Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
Premier ministre
Secrétariat général de la défense nationale

Quelles limites pour l'informatique de confiance?

Loïc Duflot, Olivier Levillain,

Direction Centrale de la
Sécurité des Systèmes d'Information

SGDN/DCSSI 51 boulevard de la Tour Maubourg 75007 Paris

loic.duflot@sgdn.gouv.fr

Introduction

- Les initiatives visant à proposer des solutions techniques pour accroître la confiance des utilisateurs dans les postes informatiques se multiplient:
  - La plus médiatisée est sans doute celle du *Trusted Computing Group* (TCG).
  - Le TCG définit un socle de confiance qui correspond à l'ensemble des fonctions matérielles et logicielles qui doivent impérativement être correctes pour que l'utilisateur puisse espérer avoir confiance dans une plateforme informatique.
  - Ces initiatives cherchent généralement à minimiser le périmètre du socle de confiance d'une machine informatique de manière à accroître la maîtrise de ce dernier (exemple: technologie Intel® TxT).

- Dans cette présentation nous montrons comment les choix effectués par le passé, notamment en matière de gestion de l'alimentation, sur les plateformes x86 classiques constituent une très forte limite à cette approche.
- Cette réflexion concerne principalement les plateformes x86 et x86-64 classiques.

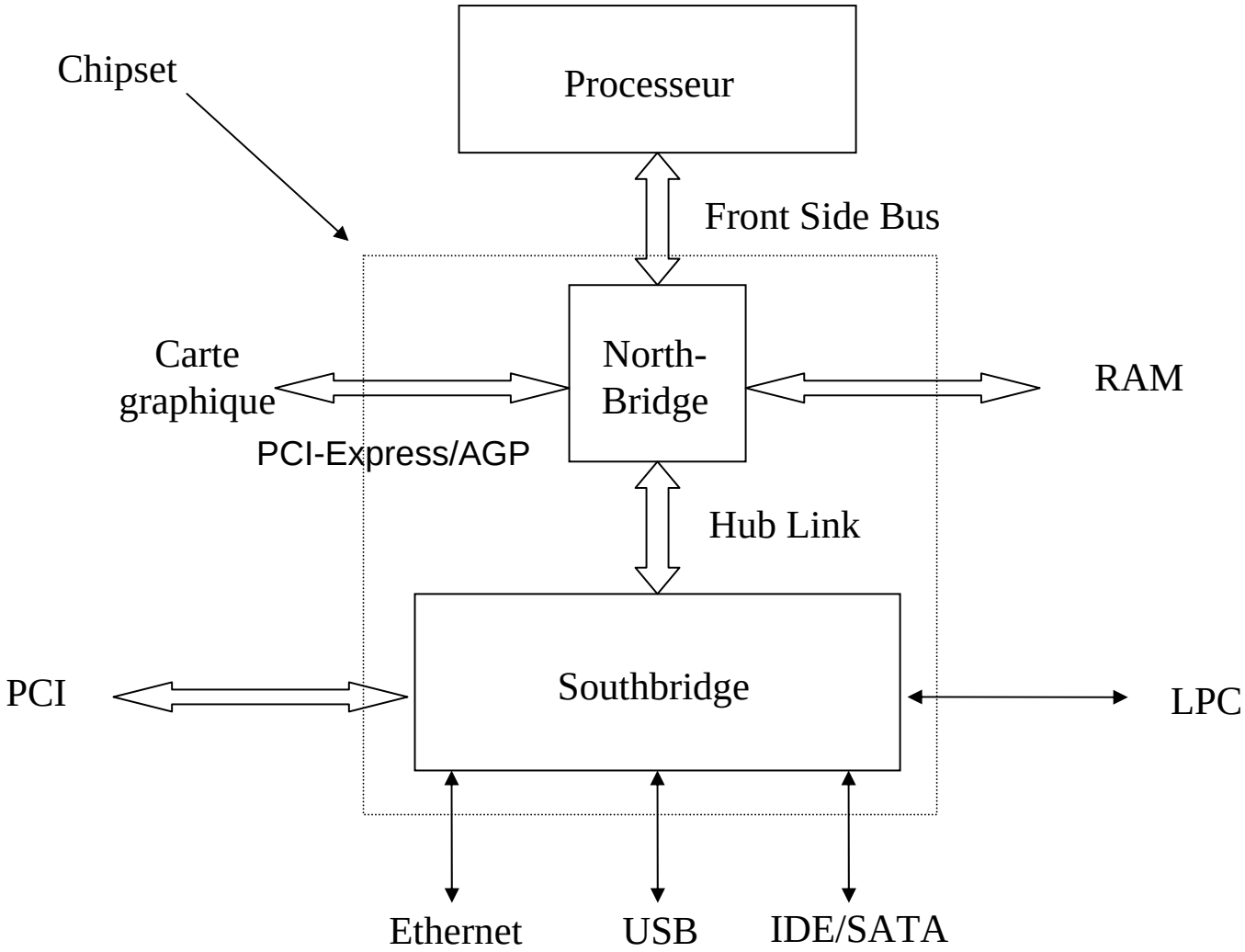
Au programme

- Introduction
- Gestion de l'alimentation et de la configuration
  - Architectures traditionnelles et informatique de confiance
  - Modèles d'attaquants considérés
- Problèmes liés au mode System Management
  - Présentation du mode SMM
  - Utilisation offensive
- Conséquences de l'utilisation de l'ACPI
  - Présentation de l'ACPI
  - Impact sur l'informatique de confiance
- Contremesures et conclusion

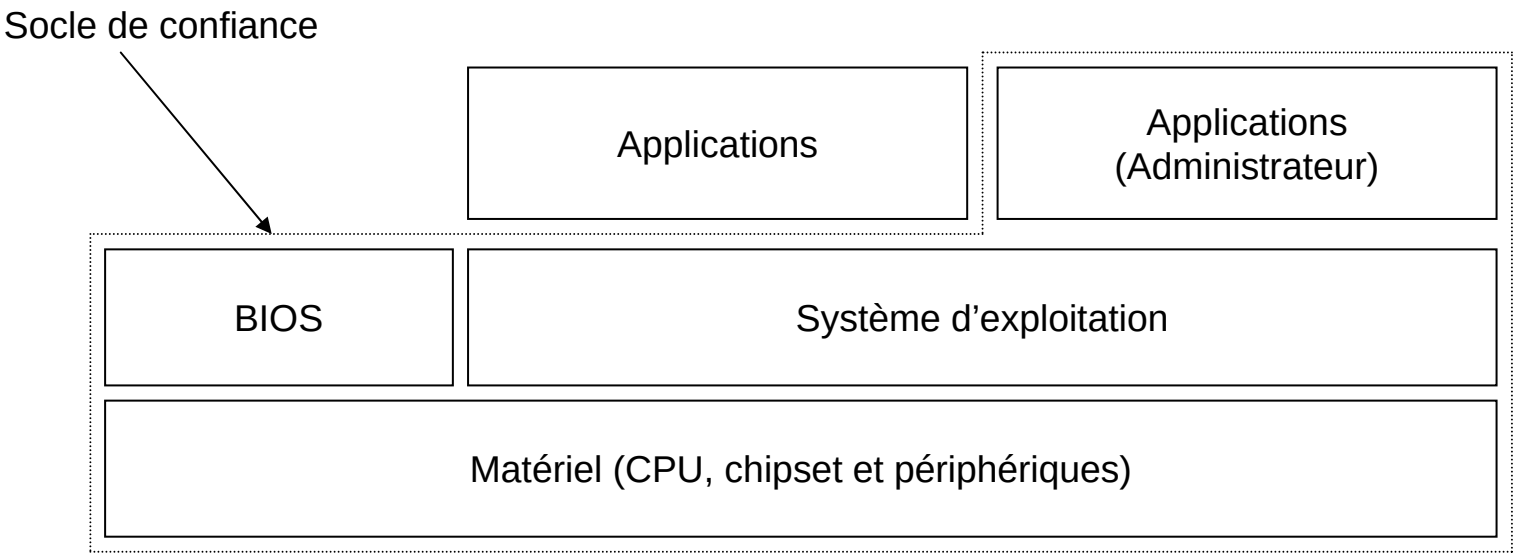
Au programme

- Introduction
- Gestion de l'alimentation et de la configuration
  - Architectures traditionnelles et informatique de confiance
  - Modèles d'attaquants considérés
- Problèmes liés au mode System Management
  - Présentation du mode SMM
  - Utilisation offensive
- Conséquences de l'utilisation de l'ACPI
  - Présentation de l'ACPI
  - Impact sur l'informatique de confiance
- Contremesures et conclusion

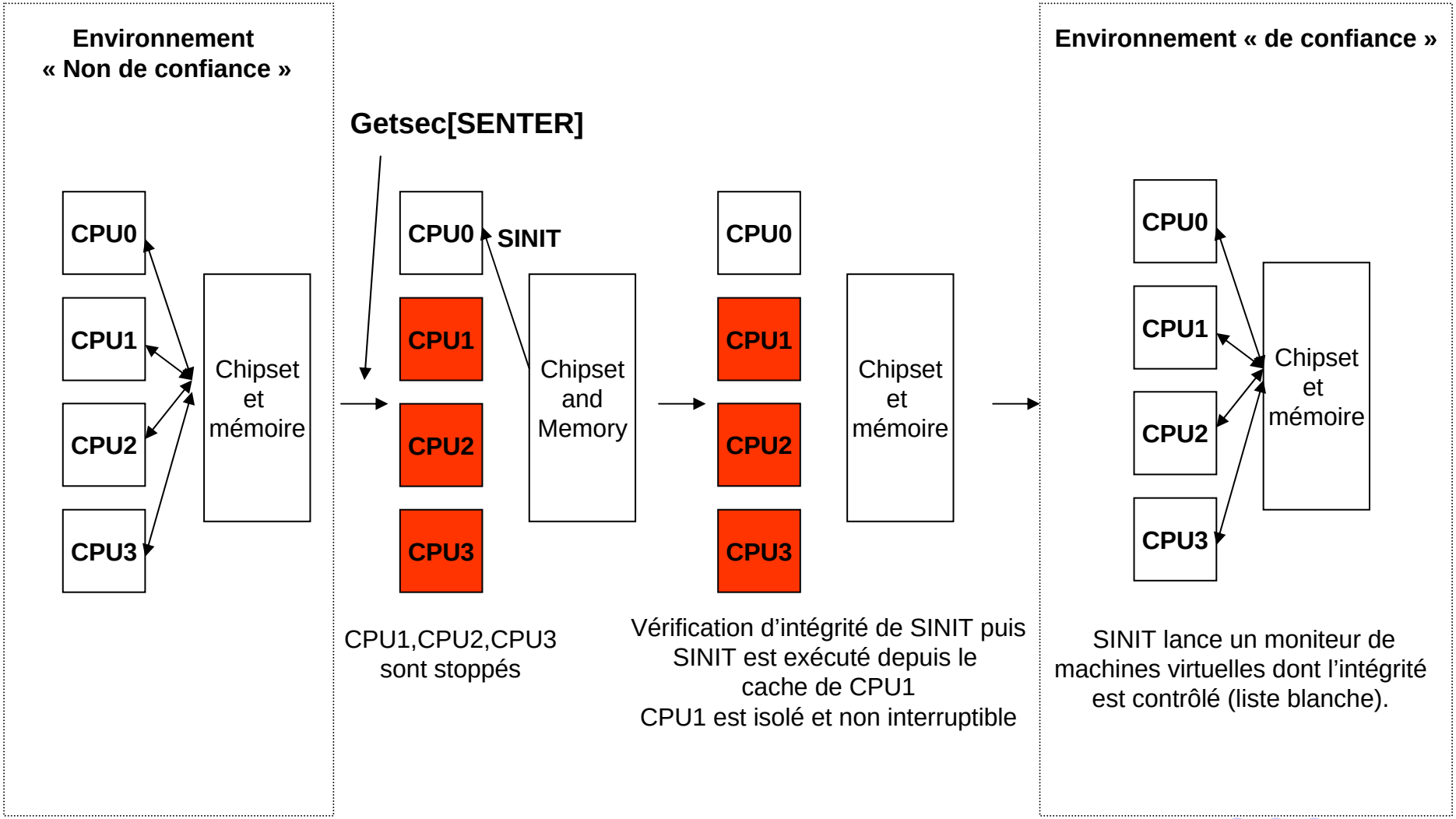
Architecture PC traditionnelle



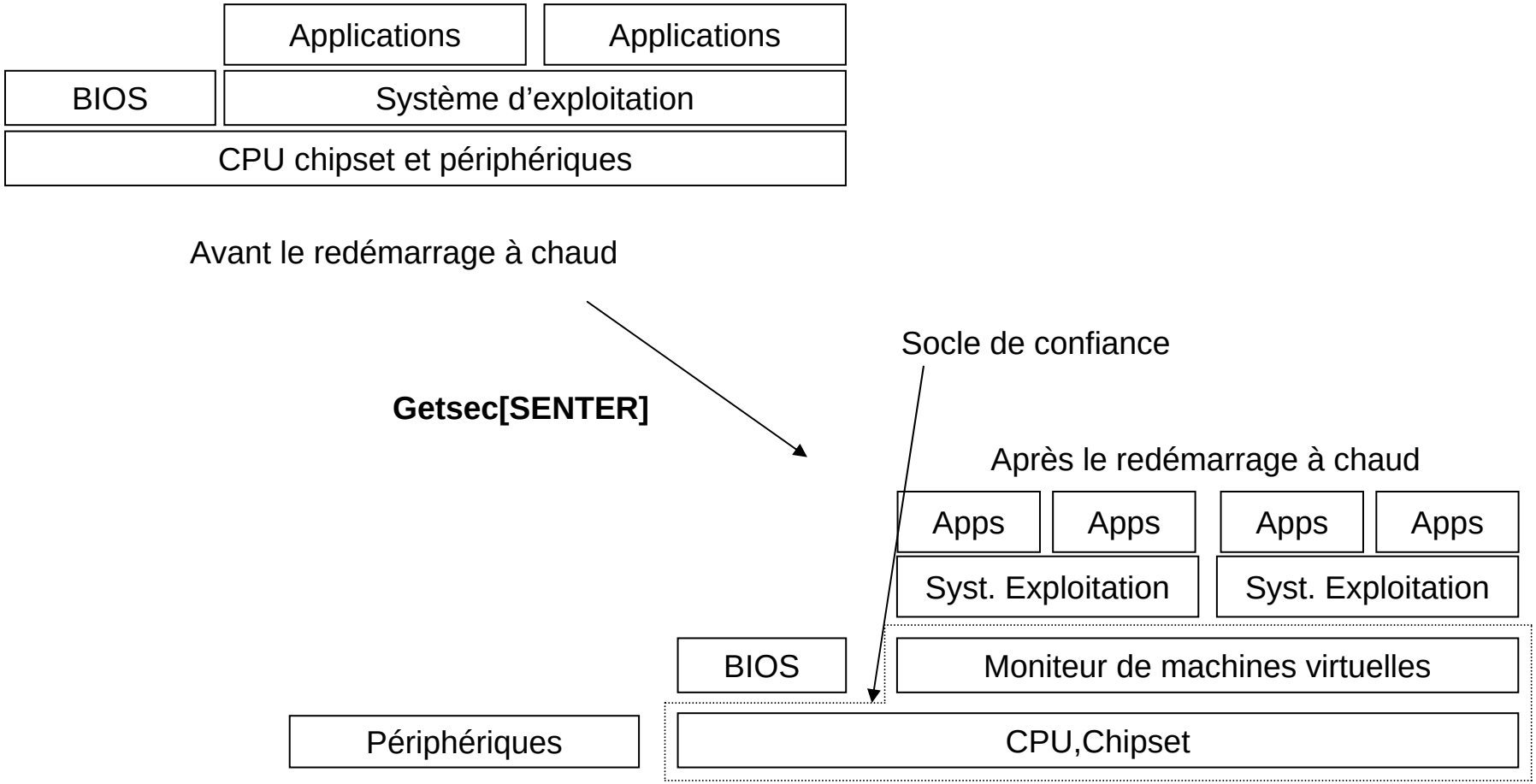
Architecture logicielle classique



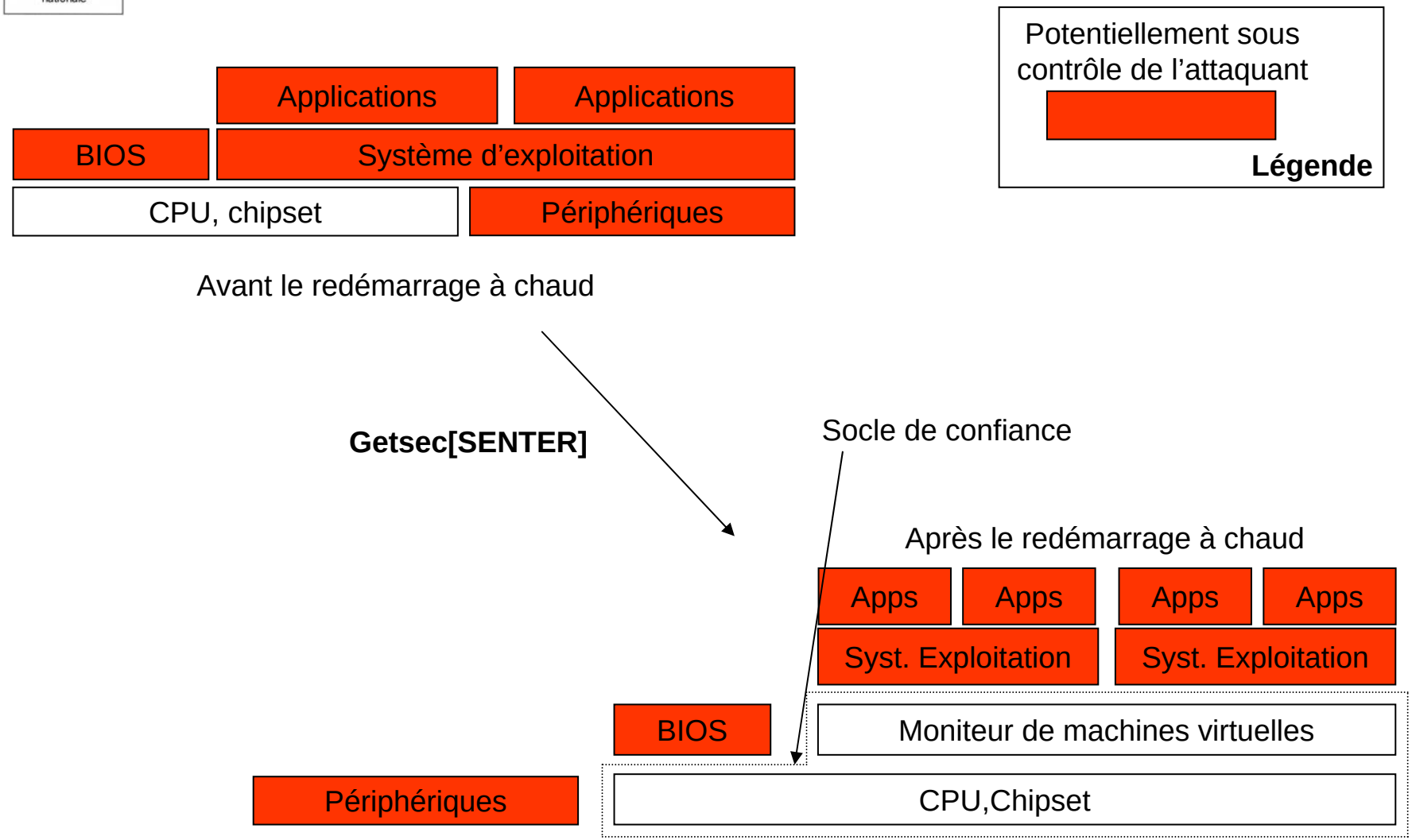
Principe du « late launch » (Intel® TxT)



Plateforme dite de confiance (rédémarrage à chaud « late launch »)



Modèle d'attaque



Gestion de l'alimentation et de la configuration

- Les périphériques d'une plateforme informatique ont sans cesse besoin d'être reconfigurés (ils passent par exemple d'un état actif à un état de veille lorsqu'ils ne sont pas sollicités).
- Les systèmes d'exploitation (et les moniteurs de machines virtuelles) sont des objets génériques, ils s'exécutent sur une multitude de plateformes différentes.
- Les caractéristiques matérielles de chaque plateforme sont différentes et un système d'exploitation ne peut connaître les caractéristiques de chacune d'entre elles.

- Comment s'opère donc la gestion d'alimentation et de configuration?

Gestion de l'alimentation

- Historiquement, le système d'exploitation n'était pas en charge des opérations de maintenance ou de gestion d'alimentation.
- Ces opérations étaient effectuées:
  - Par le BIOS (*Advanced Power Management*, APM).
  - Par le biais de la routine de traitement de la SMI (fournie par le BIOS).
- L'APM a été remplacé progressivement par l'ACPI (*Advanced Configuration and Power Interface*) qui est devenue norme de fait:
  - L'ACPI confie la gestion de l'alimentation à l'OS à partir de directives fournies par le BIOS (tables ACPI).

ACPI et routine de traitement de la SMI

- L'ACPI permet à l'OS d'effectuer la configuration et la gestion d'alimentation et de configuration d'une plateforme informatique, mais en se basant sur des informations fournies par le BIOS.
- La routine de traitement de la SMI (fournie par le BIOS) s'exécute en mode « *System Management* » alors que le système d'exploitation est « gelé », et qu'il est donc incapable de faire respecter une quelconque politique de sécurité.

- Comment dans ces conditions, exclure le BIOS du socle de confiance?
- Peut-on limiter les risques liés à l'emploi de ces deux technologies sur le plan de la sécurité?

Au programme

- Introduction
- Gestion de l'alimentation et de la configuration
  - Architectures traditionnelles et informatique de confiance
  - Modèles d'attaquants considérés
- Problèmes liés au mode System Management
  - Présentation du mode SMM
  - Utilisation offensive
- Conséquences de l'utilisation de l'ACPI
  - Présentation de l'ACPI
  - Impact sur l'informatique de confiance
- Contremesures et conclusion

System Management Mode (SMM)

- Un mode de maintenance du processeur :
  - Utile pour la gestion d'alimentation.
  - Interagit avec l'ACPI (voir plus tard).
- Un mode 16 bits particulier :
  - Accès complet à la mémoire physique.
  - Aucune protection mémoire n'est disponible dans ce mode.
  - Accès complet aux périphériques.
- Le code s'exécutant dans ce mode préempte le système d'exploitation ou le moniteur de machines virtuelles qui est « gelé » :
  - L'OS ne peut pas se rendre compte de l'exécution en mode SMM.
  - Il ne peut donc faire respecter une quelconque politique de sécurité au code s'exécutant en SMM.
  - **Exécuter du code arbitraire en mode SMM permet de prendre le contrôle d'une machine.**

Entrer en mode SMM

- System Management Interrupts (SMI) :
  - Les SMI sont des interruptions matérielles générées par le chipset (Northbridge).
  - **On ne peut entrer en mode SMM que sur réception d'une SMI.**
  - Il est possible de forcer la génération d'une SMI en écrivant dans certains registres du chipset (exemple registre APMC utilisé pour les interactions entre chipset et CPU).
    - `outl(something, 0xb2)`

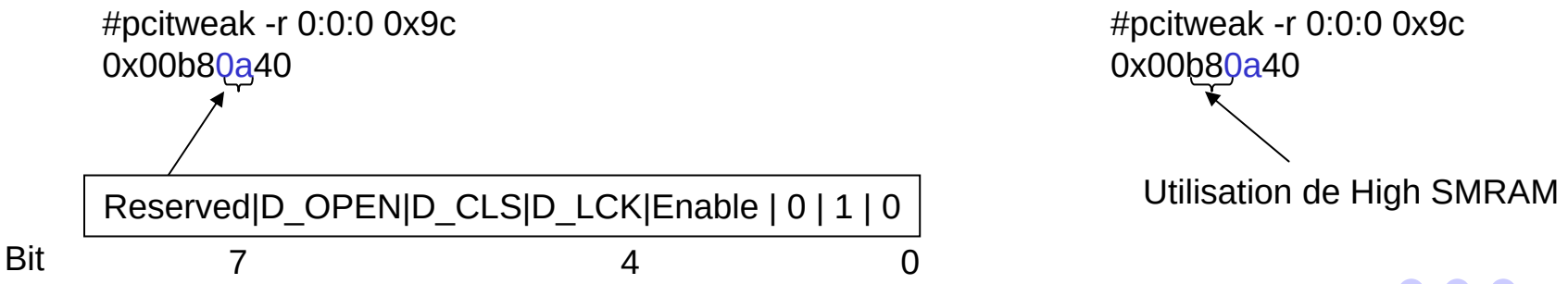
- Lorsqu'il reçoit une SMI, le CPU sauve son contexte et exécute la routine de traitement de la SMI qui se trouve dans une zone mémoire appelée SMRAM.
- Le contexte du CPU sera restaurer lorsque la routine de traitement de la SMI rendra la main au système d'exploitation.

SMRAM

- Localisation de la SMRAM
  - L'adresse de base de la SMRAM est contenue dans le registre processeur SMBASE.
  - Ce registre ne peut être ni lu ni modifié directement.
  - Il ne peut être modifié que par la routine de traitement de la SMI elle-même (utilisation de la sauvegarde des registres processeur).
- En pratique SMBASE prend généralement les valeurs suivantes :
  - 0xa0000: legacy SMRAM.
  - 0xfeda0000 (+/- 0x8000): high SMRAM.
  - Autre chose: TSEG (Extended SMRAM).
- L'adresse de la routine de traitement de la SMI est SMBASE + 0x8000 (offset fixe).

Protection de la SMRAM

- Il existe un mécanisme de contrôle d'accès à la SMRAM destiné à empêcher tout composant s'exécutant sur le CPU (hors routine de traitement de la SMI elle-même) de modifier le contenu de la SMRAM.
  - L'accès à la SMRAM n'est autorisé en général par le chipset que si le CPU est en mode SMM.
- Règle de contrôle d'accès pour permettre le chargement initial de la routine de traitement de la SMI:
  - Les zones legacy SMRAM, high SMRAM et TSEG ne sont accessibles que si le CPU est en mode *System Management* **sauf si le bit D_OPEN du chipset vaut 1**.



Protection de la SMRAM

- De plus, si le bit D_LCK du même registre vaut 1, alors le bit D_OPEN n'est plus accessible qu'en lecture jusqu'au prochain redémarrage de la plateforme.
- Le modèle est donc que le BIOS positionne le bit D_OPEN à 1, charge la routine de traitement de la SMI en mémoire, puis positionne le bit D_OPEN à 0 et le bit D_LCK à 1.
- Ce mécanisme de contrôle d'accès semble efficace sur le papier mais:
    - Seul le CPU sait où la SMRAM est effectivement localisée (le chipset ne voit pas SMBASE).
    - Le chipset ne peut donc protéger que les endroits où se situe potentiellement la SMRAM (legacy SMRAM, high SMRAM, TSEG).
    - De plus, il existe des problèmes de cohérence de cache. Une même zone mémoire peut contenir du code (ou des données) différent en mémoire principale et en mémoire cache.

SMM et informatique de confiance

- La routine de traitement de la SMI est chargée en mémoire (SMRAM) par le BIOS dès le démarrage de la machine.
- La routine persiste après le redémarrage à chaud (exécution de Getsec[SENTER].

- Si l'attaquant a réussi à piéger la routine de traitement de la SMI avant le redémarrage, il est capable d'y intégrer une porte dérobée qui soit utilisable après le rédémarrage à chaud.
    - Problème présenté par ailleurs par J. Rutkowska et R. Wojtzcuk postérieurement à notre soumission de ce papier (Blackhat DC 2009).

Problème 1: si le BIOS n'est pas de confiance

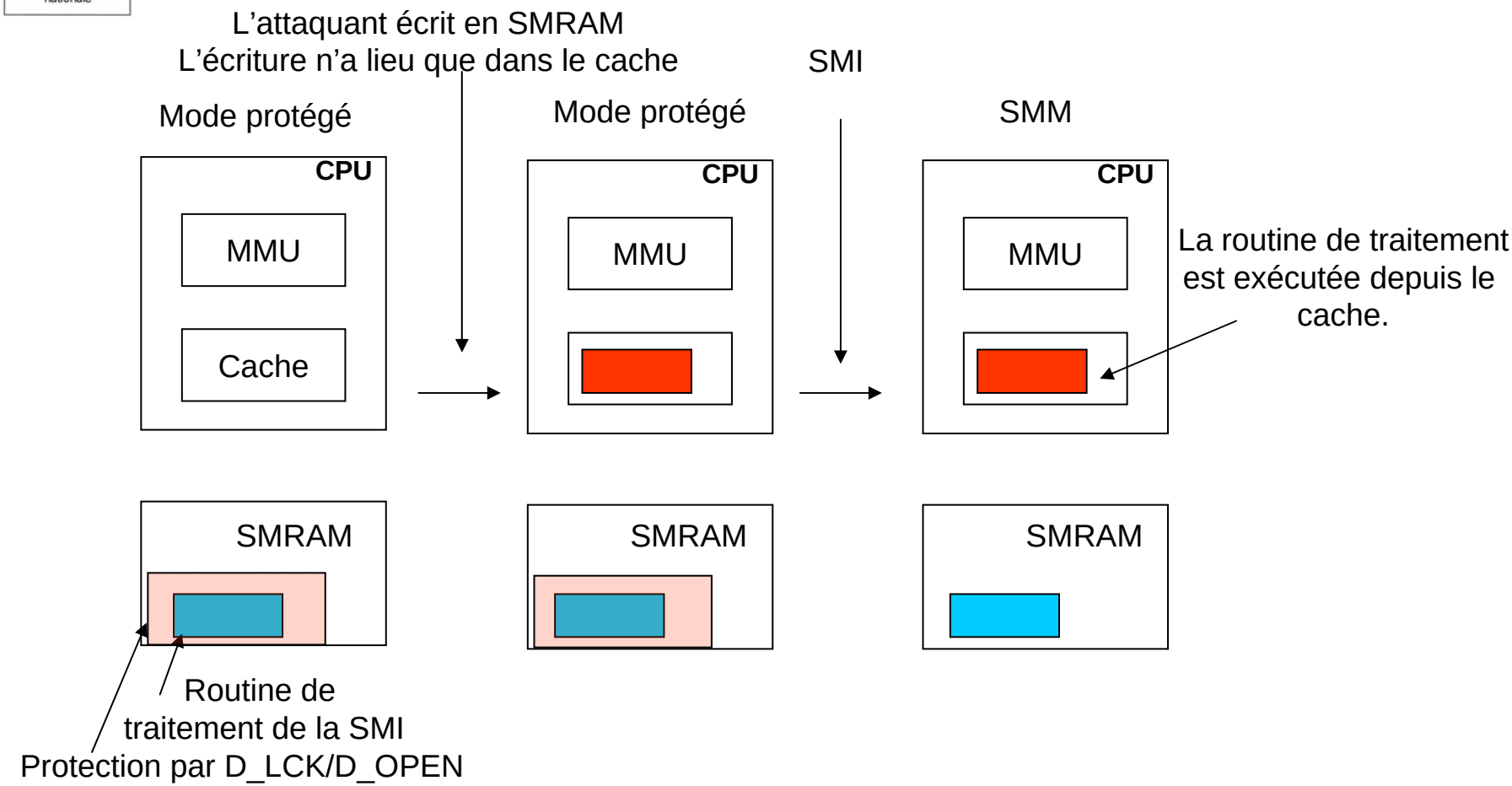
- Si le BIOS n'est pas de confiance, il peut bien entendu fournir une routine de traitement de la SMI qui soit piégée.
  - Il est donc faux de dire qu'une architecture de type Intel® TxT exclut le BIOS de la chaîne de confiance.

- Il est impossible à l'hyperviseur de contrôler la routine de traitement car elle est protégée par le mécanisme de contrôle d'accès du chipset, sauf à exploiter une vulnérabilité de ce mécanisme.

Problème 2: si le BIOS est de confiance

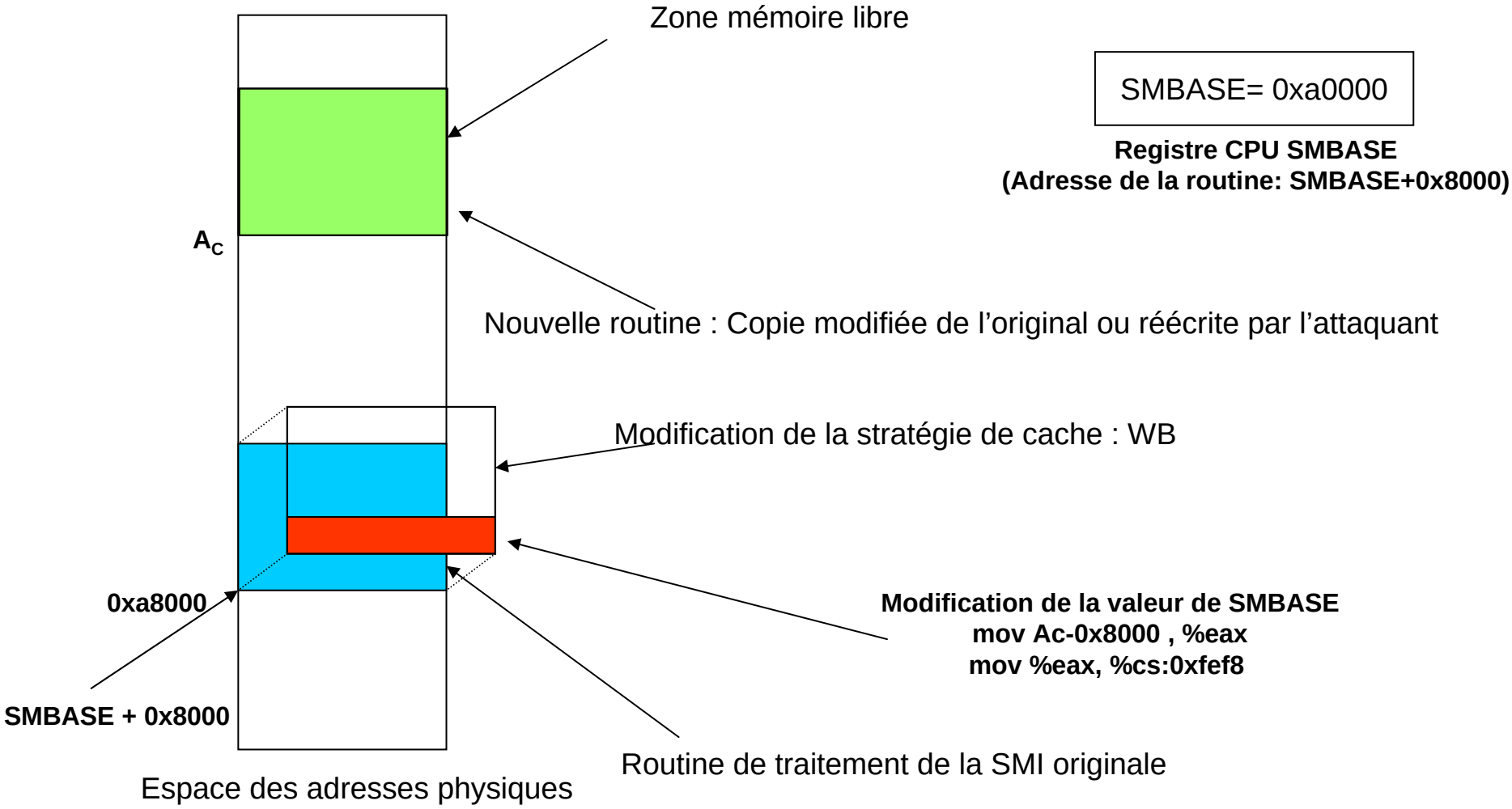
- Il est possible à l'attaquant s'il a pris le contrôle de la machine avant le redémarrage à chaud de tenter d'exploiter une vulnérabilité du mécanisme de contrôle d'accès du chipset.
- Existe-t-il de telles vulnérabilités?
    - Oui! Le modèle de sécurité est incohérent!
    - Le mécanisme de contrôle d'accès est situé dans le chipset.
    - Mais il est possible de spécifier que la SMRAM peut être mise en cache. Dans ce cas, il peut exister deux versions de la SMRAM, une dans le cache et une en mémoire.
    - Le mécanisme de contrôle d'accès ne concerne que la version en mémoire… L'attaquant peut modifier la SMRAM dans le cache et déclencher une SMI, le code sera alors exécuté depuis le cache.

L'attaque "vue du ciel"

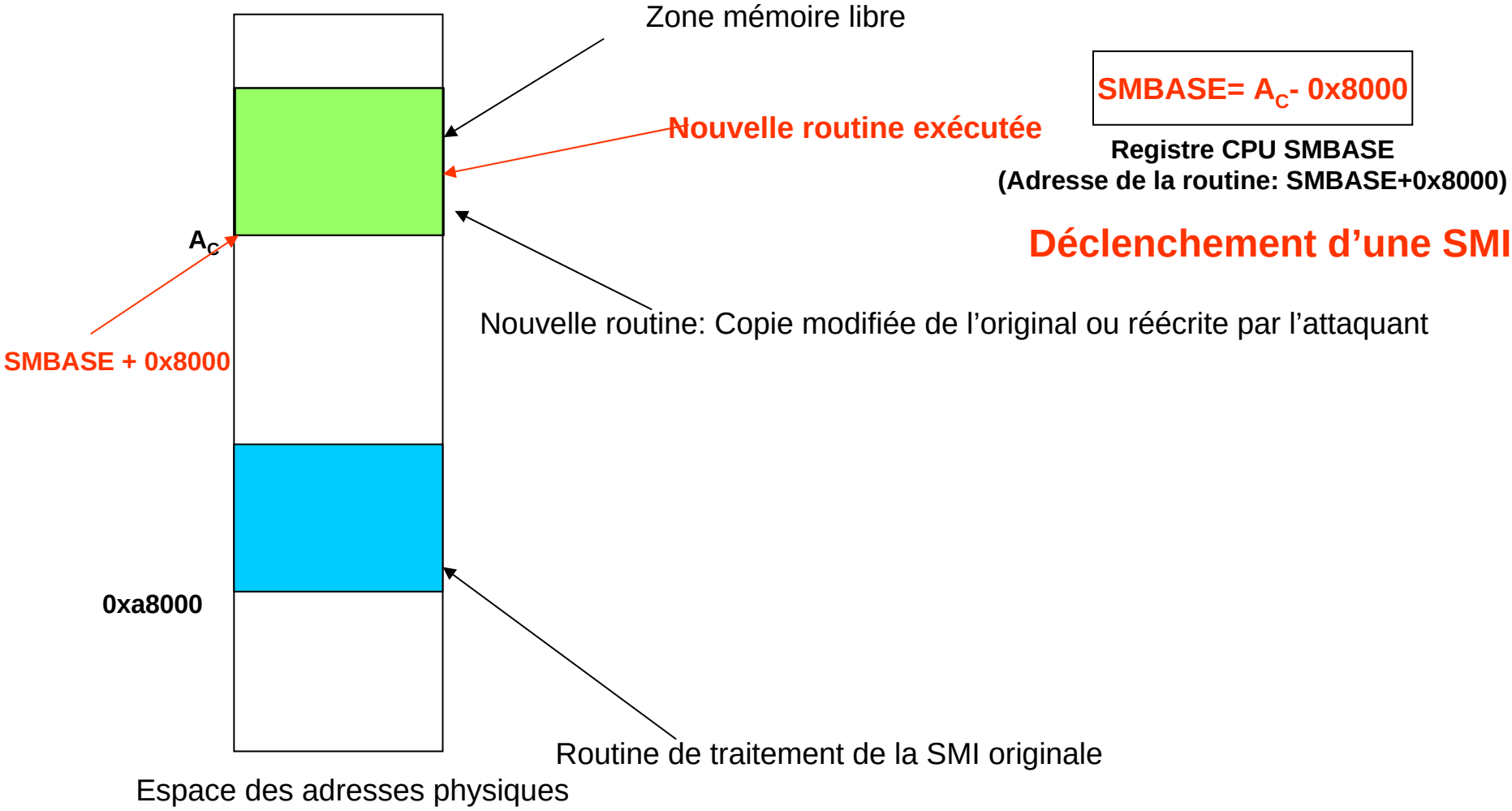


Le code de la routine est sous le contrôle de l'attaquant. Le mécanisme de contrôle d'accès est inutile.

Adaptation possible de l'attaque: Relocalisation de la SMRAM



Adaptation possible de l'attaque: Relocalisation de la SMRAM



Au programme

- Introduction
- Gestion de l'alimentation et de la configuration
  - Architectures traditionnelles et informatique de confiance
  - Modèles d'attaquants considérés
- Problèmes liés au mode System Management
  - Présentation du mode SMM
  - Utilisation offensive
- Conséquences de l'utilisation de l'ACPI
  - Présentation de l'ACPI
  - Impact sur l'informatique de confiance
- Contremesures et conclusion

ACPI en quelques mots

- L'ACPI est une norme de fait pour la gestion d'alimentation et de configuration des machines informatiques.
- Elle consiste à confier la gestion de l'alimentation au composant logiciel possédant les privilèges les plus élevés (OS ou moniteur de machines virtuelles).
- Le BIOS fournit à l'OS des tables décrivant comment configurer le matériel:
    - Ces tables contiennent une description de registres ACPI et de méthodes les mettant en œuvre.
    - Exemple: la norme ACPI spécifie que pour réveiller un contrôleur USB, l'OS doit exécuter la méthode USB0._S5 mais le contenu de cette méthode est spécifique à chaque machine.

Contenu de la DSDT

- Schématiquement, la DSDT (Differentiated System Description Table) est la table qui regroupe l'ensemble des méthodes permettant d'agir sur des périphériques.
    - Les périphériques sont organisés dans une structure arborescente.
        - _SB.PCI0.USB0
- Ces méthodes agissent sur des registres ACPI définis dans la DSDT.
    - Registres de périphériques où zones mémoire.
- L'OS interprète le contenu des tables mais n'a pas de moyen de décider de leur innocuité.
    - Il est possible de définir un registre ACPI correspondant par exemple à une partie du noyau!

Avant le redémarrage à chaud

- Sur une plateforme de confiance, sauf précaution particulière, les tables ACPI utilisées après le redémarrage à chaud seront celles qui ont été positionnées en mémoire avant le redémarrage à chaud.
- Si l'attaquant peut modifier ces tables, il peut insérer des pièges utilisables après le redémarrage à chaud.
- Un rootkit en mode noyau peut trivialement modifier les tables ACPI en mémoire.

- Ceci peut également être utilisé par un attaquant pour dissimuler des fonctions de type rootkit (hors contexte informatique de confiance).
    - Voir démonstration.

Principe de la démonstration

- Un rootkit en mode noyau a modifié la DSDT utilisée par le système d'exploitation.
  - Création d'un nouveau registre ACPI 8 bit (CTR) jouant le rôle de compteur:
    - Utilisation d'un registre inutilisé du chipset.
  - Modification de la routine BAT1._STA (vérification du statut de la batterie appelée très fréquemment, toutes les 10s en pratique) pour que cette routine remette systématiquement CTR à 0.
  - Modification de la routine ADP1._PSR (appelée systématiquement dès que le câble d'alimentation est branché ou débranché) pour qu'elle incrémente CTR et déclenche une fonction cachée quand CTR dépasse un certain seuil (4 dans notre exemple).
- La fonction cachée modifie l'appel système setuid() dès que le cordon d'alimentation est manipulé 4 fois sans que le statut de la batterie ne soit vérifié (i.e. 4 fois en 10s).

Au programme

- Introduction
- Gestion de l'alimentation et de la configuration
  - Architectures traditionnelles et informatique de confiance
  - Modèles d'attaquants considérés
- Problèmes liés au mode System Management
  - Présentation du mode SMM
  - Utilisation offensive
- Conséquences de l'utilisation de l'ACPI
  - Présentation de l'ACPI
  - Impact sur l'informatique de confiance
- Contremesures et conclusion

Impact

- Sur une machine de confiance:
  - Les problèmes présentés permettent à un attaquant de rendre inefficace le mécanisme de « late launch ».

- Sur une plateforme classique:
  - Ils permettent à un rootkit en mode noyau de dissimuler des fonctions au système d'exploitation.
  - L'impact est cependant moindre car ces fonctions ne survivent pas à un redémarrage de la machine (sauf modification du BIOS).

Contremesures: Routine de traitement de la SMI

- Les architectures actuelles ne permettent pas d'empêcher une routine de traitement de la SMI malveillante d'attaquer le système après le redémarrage à chaud.
- On peut donc uniquement se contenter de limiter les risques de compromission de la routine de traitement de la SMI avant le redémarrage à chaud.
    - Configurer correctement le chipset.
    - Utiliser les nouveaux mécanismes fournis pour empêcher les attaques par cache.

Contremesures: ACPI

- Analyse statique des tables ACPI :
  - Permet de détecter les comportements manifestement erratiques.
- Contrôle dynamique des tables ACPI:
  - Nouveau modèle: gestion de l'alimentation dans les couches moins privilégiées (démon applicatif), de manière à permettre un contrôle le plus fin possible des actions de ce composant au niveau du système d'exploitation.
  - Implémentation d'un démonstrateur à venir.

Conclusion

- Dans cette présentation nous avons montré que les choix effectués en matière de gestion de l'alimentation et de la configuration étaient des freins importants à la mise en place d'architectures de machines de confiance.
- Ces choix ont également un impact (moindre) sur les machines traditionnelles :
    - Routine de traitement de la SMI et tables ACPI peuvent être utilisés par des rootkits en mode noyau à des fins de camouflage de fonctions.
- Des contremesures existent qui nécessitent parfois une modification de l'architecture des machines.

Premier ministre

Secrétariat général de la défense nationale

Merci de votre attention

Des questions?

Contact address:

loic.duflot@sgdn.gouv.fr