Sécurité des formats

OpenDocument et Open XML

(OpenOffice.org et MS Office 2007)

SSTIC07 – 01/06/2007 – http://www.sstic.org

Philippe Lagadec – NATO/NC3A
philippe.lagadec(à)nc3a.nato.int

OpenOffice et OpenDocument

- **OpenOffice.org**
  - Version open-source de Sun StarOffice
  - Surnommée aussi "**OOo**"
  - Capable de lire/écrire la plupart des documents MS-Office
- **OpenDocument**
  - Nouveau format pour les documents OpenOffice v2
    - Très proche du format OpenOffice v1
  - De plus en plus utilisé par d'autres applications (Koffice, Abiword…)
  - Fichiers XML dans une archive ZIP
  - standard ISO depuis mai 2006, OASIS depuis 2005

MS Office 2007 et Open XML

- **Microsoft Office 2007**
  - Précédemment appelé “Office 12”
  - Nouvelle version de MS-Office, beaucoup de changements
  - Disponible depuis décembre 2006
- **Open XML**
  - Nouveau format **par défaut** pour la plupart des documents Office 2007 (Word, Excel, PowerPoint, à l’exception d’Access)
  - Fichiers XML dans une archive ZIP
  - standard ECMA depuis décembre 2006, standardisation ISO en cours

Rappel 1: Code malveillant dans les fichiers

- **La plupart des formats de fichiers usuels peuvent incorporer des contenus actifs, qui peuvent être malveillants:**
  - EXE, COM, PIF, SCR, … : *Code binaire*
  - BAT, CMD, VBS, JS, … : *Commandes, Scripts*
  - HTML, XML, XHTML : *Scripts*
  - PDF : *Scripts, Pièces jointes, Commandes*
  - Word, Excel, PowerPoint, Access, … : *Macros, Objets OLE, Fichiers inclus, Commandes*
    - Plus de détails: http://actes.sstic.org/SSTIC03
- **Risque généralement sous-estimé, car les virus « grand public » n'utilisent pas toutes les fonctionnalités de ces formats.**
  - **…mais ils peuvent tout de même servir pour un cheval de Troie !**
  - **Parfois c'est l'unique moyen d'attaquer un système sécurisé**

Rappel 2: Fuite d'information dans les documents

- **Les documents bureautiques classiques peuvent contenir beaucoup d'<u>informations cachées</u>:**
  - Nom d'utilisateur, Société
  - Historique des modifications, texte ajouté ou supprimé
  - Notes, Commentaires
  - Texte caché
  - Une feuille de calcul complète derrière un simple graphique (remplie de chiffres confidentiels !)
  - Parfois même des portions de mémoire
- **L'expérience a déjà montré que cette information peut être exploitée de façon malveillante si elle arrive dans de mauvaises mains.**

Sécurité d'OpenDocument et Open XML

- Analyse de sécurité:
  - **Inclusion de contenus actifs malveillants**
  - **Fuite d'informations sensibles cachées**
  - **Nouveaux risques**
  - **Signature, chiffrement, DRM**
  - **Protection des systèmes d'informations**
  - **Filtrage de contenu**
- Les travaux à l'origine de cette analyse ont été financés successivement par DGA/CELAR et par le programme de travail de recherche et développement de NATO/ACT.
- Analyse non exhaustive
- Précédente analyse sécurité d'OpenOffice v1.0 par F-Secure en 2003:
  - http://www.f-secure.com/weblog/archives/openoffice_security.pdf

Versions analysées

- **OpenOffice.org 2.2.0**
- Spécifications **OpenDocument v1.0**
  - http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf

- **MS Office 2007 version 12.0**.4518.1014
- spécifications **Open XML ECMA-376**, version finale publiée en décembre 2006
  - http://www.ecma-international.org/publications/standards/Ecma-376.htm

- sur Windows XP SP2 fr

Analyse des spécifications

- Un des principaux avantages de ces formats ***ouverts***, par rapport aux bons vieux formats propriétaires:
  - L'analyse de sécurité est bien plus facile :-)
- Cependant il faut quand même lire les specs…

- OpenDocument : **700 pages**
- Open XML : **6045 pages !! ;-(**
  - Et même avec tout cela, tout n'est pas décrit, par exemple les macros VBA…

Formats similaires

- Autres formats bureautiques utilisant XML et ZIP:
- **XPS**
  - Nouveau format Microsoft pour publier/imprimer les documents finalisés, similaire à PDF.
  - http://www.microsoft.com/whdc/xps/default.mspx
- **MARS**
  - Future version de PDF, en cours de définition par Adobe.
  - Format PDF intégralement transcrit en XML et SVG.
  - Plugin de lecture/écriture déjà disponible pour Adobe Acrobat 8.
  - http://labs.adobe.com/technologies/mars/

Partie 1: OpenOffice.org et OpenDocument

Les fichiers d'OpenOffice.org

Seuls les textes, classeurs, présentations et dessins OOo v2 sont couverts par les spécifications OpenDocument v1.0.

Format	Application	document OOo v2	modèle OOo v2	document OOo v1	modèle OOo v1
Texte	**Writer**	**.odt**	**.ott**	.sxw	.stw
Classeur	**Calc**	**.ods**	**.ots**	.sxc	.stc
Présentation	**Impress**	**.odp**	**.otp**	.sxi	.sti
Dessin	**Draw**	**.odg**	**.otg**	.sxd	.std
Base de données	Base	.odb			
Modèle HTML	Writer/Web	(.html)	.oth	(.html)	.stw
Document maître	Writer	.odm		.sxg	
Formule	Math	.odf		.sxm	

Structure d'OpenDocument

- **Un document est stocké dans une archive compressée ZIP**
- **Fichiers XML:**
  - **content.xml**: texte du document
  - **styles.xml**: données de styles
  - **meta.xml**: méta-données (auteur, titre, …)
  - **settings.xml**: paramètres OOo pour le document
  - **META-INF/manifest.xml**: description des fichiers

- **Fichiers optionnels:**
  - Images et vignettes: JPEG, PNG, SVG, …
  - Graphiques/dessins/documents inclus, objets OLE

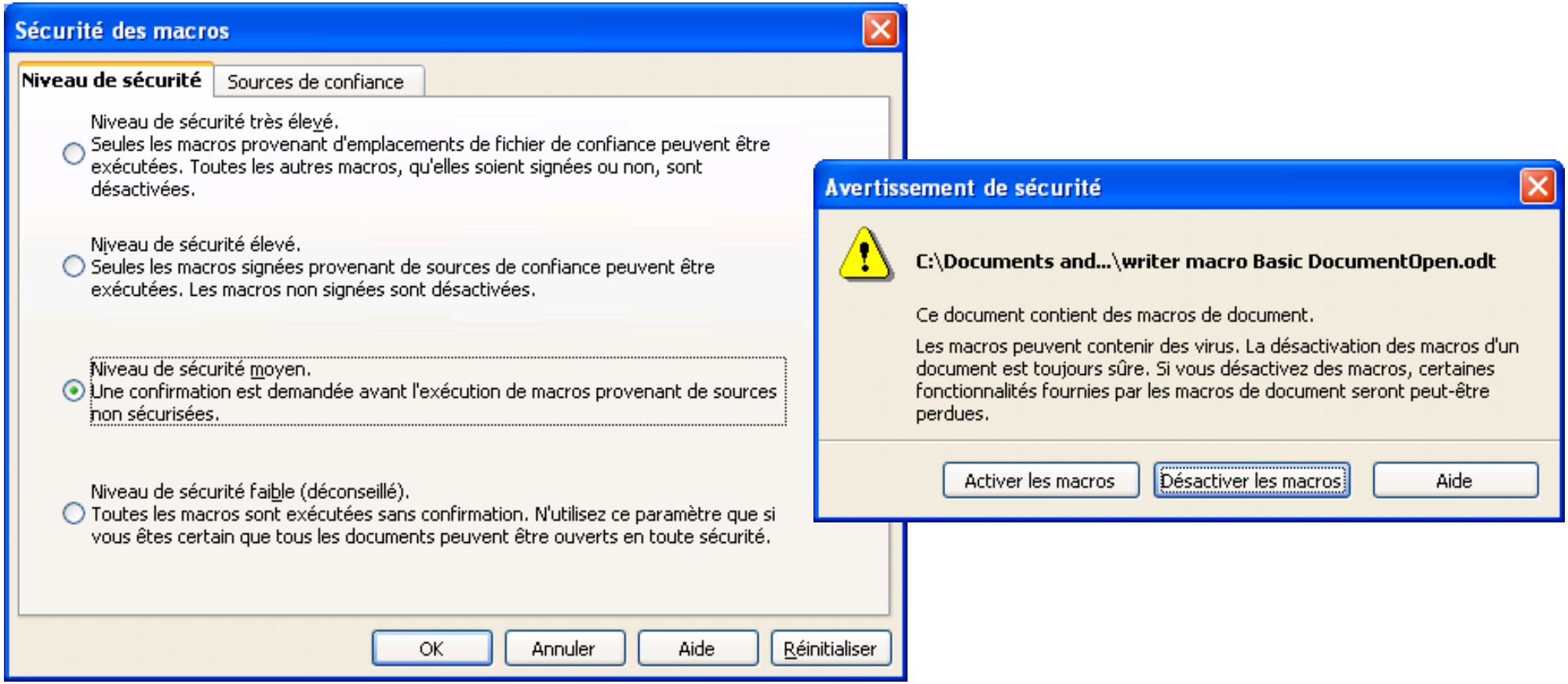
Macros OpenOffice

- **OpenOffice v2.x propose 4 langages pour les macros:**
  - Basic, Javascript, Java (Beanshell), Python
  - D'autres langages pourront être ajoutés à l'avenir.
- **Chaque langage donne accès à UNO:**
  - UNO: Universal Network Objects
  - **API très riche**: accès aux objets OpenOffice et au système d'exploitation
  - **Possibilité d'écrire du code malveillant efficace**.
- **Les macros peuvent être lancées automatiquement par des événements (ouverture du document, …) ou par des formulaires.**

Niveaux de sécurité des macros OpenOffice

- **4 niveaux, similaires à ceux de MS Office 2000-2003:**
  - **Bas** (à éviter): aucune protection
  - **Moyen (par défaut)**: macros activables par simple fenêtre d'avertissement avant ouverture du document.
  - **Haut**: Seules les macros signées ou les répertoires de confiance sont autorisés, pas d'avertissement.
  - **Très haut**: uniquement répertoires de confiance, pas de signatures, pas d'avertissement.
- **Même niveau par défaut que MS Office 97**
- Vulnérabilité d'OpenOffice 2.0.2 : possibilité de contourner l'avertissement des macros.

Niveaux de sécurité des macros OpenOffice



Stockage des macros OpenOffice

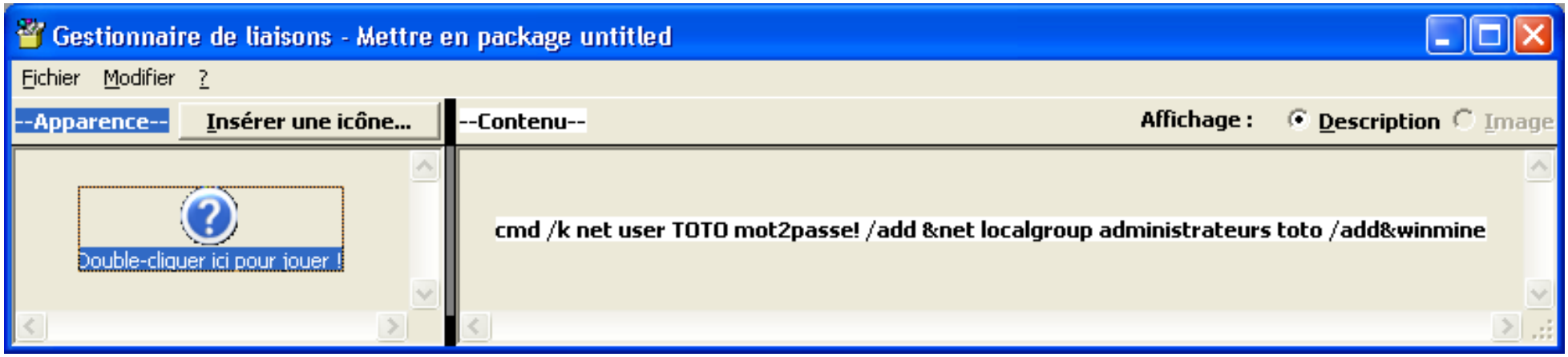
- Macros OOo **Basic** : fichiers XML, répertoire "**Basic**" de l'archive.
- Macros **Java, Javascript et Python :** fichiers scripts, dans le répertoire "**Scripts**".
- Exemples:
  - **Basic**/Standard/Module1.**xml**
  - **Scripts/beanshell**/Library1/MyMacro.**bsh**
  - **Scripts/javascript**/Library1/MyMacro.**js**
  - **Scripts/python**/MyMacro.**py**

Objets OLE dans OpenDocument

- **Les fichiers OpenDocument peuvent inclure des objets OLE** (sous Windows).
- **Un objet OLE est stocké dans un *fichier binaire* à l'intérieur du document.**
  - Au format Microsoft OLE2 storage (pas vraiment un format ouvert…)
- **Un objet OLE Package peut contenir n'importe quel fichier ou une ligne de commande (potentiellement malveillante).**
  - Si l'utilisateur double-clique sur l'objet, le fichier ou la commande est exécutée par le système.

Objet OLE Package

- Peut déclencher une commande malveillante, par exemple:



Objets OLE dans OpenDocument

- **OpenOffice n'affiche aucun avertissement avant d'ouvrir un objet OLE Package.**
  - L'avertissement provient uniquement de Windows (**packager.exe**)
  - Aucune confirmation sur de vieilles versions de Windows ! (2000 SP4)
- **Vulnérabilité (corrigée) Windows MS06-065 :**
  - Possibilité de camoufler la ligne de commande d'un objet OLE Package pour afficher un nom de fichier quelconque à la place:
    - cmd.exe /c […commandes malveillantes…] /blague.txt
  - http://secunia.com/advisories/20717

Autres risques

- Autres voies possibles pour inclure des contenus malveillants dans OpenDocument:
  - **Scripts HTML** : scripts (JavaScript ou VBscript), activés quand le document est converti en HTML puis ouvert dans le navigateur.
  - **Applets Java** : code Java exécuté dans une sandbox depuis OOo, ce qui devrait être sûr.
    - Mais par exemple OpenOffice 2.0.2 souffrait d'une vulnérabilité permettant d'échapper à la sandbox.
  - **URLs**: directement lancées dans le navigateur par défaut.
    - Heureusement les URLs Javascript et VBscript ne sont pas autorisées par OpenOffice.

Autres risques

- **Macros VBA** provenant de documents MS Office:
  - **Non exécutables** dans OpenOffice
  - Stockées dans les commentaires d'une macro OOo Basic vide lors d'une conversion vers OpenOffice.
    - Provoque les mêmes avertissements que les autres macros.
  - **Réactivées** si le document est à nouveau sauvé au format MS Office.
    - Peut servir de camouflage face à un antivirus.
  - Travaux en cours pour que les macros VBA soient directement exécutables dans les futures versions.

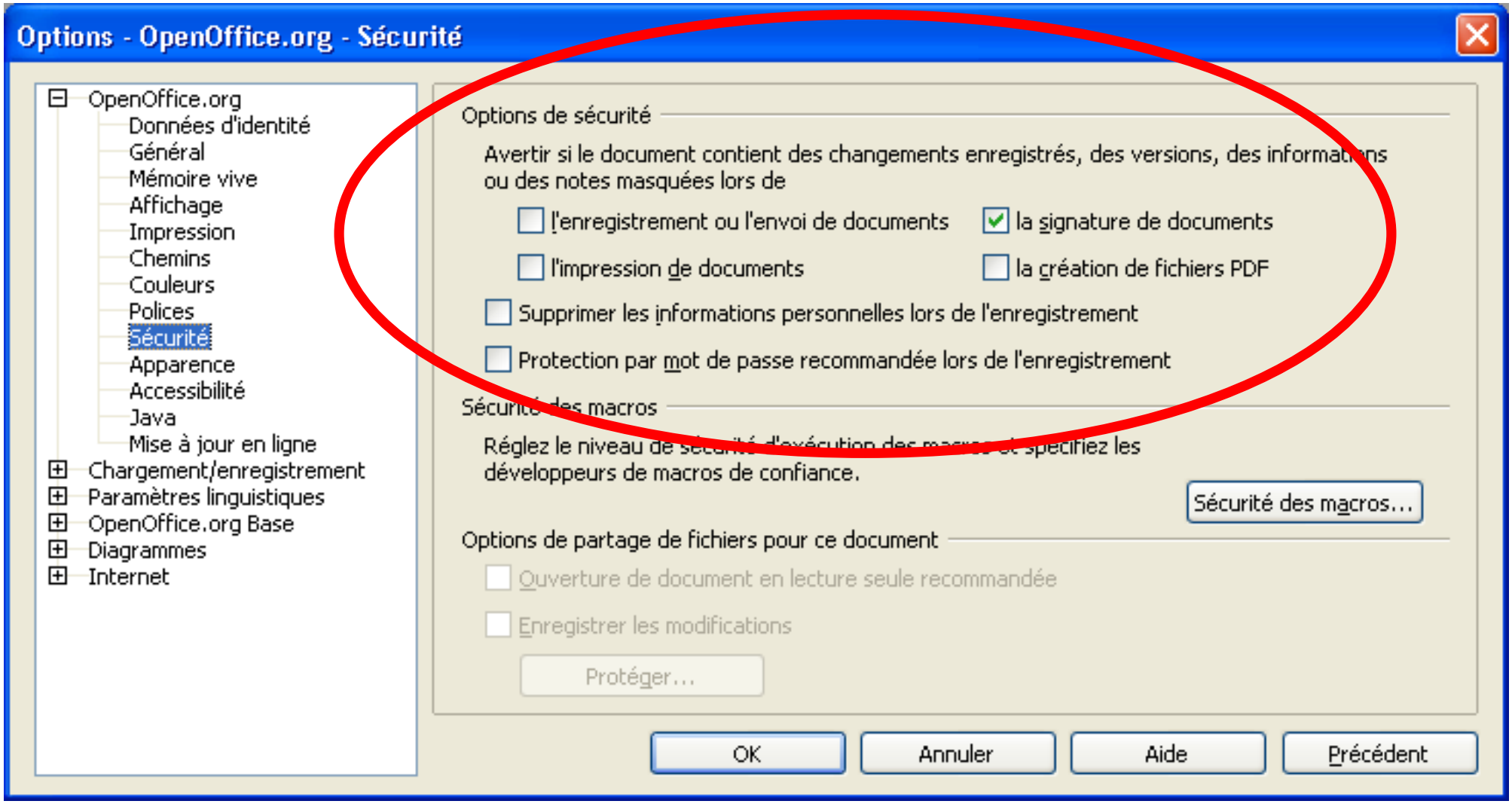
Autres risques

- Des chercheurs français de l'ESAT ont trouvé des faiblesses de conception dans la gestion des documents chiffrés/signés d'OpenOffice (entre autres) et montré qu'un virus OOo est possible:
  - Par exemple il est possible de remplacer une macro dans un document chiffré, par une autre macro en clair, sans générer d'avertissement.
    - De Drézigué, Fizaine, Hansma, *"In-depth Analysis of the Viral Threats with OpenOffice.org Documents",* Journal in Computer Virology, 2006.
      - http://www.springerlink.com/content/1772-9904/?k=openoffice
    - Filiol, Fizaine, *"Le risque viral sous OpenOffice 2.0.x",* MISC magazine n°27, 09/2006.
    - Filiol, Rump session SSTIC06, http://actes.sstic.org/SSTIC06

Les données cachées dans OpenDocument

- Comme MS Office, les documents d'OOo peuvent **camoufler des informations sensibles** à l'insu de l'utilisateur:
  - Méta-données, texte caché, commentaires, marques de révision, …
- OOo possède des fonctions pour signaler ces données cachées lors d'une signature, d'un export PDF ou d'une sauvegarde.
- Cependant cela ne couvre pas le cas des objets OLE.

Protection contre les données cachées dans OOo



Bilan: Sécurité d'OpenOffice et OpenDocument

- **OpenOffice n'est pas *absolument* plus (ou moins) sûr que MS Office, en ce qui concerne le code malveillant ou la fuite d'informations cachées.**
  - problèmes de sécurité similaires, quelques différences subtiles.
  - OpenDocument : a priori plus de possibilités pour camoufler du code malveillant, cependant certaines fonctionnalités sont plus sûres.
- **OpenDocument peut être un vecteur pour du code malveillant destiné à l'OS ou au navigateur.**
- OpenDocument peut contenir des fichiers binaires avec un format propriétaire (objets OLE)
- **Fonction de suppression d'informations cachées.**
- **Format ouvert OpenDocument : facilite la détection et le filtrage des contenus actifs et données cachées.**

Partie 2 :
MS Office 2007 et Open XML

Fichiers de MS Office 2007 : Open XML

- **Nouveaux formats Open XML par défaut :**
  - **Word**: .doc**x**, .doc**m**, .dot**x**, .dot**m**
  - **Excel**: .xls**x**, .xls**m**, .xlt**x**, .xlt**m**, .xls**b**, .xla**m**
  - **Powerpoint**: .ppt**x**, .ppt**m**, .pps**x**, .pps**m**
  - **Access**: .accdb (binaire, non Open XML)

- **Mode de compatibilité pour les anciens formats** (fichiers binaires OLE2): doc, dot, xls, xlt, ppt, pps, …

- **Pack de compatibilité** pour permettre à Office 2000, XP et 2003 de lire/écrire les documents Open XML.

Structure du format Open XML

- **Un document est stocké dans une archive ZIP compressée**
- **Open Packaging Conventions (OPC):**
  - Spécifications de base pour les nouveaux formats Microsoft : Open XML, XPS*
  - **[Content_Types].xml**: description de tous les fichiers dans l'archive
  - Fichiers **.RELS** (XML):
    - Stockent les relations indirectes entre les fichiers de l'archive OPC.
- **Fichiers XML** (exemple pour Word 2007):
  - word/**document.xml**: texte du document
  - word/**styles.xml**: styles employés
  - word/**settings.xml**: Paramètres MS Office pour le document
  - docProps/**app.xml** and **core.xml**: métadonnées (auteur, titre, …)
  - …
- **Fichiers binaires optionnels**:
  - Images et autres medias: JPEG, PNG, GIF, TIFF, WMF, …
  - Objets OLE, Macros VBA, paramètres d'impression, …

(*): XPS: nouveau format de Microsoft pour les documents finalisés, similaire à PDF

Macros Open XML

- **Open XML peut contenir des macros VBA, tout comme les anciens formats MS Office.**

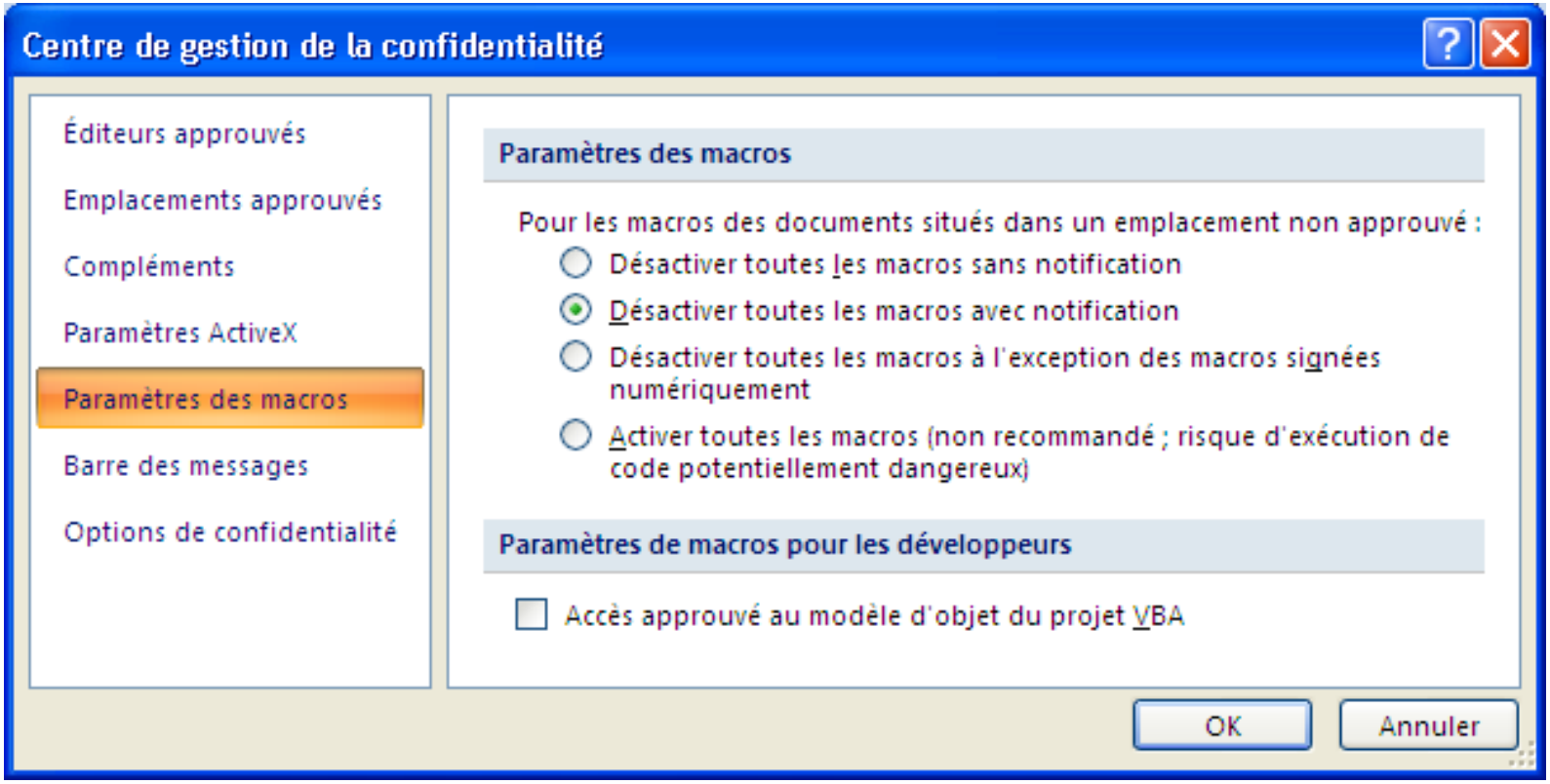
- **Nouveauté: Office 2007 distingue les documents "normaux" des documents "prenant en charge les macros" :**
  - Normal (par défaut): .doc**x**, .xls**x** et .ppt**x**
  - Avec macros: .doc**m**, .xls**m**, .ppt**m**
- **Les documents normaux "en x" ne peuvent contenir de macros.**
  - Un document "avec macro" renommé en "normal" est systématiquement rejeté par Office 2007.

Niveaux de sécurité des macros MS Office 2007

- **Office 2000/XP/2003: niveau de sécurité « haut » par défaut,** seules les macros signées peuvent êtres activées.
  - L'utilisateur doit passer au niveau « moyen » (et rouvrir le document) pour lancer les macros non signées.
  - Au niveau moyen, un message d'avertissement demande si les macros doivent être activées **AVANT** d'afficher le document.
- **Office 2007: Nouveaux niveaux et nouvelle IHM**
  - Plus de niveaux « moyen » ou « haut ».
  - **Niveau par défaut: "*Désactiver toutes les macros avec notification*"**

Niveaux de sécurité des macros MS Office 2007

- Le nouveau « Centre de gestion de la confidentialité » (Trust Center) regroupe tous les paramètres de sécurité:

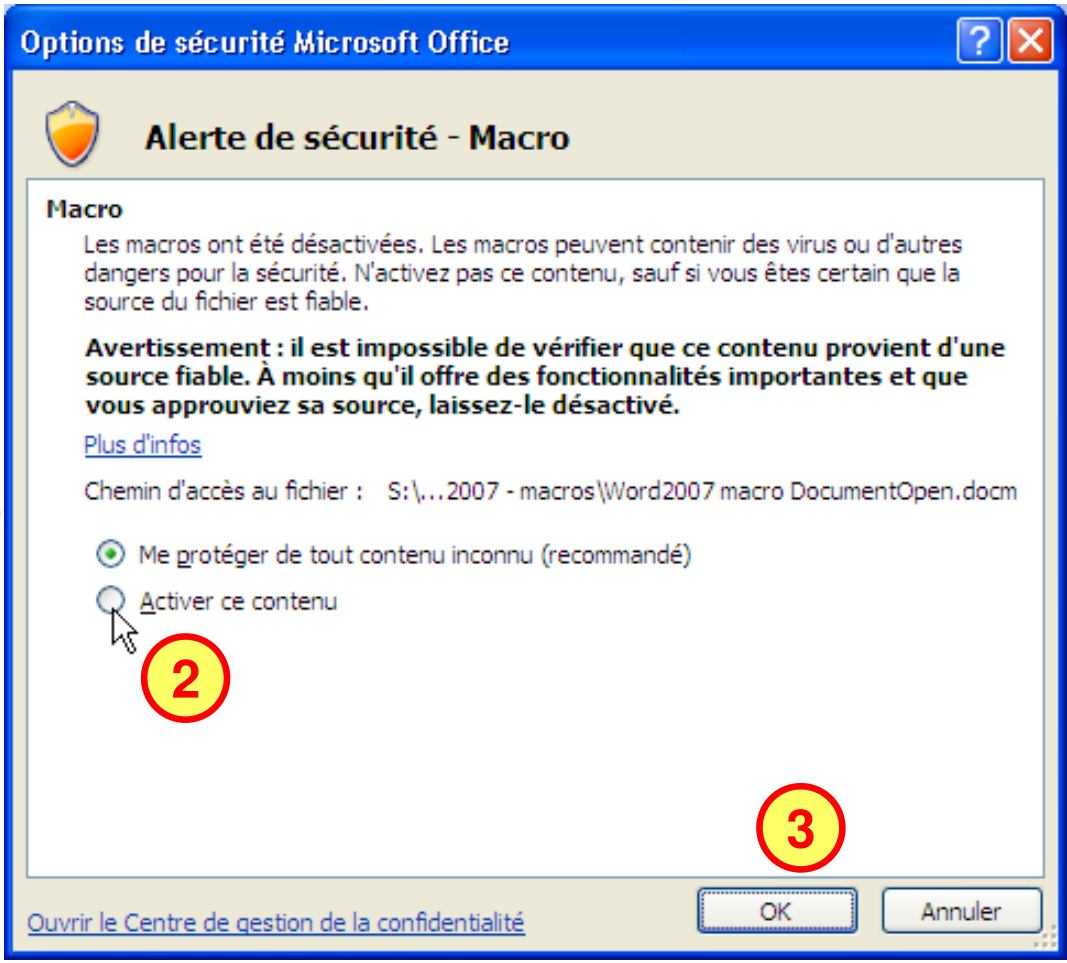


1

JEU:

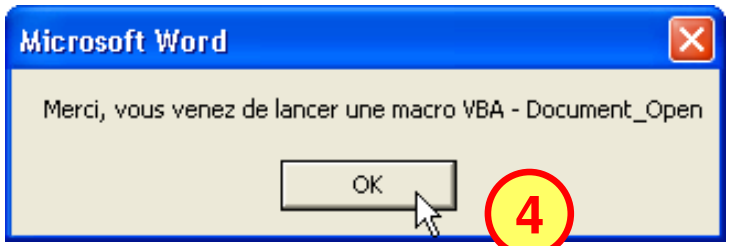
Quel est votre VRAI niveau en sécurité informatique ?

Pour démarrer, cliquez sur **Options** puis sur « **Activer le contenu** ».



2

3



4

Nouveau niveau de sécurité par défaut

- Nouveau niveau par défaut: "*désactiver toutes les macros avec notification*"
- **L'utilisateur peut activer n'importe quelle macro en 3 clics de souris (y compris les macros non signées).**
- De plus, il peut **activer les macros APRES lecture du document**.
  - **=> « ingénierie sociale » possible !**
- Le nouveau niveau de sécurité par défaut n'est donc pas vraiment plus sûr qu'avant…
  - Pour quelques explications de Microsoft : http://blogs.msdn.com/excel/archive/tags/Trust+Center/default.aspx
- Le code source des macros peut être lu avant de les activer.
  - Mais il faut être un développeur expérimenté pour reconnaître du code malveillant.

Stockage des macros MS Office 2007

- Les macros sont stockées dans un **fichier binaire OLE2**:
  - Word: **word/vbaProject.bin**
  - Excel: **xl/vbaProject.bin**
  - Powerpoint: **ppt/vbaProject.bin**

- **Cela n'est pas décrit dans les spécifications Open XML actuelles.**
  - Microsoft et l'ECMA considèrent que c'est une **technologie propriétaire, en dehors du format ouvert Open XML.**
    - http://www.ecma-international.org/news/TC45_current_work/Ecma%20responses.pdf
  - **Et OLE2 n'est pas vraiment un format ouvert.**

- Exemple: exécution automatique de macro depuis un document Word 2007 (.docm)
  - Macro nommée "Document_Open"
  - Word ajoute un tag dans word/vbaData.xml:
    - `<wne:eventDocOpen/>`

Objets OLE

- **Les documents Open XML peuvent contenir des objets OLE.**
- Exemple: possible de stocker un classeur Excel **avec macros** dans un document Word **sans macros**.
  - Quand cet objet est ouvert, Excel demande à activer les macros, même au niveau maximum “désactiver toutes les macros sans notification” !

- **Un objet OLE Package peut contenir n’importe quel fichier ou une ligne de commande (potentiellement malveillante).**
  - Lancement par double-clic sur l’objet.
  - Avertissement provenant uniquement de Windows (packager.exe).

- **Même problèmes que ceux montrés pour OpenOffice.**
  - Camouflage de ligne de commande

Stockage des objets OLE

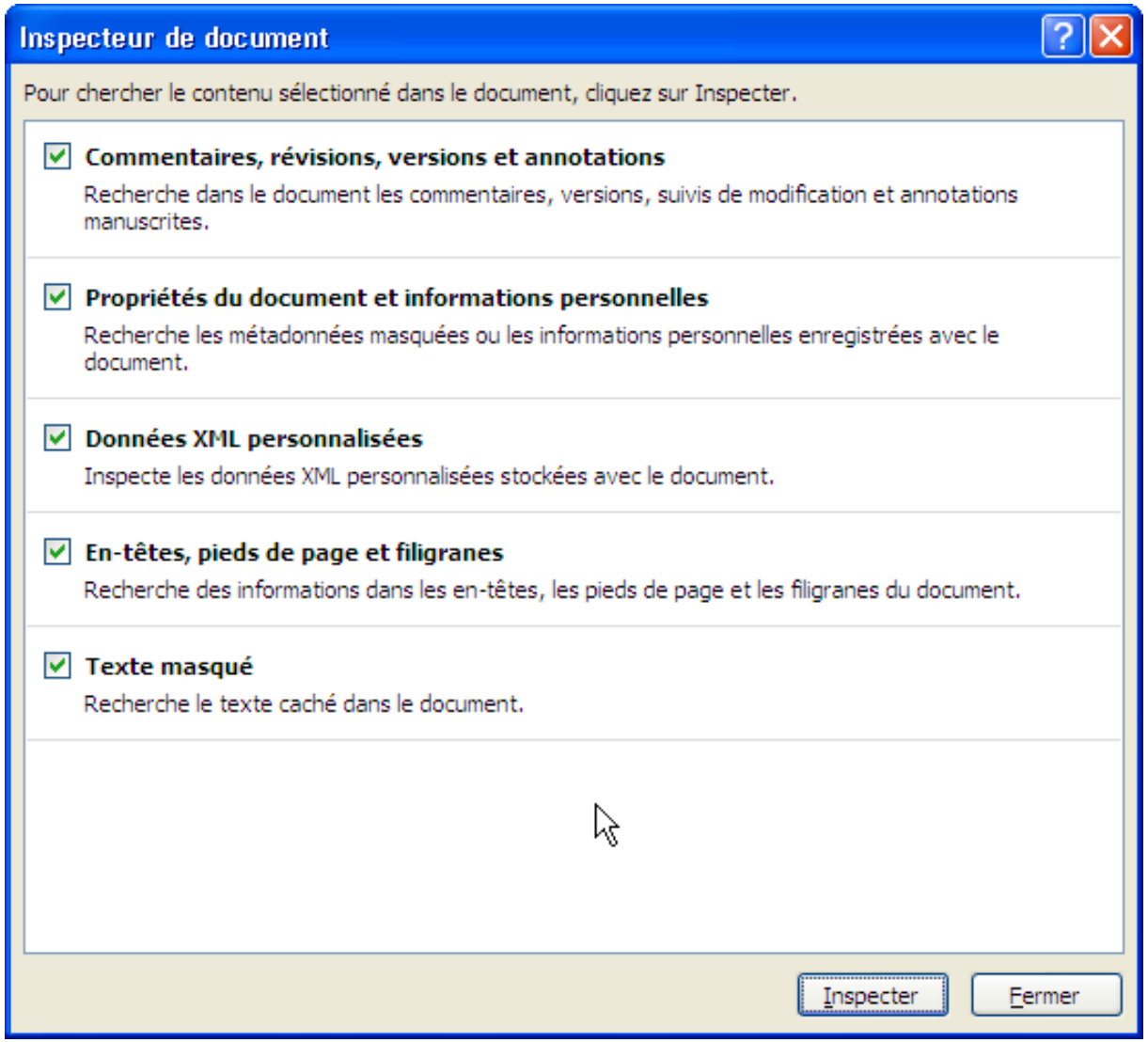
- Exemple de Word: objets OLE stockés dans **word/embeddings**.

- Objets OLE stockés dans leur **format natif** (par exemple xlsx dans docx).

- **OLE Package : format binaire Microsoft OLE2.**

Suppression des données cachées

- Office 2007 : nouvelle fonction pour supprimer les informations cachées des documents
  - “**Inspecteur de document**”, amélioration de l’outil RHDtool pour Office 2003/XP.
- Cependant les objets OLE ne sont pas signalés comme informations cachées.



Signature Open XML

- Un document Open XML peut être signé: authenticité, intégrité
- Signature conforme XML-Signature du W3C
  - http://www.w3.org/TR/xmldsig-core/
- **Attention: seuls les fichiers utiles du document sont signés, pas l'intégralité de l'archive.**
  - Le contenu affiché dans MS Office 2007 est signé.
- **Possible d'ajouter des fichiers tiers et de modifier [Content_Types].xml sans invalider la signature !**
- Fonction à utiliser avec précaution.

Bilan: Sécurité MS Office 2007 / Open XML

- ***Globalement*, mêmes problèmes de sécurité que les versions précédentes (macros, objets OLE, …)**
- **Sécurité *par défaut* pour les macros moins stricte qu'avant.**
  - **Possibilité de lancer des macros non signées.**
- **Open XML peut être un vecteur pour du code malveillant destiné à l'OS ou au navigateur.**
- Les documents Open XML peuvent contenir des fichiers binaires avec un format propriétaire : macros VBA, objets OLE, .xlsb, …
- Distinction forte des documents Open XML avec ou sans macros.
- **Fonctions améliorées de suppression d'informations cachées.**
- **Open XML facilite la détection et le filtrage des contenus actifs et informations cachées.**

Partie 3 :
Comment protéger les systèmes d'information

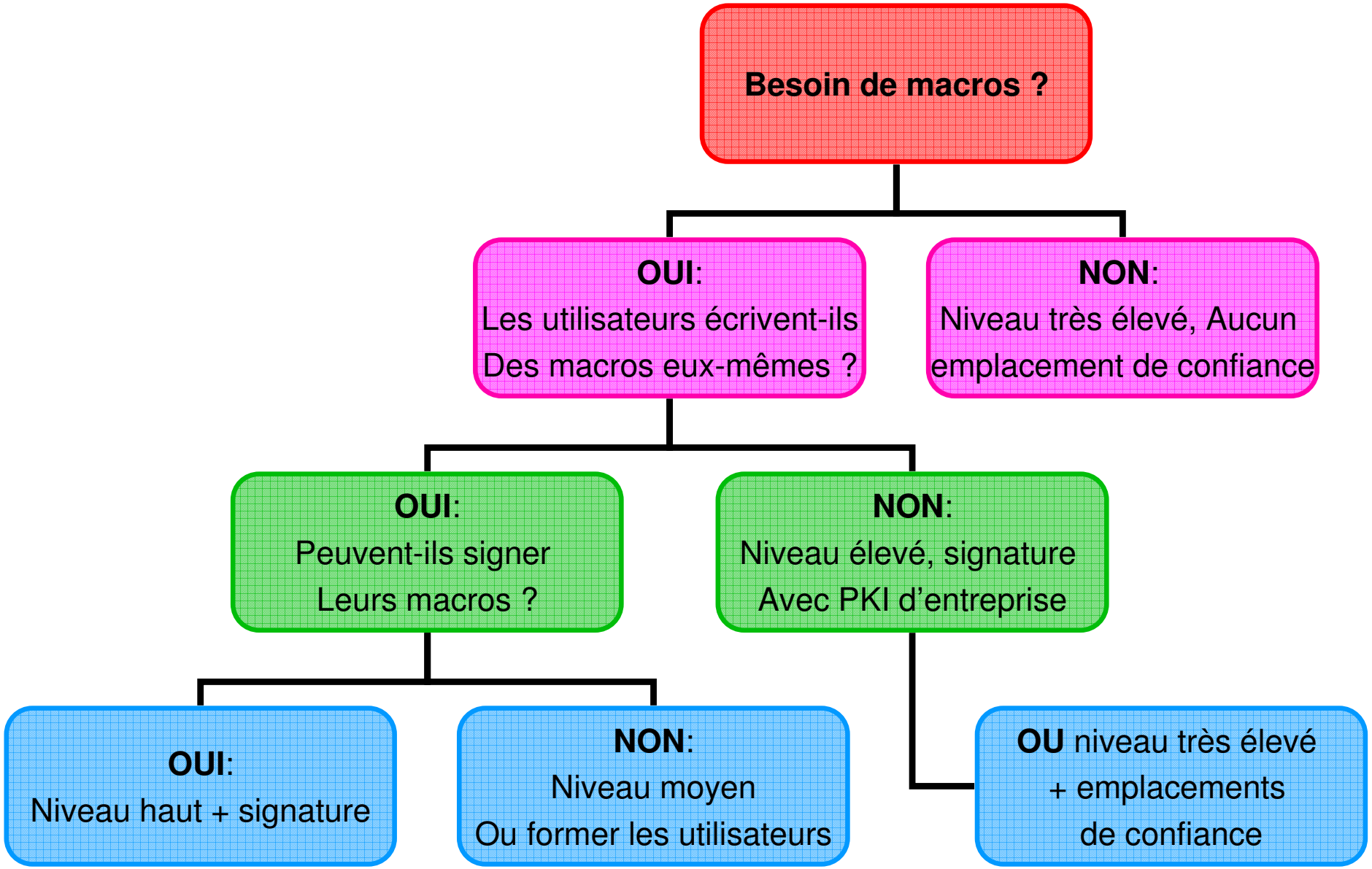
Protection

- 2 solutions complémentaires:
  - **Durcir la configuration** d'OpenOffice et MS Office 2007.
  - **Filtrer les documents entrants et sortants**
    - Sur une passerelle: SMTP, proxy HTTP, FTP
    - Sur les supports amovibles

Configuration sécurisée OpenOffice et MS Office

- **Bien sûr, installer tous les correctifs de sécurité, si possible automatiquement.** (fallait-il le préciser ? ;-)
- **Appliquer des paramètres de sécurité adaptés aux justes besoins des utilisateurs:**
  - Niveaux de sécurité pour les macros, ActiveX, …
  - Emplacements de confiance
  - Certificats
- **Protéger ces paramètres de sécurité:**
  - Ils ne devraient être modifiables que par les administrateurs.
- **Si les objets OLE Package ne sont pas utilisés:**
  - **Interdire l'exécution de** C:\Windows\System32**Packager.exe**
- **Protéger le poste de travail:**
  - Sécurité du navigateur
  - Antivirus, pare-feu personnel
  - Eviter l'utilisation des comptes administrateurs et utilisateurs avec pouvoir

Exemple:
Niveaux de sécurité OpenOffice



Sécurisation OpenOffice

- Peu documenté aujourd'hui
- Problème: paramètres de sécurité stockés dans des fichiers « .xcu » (XML) dans le profil de chaque utilisateur.
  - Impossible de verrouiller certains paramètres sans tout bloquer.
- **Désactiver complètement le support des macros:**
  - **Solution 1**: Modifier la valeur **DisableMacrosExecution** dans **common.xcs**
    - c:\Program Files\OpenOffice.org 2.2\share\registry\schema\org\openoffice\Office\
  - Ne concernera que les utilisateurs n'ayant pas encore lancé OpenOffice.

Sécurisation OpenOffice

- **Désactiver complètement le support des macros:**
  - Solution 2: Ajouter les lignes suivantes dans **common.xcu**
    - Dans c:\Documents and Settings\<utilisateur>\Application Data\OpenOffice.org2\user\registry\data\org\openoffice\Office\

```
<node oor:name="Security">
 <node oor:name="Scripting">
  <prop oor:name="DisableMacrosExecution" oor:type="xs:boolean">
   <value>false</value>
  </prop>
 </node>
</node>
```

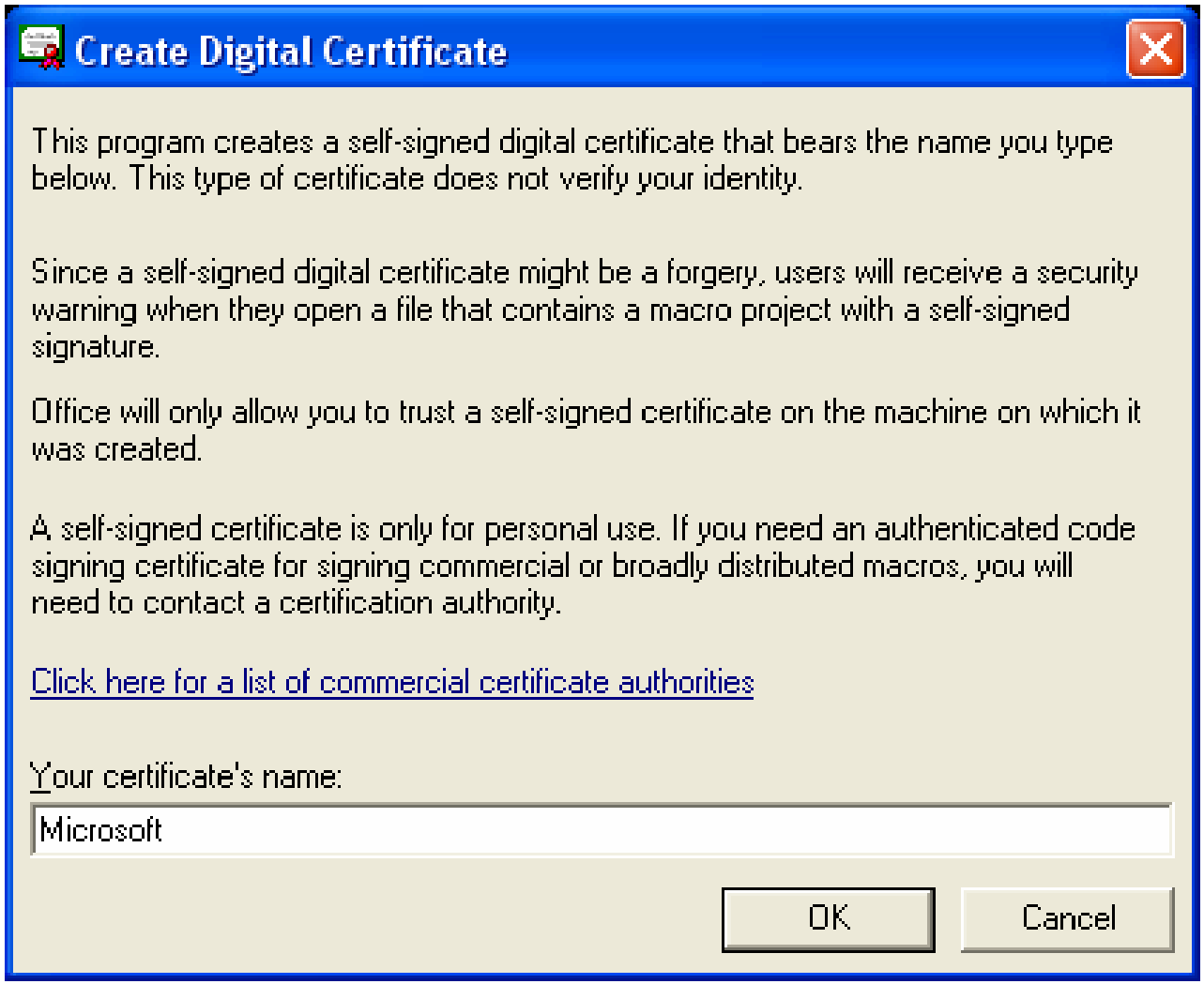
  - Effet de bord: l'utilisateur ne peut plus modifier les paramètres des macros dans l'IHM.
    - Mais il peut toujours modifier common.xcu à la main dans son profil…
  - Et certains modules utiles risquent de ne plus fonctionner…

Sécurisation MS Office 2007

- **Choisir les niveaux de sécurité les plus hauts pour les Macros et ActiveX.**
  - Macros : choisir “désactiver toutes les macros SANS notification” si possible.
  - Ou “désactiver toutes les macros sauf signées” si la signature est employée.
    - ET désactiver les notifications de la barre de messages pour bloquer les macros non signées.
  - Alternative: désactiver totalement VBA (attention aux effets secondaires…)
    - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\12.0\Common\VBAOff = 1
- **Désactiver les emplacements de confiance si non utilisés**.
  - Retirer au minimum les emplacements modifiables par les utilisateurs, à moins que les utilisateurs doivent écrire des macros sans pouvoir les signer.
- **Utiliser les clés de registre HKLM pour protéger ces paramètres des modifications des utilisateurs.**
  - Alternative: positionner des droits restreints sur les clés HKU
- **Utiliser les GPO pour déployer ces paramètres de sécurité**
  - **Chercher et télécharger « 2007 Office System Administrative Templates »**
  - http://www.microsoft.com/office/ork
  - Alternative: outil OCT, qui ne permet pas de modifier HKLM, uniquement HKU
- Documentation plus complète pour sécuriser un déploiement:
  - http://go.microsoft.com/fwlink/?LinkID=85671

Que certifient les certificats auto-signés ?

- Il est vraiment simple de créer un certificat avec n'importe quel nom…



Pour ~~éviter~~ limiter la fuite d'informations sensibles

- **Utiliser les fonctionnalités d'OpenOffice et Office 2007 pour supprimer les informations cachées.**
- **Remplacer les objets OLE par des images statiques.**
- **Si possible exporter les documents publiés au format PDF ou XPS.**
  - Mais attention: PDF/XPS peuvent toujours cacher des informations…
  - Employer de préférence des imprimantes virtuelles.

Partie 4 :
Comment filtrer les fichiers OpenDocument et Open XML

Antivirus / Filtre de contenu actif

- Un **antivirus** doit analyser tout le contenu du document pour détecter le code malveillant.

- Un **filtre de contenu** est conçu pour supprimer tout contenu actif (macros, scripts, objets, applets…) d'un document.

- **Aujourd'hui chacun de ces outils doit prendre en charge les formats OpenDocument et Open XML.**

Filtrage simple suivant les noms de fichiers Open XML

- À 1ère vue il est très simple de détecter les documents Open XML avec macros d'après leur nom:
  - .doc**x**, .xls**x** and .ppt**x**: OK, pas de macros.
  - .doc**m**, .xls**m**, .ppt**m**: NOK, macros.
- **Mais cela ne s'applique qu'aux macros, pas aux autres contenus actifs comme OLE.**
- Et il suffit de **renommer .docm en .doc**, Office 2007 ouvre le document comme si c'était un .docm…

Un filtre simple avec *zip

- On peut très simplement supprimer les fichiers indésirables de l'archive, par exemple avec zip:
  - **OpenOffice**: `zip -d mydoc.odt` **`Scripts/* Basic/* object*`**
  - **Open XML**: `zip -d mydoc.docm` **`*.bin`**

- …mais attention: zip –d est « **case sensitive** », pas les suites bureautiques !
  - « **sCriPts/*** » ne serait pas supprimé
  - Pour que ca marche il faudrait patcher le code source de zip.

Un autre filtre simple en Python

- (plus lent à cause de la recompression, mais plus sûr)

```
import zipfile, sys
try:
        infile = zipfile.ZipFile(sys.argv[1], "r")
        outfile = zipfile.ZipFile(sys.argv[2], "w")
except:
        sys.exit("usage: %s infile outfile" % __file__)
for f in infile.infolist():
        fname = f.filename.lower()
        if  not fname.startswith("scripts") \
        and not fname.startswith("basic") \
        and not fname.startswith("object") \
        and not fname.endswith(".bin") :
                data = infile.read(f.filename)
                outfile.writestr(f, data)
```

Techniques de camouflage

- But des attaquants: contourner les tests des antivirus et passerelles de filtrage

- Chaque format de fichier peut offrir des fonctionnalités « de camouflage » pour tromper les filtres.
- Exemples pour HTML:
  - Encodage UTF-8 (avec des caractères ASCII illégalement encodés)
  - Inclusion de faux tags de scripts
    - `<SCR<script>remove me</script>IPT>…`

Renommage de macros OpenOffice

- Les fichiers de macro peuvent être renommés avec **n'importe quelle extension**, si manifest.xml et content.xml sont modifiés en fonction.
  - Exemple: Scripts/python/BadMacro**.txt**

- **Conclusion: Un filtre de macros ne doit pas faire confiance aux extensions pour OOo.**
  - Heureusement il suffit de vider les répertoires Scripts et Basic.

Renommage de macros Office 2007

- **A cause de la structure modulaire d'Open XML, il est possible de renommer le conteneur des macros VBA.**
- Exemple pour Word:
  - Renommer vbaProject.bin en toto.txt
  - Mettre à jour word/_rels/document.xml.rels
  - Dans [Content_Types].xml, remplacer "bin" par "txt"
  - …et les macros fonctionnent !

- **=> Antivirus et filtres ne doivent pas s'appuyer sur les noms de fichiers dans documents Open XML !**
  - **Il faut parser le XML ou analyser le contenu.**

Camouflage Open XML « ASCII 7 bit »

- Comme Internet Explorer, Office 2007 traite bizarrement les fichiers XML avec un encodage US-ASCII (7 bits) :
  - **Le 8ème bit de chaque caractère est silencieusement retiré et la lecture continue… !**
  - Pour masquer des tags XML il suffit d'ajouter le "le bit de camouflage":

```
<?xml version="1.0" encoding="us-ascii" standalone="yes"?>
¼HIDDENTAG¾ malware[…] ¼/HIDDENTAG¾
```

  - http://www.securityfocus.com/archive/1/437948

Encodage UTF-7 Open XML

- Il est aussi possible d'utiliser un encodage UTF-7 pour camoufler des tags:

```
<?xml version="1.0" encoding="UTF-7" standalone="yes"?>
+ADw-HIDDENTAG+AD4- malware[…] +ADw-/HIDDENTAG+AD4-
```

- **Pourtant à la lecture des spécifications ECMA, Open XML ne devrait autoriser que UTF-8 ou UTF-16 !**
  - Ni UTF-7 ni US-ASCII
- (OpenOffice semble profiter d'un parseur XML plus strict)

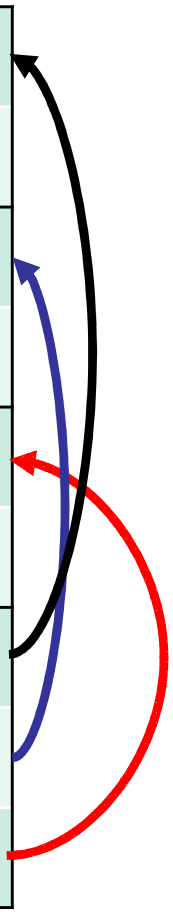
Noms de fichiers dupliqués dans une archive ZIP

- **Les noms de fichiers (et quelques autres informations) sont dupliqués:**
  - Dans chaque **entête de fichier** (avant le contenu compressé du fichier)
  - Dans le **répertoire central**, à la fin du ZIP

- **Problème: certaines applications s'appuient sur le répertoire central, d'autres sur les entêtes…**
  - Très peu d'applications vérifient la cohérence des 2.

Camouflage par ZIP malformé: exemple

Entête Fichier 1	Nom: **Document.xml**, Taille: 4000
Contenu Fichier 1	*(compressé)*
Entête Fichier 2	Nom : **vbaProject.bin**, Taille : 1024
Contenu Fichier 2	*(compressé)*
Entête Fichier 3	Nom : **CodeMalveillant.exe**, Taille : 16000
Contenu Fichier 3	*(compressé)*
Répertoire Central	Fichier 1: **Document.xml**, Taille : 4000
	Fichier 2: **toto.xml**, Taille : 1024
	Fichier 3: **titi.txt**, Taille : **0**



Camouflage par ZIP malformé: OOo et MS Office 2007

- **OpenOffice s'appuie uniquement sur le répertoire central:**
  - Un filtre/antivirus n'utilisant que les noms de fichiers des entêtes ZIP peut être contourné.

- **MS Office 2007 propose la réparation du document s'il détecte une incohérence entre les noms de fichiers ZIP.**
  - …si l'utilisateur confirme, les macros retombent toujours sur leurs pattes !
  - Un filtre/antivirus n'utilisant que les noms de fichiers des entêtes OU du répertoire central ZIP peut être contourné.

Comment bâtir un filtre ou antivirus sûr

- **Utiliser une bibliothèque ZIP robuste avec détection de ZIPs malformés**
  - Comparaison stricte du répertoire central et des entêtes de fichiers
  - Rejeter toute archive illisible ou non supportée (compression Zip64, chiffrement, …)
- **Utiliser un parseur XML complet et robuste**
  - Ne jamais se contenter d'une simple recherche de chaîne ou de motif.
  - N'autoriser que les encodages prévus (UTF-8, …)
  - Rejeter tout caractère invalide pour l'encodage (analyse stricte)
- **Utiliser les schémas XML fournis (XSD, RNG) pour valider les fichiers XML**
- **Rejeter toute anomalie de structure:**
  - Document Open XML nommée .doc, …
- …bien sûr tout cela est plus facile à dire qu'à faire. ;-)

Conclusion

Conclusion

- **OpenDocument et Open XML sont 2 nouveaux formats très prometteurs pour les suites bureautiques :**
  - Spécifications ouvertes, standardisation
  - Technologies répandues ZIP et XML, avec schémas fournis
- **OpenOffice et MS Office 2007 apportent des fonctions de suppression d'informations cachées**.
  - Mais les objets inclus/OLE ne sont pas signalés
  - Chacune propose l'export PDF (nativement ou en option)

- **Cependant il y a toujours de multiples possibilités pour inclure des contenus actifs malveillants dans ces nouveaux formats.**
  - Macros, objets OLE, scripts HTML, Applets Java, ActiveX, …

Conclusion

- **Quelques nouveaux problèmes de sécurité:**
  - Des manipulations **ZIP et XML** peuvent leurrer certains antivirus et filtres de contenu.
  - La fonction **“réparer un document malformé”** peut aussi aider à camoufler du code malveillant.
  - **Le niveau par défaut pour la sécurité des macros Office 2007** est moins strict qu’avant.
  - **Il faudra encore un peu de temps avant que ces nouveaux formats soient pris en charge de facon sûre par les antivirus et logiciels de filtrage.**

Conclusion

- **Grâce à ces formats ouverts, il devrait être plus facile de détecter les contenus actifs dans les documents.**
  - **Développer des filtres semble simple grâce à ZIP et XML.**
  - **Mais pas vraiment, en fait.**

- Quelques faiblesses à corriger pour les développeurs OOo et MS Office, et quelques fonctions de sécurité à ajouter.
  - (parsing ZIP/XML plus strict, blocage OLE, déploiement de configuration sécurisée, …).

Fin

- <? Questions ?>

voir http://actes.sstic.org

et http://lagasoft.free.fr

pour l'article correspondant à cette présentation,

et (peut-être) des informations mises à jour.

Compléments

Comparaison rapide: OpenDocument et Open XML (1)

- 2 formats ouverts, principalement constitués de **fichiers XML dans une archive ZIP**.
- Tous 2 peuvent contenir des **macros**, potentiellement malveillantes, pouvant se déclencher automatiquement.
  - Le niveau de sécurité par défaut permet de lancer des macros non signées.
  - OOo: avec 1 clic avant de voir le document.
  - Office 2007: avec 3 clics après avoir vu le document.
- Tous 2 peuvent contenir des **objets OLE Package** (stockés dans des fichier binaires MS OLE2), et seul l'OS protège l'utilisateur.

Comparaison rapide: OpenDocument et Open XML (2)

- **Open XML paraît beaucoup plus complexe qu'OpenDocument**. (spécifications, relations .rels, fonctionnalités, …)
  - Impact sur l'analyse de sécurité et sur le développement d'antivirus et de filtres.
- **Différents camouflages (ZIP, XML)** sont possibles aujourd'hui pour contourner filtres et antivirus.

OpenOffice - UNO

- UNO peut aussi être appelé par des programmes à l'extérieur des documents (depuis C++, .NET, Java, Python, …)
  - OpenOffice fonctionne alors comme serveur, contrôlé par une application cliente par des appels à UNO.
    - (dépasse le cadre de cette analyse)

Classeurs binaires Excel 2007

- .xls**x** : classeur sans macros (par défaut)
- .xls**m**: classeur avec macros
- **.xls**b**: classeur binaire**
  - Meilleures performances qu'XML pur (.xlsx/.xlsm)
  - Même structure ZIP que .xlsx/.xlsm
  - **Fichiers de données XML remplacés par fichiers binaires** (évolution du format BIFF8)
    - Except relationships (.rels), metadata, …
  - **Peut contenir des macros, ou pas (comme .xlsm)**

Scripts HTML dans les documents Open XML

- **Office 2003** permet de stocker des scripts HTML dans les documents Office (avec l'outil Script Editor)
- Scripts uniquement activés quand le document est **enregistré au format HTML/MHTML**, puis ouvert dans un **navigateur**.
  - Comme OpenOffice.

- **Office 2007** ne semble pas capable de lancer Script Editor.
  - Il est pour l'instant impossible d'inclure des scripts HTML dans les documents Open XML, mais…

- Scripts toujours pris en charge quand un document Office 2003 est sauvé en HTML par Office 2007.

How to sign a trusted macro for Office 2007

- Use « MS Office tools / Digital Certificate for VBA projects » to **create a self-signed certificate**.
  - These certificates also work for OpenOffice.
- Then **sign your trusted macro** with VBA Editor / Tools
- And **approve your certificate**.

How to use an unsigned trusted macro

- If you can't / don't want to sign a macro:
- **Put the document into a trusted location.**
- By default (example for Word):
  - C:\Program Files\Microsoft Office\Templates
    - Writable by administrators and power users only
  - C:\Documents and Settings\user\Application Data\Microsoft\Templates
    - Writable by user only
  - C:\Documents and Settings\user\Application Data\Microsoft\Word\Startup
    - Writable by user only
- **In a corporate environment, it would be wise to trust only admin-writable locations.**
- **And to protect MS-Office security parameters from users: use HKLM registry instead of HKCU.**

Filtre de contenu actif OpenDocument

- Pour retirer le contenu actif d'OpenDocument:
  - **Macros**: Supprimer tout fichier des répertoires "Basic" et "Scripts".
  - **Objets OLE** : Supprimer tout fichier "Object*"
  - Dans "content.xml":
    - Supprimer les **objets OLE** : `<draw:object-ole>`
    - Supprimer les **scripts :** `<text:script>`
    - Supprimer les **applets :** `<draw:applet>`
    - Supprimer les **plugins :** `<draw:plugin>`
    - Mettre à jour tous les tags liés aux macros, notamment les **events:** `<office:event-listeners>`

Filtre de contenu actif Open XML

- Pour retirer le contenu actif d'Open XML:
  - **Macros**: supprimer les fichiers **vbaProject.bin** et **vbaData.xml**
    - (et tous les autres objets vbaProject / vbaData, d'après [Content_Types].xml)
    - Mettre à jour les tags liés aux macros: entryMacro, exitMacro, …
  - **objets OLE** : supprimer tous les fichiers *.bin
    - (et tous les objets oleObject)
  - **Mettre à jour les relations (fichiers .rels)**

.NET API: System.IO.Packaging

- The new System.IO.Packaging API provides Open Packaging Conventions files access to .NET applications.
  - Open XML and XPS documents.
- Allows to develop Open XML security filters.
  - For example, Microsoft provides a VBA removal code snippet.
- But this only works on Windows, whereas many gateways are based on other systems.
- And of course this is not usable for OpenDocument.

OOoPy

- OOoPy is a useful Python package to open and modify OpenDocument files.
  - A simple combination of a ZIP reader/writer with a XML parser.
  - Can be used to handle Open XML files also.

- May be used to design portable security filters.

ZIP compression

- Recently new compression algorithms were added to the ZIP standard.
  - **Open XML specification explicitly allows the use of Zip64 compression.**
- But most free ZIP libraries only support standard Deflate compression.
  - Any unsupported archive should be rejected.

XML schema validation

- OpenDocument and Open XML specifications provide XML schemas (XSD or Relax-NG).
- It should be easy to validate XML content.
- You have to use a validating XML parser, with XML namespaces handling.
- It’s even possible to use modified schemas without unwanted tags.
- But beware of documents with custom schemas or Open XML “Markup compatibility and extensibility” features.