Searching for IOCs in Forensic Evidence

Santiago Trigo, Ingeniero, Ariel Podestá, Ingeniero, Gonzalo Ruiz de Angeli, Mg. Ingeniero, Bruno Constanzo, Ingeniero, Hugo Curti, Mg. Ingeniero, Juan Ignacio Alberdi, Ingeniero, Martin Castellote, Ingeniero, Ana Haydée Di Iorio, Esp. Ingeniera
Universidad FASTA, Argentina,
{santiagotrigo, ruizgon, bconstanzo, apodesta, hcurti, ignacio_alberdi, mcastellote, diana}@ufasta.edu.ar

Abstract– ***The rise of digital crime as an inevitable consequence of the transversality of technology in all aspects of life has generated until today -and will keep doing so in the future- the need for the Justice to have adequate tools to give answers to society. Digital Forensics is the branch of forensics sciences that provides the means to find solutions in crimes where technology takes a key role, be it as a method, mean or end. One of the biggest challenges in this discipline is when potentially unknown malware is involved in the case. As malware can be varied in characteristics and the threats it poses, its analysis is difficult, and drawing conclusions challenging. For this reason, it is imperative to have a guideline that provides a valid framework to act upon and analyze digital evidence originating from a malware infected device and obtain conclusive indicators that enrich the experts witness work.***

Keywords—digital forensics, malware analysis, information security.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2020.1.1.647
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Searching for IOCs in Forensic Evidence

Santiago Trigo, Ingeniero, Ariel Podestá, Ingeniero, Gonzalo Ruiz de Angeli, Mg. Ingeniero,
Bruno Constanzo, Ingeniero, Hugo Curti, Mg. Ingeniero, Juan Ignacio Alberdi, Ingeniero,
Martin Castellote, Ingeniero, Ana Haydée Di Iorio, Esp. Ingeniera
Universidad FASTA, Argentina,
{santiagotrigo, ruizgon, bconstanzo, apodesta, hcurti, ignacio_alberdi, mcastellote, diana}@ufasta.edu.ar

***Abstract*– The rise of digital crime as an inevitable consequence of the transversality of technology in all aspects of life has generated until today -and will keep doing so in the future- the need for the Justice to have adequate tools to give answers to society. Digital Forensics is the branch of forensics sciences that provides the means to find solutions in crimes where technology takes a key role, be it as a method, mean or end. One of the biggest challenges in this discipline is when potentially unknown malware is involved in the case. As malware can be varied in characteristics and the threats it poses, its analysis is difficult, and drawing conclusions challenging. For this reason, it is imperative to have a guideline that provides a valid framework to act upon and analyze digital evidence originating from a malware infected device and obtain conclusive indicators that enrich the experts witness work.**

Keywords—digital forensics, malware analysis, information security.

I. Introducción

El *malware* moderno ha evolucionado haciendo cada vez más difícil la detección de su presencia en un sistema, por ejemplo, por parte de software antivirus. Esto se ha logrado, en parte, debido a las técnicas que estos han adoptado para introducirse en los sistemas sin ser detectados y obtener así permisos de *root* o de administrador.

En el contexto de la informática forense, resulta esencial poder contar con una metodología estricta que garantice la correcta documentación del análisis realizado y del camino recorrido para la obtención de los respectivos resultados, y obtener indicadores de diferentes *malwares* reales a través del tiempo. Para esto, se debe preparar el ambiente necesario en cada caso, garantizando que sea seguro y brinde las condiciones para que el software malicioso se ejecute, se exponga y quede susceptible a su análisis en base a los indicadores de compromiso o IOCs[1] obtenidos.

En la actualidad la mayoría de estos softwares maliciosos sólo dejan rastros en memoria principal o en la red, haciendo cada vez más dificultoso encontrar indicadores de su presencia en medios persistentes como los discos rígidos más que la mera afectación de sus archivos, como es el caso de un *ransomware*[2], por ejemplo. Por este motivo, para poder detectar de manera exitosa la presencia de *malware* en un sistema, el análisis de memoria principal y de tráfico de red se convierten en la respuesta, sin dejar atrás otros indicadores que pueden provenir del análisis del archivo de configuración del Sistema Operativo, del sistema de archivos y otros indicadores basados en la utilización de los recursos del equipo.

Todo *malware* supone un tipo de amenaza, más aún cuando se desconocen sus características y su objetivo. El nivel de daño y su forma de operar son las incógnitas a esclarecer en este tipo de estudios. Dado que su sola ejecución implica un riesgo para la infraestructura informática circundante, es verdaderamente necesario contar con un entorno controlado y aislado de cualquier dispositivo que no tenga un propósito específico para el análisis. Pero cabe reparar en el concepto de "daño" aquí mencionado. Daño no sólo es cuando se ve afectada la integridad de los datos o información dentro de un sistema, sino también cuando se ven afectadas la confidencialidad de los datos o la disponibilidad de servicios. Es por esta razón, que al establecer los pilares de la Seguridad Informática, área que protege la información que circula o se almacena a través de un sistema informático, se enfatiza en que intenta asegurar las tres propiedades de la información o dato: la integridad, disponibilidad y confidencialidad y si algo o alguien afecta alguna de estas propiedades, se habla de daño.

Este trabajo tiene como objetivo presentar una metodología que colabore en el análisis forense de un dispositivo informático, con el fin de determinar la presencia de malware dentro del mismo. No forma parte de su objetivo analizar malwares en particular para conocer su funcionamiento, ya que esta tarea está más orientada a la Seguridad Informática, sino que, por el contrario, se orienta a desarrollar una metodología que permita a un informático forense detectar rastros que den indicios de la presencia de software malicioso, cualquiera sea su tipo.

II. Marco Teórico

La aplicación forense de las ciencias informáticas tiene por objetivo es extraer información que pueda resultar útil en una investigación judicial, con el fin de presentarla como prueba en un juicio. Este dato digital, útil para la investigación, es denominado evidencia digital[1].

En la realización de su tarea, el informático forense puede encontrarse ante la situación de tener que determinar, por pedido de un juez o fiscal, si un equipo en particular a peritar

[1] IOC, del inglés, *Indicator of Compromise*.

[2] Se denomina así a los *malwares* que cifran información y exigen el pago de un "rescate" por su recuperación.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2020.1.1.647
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

puede contener *malware*. Y esta comprobación debe realizarse, respetando los principios forenses y resguardando, por supuesto, la seguridad informática del laboratorio forense.

Ante el indicio de una posible presencia de *malware*, es importante que el perito informático pueda contar con el código de los softwares maliciosos que se presume puedan haber infectado el equipo y realizar las pruebas correspondientes. Esta situación es impracticable, tanto por la cantidad de *malware* existente, cómo por la imposibilidad de contar con recursos para realizar este tipo de tareas en los laboratorios forenses.

Para la realización de este estudio se seleccionaron cuatro *malwares* contemporáneos cuyo comportamiento es conocido con el fin de abarcar diferentes tipos de afectaciones a los recursos del sistema, sobre todo a la información.

Malwares seleccionados:

1. *Eternalblue*[11]: Es un *exploit*[3] que se aprovecha de una vulnerabilidad en el servicio SMB[4] de ciertos sistemas operativos de Microsoft Windows para obtener una puerta trasera (*backdoor*) con el objetivo de acceder remotamente al equipo con permisos de administrador.
2. *WannaCry*[7]: Es un *ransomware* que adquirió una gran importancia en Mayo de 2017 ya que utilizaba el *exploit* anterior no sólo para infectar el dispositivo dónde se ejecutó, sino también para expandirse a los equipos conectados por red.
3. *Locky*[12]: Es un *ransomware* que fue distribuido principalmente por correo electrónico mediante un archivo de la aplicación Word de Microsoft Office. Dicho archivo contiene macros que ejecutan el código malicioso.
4. *TeslaCrypt*[11]: Es un *ransomware* diseñado para atacar archivos de juegos como perfiles de usuario guardados, sesiones, etc.

Estos *malwares*, con distintas características y diferentes comportamientos, permiten realizar experimentos variados. La información obtenida resulta de mayor utilidad al entender distintos escenarios de ataques y diferentes indicadores a analizar. Además, estos ejemplos, dejan rastros variados que permiten al investigador analizar y observar el comportamiento del equipo infectado desde distintas perspectivas. Algunos de ellos, requieren necesariamente adquirir un volcado de memoria o un volcado del tráfico de red durante la infección; otros, requieren de la observación de cambios en los indicadores de utilización de los recursos del equipo, el estudio de las entradas de registro o rastros en el sistema de archivos.

[3] Exploit es un programa o código que se aprovecha de un agujero de seguridad (vulnerabilidad) en una aplicación o sistema, de forma que un atacante podría usarla en su beneficio [5].

[4] Server Message Block es un protocolo de red que permite compartir archivos, impresoras y puertos serie entre equipos.

III. Metodología para la Construcción de Indicadores

La metodología aplicada en este trabajo pretende brindar los lineamientos para poder llevar adelante experimentos con software malicioso en ambientes seguros y donde el *malware* se ejecute de manera completa para que, a partir de la adquisición de evidencias y la observación de lo sucedido en el sistema, sea posible construir indicadores de la presencia de éste.

De esta manera, se proponen diferentes fases que abarcan desde la preparación del ambiente hasta el análisis de resultados de las pruebas. Dichas fases, tienen un orden cronológico y se presentan a continuación:

1. **Selección de *malware* a analizar:** en esta fase, se realiza la selección de los *malwares* a analizar. Se debe realizar un estudio de los mismos para entender cuáles son sus características, sus estrategias de ataque y objetivos, para tener en claro, por ejemplo, cuáles son los riesgos a la hora de realizar las pruebas, si necesita Internet o no para ejecutarse, si un entorno virtual puede inhibir su ejecución, entre otras cosas. Esta fase definirá el ambiente necesario para las pruebas o experimentos. También se debe contemplar, a raíz del análisis del *malware*, los recursos que en su ejecución son afectados. Es importante ejecutar *malwares* que utilizan los recursos más importantes de un dispositivo, como los archivos alojados en medios persistentes, la memoria principal y la red. Esto a los fines de poder evaluar el uso de estos recursos al momento de infectar un dispositivo y cómo es la afectación de los mismos para poder tomar muestras y obtener los indicadores adecuados.
2. **Confección de los casos de prueba.** Es la fase de diseño de los casos de prueba a realizar para recolectar datos relevantes. Resulta de gran importancia hacer un diseño previo de lo que se va a probar para garantizar que las pruebas sean exitosas. Permite optimizar los tiempos y aprovechar todas las oportunidades de adquisición y observación que existan durante los experimentos en curso. Para definir un caso de prueba, se propone documentar:
    a. Identificador de la Prueba.
    b. Descripción.
    c. Fecha.
    d. Participantes.
    e. *Samples* utilizados.
    f. Procedimiento.
    g. Indicadores del sistema evaluados.
    h. Resultados obtenidos.
3. **Preparación de ambiente seguro** y con las condiciones para que el/los *malware/s* se ejecuten y expongan su comportamiento para estudiarlo. En esta

fase se diseña e implementa el ambiente. Se define si debe ser una máquina real aislada, una máquina virtual en red con otra máquina virtual con acceso a Internet: lo que sea necesario, pero con las configuraciones de cortafuegos (*firewall*) y enrutamiento necesarias para que el *malware* se ejecute sin amenazar a otros equipos. El ambiente, debe proveer lo necesario en función a las características del *malware* previamente estudiado y a los casos de prueba que se diseñaron, sin embargo, siempre se debe ser sumamente precavido respecto a impactos potenciales, por lo que se deben bloquear conexiones a máquinas locales, si las hubiera.

Se proponen tres escenarios al momento de plantear la preparación del ambiente:

a. Preparación de ambiente sobre Máquina física: escenario más cercano a las condiciones sobre las cuales el *malware* se pueda desenvolver con naturalidad.
   Características y actividades a llevar adelante:
   i. Conexión a red aislada físicamente del resto en el área local.
   ii. *Router* versátil y susceptible a realizar diferentes operaciones como por ejemplo, registrar el tráfico de red y bloquear las conexiones que ameriten.
   iii. Imagen forense del sistema previamente obtenida, tomada como punto de partida.
   iv. Configuración de Auditoría que corresponda. Por ejemplo registro de eventos de modificaciones en ciertas carpetas del sistema.
   v. Carga de software de obtención de memoria principal.
   vi. Ejecución de software de visión de llamadas al sistema por proceso. Por ejemplo: Procmon de Sysinternals.
b. Preparación de ambiente sobre Máquina virtual: escenario seguro y flexible para la experimentación. Características y actividades a llevar adelante:
   i. *Snapshot* previo de la máquina virtual para tomar como punto de partida.
   ii. Configuración de entorno virtual de red, desconectado del área local, con firewall y captura de tráfico.
   iii. Inicio en modo *debug*.
   iv. Configuración de Auditoría que corresponda. Por ejemplo registro de eventos de modificaciones en ciertas carpetas del sistema.
   v. Ejecución de software de visión de llamadas al sistema por proceso. Por ejemplo Procmon de Sysinternals.
c. Escenario combinado: máquinas físicas combinadas con máquinas virtuales.

4. **Ejecución de las pruebas.** La fase de ejecución de pruebas es una fase íntegramente experimental. Aquí, se levantará el ambiente de pruebas y se ejecutará los *malware* elegidos. Aunque parezca trivial, es importante resaltar que, en el caso de contar con más de un *malware* para estudiar, es deseable que las pruebas se realicen por separado, en ambientes diferentes para cada uno de ellos. De esta manera, se tendrá mayor control sobre las pruebas.
5. **Adquisición.** Una vez realizadas las pruebas o durante las mismas, es necesario adquirir diferentes evidencias y/o realizar observaciones sobre lo que sucede en el ambiente infectado. Se pueden identificar diferentes oportunidades en este sentido:
   a. Adquisición de datos para analizar:
      i. Volcados de memoria principal: una buena práctica es realizar un volcado de memoria principal antes y uno posterior a la ejecución del *malware* para hacer eventuales comparaciones
      ii. Obtención de paquetes de red en .pcap.
      iii. Información sobre el registro del Sistema Operativo como, por ejemplo: Registro de Windows –*Registry*.
   b. Observación: observación natural de lo que sucede en el sistema. Observación de los indicadores de utilización de los recursos del equipo (procesador, memoria, disco, red). Tomar nota de todo lo ocurrido, por ejemplo, si se observa cambio de extensión de los archivos, movimientos de los mismos, archivos borrados, cartel de notificación de infección o de pedido de recompensa (como sucede con los *ransomware*), etc.
   c. *Screenshots*: realizar capturas de pantalla para documentar todos aquellos eventos que no puedan documentarse de ninguna otra manera. Es de utilidad, también, para poder mostrar de manera gráfica lo que fue sucediendo en el sistema a medida que la prueba fue avanzando.
   d. Logs: Este paso consiste en el análisis de los registros de eventos que se hayan generado durante la ejecución de las pruebas. Tales registros pueden provenir de herramientas

propias del Sistema Operativo (ejemplo: "*Event Viewer*") como otras de terceros, previamente configuradas en el equipo (ejemplo: procmon de SysInternal [8][9]).

6. **Análisis:**
    a. De lo adquirido: para el análisis de las evidencias obtenidas, será necesario utilizar diferentes herramientas específicas en cada caso.
        i. Análisis forense de memoria principal: volcado de memoria, pagefile y archivo de hibernación. Es deseable utilizar herramientas que brinden funcionalidad para analizar cada uno. Por ejemplo, el volcado en crudo puede ser muy difícil de analizar de manera manual, sin un procesamiento correcto de los datos, un conocimiento de cuestiones de diseño de la administración de memoria de los sistemas operativos y de los artefactos en memoria y sus estructuras. Una herramienta para poder realizar esto es *Volatility Framework* [5][6]. El análisis de memoria implica analizar:
            1. Procesos en ejecución.
            2. Procesos terminados.
            3. Procesos ocultos.
            4. DLLs cargadas por los procesos.
            5. Conexiones abiertas por los procesos.
            6. Entradas de registro cargadas en memoria.
        ii. Análisis de captura de paquetes de red: suele ser de mucha utilidad para intentar conocer los equipos y dominios con los que el *malware* se conecta hacia afuera (Internet) y detectar su comportamiento.
    b. De binarios del software malicioso: en muchas ocasiones, se pueden obtener datos de importancia sobre el funcionamiento del *malware*, realizando ingeniería inversa sobre el archivo binario del *malware*. Por ejemplo, se pueden obtener las DLLs utilizadas y las funciones o métodos utilizadas de cada una de ellas para luego, contrastar o comparar con las que se fueron utilizando y que aparecen en el volcado de memoria principal.
    c. Del dispositivo "muerto": Una vez ejecutado el *malware* y verificado que esta ejecución haya sido exitosa, es posible buscar evidencias de vestigios que él *mismo* haya dejado de manera persistente en el sistema. Se debe analizar primeramente si los archivos del sistema han cambiado su extensión. En el caso de un *ransomware*, este cambio de extensión nos brindará información sobre el tipo de *ransomware* que es. También es importante analizar si el *malware* ha dejado copias temporales en alguna parte del sistema, para poder analizar y visualizar su contenido. Muchos *malwares*, como lo es un *exploit*, una vez ejecutados, podrían generar conexiones remotas desde Internet y el visor de sucesos o la captura de paquetes de red mencionada en el punto 5, nos podría brindar información sobre el origen del ataque remoto.
7. **Construcción de indicadores.** El objetivo de esta metodología es establecer un mecanismo que permita determinar si un sistema fue intervenido por un *malware*, es decir, si hubo daño informático. Se proponen indicadores que, ya sea individualmente, o en conjunto, estiman una probabilidad de la presencia de *malware* en el dispositivo.
   Por ejemplo, un indicador asociado a los procesos puede ser:

$$\mathbf{IProc = TP - PC - S}$$

Donde:
IProc: Indicador de Procesos.
TP: Total de Procesos en ejecución en el sistema.
PC: Procesos conocidos.
S: Servicios en ejecución.

Una propuesta que se desprende de la construcción de indicadores es la de asignar un peso a cada uno de ellos. Para ello, se propone contar con un vector normalizado (Vp) de pesos, donde para cada indicador hay un valor entre 0 y 1 para indicar la relevancia del indicador. Por otro lado, se construye un vector de evaluación para un escenario determinado (Ve), que indica los valores para cada indicador en ese escenario. Para obtener el indicador final, se realiza un producto escalar de los vectores Vp y Ve. La correcta determinación del vector de pesos será en función al análisis de diferentes *malwares*, requiriendo acumular experiencias para

poder determinar los pesos óptimos para cada indicador, donde aquel de mayor peso será el que determine con mayor certeza la presencia de software malintencionado. De esta manera, poder enfocar el análisis en aquellos que mayor peso tengan, dado que serán más concluyentes:

$$\mathbf{Vp = P1 + P2 + \ldots + Pn}$$
$$\mathbf{Ve = I1 + I2 + \ldots + In}$$

Dónde:
Vp: Vector de pesos normalizados
P1: Peso normalizado del indicador 1
P2: Peso normalizado del indicador 2
Pn: Peso normalizado del indicador n
Ve: Vector de evaluación
I1: Indicador 1
I2: Indicador 2
In: Indicador n

Entonces, el indicador global es igual a Vp.Ve:

$$\mathbf{Ig = Vp.Ve = P1.I1 + P2.I2 + \ldots + Pn.In}$$

Donde:
Ig: Indicador global para el escenario
Vp: Vector de pesos normalizados
Ve: Vector de evaluación del escenario

Si bien es posible contar con un listado de indicadores preliminar que sirvan como soporte para el estudio del *malware* y el análisis, el objetivo es construir nuevos indicadores e ir robusteciendo el listado de indicadores de compromisos posibles (IoC).
Como se mencionó anteriormente, este trabajo está orientado al análisis forense y los indicadores que se pretenden adquirir están orientados a ese fin exclusivamente. No obstante, en determinadas ocasiones, no es posible tomar ciertos indicadores donde, en la mayoría de los casos, los equipos infectados han sido apagados y a través de su análisis se debe determinar la presencia de algún tipo de *malware*. Si se toma como referencia el ejemplo mencionado en este apartado, si no se cuenta con una adquisición de memoria principal, este indicador será muy complejo de determinar. Para ello, el análisis forense requerirá, de alguna manera, en primera instancia, tomar todas las medidas necesarias para asegurar la evidencia del mismo y tratar, por ejemplo, de encender ese dispositivo nuevamente para intentar determinar si existen procesos desconocidos como se menciona en el ejemplo aquí dado. Si esta opción, no fuese posible de realizar, se deberá examinar el dispositivo en busca de algún archivo que podría sospecharse como malicioso, extraerlo y luego realizar una prueba de laboratorio como la que aquí se describe. De esta manera, podrían obtenerse la mayor cantidad de indicadores posibles y determinar la presencia o no de *malware* en un dispositivo. En el caso de que ciertos indicadores no se puedan obtener, se evaluarán los indicadores con los que se cuenta con el objetivo de determinar si estos indicadores podrían o no dar como resultado la presencia de algún software malicioso dentro del dispositivo. Es importante aclarar que, a mayor cantidad de indicadores, mayor será la eficacia del trabajo realizado.

8. **Conclusiones.** Finalmente, en función a las pruebas realizadas, el análisis y los indicadores obtenidos, se realizan conclusiones respecto al grado de éxito de las pruebas y la cantidad y robustez de los indicadores que se pudieron obtener.

De esta manera, la metodología propone un proceso de un total de 8 fases, dentro de las cuales se plantean diferentes actividades clave para poder llevar adelante un proceso exitoso de estudio de *malware* en ejecución para la construcción de IoCs.

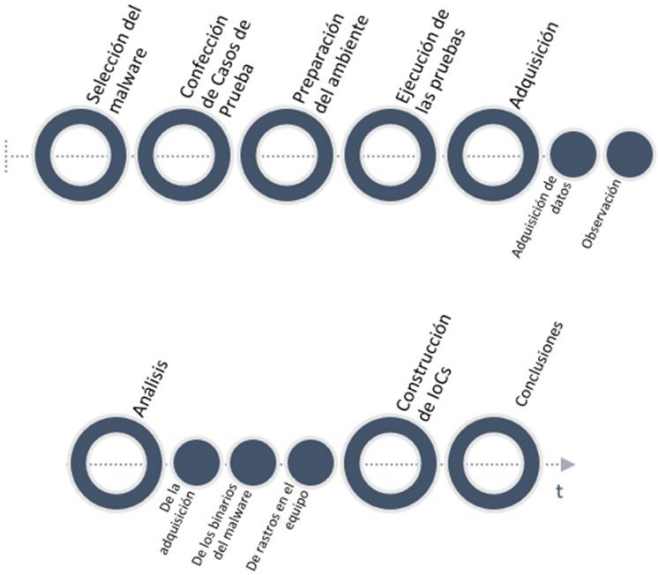

Figura 1. Proceso para la construcción de IoCs.

IV. Conclusiones

La presencia de *malware* crece día a día, de la mano con el aumento del uso de las tecnologías informáticas. Esta proporción, a futuro, no va a descender sino a aumentar, por más esfuerzos que se hagan. Es un efecto natural que, al

aumentar la cantidad de sistemas en uso, se incremente la cantidad de riesgos, dado que toda tecnología, por definición, siempre tiene vulnerabilidades, a pesar de ser desconocidas inicialmente.

En ese contexto, es imprescindible estar preparados para tal realidad, abordando la problemática desde distintos puntos de vista como lo son la seguridad informática o la informática forense, estableciendo la mejor metodología de estudio posible.

El análisis forense de evidencia digital, provenga éste de computadoras de escritorio, servidores o dispositivos móviles, enfrenta día a día nuevos desafíos. La proliferación de *malware* en sus distintas formas, ya sea *ransomware*, *spyware*, *stalkerware*, u otro tipo de software malicioso, plantea un escenario distinto que debe enfrentarse. Si durante una investigación judicial o un peritaje se sospechara la presencia de este tipo de software, los peritos informáticos cuentan con pocos recursos enfocados en su problemática.

Ante este desconocimiento inicial, la opción correcta siempre será extremar las medidas de prevención de daños colaterales. De ese modo, para quitar el error humano del proceso, resulta de vital importancia contar con una metodología estricta y correctamente documentada que describa todo lo requerido para garantizar el éxito del trabajo, permitiendo descubrir el comportamiento del *malware* sin ocasionar daños inesperados.

Una premisa correcta es considerar que el *malware* impacte en más recursos de los esperados. En respuesta a esto se debe considerar establecer tantos mecanismos de registro de actividad como se pueda. Por ejemplo, es posible que se comience un estudio con la convicción de que un *malware* solamente tiene actividad por red, pero si se desestima su potencial efecto sobre el sistema de archivos se perderá el registro de un posible comportamiento. Desde ese punto de vista es recomendable incorporar mecanismos de detección de actividades sobre todo recurso al cual pueda acceder el *malware*. De todas maneras, cada entorno de pruebas puede variar significativamente según el tipo de *malware* y el objeto de estudio a analizar. De acuerdo al interés del caso se determinará el entorno de pruebas conveniente.

Por otra parte, existe la disyuntiva de "ambiente real" versus "ambiente virtual", a través del uso de máquinas virtuales. En este punto no hay que caer en el enfoque sesgado de creer que uno siempre es mejor. Ambos tienen sus ventajas y desventajas. Por ejemplo, si bien el contexto virtual otorga más nivel de acceso a aspectos internos del sistema (como ser captura completa de la memoria principal, o imagen exacta de todos los procesos en ejecución en un instante determinado), también expone la prueba a que el comportamiento del *malware* no sea exactamente el mismo que en un entorno real. De hecho, es sabido que varios de ellos detectan el ambiente y en función de ello se ejecutan o no, para dificultar su estudio.

Una vez en el proceso se debe tener en cuenta que todo caso de análisis es distinto, y que cada uno de ellos se desarrollará con las tareas específicas que en el momento se requieran. De ese modo, a los fines de permitir replicar exactamente el mismo experimento en otros escenarios, se debe documentar cada simple actividad que se realice.

Contar con una metodología que permita replicar la experiencia fácilmente, en distintos contextos, profundizando el nivel de conocimiento del malware en cuestión y retroalimentando un proceso de mejora continua, permitirá a los informáticos forenses contar con la posibilidad de defender su trabajo en el ámbito judicial, garantizando la confiabilidad, reproducibilidad y validez de las conclusiones emanadas en el dictamen.

Referencias

[1] A. Di Iorio et al. "El rastro digital del delito. Aspectos técnicos, legales y estratégicos de la Informática forense". Ed. Universidad FASTA. Mar del Plata. 2017. Versión electrónica: http://redi.ufasta.edu.ar:8080/xmlui/handle/123456789/1593.

[2] M. Sikorski, A. Honig. "Practical malware analysis: the hands-on guide to dissecting malicious software". no starch press. 2012.

[3] M. H. Ligh, A. Case, J. Levy, A. Walters, The Art of Memory Forensics. Wiley, Inianapolis, Indiana, EEUU. 2014.

[4] A. Shaaban, K. Sapronov. "Practical Windows Forensics". 2016.

[5] Volatility Framework, https://github.com/volatilityfoundation/volatility.

[6] Volatility Foundations, http://www.volatilityfoundation.org/.

[7] Panda Security, Informe #WannaCry. 2017, disponible en: http://www.pandasecurity.com/spain/mediacenter/src/uploads/2017/05/Informe_WannaCry-es.pdf.

[8] M. E. Russinovich, D. A. Solomon, A. Ionescu. Windows internals 6th Edition. Pearson Education. 2012.

[9] M. E. Russinovich, D. A. Solomon, A. Ionescu. Windows internals 5th Edition. Pearson Education. 2009.

[10] CheckPoint Research. "EternalBlue - Everything There Is To Know". 2017. Recuperado el 30 de marzo de 2020 de https://research.checkpoint.com/2017/eternalblue-everything-know/.

[11] CheckPoint ©. "LOOKING INTO TESLACRYPT V3.0.1". 2016. Recuperado el 26 de marzo de 2020 de https://blog.checkpoint.com/wp-content/uploads/2016/05/Tesla-crypt-whitepaper_V3.pdf.

[12] CheckPoint ©. *Locky Ransomware*. 2016. Recuperado el 30 de marzo de https://blog.checkpoint.com/2016/03/02/locky-ransomware/.