DE LA RECHERCHE À L'INDUSTRIE

Mise en quarantaine du navigateur

Déport graphique

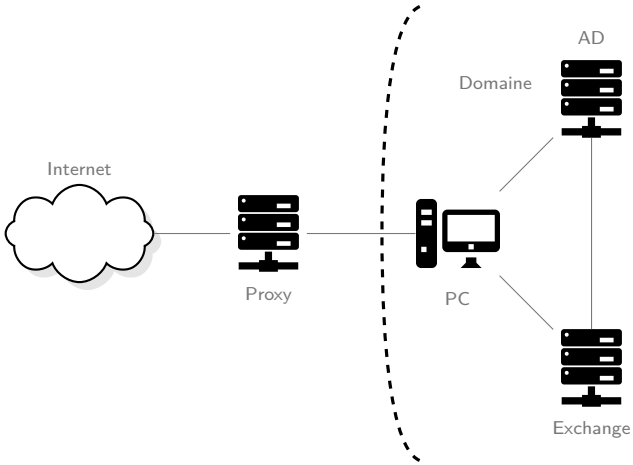
Fabrice Desclaux & Frédéric Vannière

3 Juin 2022

Accès à internet depuis les postes du domaine

Architecture réseau

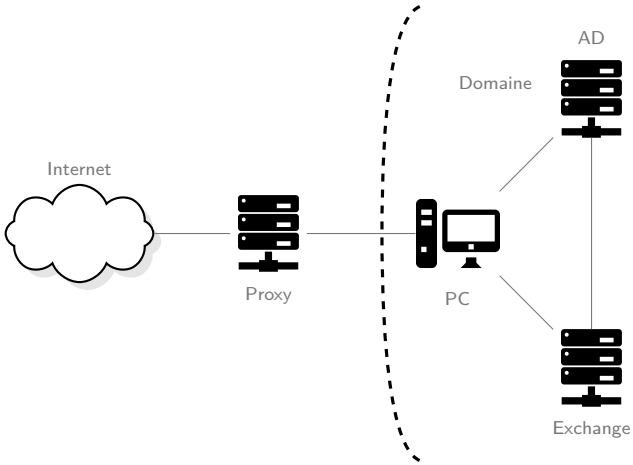
- Domaine Windows
- Proxy



Accès à internet depuis les postes du domaine

Surface d'attaque du navigateur

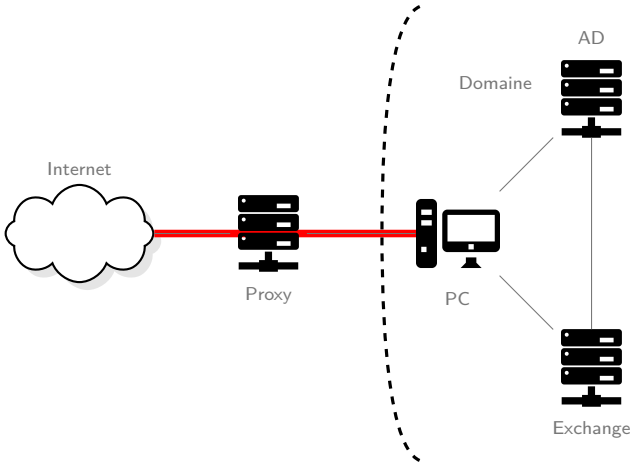
- Html, Javascript, vidéo, ...
- Jitter
- Plugins



Accès à internet depuis les postes du domaine

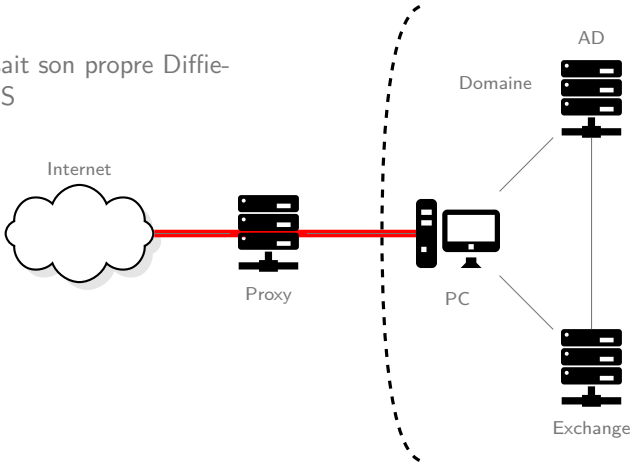
Constat sur les flux :

- 90% du traffic chiffré
- Blacklist difficile



Accès à internet depuis les postes du domaine

Déjà en 2015 Angler Exploit Kit faisait son propre Diffie-Hellman en javascript sous le HTTPS

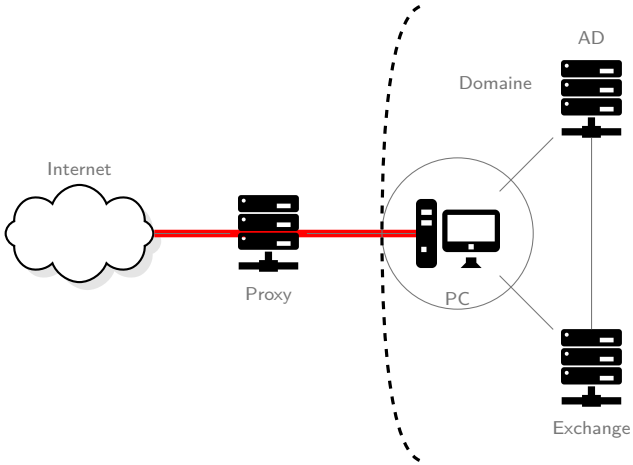


Accès à internet depuis les postes du domaine

Navigateur sur le poste client :

- Accès direct au SI
- Exfiltration aisée

→ Aveugle et désarmé



Solution

Sur le flux

- Man-in-the-middle HTTPS
- Activation des logs sur le poste (SSL, URL navigateur, ...)

Remote

- Citrix / VmWare Horizon / Rdp
- Vnc, Spice, ...

Autre

- Cloudflare browser isolation
- Qubes OS / CLIP OS

Solution : Pro gamer style

Streaming des jeux vidéos

- Le jeu est rendu sur une machine dans le “cloud”
- Le serveur envoie une vidéo streamée au client
- Le client envoie ses entrées (clavier/souris/...)

Services les plus connus

- Stadia (2019)
- GeForce Now (beta 2015)
- Shadow (2015)
- Luna (2020)
- xCloud (2019)
- Parsec (2016)

Appliquer les technologies des jeux vidéos streamés

Stream vidéo : liste au père Noël

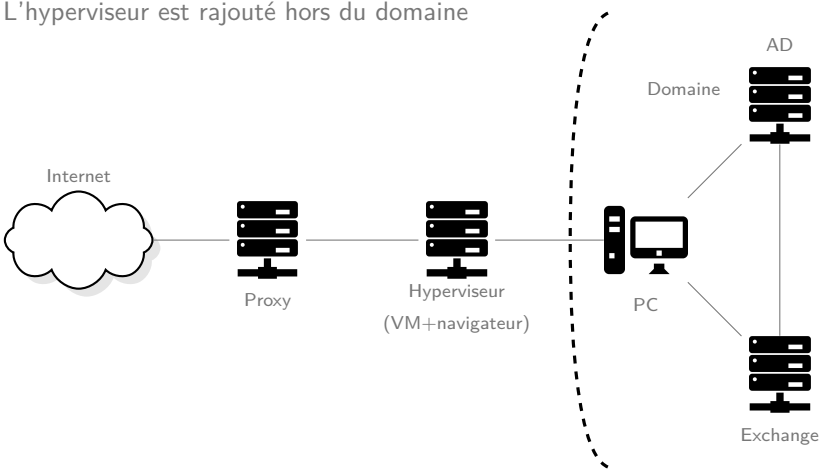
- Protocole maîtrisé / sûr
- Cloisonnement fort
- Limiter la surface d'attaque

Stream vidéo

- Développement d'une solution de stream vidéo
- Hyperviseur hors domaine hébergeant des VMs (une par client)
- Chaque VM exécute un navigateur
- Il est streamé vers le PC client du domaine

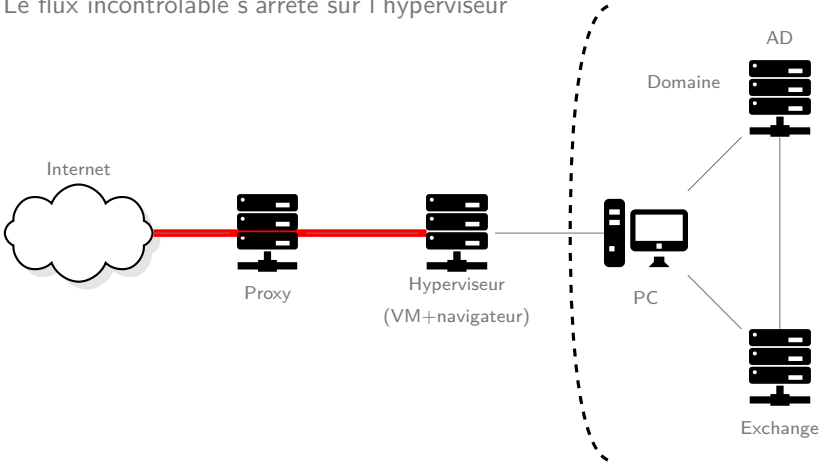
Accès à internet depuis les postes du domaine

L'hyperviseur est rajouté hors du domaine



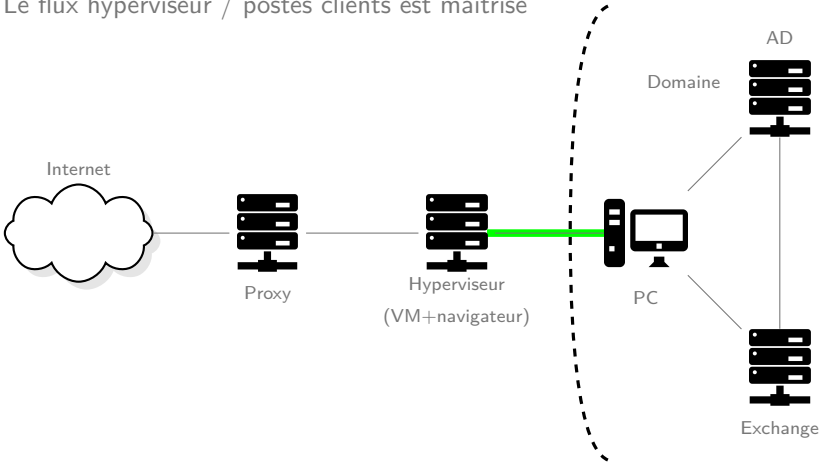
Accès à internet depuis les postes du domaine

Le flux incontrôlable s'arrête sur l'hyperviseur



Accès à internet depuis les postes du domaine

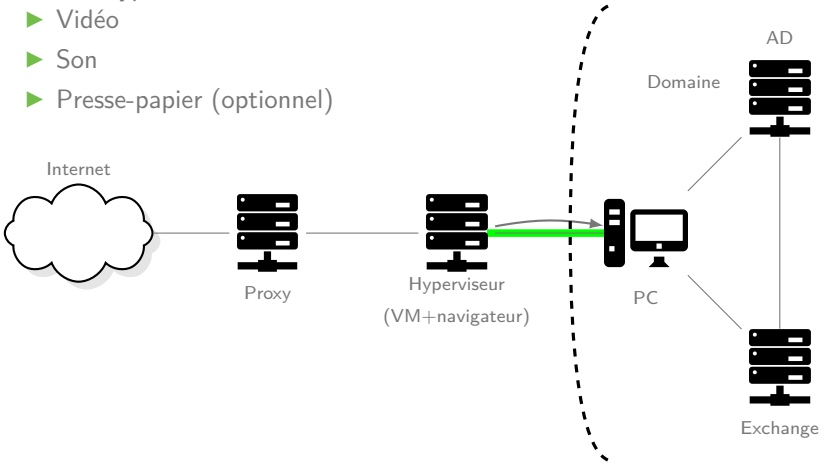
Le flux hyperviseur / postes clients est maîtrisé



Accès à internet depuis les postes du domaine

Données hyperviseur → client :

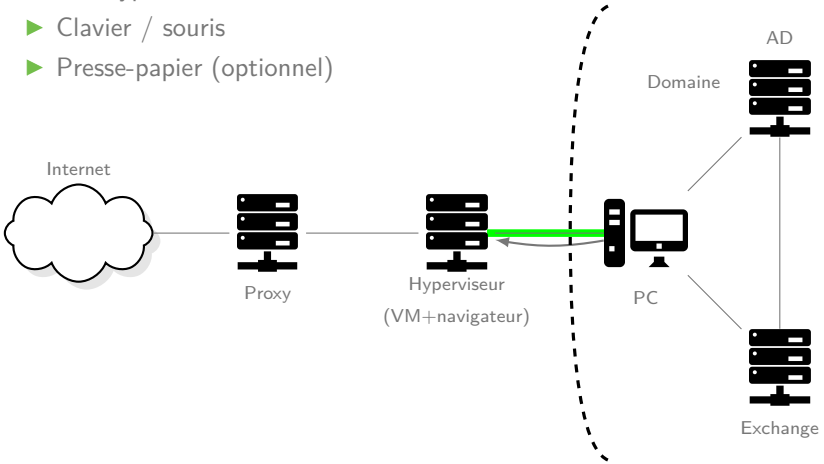
- Vidéo
- Son
- Presse-papier (optionnel)



Accès à internet depuis les postes du domaine

Données hyperviseur ← client :

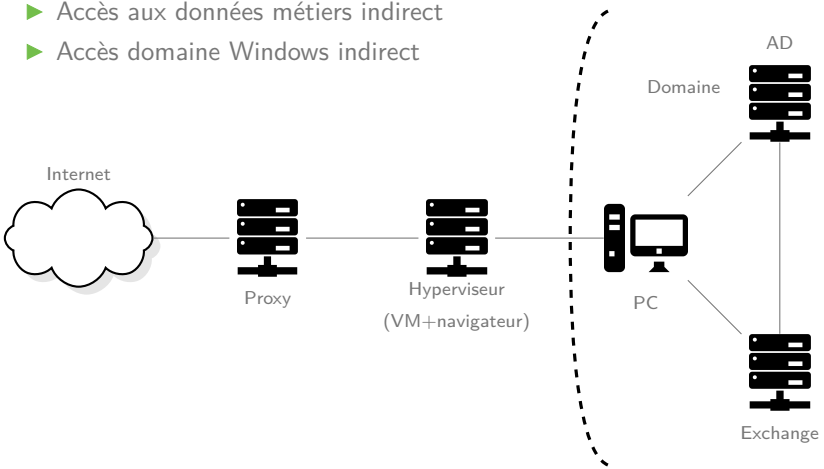
- Clavier / souris
- Presse-papier (optionnel)



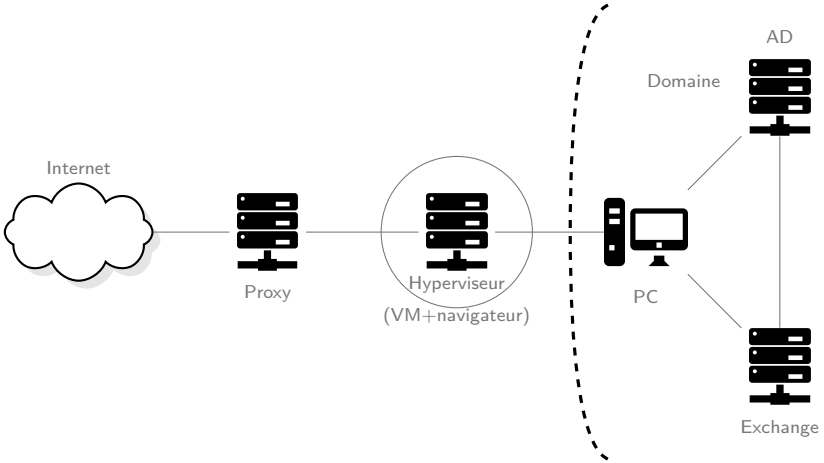
Accès à internet depuis les postes du domaine

Cloisonnement :

- Accès aux données métiers indirect
- Accès domaine Windows indirect



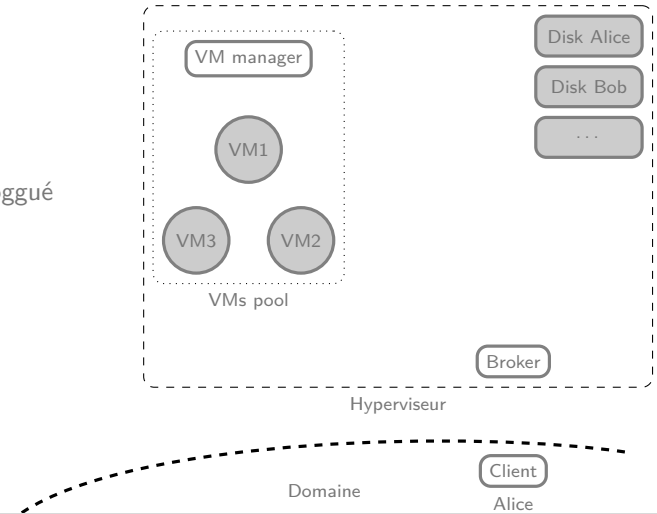
Accès à internet depuis les postes du domaine



Détail du lancement d'un client

L'hyperviseur au repos :

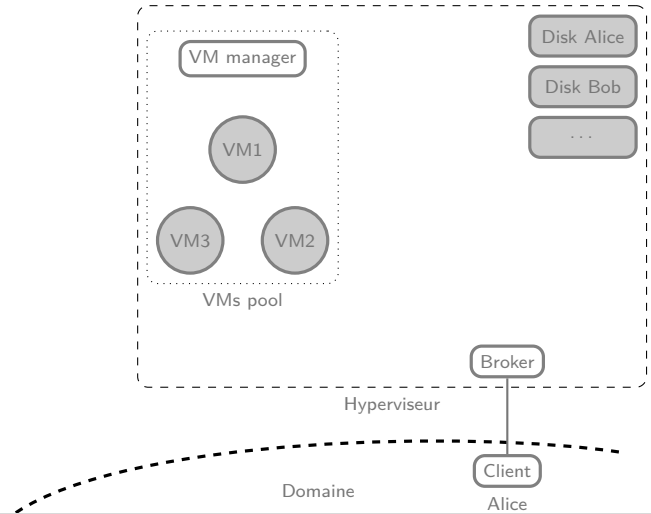
- VMs sont lancées, utilisateur loggué
- Disques des profils



Détail du lancement d'un client

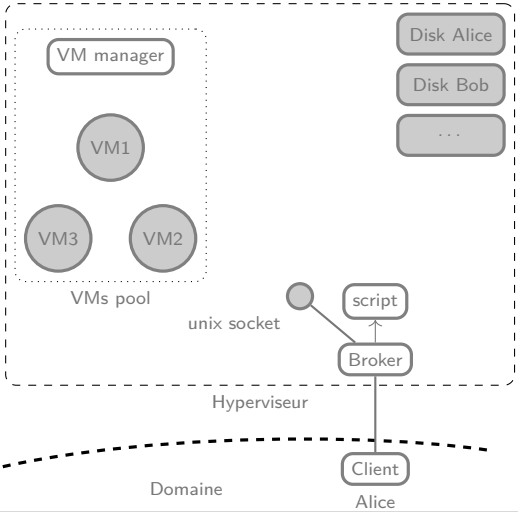
L'utilisatrice Alice se connecte :

- TLS
- Kerberos (sans mdp)



Détail du lancement d'un client

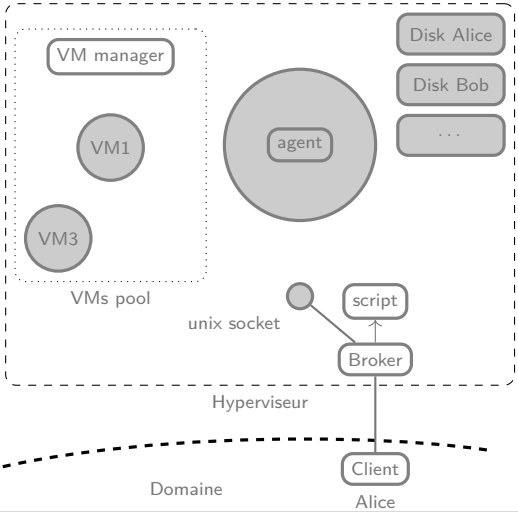
- Le broker lance un script
- Poursuit la connection



Détail du lancement d'un client

Le script :

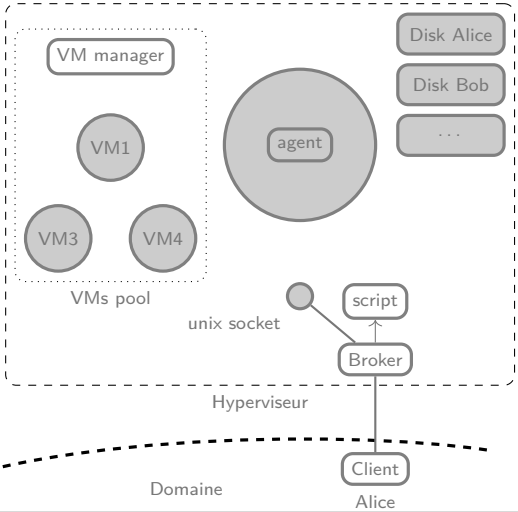
- Prend une VM libre
- L'agent est également lancé



Détail du lancement d'un client

VM Manager :

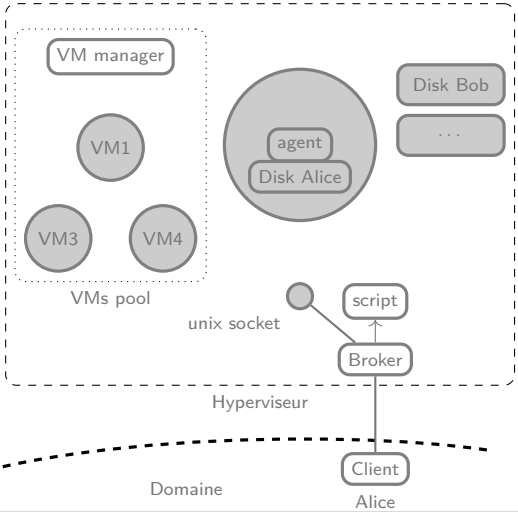
- Relance une VM



Détail du lancement d'un client

Le disque profil est monté

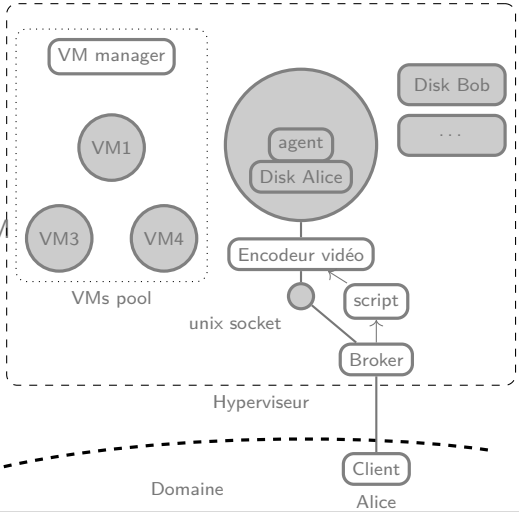
- Unique par utilisateur
- Contient le profil du navigateur



Détail du lancement d'un client

Lancement de l'encodeur vidéo :

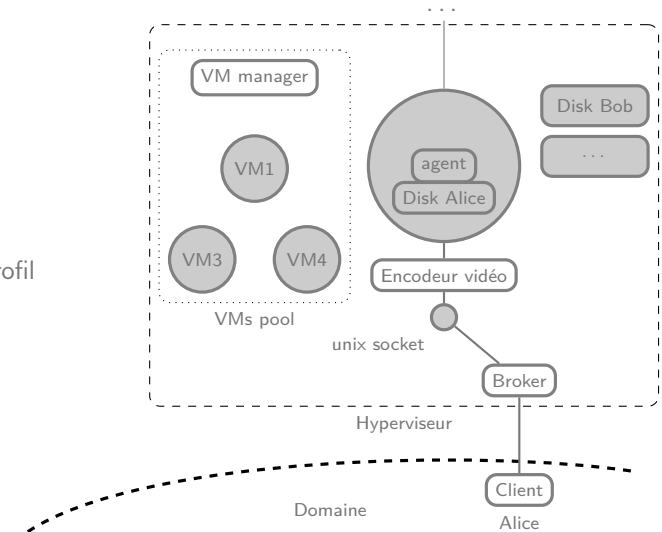
- Encodage sur le host
- Cartes graphiques non partagées avec la VM



Détail du lancement d'un client

La chaîne est prête

- La VM accède à internet
- Utilise ses crédentials
- Reprise des session grâce au profil

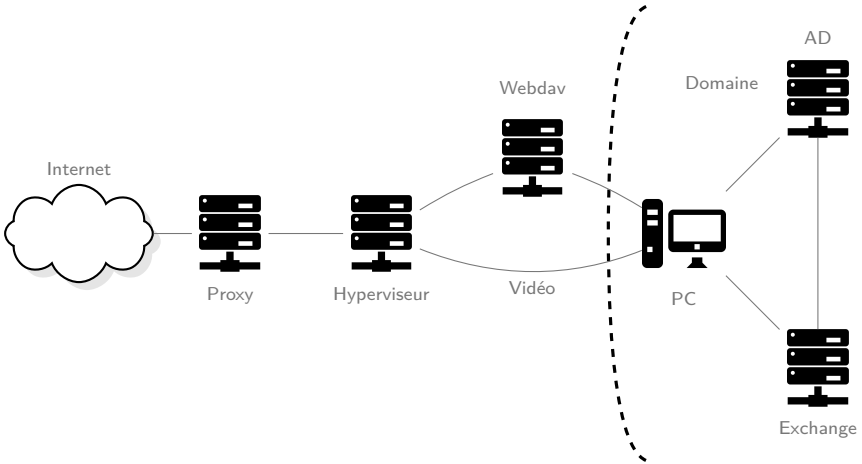


Navigation à distance

Services additionnels

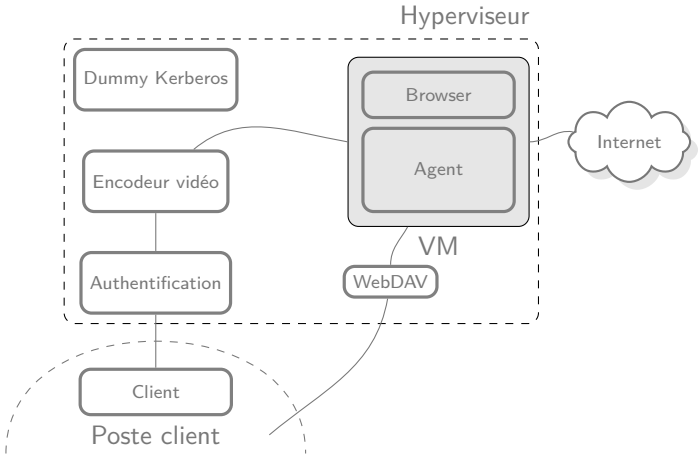
- Téléchargement des fichiers
  → Ajout d'un serveur WebDAV (HTTP) entre le client et l'hyperviseur
  - Natif sous Windows
  - Activé par défaut
  - Authentifié en kerberos (pas de mdp supplémentaire)
- Impression
  - Imprimer dans un fichier / récupération du pdf
  - Ajout d'un serveur d'impression local, envoi sur le share
  - Les imprimantes du domaines ne sont pas exposées aux VMs
- Visioconférence
  → Envoi du son micro ? (à débattre)

Accès à internet depuis les postes du domaine



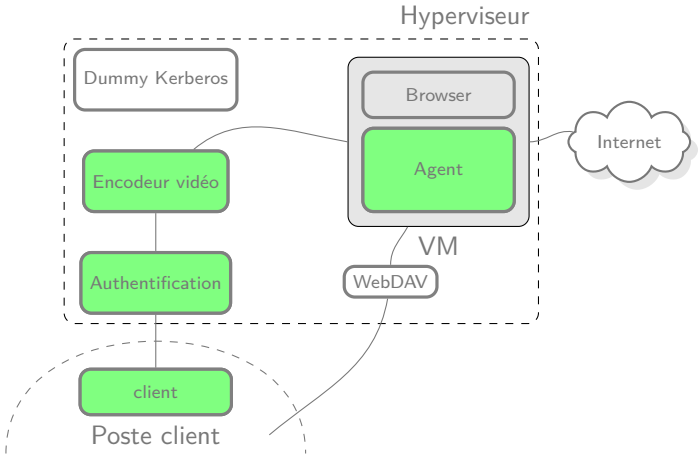
Détail de l'hyperviseur

Détail de l'hyperviseur

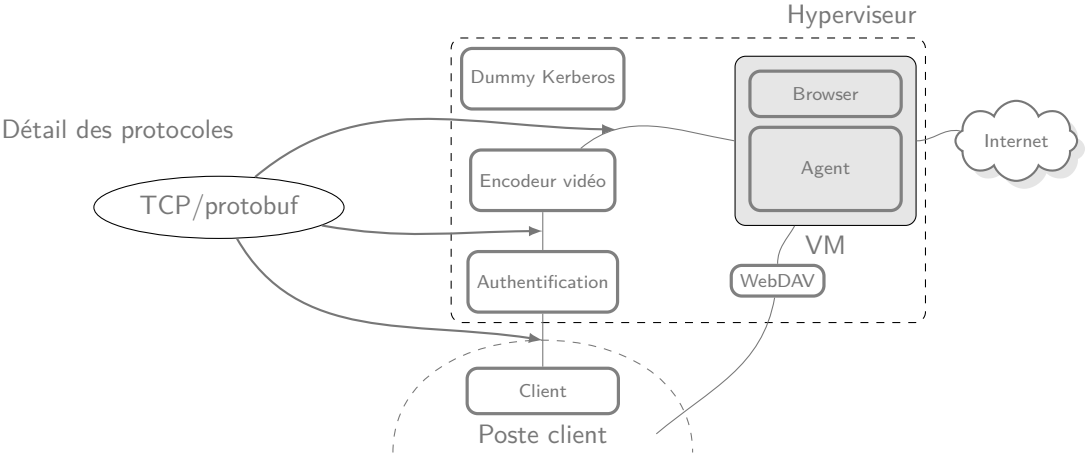


Détail de l'hyperviseur

Modules du projet

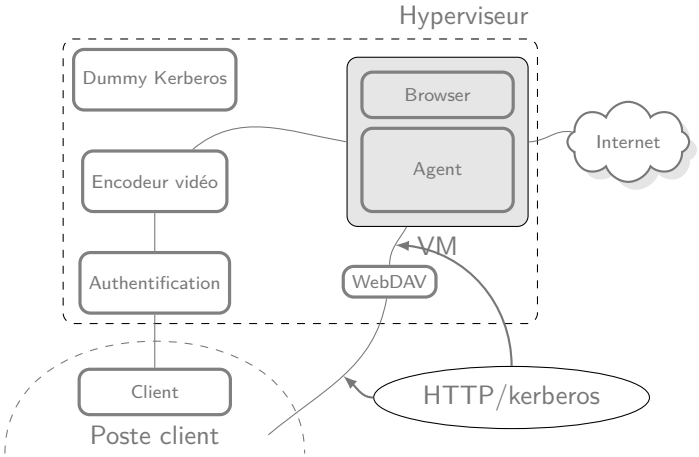


Détail de l'hyperviseur



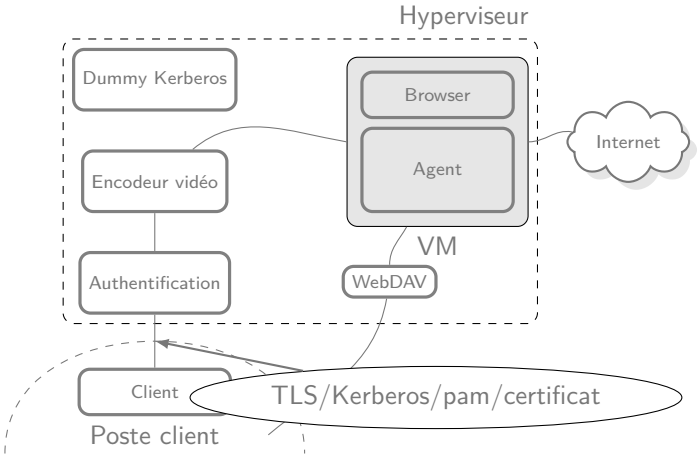
Détail de l'hyperviseur

Détail des protocoles



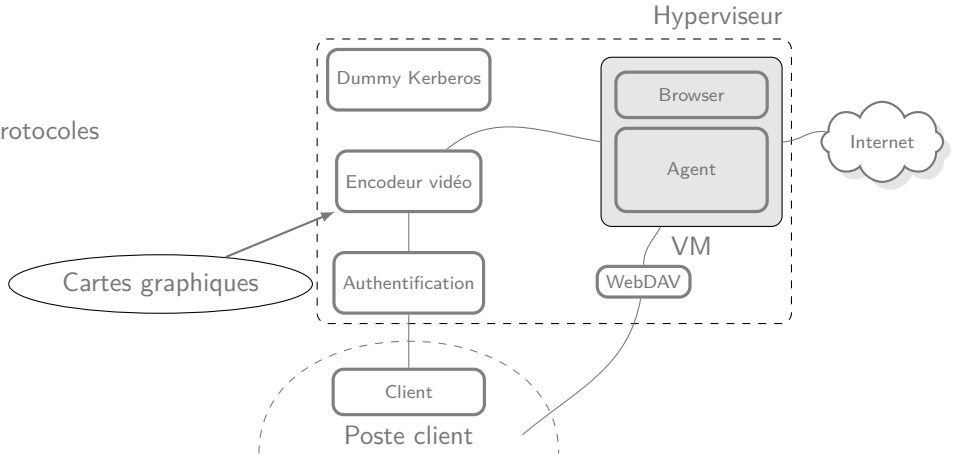
Détail de l'hyperviseur

Détail des protocoles



Détail de l'hyperviseur

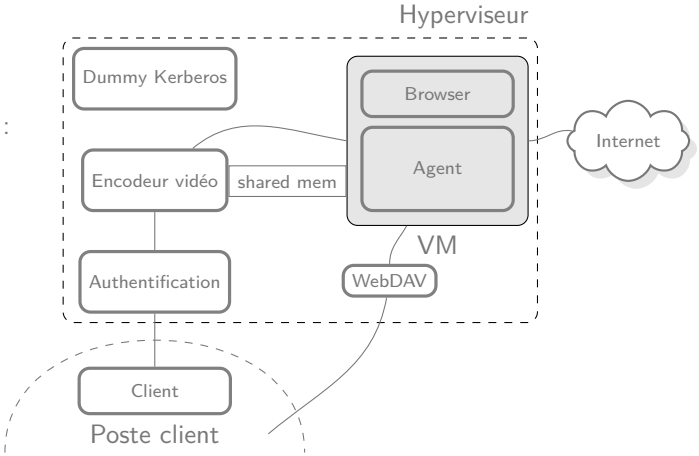
Détail des protocoles



Détail de l'hyperviseur

Mémoire partagée pour les images :

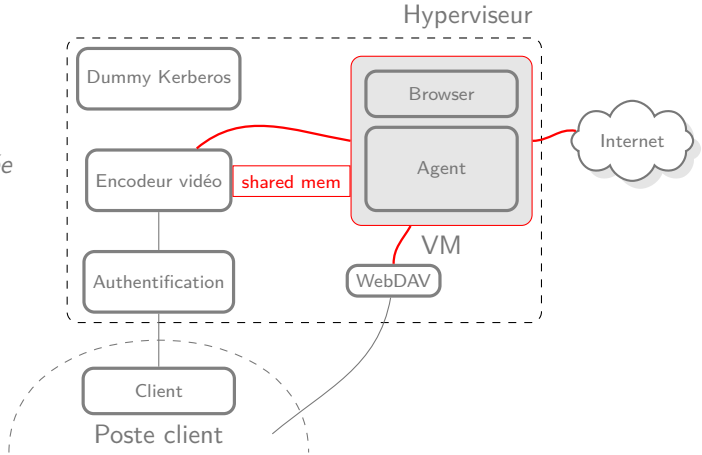
- optimisation
- host : /dev/shm/xxx
- VM : mémoire pci externe
- pixels bruts



Détail de l'hyperviseur

Surface d'attaque ajoutée

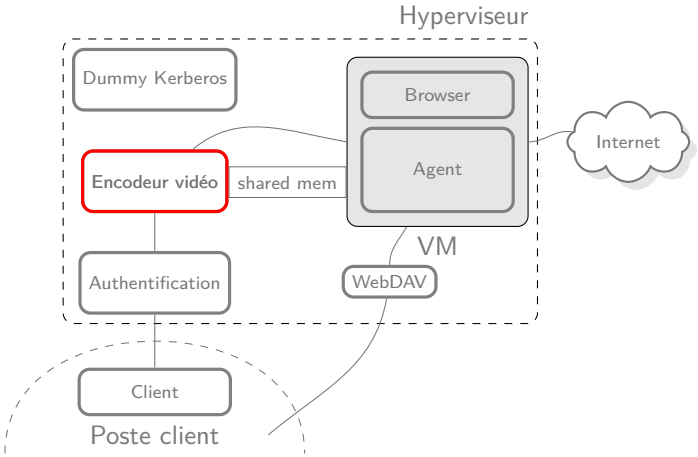
- mémoire partagée : *non parsée*
- protobuf
- HTTP
- Qemu/kvm



Détail de l'hyperviseur

Point particulier

- Encodage fait hors de l'hyperviseur
- Le décodeur vidéo du client n'est pas exposé



Contrainte

Mémoire partagée

- Exfiltration des images : 1920*1080*3*30 $\sim$ 200Mo/s
- La Vm et l'encodeur vidéo ont une certaine affinité
- Échange de pixels *bruts* (non parsés) entre le host et la VM

Airbags latéraux . . .

Propriétés de sécurité de l'hyperviseur

- L'OS du navigateur est contrôlé et personnalisé (SELinux, Grsec, AppArmor ?)
- Le navigateur peut être également personnalisé (Log des URLs, extensions, . . .)
- L'hyperviseur est hors domaine (uniquement une keytab)
- Le flux hyperviseur/poste utilisateur emploie un protocole maîtrisé
- Séquestre des fichiers envoyés/reçus depuis internet
- Passerelle antivirus
- Rebuild quotidien de l'image de la VM
- Disque layerfs conservé pour analyse post mortem

. . . Intérieur cuir . . .

Propriétés de confort de l'hyperviseur

- La compression vidéo se fait par plusieurs cartes graphiques
- Le son
- SSO (kerberos/crédentials utilisateur)
- Copié/collé (pouvant être unidirectionnel)
- Adaptation de la résolution d'écran
- Seamless
- Travail sur la latence / qualité de l'image YUV444
- Notifications navigateur native (peut être désactivé)

. . . 12 Cylindres en V

Ordre de grandeur

- Une Nvidia TeslaT4 encode autour de 20 flux FullHD 25fps
- $\rightarrow$ Test des 120 flux concluant
- 1 flux vidéo $\rightarrow$ $\sim$300Ko/s $\rightarrow$ 500 clients qui “bougent” $\rightarrow$ $\sim$150Mo/s

Performances

- Encodeur vidéo coupé quand l’image est fixe
- L’encodeur vidéo migré à chaud
- 80 fps FullHD sur cpu de laptop
- 4K, mais fps bas

I'm Too Young to Die

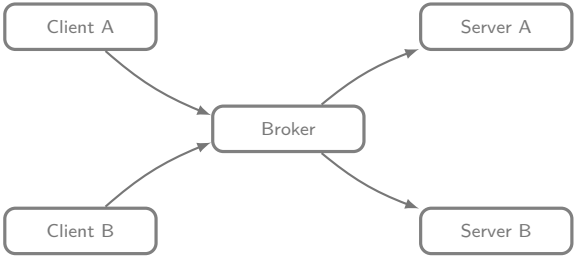
Simple déport vidéo

- *À la maison*
- Configuration simple
- Détachement des sessions à la screen/tmux
- Sur cpu Intel (QSV yuv420)

Hurt Me Plenty

Service avec broker

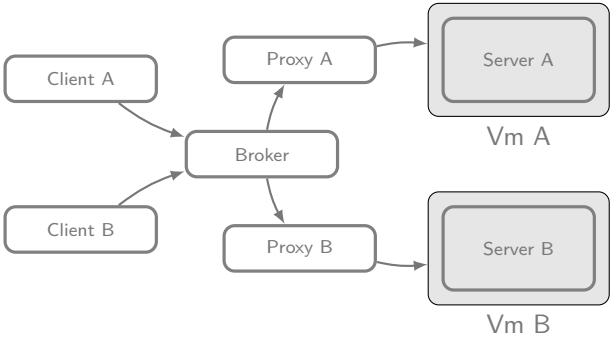
- Multi utilisateur
- *Admin déporté*



Nightmare

Cloisonnement par VM

- Multi utilisateur
- Cloisonnement des clients
- Surface d'attaque réduite



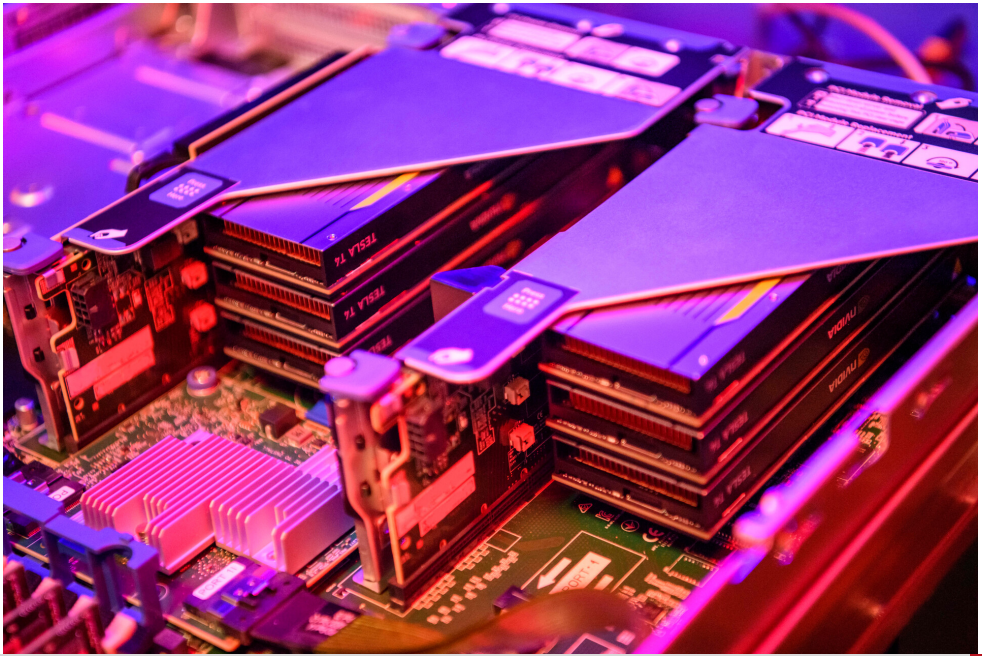
Les chiffres

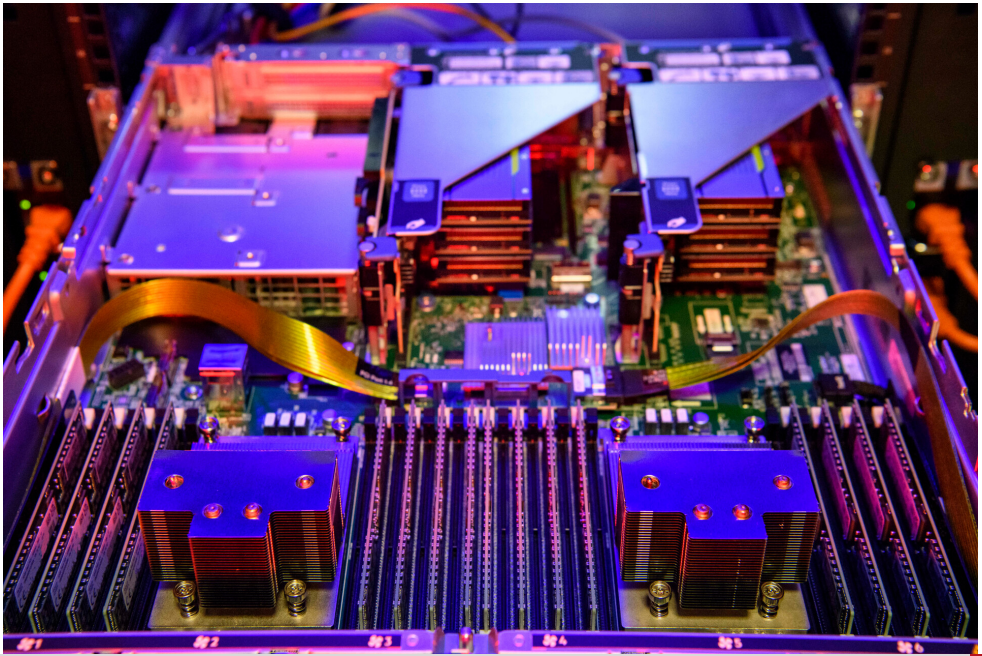
Prix

- DELL 2*AMD Epyc 32 coeurs + 6*TeslaT4 512Go ram $\rightarrow$ 25K€
- En pratique, 3 TeslaT4 pour $\sim$120 clients ?
- 4 machines $\sim$80k€pour $\sim$500 clients web simultanés ?

Mise à jour

- Ampere A2 (équivalent T4) : 1300€ht
- Ampere A16 (équivalent 4 * Tesla T4) (Q3 2021) : 3100€ht


TESLA T4
TESLA T4
NVIDIA

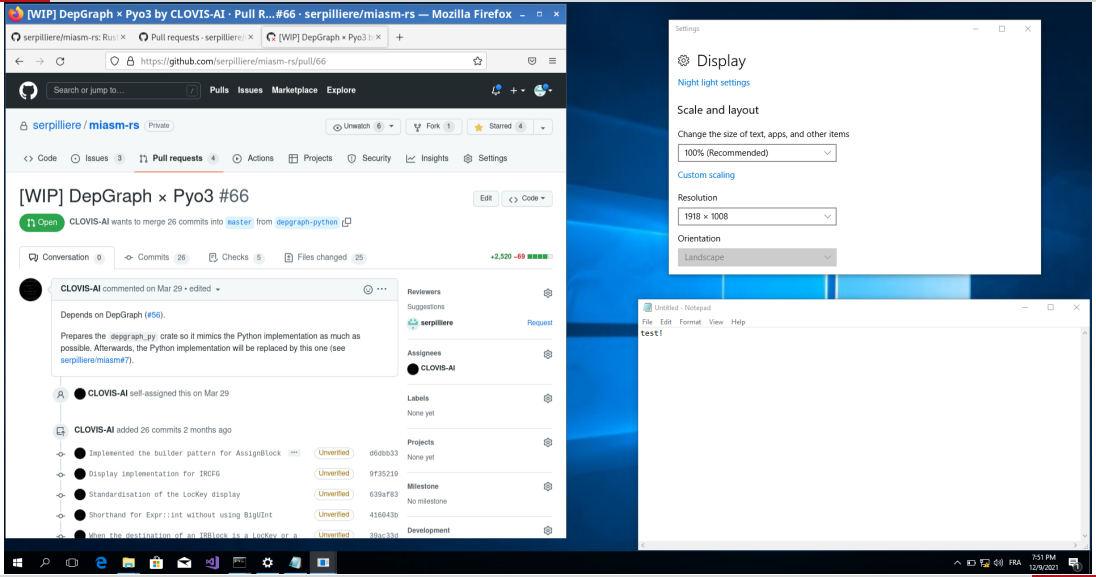


Retours

En test

- Utilisé en admin à distance (sans VM, à la Rdp, sur vpn sur ssh)
- Depuis juillet/août 2021
- Utilisé en navigation à distance (avec VM)
- Depuis septembre 2021

Seamless


[WIP] DepGraph × Pyo3 by CLOVIS-AI · Pull R...#66 · serpilliere/miasm-rs — Mozilla Firefox
https://github.com/serpilliere/miasm-rs/pull/66
Search or jump to...
Pulls Issues Marketplace Explore
serpilliere / miasm-rs Private
Unwatch 6 Fork 1 Starred 4
Code Issues 3 Pull requests 4 Actions Projects Security Insights Settings
[WIP] DepGraph × Pyo3 #66
Edit Code
Open CLOVIS-AI wants to merge 26 commits into master from depgraph-python
Conversation 0 Commits 26 Checks 5 Files changed 25
+2,520 −69
CLOVIS-AI commented on Mar 29 • edited
Depends on DepGraph (#56).
Prepares the depgraph_py crate so it mimics the Python implementation as much as possible. Afterwards, the Python implementation will be replaced by this one (see serpilliere/miasm#7).
CLOVIS-AI self-assigned this on Mar 29
CLOVIS-AI added 26 commits 2 months ago
Implemented the builder pattern for AssignBlock Unverified d6dbb33
Display implementation for IRCFG Unverified 9f35210
Standardisation of the LocKey display Unverified 639af83
Shorthand for Expr::int without using BigUInt Unverified 416043b
Reviewers
Suggestions
serpilliere Request
Assignees
CLOVIS-AI
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
Settings
Display
Night light settings
Scale and layout
Change the size of text, apps, and other items
100% (Recommended)
Custom scaling
Resolution
1918 × 1008
Orientation
Landscape
Untitled - Notepad
File Edit Format View Help
test!
FRA
7:51 PM
12/9/2021

YUV420 vs YUV444

Les aventuriers du pixel perfect

En YUV420, 4 pixels partagent la même information de couleur.

- Gain de place
- Cas pathologiques sur des textes colorés sur fonds colorés

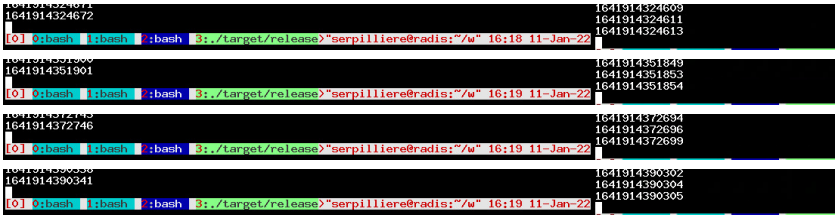
Latence de l'image

Mesure de la latence "porte à porte" (ici, en local, 60fps)

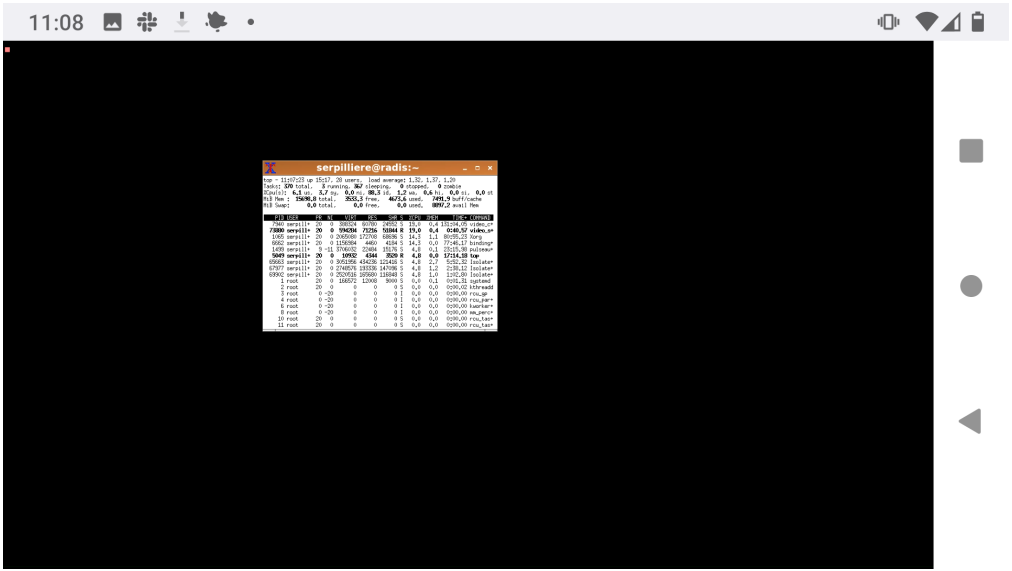
Capture et affichage sur le même display :

```
while true; do echo $(($(date +%s%N)/1000000)); done
```

59ms, 47ms, 47ms, 36ms → ∼48ms en moyenne



Client léger : version Webrtc (à la guacamole)



Déport graphique

Open Source

- Le projet a 1 an et 5 mois
- Environ 12k lignes de code
- `https://github.com/cea-sec/sanzu`

Futur

- Utilisation en remplacement de Rdesktop (Windows / Windows)
- Webrtc => Intégré au navigateur
- Sur téléphone

Démo du déport vidéo

Surf isolé dans un container docker

- Image docker avec sanzu_broker et sanzu_server
- Image docker du client X11
- Firefox est lancé automatiquement et isolé de l'OS

Merci de votre attention.

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Bruyères-le-Châtel | 91297 Arpajon Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 40 00
Établissement public à caractère industriel et commercial
RCS Paris B 775 685 019

Répartition temporelle

```
...
[client] Total: 21.0ms  events: 16.0µs  send: 63.7µs  recv: 9.8ms   decode msg: 417.0ns (dec 2.7ms   yuv 1.6ms   parse 11.9µs  set 3.9ms)
[client] Total: 12.8ms  events: 8.5µs   send: 66.8µs  recv: 4.4ms   decode msg: 1.1µs   (dec 4.6ms   yuv 1.7ms   parse 34.3µs  set 1.9ms)
[client] Total: 17.3ms  events: 10.0µs  send: 65.7µs  recv: 10.2ms  decode msg: 395.0ns (dec 2.6ms   yuv 1.8ms   parse 18.5µs  set 2.5ms)
[client] Total: 14.2ms  events: 28.2µs  send: 87.0µs  recv: 5.3ms   decode msg: 1.6µs   (dec 4.3ms   yuv 2.0ms   parse 27.7µs  set 2.5ms)
[client] Total: 16.5ms  events: 27.7µs  send: 81.6µs  recv: 9.8ms   decode msg: 386.0ns (dec 2.3ms   yuv 1.7ms   parse 12.9µs  set 2.5ms)
[client] Total: 16.4ms  events: 20.1µs  send: 65.1µs  recv: 10.1ms  decode msg: 445.0ns (dec 2.7ms   yuv 1.7ms   parse 12.9µs  set 1.8ms)
[client] Total: 16.4ms  events: 17.4µs  send: 50.8µs  recv: 10.3ms  decode msg: 372.0ns (dec 2.5ms   yuv 1.7ms   parse 17.9µs  set 1.8ms)
[client] Total: 20.1ms  events: 9.4µs   send: 62.0µs  recv: 11.7ms  decode msg: 387.0ns (dec 3.6ms   yuv 2.6ms   parse 13.0µs  set 2.2ms)
[client] Total: 18.6ms  events: 19.0µs  send: 55.4µs  recv: 9.6ms   decode msg: 405.0ns (dec 2.6ms   yuv 3.2ms   parse 13.2µs  set 3.2ms)
[client] Total: 13.3ms  events: 12.0µs  send: 69.3µs  recv: 3.1ms   decode msg: 905.0ns (dec 5.7ms   yuv 2.4ms   parse 44.1µs  set 2.0ms)

...

[server] Frame time: 16 Total: 16.1ms grab: 3.2ms event: 62.6µs encode: 12.8ms (yuv 4.0ms enc 8.8ms ) sound: 327.0ns send: 45.6µs  recv: 10.1µs
[server] Frame time: 16 Total: 11.9ms grab: 1.4ms event: 62.2µs encode: 10.3ms (yuv 3.1ms enc 7.2ms ) sound: 504.0ns send: 89.7µs  recv: 17.0µs
[server] Frame time: 16 Total: 14.0ms grab: 1.6ms event: 49.2µs encode: 12.3ms (yuv 2.9ms enc 9.5ms ) sound: 161.0ns send: 39.1µs  recv: 9.5µs
[server] Frame time: 16 Total: 14.1ms grab: 1.5ms event: 47.0µs encode: 12.5ms (yuv 3.2ms enc 9.4ms ) sound: 173.0ns send: 40.2µs  recv: 10.9µs
[server] Frame time: 16 Total: 14.1ms grab: 1.5ms event: 53.3µs encode: 12.4ms (yuv 3.1ms enc 9.4ms ) sound: 173.0ns send: 28.6µs  recv: 9.6µs
[server] Frame time: 16 Total: 15.3ms grab: 1.5ms event: 47.2µs encode: 13.7ms (yuv 3.1ms enc 10.6ms) sound: 224.0ns send: 39.0µs  recv: 10.1µs
[server] Frame time: 16 Total: 16.6ms grab: 1.9ms event: 1.6ms  encode: 13.1ms (yuv 4.3ms enc 8.8ms ) sound: 170.0ns send: 39.3µs  recv: 11.3µs
[server] Frame time: 16 Total: 12.1ms grab: 1.4ms event: 85.1µs encode: 10.6ms (yuv 4.0ms enc 6.6ms ) sound: 142.0ns send: 121.0µs recv: 29.9µs
[server] Frame time: 16 Total: 15.5ms grab: 2.4ms event: 64.6µs encode: 13.0ms (yuv 4.3ms enc 8.7ms ) sound: 114.0ns send: 33.2µs  recv: 11.7µs
[server] Frame time: 16 Total: 11.7ms grab: 1.5ms event: 60.5µs encode: 10.0ms (yuv 4.5ms enc 5.5ms ) sound: 139.0ns send: 56.3µs  recv: 12.4µs
```

Composants

Dépendances

- Utilisation de ffmpeg pour compresser et décompresser
  - API identique pour différents encodeurs supportés
  - Rajoute le support des “pixel format”
  - Ajourd'hui, YUV420, YUV444, NV12
  - Traduction des couleurs YUV $\Longleftrightarrow$ rgb en SSSE3 (2004 ?)
  - libx264/libx265/h264_nvenc/hevc_nvenc/h264_qsv
- GSS-API/SSPI pour kerberos linux/windows
- Client pour Linux (X11/xcb, wayland en cours)
- Client pour Windows (gui natif)
- Serveur X11(xcb)
- Serveur Windows natif (sans création de session)

TCP vs UDP

Choix

- TCP : pas d'artefact vidéo lors des pertes de paquets, mais rejeux
- Lors d'un freeze TCP : perte d'images
- Reprise au retour du réseau
- Idem pour le son

Cycle du rendu : latence