**INTRODUCTION TO INFORMATION SYSTEM**

[Insert your date time in here]

**Auditing Report**

**Definition &**

**PREPARED BY:**

1753042 – PHAN VIỆT ĐỨC

# Audit trail

## Khái niệm

* Audit trail, kiểm toán truy nguyên, trong lãnh vực kinh tế thì kiểm toán truy nguyên là bản ghi chép theo thứ tự thời gian của một giao dịch, khoản vay hay đầu từ, bao gồm các bản ghi nhớ tín dụng và các tài liệu liên quan. Đó là một bản ghi chép thể hine65 rõ tiến trình từng bước của một giao dịch lưu chuyến trong công ty, giúp việc kiểm tra sau sự kiện nhằm xác định thời gian và nơi sai sót.
* Kiểm toán truy nguyên trong lãnh vực Hệ thống thông tin. Auditing là giám sát và ghi lại các hành động đã được cấu hình trong database. Bao gồm các hành động từ cả 2 loại người dung **Database user** và **Nondatabase user.**
* Kiểm soát là một công việc cần thiết và quan trọng với người quản trị cơ sở dữ liệu. Nó là một công cụ hiệu quả chống lại các việc làm sai trái hay những hành động dẫn đến những sai sót trong quá trình làm việc với cở sở dữ liệu. Ngoài ra còn sử dụng cho các mục đích kiểm soát truy cập người dung, ngăn chặn sớm các vấn đề, điều tra các sai sót, cảnh báo các hoạt động bất ngờ,.... Và **Audit trail** đóng một vai trò rất quan trọng điều khiển truy cập, đặc biệt ở các ứng dụng hiền hành.

## Audit trail trong oracle

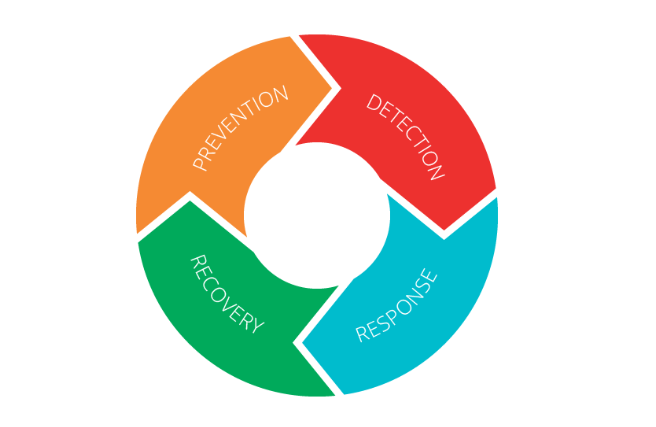
* Những thông tin được audit sẽ được lưu trong data dictionary table, gọi là **Database audit trail**, hoặc lưu trong operating sys files, lúc này gọi là **Operating system audit trail.**
* Oracle cho phép bản ghi dấu audit (Audit trail records) được trực tiếp ghi vào operating system audit trail nếu hệ điều hành tạo một audit trail sẳn cho Oracle. Nếu không thì bản audit sẽ được ghi vào file bên ngoài CSDL, với định dạng tương tự như các file dấu tích Oracle (Oracle Trace) khác.
* Oracle cho phép mợt hoài động nào đó mà luôn bị giám sát được tiếp tục, thậm chí khi mà Operating system audit trail (hay file hệ điều hành chứa bản ghi audit) không được ghép ghi lại bản ghi audit do nó bị đầy.Tuy nhiên, chú ý rằng cuấ hình Auditing để sử dụng **Database audit trail** loại bỏ điểm yếu này, bởi vì hệ CSDL Oracle ngăn ngừa sự kiện được audit khỏi xảy ra nếu audit trail không thể tiếp nhận bản ghi Database audit cho câu lệnh đó

# Security cycle

## Thực trạng

* Một ngày khác, một cuộc tấn công mạng mới. Gần đây, đó là **Ransomware**, kẻ tấn công đã mã hóa dữ liệu của công ty, làm tê liệt mục tiêu của họ trừ khi tiền chuộc được trả, sự việc này đã gây sục sôi trong ngành cũng như sục sôi ở các tờ báo nổi tiếng nhất. Nạn nhân chủ yếu là NHS UK, công ty vận chuyển Maersk và công ty hàng tiêu dùng Reckitt Benckiser. Dù cuộc tấn công đã đi qua, hệ điều hành và các ứng dụng đã được nâng cấp cập nhật để vá những lỗ hỏng trước đó, tuy nhiên con **Ransomware** đó không phải là con virus cuối cùng ở thế giới mạng hiện nay: luôn có một số mối đe dọa mới sáng tạo và nguy hiểm hơn rất nhiều đang nổi lên.
* Như chúng ta đều biết, việc bảo mật mạng chống lại các cuộc tấn công, hay đơn cử là việc bảo trì, quản lí, lưu vết thì đều như là một trò chơi rượt đuổi, rượt bắt mà sẽ chằng bào giờ có hồi kết. Mối hiểm họa mới xuất hiện, cao cấp thông minh hơn mối hiểm họa cũ, thì đồng thời các hệ thống phòng ngự cũng như quản lí tìm vết cũng phát triển theo, tạo thế cân bằng giúp đảm bảo độ an toàn của các hệ thống nói chung và các CSDL ứng dụng nói riêng.

## Chu kỳ bảo mật đề xuất

* Xây dựng mô hình quản trị linh hoạt và minh bạch cho phép chúng ta thể hiện mối quan hệ liên tục giữa Phòng ngừa, Phát hiện, Phục hồi và Phản hồi (Chu ký 4P). Nỗ lực và cố gắng cần được áp lên trên cả bốn lĩnh vực và được trình bày dưới dạng một vòng lặp liên tục. Đối với từng quá trình, số liệu cần được sử dụng để doanh nghiệp có thể trả lời các câu hỏi như:
  + Chúng ta đã làm những điều đúng đắn chưa ?
  + Chúng ta đã làm theo cách đúng đắn chưa ?
  + Chúng ta đã làm tốt chưa ?
  + Chúng ta sẽ có lợi gì trong việc này ?
* Cách tiếp cận này dẫn đến những gì được gọi là **phòng thủ theo chiều sâ**u. Ở đây, bảo mật code, mô hình hóa mối đe dọa và kiểm tra thâm nhập được sử dụng liên tục trong suốt quá trình tạo hoặc mua sắm tài sản phần mềm. Nhiều ý tưởng trong số này được thảo luận sâu ở các diễn đàn công nghệ. Những vấn đề này nhắc lại rằng bảo mật là trách nhiệm của mọi người trong tổ chức.  
  
* **Phòng ngừa**: Chính sách, tiêu chuẩn, chương trình, quy trình và quy trình của bạn mà bạn có tại chỗ là gì? Họ có giúp nhân viên của bạn đưa ra quyết định tốt liên quan đến bảo mật thông tin trong khi vẫn cho phép họ làm việc theo tốc độ mà doanh nghiệp yêu cầu không?. Bạn có cần thay đổi bất kỳ trong số chúng theo những gì bạn học được từ hồ sơ rủi ro bảo mật thông tin của bạn không?
* **Phát hiện**: Bạn biết gì? Bao nhiêu thông tin của bạn là bạn bao gồm? Làm thế nào tốt là bạn phát hiện ra lỗ hổng? Bạn có bao nhiêu Tỷ lệ thay đổi là gì? Làm thế nào để bạn phân loại chúng dựa trên rủi ro? Những cái nào bạn nên lo lắng?
* **Phục hồi**: Thời gian trung bình của chúng tôi để phục hồi là gì? Bao nhiêu phần trăm các sự kiện được xử lý một cách kịp thời và chấp nhận được? Mất bao lâu để chúng tôi phản hồi các lỗ hổng một khi chúng tôi biết về chúng? Chúng ta có cách nào để hạn chế thiệt hại cho những thứ mà chúng ta đang tìm kiếm không?
* **Phản hồi**: Mất bao lâu để chúng tôi xác định nguyên nhân gốc của các lỗ hổng đã biết và khắc phục chúng? Lấy bản vá chẳng hạn. Áp dụng các bản vá quan trọng ad hoc khi chúng được phát hành là một phản ứng ngắn hạn đối với các lỗ hổng đã biết. Nghị quyết dài hạn có thể là để thực hiện quy trình quản lý bản vá tự động, tập hợp các bản vá và áp dụng chúng theo cách có hệ thống, có thể lặp lại và có thể kiểm tra được. Nghị quyết thường phản ánh, có kế hoạch và cân nhắc nhiều hơn so với phản hồi.
* Công đoạn này thuộc về quá trình bào trì, duy trì hệ thống. Vòng lặp này cần phải duy trì liên tục khi hoàn thành xây dựng hệ thống. Với đồ án, chúng ta cần xây dựng một hệ thống quản lí đăng kí môn học, một hệ thống có thể nói đóng vai trò hết sức quan trọng và nó cũng có thể là một cửa ngõ dễ dàng cho các Hacker hay phần mềm độc hại tấn công vào. Vì Quản lí đăng kí môn học cần nhận dữ liệu đầu vào từ người dùng, chứng tỏ cũng chính là mối nguy tìm ẩn. Và sẽ ảnh hưởng hủy hoại cả hệ thống nếu không áp dụng vòng tuần hoàn bảo mật một cách có đầu tư công sức

# Application server log

* Với Application Server Control Console, bạn có thể liệt kê và tìm kiếm các tệp nhật ký trên các thành phần của Máy chủ ứng dụng Oracle. Bạn có thể xem các tệp nhật ký từ các trang Bảng điều khiển máy chủ ứng dụng hoặc tải xuống tệp nhật ký về máy khách cục bộ của bạn và xem các tệp nhật ký bằng một công cụ khác.

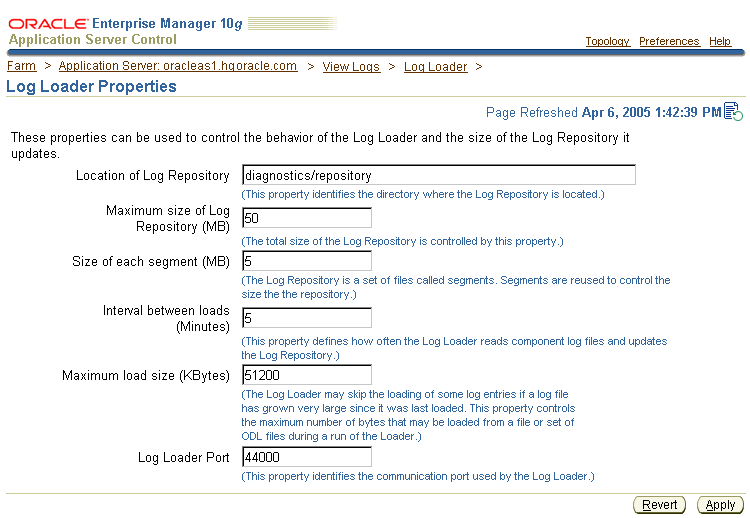
## Sử dụng Oracle Application Server Log Loader

* Oracle Application Server Log Loader khởi tạo và cập nhật các thông báo trong Kho lưu trữ nhật ký. Kho lưu trữ nhật ký lưu trữ các thông báo chẩn đoán từ nhiều tệp nhật ký trên các thành phần Máy chủ ứng dụng Oracle. Sau khi Trình tải nhật ký bắt đầu, theo các khoảng thời gian đều đặn, nó sẽ đọc nội dung của tệp nhật ký và lưu trữ nội dung vào Kho lưu trữ nhật ký.
* Thành phần Trình tải nhật ký máy chủ ứng dụng Oracle khởi tạo và cập nhật các thông báo trong Kho lưu trữ nhật ký. Kho lưu trữ nhật ký lưu trữ các thông báo chẩn đoán từ nhiều tệp nhật ký trên các thành phần Máy chủ ứng dụng Oracle. Sau khi Trình tải nhật ký bắt đầu, theo các khoảng thời gian đều đặn, nó sẽ đọc nội dung của tệp nhật ký và lưu trữ nội dung vào Kho lưu trữ nhật ký.

## Start and stop log loader

* Chúng ta truy cập vào trang web của Application Server Control Console page to start and stop the Log Loader.
* Chi tiết các bước để start:
  + Nhấp vào link “Logs” trên trang Application Server Control Console
  + Trên trang View Logs, nhấp vào link “Search Log Repository”.
  + Nhấp “Log loader” trên trang View Logs page
  + Trân trang Loader, nhấn “Start”, sau đó xác nhận.
    - Nếu Ứng dụng đang chạy, thì nhấn “Stop” đề dừng ứng dụng
  + Nhấp “Start” hoặc “Start and Load Existing Logs” để bắt đầu và khởi tạo ứng dụng Server Log

## Cài đặt Log Loader

* Chúng tá có thể cài đặt những thuộc tính ứng dụng ở trang Log Loader bằng cách
  + Chọn “log” trên trang Application Server Control Console
  + Trên trang View Log, chọn đường link “Search log repository”
  + Nhấp “Log Loader”
  + Chọn “Log loader properties” trong phần “Administration”. Sau đó tùy chọn các thuộc tính cần thao tác.
  + Cuối cùng nhấn “Apply”  
    

# Application Auditing

* Đấy là hình thức audit thường dùng nhất vì tính tự nhiên và có thể đáp ứng hầu hết mọi yêu cầu của các Auditor (Nhân viên kiểm toán). Đây là một dạng chương trình mà chính các nhên viên phát triển phải tự tay lập trình trên ứng dụng, vì vậy rất dễ chỉnh sửa nếu sai sót hoặc rất tiện lời cho việc mở rộng. Việc Auditing được lập trình cài đặt trân ứng dụng, nên các User thậm chí sẽ không biết rang đang diễn ra quá trình auditing.
* Được Trình bài, lập trình cái đặt bằng các hàm thủ tục, mục đích các hàm phục vụ mục đích auditing, theo dõi user login, hành động thao tác tương tác với dữ liệu

## Ưu điểm

* Tính linh hoạt
* Tính đa hỗ trợ
* Tính đa chiều (có thể audit chuyên biệt nhiều khía cạnh của dữ liệu)

## Nhược điểm

* Cần phải lập trình tỉ mỉ
* Truy cập không thông qua ứng dụng, application trở nên vô dụng. Có thể là một con dao 2 lưỡi nếu hacker tấn công vào được database.

# Trigger Auditing

* Dùng để ghi nhận và theo dõi các hành vi trong phạm vi cơ sở dữ liệu bằng việc sử dụng các hàm trigger, cụ thể là DML trigger.
* Cách thực hiện
  + Tự tạo bảng phụ nhằm lưu lại dữ liệu quá trình audit
  + Sử dụng các hàm trigger thủ tục ghi nhận lại các hoạt động dữ liệu vào ra trên bảng

## Ưu điểm

* Trong suốt với ứng dụng
* Phù hợp tùy theo ngữ cảnh
* Audit có chọn lọc dữ liệu khi thực thi audit
* Các trigger audit có thể được gọi bởi nhiều application

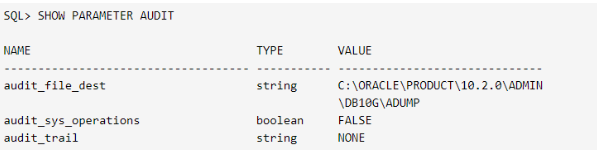
## Nhược điểm

* Dể xảy ra sai sót, dẫn đến audit thiếu hoặc không thực hiện audit
* Trigger không cho phép truyền tham số

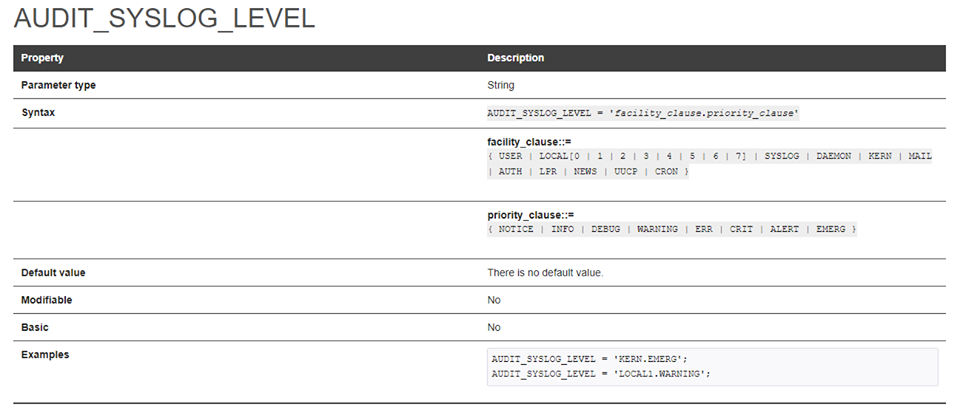
# Mandatory Auditing

* CSDL Oracle sẽ luôn ghi nhận lại các thông tin, một cách mặc định và bắt buộc:
  + Start-up database: Ghi nhận lại có bật chế độ audit hay không trường hợp admin đã tắt đi chế độ audit r khởi động lại CSDL
  + Shutdown database
  + Việc xác thực với các Role là SYSDBA hoặc SYSOPER
* Audit ở dạng này, các record được lưu ở mức hệ điều hành.
* Tính năng audit này trên Oracle là mặc định

## Chi tiết phương thức show các parameter của mandatory audit

* Đăng nhập sqlplus với role là SYSDBA
* Nhập lệnh “SHIW PARAMETER AUDIT”  
  

# Auditing SYS (User SYS)

* Cơ chế này sử dụng để chuyên biệt ghi nhận lại những hành động được thực hiện bởi SYSDBA hoặc SYSOPER. Audit record của cơ chế này được ghi vào file mứa OS. Tuy nhiên người dùng, SYSDBA hoặc SYSOPER vẫn có thể truy cập và xóa dữ liệu audit.
* Enables or disables quá trình auditing các hoạt động cao cấp nhất được chỉ định bởi chính các User là SYS, SYSDBA, SYSOPER hoặc các user có quyền hạn tượng ứng. Nếu chúng ta đặt tham số khởi tạo AUDIT\_TRAIL thành **XML** hoặc **XML, EXTENDED**, nó sẽ ghi các bản ghi kiểm toán theo định dạng XML. Trên các hệ thống **UNIX**, nếu bạn cũng đã đặt tham **số AUDIT\_SYSLOG\_LEVEL**, thì nó sẽ ghi đè tham số **AUDIT\_TRAIL**, ghi các bản ghi kiểm toán SYS vào nhật ký kiểm toán hệ thống bằng tiện ích **SYSLOG**.  
    
  

# Standard Audit

* Có thể audit trên :
  + Table / view
  + Việc thực thi proc
  + Các đặc quyền hệ thống
  + 1 số user cụ thể
  + Trến các hành động thành công hoặc không thành công

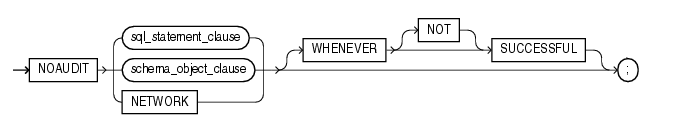
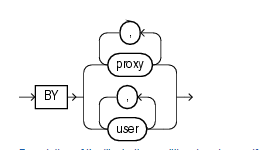
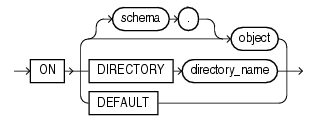
## Kịch hoạt Standard Auditing

* Chức năng Audit mặc định không được kích hoạt, nhưng có thể kích hoạt bằng lệnh sau: ALTER SYSTEM SET audit\_trail = db SCOPE = SPFILE;  Trong đó: giá trị của audit\_trail có thể thay đổi tùy theo nhu cầu của người sử dung: audit\_trail = {none | os | db | db, extended | xml | xml, extended}
  + none: tắt chế độ audit.
  + os: bật chế độ audit và các audit record sẽ được lưu trong file OS.
  + db: bật chế độ audit và các audit record sẽ được lưu trong database audit trail (SYS.AUD$)
  + xml: bật chế độ audit và các audit record sẽ được lưu file OS có định dạng XML

## Kích hoạt lựa chọn Standard Auditing

* Lệnh AUDIT thiết lập lựa chọn giám sát câu lệnh và quyền thường đi sau mệnh đề "BY" để giới hạn tầm vực của câu lệnh và lựa chọn giám sát quyền.
  + BY ACCESS: Ghi một record cho mỗi câu lệnh và hoạt động được audit
  + BY SESSION: Ghi một record cho tất cả các câu lệnh SQL cùng loại và tất cả các hoạt động cùng loại được thực hiện trên cùng một đối tượng schema trong cùng một session
  + WHENEVER SUCCESSFUL/ WHENEVER NOT SUCCESSFUL:
    - WHENEVER SUCCESSFUL: thực hiện ghi dữ liệu audit đối với những câu lệnh được thực hiện thành công. ví dụ: AUDIT SELECT ON USERS BY ACCESS WHENEVER SUCCESSFUL; -- Thực hiện audit đối với việc thực thi thành công câu lệnh SELECT trên bảng USERS. audit update on USERS by access WHENEVER successful; --Thực hiện audit đối với việc thực thi thành công câu lệnh UPDATE trên bảng USERS.
    - WHENEVER NOT SUCCESSFUL: thực hiện ghi dữ liệu audit đối với những câu lệnh được thực hiện không thành công. ví dụ: audit update on USERS by access WHENEVER not successful; -- Thực hiện audit đối với việc thực thi lỗi câu lệnh UPDATE trên bảng USERS.
* Có thể cài đặt thực hiện audit(tạo/xóa/chỉnh sửa cấu trúc bảng) trên bất kì bảng nào trên database: AUDIT DROP ANY TABLE; AUDIT CREATE ANY TABLE; AUDIT DELETE ANY TABLE;

## Tắt Standard Auditing

* Sử dụng câu lệnh NOAUDIT để tắt các lựa chọn giám sát.
  + Kết hợp với các mệnh đề WHENEVER để tắt các giám sát đối với các câu lệnh được thực hiện thành công hay không thành công.  Ví dụ: NOAUDIT SELECT ON USERS WHENEVER SUCCESSFUL; -- tắt chế độ audit khi lấy list users thành công. NOAUDIT SELECT ON USERS WHENEVER NOT SUCCESSFUL;
  + Kết hợp với "BY":  Ví dụ: NOAUDIT SELECT USERS BY user\_name;  -- tắt chế độ audit khi lấy list users chỉ đối với user có tên là user\_name.
  + Kết hợp với "ON":  NOAUDIT SELECT ON USERS  -- tắt chế độ audit khi lấy list users.
  + Ở mức tổng quát hơn, ta có thể dùng trực tiếp câu lệnh NOAUDIT với các schema\_object: ví dụ: NOAUDIT ALL PRIVILEGES;  -- tắt audit trên quyền. NOAUDIT ROLE; -- tắt audit trên role( ví dụ việc create/ drop... role) NOAUDIT ALL;  -- tắt giám sát trên câu lệnh

# Fine-grained Auditing

* Cho phép các chính sách kiểm toán được liên kết với các cột trong bảng ứng dụng cùng với các điều kiện cần thiết để tạo bản ghi kiểm toán. Các chính sách FGA được gán cho các bảng ứng dụng bằng API FGA. Chính sách Fine-grained audit chi tiết có thể được sử dụng để các records khi một bảng được truy cập trong các giai đoạn cụ thể hoặc các cột cụ thể được truy cập. FGA bổ sung cho cơ sở dữ liệu Oracle 12c những phương thức Audit mới, bằng cách cho phép các điều kiện kiểm toán được liên kết với các cột cụ thể.
* Audit records được tạo bởi Fine-prained Auditing có thể được ghi lại và phân tích trong Oracle Audit Vault và Database Firewall, tự động cảnh báo cho nhóm bảo mật về hoạt động độc hại có thể xảy ra.
* Demo thử nghiệm được thực hiện bởi chính Oracle: <https://www.oracle.com/technical-resources/articles/middleware/idm-fga-otn.html?fbclid=IwAR2NhlzgBRS5DHe914pshjOXv8ZKeOTdUsi6U8tDI1os26STZfYSsaKSZJg>

# Áp dụng vào đồ án