(WEB)?mail pourquoi mon navigateur RAM

# CoRI&IN 2016

TRACIP
Technologies & Services d'Investigation

# Introduction

# Rappels

modes de collecte

# Précisions techniques

résultats obtenus

# Présentation des travaux effectués (exemple de GMAIL)

# Démo

# <INTRODUCTION>


- Nicolas SCHERRMANN
  - Ex Ingénieur LORIA - Ingénieur R&D @ TRACIP
  - nscherrmann@tracip.fr - @nschermi
- Pierre VEUTIN
  - Ingénieur Responsable R&D @ TRACIP
  - pveutin@tracip.fr - @Geeko_forensic

# <SONDAGE>

- Qui fait de l'investigation numérique (HDD) ?
- Qui fait de l'investigation numérique (RAM) ?
- Qui fait le lien entre les 2 ?
- Qui a déjà rencontré des fichiers hiberfil.sys ?
- Quels outils utilisez-vous ?

EnCase / X-Ways / IEF / DFF … xxd ?

# <SUJET DU TALK>

- Montrer les différences significatives obtenues avec 2 méthodologies d'analyse de la RAM
- Récupérer les données provenant d'un webmail actuel (GMail) à l'aide d'outils simples et gratuits
- Objectifs du talk :
  - Introduire un traitement plus « adapté » des dumps de RAM
  - Présenter des types de données ayant de la valeur pour un enquêteur / analyste forensique.

# <RAPPELS>

- Les différents modes / outils de collecte de la RAM
  - le plus « cool » : Cold Boot Attack
  - le plus « direct » : DMA (Inception)
  - le plus « natif » : driver noyau / kernel
    - outils forensiques traditionnels : IEF, X-Ways, FTK Imager, EnCase, …
    - outils libres : (win|osx)?pmem, LiME, …
- le moins « volatile » : fichiers de crash, d'hibernation, de pagination, de swap

# PINGOUIN QUI CHUTE

Taille ~100 Ko

# <UTILISATION D'OUTILS TRADITIONNELS>

Etape 1 : Recherche Google Image : « chute de pingouin » - affichage d'une grande image

Etape 2 : Mise en veille de la machine ou dump -> hiberfil.sys ou dump.dd = 1,11Go

Etape 3 : Analyse du « dump » (Carving des images JPEG)

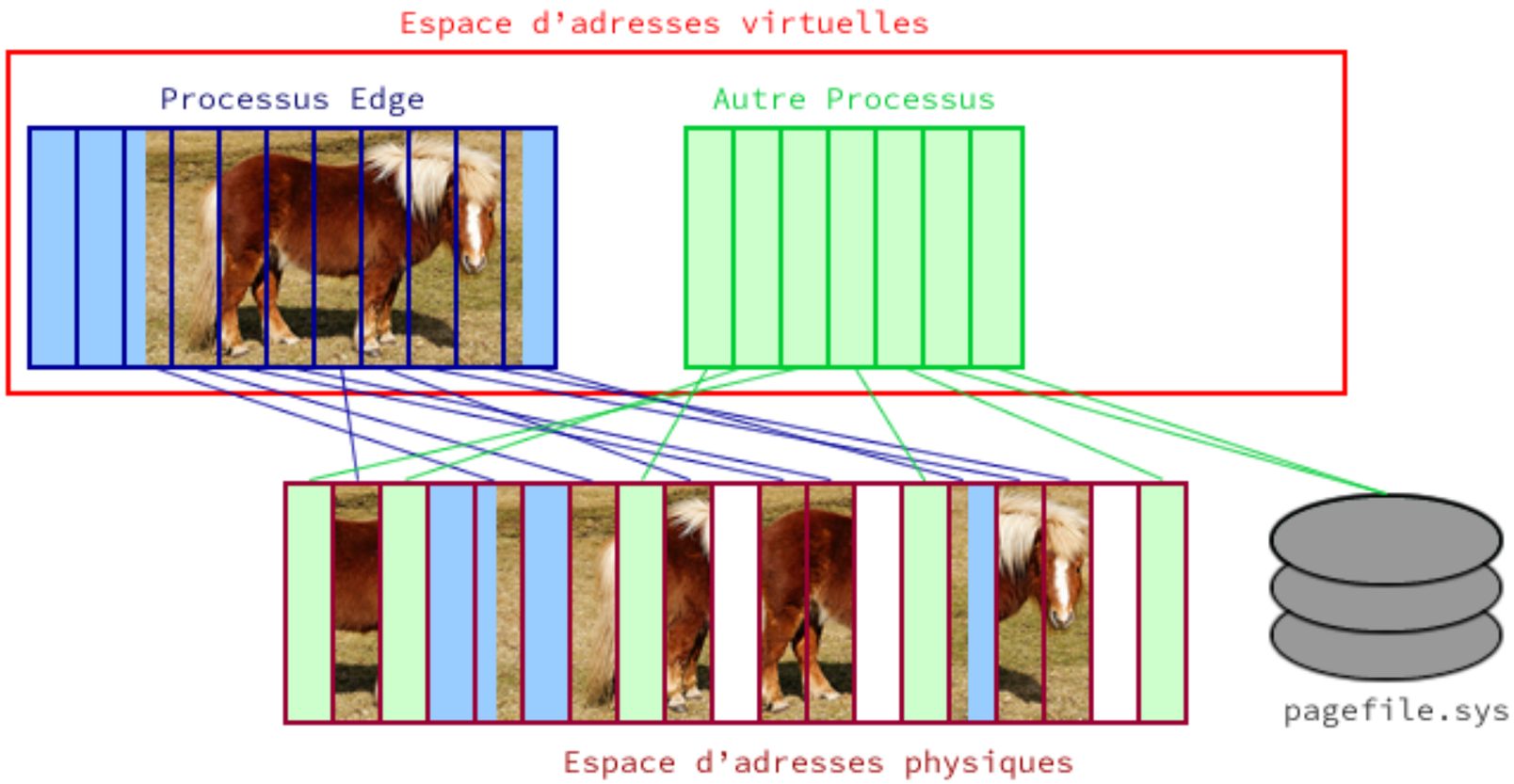

| Outils | # fichiers | Pingouin |
|---|---|---|
| IEF | 469 | *non* |
| EnCase 7 | 478 | *non* |
| X-Ways | 510 | *non* |
| FTK | 519 | *non* |
| APF | 593 | *non* |
| scalpel | 520 | *non* |

# ON PEUT, SUR LES VÉRITÉS DE FAIT, SE PASSER DE LA DÉMONSTRATION SI L'ON SAIT SE SERVIR DE L'EXPÉRIENCE.

ROGER BACON (miam . . .)

# <PRÉCISIONS TECHNIQUES>

## Explication des résultats




- Gestion de pages de 4 Ko - ASLR
- Problème : Outils traditionnels = analyse séquentielle

# <ADAPTATION DE LA MÉTHODE>

- 1 / Reconstruction des plages d'adresses (Utilisation d'outils dédiés )

```
dfir:~/$ vol.py -f dump.dd --profile=Win10x86 pslist


dfir:~/$ vol.py -f dump.dd --profile=Win10x86 memmap -p PID
  Virtual    Physical         Size DumpFileOffset
  ---------- ---------- ---------- --------------
  0x008c0000 0x1c3e8000     0x1000            0x0
  0x008d0000 0x18be5000     0x1000         0x1000
  0x008e0000 0x3e20b000     0x1000         0x2000
                <——truncated——>
  0xffd06000 0x00001000     0x1000     0x1fc35000
  0xffd09000 0x00109000     0x1000     0x1fc36000
  0xffdf0000 0x00398000     0x1000     0x1fc37000
```

# <ADAPTATION DE LA MÉTHODE (SUITE)>

- 2 / Extraction de la plage mémoire dédiée au navigateur (Microsoft Edge)

```
dfir:~/$ vol.py -f dump.dd --profile=Win10x86 memdump -D . (-p PID_Edge)
```

- 3/ Carving (exemple avec scalpel)

```
dfir:~/$ scalpel edge.dmp
```

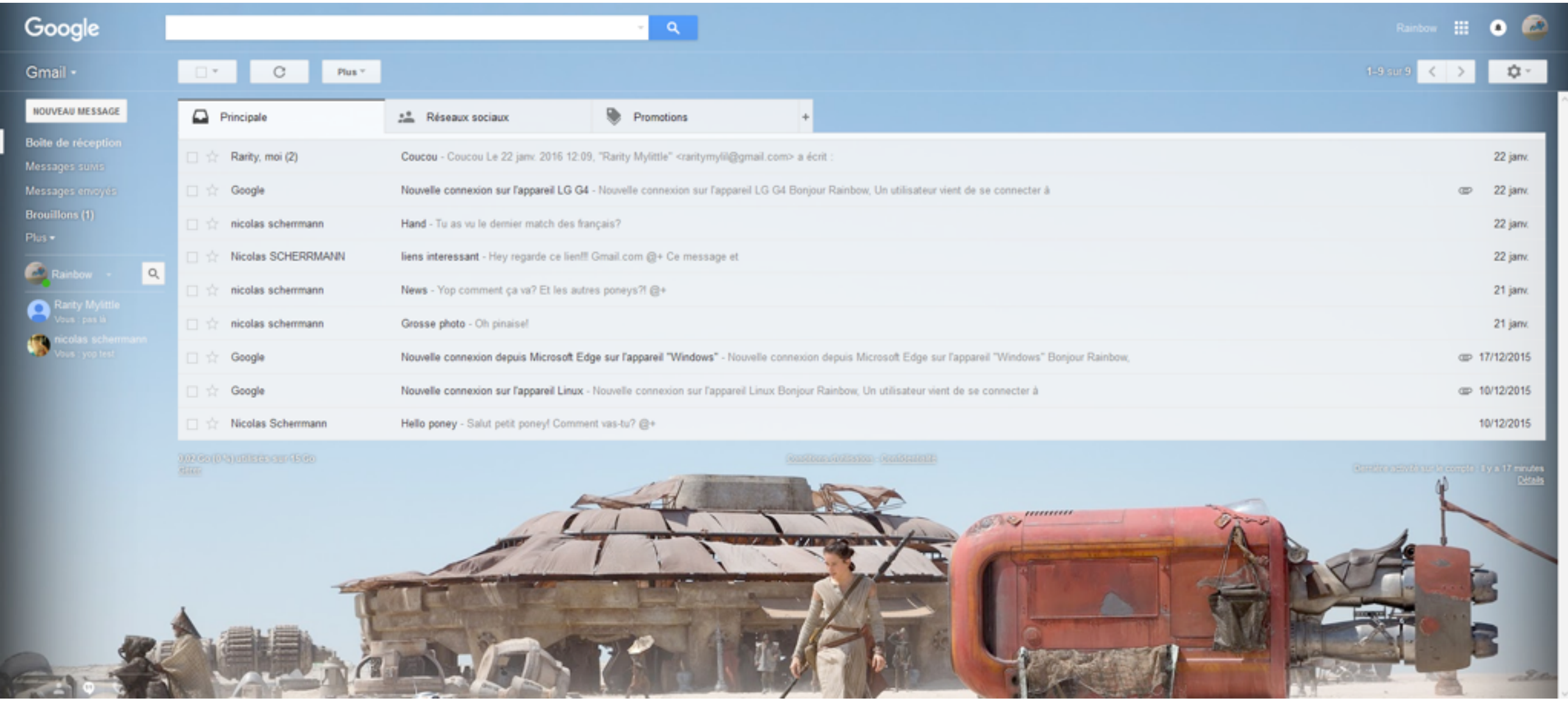
# <EXPLICATIONS>

- Tous les outils retrouvent le fichier recherché
- Le nombre de fichiers récupérés est plus restreint* mais plus qualitatif

| Outils | # fichiers | Pingouin |
|---|---|---|
| IEF | 116 | *oui* |
| EnCase 7 | 119 | *oui* |
| X-Ways | 120 | *oui* |
| FTK | 121 | *oui* |
| APF † | 118 | *oui* |
| scalpel | 119 | *oui* |

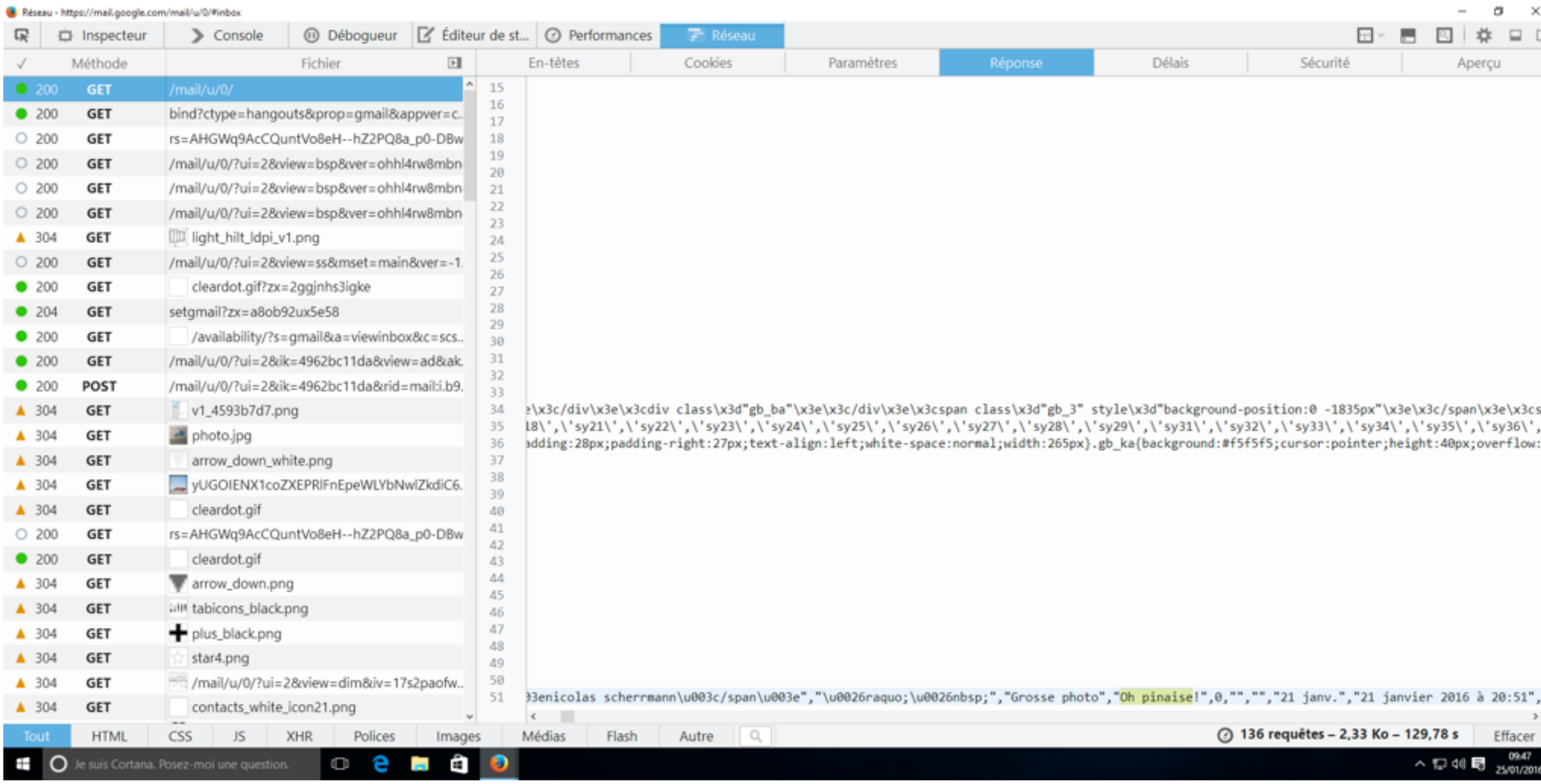
# ANALYSE EN PROFONDEUR

# <EXEMPLE AVEC LE WEBMAIL GMAIL>




... et quelques poneys

# <DERRIÈRE LE NAVIGATEUR>



# <JSON>

- L'analyse réseau de la page d'accueil fait apparaître des structures JSON de ce type :


```
["^all","^i","^iim","^io_im","^io_imc1","^io_lr","^o","^smartlabel_personal"],
[],"\u003cspan class\u003d\"yP\" email\u003d\"nico.scherrmann@gmail.com\" name
\u003d\"nicolas scherrmann\"\u003enicolas scherrmann\u003c/span
\u003e","\u0026raquo;\u0026nbsp;","Grosse photo","Oh pinaise!",0,"","","21
janv.","21 janvier 2016 à 20:51",
```


- Ce qui donne en décodant l'unicode :


```
["^all","^i","^iim","^io_im","^io_imc1","^io_lr","^o","^smartlabel_personal"],
[],"<span class="yP" email="nico.scherrmann@gmail.com" name="nicolas
scherrmann">nicolas scherrmann</span>","&raquo; ","Grosse photo","Oh
pinaise!",0,"","","21 janv.","21 janvier 2016 à 20:51",1453450240294000,,[],,0,
[],,[],,"3",[1],,"nico.scherrmann@gmail.com",,,,0,0]
```

# <A LA RECHERCHE DES STRINGS>

```
 $ strings dump.dd | grep -C2 "smartlabel" | more

--

["^all","^cob-processed-
gmr","^i","^io_unim","^os_social","^smartlabel_social","^sq_ig_i_social","^u","
^unsub"],"\u003cspan class\u003d\"zF\" email\u003d\"notification
+kjdkk3vdw5__@facebookmail.com\" name\u003d\"Facebook\"\u003eFacebook\u003c/
span\u003e","\u003cb\u003e\u0026raquo;\u003c/b\u003e\u0026nbsp;","\u003cb
\u003eWobniar, vous avez 2 nouvelles notifications et 1 message\u003c/b
\u003e","Pas mal de choses se sont passées sur Facebook depuis votre dernière
connexion. Voici quelques",0,"","","\u003cb\u003e21/12/2015\u003c/b\u003e","21
décembre 2015

--
```

# <GARDER LE CONTACT>

- La recherche d'adresses connues permet de retrouver ces contacts et aussi leur structure :


```
[1,"nicolas scherrmann","nicolas","//lh5.googleusercontent.com/-s-w-tth9bMo/AAABAAABAAI/AAAABAAABoM/
pq0JO6u7l5o/photo.jpg",
["nicoscherrmann@gmail.com","nico.scherrmann@gmail.com","nscherrmann@tracip.fr","n.i.c.o.s.c.h.e.r.r.m.a.n.
n@gmail.com"],["06 12 34 56 78"],,,,,1,2,[],[],"nicoscherrmann@gmail.com"
```


- Nous identifions ici :
  - un id
  - le nom complet
  - le nom
  - l'url de l'image de profil
  - une liste d'adresses mails
  - le numero de téléphone
  - l'adresse mail

# <COMMENCER UNE CORRESPONDANCE>

- Pour les mails, nous retrouvons la structure vue précédemment :


```
["^all","^i","^iim","^io_im","^io_imc1","^io_lr","^o","^smartlabel_personal"],[],"\u003cspan class\u003d
\"yP\" email\u003d\"nico.scherrmann@gmail.com\" name\u003d\"nicolas scherrmann\"\u003enicolas scherrmann
\u003c/span\u003e","\u0026raquo;\u0026nbsp;","Grosse photo","Oh pinaise!",0,"","","21 janv.","21 janvier
2016"
```


- Une fois décodée :


```
["^all","^i","^iim","^io_im","^io_imc1","^io_lr","^o","^smartlabel_personal"],[],"<span class="yP"
email="nico.scherrmann@gmail.com" name="nicolas scherrmann">nicolas scherrmann</
span>","&raquo; ","Grosse photo","Oh pinaise!",0,"","","21 janv.","21 janvier 2016"
```


- Nous identifions ici :
  - une structure contenant le tag du message (ici *personal*)
  - l'adresse mail et le nom de l'expéditeur
  - le sujet du mail
  - une partie du message
  - la date du message

# <VALIDATION DES RESULTATS>

- navigateur :


```
["^all","^i","^iim","^io_im","^io_imc1","^io_lr","^o","^smartlabel_personal"],
[],"<span class="yP" email="nico.scherrmann@gmail.com" name="nicolas
scherrmann">nicolas scherrmann</span>","&raquo; ","Grosse photo","Oh
pinaise!",0,"","","21 janv.","21 janvier 2016 à 20:51 »,1453450240294000,,[],,
0,[],,[],,"3",[1],,"nico.scherrmann@gmail.com",,,,0,0]
```


- en RAM :


```
["^all","^i","^iim","^io_im","^io_imc1","^io_lr","^o","^smartlabel_personal"],
[],"<span class="yP" email="nico.scherrmann@gmail.com" name="nicolas
scherrmann">nicolas scherrmann</span>","&raquo; ","Grosse photo","Oh
pinaise!",0,"","","21 janv.","21 janvier 2016
```


il ne manque pas grand chose … dans la regex « à »

# <FAIRE PASSER LE MESSAGE>

- Pour les hangouts, les structures sont les suivantes :

```
[[0,"Comment a va?"]]
```

- Il n'y a donc aucune méta donnée … pour le moment … :-)

# DÉMONSTRATION

# ( ( COMMENTAIRE ) | ( SUGGESTION ) ) ?

rot13(rainbowdash)@gmail.com

# MERCI.

N. Scherrmann
P. VEUTIN