Audit Design Based on Protocol Analysis of ORACLE Database

Xiao-liang ZHANG and Hai-su CHEN

School of control and Computer engineering, North China Electric Power University, Beijing, China

Email: zhanghino@126.com, chen_haisu@sina.com

Corresponding author

Keywords: Database audit, TNS protocol, Oracle database, Database security.

Abstract. As companies rely more and more on information systems, security of database becomes increasingly important. The database has a security mechanism to ensure completeness and correctness, however attackers or unauthorized users always want to operate database through non-formal ways. Therefore, it’s important to record user’s operations worked on the database.

This paper puts forward a new way to record user’s actions in the audit system, which accesses to the network through the bypass monitoring method, this way audits database by setting the policy. The advantages are that the audit system does not affect the communication between the client and the database server, and database problem does not affect the database audit system. In this paper, Oracle database communication protocol TNS will be example to describe the audit system. This paper also introduces TNS in the protocol framework and shows its level position, then designs a framework for the audit system according to the network model.

Introduction

A database is often the most strategic asset of an information system that needs to be protected. Attackers and illegal users always want to operate the database through informal ways, so it is very important to record operations on the database.

Audit capabilities provided by Oracle database tend to record a large number of audit operations, lack of effective analysis tool when faced with massive data. It is not easy to find attacks, illegal access and other aspects of security issues if database administrators to analyze the audit data in person. At the same time, the computer system itself and other hardware failures may cause the correctness and integrity of the data was destroyed.

In this paper, we design a database audit method to meet the needs of security audit. It has advantages of not affecting communication between the client and the database, the failures of the database and the audit system will not affect each other, and the data captured by the audit system can be further analyzed.

Related Protocol

OCI and NET8 Protocol

Oracle provides a complete network interconnection service, which is called Oracle Net Service, also known as NET8 service [1, 2]. Oracle network protocol achieves the top three layers of OSI, Oracle client OCI and server-side OPI make up the session layer. This layer is responsible for establishing a complete SQL session between client and server, and the process is as follows:

- To parse the SQL statement, and analyze the grammar,
- Apply cursor for the SQL statement,
- Bind variables for SQL statements,
- Get the metadata information of record set from server’s data dictionary,
- Execute SQL statement,
- Return the result and close the cursor.
TTC (Two-Task Common) layer corresponds to the presentation layer, which represents the character set and data type conversion between client and server [3]. As showed in Fig. 1, the following layers are NET8 layer, and the Oracle does not put this layer on the OSI model, but as a separate layer. This part consists of 3 components: Net Interface, Routing/Naming/Auth and TNS. Net Interface transmits data, while the other two layers responsible for transparent support for the underlying protocol.

![Figure 1. TNS protocol hierarchy](image)

**Analysis of Protocol**

**TNS Protocol Connection Process**

The TNS protocol defines the language used between the database and the client, allowing the implementation of services such as authentication [4]. Database software is responsible for the execution of the user’s verification process, however, because messages are transmitted back and forth between the client and the database, they must through multiple layers of software.

Oracle clients receive user’s specified parameters and pass them to the TNS layer. To initialize this connection, the TNS layer sends a connection string containing client information to a database, and the second half of the string initialize user’s connection of operating system.

After initiating a connection, client and server negotiate which authentication protocol is used. The client sends a secure message to request the desired authentication mechanism. The database would confirm whether the requested service is valid or not in reply. The above connection can be shown as the interaction process in Fig. 2.
After any requirements of the protocol by the security network services, database users are verified by using Oracle password protocol [5]. This protocol is used to avoid the third party access to the password.

**TNS Protocol Packet Structure**

In this paper, the packet structure is drawn by capturing a large number of packet and analyzing the meaning of the field corresponding to relevant type packets. You can see the entire data format frame through WireShark tool, so the obtained results in this paper are based on the capture tool.

A database audit system NIC acquired data starts from the data link layer, through the network layer and the transport layer, finally to the application layer TNS protocol [6, 7], shown in Fig. 3.

![Figure 3. TNS packet](image)

TNS data contains a generic header, which contains information about the checksum, length, type and others. Different types of data to achieve different functions of data transmission.

As showed in Fig. 4, the shaded portion has 8 bytes, which is a fixed-length header. Fig. 5 show the structure of the header. TNS packet is composed of two parts, which are fixed length and load.

![Figure 4. Resend type of TNS packet](image)
Packet length: 2 bytes, the unit is Byte, it represents the length of the entire TNS packet (including header and load section). The maximum packet length is 4086 bytes, referred to as session data unit (Session Data Unit, SDU), and the maximum transmission unit (Transport Data Unit, TDU) is 32767 bytes. TDU will not be smaller than the SDU.

Packet checksum: 2 bytes. Currently are zero filling.

Packet type: the type of TNS packet, 1 byte, which is described in Table 1.

Table 1. TNS packet types

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
<th>Type</th>
<th>Meaning</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connect</td>
<td>6</td>
<td>Data</td>
<td>11</td>
<td>Resend</td>
</tr>
<tr>
<td>2</td>
<td>Accept</td>
<td>7</td>
<td>NULL</td>
<td>12</td>
<td>Marker</td>
</tr>
<tr>
<td>3</td>
<td>ACK</td>
<td>8</td>
<td>Unknown</td>
<td>13</td>
<td>Attention</td>
</tr>
<tr>
<td>4</td>
<td>Refuse</td>
<td>9</td>
<td>ABORT</td>
<td>14</td>
<td>Control</td>
</tr>
<tr>
<td>5</td>
<td>Redirect</td>
<td>10</td>
<td>Unknown</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reserved bits: 1 bytes are not used at present.

Header checksum: 2 bytes. Currently are zero filling.

Data: according to the specific TNS type analysis.

Application of TNS Protocol

Design of Database Audit System Model

When the main program of database audit system grabs network packet and determines the protocol is Oracle communication protocol, the main program will send the data to TNS protocol analyzer module. The module will restore SQL statement according to the protocol content, and then hand over to the SQL statement module.

The function of data capture and analyze is to monitor all operations of database in the network, restore SQL statements by analyzing protocol, analyze SQL statements and extract operations on the database according security information, to lay the foundation for future database security audit and risk control. Because each database uses a different communication protocol specification, for which developer should develop different protocol analyzer modules, to complete audits of all database systems. Crawling network data is divided into the following steps: crawling data, filtering SQL data, analyzing database protocol, reunite and restore sessions, analyze SQL statements, stored data.

Process of Crawl Network Packets

The tasks of data acquisition and analysis are: real-time monitoring of network data on a database server. Protocol analysis based on demand, restoring the user’s operation of the database server. As showed in Fig. 6. The flowchart of crawling data packet is explained as follows:

- To capture the data packets in database server communication network by crawling the network data packets
- Filtering SQL data, reducing the burden of data packet capture and filtering the data related to
the target database operations.

- To resolve the information of the SQL statement
- Reunite and restore sessions. Analyze protocol identification in data packets, recombine the packets that have session information, and analyze the header, retransmission, consultation, response and other information, a complete session record is obtained.
- Analyze information in the SQL statement such as: keywords, command, design specific tables and others.

**Crawl Packets of Server**

One of the most important steps in data acquisition and analysis is to crawl the server data packets, which is the basis for obtaining the user’s operations. There are two main ways to capture data packets: in Ethernet communication environment, the network card is set to “mixed mode”, then this card can receive all data packets in the local area network. Another is to set up the mirror port on the switch or router in the exchange network, to achieve crawling through the hardware.

In this paper, the architecture of the database audit system device as shown in Fig. 7, the external network interface of database audit system is connected to the mirror port, and mirror port attach to database server, so monitoring program can crawl data of operation on database without changes on network, it does not affect the normal operation of actual business of the network.

**Filter SQL Data**

Filtering SQL data refers to filtering out relevant data onto operations according to the policy configured by the administrator. Only the packets that are in line with the policy are delivered to the next module to handle. The data that do not conform to the requirements are discarded, so that the data can be captured in the high-speed network communication, and the work efficiency is improved.
Database Audit Validation

As showed in Fig. 8, the background monitoring program detects the session information that the client connects to the server. The connection information showed in Fig. 8 verifies the correctness of resolving Resend packets.

![Figure 8. Session connection information](image)

When auditors configure the audit policy, data not meet conditions will be discarded. The result of background program is showed in Fig. 9. If the data meet the conditions will be recorded, as showed in Fig. 10, the left column is related tables, and the right is SQL statements that have been resolved successfully.

```sql
INFO.db_sql = SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS
INFO.db_target_table = USER_ROLE_PRIVS
INFO.db_user = SYSTEM
insert into db_ia_study values nextPage(db_ia_study_seq), 'SYSTEM',
('USER_ROLE_PRIVS', 'SQL', 1);
insert into db_fa_study table ok.
insert a new record
[*] table name is: USER_ROLE_PRIVS
retval = 0
[
sql] {SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS,
[sql] audit.commit enter,
[sql] audit.commit not match any policy,
[sql] audit.commit exit.
```

![Figure 9. Query statement](image)

<table>
<thead>
<tr>
<th>operate_time</th>
<th>result_rows</th>
<th>related_tables</th>
<th>sql</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0</td>
<td>syncatables</td>
<td>select TabName as Name, TabSchema, v.Definer, v.View</td>
</tr>
<tr>
<td>856</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>TEST</td>
<td>select idd FROM TEST</td>
</tr>
<tr>
<td>1070</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>749</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>TEST</td>
<td>insert into TEST values('大刀', '叮')</td>
</tr>
<tr>
<td>588</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>672</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TEST</td>
<td>insert into TEST values('2', 'd', 'a')</td>
</tr>
<tr>
<td>1103</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>56303</td>
<td>0</td>
<td>SET CURRENT LOCK TIMEOUT WAIT 1</td>
<td></td>
</tr>
<tr>
<td>40410</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current server from sysibm.sysdummy1</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current database from sysibm.sysdummy1</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current schema from sysibm.sysdummy1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>sysibm.sysdummy1</td>
<td>select current path from sysibm.sysdummy1</td>
</tr>
<tr>
<td>1843</td>
<td>0</td>
<td>questsoftware.toadse</td>
<td>SELECT type, name FROM questsoftware.toadsecurity</td>
</tr>
</tbody>
</table>

![Figure 10. Query statement record](image)
Conclusions
This paper mainly studies parsing TNS protocol based on database audit system. This paper introduces the principle of working and connecting processes of TNS protocol, and the structure of TNS packet is introduced. We have a deeper understand of communication between Oracle database server and client through the interpretation of TNS packets and parsing of TNS protocol. Then put TNS protocol module into the entire database audit system framework, to present the entire audit system.

The database audit system is designed based on the previous model, in this paper the TNS protocol is studied, but the TNS protocol is not open source and I am limited, the study is not perfect. This study is only through packet capture tool and part of open source code, is not clear about all meaning of the fields and not give specific communication process, there is still a lot of work needs to be studied.

Acknowledgment
This work is supported by “The Fundamental Research Funds for the Central Universities 2015MS34”.

References