trivadis

|Grifes 28 Octobre 2003

Sécurité Oracle

Gérard Schaller
Gerard.Schaller@trivadis.com

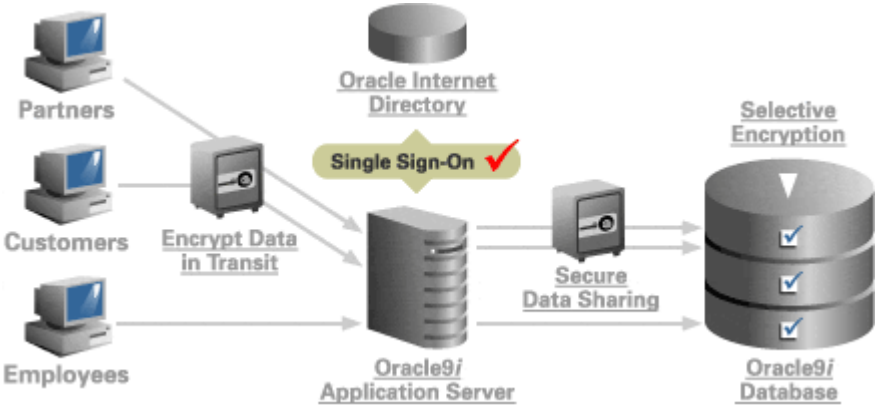
Agenda

- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - Web Services - HTTP - WebDAV – FTP - XDB
  - Identification des applications se connectant à la base de données
- Sécurité dans la base de données
  - Concept de sécurité : Compte / mot de passe / rôles / privilèges
  - Auditing : Standard , Triggered, Fine Grained Auditing
  - Chiffrement dans Oracle
  - Row Level Security, Label security

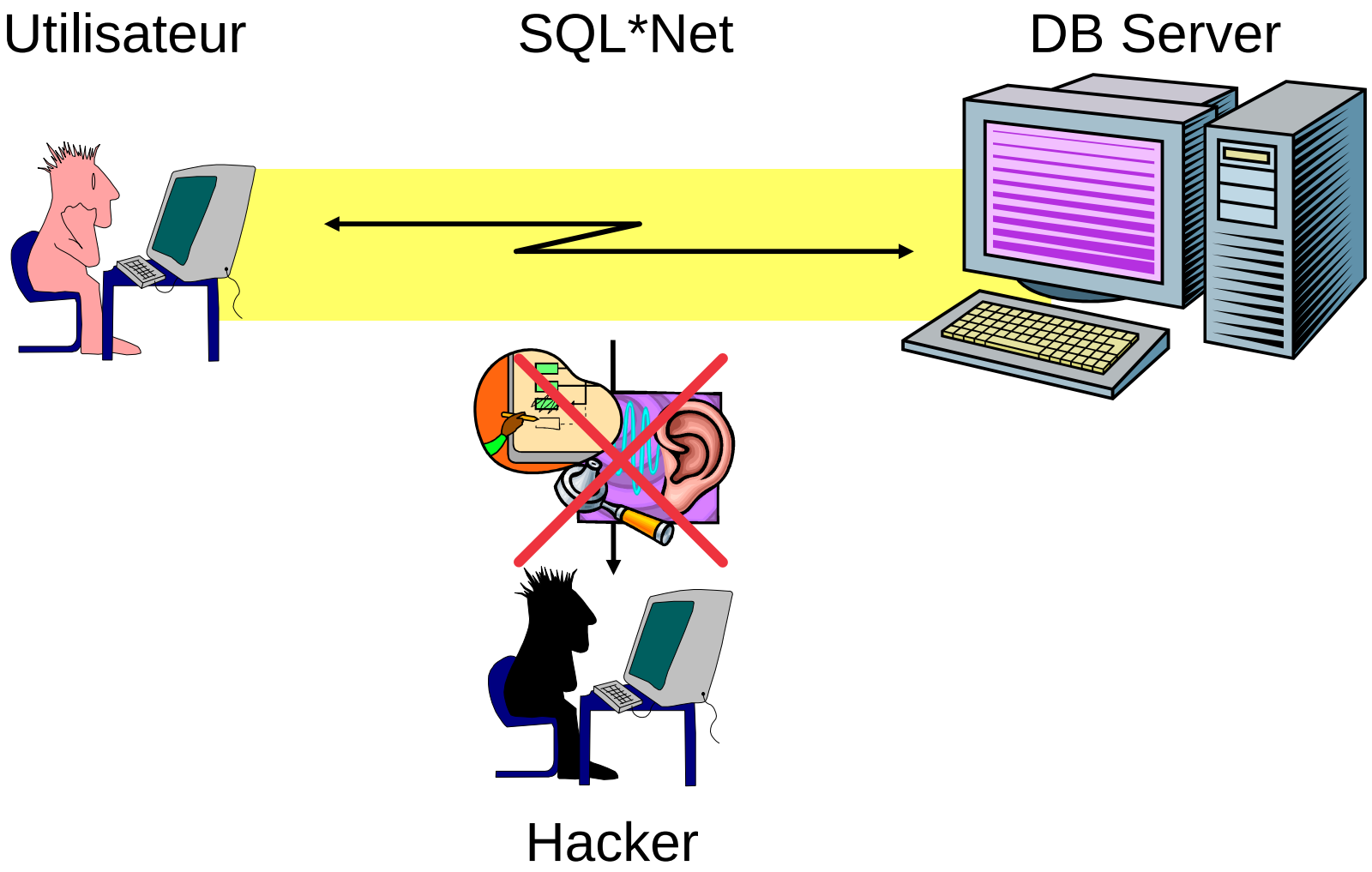
Marketing Oracle

Unbreakable
ORACLE

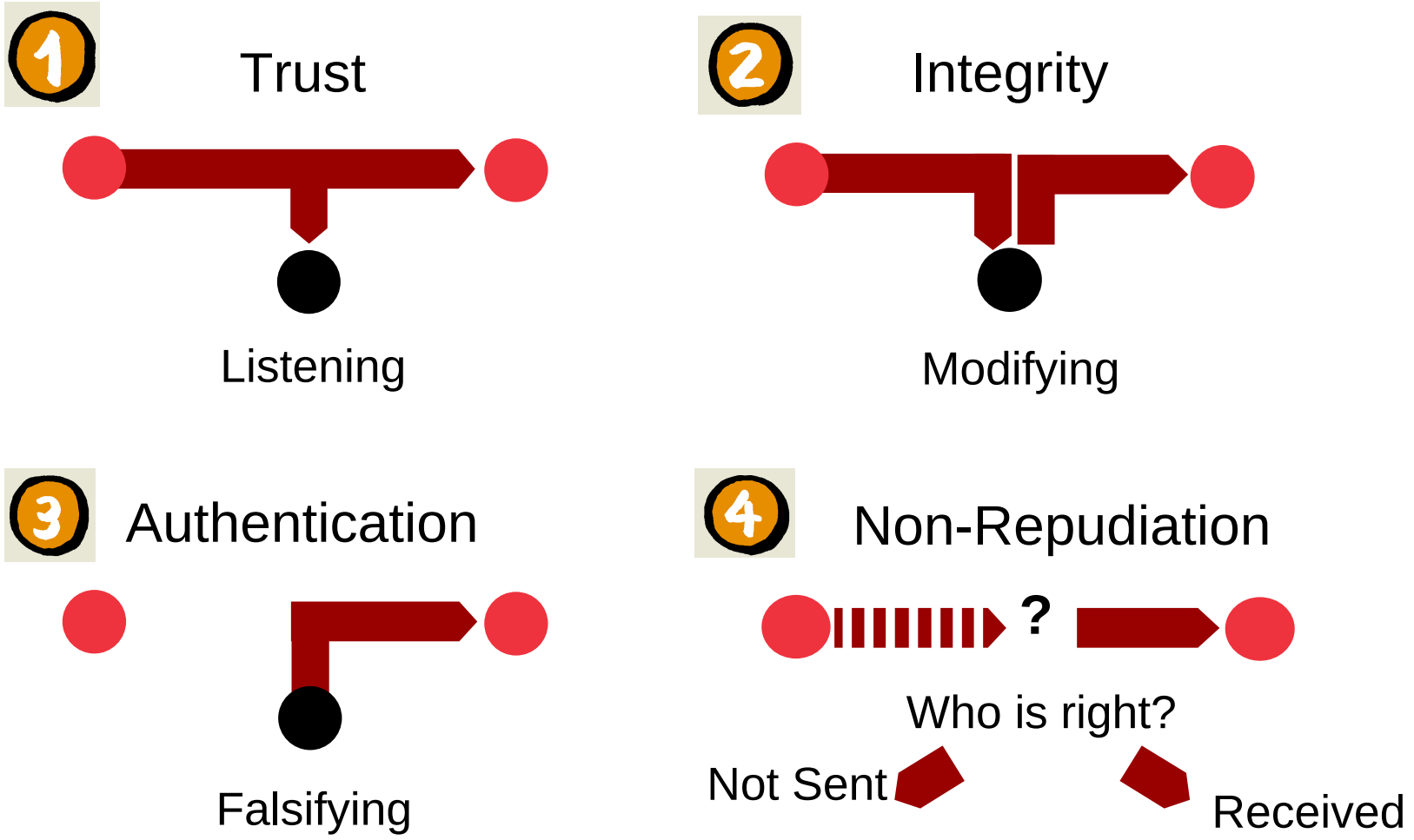
- Oracle9*i* addresses your security vulnerabilities by securing data transmitted from client to application server to database, encrypting sensitive data within the database, restricting user access at the row-level, providing a single point of entry to all authorized applications and quickly detecting data misuse. Plus you have concrete assurance against break-ins, having built upon 15 successfully completed security evaluations. IBM and Microsoft can't compete.



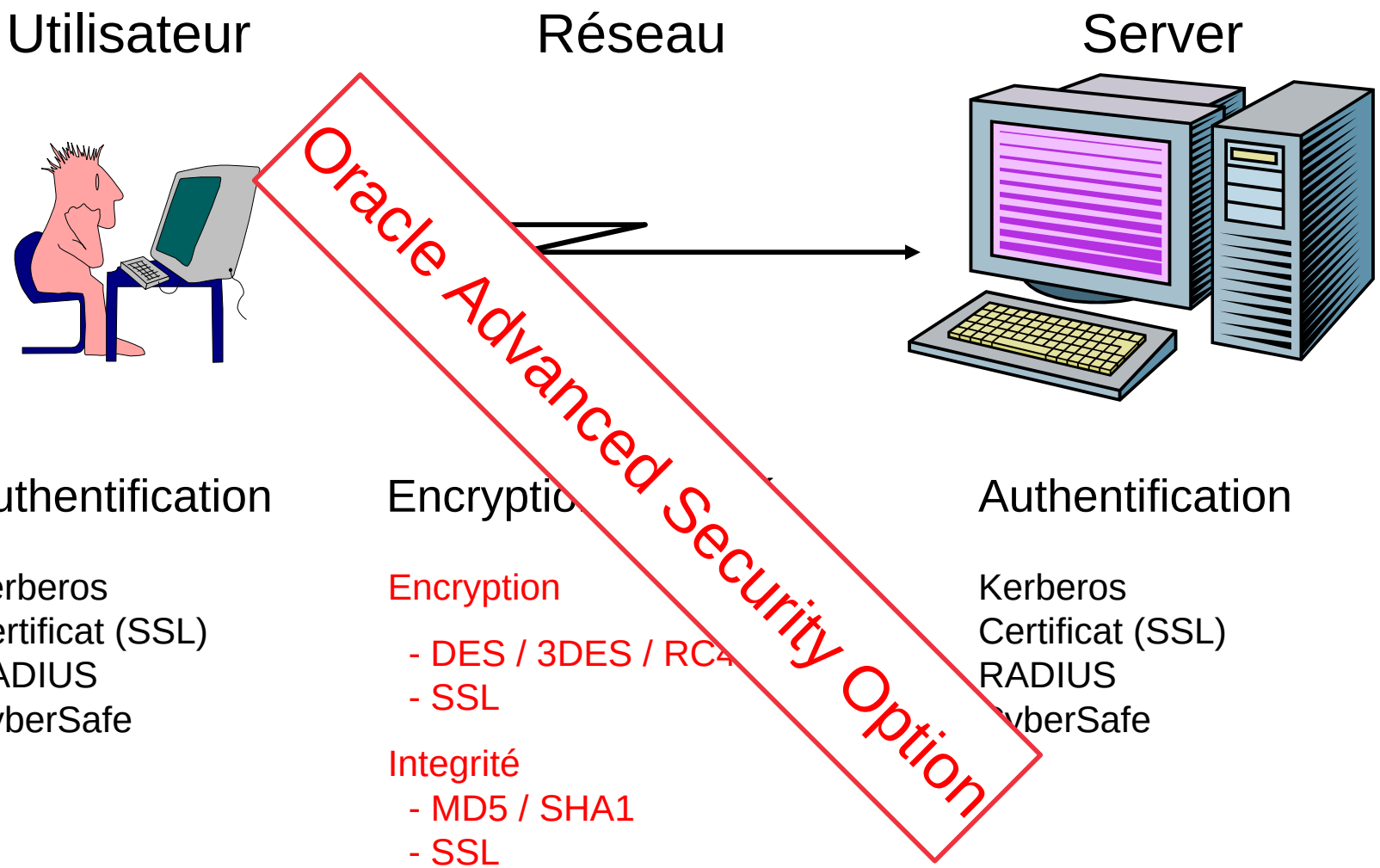
Quels sont les risques dans la transmission?



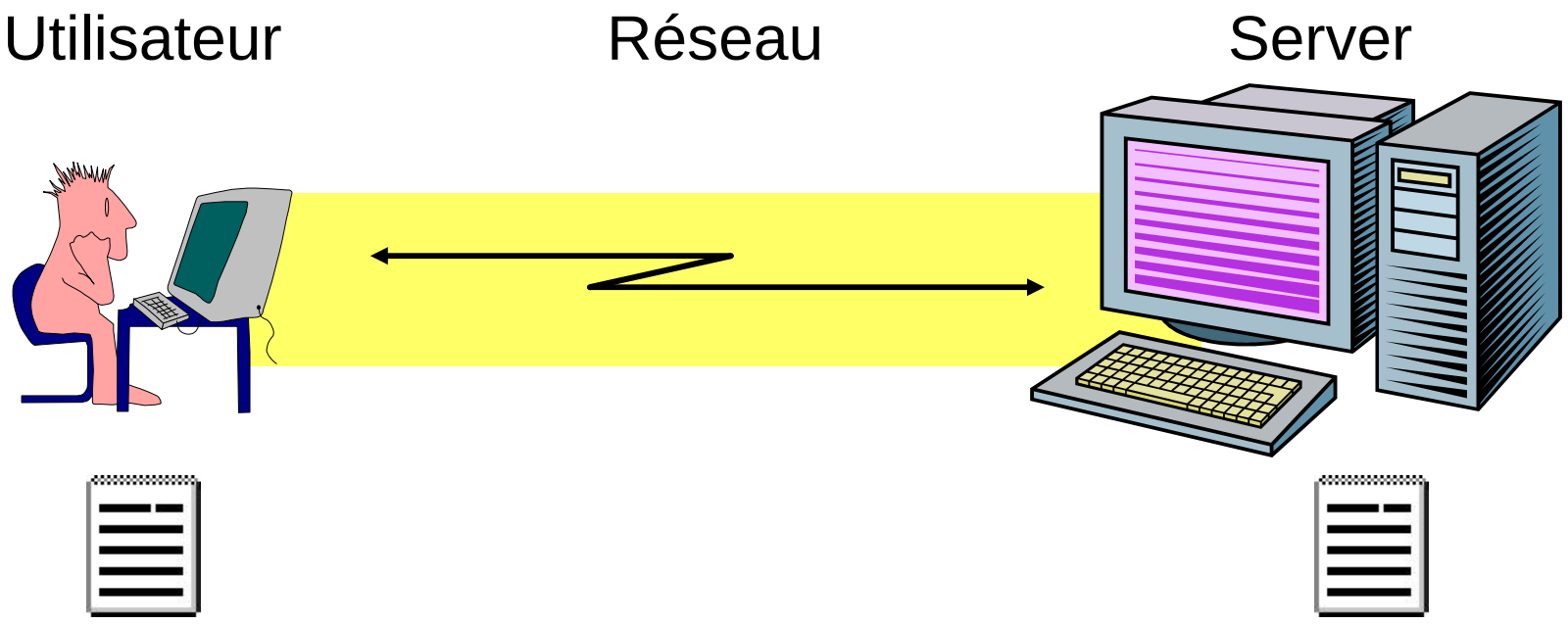
Problèmes Fondamentaux de Sécurité



Ce qu‘offre Oracle



Integrité



Integrité

Contrôle de l‘intégrité

- Le chiffrement à lui seul n‘est pas suffisant pour l‘intégrité!
- Pour le contrôle d‘intégrité Oracle détecte:
  - La modification de paquets de données
  - La perte de paquets de données
  - L’ajout de paquet de données (i.e. Replay Attacks)
- La protection de chaque paquet est faite par:
  - Un numéro de séquence
  - Un checksum (Hash)
  - L’association d’une master clé liée à la session
- Oracle dispose de 2 procédés
  - MD5 Message Digest 5 (128 Bit)
  - SHA-1 Secure Hash Algorithm (160 Bit)

Checksum faux

Numéro de séquence faux

⇩

déconnection

Configuration du crypto_checksum_*

- Server et Client Parameter doivent être en accord
  - DEFAULT: **Accepted** ⟹ **pas de contrôle de l'intégrité**

Client \ Server		CRYPTO_CHECKSUM _SERVER =			
		REJECTED	ACCEPTED	REQUESTED	REQUIRED
CRYPTO_CHECKSUM_CLIENT=	REJECTED	OFF	OFF	OFF Server demande, Client refuse	PAS de Session Server requiert, Client refuse
	ACCEPTED	OFF	OFF Ni le serveur ni le client ne demande!!!	ON Server demande, Client accepte	ON Server demande, Client accepte
	REQUESTED	OFF Client demande, Server refuse	ON Client demande, Server accept	ON Client/Server demandent et acceptent	ON Server requiert, Client demande, les 2 acceptent
	REQUIRED	PAS de Session Client requiert, Server refuse	ON Client requiert, Server accepte	ON Client requiert, Server demande, les 2 acceptent	ON Client et Server requièrent et acceptent

Exemple de configuration: sqlnet.ora

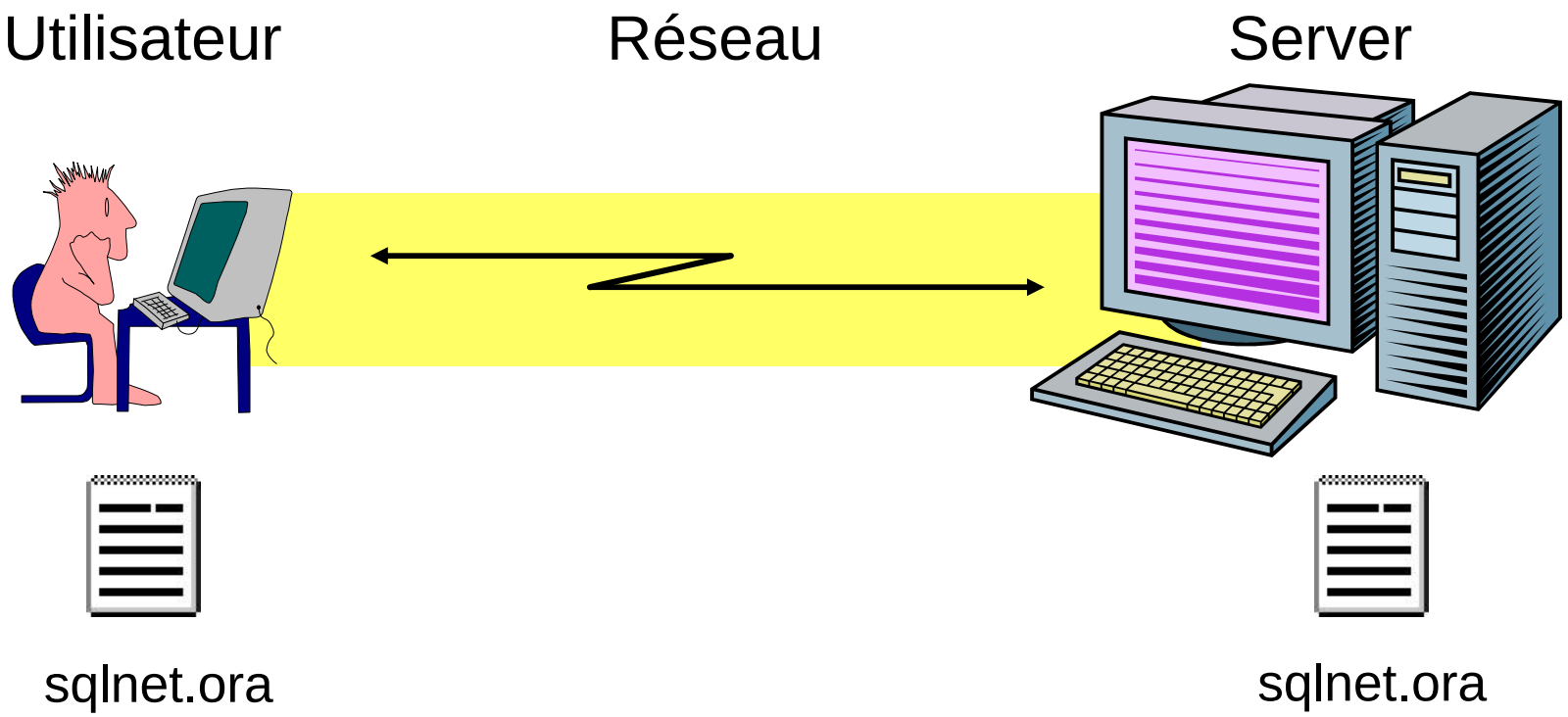
- SQLNET.ORA Client (extrait)

```
SQLNET.CRYPTO_CHECKSUM_CLIENT = required
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (SHA1)
```

- SQLNET.ORA Server (extrait)

```
SQLNET.CRYPTO_CHECKSUM_SERVER= required
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER= (SHA1)
```

Encryption


Utilisateur
Réseau
Server
sqlnet.ora
sqlnet.ora
Encryption

Encryption - Parameter

- sqlnet.ora Parameter

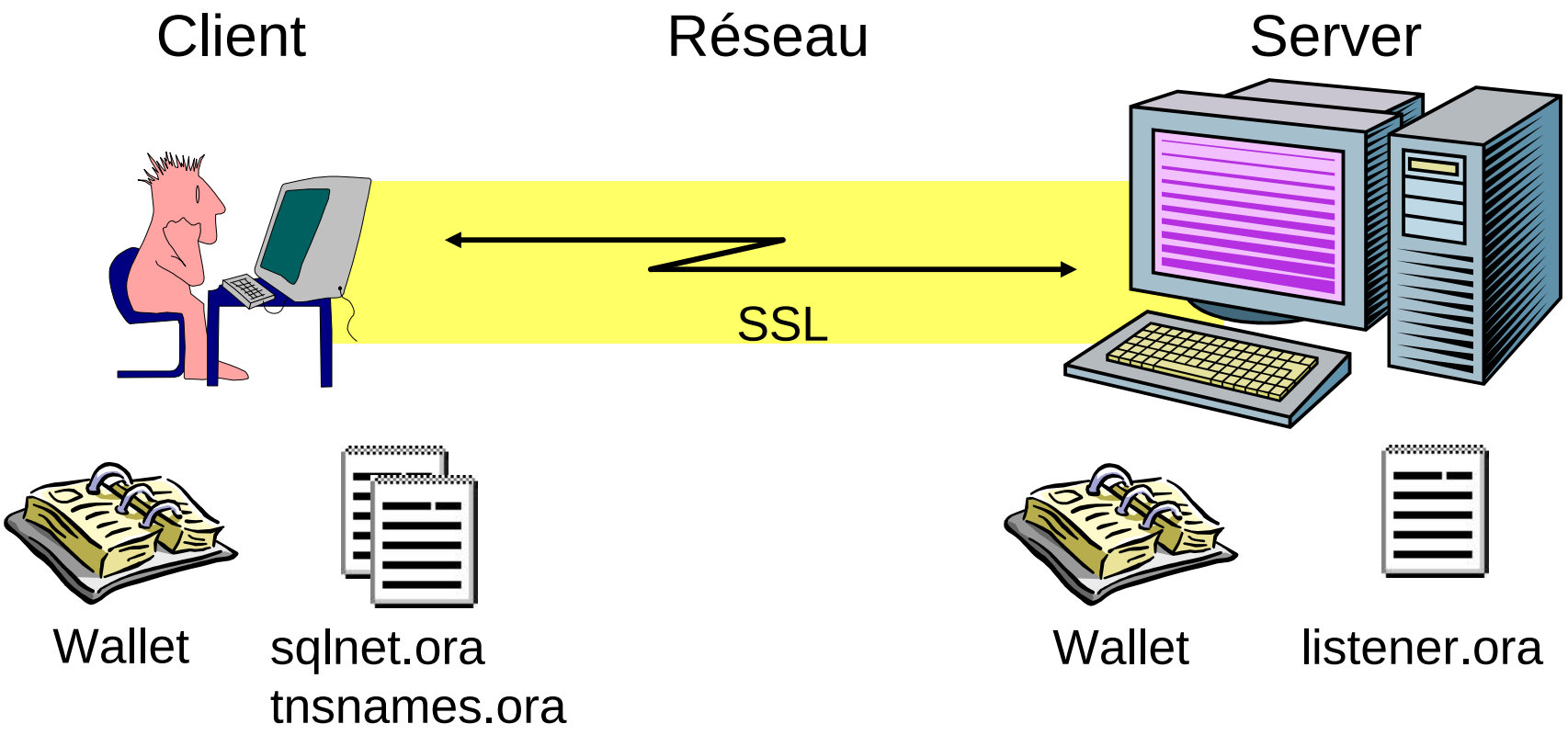
PARAMETER	VALUES
SQLNET.ENCRYPTION_SERVER SQLNET.ENCRYPTION_CLIENT	REJECTED ACCEPTED REQUESTED REQUIRED
SQLNET.ENCRYTION_TYPES_SERVER SQLNET.ENCRYTION_TYPES_CLIENT	(RC4_256, RC4_128, RC4_56, RC4_40, AES256, AES192, AES128, DES, DES40, 3DES168, 3DES112)
SQLNET.CRYPTO_SEED	10 à 70 „caractères“ i.e 4fhfguweotc3adssafjkdsfqp5

Configuration des paramètres encryption_*

- Les paramètres sur le Server et le Client doivent être en accord
  - DEFAULT: Accepted

Client \ Server		SQLNET.ENCRYPTION_SERVER=			
		REJECTED	ACCEPTED	REQUESTED	REQUIRED
SQLNET.ENCRYPTION_CLIENT=	REJECTED	OFF	OFF	OFF Server demande, Client refuse	PAS de Session Server requière, Client refuse
	ACCEPTED	OFF	OFF Ni le serveur ni le client ne demande!!!	ON Server demande, Client accepte	ON Server demande, Client accepte
	REQUESTED	OFF Client demande, Server refuse	ON Client demande, Server accept	ON Client/Server demandent et acceptent	ON Server requière, Client demande, les 2 acceptent
	REQUIRED	PAS de Session Client requière, Server refuse	ON Client requière, Server accepte	ON Client requière, Server demande, les 2 acceptent	ON Client et Server requièrent et acceptent

Encryption via SSL



CLIENT - Authentification	Encryption / Integrité	SERVER - Authentification

Encryption avec SSL

- Standard d‘Internet
  - Utilisé sur les pages Web avec des données sensibles
- Résoud plusieurs problèmes
  - Authentification (Server et Client)
  - Chiffrement
  - Intégrité
- A besoin d‘une Public Key Infrastructure ou au minimum de certificat

Composants de Oracle SSL

- Certificat (incl. Clé privée)
  - Authentification des Servers
  - Authentification des Clients (seulement si le client possède un certificat)
- Oracle Wallet
  - Fichier Standard PKCS#12
    - Contient les clés privées-publiques ainsi que le certificat
    - Nécessite un password pour y accéder
  - Est administré avec le Wallet Manager
- Oracle Net
  - Implémentation de SSL pour SQL*Net

sqlnet.ora Exemple

- Server (extrait)

```
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /etc/ORACLE/WALLETS/oracle)
    )
  )
SSL_CIPHER_SUITES= (SSL_RSA_WITH_RC4_128_MD5)
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 0
...
```

- Client (extrait)

```
...
SSL_CIPHER_SUITES= (SSL_RSA_WITH_RC4_128_MD5)
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 0
...
```

ASO SSL Performance

- Coût de l'encryption et de l'intégrité
- Exemple d'un test:
  - Sun Ultra5/10, UltraSparc IIi 333MHz,
    SunOS 5.8, Oracle 9.0.1
  - Client et Server sur le même serveur, connection via Net8
  - select * from BIG_EMP (Table avec 1 Mio Rows)
  - Output-Spool est d'env. 40MB
  - Sans encryption: **Ø 2 min 58 sec**
  - Avec encryption / Hash (SSL_RSA_WITH_RC4_128_MD5):
    Ø 3 min 37 sec (=22% de plus)
- Les performances 9*i* sont nettement supérieures à celles de la version 8.1.7 (plus de 47%)

Agenda

- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - **Web Services - HTTP - WebDAV – FTP - XDB**
  - Identification des applications se connectant à la base de données
- Sécurité dans la base de données
  - Concept de sécurité : Compte / mot de passe / rôles / privilèges
  - Auditing : Standard , Triggered, Fine Grained Auditing
  - Chiffrement dans Oracle
  - Row Level Security, Label security

Web Services - HTTP - WebDAV – FTP - XDB

- A l'installation de Oracle un serveur apache est automatiquement installé et démarré!
  - Par défaut tout le monde est administrateur (en 8*i* l'url n'est pas protégée !)
  - Doit être modifié manuellement dans le fichier wdbsvr.app

```
[PLSQL_GATEWAY]
administrators = all
adminPath = /admin_/
;admindad =
;debugModules =
defaultDAD = simpledad
;upload_as_long_raw =
;upload_as_blob =
;enablesso =
;stateful =
;custom_auth =
;error_style =
[DAD_simpledad]

;exclusion_list = dbms*
…
```

Web Services - HTTP - WebDAV – FTP - XDB

- En 8*i*
  - Il est possible d'invoquer des procédures stockées et en particulier les packages Web Toolkit installé dans le schéma de sys (anciennement dans le schema OAS_PUBLIC pour OAS)
  - Exemples:
    - http://*hostname*/pls/*dad_name*/owa_util.showsource?cname=*procedure_name*

    Donne le code source de la procédure
    - http://*hostname*/pls/*dad_name/owa_util.cellsprint?p_theQuery=select+*+from+SCOTT.EMP*

    Liste le contenu de la table EMP du schema de Scott
- En 9*i*
  - Possible si le paramètre exclusion_list est renseigné sans spécifier sys.*

Web Services - HTTP - WebDAV – FTP - XDB

- En créant une standard database 9*i* avec l'assistant de création de DB les services suivants seront activés:
  - FTP sur le port 2100
  - HTTP/WebDAV sur le port 8080
- Ces services donnent accès à la partie XML DB d'Oracle
- Recommandations:
  - N'installer que les options que vous désirez utiliser !
  - Désactiver les services non utilisé en les assignants au port 0 : exemple pour ftp

```
CALL DBMS_XDB.CFG_UPDATE(UPDATEXML
(DBMS_XDB.CFG_GET,'/xdbconfig/sysconfig/protocolconfig/ftpconfig/ftp-
port/text()',‘0'));
```

Agenda

- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - **Web Services - HTTP - WebDAV – FTP - XDB**
  - **Identification des applications se connectant à la base de données**
- Sécurité dans la base de données
  - Concept de sécurité : Compte / mot de passe / rôles / privilèges
  - Auditing : Standard , Triggered, Fine Grained Auditing
  - Chiffrement dans Oracle
  - Row Level Security, Label security

Identification des applications se connectant à la base de données

- Oracle offre depuis 8*i* un mécanisme d'authentification des applications se connectant à la base.
- Ceci se fait via la table product_user_profile
  - Exemple:

PRODUCT	USERID	ATTRIBUTE	SCOPE	VALUE
SQL*PLUS	SCOTT	CONNECT		DISABLED

Le user scott n'est pas autorisé à se connecter avec l'outil SQL*PLUS

- On peut restreindre l'usage d'une partie des commandes sql à l'outil ou contrôler le rôle de l'utilisateur :
  - ALTER AUDIT CREATE DELETE DROP GRANT INSERT LOCK NOAUDIT RENAME REVOKE SELECT UPDATE VALIDATE
- Seul quelques outils Oracle peuvent être contrôlés (SQL*PLUS, REPORT, etc …). On ne peut pas empêcher une connection via un ODBC !!

Identification des applications se connectant à la base de données

- Dans la vue v$session:

```
OSUSER          MACHINE              USERNAME     SID    SERIAL#     PROGRAM
--------------- -------------------- ------------ ------ ----------- --------------
LTGES02\ges     MSHOME\LTGES02       SCOTT        28     2930        EXCEL.EXE
LTGES02\ges     MSHOME\LTGES02       SYS          29     60          sqlplus.exe
SYSTEM          MSHOME\LTGES02       SYS          30     158         dbsnmp.exe
LTGES02\ges     MSHOME\LTGES02       SCOTT        31     28          sqlplusw.exe
LTGES02\ges     MSHOME\LTGES02       SYS          32     4197        SQLNav4.exe
```

- On peut déclencher un trigger ON LOGON qui identifiera le programme utilisé pour se connecter. Un autre programme décidera si l'utilisateur est autorisé et de l'action à entreprendre.
- Si le nom de l'exécutable est changé il ne sera pas possible de l'identifier!

Agenda

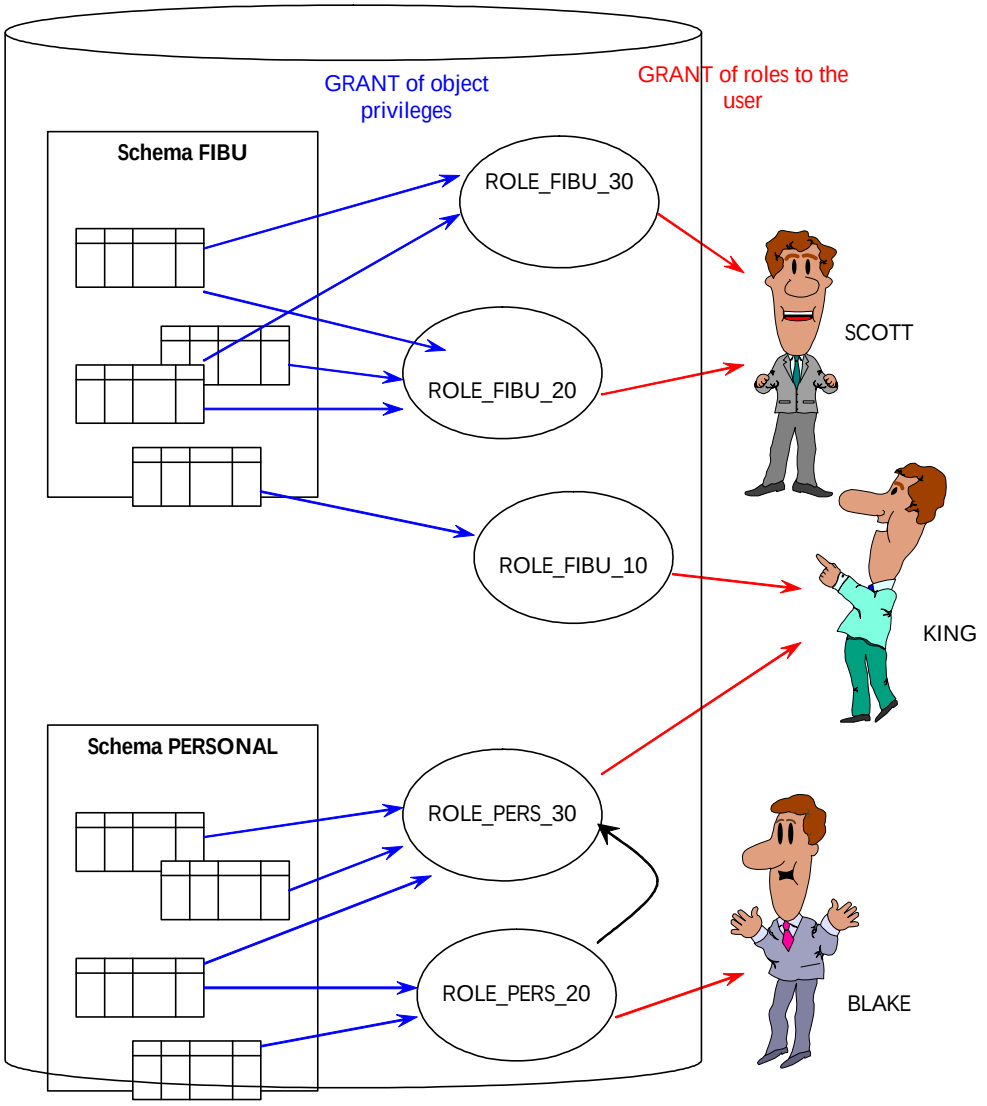
- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - Web Services - HTTP - WebDAV – FTP - XDB
  - Identification des applications se connectant à la base de données
- Sécurité dans la base de données
  - **Concept de sécurité : Compte / mot de passe / rôles / privilèges**
  - Auditing : Standard , Triggered, Fine Grained Auditing
  - Chiffrement dans Oracle
  - Row level security, Label security

Rôles et Privilèges

- Privilèges:
  - Sur des objects:
    grant select , update , etc …
  - System:
    connect, create table

- Les privilèges peuvent être donnés directement à un utilisateur ou via un rôle

- On associe à un rôle un ensemble de privilèges system ou objet

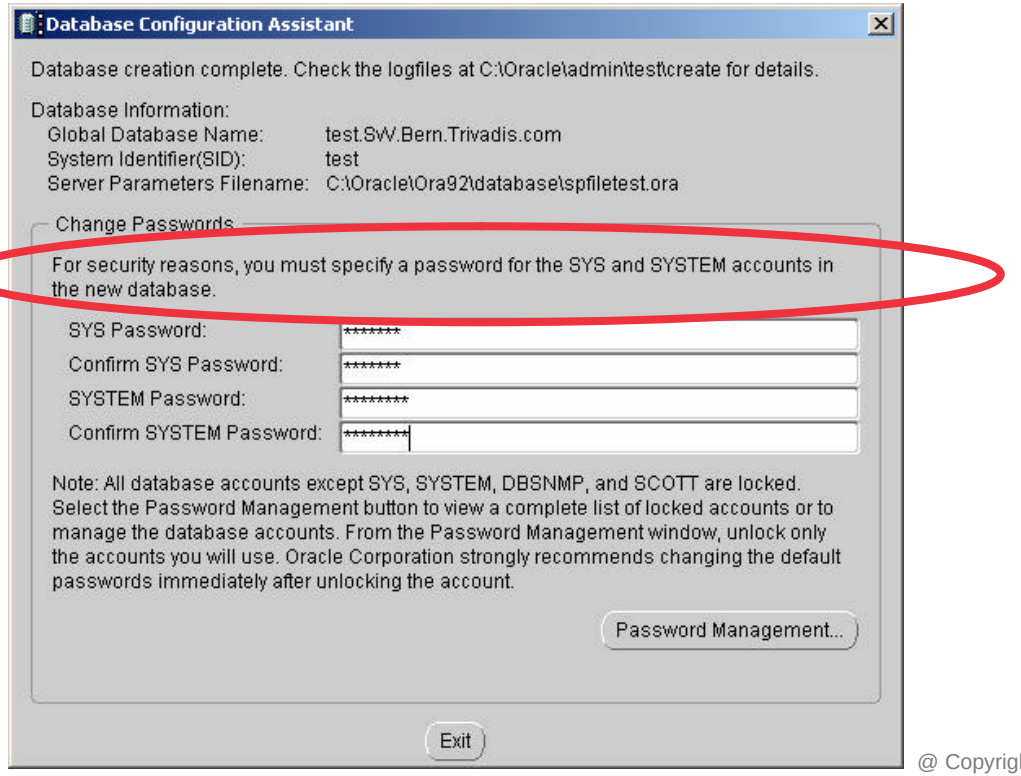


Compte utilisateurs venant avec la DB

Compte	DBA	RESOURCE (unlimited tablespace)	execute any procedure	select any table	select any dictionary	exempt access policy	alter system
CTXSYS	✓	✓	✓	✓	✓		✓
DBSNMP					✓		
LBACSYS		✓	✓	✓			
OLAPDBA		✓		✓	✓		
OLAPSVR		✓		✓	✓		
OLAPSYS		✓		✓	✓		
ORDPLUGINS		✓					
ORDSYS		✓					
OUTLN		✓	✓				
MDSYS	✓	✓	✓	✓	✓		✓
WKSYS	✓	✓	✓	✓	✓		✓
XDB				✓		✓	

Comptes utilisateurs

- Certains utilisateurs viennent par défaut avec la base de données avec de hauts privilèges (i.e outln)
- A partir de 9*i* **seulement** l'assistant de création de DB bloque ces comptes et force le changement des passwords de SYS et SYSTEM.



Comptes utilisateurs

- Outln a le privilège **execute any procedure** et peut par exemple donner le rôle DBA au user SCOTT:

```
CONNECT outln/outln
CREATE OR REPLACE
PROCEDURE grant_dba_to_scott
IS
        cur INTEGER;
BEGIN
        cur := DBMS_SQL.open_cursor;
        -- uid 0 = SYS
        SYS.DBMS_SYS_SQL.parse_as_user (cur, 'GRANT DBA TO SCOTT',
                                        DBMS_SQL.native,0);
        DBMS_SQL.close_cursor (cur);
END;
/
execute grant_dba_to_scott;
```

- Le password d'un DBLINK est visible dans la table SYS.LINK$

Agenda

- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - Web Services - HTTP - WebDAV – FTP - XDB
  - Identification des applications se connectant à la base de données
- Sécurité dans la base de données
  - Concept de sécurité : Compte / mot de passe / rôles / privilèges
  - **Auditing : Standard , Triggered, Fine Grained Auditing**
  - Chiffrement dans Oracle
  - Row level security, Label security

Standard Oracle9*i* Auditing (1)

- Oracle9*i* permet un auditing standard sur 3 niveaux (comme en Oracle8*i*)

- Auditing de Statements
  - Permet d'auditer les commandes DML ou DDL.
  - Exemples:
    - AUDIT TABLE audit toutes les commandes CREATE TABLE ou DROP TABLE
    - SELECT TABLE audit tous les select sur des tables/vues indépendamment de la table.

```
SQL> AUDIT SELECT TABLE;
```

  - Peut être restreint à un utilisateur

```
SQL> AUDIT UPDATE TABLE by SCOTT,ADAMS;
```

Oracle Standard Auditing (2)

- System privileges
  - Audit tous les statements faisant appel à un privilège particulier: i.e SELECT ANY TABLE

```
SQL> AUDIT SELECT ANY TABLE, ALTER ANY TABLE;
```

- Object
  - Audit certaines opérations sur des objets particuliers

```
SQL>  AUDIT INSERT, UPDATE, DELETE on HR.EMPLOYEES;
```

Auditing par trigger

- DML triggers: auditing des modifications
- Database Event Trigger
  - Possibilité de logger les connections à la base (qui, quand, d'où)

```
CREATE OR REPLACE TRIGGER LOGON_EMPLOYEE
  AFTER LOGON ON DATABASE
DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
 IF ( sys_context('userenv','sessionid') != 0 ) then
   INSERT INTO system.logon_events
      ( SELECT USER,
               sys_context('userenv','sessionid'),
               sys_context('userenv','os_user'),
               sys_context('userenv','host'),
               sys_context('userenv','ip_address'),
               program,
               sysdate
        FROM sys.v_$session
         WHERE AUDSID = sys_context('userenv','sessionid') );
    COMMIT;
  END IF;
END;
```

Fine Grained Auditing

- Les résultats d'un SELECT statement ne peuvent pas être audité par un audit standard
- Le Fine Grained Auditing permet de définir des conditions fines pour l'auditing.
- Le FGA est constitué de 2 parties:
- L'Audit Event Condition
  - définit la condition qui doit être vérifiée pour que le statement soie audité.
  - Exemple: DEPTNO=20 sur la colonne SAL signifie que l'on audit les statements qui accèdent au salaire des employés du département 20.
- L'Event Handler
  - Le nom d'une procédure qui sera exécutée lorsque l'Audit Event Condition est vérifiée.
  - Par exemple: envoyé un mail d'alerte, insérer dans une table le nom du user, etc …
- Chaque Audit Event Condition peut avoir son propre Event Handler

DBMS_FGA

- Exemple: on veut auditer les requêtes recherchant le salaire des employés du dept=20

```
CREATE OR REPLACE PROCEDURE ins_audit_message(
  p_schema VARCHAR2, p_table VARCHAR2, p_policy VARCHAR2) IS
BEGIN
  INSERT INTO audit_messages VALUES(
    SYSDATE, p_schema||'.'||p_table||'.'||p_policy);
END;
/
BEGIN
 DBMS_FGA.ADD_POLICY(
    object_schema=> 'SCOTT',
    object_name  => 'EMP',
    policy_name  => 'DEPTNO20_SAL',
    audit_condition      => 'DEPTNO = 20',
    audit_column         => 'SAL',
    handler_schema       => 'SYSTEM',
    handler_module       => 'INS_AUDIT_MESSAGE',
    enable               => true);
END;
```

Fine Grained Auditing

- DBA_FGA_AUDIT_TRAIL:

```
SQL> SELECT
session_id,timestamp,db_user,os_user,object_schema,object_name,
  2         policy_name,scn,sql_text
  3  FROM   dba_fga_audit_trail
  4  WHERE  policy_name = 'DEPTNO20_SAL'
  5  ORDER BY scn DESC
  6  /

SESSION_ID TIMESTAMP DB_USER  OS_USER   OBJECT_SCH OBJECT_NAM POLICY_NAME
---------- --------- -------- --------- ---------- ---------- -----------
       SCN SQL_TEXT
---------- --------------------------------------------------------------
       186 13-AUG-01 SCOTT    reb       SCOTT      EMP        DEPTNO20_SAL
    358210 SELECT 6,DEPTNO,ENAME,SAL FROM EMP WHERE DEPTNO = 20

       186 13-AUG-01 SCOTT    reb       SCOTT      EMP        DEPTNO20_SAL
    358189 SELECT 1,DEPTNO,ENAME,SAL FROM EMP
```

Fine Grained Auditing

- Fine grained auditing est supporté seulement pour le cost-based optimizer
- Attention le rule based optimiser génère des entrées fausses d'auditing!

Agenda

- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - Web Services - HTTP - WebDAV – FTP - XDB
  - Identification des applications se connectant à la base de données
- Sécurité dans la base de données
  - Concept de sécurité : Compte / mot de passe / rôles / privilèges
  - Auditing : Standard , Triggered, Fine Grained Auditing
  - **Chiffrement dans Oracle**
  - Row level security, Label security

Chiffrement dans Oracle

- A partir de Oracle 8.1.6, Oracle offre la possibilité d'encrypté le contenu dans la base avec un nouveau package: DBMS_OBFUSCATION_TOOLKIT
- Peut-être utilisé comme une procédure ou une fonction et supporte le DES3 depuis 8.1.7. (DES3Encrypt et DES3Decrypt)
- La chaîne de caractères en entrée doit avoir une longueur multiple de 8.
- Une clé doit-être passée pour réalisé l'encryption et vice versa. Où stocker cette clé?
- Une fonction de hachage (non réversible) MD5 est aussi fournie et pourrait être utilisée pour l'encryption de password.

Chiffrement dans Oracle

- Exemple d'une fonction générant l'encryption
  - Voir article http://www.trivadis.com/publikationen/F/encryption_dans_oracle_f.htm

```
FUNCTION crypt ( input_string  IN varchar2) return varchar2 IS
BEGIN
  DECLARE
   error_in_input_buffer_length EXCEPTION;
   PRAGMA EXCEPTION_INIT(error_in_input_buffer_length, -28232);
   INPUT_BUFFER_LENGTH_ERR_MSG VARCHAR2(100) :=
   double_encrypt_not_permitted EXCEPTION;
   PRAGMA EXCEPTION_INIT(double_encrypt_not_permitted, -28233);
   DOUBLE_ENCRYPTION_ERR_MSG VARCHAR2(100) :=

    encrypted_string            VARCHAR2(2048);
    encrypted_raw               RAW(2048);
    key_string          VARCHAR2(16)  := 'thisisthekeytext';
    raw_input           RAW(128) := UTL_RAW.CAST_TO_RAW(input_string);
    raw_key             RAW(128) := UTL_RAW.CAST_TO_RAW(key_string);

BEGIN
     dbms_obfuscation_toolkit.DES3Encrypt(
               input => raw_input,
               key =>  raw_key,
               encrypted_data => encrypted_raw );

    RETURN UTL_RAW.CAST_TO_VARCHAR2(encrypted_raw);
    EXCEPTION
      WHEN error_in_input_buffer_length THEN
             dbms_output.put_line('> ' || INPUT_BUFFER_LENGTH_ERR_MSG);
    END;
END;
```

Agenda

- Introduction
- Sécurité entre les clients et le serveur
  - Authentification - Intégrité - Chiffrement (SSL)
  - Web Services - HTTP - WebDAV – FTP - XDB
  - Identification des applications se connectant à la base de données
- Sécurité dans la base de données
  - Concept de sécurité : Compte / mot de passe / rôles / privilèges
  - Auditing : Standard , Triggered, Fine Grained Auditing
  - Chiffrement dans Oracle
  - **Row level security, Label security**

Row Level Security (RLS)

- Depuis 8*i* on peut implémenter une policy de sécurité au niveau du row.
- On crée une fonction retournant une clause de restriction:

```
CREATE OR REPLACE FUNCTION emp_restrict(schema IN varchar2, tab IN varchar2)
 RETURN VARCHAR2
   AS
    BEGIN
     RETURN '''' || sys_context('userenv','session_user') || '''=ename';
    END emp_restrict;
/
```

- On associe cette fonction en tant que policy à une table

```
EXEC
   dbms_rls.add_policy('scott','emp','emp_policy','secusr','emp_restrict');
```

Virtual Private Database - Roles

- Le Virtual Private Database (9*i*) est une extension du Row Level Security
- Jusqu'à Oracle8*i* les rôles peuvent être protégés par un password
- A partir de Oracle9*i* un role peut être lié a une invoker's right stored procedure
- Ceci permet de s'assurer qu'un rôle ne peut être activé que par un programme particulier.

Virtual Private Database - Roles

```
CREATE ROLE sales_role
 IDENTIFIED USING system.sales_security_procedure;
CREATE PROCEDURE sales_security_procedure
 AUTHID CURRENT_USER IS
  network_protocol VARCHAR2(30);
 BEGIN
  SELECT SYS_CONTEXT('USERENV','NETWORK_PROTOCOL')
  INTO network_protocol
  FROM dual;
  IF upper(network_protocol) != 'TCPS'
   THEN
    raise_application_error(-20000,
                            'Only secure connections allowed');
   ELSE
    dbms_session.set_role('sales_role');
  END IF;
END;
/
```

Virtual Private Database

- Label security a été introduit à partir de 9*i* release 1
- Un Label security permet de restreindre l'accès à des données en lecture/écriture seulement aux utilisateurs qui ont les privilèges appropriés
- Le modèle de sécurité a 3 dimensions/components. Le label pour les données ou les utilisateurs est constitués de 3 parties
  - Level (e.g. confidential C, secret S , top secret T)
  - Compartment (e.g. engineering EN, finance FI)
  - Group (e.g. D, CH, FL)
- Les rows avec le label S:EN:CH peuvent être lues que par les utilisateurs ayant le label T:EN:CH, mais pas de ceux ayant C:FI:D
- Complexe a mettre en œuvre!

Conclusions

- Oracle offre un grand nombre de features permettant de :
  - sécuriser la communication entre les clients et les bases de données
  - protéger les données contenues dans la base
  - auditer les accès aux données
- Ne pas utiliser d'installation par défaut (comptes utilisateurs, options etc) pour des environnements de production!