# INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO EN CÓMPUTO

## ESTEGANOGRAFÍA EN PROTOCOLOS DE RED

**T E S I S**

QUE PARA OBTENER EL GRADO DE:

**MAESTRÍA EN TECNOLOGÍA DE CÓMPUTO**

PRESENTA:

**GABRIELA HERNÁNDEZ LÓPEZ**

DIRECTORES DE TESIS:

**M. EN C. EDUARDO RODRÍGUEZ ESCOBAR**

**M. EN C. JESÚS ANTONIO ÁLVAREZ CEDILLO**

México, D. F. Noviembre 2013

SIP-14-BIS

# INSTITUTO POLITÉCNICO NACIONAL
## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

*ACTA DE REVISIÓN DE TESIS*

En la Ciudad de México, D.F. siendo las 13:00 horas del día 25 del mes de noviembre del 2013 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del CIDETEC para examinar la tesis titulada:

"ESTEGANOGRAFÍA EN PROTOCOLOS DE RED"

Presentada por el alumno:

| HERNÁNDEZ | LÓPEZ | GABRIELA |
|---|---|---|
| Apellido paterno | Apellido materno | Nombre(s) |

Con registro:

| B | 1 | 1 | 0 | 9 | 3 | 3 |
|---|---|---|---|---|---|---|

aspirante de:

Maestría en Tecnología de Cómputo

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de tesis

M. EN C. EDUARDO RODRÍGUEZ ESCOBAR
Primer Vocal

M. EN C. JESÚS ANTONIO ÁLVAREZ CEDILLO
Segundo Vocal

DR. ROLANDO FLORES CARAPIA
Presidente

DR. MAURICIO OLGUÍN CARBAJAL
Secretario

M. EN C. JUAN CARLOS GONZÁLEZ ROBLES
Tercer Vocal

DR. VÍCTOR MANUEL SILVA GARCÍA
Suplente

PRESIDENTE DEL COLEGIO DE PROFESORES

DR. OSCAR CAMACHO NIETO

S.E.P.
INSTITUTO POLITECNICO NACIONAL
CENTRO DE INNOVACIÓN Y DESARROLLO TECNOLOGICO EN COMPUTO

# *INSTITUTO POLITÉCNICO NACIONAL*

## *SECRETARÍA DE INVESTIGACIÓN Y POSGRADO*

## *CARTA CESIÓN DE DERECHOS*

En la Ciudad de México el día 29 del mes noviembre del año 2013, el (la) que suscribe Gabriela Hernández López alumno (a) del Programa de Maestría en Tecnología de Cómputo con número de registro B110933, adscrito a Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC), manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de M. en C. Eduardo Rodríguez Escobar y el M. en C. Jesús Antonio Cedillo Álvarez y cede los derechos del trabajo intitulado Esteganografía en Protocolos de Red, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección ghl_89@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Gabriela Hernández López

# Resumen

La esteganografía es una disciplina que trata las técnicas para ocultar información dentro de un medio que actuará como portador, el cual puede ser un archivo multimedia como audio, video, imágenes, o también se puede utilizar un protocolo de red.

La esteganografía tiene como propósito enviar mensajes ocultos a través de una comunicación entre dos entidades de manera que el mensaje oculto pase inadvertido en dicha comunicación.

La esteganografía de protocolos utiliza ciertas características de algún protocolo de red para ocultar información, haciendo uso de canales encubiertos que son canales de comunicación que puede ser explotado por un proceso de transferencia de información de manera que viola la política de seguridad de un sistema.

Esta investigación realiza un estudio sobre la esteganografía como medio para proteger la información, también se hace un análisis sobre la esteganografía de protocolos así como sobre las diferentes técnicas para explotarlos.

También se tratan los Flujos Alternativos de Datos (ADS) que son una característica propia de los sistemas de archivos NTFS, dicha característica permite ocultar información; también se exponen los beneficios que brinda utilizar los ADS para ocultar información.

Por último se realiza la implementación de esteganografía de red junto con los ADS, de manera que permitan ocultar información, analizando sus respectivas ventajas y desventajas.

# Abstract

Steganography is a discipline that study techniques to hide information within an environment that will act as a carrier, which can be a media file such as audio, video, images, or you can use a network protocol.

Steganography aims to send hidden messages through a communication between two entities so that the hidden message in this communication go unnoticed .

The protocol steganography uses certain characteristics of a network protocol to hide information , making use of covert channels are communication channels that can be exploited by a process to transfer information in a manner that violates the security policy of a system.

This research makes a study on steganography as a means to protect the information, also provides an analysis on the protocol steganography as well as the different techniques to exploit them.

It also discusses the Alternative Data Streams (ADS), which are a feature of NTFS file systems, this feature allows you to hide information; also outlines the benefits provided by using ADS to hide information.

Finally we implement a steganographic algorithm along with the ADS, so as to allow hiding information, analyzing their respective advantages and disadvantages.

# Agradecimientos

A Dios por acompañarme y brindarme fortaleza en los momentos de dificultad; por darme salud para poder terminar satisfactoriamente mis estudios.

A mi padre, por tu apoyo incondicional, por creer en mi, por tus palabras de apoyo en los momentos difíciles, por forjarme de una manera en la cual he podido alcanzar mis metas; se que desde el cielo me cuidas y siempre estarás guiándome.

A mi madre, por tus consejos, por tu apoyo y por tu amor incondicional, porque eres una luchadora incansable, porque de ti he aprendido a no dejarme caer, por no dejar de creer en mi.

A mis hermanos, Alma y Alfredo, por las experiencias vividas, por su ayuda, por sus palabras que nunca dejan que me rinda.

Al IPN, porque me siento sumamente orgullosa de pertenecer a esta institución, porque soy politécnica por convicción no por circunstancia.

A mis abuelitos, tíos y tías, por todo el cariño brindado a lo largo de esta etapa, por las palabras de aliento, pero sobre todo por estar junto a mi en los momentos de dificultades.

A mis maestros del CIDETEC, por sus enseñanzas, apoyo y comprensión que me han brindado a lo largo de esta etapa.

A mis directores de tesis, por su paciencia y el apoyo brindado para la realización de esta tesis, por darme la oportunidad de crecer profesionalmente y aprender cosas nuevas.

A Pablo, eres un gran amigo, agradezco infinitamente tu apoyo pero sobre todo la paciencia que me tuviste en los momentos de duda y adversidad.

A Mario, por estar a mi lado en los momentos de debilidad, por tu cariño y por tu apoyo incondicional durante las adversidades.

A cada uno de ustedes gracias.

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

La tecnología se encuentra en constante cambio, y con ello también las técnicas de seguridad que se usan para preservar la confidencialidad e integridad de los datos compartidos por dos o más usuarios.

Al utilizar el Internet como medio de comunicación, es bastante fácil interceptar información y datos que se encuentren circulando libremente a través de las redes quedando vulnerable, por lo que se han desarrollado métodos que permiten proteger la información.

Uno de estos métodos es la encriptación, que se basa en transformar y alterar la información original en otra que no sea comprensible para un intruso, de esta manera no se puede conocer el contenido de la información; pero también existen otros métodos que tienen por objetivo ocultar información de modo que un intruso no podrá notar su presencia.

La ocultación de información es la ciencia que abarca los diferentes métodos que permiten ocultar cualquier tipo de información, independiente de su naturaleza, medios o fines.

La esteganografía trata sobre el estudio de las técnicas que permiten ocultar información, basándose en la selección de un portador que contendrá un mensaje o un archivo multimedia oculto.

La esteganografía en protocolos de red también conocida como esteganografía de red, se dedica a ocultar la información haciendo uso de ciertas características de los protocolos de red.

La esteganografía de red hace uso de canales encubiertos para transferir información, un canal encubierto se puede definir como un canal que cumple otros propósitos para los que no fue hecho, en este caso, enviar información de manera secreta.

Por otra parte los ADS, son una característica particular del Sistema de Archivos de Nueva Tecnología (NTFS por sus siglas en inglés, New Technology File System), y son considerados un método para ocultar información.

En este trabajo se presenta un análisis y ejecución de un procedimiento que usa esteganografía de red junto con los ADS como técnicas para ocultar información.

## 1.1. Trabajo previo

El uso de técnicas esteganográficas para proteger la información ha ido aumentando debido a los grandes beneficios que ofrece; aunque cada vez existe una mayor documentación sobre dichas técnicas, hay cuestiones importantes que no han sido tratadas, por ejemplo, en [24] se exponen diversos puntos sobre la esteganografía y el estegoanálisis basándose en un modelo aplicable al mundo real, debido a que no es lo mismo aplicar esteganografía en un escenario ideal que en uno real.

La esteganografía sobre protocolos de red tiene diversas posibilidades para ser explotada siendo una buena opción para poder ocultar información de manera que pase desapercibida ante la presencia de terceras personas.

El uso de la esteganografía de red permite desarrollar aplicaciones esteganográficas sutiles que sean capaz de poder establecer canales de comunicación encubiertos sobre el conjunto de protocolos TCP/IP.

Existen diversos métodos para crear canales encubiertos sobre TCP/IP, sin embargo uno de los inconvenientes que presenta es que la cantidad de información que se envía es limitada, en [19] se propone un diseño de canal encubierto con un alto ancho de banda utilizando el protocolo TCP Tahoe, manteniendo un comportamiento de tráfico de red normal, su esquema propuesto puede ser utilizado por todos los protocolos de red.

Actualmente existe un gran interés en la explotación de canales encubiertos basados en TCP/IP, en [28] se hace un análisis y una mejora del protocolo TCP/IP, y se propone un programa para el uso de TCP/IP basado en un sistema embebido, enfocándose en la optimización de dichos protocolos, de acuerdo a un hardware y software para sistemas embebidos y aplicaciones específicas de Internet, permitiendo mejorar el nivel de software para el soporte de recursos de sistemas embebidos.

Por otra parte, Shah [39] presenta una forma de integrar marcas de agua en los encabezados de los paquetes de datos TCP/IP.

Existe gran interés en el uso de esteganografía con criptografía para reforzar la integridad de la información, en [21] se implementa el uso de cifrado de datos junto con esteganografía en los métodos conocidos para ocultar información en TCP/IP con el cifrado de bloques ECHAR -128 o (Cast 5).

En [13] se muestra el diseño de un sistema, que utiliza criptografía y esteganografía, donde los encabezados de los protocolos TCP e IP se utilizan como soporte para ocultar los datos cifrados, usando imágenes como portadores que se dividen en paquetes los cuales están ocultos en los campos no utilizados de la cabeceras de dichos protocolos, utilizando la curva elíptica para el proceso de cifrado.

En [36] se presenta un algoritmo esteganográfico que permite ocultar información dentro de los documentos de texto ASCII haciendo también uso de la criptografía.

En [16], se describe un canal encubierto mediante la utilización del campo llamado tiempo de vida (TTL) contenido en los paquetes del servicio de nombres de dominio (DNS), dicho campo especifica el tiempo máximo que otros servidores DNS y aplicaciones deben mantener en caché ese registro, debido a que no hay un valor establecido para este campo lo hace un portador ideal para transmitir información oculta. En [33] se propone un canal encubierto en la trama de Ethernet.

Otra propuesta de canales encubiertos se presenta en [5] donde se muestra un esquema esteganográfico basándose en una comunicación secreta robusta a través del correo electrónico.

En [20] se propone un método que permite crear un canal encubierto en la capa de transporte usando los puertos de origen, que generalmente son un número pseudo aleatorio seleccionado dentro de un rango dado, aprovechando esta flexibilidad sobre la selección del puerto de origen; se crea un canal encubierto unidireccional cuando un usuario o proceso manipula el puerto de origen para poder enviar datos.

En [6] se trata la creación de un canal encubierto basándose en el protocolo BitTorrent capaz de enviar mensajes de manera discreta y secreta entre dos partes. En [8] se propone un nuevo enfoque para la creación de canales encubiertos basado en la clasificación de paquetes, este método se basa en elegir un portador y un algoritmo de ocultación de información basado en una clasificación de paquetes que consta de tres jerarquías: la clase A, la cual se agrupa por protocolos, por ejemplo, protocolos de la capa de red (ICMP, IGMP), protocolos de la capa de transporte (TCP, UDP), protocolos de la capa de sesión (SSL) y protocolos de la capa de aplicación (DNS, HTPP, SMPT, FTP); la clase B se agrupa por cabeceras de paquetes, incluyendo los diferentes valores de algunos campos, y la clase C que se refiere a las cabeceras de los campos sin usar.

En [46] se crea un prototipo de canal encubierto llamado CCCA por sus siglas en inglés "Covert Channel Core Alteration" o canal encubierto de alteración de núcleo, en el cual mediante el uso de procesadores multi-core se crea un canal encubierto que permite la comunicación de datos secretos utilizando una transmisión unidireccional, capaz de usarse en diferentes sistemas operativos y plataformas de virtualización con procesadores multi-core.

En [17] se describe un nuevo algoritmo esteganográfico de alta capacidad que permite la incorporación de datos en las tramas inactivas de los flujos de audio de baja velocidad binaria.

La explotación de canales encubiertos no solo se enfoca a equipos de cómputo conectados en red, actualmente se esta explorando también en dispositivos móviles, su uso va en aumento siendo su principal objetivo diversos atacantes que desean acceder a su información, en [11] se presenta un trabajo en el que se implementan canales encubiertos de red que permiten transferir información oculta entre un dispositivo móvil basado en la plataforma Android y un servidor externo.

El gran interés que existe en crear canales encubiertos para la protección de la información ha llevado a realizar una investigación sobre las diferentes maneras de detectar canales encubiertos, en [38] se proporciona un método para detectar y controlar canales encubiertos mediante el empleo de entropía.

La entropía nos dice cuanta aleatoriedad existe en un conjunto de valores para una variable dada, en este caso, la variable aleatoria es un campo en una cabecera de protocolo de red. Basándose en la suposición de que la entropía de los canales encubiertos puede variar, dicho cambio proporciona un método para la identificación de canales encubiertos de almacenamiento. En [44] se presenta una primera aproximación para limitar el ancho de banda en canales encubiertos mediante el uso de un guardián activo.

### 1.1.1. Objetivo General

- Aplicar esteganografía de red con el uso de ADS para ocultar información.

### 1.1.2. Objetivos Particulares

- Investigar y conocer los diferentes métodos y técnicas esteganográficas actuales.
- Definir cual protocolo de red será utilizado en la investigación para su manipulación.
- Proponer un procedimiento para aplicar esteganografía de red con ADS para ocultar información.
- Implementar el procedimiento propuesto en una plataforma de software.
- Realización de pruebas.

### 1.1.3. Planteamiento del problema

Con la evolución constante de las tecnologías de información (TI) y su manera de estar presentes en la vida profesional y privada de las personas, aparecen asociados nuevos riesgos que son necesarios evitar o al menos minimizar. De esta manera surge el tema de la Seguridad de la información.

Es por ello que se desea dar una alternativa para que la información de los usuarios que fluye a través de Internet sea protegida mediante el uso de diversas técnicas cuya finalidad es ocultar información.

### 1.1.4. Propuesta de Solución

Existen diferentes métodos para cifrar la información y de esta manera ser difícil de entender para terceras personas, sin embargo existe el riesgo de que dicha información pueda ser descifrada.

Por tal motivo, es necesario hacer uso de otras técnicas que permitan ocultar la información y no levantar sospecha alguna, es por ello que se propone el uso de la esteganografía para proteger la información.

La mayoría de las personas que tienen acceso a Internet no tiene mucho conocimiento sobre las diversas técnicas que existen para proteger su información, además de que los diversos usuarios que acceden a Internet a través de una computadora usan el sistema operativo Windows 7.

Por esta razón se propone implementar un algoritmo esteganográfico utilizando características propias del protocolo seleccionado, reforzando dicho algoritmo con el uso de los ADS, enfocándose en equipos con sistema operativo Windows 7.

### 1.1.5. Organización de la tesis

Esta tesis se encuentra compuesta por 5 capítulos organizados de la siguiente manera:

- El segundo capítulo muestra un panorama general sobre la esteganografía de protocolos de red.
- El tercer capítulo abarca el estudio de los ADS.
- El cuarto capítulo detalla la aportación principal de este trabajo así como los resultados obtenidos en este trabajo haciendo énfasis en su funcionamiento.
- El quinto capítulo describe las conclusiones del trabajo así como los trabajos futuros que pueden desarrollarse tomando como base esta tesis.

# Capítulo 2

# Esteganografía en el Protocolo TCP/IP

La seguridad de la información tiene dos ramas principales, la criptografía y la ocultación de información, a la cual pertenecen la esteganografía y las marcas de agua, como se puede apreciar en la figura 2.1.

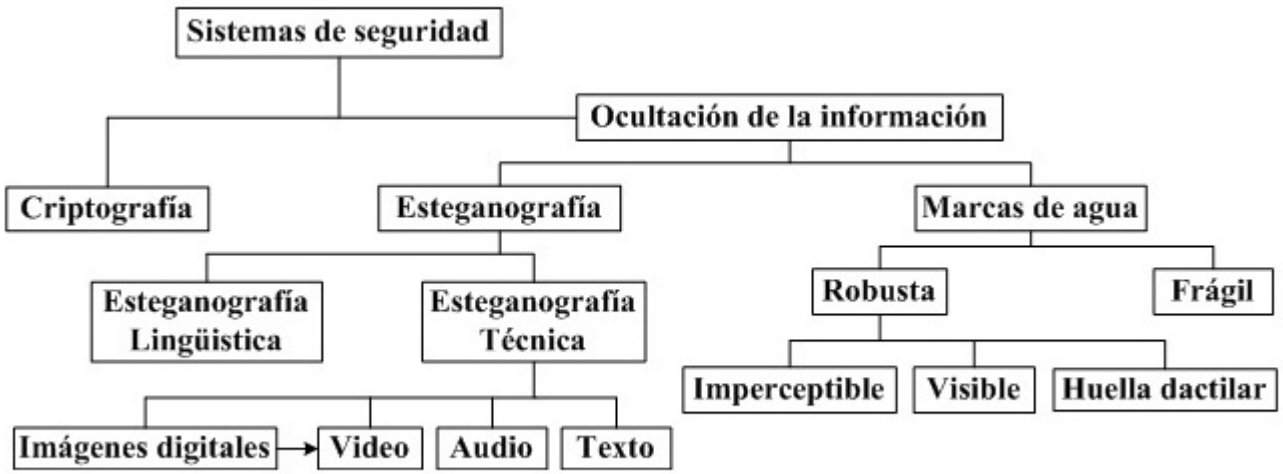


Figura 2.1: Diferentes sistemas de seguridad.

Para los que no trabajan en seguridad informática existe cierta confusión entre criptografía y esteganografía, por un lado, en la criptografía se aplica una transformación para cifrar datos de manera que sean ilegibles para una tercera persona y que solo será posible interpretarlos mediante la aplicación de una transformación inversa que consta de una llave secreta, mientras que la esteganografía pretende ocultar el hecho de que se esta enviando información de manera oculta dentro de un portador de apariencia normal.

También suele existir cierta confusión entre las marcas de agua y la esteganografía ya que ambas son métodos que ocultan información dentro de un medio, sin embargo, las marcas de agua son métodos para ocultar información sobre protección de derechos de autor buscando ser un método robusto, mientras que en la esteganografía se busca pasar de manera inadvertida utilizándose principalmente en comunicaciones secretas.

## 2.1. Definición de esteganografía

Esteganografía, término derivado de las palabras griegas "steganos" que significa encubierto, y "graphein" que significa escribir, por lo que literalmente puede traducirse como "escritura encubierta".

Es la disciplina que estudia el conjunto de técnicas que tienen por objetivo ocultar información sensible, mensajes u objetos, dentro de otros archivos denominados contenedores o portadores.

### 2.1.1. Propósito

El propósito de la esteganografía es establecer una vía de comunicación secreta entre dos partes, de modo que una tercera parte ubicada entre ambas no sea capaz de detectar la existencia de tal comunicación. Viene bajo la suposición de que si la función está visible, el punto de ataque es evidente, por lo tanto, la meta es siempre ocultar la existencia misma de los datos incorporados.

## 2.2. Historia

Los primeros documentos que describen el uso de técnicas esteganográficas fueron desarrolladas en la antigua Grecia en el siglo V a.C., el primer registro escrito de la transmisión de un mensaje secreto proviene del escrito por Heródoto "Las historias", donde cuenta el procedimiento que usó el general Histaieus para enviar un mensaje secreto; el cual consistía en rasurar la cabeza de un esclavo, (que actuaba como portador), para posteriormente tatuar el mensaje; después esperaba a que le creciera el cabello lo suficiente para que fuera capaz de ocultar el mensaje grabado y luego era enviado a la ciudad de destino; cuando el esclavo llegó ante Aristógoras (receptor), fué rasurado y de esta manera se pudo leer el mensaje, el cual indicaba que iniciara una revuelta contra el rey de Persia.

Otro método consistía en ocultar el mensaje en tablas de madera que eran cubiertas con cera, para poder leer el mensaje se debía quitar la cera y de esta manera poder leer el mensaje.

El alemán Johannes Trithemius es considerado el fundador de la esteganografía moderna. En su libro "Steganographia", escrito en el año 1499, describe un sistema esteganográfico avanzado, pero debido a la temática general de libro que era sobre magia y métodos de aprendizaje acelerado, no se tomó demasiado en serio dicha publicación.

Otro aporte a los métodos esteganográficos fue el realizado por el italiano Girolamo Cardano en el siglo XVI, en el cual inventó la Grilla de Cardano, consistía en una papel con agujeros en posiciones determinadas, previamente acordadas entre el emisor y receptor, al colocar el papel sobre un texto escrito se podían observar las letras que conformaban el mensaje oculto y sin levantar sospecha alguna.

Posteriormente, las diferentes guerras internacionales dieron lugar al desarrollo de diversos métodos esteganográficos.

Durante la Guerra Franco-Prusiana, Paris se encontraba incomunicada con el resto de Francia, si algún cartero transportaba mensajes, era capturado por los prusianos, por tal motivo, el ejercito parisino encontró una manera de poder enviar mensajes secretos a través de palomas mensajeras.

En la primera Guerra Mundial se utilizaron diversos métodos esteganográficos como las Grillas rotantes, que eran una evolución de las Grillas de Cardano, en las cuales se escribía un texto en forma de cuadrícula que se hacía rotar en la hoja utilizada. También se empleó el envió de mensajes con personas que ocultaban documentos escritos con tinta invisible en sus prendas de vestir o zapatos.

Durante la Segunda Guerra Mundial fue más eficaz el envío de mensajes secretos mediante el uso de técnicas esteganográficas que criptográficas, tales como el microtexto en un punto.

Actualmente, las fuerzas de seguridad de diversos organismos y países emplean mensajes esteganográficos incorporados en comunicaciones, con el objetivo de evitar el conocimiento de las mismas por parte de algún enemigo que constituya una amenaza de peligro.

En la actualidad el uso de esteganografía se emplea en técnicas para ocultar información en archivos digitales de formato multimedia (audio, imagen, video), y existe una gran cantidad de aplicaciones distribuidas a través de Internet que permiten aplicar diversas técnicas esteganográficas en archivos multimedia.

Otro ejemplo es el de la industria discográfica, la cual tiene especial interés en el desarrollo de técnicas esteganográficas que permitan ocultar marcas de copyright, en archivos como películas, audio, libros y otro archivos para intentar disminuir la distribución de copias ilegales.

Con el avance de las comunicaciones e Internet, la evolución de la esteganografía ha tenido cierto interés como método para mantener comunicaciones secretas.

## 2.3. Principio de la esteganografía

El concepto de canal encubierto fue introducido por Simmons [40] en 1983, como el problema de los prisioneros. Alice y Bob son cómplices de un delito por lo que han sido detenidos y son encerrados en diferentes celdas. Ellos quieren desarrollar un plan de escape pero desafortunadamente su única manera de comunicación después de que son encerrados será por medio de mensajes transmitidos por una guardia de nombre Wendy; cualquier tipo de comunicaciones entre ellos será vigilada y ella no les permitirá comunicarse a través de mensajes cifrados, en el momento en que Wendy note algún tipo de comunicación sospechosa los confinará a una celda solitaria y cancelará cualquier intercambio de mensajes.

Por tal motivo, ambas partes deberán hacer invisible su comunicación para no levantar sospechas en Wendy; en este escenario los prisioneros deberán establecer un canal subliminal en los mensajes a la vista del guardia, además de que deben evitar ser engañados por mensajes generados o modificados por ella.

La comunicación esteganográfica se realiza debido a la existencia de un intruso, es decir, involucra a una tercera persona que tiene acceso a la información transmitida a través del canal de comunicación. Si no existiera el intruso, el emisor y receptor podrán comunicarse de forma abierta, sin hacer uso de la esteganografía.

## 2.4. Esquema esteganográfico

Se define como esquema esteganográfico al conjunto de componentes que permiten llevar a cabo el proceso de la comunicación esteganográfica.

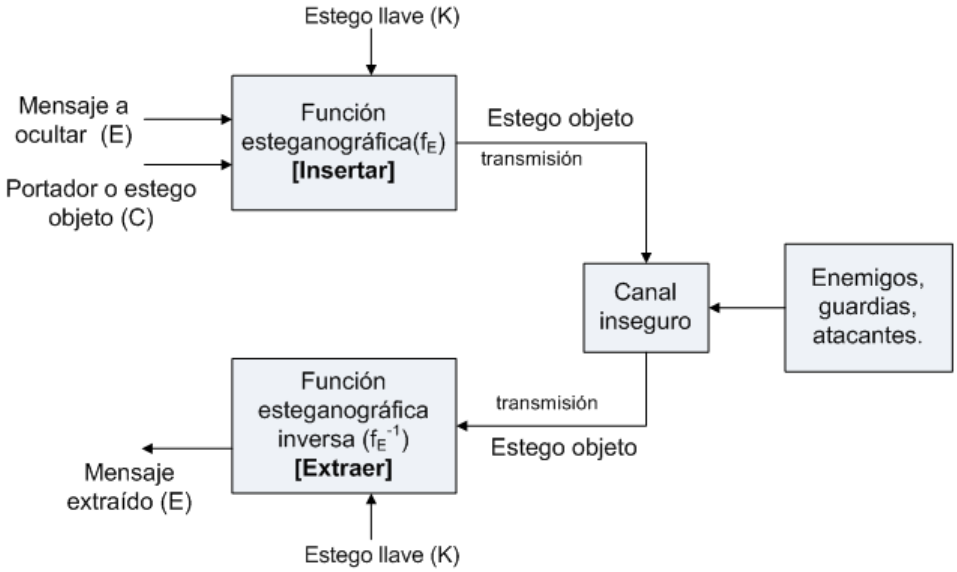


Figura 2.2: Esquema esteganográfico

La figura 2.2 muestra el esquema esteganográfico el cuál se explica a continuación.

Partiendo del hecho de que un emisor desea enviar un mensaje oculto a un receptor a través de una canal inseguro, inicialmente se debe tener un mensaje a ocultar, después se debe seleccionar el portador u objeto contenedor, que es la entidad o el conjunto de datos que son susceptibles a ser alterados para incorporarles el mensaje que se desea ocultar, el portador puede ser de varios tipos de datos o formatos.

Una vez que se tiene el mensaje y se ha seleccionado el portador se aplica un algoritmo esteganográfico (función esteganográfica o estego algoritmo) que realizará el procedimiento para insertar el mensaje a ocultar dentro del portador seleccionado.

El algoritmo esteganográfico puede usar una clave esteganográfica (estego-llave), esto depende de la selección del protocolo esteganográfico seleccionado; posteriormente se obtendrá como resultado un estego objeto, que creará un objeto portador con información oculta, la acción de ocultar el mensaje dentro del portador se denomina insertar.

Ya que se tiene el estego objeto, éste será enviado a través de un canal inseguro que es vulnerable a cualquier tipo de ataques; cuando el estego objeto llega a su destino, el receptor deberá recuperar el mensaje oculto a través de un algoritmo esteganográfico inverso, la acción de la recuperación posterior del mensaje oculto se denomina extraer. El algoritmo esteganográfico debe ser previamente conocido por el emisor y el receptor.

Se puede hacer uso de la criptografía durante un proceso esteganográfico para garantizar una mayor protección de la información, para ello debe existir una estego llave la cual es previamente conocida por el emisor y el receptor.

### 2.4.1. Tipos de portadores

Según la naturaleza del portador que se usa para ocultar información, se pueden clasificar en dos tipos: estructurados y no estructurados.

#### 2.4.1.1. Portadores no estructurados

Estos portadores carecen de una sintaxis y semántica objetiva y no tienen ninguna estructura de datos, por lo que su interpretación esta determinada por diversos parámetros, generalmente una persona interpreta este tipo de portadores a través de sus sentidos.

En este tipo de portadores no existe un parámetro de cuanta información se puede ocultar, debido a que implica el análisis del nivel alteración de la percepción para cada contenido del portador.

Algunos de los portadores no estructurados son los siguientes:

- Imágenes
- Audio
- Video

#### 2.4.1.2. Portadores estructurados

Son aquellos portadores que su semántica y sintaxis han sido previamente definidas de manera que es interpretado por un sistema informático; se aplican reglas sintácticas que definen la estructura y el formato del portador, y así determinar si es válido o es correcto con respecto a su especificaciones.

Algunos ejemplos de este tipo de portadores son:

- Documento
- Protocolos de redes

## 2.5. Clases de protocolos esteganográficos

Existen tres protocolos esteganográficos:

- **Esteganografía pura:** Es un sistema en el cual no se requiere de un previo intercambio de información, tal como una estego llave, es necesario que el algoritmo de inserción y extracción del mensaje secreto solo sea conocido por los usuarios legítimos (emisor y receptor) de la comunicación.
- **Esteganografía con llave privada:** Es un sistema que combina el uso de la esteganografía pura con criptosistemas simétricos, en el cual una clave controla el acceso a la información; el mensaje se oculta usando una llave privada que proporciona más seguridad, dado que tanto el emisor como el receptor conocen la llave no es necesaria su transmisión.

- **Esteganografía con llave pública:** Es un sistema que combina el uso de la esteganografía pura con criptosistemas de llave pública, en el cual se requieren dos claves una privada y otra pública, la clave pública se usa para ocultar la información y la privada para reconstruirla. En este caso existe la necesidad de transmitir la llave pública a través del medio.

## 2.6. Técnicas esteganográficas

Existen tres técnicas que se pueden utilizar para hacer esteganografía: sustitución, inyección y generación de nuevos ficheros.

### 2.6.1. Sustitución

Cuando se crea un archivo, éste contiene ciertas áreas de datos que no son utilizadas, o que si se alteran no presentan cambios visuales o estructurales, dichas áreas pueden ser reemplazadas en el archivo original, permitiendo ocultar información dentro del archivo.

Un ejemplo es el método del bit menos significativo (LSB), el cual sustituye el último bit de cada byte, de manera que se puede repetir este proceso con cada byte sin que el ojo humano perciba diferencia alguna. Es uno de los métodos más usados en la esteganografía sobre imágenes digitales.

### 2.6.2. Inyección

En este método se agrega la señal o mensaje secreto directamente en el objeto portador. El principal problema reside en que generalmente ésto hace que el archivo crezca de tamaño en comparación con el archivo original. Si bien, no es un factor determinante, sí hace que sea una desventaja frente a otros métodos.

### 2.6.3. Generación de nuevos ficheros

Esta técnica implica tomar el mensaje y usarlo para generar un nuevo archivo desde cero. Una de las ventajas de este método es que no existe un archivo original con el que comparar.

## 2.7. Características

Existen diversos protocolos y técnicas de inserción de información que permiten ocultar información en un objeto determinado, sin embargo, todos los protocolos y técnicas deben cumplir ciertas características para que la esteganografía se pueda aplicar correctamente.

A continuación se enlistan los principales requerimientos que las técnicas esteganográficas deben satisfacer:

- **Invisibilidad:** Corresponde al hecho de ocultar la existencia de una comunicación hacia terceras personas, solo el emisor y el receptor tienen conocimiento de la presencia de un mensaje secreto.
- **Confiabilidad:** Es la probabilidad de que un algoritmo de extracción esteganográfico dé como resultado el mensaje secreto de manera que éste sea correcto.
- **Robustez:** Es el grado de inmunidad de un estego objeto frente a posibles alteraciones realizadas por terceras personas.
- **Capacidad para ocultar:** Corresponde al número de bits máximo que pueden ser ocultos en un portador determinado.

## 2.8. Aplicaciones

El uso de esteganografía aplicado en el campo de la información digital tiene diversos ámbitos, sin embargo, puede ser usada para malas intenciones.

Un uso poco ético para el que se a empleado la esteganografía es en el robo de información ya sea en el ámbito corporativo, militar o gubernamental.

Son numerosos los ejemplos de las aplicaciones de la esteganografía en los ámbitos militares y de espionaje, estando presentes en la vida cotidiana.

No todas las aplicaciones de la esteganografía son con fines maliciosos. Por ejemplo, se puede usar para insertar información de pacientes en radiografías,Tomografías Axiales Computarizadas (TAC's), entre otras, también se puede usar para ser integrada en mecanismos de autenticación o clasificación de contenidos multimedia.

Otro ejemplo es la impresión de dibujos complicados en los billetes para evitar su falsificación, algunos de ellos con tintas que solo son visibles bajo una iluminación especial.

La esteganografía puede proporcionar la integridad y autenticación de datos, detectando si algún objeto, entidad o conjunto de datos ha sido manipulado de algún modo e identificar al dueño o propietario. Un ejemplo de una aplicación basada en la integridad y autenticación se encuentra en el campo de la seguridad y vigilancia frente a robos u otros delitos semejantes.

También se utiliza en la protección de información frente a copias ilícitas, siendo la protección de los derechos de propiedad intelectual la aplicación más común de las marcas de agua digitales.

Otra aplicación interesante sobre el uso de la esteganografía consiste en proveer a los usuarios de múltiples niveles de acceso a la información. Estas técnicas se pueden usar para crear canales secretos de información accesibles solo a determinados usuarios, ampliando de esta forma la cantidad de información transportada en los objetos. Por ejemplo en una película difundida en un canal de televisión podría incorporar bandas sonoras en múltiples idiomas.

La esteganografía ha sido puesta a la vanguardia en las técnicas actuales de seguridad debido al notable crecimiento del poder computacional y el incremento del conocimiento sobre la seguridad.

## 2.9. Herramientas esteganográficas

Actualmente existe una gran variedad de herramientas esteganográficas que facilitan ocultar información en diferentes tipos de portadores. La tabla 2.1 muestra una lista de herramientas esteganográficas para diversos tipos de portadores.

Tabla 2.1: Herramientas esteganográficas diversas.

| **Herramientas esteganográficas** | **Medio portador** |
|---|---|
| GZSteg | Archivos .gz |
| InfoSteg | Imagen, audio, video |
| KPK File | Word, BMP |
| S-Mail | Archivos .exe y .dll |
| Hiderman | Diferentes medios de archivos |
| StegMark | Imagen, audio, video |
| Steghide | JPEG, BMP, WAV, AU |
| S-Tools | BMP, GIF, WAV |
| Hydan | Programas binarios |
| Covert.tcp | TCP/IP |

En [15] se realiza un estudio sobre las diferentes herramientas para esteganografía y estegoanálisis, desde un punto de vista forense para poder identificar qué herramientas están disponibles en Internet y cuáles de ellas podrían ser utilizadas por organizaciones terroristas.

Dichas herramientas están clasificadas de acuerdo a los diferentes tipos de portadores, e indica el tipo de licencia que tienen, así como si dispone de código fuente o no.

Cabe mencionar que actualmente la mayoría de herramientas existentes para la esteganografía de red están enfocadas principalmente para computadoras con sistemas operativos basados en Linux.

## 2.10. Definición de canal encubierto

La evolución en las redes de computadoras en los últimos años ha ocasionado el desarrollo de nuevos servicios, pero también de manera simultánea han surgido nuevas amenazas para los sistemas que se encuentran interconectados.

Un canal de comunicación es el medio que se utiliza para transmitir un mensaje de un emisor hacia un receptor. Los canales encubiertos también llamados *covert channel* son una manera de crear una comunicación oculta que puede vulnerar la integridad de un sistema; dicho concepto fue introducido por primera vez en el año de 1973 por Lampson.

La definición de Lampson [26], describe un canal encubierto como uno que se utiliza para la transmisión de información, pero que no está diseñado ni destinado para las comunicaciones.

En [35], en 1985 de acuerdo con una publicación del Departamento de Defensa de los EE.UU, titulada "Evaluación de sistema informático de confianza", un canal secreto se define como:

> ". . . Cualquier canal de comunicación que puede ser explotado por un proceso de transferencia de información de manera que viola la política de seguridad del sistema."

El uso de canales encubiertos como método para ocultar información ha sido considerado una amenaza para la seguridad, ya sea en entornos de red o en sistemas centralizados, a pesar de ello, también pueden ser tratados como alternativa para proteger y conservar la integridad de los datos.

## 2.11. Clasificación de canales encubiertos

Los canales encubiertos pueden clasificarse de acuerdo al mecanismo de ocultación en:

- **Canales encubiertos de almacenamiento:** Son aquellos canales en los que el proceso de la comunicación secreta entre el emisor y el receptor se lleva a cabo alterando el valor de las variables de un recurso compartido o un atributo almacenado o transmitido.

  Como ejemplo de este tipo de canales se tiene: la alteración en ciertos campos que no son utilizados en la cabecera de un protocolo de comunicación, o la modificación de los colores que componen a cada uno de los píxeles de una imagen en un archivo.

- **Canales encubiertos de temporización:** Son aquellos canales en los que el proceso de la comunicación secreta entre el emisor y el receptor se lleva a cabo modificando el período de tiempo de ejecución de un determinado proceso o tarea.

  Como ejemplo de este tipo de canales se tiene: el tiempo de uso de la unidad central de proceso (CPU) o el tiempo de comienzo de un proceso.

Esta clasificación es la más conocida, sin embargo, hay otras clasificaciones de canales encubiertos, existen diversos trabajos dedicados al análisis de los canales encubiertos y a sus clasificaciones.

Por otra parte, hay una taxonomía basada en la cantidad de ruido que puede afectar a un canal encubierto, ésta conformada por los canales sin ruido y los canales con ruido. Sin embargo, se considerá que este tipo de clasificación no es adecuada en el área de las redes informáticas.

Otra clasificación se encuentra enfocada en el número de procesos que establecen una comunicación simultánea haciendo uso de zonas de memoria de una variable, esta agrupación la conforman los canales agrupados y los canales desagrupados.

Otra categoria de canales encubiertos es la propuesta por Venkatraman [42], definé los canales espaciales y canales temporales. En 2002, Ahsan [1] propone un canal encubierto llamado canal de ordenación en el cual se almacena la información secreta dentro de las diversas odernaciones pósibles de los paquetes de datos que realizán una comunicación a través de una red.

Dado que no existe una nomenclatura definida sobre la clasificación de canales encubiertos, diferentes autores han hecho uso de las taxonomías que más convenían en sus estudios [4], [14], [27], [30], [31], [45].

## 2.12. Características de los canales encubiertos

Los canales encubiertos presentan ciertas características:

- **Capacidad:** Es la cantidad de información que puede ser transmitida a través del canal.
- **Ruido:** Es la cantidad de perturbaciones que pueden interferir con la información mientras es transmitida a través del canal.
- **Modo de transmisión:** Puede ser síncrona, donde la transferencia de información es controlada por una señal de reloj, de otro modo es asíncrona.

La capacidad es una parte muy importante de la calidad global de un canal. Desde el punto de vista de seguridad, un canal de mayor capacidad hará posible que más información se filtre.

## 2.13. Condiciones para canales encubiertos

Existen algunas condiciones que deben cumplirse para que sea posible la existencia de canales encubiertos:

1. **Potencial para la comunicación:** Se refiere al hecho de que entre el emisor y el receptor se pueda efectuar una comunicación.

2. **Restricción sobre la comunicación:** No esta permitida una comunicación entre el emisor y el receptor en circunstancias normales debido a la política de seguridad.

3. **Existencia de una variable o recurso compartido:** Debe haber algún recurso compartido de manera general dentro del sistema de comunicación entre emisor y el receptor.

4. **Acceso pleno sobre la variable o recurso compartido:** Un recurso compartido debe ser visible ante el remitente y el receptor. También el remitente debe ser capaz de alterar el recurso de alguna manera, y el receptor debe ser capaz de notar dicho cambio.

5. **Capacidad para sincronizar:** El emisor y el receptor deben ser capaces de sincronizar sus operaciones a fin de que la transmisión tenga lugar.

## 2.14. Ejemplos del uso de canales encubiertos

Algunos ejemplos del uso de canales encubiertos se han dado desde hace mucho tiempo.

En la 1ª Guerra Mundial, los rusos utilizaron canales encubiertos en sus radio frecuencias nacionales para comunicar mensajes cortos a las fuerzas militares. Por lo general, ocultan sus canales con ruido aleatorio de modo que el canal secreto no produce ninguna sospecha en sus enemigos.

De manera similar, la transmisión de datos sobre los sistemas de difusión de televisión se puede utilizar para mantener señales de vídeo o audio de baja frecuencia, de manera que la señal original no se distorsione. Dado que las transmisiones de televisión digital contienen ruido aleatorio, si la transmisión encubierta (codificada para ser aleatoria) no interfiere con la señal original, unicamente generará ruido aleatorio, y por lo tanto no puede producir ningún tipo de sospecha.

Por otra parte, la interferencia electromagnética puede ser utilizada para crear canales encubiertos y poder comunicarse de manera secreta.

## 2.15. Aplicaciones

Por ahora, los canales encubiertos se utilizan únicamente para la comunicación a distancia de una señal portadora de confianza. Por lo tanto, tiene pocas aplicaciones legales y varias ilegales como:

1. La comunicación entre los cuerpos de seguridad con el fin de ocultar sus movimientos y técnicas.
2. Espionaje.
3. Robo de información.
4. Transmisión de datos encriptados sobre una comunicación secreta, segura e indetectable.
5. Hacking.
6. Fuga de información de una organización o sistema.
7. Ocultación de información para negar su existencia.

## 2.16. Esteganografía de protocolo

La esteganografía de protocolo o de red, se basa en transmitir mensajes a través de canales encubiertos en paquetes o bits sobre un tráfico de red. Estos mensajes pueden ser textuales, comandos, datos estructurados o no estructurados, o cualquier otra señal conocida por el receptor.

El único propósito es tener una sesión de comunicación anónima entre un emisor y un receptor por una razón determinada, que puede ser por una comunicación de negocios protegida, defensa o actividades delictivas.

## 2.17. Explotación de un protocolo

La explotación de un protocolo en el contexto de Internet y seguridad de redes, es el concepto de tomar ventaja de los protocolos y estándares que regulan la transmisión de datos entre computadoras, y manipular dichos protocolos.

En la explotación de un protocolo se incorporan la estrategia y el diseño de un canal encubierto, ya que es una técnica eficaz para evadir al NIDS (Sistema de Detección de Intrusos de Red).

### 2.17.1. Razones para hacer uso de la explotación de un protocolo

- Debido a la explotación de un protocolo, los canales encubiertos tienden a tener menos ruido dentro de ellos, en comparación con otros portadores de señal.
- Los canales encubiertos pueden permanecer dentro de los protocolos.
- Los canales encubiertos pueden ser usados fácilmente por los involucrados en la comunicación.
- Debido a la amplia difusión de Internet como una red de datos, siempre se puede encontrar un canal encubierto en cualquier parte del mundo.

### 2.17.2. Ambigüedad en los protocolos de red

La ambigüedad en un protocolo hace que sea un fuerte candidato para su explotación.

Esta ambigüedad surge debido a las diferentes implementaciones que tiene un protocolo, en el que las diversas prácticas de implementación llevan a diferentes manejos de condiciones erróneas y excepcionales de una manera no estructurada. Por lo que algunas de las características o etapas del protocolo se vuelven inconsistentes.

También la mayoría de las implementaciones no llevan a cabo la ejecución completa de la máquina de estado del protocolo tal como se define en su RFC, por lo tanto abre oportunidades en la implementación del protocolo, que cuando son explotados, el protocolo entra en un estado indeterminado.

Los estados extraños dentro de un protocolo no contribuyen al funcionamiento real de dicho protocolo, por lo que también pueden ser explotados en la comunicación secreta y por lo tanto aumentan la ambigüedad en el protocolo.

### 2.17.3. Técnica para evadir la seguridad de red mediante la explotación de un protocolo

La explotación de un protocolo también se utiliza con el fin de eludir los nodos de seguridad o gestión de red como firewalls, servidores proxy, NIDS, pasarelas, entre otras.

- Un nodo de seguridad no puede realizar un análisis completo de todo el comportamiento para un protocolo en particular. Por lo tanto, no puede detectar cierta actividad que tenga un comportamiento especifico que no se analizó y procesó por el nodo de la seguridad.
- A menos que el nodo de seguridad conozca un sistema de implementación del protocolo completo, no puede saber si dicho sistema manejará un tráfico particular o inusual, ya que las implementaciones de protocolo pueden variar considerablemente.
- Si un nodo de seguridad no conoce la topología de la red nunca puede saber si el tráfico llegará a su destino previsto.

La cuestión principal es cómo modificar los paquetes de protocolo sin crear sospechas en los sistemas intermedios (switches, routers, firewalls, IDS, etc.). También cómo el emisor puede eludir intrusos en las redes de modo que sus paquetes sean recibidos con éxito por el receptor.

Hay una gran comunidad que trabaja activamente en este campo de la seguridad, por una parte están desarrollando nuevas técnicas de esteganografía sobre protocolos; mientras que por otra parte está trabajando en métodos para detectar este tipo de tráfico que está utilizando la esteganografía sobre protocolos.

La esteganografía sobre protocolo se puede aplicar en cualquier capa, se puede hacer en tramas de Ethernet, en las tramas IP, en las tramas del protocolo de transporte, o en las tramas de protocolos de capas superiores. Así, la idea principal es aprovechar los canales encubiertos en estos protocolos tales que su uso sólo es conocida por la parte receptora.

## 2.18. Criterios para la selección de un protocolo

Para el diseño de un protocolo basado en un canal encubierto, se debe elegir un protocolo. Sería inútil hacer un análisis exhaustivo de todos los protocolos para ver si es un candidato adecuado.

Tener conocimiento de las especificaciones del protocolo y sus ambigüedades es útil y la selección puede limitarse en base a unos criterios:

1. El protocolo debe ser manipulado de tal manera que no afecte las operaciones de la red o el sistema. Si así fuera, existe la posibilidad de que el tráfico resultante exhibiera características anómalas de manera abierta, que podrían ser observadas por un sistema de detección de intrusos (IDS); también, existe la posibilidad de que el canal encubierto pierda los paquetes durante la transmisión.

2. El atributo del protocolo que se explota para convertirse en un canal encubierto debe tener suficiente espacio para permitir un ancho de banda adecuado para los datos a ocultar. El atacante también debe diseñar una esquema de codificación/decodificación para los datos adjuntos dentro del atributo explotado.

3. La relación señal/ruido (SNR) debe ser aceptable. Este factor tiene que ver más con canales encubiertos ruidosos que los canales encubiertos silenciosos. La pregunta es cuánto ruido del canal encubierto se puede tolerar antes de que la información que se envía sea confusa e ilegible por un receptor.

4. El canal secreto debe tener suficientes privilegios o permisos para operar en el sistema destinado. Por ejemplo, si el sistema de destino es en una máquina Linux, el canal encubierto debe contar con el software necesario para ser ejecutado con privilegios de root y poder comunicarse con el sistema en el otro extremo del canal.

5. Mientras menor sea la correlación, más fiable es un canal encubierto.

Por lo tanto, con los puntos antes mencionados es poco lo que hay que tener en cuenta para la selección de un protocolo.

## 2.19. Familia de protocolos TCP/IP

En Internet se encuentran conectadas máquinas totalmente diferentes, tanto en hardware como en software, recorriendo diferentes sistemas operativos, la familia de protocolos TCP/IP hacen que sea posible la comunicación entre ellas, traduciendo la información a un lenguaje común.

El conjunto de protocolos TCP/IP originalmente fue creado con fines militares y está diseñado para cumplir con una ciertas características entre las cuales destacan: dividir los mensajes en paquetes de datos, utilizar un sistema de direcciones, enviar datos a través de la red y detectar errores en las transmisiones de datos.

TCP/IP esta organizado en cuatro capas conceptuales como podemos observar en la figura 2.3.

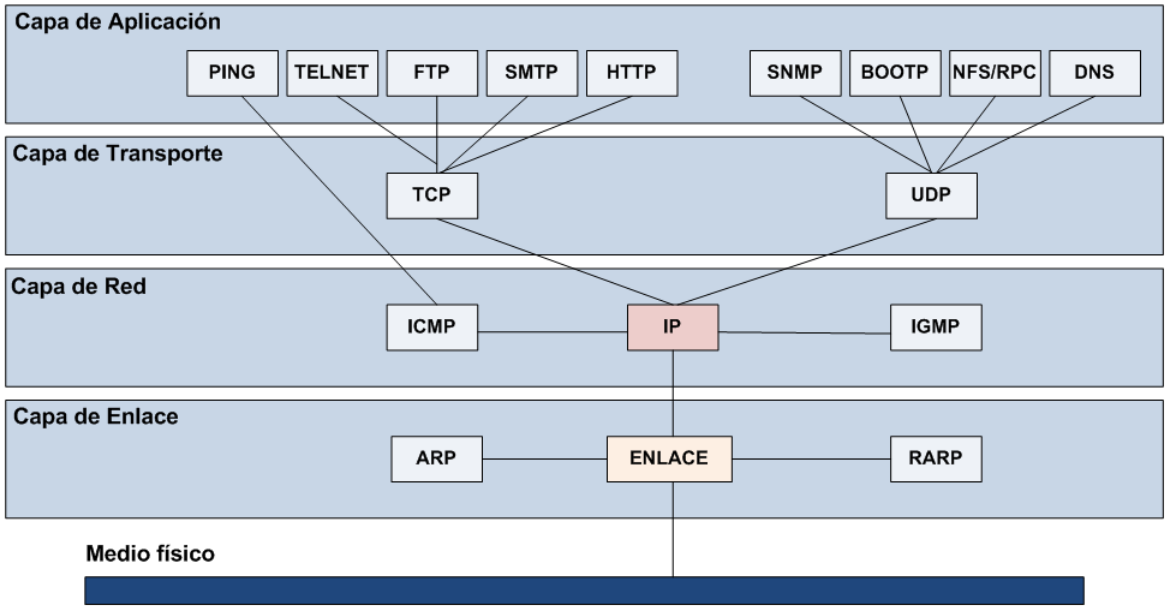


Figura 2.3: Modelo TCP/IP.

- **Capa de Aplicación:** En esta capa los usuarios llaman a una aplicación para que acceda a los servicios disponibles en la red, dichas aplicaciones interactuan con la capa de transporte, seleccionando en cada programa de aplicación el tipo de transporte necesario y traspasando sus datos de la forma requerida.

  En este nivel se encuentran protocolos destinados a proporcionar servicios, tales como correo electrónico (SMPT), transferencia de archivos (FTP), conexión remota (TELNET) y otros como el protocolo HTTP.

- **Capa de Transporte:** Los protocolos de este nivel se encargan de proporcionar comunicación entre los programas de aplicación. En este nivel se regula el flujo de información proporcionando fiabilidad en el transporte, asegurando que los datos llegan sin errores. Para ello los datos se dividen en paquetes añadiendo un código de control que se valida en el destino, de manera que si algún paquete es erróneo éste se vuelve a enviar. En estos paquetes se añade la dirección destino que se utilizará para la transmisión.

  En esta capa se encuentran protocolos tales como TCP y UDP.

- **Capa de Red:** En esta capa se maneja la transmisión de datos desde una máquina a otra. En esta capa se recibe la solicitud de la capa de transporte de enviar un paquete en el que se ha incluido la dirección destino. Esta capa verifica estos paquetes y aplicando un algoritmo de ruteo los direcciona al lugar de destino.

  Aquí se encuentra el protocolo IP.

- **Capa de Enlace:** Los protocolos que se encuentran en este nivel son los encargados de la transmisión a través del medio físico al que se encuentra conectado cada equipo.

## 2.20. Técnicas generales de esteganografía sobre protocolos de red

Existen diversas maneras de explotar un protocolo para propósitos esteganográficos, a continuación se describen algunas de ellas.

### 2.20.1. Alteración de los campos o atributos de la cabecera de un protocolo

Es la técnica más común, se trata de producir cambios en los campos de la cabecera de un protocolo de manera que no cambian las operaciones de la red o del sistema. Por lo tanto, estas modificaciones sólo deben ser llevadas a aquellos campos que no modifiquen las funciones del flujo de tráfico en una red.

Los canales encubiertos en este caso son fáciles de detectar debido al hecho de que los cambios en los atributos, como una función directa del mensaje, producen un tráfico correlacionado que un nodo de seguridad puede notar fácilmente. Por lo tanto, es esencial codificar el mensaje de tal manera que se convierta en aleatorio. De esta manera los atributos o campos con cierto ruido son buenos candidatos para ocultar un mensaje.

### 2.20.2. Envío de mensajes ocultos como función del cambio en el estado del protocolo

Otra técnica interesante que no es tan fácil de detectar, es el uso de canales encubiertos en las transiciones de estado de un máquina de estados de un protocolo, son muy parecidos a los canales encubiertos de temporización.

Por lo general, ofrecen muy poco ancho de banda para la transmisión de mensajes secretos, y por lo tanto sólo se pueden usar para enviar distintos comandos a un sitio remoto.

Trabajan solamente sobre protocolos con estados, por lo que los protocolos con máquinas de estado complejas son ideales de usar para este tipo de técnica.

### 2.20.3. Envío de mensajes ocultos como función del orden de la secuencia de paquetes

Un mensaje puede ser codificado de una manera que permita ser reproducido por el receptor si éste recibe el tráfico en un cierto orden. Por lo tanto el orden de la secuencia del paquete es el canal encubierto.

Esta técnica proporciona mayor resistencia al canal encubierto, permitiendo que sea altamente indetectable; esta basada en la suposición de que es poco probable que esté disponible en la mayoría de las configuraciones de red. La suposición es que la red debe proporcionar garantía para transmitir el tráfico sin reordenar los paquetes.

La mayoría de los enrutadores y conmutadores utilizan técnicas de programación que no se preocupan por el orden del tráfico, por lo tanto, los errores de secuencia es el punto débil de esta técnica.

Sin embargo, si la red proporciona la suposición básica, entonces esta técnica es una buena opción para ser usada.

### 2.20.4. Protocolo de túnel

Es diferente del protocolo normal de tunelización. Básicamente consiste en encapsular un protocolo dentro de otro con la única diferencia es que el cuerpo intermedio desconoce que el flujo contiene un mensaje secreto, ya que todo lo que ve es un flujo normal con un área de datos inocente.

Protocolos de alto nivel como HTTP, FTP y TELNET pueden ser buenos candidatos para esta técnica, sin embargo ICMP también puede ser utilizado para hacer un túnel.

Esta técnica se utiliza sobre todo para la mensajería interactiva como charlas seguras u ocultas, o una sesión de comandos shell.

### 2.20.5. Asignación de mensajes secretos en el área de datos de un protocolo

Es una técnica altamente sofisticada en la que el mensaje secreto es incorporado al área de datos de un protocolo de manera que parece normal, y de esta forma no hay ninguna correlación entre la carga de datos posteriores por lo que no hay relación entre los bloques subsiguientes con mensajes secretos.

Esta técnica puede evadir fácilmente la seguridad de la red, pero todo depende de la función que se utilice para incorporar el mensaje secreto.

### 2.20.6. Asignación de mensajes secretos en una secuencia de paquetes

Un mensaje secreto puede ser transmitido como una secuencia particular o única de paquetes multi protocolo. Por lo tanto aquí el mensaje secreto no es obvio, ya que se transmite como una secuencia de paquetes de varios protocolos y cada secuencia corresponde a un bloque de mensaje. Aunque en esta técnica el orden de los paquetes puede ser necesario.

ICMP, ARP, TCP/IP, y otros paquetes de protocolos de alto nivel pueden ser usados para generar varias secuencias de paquetes de datos y ser enviados en las señales del destinatario para obtener un único mensaje secreto.

## 2.21. Canales encubiertos en algunos protocolos de red

### 2.21.1. Protocolo de Resolución de Direcciones (ARP)

ARP y RARP son de los protocolos mas utilizados en las redes Ethernet. Por lo tanto pueden ser utilizados como canales encubiertos.

Por ejemplo, el emisor puede generar un paquete de solicitud ARP para la dirección MAC de destino con su propia dirección IP de origen y direcciones MAC, entonces extrae la dirección IP de origen y direcciones MAC, de tal manera que las direcciones resultantes se conviertan en otras direcciones de host en la red.

Debido a que ARP transmite la petición a cada host, también será recibido por el destinatario, si también se encuentra en la misma red. Por lo tanto si el destinatario conoce la dirección IP de origen y las direcciones MAC, se puede calcular las discrepancias en las direcciones de origen y relacionarlo con un mensaje secreto.

### 2.21.2. Protocolo de Internet (IP)

El protocolo IP se compone de diversos campos y atributos, y cada uno contribuye al flujo de tráfico en diferentes niveles y con distinta importancia. Por lo tanto existen varias formas para crear canales encubiertos en el protocolo IP.

Algunas de ellas son las siguientes:

1. **Canales encubiertos en los campos de opciones y de relleno.** Si estos campos están vacíos, se pueden utilizar para contener un mensaje secreto.

2. **Canal encubierto en el campo de comprobación.** Como se indica en [2], el campo de comprobación también se puede utilizar para contener un canal encubierto, si se usa de manera adecuada.

3. **Canal encubierto en la parte de superposición de fragmentos consecutivos.** En [11] se describe una técnica para obtener un canal encubierto con un alto ancho de banda, pero solo si no hay ningún nodo intermedio que pueda montar y fragmentar el flujo por sí mismo.

4. **Canal encubierto en el campo de identificación.** Como se indica en [37], también se propone la creación de un canal encubierto en el campo de identificación si se utiliza de una manera adecuada.

### 2.21.3. Protocolo de Mensajes de Control de Internet (ICMP)

Este protocolo es el protocolo encargado de detectar y reportar errores al protocolo IP, teniendo como principal objetivo comunicar los mensajes de error, solicitudes y respuestas entre nodos de la red. Dichos mensajes son enviados dentro de un datagrama IP.

El protocolo ICMP es el más adecuado de usar en la esteganografía de Túnel y en la asignación de mensajes secretos en el área de datos de un protocolo, debido a que cuenta con más de 29 tipos de mensajes de control y cada uno tiene su propio conjunto de sub mensajes. Por lo tanto estos tipos y sub tipos de mensajes se pueden utilizar de manera que se pueda crear un canal encubierto para realizar una encapsulación de datos secretos.

El principal problema es que los paquetes ICMP suelen caer en nodos de enrutamiento y pasarela, por tal motivo se puede filtrar el flujo del tráfico de una red a partir de las actividades normales dentro de la red de acuerdo a patrones históricos de tráfico, de manera que cualquier otro tipo de tráfico es marcado como malicioso y por tal motivo el paquete ICMP no podría llegar a su destino. Sin embargo, una solución es imitar el tráfico autorizado normal de manera que los mensajes que se envíen pasen desapercibidos.

### 2.21.4. Protocolo de Control de Transmisión (TCP)

El encabezado del protocolo TCP tiene un amplio conjunto de atributos, que al igual que en el protocolo IP, pueden crearse varios canales encubiertos dentro de su encabezado. Algunos de los más explorados son los siguientes:

1. **Canales encubiertos en los campos de número de "acuse de recibo" y "secuencia".** En [37] describe cómo los campos de número de secuencia y de acuse de recibo, pueden ser explotados para crear un canal encubierto con un bajo ancho de banda.

2. **Canal encubierto en el campo de marca de tiempo (timestamp).** En [12] se discute la posibilidad de utilizar el campo de marca de tiempo para mantener los datos secretos.

3. **Canal encubierto en el estado de TCP.** Dado que TCP es un protocolo con estado, los estados se puede utilizar para activar eventos para el receptor.

4. **Canal encubierto en el campo banderas (flags) de TCP.** La mayoría de los sistemas operativos utilizan la familia de protocolos TCP/IP, por lo que no se preocupan por las combinaciones inusuales de los campos banderas de TCP.

### 2.21.5. Protocolo de Transferencia de Hiper Texto (HTTP)

El protocolo HTTP es también una buena opción para crear canales encubiertos. Este método se basa en ocultar información dentro de la cabecera del protocolo HTTP.

## 2.22. Técnicas de transformación o codificación de bloques de mensajes

La mayoría de las veces los mensajes secretos producen una alta correlación entre los paquetes, esto se convierte en una razón para la detección de canales encubiertos.

Con el fin de disminuir la correlación, es necesario codificar los mensajes a ocultar de una manera en la que se conviertan en datos aleatorios, lo cual permitirá que no haya una correlación entre los bloques de mensajes.

Existen varias técnicas para transformar mensajes o codificarlos, algunas de estas técnicas son las siguientes:

- **Codificación diferencial**

El mensaje a ocultar puede ser codificado diferencialmente comenzando desde el primer carácter, los caracteres subsiguientes se restan del anterior y el valor de diferencia es guardado. De esta manera, dos caracteres idénticos serán codificados con diferentes valores si se encuentran precedidos por caracteres diferentes.

Para un mayor grado de una secuencia de no correlación se puede tomar una diferencia de dos o más caracteres precedidos.

- **Codificación de números aleatorios con XOR**

La operación XOR se puede utilizar para codificar un mensaje secreto con un conjunto de números aleatorios generados a partir de una semilla que sólo conocen el emisor y el receptor.

- **Automorfismo Toral**

El Automorfismo Toral es otra manera de mapear un mensaje secreto para eliminar la correlación en un flujo de datos.

- **Codificación aritmética**

Es un esquema que transforma las palabras de un texto de un mensaje oculto a números reales de acuerdo a las frecuencias de ocurrencia de caracteres en el mensaje.

Sin embargo, el mensaje codificado tiene un patrón de correlación detectable, pero debido al hecho de que el texto del mensaje está codificado en números reales de acuerdo con una tabla secreta de frecuencias de caracteres, es poco probable que aunque se detecte la información se pueda recuperar el mensaje original.

- **Codificación Lempel Ziv Welsh (LZW)**

LZW es más que un esquema de codificación, se trata de un algoritmo de compresión dinámica que puede comprimir un flujo de datos sin necesidad de una tabla de búsqueda predefinido. Por lo tanto, genera una tabla de búsqueda a medida que avanza.

Tampoco hay necesidad de transportar la tabla de búsqueda del lado del receptor, ya que generará su propia tabla cuando reciba el flujo codificado.

El flujo codificado tiene una correlación cercana a cero. Por lo tanto, la única persona que puede recuperar el flujo codificado es quien conoce el flujo y que es decodificado utilizando el algoritmo LZW.

- **Codificación dinámica Huffman**

Otro esquema de codificación de compresión que también se puede utilizar en la transformación de mensajes ocultos es el algoritmo de codificación dinámico Huffman. Tiene los mismos requisitos que la codificación LZW, pero funciona de forma diferente.

Se basa en el análisis de la entropía del mensaje, y por lo tanto, trata de reducir la longitud en bits de los diferentes caracteres del mensaje de acuerdo con sus frecuencias en el mensaje.

El mensaje secreto codificado tendrá caracteres de longitud variable, con una correlación mínima entre ellos. Sin embargo, como con la codificación LZW, si el esquema de codificación es conocido por un espía, puede recuperar fácilmente el mensaje secreto.

### 2.22.1. Criptografía simétrica

Es un método criptográfico en el cual se hace uso de una clave privada para cifrar o descifrar un mensaje, en este caso el mensaje secreto puede ser transformado usando cualquier algoritmo de cifrado asimétrico y la correlación del flujo cifrado dependerá directamente de la fuerza del algoritmo de cifrado usado.

El principal inconveniente con este tipo de sistemas de cifrado está relacionado con el intercambio de claves, ya que al momento en el que el remitente y el destinatario hayan intercambiado las claves pueden usarlas para comunicarse con seguridad, pero existe la posibilidad de que no haya un canal de comunicación que sea seguro para transmitirse las claves.

Ejemplos de algoritmos simétricos son:

- DES
- Triple DES
- RC5
- AES
- Blowfish
- IDEA.

## 2.23. Técnicas de selección de mensaje de bloque o bits

Uno de los parámetros de diseño esencial en el procedimiento de esteganografía sobre protocolos, es cómo seleccionar un orden de los mensajes de bloques o bits, debido a que ayuda en la producción de la aleatoriedad y el anonimato en el canal encubierto.

Una de las técnicas que se puede utilizar es mediante el uso de un generador de pseudo-aleatoriedad que decide el orden de los bloques o conjunto de bits, siendo ésta la manera más trivial y eficiente. Por lo tanto, es importante que cada bloque o conjunto de bits en el mensaje sean colocados en el orden indicado.

La replica de los bloques o bits puede ser hecha con el fin de aumentar la aleatoriedad y disminuir la correlación en el canal encubierto. Esta réplica también sirve como un mecanismo de tolerancia a fallos.

## 2.24. Gestión del mensaje a ocultar

Se requiere un mecanismo de control que proporcione operaciones de gestión para la transmisión de mensajes secretos a través de canales encubiertos de tal manera que ayude al emisor a realizar la transmisión con éxito y que sea únicamente recibido por el destinatario.

Este control es esencial para contrarrestar el reordenamiento, la caída, y los errores de transmisión en el tráfico.

Algunos de estos mecanismos de control son los siguientes:

### 2.24.1. Control de la señalización en otro canal encubierto

Las señales de control pueden ser transmitidas sobre otros canales encubiertos que pueden estar en paquetes separados, por otra parte, el control y el mensaje pueden ser codificados para convertirse en un datos secreto entero, y luego se pueden transmitir juntos.

### 2.24.2. Retroalimentación y reconocimiento de paquetes

Una pequeña petición de respuesta, como el estado del protocolo, puede ser desarrollado para comunicaciones secretas.

De esta manera el emisor puede hacer esperar por una respuesta de retroalimentación o reconocimiento de paquetes del receptor, después de "n" paquetes de tráfico secreto para transmitir los siguientes "n" paquetes o participar en la retransmisión si algún "m" fuera de "n" paquetes transmitidos previamente es eliminado.

## 2.25. Cuestiones de Diseño

Algunas cuestiones a considerar en el diseño de un canal encubierto se describen a continuación.

### 2.25.1. Ancho de banda

El ancho de banda de los canales encubiertos es una de las cuestiones fundamentales en la esteganografía de protocolo. Mientras mayor sea el ancho de banda más rápida será la transmisión del mensaje secreto.

La codificación del mensaje también puede reducir el ancho de banda resultante. De esta manera, esquemas de codificación como LZW y Huffman pueden ser usados no solo para transformar el mensaje sino también para comprimirlo y reducir su longitud. Algunas veces se convierte necesario para mantener un cierto ancho de banda bajo, con el fin de conservar el canal encubierto fuera de vista.

### 2.25.2. Aleatoriedad

El grado de aleatoriedad en el canal encubierto es uno de los puntos de interés más importantes en la manera de transmitir mensajes secretos sin conseguir ser detectados.

Esta aleatoriedad puede ser creada mediante la transformación del mensaje secreto en una serie de bits aleatorios a través de esquemas de codificación como los que se han mencionado anteriormente.

### 2.25.3. Grado de correlación entre los paquetes

Es importante también controlar la correlación dentro del flujo de paquetes que contiene el canal encubierto. Esta correlación puede ser creada debido al mismo canal encubierto o al mensaje que contiene.

Por ejemplo, si se utilizan los estados de TCP como un canal encubierto, entonces habrá cambios radicales en dichos estados, de manera peculiar pueden llamar fácilmente la atención de un nodo de seguridad en la red.

Por lo tanto, esto puede ser enmascarado mediante la reducción del ancho de banda del canal encubierto y mediante la introducción de ruido de tráfico aleatorio entre los paquetes portadores en el flujo.

### 2.25.4. Canales encubiertos ruidosos y silenciosos

Los canales encubiertos pueden ser ruidosos y silenciosos. Si el emisor y el receptor utilizan un canal secreto silencioso, esto significa que se comunican utilizando un recurso compartido unicamente exclusivo de ellos. Si utilizan un canal secreto ruidoso, significa que se comunican a través de un recurso compartido que no es exclusivo de ellos y que también está disponible para otros sujetos.

Los canales ruidosos son más difíciles de usar ya que el recurso compartido puede ser potencialmente modificado por personas que no pertenecen al canal encubierto, lo que hace más difícil para el emisor y el receptor distinguir la información extraña dentro del flujo del canal encubierto actual.

Por lo tanto, es importante en la selección del portador para la esteganografía de protocolo, que el canal encubierto sea silencioso. Sin embargo, los canales encubiertos silenciosos no siempre son la mejor opción porque muchos detectores y sistemas de seguridad sólo analizan el tráfico de red de los canales encubiertos silenciosos.

### 2.25.5. Grado de ruido en el tráfico

El ruido durante la transmisión de los paquetes portadores de los canales encubiertos ayuda mucho al anonimato del tráfico.

La mayoría de los sistemas de seguridad reducen sus parámetros de seguridad en los momentos de grandes cargas de red con el fin de salvarse de ser el cuello de botella en la red. Así, podemos utilizar este hecho para hacer la transmisión en ciertas cargas de red, o crear una carga que imite el ruido para nuestros canales encubiertos.

### 2.25.6. Pérdidas de paquetes

La mayoría de los procedimientos de esteganografía de protocolo asumen que sus paquetes portadores no se perderán durante su transmisión. Pero no es el caso en la realidad.

Un procedimiento esteganográfico debe tener algún diseño que se encargue de esta situación mediante la retransmisión de los paquetes perdidos o haciendo una replica de los bloques de mensajes en el canal encubierto de manera que si se pierde un bloque, pueda ser replicado y continúe el trabajo.

También un sistema de retroalimentación puede ser utilizado para saber qué bloque o bits son transmitidos satisfactoriamente y cuales requieren de una retransmisión.

### 2.25.7. Reordenamiento de paquetes en los nodos intermedios

Algunos procedimientos esteganográficos utilizan canales encubiertos que son sensibles a la reordenación de paquetes, y asumen que el receptor obtendrá los paquetes en el mismo orden de transmisión. Pero no siempre es así, especialmente cuando hay un router o un conmutador en el medio del camino.

Por lo tanto, debe haber algún mecanismo que cumpla con el ordenamiento de los paquetes en el nivel de red o que tenga la capacidad de que el receptor pueda reordenar los paquetes recibidos en su forma original.

### 2.25.8. Bloques de mensajes y bits

El tamaño de la unidad de transmisión de los datos secretos es otro importante problema de diseño que está directamente relacionado con el ancho de banda de canal encubierto. Algunos procedimientos prefieren conjuntos de bits, mientras que otros prefieren bits de manera individual. Ambos tienen sus ventajas y desventajas.

Tomando unidades de gran tamaño se incrementará el ancho de banda resultante, también aumenta la posibilidad de tener paquetes perdidos. Por otra parte, considerando que las unidades de menor tamaño (1 bit de ancho) reducen el ancho de banda efectivo de manera significativa, tienen una perdida muy pequeña o insignificante en la retransmisión de paquetes.

También, las grandes unidades presentan una mayor correlación entre sí (que es suficiente para ser detectado), mientras que las unidades más pequeñas muestran alta aleatoriedad. Es por eso que los canales encubiertos con unidades más pequeñas son más difíciles de detectar.

### 2.25.9. Uso de tráfico existente y tráfico generado

Otro parámetro de decisión es si se utiliza el tráfico existente o se genera uno nuevo para la transmisión de mensajes secretos.

Utilizar el tráfico existente significa espiar en la red y usar canales encubiertos existentes en el tráfico para la transmisión de mensajes secretos; también significa que estamos haciendo esteganografía por inyección o sustitución.

Es difícil de hacer esteganografía de propagación ya que tenemos que generar un nuevo tráfico que imite un tráfico normal diseñado para un propósito bajo circunstancias normales. Esto significa que tenemos que mostrar un patrón de actividad en el tráfico generado para que a los sistemas de seguridad les parezca normal e inofensivo.

### 2.25.10. Explotación del área de datos y los campos de la cabecera de un protocolo

Otro parámetro de diseño importante es que parte del tráfico debe ser utilizado como un canal encubierto, y se tienen dos opciones, el área de datos o el encabezado de un protocolo.

El área de datos de un protocolo resulta más atractiva debido al hecho de que los sistemas de seguridad por lo general trabajan sobre los encabezados, pero la inyección o la sustitución del área de datos puede distorsionar los datos originales requeridos por el protocolo.

Por otra parte, es muy fácil de usar campos de la cabecera como un canal encubierto, pero puede ser detectado.

### 2.25.11. Dependencia entre paquetes

Otro punto importante es que debe existir una dependencia mínima entre paquetes. Mientras menos dependencia haya en un procedimiento esteganográfico más resistente será ante fallos y por lo tanto hace que la reconstrucción de mensaje sea más probable y efectiva de llegar a su destino.

# Capítulo 3

# Flujo Alternativo de Datos (ADS)

La esteganografía tiene por objetivo el ocultar información en archivos, sin despertar la sospecha de terceros. Actualmente existe una gran variedad de herramientas esteganográficas que varían de acuerdo al tipo de archivos utilizados para ocultar la información.

El tamaño de la información a ocultar depende del tamaño del archivo portador; en algunos casos es 10 % del tamaño del archivo portador. Para poder recuperar la información oculta es necesario que el receptor cuente con la herramienta que se usó para ocultarla. Una opción para ocultar información, dentro de archivos que no requieren de herramientas especiales, es mediante el uso de ADS.

## 3.1. Sistema de archivos

Un sistema de archivos se encarga de gestionar los archivos y carpetas, así como la información necesaria para localizar y acceder a estos elementos ya sea por los usuarios locales o remotos; es una parte necesaria del sistema operativo que determina cómo son nombrados, almacenados y organizados los archivos dentro de un volúmen.

Los sistemas operativos de la familia Windows pueden usar dos sistemas de archivos:

- El sistema FAT (File Allocation Table).
- El sistema NTFS.

## 3.2. Sistema NTFS

NTFS es un sistema de archivo desarrollado por Microsoft para su línea de sistemas operativos Windows NT, a partir de Windows NT 3.1 y Windows 2000, incluyendo Windows XP, Windows Server 2003, y todos sus sucesores hasta la fecha.

### 3.2.1. Historia

A mediados de 1980, Microsoft e IBM formaron un proyecto conjunto para crear la próxima generación de sistemas operativos gráficos. El resultado del proyecto fue OS/2, pero hubo muchos desacuerdos entre Microsoft e IBM por lo que decidieron separarse. OS/2 siguió siendo un proyecto de IBM mientras que Microsoft comenzó a trabajar en Windows NT. Cuando Microsoft creó su nuevo sistema operativo, tomo prestado varios conceptos del sistema de archivos HPFS para NTFS.

Los desarrolladores de NTFS son: Tom Miller, Gary Kimura, Brian Andrew y David Goebel.

### 3.2.2. Versiones

El formato en disco NTFS cuenta con cinco versiones de lanzamiento:

- v1.0 con NT 3.1, lanzado a mediados de 1993
- v1.1 con los de NT 3.5, lanzada el otoño de 1994
- v1.2 con NT 3.51 y NT 4

- v3.0 de Windows 2000
- v3.1 de Windows XP

La versión de NTFS.sys no debe ser confundida con la versión del formato NTFS en el disco. El formato en disco NTFS v3.1 no ha cambiado desde la introducción de Windows XP y se utiliza en Windows Server 2003, Windows Server 2008, Windows Vista y Windows 7. La confusión surge cuando se implementan las características en el controlador NTFS.sys dentro del sistema operativo Windows en lugar de el formato en disco NTFS. Un incidente de esto fue cuando Microsoft detalló nuevas características dentro de NTFS en Windows 2000 y lo llamaron NTFS v5.0, sin embargo, es el controlador NTFS.sys que está en esa versión y el formato en disco continuó en la versión 3.0.

### 3.2.3. Arquitectura del Sistema de Archivos NTFS

La figura 3.1 muestra la arquitectura de un sistema de archivos NTFS, los componentes que intervienen son:

- **Disco duro:** Contiene una o varias particiones.
- **Sector de Inicio:** Partición de arranque que almacena información sobre el esquema del volúmen y la estructura del sistema de archivos, así como el código de arranque que carga Ntdlr.
- **Registro Maestro de Arranque (RMB):** Contiene el código ejecutable que el BIOS carga en la memoria. El código escanea el MBR para encontrar la tabla de partición que determina cual partición está activa, o es de arranque.
- **Ntldlr.dll:** Cambia la CPU a modo protegido, se inicia el sistema de archivos, y luego lee el contenido del archivo Boot.ini. Esta información determina las opciones de inicio y las selecciones iniciales del menú de arranque.
- **NTFS.sys:** Es el controlador del sistema de archivos NTFS.
- **Ntoskml.exe:** Extrae información sobre los controladores de dispositivos del sistema para la carga y el orden de carga.

- **Modo Kernel:** Es el modo de procesamiento que permite al código tener acceso directo a todo el hardware y a la memoria en el sistema.
- **Modo usuario:** Es el modo de procesamiento en el cual las aplicaciones son ejecutadas.

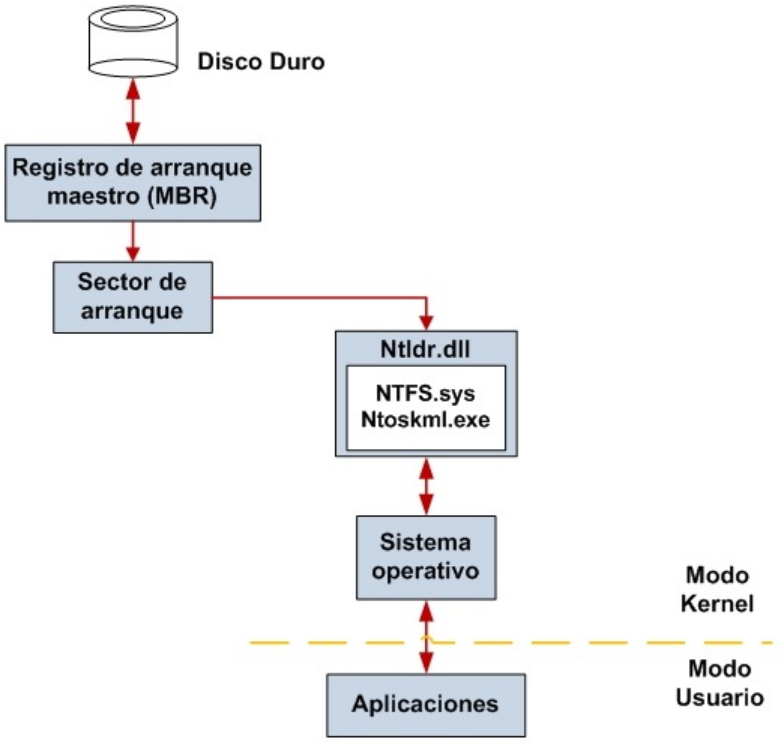


Figura 3.1: Arquitectura del sistema de archivos NTFS

Durante el formato y configuración de un sistema de archivos de un volúmen en un disco duro, se crea un registro maestro de arranque (MBR). El MBR incluye una pequeña cantidad de código ejecutable denominado código de arranque maestro, también tiene una tabla de particiones para el disco. Cuando se monta un volúmen, el MBR ejecuta el código de arranque maestro y cede el control al sector de arranque del disco que se encarga de ejecutar el archivo Ntldlr.dll para iniciar el sistema de archivos, lo que permite que el servidor inicie el sistema operativo con el sistema de archivos de ese volúmen específico.

### 3.2.4. Estructura Física de NTFS

A continuación se describe como están organizados los clusters y los sectores en un volúmen NTFS.

#### 3.2.4.1. Clusters y sectores en una partición NTFS

Un **cluster** (o unidad de asignación) es la menor cantidad de espacio en disco que se puede asignar para contener un archivo.

Todos los sistemas de archivos utilizados por Windows organizan los discos duros basándose en el tamaño del cluster, el cual está determinado por el número de **sectores** (unidades de almacenamiento en un disco duro) que contiene el cluster.

Por ejemplo, en un disco que utiliza sectores de 512 bytes, un cluster de 512 bytes contiene un sector, mientras que un cluster de 4 kilobytes (KB) contiene ocho sectores.

#### 3.2.4.2. Secuencia de Clusters en un volúmen NTFS

Los clusters en un volúmen NTFS se numeran secuencialmente desde el principio de la partición en el número de clusters lógicos.

NTFS almacena todos los objetos en el sistema de archivos usando un registro llamado la tabla maestra de archivos (MFT), similar en estructura a una base de datos.

Los clusters comienzan en el sector cero, por lo tanto, cada cluster se alinea con el límite del cluster anterior. Para el almacenamiento de archivos, los clusters contiguos, permiten un procesamiento más rápido.

### 3.2.5. Organización de un volúmen NTFS

La figura 3.2 muestra como está organizada la estructura de un volúmen NTFS.

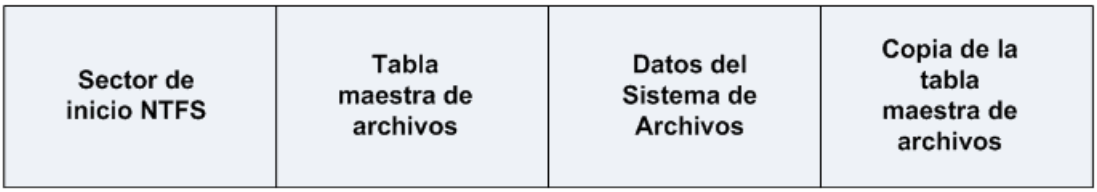


Figura 3.2: Organización de un volúmen NTFS.

Los componentes que forman la estructura organizacional en un volúmen NTFS son:

- **Sector de inicio NTFS:** Contiene el bloque de parámetros del BIOS que almacena información acerca de la disposición del volúmen y las estructuras del sistema de archivos, así como el código de arranque que carga Windows.

- **Tabla maestra de archivos:** Contiene la información necesaria para recuperar los archivos desde la partición NTFS, así como los atributos de un archivo.

- **Datos del sistema de archivos:** Almacena los datos que no están contenidos dentro de la tabla maestra de archivos.

- **Copia de la tabla maestra de archivos:** Incluye copias de los registros esenciales para la recuperación del sistema de archivos si hay un problema con la copia original.

Los primeros 8 kilobytes del sector de inicio NTFS incluye información sobre el volúmen además contiene el código de arranque del sistema operativo, por otro lado, la tabla maestra de archivos almacena la información sobre dónde y cómo están almacenados los archivos y sus atributos. Los datos del sistema de archivos incorpora la información sobre los datos y las operaciones que se realizan sobre el sistema de archivos, la copia de la tabla maestra de archivos contiene un respaldo de los primeros cuatro registros de la MFT.

#### 3.2.5.1. Tabla maestra de archivos

Cuando se da formato a un volúmen con NTFS, Windows crea una MFT y archivos de metadatos en la partición. La MFT es una base de datos relacional que consiste en filas de registros de archivos y columnas de atributos de archivo.

Contiene al menos una entrada para cada archivo en un volúmen NTFS, incluido la propia MFT, la cual almacena la información necesaria para recuperar los archivos de la partición NTFS.

**Archivos de metadatos MFT**

Debido a que la MFT almacena información sobre sí misma, NTFS reserva los primeros 16 registros de la MFT para archivos de metadatos (aproximadamente 16 KB), que se utilizan para describir la MFT.

Los archivos de metadatos comienzan con un signo de dólar ($) y son descritos en la tabla de archivos de metadatos, almacenada en la MFT. El resto de los registros de la MFT contiene los registros de los archivos y directorios para cada archivo y carpeta en el volúmen.

La tabla 3.1 muestra la descripción de los archivos de metadatos almacenados en la MFT.

Tabla 3.1: Archivos de metadatos almacenados en la MFT

| **Archivo del sistema** | **Nombre del archivo** | **Registro MTF** | **Propósito del archivo** |
|---|---|---|---|
| Tabla maestra de archivo | $Mft | 0 | Contiene un registro de archivo base para cada archivo y carpeta en un volúmen NTFS. |
| Espejo de la tabla maestra de archivos | $MftMirr | 1 | Es una imagen duplicada de los primeros cuatro registros de la MFT. |
| Archivo log | $LogFile | 2 | Se usa para restaurar la consistencia de metadatos después de un fallo del sistema. |
| volúmen | $Volume | 3 | Contiene información sobre el volúmen, como lo es el diseño del volúmen y su versión. |
| Definiciones de atributo | $AttrDef | 4 | Listas de los nombres de atributos, números y descripciones. |
| Archivo root nombre index | . | 5 | El directorio raíz del sistema de archivos. |
| Clúster bitmapl | $Bitmap | 6 | Representa el volúmen, mostrando clusters libres y sin usar. |
| Sector de inicio | $Boot | 7 | Sector de inicio y código de arranque para el sistema de achivos. |
| Archivo de cluster dañado | $BadClus | 8 | Contiene los clusters dañados de un volúmen. |
| Archivo de seguridad | $secure | 9 | Contiene los descriptores de seguridad únicos para todos los archivos de un volúmen. |
| Tabla de upcase | $Upcase | 10 | Convierte los caracteres en minúsculas a mayúsculas de cada carácter Unicode. |
| Archivo de extensión NTFS | $Extend | 11 | Directorio que almacena ficheros para extensiones opcionales. |
| | | 12-15 | Reservados para usos futuros. |

Las ubicaciones de los segmentos de datos, tanto para la MFT y la copia de seguridad de la MFT, $Mft y $MftMirr respectivamente, se registran en el sector de arranque. La $MftMirr es una copia duplicada de los cuatro primeros registros de la $Mft o el primer cluster de la $Mft, dependiendo del que sea mayor.

Si algún registro de la MFT dentro del rango que abarca la $MftMirr está dañado o ilegible, entonces NTFS lee el registro $MftMirr y utiliza la información en lugar de la información de $Mft. Si es posible, los datos correctos de la $MftMirr se vuelven a escribir en la ubicación correspondiente en la $Mft.

#### 3.2.5.2. Zona MFT

Para evitar que la MFT se fragmente, NTFS reserva por defecto el 12.5 por ciento del volúmen para el uso exclusivo de la MFT. Este espacio es conocido como la zona MFT, no se utiliza para almacenar datos a menos que el resto del volúmen se encuentre lleno.

#### 3.2.5.3. Archivo de atributos de registro en NTFS

Cada sector asignado en un volúmen NTFS pertenece a un archivo. Incluso los metadatos del sistema de archivos son parte de un archivo. NTFS ve a cada archivo (o carpeta) como un conjunto de atributos de archivo.

Elementos de archivo, como el nombre, la información de seguridad, e incluso sus datos son atributos de archivo. Cada atributo se identifica por un código de tipo de atributo y un nombre de atributo opcional.

La tabla 3.2 muestra los Tipos de atributos de archivos NTFS

Tabla 3.2: Tipos de atributos de archivos en NTFS

| **Tipo de atributo** | **Descripción** |
|---|---|
| Información estándar | La información como modo de acceso (de sólo lectura, lectura / escritura, etc) marca de tiempo y número de enlaces. |
| Lista de atributo | La ubicación de todos los registros de atributos que no encajan en el registro MFT. |
| Nombre del archivo | Es un atributo repetible para nombres de archivo largos y cortos. El nombre largo del archivo puede ser de hasta 255 caracteres Unicode. |
| Datos | Los datos del archivo. NTFS admite varios atributos de datos por archivo. Cada archivo tiene típicamente un atributo de datos sin nombre. Un archivo también puede tener uno o más atributos de datos con nombre. |
| ID Objeto | Un identificador de archivo de volúmen único. Utilizado por el servicio de seguimiento de vínculos distribuidos. No todos los archivos tienen identificadores de objetos. |
| . | Al igual que en un flujo de datos, pero las operaciones se registran en el archivo de registro NTFS como cambios en los metadatos de NTFS. Este atributo es utilizado por EFS. |
| Punto de análisis | Se utiliza para las unidades montadas. Esto también es utilizado por el Sistema de archivos instalable (IFS) de controladores de filtro para marcar ciertos archivos como especial para ese controlador. |
| Indice root | Se utiliza para poner en práctica las carpetas y otros índices. |
| Indice de ubicación | Se utiliza para poner en práctica la estructura de árbol-B para las carpetas de gran tamaño y otros índices de gran tamaño. |
| Bitmap | Se utiliza para poner en práctica la estructura de árbol-B para las carpetas de gran tamaño y otros índices de gran tamaño. |
| Información del volúmen | Sólo se utiliza en el sistema de archivos $Volume. Contiene la versión volúmen. |

## 3.3. Métodos específicos de ocultación de datos en NTFS

Existen diferentes métodos para ocultar información dentro de NTFS, los cuáles deben cumplir los siguientes criterios:

- Cuando el sistema de archivos sea revisado con una utilidad no deberá registrar ningún error.
- Los datos ocultos no serán sobre-escritos, o en su defecto, la posibilidad de que los datos sean sobre-escritos es baja.
- Los usuarios normales no tendrán noticia de los datos ocultos.
- Puede ser guardada una considerable cantidad de información.

### 3.3.1. Categorías de los métodos de ocultación de datos en NTFS

La imagen 3.3 muestra las dos principales categorías de los métodos existentes para la ocultación de datos en el sistema de archivos NTFS.

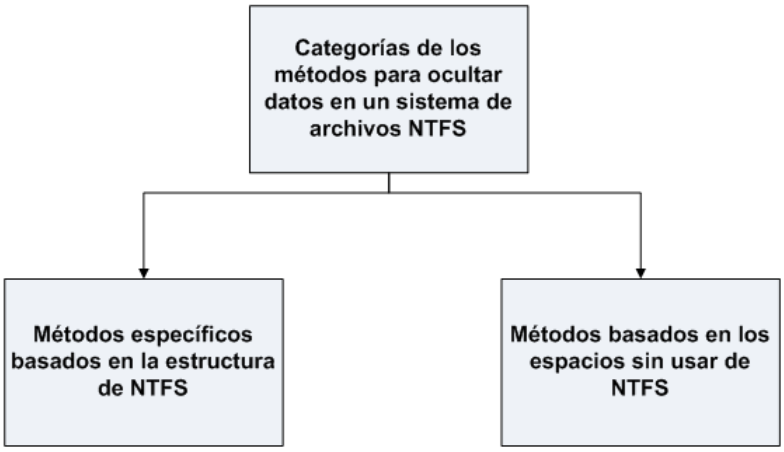


Figura 3.3: Categorías de los métodos de ocultación de datos en NTFS.

#### 3.3.1.1. Métodos basados en los espacios sin usar de NTFS

Los espacios sin usar o "Slack spaces" se refieren a los espacios sobrantes entre el final del archivo y el cluster que se considera usado, por lo que no se encuentra disponible o libre para ser usado.

La existencia de espacios sin usar es una característica de todos los sistemas de archivos, no solamente de NTFS. Pero el análisis de los diferentes métodos para ocultar de datos, no estaría completo si no se abarcan los espacios sin usar. A continuación se mostrarán los aspectos de los métodos que son específicos en NTFS.

**Espacio sin usar en un volúmen**

Se refiere a los espacios sin uso entre el final del volúmen y el final de la partición. El tamaño de los datos ocultos en el espacio desperdiciado del volúmen, esta unicamente limitado por el espacio disponible en el disco duro de una partición. El tamaño de la partición puede ser cambiado en relación al tamaño del volúmen para ocultar más datos.

La figura 3.4 muestra el espacio disponible en el que se pueden ocultar datos dentro de un volúmen.

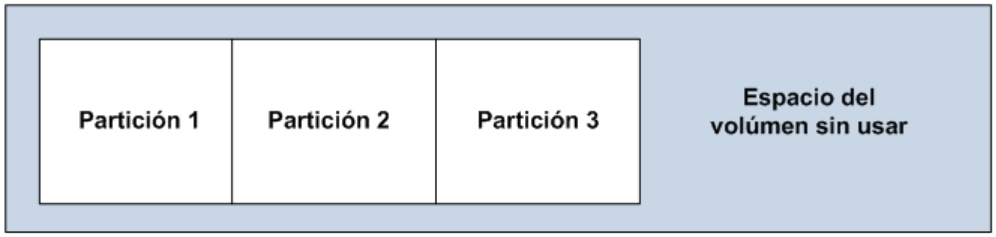


Figura 3.4: Espacio desperdiciado dentro de un volúmen.

**Espacio sin usar en el Sistema de archivos**

Es el espacio no usado al final del sistema de archivos que no es asignado a ningún cluster.

Esto sucede porque el tamaño de la partición no es múltiplo del tamaño del cluster, por ejemplo, si hay 10001 sectores en una partición, los primeros 10000 sectores estarán ubicados en 2500 clusters con un tamaño de cluster de 4 sectores, y el ultimo sector pertenecería al espacio si usar.

El tamaño de los datos que pueden ser ocultos en el espacio libre del sistema de archivos dependerá del tamaño del cluster, por ejemplo, para un cluster estándar de NTFS con un tamaño de 8 sectores, el máximo tamaño del espacio sin usar del sistema de archivos es de 7 sectores.

### Espacio sin usar en el archivo

Es el espacio sin usar entre el final de un archivo y el final del último cluster. El espacio sin usar aparece debido que un cluster es la unidad más pequeña de espacio en disco asignado en NTFS, todo el cluster tiene que ser usado incluso si el archivo no lo llena.

Hay dos tipos de espacio sin usar dentro de un archivo:

- **Espacio sin usar RAM:** Es el espacio del final de un archivo hasta el final del último sector parcialmente usado, en el último cluster asignado.

- **Espacio sin usar controlador:** Es el espacio del inicio del próximo sector hasta el final del último cluster asignado.

Un ejemplo se muestra en la figura 3.5 donde consideramos que un archivo de 600 bytes es almacenado en NTFS con un cluster de 2048 bytes y un sector de 512 bytes, el espacio sin usar dentro de la RAM se extiende desde el final del archivo hasta el final del sector 2 y el espacio sin usar dentro del controlador esta conformado por el sector 3 y 4.

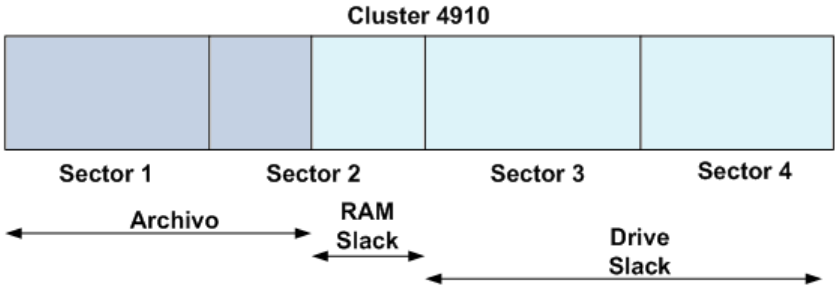


Figura 3.5: Espacio sin usar RAM y controlador.

#### 3.3.1.2. Métodos específicos basados en la estructura de datos de NTFS

Cualquier objeto es un archivo dentro de NTFS, por tal motivo hereda las características de un archivo incluyendo los metadatos del sistema de archivos.

Un atributo es una pequeña estructura que tiene un propósito especifico. Como hemos visto existen once diferentes tipos de archivos de metadatos, por ejemplo, $STANDAR_INFORMATION, $FILE_NAME y $DATA se encuentran presentes en cada registro de archivo independientemente del tipo de archivos que sean, mientras que existen otros atributos que son específicos para un determinado tipo especifico de datos como $DATA, que es usado para los archivos de datos, $INDEX_ROOT que es usado para archivos de directorios.

Algunos atributos como $DATA, $INDEX_ROOT e $INDEX_ALLOCATION pueden aparecer varias veces en un solo archivo y en ese caso se les dá nombres únicos. NTFS no es sensible a un archivo que incluye atributos innecesarios, por tal motivo esta característica puede ser explotada sin afectar el correcto funcionamiento del sistema.

A excepción del atributo $DATA, la mayoría de los otros atributos tienen formatos específicos, por lo que sirven a su propósito destinado. Cualquier manipulación del contenido de estos atributos puede afectar la integridad del sistema, por lo que no son adecuados para ocultar datos.

**Excepciones**

Cuando un atributo es proporcionado para mantener la compatibilidad con versiones anteriores, ya no se utilizan. Por ejemplo $SECURE_DESCRIPTOR o los atributos extendidos $EA y $EA_INFORMATION. En teoría, podría ser posible usar estos atributos para ocultar datos, suponiendo que una herramienta de software está disponible para la adición de estos atributos de archivos. En la práctica, este enfoque no parece eficaz, debido a que no hay certeza de que la compatibilidad con versiones anteriores se mantendrá en las futuras versiones de NTFS.

**Archivos de metadatos basados en métodos**

Es el potencial para la ocultación de algunos datos en archivos de metadatos sin afectar el funcionamiento del sistema. La dificultad es que no se garantiza que NTFS permanecerá insensible a las adiciones en sus versiones futuras. Además de que herramientas de software especializado son requeridas para manipular los archivos de metadatos, y por lo tanto la misma presencia de estas herramientas es suspicaz.

Todos los atributos de estos archivos serán definidos, incluso el atributo $DATA, donde esté presente, y tiene un formato interno definido para soportar las operaciones de NTFS. La única excepción es el atributo de archivo $BADCLUS.

#### 3.3.1.3. Ocultación de datos en el Atributo $BADCLUS

Actualmente los controladores del disco duro manejan los sectores defectuosos por si mismos, sin la intervención del sistema operativo. Esto puede ser mediante:

- **Deslizamiento:** Modificación del número de bloque lógico (LBN) del mapeo físico para omitir el sector defectuoso.
- **Re mapeo:** Re asignación del LBN del área defectuosa al sector de repuesto.

También el gestor del volúmen junto con Windows, son capaces de reasignar los sectores dañados.

En los discos duros que no tienen estas capacidades, el sistema de archivos y el sistema operativo tienen que conservar la habilidad de detectar y marcar los sectores defectuosos, así como los clusters dañados. Esta habilidad puede ser usada para excluir los cluster que se encuentran sin daños de las actividades normales del sistema de archivos, y usarlos para ocultar datos en cualquier sistema.

En NTFS los clusters dañados son marcados en el archivo de metadatos $BADCLUS que tiene la entrada 8 a la MFT. $BADCLUS es un archivo con un tamaño fijo, por lo que los clusters dañados son detectados y asignados a este archivo.

En este caso, el tamaño de la ocultación de datos es igual a la capacidad del total del sistema de archivos. Cualquier número de clusters puede ser asignado a $BADCLUS y usado para almacenar datos.

#### 3.3.1.4. Ocultación de datos en el Atributo $BOOT

En NTFS el registro de arranque es almacenado en un archivo de metadatos llamado $BOOT, es el único archivo con una ubicación fija; siempre comienza en el primer cluster del sistema de archivos y Windows asigna 16 sectores a este archivo, pero generalmente la mitad de estos sectores contienen bits nulos.

Bytes no usados en el sector de arranque. Windows no montará el sistema de archivos si hay valores que no sean nulos en los bytes no usados, como resultado este espacio no puede ser usado para ocultar datos.

Los bytes asignados al código de arranque en el archivo $BOOT del sistema de archivos, pueden ser usados para ocultar datos. El código de arranque es esencial para arrancar el sistema de archivos, para localizar los archivos necesarios para iniciar Windows.

Para un sistema de archivos que no sea de arranque, se utiliza para almacenar el mensaje de error que es mostrado si se ha hecho un esfuerzo para arrancar desde esta partición.

El tamaño de los datos que pueden ser ocultos de esta manera está limitado por el número de bytes nulos en el archivo $BOOT.

#### 3.3.1.5. Atributo $DATA

Los métodos más efectivos de ocultación de datos son los basados en el atributo de archivo $DATA. Este atributo es el único sin un formato implícito así que su contenido puede ser arbitrario.

Este atributo puede tener cualquier tamaño sin levantar sospechas, y esto es razonable para asumir que siempre seguirá siendo una característica de NTFS. Además es posible añadir varios datos ocultos en el atributo $DATA sin perturbar los datos ya presentes. El atributo $DATA es de lectura y escritura por lo que los datos añadidos pueden ser sobreescritos.

La mayoría de los archivos de metadatos contienen al atributo $DATA, excepto los directorios y las extensiones de archivos de metadatos; así como el metadato $BADCLUS pues su contenido es usado en operaciones de NTFS y el formato dependerá del rol que dicho archivo esté cumpliendo dentro del sistema de archivos.

Algunos de los archivos de metadatos son estáticos, por ejemplo los atributos $Bitmap, $Boot, $AtrDef y $Upcase. Algunos clusters pueden ser asignados por el atributo $DATA para ocultar datos de esos archivos de metadatos.

**Métodos basados en archivos de datos**

Cada archivo y cada directorio tiene por lo menos una entrada a la MFT. En una típica interface GUI para el sistema de archivos, se asume que cada archivo de datos tienen un único atributo $DATA sin nombre y que cada directorio tiene un único atributo $INDEX_ROOT.

El atributo $DATA es aquel que proporciona un mayor alcance para ocultar datos, en términos de capacidad y longitud de los datos ocultos.

## 3.4. Definición de ADS

En 1994, los "Alternate Data Streams" o "Flujos Alternativos de Datos" (ADS) entraron en vigor junto con el sistema de archivos NTFS. De acuerdo a Microsoft, los ADS son una característica diseñada para la compatibilidad con el sistema operativo Macintosh, que usa una forma de flujos llamados forks, en su sistema de archivos, también indico que los ADS son de vital importancia en su línea de productos.

Los ADS son una característica del sistema de archivos NTFS que permite almacenar metadatos dentro de un archivo o directorio.

Dicho de otra manera se puede agregar un ADS a un archivo o directorio dentro de particiones que cuenten con el Sistema de archivos NTFS, como vemos en la figura 3.6, un archivo puede actuar de portador para almacenar varios ADS sin que éstos sean visibles. En este trabajo únicamente nos enfocaremos a los ADS que se pueden agregar a un archivo.

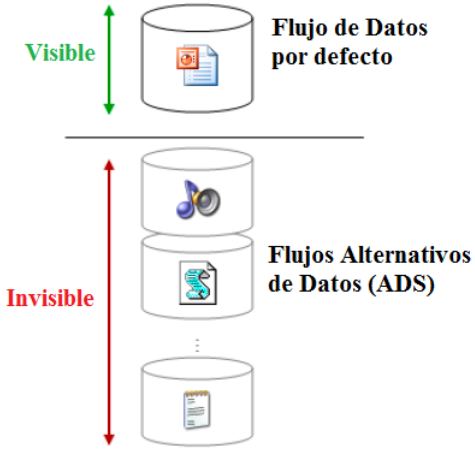


Figura 3.6: Flujo Alternativo de Datos.

## 3.5. Estructura de un archivo con múltiples ADS

La figura 3.7 muestra la estructura de un archivo que cuenta con múltiples flujos alternativos de datos, en el cual solamente el flujo principal será visible para cualquier sistema de archivos, mientras que los ADS solo podrán ser visualizados por el sistema de archivos NTFS.

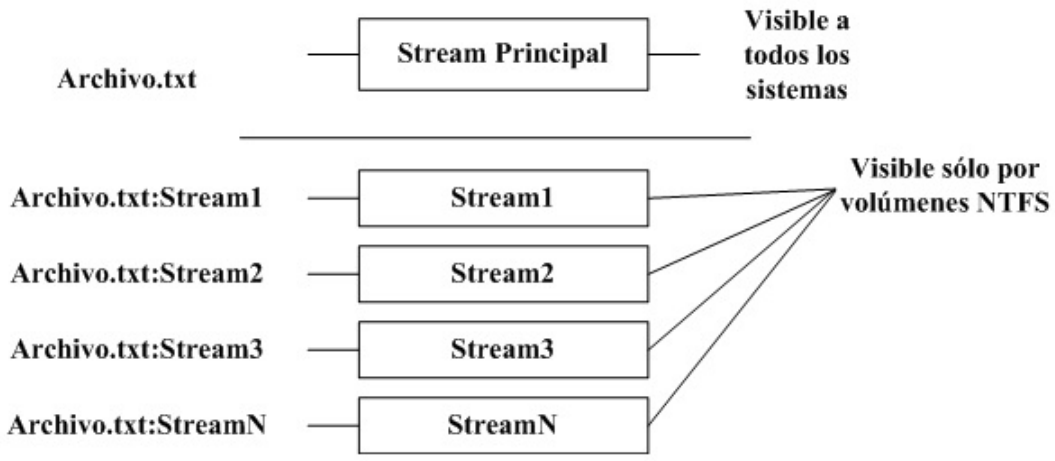


Figura 3.7: Estructura de un archivo con múltiples ADS.

## 3.6. Recorrido de registros dentro de la MFT

Se ha mencionado que la MFT almacena una lista de registros que contienen atributos. Los atributos consisten en agrupaciones distintas de datos en el registro que se ajustan a una cierta estructura.

Cada tipo de atributo almacena una diferente parte de información sobre el archivo o directorio de acuerdo a su alcance, van desde un bit para indicar si el documento es de solo lectura, hasta el contenido completo de un archivo de video. El tamaño de cada uno de los registros en la MFT es igual al tamaño del volúmen del cluster.

NTFS cuenta con más de una docena de atributos de usuario definidos, sin embargo para entender los ADS solo se requiere del conocimiento de tres de ellos:

- Atributo de Información Estándar (SI)
- Atributo de Nombre de Archivo (FN)
- Atributo de Datos (DATA)

Cada atributo en la MFT se puede dividir en dos partes: su cabecera y el contenido.

- **Cabecera:** Puede haber cuatro diferentes tipos de cabeceras para un atributo, dependiendo de si el atributo tiene o no un nombre, y si la parte contenida es almacenada inmediatamente después de la cabecera en la MFT (residente) o en una ubicación alternativa en el volúmen (no residente).

  Los cuatro tipos de cabeceras almacenan información incluyendo el tipo de atributo (SI, FN, DATA, entre otros), la longitud del atributo, la ubicación del contenido, y en el caso de los atributos de datos, en estado de compresión del contenido. Sin embargo en el caso del contenido no residente, la cabecera almacena información identificando la parte de datos referenciada en el contenido, así como su tamaño.

- **Contenido:** Depende del tamaño de los datos. Cuando el tamaño es apropiado, el contenido existe inmediatamente después de la cabecera en el propio registro de la MTF. Si la cantidad de datos a ser almacenados excede el tamaño restante del registro, el contenido tendrá una lista de ejecución con un apuntador desde la ubicación actual de los datos hasta la otra parte en el disco donde reside el resto de la información.

  Cada parte de datos en el volúmen es almacenado en un cluster. La lista de ejecución almacena una lista de elementos que contiene tanto el número de cluster lógico (LCN), o el cluster para un conjunto de datos, y el número de clusters en un tiempo determinado.

La figura 3.8 muestra un recorrido en la MFT con el recorrido en la cabecera en color rojo, y cada uno de los tres atributos (Información estándar, el nombre de archivo y datos) con cabecera en color verde y contenido en color azul.

Este registro en partícular contiene un flujo no residente compuesto de clusters almacenados en distintos lugares del volúmen. Cada elemento de la lista de ejecución en color azul, en los atributos de datos identifica a un conjunto de clusters que contienen flujos de datos, mostrados en color amarillo. Si el flujo fué suficientemente pequeño como para ser residente, la lista de ejecución podría ser sustituida por los datos e inmediatamente pasará a la siguiente cabecera en color verde.

Por otro lado, si la lista de ejecución es demasiado grande para ser almacenada en el registro, NTFS tiene un mecanismo para almacenar uno o más atributos externos en un registro separado de la MFT.

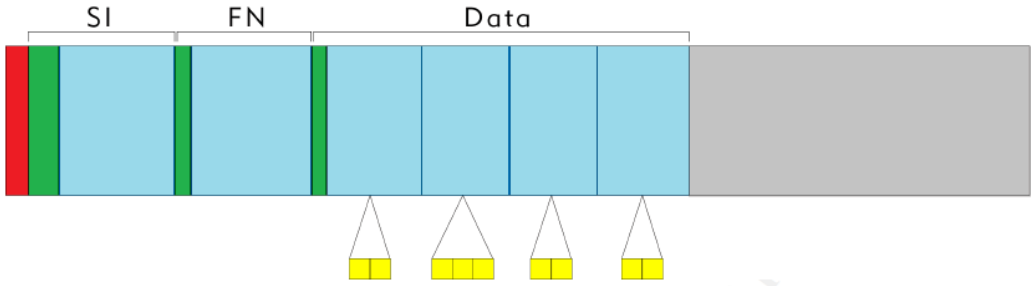


Figura 3.8: Recorrido de un archivo en la MFT.

## Estructura de un recorrido con ADS

En un archivo con un ADS, el recorrido parece exactamente el mismo que el descrito en la figura 3.8, excepto que en lugar de tener solamente un atributo de datos tiene dos o más.

NTFS permite unicamente un atributo de datos sin nombre por cada registro, así que cualquier atributo de datos adicional debe ser nombrado. Estos atributos de datos adicionales que tienen un nombre contienen el ADS.

La estructura de un registro de archivo con ADS se ilustra en la figura 3.9

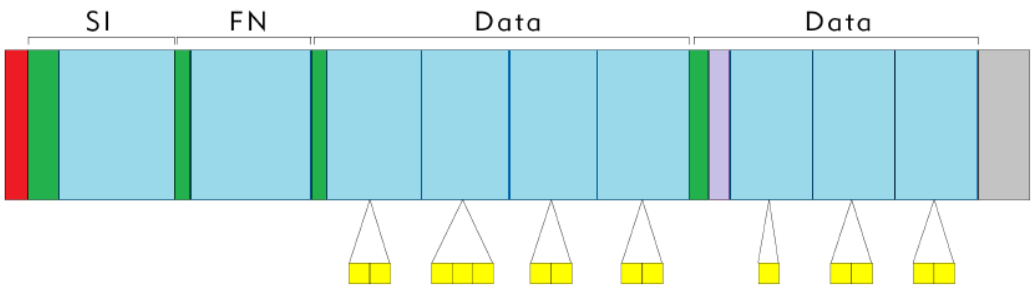


Figura 3.9: Recorrido de un archivo con ADS.

El recorrido parece muy similar al recorrido en la figura 3.8, con la adición de un segundo atributo de datos. Este segundo atributo de datos se encuentran ubicados en el área de color morado, que se encuentra al final del atributo cabecera.

## 3.7. Sintaxis de un ADS

La sintaxis para referenciar los ADS en archivos de NTFS es la siguiente:

***archivo[:flujo[:tipo]]***

Donde, lo que está entre "[" y "]" es opcional.

Si únicamente se específica el archivo se obtendrá el flujo principal de datos, que es la información que se conoce y es visible para cualquier usuario; debido a que el flujo principal de datos no tiene un nombre "de flujo", en NTFS, los siguientes son equivalentes:

***archivo***

***archivo::$DATA***

Puede presentarse el caso en que exista cierta confusión entre la ubicación de un archivo y el flujo de datos dentro de un archivo.

Por ejemplo, D:texto.txt, donde no queda claro si se esta refiriendo a un archivo en la unidad de disco D: o a un flujo de datos dentro del archivo "D" en el directorio actual. El sistema siempre se decide por la primera opción, por tal motivo se recomienda especificar la ubicación completa del archivo para no generar posibles confusiones.

## 3.8. Herramientas para trabajar con ADS

Existen varias herramientas que nos permiten manipular y detectar los ADS como las que se enlistan a continuación:

- LADS
- LAGADS
- ADS Spy
- NTFS Streams Info
- ADS Locator
- List NTFS Streams (LNS)
- StreamArmor
- Streams
- ADS Manager

## 3.9. Eliminación de ADS

Una manera práctica de eliminar ADS es auxiliándose de una unidad lógica de disco que tenga un sistema de archivos FAT, de manera tal que al copiar o mover el archivo que contiene ADS de la unidad lógica con NTFS a la unidad lógico con FAT, el sistema operativo alterará de inmediato que existe información adjunta al archivo seleccionado.

# Capítulo 4

# Marco teórico, desarrollo y resultados

## 4.1. Marco teórico

A continuación se presentan las diversas maneras para ocultar diferentes tipos de archivos mediante el uso de ADS y posteriormente manipularlos para la correcta ejecución de los archivos ocultos.

### 4.1.1. Adición de texto plano en un ADS

La sintaxis de los ADS no es reconocida por las interfaces de usuario de la mayoría de los programas, pero podemos apoyarnos en el uso de los operadores de redirección "<" y ">".

Para añadir información de texto plano a un ADS desde la línea de comandos de Windows se puede utilizar el comando **ECHO** y el operador de redirección ">" de la siguiente manera:

***Echo Mensaje*** > archivo***.txt:ads.txt***

Para comenzar a trabajar se creó una carpeta de nombre "pruebas" y la ubicación de dicha carpeta fue "C:\pruebas"; posteriormente se creó el archivo de nombre "ficheroPrueba.txt" que contendrá el ADS.

Después se agrego el mensaje secreto dentro del ADS, mediante el uso del comando **ECHO** y el operador "$>$" como se muestra en la figura 4.1.

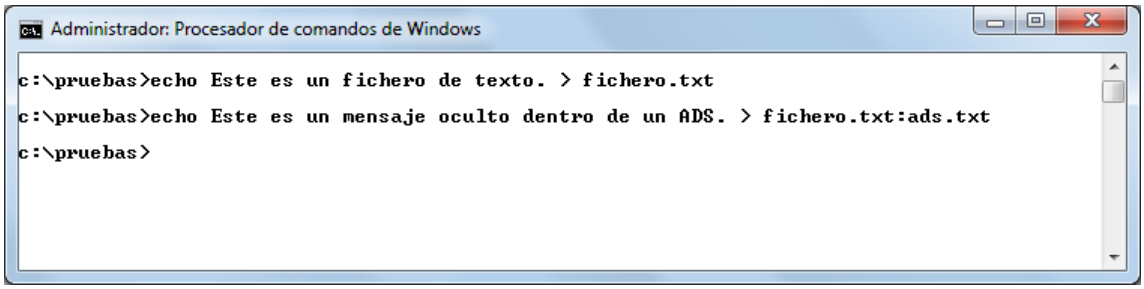


Figura 4.1: Adición de texto plano dentro de un ADS.

### 4.1.2. Lectura de un ADS que contiene texto plano desde la línea de comandos de Windows

Para obtener un mensaje desde la línea de comandos de Windows se usa el comando **TYPE** seguido del nombre del archivo, como se muestra en la figura 4.2.

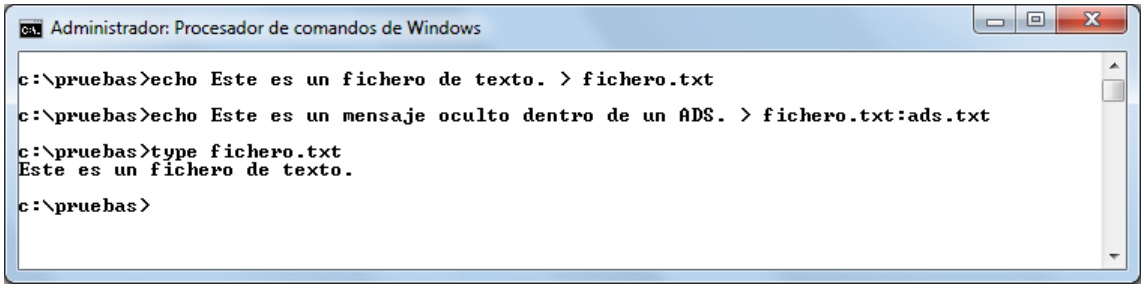


Figura 4.2: Lectura de un archivo de texto mediante el comando TYPE.

Como se observa en la figura 4.2, si se indica únicamente el nombre del archivo, se mostrará el mensaje original y no se podrá visualizar el mensaje oculto en el ADS, para ello se debe hacer mención al ADS contendido en el archivo, si referenciamos el ADS de la misma manera que el archivo se obtendrá un error que indica que el nombre, directorio o sintaxis no son correctos, esto se debe a que no se puede hacer uso del operador **TYPE** para ver un ADS, tal como se muestra en la figura 4.3.

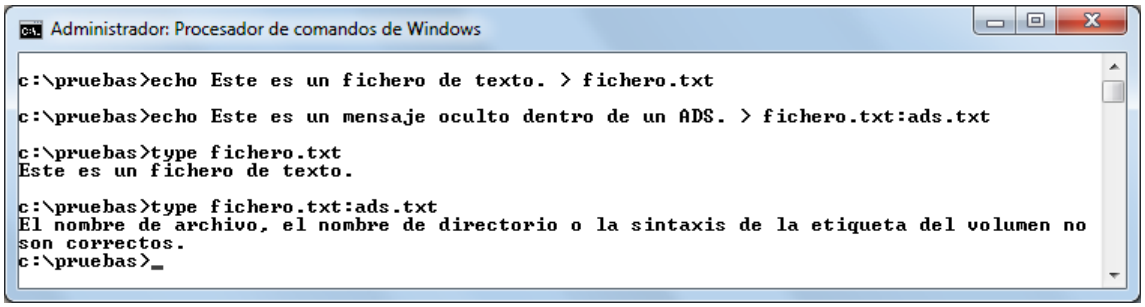


Figura 4.3: Error en la lectura de un archivo que contiene un ADS con texto plano mediante el comando TYPE.

Para evitar este error se debe hacer uso del comando **MORE** y el operador de redirección "<" de la siguiente manera:

$$\textbf{\textit{More}} < \text{archivo}.\textbf{\textit{txt:ads.txt}}$$

En la figura 4.4 se puede observar que mediante el uso de los operadores de redirección se puede crear un ADS y posteriormente visualizarlo.

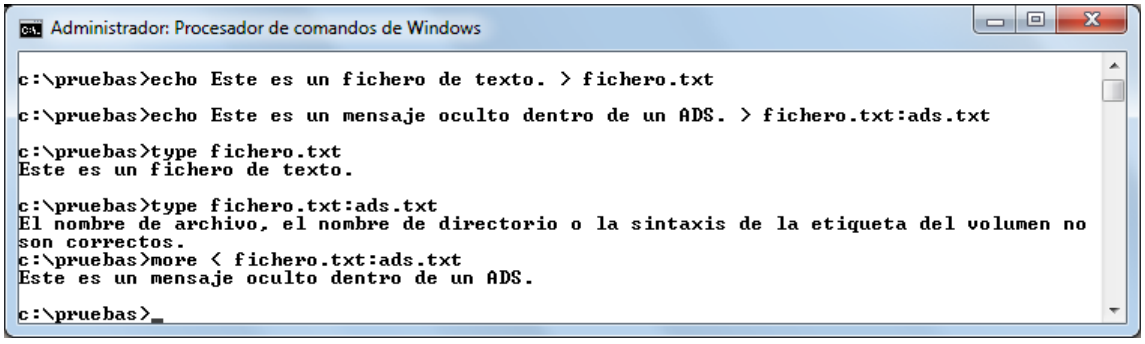


Figura 4.4: Lectura de un ADS mediante el comando MORE.

### 4.1.3. Lectura de un ADS que contiene texto plano mediante el uso del editor de texto NOTEPAD

Otra manera de leer un texto plano contenido en un ADS es mediante el editor de texto Notepad, para ello simplemente se invoca al editor y se indica la referencia del archivo que contiene el ADS de la siguiente manera:

***Notepad*** archivo.***txt:ads.txt***

La figura 4.5 muestra la lectura de ADS que contiene texto mediante el editor de texto Notepad.

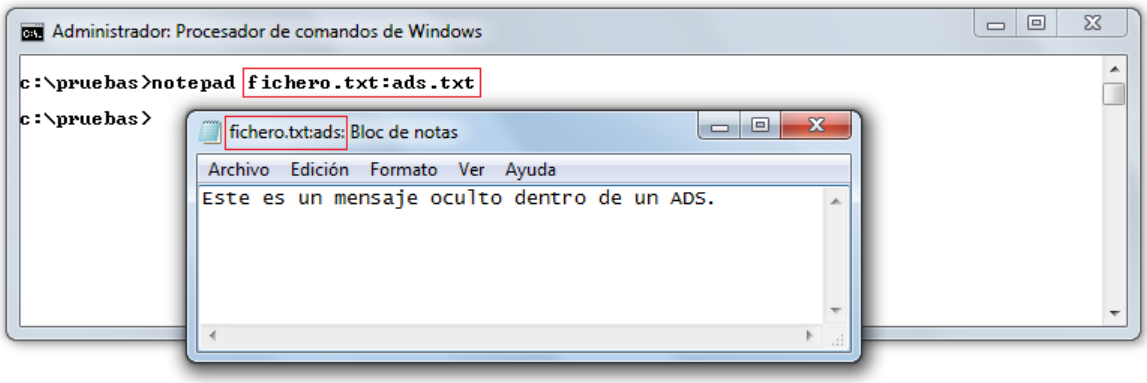


Figura 4.5: Lectura de un ADS que contiene texto plano mediante el uso del editor de texto Notepad.

### 4.1.4. Creación y lectura de un ADS que contiene texto mediante un Script

Se puede hacer uso de scripts para agregar un texto plano a un ADS, en este caso se uso un script en PHP, el cual se muestra a continuación:

```
<?php
// Se abre el archivo ficheroPrueba.txt con el ADS stream.txt
$canal = fopen('ficheroPrueba:stream.txt','w');
// Se escribe el mensaje oculto en el canal
fwrite($canal, Éste es un mensaje oculto....');
// Se cierra el canal
fclose ($canal);
?>
```

Primero se debe crear un archivo que contendrá el mensaje oculto, en este caso tiene el nombre de "fichero2.txt", a continuación se crea un ADS vacío el cual lleva por nombre "ads2.txt", posteriormente se ejecuta el script que contiene el mensaje secreto. Para finalizar se ejecuta el comando **MORE** y de esta manera se podrá visualizar el contenido del ADS, tal como se muestra en la figura 4.6.

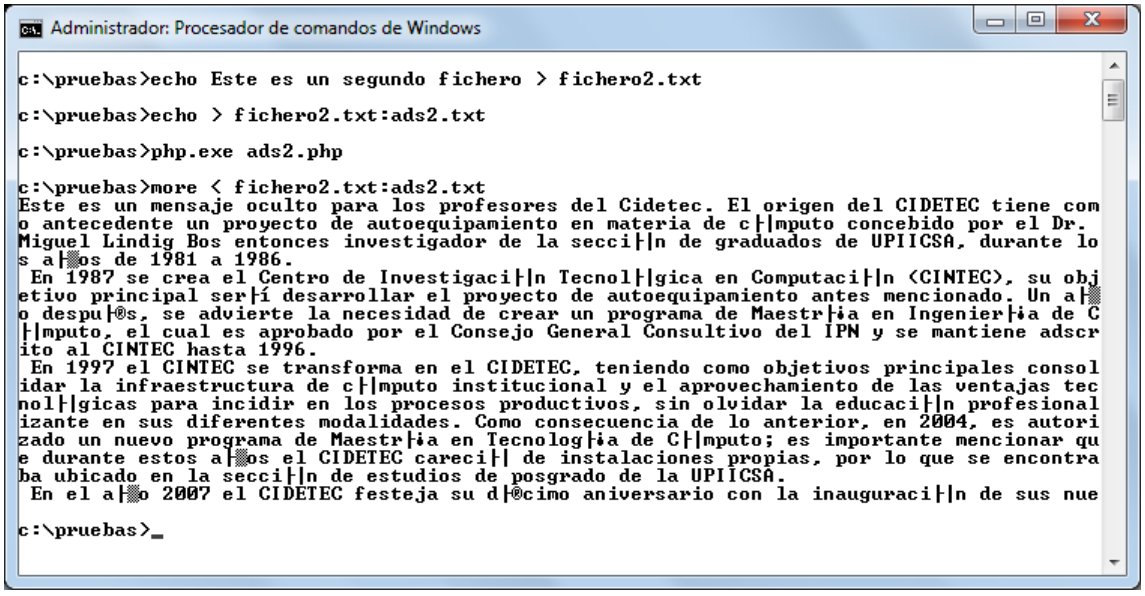


Figura 4.6: Creación y lectura de un ADS que contiene texto plano mediante un script en PHP.

### 4.1.5. Lectura de un texto plano mediante el uso de enlaces simbólicos

Otra manera de ejecutar un ADS es mediante el uso de enlaces simbólicos.

Un enlace simbólico es un tipo de archivo que hace referencia a otro recurso, generalmente un archivo o directorio y puede tener una dirección absoluta o una relativa. Dicho de otra manera, al crear un enlace simbólico el sistema operativo lo interpreta como una dirección que hace referencia a un archivo o directorio, cumpliendo una función similar a la de los accesos directos de Windows.

Los enlaces simbólicos no ocupan espacio en el disco duro ya que simplemente es un apuntador que lleva a otra ubicación dentro del disco duro.

A partir del sistema operativo Windows Vista se puede utilizar un comando que crea enlaces simbólicos: **MKLINKS**. Éstos enlaces, como iremos viendo, nos ayudaran a ejecutar archivos que tienen ADS.

A continuación se muestra un ejemplo de como hacer uso de los enlaces simbólicos para abrir un ADS con Word mediante el uso del comando **MKLINK**.

## Abrir un ADS que contiene texto plano con Word mediante el uso de un enlace simbólico

Primero se debe crear el archivo donde se contendrá el ADS, en el ejemplo lleva el nombre de "mainFile.txt", después se crea el ADS que tiene el nombre de "flujoDatos.txt", posteriormente para abrir el ADS con Word se hace uso del enlace simbólico de la siguiente manera:

***Mklink nombreEnlaceSimbólico rutaDelArchivoADS***

En este caso la ruta corresponde a la del archivo que contiene el ADS, en la figura 4.7 se puede observar la creación del enlace simbólico.

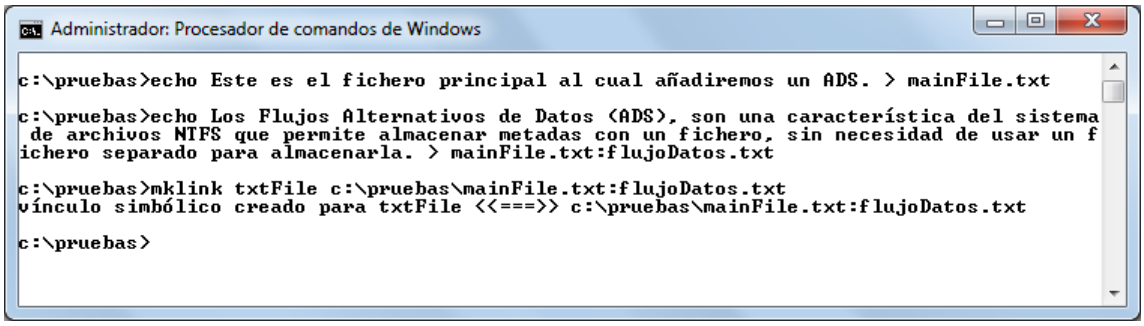


Figura 4.7: Creación de un enlace simbólico.

Una vez que se creó el enlace simbólico aparecerá en la carpeta que indicamos, tal como se muestra en la figura 4.8.

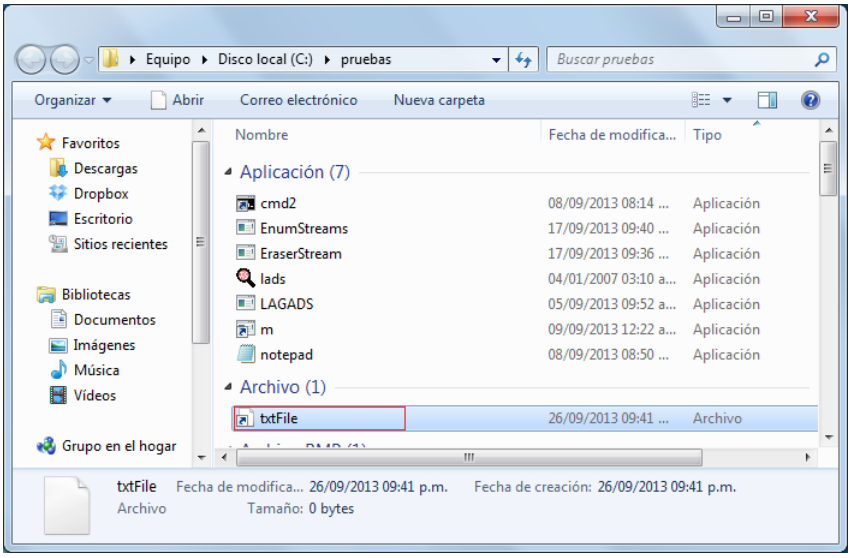


Figura 4.8: Ubicación del enlace simbólico.

Si ejecutamos el enlace simbólico nos aparecerá una ventana donde nos preguntará con que programa deseamos abrir el archivo como indica la figura 4.9

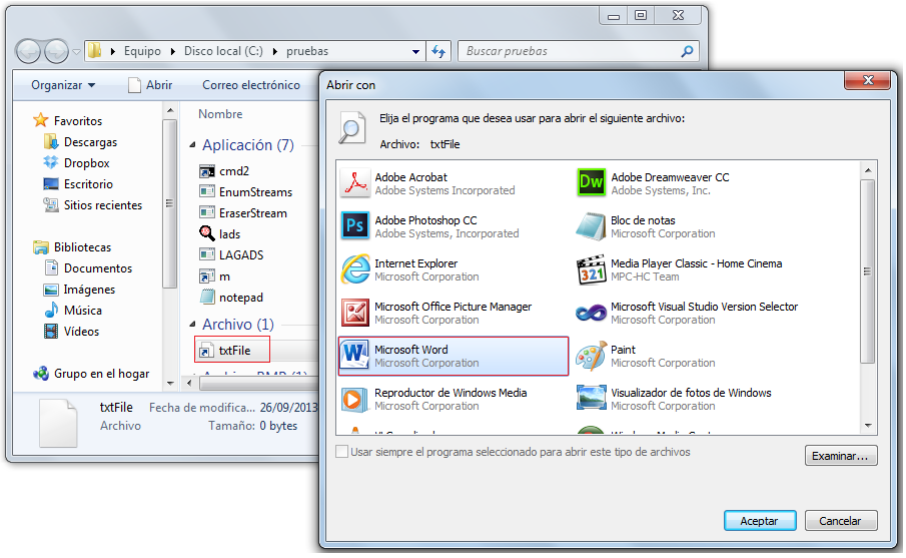


Figura 4.9: Selección del programa para ejecutar el enlace simbólico.

Se da doble clic al enlace simbólico y se selecciona el programa Word dando clic en aceptar, a continuación se puede visualizar la información que se añadió al archivo como se observa en la figura 4.10.

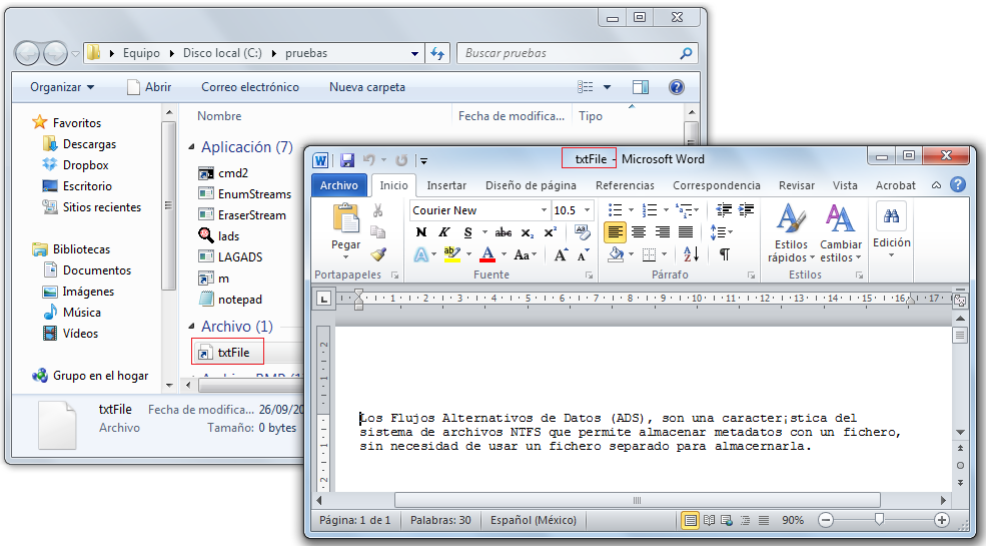


Figura 4.10: Ejecución de un enlace simbólico para la visualización de un ADS que contiene texto plano.

### 4.1.6. Adición de una imagen en un ADS

Se puede añadir imágenes de cualquier formato dentro de un ADS, para ello se hace uso del comando **TYPE** y el operador de selección "$>$" de la siguiente manera:

$$\textbf{\textit{Type Imagen.png}} > \text{archivo}.\textbf{\textit{txt:ads.png}}$$

En la figura 4.11 se observa que la carpeta "pruebas" contiene una imagen de nombre "penguins.jpg", la cual se ocultará en un archivo de texto.

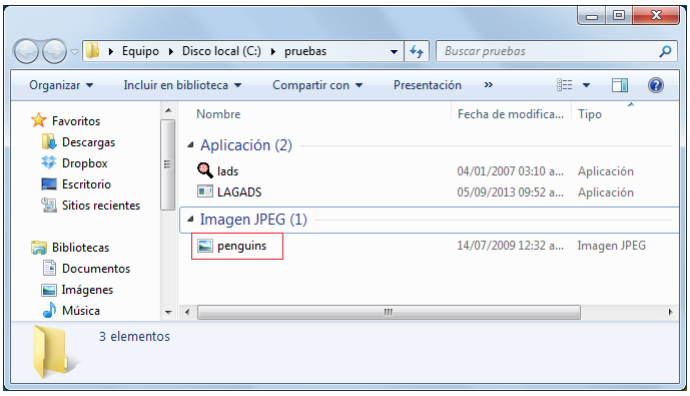


Figura 4.11: Ubicación de la imagen a ocultar.

Después se crea el ADS que contendrá la imagen dentro de un archivo llamado "prueba.txt" como se observa en la figura 4.12.

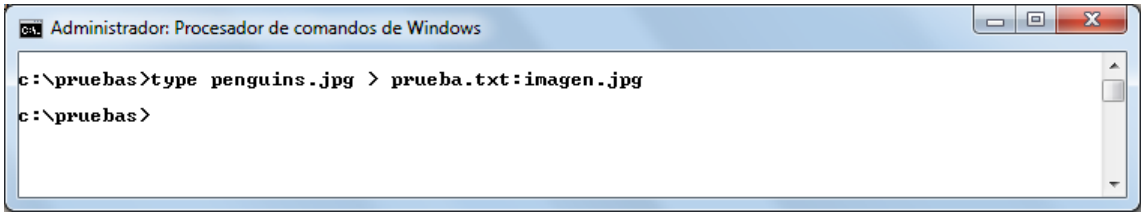


Figura 4.12: Creación de un ADS que contiene una imagen.

### 4.1.7. Visualización de una imagen contenida en un ADS

Para poder visualizar una imagen oculta no se puede utilizar el comando **MORE**, ya que lo único que mostrará la línea de comandos de Windows será los bytes impresos en código ASCII, lo cual unicamente servirá para ver la cabecera de la imagen que está oculta. En este caso, como se puede observar en la figura 4.13, no se puede visualizar la cabecera pero se percibe que dice JFIF lo cual es característico de las imágenes con formato JPG.

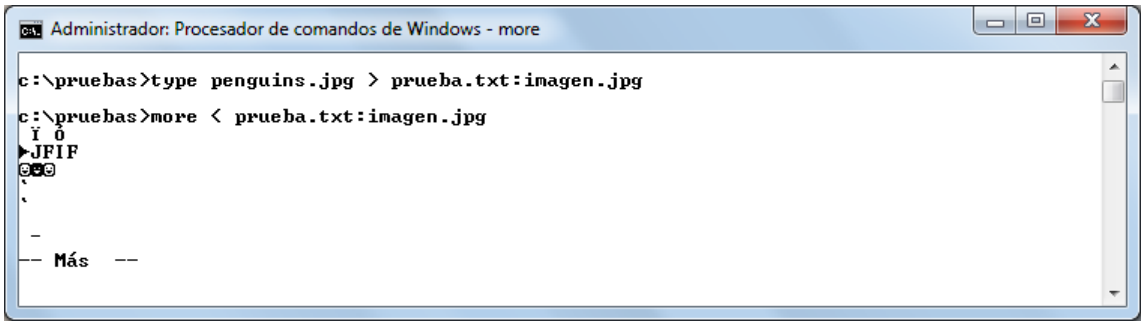


Figura 4.13: Visualización de una imagen con el comando TYPE.

El comando **MORE** daña los bytes, ya que el uso de dicho comando no asegura la integridad de los bytes, como se puede observar en la imagen 4.14.

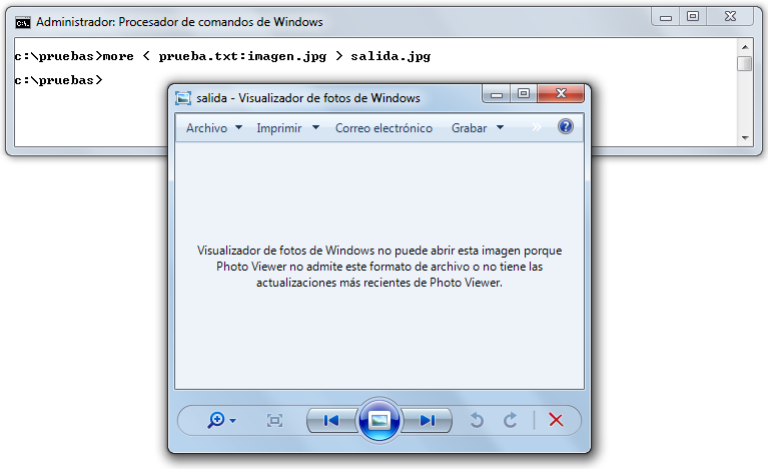


Figura 4.14: Error al visualizar una imagen con el comando MORE.

Una manera sencilla para visualizar una imagen añadida a un ADS, es invocando el visualizador de imágenes paint de la siguiente manera:

***Mspaint archivo.txt:imagen.jpg***

En la figura 4.15 se muestra la visualización de un ADS que contiene una imagen mediante el visualizador de imágenes Paint.

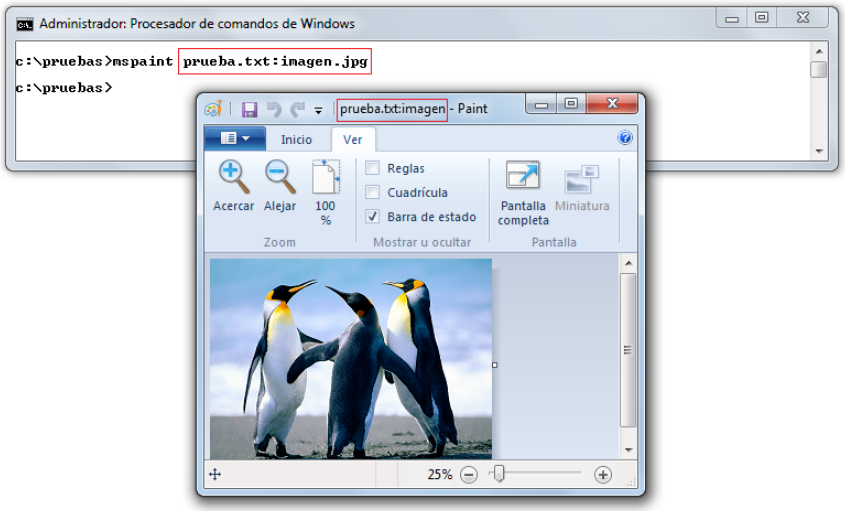


Figura 4.15: Visualización de una imagen oculta en un ADS mediante el programa Paint.

Como se ha mostrado se añadió una imagen a un archivo de texto, pero también se puede realizar para otro tipo de archivos.

A continuación se muestra como agregar una imagen dentro de otra. En la figura 4.16 se pueden observar las dos imágenes con las que se trabajaran que llevan por nombre “tulipán.png” y “faro.png” respectivamente.

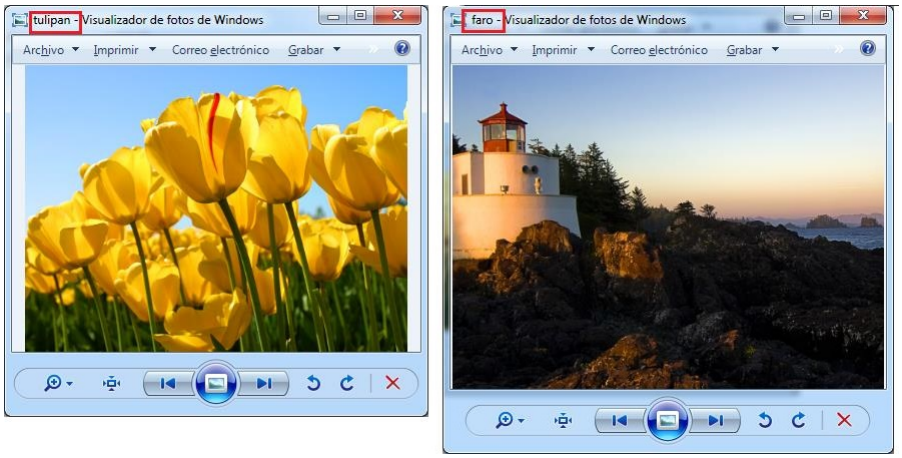


Figura 4.16: Ocultación de una imagen dentro de otra imagen.

Después se crea un ADS que contiene la imagen llamada "tulipán.png" dentro de la imagen "faro.png" y al momento de la visualización queda como se muestra en la figura 4.17.

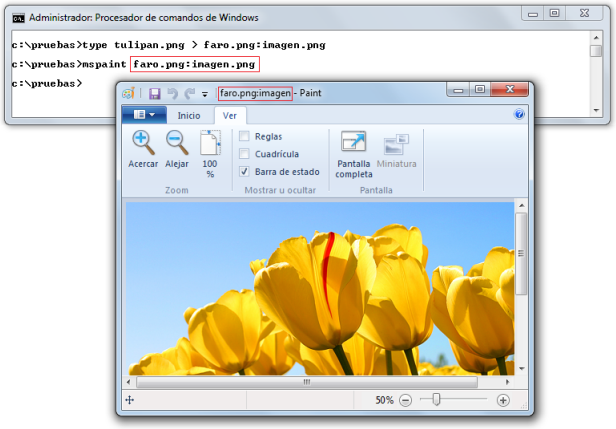


Figura 4.17: Visualización de una imagen oculta en un ADS dentro de otra imagen.

Por ultimo se agrega una imagen a un archivo ejecutable, en este caso, se realizó una copia de la aplicación Notepad a la carpeta en la cual se están realizando las pruebas, también se tiene una imagen que lleva el nombre de "koala.bmp", posteriormente con el comando TYPE se procede a crear el ADS que contendrá dicha imagen, como se puede observar en la figura 4.18.

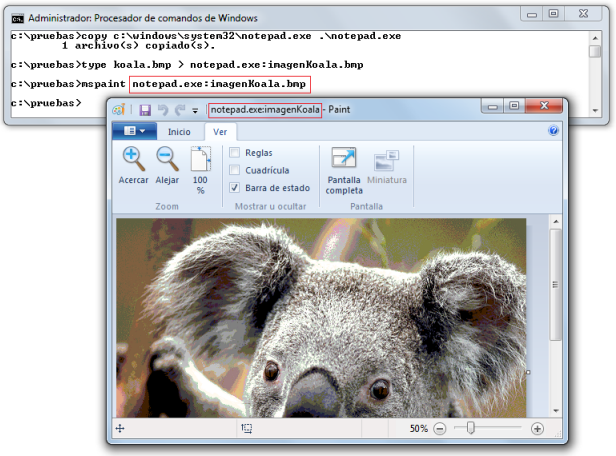


Figura 4.18: Ocultando una imagen en un archivo .exe mediante un ADS.

### 4.1.8. Visualización de una imagen oculta en un ADS mediante el uso de un script

Otra manera de visualizar una imagen oculta es mediante el uso de un script.

En este caso se ha utilizado el siguiente script en PHP:

```
<?php
$file = file_get_contents('prueba.txt:imagenOsos.jpg');
echo $file;
?>
```

Se crea el ADS de la manera en la que se ha visto, posteriormente se ejecuta el script en PHP y por último se puede visualizar la imagen, como indica la figura 4.19.

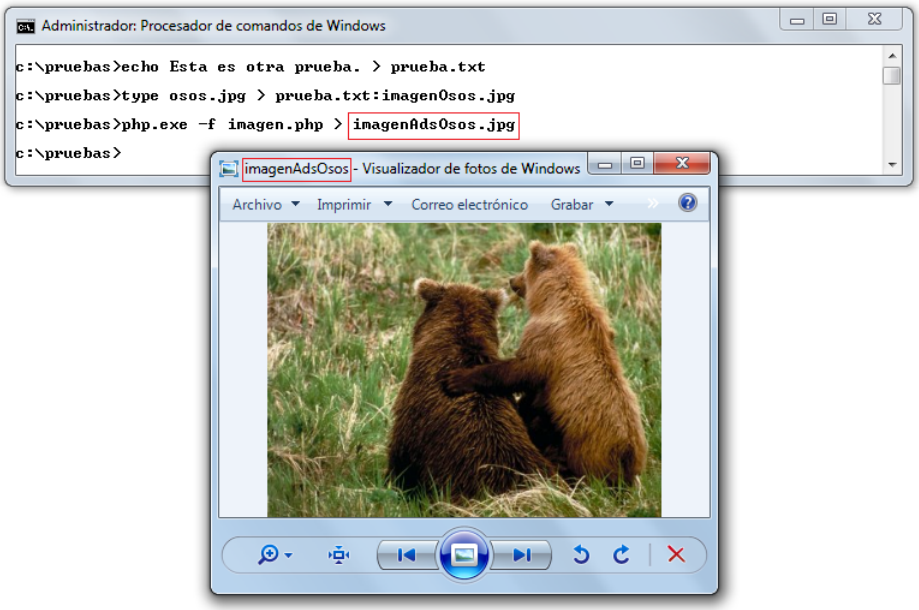


Figura 4.19: Visualización de una imagen mediante un script.

### 4.1.9. Adición de un archivo .exe dentro de un ADS

Como se ha visto anteriormente, con el uso del comando **TYPE** se puede crear un ADS contiene un archivo de cualquier tipo, también se ha observado que al momento de crear un ADS no se ha presentado ningún problema.

Sin embargo, comienzan a surgir dificultades al momento en que se desea ejecutar ADS como incluye diferentes tipos de archivos como un ejecutable o un video.

A continuación se muestra como crear un ADS que contiene un archivo ejecutable (.exe), en este caso se efectuó una copia de la aplicación calc.exe que se incluye por defecto en los sistemas operativos de Windows, como se muestra en la figura 4.20.

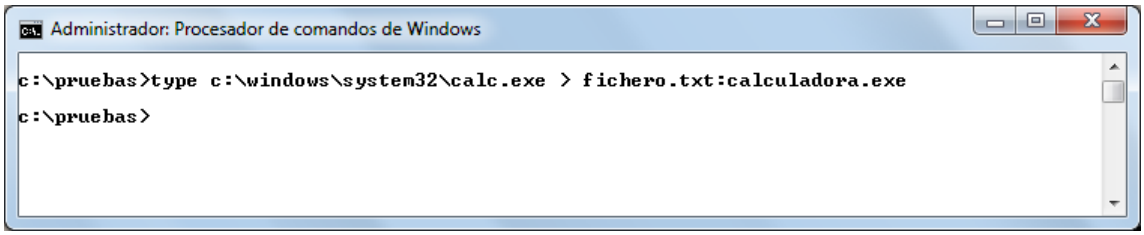


Figura 4.20: Agregando un archivo .exe a un ADS.

## Errores al intentar abrir un ADS que contiene un archivo ejecutable

- Comando **TYPE**

No se puede hacer uso del comando **TYPE** como en el caso en el cual se oculta un archivo de texto plano dentro de un ADS, pues nos aparecerá un mensaje diciendo que no se puede encontrar el archivo.

Este error se produce debido a la política de Microsoft con respecto a los ADS, en la cual se indica que los ADS no necesitan o no deberían ser accedidos por el usuario final, sino unicamente por aquellas aplicaciones que los utilicen, es por esto que no existen interfaces de usuario que reconozcan la sintaxis de los ADS y permitan trabajar directamente con ellos. Por ejemplo: el comando **TYPE** no sirve para ver por pantalla un ADS, ni el comando **DEL** permite borrar ADS, entre otros. De manera que, puede resultar difícil acceder a cierto tipo de información contenida dentro de un ADS.

- Comando **START**

En versiones anteriores a Windows Vista, se podía hacer uso del comando **START** para ejecutar un ADS de la siguiente manera:

$$start\ .\backslash Fichero{:}ADS$$

Sin embargo, al utilizar dicho comando en equipos con el sistema operativo Windows 7 aparecerá una ventana con el mensaje que se muestra en la figura 4.21.

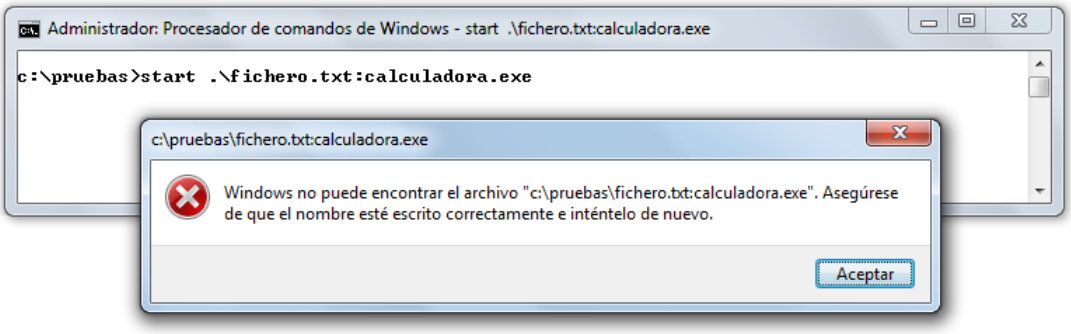


Figura 4.21: Error al ejecutar un ADS con el comando START.

Como se puede observar en la figura 4.22 al momento de ejecutar el comando **START** en equipos con sistema operativo Windows 7, se manda a ejecutar el ADS, pero el explorador de Windows no encuentra una aplicación adecuada para poder abrir el archivo.

Si se realiza la misma operación, pero enfocándose en las llamadas que se realizan al sistema operativo, se puede observar que el comando es válido y que accede al ADS correctamente, sin embargo el explorador de Windows es quien no puede encontrar una aplicación adecuada para abrir este tipo de archivo. El resultado que obtiene el explorador de Windows para este evento es NAME NOT FOUND.

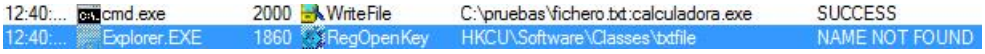


Figura 4.22: Evento NAME NOT FOUND.

En este caso para que Windows 7 ejecute la aplicación asociada a este archivo, se debe invocar directamente a la aplicación que se desea utilizar para visualizar el contenido de un ADS, de manera que se deben realizar pruebas para analizar que aplicaciones se pueden ejecutar desde la línea de comandos y cuáles no.

En equipos con sistema operativo Windows 7 se ha cambiado la manera de ejecutar ADS que almacenan archivos ejecutables. Si se introduce un archivo ejecutable dentro de un ADS y posteriormente se ejecuta alguno de los siguientes comandos:

$$\textbf{\textit{type ejecutable.exe}} > \text{archivo}.\textbf{\textit{txt:ads.exe}}$$

$$\textbf{\textit{start .}}|\text{archivo}.\textbf{\textit{txt:ads.exe}}$$

Se generará un error debido a que en equipos con sistema operativo Windows 7 no se pueden invocar archivos ejecutables a través de ADS, sin embargo, se puede deducir que existe el soporte para ADS en su sistema de archivos NTFS, debido a la inclusión del parámetro /R dentro del comando **DIR**, el cual lista los ADS que contiene un archivo, como se observa en la imagen 4.23 .

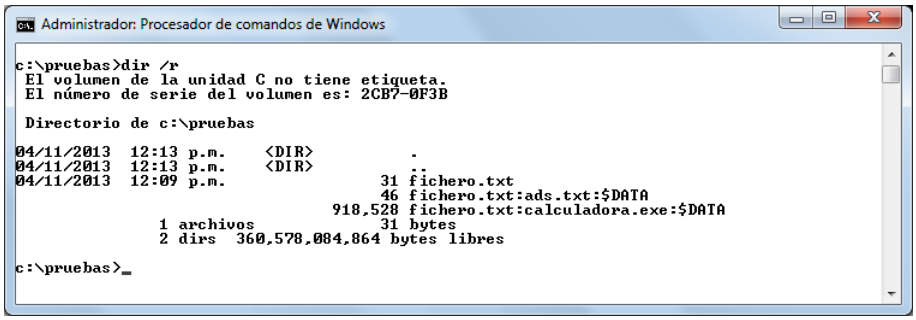


Figura 4.23: Ejecución del comando DIR /R.

El soporte, en este caso y bajo este sistema operativo, es limitado en funciones y se conserva en diversos programas locales como por ejemplo el bloc de notas o la aplicación Paint. Este tipo de soporte, puede cambiar según la política interna de Microsoft con respecto a los ADS.

Como se ha visto en las pruebas realizadas con archivos ejecutables, se observa que el comando **START**, no soporta la ejecución de ADS bajo el sistema operativo Windows 7, debido a que el explorador de Windows deniegan la ejecución. En su lugar, añaden el parámetro /R al comando DIR para buscar ADS.

Por otro lado, mientras exista el soporte para ADS se deben ejecutar, lenguajes como Visual Basic Script, Perl o Python soportan la ejecución de scripts, incluso si están dentro de un ADS. Para ello se han realizado las siguientes pruebas:

### Script en Python

El script en Python utilizado se muestra a continuación:

```
import os # Importamos el módulo OS
ruta = 'C:\Windows\system32\calc.exe'# Path de la aplicación
os.system(ruta) # Ejecución de la aplicación
```

En la figura 4.24 se puede observar la ejecución del script Python.

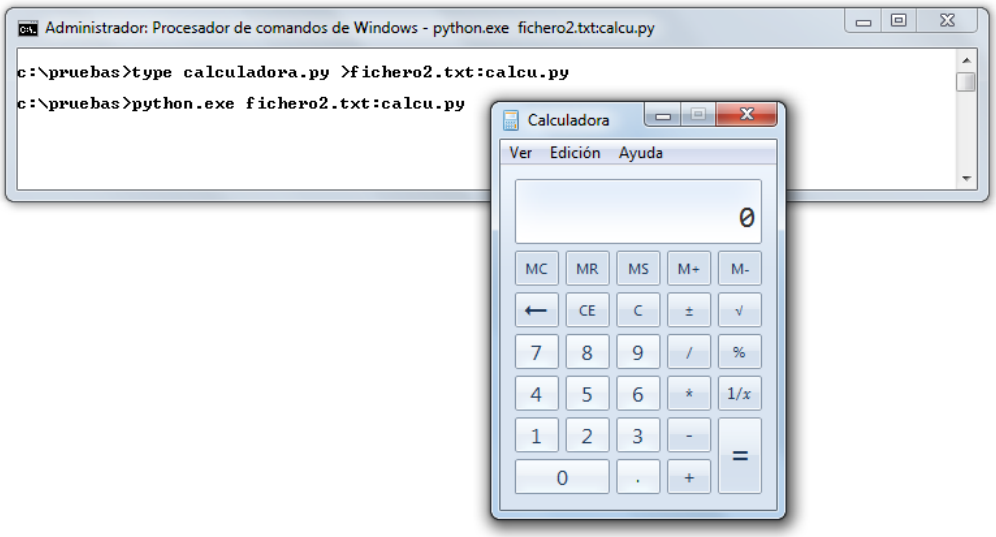


Figura 4.24: Ejecutando un archivo .exe con un script Python.

### Script en Perl

```
system('calc.exe');
```

En la figura 4.25 se puede visualizar la ejecución del script Perl.

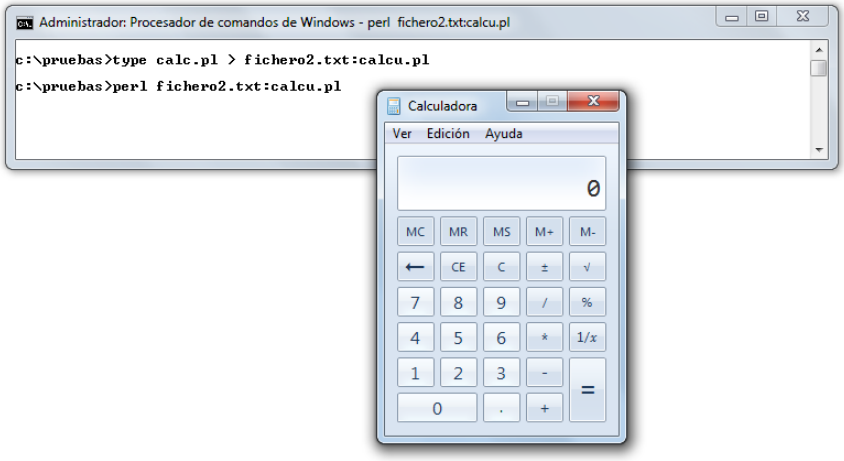


Figura 4.25: Ejecutando un archivo .exe con un script Perl.

### Script en VBS

```
msgbox("Este es un mensaje oculto!!")
```

La figura4.26 muestra la ejecución del script Visual Basic.

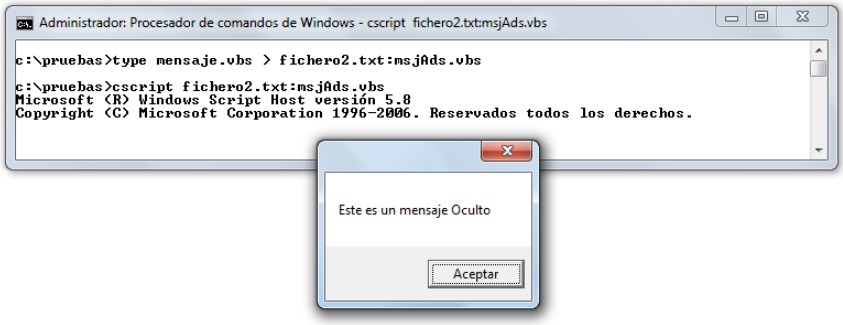


Figura 4.26: Ejecutando un archivo .exe con un script Visual Basic.

### 4.1.10. Ocultar el programa NETCAT dentro de un ADS

Netcat es una herramienta de red que permite escribir y leer datos a través de conexiones de red haciendo uso de los protocolos TCP o UDP.

Para ocultar un programa como Netcat dentro de un ADS primero se debe tener el archivo ejecutable en la carpeta donde se esta trabajando, en este caso tiene el nombre de "nc.exe", o en su defecto conocer la ubicación de dicho archivo, luego se debe añadir el ejecutable al archivo mediante el uso del comando **TYPE**, por último se puede hacer uso de un enlace simbólico de tal manera que se pueda ejecutar el programa Netcat como se indica en la figura 4.27.

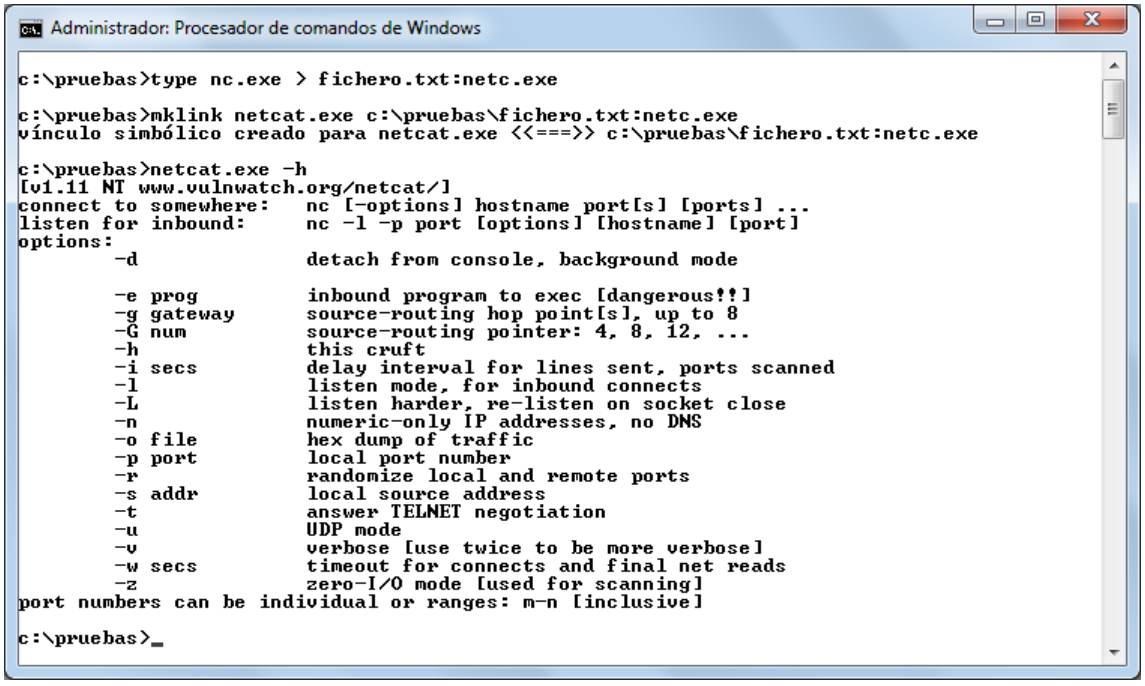


Figura 4.27: Ejecución el programa netcat dentro de un ADS.

### 4.1.11. Adición del programa cmd.exe a un ADS

Como se ha visto la manera para crear un ADS es la misma, lo único que cambia es la manera en que se ejecutan los diversos tipos de archivos.

Agregar el programa cmd.exe de Windows es muy similar al ejemplo anterior; en este caso se creará el ADS dentro del archivo ejecutable notepad.exe, para ello se realiza la copia de dicho archivo a la carpeta en la que se esta trabajando, en este para ejecutarlo se utilizará un enlace simbólico como se muestra en la figura 4.28.

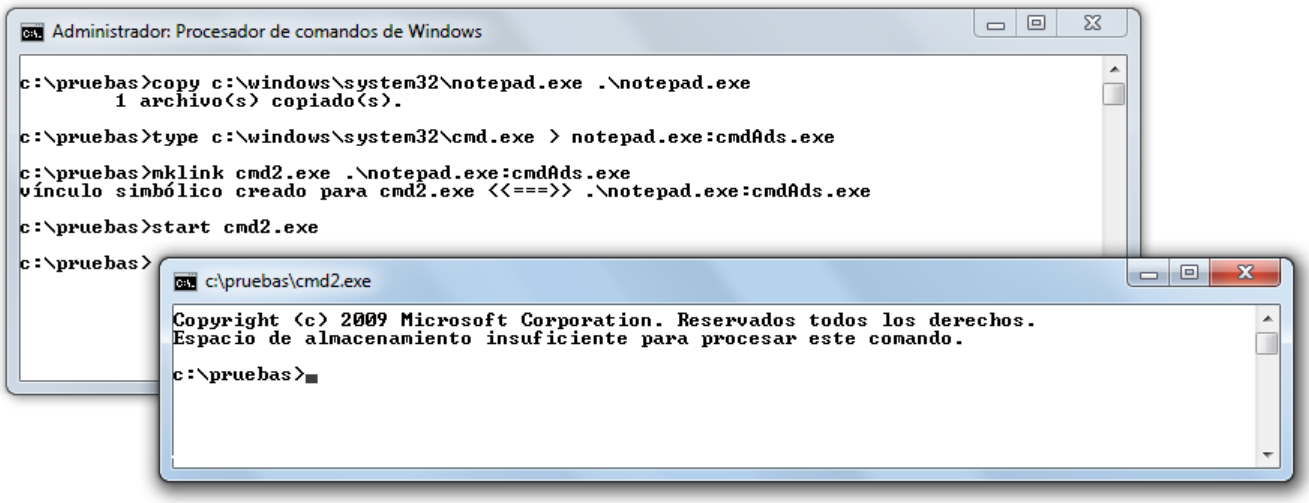


Figura 4.28: Ejecución de un ADS que contiene el archivo ejecutable cmd.exe mediante el uso de un enlace simbólico.

También se puede efectuar el uso de un script en este caso se utilizo un script en Python.

### Script en Python

El script en python es el siguiente:

```
import os
ruta = 'C:\Windows\system32\cmd.exe'
os.system(ruta)
```

Se creo el ADS y posteriormente se invocó de la manera mostrada en la figura 4.29.

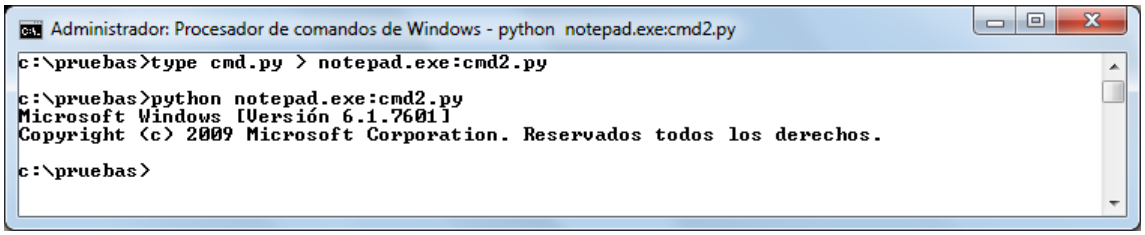


Figura 4.29: Ejecución de un ADS que contiene el archivo ejecutable cmd.exe mediante el uso de un script.

### 4.1.12. Adición de un video dentro de un ADS

Anteriormente se comentó que en equipos con sistema operativo Windows 7 se ha restringido la ejecución de ADS mediante el uso del comando **START**, por ello se debe realizar un estudio sobre los programas que permiten la ejecución de ADS.

A continuación realizan unas pruebas de programas que reproducen video y audio, y que permiten ejecutar ADS que contienen dichos archivos multimedia.

Primero se creó un ADS al cual se le añadió un video como se muestra en la figura 4.30.

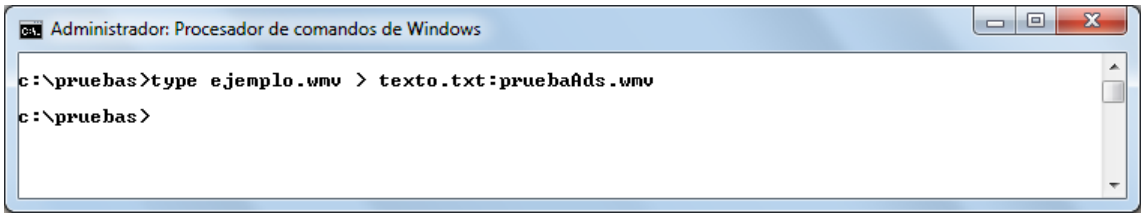


Figura 4.30: Ocultando un video en un ADS.

Para reproducirlo se comenzaron las pruebas con el reproductor por defecto Windows Media Player, sin embargo, como se observa en la figura 4.31 no se reproduce el archivo que contiene el ADS, unicamente abre el reproductor pero no ejecuta nada.

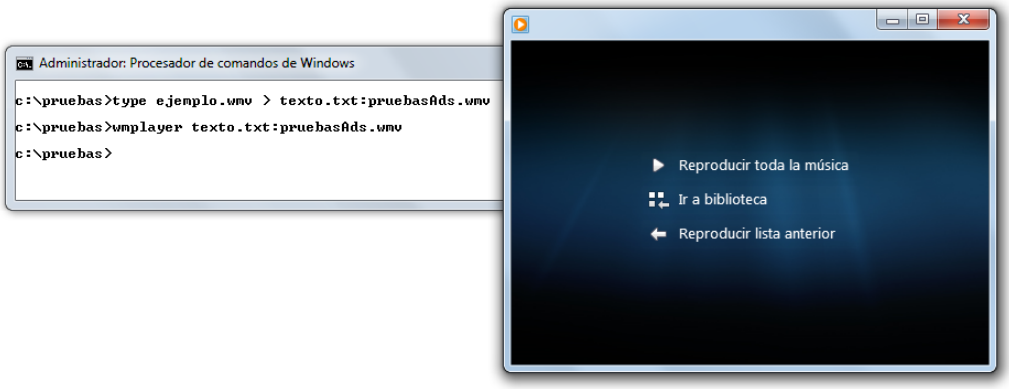


Figura 4.31: Reproducción de video oculto en un ADS con el reproductor Windows Media Player.

Después se realizaron otras pruebas con otros reproductores de video y audio, y con archivos multimedia de diferente extensión.

Se agregó al mismo archivo otro video con extensión ".rmvb" y posteriormente se ejecutó el reproductor VLC, como se observa en la figura 4.32 este reproductor permite la ejecución de ADS.

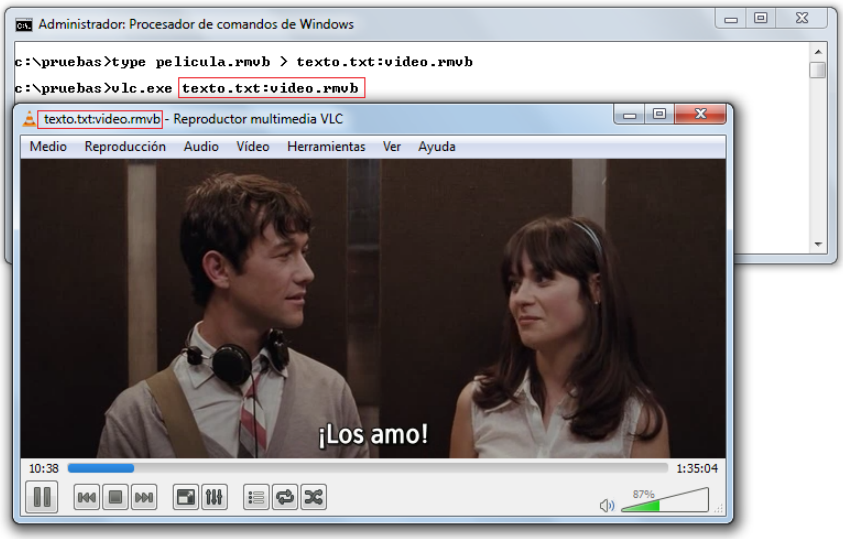


Figura 4.32: Reproduciendo un video oculto en un ADS con el reproductor VLC.

Se realizó la misma prueba con el reproductor MPC tal como se muestra en la figura 4.33, y como se observa estos reproductores de archivos multimedia permiten ejecutar ADS sin ningún problema

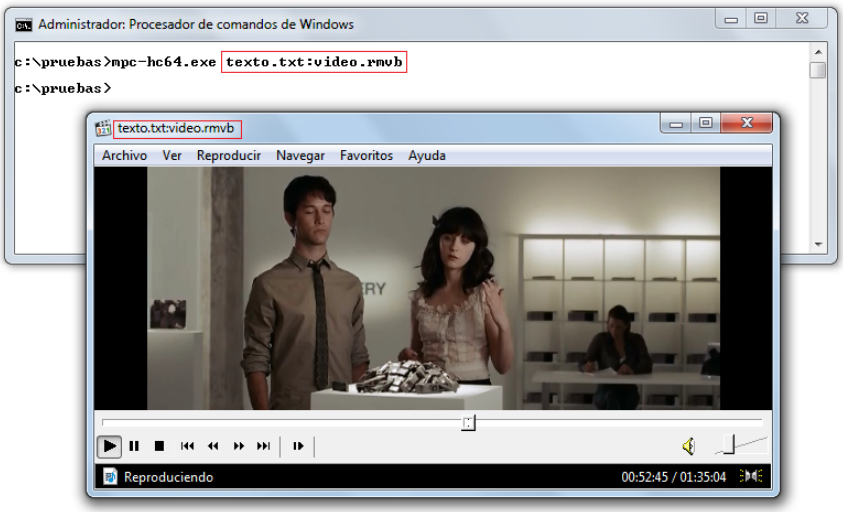


Figura 4.33: Reproduciendo un video oculto en un ADS con el reproductor MPC.

### 4.1.13. Adición de un archivo de audio dentro de un ADS

Hemos visto que aunque el reproductor de Windows no ejecuta los ADS directamente existen otros reproductores que si lo hacen, dependiendo del tipo de reproductor y los formatos que soporte podremos reproducir los ADS que contiene archivos multimedia.

Con los tres reproductores que hemos visto, lamentablemente ninguno reproduce ADS que contiene un archivo en formato mp3, pero como hemos visto existen diferentes técnicas para poder ejecutar los ADS.

La manera más fácil es mediante un script, en este caso el script como en la mayoría de los ejemplos que hemos visto, esta hecho en python, como hemos visto los script en python son muy sencillos pues únicamente basta con indicar la ruta donde se encuentra ubicado el archivo que deseamos añadir.

### Script en python

El script en Python es el siguiente:

```
import os
ruta = 'C:\pruebas\c.mp3'
os.system(ruta)
```

Posteriormente se generó el ADS y se le adjunto el script, después se invocó el ADS y de esta manera se reproducirá automáticamente el archivo como se observa en la figura 4.34.


Figura 4.34: Reproducción de una canción oculta dentro de ADS mediante un script.

Otra manera como se ha visto anteriormente, es mediante el uso de enlaces simbólicos. Para su reproducción, al momento de crear un enlace simbólico se genera un icono en la carpeta que se haya indicado tal como se muestra en la figura 4.35.

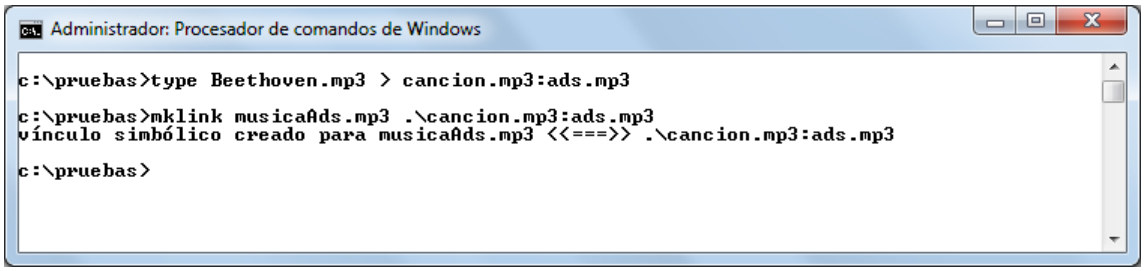


Figura 4.35: Creación de un ADS que contiene un archivo de audio.

Para reproducir el archivo solo se debe dar doble clic al enlace simbólico y se reproducirá tal como se observa en la figura 4.36.

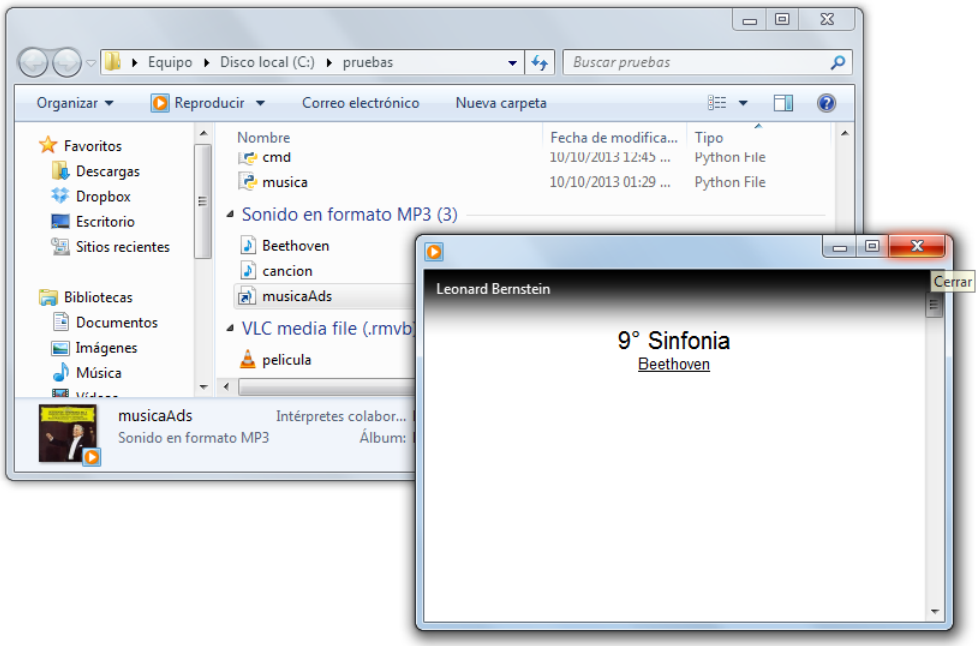


Figura 4.36: Reproducción de una canción oculta dentro de ADS mediante un enlace simbólico.

Si se desea reproducir un archivo de audio mediante la línea de comandos de Windows el único reproductor que nos permite esta opción es el reproductor MPC.

Se realizaron diferentes pruebas que permitieron la creación y ejecución de diferentes archivos ocultos dentro de un ADS, sin importar el tipo de archivo.

Aunque de manera directa Microsoft no permite la manipulación de ADS existen diferentes formas de poder ejecutarlos.

### 4.1.14. Algoritmos básicos de los ADS

La mayoría de las herramientas que se encargan de detectar o manipular ADS se basan en dos algoritmos básicos para su correcto funcionamiento.

A continuación se muestran dichos algoritmos.

### 4.1.15. Algoritmo de recorrido en un directorio para la búsqueda de ADS

Este algoritmo se encarga de recorrer todos o algún directorio especificado por el usuario con la finalidad de encontrar ficheros o directorios que contengan ADS. Su estructura en pseudocódigo es la siguiente:

```
NIVEL SUPERIOR:
Configurar los privilegios de procesos para la copia de seguridad
IF no hay ningún directorio especificado THEN
 Obtener directorio actual
 Guardar la ruta del directorio para un recorrido posterior
 Enumerar los ADS
ELSE
 FOR EACH directorio específico LOOP
 Poner la ruta del directorio en forma canónica
 Guardar la ruta del directorio para un recorrido posterior
 Enumerar los ADS
 END LOOP
END IF
FOR EACH directorio guardado para el recorrido LOOP
 Limpiar el valor de ruta si es necesario
```

```
 FOR EACH elemento en el directorio LOOP
 IF se trata de un directorio,
 pero no es un directorio de puntos THEN
 Guardar para un recorrido posterior
 END IF
 Enumerar y obtener los ADS
 END LOOP
END LOOP
```

### 4.1.16. Algoritmo de enumeración de ADS

Como su nombre lo indica, éste algoritmo se encarga de listar o enumerar los ADS que se encuentran en una determinada ubicación. Su estructura en pseudocódigo es la siguiente:

```
Enumeración de ADS:
IF la ruta es para un archivo THEN
 Guarda la ruta como el primer flujo
END IF
Abrir el objeto representado por la ruta de lectura
LOOP
 Usando BackupRead(),
 leer la primera parte de la cabecera del ADS
 IF no hay nada que leer THEN
 Detener el loop
 END IF
Determinar si la cabecera pertenece a un ADS
Determinar el tamaño de los datos
Obtener la parte de la cabecera del ADS
IF este es un flujo alternativo de datos THEN
 Obtener el nombre del ADS
 Guardar la ruta completa con el nombre del ADS
END IF
```

```
Usando BackupSeek() con el tamaño del ADS,
encontrar la siguiente cabecera del flujo
END LOOP
Usando BackupRead() con parametros especiales
para detener la lectura del objeto
Cerrar el objeto
```

## 4.2. Desarrollo

De manera general la metodología que se propone en este trabajo parte del hecho de que existe un emisor y un receptor que desean establecer una comunicación secreta de manera que una tercera parte ubicada entre ambas no sea capaz de detectar la existencia de tal comunicación, tal como se muestra en la figura 4.37.

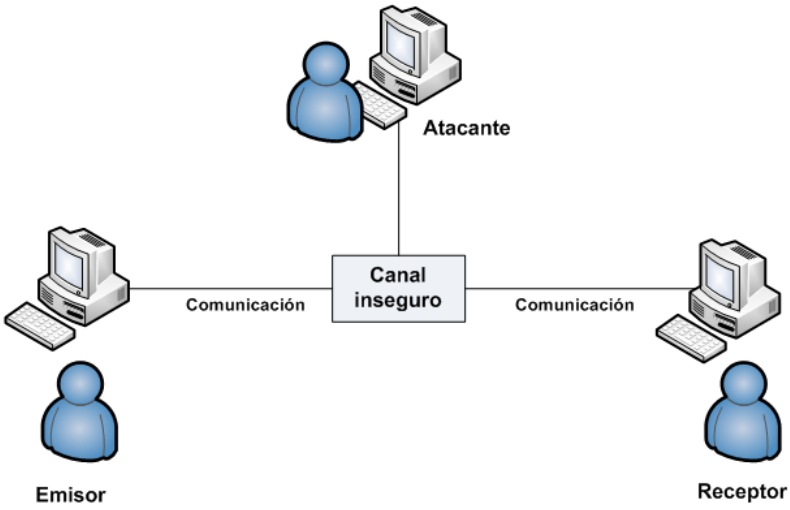


Figura 4.37: Comunicación oculta entre emisor y receptor en un canal inseguro.

Para ello se propone la utilización de esteganografía de red de manera que permita ocultar información mediante el uso de canales encubiertos a través de un protocolo de red, junto con el uso de ADS.

De manera general, esta propuesta de solución se divide en dos partes:

1. **Creación y manipulación de ADS:** Esta primer parte se basa en la creación de ADS dentro de un archivo, permitiendo ocultar cualquier tipo de información sin tener un límite sobre el tamaño de los archivos a ocultar.

   Existiendo la posibilidad de ocultar archivos multimedia como audio, video, imágenes, o cualquier otro tipo de archivos; generando un archivo con información oculta para el posterior envió a un receptor.

2. **Creación y manipulación de un canal encubierto:** En esta parte se propone crear un canal encubierto que lleve unicamente un mensaje oculto, esto debido a que una de las limitaciones del uso de canales encubiertos es su ancho de banda.

   El mensaje a ocultar simplemente contendrá información referente al archivo que contiene archivos ocultos.

### 4.2.1. Selección del protocolo a utilizar para la creación del canal encubierto

La técnica que se escogió para la creación de un canal encubierto es el de asignación de mensajes secretos en el área de datos de un protocolo, permitiendo de esta manera la encapsulación de información dentro del área de datos de un protocolo.

Este caso se da cuando un protocolo de uso común en la red incluye un campo de datos que se puede alterar, de manera que si alguien estuviera vigilando la red podrá notar la presencia del uso del protocolo pero, solamente en el caso de que realice un análisis detallado del área de datos de dicho protocolo, posiblemente pueda notar la presencia de información oculta. Sin embargo, si se realiza un procedimiento efectivo para establecer una comunicación secreta, no será capaz de detectar que se está transmitiendo información de manera oculta.

Se seleccionó el uso de esta técnica por una razón principal, trabaja con bloques de bits lo que permite que se pueda enviar más información a comparación de otros campos por lo que presenta un mayor ancho de banda.

Posteriormente el protocolo elegido a partir de la selección de la técnica de esteganografía de red, fue el protocolo ICMP.

Razones por las que se seleccionó dicho protocolo:

- El protocolo ICMP trabaja con mensajes de errores, ofreciendo una gran variedad de opciones para crear un canal encubierto.
- Soporta el tráfico de difusión, es decir, se puede transmitir información un un nodo emisor a varios nodos receptores, lo que permite que exista la posibilidad de tener diversas comunicaciones secretas con varios receptores sin limitarse a un número determinado.
- El Ping es la aplicación usada por el protocolo ICMP, está sera la base para la creación del canal encubierto que se propone crear.
- La estructura de ICMP es ambigüa lo que permite que dicho protocolo pueda ser manipulado de manera que no altere las operaciones de la red o del sistema.
- El área de datos del protocolo tiene un tamaño opcional, por lo que proporciona un ancho de banda adecuado para la creación del canal encubierto.
- Se cuenta con los permisos y recursos necesarios para explotar dicho protocolo.

#### 4.2.1.1. Estructura del protocolo ICMP

Los datagramas que se transmiten a través de la red pueden perderse sin llegar a su destino o incluso pueden llegar dañados. El protocolo ICMP se encarga de informar al origen si se ha generado algún tipo de error al momento de realizar la entrega del mensaje. Por lo que se encarga de transportar distintos mensajes de control.

El protocolo ICMP es un protocolo que se encuentra en el nivel de red y que se encarga de detectar errores, por tal motivo genera mensajes de control y de error, notificándole al protocolo IP, ICMP no utiliza ningún protocolo de transporte ya que el mensaje ICMP no va dirigido a la aplicación que generó el error.

Los mensajes de error ICMP se envían a través de la red en forma de datagramas, ocasionando de esta manera que los mismos mensajes de error pueden contener errores.

Por tal motivo el formato de ICMP cambia dependiendo del tipo de mensaje, pero de manera general su formato es el mostrado en la figura 4.38.

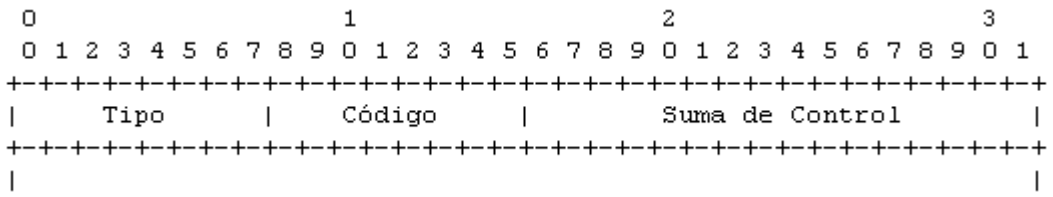


Figura 4.38: Datos ICMP dependiendo del tipo de mensaje.

Todos los mensajes ICMP comienzan con los mismos campos:

- **Tipo:** Es un campo de 8 bits que identifica el tipo de mensaje ICMP.
- **Código:** Campo de 8 bits que muestra información sobre el tipo de mensaje, la interpretación depende del tipo de mensaje.
- **Suma de comprobación:** Campo de 16 bits, proporciona un método para la determinación de la integridad del mensaje ICMP.

Los tipos de mensajes que ofrece ICMP se muestran en la tabla 4.1.

Tabla 4.1: Tipos de mensajes ICMP

| **Campo** | **Tipo de mensaje ICMP** |
|---|---|
| 0 | Respuesta de eco |
| 3 | Destino inalcanzable |
| 4 | Disminución de flujo de la fuente |
| 5 | Redireccionar (cambiar ruta) |
| 8 | Solicitud de eco |
| 11 | Tiempo excedido por el datagrama |
| 12 | Problema de parámetro de un datagrama |
| 13 | Solicitud de fecha y hora |
| 14 | Respuesta de fecha y hora |
| 15 | Solicitud de información |
| 16 | Respuesta de información |
| 17 | Solicitud de máscara de direcciones |
| 18 | Respuesta de máscara de direcciones |

ICMP cumple diferentes propósitos en comparación con los protocolos TCP y UDP ya que generalmente no se utiliza por las aplicaciones de usuario en la red, pero existe una excepción y es la herramienta Ping, la cual determina si un equipo se encuentra disponible a través de la red, de tal manera que si se recibe respuesta significa que hay conectividad con dicho equipo.

Lo que realmente ocurre es que el equipo que envía el ping emite un paquete de datos de tipo ICMP ECHO-REQUEST (Solicitud de eco) de manera que el equipo destino al recibir este paquete, genera otro de respuesta denominado ICMP ECHO-REPLY (Respuesta de eco). Cuando el equipo que envía el ping comienza a recibir paquetes ICMP ECHO-REPLY puede concluir que el equipo destino está disponible.

Por tal motivo resulta un candidato interesante para ocultar información, ya que se puede utilizar el campo de área de datos.

Los sistemas operativos utilizan el campo ICMP ECHO como una firma llenándolo con una secuencia de datos que se encuentra más o menos predefinida, por ejemplo, en Windows se incluye el abecedario, mientras que en Linux se usa una secuencia de dato más aleatoria que finaliza con los caracteres "01234567", basándose en la idea de que si los paquetes de respuesta contienen la misma secuencia de datos entonces son correctos.

Sin embargo, no hay un estándar definido sobre cual debe ser la secuencia de datos por lo que nosotros podemos utilizar esta ambigüedad para ocultar información y enviarlos a un destinatario mediante el uso de un ping.

### 4.2.2. Procedimiento de comunicación secreta entre el emisor y el receptor.

El diagrama a bloques propuesto para el proceso de envió de información, se muestra en la figura 4.39.

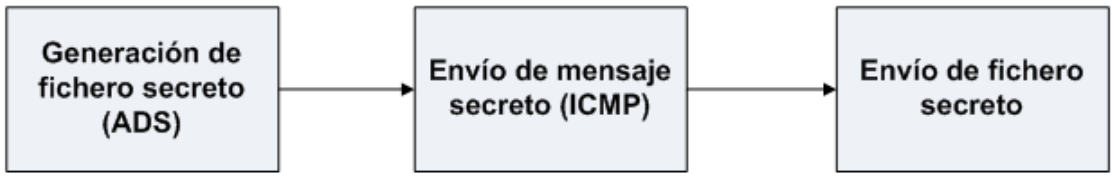


Figura 4.39: Diagrama a bloques del proceso de ocultación de información.

Los pasos que se proponen para una comunicación secreta entre el emisor y el receptor son los siguientes:

1. El emisor genera un archivo con archivos ocultos mediante el uso de ADS.
2. El emisor y el receptor deben conocer sus direcciones IP.
3. El emisor enviará un mensaje oculto con información sobre el archivo secreto a través de un ping.

4. El receptor estará a la espera de la solicitud para obtener información.

5. El emisor enviará una serie de ficheros para no levantar sospechas.

6. El receptor obtendrá el archivo y podrá consultar la información oculta.

A continuación se muestra el desarrollo que se llevo a cabo para efectuar el procedimiento de ocultar información mediante la utilización de esteganografía de red junto con el uso de ADS.

### 4.2.3. Requerimientos

- Computadoras con Sistema operativo Windows 7 y sistema de archivos NTFS

### 4.2.4. Herramientas utilizadas

- **Scapy:** Es una utilidad para crear y manipular paquetes de datos.
- **Wireshark:** Es un analizador de protocolos d red.
- **Windump:** Herramienta que trabaja sobre la línea de comandos, su finalidad es analizar el tráfico de datos que circula por la red.
- **Arcanum Editor:** Herramienta que no requiere de una instalación, permite encriptar archivos de texto en Base64, Rot13, MD5 o SHA-1, AES.

### 4.2.5. Generación de un archivo que contiene ADS

Para la generación de ADS, se tiene una carpeta con los archivos a ocultar la cual tiene por nombre "Archivos", en la figura 4.40 se pueden observar doce tipos de archivos diferentes a ocultar.

Esta carpeta se encuentra ubicada en la máquina del emisor, con la siguiente ruta: "C:\Archivos".

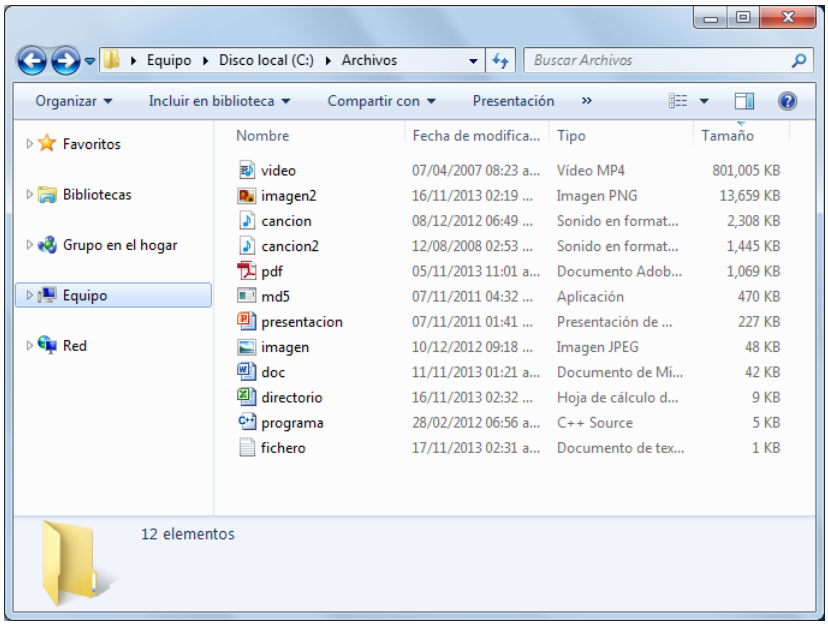


Figura 4.40: Visualización de la carpeta “Archivos” con los documentos a ocultar.

A continuación se comenzará con la creación de los ADS dentro un archivo de texto llamado “notas.txt”; pero para ello se debe crear el archivo.

Éste archivo se encuentra en otra carpeta llamada “emisor” y ubicada en “C:\emisor”. Como se puede observar en la figura 4.41 el archivo “notas.txt” tiene un tamaño de 2 976 bytes y fue creado a las 6:52 pm; la carpeta de momento solamente contiene este archivo.

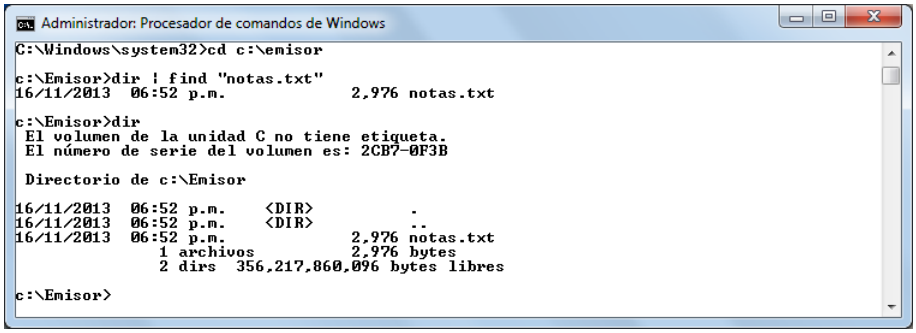


Figura 4.41: Creación y propiedades del archivo “notas.txt”.

Ahora se pasará a la creación de los ADS con los archivos contenidos en la carpeta "Archivos" dentro del archivo "notas.txt", como se aprecia en la figura 4.42.

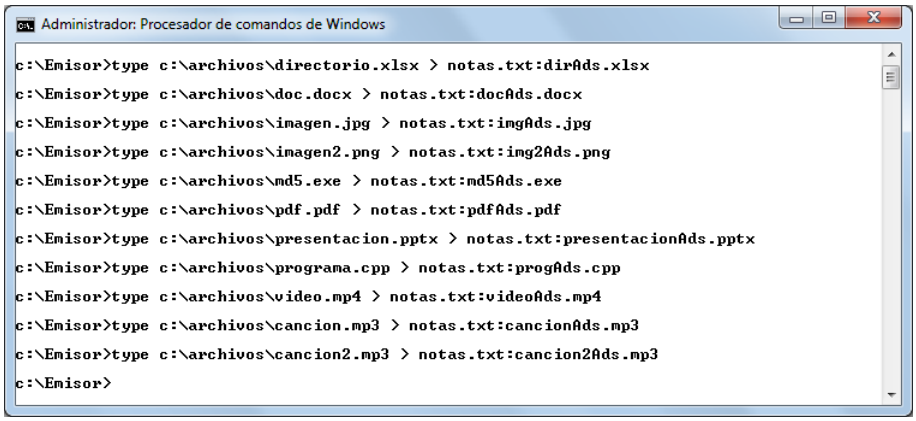


Figura 4.42: Creación de ADS dentro del archivo "notas.txt".

Se realizó una consulta sobre la información del archivo para analizar sus propiedades y se pudo observar que conservo el mismo tamaño, existiendo únicamente un cambio en la fecha de modificación. Después, se realizó otra consulta para verificar la creación de los ADS dentro del archivo, comprobando que efectivamente se crearon once ADS que fueron creados anteriormente, como se muestra en la figura 4.43.

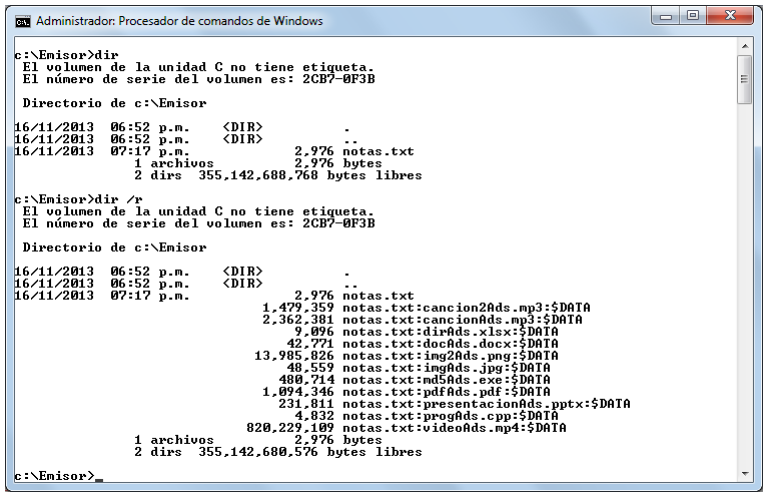


Figura 4.43: Propiedades del archivo "notas.txt" después de añadirle ADS.

### 4.2.6. Direcciones IP

Para poder continuar con el envió del mensaje secreto se necesita conocer las direcciones IP del emisor y receptor, esto se obtiene mediante la ejecución del comando "IPCONFIG" a través de la línea de comandos, en cada maquina respectivamente, las figuras 4.44 y 4.45 muestran las direcciones IP del emisor y del destinatario.

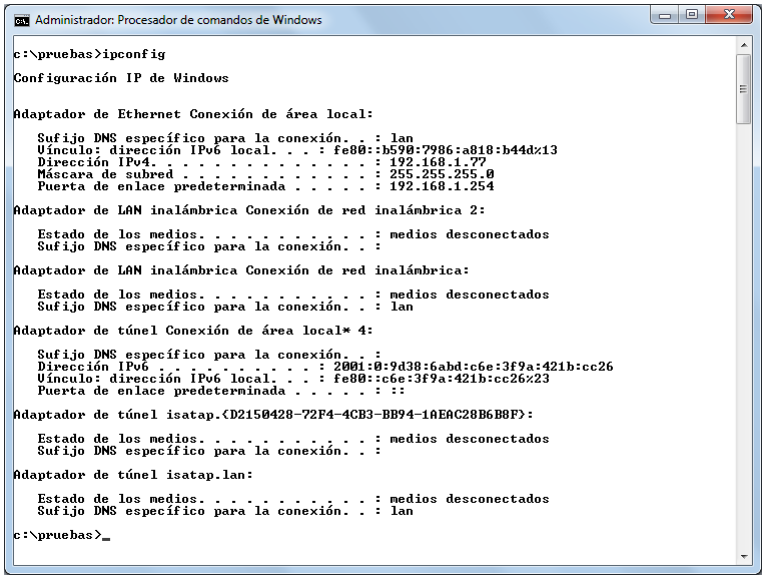


Figura 4.44: Dirección IP del emisor.

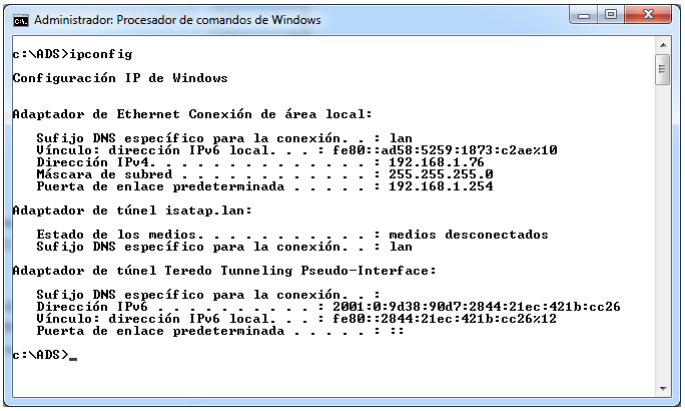


Figura 4.45: Dirección IP del receptor.

### 4.2.7. Envió del mensaje oculto

Para realizar esta parte del proceso se hizo uso de la herramienta Scapy con la cual se creó el mensaje oculto y se encapsuló dentro del protocolo ICMP. A continuación en la figura 4.46 se muestra el diagrama de flujo de dicho procedimiento.

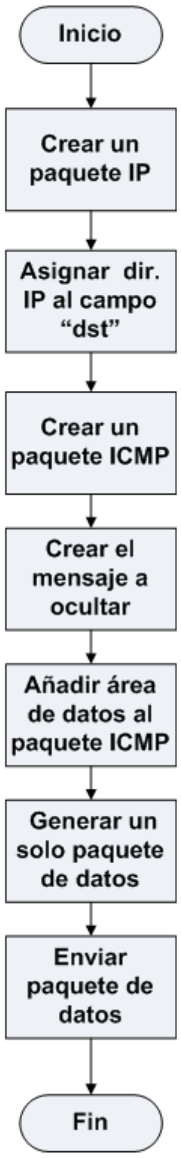


Figura 4.46: Diagrama de flujo del envío de un mensaje oculto a través del protocolo ICMP.

El siguiente código muestra el procedimiento que se siguió para generar el canal encubierto:

```
ip = IP()

ip.dst = 192.168.1.76

icmp = ICMP()

mensaje = Este es un mensaje oculto.

icmp.add_payload(mensaje)

packet = ip / icmp

sr1(packet)
```

**Explicación del código**

1. Se crea un paquete IP con los valores por defecto.
2. Se asigna el valor de la dirección IP de la máquina del emisor al campo "dst" de paquete IP que se creo en el paso 1.
3. Se crea un paquete ICMP con los valores por defecto.
4. Se escribe el mensaje secreto.
5. Se añade el mensaje secreto dentro del área de datos del protocolo ICMP.
6. Unión de capas para generar un paquete de datos.
7. Se envía el paquete de datos generado en paso 6, trabajando únicamente con la capa tres (Capa de Red) y se recibe la primer respuesta.

Una vez que se ingresaron los comandos a través de la consola de Scapy aparecerá un mensaje indicando que se ha enviado el paquete de datos como se muestra en la figura 4.47.

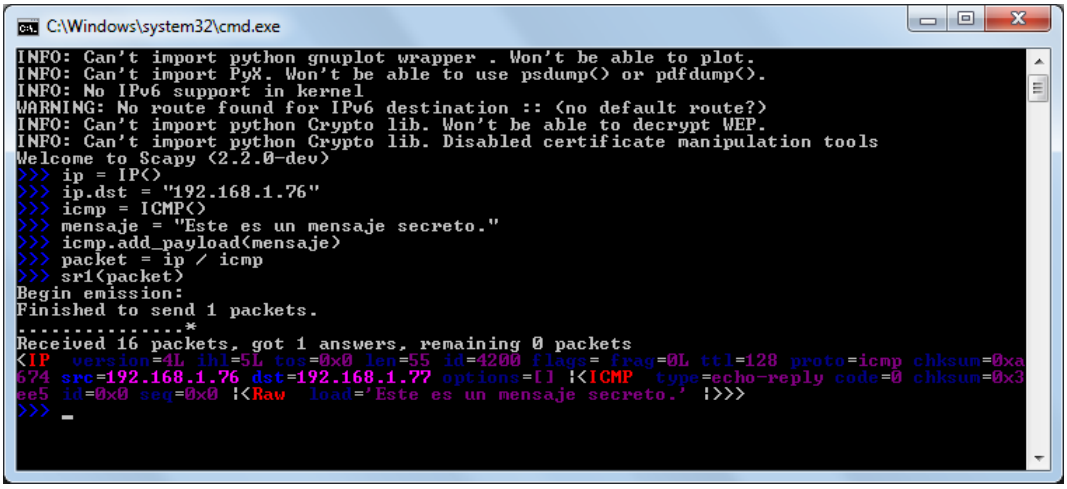


Figura 4.47: Encapsulación de un mensaje en el protocolo ICMP.

Posteriormente el emisor podrá visualizar dicho mensaje con la herramienta Windump.

### 4.2.8. Envió del archivo oculto

Es importante mencionar que para el envío de ficheros que contienen ADS se necesita estar conectados a una misma Red de Área Local (LAN).

Si se desea enviar un archivo que contiene ADS por correo electrónico o se desea comprimir, con estos métodos se pierden los ADS y no podrán llegar a su destino.

Para enviar un archivo a través de una LAN es necesario configurar los equipos para que permitan el uso compartido archivos a través de la red.

Una vez que se ha realizado dicha configuración, desde la línea de comandos se pueden enviar los ficheros con ADS sin ningún problema, en este caso se hace el envío de varios ficheros que contienen ADS, como se muestra en la figura 4.48 .

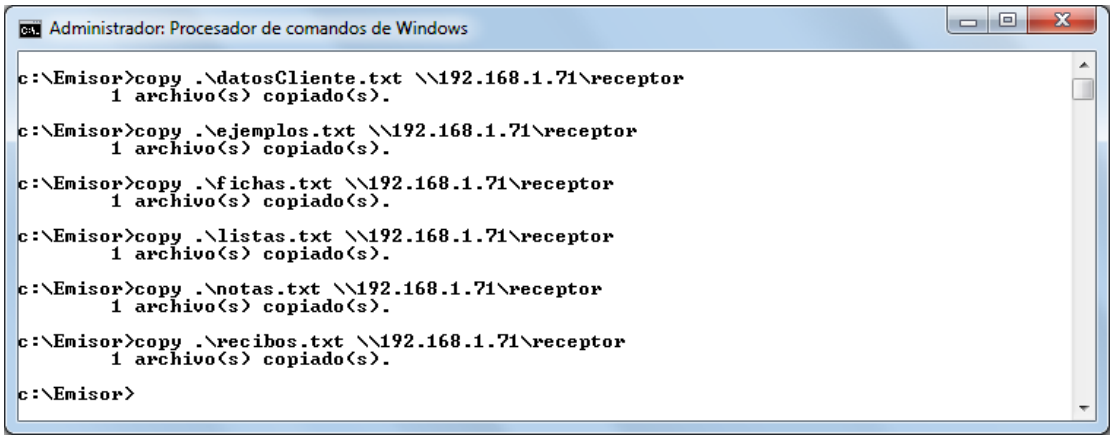


Figura 4.48: Envió de ficheros a través de una red LAN.

## 4.3. Resultados

A continuación se muestran los resultados obtenidos para la propuesta de una comunicación secreta utilizando esteganografía de red y ADS.

Primero se analiza la parte enfocada al mensaje oculto para posteriormente concluir con los ADS ocultos dentro de un archivo.

### 4.3.1. Mensaje oculto

Durante el desarrollo se observó como se realizó la comunicación entre el emisor y el receptor a través de un envió ping.

Para realizar un monitoreo de los paquetes de datos que circulaban a través de la red se utilizó la herramienta Wireshark la cual se estuvo ejecutándo antes del envió de la información para realizar el monitoreo de los paquetes que fluían a través de la red.

Se aplicó un filtro para que mostrará unicamente las actividades relacionadas con el protocolo ICMP.

Wireshark establece 3 zonas de datos:

- La primer zona, ubicada en la ventana superior muestra la lista de los paquetes capturados. Incluye hora, fuente, destino, protocolo y una descripción breve de cada uno. Según el paquete que esté seleccionado, se controla la información que aparece la ventana intermedia.

- La segunda zona, ubicada en la ventana intermedia muestra detalladamente información sobre el paquete seleccionado. Incluye el nombre de los protocolos empleados en los distintos niveles de la arquitectura y los valores correspondientes a los campos de cada uno de los protocolos en listas desplegables.

- La tercer zona, corresponde a la ventana inferior la cual muestra el valor los datos del paquete en hexadecimal y ASCII. Al seleccionar alguno de los campos en la ventana intermedia, se destaca el rango de valores correspondientes a dicho campo en el paquete.

De manera general se hizo un análisis de los resultados obtenidos, en los cuales no se mostraron cambios significativos, salvo que como podemos observar en la figura 4.49, en la zona de los valores de datos, aparece el mensaje que el emisor envió al receptor. Esto se hizo con el motivo de que se apreciará y ubicará la parte del paquete de datos que se manipuló.

Como se pudo apreciar en la captura del paquete ICMP, se muestra claramente el contenido del mensaje en el campo Data. Para evitar la detección y además generar confidencialidad e integridad al mensaje se recomienda usar un método de cifrado, para posteriormente ser empaquetado y enviado a través del ping.

De esta manera si hay alguien interceptando la comunicación solamente verá una serie de bits aparentemente aleatorios que no tendrían por qué levantar sospecha alguna.

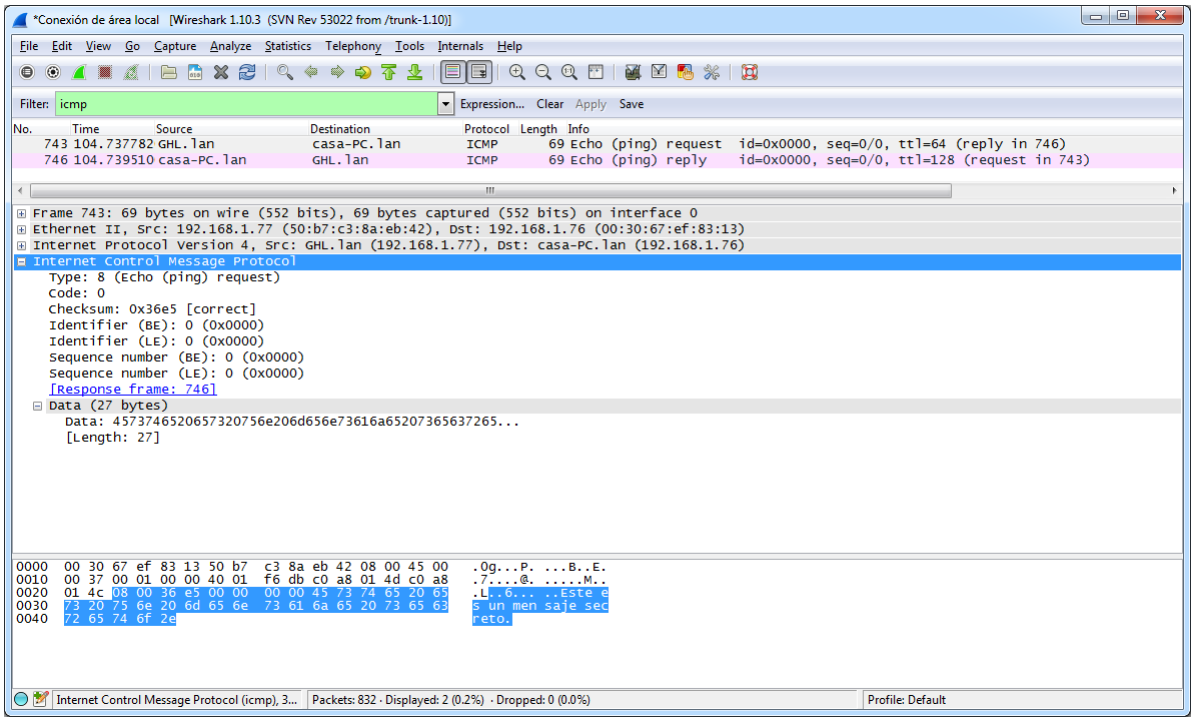


Figura 4.49: Captura de los paquetes del protocolo ICMP a través de Wireshark.

Por otra parte, se hizo uso de la herramienta Windump para analizar los paquetes de datos que fueron enviados a la máquina del emisor, como podemos observar en la figura 4.50, el mensaje llegó sin sufrir modificación alguna.

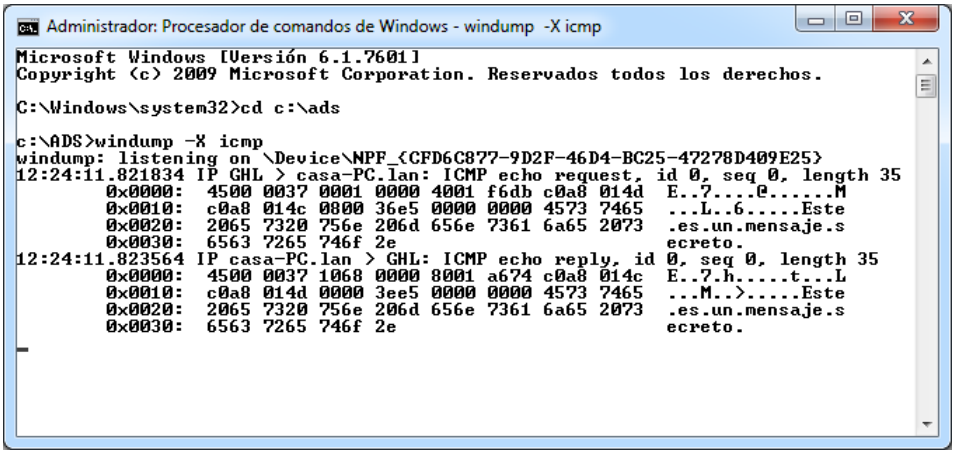


Figura 4.50: Captura de paquetes de datos con Windump.

En la figura 4.51 se muestra que se ha enviado otro paquete de datos con información oculta al emisor, dicha información fue previamente cifrada mediante el uso de la herramienta Arcanum utilizando un cifrado en base 64.

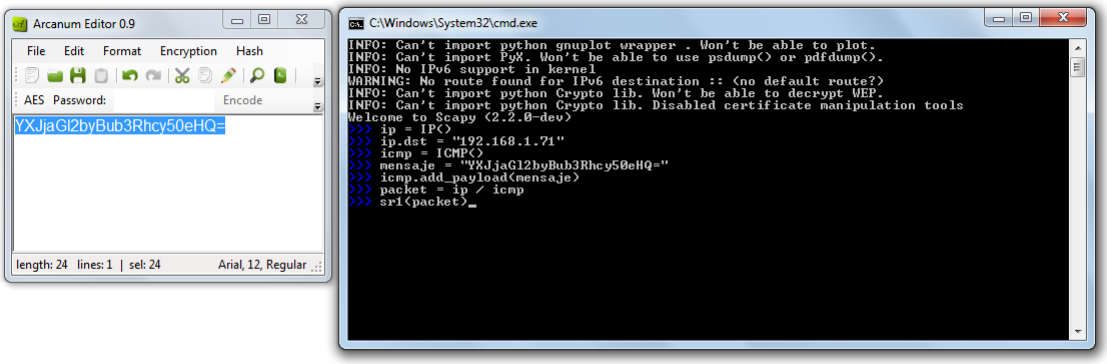


Figura 4.51: Envió de paquete de datos cifrado.

A continuación la figura 4.52 muestra la captura de dicho paquete de datos a través de Wireshark.

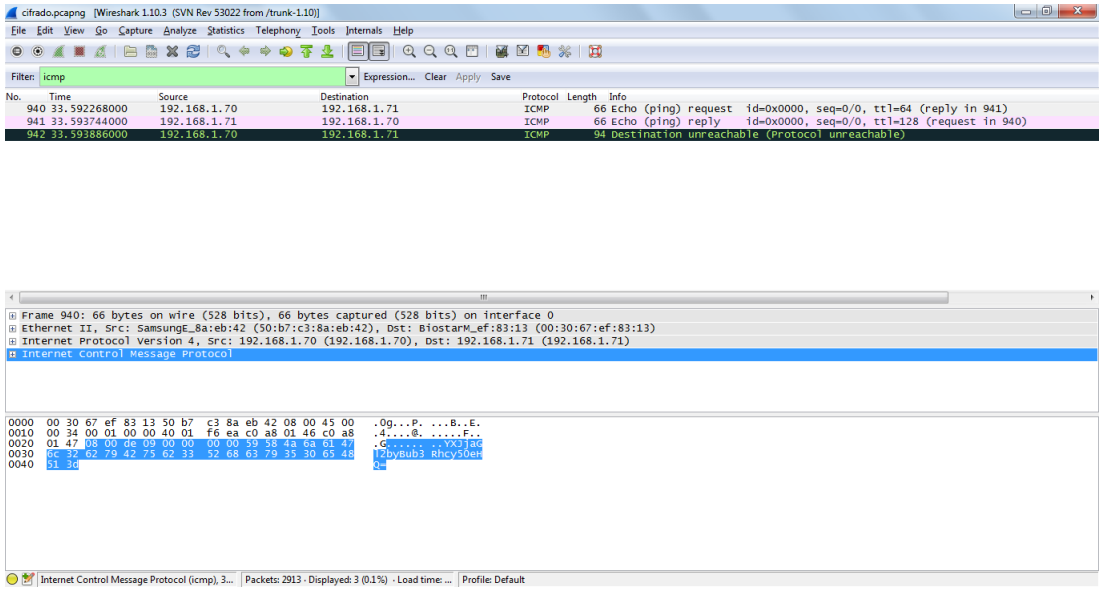


Figura 4.52: Visualización del mensaje cifrado a través de Wireshark.

El receptor por su parte visualizará la información mediante el uso de Windump como se muestra en la figura 4.53.

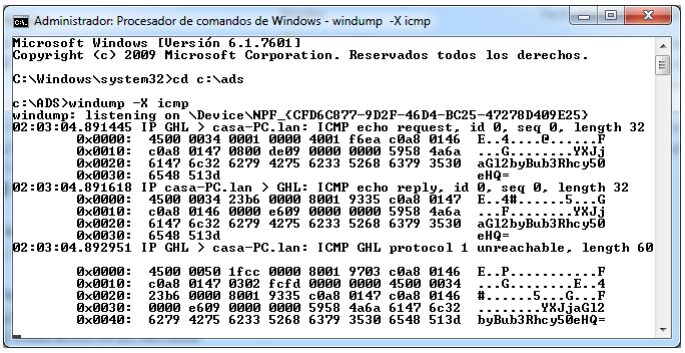


Figura 4.53: Visualización del mensaje oculto a través de Windump.

Posteriormente copiará el mensaje cifrado y lo introducirá en el programa Arcanum Editor para poder obtener el mensaje original como indica la figura 4.54.

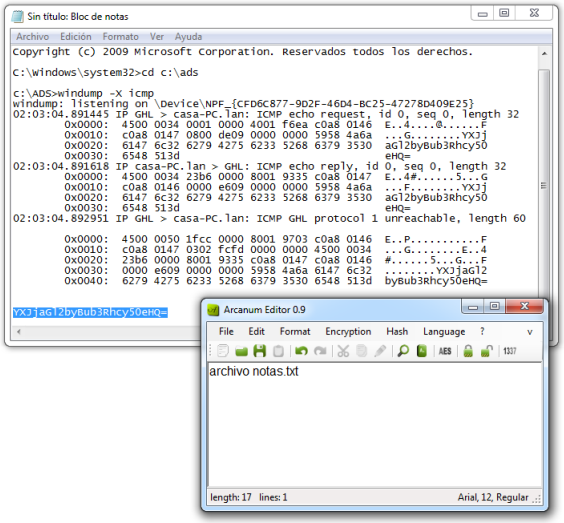


Figura 4.54: Obtención del mensaje cifrado.

En el caso en que se deseará enviar información que sobrepase el tamaño del campo, se puede separar y enviar en varios paquetes, por lo que en vez de enviar un único ping se realizará el envío de varias solicitudes, lo cual hasta cierto punto esta dentro del rango de lo normal, cuando se utiliza legítimamente este protocolo.

### 4.3.2. Archivo oculto

Una vez que el receptor tenga conocimiento sobre el archivo que tiene los archivos ocultos, se procede a revisar la carpeta donde se enviaron los ficheros, que previamente se acordó con el receptor para su posterior visualización.

En la figura 4.55 se muestra la carpeta de nombre "receptor" con los archivos que fueron transmitidos por el emisor a la máquina del receptor.

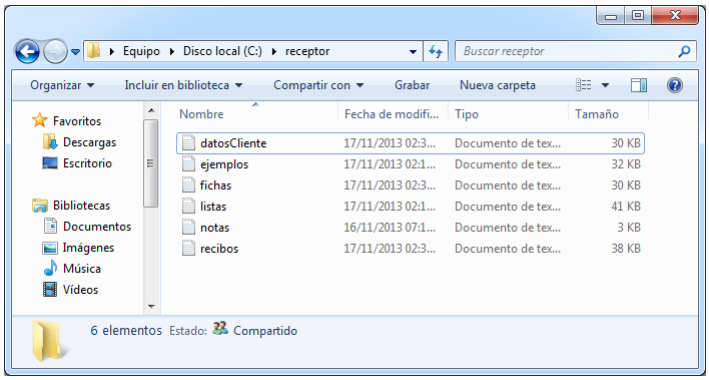


Figura 4.55: Visualización de archivos recibidos por el emisor.

### 4.3.3. Información adicional

Como información adicional, se realizó el envío de diferentes ficheros que contenían varios ADS, algunos de gran tamaño y otros muy pequeños. Al momento de realizar un monitoreo sobre todos los ficheros enviados, el analizador Wireshark dejaba de funcionar trabando la aplicación, se realizaron tres veces esta prueba y en las tres ocasiones se obtuvieron los mismos resultados, aunque se pudo notar que se detectaba la transferencia de ADS.

Por tal motivo, algunas de las desventajas que surgieron al momento de enviar ADS fueron:

- Primeramente el hecho de que solo se reduce a un envío dentro de una LAN, presentando una limitación para la transferencia de archivos, en caso de que una de las partes no se encuentre dentro de la red.

- Al momento de monitorizar el tráfico de datos a través de la red, es notoria la existencia de que se están transmitiendo ficheros que contienen ADS, por lo no pasa inadvertida ante un posible intruso y deja de existir la comunicación oculta.

Una posible solución a dichos inconvenientes pueden ser:

- Crear una copia de seguridad de la carpeta con los archivos que contienen ADS, esta opción permite conservar los flujos alternativos de manera que si se envía este copia de seguridad al emisor podrá recibir la información con los ADS.

Para fines de esta investigación no se analizó esta posible solución.

# Capítulo 5

# Conclusiones y trabajo a futuro

A continuación se desglosan las conclusiones que se obtuvieron al realizar este trabajo de investigación.

### 5.0.4. Conclusiones

Los objetivos que se cumplieron en esta investigación fueron:

- Se aplicó un procedimiento que mediante el uso de esteganografía de red y ADS permitió ocultar información para un posterior envío a través de una comunicación entre un emisor y un receptor.

- Se realizó una investigación sobre las diferentes técnicas esteganográficas que existen en la actualidad, dando un enfoque principal a aquellas técnicas aplicadas a la esteganografía de red.

- A partir de la investigación realizada se definió la técnica a utilizar lo cual permitió realizar la selección del protocolo ICMP como medio para explotarlo y crear un canal encubierto, debido a la gran ambigüedad que presenta.

- A partir del protocolo seleccionado se propuso un procedimiento que implicó el uso de esteganografía de red y de ADS, teniendo como principal objetivo proteger la información que fluye a través de Internet mediante el uso de técnicas de ocultación de información.

- Se llevó a cabo la implementación del procedimiento propuesto en dos equipos de cómputo con sistema operativo Windows 7 y sistema de archivos NTFS.
- Se realizarón las pruebas que permitieron comprobar que es posible ocultar información a través de las técnicas seleccionadas.

Por otra parte, se pudo observar que mediante el uso de la esteganografía de red se cumple el objetivo de establecer una comunicación oculta sin generar sospecha alguna, sin embargo el uso de ADS permite ocultar información pero se puede detectar la comunicación oculta al momento de transferir los archivos a través de un red.

Sin embargo, ésta característica del sistema de archivos NTFS cumple el objetivo de proteger la información mediante su ocultación a pesar de que no se envíe por un canal de comunicación.

Cabe destacar que aunque existe documentación sobre ADS, dicha información se limita únicamente a la creación y uso de herramientas para detectarlos y no existe mucha información sobre como ejecutar diferentes tipos de archivos en un equipo con sistema operativo Windows 7.

### 5.0.5. Trabajo a futuro

Como trabajo a futuro sobre esta investigación se puede automatizar todo este procedimiento propuesto en un programa que añada ciertas características más complejas como el control de errores y su corrección mediante el reenvío de paquetes, debido a que puede haber el caso de que se pierda un ping, incluso en redes locales. Sin embargo hay que tener en cuenta que si se realiza una integración en un programa, diversos dispositivos IDS controlan el flujo de pings que se pueden enviar entre dos puntos durante un intervalo de tiempo bloqueando los que excedan este umbral, lo cual llevaría a una investigación complementaria a este trabajo.

Por otra parte, aunque en la Informática forense el método de ocultación por ADS es uno de los métodos que se puede detectar fácilmente con el uso de ciertas herra-

mientas; hay que tener en cuenta que, a pesar de que Microsoft tiene una política muy reservada sobre el uso de ADS para los usuarios, no ha eliminado esta característica, siendo un punto importante para analizar y realizar una investigación al respecto.

Actualmente la mayoría de las herramientas que existen para ocultar información dentro de un protocolo están orientadas a equipos con sistema operativo basado en Linux, por lo se podrían desarrollar más investigaciones sobre el tema enfocándose en la plataforma de Windows 7.

# Referencias


1. Ahsan, K. “Covert Channel Analysis and Data Hiding in TCP/IP,” Ph.D. dissertation, University of Toronto. Department of Electrical and Computer Engineering, 2002.

2. Abad Christopher. "IP checksum covert channels and selected hash collision." USA, University of California. 2001.

3. Broomfield M. “ NTFS Alternate Data Streams: focused hacking”. Network Security, 2006(8), pp. 7-9.

4. Cabuk S., Brodley C. E., and Shields, C. “IP Covert Timing Channels: Design and Detection,” in CCS ’04: Proceedings of the 11th ACM Conference on Computer and Communications Security. New York, NY, USA: ACM Press, 2004, pp. 178–187.

5. Castiglione A., De Santis A., Fiore U. and Palmieri F. "E-mail-based covert channels for asynchronous message steganography". In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on. IEEE. 2011. pp. 503-508.

6. Desimone J., Johnson D., Yuan B., and Lutz P. "Covert Channel in the BitTorrent Tracker Protocol". The 2012 International Conference on Security and Management. 2012.

7. Dima A. “A Win32-Based Technique for Finding and Hashing NTFS Alternate Data Streams”.

8. Dong P., Qian H., Lu Z., and Lan S. "A Network Covert Channel Based on Packet Classification". IJ Network Security, 14(2). 2012. pp. 109-116.

9. Ettinger J. M. “Steganalysis and game equilibria”. In Information Hiding (pp. 319-328). Springer Berlin Heidelberg. 1998.

10. Fisk G., Fisk M., Papadopoulos C., and Neil J. “Eliminating steganography in Internet traffic with active wardens”. In Information Hiding (pp. 18-35). Springer Berlin Heidelberg. 2003.

11. Gasior W., Chattanooga T. N., and Yang L. “Exploring Covert Channel in Android Platform” 2012.

12. Giffin J., Greenstadt R., Litwack P. and Tibbetts R. "Covert messaging through TCP timestamps". In Privacy Enhancing Technologies (pp. 194-208). Springer Berlin Heidelberg. 2003.

13. Goudar R. M., Patil P. N., Meshram A. G., Yewale S. M. and Fegade, A. V. "Secure Data Transmission by using Steganography". In Information and Knowledge Management Vol. 2, No. 1, 2012. pp. 1-7.

14. Gray III J. W. “Countermeasures and Tradeoffs for a Class of Covert Timing Channels,” Hong Kong University of Science and Technology, Tech. Rep., 1994.

15. Hayati P., Potdar V., and Chang E. “A survey of steganographic and steganalytic tools for the digital forensic investigator”. In Workshop of Information Hiding and Digital Watermarking to be held in conjunction with IFIPTM, Moncton, New Brunswick, Canada. 2007.

16. Hoffman, C., Johnson, D., Yuan, B., and Lutz, P. “A Covert Channel in TTL Field of DNS Packets”. The 2012 International Conference on Security and Management, 2012.

17. Huang, Y. F., Tang, S., and Yuan, J. "Steganography in inactive frames of VoIP streams encoded by source codec". Information Forensics and Security, IEEE Transactions on, 6(2). 2011. pp. 296-306.

18. Huebner, E., Bem, D., and Wee, C. K. “Data hiding in the NTFS file system”. digital investigation, 3(4). 2006. pp. 211-226.

19. Hussain, M. “A high bandwidth covert channel in network protocol”. In Information and Communication Technologies (ICICT), 2011 International Conference on.2011. pp. 1-6. IEEE.

20. James R. F. Gimbi, Daryl Johnson, Peter Lutz and Bo Yuan. "A Covert Channel Over Transport Layer Source Ports". B. Thomas Golisano College of Computing & Information Sciences. 2012.

21. Joshi Rana, Amanpreetkaur and Nitin Malik. "Network-based Steganography using Encryption in TCP/IP Header". International Journal of Computer Applications. Volume 74– No.4, July 2013. pp. 0975 – 8887.

22. Kai Z., En C., and Qinquan G. “Analysis and Implementation of NTFS File System Based on Computer Forensics”. In Education Technology and Computer Science (ETCS), 2010 Second International Workshop on (Vol. 1, pp. 325-328). 2010. IEEE.

23. Kayarkar H., and Sanyal S. “A Survey on Various Data Hiding Techniques and their Comparative Analysis”. arXiv preprint arXiv:1206.1957. 2012.

24. Ker A. D., Bas P., Böhme R., Cogranne R., Craver S., Filler T., and Pevný T. "Moving steganography and steganalysis from the laboratory into the real world". In Proceedings of the first ACM workshop on Information hiding and multimedia security. 2013. pp. 45-58. ACM.

25. Kundur D. and Ahsan, K. "Practical Internet steganography: data hiding in IP". In Proceedings of the Texas workshop on security of information systems (Vol. 2). 2003.

26. Lampson, B. W. “A Note on the Confinement Problem,” Commun. ACM, vol. 16, no. 10, 1973. pp. 613–615.

27. Lipner, S. B. “A Comment on the Confinement Problem,” in SOSP ’75: Proceedings of the fifth ACM symposium on Operating systems principles. New York, NY, USA: ACM Press, 1975. pp. 192–196.

28. Luo Q., Zhao J., and Chen M. “Research of TCP/IP protocol stack based on embedded system”. In Computer Research and Development (ICCRD), 2011 3rd International Conference on (Vol. 3, pp. 412-415). IEEE.

29. Mahant, Sameer H., and B. B. Meshram. "ADS Examiner: Tool for NTFS Alternate Data Streams Forensics Analysis." International Journal of Engineering 1.4. 2012.

30. Meadows C., and Moskowitz, I. S. “Covert Channels – A Context-Based View,” in Proceedings of the First International Workshop on Information Hiding. London, UK: Springer- Verlag, 1996, pp. 73–93.

31. Moskowitz I. S. and Kang M. H.“Covert Channels – Here to Stay?” in Compass’94: 9th Annual Conference on Computer Assurance. Gaithersburg, MD: National Institute of Standards and Technology, 1994, pp. 235–243.

32. Moskowitz I. S. and Miller A. R. “Simple Timing Channels,” in SP ’94: Proceedings of the 1994 IEEE Symposium on Security and Privacy. Oakland, CA: IEEE Computer Society, 1994, pp. 56–64.

33. Muchene D. N., Luli K., and Shue C. A. “Reporting Insider Threats via Covert Channels”. IEEE Security and Privacy Workshops. 2013.

34. Qian Y., Song H., Wang F. and Wang Z. "Network Covert Channel Encoding by Packet Length: Design and Detection". Journal of Computational Information Systems, 7(5), 2001. pp. 1463-1471.

35. Owens M. "A discussion of covert channels and steganography". SANS institute, 1, 2002. pp. 1-18.

36. Rafat K. F. and Sher M. "StegRithm: Steganographic Algorithm for Digital ASCII Text Documents". ACSIT International Journal of Engineering and Technology, Vol. 4, No. 6. 2012.

37. Rowland, Craig H. "Covert channels in the TCP/IP protocol suite." First Monday 2.5, 1997.

38. Sanders C., Valletta J., Yuan B., Johnson D., and Lutz P. “Employing Entropy in the Detection and Monitoring of Network Covert Channels”. The 2012 International Conference on Security and Management. 2012.

39. Shah M. K. and Patel S. B. "Network based packet watermarking using TCP/IP protocol suite". In Engineering (NUiCONE), 2011 Nirma University International Conference on. 2011. pp. 1-5. IEEE.

40. Simmons G. J. "The prisoners' problem and the subliminal channel". In Advances in Cryptology. 1984. pp. 51-67. Springer US.

41. Starzetz, Paul. "Ambiguities in TCP/IP-firewall bypassing." 2002.

42. Venkatraman B. R., and Newman-Wolfe R. E. “Capacity Estimation and Auditability of Network Covert Channels,” in SP ’95: Proceedings of the 1995 IEEE Symposium on Security and Privacy. Washington, DC, USA: IEEE Computer Society, 1995, p. 186.

43. Wang Z., and Lee R. B. “New Constructive Approach to Covert Channel Modeling and Channel Capacity Estimation,” in 8th Information Security Conference (ISC ’05). Berlin Heidelberg: Springer-Verlag, 2005, pp. 498–505.

44. Wendzel S. and Keller J. “Design and implementation of an active warden addressing protocol switching covert channels”. In ICIMP 2012, The Seventh International Conference on Internet Monitoring and Protection. 2012. pp. 1-6.

45. Wray, J. C. “An Analysis of Covert Timing Channels,” in IEEE Computer Society Symposium. Los Alamitos, CA, USA: IEEE Computer Society, 1991, pp. 2–7.

46. Yangwei Li, Qingni Shen, Cong Zhang, Pengfei Sun, Ying Chen and Sihan Qing. "A covert channel using Core Alteration". 26th International Conference on Advanced Information Networking and Applications Workshops. 2012.