# Generating Volatility Linux Profile

*demantos*

*demantos@gmail.com*

*http://malwarel4b.blogspot.kr*

*Cho Hoon*

# Index

# Overview

- **Volatility Profile**
  - 윈도우용 프로파일은 기본적으로 제공
  - 리눅스용은 2012년 9월경에 여러 배포판에 대한 프로파일 제공
    - ✓ but, 배포판과 커널 버전이 맞지 않을 경우 제대로 분석되지 않음

```
root@LUCKYSTRIKE:~# vol.py -f /data/forensics/memdump/CentOS58_2.6.18-308.el5.dd -profile=LinuxCentOS63x64 linux_pslist
Volatile Systems Volatility Framework 2.3_beta
Offset     Name                 Pid             Uid             Gid    DTB        Start Time
---------- -------------------- --------------- --------------- ------ ---------- ----------
No suitable address space mapping found
Tried to open image as:
 MachOAddressSpace: mac: need base
 LimeAddressSpace: lime: need base
 WindowsHiberFileSpace32: No base Address Space
 WindowsCrashDumpSpace64: No base Address Space
 HPAKAddressSpace: No base Address Space
 VirtualBoxCoreDumpElf64: No base Address Space
 VMWareSnapshotFile: No base Address Space
 WindowsCrashDumpSpace32: No base Address Space
 AMD64PagedMemory: No base Address Space
 IA32PagedMemoryPae: No base Address Space
 IA32PagedMemory: No base Address Space
 MachOAddressSpace: MachO Header signature invalid
 MachOAddressSpace: MachO Header signature invalid
 LimeAddressSpace: Invalid Lime header signature
 WindowsHiberFileSpace32: No xpress signature found
 WindowsCrashDumpSpace64: Header signature invalid
 HPAKAddressSpace: Invalid magic found
 VirtualBoxCoreDumpElf64: ELF64 Header signature invalid
 VMWareSnapshotFile: Invalid VMware signature: 0x0
 WindowsCrashDumpSpace32: Header signature invalid
 AMD64PagedMemory: Incompatible profile WinXPSP2x86 selected
 IA32PagedMemoryPae: No valid DTB found
 IA32PagedMemory: No valid DTB found
 FileAddressSpace: Must be first Address Space
 ArmAddressSpace: No valid DTB found
```

## Volatility Linux Profile

- 배보판과 커널 버전이 다를 경우 Volatility에서 제공하는 프로파일을 통해 메모리 분석 불가

  ✓ You must ensure the profile you build matches the target system in

  1) Linux distribution

  2) exact kernel version

  3) CPU architecture (32-bit, 64-bit, etc).

  - https://code.google.com/p/volatility/wiki/LinuxMemoryForensics

- **Needs (in Korea)**
  - 클라이언트 리눅스의 메모리를 분석할 경우는 극히 드뭄
  - 국내 리눅스 서버는 대부분 CentOS 5.x or RHEL
    - ✓ Volatility에서 제공하는 프로파일만으로는 국내 환경의 리눅스 서버 메모리 분석 불가
    - ✓ 해외에서는 SuSe를 많이 사용하는 듯...(?)

# Generating

- **Requirement**
- **Generate linux profile**
- **Confirmation**

# Generating

- **Requirement**
  - dwarfdump
    - ✓ Debian/Ubuntu
      - apt-get install dwarfdump
    - ✓ OpenSuSe, Fedora, other
      - libdwarf-tools : http://reality.sgiweb.org/davea/dwarf.html
  - gcc / make
  - kernel header
    - ✓ kernel-devel
    - ✓ kernel-headers

# Generating

- **최신 버전 Volatility 다운로드**
- **프로파일 생성**
  - vtypes(kernel data structure) 생성

```
[root@cent58_x86 ~]# clear
[root@cent58_x86 ~]# cd volatility-read-only/tools/linux/
[root@cent58_x86 linux]# make
make -C //lib/modules/2.6.18-348.12.1.el5/build CONFIG_DEBUG_INFO=y M=/root/volatility-read-only/tools/linux modules
make[1]: Entering directory `/usr/src/kernels/2.6.18-348.12.1.el5-i686'
  CC [M]  /root/volatility-read-only/tools/linux/module.o
/root/volatility-read-only/tools/linux/module.c:303:5: warning: "STATS" is not defined
/root/volatility-read-only/tools/linux/module.c:319:5: warning: "DEBUG" is not defined
  Building modules, stage 2.
  MODPOST
  CC      /root/volatility-read-only/tools/linux/module.mod.o
  LD [M]  /root/volatility-read-only/tools/linux/module.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.18-348.12.1.el5-i686'
dwarfdump -di module.ko > module.dwarf
make -C //lib/modules/2.6.18-348.12.1.el5/build M=/root/volatility-read-only/tools/linux clean
make[1]: Entering directory `/usr/src/kernels/2.6.18-348.12.1.el5-i686'
  CLEAN   /root/volatility-read-only/tools/linux/.tmp_versions
make[1]: Leaving directory `/usr/src/kernels/2.6.18-348.12.1.el5-i686'
```

## ▪ 프로파일 생성

- Symbol 가져오기 (System.map-*{kernel version}*)
  - ✓ 현재 커널 버전과 일치하는 System.map 파일을 사용해야 함
- 압축
  - ✓ 리눅스 프로파일은 앞서 생성한 vtypes 파일과 심볼 파일을 압축한 형태

```
[root@cent58_x86 linux]# zip CentOS58_`uname -r`.zip ./module.dwarf /boot/System.map-`uname -r`
  adding: module.dwarf (deflated 90%)
  adding: boot/System.map-2.6.18-348.12.1.el5 (deflated 73%)
[root@cent58_x86 linux]# ls -l ~/volatility-read-only/volatility/plugins/overlays/linux/
total 848
-rw-r--r-- 1 root root 390800 Jul 30 13:31 CentOS58_2.6.18-348.12.1.el5.zip
-rw-r--r-- 1 root root   4842 Jul 29 05:38 elf.py
-rw-r--r-- 1 root root      0 Jul 29 05:38 __init__.py
-rw-r--r-- 1 root root   1636 Jul 29 05:38 linux64.py
-rw-r--r-- 1 root root  36616 Jul 29 05:38 linux.py
```

  - ✓ 압축 파일은 volatility/plugins/overlays/linux /디렉토리로 이동 or 복사
  - ✓ 허무하지만 프로파일은 이게 끝.
    - 단, 생성된 프로파일이 해당 시스템에서 덤프된 메모리를 정상적으로 분석할 수 있는지 확인해야 함

- **Profile 목록 확인**

```
E:\memdump>volatility.py --info | grep -i linux
Volatile Systems Volatility Framework 2.3_beta
LinuxCentOS57_2_6_18-274_el5x86           - A Profile for Linux CentOS57_2.6.18-274.el5 x86
LinuxCentOS58_2_6_18-308_4_1_el5x86       - A Profile for Linux CentOS58_2.6.18-308.4.1.el5 x86
LinuxCentOS58_2_6_18-308_el5x86           - A Profile for Linux CentOS58_2.6.18-308.el5 x86
LinuxCentOS59_2_6_18-348_el5x86           - A Profile for Linux CentOS59_2.6.18-348.el5 x86
LinuxCentOS5x_2_6_18-348_12_1_el5x86      - A Profile for Linux CentOS5x_2.6.18-348.12.1.el5 x86
LinuxCentOS60_2_6_32-71_el6_i686x86       - A Profile for Linux CentOS60_2.6.32-71.el6.i686 x86
LinuxCentOS61_2_6_32-131_0_15_el6_i686x86 - A Profile for Linux CentOS61_2.6.32-131.0.15.el6.i686 x86
LinuxCentOS62_2_6_32-220_el6_i686x86      - A Profile for Linux CentOS62_2.6.32-220.el6.i686 x86
LinuxCentOS63_2_6_32-279_el6_i686x86      - A Profile for Linux CentOS63_2.6.32-279.el6.i686 x86
LinuxCentOS63x64                          - A Profile for Linux CentOS63 x64
LinuxRHEL5_2_6_18-164_el5PAEx86           - A Profile for Linux RHEL5_2.6.18-164.el5PAE x86
LinuxRHEL5x86                             - A Profile for Linux RHEL5 x86
LinuxUbuntu12_04_3_LTS_x64x64             - A Profile for Linux Ubuntu12.04.3_LTS_x64 x64
...
```

# Analyzing

- 플러그인

```
root@LUCKYSTRIKE:~# vol.py --info | grep linux
Volatile Systems Volatility Framework 2.3_beta
linux_arp               - Print the ARP table
linux_bash              - Recover bash history from bash process memory
linux_check_afinfo      - Verifies the operation function pointers of network protocols
linux_check_creds       - Checks if any processes are sharing credential structures
linux_check_evt_arm     - Checks the Exception Vector Table to look for syscall table hooking
linux_check_fop         - Check file operation structures for rootkit modifications
linux_check_idt         - Checks if the IDT has been altered
linux_check_modules     - Compares module list to sysfs info, if available
linux_check_syscall     - Checks if the system call table has been altered
linux_check_syscall_arm - Checks if the system call table has been altered
linux_check_tty         - Checks tty devices for hooks
linux_cpuinfo           - Prints info about each active processor
linux_dentry_cache      - Gather files from the dentry cache
linux_dmesg             - Gather dmesg buffer
linux_dump_map          - Writes selected memory mappings to disk
linux_find_file         - Recovers tmpfs filesystems from memory
linux_ifconfig          - Gathers active interfaces
linux_iomem             - Provides output similar to /proc/iomem
linux_keyboard_notifier - Parses the keyboard notifier call chain
linux_lsmod             - Gather loaded kernel modules
linux_lsof              - Lists open files
linux_memmap            - Dumps the memory map for linux tasks
linux_moddump           - Extract loaded kernel modules
linux_mount             - Gather mounted fs/devices
linux_mount_cache       - Gather mounted fs/devices from kmem_cache
linux_netstat           - Lists open sockets
linux_pidhashtable      - Enumerates processes through the PID hash table
linux_pkt_queues        - Writes per-process packet queues out to disk
linux_proc_maps         - Gathers process maps for linux
linux_psaux             - Gathers processes along with full command line and start time
linux_pslist            - Gather active tasks by walking the task_struct->task list
linux_pslist_cache      - Gather tasks from the kmem_cache
linux_pstree            - Shows the parent/child relationship between processes
linux_psxview           - Find hidden processes with various process listings
linux_route_cache       - Recovers the routing cache from memory
linux_sk_buff_cache     - Recovers packets from the sk_buff kmem_cache
linux_slabinfo          - Mimics /proc/slabinfo on a running machine
linux_tmpfs             - Recovers tmpfs filesystems from memory
linux_vma_cache         - Gather VMAs from the vm_area_struct cache
linux_volshell          - Shell in the memory image
linux_yarascan          - A shell in the Linux memory image
```

- **Just Testing...**

```
E:\memdump>volatility.py -f CentOS58_2.6.18-308.4.1.el5.dd --profile=LinuxCentOS58_2_6_18-308_4_1_el5x86
linux_pslist
Volatile Systems Volatility Framework 2.3_beta
Offset     Name                 Pid             Uid             Gid    DTB        Start Time
---------- -------------------- --------------- --------------- ------ ---------- ----------
0xf7d1baa0 init                 1               0               0      0x019c8000 2013-09-04 23:07:43 UTC+0000
0xf7d1b550 migration/0          2               0               0      ---------- 2013-09-04 23:07:43 UTC+0000
0xf7d1b000 ksoftirqd/0          3               0               0      ---------- 2013-09-04 23:07:43 UTC+0000
0xf7fefaa0 events/0             4               0               0      ---------- 2013-09-04 23:07:43 UTC+0000
0xf7fef550 khelper              5               0               0      ---------- 2013-09-04 23:07:43 UTC+0000
0xf7fef000 kthread              6               0               0      ---------- 2013-09-04 23:07:43 UTC+0000
0xf7fd6000 kblockd/0            9               0               0      ---------- 2013-09-04 23:07:43 UTC+0000
0xf7fb4aa0 kacpid               10              0               0      ---------- 2013-09-04 23:07:43 UTC+0000
...[snip]...
0xe9cd2000 gnome-terminal       4413            0               0      0x2090c000 2013-09-04 23:21:10 UTC+0000
0xe1b74000 gnome-pty-helpe      4416            0               0      0x1f308000 2013-09-04 23:21:12 UTC+0000
0xe9cd2aa0 bash                 4417            0               0      0x1e6ed000 2013-09-04 23:21:12 UTC+0000
0xddd2aaa0 notification-da      5180            0               0      0x1e1f2000 2013-09-05 00:09:56 UTC+0000
0xe25ceaa0 sshd                 6351            0               0      0x1d2eb000 2013-09-05 01:23:57 UTC+0000
0xc1944000 bash                 6354            0               0      0x1e3e9000 2013-09-05 01:23:57 UTC+0000
0xddd2a550 back                 10301           0               0      0x16e5d000 2013-09-05 01:27:08 UTC+0000
0xe2101000 syslogd/0            10353           0               0      ---------- 2013-09-05 01:28:38 UTC+0000
0xf704a550 insmod               10755           0               0      0x16e8f000 2013-09-05 01:40:01 UTC+0000
0xd5bf4550 pdflush              10759           0               0      ---------- 2013-09-05 01:40:06 UTC+0000
0xf7c75550 pdflush              10765           0               0      ---------- 2013-09-05 01:40:15 UTC+0000
0xf7c75000 pdflush              10772           0               0      ---------- 2013-09-05 01:40:36 UTC+0000
```

- **플러그인에 따라서 기능을 지원해주지 않는 프로파일이 존재함**

```
E:\memdump>volatility.py -f sujan.memdump.lime --profile=LinuxRHEL5_2_6_18-164_el5PAEx86 linux_psxview
Volatile Systems Volatility Framework 2.3_beta
Offset(V)  Name                    PID pslist pid_hash kmem_cache
---------- -------------------- ------ ------ -------- ----------
ERROR   : volatility.plugins.linux.pidhashtable: calculate_v2: This profile is currently unsupported by this
plugin. Please file a bug report on our issue tracker to have support added.
```

```
E:\memdump>volatility.py -f sujan.memdump.lime --profile=LinuxRHEL5_2_6_18-164_el5PAEx86 linux_check_creds
Volatile Systems Volatility Framework 2.3_beta
PIDs
--------
ERROR   : volatility.plugins.linux.check_creds: This command is not supported in this profile.
```

- 이 외에도 분석에 꼭 필요할 것으로 판단되는 플러그인 중에 안되는게 몇 개 있음

# Conclusion

- **Future Work**
- **Discussion**
- **Reference**

# Future Work

- **64bit용 프로파일 생성**
  - CentOS에서 동일한 방식으로 64bit용 프로파일 생성 실패
  - 현재까지는 원인은 찾지 못했으며 리눅스 메모리 구조에 대한 연구 후 진행해야 할 듯...
- **생성한 프로파일에 대한 검증**
  - 실제 사고가 발생한 리눅스 서버에서 메모리 덤프 후 만들어 둔 프로파일을 이용하여 정상적으로 분석 가능한지 검증 필요
- **How to analyze...**

- **사고 현장에서 조사 대상 리눅스 서버의 커널 버전에 대한 프로파일이 없는 경우는?**
  - CentOS의 경우 현재 시점을 기준으로 업데이트 할 경우
    - ✓ 5.x 버전은 5.9 버전과 동일한 커널로 업데이트
    - ✓ 6.x 버전은 6.3 버전과 동일한 커널로 업데이트
  - 즉, 최신 커널로 업데이트 되는데 최신 커널 발표 전에 업데이트 후 최신 커널로 업데이트 하지 않은 경우 해당 커널에 대한 프로파일 존재하지 않음
    - ✓ 조사 대상 시스템에서 해당 커널 버전에 대한 System.map 파일 채증 가능한데 module.dwarf는?
      - 조사 대상 시스템에서 컴파일은 risk가 존재함
      - 단, 2.6.18-308.el5 버전(clean)에서 컴파일하여 생성한 module.dwarf와 2.6.18-308.4.1.el5 버전에서 채증한 System.map 파일로 프로파일을 만들어 테스트한 결과 정상 동작하였음
      - **다른 배포판/커널 버전에서도 동일한 방식으로 프로파일을 생성해서 사용 가능한지 테스트 필요**

# Reference

- https://code.google.com/p/volatility/wiki/LinuxMemoryForensics
- http://www.infosecisland.com/blogview/22406-Analyzing-the-Average-Coder-Rootkit-Bash-History-and-Elevated-Processes-with-Volatility.html

# Question and Answer