M. "Flexible Pattern Matching in Strings". Cambridge: Cambridge University Press,2002

[6] Miao Chang-sheng , Chang Gui－ran , Wang Xing-wei . "Filtering Based Multiple String Matching Algorithm Combining q-Grams and BNDM" Proceedings of the 2010 Fourth International Conference on Genetic and Evolutionary Computing. ACM, 2010 : 582−585