Voyage (rapide) au cœur de la mémoire

Damien AUMAITRE
damien(at)security-labs.org
damien.aumaitre(at)sogeti.com

Introduction

- M. Dornseif en 2005 et A. Boileau en 2006 montrent comment compromettre un poste utilisateur en utilisant le bus Firewire.
- Pendant mon stage, reproduction des démos.
- Les connaissances acquises ont débouché sur le développement d'un outil permettant l'analyse et la reconstruction de la mémoire virtuelle à partir de la mémoire physique.

Pourquoi utiliser la mémoire physique ?

Avantages

- Vue de la mémoire indépendante de l'API du système d'exploitation.
- Multiples points d'accès à la représentation de la mémoire physique.

Inconvénients

- Besoin de reconstruire l'espace d'adressage virtuel pour retrouver les données.
- Structures très dépendantes du système d'exploitation.

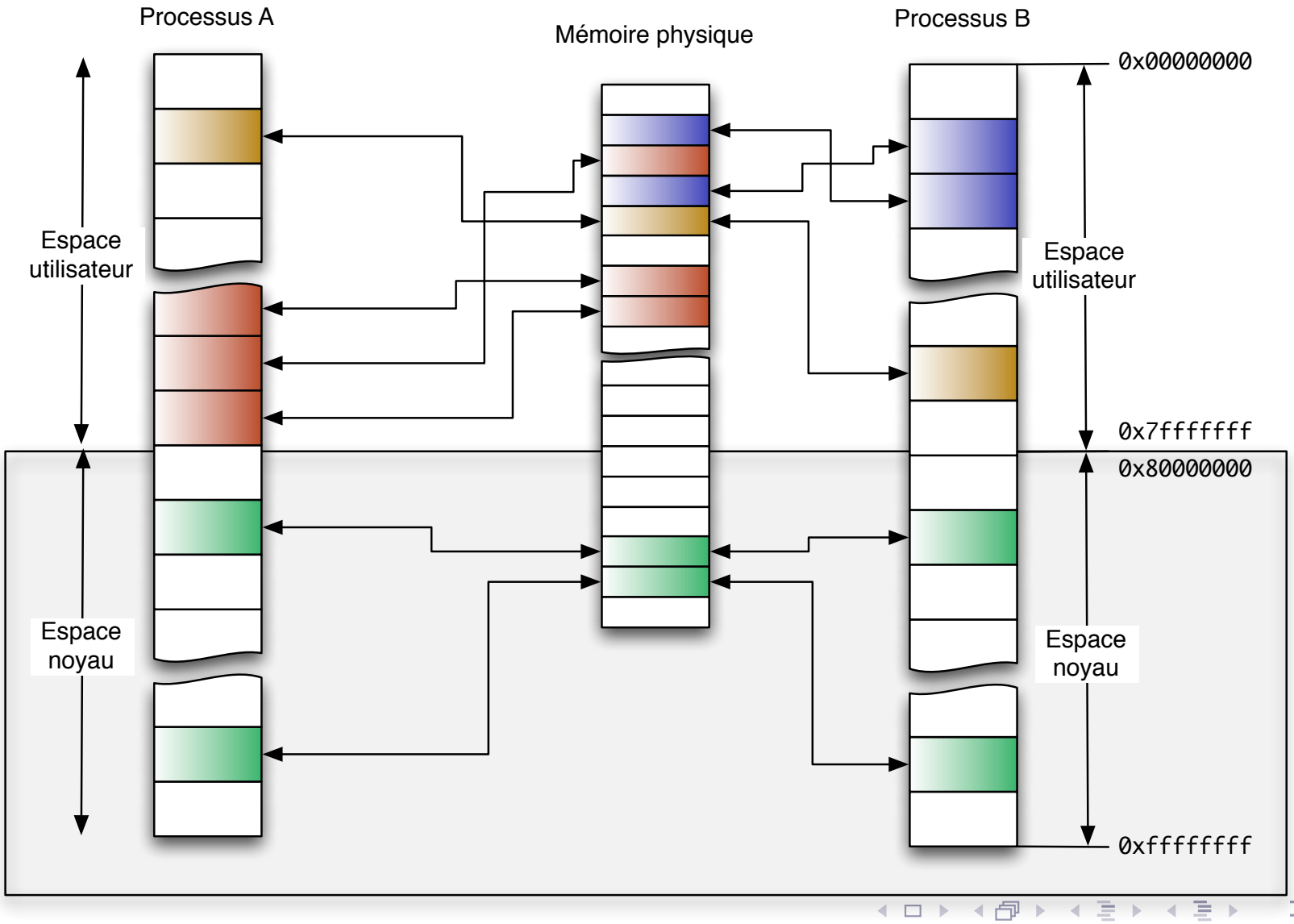
Plan

1. Organisation de la mémoire
   - Segmentation / pagination
   - Reconstruction de la mémoire virtuelle
2. Comment accéder à la mémoire physique ?
3. RWX

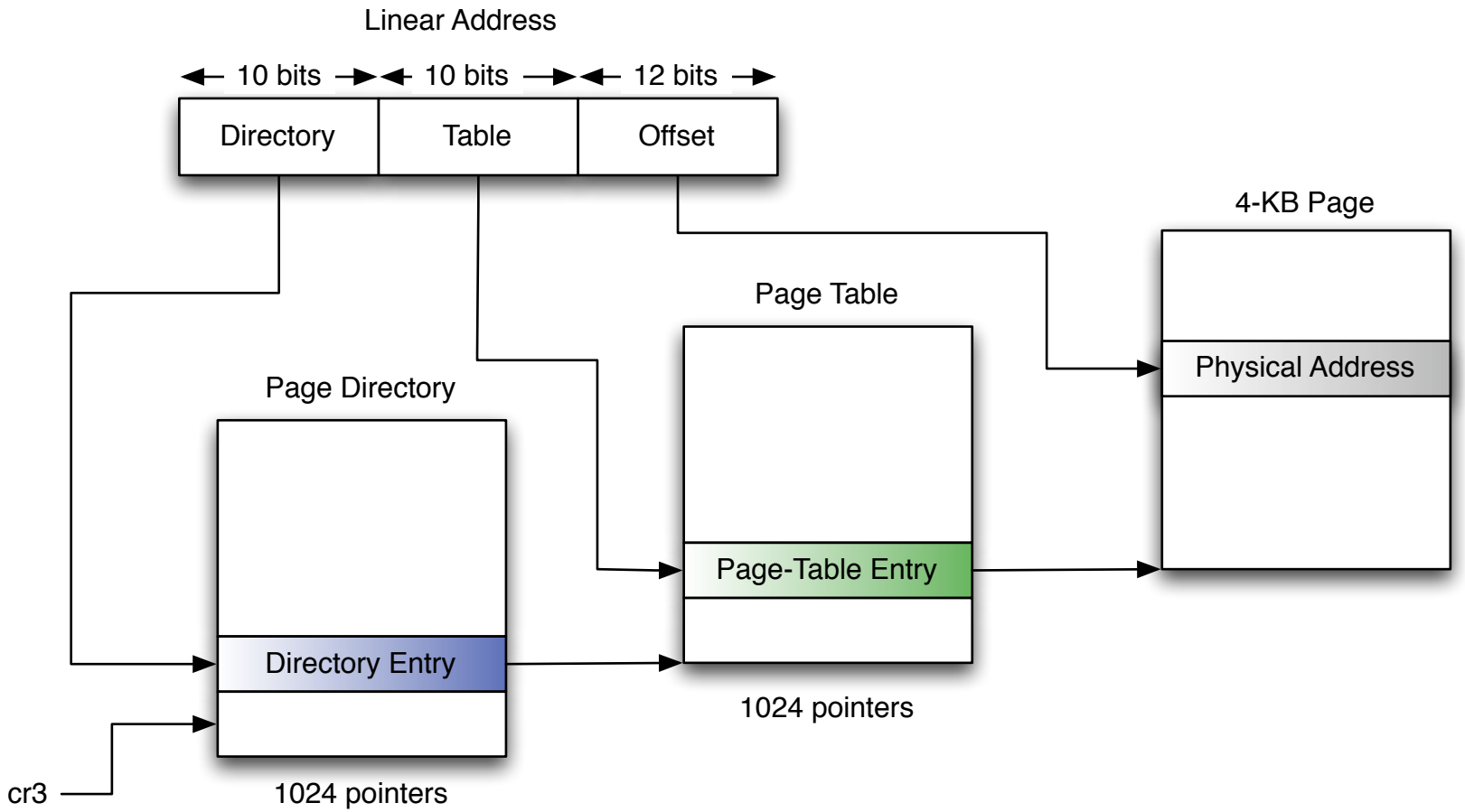
Plan

1. Organisation de la mémoire
   - Segmentation / pagination
   - Reconstruction de la mémoire virtuelle
2. Comment accéder à la mémoire physique ?
3. RWX

Correspondance entre adresse virtuelle et adresse physique



Correspondance entre adresse virtuelle et adresse physique



Plan

1. Organisation de la mémoire
   - Segmentation / pagination
   - Reconstruction de la mémoire virtuelle
2. Comment accéder à la mémoire physique ?
3. RWX

Registre cr3 ?

- Utilisé dans le mécanisme de conversion d'adresse.
- Permet de mapper l'intégralité de l'espace virtuel d'un processus.
- Stocké dans la structure _KPROCESS dans le champ DirectoryTableBase.

```
typedef struct _KPROCESS  // 29 elements, 0x6C bytes (sizeof)
{
/*0x000*/     struct _DISPATCHER_HEADER Header;  // 6 elements, 0x10 bytes (sizeof)
/*0x010*/     struct _LIST_ENTRY ProfileListHead;  // 2 elements, 0x8 bytes (sizeof)
/*0x018*/     ULONG32      DirectoryTableBase[2];
[...]
}KPROCESS, *PKPROCESS;
```

Retrouver la structure _KPROCESS

- La structure _DISPATCHER_HEADER a une signature particulière
- Les champs Type et Size ont des valeurs fixes pour chaque version de Windows.
- Par exemple, pour Windows XP SP2, Type = $0x3$ et Size = $0x1b$.

```
typedef struct _DISPATCHER_HEADER     // 6 elements, 0x10 bytes (sizeof)
{
/*0x000*/     UINT8          Type;
/*0x001*/     UINT8          Absolute;
/*0x002*/     UINT8          Size;
/*0x003*/     UINT8          Inserted;
/*0x004*/     LONG32         SignalState;
/*0x008*/     struct _LIST_ENTRY WaitListHead; // 2 elements, 0x8 bytes (sizeof)
}DISPATCHER_HEADER, *PDISPATCHER_HEADER;
```

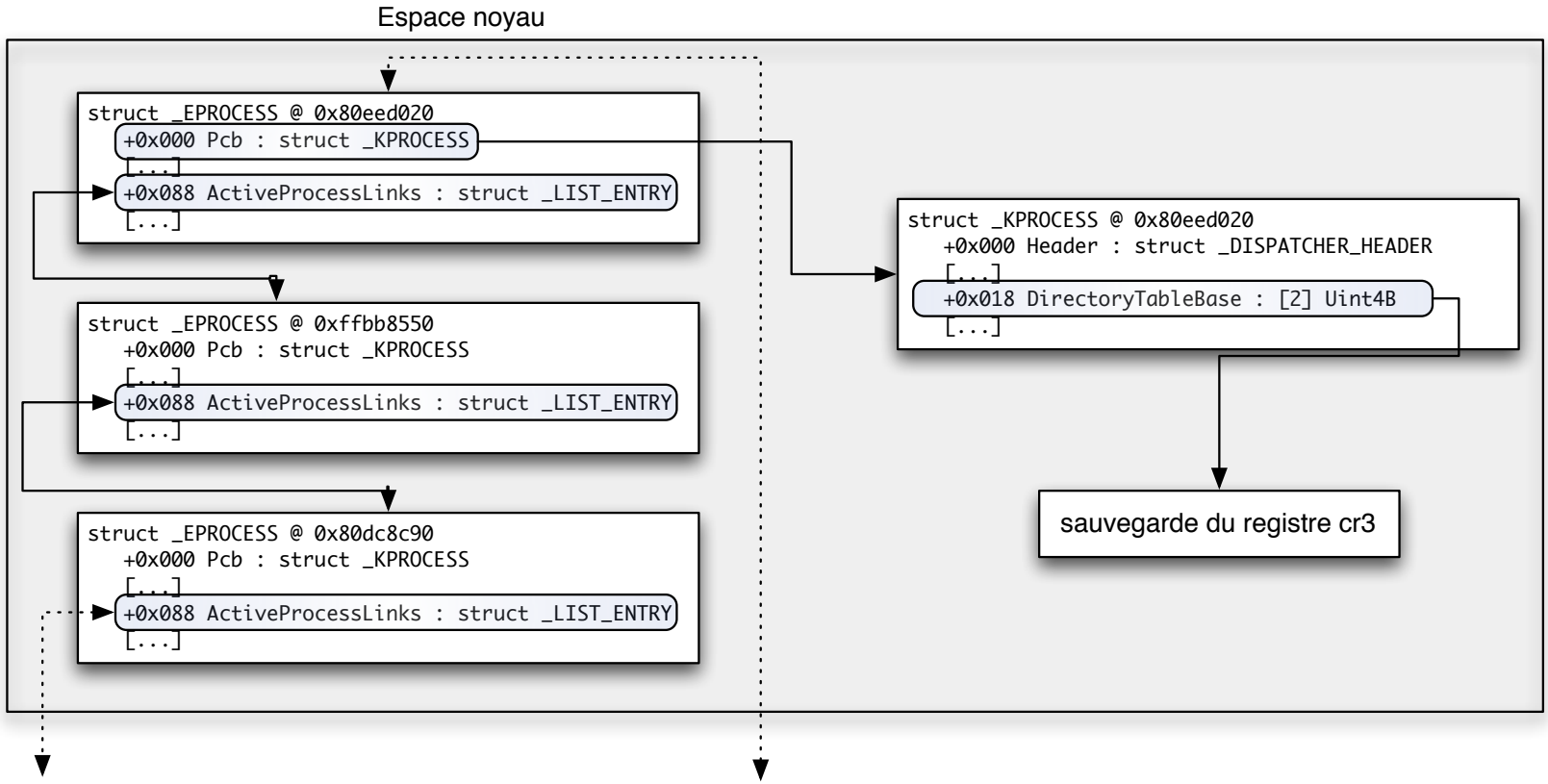
Retrouver la structure _KPROCESS

Méthode proposée par Andreas Schuster

Principe

- Scanner la mémoire en cherchant des structures _DISPATCHER_HEADER
- Valider les candidats potentiels en vérifiant la cohérence des champs de la structure (par exemple, les structures noyau doivent pointer en espace noyau).

Retrouver la structure _KPROCESS



Conclusion

Résultat

- Équivalence entre mémoire physique et mémoire virtuelle
- Traduction de l'espace virtuel de TOUS les processus.

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
   - Moyens
   - Zoom sur le firewire
3. RWX

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
   - Moyens
   - Zoom sur le firewire
3. RWX

Comment accéder à la mémoire physique ?

Plusieurs moyens :

- Firewire
- VMWare
- Fichiers d'hibernation (via Sandman par exemple)
- Coldboot attacks
- Outils de forensics etc.

cf. SSTIC 07, "Autopsie d'une intrusion tout en mémoire sous Windows", Nicolas Ruff

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
   - Moyens
   - Zoom sur le firewire
3. RWX

Zoom sur le firewire

Firewire ?

- Développé par Apple à la fin des années 80 et standardisé par l'IEEE en 1995.
- Permet l'accès à la mémoire physique grâce à l'utilisation du DMA (Direct Memory Access).
- Le DMA est un mécanisme qui décharge le processeur des tâches d'entrées/sorties longues.

Zoom sur le firewire

Accès à la mémoire

- L'accès à la mémoire physique est contrôlé par deux registres dans la mémoire du contrôleur firewire.
- Interdit par défaut sous Windows.
- Sauf pour les périphériques de stockage de masse
  - par exemple un iPod

Transformation en iPod

Spécification OHCI 1394

- Chaque périphérique firewire a une carte d'identité.
- L'identification du périphérique firewire peut être modifiée.

Bibliothèque `libraw1394`

- Bibliothèque en mode utilisateur permettant de manipuler le bus firewire
- Fonction `raw1394_update_config_rom`

Transformation en iPod

Avant

Portable sous Linux.

```
00000000 04 04 0d ef 31 33 39 34 e0 64 a2 32 42 4f c0 00 |....1394.d.2BO..|
00000010 3c c4 44 50 00 03 03 5d 03 42 4f c0 81 00 00 02 |<.DP.....BO.....|
00000020 0c 00 83 c0 00 06 2c 2a 00 00 00 00 00 00 00 00 |......,*........|
00000030 4c 69 6e 75 78 20 2d 20 6f 68 63 69 31 33 39 34 |Linux - ohci1394|
```

Transformation en iPod

Après

Un iPod :)

```
00000000 04 04 72 86 31 33 39 34 00 ff a0 12 00 0a 27 00 |..r.1394......'.|
00000010 02 aa 6b a7 00 04 f9 3c 0c 00 83 c0 03 00 0a 27 |..k....<.......'|
00000020 81 00 00 11 d1 00 00 01 00 0e e5 a0 12 00 60 9e |..............`.|
00000030 13 01 04 83 21 00 00 01 3a 00 0a 08 3e 00 4c 10 |............>.L.|
00000040 38 00 60 9e 39 01 04 d8 3b 00 00 00 3c 0a 27 00 |8.`.9.......<.'.|
00000050 54 00 40 00 3d 00 00 03 14 0e 00 00 17 00 00 21 |T.@.............|
00000060 81 00 00 0a 00 08 96 bc 00 00 00 00 00 00 00 00 |................|
00000070 41 70 70 6c 65 20 43 6f 6d 70 75 74 65 72 2c 20 |Apple Computer,.|
00000080 49 6e 63 2e 00 00 00 00 00 04 34 e7 00 00 00 00 |Inc.......4.....|
00000090 00 00 00 00 69 50 6f 64 00 00 00 00 00 00 00 00 |....iPod........|
```

Transformation en iPod

Conclusion

Windows croit avoir affaire à un iPod et autorise l'accès à la mémoire en lecture/écriture.

Pour plus de détails

Mécanisme expliqué en détail sur le site d'Adam Boileau : `http://storm.net.nz/projects/16`

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
3. RWX
   - Read : rassembler des informations
   - Write : tout est permis
   - eXecute : Welcome to Paradise

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
3. RWX
   - Read : rassembler des informations
   - Write : tout est permis
   - eXecute : Welcome to Paradise

Process Explorer 101

Contexte

Accès en lecture à la mémoire physique

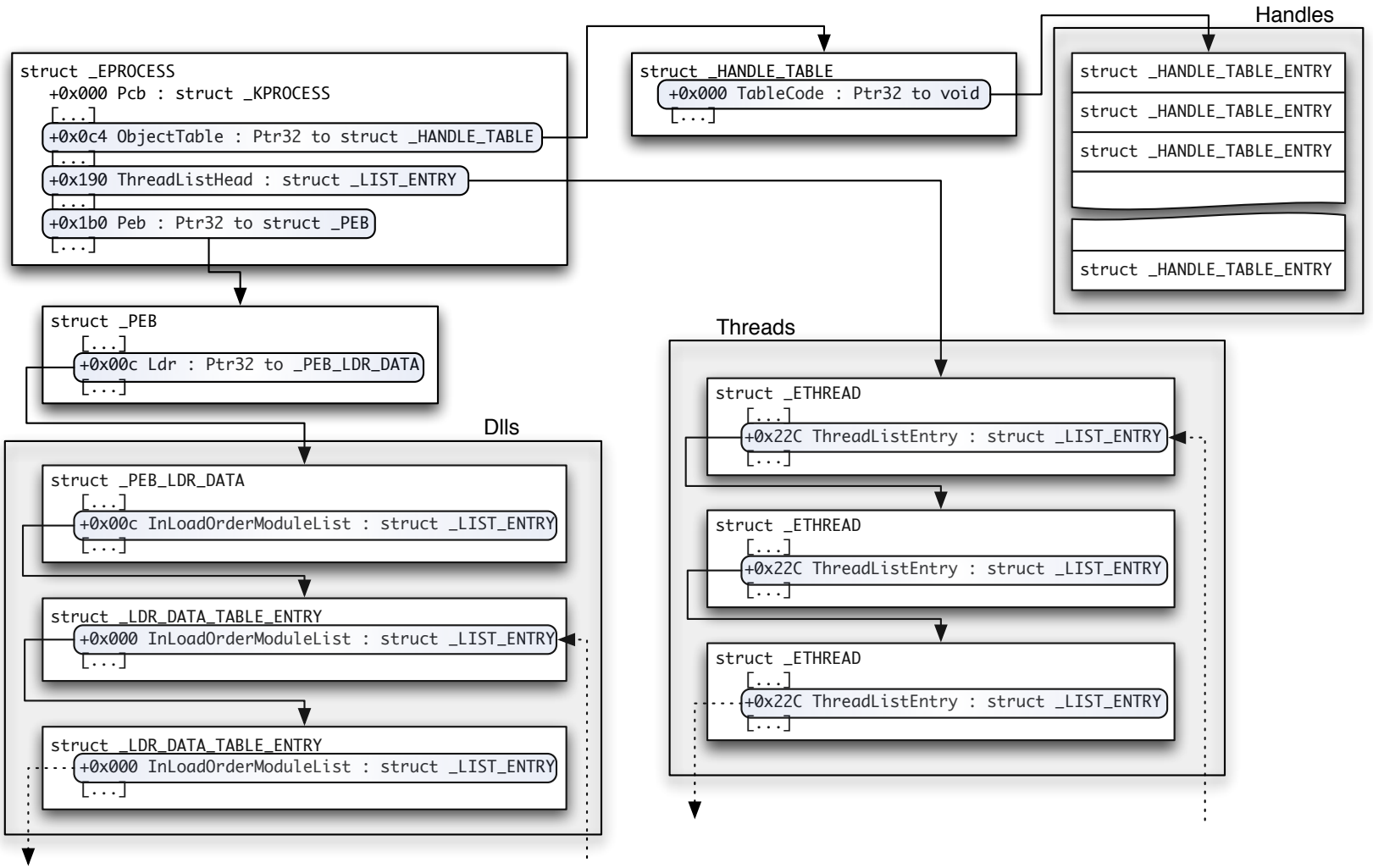
But

Trouver les informations nécessaires à la réalisation d'un clone de Process Explorer

Informations nécessaires

- Processus et threads existants au moment du dump
- Handles ouverts, DLLs chargées en mémoire.

Process Explorer 101



Regedit 101

Contexte

Même contexte que pour Process Explorer

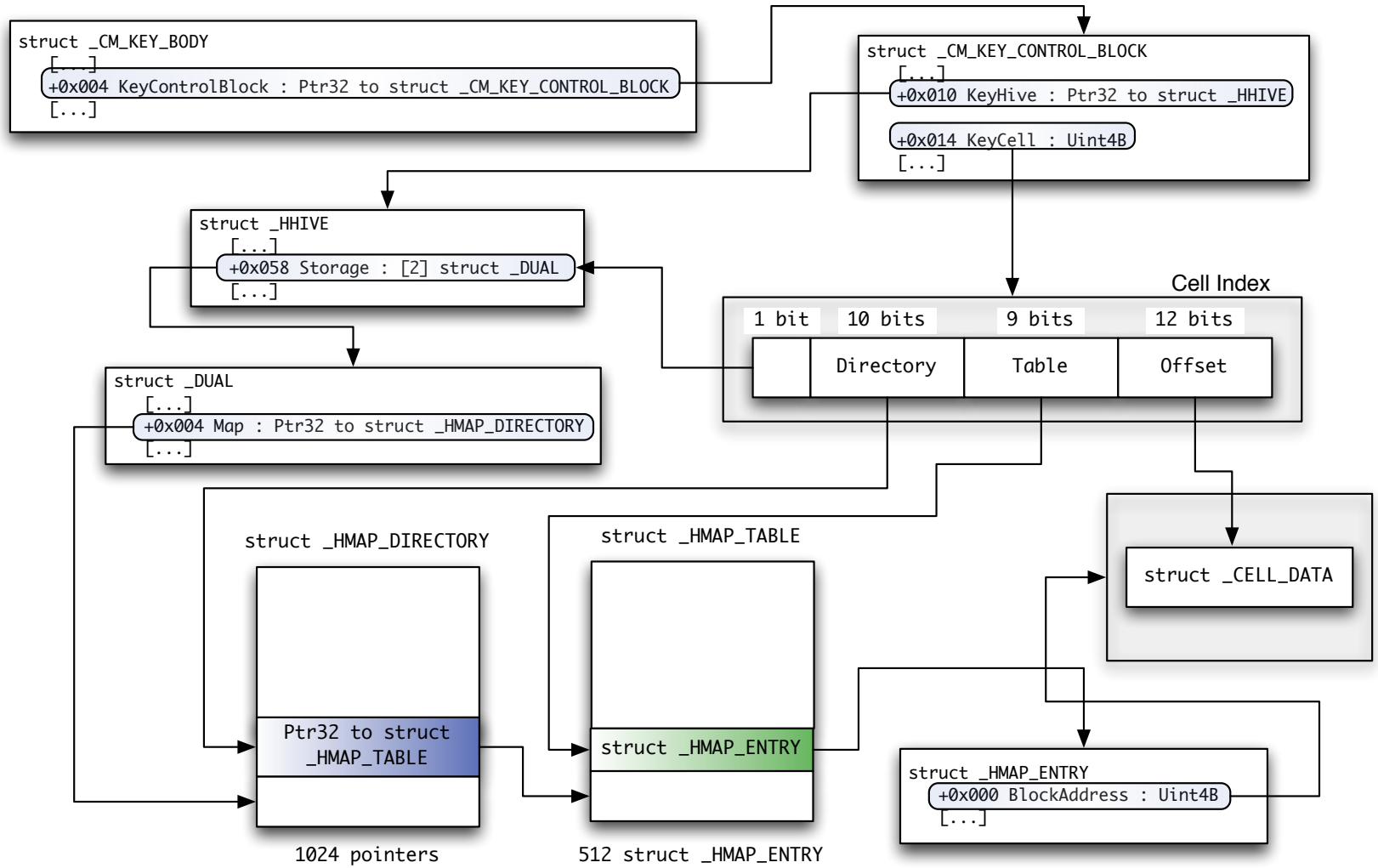
But

Informations pour réaliser un clone de Regedit

Informations nécessaires

Ruches et clés de la base de registres

Regedit 101


struct _CM_KEY_BODY
[...]
+0x004 KeyControlBlock : Ptr32 to struct _CM_KEY_CONTROL_BLOCK
[...]
struct _CM_KEY_CONTROL_BLOCK
[...]
+0x010 KeyHive : Ptr32 to struct _HHIVE
+0x014 KeyCell : Uint4B
[...]
struct _HHIVE
[...]
+0x058 Storage : [2] struct _DUAL
[...]
Cell Index
1 bit
10 bits
9 bits
12 bits
Directory
Table
Offset
struct _DUAL
[...]
+0x004 Map : Ptr32 to struct _HMAP_DIRECTORY
[...]
struct _HMAP_DIRECTORY
struct _HMAP_TABLE
struct _CELL_DATA
Ptr32 to struct _HMAP_TABLE
struct _HMAP_ENTRY
struct _HMAP_ENTRY
+0x000 BlockAddress : Uint4B
[...]
1024 pointers
512 struct _HMAP_ENTRY

Dump de crédentials

- Base SAM.
- Secrets LSA.
- Cache de domaine.

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
3. RWX
   - Read : rassembler des informations
   - Write : tout est permis
   - eXecute : Welcome to Paradise

Se logger sans mot de passe

Contexte

- Accès en lecture/écriture à la mémoire
- Peut-on se logguer sans mot de passe ?

Plusieurs possibilités :

- winlockpwn d'Adam Boileau (patch la fonction responsable de l'authentification)
- ou . . . patch de 2 octets dans la base de registres :)

Plan

1. Organisation de la mémoire
2. Comment accéder à la mémoire physique ?
3. RWX
   - Read : rassembler des informations
   - Write : tout est permis
   - eXecute : Welcome to Paradise

Éxécution de code arbitraire

Contexte

- Accès en lecture-écriture
- Comment exécuter du code ?

Une solution

- Hooker des pointeurs de fonctions

Éxécution de code arbitraire

Quels pointeurs ?

- Structure `_KUSER_SHARED_DATA`
- Champ `SystemCall`
- Appelé avant chaque appel système

Où stocker le code ?

- La structure `_KUSER_SHARED_DATA` a une taille de 334 octets, le reste de la page est disponible

Éxécution de code arbitraire

Comment ça marche ?

- Un seul Desktop peut interagir avec l'utilisateur
- Chaque application a besoin d'un desktop
- Pour un utilisateur, 3 desktops `Default`, `Disconnect` et `Winlogon`
- Pour lancer un `cmd.exe` avant l'authentification, il suffit juste de préciser que le desktop du `cmd` sera celui de `Winlogon`

Conclusion et pistes futures

iPod 101

- Accès physique = root

- Il est possible de reconstruire un instantané du système d'exploitation à partir de la mémoire physique.
- Beaucoup d'applications possibles : forensics, debug, intrusion.
- Pistes futures : support de Linux, interface avec Sandman et utilisation du fichier de swap.

Questions ?

- Merci de votre attention
- Des questions ?