CTT201

An toàn và Bảo mật Dữ liệu trong HTTT

Chương 5: Auditing

TS. Phạm Thị Bạch Huệ
Ths. Hoàng Anh Tú

Khoa Công nghệ thông tin – Đại học Khoa học tự nhiên

Nội dung

1. Giới thiệu
2. Các phương pháp auditing
3. Các kiểu auditing

Chương 5: Auditing

GIỚI THIỆU

VÒNG ĐỜI BẢO MẬT (SECURITY CYCLE)
MỤC ĐÍCH CỦA VIỆC AUDITING

Vòng đời bảo mật

- Ngăn ngừa (Prevention) → Dò tìm để phát hiện (Detection) → Hồi đáp (Response)
  - Ngăn ngừa: dùng cơ chế điều khiển truy cập (access control).
  - Dò tìm để phát hiện tấn công: thực hiện lúc xảy ra tấn công và đang có người theo dõi tấn công đó.
  - Hồi đáp: Phản ứng lại những tấn công.

- Khó có thể xây dựng 1 ứng dụng trên máy tính hoàn toàn bảo mật.
- **Auditing** nhằm phục vụ cho cơ chế *dò tìm để phát hiện* tấn công. Hồi đáp được thực hiện dựa trên dữ liệu có được trong quá trình auditing.

Mục đích việc auditing

- Auditing cho phép ta bắt các user phải có trách nhiệm về hành động mà họ thực hiện, bằng cách theo dõi hành vi của họ.
- Dữ liệu audit giúp phát hiện lỗ hổng trong chính sách bảo mật.
- Liên quan đến trách nhiệm giải trình của user.
  - Cần phải đảm bảo rằng user chỉ được thực hiện những gì họ được phép.
  - Ghi nhận sự lạm quyền hoặc dùng sai quyền.

Mục đích việc auditing

- Auditing để ghi nhận lại những gì **đã** xảy ra và có hồi đáp thích hợp.
- Auditing giúp kiểm chứng được khía cạnh bảo mật của hệ thống có được đảm bảo không; hoặc có ai đã đọc / cập nhật dữ liệu một cách bất hợp pháp không.
- Việc auditing hiệu quả khi:
  - Có kế hoạch thực hiện auding.
  - Đọc lại và phân tích dữ liệu của quá trình auditing.

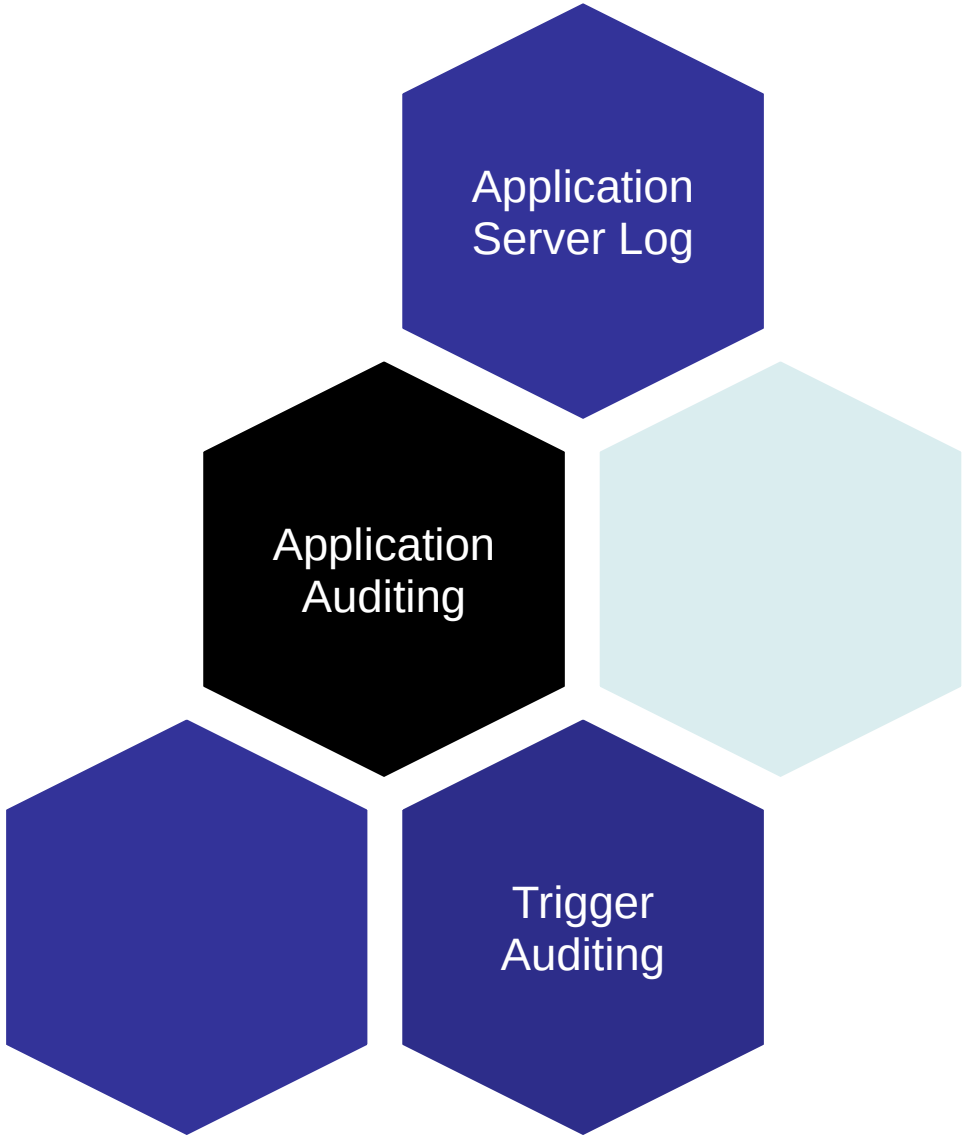
Một số nhận xét

- Auditing tất cả các hành động của tất cả các user trên tất cả dữ liệu sẽ không có ích mà còn làm chậm hệ thống, và dữ liệu có được từ quá trình audit khó sử dụng.

- Audit một cách có chọn lọc và đúng đắn, dựa trên dữ liệu, xử lý và người dùng có thật.

Chương 5: Auditing

CÁC PHƯƠNG PHÁP AUDITING



Application Server Log

- Nhật ký truy cập trên Application Server hay Web Server là dạng auditing cơ bản.
- Có nhiều thông tin chứa trong các tập tin nhật ký: các tài nguyên được truy cập, ai đã truy cập, khi nào, như thế nào (thành công, thất bại hay không biết).
- Dạng audit này có ích cho việc phát hiện những hành vi đáng nghi ngờ (ví dụ tấn công DoS).
- Admin dùng file log này để kiểm tra hành vi của user khi họ duyệt web.

- *Khuyết điểm:*
  - Thông tin không trực tiếp: chỉ có IP address mà không biết user nào, chỉ có URL mà không biết chương trình nào.
  - Dạng audit này thường được dùng kết hợp với dạng khác (Application Auditing) để biết ai đã thực hiện hành vi trên đối tượng nào.

Trigger đặt lại IP Address thành Client Identifier

```
sec_mgr@KNOX10g> CREATE OR REPLACE TRIGGER set_ip_in_id
  2    AFTER LOGON ON DATABASE
  3  BEGIN
  4    DBMS_SESSION.set_identifier
  5                          (SYS_CONTEXT ('userenv',
  6                                        'ip_address'));
  7  END;
  8  /
```

Application Auditing

- Đây là hình thức audit thường dùng nhất vì tính tự nhiên và có thể đáp ứng hầu hết mọi yêu cầu.
- Được lập trình một cách thủ công trên ứng dụng và có thể mở rộng, chỉnh sửa.

- Vì việc audit được thực hiện trên ứng dụng, vì vậy trên CSDL user không biết rằng có diễn ra quá trình auditing.
  - Ứng dụng sẽ ghi nhận có chọn lọc các thông tin cần thiết
  - Ví dụ, user login, các hành động thao tác dữ liệu, các thao tác quản lý.

Application Auditing

- Ứng dụng sẽ gọi thi hành những thủ tục thực hiện audit.
- Ví dụ sau thực hiện audit thao tác update trên trường SAL của bảng EMP.

```
scott@KNOX10g> CREATE TABLE aud_emp (
  2    username    VARCHAR2(30),
  3    action      VARCHAR2(6),
  4    empno       NUMBER(4),
  5    column_name VARCHAR2(255),
  6    call_stack  VARCHAR2(4000),
  7    client_id   VARCHAR2(255),
  8    old_value   VARCHAR2(10),
  9    new_value   VARCHAR2(10),
 10    action_date DATE DEFAULT SYSDATE
 11  )
```

```
scott@KNOX10g> CREATE OR REPLACE PROCEDURE audit_emp (
  2     p_username      IN  VARCHAR2,
  3     p_action        IN  VARCHAR2,
  4     p_empno         IN  NUMBER,
  5     p_column_name   IN  VARCHAR2,
  6     p_old_value     IN  VARCHAR2,
  7     p_new_value     IN  VARCHAR2)
  8   AS
  9   BEGIN
 10   -- check data format and length
 11   -- not shown here
 12     INSERT INTO aud_emp
 13                 (username,
 14                  action,
 15                  empno,
 16                  column_name,
 17                  call_stack,
 18                  client_id,
 19                  old_value,
 20                  new_value,
 21                  action_date)
 22          VALUES (p_username,
 23                  p_action,
 24                  p_empno,
 25                  p_column_name,
 26                  DBMS_UTILITY.format_call_stack,
 27                  SYS_CONTEXT ('userenv',
 28                               'client_identifier'),
 29                  p_old_value,
 30                  p_new_value,
 31                  SYSDATE);
 32   END;
 33   /
```

```
scott@KNOX10g> CREATE OR REPLACE PROCEDURE update_sal (
  2    p_empno    IN  NUMBER,
  3    p_salary   IN  NUMBER)
  4  AS
  5    l_old_sal  VARCHAR2 (10);
  6  BEGIN
  7    SELECT     sal
  8          INTO l_old_sal
  9          FROM emp_copy
 10         WHERE empno = p_empno
 11    FOR UPDATE;
 12    UPDATE emp_copy
 13       SET sal = p_salary
 14     WHERE empno = p_empno;
 15    audit_emp
 16      (p_username       => USER,
 17       p_action         => 'UPDATE',
 18       p_empno          => p_empno,
 19       p_column_name    => 'SAL',
 20       p_old_value      => l_old_sal,
 21       p_new_value      => p_salary);
 22  END;
 23  /
```

```
scott@KNOX10g> CREATE OR REPLACE PROCEDURE show_aud_emp
  2  AS
  3  BEGIN
  4    FOR rec IN (SELECT   *
  5                  FROM aud_emp
  6                ORDER BY action_date DESC)
  7    LOOP
  8      DBMS_OUTPUT.put_line (   'User:        '
  9                            || rec.username);
 10      DBMS_OUTPUT.put_line (   'Client ID:   '
 11                            || rec.client_id);
 12      DBMS_OUTPUT.put_line (   'Action:      '
 13                            || rec.action);
 14      DBMS_OUTPUT.put_line (   'Empno:       '
 15                            || rec.empno);
 16      DBMS_OUTPUT.put_line (   'Column:      '
 17                            || rec.column_name);
 18      DBMS_OUTPUT.put_line (   'Old Value:   '
 19                            || rec.old_value);
 20      DBMS_OUTPUT.put_line (   'New Value:   '
 21                            || rec.new_value);
 22      DBMS_OUTPUT.put_line (   'Date:        '
 23                            || TO_CHAR
 24                                 (rec.action_date,
 25                                  'Mon-DD-YY HH24:MI'));
 26      DBMS_OUTPUT.put_line
 27                     ('------------------------------');
 28    END LOOP;
 29  END;
```

```
scott@KNOX10g> GRANT EXECUTE ON update_sal TO blake;

Grant succeeded.

scott@KNOX10g> GRANT SELECT ON emp_copy TO blake;

Grant succeeded.

blake@KNOX10g> SELECT empno, sal
  2    FROM scott.emp_copy
  3   WHERE ename = 'BLAKE';

     EMPNO        SAL
---------- ----------
      7698       2850

blake@KNOX10g> EXEC scott.update_sal(p_empno=>7698, p_salary=>3000);

PL/SQL procedure successfully completed.

blake@KNOX10g> COMMIT ;

Commit complete.

blake@KNOX10g> SELECT empno, sal
  2    FROM scott.emp_copy
  3   WHERE ename = 'BLAKE';

     EMPNO        SAL
---------- ----------
      7698       3000
```

```
scott@KNOX10g> EXEC show_aud_emp
User:        BLAKE
Client ID:   192.168.0.100
Action:      UPDATE
Empno:       7698
Column:      SAL
Old Value:   2850
New Value:   3000
Date:        Mar-24-04 13:34
------------------------------
```

Application Auditing

- **Ưu điểm:**
  - Dễ mở rộng, chỉnh sửa.
  - Hỗ trợ nhiều yêu cầu trong quá trình audit, có thể điều khiển cách thức audit.
  - Ứng dụng trên application server có thể chỉ định lưu dữ liệu audit vào file hoặc trên 1 CSDL khác phòng admin (của CSDL được audit) nhìn thấy.
  - Có thể audit trên nhiều khía cạnh của ứng dụng: việc truy cập dữ liệu, audit trên nhiều CSDL liên quan đến ứng dụng, audit trên file, trên web service, …
- **Khuyết điểm:**
  - Coding → có thể xảy ra lỗi và phải bảo trì.
  - Có thể có những truy cập không thông qua ứng dụng thì application audit không có tác dụng.

Trigger Auditing

- Dùng để ghi nhận và theo dõi các hành vi trong phạm vi cơ sở dữ liệu, dùng các database trigger, cụ thể là DML trigger.
- Mang tính trong suốt, thực hiện auditing mà không cần thực hiện trên ứng dụng.

- Thực hiện:
  - Tạo bảng phụ lưu dữ liệu của quá trình audit.
  - Trigger gọi thủ tục ghi nhận lại dữ liệu vào bảng trên.

Ví dụ

```
scott@KNOX10g> CREATE OR REPLACE TRIGGER update_emp_sal_trig
  2    BEFORE UPDATE OF sal
  3    ON emp_copy
  4    FOR EACH ROW
  5  DECLARE
  6  BEGIN
  7    audit_emp (p_username       => USER,
  8               p_action         => 'UPDATE',
  9               p_empno          => :OLD.empno,
 10               p_column_name    => 'SAL',
 11               p_old_value      => TO_CHAR (:OLD.sal),
 12               p_new_value      => TO_CHAR (:NEW.sal));
 13  END;
 14  /

Trigger created.
```

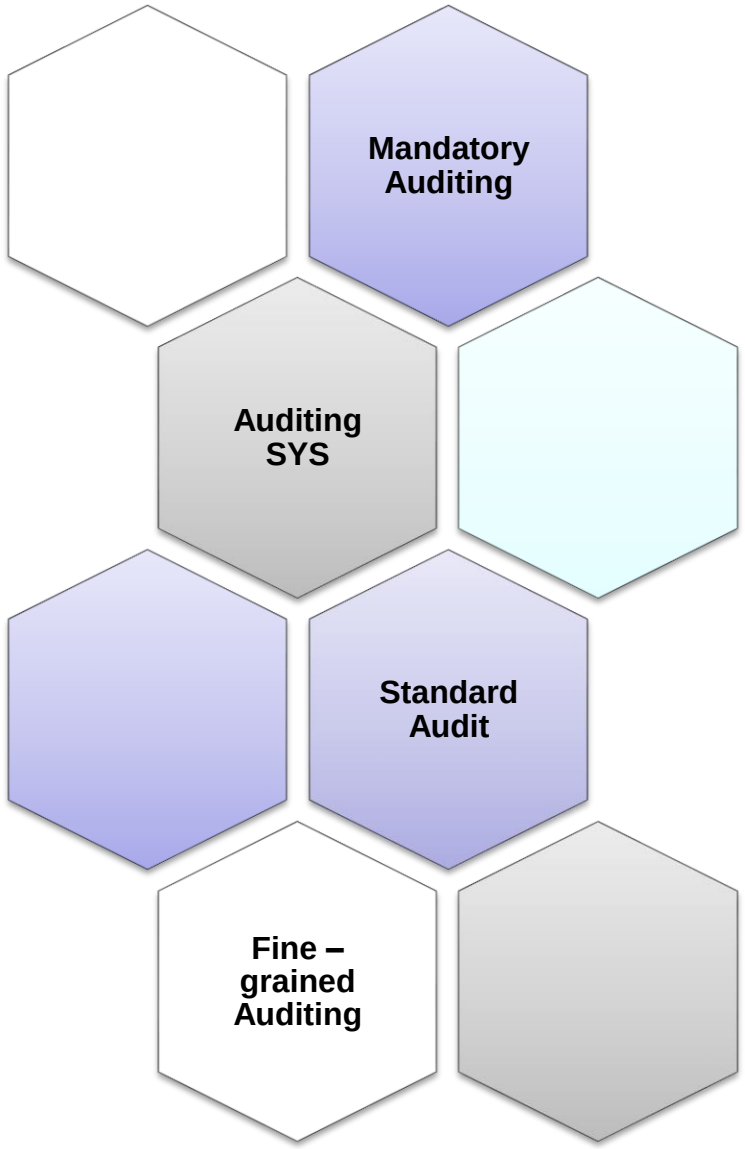
Trigger auditing

- Ưu điểm:
  - Trong suốt đối với ứng dụng.
  - Phù hợp với ngữ cảnh mua ứng dụng và không thể chỉnh sửa code của ứng dụng.
  - Có thể thực hiện audit trên từng cột, từng dòng cho từng câu lệnh → audit có chọn lọc và giảm bớt những dữ liệu không cần thiết khi thực hiện audit.
  - Trigger auditing có thể được gọi thực hiện bởi nhiều application.
- Khuyết điểm:
  - Có thể trigger không thực thi trong một số trường hợp.
  - Trigger thì không cho phép truyền thêm tham số (ngoài giá trị cũ và giá trị mới, username, IP address).

Chương 5: Auditing

CÁC KIỂU AUDITING

Các dạng Auditing chuẩn mức CSDL của Oracle 10g



1. Mandatory Auditing

- CSDL **luôn** ghi nhận lại các thông tin về:
  - Database startup.
  - Database shutdown.
  - Các user được xác thực với role là SYSDBA hoặc SYSOPER.
  - Database startup: Auditing record ghi nhận lại có bật chế độ đang audit lên hay không, phòng trường hợp admin đã tắt chế độ audit và khởi động lại CSDL.
  - Auditing record được lưu ở mức hệ điều hành.

2. Auditing SYS

- Ghi nhận những hành động được thực hiện bởi user được xác thực với role là SYSDBA hoặc SYSOPER.
- Auditing record được ghi vào tập tin mức OS.
- Các role này có thể thực hiện các quyền tối quan trọng và có thể xóa dữ liệu audit.

3. Standard audit

- Có thể audit:
  - Trên đối tượng là table/ view.
  - Việc thực thi procedure.
  - Các đặc quyền hệ thống (VD: tắt kích hoạt 1 trigger).
  - Trên 1 số user cụ thể.
  - Trên các hành động thành công hoặc không thành công.

4. Fine – grained auditing

- Cung cấp thêm một số tính năng so với standard auditing: kiểm tra điều kiện trước khi audit, column sensitivity, …

Tóm lại

- Về nguyên tắc, có thể audit để lấy mọi thông tin:
  - Audit logon, logoff into the database.
  - Audit source of database usage.
  - Audit database usage outside normal operating hours.
  - Audit DDL activity.
  - Audit database errors.
  - Audit changes to sources of stored procedures and triggers.
  - Audit changes to privileges, user/login definitions, and orther security attributes.
  - Audit creations, changes, and usage of database links and replication.
  - Audit change to sensitive data.
  - Audit SELECT statements for privacy sets.
  - Audit any changes made to the definition of what to audit.
- *Theo Ron Ben Natan, Implementing Database Security and Auditing, Elsevier Digital Press, ISBN 1-55558-334-2, 2005.*

Tham khảo

- ***David Knox, Effective Oracle Database 10g Security by Design, McGraw-Hill Osborne, ISBN 0-07-223130-0, 2004.***

Câu hỏi

TS. Phạm Thị Bạch Huệ - ptbhue@fit.hcmus.edu.vn

Ths. Hoàng Anh Tú – hatu@fit.hcmus.edu.vn