HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

OSSIR Groupe SUR
8 juillet 2008
VMware et sécurité

Julien Raeis <Julien.Raeis@hsc.fr>
Nicolas Collignon <Nicolas.Collignon@hsc.fr>

Agenda

- Rappels sur la virtualisation
- Mesures de sécurité intégrées à VMware
- Virtualisation et DMZ
- Revue des vulnérabilités
- Retour d'expérience HSC

Rappels sur la virtualisation

Définitions

- « Virtuel : Se dit des éléments (terminaux, mémoire...) d'un système informatique considérés comme ayant des propriétés différentes de leurs caractéristiques physiques » - *Grand Dictionnaire Encyclopédique Larousse*

- « Virtualisation : abstraction des ressources d'un système informatique. »

Historique

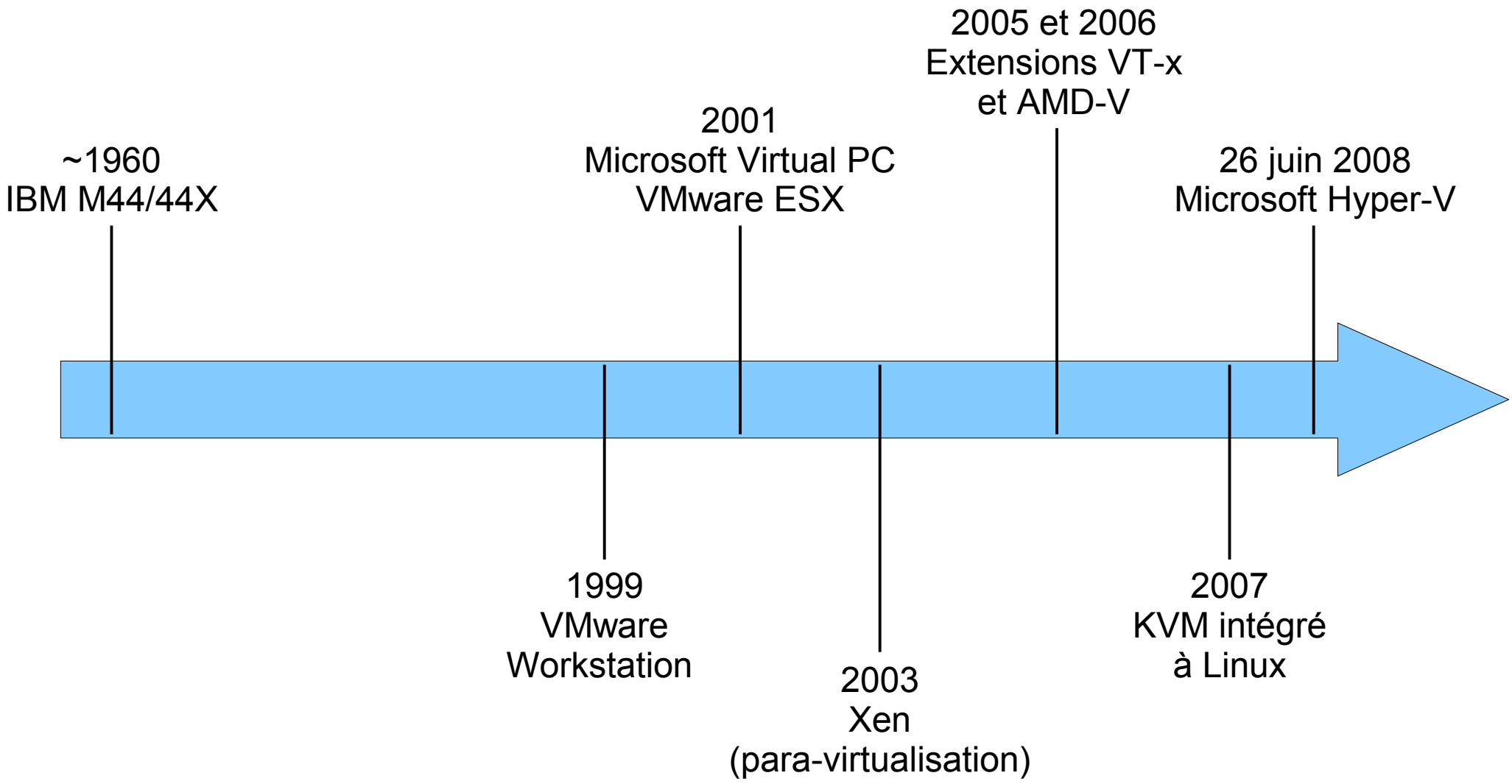
- Concept introduit dans les années 60
  - But : partitionner les ressources des coûteux mainframes de l'époque
  - IBM M44/44X, naissance du terme « pseudo-machine »
    - Première implémentation de machines virtuelles
  - IBM CP-40
    - Système tournant sur S/360-40
- Perte d'intérêt dans les années 80
  - Déport des applications sur des clients et serveurs x86
    - Architecture « bon marché »
  - Mais coûts d'infrastructure physique élevés, manque de protection en cas de panne, maintenance des postes de travail coûteuse, etc.

Virtualisation de l'architecture x86

- Problème
  - Architecture non-prévue pour la virtualisation
  - 17 instructions ne peuvent être virtualisées simplement

- 1999 : VMware propose une solution
  - Interception (« trap ») et conversion de ces instructions
  - Exécution directe des autres instructions par le processeur

Chronologie



Virtualisation(s) - 1/6

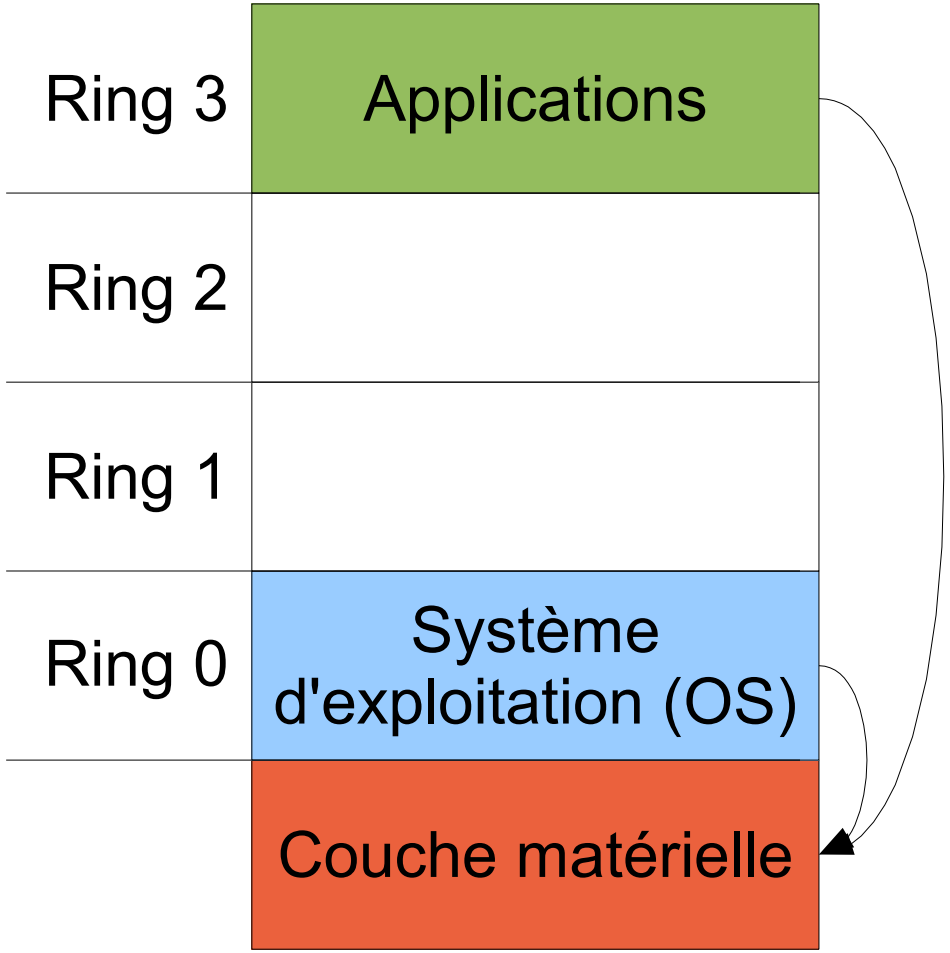
- Au moins 7 types différents !
  - Émulation
  - Virtualisation partielle
  - Virtualisation complète
  - Para-virtualisation
  - Virtualisation native
  - Virtualisation par zones
  - Virtualisation applicative
  - etc.

Virtualisation(s) - 2/6

- Émulation
  - Simulation intégrale du matériel
  - QEMU, PearPC, Bochs
  - Principe des émulateurs des vieux ordinateurs/consoles de jeu
    - Amiga, Atari, etc.
- Virtualisation partielle
  - Partage de ressources matérielles par abstraction
  - Implémentation répandue
    - Adressage virtuel des processus
    - Linux, Windows, etc.

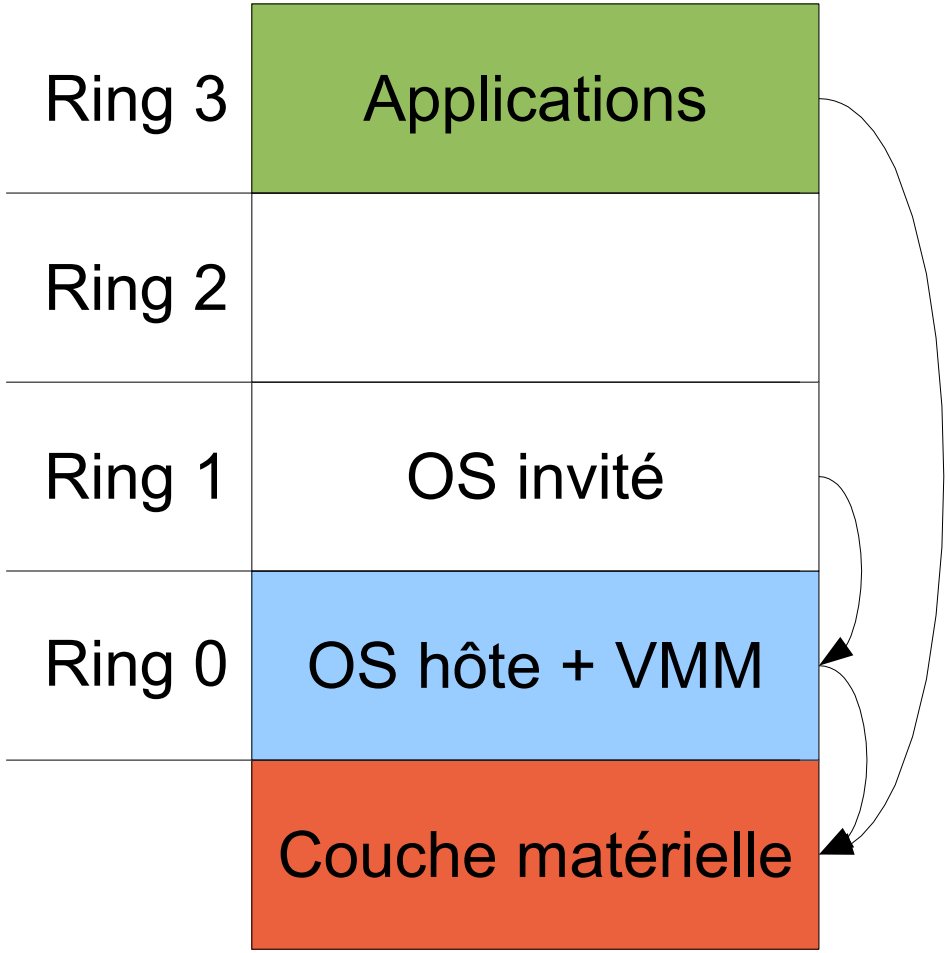
Virtualisation(s) - 3/6

- Exécution classique
  - Applications en Ring 1, 2 ou 3
  - Système d'exploitation en Ring 0
  - Exécution indépendante



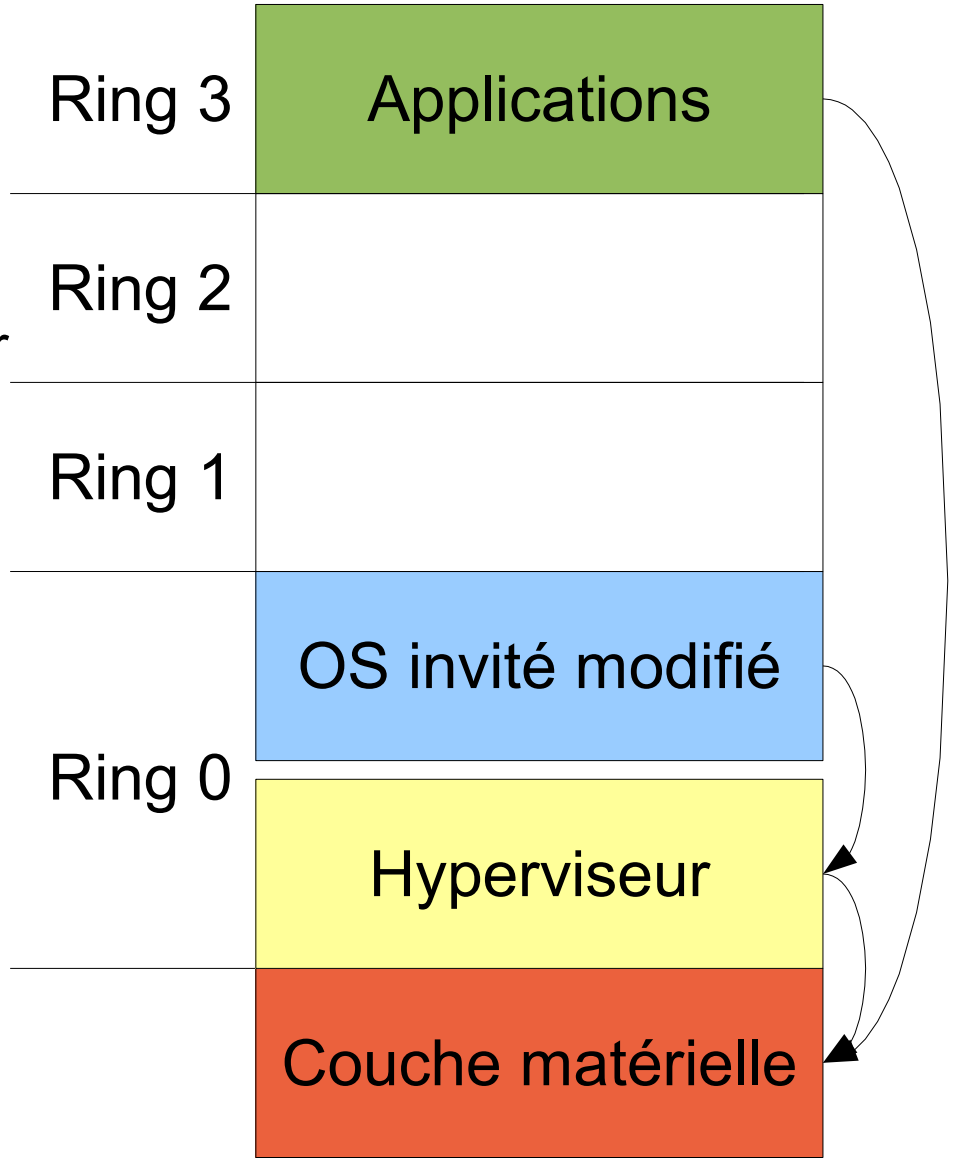
Virtualisation(s) - 4/6

- Virtualisation complète
  - Applications en Ring 3
  - Système invité en Ring 1
  - En Ring 0
    - Système d'exploitation hôte
    - Moniteur de machines virtuelles
  - Technique de traduction binaire
    - À la volée par VMware par exemple
    - Traduit les instructions non-virtualisables



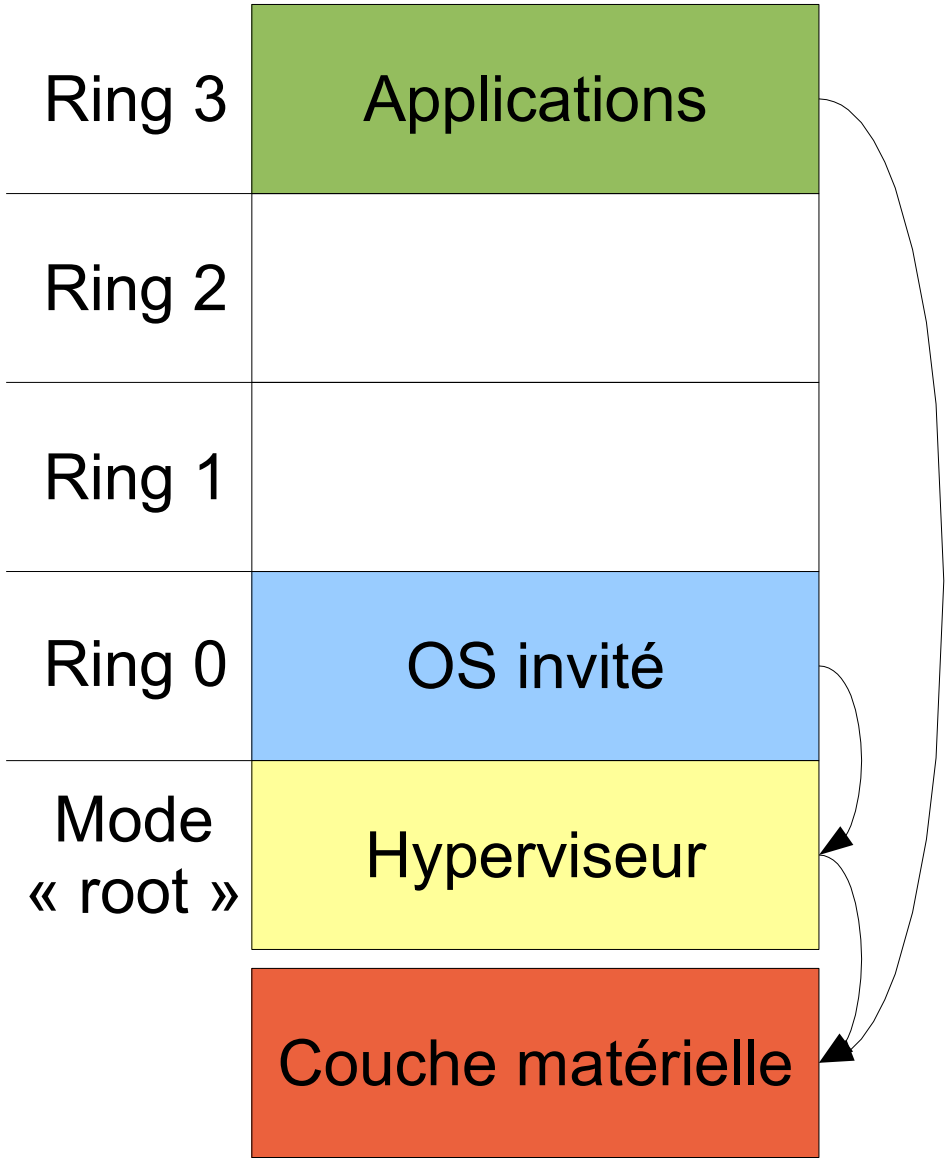
Virtualisation(s) - 5/6

- Para-virtualisation
  - Applications en Ring 3
  - En Ring 0
    - Système d'exploitation modifié pour réaliser des appels (« hypercalls ») directement à la couche de virtualisation
    - Couche de virtualisation tournant dans l'OS hôte
  - Les instructions non-virtualisables sont directement traduites par le biais d'appels spéciaux



Virtualisation(s) - 6/6

- Virtualisation native
  - Applications en Ring 3
  - En Ring 0
    - Système d'exploitation invité NON-modifié
  - Couche de virtualisation tourne « sous » le mode Ring 0
  - Traitement systématique des instructions problématiques par la couche de virtualisation
    - Extensions des processeurs jouent le rôle des hypercalls
  - État des invités stocké dans des structures dédiées du mode racine



Agenda

Mesures de sécurité intégrées à VMware

Deux types de produits

- Deux types de produits
  - Hébergés (« hosted »)
    - VMware Workstation, Server, Player, Fusion
    - Moniteur de machines virtuelles tourne sur l'OS hôte
  - Hyperviseur
    - VMware ESX et ESXi
    - « VMKernel » pour le rapport avec le matériel et la virtualisation
    - Système Linux pour le charger en mémoire, ensuite virtualisé

VMware ESX

- Avant la 3.0
  - Démarrage sur un système Linux comme support
  - Chargement de modules propriétaires VMware
    - Moniteur de machines virtuelles
    - Gestion du système de fichiers VMFS
- Depuis la 3.0
  - Démarrage sur un système Linux
  - Chargement de modules propriétaires
    - Pivot sur « VMkernel » qui prend la main
    - Chargement des pilotes matériels par VMkernel (et non Linux)
    - Virtualisation du système Linux sous-jacent (« Service console »)
    - Lancement du moniteur de machines virtuelles, pilote VMFS, etc.

Mécanismes de sécurité - 1/3

- Authentification et contrôle d'accès
  - Utilisation de PAM pour VMware Server et ESX sous Linux
  - Interfaçage possible avec Active Directory
- Communications chiffrées
  - Entre VMware Infrastructure Client/Server Console et le serveur
- Isolation entre hôte et invités
  - Par l'hyperviseur, au niveau système et réseau (virtuel, bien sûr)
- Bientôt : VMSafe
  - API de communication avec l'hyperviseur
  - Prochaine version d'ESX
  - Tellement sûr que VMware demande un NDA pour avoir des infos

Mécanismes de sécurité - 2/3

- Options de configuration (pas toujours) documentées
  - http://sanbarrow.com/vmx.html
  - Notamment, pour la sécurisation :

```
Isolation.tools.copy.enable = FALSE        # Copier
isolation.tools.paste.enable = FALSE       # Coller
isolation.tools.hgfs.disable = TRUE        # Dossiers partagés
isolation.tools.dnd.disable = TRUE         # Drag'n'Drop
...
```

- Restrictions d'authentification par PAM

```
#%PAM-1.0
auth       required     pam_unix.so shadow nullok
account    required     pam_listfile.so item=group sense=allow
                        file=/etc/vmware/vmwaregroup onerr=fail
account    required     pam_unix.so
```

Mécanismes de sécurité - 3/3

- *Service console* de VMware ESX
  - Pare-feu par l'outil « esxcfg-firewall »
  - Basé sur netfilter/iptables
  - MAIS ! Interdiction de rajouter des règles manuellement sous peine de perdre le support VMware
- Autres mécanismes d'ESX
  - Propagation du bit NX aux processeurs virtuels
  - Désactivation de l'Hyper-threading
  - Système de rôles (type RBAC) pour les utilisateurs de VI
  - Protections réseau niveau 2
  - Segmentation réseau par VLANs (sur les commutateurs virtuels par exemple)

Revue des vulnérabilités

Historique des avis de sécurité

- Diffusion de correctifs de sécurité
  - De 2003 à 2005
    - Pas de « centre de sécurité VMware »
    - 10 vulnérabilités corrigées
  - Puis mise en place d'un système d'alertes et d'avis
    - 2006 : 10 avis pour 17 vulnérabilités
    - 2007 : 6 avis pour 25 vulnérabilités
    - 2008 : 10 avis pour pour 31 vulnérabilités (au 16 juin 2008)
- Avis touchent à la fois
  - Les produits VMware
  - Les paquets tiers (Service Console de ESX Server)

Récapitulatif des vulnérabilités

Année	Total	ESX	Virtualisation hébergée	Produits VMware	Produits tiers	Sortie de l'isolation
2003	3	3	1	0	3	0
2004	6	5	2	3	3	0
2005	2	0	2	2	0	1
2006	17	16	2	7	10	0
2007	25	23	8	8	17	1
2008	31	25	14	8	23	2

- 4 sorties d'isolation
  - Depuis l'invité vers l'hôte
  - 3 nécessitent des conditions particulières, non-présentes par défaut
  - 2 sont uniquement valables sur VMware Workstation et Player
- Quelques élévations de privilèges
  - Dans l'invité avec les VMware Tools notamment
  - Dans l'hôte par des produits tiers ou vmware-authd

Retour d'expérience

Expériences HSC

- Audits sur plates-formes de virtualisation
  - ESX Server uniquement
  - Audit de configuration
  - Audit d'architecture
  - Intégration de la virtualisation en DMZ
- Travaux de recherche
  - Communications avec l'hyperviseur
  - Protocoles de communication
  - Gestion des sessions
  - Les « Shared Folders »
  - Comment exploiter une vulnérabilité dans l'hyperviseur

Configuration - 1/2

- Correctifs de sécurité
  - Pas d'automatisation
  - Déploiement régulier tout de même
- « Service console »
  - Minimisation des services déployés
    - Outils de supervision CIM
    - Activation du SNMP
    - Serveur HTTP de gestion désactivé
  - Activation du pare-feu
  - Restriction du service SSH

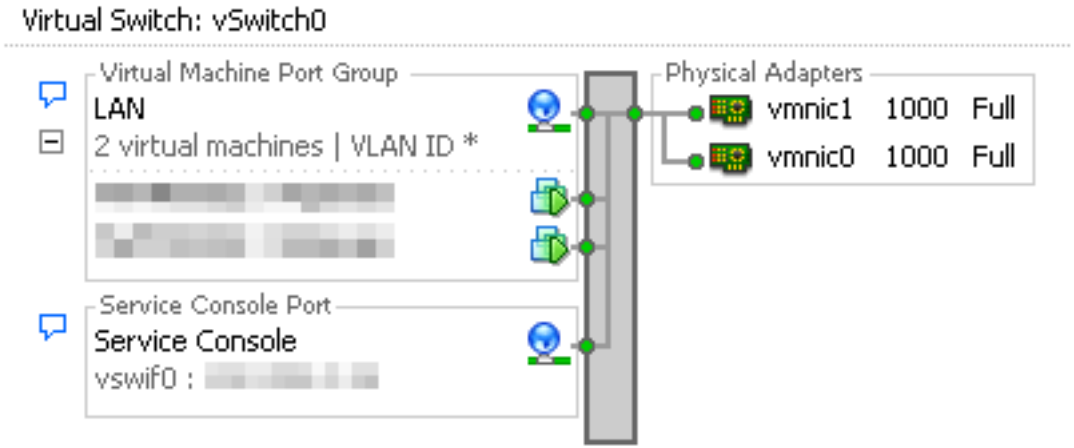
Configuration - 2/2

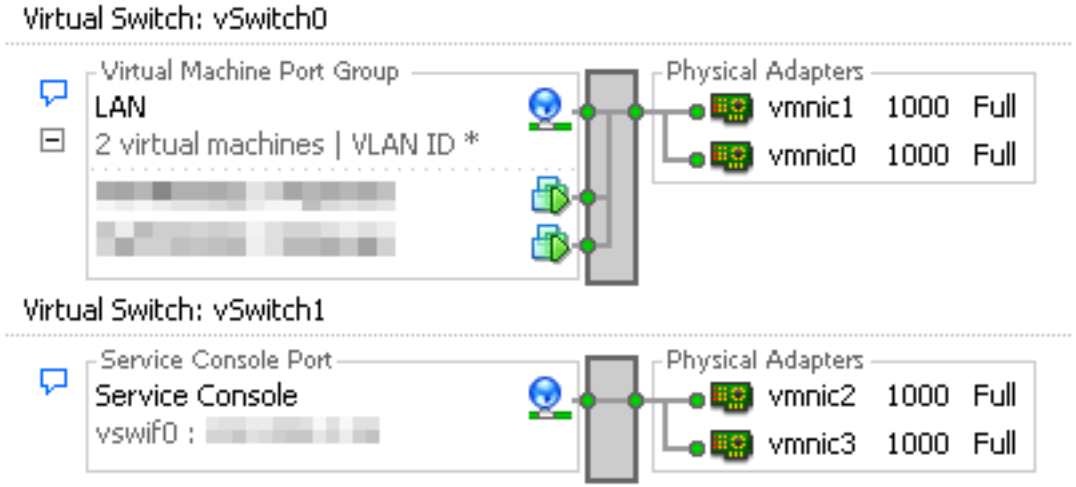
- VMware Virtual Center
  - Création d'utilisateurs et rôles précis
    - Administrateurs de machines virtuelles (avec droit de redémarrage)
    - Administrateurs ESX (accès à la configuration des VMs)
  - Partage des ressources strict pour éviter les dénis de service
- Systèmes invités
  - Déploiement des VMware Tools (minimisés)
  - Considérés comme des machines physiques
  - Options d'isolation activées
    - Copier/coller, *Drag'n'Drop*, etc.
  - Suppression des périphériques virtuels inutiles

Architecture - 1/2

- Séparation stricte de la console de service

Pas bien !





Bien !

Architecture - 2/2

- Autres possibilités d'amélioration
  - Pare-feu de périmètre
  - Pare-feu virtuel (cf. plus loin)
  - Utilisation de VLANs
    - Dans ce cas, première architecture vue précédemment peut être OK
    - Trois niveaux :
      - Invités
      - Commutateurs virtuels
      - Commutateur physique à la sortie du réseau virtuel
  - Dans tous les cas, filtrer les entrées/sorties de la console de service
    - Point critique : hôte compromis == architecture compromise

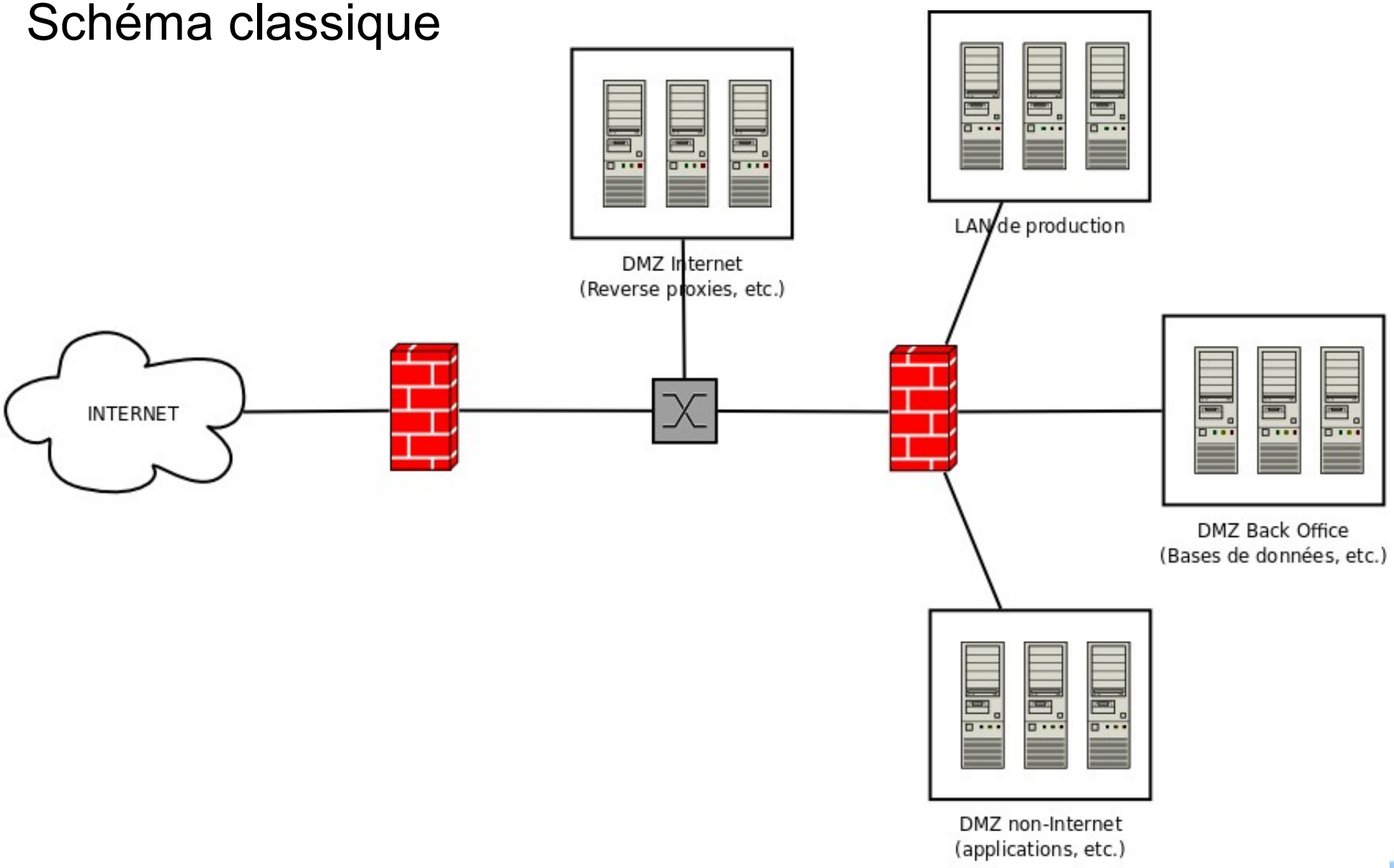
Intégration en DMZ - 1/5

- Trois possibilités
  - Chacune offre un degré supplémentaire de virtualisation :

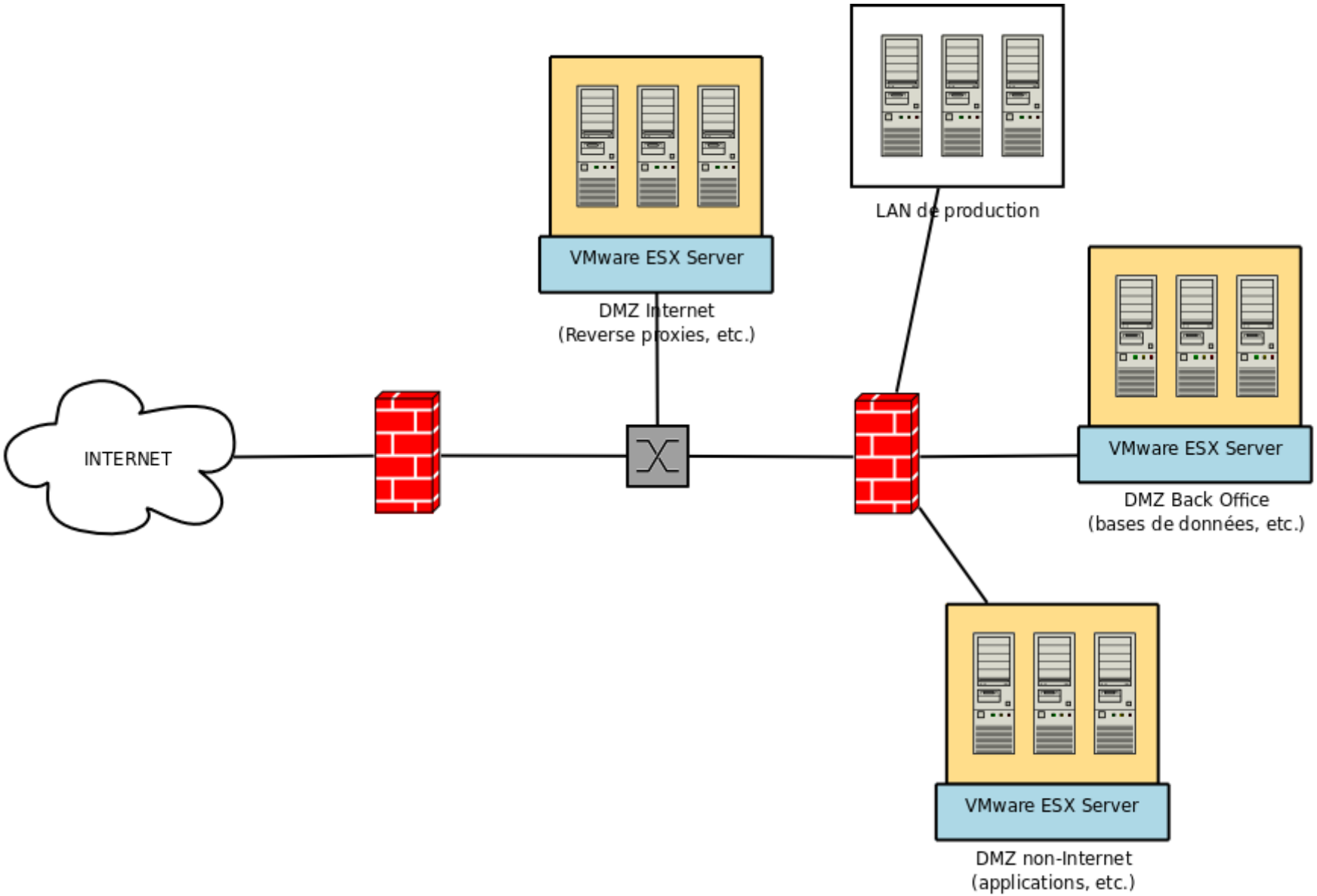
1. Virtualisation de chaque DMZ
2. Virtualisation globale de toute les DMZ avec segmentation
3. Virtualisation globale, pare-feu inclus

Intégration en DMZ - 2/5

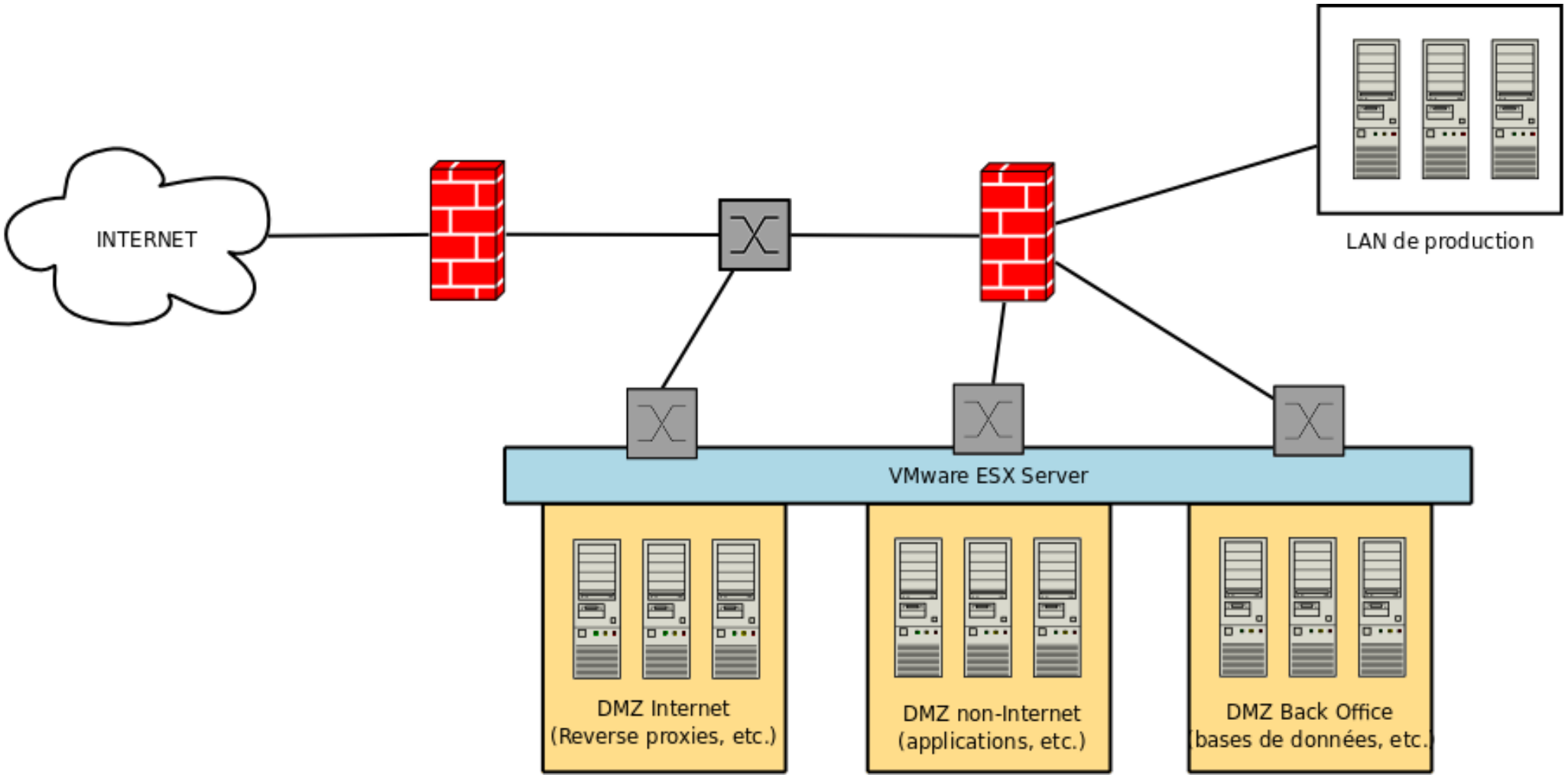
- Schéma classique



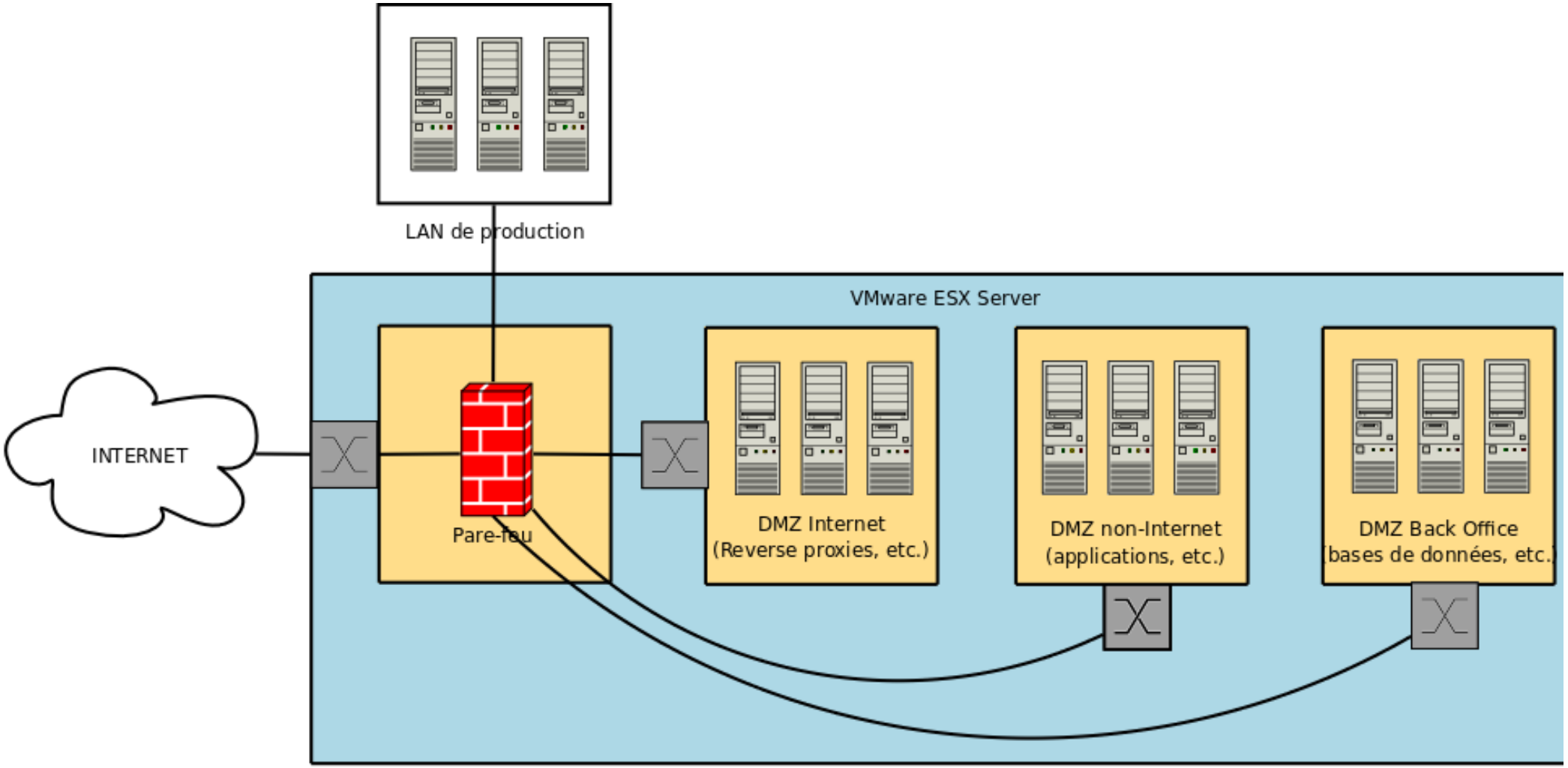
Intégration en DMZ - 3/5



Intégration en DMZ - 4/5



Intégration en DMZ - 5/5



Un peu de recherche

Les interrogations

- Comment l'hyperviseur peut-il obtenir des informations sur les systèmes virtualisés ?
  - Adresses MAC
  - Adresses IP
  - ...
- Par où passent les I/O sur les « Shared Folders » ?
- Copy/paste, DND, obscurité, ...

Backdoor : pourquoi ?

- Nécessité de communiquer entre l'invité et l'hyperviseur
- Doit fonctionner sans que l'invité puisse établir des connexions réseaux vers l'hyperviseur
- Qui ?
  - Driver HGFS
  - VMware Tools
  - API Vix
  - ...

Backdoor

- Terminologie non fantasmée :)

```
$ strings vmware-vmx | grep -i backdoor | wc -l
23
```

- 2 ports I/O

0x5658	VX	commandes
0x5659	VY	lectures / écritures

Backdoor : authentification ?

- Qui peut utiliser la backdoor depuis l'invité ?
  - Le kernel (ex: `hgfs.sys`)
  - Les applications (ex: `VmwareHgfsClient.exe`)

- Aucun point de centralisation dans l'invité
  - Impossible de déterminer les droits du client
  - Du point de vue de l'hyperviseur :

```
root = nobody
SYSTEM = Invité
```

Backdoor : utilisation sur x86

- Paramètres des commandes passés par les registres

Registre	Contenu
eax	Signature « VMXh »
ebx	Argument spécifique à la commande
ecx	Commande
edx	Descripteur de canal (Channel handle)
esi edi ebp	Dépend du type de commande VX ou VY

- Lecture (in) ou écriture (out) sur les 2 ports d'I/O

```
movl   $0x00005658, %edx      /* VX */
movl   $0xffff000a, %ecx
movl   $0xa9b2a797, %ebx
movl   $0x564d5868, %eax      /* VMXh */
inl    (%dx), %eax
```

Backdoor : utilisation sur x86

```
C:\temp> vmrpc -v call "info-get guestinfo.ip"
io[c] ebx=c9435052 ecx=0000001e edx=ffff5658 esi=00000000 edi=00000000 ebp=00000000
      ebx=c9435052 ecx=00010000 edx=00050000 esi=9c0eefdb edi=7d721754 ebp=00000000 ==> 1
io[c] ebx=00000015 ecx=0001001e edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000
      ebx=00000015 ecx=00810000 edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000 ==> -127
io[w] ebx=00010000 ecx=00000015 edx=00055659 esi=003d23cf edi=7d721754 ebp=9c0eefdb
      ebx=00010000 ecx=00000000 edx=00055659 esi=003d23e4 edi=7d721754 ebp=9c0eefdb ==> 1
io[c] ebx=00000015 ecx=0003001e edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000
      ebx=00000009 ecx=00830000 edx=00010000 esi=9c0eefdb edi=7d721754 ebp=00000000 ==> -125
io[r] ebx=00010000 ecx=00000009 edx=00055659 esi=9c0eefdb edi=003d2510 ebp=7d721754
      ebx=00010000 ecx=00000000 edx=00055659 esi=9c0eefdb edi=003d2519 ebp=7d721754 ==> 1
io[c] ebx=00000001 ecx=0005001e edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000
      ebx=00000001 ecx=00010000 edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000 ==> 1
 ==> 1 192.168.0.11
io[c] ebx=00000000 ecx=0006001e edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000
      ebx=00000000 ecx=00010000 edx=00055658 esi=9c0eefdb edi=7d721754 ebp=00000000 ==> 1
```

Backdoor : utilisation « annexe »

- Détecter si un système est virtualisé ou pas

  Une méthode parmi beaucoup d'autres (IDT, GDT, ACPI ...)

- Historique
  2002 Premiers virus détectant VMware
  2005 Premiers virus infectant les invités depuis l'hôte
  2007 Une variante de « Storm » détecte VMware et Virtual PC

- Modification du flot d'exécution
  - Bloquer le code malveillant pour éviter l'analyse dans une « sandbox »
  - Modifier la charge utile pour infecter l'environnement virtualisé

Backdoor : protocoles

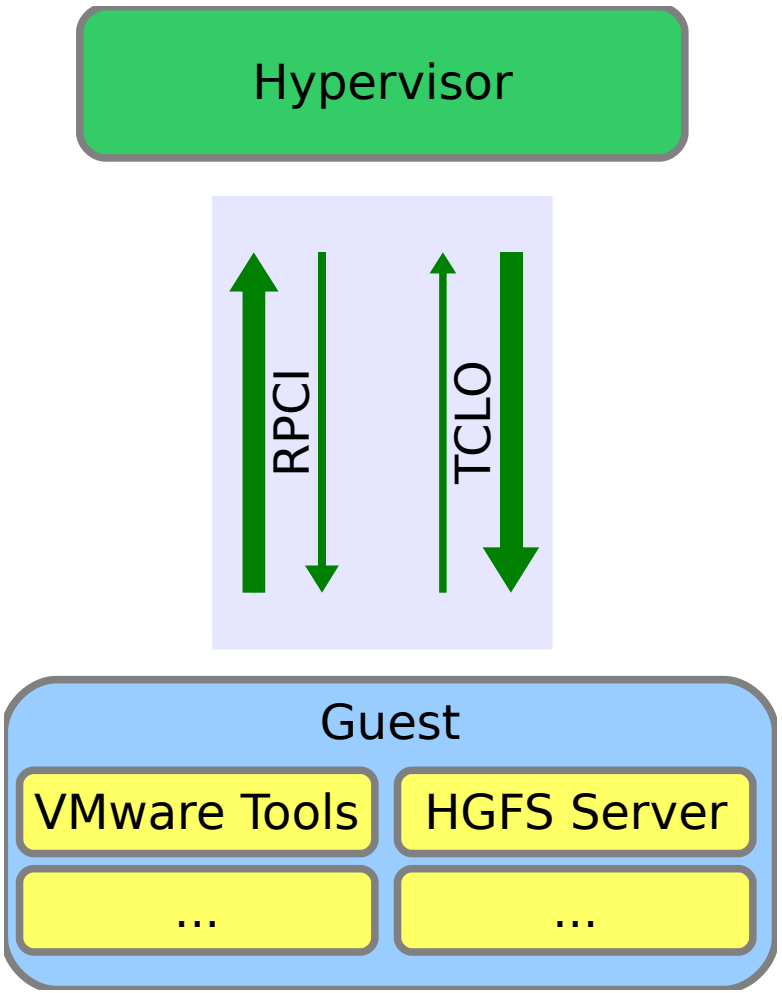
- 2 ~protocoles~ de communication

- **RPCI**

  À l'initiative du guest

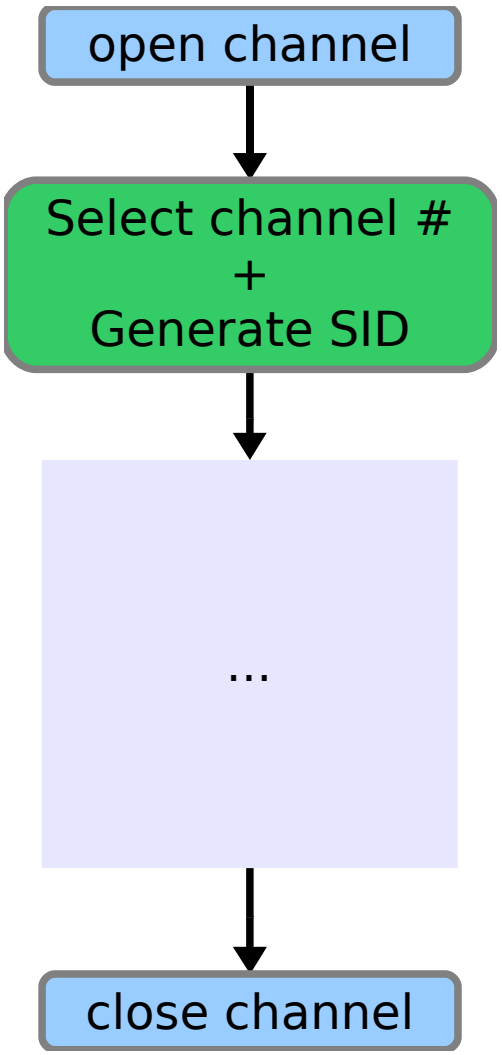
- **TCLO**

  À l'initiative de l'hôte



Concepts

- Protocole « connecté »
- Principalement ASCII
- Descripteur de canal
  - Codé sur 16 bits
  - ~8 descripteurs disponibles (0-8)
- Identifiant de session
  - Codé sur 64 bits
  - Généré par l'hyperviseur (aléatoire)



Protocole RPCI

- ~ **50** commandes

```
log <msg>
tools.set.version <ver>
upgrader.setGuestFileRoot <int> <path>
info-get guestinfo.<key>
info-set guestinfo.<key> <val>
...
```

- Réponse de l'hyperviseur : `<0|1> [data]`

- Ex : commande `log`

```
C:\temp> vmrpc call "log coucou c'est nous"
 ==> 1
```

```
vcpu-0| Guest: coucou c'est nous
```

Protocole RPCI

- Possibilité de bloquer **presque** toutes les commandes

Commande	Paramètre
`tools.capability.resolution_set`	`resolutionSetDisable`
`tools.capability.resolution_min`	`resolutionMinDisable`
`tools.capability.printer_set`	`printerSetDisable`
`vmx.capability.edit_scripts`	`scriptEditDisable`
`copypaste.hg.copy.files`	`copyDisable`
`disk.shrink`	`diskShrinkDisable`
`...`	`...`

- Attention aux effets de bord ... :)

Protocole TCLO

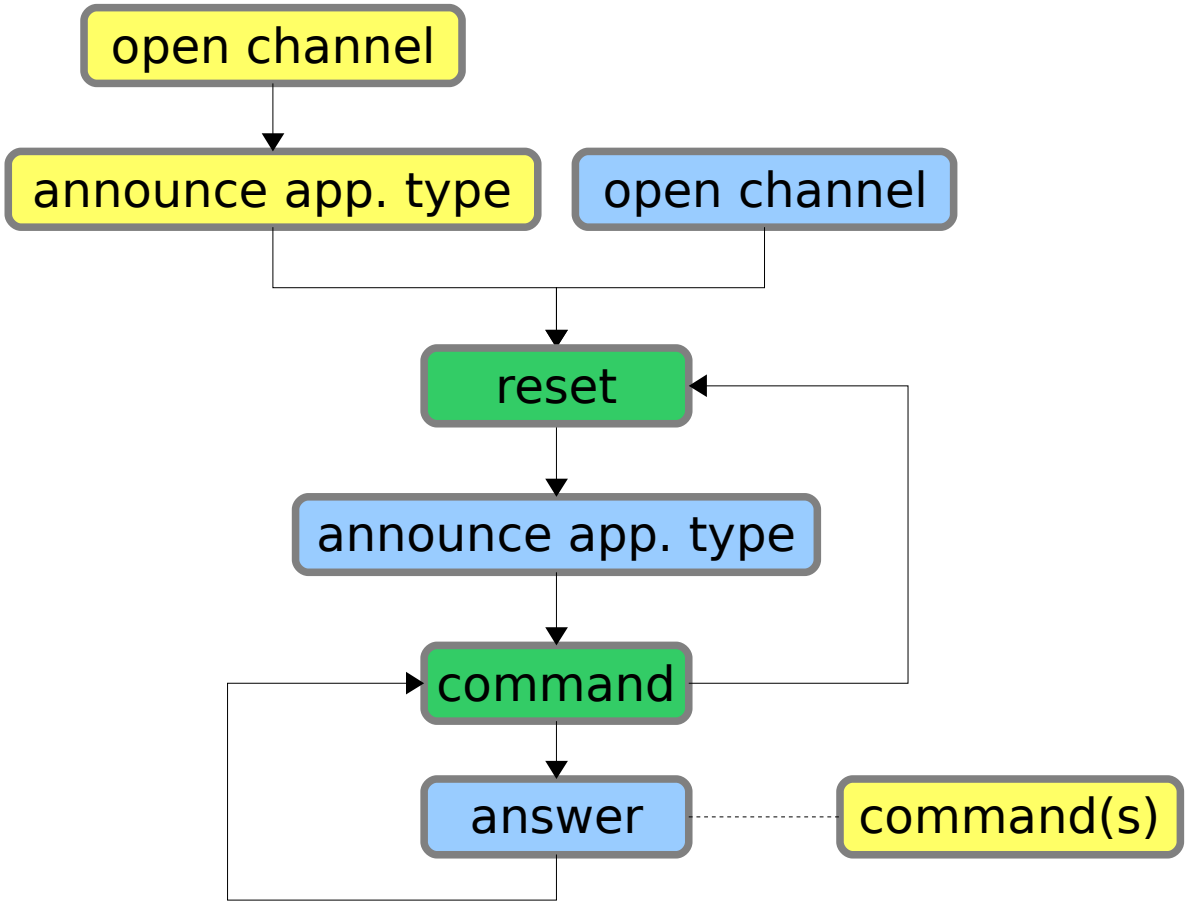
- Lecture de commandes TCLO à intervalles prédéfinis
- ~ 30 commandes gérées par l'hyperviseur

```
Capabilities_Register
Set_Option <str> <str>
Vix_1_Open_Url <str> <str> <str>
Time_Synchronize <int>
Resolution_Set <int> <int>
OS_Suspend
...
```

- Réponse de l'invité : `<OK|ERROR> [data]`

Protocole TCLO : flux d'exécution

```
tools.capability.hgfs_server toolbox 1
reset
OK ATR toolbox
Capabilities_Register
tools.capability.statechange
tools.capability.auto_upgrade 2
tools.capability.guest_temp_directory 1 C:\temp
vmx.capability.unified_loop toolbox
OK
Set_Option broadcastIP 1
info-set guestinfo.ip 0.0.0.0
OK
```



RPCI / guest

TCLO / guest

TCLO / host

Les sessions

- Secrets de sessions
  - généré via `/dev/urandom` sous Linux (pas de `#ifdef PURIFY`)
  - 64 bits répartis sur 2 registres
    - `esi` + `edi` pour les requêtes VX
    - `ebp` + `edi` pour les I/O VY
- Expiration des sessions après inactivité : ~1 min
- Certaines sessions sont quasi-permanentes
  - Ex: service VMware, HGFS, etc.

Attaques sur les sessions ?

- Pourquoi ?
  - Pas d'authentification
  - « spoofer » un client RPCI/TCLO depuis un compte non privilégié de l'invité vis-à-vis de l'hyperviseur.
  - Intercepter les opérations de DND sur la console à distance
  - Remonter des fausses informations dans la console de supervision depuis un compte non privilégié
  - Empêcher le bon fonctionnement des agents VMware depuis un compte non privilégié
- Intérêt limité si accès local privilégié

Sessions : pseudo DoS

- Blocage des canaux RPC disponibles

```
C:\temp> vmrpc block
blocked channel 5
blocked channel 6
blocked channel 7
blocked channel 1
blocked channel 0
```

```
vcpu-0| GuestMsg: Too many channels opened
```

- Seule les sessions permanentes sont utilisables

```
C:\Program Files\VMware\VMware Tools> VMwareHgfsClient.exe
[hgfsclient] WARNING: Failed to create RPC channel
```

Sessions : MITM

- Trouver l'identifiant de 64 bits ?
  - pour intercepter les messages émis par l'hyperviseur
  - pour usurper l'identité d'un client RPC de l'invité

- Journalisation de l'attaque

```
vcpu-0| GuestMsg: Channel 1, Wrong cookie. Man in the middle attack?
vcpu-0| GuestMsg: Channel 1, Wrong cookie. Man in the middle attack?
vcpu-0| GuestMsg: Channel 1, Wrong cookie. Man in the middle attack?
vcpu-0| GuestMsg: Channel 1, Wrong cookie. Man in the middle attack?
...
```

rotation des journaux + commande `log` pour nettoyer les journaux ...

- Trop long ... 100 000 tests / seconde

Sessions : identification applicative

- L'hyperviseur identifie les applications avec un label
  - Spécifié par l'invité
  - Permet souvent de filtrer les messages TCLO envoyés à l'invité

```
vmx.capability.unified_loop <appname>
tools.capability.hgfs_server <appname> <int>
```

- Quelques labels reconnus par l'hyperviseur
  - toolbox, toolbox-ui, toolbox-dnd
  - tools-upgrader, tools-sso, tools-hgfs
  - ...

Sessions : identification applicative

- Journalisation des labels inconnus

```
C:\temp> vmrpc tclo "OK ATR test"
> reset
> Capabilities_Register
```

```
vcpu-0| Tools_SetAppRunningStatus: Unknown app name 'test'
```

- Journalisation des identifications

```
vcpu-0| GuestRpc: Channel 5, registration number 1, guest application tools-upgrader.
```

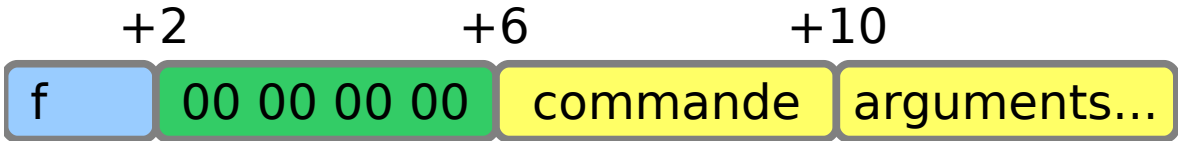
- L'hyperviseur vérifie si l'application est déjà identifiée

```
vcpu-0| GuestRpc: Channel 5, conflict: guest application toolbox-dnd tried to register
                  but it is still registered on channel 2
```

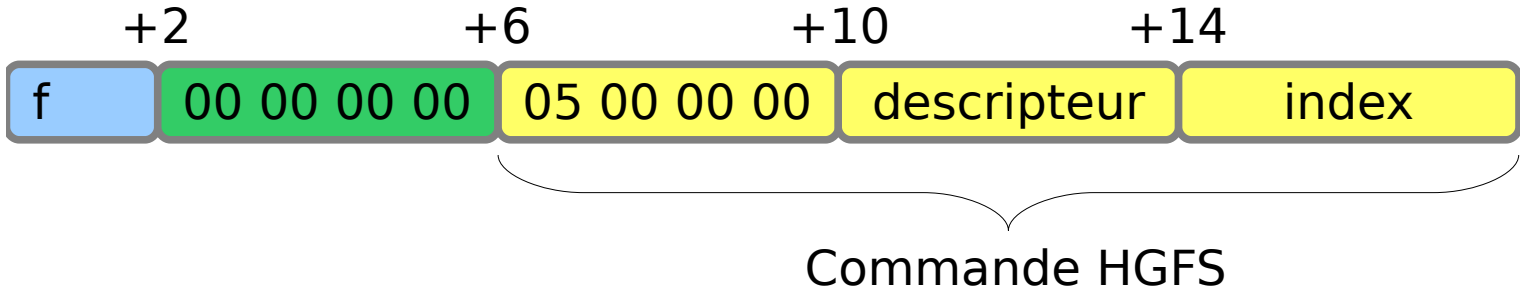
HGFS

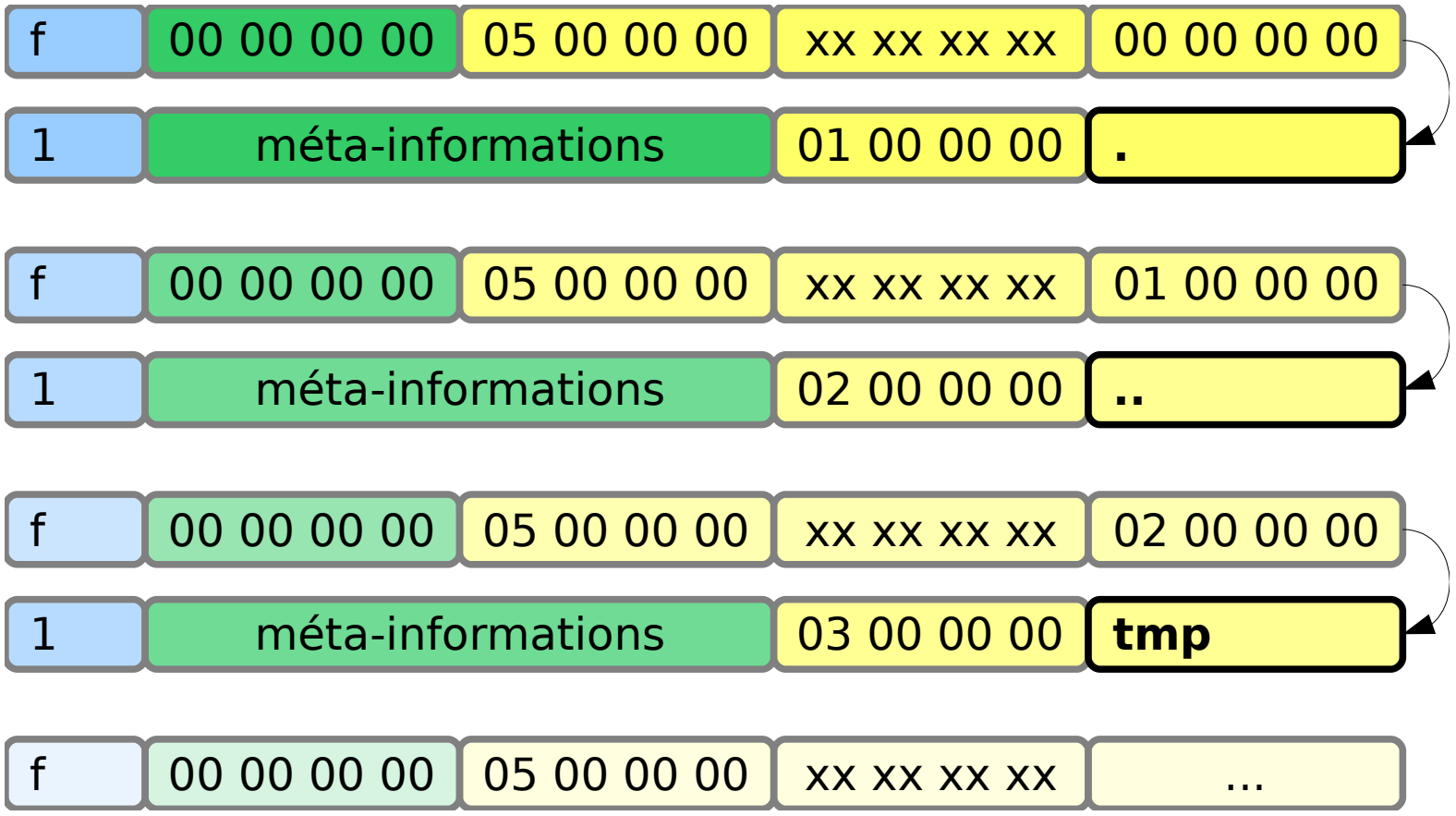
- **H**ost **G**uest **F**ile **S**ystem
- Partage plusieurs répertoires entre l'invité et l'hyperviseur
- I/O à l'initiative de l'invité ► protocole RPCI

  Commande : **`f`**
- Messages binaires
  - ~10 commandes HGFS



HGFS : listage d'un répertoire





Backdoor : en bref ...

- Impossible d'authentifier le client RPCI ou TCLO
- 2 protocoles de communication, **~90** commandes
- Une soupe de messages ASCII, binaires, base64, ...

Un rêve ...

- Et si à tout hasard ...

  Une vulnérabilité dans l'hyperviseur ...

  permettait d'écrire dans la mémoire du processus hôte

- Et si ...

  Une fuite d'information dans l'hyperviseur ...

  permettait de localiser la mémoire de l'invité dans le processus hôte

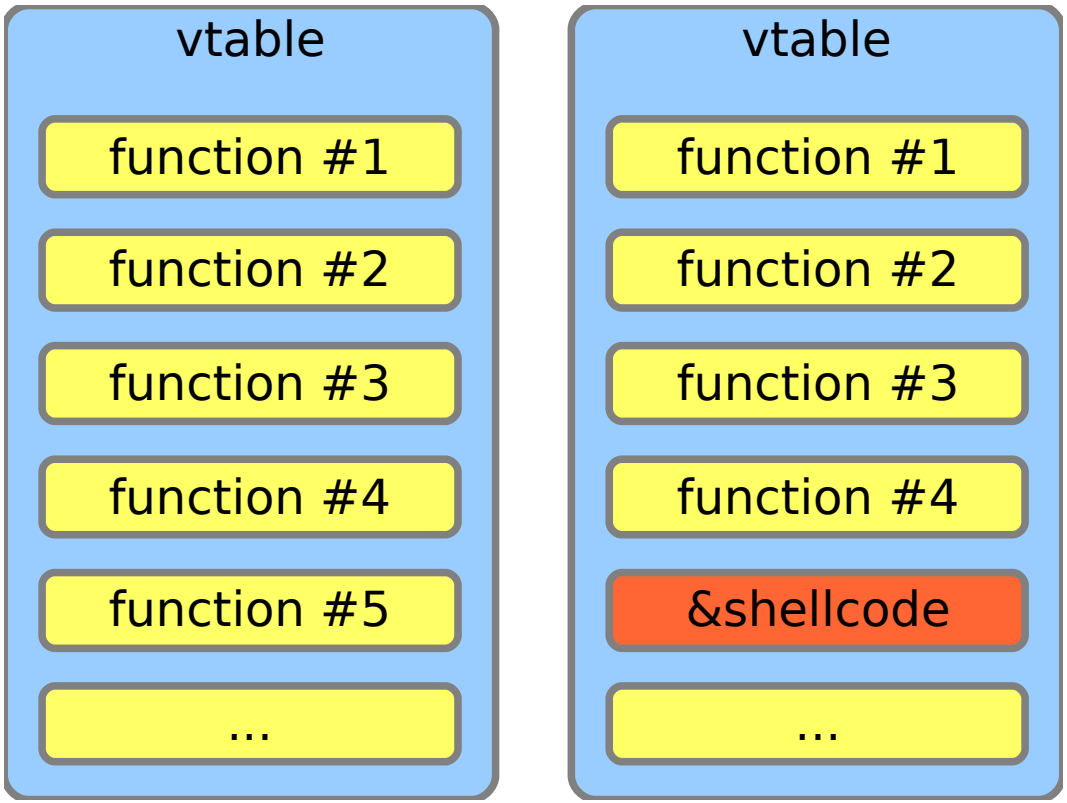
- Alors ...

Exploitation

- Intérêt
  - Obtenir un accès sur l'hyperviseur depuis un compte non privilégié de l'invité
- Problème**S**
  - Ne pas tout casser ...
  - Récupérer le résultat d'un shellcode
- Faits
  - 1 processus dans l'hôte / machine virtuelle
  - 1 thread / session RPCI ►moins de risques de bloquer l'invité

Exploitation

- Surcharge d'une commande RPCI déjà enregistrée
  - Commandes RPCI ► tableau dynamique
  - Commandes HGFS ► tableau statique

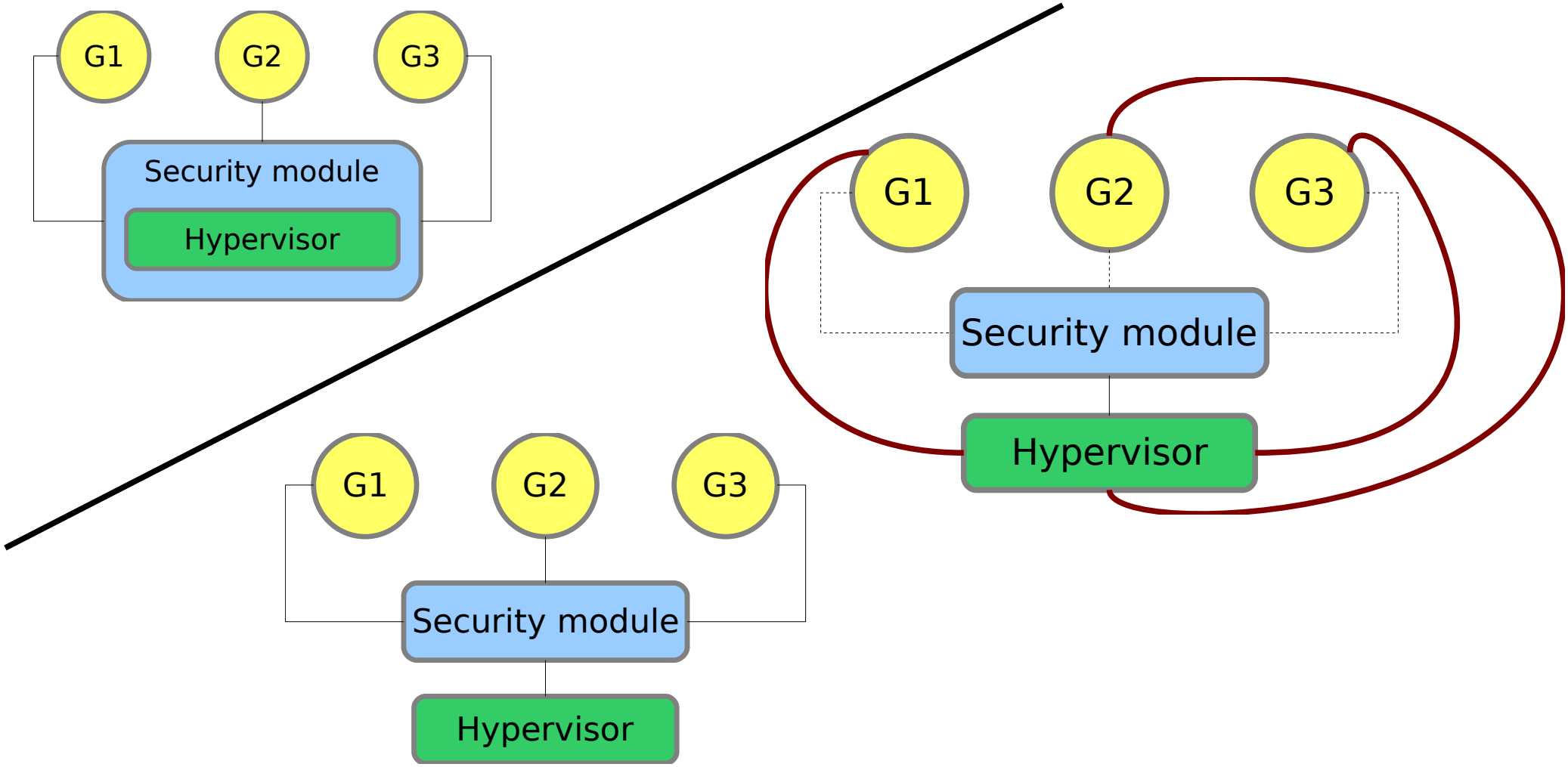


Exploitation

- HGFS
  - Localisation du tableau en identifiant la version de l'hyperviseur ex: commandes RPCI, contenu de l'IDT ou de la GDT, ...
  - Pas disponibles sur toutes les versions de VMware
  - Dysfonctionnement de HGFS
- RPCI
  - Localisation du tableau avec une vulnérabilité annexe
  - Impact minimal
- Shellcode stocké dans les tampons d'I/O RPCI
  - Adresse +/- stable
  - Toujours en mémoire tant que l'I/O n'est pas terminée

Conclusion

Sécurité de la virtualisation ou virtualisation de la sécurité ?



Références

- http://www.vmware.com/
- http://www.vmware.com/security/
- http://communities.vmware.com/
- http://www.virtualization.info/
- http://chitchat.at.infoseek.co.jp/vmware/backdoor.html
- http://sanbarrow.com/

Merci !

Merci de votre attention

Questions ?

Julien.Raeis@hsc.fr

Nicolas.Collignon@hsc.fr

http://www.hsc.fr/