斜め下を行く
バイナリ書き換えの探求

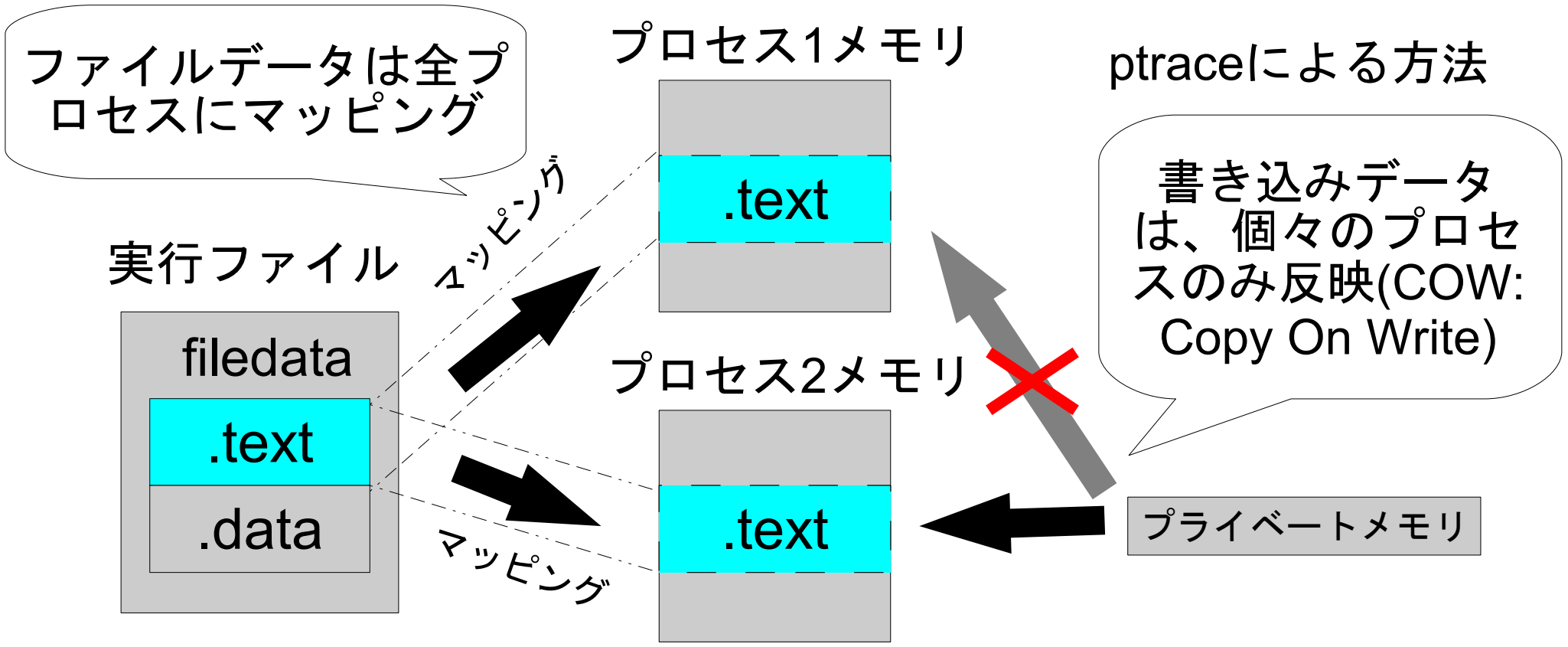
後藤 正徳
gotom@debian.org

2005-12-02 binary 2.0 conference

動作中のプロセスを書き換える技術は様々

- 一例:
  - 自己書き換えプログラム
  - バイナリパッチ（ptraceなど）
- Pannusのアレゲさとlivepatchのスマートさにインスパイア
- もっと手軽で斜め下の方法はないか？

実行中のプロセス書き換えとは？

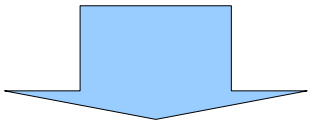


- 普通の方法：個々のプロセスメモリを変更(右側)
- ファイルから変更をかけるとどうか？(左側)

ウィザードたるもの…

- “Unixの階層”より

  『ウィザードは、
  バイナリにパッチを当ててバグ修正する』



そこで…

- ディスク上の実行ファイルをエディタで編集することで、
- プロセスメモリも修正してみよう
- しかし…

実行プロセスを持つファイルの変更

- 次のような経験はありますか?（Linux/BSDの場合）

```
/tmp> ./sleep 1000 &
[1] 21733
/tmp> echo aaa >> ./sleep
./sleep: Text file busy.
```

- ユーザのミスを防ぐため、open(O_RDWR or O_WRONLY)でETXTBSYエラー
- 今は余計なお世話

この制限を取り払うよう Linuxカーネルを変更

```
--- linux/include/linux/mman.h
+++ linux/include/linux.hack/mman.h
 calc_vm_flag_bits(unsigned long flags)
 {
      return _calc_vm_trans(flags, MAP_GROWSDOWN,  VM_GROWSDOWN ) |
-            _calc_vm_trans(flags, MAP_DENYWRITE,  VM_DENYWRITE ) |
             _calc_vm_trans(flags, MAP_EXECUTABLE, VM_EXECUTABLE) |
             _calc_vm_trans(flags, MAP_LOCKED,     VM_LOCKED    );
 }
--- linux/include/asm-i386/mman.h
+++ linux/include/asm-i386.hack/mman.h
 #define MAP_GROWSDOWN  0x0100      /* stack-like segment */
+#if 0
 #define MAP_DENYWRITE  0x0800      /* ETXTBSY */
+#else
+#define MAP_DENYWRITE  0x0         /* ETXTBSY */
+#endif
 #define MAP_EXECUTABLE 0x1000      /* mark it as an executable */
```

- （余談：後で気づいたが、実はSolarisのデフォルトカーネルで何も変更しなくて良いらしい）

サンプルで効果を試そう

プログラム例:

```
#include <time.h>
#include <stdio.h>
int main(void)
{
        int a = 0;
        struct timespec t = {1 , 0};
        while (1) {
                if (a == 0) {
                        printf("sleep...\n");
                        (void) nanosleep(&t, NULL);
                } else {
                        printf("succeeded\n");
                        break;
                }
        }
}
```

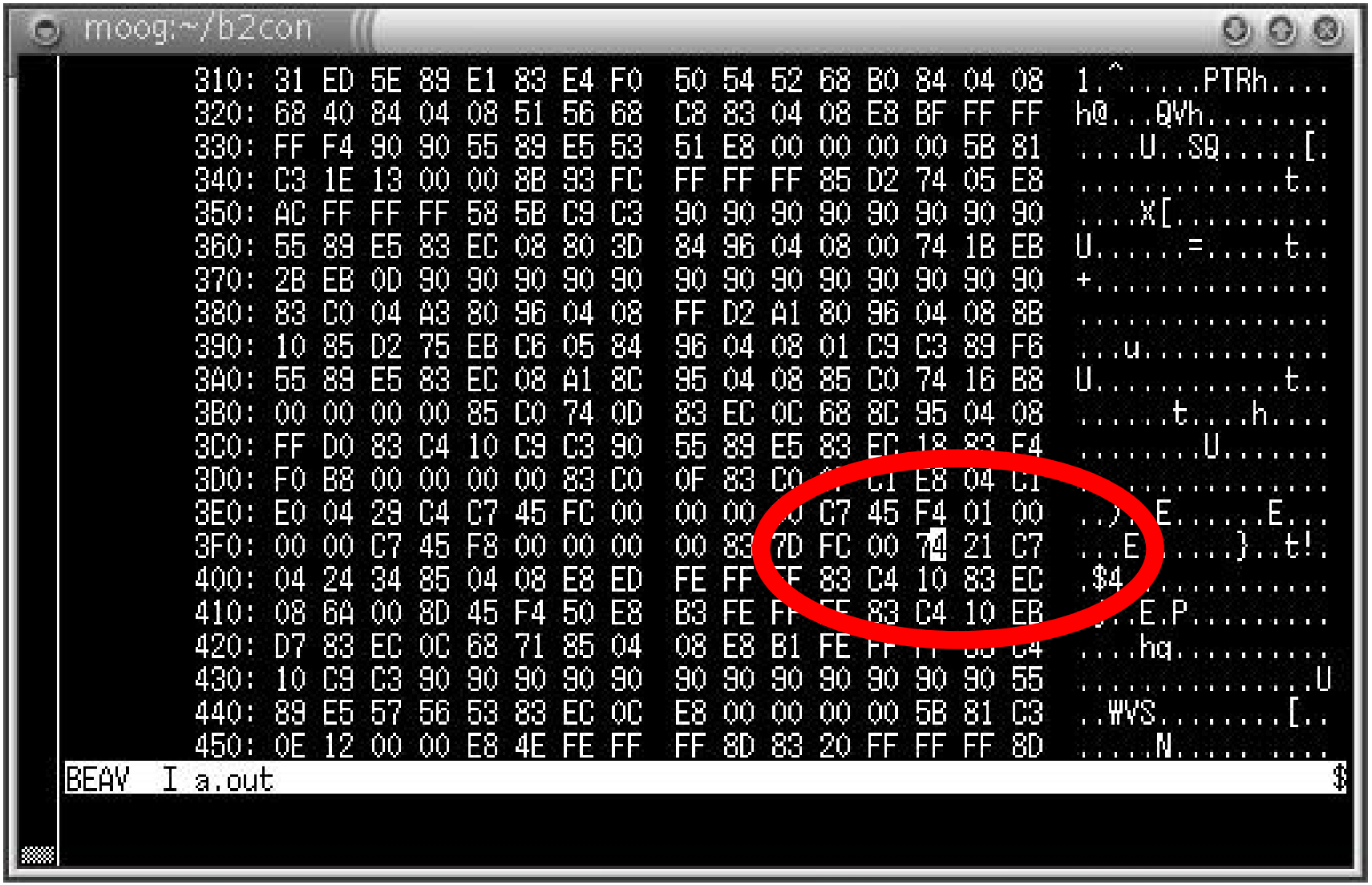
実行例:

```
~/> ./a.out
sleep...
sleep...
sleep...
sleep...
```

試しにこの行を
(a == 0)
↓
(a != 0)
に変更してみる

Let's バイナリエディット!

> beav ./a.out



実行プロセスを持つバイナリファイルをエディット

```
3c4 <main>:
3c4:      55                      push   %ebp
3c5:      89 e5                   mov    %esp,%ebp
3c7:      83 ec 18                sub    $0x18,%esp
3ca:      83 e4 f0                and    $0xfffffff0,%esp
3cd:      b8 00 00 00 00          mov    $0x0,%eax
3d2:      29 c4                   sub    %eax,%esp
3d4:      c7 45 fc 00 00 00 00    movl   $0x0,0xfffffffc(%ebp)
3db:      c7 45 f0 01 00 00 00    movl   $0x1,0xfffffff0(%ebp)
3e2:      c7 45 f4 00 00 00 00    movl   $0x0,0xfffffff4(%ebp)
3e9:      83 7d fc 00             cmpl   $0x0,0xfffffffc(%ebp)
3ed:      74 21                   je     8048410 <main+0x4c>
3ef:      c7 04 24 34 85 04 08    movl   $0x8048534,(%esp)
3f6:      e8 ed fe ff ff          call   80482e8 <_init+0x48>
3fb:      c7 44 24 04 00 00 00    movl   $0x0,0x4(%esp)
```

この行を
「74 je」
から
「75 jne」
に変更

~/> binary-editor ./a.out

- 実行バイナリを編集
- 0x3edを75に変更、保存
  →うまくいった！

実行例:

```
~/> ./a.out
sleep...
sleep...
sleep...
sleep...
succeeded
~/>
```

注意する点など

- エディタにはよってはうまくいかないことも…
  - beav: 保存先ファイルは別inodeになるためプログラム実行前に編集開始しておく
  - bvi: open(2)時O_TRUNCするため、ファイルデータが消えてプロセスがクラッシュ
- 実行属性領域（.text）以外のセクション（.bss, heap, stack）の変更は不可能
- コード領域が足りない場合、何とか頑張る：
  - すき間を使う（不要なコード・_start・デバッグ情報など）、自前でmmapするなど

最後に

- 実行プロセスがあるバイナリファイルも編集可能なLinuxカーネルを作成
  - ファイルの書き換えで、実行プロセスすべてを一瞬で変更（ptraceとは異なり高速）
  - ちゃんと動作
  - セキュリティもパーミッションで万全
- バイナリパッチを当てるバイナリアン・ウィザード御用達
  - 手軽にクラッシュさせる諸刃の剣
  - 素人にはおすすめできない