Dans le cadre de

SECURIDAY 2010

Et sous le thème de

Computer Forensics Investigation

VOUS PRÉSENTE L'ATELIER :

Analyse des bases de données

Chef Atelier : Dhikra DABBOUSSI (Réseau Télécom 5)

- Hamza DARGHOUTH (Réseau Télécom 4)
- HADJ TAIEB Abdelbasset (Réseau Télécom 5)
- FATHALLI Seif (Réseau Télécom 4)

1. Présentation de l'atelier

Les serveurs de base de données contiennent des informations financières sensibles et confidentielles, c'est l'objet d'une enquête judiciaire.

Le but de notre atelier est de déterminer en cas d'une attaque sur une base de données, le responsable de l'attaque, comment et quand il a pu accéder à la base, les modifications qu'il a effectuées et éventuellement la correction des modifications non désirables.

2. Environnement logiciel

Pour la réalisation de cet atelier, on a choisi quelques SGBQ qui sont très utilisés dans les entreprises.

1. Oracle 11g :

Oracle est un système de gestion de base de données relationnelle produite par Oracle Corporation, c'est un des SGBD les plus rapide et performant disponible sur le marché.

2. Microsoft SQL Server 2005 Express Edition:

SQL Server est un SGBD relationnel produit par Microsoft, c'est un des concurrents direct d'oracle. Pour ce tutorial on a utilisé la version express qui a l'avantage d'être gratuite et fournit l'ensemble des fonctionnalités dont on a besoin.

3. MySQL 5.1:

MySQL est un SGBD relationnel open source très populaire, il est largement utilisé dans les sites web vu sa licence flexible et son faible cout.

3. Présentation des outils d'investigation

Oracle LogMiner : LogMiner est un utilitaire distribué avec Oracle qui permet de visualiser le contenu des fichiers redo-logs d'oracle qui contiennent l'ensemble de transactions effectuées dans la base.

Oracle Flashback : Flashback est utilitaire Oracle qui travail en tandem avec LogMiner. Cet utilitaire permet l'annulation d'une ou plusieurs transactions sans faire un rollback complet de la base de données.

4. Installation et configuration

1. Oracle 11g :

- **Installation sous Windows vista :**

➢ Lancer l'exécutable setup.exe :

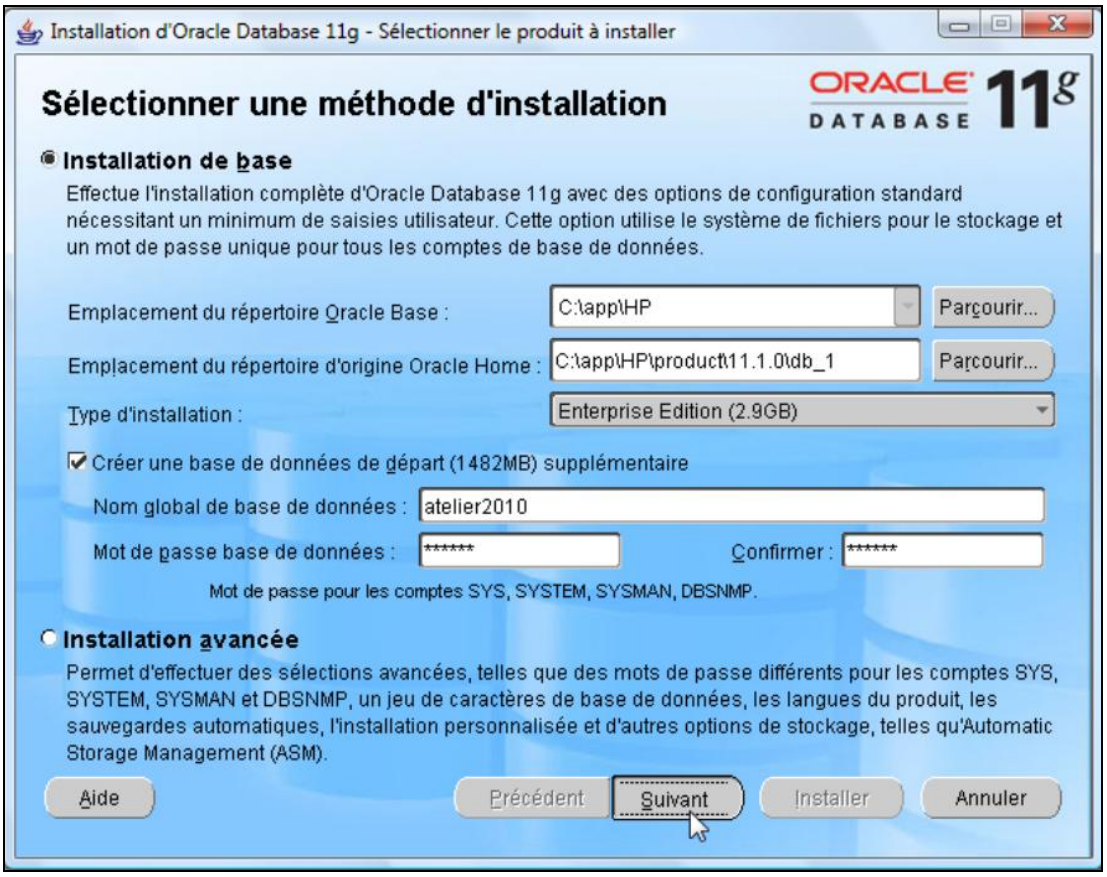

Figure 1:Installation d'oracle

➢ Indiquer les répertoires qui vont accueillir oracle, le nom de la base de données générée et le mot de passe des supers administrateurs de la base.

➢ Ne pas changer les valeurs par défauts pour le reste des étapes de l'installation.

- **Configuration**

➢ Ouvrir SQLplus (contenu dans le répertoire bin d'oracle).

➢ Se connecter avec le compte du super administrateur.

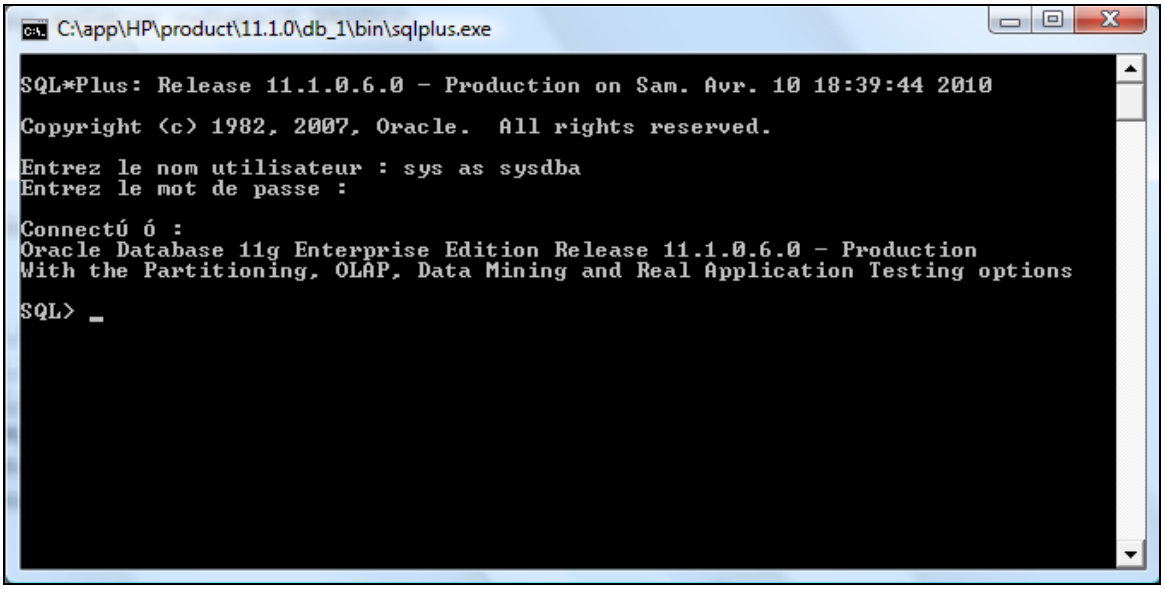

Figure 2:Invite de commande SQLplus

- Exécuter les commandes SQL suivantes pour configurer le format du nom et l'emplacement des fichiers log d'oracle et activer le mode d'audit le plus poussé qui permet d'enregistrer toutes les requêtes exécutées.

```
ALTER system set log_archive_format='redo_%S_%R_%T.arc'
SCOPE=SPFILE;
ALTER SYSTEM SET log_archive_dest_1="LOCATION=c:\logoracle\"
SCOPE=SPFILE;
ALTER SYSTEM SET audit_trail=db_extended SCOPE=SPFILE;
```

- Redémarrer la base de données.

```
shutdown immediate;
startup mount;
```

- Passer la base en mode « archivelog », ce mode permet de journaliser tout les changements qui s'effectue dans la base et évite d'effacer les anciens journaux de la base.

```
alter database archivelog;
alter database open;
```

- Augmenter le niveau de log de la base pour autoriser l'annulation de transaction d'une façon unitaire sans faire de rollback global de la base.

alter database add supplemental log data;
alter database add supplemental log data (primary key) columns;

2. SQL Server 2005:

- **Installation sous Windows server 2003 :**

- Lancer l'exécutable setup.exe :
- Ne pas changer les valeurs par défauts pour le reste des étapes de l'installation.
- Connecter au serveur en utilisant le nom de votre PC comme un nom du serveur et une authentification Windows:

Figure3: Connexion au serveur

- **Configuration**

➢ Activer le modèle recovery

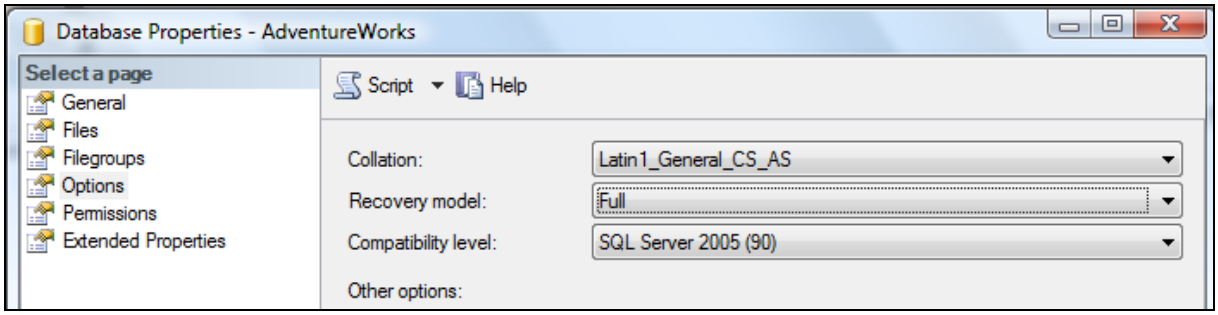

Figure4 : Configuration de la base de données

➢ Trouver le nom logique des bases de données avec la requête SQL suivante

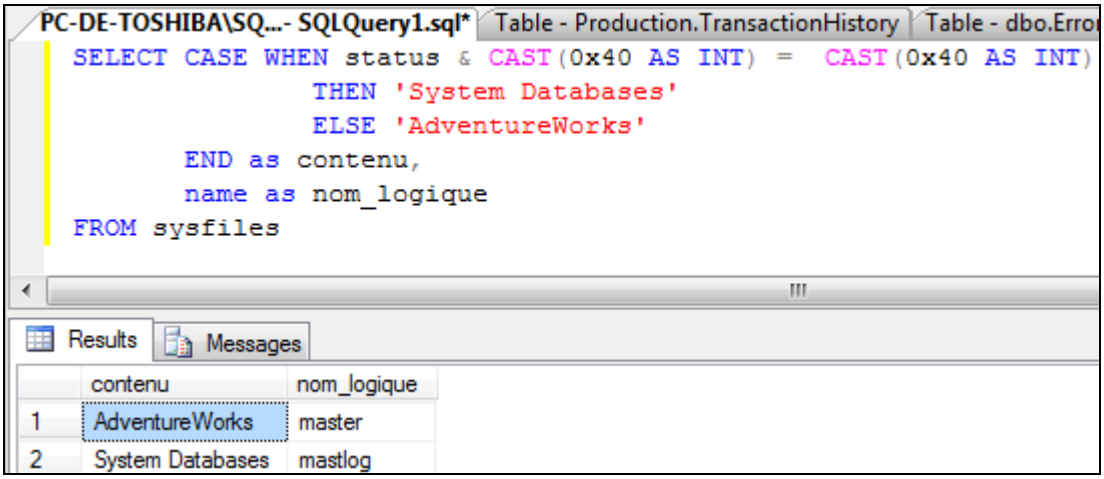

Figure5 : Les noms logiques des bases de données

➢ Activer l'enregistrement du rapport d'erreur

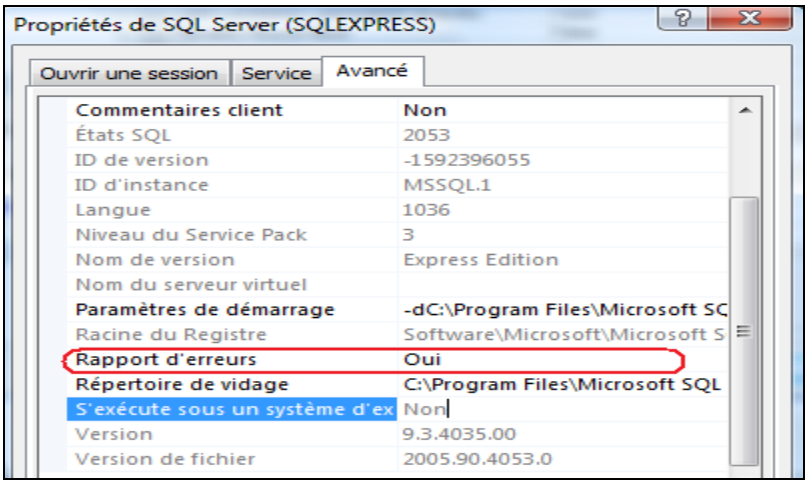

Figure6 : Activer la génération des rapports d'erreurs

3. MySQL 5.1 :

- **Installation sous Windows vista :**

L'installation de mySQL est très simple, il suffit de suivre les étapes guidées et il faut définir le nom de l'administrateur et son mot de passe, on a aussi besoin d'apache pour lancer le myadmin.

- **Configuration :**

Par défaut, MySQL enregistre les logs d'erreurs seulement dans le fichier « logs\mysql_error », il ne permet pas d'enregistrer les transactions.

Alors comme première étape, on va ajouter dans le fichier de configuration de MySQL « mysql\my.ini », la ligne suivante pour enregistrer toutes les transactions effectuées sur la base:

« Log=c:/logs/mysqllog.log » c'est le chemin et le nom du fichier *.log.

```
#log file
log-error=c:/logs/mysql_error.log
log=c:/logs/mysqllog.log
```

5. Scénario de test

Un employé de la société « x » a constaté que tout l'historique des transactions de la société a été effacé, soupçonnant une intrusion dans la base de donnée , le responsable informatique nous a contacter pour vérifier s'il y a eu un accès non autorisé à la base et si c'est le cas , déterminer les actions effectués par le hacker , les annuler et collecter des informations sur l'attaque .

Les 2 tables sensibles dans la base sont la table « t_credit » qui contient les numéros de cartes de crédit des clients de la société et la table « t_historique » qui contient l'historique de toutes les transactions effectuées entre la société et ses clients.

Il n'existe qu'un seul compte administrateur « sys » de la base de données.

- *Commandes exécuté par le pirate :*

```
delete from sys.t_historique;
select * from sys.t_credit;
```

6. Investigation

Au cours d'une enquête, il faut identifier les objets volatiles et non volatiles et collectées toutes informations utiles du système de la victime comme les fichiers logs, le journal des transactions...

1. Sous Oracle 11g

En premier lieu on doit déterminer à partir de quel compte le contenu de la table « t_historique » a été effacé pour faire ceci, on lance l'outil LogMiner (*Disponibilité>Visualiser et gérer les transactions*) et on analyse les transactions qui ont été effectué sur la table « t_historique ».

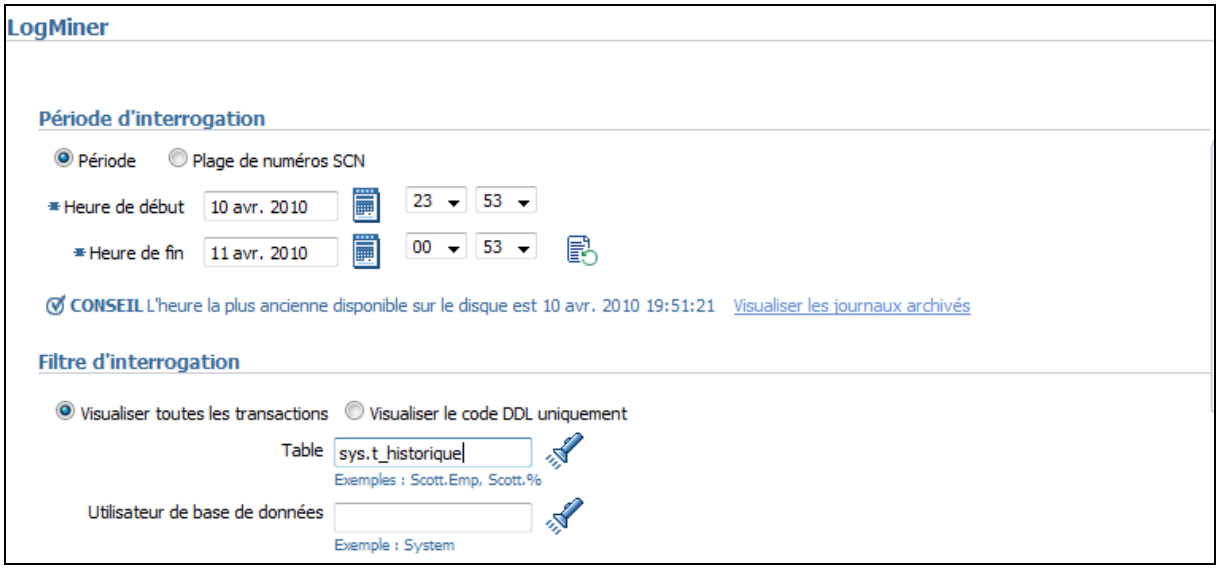

Figure 7:Interface de LogMiner

On indique qu'on veut visualiser les transactions effectuées sur la table « t_historique » ainsi que l'intervalle de temps à examinés et on valide.

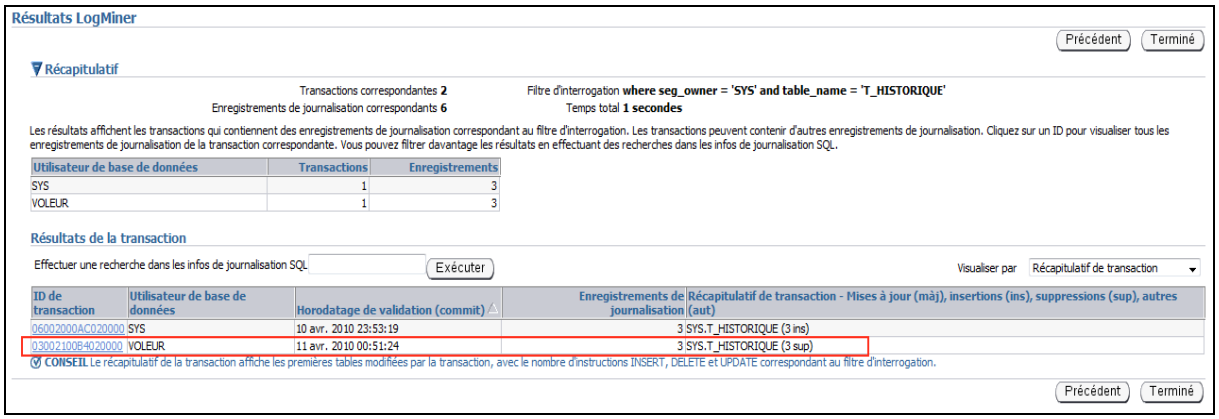

Figure 8:Liste des transactions sur la table

On remarque qu'il y a un compte « voleur » qui ne respecte pas les règles de nommage de compte appliqué par l'administrateur et qui a effectué les requêtes de suppression sur la table.

En cliquant sur l'id de la transaction on obtient une fenêtre qui contient les détails de la transaction. On peut annuler la transaction avec bouton « faire un flashback de la transaction »

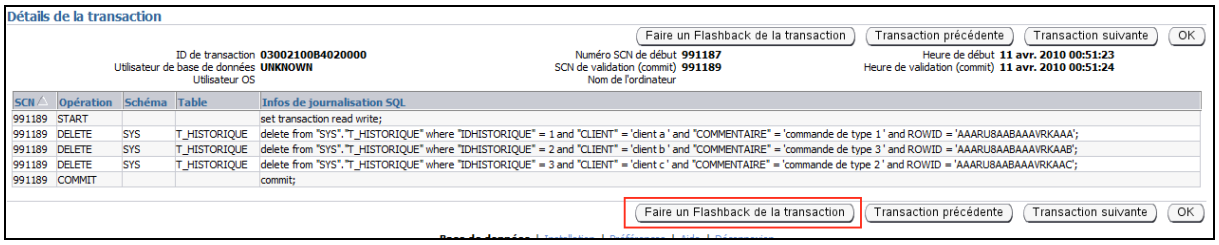

Figure9: Interface de Flashback

Une fois les transactions annulées, on vérifie si le saboteur a exécuté d'autres requêtes qui ne modifient pas les données de la base, on ouvre une feuille et on exécute la requête suivante qui permet de lister les objets visualisé par le compte « voleur » ainsi que les hôtes a partir desquels il s'est connecté

```
Select  OBJ_NAME,ACTION_NAME,SQL_TEXT ,USERHOST,TIMESTAMP
from dba_audit_trail where username='VOLEUR' and
ACTION_NAME='SELECT';
```

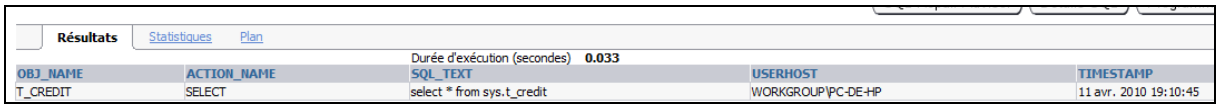

Figure10 : Résultat de l'exécution de la requête

On trouve que le pirate a eu accès à tous les numéros des cartes de crédit enregistrés dans la table.

Il nous reste maintenant à déterminer comment le pirate a pu créer le compte voleur et a partir de quel poste il a accéder a la base.

On exécute la requête suivante pour voir s'il y a eu des tentatives de bruteforce du compte de l'administrateur.

```
SELECT userhost, username, terminal, sessionid, action_name,
to_char(timestamp,'DD-MON-YY HH24:MI:SS') LOGIN, FROM
```

dba_audit_session WHERE action_name='LOGON' and username='SYS' AND returncode='1017';

USERHOST	USERNAME	TERMINAL	SESSIONID	ACTION_NAME	LOGIN
PC-de-HP	SYS	unknown	80506	LOGON	11-AVR. -10 14:53:26
PC-de-HP	SYS	unknown	80507	LOGON	11-AVR. -10 14:53:29
PC-de-HP	SYS	unknown	80508	LOGON	11-AVR. -10 14:53:32
PC-de-HP	SYS	unknown	80509	LOGON	11-AVR. -10 14:53:34
PC-de-HP	SYS	unknown	80510	LOGON	11-AVR. -10 14:53:36
PC-de-HP	SYS	unknown	80511	LOGON	11-AVR. -10 14:53:38
PC-de-HP	SYS	unknown	80513	LOGON	11-AVR. -10 14:53:41
PC-de-HP	SYS	unknown	80514	LOGON	11-AVR. -10 14:53:43
PC-de-HP	SYS	unknown	80515	LOGON	11-AVR. -10 14:53:46
PC-de-HP	SYS	unknown	80516	LOGON	11-AVR. -10 14:53:48
PC-de-HP	SYS	unknown	80517	LOGON	11-AVR. -10 14:53:50
PC-de-HP	SYS	unknown	80518	LOGON	11-AVR. -10 14:53:53

Figure 11: Liste de connexions échouées

On remarque qu'il y a eu un nombre important de connexions échouées dans un petit intervalle de temps on en déduit que le compte de l'administrateur a subit une attaque brute force à partir du hôte « pc-de-hp ».

2. Sous SQL Server 2005 Express Edition

Dans SQL Server les preuves référentielles qu'on a besoin pour notre enquête sont:

- Les fichiers de base de données : C:\ProgramFiles\MicrosoftSQLServer\MSSQL.1\MSSQL\Data*.mdf
- Les fichiers log la base de données: C:\ProgramFiles\MicrosoftSQLServer\MSSQL.1\MSSQL\Data*.ldf (Le nom du fichier log est sous la forme : nombase_log) Le fichier journal (. ldf) détient toutes les données nécessaires pour inverser les transactions et de récupérer la base de données
- Les fichiers de traces : C:\ProgramFiles\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\.*.TRC
- Les logs des erreurs: C:\ProgramFiles\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\ERRORLOG

Les informations sauvegardées dans ces fichiers:

- Les tentatives de connexion réussies ou non
- L'historique des transactions
- Sauvegarde et restauration des informations
- Base de données (sp_dboption) et les options de serveur (sp_configure)
- Messages d'erreur

Le but de notre investigation est de déterminer quel utilisateur a fait l'effacement des données de la table, après on cherche les modifications faite par cet utilisateur (IDtransaction, IDuser…).
Comme toute enquête, il faut voir toutes les preuves et trouver une relation entre elles, on va voir surtout le champ de l'exécution des transactions, par quel utilisateur….

- Les logs d'erreur et les fichiers de traces

En premier lieu, on va consulter le fichier log des erreurs et voilà un extrait:

```
2010-04-17 06:13:16.91        Erreur : 18452, Gravité : 14, État : 1.
2010-04-17 06:13:16.91        Login failed for user 'admin'. The user is not associated with a trusted SQL Server connection. [CLIENT : <local machine>]
2010-04-17 06:13:26.60        Erreur : 18452, Gravité : 14, État : 1.
2010-04-17 06:13:26.60        Login failed for user 'superadmin'. The user is not associated with a trusted SQL Server connection. [CLIENT : <local mach
2010-04-17 06:13:31.41        Erreur : 18452, Gravité : 14, État : 1.
2010-04-17 06:13:31.41        Login failed for user 'aa'. The user is not associated with a trusted SQL Server connection. [CLIENT : <local machine>]
2010-04-17 06:13:35.79        Erreur : 18452, Gravité : 14, État : 1.
2010-04-17 06:13:35.79        Login failed for user 'ab'. The user is not associated with a trusted SQL Server connection. [CLIENT : <local machine>]
```

Figure11 : Suite de tentatives de connexion échouées (ERRORLog)

Avec SQL Server Profiler, on peut aussi consulter les fichiers de traces :

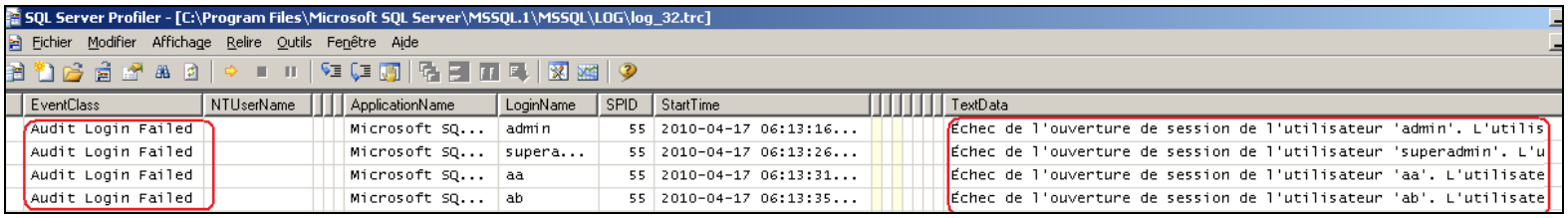

Figure12 : Suite de tentatives de connexion échouées (log.trc : fichier de trace)

On remarque qu'il ya une succession d'échecs de tentatives de connexion en utilisant plusieurs noms d'utilisateur (admin, superadmin, aa…) à des instants très proches provenant de la même adresse IP (ici c'est local). On peut conclure qu'il s'agit d'une attaque par brute force, ce point sera le début de notre enquête.
Après cette suite d'échec de connexion, on remarque un succès de connexion avec la même adresse IP, on peut confirmer qu'une intrusion a eu lieu par cette personne.
On remarque aussi qu'un nouvel utilisateur «voleur» non reconnu dans les dossiers de l'entreprise, réussi à se connecter à la base.
On peut conclure que le pirate en réussissant la connexion à la base, il a créé un nouvel user qui va l'utiliser pour son attaque.

- Journal des transactions

Maintenant on va consulter le journal des transactions en utilisant la requête suivante :

```
select [current LSN], Operation,Context, [Transaction ID], [Begin Time], [Transaction
Name], [Transaction SID]
from ::fn_dblog(null,null)
```

Le résultat de cette requête est comme suit :

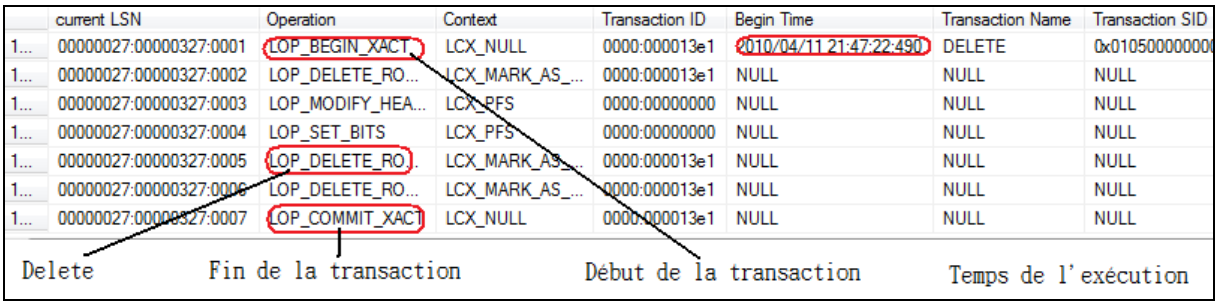

Figure13 : La transaction enregistrée dans le fichier log

La transaction ID 13e1 a été exécutée en réponse à la commande DELETE.
On remarque que l'heure du début de l'exécution de cette transaction correspond à la période de connexion de l'utilisateur « voleur », avec les champs SID, UserID...
On voit très bien que la transaction effectuée est « DELETE ».

- Le plan cache

Le plan cache permet de surveiller l'utilisation de la mémoire par SQL Server pour stocker des objets comme des transactions, des intrusions...
Maintenant pour voir la transaction effectuée, on va consulter le « plan cache » de la base en exécutant la requête suivante :

```
select * from sys.dm_exec_cached_plans cross apply
sys.dm_exec_sql_text(plan_handle)
```

On trouve dans la cache de la base de données l'ensemble des requêtes lancées, on peut voir ici la requête qui a causé l'effacement des données de la table « t_hitorique ».

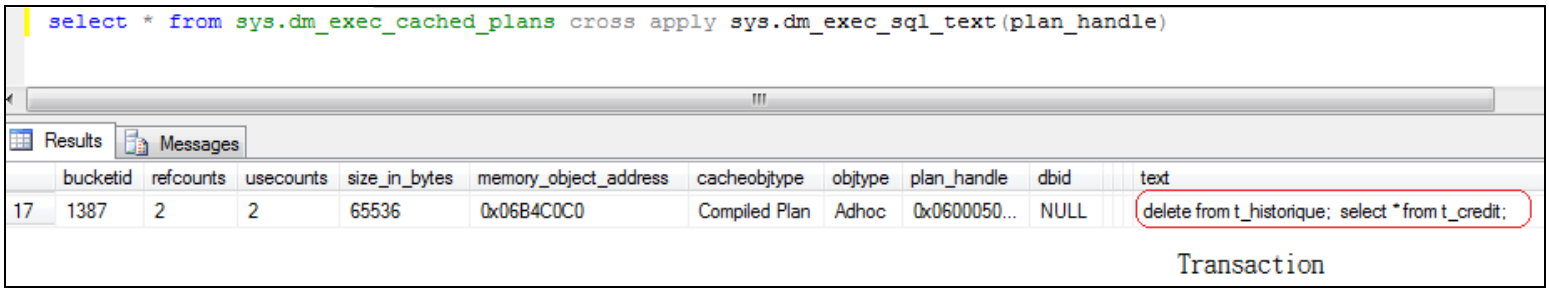

Figure14: La requête enregistrée dans le cache plan

On doit annuler la transaction « delete » afin récupérer la table.
Avec la requête suivante on peut annuler la transaction DELETE:

```
ROLLBACK TRANSACTION DELETE
```

Si l'entreprise fait des sauvegardes de sa base à part (un backup):

```
Backup Database nom_base
```

3. Sous MySQL 5.1

En premier, on va consulter le fichier log des transactions :mysqllog.log

On obtient les traces suivantes :

- Une suite de tentatives de connexion presque au même instant ou à des intervalles de temps très réduit :

```
100412  1:34:01      86 Connect     Access denied for user 'administrateur'@'localhost' (using password: YES)
100412  1:34:20      87 Connect     Access denied for user 'administrator'@'localhost' (using password: YES)
100412  1:34:54      88 Connect     Access denied for user 'root'@'localhost' (using password: YES)
```

Figure14 : Suite de tentatives de connexion échouées

On peut conclure ici d'une possibilité de brute force.

- Une transaction d'ajout d'utilisateur

```
100412  1:06:43      42 Connect     root@localhost on
                     42 Query       SELECT VERSION()
                     42 Query       SET NAMES utf8
                     42 Query       SET collation_connection = 'utf8_unicode_ci'
                     42 Query       SET NAMES utf8
                     42 Query       SET collation_connection = 'utf8_unicode_ci'
                     42 Query       CREATE USER 'voleur'@'localhost' IDENTIFIED BY 'bonjour'
```

Figure15 : Requête d'ajout d'un nouvel utilisateur

- Les transactions exécutées par le pirate avec le compte « voleur»:

```
100412  1:44:23     102 Connect     voleur@localhost on
100412  1:44:23      82 Connect     root@localhost on
             82 Query       SELECT VERSION()
             82 Query       SET NAMES utf8
             82 Query       SET collation_connection = 'utf8_unicode_ci'
             82 Query       SET NAMES utf8
             82 Query       SET collation_connection = 'utf8_unicode_ci'
             82 Query       SHOW SESSION VARIABLES LIKE 'collation_connection'
             82 Query       SHOW SESSION VARIABLES LIKE 'character_set_connectior
             82 Query       SHOW CHARACTER SET
             82 Query       SHOW COLLATION
             82 Query       SELECT COUNT(*) FROM mysql.user
             82 Query       SHOW GRANTS
             82 Init DB     sys
             82 Query       delete from sys.t_historique
             82 Query       SELECT COUNT(*) FROM mysql.user
             82 Init DB     sys
             82 Query       select * from sys . t_credit
```

Figure16 : Requêtes exécutées par l'utilisateur « voleur »

On arrive à voir exactement les requêtes exécutées et par quel utilisateur.

Avec MySQL on peut trouver toutes les traces pour l'investigation mais on ne peut pas annuler la transaction.

4.Conclusion

Dans ce tutorial, nous avons abordé les aspects fondamentaux d'investigations liés aux principaux systèmes de gestions de bases de données du marché. Mais, il est important de noter que non seulement la base de données est l'objet de l'investigation, il y a les logs de firewall ou IDS qui peuvent donner de nouvelles preuves pour l'enquête.