
USNジャーナル解析の追求

Japan Security Analyst Conference 2018

株式会社サイバーディフェンス研究所 山崎 輝

自己紹介

- ▶ 山崎 輝/YAMAZAKI Teru
- ▶ 株式会社サイバーディフェンス研究所 情報分析部
 - ▶ フォレンジック調査/インシデント対応支援
 - ▶ フォレンジック技術トレーニング/研修
 - ▶ フォレンジック関連ツール開発
- ▶ Twitter : @4n6ist
- ▶ Web : <https://www.kazamiya.net/>

Windows フォレンジック

イベントログ

レジストリ

プリフェッチ

\$MFT

\$UsnJrnl

Amcache

SRUDB

ショートカット

ジャンプリスト

ESEDB

個別ログ

...

本テーマの背景

- ▶ USNジャーナル (\$UsnJrnl) は通常残っている
- ▶ 既存のツール/アプローチのみでは不十分な可能性
 - ▶ 過去のデータを復元できていないケース
 - ▶ 大量のデータを見きれていないケース
- ▶ 最終的に人が見るべきデータに落とし込む

トピック

1. USNジャーナルの基本
2. USNジャーナルのカービング
3. USNジャーナルの解析

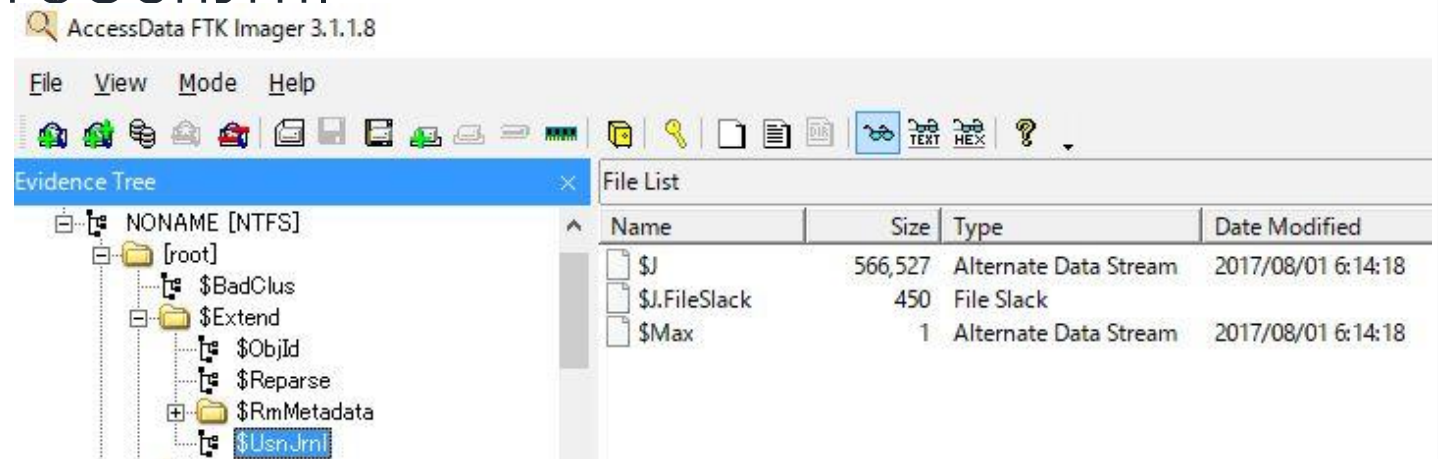
1. USNジャーナルの基本

USNジャーナルとは (1)

- ▶ バックアップ/アンチウイルス等の処理高速化を目的とした機能
- ▶ NTFS上の**ファイル/フォルダに対する変更処理を記録**
- ▶ Windows 2003 SP2/Vistaから**システムドライブで標準有効**
- ▶ ファイルパス \$Extend¥\$UsnJrnl

- ▶ **\$J … メイン**

- ▶ **\$Max … 管理情報**



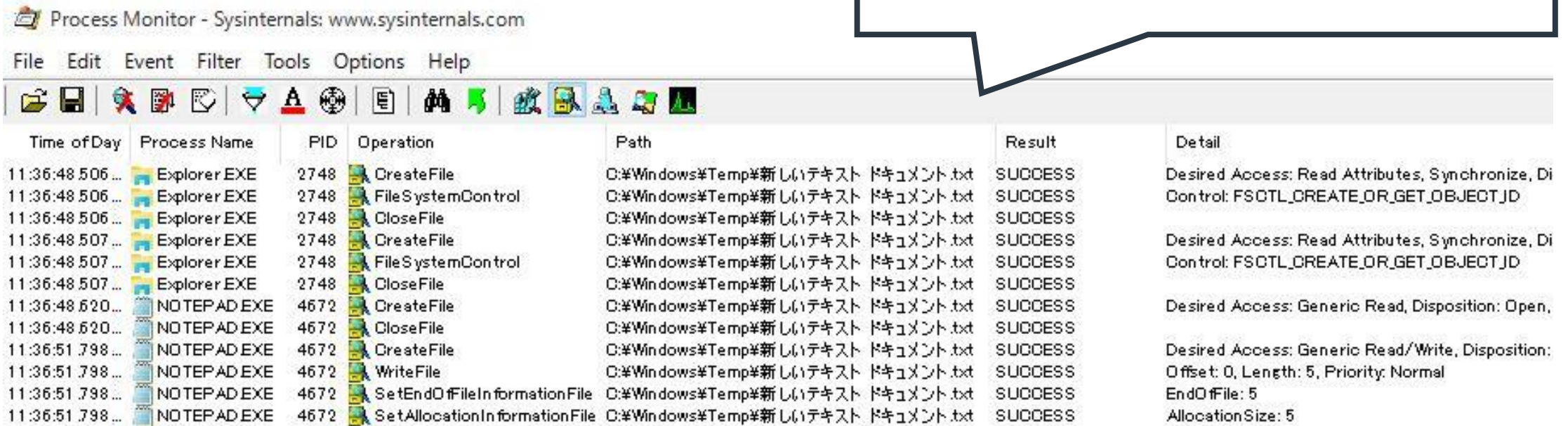
Change Journals

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa363798\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363798(v=vs.85).aspx)

USNジャーナルとは (2)

- ▶ Process Monitorの以下の条件で取得するデータに近い
 - ▶ File System Activity
 - ▶ Query関連のオペレーションを除外

- ✓ いつ (時間)
- ✓ どこで (ファイル/フォルダ)
- ✓ 何が (オペレーション)



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path	Result	Detail
11:36:48.506...	Explorer.EXE	2748	CreateFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Desired Access: Read Attributes, Synchronize, Di
11:36:48.506...	Explorer.EXE	2748	FileSystemControl	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Control: FSCTL_CREATE_OR_GET_OBJECT_ID
11:36:48.506...	Explorer.EXE	2748	CloseFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	
11:36:48.507...	Explorer.EXE	2748	CreateFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Desired Access: Read Attributes, Synchronize, Di
11:36:48.507...	Explorer.EXE	2748	FileSystemControl	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Control: FSCTL_CREATE_OR_GET_OBJECT_ID
11:36:48.507...	Explorer.EXE	2748	CloseFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	
11:36:48.620...	NOTEPAD.EXE	4672	CreateFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Desired Access: Generic Read, Disposition: Open,
11:36:48.620...	NOTEPAD.EXE	4672	CloseFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	
11:36:51.798...	NOTEPAD.EXE	4672	CreateFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Desired Access: Generic Read/Write, Disposition:
11:36:51.798...	NOTEPAD.EXE	4672	WriteFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	Offset: 0, Length: 5, Priority: Normal
11:36:51.798...	NOTEPAD.EXE	4672	SetEndOfFileInformationFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	EndOfFile: 5
11:36:51.798...	NOTEPAD.EXE	4672	SetAllocationInformationFile	C:\Windows\Temp\新しいテキストドキュメント.txt	SUCCESS	AllocationSize: 5

USNジャーナルの管理情報 (1)

- ▶ \$MaxにUSNジャーナルの管理情報を記録 (32バイト)
- ▶ 8バイト毎に4種類の情報を保持

最大サイズ	割り当て差分
USNジャーナルID	最も下位の有効なUSN

- ▶ サンプル

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
00000000	00	00	00	02	00	00	00	00	00	00	80	00	00	00	00	00
00000010	B3	FD	74	68	8D	0A	D3	01	00	00	00	00	00	00	00	00	..th.....

0x00000000002000000	0x00000000000800000
0x01D30A8D6874FDB3	0x00000000000000000

USNジャーナルの管理情報 (2)

- ▶ Windows標準コマンドで参照可能

> *fsutil usn queryjournal* ドライブ

```
管理者: Windows PowerShell
PS C:\> fsutil usn queryjournal F:
USN ジャーナル ID       : 0x01d30a8d6874fdb3
最初の USN              : 0x0000000020780000
次の USN                : 0x00000000229400c0
最も下位の有効な USN   : 0x0000000000000000
最大 USN                : 0x7fffffff0000
最大サイズ              : 0x0000000020000000
割り当て差分            : 0x0000000008000000
サポートされている最小レコードバージョン: 2
サポートされている最大レコードバージョン: 4
書き込み範囲の追跡: 無効
```

各情報の変換後

内容	16進表記	変換後
最大サイズ	0x0000000002000000	33,554,432
割り当て差分	0x0000000008000000	8,388,608
USNジャーナルID※	0x01D30A8D6874FDB3	2017/08/01 06:14:18
最も下位の有効なUSN	0x0000000000000000	0

※ USNジャーナルIDはUSNジャーナル機能有効時のタイムスタンプ

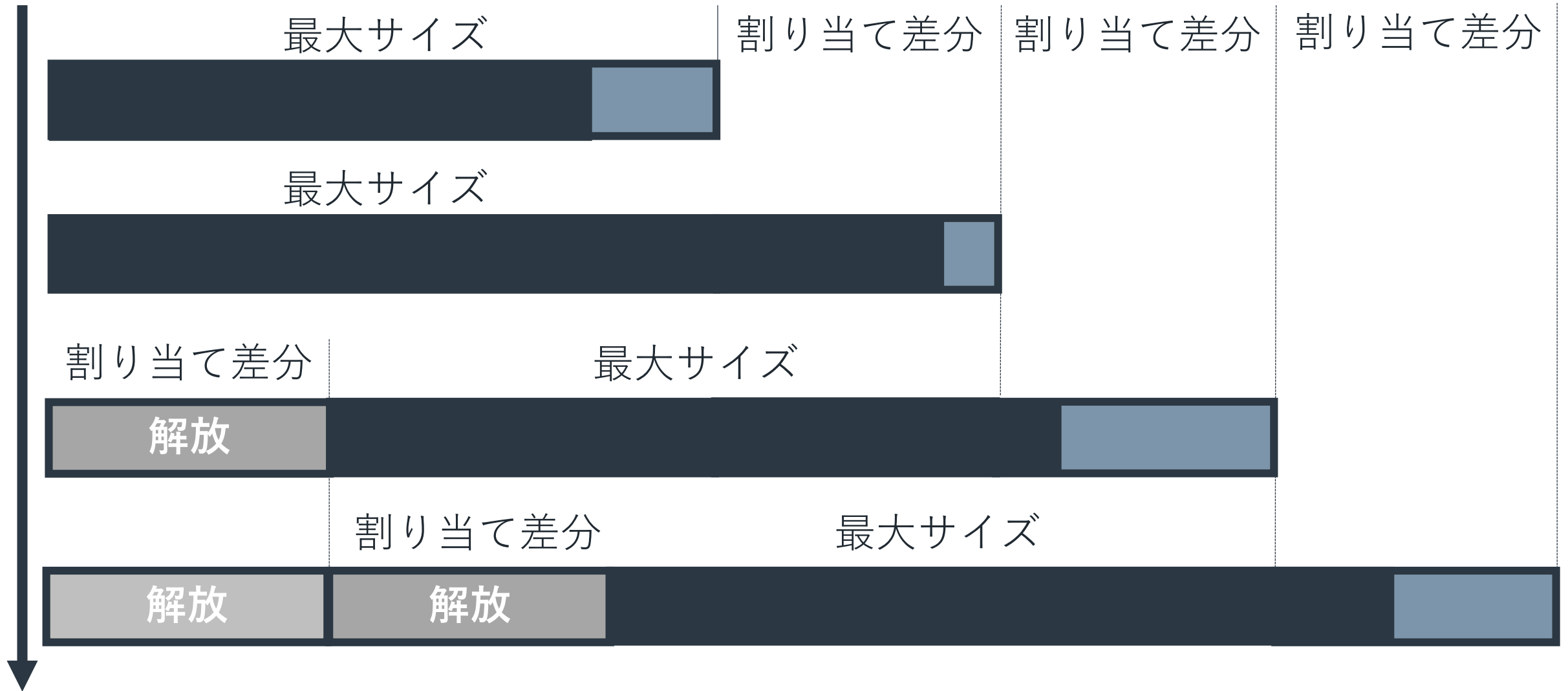
USNジャーナルのサイズ設定デフォルト値

- ▶ 検証環境で確認した最大サイズと割り当て差分の組合せ

バージョン	最大サイズ	割り当て差分
Vista, 7	33,544,432 (32MB)	4,194,304 (4MB)
2008	1,048,576 (1MB)	262,144 (256KB)
8.1, 10, 2012, 2016	33,544,432 (32MB)	8,388,608 (8MB)

- ▶ 条件によっては上記と異なる可能性あり
- ▶ サードパーティアプリによって変更されることもある

USNジャーナルの記録挙動 (1)



USNジャーナルの記録挙動 (2)

- ▶ 最大サイズ32MB+割り当て差分8MB=40MBの場合
- ▶ 40MB (41,943,040) 到達前後のデータ保存状況 (Data run)

到達前

Name	Offset	Value
Allocated size	130	40,894,464
Real size	138	40,470,904
Initialized size	140	40,470,904
▼ \$DATA	158	
▼ Data run	158	
Size	158	0x32
Cluster count	159	2,016
First cluster	15B	1,535,606
▼ Data run	15E	
Size	15E	0x32
Cluster count	15F	128
First cluster	161	1,050,425
▼ Data run	164	
Size	164	0x21
Cluster count	165	16
First cluster	166	1,031,652

割り当て差分のクラスタ数
=8388608/4096=2048個を解放
クラスタ番号1535606から2016個
クラスタ番号1050425から32個

解放したクラスタ数分を
スパーズ (0埋め) として登録

到達後

Name	Offset	Value
Allocated size	130	42,074,112
Real size	138	42,068,008
Initialized size	140	42,068,008
▼ \$DATA	158	
▼ Data run	158	
Size	158	0x02
Cluster count	159	2,048
Sparse	15B	Yes
▼ Data run	15B	
Size	15B	0x31
Cluster count	15C	96
First cluster	15D	1,050,457
▼ Data run	160	
Size	160	0x21
Cluster count	161	16
First cluster	162	1,031,652

USNジャーナルの記録挙動 (3)



- ▶ 長期間の稼働に伴い見た目のファイルサイズは肥大化

Name	Size	Type
\$J	<u>122,449,492</u>	Alternate Data Stream
1d31a7ffd0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
1d31a7ffe0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
1d31a7fff0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
1d31a80000	60 00 00 00 02 00 00 00-6C 9A 01 00 00 00 B8 101.....
1d31a80010	B9 18 01 00 00 00 04 00-00 00 A8 31 1D 00 00 001.....
1d31a80020	16 22 B1 3F 6F B1 D0 01-02 00 00 80 00 00 00 00"±?o±D.....

- ▶ 実サイズは (最大サイズ) ~ (最大サイズ + 割り当て差分) の範囲

現存のUSNジャーナルの抽出

- ▶ ファイルシステムを直接処理するツールが必要
- ▶ スペース領域をスキップすることが望ましい
- ▶ ツール例
 - ▶ CDIR (<https://www.cyberdefense.jp/products/cdir.html>)
 - ▶ ExtractUsnJrnl (<https://github.com/jschicht/ExtractUsnJrnl>)

USNレコードの構造

▶ 変更処理はレコード単位で記録

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00h	レコード長			メジャーバージョン		マイナーバージョン		ファイルID									
10h	親ファイルID								USN								
20h	タイムスタンプ								理由				ソース情報				
30h	セキュリティID			ファイル属性				ファイル名の長さ		ファイル名オフセット		ファイル名…					

USN_RECORD_V2 structure

[https://msdn.microsoft.com/ja-jp/library/windows/desktop/aa365722\(v=vs.85\).aspx](https://msdn.microsoft.com/ja-jp/library/windows/desktop/aa365722(v=vs.85).aspx)

※ USN_RECORD_V3, USN_RECORD_V4も用意されているがデフォルト無効

USNレコード

- ▶ レコード毎にUpdate Sequence Number (USN) を付与
- ▶ USNはメインデータ上のオフセット値

	USN	サイズ
USNレコード1	0	100
USNレコード2	100	80
USNレコード3	180	120
USNレコード4	300	100

・
・
・

記録順序の識別に使える

理由

定義 (USN_REASON)	内容	定義 (USN_REASON)	内容
DATA_OVERWRITE	データ上書き	INDEXABLE_CHANGE	NOT_INDEXED属性の変更
DATA_EXTEND	データ追記	BASIC_INFO_CHANGE	属性/タイムスタンプの変更
DATA_TRUNCATION	データ切り詰め	HARD_LINK_CHANGE	ハードリンクの追加/削除
NAMED_DATA_OVERWRITE	ストリームデータ上書き	COMPRESSION_CHANGE	圧縮状態の変更
NAMED_DATA_EXTEND	ストリームデータ追記	ENCRYPTION_CHANGE	暗号状態の変更
NAMED_DATA_TRUNCATION	ストリームデータ切り詰め	OBJECT_ID_CHANGE	オブジェクトIDの変更
FILE_CREATE	作成	REPARSE_POINT_CHANGE	リパーズポイントの変更
FILE_DELETE	削除	STREAM_CHANGE	ストリームの変更
EA_CHANGE	拡張属性 (EA) の変更	TRANSACTIONED_CHANGE	ストリーム変更 (TxF)
SECURITY_CHANGE	アクセス権の変更	INTEGRITY_CHANGE	INTEGRITY変更 (ReFS)
RENAME_OLD_NAME	ファイル名の変更 (前)	CLOSE	クローズ
RENAME_NEW_NAME	ファイル名の変更 (後)		

ファイル属性

定義 (FILE_ATTRIBUTE)	内容	定義 (FILE_ATTRIBUTE)	内容
ARCHIVE	アーカイブ	NO_SCRUB_DATA	ReFS用
COMPRESSED	圧縮	OFFLINE	オフライン
DEVICE	予約	READONLY	読み取り専用
DIRECTORY	ディレクトリ	RECALL_ON_DATA_ACCESS	仮想ファイル用
ENCRYPTED	暗号化	RECALL_ON_OPEN	仮想ファイル用
HIDDEN	表示しない	REPARSE_POINT	リパーズポイント
INTEGRITY_STREAM	ReFS用	SPARSE_FILE	スパース
NORMAL	通常	SYSTEM	システム
NOT_CONTENT_INDEXED	非インデックス	TEMPORARY	テンポラリ

File Attribute Constants

[https://msdn.microsoft.com/ja-jp/library/windows/desktop/gg258117\(v=vs.85\).aspx](https://msdn.microsoft.com/ja-jp/library/windows/desktop/gg258117(v=vs.85).aspx)

USNレコードのパーズ

- ▶ アタッチされたドライブであれば標準コマンドでパーズ可能
 > *fsutil usn readjournal ドライブ csv > 保存先*



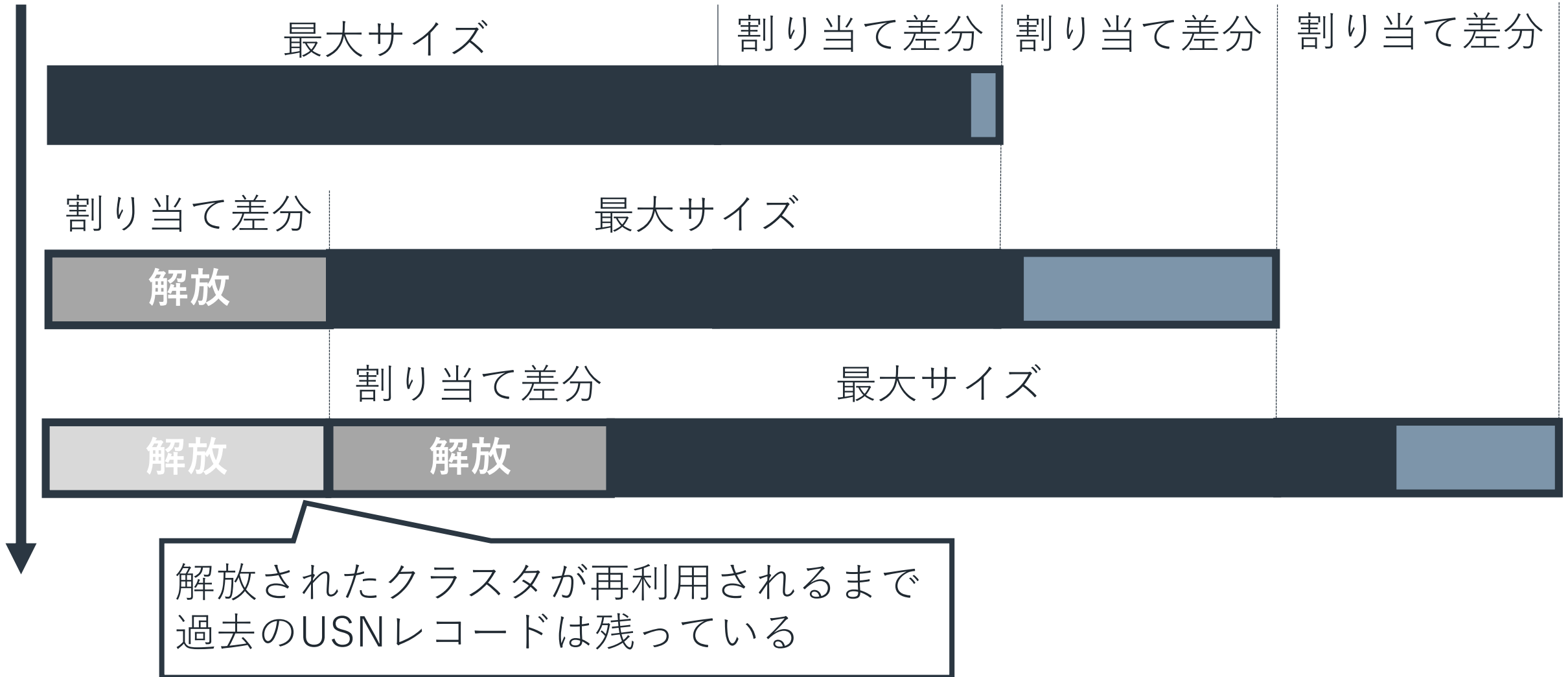
```
管理: Windows PowerShell (x86)
PS C:\> fsutil usn readjournal C: csv > USN.csv
PS C:\>
```



	A	B	C	D	E	F	G	H
1	USN ジャーナル ID	: 0x01d0d4acaec8b8fa						
2	最初の USN	: 0						
3	次の USN	: 18632344						
4	開始 USN	: 0						
5	最小メジャーバージョン:	サポート対象バージョン = 2、要求されたバージョン = 2						
6	最大メジャーバージョン:	サポート対象バージョン = 4、要求されたバージョン = 4						
7								
8	USN	ファイル名	ファイル名の長さ	理由 #	理由	タイム スタンプ	ファイル属性 *	ファイル属性
9		0 System Volume Information		50 0x00000100	ファイルの作成	2015/08/12 12:12:12	0x00000016	表示しない システム ディレクトリ
10		112 System Volume Information		50 0x80000100	ファイルの作成 閉じる	2015/08/12 12:12:12	0x00000016	表示しない システム ディレクトリ

2. USNジャーナルのカービング

USNジャーナルの残存



USNジャーナルの削除挙動 (1)

- ▶ マルウェアによるUSNジャーナルの削除処理の登場
- ▶ NotPetya/Nyetyaは以下のコマンドを呼び出す
> *fsutil usn deletejournal /D %c:*

実行前

Name	Size	Type	Date Modified
\$J	49,772	Alternate Data Stream	2015/08/12 3:12:12
0ffffc0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
0ffffd0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
0ffffe0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
0fffff0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
1000000	58 00 00 00 02 00 00 00 00-6F 6D 00 00 00 00 17 00	X.....om.....	
1000010	59 57 01 00 00 00 01 00 00-00 00 00 01 00 00 00	YW.....	
1000020	72 72 F4 E3 59 E6 D1 01-02 C1 00 80 00 00 00 00 00	rrôãYæñ.Á.....	
1000030	00 00 00 00 20 28 00 00 00-1C 00 3C 00 52 00 65 00 (<R.e.....	
1000040	70 00 6F 00 72 00 74 00 00-2E 00 77 00 65 00 72 00	p.o.r.t..w.e.r.....	
1000050	2E 00 74 00 6D 00 70 00 00-50 00 00 00 02 00 00 00	.t.m.p.P.....	
1000060	70 6D 00 00 00 00 16 00 00-59 57 01 00 00 00 01 00	pm.....YW.....	
1000070	58 00 00 01 00 00 00 00 00-72 72 F4 E3 59 E6 D1 01	X.....rrôãYæñ.....	
1000080	00 02 00 80 00 00 00 00 00-00 00 00 00 20 28 00 00 (<.....	
1000090	14 00 3C 00 52 00 65 00 00-70 00 6F 00 72 00 74 00	<R.e.p.o.r.t.....	
10000a0	2E 00 77 00 65 00 72 00 00-58 00 00 00 02 00 00 00	.w.e.r.X.....	

Cursor pos = 0x1000000; dus = 0x19edca; log sec = 0xcf6e50



実行後

Name	Size	Type	Date Modified
\$J	8	Alternate Data Stream	2018/01/10 6:34:08
0000	50 00 00 00 02 00 00 00 00-6F 55 01 00 00 00 01 00	P.....oU.....	
0010	6A 55 01 00 00 00 01 00 00-00 00 EA 00 00 00 00 00	jU.....ê.....	
0020	9A 6B 1F F9 9A 94 D1 01-01 00 00 80 00 00 00 00 00	.k.ù.ñ.....	
0030	00 00 00 00 20 00 00 00 00-0E 00 3C 00 55 00 53 00<U.S.....	
0040	53 00 2E 00 63 00 68 00 00-6B 00 00 00 00 00 00 00	S..c.h.k.....	
0050	50 00 00 00 02 00 00 00 00-70 55 01 00 00 00 01 00	P.....pU.....	
0060			
0070			
0080			
0090			
00a0			
00b0			
00c0			
00d0			
00e0	6F 00 72 00 65 00 2E 00-76 00 6F 00 6C 00 00 00 00	o.r.e..v.o.l.....	

Cursor pos = 0; dus = 1411853; log sec = 11294824

USNジャーナルID更新
ファイルサイズリセット

USNジャーナルの削除挙動 (2)

実行前

Name	Size	Type	Date Modified
\$J	49,772	Alternate Data Stream	2015/08/12 3:12:12
0ffffc0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
0ffffd0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
0ffffe0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
0fffff0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
1000000	58 00 00 00 02 00 00 00-6F 6D 00 00 00 00 17 00	X.....om.....	
1000010	59 57 01 00 00 00 01 00-00 00 00 01 00 00 00 00	YW.....	
1000020	72 72 F4 E3 59 E6 D1 01-02 C1 00 80 00 00 00 00 00	rrôãYæÑ-Á.....	
1000030	00 00 00 00 20 28 00 00-1C 00 3C 00 52 00 65 00 00(<R-e	
1000040	70 00 6F 00 72 00 74 00-2E 00 77 00 65 00 72 00 00	p-o-r-t..w-e-r	
1000050	2E 00 74 00 6D 00 70 00-50 00 00 00 02 00 00 00 00	..t-m-p.P.....	
1000060	70 6D 00 00 00 00 16 00-59 57 01 00 00 00 01 00 00	pm.....YW.....	
1000070	58 00 00 01 00 00 00 00-72 72 F4 E3 59 E6 D1 01 00	X.....rrôãYæÑ-	
1000080	00 02 00 80 00 00 00 00-00 00 00 00 20 28 00 00 00(<	
1000090	14 00 3C 00 52 00 65 00-70 00 6F 00 72 00 74 00 00	..<R-e-p-o-r-t	
10000a0	2E 00 77 00 65 00 72 00-58 00 00 00 02 00 00 00 00	..w-e-r-X.....	

Sel start = 0x1000000, len = 0x0058, dus = 0x19edca, log sec = 0xcfbes0

実行後

Evidence Tree			
NONAME [NTFS]			
[root]			
19edc9fc0	02 C1 00 00 00 00 00 00-00 00 00 00 28 00 00 00	Á.....(
19edc9fd0	1C 00 3C 00 52 00 65 00-70 00 6F 00 72 00 74 00 00	..<R-e-p-o-r-t	
19edc9fe0	2E 00 77 00 65 00 72 00-2E 00 74 00 6D 00 70 00 00	..w-e-r..t-m-p	
19edc9ff0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00	
19edca000	58 00 00 00 02 00 00 00-6F 6D 00 00 00 00 17 00	X.....om.....	
19edca010	59 57 01 00 00 00 01 00-00 00 00 01 00 00 00 00 00	YW.....	
19edca020	72 72 F4 E3 59 E6 D1 01-02 C1 00 80 00 00 00 00 00 00	rrôãYæÑ-Á.....	
19edca030	00 00 00 00 20 28 00 00-1C 00 3C 00 52 00 65 00 00 00(<R-e	
19edca040	70 00 6F 00 72 00 74 00-2E 00 77 00 65 00 72 00 00 00	p-o-r-t..w-e-r	
19edca050	2E 00 74 00 6D 00 70 00-50 00 00 00 02 00 00 00 00 00	..t-m-p.P.....	
19edca060	70 6D 00 00 00 00 16 00-59 57 01 00 00 00 01 00 00 00	pm.....YW.....	
19edca070	58 00 00 01 00 00 00 00-72 72 F4 E3 59 E6 D1 01 00	X.....rrôãYæÑ-	

Sel start = 0x19edca000, len = 0x0058, dus = 0x19edca, log sec = 0xcfbes0

元のクラスタは解放されるだけ

ファイルサイズ

削除前



ファイルサイズ

削除後



USNレコードが残る場所

- ▶ USNレコードは\$UsnJrnl以外にも残る
 - ▶ メモリ
 - ▶ ファイルシステムジャーナル (\$LogFile)
 - ▶ スワップファイル (pagefile.sys)
 - ▶ ハイバネーションファイル (hiberfil.sys)
 - ▶ VSSスナップショット
 - ▶ 未割当領域
- ▶ 過去のUSNレコードは潜在的に多数存在

USNレコードのカービング

- ▶ 未割当領域だけでなくファイルシステム全域を探索した方がよい
- ▶ ノイズを減らすためにうまくシグネチャを作る必要がある



カービングシグネチャ作成のための調査

- ▶ サンプルングにより得られた正常なUSNジャーナルの特徴
 - ▶ レコード長：必ず8の倍数（64～464）
 - ▶ ファイル名長：必ず偶数（2～356）
 - ▶ ソース情報：大半は0
 - ▶ セキュリティID：大半は0
- ▶ NTFSの仕様
 - ▶ ファイル名の文字コードはUTF16-LE
 - ▶ ファイル名の最大長は255文字（510バイト）

USNレコードのカービングシグネチャ

- ▶ サンプルング結果と仕様を踏まえてシグネチャを決定

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00h	8の倍数かつ 60~600			02 00		00 00		ファイルID								
10h	親ファイルID								USN							
20h	タイムスタンプ								理由				ソース情報			
30h	セキュリティID			ファイル属性				2~512		0x3C		ファイル名...				

追加シグネチャ

Bulk Extractor

- ▶ https://github.com/simsong/bulk_extractor
- ▶ 与えられたデータから様々なパターンを抽出/パースするツール
 - ▶ メールアドレス
 - ▶ URL
 - ▶ EXIF
 - ▶ ZIP
 - ▶ ...
- ▶ 入力データはファイル、ドライブ、ディスク、E0イメージなど
- ▶ **抽出用シグネチャをスキャナとして実装、拡張することが可能**

Bulk Extractor with Record Carving

- ▶ https://www.kazamiya.net/bulk_extractor-rec
- ▶ レコードカービング用のスキャナを追加
- ▶ USNジャーナルレコードのカービングスキャナは**ntfsusn**
- ▶ hiberfileスキャナとの併用によりハイバネーションファイル
hiberfil.sys内も適切に取り扱える
(ただし、Windows 8以上の形式には非対応)

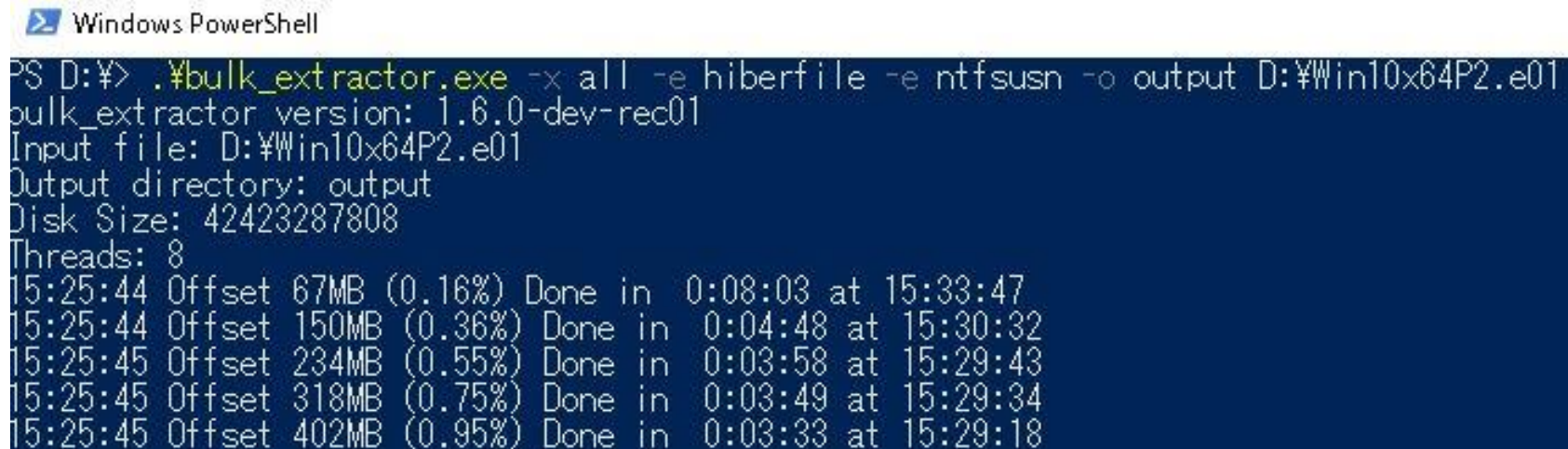
Bulk Extractor with Record Carvingの使い方 (CMD)

▶ hiberfil.sys内を含めUSNレコードをカービングする例

> *bulk_extractor -x all -e hiberfile -ntfsusn -o 出力フォルダ* イメージファイル

> *bulk_extractor -x all -e hiberfile -ntfsusn -o 出力フォルダ* ¥¥.¥C:

> *bulk_extractor -x all -e hiberfile -ntfsusn -o 出力フォルダ* ¥¥.¥PhysicalDrive0

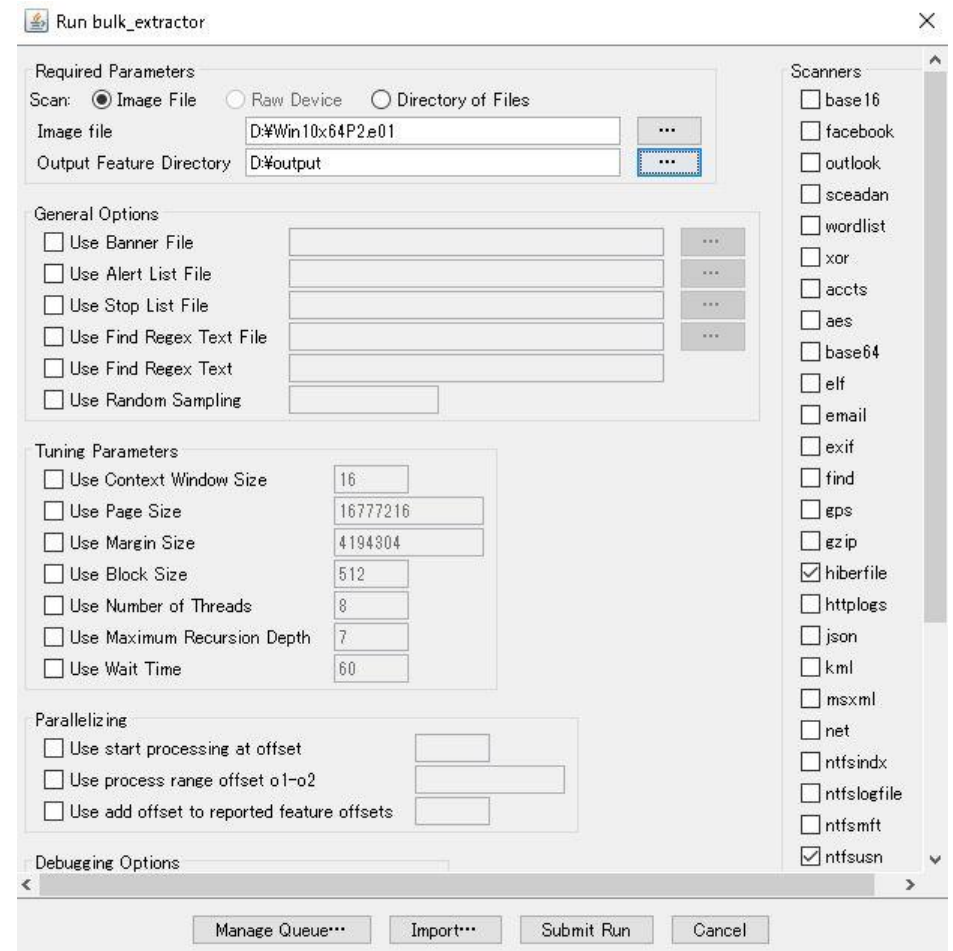


```
Windows PowerShell
PS D:¥> .¥bulk_extractor.exe -x all -e hiberfile -e ntfsusn -o output D:¥Win10x64P2.e01
bulk_extractor version: 1.6.0-dev-rec01
Input file: D:¥Win10x64P2.e01
Output directory: output
Disk Size: 42423287808
Threads: 8
15:25:44 Offset 67MB (0.16%) Done in 0:08:03 at 15:33:47
15:25:44 Offset 150MB (0.36%) Done in 0:04:48 at 15:30:32
15:25:45 Offset 234MB (0.55%) Done in 0:03:58 at 15:29:43
15:25:45 Offset 318MB (0.75%) Done in 0:03:49 at 15:29:34
15:25:45 Offset 402MB (0.95%) Done in 0:03:33 at 15:29:18
```

Bulk Extractor with Record Carvingの使い方 (GUI)

▶ hiberfil.sys内を含めUSNレコードをカービングする例

1. Tools > Run bulk_extractor
2. Image Fileを指定
3. Output Feature Directoryを指定
4. Scannersのhiberfileとntfsusnのみチェック
5. Submit Run



Bulk Extractor with Record Carvingの処理速度

- ▶ マルチスレッドのためCPUよりI/Oがボトルネックになりやすい
- ▶ SSDやRAIDディスクを利用すると高速

```
Windows PowerShell
16:29:48 Offset 42177MB (99.42%) Done in 0:00:00 at 16:29:48
16:29:48 Offset 42261MB (99.62%) Done in 0:00:00 at 16:29:48
16:29:48 Offset 42345MB (99.82%) Done in 0:00:00 at 16:29:48
All data are read; waiting for threads to finish...
Time elapsed waiting for 4 threads to finish:
  (timeout in 60 min.)
All Threads Finished!
Producer time spent waiting: 0 sec.
Average consumer time spent waiting: 100.538 sec.
*****
** bulk_extractor is probably I/O bound. **
**   Run with a faster drive           **
**   to get better performance.       **
*****
MD5 of Disk Image: 57d73b4684bb69c8e0e507d6b0a97362
Phase 2. Shutting down scanners
Phase 3. Creating Histograms
Elapsed time: 142.415 sec.
Total MB processed: 42423
Overall performance: 297.885 MBytes/sec (37.2356 MBytes/sec/thread)
```

入力データがSSD上にある場合の例
40GBのディスクイメージ処理に142秒
300MB/sec ⇒ 1TBを1時間で処理可能

評価用サンプルデータ

- ▶ 一定期間以上利用されていたコンピュータを評価データに利用

バージョン	最大サイズ	割り当て差分	台数	ディスクサイズ 平均
Vista	32MB	4MB	5	220GB
7	32MB	4MB	77	260GB
2008	1MB	256KB	12	300GB
2008	512MB	1MB	6	430GB
8.1	32MB	8MB	2	370GB
2012	32MB	8MB	2	220GB
10	32MB	8MB	5	360GB

Bulk Extractor with Record Carving 評価結果

▶ 既存データとカービングデータの平均サイズ比較

バージョン	最大サイズ	割り当て差分	既存データ	カービングデータ
Vista	32MB	4MB	33.8MB	114.9MB
7	32MB	4MB	33.8MB	2191.9MB
2008	1MB	256KB	1.13MB	241.2MB
2008	512MB	1MB	512.4MB	867.1MB
8.1	32MB	8MB	35.5MB	293.2MB
2012	32MB	8MB	35.9MB	77.7MB
10	32MB	8MB	36.9MB	1357.6MB

3. USNジャーナルの解析

既存のUSNジャーナル解析ツール

- ▶ フリー/商用を含め複数の解析ツールが存在する
- ▶ 基本はUSNレコードのパーズ
 - ▶ 人間にとって読みやすい表記に変換
 - ▶ 表形式 (CSV/TSV) での出力が一般的
- ▶ ツールによっては独自の機能
 - ▶ 別データ (\$MFT, \$LogFile) との連携
 - ▶ 時間フィルタ/ファイル名等の検索
 - ▶ グラフ描画

既存ツールの課題

- ▶ 実際の調査で…
 - ▶ 単純なパース結果 ⇒ 量が多すぎて事象追跡しにくい
 - ▶ 時間/ファイル名等のフィルタ ⇒ 関連情報が欠落し事象追跡しにくい
- ▶ 課題解決に加えて実調査で得たノウハウの機能化を試みる
 - ▶ うまく量を減らして流れを追いやすくする
 - ▶ 特徴的な情報/事象（インジケータ）に気づきやすくする

USN Analytics

▶ https://www.kazamiya.net/usn_analytics/

▶ マルチプラットフォーム (C++11準拠)

▶ オープンソース

▶ コマンドライン

> *usn_analytics [-ru] -o 出力フォルダ 入力ファイル*

-r 全レコードのパーズ結果のみ出力

-u タイムスタンプをUTCで表示 (デフォルトはPCのタイムゾーン設定)

入力ファイルはUSNレコードで構成される任意のファイル

全USNレコードのパーズ

- ▶ USNレコードの構造に従って全フィールドをパーズ (TSV)

Offset	RecLength	Major	Minor	FileID	FileParentID	ParentUsn	TimeStamp(+09:00)	Reason	So	Se	FileAttr	FileN	FileN	FileName
0	88	2	0	34235	1	6996	1 0 2018/01/10 13:11:49.882448	CLOSE(80000000)	0	0	ARCHIVE	22	60	autochk.exe
88	112	2	0	106976	2	5	5 88 2018/01/10 13:11:49.991746	CREATE(00000100)	0	0	HIDDEN SYSTEM	50	60	System Volume Information\
200	112	2	0	106976	2	5	5 200 2018/01/10 13:11:49.991746	CREATE CLOSE(80000100)	0	0	HIDDEN SYSTEM	50	60	System Volume Information\
312	128	2	0	106977	2	106976	2 312 2018/01/10 13:11:49.991746	CREATE(00000100)	0	0	HIDDEN SYSTEM	62	60	MountPointManagerRemoteDatabase
440	128	2	0	106977	2	106976	2 440 2018/01/10 13:11:50.054631	CREATE CLOSE(80000100)	0	0	HIDDEN SYSTEM	62	60	MountPointManagerRemoteDatabase
568	88	2	0	106978	2	4635	1 568 2018/01/10 13:11:50.140354	CREATE(00000100)	0	0	ARCHIVE	26	60	EtwRTUBPM.etl
656	88	2	0	106978	2	4635	1 656 2018/01/10 13:11:50.140354	CREATE EXTEND(00000102)	0	0	ARCHIVE	26	60	EtwRTUBPM.etl
744	72	2	0	85496	1	3741	1 744 2018/01/10 13:11:50.280948	EXTEND(00000002)	0	0	ARCHIVE	12	60	SYSTEM
816	112	2	0	106979	2	4635	1 816 2018/01/10 13:11:50.515345	CREATE(00000100)	0	0	ARCHIVE	52	60	EtwRTEventlog-Security.etl
928	112	2	0	106979	2	4635	1 928 2018/01/10 13:11:50.515345	CREATE EXTEND(00000102)	0	0	ARCHIVE	52	60	EtwRTEventlog-Security.etl

- ▶ USNの昇順 (= 記録順を示す) で書き出し
- ▶ タイムスタンプは100ナノ秒単位で書き出し
- ▶ 属性からフォルダとわかる場合はファイル名末尾に「¥」 追記

全USNレコードのパーズ結果

- ▶ 既存データとカービングデータの平均レコード数比較

バージョン	既存データ	カービングデータ
Vista	324,000	1,013,000
7	362,000	14,641,000
2008	13,000	2,534,000
2008	6,244,000	9,415,000
8.1	375,000	1,443,000
2012	396,000	843,000
10	338,000	8,280,000

- ▶ 期間は対象に大きく依存する

解析用のフィールド整理

- ▶ 調査上活用価値の高いフィールドに厳選する
 - ▶ オフセット
 - ▶ USN
 - ▶ タイムスタンプ
 - ▶ ファイル名
 - ▶ 理由
 - ▶ ファイル属性
 - ▶ ファイルID
 - ▶ 親ファイルID

USNレコードの連結

- ▶ 同一ファイルに対する一連の処理が複数記録されるパターン

Offset	RecLength	Major	Minor	FileID	File\$	ParentID	Parent	Usn	TimeStamp(+09:00)	Reason	So	Se	FileAttr	FileN	FileN	FileName
69896	96	2	0	107020	2	1885	1	69896	2018/01/10 13:12:33.405843	EXTEND(00000002)	0	0	ARCHIVE	32	60	setupapi.dev.log
69992	96	2	0	107020	2	1885	1	69992	2018/01/10 13:12:33.405843	EXTEND TRUNCATION(00000000)	0	0	ARCHIVE	32	60	setupapi.dev.log
70088	96	2	0	107020	2	1885	1	70088	2018/01/10 13:12:33.405843	EXTEND TRUNCATION CLOSE(6	0	0	ARCHIVE	32	60	setupapi.dev.log
70184	96	2	0	107020	2	1885	1	70184	2018/01/10 13:12:33.405843	EXTEND(00000002)	0	0	ARCHIVE	32	60	setupapi.dev.log
70280									12:33.405843	EXTEND TRUNCATION(00000000)	0	0	ARCHIVE	32	60	setupapi.dev.log
70376									12:33.405843	EXTEND TRUNCATION CLOSE(6	0	0	ARCHIVE	32	60	setupapi.dev.log
70472									12:33.423949	EXTEND(00000002)	0	0	ARCHIVE	32	60	setupapi.dev.log
70568									12:33.423949	EXTEND TRUNCATION(00000000)	0	0	ARCHIVE	32	60	setupapi.dev.log
70664									12:33.423949	EXTEND TRUNCATION CLOSE(6	0	0	ARCHIVE	32	60	setupapi.dev.log
70760									12:33.423949	EXTEND(00000002)	0	0	ARCHIVE	32	60	setupapi.dev.log
70856									12:33.423949	EXTEND TRUNCATION(00000000)	0	0	ARCHIVE	32	60	setupapi.dev.log
70952	96	2	0	107020	2	1885	1	70952	2018/01/10 13:12:33.423949	EXTEND TRUNCATION CLOSE(6	0	0	ARCHIVE	32	60	setupapi.dev.log
71048	96	2	0	107020	2	1885	1	71048	2018/01/10 13:12:33.423949	EXTEND(00000002)	0	0	ARCHIVE	32	60	setupapi.dev.log
71144	96	2	0	107020	2	1885	1	71144	2018/01/10 13:12:33.423949	EXTEND TRUNCATION(00000000)	0	0	ARCHIVE	32	60	setupapi.dev.log
71240	96	2	0	107020	2	1885	1	71240	2018/01/10 13:12:33.423949	EXTEND TRUNCATION CLOSE(6	0	0	ARCHIVE	32	60	setupapi.dev.log

setupapi.dev.logに対する
追記処理

- ▶ 情報量を極力落とさずに扱うためのフィールドを追加
 - ▶ レコード数
 - ▶ 一連の処理の所要時間

ファイルパスの構築

- ▶ 親ファイルIDのレコードが残っていればパスの構築が可能

Offset	RecLength	Major	Minor	FileID	FileParentID	ParentUsn	TimeStamp(+09:00)	Reason	So	Se	FileAttr	FileN	FileN	FileName
637592	80	2	0	76189	2	4815	1 637592 2018/01/10 13:13:40.849858	CREATE(00000100)	0	0	DIRECTORY	14	60	Panther\
637672	80	2	0	76189	2	4815	1 637672 2018/01/10 13:13:40.849858	CREATE CLOSE(80000100)	0	0	DIRECTORY	14	60	Panther\
637752	64	2	0	76190	2	76189	2 637752 2018/01/10 13:13:40.849858	CREATE(00000100)	0	0	DIRECTORY	4	60	IE\
637816	64	2	0	76190	2	76189	2 637816 2018/01/10 13:13:40.849858	CREATE CLOSE(80000100)	0	0	DIRECTORY	4	60	IE\
637880	88	2	0	76192	2	76190	2 637880 2018/01/10 13:13:40.849858	CREATE(00000100)	0	0	ARCHIVE	24	60	setupact.log
637968	88	2	0	76193	2	76190	2 637968 2018/01/10 13:13:40.849858	CREATE(00000100)	0	0	ARCHIVE	24	60	setuperr.log

setupact.logとsetuperr.logのパスはPanther¥IE¥

- ▶ 全レコードを走査してパス情報のリストを作成
- ▶ 候補が複数ある場合、USN値を考慮して妥当なパス情報を選択
- ▶ ファイルパス用のフィールドを追加

リネーム/移動の追跡

- ▶ 理由がOLDNAMEとNEWNAMEのレコード
 - ▶ ファイル名の変更有無でリネーム (RENAME) か移動 (MOVE) か判明
- ▶ リネーム時
 - ▶ ファイル名に「変更前ファイル -> 変更後ファイル」と表示
- ▶ 移動時
 - ▶ ファイル名に「(変更前フォルダ¥ -> 変更後フォルダ¥)」と表示

Offset	RecLength	Major	Minor	FileID	File	ParentID	Parent	Usn	TimeStamp(+09:00)	Reason	So	Se	FileAttr	FileN	FileN	FileName
645872	88	2	0	107101	2	4799	1	645872	2018/01/10 13:13:47.811402	CREATE EXTEND(00000102)	0	0	HIDDEN ARCHIVE	24	60	data.dat.tmp
645960	88	2	0	107101	2	4799	1	645960	2018/01/10 13:13:47.811402	CREATE EXTEND CLOSE(800000)	0	0	HIDDEN ARCHIVE	24	60	data.dat.tmp
646048	88	2	0	107101	2	4799	1	646048	2018/01/10 13:13:47.811402	OLDNAME(00001000)	0	0	HIDDEN ARCHIVE	24	60	data.dat.tmp
646136	88	2	0	107101	2	4799	1	646136	2018/01/10 13:13:47.811402	NEWNAME(00002000)	0	0	HIDDEN ARCHIVE	24	60	data.dat.bak
646224	88	2	0	107101	2	4799	1	646224	2018/01/10 13:13:47.811402	NEWNAME CLOSE(80002000)	0	0	HIDDEN ARCHIVE	24	60	data.dat.bak
646312	80	2	0	88663	1	4799	1	646312	2018/01/10 13:13:47.855263	DELETE CLOSE(80000200)	0	0	HIDDEN ARCHIVE	16	60	data.dat

解析処理結果

レコードの連結

所要時間

Offset	Usn	Record	TimeStamp(+09:00)	TimeTaken	FileName	Reason	FileAttr	FileID	ParentID	Path
111176	111176	3	2018/01/10 13:12:50.122384	0	ServerManager	EXTEND TRUNCATION CLOSE	ARCHIVE	88731	4872	
111440	111440	1029	2018/01/10 13:12:50.174435	2.822831	setupapi.dev.log	EXTEND OVERWRITE TRUNCATION CLOSE	ARCHIVE	107020	1885	
111728	111728	3	2018/01/10 13:12:50.345288	0	XblGameSaveTaskLogon	EXTEND TRUNCATION CLOSE	ARCHIVE	88746	4902	

パスの構築

Offset	Usn	Record	TimeStamp(+09:00)	TimeTaken	FileName	Reason	FileAttr	FileID	ParentID	Path
637592	637592	2	2018/01/10 13:13:40.849858	0	Panther\	CREATE CLOSE	DIRECTORY	76189	4815	
637752	637752	2	2018/01/10 13:13:40.849858	0	IE\	CREATE CLOSE	DIRECTORY	76190	76189	Panther\
637880	637880	3	2018/01/10 13:13:40.849858	0.00404	setupact.log	CREATE EXTEND CLOSE	ARCHIVE	76192	76190	Panther\IE\

リネームの追跡

Offset	Usn	Record	TimeStamp(+09:00)	TimeTaken	FileName	Reason	FileAttr	FileID	ParentID	Path
646048	646048	3	2018/01/10 13:13:47.811402	0	data.dat.tmp -> data.dat.bak	RENAME	HIDDEN ARCHIVE	107101	4799	
646312	646312	1	2018/01/10 13:13:47.855263	0	data.dat	DELETE CLOSE	HIDDEN ARCHIVE	88663	4799	
646392	646392	3	2018/01/10 13:13:48.139388	0.015641	data.dat.bak -> data.dat	RENAME	HIDDEN ARCHIVE	107101	4799	

インジケータの発見、抽出

- ▶ システム/ユーザの挙動を示すアーティファクトの活用
 - ▶ プログラム実行履歴
 - ▶ ファイル参照履歴

- ▶ 特徴的なファイル名/拡張子の活用

プログラム実行履歴

- ▶ プリフェッチ (pf) ファイルの作成/更新処理から判断
- ▶ メリット
 - ▶ pfファイルが現存しなくてもUSNレコードは残っている可能性がある
- ▶ デメリット
 - ▶ プリフェッチ無効の環境 (Windows Server、仮想環境) では使えない

Usn	TimeStamp(+09:00)	ExeName	ExeCount	FileName	Reason
447491984	2017/08/16 11:08:59.639800	whoami.exe	1	WHOAMI.EXE-B8288E39.pf	CREATE EXTEND CLOSE
447492304	2017/08/16 11:08:59.655400	cmd.exe	6	CMD.EXE-4A81B364.pf	EXTEND TRUNC CLOSE
447494264	2017/08/16 11:10:15.555400	cmd.exe	7	CMD.EXE-4A81B364.pf	EXTEND TRUNC CLOSE
447495960	2017/08/16 11:11:27.818600	reg.exe	1	REG.EXE-E7E8BD26.pf	CREATE EXTEND CLOSE
447498672	2017/08/16 11:15:54.656600	ipconfig.exe	2	IPCONFIG.EXE-912F3D5B.pf	EXTEND TRUNC CLOSE
447501440	2017/08/16 11:16:40.629800	systeminfo.exe	1	SYSTEMINFO.EXE-1905EE9D.pf	CREATE EXTEND CLOSE
447501776	2017/08/16 11:16:46.869800	wmiprvse.exe	10	WMIPRVSE.EXE-1628051C.pf	EXTEND TRUNC CLOSE
447502112	2017/08/16 11:16:46.979000	trustedinstaller.exe	15	TRUSTEDINSTALLER.EXE-3CC531E5.pf	EXTEND TRUNC CLOSE
447502984	2017/08/16 11:18:07.053800	tasklist.exe	1	TASKLIST.EXE-C6CEE193.pf	CREATE EXTEND CLOSE
447503632	2017/08/16 11:19:40.653800	net.exe	1	NET.EXE-DF44F913.pf	CREATE EXTEND CLOSE

ファイル参照履歴

- ▶ ユーザによるファイル参照時の処理から判断
 - ▶ ショートカット (lnk) ファイルの作成
 - ▶ 該当ファイルにオブジェクトID付与
- ▶ メリット
 - ▶ エントリポイントの特定に使える可能性がある
- ▶ デメリット
 - ▶ 量が多くなる傾向にありチェックに時間がかかる

Usn	TimeStamp(+09:00)	Path	FileName	Reason
447387376	2017/08/16 10:24:50.422400	Temp1_eTicket_234681230.zip\eTicket_234681230\	eTicket(1).lnk	CREATE EXTEND INFO INDEX STREAM NAMED_E CLOSE
447389224	2017/08/16 10:24:52.622000		powershell_ise.exe	OBJECTID CLOSE
447389416	2017/08/16 10:24:52.622000		hh.exe	OBJECTID CLOSE
447413272	2017/08/16 10:25:19.781000		eTicket_234681230.lnk	CREATE EXTEND CLOSE
447413736	2017/08/16 10:25:19.796600		ダウンロード.lnk	CREATE EXTEND CLOSE
447414872	2017/08/16 10:25:25.927400	eTicket_234681230\	eTicket(1).lnk	CREATE EXTEND INFO STREAM NAMED_E CLOSE

特徴的なファイル拡張子

▶ 特定の拡張子に着目して結果を集約することにより
インジケータの発見につながる可能性がある

- ▶ ジョブ (job)
- ▶ PE形式ファイル (scr)
- ▶ スクリプト (bat/vbe/ps1)

```
usn_analytics_report.txt - XE帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
[job] 1 files (name, count)
At1.job, 3
[tkc] 0 files (name, count)
[ps1] 14 files (name, count)
CL_Utility.ps1, 2
RS_AdminDiagnosticHistory.ps1, 2
```

Usn	Record	TimeStamp(+09:00)	TimeTake	FileName	Reason	FileAttr	File	Parent	Path
660748608	4	2017/08/16 14:53:42.116297	0.0156	A64.exe	CREATE EXTEND OVERWRITE INFO	ARCHIVE	17214	58	
660748928	1	2017/08/16 14:53:55.688297	0	A64.exe	CREATE EXTEND OVERWRITE INFO CL	ARCHIVE	17214	58	
660749008	6	2017/08/16 14:55:00.662297	0.1248	At1.job	CREATE EXTEND OVERWRITE CLOSE	ARCHIVE	17217	3939	
660749248	5	2017/08/16 14:55:00.709097	0.0312	d42cc0c3858a58db2db376	CREATE EXTEND OVERWRITE CLOSE	SYSTEM ARCHIVE N	17219	155348	
660750248	2	2017/08/16 14:55:00.771497	0	At1	CREATE EXTEND SECURITY TRANSA	ARCHIVE	71368	3513	
660750648	2	2017/08/16 14:58:00.109097	0	At1.job	OVERWRITE CLOSE	ARCHIVE	17217	3939	
660750808	3	2017/08/16 14:58:01.575497	0	RecentFileCache.bcf	EXTEND OVERWRITE CLOSE	SYSTEM ARCHIVE	16718	634	
660751120	3	2017/08/16 14:58:01.996697	0.1404	TMP58F501B6E02209E5	CREATE EXTEND DELETE CLOSE	ARCHIVE TEMP	71533	3940	
660751432	3	2017/08/16 14:58:10.139897	0	TASKENG.EXE-48D4E289	EXTEND TRUNC CLOSE	ARCHIVE NOINDEX	70121	61538	Prefetch\

今回のアウトプットと今後の展開

- ▶ アウトプットとして
 - ▶ USNジャーナルに対する理解
 - ▶ Bulk Extractor with Record Carvingによるカービング
 - ▶ USN Analyticsによる解析（≠パース）
- ▶ 今後の展開
 - ▶ ノウハウの機能化の拡充
 - ▶ グラフ、ヒストグラム等を活用した可視化
 - ▶ 機械学習の活用