Filtrage de messagerie et Analyse de contenu

SSTIC04 – 03/06/2004

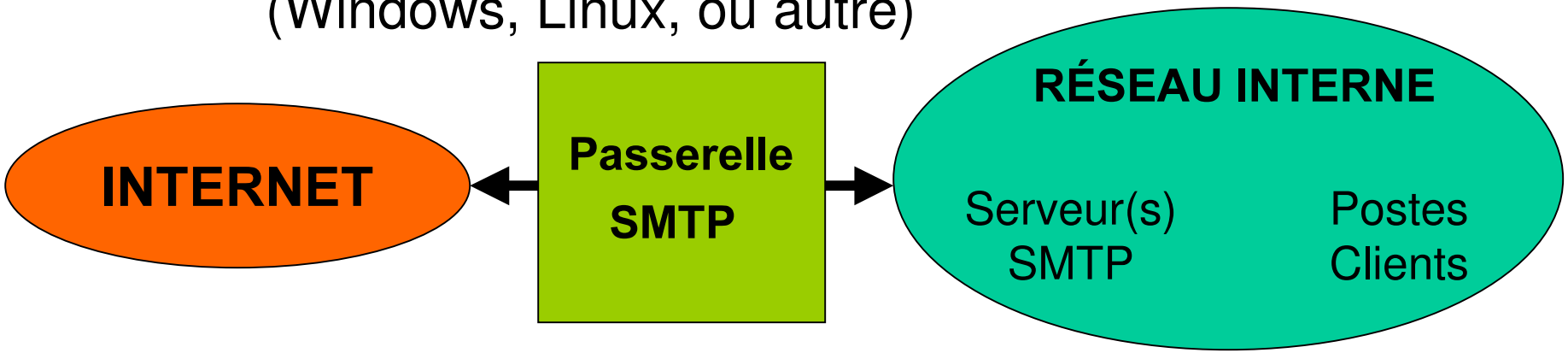
Philippe Lagadec
DGA / CELAR
philippe.lagadec (at) dga.defense.gouv.fr

Sommaire

1. **Pourquoi filtrer la messagerie SMTP ?**
   - Les risques
2. **Comment filtrer ?**
   - Les objectifs
   - Les techniques d'analyse de contenu
3. **Analyse de contenu**
   - Limites et faiblesses constatées
   - Pistes d'amélioration

Hypothèses

- Un réseau interne constitué essentiellement de **postes clients et de serveurs Windows**.
  - (Avec d'autres OS la messagerie serait moins amusante… :-)
- Interconnexion directe à **Internet** ou via une passerelle.
- Echange de messagerie SMTP avec Internet, grâce à 1 ou plusieurs **serveurs SMTP** (Windows, Linux, ou autre)



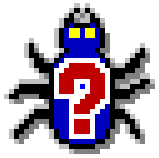
1) Pourquoi filtrer la messagerie ?

Les RISQUES

Les risques liés à la messagerie

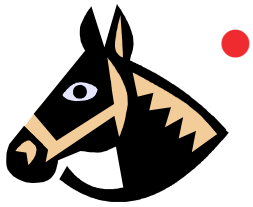
- Nombreux risques de niveau applicatif:
  - Virus et vers
  - Pièces jointes malveillantes (Chevaux de Troie)
  - Code malveillant dans le corps HTML
  - Spam (messages non sollicités)
  - Usurpation d'identité
  - Relayage
  - Vulnérabilités des clients et serveurs
  - Mailbombing
  - Web bugs
  - …

Code malveillant: formes diverses



- **Virus:**
  - Code malveillant qui s'attache à d'autres programmes pour se répliquer. Nécessite généralement **l'action de l'utilisateur**. Attaque en général non ciblée, lente.

- **Ver:**
  - Code malveillant qui se réplique **tout seul** par le réseau ou la messagerie. Attaque en général non ciblée, rapide.
  - Aujourd'hui la plupart des « virus » sont aussi des vers.



- **Cheval de Troie:**
  - Logiciel ou fichier apparemment inoffensif qui exécute des actions malveillantes à l'insu de l'utilisateur.
  - Peut être envoyé par mail, sur CDROM ou disquette, placé sur un site web, …
  - Attaques généralement ciblées

- **Bombe logique, Spyware, etc…**

Pièces jointes malveillantes

- **Problème principal:** un message peut comporter des **pièces jointes**, avec un **contenu malveillant.**
  - → Virus, Vers, Chevaux de Troie, attaques ciblées, …

- **Nombreuses formes possibles (Windows):**
  - **Fichiers exécutables:** exe, com, bat, cmd, scr, pif, lnk, vbs, vbe, js, jse, …
  - **Documents** avec code actif: html, xml, doc, xls, ppt, mdb, pdf, …
  - **Conteneurs**: zip, cab, msi, tar.gz, rar, arj, lzh, rtf, …
  - Cf. archives SSTIC03: présentation « Formats de fichiers et code malveillant »
    - http://www.sstic.org/SSTIC03/interventions03.shtml
    - ou version mise à jour pour l'OSSIR: http://www.ossir.org/windows/supports/liste-windows-2003.shtml

Rappel: formats de fichiers et code malveillant (cf. archives SSTIC03)

Format	Extension(s)	Risque	Conteneur	Type
Exécutable binaire	EXE, SCR	1	X	Binaire
Exécutable binaire	COM	1	X	Binaire/Texte
Fichier batch	BAT, CMD	1		Texte
Scripts WSH	VBS, JS, VBE, JSE, …	1		Texte
Base Access	MDB	2	X	Binaire
HTML, XML	HTM, HTML, XML, …	2 ou 3 (*)		Texte
Document Word	DOC, DOT, WBK, …	2 ou 3 (**)	X	Binaire
Document PDF	PDF	3		Texte
Document RTF	RTF	4	X	Texte
Image Bitmap	GIF, JPEG, PNG, BMP	5		Binaire

(*) : suivant la version d'Internet Explorer installée et les correctifs appliqués.
(**) : suivant la version d'Office installée et les correctifs appliqués.

Rappel: formats de fichiers et code malveillant (cf. archives SSTIC03)

- **Risque classé de 1 à 5:**
  - 1) Format contenant toujours du code
  - 2) Format contenant parfois du code, et celui-ci peut s'exécuter directement
  - 3) Format contenant parfois du code, et celui-ci ne peut s'exécuter qu'après confirmation
  - 4) Format contenant parfois du code, et celui-ci ne peut s'exécuter qu'après action volontaire de l'utilisateur
  - 5) Format ne contenant jamais de code

- **Conteneur**:
  - format pouvant contenir d'autres fichiers (archives zip, documents Word, …)

Message HTML avec contenu actif

- Le **corps d'un message HTML** peut aussi contenir du **code malveillant** :
  - Généralement sous forme de **script** (Vbscript, Javascript ou Jscript) dans le code HTML.
    - Script exécuté par le client de messagerie
  - Exploitation de **failles** du client de messagerie
    - (souvent Outlook / Outlook Express + Internet Explorer)
  - **Lancement automatique** d'une pièce jointe (virus, cheval de Troie) sans action de l'utilisateur
  - Ou action directe sur le système.

Spam spam spam…

Il fut un temps où le Spam ne débordait pas encore de nos boîtes aux lettres…

Spam

- **Messages publicitaires non sollicités**
  - Envoyés en masse (milliers ou millions d'exemplaires).
  - Occupent aujourd'hui 50% du trafic mondial de messagerie sur Internet.
  - Adresse émetteur usurpée.
  - Perte de temps pour les utilisateurs.
  - Réduction de la bande passante réseau, et des ressources des serveurs de messagerie.

Usurpation d'identité

- **Le protocole SMTP ne garantit absolument pas l'identité de l'émetteur:**
  - Usurpation possible de n'importe quelle adresse (existante ou non)
  - Risque de « social engineering »
    - Message semblant provenir de la hiérarchie ou de l'administrateur, ordonnant une action ou demandant une information
  - Très utilisé par les virus, qui exploitent les carnets d'adresses des clients de messagerie
    - On se méfie moins d'un message provenant d'un ami
- Les utilisateurs sont rarement conscients de ce problème et de ses implications

Relayage

- Si un serveur de messagerie est configuré sans précaution, il peut fonctionner en mode « **open relay** »
  - Autorise le **relayage non contrôlé** de messages provenant de n'importe quel serveur externe vers une destination quelconque.
  - Permet de **camoufler la source réelle** d'un message, par rebond.
  - Utilisé frauduleusement pour l'envoi de spams, de virus, ou toute autre attaque.
  - Un serveur open relay risque d'être considéré comme complice en cas de poursuite judiciaire.
- Ou relayage par adresse avec plusieurs domaines
  - Exemple: « victime@societe.com »@relais.fr

Vulnérabilités des clients et serveurs

- Un serveur ou un client de messagerie peut comporter des vulnérabilités, comme tout logiciel :
  - Un simple message habilement forgé peut alors provoquer un **déni de service**, ou l'**exécution de code** sur le serveur ou le client
  - Exemple: débordement de tampon sur un champ SMTP ou MIME
    - Cas de Sendmail en 2003: cf. « Sendmail Header Processing Buffer Overflow Vulnerability », BUGTRAQ, http://www.securityfocus.com/bid/6991

Mailbombing

- Il est facile d'émettre rapidement un grand nombre de messages pour saturer une boîte aux lettres ou un serveur
  - Nombreux outils existants
  - Attaque à portée limitée, sans grand intérêt
  - …sauf si la boîte aux lettres visée a une fonction vitale ou de sécurité
    - Par exemple, une BAL de supervision qui reçoit les alertes IDS pourrait être saturée pour camoufler une autre attaque

Web bugs

- Un « web bug » est une image incorporée sous forme de lien HTTP dans un message HTML:
  - A la visualisation du message, le client de messagerie télécharge automatiquement cette image depuis le serveur web indiqué
  - Celui-ci est alors prévenu que le destinataire a ouvert le message
    - La requête HTTP est enregistrée dans les logs du serveur web (adresse IP, date/heure, …)
    - Cela peut servir à faire des statistiques, ou à glaner des informations pour les spammeurs
    - Plus efficace et plus furtif qu'un accusé de réception !
  - L'image peut même être invisible (1 pixel blanc suffit)
- …heureusement, toutes les images des messages ne sont pas des web bugs.

Protection

- 2 approches:
  - **1) Assurer la sécurité au niveau des clients**:
    - Appliquer systématiquement tous les correctifs (OS et tous les logiciels installés)
    - Supprimer toutes les fonctionnalités non nécessaires
    - Durcir le paramétrage (OS et tous logiciels)
    - => coûteux en ressources, risque de brider les utilisateurs
  - **2) Filtrer les flux à l'entrée du réseau**
  - **…Ou les deux à la fois**

2) Comment filtrer ?

Objectifs d'un filtrage de messagerie « idéal »

Techniques de filtrage

Protections contre ces risques

Risques	Protections
Attaques réseau sur port SMTP	Pare-feu, routeur, IDS/IPS, …
Virus, vers	Antivirus, Filtrage des exécutables et contenus actifs
Code malveillant, Chevaux de Troie	Filtrage des exécutables et contenus actifs
Spam	Filtrages antispam (listes noires, Bayes, …)
Usurpation d'identité	Signature cryptographique (S/MIME, …)
Relayage externe	Filtrage sur adresses émetteur / destinataire
Vulnérabilités des serveurs et clients	Mises à jour régulières
Mailbombing	Filtrage sur nombre de messages
Web bugs	Filtrage HTML, Configuration des clients de messagerie, Filtrage des connexions HTTP

Objectifs d'un filtrage « idéal »

- Détection et blocage des **virus/vers**
  - Connus ou inconnus
- Nettoyage de **tout contenu actif** potentiellement **malveillant**:
  - Dans les pièces jointes
  - Dans le corps HTML des messages
- Destruction des messages de **spam**

- …le tout **sans perturber le trafic normal** de messagerie ! (pas de faux-positifs)
- …avec des **performances** et une **stabilité** acceptables.

Filtrage de code malveillant

- **Il n'existe aucun moyen sûr pour distinguer un code normal d'un code malveillant**
  - On peut s'en approcher dans certains cas : par exemple Jscript, Vbscript (langage limité)
  - Ou signature de code pour une relation de confiance

- => Pour protéger un réseau sensible, la seule solution est de **filtrer tout contenu actif**.

Filtrage de tout contenu actif

- Aucun **fichier purement exécutable** ne doit parvenir à l'utilisateur
- Aucun **document** (HTML, XML, PDF, Office, …) ne doit contenir de **code actif** (macro, script, …)
  - Le code éventuel doit être **nettoyé** ou **bloqué**
- Ces traitements sont aussi valables à l'intérieur des fichiers **conteneurs** (archives, documents imbriqués, …)

- **=> Analyse de contenu**
  - Messages et pièces jointes

3) Analyse de contenu

Les techniques

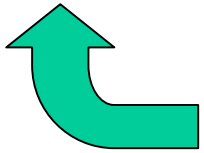
Les limites

Pistes d'amélioration

Analyse de contenu récursive

1. **Détection du format** du message ou du fichier
   - Texte, HTML, Office, Zip, PDF, …

2. **Action suivant le format** détecté et la politique de filtrage:
   - Blocage (du message ou de la PJ)
   - Nettoyage de code
   - Extraction du contenu (si conteneur)
   - Acceptation du contenu sans modification



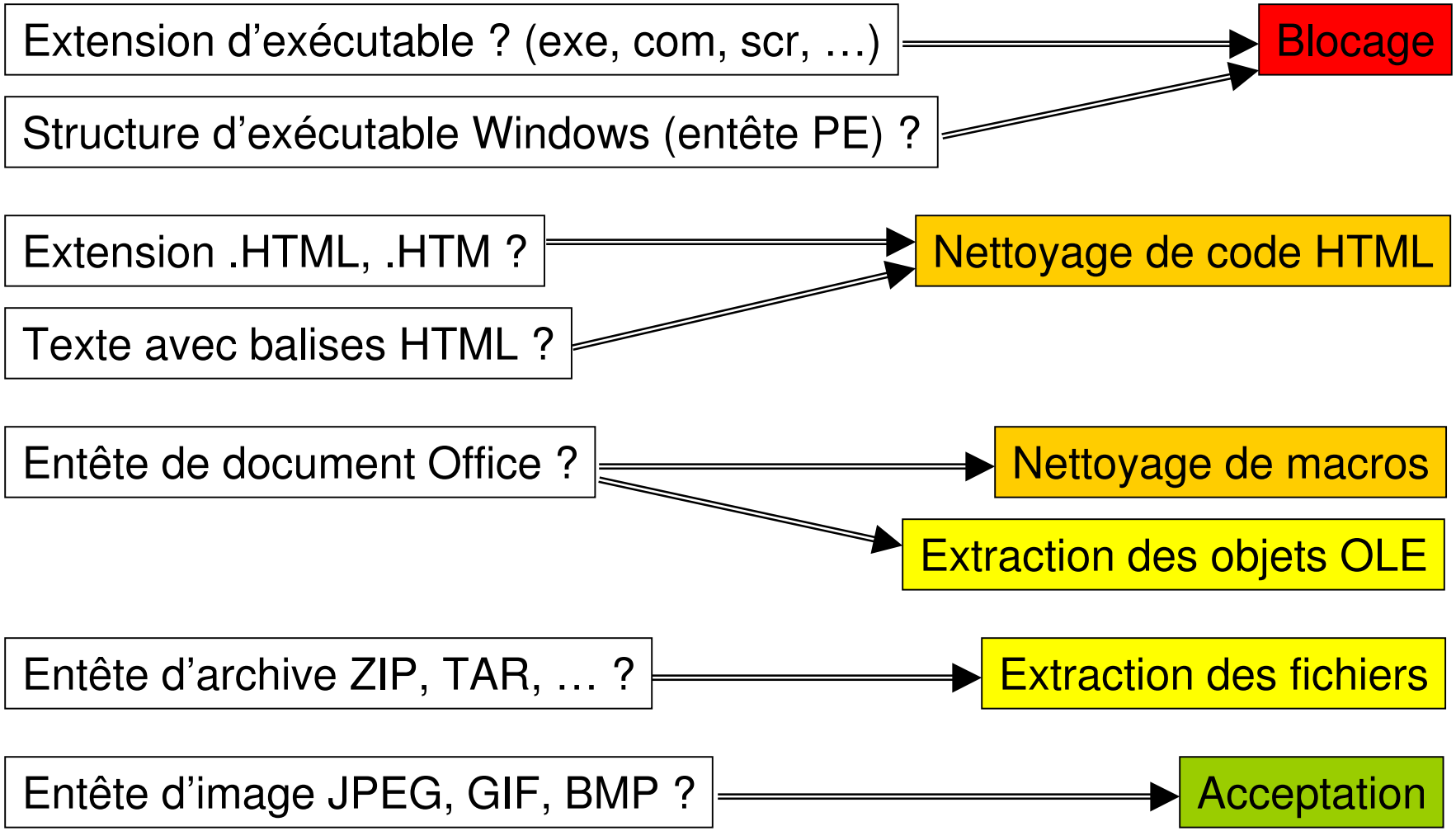
3. **Analyse récursive** des éléments extraits s'il y en a

Détection de format

- Pour être efficace et sûr, le filtrage des pièces jointes doit s'appuyer sur une bonne **détection des formats de fichiers / messages**

- **Techniques** pour déterminer le format:
  - **Extension** du nom de fichier
  - Champ MIME **content-type**
  - **Contenu** interne du fichier:
    - « **Magic** » = 2 ou 4 premiers octets: « PK » pour Zip, « D0 CF » pour Office, …
    - **Structure binaire particulière**: Entête PE pour exécutables Win32
    - **Mots-clés**: « %PDF » pour PDF, « <XML> » pour XML, « {\rtf » pour RTF, …

Exemple d'analyse de contenu



Filtrage de fichiers par nom

- La plupart des formats de fichiers à risque (exécutables Windows) peuvent être détectés simplement par leur **extension**
  - Exe, com, bat, cmd, scr, pif, lnk, vbs, js, …
- Cette détection est suffisante **pour bloquer un format interdit** car Windows choisit l'application à lancer suivant l'extension.
  - …la plupart du temps.

Le filtrage par nom ne suffit pas

- Plusieurs raisons:
  - Certains types de fichiers comme les documents Office peuvent être renommés avec des extensions inconnues (.xyz par exemple), tout en restant utilisables
    - Possibilité d'échapper au nettoyage de macros
  - Avec un message convaincant, on peut amener un utilisateur à renommer un fichier avant de l'ouvrir.
  - Il existe une pléthore d'extensions possibles sous Windows, et il est difficile de couvrir tous les cas de figure avec certitude.
  - Le comportement du client de messagerie n'est pas toujours celui attendu
    - Exemple d'Outlook Express avec triple extension « .jpg.exe.jpg »

Le filtrage par contenu ne suffit pas

- Certains formats sont **difficiles à reconnaître** par leur simple contenu
  - Exemples: HTML, VBScript, Jscript, …
  - Erreurs possibles: faux-positifs ou faux-négatifs
- Certains formats sont **permissifs** dans leur structure
  - Exemple: PDF, qui peut contenir du texte avant le marqueur « %PDF » → possibilité de leurrer le logiciel de filtrage

- **Conclusion**: l'analyse doit prendre en compte le nom ET le contenu du fichier

Nettoyage de code dans HTML

- A nettoyer: **scripts** (Javascript ou Vbscript), **objets ActiveX**, **applets Java**, ainsi que codes destinés à exploiter les **failles** du navigateur.
  - Dans les PJ HTML ou le corps des messages
- Diverses techniques pour neutraliser un script:
  - Suppression simple du code
  - Mise en commentaires du code: <!-- … -->
  - Remplacement par des balises inactives

Nettoyage des scripts dans HTML

- Faiblesses: Certaines techniques de nettoyage sont parfois contournables.
- **Exemple 1: scripts imbriqués**

```
<SCR<SCRIPT>…</SCRIPT>IPT>
// code malveillant
</SCRIPT>
```

Nettoyage par simple suppression
(ce qui active le véritable script)

```
<SCRIPT>
// code malveillant
</SCRIPT>
```

Nettoyage des scripts dans HTML

- **Exemple 2: camouflage de mots-clés**
  - Grâce aux fonctionnalités des langages HTML, XML et SGML:

```
<A HREF=«ja&#118ascript:…code malveillant…»>
lien
</A>
```

- **Exemple 3: formats peu répandus**
  - Internet Explorer reconnaît les fichiers HTML/XML au format **Unicode** (UCS-2 Big Endian ou Little Endian), mais pas tous les logiciels de filtrage HTML…

Nettoyage de macros

- Suppression de toutes les macros des documents MS Office
- Exemples:
  - Antivir
  - F-Prot
  - Macrostripper (Clearswift)
  - …

Formats conteneurs

- Archives: Zip, Rar, Arj, Lzh, Tar, Gz, Bz2, Cab, Msi, …
  - Nombreux formats répandus
  - Certains cas de figure parfois mal pris en compte: noms de fichiers accentués, commentaires, …
- Objets OLE (dans Word, Excel, RTF, …):
  - Très nombreux formats et versions
  - Exemple: OLE package dans RTF
    - Souvent non détecté par filtres de messagerie ou antivirus

« Zip of death »

- Plusieurs Giga-octets de caractères identiques peuvent être compressés dans quelques kilo-octets (facteur 1000 voire 1.000.000)
  - Exemple: un fichier d'1 Go de zéros = Zip d'1 Mo, ou RAR d'1 Ko
- La plupart des logiciels de filtrage décompressent les archives sur disque pour en analyser le contenu
  - Conséquence: l'analyse peut facilement saturer le disque, et occuper le CPU à 100% pendant des heures
  - →Déni de service
- **Solution: fixer des limites** pour l'espace disque temporaire, le temps d'analyse, et la taille cumulée des fichiers décompressés

Webmail

- Problème: lorsque les utilisateurs reçoivent des messages via une interface webmail, les pièces jointes transitent par HTTP, et non par la passerelle SMTP
- Il faudrait donc appliquer la même politique de filtrage et d'analyse de contenu sur HTTP
- …mais les contraintes de **performances** ne sont pas du tout les mêmes
  - SMTP = asynchrone / HTTP = temps réel
- Le webmail peut donc être plus difficile à sécuriser

Formats chiffrés

- Un message ou un fichier chiffré ne peut être analysé (sauf si séquestre de clé):
  - Message S/MIME
  - Archive chiffrée par mot de passe: Zip, Rar, …
  - Document protégé par mot de passe: Word, Excel, …
- => Blocage nécessaire

Formats « découpés »

- Certains formats peuvent être découpés en plusieurs morceaux, ce qui peut perturber l'analyse de contenu
  - Messages SMTP/MIME
  - Archives compressées: Zip, Rar, …
- => Blocage nécessaire

Stéganographie

- Quelle que soit la politique de sécurité et le filtrage appliqué, il est toujours possible de **maquiller un format de fichier interdit dans un autre format autorisé**
  - Exemple: EXE dans une image GIF

- Cependant cela nécessite une **complicité** de l'utilisateur qui reçoit le fichier, afin d'extraire le fichier camouflé
  - …social engineering ?

Pistes d'amélioration

- **Un logiciel de filtrage doit être aussi proche que possible des applications qu'il protège**, afin d'éviter les possibilités de **camouflage** ou de **faux-négatifs**
  - Formats reconnus, encodages, algorithmes, Mots-clés
  - Exemple: Internet Explorer
- Cependant il ne faut pas qu'il souffre des mêmes **vulnérabilités**
- Après nettoyage d'un fichier, la présence de code doit être **revérifiée une seconde fois**
- Les détections par nom et par contenu doivent être complémentaires pour éviter les erreurs

Conclusion

- La messagerie est un des services les plus utilisés sur Internet, et pourtant **les risques sont nombreux**
  - Virus, vers, code malveillant, spam, …
- Il existe beaucoup de techniques et de logiciels de filtrage pour la messagerie
- Cependant **il est très difficile de protéger parfaitement un réseau Windows** avec les logiciels actuels en raison de leurs limites
  - En particulier pour la détection et le nettoyage de code malveillant, ainsi que pour la lutte antispam
- Il est possible d'améliorer les techniques de filtrage
- La recherche dans ce domaine est donc capitale.

Questions