Windows Vista and NTFS File System Internals

Exploration of Windows Vista Advanced Forensic Topics – Day 1

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Windows Client Forensics (Windows Vista Advanced Topics)

Transactional NTFS and Registry Explained

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

New Features

Important Changes of Interest

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Self-Healing File System

- Vista includes a "self-healing" function which can correct certain errors in the system
- Vista maintains a list of hashes of known files and checks the hashes periodically
- On non-system files Vista will validate the file metadata
- Files whose hashes do not match or metadata is not valid will be replaced the next time the system is rebooted

Microsoft

ervices

Self-Healing File System

- Equivalent to Chkdsk and defrag processes constantly running in the background
- This has the potential to decrease the usefulness of the free space and slack space on the disk as well as limit the ability to recover deleted files

Self-Healing File System

 Healed files can be identified by an examination of the event viewer logs
 –Event ID 130-133 in the System Event Log

- If there are too many healing event messages it will stop recording them and provide a summary event instead
 - -How many files were healed

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

New File System Tools

• Format

Now has a switch to zero out every sector on a volume a specified number of times

/F:size
/T:tracks
/N:sectors
/P:passes

Specifies the size of the floppy disk to format (1.44) Specifies the number of tracks per disk side. Specifies the number of sectors per track. Zero every sector on the volume passes times. This switch is not valid with /Q

Microsoft Confidential LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

ADS Exposed!

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N] [/0[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4] [drive:][path][filename] Specifies drive, directory, and/or files to list. ZA. Displays files with specified attributes. attributes **D** Directories R Read-only files Files ready for archiving Н Hidden files Ĥ System files S Ι Not content indexed files L Reparse Points Prefix meaning not ∕B Uses bare format (no heading information or summary). ∕C Display the thousand separator in file sizes. This is the default. Use /-C to disable display of separator. Same as wide but files are list sorted by column. ∕D Uses lowercase. New long list format where filenames are on the far right. ∕N ∕0 List by files in sorted order. S By size (smallest first) sortorder N By name (alphabetic) By extension (alphabetic) D By date/time (oldest first) Ε Prefix to reverse order G Group directories first Pauses after each screenful of information. /Q /R Display the owner of the file. Display alternate data streams of the file. /S Displays files in specified directory and all subdirectories. **∕**T Controls which time field displayed or used for sorting timefield C Creation

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

NTFS, TxF and WinFS

Explanations and Misconceptions

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved



The (first) NTFS partition (Volume Boot Record) in Windows Vista starts at sector 2048, not sector 63.

1C6	Sectors preceding partition 1	2048
1CA	Sectors in partition 1	4145152

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Bitlocker Volume Boot Sector

 Physical level view of the header of the boot sector of the second partition, the Bitlocker protected volume:

-EB 52 90 2D 46 56 45 2D 46 53 2D -ëR•FVE-FS-

Offset	0	1	2	3	4	5	6	- 7	8	- 9	A	В	С	D	Е	F	▼ [2
007E900000	EΒ	52	90	2D	46	56	45	2D	46	53	2D	00	02	08	00	00	ëR-FVI	E-FS
007E900010	00	00	00	00	00	F8	00	00	ЗF	00	\mathbf{FF}	00	00	48	ЗF	00	6	ø?.ÿH?.
007E900020	00	00	00	00	80	00	80	00	F0	\mathbf{DF}	AA	06	00	00	00	00		.∎.ðߪ

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Bitlocker Volume: Physical View

🔒 Boot Sector NTFS, Base Offset: 0

Offset	Title	Value
0	JMP instruction	EB 52 90
3	File system ID	-FVE-FS-
В	Bytes per sector	512
D	Sectors per cluster	8
E	Reserved sectors	0
10	(always zero)	00 00 00
13	(unused)	00 00
15	Media descriptor	F8
16	(unused)	00 00
18	Sectors per track	63
1A	Heads	255
1C	Hidden sectors	4147200
20	(unused)	00 00 00 00
24	(always 80 00 80 00)	80 00 80 00
28	Total sectors	111861744
30	Start C# \$MFT	786432
38	Start C# \$MFTMirr	44193
40	FILE record size indicator	-10
41	(unused)	0
44	Clusters per INDX block	1
45	(unused)	0
48	32-bit serial number (hex)	47 AF 10 08
48	32-bit SN (hex, reversed)	810AF47
48	64-bit serial number (hex)	47 AF 10 08 C2 10 08 4A
50	Checksum	0
1FE	Signature (55 AA)	55 AA

Services

Microsoft

Surprise!

Logical level view of the header of the boot sector of the Bitlocker protected volume (same physical sector):

-EB 52 90 4E 54 46 53 20 -ëRNTFS

Offset		1	2	3	4	5	6	- 7	8	- 9	Å	В	C .	D	Ε	F	
000000000	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ër NTFS
000000010	00	00	00	00	00	F8	00	00	3F	00	\mathbf{FF}	00	00	48	3F	00	ø?.ÿH?.
000000020	00	00	00	00	80	00	80	00	F0	DF	ÅÅ	06	00	00	00	00	∎.∎.ðߪ
000000000	0.0	0.0	00	0.0	0.0	00	00	0.0		$^{\circ}$ T		0.0	0.0	0.0	0.0	00	·· ,

Microsoft Services

Bitlocker Volume: Logical View

Boot Sector NTFS, Base Offset: 0

Offset	Title
0	JMP instruction
3	File system ID
В	Bytes per sector
D	Sectors per cluster
Е	Reserved sectors
10	(always zero)
13	(unused)
15	Media descriptor
16	(unused)
18	Sectors per track
1A	Heads
1C	Hidden sectors
20	(unused)
24	(always 80 00 80 00)
28	Total sectors
30	Start C# \$MFT
38	Start C# \$MFTMirr
40	FILE record size indicator
41	(unused)
44	Clusters per INDX block
45	(unused)
48	32-bit serial number (hex)
48	32-bit SN (hex, reversed)
48	64-bit serial number (hex)
50	Checksum
1FF	Signature (55 AA)

	Value
EB 52 90	
NTFS	
512	
8	
0	
00 00 00	
00 00	
F8	
00 00	
63	
255	
4147200	
00 00 00 00	
80 00 80 00	
111861744	
786432	
7679999	
-10	
0	
1	
0	
47 AF 10 08	
810AF47	
47 AF 10 08 C2 10 08 4A	
0	
55 AA	

Microsoft Services

Interesting

🔒 Boot Sect	or NTFS, Base Offset: 0		🔒 Boot Sect	or NTFS, Base Offset: 0		
Offset	Title		Offset	Title		Value
0	JMP instruction	EB 52 90	0	JMP instruction	EB 52 90	
3	File system ID	·FVE·FS·	3	File system ID	NTFS	
В	Bytes per sector	512	В	Bytes per sector	512	
D	Sectors per cluster	8	D	Sectors per cluster	8	
Е	Reserved sectors	0	E	Reserved sectors	0	
10	(always zero)	00 00 00	10	(always zero)	00 00 00	
13	(unused)	00 00	13	(unused)	00 00	
15	Media descriptor	F8	15	Media descriptor	F8	
16	(unused)	00 00	16	(unused)	00 00	
18	Sectors per track	63	18	Sectors per track	63	
1A	Heads	255	1A	Heads	255	
1C	Hidden sectors	4147200	1C	Hidden sectors	4147200	
20	(unused)	00 00 00 00	20	(unused)	00 00 00 00	
24	(always 80 00 80 00)	80 00 80 00	24	(always 80 00 80 00)	80 00 80 00	
28	Total sectors	111861744	28	Total sectors	111861744	
30	Start C# \$MFT	786432	30	Start C# \$MFT	786432	
38	Start C# \$MFTMirr	44193	38	Start C# \$MFTMirr	7679999	
40	FILE record size indicator	-10	40	FILE record size indicator	-10	
41	(unused)	0	41	(unused)	0	
44	Clusters per INDX block	1	44	Clusters per INDX block	1	
45	(unused)	0	45	(unused)	0	
48	32-bit serial number (hex)	47 AF 10 08	48	32-bit serial number (hex)	47 AF 10 08	
48	32-bit SN (hex, reversed)	810AF47	48	32-bit SN (hex, reversed)	810AF47	
48	64-bit serial number (hex)	47 AF 10 08 C2 10 08 4A	48	64-bit serial number (hex)	47 AF 10 08 C2 10 08 4A	
50	Checksum	0	50	Checksum	0	
1FE	Signature (55 AA)	55 AA	1FE	Signature (55 AA)	55 AA	

Microsoft Services

Bitlocker Volume: Physical View



LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Bitlocker Volume: Physical View

		Root directory)
— _□ , ¬ 🔂 □ , Pù£'yE, ¬Ë,		
ြက္ကြ 🚘 🛶 စ0 ခါ ကုန်းလိုက္က အန	_ <u> </u> _ <u>∨</u>	folume slack 8.0 KB
		dle space
-D © Vñ″µ~ ₽ ÷.{ R		
	4383EF400	~Ò,ójFç,» .ü−¥.þôîºg∎¯∎.7cF¢
	4383EF440	¢∥a:a∥¢ö∥z∥ý.54ö∥-".815ñ%C3&IŨ
	4383EF480	¥D%9.ÄÙ.∎∎Æ*æ MGg∎ðc?.Aý.∎-∎t;
<u>>_</u> [] [] [] [] [] [] [] [] [] [] [] [] []	4383EF4C0	1∎-WgS%∎ÚXm&žñë.¹Cé.Τ∎6V. (
	4383EF500	.å.∎á.~.¶ 9²¥ I.o∎0S.ٰd∎£¼Ã≸
	4383EF540	^7~o@.V0ÀÖ¥≪ûÁ.@2∎Ï∎Àne∎ö-#Cø{%∎
-D-1000 Blua FVI.0~U	4383EF580	%¿Sl:tâXKáX∎äá%.∎o2eÄ© ¼Å.òi.È4w
— <u> </u>	4383EF5C0	Zº¹.=åàJ.¶íüÌôVdÍ′RÚWñkFtZG∥7[
—————————————————————————————————————	4383EF600	H∎^oJ.[àBræÒ·¦WMÑ¢ñ∎¬·∎.cÃ∎¦á°
	4383EF640	dÒĮ.ÏĮĺôhĮF¦∖Õß`Ô+ªëð.È.»ýÊiî-
⊢D□ 🙀 J zYOI2O.{Æ@	4383EF680	õÞïôu∥.∥ÌÙ'So°yÀ½BK∥u∥£ÀwãÈñ¼}c÷
L ि Jºžk□őàM.Ë\ ƒ	4383EF6C0	ÊHîBkÙ¬KÁ°AµÖ¬à&ðÝ∎ÖÔU≪.Î\$;#s.∖.
	4383EF700	ë¬Á.∎á>Th∎.µÈkØu.æl¨Bt.²%ýÿ∎g∎
	4383EF740	¦Ñ\$ð² !.x;Sÿ8µúµ`Q.?iWÌ.rØSÈá]
<u>D</u> <mark>C</mark> _UI-"ø0á.⊤ÓÐ	4383EF780	D∎CÁ·∥∥_ª°MÚq©.`Öê.'Ó.@.}.£Íº_
	4383EF7C0	9∎¶E±à%yéîÛ¨üqú∎ÇE∎ýÈGÿÃ.ÿ <uî?p.< td=""></uî?p.<>

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Bitlocker Volume: Logical View



Bitlocker Volume: Logical View

	🗆 📄 \$BadClus		0 bytes	: 09/05/2006 17:11
É−D□∞g⊂	🗆 🛅 \$BadClus:\$B	ad	(O bytes	09/05/2006 17:11
🛖 🕞 🦳 🍋 \$Extend	🗆 📄 \$Bitmap		1.7 ME	09/05/2006 17:11
	🗆 🦲 \$Boot		8.0 KE	09/05/2006 17:11
	🗆 🏹 \$LogFile		64.0 ME	09/05/2006 17:11
œ-D □ 🔂 Boot	🗆 🏹 \$MFT		101 ME	09/05/2006 17:11
-D 🔂 🔁 documents and se	🗆 🏹 \$MFT:\$Bitma	ip	16.0 KE	09/05/2006 17:11
Lo Co Co idwlog	🗆 🏹 \$MF ^{set}	-		5/2006 17:11
	□ 🚺 \$Sec ¹⁰⁰⁰	9./.7./.2.0.0.67.:	.3.7.:.4.5.	
		?.?.N.D.:.N.W.i.n.d.c).w.s.\.s.y.s	t.e. 5/2006 17:11
🖶 🕞 🔂 🔁 🔁 🔁 🖶		V.E.R.S. \.W. 3.2. A.8.t).\.J.\.N.e.W	2 5/2006 17:11
🛱 🕞 🦳 🔂 Program Data	21100	R.S.N.W.3.2.X.8.6.N.3	}.∖.F.X.S.U.I	D.
System Volume In	Formation ²¹¹⁴⁰	a9./.7./.2.0.0.6	57.:.3.7.:	.4.5.
	21180	N.?.?.N.D.:.N.₩.i	n.d.o.w.s.\	.s.y.
🛱 🕞 🔂 🔂 🔂	21100	∧.f.x.s.F.D.7.6t.m	1.p., .d.e	.1.e.
	21200	,U.X.C.U.U.U.U.U.	3.49././	./.2.
	21240	9 / 7 / 2 0 0 6 2	$3 \cdot 3 - 2 \cdot 2$	0
		.\.?.\.C.:.\.W.i.r	1.d.o.w.s.∖.s	.y.s.
	C1300	R.I.V.E.R.S.\.W.3.2.X	.8.6.×.3.×.N	.e.w.
	C1340	?.?.∖.C.:.∖.W.i.n.d.c).₩.S.∖.S.Y.S	.t.e.
	C1380	V.E.R.S.\.W.3.2.X.8.6	5.\.3.\.m.d.i	.u.i.
	C13C0	0.3.a9./.7./.2.0).0.62.3.:	. 3 . 2 .
		r.o.r.:	. N. W. 1. n. d. O	w.s.
NFORCEMENT SENSITIVE INFORMATION - D	O NOT SHARE THESE	MATERIALS	Vicrosoft	Services

©2007 Microsoft Corporation – All Rights Reserved

LAW E

NTFS Benefits over FAT

- Data recovery
- Security
- Fault tolerance
- Larger files and file systems
- Multiple data streams

- UNICODE names
- Sparse files
- Encryption
- Journaling
- Volume Shadow
 Copies
- File compression

Microsoft Services

Journaling in NTFS

 Before a change is made to the metadata of a file, a transaction is logged in \$LOGFILE. These transactions list the operations required to redo or undo the changes. Once the transaction has been logged, the file system can go ahead and perform the change. Once it has completed the actual change to the data, a commit record is added to the log to show that has been successful

Journaling in NTFS

- With this transaction log, it's possible for NTFS to quickly recover from a system failure by replaying the redo or undo transactions that do not have a commit record
- To keep the size of the \$LOGFILE to a minimum, a checkpoint record is written to the log every so often (for example every five seconds)

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Journaling in NTFS

 Note that Journaling only protects file system metadata, it does not actually protect the data stored within files

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Transactional NTFS (TxF)

- NTFS APIs and on-disk structure do not change
- Allows both files and registry keys to be written to transactionally

Microsoft

bervices

Transactional NTFS (TxF)

- TxF basically adds a durability component to the previous NTFS transaction model as well as application to the full file, and not just the metadata
- Previously, a power outage during a system update would result in file corruption

http://msdn2.microsoft.com/en-us/library/aa365456.aspx

Microsoft

ervices

Are TxF and WinFS the same thing? NO

- -TxF refers to the way data is written to the file system in an ACID fashion
- WinFS as it was planned was primarily a presentation layer function to facilitate better indexing and searches of the file system

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

 Doesn't Windows Vista use a new version of NTFS called TxF? NO

–Windows Vista uses NTFS 3.1 which is the same version used in Windows XP and Windows Server 2003.

- With transactions in NTFS and the registry, there will no longer be any data corruption problems right? NO
 - –Transactions limit the possibility of software based corruption and some hardware based corruption such as power outages, but other hardware problems can result in corruption such as a failing disk drive or memory error

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

 Do all files use transactions to write data in Windows Vista? NO

 Applications must be written to take advantage of TxF

Microsoft

Services

- -The OS itself does not use TxF for all files
- Most Microsoft Applications do not use TxF....yet

Are there log files that I can examine to identify recent TxF activities? KIND OF
Log files exist, but will be limited in their duration and forensic content

What TxF Doesn't Do

• TxF Does Not

- -Change the on-disk format
- -Force you to use new file system APIs
- -Block multi-boot scenarios
- -Slow down normal file operations

Common Log File System

Shipped in W2003 R2 Used by KTM, TxF, TxR, and Cluster in Vista Features:

- User and Kernel mode APIs
- High-performance logging
- Shared IO of multistream writers
- Policy based management

- Restart Records, Reservations, Record Chaining
- Supports Circular and Linear Logging
- Archiving
- Torn write detection
- Et cetera...

Microsoft Services

Important Tools

• FSUtil

- -View transaction status
- -View file participation
- -Query metadata use
- -Manage log
- -Start/Stop secondary RMs

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Important Tools

FSUtil

– Resource Info >check status and settings - Resource setlog >Size : log size in containers >Minextents : policy -- max number of containers >Maxextents : policy – min number of containers >Growth : by how much should it grow >Shrink : by how much should it shrink - Resource SetAutoReset >Totally wipes out all TxF metadata on reboot! Microsoft Corporation Day Rights Reserved LISU

Services

Administrator: Command Prompt

C:\Windows\System32>fsutil resource info C:\ RM Identifier: 3ABEC360-B8EA-11DB-9321-001641E2F160 <u>\Device\HarddiskUolume2\\$Extend\\$RmMetadata\\$TxfLog\\$TxfLog::KtmLog</u> KTM Log Path for RM: Space used by TOPS: 20 Mb TOPS free space: $100 \times$ RM State: Active Running transactions: 0 2 One phase commits: Two phase commits: System initiated rollbacks: 0 Age of oldest transaction: 00:00:00 Logging Mode: Simple Number of containers: 2 10 Mb Container size: Total log capacity: 20 Mb Total free log space: 19 Mb Minimum containers: 2 20 Maximum containers: Log growth increment: 2 container(s) Auto shrink: Not enabled C:\Windows\System32>fsutil resource setlog --- SETLOG Commands Supported ---growth Change the automatic growth settings maxextents Change the maximum number of containers Change the minimum number of containers minextents mode Switch between undo only logging and full logging rename Change the RM's Guid shrink Change the automatic shrink settings size Change the number of containers explicitly C:\Windows\System32>fsutil transaction ---- TRANSACTION Commands Supported ---commit Commit a specified transaction list Display currently running transactions fileinfo Display transaction information for a specific file Display information on a specified transaction query rollback Rollback a specified transaction C:\Windows\System32>_

Important Tools

• KTMUtil

Lists existing transactions
i.e.. C:\>Ktmutil list tms
Force transaction outcome

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

C:\Windows\System32>ktmutil — Commands Supported · list List objects resolve Resolve force Force an Indoubt transaction to a particular outcome forget Forget a committed transaction C:\Windows\System32>ktmutil list Usage : ktmutil list transactions [{tmguid}] ktmutil list tms Eg : ktmutil list transactions {tmguid} <u>C:\Window</u>s\System32>ktmutil list tms {2e0be609-6a72-11db-84c3-0014220f08bb> {e28e381b-b8ed-11db-932c-0016cff24c99> {3abec35b-b8ea-11db-9321-001641e2f160} {3abec35f-b8ea-11db-9321-001641e2f160> {20011467-b86d<u>-11db-847b-0016cff24c99}</u> <u>{62c95faf-7fd5-11db-8180-0016179decf4}</u> {e28e38b1-b8ed-11db-932c-0016cff24c99> {42133cde-6a70-11db-bbc9-806e6f6e6963> {42133ce7-6a70-11db-bbc9-fdca8d8bcc9d> {42133ceb-6a70-11db-bbc9-fdca8d8bcc9d}

 ΣZ

C:\Windows\System32>_

Windows File System (WinFS)

- WinFS (Windows File System) was intended to facilitate the user experience by enhancing indexing and search functions of the file system including
 - Integrated Storage
 - Full Text Search
 - Advanced Search and Data Aggregation
 - Data Mining
- WinFS has been shelved delayed as a core component in Windows
 - Components of it are being included in other programs such as the .NET Framework.
 - No announcement has been made as to how or when/it will be delivered as a component in Windows

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved

Solid State Drives and Windows 7

- Flash vs DRAM vs Plattered Disks
- SSD and Flushing memory cells
- Windows 7 supports Trim Operations
- Prefetch/Superfetch/ReadyBoost/ReadyDrive are all disabled by defalt
 - -As long as the SSD meets performance metrics

Microsoft

ervices

- SSD and Bitlocker
- SSD and the Pagefile



Questions?

LAW ENFORCEMENT SENSITIVE INFORMATION – DO NOT SHARE THESE MATERIALS ©2007 Microsoft Corporation – All Rights Reserved