# 윈도우 포렌식

## 1 프리패치

- 목적 : 하드디스크의 속도 때문에 초기 참조하는 파일의 순서를 저장해서 부팅시 빠르게 로드할 수 있도록 함.

- 윈도우 프리패칭 (Windows Prefetching)
  - 실행 파일이 사용하는 시스템 자원 정보를 특정 파일에 저장  프리패치 파일
  - 윈도우 부팅 시 프리패치 파일을 모두 메모리에 로드
  - 사용자가 파일을 실행할 경우 미리 저장된 정보를 이용해 초기 실행 속도 향상
  - 윈도우 XP 이후 (2003, Vista, 2008, 7, 8, 10)의 운영체제에서 제공

- 프리패칭 유형
  - 부트 프리패칭 (Boot Prefetching) : XP, 2003, Vista 2008, 7
    - 부팅을 빠르게 하도록 하는 파일
  - 응용프로그램 프리패칭 (Application Prefetching) : XP, Vista, 7, 8, 10

### 1.1 경로

- SystemRoot%₩Prefetch
  - 부트 프리패치 : NTOSBOOT-B00DFAAD.pf
  - 응용프로그램 프리패치 : <filename>-<filepath hash>.pf

- 이 경로에 들어가면 db파일도 있는데 이 것은 프리패치파일을 메모리에 올렸는데 안 쓸 경우 페이지 파일로 내리니까 그것을 막기 위한 파일이다.

### 1.2 부트프리패칭

- 부팅과 관련된 파일이 저장장치에 흩어져 있거나 단편화되어 있음 -> 부팅 속도 저하
  프리패처에 의해 시스템 부팅 시 최대 120초 까지 모니터링

- 부팅 시 사용하는 파일을 모니터링한 후 결과를 파일에 저장

- 프리패칭된 파일을 이용하여 부팅 속도 향상

### 1.3 응용 프로그램 프리패칭

- 응용프로그램 초기 실행 시 캐시 관리자가 처음 10초를 모니터링
- 10초 동안 사용한 파일을 모니터링한 후 결과를 파일로 저장
- 프리패칭된 응용프로그램 다시 실행 시, 프리패치 파일을 이용해 초기 실행 속도 향상
- 파일 개수는 최대 128개로 제한 -> 한계치를 넘으면 사용되지 않는 파일부터 자동 삭제

### 1.4 획득가능한 정보

- 응용프로그램 이름
- 응용프로그램 실행 횟수
- 응용프로그램 마지막 실행 시각 (FILETIME, 64-Bit Timestmamp)
- 참조 목록 (실행 시 필요한 DLL, SDB, NLS, INI 등의 경로)
- 파일 시스템 시간 정보 (생성, 수정, 접근 시간)을 이용한 통합 분석

### 1.5 프리패치 활용

- 악성코드가 실행될 경우, 프리패치 파일 자동 생성
- 부트 프리패치 파일을 이용해, 부팅 시 로드되는 악성코드 탐지 가능
- 참조 목록을 통해, 로드한 라이브러리, 파일 목록 확인 가능

# 2 로그파일

- 파일시스템의 I/O 혹은 트랜잭션에 대한 로그
- NTFS 파일시스템 로그
  - %SystemDrive%₩$LogFile
  - %SystemDrive%₩$Extend₩$UsnJrnl:$J
- 파일시스템 로그의 장점
  - 특정 기간 동안 일어난 상세한 파일시스템 이벤트 분석 가능
  - 삭제된 파일의 흔적 추적 가능

## 2.1 $LogFile

- 트랜잭션 로그 파일
  - 시스템 비정상 동작을 대비하기 위한 트랜잭션 로그
  - 파일 생성, 삭제, 수정, 파일명 변경, 이동 등의 행위 파악 가능
- 트랜잭션 단위의 로그 기록
  - 파일/디렉터리 생성
  - 파일/디렉터리 삭제
  - 파일/디렉터리 변경
  - MFT 레코드 변경
- 일반적으로 64MB 크기
- PC의 일반적 작업이라면 2~3시간 정도의 로그가 보관

## 2.2 $UsnJrnl

- NTFS 변경 로그
  - 파일이나 디렉터리의 변경 내용 기록
  - 윈도우 7부터 기본 활성화
- 로그에 기록되는 정보
  - 변경된 시간
  - 변경 이유
  - 파일/디렉터리의 이름
  - 파일/디렉터리의 속성
  - 파일/디렉터리의 MFT 레코드 번호
  - 파일의 부모 디렉터리에 대한 파일 참조 주소
  - 보안 ID (Security ID)
  - 레코드의 USN (Update Sequence Number)
- 컴퓨터를 계속 사용할 경우, 보통 1~2일의 로그 저장
- 하루 8시간 정도 사용할 경우, 보통 4~5일의 로그 저장
- $MAX
  - 변경 로그에 대한 메타데이터

- $J
  - 실제 변경 레코드

### 2.3 필기

- $Logfile은 보통 2시간정도만 기록한다. 이것은 롤백을 위한 용도이다.. 하지만 $UsnJrnl은 단순 히 기록하는 것이기 때문에 매우 도움이 된다.
- 경로에 들어가면 ADS 속성( 스트림이 2개이다..)
- 추가 스트림으로 $J인데 winhex로보면 왼쪽 아이콘에 ... 모양이 있다. 이것이 ADS 속성이라는의미이다. 이것을 더블클릭하면 $j는 2기가이다..원래는 32메가 만 쓴느건데..앞에가다 ㅇ으로채워져있다. 이것또한 윈헥스에 (스파스)라고 표시되어있다. 그냥 다 가져온다. Sqlite도구를 쓴다음 select EventTime, Event, Detail, FullPath From LogFile을 친다.
- Detail에서 클러스터 넘버를 쓰면 (24) <--이런식으로써져있는데 이것은 결구 24개의 클러스터를 쓴다는 의미..파일의 크기를 추정해볼 수 있다.
- select TimeStamp, Event, FullPath From USNJRNL
  where Event like %file_Deleted%' 이렇게하면 삭제된파일만나온다.
  where FullPath like '%Desktop%' 하면 바탕화면만나온다.
  where TimeStamp like '%2014-11-27 13:%' 이러면 이날 이후로만나옴
  where TimeStamp >'2014-11-27' and TimeStamp < '2014-11-28'

# 3 바로가기

- 링크 파일(LNK)이라고도 불리며 영문 명칭은 "Windows Shortcut", "Shell Link" 윈도우에만 존재하는 기능으로 파일, 디렉터리 등 객체를 참조하는 파일
- 커맨드라인이 아닌 GUI에서만 동작
- .lnk 확장자를 가짐

### 3.1 저장위치

- 메뉴
  - %UserProfile%₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu
- 바탕화면
  - %UserProfile%₩Desktop

· 사용자의 내 음악(My Music), 내 그림(My Pictures), 내 비디오(My Videos) 폴더
  - %UserProfile%  "%SystemDrive%₩Users₩Public" 하위 폴더 링크

· Send To 폴더
  - %UserProfile% ₩AppData₩Roaming₩Microsoft₩Windows₩SendTo

· 빠른 실행 (Quick Launch) 폴더
  - %UserProfile%₩AppData₩Roaming₩Microsoft₩Internet Explorer₩Quick Launch

· 최근 문서 (Recent)
  - %UserProfile%₩AppData₩Roaming₩Microsoft₩Windows₩Recent

· 응용프로그램 최근 문서 (순서는 레지스트리 MRU에 저장)
  - MS Office: %UserProfile%₩AppData₩Roaming₩Microsoft₩Office₩Recent
  - Hangul: %UserProfile%₩AppData₩Roaming₩HNC₩Office₩Recent

3.2 파일 구조 정보

· SHELL_LINK_HEADER
  - 링크 대상 파일의 속성 (읽기 전용, 숨긴 파일, 시스템, 볼륨 레이블, 암호화, 압축 등)
  - 링크 대상 파일의 생성, 수정, 접근 시간, 크기

· LINKINFO
  - 링크 대상 파일의 크기
  - 링크 대상 파일이 위치한 드라이브 형식
  - 링크 대상 파일이 위치한 드라이브 시리얼 번호
  - 링크 대상 파일의 경로  외장저장장치 흔적

· EXTRA_DATA
  - NetBIOS 이름
  - MAC 주소

3.3 바로가기파일활용

· 자동 생성된 바로가기 파일을 이용해 폴더나 파일의 실행 흔적 분석

· 링크 대상의 위치를 이용해 외장저장장치를 이용한 데이터 이동 흔적 분석

· 애플리케이션 취약점을 악용하는 악성코드일 경우, 실행 흔적 분석

· 바로가기 파일 자체로 침해를 확인하기는 어렵기 때문에 타임라인 분석과 연계 분석

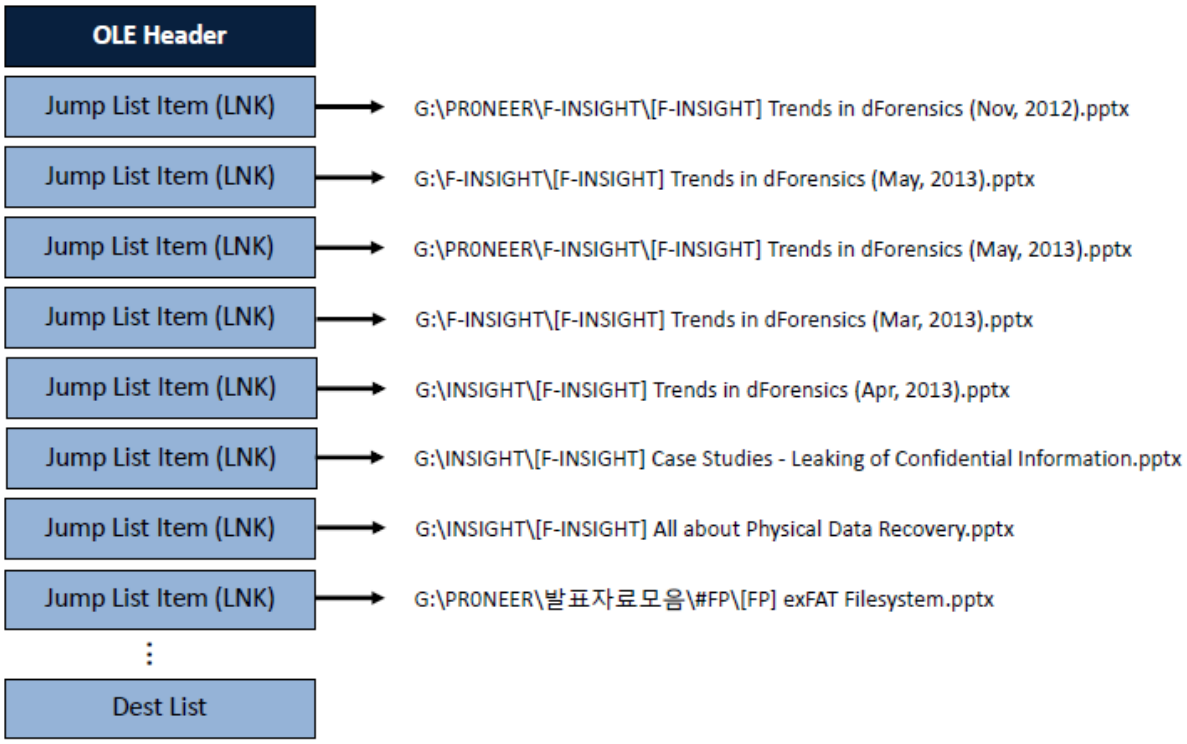
# 4 점프목록

- 윈도우 7부터 새롭게 추가된 응용프로그램 사용 로그로 기본 활성화
- 모든 응용프로그램에 대한 접근 이력 보관

## 4.1 경로

- %UserProfile%₩AppData₩Roaming₩Microsoft₩Windows₩Recent₩
- AutomaticDestinations 폴더
  - 운영체제가 자동으로 남기는 항목
    최근 사용한 목록(Recent)이나 자주 사용되는 목록(Frequent)
- CustomDestinations 폴더
  - 응용프로그램이 자체적으로 관리하는 항목
    작업(Task) 목록
- 점프목록 파일명
  - 각 응용프로그램 별로 고유한 16자리 사용

4.2 포렌식적 의미




- 복합문서 구조의 스트림에 바로가기 파일 형식으로 점프목록 저장
- 바로가기 파일에서 획득 가능한 모든 정보
  - 링크 대상의 속성, 크기, 경로, (생성, 수정, 접근) 시간
  - 링크 대상이 위치한 곳의 드라이브의 형식, 시리얼 번호, NetBIOS 이름, MAC 주소

4.3 점프목록 활용

- 윈도우 7 기본 활성화
- 최근 접근 문서(Recent)나 UserAssist 키보다 더 많은 정보 포함
- 사용자가 직접 삭제하지 않는 이상 운영체제 설치 시부터 지속적으로 로그 저장
- 악성 파일 실행 흔적
- 외장저장장치 파일 열람 흔적
- 웹 사이트 접속 이력

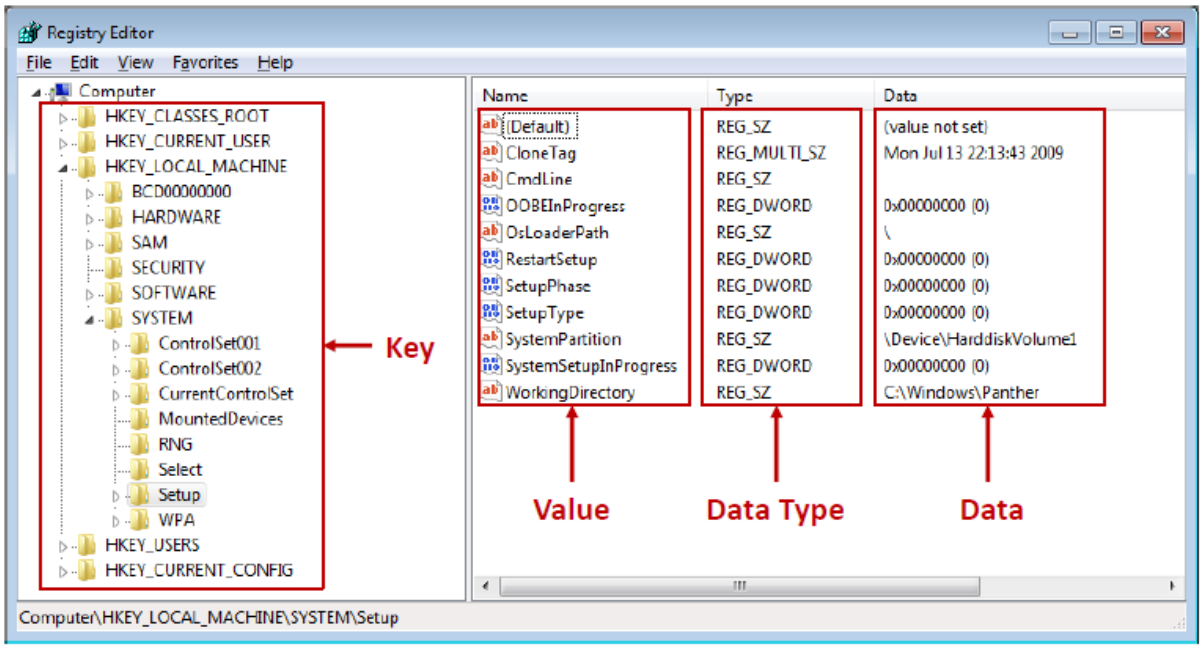
# 5 레지스트리

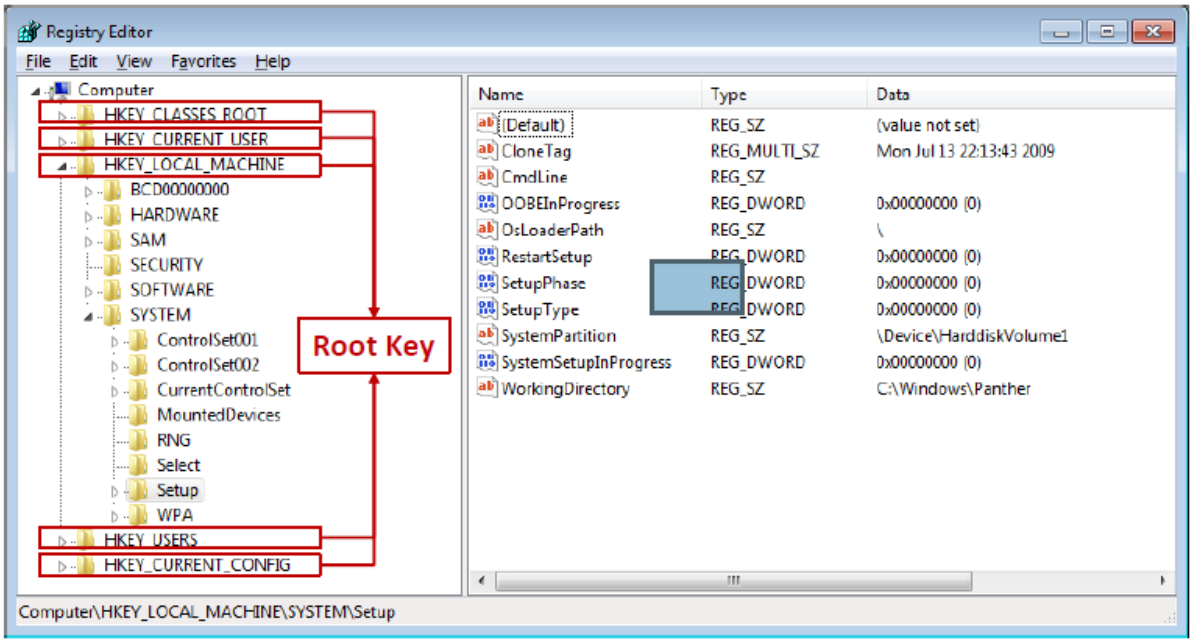
- 윈도우 운영체제에서 운영체제와 응용프로그램 운영에 필요한 정보를 저장하기 위해 고안한 계층형 데이터베이스 (http://support.microsoft.com/kb/256986)
- 부팅 과정부터 로그인, 서비스 실행, 응용프로그램 실행, 사용자 행위 등 모든 활동에 관여함
- 윈도우 3.11, 9x, Me, NT, 2000, XP, 2003, Vista, 2008, 7, 2012, 8 에서 사용

## 5.1 필요성

- 윈도우 시스템 분석의 필수 요소
  - 운영체제 정보, 사용자 계정 정보, 시스템 정보, 응용프로그램 실행 흔적, 최근 접근
- 문서 등
  - 자동 실행 항목(Autoruns), 악성코드 탐지, 저장장치 연결 흔적 등
- 사용자/시스템/저장매체 사용 흔적 분석 추가적인 포렌식 분석 대상 선별

## 5.2 구조






| 레지스트리 경로 | 하이브 파일 경로 |
|---|---|
| HKEY_LOCAL_MACHINE\BCD00000000 | {Boot Partition}\Boot\BCD |
| HEKY_LOCAL_MACHINE\COMPONENTS | %SystemRoot%\System32\Config\COMPONENTS |
| HEKY_LOCAL_MACHINE\SYSTEM | %SystemRoot%\System32\Config\SYSTEM |
| HEKY_LOCAL_MACHINE\SAM | %SystemRoot%\System32\Config\SAM |
| HEKY_LOCAL_MACHINE\SECURITY | %SystemRoot%\System32\Config\SECURITY |
| HEKY_LOCAL_MACHINE\SOFTWARE | %SystemRoot%\System32\Config\SOFTWARE |
| HEKY_LOCAL_MACHINE\HARDWARE | Volatile |
| HKEY_USERS\<SID of local service account> | %SystemRoot%\ServiceProfiles\LocalService\NTUSER.DAT |
| HKEY_USERS\<SID of network service account> | %SystemRoot%\ServiceProfiles\NetworkService\NTUSER.DAT |
| HKEY_USERS\<SID of username> | %UserProfile%\NTUSER.DAT |
| HKEY_USERS\<SID of username>_Classes | %UserProfile%\AppData\Local\Microsoft\Windows\Usrclass.dat |
| HKEY_USERS\.DEFAULT | %SystemRoot%\System32\Config\DEFAULT |
| HKEY_USERS\systemprofile | %SystemRoot%\System32\Config\systemprofile\NUSER.DAT |

- "CURRENT"가 들어가는 루트키는 메모리에서만 유지
- "CLASSES_ROOT"도 타 루트키가 링크된 가상 공간
- 실제 하이브 파일로 존재하는 루트키

- HKEY_USERS  Default, NTUSER.DAT
- HKEY_LOCAL_MACHINE  SAM, SECURITY, SYSTEM, SOFTWARE

| 루트키 | 약어 | 설명 |
|---|---|---|
| HKEY_CLASSES_ROOT | HKCR | HKLM₩SOFTWARE₩Classes와 HKU₩<SID>₩Classes 모음 |
| HKEY_CURRENT_USER | HKCU | HKU 아래 사용자 프로파일 중 현재 로그인한 사용자의 하위키 |
| HKEY_LOCAL_MACHINE | HKLM | 시스템에 존재하는 하이브 파일과 메모리 하이브 모음 |
| HKEY_USERS | HKU | 사용자 루트 폴더에 존재하는 NTUSER.DAT 파일의 내용 |
| HKEY_CURRENT_CONFIG | HKCC | HKLM₩SYSTEM₩CurrentControlSet₩Hardware Profiles₩Current의 내용 |
| HKEY_PERFORMANCE_DATA | HKPD | 성능 카운트(레지스트리 편집기를 통해 접근 불가, 레지스트리 함수로만 접근) |

5.3 백업과 로그

· 레지스트리 파일 백업
  - %SystemRoot%₩System32₩config₩RegBack₩

· 레지스트리 로그
  - %SystemRoot%₩System32₩config₩[hive name].LOG
  - %SystemRoot%₩System32₩config₩[hive name].LOG1
  - %SystemRoot%₩System32₩config₩[hive name].LOG2

5.4 볼륨 섀도 복사본

· 비스타 이후부터는 시스템 복원 지점을 위해 VSS 사용

· 특정 시점에 파일, 폴더에 대한 스냅샷 저장

· ₩System Volume Information₩

5.5 침해 아티팩트

· UserAssist
  · NTUSER.DAT₩Software₩Microsoft₩Windows₩CurrentVersion₩Explorer₩UserAssist₩
  · 응용프로그램 사용 로그 – 응용프로그램 종류, 최종 실행 시각, 실행 횟수 등 확인 가능

- http://www.symantec.com/connect/forums/system-tool-malware-or-spyware

- MUICache
  - UsrClass.dat₩Software₩Classes₩LocalSettings₩MuiCache₩
  - MUI (Multilingual User Interface) 다중 언어 지원을 위해 프로그램 이름 캐시
  - 새로운 프로그램 실행 시, 자동으로 리소스 영역에서 프로그램 이름을 추출하여 저장

- LEGACY_*
  - HKLM₩System₩ControlSet00#₩Enum₩Root₩
  - 악성코드가 직접 생성하기 보다는 실행 시 운영체제에 의해 생성되는 키
    윈도우 서비스로 동작하는 악성코드 정보
  - 레지스트리 키 마지막 수정 시간은 악성 서비스의 처음 실행 시간

- Tracing
  - HKLM₩SOFTWARE₩Microsoft₩Tracing
  - 라우팅 및 원격 액세스(Routing and Remote Access) 서비스가 기록하는 추적정보
  - 복잡한 네트워크 장애를 해결할 목적으로 저장

# 6 AV로그

## 6.1 안랩

- 로그 -> (텍스트 인코딩 | 바이너리)
  - %SystemDrive%₩Program Files₩AhnLab₩[ProductName]₩log
- 검역소 -> 바이너리
  - %SystemDrive%₩Program Files₩AhnLab₩[ProductName]₩Quarantine
- 내보내기 기능을 이용해 텍스트로 저장

## 6.2 이스트소프트 알약

- 로그 -> 바이너리(바이너리로 되어있기 때문에 알아볼 수 없다)

- %SystemDrive%₩ProgramData₩ESTSoft₩AlYac₩log

· 검역소 -> 바이너리

- %SystemDrive%₩ProgramData₩ESTSoft₩AlYac₩quarantine

6.3 윈도우 디펜더

· 로그 -> 이벤트 로그와 통합 (윈도우 이벤트와 통합했기 때문에 알아보기 쉽다)

· 검역소 -> 바이너리

- %SystemDrive%₩ProgramData₩Microsoft₩Windows Defender₩Quarantine

6.4 시만텍 엔드포인트 프로텍션 (SEP)

· 로그 -> 텍스트 형식 (시간만 인코딩 되어 알아보기 쉽다)

· - %SystemDrive%₩Program Files₩Symantec₩Symantec Endpoint Protection₩[...]
검역소 -> 바이너리

· - %SystemDrive%₩Program Files₩Symantec₩Symantec Endpoint Protection₩Quarantine

# 7 볼륨섀도복사본

· 윈도우의 시스템 복원 기능으로 XP의 "시스템 복원 지점""이 Vista 이후 변화됨
- 특정 시각의 파일, 폴더 등을 수동 또는 자동으로 복사본(스냅샷)을 생성하는 서비스
- 윈도우 포렌식 분석의 필수 데이터!!!

· VSS 목적
- 볼륨 백업본을 통해 시스템 복원 기능 제공 (이전 버전, 삭제된 파일 복구 등)
- 파일 잠금 문제 회피
- 읽기 전용인 볼륨 섀도 복사본으로 다른 파일의 쓰기 간섭을 회피

· 볼륨 스냅샷 서비스 지원
- 윈도우 Server 2003/2008/2012
- 윈도우 Vista, 7, 8

7.1 경로

· 저장 경로 : %SystemDrive%₩System Volume Information

· 제외 파일 목록 :

HKLM₩SYSTEM₩CurrentControlSet₩Control₩BackupRestore₩FilesNotToSnapshot

### 7.2 생성시점

- 수동 (Vista/7)
- 매 24시간 마다 (Vista)
- 매 7일 마다 (7)
- 윈도우 업데이트 전 (Vista/7)
- 서명되지 않은 드라이버 설치 시 (Vista/7)
- 프로그램에서 스냅샷 API 호출 시 (Vista/7)

### 7.3 획득 가능 정보

- 백업된 시점의 시스템 흔적 (파일시스템, 레지스트리 등)
- 백업 시점 간에 변화된 시스템 흔적 비교

### 7.4 분석과정

1) 라이브 분석이 가능하다면 라이브 분석!!!
2) 이미지를 획득한 경우, EnCase PDE나 Arsenal Image Moutner 사용!!
3) 2번이 가능하지 않을 경우, VHD로 변환 후 [VHD 연결]!!!
4) 스냅샷 목록 및 생성 일자 확인!!!
5) 필요한 파일만 선별 수집하여 분석 진행!!!

# 8 호환성 아티팩트

- 운영체제의 업데이트에 따라 이전 버전과의 호환성 기능 제공
- 호환성 문제의 대부분은 버전에 따라 달라진 API 문제
- 운영체제는 호환성 문제가 있는 API를 탐지한 후, 대체 API로 연결

### 8.1 필기

- 취약한 API는 윈도우즈에서 알아서 제거한다. 하지만 기존의 프로그램은 해당 API를 이용하므로 윈도우즈에서 대체해주어야한다 따라서 그 기록은 로그에 남게되는데 이러한 로그를 분석할 일이 생길것이다.

8.2 응용프로그램 호환성 캐시

- 호환성 문제가 발생했던 응용프로그램의 정보 저장
- 저장경로 -> 레지스트리
  - 키 : HKLM₩SYSTEM₩ControlSet00#₩Control₩Session Manager₩AppCompatCache
  - 값 : AppCompatCache

8.3 응용프로그램 호환성 데이터베이스

- 호환성 데이터베이스 파일 경로
  - %SystemRoot%₩AppPatch(64)₩
- SDB(Shim Database)
  - 호환성 문제가 있는 프로그램 목록과 해결 방안
  - sysmain.sdb
  - drvmain.sdb
  - msimain.sdb
  - pcamain.sdb

8.4 응용프로그램 호환성 플래그

- "이 프로그램이 제대로 설치되었습니다" 선택 시 레지스트리에 정보 저장
  - HKCU₩Software₩Microsoft₩Windows NT₩CurrentVersion₩ AppCompatFlags₩Compatibility Assistant₩Persisted
  - HKLM₩Software₩Microsoft₩Windows NT₩CurrentVersion₩ AppCompatFlags₩Compatibility Assistant₩Persisted
- 프로그램 속성 -> 호환성 탭 설정 변경
  - HKCU₩Software₩Microsoft₩Windows NT₩CurrentVersion₩AppCompatFlags₩Layers
  - HKLM₩SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩AppCompatFlags₩Layers

8.5 최근파일캐시

- 프로그램 실행 시 경로를 임시 저장하기 위한 공간으로 단순 경로 나열
  - %SystemRoot%₩AppCompat₩Programs₩RecentFileCache.bcf
- 캐시 되는 경우
  - 실행 파일 -> 실행 파일로 파생된 경우 (드롭퍼)
  - 다른 볼륨이나 시스템에서 복사(인터넷 다운로드 포함)된 경우

- 필기
  - 인터넷에 파일을 다운받으면 캐시폴더에 다운받고 파일을 옮긴다. 그때의 캐쉬가 여기이다.드로퍼의 리스트도 여기에 저장된다.
  - 결론은 윈도우에서는 호환성문제가 있을 경우 레지스트레 관련된 로그를 저장해 놓는다. 따라서 호환성캐쉬만 보면 눈에 뛸 것이다.

# 9 윈도우WOW64

## 9.1 레지스트리 아티팩트

- 32비트 소프트웨어의 레지스트리 또한 리다이렉트된다.
- 64비트 환경에서 동작한 32비트 악성코드의 흔적이 남음
- HKLM₩SOFTWARE₩Wow6432Node

## 9.2 파일시스템 아티팩트

- 64비트 프로그램을 돌리면 system32 폴더에 저장된다
- 만약 32비트프로그램을 64비트 환경에 돌리면 SyWOW64에 기록이 남는다.
- %SystemRoot%₩SysWOW64
  - %SystemRoot%₩system32 폴더의 리다이렉트 폴더
  - System32 폴더 하위에 접근하는 32비트 프로그램 데이터는 해당 폴더로 리다이렉트
  - %SystemRoot%₩SysWOW64₩config 폴더 조사

# 10 윈도우 문제 보고(WER, Windows Error Reporting)

- XP부터 추가된 기능으로 오류 발생 시 디버깅 정보를 수집하여 보고하는 기능
- MS 파트너(ISV, IHV, OEM 등)일 경우, 보고된 정보 확인 가능
- 발생 빈도가 높거나 심각한 오류는 핫픽스(Hotfix)를 통해 업데이트

## 10.1텍스트 파일 로그 경로

- %SystemDrive%₩ProgramData₩Microsoft₩WER₩ReportArchive
- %UserProfile%₩AppData₩Local₩Microsoft₩Windows₩WER

10.2이벤트로그 경로

- %SystemRoot%₩system32₩winevt₩Logs₩Microsoft-Windows-WER-Diag%4Operational.evtx