Mac OS X Physical Memory Analysis

Matthieu Suiche
BlackHat DC – February 2010

Netherlands Forensic Institute
www.forensicinstitute.nl

Who am I?

Researcher for the Netherlands Forensics Institute (NFI).

Microsoft Enterprise Security MVP

Speaker at various security events, such as *PacSec*, *BlackHat USA*, *Europol High Tech Crime Meeting*, *Shakacon*, etc.

Past work:

- SandMan Framework (Windows hibernation file)
- Win**32/64**dd (Windows memory acquisition utility).

Agenda

Introduction

Analysis

Who ?

Forensics Experts

Investigators

Incident Response Engineers

…

Why ?

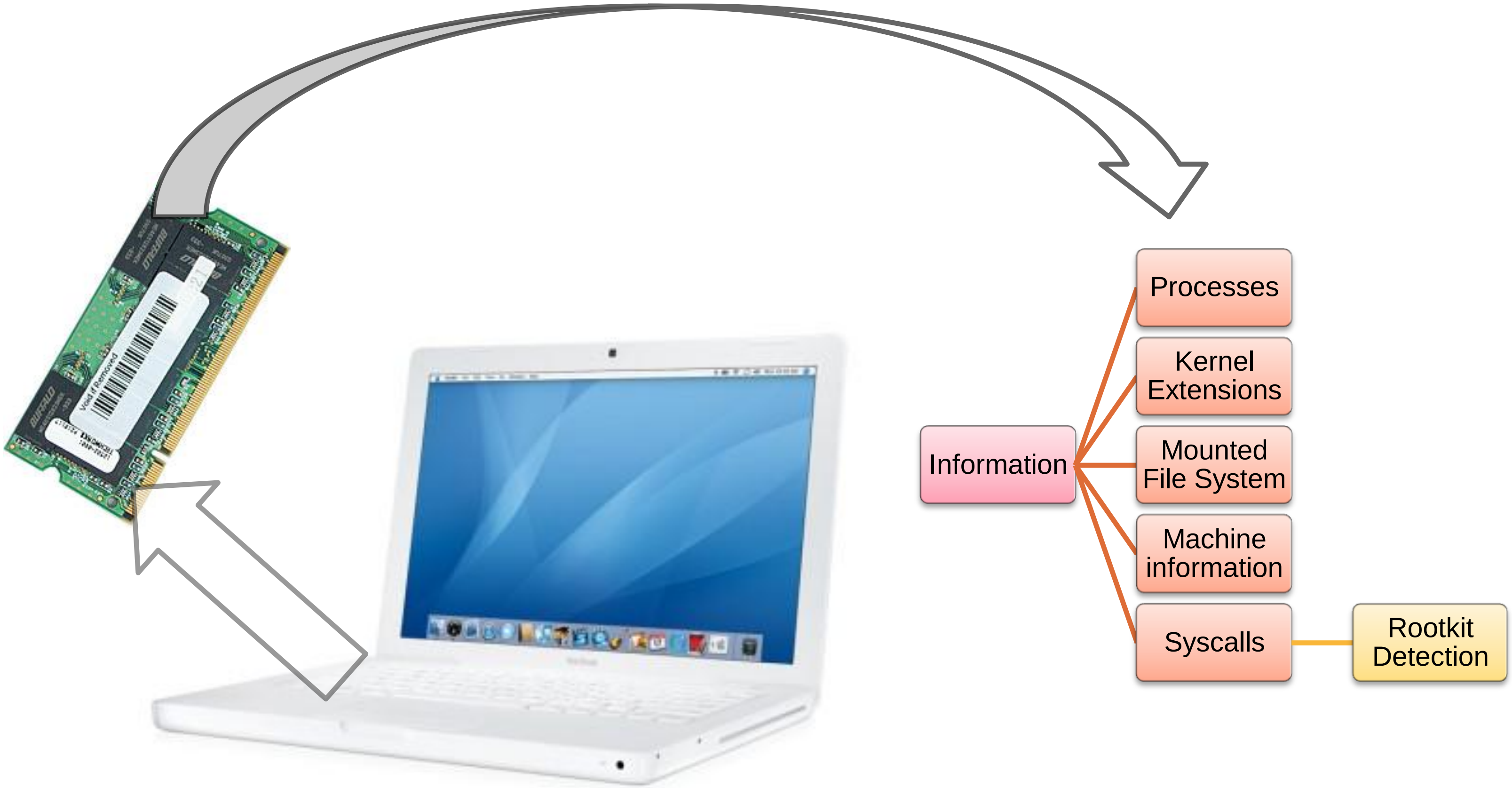
Pros:

1. Sometimes non-volatile memory is not enough, then we need volatile memory (Physical Memory).

Cons:

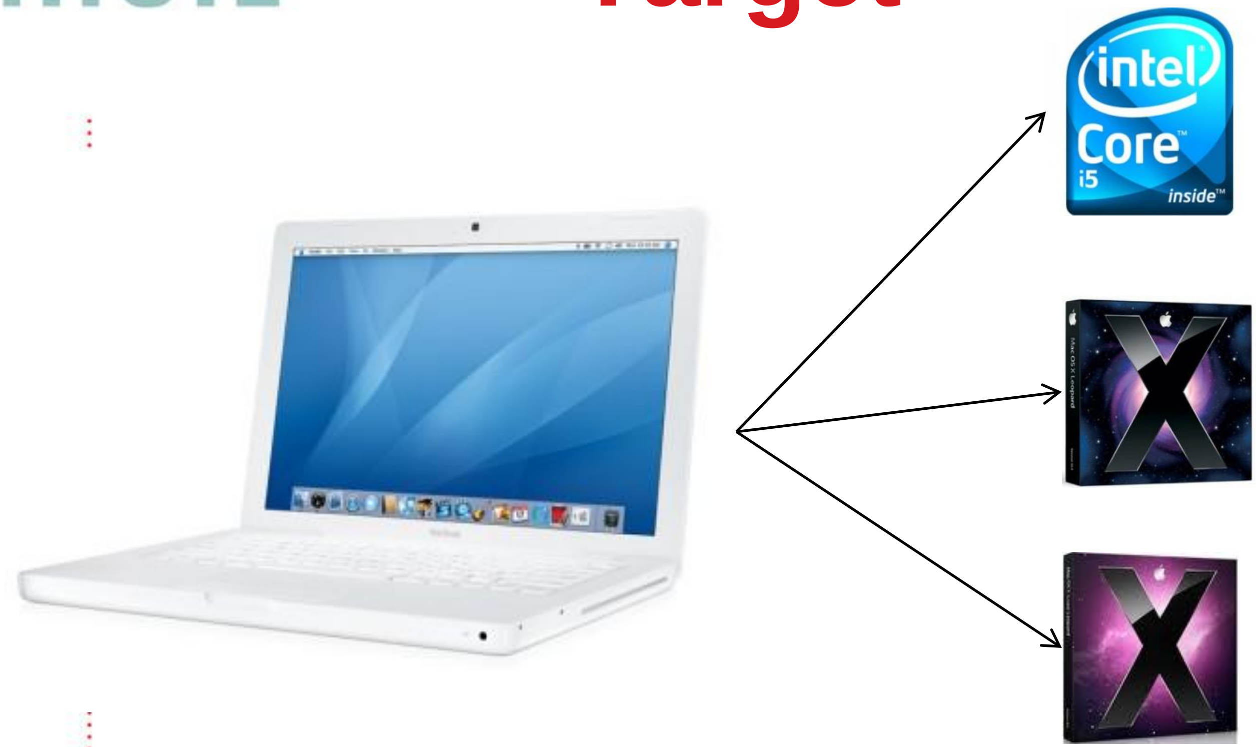
1. Very complex.
2. Lack of research.

Overview



- Information
  - Processes
  - Kernel Extensions
  - Mounted File System
  - Machine information
  - Syscalls
    - Rootkit Detection

Target



Intel Processor (x86/x64)

Mac OS X Leopard 10.5

Mac OS X Snow Leopard 10.6

Software-based acquisition

/dev/mem

Cons: Disabled by default.

Pros: We can write our own driver.

Hibernation a.k.a. “safe sleep”

Pros: Present on all modern O.S.

Cons: Compressed, and can be encrypted if *secure virtual memory* mechanism is used. (`hibernatemode == 5`)

Agenda

Introduction

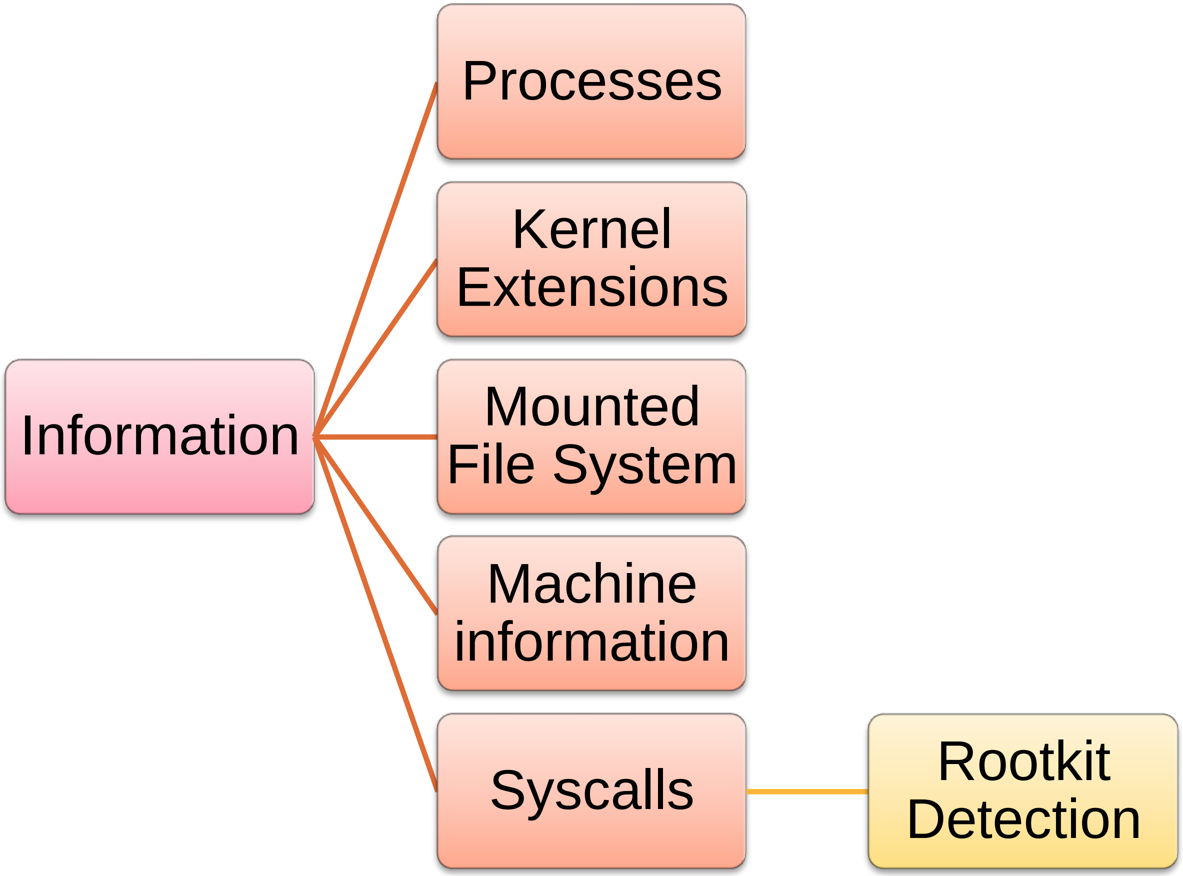
Analysis

Analysis

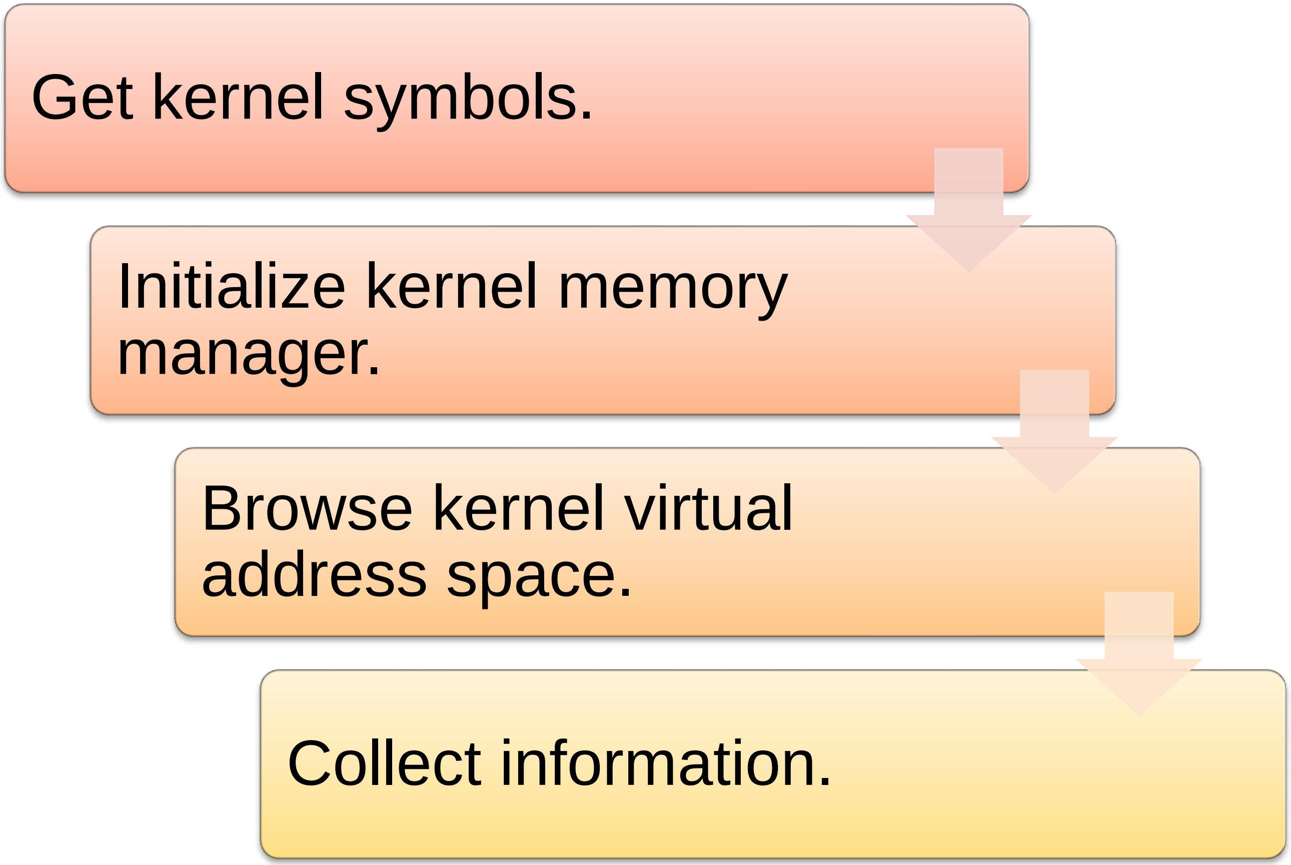
Goal: To avoid random string searching.

To be precise and efficient.

Information Goldmine



Analysis



Get kernel symbols.

Initialize kernel memory manager.

Browse kernel virtual address space.

Collect information.

Kernel Symbols

Windows compiler stores symbols in externals files called *.PDB

Mac OS X compiler stores symbols inside a section which is part of the executable.

Mac OS X kernel executable (mach_kernel) as symbol database.

Kernel Symbols

Why?

__KLD, __LINKEDIT, __PRELINK and __symtab kernel sections are destroyed as soon as the kernel (mach_kernel) is loaded by `removeKernelLinker()` function.

What?

__LINKEDIT section contains variable names and offsets.

Kernel Symbols

Quick **K**ernel **V**irtual To **P**hysical **A**ddress Formula is:

Operating System	Quick translation Formula
i386 Linux	KPA = KVA – 0xC0000000
Playstation 3 Linux	KPA = KVA – 0xC000000000000000
Windows	KPA = KVA & 0x1FFFF000
Mac OS X	**KPA = KVA**

Now we can read variables from the symbol section in the physical memory.

Kernel Symbols

Works only for the mapped executable kernel (__text and __data sections)

Does not work for allocated buffers.

.data interesting exported variables:

Memory manager variables

Memory Manager

Super interesting variables

- `_IdlePDPT`
- `_IdlePDPT64`
- `_IdlePML4`
- `_IdlePTD`

Page Map Level 4 is initialized on x86 version even if x86 only use PAE.

PML4 ?

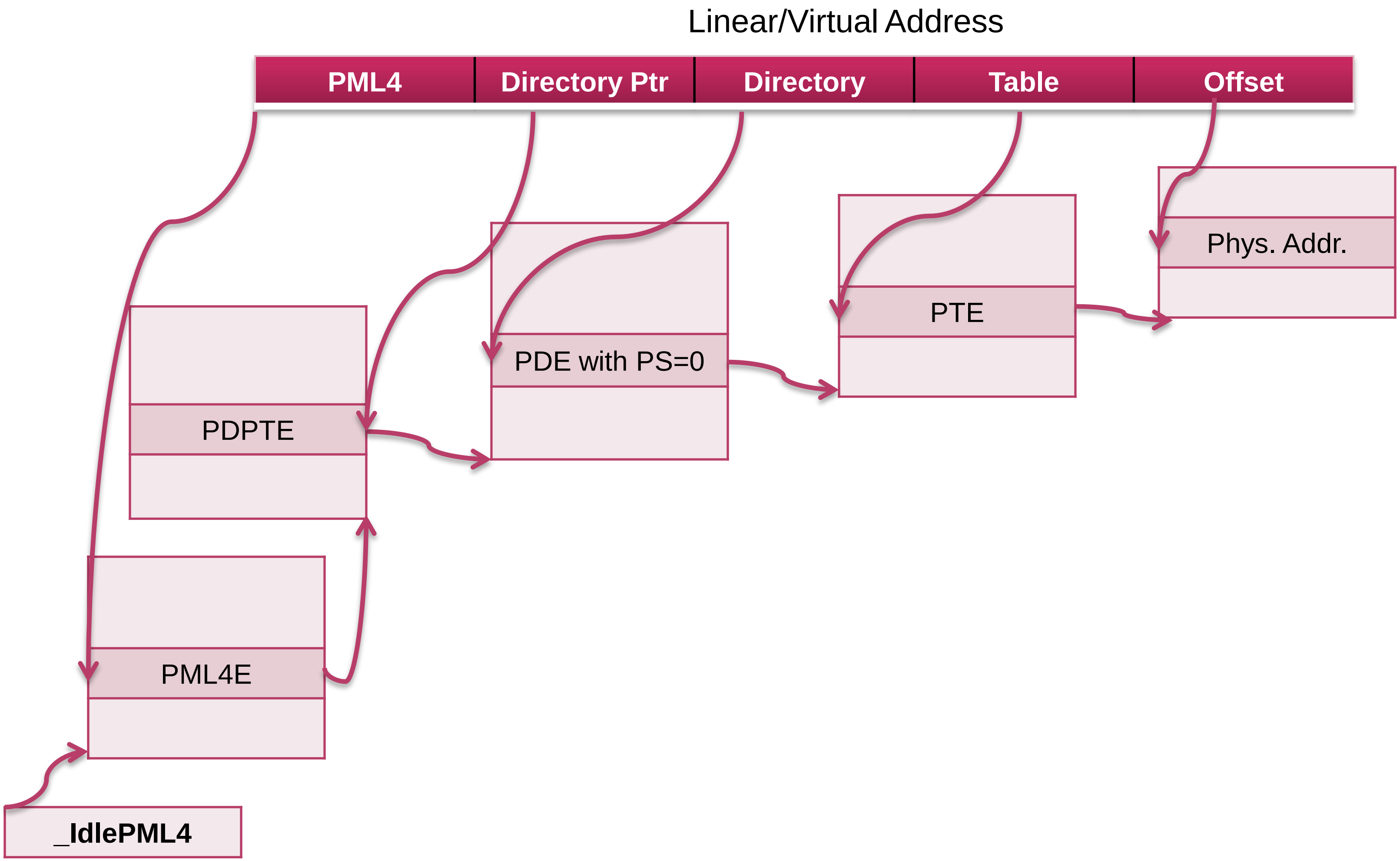
Page Map Level 4 paging method.

Supports 48-bits linear/virtual addresses.

Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide

4.5 IA-32E Paging

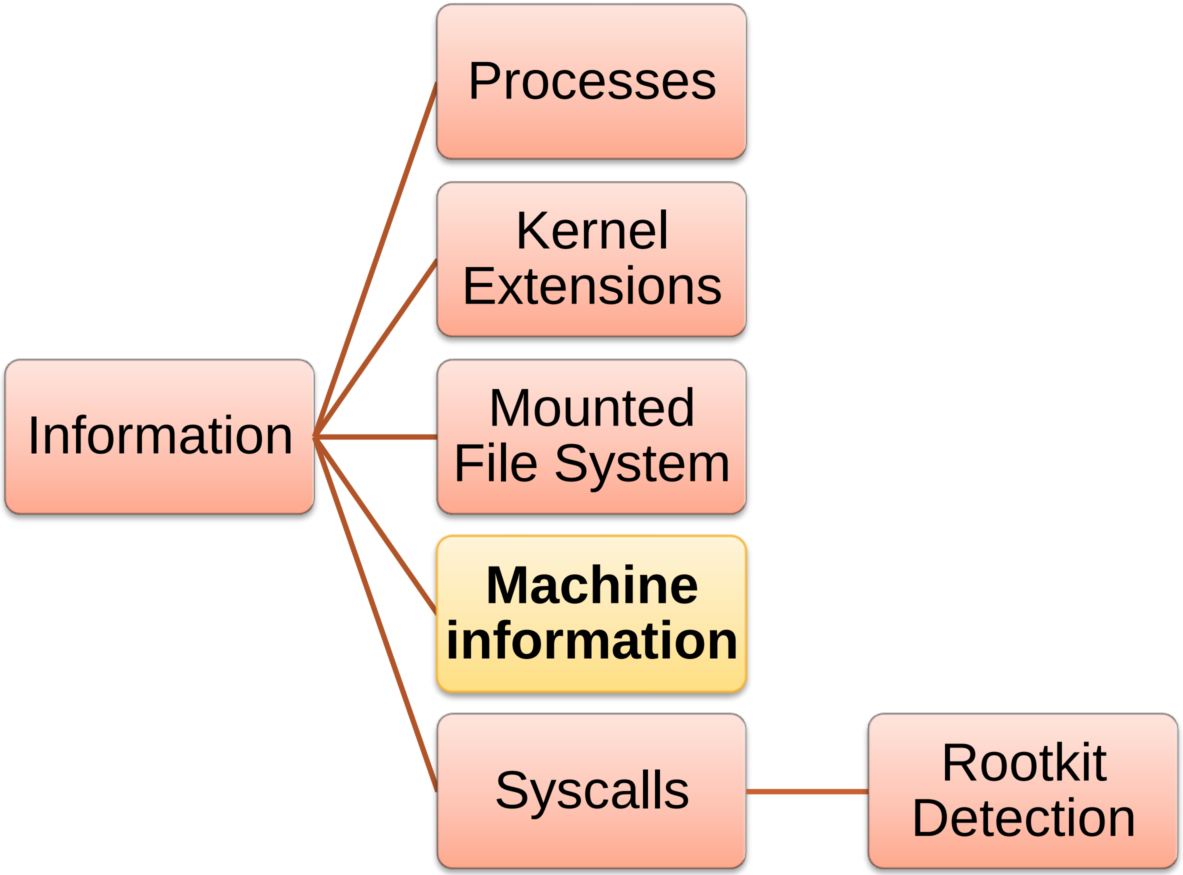
PML4


Linear/Virtual Address
PML4
Directory Ptr
Directory
Table
Offset
Phys. Addr.
PTE
PDE with PS=0
PDPTE
PML4E
_IdlePML4

Information

Now, we can browse the kernel virtual address space.

Machine Information



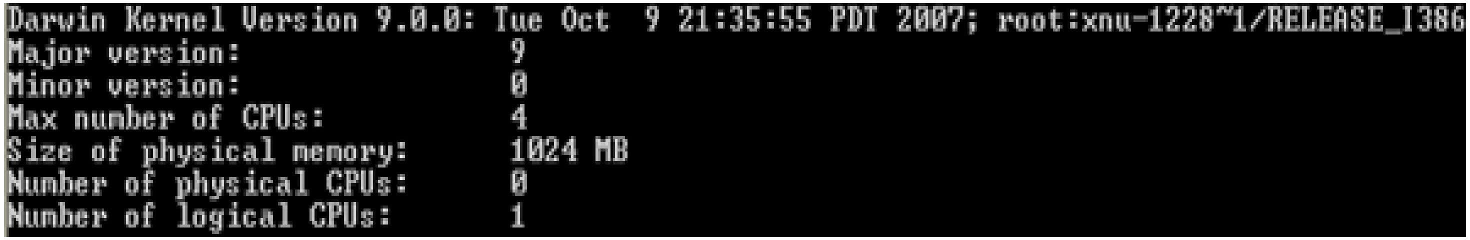
Machine Information

`version` variable contains a string with kernel version and compilation time

`machine_info` variable / structure contains:

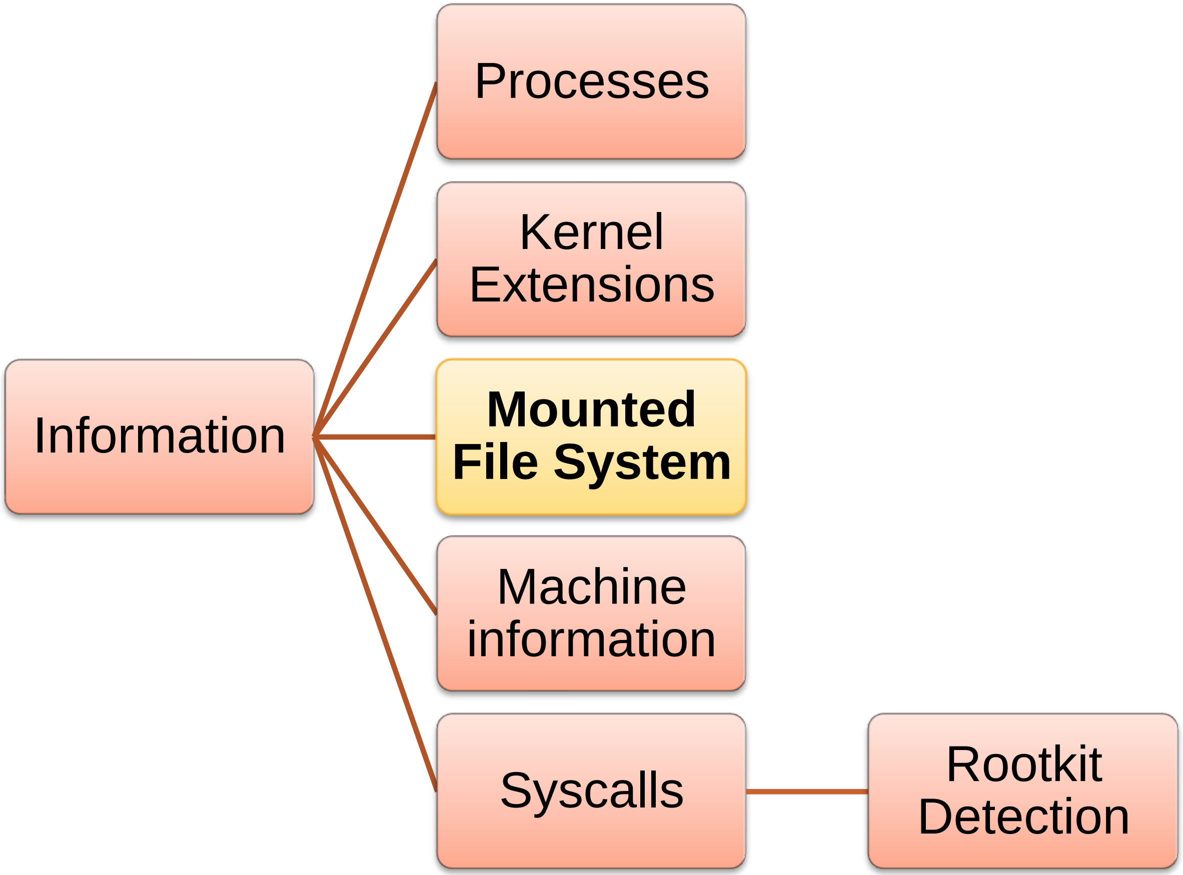
Field Name	Description
major_version	Major OS Version
minor_version	Minor OS Version
max_mem	Physical Memory size
physical_cpu	Number of physical CPU
logical_cpu	Number of logical CPU

Machine Information



```
Darwin Kernel Version 9.0.0: Tue Oct  9 21:35:55 PDT 2007; root:xnu-1228~1/RELEASE_I386
Major version:              9
Minor version:              0
Max number of CPUs:         4
Size of physical memory:    1024 MB
Number of physical CPUs:    0
Number of logical CPUs:     1
```

Mounted File System

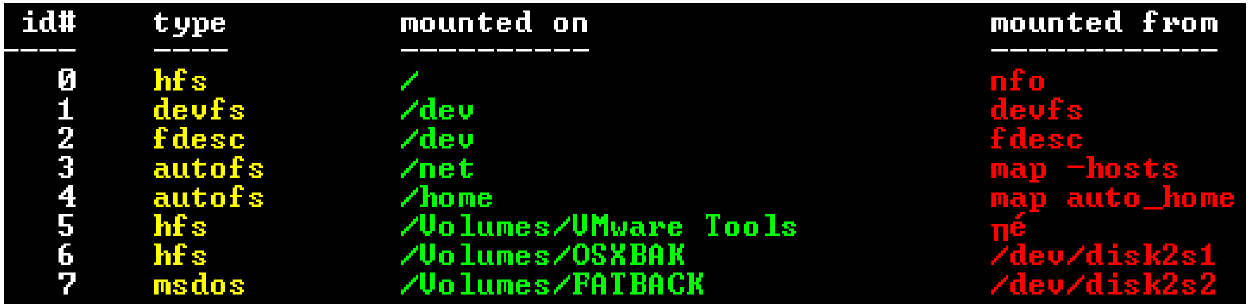


Mounted File System

Link-list called `mountlist`, defined by `mount` structure.

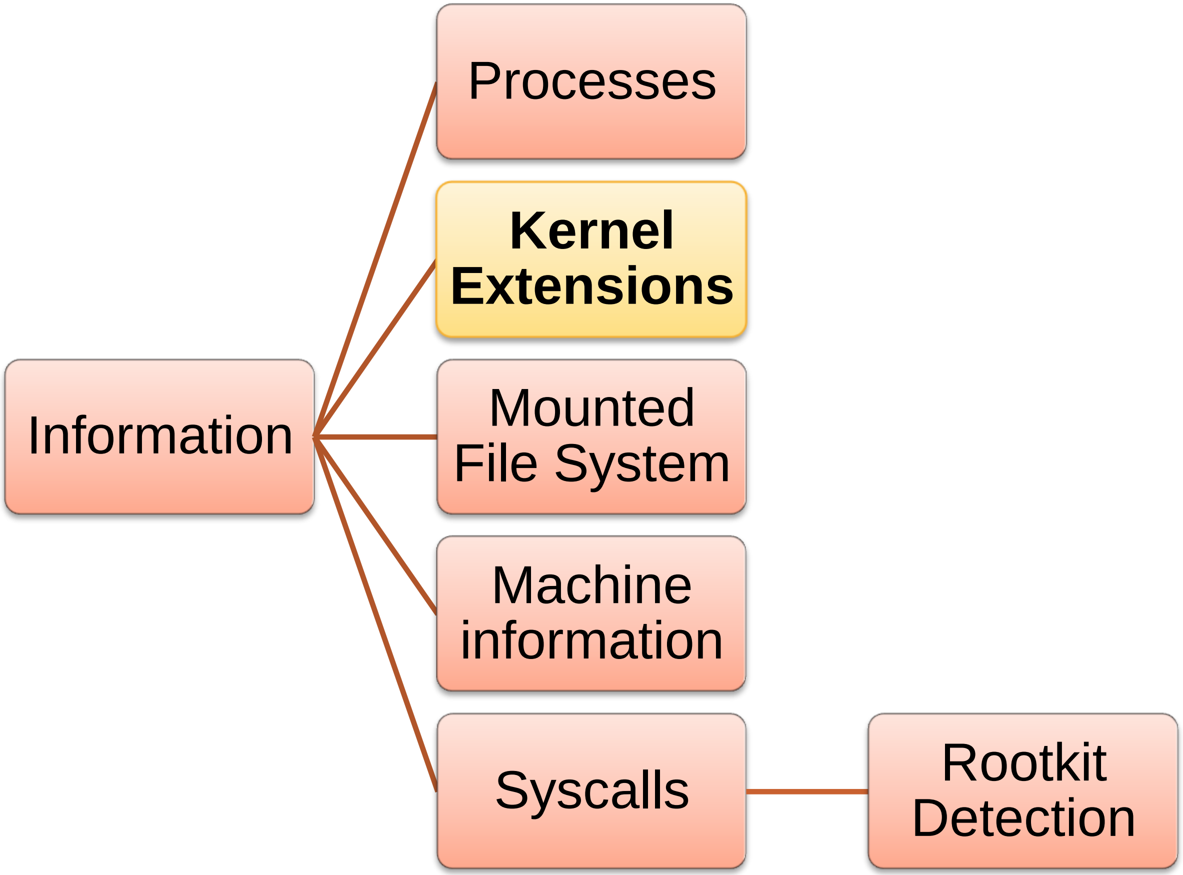
Field Name	Description
f_fstypename	File system type
f_mntonname	Mounted directory
f_mntfromname	Mounted file system

Mounted File System



```
 id#       type         mounted on                     mounted from
----       ----         ----------                     ------------
   0       hfs          /                              nfo
   1       devfs        /dev                           devfs
   2       fdesc        /dev                           fdesc
   3       autofs       /net                           map -hosts
   4       autofs       /home                          map auto_home
   5       hfs          /Volumes/VMware Tools          πé
   6       hfs          /Volumes/OSXBAK                /dev/disk2s1
   7       msdos        /Volumes/FATBACK               /dev/disk2s2
```

Kernel Extensions


Information

Processes

Kernel Extensions

Mounted File System

Machine information

Syscalls

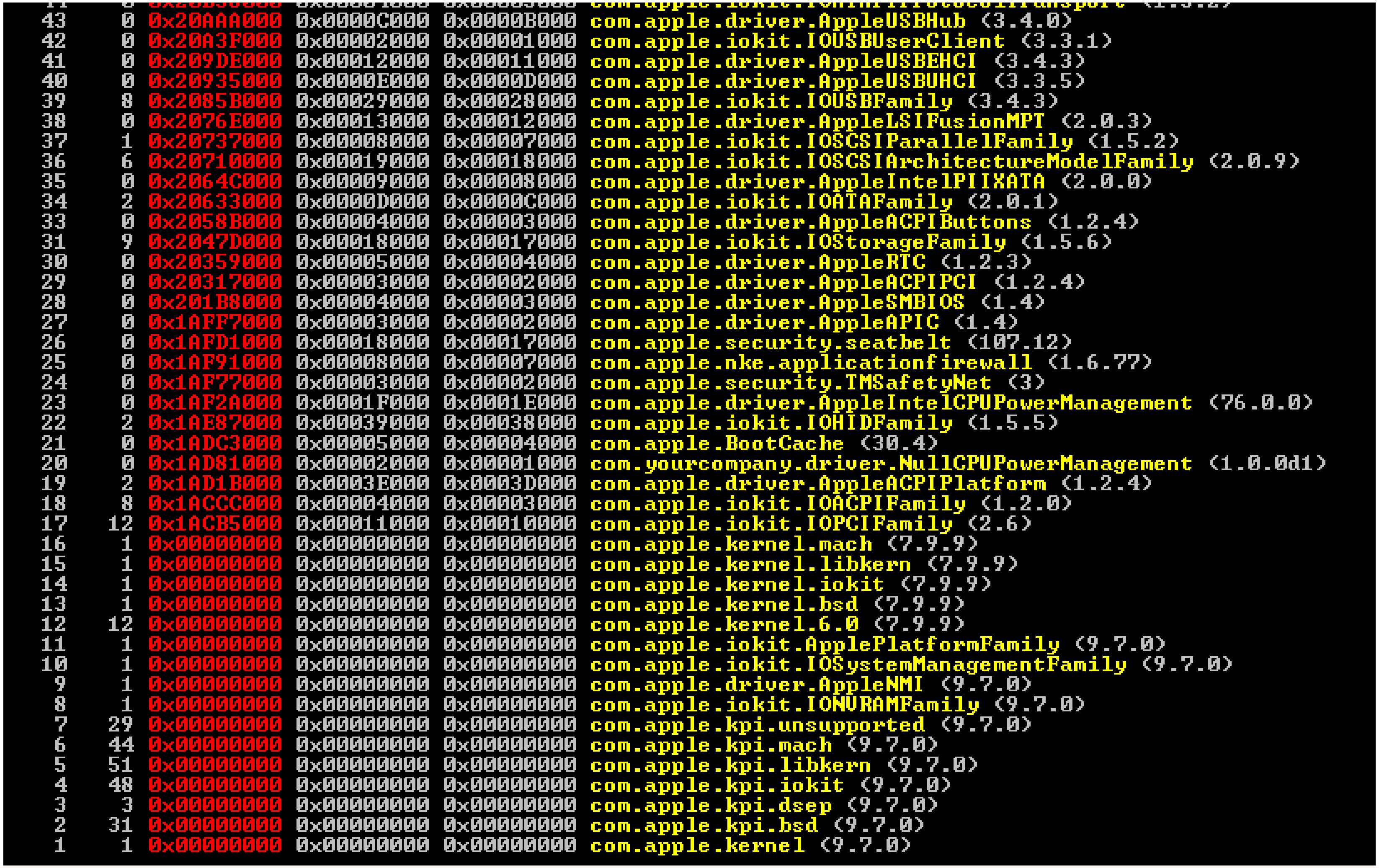
Rootkit Detection

Kernel Extensions

`kmod` variable is the list-head of every loaded kernel extensions defined by `kmod` structure.

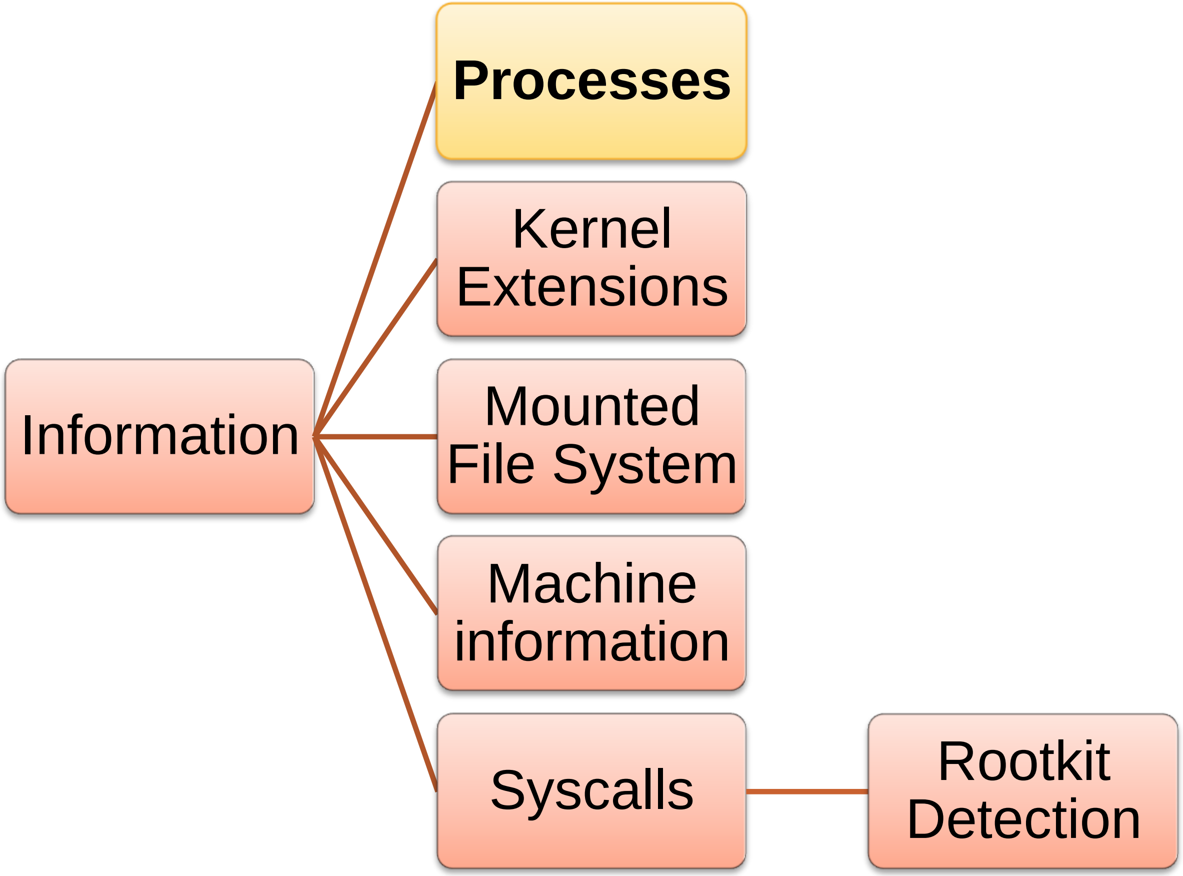
Field Name	Description
address	Base Address
size	Total Size
hdr_size	Header Size
name	Extension Name
version	Version
next	Pointer to the next entry

Kernel Extensions



```
   43    0 0x20AAA000 0x0000C000 0x0000B000 com.apple.driver.AppleUSBHub (3.4.0)
   42    0 0x20A3F000 0x00002000 0x00001000 com.apple.iokit.IOUSBUserClient (3.3.1)
   41    0 0x209DE000 0x00012000 0x00011000 com.apple.driver.AppleUSBEHCI (3.4.3)
   40    0 0x20935000 0x0000E000 0x0000D000 com.apple.driver.AppleUSBUHCI (3.3.5)
   39    8 0x2085B000 0x00029000 0x00028000 com.apple.iokit.IOUSBFamily (3.4.3)
   38    0 0x2076E000 0x00013000 0x00012000 com.apple.driver.AppleLSIFusionMPT (2.0.3)
   37    1 0x20737000 0x00008000 0x00007000 com.apple.iokit.IOSCSIParallelFamily (1.5.2)
   36    6 0x20710000 0x00019000 0x00018000 com.apple.iokit.IOSCSIArchitectureModelFamily (2.0.9)
   35    0 0x2064C000 0x00009000 0x00008000 com.apple.driver.AppleIntelPIIXATA (2.0.0)
   34    2 0x20633000 0x0000D000 0x0000C000 com.apple.iokit.IOATAFamily (2.0.1)
   33    0 0x2058B000 0x00004000 0x00003000 com.apple.driver.AppleACPIButtons (1.2.4)
   31    9 0x2047D000 0x00018000 0x00017000 com.apple.iokit.IOStorageFamily (1.5.6)
   30    0 0x20359000 0x00005000 0x00004000 com.apple.driver.AppleRTC (1.2.3)
   29    0 0x20317000 0x00003000 0x00002000 com.apple.driver.AppleACPIPCI (1.2.4)
   28    0 0x201B8000 0x00004000 0x00003000 com.apple.driver.AppleSMBIOS (1.4)
   27    0 0x1AFF7000 0x00003000 0x00002000 com.apple.driver.AppleAPIC (1.4)
   26    0 0x1AFD1000 0x00018000 0x00017000 com.apple.security.seatbelt (107.12)
   25    0 0x1AF91000 0x00008000 0x00007000 com.apple.nke.applicationfirewall (1.6.77)
   24    0 0x1AF77000 0x00003000 0x00002000 com.apple.security.TMSafetyNet (3)
   23    0 0x1AF2A000 0x0001F000 0x0001E000 com.apple.driver.AppleIntelCPUPowerManagement (76.0.0)
   22    2 0x1AE87000 0x00039000 0x00038000 com.apple.iokit.IOHIDFamily (1.5.5)
   21    0 0x1ADC3000 0x00005000 0x00004000 com.apple.BootCache (30.4)
   20    0 0x1AD81000 0x00002000 0x00001000 com.yourcompany.driver.NullCPUPowerManagement (1.0.0d1)
   19    2 0x1AD1B000 0x0003E000 0x0003D000 com.apple.driver.AppleACPIPlatform (1.2.4)
   18    8 0x1ACCC000 0x00004000 0x00003000 com.apple.iokit.IOACPIFamily (1.2.0)
   17   12 0x1ACB5000 0x00011000 0x00010000 com.apple.iokit.IOPCIFamily (2.6)
   16    1 0x00000000 0x00000000 0x00000000 com.apple.kernel.mach (7.9.9)
   15    1 0x00000000 0x00000000 0x00000000 com.apple.kernel.libkern (7.9.9)
   14    1 0x00000000 0x00000000 0x00000000 com.apple.kernel.iokit (7.9.9)
   13    1 0x00000000 0x00000000 0x00000000 com.apple.kernel.bsd (7.9.9)
   12   12 0x00000000 0x00000000 0x00000000 com.apple.kernel.6.0 (7.9.9)
   11    1 0x00000000 0x00000000 0x00000000 com.apple.iokit.ApplePlatformFamily (9.7.0)
   10    1 0x00000000 0x00000000 0x00000000 com.apple.iokit.IOSystemManagementFamily (9.7.0)
    9    1 0x00000000 0x00000000 0x00000000 com.apple.driver.AppleNMI (9.7.0)
    8    1 0x00000000 0x00000000 0x00000000 com.apple.iokit.IONVRAMFamily (9.7.0)
    7   29 0x00000000 0x00000000 0x00000000 com.apple.kpi.unsupported (9.7.0)
    6   44 0x00000000 0x00000000 0x00000000 com.apple.kpi.mach (9.7.0)
    5   51 0x00000000 0x00000000 0x00000000 com.apple.kpi.libkern (9.7.0)
    4   48 0x00000000 0x00000000 0x00000000 com.apple.kpi.iokit (9.7.0)
    3    3 0x00000000 0x00000000 0x00000000 com.apple.kpi.dsep (9.7.0)
    2   31 0x00000000 0x00000000 0x00000000 com.apple.kpi.bsd (9.7.0)
    1    1 0x00000000 0x00000000 0x00000000 com.apple.kernel (9.7.0)
```

Processes


Processes
Kernel Extensions
Information
Mounted File System
Machine information
Syscalls
Rootkit Detection

Processes

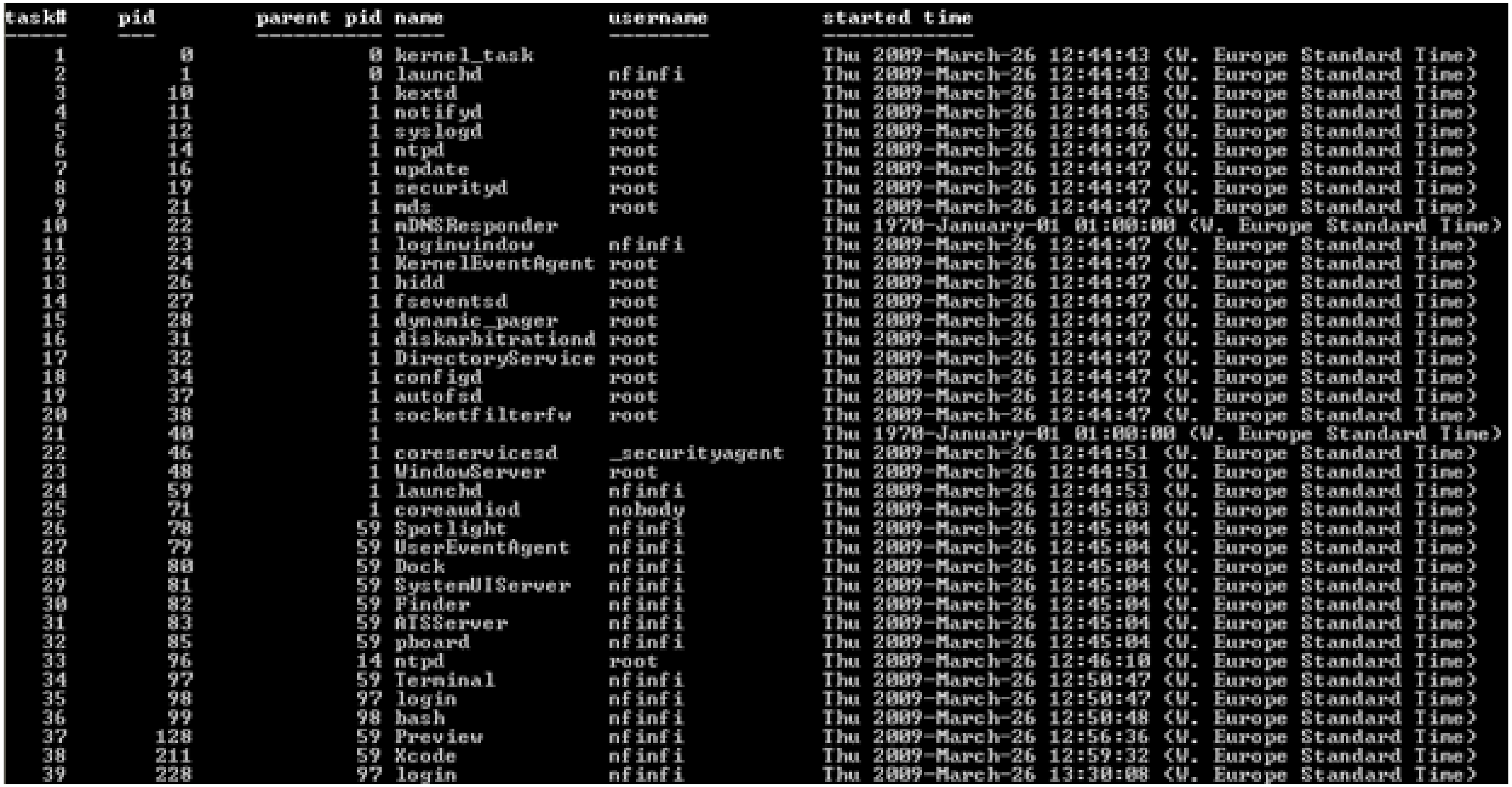
`kernproc` variable is list-head of every BSD processes defined by `proc` structure.

Contains PID, Parent PID, open files (file descriptors), children, threads, name and a pointer (`p_pgrp` field) to process group (`pgrp` structure).

`pgrp` structure contains a pointer to `session` structure (`pg_session` field).

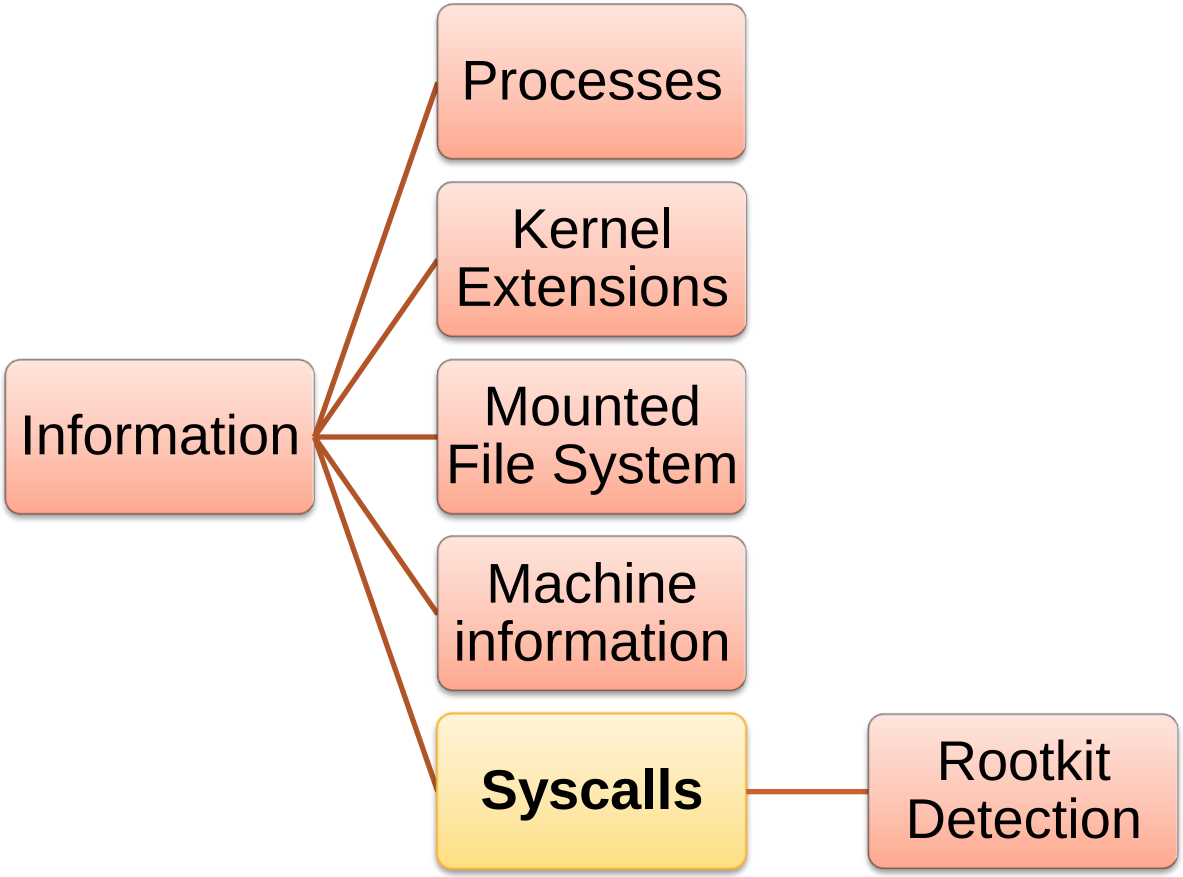
`session` structure contains username (`s_login` field) who launched the process.

Processes



```
task#    pid        parent pid name             username         started time
-----    ---        ---------- ----             --------         ------------
     1        0              0 kernel_task                       Thu 2009-March-26 12:44:43 (W. Europe Standard Time)
     2        1              0 launchd          nfinfi           Thu 2009-March-26 12:44:43 (W. Europe Standard Time)
     3       10              1 kextd            root             Thu 2009-March-26 12:44:45 (W. Europe Standard Time)
     4       11              1 notifyd          root             Thu 2009-March-26 12:44:45 (W. Europe Standard Time)
     5       12              1 syslogd          root             Thu 2009-March-26 12:44:46 (W. Europe Standard Time)
     6       14              1 ntpd             root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
     7       16              1 update           root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
     8       19              1 securityd        root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
     9       21              1 mds              root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    10       22              1 mDNSResponder                     Thu 1970-January-01 01:00:00 (W. Europe Standard Time)
    11       23              1 loginwindow      nfinfi           Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    12       24              1 KernelEventAgent root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    13       26              1 hidd             root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    14       27              1 fseventsd        root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    15       28              1 dynamic_pager    root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    16       31              1 diskarbitrationd root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    17       32              1 DirectoryService root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    18       34              1 configd          root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    19       37              1 autofsd          root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    20       38              1 socketfilterfw   root             Thu 2009-March-26 12:44:47 (W. Europe Standard Time)
    21       40              1                                   Thu 1970-January-01 01:00:00 (W. Europe Standard Time)
    22       46              1 coreservicesd    _securityagent   Thu 2009-March-26 12:44:51 (W. Europe Standard Time)
    23       48              1 WindowServer     root             Thu 2009-March-26 12:44:51 (W. Europe Standard Time)
    24       59              1 launchd          nfinfi           Thu 2009-March-26 12:44:53 (W. Europe Standard Time)
    25       71              1 coreaudiod       nobody           Thu 2009-March-26 12:45:03 (W. Europe Standard Time)
    26       78             59 Spotlight        nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    27       79             59 UserEventAgent   nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    28       80             59 Dock             nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    29       81             59 SystemUIServer   nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    30       82             59 Finder           nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    31       83             59 ATSServer        nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    32       85             59 pboard           nfinfi           Thu 2009-March-26 12:45:04 (W. Europe Standard Time)
    33       96             14 ntpd             root             Thu 2009-March-26 12:46:10 (W. Europe Standard Time)
    34       97             59 Terminal         nfinfi           Thu 2009-March-26 12:50:47 (W. Europe Standard Time)
    35       98             97 login            nfinfi           Thu 2009-March-26 12:50:47 (W. Europe Standard Time)
    36       99             98 bash             nfinfi           Thu 2009-March-26 12:50:48 (W. Europe Standard Time)
    37      128             59 Preview          nfinfi           Thu 2009-March-26 12:56:36 (W. Europe Standard Time)
    38      211             59 Xcode            nfinfi           Thu 2009-March-26 12:59:32 (W. Europe Standard Time)
    39      228             97 login            nfinfi           Thu 2009-March-26 13:30:08 (W. Europe Standard Time)
```

Syscalls


Information
Processes
Kernel Extensions
Mounted File System
Machine information
Syscalls
Rootkit Detection

Syscalls

- Syscall address is not exported

<u>Leopard</u>

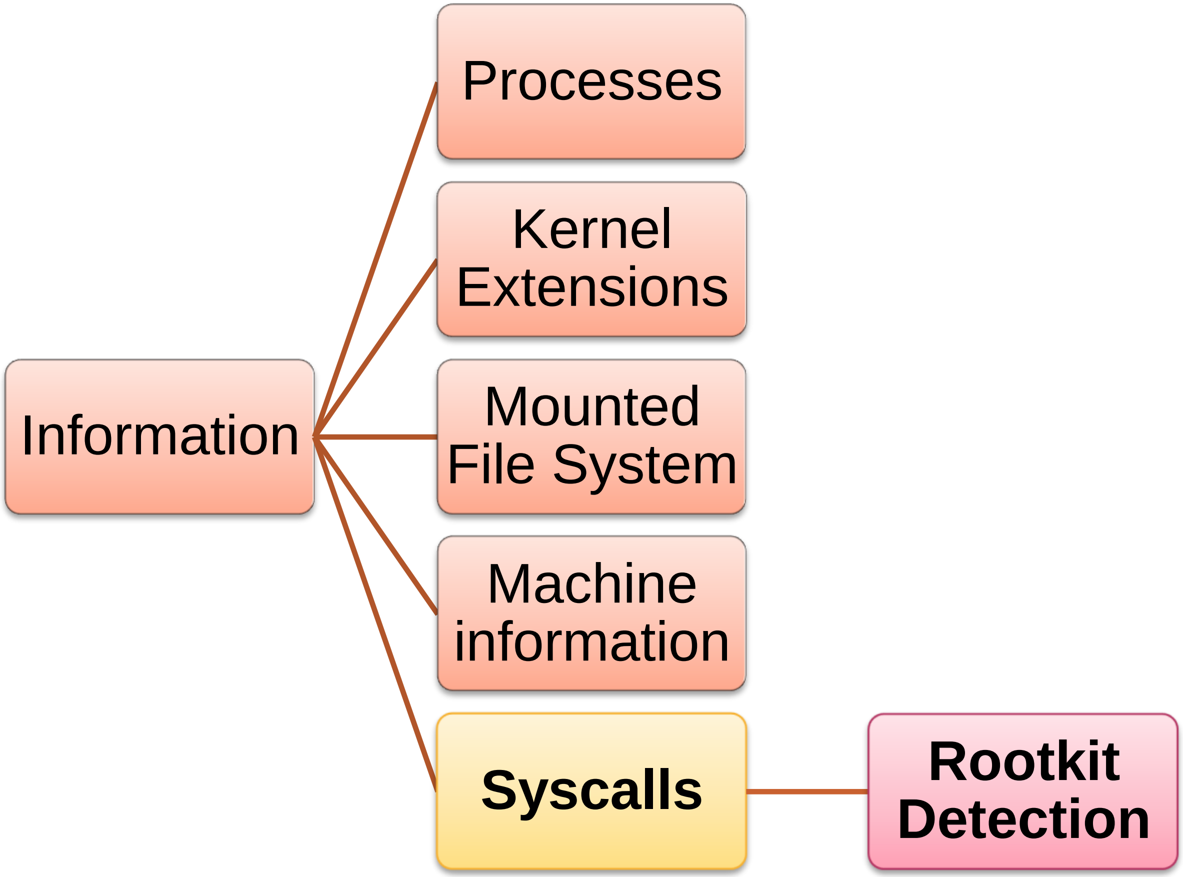
As explained by Jesse D’Aguanno at BH US 2008

```
&sysent = &nsysent + 0x20
```

<u>Snow Leopard</u>

```
&sysent = &nsysent - ((nsysent) * sizeof(sysent))
```

Syscalls



Information

Processes

Kernel Extensions

Mounted File System

Machine information

Syscalls

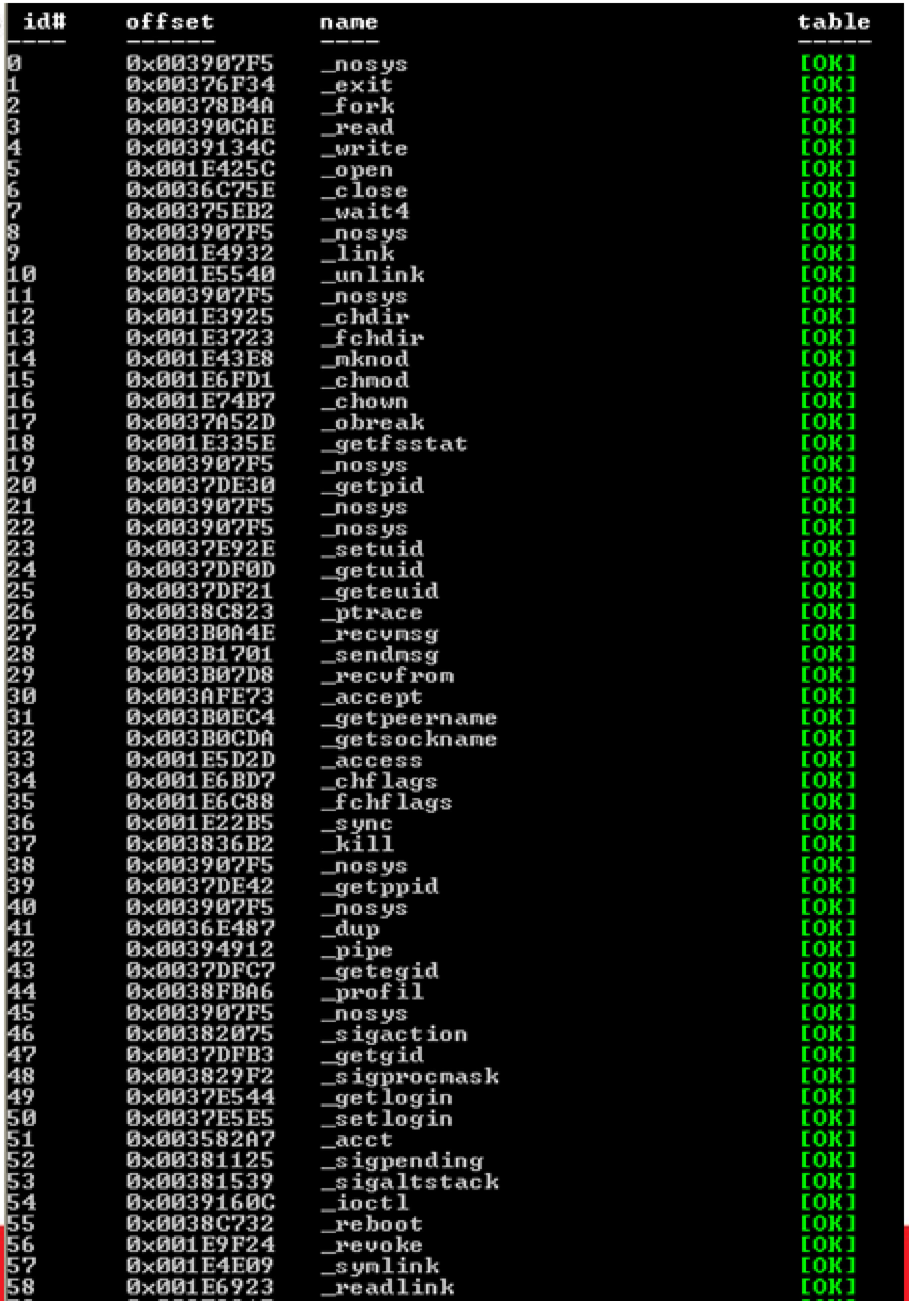
Rootkit Detection

Syscalls

If an offset from a syscall entry is not in kernel symbols.

Then, this is not normal ☺

Easy & Fast



```
 id#        offset         name                            table
----        ------         ----                            -----
0           0x003907F5     _nosys                          [OK]
1           0x00376F34     _exit                           [OK]
2           0x00378B4A     _fork                           [OK]
3           0x00390CAE     _read                           [OK]
4           0x0039134C     _write                          [OK]
5           0x001E425C     _open                           [OK]
6           0x0036C75E     _close                          [OK]
7           0x00375EB2     _wait4                          [OK]
8           0x003907F5     _nosys                          [OK]
9           0x001E4932     _link                           [OK]
10          0x001E5540     _unlink                         [OK]
11          0x003907F5     _nosys                          [OK]
12          0x001E3925     _chdir                          [OK]
13          0x001E3723     _fchdir                         [OK]
14          0x001E43E8     _mknod                          [OK]
15          0x001E6FD1     _chmod                          [OK]
16          0x001E74B7     _chown                          [OK]
17          0x0037A52D     _obreak                         [OK]
18          0x001E335E     _getfsstat                      [OK]
19          0x003907F5     _nosys                          [OK]
20          0x0037DE30     _getpid                         [OK]
21          0x003907F5     _nosys                          [OK]
22          0x003907F5     _nosys                          [OK]
23          0x0037E92E     _setuid                         [OK]
24          0x0037DF0D     _getuid                         [OK]
25          0x0037DF21     _geteuid                        [OK]
26          0x0038C823     _ptrace                         [OK]
27          0x003B0A4E     _recvmsg                        [OK]
28          0x003B1701     _sendmsg                        [OK]
29          0x003B07D8     _recvfrom                       [OK]
30          0x003AFE73     _accept                         [OK]
31          0x003B0EC4     _getpeername                    [OK]
32          0x003B0CDA     _getsockname                    [OK]
33          0x001E5D2D     _access                         [OK]
34          0x001E6BD7     _chflags                        [OK]
35          0x001E6C88     _fchflags                       [OK]
36          0x001E22B5     _sync                           [OK]
37          0x003836B2     _kill                           [OK]
38          0x003907F5     _nosys                          [OK]
39          0x0037DE42     _getppid                        [OK]
40          0x003907F5     _nosys                          [OK]
41          0x0036E487     _dup                            [OK]
42          0x00394912     _pipe                           [OK]
43          0x0037DFC7     _getegid                        [OK]
44          0x0038FBA6     _profil                         [OK]
45          0x003907F5     _nosys                          [OK]
46          0x00382075     _sigaction                      [OK]
47          0x0037DFB3     _getgid                         [OK]
48          0x003829F2     _sigprocmask                    [OK]
49          0x0037E544     _getlogin                       [OK]
50          0x0037E5E5     _setlogin                       [OK]
51          0x003582A7     _acct                           [OK]
52          0x00381125     _sigpending                     [OK]
53          0x00381539     _sigaltstack                    [OK]
54          0x0039160C     _ioctl                          [OK]
55          0x0038C732     _reboot                         [OK]
56          0x001E9F24     _revoke                         [OK]
57          0x001E4E09     _symlink                        [OK]
58          0x001E6923     _readlink                       [OK]
```

DEMO

Special thanks to

- Dino Dai Zovi
  - (Co-Author of **The Mac Hacker's Handbook**)

- Vincenzo Iozzo

Thanks for your attention

QUESTIONS ?