RéSIST

Greylisting : Etat de l'art

RéSIST

Contexte

- Enseignement supérieur
  - Plus particulièrement UT1

RéSIST

Principe

- Tout mail qui arrive est constitué d'un triplet
  - IP de l'expéditeur
  - @ email de l'expéditeur
  - @ email du destinataire

RéSIST

Principe (2)

- Tout mail avec un triplet inconnu est
  - est refusé temporairement (451)
  - son triplet est inséré dans une base en gris.
- Tout mail arrive avec un triplet gris
  - le mail passe
  - le triplet est blanchi
- Tout mail qui arrive avec un triplet blanc
  - le mail passe
  - Le triplet est « mis à jour »

RéSIST

Principe (3) : les délais

- Un triplet ne peut être blanchi avant x minutes (souvent 10-20 minutes)
- Un triplet gris est effacé au bout de x heures (allant de 4h à 25 heures)
- Un triplet blanc est effacé au bout de x jours (allant de 15 jours à 35 jours)

RéSIST

Les possibilités

- Sendmail
  - Milter-greylist http://hcpnet.free.fr/milter-greylist/
  - Relaydelay http://projects.puremagic.com/greylisting/
  - J-Chkmail http://j-chkmail.ensmp.fr/
  - etc.
- PostFix
  - Postgrey http://isg.ee.ethz.ch/tools/postgrey/
  - SQLGrey http://sqlgrey.sourceforge.net/
  - etc.

RéSIST

Caractéristiques utiles

- Travailler avec la classe C au lieu de l'IP (quasi indispensable)
- Travailler avec plusieurs MX (base centrale)
- Notion de whitelist expéditeurs
- Notion de whitelist destinataires
- Affichage du délai dans le message

RéSIST

Outils choisis

- D'abord des développements locaux en perl
- Puis PostGrey
  - http://isg.ee.ethz.ch/tools/postgrey/
- Complétés par d'autres outils
  - Bogofilter
  - Tests (non bloquants) SPF et RBL (RBOWL)

RéSIST

Installation de Postgrey

- Très simple
  - Récupération du script
  - Installation de modules perl
  - Création des scripts de démarrage (non fournis)
  - Ajout d'un utilisateur
  - Création de 2 répertoires

RéSIST

Configuration de Postgrey

- Très simple
  - 3-4 paramètres au lancement du daemon
    - `--inet=0.0.0.0:10023 -d --verbose --group postgrey`
  - une ligne de plus dans le main.cf PostFix
    - `check_policy_service inet:127.0.0.1:10023`
- Pour la phase d ’apprentissage
  - ajout du paramètre
    - `--greylist-action=DUNNO`

RéSIST

Particularités de PostGrey

- Le triplet, par défaut, utilise le réseau IP
- Maintenance automatique
- Une whitelist à jour
- Une whitelist de destinataires
- Mono-processus
- Affiche le délai de chaque mail
- Auto-whitelisting

RéSIST

Paramétrage du GreyListing

- Délai de latence de 10 minutes
- Durée de vie des GREYlistés: 25 heures
- Durée de vie des WHITElistés: 35 jours

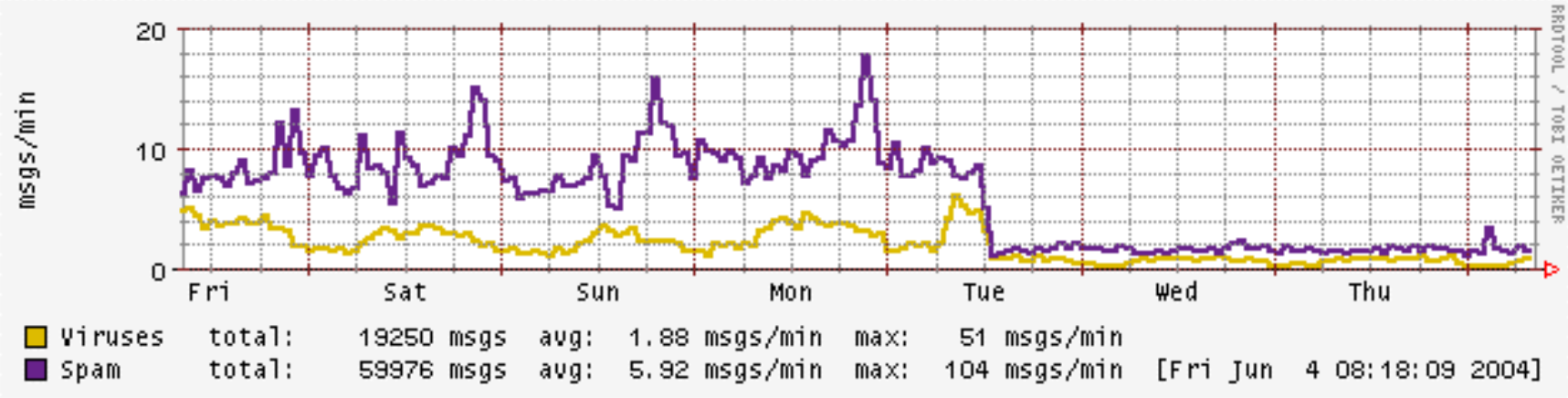
RéSIST

Résultats

RéSIST

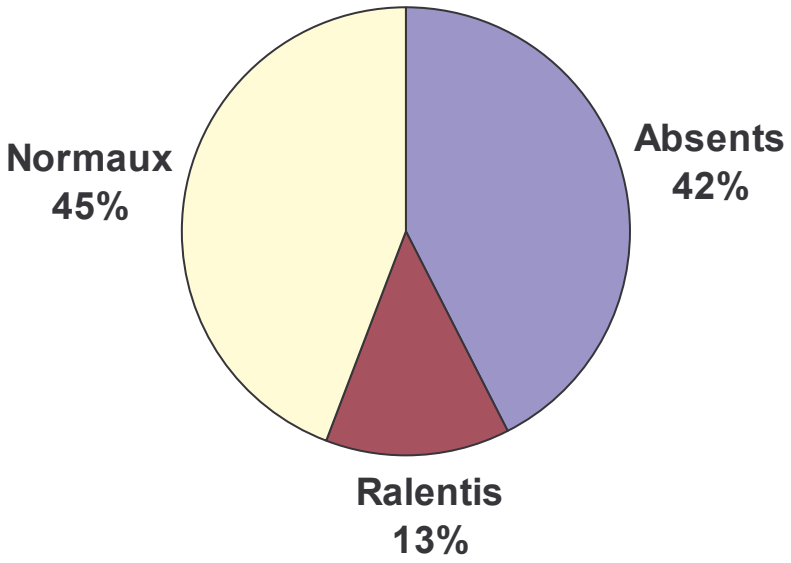
Résultats :

- 80% de Spams en moins
- 75% de Virus en moins



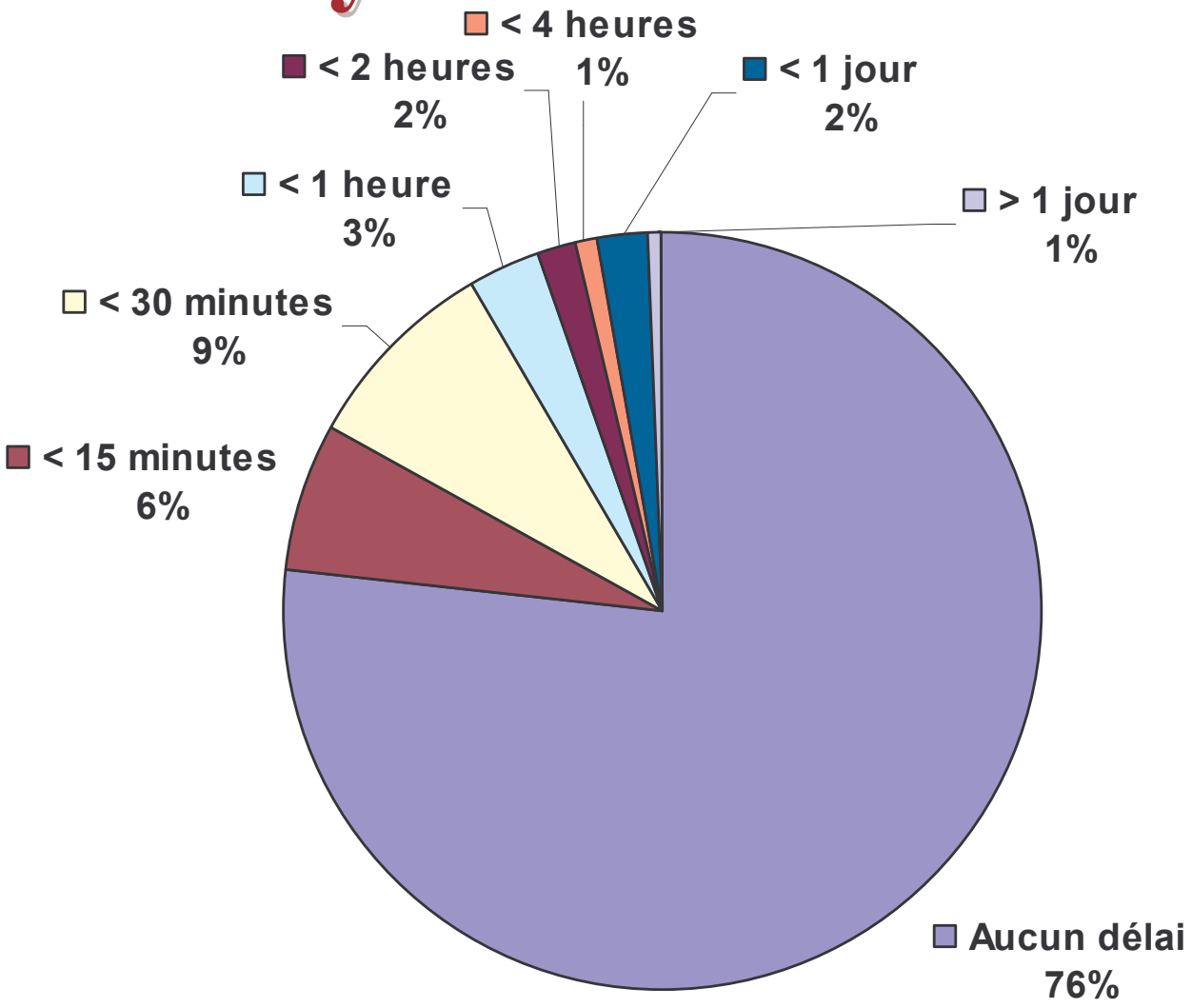
RéSIST

Efficacité du greylisting



RéSIST

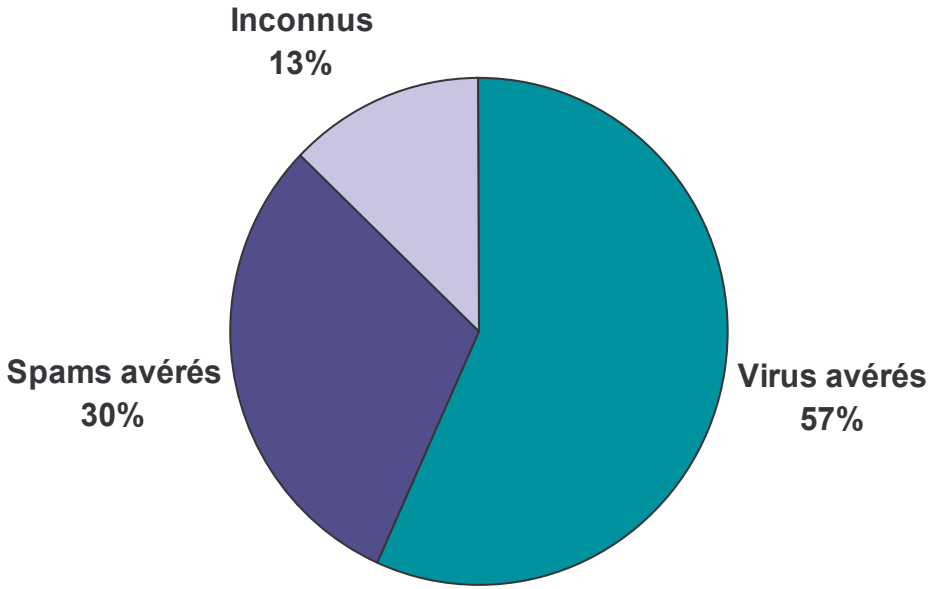
Délai moyen



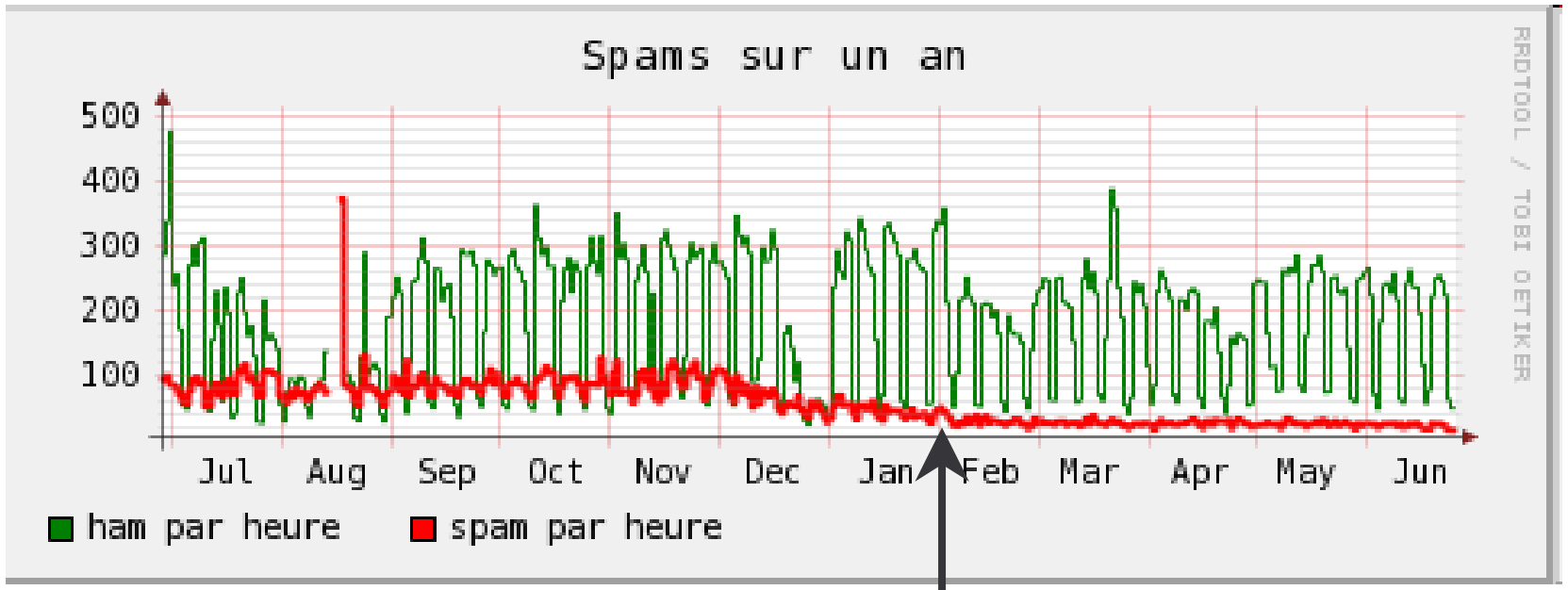
RéSIST

Plus de 1 jour !?!?

- Mais ça va pas non ?
- Ben si :



Taux Résiduel Spams/Viral



Application totale
du greylisting

RéSIST

Et les utilisateurs ?

- Quelques plaintes (5 en plus de 1 an) sur des retards
- Plus quelques habituels
  - Pour eux : « on vous sort du greylisting ? »

RéSIST

A faire

- Une procédure de manipulation de la DB :
  - Enlever les triplets incorrects (tous ceux avec yandex.ru, rick@site.net, etc.)
  - Récupération des IP expéditeurs.
  - etc.

RéSIST

Limiter les inconvénients

- Accélérer la réexpédition des mails (parce que les autres ont du greylisting). ½ heure est un bon compromis
- Placer un enregistrement SPF (utile de toute manière)
- Ne pas « whitelister » des réseaux complets, mais plutôt en conjonction avec SPF

Conclusions

RéSIST

Conclusions

- La méthode la plus efficace contre les spams et les virus
- Elle est très simple
- Les dégâts collatéraux sont faibles
- Mais cela ne dispense pas de
  - Filtrer le port 25
  - Mettre un enregistrement SPF
  - Mettre un bayésien
  - Mettre un MX final avec une faible priorité (note de 95%)