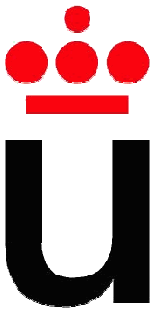


Universidad Rey Juan Carlos

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

Curso Académico 2009/2010

Proyecto de Fin de Carrera

ESTUDIO E IMPLANTACIÓN DE AUDIT VAULT

Autora: Jenifer Benayas Montemayor

Tutores: José María Cavero

Ana María García

A mi familia,
por el gran apoyo que ha sido siempre para mí.
A José María Cavero y Ana María García,
que me han ayudado y animado para que este día
se haya convertido en una realidad.

CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

Resumen

Uno de los desafíos de seguridad más importantes a los que se enfrentan las empresas hoy en día con respecto a sus BBDD (Bases de Datos) es mitigar el riesgo relacionado con las amenazas internas, pudiendo éstas ser causadas por los administradores de las bases de datos (DataBase Administrators, DBAs), ya que poseen un gran número de privilegios y externas, como pueden ser por ejemplo los hackers, etc.

La combinación de la presión de la opinión pública (debido al actual almacenamiento masivo de sus datos personales en sistemas informatizados) y las normas legales (LOPD[1], LSSICE[2], etc.) obligan a la mayoría de las empresas a proteger y a auditar sus bases de datos más importantes.

Esto conlleva que cada vez haya un mayor número de organizaciones que consideran que la información y la tecnología asociada a dicha información, representa uno de sus activos más importantes. Por lo tanto los requerimientos de calidad, control, seguridad e información, al igual que se exigen para otros activos de la empresa, se han convertido en indispensables para este ámbito. La comprobación de la aplicación de los mencionados controles es tarea de la auditoría informática.

El presente proyecto se centra en la realización de un estudio a partir de la implantación del producto Audit Vault, en un sistema de bases de datos Oracle, para analizar si realmente es rentable su instalación con respecto al consumo de recursos, personal necesario, etc. que requieren las BBDD y si además en la recopilación de la información realmente se obtienen unos beneficios considerables con respecto a las formas tradicionales de captura de información de auditoría.

[1] LOPD, Ley Orgánica 15/99 de Regulación del tratamiento Automatizado de los Datos de carácter personal (DCP) permite cumplir el mandato constitucional del artículo 18.4 que "limita el uso de la informática para garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos" [De Pablos, C..,2006]

[2] LSSICE, La ley de los Servicios de la Sociedad de la Información y Comercio Electrónico, entró en vigor 12/10/02 (impone sanciones mayores que la LOPD). [De Pablos, C.., 2006]

Capítulo 1 Introducción

1.1 Motivación

En la actualidad, las organizaciones poseen gran cantidad de BBDD que contienen información de alto valor. No solamente guardan datos sensibles para la compañía, como pueden ser registros financieros o cálculos de nómina de los empleados, sino que además almacenan, entre otros, datos confidenciales de sus propios clientes como pueden ser códigos de tarjetas de crédito, identificaciones fiscales, etc.

Centrándose en el ámbito de las universidades españolas, específicamente en la Universidad Rey Juan Carlos, tema que concierne a este proyecto aunque igualmente útil para otros sectores, los sistemas de información adquieren gran importancia debido a toda la clase de información que se recogen en sus BBDD. En ellas, aparte de encontrarse datos pertenecientes a la propia gestión de la universidad, también se pueden encontrar datos sensibles referentes tanto al alumnado, como al Personal Docente e Investigador (PDI) o al Personal de Administración y Servicios (PAS).

Por lo tanto, la creación e imposición de procedimientos de seguridad permiten proteger lo que se está convirtiendo rápidamente en el bien corporativo más importante: los datos. Aunque almacenar datos en una base de datos los hace ser más útiles y estar disponibles para toda la empresa, también los hace más vulnerables a posibles accesos no autorizados, en los cuales se realicen creaciones, modificaciones y borrados que puedan producir efectos muy perjudiciales para la organización o posibles consultas a datos altamente sensibles. Estos intentos de acceso deben detectarse y evitarse.

Esta circunstancia nos lleva a mostrar un gran interés por la auditoría informática, que conlleva la utilización de un conjunto de técnicas, actividades y procedimientos, destinados a analizar, evaluar, verificar y recomendar en asuntos relativos a la planificación, control, eficacia, seguridad y adecuación del servicio informático en la empresa, por lo que comprende un examen metódico, puntual y discontinuo del servicio informático, con vistas a mejorar tanto en rentabilidad, como en seguridad y eficacia, alcanzando de esta manera un rol cada vez más importante en las áreas de cumplimiento, privacidad y seguridad [De Pablos, C.., 2006].

Hoy en día, el uso de los datos de auditoría como recurso de seguridad continúa siendo un proceso manual que requiere que el personal de auditoría y de

seguridad de TI (Tecnologías de la Información) primero recopile estos datos, a través de sentencias de auditoría y más tarde realice investigaciones sobre una gran cantidad de datos de auditoría utilizando scripts personalizados y otros métodos, sobre las tablas, en las cuales se recogen estos datos, existentes en la propia BBDD (Auditoría tradicional). Esto se convierte en una tarea tediosa y poco intuitiva. En este momento existen nuevos productos como Oracle Audit Vault que automatizan el proceso de análisis y recopilación de la información de auditoría, convirtiendo los datos de auditoría en un recurso de seguridad clave para ayudar a abordar los actuales desafíos de cumplimiento y seguridad, facilitando su uso a través de una interfaz gráfica.

Por otro lado se debe tener en cuenta que la instalación de las aplicaciones que llevan a cabo las auditorías de BBDD, y aseguran su correcto seguimiento y funcionamiento, que es lo que se pretende en este proyecto, consumen recursos y requieren un gran esfuerzo por parte del DBA. Conforme aumenta el número de BBDD que además contienen un tipo de datos sensibles, también aumenta la carga de trabajo para garantizar su seguridad.

Y, por último, y como motivación principal, contribuir con mi esfuerzo si es posible en la mejora de la seguridad de los sistemas de información de nuestra universidad, con la instalación y posterior estudio del producto Oracle Audit Vault, un producto "muy novedoso y unos de los más completos del mercado" según Oracle, ya que, aparte de las innovaciones que presenta, es compatible con distintos tipos de BBDD. Debido a lo novedoso que es tiene un gran inconveniente, y es que la mayoría de la información que se puede adquirir acerca del uso, novedades, beneficios, etc, viene dada prácticamente en su totalidad por Oracle, por lo tanto necesita ser contrastada y evaluada, adecuándose a las características de nuestros sistemas.

Cabe destacar que la introducción de un nuevo producto como éste, en un sistema de explotación, requiere un estudio previo para determinar las características de implantación como para realizar una correcta instalación. Si además se tiene en cuenta, como ya se ha dicho, que se trata de una novedad en el mercado, la tarea se complica por la falta de información y ayuda externa, convirtiéndose en un proceso arduo, largo y complicado.

1.2 Objetivos

De acuerdo a los motivos expuestos anteriormente, los objetivos que pretende este proyecto son los siguientes:

- Instalación de un producto de auditoría (Audit Vault) de Oracle, muy novedoso y completo en un entorno de pruebas.
- Estudio comparativo de la realización de la auditoría entre la auditoría tradicional y la auditoría realizada a través de Audit Vault.
- Estudio de los beneficios que aporta la utilización de este producto con respecto a la creación de informes de auditoría (Audit Report).
- Estudio de rentabilidad de dicho producto en el consumo de recursos de un determinado sistema.
- Creación de manuales de instalación y utilización del producto, claros y concisos.

Por lo tanto, se pretende utilizar un proceso de auditoría para determinar, por medio de la investigación a través de un gran número de pruebas e implementación de los scripts que simulan y obtienen información de una máquina de pruebas asemejándola a una BBDD en explotación, la eficiencia de su implantación.

1.3 Método de trabajo

El método de trabajo que se va a seguir en este proyecto, es el siguiente:

a) Recopilación de información acerca de auditorías informáticas.
b) Estudio de la realización de la auditoría tradicional.
c) Recopilación de información del producto Audit Vault, elegido por unas especificaciones determinadas.
d) Descripción del problema.
e) Instalación del producto a partir de la arquitectura.
f) Creación del entorno de pruebas del sistema y realización de dichas pruebas.
g) Análisis de los resultados de las pruebas.

1.4 Estructura de la memoria

En este apartado se describe la estructura de la presente memoria

En el capítulo 1, **"Introducción",** en este capítulo se describe en qué se ha motivado el proyecto, presentando los objetivos y el método de trabajo seguido en su desarrollo.

En el capítulo 2, **"Estado del Arte",** comienza explicando qué es la auditoría y cómo se desarrolla este concepto desde el punto de vista informático. A continuación se centra en la realización de dicha auditoría y por último aparecen los diferentes tipos de auditoría aplicadas en Oracle, específicamente el modo de realizar la auditoría tradicional y una explicación del producto Audit Vault que se verá más en profundidad en capítulos posteriores.

En el capítulo 3, **"Instalación de Audit Vault",** describe el problema con el que nos encontramos inicialmente, que es si la implantación de un nuevo producto de auditoría bastante novedoso es rentable para nuestro sistema. En este capítulo se incluyen las instalaciones necesarias, la creación de un entorno de pruebas, la generación de actividad y el registro de la actividad.

En el capítulo 4, **"Realización de auditoría",** este capítulo describe en qué entorno se realiza la auditoría y cómo se lleva a cabo con el producto Audit Vault. Finalmente aparece la información recogida de la actividad de los distintos procesos que se han ido ejecutando y un análisis de lo que representa dicha actividad.

Por último en el capítulo 5, **"Conclusiones y trabajos futuros",** se describe la respuesta hallada a partir de las pruebas anteriores, analizando los logros alcanzados y las dificultades que han ido apareciendo a lo largo de la investigación, así como los posibles trabajos futuros que se pudieran llevar a cabo a partir de la investigación realizada.

A continuación, se concluye con una serie de anexos que contienen información acerca de la instalación del producto llevada a cabo en un entorno de pruebas. Éstos son necesarios para la posterior implantación en explotación.

Capítulo 2 Estado del arte

El objetivo de esta sección es realizar una recopilación y un análisis previo desde el significado de auditoría, hasta su aplicación en los sistemas de tecnologías de la información, como de las distintas tecnologías que son de uso en la realización de auditorías. Asimismo, analizaremos las aplicaciones existentes para auditar, en especial Audit Vault.

2.1 Concepto de auditoría

A continuación se hace una breve descripción del concepto de auditoría, desde cuál es su origen, hasta su significado aplicado a las nuevas tecnologías de la información, más concretamente a las bases de datos Oracle.

2.1.1 Auditoría informática

Inicialmente este concepto aparece por un préstamo del inglés 'to Audit' que significa examinar, revisar, o intervenir cuentas. Y éste del latín auditus, participio de audire, 'oír'. A partir de aquí surge el término de auditoría orientado especialmente a la contabilidad de las empresas o entidades.

Sin embargo la utilización cada vez más importante de las tecnologías de la información, y más concretamente de la informática, ha implicado que este término amplíe su significado a este ámbito mediante la definición de normas, procedimientos y controles para su mejor aprovechamiento.

La auditoría informática podemos entenderla como la auditoría aplicada al campo informático y auditoría se puede definir como "el examen metódico de una situación relativa a un producto, proceso u organización, realizado en cooperación con los interesados para verificar la concordancia de la realidad con lo preestablecido y la adecuación al objetivo buscado" (Norma AENOR X50-109) o "la actividad para determinar, por medio de la investigación, codificaciones, estándares, y otros requisitos, la adhesión a los mismos y la eficacia de su instrumentación" (ANSI – Asociación Americana de Normalización).

Trasladadas y ajustadas las definiciones anteriores, se puede entender por auditoría informática como "la revisión, verificación y evaluación con un conjunto de

métodos, técnicas y herramientas de los sistemas de información de una organización, de forma discontinua y a petición de su dirección y con el fin de mejorar su rentabilidad, seguridad y eficacia" [De Pablos, C., 2006].

Por lo tanto una buena auditoría informática es de vital importancia para el buen desempeño de los sistemas de información, ya que ésta proporciona los controles necesarios para que los sistemas sean confiables y con un buen nivel de seguridad. Esta auditoría comprende aspectos muy diversos de la empresa, para que esta evaluación sea lo más acertada posible. Puede recoger la evaluación de los sistemas de información en general (desde sus entradas, procedimientos, controles, archivos, seguridad y obtención de información) hasta la organización de los centros de información, hardware, software, etc. Esto implica que los mecanismos de control implantados en una empresa u organización adquieren gran importancia ya que la auditoría determina si los mismos son adecuados y cumplen unos determinados objetivos o estrategias, estableciendo los cambios que se deberían realizar para la consecución de los mismos. Los mecanismos de control pueden ser directivos, preventivos, de detección, correctivos o de recuperación ante una contingencia.

Una vez analizado el concepto, se puede determinar que los objetivos principales de la auditoría Informática son: el *control* de la función informática, el *análisis* de la *eficiencia* de los Sistemas Informáticos, la *verificación* del cumplimiento de la Normativa en este ámbito y la revisión de la eficaz *gestión* de los recursos informáticos. Y su función es mejorar ciertas características en la empresa como la *eficiencia*, la *eficacia*, la *rentabilidad* y la *seguridad.*

Una vez descrito el significado de auditoría informática se pueden determinar diferentes **tipos de auditoría informática**. Entre ellos destacan los siguientes:

- *Auditoría de la gestión***:** Referido a la contratación de bienes y servicios, documentación de los programas, etc.
- *Auditoría legal del Reglamento de Protección de Datos*: Cumplimiento legal de las medidas de seguridad exigidas por el Reglamento de desarrollo de la Ley Orgánica de Protección de Datos.
- *Auditoría de los datos*: Clasificación de los datos, estudio de las aplicaciones y análisis de los flujogramas.
- *Auditoría de las bases de datos*: Controles de acceso, de actualización, de integridad y calidad de los datos.

- *Auditoría de la seguridad:* Referidos a datos e información verificando disponibilidad, integridad, confidencialidad, autenticación y no repudio.
- *Auditoría de la seguridad física*: Referido a la ubicación de la organización, evitando ubicaciones de riesgo, y en algunos casos no revelando la situación física de esta. También está referida a las protecciones externas (arcos de seguridad, CCTV, vigilantes, etc.) y protecciones del entorno.
- *Auditoría de la seguridad lógica*: Comprende los métodos de autenticación de los sistemas de información.
- *Auditoría de las comunicaciones*. Se refiere a la auditoría de los procesos de autenticación en los sistemas de comunicación.
- *Auditoría de la seguridad en producción*: Frente a errores, accidentes y fraudes.

2.1.2 Auditoría de bases de datos

Como se cita en el apartado anterior la auditoría se puede desarrollar en diferentes áreas, de manera que este apartado se va a centrar más concretamente en la protección y seguridad de las bases de datos.

2.1.2.1 El camino hacia el cumplimiento de normas

Hasta hace poco tiempo, las organizaciones no eran muy cuidadosas a la hora de cumplir con sus obligaciones respecto a la seguridad de la información contenida en sus bases de datos. No obstante, la forma tan repentina y sorprendente en que se han empezado a poner en evidencia los resultados de las auditorías y los requisitos de cumplimiento obligatorio e ineludible a que deben hacer frente tanto las organizaciones como las personas responsables, están ocasionando problemas realmente importantes.
Dos problemas fundamentales vienen dados por la gran innovación que han supuesto las TIC (tecnologías de información y las comunicaciones) uno de ellos es la educación y el otro la comunicación. El primero es fundamental y necesario, ya que es imprescindible saber el significado de proteger y auditar una BBDD, pero por otro lado la comunicación tiene que ser efectiva, es decir, es necesario saber qué es lo que está pidiendo el auditor, ya que a veces el uso de cierta terminología utilizada en auditoría puede ser desconocida para los usuarios afectados obligando a aprender los conceptos básicos para poder realizar dicha comunicación con los auditores.

Los DBAs no sólo deben tener un conocimiento preciso de las bases de datos en uso, sino también de qué tipo de datos hay en cada una y quién accede a ellas. Cuando los DBAs se enfrentan a centenares de bases de datos, esta labor se convierte en un proceso largo, al cual se suele referir como "catalogación de activos".

Una vez terminada la contabilización de todos los activos básicos (entre otros las bases de datos, los propios datos y los usuarios), se pasa a la segunda fase. En ella se han de identificar las bases de datos con información sensible. Con su ayuda, los auditores pueden determinar qué bases de datos contienen información sujeta a medidas especiales de seguridad. Conforme aumenta el número de bases de datos de este tipo, también aumenta la carga de trabajo para garantizar su seguridad.

Mantener un estricto control de seguridad de una base de datos es una tarea complicada. Solamente puede darse acceso a aquellas personas que lo necesitan, y tiene que restringirse a aquel conjunto mínimo de objetos que cada uno necesita. Además es importante también documentar cada cambio que se haga sobre la configuración de la base de datos a partir de ese momento, y las políticas que lo garantizan, a fin de mantener a lo largo del tiempo el nivel de seguridad establecido inicialmente.

La auditoría exige la revisión de todas y cada una de las actividades producidas dentro de la base de datos y asegurarse de que se corresponden fielmente con las decisiones acordadas con los auditores. Si no se dispone de las herramientas necesarias, es simplemente imposible.

2.1.2.2 El problema de la Auditoría

El concepto de auditoría como se ha explicado en los apartados anteriores, abarca todo lo que tiene que ver con los riesgos y el control, en definitiva todo lo relacionado con la identificación de riesgos y establecimiento de controles necesarios para mitigarlos.

Está claro que la figura que posee mayor riesgo es el DBA, el administrador de base de datos. Esto es debido a que sus actividades dan cuenta de hasta un 80 % de las amenazas que afectan a las bases de datos, ya que ellos poseen todas las claves y conocen todas las puertas y ventanas de acceso a ella. Por lo tanto es muy importante que exista una herramienta de gestión de la base de datos que permita separar el rol del administrador del auditor (hasta ahora como norma general, residente en la misma persona).

Esto ha llevado a que en la actualidad el mercado de auditoría de bases de datos tenga un gran potencial económico. Por lo que muchas empresas compiten actualmente por una porción de esta tarta. Aparte de las tecnologías más tradicionales, las herramientas de monitorización del rendimiento contienen información detallada sobre la actividad de la base de datos y por eso muchas de estas herramientas se venden ahora como soluciones de auditoría por lo que conviene identificar cuál de estos nuevos productos es el más indicado para cumplir los objetivos de la organización en cuestión.

2.1.2.3 Factores elementales de una herramienta de auditoría

Hay tres cosas importantes y que se deberían tener en cuenta a la hora de elegir una herramienta de auditoría en una base de datos. Éstas son las siguientes:

- Los datos que se recopilan.
- La tecnología empleada.
- Las funcionalidades del producto.

A continuación profundizaremos en cada uno de los puntos anteriores.

- **Los datos que se recopilan**

Los datos a recabar son tan importantes porque constituyen la base del sistema. Cuando hablamos de auditoría de base de datos, el asunto más importante es **quién** toca **qué** datos y **cuándo** los toca. También es importante saber **qué** tablas se han visto afectadas y **cómo.**

Para responder al "**quién**", el producto debe ser capaz de capturar todo el contexto de sesión sin excepciones. Tanto si se trata de conexiones en entornos Oracle, o conexiones de memoria compartida (en SQL Server) o cualquier otra modalidad de conexión. La posibilidad de pasar por alto algún tipo de conexión puede convertirse en un tremendo agujero de seguridad, por lo que hay que asegurarse de que el DBA conoce todas las posibles formas de conectarse a la base de datos. La captura de conexiones fallidas a la base de datos es obviamente importante, ya que nos puede ayudar a detectar intentos de intrusión, pero se trata de algo secundario en comparación con la necesidad de capturar todas las sesiones.

Para responder a la pregunta "**qué**", el producto debe ser capaz de identificar todos los accesos a las tablas físicas y saber si el acceso ha sido para lectura o para escritura. Al final de la jornada, los datos confidenciales que requieren protección están dentro de tablas. El producto tiene que registrar todos los accesos a estas tablas, tanto si

se hace a través de vistas o de un sinónimo. Tampoco es relevante el dato de si se accede a la información desde procedimientos almacenados, a consecuencia de un trigger o desde SQL dinámico generado desde un bloque de código. Hay muchas formas de acceder a los datos y todas ellas tienen que estar registradas habida cuenta de que el DBA se las conoce.

- **La tecnología empleada**

La tecnología elegida es también esencial para una solución de auditoría. Hay que asegurarse de que la tecnología no permite que un DBA la desactive, como sucede con las herramientas de auditoría nativas. Además, es preciso asegurarse de que realmente captura todos los datos considerados críticos (conexiones de red, conexiones BEQ, SQL dinámico, procedimientos almacenados, triggers, sinónimos y vistas). Finalmente, es esencial que no se pierda ni se pase por alto ningún dato, independientemente de la duración de las transacciones (sentencias SQL cortas), nivel de carga del sistema o cualquier limitación más inusual.

A diferencia de lo que sucede cuando se inspeccionan los registros financieros, resulta imposible revisar a mano todas y cada una de las sentencias ejecutadas en la base de datos. En consecuencia, estamos a merced del motor de reglas que incorpora el producto. Este motor tiene que ser capaz de identificar las actividades sospechosas que tratamos de localizar y evitar en lo posible los falsos positivos.

La facilidad de gestión del producto es otro detalle muy importante. Existen cientos de bases de datos con distintos tipos de usuarios y datos. El producto debe gestionar de manera razonable todos estos activos con un nivel adecuado de escalabilidad. La aplicación de las reglas acordadas con los auditores a todas las bases de datos corporativas es el reto al que nos enfrentamos y resulta extremadamente importante que el producto nos deje ejecutar esta labor de una manera razonable.

- **Las funcionalidades del producto**

La facilidad de gestión del producto es otro detalle muy importante. Existen cientos de bases de datos con distintos tipos de usuarios y datos. El producto debe gestionar de manera razonable todos estos activos con un nivel adecuado de escalabilidad. La aplicación de las reglas acordadas con los auditores a todas las bases de datos corporativas es el reto al que nos enfrentamos y resulta extremadamente importante que el producto nos deje ejecutar esta labor de una manera razonable.

2.1.2.4 Otras consideraciones en la Seguridad de las Bases de Datos.

Es muy importante determinar en una BBDD qué es lo que puede afectar a su seguridad. Los siguientes aspectos están considerados como unos de los más importantes a tener en cuenta en la seguridad en los Sistemas de Información:

• Amenaza externa.

• Discos reemplazados que no son adecuadamente desechados.

• Cintas de respaldo perdidas o robadas durante el transporte.

• Amenaza interna.

• DBAs con acceso a datos de aplicaciones, incluyendo registros financieros o de R.R.H.H.

• Modificaciones no autorizadas a las aplicaciones o bases de datos.

• Conformidad normativa.

• Quién estuvo accediendo a información clasificada, cuándo, dónde, y cómo.

A partir de estos condicionantes es posible determinar qué es lo que interesa controlar y estudiar dependiendo de los aspectos que sean considerados críticos en la BBDD. Para ello se encuentra una buena definición en la siguiente cita:

"Encryption isn't "buy and forget" security, understand its limits and when it is appropriate"

-Gartner: When and How to Use Enterprise Data Encryption, [http://www.gatner.com]

2.1.2.5 Auditoría Informática según los realizadores

En función de quién realice la auditoría se puede distinguir entre:

- Auditoría Informática Interna.
- Auditoría Informática Externa.
- Auditoría Mixta.

- **Auditoría Informática Interna**

Es realizada por alguien perteneciente a la propia estructura organizativa de la empresa .

La principal ventaja es que al pertenecer a la propia empresa conoce más directamente su problemática. Por otro lado su coste será menor.

La principal inconveniente será la posible falta de objetividad de las personas que las llevan a cabo, puesto que pueden estar directamente implicados en el propio sistema de información

- **Auditoría Informática Externa**

La empresa contrata un servicio para auditar su sistema de información por personas externas a la empresa.

La principal ventaja es la objetividad que presenta la persona que realiza la auditoría, ya que ésta es ajena a la empresa.

El principal inconveniente viene dado por el desconocimiento de la problemática de la empresa. Y desde el punto de vista económico ésta resulta más costosa al no realizarse con recursos propios como la interna.

- **Auditoría Mixta**

Se trata, mediante la creación de un equipo mixto de auditores (internos y externos), de esta manera se intenta llevar a cabo el trabajo intentando evitar los problemas anteriores. Su principal inconveniente será la falta de homogeneidad de conocimientos y experiencias de quienes deben llevar a cabo dicha auditoría.

2.1.3 Auditoría Oracle

Se va a estudiar principalmente la auditoría Oracle debido a que más del 90% de las BBDD de la Universidad son BBDD Oracle. De esta manera, debido a las especificaciones dadas por el entorno en el cual se va a realizar el estudio, el producto que interesa investigar es aquel compatible con dichas BBDD. Lo que quiere decir que si son BBDD Oracle se estudiarán productos Oracle.

En el caso de las BBDD Oracle, la auditoría es un conjunto de características que permite al DBAs y a los usuarios hacer un seguimiento del uso de la BBDD. El DBAs puede definir una actividad de auditoría predeterminada. La información de las auditorías se almacena en el diccionario de datos, en la tabla SYS.AUD$ o en la pista de auditoría del sistema operativo (si lo permite). Existen varias vistas que se basan en esta tabla (SYS.AUD$) para mostrar distintos resultados, según la información que se quiera obtener.

Lo anterior viene definido en el parámetro AUDIT_TRAIL, esto quiere decir que para que los datos de auditoría se vayan almacenando en la BBDD este parámetro

debe estar activado como ya se explicará en el siguiente apartado. Cabe destacar que la BBDD Oracle tiene varias capas de seguridad y proporciona la capacidad de auditar cada nivel.

En una auditoría es importante auditar tres tipos de acciones: intentos de inicio de sesión, accesos a objetos y acciones de la base de datos. Cuando se realizan auditorías, la funcionalidad de la BBDD es dejar constancia de los comandos correctos e incorrectos que se realizan sobre ésta. Esto puede modificarse cuando se configura cada tipo de auditoría.

Por ejemplo, se pueden registrar todos los intentos de actualizar los datos de una tabla o sólo los intentos fallidos, también se pueden registrar todos los inicios de sesión en Oracle o sólo los intentos fallidos.

Oracle es un producto de pago, con un soporte técnico prácticamente indispensable para manejar las BBDD Oracle, lo que conlleva a que la mayoría de sus productos son únicamente compatibles con los mismos productos de Oracle. Sin embargo, en Audit Vault (producto que se va a estudiar en este proyecto) se produce una excepción, siendo éste compatible con productos no fabricados por Oracle como puede ser SQL Server Database, abriéndose de esta manera a un mercado más amplio. La auditoría en Oracle se puede llevar a cabo de forma tradicional o a través de productos especializados en esta actividad.

La forma tradicional es poco intuitiva como se explicará más adelante, no posee interfaz gráfica y requiere un conocimiento más profundo de la BBDD, adquiriendo una misma persona tanto el rol de administrador como de auditor, ya que para auditar y ver los resultados almacenados se necesita tener acceso a tablas que se encuentran en el usuario sys (como norma general), debido a que éstas vienen creadas en este usuario por defecto, lo que significa que el auditor debe tener privilegios de administrador. Sin embargo, en productos novedosos como Audit Vault, el rol de administrador viene diferenciado con respecto al rol del auditor, proporcionando gran simplicidad y facilidad en su uso, sin necesidad que el auditor tenga un gran conocimiento en BBDD, lo necesario es que atienda a las especificaciones determinadas por la organización y analice los informes que emite el producto, pudiendo ser de esta manera la misma o diferentes personas quien realicen la auditoría, como se ha explicado en el punto anterior.

Desde sus inicios Oracle siempre ha buscado esa seguridad tan necesaria y demandada para las BBDD. Por ello a lo largo de la historia ha ido creando productos relacionados con dicha seguridad, hasta llegar a Audit Vault el producto en el cual se basa el desarrollo de esta investigación. A continuación se muestra la evolución que ha habido en el transcurso de los años, mostrando algunos de los productos que se han ido creando para proporcionar seguridad en este entorno de explotación específico:

1977 Desde sus inicios

Government customer

Trusted Oracle7 MLS DB

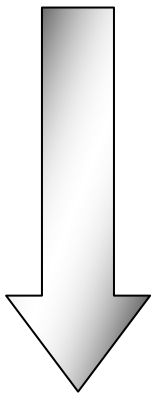
First Orange Book B1 evaluation (1993)

...

Database CC Security Eval #18 (10g R1)

Oracle Database Vault

Oracle Audit Vault



2010 En la actualidad

Tabla 1: Evolución de productos de seguridad en Oracle

Estos productos acceden al diccionario de datos o a la pista de auditoría del sistema, recogiendo los datos auditados y obteniendo unos resultados de su uso. Cada uno de ellos ha ido evolucionando hasta llegar al producto actual Oracle Audit Vault.

Estos productos facilitan la tarea de auditoría, ya que por ejemplo Audit Vault posee interfaces gráficas que permiten una visión mucho más clara y rápida de los resultados obtenidos en las auditorías realizadas. De la manera tradicional resulta mucho más tedioso y trabajoso interpretar los resultados, como ya se comentó anteriormente.

2.2 Productos de Oracle

Este apartado se centra en los productos específicos para Oracle, principalmente en la realización de la auditoría tradicional a través de comandos y la explicación de las novedosas características que implica la utilización del producto Audit Vault en la realización de auditorías.

2.2.1 Auditoría tradicional (a través de comandos)

Una auditoría, como ya se ha visto, puede ayudar a resolver fechas y tiempos de acceso de usuarios no autorizados a la BBDD.

Hay varios tipos de actividades sospechosas como ya se han ido describiendo. Éstas pueden ocurrir cuando un usuario que intenta conectarse a la BBDD. utiliza distintas combinaciones de usuario y password, cuando se producen intentos de acceso a una tabla confidencial o intentos de ejecución de comandos de creación, modificación o borrado de objetos, por lo que es importante que estas acciones queden registradas.

El diccionario de datos Oracle como ya se comentó en el apartado 2.1.3, contiene una tabla llamada SYS.AUD$ que contiene los registros de auditoría.

En algunos S.O. (sistemas operativos) se soportan auditorías que pueden acceder a la BBDD. (aparte de al S.O.) y dejar en una sola operación auditadas ambas cosas (S.O. y BBDD). En este caso, consideraremos las ventajas de auditar a nivel de BBDD.

Los pasos principales para llevar a cabo una auditoría tradicional son los siguientes:

1. Comprobar si una instancia de Oracle tiene activada la auditoría

La activación de la auditoría en Oracle Database viene definida por el valor del parámetro: AUDIT_TRAIL. Para comprobar si la auditoría de la base de datos está activada ejecutaremos el siguiente comando SQL:

```
select name, value
from v$parameter
where name like 'AUDIT_TRAIL'
```

Posibles valores del parámetro AUDIT_TRAIL:

- **NONE**: desactiva la auditoría de la base de datos. Es el equivalente a FALSE.
- **OS**: activa la auditoría de la base de datos. Los sucesos auditados se escribirán en la pista de auditoría del sistema operativo, no se auditará en Oracle sino en el sistema operativo anfitrión. Esta opción funcionará dependiendo del sistema operativo.

- **DB**: activa la auditoría y los datos se almacenarán en la taba SYS.AUD$ de Oracle. Es equivalente a TRUE.
- **DB_EXTENDED**: activa la auditoría y los datos se almacenarán en la taba SYS.AUD$ de Oracle. Además se escribirán los valores correspondientes en las columnas SQLBIND y SQLTEXT de la tabla SYS.AUD$.
- **XML**: activa la auditoría de la base de datos, los sucesos será escritos en ficheros del sistema operativo.
- **XML, EXTENDED**: activa la auditoría de la base de datos, los sucesos será escritos en el formato del sistema operativo, además se incluirán los valores de SqlText y SqlBind. [Bob Bryla, kevin Loney (2008)]

2. Activar la Auditoría de la B.D.

Para generar registros en las tablas de auditoría no basta con utilizar el comando AUDIT, es necesario activar la escritura en las tablas de auditoría activando al parámetro de inicialización AUDIT_TRAIL.

Por un lado se puede activar o desactivar modificando dicho parámetro desde el INIT.ora y por otro lado por ejemplo se puede hacer a través de sentencias SQL.

Para **activar** la auditoría:

```
ALTER SYSTEM SET AUDIT_TRAIL = "DB" SCOPE=SPFILE;
```

Para **desactivar** la auditoría ejecutaremos el siguiente comando:

```
ALTER SYSTEM SET AUDIT_TRAIL = "NONE" SCOPE=SPFILE;
```

Nota:

En Oracle 9i la auditoría viene desactivada por defecto, el valor del parámetro " AUDIT_TRAIL " está a "NONE". Al igual que ocurre con Oracle 10g.

En Oracle 11g la auditoría viene activada por defecto, el valor del parámetro " AUDIT_TRAIL " está a "DB".

3. Comandos Audit y Noaudit

- **Sintaxis comando Audit**

```
AUDIT
opc_sentencia.
 { BY usuario}
 [ BY { SESSION | ACCESS } ]

 [ WHENEVER [ NOT ] SUCCESSFUL ] ;
```

En la tabla 2 aparecen explicadas cada una de las opciones que se pueden encontrar a la hora de llevar a cabo la ejecución del comando audit o no audit:

opc_sentencia	Especifica la sentencia/s SQL que se desea auditar.
BY usuario	Indica que se quieren auditar las sentencias SQL requeridas para el usuario/s indicados. Si se omite, la auditoría se realiza para todos los usuarios de la B.D.
BY SESSION	Provoca que Oracle inserte un único registro resumen en la tabla de auditoría aunque la sentencia se ejecute varias veces en la misma sesión.
BY ACCESS	Provoca la escritura de un registro en las tablas de auditoría cada vez que la sentencia se ejecuta. Cuando se especifican auditorías de sentencias DDL o de privilegios del sistema, la auditoría por defecto es por accesos. Cuando se auditan sobre objetos o sentencias DML, la auditoría por defecto es por sesión.
WHENEVER SUCCESSFUL	Se realiza la auditoría cuando la sentencia auditada haya concluido satisfactoriamente
WHENEVER NOT SUCCESSFUL	Se realiza la auditoría cuando la sentencia auditada NO concluya satisfactoriamente.

Tabla 2: Parámetros de auditoría

En resumen y como ejemplo:

Se debe tener en cuenta que el parámetro que habilita la posibilidad de auditar la base de datos ORACLE en el init.ora es **AUDIT_TRAIL,** que el comando SQL que activa la auditoría sobre algo es **AUDIT** (para desactivar NOAUDIT) y por ejemplo que la tabla para mirar (usuario sys) el seguimiento de auditoría es **DBA_AUDIT_TRAIL.** Estos son los parámetros más importantes que se deben tener en cuenta a la hora de poder realizar una auditoría y su seguimiento. Para poder realizar el seguimiento correctamente existen diferentes vistas del diccionario de datos para hacer consultas, aparte de DBA_AUDIT_TRAIL se pueden encontrar más en el manual de Oracle 11g, [Bob Bryla, Kevin Loney (2008)].

- **Funcionamiento comando Audit:**

El comando **Audit** permite iniciar los tipos de auditoría que a continuación se detallan. Este comando puede funcionar aunque no esté activada la auditoría de la base de datos, pero no dejará constancia, ya que para que funcione correctamente es necesario que la auditoría esté activada, de la forma que ya se ha indicado.

- **Auditorías de inicio de sesión**: cada intento de conexión con la base de datos por parte de un usuario (bien una aplicación externa o las aplicaciones del propio Oracle) puede ser auditado. El comando para iniciar la auditoría de los intentos de inicio de sesión es, Audit Session, este comando auditará tanto los intentos fallidos como los aciertos.
  Para auditar sólo los intentos fallidos se utiliza el comando:

  ```
  audit session whenever not successful;
  ```

  Para auditar sólo las conexiones **correctas** se utiliza el comando:

  ```
  audit session whenever successful;
  ```

- **Auditorías de acción**: : cualquier acción que afecte a un objeto de la base de datos (tabla, enlace de base de datos, espacio de tablas, sinónimo, segmento de anulación, usuario, índice, etc.) puede auditarse. Las posibles acciones que pueden auditarse (create, alter, drop) sobre estos objetos pueden agruparse para simplificar la cantidad de esfuerzo administrativo necesario para determinar y mantener las opciones de configuración de la auditoría.

  `audit role;` Este comando activará la auditoría de las acciones: `create role, alter role, drop role y set role.`

  También se puede ser más selectivo, por ejemplo, si se quiere auditar a un usuario concreto cuando realiza la acción "update" .Ejemplo:

  ```
  audit update table by nombre_usuario;
  ```

  De esta forma se activará la auditoría para el usuario "nombre_usuario" sólo cuando ejecute el comando "update" para cualquier tabla.

- **Auditorías de objeto**: además de las acciones a nivel de sistema sobre objetos, también es posible auditar las acciones de manipulación de datos sobre objetos. Se pueden auditar operaciones de *select*, *insert*, *update* y *delete* (selección, inserción, modificación y borrado) sobre tablas. Este tipo de auditoría es similar a la anterior de auditoría de acción, la única diferencia es que el comando "Audit" incorpora un parámetro nuevo "by session" (el registro de auditoría se

escribirá una única vez por sesión) o "by access" (el registro de auditoría se escribirá cada vez que se acceda al objeto auditado).

Por ejemplo, para auditar los "insert" realizados sobre la tabla facturación" por acceso, el comando será:

```
audit insert on FACTURACION by access;
```

Nota: al indicar "by Access" hay que tener cuidado pues registrará un suceso de auditoría por cada insert, esto puede afectar al rendimiento. De ser así siempre será mejor optar por "by session" que sólo registrará un suceso de auditoría por sesión, aunque es menos exhaustivo.

Otro ejemplo, para auditar todas las acciones realizadas en la tabla "contabilidad" por sesión utilizaremos el siguiente comando:

```
audit all on CONTABILIDAD by session;
```

El comando anterior auditará todas las acciones realizadas sobre la tabla FACTURACION (select, insert, update, delete), pero sólo un registro de auditoría por cada sesión.

Otro ejemplo, para auditar las eliminaciones de registros de la tabla "nóminas":

```
audit delete NOMINAS by access;
```

- **Prerrequisitos para poder ejecutar Audit y Noaudit**

Para activar y desactivar la auditoría de las instrucciones SQL con el comando Audit se necesita el privilegio de sistema AUDIT SYSTEM.

El usuario que desee puede activar o desactivar la auditoría de objetos de un esquema, para ello tiene que ser el propietario del objeto o disponer del privilegio de sistema AUDIT ANY. (para desactivarlo, si el objeto que se eligió para la auditoría se ubica en un directorio, incluso habiéndolo creado uno mismo, se necesita el privilegio de sistema AUDIT ANY)

Para obtener los resultados de la auditoría hay que definir correctamente el parámetro de inicialización AUDIT_TRAIL. Se podrán definir las opciones de auditoría

con el comando ***Audit*** pero, si no está activada la auditoría en la base de datos, Oracle no generará los registros de auditoría.

2.2.2. Auditoría realizada con Audit Vault

En este apartado se explica teóricamente las características y el funcionamiento de Audit Vault, su utilización se mostrará en el siguiente capítulo a la hora de realizar las pruebas pertinentes.

La estructura del producto Audit Vault es básicamente la representada por la siguiente ilustración.

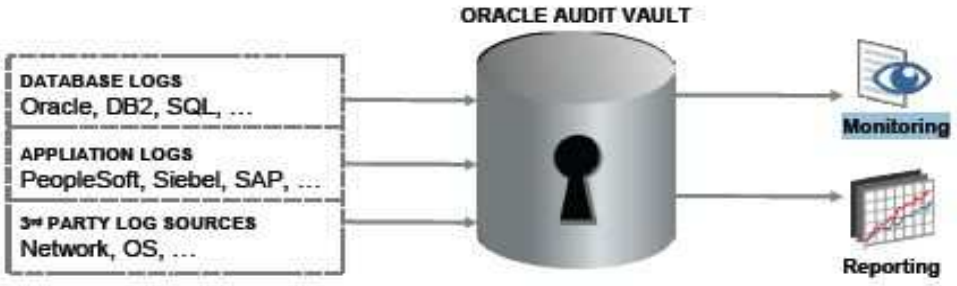


Ilustración 1: Estructura Audit Vault

El entorno en el cual se va a realizar la implantación del nuevo agente para el control de las BBDD es en la versión 10.2.0.4.0 de Oracle. La herramienta que se va a utilizar por motivos de compatibilidad, y debido a las especificaciones dadas para el estudio, va a ser Audit Vault.

Audit Vault automatiza la recopilación y consolida los datos de auditorías. Ofrece un depósito seguro, que permite obtener informes simplificados, análisis, y detección de amenazas respecto de los datos de auditorías (estas características se estudian en capítulos posteriores). Asimismo, el entorno de auditoría de la BBDD se administra y monitorea centralmente desde Audit Vault como se muestra en la Ilustración 2.

Ilustración 2: Información general de Oracle Audit Vault

Recopilación y Consolidación de Datos de Auditoría

Oracle Audit Vault recopila los datos de auditoría de la BBDD desde las pistas de auditoría de Oracle. También puede leer las conexiones de transacción para capturar los valores anteriores / posteriores relacionados con las transacciones realizadas. Oracle Audit Vault soporta Oracle9*i* Database Versión 2, Oracle Database 10*g* Versión 1 y posteriores versiones. Audit Vault es compatible con BBDD fuentes no pertenecientes a Oracle pudiendo monitorear por ejemplo BBDD Microsoft SQL Server.

Informe Simplificado de Cumplimiento

Oracle Audit Vault ofrece informes de evaluación de auditorías estándar sobre los usuarios privilegiados, la administración de cuentas, los roles y privilegios, la administración de objetos y la administración de sistemas en toda la empresa. Los informes impulsados por parámetros pueden definirse mostrando la actividad de conexión del usuario en múltiples sistemas y dentro de períodos específicos, como por ejemplo, los fines de semana. Oracle Audit Vault ofrece un esquema de depósito de auditoría abierto al cual se puede acceder desde Oracle BI Publisher, Oracle Application Express, o cualquier herramienta de informes de terceros.

Detección de Amenazas con Alertas

Las alertas actúan de la siguiente manera. En primer lugar se auditan los eventos que se quieran registrar en la BBDD y en segundo lugar sobre esos eventos auditados se pueden crear alertas tanto de advertencias como alertas críticas, pasando a estar registradas inmediatamente en el nuevo producto, sin esperar a que todos los elementos auditados se transfieran a Audit Vault. Las alertas pueden estar relacionadas con cualquier evento de BBDD que pueda auditarse, con inclusión de los eventos del sistema como los cambios en las tablas de aplicaciones, el otorgamiento de roles, y la

creación de usuarios privilegiados en sistemas sensibles. Oracle Audit Vault crea resúmenes gráficos de las actividades que generan alertas.

Costos de TI con las Políticas de Oracle Audit Vault

El personal de seguridad de TI trabaja con auditores para definir el entorno de auditoría en las BBDD y otros sistemas de toda la empresa a fin de satisfacer tanto los requerimientos de cumplimiento como las políticas de seguridad interna. Oracle Audit Vault tiene la capacidad de abastecer y revisar los entornos de auditoría en múltiples BBDD desde una consola central, "reduciendo el costo y la complejidad relacionada con la administración de los entornos de auditoría de toda la empresa." [Best Practices (2007)]. Esto se comprobará en el siguiente punto con la realización de las pruebas pertinentes.

Por lo tanto Oracle Audit Vault se supone que automatiza el proceso de consolidación y análisis, convirtiendo los datos de auditoría en un recurso de seguridad para ayudar a abordar los desafíos de cumplimiento y seguridad, a través de informes y gráficas. Por lo tanto el producto queda resumido en las siguientes características según la información adquirida del producto:

- Asegura la integridad de los datos de auditoría.
- Oracle Audit Vault – warehouse (almacén de datos) especializado para datos de auditoría.
- Agrega los datos de auditoría – fuentes Oracle y no Oracle.
- Asegura datos de auditoría valiosos.
- Monitorea cambios asociados con usuarios privilegiados.
- Informes para la conformidad.

Estas características serán estudiadas para comprobar su veracidad en un entorno de explotación mediante la simulación de un entorno de pruebas.

Capítulo 3 Instalación de Audit Vault

Tras haber finalizado un estudio previo del estado del arte actual y analizado las distintas opciones en cuestión de tecnologías y entornos de desarrollo, pasamos a describir el problema en cuestión y la solución técnica aportada con el fin de cubrir los objetivos marcados.

3.1 Descripción del problema

Es fundamental entender por qué es necesario auditar la BBDD y saber elegir cuál es el tipo de auditoría apropiada en cada caso, como ya se ha explicado en puntos anteriores. Partiendo del entorno actual en el cual se pretende ver la viabilidad de Audit Vault como producto de auditoría hay que tener en cuenta los siguientes requisitos:

- El 95% de las bases de la URJC se encuentran implementadas en Oracle.
- Existen personal de la universidad trabajando sobre estas bases de datos ya sea desde una aplicación o directamente sobre ellas.
- Existe personal externo a la universidad que tiene acceso a nivel de administración a estas bases de datos y sería conveniente auditar las conexiones y cambios de estos.

Dados estos requisitos Audit Vault podría ser el producto más óptimo para monitorizar este entorno debido a la compatibilidad con las bases de datos del mismo fabricante.

En este caso, el producto que se encuentra instalado y del cual se puede obtener soporte por condiciones previas definidas es Oracle, y su producto de auditoría más novedoso, como ya se ha hecho referencia, es Audit Vault, por lo que es necesario realizar un estudio para ver si es conveniente, adecuado y además factible la instalación de este producto para auditar unos sistemas de información realmente grandes, con un gran número de accesos y secciones críticas como pudiera ser la universidad.

Es necesario saber si el consumo de recursos es o no excesivo para saber si se encuentra en un sistema capacitado para soportar esta instalación en estos momentos y si la actuación de Audit Vault se adecua a los resultados que se quieren y se esperan conseguir.

3.2 Análisis y Especificación de requisitos para Audit Vault.

La arquitectura de Audit Vault consta de dos componentes principales que trabajan para almacenar y proteger los datos de auditoría. Estos son:

- Audit Vault Server (Servidor de Audit Vault)
- Audit Vault Collection Agent (Colección de agentes de Audit Vault)

Estos se encuentran relacionados entre sí, y el administrador y el auditor directamente con el servidor. Explicando de manera breve la siguiente ilustración se ve como el administrador realiza la administración del servidor y el rol del auditor se encuentra separado de éste, encargándose de las alertas e informes generados por la actividad. El agente trasfiere los datos al servidor a través del colector, y éstos son recogidos en el almacén de datos del servidor, visibles a través de su interfaz gráfica.

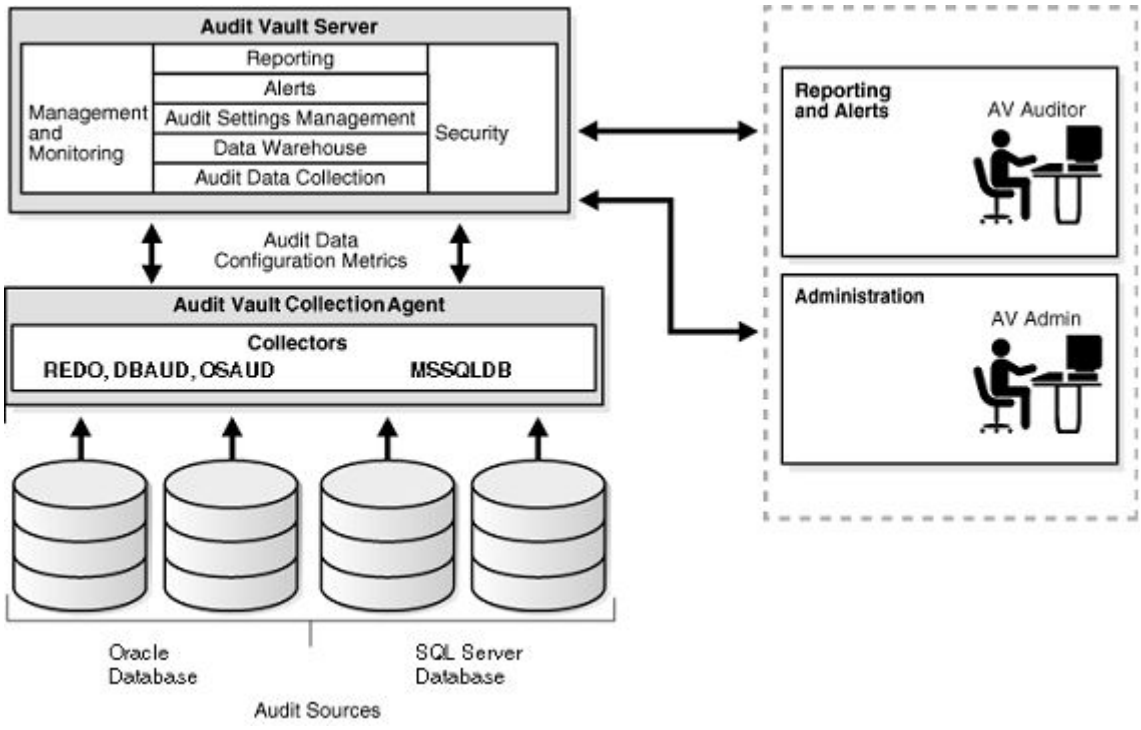


Ilustración 3: Relación entre Servidor, Agente, Auditor y Administrador.

Haciendo un más hincapié tanto en el servidor como en el agente se puede decir de ellos lo siguiente:

- **El Servidor de Audit Vault** es una aplicación (apilada) que contiene un almacén de datos, basado en una instalación personalizada de Oracle Database 10g (10.2.0.3)

con Oracle Database Vault para aportar seguridad, y componentes OC4J (Oracle Application Server containers for (4) J2EE) para dar soporte a una consola de Audit Vault y control a la BBDD de Enterprise Manager

- **La Colección de Agentes de Audit Vault** es el grupo de agentes que es responsable de la gestión de los colectores y del mantenimiento de la "cartera" de Audit Vault

Los Colectores (Collectors) es específico de una BBDD fuente, y actúa como intermediario entre la fuente y el servidor de Audit Vault, arrastrando el 'Audit Trail data' desde la fuente y al servidor Audit Vault (a través de SQL * Net)
Los datos enviados pueden tener una tipología diferente dependiendo del valor asignado al parámetro AUDIT_TRAIL en el fichero INIT.ora (NONE, OS, DB)

La cartera (Audit Vault Wallet) se utiliza para mantener la contraseña del colector para conectarse a las BBDD fuentes y de esta manera extraer los datos de auditoría de la BBDD.

Por lo tanto la colección de agentes de Audit Vault proporciona apoyo para la recogida de datos de auditoría. El agente carga los colectores, proporcionándolos una conexión con el servidor Audit Vault para enviar los datos de auditoría y métricas en tiempo de ejecución de los colectores. Audit Vault se comunica con BBDD fuente a través de su agente.

3.2.1 Plan de implantación

Mientras Audit Vault ofrece la consolidación y el almacenamiento seguro de los datos de la auditoría, la planificación de la instalación de los componentes de Audit Vault garantizará una instalación más rápida y el éxito global de la implantación del producto en un entorno de pruebas. Para ello a continuación aparecen los prerrequisitos, es decir, los aspectos previos a la instalación tanto para 'Audit Vault Server' como para 'Audit Vault Colection Agent'.

- **Servidor de Audit Vault**

El servidor de Audit Vault debe estar instalado en su propio host o en un host que contenga otro repositorio de BBDD como Entreprise Manager Grid, Control o RMAN (recovery manager). Al instalar el servidor de Audit Vault independiente de los servidores de la BBDD fuente proporciona las siguientes ventajas:

- **Mayor disponibilidad.** Cuando el servidor de Audit Vault está en un servidor independiente de las BBDD fuente lo que sucede es que la disponibilidad no dependerá del estado de levantado o bajado (up/down) de la máquina fuente y por lo tanto los datos de auditoría siguen siendo recogidos de todas las fuentes que se ejecutan.
- **Audit Trail asegurado**. Al extraer los registros de auditoría fuera de la BBDD de origen lo más rápido posible, hay muy pocas oportunidades para que usuarios con privilegios tanto de la BBDD, como del sistema operativo puedan modificar dichos registros de auditoría, lo que proporciona una gran seguridad.

- **Colección de agentes de Audit Vault**

La BBDD Oracle puede escribir datos en una pista de auditoría dentro de la base de datos (SYS.AUD $ / SYS.FGA_LOG $) y / o archivos del sistema operativo. El registro en línea (redo log) de la BBDD de Oracle también contiene información de antes / después de los cambios de valor de los datos. Como ya se ha dicho en el punto anterior Audit Vault despliega un proceso que se llama 'colector' que es específico de la pista de auditoría de BBDD Oracle para extraer dichos datos y enviarlos al servidor de Audit Vault. Los tres tipos de colectores se llaman DBAUD de la auditoría de base de datos, OSAUD de archivos del sistema operativo escritos por la base de datos Oracle, y REDO para extraer datos de auditoría desde los redo.

El agente puede ser instalado sobre la misma máquina que va a ser auditada, sobre las máquinas de Audit Vault Server, o sobre una máquina diferente a ambas.

A continuación aparece cada uno de los escenarios para determinar la mejor ubicación dentro de su entorno para Audit Vault Collection.

- **En la misma máquina que la BBDD auditada (recomendado)** Si los datos se guardan en el sistema operativo, el agente debe estar instalado en la misma máquina que estos archivos del sistema operativo

Ilustración 4: Agente y BBDD fuente en la misma máquina, [Best Practices (2007)]

- **En la máquina de Audit Vault Server** Si el destino de los datos de auditoría son las siguiente tablas SYS.AUD $ / $ SYS.FGA_LOG, entonces el agente debe ser instalado en el servidor. Esto significaría que todos los componentes de software utilizado por Audit Vault se consolidarán en una sola máquina.

Ilustración 5: Agente y Servidor en la misma máquina, [Best Practices (2007)]

- **En una máquina diferente** El agente se encuentra instalado en una máquina diferente a la BBDD fuente y al servidor**.**

Ilustración 6: Agente, Servidor y BBDD fuente en diferentes máquinas, [Best Practices (2007)]

- **Selección de las Fuentes y Uso de los Colectores**

La elección de la fuente es lo primero que se debe realizar y a partir de ahí se elige el tipo del colectores idóneo sobre la BBDD origen que se haya seleccionado. La tabla siguiente muestra las fuentes de apoyo sobre las cuales se puede instalar Audit Vault, los tipos de fuentes dependiendo del origen de la BBDD, los diferentes tipos de colectores para recoger los datos de auditoría y por último que tipo de datos se encuentran recogidos en los diferentes tipos de colectores.

Fuente	Tipo de Fuente	Tipo de Colector	Audit Trail
BBDD Oracle	ORCLDB	OSAUD DBAUD REDO	Logs del sistema operativo (Operating system logs) Tablas de auditoría de la BBDD (Database tables) Redo logs
BBDD SQL Server	MSSQLDB	MSSQLDB	C2 audit logs, server-side trace logs, Windows Event log

Tabla 3: Fuentes de apoyo, tipos de fuentes, tipos de colectores, y pistas de auditorías

El tipo de colector elegido en el caso de BBDD Oracle va a depender del tipo de datos que se quieran registrar. Si la recogida de los datos procede desde diferentes BBDD origen, hay que repetir el proceso de selección del colector para cada origen diferente.

En la siguiente tabla se describen las características de cada uno de ellos:

Audit Operation	OS Log	DB Audit Table	Redo Log
SELECT	✓	✓	
DML	✓	✓	✓
DDL	✓	✓	✓
Before and After Values			✓
Success and Failure	✓	✓	
SQL Text	✓ (for SYS)	✓	
SYS Auditing	✓		✓
Other considerations	Separation of Duties	FGA data	Supplemental logging for all values

Tabla 4: Características de los diferentes colectores (valor de AUDIT_TRAIL)

Los tres tipos de colectores DBAUD, OSAUD, y REDO recuperan los registros de auditoría de diferentes lugares de la fuente de base de datos de Oracle.

El elegido en esta investigación has sido **DBAUD**, ya que aún siendo muy interesante el colector del sistema operativo, se disponen de otros medios y otros recursos que no hacen necesario este tipo de colector.

A continuación aparece una vista detallada de la arquitectura tanto de la fuente, como del agente, como del servidor de Audit Vault. En la ilustración 7 se puede ver los dos tipos de fuente que Audit Vault soporta, tanto BBDD Oracle como BBDD Microsoft SQL Server. Centrándose en las BBDD Oracle, se ve en la ilustración donde se encuentran almacenados los datos y como el agente a través de los colectores los trasfiere al repositorio de auditoría situado en el servidor. La arquitectura se encuentra basada en OAS (Oracle Application Server) y ofrece su uso como una plataforma completa e integradora. Es un servidor que permite la integración de aplicaciones existentes y la personalización en tiempo real. Por otro lado permite que el control de seguridad y la administración y configuración del sistema estén centralizados facilitando su uso y gestión. Uno de sus componentes es OC4J que ofrece un contenedor Java2

estándar que sirve para desarrollar, integrar e implementar portales, aplicaciones y servicios web.

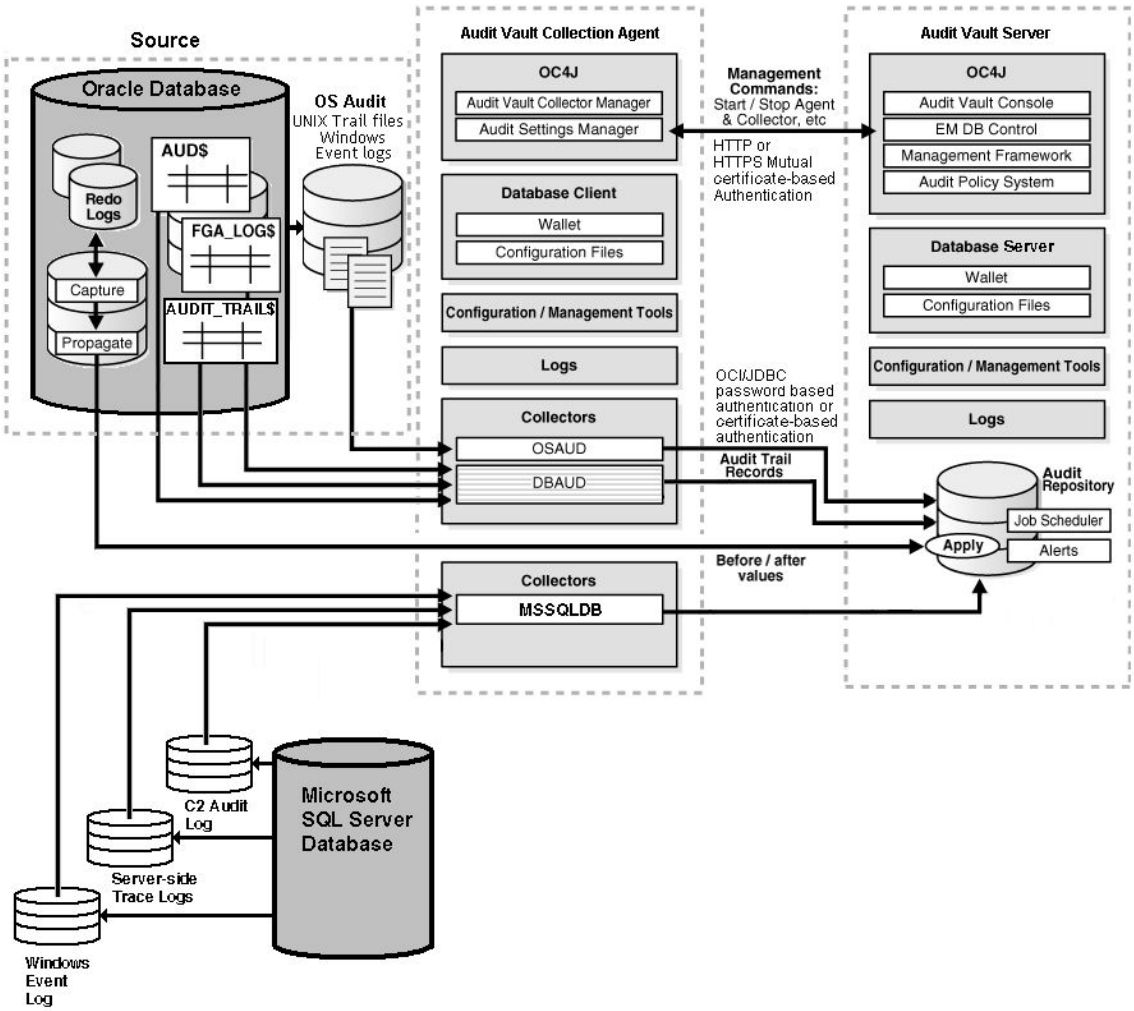


Ilustración 7: Descripción los componentes del servidor Audit Vault [Administrator's Guide (2008)]

3.3 Implantación de entorno de pruebas

Una vez visto aspectos fundamentales del producto Audit Vault, se lleva a cabo la instalación de dicho producto, para ello se irá haciendo referencia a manuales de instalación , los cuales se encuentran en los anexos, realizados para tener una guía práctica y útil a la cual poder acudir rápidamente en cualquier caso de duda con respecto a dicha instalación.

En primer lugar se instalará el servidor de Audit Vault, (Anexo I,) en una máquina independiente, ya que es más ventajoso para su posterior utilización como se explica en el punto 3.2.1 Plan de Implantación, Audit Vault Server.

En segundo lugar se realizará la instalación del Agente de Audit Vault, en este caso en una máquina Linux, (Anexo II). En esta máquina se encuentra instalada una BBDD sobre la cual se irán realizando las pruebas para su posterior estudio.

Una vez instalado tanto el servidor como el agente se deben conectar el uno con el otro. Esto se realiza añadiendo un nuevo host a Audit Vault, es imprescindible que anteriormente se haya instalado un servidor de Audit Vault y además se tenga creada una BBDD sobre la cual se quieran realizar las pruebas de auditoría, Anexo III.

A continuación se muestra el esquema de las máquinas sobre las cuales se instalan tanto el servidor como el agente:

SIL.URJC.ES
Servidor Audit Vault
[AV_SERVER]

Máquina donde se realiza la instalación de Audit Vault.

Para añadir un nuevo host a monitorizar de Audit Vault esta máquina tiene que estar correctamente instalada.

Máquina que contiene una base de datos a monitorizar.
En esta máquina se instalará el agente de Audit Vault. Antes de instalar el agente hay que registrar en el servidor el agente que se quiere instalar. (Ver anexo III)

COFIO.URJC.ES
Instalación de agente
Audit Vault
Base de datos reco1
Puerto:2485
[SOURCE_DATABASE]

Ilustración 8: Instalación de Oracle Audit Vault

Notas:

[AV_SERVER] → Acciones realizadas en el servidor de Audit Vault

[SOURCE_DATABASE] → Acciones realizadas en la máquina cliente

3.4 Creación de entorno de pruebas

Las pruebas de auditoría buscan obtener evidencia de que los controles establecidos existen en realidad se utilizan y ejecutan correctamente (Al conjunto de pruebas resultante se denomina Programa de auditoría.).

Con la creación de un entorno de pruebas se pretende recrear la actividad de una BBDD, a la cual se accede con mucha asiduidad y se realizan una gran cantidad de consultas, borrados, modificaciones y creaciones sobre dicha BBDD. Por lo tanto, se intenta asimilar lo máximo posible al entorno en el cual se instalaría el producto si los resultados de las pruebas fueran favorables.

3.4.1 Simulación de un entorno real

Es necesario adquirir una visión general de cómo estará formado el entorno de pruebas. Para la creación del entorno se han utilizado 4 máquinas diferentes para la simulación de la actividad de la BBDD, la cual va a ser auditada.

Desde las 4 máquinas se van a lanzar scripts los cuales contienen definidas sentencias SQL. En su inmensa mayoría son consultas select, asimilándolo a ciertas BBDD de la universidad que tienen gran actividad en consultas. En una pequeña parte de la actividad se realizan modificaciones, creaciones, borrados y otras acciones que no se producen de manera tan asidua, siendo más susceptibles de ser auditadas. Estos script son llamados desde el bucle de otro script para que estas sentencias se ejecuten un número elevado de veces para recrear una actividad considerable sobre la BBDD fuente.

La actividad se va a realizar sobre los esquemas de ejemplo de la BBDD Oracle, ya que son esquemas muy completos que representan el funcionamiento de una empresa. Y de esta manera asimilarlo lo máximo posible a un entorno de explotación real.

Estos scripts se van a situar en el cron de linux (en windows, tareas programadas) y se van a ir ejecutando según han sido programados. Habrá momentos en los que se recree una mayor actividad por la coincidencia en ejecución de un mayor número de script y en otros en los que la BBDD se encuentre en una actividad mínima.

Se genera actividad en la BBDD de pruebas desde diferentes máquinas.

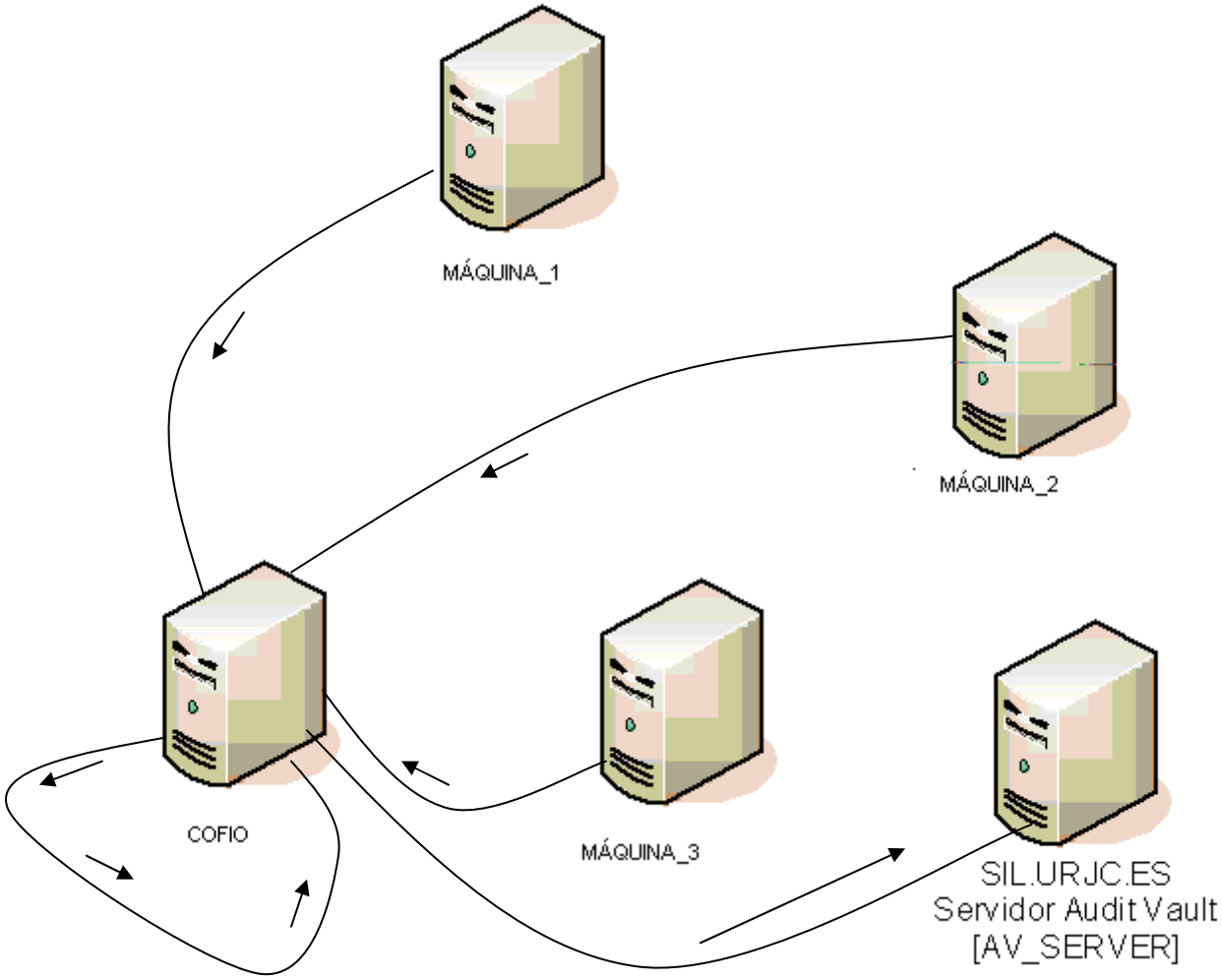


Ilustración 9: Simulación de un entorno real

3.4.2 Generación de actividad

Se realiza mediante scripts que simulan la actividad que generarían tanto los usuarios como los administradores accediendo a la BBDD. Aparte de los scripts también se realizan manualmente acciones críticas que se llevan a cabo de manera extraordinaria en la BBDD, como podría ser el borrado de un tablespace pudiendo ser registrada como una alerta crítica como aparece en el punto 4.2 Realización de la Auditoría Informática.

Estos scripts como ya se ha dicho en el punto anterior se van a ir ejecutando según se encuentren programados en el cron o tareas programadas según se trate de Linux o Windows respectivamente.

Parte de los scripts son los encargados de generar una actividad asidua en la BBDD, sin que implique una necesidad exhaustiva de un registro de estos datos por parte de la auditoría, ya que como se ha explicado todos los datos son susceptibles de ser auditados pero no es recomendable que esto ocurra. Una auditoría debe de centrarse en la actividad que afecte de manera más 'importante' a la BBDD según se haya establecido en los criterios para la realización de dicha auditoría.

Por otro lado, parte de los script se dedicarán a crear actividad que aparte de ser auditada quedará registrada en el producto para posteriores estadísticas tanto en alertas de advertencia como alertas críticas.

A continuación aparecen las ejecuciones que generan la actividad, las cuales se encuentran programadas en el cron de cada una de las diferentes máquinas.

MÁQUINA1(guadiana2)	MÁQUINA2(pisuerga2)
00,30 * * * * generahrcountries.sh 150r 15,45 * * * * generahrdepartaments.sh 160r 30,55 * * * * generashpromotion 180r	00,30 * * * * generashpromotion.sh 180r 15,45 * * * * generashproducts.sh 200r

COFIO(Actividad critica)	MÁQUINA3(nalon2)
05 * * * * script_HR.sh 35 * * * * script_HRyOE.sh 15 * * * * script_OE.sh 55 * * * * script_SHyHR.sh	10,40 * * * * generashcountries.sh 150r 20,50 * * * * generashproducts.sh 200r

Tabla 5: Scripts lanzados en el cron

En la máquina 1, máquina 2 y máquina 3 se lanzan los scripts con consultas select. Su forma de ejecución situándose por ejemplo en la máquina 1, en el primer script es la siguiente:

Este script se ejecutará a todas las horas a en punto y a y media a lo largo de todo el día, durante todos los días, de todas las semanas, de todos los meses. Las otras dos máquinas actúan igual, pero su ejecución es en diferentes minutos.

COFIO actúa de la misma manera pero al generar una actividad más crítica cada script sólo se lanza una vez cada hora.

3.4.3 Esquemas de pruebas

Los esquemas para las pruebas que se han utilizado han sido los pertenecientes a los esquemas de ejemplo de Oracle. Estos esquemas han sido elegidos debido a la gran variedad de dependencias establecidas entre ellos.

Estos esquemas de muestra de la BBDD son un conjunto de esquemas interrelacionados entre sí, proporcionando a través de capas cierta complejidad.

Los esquemas son los siguientes:

Un sencillo esquema de *Human Resources* (HR) es útil para la introducción de temas básicos. Una extensión de este esquema es compatible con Oracle Internet demos.

Un segundo esquema, Order Entry (OE), es útil para tratar asuntos de complejidad intermedia. Muchos tipos de datos está disponibles en este esquema, incluyendo tipos de datos escalares.

El subesquema *Online Catalog* (CO) es una colección de objetos de BBDD relacionales con objetos construidos dentro del esquema de la OE.

El esquema *Product Media* (PM) se dedica a los tipos de datos multimedia.

Un conjunto de esquemas reunidos bajo el nombre de esquema principal de *Information Exchange* (IX), puede demostrar las capacidades de Oracle Advanced Queue Server.

El esquema *Sales History* (SH) está diseñado para permitir demostraciones con grandes cantidades de datos. Una extensión a este esquema proporciona soporte para el proceso de análisis avanzado.

De esta manera realizando las pruebas sobre una BBDD con unas relaciones entre sus esquemas similares a los que pudiera tener una BBDD de explotación en la universidad se pueden obtener unos resultados en la investigación más fiables.

Los esquemas sobre los cuales se van a realizar las pruebas son los siguientes. [Sample Schemas (2008)].

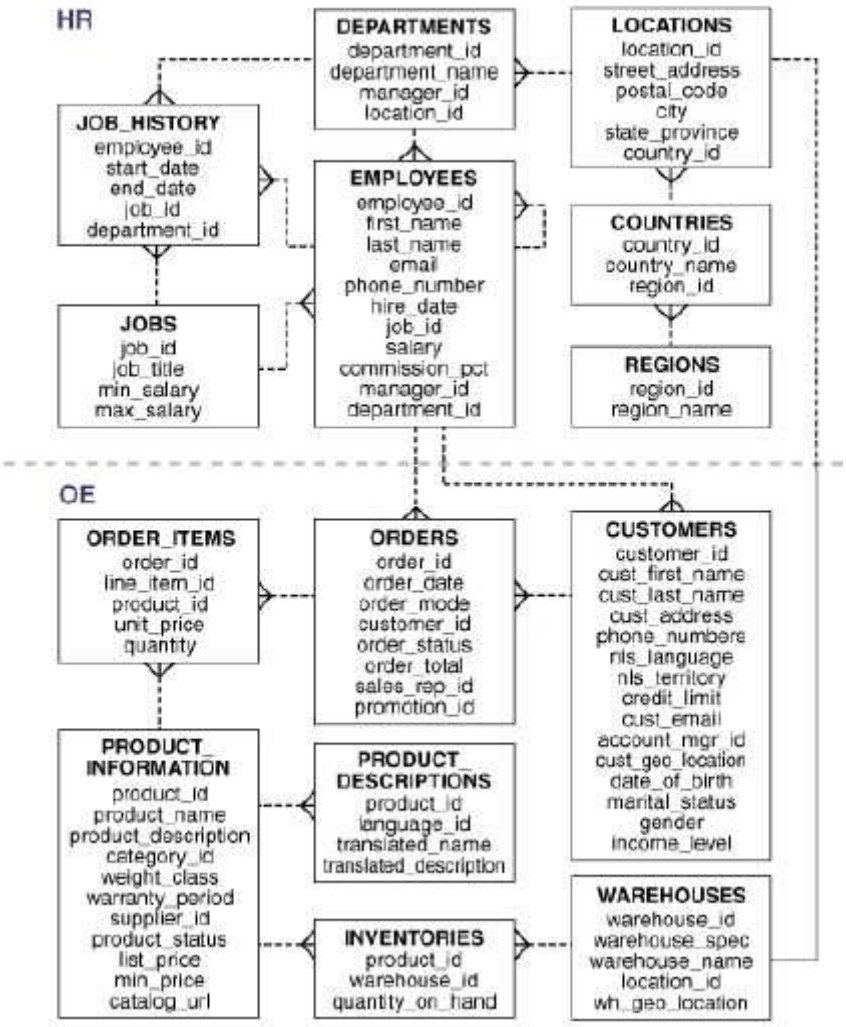


Ilustración 10: Esquemas de pruebas HR y OE [Sample Schemas (2008)]

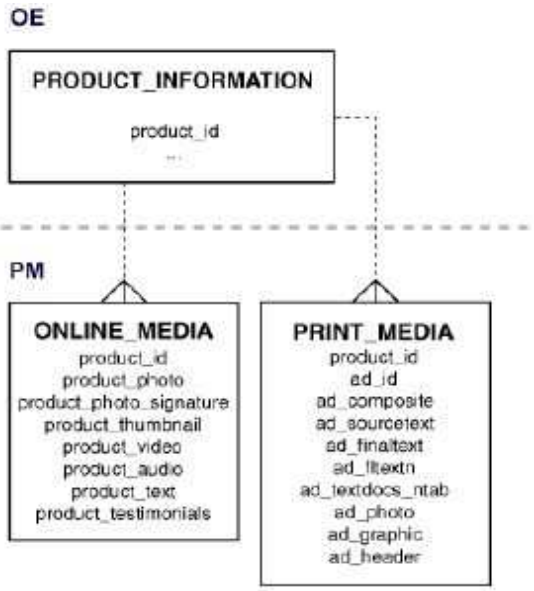


Ilustración 11: Esquemas de pruebas OE y PM [Sample Schemas (2008)]

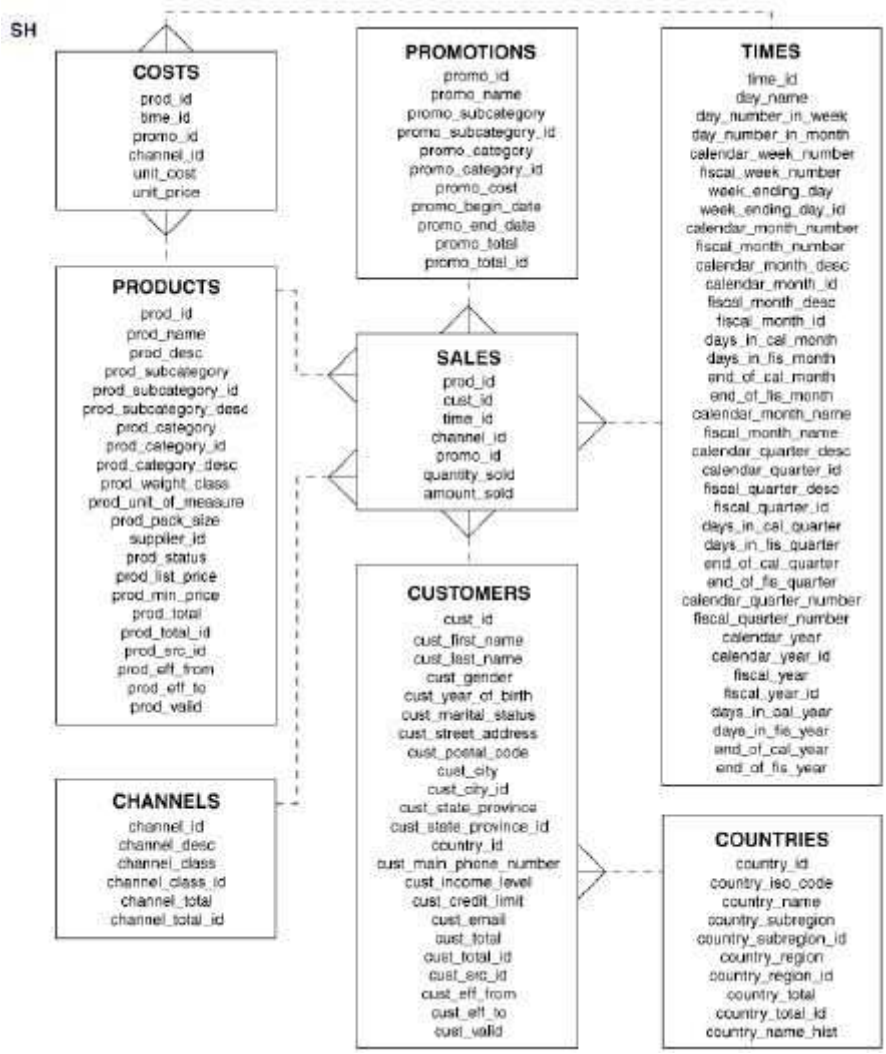


Ilustración 12: Esquema de pruebas SH [Sample Schemas (2008)]

3.4.4 Registro de actividad

A partir de ahora con la creación de otros tres script se analiza el consumo de los procesos más relevantes en Audit Vault:

- El consumo emagent
- El consumo avaudcoll
- El consumo java
- El consumo de cpu por Oracle al realizar la actividad

A la vez que se va generando la actividad, a través de la realización de otros scripts, ésta va quedando registrada en unos logs.

Estos son colocados en el cron o tareas programadas según corresponda a Linux o Windows respectivamente, y se van ejecutando de manera periódica (más concretamente cada minuto) para ir recogiendo de manera continuada la actividad de la BBDD y el consumo de recursos que lleva a cabo la auditoría con el producto Audit Vault.

Estos scripts se encuentran programados de la siguiente manera, para que la información más relevante como es el consumo de cpu y de memoria queden registrados en los logs a los cuales se les envía la información:

```
*/1 * * * * /database/script/tope.sh >> ~/emagent.log
#*/1 * * * * /database/script/topa.sh >> ~/avaudcoll.log
#*/1 * * * * /database/script/topo.sh >> ~/oracle.log
#*/1 * * * * /database/script/topjava.sh >> ~/java.log
```

Capítulo 4 Realización de Auditoría

Las fases de una metodología típica de auditoría son:

- Definición de ámbito y objetivos
- Estudio previo
- Determinación de recursos
- Elaboración del Plan
- Realización
- Elaboración del Informe Final.

Una vez vistas estas fases y sabiendo que este proyecto se orienta a la cantidad de consumo de recursos por parte del producto de Audit Vault, estudio de uso y beneficios e inconvenientes que reporta su instalación a la URJC, y no centrarse específicamente en el tipo de control, seguridad y fiabilidad de la BBDD. Estas seis fases metodológicas pueden agruparse en dos grandes etapas: la primera, de preparación y planificación de la Auditoría informática, y la segunda, la propia realización de la Auditoría informática. Ambas etapas se clasifican a continuación.

4.1 Preparación y Planificación de Auditoría Informática

El ámbito de la auditoría es lo que va a determinar los límites de la misma. Es fundamental y necesario un pleno acuerdo entre auditores y "clientes" en el caso de que los haya, sobre las funciones las materias y las organizaciones a auditar. En este caso el ámbito se encuentra determinado por las BBDD pertenecientes a la URJC, habiendo creado entornos de pruebas que se asimilen a éstas.

Sabiendo que en una universidad se recoge una gran actividad en sus BBDD tanto por parte de los alumnos, como de los PDI, PAS y demás personal perteneciente a ésta, es conveniente clarificar las materias, funciones u organizaciones que no serán auditadas. En los esquemas de pruebas la actividad generada corresponde a aquella que más se realiza en la universidad (consultas, modificaciones, creaciones y borrados) y la información auditada es aquella que se cree más conveniente para asemejarlo con las BBDD reales en explotación, pero en estas pruebas, como no es el fin, no se realiza un estudio sobre lo que mejor debería ser auditado en los esquemas de prueba de Oracle.

Aunque en cualquier auditoría a realizar es fundamental fijarse desde un principio los objetivos específicos de la Auditoría informática que se va a llevar a cabo.

Por lo tanto en este apartado de preparación y planificación de auditoría informática se concluye lo siguiente: como en este caso son BBDD de pruebas con esquemas de pruebas, no hay aspectos realmente críticos, por lo que la auditoría se realizará sobre la actividad que se crea que es más conveniente y más sensible de ser auditada.

Esta información registrada en el producto Audit Vault se divide en tres:

- Datos de auditoría recogidos.
- Alertas de advertencias sobre los datos de auditoría recogidos.
- Alertas críticas sobre los datos de auditoría recogidos.

El producto da posibilidad de que aparte de recoger los datos, los distinga en diferentes categorías y aparezcan recogidos en diferentes secciones de Audit Vault, ya que su grado de importancia en cuanto a su revisión varía de unos datos a otros. Esto es una gran novedad con respecto a la auditoría tradicional porque facilita la distinción de categorías entre los diferentes datos auditados a través de una interfaz gráfica.

Una vez delimitado el ámbito y establecidos los objetivos, se realizará un estudio previo que permita obtener la información y visión global suficiente para estimar los recursos necesarios en la elaboración de la Auditoría informática. En este caso el entorno organizativo, el entorno operativo software y el entorno operativo hardware en el que se va a realizar el estudio es el siguiente:

- **Entorno organizativo:** Está compuesto por las BBDD de la universidad, este entorno se asimila a través de los esquemas de ejemplo de Oracle. Estos ejemplos han sido elegidos para realizar las pruebas debido a que su esquema relacional es muy completo y puede representar en cierta manera muchas relaciones que se encuentran en el entorno de explotación real, como pueden ser claves ajenas, etc. (3.5.3 Esquemas de pruebas).

- **Entorno operativo software:** El software utilizado en la investigación es el siguiente:

  Sil, Servidor

  Audit Vault 10.2.0.3

Red Hat Enterprise Linux Server release 5.2 (Tikanga)

Cofio, BBDD fuente.

Oracle Enterprise Edition 10.2.0.4.0

Agente de Grid Control (para utilizar el producto Enterprise Manager)

Agente Audit Vault (para utilizar el producto Audit Vautl)

Red Hat Enterprise Linux AS release 4 (Nahant Update 7)

Clientes Linux

Cliente Oracle 10.2.0.4.0 tnsnames configuración de los clientes para tener acceso a la máquina cofio

Cliente Windows

Cliente Oracle 9.2

Toad a través de su interfaz utiliza el cliente de Oracle.

Agente de Grid control para utilizar Enterprise Manager.

- **Entorno operativo hardware:** Las máquinas utilizadas son las siguientes.

Sil, Servidor

AMD Athlon(TM) XP 2200+

Red Hat Enterprise Linux Server release 5.2 (Tikanga)

Cofio, BBDD fuente

Tiene dos procesadores: Intel(R) XEON(TM) CPU 2.00GHz

Red Hat Enterprise Linux AS release 4 (Nahant Update 7)

Clientes Linux

Cliente oracle 10.2.0.4.0 Es necesaria la configuración del tnsnames en los clientes para que estos puedan tener acceso a la máquina Cofio (puerto por el que van a acceder a la BBDD fuente).

Cliente Windows

Intel® Pentium® 4 CPU 2.00GHz 2.00GHz, 1,00GB de RAM

Microsoft Windows XP, Profesional, Versión 2002, seviche Pack 3

El tnsnames al igual que en las máquinas Linux debe estar configurado.

4.2 Realización de la Auditoría Informática

Para realizar la auditoría primero se determinan los elementos a auditar. Estos pueden ser tanto de sentencias, como de objetos, privilegios, FGA (auditoría de grano fino) o como establecer reglas de captura propias. Los datos de auditoría pueden ser

capturados como se muestra a continuación asociando la actividad con los privilegios asignados.

Privilegio	Usuario	Usuario de Proxy	Condición de Ejecución	Granularidad de Auditoría	En Uso	Necesario
DROP ANY TABLE			Ambos	BY ACCESS	⇧	✓
ALTER ANY TABLE			Ambos	BY ACCESS	⇧	✓
CREATE ANY TABLE			Ambos	BY ACCESS	⇧	✓
ALTER DATABASE			Ambos	BY ACCESS	⇧	✓
ALTER ANY PROCEDURE			Ambos	BY ACCESS	⇧	✓
CREATE ANY JOB			Ambos	BY ACCESS	⇧	✓
CREATE ANY LIBRARY			Ambos	BY ACCESS	⇧	✓
CREATE EXTERNAL JOB			Ambos	BY ACCESS	⇧	✓
CREATE PUBLIC DATABASE LINK			Ambos	BY ACCESS	⇧	✓
GRANT ANY OBJECT PRIVILEGE			Ambos	BY ACCESS	⇧	✓
GRANT ANY PRIVILEGE			Ambos	BY ACCESS	⇧	✓
GRANT ANY ROLE			Ambos	BY ACCESS	⇧	✓
CREATE SESSION	JENIFER		Ambos	BY ACCESS	⇧	✓

Ilustración 13: Sentencias de auditoría sobre privilegios

La ilustración 13 corresponde a la interfaz del producto Audit Vault, conectado con el perfil auditor (no administrador). Se pueden ver diferentes sentencias de auditoría que se generan a través de la interfaz gráfica, la cual facilita mucho su creación. Esto es debido a que a la hora de crear una sentencia de auditoría aparecen directamente todas las sentencias que se pueden auditar en una BBDD, generando mayor sencillez y evitando posibles errores de sintaxis debido a que estas sentencias se encuentran recogidas en tablas y sólo hay que elegir cuál se necesita, por lo tanto no es necesario tener un conocimiento extenso de todas las sentencias de auditoría aplicables en la BBDD como es el caso de la auditoría tradicional.

Si se analiza por ejemplo el significado que tiene la primera línea de la tabla en la ilustración 15, consiste en que va a ser auditada y por lo tanto registrada toda la

actividad que realice cualquier usuario, relacionada con el borrado de tablas en la BBDD. Esta actividad se registrará por acceso.

Estos datos de auditoría quedan registrados en la máquina fuente, es decir, donde se encuentra la BBDD a auditar (Cofio). Estos datos son recogidos por el colector y transfieren al servidor (Sil) cada cierto tiempo, según lo haya determinado el perfil del administrador del producto Audit Vault. En este caso se transmiten diariamente, ya que es la mejor opción para que no haya un tráfico excesivo de datos, para que en el caso de que se produzcan daños en la máquina fuente estos datos se encuentren seguros en otra máquina y para que la carga de datos acumulados en el tablespace, en el cual se guardan los datos de auditoría de la máquina fuente no lo sobrecarguen y puedan ser eliminados diariamente.

La tarea de eliminación de datos de auditoría es una tarea manual, tanto en la auditoría tradicional como en este nuevo producto. Esto se denominaría tareas de mantenimiento de la auditoría y se debe realizar a través de la programación de script que se ejecuten diariamente

Una vez auditada toda la actividad de la BBDD que se crea conveniente, se puede determinar qué datos son más sensibles como para establecer alertas tanto de advertencias como críticas sobre estos datos que han sido recogidos en la tabla AUD (como ya se ha explicado en la auditoría tradicional) en la máquina fuente.

A continuación en la ilustración 14 aparece una tabla en la cual se recogen las alertas que se han ido generando debido a la actividad de la BBDD. El significado que adquieren las alertas es el siguiente:

Eligiendo una al azar, como podría ser la tercera alerta ‘GRANT da permiso’ significa que la sentencia auditada en la cual se registran todos los usuarios que realizan la actividad de dar permisos no sólo va a ser auditada sino que también se va a registrar como alerta. La diferencia entre una actividad únicamente auditada y una actividad auditada y luego determinada como alerta es que esta actividad se transfiere inmediatamente al servidor apareciendo en la página principal de la interfaz del producto mediante unos gráficos como alarma y quedando registrada en un apartado diferente, en el cual se encuentra toda la actividad auditada y registrada como alarma. Sin embargo si sólo es auditada se transfiere según se encuentre determinado, en este caso una vez al día.

Nombre de Alerta	Descripción	Origen de Auditoría	Tipo de Origen de Auditoría	Categoría de Eventos de Auditoría	Eliminar
insert ana	cuando ana inserta	RECO1	ORCLDB	DATA ACCESS	
AUDIT SYSTEM	AUDITA LA EXITENCIA DE CONEXION	RECO1	ORCLDB	AUDIT	
GRANT da permisos	GRANT da permisos a otros usuarios	RECO1	ORCLDB	ROLE AND PRIVILEGE MANAGEMENT	
HR Delete Countries	HR Realiza delete en la tabla Countries	RECO1	ORCLDB	DATA ACCESS	
HR Delete Jobs	HR Realiza delete en la tabla Jobs	RECO1	ORCLDB	DATA ACCESS	

Ilustración 14: Creación de Alertas Audit Vault

En la ilustración 15 se encuentra registrada una alarma crítica a partir del dato de auditoría que fue recogido inicialmente en la BBDD fuente y a continuación pasó inmediatamente al servidor Audit Vault en el cual se encuentran las alarmas. De esta forma queda registrada la información para posteriores informes.

El resto de los datos de auditoría como ya se ha dicho se trasminen al servidor cada 24 horas, porque ha sido determinado por el administrador de la auditoría.

Critical Alerts

Filas 15 Ir

El texto de la fila contiene 'TABLESPACE'

Nombre de Alerta	Objeto	Evento	Categoría de Eventos	Usuario
TABLESPACE	PARABORRAR2	CREATE TABLESPACE	SYSTEM MANAGEMENT	JENIFER

Ilustración 15: Vista de Alertas Críticas, Audit Vault.

Para que aparezca exclusivamente la alerta generada sobre un tablespace como es en este caso, se ha hecho una búsqueda con dicha palabra. De esta manera se puede observar rápidamente todas las acciones determinadas como alertas que se hayan producido en la BBDD sobre un tablespace. Esta posibilidad genera gran rapidez en la búsqueda de cualquier tipo de actividad, ya que ésta se puede realizar tanto en las tablas de alertas como en las de los datos simplemente auditados.

4.3 Informe de Auditoría general

Una vez ejecutada la Auditoría informática se procede a la realización de un informe final, que será el exponente principal de la calidad del trabajo realizado.

Por lo tanto el informe final debe estructurarse claramente en tres partes: preparación/planificación, situación actual y diagnóstico, y además y por último las recomendaciones que se crean convenientes.

En esta investigación, la situación actual y diagnóstico se basan en si las BBDD pertenecientes a la URJC soportan el producto Audit Vault. Además mediante su estudio, analizar los beneficios que aporta utilizar este producto pero también las carencias o inconvenientes que se le han encontrado, considerándose realizar ciertas recomendaciones a Oracle como posibles modificaciones en posteriores versiones que facilitarían su uso incluyendo más utilidades.

Los datos obtenidos en la investigación se obtienen a partir del siguiente script ejecutándose en el cron, y enviando su salida a un log cada minuto.

```
Topa.sh
   #!/bin/sh
   date
   ps axo comm,user,cputime,pcpu,pmem,start_time | grep avaudcoll
```

Se realiza el mismo script para cada uno de los diferentes procesos que más consumen en la utilización del producto Audit Vault y por lo tanto que más interesan estudiar.

```
   | grep oracle
   | grep emagent
   | grep java
```

Los logs a los cuales se enviará la información obtenida a través de los scripts ejecutados son Avaudcoll.log, Oracle.log, Java.log y Emagent.log correspondiendo cada uno de ellos a los procesos vistos. A partir de estos logs, se analizará y se obtendrá la información pertinente para determinar si la implantación de este producto es o no factible en un entorno de explotación como es el de la Universidad Rey Juan Carlos.

Una parte importante y fundamental a analizar es el traspaso de los datos de la BBDD fuente al servidor, ya que requiere la actividad del colector (proceso Avaudcoll) y por lo tanto un consumo de recursos en la máquina.

4.3.1 Alerta individual

Una vez auditada una actividad que está determinada como alerta se transfiere directamente de la BBDD fuente al servidor sin esperar al traspaso general de datos de auditoría, como ya se ha venido diciendo a lo largo de capítulo. Esta acción consume recursos y es conveniente que sea estudiada.

A continuación aparece el consumo de recursos, en un momento determinado, por cada uno de los procesos que se están analizando.

Se puede observar en el siguiente log (Avaudcoll.log) que el consumo de cpu es nulo y sólo consume un mínimo de memoria por lo que el consumo del colector es constante, sin embargo en el momento que se produce una alerta aparece un pico momentáneo que se produce como ya se ha explicado por el traspaso de datos. Este consumo no supera el 10% de la cpu, por lo que es perfectamente asumible por el entorno.

Avaudcoll.log

Aparece un pico de cpu producido por una alerta a las 08:02.

```
Thu Mar 11 07:59:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
Thu Mar 11 08:00:02 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
Thu Mar 11 08:01:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
Thu Mar 11 08:02:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00  6.7  0.2 08:01
Thu Mar 11 08:03:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
Thu Mar 11 08:04:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
 Thu Mar 11 08:05:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
Thu Mar 11 08:06:01 CET 2010
avaudcoll      ora10g   00:00:35  0.0  0.3 Mar09
```

En este mismo instante, se analizan los datos registrados en Oracle.log

En este log aparecen todos los procesos de Oracle que hay en la máquina ejecutándose en este determinado momento. Es importante determinar el consumo total que tienen todos los procesos de Oracle cuando se produce una actividad tan crítica como el traspaso de datos.

Debido a la alerta se produce un pico en la cpu apareciendo un consumo elevado de dos procesos pero es algo momentáneo y por lo tanto asumible tanto por la máquina utilizada para las pruebas como para la máquina instalada en el entorno de explotación de la universidad.

Si hay algún pico en la cpu generado por los procesos de Oracle, normalmente es debido a un único proceso y no a dos. Este segundo pico viene generado por la aparición de la alerta .

Oracle.log

Thu Mar 11 08:02:01 CET 2010

```
oracle      ora10g  00:04:50  0.1  0.7 Mar09
oracle      ora10g  00:00:17  0.0  0.3 Mar08
oracle      ora10g  00:00:00  0.0  0.2 Mar08
oracle      ora10g  00:00:00  0.0  1.2 Mar08
oracle      ora10g  00:00:14  0.0  3.0 Mar08
oracle      ora10g  00:00:18  0.0  0.4 Mar08
oracle      ora10g  00:00:57  0.0  0.5 Mar08
oracle      ora10g  00:00:26  0.0  2.4 Mar08
oracle      ora10g  00:00:00  0.0  0.4 Mar08
oracle      ora10g  00:04:37  0.1  1.0 Mar08
oracle      ora10g  00:00:38  0.0  1.5 Mar08
oracle      ora10g  00:00:27  0.0  0.5 Mar08
oracle      ora10g  00:00:00  0.0  0.2 Mar08
oracle      ora10g  00:00:00  0.0  0.2 Mar08
oracle      ora10g  00:00:01  0.0  0.5 Mar08
oracle      ora10g  00:00:06  0.0  0.5 Mar08
oracle      ora10g  00:00:00  0.0  0.2 Mar08
oracle      ora10g  00:03:34  0.0  5.5 Mar08
oracle      ora10g  00:00:08  0.0  1.6 Mar08
oracle      ora10g  00:00:00  0.0  0.2 Mar08
oracle      ora10g  00:03:26  0.0  2.2 Mar08
oracle      ora10g  00:00:42  0.0  1.2 Mar08
oracle      ora10g  00:00:00 58.0  0.3 08:02
oracle      ora10g  00:00:00 31.0  0.3 08:02
oracle      ora10g  00:00:00  0.0  0.8 Mar10
 oracle      ora10g  00:00:00  0.0  0.2 Mar10
```

Se ve como en el siguiente minuto ya no hay picos de consumo en la cpu. Esto se debe a que los datos de auditoría englobados en la alerta ya han sido trasladados al servidor.

Oracle .log

Thu Mar 11 08:03:01 CET 2010

```
oracle      ora10g  00:04:51  0.1  0.7 Mar09
oracle      ora10g  00:00:17  0.0  0.3 Mar08
oracle      ora10g  00:00:00  0.0  0.2 Mar08
oracle      ora10g  00:00:00  0.0  1.2 Mar08
```

```
oracle       ora10g   00:00:14  0.0  3.0 Mar08
oracle       ora10g   00:00:18  0.0  0.4 Mar08
oracle       ora10g   00:00:57  0.0  0.5 Mar08
oracle       ora10g   00:00:26  0.0  2.4 Mar08
oracle       ora10g   00:00:00  0.0  0.4 Mar08
oracle       ora10g   00:04:37  0.1  1.0 Mar08
oracle       ora10g   00:00:38  0.0  1.5 Mar08
oracle       ora10g   00:00:27  0.0  0.5 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:00:01  0.0  0.5 Mar08
oracle       ora10g   00:00:06  0.0  0.5 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:03:34  0.0  5.5 Mar08
oracle       ora10g   00:00:08  0.0  1.6 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:03:26  0.0  2.2 Mar08
oracle       ora10g   00:00:42  0.0  1.2 Mar08
oracle       ora10g   00:00:00  0.0  0.3 08:03
oracle       ora10g   00:00:00  0.0  0.3 08:03
oracle       ora10g   00:00:00  0.0  0.8 Mar10
oracle       ora10g   00:00:00  0.0  0.2 Mar10
```

El consumo del proceso java se mantiene constante incluso cuando se produce el traspaso de datos debido a la alerta, por lo que este proceso no afecta al correcto funcionamiento de la máquina.

Java.log

```
Thu Mar 11 08:01:01 CET 2010
java         ora10g   00:02:03  0.0  1.3 Mar09
Thu Mar 11 08:02:01 CET 2010
java         ora10g   00:02:03  0.0  1.3 Mar09
Thu Mar 11 08:03:01 CET 2010
java         ora10g   00:02:03  0.0  1.3 Mar09
Thu Mar 11 08:04:01 CET 2010
java         ora10g   00:02:03  0.0  1.3 Mar09
Thu Mar 11 08:05:01 CET 2010
java         ora10g   00:02:03  0.0  1.3 Mar09
Thu Mar 11 08:06:01 CET 2010
java         ora10g   00:02:03  0.0  1.3 Mar09
```

Al igual que ocurre con el proceso java, el consumo de emagent tanto de memoria como de cpu es constante y mínimo durante todo el tiempo que este proceso se encuentra en ejecución.

Emagent.log

```
Thu Mar 11 07:59:01 CET 2010
emagent      ora10g   00:06:09  0.1  0.7 Mar08
Thu Mar 11 08:00:02 CET 2010
emagent      ora10g   00:06:09  0.1  0.7 Mar08
Thu Mar 11 08:01:01 CET 2010
```

```
emagent      ora10g   00:06:10  0.1  0.7 Mar08
Thu Mar 11 08:02:01 CET 2010
emagent      ora10g   00:06:10  0.1  0.7 Mar08
Thu Mar 11 08:03:01 CET 2010
emagent      ora10g   00:06:10  0.1  0.7 Mar08
Thu Mar 11 08:04:01 CET 2010
emagent      ora10g   00:06:10  0.1  0.7 Mar08
Thu Mar 11 08:05:01 CET 2010
emagent      ora10g   00:06:10  0.1  0.7 Mar08
Thu Mar 11 08:06:01 CET 2010
emagent      ora10g   00:06:10  0.1  0.7 Mar08
```

4.3.2 Traspaso de todos los datos de auditoría

En el traspaso de todos los datos de auditoría de la BBDD fuente al servidor a una hora determinada del día ocurre algo muy similar a lo explicado en la alerta individual, es decir, el consumo de memoria sigue siendo pequeño y el consumo de cpu ronda alrededor del 10% de la máquina no superando el 15%. La diferencia es que el tiempo de transmisión de datos es mayor debido esencialmente a que el volumen de datos que transfiere el recolector también es mayor. En este caso el tiempo estimado en el que el recolector traspasa los datos de una máquina a otra es de unos cuarenta minutos aproximadamente.

Por lo tanto mirando los datos obtenidos a continuación se puede observar que avaudcoll es un proceso recolector constante con un consumo aceptable.

Avaudcoll.log

```
Fri Mar 12 13:00:01 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00  7.0  0.2 12:59
Fri Mar 12 13:01:02 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00  6.5  0.2 13:01
Fri Mar 12 13:02:01 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00 14.0  0.2 13:02
Fri Mar 12 13:03:01 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00 13.0  0.2 13:03
Fri Mar 12 13:04:01 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00 13.0  0.2 13:04
Fri Mar 12 13:05:01 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00 13.0  0.2 13:05
Fri Mar 12 13:06:01 CET 2010
avaudcoll    ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll    ora10g   00:00:00 14.0  0.2 13:06
```

```
Fri Mar 12 13:07:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00 13.0  0.2 13:07
Fri Mar 12 13:08:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00 12.0  0.2 13:08
Fri Mar 12 13:09:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00  6.5  0.2 13:08
Fri Mar 12 13:10:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00  6.5  0.2 13:09
Fri Mar 12 13:11:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00  6.5  0.2 13:10
Fri Mar 12 13:12:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00  6.5  0.2 13:11
Fri Mar 12 13:13:01 CET 2010
avaudcoll      ora10g   00:00:54  0.0  0.3 Mar09
avaudcoll      ora10g   00:00:00  6.5  0.2 13:12
```

Se puede observar en los siguientes datos recogidos que aunque se esté produciendo la transmisión de datos de auditoría, los procesos Oracle se mantienen constantes, produciéndose cada cierto tiempo picos de consumo en la cpu. En este caso aparecen dos minutos en los cuales se está produciendo el traspaso de datos y se ve que no hay un aumento significativo en su consumo.

Oracle.log

```
Fri Mar 12 13:05:01 CET 2010
oracle         ora10g   00:07:59  0.1  0.7 Mar09
oracle         ora10g   00:00:20  0.0  0.3 Mar08
oracle         ora10g   00:00:00  0.0  0.2 Mar08
oracle         ora10g   00:00:00  0.0  1.2 Mar08
oracle         ora10g   00:00:18  0.0  3.2 Mar08
oracle         ora10g   00:00:24  0.0  0.4 Mar08
oracle         ora10g   00:01:15  0.0  0.5 Mar08
oracle         ora10g   00:00:34  0.0  2.6 Mar08
oracle         ora10g   00:00:00  0.0  0.4 Mar08
oracle         ora10g   00:06:25  0.1  1.1 Mar08
oracle         ora10g   00:00:50  0.0  1.5 Mar08
oracle         ora10g   00:00:36  0.0  0.5 Mar08
oracle         ora10g   00:00:00  0.0  0.2 Mar08
oracle         ora10g   00:00:00  0.0  0.2 Mar08
oracle         ora10g   00:00:02  0.0  0.5 Mar08
oracle         ora10g   00:00:09  0.0  0.5 Mar08
oracle         ora10g   00:00:00  0.0  0.2 Mar08
oracle         ora10g   00:04:24  0.0  5.6 Mar08
```

```
oracle       ora10g   00:00:08  0.0  1.6 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:04:43  0.0  2.4 Mar08
oracle       ora10g   00:00:57  0.0  1.2 Mar08
oracle       ora10g   00:00:42  0.0  5.1 Mar10
oracle       ora10g   00:00:02  0.0  1.2 Mar10
Fri Mar 12 13:06:01 CET 2010
oracle       ora10g   00:07:59  0.1  0.7 Mar09
oracle       ora10g   00:00:20  0.0  0.3 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:00:00  0.0  1.2 Mar08
oracle       ora10g   00:00:18  0.0  3.2 Mar08
oracle       ora10g   00:00:24  0.0  0.4 Mar08
oracle       ora10g   00:01:15  0.0  0.5 Mar08
oracle       ora10g   00:00:34  0.0  2.6 Mar08
oracle       ora10g   00:00:00  0.0  0.4 Mar08
oracle       ora10g   00:06:25  0.1  1.1 Mar08
oracle       ora10g   00:00:50  0.0  1.5 Mar08
oracle       ora10g   00:00:36  0.0  0.5 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:00:02  0.0  0.5 Mar08
oracle       ora10g   00:00:09  0.0  0.5 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:04:24  0.0  5.6 Mar08
oracle       ora10g   00:00:08  0.0  1.6 Mar08
oracle       ora10g   00:00:00  0.0  0.2 Mar08
oracle       ora10g   00:04:43  0.0  2.4 Mar08
oracle       ora10g   00:00:57  0.0  1.2 Mar08
oracle       ora10g   00:00:42  0.0  5.1 Mar10
oracle       ora10g   00:00:02  0.0  1.2 Mar10
```

Una vez visto el consumo del proceso avaudcoll, conociendo por otro lado que tanto el proceso java como emagent como se vio en el punto anterior son constantes en toda su ejecución con un consumo mínimo y que otros procesos Oracle aumentan su consumo en picos puntuales se considera que el uso del producto Audit Vault es asumible por la estructura de la URJC.

Es importante decir que sería conveniente que el traspaso de datos se realizase en un momento en el cual la BBDD tuviera poca actividad y que es de gran importancia auditar estrictamente lo necesario creando las alertas sobre los datos recogidos más significativos y que más importancia tengan sobre la BBDD auditada, ya que el exceso de información provoca falta de claridad en la información.

Capítulo 5 Conclusiones y trabajos futuros

5.1 Logros alcanzados

Principalmente en toda empresa, privada o pública, que posea sistemas de información medianamente estructurados en sistemas informáticos deben someterse a un control estricto de evaluación de eficacia y eficiencia. Para ello, ha aparecido Audit Vault, que es con uno de los productos más completos que actualmente hay en el mercado, es muy innovador por su pantalla gráfica desde la cual se pueden emitir sentencias de auditoría de una manera mucho más sencilla en comparación con la auditoría tradicional, facilitando la tarea tanto al auditor como al administrador de BBDD. Por otro lado este producto genera directamente gráficas e informes, a partir de los datos tomados por el producto sobre la base de datos origen, que resultan mucho más sencillos de interpretar que los que aparecían en antiguas auditorías, ya que para acceder a los datos mediante la auditoría tradicional había que acceder a las tablas en las cuales estuvieran los datos auditados.

Por lo tanto, se apuesta por la facilidad de uso, ahorro de tiempo en el análisis de los informes y una mayor claridad, en definitiva, en cualquier acción de auditoría siendo una vez más elegida a medio plazo (por ser tan innovador) una aplicación gráfica VS sentencias a partir de consolas de texto.

A continuación se muestran los beneficios específicos obtenidos a partir de la implantación de Audit Vault. Éstos son los siguientes:

Simplificar los informes de cumplimiento: Analizar fácilmente los datos de auditoría y tomar medidas de manera puntual con informes listos para usar o información personalizada mediante el único esquema de depósito abierto del sector para la información de auditoría Anteriormente los datos se encontraban en una tabla a la cual había que acceder manualmente para obtener la información y poder realizar los informes.

Detectar las amenazas con rapidez: Detecta rápida y automáticamente las actividades no autorizadas que violan la seguridad y las políticas establecidas mediante alertas tanto críticas como de advertencia. Esto es una funcionalidad del producto sencilla de utilizar

sin embargo en la auditoría tradicional se debía realizar a través de scripts programados a mano. Por lo que su utilización resultaba mucho más trabajosa y tediosa.

Reducir los costos de TI con políticas de auditoría: Administrar centralmente las configuraciones de auditoría en todas las BBDD desde una única consola facilitando de esta manera el seguimiento de todas las BBDD auditadas. Sin embargo antes había que acceder una a una a las máquinas que estuvieran siendo auditadas.

Recopilar y consolidar los datos de auditoría de manera transparente: Recopilar los datos de auditoría de manera puntual a través de diversos sistemas. Cuando se producen las alertas es instantáneo el traspaso de datos, el resto se realiza a la hora establecida por el administrador.

Ofrecer un repositorio seguro: Evita que el culpable de una infracción cometida sobre la BBDD elimine su rastro. Si su acción está registrada como alarma, la actividad auditada se traspasa al servidor, por lo que aunque el infractor acceda a las tablas de auditoría y elimine los datos como podía ocurrir en la auditoría tradicional, su actividad ha quedado registrada en otra máquina (debido a que los datos auditados de todas las máquinas se almacenan en otra diferente.).

5.2 Dificultades

Oracle es un producto de pago, el cual necesita certificación para ser usado.

El mercado de auditoría de BBDD está en una etapa muy inicial de desarrollo. Esperar a encontrar el producto más nuevo y maduro probablemente sólo conduce a la decepción, ya que a la hora de su utilización se encuentran aspectos susceptibles de mejoras. La decisión de la elección debe basarse en los requisitos que determinan qué es lo que se necesita tener y a partir de ahí se puede buscar una herramienta que cumpla con estos requisitos. Esa es la mejor manera de tener lo que realmente se necesita.

En definitiva, las dificultades encontradas han estado relacionadas con la falta de información del producto, debido a la gran novedad que presenta el terreno de auditorías. Por lo que prácticamente para cualquier tipo de incidencia tenida durante el proceso de instalación de dicho producto, la única ayuda que se ha podido obtener ha estado localizada en su propio soporte, Metalink. Este soporte es de pago, y se encuentra instalado y financiado por la universidad, sin el cual es prácticamente imposible resolver cualquier tipo de dudas, ya que como se ha dicho antes, la información de Audit Vault es escasa, y en los foros se ve el desconocimiento que

todavía impera acerca de él. Esto llega a ralentizar muchísimo todo el proceso de instalación.

Con respecto a la documentación, se encuentra en los manuales oficiales en los cuales falta bastante información con respecto al manejo de la interfaz, como por ejemplo puede ser lo referente a la configuración de políticas FGA, borrado de políticas, etc.

Añade complejidad a la arquitectura actual del sistema. Puesto que hay que mantener en cada máquina de BBDD un agente y una máquina con otra BBDD destinada a recoger la información.

Otra cuestión a tener en cuenta sería quién audita la propia BBDD que guarda la auditoría.

Por lo tanto facilita a nivel de auditoría y se complica su mantenimiento por su complejidad. Esto es debido a la utilización de la plataforma software Oracle Application Server que consta principalmente de Oracle HTTP Server (basado en Apache HTTP Server) y OC4J (OracleAS Contenedores para Java EE), que implementa Java EE basado en aplicaciones (da apoyo a la arquitectura orientada a servicios).

Aparte de las dificultades encontradas se aportan a continuación unas **críticas** realizadas directamente sobre el producto, ya que se han observado carencias a la hora de su utilización.

Se puede considerar un sistema de alertas incompleto comparándolo con otros sistemas de Oracle como puede ser GRID CONTROL, ya que sin ser un producto de auditoría éste tiene la posibilidad de enviar correos electrónicos de advertencia y Audit Vault no tiene la posibilidad de enviar por correo electrónico ningún tipo de alerta, ni siquiera las alertas críticas seleccionadas. Para que se envíe un correo debe ser programado. Por lo tanto la interfaz carece de esa opción. Esto implica una dependencia del producto, al tener que estar continuamente revisando su interfaz gráfica para ver si se ha producido alguna alerta.

Se debe realizar un mantenimiento local de los datos auditados en la BBDD fuente. Esto es debido a que la aplicación transfiere los datos de auditoría a su Warehouse (almacen de datos) para su tratamiento, pero una vez transferidos al servidor

no da la posibilidad de borrarlos. El borrado es esencial, ya que los datos se almacenan en una tabla en la BBDD fuente y ésta como es lógico tiene espacio finito.

Por otro lado, aunque hay separación de roles, sigue habiendo una dependencia lógica del administrador de la BBDD con el auditor.

La creación de los colectores es de manera manual, en la interfaz aparecen los colectores que hay y en qué estado se encuentran pero no da la posibilidad de crearlos. Esto se debe hacer a través de comandos.

5.3 Posibles trabajos futuros

Los trabajos futuros que se pueden llevar a cabo a partir de la investigación realizada de la implantación del producto Audit Vault, podrían dirigirse al estudio de la verdadera compatibilidad que presenta este producto en un entorno de explotación de BBDD que no pertenezcan a Oracle. Por lo que habría que realizar el estudio tanto de instalación del agente en la BBDD fuente (diferente a Oracle), como el estudio de su uso con respecto a la auditoría realizada anteriormente en dicha BBDD y los beneficios y desventajas que aporta este producto en este tipo de BBDD.

Otra investigación que se podría llevar a cabo sería realizar la auditoría eligiendo como colector OSAUD, para ver si de esta manera la Universidad Rey Juan Carlos podría utilizar únicamente este producto, en vez de utilizar uno dedicado a los datos de las BBDD y otro tipo de productos al SO.

Bibliografía

[Acha, J.J (1994)] "Auditoría Informática en la empresa". Paraninfo.

[Administrator's Guide (2008)] "Oracle® Database, Administrator's Guide Release 10.2.3 E11059-03, July 2008"

[Alonso, G. (1989)] "Auditoría Informática" Díaz de Santos.

[Auditor's Guide (2007)] "Oracle® Audit Vault Auditor's Guide 10g Release 2 (10.2.2) B28853-02, August 2007"

[Auditor's Guide (2009)] "Oracle® Audit Vault Auditor's Guide Release (10.2.3.1) E13842-02, May 2009"

[Best Practices (2007)] "Oracle Audit Vault, Best Practices, Nov 2007."

[Bob Bryla, kevin Loney (2005)] "Oracle database 10g DBA Handbook, Manage a Robust, Sacalable, and Highly Available Oracle Database." Oracle Press ,Osborne, Mc Braw Hill.

[Bob Bryla, kevin Loney (2008)] "Oracle database 11g DBA Handbook, Administer a Scalable, Secure Oracle Enterprise Database." Oracle Press Osborne, Mc Braw Hill.

[De Pablos, C. et all 2006] Carmen de Pablos Heredero, José Joaquín López-Hermoso Agius, Santiago Martín-Romo Romero, Sonia Medina Salgado, Antonio Montero Navarro, Juan José Nájera Sanchez.(2006). "Dirección y gestión de los sistemas de información en la empresa, una visión investigadora" ,ESIC, Madrid.

[kevin Loney, Marlene Theriault (2002)] "ORACLE9i Manual del administrador (técnicas de gestión de bases de datos Oracle robustas y de alto rendimiento)", edición Oracle Press ofical, Osborne, McGraw-Hill.

[Michael Abbey, Mike Corey, Ian Abramson (2002)] "ORACLE9i Guía de aprendizaje, aprenda lo esencial de Oracle9i, edición Oracle Press ofical, Osborne, McGraw-Hill.

[Paiattini, M; Del Peso, E. (2000)] "Auditoría Informática. Un enfoque práctico". RAMA

[Sample Schemas (2008)] "Oracle® Database, Sample Schemas 11gRelease 1(11.1) B28328-03, July 2008"

Web Relacionadas:

http://www.ag-protecciondatos.es

http://www.macroseguridad.com

http://www.lopdplan.com

http://www.Audit.gov.tw/span/span2-2.htm

http://www.oracle.com

http://www.gatner.com

Anexos

ANEXO I Instalación Audit Vault (servidor) en una máquina linux

1 OBJETIVOS

El objetivo de este documento es mostrar la correcta instalación del servidor de Oracle Audit Vault en una máquina linux.

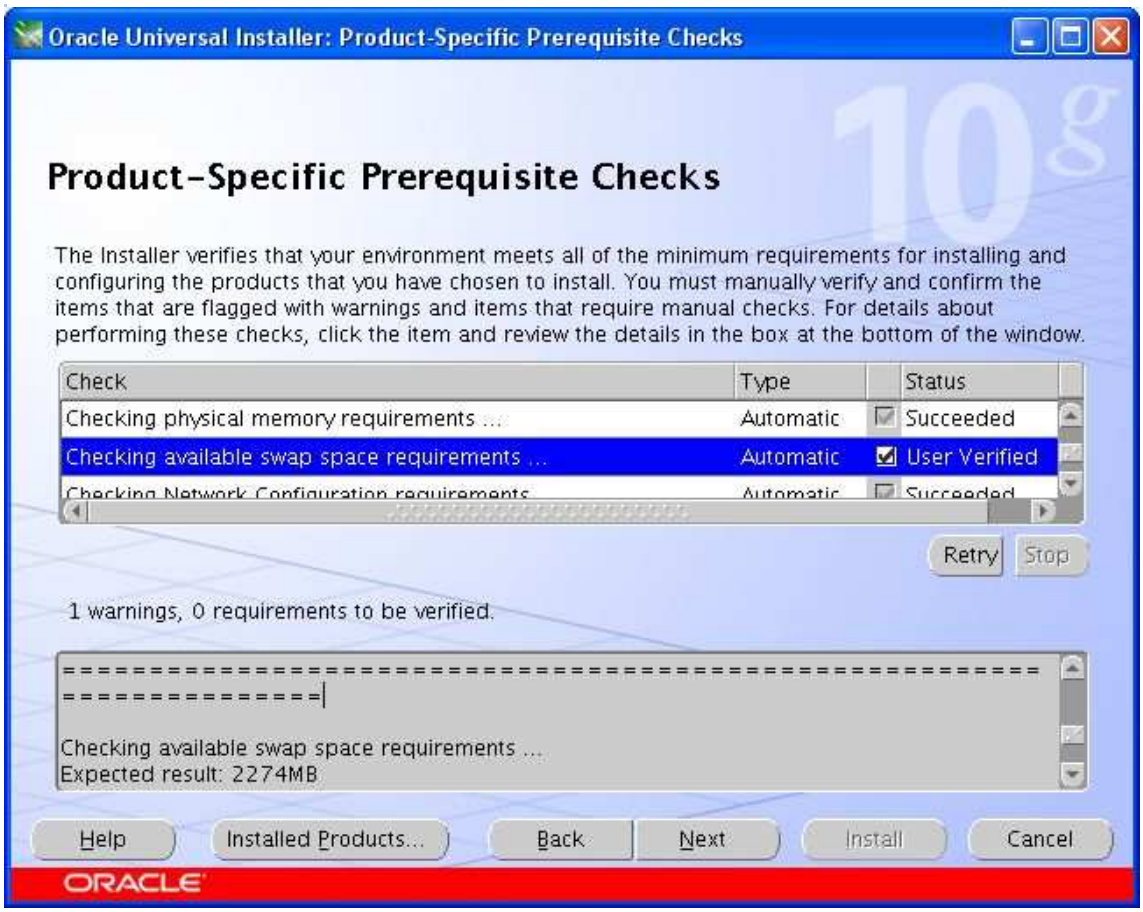
2.INSTALACION AUDIT VAULT (SERVIDOR)

En primer lugar se debe conseguir el CD de instalación o descargarlo de Internet en el caso de que se encuentre registrado a Oracle y copiar el software al servidor. A continuación después de situarse en la ubicación de dicho software se ejecuta runInstaller: **./runinstaller**

Una vez iniciado el instalador aparece la siguiente pantalla, es la pantalla inicial de la instalación del servidor del producto Audit Vault de oracle. Aquí se puede seleccionar el tipo de instalación que se quiere, puede ser tanto una instalación básica, como la que se va a realizar en este caso, como una instalación avanzada.

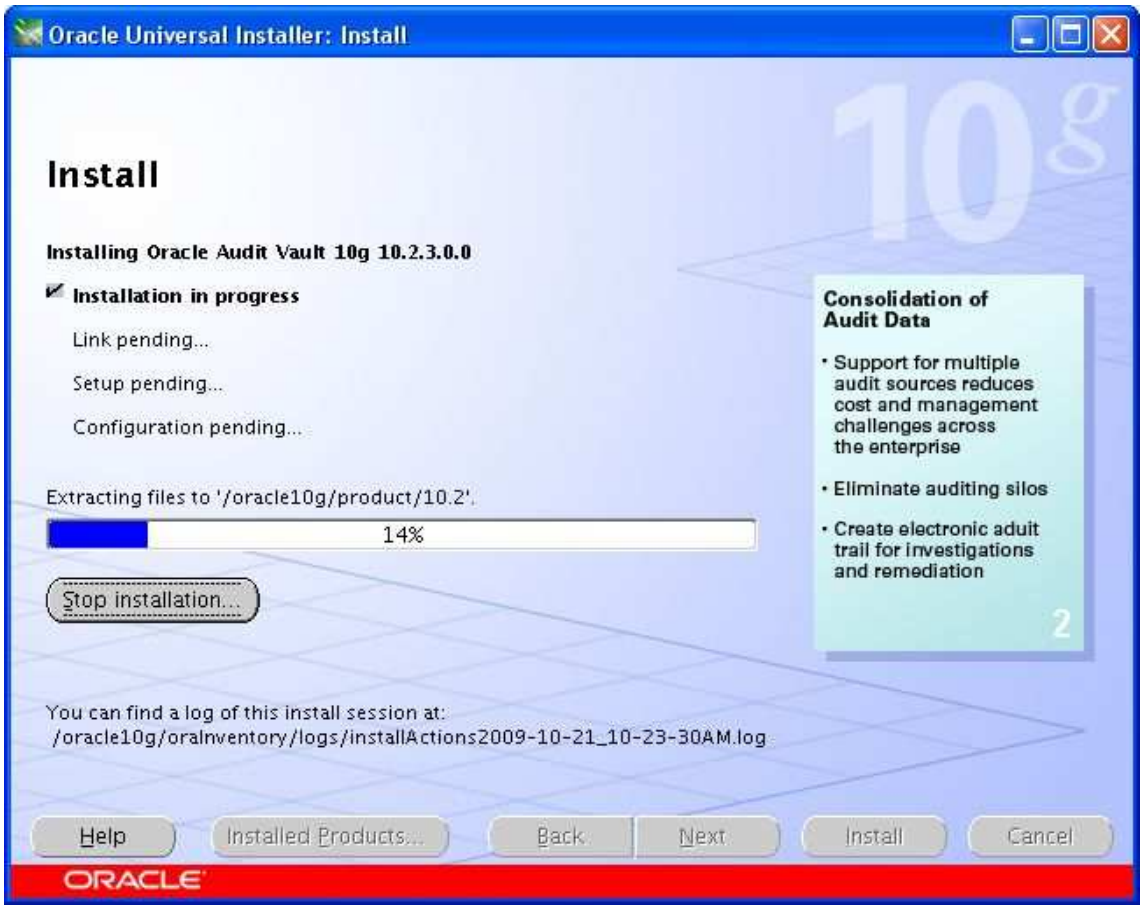
En la siguiente pantalla se chequean los prerrequisitos del producto específico. El instalador verifica que el entorno cumple todos los requisitos mínimos para la correcta instalación y la configuración del producto que se va a instalar.

Se supone que cuando se va a instalar un cliente de AV ya existe un usuario Oracle y las variables del kernel… etc. están bien seteadas.

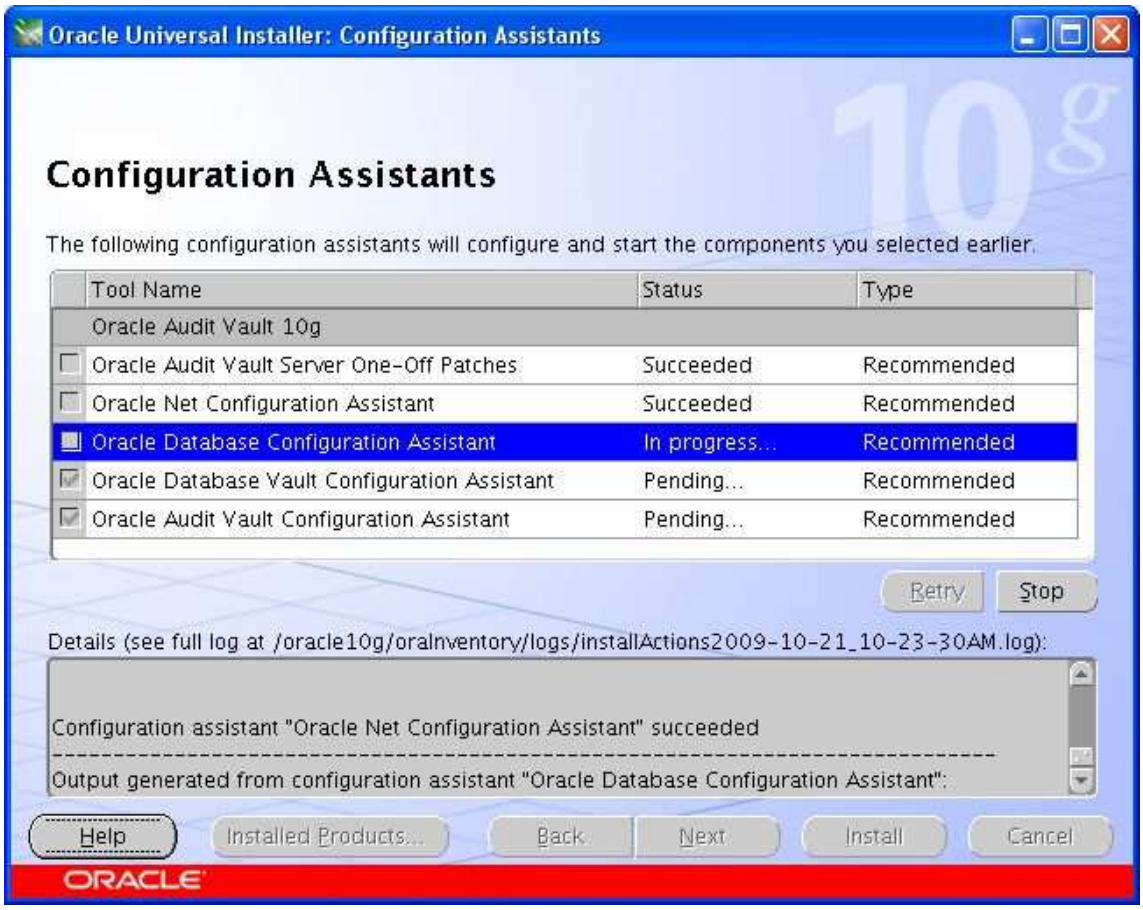


Si los prerrequisitos son correctos se procede a la instalación del producto. La siguiente pantalla correspondería al resumen de la instalación, una vez revisado se pulsa a instalar.

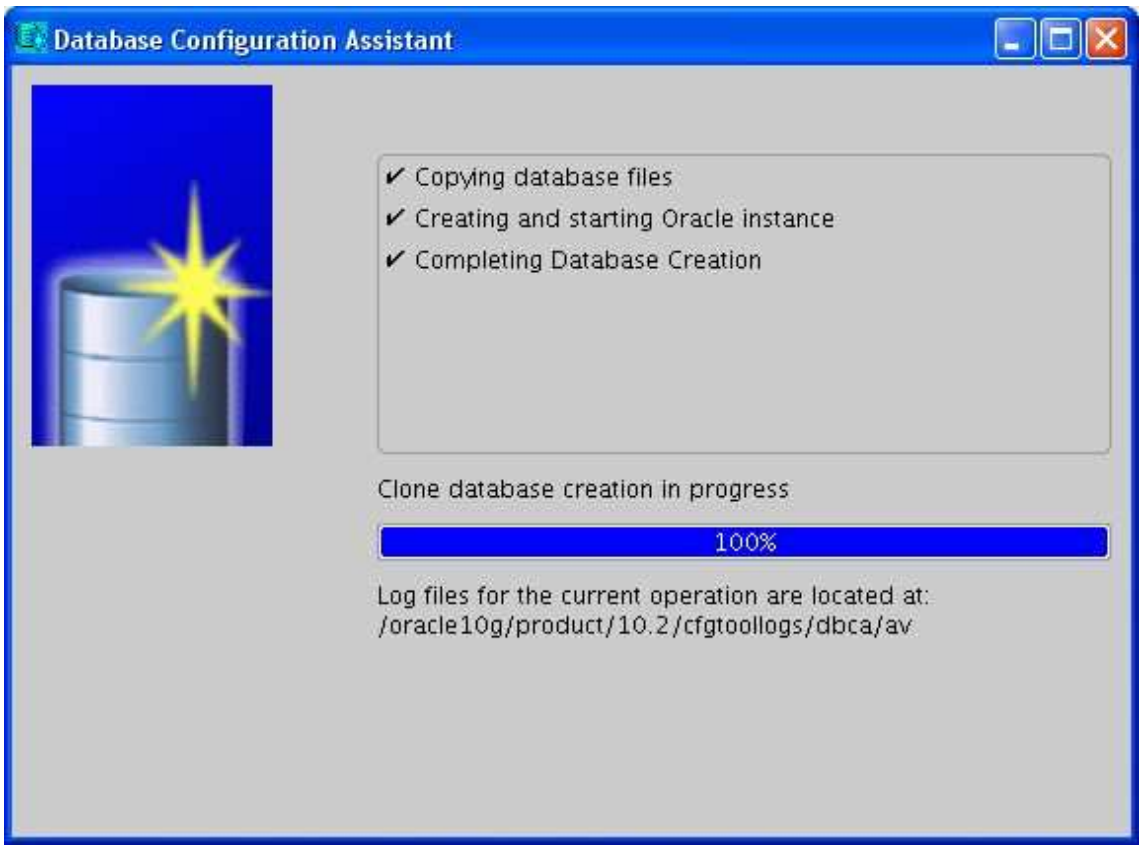
En esta pantalla aparece el progreso de instalación. También tiene una opción debajo de la barra de progreso con la cual se puede parar la instalación



Al final de la instalación nos aparecerá una ventana en la cual el producto realizará algunas configuraciones para el buen funcionamiento del producto.

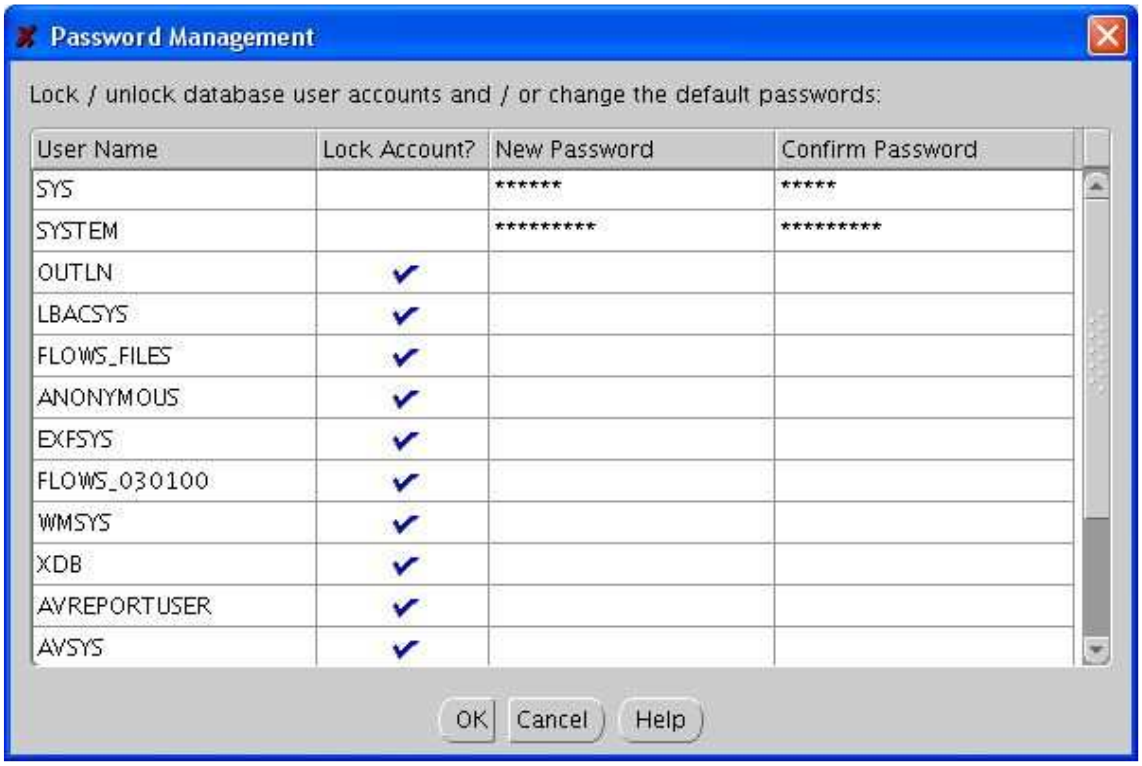


A continuación aparece la configuración del asistente.

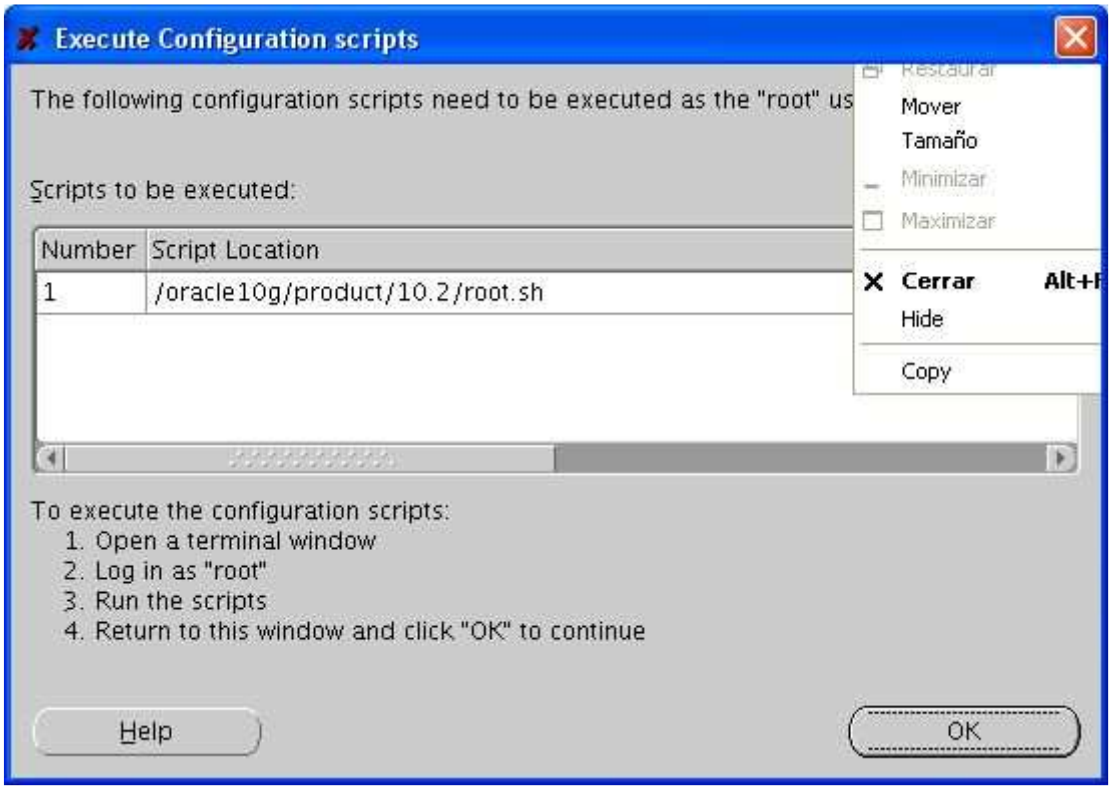


En la configuración del asistente aparece una ventana en la cual se pueden gestionar las contraseñas. Si se clicka en password Management se permitirá realizar esta acción.

En este caso se modifican las contraseñas correspondientes a los usuarios SYS y SYSTEM, se pide la nueva contraseña y posteriormente se confirma. Una vez finalizada la acción se clicka en ok.



Antes de la finalización de la aplicación es necesaria la ejecución del script que aparece especificado en la siguiente pantalla.



Una vez situados en la ubicación pertinente se realiza la ejecución del root.sh

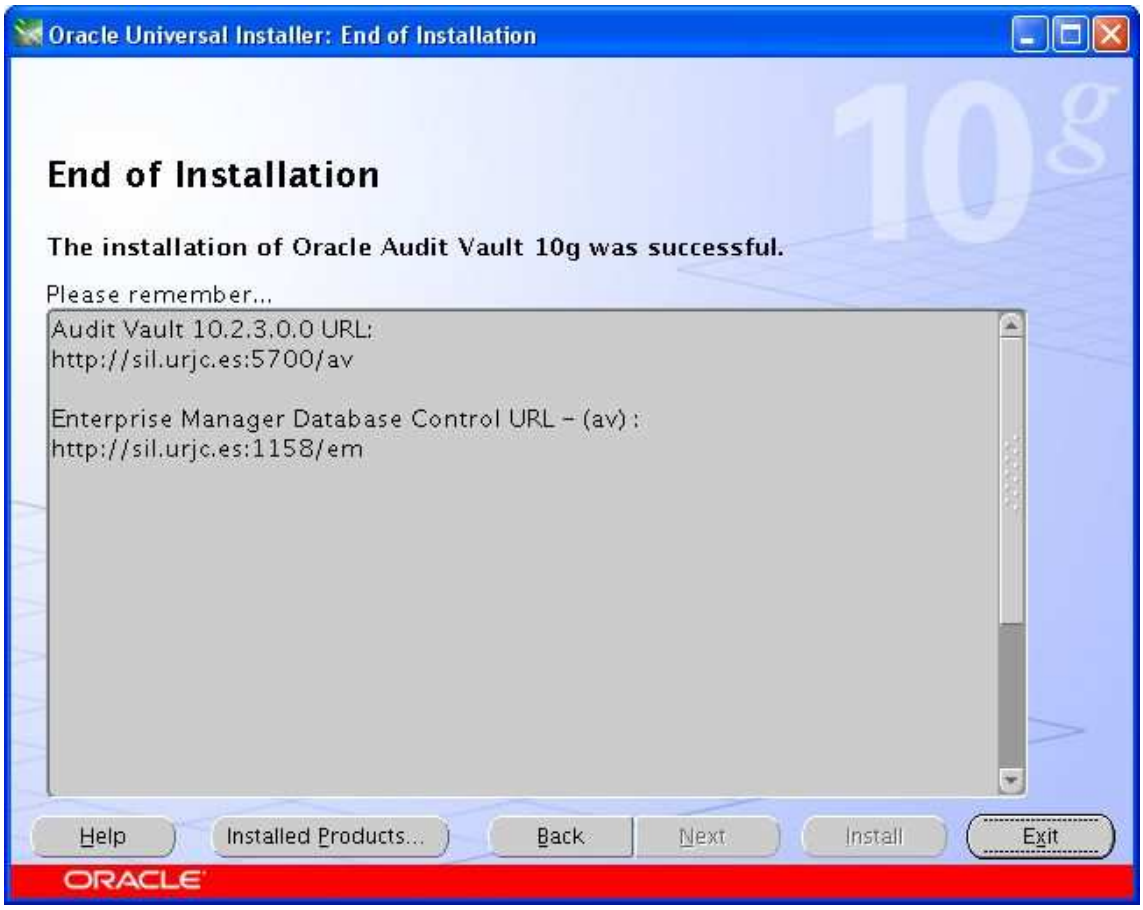
[root@sil 10.2]# sh root.sh
Running Oracle 10g root.sh script...

The following environment variables are set as:
ORACLE_OWNER= ora10g
ORACLE_HOME= /oracle10g/product/10.2

Enter the full pathname of the local bin directory: [/usr/local/bin]:
The file "dbhome" already exists in /usr/local/bin. Overwrite it? (y/n)
[n]: y
Copying dbhome to /usr/local/bin ...
The file "oraenv" already exists in /usr/local/bin. Overwrite it? (y/n)
[n]: y
Copying oraenv to /usr/local/bin ...
The file "coraenv" already exists in /usr/local/bin. Overwrite it? (y/n)
[n]: y
Copying coraenv to /usr/local/bin ...

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.

En este momento la instalación ha finalizado, para salir de la aplicación se pulsa exit.



Antes de finalizar por completo la instalación, aparece una nueva ventana preguntando de nuevo si realmente se quiere salir de la aplicación, en caso afirmativo la instalación se da por terminada, en caso contrario continua abierta.

ANEXO II Instalación Agente Audit Vault en una máquina linux

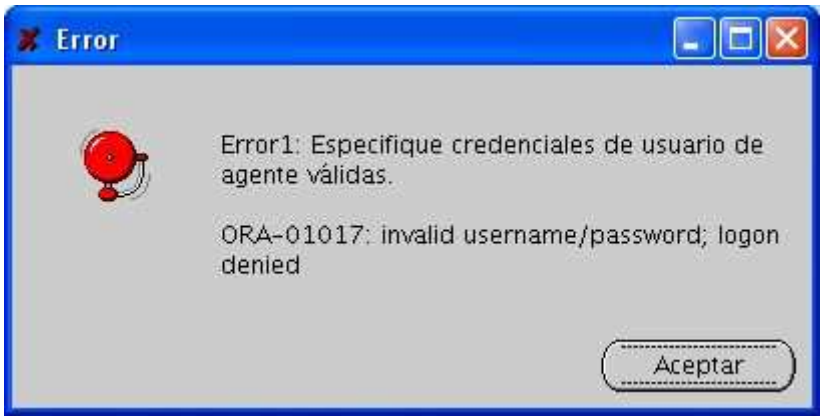
1 OBJETIVOS

El objetivo de este documento es mostrar la correcta instalación del agente de Oracle Audit Vault en una máquina linux.

2 PREINSTALACIÓN AGENTE AUDIT VAULT

Antes de la instalación, se debe realizar un paso fundamental sin el cual no se podría avanzar en la aplicación de instalación ya que aparecería una ventana de error por la cual no se podría continuar.

El error que aparece a continuación se produce en el caso de que no se haya realizado la preinstalación, es decir, que no se haya añadido un agente de Audit Vault y el nombre de un usuario Audit Vault con su correspondiente contraseña. Por lo tanto se deberá cerrar la aplicación y realizar el paso de preinstalación.



Para que no aparezca este error hay que registrar el agente que se va a instalar en el servidor de Audit Vault.

Para registrarlo se utiliza en el servidor de Audit Vault el comando **AVCA**

SINTAXIS

avca add_agent -agentname <agent name> [-agentdesc <desc>] -agenthost <host>

- avca add_agent : Añade el agente al servidor.
- -agentname : Indica el nombre que tendrá el agente.
- [-agentdesc <desc>] : Opcionalmente especifica una descripción del agente.
- -agenthost : Especifica el nombre del agente de la máquina donde el agente será instalado (es la máquina desde la cual nos queremos conectar al servidor de Audit Vault).

Ejemplo de registro de un agente en el servidor previa instalación

```
[ora10g@sil admin]$ avca add_agent -agentname avcofio -agenthost
cofio.urjc.es
AVCA started
Adding agent...
Enter agent user name: avcofio
Enter agent user password:
Re-enter agent user password:
Agent added successfully.
```

El nombre del agente y nombre del usuario del agente no tiene por qué coincidir.

Otro error que se puede evitar en la instalación es el que aparece a continuación, dicho error se obtiene al dar al botón de siguiente después de haber rellenado la pantalla de incio de la aplicación.

Este error se produce porque no reconoce la cadena de conexión, por lo tanto no se puede establecer la conexión. Se puede solucionar añadiendo al listener del servidor una nueva entrada en el SID_LIST. Esta entrada va a corresponder al nuevo servicio sobre el cual acepto las peticiones de la aplicación Audit Vault para el puerto que se encuentra determinado para dicha entrada.

Ejemplo:

```
(SID_DESC =
(GLOBAL_DBNAME = av )
(ORACLE_HOME = /oracle10g/product/10.2)
(SID_NAME = av )
)
```

3 INSTALACION AGENTE AUDIT VAULT (CLIENTE)

En primer lugar se debe conseguir el CD de instalación o descargarlo de Internet en el caso de que se encuentre registrado a Oracle y copiar el software al servidor. A continuación después de situarse en la ubicación de dicho software se ejecuta runInstaller: **./runinstaller**

Esta es la primera pantalla que nos aparece para la instalación del agente de AV. Los datos que se nos piden son los siguientes:

- Nombre del Agente de Audit Vault
- Dirección raíz del Agente Audit Vault
- Nombre de usuario del Agente
- Contraseña de usuario del Agente

Estos nombres tienen que coincidir con los registrados anteriormente en el servidor (ver prerrequisitos de instalación, apartado 2).

A continuación aparece la página de inicio rellenada con los datos correspondientes a cada apartado.

Si no hubiera ningún problema saldría la pantalla de progreso de instalación.

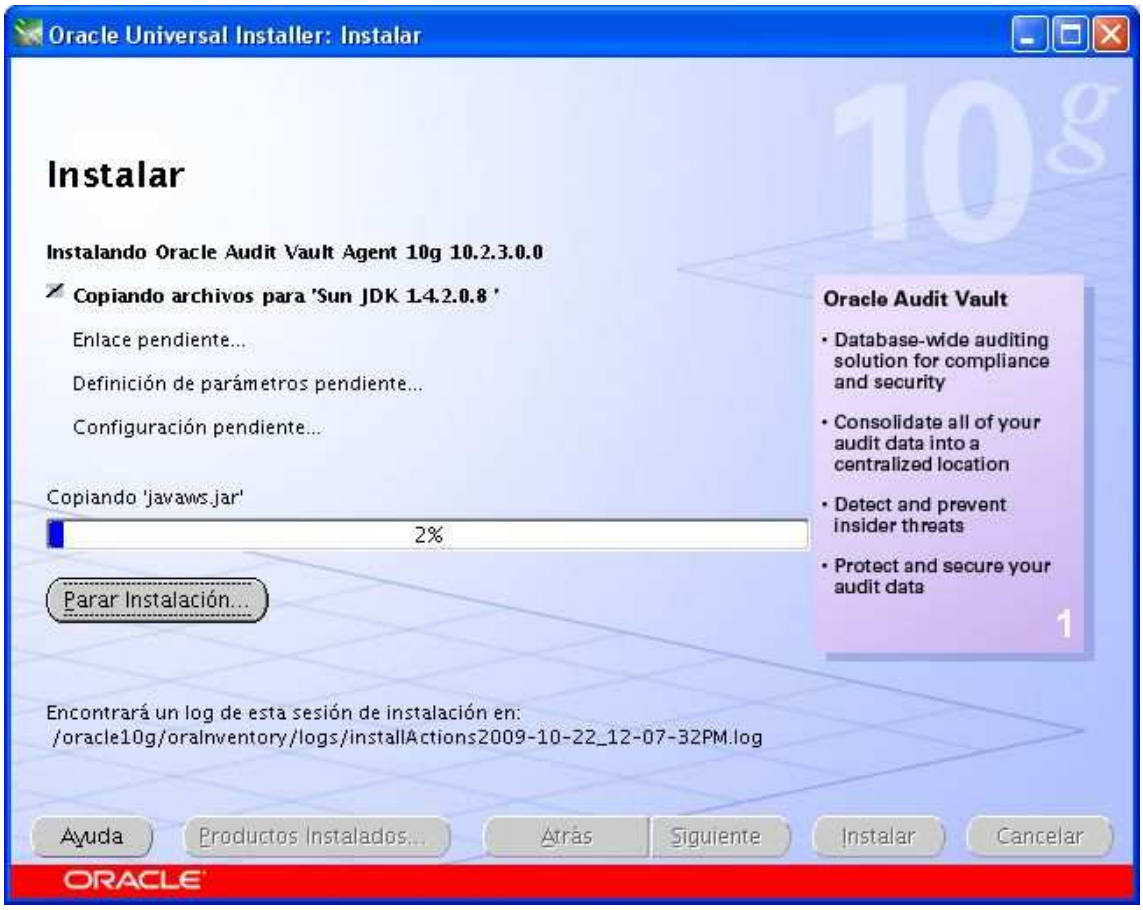
Si los datos han sido introducidos correctamente y no hay ningún problema más pasaríamos a la pantalla de comprobación de prerrequisitos.

En esta pantalla se verifica que el entorno cumple todos los requisitos mínimos para instalar y configurar los productos seleccionados para la instalación.

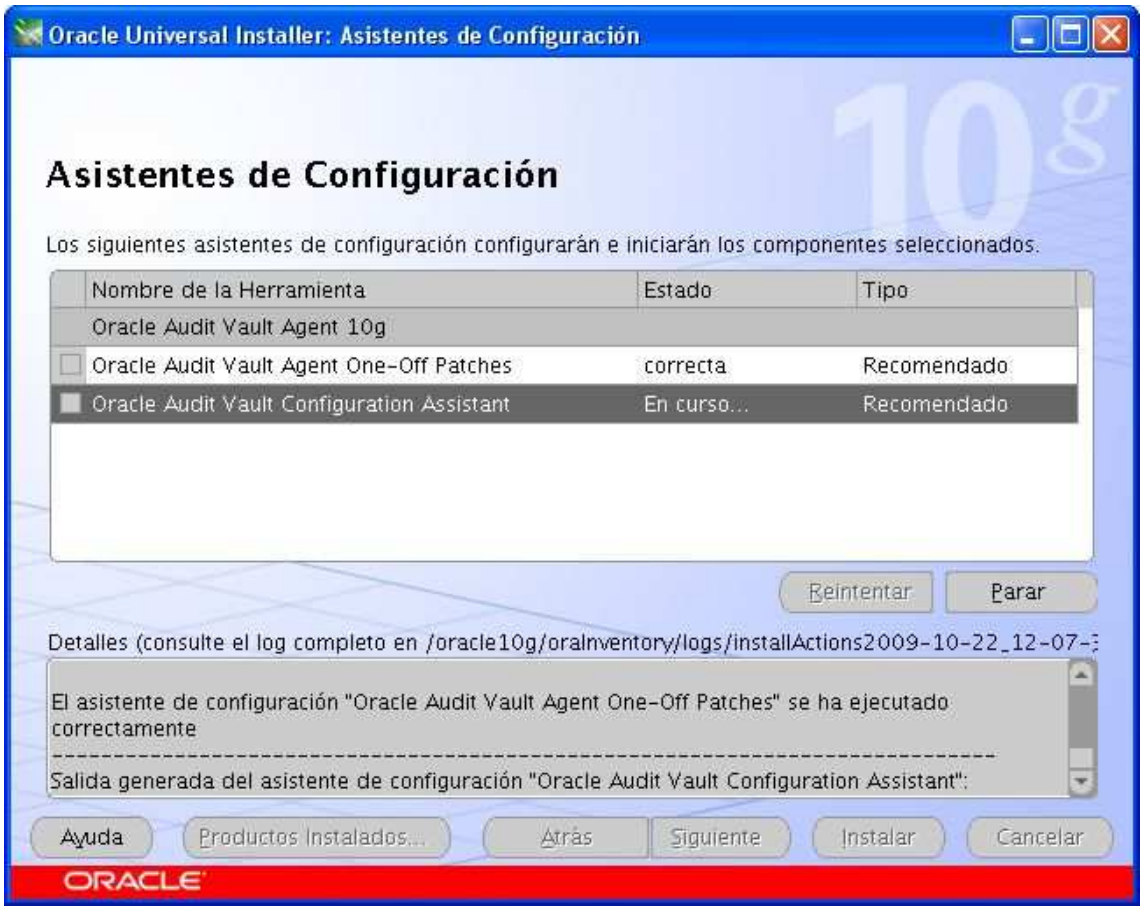
Se supone que cuando vas a instalar un cliente de AV ya existe un usuario Oracle y las varibles del kernel… etc. están bien seteadas.

Si los prerrequisitos son correctos se procede a la instalación del producto. La siguiente pantalla correspondería al resumen de la instalación, una vez revisado se pulsa a instalar.

Se irá viendo la progresión de la instalación en la siguiente pantalla


Oracle Universal Installer: Instalar
Instalar
Instalando Oracle Audit Vault Agent 10g 10.2.3.0.0
Copiando archivos para 'Sun JDK 1.4.2.0.8 '
Enlace pendiente...
Definición de parámetros pendiente...
Configuración pendiente...
Copiando 'javaws.jar'
2%
Parar Instalación...
Oracle Audit Vault
• Database-wide auditing solution for compliance and security
• Consolidate all of your audit data into a centralized location
• Detect and prevent insider threats
• Protect and secure your audit data
Encontrará un log de esta sesión de instalación en:
/oracle10g/oraInventory/logs/installActions2009-10-22_12-07-32PM.log
Ayuda
Productos Instalados...
Atrás
Siguiente
Instalar
Cancelar
ORACLE

Al final de la instalación nos aparecerá una ventana en la cual el producto realizará algunas configuraciones para el buen funcionamiento del producto.

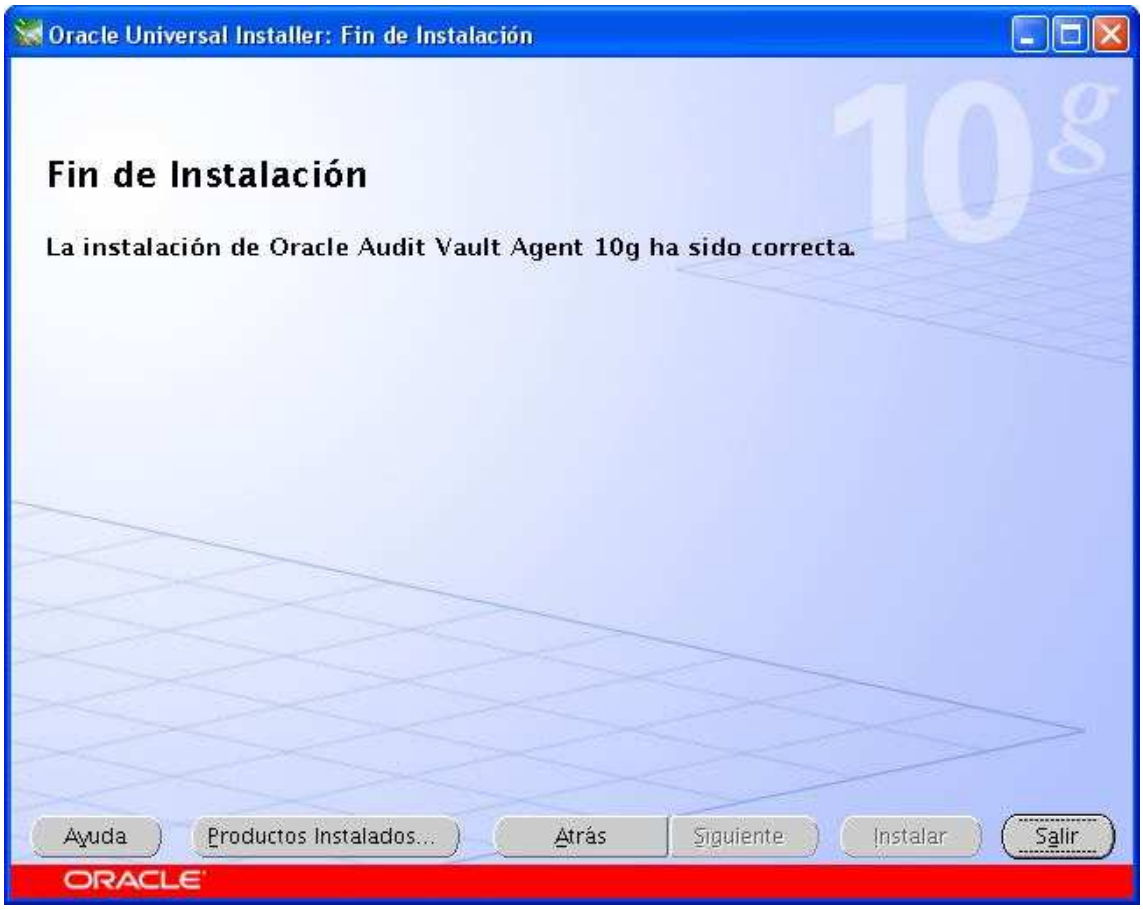


Una vez que el asistente de instalación ha realizado su cometido, aparece la siguiente pantalla, en la cual se encuentra la ubicación del archivo root.sh y las instrucciones de cómo se debe ejecutar.

Se siguen las instrucciones y se ejecuta el script que aparece en la pantalla.

[root@cofio ~]# cd /oracle10g/product/10.2/av/
[root@cofio av]# sh root.sh

La instalación ha llegado a su fin.



Antes de salir de la instalación la aplicación siempre pide confirmación, en caso afirmativo se termina y en caso negativo continua abierta.

ANEXO III Añadir un nuevo host a Audit Vault

PASOS A SEGUIR

1. **[AV_SERVER]** → Añadimos un nuevo agente

➢ avca add_agent –agentname avcofio –agent_host cofio.urjc.es

AVCA started

Adding agent...

Enter agent user name: avcofio

Enter agent user password:

Re-enter agent user password:

Agent added successfully.

2. **[SOURCE_DATABASE]** → Instalamos en la máquina donde se encuentra la base de datos a monitorizar el agente de base de datos.

Necesitaremos los datos que hemos introducido en el paso 1.

3. **[SOURCE_DATABASE]** → Creamos un usuario de base de datos con el cual vamos a recolectar todos los datos. A este usuario hay que darle una serie de permisos que se encuentran en los scripts zarsspriv.sql ubicado en

3.1 Crear usuario:

SQL> connect /as sysdba

Connected to an idle instance.

SQL> startup

Se crea un usuario en SOURCE_DATABASE (la base de datos Oracle origen) éste será necesario para el uso de los colectores.

(SQL> Create user srcuser1 identified by password)

SQL> create user srccofio identified by srcc0f10;

Usuario creado

3.2 Asignación de permisos:

Después de haber creado un usuario en la base de datos origen, este usuario requiere unos privilegios y roles determinado para él. Los roles y privilegios necesarios se encuentran listados en:

$ORACLE_HOME/av/scripts/streams/source/zarsspriv.sql.

En nuestro caso se encuentra en la siguiente ubicación:

```
 $ pwd
/oracle10g/product/10.2/av/av/scripts/streams/source
$ ls -la
total 208
drwxr-x---  2 ora10g dba10g   4096 oct 22 12:16 .
drwxr-x---  3 ora10g dba10g   4096 oct 22 12:16 ..
-rw-r--r--  1 ora10g dba10g  18625 jun 19  2006 zarsscfs92.pl
-rw-r--r--  1 ora10g dba10g 114590 jul 29  2007 zarsscol.sql
-rw-r--r--  1 ora10g dba10g  12553 jun 13  2007 zarsscv92.sql
-rw-r--r--  1 ora10g dba10g     63 feb 21  2007 zarssnull.sql
-rw-r--r--  1 ora10g dba10g   2091 feb 14  2007 zarsspcfs92.pl
-rw-r--r--  1 ora10g dba10g   5017 nov 29  2007 zarsspriv.sql
-rw-r--r--  1 ora10g dba10g  26854 jun 19  2006 zarsspsc92.plb
-rw-r--r--  1 ora10g dba10g   3761 jun 13  2007 zarsspsc92.sql
```

Este script se encuentra localizado tanto en el servidor Audit Vault como en el agente Audit Vault, después de haber realizado la instalación.

La ejecución del script en SOURCE_DATABASE (la base de datos origen) se debe realizar con el usuario SYS para asignar al usuario creado anteriormente los privilegios y roles requeridos. Esto se realiza usando la siguiente sintaxis.

Zarspriv.sql srccofio mode

Opciones:

- SETUP Para los colectores OSAUD y DBAUD, para una política de gestión.
- REDO_COLL Para el colector REDO log; incluye todos los privilegios que se conceden al usar el argumento SETUP

Me aseguro de que el usuario es SYS.

SQL> SHOW USER

USER es "SYS"

Se conceden al usuario origen los privilegios requerido en la política de gestión.

SQL> @/oracle10g/product/10.2/av/av/scripts/streams/source/zarsspriv.sql srccofio SETUP

Granting privileges to SRCCOFIO ... Done.

4. **[AV_SERVER]** → Verificamos desde el servidor si la configuración de la base de datos que queremos monitorizar es correcta
   Conexion de AV_SERVER a source_database :
   host (source_database):puerto:source_database

- avorcldb verify –src cofio.urjc.es:2485:reco1 –colltype ALL
  Enter Source user name: srccofio
  Enter Source password:
  source RECO1 verified for OS File Audit Collector collector
  source RECO1 verified for Aud$/FGA_LOG$ Audit Collector collector
  parameter _JOB_QUEUE_INTERVAL is not set; recommended value is 1
  ERROR: parameter UNDO_RETENTION = 900 is not in required value range [3600 - ANY_VALUE]
  ERROR: parameter GLOBAL_NAMES = false is not set to required value true
  ERROR: source database must be in ARCHIVELOG mode to use REDO LOG collector

  ERROR: global dbname for source database must include domain to use REDO LOG collector

  ERROR: set the above init.ora parameters to recommended/required values

En caso de obtener alguna indicación sobre la configuración de la base de datos fuente, modificar los parámetros indicados y volver a lanzar el comando para asegurarse que la configuración es correcta para el tipo de colector que vamos a instalar.

Como ha habido errores se edita y se añade en el INIT .ora los parámetros que hacen falta. En este caso:

*.log_archive_dest_1 = "location=/database/archivelogs"
*.log_archive_format = arch_%t_%s.arc <= eliminado por el formato falta %r
*._JOB_QUEUE_INTERVAL = 1
*.UNDO_RETENTION = 4000
*.GLOBAL_NAMES = true

El comando se vuelve a ejecutar una y otra vez hasta que todo sea correcto. En nuestro caso como no es necesario que el parámetro colltype sea ALL, utilizamos la verificación de parámetros para el tipo de colector que se va a utilizar.

Tabla1 Fuentes de apoyo, tipos de fuentes, tipos de colectores, y pistas de auditorías

Source	Source Type	Collector Types	Audit Trail
Oracle Database	ORCLDB	OSAUD DBAUD REDO	Operating system logs Database Audit tables Redo logs
SQL Server Database	MSSQLDB	MSSQLDB	C2 audit logs, server-side trace logs, Windows Event log

- avorcldb verify -src cofio.urjc.es:2485:reco1 -colltype DBAUD

Enter Source user name: srccofio
Enter Source password:
source RECO1 verified for Aud$/FGA_LOG$ Audit Collector collector

5. **[AV_SERVER]** →Añadimos en el servidor el origen a monitorizar

- avorcldb add_source -src cofio.urjc.es:2485:reco1 -desc cofio -agentname avcofio

Enter Source user name: srccofio
Enter Source password:
Adding source...
Source added successfully.
source successfully added to Audit Vault

remember the following information for use in avctl
Source name (srcname): RECO1
Storing user credentials in wallet...
Create credential oracle.security.client.connect_string3
done.
Mapping Source to Agent...

6. **[AV_SERVER]** → Añadimos un colector. Existen tres tipos de recolectores según lo que queramos monitorizar (tabla1)

   En este caso vamos a añadir un collector de tipo DBAUD

- avorcldb add_collector –srcname RECO1 -agentname avcofio -colltype DBAUD –collname RECO1_DBAUD

  source RECO1 verified for Aud$/FGA_LOG$ Audit Collector collector
  Adding collector...
  Collector added successfully.
  collector successfully added to Audit Vault

  remember the following information for use in avctl
  Collector name (collname): RECO1_DBAUD

7. **[AV_SERVER]** → Completa la configuración de la fuente

- avorcldb setup -srcname RECO1
  Introducir Nombre de Usuario de Origen: srccofio
  Introducir Contraseña de Origen:

agregando credenciales para el usuario srccofio para la conexión [SRCDB1]

Almacenando credenciales de usuario en la cartera...

Create credential oracle.security.client.connect_string3

listo.

tnsnames.ora se ha actualizado con el alias [SRCDB1] en la base de datos origen

verificando la conexión de SRCDB1 con la cartera

8. **[AV_SERVER]** → Levantar el agente.

- avctl start_agent -agentname avcofio

  AVCTL started

  Starting agent...

  Agent started successfully.

9. **[AV_SERVER]** → Levantar o iniciar los colectores.

- avctl start_collector -collname RECO1_DBAUD -srcname RECO1

  AVCTL started

  Starting collector...

Collector started successfully.