

SQL Server Anti-Forensics

proneer

proneer@gmail.com

<http://forensic-proof.com>

Security is a people problem...





1. SQL Server 포렌식
2. SQL Server 안티포렌식
3. 대응 방안



- **대상**

- SQL Server 2005 SP2

- **참고**

- Black Hat DC 2009, SQL Server Anti-Forensics

- **살펴볼 내용**

- 에러 로그 (SQL Server Error Log)
- 윈도우 응용프로그램 로그 (Windows Application Log)
- 기본 흔적 로그 (Default Trace)
- 트랜잭션 로그 (Transaction Log)
- 데이터 파일 (Data File)
- 메모리 (Memory) : 데이터 캐시, 프로시저 캐시

SQL Server Forensics

- SQL Server Error Log
- Windows Application Log
- SQL Server Default Trace
- SQL Server Transaction Log
- SQL Server Data File
- SQL Server Memory



에러 로그

- SQL Server의 장애 복구를 위한 로그
- 항상 활성화되어 있으며 비활성화 할 수 없음
- 에러 로그에 기록되는 로그는 윈도우 응용프로그램 로그에도 기록됨



에러 로그

로그 파일 경로 및 특징

- 경로 : C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\
- 기본 6개의 로그 파일 유지

✓ ERRORLOG (현재 로그)

✓ ERRORLOG.1

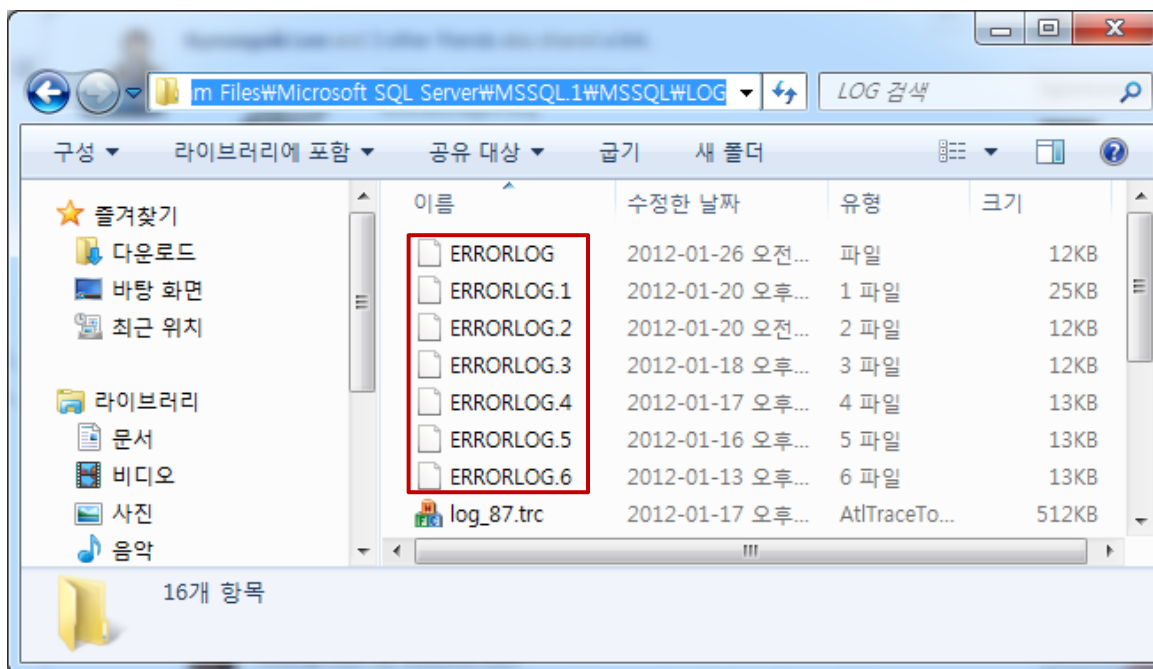
✓ ERRORLOG.2

✓ ERRORLOG.3

✓ ERRORLOG.4

✓ ERRORLOG.5

✓ ERRORLOG.6

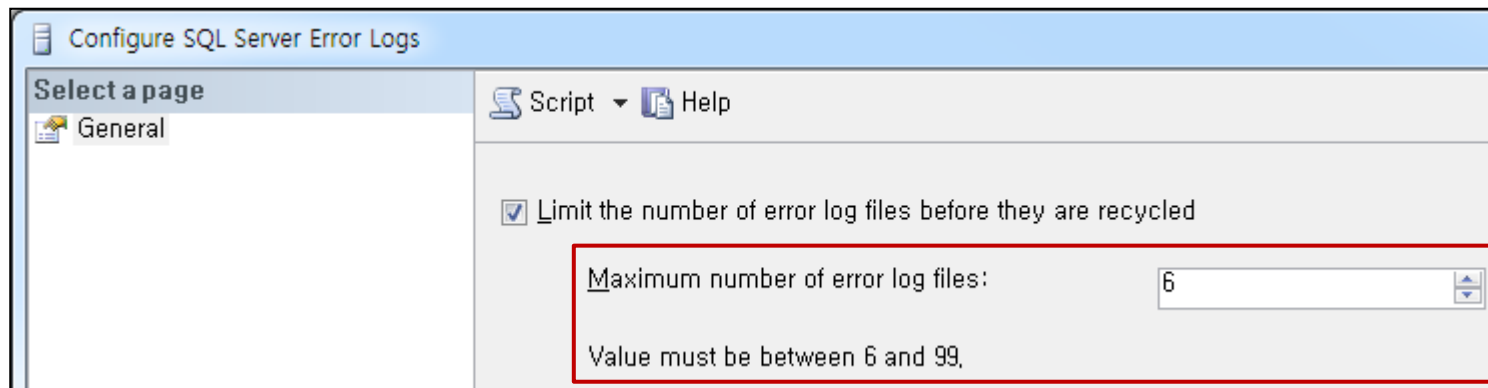




에러 로그

- 로그 파일 수 변경

- Management Studio → Management → SQL Error Logs → Configure



- 레지스트리 변경

```
USE [master]
GO
EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
N'Software\Microsoft\MSSQLServer\MSSQLServer', N'NumErrorLogs', REG_DWORD, 50
GO
```

✓ 기본 설정일 경우에는 레지스트리 값이 존재하지 않음, 로그 파일 수 변경 시 생성



에러 로그

- 로그 파일 생성

- SQL Server 인스턴스가 새로 시작될 때(SQL Server 재시작)마다 새로운 파일 생성

- ✓ `net start | stop MSSQL$(Instance Name)`

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\proneer>net stop MSSQL$INSIGHT
SQL Server (INSIGHT) 서비스를 멈춥니다..
SQL Server (INSIGHT) 서비스를 잘 멈추었습니다.

C:\Documents and Settings\proneer>net start MSSQL$INSIGHT
SQL Server (INSIGHT) 서비스를 시작합니다...
SQL Server (INSIGHT) 서비스가 잘 시작되었습니다.
```

- DBCC (Database Consistency Check) 명령 사용

- ✓ `DBCC ERRORLOG`

- 시스템 저장 프로시저 (System Stored Procedure) 사용

- ✓ `EXEC master.sys.sp_cycle_errorlog`



에러 로그

▪ 로그 파일 유지

- 새로운 로그 파일이 생성될 때마다 사이클 전환
- ERRORLOG는 현재 로그 파일, 과거 로그 파일은 뒤에 숫자가 증가함
 - ✓ ERRORLOG 생성 (현재 로그)
 - ✓ ERRORLOG → ERRORLOG.1
 - ✓ ERRORLOG.1 → ERRORLOG.2
 - ✓ ERRORLOG.2 → ERRORLOG.3
 - ✓ ERRORLOG.3 → ERRORLOG.4
 - ✓ ERRORLOG.4 → ERRORLOG.5
 - ✓ ERRORLOG.5 → ERRORLOG.6
 - ✓ ERRORLOG.6 → 삭제됨



에러 로그

로그 파일 시간 정보

- 로그 파일 수정 날짜를 통해 인스턴스 재시작 시간 확인 가능

이름	만든 날짜	수정한 날짜	유형	크기
ERRORLOG	2011-10-10 오후 8:38	2012-01-26 오후 2:25	파일	2KB
ERRORLOG.1	2011-10-10 오후 8:38	2012-01-26 오후 2:25	1 파일	12KB
ERRORLOG.2	2011-10-10 오후 8:38	2012-01-26 오전 10:36	2 파일	14KB
ERRORLOG.3	2011-10-10 오후 8:38	2012-01-20 오후 1:59	3 파일	25KB
ERRORLOG.4	2011-10-10 오후 8:38	2012-01-20 오전 12:00	4 파일	12KB
ERRORLOG.5	2011-10-10 오후 8:38	2012-01-18 오후 9:23	5 파일	12KB
ERRORLOG.6	2011-10-10 오후 8:38	2012-01-17 오후 10:49	6 파일	13KB
log_88.trc	2012-01-18 오전 9:27	2012-01-18 오전 9:27	AtlTraceTo...	3KB

ERRORLOG 수정한 날짜: 2012-01-26 오후 2:25 만든 날짜: 2011-10-10 오후 8:38
파일 크기: 1.31KB

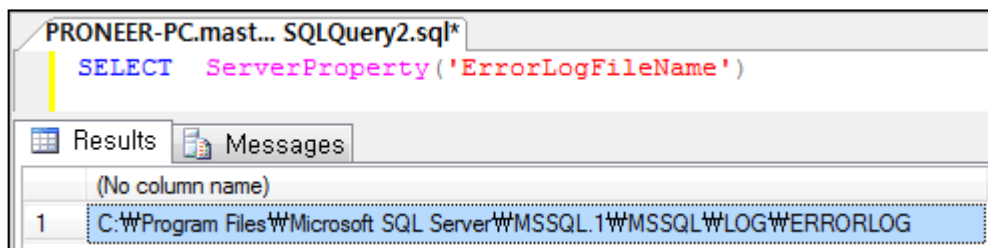


에러 로그

로그 파일 설정 확인

- 사용 중인 로그 파일명 확인

✓ `SELECT ServerProperty('ErrorLogFileName')`



- 전체 로그 파일 정보 나열

✓ `EXEC master.dbo.xp_enumerrorlog`

The screenshot shows a SQL query window titled 'PRONEER-PC.mast... SQLQuery2.sql*'. The query is `EXEC master.dbo.xp_enumerrorlog`. The 'Results' tab is active, showing a table with 4 columns: 'Archive #', 'Date', and 'Log File Size (Byte)'. The table contains 7 rows of data.

	Archive #	Date	Log File Size (Byte)
1	0	01/26/2012 14:25	1346
2	1	01/26/2012 14:25	11618
3	2	01/26/2012 10:36	13802
4	3	01/20/2012 13:59	25142
5	4	01/20/2012 00:00	12032
6	5	01/18/2012 21:23	11964
7	6	01/17/2012 22:49	12296



에러 로그

- 로그 파일 내용 확인 cont'd

1. 텍스트 편집기로 확인

```
C:\Program Files\Microsoft SQL Server\MSSQL1\MSSQL\LOG\ERRORLOG.1 - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(I) 매크로 실행 플러그인 창 ?
[Icons]
ERRORLOG.1
1 2012-01-26 10:37:50.56 Server      Microsoft SQL Server 2005 - 9.00.1399.0
2      Oct 14 2005 00:33:37
3      Copyright (c) 1988-2005 Microsoft Corporation
4      Developer Edition on Windows NT 6.1 (Build 7601: Service Pack 1)
5
6 2012-01-26 10:37:50.56 Server      (c) 2005 Microsoft Corporation.
7 2012-01-26 10:37:50.56 Server      All rights reserved.
8 2012-01-26 10:37:50.56 Server      Server process ID is 672.
length : 5816  lines : 59  Ln : 9  Col : 26  Sel : 0  Dos#Windows  UCS-2 Little Endian  INS
```



에러 로그

- 로그 파일 내용 확인 cont'd

2. Management Studio를 이용해 확인

The screenshot displays the Microsoft SQL Server Management Studio interface. On the left, the 'Log File Viewer - PRONEER-PC' window is open, showing a list of log files under 'Select logs'. The 'SQL Server' log is selected, and a list of log files is shown, including 'Current - 2012-01-26 오후 2:28' and several archives. The 'Status' section shows 'Last Refresh: 2012-01-26 오후 2:48:00' and 'Filter: None'. The 'Progress' section shows 'Done (97 records)'.

The main window displays the 'Log file summary: No filter applied' and a table of log entries. The table has columns for 'Date', 'Source', and 'Message'. The selected row details are shown at the bottom:

Date	Source	Message
2012-01-20 오후 1:59:39	spid12s	The current event was not reported to the Windows Events log. Operating system error = 1717(알 수 없는 인터페이스입니다.). You may need to clear the Windows Events log if it is full.

On the right, the 'Object Explorer' shows the server structure. The 'SQL Server Logs' folder is expanded, showing a list of log files. A context menu is open over the 'Archive #3 - 2012-01-20 오후 1:59:00' file, with options 'View SQL Server Log' and 'Refresh'.



에러 로그

▪ 로그 파일 내용 확인 cont'd

3. 시스템 확장 저장 프로시저 (System Extended Stored Procedure)를 이용해 확인

- ✓ `EXEC master.dbo.xp_readerrorlog 0, 1, "Error", "Login", '2012-01-01', '2012-01-12', ['desc' | 'asc']`
 - **파라미터 1** : 읽을 에러 파일 번호 (0은 현재 에러 로그 → ERRORLOG)
 - **파라미터 2** : **1** → SQL Server error log, **2** → SQL Server Agent log
 - **파라미터 3** : 검색어 (기본값은 NULL)
 - **파라미터 4** : 파라미터 3 검색어에 만족하는 결과 내 추가 검색어 (기본값은 NULL)
 - **파라미터 5** : 시작 날짜
 - **파라미터 6** : 마지막 날짜
 - **파라미터 7** : 정렬 여부



에러 로그

- 로그 파일 내용 확인

3. 시스템 확장 저장 프로시저 (System Extended Stored Procedure)를 이용해 확인

PRONEER-PC.mast... SQLQuery1.sql*			
<pre>exec master.dbo.xp_readerrorlog 3, 1, "Error", NULL, '2012-01-01', NULL, 'desc'</pre>			
'''			
Results Messages			
	LogDate	ProcessInfo	Text
1	2012-01-20 13:59:39.020	spid12s	The current event was not reported to the Windows Events log. Operating system error = 1717(알 수 없는 인터페이스)
2	2012-01-20 13:59:39.020	spid12s	Error: 17054, Severity: 16, State: 1.
3	2012-01-20 11:14:53.550	spid5s	Error: 8355, Severity: 16, State: 1.
4	2012-01-20 11:14:45.690	Server	-e C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\ERRORLOG
5	2012-01-20 11:14:45.690	Server	Logging SQL Server messages in file 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\ERRORLO

SQL Server Forensics

~~— SQL Server Error Log~~

- **Windows Application Log**
- **SQL Server Default Trace**
- **SQL Server Transaction Log**
- **SQL Server Data File**
- **SQL Server Memory**



윈도우 응용프로그램 로그

- 윈도우에서 동작하는 응용프로그램 관련 로그
- 에러 로그의 내용은 응용프로그램 로그에도 기록됨
- 위치
 - **XP/2003** : C:\Windows\System32\config\AppEvent.evt
 - **Vista/7** : C:\Windows\System32\winevt\Logs\Application.evtx



윈도우 응용프로그램 로그

- 로그 파일 내용 확인

- 이벤트 로그 뷰어를 이용한 분석

The screenshot shows the Windows Event Viewer window titled '이벤트 뷰어'. The left pane shows the tree structure with '응용 프로그램' (Application) selected under 'Windows 로그'. The right pane displays a list of events for '응용 프로그램' with 23,931 events. The table below shows the details of the events:

수준	날짜 및 시간	원본	이벤트 ID	작업 범주
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17162 (2)	
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17110 (2)	
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17176 (2)	
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17111 (2)	
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17104 (2)	
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17103 (2)	
정보	2011-11-14 오전 9:35:18	MSSQLSERVER	17101 (2)	

The event details for ID 17162 are shown in the bottom pane:

이벤트 17162, MSSQLSERVER

일반 | 자세히

SQL Server is starting at normal priority base (=7). This is an informational message only. No

로그 이름(M): 응용 프로그램

원본(S): MSSQLSERVER

로그된 날짜(D): 2011-11-14 오전 9:35:18



SQL Server 에러 로그 vs 응용프로그램 로그

- 두 로그 모두 동일한 정보 저장

- 약간의 시간차 존재

- ✓ 응용프로그램 로그가 에러 로그보다 기록 시간이 빠름

- 시간 단위 기록 (시간 필드로 정렬 시 주의)

- ✓ 응용프로그램 로그 : 초 단위 기록

- ✓ SQL Server 에러 로그 : ms 단위 기록

- 로그의 신뢰성

- 응용프로그램 로그 : 관리자만 삭제 가능

- SQL Server 에러 로그 : MSSQL 관리자(sa)만 삭제 가능

- SQL Server가 윈도우 관리자(LOCAL SYSTEM 혹은 Administrator)로 실행되지 않는 환경이라면?



SQL Server 에러 로그 vs 응용프로그램 로그

▪ 에러 로그 및 응용프로그램 로그에 기록되는 이벤트

- 로그인 실패 정보
- 백업/복구 정보
- 확장 저장 프로시저 DLL 로딩 정보
- 서버 옵션 활성화 여부(sp_configure)
- 일부 DBCC 명령 실행 정보
- 에러 메시지
-



SQL Server 에러 로그 vs 응용프로그램 로그

▪ 한계

- 에러 로그와 응용프로그램 로그에 남는 정보는 포렌식 분석에 크게 도움이 되지 않는 정보
- 포렌식 분석을 위해 더 중요한 정보
 - ✓ 저장 프로시저(확장 프로시저 포함) 실행 정보
 - ✓ DBCC 명령 실행 정보
 - ✓ DDL (Data Definition Language) 구문 사용 정보
 - CREATE, ALTER, DROP, RENAME, TRUNCATE
 - ✓ DML (Data Manipulation Language) 구문 사용 정보
 - INSERT, UPDATE, DELETE, SELECT, COMMIT, ROLLBACK
- 추가적인 로그 파일 분석 필요
 - ✓ 기본 흔적 파일, 트랜잭션 파일

SQL Server Forensics

~~— SQL Server Error Log~~

~~— Windows Application Log~~

- SQL Server Default Trace
- SQL Server Transaction Log
- SQL Server Data File
- SQL Server Memory



기본 흔적 로그

- 기본적으로 활성화되어 있음
- 인스턴스 실행 시 항상 백그라운드로 동작
- 시스템 장애를 해결하기 위한 목적의 이벤트만 캡처 (모든 이벤트를 캡처하지는 않음)
- 위치
 - C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\ (에러 로그 위치와 동일)



기본 흔적 로그

▪ 흔적 로그 생성

1. SQL Server 시작 또는 재시작 될 때
2. Default Trace 활성 옵션이 0 → 1 변경될 때
3. 사용 중인 흔적 로그가 20 MB를 넘었을 때

▪ 로그명

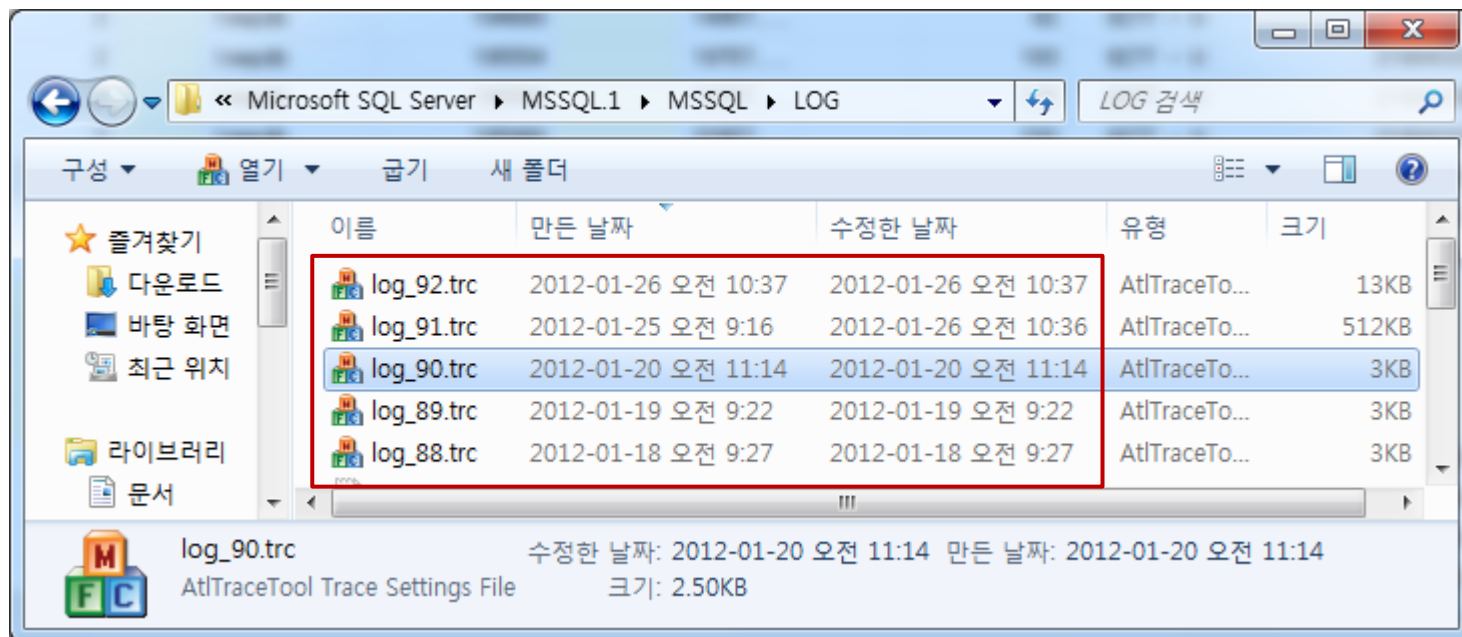
- **log_#.trc** (# = 숫자)
- 새로운 로그가 생성되면 기존 로그명의 숫자 + 1
- 기본적으로 5개의 흔적 로그만 유지 (새로운 로그가 생성되면 오래된 로그는 삭제됨)



기본 흔적 로그

■ 흔적 로그 시간 정보

- 흔적 로그 생성 시간을 통해 인스턴스의 시작 시점 및 옵션 변경 여부 확인 가능



기본 흔적 로그

■ 흔적 로그 설정 확인 cont'd

1. fn_trace_getinfo

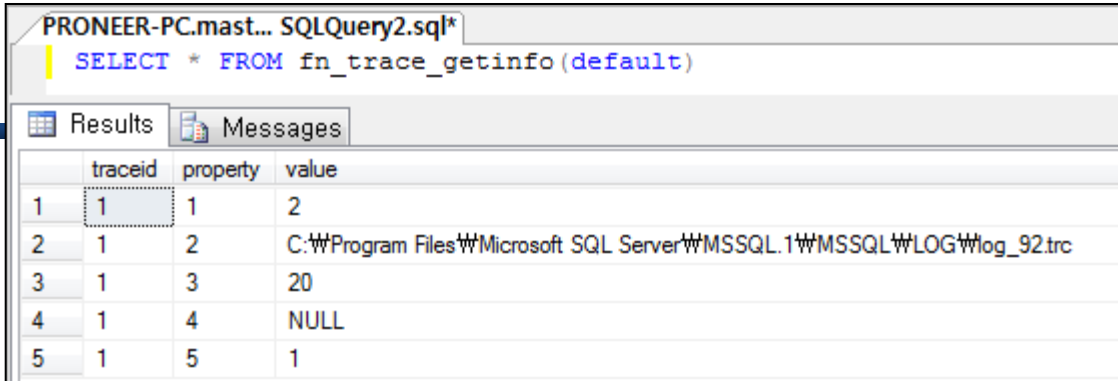
✓ `SELECT * FROM fn_trace_getinfo(default)`

✓ **traceid** : 흔적 식별자

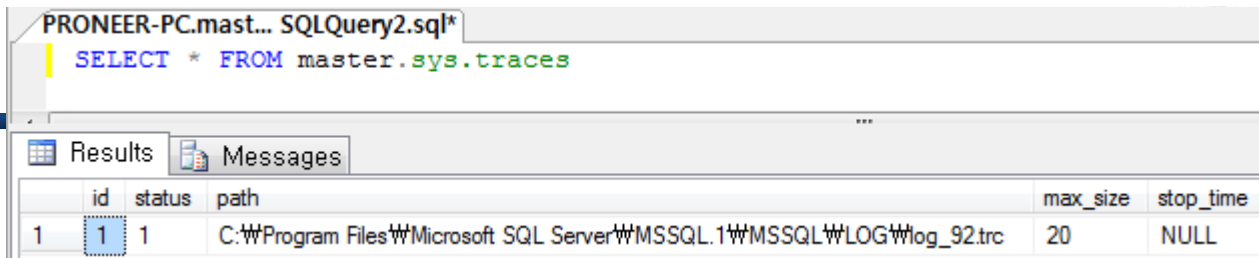
✓ **property** :

- 1 : 구성된 흔적 옵션
- 2 : 흔적 로그명
- 3 : 흔적 로그 최대 크기
- 4 : 중지 시간
- 5 : 현재 상태 (0 = Off, 1 = On)

✓ **value** : 값



	traceid	property	value
1	1	1	2
2	1	2	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\log_92.trc
3	1	3	20
4	1	4	NULL
5	1	5	1



기본 흔적 로그

■ 흔적 로그 설정 확인

2. master.sys.traces

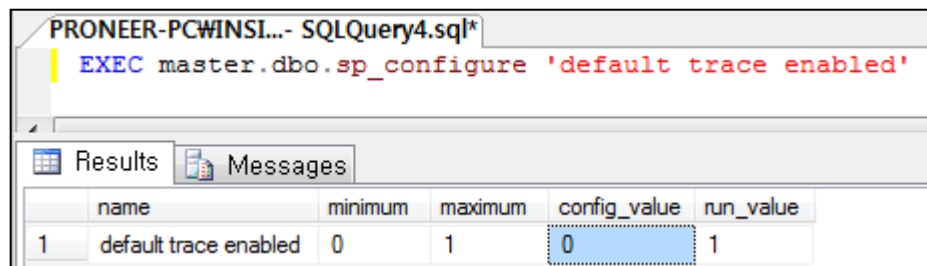
- ✓ `SELECT * FROM master.sys.traces`
- ✓ 18개의 컬럼 정보 반환
- ✓ **id** : 흔적 식별자
- ✓ **status** : 현재 상태 (0 = 중지됨, 1 = 동작중)
- ✓ **path** : 사용 중인 흔적 로그 경로
- ✓ **max_size** : 최대 크기
- ✓ **max_files** : 유지할 흔적 파일 수
- ✓ **start_time** : 흔적 기록 시작 시간
- ✓ **last_event_time** : 가장 최근 이벤트 기록 시간
- ✓ **event_count** : 기록된 이벤트 수



기본 흔적 로그

■ 흔적 로깅 상태 확인

- EXEC master.dbo.sp_configure 'default trace enabled',



PRONEER-PCWINSI...- SQLQuery4.sql*

```
EXEC master.dbo.sp_configure 'default trace enabled'
```

Results Messages

	name	minimum	maximum	config_value	run_value
1	default trace enabled	0	1	0	1

■ 흔적 로깅 비활성화

- EXEC master.dbo.sp_configure 'default trace enabled', 0

```
EXEC master.dbo.sp_configure 'allow updates', 1;  
EXEC master.dbo.sp_configure 'show advanced options', 1;  
EXEC master.dbo.sp_configure 'default trace enabled', 0;  
RECONFIGURE WITH OVERRIDE;  
EXEC master.dbo.sp_configure 'show advanced options', 0;  
EXEC master.dbo.sp_configure 'allow updates', 0;
```



기본 흔적 로그

- 흔적 로그 내용 확인 cont'd

- SQL Server Profiler를 이용해 확인

SQL Server Profiler - [C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\log_91.trc]

EventClass	NTUser...	NTDomain...	HostN...	ClientProcessID	ApplicationNa...	LoginN...	SPID	StartTime
Trace Start								2012-01-25 09:16:21..
Audit Server Starts And S...						sa	5	2012-01-25 09:16:21..
Object:Created	proneer	proneer-PC	PRONE...	5040	Microsoft SQ...	pronee...	51	2012-01-26 10:28:29..
Object:Deleted	proneer	proneer-PC	PRONE...	5040	Microsoft SQ...	pronee...	51	2012-01-26 10:28:30..
Object:Created	proneer	proneer-PC	PRONE...	1148	Microsoft SQ...	pronee...	51	2012-01-26 10:31:11..
Object:Deleted	proneer	proneer-PC	PRONE...	1148	Microsoft SQ...	pronee...	51	2012-01-26 10:31:11..
Object:Created	proneer	proneer-PC	PRONE...	4212	Microsoft SQ...	pronee...	51	2012-01-26 10:33:02..
Object:Deleted	proneer	proneer-PC	PRONE...	4212	Microsoft SQ...	pronee...	51	2012-01-26 10:33:02..
Object:Created	proneer	proneer-PC	PRONE...	1560	Microsoft SQ...	pronee...	51	2012-01-26 10:35:54..
Object:Deleted	proneer	proneer-PC	PRONE...	1560	Microsoft SQ...	pronee...	51	2012-01-26 10:35:54..

Done. Ln 0, Col 0 Rows: 47 Connections: 0



기본 흔적 로그

- 흔적 로그 내용 확인

2. fn_trace_gettable 함수를 이용해 확인

PRONEER-PC.mast... SQLQuery2.sql*										
<pre>SELECT * FROM fn_trace_gettable ('C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\log_88.trc', default)</pre>										
Results Messages										
	TextData	BinaryData	DatabaseID	TransactionID	LineNumber	NTUserName	NTDomainName	HostName	ClientProcessID	ApplicationName
10	NULL	NULL	2	194613	NULL	proneer	proneer-PC	PRONEER-PC	5040	Microsoft SQL
11	NULL	NULL	2	194660	NULL	proneer	proneer-PC	PRONEER-PC	5040	Microsoft SQL
12	NULL	NULL	2	195554	NULL	proneer	proneer-PC	PRONEER-PC	1148	Microsoft SQL
13	NULL	NULL	2	195566	NULL	proneer	proneer-PC	PRONEER-PC	1148	Microsoft SQL
14	NULL	NULL	2	195980	NULL	proneer	proneer-PC	PRONEER-PC	4212	Microsoft SQL
15	NULL	NULL	2	196014	NULL	proneer	proneer-PC	PRONEER-PC	4212	Microsoft SQL
16	NULL	NULL	2	196722	NULL	proneer	proneer-PC	PRONEER-PC	1560	Microsoft SQL
17	NULL	NULL	2	196755	NULL	proneer	proneer-PC	PRONEER-PC	1560	Microsoft SQL
18	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
19	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



기본 흔적 로그

▪ 흔적 로그에 기록되는 이벤트

- 실패한 로그인 시도
- 로그인 생성/수정/삭제 정보
- 테이블, 함수, 저장 프로시저와 관련된 흔적
- 서버 옵션 변경 정보 (sp_configure)
- 오브젝트 생성/삭제
- BACKUP, RESTORE 구문 관련 정보
- DBCC 명령 흔적
- DENY, GRANT, REVOKE 구문 관련 정보
-



기본 흔적 로그

- **한계**

- 시스템 확장 저장 프로시저 실행 정보
- DML (Data Manipulation Language) 구문 사용 정보
 - ✓ INSERT, UPDATE, DELETE, SELECT, COMMIT, ROLLBACK

SQL Server Forensics

~~— SQL Server Error Log~~

~~— Windows Application Log~~

~~— SQL Server Default Trace~~

- SQL Server Transaction Log
- SQL Server Data File
- SQL Server Memory



트랜잭션 로그

- 대부분의 데이터베이스는 롤백을 위해 트랜잭션 정보 저장
- SQL Server는 복구 모델에 따라 트랜잭션 정보 저장
 - Simple Recovery*
 - Full Recovery*
 - Bulk-logged Recovery
- 하나의 데이터 파일(.mdf) vs. 여러 개의 트랜잭션 파일(.ldf)



트랜잭션 로그

▪ 복구 모델 cont'd

• 단순 복구 (Simple Recovery)

- ✓ 일관성 검사만을 위한 최소한의 트랜잭션 정보만 기록
- ✓ 마지막 백업 시점으로만 복구 가능 (마지막 백업 이후에 변경된 내용은 복구 불가능)
- ✓ 데이터 백업 주기를 짧게 해야 함
- ✓ 읽기 전용 데이터이거나 테스트 작업 시 사용

• 전체 복구 (Full Recovery)

- ✓ 백업 이후 모든 트랜잭션 로그 기록 (다음 백업 시까지)
- ✓ 실패 시점으로 복구 가능
- ✓ 데이터 페이지 단위의 복구도 가능



트랜잭션 로그

▪ 복구 모델

• 대량 로그 복구 (Bulk-logged Recover)

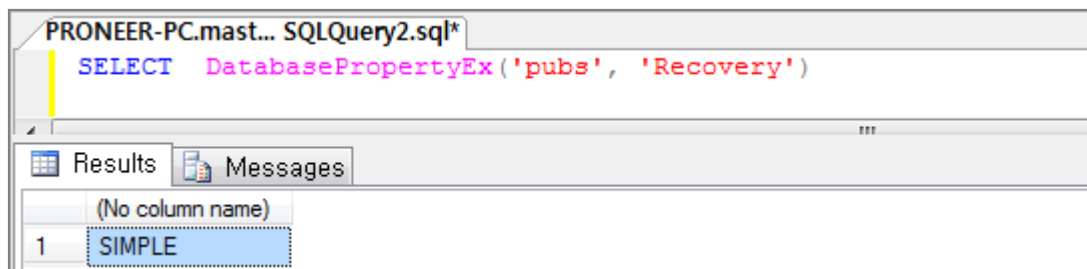
- ✓ 로그의 양이 대량일 때 사용하는 복구 방식
- ✓ 인덱스 생성, 대량 로드와 같이 많은 시간을 요하는 작업은 최소한의 정보만 기록
 - CREATE INDEX, SELECT INTO, BCP, BULK INSERT ...
- ✓ 대량 작업 시 성능 향상



트랜잭션 로그

▪ 데이터베이스 복구 모델 확인

- `SELECT DatabasePropertyEx('databasename', 'Recovery')`



▪ 시스템 데이터베이스 기본 복구 모델

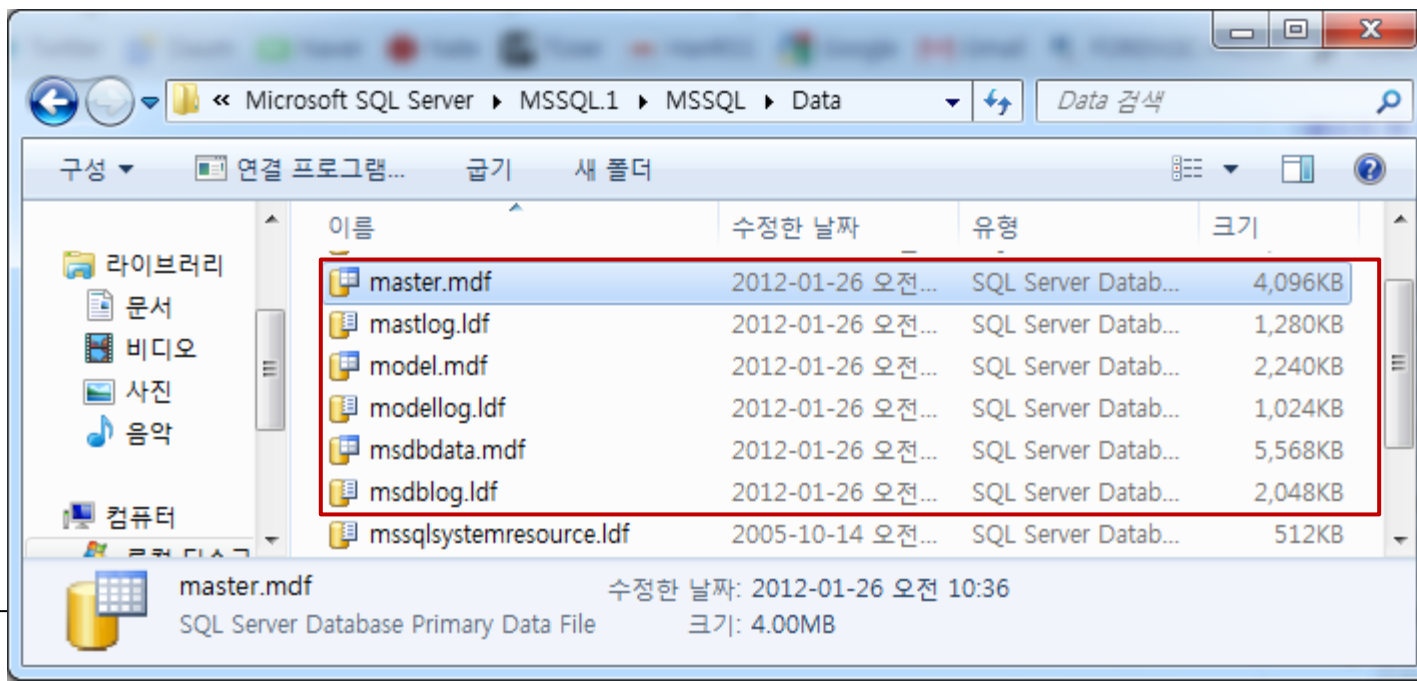
- **master** : 단순 복구
- **model** : 전체 복구 (새로 생성되는 데이터베이스는 "model" 데이터베이스의 복구 모델을 따름)
- **msdb** : 단순 복구
- **tempdb** : 단순 복구



트랜잭션 로그

■ 데이터 파일과 트랜잭션 로그 파일

- 데이터 파일 : *.mdf (예, insight.mdf)
- 트랜잭션 로그 파일 : *.ldf (예, insight_log.ldf)
- 데이터 파일 뒤에 "_log" 붙는 것이 일반적이나 시스템 데이터베이스는 별도의 규칙을 사용
- 보통 데이터 파일과 트랜잭션 파일은 분리해서 저장 (로그는 별도로 관리)

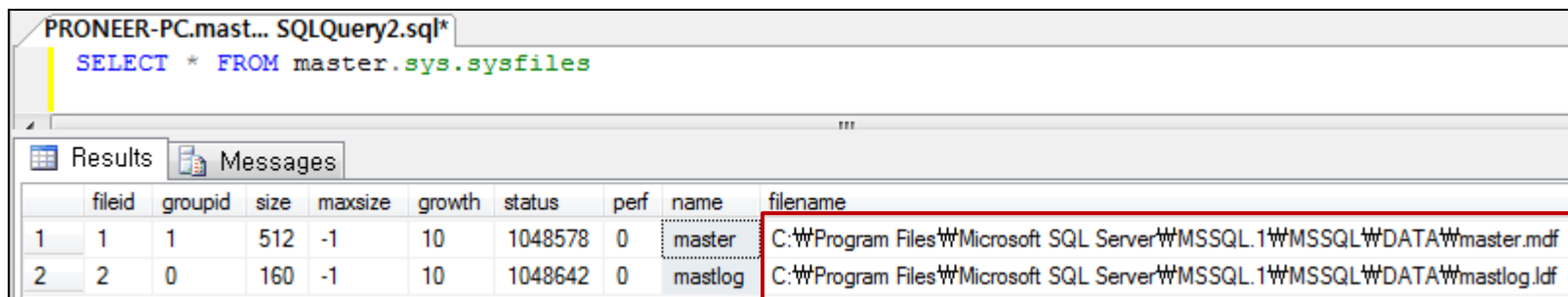




트랜잭션 로그

■ 데이터 파일과 트랜잭션 로그 파일

- 데이터 파일과 트랜잭션 로그가 분리되어 있는 경우, 트랜잭션 로그 위치 검색
- `SELECT * FROM databasename.sys.sysfiles`



	fileid	groupid	size	maxsize	growth	status	perf	name	filename
1	1	1	512	-1	10	1048578	0	master	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\master.mdf
2	2	0	160	-1	10	1048642	0	mastlog	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\mastlog.ldf



트랜잭션 로그

▪ 트랜잭션 로그 관리

- 트랜잭션 로그는 초기 생성된 이후 계속적으로 크기가 증가
- 트랜잭션 로그는 주기적으로 절단(truncation)이 필요
- 절단 작업은 로그 내부 구조 변경을 통해 로그 파일을 재사용 (덮어쓰기)
- **단순 복구**
 - ✓ CHECKPOINT 이후에 자동적으로 수행
 - ✓ CHECKPOINT : SQL Server 버퍼 캐시 내용(RAM) ➔ 디스크에 저장 (데이터 파일, 트랜잭션 로그)
- **전체 복구, 대량 로그 복구**
 - ✓ CHECKPOINT가 발생하고, 로그가 백업된 이후



트랜잭션 로그

▪ 트랜잭션 로그 관리

- 절단(truncation) 작업은 로그를 재사용하기 위한 것이므로 로그 크기는 줄어들지 않음
- 로그 크기를 줄이기 위해서는 **SHRINKFILE** 실행
- **DBCC SHRINKFILE** (log_name | id, size)

The screenshot shows a SQL query window titled 'PRONEER-PC.temp... SQLQuery3.sql*' containing the command `DBCC SHRINKFILE (templog, 1)`. Below the query window, the 'Results' tab is active, displaying a table with the following data:

	DblId	FileId	CurrentSize	MinimumSize	UsedPages	EstimatedPages
1	2	2	64	64	64	64



트랜잭션 로그

트랜잭션 로그 내용 확인 cont'd

1. fn_dblog (@StartingLSN, @EndingLSN)

✓ `SELECT * FROM ::fn_dblog (null, null)`

PRONEER-PC.mast... SQLQuery5.sql*							
<pre>SELECT * FROM ::fn_dblog (null, null)</pre>							
Results Messages							
	Current LSN	Operation	Context	Transaction ID	Tag Bits	Log Record Fixed Length	Log Record Length
1	000000ff:00000168:0001	LOP_BEGIN_CKPT	LCX_NULL	0000:00000000	0x0000	96	96
2	000000ff:00000170:0001	LOP_XACT_CKPT	LCX_NULL	0000:00000000	0x0000	24	420
3	000000ff:00000170:0002	LOP_END_CKPT	LCX_NULL	0000:00000000	0x0000	136	136
4	000000ff:00000178:0001	LOP_BEGIN_XACT	LCX_NULL	0000:00003072	0x0000	48	72
5	000000ff:00000178:0002	LOP_LOCK_XACT	LCX_NULL	0000:00003072	0x0000	24	44
6	000000ff:00000178:0003	LOP_MODIFY_ROW	LCX_CLUSTERED	0000:00003072	0x0000	62	112
7	000000ff:00000178:0004	LOP_PREP_XACT	LCX_NULL	0000:00003072	0x0000	64	68
8	000000ff:00000180:0001	LOP_COMMIT_XACT	LCX_NULL	0000:00003072	0x0000	48	52
9	000000ff:00000188:0001	LOP_FORGET_XACT	LCX_NULL	0000:00003072	0x0000	36	36
10	000000ff:00000188:0002	LOP_BEGIN_XACT	LCX_NULL	0000:00003073	0x0000	48	112
11	000000ff:00000188:0003	LOP_LOCK_XACT	LCX_NULL	0000:00003073	0x0000	24	44
12	000000ff:00000188:0004	LOP_COMMIT_XACT	LCX_NULL	0000:00003073	0x0000	48	52
13	000000ff:00000190:0001	LOP_BEGIN_XACT	LCX_NULL	0000:00003074	0x0000	48	72



트랜잭션 로그

트랜잭션 로그 내용 확인 cont'd

1. fn_dblog (@StartingLSN, @EndingLSN)

- ✓ `SELECT suser_sname(convert(varbinary, "Transaction SID")),
Operation, AllocUnitName, "Begin Time", "End Time" FROM ::fn_dblog(null, null)`

PRONEER-PC.mast... SQLQuery5.sql*

```
SELECT  
suser_sname(convert(varbinary, "Transaction SID")),  
Operation, AllocUnitName, "Begin Time", "End Time" FROM ::fn_dblog(null, null)
```

Results Messages

	(No column name)	Operation	AllocUnitName	Begin Time	End Time
1	NULL	LOP_BEGIN_CKPT	NULL	NULL	NULL
2	NULL	LOP_XACT_CKPT	NULL	NULL	NULL
3	NULL	LOP_END_CKPT	NULL	NULL	NULL
4	NULL	LOP_BEGIN_XACT	NULL	2012/01/26 10:37:51:723	NULL
5	NULL	LOP_LOCK_XACT	NULL	NULL	NULL
6	NULL	LOP_MODIFY_ROW	sys.sysdbreg.clst	NULL	NULL
7	NULL	LOP_PREP_XACT	NULL	NULL	NULL
8	NULL	LOP_COMMIT_XACT	NULL	NULL	2012/01/26 10:37:51:740
9	NULL	LOP_FORGET_XACT	NULL	NULL	NULL
10	NULL	LOP_BEGIN_XACT	NULL	2012/01/26 10:37:51:787	NULL
11	NULL	LOP_LOCK_XACT	NULL	NULL	NULL
12	NULL	LOP_COMMIT_XACT	NULL	NULL	2012/01/26 10:37:51:803



트랜잭션 로그

- 트랜잭션 로그 내용 확인 cont'd

2. DBCC LOG (database_name, [0|1|2|3|4])

✓ DBCC LOG (master, 4)

PRONEER-PC.mast... SQLQuery5.sql*							
DBCC LOG (master, 4)							
Results Messages							
	Current LSN	Operation	Context	Transaction ID	Tag Bits	Log Record Fixed Length	Log Record Length
1	000000ff:00000168:0001	LOP_BEGIN_CKPT	LCX_NULL	0000:00000000	0x0000	96	96
2	000000ff:00000170:0001	LOP_XACT_CKPT	LCX_NULL	0000:00000000	0x0000	24	420
3	000000ff:00000170:0002	LOP_END_CKPT	LCX_NULL	0000:00000000	0x0000	136	136
4	000000ff:00000178:0001	LOP_BEGIN_XACT	LCX_NULL	0000:00003072	0x0000	48	72
5	000000ff:00000178:0002	LOP_LOCK_XACT	LCX_NULL	0000:00003072	0x0000	24	44
6	000000ff:00000178:0003	LOP_MODIFY_ROW	LCX_CLUSTERED	0000:00003072	0x0000	62	112
7	000000ff:00000178:0004	LOP_PREP_XACT	LCX_NULL	0000:00003072	0x0000	64	68
8	000000ff:00000180:0001	LOP_COMMIT_XACT	LCX_NULL	0000:00003072	0x0000	48	52
9	000000ff:00000188:0001	LOP_FORGET_XACT	LCX_NULL	0000:00003072	0x0000	36	36
10	000000ff:00000188:0002	LOP_BEGIN_XACT	LCX_NULL	0000:00003073	0x0000	48	112
11	000000ff:00000188:0003	LOP_LOCK_XACT	LCX_NULL	0000:00003073	0x0000	24	44
12	000000ff:00000188:0004	LOP_COMMIT_XACT	LCX_NULL	0000:00003073	0x0000	48	52
13	000000ff:00000190:0001	LOP_BEGIN_XACT	LCX_NULL	0000:00003074	0x0000	48	72



트랜잭션 로그

- 트랜잭션 로그 내용 확인

- 3. 상업적인 도구 사용

- ✓ RedGate's SQL Log Rescue - <http://www.red-gate.com/products/dba/sql-log-rescue/>
 - ✓ ApexSQL's ApexSQL Log - http://www.apexsql.com/sql_tools_log.aspx



트랜잭션 로그

▪ 트랜잭션 로그에 기록되는 이벤트

- 각 트랜잭션의 시작, 종료 시간 정보
- 모든 데이터 수정(INSERT, UPDATE, DELETE 정보)
 - ✓ 저장 프로시저, 데이터 정의어(DDL), 시스템 테이블 변경 정보도 포함
- 모든 페이지 할당/해제 정보
- 테이블/인덱스 생성, 삭제 정보
- 롤백 정보
- 트랜잭션 SID



트랜잭션 로그

▪ 한계

- DBCC 명령 실행 정보 (에러 로그와 기본 흔적 파일에서 확인 가능)
- 시스템 확장 저장 프로시저 실행 정보
- SELECT 구문 사용 정보

SQL Server Forensics

- ~~— SQL Server Error Log~~
- ~~— Windows Application Log~~
- ~~— SQL Server Default Trace~~
- ~~— SQL Server Transaction Log~~
- SQL Server Data File
- SQL Server Memory



데이터 파일

- 데이터베이스 데이터가 저장되는 물리적인 저장소
- 하나의 데이터베이스는 여러 개의 데이터 파일을 가질 수 있음
 - **.mdf** : 주 (Primary) 데이터 파일
 - **.ndf** : 보조(Secondary) 데이터 파일
- 데이터를 삭제하면 트랜잭션 로그와 마찬가지로 비할당 공간으로 관리 (초기화 없음)
 - 삭제된 데이터 복구
- 데이터 파일은 SQL Server 엔진을 통해 확인하거나 다양한 뷰어가 존재

SQL Server Forensics

- ~~— SQL Server Error Log~~
- ~~— Windows Application Log~~
- ~~— SQL Server Default Trace~~
- ~~— SQL Server Transaction Log~~
- ~~— SQL Server Data File~~
- SQL Server Memory



메모리

- 데이터베이스는 성능을 위해 캐시(메모리) 사용
 - 데이터 캐시 (data cache)
 - ✓ 데이터 파일을 읽고 쓰는 캐시
 - 프로시저 캐시 (procedure cache)
 - ✓ 정기적으로 실행되는 SQL 구문의 실행 계획
- 관리되는 캐시 이외의 메모리 영역에 남아 있는 데이터
- 페이지 아웃된 데이터



메모리

데이터 캐시 내용 확인

- DBCC PAGE ({id | dbname}, filenum, pagenum [, printopt={0 | 1 | 2 | 3}])

PRONEER-PC\WINSI... SQLQuery3.sql* PRONEER-PC\WINSI... SQLQuery2.sql* PRONEER-PC\WINSI... SQLQuery1.sql*

DBCC PAGE (insight, 1, 11, 3)

Results Messages

	Fileid	Pageid	R...	Le...	xsdid (key)	uriord (key)	qual (key)	nameid (key)	symospace (key)	nmscope (key)	id	KeyHashValue
1	1	11	0	0	1	1	1	3	T	0	6	(5a00d55965f1)
2	1	11	1	0	1	1	1	4	T	0	11	(5b001c340495)
3	1	11	2	0	1	1	1	5	T	0	15	(5c005f20782)
4	1	11	3	0	1	1	1	6	T	0	16	(5d009a1cf2bb)
5	1	11	4	0	1	1	1	7	T	0	17	(5e00d90889ac)
6	1	11	5	0	1	1	1	8	T	0	18	(5f0008c73073)
7	1	11	6	0	1	1	1	9	T	0	19	(60004bd34b64)
8	1	11	7	0	1	1	1	10	T	0	20	(61008eefc65d)
9	1	11	8	0	1	1	1	11	T	0	21	(6200cdfb4a)
10	1	11	9	0	1	1	1	12	T	0	22	(63000496dc2e)
11	1	11	10	0	1	1	1	13	T	0	23	(64004782a739)
12	1	11	11	0	1	1	1	14	T	0	24	(650082be2a00)
13	1	11	12	0	1	1	1	15	T	0	25	(6600c1aa5117)
14	1	11	13	0	1	1	1	16	T	0	26	(670061272864)
15	1	11	14	0	1	1	1	17	T	0	27	(680022335373)
16	1	11	15	0	1	1	1	18	T	0	28	(6900e70fde4a)



메모리

데이터 캐시 메모리 내용 확인

- 가상 주소 얻어오기

✓ `SELECT * FROM sys.dm_os_virtual_address_dump`

PRONEER-PC\INSI... SQLQuery3.sql*						
PRONEER-PC\INSI... SQLQuery2.sql*						
PRONEER-PC\INSI... SQLQuery1.sql*						
<pre>SELECT * FROM sys.dm_os_virtual_address_dump</pre>						
Results Messages						
	region_base_address	region_allocation_base_address	region_allocation_protection	region_size_in_bytes	region_state	region_current_protection
1	0x00000000	0x00000000	0x00000000	65536	0x00010000	0x00000001
2	0x00010000	0x00010000	0x00000004	65536	0x00001000	0x00000004
3	0x00020000	0x00020000	0x00000002	4096	0x00001000	0x00000002
4	0x00021000	0x00000000	0x00000000	61440	0x00010000	0x00000001
5	0x00030000	0x00030000	0x00000004	479232	0x00002000	0x00000000
6	0x000A5000	0x00030000	0x00000004	4096	0x00001000	0x00000104
7	0x000A6000	0x00030000	0x00000004	40960	0x00001000	0x00000004
8	0x000B0000	0x000B0000	0x00000002	16384	0x00001000	0x00000002
9	0x000B4000	0x00000000	0x00000000	49152	0x00010000	0x00000001
10	0x000C0000	0x000C0000	0x00000002	4096	0x00001000	0x00000002
11	0x000C1000	0x00000000	0x00000000	61440	0x00010000	0x00000001
12	0x000D0000	0x000D0000	0x00000004	4096	0x00001000	0x00000004
13	0x000D1000	0x00000000	0x00000000	61440	0x00010000	0x00000001
14	0x000E0000	0x000E0000	0x00000004	4096	0x00001000	0x00000004
15	0x000E1000	0x00000000	0x00000000	61440	0x00010000	0x00000001



메모리

데이터 캐시 메모리 내용 확인

- 얻어온 가상 주소를 가지고 메모리 내용 얻어오기

✓ DBCC BYTES (starting_addrss, bytes)

The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'PRONEER-PCWINSI...- SQLQuery3.sql*' and 'PRONEER-PCWINSI... SQLQuery2.sql*'. The active tab displays the following SQL commands:

```
DBCC TRACEON (3604)
DBCC BYTES (1572864, 4096)
```

Below the SQL editor, the 'Messages' pane shows the output of the DBCC BYTES command, displaying memory dump data in hexadecimal and ASCII format:

```
00180000: 0277a73c 0277a73c 00000001 0277a73c <.w.<.w.....<.w.
00180010: 001a6170 001a6170 00000000 02a0b710 pa..pa.....
00180020: 000666de 00000000 000041d2 00000000 .f.....A.....
00180030: 00000000 00000000 00000000 00000000 .....
00180040: 00000000 00000000 000004d0 00000000 .....
00180050: 000002eb 000001e4 00000000 00000001 .....
00180060: 00180000 00000000 00000a59 00000000 .....Y.....
00180070: 00190050 00000001 00000001 00000000 P.....
00180080: 00180080 00180080 00000001 00000000 .....
```



메모리

프로시저 캐시

- `SELECT * FROM sys.syscacheobjects`
- `sql` 컬럼을 통해 SQL 구문 확인 가능

PRONEER-PC\WINSI... SQLQuery2.sql* PRONEER-PC\WINSI... - SQLQuery1.sql*

SELECT *
FROM sys.syscacheobjects

Results Messages

	bucketid	cacheobjtype	objtype	objid	dbid	dbidexec	uid	sql
1	16	Parse Tree	View	217	32767	1	4	CREATE VIEW sys.server_principals AS SELECT p.name, p.id AS principa...
2	32	Parse Tree	View	434	32767	5	4	CREATE VIEW sys.partition_parameters AS SELECT idmajor AS function_id,...
3	49	Extended Proc	Proc	1057316729	32767	0	4	xp_instance_regread
4	58	Extended Proc	Proc	524454186	32767	0	4	xp_msver
5	62	Compiled Plan	Adhoc	433753477	1	0	-2	SELECT s.name AS [Name], s.langid AS [LangID], s.dateformat AS [DateFo...
6	89	Parse Tree	View	135	32767	1	4	CREATE VIEW sys.sysindexkeys AS SELECT id = object_id, indid = con...
7	92	Compiled Plan	Adhoc	96495518	5	0	-2	SELECT 'Server[@Name=' + quotename(CAST(serverproperty(N'Servename')...
8	94	Parse Tree	View	392	32767	5	4	CREATE VIEW sys.system_columns AS SELECT id AS object_id, name c...
9	98	Parse Tree	View	194	32767	1	4	CREATE VIEW sys.syslanguages AS SELECT langid, dateformat, datefirst, ...
10	103	Compiled Plan	Adhoc	142109510	1	0	-2	SELECT s.name AS [Name], s.langid AS [LangID], s.dateformat AS [DateFo...
11	118	Parse Tree	View	213	32767	1	4	CREATE VIEW sys.databases AS SELECT d.name, d.id AS database_id, r...
12	138	Compiled Plan	Adhoc	600119928	1	0	-2	SELECT dtb.collation_name AS [Collation], dtb.name AS [DatabaseName2] F...
13	140	Parse Tree	UsrTab	386712786	32767	0	4	NULL
14	140	Parse Tree	UsrTab	386712786	32767	0	4	NULL



메모리

- 추가적인 메모리 분석
 - SQL Server 관련 프로세스 영역 덤프 후 분석
 - 페이지 파일 분석

SQL Server Anti-Forensics

- Introduction
- Disabling Error Logging
- Disabling the Default Trace
- Clearing the Master Transaction Log and Data File
- Clearing up SQL Server Memory



SQL Server 위협

▪ 공격 목적

- 데이터 조작 (생성/수정/삭제)
- 데이터 획득 (중요 데이터 유출)

▪ 안티포렌식

- 공격자는 자신의 공격 행위를 숨기기 위해 흔적을 지우거나 혼란시킴
- 흔적을 조작하기 위해서는 데이터베이스 관리자 권한이 필요



SQL Server 위협

- 관리자 권한(sysadmin)을 얻기 위한 방법
 - SQL Server의 취약점 익스플로잇
 - 아이디/패스워드로 로그인 시도 (전수조사, 사전공격 등)
 - 접근 제어 취약점 이용
 - 운영체제나 다른 응용프로그램의 취약점 익스플로잇
 - DBA에게 사회 공학 공격



SQL Server 보안성

- 로그인 실패 기록은 에러 로그와 응용프로그램 로그에 기록됨
- 기본 흔적(Default Trace) 파일의 기록
- 복구 모델에 의한 트랜잭션 기록
- xp_cmdshell은 기본 비활성화
- SQL Server는 일반적으로 운영체제 일반 사용자 계정으로 동작함

SQL Server Anti-Forensics

— Introduction

- Disabling Error Logging (include Application Log)
- Disabling the Default Trace
- Clearing the Master Transaction Log and Data File
- Clearing up SQL Server Memory



에러 로깅 우회

- **에러 로깅 우회 방법**
 - 에러 로그 삭제
 - 에러 로깅 비활성화
- 하지만, 우회 방법 모두 로깅됨 (응용프로그램 로그 파일, 기본 흔적 파일)
- 결국, 메모리 패칭이나 API 후킹이 필요



에러 로깅 메모리 패칭

- 메모리 패칭 cont'd

1. xp_cmdshell 이용

- ✓ 메모리 패칭 코드 실행
- ✓ xp_cmdshell은 특별한 경우를 제외하고는 활성화하지 않음 ➔ 가능성 낮음



에러 로깅 메모리 패칭

■ 메모리 패칭

2. 확장 저장 프로시저(System Extended Stored Procedure, XP) 활용

- ✓ SP는 SQL 구문으로 동작하지만, XP는 DLL로 구현됨
- ✓ 초기 XP 실행 시, 메모리에 DLL 로드
- ✓ `sp_addextendedproc`를 활용하여 사용자 정의 XP 등록
 - `EXEC master..sp_addextendedproc 'log_patching', 'log_patch.dll'`
 - 내부적으로 `DBCC ADDEXTENDEDPROC (@funcname, @dllname)` 호출
- ✓ 등록된 확장 프로시저 호출
 - `EXEC master..log_patching`



에러 로깅 메모리 패칭

■ 메모리 패칭 확인 방법

1. xp_cmdshell 이용

- ✓ 기본적으로 xp_cmdshell은 로깅되지 않음

2. 확장 저장 프로시저(System Extended Stored Procedure, XP) 활용

- ✓ 확장 프로시저 등록 시 로깅됨
 - 기본 흔적 파일에 로깅 (에러, 응용프로그램 로그에는 로깅되지 않음)
- ✓ 확장 프로시저가 등록되면 master 테이블에 추가
- ✓ master 테이블을 조사하여 정상적이지 않게 등록된 프로시저 확인
 - `SELECT * FROM master.sys.extended_procedures`
- ✓ 확장 프로시저 실행 시, DLL이 로그되어 DllMain() 실행
 - 에러 로그와 응용프로그램 로그에 로깅 (DllMain의 수정으로 로깅 방지 가능)



에러 로깅 API 후킹

- 에러 로깅에 사용하는 API
 - SQL Server 에러 로그
 - ✓ Ntdll.dll의 NTWriteFile()
 - 윈도우 응용프로그램 로그
 - ✓ Advapi32.dll의 ReportEventW()



에러 로깅 우회의 흔적

▪ 메모리 패칭

- xp_cmdshell은 기본 비활성화
- 확장 저장 프로시저 사용
 - ✓ 프로시저 등록 시 **기본 흔적(Default Trace)**에 로깅됨

▪ API 후킹

- 데이터베이스에 추가적으로 운영체제도 공격해야 함

SQL Server Anti-Forensics

— Introduction

— ~~Disabling Error Logging~~

- Disabling the Default Trace
- Clearing the Master Transaction Log and Data File
- Clearing up SQL Server Memory



기본 흔적 로깅 우회

▪ 기본 흔적 로깅 비활성화

- 비활성화되면 에러 로그와 응용프로그램 로그에 로깅됨
- 기본 흔적 로깅 비활성화 전, 에러 로그와 응용프로그램 로그를 비활성화

▪ 저장 프로시저를 이용한 비활성화

- EXEC master.dbo.sp_configure 'default trace enabled', 0

SQL Server Anti-Forensics

— Introduction

— ~~Disabling Error Logging~~

— ~~Disabling the Default Trace~~

- Clearing the Master Transaction Log and Data File
- Clearing up SQL Server Memory



트랜잭션 로그 및 데이터 파일 흔적 제거

▪ 트랜잭션 로그, 데이터 파일 흔적

- 공격자의 주요 목표는 중요 데이터 유출이나 조작
- 유출이나 조작 과정 ➔ 에러, 응용프로그램, 기본 흔적 로그
- 유출이나 조작 결과 ➔ 트랜잭션 로그, 데이터 파일
 - ✓ 확장 저장 프로시저를 이용한 공격 수행 시
 - master 파일에 프로시저 등록
 - 트랜잭션 로그에 트랜잭션 기록



트랜잭션 로그 및 데이터 파일 흔적 제거

- 관련 흔적을 삭제한 후, 불필요한 데이터(재사용 데이터) 운영체제에게 반환
- 트랜잭션 로그와 데이터 파일 줄이기(shrink)
 - `DBCC SHRINKFILE ({id | dbname} [, target_percent])`
 - master 데이터 파일 크기 축소
 - ✓ `DBCC SHRINKFILE (1, 1)`
 - ✓ `DBCC SHRINKFILE (1, 0)`
 - ✓ `DBCC SHRINKFILE (1, 1)`
 - master 트랜잭션 로그 크기 축소
 - ✓ `DBCC SHRINKFILE (2, 1)`
 - ✓ `DBCC SHRINKFILE (2, 0)`
 - ✓ `DBCC SHRINKFILE (2, 1)`



트랜잭션 로그 및 데이터 파일 흔적 제거

- 트랜잭션 로그와 데이터 파일의 재사용 공간 완전삭제(wiping)

```
WHILE @i < 1000
BEGIN
    BEGIN TRAN
    ... (code setting @randomvalue in each iteration)
    DBCC ADDEXTENDEDPROC (@randomvalue, @randomvalue)
    ROLLBACK TRAN
    SET @i = @i + 1
END
```

```
WHILE @i < 1000
BEGIN
    CHECKPOINT
    SET @i = @i + 1
END
```

SQL Server Anti-Forensics

- Introduction
- ~~Disabling Error Logging~~
- ~~Disabling the Default Trace~~
- ~~Clearing the Master Transaction Log and Data File~~
- Clearing up SQL Server Memory



SQL Server 메모리 흔적 제거

- 프로시저 캐시 흔적 제거
 - DBCC FREESYSTEMCACHE ('ALL')
- 데이터 캐시 흔적 제거
 - CHECKPOINT
 - DBCC DROPCLEANBUFFERS
- 저장 프로시저 캐시 제거
 - DBCC FLUSHPROCINDB (id)

Countermeasures



다양한 공격

- 에러 로그, 기본 흔적 로그의 비활성화 혹은 삭제 (완전 삭제)
- 로그의 비활성화가 아닌 자신의 로그를 수정
- 트랜잭션 로그 조작으로 인해 백업 체인 망가트림
- 공격 행위를 다른 사용자 계정으로 실행
 - SETUSER
 - EXECUTE AS
- 메모리에 백도어 삽입
- 공격 목적이 정보 유출이 아닌 시스템 장애인 경우?
- SELECT 구문 vs. 백업 파일 생성 후 유출



가상 공격 시나리오

1. 확장 프로시저 등록 후 실행 ➔ 에러 로그와 윈도우 응용프로그램 로그 비활성화
2. 기본 흔적 로그 비활성화
3. 기본 흔적 로그를 조작하거나 덮어쓰
4. 공격 작업 수행
5. 미리 작성한 확장 프로시저를 이용해 흔적 삭제
 - 등록한 확장 프로시저 삭제
 - 흔적 삭제
 - 기본 흔적 로그 활성화
 - 로드한 DLL 언로드 ➔ 에러 로그와 윈도우 응용프로그램 로그 활성화



그렇다면 방법은?

- 데이터베이스 모니터링
 - DBA 활동 모니터링
 - 내장 모니터 도구가 아닌 추가 도구 도입
- 주기적인 취약점 패치, 설정 확인
- 비밀번호 정책 강화
- 운영체제 계정과 데이터베이스 계정 분리

