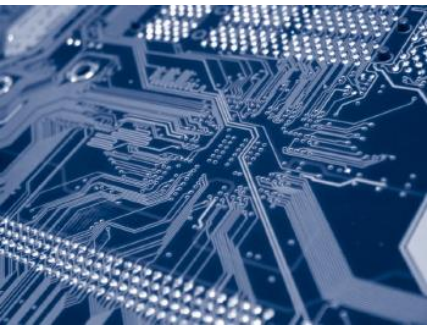
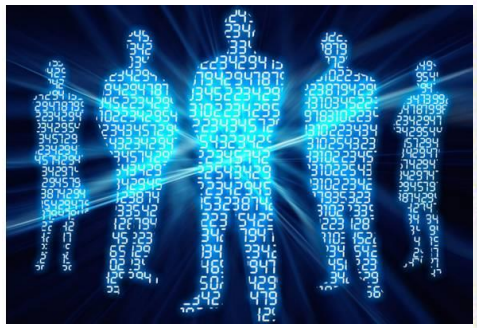
ITL TRAINING LEADERS

Mac OS X 디지털 포렌식 분석기법

보안 | 신뢰


소개

- 국방과학연구소
- 포렌식 인사이트

- DC3 Digital Forensic Challenge 2010

- 관심분야
  - Unix & OS X 포렌식 기술 연구
  - 메모리 포렌식 및 악성코드 분석
  - 상용 소프트웨어 역공학을 통한 취약점 분석

- 개발 도구 : volafox, chainbreaker, walitean

- 사이트 : http://forensic.n0fate.com
- 이메일 : n0fate@n0fate.com

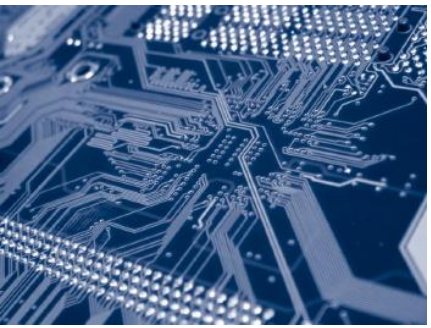
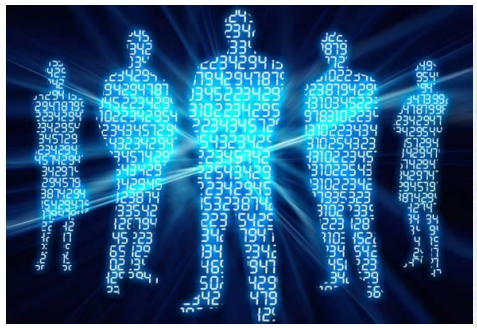
목차

- 증거 수집 방법
  - 디스크 이미징
  - 메모리 이미징

- 증거 분석 방법
  - OS X의 특징
  - 파일시스템 분석
  - OS X 주요 아티팩트
  - 주요 파일 포맷 분석
  - 암호화 데이터 분석
  - 메모리 분석

Mac OS X 디지털 포렌식 분석 기법

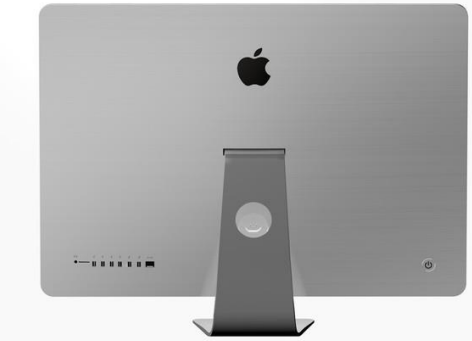
1. 증거 수집 방법


디스크 이미징 - 가장 많은 질문

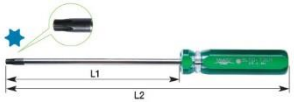
- 맥은 어떻게 뜯나요?
  - 물리적 이미징 방법

- 장치를 뜯지말고 이미징하라는데 방법 좀..
  - 논리적 이미징 방법

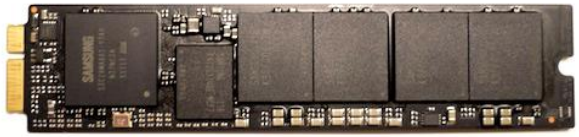


물리적 디스크 이미징

- 애플 제품은 독특한 나사 규격을 사용하고 있음
  - 외부 : 십자 드라이버
  - 내부 : 별 드라이버, 육각렌치



- 하드웨어 인터페이스
  - SATA : 맥북프로, 맥미니, 2013년 iMac, 2012 맥북프로 레티나
  - Mini PCI-Ex : 2013 맥북프로 레티나, 2014 iMac



- 대부분의 맥 제품의 분해를 다루는 사이트 참고
  - ‘www.ifixit.com’에서 제품 분해 방법, 필요한 도구 정보/판매

논리적 디스크 이미징 – TDM

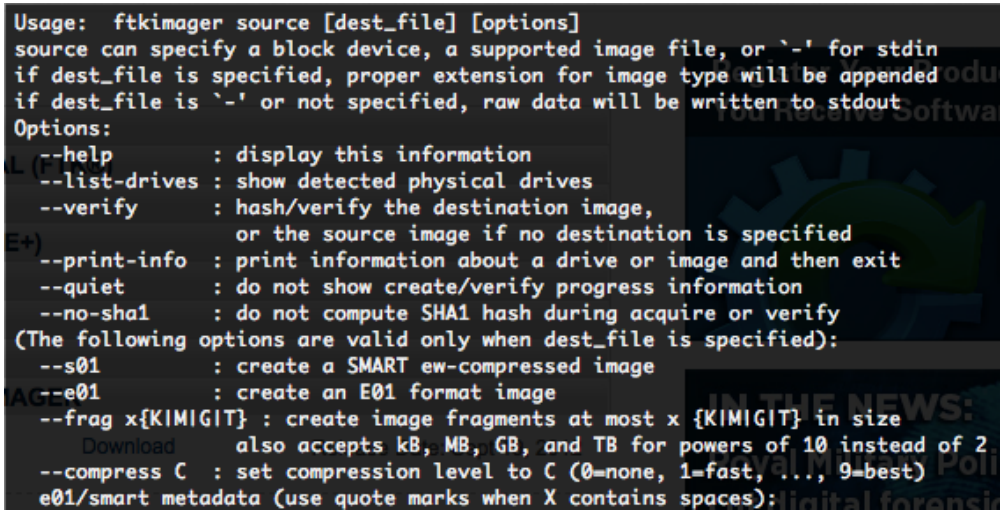
- 타겟 디스크 모드(Target Disk Mode)
- OS X 커널에 문제가 발생할 경우 복구를 위해 사용
- 절차
  - 두 시스템을 Firewire나 Thunderbolt 케이블로 연결
  - 대상 부팅 시 command+T를 누름 -> TDM 연결
  - 분석가 시스템에 각 파티션 마운트
- 장점
  - 별다른 도구 없이 파인더(탐색기)로 분석 가능
- 단점
  - 쓰기 모드가 활성화된 상태로 마운트 -> Write-Blocker 사용
  - ~~분석가의 시스템도 OS X이어야 함.~~ -> HFSExplorer로 가능

논리적 디스크 이미징 – 싱글부팅

- 유닉스의 싱글 사용자 모드 접근 방법을 사용
- 절차
  - 부팅 시 command + s를 누름
  - 쓰기 권한 재마운트
    - ❖기본으로 read-only로 마운트
    - ❖외장 디스크 연결을 위해 필요
  - DD를 이용한 덤프
    - ❖NTFS 쓰기를 지원하지 않음
    - ❖FAT, EXFAT, HFS 포맷한 디스크 사용
- 장점
  - 사용자 비밀번호를 요구하지 않음
  - 하드웨어 분해 없이 이미징
- 단점
  - 무결성 손상

논리적 디스크 이미징 - FTK

- Command Line Versions of FTK imager
- 지원 포맷 : raw, smart ew-compressed, E01, fragments, compress
- E01, S01은 메타데이터 정보 기록
  - Case Number, Evidence Number, Description, Examiner, Notes
- 해시 값 산출



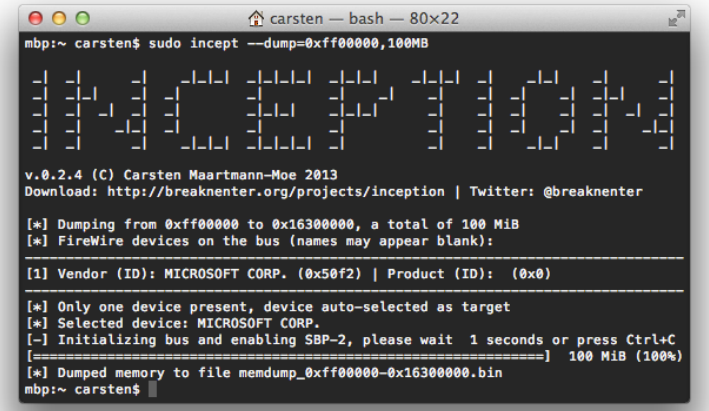
메모리 이미징

- DMA 기반 이미징
  - FW를 이용한 이미징
    - ❖메모리 이미지의 무결성 확보
    - ❖최대 4기가의 메모리 이미징 가능
  - 썬더볼트를 이용한 이미징
    - ❖일부 메모리 조작 가능성만 보여준 상태 (Funterbolt, BH 13')
    - ❖현재 FW 인터페이스를 이용 (동일한 한계)

- 소프트웨어 기반 이미징
  - 현재 가장 높은 효율성을 보임
  - 애플리케이션으로 메모리 이미징을 수행
    - ❖무결성을 해칠 수 있음
    - ❖현재 메모리 이미징을 할 수 있는 유일한 방법

DMA기반 메모리 이미징

- Inception
  - 지원 프로토콜 : Firewire, Thunderbolt, Express Card, PC Card, PCI/PCIe Interface
  - 최대 4기가 메모리 덤프 가능
  - 지원 운영체제 : Windows XP ~ 8, OS X SL ~ Mavericks, Ubuntu 11 ~ 13, Linux Mint 11 ~ 13
  - 전체 덤프 시 간헐적 커널 패닉 발생 (bugs)



소프트웨어 메모리 이미징

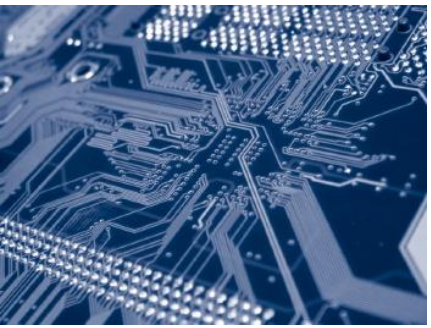
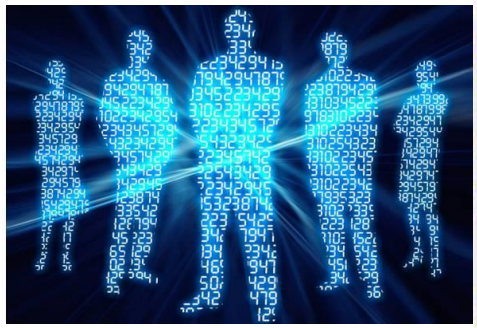
- MacMemoryReader
  - Mountain Lion까지 지원 -> 업데이트 정지
  - 커널 풀 코어 덤프 형태로 이미징 -> 변환 과정 필요

- OSXPMEM
  - Mavericks까지 지원
  - ELF, MACHO, RAW 형태로 이미징

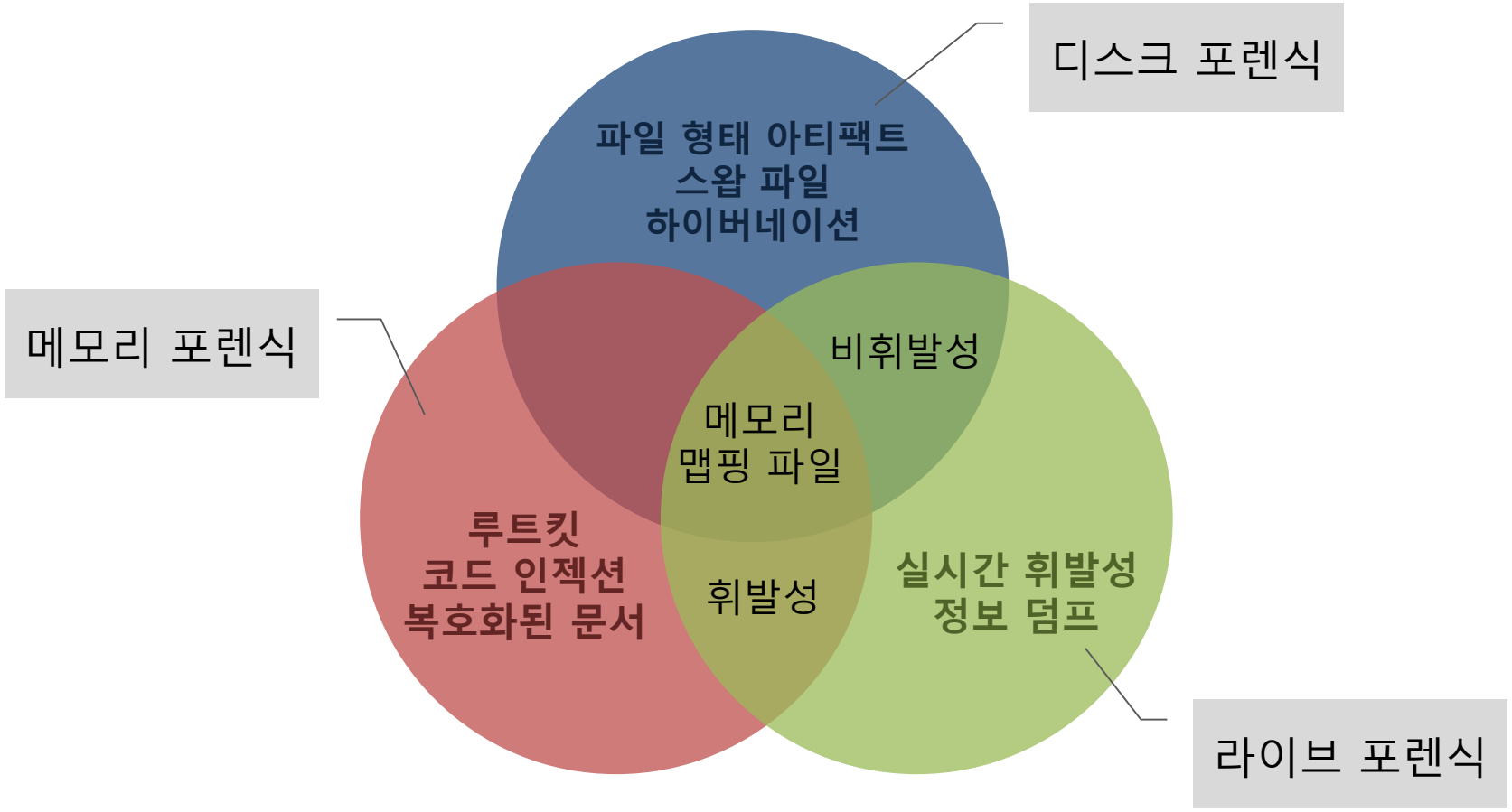
- Memoryze for Mac (Mandiant)
  - Mountain Lion까지 지원
  - RAW 형태로 이미징

Mac OS X 디지털 포렌식 분석 기법

2. 증거 분석 방법


포렌식 방법



OS X 의 특징

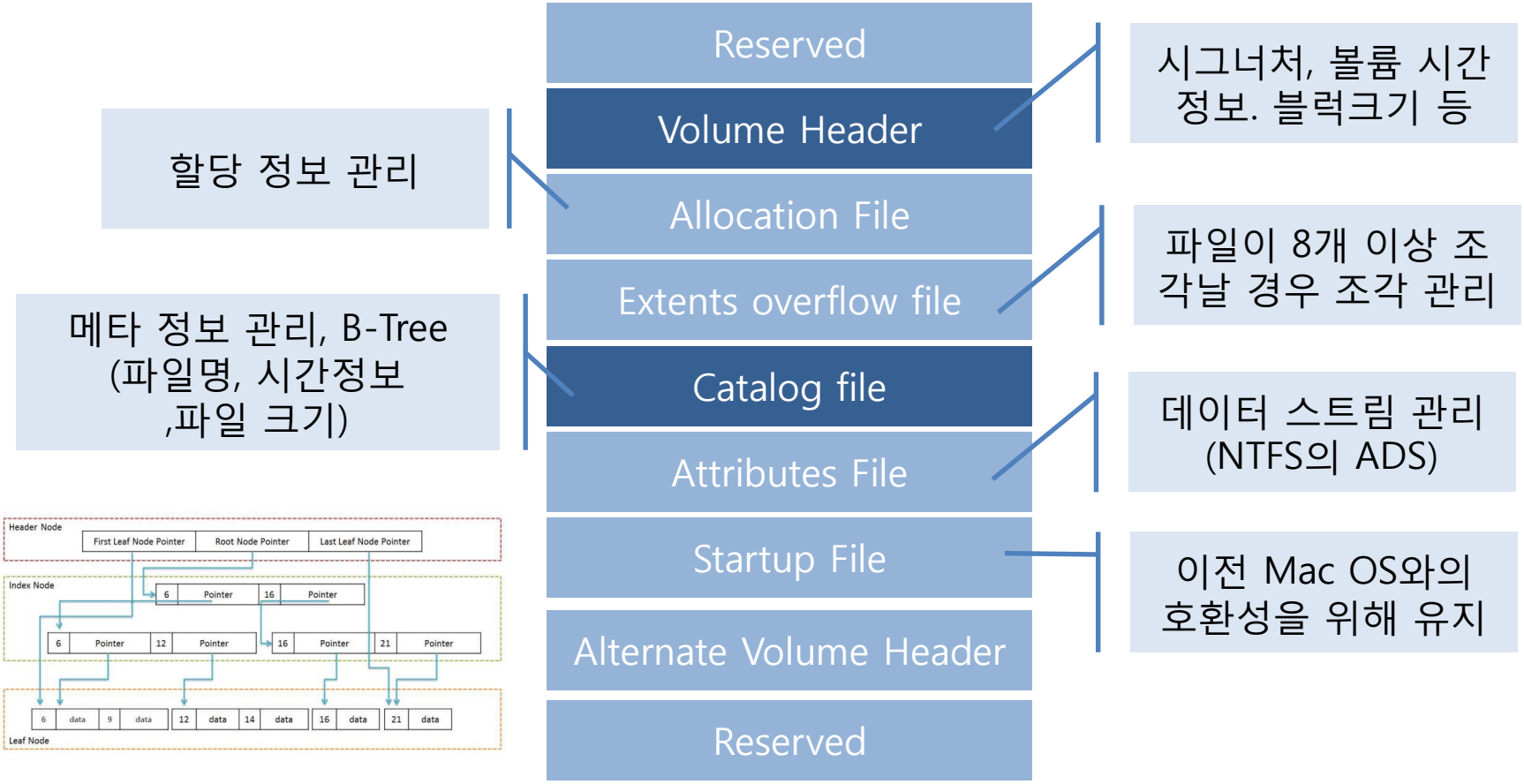
- 2001년 OS X 공개
- XNU 커널 : Mach 커널 + BSD 컴포넌트
  - Mach : 테스크(작업) 관리
  - BSD : 가상 메모리, IPC, 보안, 가상 파일 시스템
- 기존 커널, 컴포넌트의 아티팩트를 가지고 있음.

파일 시스템 분석

- HFS+ (Mac OS Extended)
  - OS X 운영체제 기본 파일 시스템
  - 모든 문자를 UTF16으로 인코딩, 최대 255자의 파일 명
  - Balanced 트리로 디렉터리, 데이터/불량 블록을 관리
  - 저널링 파일 시스템 기능
  - iOS나 아이팟에서도 거의 동일한 파일시스템 사용
  - Technical Note : HFS Plus Volume Format
    - ❖http://dubeiko.com/development/FileSystems/HFSPLUS/tn1150.html

파일 시스템 분석

- 파일 시스템 구조



파일 시스템 분석

- 파일 삭제 시
  - 파일 삭제 시 중간 연결 데이터 정보 삭제
  - 카빙을 통해 데이터 복구 가능

- 휴지통 분석
  - 윈도우와 같이 사용자 파일 삭제 시 임시 저장소로 사용
    - ❖/.Trashes : 루트 권한의 파일 삭제
    - ❖~/.Trash : 특정 사용자에 의해 파일 삭제
    - ❖/Volume/<NAME>/.Trashes : 외장저장장치 내 파일 삭제
  - 파일을 휴지통으로 옮기더라도 MAC의 변경되지 않음

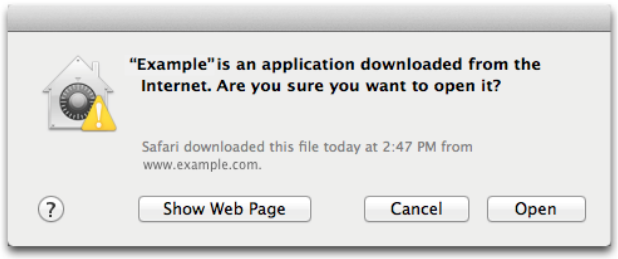
파일 시스템 분석

- 저널링 분석
  - 최근에 일어난 파일 생성/삭제/수정에 대한 기록을 유지
  - 원형 버퍼로 고정 크기를 가짐 -> 용량이 차면 이전 데이터 삭제
    - ❖ 부트 볼륨 : 몇 분 전 데이터
    - ❖ 보조 하드 : 몇 시간 전 데이터
  - 저널은 버퍼가 덮어씌여질 때까지 삭제되지 않음.

- AHJP(Advanced HFS+ Journal Parser), Windows only
  - 저널 정보 추출 도구
  - 파일 명, 생성/수정/접근 시간, 경로, 크기 정보를 가짐
  - 필수 요소
    - ❖ Catalog File : 저널 데이터와 실제 파일 정보 매핑
    - ❖ Journal File : 저널 데이터 분석
    - ❖ Volume Header : 블록 크기 등 파일시스템 기본 정보 추출

파일 시스템 분석

- 확장 메타데이터 속성 (Extensible Metadata Attributes)
  - 애플리케이션의 추가적인 정보를 저장
    - ❖com.apple.metadata : 애플 애플리케이션 메타데이터
      - kMDItemWhereFroms : 다운로드 받은 URL 정보
      - kMDItemDownloadedDate : 다운로드된 시간
      - kMDItemFinderComment : 파인더에서 설정한 사용자 코멘트
    - ❖com.apple.quarantine : Quarantine을 위한 메타데이터
    - ❖com.apple.progress : 다운로드 진행률



- 분석 도구 (OS X only)
  - xattr : 파일의 확장 속성을 읽기/쓰기
  - mdls : 파일의 메타데이터 속성을 목록화 하고 export 제공

OS X 주요 아티팩트

종류	경로	파일 포맷
부팅 로그	/var/log/	Text
감사 로그(OpenBSM)	/var/audit/	BSM
로그인,로그아웃 로그	/var/log/asl/	ASL
USB 접속 로그	/var/log/kernel.log & system.log	Text
애플리케이션 설치 로그	/Library/Receipts/InstallHistory.plist	PList
소프트웨어 업데이트	/Library/Receipts/com.apple.SoftwareUpdate.plist	Binary PList
운영체제 버전	/System/Library/CoreServices/SystemVersion.plist	PList
iCloud 문서	~/Library/Mobile Documents/	-
운영체제 및 소프트웨어 패키지 설치 로그	/var/log/install.log	Text
타임 존	/etc/localtime	zonename
OS X 서비스 동작	/private/var/db/launchd.db/com.apple.launchd/overrides.plist	PList

OS X 주요 아티팩트

종류	경로	파일 포맷
자동 로그인, 애플리케이션 자동 실행	/Library/Preferences/com.apple.loginwindow.plist	PList
방화벽 설정	/Library/Preferences/com.apple.alf.plist	PList
데스크탑 서비스	<anyware>/.DS_Store	DS Format
콘솔 명령어 히스토리	~/.bash_history	Text
파인더 사이드바	~/Library/Preferences/com.apple.sidebarlists.plist	PList
프린터 스풀	/private/var/spool/cups	-
트위터 / 페이스북	~/Library/Twitter, ~/Library/Facebook	PList
연락처	~/Library/Application Support/AddressBook/AddressBook-v22.adcddb	SQLite
이메일	~/Library/Mail/	SQLite, emlx
일정	~/Library/Calendars	SQLite, ics
최근 마운트한 볼륨 정보	~/Library/Preferences/com.apple.finder.plist	PList

OS X 주요 아티팩트

종류	경로	파일 포맷
WiFi 연결 정보	~/Library/com.apple.airport.preferences.plist (SSID, 최종접속시간)	PList
	system.log (airportd, 이전 접속 시간)	Text
	kernel.log (country code, 국가 정보)	Text
Safari 자동 완성 콘솔 ssh 접속 정보 메신저 계정 정보 이메일 계정 정보 WiFi 비밀번호 에버노트 계정 정보	~/Library/Keychain/login.keychain	Keychain Database
타임머신 백업 정보	/var/log/system.log (com.apple.backd)	Text
	/Library/Preferences/com.apple.TimeMachine.plist (타임캡슐 명, 디스크 UUID, 원격지 주소, 계정명)	PList
	/var/db/com.apple.TimeMachine.SnapshotDates.plist (백업 수행 시간)	PList

OS X 주요 아티팩트

종류	경로	파일 포맷
Safari 캐시	~/Library/Caches/com.apple.Safari/Cache.db	SQLite
Safari 히스토리	~/Library/Safari/History.plist	Binary Plist
Safari 쿠키	~/Library/Safari/Cookies.plist	Binary Plist
Safari 세션	~/Library/Safari/LastSession.plist	Binary Plist
데스크탑 화면	~/Library/Preferences/com.apple.desktop.plist	Plist
디스크 관리자 로그	~/Library/Logs/DiskUtility.log	Text
최근 오픈 파일	~/Library/Preferences/com.apple.recentitmes.plist	Plist
애플리케이션 상태 저장	~/Library/Saved Application State/	-
스팟라이트	/.Spotlight	-
로그인 세션정보	/var/run/utmpx	user accounting database

주요 파일 포맷 - SQLite

- SQLite
  - 단일 파일 형태의 포터블 데이터베이스
  - 저널링 기능 지원 : Rollback, WAL(Written-Ahead Log)
    - ❖저널링 분석을 통해 최근 사용자의 행위 유추 가능
  - 페이지 단위로 관리 (헤더에 정의)
  - B-Tree 운용

- 포렌식적으로 유용
  - 개인 정보를 보관하는 주요 애플리케이션에서 사용
    - ❖연락처, 캘린더, 이메일, 사파리, 트위터 등
  - 스키마 정보만 있으면 단일 페이지 해석 가능
    - ❖비할당 영역에서 페이지만 획득하더라도 자료형 기반 분석 가능

주요 파일 포맷 - SQLite

- SQLite Database Browser
  - GUI 기반 SQLite 관리 프로그램 (멀티플랫폼 지원)
  - 저널링 데이터(WAL) 분석 불가능
  - 사용자 쿼리 작성 가능
  - 로깅 기능
    - ❖사용자 쿼리 로깅
    - ❖애플리케이션 자동 쿼리 로깅

- walitean (WAL Analyzer for SQLite)
  - Python 기반 WAL 분석 도구
  - WAL : 레코드 복구 가능하나 컬럼 식별 불가
  - WAL + DB : WAL 레코드 복구 및 컬럼 식별

주요 파일 포맷 - PList

- Property List (PList)
  - XML 형태의 정보 저장 파일
  - Name/value, list, dictionary 등 다양한 데이터형 제공
  - OS X 운영체제 및 애플리케이션 설정 정보 저장

- Binary PList
  - Binary XML과 유사
    - ❖PList에 비해 파일 크기 감소, 성능 저하
    - ❖바이너리 정보를 저장할 경우 주로 사용
  - plutil과 같은 도구를 이용하여 PList화 시킨 후 분석 필요

주요 파일 포맷 - PList

- plutil (property list utility)
  - OS X 기본 설치 도구 -> OS X에서만 사용 가능
  - PList 파일 검증, 포맷 변환, 값 추가
    - ❖지원 포맷 : XML, JSON, BINARY

- ccl-plist
  - python 라이브러리, 운영체제 독립적
  - 라이브러리 형태이기 때문에 사용에 학습이 필요함.
  - https://code.google.com/p/ccl-bplist/

주요 파일 포맷 - OpenBSM

- Basic Security Model
- Sun BSM의 보안 감사 API와 파일 포맷을 오픈소스화
  - BSD 6.2 이상, Mac OS 10.6 이상

- 조사에 필요한 여러가지 정보를 보유
  - 사용자 로그인/로그아웃, SSH 로그인/로그아웃, 소프트웨어 설치 시 계정 인증 성공/실패, 사용자 계정 생성/제거

- 위치 : /var/audit
  - 파일 명 규칙 : 'YYYYMMDDHHMMSS. YYYYMMDDHHMMSS'
  - 현재 작성 중인 파일은 종료시간에 'not_terminated' 기록

주요 파일 포맷 - OpenBSM

- praudit (OS X only)
  - 감사 파일의 내용을 출력해주는 도구
  - XML 파일로도 출력 가능

```
# praudit  20131024072553.20131115053015
...
header,146,11,user authentication,0,Fri Nov 15 14:19:55 2013, + 462 msec
subject,chainbreaker,root,staff,root,staff,11957,100004,11957,0.0.0.0
text,Verify password for record type Users 'chainbreaker' node '/Local/Default'
return,success,0
trailer,146
...
```

2013년 10월 15일 금요일 14시 19분 55초 사용자 권한 관련된 이벤트(header)이다. chainbreaker 사용자가 root로 권한 상승을 시도(subject)한다. 권한 상승은 성공(return)하였으며, 이에 대한 설명(text)로 "Verify password for record type Users 'chainbreaker' node '/Local/Default'"를 남긴다.

- 기타 상용 도구
  - Audit Explorer for Mac

주요 파일 포맷 - ASL

- Apple System Log
- OS 10.4 이상에 있는 바이너리 형태의 시스템 로그

- 조사에 필요한 여러가지 정보를 보유
  - 방화벽, 로그인 정보, 프로그램 에러, 네트워크 정보, 설치 정보, 시스템 부팅/재부팅/종료, 권한 상승

- 위치 : /var/log/asl

- 파일 명 규칙
  - 현재 로깅 데이터 : 'YYYY.MM.DD.[UID]/[GID].asl'
    - ❖ UID/GID 별로 분류
  - 백업 데이터 : 'BB.YYYY.MM.DD.[GID].asl'
    - ❖ 매달 한번 백업 수행
  - 에러 메시지 : 'AUX.YYYY.MM.DD'
    - ❖ 에러 발생 당시 스택 상태 등 에러 프로세스 주요 정보 저장

주요 파일 포맷 - ASL

- syslog
  - 애플 시스템 로그 도구 (OS X에서만 사용 가능)
    - ❖ syslog –f filename
  - 주요 기능
    - ❖ 이벤트 유형별 필터링 가능
    - ❖ 정규 표현식 형태의 필터링 가능
      - 특정 프로세스, 특정 시간, 특정 이벤트
  - 출력 타입 : raw, standard, bsd, xml

- ccl-asl
  - Python 기반의 분석 도구
  - ASL 파일을 해석하여 전시
    - ❖ 수행 시간, 호스트, PID, 참조 프로세스, 수행한 내용
  - 로그인 타임라인 구성 (OSX_asl_login_timeline.py)
    - ❖ 시간, 로그인 여부, 사용자 명, 세부 정보

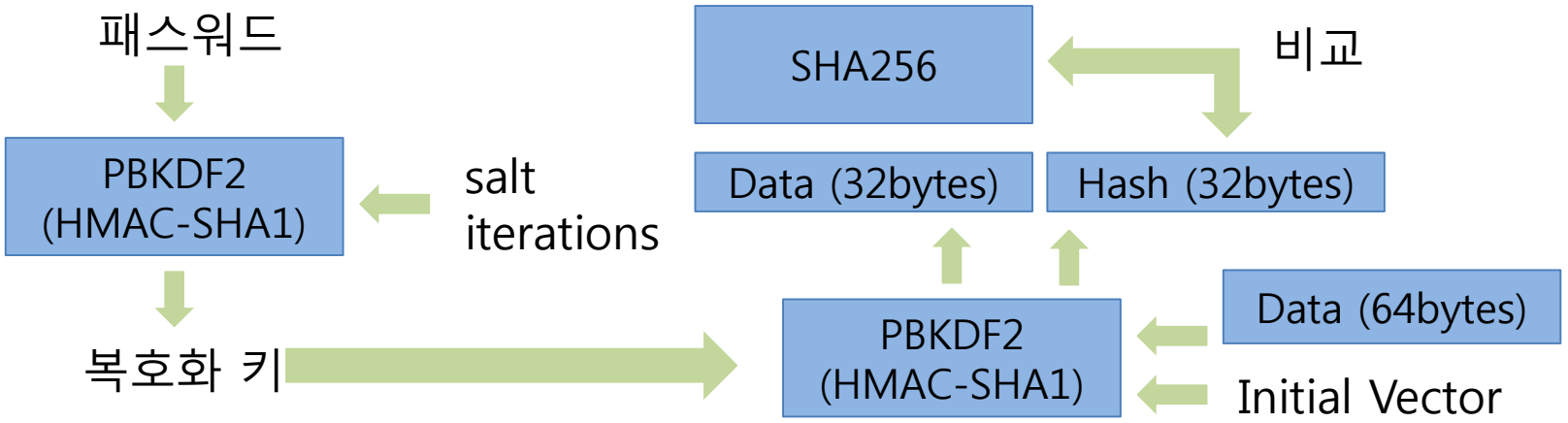
주요 파일 포맷 - iWork

- 패키지 형태의 문서 구조
  - 문서 명을 디렉터리 명으로 내부 데이터를 파일로 관리
  - 각 슬라이드, 첨부 이미지, 영상을 별도 파일로 보관

- 키노트 파일 구조
  - 이미지, 영상 : Data 디렉터리
    - ❖mt[X]xxxxjpg : 마스터 슬라이드 미리보기 이미지
    - ❖st[X]-xxx.jpg : 사용자가 작성 슬라이드 미리보기 이미지
    - ❖pasted-image-xxx : 사용자가 로드한 이미지
  - 슬라이드 : Index.zip 내에 iWA(iWork Archive) 파일로 존재
    - ❖Metadata.iwa : 최종 편집 지점, 슬라이드 미리보기와 문서 연결
    - ❖MasterSlide-X.iwa : 마스터 슬라이드 정보
    - ❖Slide-X.iwa : 사용자 작성 슬라이드 (문자열, 이미지 맵핑 정보 등)

주요 파일 포맷 - iWork

- 파일 암호화
  - AES-128 with PKCS7
  - 패키지 내에 모든 파일을 암호화
    - ❖.iwph : 패스워드 힌트 (텍스트 메시지)
    - ❖.iwpv2 : 패스워드 검증 데이터
      - Iterations, salt(16bytes), Initial Vector(16bytes), Data(64bytes)



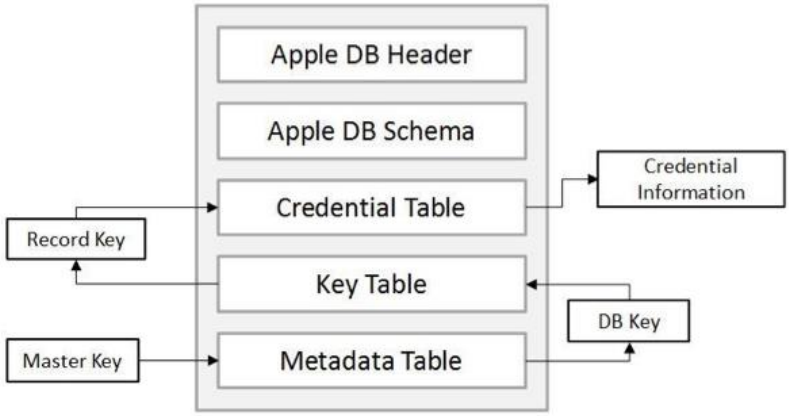
암호화 데이터 분석 - 키체인

- 키체인
  - 애플의 패스워드 관리 시스템
  - 다양한 기밀 정보를 암호화하여 보관
    - ❖패스워드 : 웹사이트, FTP, SSH, 네트워크 공유, 무선 네트워크, 그룹웨어 애플리케이션, FileVault, iWork 문서
    - ❖개인키, 인증서, 보안 노트 등
  - 데이터 암호화를 위해 3DES 사용

- OS X Lion에서부터 키체인에 별도 인증 불필요
  - 로그인 시, 마스터 키를 생성하여 메모리에 보관
  - 키체인 접근 시, 마스터 키로 데이터베이스 키를 해독
  - 데이터베이스 키를 메모리에 보관하여 필요한 정보를 복호화

암호화 데이터 분석 - 키체인

- 키체인 구조
  - 키체인은 애플 데이터베이스 포맷으로 되어 있음.
  - SQLite와 같이 단일 파일 형태
  - 별도의 스키마는 없으며, 테이블 타입 따라 구조 변경
    - ❖데이터베이스 키는 RECORD_METADATA에 정의
    - ❖레코드에 대한 개별 키는 RECORD_SYMMETRIC_KEY에 정의
      - 데이터베이스 키로 복호화 (3DES)
  - 개별 키로 각 레코드 복호화



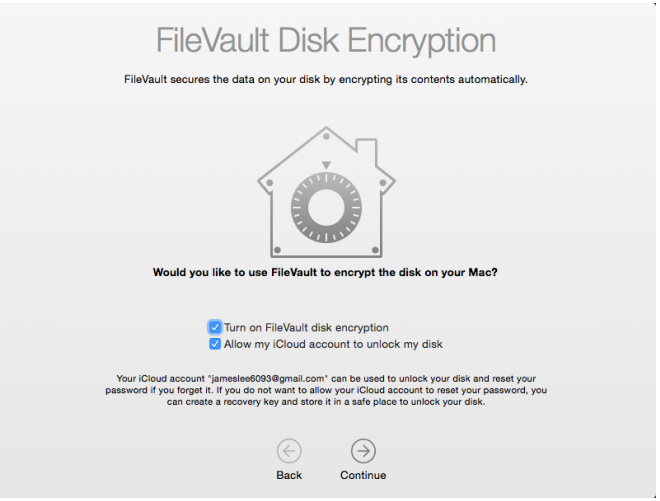
암호화 데이터 분석 - 키체인

- 키체인 분석 방법
  - 도구
    - ❖메모리 포렌식 : volafox, volatility
    - ❖키체인 분석 도구 : chainbreaker
  - 절차

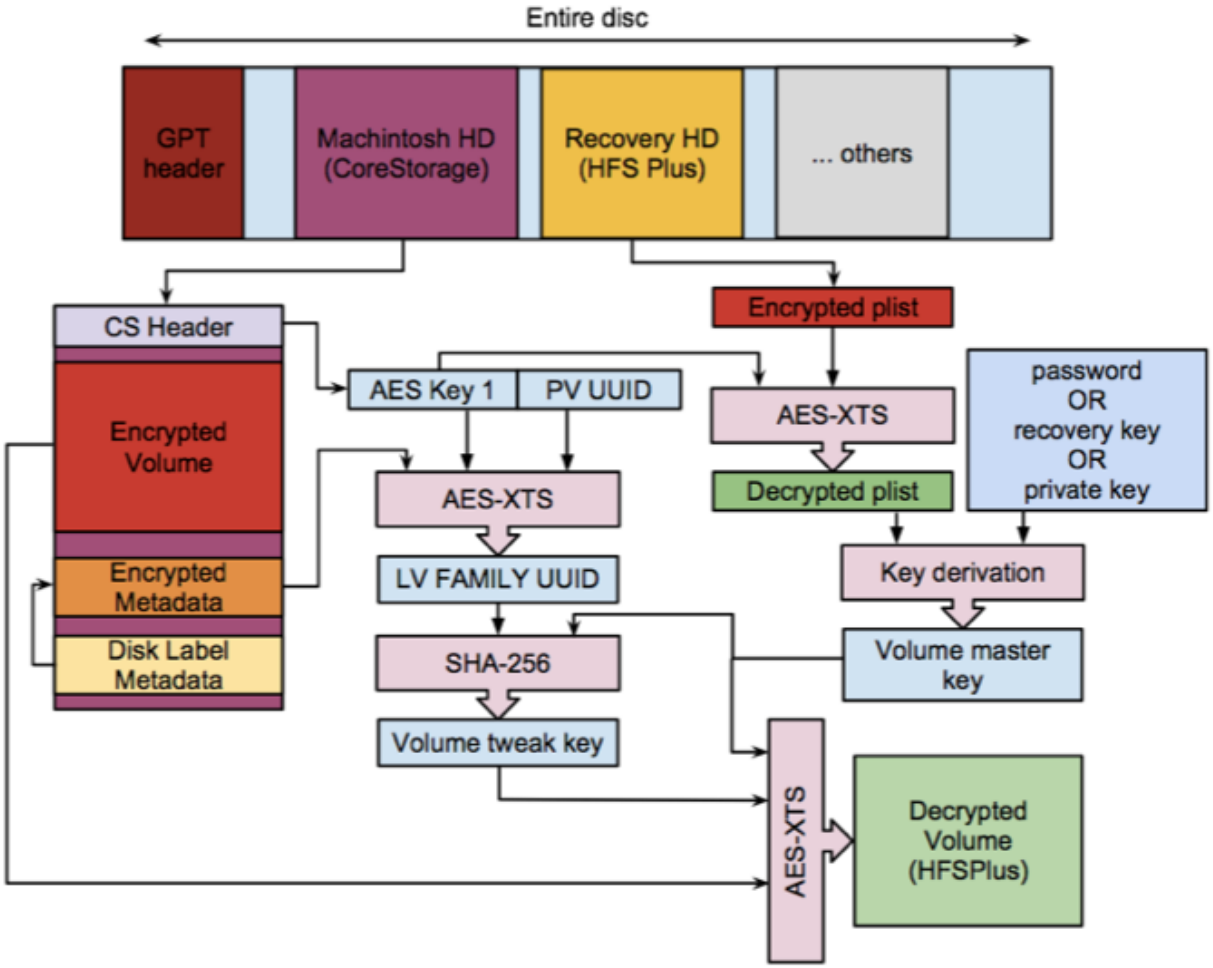
VOLATILITY

암호화 데이터 분석 - FileVault

- 디스크 암호화 (FileVault 2)
  - OS X Lion에서 공개된 디스크 암호화 기술
    - ❖Snow Leopard까지는 홈 디렉터리만 암호화(FileVault)
  - 애플 파일시스템 컴포넌트(CoreStorage)에서 암호화
    - ❖부트 캠프는 암호화할 수 없음
  - AES-XTS 암호화 (128bit)
  - Yosemite에서는 기본 활성화
  - 인증 방법
    - ❖사용자 패스워드
    - ❖36자리 복구 키



암호화 데이터 분석 - FileVault



암호화 데이터 분석 - FileVault

- Joachim Metz의 libfvde 프로젝트
  - Linux, OS X를 지원
  - 요구 사항
    - ❖복구 파티션의 Encrypted.WipeKey.plist
    - ❖패스워드 또는 복구 키
  - 공식적으로 Mountain Lion 까지 지원
    - ❖Mavericks에서는 지원되지 않음

- OS X 자체 기능 활용
  - hdiutil : 디스크 이미지를 검증하고 수정
    - ❖이미지를 시스템에 읽기 전용으로 연결하는데 사용
  - diskutil : 로컬 디스크를 분석
    - ❖퓨전 드라이브나 FileVault를 설정
    - ❖암호화 해제를 위해 사용

메모리 분석

- 라이브 분석의 효용성 저하
  - 메모리 분석으로 대체 가능
  - 공격자의 지능화된 수법으로 라이브 데이터 변조 가능
  - 악성코드 공격 사례 증가

- 침해 대응에 유용한 여러 기능 내장
  - 신뢰성 있는 휘발성 정보 수집
  - 은닉 기능 탐지 및 의심가는 프로세스, KEXT 추출

- 2010년 volafox 이후로 여러 분석 도구 등장
  - 현재 volatility framework, Memoryze for Mac 지원

메모리 분석

기능		volafox	volatility	Memory for Mac
운영체제 기본 정보		O	O	X
휘발성 정보 (프로세스, 네트워크 등)		O	O	O
프로세스 메모리 영역 덤프		O	O	O
프로세스 덤프		O	O	X
KEXT 덤프		O	O	X
KEXT 스캔		O	O	X
오픈파일 덤프		X	O	X
은닉	프로세스, 네트워크	O	O	X
	시스템 콜/Mach Trap 후킹	O	O	X
	TrustedBSD	O	O	X
	권한 상승	O	O	X
	인라인 함수 후킹	△	X	X
키체인 마스터 키 덤프		O	O	X

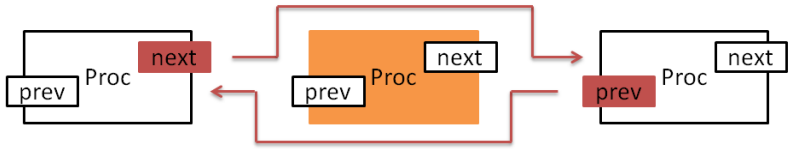
메모리 분석 - 루트킷 사례

- 루트킷은 다음 임무를 수행
  - 프로세스 은닉
  - 특정 명령어로부터 은닉 : kextstat, netstat, grep, w, who
  - Finder에서 루트킷 관련 파일 은닉
  - 루트킷 콘솔 프로세스 권한 상승

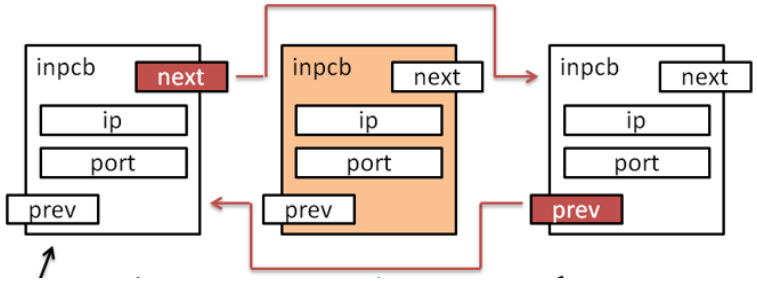
기법	라이브	메모리	디스크
프로세스 은닉	X	O	X
명령어로부터 은닉	X	O	X
파일 은닉	X	O	O
권한 상승	X	O	X

메모리 분석 - 루트킷 사례

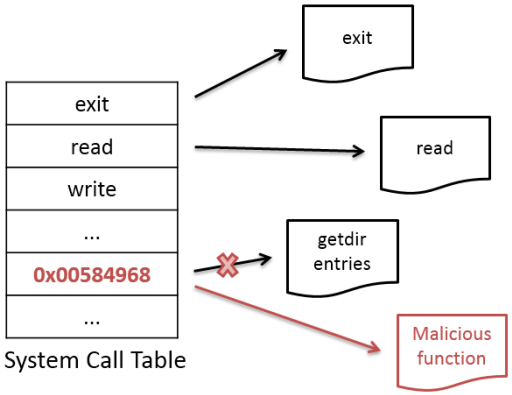
프로세스 DKOM



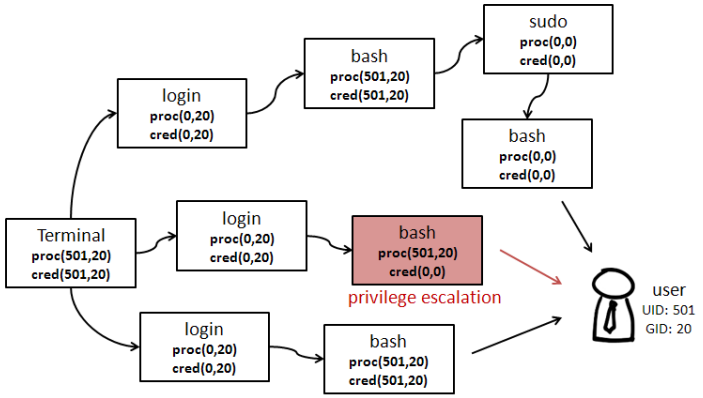
네트워크 DKOM



시스템 콜 후킹



프로세스 권한 상승



메모리 분석 - 루트킷 사례

```
n0fate@n0fate-ui-MacBook-Pro: ~/volafox$ python vol.py -i rubilyn.flat -o systab | grep "hooked"
222          8    0     0              0xFFFFFF7F80BFF41D 0xFFFFFF7F80BFF41D 0xFFFFFF80005CC110      0x00000000       1       32 Maybe hooked
344          4    0     0              0xFFFFFF7F80BFF2EE 0xFFFFFF7F80BFF2EE 0xFFFFFF80005CC0D0      0x00000000       6       16 Maybe hooked
397          3    0     0              0xFFFFFF7F80BFFA7E 0xFFFFFF7F80BFFA7E 0xFFFFFF80005CC0C0      0x00000000       6       12 Maybe hooked
427          4    0     0              0xFFFFFF8000300710 0xFFFFFF8000300710 0xFFFFFF80005CC240      0x00000000       6       20 Maybe hooked
n0fate@n0fate-ui-MacBook-Pro: ~/volafox$ python vol.py -i rubilyn.flat -o mtt | grep "hooked"
n0fate@n0fate-ui-MacBook-Pro: ~/volafox$
```

시스템 콜 후킹 탐지

```
n0fate@n0fate-ui-MacBook-Pro: ~/volafox$ egrep -e "222|344|397" /usr/include/sys/syscall.h
#define SYS_getdirentriesattr 222
#define SYS_getdirentries64 344
#define SYS_write_nocancel 397
n0fate@n0fate-ui-MacBook-Pro: ~/volafox$
```

```
[+] Kernel Extention List
OFFSET(P)  INFO KID                                                KEXT_NAME          VERSION REFER_COUNT        REFER_LIST
        ADDRESS    SIZE HDRSIZE          START_PTR STOP_PTR
0x314651D0    1  92                                  com.atc-nycorp.devmem.kext          3.0.2           0 0xFFFFFF800818A080 0
xFFFFFF7F80C09000   12288       0 0xFFFFFF7F80C0AAFF 0xFFFFFF7F80C0AB4B
0x2E52C7C8    1  91                                  com.hackerfantastic.rubilyn           1           0 0xFFFFFF8006A0FF20 0
xFFFFFF7F80BFE000   20480       0 0xFFFFFF7F80BFFE6A 0xFFFFFF7F80BFFEB6
0x134D1CD0    1  90                                  com.apple.driver.AppleHWSensor      1.9.4d0           0 0xFFFFFF8008181EC0 0
xFFFFFF7F80C73000   20480       0 0xFFFFFF7F80C75A97 0xFFFFFF7F80C75AE3
0x11CE64B0    1  89                                  com.apple.filesystems.autofs          3.0           0 0xFFFFFF800836DB00 0
xFFFFFF7F80C68000   36864       0 0xFFFFFF7F80C6EFFA 0xFFFFFF7F80C6F046
0x12490020    1  88                                    com.apple.kext.triggers           1.0           1 0xFFFFFF80069C9E00 0
xFFFFFF7F80C63000   20480       0 0xFFFFFF7F80C65BF6 0xFFFFFF7F80C65C42
0x06CE5290    1  87                                  com.apple.iokit.IOUserEthernet      1.0.0d1           0 0xFFFFFF80082DD300 0
xFFFFFF7F80C5A000   24576       0 0xFFFFFF7F80C5CC31 0xFFFFFF7F80C5CC7D
0x08326B10    1  86                                   com.apple.iokit.IOSurface          80.0           0 0xFFFFFF80082DD500 0
xFFFFFF7F80C48000   73728       0 0xFFFFFF7F80C5005C 0xFFFFFF7F80C500A8
```

메모리 분석

라이브 분석

```
   83    0 0xffffff7f80c25000 0xe000     0xe000     com.apple.iokit.IOSerialFamily (10.0.5) <7
   84    0 0xffffff7f80c33000 0xe000     0xe000     com.apple.iokit.IOBluetoothSerialManager (
 3 1>
   86    0 0xffffff7f80c48000 0x12000    0x12000    com.apple.iokit.IOSurface (80.0) <7 5 4 3
   87    0 0xffffff7f80c5a000 0x6000     0x6000     com.apple.iokit.IOUserEthernet (1.0.0d1) <
   88    1 0xffffff7f80c63000 0x5000     0x5000     com.apple.kext.triggers (1.0) <7 6 5 4 3 1
   89    0 0xffffff7f80c68000 0x9000     0x9000     com.apple.filesystems.autofs (3.0) <88 7 6
   90    0 0xffffff7f80c73000 0x5000     0x5000     com.apple.driver.AppleHWSensor (1.9.4d0) <
victims-MacBook-Pro:/ victim$
```

메모리 분석 - 루트킷 사례

권한 상승 탐지

```
170  116    128    0 AirPort Base Sta          victim(501,20)      (501,20) Wed Apr 17 14:08:30 2013
179  116    255    0         mdworker          victim(501,20)      (501,20) Wed Apr 17 14:09:33 2013
192  125    128    0            login            victim(0,20)        (0,20) Wed Apr 17 14:09:55 2013
194  192    128    0             bash          victim(501,20)        (0,0) Wed Apr 17 14:09:55 2013
212    1    128    0            ocspd       _coreaudiod(0,0)        (0,0) Wed Apr 17 14:11:33 2013
220  194    128    0             sudo          victim(0,0)         (0,0) Wed Apr 17 14:12:29 2013
221  220    128    0  MacMemoryReader          victim(0,0)         (0,0) Wed Apr 17 14:12:29 2013
222    1    128    0       taskgated       _coreaudiod(0,0)        (0,0) Wed Apr 17 14:12:29 2013
228  221    128    0            image          victim(0,0)         (0,0) Wed Apr 17 14:12:31 2013
```

```
[+] Unlinked task list
TASK CNT  OFFSET(P) REF_CNT Active Halt          VM_MAP(V) PID    PROCESS USERNAME
65       0x04F031A8       2      1    0 0xFFFFFF80065D29F8 190 rubilyncon   victim
n0fate@n0fate-ui-MacBook-Pro: ~/volafox$
```

은닉된 프로세스 탐지

```
victims-MacBook-Pro:~ victim$ netstat -nat
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  *.88                   *.*                    LISTEN
tcp6       0      0  *.88                   *.*                    LISTEN
tcp4       0      0  *.548                  *.*                    LISTEN
tcp6       0      0  *.548                  *.*                    LISTEN
tcp4       0      0  127.0.0.1.631          *.*                    LISTEN
tcp6       0      0  ::1.631                *.*                    LISTEN
udp4       0      0  *.*                    *.*
udp6       0      0  *.59364                *.*
udp4       0      0  *.59364                *.*
```

The output with *netstat*

```
[+] NETWORK INFORMATION (hashbase)
Proto      Local Address   Foreign Address (state)
tcp          0.0.0.0:22          0.0.0.0:0 808000
tcp          0.0.0.0:88          0.0.0.0:0 8000
tcp      10.211.55.7:22 10.211.55.2:57583 8000
tcp        0.0.0.0:548           0.0.0.0:0 8000
tcp      127.0.0.1:631           0.0.0.0:0 8000
udp      0.0.0.0:64448           0.0.0.0:0 808300
udp      0.0.0.0:54625           0.0.0.0:0 808300
```

The output with *volafox*

네트워크 은닉 탐지

ITL TRAINING LEADERS

Q & A