UKOUG UNIX SIG – DEMOS

These are the demos for the two part talk Unix SIG, September 8th TVP

Prep

1. Run file in c:\jdbc setup.bat
2. As SYSTEM Clean importer – clean_imp.sql
3. As ORABLOG – run updcc.sql

Demo 1 – exploit

Demo running the importer hack

1. Find a suitable user
2. Assume we have access as orascan
3. Run who_has_priv for BECOME USER
4. Run who_can_access for SYS.KUPP$PROC
5. We can choose IMPORTER or SUDO
6. Simply try and connect – assume we can
7. Show privs, run check_priv.sql
8. Run check.sql
9. Select password from sys.user$ - FAILS
10. Exec kupp$proc.change_user
11. Run check.sql again
12. Grant dba to importer
13. Select user$

Demo 2 – realistic hack

1. Explain simple setup of machines
2. Nmap is pointless for this demo – ip is 192.168.254.2
3. Show lsrnctl
    a. Set current_listener 192.168.254.2 –
    b. Show status – fails but we know its there as we got auth error
    c. Show version
    d. Issue is default port – we know port is 1521
4. Show sid guess – find ORCL
5. Sho ora-user-enum
6. Now we have most details
7. Try and connect as dbsnmp
8. Finding another user is harder but can be done with time,
9. Look for credit cards
    a. First table called “%CREDIT%” – select owner,table_name from dba_tables where table_name like %CREDIT%

10. Look at the data, its encrypted
11. Look for crypto package
    a. Run dep1.sql
12. Select the data –

```
SQL> select orablog.orablog_crypto.decrypt(orablog.credit_card.pan) from orablog.credit_card;

ORABLOG.ORABLOG_CRYPTO.DECRYPT(ORABLOG.CREDIT_CARD.PAN)
--------------------------------------------------------------------------------
4049877198543457
3742345698766678
4049657443219878
3742112366758976
4049990855468731
```

Stop there

Demo 3 – evidence?

Does the audit trail show any evidence?

1. Connect orascan/orascan – show we are a DBA
2. When do we look? Now, an hour ago?, last week, by which user?, audit can only ever be based on “we want to know something”
3. Its hard – REMEMBER WE DONT KNOW WHAT WE ARE LOOKING FOR
4. Run check_aud.sql
5. Run check_aud_obj.sql
6. Run aud.sql and get session id
7. Run SQL> exec print_table('select * from dba_audit_object where sessionid=''513498''');
8. Find the listener log – any evidence?
9. Connect root/oracle, lsnrctl, edit the file
10. Auditing key data, key procedures and also core procedures is useful
11. Finding evidence is hard as it doesn’t say “THIS IS EVIDENCE!”
12. REMEMBER THE ATTACKER CAN LOOK AT AUDIT SETTINGS AND DECIDE WHAT ROUTE TO TAKE TO THE DATA

Demo 4 – Stealth attacks

We need to look at widening the scope, finding ways to avoid audit

1. Assume connected as DBSNMP or a users account
2. Run cracker-v2.0.sql – choose a better account based on knowing the password
3. Check privileges check_priv.sql and find_all_privs.sql
4. Also look at ALL_USERS if cracker doesn’t work
5. From attackers perspective – check audit check_aud.sql and check_aud_obj.sql
6. Check v$session with sess.sql concocted of course

7. Could escalate – via become user or other method? – simply findining a better user is escalating
8. Can spoof connection, avoid audit...
9. Spoof connection using Java

C:\00_00_ukoug\jdbc>java DBC jdbc:oracle:thin:@192.168.254.2:1521:orcl orascan orascan 1

=>Made Up User :=> Made Up Program

C:\00_00_ukoug\jdbc>java DBC jdbc:oracle:thin:@192.168.254.2:1521:orcl orascan orascan 0

=>Pete :=> JDBC Thin Client

C:\00_00_ukoug\jdbc>

10. Find data on OS, look at cards.lis
11. Get data from SGA – cc.sql

Demo 5 – True access to data

1. Start with CREDIT_CARD
2. Run who_can_access.sql run get_tab2.sql
3. Run dep.sql
4. Recurse
5. Look for copies of data – table_name,column_name from dba_tab_columns – find copies
6. Run get_tab2, dep
7. Finally run get_data.sql

Demo 6 – Analysis of users

Simple check of use.sql

And cracker-v2.0.sql

And profiles.sql