# DATABASE SECURITY IN PRIVATE DATABASE CLOUDS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR ÇINAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INFORMATION SYSTEMS

JULY 2015

Approval of the thesis:

**DATABASE SECURITY IN PRIVATE DATABASE CLOUDS**

submitted by **ONUR ÇİNAR** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Nazife Baykal _____
Director, Graduate School of **Informatics Institute**

Prof. Dr. Yasemin Yardımcı Çetin _____
Head of Department, **Information Systems**

Prof. Dr. Adnan Yazıcı _____
Supervisor, **Computer Engineering Department, METU**

Prof. Dr. Nazife Baykal _____
Co-supervisor, **Information Systems, METU**

**Examining Committee Members:**

Prof. Dr. Yasemin YARDIMCI ÇETİN
Information Systems, METU _____

Prof. Dr. Adnan YAZICI
Computer Engineering Department, METU _____

Prof. Dr. Nazife BAYKAL
Information Systems, METU _____

Prof. Dr. Ahmet COŞAR
Computer Engineering Department, METU _____

Assist. Prof. Dr. Tuğba TAŞKAYA TEMİZEL
Information Systems, METU _____

**Date:** _____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: ONUR ÇİNAR

Signature :

# ABSTRACT

DATABASE SECURITY IN PRIVATE DATABASE CLOUDS

Çinar, Onur

M.S., Department of Information Systems

Supervisor : Prof. Dr. Adnan Yazıcı

Co-Supervisor : Prof. Dr. Nazife Baykal

July 2015, 88 pages

Cloud computing and systems are very popular in today's Information Technology and Systems and have grown extremely. Cloud computing helps organizations solve increasing IT costs such as their licenses, power consumption, physical protection, by providing better standardization, higher benefit, greater performance, and quick responses of information services. This thesis presents the study of ensuring database security in cloud computing, protecting data especially from internal attacks, and managing high volumes of log data. Security requirements of a network system differ from each other according to the system which you are using. Carrying vital importance for our private and secret information, cloud systems must provide extra security solutions for database systems with the organizations. While using firewalls, intrusion detection systems on the network for external attacks, improved access controls must be thought for internal attacks. Using database actions logs and audit can be very helpful on developing a database protection system. However, storing these logs will cause a storage and performance problem. The other problem is to manage many databases at the same time. This system provides managing many databases at the same time from the same place. Logs from each database is transmitting to the central point asynchronously.

Administrating database protection solution from a central point in cloud helps us to manage security of databases and handling huge volumes of data.The performance of our proposed system has been evaluated against one of the most commonly used products on the market, Oracle audit Vault Server and better results have been obtained on some topics in this thesis.

Keywords: Database Security, Private Database Clouds, Infofence

# ÖZ

ÖZEL VERİTABANI BULUT SİSTEMLERİNDE VERİTABANI GÜVENLİĞİ

Çinar, Onur
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Prof. Dr. Adnan Yazıcı
Ortak Tez Yöneticisi : Prof. Dr. Nazife Baykal

Temmuz 2015 , 88 sayfa

Bilişim teknolojilerinde bulut bilişimi ve sistemleri son zamanlarda hızlı bir şekilde gelişmekte ve önem kazanmaktadır. Bulut bilişimi kurumlara, firmalara, ve sektördeki diğer kuruluşlara, artan bilişim maaliyetine (lisans masrafları, enerji tüketimi, sunucuların fiziksel güvenliği,..), daha yüksek standartlarda daha iyi ve hızlı hizmet sunarak çözüm bulmaktadır. Bu tezin amacı bulut bilişiminde yer alan veritabanlarının güvenliğini, içeriden ve dışarıdan gelebilecek olan tehlikelere karşı önlem alan bir system geliştirmektir. Bulut sisteminde çok önemli verilerimizin bulunduğu veritabanları için daha kapsamlı önlemler almalıyız. Dışarıdan gelebilecek tehlikelere karşı ağ içerisine Firewall ve IPS yerleştirebiliriz, fakat veritabanına erişebilen kişiler için ayrıca önlemler almalıyız. Bunun için de veritabanında gerçekleşen tüm aktivitelerin logunu toplayıp, sisteme bağlanan kişileri görevlerine göre yetkilendirmeliyiz. Zaman içerisinde topladığımız bu loglar, yönetmesi çok zor bir büyüklüğe ulaşacaktır. Ayrıca bulut bilişiminde çok sayıda veritabanı olabilir ve onları aynı anda aynı yerden yönetmek durumunda kalabiliriz. Yapmış olduğumuz system ortak bir sunucudan, bulut sisteminde olan tüm veritabanlarına bağlanıyor ve onların toplamış oldukları logları asenkron bir şekilde

kendi üzerine alıyor. Bu şekilde büyüyen loglar veritabanları üzerinde performans sıkıntısı yaratmayacak ve aynı anda bir çok veritabanını ortak bir noktadan yönetmiş olacağız. Son olarak geliştirdiğimiz sistemi, piyasada bulunan en çok kullanılan ürünlerden biri olan Oracle Audit Vault Server ile karşılaştırdık ve bazı alanlarda daha iyi sonuçlar aldığını gördük.

Anahtar Kelimeler: Veritabanı Güvenliği, Özel Veritabanı Bulut Sistemleri, Infofence

*To my wife*

# ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Adnan Yazici for his guidance, support and motivation throughout the development of this thesis; and for giving me freedom I needed for my graduate studies.

I would like to express my thanks to the jury members, Prof. Dr. Yasemin Yardimci Çetin, Prof. Dr. Nazife Baykal, Prof. Dr. Ahmet Cosar and Assist. Prof. Dr. Tugba Taskaya Temizel for reviewing and evaluating my thesis.

I would like to thank to Helezon Computer Company and our manager Mr. Haluk Günçer for supporting my academic studies.

I would like to thank to my family for bringing me up and making me who I am with their love, trust, understanding and every kind of support throughout my life.

Finally special thanks to my wife, Nesibe ÖZEN ÇINAR, for her support and patience during my academic studies.

# TABLE OF CONTENTS

# LIST OF TABLES

TABLES

# LIST OF FIGURES

FIGURES

# LIST OF ALGORITHMS

ALGORITHMS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AV | Oracle Audit Vault Server |
| AWR | Automatic Workload Repository |
| DAM | Database Activity Monitoring |
| DBA | Database Administrators |
| DBF | Oracle Database Firewall |
| IPS | Intrusion Prevention Systems |
| ORM | Object Relational Mapping |
| OEL | Oracle Enterprise Linux |
| RAC | Real Application Cluster |
| RDBMS | Relational Database Management System |
| SIEM | Security Information and Event Management |
| VPN | Virtual Private Networks |

# CHAPTER 1

# INTRODUCTION

In our daily lives, almost everything we use on electronic environments uses information data in databases. This data stored in databases is sometimes utterly important for business interest of organizations and institutions and thought as corporate assets. Therefore, in order to protect the vital information and ensure privacy and access of the data, and in order not to allow unauthorized people and applications to reveal data for any reason, organizations must provide some security solutions [23]. Nowadays, these big information systems are under the attack of hackers. Some of them want to prove how capable of accessing very secret information. Some of the hackers want to steal information and then sell this information to unauthorized people. According to Lane [14], not just simple organizations are under the attack of these hackers, militaries, governments, institutions, firms providing services, everywhere which stores information are under risks of attack.

As Cloud Computing has evolved and developed increasingly, big information systems and many IT cooperation firms choose cloud computing for their technology solution because of growing IT costs, huge software license cost, manageability, and safety server rooms [32]. Cloud computing helps organizations to solve the related problems by reducing costs, better performance and fast IT service responses. However, many organizations concern on the security of cloud computing systems because of storing important information on someone else storage. Securing information data in cloud systems can be an important issue because of cloud systems is a collection of many system and cannot be managed by specific data and software owners. We thought to develop a private database cloud system instead of classic cloud systems. There are many security solutions for private database clouds.

While databases are under these types of attacks, some precautions must be taken in order

to provide database security. For database security, monitoring all activities and operations, some kind of security controls must be implemented to ensure institutions information security and business objectives [1]. Vanhorn [28] states that embedding a monitory and auditing system on databases will make it easy to take and implement security policies on organizations. All activities on the system and all user access attempts to the system will be monitored and classified according to system security rule sets. Users and end- users will be granted differently between each other in order to avoid confusion. All these definitions must be controlled from the same point by the same people. Security must be separated from other areas in order to trustworthy in all departments. In parallel to Vanhorn [28], Burtescu [5] points out that defining security policies and rule sets must be controlled from a central point. These should be include control elements, administrative control agreements which help to make sure of security of databases. In the same way, Montesino [17] points out that a security control can be automated if the operation of the control requires only machine-readable and processable resources (for example, controls such as awareness and security training cannot be automated because they require the training of humans). The centralized management of security tools is essential for automation in that it allows the control of the entire system from a single point.

Security Information, Event Management, and Database Activity Monitoring are two types of existing solutions which provide security in private database cloud systems which are Security Information and Event Management (SIEM) and Database Activity Monitoring (DAM). SIEM detects anomalies in network systems by gathering and analyzing log and audit in information systems. SIEM is a box solution which is pre-installed server and placed in network system by using a duplicate switch port. It listens devices like firewalls, application and data servers, management systems, on the network by using this port policy [30]. In addition to the Wenge, Bedwell [2] explained SIEM as, it is a real time monitoring and analyzing all activities on the management system. It is known as storing information logs and detecting intrusions by analyzing these logs. There is a management console which makes you define rules which separate anomaly access attempts. According to Nicolett, Litan, and Proctor [19], SIEM is a very successful tool which gathers information from managements servers and compliance tools in cloud systems, database and application server logs. After undergoing these logs, SIEM managers filter and match users for privileges on the system in order for real-time intrusion detections.

DAM is a security solution only for databases in a network system. It is running by installing software agents to the database servers or direct running on database itself [18]. DAM solution provides monitoring and managing all database activities in database management systems. This product actually works as SIEM product by listening database servers and it captures all defined terms in log files and listeners. Additionally, some of them protect database from unauthorized users and activities by alerting and blocking anomalies. According to Gartner [26], DAM is a technology which also ensures database security by analyzing all database activities like reading and updating data from the application environment. DAM ensure security by defining policies. After gathering events from databases, DAM manager monitors log and define policies like access controls on database management systems. It defines privileges for users on specific objects. The system alert when a policy violation is detected. According to definition on the management console, the system can mail violation, automatic prevent from violation, or just lists and reports violations on the system [14].

In our solution, we use classical Access Control Mechanisms which every Relational Database Management System (RDBMS) includes itself. These access control mechanisms enable us to define policies for specific users and applications by using audit and activity logs from the database. By defining privileges for database objects and actions, users are enforced to obey security policies. We developed a software agent which runs at database level on Oracle Database. However our solution stores activity logs on database itself, because of running on database level. As a result of this, using only this mechanism is not enough to secure database. Chen and Wesley [7] think similar, such techniques are insufficient because not only defining policies according to data logs and auditing from the database is important but also securing these logs and auditing from malicious users which may get rid of a series of information. We use a private database cloud and a central management point in order for transferring these logs and managing database system from remote locations. This also enables us to secure multiple databases from the same location.

Existing solutions use running software agents at database servers. That is, these agents gather logs from database by using listener and network logs. Then, they transmit these logs to DAM's or SIEM's Administrator Server. This causes nearly 3% overhead on database server CPU and RAM. As a result of its working principle, it has defined database terms and according to these terms, they gather matching lines from listener and network logs. Therefore, it misses some logs from database server because of high-performance processes

[28]. Performance is the most important issue for us and because of our solutions are being run on database level, we exclude high load queries from solution and find work around for them. Securing logs on both database server and on a distributed server is a common problem [9]. We encrypted logs which will be transmitted to the management server with timestamp. Additionally we put a firewall between database server and management server. Using this firewall, it will help us to define rule sets to check applications and users which want to access servers. Only allowed users and applications defined by us will access to the server. By using both encryption with timestamp and a firewall, we can secure logs on both database server and management server.

The scope of this study is to develop a database activity monitoring and preventing systems which enable to work with many databases at the same time. By using encryption approach, it is considered to be effective to use, secure from internal and external attacks. In following sections, Chapter 2 presents background and related work of the area. Chapter 3 presents our solution deeply as a concept and introduce how it is worked. Chapter 4 lists evaluation and experiment result of our study. It compares our solution with existing solution according to performance and security. Finally, Chapter 5 shows conclusion of our study.

# CHAPTER 2

# BACKGROUND

In cloud systems, some precautions are taken to prevent external attacks [32]. There are Intrusion Prevention Systems (IPS), Firewalls, Antiviruses, Malware Protection solutions, and etc. for network security. However, none of them is actually enough for an exact database security solution [13]. It is clear that we take precautions for external attack in cloud systems by using these security solutions. While RDBMS are considered parts of the cloud systems, security solution must meet all applications in the cloud system [31]. From the perspective of cloud security, there are four kinds of security:

- Security of Application: It ensures that all applications in cloud system must be protected against all various attacks. Application Servers, Database servers, applications which use and manage information data must be fenced.

- Security of Physical Environment: Cloud system provider must provide the security of all servers and devices. Server rooms must be designed carefully according to fire, flood, and etc.

- Security of Network: Security solutions like firewalls, IPS, Virtual Private Networks (VPN) must be implemented in network. These solutions will protect system from external attacks.

- Security of Host: All applications are running on servers which are called hosts. Like every personal computers, these hosts run operating systems on themselves. To secure these operating systems, anti-virus programs, and malware preventive solutions, auditing and logging of all operations must be provided [23].

Likewise Chen and Chu [7] also highlighted the importance of databases in network environments. They said that, continuously, databases are under attack of malicious people who want to get information and data leakage. Therefore, when these users gain access to database systems, they cause large damages to organizations. While there is such a huge dangerous, it is not easy to protect database systems. It is argued by people, when these attacks will be ended. According to Anderson's Rule, if it is wanted to construct a database system and want to serve it to public, it is nearly impossible 100% secure because of this big information system are open to external applications. It must be closed all opened rear doors in databases and networks [17].

From time to time, organization are not aware of being under attacks or their data is accessed and stolen by malicious users. In 2012, Verizon conducted a study on Data Breach Investigations with cooperation from Australian Federal Police Department. As a result of this study, 98% of information was stolen from information systems and while 92% of information was stolen, organization had not recognized being under attack and losing data. According to key findings from the 2013 United States of Cybercrime Survey, 80% of database attacks is internal attacks [18]. Some malicious employees use their privileges to access and attack their databases. Some of them are infiltrators who work with outside for information sharing with black market [6]. Imperva, one of the Database Security Solutions firms in market, stated that most attacks preformed from internal users and these users are not hackers [27]. Also emphasized that, most of these attacks can be prevented by using sample access controls on databases.

While databases are under these types of attacks, some precautions must be taken in order to provide database security. According to the 2009 Verizon Business [18] , Data Breach Investigations Report, 97% of database attacks can be easily prevented by using some access controls. Most of the RDBMS have basic Access Control Mechanisms in themselves. A security mechanism provides users to adapt a security policy which enables them to access specific objects and do defined actions. These security mechanisms are called Access Controls. In classical RDBMS, Database Administrators (DBA) provides access control for every users before allowing them to connect database. DBAs are the main responsible persons in databases. They are responsible from access controls and users audit logs, in that, if some users happen to connect databases, they can do everything according to their privileges.

**EMPLOYEE**

| Name | Ssn | Bdate | Address | Sex | Salary | Dno |
|---|---|---|---|---|---|---|

**DEPARTMENT**

| Dnumber | Dname | Mgr_ssn |
|---|---|---|

Figure 2.1: Sample Table Definition

For access control testing, use sample Employee and Department tables. Figure 2.1 shows the columns of sample tables. In this example, it is not important how many columns there are. Access controls can also be used for specific columns on tables. Suppose that the DBA creates four accounts which are:

**Example 2.0.1** *Database Users*

- User 1, User 2, User 3, User 4

In Example 2.0.1, there are four users, User 1, User 2, User 3, and User 4. Object owner or creator has all access right on its objects in each RDBMS. After DBA create accounts, User1 creates the two tables which are:

**Example 2.0.2** *Database Tables*

- EMPLOYEE and DEPARTMENT.

User 1 has all access on Employee and Department tables, so it can grant access controls to another user. User 1 wants User 2 to add, and delete data from these tables and run below query:

**Example 2.0.3** *Grant Access to User 2*

- GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO User2;

User 2 has gained Insert and Delete accesses on these tables; however, it cannot run Select query on these tables because User 1 did not give any access on Select statements. Then, User 1 give select privilege on these tables to User 3 and wants User 3 to can give privilege to whom he want to give.

**Example 2.0.4** *Grant Access to User 3*

- GRANT SELECT ON EMPLOYEE, DEPARTMENT TO User3 WITH GRANT OPTION;

In Example 2.0.4, User 3 gained Select access and grant Select access to other users. Now User 3 can give select privilege for these tables whoever he wants:

**Example 2.0.5** *Grant Access to User 4*

- GRANT SELECT ON EMPLOYEE TO User4;

After User3 gained privilege on Select query, it grant this privilege to User4. However, if User1 wants to take privilege back from User3, all privileges User3 gave will also be taken:

**Example 2.0.6** *Revoke Select Access from User3*

- REVOKE SELECT ON EMPLOYEE FROM User3;

At the end, User1 revoke privileges from User3 and, because of that select privilege is taken from User4, too.

In each RDBMS, DBAs are responsible for these access controls. In addition they have to control users audit because of deciding on access control of which needs privilege on which object. These basic access controls help DBAs to encounter internal attacks; however, to decide which user must access which actions and objects, for a period of time, all action logs and audits must be monitored. For database security, monitoring all activities and operations, some kind of security controls must be implemented to ensure institutions information security and business objectives [1].

Auditing has become an important tool over the past 10 years for both compliance and forensic analysis of data breaches. Audit records provide an irrefutable record of actions taken depending on whether they are generated by a database, directory, or operating system. Information such as the event type (create table, drop table, create procedure, truncate table, select, insert, update, delete) coupled with the context of the event such as the initiating IP address, event time, and actual SQL statement, are just a few examples of audit information that is commonly needed in compliance and forensic reports.

Vanhorn [28] indicates that embedding a monitory and auditing system on databases will make it easy to take and implement security policies on organizations. All activities on the system and all user access attempts to the system will be monitored and classified according to system security rule sets. Users and end-users will be granted differently from each other in order to avoid confusion. All these definition must be controlled from same point from same people. Security must be separated from other areas in order for maintaining trustworthiness in all departments.

Burtescu [5] supports Vanhorn's ideas on defining security policies and rule sets must be controlled from a central point because of logs security. These should include control elements, administrative control agreements which help to make sure of security of databases. In the same way, Montesino and Baluja [17] speculate that a security control can be automated if the operation of the control requires only machine-readable and processable resources. To illustrate, controls such as awareness and security training cannot be automated because they require the training of humans. On the other hand, each database attack preventing system has some cost on not only servers, but also on cloud systems. These preventing systems are also using RDBMS resources. RDBMS needs to calibrate its resources in order to be aware of cloud environment and produce an accurate estimated cost. All monitoring tools allocate some resources (RAM, CPU, storage) on database servers. What is more all of them generate a huge amount of log data to analyze for security? There are many databases and at the end of calculation, a lot of resources will be taken for database security and a huge amount of storage of each RDBMS will be left for these systems. The centralized management of security tools is essential for automation because it allows the control of the entire system from a single point.

There are some distributed systems in cloud systems to manage all separate databases in cloud from a central point. A centralized RDBMS gathers logs from all databases. The system works as collecting data from multiple physical locations. In short, a distributed database is a logically interrelated collection of shared data, and a description of the data is physically distributed over a computer network. However, both having many databases and having huge amount of data is a problem for analyzing and managing security systems [29]. There are two existing solutions according to monitoring and preventing database security in cloud systems, Security Information and Event Management (SIEM) and Database Activity Monitoring (DAM).

## 2.1 Security Information and Event Management (SIEM)

SIEM is a box solution, which is placed in network system and collect logs from all network devices include firewalls, application servers, database servers. SIEM is combined from Security Information Management (SIM) and Security Event Management (SEM), which is a solution for security of entire network system by providing real-time analysis and alerts from network devices. These logs are transferring to a management system which monitor and analyze all logs. When logs are gathered, SIEM undergoes analyzing these logs for filtering and matching for defined rule sets. In cloud systems, these SIEM solutions are very popular because of securing network completely from external attacks [2].

Stephenson [26] described the main function of SIEM solutions is like an intrusion detection and prevention system. It gathers logs to a central point, analyzes logs according to some rule sets defined by security administrator, then detects anomalies in the information systems. It is a compliance tool for operating systems, application and database servers, and other network devices. Security administrator gives privileges to users and applications on services by providing log auditing and reviewing incidents before it has happened [29]. SIEM can be a good solution for network security, however, it cannot protect database from internal and external attacks exactly. Its working principle is consisted from listening network and host listener logs by selecting defined database terms. It does not provide 100% security of a system. To ensure the security of system, you must define all possibilities which can be occur on the system [17]. Also by running as an agent on the operating system, it cost some RAM and CPU on the host which runs database system. On the databases which process high

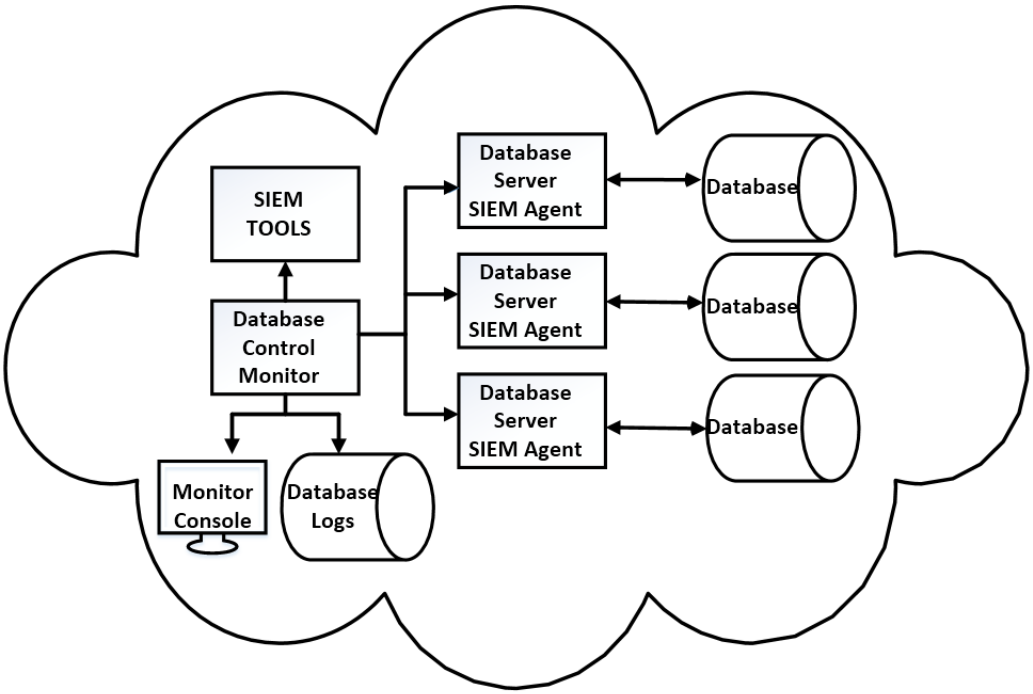



Figure 2.2: SIEM Solution in Private Database Cloud

transactions, it will cause a huge performance problem.

## 2.2 Database Activity Monitoring (DAM)

The best way to secure database is monitoring logs and analyzing these logs to determine a good solution for database and its users [18]. DAM is a security solution like SIEM for databases which can be run independently from database server or run at directly database server. The main working principle of this solution is monitoring and analyzing all activities which occur on databases, and manage these logs from a central administration. DAM makes it enable real-time monitoring of database activities, also by defining permission rules for users, it can protect database from unauthorized people [8]. DAM is an important solution for protecting information from external attacks. According to the Gartner [29], DAM is a very well-formed access control mechanism which gives privileges to users and applications to access database and monitor all native actions and functions. It provides separation of duties with authorized users on the database, which enables to monitor all administrator activities

easily.

There can be defined rule sets and policies on database by using DAM solution. This enables to monitor real-time activities and detects unusual database operations from application layer. Apart from central administration, DAM aggregates and correlates database logs without enabling native database audits and logs which saves Database from extra loads. Enabling native database logs and audits decrease database performance because of intensive level auditing of resources [14]. The main goal of DAM is to separate normal actions and prevent attacks from both inside and out.

There are two running types of DAM solutions. First one is like SIEM tools, has an agent which run at database server host. This agent collects logs from network logs and database listeners [25]. This DAM solutions have disadvantages on host performance because of using host CPU and RAM. However, these agents get monitoring operations and database action logs without interfering with database business process. Therefore, because of running separately from database, it takes some time to take action on when a policy violation is detected. When DAM is running on protection mode, it traces all activities and according to policies, it protects database from unauthorized actions which will be analyzed by central administration, and then applied on the database system [32]. Second one is like database packages and procedures which runs at database level. While it is running on database server, it can directly trace all logs and take action when process has just started. It uses database itself functions and triggers for logging and analyzing data. A disadvantage of these solutions is that, it can have performance problem on database server because of controlling all actions. This solution ensures advanced security with access control mechanisms.

Cobb [8] argued that, DAM solutions are much better on database security then SIEM solution. When we compare these solutions, SIEM focuses on entire network system and tries to secure all attacks which can be occurred from outside. It has disadvantages on internal attacks. On the other hand, DAM solutions focus on only real-time database monitoring and prevent attacks according to policy violation. When all is taken into consideration, network systems are protected from external attacks by security devices like firewalls, IPS every time. Main problem is to take precautions for attacks coming from inside users and applications. DAM's are being run at database level also make it more reliable for internal attacks. According to Forrester research on database security [9], most of the companies do not take detailed

precautions for database security. Also, 97% of database attacks are coming from internal users. In our system, we choose DAM solution because we thought that preventing internal attacks is the main issues on database security.

We research Database Auditing and Protection products on the market because of our solution is a kind of DAM solution. There are many DAM products some of them have huge prices and some of them are free. Forrester, an expert research company, researched on DAM products according to their Performance, Stability, Current Offering, Strategy, and Market Presence in 2011. Forrester's vendor selection criteria was:

- The product must be a comprehensive enterprise-class database auditing solution which can help to meet regulatory compliance requirements and protect data against theft, real-time data protection, and storage of audit trails.
- The product must have a customer base of 100 or more enterprise customers.
- The product must be readily available until August 15, 2010.

There were 147 database auditing and real-time protection vendors and there were 3 categories: Current Offering, Strategy, and Market Presence. Current offering means, products properties which they serve as securing database like audit policies, real-time protection, and manageability. Strategy means, how they are selling their products on the market, customers' demands and how they respond to those demands. Market presence is their services after selling their products. As you see in Figure 2.1, IBM, Imperva and Sentrigo are the leaders according to current offering. They have sold much more products according to other vendors. In Figure 2.3, ranking is IBM, Imperva and Sentrigo. Also, these vendors strong user activity auditing, policy management, real-time protection, and application support capabilities as well as their forward-thinking strategies are much more successful compared to other vendors, products. IBM stands at the first place on Current Offering, Strategy and Market Place.

In Table 2.1, all scores are based on a scale of 0(weak) to 5(strong). IBM has 4.67 points from Forrester and it is leader of market on current offering [9]. Under current offering topic, Forrester analyzed products on database auditing, user and application auditing, audit policies, auditing repository, reporting and analytics, real-time protection, architecture, and manageability. In the end, IBM has 4.67 points of average current offering.

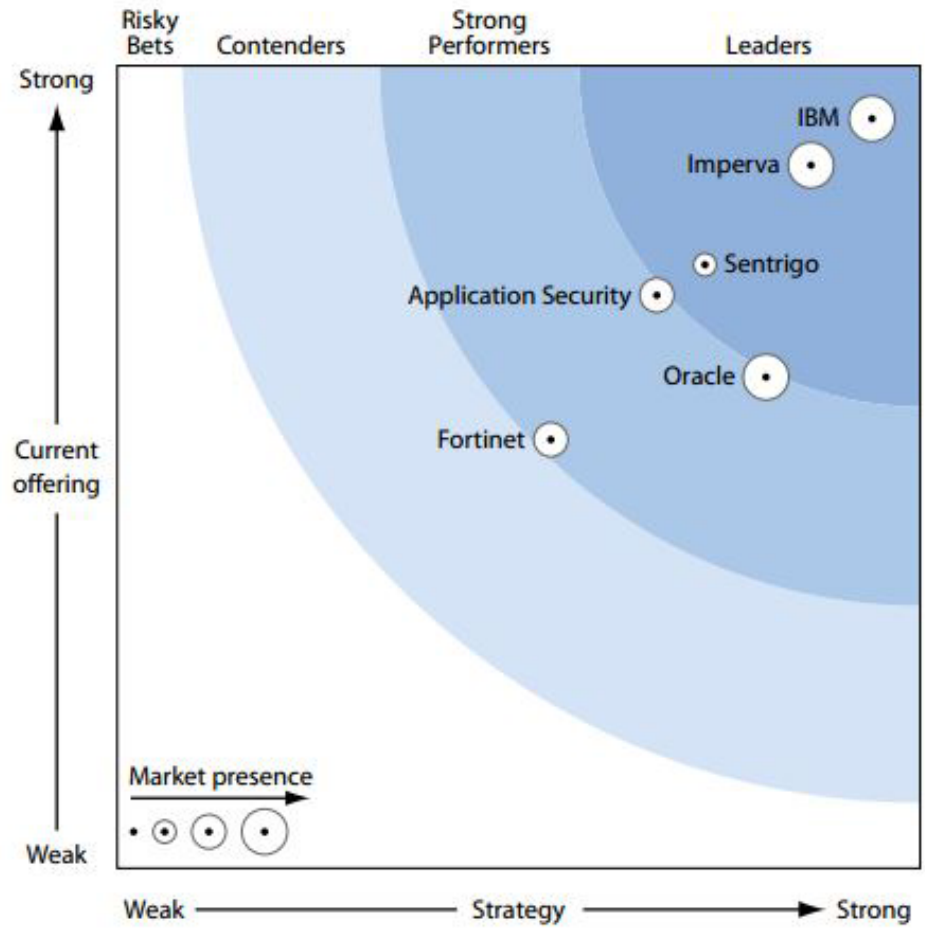



Figure 2.3: DAM Vendors on the Market

Table2.1: Forrester Research on DAM Solutions According to Offering

| | Forrester's Weightining | Application Security | Fortinet | IBM | Imperva | Oracle | Sentrigo |
|---|---|---|---|---|---|---|---|
| **CURRENT OFFERING** | **50%** | **3.57** | **2.67** | **4.67** | **4.38** | **3.06** | **3.76** |
| Database Auditing | 10% | 3.82 | 3.52 | 4.88 | 4.40 | 3.92 | 4.12 |
| User and App Auditing | 15% | 3.08 | 2.56 | 4.68 | 4.44 | 2.68 | 3.56 |
| Audit Policies | 10% | 3.90 | 3.30 | 5.00 | 4.60 | 3.20 | 4.60 |
| Auditing Repository | 10% | 3.80 | 3.24 | 5.00 | 4.04 | 4.52 | 4.28 |
| Reporting and Analytics | 10% | 4.46 | 3.44 | 4.76 | 4.64 | 3.40 | 3.56 |
| Real-time Protection | 15% | 2.70 | 1.10 | 4.80 | 4.80 | 2.20 | 4.20 |
| Architecture | 15% | 3.57 | 3.00 | 4.19 | 4.15 | 3.17 | 2.94 |
| Manageability | 15% | 3.80 | 2.15 | 4.40 | 4.04 | 2.29 | 3.35 |

Table2.2: Forrester Research on DAM Solutions According to Strategy

| | Forrester's Weightining | Application Security | Fortinet | IBM | Imperva | Oracle | Sentrigo |
|---|---|---|---|---|---|---|---|
| **STRATEGY** | **50%** | **3.36** | **2.70** | **4.70** | **4.32** | **4.04** | **3.66** |
| Product Strategy | 60% | 2.80 | 2.50 | 4.50 | 4.40 | 3.40 | 3.30 |
| Compare Strategy | 40% | 4.20 | 3.00 | 5.00 | 4.20 | 5.00 | 4.20 |
| Cost | 0% | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Likewise, in Table 2.2, Forrester has analyzed DAM products according to strategy topic. Forrester points out product strategy, corporate strategy, and cost of products. Again, IBM has top points and became the leader according to Strategy topic.

Table2.3: Forrester Research on DAM Solutions According to Market Presence

| | Forrester's Weightining | Application Security | Fortinet | IBM | Imperva | Oracle | Sentrigo |
|---|---|---|---|---|---|---|---|
| **MARKET PRESENCE** | **0%** | **3.69** | **3.49** | **4.92** | **4.18** | **4.88** | **2.64** |
| Installed Base | 20% | 3.00 | 2.00 | 5.00 | 5.00 | 5.00 | 2.00 |
| Revenue | 10% | 2.60 | 4.00 | 4.20 | 2.40 | 3.80 | 2.60 |
| Services | 20% | 3.60 | 5.00 | 5.00 | 2.90 | 5.00 | 2.20 |
| Employees | 20% | 4.05 | 4.25 | 5.00 | 4.30 | 5.00 | 2.40 |
| Technology Partners | 20% | 5.00 | 1.70 | 5.00 | 5.00 | 5.00 | 3.68 |
| International Presence | 10% | 3.00 | 5.00 | 5.00 | 5.00 | 5.00 | 3.20 |

The other topic is Market Presence, in Table 2.3, Forrester has analyzed DAM products on according to their Installed Base, Revenue, Services, Employees, Technology Partners, and International Presence. Again, IBM has top points, 4.92 and became the leader according to Market Presence topic.

In the same way, TechTarget, computing science research firm, studied on DAM solutions in 2008 and indicated two database activating products which are IBM Guardium and Imperva's SecureSphere products. On the report, they tested these products on the same hosts (IBM AIX 5.3, Linux 2.6) and on the same database servers with same versions (IBM DB2, Oracle 10g, and SQL Server 2005). With the help of this study, they pointed out Guardium and SecureSphere is a well-designed and developed products for large organizations. Investigating these leading top products on the market helps us to develop our system.

### 2.2.1 IBM Guardium

Messmer [16] concluded that, IBM's Guardium DAM solution ensures the security of the trusted information in your database systems, and also by running on different heterogeneous environments, it reduces costs by automating the entire compliance auditing process. Prince [21] agrees with Messmer in that IBM's database auditing and protecting product has high performance and scalability, easy to manage, and enabling real-time database protection. In private database cloud systems, Guardium can be deployed centralized and monitor for real-time database security, database actions auditing, take reports according to regulations, strict security on data-level access control, database vulnerability management and auto-discovery of sensitive data. Today, IBM InfoSphere Guardium has been deployed across many large enterprises.

### 2.2.2 IMPERVA SecureSphere

Imperva is one of the most popular DAM solutions in the market with a strong feasible database auditing solutions. It has strong sides on real time monitoring and protecting, database transactions and functions, high-level privilege, and policy administrating. According to Eddy [10], Imperva will increase in market in following years as corporate strategy and investment on the market presence. Imperva's database auditing solution is one of the best solutions according to other vendors. As Lemos [15] argued that, Imperva has defines rule sets very well and policies via over 350 security test on different types of databases and default user accounts in those databases. It is easy to detect fraud-related users and actions by using Imperva's platform. There can be many databases from different locations, or from a central point, Imperva can handle all of them.

### 2.2.3 MCAFEE Sentrigo

Prince [22] listed Sentrigo's strong points like its auditing policies, easy usability of product, real-time monitoring and protecting, and reports according to regulations. According to Forrester, although Sentrigo does not have as many customers as IBM or Imperva, it has very large Fortune 1000 companies that are using its product to support hundreds of critical databases. Rouse [25] maintained in another article that, Sentrigo automatically finds

databases on your network, protects them with a set of pre-configured policies, and helps you define a new custom security rule sets for your database systems and making it easier to improve critical asset data protection in virtualized or cloud computing environments.

### 2.2.4 ORACLE Audit Vault Server and Database Firewall

Oracle is not a strong leader in listed areas according to Forrester Research; however, Oracle is one of the biggest and most used database in the big data and transaction world. A generally accepted thought on Oracle Databases is that, Oracle's database activity monitoring solutions are performing in much more stable and protective ways. According to HighBeam Research (2007), Oracle offers a comprehensive database security solution which is far beyond of other auditing solutions by using Oracle Audit Vault Server (AV) and Oracle Database Firewall (DBF). AV and DBF together form a database activity and monitoring solution and provides full security for databases and logs audit data from databases, operating systems, and Oracle LDAP directories. Also Prince [20] argued that, on administrator panel, security experts can monitor a highly accurate SQL grammar and block policy violation SQL queries before it reaches the database. Oracle provides detailed audit information for reporting of easy compliance and alerting by combining with information from the network. According to the Forrester report, with AV and DBF, monitoring controls privately made to meet enterprise security requirements. Therefore, Oracle has an extra product for protecting information from even the most authorative users i.e., database administrators, by running at the database level by using the Oracle Database Vault product. Oracle Database Vault provides separation of duties on databases which helps DBAs to adapt regulations easily. According to Oracle Database Vault Data Sheet by Oracle Inc., Database Vault products use multifactor policies that are enforced in the database for high security and performance and enable the reduction of attack surface without disrupting business activity.

### 2.2.5 INFOFENCE

Infofence is a domestic Oracle Database control and security software. It is a kind of database activity which runs only at Oracle databases and protects solutions. Infofence was produced by Helezon Computer Consulting in Ankara with TUBİTAK 1507 Small and Medium Sized Enterprise research and development support. It is tested on Oracle 10g, 11g, and 12c and

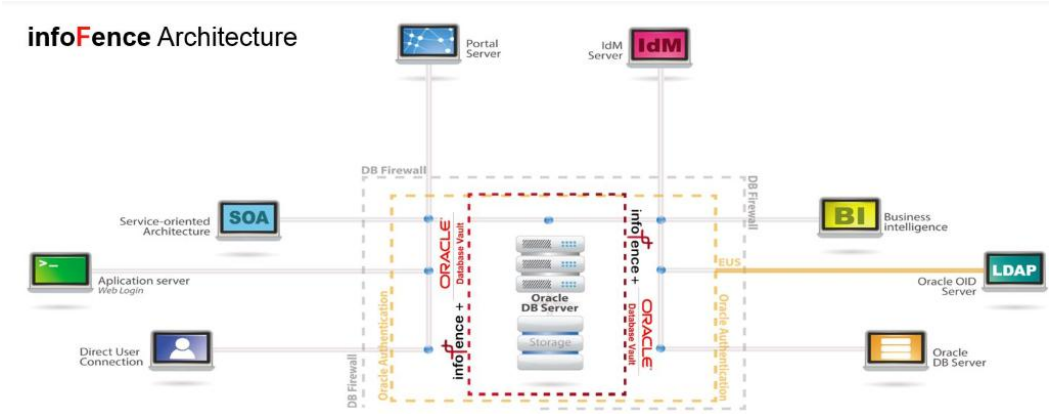



Figure 2.4: InfoFence Architecture

runs at database level like Oracle Database Vault using Oracle Database Triggers. In addition, Infofence can be integrated with Oracle Database Vault very well by managing its policies and rule sets.

Like all other solutions, Infofence has two running modes. First one is sniffer mode which makes product only monitor all actions on the database. Other mode is the protecting mode. When protecting mode is enabled, it prevents all actions from unauthorized users. Lane [14] maintained, many firms are afraid of to enable security product active mode on. There are many defined conditions on security products which provides normal and anomaly access and activities on databases. After enabling active mode, users who are outside of these normal activities and accesses are not be able to do any operations on databases. However, how DBAs may be sure that an activity is normal or abnormal. Therefore, it is not easy to activate protecting mode because of not being sure. After activating protecting mode, all off the operations on databases can be stopped, important jobs can be disabled, important data can be manipulated. These events discourage DBAs to take actions. Although it is not easy to use protecting mode, Infofence's Sniffer mode makes it easy to list actions normal and abnormal. Infofence admin enables sniffer mode. Every user continues on doing their jobs on databases. Nobody is going to be blocked. However, as seen in Figure 2.4, in Infofence control panel, all logins which are not defined in Infofence will be seen as BLOCKED in red color.

Infofence is a domestic product which has been tested by several Turkish Banks and Hospitals (Günçer, 2014). In 2013, in Ministry of Finance, Infofence is tested by 10000 (ten thousand)

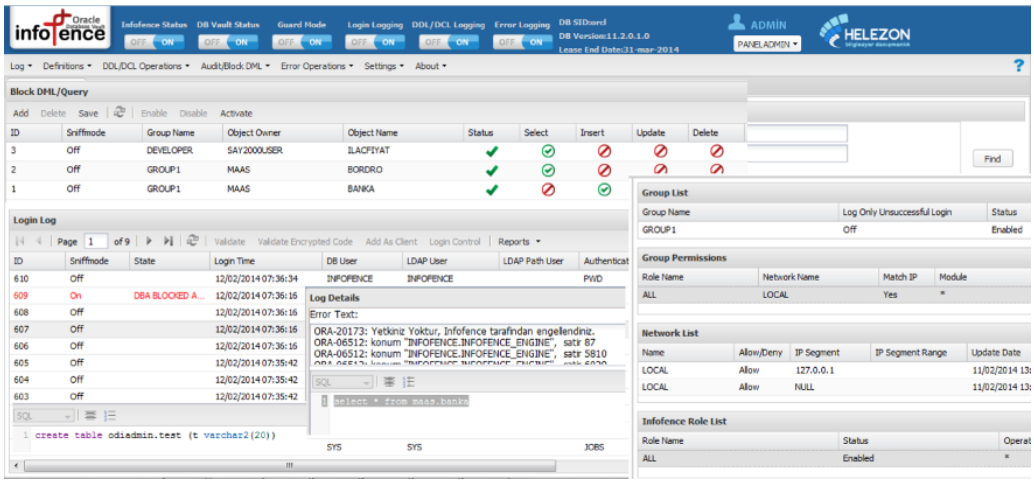



Figure 2.5: InfoFence Control Panel

simultaneous users on a running ORACLE Real Application Cluster (RAC) production system. While Infofence was running on production system, all queries were investigated carefully, and there were no waits on systems resource (disk write, disk read, CPU, and RAM). Infofence stores its database logs and audits in its tablespace and schema objects. Storing all audit and database log on the same server with database is not secure enough. Also it must be controlled these log cannot be changed by anyone and anything [3]. As a result of this, log should be encrypted and stored on a different server.

# CHAPTER 3

# DATABASE SECURITY ON PRIVATE DATABASE CLOUDS

First, a private cloud of databases has been created in our security solution. There can be many databases for monitoring and logging. If we use SIEM based solutions, we cannot log all actions happened in databases or sometimes monitoring and analyses data for security takes some time to respond which will cause our solution not work properly. Therefore, our solution is based on DAM solutions which run directly at database level. It monitors and logs all database activities. Afterwards according to defined policies, security mechanism decides whether actions are normal or anomaly. It is a well-developed infrastructure of security system. It can be shaped by security administrators, hands.

A RDBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security portions of a database against unauthorized access. To hinder these problems, monitoring and preventing systems can be developed and managed from a central point. Using this system can detect violations and keep track of each user on the system. Likewise this system can keep all user activities, what they should do or do not. This inference detection approach should be based on a security admin on the system. In short, our solution creates a new security layer on database by not changing the defined authorizations and it will managed from a central point in a private cloud.

While testing DAM solutions on the market, we want to use Infofence infrastructure for our system. However, from our security perspective, it has some problems related to security and ensuring the safety of database logs. Therefore Infofence stores all logs and information about database, in that database itself. So we wanted to improve Infofence Database Activity and protection product by developing a private database cloud which is administrated from a central point. Infofence is an installable software which run at database level. However,

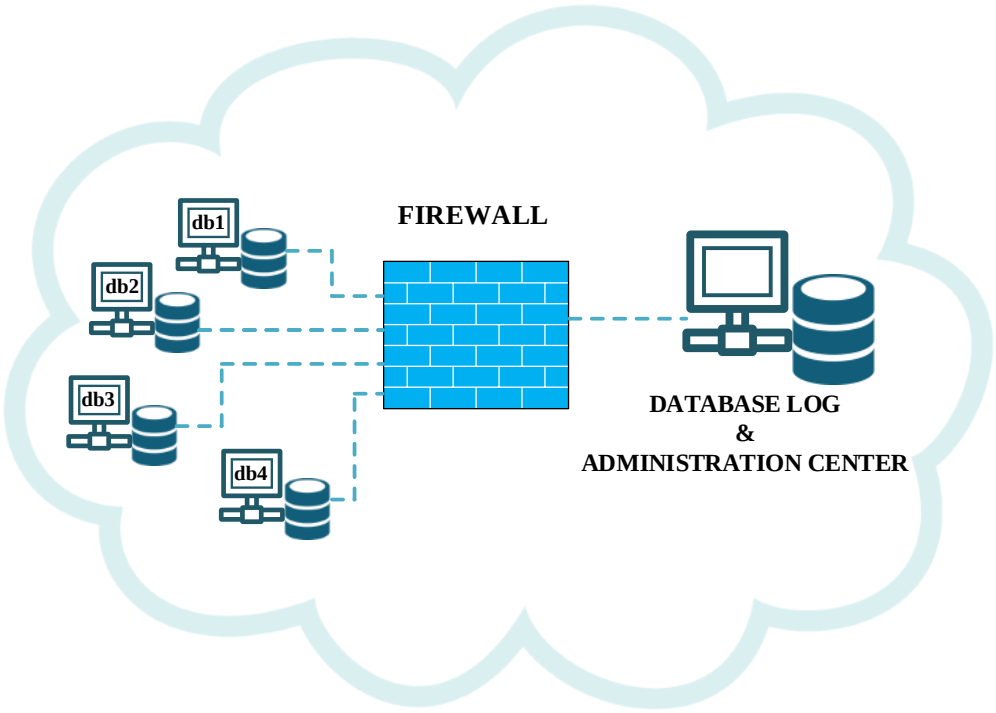



Figure 3.1: Database Security on Private Database Clouds

in private clouds, there are many different databases in same organizations. Sometimes, all users, all security conditions are same with other databases. Existing Infofence software is designed for only one database. When there are many databases, it must be installed and configured differently for every databases.

Our security system consists of two parts. First part ensures security for only one database. Second part manages many database securities from a central administration. By using this two parts, we developed a security system for databases on a private database clouds. There are many kinds of database managements systems like Oracle, IBM DB2, SQL Server, MySQL, and etc. What is more, each of them has many different versions of their own. Parallelly, there are many types of operating systems like UNIX, Linux, and Windows. Developing a general database activity monitoring system to cover all of these systems and is not easy to handle for limited resources and supports like time, money, man power. We decided to develop a pilot DAM solution for Oracle Database with 11G version on Linux 6.4 operating systems. What is more, due to fact that Oracle is having differences between its versions,

this system is tested only on Oracle 11G, which can have problems with previous and next versions.

## 3.1 Performance

Whether your database is secure or not, the most important characteristics of a RDBMS and database solution is performance [3]. Database activity monitoring and protecting tools collect and monitor activity log in RDBMS by auditing or some database features which gather activity data output. This causes some overhead on the system which is used for monitoring the system. In DAM running database systems, logging and auditing causes approximately 3% load of system performance [11]. According to platform choice and organization security level, some of the overhead is acceptable. However, generally database administrators do not choose any tools or product which cause them extra load on production systems. Fiorilla [12] claimed that, when they test Oracle auditing and monitoring tools on Oracle 10g, the tools load on database system is very huge for not being acceptable for system performance. This performance issue is the main reason for organization to choose a database activity monitoring and protecting products.

Table3.1: Oracle Automatic Workload Repository (AWR)

| CPU Time (s) | Executions | CPU per Exec (s) | %Total | Elapsed Time (s) | %CPU | SQL Text |
|---|---|---|---|---|---|---|
| 2,669.91 | 3,995 | 0.67 | 22.28 | 2,672.68 | 99.9 | SELECT * FROM INNOVAPRODUCTION... |
| 1,475.30 | 285,856 | 0.01 | 12.31 | 2,291.61 | 64.38 | SELECT COUNT(*) FROM DBA_SCHED... |
| 1,016.93 | 0 | | 8.49 | 1,842.48 | 55.19 | DECLARE threads pls_integer :=... |
| 957.22 | 1,511 | 0.63 | 7.99 | 960.17 | 99.69 | select entegrasyo0_."Id" as co... |
| 476.85 | 363 | 1.31 | 3.98 | 477.8 | 99.8 | select aktorveri0_."Id" as col... |
| 345.09 | 556 | 0.62 | 2.88 | 345.76 | 99.81 | select aktorveri0_."AktorId" a... |
| 231.9 | 140 | 1.66 | 1.94 | 232.47 | 99.76 | select taslakbelg0_."Id" as co... |
| 203.77 | 38 | 5.36 | 1.7 | 204.46 | 99.66 | select col_0_0_, col_1_0_, col... |
| 155.95 | 19,205 | 0.01 | 1.3 | 259.69 | 60.05 | begin infofence_engine.db_logi... |
| 132.56 | 9 | 14.73 | 1.11 | 132.87 | 99.77 | SELECT COUNT (B. "Id") AS "Cou... |
| 126.79 | 9 | 14.09 | 1.06 | 127.19 | 99.69 | SELECT * FROM (SELECT TABLO.BE... |

In Infofence Datasheet and catalog, Haluk Günçer, the manager of Helezon Computer and Consulting, states that when they developing Infofence product on Oracle Systems, they paid more attention to performance more than other attributes. He futherly said that, when some properties of Infofence cause overhead on the systems, they find a work around for security

property or completely remove that properties. Performance is the main characteristic in Infofence.

In Table 3.1, there is an example of SQL list according to CPU cost time, generated by Oracle AWR tool. There are also, SQL list according to elapsed time and User I/O wait time. AWR reports help Oracle DBAs to tune high cost SQL statements and perform operations on database objects.

## 3.2 Encryption with Timestamp and Firewall

BIMOL, SINGH, and DEVI [3] explained the importance of logs not being changed by anyone. In classic databases, like DBAs , and SYS user, the most authorative user in database, can change every object and data in databases. We must be sure our logs and audit file cannot be changed by anyone. Timestamp is used for proving documents, logs, contracts, and electronic documents are existed before a certain time and not changed in that period of time. For example, timestamp can be used to prove that a log file is saved originally in storage and not changed in a period of time. In 30 November 2007, Turkish Government enacted to store Internet documents and soft copies with Timestamp law. In this scope, even a domain host services which stores web pages, must use Timestamp for auditing.

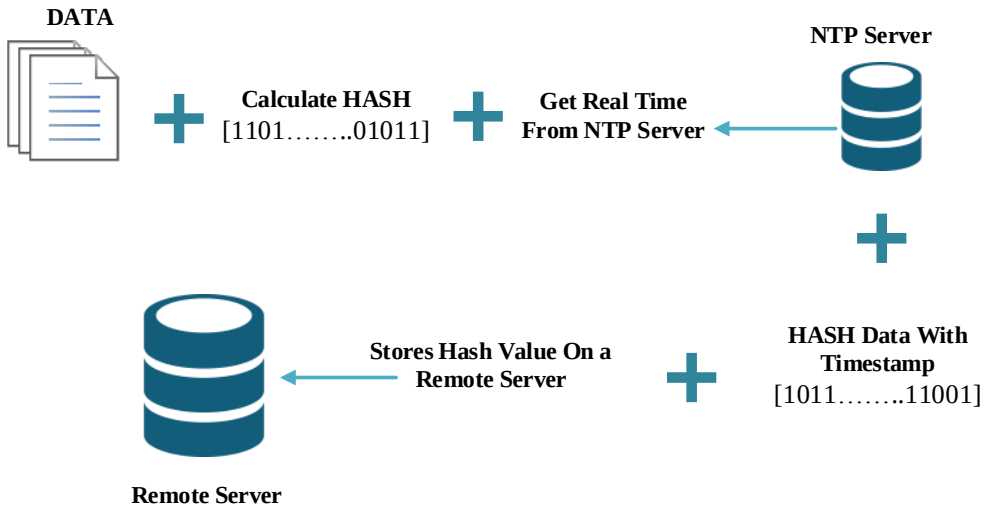



Figure 3.2: Timestamp Algorithm

In Figure 3.2, an example of timestamp algorithm for using to prove log files not being changed by anyone or anything. By using this algorithm, after storing data with timestamp, if it is wanted to be checked whether log file is original or changed, just calculating hash value of file and comparing it with stored in database. If these two values are equal, we can surely say that file is original. Therefore, while storing these hash values or log file on remote database, some external attacks may happen. At least, using a firewall will help us to define rule sets to check applications and users which want to access database. In firewall administration panel, we can define accept rules and reject rules [9]. For example, in Figure 3.3, we can define rules for Application source, source port, destination, destination port, and protocol. In a basic way, we can only accept our server to talk with each other on defined ports.

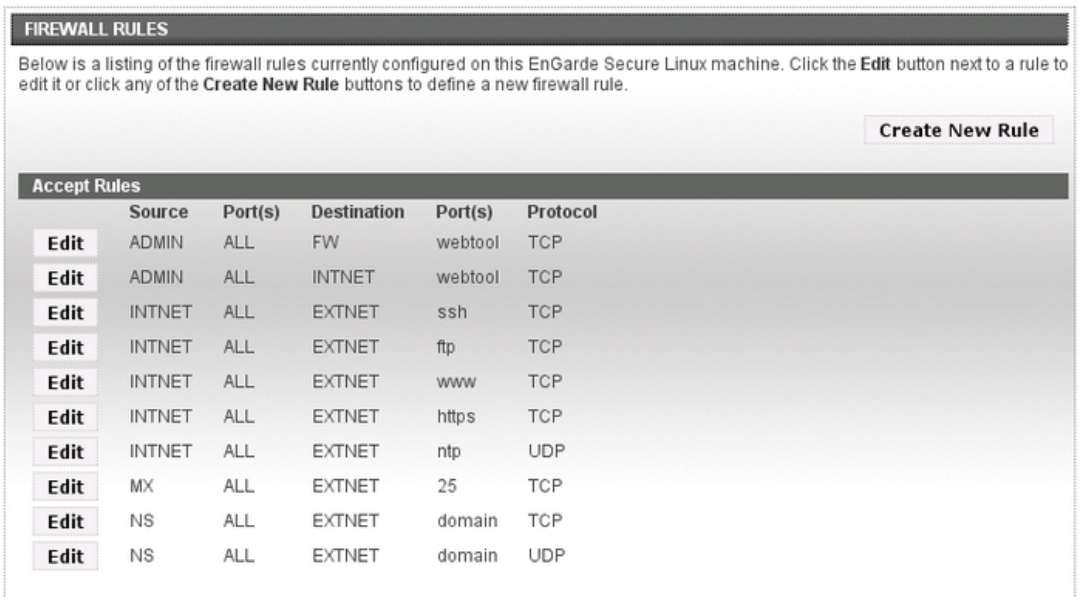



Figure 3.3: Example of Firewall Rules

## 3.3 Our Proposed System

Our proposed system is based on Infofence product which provides database security in only Oracle Databases. The biggest disadvantage of Infofence product according to its opponents is storing audit logs on the same server as Oracle Database. IBM Guardium, Imperva, Oracle Security Products (Audit Vault, Database Firewall) keep audit logs on different servers. The main working principle of these products are nearly same, agents gather logs from databases

and transferring these logs to other agents on a different server. Other agents store these logs on their databases. The end users connect second servers and manage security options and take reports from those databases. In this system, we want to improve Infofence product to meet requirements of a database security system. Infofence will be used on many databases at the same time, collect and transfer logs to the main server. End users will connect main server and run reports from main server.

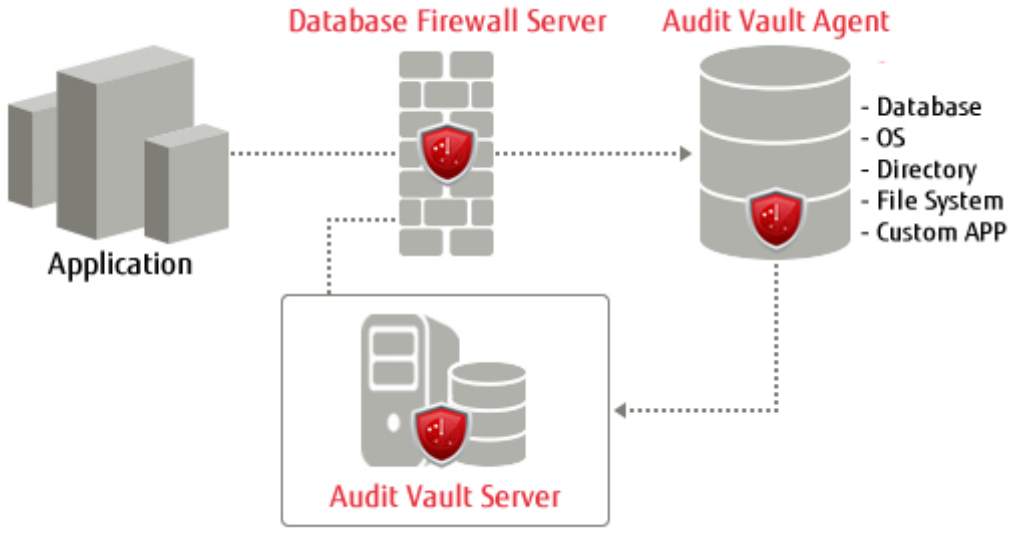



Figure 3.4: Oracle Audit Vault Working Principle

### 3.3.1 Two Based Agent System

In two based agent system, there are two different agents on the environment system. In database security environments, products collect audit logs and define security parameter according to these audit logs. Therefore, being secure and accurate of these logs becomes very important for security products. While keeping logs outside of production database, also it is important to secure these logs while transferring. Data can be corrupted or lost while transmission process. It is an unacceptable situation. It is vital that these agents must be careful on log shipping and they ship all the logs as well.. When agent is stopped for some reasons, agent must continue from where it last left off. On the other hand, agents must be careful on duplicate logs and rows. We described two based agent system like a pool system. When

the facet is opened, water must flow to the pool. Opening and closing facet must not affect the main process. When it is opened, water must be flowed, when it is closed, water must be stopped.

In two based agent system, first agent is installed on production database server. This agent runs at background, collects, and transmit logs to centralized server. IBM Guardium and Imperva agents listen network and filter pre-defined SQL words and commands. Oracle Audit Vault agents run likewise on Non-Oracle databases. Oracle database generate audit trails to database tables or Operating System files. Audit vault agents, collect these trails and transfer them to centralized server.

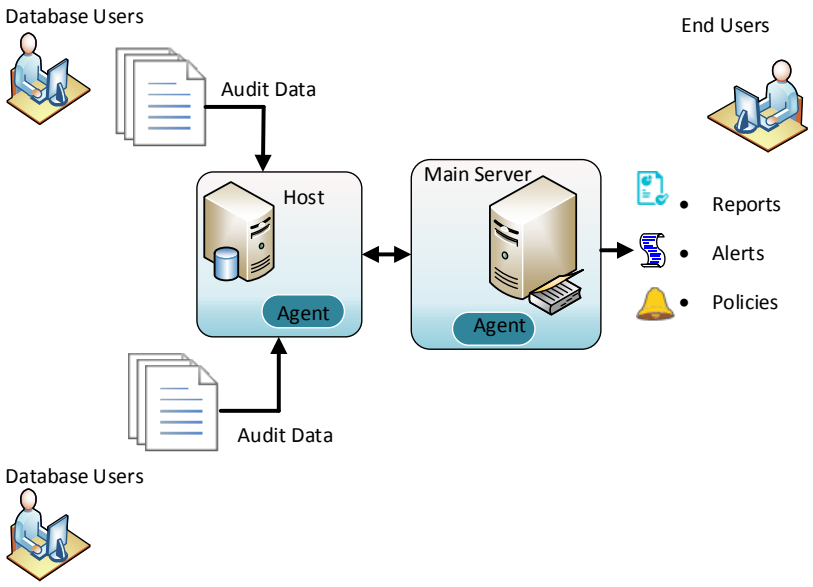



Figure 3.5: Our Two Based Agent System

As seen on Figure 3.5, we developed a two based agent system for real-time data transferring. Agents are running at operating system. First agent collect audit data like database operations, file system operations or OS operations. On the database servers, logs are collected and analyzed according to security operations. After that, it sends these data to second agent on the centralize server. Second agent is set for getting data from only first agent. Finally, logs are stored in databases on the centralized server. End users and security managers connect to security product control panel on the centralized server and generate reports, activity logs and

manage security operations.

Infofence is developed for only Oracle Databases. Mr. Günçer stated that, they have a great knowledge and know-how on Oracle databases. They are very successful on Performance and Tuning problems on Oracle Databases. Using this knowledge, they have become very well-known in the database security market thanks to Infofence performance. On the contrary to most database security products, Infofence end users have not problems on generating reports from server.

From that perspective of view, we developed our proposed system according to compete with the other products. System gets logs from Infofence tables on the production databases, and transfers securely to the centralized server. On a centralized sever, we used MySQL Database. If we use Oracle Database instead of MySQL database on the centralized server, probably we will get better solution on performance. However, Oracle is a paid product and its price is calculated according to CPU core count and processor type. When we calculate license price for a big system, Infofence will be much more expensive than other products. On the other hand, MySQL is an open source and free-licensed database system. In addition to many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, and Alcatel Lucent rely on MySQL to save time and money which boost their high-volume Web sites, business-critical systems and packaged software. On the other hand, MySQL INNODB engine infrastructure is just like Oracle Database infrastructure. Therefore, both Oracle and MySQL are Relational Database Systems which enable us to use transactions on data transferring. Depending on these features, MySQL is the best solution for our proposed system.

Our proposed system consists of two parts. First part is Data Transferring Process, and the other one is MySQL Database Data Parsing Process.

#### 3.3.1.1 Data Transferring Process

Infofence already collects audit logs on Oracle Database and stores them on Infofence tables. Likewise the current Infofence already provides database security on Oracle Databases as well. For logs security and managing multiple databases on a central server, Infofence must transfer these logs to the main server. Our proposed system's first agent works for getting

logs from Infofence tables and transferring them to the central server. The main point is that, the agent should not cause any delay on production database and system must provide log security for transmission. Our proposed system can support multiple database at the same time. Infofence will be installed every Oracle Database. On the MySQL database, there will create different schemas for every Infofence on Oracle Databases. Audit logs in Infofence tables will be transferred into defined schemas in MySQL database.

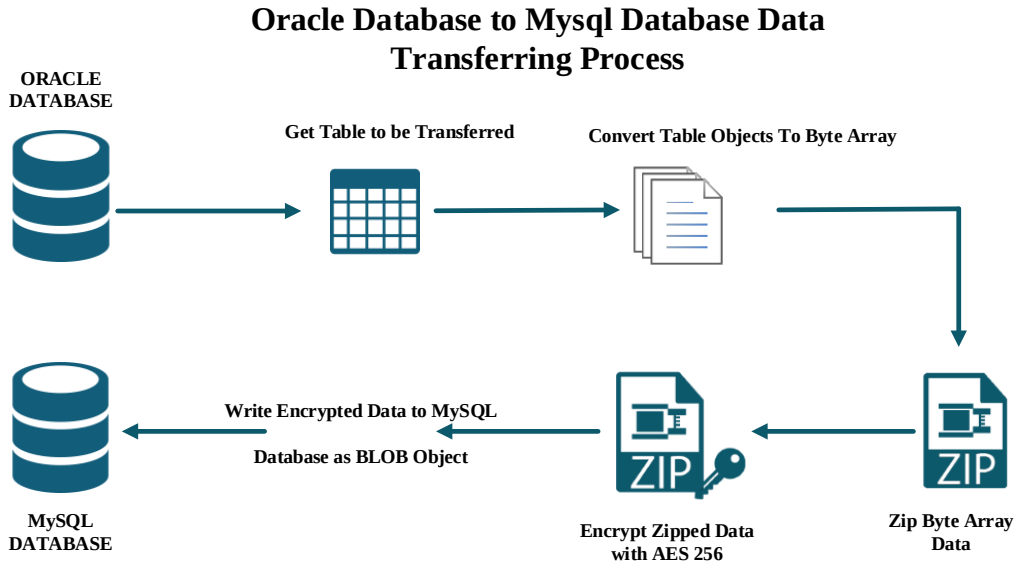



Figure 3.6: Data Transferring Between Oracle Database and MySQL Database

As seen in Figure 3.6, data transferring occurs between Oracle Database and MySQL Database. Actually, our first process is running with multi-thread. From control panel table, thread count can be set. In our development environment, we tested our proposed system with 2, 4, and 8 threads. When we increased thread count to 8 thread, our proposed system had some performance issues on Oracle Database. Hence, we decided to use 4 threads for multi-threading. Program automatically calculates data count which will be transferred and then it evenly distribute data to threads. In Figure 3.6, working principle of our program is shown. Every thread in our first process runs with using this algorithm. There are 6 phases of data transferring and log shipping is occurred securely from servers. Data processing steps are listed as follows:

- **Getting Tables to be transferred from Oracle Database**

Program gets list of tables from Infofence which will be transferred. For each table in the list, we checked previous transfers. Logs accuracy and not having duplicate logs are important for our system. To provide log accuracy, get maximum ID of last transferred log. ID columns are unique in all Infofence tables. We can follow transmission by observing ID values. Transferred logs count, ID, checksum, size is stored on database. When transmission restarts, it checks last successful transferred logs; afterwards, starts process from minimum ID of logs to be transferred.

We use Object Relational Mapping (ORM) framework on our proposed system. Every data in Oracle and MySQL Tables are used as objects in programs. We have specific tables in Infofence schema, so according to it, we defined classes for every table in the Infofence. By using these classes, we can map every table and use them as an object in coding. When we think about huge volumes of data in production systems, ORM framework will save us very important time and performance. Before we had finish our proposed system, we tested classical way of transferring data and parsing into MySQL tables. Then, we use ORM framework on the same data. While the volume of data increases, the performance of classical way is decreasing. On the other hand, ORM framework is nearly 20 times faster than classical way. Hence, we decided to use ORM framework.

- **Converting Table Objects to Byte Array**

We get data from tables which will be transferred in the first step. However, before log shipping, these tables must be converted to byte array. The main purpose of converting table object is that our proposed system does not generate or create any OS files from server. Every action is processed on the fly, which means, agents do not create audit trails (files) for log shipping. Byte array objects are transmitting over network to other server.

- **Zipping Byte Array Data**

Zip process is used for compression. One of the main difficulties of database security problems is to transfer huge log files to centralized server. Database servers and main server are connected by network cables or they can connect each other over network switches. Every

network connection has a quota for uploading and downloading files. To reduce transferring log size, we used zip method. Our system takes bytes array objects and returns as a zipped byte array object nearly 10 timed smaller than previous size. This enables us to gain time and performance on log shipping.

- **Encrypting Zip Data With 256 Bit AES**

One of the most important phases of log shipping process is encryption. Logs are transferring over network to central servers. Malicious users might listen network and get our sensitive data which is very important. We use database firewall for unauthorized accesses. However, according Burtescu [5], most of the attacks occurs from inside of the organization. Some users can pass database firewall and do not have any authority to access audit files. Encryption is the best option on securing data from both inside and outside attackers. In encryption phase, we use 256 bit block sized Advanced Encryption Standards (AES). It is nearly impossible to crack this algorithm.

- **Writing Encrypted Data to MySQL Database as a BLOB Object**

Final phase of data transferring process is to write encrypted byte array data to MySQL as a BLOB object. Database management systems have different data types like, number, varchar, date, BLOB, and etc. Data transferring part works for only log shipping. In this part, we transfer data as byte array objects and store these objects as BLOB data in MySQL Database. Parallelly, in Oracle we have another Migration table which stores tables transferring status. We can double check log shipping process, by using to Migration tables.

- **Data Transferring PSEUDOCODE**

**Algorithm 1:** Data Transfer

**Input** : No Input

**Output**: List of transferred table names

**begin**

Open a new Oracle Hibernate Session

Initialize tableName to empty

Get status from database controlPanel table

```
/* if status is true, data will be transferred to MySQL
Database */
```

**while** *status is true* **do**

Get list of table names to be transferred from Oracle Database

**for** *Each table in the list* **do**

Set tableName to next iterator

Initialize result with CALLing Transfer with tableName

**if** *result is true* **then**

print tableName is transferred

**else**

print There is problem on tableName transmission

Get status from database controlPanel table

Close Hibernate session

In Algorithm 1, first agent runs main program. It get list of tables to be transferred from Oracle Database. Infofence already collects and stores audit logs to Infofence tables. We get list of these tables and by using for loop, we call Transfer function with table name. Transfer function returns "True" or "False" Boolean value. If return value is true, selected table has successfully transferred to MySQL database, if value is False, these is a problem on transferring process.

**Algorithm 2:** Transfer Table

```
Input  : Table name to be transferred
Output : Status of Transmission
begin
    Open a new Oracle Hibernate Session
    Initialize threadCount to 4                          /* default 4-thread-multiThreading */
    Initialize firstId and lastId to empty
    Get lastId in the records of the table migrated to the MySQL
    if lastId is not empty then
        Set firstId to lastId
    if firstId is not empty then
        Get Id column values From tableName Where ID column is greater than firstId
    else
        Get Id column value From tableName
    Initialize minId to table list's minimum Id
    Initialize maxId to table list's maximum Id
    if minId is not empty and maxId is not empty then
        Initialize diff to dividing the value of difference between max Id and min Id to thread count
        Initialize threadList with threadCount
        Initialize tMinId to minId
        Initialize tMaxId to minId plus diff
        for threadId =1 to threadCount do
            Initialize thread to CALLing TransferData with tMinId and tMaxId and tableName
            Add thread to the threadList
            Start thread                                 /* Thread will run at background  */
            if threadId equals to threadCount then
                SET tMinId to tMaxId
                SET tMaxId to maxId
            else
                SET tMinId to tMaxId
                SET tMaxId to tMaxId plus diff
    else
        return false
    return true
    Close Hibernate session
```

Algorithm 2, shows the process of transfer tables. Our proposed system runs 4 multi-thread by default. 4 threads provided best performance on our development environment. We tested

with 2, 4, and 8 threads, and we had some performance issues on Oracle database when using 8 threads. However, thread count can be set from control panel on the database.

**Algorithm 3:** Transfer Table

**Input** : minimumID, Maximum ID, and Table name to be transferred

**Output**: No Output

**begin**

Open a new Oracle Hibernate Session

Initialize arraySize to 2000 `/* Default Array Size is 2000 */`

Get all data From tableName Where Id Column between minId and maxId

Initialize tableList to table data

**if** *size of tableList is greater than arraySize* **then**

Initialize mod to divide tableList size to arraySize

Initialize remain to tableList size MODE arraySize

**for** *i=0 to size of tableList* **do**

Initialize nTableList to sub list of tableList from i to i plus mod

CALL WriteList with tableName, nTableList, maximum Id from nTableList

Increment i with mod

**if** *remain is greater than zero* **then**

Initialize tSize to (mod multiply arraySize)

Initialize nTableList to sublist of tableList from tSize to tableList size

CALL WriteList with tableName, nTableList,maxId

**else if** *size of tableList is zero* **then**

CALL WriteList with tableName, tableList,maxId

**else**

CALL WriteList with tableName, tableList,maxId

Close Hibernate session

Transferring function checks tables' previous transmission logs. Then, it decides to start from minimum ID which is not transferred before. It calculates maximum and minimum IDs to be transferred and divides records in accordance with thread count. Every thread calls Transfer

Table function with start and finish IDs.

Main function of Algorithm 3 is to get table names, to start and to finish IDs from previous function and call WriteList function. In this function, there is a limit on records of transferred tables. Our proposed system get data from tables, zip and encrypt data to byte array data. While transferring these data to centralized server, we store these data on BLOB columns in MySQL tables. There is a limit on maximum size of data types. By default, maximum record to be transferred is 2000 rows for not exceeding maximum size of BLOB columns. This value can be set from control panel; however, we do not recommend to any bigger values than 2000.

**Algorithm 4:** Write Encryppted Data to MySQL Tables

**Input** : Table name to be transferred, Sublist of Table Data, Maximum ID

**Output**: No Output

Open a new Oracle Hibernate Session

Open a new MySQL Hibernate Session

BEGIN Oracle Transaction

BEGIN MySQL Transaction

**begin**

Initialize arrayData with Converting tableList to BYTE Array

Initialize cksum with Getting Checksum of arrayData

Initialize zipArrayData with Zipping arrayData

Initialize encryptedData with Encrypting zipArrayData

```
/* Data is stored in MYSQL with encrypted and zipped */
```

Insert into MySQL Migration table with values tableName,encryptedData

```
/* Oracle Migration table stores last transferred records */
```

Insert into Oracle Migration Table with values tableName,cksum and maxId

End Oracle Transaction

End MySQL Transaction

Close Oracle Hibernate session

Close MySQL Hibernate session

Final phase of Data Transferring process is Write function. In Algorithm 4, data is first zipped and then encrypted. As a result of this, we can provide secure and low-size transferring. For preventing from duplicate records or untransferred records, all activity of transmission is log on Migration tables. On Oracle side, last transferred table and row unique ID stores in

Migration Table. Program first checks the last transferred data from this table. On MySQL side, Migration table stores last parsed data. As a result of these, transferring process is checked from both databases. All operations on the database are done with in transactions. If there is a problem on somewhere of transmission, all actions of those tables will rollback. If there is not a problem on log transferring, commit process is done.

#### 3.3.1.2 MySQL Parse Process

The most important part of our proposed system is inserting data to MySQL without any changes or data loss. Second agent works only for parsing encrypted BLOB data to predefined tables. Basically, in log shipping, we could transfer tables, data directly from Oracle Database to MySQL Database tables. However, this process must be worked synchronously. If there is problem on first agent, this will cause wait on second agent or vice versa if there is a problem on MySQL database, it will cause first agent to wait until problem is solved. Hence, our aim is to ship data to the centralized server asynchronously, and second agent will parse data at background without effecting any production database processes and performance.

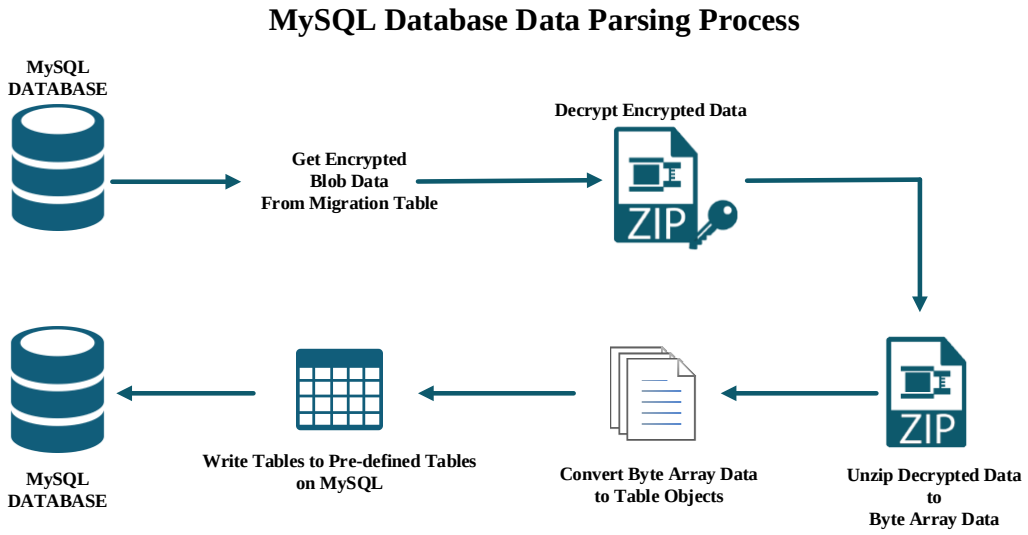



Figure 3.7: MySQL Database Encrypted Data Parsing into Tables Process

As seen in Figure 3.7, encrypted data parsing occurs on only centralized server and only at MySQL Database. Parsing processing steps are exact opposite of data transferring processes.

- **Getting Encrypted Blob Data from Migration Table on MySQL Database**

In this part, our second agent gets unparsed data from Migration table in MySQL database. Migration table stores, parsing status, data, and table name of logs to be transferred. We store encrypted data as BLOB type in tables. When it gets encrypted data from BLOB columns, agent controls checksum of data with its original checksum. If these values are different from each other, agent warns that there is a corruption on that table transmission.

- **Decrypting Encrypted Blob**

First agent, encrypted BLOB data with 256 Bit AES algorithm. The data is encoded using some encoding algorithm. We generate specific key to encrypt and decrypt log data. An unauthorized user who access encoded data will have difficulty deciphering it. Only second agent has keys to decipher data. Key can be changed from control panel, however, you must be careful on not changing already encrypted data's key.

- **Unzipping Decrypted Data to Byte Array Data**

To gain from performance and time, we use zipping method in the first agent. In this sense, we must unzip data before use it. We perform zipping operations on server sides, which enables us to carry small data and gain more performance. Zipping and unzipping processes takes some time on our system; however, it takes much more time to transferred original size of data from Oracle to centralized server.

- **Convertting Byte Array Data to Table Objects**

We use ORM framework for data transmission and in ORM framework, tables stored as classes, objects with data in program. First agent converts these objects into byte array to use zipping and encryption algorithm. Second agent performs exactly opposite of the first agent and converts these byte array to previous ORM table objects. Also, data is automatically generated in these tables. We create uniques classes for Infofence tables which are suited for both Oracle and Mysql tables. Hence, transferring classes are unchanged in both Oracle and MySQL database, so this makes it easy to insert data to MySQL tables.

- **Writing Table Objects to Pre-defined Tables in MySQL Database**

Final phase of second agent and our proposed system is writing data in table objects into predefined tables in MySQL Database. Before inserting data to tables, we control checksum of data in first agent with in the second agent. If checksum values are different from each other, we accept it is a corruption and do not insert data into tables. What is more, we control not inserting duplicate rows into tables and not leaving any logs not to be transferred. These controls are very important for transferring process. The biggest feature of the audit systems must be providing log accuracy in log shipping and log storing. After controls has finished, we insert data into tables and delete BLOB data from Migration tables. All operations are occurred by database transactions. We control that all the data are transferred successfully, or no data will be transferred.

- **MySQL Data Parsing PSEUDOCODE**

**Algorithm 5:** Parse Encryppted Data to MySQL Tables

**Input** : No Input

**Output**: Print Name of Parsed Tables

**begin**

Open a new MySQL Hibernate Session

BEGIN MySQL Transaction

`/* if status is true, data will be parsed to MySQL Tables */`

Get status from database controlPanel table

**while** *status is true* **do**

Get list of not parsed encrypted data, tableName and cksum value from Migration Table

**for** *Each data in the list* **do**

Initialize tableName to next iterator's tableName

Initialize encryptedData to next iterator's encryptedData

Initialize decryptedData with Decrypting encryptedData

Initialize unzipArrayData with Unzipping decryptedData

Initialize tableList with Converting unzipArrayData to tableList by tableName

Initialize cksum with Getting Checksum of the arrayData

**if** *cksum equals to orjinal checksum of the arrayData* **then**

Insert all records in tableList to MYSQL database

Delete transferred record from MySQL Migration tableList

**else**

Print Data is corrupted

Get status from database controlPanel table

End MySQL Transaction

Close MySQL Hibernate session

In Algorithm 5, program gets list of untransferred tables. Using a "For Loop", data is respectively encrypted and unzipped and then create table objects with using ORM framework. The program also checks the counting of the tables, checksum of data sizes. We stores original

checksum of data on Oracle Migration tables. Before parsing data to MySQL tables, checksums are automatically controlled on both MySQL and Oracle. If checksums and row sizes of tables are equal to each other, then program writes table object data to MySQL tables by using transactions algorithm. If program raises an error in the middle of parsing operation, all actions and data will be rollback.

#### 3.3.1.3 Salient features of the Proposed Model

Our proposed system enables InfoFence to provide database security for multiple databases in that it can be installed into many Oracle Databases simultaneously. After Infofence is installed, we create new schema for each Infofence in MySQL Databases. Infofence has constant tables and procedures, so after creating schemas in MySQL Database, we create tables and functions, too.

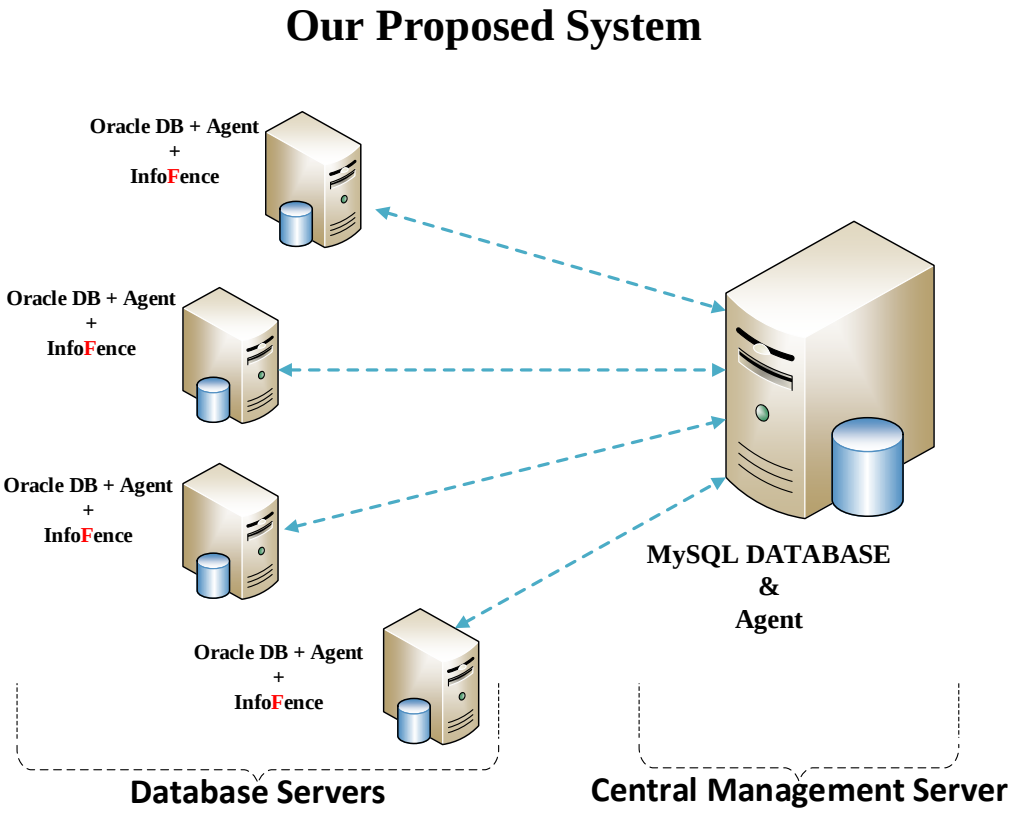



Figure 3.8: Our Proposed system - Two Based Agent System

As seen in Figure 3.8, our proposed system can support database security for multiple Oracle databases by using Infofence. One of the main features of our system is that there is a firewall between centralized server and Oracle database servers. Purpose of using firewall is to block unauthorized users and machines to access centralized server. There is very important sensitive information on centralized server. It is of utter importance that it remains unchanged for database security. Only security admin is allowed to access centralized server.

On the other hand, our proposed system support log shipping between hybrid databases synchronously. The system gets the data from Oracle databases and insert into MySQL databases without any data loses or corruption.

#### 3.3.1.4 Oracle Audit Vault Encountered Problems

We have developed our proposed system based on AV working principle. Infofence runs at only Oracle Databases and the best DAM product for Oracle Databases in the market is ORACLE Company's own product, Oracle Audit Vault. However, before developing our proposed model, we search about AV errors, failures, and lacks.

There are some issues proposed by clients which user using AV. Oracle Corporation has a customer support site which is support.oracle.com to help customer on their products. In addition, there can be bugs which customers are confronted. For instance, in Oracle Audit vault version 10.2 and later there is a bug that is caused a bad performance of the Oracle Audit Vault Database which is caused by the audit agent collectors.

On the Audit Vault database the processes created by the collectors are consuming much of the CPU power. The AWR reports show that there are significant waits caused by sequence contention. Normally the sequences associated with the AV DW dimension tables have a cache of 20 transactions. If we can increase cache to 2000 on the Audit vault database, there will be no more waits but the server's CPU usage will increase even more. In other words, this problem can happen for any dimension table if that dimension table has a large number of rows. To illustrate, the audit vault agent collectors and the AV database are slow whenever the TARGET dimension table has a large number of rows for a certain OWNER. Let me explain the problem by a specific example:

**Output 3.3.1** *Audit Vault Server Target Count*

```
SQL> select count(*), substr(owner_name,1,20) from target_dim
                group by owner_name order by 1 desc


COUNT(*)        OWNER_NAME
--------        ----------------------------------
   16227        SCOTT
      59        SDNG_DBA
       7        SYSMAN
       6        TONI
       6        DBSNMP
       4        TEST
       3        ORALINUX
       3        ORA_LINUX
```

Now we can see that SCOTT owner has many target dimensions compared to the other owners. Because of this while the agent collector is running the following statement is consuming a lot of resources and has a lot of waits:

**Output 3.3.2** *An Example of Huge-Cost Statement*

```
MERGE INTO AVSYS.TARGET_DIM USING (SELECT :B1 TARGET_OWNER FROM DUAL) D
        ON(TARGET_DIM.OWNER_NAME = D.TARGET_OWNER)
                WHEN NOT MATCHED THEN INSERT(TARGET_DIM.DIMENSION_KEY,
 TARGET_DIM.OWNER_ID,TARGET_DIM.OWNER_NAME) VALUES(
 TARGET_DIM_SEQ.NEXTVAL, TARGET_DIM_SEQ.CURRVAL, D.TARGET_OWNER)
LOG ERRORS (TO_CHAR (SYSDATE, 'dd-mon-yy')) REJECT LIMIT UNLIMITED

call count cpu elapsed disk query current rows
------- ----- -------- ---------- ---------- ---------- -------
Parse 0 0.00 0.00 0 0 0 0
Execute 53 67.29 111.09 0 7062 44139 0
Fetch 0 0.00 0.00 0 0 0 0
------- ------ -------- ---------- ---------- ---------- ------

Total 53 67.29 111.09 0 7062 44139 0

Misses in library cache during parse: 0
Optimizer mode: ALL_ROWS
Parsing user id: 46 (AVSYS) (recursive depth: 2)
```

This SQL statement provides synchronization between target database and AV database. The fact that Oracle get rid of these symptoms can be seen only after the upgrade to 10.2.3.2 since in earlier AV versions the collectors are not updating the data warehouse fact table or the dimension tables.

Therefore on AV 10.2 Version and later, Audit vault causes huge CPU cost on generating log reports. As a result of this, audit vault is hung and log shipping is cancelled. The Audit Vault OS Collector can either consume all the CPU power on the source machine while moving audit data to the AV Repository or it can consume less resource yet not moving any data to the repository. This problem occurs because of the fact that in the audit file destination folder of the source database there are more than 10000 log files. If you want to avoid this you must remove the audit files produced by the source database as soon as they are collected. The audit files can be removed with a custom OS script or can be removed automatically via the AUDIT purge job. The advantage of using this job is that you can be sure that only the collected files will be removed. To enable this job, run the below command.

**Example 3.3.1** *Audit File Clean Job*

```
BEGIN
   DBMS_AUDIT_MGMT.INIT_CLEANUP(
        AUDIT_TRAIL_TYPE => DBMS_AUDIT_MGMT.AUDIT_TRAIL_FILES,DEFAULT_CLEANUP_INTERVAL => 2 );
END;
/
```

When the DBMS_AUDIT_MGMT purge job runs, it will remove all the audit files older than LAST_ARCHIVE_TS. If the purge job is executed frequently then the number of the audit files will not increase too much. If DBMS_AUDIT_MGMT is configured on the source database, the collector is updating (increasing) the last archive timestamp value from audit view, DBA_AUDIT_MGMT_LAST_ARCH_TS. We can query this view for audit trails.

Another strategy to reduce the number of files is to make sure that the audit data for one session is not split into many small audit files. For this set on the source the following two parameters in Example 3.3.2.

**Output 3.3.3** *Audit Vault Server Last Archive Status*

```
Select * from DBA_AUDIT_MGMT_LAST_ARCH_TS;

AUDIT_TRAIL              RAC_INSTANCE   LAST_ARCHIVE_TS
----------------         ------------   -----------------------
OS AUDIT TRAIL                      1   30/10/08 13:07:08,000000 +01:00
XML AUDIT TRAIL                     1   30/10/08 13:07:08,000000 +01:00
STANDARD AUDIT TRAIL                0   08/01/09 13:20:27,000000 +00:00
FGA AUDIT TRAIL                     0   08/01/09 13:20:27,000000 +00:00
```

To avoid these problems, you must ensure removing more aggressively the reference to audit files as soon as they are physically deleted from the audit file destination directory of the source. In this way the collector will consume less memory and less CPU.

**Example 3.3.2** *Setting Number of Audit Files for Each Session*

```
BEGIN
DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY(
audit_trail_type => DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS,
audit_trail_property => DBMS_AUDIT_MGMT.OS_FILE_MAX_SIZE,
audit_trail_property_value => 102400 /* 100MB*/ );
END;
/

BEGIN
DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY(
audit_trail_type => DBMS_AUDIT_MGMT.AUDIT_TRAIL_OS,
audit_trail_property => DBMS_AUDIT_MGMT.OS_FILE_MAX_AGE,
audit_trail_property_value => 1 /* day */);
END;
/
```

The other problem related to performance on Audit vault is caused by database log collector agents when AUD$ Table is very large. Firstly, Audit vault stores audit and action logs on database AUD$ table. Secondly, agents are transmitting logs to Audit vault Server asynchronously. When logs are transmitted to AV server, they are being deleted from AUD$ table. Sometimes, agents are being stopped or there can be some problems on agents working. In this sense, the database collector of the Audit Vault causes massive performance issues on source databases because of time consuming full table scans on the aud$ table. This happens

when the aud$ table is very large. Since the collector brings the data in batches of records based on time, this statement using a filter on the NTIMESTAMP# column is used for this purpose. There never was an index on the NTIMESTAMP# column in aud$. However, with audit vault, it is supposed that the audit records would be stored inside the audit vault database and aud$ would be of smaller sizes. The performance effort was directed towards the audited sessions, by eliminating all updates on aud$ and by removing the indexes on this table (in order to avoid the hot index blocks when many audited sessions run simultaneously). The aud$ should be of small sizes in 10g and higher releases. There are some solutions for these problems. The first solution is to control the working status of agents in short intervals. If agents are not working, then automatic start scripts will be run. When AUD$ becomes very large, it is suggested to create index on the ntimestamp# column. Create the index with a syntax similar to:

**Example 3.3.3** *Create Index on AUD$ Table*

```
create index ntime_aud$ on aud$(NTIMESTAMP#) tablespace <new index tablespace>;
```

# CHAPTER 4

# EXPERIMENTS AND EVALUATIONS

In this chapter, a prototype implementation will be presented for context awareness, and log shipping between hybrid databases for multi-domain environment presented in Chapter 3. The aim of this prototype is to validate and show the applicability of the creating a private database security cloud with using Infofence. Prototype will also be configured for two different cases in order to analyze proposed model's applicability. Firstly, system architecture and detailed design of the prototype will be presented, and then prototype will be compared to one of the audit product on the market, Oracle Audit vault.

## 4.1 System Design

Firstly, to generate a prototype system, we need at least two different databases, Oracle database and MySQL database. Actually, these two databases can run on the same server. However, log files cannot be on the same server with production database. If your production system facilitates an extra storage for database servers, you can separate different storage areas for Oracle and MySQL. Our prototype is implemented on two different servers; that is, one of them runs Oracle database, and the other one runs MySQL database. To run two servers on the personal computer, we use Oracle Virtual Box, an enterprise virtualization product. By using a virtualization tool, many servers can be created by sharing system disk, memory, and CPU.

In Figure 4.1, Oracle Database Server, in Figure 4.2 MySQL Database Server in prototype are presented. We have created a 4GB Ram, 2CPU, and 200GB HDD virtual machine for both Oracle and MySQL databases. We installed databases on Oracle Enterprise Linux (OEL)

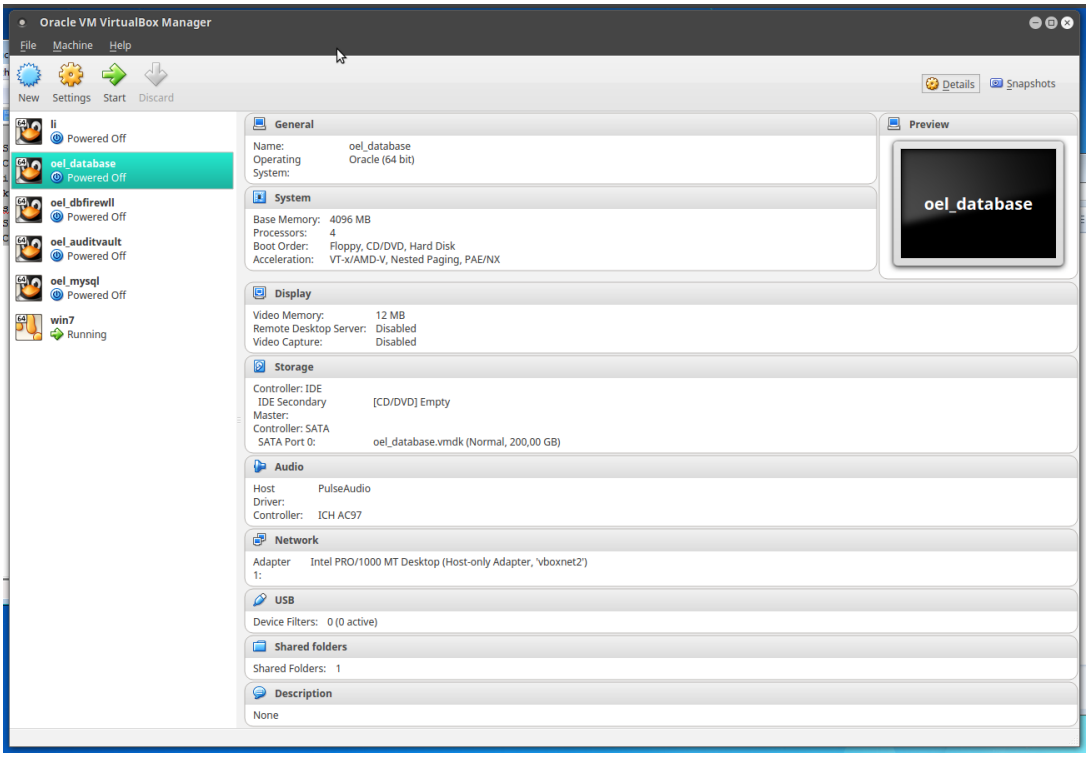



Figure 4.1: Oracle Database Server Properties

6.4 version. While we were creating virtual machines, Oracle database 11.2.0.4 version was certified by OEL 6.4, so we installed OEL 6.4 version on all servers. Both servers are designed for J2EE technology and is capable of running java files.

## 4.2 Evaluation Environment

To evaluate our proposed system's performance, we have created total 5 different virtual machines using Oracle Virtual Box. These virtual machines run, 2 different Oracle 11G Database Servers for transferring data from multiple databases, 1 MySQL Database Server for gathering and storing logs from Oracle Databases, 1 Oracle Audit Vault Server and 1 Oracle Database Firewall Server for comparing our proposed system.

Figure 4.3 shows AV properties and Figure 4.4 shows DBF properties. We installed audit vault with same characteristic with our proposed system to evaluate and compare equally.

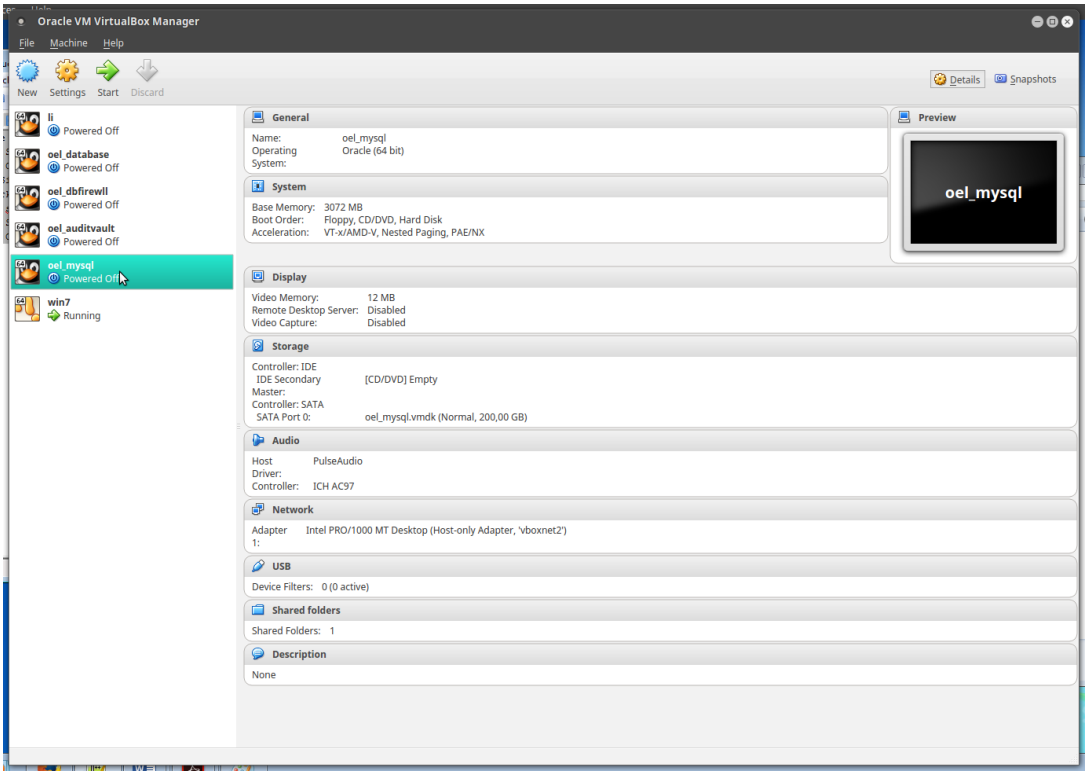



Figure 4.2: MySQL Database Server Properties

## 4.3 Performance

After analyzing AV, we stated that performance is the main problem on log shipping. It can cause both production database running slow and generating reports from audit logs. On the other hand, generated logs and reported logs must be same. Log shipping must not cause any log corruption. Main problems of Audit Vault server is having problems on stability and report generating on High Volumes of Data.

Audit products firstly start with collecting logs from databases, users, and operations. After a period of time, logs are shipping to main server for managing security and generating reports. In our environment, we installed Infofence on Oracle database and Infofence started to collect logs from all activities in the database. These activities are loginning to database, DML, and DDL logs, auditing users, and auditing statements on database objects, tables, index, sequences. After log sizes reached 2G, we started our agents to log shipping.

According to these result, we have evaluated our proposed system,s performance for 4 differ-

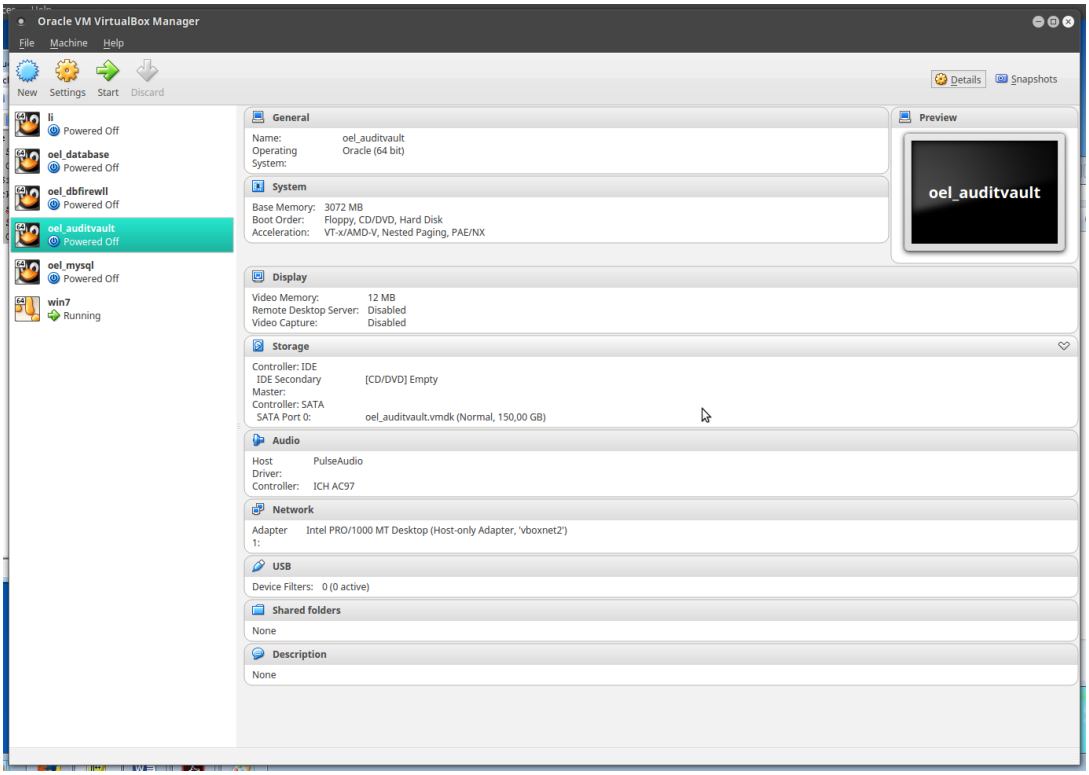



Figure 4.3: Oracle Audit Vault Server Properties

ent criteria: Elapsed time between log shipping, Accuracy of transferred logs, Stability, and Comparison with Audit Vault Server.

### 4.3.1 Elapsed Time

We use Multi-threading algorithm in our proposed system, the best thread count for our development environment was 4 threads. The network connection speed between Oracle Server and MySQL Server is 10MB/s. To calculate the network bandwidth between servers, we transfer a 4 GB file from Oracle server to MySQL server and transfer time is 7 minutes. It shows us the network bandwidth is 10MB/s. In order not to exceed network upload size and BLOB data type limits (2GB) in MySQL, we limited maximum row in an audit object with 2000. Some Infofence tables store BLOB, CLOB columns. So, a row maximum size can be nearly 1M for these tables and we limited row count in the audit object with 2000 rows. After we run our database transferring agent, the elapsed time for log shipping is 66sn. Total log size is 2432M. If we did not zip data on transferring process, log shipping process would be

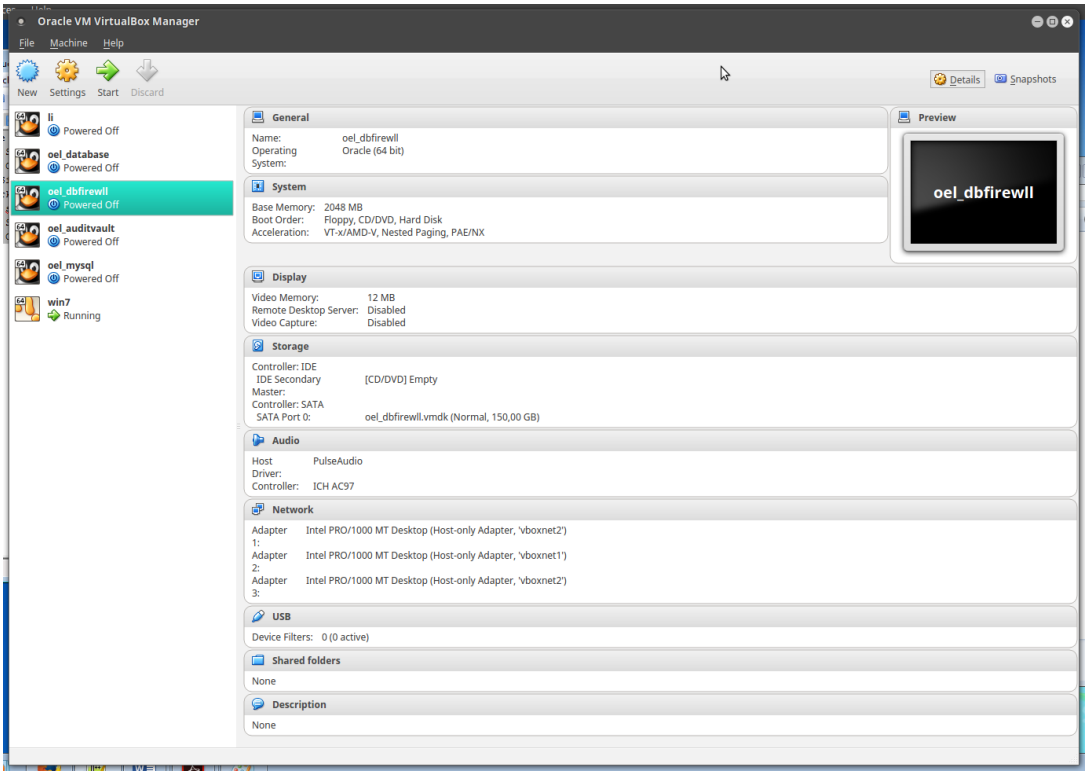



Figure 4.4: Oracle Database Firewall Server Properties

at least 4 minutes. Log sizes are decreased nearly 5 times smaller than original size by using zipping algorithm. After 30 sn from first agent on Oracle Server, we started second agent on MySQL server for parsing encrypted logs to pre-defined tables in MySQL. The elapsed time for parsing all data is 5 minutes 56 seconds. This time is much more than first agent because of writing data to disks are slower than reading data from disks. Total elapsed time for log shipping and data parsing is 7 minutes and 2 seconds for 2432 M log size. This result is an accepted for our system's performance. In production systems, our proposed system will instantly transfer all generated data from Oracle to MySQL. Hence, transferring 2 GB data in 66 seconds on our test environment is acceptable.

### 4.3.2 Accuracy of Logs

Our system's first agent is transferring every data into MySQL tables from minimum ID to maximum ID. After shipping a log to MySQL server, system stores last transferred record's ID. Therefore, if agent is stopped by some unexpected reasons, we started agent again, and

it continued from where it had stopped. To evaluate accuracy of logs, we demonstrated Infofence log tables in the Appendix A.

After second agent finished parsing of all encrypted data, we compared each table's row count in Oracle database and MySQL database. All tables' row counts are compared and both in MySQL and Oracle, row counts are same. After that, we compare data in the tables. One of the problem related to data transferring between different databases is column type diversities.

Table 4.1 shows the data type differences when replicating data from Oracle to MySQL. To check any data corruption, each data type is checked separately with tables. For example, the values are checked in DB_LOGIN_LOG table both in ORACLE and MySQL. The query we run in both databases is show in Output 4.3.1.

**Example 4.3.1** *Oracle Database and MySQL Database Db_Login_Log Table Comparison*

**Output 4.3.1** *Oracle Database Db_Login_Log Table:*

```
Select id, sessionid, db_user, os_user, login_time from db_login_log where ID=671;

ID     SESSIONID    DB_USER     OS_USER      LOGIN_TIME
--     ---------    -------     --------     ---------------
671    360584       SYS         oracle       13/04/2015 22.53.07.093410000
```

Table4.1: Data Type differences when replicating data from MySQL to Oracle

| MySQL Datatype | Oracle Datatype |
|---|---|
| Int | Number(10, 0) |
| BigInt | Number(19, 0) |
| Decimal(x,y) | Number(x, y) |
| Float | Float |
| Char(n) | Char(n) |
| Varchar(n) | Varchar2(n) |
| Date | Date |
| Datetime | Date |
| Timestamp | Date |
| Text | CLOB |
| BLOB | BLOB |

**Output 4.3.2** *MySQL Database Db_Login_Log Table:*

```
ID     SESSIONID    DB_USER     OS_USER      LOGIN_TIME

671    360584       SYS         ORACLE       2015-04-13 22:53:07.09341
```

Database security solutions must be careful on storing audit data and keeping data unchanged. Hence, we might be sure on transferring data securely to centralized server without any data loss or data corruption. We checked all tables' count on both Oracle Database and MySQL Database. They are all the same; however, there are some critical data types which might be problem for data transferring like BLOB columns. In Infofence, we store big text files whose sizes are bigger than 100M, on CLOB and BLOB columns. BLOB columns stores byte array and we cannot see data by using queries. Hence, we tested BLOB column's character size by using Length function in RDBMS. We tested data in each incompatible data type between Oracle and MySQL. All data in Oracle tables are transferred to MySQL database without any corruption.

**Example 4.3.2** *Oracle Database and MySQL Database BLOB Column Comparison*

**Output 4.3.3** *Oracle Database:*

```
select length(sql_text) from db_ddl_operation_log where id=15;

LENGTH(SQL_TEXT)
----------------------------------------
1261
```

**Output 4.3.4** *MySQL Database:*

```
select length(sql_text) from infofence.db_ddl_operation_log where id=15;
LENGTH(SQL_TEXT)
----------------------------------------
1261
```

We tested data in every incompatible data types between Oracle and MySQL. All data in Oracle tables are transferred without any corruption. These results show us the accuracy of our proposed system.

### 4.3.3 Stability

There are some problems on stability of DAM products. The agents collecting logs and making log shipping, might stop for unexpected reasons [25]. Security managers should usually control whether agents are running or not. It is one of the most important problems of AV. AV agents can be stop abruptly if you cannot observe audit agent's log files. Hence, we may sure on stability of our proposed system. We have run our proposed system for two weeks in our test environment. In these two weeks, it did not stop or not cause any waits on both production system and central server. Both our Oracle Server and MySQL server runs on Linux Operating systems. For understanding stability and system performance on the servers, we used 4 commands on both servers: Top, Vmstat, Tcpdump, and Iostat. Blum [4] states that for testing Linux Operating System's performance, we might use these four commands.

- **Top Command**

The top command is a performance monitoring program and used to display all the running and active real-time processes in ordered list and updates it regularly. It displays CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, Commands and much more. It also shows high memory and CPU utilization of running processes. It continuously updates screen in every 3 seconds.

**Output 4.3.5** *Oracle Database Server:*

```
top - 11:16:44 up 62 days 2:06,  2 users,  load average: 0,03, 0,02, 0,05
Tasks: 282 total,   11 running, 271 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.2%us,  0.2%sy,  0.0%ni, 99.4%id,  0.2%wa,  0.0%hi,  0.0%si
Mem:  16137980k total, 16030360k used,   107620k free,   669136k buffers
Swap: 16433148k total,   256608k used, 16176540k free,  8947644k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
12259 oracle    -2   0 3261m  15m  13m S  2.0  0.1 698:25.04 oracle
25201 oracle    20   0 3263m  34m  31m S  1.0  0.2   0:00.03 oracle
 9704 root      20   0 18392  596  444 S  0.3  0.0  13:10.60 irqbalance
10577 oracle    20   0 3391m 229m  916 S  0.3  1.5 453:36.72 mysqld
12011 oracle    20   0 9221m 649m 5132 S  0.3  4.1 403:58.07 java
12269 oracle    20   0 3266m  39m  33m S  0.3  0.2  77:33.12 oracle
12285 oracle    20   0 3262m 181m 178m S  0.3  1.2  26:27.38 oracle
12950 oracle    20   0 1107m 7480 1072 S  0.3  0.0 107:25.38 opmn
19828 oracle    20   0 7140m 332m 7192 S  0.3  0.1  428:19.99 java
52414 oracle    20   0 7140m 332m 7192 S  0.3  0.1  428:19.99 infofence
20372 root      20   0     0    0    0 S  0.3  0.0   5:01.68 kworker/6:2
22933 oracle    20   0 3261m  18m  16m S  0.3  0.1   0:39.66 oracle
    1 root      20   0 19416 1084  880 S  0.0  0.0   0:32.21 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.39 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   3:17.49 ksoftirqd/0
    5 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0H
    7 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kworker/u:0H
    8 root      RT   0     0    0    0 S  0.0  0.0   0:07.62 migration/0
    9 root      20   0     0    0    0 S  0.0  0.0   0:00.00 rcu_bh
   10 root      20   0     0    0    0 S  0.0  0.0  39:58.61 rcu_sched
```

Our agent is displayed by red colors and agent's process ID is 52414. As seen in top command, InfoFence process, run by a java process, and running for 428 hours. Its memory usage is 0.1 % of total memory of 16G. Its CPU usage is 0.3 %. Our Oracle Server is running for 62 days and actively there are 2 users on the system. There are total 282 processing and 11 of them is running. Remain processes are sleeping. Therefore, on CPU and Memory lines, it is clearly understood that our server have many free resources.

**Output 4.3.6** *MySQL Database Server:*

```
top - 11:16:44 up 62 days 1:06,  2 users,  load average: 0,00, 0,01, 0,05
Tasks: 251 total,   1 running, 250 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,0 us,  0,1 sy,  0,0 ni, 99,9 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 32748252 total, 23365224 free,  1531892 used,  7851136 buff/cache
KiB Swap: 68153344 total, 68153344 free,        0 used. 31079884 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   21 root      20   0       0      0      0 S   0,3  0,0   0:06.17 rcu_sched
24631 root      20   0  130152   2032   1312 R   0,3  0,0   0:00.04 top
    1 root      20   0   57180   4448   2476 S   0,0  0,0   0:04.02 systemd
  828 oracle    20   0    4926    532   8142 S  0.1  0.1  356:13.99 java
 2414 oracle    20   0    3836    532    8142 S  0.1  0.1  356:13.99 infofence
    2 root      20   0       0      0      0 S   0,0  0,0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S   0,0  0,0   0:00.09 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0,0  0,0   0:00.00 kworker/0:0H
    7 root      rt   0       0      0      0 S   0,0  0,0   0:00.15 migration/0
    8 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcu_bh
    9 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/0
   10 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/1
   11 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/2
   12 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/3
   13 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/4
   14 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/5
   15 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/6
   16 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/7
   17 root      20   0       0      0      0 S   0,0  0,0   0:00.00 rcuob/8
```

Our agent's process ID is 2414. As seen in top command, InfoFence process, run by a java process, and running for 428 hours. Its memory usage is 0.1% of total memory of 32G. Its CPU usage is 0.1%. Both in Oracle Server and MySQL server, our proposed system does not cause any wait or have huge memory and CPU usage. This command showed us to know what is happening in our MySQL Server at that moment. System resources and high-cost resources might be optimized by using top command in real-time.

- **Vmstat Command**

Vmstat command is a useful program which enables to collect and report data about linux server. Hence, it is used to monitor system resources usage including RAM, swap, and storage input/output and used to display statistics of virtual memory, kernel threads, disks, system

processes, I/O blocks, interrupts, CPU activity. Vmstat command can be used for determining the main reasons which cause the performance problems on memory use.

**Output 4.3.7** *Oracle Database Server:*

```
[oracle@ovmmanager ~]$ vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 256608 4090412 669136 8947708    0    0     0    36    0    1  0  0 99  0  0
```

Output 4.3.7 show that in the first line of the output, there are average values of our system resource since the last boot time of our server. It shows us the current state of a Linux system. In our Oracle Server, as seen from Vmstat command, free memory is 4090412. In this server, Oracle database is running and it uses 12GB ram. Remaining memory is nearly 4GB. So, our proposed system does not cost huge memory on the system. On the other hand, on the CPU tab, the idle CPU is 99%. Total system uses 1% of total CPU. when interpreting these results, it is clearly stated that our proposed system has no cost on system resources.

**Output 4.3.8** *MySQL Database Server:*

```
[root@mysqldb ~]# vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 23365508  58252 7792840    0    0     0    16    5   11  0  0 98  0  0
```

The “r” column in the “procs” tab display the total number of waiting processes which want to access processor to run. “b” column shows the sleeping processes. Both columns’ value are “0”. Hence, in our MySQL Server, as you see from Vmstat command, free memory is 24365508. In this server, MySQL database is running and it uses 8GB RAM. Total RAM of this server is 32 GB and total cost of all remaining process including our proposed system is nearly 1GB. In this server, using this much memory is an acceptable situation. On the other hand, on the CPU tab, the idle CPU is 98%. Total system uses 2% of total CPU. When this results exceed total CPU and RAM counts on the server, it shows that there is a CPU bottleneck exists, and some processes cause others for waiting to execute.

- **Tcpdump Command**

Tcpdump is a network packet analyzer or packets sniffer program that is used capture or filter TCP/IP packets that received or transferred on a specific interface over a network.

**Output 4.3.9** *Oracle Database Server:*

```
[root@ovmmanager ~]# tcpdump -i eth0 | more
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
:23:19.797323 IP ovmmanager.ssh > 192.168.2.122.63911: Flags [P.], seq 1392810370:1392810562
:23:19.797801 IP ovmmanager.47875 > Superonline-DNS.domain: 29162+ PTR? 122.2.168.192
:23:19.805256 IP ovmmanager.34070 > Superonline-DNS.domain: 25062+ PTR? 1.0.74.213
:23:19.812290 IP Superonline-DNS.domain > ovmmanager.34070: 25062 1/4/4 PTR
11:23:23.438432 IP ovmmanager.47875 > 192.168.2.14.58975: UDP, length 16
11:23:23.438486 IP ovmmanager.47875 > 192.168.2.14.52487: UDP, length 16
11:23:23.438617 IP ovmmanager.39300 > Superonline-DNS.domain: 19445+ PTR? 255.2.168.192
11:23:23.446497 IP Superonline-DNS.domain > ovmmanager.39300: 19445 NXDomain 0/1/0 (121)
11:23:23.446673 IP ovmmanager.62275 > Superonline-DNS.domain: 21786+ PTR? 120.2.168.192
11:23:23.461338 IP Superonline-DNS.domain > ovmmanager.34493: 15353 1/6/12 PTR
11:23:23.902691 IP 192.168.2.12.netbios-ns > 192.168.2.255.netbios-ns: NBT UDP PACKET(137):
11:23:24.666265 IP 192.168.2.12.netbios-ns > 192.168.2.255.netbios-ns: NBT UDP PACKET(137):
11:23:25.430719 IP 192.168.2.12.netbios-ns > 192.168.2.255.netbios-ns: NBT UDP PACKET(137):
11:23:26.280999 IP ovmmanager.47875 > 192.168.2.14.17500: UDP, length 124
```

Tcpdump command captured all network packets which are flowing through all the network interfaces of Oracle server. As seen in Output 4.3.9, this command is listening ETH0 port of our system and Ethernet bandwidth is 10MB. Our MySQL server IP address is 192.168.2.14. TCP packets sent from Oracle server to MySQL server is colored by red. Our first agent transferring encrypted logs to MySQL server and as seen in Tcpdump command, server is continuously sending TCP packets to MySQL. We cannot see inside of data packages by Tcpdump. This command shows that the agent is running and sending data from one server to another server. Also, Tcpdump shows the size of the sending data. For example, Oracle database is sending a 124 bytes data to MySQL server on 11.23.26.280999.

**Output 4.3.10** *MySQL Database Server:*

```
[root@mysqldb ~]# tcpdump -i enp7s0f0 | more
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp7s0f0, link-type EN10MB (Ethernet), capture size 65535 bytes
11:23:23.856294 IP mysqldb.ssh > 192.168.2.122.32893: Flags [P.], seq 3577855268:3577855464
11:23:23.856803 IP mysqldb.10840 > 192.168.2.1.domain: 48384+ PTR? 122.2.168.192.in-addr.arpa
11:23:23.857752 IP 192.168.2.122.32893 > mysqldb.ssh: Flags [.], ack 196, win 2267, options
11:23:23.873627 IP 192.168.2.15.domain > mysqldb.10840: 48384 NXDomain 0/1/0 (121)
11:23:24.874120 IP mysqldb.44975 > 192.168.2.1.domain: 28592+ PTR? 1.2.168.192.in-addr.arpa
11:23:24.890238 IP 192.168.2.15.domain > mysqldb.44975: 28592 NXDomain 0/1/0 (119)
11:23:25.320603 IP 192.168.2.12.netbios-ns > 192.168.2.255.netbios-ns: NBT UDP PACKET(137):
11:23:25.320750 IP mysqldb.26478 > 192.168.2.1.domain: 37665+ PTR? 255.2.168.192.in-addr.arpa
11:23:26.620912 IP 192.168.2.15.domain > mysqldb.26478: 37665 NXDomain 0/1/0 (121)
11:23:26.621129 IP mysqldb.38942 > 192.168.2.1.domain: 3922+ PTR? 12.2.168.192.in-addr.arpa
11:23:26.923756 IP 192.168.2.1.domain > mysqldb.38942: 3922 NXDomain 0/1/0 (120)
11:23:28.071809 IPX 00000000.00:04:00:aa:6f:0f.83c2 > 00000000.ff:ff:ff:ff:ff:ff.0452:
11:23:28.071820 (NOV-ETHII) IPX 00000000.00:04:00:aa:6f:0f.83c2 > 00000000.ff:ff:ff:ff:ff:ff.0452:
```

By using Tcpdump command, we can capture, display, and save network packets which goes through our network cards. Our aim is to show network traffic of our systems. As seen in Output 4.3.10, Oracle server IP address is 192.168.2.15 and red colors show that Oracle server is sending packages to MySQL server continuously. We can figure out that our first agent is sending audit data to centralized server and our system is running perfectly.

- **Iostat Command**

Iostat command is used for CPU statistics and collects and shows system's input and output storage device statistics. Iostat command generates report for average transfer rates related to our devices. Our proposed system reading data from Oracle Database and writing data to MySQL database. Hence, this command helps us to understand better balance of the input/output load on our storage disks.

**Output 4.3.11** *Oracle Database Server:*

```
root@ovmmanager ~]# iostat
Linux 3.8.13-44.1.1.el6uek.x86_64 (ovmmanager)         07/05/2015         _x86_64_        (12 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.30    0.00    0.11    0.38    0.00   99.21

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
sda              310.08       298.31       446.73   31473548 2226629590
dm-0             412.86       429.54       155.95    2670738  777324472
dm-1             270.02       278.01         0.16      69160     794712
dm-2             261.91       263.01        59.95      66162  298793120
dm-3             204.14       268.81        66.74   14015674  332637096
dm-4             342.16       264.45       129.77   12200706  646804650
dm-5             256.49       250.49        34.16    2422818  170275492
```

In this command, observing the time the devices are active in relation to their average reading and writing rates. As you see, there are total 6 disks on the Oracle Server and reading data is expected to be high values, because agent is reading data from Oracle Database.

**Output 4.3.12** *MySQL Database Server:*

```
[root@mysqldb ~]# iostat
Linux 3.10.0-229.el7.x86_64 (mysqldb)         07/05/2015         _x86_64_        (12 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,03    0,00    0,02    0,13    0,00   99,83

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda              303,77    345,99       889,05     305546      14477094
dm-0             300,44    300,57       743,94     273193      72242548
dm-1             250,00         258,01         973,00        964          5345335
dm-2             256,00         249,01         980,00       1065         24674678
dm-3            251,49        300,06        988,10      4260        144054836
```

In MySQL server there are 4 disks. It is important to have high values on both read and write data. As you see from output, writing rates are nearly 4 5 time greater than reading rates. The reason of these values is there are 6 disks on Oracle server and there are 4 disks on MySQL

disks. If we thought total read and write data are equal to each other, the values on MySQL server must be higher than Oracle Servers. Hence, our proposed system zip data on Oracle Server at first, and then unzip data on MySQL server. It is clear that writing data rate is much bigger than read data rate.

### 4.3.4 Comparison With Oracle Audit Vault Server

We compared Infofence and our proposed system with AV in terms of some features:

Table4.2: Our Proposed System Comparison with Audit Vault Server

| | **Oracle Audit Vault Server** | **Infofence + Proposed System** |
|---|---|---|
| Auditing Users and Their Objects | ✔ | ✔ |
| Using Existence of Centralized Server | ✔ | ✔ |
| Has an Easy Centralized Server Management | ✖ | ✔ |
| Database Dependency on Centralized Server | Oracle Database Dependent | Independent on Databases |
| Defining Alarms on Actions | ✔ | ✔ |
| Easy to Customize Design and Functions | ✖ | ✔ |
| Generating Reports | ✔ | ✔ |
| Has an Infrastructure Cost and Licence Price | Costly | No License Price |
| Logging Source Code of Stored Procedures | ✔ | ✔ |
| Performance Cost on Production Database | Waits on Database | No Effect |
| Real-time Database Monitoring | ✔ | ✔ |
| Secure Log Transferring | ✔ | ✔ |
| Supporting Multiple Languages | ✔ | ✔ |
| Supporting Multiple Databases | ✔ | ✔ |

Infofence is run only at Oracle databases, so the best product for comparing our proposed system is AV. On the other hand, IBM has top scores on DAM comparison conducted by Forrester Research, the best product runs on Oracle Databases is Oracle Audit Vault. Therefore, we compare our product with AV on some basis areas.

- Both AV and our systems support auditing tables and users. As seen in Figure 4.5,

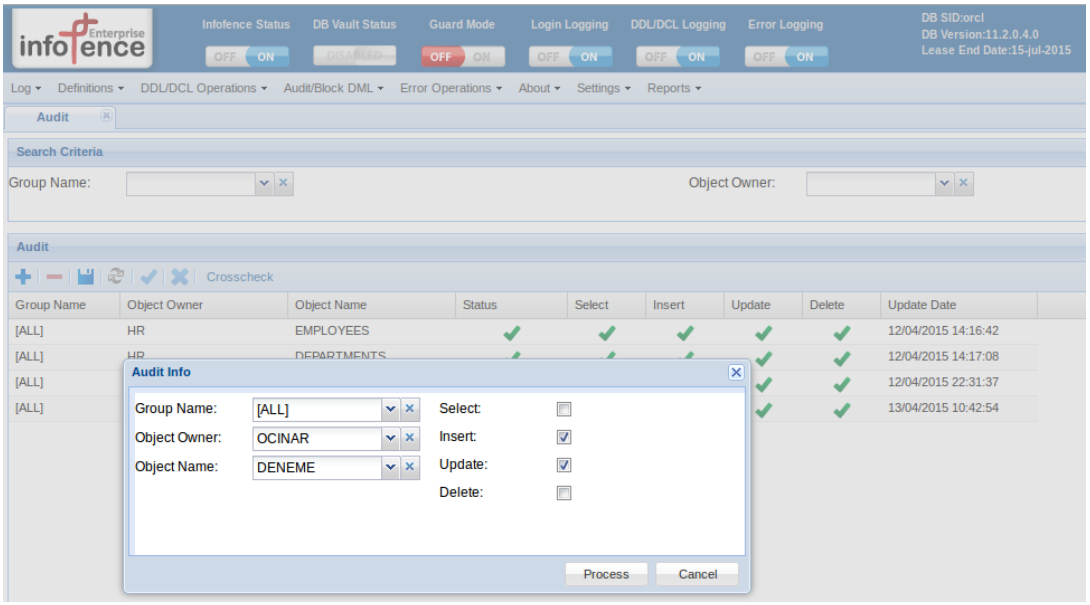



Figure 4.5: Infofence - Add Audit Window

we can both audit users and objects by using Infofence. By auditing, all defined user actions are logging.

- By using our proposed system, Infofence has a centralized server for gathering all audit logs into a main server.

- AV limited operating systems for making important changes. As seen in Figure 4.7, AV console enable users to change only network IP and interface. If you want to increase storage size, you had to add disks as raw devices and increase disks by using ASM commands. It is a very risky and High-Level DBA jobs. On the other hand, it is very easy to manage our proposed system's centralized server.

- AV is dependent on using Oracle databases on centralized server. On the other hand, in our proposed system is independent on databases. We used MySQL database. We can use another database like SQL SERVER, IBM2, and MongoDB.

- In both AV and our proposed system, we can define alarms on actions. For example, you can define an action for unauthorized users which tries to query important data.

- Oracle Company is one of the most successful companies in the world and AV is an enterprise generic product. Oracle does not customize AV and its features for some

customers. In contrast to Oracle, we can customize our product according to customers, special requests.

- End users can generate reports from both AV and our proposed system. In Infofence, you can take any reports from audit logs like, login reports, error reports, and audit log reports. Similarly, in AV, by using All Activity window, any reports can be generated with filters.

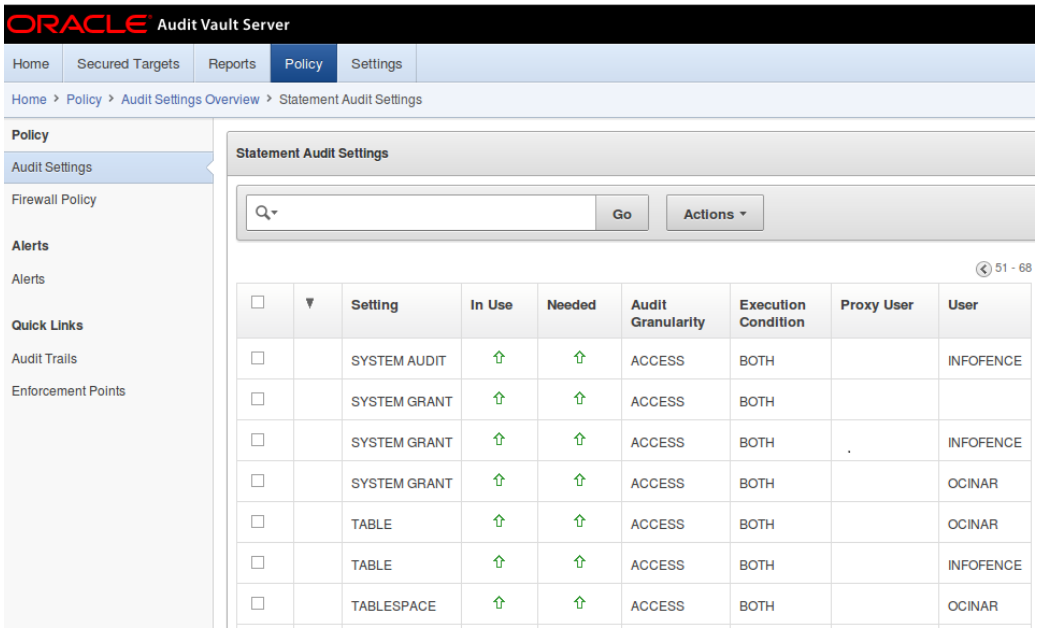



Figure 4.6: Oracle Audit Vault - Audit Settings

- Audit Vault has very big price according to database size and CPU. Whereas using Oracle database at management server causes extra license for the proposed system and Audit Vault costs too much in terms of database size and CPU, Infofence uses free-licensed database, MySQL.

- Most of DAM product does not support logging source code of stored procedures due to the fact that these products run at operating system level and listening only network logs. AV server can log any data from Oracle databases. On the other hand, Infofence runs at database level and can log any data from Oracle databases by using system triggers.

- We have tested AV with 10.2 version. According to Oracle company support site, there is a bug that cause a bad performance of database, whereas our proposed system has only minimal effects on production databases which can be ignored.

- All of the DAM products can support real-time database monitoring. It is crucial that database security must be provided to log all the data activities. Both Av and our proposed system support real-time monitoring and log transferring.

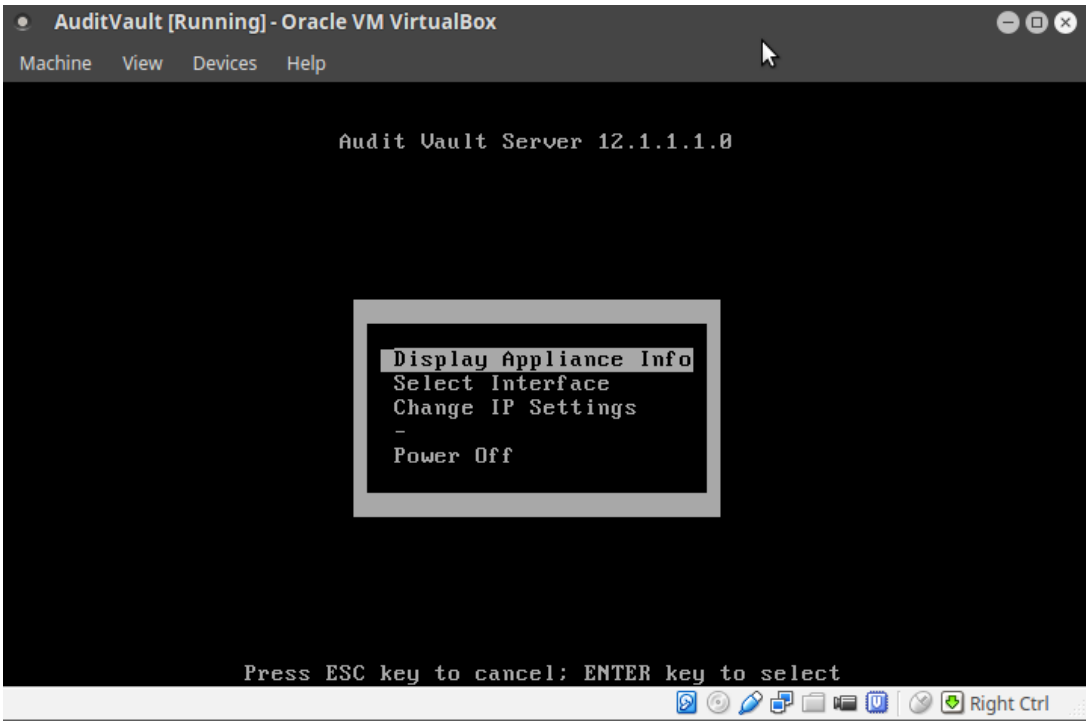



Figure 4.7: Oracle Audit Vault Console

- During transferring audit logs, we encrypt them with AES 256bit encryption. If malicious users try to capture network data, they will only get encrypted and zipped data which mean nothing to them. On the other hand, Audit vault has its own encryption policy. In its white data sheet, they reclaim that they are provide secure log transferring.

- Both our product and Oracle Audit Vault server support multiple databases and languages. In Infofence, all alerts and reports can be displayed by multi-languages. Also by using Infofence control panel, new languages can be added to system.

## 4.4 Test Environment

We developed and tested our proposed system on installed virtual machines. However, this system will be used on large-scale enterprise systems. Therefore, we installed our proposed system on two real physical servers which properties are same. On Table 4.3, there is FUJITSU Server PRIMERGY RX300 S8 technical details. We installed Oracle Database Server and MySQL Database server separately on these different servers.

Table4.3: Fujitsu Server Technical Details

| | |
|---|---|
| **Chipset** | Intel C600 (Patsburg A) |
| **Mainboard Type** | D2939 |
| **Product Type** | Dual Socket Rack Server |
| **Processor** | Intel Xeon processor E5-2600 v2 product family |
| **Memory** | 16 GB, DIMM (DDR3) |
| **Memory Protection** | Advanced ECC, Memory Scrubbing, SDDC (Chip), Rank sparing memory support, Memory Mirroring support |
| **Storage Drives** | HDD SAS, 6 Gb/s, 300 GB, 15,000 rpm, hot-plug, 3.5-inch, enterprise<br>HDD SAS, 6 Gb/s, 300 GB, 15,000 rpm, hot-plug, 3.5-inch, enterprise<br>HDD SAS, 6 Gb/s, 300 GB, 15,000 rpm, hot-plug, 3.5-inch, enterprise<br>HDD SAS, 6 Gb/s, 300 GB, 15,000 rpm, hot-plug, 3.5-inch, enterprise |
| **Service Weblink** | http://www.fujitsu.com/fts/products/product-support-services/ |

These two servers are placed in the same room and connected to same switch on the network. The network Input/output rate between these two servers is 10MB/s.

To test our proposed system on high volumes of data, we created a sample of employees department and job tables on both Oracle and MySQL databases. What is more, we generated data from a data generator, [1]. We paid attention on not creating any constraint on tables except

[1] `https://www.mockaroo.com/`, Last access: 20.05.2015

for primary keys because of that while we insert data to these tables with unique and key constraint, it will raise error on constraint violation error. We generated these data from data generator, so there might be any duplicate keys on foreign key or unique key constraints. By not encountering these errors, we only generated primary keys for these tables which provides faster query retrieving. It is easy to generate primary key values which has an incremental value and easy to generate. We downloaded the generated data as SQL format from website.

**Output 4.4.1** *Test Data Create Scripts:*

- **Employees Table Create Script on Oracle :**

```
CREATE TABLE "HR"."EMPLOYEES"
   (        "EMPLOYEE_ID" NUMBER(6,0),
        "FIRST_NAME" VARCHAR2(20 BYTE),
        "LAST_NAME" VARCHAR2(25 BYTE) CONSTRAINT "EMP_LAST_NAME_NN" NOT NULL ENABLE,
        "EMAIL" VARCHAR2(25 BYTE) CONSTRAINT "EMP_EMAIL_NN" NOT NULL ENABLE,
        "PHONE_NUMBER" VARCHAR2(20 BYTE),
        "HIRE_DATE" DATE CONSTRAINT "EMP_HIRE_DATE_NN" NOT NULL ENABLE,
        "JOB_ID" VARCHAR2(10 BYTE) CONSTRAINT "EMP_JOB_NN" NOT NULL ENABLE,
        "SALARY" NUMBER(8,2),
        "COMMISSION_PCT" NUMBER(2,2),
        "MANAGER_ID" NUMBER(6,0),
        "DEPARTMENT_ID" NUMBER(4,0),
         CONSTRAINT "EMP_EMP_ID_PK" PRIMARY KEY ("EMPLOYEE_ID"));
```

- **Departments Table Create Script on Oracle :**

```
CREATE TABLE "HR"."DEPARTMENTS"
   (        "DEPARTMENT_ID" NUMBER(4,0),
        "DEPARTMENT_NAME" VARCHAR2(30 BYTE) CONSTRAINT "DEPT_NAME_NN" NOT NULL ENABLE,
        "MANAGER_ID" NUMBER(6,0),
        "LOCATION_ID" NUMBER(4,0),
         CONSTRAINT "DEPT_ID_PK" PRIMARY KEY ("DEPARTMENT_ID"));
```

- **Jobs Table Create Script on Oracle :**

```
CREATE TABLE "HR"."JOBS"
   (        "JOB_ID" VARCHAR2(10 BYTE),
        "JOB_TITLE" VARCHAR2(35 BYTE) CONSTRAINT "JOB_TITLE_NN" NOT NULL ENABLE,
        "MIN_SALARY" NUMBER(6,0),
        "MAX_SALARY" NUMBER(6,0),
         CONSTRAINT "JOB_ID_PK" PRIMARY KEY ("JOB_ID"));
```

After created tables on both Oracle and MySQL Databases, we inserted generated data to only Oracle Database by using SQL DEVELOPER product. Insert scripts have already been generated, so we just run the commands on the Output 4.4.2 for inserting data.

**Output 4.4.2** *Test Data Insert Scripts:*

- **Sample insert scripts for Employees table:**

```
insert into EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, commission_pct, manager_id, department_id)
    values (1, 'William', 'Day', 'wday0@google.ca', '6-(576)737-4357',
 '4/16/2013', 85883815, 5416, 54, 43028787, 73749559);

insert into EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, commission_pct, manager_id, department_id)
     values (2, 'Dennis', 'Gardner', 'dgardner1@phpbb.com', '7-(655)694-4282',
'3/30/2015', 39368150, 8512, 32, 85671480, 84644877);

insert into EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, commission_pct, manager_id, department_id)
     values (3, 'Ronald', 'Garcia', 'rgarcia2@goodreads.com', '7-(151)899-4438',
'7/24/2013', 89238693, 6031, 36, 78239615, 15674351);

insert into EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, commission_pct, manager_id, department_id)
     values (4, 'Antonio', 'Johnston', 'ajohnston3@chron.com', '9-(315)548-0433',
 '8/13/2013', 29939036, 4135, 62, 81486634, 84231255);

insert into EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, commission_pct, manager_id, department_id)
     values (5, 'Jon', 'Smith', 'johns@smith.com', '5-(215)948-0433',
'5/25/2011', 1088616, 7541, 27, 1490895, 270207);
```

- **Sample insert scripts for Departments table:**

```
insert into DEPARTMENTS (department_id, department_name, manager_id, location_id)
values (1, 'Recruiter', 48790266, 25910333);

insert into DEPARTMENTS (department_id, department_name, manager_id, location_id)
 values (2, 'Data Coordiator', 18522041, 52588905);

insert into DEPARTMENTS (department_id, department_name, manager_id, location_id)
values (3, 'Geological Engineer', 13113871, 607297);
```

```
insert into DEPARTMENTS (department_id, department_name, manager_id, location_id)
values (4, 'Clinical Specialist', 80525819, 6527386);

insert into DEPARTMENTS (department_id, department_name, manager_id, location_id)
values (5, 'Analog Circuit Design manager', 13006699, 89713880);
```

- **Sample insert scripts for Jobs table:**

```
insert into JOBS (job_id, job_title, min_salary, max_salary)
values (1, 'Senior Financial Analyst', 3415, 6625);

insert into JOBS (job_id, job_title, min_salary, max_salary)
values (2, 'Web Developer IV', 1120, 5270);

insert into JOBS (job_id, job_title, min_salary, max_salary)
values (3, 'Help Desk Technician', 4603, 5823);

insert into JOBS (job_id, job_title, min_salary, max_salary)
values (4, 'Accounting Assistant IV', 1086, 7011);

insert into JOBS (job_id, job_title, min_salary, max_salary)
values (5, 'Cost Accountant', 1785, 7214);
```

We generated one hundred millions records for each table. That is, we generated distinct values for only ID columns from data generator [2] for each table. Because of using a data generator, there can be many duplicate records; however, primary columns must have distinct values. After populating tables, we query counts of tables and as you see from Output 4.4.3, total count of Employees table is:

**Output 4.4.3** *Employees Table Count:*

```
select count(1) from hr.employees;

COUNT(1)
----------
100000000
```

[2] https://www.mockaroo.com/

**Output 4.4.4** *Departments Table Count:*

```
select count(1) from HR.DEPARTMENTS;

COUNT(1)
----------
100000000
```

**Output 4.4.5** *Jobs Table Count:*

```
select count(1) from HR.JOBS;

COUNT(1)
----------
100000000
```

Average record length of each table is nearly 100 bytes. We populated each table with 100000000 records. Thus generated size of total data is nearly 9.5 GB. Select statement with this much records cost very high resources on Oracle database. We run our first agent with 4 threads. At the same time, 4 different select queries run on the Oracle databases which retrieve data respectively from Employees, Departments, and Jobs tables. Our first agent transferred all data to other server in nearly 15 minutes. We run second agent on MySQL database after one minute from first agent. Second agent parsed all encrypted data to specified tables in nearly 120 minutes. This result shows us, our proposed system transferred three hundred millions records and 9.5 GB audit logs and insert into MySQL tables in total 120 minutes. On our test environment, our data transfer rate was 1.1 MB/s. While running our proposed system, there were no other running processes on the Fujitsu servers. We do not want to use high memory, CPU, and network bandwidth of server. These servers might be use for production systems which service many users at the same time. Allocating high RAM, CPU, and network bandwidth for our system will cause waits for other users and applications. Hence, transferring data with 1.1MB/s is an acceptable situation for us. More importantly, it is clear that our proposed system runs continuously and as a result of this, there might not be huge volumes of untransferred data on Oracle Servers.

## 4.5 Limitations

The limitations that affected the performance of our experiments can be listed as follows:

- Huge volumes of data is affecting all DAM products' performance. While agents are running, they retrieves data from Oracle databases. Every RDBMS has a query optimizer, its response time depends on data sizes. When size of data is increased, because of optimizer's slowness, run time of our proposed system is increased more than expected duration, that's why, to get huge data from Oracle database is slowed down by the data itself.

- Other running processes on databases servers might affect our proposed system performance. All processes on a system uses CPU and RAM resources, and the system shares remaining resources to our agents. If there is not enough RAM and CPU on the system, our proposed system will run slowly. As a result of this, we designed our system to run with minimal resources.

- Network connection between Oracle and MySQL servers are of vital importance. By using control panel, we can increase thread counts and maximum row count to be transferred in an encrypted data. By default these values are 4 threads and 2000 rows. If we increase these values, it will use more bandwidth, RAM, and CPU of the servers. Hence, the more bandwidth there are between servers, the faster our proposed system runs.

- Every programs run slowly because of a bottleneck. If your designed your program for running with high resource, it will slow on the least resource of your system. If servers have high CPU, RAM, and bandwidth, it might be slow because of slowness of storage disks. Disk read write speed in servers might affect performance of our system. Our second agents write data to tables in centralized server. System runs faster on Solid State Drive (SSD) disks rather than normal disks, SATA, and SAS. Average reading and writing speed of SSD disks are nearly 200MB/s and 5 times faster than SATA disks. It enables the system to write data into MySQL tables in shorter period of time.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

In this study, we have presented ensuring multiple database security in private cloud computing, and managing high volumes of log data. Moreover, we have supported a 100% domestic database activity monitoring tool, Infofence on transferring logs to centralized server. Infofence is working on only Oracle databases, and using another Oracle database in centralized server will increase the cost because of Oracle's high prices. Hence, in this system we used MySQL database in centralized server is used in this system for managing database security and generating reports of audit logs. Oracle Company has already a DAM product, called Audit Vault Server. We designed our system based on Audit Vault working principle. Audit Vault server has some problems on its previous versions like high CPU and memory cost, getting no reports from huge volumes of data, and stability. Our recently proposed system fixed the problems related to previous versions of the audit vault server. However, Oracle Company has released new versions of Audit Vault server and fixed almost every issues and bugs. Our proposed model supports as many as database when there are enough resources (CPU, RAM, Storage, Network Bandwidth) on a centralized server. It is not limited by a constant number of databases. The main contributions of our study are summarized as follows:

- As Dorai [9] states that, by tuning tables, indexes, and queries on Oracle Database, we got better performance of getting data from Oracle. The faster system retrieves data from Oracle, the more quickly system transfers data to centralized server.
- By using multi-threading, we maximize log shipping between servers. Our proposed system provides threads to run parallelly at the same time. System multiplies instant transmitted data with thread count and this enable system to finish transferring processes in a short time.

- By using unique IDs as primary keys on tables, tracking and managing log transferring is very easy to handle data loss and data corruption.

- We tested log transferring processes run quickly with smaller data. Hence, we decided to use zipping algorithm on log transferring. By zipping log data provides us to reduce size of log, which helps us save time for log shipping and low cost on network traffic.

- By encrypting log data with 256 Bit AES, we secure log transferring from malicious users who can want to attack network. Tcpdump command on linux servers enables users to capture data from network traffic. By not allowing anyone to get our precious data, we use 256 Bit AES encryption algorithm with a specific key. If anyone captures our network data, they cannot infer anything from these data.

- Our main contribution is to develop and provide a 100% domestic DAM product, InfoFence to compete with its opponents on the market. Previous Infofence already supports Database security and monitoring on single Oracle Databases. What we developed on Infofence is providing database security on many Oracle databases at the same time from a centralized server. Also by using MySQL database on centralized server, we have created independent database system for log transferring which has not to use Oracle Database at the centralized server.

Our proposed model is designed for multi-database environments. It is expected that it works well in all version of Oracle databases and it is scalable to larger big databases. However, analyzing the performance of the proposed model in multi databases and big data environments and assessing its scalability might be futherly investigated by future researches, and designing the proposed model with other types of RDBMS like (SQL Server, IBM DB2) might be also another future research direction.

Another future work might be encrypt log with timestamp. It will provide us evidence for a particular data have existed at a specific time. For example, a contract has been signed about the money is transferred, and the application is made. In Turkey, TUBİTAK has administered time stamp usage and it sells limited time stamp with prices like 1000 time stamps cost 141 TL. In the future, we might use the time stamp with log shipping.

# REFERENCES


[1] I. (2005a). Information technology, security techniques, code of practice for information security management. Technical report, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Geneva, 2005.

[2] P. Bedwell. Finding a new approach to siem to suit the sme environment. *Network Security*, pages 12–16, 2014.

[3] S. Bimol, K. M. Singh, Y. J. Singh, and L. P. Devi. Assessment of transaction performance under distributed real time database system. *Electrical and Electronics Engineering*, pages 649–655, 2008.

[4] R. Blum. *Linux command line and shell scripting bible*, volume 481. John Wiley & Sons, 2008.

[5] E. Burtescu. Database security-attacks and control methods. *Journal of Applied Quantitative Methods*, pages 449–454, 2009.

[6] S. Castano, M. G. Fugini, G. Martella, and P. Samarati. Database security. *ACM Press Books, Wokingham, England: Addison-Wesley,| c1995*, 1, 1995.

[7] Y. Chen and W. W. Chu. *Protection of database security via collaborative inference detection*. Springer, 2008.

[8] M. Cobb. Siem vs. dam technology: Enterprise dam implementation best practices, 2014.

[9] R. Dorai and V. Kannan. Sql injection-database attack revolution and prevention. *J. Int'l Com. L. & Tech.*, page 224, 2011.

[10] N. Eddy, editor. *Imperva Enhances SecureSphere Data Monitoring Software*, jul 2012.

[11] R. Elmasri and S. B. Navathe. *Fundamentals of database systems*. Pearson, 2014.

[12] C. Fiorillo. *Oracle Database 11gR2 Performance Tuning Cookbook: Over 80 Recipes to Help Beginners Achieve Better Performance from Oracle Database Applications*. Packt Publishing Ltd, 2012.

[13] A. Juels and A. Oprea. New approaches to security and availability for cloud data. *Communications of the ACM*, pages 64–73, 2013.

[14] A. Lane. Database activity monitoring keeps watch over your data, 2008.

[15] R. Lemos. Cyberattacks constantly hit web apps hard fast imperva study. *eWeek*, 2012.

[16] E. Messmer. Ibm makes security push with cloud services, products aimed at mobile and big data; analysts say ibm basing much of its advances on acquisitions worklight, bigfix, guardium, q1 labs. *Network World*, 2012.

[17] R. Montesino, S. Fenz, and W. Baluja. Siem-based framework for security controls automation. *Information Management & Computer Security*, pages 248–263, 2012.

[18] A. P. P. E. Nicolett, Mark; Litan. Pattern discovery with security monitoring and fraud detection technologies. Technical report, Gartner Research, 2009.

[19] M. Nicolett. Siem market overview. Technical report, Gartner Research, 2005.

[20] B. Prince. Oracle makes bid to streamline data auditing. *eWeek*, 2007.

[21] B. Prince. Ibm confirms acquisition of database security vendor guardium. *eWeek*, 2009.

[22] B. Prince. Sentrigo adds database security vulnerability assessment. *eWeek*, 2010.

[23] R. Ramakrishnan and J. Gehrke. Database management systems. 2000.

[24] K. Ren, C. Wang, and Q. Wang. Security challenges for the public cloud. *IEEE Internet Computing*, pages 69–73, 2012.

[25] M. Rouse. Database activity monitoring (dam), 2009.

[26] P. Stephenson. For it security professionals. *SC Magazine*, pages 28–28, jul 2013.

[27] D. Tamir. Anatomy of a database attack. Technical report, Imperva, 2012.

[28] T. Vanhorn. Activity monitoring and database security. *Network World*, pages 26–26, 2007.

[29] O. Wenge, U. Lampe, C. Rensing, and R. Steinmetz. Security information and event monitoring as a service: A survey on current concerns and solutions. *PIK-Praxis der Informationsverarbeitung und Kommunikation*, pages 163–170, 2014.

[30] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso. Database replication techniques: A three parameter classification. In *Reliable Distributed Systems, 2000. SRDS-2000. Proceedings The 19th IEEE Symposium on*, pages 206–215. IEEE, 2000.

[31] A. Yadav, Arun K.; Agarwal. A distributed architecture for transactions synchronization in distributed database systems. *International Journal on Computer Science and Engineering (IJCSE)*, pages 1985–1991, 2010.

[32] N. Yuhanna. Database auditing and real-time protection. Technical report, The Forrester Research, 2011.

# APPENDIX A

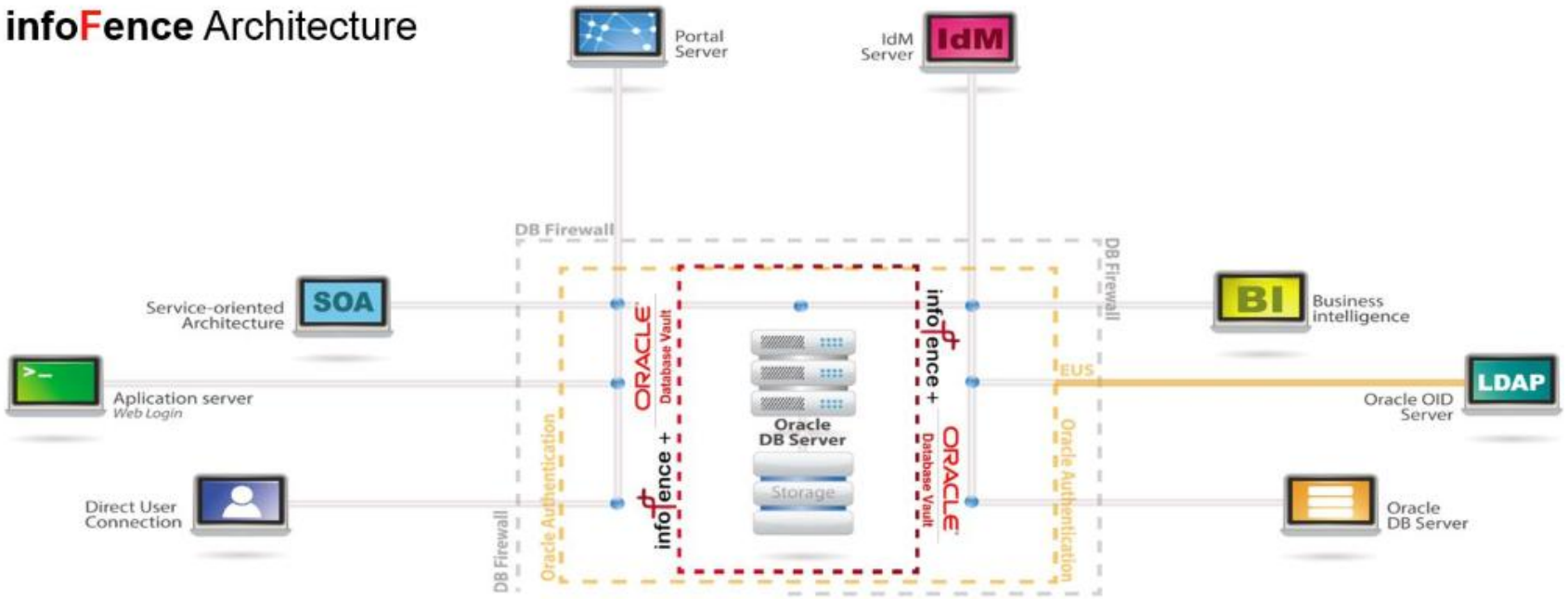
# INFOFENCE LOG TABLES

- DB_ALARM_ACTION_TAB
- DB_APPLICATION_ERROR_LOG
- DB_APPLICATION_LOGIN_LOG
- DB_CONTROL_PANEL_LOG
- DB_CONTROL_PANEL_USER_TAB
- DB_DDL_ENCRYPTED_LOG
- DB_DDL_OPERATION_LOG
- DB_DDL_PROTECT_TAB
- DB_DEBUG_ERROR
- DB_DEFAULT_VALUES_TAB
- DB_DML_QRY_BLOCKED_LOG
- DB_ERROR_MESSAGES_NLS_TAB
- DB_FGA_EXT_LOG
- DB_FILTER_APP_ERROR_TAB
- DB_GROUP_BLOCK_DML_QRY_TAB
- DB_GROUP_FGA_TAB
- DB_GROUP_MASK_QRY_TAB
- DB_GROUP_NODE_TAB
- DB_GROUP_PERM_TAB
- DB_GROUP_TAB
- DB_HIST_ALARM_ACTION_TAB_LOG

- DB_HIST_DDL_PROTECT_TAB_LOG
- DB_HIST_DEFAULT_VALUES_TAB_LOG
- DB_HIST_FLTR_APP_ERROR_TAB_LOG
- DB_HIST_GROUP_FGA_TAB_LOG
- DB_HIST_GROUP_MASK_QRY_TAB_LOG
- DB_HIST_GROUP_NODE_TAB_LOG
- DB_HIST_GROUP_PERM_TAB_LOG
- DB_HIST_GROUP_TAB_LOG
- DB_HIST_GRP_BLK_DMLQRY_TAB_LOG
- DB_HIST_NETWORK_TAB_LOG
- DB_HIST_NODE_TAB_LOG
- DB_HIST_ORA_ROLE_PRIVS_LOG
- DB_HIST_ROLE_OBJ_PERM_TAB_LOG
- DB_HIST_ROLE_OPERATION_TAB_LOG
- DB_HIST_ROLE_TAB_LOG
- DB_HIST_SYS_PRIVS_LOG
- DB_HIST_TAB_PRIVS_LOG
- DB_LOGIN_ENCRYPTED_LOG
- DB_LOGIN_LOG
- DB_NETWORK_TAB
- DB_NODE_TAB
- DB_ROLE_OBJ_PERM_TAB
- DB_ROLE_OPERATION_TAB

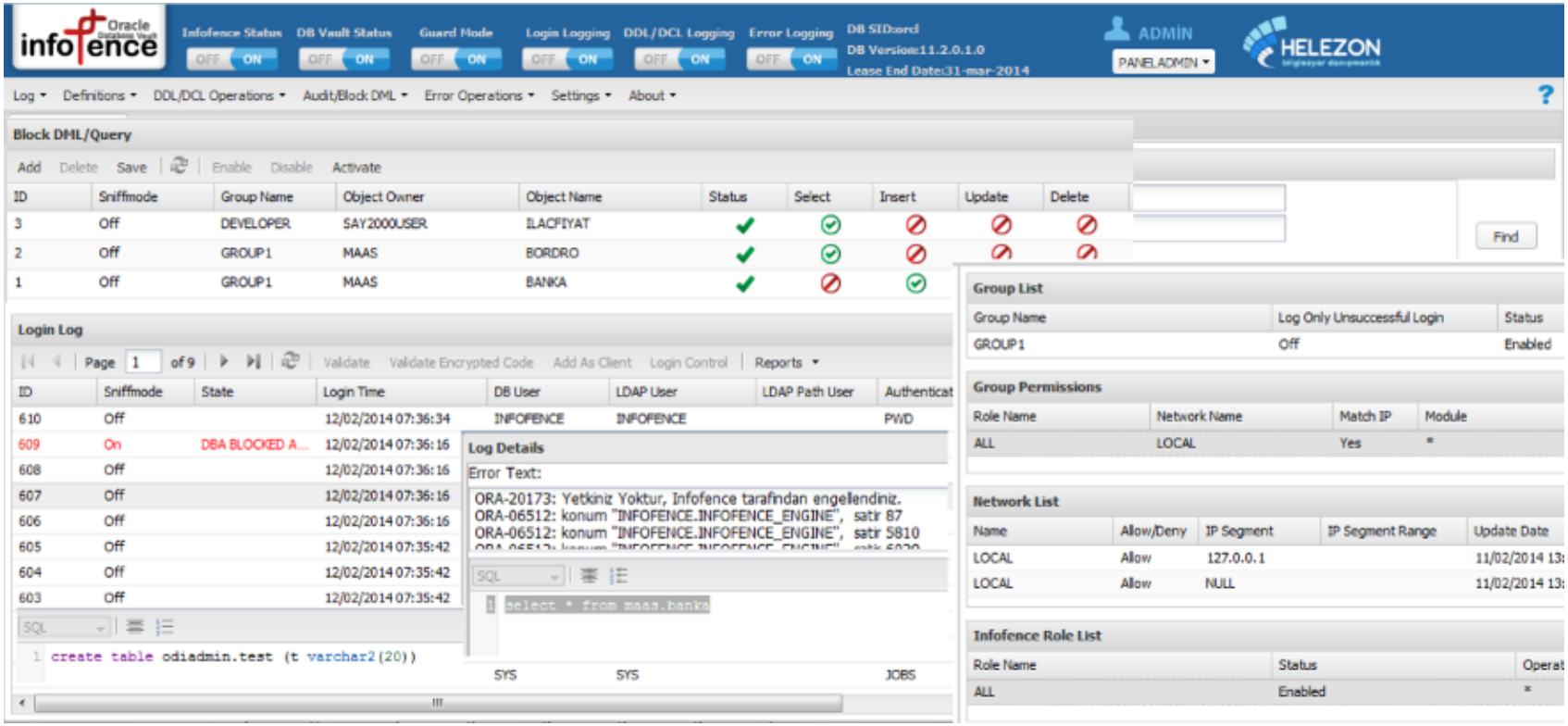
# APPENDIX B

# INFOFENCE ARCHITECTURE


infoFence Architecture
Portal Server
IdM Server
IdM
DB Firewall
Service-oriented Architecture
SOA
Aplication server
Web Login
Direct User Connection
ORACLE Database Vault
infofence +
Oracle DB Server
Storage
infofence +
ORACLE Database Vault
DB Firewall
Oracle Authentication
Oracle Authentication
DB Firewall
EUS
BI
Business intelligence
LDAP
Oracle OID Server
Oracle DB Server

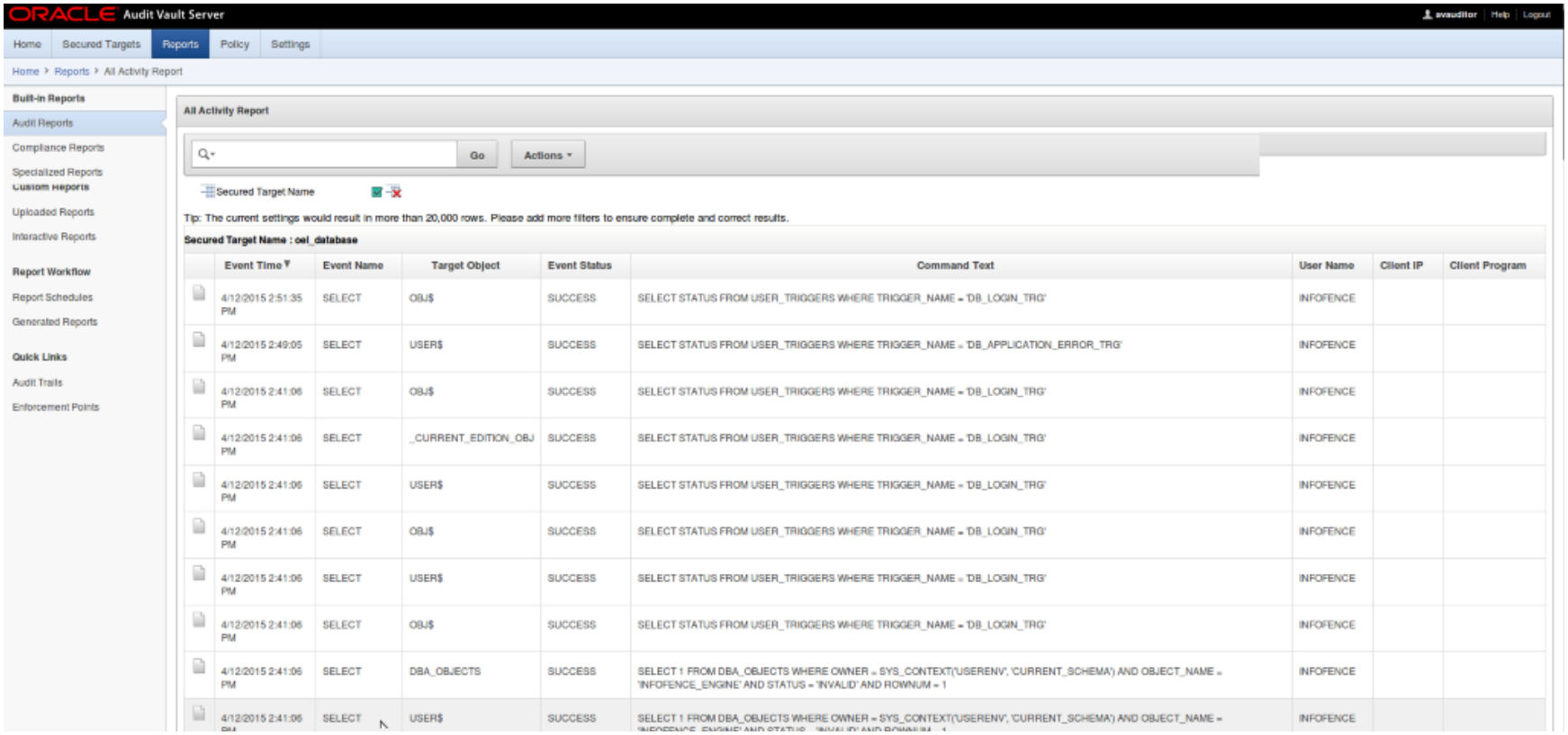
# APPENDIX C

# INFOFENCE CONTROL PANEL


infofence
Oracle Database Vault
Infofence Status
DB Vault Status
Guard Mode
Login Logging
DDL/DCL Logging
Error Logging
DB SIDcard
DB Version:11.2.0.1.0
Lease End Date:31-mar-2014
ADMİN
PANELADMIN
HELEZON
Log
Definitions
DDL/DCL Operations
Audit/Block DML
Error Operations
Settings
About
Block DML/Query
Add
Delete
Save
Enable
Disable
Activate
ID
Sniffmode
Group Name
Object Owner
Object Name
Status
Select
Insert
Update
Delete
3 Off DEVELOPER SAY2000USER ILACFIYAT
2 Off GROUP1 MAAS BORDRO
1 Off GROUP1 MAAS BANKA
Find
Login Log
Page 1 of 9
Validate
Validate Encrypted Code
Add As Client
Login Control
Reports
ID
Sniffmode
State
Login Time
DB User
LDAP User
LDAP Path User
Authenticat
610 Off 12/02/2014 07:36:34 INFOFENCE INFOFENCE PWD
609 On DBA BLOCKED A... 12/02/2014 07:36:16
608 Off 12/02/2014 07:36:16
607 Off 12/02/2014 07:36:16
606 Off 12/02/2014 07:36:16
605 Off 12/02/2014 07:35:42
604 Off 12/02/2014 07:35:42
603 Off 12/02/2014 07:35:42
Log Details
Error Text:
ORA-20173: Yetkiniz Yoktur, Infofence tarafından engellendiniz.
ORA-06512: konum "INFOFENCE.INFOFENCE_ENGINE", satir 87
ORA-06512: konum "INFOFENCE.INFOFENCE_ENGINE", satir 5810
SQL
1 select * from maas.banka
SQL
1 create table odiadmin.test (t varchar2(20))
SYS SYS JOBS
Group List
Group Name
Log Only Unsuccessful Login
Status
GROUP1 Off Enabled
Group Permissions
Role Name
Network Name
Match IP
Module
ALL LOCAL Yes *
Network List
Name
Allow/Deny
IP Segment
IP Segment Range
Update Date
LOCAL Allow 127.0.0.1 11/02/2014 13:
LOCAL Allow NULL 11/02/2014 13:
Infofence Role List
Role Name
Status
Operat
ALL Enabled *

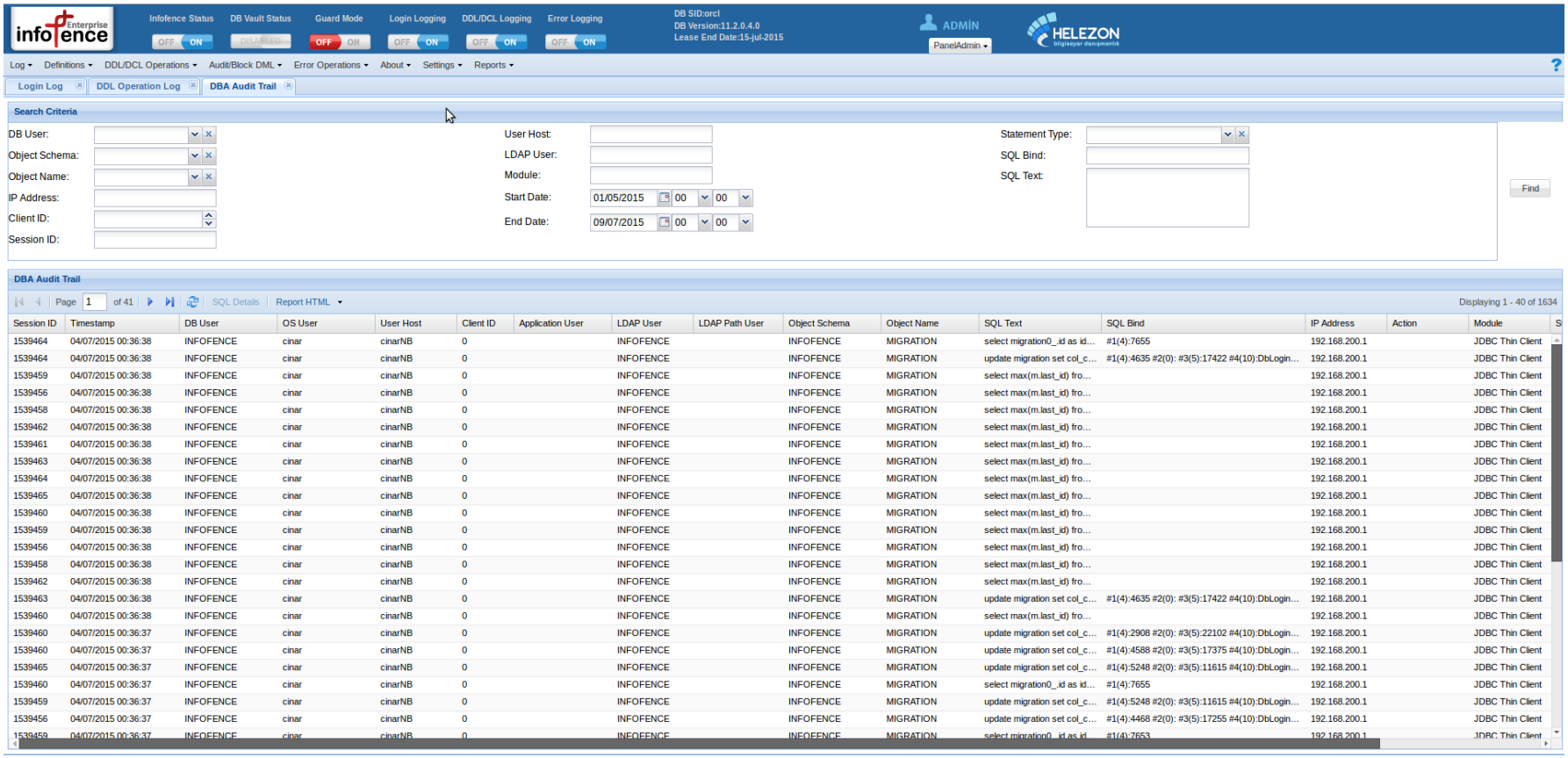
# APPENDIX D

# AUDIT VAULT ALL ACTIVITY REPORT


ORACLE Audit Vault Server
avauditor | Help | Logout
Home
Secured Targets
Reports
Policy
Settings
Home › Reports › All Activity Report
Built-in Reports
Audit Reports
Compliance Reports
Specialized Reports
Custom Reports
Uploaded Reports
Interactive Reports
Report Workflow
Report Schedules
Generated Reports
Quick Links
Audit Trails
Enforcement Points
All Activity Report
Go
Actions
Secured Target Name
Tip: The current settings would result in more than 20,000 rows. Please add more filters to ensure complete and correct results.
Secured Target Name : oel_database
Event Time
Event Name
Target Object
Event Status
Command Text
User Name
Client IP
Client Program
4/12/2015 2:51:35 PM SELECT OBJ$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:49:05 PM SELECT USER$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_APPLICATION_ERROR_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT OBJ$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT _CURRENT_EDITION_OBJ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT USER$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT OBJ$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT USER$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT OBJ$ SUCCESS SELECT STATUS FROM USER_TRIGGERS WHERE TRIGGER_NAME = 'DB_LOGIN_TRG' INFOFENCE
4/12/2015 2:41:06 PM SELECT DBA_OBJECTS SUCCESS SELECT 1 FROM DBA_OBJECTS WHERE OWNER = SYS_CONTEXT('USERENV', 'CURRENT_SCHEMA') AND OBJECT_NAME = 'INFOFENCE_ENGINE' AND STATUS = 'INVALID' AND ROWNUM = 1 INFOFENCE
4/12/2015 2:41:06 PM SELECT USER$ SUCCESS SELECT 1 FROM DBA_OBJECTS WHERE OWNER = SYS_CONTEXT('USERENV', 'CURRENT_SCHEMA') AND OBJECT_NAME = INFOFENCE

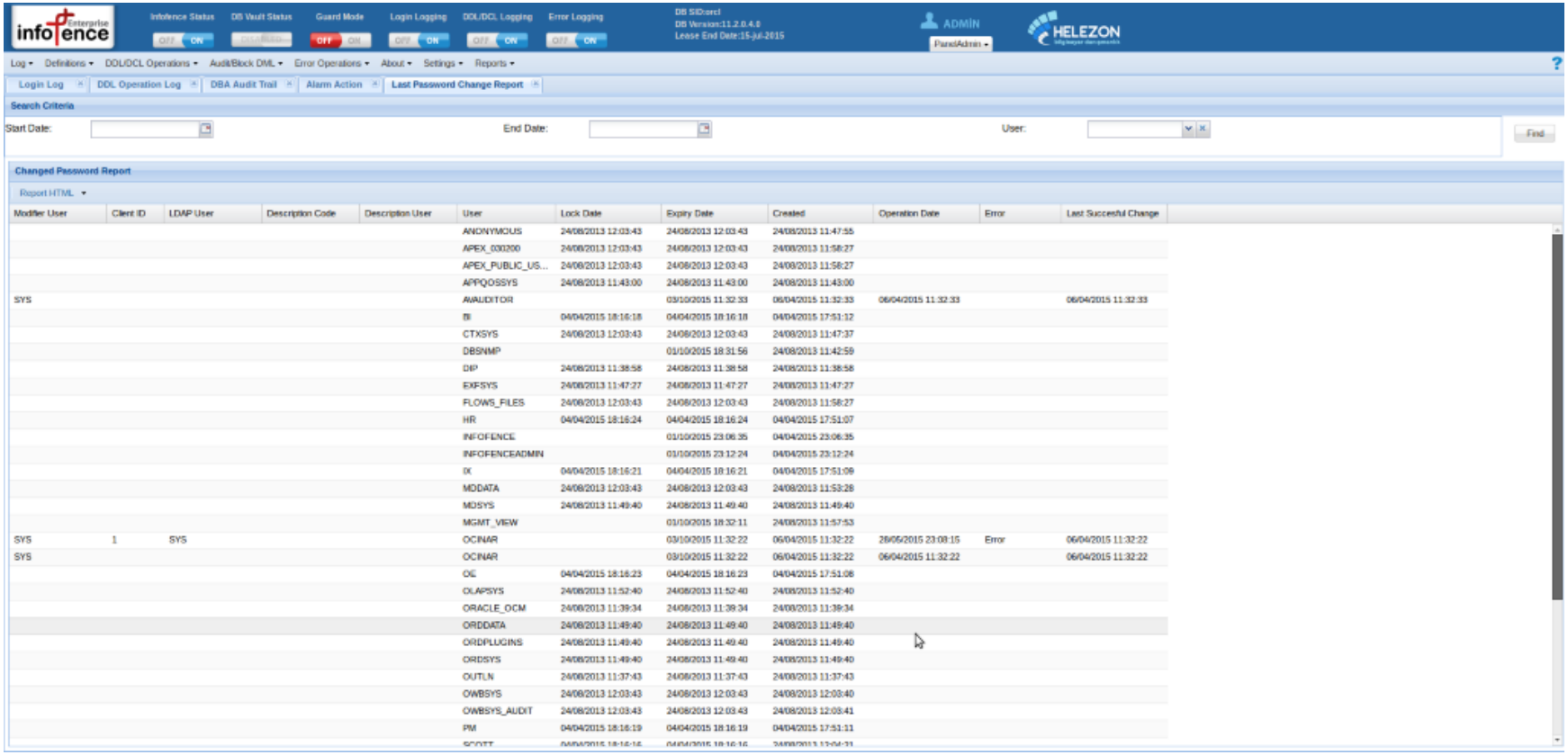
# APPENDIX E

# INFOFENCE VAULT LOG REPORT


infofence Enterprise
Infofence Status
DB Vault Status
Guard Mode
Login Logging
DDL/DCL Logging
Error Logging
DB SID:orcl
DB Version:11.2.0.4.0
Lease End Date:15-jul-2015
ADMIN
PanelAdmin
HELEZON
Log
Definitions
DDL/DCL Operations
Audit/Block DML
Error Operations
About
Settings
Reports
Login Log
DDL Operation Log
DBA Audit Trail
Search Criteria
DB User:
Object Schema:
Object Name:
IP Address:
Client ID:
Session ID:
User Host:
LDAP User:
Module:
Start Date:
01/05/2015
End Date:
09/07/2015
Statement Type:
SQL Bind:
SQL Text:
Find
DBA Audit Trail
Page 1 of 41
SQL Details
Report HTML
Displaying 1 - 40 of 1634
Session ID | Timestamp | DB User | OS User | User Host | Client ID | Application User | LDAP User | LDAP Path User | Object Schema | Object Name | SQL Text | SQL Bind | IP Address | Action | Module
1539464 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select migration0_.id as id... | #1(4):7655 | 192.168.200.1 | | JDBC Thin Client
1539464 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):4635 #2(0): #3(5):17422 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client
1539459 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539456 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539458 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539462 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539461 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539463 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539464 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539465 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539460 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539459 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539456 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539458 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539462 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539463 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):4635 #2(0): #3(5):17422 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client
1539460 | 04/07/2015 00:36:38 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select max(m.last_id) fro... | | 192.168.200.1 | | JDBC Thin Client
1539460 | 04/07/2015 00:36:37 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):2908 #2(0): #3(5):22102 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client
1539460 | 04/07/2015 00:36:37 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):4588 #2(0): #3(5):17375 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client
1539465 | 04/07/2015 00:36:37 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):5248 #2(0): #3(5):11615 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client
1539460 | 04/07/2015 00:36:37 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | select migration0_.id as id... | #1(4):7655 | 192.168.200.1 | | JDBC Thin Client
1539459 | 04/07/2015 00:36:37 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):5248 #2(0): #3(5):11615 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client
1539456 | 04/07/2015 00:36:37 | INFOFENCE | cinar | cinarNB | 0 | | INFOFENCE | | INFOFENCE | MIGRATION | update migration set col_c... | #1(4):4468 #2(0): #3(5):17255 #4(10):DbLogin... | 192.168.200.1 | | JDBC Thin Client

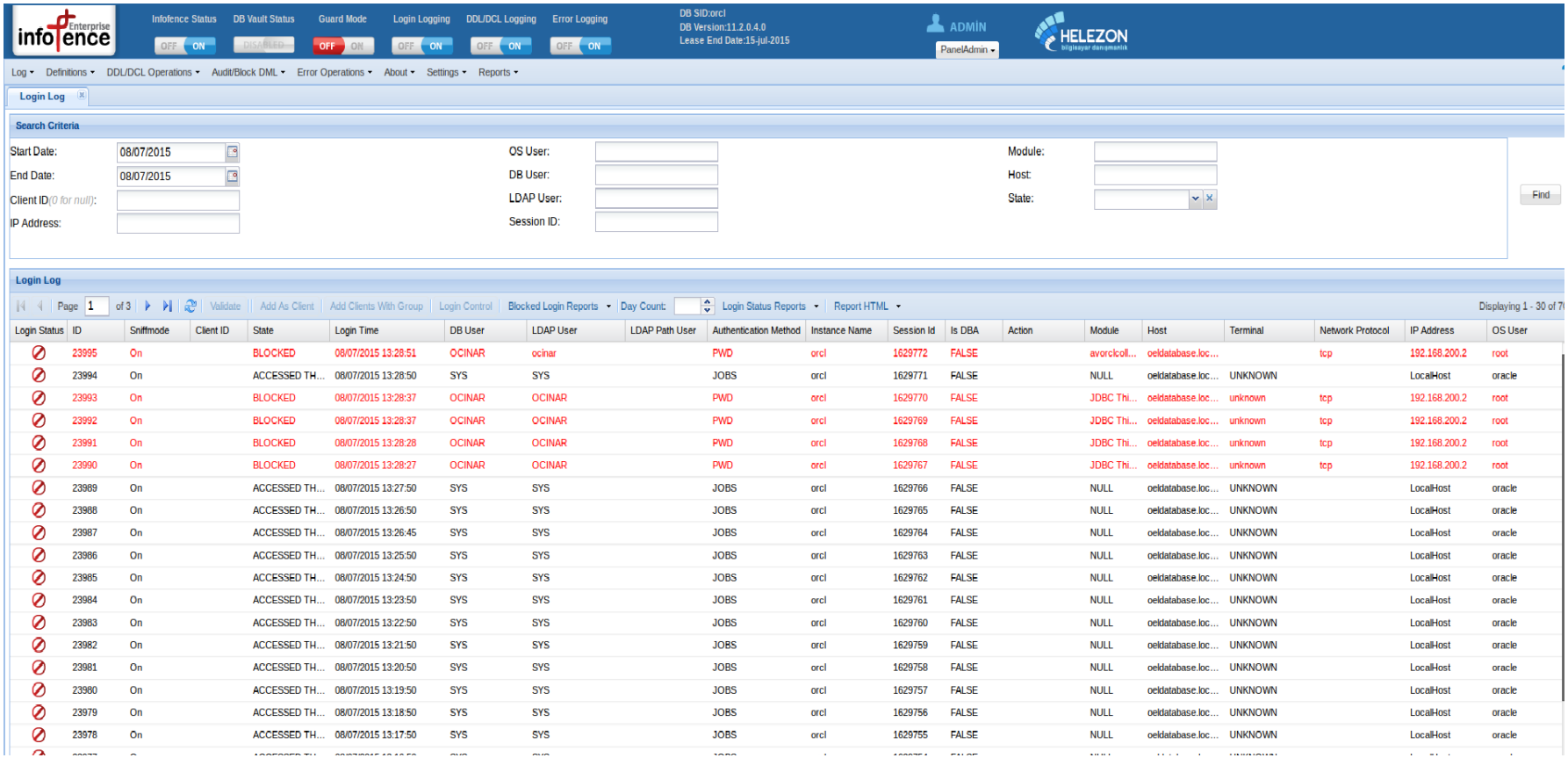
# APPENDIX F

# INFOFENCE LAST PASSWORD CHANGE REPORT


infofence Enterprise
Infofence Status
DB Vault Status
Guard Mode
Login Logging
DDL/DCL Logging
Error Logging
DB SID:orcl
DB Version:11.2.0.4.0
Lease End Date:15-jul-2015
ADMIN
ParolAdmin
HELEZON
Log
Definitions
DDL/DCL Operations
Audit/Block DML
Error Operations
About
Settings
Reports
Login Log
DDL Operation Log
DBA Audit Trail
Alarm Action
Last Password Change Report
Search Criteria
Start Date:
End Date:
User:
Find
Changed Password Report
Report HTML
Modifier User
Client ID
LDAP User
Description Code
Description User
User
Lock Date
Expiry Date
Created
Operation Date
Error
Last Succesful Change
ANONYMOUS 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 11:47:55
APEX_030200 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 11:58:27
APEX_PUBLIC_US... 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 11:58:27
APPQOSSYS 24/08/2013 11:43:00 24/08/2013 11:43:00 24/08/2013 11:43:00
SYS AVAUDITOR 03/10/2015 11:32:33 06/04/2015 11:32:33 06/04/2015 11:32:33 06/04/2015 11:32:33
BI 04/04/2015 18:16:18 04/04/2015 18:16:18 04/04/2015 17:51:12
CTXSYS 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 11:47:37
DBSNMP 01/10/2015 18:31:56 24/08/2013 11:42:59
DIP 24/08/2013 11:38:58 24/08/2013 11:38:58 24/08/2013 11:38:58
EXFSYS 24/08/2013 11:47:27 24/08/2013 11:47:27 24/08/2013 11:47:27
FLOWS_FILES 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 11:58:27
HR 04/04/2015 18:16:24 04/04/2015 18:16:24 04/04/2015 17:51:07
INFOFENCE 01/10/2015 23:06:35 04/04/2015 23:06:35
INFOFENCEADMIN 01/10/2015 23:12:24 04/04/2015 23:12:24
IX 04/04/2015 18:16:21 04/04/2015 18:16:21 04/04/2015 17:51:09
MDDATA 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 11:53:28
MDSYS 24/08/2013 11:49:40 24/08/2013 11:49:40 24/08/2013 11:49:40
MGMT_VIEW 01/10/2015 18:32:11 24/08/2013 11:57:53
SYS 1 SYS OCINAR 03/10/2015 11:32:22 06/04/2015 11:32:22 28/05/2015 23:08:15 Error 06/04/2015 11:32:22
SYS OCINAR 03/10/2015 11:32:22 06/04/2015 11:32:22 06/04/2015 11:32:22 06/04/2015 11:32:22
OE 04/04/2015 18:16:23 04/04/2015 18:16:23 04/04/2015 17:51:08
OLAPSYS 24/08/2013 11:52:40 24/08/2013 11:52:40 24/08/2013 11:52:40
ORACLE_OCM 24/08/2013 11:39:34 24/08/2013 11:39:34 24/08/2013 11:39:34
ORDDATA 24/08/2013 11:49:40 24/08/2013 11:49:40 24/08/2013 11:49:40
ORDPLUGINS 24/08/2013 11:49:40 24/08/2013 11:49:40 24/08/2013 11:49:40
ORDSYS 24/08/2013 11:49:40 24/08/2013 11:49:40 24/08/2013 11:49:40
OUTLN 24/08/2013 11:37:43 24/08/2013 11:37:43 24/08/2013 11:37:43
OWBSYS 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 12:03:40
OWBSYS_AUDIT 24/08/2013 12:03:43 24/08/2013 12:03:43 24/08/2013 12:03:41
PM 04/04/2015 18:16:19 04/04/2015 18:16:19 04/04/2015 17:51:11

# APPENDIX G

# INFOFENCE LOGIN LOG REPORT


infofence Enterprise
Infofence Status
DB Vault Status
Guard Mode
Login Logging
DDL/DCL Logging
Error Logging
DB SID:orcl
DB Version:11.2.0.4.0
Lease End Date:15-jul-2015
ADMIN
PanelAdmin
HELEZON
Log
Definitions
DDL/DCL Operations
Audit/Block DML
Error Operations
About
Settings
Reports
Login Log
Search Criteria
Start Date:
08/07/2015
End Date:
08/07/2015
Client ID(0 for null):
IP Address:
OS User:
DB User:
LDAP User:
Session ID:
Module:
Host:
State:
Find
Login Log
Page 1 of 3
Validate
Add As Client
Add Clients With Group
Login Control
Blocked Login Reports
Day Count:
Login Status Reports
Report HTML
Displaying 1 - 30 of 7
Login Status
ID
Sniffmode
Client ID
State
Login Time
DB User
LDAP User
LDAP Path User
Authentication Method
Instance Name
Session Id
Is DBA
Action
Module
Host
Terminal
Network Protocol
IP Address
OS User
23995 On BLOCKED 08/07/2015 13:28:51 OCINAR ocinar PWD orcl 1629772 FALSE avorclcoll... oeldatabase.loc... tcp 192.168.200.2 root
23994 On ACCESSED TH... 08/07/2015 13:28:50 SYS SYS JOBS orcl 1629771 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23993 On BLOCKED 08/07/2015 13:28:37 OCINAR OCINAR PWD orcl 1629770 FALSE JDBC Thi... oeldatabase.loc... unknown tcp 192.168.200.2 root
23992 On BLOCKED 08/07/2015 13:28:37 OCINAR OCINAR PWD orcl 1629769 FALSE JDBC Thi... oeldatabase.loc... unknown tcp 192.168.200.2 root
23991 On BLOCKED 08/07/2015 13:28:28 OCINAR OCINAR PWD orcl 1629768 FALSE JDBC Thi... oeldatabase.loc... unknown tcp 192.168.200.2 root
23990 On BLOCKED 08/07/2015 13:28:27 OCINAR OCINAR PWD orcl 1629767 FALSE JDBC Thi... oeldatabase.loc... unknown tcp 192.168.200.2 root
23989 On ACCESSED TH... 08/07/2015 13:27:50 SYS SYS JOBS orcl 1629766 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23988 On ACCESSED TH... 08/07/2015 13:26:50 SYS SYS JOBS orcl 1629765 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23987 On ACCESSED TH... 08/07/2015 13:26:45 SYS SYS JOBS orcl 1629764 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23986 On ACCESSED TH... 08/07/2015 13:25:50 SYS SYS JOBS orcl 1629763 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23985 On ACCESSED TH... 08/07/2015 13:24:50 SYS SYS JOBS orcl 1629762 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23984 On ACCESSED TH... 08/07/2015 13:23:50 SYS SYS JOBS orcl 1629761 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23983 On ACCESSED TH... 08/07/2015 13:22:50 SYS SYS JOBS orcl 1629760 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23982 On ACCESSED TH... 08/07/2015 13:21:50 SYS SYS JOBS orcl 1629759 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23981 On ACCESSED TH... 08/07/2015 13:20:50 SYS SYS JOBS orcl 1629758 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23980 On ACCESSED TH... 08/07/2015 13:19:50 SYS SYS JOBS orcl 1629757 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23979 On ACCESSED TH... 08/07/2015 13:18:50 SYS SYS JOBS orcl 1629756 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle
23978 On ACCESSED TH... 08/07/2015 13:17:50 SYS SYS JOBS orcl 1629755 FALSE NULL oeldatabase.loc... UNKNOWN LocalHost oracle