NGS Consulting
Next Generation Security Software Ltd.

Oracle Forensics
–
Dissection of an Oracle Attack (Talk and Demo)

David Litchfield
(davidl@ngssoftware.com)

Note to reader:

- These slides were compiled on 29th June 2007
- They're liable to change from now 'til when I give the presentation
- Cheers! David

Why Oracle Forensics?

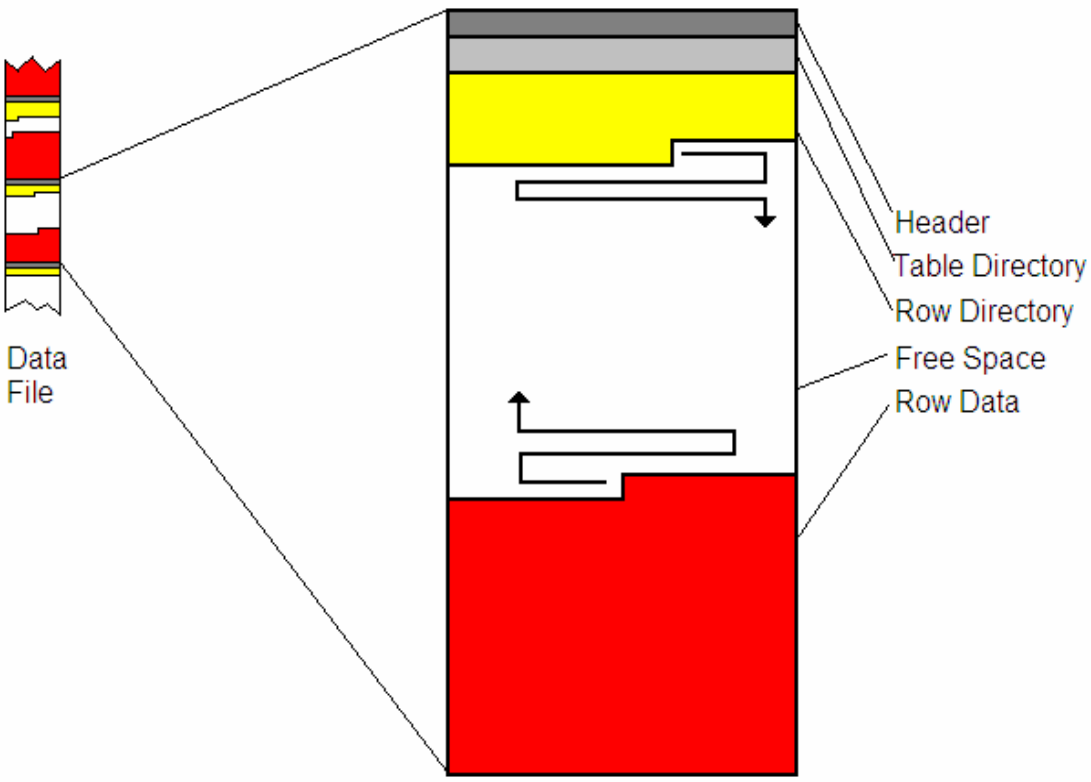
- Since the state of California passed the Database Security Breach Notification Act (SB 1386) in 2003 another 34 states have passed similar legislation with more set to follow.
- In January 2007 TJX announced they had suffered a database security breach with 45.6 million credits card details stolen – the largest known breach so far.
- In 2006 there were 335 publicized breaches in the U.S.; in 2005 there were 116 publicized breaches; between 1st January and March 31st of 2007, a 90 day period, there have been 85 breaches publicized.
- There are 0 (zero) database-specific forensic analysis and incident response tools on the market – free or commercial.

Where is the evidence?

- Evidence of a compromise can be found in many places
  - TNS Log files
  - Trace files
  - Redo Logs
  - Datafiles
  - Apache logs (Oracle Application Server)
- This talk and demo specifically covers the datafiles, redo logs and Apache logs.
- In the essence of time we'll be cutting out several parts of the forensic process which you wouldn't do in a real scenario of course!
- To start with we'll look at an Oracle Data Block

Oracle Data Block

- Header
  - Object ID (25th Byte)
  - Checksum
- Row Directory
  - Each row has
  - 2 byte entry
  - pointing to offset



Oracle Data Block…row of data

- Consists of a 3 byte Row Header
  - Byte 1: Flags to indicate row state
    - If row of data has been deleted the 5th bit of 1st byte (Flags) is set – e.g. 0x2C becomes 0x3C
  - Byte 2: Lock Status
  - Byte 3: Number of columns

Oracle Data Block…row of data

```
189d3790h:                                              2C 01 11
189d37a0h: 04 C3 06 13 2F 04 C3 06 13 2F 02 C1 37 0D 4D 59
189d37b0h: 5F 54 45 4D 50 5F 54 41 42 4C 45 02 C1 02 FF 02
189d37c0h: C1 03 07 78 6B 03 17 12 08 38 07 78 6B 03 17 12
189d37d0h: 08 38 07 78 6B 03 17 12 08 38 02 C1 02 FF FF 01
189d37e0h: 80 FF 02 C1 07 02 C1 02
```

Oracle Data Block…row of data

```
Col 1           04 C3 06 13 2F
Col 2           04 C3 06 13 2F
Col 3           02 C1 37
Col 4           0D 4D 59 5F 54 45 4D 50 5F 54 41 42 4C 45
Col 5           02 C1 02
Col 6           FF
Col 7           02 C1 03
Col 8           07 78 6B 03 17 12 08 38
Col 9           07 78 6B 03 17 12 08 38
Col 10          07 78 6B 03 17 12 08 38
Col 11          02 C1 02
Col 12          FF
Col 13          FF
Col 14          01 80
Col 15          FF
Col 16          02 C1 07
Col 17          02 C1 02
```

Oracle Data Block…row of data

```
04 C3 06 13 2F = ((6-1)*10000) + ((19-1)*100) + (47-1) = 51846
04 C3 06 13 2F = ((6-1)*10000) + ((19-1)*100) + (47-1) = 51846
02 C1 37 = 55
0D 4D 59 5F 54 45 4D 50 5F 54 41 42 4C 45 = MY_TEMP_TABLE
02 C1 02 = 1
FF = NULL
02 C1 03 = 2
07 78 6B 03 17 12 08 38 = 23/03/2007 17:07:55
07 78 6B 03 17 12 08 38 = 23/03/2007 17:07:55
07 78 6B 03 17 12 08 38 = 23/03/2007 17:07:55
02 C1 02 = 1
FF = NULL
FF = NULL
01 80 = 0
FF = NULL
02 C1 07 = 6
02 C1 02 = 1
```

Locating Dropped Objects

To locate dropped objects we need to know what happens when an object is created:

- A row is entered in the OBJ$ table, I_OBJ1, I_OBJ2, I_OBJ3 indexes
- Depending upon object

  TAB$, COL$ for table objects

  SOURCE$, IDL_UB1$, IDL_CHAR$ for functions
- Information about new objects scattered all over the datafile.

Locating Dropped Objects

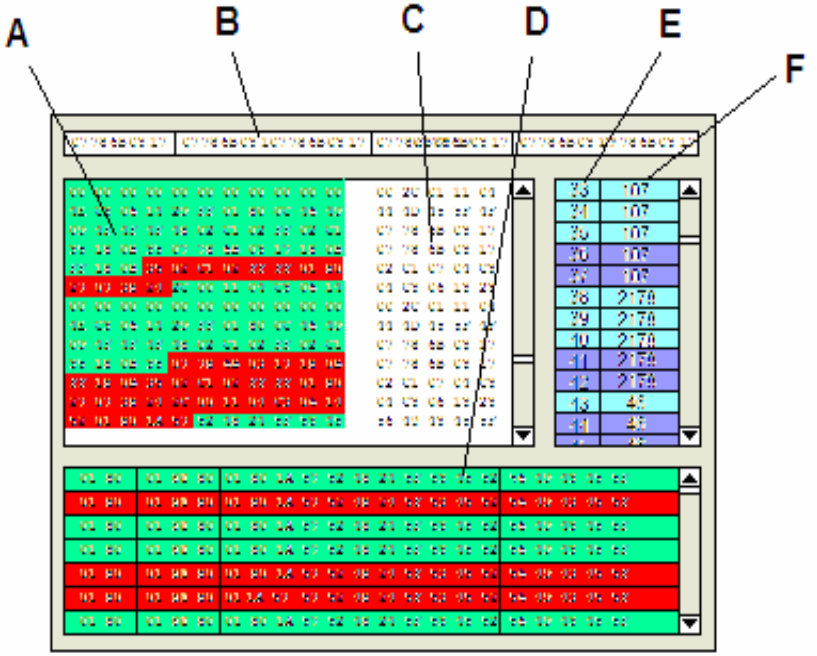
- Open datafile that has the SYSTEM tablespace
- Locate all blocks with object ID of 18 – object ID of the OBJ$ table.
- Follow each entry in the row directory
  - Some of these will point to “live” (0x2C) rows
  - Others “deleted” (0x3C)
  - All data that has not been “blocked out” is deleted data – may only be fragments though!
- Rinse and Repeat for all “interesting” object IDs – e.g. SOUCRCE$,

```
1d398000h: 06 A2 00 00 CC E9 40 00 D2 9A 0F 00 00 00 01 06 ; .¢..Ìé@.Òš......
1d398010h: 12 F4 00 00 01 00 00 00 48 00 00 00 C4 9A 0F 00 ; .ô......H...Äš..
1d398020h: 00 00 00 00 02 00 03 00 CD E9 40 00 03 00 0A 00 ; ........Íé@.....
1d398030h: 4A 02 00 00 BB 05 80 00 BB 01 04 00 09 20 59 01 ; J...».€.».... Y.
1d398040h: D2 9A 0F 00 03 00 0B 00 4A 02 00 00 BA 05 80 00 ; Òš......J...°.€.
1d398050h: BB 01 0E 00 00 80 00 00 74 9A 0F 00 00 01 39 00 ; »....€..tš....9.
1d398060h: 20 00 84 00 AB 11 B5 16 20 18 00 00 39 00 25 1D ;  .„.«.µ. ...9.%.
1d398070h: 84 1D 92 1D C3 1D 31 1E 5C 1E BB 1E C9 1E E2 1E ; „.'.Ã.1.\.».É.â.
1d398080h: F0 1E 61 1F 79 1C C4 1C D2 1C FD 1C 0B 1D 20 1C ; ð.a.y.Ä.Ò.ý... .
1d398090h: CC 1B 1C 1B 4B 1B 5D 1B BC 1B 2D 17 66 17 89 17 ; Ì...K.].¼.-.f.‰.
1d3980a0h: B4 17 C6 17 FF 17 13 18 30 18 16 18 4B 18 21 00 ; ´.Æ.ÿ...0...K.!.
1d3980b0h: 22 00 23 00 24 00 25 00 26 00 27 00 2E 00 6B 13 ; ".#.$.%.&.'...k.
1d3980c0h: 7A 13 A5 13 B7 13 3D 14 51 14 FF FF 6B 14 AB 11 ; z.¥.·.=.Q.ÿÿk.«.
1d3980d0h: E4 11 04 12 13 12 3E 12 50 12 D8 12 EC 12 06 13 ; ä.....>.P.Ø.ì...

1d399200h: 00 00 00 00 00 00 00 3C 01 03 04 C3 06 13 33 02 ; .......<...Ã..3.
1d399210h: C1 02 2D 46 55 4E 43 54 49 4F 4E 20 45 58 54 52 ; Á.-FUNCTION EXTR
1d399220h: 41 43 54 5F 53 59 53 5F 50 41 53 53 57 4F 52 44 ; ACT_SYS_PASSWORD
1d399230h: 20 52 45 54 55 52 4E 20 56 41 52 43 48 41 52 0A ;  RETURN VARCHAR.
1d399240h: 3C 01 03 04 C3 06 13 33 02 C1 03 14 41 55 54 48 ; <...Ã..3.Á..AUTH
1d399250h: 49 44 20 43 55 52 52 45 4E 54 5F 55 53 45 52 0A ; ID CURRENT_USER.
1d399260h: 3C 01 03 04 C3 06 13 33 02 C1 04 02 49 53 0A 3C ; <...Ã..3.Á..IS.<
1d399270h: 01 03 04 C3 06 13 33 02 C1 06 1F 50 52 41 47 4D ; ...Ã..3.Á..PRAGM
1d399280h: 41 20 41 55 54 4F 4E 4F 4D 4F 55 53 5F 54 52 41 ; A AUTONOMOUS_TRA
1d399290h: 4E 53 41 43 54 49 4F 4E 3B 0A 3C 01 03 04 C3 06 ; NSACTION;.<...Ã.
1d3992a0h: 13 33 02 C1 06 06 42 45 47 49 4E 0A 3C 01 03 04 ; .3.Á..BEGIN.<...
1d3992b0h: C3 06 13 33 02 C1 07 7C 45 58 45 43 55 54 45 20 ; Ã..3.Á.|EXECUTE
1d3992c0h: 49 4D 4D 45 44 49 41 54 45 20 27 49 4E 53 45 52 ; IMMEDIATE 'INSER
1d3992d0h: 54 20 49 4E 54 4F 20 53 43 4F 54 54 2E 4D 59 5F ; T INTO SCOTT.MY_
1d3992e0h: 54 45 4D 50 5F 54 41 42 4C 45 20 56 41 4C 55 45 ; TEMP_TABLE VALUE
1d3992f0h: 53 20 28 28 53 45 4C 45 43 54 20 50 41 53 53 57 ; S ((SELECT PASSW
1d399300h: 4F 52 44 20 46 52 4F 4D 20 53 59 53 2E 44 42 41 ; ORD FROM SYS.DBA
1d399310h: 5F 55 53 45 52 53 20 57 48 45 52 45 20 55 53 45 ; _USERS WHERE USE
1d399320h: 52 4E 41 4D 45 20 3D 20 27 27 53 59 53 27 27 29 ; RNAME = ''SYS'')
1d399330h: 29 27 3B 0A 3C 01 03 04 C3 06 13 33 02 C1 08 08 ; )';.<...Ã..3.Á..
1d399340h: 43 4F 4D 4D 49 54 3B 0A 3C 01 03 04 C3 06 13 33 ; COMMIT;.<...Ã..3
1d399350h: 02 C1 09 0E 52 45 54 55 52 4E 20 27 46 4F 4F 27 ; .Á..RETURN 'FOO'
1d399360h: 3B 0A 3C 01 03 04 C3 06 13 33 02 C1 0A 04 45 4E ; ;.<...Ã..3.Á..EN
```

F.E.D.S. (Forensic Examiner's Database Scalpel)

A: Hex Dump
B: Block Info
C: ASCII View
D: Row Data
  Green: "live"
  Red: Deleted
E: Block Number
F: Object ID for block



F.E.D.S. is still in design and research stage

Oracle Redo Logs

Binary file that keeps a record of changes (called redo entries) so in the event of a database failure all actions can be redone.

Redo Entry

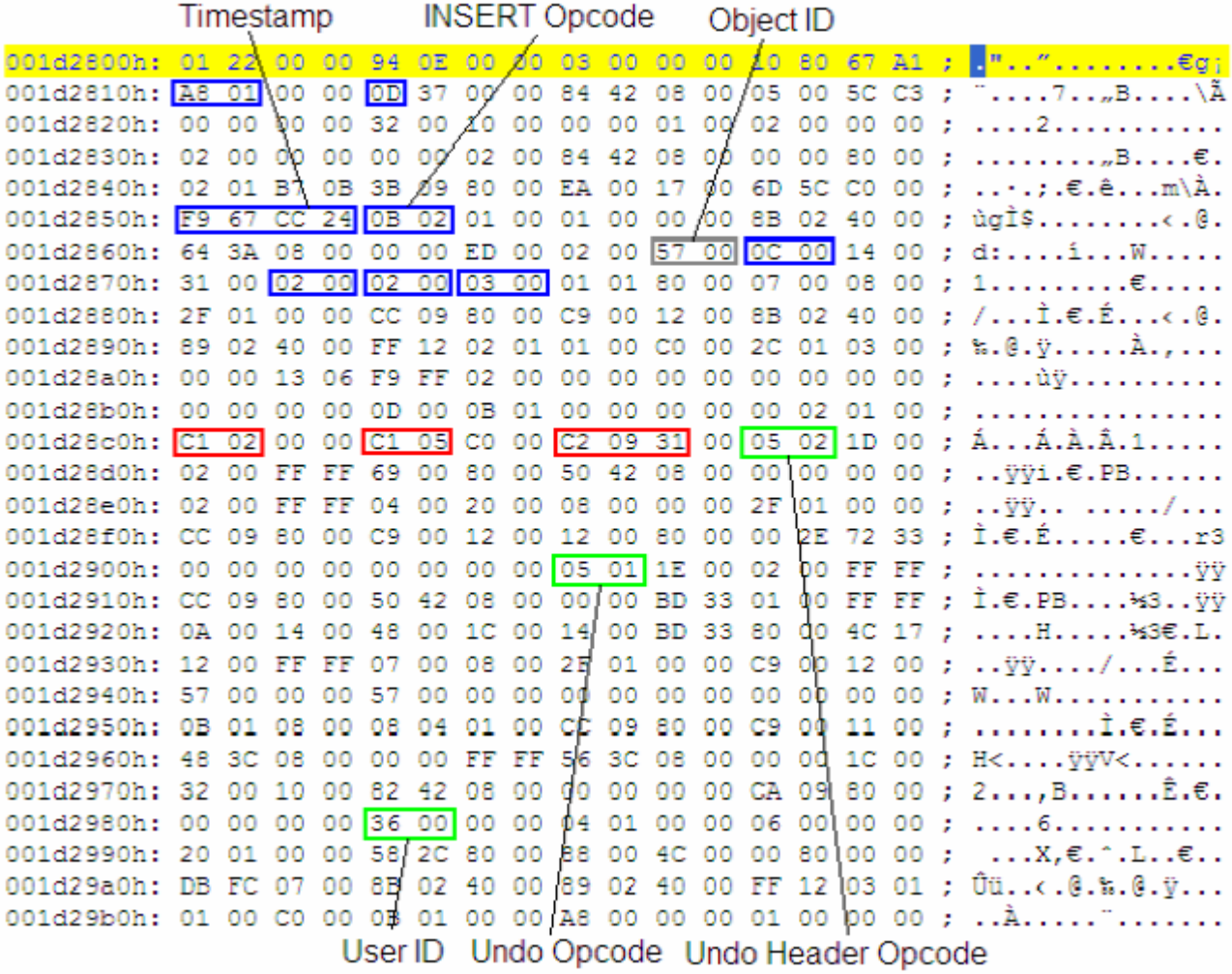
- Contains all changes for a given SCN (System Commit number)
- Header and one or more change vectors

Change Vector operation codes

- 5.1 Undo Record
- 5.4 Commit
- 11.2 INSERT on single row
- 11.3 DELETE
- 11.5 UPDATE single row
- 11.11 INSERT multiple rows
- 11.19 UPDATE multiple rows
- 10.2 INSERT LEAF ROW
- 10.4 DELETE LEAF ROW
- 13.1 Allocate space [e.g. after CREATE TABLE]
- 24.1 DDL

We can use these to determine what actions were taken

INSERT Example



Time stamp

Timestamp is when the redo entry was written – not when the action was taken.

Records to the second from midnight of 1st January 1988.

Demo followed by questions

Any questions?

NGS Consulting
Next Generation Security Software Ltd.

Thank You

http://www.ngsconsulting.com/