

SMRAM メモリフォレンジック

ネットエージェント株式会社
愛甲健二

x86

x86は以下の4つのモードを持つ

リアル・モード

ブートローダー

プロテクト・モード

OS本体、通常のアプリケーション

仮想8086モード

16bit コマンドプロンプト (.comファイル実行時など)

システム管理モード

電源管理、ハードウェア制御

x86

x86は以下の4つのモードを持つ

リアル・モード

ブートローダー

プロテクト・モード

OS本体、通常のアプリケーション

仮想8086モード

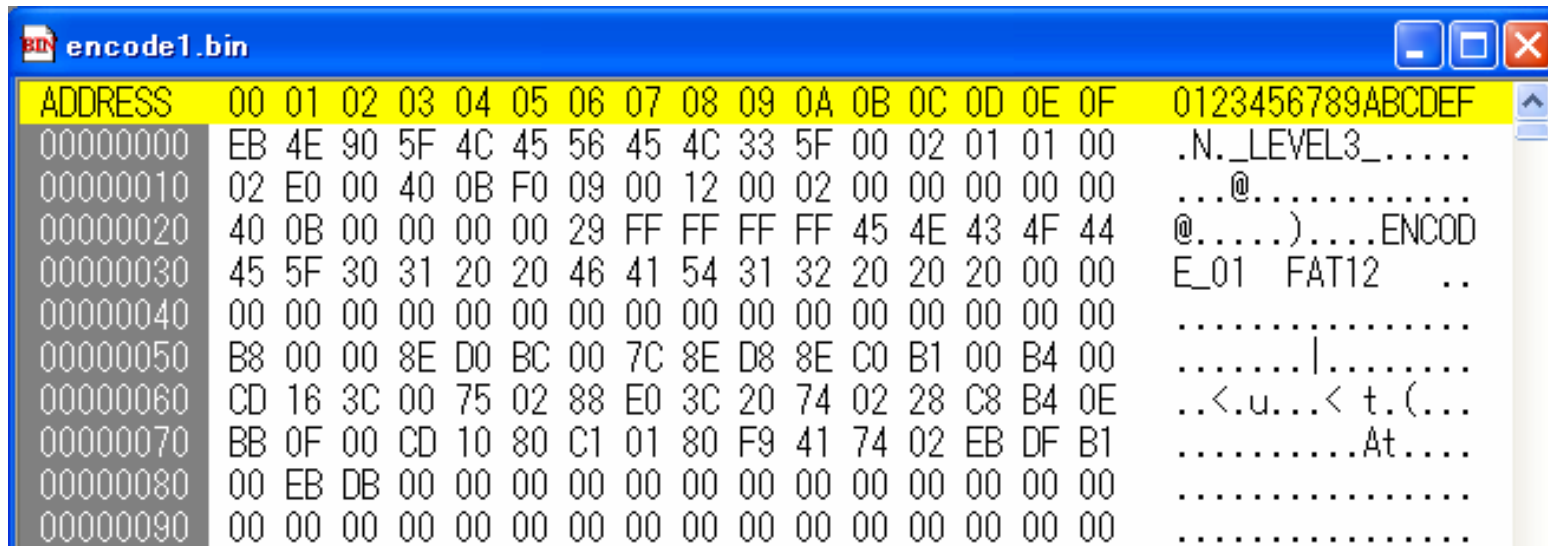
16bit コマンドプロンプト (.comファイル実行時など)

システム管理モード

電源管理、ハードウェア制御

リアル・モード(壺)

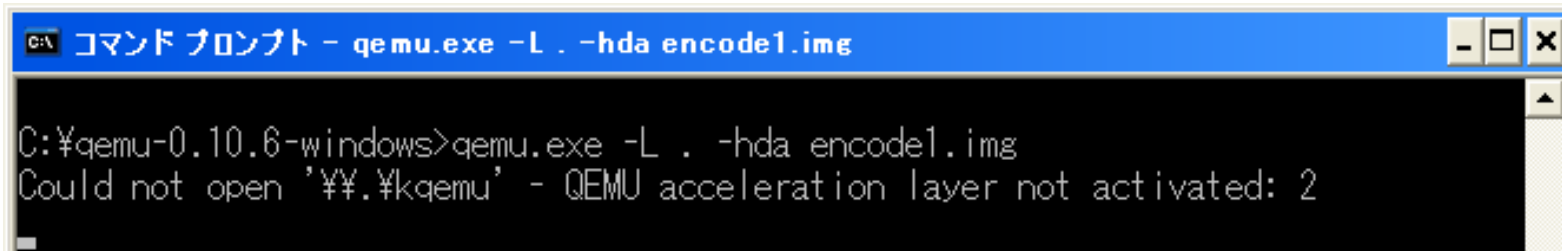
リアル・モードで動くプログラム



```
encode1.bin
ADDRESS 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
00000000 EB 4E 90 5F 4C 45 56 45 4C 33 5F 00 02 01 01 00 .N._LEVEL3_.....
00000010 02 E0 00 40 0B F0 09 00 12 00 02 00 00 00 00 00 ...@.....
00000020 40 0B 00 00 00 00 29 FF FF FF FF 45 4E 43 4F 44 @.....)....ENCOD
00000030 45 5F 30 31 20 20 46 41 54 31 32 20 20 20 00 00 E_01 FAT12 ..
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050 B8 00 00 8E D0 BC 00 7C 8E D8 8E C0 B1 00 B4 00 .....|.....
00000060 CD 16 3C 00 75 02 88 E0 3C 20 74 02 28 C8 B4 0E ..<.u...< t.(...
00000070 BB 0F 00 CD 10 80 C1 01 80 F9 41 74 02 EB DF B1 .....At....
00000080 00 EB DB 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

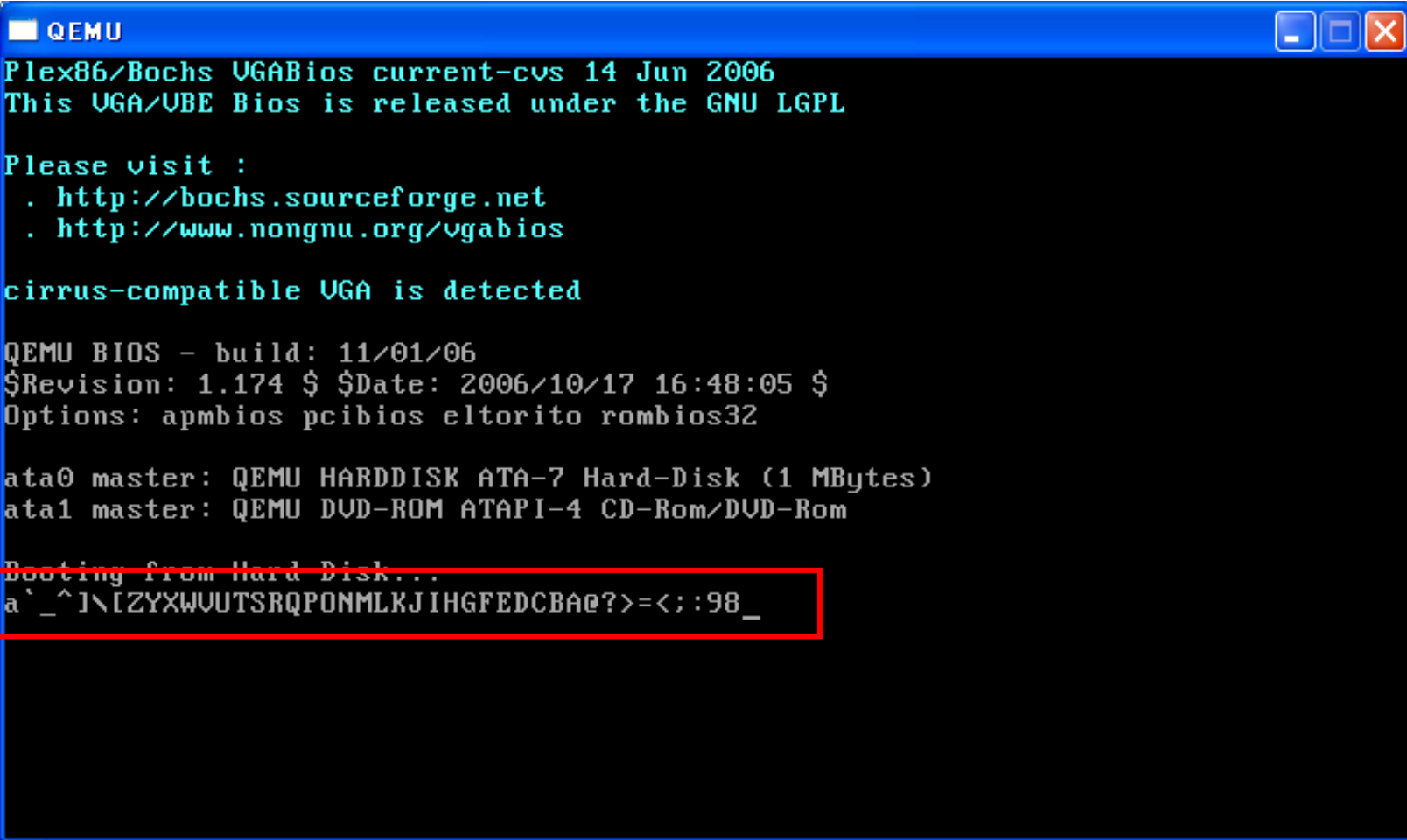
リアル・モード(弐)

実行



```
C:\¥qemu-0.10.6-windows>qemu.exe -L . -hda encode1.img
Could not open '¥¥.¥kqemu' - QEMU acceleration layer not activated: 2
```

リアル・モード(参)



```
QEMU
Plex86/Bochs VGABios current-cvs 14 Jun 2006
This UGA/VBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

cirrus-compatible UGA is detected

QEMU BIOS - build: 11/01/06
$Revision: 1.174 $ $Date: 2006/10/17 16:48:05 $
Options: apmbios pcibios eltorito rombios32

ata0 master: QEMU HARDDISK ATA-7 Hard-Disk (1 MBytes)
ata1 master: QEMU DVD-ROM ATAPI-4 CD-Rom/DVD-Rom

Booting from Hard Disk...
a`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<:;98_
```

x86

x86は以下の4つのモードを持つ

リアル・モード

ブートローダー

プロテクト・モード

OS本体、通常のアプリケーション

仮想8086モード

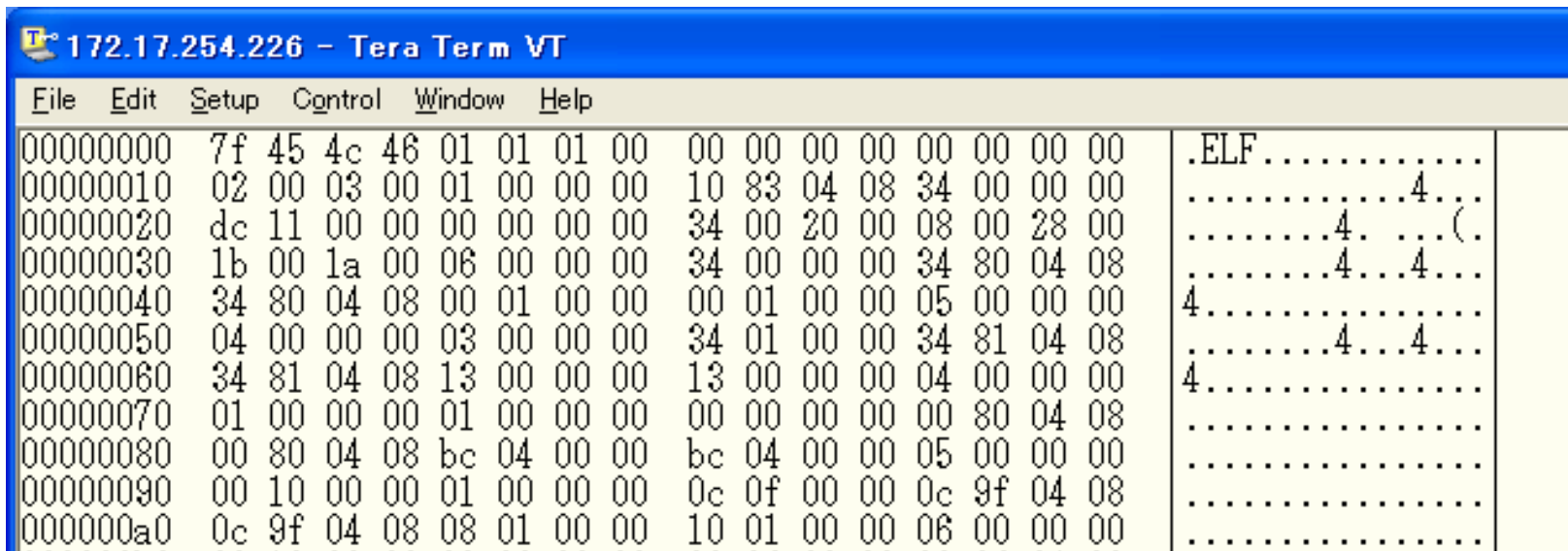
16bit コマンドプロンプト (.comファイル実行時など)

システム管理モード

電源管理、ハードウェア制御

プロテクト・モード

プロテクト・モードで動くプログラム



The screenshot shows a terminal window titled "172.17.254.226 - Tera Term VT". The window contains a hex dump of an ELF file header. The hex dump is displayed in a grid format with columns for hexadecimal values and their corresponding ASCII characters. The first few lines of the hex dump are:

```
00000000  7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00  .ELF.....
00000010  02 00 03 00 01 00 00 00 10 83 04 08 34 00 00 00  .....4...
00000020  dc 11 00 00 00 00 00 00 34 00 20 00 08 00 28 00  .....4. ....(
00000030  1b 00 1a 00 06 00 00 00 34 00 00 00 34 80 04 08  .....4...4...
00000040  34 80 04 08 00 01 00 00 00 01 00 00 05 00 00 00  4.....
00000050  04 00 00 00 03 00 00 00 34 01 00 00 34 81 04 08  .....4...4...
00000060  34 81 04 08 13 00 00 00 13 00 00 00 04 00 00 00  4.....
00000070  01 00 00 00 01 00 00 00 00 00 00 00 00 80 04 08  .....
00000080  00 80 04 08 bc 04 00 00 bc 04 00 00 05 00 00 00  .....
00000090  00 10 00 00 01 00 00 00 0c 0f 00 00 0c 9f 04 08  .....
000000a0  0c 9f 04 08 08 01 00 00 10 01 00 00 06 00 00 00  .....

```

要するに普通の実行ファイル

x86

x86は以下の4つのモードを持つ

リアル・モード

ブートローダー

プロテクト・モード

OS本体、通常のアプリケーション

仮想8086モード

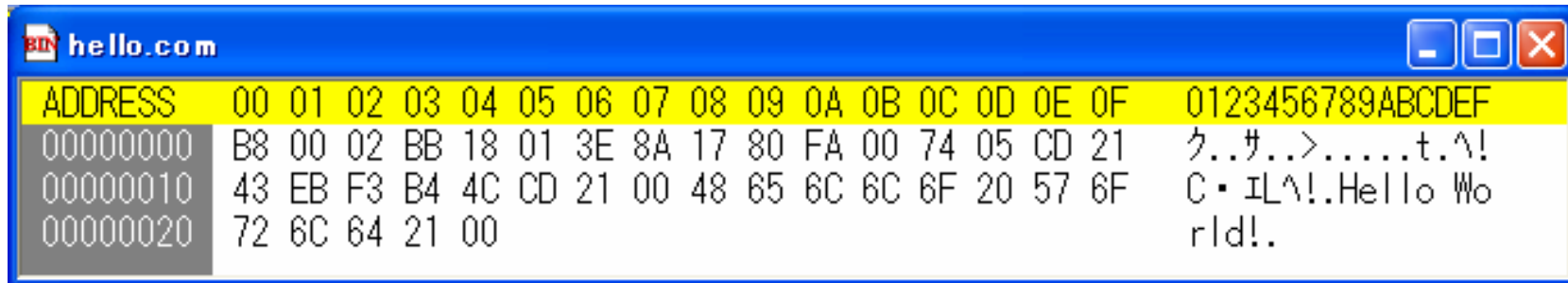
16bit コマンドプロンプト (.comファイル実行時など)

システム管理モード

電源管理、ハードウェア制御

仮想8086モード(壱)

仮想8086モードで動作するプログラム



The screenshot shows a debugger window titled 'hello.com'. The main area displays a memory dump with the following content:

ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	B8	00	02	BB	18	01	3E	8A	17	80	FA	00	74	05	CD	21	ク..サ..>.....t.^!
00000010	43	EB	F3	B4	4C	CD	21	00	48	65	6C	6C	6F	20	57	6F	C・IL^!.Hello Wo
00000020	72	6C	64	21	00												rld!.

仮想8086モード(弐)

コマンドプロンプトからcomファイル実行



```
C:\> コマンド プロンプト
Microsoft (R) KKCFUNC バージョン 1.10
Copyright (C) Microsoft Corp. 1991,1993. All rights reserved.

KKCFUNC が組み込まれました。

マイクロソフトかな漢字変換 バージョン 2.51
(C)Copyright Microsoft Corp. 1992-1993

C:¥>
C:¥>hello.com
Hello World!
C:¥>
```

x86

x86は以下の4つのモードを持つ

リアル・モード

ブートローダー

プロテクト・モード

OS本体、通常のアプリケーション

仮想8086モード

16bit コマンドプロンプト (.comファイル実行時など)

システム管理モード

電源管理、ハードウェア制御

システム管理モード

システムに接続されているハードウェアからの外部割り込みSMI(システム管理割り込み)によりこのモードへ移行し、割り込みハンドラが実行される

詳しい説明は以下参照

SMM(システム管理モード)を悪用した見えない攻撃

<http://itpro.nikkeibp.co.jp/article/COLUMN/20091004/338335/>

SMRAM

- 基本的に、システム管理モードからしかアクセスできないメモリ領域
- 物理アドレスA0000～BFFFFFFまでを占有しているが、これはVGAのメモリ領域と競合
- プロセッサの動作モードが「システム管理モード」の場合にのみ、MCH(メモリー・コントローラ・ハブ)によって転送先がSMRAMになる仕組み

SMM Rootkits

- SMRAM内にあるSMI(システム管理割り込み)をフックするRootkit
- 「システム管理モードからしかアクセスできない」という制限を解除し、システム内に侵入

詳しい説明は以下参照

SMM Rootkits: A New Breed of OS Independent Malware
<http://www.cs.ucf.edu/~czou/research/SMM-Rootkits-Securecom08.pdf>

SMRAMメモリフォレンジック(壺)

CRASH DUMPで完全メモリダンプを取得

マイコンピュータ(右ク)

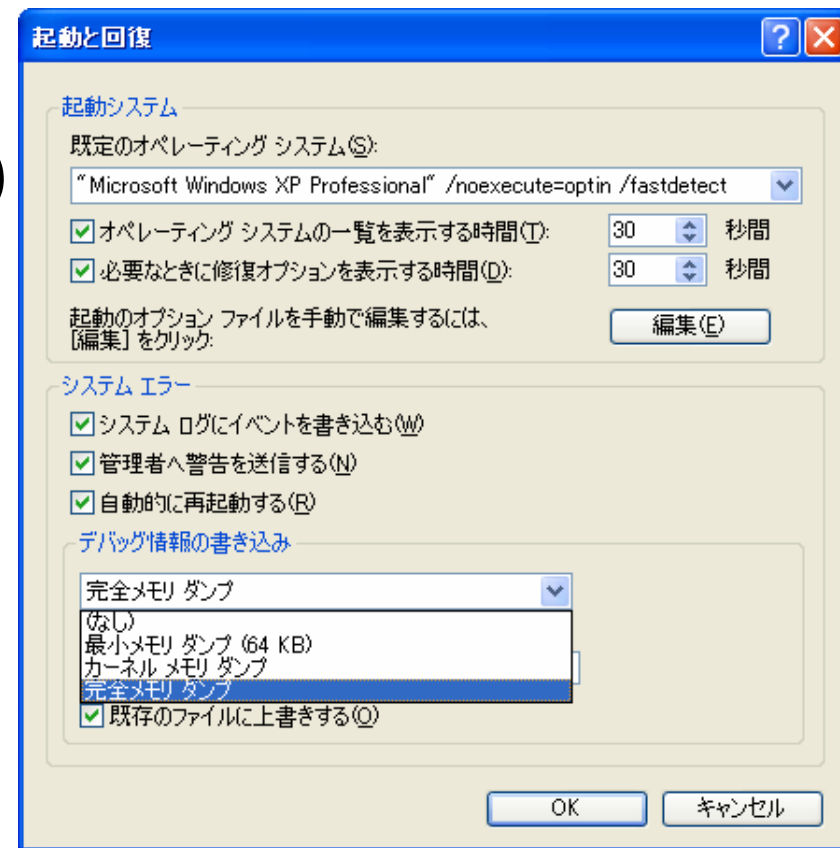
プロパティ

詳細設定

起動と回復

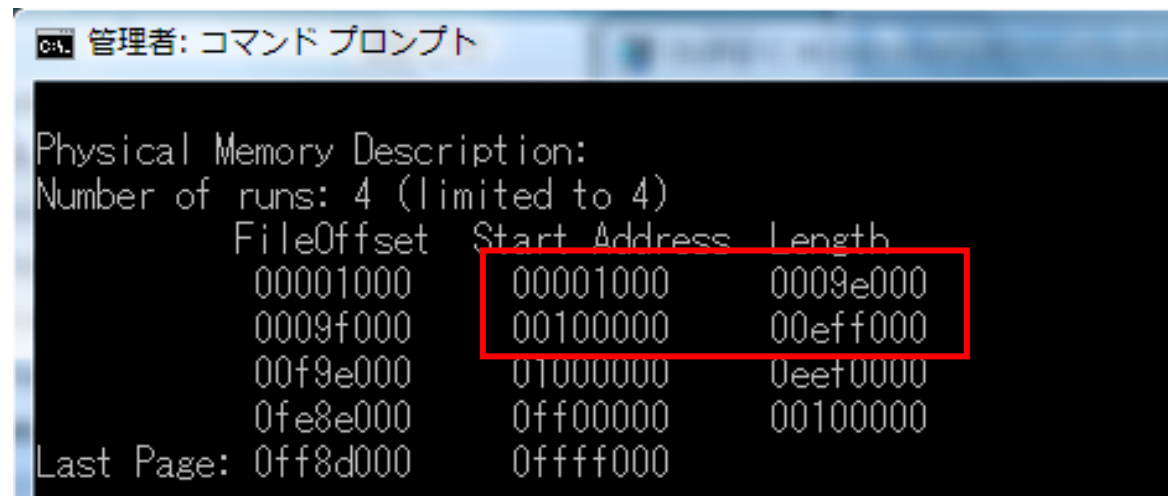
そして

OS CRASH!



SMRAMメモリフォレンジック(弐)

作成されるMEMORY.DMPファイルを解析
環境はVMWare Player 2.5.3 build-185404
WindowsXP SP3
dumpchk.exeの出力結果

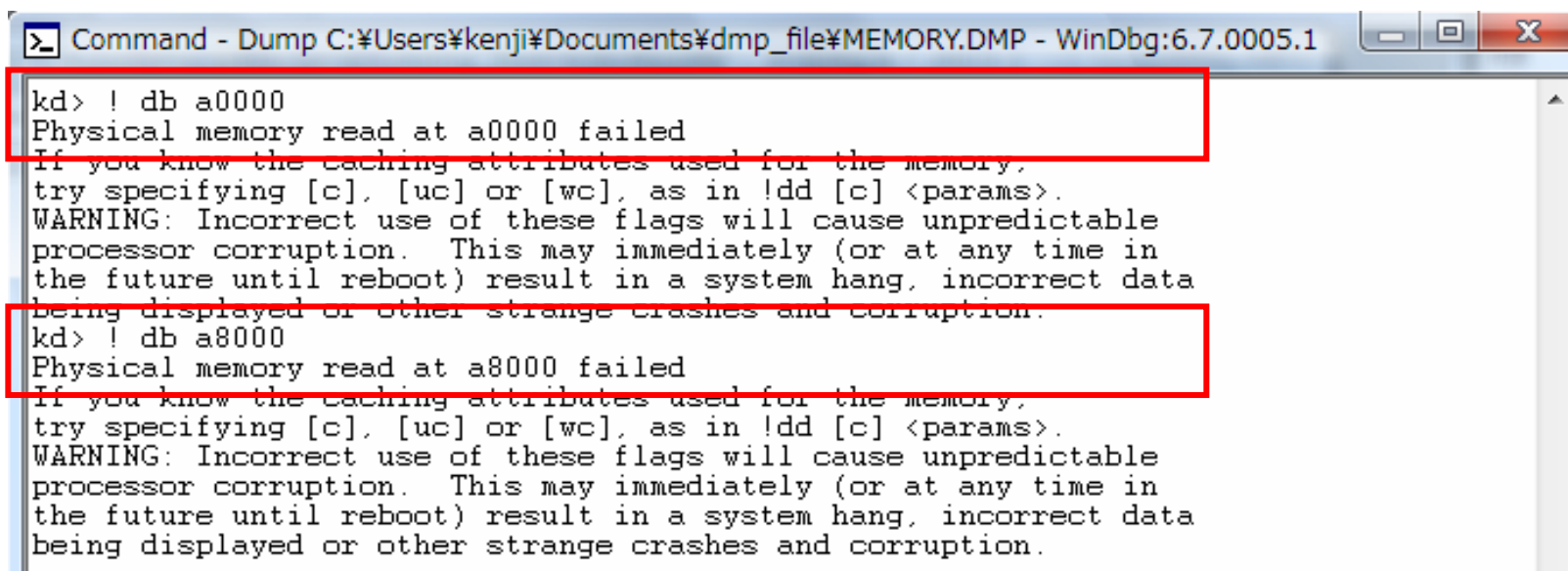


```
管理者: コマンド プロンプト

Physical Memory Description:
Number of runs: 4 (limited to 4)
FileOffset  Start Address  Length
00001000    00001000    0009e000
0009f000    00100000    00eff000
00f9e000    01000000    0eef0000
0fe8e000    0ff00000    00100000
Last Page: 0ff8d000    0ffff000
```

SMRAMメモリフォレンジック(参)

WinDbgで解析

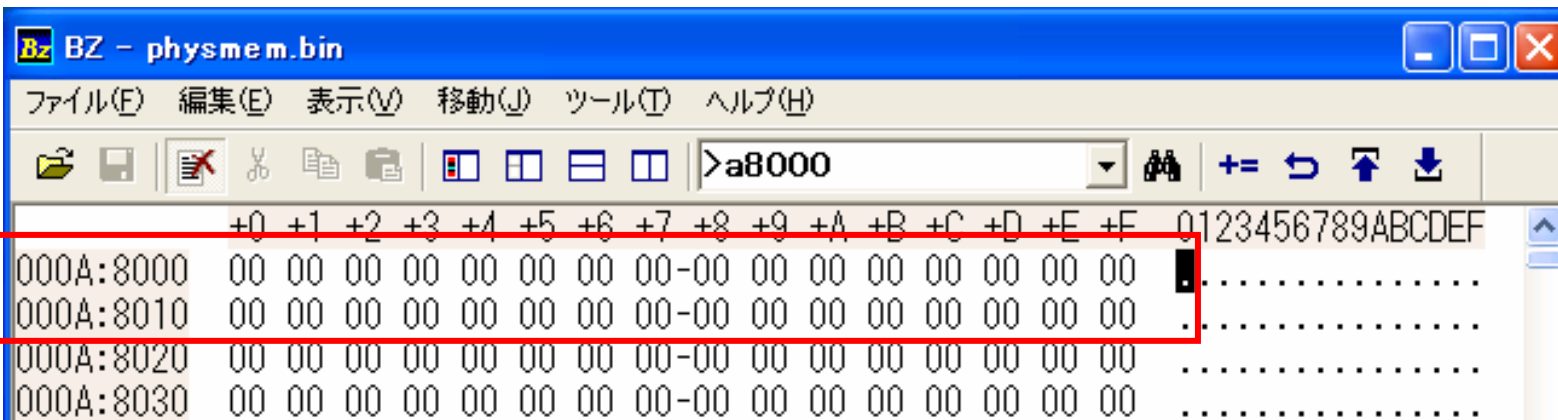


```
Command - Dump C:\Users\kenji\Documents\dmp_file\MEMORY.DMP - WinDbg:6.7.0005.1
kd> ! db a0000
Physical memory read at a0000 failed
If you know the caching attributes used for the memory,
try specifying [c], [uc] or [wc], as in !dd [c] <params>.
WARNING: Incorrect use of these flags will cause unpredictable
processor corruption. This may immediately (or at any time in
the future until reboot) result in a system hang, incorrect data
being displayed or other strange crashes and corruption.
kd> ! db a8000
Physical memory read at a8000 failed
If you know the caching attributes used for the memory,
try specifying [c], [uc] or [wc], as in !dd [c] <params>.
WARNING: Incorrect use of these flags will cause unpredictable
processor corruption. This may immediately (or at any time in
the future until reboot) result in a system hang, incorrect data
being displayed or other strange crashes and corruption.
```

残念...orz

SMRAMメモリフォレンジック(四)

win32dd.exeで物理メモリをそのままダンプ



```
BZ - physmem.bin
ファイル(F) 編集(E) 表示(V) 移動(J) ツール(T) ヘルプ(H)
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F 0123456789ABCDEF
000A:8000 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000A:8010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000A:8020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
000A:8030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

残念...orz

SMRAMメモリフォレンジック(伍)

文献に書かれてある通りシステム管理モードでなければ、SMRAMはダンプできないようなので、Rootkitと同じ方法(?)で再挑戦

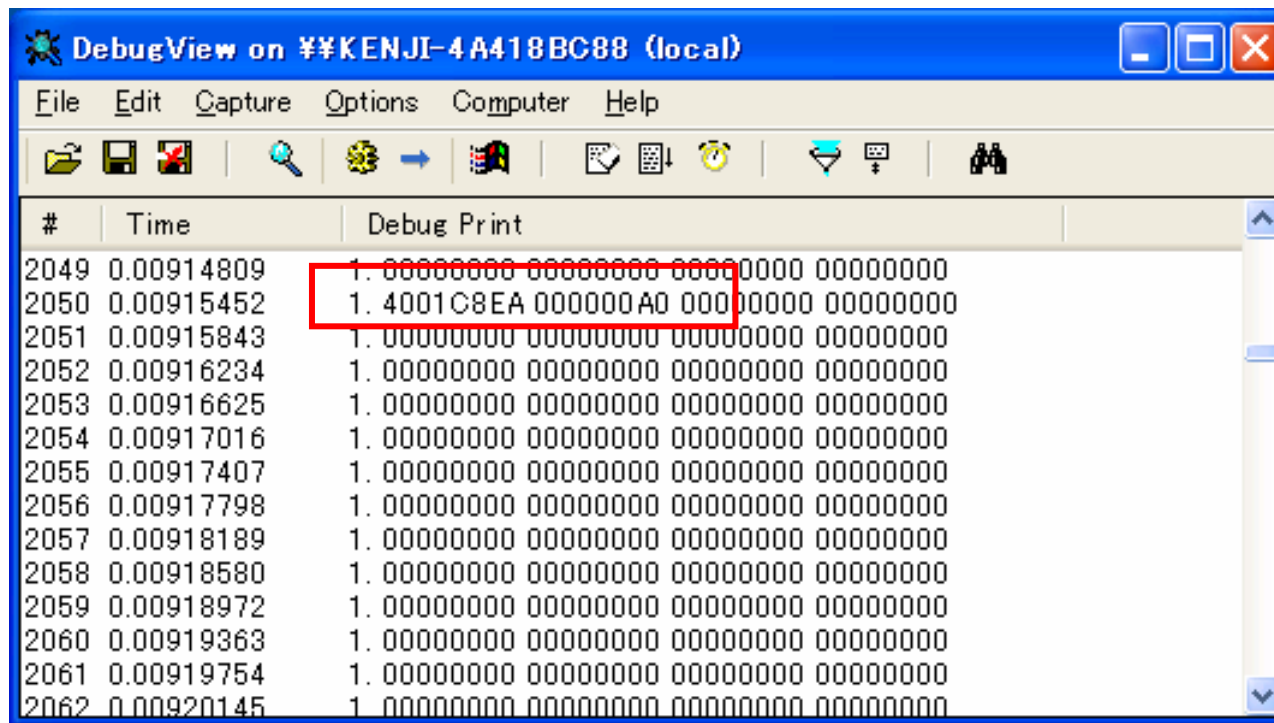
SMRAMメモリフォレンジック(伍)

文献に書かれてある通りシステム管理モードでなければ、SMRAMはダンプできないようなので、Rootkitと同じ方法(?)で再挑戦

1. MCHからSMRAM制御レジスタ値を取得
2. D_OPEN_BITをONにして書き込み
3. 物理アドレスA0000~BFFFFFFがSMRAMに
4. SMRAM領域を読み込み

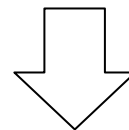
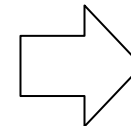
SMRAMメモリフォレンジック(六)

kernelモジュールから、Rootkitと同じ手法でSMRAMへアクセス後、DebugViewに出力



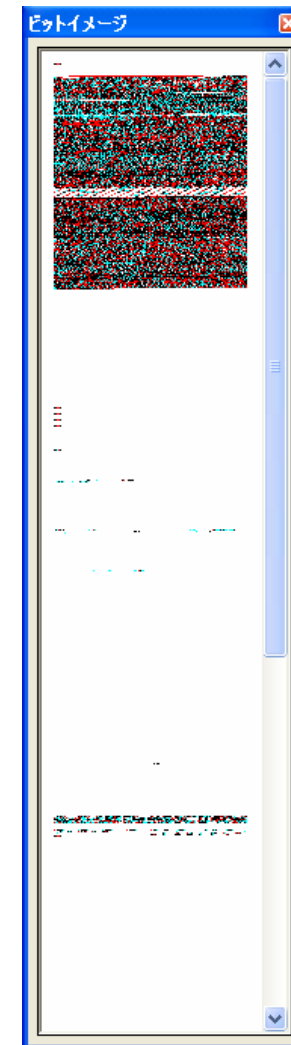
SMRAMメモリフォレンジック(七)

- ダンプデータのビットイメージ
- ダンプデータの先頭



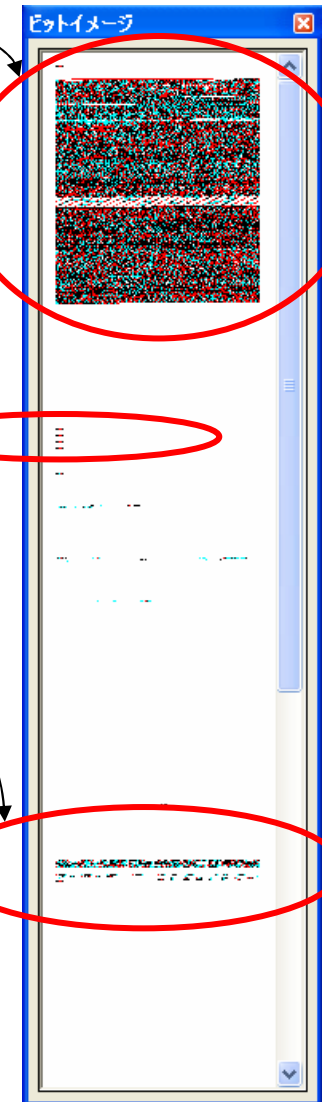
```
smram.bin
ADDRESS 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
00000000 EA 8C 36 40 A0 00 00 00 00 00 00 00 00 00 00  鎧6@.....
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

<http://07c00.com/tmp/smram.bin>



SMRAMメモリフォレンジック(八)

- 16bitアセンブルコード
- jmp命令？(EA8C3640A0)
- 何かのテーブル？ コンテキスト？



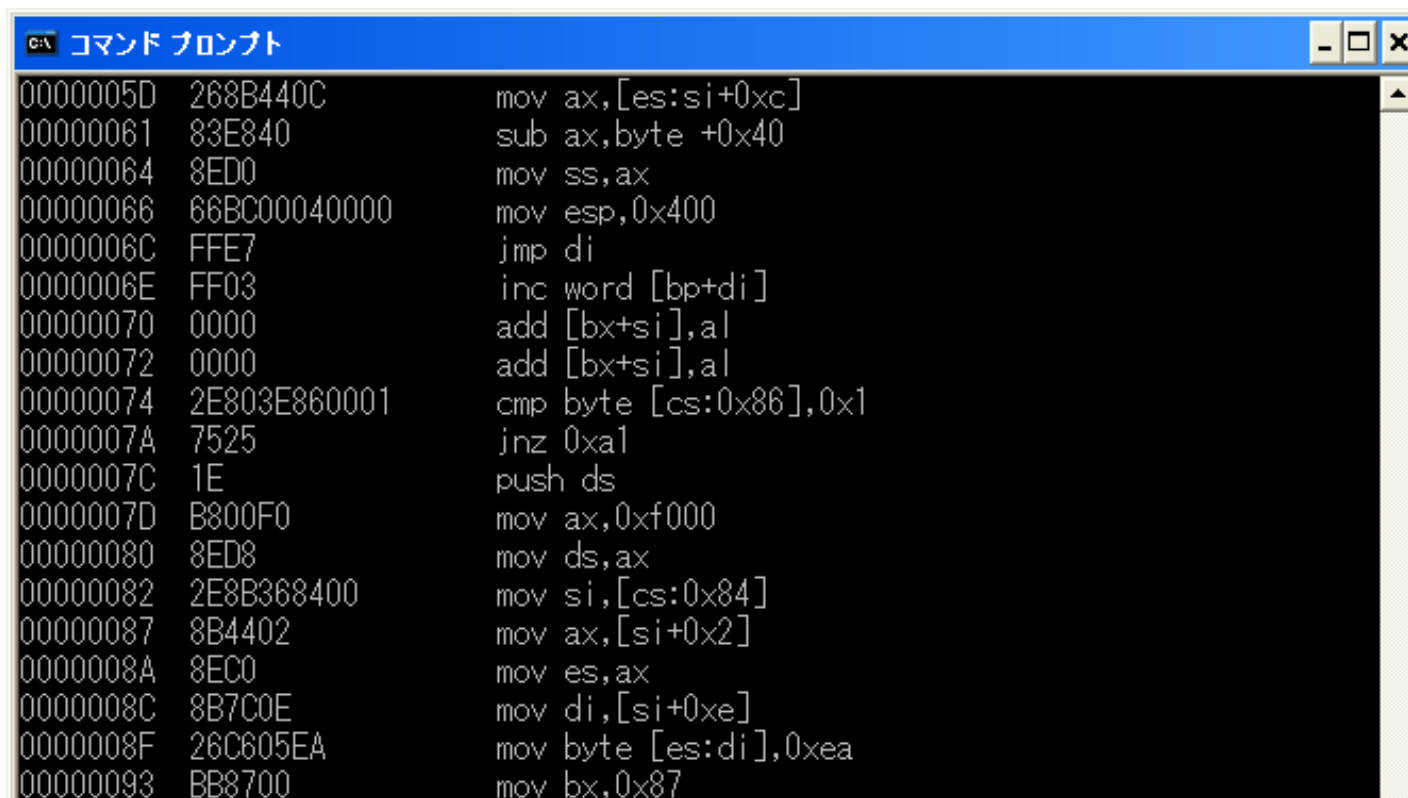
SMRAMメモリフォレンジック(九)

こんな文字列も...

ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
000003F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000400	00	00	00	00	00	00	00	00	00	00	50	68	6F	65	6E	69Phoenix
00000410	78	20	50	6F	77	65	72	4D	61	6E	61	67	65	6D	65	6E	x PowerManagemen
00000420	74	2C	43	6F	70	79	72	69	67	68	74	20	31	39	38	35	t, Copyright 1985
00000430	2D	31	39	39	38	2C	50	68	6F	65	6E	69	78	20	54	65	-1998, Phoenix Te
00000440	63	68	6E	6F	6C	6F	67	69	65	73	20	4C	74	64	2E	41	chnologies Ltd. A
00000450	6C	6C	20	72	69	67	68	74	73	20	72	65	73	65	72	76	ll rights reserv
00000460	65	64	2E	00	00	00	00	00	00	00	00	00	00	00	00	00	ed.....

SMRAMメモリフォレンジック(十)

読めば何か分かるかも...



```
コマンド プロンプト
0000005D 268B440C      mov ax,[es:si+0xc]
00000061 83E840        sub ax,byte +0x40
00000064 8ED0          mov ss,ax
00000066 66BC00040000  mov esp,0x400
0000006C FFE7          jmp di
0000006E FF03          inc word [bp+di]
00000070 0000          add [bx+si],al
00000072 0000          add [bx+si],al
00000074 2E803E860001  cmp byte [cs:0x86],0x1
0000007A 7525          jnz 0xa1
0000007C 1E           push ds
0000007D B800F0        mov ax,0xf000
00000080 8ED8          mov ds,ax
00000082 2E8B368400    mov si,[cs:0x84]
00000087 8B4402        mov ax,[si+0x2]
0000008A 8EC0          mov es,ax
0000008C 8B7C0E        mov di,[si+0xe]
0000008F 26C605EA      mov byte [es:di],0xea
00000093 BB8700        mov bx,0x87
```

まとめ

- SMRAMはメモリダンプツールでは見えない
- サイズは20000hバイトある
- Rootkit的には楽しい空間(SMIフックとか)
- 最近是对策されつつあるらしい...

ご清聴、有難うございました m(_ _)m

Any Questions?