프리패치 포렌식

JK Kim

@pr0neer

forensic-proof.com

proneer@gmail.com

개요

1. 윈도우 메모리 관리 기법
2. 프리패치
3. 슈퍼패치
4. 레디부스트

윈도우 메모리 관리 기법

메모리 관리 기법

- **가상 메모리(Virtual Memory)**
  - 페이징 파일(pagefile.sys)을 이용하여 물리메모리 용량 한계 해결 (다중 프로그램 실행)

- **프리패치(Prefetch)**
  - 운영체제 부팅, 응용프로그램 실행 시 속도 향상을 위해 프리패칭(prefetching) 기법 사용

- **슈퍼패치(Superfetch)**
  - 프리패치에 부족했던 요소들을 추가하여 성능 향상

- **레디부스트(ReadyBoost)**
  - USB, SD 카드, CF 카드 등의 플래시 메모리를 이용한 캐싱 기법

SSD와 메모리 관리 기법

- **윈도우 7 자동 최적화**
  - 윈도우 7부터는 SSD에 대한 최적화 기능이 추가
  - 저장매체의 속도가 빠르다면 메모리 관리 기술은 불필요
  - **SSD 사용 시 윈도우 7 자동 설정**
    - ✓ TRIM 명령 활성화
    - ✓ 슈퍼패치가 자동으로 중지
    - ✓ 레디부스트의 사용 불필요

프리패치

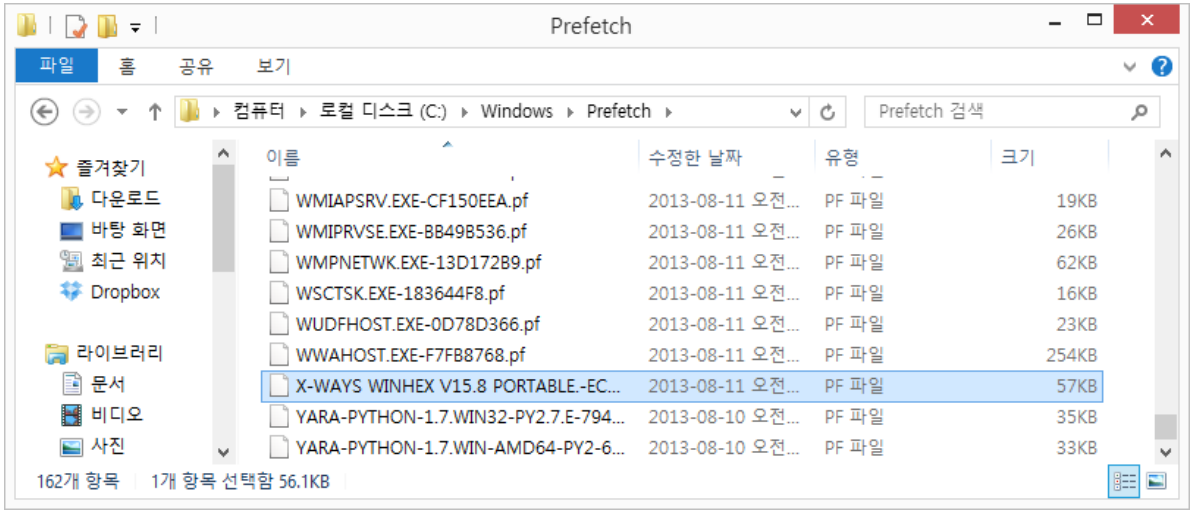
프리패치 소개 (1/7)

- **윈도우 프리패치 (Windows Prefetch)**
  - 실행 파일이 사용하는 시스템 자원을 특정 파일에 미리 저장 → 프리패치 파일
  - 윈도우 부팅 시 프리패치 파일을 모두 메모리에 로드
  - 사용자가 파일을 실행할 경우 미리 저장된 정보를 메모리에서 실행하여 실행 속도를 향상
  - 윈도우 XP 이후 (2003, Vista, 2008, 7) 의 운영체제에서 제공

  - **분류**
    - ✓ 부트 프리패칭 (Boot Prefetching) : XP, 2003, Vista 2008, 7
    - ✓ 응용프로그램 프리패칭 (Application Prefetching) : XP, Vista, 7, 8

프리패치 소개 (2/7)

- **프리패치 저장 경로**
  - **%SystemRoot%Prefetch**



- **파일명**
  - 부트 프리패치 파일 : NTOSBOOT-B00DFAAD.pf
  - 응용프로그램 프리패치 파일 : <filename>-<filepath hash>.pf

프리패치 소개 (3/7)

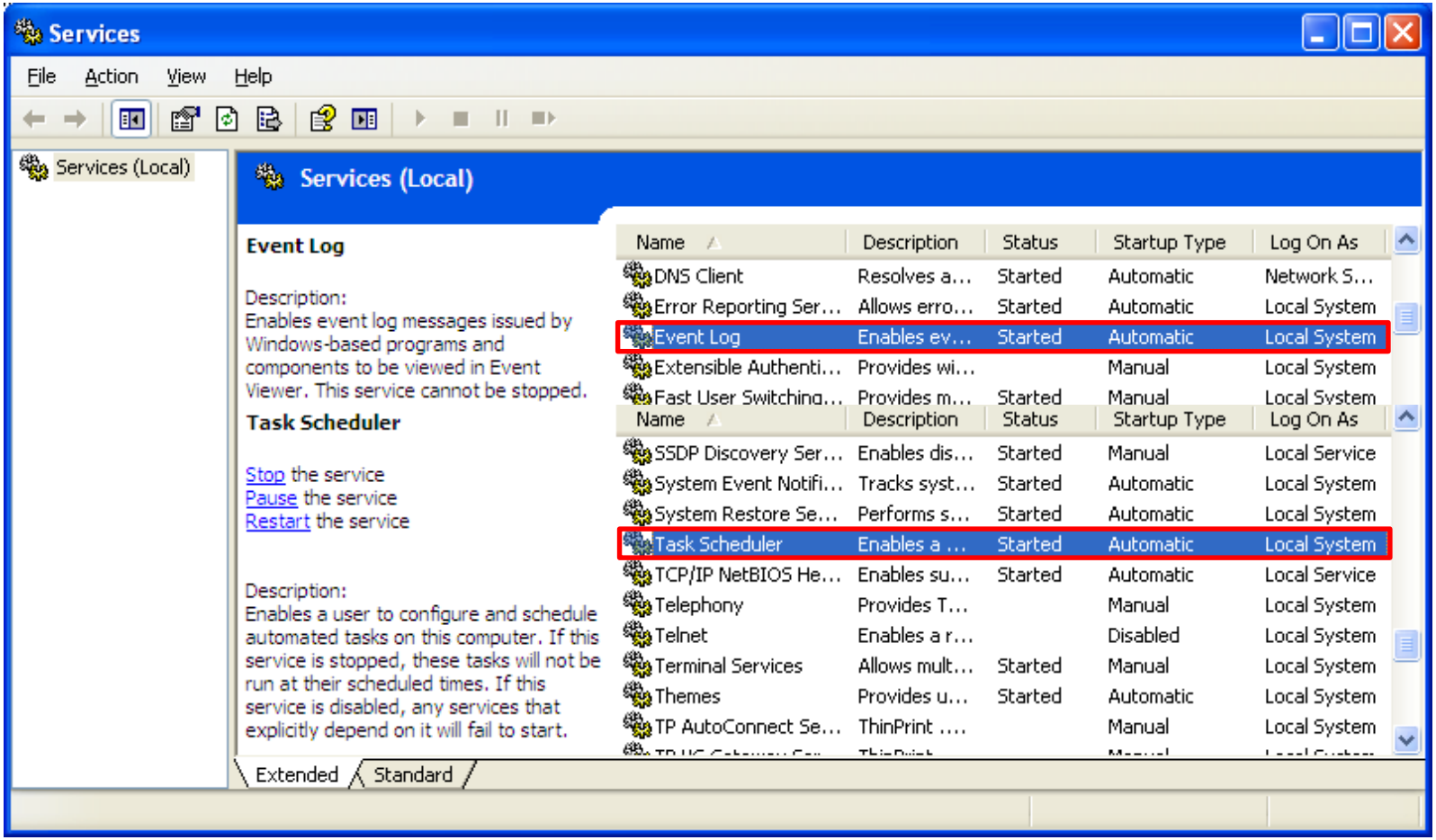
- **부트 프리패칭**
  - 윈도우는 부팅 중 다양한 파일을 사용
  - 부팅과 관련된 파일은 저장매체에 흩어져 있거나 단편화되어 있을 수 있음 ➔ 부팅 속도 저하
  - 프리패처에 의해 시스템이 부팅을 시작한 후 최대 120초 까지 모니터링
  - 부팅 시 사용되는 코드와 데이터를 모니터링한 후 결과를 파일에 저장
  - 부트 프리패칭된 파일을 이용하여 부팅시 속도 향상

프리패치 소개 (4/7)

- **응용프로그램 프리패칭**
  - 응용프로그램 초기 실행시 캐시 관리자는 처음 10초를 모니터링
  - 10초 동안 메모리에 로드한 코드와 데이터의 일부 혹은 전체를 파일로 생성 ➔ 프리패치 파일
  - 프리패칭된 응용프로그램 다시 실행 시 프리패치 파일을 이용해 초기 실행 속도 향상
  - 파일 개수는 최대 128개로 제한 → 최대 한계를 넘으면 오래된 프리패치 파일은 자동 삭제

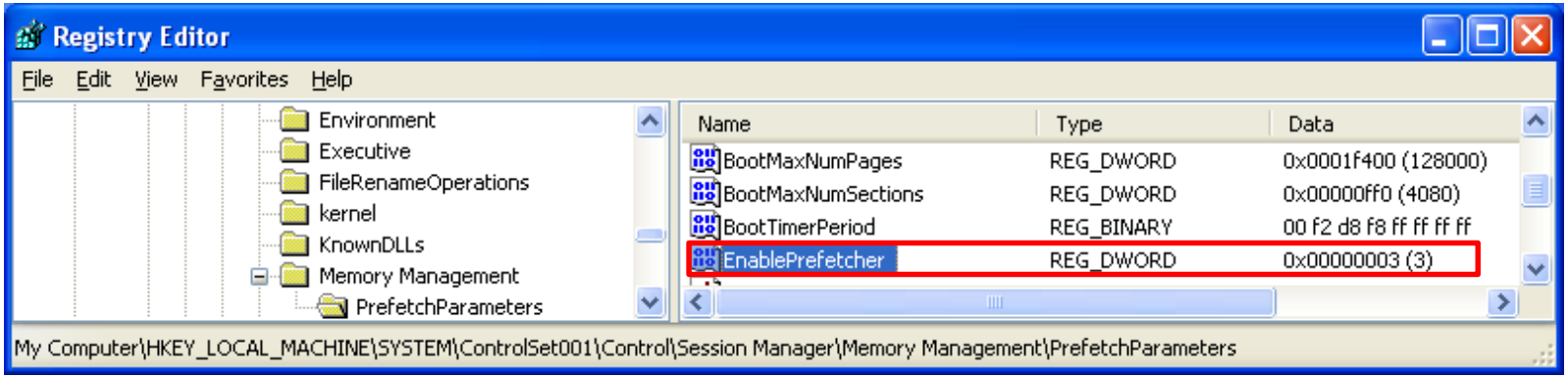
프리패치 소개 (5/7)

- **프리패칭 동작 확인**
  - **[Event Log], [Task Scheduler]** – 두 서비스 중 하나라도 중지되면 프리패칭 동작 중지



프리패치 소개 (6/7)

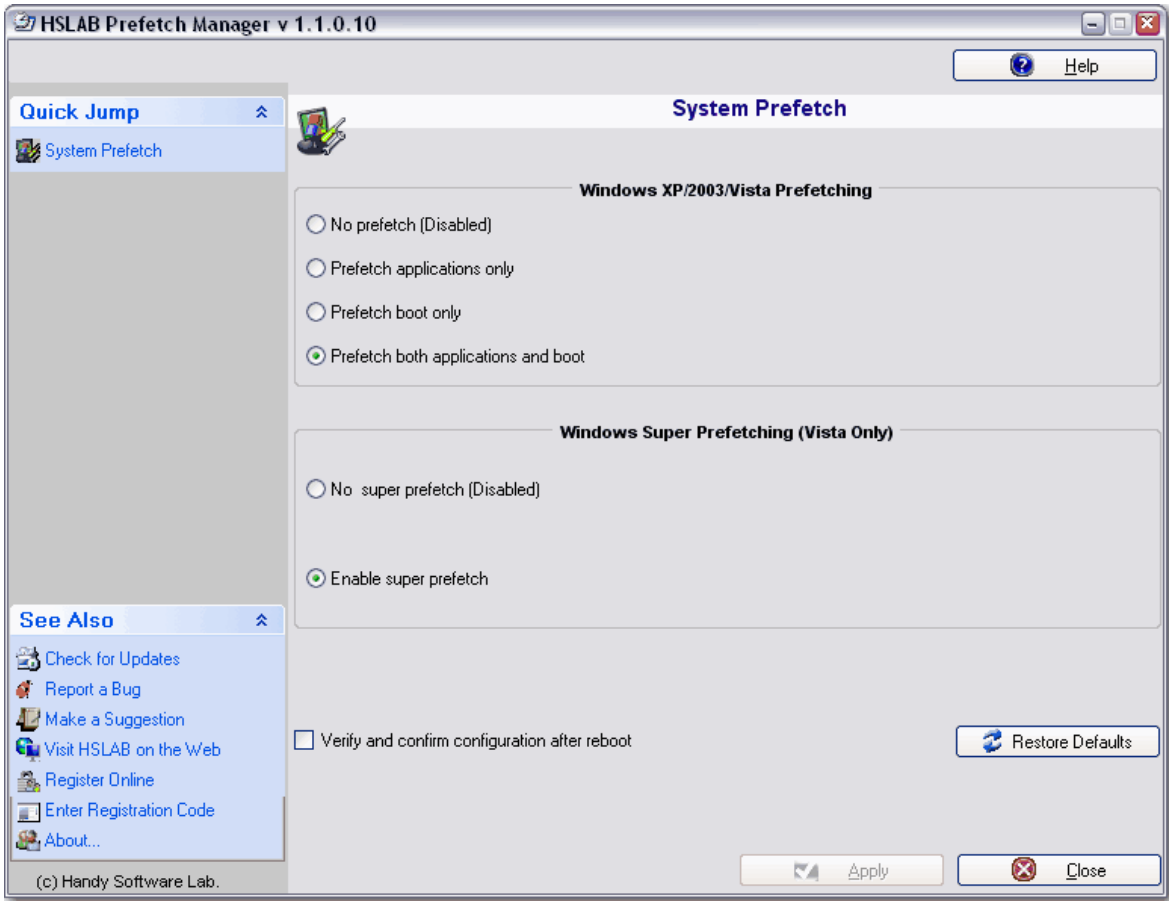
- **레지스트리를 이용한 활성화 설정**
  - HKLM₩SYSTEM₩ControlSet00#₩Control₩Session Manager₩Memory Management₩Prefetch Parameter₩EnablePrefetcher



  - 레벨 0 : 비활성화
  - 레벨 1 : 응용프로그램 프리패칭만 사용
  - 레벨 2 : 부트 프리패칭만 사용
  - 레벨 3 (기본) : 응용/부트 프리패칭 모두 사용

프리패치 소개 (7/7)

- **도구를 이용한 간편 설정**
  - **Prefetch Manager** (XP/2003/Vista) (http://www.hs-lab.com) – Commercial ??



프리패치 파일 구조 (1/6)

- **프리패치 파일의 포렌식적 의미**
  - 프리패치 파일은 다양한 정보를 저장 → 포렌식적으로 중요한 정보 활용 가능
  - 프리패치 파일에서 획득 가능한 정보
    - ✓ 응용프로그램 이름
    - ✓ 응용프로그램 실행 횟수
    - ✓ 응용프로그램 마지막 실행 시각 (Windows 64-bit Time Stamp, FILETIME)
    - ✓ 참조 목록 (파일 수행에 필요한 DLL, SDB, NLS, INI 등의 경로)
    - ✓ 파일시스템 시간 정보 (생성, 수정, 마지막 접근 시간 등)를 이용한 통합 타임라인 분석

프리패치 파일 구조 – 윈도우 XP/2003 (2/6)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Prefetcher Version (0x00000011)				Signature ("SCCA")				Prefetcher Management Service Version (0x0000000F)				File Size			
0x10	Executable File Name (길이가 58byte 를 넘을 경우 파일이름 끝에 0x0000 기록)															
0x20																
0x30																
0x40											파일이름 58일때 0x0000		Full Path Hash Value			
0x50	0x00000000				SectionInfoOffset				NumSections				PageInfoOffset			
0x60	NumPages				FileNameInfoOffset				FileNameInfoSize				MetadataInfoOffset (디스크 볼륨정보)			
0x70	NumMetadataRecords (디스크 볼륨 갯수)				MetadataInfoSize				LastLaunchTime (최종 실행 시각)							
0x80	MinRePrefetchTime								MinReTraceTime							
0x90	NumLaunches (실행 횟수)				Sensitivity											

프리패치 파일 구조 – 윈도우 XP/2003 (3/6)

- **Ex) ANALYZEMFT.EXE-283ED250.pf**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
00000000	11	00	00	00	53	43	43	41	0F	00	00	00	7C	5C	01	00	....SCCA....|\..
00000010	41	00	4E	00	41	00	4C	00	59	00	5A	00	45	00	4D	00	A.N.A.L.Y.Z.E.M.
00000020	46	00	54	00	2E	00	45	00	58	00	45	00	00	00	00	00	F.T...E.X.E.....
00000030	00	00	00	00	00	0D	DB	BA	40	5D	94	B1	00	B0	AB	B1	.........@].....
00000040	18	5C	94	B1	20	D0	A1	81	40	5D	94	B1	50	D2	3E	28	.\.. ...@]..P.>(
00000050	00	00	00	00	98	00	00	00	8F	00	00	00	C4	0B	00	00	....
00000060	80	15	00	00	C4	0D	01	00	56	47	00	00	20	55	01	00	........VG.. U..
00000070	01	00	00	00	5C	07	00	00	E8	0A	42	82	D6	CC	CB	01	....\.....B.....
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000090	05	00	00	00	03	00	00	00	00	00	00	00	45	00	00	00	............E...

- 실행 파일 이름 : ANALYZEMFT.EXE
- 실행 파일 경로 해쉬값 : 0x283ED250
- 실행 파일 마지막 실행 시각 : 2011. 02. 15. (Tue) 06:06:39 (UTC)
- 실행 파일 실행 횟수 : 5

프리패치 파일 구조 – 윈도우 Vista/7 (4/6)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Prefetcher Version (0x00000017)				Signature ("SCCA")				Prefetcher Management Service Version (0x00000011)				File Size			
0x10	Executable File Name (길이가 58byte 를 넘을 경우 파일이름 끝에 0x0000 기록)															
0x20																
0x30																
0x40											파일이름 58일때 0x0000		Full Path Hash Value			
0x50	0x00000000				SectionInfoOffset				NumSections				PageInfoOffset			
0x60	NumPages				FileNameInfoOffset				FileNameInfoSize				MetadataInfoOffset (디스크 볼륨정보)			
0x70	NumMetadataRecords (디스크 볼륨 갯수)				MetadataInfoSize				UnKnown							
0x80	LastLaunchTime (최종 실행 시각)								Unknown				0x00000000			
0x90	Unknown				0x00000000				NumLaunches (실행 횟수)				Sensitivity			

프리패치 파일 구조 – 윈도우 Vista/7 (5/6)

- **Ex) CHROME.EXE-917B6FB8.pf**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
00000000	17	00	00	00	53	43	43	41	11	00	00	00	36	F1	00	00	....SCCA....6...
00000010	43	00	48	00	52	00	4F	00	4D	00	45	00	2E	00	45	00	C.H.R.O.M.E...E.
00000020	58	00	45	00	00	00	FF	FF	00	00	00	00	00	00	00	00	X.E..............
00000030	58	00	00	00	80	FA	FF	FF	00	00	00	00	00	00	00	00	X...............
00000040	00	00	00	00	00	00	00	00	1E	7B	EE	02	B8	6F	7B	91	.........{...o{.
00000050	00	00	00	00	F0	00	00	00	37	00	00	00	D0	07	00	00	........7.......
00000060	61	10	00	00	5C	CC	00	00	F4	1A	00	00	50	E7	00	00	a...\.......P...
00000070	01	00	00	00	E6	09	00	00	0F	00	00	00	01	00	00	00	....
00000080	7A	66	B8	14	3E	CF	CB	01	00	8C	86	47	00	00	00	00	zf..>......G....
00000090	00	8C	86	47	00	00	00	00	16	07	00	00	08	00	00	00	...G............

- 실행 파일 이름 : CHROME.EXE
- 실행 파일 경로 해쉬값 : 0x917B6FB8
- 실행 파일 마지막 실행 시각 : 2011. 02. 18. (Fri) 07:33:05 (UTC)
- 실행 파일 실행 횟수 : 1814

프리패치 파일 구조 (6/6)

- **파일시스템 시간 정보와의 통합 분석**
  - 프리패치 파일도 파일시스템 시간 정보를 가짐 (FAT, NTFS 등 파일시스템에 따라 다름)
  - 일반적으로 파일시스템은 파일의 생성, 수정, 마지막 접근 시간을 유지

  - **프리패치 파일 시간 분석**
    - ✓ **생성 시간** : 응용프로그램을 처음 실행한 시각은 아님 (사용에 따라 삭제된 후 재생성될 수 있음)
    - ✓ **수정 시간** : 프리패치 파일이 마지막으로 수정한 시간
    - ✓ **마지막 접근 시간** : 프리패치 파일을 사용하는 응용프로그램이 마지막으로 접근한 시간
    - ✓ 응용프로그램의 마지막 실행 시간은 파일 내부에서 확인해야 함

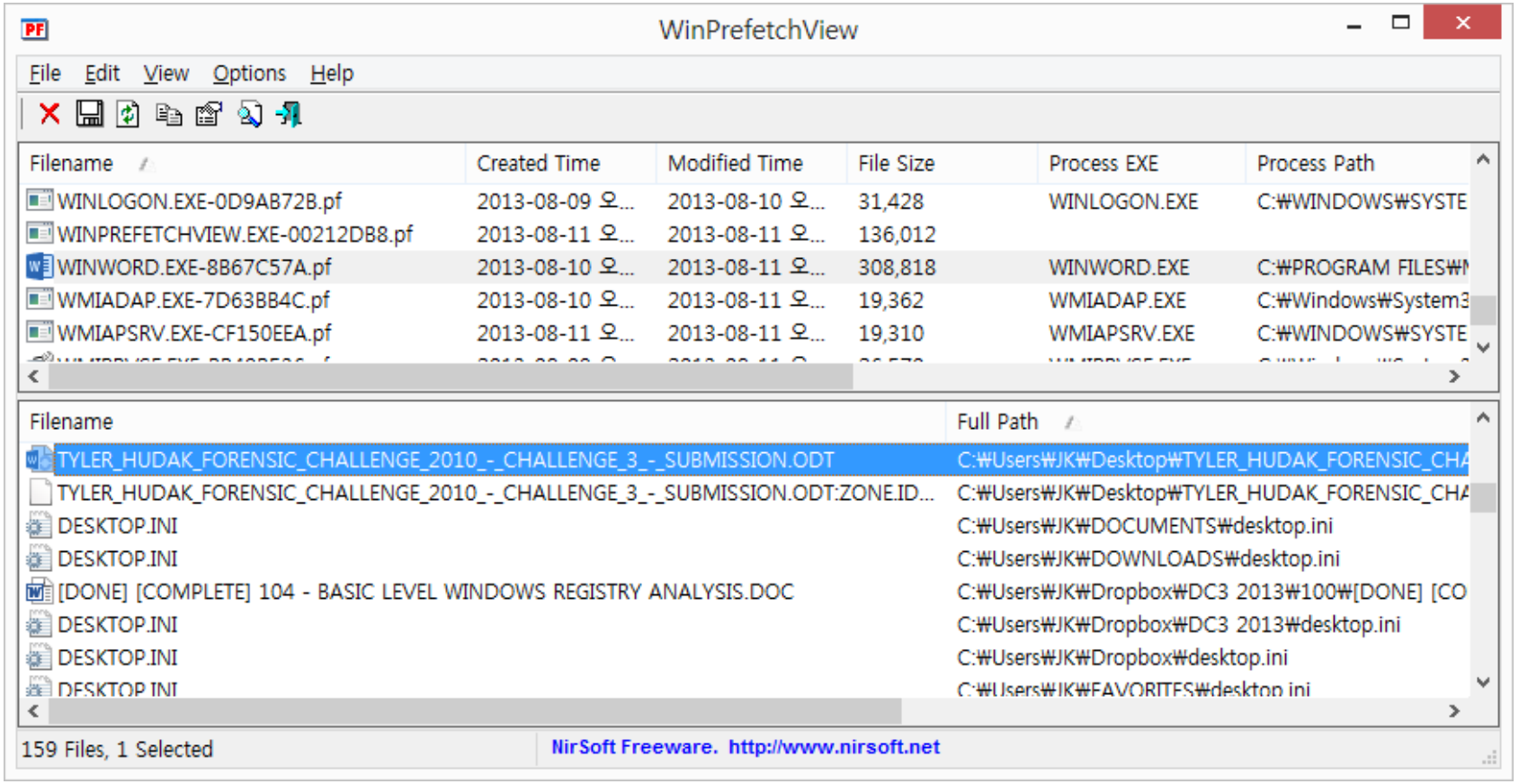
프리패치 활용

- **부트 프리패치**
  - 참조 목록을 통해 부팅 과정에서 로드되는 악성코드 확인 가능

- **응용프로그램 프리패치**
  - 프리패치 목록을 통해 악성코드의 직접 실행 흔적 발견
  - 참조 목록을 통해 참조된 혹은 삽입된 악성흔적 발견
  - 참조 목록을 통해 프로그램 관련 파일 목록 확인
    - ✓ 워드, 엑셀, 한글 – 로드한 문서
    - ✓ 7z, WinZip – 압축 해제한 파일 목록
    - ✓ 인터넷 익스플로러, 크롬 – 임시 파일, 파비콘 등
    - ✓ NOTEPAD++, EditPlus – 편집 기록된 파일 등
    - ✓ 등등

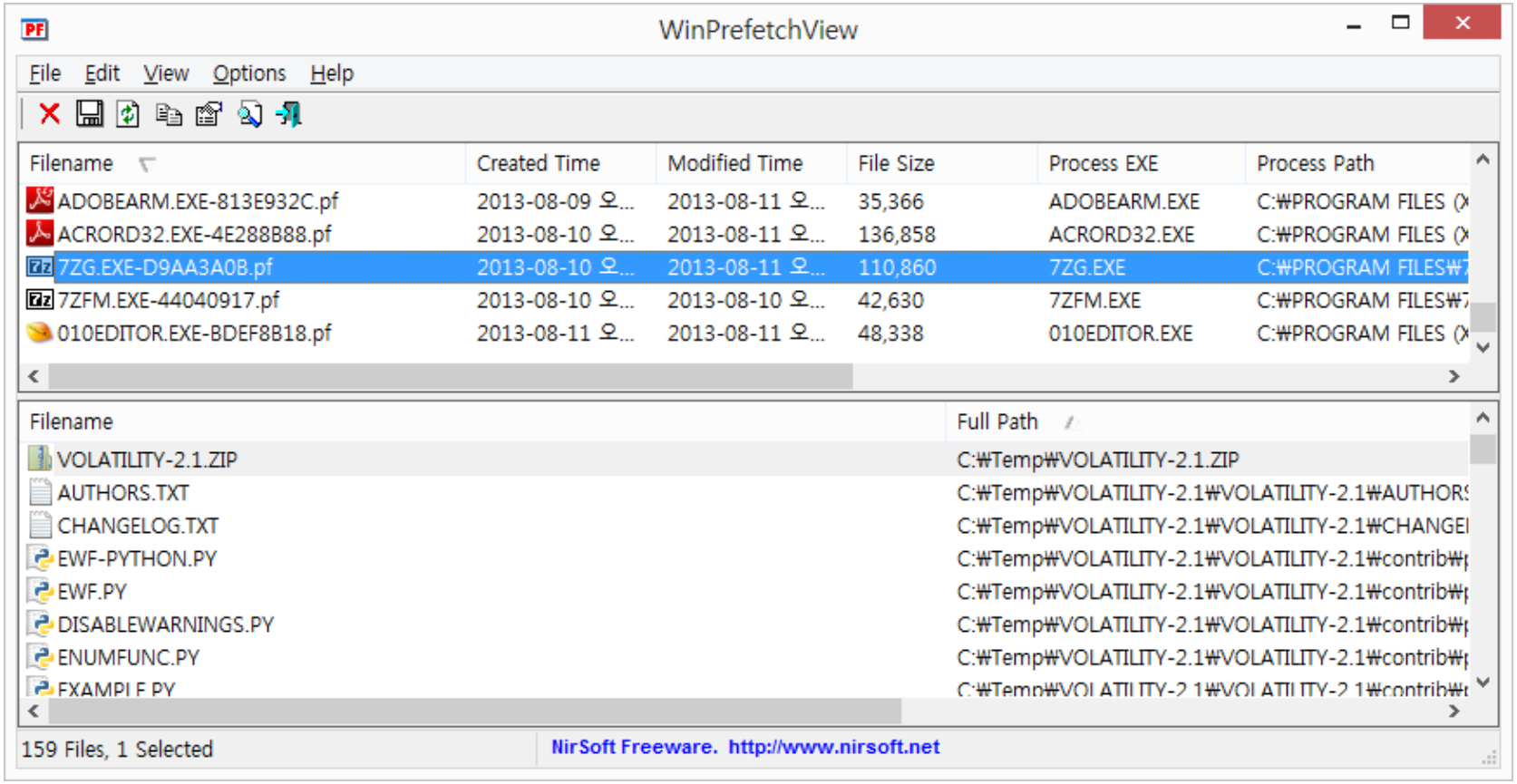
프리패치 활용

- **Ex) WINWORD.EXE**



프리패치 활용

- **Ex) 7ZG.EXE**



프리패치 파일 카빙

- **윈도우 XP, 2003**
  - Prefetcher Version : 0x00000011
  - 시그니처 : "SCCA" (0x41434353)
  - 8바이트로 프리패치 헤더를 찾은 후 파일 크기만큼 카빙
  - 헤더 구조의 추가적인 데이터 검증을 통해 오탐을 줄일 수 있음 (하지만, 8바이트 시그니처는 오탐율이 적음)

- **윈도우 Vista, 7**
  - Prefetcher Version : 0x00000017
  - 시그니처 : "SCCA" (0x41434353)
  - 8바이트로 프리패치 헤더를 찾은 후 파일 크기만큼 카빙
  - 헤더 구조의 추가적인 데이터 검증을 통해 오탐을 줄일 수 있음 (하지만, 8바이트 시그니처는 오탐율이 적음)

실습 #1

- 자신의 시스템에서 프리패치 파일 흔적 확인하기!!

실습 #1

- 비할당 영역에서 프리패치 파일 카빙한 후 분석하기!!!

슈퍼패치

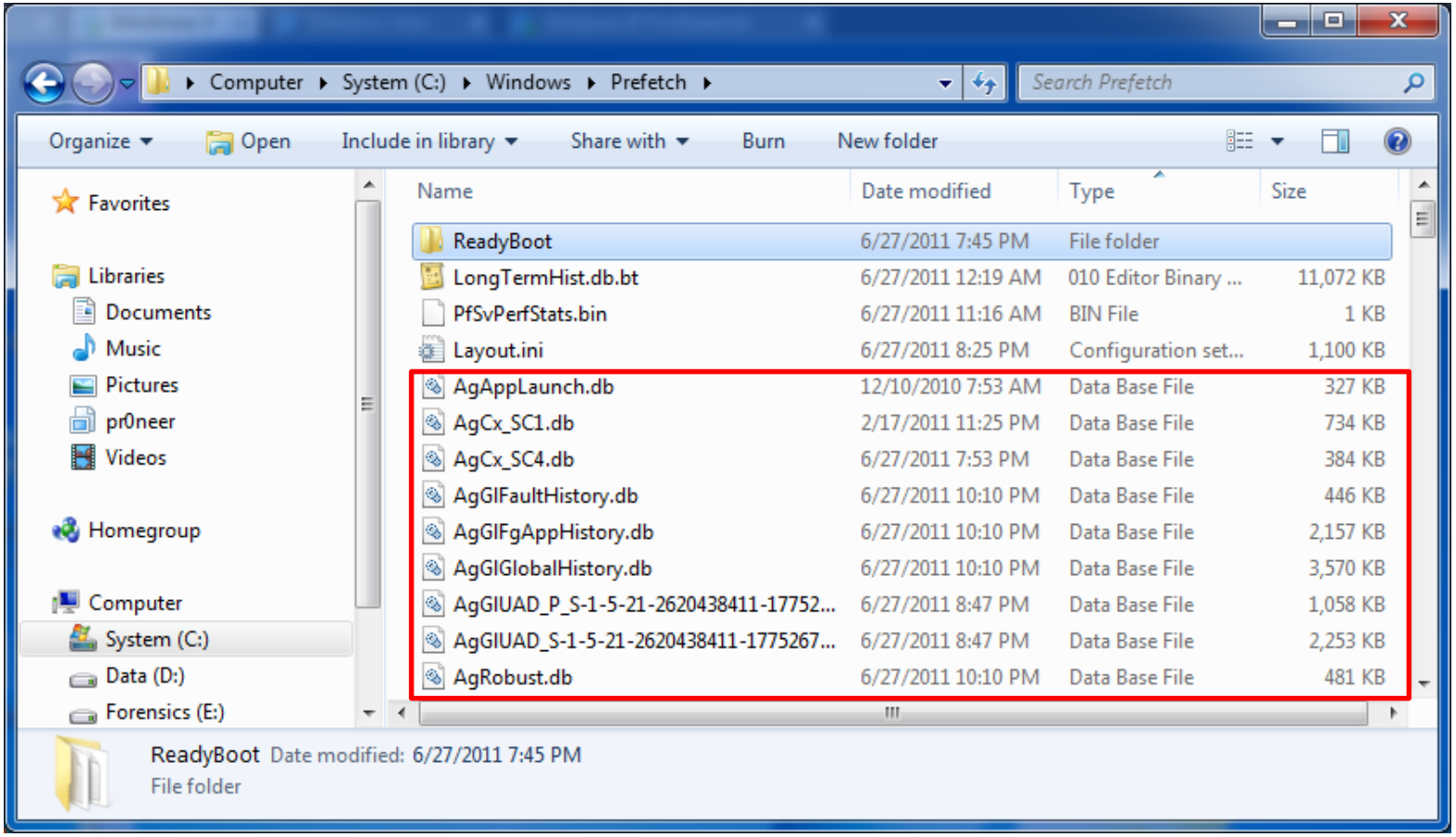
슈퍼패치 소개 (1/5)

- **프리패치 문제점**
  - 프리패치는 응용프로그램 실행 전 미리 메모리에 로딩하는 기술 → 메모리를 이용한 빠른 실행
  - 메모리의 한계로 인해 메모리에 로딩된 프리패치 데이터는 페이징 파일로 이동
  - 다시 응용프로그램 실행 시 페이징 파일로부터 로딩 → 성능 저하

- **슈퍼패치 (Superfetch) 개선점**
  - 사용자의 프로그램 사용 패턴(얼마나 자주, 언제, 얼마 동안) 추가적인 파일에 기록/추적
  - 프리패치 데이터가 페이징 된 후, 프로그램이 종료되면 이를 감지하여 페이징된 데이터를 다시 메모리 로드
    - ✓ 자주 사용하는 프로그램일 경우 지속적으로 메모리 로드

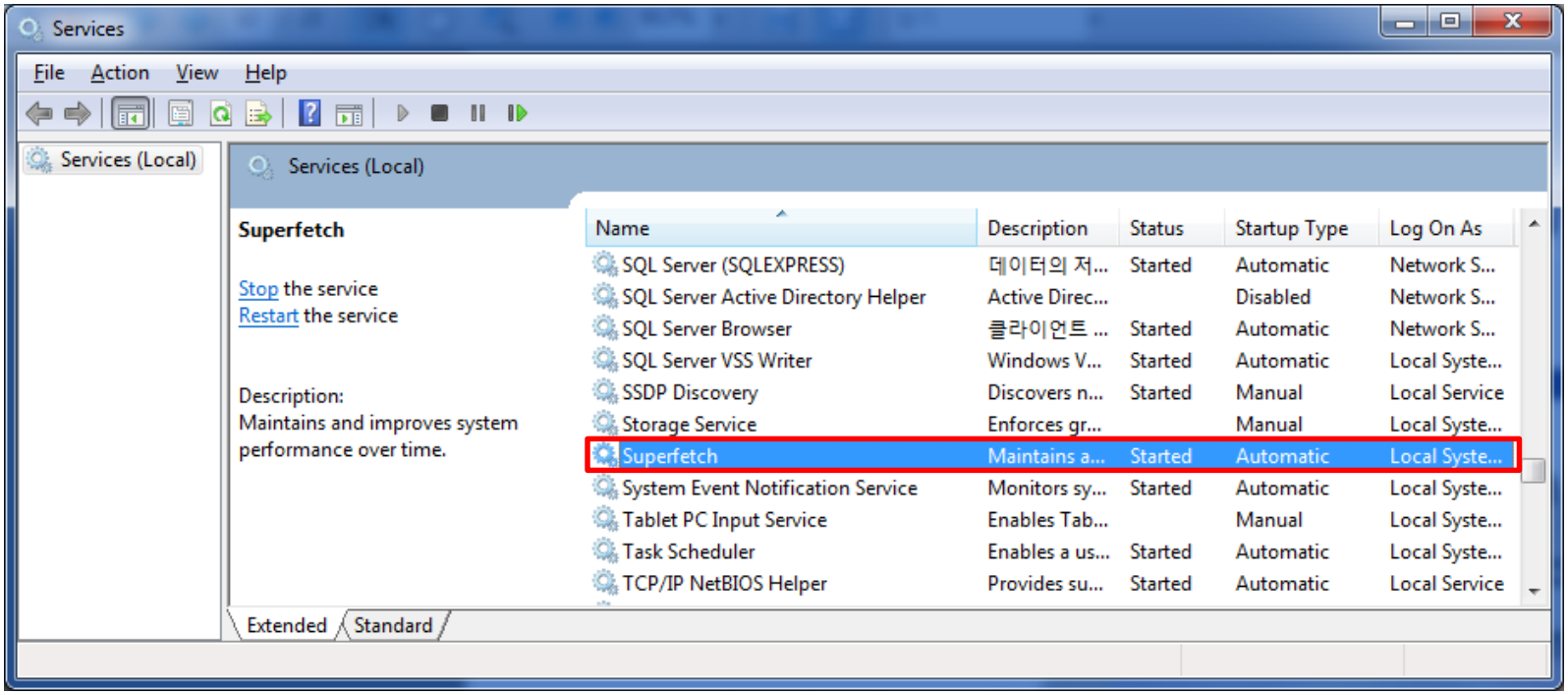
슈퍼패치 소개 (2/5)

- 슈퍼패치 저장 경로 (프리패치와 동일)
  - %SystemRoot%Prefetch



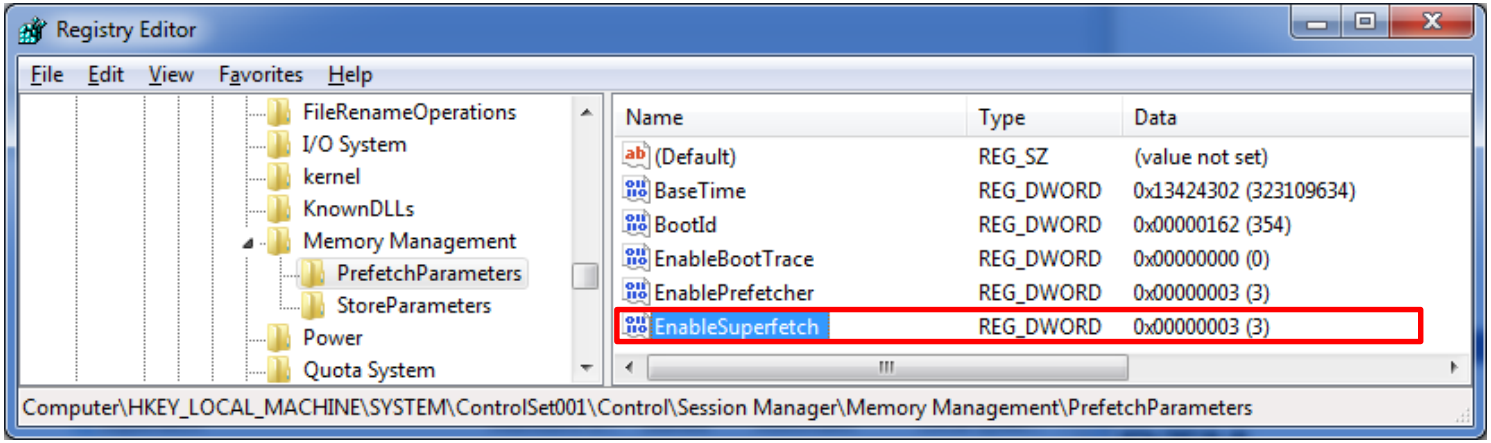
슈퍼패치 소개 (3/5)

- **슈퍼패치 동작 확인**
  - **[Superfetch]** 서비스 실행 시 슈퍼패칭 동작



슈퍼패치 소개 (4/5)

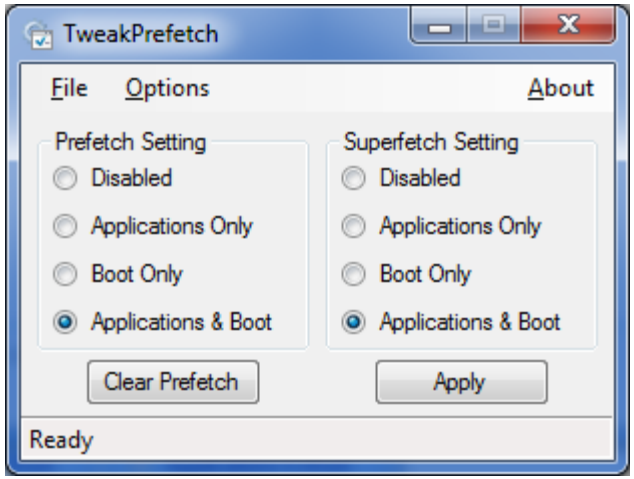
- **레지스트리를 이용한 활성화 설정**
  - HKLM₩SYSTEM₩ControlSet00X₩Control₩Session Manager₩Memory Management₩Prefetch Parameter₩EnableSuperfetch



- 레벨 0 : 비활성화
- 레벨 1 : 응용프로그램 프리패칭시 사용
- 레벨 2 : 부트 프리패칭시 사용
- 레벨 3 (기본) : 응용/부트 프리패칭시 모두 사용

슈퍼패치 소개 (5/5)

- 도구를 이용한 간편 설정
  - **TweakPrefetch** (http://exilesofthardware.blogspot.com/2009/09/tweakprefetch.html)



슈퍼패치 파일 구조 (1/4)

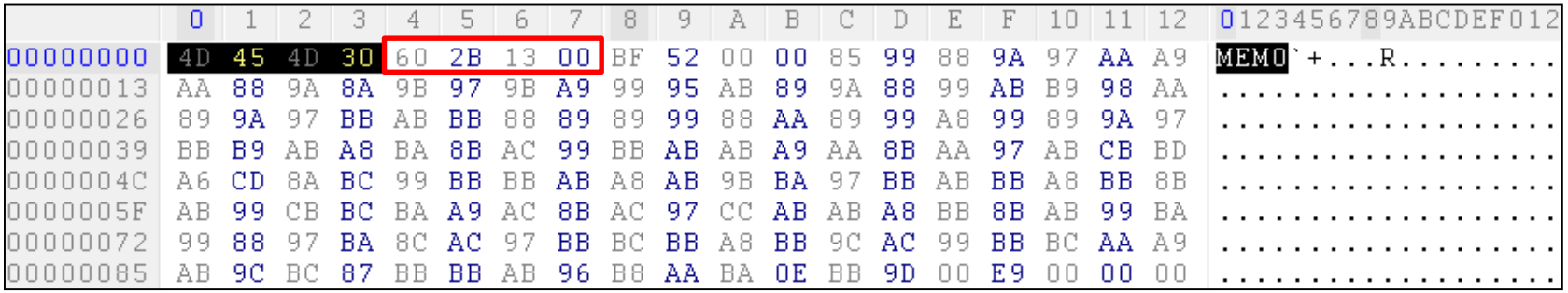
- **파일명**
  - Ag 접두어, .db 확장자 (데이터베이스 파일 ?)
  - TRX 파일

- **압축 vs. 비압축**

AgAppLaunch.db
AgRobust.db
AgCx_SC[number].db
AgGIFaultHistory.db
AgGIFgAppHistory.db
AgGIGlobalHistory.db
AgGIUAD_P_[SID].db
AgGIUAD_[SID].db
LongTemHist.db

슈퍼패치 파일 구조 (2/4)

- **압축 파일 구조 (.db)**



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	0123456789ABCDEF012
00000000	4D	45	4D	30	60	2B	13	00	BF	52	00	00	85	99	88	9A	97	AA	A9	MEMO`+...R.....
00000013	AA	88	9A	8A	9B	97	9B	A9	99	95	AB	89	9A	88	99	AB	B9	98	AA	.....
00000026	89	9A	97	BB	AB	BB	88	89	89	99	88	AA	89	99	A8	99	89	9A	97	.....
00000039	BB	B9	AB	A8	BA	8B	AC	99	BB	AB	AB	A9	AA	8B	AA	97	AB	CB	BD	.....
0000004C	A6	CD	8A	BC	99	BB	BB	AB	A8	AB	9B	BA	97	BB	AB	BB	A8	BB	8B	.....
0000005F	AB	99	CB	BC	BA	A9	AC	8B	AC	97	CC	AB	AB	A8	BB	8B	AB	99	BA	.....
00000072	99	88	97	BA	8C	AC	97	BB	BC	BB	A8	BB	9C	AC	99	BB	BC	AA	A9	.....
00000085	AB	9C	BC	87	BB	BB	AB	96	B8	AA	BA	0E	BB	9D	00	E9	00	00	00	.....

- 시그니처 : 0x4D454D30(“MEMO”)
- 압축 해체 크기 : 0x132B60
- 압축 데이터 시작 위치 : 0x08
- 압축 데이터 블록 시작 값 : 0xB### (각 블록마다)
  - ✓ ### : 압축 데이터 블록 크기

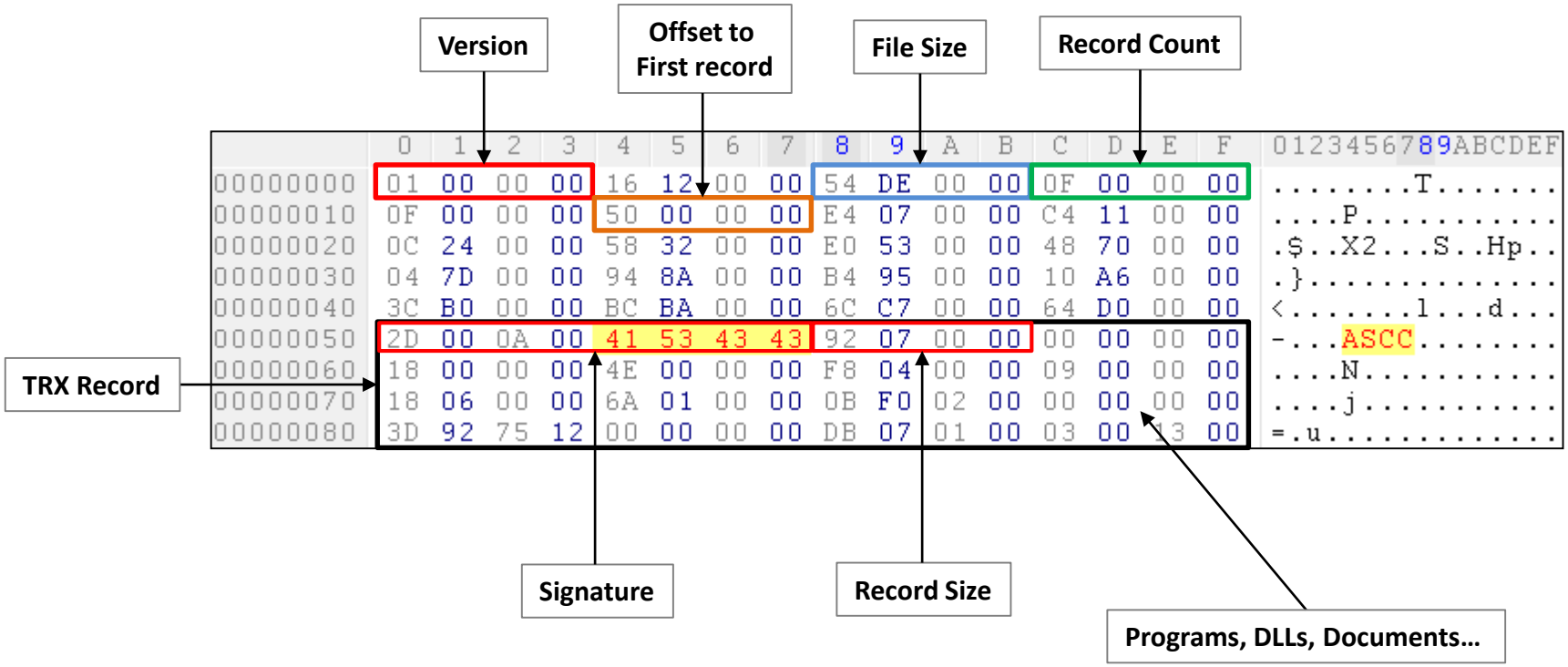
슈퍼패치 파일 구조 (3/4)

- **압축 해제 (.db)**
  - 슈퍼패치는 Superfetch Service Host (sysmain.dll)에 의해 동작
  - Sysmain.dll이 참조하는 ntdll.dll의 임포트 함수 목록 (http://www.win7dll.info/sysmain_dll.html)
    - ✓ RtlCompressBuffer
    - ✓ RtlDecumpressBuffer (http://msdn.microsoft.com/en-us/library/ff552191(v=vs.85).aspx)

```
NTSTATUS RtlDecompressBuffer (
    __in USHORT CompressionFormat,  /* COMPRESSION_FORMAT_LZNT1 */
    __out PUCHAR UncompressedBuffer,
    __in ULONG UncompressedBufferSize,
    __in PUCHAR CompressedBuffer,
    __in ULONG CompressedBufferSize,
    __out PULONG FinalUncompressedSize
);
```

슈퍼패치 파일 구조 (4/4)

- **TRX 파일**
  - 압축되지 않은 파일로 연속된 레코드를 가짐



레디부스트

레디부스트 소개 (1/4)

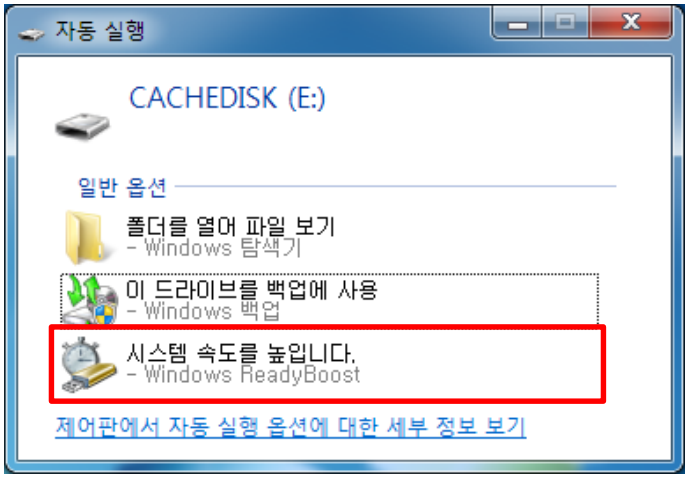
- **레디부스트(ReadyBoost)란?**
  - 윈도우 Vista에서 처음 소개된 디스크 캐시 기법
  - 플래시 메모리, SD 카드, CF 카드 등의 포터블 플래시 저장 장치를 캐시로 활용 ➔ 성능 향상
  - 레디부스트 설정 시 슈퍼패치 데이터도 캐시됨
  - 레디부스트 조건
    - ✓ 이동식 디스크 용량이 최소 256 MB 이상
    - ✓ 윈도우 7은 최대 8개 장치, 256 GB 용량 지원
    - ✓ 이동식 디스크 접근 시간이 1 ms 보다 적어야 함
    - ✓ 4 KB의 랜덤 읽기 성능은 2.5 MB/s, 512 KB의 랜덤 쓰기 성능은 1.75 MB/s 이상이 되야 함
    - ✓ 디스크 성능 측정

```
C:\> winsat disk -read -ran -ransize 4096 -drive [driveletter]
```

```
C:\> winsat disk -write -ran -ransize 524288 -drive [driveletter]
```

레디부스트 소개 (2/4)

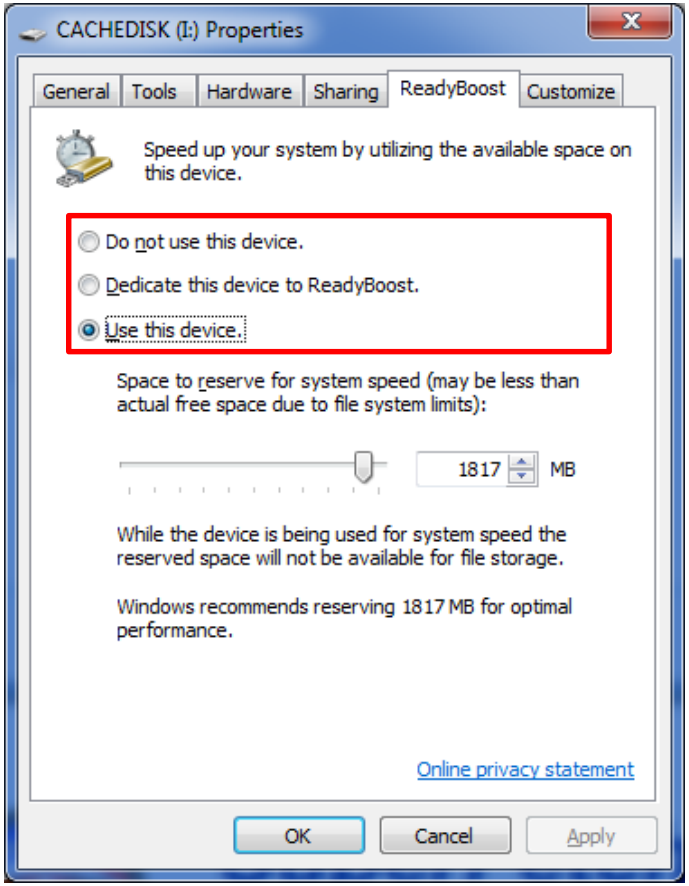
- **레디부스트 사용**
  - **sysmain** (Superfech Service Host)이 동작하는 경우 이동식 디스크 연결



  - 슈퍼패치에 의해 페이지 아웃되면 가상메모리가 아닌 플래시 메모리에 저장
  - 플래시 메모리로부터 필요 페이지를 로드 ➔ 성능 향상

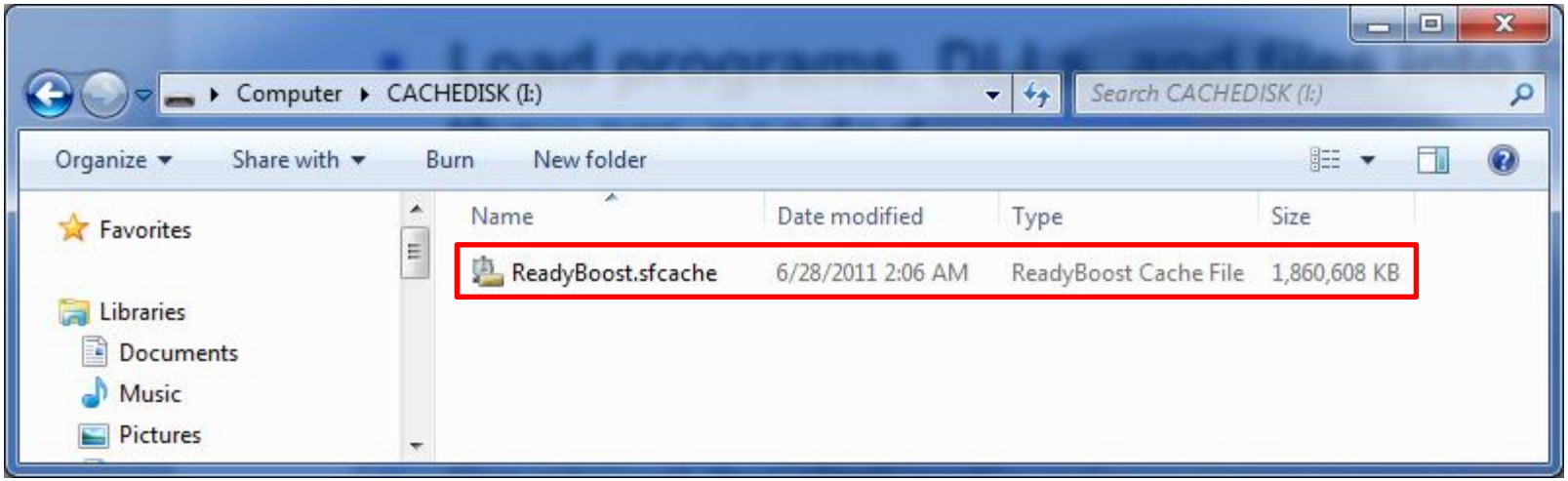
레디부스트 소개 (3/4)

- **레디부스트 사용 옵션**
  - 이동식 디스크 마우스 우클릭



레디부스트 소개 (4/4)

- **디스크 캐시 파일**
  - **ReadyBoost.sfcache** 캐시 파일 생성 (암호화되어 저장)



질문 및 답변