Oracle Database Vault: Design Failures

What is Database Vault?

- Helps protecting against insider threats even when these comes from privileged database users (SYS)
- Mandatory in certain countries: laws
- Can be considered as a war declaration against many DBAs...

Design Failures

- “Database Vault” administrator and auditor
- Operative System Level
- File System Level
- RDBMS level
- The TNS protocol

Database Vault's administrator and auditor

- The most obvious failure (if it can be considered a failure...)
  - Who controls the police?
  - Who should be the responsable?
    - And who controls the one who controls the auditors and administrators
    - Another department who controls the departament that controls the department who controls the department...?

Failures: Operative System Level

- Fact: Database system runs as only one operative system user
  - Oracle under Unix/Linux
  - Local System under Windows
- Database Vault's auditor, administrator, database's administrator and final users, all of they, runs their queries in the same user space owned by the user who runs the database

Failures: Operative System Level

- Fact: Database administrator can trojanize the database at operative system level
  - libclntsh.so (or .dll)
  - A trojan version of the TNS Listener or, quicker, a proxy between the end user and the real TNS Listener
  - A trojanized Oci library
  - Any Oracle component can be trojanized

Failures: Operative System Level

- Fact: DBA has Oracle or Local System privileges in the operative system
  - (S)he can attach with a debugger to any oracle process and record all operations
  - Set function and/or address breakpoints and modify the common way the database system works
  - Can change local or global variables, the uid of a running SQL session, etc...
  - Can do whatever (s)he wants...

Failure: Filesystem Level

- Fact: DBA has file system access
  - Able to read or write datafiles in raw mode
  - There are many libraries and tools to do it
    - Data Unloader
      - Oracle's own tool
    - DUDE (Database Unloading by Data Extraction)
      - http://www.ora600.nl/introduction.htm

Failures: Filesystem Level

- Fact: DBA can do a backup. (S)he can copy the complete database to any other disk or machine
  - RMAN
  - ALTER TABLESPACE *XXX* BEGIN BACKUP
- Can reimport complete database
  - EXP/IMP doesn't work as expected but...
  - (S)he can use RMAN
  - Do a manual recover: damage one datafile and put the manipulated version to recover
    - Hard but possible

Failures: Filesystem Level

- Problems
  - Recovering and/or editing a large datafile can be very hard and would be a reason to audit the complete database (by the police, of course)
  - The data stored in the datafiles may be encrypted

Failures: Filesystem Level

- Solutions
  - Trojanize the database if the encryption mechanism is in the database (i.e., PL/SQL)
  - An attacker (DBA) can wait for a system failure to apply the changes made in a datafile without making it a suspicious think
    - You wil always found a system failure
- You will always found a solution, if you're the DBA or the system administrator you're god :)

Fallas: Database System

- Is hard to install a trojanized PL/SQL package when database vault is installed
- Install the trojan prior to install database vault option :)
  - While DBA is doing the testing
- But... What can be trojanized?

Failures: What to trojanize?

- DBMS_OBFUSCATION_TOOLKIT
- *_USERS, *_PRIVS Views
- DBMS_STANDARD, in example...
- The installer and installer's scripts
  - Database vault's own scripts ;)

Failures: Backdoors

- A “wrapped” (to *hide* the code) PL/SQL package during database vault install
  - To escalate privileges
  - To remove any evidence of an attack
  - To simply subvert Database Vault's behaviour

Failures: Again, trojanize at OS level

- We can trojanize at OS level
  - As explained in other chapter
    - libclntsh.(so|dll)
    - oracle[.exe]
    - libocci.[so|dll]
    - libnnz*XX*.[so|dll]
    - extjob[.exe]
    - sqlplus[.exe]

Failures: Hooks

- Every time you applies a patch you should reapply the trojan
  - But you can trojanize the “rebuild” script...
- Is better to write a tool to hook interesting Oracle functions
  - oci_prepare_stmt, in example?
  - Any of the kk* internal functions

Failures: TNS Protocol

- There are various rule sets that allows or denies the privilege to do something if you connect from some domain or ip address:
  - IP Address, OS username, program, machine, etc... Are fully user controllable
- They are simply strings in a TNS Packet
  - NV strings
  - Not trusted

Failures: TNS Protocol

- An example TNS packet's NV string :
  - (CONNECT_DATA=(CID=(PROGRAM=himom.exe)(HOST=192.168.1.5)(USER=oracle))(COMMAND=connect)(ARGUMENTS=64)(SERVICE=LISTENER)(VERSION=169869568))
- OS username and ip address are fully controllable by an attacker
  - As well as many other options...
  - They are fields of a TNS packet

Conclusions

- Interesting product but...
  - I think that is unreal
  - Has no privilege separation at os level
  - Root or System can do whatever s(he) wants
  - You can't hide nothing to the kernel and the root/system may alter the kernel behaviour without being noticed by final users
    - To subvert database's behaviour, i.e.

Possible solutions

- Administrator or root shouldn't have privileges to do whatever (s)he want, otherwise, (s)he is able to “attack” the database system
  - Note the quotes (root attacking the system...)
  - Google like question: What is broken in Unix?
- Privilege separation at os level, by creating different users and groups for different task is fundamental
  - Remember: All run in the same user space

End

Send comments, questions, criticisms, insults, threats, invitations for sex or for a drink to:

joxeankoret@yahoo.es