Técnicas Anti-Forenses en Informática: Ingeniería Reversa Aplicada a TimeStomp

Armando Botero, Iván Camero y Jeimy Cano

Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana,
Carrera 7 No. 40 – 62, Bogotá, Colombia
{armando.botero, icamero, j.cano}@javeriana.edu.co

Resumen. Cada vez más las técnicas de evasión y las vulnerabilidades materializadas por los atacantes son más creativas y sofisticadas. En este contexto y conscientes del reto propio que esto implica para las investigaciones forenses en informática, se presenta en este documento un análisis de una de las herramientas anti-forenses conocidas como lo es timestomp, la cual es analizada en sus detalles e impactos sobre el sistema de archivo NTFS. El documento concluye con evaluación del funcionamiento detallado de timestomp para lo cual se utilizan técnicas de ingeniería reversa ilustrando los puntos clave para su detección y rastreo en NTFS.

Palabras Clave: Métodos Anti-Forenses, Timestomp, Computación Forense, NTFS, Ingeniería Reversa.

1. INTRODUCCIÓN

En la actualidad, para los investigadores de cómputo forense, se presentan retos cada vez más exigentes en cuanto al rastreo y detección de un atacante o "inquieto"[2], dada la alta creatividad de los intrusos en sus técnicas de evasión, las cuales cada vez son más sofisticadas y efectivas en los sistemas informáticos.

El reconocimiento de las vulnerabilidades en las herramientas utilizadas para adelantar procedimientos de informática forense, ha generado la aparición de las llamadas técnicas anti-forense que se definen como: "*cualquier intento de comprometer la disponibilidad de la evidencia para un proceso forense"* [1]. Estas técnicas buscan manipular el material más sensible de una investigación al destruir, ocultar, eliminar y falsificar la evidencia.

Para efectos de este artículo se revisarán las características de la herramienta timestomp, que se especializa en destruir y falsificar evidencia digital aprovechándose de las vulnerabilidades del sistema de archivos NTFS, atacando los atributos de tiempo MACE (Modificado, Accedido, Creado y Entry Modify) de cada archivo [10].

De otra parte, considerando que un método para determinar aspectos de fondo y observar el comportamiento en una aplicación de software o uno de sus componentes es el de la ingeniería reversa (IR) [19], se buscará identificar deficiencias en Timestomp para determinar los posibles rastros de la aplicación de esta herramienta, en sistemas Windows XP.

Para ilustrar el uso y funcionamiento de Timestomp, se presenta un ejemplo práctico, donde se detalla cómo opera dicha herramienta con los Atributos MACE y en general con la MFT (Master File Table). Luego, se realiza el análisis del código fuente de dicha herramienta basada en la ingeniería reversa aplicada sobre la misma.

2. DEFINICIÓN DE TÉCNICAS ANTI-FORENSES

Las herramientas o técnicas anti–forenses se definen según (Harris, 2006) como "*cualquier intento de comprometer la disponibilidad de la evidencia para un proceso forense."*[4] Del mismo modo si se profundiza un poco más en éste concepto y se desarrolla en términos más técnicos se genera la siguiente definición: "Cualquier intento exitoso efectuado por un individuo o proceso que impacte de manera negativa la identificación, la disponibilidad, la confiabilidad y la relevancia de la evidencia digital en un proceso forense" [6]

Estas técnicas proporcionan a los atacantes una ventaja inusual sobre los investigadores en cómputo forense, ya que al hacerse efectivas sobre la evidencia digital, pueden comprometer fácilmente la confianza y claridad de la misma en un proceso.

Así mismo, sugiere a los investigadores observar con un mayor detalle las evidencias digitales encontradas en una escena del crimen, lo que exige replantear los protocolos para investigaciones pasadas y futuras.

3. CLASIFICACIÓN DE MÉTODOS ANTI-FORENSES

A medida que se explora y se investiga más sobre las técnicas anti-forenses se han generado varias clasificaciones y del mismo modo se han definido varios métodos. Para efectos de este trabajo se tomará la clasificación planteada por (Harris 2006) a saber [7]:

- ✓ Destrucción de la evidencia.
- ✓ Ocultar la evidencia.
- ✓ Eliminación de las fuentes de la evidencia.
- ✓ Falsificación de la evidencia.

La sofisticación y complejidad de cada uno de estos métodos demuestra que los personajes interesados en su creación y ejecución -llamados normalmente intrusos- realizan muchas más cosas y acciones que lo que indican los manuales de los proveedores de software o hardware. [2]

A continuación se establece una aproximación a cada método propuesto por Harris de las herramientas anti-forenses:

- ✓ *Destrucción de la evidencia:*

  El principal objetivo de esta técnica es evitar que la evidencia sea encontrada por los investigadores y en caso de que estos la encuentren, disminuir sustancialmente el uso que se le puede dar a dicha evidencia en la investigación formal. Este método no busca que la evidencia sea inaccesible si no que sea irrecuperable. [7]

  Esto implica que se deben destruir, desmantelar o en su defecto modificar todas las pruebas útiles para una investigación (Harris, 2006) [4]. Así como en la vida real cuando ocurre un crimen y el criminal quiere destruir todo rastro o evidencia se vale de una serie de herramientas que le facilitan este objetivo.

  Existen dos niveles de destrucción de la evidencia [7]:
  - ✓ Nivel Físico: A través de campos magnéticos.
  - ✓ Nivel Lógico: Busca reinicializar el medio, cambiar la composición de los datos, sobrescribir los datos o eliminar la referencia a los datos.

  Existe una variedad de herramientas para la destrucción de evidencia de las cuales se pueden valer los intrusos para realizar este método anti-forense. Un ejemplo de herramientas son: Wipe, Shred, PGP Secure Delete, Evidence Eliminator y Sswap. [7]

- ✓ *Ocultamiento de la Evidencia:*

  Este método tiene como principal objetivo hacer inaccesible la evidencia para el investigador. No busca manipular, destruir o modificar la evidencia sino hacerla lo menos visible para el investigador. [4]

  Esta técnica puede llegar a ser muy eficiente de ser bien ejecutada pero conlleva muchos riesgos para el atacante o intruso, puesto que, al no modificar la evidencia de ser encontrada puede ser válida en una investigación formal y por lo tanto servir para la incriminación e identificación del autor de dicho ataque.

Este método puede valerse de las limitaciones del software forense y del investigador atacando sus puntos ciegos o no usuales de búsqueda de alguna anomalía. [4]

Una de las herramientas utilizadas por los atacantes es la esteganografía la cual versa sobre técnicas que permiten la ocultación de mensajes u objetos, dentro de otros, llamados portadores, de modo que no se perciba su existencia. [8] En el mercado se pueden encontrar muchos instrumentos fáciles de usar, de bajo costo que pueden ayudar a realizar esta técnica anti-forense, como por ejemplo StegoArchive [9].

- ✓ *Eliminación de la fuentes de la evidencia:*

  Este método tiene como principal objetivo neutralizar la fuente de la evidencia, por lo que no es necesario destruir las pruebas puesto que no han llegado a ser creadas. Por ejemplo, en el mundo real cuando un criminal utiliza guantes de goma para utilizar un arma lo que está haciendo es neutralizando y evitando dejar huellas dactilares en el arma. Así mismo en el mundo digital esta neutralización de las fuentes de la evidencia aplica. [4]

  Una de las acciones que los atacantes pueden llevar a cabo para realizar este método anti-forense es la desactivación de los log de auditoría del sistema que esté atacando.

- ✓ *Falsificación de la evidencia:*

  Esta método busca engañar y crear falsas pruebas para los investigadores forenses logrando así cubrir a el verdadero autor, incriminando a terceros y por consiguiente desviar la investigación con lo cual sería imposible resolverla de manera correcta.

  El ejercicio de este método se vale en una edición selectiva de las pruebas creando evidencias incorrectas y falsas que corrompen y dañan la validez de dichas pruebas en una investigación forense formal, por lo cual no podrán ser tomadas en cuenta como evidencias. [4]

4. FUNDAMENTOS DEL NTFS

Revisando la historia del sistema de archivos NTFS, encontramos que es un sistema de archivos diseñado e implementado por Microsoft, el cual surge como una necesidad para solucionar las fallas de seguridad, desempeño y confiabilidad que el sistema de archivos

FAT poseía [13]. Dichos atributos se optimizan en NFTS al manejar la mayoría de los datos como archivos, de esta manera se hace más sencillo el control de la partición, ya que se tiene un bloque de información de control almacenado en archivos con metadata desde el momento en que la partición es creada, lo que le permite al sistema operativo identificar y localizar cualquier archivo de manera más eficiente [14].

La estructura que maneja un volumen de este sistema de archivos se ilustra en la siguiente imagen.

Figura 1 Estructura del NTFS [5]

- ✓ Cabe resaltar que no todos los datos son archivos dentro de la partición. Por ejemplo, el Partition Boot Sector (PBS) no los maneja, ya que dicho fragmento en la estructura, es el encargado de hacer las operaciones del sistema de archivos.

  Para realizar las operaciones del sistema, el PBS se divide en dos sectores; el BIOS Parameter Block (BPB) y el Volume Boot Code (VBC). El BPB es el encargado de describir el formato que tiene la partición y la estructura de datos de metadata y archivos que maneja la capa física del volumen. Así mismo, posee el Boot Code, el cual se encarga de comunicarle al sistema operativo cuales son los recursos con los que la máquina cuenta [16]. Sabiendo las características físicas del computador, el VBC carga el sistema operativo con el código que es único para cada uno.

- ✓ La Master File Table (MFT) actúa como una base de datos relacional en la cual las filas son archivos de historiales y las columnas con archivos de atributos. Todos los archivos de una partición NTFS deben tener por lo menos una ocurrencia dentro de la MFT [12]. Los primeros dieciséis registros de tabla son usados para describirse. A partir del diecisieteavo registro comienzan todos los registros de la partición. El tamaño de los registros se asocia con el tamaño del cluster del volumen, sin embargo, tienen un mínimo de 1024 bytes y un máximo de 4096 bytes, así, si un registro tiene 512 bytes, el tamaño que se le asigna es de 1024 [15]. El ejemplo más claro de este tipo de atributo es el nombre del archivo o su *Time Stamp*. La clasificación de los registros se da por la información que tienen y los tamaños dados, en este contexto se hacen llamar atributos, que pueden ser residentes y no residentes. Anteriormente, se cito un ejemplo de lo que pasa cuando el tamaño de un registro es menor que el mínimo establecido; ese tipo de registros y los que son menores o iguales al máximo establecido, se

consideran residentes ya que se pueden almacenar en una sola tabla. Por otro lado, si un atributo llega a ser mayor de valor máximo estipulado, los clusters restantes se almacenar en una tabla adicional; este tipo se hace llamar atributos no residentes.

Cada registro posee una lista de atributos que se caracteriza no sólo como residentes o no, sino también por su tipo. Los tipos que concretamente atañen la investigación del artículo son:

- Standard Information Attribute (SIA): contiene información sobre modo de acceso, timestamps (marcas de tiempo) y cuenta de acoplamiento
- File Name (FN): posee información sobre nombre del archivo, un atributo repetible para los nombres corto y largo de un archivo.

Una lista detallada de todos los tipos se puede encontrar en [12].

- ✓ Los System Files o archivos de sistema son los archivos formales en los que se almacena información en forma de metadatos [16]. Se encuentran dentro de la MFT y guardan datos que la MFT no.
- ✓ El File Area contiene una copia de la MFT para efectos de recuperación de los datos en caso de problemas con la copia original [17].

5. TIMESTOMP

Timestomp es una herramienta anti-forense que sirve para leer y escribir los registros de tiempo o *time-stamps* de los archivos en NTFS (*New Tecnology File System*), desarrollada por *Metasploit Anti-Forensics Project* y distribuida de forma gratuita. Junto a herramientas como *slacker*, para ocultar archivos y *samjuicer* para obtener contraseñas de Windows NT/2000/XP/2003 conforman el *Metasploit Anti-Forensic Investigation Arsenal* (MAFIA) [10].

Los registros de tiempo o *time-stamps* de NTFS sirven para almacenar información de cuándo fue modificado (M), accedido(A), creado(C) y *entry modified* (E) un archivo en el sistema NTFS, lo cual se conoce como MACE. La primera sigla, M, se refiera a la fecha y hora del último cambio que se hizo sobre el atributo Datos de la *Master File Table* (MFT) de NTFS, lo que normalmente se conoce como "el archivo" [11], la segunda, A, se refiera a la última vez que el archivo se vio envuelto en una actividad, la tercera C, hace referencia a la fecha y hora en que fue creado, y el último, *entry modified* se refiere al tiempo en que fue modificado por última vez cualquier atributo del archivo dentro de la MFT, como su nombre, metadatos, datos, etc.

Según la clasificación que aparece en [10] y por las características ofrecidas por timestomp, esta es una herramienta anti forense que destruye y falsifica información.

Con *timestomp* se consigue evitar que un analista forense obtenga una línea de tiempo de sucesos en un sistema, dificultando la correlación de eventos y desacreditando la evidencia digital. Todo lo anterior perturba la etapa de análisis de datos dentro del proceso digital forense.

6. INGENIERÍA REVERSA EN APLICACIONES DE SOFTWARE

En ciencias de la computación, la ingeniería reversa (IR) se define como un proceso de análisis de un software o hardware para poder identificar sus componentes y cómo se relacionan entre sí y así crear una abstracción diferente o mayor, a la que el sistema presentaba originalmente [18]. Ya que se pretende aplicar alguna metodología de IR a Timestomp, en esta sección presentará de manera breve lo relacionado con la IR en las aplicaciones.

Definir una metodología específica para llevar a cabo la IR en el software no es labor sencilla, ya que es un proceso relativo a las entradas y lo que se espera como salida. Dentro de las entradas encontramos [19]:

- ✓ Código fuente
- ✓ Documentación
- ✓ Material de Pruebas
- ✓ Diagramas de diseño y arquitectura.
- ✓ Manuales de usuario, instalación y mantenimiento.
- ✓ Entrevistas con los desarrolladores del sistema a analizar.

Como posibles salidas encontramos [20]:

- ✓ Nuevos diagramas.
- ✓ Re-Ingeniería en el código fuente.
- ✓ Nueva documentación
- ✓ Bases de datos modificadas.
- ✓ Informes analíticos sobre falencias o fallos encontrados.

Para delimitar el problema de tener una amplia gama de posibles mezclas de entradas y salidas, se usará como entrada el código fuente de Timestomp para obtener como salida nueva documentación que pueda mostrar el funcionamiento de esta herramienta desde una perspectiva de la informática forense.

Para nuestro caso de estudio, se hará una exploración detallada y una descomposición en subrutinas, que busca encontrar en qué funciones se generan rastros que son el resultado

del uso de esta herramienta por un atacante, rastros con los cuales se pretende determinar la ejecución de una técnica anti-forense.

En conclusión, la salida que se espera al culminar el proceso de IR, es un informe analítico que refleje cómo Timestomp dejó algún rastro en los atributos MACE modificados.

7. ANÁLISIS DE TIMESTOMP CON IR

Cómo se describió anteriormente TimeStomp se vale de la alteración de los atributos MACE, los cuales se ubican en la MTF (Master File Table) de un sistema de archivos NTFS. Esta herramienta se enfoca específicamente en los Standard Information Attribute (SIA) de la MTF, donde se encuentran los Timestamp (estampillas de tiempo) que hacen referencia al momento de tiempo actual de un acontecimiento registrado por un computador [21].

Sin embargo, estos no son los únicos registros de tiempos existentes, puesto que hay otros almacenados en el File Name Attribute (FN) [11], [12] y que no son modificados por la herramienta *TimeStomp* lo que sugiere una posible debilidad de esta aplicación. La diferencia entre los registros de tiempo del SIA y los de FN radica en que en SIA la información MACE se modifica cada vez que llevamos a cabo alguna acción sobre un archivo como accederlo o modificarlo, en FN la información solamente se modifica cuando creamos o movemos al archivo de una ruta a otra [10]. Por lo tanto, las fechas y horas MACE registradas en el atributo FN - en la mayoría de los casos - va a ser más antigua a las registradas en el atributo SIA ya que las operaciones que modifican éste atributo se realizan con mayor frecuencia con respecto a los que cambian el FN.

Siguiendo con el análisis de la herramienta y dejando la anterior afirmación como inquietud a resolver más adelante, se tiene que la estructura general de los SIA es representada en la tabla 1, donde los atributos más importantes para el análisis que se está efectuando son: File Creation Time, File Alteration Time y File Read File; que son los que usa TimeStomp.

0x00	**8**		**File Creation Time**
0x08	8		**File Alteration Time**
0x10	8		MFT Change
0x18	8		**File Read Time**
0x20	4		DOS File Permissions
0x24	4		Maximum number of versions
0x28	4		Version number
0x2C	4		Class ID
0x30	4	2K	Owner ID

Tabla 1 Estructura de SIA 22

Para poder manipular y modificar los atributos arriba mencionados, TimeStomp utiliza principalmente las funciones NtQueryInformationFile(), NtSetInformationFile() de la librería ntdll.dll que proporciona Windows, las cuales son una colección de comandos que pueden ser compartidos por diferentes programas en una misma plataforma [23].

Dichas funciones se utilizan para modificar (NtSetInformationFile()) y para consultar NtQueryInformationFile() los File Standard Information donde se encuentran los atributos MACE.

La utilización de estas funciones se evidencia en el código fuente de TimeStomp, específicamente en sus principales rutinas **SetFileMACE y RetrieveFileBasicInformation** [26]**.** Las cuales cargan la librería ntdll y hacen el respectivo llamado a las funciones que realizan el trabajo de modificar los atributos MACE mencionadas anteriormente.

Una de las principales ventajas de timestomp radica en el uso de funciones de la librería ntdll.dll, ya que esta puede ser invocada por cualquier programa en modo usuario, es decir sin necesidad de privilegios [24], como lo muestra la figura 2. Además la utilización y llamado a estas funciones no queda registrada por el sistema, lo que implica que no se genera evidencia del uso de dichas funciones.

Asimismo al valerse de la ntdll.dll, no hace necesario que timestomp inyecte nuevas librerías dll o APIs, porque las que utiliza son proporcionadas por el mismo sistema comprometido.

Estas ventajas conllevan a la disminución de la visibilidad que generaría la utilización de timestomp, a los ojos de una investigación de cómputo forense. Además se puede observar que el desarrollador de esta herramienta, posee un nivel medio de sofisticación y conocimientos técnicos puesto que el uso de la librería ntdll está prácticamente sin documentación [24].

```
ntdll = LoadLibrary("ntdll.dll");
if (ntdll == NULL) {
    return 0;
}
```

Figura 2. Invocación Librería ntdll.dll [26]

A continuación se realizará un estudio detallado de las funciones del código fuente de la herramienta.

8.1. Estructura general de la clase timestomp.c

Las estructuras declaradas dentro del código y las funciones dentro de él son las siguientes:

```
DWORD ParseDateTimeInput(char *inputstring, SYSTEMTIME *systemtime);
HANDLE RetrieveFileBasicInformation(char *filename, FILE_BASIC_INFORMATION *fbi);
DWORD ConvertLocalTimeToLargeInteger(SYSTEMTIME localsystemtime, LARGE_INTEGER *largeinteger);
DWORD ConvertLargeIntegerToLocalTime(SYSTEMTIME *localsystemtime, LARGE_INTEGER largeinteger);
DWORD SetFileMACE(HANDLE file, FILE_BASIC_INFORMATION fbi);
DWORD TheCraigOption(char *directoryname);
```

Figura 3. Funciones de TimeStomp [26]

```
typedef struct _IO_STATUS_BLOCK {
    union {
        NTSTATUS Status;
        PVOID Pointer;
    };
    ULONG_PTR Information;
} IO_STATUS_BLOCK, *PIO_STATUS_BLOCK;

typedef enum _FILE_INFORMATION_CLASS {
    FileBasicInformation = 4,
    FileStandardInformation = 5,
    FilePositionInformation = 14,
    FileEndOfFileInformation = 20,
} FILE_INFORMATION_CLASS, *PFILE_INFORMATION_CLASS;

typedef struct _FILE_BASIC_INFORMATION {
    LARGE_INTEGER CreationTime;                                    // Created
    LARGE_INTEGER LastAccessTime;                    // Accessed
    LARGE_INTEGER LastWriteTime;                     // Modifed
    LARGE_INTEGER ChangeTime;                        // Entry Modified
    ULONG FileAttributes;
} FILE_BASIC_INFORMATION, *PFILE_BASIC_INFORMATION;
```

Figura 4. Estructuras de TimeStomp [26]

En las siguientes sub-secciones se mostrará como las funciones *SetFileMACE*, *RetrieveFileBasicInformation* y *TheCraigOption* interactúan con el sistema. Las funciones restantes al no acceder relevantemente a la MFT se describirán brevemente a continuación.

- ParceDateTimeInput: esta función convierte una cadena de caracteres al tiempo manejado por el sistema opertativo (SYSTEMTIME) en el formato adecuado, retorna 1 si la conversión fue exitosa y 0 en el caso contrario.
- ConvertLocalTimeToLargeInteger: ya que los timestamps de un archivo son almacenados como tiempo universal coordinado o UTC, para hacer operaciones sobre estos datos es necesario convertirlos a un formato que lo permita. Esta función se encarga de transformar los UTC a un LARGE_INTEGER para realizar operaciones sobre ellos.
- ConvertLargeIntegerToLocalTime: realiza la operación inversa a la anterior.

8.2. SetFileMACE

- ✓ Aspectos importantes vistos desde la informática forense: SetFileMACE() es la función clave de TimeStomp ya que es la encargada de realizar la modificación de los atributos MACE principal objetivo de esta herramienta. Al analizar su funcionamiento, se observa que las funciones LoadLibrary(), GetProcAddress() y FreeLibrary() proporcionadas por el API de Windows [27] y NtSetInformationFile() que proporciona ntdll.dll no dejan ningún registro o evidencia de la utilización o llamados por parte de SetFileMACE() a estas;

además Timestomp al utilizar librerías propias del sistema operativo y el sistema de archivos, evita inyectar dll's diferentes que podrían dejar algún rastro y evidenciar de esta herramienta.

8.3. RetrieveFileBasicInformation

- ✓ Aspectos relevantes vistos desde la informática forense: El principal objetivo de esta función es almacenar los datos retornados NtQueryInformationFile() en la estructura FILE_BASIC_INFORMATION, los cuales son tomados directamente de los SIA de la MFT. Sin embargo, esto no se puede tomar como evidencia ya que esta función es directamente tomada de la librería ntdll.dll y no deja rastro alguno de su uso.

8.4. TheCraigOption

- ✓ Aspectos relevantes vistos desde la informática forense: Esta función trabaja principalmente con la estructura WIN32_FIND_DATA [28], que contiene toda la información importante de un archivo, de la que hace parte los atributos MACE. Asimismo valiéndose de esta estructura utiliza sus funciones, tales como FindFirstFile(), FindNextFile() y FindClose() para lograr el objetivo de eliminar los valores MACE en un análisis hecho con la herramienta de investigación forense EnCase. La utilización de estas funciones no deja rastro alguno, así que es imposible para EnCase conocer y darse cuenta de la utilización de estas, además está atacando los atributos de SIA, que son precisamente los que valida y utiliza para sus informes y generación de líneas de tiempo la herramienta EnCase.

8. VULNERABILIDADES DE TIMESTOMP

Timestomp presenta una debilidad que puede ser aprovechada por los analistas forenses para evidenciar se ha usado este programa y conforme a esto responder a las circunstancias apropiadamente.

En la figura 6 se muestra una imagen tomada con la herramienta EnCase, en la cual se ilustra una MFT al crear un archivo cualquiera. En ella se ven los valores almacenados en el SIA y el FN del archivo, junto con sus respectivos atributos MACE. Posterior a la creación del archivo, se accedió al archivo y se realizaron una serie de modificaciones con el fin de ilustrar que dichos cambios se almacenan únicamente en el SIA, más no en el FN.

Es por esto, que el principal defecto de timestomp radica en la modificación de la información MACE del SIA y no del FN dentro de la MFT, por tanto en condiciones normales, las fechas y horas registradas en SIA deben ser mayores o iguales a las

registradas en FN, entonces es posible asegurar que han cambiado las fechas del SIA y sospechar que han usado timestomp, al no ver el cambio en el FN.

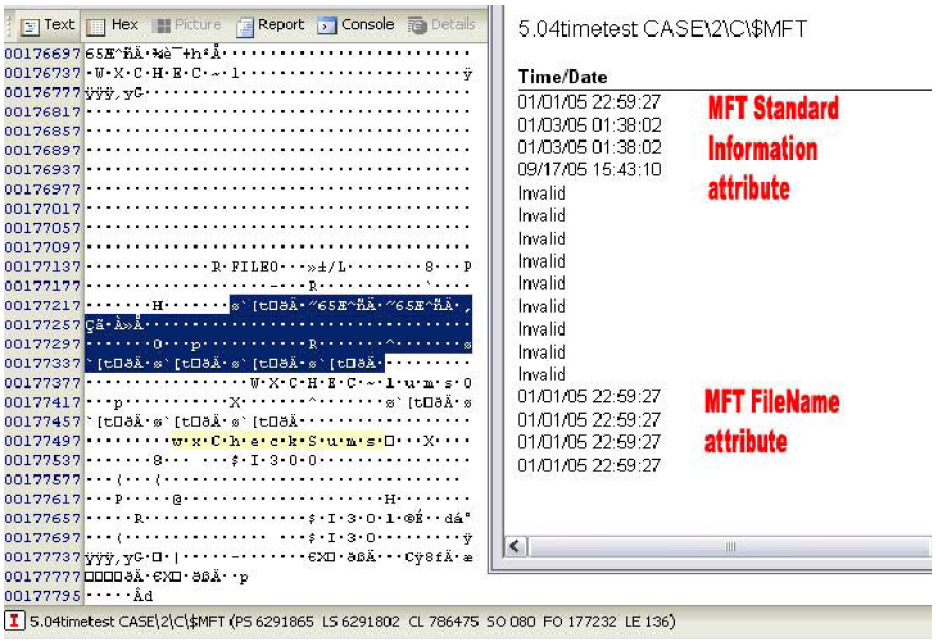

Figura 6. Estructuras de TimeStomp [29]

Una potencial solución a esta debilidad de timestomp consiste en modificar los valores FN con una herramienta que permita alterar datos "crudos" como WinHex [10], de tal forma se evita la inconsistencia en los datos del tiempo.

9. CONCLUSIONES

- ✓ A medida que las técnicas y tecnología que utilizan los atacantes informáticos evolucionan, la administración de justicia requiere mayores elementos de análisis e investigación, para identificar y determinar quien cometió el ataque.
- ✓ El trabajo de la computación forense se hace cada vez más complejo ya que con la utilización de las técnicas anti-forenses se está atacando y vulnerando tanto la evidencia como las herramientas utilizadas.
- ✓ A la fecha no existe un modelo certificado y unificado con el que los investigadores forenses cuenten para la identificación y análisis que permita determinar que se están utilizando herramientas anti-forenses.

- ✓ Timestomp utiliza funciones de la librería ntdll.dll las cuales no dejan ningún registro o rastro de su utilización, lo que dificulta el comprobar si un archivo fue intervenido con esta herramienta o no.
- ✓ Las herramientas de cómputo forense podrían evaluar y analizar los atributos FN y compararlos con los SIA, para realizar las líneas de tiempo y así minimizar las inconsistencias que se puedan presentar.

10. REFERENCIAS

1. Arckoff R., Addisson H. (2007), Management F-Law. How Organizations Really Work.
2. Jeimy. J. Cano. (2007). Inseguridad informática y computación anti-forense: Dos conceptos emergentes en seguridad de la información. http://www.virusprot.com/Archivos/Antifore07.pdf
3. Jeimy. J. Cano. (2007). Introducción a las técnicas anti forenses: Conceptos e implicaciones para investigadores. http://www.acis.org.co/fileadmin/Base_de_Conocimiento/VII_JornadaSeguridad/VIIJNSI_JCano.pdf.
4. R.Harris. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. http:// dfrws.org/2006/proceedings/6-harris.pdf
5. López, Oscar; Amaya, Haver; León Ricardo; Acosta Beatriz. (2002). Informática Forense: Generalidades, aspectos técnicos y herramientas. http://www.criptored.upm.es/guiateoria/gt_m180b.htm
6. Jeimy. J. Cano. (2007). Introducción a la informática forense. http://www.acis.org.co/fileadmin/Revista_96/dos.pdf
7. Andres R. Almanza. (2007). Ciencias Anti-forense. Un nuevo reto para las organizaciones. http://www.acis.org.co/fileadmin/Base_de_Conocimiento/VII_JornadaSeguridad/VIIJNSI_AAlmanza.pdf
8. Info Seguridad: Esteganografia. http://www.infoseguridad0.es/Estenografia.htm
9. Stegoarchive. What is Steganography?. http://www.stegoarchive.com/
10. Metasploit Anti-Forensics Project. http://www.metasploit.com/data/antiforensics/BlueHat-Metasploit_AntiForensics.ppt
11. MicrosoftTech. NTFS TimeStamps. http://blogs.technet.com/ganand/archive/2008/02/19/ntfs-time-stamps-file-created-in-1601-modified-in-1801-and-accessed-in-2008.aspx
12. MicrosoftTech. How NTFS Works. http://technet.microsoft.com/en-us/library/cc781134.aspx
13. PCGuide. Overview and History of the NTFS. http://www.pcguide.com/ref/hdd/file/ntfs/ver.htm

14. Ariza. A., Ruiz. J., Análisis de Metadatos en archivos Office y Adobe. (2008). http://www.criptored.upm.es/guiateoria/gt_m142g1.htm
15. PCGuide. NTFS file Atrtibutes. http://www.pcguide.com/ref/hdd/file/ntfs/files_Attr.htm
16. PCGuide. NFTS System (Metadata) Files. http://www.pcguide.com/ref/hdd/file/ntfs/arch_Files.htm
17. Microsoft Corporation. How NTFS works. http://technet.microsoft.com/en-us/library/cc781134.aspx
18. Chikofsky. E., Cross. J., Reverse Engineering and Design Recovery: A Taxonomy, IEEE Software, pp. 13-17, 1990.
19. Galen Lab. Reverse Engineering. http://calla.ics.uci.edu/reveng/
20. Electronic Design. Reverse Engineering. http://electronicdesign.com/Articles/Index.cfm?AD=1&ArticleID=11966
21. Müller H., Jahnke J., Smith D., Storey M., Tilley S., Wong K.. Reverse Engineering. A Roadmap. (2005). http://www.cs.ucl.ac.uk/staff/A.Finkelstein/fose/finalmuller.pdf
21. What is: The Leading TI encyclopedia and learning Center. What is TimeStamp?. http://whatis.techtarget.com/definition/0,,sid9_gci817089,00.html
22. Santa Clara University, School of Engineering. NTFS File System. http://www.cse.scu.edu/~tschwarz/coen152_05/PPtPre/NTFSFS.ppt
23. Help with PCs. DLL. http://www.helpwithpcs.com/jargon/dll.htm
24. Win32 API Obscurity for I/O Blocking and Intrusion Prevention (2005). http://www.ddj.com/security/184406098
25. MetaSploit Anti-Forsensics Project. http://www.metasploit.com/data/antiforensics/BH2005-Catch_Me_If_You_Can.ppt
26. Metasploit Anti-Forsensics Project. Timestomp codigo fuente. http://trac.metasploit.com/browser/framework3/trunk/external/source/meterpreter/source/extensions/priv/server/timestomp.c?rev=6357
27. Microsoft Developer Network. Windows API. http://msdn.microsoft.com/en-us/library/aa383750.aspx
28. Microsoft Developer Network. WIN32_FIND_DATA. http://msdn.microsoft.com/en-us/library/aa365740(VS.85).aspx
29. Black. G., Evidence of Folder Renaming using MFT Standard Information Attribute and FileName Attribute. (2005). http://www.geoffblack.com/forensics/Evidence_of_Folder_Renaming.pdf