Sesión 3

Laboratorio con WannaCry

Gabriel Martin Blázquez
Grupo 1 Francisco Pinto Santos

Contenido

Introducción

Esta sesión consto de dos partes: una primera en la que se expuso el protocolo de actuación seguido por Deloitte ante un ciberataque a gran escala, tras lo cual se procedió a la segunda parte, la cual consistió en cómo realizar una análisis forense sobre un volcado de memoria principal de un equipo afectado por el ransomware "WannaCry".

Protocolo de actuación ante un ciberataque masivo

En esta primera parte, vimos cómo se actúa ante un ciberataque masivo que afecta a todos o una gran cantidad de los equipos de una empresa.

Para comenzar, como es un proceso que debe tener validez legal, tiene que haber un notario presente y se tienen que tomar actas de todo el proceso, para lo cual en Deloitte tienen una plataforma llamada "DoyFe" que genera las actas automáticamente.

Tras ello, se procede a tomar una muestra de los equipos afectados en dos fases:

- En una primera fase, se toman muestras de la memoria volátil de los equipos afectados que continúen encendidos (RAM, caches, buffers, …).
- En una segunda fase se toman muestras de la memoria secundaria de los equipos afectados, tanto encendidos como apagados.
  Dentro de los equipos apagados, se toman muestras de los equipos que se hayan apagado mediante un protocolo ordenado y los apagados de una manera más brusca (como puede ser tirando del cable).

Una vez hecho esto, se toma una copia de seguridad de cada una de las evidencias tomadas y para trabajar sobre ellas se hacen copias de estas para poder adulterarlas de forma segura sin alterar las evidencias originales ni las copias de las evidencias.

Por último, se nos enseño como se hace el proceso de copia, con maquinas copiadoras especializadas en realizar distintas copias de distintos dispositivos (memoria magnética, semiconductora, …) de una forma muy rápida.
El único problema que tenían estos equipos copiadores es que eran muy caros, delicados y difíciles de conseguir.

Por último, en esta parte, se nos dio el consejo de que, ante un ataque informático, apaguemos nuestros equipos de forma abrupta para facilitar el posterior análisis forense.

Análisis forense del fichero binario del ransomware WannaCry

Un ransomware es un software de carácter malicioso, que cifra los ficheros de un equipo. Tras ello se le indica al usuario que para descifrarlos y recuperar su información tiene que pagar una cantidad de dinero significativa como chantaje.

En este caso, estuvimos tratando con el malware WannaCray, que fue un ransomware utilizado en 2017 para lanzar un ciberataque a escala global, supuestamente realizado por un agente del gobierno norcoreano.
Muchas empresas se vieron afectadas por este malware, pero en España las principales afectadas fueron unas de las empresas mas grandes a nivel nacional, como son Telefónica, Iberdrola y Gas natural, y algunos organismos de carácter público como el CNI.

En cuanto a la manera de actuar de este malware, se establecía en un host y se distribuía por el protocolo SAMBA, utilizado por los equipos Windows para el intercambio de archivos e información en redes locales, y el cual viene activado por defecto en todos los equipos Windows. Debido a esto, el malware se distribuyó fácilmente por varias redes empresariales y de organismos públicos.
Para neutralizar el ataque a nivel mundial, tras varios días de investigación, se descubrió un killswitch, consistente en un domino que cuando respondiera a una petición GET pararía el ataque. No obstante, hasta que se descubrió causo una gran cantidad de pérdidas económicas, considerándose uno de los ciberataques mas destructivos hasta la fecha actual.

Para realizar análisis forense, utilizamos una imagen del propio malware "WannaCry", y un volcado (o dump) de la memoria volátil (RAM) de un equipo afectado por el ransomware WannaCray, sobre el cual realizamos los siguientes pasos:
Primero analizamos el propio malware, ubicado en el fichero `24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c.bin`, con el comando `strings` sobre dicho fichero:

```
strings 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c.bin
```

Tras lo cual procedimos a filtrar los resultados obtenidos con el comando grep:

```
grep [-n] regex
```

Nota: con -n obtienes el número de línea en el que se encuentra el match
Concretamente, filtramos por los siguientes campos:

- En primer lugar, al tratarse de un software, realizamos búsqueda por ficheros compilados y scripts con las extensiones .bat, .o, .class, .pyc, … pero no obtuvimos resultados. Tras lo cual probamos con ficheros de enlazado dinámico (.dll) sin obtener resultado.
- Tras el fracaso anterior, y debido a que sabíamos que se desactivaba mediante una petición http al dominio de killswitch, realizamos una búsqueda con grep por http, tras la cual obtuvimos que la URL que actuaba de killswitch es la siguiente:

```
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
```

- Después, debido a que sabíamos que se transmitía mediante el protocolo samba, buscamos por smb que es el principio de las URL/URIs del protocolo SAMBA. Pero a diferencia de antes, no obtuvimos resultado.
- Por último, seguimos las recomendaciones de Carlos buscando por el nombre del malware WNcry@2017 y por attrib y realizamos nuestro primer descubrimiento.
  Este consistía en que de las líneas 1800 a 18003 del malware, había un conjunto de comandos, entre ellos `1800:attrib +h .`, con el cual se ponían los ficheros en oculto, tras lo cual se procede a cifrarlos.

Análisis forense de RAM de equipo afectado por ransomware WannaCry

Una vez encontrado en el fichero binario del malware WannaCray donde se realiza el cifrado y ocultación de los ficheros, procedimos a analizar el dump de memoria volátil de un equipo afectado por este malware.
Para proceder, utilizamos la herramienta volatility, programada en Python 2.X, la cual es un framework que nos ofrece un set de herramientas para extraer información de dumps de memoria RAM de una gran cantidad de versiones, distribuciones de sistemas operativos, tanto Linux, como Windows y MacOS.

Lo primero que hicimos, fue obtener información del equipo afectado, como sistema operativo, versión, … para proveer dicha información a volatility, y poder hacer así un análisis más exhaustivo. Esto se realizó mediante el comando:

```
python ../volatility-master/volatility-master/vol.py -f malware.raw imageinfo
> volatility_imageinfo.txt
```

tras lo cual, conseguimos saber que el sistema del equipo afectado consistía en un WindowsXP Service pack 3 con una arquitectura de procesador de x86; información que aportamos en el resto de los comandos (con la opción `--profile=WinXPSP2x86`), para poder realizar un análisis más preciso.

Una vez obtenida la información del host, se procedió a investigar sobre los siguientes elementos:

- Se buscaron "apIhooks" sobre el dump, los cuales consisten en eventos significativos del sistema operativo. En estos obtuvimos que solo había uno que era taskdl, el cual se trata de un evento generado por el propio sistema de Windows para gestionar el administrador de tareas, por lo que no se consideró sospechoso.

  ```
  python ../volatility-master/volatility-master/vol.py -k apihooks
  malware.raw
  ```

- Tras esto, se procedió a examinar los procesos activos en el momento del volcado de memoria.

  ```
  python ../volatility-master/volatility-master/vol.py -f malware.raw
  --profile=WinXPSP2x86 pslist > pslist.txt
  ```

  Tras lo cual, vimos unos 20 procesos, entre los cuales se contaban smss.exe, svchost.exe, ctfmon.exe, … y una gran cantidad de servicios del sistema. No obstante, se apreciaba un proceso sospechoso, ya que tenia el nombre del malware a analizar, `@WanaDecryptor@`. Este proceso estaba identificado por el PID 740 y el PPID 1940.

  ```
  0x81fde308 @WanaDecryptor@ 740 1940 2 70 0 0 2017-05-12 21:22:22
  UTC+0000
  ```

- Una vez detectado un proceso sospechoso, procedimos a obtener más información de este. Concretamente a que otros procesos ha afectado, para lo cual utilizamos el siguiente comando:

```
python ../volatility-master/volatility-master/vol.py -f malware.raw
  --profile=WinXPSP2x86 psscan | grep 1940 > salida1940.txt
```

Del cual obtenemos la siguiente salida:

```
0x0000000001f4daf0 taskdl.exe          860   1940 0x199f6000 2017-05-12 21:26:23 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f53d18 taskse.exe          536   1940 0x1986c000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001f69b50 @WanaDecryptor@     424   1940 0x18fa2000 2017-05-12 21:25:52 UTC+0000   2017-05-12 21:25:53 UTC+0000
0x0000000001f8ba58 @WanaDecryptor@     576   1940 0x19671000 2017-05-12 21:26:22 UTC+0000   2017-05-12 21:26:23 UTC+0000
0x0000000001fde308 @WanaDecryptor@     740   1940 0x0de3a000 2017-05-12 21:22:22 UTC+0000
0x0000000002218da0 tasksche.exe       1940   1636 0x0c0a2000 2017-05-12 21:22:14 UTC+0000
```

De la que se pueden obtener dos conclusiones: La primera de ellas es, que WannaCry toma el control del taskdel.exe que se trata del administrador de tareas, y la segunda, debido a los tiempos de ejecución, es que se ejecuta de forma periódica cada 30 segundos aproximadamente.
Esto último, según nos indicó Carlos, es una característica muy común en malware de este tipo, pues una de las primeras cosas que hace, es persistirse, y crear un servicio que ejecute el malware al inicio del sistema y de forma periódica, para hacer aún más daño. En este caso, WannaCray se aprovecha de la invocación periódica del demonio taskdl, para provocar que también se cree el demonio propio de WannaCry.

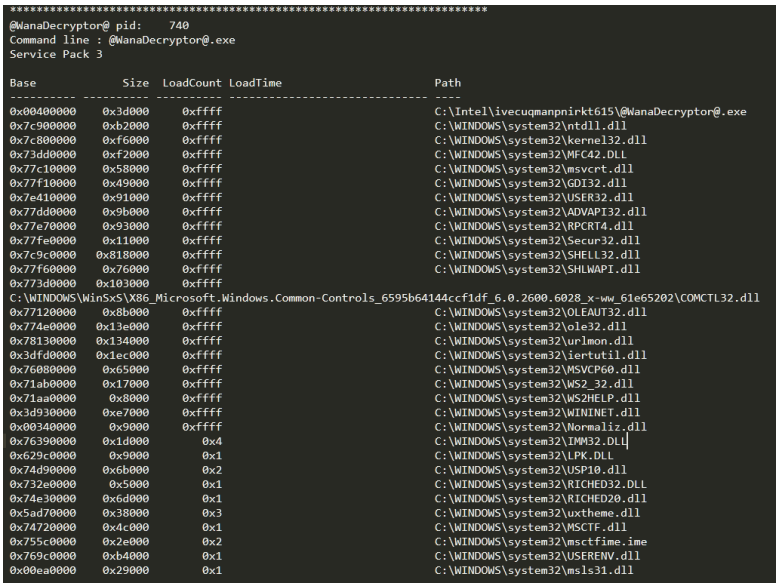
- Una vez conocido esto, intentamos obtener información de que librerías de enlazado dinámico (DLL) utiliza el proceso utilizado por el WannaCry para hacer daño. Para lo cual utilizamos los siguientes comandos:

```
python ../volatility-master/volatility-master/vol.py -f malware.raw
   --profile=WinXPSP2x86 dlllist -p 1940 > dlllist1940.txt
python ../volatility-master/volatility-master/vol.py -f malware.raw
    --profile=WinXPSP2x86 dlllist -p 740 > dlllist740.txt
```

En cuanto a las salidas que se apreciaban de los dos procesos, como se puede ver en las imágenes posteriores, no había ningun DLL que se saliera de lo normal, dado que estaban algunos como ole32.ddll, SMBLIB.dll, USERENV.dll,… que se utilizan por defecto en la mayoría de las aplicaciones de Windows para realizar algunas de las tareas mas básicas. No obstante, había un dll que destacaba `ivecuqmanpnirkt615`, porque no es uno del sistema y se encontraba en ambos procesos.

```
tasksche.exe pid:   1940
Command line : "C:\Intel\ivecuqmanpnirkt615\tasksche.exe"
Service Pack 3

Base             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x00400000   0x35a000     0xffff                                C:\Intel\ivecuqmanpnirkt615\tasksche.exe
0x7c900000    0xb2000     0xffff                                C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf6000     0xffff                                C:\WINDOWS\system32\kernel32.dll
0x7e410000    0x91000     0xffff                                C:\WINDOWS\system32\USER32.dll
0x77f10000    0x49000     0xffff                                C:\WINDOWS\system32\GDI32.dll
0x77dd0000    0x9b000     0xffff                                C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x93000     0xffff                                C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000     0xffff                                C:\WINDOWS\system32\Secur32.dll
0x77c10000    0x58000     0xffff                                C:\WINDOWS\system32\MSVCRT.dll
0x76390000    0x1d000        0x1                                C:\WINDOWS\system32\IMM32.DLL
0x629c0000     0x9000        0x1                                C:\WINDOWS\system32\LPK.DLL
0x74d90000    0x6b000        0x1                                C:\WINDOWS\system32\USP10.dll
0x77b40000    0x22000        0x1                                C:\WINDOWS\system32\Apphelp.dll
0x77c00000     0x8000        0x1                                C:\WINDOWS\system32\VERSION.dll
0x68000000    0x36000        0x1                                C:\WINDOWS\system32\rsaenh.dll
0x7c9c0000   0x818000        0x1                                C:\WINDOWS\system32\SHELL32.dll
0x77f60000    0x76000        0x3                                C:\WINDOWS\system32\SHLWAPI.dll
0x773d0000   0x103000        0x2
C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.6028_x-ww_61e65202\comctl32.dll
0x76080000    0x65000        0x1                                C:\WINDOWS\system32\MSVCP60.dll
0x77690000    0x21000        0x1                                C:\WINDOWS\system32\NTMARTA.DLL
0x774e0000   0x13e000        0x1                                C:\WINDOWS\system32\ole32.dll
0x71bf0000    0x13000        0x1                                C:\WINDOWS\system32\SAMLIB.dll
0x76f60000    0x2c000        0x1                                C:\WINDOWS\system32\WLDAP32.dll
0x769c0000    0xb4000        0x1                                C:\WINDOWS\system32\USERENV.dll
0x5ad70000    0x38000        0x2                                C:\WINDOWS\system32\uxtheme.dll
```



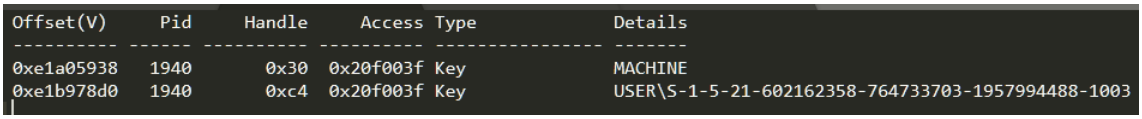
```
**********************************************************************
@WanaDecryptor@ pid:    740
Command line : @WanaDecryptor@.exe
Service Pack 3

Base             Size  LoadCount LoadTime                   Path
---------- ---------- ---------- ------------------------------ ----
0x00400000    0x3d000     0xffff                              C:\Intel\ivecuqmanpnirkt615\@WanaDecryptor@.exe
0x7c900000    0xb2000     0xffff                              C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf6000     0xffff                              C:\WINDOWS\system32\kernel32.dll
0x73dd0000    0xf2000     0xffff                              C:\WINDOWS\system32\MFC42.DLL
0x77c10000    0x58000     0xffff                              C:\WINDOWS\system32\msvcrt.dll
0x77f10000    0x49000     0xffff                              C:\WINDOWS\system32\GDI32.dll
0x7e410000    0x91000     0xffff                              C:\WINDOWS\system32\USER32.dll
0x77dd0000    0x9b000     0xffff                              C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x93000     0xffff                              C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000     0xffff                              C:\WINDOWS\system32\Secur32.dll
0x7c9c0000   0x818000     0xffff                              C:\WINDOWS\system32\SHELL32.dll
0x77f60000    0x76000     0xffff                              C:\WINDOWS\system32\SHLWAPI.dll
0x773d0000   0x103000     0xffff
C:\WINDOWS\WinSxS\X86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.6028_x-ww_61e65202\COMCTL32.dll
0x77120000    0x8b000     0xffff                              C:\WINDOWS\system32\OLEAUT32.dll
0x774e0000   0x13e000     0xffff                              C:\WINDOWS\system32\ole32.dll
0x78130000   0x134000     0xffff                              C:\WINDOWS\system32\urlmon.dll
0x3dfd0000   0x1ec000     0xffff                              C:\WINDOWS\system32\iertutil.dll
0x76080000    0x65000     0xffff                              C:\WINDOWS\system32\MSVCP60.dll
0x71ab0000    0x17000     0xffff                              C:\WINDOWS\system32\WS2_32.dll
0x71aa0000     0x8000     0xffff                              C:\WINDOWS\system32\WS2HELP.dll
0x3d930000    0xe7000     0xffff                              C:\WINDOWS\system32\WININET.dll
0x00340000     0x9000     0xffff                              C:\WINDOWS\system32\Normaliz.dll
0x76390000    0x1d000        0x4                              C:\WINDOWS\system32\IMM32.DLL
0x629c0000     0x9000        0x1                              C:\WINDOWS\system32\LPK.DLL
0x74d90000    0x6b000        0x2                              C:\WINDOWS\system32\USP10.dll
0x732e0000     0x5000        0x1                              C:\WINDOWS\system32\RICHED32.DLL
0x74e30000    0x6d000        0x1                              C:\WINDOWS\system32\RICHED20.dll
0x5ad70000    0x38000        0x3                              C:\WINDOWS\system32\uxtheme.dll
0x74720000    0x4c000        0x1                              C:\WINDOWS\system32\MSCTF.dll
0x755c0000    0x2e000        0x2                              C:\WINDOWS\system32\msctfime.ime
0x769c0000    0xb4000        0x1                              C:\WINDOWS\system32\USERENV.dll
0x00ea0000    0x29000        0x1                              C:\WINDOWS\system32\msls31.dll
```

- Una vez detectado en que DLL se podría encontrar el código malicioso instalado en el equipo, procedimos a buscar las funciones manejadoras (handles) de los procesos sospechosos, para las DLL que utilizan, en concreto, para la DLL sospechosa (`ivecuqmanpnirkt615`), mediante el siguiente comando:

```
python ../volatility-master/volatility-master/vol.py -f malware.raw
--profile=WinXPSP2x86 handles -p 1940 -t Key > handles1940.txt
```

En la salida de dicho comando, según las indicaciones de Carlos, se podía apreciar un UserID de Windows en la última línea al final:



```
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0xe1a05938   1940       0x30  0x20f003f Key              MACHINE
0xe1b978d0   1940       0xc4  0x20f003f Key              USER\S-1-5-21-602162358-764733703-1957994488-1003
```

No obstante, no se podía obtener mucha información, por lo que procedimos a buscar handles, pero con más detalles como que ficheros de memoria secundaria estaban utilizando y alterando los procesos sospechosos a estudio (1940 y 740):

```
python ../volatility-master/volatility-master/vol.py -f
malware.raw --profile=WinXPSP2x86 handles -p 1940 -t File >
handles1940File.txt
python ../volatility-master/volatility-master/vol.py -f
malware.raw --profile=WinXPSP2x86 handles -p 1940 -t Mutant >
handles1940Mutant.txt
python ../volatility-master/volatility-master/vol.py -f
malware.raw --profile=WinXPSP2x86 handles -p 740 -t File >
handles740File.txt
python ../volatility-master/volatility-master/vol.py -f
malware.raw --profile=WinXPSP2x86 handles -p 740 -t Mutant >
handles740Mutant.txt
```

De estos, obtuvimos mucha información, pero la mas importante se encuentra en la imagen posterior, resultado de buscar que ficheros estaba alterando el proceso 1940 (taskdl), en la que vemos que esta alterando el DLL en el que se sospecha que se

encuentra el código malicioso. Gracias a esto, podemos confirmar que WannaCry lo primero que hace es tomar control del proceso taskdl y persistirse, tras lo cual procede a realizar rutinas periódicas de ocultado y encriptado de ficheros cada 30 segundos.

```
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- --------------- -------
0x81fbce00   1940        0xc   0x100020 File
\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.6028_x-ww_61e65202
0x82233f18   1940       0x34   0x100020 File            \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615
0x822386a8   1940       0x48   0x100001 File            \Device\KsecDD
0x823a0cd0   1940       0x50   0x100020 File
\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.6028_x-ww_61e65202
```

- Una vez conocido como actúa el proceso daemon de WannaCry cuando ya está creado, procedemos a revisar los registros del sistema, para conocer como es el proceso de arranque de WannaCry al inicio del sistema:

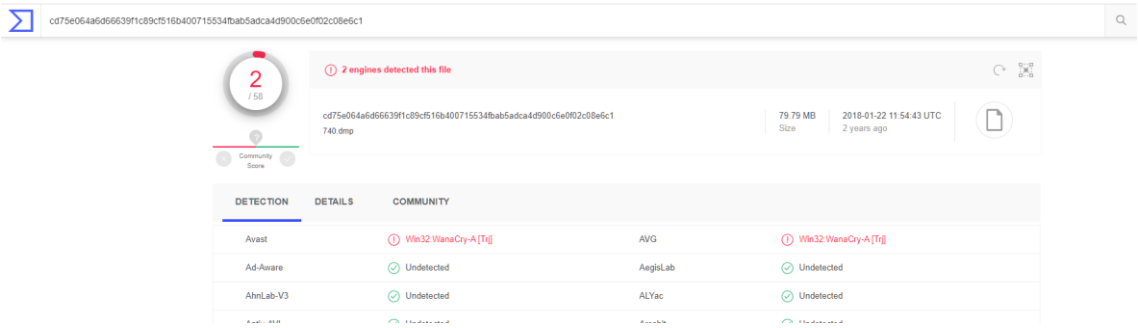
```
python ../volatility-master/volatility-master/vol.py -f malware.raw
                --profile=WinXPSP2x86 printkey -K
        "Microsoft\Windows\CurrentVersion\Run" > printkey.txt
```

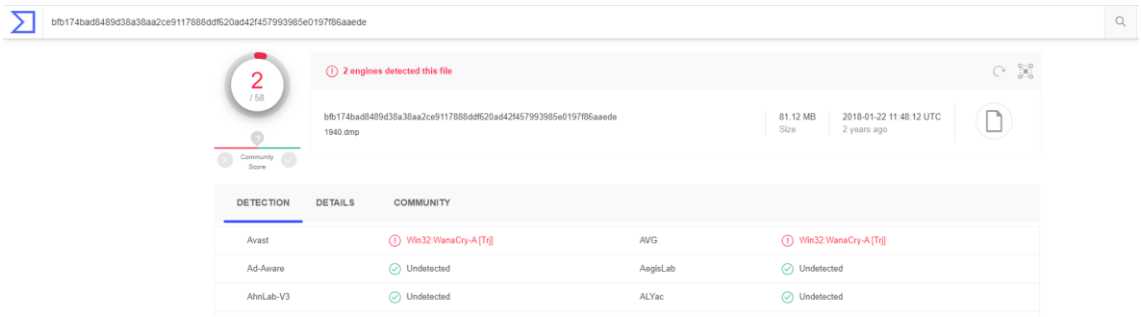
  Con el cual, confirmamos nuestras sospechas, y en la salida se aprecia como el malware WannaCry ha creado un registro que hace que se inicie un proceso de WannaCry al comienzo del sistema, haciendo que se dispare el mecanismo de tomar el proceso taskdl (gestor de tareas de Windows) y modificarlo para que lance de forma periódica el proceso de WannaCryDecriptor que va tomando distintos ficheros y los va ocultando y cifrando, a menos que se haya registrado el dominio de killswitch (mencionado anteriormente).

Una vez que conocemos el funcionamiento de este malware, mediante el siguiente comando, extrajimos las imágenes de los procesos 740 y 1940, que son los implicados en el WannaCry como hemos visto:

```
python ../volatility-master/volatility-master/vol.py -f malware.raw
      --profile=WinXPSP2x86 memdump -p 1940,740 -D memdump/
```

Estas imágenes de procesos, según nos indico Carlos, podemos confirmar que se tratan de los procesos utilizados para atacar por WannaCry si los subimos a la web virustotal.com y lo corroboramos.
Tras realizar este proceso, como se pude ver en las capturas posteriores, pudimos ver que efectivamente eran dos de los procesos utilizados por WannaCry para realizar el ataque:





Por último, Carlos nos dijo que podíamos ver los ficheros que habían sido cifrados por WannaCry en la imagen del equipo afectado, buscando que ficheros concretos habían modificado los dos procesos sospechosos a estudio:

```
python ../volatility-master/volatility-master/vol.py -f malware.raw --
        profile=WinXPSP2x86 filescan | grep ivecuqmanpnirkt615 >
                        filescanivecuqmanp.txt
```

```
0x0000000001f871a0      1      0 R--rw- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\@WanaDecryptor@.exe
0x0000000001fb17a8      1      0 R--r-d \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\@WanaDecryptor@.exe
0x0000000001fb2278      1      0 R--r-d \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\taskse.exe
0x0000000001fbcef8      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\u.wnry
0x000000000209dbe8      1      0 -W-r-- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\00000000.res
0x000000000209de48      1      0 R--r-- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\b.wnry
0x00000000021d8ac0      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\s.wnry
0x00000000021dc028      1      0 R--r-d \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\taskdl.exe
0x00000000021f3870      1      0 R--rw- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\tasksche.exe
0x000000000220ec40      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_turkish.wnry
0x0000000002212028      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_russian.wnry
0x0000000002217528      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_spanish.wnry
0x0000000002219b30      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_slovak.wnry
0x0000000002229748      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_vietnamese.wnry
0x0000000002232418      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_swedish.wnry
0x0000000002233f18      1      1 R--rw- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615
0x00000000022456e0      1      1 R--rw- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615
0x0000000002256c88      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\t.wnry
0x00000000022bb7f8      1      0 R--r-- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\00000000.pky
0x00000000022c72b0      1      0 R----- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_english.wnry
0x00000000022d2f28      1      0 R--r-d \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\tasksche.exe
0x00000000022ec718      1      0 R--rw- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\c.wnry
0x00000000022f06f8      1      0 -W---- \Device\HarddiskVolume1\Intel\ivecuqmanpnirkt615\msg\m_romanian.wnry
```

Por último, Carlos nos comento que este proceso suele ir de la mano con un análisis de los distintos mensaje enviados por la red y de las estructuras de memoria de los procesos implicados en las comunicaciones de red de los equipos, para poder hacer un timeline más detallado del proceso de expansión del malware.

Fuentes

- https://www.virustotal.com/gui/
- https://es.wikipedia.org/wiki/WannaCry
- https://github.com/volatilityfoundation/volatility