Sécurité des systèmes d'exploitation

Virtualisation

Plan

- Mécanismes de virtualisation
  - Types d'hyperviseur
  - Extensions matérielles
- Classes d'attaques spécifiques
- Hyperviseur, composant de sécurité ?
  - Windows 10 « Virtual Secure Mode »

Mécanismes de virtualisation

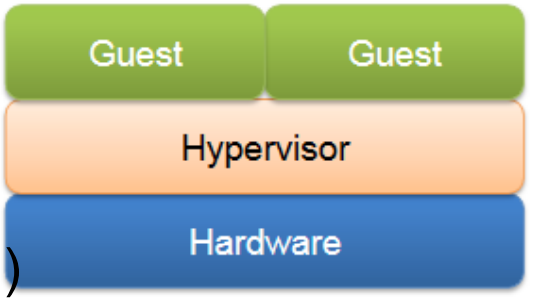
- Virtualisation : technique permettant de faire fonctionner une machine dite virtuelle sur un matériel abstrait
  - Matériel géré par un hyperviseur qui assure une médiation avec le matériel réel
  - Paravirtualisation : nécessite une modification du système d'exploitation de la VM pour accepter ce matériel
  - Virtualisation complète : l'abstraction doit être représentative du matériel réel et ne pas entraîner de modification du système d'exploitation de la VM
- Première implémentation à la fin des années 60 (IBM), suivie de travaux de spécification

Mécanismes de virtualisation

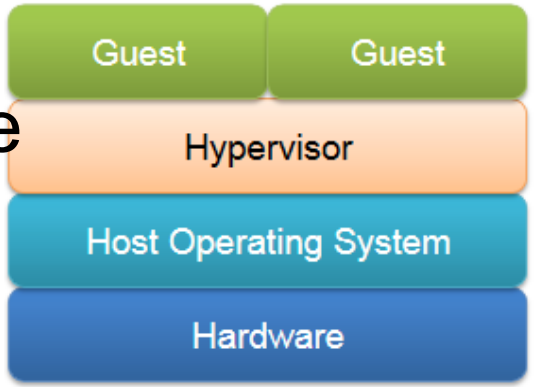
- Cas d'usage
  - Modifier et tester un système d'exploitation
  - Faire fonctionner des systèmes d'exploitations différents ou en version différente
  - Faire fonctionner un système d'exploitation avec une configuration différente du système réel (mémoire, nombre de processeur, …)
  - Modifier la configuration matérielle du système réel sans impacter la configuration d'un système d'exploitation
    - Par ex. obsolescence ou panne du matériel sous-jacent

Types d 'hyperviseur

- VMM (*Virtual Machine Monitor*)
  - Type 1 : Xen, Hyper-V, VMWare ESXi
    - Parfois appelé *bare metal hypervisor*
    - Indépendant du système d 'exploitation
      - Hébergement des pilotes dans le Dom0 (Xen) ou la partition parente (Hyper-V)
    - Taille réduite, code limité au partage et à la gestion des ressources matérielles entre les VM (dites invitées)
  - Type 2 : VirtualBox, KVM (*), VMWare Workstation
    - Installé sur un système d'exploitation hôte
      - La sécurité de l'hôte impacte la sécurité des VM



Type 1



Type 2

Source : Intel

Extensions matérielles

- Nouveaux jeux d'instruction
  - Un hyperviseur fonctionne en interceptant et émulant d'une manière sûre des opérations sensibles réalisées dans la VM
    - Par exemple, un changement de table de pages, qui pourrait donner une accès illégitime à une zone de la mémoire)
  - Une assistance matérielle doit permettre d'améliorer la performance et faciliter l'implémentation de l'hyperviseur
  - Intel VT-X / AMD-V
    - Jeux d'instruction incompatibles, implique de choisir une technologie ou de faire 2 implémentations en parallèle

Extensions matérielles

- Cas de Intel VT-X
  - Instructions VMXON et VMXOFF à utiliser par l'hyperviseur
  - Le basculement (ordonnancement ou gestion d'exception) entre les VM est marqué par des événements « VM Entry » et « VM Exit »
    - Chaque VM est caractérisée par une structure de données appelée VMCS (*VM Control Structure*)

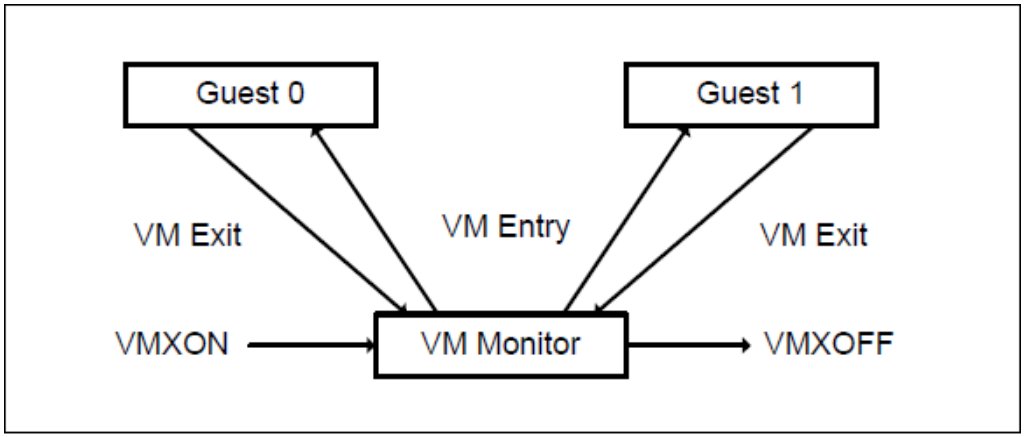


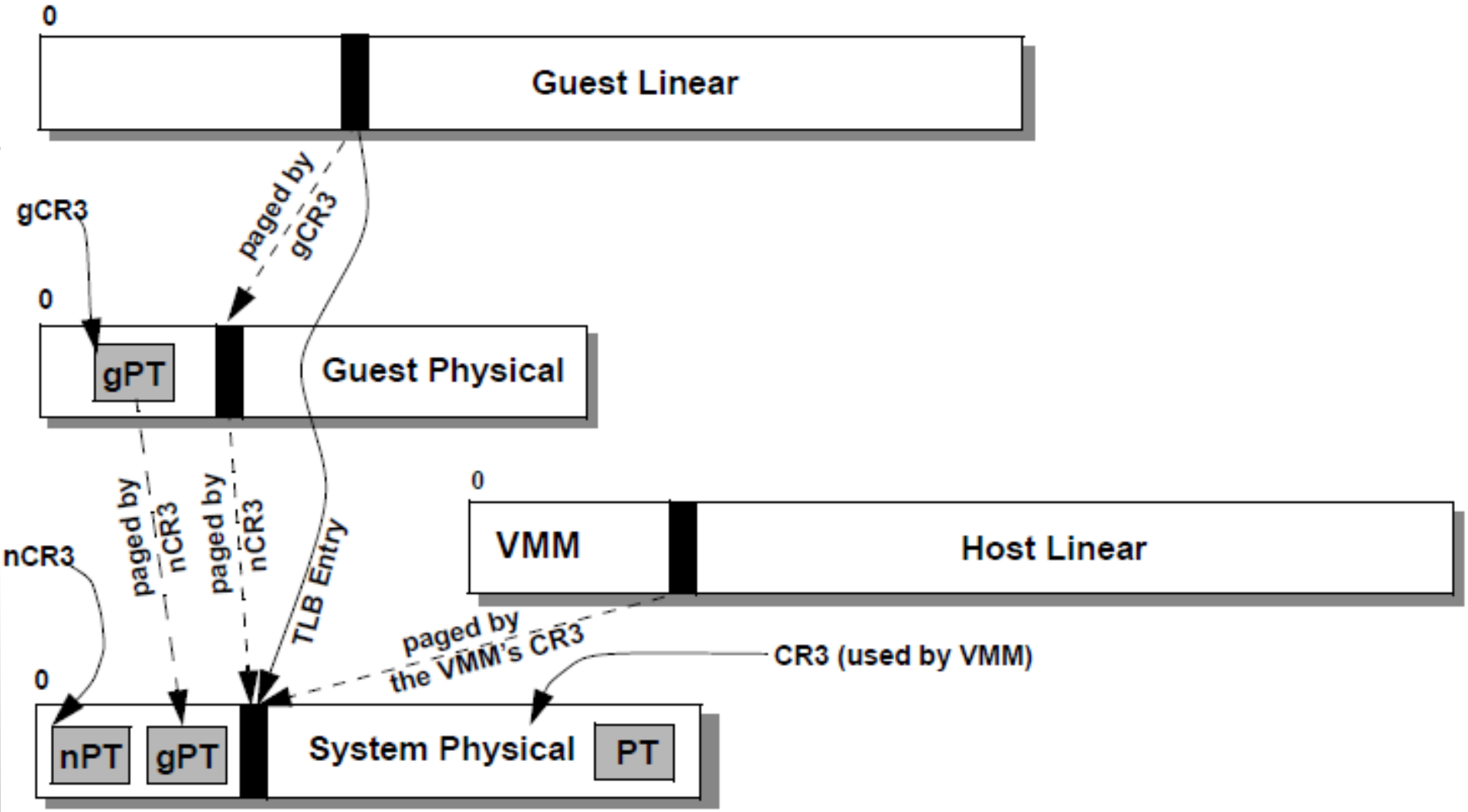
Figure 23-1. Interaction of a Virtual-Machine Monitor and Guests

Source : Intel

Extensions matérielles

- Gestion des tables de pages
  - L'hyperviseur a besoin de contrôler (et restreindre dans certains cas) l'accès des machines virtuelles à la mémoire physique
- Introduction d'une nouvelle « hiérarchie » de table de pages (Intel EPT/ AMD RVI)
  - Permet la translation « guest physical address » / « machine physical address »
  - Transparent pour l'OS de la VM qui continue à gérer sa mémoire virtuelle
  - L'hyperviseur peut positionner des permissions RWX sur des pages
  - Les fautes de page déclenchent un « VM Exit »

Extensions matérielles



Source : AMD

Figure 15-13. Address Translation with Nested Paging

Extensions matérielles

- I/O Memory Management Unit (Intel VT-D, AMD I/O VT)
  - Contrôle (permissions et translation d'adresses) sur les accès DMA des périphériques
  - Peut également être utilisés pour d'autres fonctions pour des accès depuis une VM (périphérique, interruptions, ...)

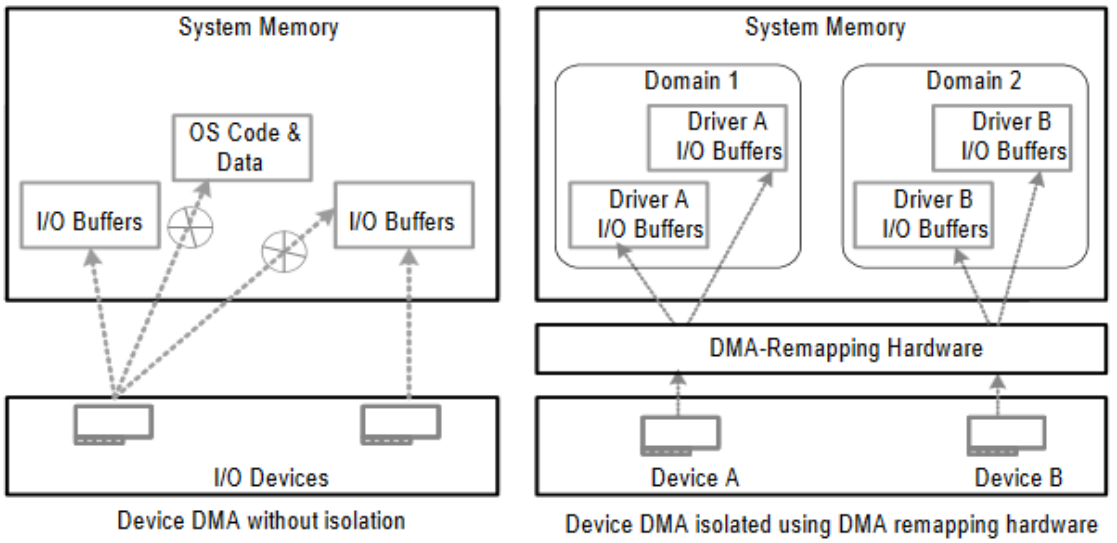
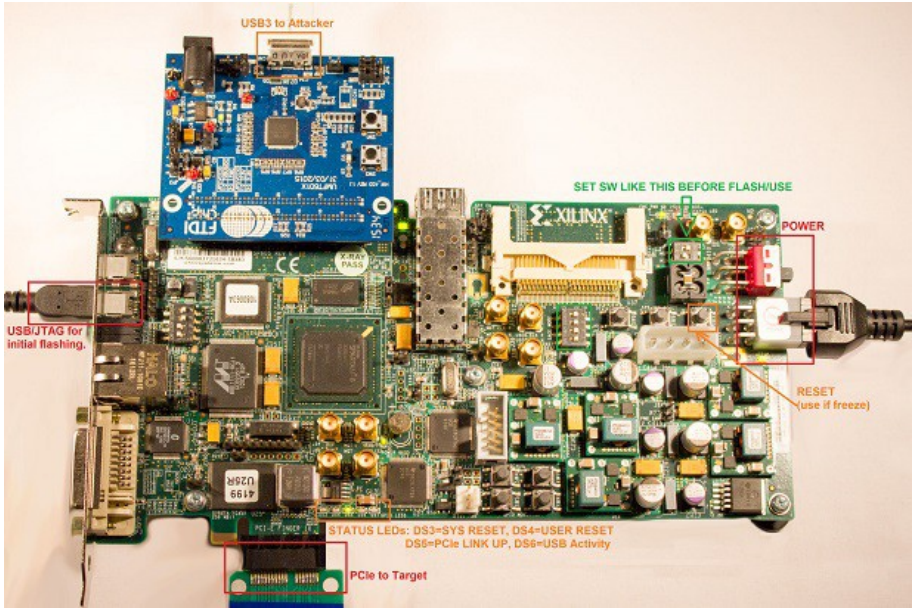


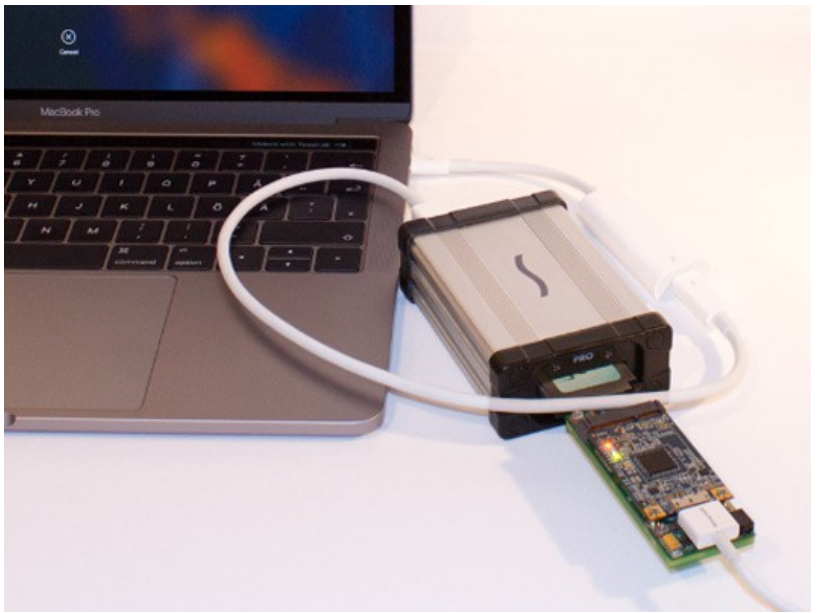
Figure 2-2. Example OS Usage of DMA Remapping

Source : Intel

Extensions matérielles

- Attaques DMA
  - Utilisation de périphériques PCIe pour accéder à la RAM sans le contrôle de la MMU
  - Exemple : PCI Leech
    - Utilisation de cartes FPGA PCIe ou de cartes USB3 mini-PCIe
    - Lecture/écriture de la RAM
    - Injection de code noyau
    - Récupération et injection de fichiers





Source : Ulf Frisk

Extensions matérielles

- SR-IOV (*Single Root I/O Virtualization*)
  - Extension PCI-Express
  - Permet à un périphérique, comme une carte réseau, de séparer les accès à des ressources de fonctions matérielles
    - Une *physical function* (PF) et une ou plus *virtual function* (VF)
  - Permet à du trafic réseau d'outrepasser la couche logicielle de l'hyperviseur et créer une lien direct entre la *virtual function* et la machine virtuelle
    - Augmente les performances est évitant de dédier une carte réseau physique à chaque VM

Classes d'attaques spécifiques

- Attaquer les autres VM
  - Un attaquant peut profiter d'un contrôle d'accès réduit ou d'une communication inter-VM légitime.
    - Cette attaque dépend de la configuration de l'hôte et du contrôle d'accès
- Attaquer l'hyperviseur
  - Généralement initié depuis une VM et dépendant de l'hyperviseur
    - Les pilotes de paravirtualisation, le partage de presse-papier, la sortie d'affichage et le trafic réseau sous des cibles potentielles

Classes d'attaques spécifiques

- Attaquer le matériel de l'hôte (depuis la VM)
  - Les plateformes matérielles reçoivent des mises à jour de micrologiciel
  - Si le mécanisme est accessible depuis la VM un attaquant peut charger un micrologiciel malveillant
    - L'hyperviseur devrait filtrer ce type de commande
- Attaquer l'architecture de l'hôte
  - Cas typique d'une attaque par canaux auxiliaires sur un composant partagé.
    - Par exemple, l'utilisation malveillante de l'allocation dynamique de mémoire (*memory-ballooning*)

Classes d'attaques spécifiques

- Attaque des VM par l'hôte
  - Peut mener à de la fuite d'informations, une altération du fonctionnement ou une interruption de service
    - Les faiblesses exploitées sont généralement des défauts de validation des entrées ou des certificats, l'élévation de privilège et des problèmes de manipulation des données
- Attaquer l'abstraction de la plateforme
  - Peut mener à des échappements de la VM ou des dénis de service (par l'arrêt de l'hyperviseur)
    - Par exemple, défaut d'implémentation du code qui simule un matériel ou du micrologiciel au niveau de l'hyperviseur

Classes d'attaques spécifiques

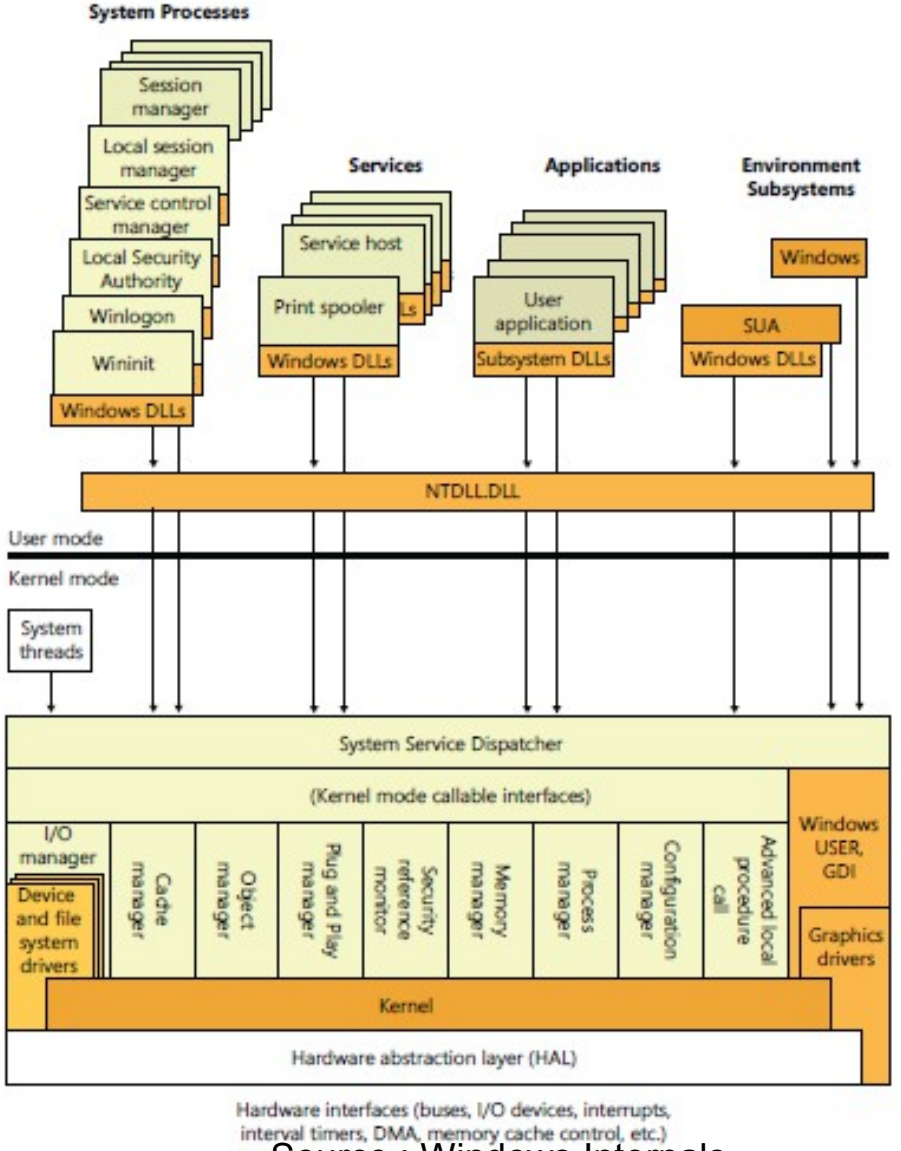
- Interception de données réseau
  - Pas spécifique mais peut-être amplifiée dans un environnement virtuel avec le partage de matériel
    - Par exemple, mutualisation d'une carte réseau ayant accès à plusieurs VLAN
- Attaque sur la couche de gestion du stockage
  - Peut mener à des interruptions ou des usurpations en s'appuyant sur des faiblesses classiques comme la validation d'entrées, l'injection ou du *cross-site scripting*

Hyperviseur, composant de sécurité ?

- L'hyperviseur met en œuvre des fonctions d'isolation de la mémoire et de contrôle des ressources
  - intérêt pour la sécurité
- Hyperviseur = logiciel
  - Taille et complexité du code ?
  - Conception prouvée ?
  - Développement sécurisé ?
  - ...
- Utilisation malveillante possible : rootkit en mode hyperviseur

Windows 10 « Virtual Secure Mode »

- Quelle TCB pour Windows ?
  - Idéalement le matériel, le noyau, le gestionnaire de mémoire, le gestionnaire d'objet et le moniteur de référence (Security reference monitor)
  - En pratique, tout ce qui est dans l'espace noyau (y compris les pilotes tiers) et le service LSASS



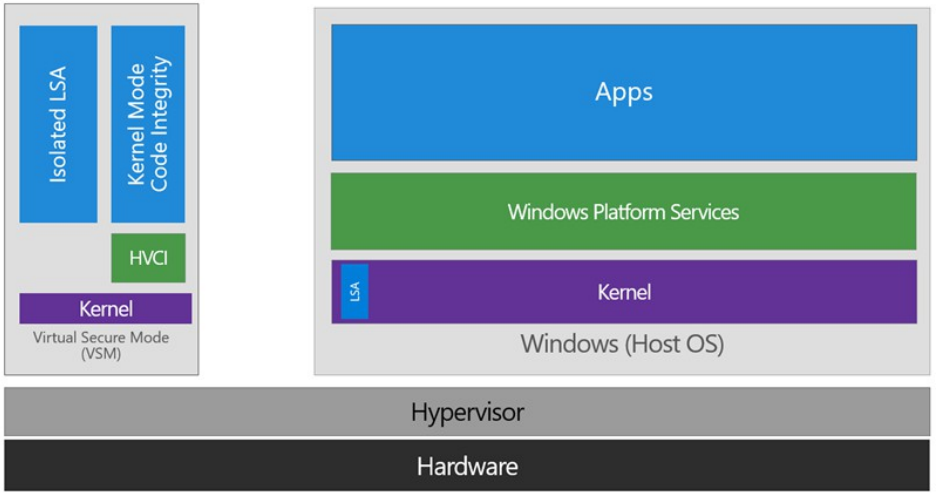
Source : Windows Internals

Windows 10 « Virtual Secure Mode »

- Disponible depuis la première version de Windows 10 (« 1507 »)
  - Uniquement dans la version « Enterprise »
- Terminologie
  - VBS : Virtualization Based Security
  - VTL : Virtual Trust Level
    - 0 : le plus bas niveau
    - 15 : le plus haut niveau
    - A ce jour, seuls VTL0 et VTL1 sont utilisés
  - *Truslet* : services hébergés par le VTL1
    - *Credential Guard*, *Device Guard*, vTPM

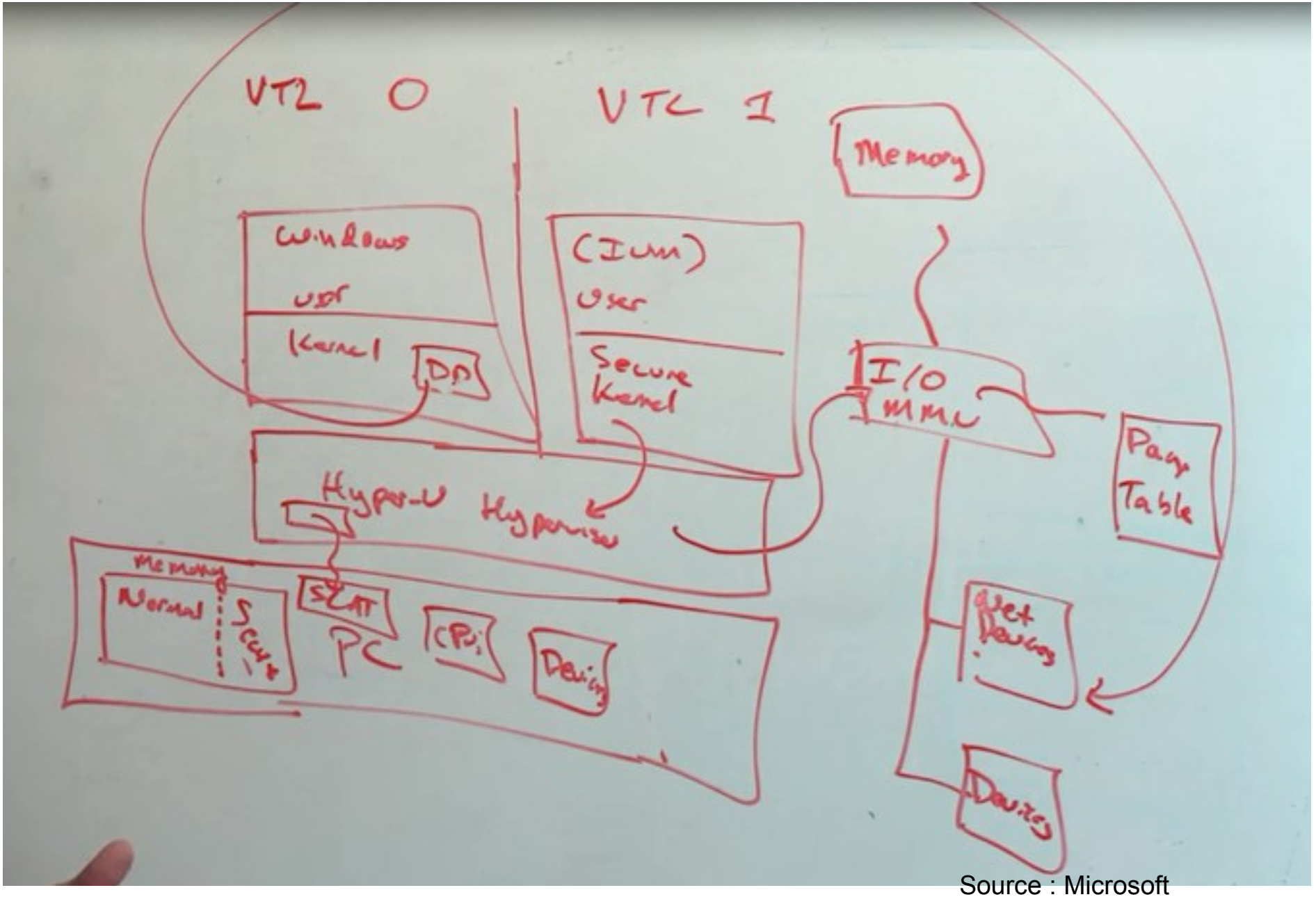
Windows 10 « Virtual Secure Mode »

- Séparation entre 2 environnements :
  - VTL0 qui contient le noyau NT et les applications
  - VTL1 qui contient le « SecureKernel » et des services protégés dans le « Isolated User Mode »
    - Ne contient pas de code tiers comme des pilotes



Source : Microsoft

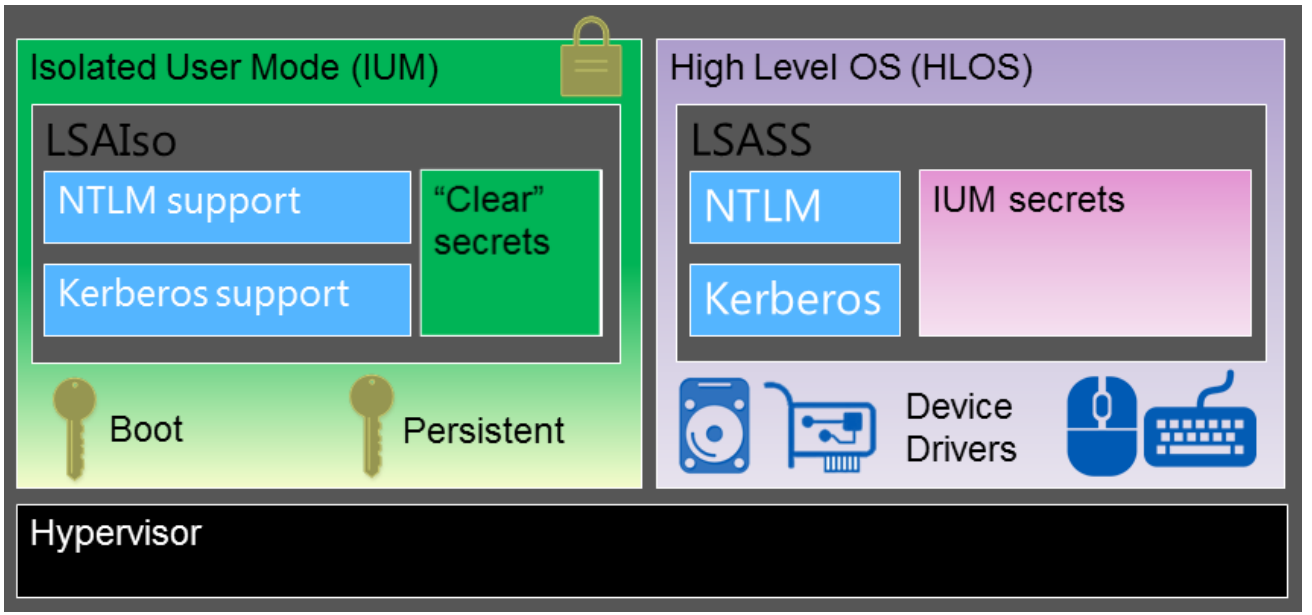
Windows 10 « Virtual Secure Mode »



Source : Microsoft

Windows 10 « Virtual Secure Mode »

- Credential Guard
  - Protection contre les outils du type « Mimkatz » qui récupère les éléments d'authentification (hash NTLM, ticket Kerberos) en mémoire



Windows 10 « Virtual Secure Mode »

- Device Guard fournit une solution de contrôle du code exécuté (approche liste blanche)
  - Réutilisation de composants présents depuis Windows 8
    - UEFI Secure Boot : vérification de la signature du chargeur de démarrage
    - KMCI et UMCI : vérification de la signature du code exécutable (noyau et espace utilisateur)
  - Extension
    - Introduction d'un module de vérification de la signature du code qui fonctionne au niveau hyperviseur : HVCI
    - KMCI est déplacé au niveau VTL1 avec HVCI
    - La liste blanche de Code Integrity doit être configurée
      - via des cmdlets PowerShell

Windows 10 « Virtual Secure Mode »

- Code Integrity indique (depuis VTL1) à l'hyperviseur quelles sont les pages contenant du code signé
  - Utilisation des permissions de pages sur les tables EPT
- Si l'OS tente d'exécuter une page définie comme non-exécutable au niveau EPT, déclenchement d'un VMExit
  - Même si la page est vue comme exécutable au niveau de l'OS

Pour aller plus loin (1/2)

- Intel® 64 and IA-32 Architectures Software Developer's Manual (Vol. 3, ch. 23 à 33)
- Security aspects of virtualization, ENISA, 2017
- Problématiques de sécurité associées à la virtualisation des systèmes d'information, ANSSI, 2013
- Windows 10 Virtual Secure Mode with David Hepkin, Channel 9
- Hypervisor Top Level Functional Specification v5.0a, Microsoft

Pour aller plus loin (2/2)

- Battle Of SKM And IUM, A. Ionescu, Black Hat USA 2015
- Analysis Of The Attack Surface Of Windows 10 Virtualization-Based Security, R. Wotjtzuck, Black Hat USA 2016
- Introduction to Windows Device Guard, Matt Graeber, 2016

Questions ?