Qubes OS

La sécurité par l'isolement

Wéry Benoît

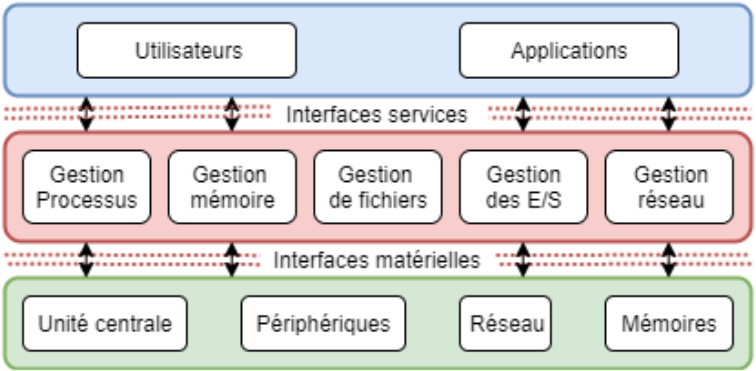
ECAM - 5MIN

Cours de Virtualisation, 8 Janvier 2020

Inroduction

1. Sécurité des OS
   - Rappel
   - Les points critiques
2. Qubes OS
   - Informations générales
   - Philosophie
3. Virtualisation dans Qubes
   - Hyperviseur Xen
   - Types de VMs
   - Mécanisme de templates
   - Exemple réception d'email
   - Autres aspects notables
4. Les faiblesses de Qubes
5. Conclusion

Le système d'exploitation est un logiciel **complexe** qui permet aux applications d'utiliser les ressources matérielles de la machine. Il a un accès privilégié au matériel et est responsable de nombreux **services** à disposition des programmes.



- Problèmes :
  - Nombreuses interactions des composants et **manque d'isolement**
    -> **propagation** d'un malware, système entier potentiellement compromis

    Cause : **Kernel monolithique**
    - services exécutés en mode noyau, accessibles depuis les applications
    - code complexe, difficile à maintenir et impossible à sécuriser de toutes parts
  - Différentes formes d'attaques possibles et codes malveillants complexes
  - Difficultés de détection d'une infection

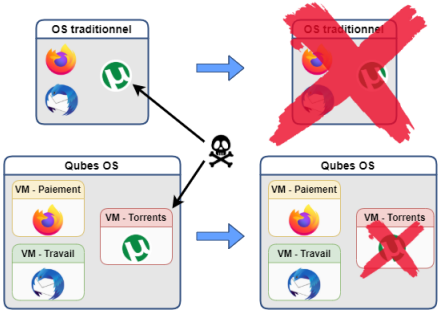
- Contre mesure :
  - Principe de **sécurité par la correction** : la plupart des OS ont une **approche réactive** (mises à jours, corrections d'erreurs, patchs,...) face aux failles de sécurité.

- Un OS gratuit, Open-Source, orienté sécurité
- Fondé par Joanna Rutkowska et Rafal Wojtczuk, maintenu par l'équipe *Invisible things Lab*
- Lancement du projet fin 2009
- Première version officielle stable 1.0, 2012
- Version actuelle 4.0.x, 29 mars 2018

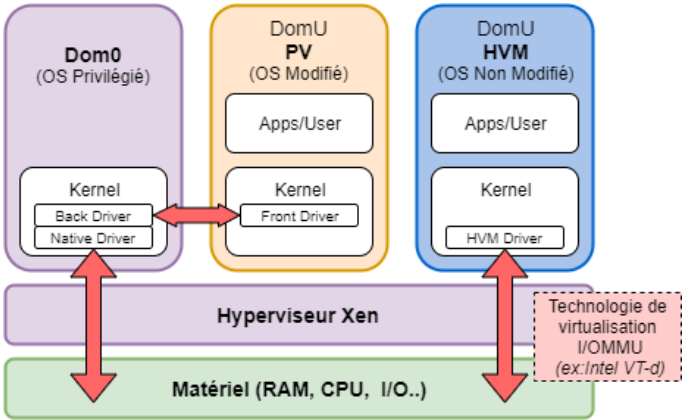
Ce que Qubes OS n'est pas...

- un OS codé de toute part. Il s'agit plutôt d'un système qui utilise des technologies existantes et les combine à des fins de sécurité (réduction des surfaces d'attaques)
- un hyperviseur
- une distribution GNU/Linux

- **Sécurité par l'isolement** : partir de l'hypothèse qu'il est impossible d'avoir un système sécurisé à 100% et qu'une attaque aura lieu tôt ou tard. Le but est alors de limiter son impact en divisant le système en plus petits espaces de travail cloisonnés et moins exposés aux risques, de façon à empêcher la propagation.
- Compartimentation de la vie digitale en **domaines/qubes**

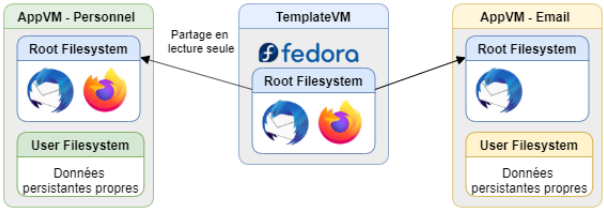


- **Hyperviseur de Type 1**
  Bare-metal/Native
- Accès privilégié aux **ressources matérielles**
- Gestionnaire de VMs/domaines
- Responsable de l'isolement entre les VMs
- Supporte :
  1. **Para-virtualisation (PV)**
  2. **Virtualisation complète assistée par le matériel (HVM)**

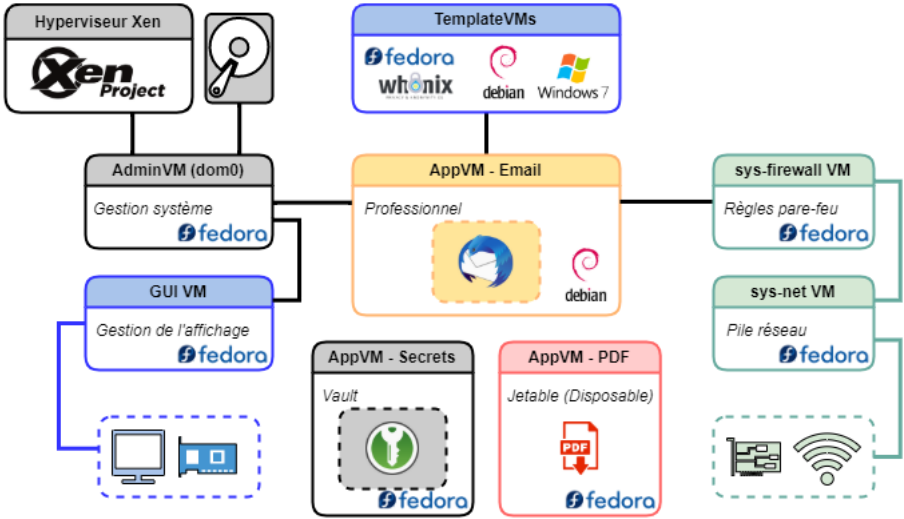


Type	Usages
dom0	. OS privilégié de Xen : accès aux ressources matérielles + gestion VMs . critique : pas de code qui l'exposerait au monde extérieur
TemplateVM	. fournit son système de fichiers (racine) aux autres en lecture seule . installation et mise à jour des applications
AppVM (qube)	. compartimentation en domaines de sécurité . exécution des applications
DisposableVM (jetable)	. unique (*ouverture de documents, exécution de code suspicieux, ...*)
ServiceVM (SystemVM)	. fournit un service spécifique aux autres VMs (*Réseau, Stockage, ...*) . accès non privilégié aux ressources via IOMMU/Intel VT-d
StandaloneVM	. propre système de fichier
GUI VM	. gestion de l'affichage unifié

Execution de machines virtuelles "applicatives" dont le système de fichiers racine dépend d'un **template partagé en lecture seule**.



- **Sécurité** : accès en lecture seule au système de fichiers du templateVM. Une attaque dans un qube n'impact ni le template ni les autres qubes.
- **Vitesse** : un système de fichiers déjà existant pour créer de nouveaux qubes.
- **Mises à jour** : centralisées au niveau de chaque templateVM, affectent tous les qubes associés.
- **Stockage** : partage du système de fichiers d'un même template. Le qube n'a besoin que de l'espace nécessaire au stockage de ses propres données.
  -> persistance des données : */home, /usr/local, /rw/config (user filesystem)*


Hyperviseur Xen
Xen Project
TemplateVMs
fedora
whonix
debian
Windows 7
AdminVM (dom0)
Gestion système
fedora
AppVM - Email
Professionnel
debian
sys-firewall VM
Règles pare-feu
fedora
GUI VM
Gestion de l'affichage
fedora
sys-net VM
Pile réseau
fedora
AppVM - Secrets
Vault
fedora
AppVM - PDF
Jetable (Disposable)
PDF
fedora

1. **Séparation physique VS compartimentation logicielle** : communication inter-VMs et partage de données via des drivers spécifiques plus légers (basés sur TCP/IP) et des technologies de partage de mémoire propres à Xen. Moins complexe que les drivers de réseaux complets (WiFi, DHCP, network stack...) uniquement disponibles dans les netVMs ce qui a pour conséquence de diminuer la surface d'attaque.
2. **Domaine de stockage** : chaque AppVMs possède un bloc mémoire dédié qui est géré par le domaine de stockage et qui possède un chiffrement propre à l'AppVM. Ainsi en cas de corruption du domaine, il n'est pas possible d'accéder aux données des AppVMs.
3. **Affichage** : La GUI VM permet un affichage unifié des différentes VMs pour donner l'illusion que les applications s'exécutent nativement sur le bureau.

- *"A reasonably secure operating system."*
- Les attaques possibles contre ...
  1. Les technologies de virtualisation niveau matériel (IOMMU/VT-d)
  2. L'hyperviseur
  3. Les logiciels/services utilisés par le système de virtualisation
- L'utilisateur reste responsable des applications qu'il installe dans les templateVMs.
- L'OS n'empêche pas l'exécution d'un code malveillant MAIS apporte une solution à la propagation des attaques et diminue les surfaces d'attaques.
- La sécurité par l'isolement ne s'applique pas au sein d'une même VM

Conclusion

- Utilisation judicieuse (car pas une "solution miracle") de la virtualisation pour des besoins de **sécurité** d'un OS.
- **Compartimentation** de la vie digitale de l'utilisateur en domaines.
- Système exécute plusieurs **machines virtuelles isolées** (grâces à des technologies tel que : *Hyperviseur Xen, IOMMU/VT-d, ...)*.
- **Diminution des interfaces** de communication entre les composants donc diminution des surfaces d'attaques.
- Principe de **sécurité par l'isolement** : contenir la propagation d'une attaque pour limiter son impact sur le reste du système.