Emergis | assure SECURITY

SQL Server Database Forensics

Kevvie Fowler, GCFA Gold, CISSP, MCTS, MCDBA, MCSD, MCSE

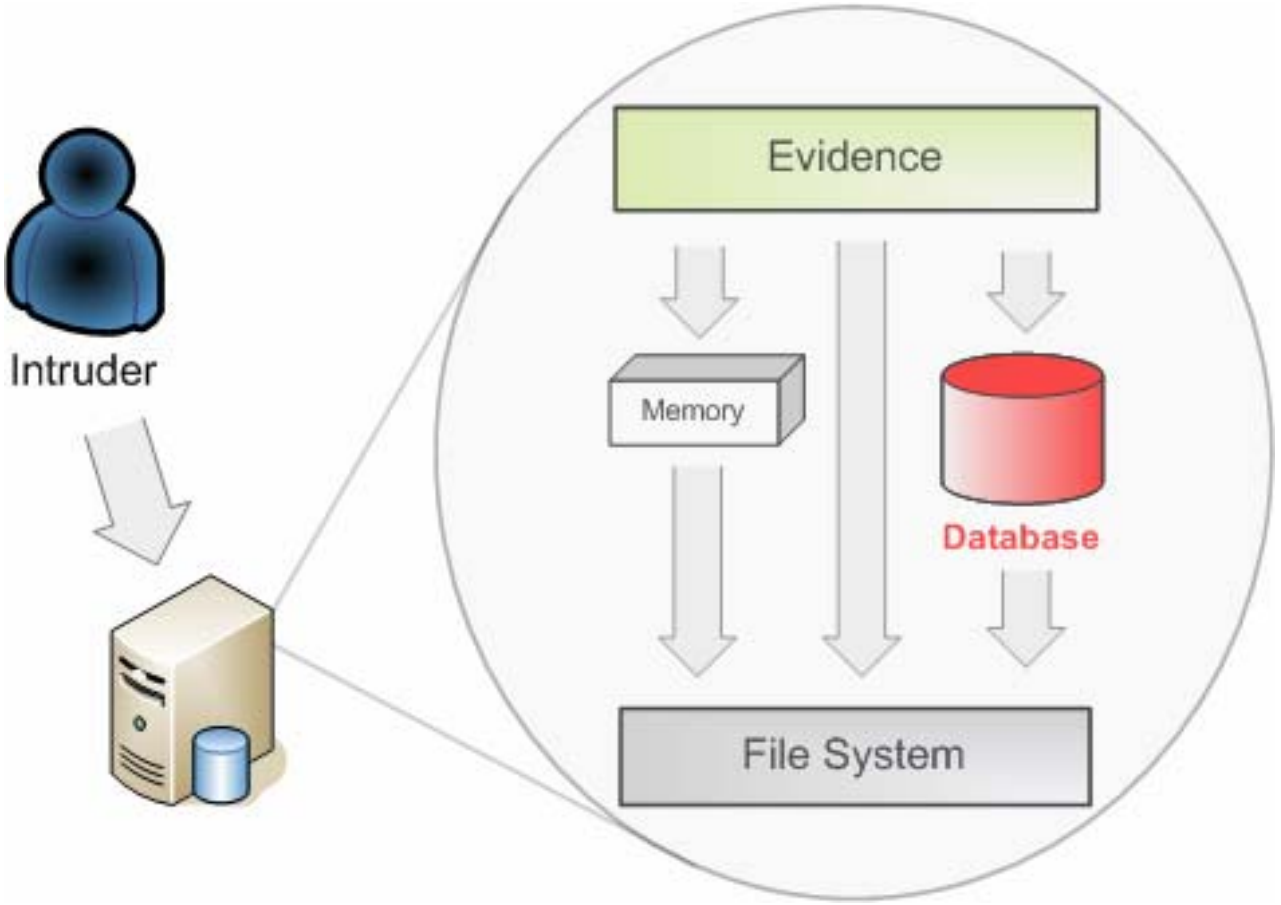
Black Hat USA 2007

SQL Server Forensics | Why are Databases Critical Assets?

- Why are databases critical assets?
  - Databases hold critical information
  - Industry trends are scaling in versus out
  - Database servers today hold more sensitive information than ever before
  - Data security legislations & regulations dictate that security breaches must be reported
  - Database security breaches are “Front Page” news
    - T.J. Maxx | 45.7 million credit/debit cards disclosed
    - CardSystems Solutions | 200,000 credit/debit cards disclosed

Emergis | assure SECURITY

- Traditional investigations often exclude databases



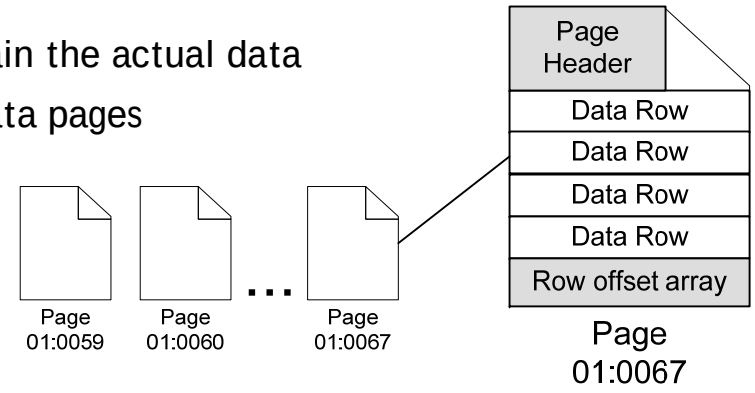
- Database Forensics

  The application of computer investigation and analysis techniques to gather database evidence suitable for presentation in a court of law

  Benefits

  - Retrace user DML & DDL operations
  - Identify data pre and post transaction
  - Recover previously deleted data rows
  - Can help prove/disprove a data security breach
  - Can help determine the scope of a database intrusion
  - For the "real world": No dependency on 3rd party auditing tools or pre-configured DML or DDL triggers

- Database files
  - Data files (.mdf) contain the actual data
  - Consists of multiple data pages



  - Data rows can be fixed or variable length
  - Log files (.ldf) hold all data required to reverse transactions and recover the database
  - Physical log files consist of multiple Virtual Log Files (VLF)

VLF #1 (Inactive)	VLF #2 (Inactive)	VLF #3 (Active)	VLF #4 (Inactive)	Free Space

  - A VLF is the unit of truncation for the transaction log
  - According to Microsoft:

> **"Although you might assume that reading the transaction log directly would be interesting or even useful, it's just too much information."**
>
> Inside SQL Server 2005: The Storage Engine, Microsoft Press, 2006

Emergis | assure SECURITY

SQL Server Forensics | Database Forensics Primer(2)

Inside the transaction log:

1. CurrentLSN
2. **Operation**
3. Context
4. **Transaction ID**
5. Tag Bits
6. Log Record Fixed Length
7. Log Record Length
8. PreviousLSN
9. Flag Bits
10. AllocUnitID
11. AllocUnitName
12. **Page ID**
13. **Slot ID**
14. Previous Page LSN
15. PartionID
16. RowFlags
17. Num Elements
18. **Offset in Row**
19. Checkpoint Begin
20. CHKPT Begin DB Version
21. MaxXDESID
22. Num Transactions
23. Checkpoint End
24. CHKPT End DB Version
25. Minimum LSN
26. Dirty Pages
27. Oldest Replicated Begin LSN
28. Next Replicated End LSN
29. Last Distributed End LSN
30. Server UID
31. UID
32. **SPID**
33. BeginLogStatus
34. Begin Time
35. Transaction Name
36. Transaction SID
37. End Time
38. Transaction Begin
39. Replicated Records
40. Oldest Active LSN
41. Server Name
42. Database Name
43. Mark Name
44. Master XDESID
45. Master DBID
46. PrepLogBegin LSN
47. PrepareTime
48. Virtual Clock
49. Previous Savepoint
50. Savepoint Name
51. Rowbits First Bit
52. Rowbits Bit Count
53. Rowbits Bit Value
54. Number of Locks
55. Lock Information
56. LSN Before Wrties
57. Pages Written
58. Data Pages Delta
59. Reserved Pages Delta
60. Used Pages Delta
61. Data Rows Delta
62. Command Type
63. Publication ID
64. Article ID
65. Partial Status
66. Command
67. Byte Offset
68. New Value
69. Old Value
70. New Split Page
71. Rows Deleted
72. Bytes Freed
73. CI Table ID
74. CI Index ID
75. NewAllocationUnitID
76. FIlegroupID
77. Meta Status
78. File Status
79. File ID
80. Physical Name
81. Logical Name
82. Format LSN
83. RowsetID
84. TextPtr
85. Column Offset
86. Flags
87. Text Size
88. Offset
89. Old Size
90. New Size
91. Description
92. Bulk allocated extent count
93. Bulk rowinsertID
94. Bulk allocationunitID
95. Bulk allocation first IAM Page ID
96. Bulk allocated extent ids
97. **RowLog Contents 0**
98. **RowLog Contents 1**
99. RowLog Contents 2
100. RowLog Contents 3
101. RowLog Contents 4

Emergis | assure SECURITY

- Server Process ID (SPID)
  - A unique value used by SQL Server to track a given session within the database server
  - Transaction log activity is logged against the executing SPID

- Data type storage and retrieval
  - 31 different data types
  - Data types are stored and retrieved differently within SQL Server
  - Storing and retrieving value: **21976** in various data types results in the following:

Data type	On disk value	Retreived value
CHAR	3231393736	21976
INT	D855	21976
DATETIME	D855	3/3/1960

  - Big endian ordering (BEO) is applicable to number formats

Procedure Cache

- Ah-hoc statement and procedure execution plans

- SQL Server data resides natively within SQL Server and stored externally within the native Windows operating system

- Evidence repositories
  - **SQL Server**
    - Volatile database data
    - Database data files
    - Database log files
    - Plan cache
    - Data cache
    - Indexes
    - Tempdb
    - Version store
  - **Operating System**
    - Trace files
    - System event logs
    - SQL Server error logs
    - Page file
    - NTFS journal
    - Memory

- SQL Server Management Studio Express
- SQLCMD
- Windows Forensic Toolchest
- DD\DCFLDD
- MD5SUM
- Netcat\CryptCat
- WinHex
- Native SQL Server views, functions and statements
  - Dynamic Management Views (DMV)
  - Database Consistency Checker (DBCC) commands
  - FN_*
- Lots of sanitized acquisition media

Evidence Collection

SQL Server Forensics | Evidence Collection(2)

- Determine the scope of evidence collection

- Prioritize evidence collection
  1. Volatile database data (sessions/connections, active requests, plan cache, etc.)
  2. Transaction logs
  3. Database files
  4. SQL Server error logs
  5. System event logs
  6. Trace files

Emergis | assure SECURITY

SQL Server Forensics | Evidence Collection(3)

- Collecting volatile database data
  - Can be automated using WFT & command line SQL tools
  - GUI front end, binary validation and thorough logging
  - Gathers volatile data internal and external to SQL Server

Menu

MAIN
ABOUT
LOG
CONFIG

START

START TIME

SQL SERVER

DB LISTING
DM_EXEC_CONNECTIONS
DM_EXEC_SESSIONS
SQL_LOGINS
DM_EXEC_REQUESTS

MEMORY

PCCLIP
MEM

SQL_LOGINS

Command md5=28731C04B854CC1570DBDACC89A6C3F2

SQLCMD.exe -E -Q "select name, type_desc, create_date, modify_date from sys.sql_logins order by create_dat
modify_date" > e:\sql_logins.txt

Description

SQL_LOGINS

File sql_logins.txt md5=42AB71BA778BBDD2F89C7587902705C0

```
name
------------------------------------------------------------------------------------------

sa
EASYACCESS

(2 rows affected)
```

Computer Name: PRODSQL05 **Date/Time: 03/02/2007 10:17:4**

Windows Forensic Toolchest (WFT) v1.0.03 (2003.09.20) Copyright (C) 2003 Monty McDougal. All rights

Emergis assure SECURITY

SQL Server Forensics | Evidence Collection(4)

- SQLCMD
  - Load command line tool and establish logging

```
D:\FResponse>Sqlcmd -S RZ-SQL-2005 -e -s"," -E
1> :out z:\initialconnection.txt
```

- Collecting the active transaction log
  - Determine on disk locations of the transaction log files

```
D:\FResponse>Sqlcmd -S RZ-SQL-2005 -e -s"," -E
1> sp_helpdb OnlineSales
2> go
```

Results:

name	fileid	filename	…
OnlineSales	1	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\OnlineSales.mdf	…
OnlineSales_log	2	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\OnlineSales_log.ldf	…
OnlineSales_log2	3	C:\OtherLogs\OnlineSales_log2.ldf	…

Emergis assure SECURITY

SQL Server Forensics | Evidence Collection(5)

- Collecting the active transaction log (cont'd)
  - Gather the VLF allocations

```
D:\FResponse>Sqlcmd -S RZ-SQL-2005 -e -s"," -E
1> :out z:\initialconnection.txt
```

Results:

FileId	FileSize	StartOffset	FSeqNo	Status	Parity	CreateLSN
2	14352384	8192	16	2	64	0
2	14352384	14360576	0	0	0	0
2	14352384	28712960	0	0	0	0
2	14606336	43065344	0	0	0	0
3	14352384	8192	0	0	0	0
3	14352384	14360576	0	0	0	0
3	14352384	28712960	0	0	0	0
3	14606336	43065344	0	0	0	0

2 = Active

0 = Recoverable or unused

- Collecting the active transaction log (cont'd)
  - Fn_dblog filters transactions by:
    - Target database object
    - Specific columns
    - SPID and/or date/time range

    *Select * from ::fn_dblog(NULL, NULL)*

  - DBCC Log
    - More resource intensive
    - Dumps transaction log in its entirety

    dbcc log(<databasename>, 3)

    0 = minimal info

    1 = slightly more info

    2 = detailed info including (page id, slot id, etc.)

    3 = full information about each operation

    4 = full information on each operation in addition to hex dump of current data row

SQL Server Forensics | Evidence Collection(7)

- Collecting the database plan cache
  - **Collecting the plan cache**

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

  - **Collect additional plan cache specifics**

    *select * from sys.dm_exec_query_stats*

- Collecting database data files & logs (\\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\)
- Collecting default trace files and error logs (\\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\)
- Collecting SQL Server error logs (\\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\)
- Collecting system event log (WFT)

Emergis | assure SECURITY

Evidence Analysis

SQL Server Forensics | Evidence Analysis(2)

- Windows event log
  - SQL Server Windows-based authentication data (failures, successful Log-on/off)
  - Server startup and shutdown
  - IP addresses of SQL Server client connections

- Error log
  - Complete authentication history
  - Server startup and shutdown
  - IP addresses of SQL Server client connections

```
17-03-02 07:39:10.80 Logon      Login failed for user 'sa'. [CLIENT: 192.168.1.20]
17-03-02 07:39:11.00 Logon      Error: 18456, Severity: 14, State: 8.
17-03-02 07:39:11.00 Logon      Login failed for user 'sa'. [CLIENT: 192.168.1.20]
17-03-02 07:39:11.20 Logon      Error: 18456, Severity: 14, State: 8.
17-03-02 07:39:11.20 Logon      Login failed for user 'sa'. [CLIENT: 192.168.1.20]
17-03-02 07:53:07.39 Logon      Login succeeded for user 'sa'. Connection: non-trusted. [CLIENT: 192.168.1.20]
17-03-02 08:09:37.60 Logon      Login succeeded for user 'EASYACCESS'. Connection: non-trusted. [CLIENT: 192.168.1.20]
```

Emergis assure SECURITY

SQL Server Forensics | Evidence Analysis(3)

- Default database trace
  - Complete authentication history
  - DDL Operations (Schema changes)
  - IP addresses of SQL Server client connections

ssID	ApplicationName	LoginName	SPID	StartTime	EventSubClass	Success	LoginSid	RequestID	EventSe
160	squelda 1.0	sa	51	2007-03-02 07:39:11.003		0		0	
160	squelda 1.0	sa	51	2007-03-02 07:39:11.203		0		0	
1300	OSQL-32	sa	51	2007-03-02 07:54:07.180	1 - Add	1	0X01	0	
1300	OSQL-32	sa	51	2007-03-02 07:54:34.030	1 - Commit		0X01	0	
1300	OSQL-32	sa	51	2007-03-02 07:54:35.740			0X01	0	
1300	OSQL-32	sa	51	2007-03-02 07:54:35.903	0 - Begin		0X01	0	
1300	OSQL-32	sa	51	2007-03-02 07:54:35.913	1 - Commit		0X01	0	
1300	OSQL-32	sa	51	2007-03-02 07:55:52.783	3 - Grant ...	1	0X01	0	
1300	OSQL-32	sa	51	2007-03-02 07:56:18.440	1 - Add	1	0X01	0	
1660	OSQL-32	EASYACCESS	51	2007-03-02 08:09:33.773	1 - Commit		0XB89...	0	
			2	2007-03-02 08:13:29.350	1 - Increase			0	
1660	OSQL-32	EASYACCESS	51	2007-03-02 08:13:31.433	1 - Commit		0XB89...	0	
1660	OSQL-32	EASYACCESS	51	2007-03-02 08:13:32.667	1 - Commit		0XB89...	0	

Emergis assure SECURITY

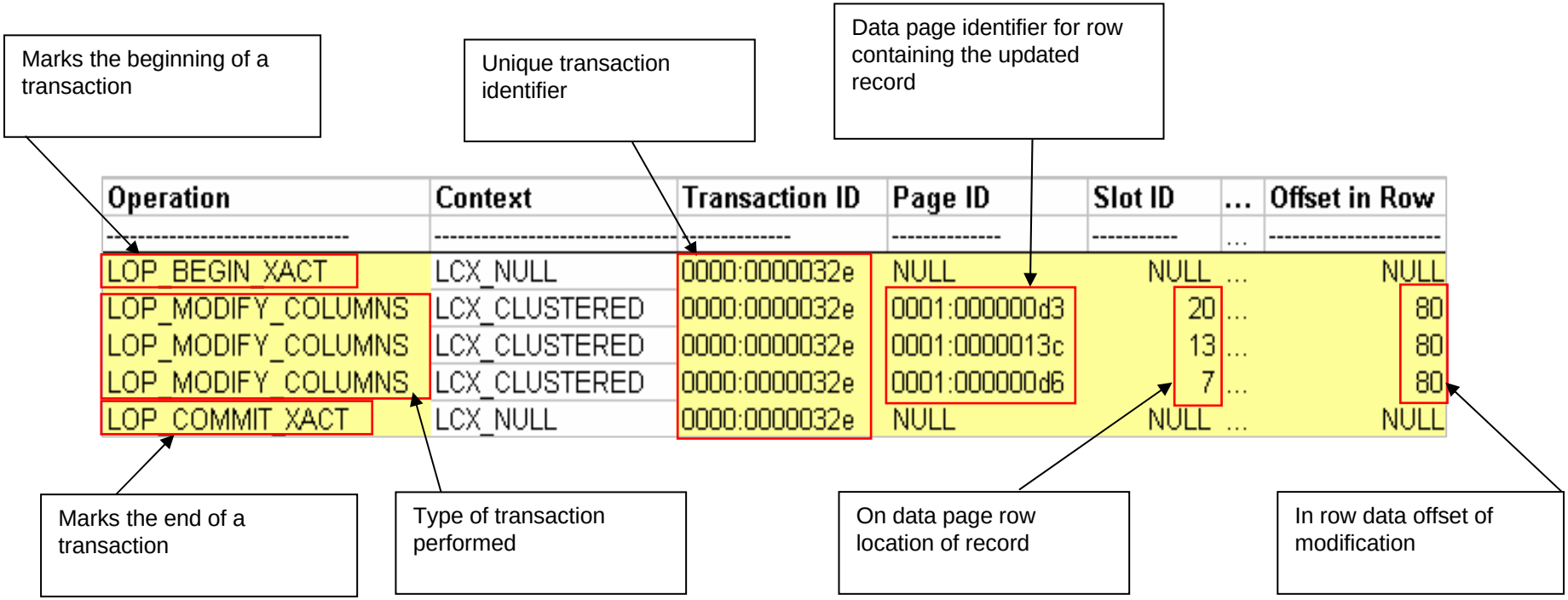
SQL Server Forensics | Evidence Analysis(4)

- Data files & Log files
  - Attach files
  - Use to obtain on-demand schema info, data page contents, etc.

- Active transaction log
  - Import into Excel / Access for viewing
  - Identify DML & DDL statements
  - Map transactions to a SPID

Emergis | assure SECURITY

- Transaction Log – Update operations



Operation	Context	Transaction ID	Page ID	Slot ID	...	Offset in Row
LOP_BEGIN_XACT	LCX_NULL	0000:0000032e	NULL	NULL	...	NULL
LOP_MODIFY_COLUMNS	LCX_CLUSTERED	0000:0000032e	0001:000000d3	20	...	80
LOP_MODIFY_COLUMNS	LCX_CLUSTERED	0000:0000032e	0001:0000013c	13	...	80
LOP_MODIFY_COLUMNS	LCX_CLUSTERED	0000:0000032e	0001:000000d6	7	...	80
LOP_COMMIT_XACT	LCX_NULL	0000:0000032e	NULL	NULL	...	NULL

Identifier	Hex	Decimal
Transaction ID	0000:0000032e	0:814
Data Page	0001:000000d3	1:211

Emergis assure SECURITY

- DBCC Page will pull up the modified data page

  dbcc page (OnlineSales, 1, 211, 1)

- Viewing the page header will detect the owning object

```
Page @0x04304000

m_pageId = (1:211)                    m_headerVersion = 1                   m_type = 1
m_typeFlagBits = 0x0                  m_level = 0                           m_flagBits = 0x0
m_objId (AllocUnitId.idObj) = 87      m_indexId (AllocUnitId.idInd) = 256
Metadata: AllocUnitId = 72057594043629568
Metadata: PartitionId = 72057594039500800                                   Metadata: IndexId = 1
Metadata: ObjectId = 629577281        m_prevPage = (1:314)                  m_nextPage = (1:315)
pminlen = 108                         m_slotCnt = 22                        m_freeCnt = 3263
m_freeData = 5918                     m_reservedCnt = 0                     m_lsn = (16:3686:2)
m_xactReserved = 0                    m_xdesId = (0:0)                      m_ghostRecCnt = 0
m_tornBits = -1731484635
```

- Lookup the owning object: *Select * from sysobjects where id = 629577281*

Results:

	name	id	xtype	uid	info	status	base_schema_ver	replinfo	parent_obj	crdate
1	Order	629577281	U	1	0	0	0	0	0	2007-02-26 16:08:21.320

- Gather the object schema

  "SELECT sc.colorder, sc.name, st.name as 'datatype', sc.length FROM syscolumns sc, systypes st

  WHERE sc.xusertype = st.xusertype and sc.id = 629577281

  ORDER BY colorder"

- Results:

colorder	name	datatype	length
1	OrderID	int	4
2	FirstName	varchar	20
3	LastName	varchar	20
4	Address	varchar	50
5	City	nchar	40
6	State	nchar	4
7	ZIP	nchar	10
8	CCType	varchar	15
9	CCNumber	varchar	20
11	ShipStatusID	int	4
12	OrderDate	datetime	8
13	Product	nvarchar	100
14	Price	nchar	30

Emergis assure SECURITY

- Viewing data page 1:211 modified using Slot 20 & Row offset 80

```
Slot 20 Offset 0x147f Length 237

Record Type = PRIMARY_RECORD          Record Attributes =  NULL_BITMAP VARIABLE_COLUMNS

Memory Dump @0x2F3AD47F

00000000:   30006c00 6f000000 53007000 72006900 †0.l.o...S.p.r.i.
00000010:   6e006700 4c006100 6b006500 20002000 †n.g.L.a.k.e. . .
00000020:   20002000 20002000 20002000 20002000 † . . . . . .
00000030:   41005a00 31003400 34003100 30000a00 †A.Z.1.4.4.1.0...
00000040:   00000100 00000000 0000e498 00003300 †..............3.
00000050:   2e003500 30002000 20002000 20002000 †..5.0. . . . . .
00000060:   20002000 20002000 20002000 0e0000c0 † . . . . . .....
00000070:   06008400 88009900 9d00ad00 ed00416e †..............An
00000080:   6f736f6e 456d696c 37322053 74617266 †osonEmil72 Starf
00000090:   656c6c20 44726976 65566973 61343931 †ell DriveVisa491
000000A0:   36383833 38343033 38323330 3056006f †6883840382300V.o
000000B0:   006c0063 0061006e 006f0020 00360032 †.l.c.a.n.o. .6.2
000000C0:   00200069 006e0063 00680020 0050006c †. .i.n.c.h. .P.l
000000D0:   00610073 006d0061 00200054 00560020 †.a.s.m.a. .T.V.
000000E0:   00560043 00320033 00330032 00†††††††.V.C.2.3.3.2.
```

Transaction

Start of column

- Price column pre and post transaction modification

…	RowLog Contents 0	RowLog Contents 1
…	0x3500300030002E00300030	0x2E00350030002000200020

- Price column pre and post transaction modification

RowLog0

Hex	35	00	30	00	30	00	2E	00	30	00	30
ASCII	5		0		0		.		0		0

RowLog1

Hex	2E	00	35	00	30	00	20	00	20	00	20
ASCII	.		5		0		SP		SP		SP

- 1st record affected by transaction 814 had the price column updated from “3500.00” to “3.50” Including leading byte “33”

- Transaction Log - Insert Operations

Operation	Context	Transaction ID	Page ID	Slot ID	...	Offset in Row
LOP_BEGIN_XACT	LCX_NULL	0000:00000330	NULL	NULL		NULL
LOP_INSERT_ROWS	LCX_CLUSTERED	0000:00000330	0001:00000138	8		NULL
LOP_COMMIT_XACT	LCX_NULL	0000:00000330	NULL	NULL		NULL

- Reconstruct the data row
- RowLog Contents 0:

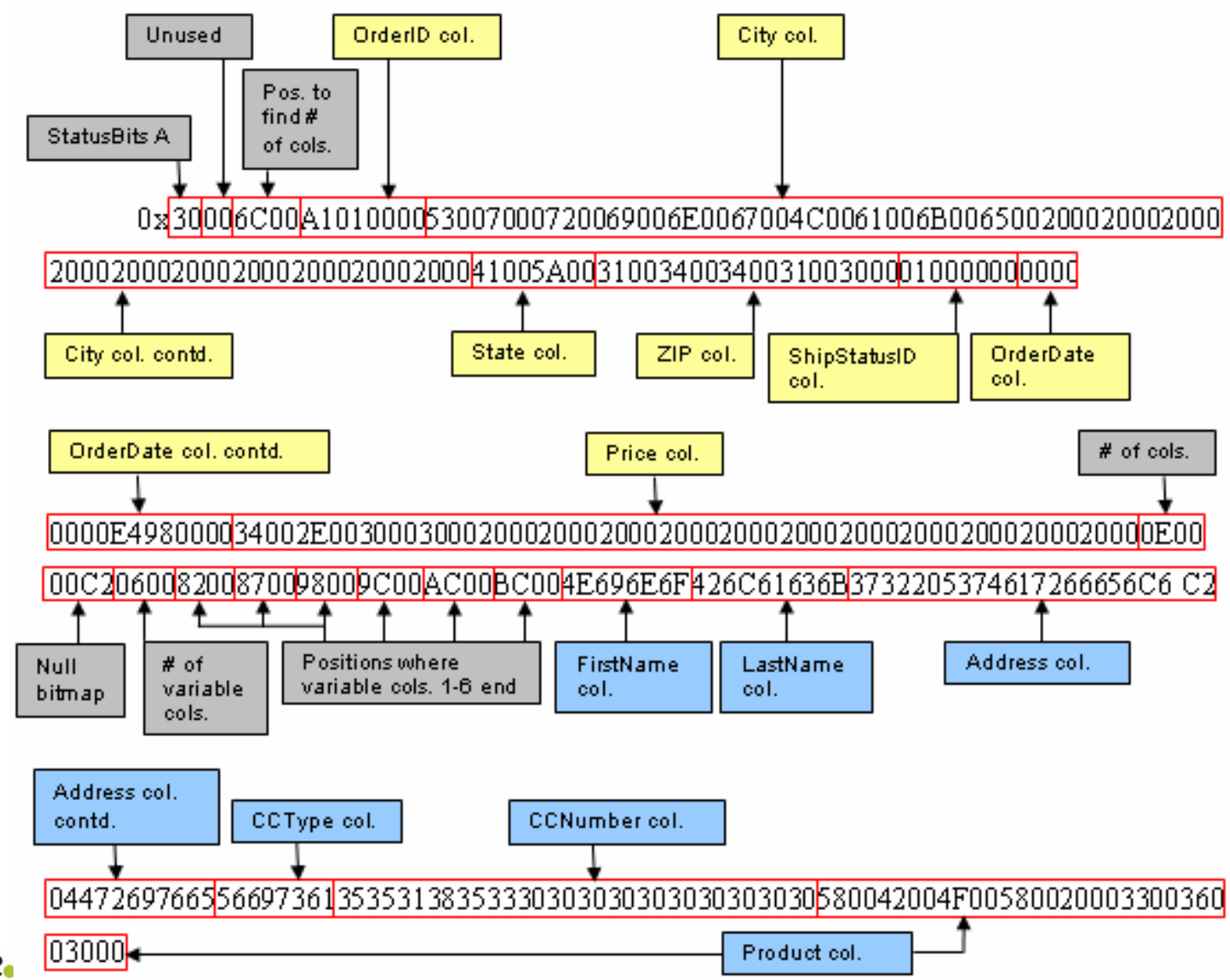
"0x30006C00A10100005300700072006900 6E0067004C0061006B006500200020002000
02000200020002000200020002000200041005A00310034003400310030000100000000000
000E498000034002E00300030002000200020002000200020002000200020002000200020002000
0E0000C20600820087009800 9C00AC00BC004E696E6F426C61636B3732205374617266
656C6C20447269766556697361353531383533303030303030303030303030580042004F00
5800200033003600 3000"

- Lookup the schema and reconstruct the data row
- Structure of a variable length data row:

Source: Inside SQL Server 2005 The Storage Engine

Legend		
Item	**Storage Allocation**	**Description**
1	1 byte	StatusBits A contains data row properties[5]
2	1 byte	Unused in SQL Server 2005[5]
3	2 bytes	Row offset to in row location containing the number of columns in the data row[5]
Fixed length columns	Fixed column length for all fixed columns	Location of in row fixed length data columns[5]
4	2 bytes	Total number of columns in data row[5]
5	1 bit for each row column	Null Bitmap[5]
6	2 bytes	Number of variable length columns within data row[5]
7	2 bytes for each variable length column	Row offset marking the end of each variable length column[5]
Variable length columns	Used length of all variable length columns	Location of in row variable length data columns[5]

SQL Server Forensics | Evidence Analysis(12)



Emergis | assure SECURITY

SQL Server Forensics | Evidence Analysis(13)

- Swap the bytes (endian ordering)
- Translate data types
- The inserted record was:
  - OrderID: 4122
  - FirstName: Nino
  - LastName: Black
  - Address: 72 Starfell Drive
  - City: SpringLake
  - State: AZ
  - ZIP: 14410
  - CCType: Visa
  - CCNumber: 5518530000000000
  - ShipStatusID: 1
  - OrderDate: March 1st, 2007
  - Product: XBOX 360
  - Price: 4.00

Emergis | assure SECURITY

- Transaction Log – Delete operations
- Ghost records

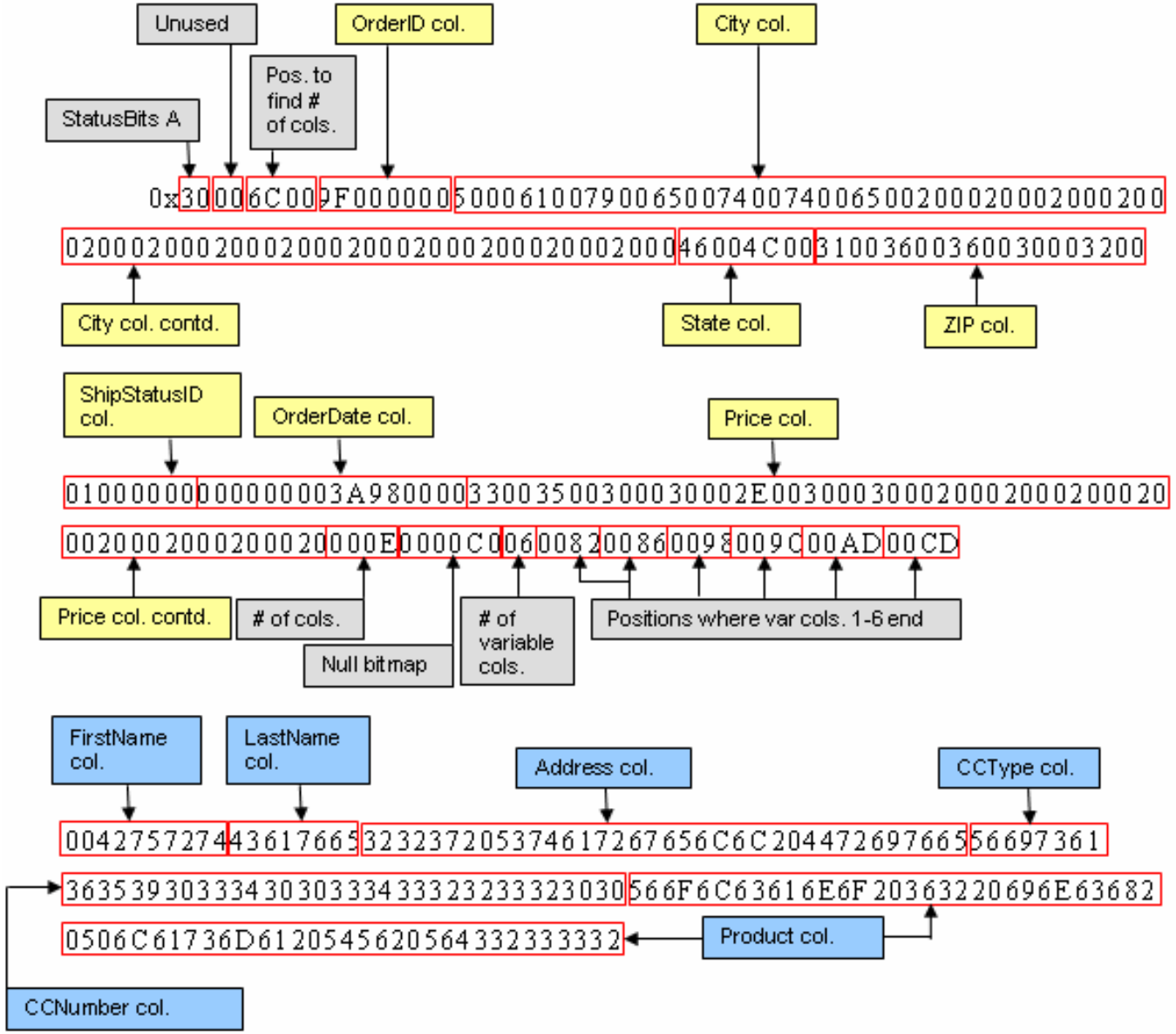
Operation	Context	Transaction ID	Page ID	Slot ID	...	Offset in Row
LOP_BEGIN_XACT	LCX_NULL	0000:00000332	NULL	NULL		
LOP_DELETE_ROWS	LCX_MARK_AS_GHOST	0000:00000332	0001:00000158	24		
LOP_SET_BITS	LCX_PFS	0000:00000000	0001:00000001	0		
LOP_COMMIT_XACT	LCX_NULL	0000:00000332	NULL	NULL		

RowLog Contents 0:

"0x30006C009F0000005000610079006500740074006500200020002000200020002000200020
00200020002000200020002000200046004C00310036003600300032000200000000000000
3A980000330035003000300030002E00300030002000200020002000200020002000200020000E
0000C006008200860098009C00AD00CD00427572744361766532323720537461726765
6C6C20447269766556697361363539303334303033343332323323030566F6C63616E
6F20363220696E636820506C61736D6120545620564332333332"

- Reconstruct the data row

SQL Server Forensics | Evidence Analysis(15)



SQL Server Forensics | Evidence Analysis(16)

- Swap the bytes (endian ordering)
- Translate data types
- The deleted record was:
  - OrderID: 159
  - FirstName: Burt
  - LastName: Cave
  - Address: 227 Stargell Drive
  - City: Payette
  - State: FL
  - ZIP: 16602
  - CCType: Visa
  - CCNumber: 65903400343223200
  - ShipStatusID: 1
  - OrderDate: September 12th, 2006
  - Product: Volcano 62 inch Plasma TV VC2332
  - Price: 3500.00

Emergis | assure SECURITY

- **Plan cache**
  - Review for applicable statements within scope of investigation (date, objects, etc.)
  - Look for **non-standard** statements

bucketid	refcounts	usecounts	cacheobjtype	objtype	plan_ha...	text
6514	2	1	Compiled Plan	Adhoc	0x06000...	SELECT CAST(fti.is_enabled AS bit) AS [IsEnabled], OBJECTPROPERTY(fti.object_id,'TableFullT...
3485	2	1	Compiled Plan	Adhoc	0x06000...	SELECT col.name AS [Name] FROM sys.tables AS tbl INNER JOIN sys.fulltext_indexes AS fti ON ...
942	2	1	Compiled Plan	Adhoc	0x06000...	SELECT dtb.is_fulltext_enabled AS [IsFullTextEnabled] FROM master.sys.databases AS dtb WHE...
6312	2	1	Compiled Plan	Adhoc	0x06000...	select CCNumber, Firstname, Lastname from OrderHistory where OrderID = 1967
551	2	1	Compiled Plan	Adhoc	0x06000...	select * from OrderHistory
8567	2	4	Compiled Plan	Adhoc	0x06000...	select * from sys.dm_exec_cached_plans cross apply sys.dm_exec_sql_text(plan_handle)
944	2	1	Compiled Plan	Adhoc	0x06000...	select CCNumber, Firstname, Lastname from OrderHistory where OrderID = 1
9441	2	1	Compiled Plan	Adhoc	0x06000...	select CCNumber, Firstname, Lastname from OrderHistory where OrderID = 5
8278	2	1	Compiled Plan	Adhoc	0x06000...	select CCNumber, Firstname, Lastname from OrderHistory where OrderID = 22
6913	2	1	Compiled Plan	Adhoc	0x06000...	select CCNumber, Firstname, Lastname from OrderHistory where OrderID = 1823
3005	2	1	Compiled Plan	Adhoc	0x06000...	select CCNumber, Firstname, Lastname from OrderHistory where OrderID = 1639

Investigation Pitfalls

SQL Server Forensics | Investigation Pitfalls(2)

- What to look out for!
  - Know the schema your working with
  - Data type storage formats
  - Reduce large data sets
  - Correlate on-disk values with transaction log data
  - Encryption
  - This takes time so be patient!

Emergis | assure SECURITY

Conclusion

SQL Server Forensics | Conclusion(2)

- Conclusion
  - Don't ignore the database when conducting computer forensics investigations
  - Database forensics techniques learned today can augment traditional forensics skills to uncover the evidence needed to support your case

- Additional information within the presentation white paper
  - Real world database forensics scenario
  - Database forensics methodology
  - Additional evidence collection and analysis techniques
    - Code pages and collation settings
    - Obtaining server configuration
    - Identifying user account creation and elevation of privilege activity
    - Transaction log data carving
    - And more…

Emergis | assure SECURITY

Questions

- Presentation References
  - Kalen Delaney, Inside SQL Server 2005 The Storage Engine, Microsoft Press, 2007
  - Mark Horninger, How to Cheat at Securing SQL Server 2005, Syngress Publishing, 2007
  - "MSDN Blog Pages" http://blogs.msdn.com/sqlserverstorageengine/default.aspx
  - SQL Server 2005 Books Online, http://msdn2.microsoft.com/enus/library/ms130214.aspx