Détection de défaillance de gestionnaires de machines virtuelles

Benoît Morgan (bmorgan@laas.fr)*
Éric Alata (ealata@laas.fr) *

Résumé : Ce papier décrit une nouvelle méthode pour la détection de compromission de gestionnaires machines virtuelles dans un contexte de cloud computing. L'approche proposée s'appuye sur la virtualisation récursive.

Mots Clés : cloud computing, gestionnaires de machines virtuelles, compromission, observation, détection.

1 Cadre des travaux

L'utilisation de l'informatique en nuage (*Cloud Computing*) est aujourd'hui en plein essor mais pose également beaucoup de problèmes de sécurité. Effectivement, les clients d'un *cloud* exécutent leurs applications sur des machines virtuelles déployées sur les mêmes machines physiques. Or, ces clients ne se font pas mutuellement confiance. Les machines virtuelles sont orchestrées par une entité nommée "gestionnaire de machines virtuelles" ou *hyperviseur*, qui est la plupart du temps très complexe, parfois propriétaire, et dont le niveau de sécurité est inconnu. Ces gestionnaires peuvent donc contenir des vulnérabilités. Leur compromission est critique car elle met en danger l'ensemble des informations des clients déployés au dessus de ce gestionnaire. Le but de nos travaux est la détection de cette compromission.

Afin de détecter la compromission d'un gestionnaire de machines virtuelles, il est nécessaire de la caractériser et de choisir les mécanismes d'observation adéquats. L'approche que nous avons choisie s'appuye essentiellement sur la virtualisation assistée par le matériel. Or cette virtualisation matérielle est dans la plupart des cas déjà utilisée par les gestionnaires de machines virtuelles, ce qui nous conduit à considérer la virtualisation récursive (*nested virtualization*). Pour atteindre cet objectif, il est nécessaire de mettre en place un gestionnaire de machines virtuelles de niveau 1, noté **l1**, n'exécutant qu'une seule machine virtuelle qui correspond au gestionnaire observé, de niveau 2, noté **l2**. Ce gestionnaire d'observation sera vérifiable du fait de sa petite taille. Lorsqu'il détecte une compromission de **l2**, son rôle est d'émettre une alerte auprès d'un *Security Information and Event Manager* (SIEM), voire de redéployer les machines virtuelles concernées via une machine de confiance.

*. LAAS-CNRS 7 Avenue du Colonel Roche, 31400 Toulouse
Tel : 33 (0) 5 61 33 62 00 - Fax : 33 (0) 5 61 55 35 77

2 Détection de compromission

2.1 Architecture de l'hyperviseur

Afin de surveiller le fonctionnement de **l2**, nous avons besoin d'installer **l1** en premier. Nous avons opté pour un *hyperviseur bare metal* indépendant, ne nécessitant pas de système d'exploitation pour s'installer. Ce type d'*hyperviseur* s'installe juste après l'exécution du *firmware* de la machine physique. Il existe deux types de *firmware*, le *firmware* "historique" (le BIOS), et la nouvelle génération (l'UEFI). Ce dernier standardise son interface avec le système d'exploitation. Il peut charger des images permettant, par exemple, de piloter des composants. Se comportant comme un système d'exploitation rudimentaire, il peut protéger l'espace mémoire de ces images et indiquer à **l2** que cet espace est réservé. Ce mécanisme évite la virtualisation des anciens services proposés par le BIOS [1]. Ces avantages nous ont amenés à implémenter une version UEFI de **l1**.

En cas de compromission de **l2**, **l1** doit alerter le SIEM. Pour communiquer ces alertes, notre choix s'est porté sur un transfert réseau [2]. Aussi, un pilote minimaliste UEFI configure la carte réseau et partage son adresse PCI avec **l1** afin que ce dernier la masque aux yeux de **l2**. Sans ce masquage, **l2** s'approprierait cette carte lors de son lancement et elle ne serait plus utilisable par **l1**.

Lors de la phase d'installation, le pilote réseau UEFI ainsi que **l1** sont chargés en mémoire. Ce dernier active la virtualisation puis crée et initialise une structure de contrôle de machine virtuelle ou VMCS. Cette structure est utilisée par le processeur pour effectuer les changements de contexte hyperviseur / machine virtuelle et définir les évènements provoquant ces changements. Une fois chargés, **l2** s'installe tout en contrôlant son exécution. En particulier, **l2** utilise les instructions de virtualisation. **l1** les utilise déjà. Il lui est donc nécessaire de les simuler pour permettre à **l2** de fonctionner sans modification.

L'assistance matérielle pour la virtualistation impose un gestionnaire de machine virtuelle par coeur. Afin que **l2** ne s'accapare pas les ressources de virtualisation, il nécessaire d'installer au préalable **l1** sur chaque coeur avant de donner la main au niveau supérieur.

2.1.1 Moyens d'observation

Il existe plusieurs moyens d'observation offerts par l'assistance matérielle pour la virtualisation. Le premier s'effectue au travers de la VMCS. Il est possible, grâce à des *I/O bitmaps* stockés dans cette structure, de contrôler les accès *I/O* de **l2** [3] qui doivent être contrôlés par **l1**. De la même manière, il est possible d'indiquer dans la VMCS, les registres de contrôle (CR3, CR0, IDTR, GDTR, etc) dont les accès par **l2** doivent être contrôlés par **l1**. Le second repose sur l'observation, lorsque **l1** s'exécute, du contenu des caches, des compteurs du processeur ou de la durée entre différents événements. Par exemple, la fréquence faible de changements de contexte peut être un élément révélateur d'un déni de service.

1. en particulier, l'interruption *e820*
2. certains *hyperviseur* choisissent de communiquer via USB
3. réalisés par les instructions assembleur `in` et `out`

2.1.2 Approche pour la détection

La détection de compromission s'effectue par une machine de confiance à laquelle **l1** envoie régulièrement tous les éléments observés cités ci-dessus. Cette machine doit posséder un modèle de référence du comportement attendu de **l2**. Le modèle peut être formalisé, par exemple, par des automates finis.

Remerciements

Ces travaux de recherche sont soutenus par le projet français d'Investissements d'Avenir Secured Virtual Cloud (SVC).