SQL Server Forensics 2

Presented at AppSec Asia

November 18, 2009

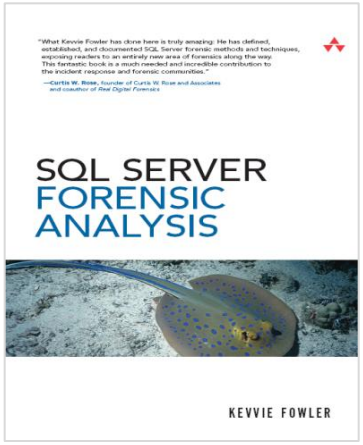
Kevvie Fowler, GCFA Gold, CISSP,
MCTS, MCDBA, MCSD, MCSE

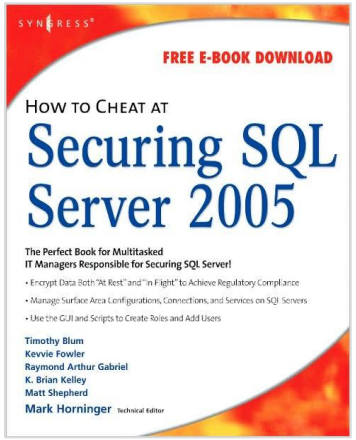
ringzero

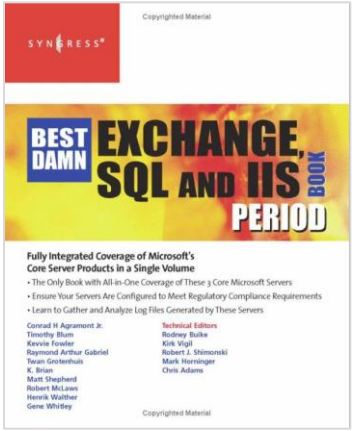
About me

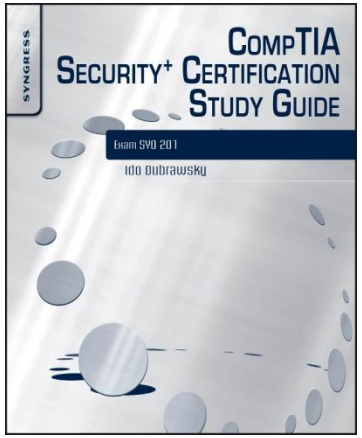
- Day job: Director Security Services, TELUS | backed by Emergis
- Night job: Security researcher, Ringzero
- Security industry contributions:

ApplicationForensics.com







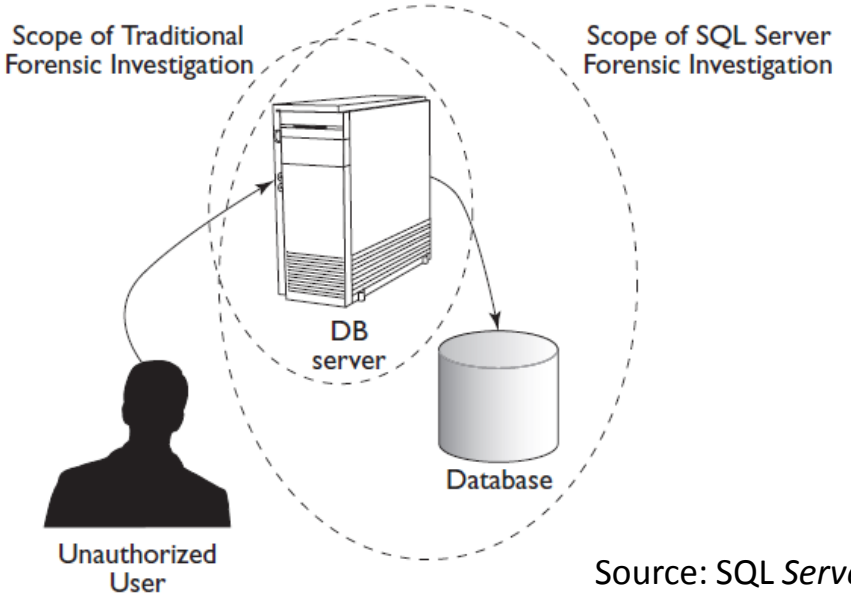


What is database forensics?

- Database forensics:
  - The application of computer investigation and analysis techniques to gather database evidence suitable for presentation in a court of law
    - Retrace user DML & DDL operations
    - Identify data pre and post transaction
    - For the “real world”: No dependency on 3rd party auditing tools or pre-configured DML or DDL triggers

What is database forensics? (2)

- Why do we need it?
  - Typical investigations have little focus on the database



Source: SQL *Server Forensic Analysis*, Kevvie Fowler

- The right approach taken by a major AV vendor this year

What is database forensics? (3)

- Aren't databases stored in the file system therefore part of a typical investigation?
  - 33 different data types many with differences in the way they are stored and retrieved (example value: 21976)

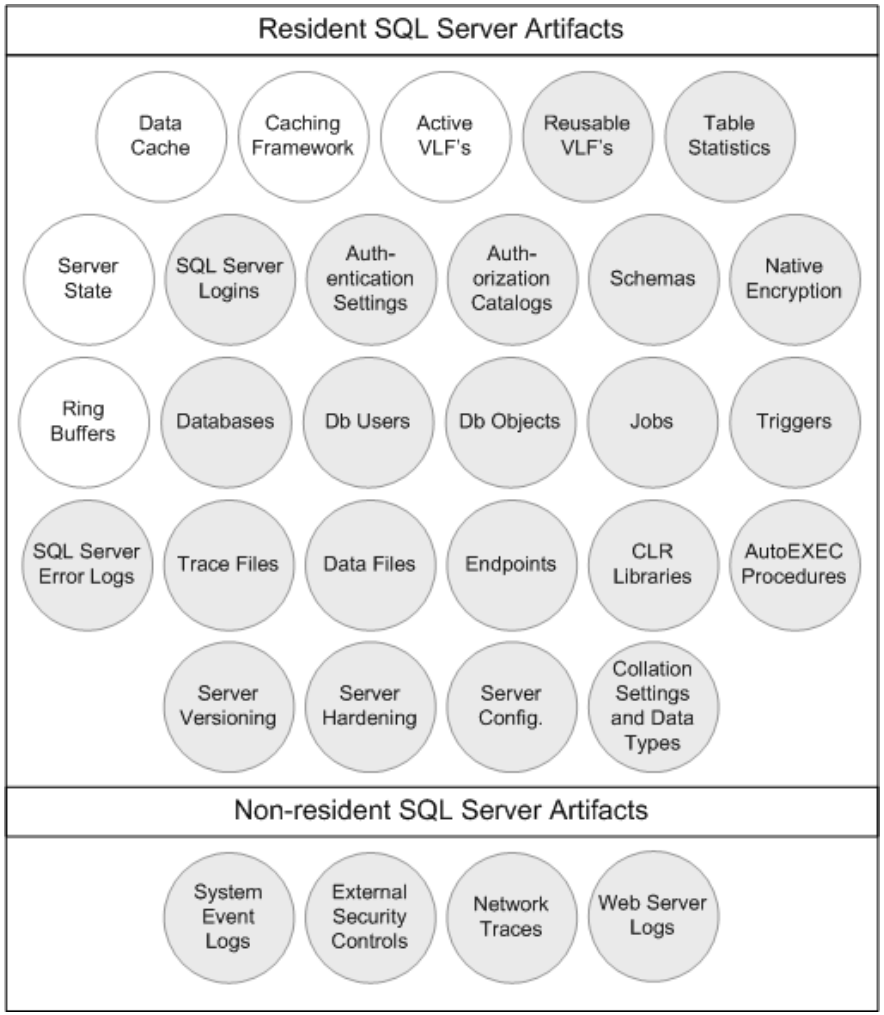
Data type	On disk value	Retreived value
CHAR	3231393736	21976
INT	D855	21976
DATETIME	D855	3/3/1960

  - Random data without understanding the context is just random data:

0x300004000100FE01004B0049006E00740065007200660061006300650003A00200031003900320020002E00310036003800320020002E00310020002E00310037003600200020002D002D002D002D00200030007800380030

Database artifacts

- 33 defined artifacts
  - Volatile
  - Non-volatile
- Artifacts contain traces of a past database attack

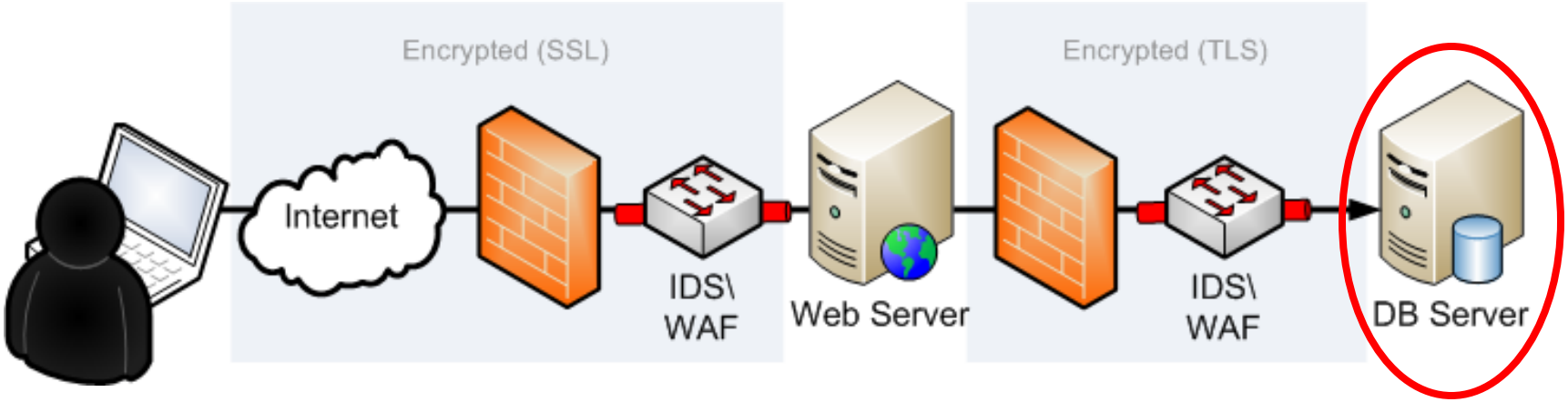


Source: SQL *Server Forensic Analysis*, Kevvie Fowler

ringzero

A typical database attack

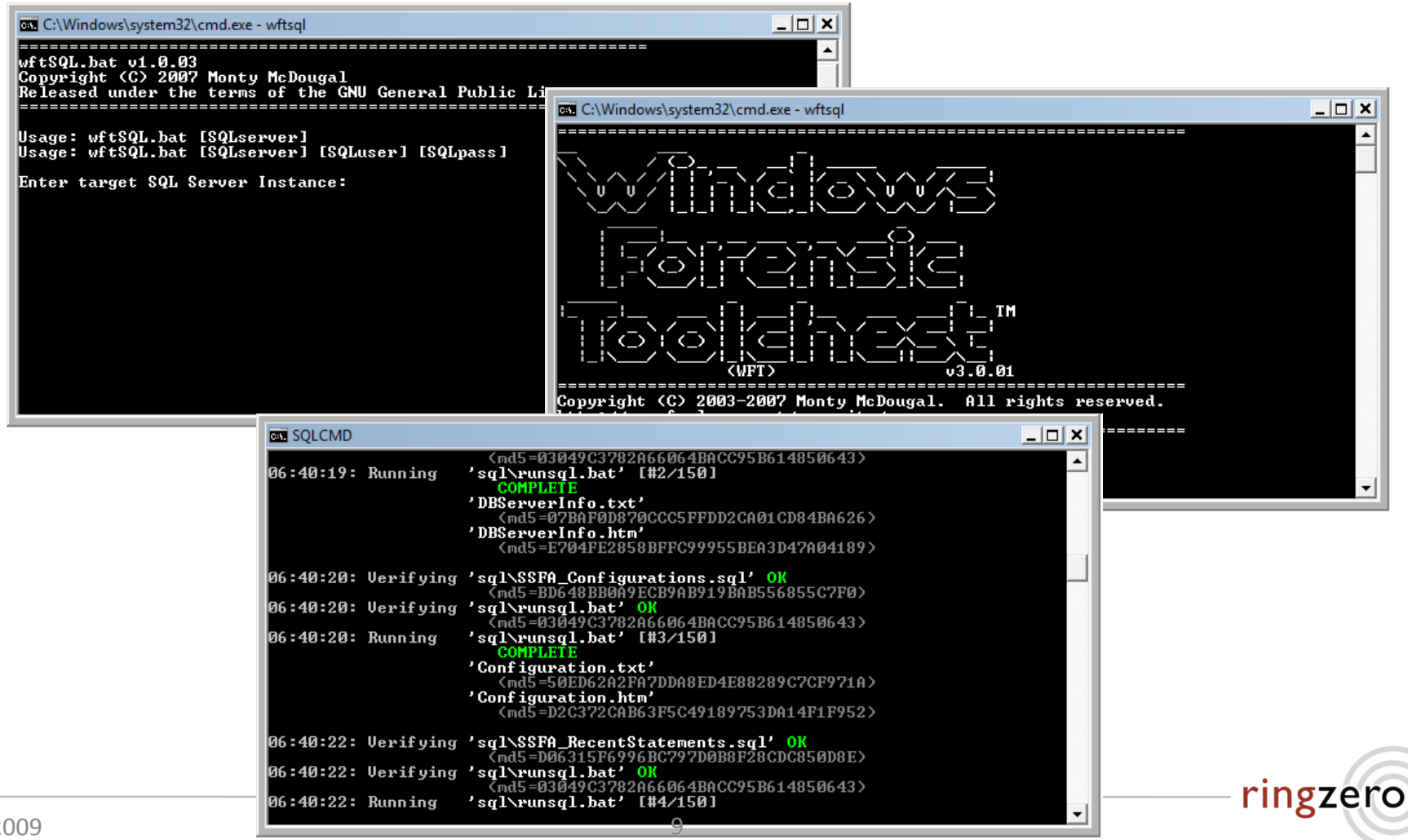
- Credit card data stolen via a SQLi attack
  - Pangolin exploitation video



Incident Verification

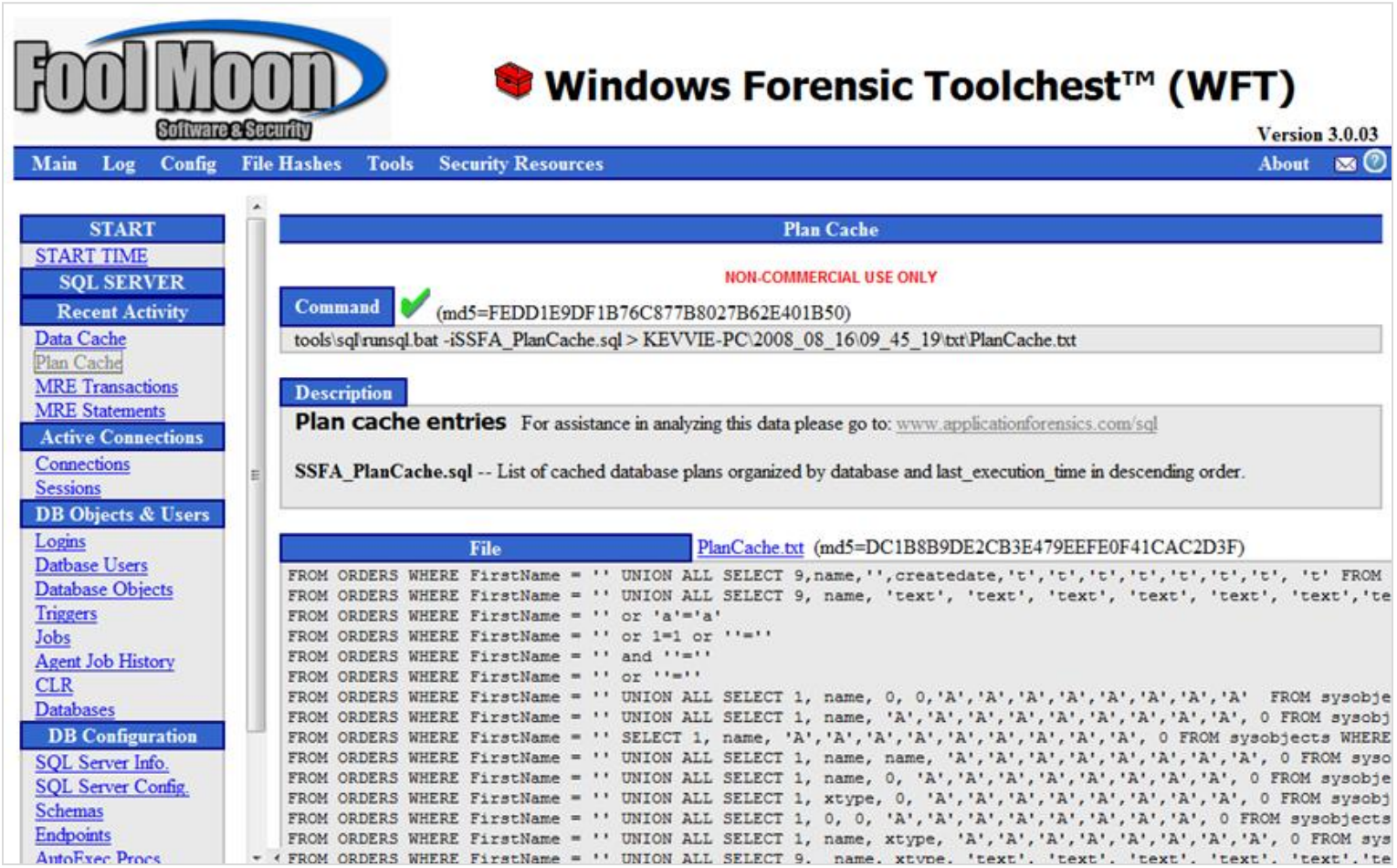
Verifying if an attack has occurred

- WFTSQL (wftsql <instance> <credentials>)



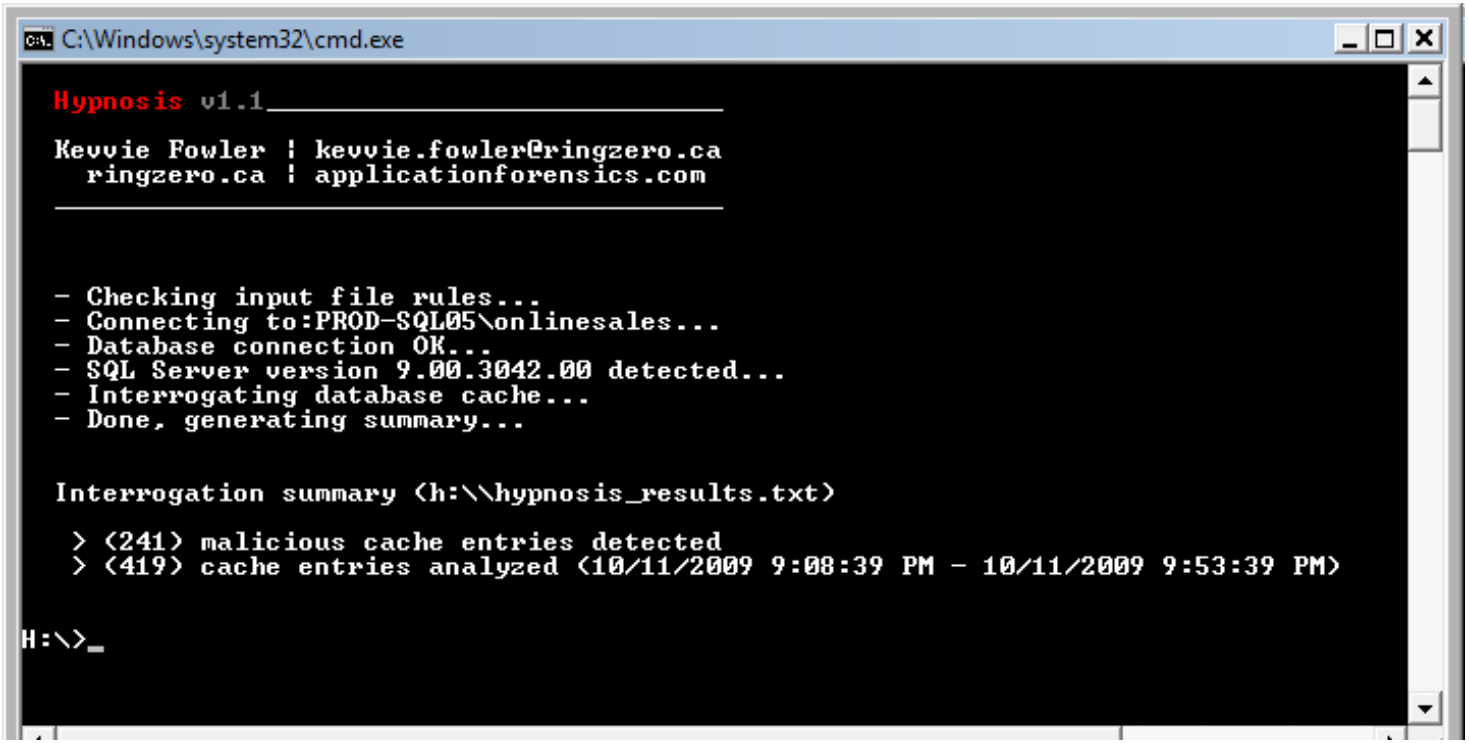
Verifying if an attack has occurred (2)

- WFTSQL results



Verifying if an attack has occurred (3)

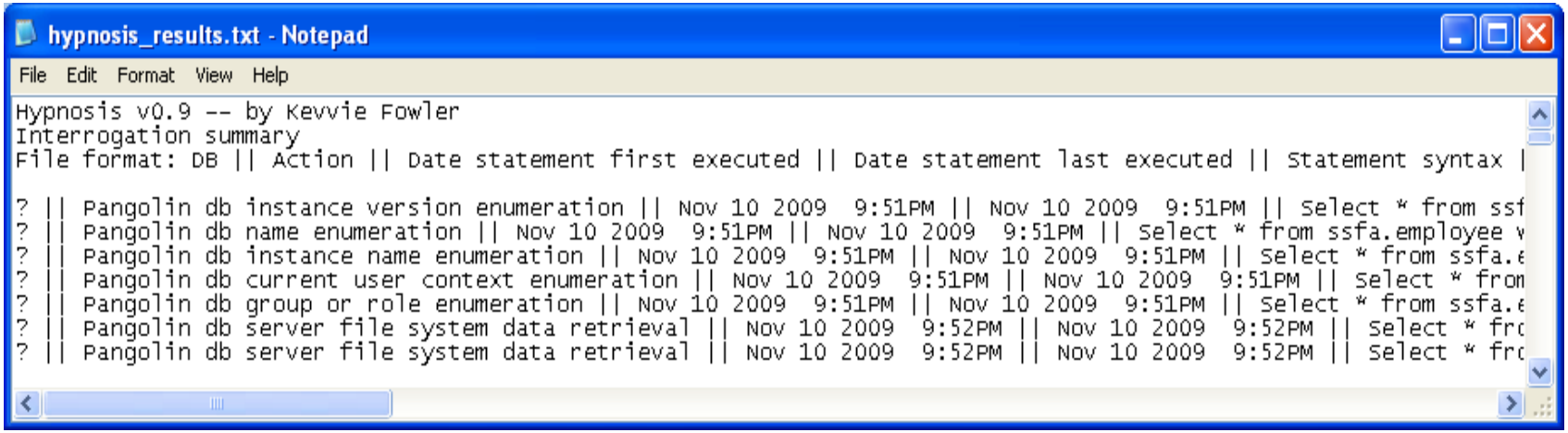
- Hypnosis



Verifying if an attack has occurred (4)

- Hypnosis results file (sample entry)

? || Pangolin db instance version enumeration || Nov 10 2009 9:51PM || Nov 10 2009 9:51PM || Select * from ssfa.employee where fname='Mikaela' and 1=2 union all select null,char(94)+char(94)+char(94)+cast(@@version as nvarchar(4000))+char(94)+char(94)+char(94),null,null,null --' || 1



Artifact Collection

SQL Server Artifact collection

- WFTSQL has already preserved 18 artifacts
- Collect other applicable artifacts

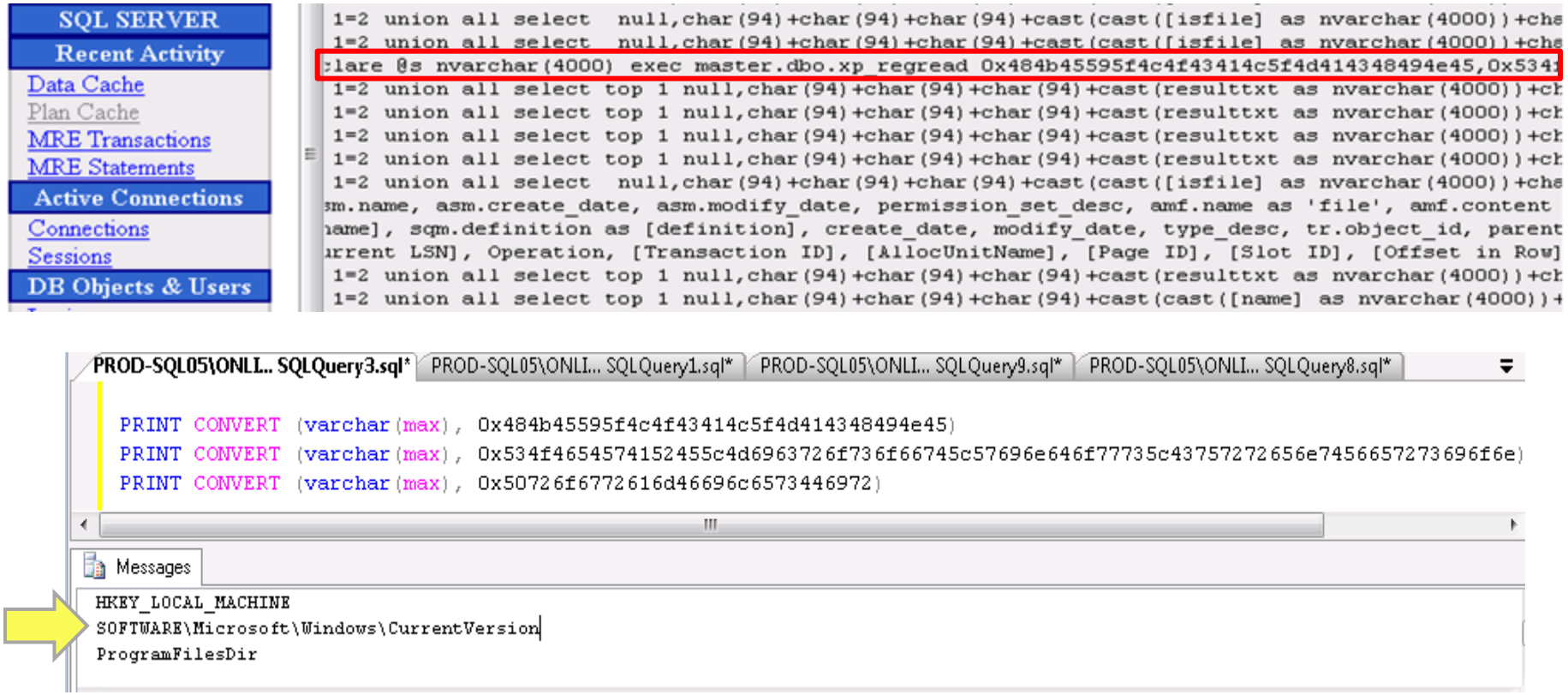
Source: *Server Forensic Analysis*, Kevvie Fowler

ringzero

Artifact Analysis

Analyzing collected artifacts

- Plan Cache
  - Sample cache entry within WFTSQL

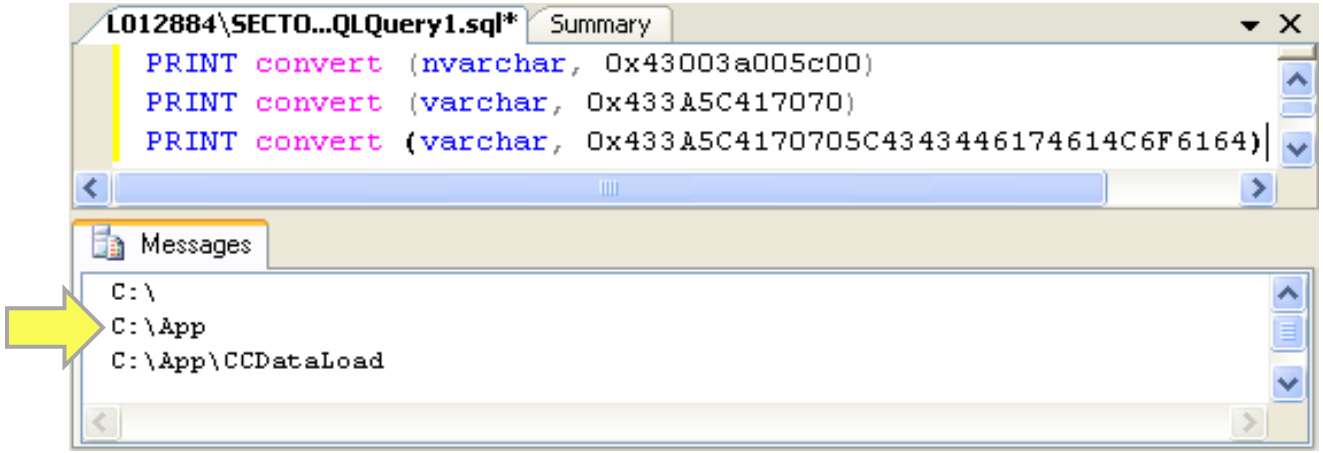


ringzero

Analyzing collected artifacts (2)

- Plan Cache (continued)
  - Sample finding from Hypnosis:

? || **Pangolin db server file system data retrieval** || Oct 3 2009 9:23 PM|| Oct 3 2009 9:23PM || Select * from ssfa.employee where status = 1 and fname='Mikaela' ;declare @z nvarchar(4000) set @z=0x43003a005c00 insert pangolin_test_table execute master..xp_dirtree @z,1,1--' || 1



Analyzing collected artifacts (3)

- Transaction log
  - DML statements
  - DDL statements
  - Login failures…
- Walk through of reconstructing an INSERT operation

Analyzing collected artifacts (4)

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

Analyzing collected artifacts (5)

- INSERTED data row (Rowlog contents 0)

  0x300004000100FE01004B0049006E00740065007200660061006300650003A00200031003900320002E00310036003800 2E0031002E00310037003600020002D002D002D00200030007800 3800

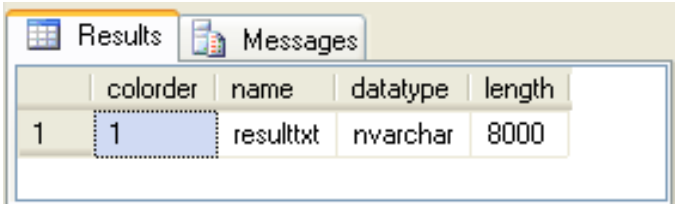
- “Before you can analyze data you need to first understand the format in which it was written”

  Kevvie Fowler quoting himself…

Analyzing collected artifacts (6)

- Understanding the data row structure
  - Syscolumns (existing object)

    SELECT sc.colorder, sc.name, st.name as 'datatype', sc.length FROM syscolumns sc, systypes st WHERE sc.xusertype = st.xusertype and sc.id = Object_ID('Object_Name') ORDER BY colorder



  - Plan cache (existing or deleted object)

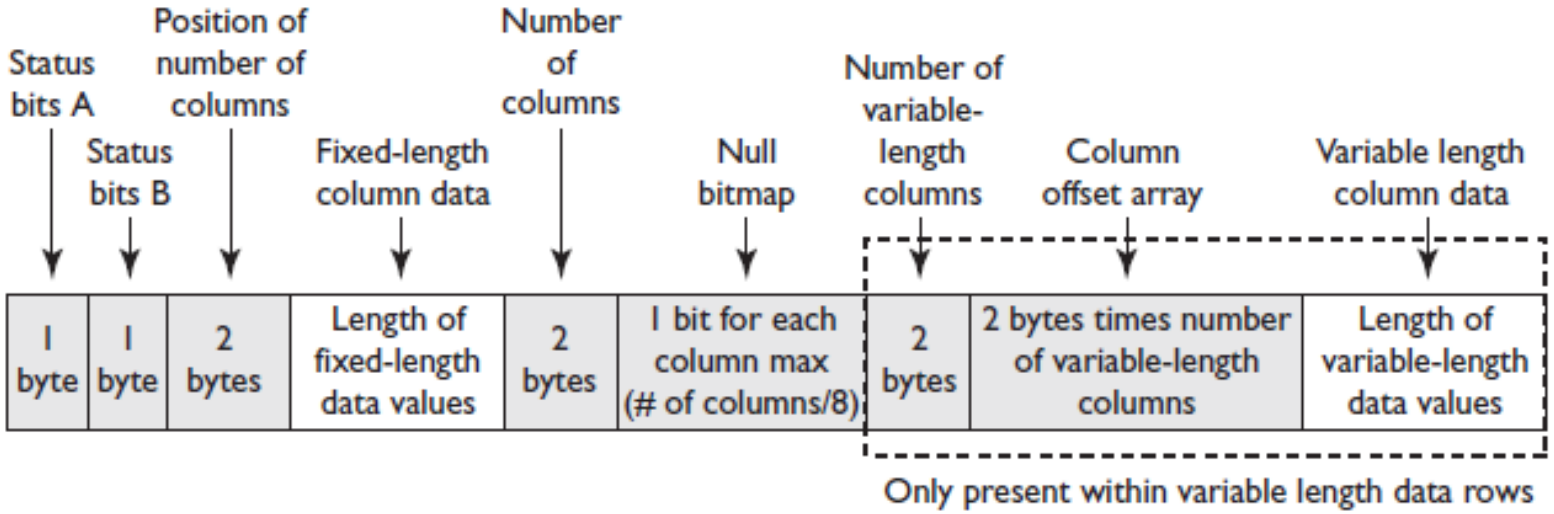
    Select * from ssfa.employee where fname='Mikaela' ;create table [pangolin_test_table]([resulttxt] nvarchar(8000) null);--'

  - Transaction log (existing or deleted object)

    0x30002D0064EFCA5D0000**01**000000**E7**E7000000401F000008D0000002000000401F00000000000000000000010000080010047007200650073007500**6C**007400740078007400

Analyzing collected artifacts (7)

- Understanding the data row structure (continued)

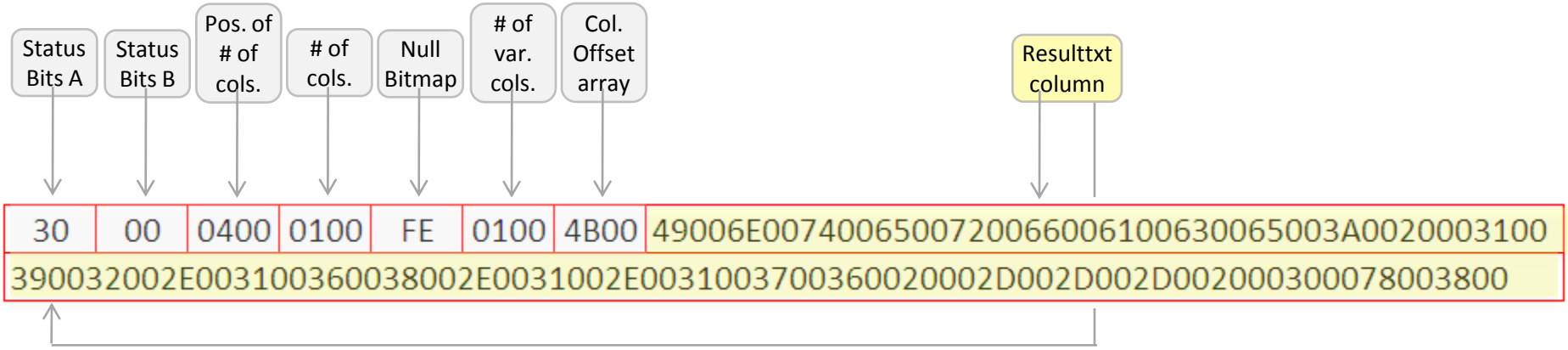


Source: *Server Forensic Analysis*, Kevvie Fowler

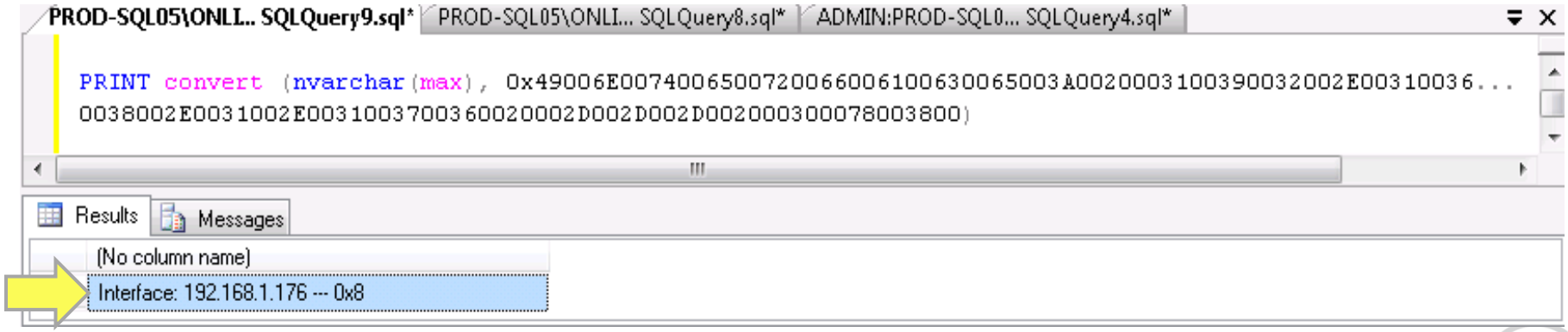
Remember the row structure?

Analyzing collected artifacts (8)

- Applying the structure to our data row



- Reconstructing the insert operation



Analyzing collected artifacts (9)

- 5 sample insert operations retrieved

1) 0x300004000100FE01004B0049006E00740065007200660061006300650003A002000310039003200 2E00310036003800 2E0031002E00310037003600 20002D002D002D00200030007800 3800
2) 0x300004000100FE01006F00200020004900 6E00740065007200 6E0065007400200041006400640072006500730073002000200020002000200020005000680079007300690063006100 6C00200041006400640072006500730073002000200020002000200020005400790070006500
3) 0x300004000100FE01007B00200020003100390032002E00310036003800 2E0031002E00320020002000200020002000200020002000200020002000200020003000300 02D00300030002D00320031002D00640030002D00620034002D00330037002000200020002000200064007900 6E0061006D00690063002000200020002000
4) 0x300004000100FE01007B00200020003100390032002E00310036003800 2E0031002E00330020002000200020002000200020002000200020002000200020003000300 02D00360030002D00620030002D00650065002D00320034002D00630065002000200020002000200064007900 6E0061006D00690063002000200020002000
5) 0x300004000100FE01007B00200020003100390032002E00310036003800 2E0031002E00320030002000200020002000200020002000200020002000200020003000300 02D00360030002D00620030002D00620034002D00640035002D00350037002000200020002000200064007900 6E0061006D00690063002000200020002000

Analyzing collected artifacts (10)

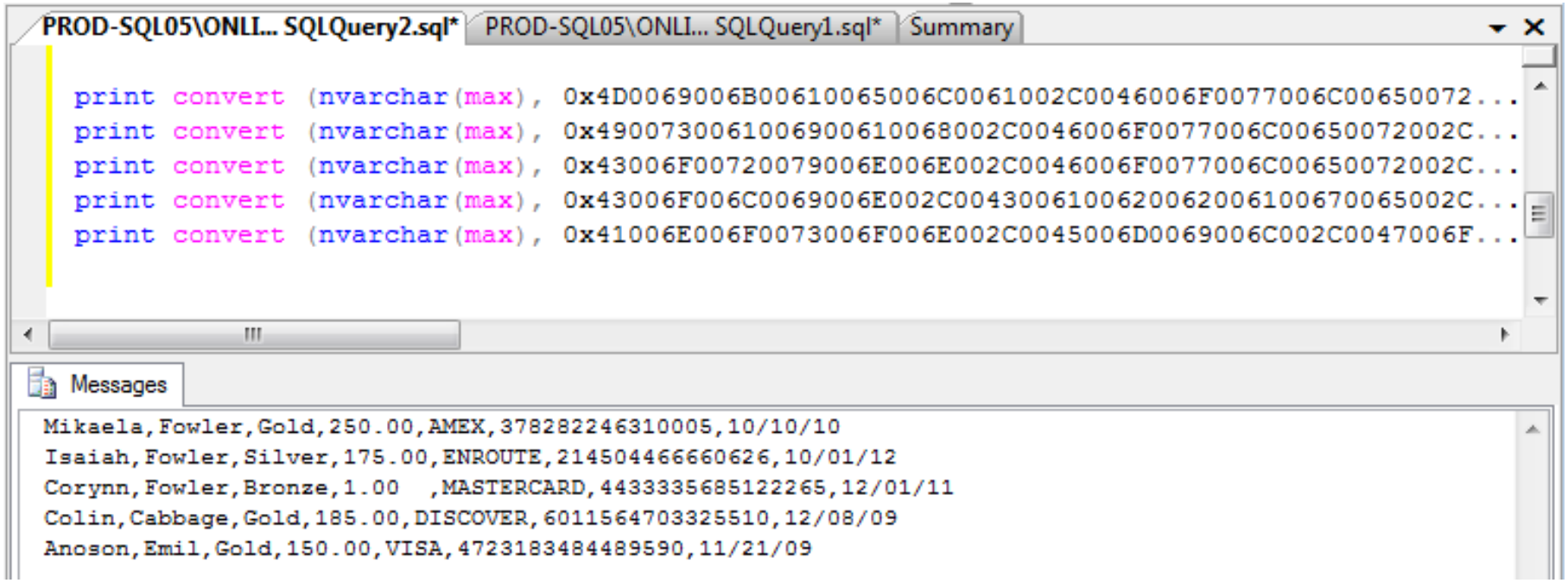
- Reconstruction of data rows inserted into the table and extracted line-by-line by the attacker:

```
(1) Interface: 192.168.1.176 --- 0x8
(2) Internet Address   Physical Address       Type
(3) 192.168.1.2        00-00-21-d0-b4-37      dynamic
(4) 192.168.1.3        00-60-b0-ee-24-ce      dynamic
(5) 192.168.1.20       00-60-b0-b4-d5-57      dynamic
...
```

Can you tell what command was executed during the attack ?

Analyzing collected artifacts (10)

- Using the same process allows us to reconstruct the credit card data viewed by the attacker:



Antiforensics

The Anti-antiforensics

- SQL Server antiforensics research released at Black Hat DC, 2009
- Traditional forensic practices can overcome some of the published antiforensics techniques
- Top 3 points for discussion
  1. Fill transaction logs with CHECKPOINT entries to overwrite DML\DDL data
     - Active transactions and replication is not factored in

The Anti-antiforensics (2)

2. Using sp_cycle_errorlog to overwrite SQL Server errorlog
   - SQL Server doesn't securely delete overwritten error logs which can be easily recovered
   - Number of historical error logs are user configurable

3. Filling default trace files with 20MB of data to force trace rotation
   - SQL Server doesn't securely delete overwritten traces which can be easily recovered

Conclusion

Conclusion

- Don’t ignore the database when conducting computer forensics investigations
- Database forensics techniques learned today can augment traditional forensics skills to uncover the additional evidence needed to support your case
- Additional info on SQL Server Forensics can be found on www.applicationforensics.com
  - SQL Server Forensics (1.0)
  - Double Trouble - Database rootkits and native encryption flaws
  - To Cache a Thief – Cache based SQL injection attack detection
  - Database incident response tools WFTSQL and Hypnosis

Thank-you | additional information

Thank-you! - additional information and questions:

kevvie.fowler@ringzero.ca