Mac OS X Memory Analysis

Finding treasure in your memory

n0fate@live.com

Who Am I?

- Kyeongsik Lee
- Experience
  - Past: Researcher at DFRC, CIST
  - Current: Researcher at Agency for Defense Development
- Groups and Associations
  - Member at ForensicInside
  - Administrator at volafox project
  - CTF staff at Codegate 2010, 2011
- Detail Profile
  - http://www.linkedin.com/pub/kyeongsik-lee/28/8a1/328

n0fate@live.com

Contents

- Introduction
- Physical Memory acquisition procedure
- Analysis Physical Memory Image
- Rootkit Detection & Analysis
- Q & A

Memory Forensics

- 메모리 이미지에서 디지털 증거를 획득하는 기술
- 악성코드 분석에도 많이 사용되었음.
  - 메모리에서 Unpacking 된 Malware 추출하여 분석
  - 루트킷 탐지 및 추출
- Tool
  - Windows : volatility, HB Gary Respose
  - Linux : volatilitux
  - Mac OS X, FreeBSD : volafox

Mac OS X

- Mac OS X is a series of Unix-based OS sold by Apple
- KERNEL : Darwin Kernel + FreeBSD = XNU
- I/OKit에서 device driver 관리
  - KEXT : Kernel Extension 으로 Driver Package
    - MACH-O Format을 가지고 있음.
- 실행 파일은 Universal Binaries Format 형태임
  - 다수의 MACH-O File Format 조합으로 이루어짐.

Mac OS X Memory Forensics

- Mac OS X 시스템의 물리 메모리를 이미징하고 분석하는 과정
  - 크게 물리 메모리 추출과 분석으로 나뉨
- Mac OS X 메모리 분석 도구인 volafox의 필요 요소
  - Requirement: Mac Physical Memory Image
  - Optional : Kernel Image (mach_kernel)

Mac OS X 메모리 장치(/dev/mem)가 제거되어 있다. 이를 대체할 수 있는 메모리 추출 기법을 알아보자.

물리 메모리 추출 작업

n0fate@live.com

Imaging Physical Memory

- 수집 방법
  - Firewire Tool
  - ~~/dev/kmem~~
  - Mac Memory Reader
  - Hibernation Image

Imaging Physical Memory

- Firewire(IEEE1394)를 이용한 수집방법
  - Apple의 표준
  - Physical Memory 를 linear하게 이미징
  - 메모리의 무결성 침해를 최소화하는 수집 방법
  - Python Code로 된 메모리 수집 툴을 이용할 수 있음
    - Pyfw : Mac OS X Firewire bindings for Python
    - Cansecwest 2005 : Firewire: all your memory are belong to us

Imaging Physical Memory

- Mac Memory Reader(MMR)를 이용한 수집
  - ATC NY 의 hajime Inoue 가 개발한 Mac Memory Reader를 이용함.
  - 선형 주소로 물리 메모리를 덤프하진 않음
    - 분석 과정에선 이를 선형 주소화하는 작업이 필요함.

ATC-NY
ARCHITECTURE TECHNOLOGY CORPORATION

Mac Memory Reader™

Mac Memory Reader is a simple command-line utility to capture the contents of physical RAM on a suspect computer, letting an investigator gather volatile state information prior to shutting the machine down. Results are stored in a Mach-O binary file for later off-line analysis by the investigator.

Mac Memory Reader is available free of charge. It executes directly on 32- and 64-bit target machines running Mac OS X 10.4 through 10.7 and requires a PowerPC G4 or newer or any Intel processor.

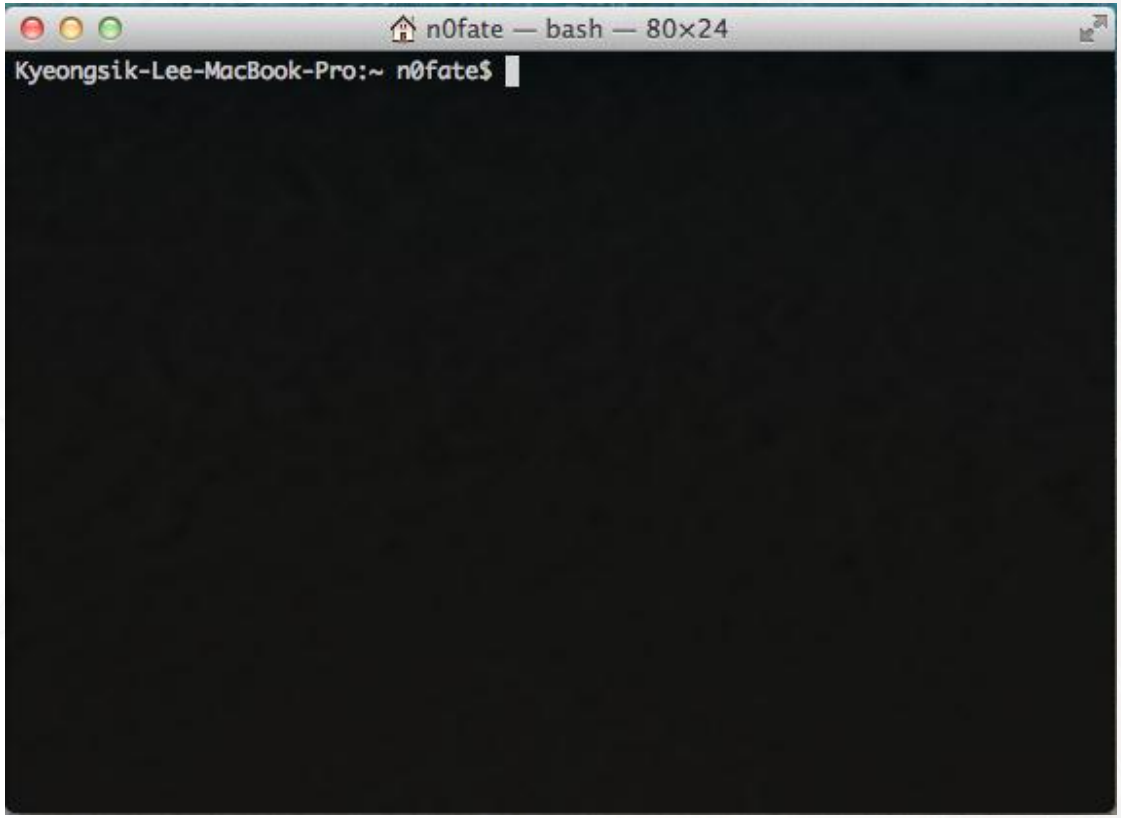
Imaging Physical Memory

- Hibernation Image
  - Macbook Series는 기본적으로 Hibernation 활성화
  - Hibernation Mode는 총 5개가 존재
    - Pmset 명령으로 hibernatemode 옵션에 1을 주어 이미지 생성

모드	내용
0	하이버네이션 비 활성화. Sleep mode, 물리 메모리에 전원을 공급
1	하이버네이션 활성화. 물리 메모리 내용을 하드디스크에 기록
3	Sleep mode, 물리 메모리 내용을 하드디스크에 일부 기록
5	모드 1을 기준으로 이미지 암호화
7	모드 3을 기준으로 이미지 암호화

Example

- Mac Memory Reader(MMR)를 이용한 수집



앞에서 수집한 정보를 이용하여 Volafox를 이용한 물리 메모리 분석 방법을 알아보자.

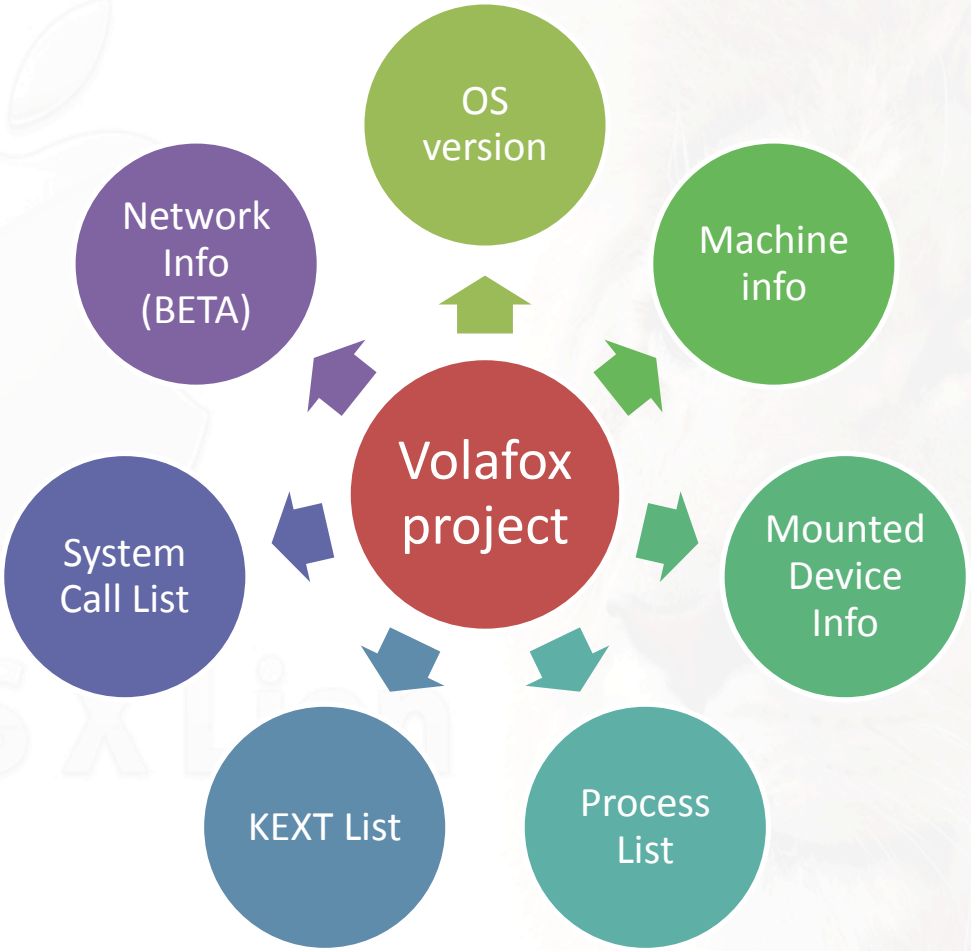
물리 메모리 분석 작업

n0fate@live.com

Analysis Physical Memory Image

- Volafox Project
  - Mac OS X 물리 메모리 이미지에서 정보 추출
  - Site: http://code.google.com/p/volafox
  - Linear format memory image를 입력으로 받음.
  - Kernel image의 symbol을 기반으로 정보를 추출
  - CPU Type: Intel x86
  - Supported OS
    - Mac OS X Leopard(10.5)
    - Mac OS X Snow Leopard(10.6)
    - Mac OS X Lion(10.7) – MMR의 결과는 아직 분석 안됨.

Volafox project



n0fate@live.com

Pre-processing for volafox

- 물리 메모리 이미지 준비
  - 물리 메모리 이미지를 linear format으로 변경
    - 심볼 기반 정보 추출이기 때문에 OS 가 접근하는 물리메모리와 동일 해야함.
  - 추출 방식에 따른 변경 방법
    - Firewire : 해당 사항 없음.
    - MMR : flatten.py 로 변경 (volafox rev.34부터 가능)
    - Vmware VMEM Image : 해당 사항 없음.
    - Hibernation Image : 지원 안함 (현재 분석 중)

Pre-processing for volafox

- 물리 메모리 이미지의 버전 정보 추출
  - 커널 이미지가 없을 경우 volafox DB에 있는 overlay를 사용하기 위함
    - Volafox DB : 각 버전 별 커널 이미지의 심볼 정보를 추출하여 보관하고 있는 파일로 overlays 디렉터리에 존재.
  - Imageinfo.py 로 버전 정보 추출
    - Volafox rev.24 부터 가능 (Intel x86)
    - 시스템 시작 시점에 0x2000에 맵핑하는 테이블 정보 이용
    - “Catfish “ Signature로 위치를 찾아 정보 추출
    - Kernel version string, KEXT Listing Pointer, osversion
  - 해당 overlay가 없을 경우 overlay_generator.py로 생성

Analysis Time!

- Volafox는 다음 인자를 받아 수행
  - -i : Mac OS X의 linear format image
  - -s : kernel overlay image or kernel image
  - Extract, dump option
    - -o : information
    - -x : process dump
    - -m : kext dump

Analysis Time! - example

- Gathering Information using volafox
  - 기본 정보 추출
  - Bash 프로세스를 덤프하여 사용한 명령어 내역 추출
  - VI를 덤프하여 에디팅 중인 메시지 추출

volafox는 메모리 이미지에서 루트킷을 찾고 이를 덤프하여 분석할 수 있는 환경을 제공한다.

ROOTKIT DETECTION & ANALYSIS

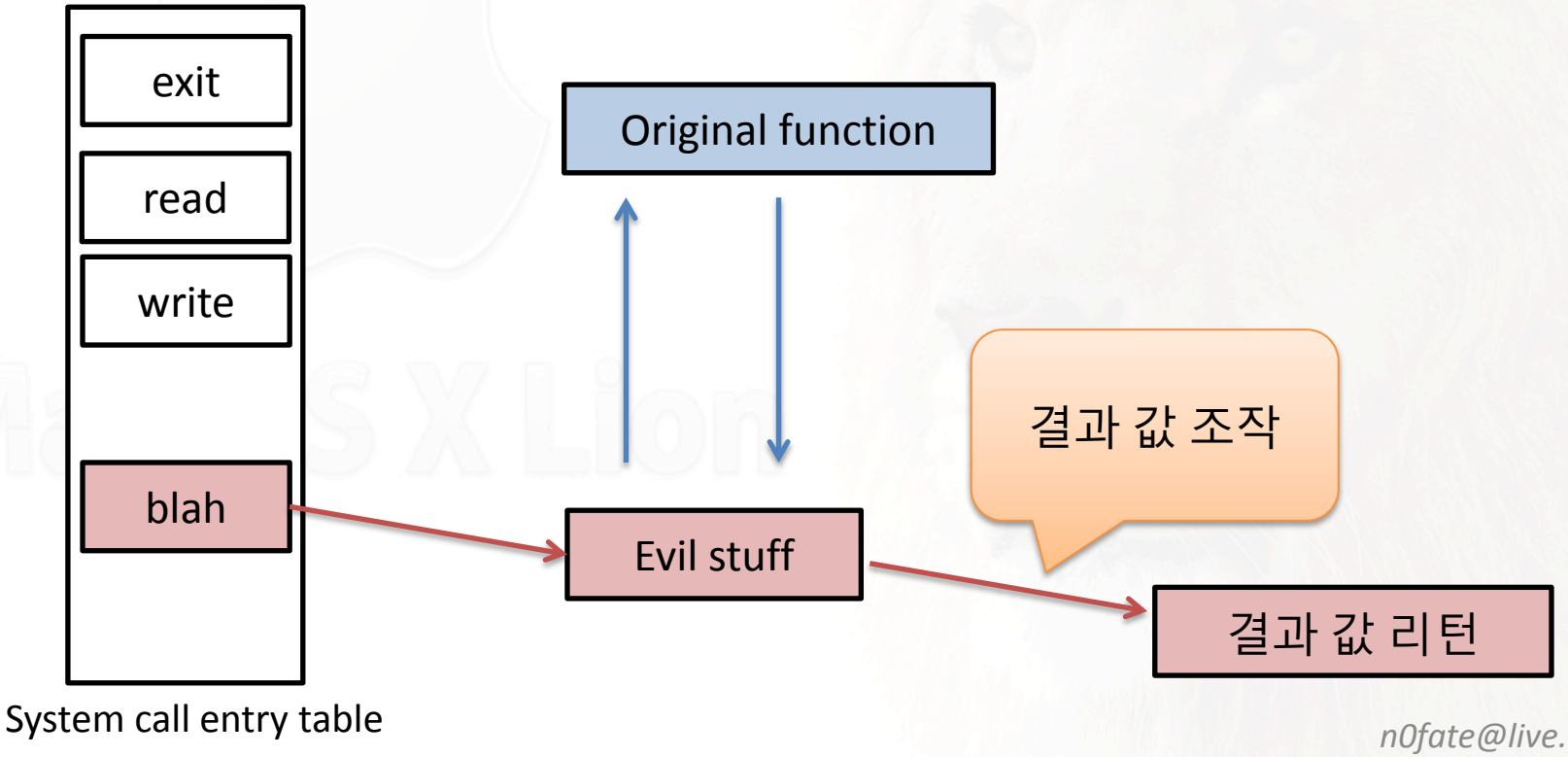
n0fate@live.com

Mac OS X Rootkit

- Mac OS X Rootkit는 KEXT로 동작
  - KEXT 로드 시에 동작하도록 하거나 별도의 Process를 이용해 트리거를 발생해 명령 수행
- Rootkit의 주요 변조 기술
  - System Call Hooking
  - Inline function hooking
  - Direct Kernel Object Manipulation

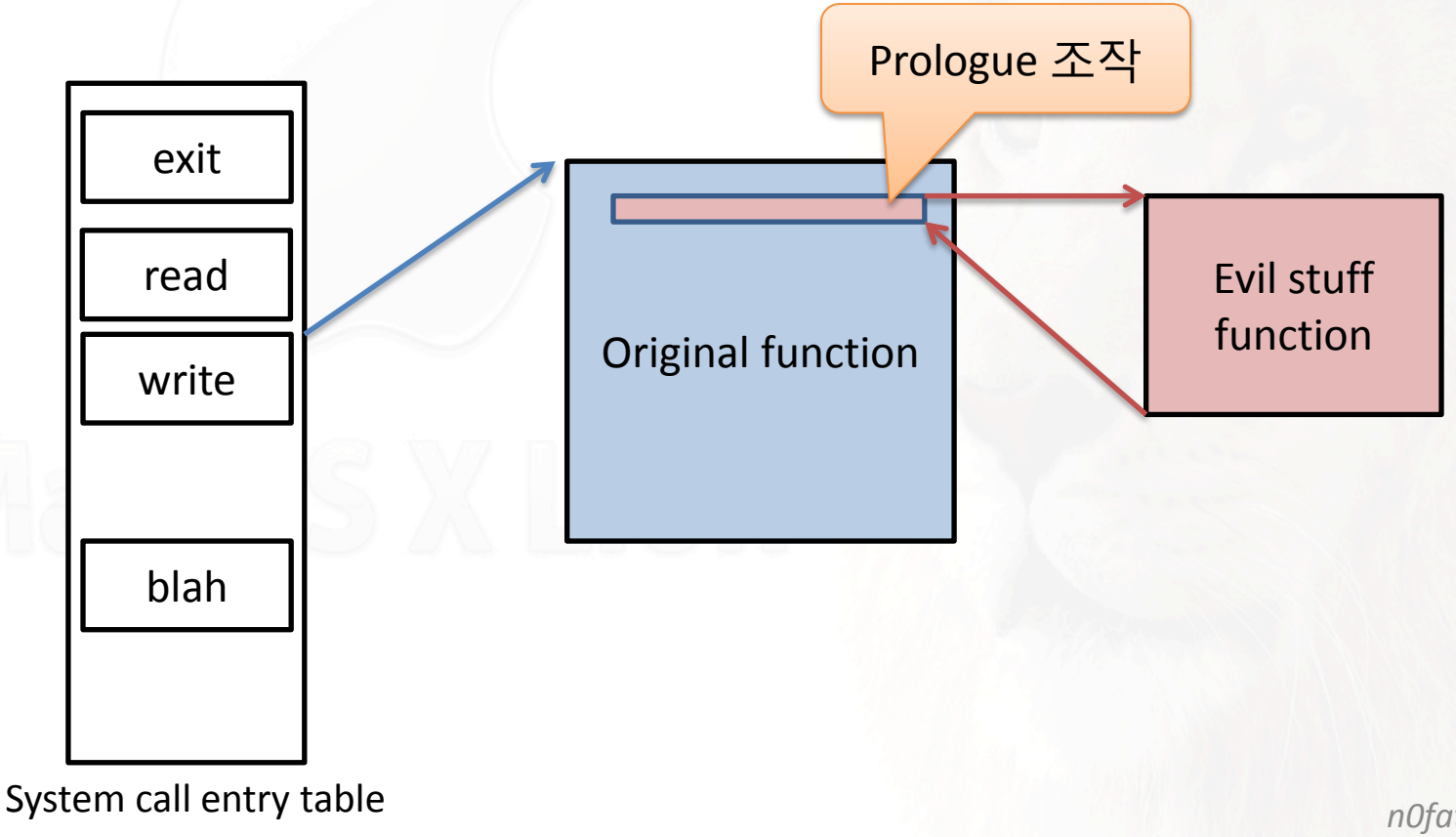
Rootkit technique

- System Call Hooking
  - System call Entry Table 을 Hooking 하여 원하는 명령 수행



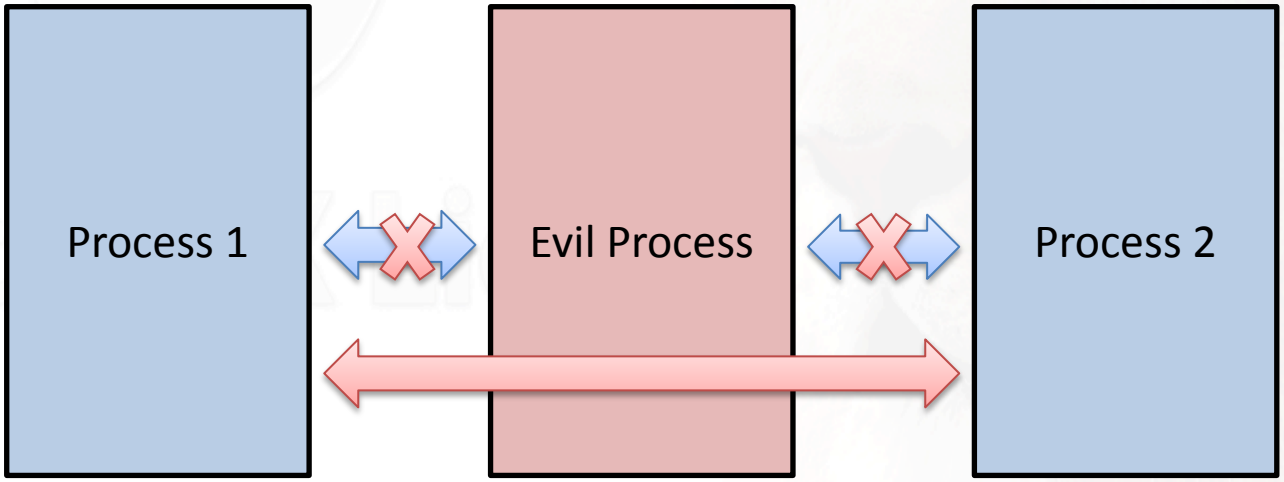
Rootkit technique

- Inline function hooking
  - 함수의 프롤로그나 함수 전체를 대체하는 방법



Rootkit technique

- DKOM (Direct Kernel Object Manipulation)
  - 커널 객체를 조작하여 정보 은닉
  - Process 나 KEXT의 List Entry를 조작하여 은닉
    - KEXT는 Snow Leopard 부터 공개된 은닉 기법이 통하지 않음.



Rootkit Detection

- Detection Method
  - System call Hooking
    - Kernel Image 와 메모리 이미지의 System call entry table 비교
  - Inline function hooking
    - Kernel Image 의 System call 과 memory image 의 함수를 비교
  - DKOM
    - Process Hiding
      - Hash Table 을 이용한 탐지
      - Process Carving 을 이용한 탐지
    - KEXT Hiding
      - KEXT Carving 을 이용한 탐지

Example

- System Call Hooking을 이용하여 파일을 은닉하는 루트킷을 탐지 및 덤프
  - 덤프한 메모리 이미지에서 KEXT 정보를 추출하고 System call Hooking 여부를 확인
  - System call hooking 된 주소를 확인하고 KEXT가 로드된 주소를 비교하여 악성 KEXT 판단
  - KEXT 추출
  - Disassembler 를 이용하여 KEXT 분석

Q & A

Contact: rapfer@gmail.com