

削除済みVSSスナップ ショットの復元

株式会社インターネットイニシアティブ
小林 稔

Who am I

- 小林 稔
- 株式会社インターネットイニシアティブ
セキュリティ本部 セキュリティ情報統括室に所属
 - 社内外のインシデントレスポンスに従事
- 外部活動
 - 2015年8月～2018年1月 社会保障審議会年金事業管理部会運営担当参与
 - *Mauritius 2016 FIRST Technical Colloquium* スピーカーおよびトレーニング講師
 - 2017年セキュリティキャンプ全国大会講師
- Twitter : @unkn0wnbit

Agenda

- 本発表の概要とモチベーション、目標
- VSSスナップショットのデータ構造
- VSSスナップショットの仕組み
- 主なスナップショットパーサ
- 削除済みVSSスナップショットアクセス手法の検討
- 作成したツールの概要と復元テスト
- Demo
- まとめ

概要

- 本発表はVolume Shadow Copy Service (VSS)に関する調査研究である。
- VSSはWindowsに標準搭載されているバックアップ関連機能で、NTFSボリュームのVSSスナップショット（以下、スナップショット）を作成することができる。
- スナップショットを参照すると、スナップショット作成時のデータ、つまり過去のデータにアクセスできる。そのため、攻撃の痕跡を発見できる場合があり、インシデントレスポンスにおいて重要な役割を果たす。しかし、容量の上限によって自然に古いものが消える、もしくは攻撃者やマルウェアによって削除されてしまう場合がある。
- 本発表では、削除されたスナップショットにアクセスする手法の検討とそれを基に実装したツールの実験結果を解説する。

インシデントレスポンス時のVSSスナップショットの活用方法

- 時間の経過や攻撃者により削除されてしまったマルウェアや攻撃者の痕跡を復元し、解析することで、より深く解析を行う
 - 攻撃者が使用したツール
 - 攻撃者が一時的に作成したアーカイブファイル
 - 現在のイベントログから削除されてしまったログ
- ランサムウェアに暗号化されてしまったファイルの復元

モチベーション (1)

- スナップショットは重要なアーティファクトであるが、削除されたスナップショットにWindowsからアクセスする方法は存在しない。
- X-waysなど一部のツールは特定の条件下で削除済みのスナップショットにアクセスできることをCDI山崎氏が確認している。
 - <http://www.kazamiya.net/DeletedSC>
- このことから、VSS関連のファイルを復元できれば、他のツールでもデータを復元できる可能性がある。これが第1のモチベーションである。

モチベーション (2)

- 削除済みのスナップショットの一部にアクセスする手段の一つとして、カービングが存在する。しかし、この手法には致命的な欠陥が存在する。
- カービングはシグネチャを利用して、連続した領域をファイルやレコードとして復元する。しかし、スナップショットは16KBのデータブロック単位でバックアップが行われる。そのため、カービングを行っても最大で16KBのファイルやレコードしかカービングすることができない。また、ファイル生成日時などのメタ情報も復元できない。
- スナップショットのデータにアクセスする際には、現在のNTFSボリュームとスナップショット内のバックアップデータを正しく組み合わせる必要がある。したがって、NTFSボリュームをパースするカービングツールを使用しても対象ファイル全体をカービングすることはできない。
- このように、削除されたスナップショットのアクセスには専用のツールが必要となるが、誰でも自由に使えるソフトウェアがなかった。これが第2のモチベーションである。

目標

- 以下のような状況で削除されたスナップショットからファイルを復元するツールを作成する。
 - スナップショットの容量オーバーで自動的に削除されたスナップショット
 - 攻撃者やランサムウェア等に削除されたスナップショット

VSSスナップショット のデータ構造

VSSスナップショットのファイル

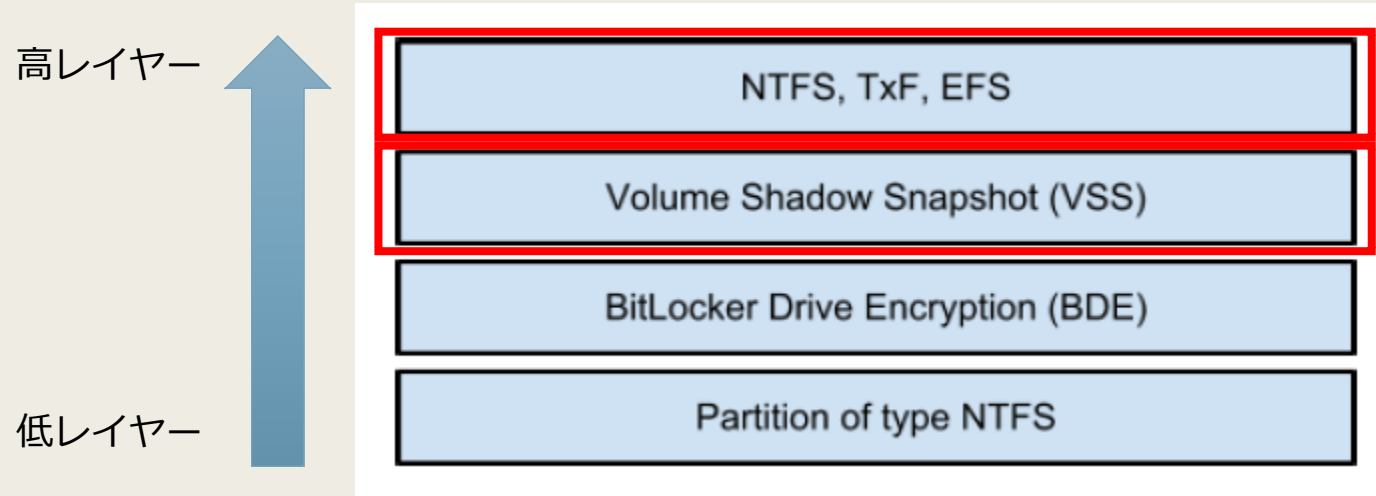
- VSSスナップショットの管理データはルートフォルダ直下のSystem Volume Informationに保存される

The screenshot shows an 'Evidence Tree' window with a 'File List' pane. The 'System Volume Information' folder is expanded, and a file with the GUID '{3808876b-c176-4e48-b7ae-04046e6cc752}' is highlighted. A callout box points to this file, stating: 'Catalog : メタ情報を管理 (スナップショット生成日時など)'. Another callout box points to a file with the GUID '{a68f1700-d9b1-11e7-a9a7-7c7a91d0d869}' and the same GUID, stating: 'Store : スナップショットのバックアップデータ'.

Name	Size	Type
SPP		Folder
SystemRestore		Folder
\$I30		File
IndexerVolumeGuid		File
MountPointManagerRemoteDatabase		File
tracking.log	20	Regular File
Wcifs.md	1	Regular File
WPSettings.dat	1	Regular File
{3808876b-c176-4e48-b7ae-04046e6cc752}	64	Regular File
{a68f1700-d9b1-11e7-a9a7-7c7a91d0d869}{3808876b-c176-4e48-b7ae-04046e6cc752}	327,680	Regular File

VSSシステムのレイヤー

- VSSスナップショットの管理データはファイルとしても管理されているが、VSSはNTFSより下のレイヤーで動作しているため、VSSがスナップショットのデータを参照する際、NTFSをパースするのではなくオフセットをたどる。



<https://github.com/libyal/documentation/blob/master/Paper%20-%20Windowless%20Shadow%20Snapshots.pdf>

VSSスナップショットへのアクセス

- オフセットをたどることによってVSSスナップショットにアクセス



VSS Volume Header (1)

- NTFSボリュームの先頭から0x1e00に保存される情報
- VSS Identifier
 - VSSが有効な場合にセットされる
 - *Catalog*や*Store*のデータブロックにもシグネチャとしてセットされる
- Catalog Offset
 - *Catalog*が保存されているボリュームの先頭からのオフセット
 - スナップショットが0個の場合、0x0がセットされる

VSS Volume Header (2)

000001e00	6B 87 08 38 76 C1 48 4E-B7 AE 04 04 6E 6C C7 52	VSS Identifier	1ÇR
000001e10	01 00 00 00 01 00 00 00-00 1E 00 00 00 00 00 00	
000001e20	00 1E 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
000001e30	00 00 2F 0B 00 00 00 00-3	3 Catalog Offset	00 00 00
000001e40	5A 29 7D E5 C0 B4 E7 11-A9 A4 80 6E 6F 6E 69 63		Z) }âÀ'ç·@ª·nonic
000001e50	5A 29 7D E5 C0 B4 E7 11-A9 A4 80 6E 6F 6E 69 63		Z) }âÀ'ç·@ª·nonic
000001e60	01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
000001e70	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
000001e80	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	

Catalog (1)

- Catalog Block Header
 - *VSS Identifier (Signature)*
 - *Next offset*
 - Catalogは初期状態で4データブロック確保されるため、オフセットがセットされる
- Catalog Entry
 - *Catalog Entry Type 0x02と0x03の2つ一組で1つのスナップショットを管理している*
 - *Catalog Entry Type 0x02*
 - スナップショット生成日時など
 - *Catalog Entry Type 0x03*
 - Store Header Offset, Store Block List Offset, Store Block Range Offset, Store Current Bitmap Offset, Store Previous Bitmap Offset, など

Catalog (2)

Catalog Block Header

0000	6B 87 08 38 76 C1 48 4E-B7 AE 04 04 6E 6C C7 52
0010	01 00 00 00 02 00 00 00-00 00 00 00 00 00 00 00
0020	00 00 2F 0B 00 00 00 00-00 40 2F 0B 00 00 00 00
0030	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0040	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0070	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0080	02 00 00 00 00 00 00 00-00 00 A0 E0 09 00 00 00
0090	00 17 8F A6 B1 D9 E7 11-A9 A7 7C 7A 91 D0 D8 69
00a0	01 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00
00b0	DD C3 5C BF BE 6D D3 01-00 00 00 00 00 00 00
00c0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
00d0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
00e0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
00f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0100	03 00 00 00 00 00 00 00-00 40 50 F0 04 00 00 00
0110	00 17 8F A6 B1 D9 E7 11-A9 A7 7C 7A 91 D0 D8 69
0120	00 00 50 F0 04 00 00 00-00 80 50 F0 04 00 00 00
0130	00 00 51 F0 04 00 00 00-1D 7C 00 00 00 00 02 00
0140	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0150	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0160	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0170	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

VSS Identifier R

Next offset

Entry Type 0x02

スナップショット生成日時
(Windows FILETIME形式)

Entry Type 0x03

Block List Offset

Block Range Offset

Previous Bitmap Offset

Current Bitmap Offset

Store Header Offset

Store (1)

- Store Block Header
 - 4種類のRecord TypeのStore Blockで1つのStoreは管理されている
- Store Header (Store Information) : Record Type 4
 - スナップショットGUID、属性フラグ、マシン名など
- Store Block List : Record Type 3
 - データブロックが記録されていたオリジナルのオフセットとバックアップ先のオフセットのテーブル
- Store Block Range : Record Type 5
 - Store自身が使用しているファイル領域のオフセットとレンジのリスト
- Store Current Bitmap / Store Previous Bitmap : Record Type 6
 - ボリュームのデータブロックの利用状況を表すビットマップ
- Store Data Block
 - バックアップされたデータブロック

Store (2) - Store Block Listの例

00004000	6B 87 08 38 76 C1 48 4E-B7 AE 04 04 6E 6C C7 52	VSS Identifier
00004010	01 00 00 00 03 00 00 00 00 40 00 00 00 00 00 00@.....
00004020	00 40 50 F0	-@P@.....
00004030	00 00 00 00
00004040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00004080	00 80 29 B9 00 00 00 00 00 C0 00 00 00 00 00 00	..}^.....À.....
00004090	00 C0 50 F0 04 00 00 00 00 00 00 00 00 00 00 00	..ÀP@.....
000040a0	00 80 29 B9 00 00 00 00 00 00 06 00 00 00 00 00	..}^.....
000040b0	00 00 56 F0 04 00 00 00 00 02 00 00 00 00 FF 00	..v@.....ÿ.....
000040c0	00 80 29 B9 00 00 00 00 00 01 00 00 00 00 00 00	..}^.....
000040d0	00 00 56 F0 04 00 00 00 00 02 00 00 00 00 FF 00	..V@.....ÿ.....
000040e0	00 80 29 B9 00 00 00 00 00 01 00 00 00 00 00 00	..}^.....
000040f0	00 00 56 F0 04 00 00 00 00 02 00 00 00 00 00 FF	..V@.....ÿ.....

Record Type 3 = Store Block List

Original data block offset

Relative store data block offset

Store data block offset

Flag

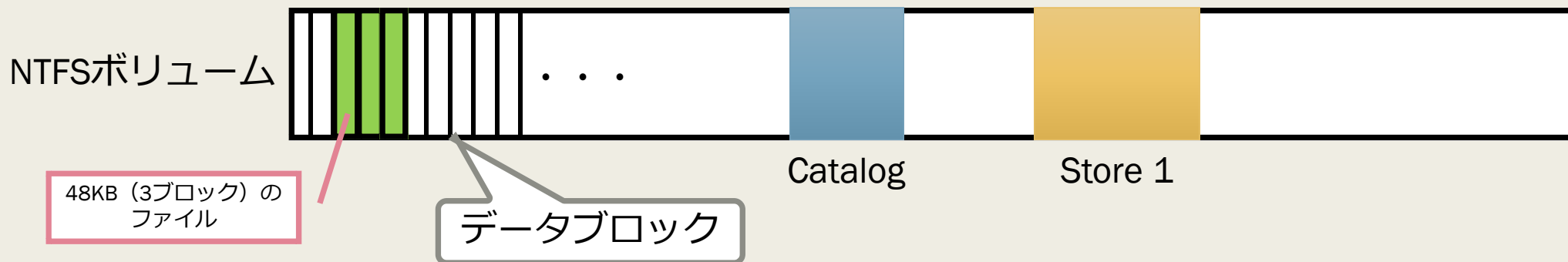
Allocation bitmap

VSSスナックプシヨット の仕組み

VSSスナップショットの データ保存

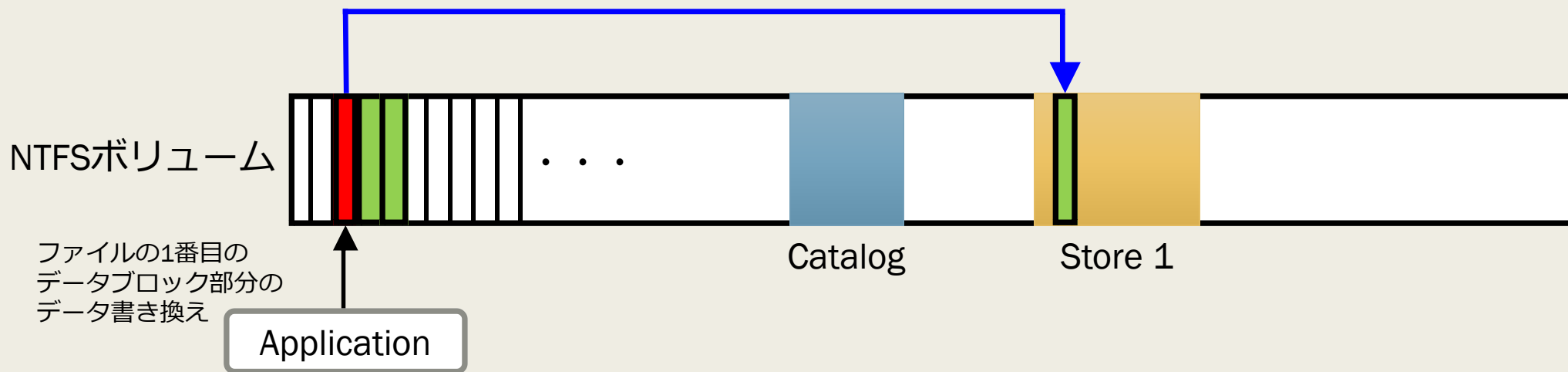
VSSスナップショットのデータ保存 (1)

- スナップショットを作成するとスナップショット作成日時などのメタ情報を保存するCatalogとバックアップデータを保存するStoreが用意される。
- NTFSボリュームは16KBごとのデータブロックと呼ばれる単位で管理される。
- 例として、3つのデータブロックを使用するファイルのスナップショットについて考えてみる。



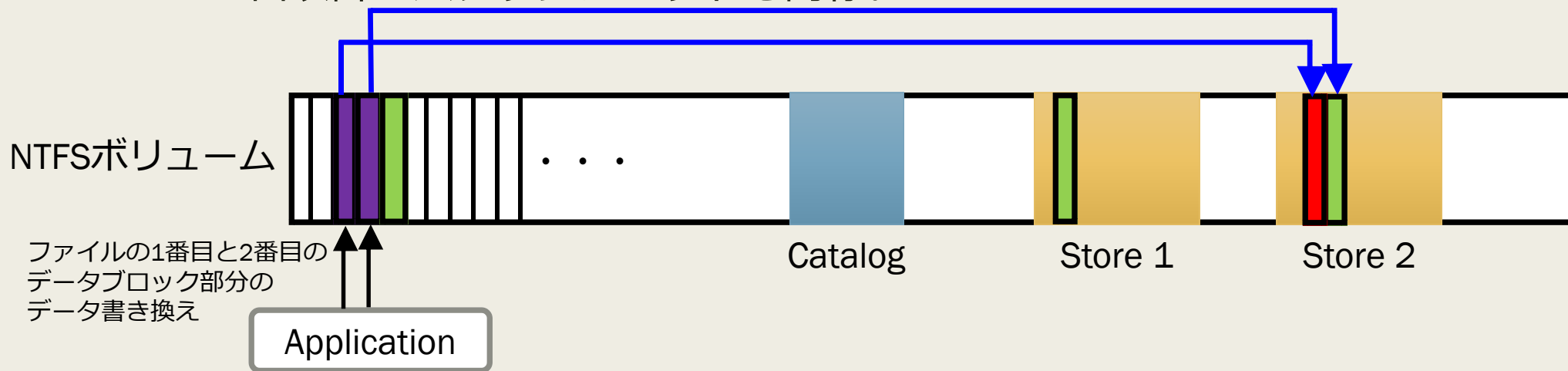
VSSスナップショットのデータ保存 (2)

- ファイルの保存などでデータ変更が発生したデータブロックはデータが書き込まれる前にStoreにデータブロック単位でバックアップされる。



VSSスナップショットのデータ保存 (3)

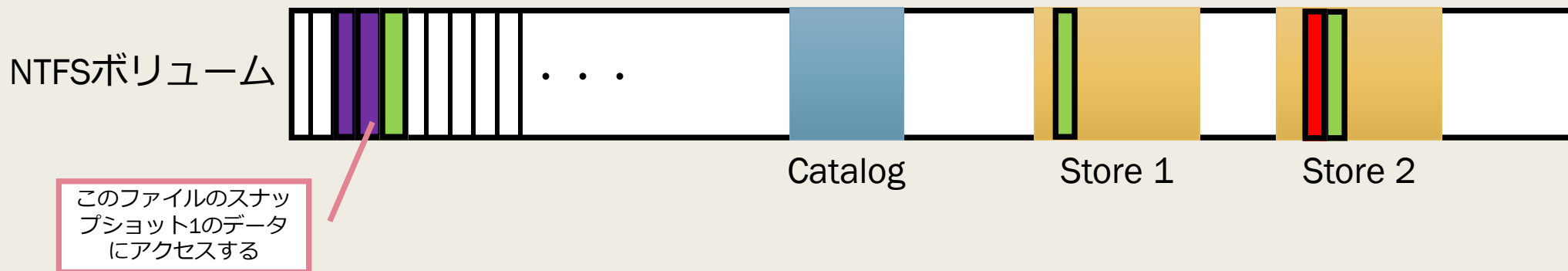
- 2つ目のスナップショットを作成するとCatalogへのメタ情報の追加と2つ目のStoreが用意され、変更が発生したデータブロックはStore 2にバックアップされる。
- 3つ目以降のスナップショットも同様。



VSSスナップショットの データアクセス

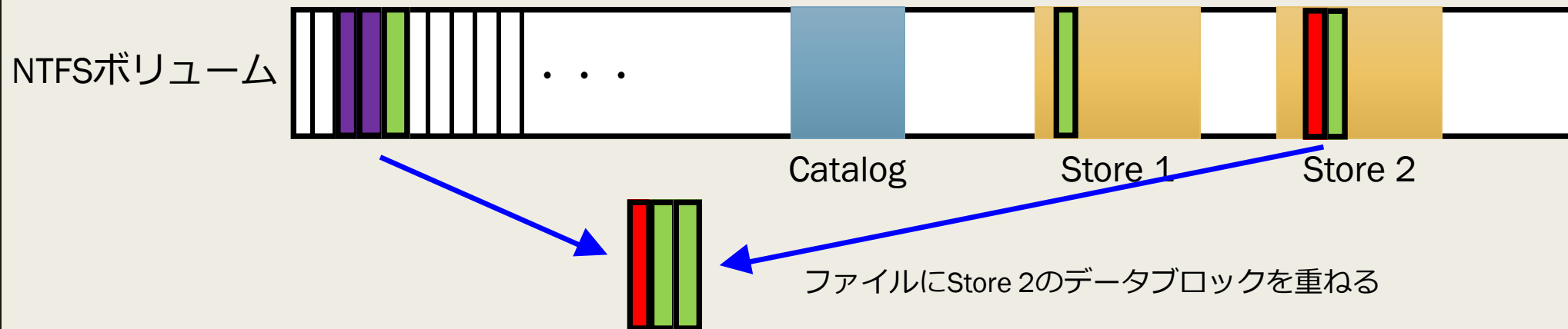
VSSスナップショットのデータアクセス ス (1)

- スナップショットのデータにアクセスする際、Storeに保存されているデータブロックと現在のNTFSボリュームを組み合わせて、スナップショット作成時のデータブロックを再現する。
- 例として、スナップショット1作成時のファイルにアクセスする場合を考える。



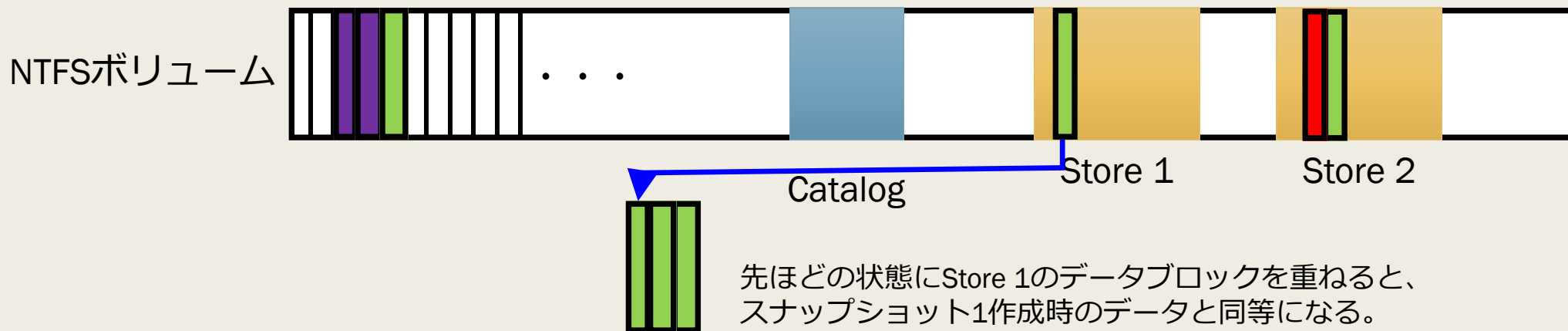
VSSスナップショットのデータアクセス ス (2)

- Store2に保存されているデータブロックをファイルに重ね合わせる。



VSSスナップショットのデータアクセス ス (3)

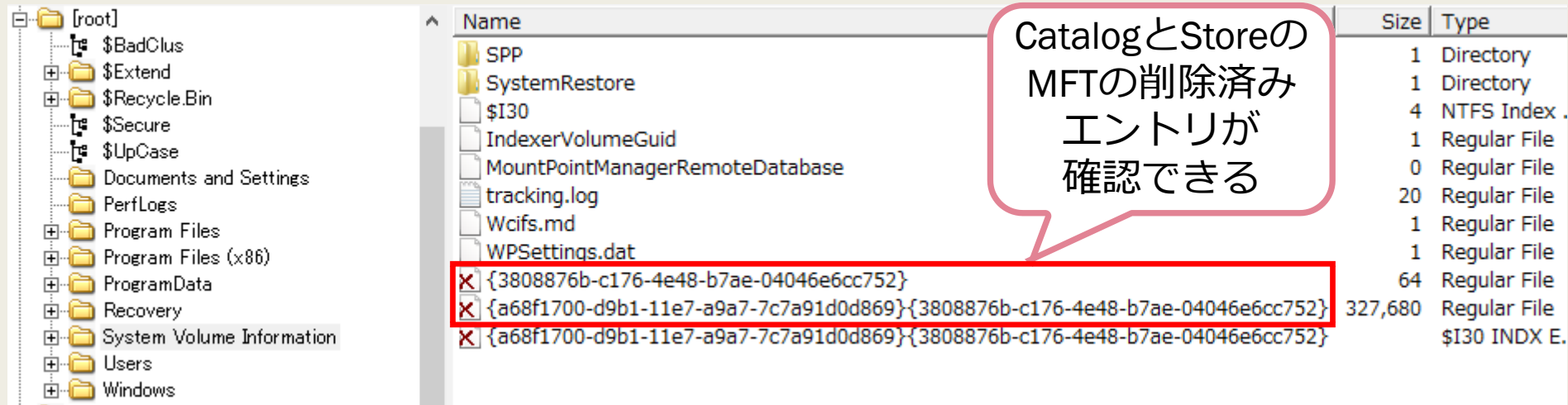
- Store1に保存されているデータブロックをファイルに重ね合わせる。
- このように、スナップショット内にバックアップされたデータブロックを新しい順にNTFSボリュームのデータブロックに重ねることで、スナップショット作成時のデータにアクセスすることができる。



VSSスナップショットの削除

スナップショット削除時の動作 (1)

- スナップショット削除コマンド
 - `vssadmin.exe delete shadows /all`
- スナップショット削除直後のCatalogとStoreの状態



Name	Size	Type
SPP	1	Directory
SystemRestore	1	Directory
\$I30	4	NTFS Index
IndexerVolumeGuid	1	Regular File
MountPointManagerRemoteDatabase	0	Regular File
tracking.log	20	Regular File
Wcifs.md	1	Regular File
WPSettings.dat	1	Regular File
{3808876b-c176-4e48-b7ae-04046e6cc752}	64	Regular File
{a68f1700-d9b1-11e7-a9a7-7c7a91d0d869}{3808876b-c176-4e48-b7ae-04046e6cc752}	327,680	Regular File
{a68f1700-d9b1-11e7-a9a7-7c7a91d0d869}{3808876b-c176-4e48-b7ae-04046e6cc752}		\$I30 INDX E.

スナップショット削除時の動作 (2)

- スナップショット削除直後のCatalog のファイル内容

```
0000 6B 87 08 38 76 C1 48 4E-B7 AE 04 04 6E 6C C7 52 k--8vÄHN·@·nlÇR
0010 01 00 00 00 02 00 00 00-00 00 00 00 00 00 00 00 .....
0020 00 00 2F 0B 00 00 00 00-00 40 2F 0B 00 00 00 00 --/.....@/.....
0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0080 01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0090 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00a0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00b0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00c0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00d0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00e0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
00f0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100 01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
Jap 0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

Entry Typeが全て0x01
となり、データは
0x00で埋められる

スナップショット削除時の動作 (3)

- スナップショット削除直後のStore のファイル内容

削除前

ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	6B	87	08	38	76	C1	48	4E	B7	AE	04	04	6E	6C	07	52	k..8vFHNa..nl双
00000010	01	00	00	00	04	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	50	F0	04	00	00	00	00	00	00	00	00	00	00	00	..P.....
00000030	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	X.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	45	7D	DE	E5	F2	49	A4	40	81	7C	7D	C8	2B	72	58	7F	E} 褪I、@- }ネ+rX.
00000090	C0	D1	72	9A	AF	53	54	4E	AE	DF	B3	14	57	64	6F	21	舛r垂STNa°ウ.Wdo!
000000A0	24	E0	5B	CD	90	60	23	4D	A0	52	04	3F	1F	53	95	69	\$黒^泰#M.R.?.S品
000000B0	0D	00	00	00	01	00	00	00	0D	00	42	00	00	00	00	00B.....
000000C0	0A	00	57	00	49	00	4E	00	31	00	30	00	0A	00	57	00	..W.I.N.1.0...W.
000000D0	49	00	4E	00	31	00	30	00	00	00	00	00	00	00	00	00	I.N.1.0.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

削除後

ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	6B	87	08	38	76	C1	48	4E	B7	AE	04	04	6E	6C	C7	52	k..8vFHNa..nl双
00000010	01	00	00	00	04	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	50	F0	04	00	00	00	00	00	00	00	00	00	00	00	..P.....
00000030	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	X.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	B4	42	21	F1	4B	9A	AF	49	A8	51	70	0C	42	FD	C2	BE	IB!・垂IィQp.B.泔
00000090	C0	D1	72	9A	AF	53	54	4E	AE	DF	B3	14	57	64	6F	21	舛r垂STNa°ウ.Wdo!
000000A0	24	E0	5B	CD	90	60	23	4D	A0	52	04	3F	1F	53	95	69	\$黒^泰#M.R.?.S品
000000B0	0D	00	00	00	01	00	00	00	0D	00	42	00	00	00	00	00B.....
000000C0	0A	00	57	00	49	00	4E	00	31	00	30	00	0A	00	57	00	..W.I.N.1.0...W.
000000D0	49	00	4E	00	31	00	30	00	00	00	00	00	00	00	00	00	I.N.1.0.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

ヘッダの一部 (動作に影響のないGUID) が変更される

スナップショット削除時の動作 (4)

- スナップショット削除から数分経過後

Name	Size	Type
SPP	1	Directory
SystemRestore	1	Directory
\$I30	4	NTFS Index
IndexerVolumeGuid	1	Regular File
MountPointManagerRemoteDatabase	0	Regular File
tracking.log	20	Regular File
Wcifs.md	1	Regular File
WPSettings.dat	1	Regular File
{a68f1700-d9b1-11e7-a9a7-7c7a91d0d869}{3808876b-c176-4e48-b7ae-04046e6cc752}		\$I30 INDX E.

主なVSSスナック ショットパーサ

主なVSSスナップショットパーサ

- Forensic Tool Kit、X-ways Forensics、AXIOM、EnCaseなどの商用製品
- ShadowExplorer、ShadowKitなどのフリーソフトウェア
- libvshadowなどのオープンソースソフトウェア
- ほとんどのツールは削除されたスナップショットにアクセスできない
 - X-WaysはMFTにCatalogとStoreの削除エントリが残っていればスナップショットにアクセス可能。
 - しかし、スナップショット削除後、数分経過するとこれらのエントリはMFTから削除されてしまうため、実用性は高くない。
- 今回は、libvshadow を作成するツールのベースとして採用した。

なぜ、libvshadowなのか

- 商用製品がVSSスナップショットをうまく処理できない場合でも、libvshadowであれば正常に処理できることが多くある。
- ライブラリだけではなく、スナップショットをディスクイメージとして再現するvshadowmountというコマンドがあるため、他のディスクイメージを処理するツールと組み合わせやすい。
- Windowsの機能を使用せずにVSSスナップショットパーサを実装しており、オープンソースソフトウェアであるため、機能の拡張がしやすい。
- 誰でも使用することが可能。
- <https://github.com/libyal/libvshadow>

削除済みVSSスナップ ショットアクセス手法の 検討

削除済みVSSスナップショットアクセス手法の検討 (1)

- vshadowmountはディスクイメージ内のCatalogとStoreを読み込んでスナップショットのデータにアクセスするため、CatalogとStoreを復元または再生成する必要がある。
- しかし、CatalogとStoreの復元には以下のような問題が存在する。
 1. Storeについては動作に影響しない一部のデータのみが書き換わるだけであるため、ディスクイメージからカービングしたデータを利用できることが期待できる。しかし、Storeは4種類のストアブロックから構成されるため、カービングしたストアブロックを1つのStoreしてグルーピングする方法が必要となる。
 2. Catalogのデータは完全に失われてしまうため、カービングしたStoreなどから再生成する必要がある。
 3. 複数のStoreがカービングできた場合、Storeが作成された順番が分からない。

削除済みVSSスナップショットアクセス手法の検討 (2)

■ 問題点1

- Storeについては動作に影響しない一部のデータのみが書き換わるだけであるため、ディスクイメージからカービングしたデータを利用できることが期待できる。しかし、Storeは4種類のストアブロックから構成されるため、カービングしたストアブロックを1つのStoreしてグルーピングする方法が必要となる。

■ 解決策1

- 各種ストアブロックのNTFSボリューム上での配置を調査して、グルーピングの方法を検討する。

削除済みVSSスナップショットアクセス手法の検討 (3)

- CatalogのEntry Type 0x03に記録されている各種ストアブロックのオフセットを見るところの範囲に収まっていることが分かる (NTFSボリュームの大きさによって変化する)。

- Store Header Offset: 0x02F1BA8000
- Store Block List Offset: 0x02F1BAC000
- Store Block Range Offset: 0x02F1BB0000
- Store Current Bitmap Offset: 0x02F1BD4000
- Store Previous Bitmap Offset: 0x02F1C14000

Entry Type 0x03

0100	03 00 00 00 00 00 00 00 00 00	00 C0 BA F1 02 00 00 00	Store Block List Offset
0110	8F 0C DF F9 12 70 E7 11-93 25 00 50 56 A3 26 03	00 00 BB F1 02 00 00 00	Store Block Range Offset
0120	00 80 BA F1 02 00 00 00	4F 48 00 00 00 00 17 00	Store Previous Bitmap Offset
0130	00 40 BD F1 02 00 00 00	00 40 C1 F1 02 00 00 00	Store Current Bitmap Offset
0140	02 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	Store Header Offset
0150	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0160	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0170	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	

削除済みVSSスナップショットアクセス手法の検討 (4)

- 次に、ディスクイメージ内のストアブロックを検索するツールを作成し、他のStoreのストアブロックと入り組んだ配置になっていないか確認すると、Storeごとに離れて配置されていることが確認できる。
- また、StoreはRecord Type 4, 3, 5, 6, 6の順番でストアブロックが現れるため、これらを1つのStoreとしてカービングしても問題ないであろうと考えられる。

```
0x2e8044000-0x2e8140000(0x100000) : Ver:1 RType:3 Next:0x2f81d8000
0x2f1ba8000-0x2f1ba8000(0x4000) : Ver:1 RType:4 Next:0x0
0x2f1bac000-0x2f1bac000(0x4000) : Ver:1 RType:3 Next:0x2f2490000
0x2f1bb0000-0x2f1bb0000(0x4000) : Ver:1 RType:5 Next:0x0
0x2f1bd4000-0x2f1bfc000(0x2c000) : Ver:1 RType:6 Next:0x0
0x2f1c14000-0x2f1c3c000(0x2c000) : Ver:1 RType:6 Next:0x0
0x2f2490000-0x2f258c000(0x100000) : Ver:1 RType:3 Next:0x3065c4000
0x3065c4000-0x3065c4000(0x4000) : Ver:1 RType:4 Next:0x0
0x3065c8000-0x3065c8000(0x4000) : Ver:1 RType:3 Next:0x3065e84000
0x3065cc000-0x3065cc000(0x4000) : Ver:1 RType:5 Next:0x0
0x3065e4000-0x30660c000(0x2c000) : Ver:1 RType:6 Next:0x0
0x30661c000-0x306644000(0x2c000) : Ver:1 RType:6 Next:0x0
0x306e84000-0x306f80000(0x100000) : Ver:1 RType:3 Next:0x31820c000
```

1つ目のStore

Record Type 4, 3, 5, 6, 6を
1つのStoreとして扱う

2つ目のStore

削除済みVSSスナップショットアクセス手法の検討 (5)

■ 問題点2

- *Catalog*のデータは完全に失われてしまうため、カービングしたStoreなどから再生成する必要がある。

■ 解決策2

- *Catalog*に必要な主な情報は以下の通り。
 - スナップショット生成日時
 - Store Header Offsetなどの各オフセット
- オフセットはカービングしたStoreから取得することができるが、スナップショット生成日時は完全に失われている。
- *vshadowmount*はスナップショット生成日時の新しい順にスナップショットを並べ替えてスナップショットのデータにアクセスする。つまり、スナップショット間で前後関係が正しければ、生成日時はいつでも良い。
- このことから、カービングを行った日時を基準にスナップショット生成日時をセットすることとした（問題点3と関連）。

削除済みVSSスナップショットアクセス手法の検討 (6)

■ 問題点3

- 複数のStoreがカービングできた場合、Storeが作成された順番が分からない。
 - スナップショットのデータを参照する際に、正しい順番でデータブロックを再構成しないと正しいデータを再現することができない。

■ 解決策3

- 新しいStoreはNTFSボリュームのオフセットが大きい位置に配置されると仮定し、Catalogを生成する際に、一番大きなオフセットのStoreのスナップショット生成日時にカービングを行った日時をセットし、1つ前のスナップショット生成日時には、1時間前の日時をセットするようにした（以降、同様に1時間ずつ前の日時をセットする）。
- しかし、実際には新しいStoreが小さなオフセットに作成される場合もある。このような状況はツールで自動的に判断できないため、利用者がスナップショットを読み込む順番を簡単に入れ替えられるようなツールを別途作成した。

作成したツールの概要と 復元テスト

作成したツール

- vss_carver.py (カービングツール)
 - ディスクイメージからStoreをカービングする
 - カービングしたStoreからCatalogを生成する
 - ディスクイメージ内にCatalogがあれば、その情報とカービングした情報をマージする (情報はCatalogを優先する)
- vss_catalog_manipulator.py (カタログを操作するツール)
 - カタログエントリの順番の変更や削除などを行う
- 拡張版vshadowmount (libvshadow-20170902をベースに作成)
 - vss_carver.pyで復元したCatalogとStoreからスナップショットの情報を読み込むオプションを追加した

vss_carver.pyの使い方

- -o / --offset : ディスクイメージ内のNTFSボリュームのオフセット
- -i / --image : ディスクイメージのファイルパス
- -c / --catalog : Catalog書き出し先のファイルパス
- -s / --store : Store書き出し先のファイルパス
- `vss_carver.py -o 123456 -i y:¥image -c z:¥catalog -s z:¥store`

vss_catalog_manipulator.pyの使い方

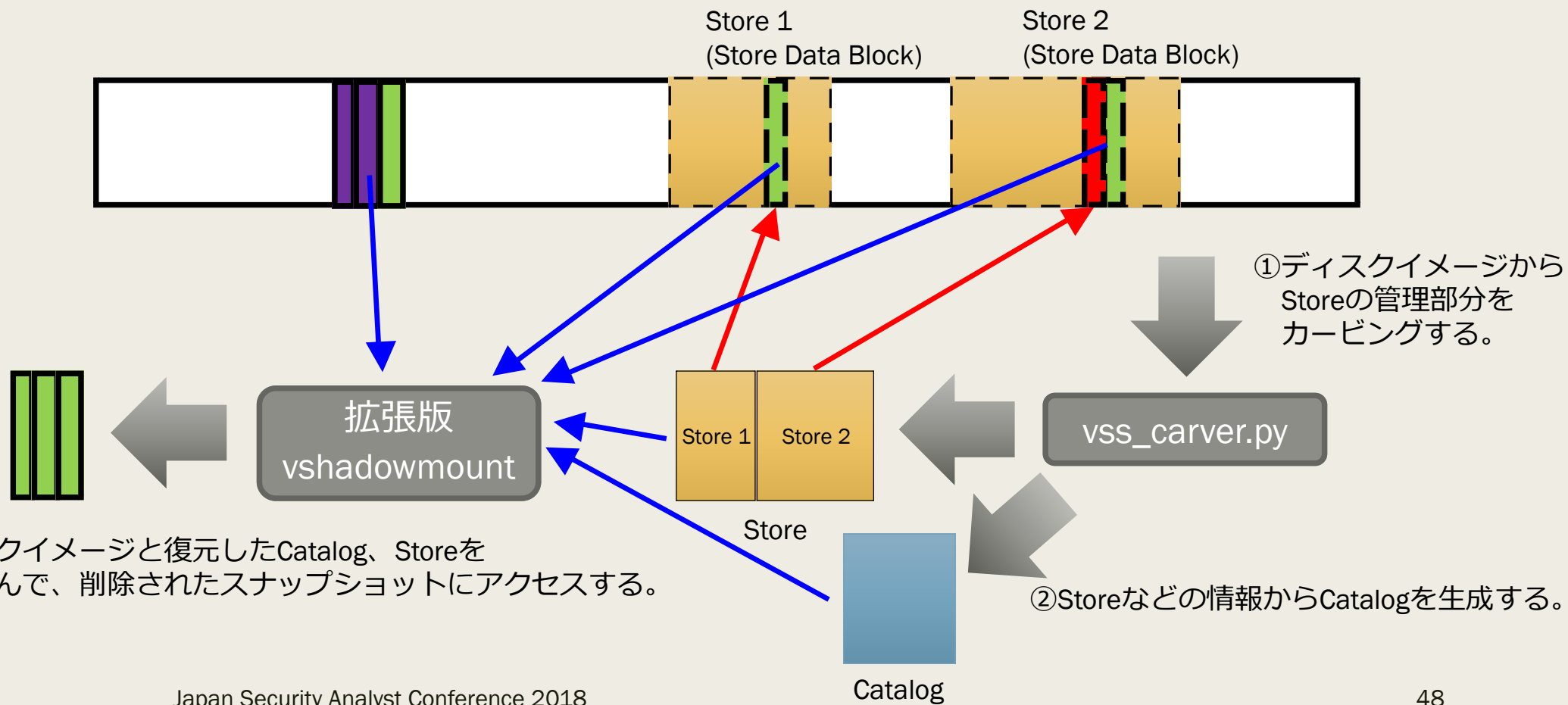
- list : Catalog内の情報を出力
 - `vss_catalog_manipulator.py list z:¥catalog`
- move : Catalogの5番のエントリを3番の位置に移動する
 - `vss_catalog_manipulator.py move z:¥catalog 5 3`
- remove : Catalogの2番のエントリを削除する
 - `vss_catalog_manipulator.py remove z:¥catalog 2`
- enable : Catalogの4番のエントリを有効化する
 - `vss_catalog_manipulator.py enable z:¥catalog 4`
- disable : Catalogの7番のエントリを無効化する
 - `vss_catalog_manipulator.py disable z:¥catalog 7`

拡張版vshadowmountの使い方

- 既存のオプションは変更せず、新たにオプションを追加した
- -c : Catalogの情報としてvss_carver.pyで生成したCatalogファイルを使用する
- -s : Storeの情報としてvss_carver.pyでカービングしたStoreファイルを使用する
- vshadowmount.exe -o 123456 -c z:¥catalog -s z:¥store y:¥image x:

作成したツールの概要

→ Store Data Block (バックアップされたデータブロック) 参照先
→ データ/ファイル読込



ファイル復元テスト

- 以下の操作を行ったディスクイメージからファイルを復元できるか、複数のスナップショットパーサでテストを行った。
- 事前準備
 - 3KB, 5MB, 15MB のファイルを10個ずつ保存し、スナップショットを作成後、各ファイルの先頭に1バイトのデータを追加してファイルを保存。
- テスト1
 - 全てのスナップショット削除後、1分以内のディスクイメージ (MFTに削除エントリが残っている状態)
- テスト2
 - 全てのファイルを削除し、別の10個の5MBのファイルのコピーと削除を5回繰り返す。
- テスト3
 - `Teslacrypt`を実行して、ファイルを暗号化させる。
 - 外部への通信ができない検証環境で実行する必要があったため、通信ができなくともファイルを暗号化する `Teslacrypt`を使用した。

ファイル復元テスト結果

ソフトウェア	テスト1	テスト2	テスト3	備考
商用製品A (Ver. X)	○	×	×	MFTにStoreのエントリ がある場合のみ復元可能
商用製品A (Ver. Z)	×	×	×	上記と同じ製品だが新しいバージョンにも関わらず、エンバグしている可能性がある
商用製品B	×	×	×	
フリーソフトウェアC	×	×	×	
libvshadow + vss_carver.py	○	○	○	

○ : すべてのファイルが復元できた
× : 1つもファイルが復元できなかった



DEMO

Demo 1: 自動的に削除されたスナップショットの復元

- 1か月程度運用したWindows 7のディスクイメージ
- ディスクイメージ内のCatalogには3つのスナップショットのエントリが記録されているが、カービングを行うと、自動的に削除されたスナップショットのエントリを見つけることができる。
- この削除されたスナップショットから既存のスナップショットより古いデータを復元できる可能性がある。

Demo 2: ランサムウェアに削除されたスナップショットの復元

- 被害者コンピュータ：Windows 10
 - スナップショットを作成。既存のファイルを編集・保存する。
 - *Teslacrypt*を実行後、暗号化が完了したことを確認して、VMwareのスナップショットを作成する。
- 解析コンピュータ：Windows 7
 - 被害者コンピュータのVMwareディスクイメージマウントして、*vshadowinfo*でスナップショットがないことを確認する（*Teslacrypt*による削除）。
 - *vss_carver.py*がカービングしたCatalogとStoreを使って、拡張版*vshadowmount*でVMwareディスクイメージをマウントする。
 - 暗号化される前のファイルが復元できるか確認する。

ScopeSnapshotsの確認

- VSSスナップショットには、もう一つ考慮しなければならない点がある。
- Windows 8以降、デフォルトでScopeSnapshotsが有効になっている。
- ScopeSnapshotsが有効になっていると、VSSスナップショットにはシステム関連のファイルのみが保存され、ユーザが作成したデータは保存されない。
- ScopeSnapshotsを無効化するには、「HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥SystemRestore」キーに「ScopeSnapshots」という名前でDWORD値「0」を設定し、OSを再起動する。
- 詳細は弊社発行の冊子IIR Vol.37を参照。
 - <https://www.iij.ad.jp/dev/report/iir/037.html>

Future Work

- 拡張版vshadowmountのWindows以外のOSへの対応
- ストアブロックの一部が上書きされている場合の補完処理

まとめ

- ディスクイメージからStoreとCatalogを復元し、これらのファイルを機能拡張したvshadowmountで読み込むことで削除されたVSSスナップショットのデータにアクセスできるようになった。
- また、自然に削除されたスナップショットやランサムウェアによって削除されたスナップショットにも有効であることが確認できた。
- 作成したツールの公開先と公開予定時期
 - https://github.com/mnrkbys/vss_carver
 - 拡張版vshadowmountパッチ公開予定時期：2018年2月
 - バイナリは先行して公開中

Q & A