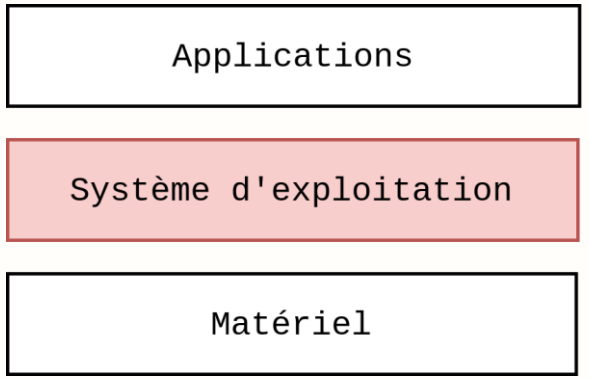
Conception et implémentation d'un langage dédié à l'introspection d'une machine virtuelle

Lionel HEMMERLE, Guillaume HIET, Frédéric TRONEL, Pierre WILKE, Jean-Christophe PREVOTET

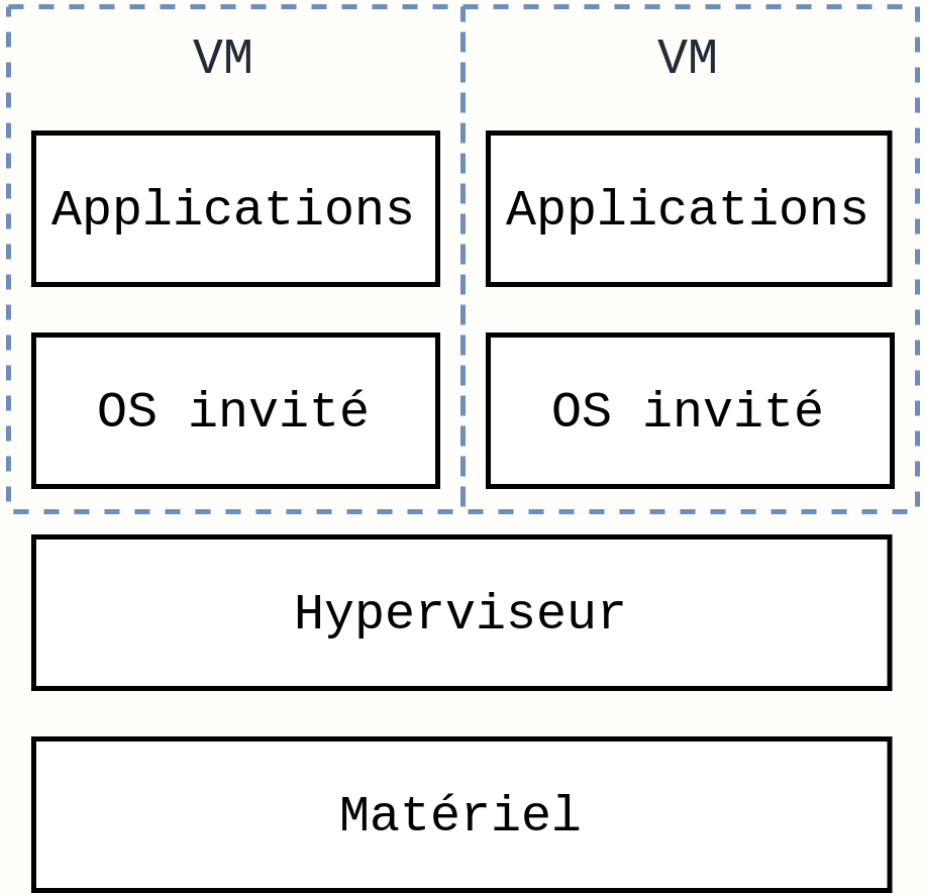
Contexte : Intrusions

- **Comment détecter des attaques quand l'OS est compromis**
  - L'attaquant peut communiquer des informations fausses aux autres programmes
  - Il peut désactiver les IDS
- **On ne peut pas dépendre des données fournies par l'OS**



Contexte : Virtualisation

- **Extension de virtualisation** :
  - Permet d'exécuter plusieurs Machines Virtuelles (VM)
  - Un logiciel nommé hyperviseur supervise ces VM
- **Avantages :**
  - Les VM sont isolées les unes des autres
  - Même si l'attaquant contrôle entièrement une VM, l'hyperviseur reste sécurisé
  - On peut placer l'IDS au niveau de l'hyperviseur pour le protéger



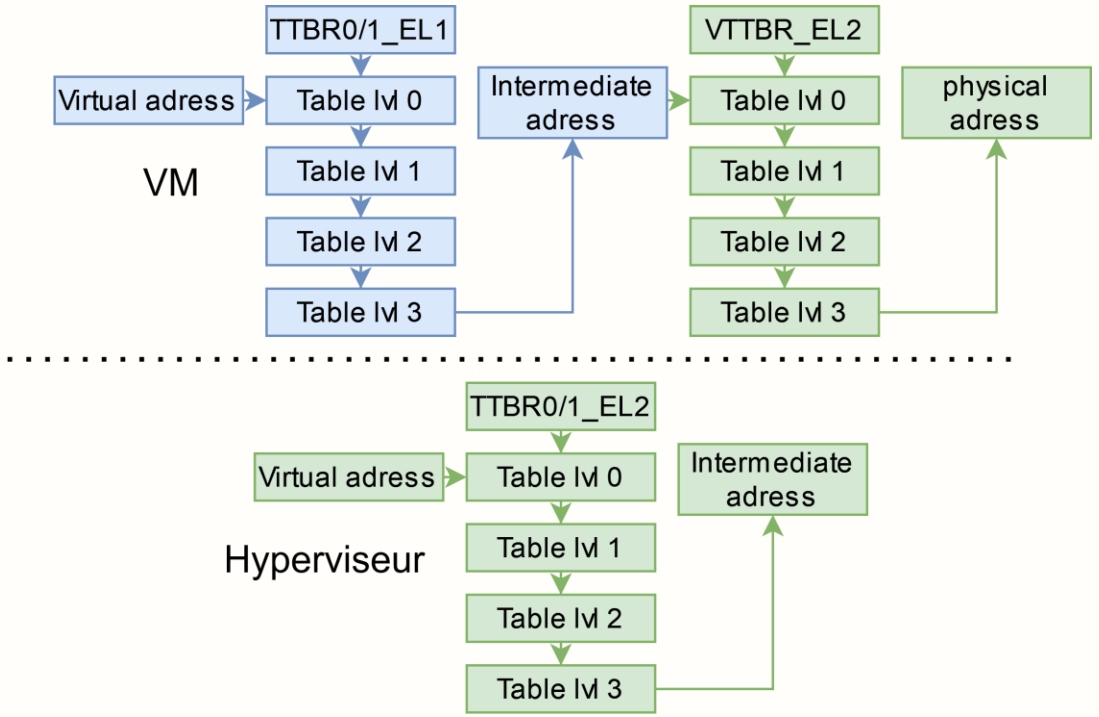
Contexte : Fossé sémantique

- **Fossé sémantique : Comment retrouver les informations dans la mémoire de la VM**
  - Comment elles sont stockées ?
  - Où elles sont stockées ?
  - Quand elles sont modifiées ?

- **Ex : liste des processus**

Contexte : Fossé sémantique

- **Exemple : Où sont stockées les données ?**
  - Mécanisme de traduction d'adresse partiellement effectué par la VM



Approche intéressante : Hyperupcalls

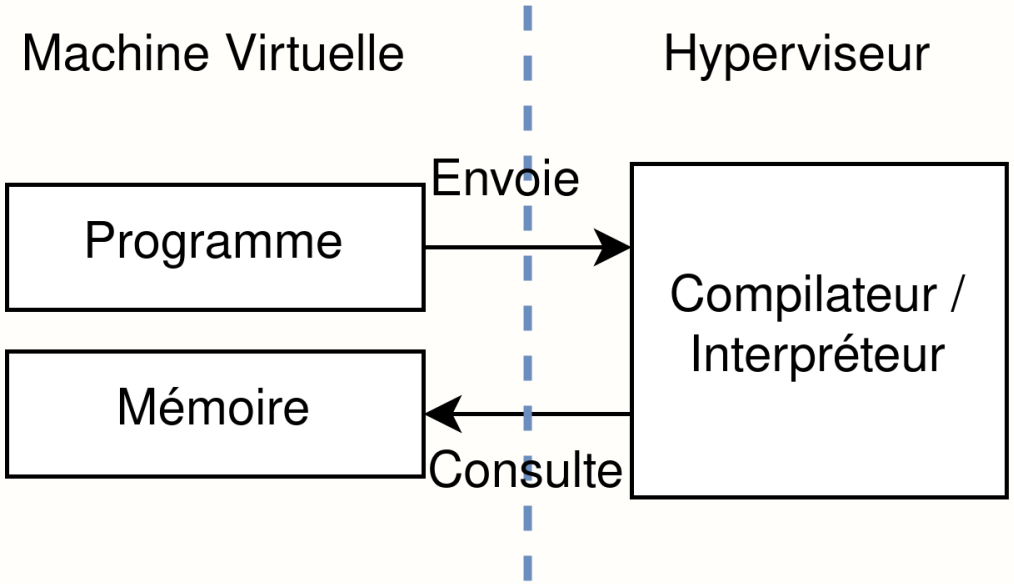
- **Les VMs envoient des programmes eBPF à l'hyperviseur**
  - L'hyperviseur exécute ces programmes
  - Ils permettent d'avoir des informations sur le fonctionnement de la VM
  - Utiliser pour que l'hyperviseur utilise mieux les ressources matérielles

- **Inconvénient**
  - Pas assez protégé pour faire de la sécurité
  - Les programmes peuvent être envoyés n'importe quand par la VM
  - Un attaquant peut modifier l'agencement des structures en mémoire pour désactiver les programmes

Michael Wei et Nadav Amit. « Leveraging Hyperupcalls To Bridge The Semantic Gap: An Application Perspective ». IEEE Data Eng. Bull. (2019).

Objectifs de la thèse

- **Créer un langage dédié à l'introspection**
  - Les VM fournissent des programmes
  - L'hyperviseur les compile et les exécute
  - Ils permettent alors de lever des alertes pour détecter des attaques
- **Implémentation**
  - Architecture Arm64 v8
  - Modification de XVisor



Contraintes de sécurité

- **L’envoi des programmes ne doit pas créer de nouvelle vulnérabilité**

- **Contremesure**
  - La VM est initialement considérée saine
  - La VM envoie un signal après avoir envoyé le dernier programme
  - Tout programme envoyé après est ignoré par l'hyperviseur
  - Des vérifications doivent être effectuées lors de la compilation ou de l’exécution
  - Surveiller les structures liées à la traduction d’adresse à chaque changement de contexte

Travaux effectués

- **Implémentation de deux rootkits**
  - Modifient les structures du noyau pour masquer un processus
  - Structures modifiées : table d'appel système et système de fichier virtuel

- **Modification de Xvisor**
  - Détection des écritures dans la mémoire de la VM
  - Envoi à l'hyperviseur de la liste des adresses mémoire à protéger
  - Détection des attaques implémentées

Conclusion