UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

PROCEDIMIENTO DE INTEGRACIÓN DE LA ESTEGANOGRAFÍA AL PROTOCOLO HTTP

TESIS

PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

JUAN MARTIN CORONA FALCON

DIRECTOR DE TESIS:

M.C. MARIA JAQUELINA LOPEZ BARRIENTOS

México, Ciudad Universitaria, Noviembre 2015

AGRADECIMIENTOS

A mi madre, la grandiosa ama de casa que siempre ha estado a mi lado, que ha compartido alegrías y tristezas, pero sobre todo gracias porque siempre dejaste a un lado tus compromisos e hiciste lo imposible por hacerme feliz y ayudarme a conseguir muchos logros deportivos y estudiantiles, por ser madre y padre a la vez. Aunque amores yo tenga en la vida, que me llenen de felicidad, como el tuyo jamás madre mía…como el tuyo no habré de encontrar. Muchas gracias mamá.

A mi abuelo Juan Falcón, por ser mi ejemplo a seguir en la vida, por ayudarme y enseñarme tantas cosas que simplemente uno lo aprende trabajando, aquel hombre experto en el uso de la herramienta, aquel hombre que me enseñó a ser humilde y sencillo, a saber en que momento ser valiente, aquel hombre de palabras sabias que siempre tenía un refrán para alguna situación de la vida, aquel hombre que fue mi motivación para ser ingeniero y que si hoy lo soy se lo dedico solamente a él, gracias por todo abuelo y aunque ya no está a mi lado físicamente, sé que espiritualmente siempre cuento contigo.

A mi familia, mi hermano, mis primos y tíos, por siempre confiar en mí y considerarme un ejemplo a seguir. Este trabajo lo dedico a mi hermano: Luis Antonio porque siempre estuvo a mí lado y por ser un gran amigo, mis primos: Daniel, Francisco, Cesar, Luis Enrique, Janet, Jaqueline, Mario, Ana Laura, Alejandra, Mitzi, Usiel, Gustavo, Gerardo, Areli, Diana, Sonia, Miguel, María Luisa y Gerardo que siempre están conmigo tanto en mis logros y fracasos, a mis tíos: Elisa, Luisa, Fortino, Rosario, Carmen, Roberto, Sonia, Gustavo, Mria Elena, Usiel y Leticia.

Una especial dedicatoria a Paulina Falcón, mi hermana y que siempre me ha dado su apoyo en cualquier circunstancia, gracias fea.

A mis tíos Gerardo y Honorio Falcón por el apoyo que me dieron, como el que un padre le da a un hijo, nunca se los podré pagar, gracias por todo.

A mis amigos porque son pieza fundamental para que yo haya logrado esto en especial a Alejandro, Amador, Armando, Daniel Aguilera, Daniel Deschamps, Erandi, Fernando, Francisco, Imanol, Isaac, Jorge Armando, Jorge Luis, Luis Alberto, Mauricio, Nicolás, Oscar, Ricardo, Sebastián Camilo, Valeria y al final pero no menos importante Norma que me apoyo en la recta final de este trabajo y cuya presencia ha sido un parteaguas en mi vida.

Al Ing. Roberto Miranda por convertirse en un gran apoyo al final de la carrera, por siempre ser una imagen y ejemplo a seguir, aquella persona que se convirtió en un amigo y que me aconsejaba como si fuera su hijo, muchas gracias por todos los consejos y el apoyo.

Al señor Alberto Martínez quien me ha dado su apoyo incondicional, como si fuera uno de sus hijos.

A Raúl Vázquez, por darme la oportunidad de pertenecer al grupo de entrenadores del equipo Representativo de Media Superior de la UNAM, a mis compañeros Martín Alcántara y Eduardo Salcedo, gracias por sus enseñanzas.

A mi maravillosa y tan querida Universidad Nacional Autónoma de México, por permitirme desarrollarme como universitario y darme la oportunidad de concluir una licenciatura. A los representativos de Futbol Soccer y Futbol Rápido de la UNAM, por brindarme un espacio en sus equipos y permitirme vivir tantas alegrías como digno representante de esta máxima casa de estudios en competencias estatales, regionales y nacionales. Especial dedicación a mis entrenadores: Daniel Gómez, Enrique Gómez, Juan Manuel Calderón, Juan Rodríguez y Jorge Rivera, por todas sus enseñanzas y todos sus consejos dentro y fuera de la cancha de futbol.

A la Facultad de Ingeniería, por permitirme desarrollarme como estudiante, deportista y entrenador de Futbol de los equipos representativos varoniles de primera y segunda fuerza de la facultad. En especial al Lic. Miguel Figueroa por todo su apoyo escolar y deportivo.

Por último pero no menos importante a mi asesora M.C. María Jaquelina López Barrientos, por su guía, consejo y paciencia en la elaboración de este trabajo de titulación.

A todos y cada uno de ustedes muchísimas gracias.

“Es un orgullo ser de la UNAM…pero es un don de dios ser de Ingeniería”

Juan Martín Corona Falcón

ÍNDICE DE CONTENIDOS

Índice General

Índice de tablas

Índice de imágenes

INTRODUCCIÓN

INTRODUCCIÓN

La historia ha proporcionado incontables situaciones por las que la información ha atravesado por territorio hostil o enemigo para poder alcanzar su destino. La gente ha usado diferentes métodos, algunos ingeniosos, para encubrir la información y que con el paso del tiempo han ido mejorando.

Una de tantas historias narra que En la Grecia antigua, había un método en el cual se reclutaban personas con ciertas características para que fungieran como mensajeros, a quienes se les afeitaba la cabeza, una vez hecho esto se les tatuaba un mensaje secreto en la cabeza, tras lo cual le dejaban crecer el pelo a su tamaño normal. El mensajero entonces debía llevar a su destino el mensaje, tarea que se cumplía exitosamente ya que el mensajero lograba pasar cualquier control de seguridad enemigo al no percibirse en él nada sospechoso. Una vez presentado al receptor de la información, éste le afeitaba la cabeza nuevamente para leer el texto secreto. Una desventaja importante en este método era que se tenía que esperar a que el pelo creciera suficientemente para cubrir el texto antes de que el mensaje pudiera ser entregado. Otra desventaja que el mensajero quedaba marcado con el tatuaje de por vida sin poder ser destruido.

Otro método usado en la Grecia antigua eran las tabletas cubiertas de cera. El mensaje era escrito sobre la madera y posteriormente esta se cubría con cera. El receptor de la tableta debía raspar o derretir la cera para revelar el mensaje.

Durante las Segunda Guerra Mundial, las tintas invisibles fueron utilizadas para encubrir la información en notas o letras aparentemente estándares e inofensivas. Entre las fuentes más comunes para las tintas invisibles están la leche, el vinagre y la orina.

En esta misma guerra los alemanes hicieron gran uso de diversas variantes de una máquina de rotores electromecánica llamada Enigma, este fue el mayor avance del criptoanálisis en más de mil años.

Uno de los métodos más ingeniosos está desarrollado por Gaspar Schott y se detalla en su libro Schola Steganographica. El método implicaba el codificar la información emparejando letras a las notas musicales específicas sobre una hoja. Aparentemente parecería como una partitura musical normal, sin embargo si uno tocara el trozo de dicha partitura musical en un instrumento, el resultado no sería lo más agradable que esperaríamos.

Cuenta que estos pasajes de la historia permiten reflexionar acerca de la importancia y la necesidad de la humanidad por resguardar la información desde la antigüedad, y esto persiste hasta nuestros días, el hombre sigue necesitando ocultar su información sobretodo aquella que se considera sensible o confidencial y en innumerables situaciones requiere hacerla llegar a un destino específico sin siquiera dejar rastro de ello, así surge el presente trabajo de tesis cuyos objetivos se presentan a continuación.

Objetivo General

Desarrollar una metodología que permita integrar la esteganografía en el protocolo HTTP atendiendo a los requerimientos técnicos de ambas herramientas, con el fin de lograr enviar mensajes ocultos en el protocolo HTTP el cual es uno de los más utilizados en internet por todo tipo de personas aunque no tengan conocimientos informáticos.

Objetivos Particulares

1. Desarrollar un método para la insertar información (texto, audio, video, entre otros) en un objeto portador de acuerdo a los principios de la esteganografía.
2. Realizar pruebas con diversos tipos de información y con diferentes tipos de objetos portadores para ver cómo interactúan entre sí, que objeto portador puede ser más confiable y cuál es el más adecuado según la información que se desea inserta en ellos.
3. Llevar a cabo análisis comparativo de los diversos tipos de objetos portadores, cuales son los más recomendables y cuales no llegan a cumplir los objetivos de la esteganografía.
4. Documentar los resultados y sobre ellos dar recomendaciones sobre qué información es más conveniente enviar y sobre que objeto portador para no ser detectada por algún administrador de red o un dispositivo.

Definición del problema

La palabra esteganografía viene del griego 'steganós', que significa 'cubierto', y 'grafía' que significa 'escritura'. La esteganografía, en la actualidad y en términos informáticos, se refiere a información o a un archivo cualesquiera que se encuentra oculto dentro de otro, normalmente multimedia, es decir, el portador es una imagen digital, un vídeo o archivo de audio.

La Esteganografía ha aparecido ante los ojos de mucha gente a raíz de los hechos del 11 de septiembre, pero data desde tiempos antiguos donde la guerra era algo común, el hecho de enviar información importante y que los enemigos no pudieran encontrarla era algo fundamental. Así, la esteganografía en términos informáticos, se refiere a cualquier archivo con información que se encuentra oculto dentro de otro.

Pero muchas personas pueden confundir la esteganografía con la criptografía ya que las dos son métodos de ocultación de la información, por lo que se debe de aclarar cuál es la diferencia entre la criptografía y la esteganografía. Ambas son disciplinas distintas, tanto en su forma de implementar como en su objetivo mismo. Mientras que la criptografía se utiliza para cifrar información de manera que sea ininteligible para un probable intruso a pesar del conocimiento de su existencia, la esteganografía oculta la información en un portador de modo que no sea advertido el hecho mismo de su existencia y envío. De esta última forma, un probable intruso ni siquiera sabrá que se está transmitiendo información sensible.

Para este trabajo se estableció la integración de la esteganografía al protocolo de red HTTP ya que es uno de los as utilizados por todo tipo de personas aunque no cuenten con conocimientos informáticos, ya que es el protocolo que se usa para el diseño de los sitios web, por lo que es un gran ejemplo para poner en práctica los principios de la esteganografía.

Para lograr la integración de la esteganografía al protocolo HTTP, el cual como ya **se** mencionó es altamente utilizado en internet, es necesario usar la capa de aplicación, la cual es la capa superior de modelo OSI y es la que interactúa con el usuario.

Para los espías que intentan constantemente encontrar contenido oculto y/o cifrado en las transmisiones de su red, no será una tarea sencilla detectar las comunicaciones a nivel de aplicación, pues a simple vista el protocolo se comporta normalmente.

El protocolo HTTP (Hypertext Transfer Protocol) es un protocolo de nivel de aplicación que realiza la transferencia de información entre sistemas bajo el diseño cliente/servidor. La especificación completa del protocolo HTTP 1.0 está en el RFC 1945, el cual fue propuesto por Tim Berners-Lee y su uso actual es la distribución global de información, sobretodo en la World Wide Web.

En el tema de comunicaciones, el protocolo está soportado sobre los servicios de conexión prestados por los protocolos TCP/IP y se basa en sencillas operaciones entre el cliente y el servidor de solicitud y respuesta.

CAPÍTULO I

MARCO TEÓRICO

I.1 ASPECTOS TÉCNICOS DE LA ESTEGANOGRAFÍA

En este capítulo se hablará de la clasificación de la estenografía, de cómo era utilizada en la época antigua y como se usa en la actualidad. Así como su funcionamiento y cuáles son los medios de esteganografía más utilizados en la actualidad.

I. 1.1 DEFINICIÓN DE ESTEGANOGRAFÍA

Para iniciar es pertinente establecer la diferencia entre la criptografía y la esteganografía. Si bien la esteganografía suele confundirse con la criptografía, por ser ambas parte de los procesos de protección de la información, son disciplinas distintas, tanto en su forma de implementar como en su objetivo mismo. Mientras que la criptografía se utiliza para cifrar información de manera que sea ininteligible para un probable intruso a pesar del conocimiento de su existencia, la esteganografía oculta la información en un portador de modo que no sea advertido el hecho mismo de su existencia y envío. De esta última forma, un probable intruso ni siquiera sabrá que se está transmitiendo información sensible.

Por otro lado, la criptografía y la esteganografía pueden complementarse, dando un mayor nivel de seguridad a la información, es decir, es muy común (aunque no imprescindible) que el mensaje a esteganografiar sea previamente cifrado, de tal modo que a un eventual intruso no sólo le costará advertir la presencia de la mensajería oculta, y si la llegara a obtener, la encontraría cifrada.

Ya establecidos estos términos se puede decir que la esteganografía es la disciplina en la que se estudian y aplican técnicas que permiten el ocultamiento de mensajes u objetos, dentro de otros, llamados portadores, de modo que no se perciba su existencia.

El origen de esta palabra deriva de la composición de los vocablos griegos "*στεγανος" steganos*, que significa cubierto u oculto, y "*γραφος" graphos*, que significa escritura.

La esteganografía, en la actualidad y en términos informáticos, se refiere a información o a un archivo cualesquiera que se encuentra oculto dentro de otro, normalmente multimedia, es decir, el portador es una imagen digital, un vídeo o archivo de audio.

I.1.2 TIPOS DE ESTEGANOGRAFÍA

Al igual que la criptografía, la esteganografía también se divide en dos grandes grupos; la esteganografía clásica o antigua y la esteganografía moderna.

a) Esteganografía clásica

Probablemente uno de los ejemplos más antiguos del uso de la esteganografía sea el referido por Herodoto en su libro *Las historias*. En este libro describe cómo un personaje tomó un cuadernillo de dos hojas o tablillas; rayó bien la cera que las cubría y en la madera misma grabó el mensaje y lo volvió a cubrir con cera regular como se muestra en la figura 1.1. Otra historia, en el mismo libro, relata cómo otro personaje había rasurado a navaja la cabeza de su esclavo de mayor confianza, le tatuó el mensaje en el cuero cabelludo, esperó después a que le volviera a crecer el cabello y lo mandó al receptor del mensaje, con instrucciones de que le rasuraran la cabeza.[1]

Fig. 1.1 Tablilla para escribir con mensaje oculto grabado en la madera bajo la cera,
"Esteganografía, el arte de ocultar información", **Observatorio de la Seguridad de la Información.**

El científico italiano *Giovanni Battista della Porta* descubrió cómo esconder un mensaje dentro de un huevo cocido. El método consistía en preparar una tinta mezclando una onza de alumbre y vinagre, y luego se escribía en la cáscara. La solución penetra en la cáscara porosa y deja un mensaje en la superficie de la albúmina del huevo duro, que sólo se puede leer si se pela el huevo.

El abad alemán *Johannes Trithemius* escribió un libro al que tituló *Steganographia.* En él se trataban temas referentes a la ocultación de mensajes, así como métodos para conjurar a los espíritus. El libro en cuestión está hoy considerado como un libro maldito y es muy apreciado por los esoteristas del mundo entero. Aparte de este libro, también publicó *Polygraphiae Libri Sex*, un compendio de seis libros sobre criptografía que no participaba de los elementos esotéricos de su otro gran libro.

Otro ejemplo histórico más de uso de la esteganografía es el libro *Hypnerotomachia Poliphili de Francesco Colonna*, que data de 1499. En él, tomando la primera letra de sus 38 capítulos se puede leer "Poliam frater Franciscus Columna peramavit", que se traduce como "El hermano Francesco Colonna ama apasionadamente a Polia".[1]

[1] *"Esteganografía, el arte de ocultar información"*, Observatorio de la Seguridad de la Información, Instituto Nacional de las Tecnologías de la Comunicación.

Bastante más familiar para las personas resulta el ejemplo de la tinta invisible. Son muchos los niños que juegan a enviarse mensajes escritos con jugo de limón o substancias similares (con alto contenido en carbono), de tal forma que al calentar la superficie sobre la que se escribe el mensaje, éste aparece en un tono color café. Esta técnica se puede hacer más compleja si se involucran reacciones químicas.

Queda patente que la esteganografía ha estado presente desde tiempos inmemoriales y ha sido tradicionalmente empleada por las agencias militares y de inteligencia. Ahora bien, mientras la esteganografía clásica se basaba únicamente en el desconocimiento del canal encubierto bajo uso, en la era moderna se emplean canales digitales (imagen, video, audio, protocolos de comunicaciones, etc.) para alcanzar el objetivo. En muchos casos el objeto contenedor es conocido, lo que se ignora es el algoritmo de inserción de la información en dicho objeto.

b) Esteganografía moderna

La esteganografía en la actualidad se refiere a la información o a un archivo cualesquiera que se encuentra oculto dentro de otro, normalmente multimedia, es decir, el portador es una imagen digital, un vídeo o archivo de audio.

En las últimas décadas la esteganografía y en particular las técnicas de incorporación de mensajes de *copyright (Watermarking)* en documentos, imágenes y archivos de audio han experimentado un notable auge. En particular las técnicas conocidas como “marcas de agua” (*Watermarking*) para ocultar mensajes de *copyright* y la inclusión de “huellas dactilares” (*fingerprinting*), para identificar números de serie o distinguir objetos concretos entre otros similares han tenido un notable crecimiento.

Su importancia radica en que la protección de los derechos de *copyright* de imágenes, bandas sonoras y documentos escritos se ha hecho cada vez más difícil en un mundo en el que bajar de la web una imagen o un archivo MP3 está a tan sólo un clic de distancia.

En la figura 1.2 se observa cómo se ingresa mediante sistema hexadecimal un mensaje en la imagen.

Fig. 1.2 Ejemplo de esteganografía moderna en imágenes

I.1.3 FUNCIONAMIENTO DE LA ESTEGANOGRFÍA

Para esconder un mensaje mediante esteganografía, en primer lugar se escoge un fichero cualquiera, un documento Word o PDF, un fichero de imagen BMP o uno de sonido WAV o MP3, y se escoge el mensaje que se quiere ocultar, un mensaje de texto u otro fichero. El programa que implementará la esteganografía modifica el portador de varias formas posibles: alterando los valores de algunos de los puntos de la imagen, sumándoles o restándoles uno (+1 para indicar el bit 1, por ejemplo, y -1 para indicar el bit 0), de forma que sea imperceptible al usuario, pero que alguien que sepa que en esa imagen hay un mensaje, pueda recuperarlo. Existen otros métodos para ocultar información que serán estudiados más adelante. La figura 1.3 muestra cómo funciona a grandes rasgos la esteganografía:

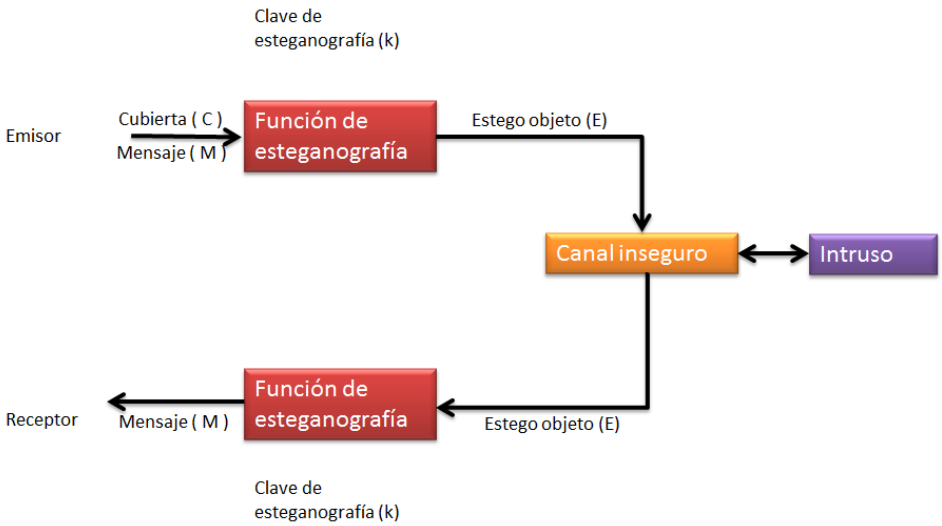

Fig. 1.3 Funcionamiento del algoritmo de esteganografía, Instituto Juan Velázquez de Velasco de Investigación en Inteligencia para l seguridad y la Defensa, Universidad Carlos III de Madrid

Descripción de las partes del ciclo de un mensaje usando esteganografía.

- f_E = Función de esteganografía.
- f_E^{-1}= Función Inversa de esteganografía.

- Cubierta(C) = Objeto donde embeber el mensaje (foto, audio o video).
- Mensaje (M) = Mensaje incrustado.
- Clave de esteganografía (K) = Clave de esteganografía.
- Estego objeto = Objeto con el mensaje embebido.
- Canal inseguro.
- Emisor.
- Receptor.
- Guardián o Intruso.

El emisor envía un mensaje oculto (M) "incrustado", en un mensaje de apariencia inocua (C) que servirá de cubierta. A esto se le aplica una función de esteganografía $f = (E)$, que con una clave asociada (K) permite extraer el mensaje (necesaria aun si se conoce el algoritmo esteganográfico). Posteriormente, el mensaje oculto y la cubierta forman el objeto de la esteganografía (E), que puede enviarse por un canal inseguro y ser visto sin problemas por el guardián o intruso. Finalmente, el receptor recibe el objeto compuesto y, aplicando la función inversa $f\ (O) = f(E)-_1$, puede recuperar el mensaje oculto.

Se pueden observar distintos actores implicados en el campo de la esteganografía:

- Mensaje oculto: Mensaje a enviar
- Objeto encubridor: Objeto en el que el mensaje oculto será insertado
- Estego-objeto: Objeto encubridor contenido en el mensaje oculto.
- Guardián: Alguien que monitoriza la comunicación
    - Pasivo: Sólo lectura.
    - Activo: Puede efectuar modificaciones ligeras.
    - Malicioso: Puede hacer cualquier cosa (No es realista en muchas situaciones).
- Estegoanálisis: Ciencia que estudia la detección (ataques pasivos) y/o anulación (ataques activos) de información oculta en distintos archivos portadores, así como la posibilidad de localizar información útil dentro de ellos mismos.

El algoritmo esteganográfico debe contar con las siguientes características:

- Capacidad: Cantidad de información que puede ser ocultada.
- Seguridad: Dificultad para un tercero de detectar información oculta.
- Robustez: Cantidad de modificaciones que el medio puede soportar antes de que se pierda la información oculta.

I.2 MODELO TCP/IP Y OSI

El modelo TCP/IP es un modelo de descripción de protocolos de red creado en la década de 1970 por DARPA, una agencia del Departamento de Defensa de los Estados Unidos. El modelo TCP/IP, describe un conjunto de guías generales de diseño e implementación de protocolos de red específicos para permitir que un equipo pueda comunicarse en una red. TCP/IP provee conectividad de extremo a extremo especificando como los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario. Existen protocolos para los diferentes tipos de servicios de comunicación entre equipos.

TCP/IP tiene cuatro capas de abstracción según se define en el RFC 1122 y se muestra en la figura 1.4. Esta arquitectura de capas a menudo es comparada con el Modelo OSI de siete capas.

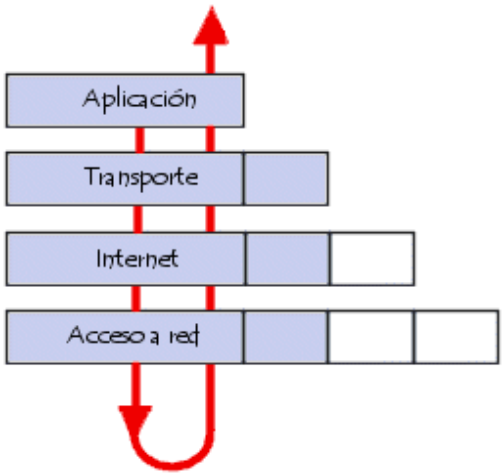

Fig. 1.4 Modelo TCP/IP

El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (Open System Interconnection) es el modelo de red descriptivo creado por la Organización Internacional para la Estandarización ISO en el año 1984. Es decir, es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

Este modelo está dividido en siete capas (véase la figura 1.5).

Fig. 1.5 Modelo OSI

I.2.1 CAPA DE APLICACIÓN

Se hace un análisis de la funcionalidad de la capa de aplicación de manera particular, dado que es la capa en la que actúa el protocolo HTTP que se utiliza en el presente proyecto de tesis. Esta capa ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y protocolos de transferencia de archivos (FTP).

Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente. Así por ejemplo un usuario no manda una petición «GET /index.html HTTP/1.0» para conseguir una página en html, ni lee directamente el código html/xml. O cuando se utiliza programas para conversar por internet como el Messenger, no es necesario que codifiquemos la información y los datos del destinatario para entregarla a la capa de Presentación (capa 6) para que realice el envío del paquete. Esto se observa en la figura 1.6.

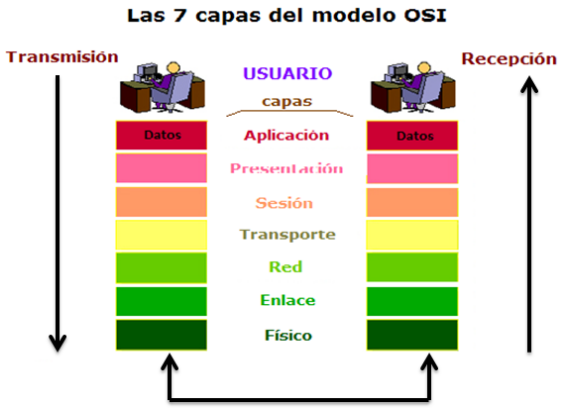

Fig. 1.6 Capa de aplicación del modelo OSI

I.3 TÉCNICAS ESTEGANOGRÁFICAS MÁS UTILIZADAS SEGÚN EL TIPO DE MEDIO

La esteganografía puede ser usada en prácticamente casi cualquier tipo de fichero sin embargo en el presente proyecto de tesis se abordan los que se consideran los medios más comunes.

a) En documentos

El uso de esteganografía en los documentos puede funcionar con sólo añadir un espacio en blanco y las fichas a los extremos de las líneas de un documento como se muestra en la figura 1.7. Este tipo de esteganografía es extremadamente eficaz, ya que el uso de los espacios en blanco y tabs no es visible para el ojo humano, al menos en la mayoría de los editores de texto, y se producen de forma natural en los documentos, por lo que en general es muy difícil que levante sospechas.

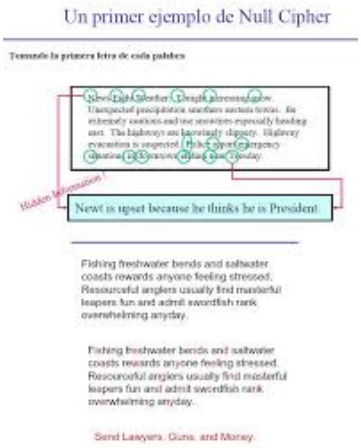

Fig. 1.7 Esteganografía en documentos

b) En imágenes

El método más utilizado es el LSB, puesto que para una computadora un archivo de imagen es simplemente un archivo que muestra diferentes colores e intensidades de luz en diferentes áreas (pixels). El formato de imagen más apropiado para ocultar información es el BMP color de 24 bit Bitmap, debido a que es el de mayor proporción (imagen no comprimida) y normalmente es de la más alta calidad.

Eventualmente se prefiere optar por formatos BMP de 8 bits o bien otros tales como el GIF, por ser de menor tamaño por lo que se debe tomar en cuenta si se transportan imágenes de gran tamaño por internet ya que en ese caso es posible levantar sospechas.

Es importante recalcar que si se oculta información dentro de un archivo de imagen y éste es convertido a otro formato, lo más probable es que la información oculta dentro sea dañada y, consecuentemente, resulte irrecuperable.

Otro método utilizado es el de máscara. La compresión de una imagen puede, en ocasiones, tener efectos en la integridad final del mensaje oculto. En la figura 1.8 se muestra la técnica LSB. Existen dos tipos de compresiones:

- Lossy, usado por el famoso formato JPEG.
- Lossless, usados por los formatos BMP y GIF.

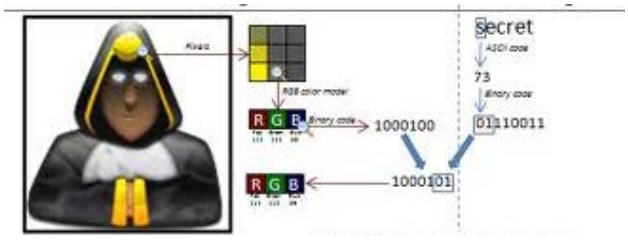

Fig. 1.8 Esteganografía en imágenes

c) En audio

El odio humano es extremadamente sensible al cambio en los patrones de audio, pero no tanto como para percibir cambios dentro de una misma frecuencia. Al ocultar un mensaje en audio, es importante saber el medio por el que se va a transmitir el mensaje, ya que no es lo mismo entre medios digitales (entre ordenadores) o transmitir a través del medio ambiente (bocinas). Cuando se quiere ocultar información sensible dentro de un fichero de sonido, se suelen utilizar las siguientes cuatro técnicas:[2]

[2] Esteganografía, Universidad Rey Juan Carlos Móstoles, España, Álvaro Navarro Clemente, (2005), pp 6

- **Codificación Low-Bit**. El mensaje puede ser almacenado en ficheros de sonido de la misma manera que la tenca LSB hace con las imágenes. El problema con el low bit encoding es que en general es
- perceptible para el oído humano ya que se escucha la diferencia de sonido, por lo que es más bien un método arriesgado que alguien lo use si se está tratando de ocultar información dentro de un archivo de audio.[2]
- **Spread Spectrum**. Es el método de ocultar un mensaje de baja señal dentro de otro de señal mayor. Este método añade ruido aleatorio para completar perfectamente la ocultación final y la persona que escuche el mensaje asocia que el ruido es parte de la grabación y no de que fue agregado intencionalmente para ocultar algún mensaje.[2]
- **Echo Data Hiding**. Este método usa el eco de un fichero de sonido para ocultar en él la información secreta. Lo que este método consigue mejor que otros es que puede mejorar realmente el sonido del audio dentro de un archivo de audio.[2]
- **Máscara perceptual**. Este método usa el concepto de ocultar un sonido tras otro de la misma frecuencia, por lo que es imperceptible al odio humano por la similitud de frecuencias.[2]

Las tres últimas técnicas son las más recomendadas para colocar un mensaje en un archivo de audio ya que las modificaciones al archivo de audio original no son percibidas por el oído humano.

d) En vídeo

En vídeo, suele utilizarse el método DCT (*Discrete Cosine Transform*). DCT funciona cambiando ligeramente cada una de las imágenes en el vídeo, buscando que no sea perceptible por el ojo humano. Para ser más precisos acerca de cómo funciona, DCT altera los valores de ciertas partes de las imágenes, por lo general las redondea. Por ejemplo, si parte de una imagen tiene un valor de 6,667, lo aproxima hasta 7.

La esteganografía en vídeo es similar a la aplicada en las imágenes, además de que la información está oculta en cada fotograma de vídeo. Cuando sólo una pequeña cantidad de información que está oculta dentro del código fuente por lo general no es perceptible a todos. Sin embargo, cuanta mayor información se oculte, más perceptible será. En la figura 1.9 se da un ejemplo de cómo sería la esteganografía en video.

Fig. 1.9 Esteganografía en video

[2] Esteganografía, Universidad Rey Juan Carlos Móstoles, España, Álvaro Navarro Clemente, (2005), pp 6

e) En red

Se han escrito muchos artículos acerca de la ocultación de datos en imágenes y fotografías. También resulta bastante conocida la posibilidad de guardar información en el ruido de fondo de una canción, de manera imperceptible al oído humano, o en el de una secuencia de vídeo. La que resulta menos conocida es la denominada esteganografía de red, la cual utiliza determinadas características de los protocolos de red para encapsular datos y transmitirlos camuflados por Internet.

La esteganografía de red se basa en tres métodos principales los cuales se detallaran en el capítulo siguiente:

- **Encapsulación de un protocolo en otro**.
- **Encapsulación de información en el campo de datos de un protocolo.**
- **Encapsulación de información en un campo numérico de un protocolo.**

I.3.1 MÉTODOS DE TRANSFORMACIÓN

Existen numerosos métodos y algoritmos utilizados para ocultar la información dentro de archivos multimedia: imágenes, audio y vídeo. A continuación se indican algunos de los más usados.

a) Enmascaramiento y filtrado

En este caso la información se oculta dentro de una imagen digital empleando marcas de agua que incluyen información, como el derecho de autor, la propiedad o licencias. De este modo se amplía la cantidad de información presentada.

b) Algoritmos y transformaciones

Esta técnica oculta datos basados en funciones matemáticas que se utilizan a menudo en algoritmos de la compresión de datos. La idea de este método es ocultar el mensaje en los bits de datos menos importantes.

c) Inserción en el bit menos significativo

Este es el método moderno más común y popular usado para esteganografía. Este método consiste en hacer uso del bit menos significativo de los pixels de una imagen y alterarlo, esta técnica se puede emplear también en audio y video, aunque no es lo más común. Así, la distorsión de la imagen en general se mantiene al mínimo (la perceptibilidad es prácticamente nula), mientras que el mensaje es esparcido a lo largo de sus píxeles, esta técnica es mucho más efectiva cuando el archivo de imagen es grande, posee fuertes variaciones de color ("imagen ruidosa") y también aventaja cuanto mayor sea la profundidad de color. Asimismo esta

técnica puede utilizarse eficazmente en imágenes a escala de gris, pero no es apropiada para aquellas en color de 8 bit paletizadas (misma estructura que las de escalas de gris, pero con paleta en color).

En general, los mejores resultados se obtienen en imágenes con formato de color RGB (tres bytes, componentes de color, por píxel).

Ejemplo:

Se tiene un archivo con los siguientes bytes:

132 134

Pues ahora se obtiene el binario:

10000100 10000110

Si se desea ocultar el número 1 (01) en el código anterior solamente se tiene que modificar un número.

10000100 10000111 = 132 135

El método del LSB funciona mejor en los archivos de imágenes que tienen una alta resolución y usan gran cantidad de colores, en el caso de archivos de audio, funciona mejor en ficheros que tengan ruido, es decir, que tengan diferentes cambios de frecuencias. Por tanto, cuanto más ruido tengo al fichero más difícil será que una persona sea consciente de la manipulación realizada.

Además, este método no altera en absoluto el tamaño del archivo portador o cubierta (por eso se le conoce como "una técnica de sustitución"). Posee la desventaja de que el tamaño del archivo portador debe ser mayor cuanto más grande sea el mensaje a embeber; se necesitan 8 bytes de imagen por cada byte de mensaje a ocultar; es decir, la capacidad máxima de una imagen para almacenar un mensaje oculto es de ~~su~~ 12,5%. Si se pretende emplear una mayor porción de bits de la imagen (por ejemplo, no sólo el último, sino los dos últimos), puede comenzar a ser perceptible al ojo humano la alteración general provocada.

d) Método de Sustitución

Cada fichero que es creado contiene áreas de datos no usadas o que no son importantes, éstas se pueden remplazar sin aparentes cambios visuales o estructurales del fichero original. Esto permite esconder información sensible dentro del fichero y tener la certeza de que el fichero original no ha sufrido ninguna mutación. El método anterior del bit menos significativo (LSB) sustituye el último bit de cada byte, de tal forma que se repite este proceso con cada byte sin que el ojo u oído humano aprecie diferencia alguna.

El método de sustitución no incrementa el tamaño del archivo portador sin embargo se debe tomar en cuanto el tamaño del mensaje que se desea ocultar. Así pues, este método es utilizado debido a su rapidez y facilidad de uso.

I.4 ESTEGANOGRAFÍA AVANZADA

La esteganografía es un arte complejo y con muchos matices. Sin llegar aún a la combinación de esteganografía y criptografía, es posible el uso de determinadas técnicas avanzadas que permiten aumentar la eficacia de una información oculta mediante esteganografía. Algunas son:

a) Uso de múltiples claves

Esta técnica es heredada directamente de la criptografía, pero con distinta forma de aplicación. Consiste en usar distintas codificaciones para cada porción arbitraria del mensaje a ocultar. Así, una frase de cinco palabras puede tener una clave de codificación para cada una de las palabras, por ejemplo: en la primera realizar una operación de sustracción en una unidad en los ceros y de adición en una unidad en los unos; en la segunda se llevan cabo las mismas operaciones pero invirtiendo el orden de los bits; y en una tercera aplicar la función XOR de los bits.

Cabe mencionar varios aspectos, entre los cuales es importante tener presente que la clave ha de ser conocida por el destinatario y que se pueden usar tantas claves como se juzgue conveniente.

b) Esteganografía en capas

Mediante esteganografía en capas establece una relación lineal entre los elementos ocultos. Así, la codificación de la segunda palabra o letra de un mensaje depende de la primera (puede depender del último valor de la cifra, del último valor modificado, o de la posición entre otros). Así, se establece un orden estricto de decodificación que impide obtener completamente el mensaje sin la primera parte, con lo cual únicamente se debe comunicar la clave para obtener esta parte y la pauta a seguir para encadenar los fragmentos.

c) Adición de ruido

Aunque en un mensaje esteganografiado todo el fichero es considerado ruido, se puede añadir ruido en el proceso de esteganografiado. Así, además de modificar los bits necesarios para inyectar el mensaje, es posible modificar unos cuantos bits aleatorios del mensaje de forma que aun teniendo el fichero original, un posible atacante deba conocer el sistema de codificación usado.

d) Uso de distintas magnitudes

Aunque lo habitual es variar en 1 bit el byte del mensaje original, nada impide variarlo en más bits. Así, se pueden establecer claves complejas, como por ejemplo: ocultar una frase de cinco palabras, y al esconder la primera de las palabras se suma 1 bit en la codificación de la primera letra, 2 bits en la codificación de la segunda, 3 bits en la tercera, etcétera, hasta que vuelva a aparecer una modificación de 1 bit, que significará el inicio de otra palabra.

Mientras se manejen ficheros que usen mucha información (imágenes de 24 bits o más por ejemplo) no se notará que varía la escala en 1 o 10 unidades, y proporciona un tipo de clave más compleja.

I.5 ESTEGOANÁLISIS

Lo que la esteganografía esencialmente hace es explotar las limitaciones de la percepción humana, ya que los sentidos humanos no están capacitados para buscar archivos que tienen información escondida dentro de ellos, aunque hay programas disponibles que pueden hacer lo que se llama esteganálisis *(Steganalysis).* Debido a que la esteganografía es invasiva, es decir, deja huellas en el medio de transporte utilizado, las técnicas de esteganálisis se basan en cómo detectar estos cambios.

I.5.1 TIPOS DE ESTEGOANÁLISIS

a) Estegoanálisis manual

Como su nombre lo dice, consiste en buscar de forma manual diferencias entre el fichero original y el fichero esteganografiado buscando cambios en la estructura para localizar datos ocultos. Los principales inconvenientes de esta técnica son:

- Se necesita tener un fichero original.
- Es una técnica que solo puede detectar el fichero esteganografiado pero es casi imposible descifrar el mensaje.

b) Estegoanálisis estadístico

Consiste en el cotejo de la frecuencia de distribución de colores en el caso de un fichero de imagen esteganografiado. Es una técnica lenta para la que se deben emplear software especializado. Aunque estos programas suelen buscar pautas para ocultar los mensajes que utilizan los programas más habituales de esteganografía, lo que los hace muy eficaces cuando se trata de mensajes ocultos con programas. Los mensajes que han sido ocultados manualmente son casi imposibles de encontrar para estos programas.

El estegoanálisis en casos muy concretos se convierte en una tarea muy trivial pero en general es un tema muy complejo y de difícil aplicación principalmente por el gran número de medios de información existentes y también por las diversas técnicas de esteganografía en el medio.

c) Ataque activo

Este tipo de ataques implica el destruir el mensaje oculto. Es muy frecuente en tecnologías con marcas de agua digitales donde el principal objetivo es inutilizar dichas marcas. Los ataques activos también son útiles en situaciones donde se sospecha que existe esteganografía pero que el mensaje oculto no es importante. Un buen ejemplo son las imágenes donde se puede aplicar algún efecto digital sin que lo perciba el ojo humano pero que modificará el mensaje oculto que está embebido en la imagen dejándolo totalmente irrecuperable.[2]

d) Ataque pasivo

Un ataque pasivo implica la detección del uso de esteganografía y es una forma de descifrar los mensajes ocultos. Los tipos de ataque incluyen:[2]

- Visión del fichero.
- Escuchas del fichero.
- Ejecución de comparaciones en un fichero (si se tiene el fichero original).
- Ataques estadísticos. Éstos implican la detección de cambios en los patrones de los pixeles de los Bits Menos Significativos (LSB).
- Firma.

Obviamente los dos primeros métodos del análisis no devolverán resultados exactos. El propósito de la esteganografía es que los cambios estén ocultos. Por lo tanto simplemente viendo o escuchando el archivo no significa revelar el mensaje secreto. Los primeros cuatro métodos implican el realizar comparaciones contra el archivo original (esto puede indicar a menudo que un archivo es portador del mensaje oculto y por lo tanto ser acertado).[2]

Si en la esteganografía se utiliza el método *The Right Way*, es porque el atacante no tiene acceso al archivo original sin modificar. Si una persona quiere utilizar esteganografía para ocultar un mensaje secreto, sería absurdo utilizar un archivo bien conocido o fácilmente disponible para encubrir el mensaje dentro. El sentido común dice que lo mejor es utilizar un archivo que nunca antes haya sido visto por cualquier persona o por lo menos, que haya sido elegido de una localización poco conocida dentro de Internet.[2]

[2] Esteganografía, Universidad Rey Juan Carlos Móstoles, España, Álvaro Navarro Clemente, (2005), pp 6,7

I.6 ASPECTOS TÉCNICOS DEL PROTOCOLO HTTP

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. Fue propuesto por Tim Berners-Lee, Estableció la primera comunicación entre un cliente y un servidor usando el protocolo HTTP en noviembre de 1989. En octubre de 1994 fundó el Consorcio de la World Wide Web con sede en el MIT, para supervisar y estandarizar el desarrollo de la tecnología sobre las que se fundamenta la Web y que permiten el funcionamiento de Internet.[3]

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL. La comunicación entre el navegador y el servidor se muestra en la figura 1.10.

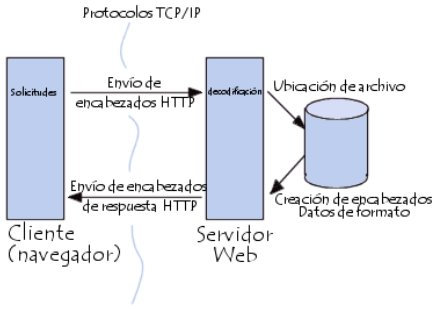

Fig. 1.10 Esquema de la comunicación entre navegador y servidor

- El navegador realiza una solicitud HTTP
- El servidor procesa la solicitud y después envía una respuesta HTTP

[3] Análisis de vulnerabilidades esteganográficas en protocolos de comunicación IP y HTTP, Tesis de Ingeniería en Informática, Universidad de Buenos Aires, Argentina, Pablo Andres Deymonnaz, (2012) pp 60

I.6.1 Etapas de una transacción HTTP.

Para profundizar más en el funcionamiento de HTTP, se verá primero un caso particular de una transacción HTTP; en los siguientes apartados se analizarán las diferentes partes de este proceso.

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
- El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD, entre otros), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (regularmente HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor.
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
- Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

a) **Solicitud HTTP**

Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor. Incluye:

Una línea de solicitud: es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:

- El método
- La dirección URL
- La versión del protocolo utilizada por el cliente (por lo general, HTTP/1.0).

Los campos del encabezado de solicitud es un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

El cuerpo de la solicitud: es un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

A continuación se encuentra un ejemplo de una solicitud HTTP:[4]

- GET http://es.kioskea.net HTTP/1.0
- Host : www.educacion.edu
- Connection : close
- User-Agent : Mozilla/4.0
- Acept language: fr

b) Comandos

Los comandos son instrucciones u órdenes que el usuario proporciona, desde la línea de comandos o desde una llamada de programación.

Suele admitir parámetros o argumentos de entrada, lo que permite modificar su comportamiento predeterminado. Suelen indicarse tras una barra "/" (en sistemas operativos DOS) o un guion simple "-" o doble "--" (en sistemas operativos Unix).

La tabla 1.1 muestra los comandos que se utilizan para el protocolo HTTP.

Tabla 1.1 Comandos del protocolo HTTP

Comando	*Descripción*
GET	*Solicita el recurso ubicado en la URL especificada*
HEAD	*Solicita el encabezado del recurso ubicado en la URL especificada*
POST	*Envía datos al programa ubicado en la URL especificada*
PUT	*Envía datos a la URL especificada*
DELETE	*Borra el recurso ubicado en la URL especificada*

[4] Redes y comunicaciones, Zamora, David Rodríguez Hernández (2007) pp 6

a) Encabezados

Dentro de un mensaje HTTP los encabezados son muy importantes. Definen en gran parte la información que se intercambia entre clientes y servidores dándole flexibilidad al protocolo. Estas líneas permiten que se envíe información descriptiva en la propia transacción, permitiendo cosas como la autenticación o identificación de usuarios. La sintaxis de una línea simple de encabezado es la siguiente:

Nombre-de-Encabezado: Valor

La tabla 1.2 muestra los encabezados que se utilizan para el protocolo HTTP.

Tabla 1.2 Encabezados del protocolo HTTP

Nombre del encabezado	Descripción
Accept	Tipo de contenido aceptado por el navegador (por ejemplo, texto/html).
Accept-Charset	Juego de caracteres que el navegador espera
Accept-Encoding	Codificación de datos que el navegador acepta
Accept-Language	Idioma que el navegador espera (de forma predeterminada, inglés)
Authorization	Identificación del navegador en el servidor
Content-Encoding	Tipo de codificación para el cuerpo de la solicitud
Content-Language	Tipo de idioma en el cuerpo de la solicitud
Content-Length	Extensión del cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud (por ejemplo, texto/html).
Date	Fecha en que comienza la transferencia de datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
From	Permite especificar la dirección de correo electrónico del cliente

From	Permite especificar que debe enviarse el documento si ha sido modificado desde una fecha en particular
Link	Vínculo entre dos direcciones URL
Orig-URL	Dirección URL donde se originó la solicitud
Referer	Dirección URL desde la cual se realizó la solicitud
User-Agent	Cadena con información sobre el cliente, por ejemplo, el nombre y la versión del navegador y el sistema operativo

c) Respuesta HTTP

Una respuesta HTTP es un conjunto de líneas que el servidor envía al navegador. Está constituida por:

- Una línea de estado: es una línea que especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código. La línea está compuesta por tres elementos que deben estar separados por un espacio: La línea está formada por tres elementos que deben estar separados por un espacio:
    - la versión del protocolo utilizada
    - el código de estado
    - el significado del código
- Los campos del encabezado de respuesta: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos (:) y por el valor del encabezado Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.
- El cuerpo de la respuesta: contiene el documento solicitado.

A continuación se encuentra un ejemplo de una respuesta HTTP:[4]

- HTTP/1.1 200 ok
- Connection: close
- Date: Sat, 15 Jan 2000 14:37:12 GMT
- Server : Microsoft-IIS/2.0
- Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT

[4] Redes y comunicaciones, Zamora, David Rodríguez Hernández (2007) pp 6

- Content-Length : 1245
- Content-Type : text/HTML

La tabla 1.3 muestra los encabezados de respuesta usados por el protocolo HTTP.

TABLA 1.3 Encabezados de respuesta del protocolo HTTP

Nombre del encabezado	Descripción
Content-Encoding	Tipo de codificación para el cuerpo de la respuesta
Content-Language	Tipo de idioma en el cuerpo de la respuesta
Content-Length	Extensión del cuerpo de la respuesta
Content-Type	Tipo de contenido del cuerpo de la respuesta (por ejemplo, texto/html). Consulte Tipos de MIME
Date	Fecha en que comienza la transferencia de datos
Expires	Fecha límite de uso de los datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
Location	Redireccionamiento a una nueva dirección URL asociada con el documento
Server	Características del servidor que envió la respuesta

I.7 VULNERBILIDAD DE LOS SERVICIOS EN LA WEB

El protocolo HTTP (o HTTPS) representa el estándar que posibilita la transferencia de páginas Web a través de un sistema de solicitud y respuesta. Internet, que se utiliza principalmente para transferir páginas Web estáticas, se ha convertido rápidamente en una herramienta interactiva que permite proporcionar servicios en línea. El término "aplicación Web" se refiere a cualquier aplicación a cuya interfaz se pueda acceder en la Web desde un simple navegador. Hoy en día, el protocolo HTTP, la base para una determinada cantidad de tecnologías (como por ejemplo: SOAP, Javascript, XML-RPC), juega un indudable papel estratégico en la seguridad de sistemas de información.

Como tal, la seguridad de los servicios de Internet debe tenerse en cuenta al momento del diseño y desarrollo. En la figura 1.11 se observa el esquema de la seguridad en la red.

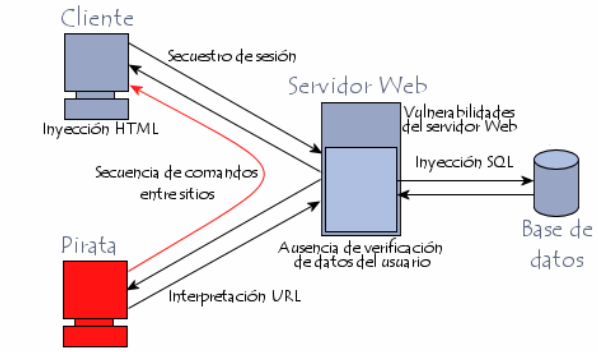

Fig 1.11 Tipos de vulnerabilidades

Las vulnerabilidades de aplicaciones Web se pueden clasificar de la siguiente manera:

- Vulnerabilidades del servidor Web. Este tipo es cada vez más atípico ya que la mayoría de los desarrolladores de servidores Web han aumentado su seguridad con los años.
- Manipulación de URL, incluida la modificación manual de parámetros de URL para modificar el comportamiento esperado del servidor Web.
- Aprovechamiento de las debilidades de los identificadores de sesión y sistemas de autenticación.
- Inyección de código HTML y Secuencia de comandos entre sitios.
- Inyección de comandos SQL.

El protocolo HTTP se utiliza por naturaleza para administrar las solicitudes, es decir, para recibir los datos de entrada y enviar los datos de retorno. Los datos se pueden enviar de varias maneras:

- La URL de la página Web
- En encabezados HTTP
- En el cuerpo de la solicitud (solicitud POST)
- A través de una cookie

En general, la idea básica a tener en cuenta durante el proceso de desarrollo es que nunca se debe confiar en los datos enviados por el cliente.

Casi todas las vulnerabilidades de los servicios Web están vinculadas a la negligencia por parte de los diseñadores, quienes no han verificado el formato de los datos ingresados por los usuarios.

Los ataques a las aplicaciones Web siempre son dañinos ya que proporcionan una mala imagen a la empresa. Un ataque exitoso puede provocar cualquiera de las siguientes consecuencias:

- Desfiguración de la página Web;
- Robo de información;
- Modificación de datos, y en particular la modificación de datos personales de los usuarios;
- Intrusión en el servidor Web.

I.8 NORMATIVIDAD

En una organización la gestión de seguridad puede tornarse compleja y difícil de realizar, esto no por razones técnicas, más bien por razones organizativas, coordinar todos los esfuerzos encaminados para asegurar un entorno informático institucional, mediante la simple administración del recurso humano y tecnológico, sin un adecuado control que integre los esfuerzos y conocimiento humano con las técnicas depuradas de mecanismos automatizados, tornará en la mayoría de los casos el ambiente en uno inimaginablemente hostil, para ello es necesario emplear mecanismos reguladores de las funciones y actividades desarrolladas por cada los empleados de la institución. Las normas y políticas de seguridad, integran estos esfuerzos de una manera conjunta. Pretenden, ser el medio de comunicación en el cual se establecen las reglas, normas, controles y procedimientos que regulen la forma en que la institución, prevenga, proteja y maneje los riesgos de seguridad en diversas circunstancias. Estas normas y políticas sirven de referencia, en ningún momento pretenden ser normas absolutas, las mismas están sujetas a cambios realizables en cualquier momento, siempre y cuando se tengan presentes los objetivos de seguridad. Toda persona que utilice los servicios que ofrece la red, deberá conocer y aceptar el reglamento vigente sobre su uso, el desconocimiento del mismo, no exonera de responsabilidad al usuario, ante cualquier eventualidad que involucre la seguridad de la información o de la red institucional.

a) Seguridad Organizacional

Se establece el marco formal de seguridad que debe sustentar la institución, incluyendo servicios o contrataciones externas a la infraestructura de seguridad, integrando el recurso humano con la tecnología, denotando responsabilidades y actividades complementarias como respuesta ante situaciones anómalas a la seguridad.

b) Seguridad Lógica

Trata de establecer e integrar los mecanismos y procedimientos, que permitan monitorear el acceso a los activos de información, que incluyen los procedimientos de administración de usuarios, definición de responsabilidades, perfiles de seguridad, control de acceso a las aplicaciones y documentación sobre sistemas, que van desde el control de cambios en la configuración de los equipos, manejo de incidentes, selección y aceptación de sistemas, hasta el control de software malicioso.

c) Seguridad Física

Identifica los límites mínimos que se deben cumplir en cuanto a perímetros de seguridad, de forma que se puedan establecer controles en el manejo de equipos, transferencia de información y control de los accesos a las distintas áreas con base en la importancia de los activos.

d) Seguridad Legal

Integra los requerimientos de seguridad que deben cumplir todos los empleados, socios y usuarios de la red institucional bajo la reglamentación de la normativa interna de políticas y manuales de procedimientos en cuanto al recurso humano, sanciones aplicables ante faltas cometidas, así como cuestiones relacionadas con la legislación del país y contrataciones externas.

I.9 SEGURIDAD DE LA INFORMACIÓN

La seguridad de la información es el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permitan resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de la misma.

La seguridad de la información no debe ser confundido con el de seguridad informática, ya que esta última sólo se encarga de la seguridad en el medio informático, pero la información puede encontrarse en diferentes medios o formas, y no solo en medios informáticos.

El campo de la seguridad de la información ha crecido y evolucionado considerablemente a partir de la Segunda Guerra Mundial, convirtiéndose en una carrera acreditada a nivel mundial. Este campo ofrece muchas áreas de especialización, incluidos la auditoría de sistemas de información, planificación de la continuidad del negocio, ciencia forense digital y administración de sistemas de gestión de seguridad, entre otros.

Concepción de la seguridad de la información

La información se clasifica como:

Crítica: Es indispensable para la operación de la empresa.

Valiosa: Es un activo de la empresa y muy valioso.

Sensible: Debe de ser conocida por las personas autorizadas

Existen dos palabras muy importantes que son riesgo y seguridad:

Riesgo: Es la materialización de vulnerabilidades identificadas, asociadas con su probabilidad de ocurrencia, amenazas expuestas, así como el impacto negativo que ocasione a las operaciones de negocio.

Seguridad: Es una forma de protección contra los riesgos.

La seguridad de la información comprende diversos aspectos entre ellos la disponibilidad, comunicación, identificación de problemas, análisis de riesgos, la integridad, confidencialidad, recuperación de los riesgos.

Precisamente la reducción o eliminación de riesgos asociado a una cierta información es el objeto de la seguridad de la información y la seguridad informática. Más concretamente, la seguridad de la información tiene como objeto los sistemas el acceso, uso, divulgación, interrupción o destrucción no autorizada de información. Los términos seguridad de la información, seguridad informática y garantía de la información son usados frecuentemente como sinónimos porque todos ellos persiguen una misma finalidad al proteger la confidencialidad, integridad y disponibilidad de la información.

La seguridad de la información involucra la implementación de estrategias que cubran los procesos en donde la información es el activo primordial. Estas estrategias deben tener como punto primordial el establecimiento de políticas, controles de seguridad, tecnologías y procedimientos para detectar amenazas que puedan explotar vulnerabilidades y que pongan en riesgo dicho activo, es decir, que ayuden a proteger y salvaguardar tanto información como los sistemas que la almacenan y administran. La seguridad de la información incumbe a gobiernos, entidades militares, instituciones financieras, los hospitales y las empresas privadas con información confidencial sobre sus empleados, clientes, productos, investigación y su situación financiera.

La Seguridad de la Información ha declarado que la confidencialidad, integridad y disponibilidad (conocida como la Tríada CIA, del inglés: "Confidentiality, Integrity, Availability") son los principios básicos de la seguridad de la información.

La correcta Gestión de la Seguridad de la Información busca establecer y mantener programas, controles y políticas, que tengan como finalidad conservar la confidencialidad, integridad y disponibilidad de la información, si alguna de estas características falla no estamos ante nada seguro.

a) Confidencialidad

Es la propiedad de prevenir la divulgación de información a personas o sistemas no autorizados. A groso modo, la confidencialidad es el acceso a la información únicamente por personas que cuenten con la debida autorización.

La pérdida de la confidencialidad de la información puede adoptar muchas formas. Cuando alguien mira por encima de su hombro, mientras usted tiene información confidencial en la pantalla, cuando se publica información privada, cuando un laptop con información sensible sobre una empresa es robado, cuando se divulga información confidencial a través del teléfono, etc. Todos estos casos pueden constituir una violación de la confidencialidad

b) Integridad

Es la propiedad que busca mantener los datos libres de modificaciones no autorizadas. A groso modo, la integridad es el mantener con exactitud la información tal cual fue generada, sin ser manipulada o alterada por personas o procesos no autorizados.

La violación de integridad se presenta cuando un empleado, programa o proceso (por accidente o con mala intención) modifica o borra los datos importantes que son parte de la información, así mismo hace que su contenido permanezca inalterado a menos que sea modificado por personal autorizado, y esta modificación sea registrada, asegurando su precisión y confiabilidad.

c) Disponibilidad

La disponibilidad es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones. La disponibilidad es el acceso a la información y a los sistemas por personas autorizadas en el momento que así lo requieran.

En el caso de los sistemas informáticos utilizados para almacenar y procesar la información, los controles de seguridad utilizados para protegerlo, y los canales de comunicación protegidos que se utilizan para acceder a ella deben estar funcionando correctamente. La Alta disponibilidad sistemas objetivo debe estar disponible en todo momento, evitando interrupciones del servicio debido a cortes de energía, fallos de hardware, y actualizaciones del sistema.

Garantizar la disponibilidad implica también la prevención de ataque de denegación de servicio. Para poder manejar con mayor facilidad la seguridad de la información, las empresas o negocios se pueden ayudar con un sistema de gestión que permita conocer, administrar y minimizar los posibles riesgos que atenten contra la seguridad de la información del negocio.

La disponibilidad además de ser importante en el proceso de seguridad de la información, es además variada en el sentido de que existen varios mecanismos para cumplir con los niveles de servicio que se requiera.

Otros principios de la seguridad de la información que complementan a los anteriores son:

d) Autenticación o autentificación

Es la propiedad que permite identificar el generador de la información. Por ejemplo al recibir un mensaje de alguien, estar seguro que es de ese alguien el que lo ha mandado, y no una tercera persona haciéndose pasar por la otra (suplantación de identidad). En un sistema informático se suele conseguir este factor con el uso de cuentas de usuario y contraseñas de acceso.

e) No- Repudio

Proporciona protección contra la interrupción, por parte de alguna de las entidades implicadas en la comunicación, de haber participado en toda o parte de la comunicación.

No Repudio de origen: El emisor no puede negar que envío porque el destinatario tiene pruebas del envío, el receptor recibe una prueba infalsificable del origen del envío, lo cual evita que el emisor, de negar tal envío, tenga éxito ante el juicio de terceros. En este caso la prueba la crea el propio emisor y la recibe el destinatario.

- Prueba que el mensaje fue enviado por la parte específica.

No Repudio de destino: El receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción. Este servicio proporciona al emisor la prueba de que el destinatario legítimo de un envío, realmente lo recibió, evitando que el receptor lo niegue posteriormente. En este caso la prueba irrefutable la crea el receptor y la recibe el emisor.

- Prueba que el mensaje fue recibido por la parte específica.

Si la autenticidad prueba quién es el autor de un documento y cual es su destinatario, el “no repudio” prueba que el autor envió la comunicación (no repudio en origen) y que el destinatario la recibió (no repudio en destino). El no repudio evita que el emisor o el receptor nieguen la transmisión de un mensaje. Así, cuando

se envía un mensaje, el receptor puede comprobar que, efectivamente, el supuesto emisor envió el mensaje. De forma similar, cuando se recibe un mensaje, el emisor puede verificar que, de hecho, el supuesto receptor recibió el mensaje.

f) Control de acceso

El control de acceso constituye una importante ayuda para proteger al sistema operativo de la red, al sistema de aplicación y demás software de la utilización o modificaciones no autorizadas; para mantener la integridad de la información (restringiendo la cantidad de usuarios y procesos con acceso permitido) y para resguardar la información confidencial de accesos no autorizados.

Estos controles pueden implementarse en el Sistema Operativo, sobre los sistemas de aplicación, en bases de datos, en un paquete específico de seguridad o en cualquier otro tipo de software

Es conveniente tener en cuenta otras consideraciones referidas a la seguridad lógica, como por ejemplo las relacionadas al procedimiento que se lleva a cabo para determinar si corresponde un permiso de acceso (solicitado por un usuario) a un determinado recurso.

I.10 PLANIFICACIÓN DE LA SEGURIDAD

La rápida evolución del entorno técnico requiere que las organizaciones adopten un conjunto mínimo de controles de seguridad para proteger su información y sistemas de información. El propósito del plan de seguridad del sistema es proporcionar una visión general de los requisitos de seguridad del sistema y se describen los controles en el lugar o los previstos para cumplir esos requisitos. El plan de seguridad del sistema también delinea las responsabilidades y el comportamiento esperado de todos los individuos que acceden al sistema. Debe reflejar las aportaciones de distintos gestores con responsabilidades sobre el sistema, incluidos los propietarios de la información, el propietario de la red, y el alto funcionario de la agencia de información de seguridad (SAISO).

Los administradores de programas, los propietarios del sistema, y personal de seguridad en la organización debe entender el sistema de seguridad en el proceso de planificación. Los responsables de la ejecución y gestión de sistemas de información deben participar en el tratamiento de los controles de seguridad que deben aplicarse a sus sistemas.

A. CREACIÓN DE UN PLAN DE RESPUESTA A INCIDENTES

Es importante formular un plan de respuestas a incidentes, soportarlo a lo largo de la organización y probarlo regularmente. Un buen plan de respuestas a incidentes puede no sólo minimizar los efectos de una violación sino también, reducir la publicidad negativa.

Desde la perspectiva del equipo de seguridad, no importa si ocurre una violación o abertura (pues tales eventos son una parte eventual de cuando se hacen negocios usando un método de poca confianza como lo es Internet), sino más bien cuando ocurre. El aspecto positivo de entender la inevitabilidad de una violación a los sistemas (cualquier sistema donde se procese información confidencial, no está limitado a servicios informáticos) es que permite al equipo de seguridad desarrollar un curso de acciones para minimizar los daños potenciales. Combinando un curso de acciones con la experiencia le permite al equipo responder a condiciones adversas de una manera formal y oportuna.

El plan de respuesta a incidentes puede ser dividido en cuatro fases:

- Acción inmediata para detener o minimizar el incidente.
- Investigación del incidente.
- Restauración de los recursos afectados.
- Reporte del incidente a los canales apropiados.

Una respuesta a incidentes debe ser decisiva y ejecutarse rápidamente. Debido a que hay muy poco espacio para errores, es crítico que se efectúen prácticas de emergencias y se midan los tiempos de respuesta. De esta forma, es posible desarrollar una metodología que fomenta la velocidad y la precisión, minimizando el impacto de la indisponibilidad de los recursos y el daño potencial causado por el sistema en peligro.

Un plan de respuesta a incidentes tiene un número de requerimientos, incluyendo:

- Un equipo de expertos locales (un Equipo de respuesta a emergencias de computación).
- Una estrategia legal revisada y aprobada.
- Soporte financiero de la compañía.
- Soporte ejecutivo de la gerencia superior.
- Un plan de acción factible y probado.
- Recursos físicos, tal como almacenamiento redundante, sistemas en stand by y servicios de respaldo.

B. CONSIDERACIONES LEGALES

Otros aspectos importantes a considerar en una respuesta a incidentes son las ramificaciones legales.

Los planes de seguridad deberían ser desarrollados con miembros del equipo de asesoría jurídica o alguna forma de consultoría general. De la misma forma en que cada compañía debería tener su propia política de seguridad corporativa, cada compañía tiene su forma particular de manejar incidentes desde la perspectiva legal. Las regulaciones locales, de estado o federales están más allá del ámbito de este documento, pero se mencionan debido a que la metodología para llevar a cabo el análisis post-mortem, será dictado, al menos en parte, por la consultoría jurídica. La consultoría general puede alertar al personal técnico de las ramificaciones legales de una violación; los peligros de que se escape información personal de un cliente, registros médicos o financieros; y la importancia de restaurar el servicio en ambientes de misión crítica tales como hospitales y bancos.

C. PLANES DE ACCIÓN

Una vez creado un plan de acción, este debe ser aceptado e implementado activamente. Cualquier aspecto del plan que sea cuestionado durante la implementación activa lo más seguro es que resulte en un tiempo de respuesta pobre y tiempo fuera de servicio en el evento de una violación. Aquí es donde los ejercicios prácticos son invalorables. La implementación del plan debería ser acordada entre todas las partes relacionadas y ejecutada con seguridad, a menos que se llame la atención con respecto a algo antes de que el plan sea colocado en producción.

La respuesta a incidentes debe ir acompañada con recolección de información siempre que esto sea posible.

Los procesos en ejecución, conexiones de red, archivos, directorios y mucho más deberían ser auditados activamente en tiempo real. Puede ser muy útil tener una toma instantánea de los recursos de producción al hacer un seguimiento de servicios o procesos maliciosos. Los miembros de CERT y los expertos internos serán recursos excelentes para seguir tales anomalías en un sistema.

D. EL MANEJO DE RIESGOS

Dentro de la seguridad en la información se lleva a cabo la clasificación de las alternativas para manejar los posibles riegos que un activo o bien puede tener dentro de los procesos de organización. Esta clasificación lleva el nombre de manejo de riegos. El manejo de riesgos, conlleva una estructura bien definida, con un control adecuado y su manejo, habiéndolos identificado, priorizados y analizados, a través de acciones factibles y efectivas. Para ello se cuenta con las siguientes técnicas de manejo del riesgo:

- Evitar. El riesgo es evitado cuando la organización rechaza aceptarlo, es decir, no se permite ningún tipo de exposición. Esto se logra simplemente con no comprometerse a realizar la acción que origine el riesgo. Esta técnica tiene más desventajas que ventajas, ya que la empresa podría abstenerse de aprovechar muchas oportunidades. Ejemplo:

  No instalar empresas en zonas sísmicas

- Reducir. Cuando el riesgo no puede evitarse por tener varias dificultades de tipo operacional, la alternativa puede ser su reducción hasta el nivel más bajo posible. Esta opción es la más económica y sencilla. Se consigue optimizando los procedimientos, la implementación de controles y su monitoreo constante. Ejemplo:

  No fumar en ciertas áreas, instalaciones eléctricas anti flama, planes de contingencia.

- Retener, Asumir o Aceptar el riesgo. Es uno de los métodos más comunes del manejo de riesgos, es la decisión de aceptar las consecuencias de la ocurrencia del evento. Puede ser voluntaria o involuntaria, la voluntaria se caracteriza por el reconocimiento de la existencia del riesgo y el acuerdo de asumir las perdidas involucradas, esta decisión se da por falta de alternativas. La retención involuntaria se da cuando el riesgo es retenido inconscientemente. Ejemplo de asumir el riesgo:

  Con recursos propios se financian las pérdidas.

- Transferir. Es buscar un respaldo y compartir el riesgo con otros controles o entidades. Esta técnica se usa ya sea para eliminar un riesgo de un lugar y transferirlo a otro, o para minimizar el mismo, compartiéndolo con otras entidades. Ejemplo:

  Transferir los costos a la compañía aseguradora

I.11 RFC 3205

El RFC 3205 especifica reglas y buenas prácticas orientadas a diseñadores de protocolos para utilizar HTTP como sustrato de otro protocolo. Esto es, construir un protocolo en una capa superior a HTTP. Si bien menciona que HTTP no fue diseñado para ser la capa inferior de otro protocolo, entre las ventajas para esta utilización, el RFC3205 menciona: su familiaridad y popularidad, la posibilidad de reutilizar bibliotecas de funciones existentes, la posibilidad de utilizar mecanismos de autenticación y comunicación cifrada, la capacidad de HTTP de atravesar firewalls (lo que incentiva a utilizarlos para diseñar otros protocolos) y la necesidad, en ciertos casos, de construir aplicaciones que obligatoriamente estén sustentadas sobre HTTP.[5]

[5] Network Working Group, University of Tennessee, (2002), pp 2

Entre las desventajas, el RFC3205 menciona que HTTP comenzó como un protocolo simple y se hizo más complejo al agregarle funcionalidad no anticipada en el diseño original. Cabe mencionar que la versión 1.1 de HTTP introdujo varias modificaciones respecto a la versión 1.0, algunas de ellas se introdujeron sin una evaluación práctica real.

La sobrecarga en la comunicación que requiere HTTP, dados los campos que requieren los mensajes, resulta en un rendimiento inferior respecto al diseño de un protocolo ad hoc. Por otra parte, el RFC3205 indica que una aplicación que no pueda operar en presencia de intermediarios en el trayecto del mensaje no debe utilizar HTTP como sustrato.

En el caso de utilizar un protocolo como sustrato, los mensajes del protocolo de la capa superior se colocan en el campo de datos del protocolo inferior como una capa más en la pila de capas de la arquitectura de protocolos. Cabe destacar que la esteganografía no utiliza al protocolo estrictamente como sustrato sino como portador de sus mensajes. Más allá de esta distinción, las ventajas y desventajas mencionadas por la RFC 3205 pueden hacerse extensivas a la aplicación de esteganografía sobre HTTP.

CAPÍTULO II

POPUESTA DE PROCEDIMIENTO DE INTEGRACIÓN DE LA ESTEGANOGRAFÍA AL PROTOCOLO HTTP

II.1 CAPA DE APLICACIÓN

La capa de aplicación define las aplicaciones de red y los servicios de Internet estándar que puede utilizar un usuario.

El usuario normalmente no interactúa directamente con el nivel de aplicación, suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente. Así por ejemplo un usuario no manda una petición «GET /index.html HTTP/1.0» para conseguir una página en html, ni lee directamente el código html/xml, o cuando los usuarios conversan a través del Messenger, no es necesario que los interlocutores codifiquen la información y los datos del destinatario para entregarla a la capa de Presentación (capa 6) para que realice el envío del paquete. En la figura 2.1 se observa la interacción del usuario y la capa de aplicación.

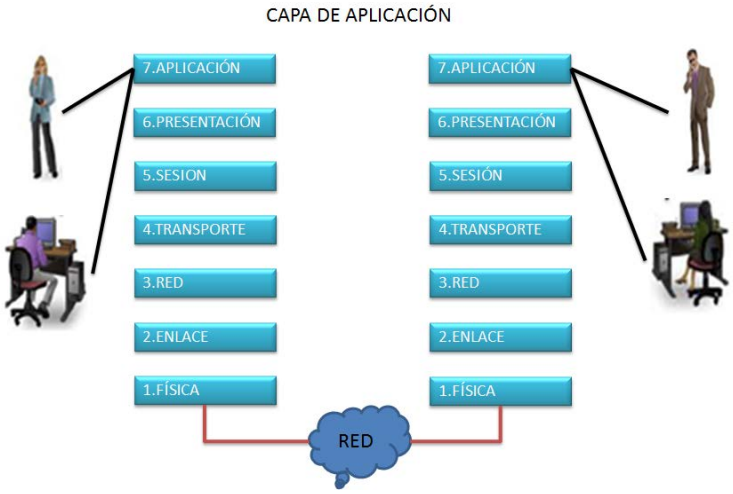

Fig. 2.1 Capa de aplicación del modelo OSI

II.2 APLICACIONES: LA INTERFAZ ENTRE REDES

El modelo de referencia de interconexión de sistemas abiertos es una representación abstracta en capas, creada como guía para el diseño del protocolo de red. El modelo OSI divide el proceso de networking en diferentes capas lógicas, cada una de las cuales tiene una única funcionalidad y a la cual se le asignan protocolos y servicios específicos.

En este modelo, la información se pasa de una capa a otra, comenzando en la capa de Aplicación en el host de transmisión, siguiendo por la jerarquía hacia la capa Física, pasando por el canal de comunicaciones al host de destino, donde la información vuelve a la jerarquía y termina en la capa de Aplicación. La figura 2.2 ilustra los pasos en este proceso.

La capa de Aplicación, Capa siete, es la capa superior de los modelos OSI y TCP/IP. Es la capa que proporciona la interfaz entre las aplicaciones que se utilizan para la comunicación y la red subyacente en la cual se transmiten los mensajes. Los protocolos de capa de aplicación se utilizan para intercambiar los datos entre los programas que se ejecutan en los hosts de origen y destino. Existen muchos protocolos de capa de aplicación y siempre se desarrollan protocolos nuevos.

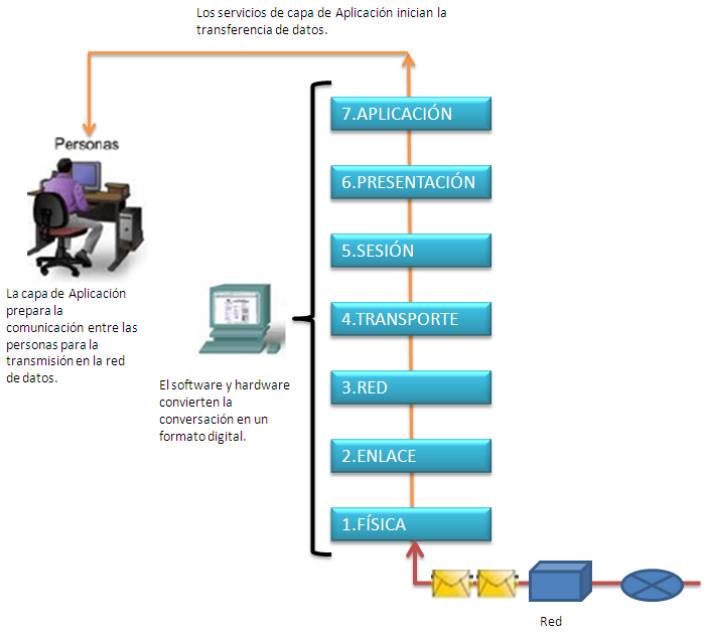

Fig 2.2. Interfaz entre redes

Dentro de la capa de aplicación se encuentran dos tipos de programas o aplicaciones:

- Aplicaciones de red
- Servicios de la capa de aplicación

a) Aplicaciones de red

Son aquellos programas que utiliza el usuario final para comunicarse en la red, ya sean programas de mensajería, navegadores web, y/o clientes de correo electrónico, por mencionar algunos.

b) Servicios de la capa de aplicación

Son los programas que el usuario no ve, pero que son necesarios para que las aplicaciones funcionen correctamente. Estos servicios son por ejemplo, la trasferencia de archivos, funciones de prioridades en red, cola de impresión en red, etcétera.

Los servicios deben implementar varios protocolos, ya que son muchas las distintas aplicaciones que se comunican en una red, como se muestra en la figura 2.3.

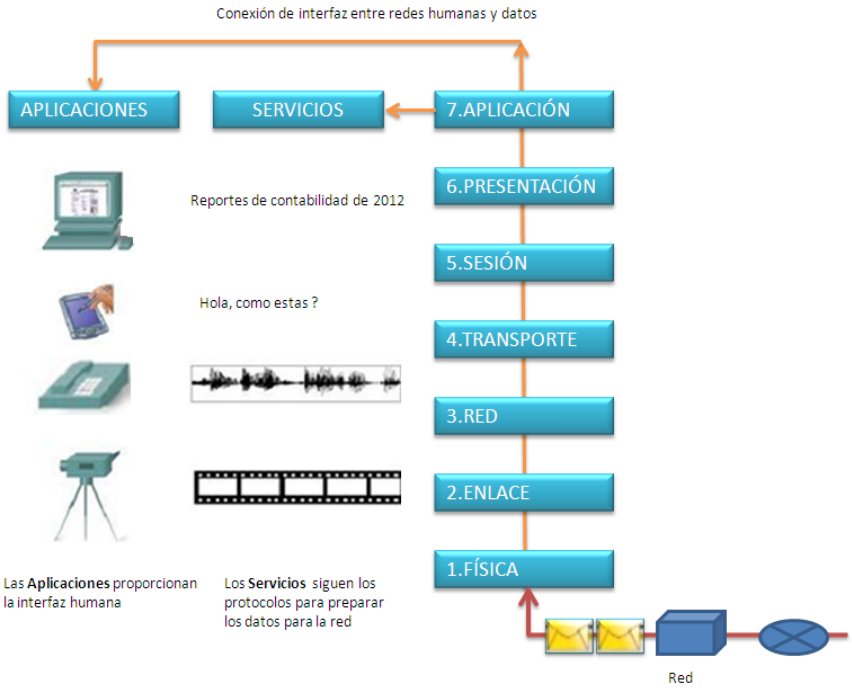

Fig. 2.3 Aplicaciones y servicios de la capa de aplicación

Los protocolos de la capa de aplicación son aquellos que se utilizan para intercambiar los datos entre los programas que se están ejecutando en el origen y destino.

Dicho esto, se enlistan algunos protocolos utilizados en esta capa:

- Protocolo de servicio de nombres (DNS)
- Protocolo de trasferencia de hipertexto (HTTP)
- Protocolo de trasferencia de correo (SMTP)
- Protocolo de emulación de terminal (Telnet)
- Protocolo de trasferencia de archivos (FTP)

Todos estos protocolos proporcionan la información de control y su formato necesario para las funciones de comunicación más comunes dentro de Internet.

II.3 FUNCIONES DE LOS PROTOCOLOS

Los protocolos establecen reglas para el intercambio de datos entre las diferentes aplicaciones y servicios instalados en los dispositivos de origen y destino dentro de una red.

Además, los protocolos son los encargados de estructurar los mensajes que se envían entre origen y destino.

Se debe destacar que cada protocolo realiza una acción específica y que por este motivo son muchísimos los protocolos existentes hoy en día.

Para que exista una comunicación fiable entre dispositivos los protocolos se enfocan en:

a) Sintaxis: Formato de los datos, los niveles de tensión y de codificación de bit.

b) Semántica: Información de control para controlar las funciones de red.

c) Tiempo: Sincronización y control de flujo.

Algunas de las funciones que realizan los protocolos son:

a) La segmentación / reensamblado

- **Segmentación de datos**

Debido a que cada aplicación genera un stream de datos para enviar a una aplicación remota, estos datos deben prepararse para ser enviados por los medios en partes manejables. Los protocolos de la capa de Transporte describen los servicios que segmentan estos datos de la capa de Aplicación. Esto incluye la encapsulación necesaria en cada sección de datos. Cada sección de datos de aplicación requiere que se agreguen encabezados en la capa de Transporte para indicar la comunicación a la cual está asociada.

- **Reensamble de segmentos**

En el host de recepción, cada sección de datos puede ser direccionada a la aplicación adecuada. Además, estas secciones de datos individuales también deben reconstruirse para generar un stream completo de datos que sea útil para la capa de Aplicación. Los protocolos de la capa de Transporte describen cómo se utiliza la información de encabezado de dicha capa para reensamblar las secciones de datos en streams y enviarlas a la capa de Aplicación.

b) Encapsulación

Encapsulación es un método de diseño modular de protocolos de comunicación en el cual las funciones lógicas de una red son abstraidas ocultando información a las capas de nivel superior.

La encapsulación es una característica en la mayoría de modelos de redes, incluyendo el modelo OSI y la familia de protocolos TCP/IP.

c) Conexión de control

La transferencia de datos puede ser o bien orientada a la conexión u orientada a la desconexión

En las transferencias sin conexión cada unidad de datos del protocolo (PDU) es independiente de todos los demás enviados.

En transferencias orientadas a la desconexión una conexión lógica es establecida antes de la transferencia de datos, entonces cada PDU enviado tiene un número de secuencia.

La secuenciación admite la entrega ordenada, control de flujo y control de errores

La función de conexión de control de un protocolo gestiona el establecimiento y la desconexión de un enlace.

d) Entrega ordenada

La unidad de datos del protocolo (PDU) puede viajar por rutas diferentes, y pueden llegar fuera de orden con respecto a la orden de transmitir un protocolo debe ser capaz de reordenar las unidades PDU en el orden correcto.

e) Control de Flujo

Un receptor puede no ser capaz de procesar la unidad de datos del protocolo tan rápido como el transmisor puede enviar.

Un receptor necesita alguna manera de limitar la tasa del transmisor, funciones de control de flujo asegurar que los datos enviados no abrumar al receptor

f) Control de errores

La unidad de datos del protocolo pueden ser perdidos o dañados

La retransmisión en caso de fallo de acuse de recibo es un método común para el manejo de las PDU perdidas

Comprobaciones de redundancia cíclica se utilizan a menudo para detectar PDU dañadas

g) Direccionamiento

Un protocolo debe tener un medio para la identificación de un usuario en particular utilizando una aplicación particular en un huésped que reside en alguna red.

Direccionamiento es un medio para protocolos para identificar estas necesidades

h) Multiplexación

Multiplexación se utiliza para mejorar la eficiencia y el uso del medio de transmisión.

Existen para apoyar las funciones de división de frecuencia o tiempo de multiplexación así como multiplexar las conexiones.

i) Servicios de Transmisión

Existen otros tipos de servicios a las capas superiores.

Existen tres servicios comunes son: prioridad, nivel de servicio, y la seguridad.

II.4 ELEMENTOS QUE INTERACTUAN CON LOS PROTOCOLOS

Se debe saber que para que la comunicación tenga éxito, se deben tener implementados tanto en el origen como en el destino los mismos protocolos.

Hay varios elementos que interactúan junto con los protocolos como:

- Servidores
- Modelo cliente / servidor

- Redes y aplicaciones P2P

Todos ellos relacionados con la capa de aplicación:

a) Servidores

Son dispositivos que responden a una solicitud de aplicaciones de cliente, en general, un servidor suele ser un ordenador que contiene mucha información para ser compartida con muchos sistemas clientes.

Cada servidor suele tener un servicio o proceso denominado *daemon*, que se encarga de escuchar las peticiones, darles prioridades y ejecutar las respuestas en los formatos adecuados.

Dependiendo del fin de cada servidor, algunos requerirán sistemas de seguridad como usuario y contraseña, por lo que tendrán listas con los usuarios y contraseñas permitidos para dar respuesta a las solicitudes.

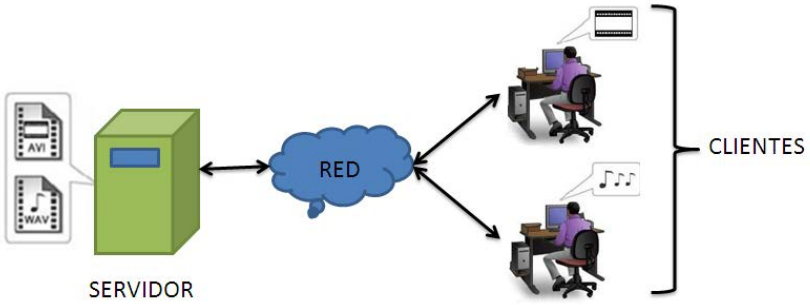

Fig. 2.4 Funcionamiento de un servidor

Se debe tener en cuenta que una aplicación puede emplear varios servicios diferentes de la capa de aplicación. De esta forma, un usuario envía una petición y el servidor realmente puede llegar a recibir muchas más para esa única petición que hace el cliente.

Por otro lado, el servidor suele recibir varias peticiones de clientes distintos a la vez, como se observa en la figura 2.4, y es aquí donde entran en juego los protocolos.

b) Modelo cliente-servidor

Cómo ya se sabe, si por ejemplo, se quiere ver una página web desde un dispositivo móvil u ordenador, primero se tiene que hacer una petición al servidor que contiene esa información. Bien, pues esto es básicamente en lo que consiste este modelo. El usuario hace una petición al servidor y este contesta.

Este modelo se encuentra dentro de la capa de aplicación, ya que es la forma más directa que tiene un cliente para recibir información. Los protocolos de esta capa son los responsables de darle un formato a esas solicitudes y respuestas.

Para que quede más claro, este tipo de modelo de red puede ser un cliente de correo. Se abre el cliente de correo y este hace una petición al servidor de correo para que le envíe los correos nuevos que tenga.

Dentro de este modelo de red estaría la carga y descarga de datos de un cliente a un servidor o viceversa, como se ejemplifica en la figura 2.5.

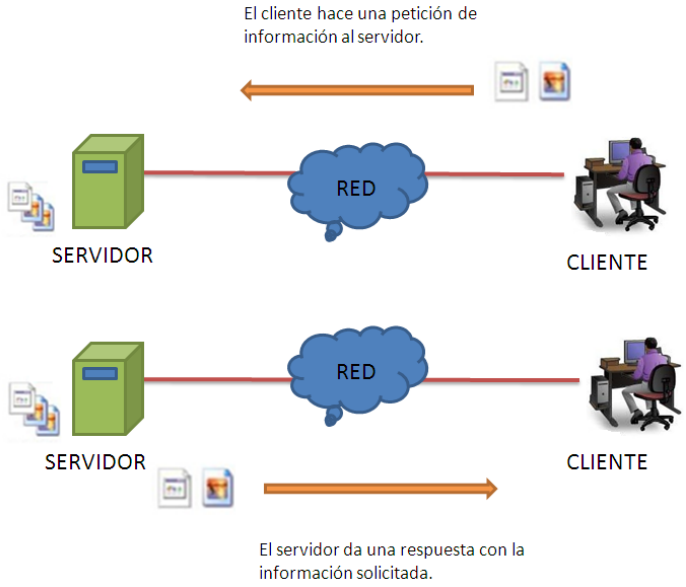

Fig 2.5 Modelo Cliente/Servidor

c) Redes y aplicaciones P2P

- **Redes P2P**: En este tipo de redes se encuentran dos o más equipos conectados entre sí por una red, pero no dependen de un servidor para compartir información (fig.2.6). Cada equipo funciona como cliente y servidor a la vez. Las funciones de cliente o servidor se activan por solicitud. Un ejemplo de este tipo de redes son las que se tienen en casa, es decir, un par de ordenadores conectados entre sí para compartir archivos. Incluso si se coloca una impresora en un equipo y se comparte, también se está hablando de red punto a punto. (siempre y cuando la impresora esté conectada a un equipo de forma directa y no a la red). El problema de este tipo de redes es la seguridad, ya que cada equipo debe administrar su propia seguridad al no tener un servidor que centralice las peticiones, las cuentas de usuario y sus contraseñas.

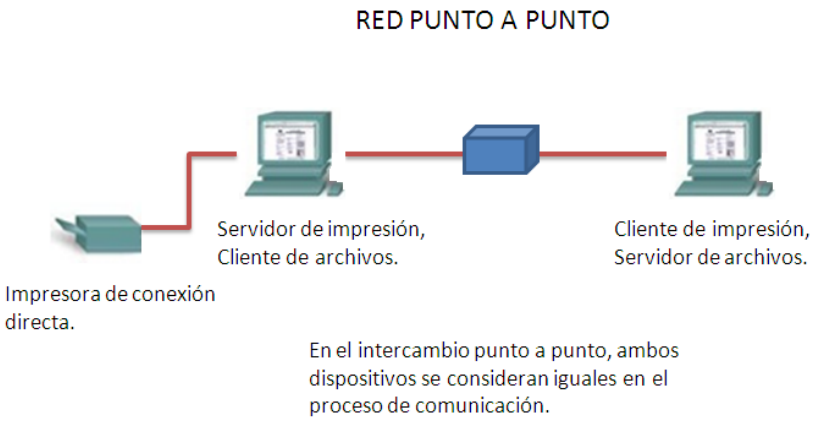

Fig.2.6 Redes P2P

- **Aplicaciones P2P:** Una aplicación punto a punto permite a un dispositivo actuar como cliente o como servidor dentro de la misma comunicación (fig. 2.7). En este tipo de aplicaciones cada cliente es un servidor y cada servidor es un cliente. Este tipo de aplicaciones requieren de una interfaz de usuario, aunque luego tengan servicio ejecutándose en segundo plano. Este tipo de aplicaciones puede darse entre las redes cliente-servidor, en las redes punto a punto y en Internet. Un ejemplo de este tipo de aplicación serían las ya mundialmente conocidas como aplicaciones de intercambio de archivos (Emule, Torrent, etcétera) o los sistemas de mensajería instantánea (Messenger, WhatsApp, entre otros).

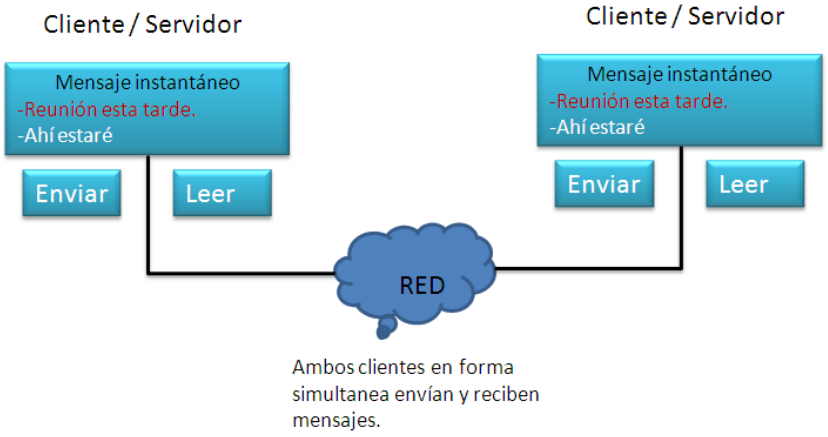

Fig 2.7 Aplicaciones P2P

II.5 SERVICIO WWW Y HTTP

Cuando se escribe una dirección Web (o URL) en un explorador de Internet, el explorador establece una conexión con el servicio Web del servidor que utiliza el protocolo HTTP. URL (Localizador uniforme de recursos) y URI (Identificador uniforme de recursos) son los nombres que la mayoría de las personas asocian con las direcciones web. El URL http://www.google.com/mail es un ejemplo de un URL que se refiere a un recurso específico: una página Web denominada mail en un servidor identificado como google.com. Los exploradores Web son las aplicaciones de cliente que utilizan nuestras computadoras para conectarse con la World Wide Web (www) y para acceder a los recursos almacenados en un servidor Web. Al igual que con la mayoría de los procesos de servidores, el servidor Web funciona como un servicio básico y genera diferentes tipos de archivos disponibles. Para acceder al contenido, los clientes Web realizan conexiones al servidor y solicitan los recursos deseados. El servidor responde con los recursos y, una vez recibidos, el explorador interpreta los datos y los presenta al usuario. Los exploradores pueden interpretar y presentar muchos tipos de datos, como texto sin formato o Lenguaje de marcado de hipertexto (HTML, lenguaje que se utiliza para construir una página Web). Otros tipos de datos, sin embargo, requieren de otro servicio o programa. Generalmente se los conoce como plug-ins o complementos. Para ayudar al explorador a determinar qué tipo de archivo está recibiendo, el servidor especifica qué clase de datos contiene el archivo. Para comprender mejor cómo interactúan el explorador Web con el cliente Web, podemos analizar cómo se abre una página web en un explorador. Para este ejemplo, utilizaremos la dirección URL: https://www.segurosbancomer.com.mx/seguros/tleu/segurosban/index.jsp, el explorador interpreta las tres partes de la URL:

1. http (el protocolo o esquema).
2. www.segurosbancomer.com.mx (el nombre del servidor).
3. seguros/tleu/segurosban/index.jsp (el nombre específico del archivo solicitado).

El explorador luego verifica con un servidor de nombres para convertir a www.segurosbancomer.com.mx en una dirección numérica que utilizará para conectarse con el servidor. Al utilizar los requerimientos del protocolo HTTP, el explorador envía una solicitud GET al servidor y pide el archivo web-server.htm.

El servidor, a su vez, envía al explorador el código HTML de esta página Web. Finalmente, el explorador descifra el código HTML y da formato a la página para la ventana del explorador. En la figura 2.8 se ejemplifica esta comunicación.

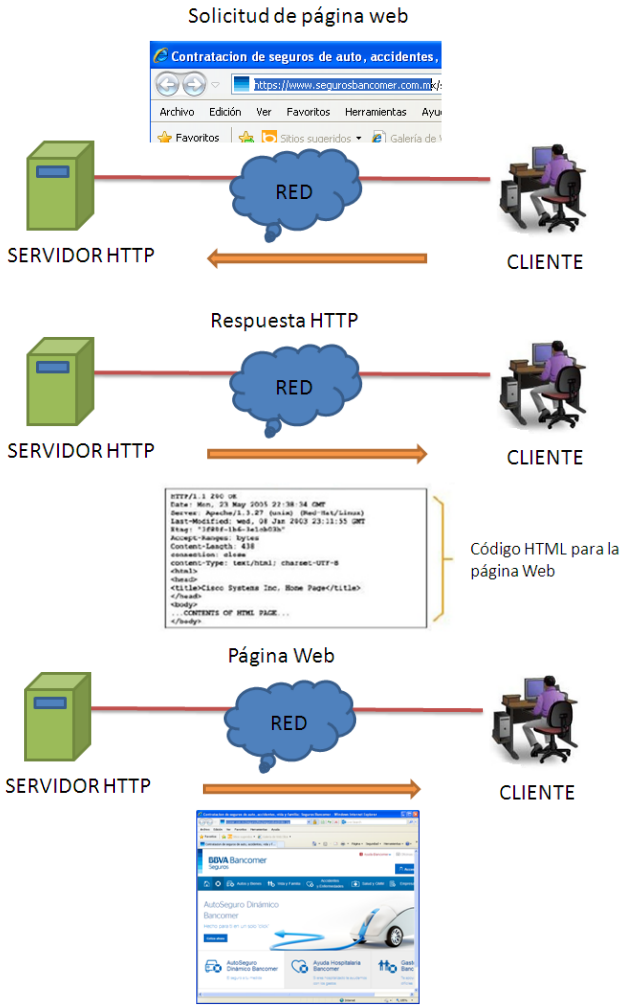

Fig 2.8 Servicio www y HTTP

II.6 ESTEGANOGRAFÍA DE RED

Se han escrito muchos artículos acerca de la ocultación de datos en imágenes y fotografías. También resulta bastante conocida la posibilidad de guardar información en el ruido de fondo de una canción, de manera imperceptible al oído humano, o en el de una secuencia de vídeo. La que resulta menos conocida es la denominada esteganografía de red, la cual utiliza determinadas características de los protocolos de red para encapsular datos y transmitirlos camuflados por Internet.

Como se mencionó en el capítulo anterior la esteganografía de red se basa en tres métodos principales:

- **Encapsulación de un protocolo en otro.**
- **Encapsulación de información en el campo de datos de un protocolo.**
- **Encapsulación de información en un campo numérico de un protocolo.**

a) Encapsulamiento de un protocolo en otro

Se conoce como túnel o tunneling, es la técnica que consiste en encapsular un protocolo de red sobre otro (protocolo de red encapsulador) creando un túnel de información dentro de una red de computadoras. El uso de esta técnica persigue diferentes objetivos, dependiendo del problema que se esté tratando, como por ejemplo la comunicación de islas en escenarios multicast, la redirección de tráfico, entre otras. La técnica de tunelizar se suele utilizar para trasportar un protocolo determinado a través de una red que, en condiciones normales, no lo aceptaría. Otro uso de la tunelización de protocolos es la creación de diversos tipos de redes privadas virtuales.

Uno de los ejemplos más claros de utilización de esta técnica consiste en la redirección de tráfico en escenarios IP Móvil. En escenarios de IP móvil, cuando un nodo-móvil no se encuentra en su red base, necesita que su home-agent realice ciertas funciones en su puesto, entre las que se encuentra la de capturar el tráfico dirigido al nodo-móvil y redirigirlo hacia él. Esa redirección del tráfico se realiza usando un mecanismo de tunneling, ya que es necesario que los paquetes conserven su estructura y contenido originales (dirección IP de origen y destino, puertos, etc.) cuando sean recibidos por el nodo-móvil.

Es relativamente conocido y muy utilizado hoy día para el ocultamiento de sistemas de descargas P2P. Un ejemplo es el encapsulado en el protocolo HTTP para eludir las restricciones de los firewall perimetrales. Este último es el enfoque de herramientas como **HTTP-Tunnel** que, mediante la instalación de un servidor de SOCKS en el PC, permiten establecer un túnel cifrado con un servidor remoto, que hace de intermediario, a través del puerto de HTTP (abierto generalmente en el firewall) simulando que es tráfico web. El servidor que hace de intermediario, situado en el exterior del firewall, desencapsula luego el tráfico y lo remite a sus destinatarios originales.

b) Encapsulación de información en el campo de datos de un protocolo

Este caso se da cuando un protocolo de uso común en la red incluye un campo de datos que el usuario puede llenar con un mensaje. Alguien que vigila la red advertiría el uso del protocolo pero a no ser que se pusiera a investigar el contenido de los campos de los paquetes intercambiados sería incapaz de detectar que se está transmitiendo información.

Un claro ejemplo de este tipo de técnicas es el protocolo ICMP. Este protocolo se utiliza para tareas de mantenimiento y control interno de redes. La herramienta más conocida que utiliza este protocolo es *Ping*. Como ya es sobradamente conocido, la herramienta ping permite averiguar si un determinado equipo está alcanzable a través de la red. Para ello se le lanza un Ping y si se recibe respuesta significa que hay conectividad con dicho equipo. Lo que ocurre entre bastidores es que el equipo que lanza el ping emite un paquete de tipo ICMP ECHO-REQUEST y el equipo destino, al recibir este paquete, genera otro de respuesta denominado ICMP ECHO-REPLY. Cuando el equipo que lanza los pings comienza a recibir paquetes ICMP ECHO-

REPLY puede concluir que el equipo destino está “vivo”. Dado lo habitual de este protocolo en las redes modernas resulta un candidato interesante como cobertura a nuestro enlace de datos. Para ello se puede utilizar el campo de 56 bytes con el que cuentan los paquetes ICMP ECHO. El propósito original de este campo no está muy claro. En apariencia, los sistemas operativos actuales lo usan como una firma llenándolo con una secuencia de datos más o menos predefinida (p.ej. Windows mete siempre el abecedario “abcd...” mientras que Linux que los caracteres “01234567”) con la idea de que si los paquetes de respuesta contienen la misma secuencia de datos entonces son correctos. Sin embargo, nadie dice cuál debe ser esa secuencia de datos por lo que nosotros podemos meter los que queramos y enviárselos al equipo destino en un Ping.

Vamos a ilustrar esto con un ejemplo usando **Scapy** . Supongamos que Alice quiere enviarle a Bob (10.0.0.105) el siguiente mensaje: “Creo que sospechan de ti.”. Podríamos encapsular este mensaje en un paquete ICMP de la siguiente manera desde la consola de Scapy:

```
Welcome to Scapy (v1.1.1 / -)
>>> ip = IP()
>>> ip.dst = "10.0.0.105"
>>> icmp = ICMP()
>>> mensaje = "Creo que sospechan de ti."
>>> icmp.add_payload(mensaje)
>>> packet = ip / icmp
>>> sr1(packet)
Begin emission:
.*Finished to send 1 packets.
Received 2 packets, got 1 answers, remaining 0 packets
<\IP version=4L ihl=5L tos=0x0 len=53 id=53122 flags= frag=0L ttl=64 proto=icmp chksum=0x96d9 src=10.0.0.105 dst=10.0.0.4 options='' |<\ICMP type=echo-reply code=0 chksum=0xd436 id=0x0 seq=0x0 |>>
>>>
```

Bob podría visualizar este mensaje activando un tcpdump en su interfaz que capturase todos los paquetes icmp que le llegasen.

Si un administrador estuviese monitorizando la red sólo vería un ping, tráfico que no tiene por qué ser sospechoso ya que, a priori, no es un protocolo susceptible de transportar información (aunque se ve que en realidad sí puede). Además, un ping y su respuesta resultan imperceptibles dentro del maremagnum que es una red moderna. Si aún con todo el administrador ha sido tan minucioso como para detectar el intercambio de paquetes y capturarlos con un sniffer, en la fig. 2.9 se muestra lo que se vería:

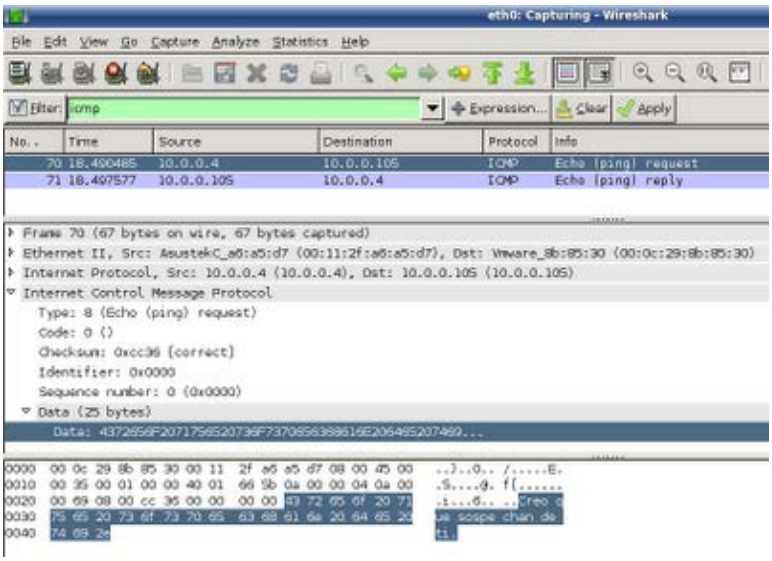

Fig. 2.9 Mensaje oculto usando el protocolo ICMP

Como se puede apreciar, la captura del paquete ICMP sí que muestra claramente el contenido del mensaje en el campo Data. Para evitar la detección y además dotar de confidencialidad el mensaje lo normal es que en vez de encapsular el mensaje en claro en el campo Data, se cifre primero (por ejemplo con una clave simétrica conocida por ambos extremos) y luego se empaquete en el ping. De esta manera el administrador lo que vería sería un amasijo de bytes aparentemente aleatorios que no tienen por qué hacerle sospechar de que lo que está viendo realmente es un mensaje cifrado.

c) Encapsulación de información en un campo numérico de un protocolo.

El tercer método de esteganografía de red, el uso de un **campo numérico**, no es sino una variante del anterior. La diferencia es que en vez de utilizar un campo específicamente destinado a datos se trata de utilizar uno dedicado a contener valores numéricos, como por ejemplo:

- **El campo de Identificación de la cabecera IP.** Con 16 bits de longitud, se utiliza para darle un número identificativo a cada paquete de manera que sea posible reconstruirlo en caso de que se tenga que realizar fragmentación en alguno de los nodos intermedios del trayecto.
- **El campo de Número Inicial de Secuencia de la cabecera TCP**. Sus 32 bits de longitud permite enviar cuatro caracteres ASCII por paquete. Se utiliza para establecer el número inicial a partir del cual se numerarán el resto de paquetes, de manera que se puedan detectar pérdidas o llegadas desordenadas de paquetes. Lo fija el que envía el paquete inicial.
- **El campo de Número de Secuencia Reconocido de la cabecera TCP**. También de 32 bits, lo fija el extremo que responde al paquete inicial cogiendo el Número Inicial de Secuencia de ese paquete y sumándole uno. Este campo, se utiliza para realizar transmisiones ocultas de datos de manera indirecta.

Los dos primeros campos se utilizarían igual que el de datos del ping que se vió antes, sólo habría que tener en cuenta la menor longitud de los campos y que habría que codificar las cadenas a un valor numérico de la longitud necesaria en función de sus valores ASCII (y posteriormente decodificarlos en el extremo receptor). **Stegtunnel** utiliza precisamente estos dos campos.

El uso del campo de Número de Secuencia Reconocido de la cabecera TCP/IP es más interesante y es el que utilizan herramientas como **Ncovert** . Supongamos que Alice quiere enviarle un mensaje a Bob, pero ella sabe que hay un administrador de red muy paranoico que no le quita el ojo de encima a Bob y monitoriza todo el tráfico de su ordenador. Resulta que Alice no conoce oficialmente a Bob y sabe que cualquier acercamiento a él levantaría sospechas y lo mismo pasaría a nivel de red si se detectase tráfico desde el PC de Alice hacia el de Bob. Pero Alice sabe que puede ocultar su mensaje entre el tráfico "legal" de Bob. Para ello Alice eligiría un servidor al que acceda usualmente Bob, llamémoslo por ejemplo Mercurio, y lo utilizaría de intermediario para pasarle el mensaje, esto se muestra en la figura 2.10.

La idea es la siguiente:

1. Alice crearía sucesivos paquetes SYN (de establecimiento de conexión) con Mercurio poniendo en el campo de Número Inicial de Secuencia TCP (o en el de Identificación IP) los caracteres de su mensaje tal y como veíamos antes pero falsificaría el origen del paquete SYN poniendo que proviene de la dirección IP de Bob.
2. Mercurio iría creando paquetes de SYN+ACK con destino a Bob (ya que él cree que es Bob el que le envía los paquetes SYN), utilizando en el campo de Número de Secuencia Reconocido el número fijado por Alice más 1.
3. Bob sólo tendría que ir recopilando dichos paquetes SYN+ACK, extraer sus campos de Número de Secuencia Reconocido, restarle 1 y pasarlo a ASCII.

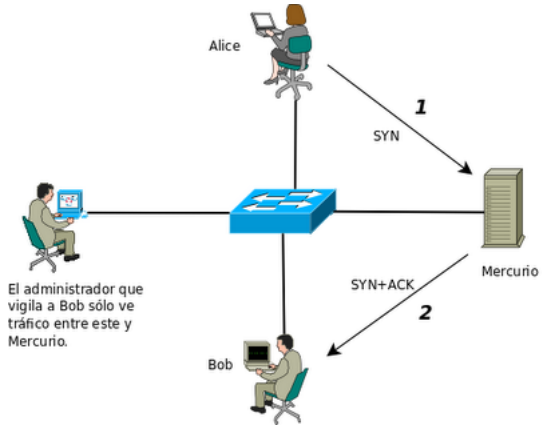

Fig. 2.10 Comunicación usando la encapsulación de información en un campo numérico de un protocolo.

La ventaja de este sistema es que el acechante administrador sólo vería un flujo de paquetes entre Bob y Mercurio por lo que no sospecharía nada. Además, no es necesario dirigir los paquetes a un puerto a la escucha de Mercurio, ya que en caso de utilizar un puerto cerrado, Mercurio reenviará los datos a Bob en un

paquete RST+ACK en vez de en un SYN+ACK. La pega es que los paquetes RST+ACK llaman bastante más la atención en una captura de red que los de SYN+ACK al señalar errores en las conexiones, por lo que si Alice quisiese minimizar la probabilidad de llamar la atención del administrador fijaría un puerto de destino abierto (por ejemplo, si Mercurio fuese un servidor Web utilizaría el puerto 80).

La creación de herramientas para realizar este tipo de comunicaciones es extremadamente sencilla. A continuación se muestra un ejemplo programado con Python y Scapy.

La herramienta que utilizaría Alice podría ser:

```
##############################################
# STEGTRANSPORT
#
# Programado por: Dante Signal31
#
# Stegtransport permite transferir datos camuflados
# en las cabeceras TCP de paquetes SYN de apariencia
# perfectamente normal.
#
##############################################

##########
# LIBRERIAS
##########
import scapy
import time
import random
############
# CONSTANTES
############
# Por simplicidad marcamos el final del mensaje con una '#'.
MENSAJE = "Creo que sospechan de ti. Sera /
mejor que tengas cuidado.#"
IP_DESTINO = "192.168.10.105"
IP_INTERMEDIARIO = "192.168.10.106"
PUERTO_ESCUCHA_INTERMEDIARIO = 80
INTERVALO_DE_ESPERA=0.1

##################
# PROGRAMA PRINCIPAL
##################
print "Mensaje a enviar: " + MENSAJE
print "Destinatario del mensaje: " + IP_DESTINO
print "Intermediario: " + IP_INTERMEDIARIO
for caracter in MENSAJE:
   caracter_valor_ascii = ord(caracter)
   print "Enviando caracter: " + caracter + /
```

```
" con valor ASCII: " + str(caracter_valor_ascii) + "..."
    paquete_ip = scapy.IP()
    paquete_ip.dst = IP_INTERMEDIARIO
    # Falsificamos el origen del paquete
# para que el intermediario responda al
# destinatario del mensaje oculto (IP_DESTINO)
# y no a nosotros.
    paquete_ip.src = IP_DESTINO
    paquete_tcp = scapy.TCP()
    paquete_tcp.seq = caracter_valor_ascii
    paquete_tcp.dport = PUERTO_ESCUCHA_INTERMEDIARIO
    # Si no fijamos un puerto origen, Scapy utilizará
# siempre el 20, lo que puede ser muy sospechoso.
    paquete_tcp.sport = random.randint(49152, 65535)
    paquete = paquete_ip / paquete_tcp
    scapy.sr1(paquete)
    print "enviado caracter."
    # Esperar entre un paquete y otro reduce la probabilidad
# de pérdidas, de llegadas desordenadas y de detecciones
# por parte de los IDS.
    time.sleep(INTERVALO_DE_ESPERA)
print "Mensaje enviado."
print "¡Que tenga un buen dia!"
```

En su lado Bob ejecutaría el siguiente programa:

```
################################################
# STEGRECEIVE
#
# Programado por: Dante Signal31
#
# Stegreceive permite recibir datos camuflados en
# las cabeceras TCP de paquetes SYN de apariencia
# perfectamente normal.
#
################################################

##########
# LIBRERIAS
##########
import scapy
import sys

#############
# CONSTANTES
#############
TRUE = 1
FALSE = 0
```

```
IP_INTERMEDIARIO = "192.168.10.106"

###################
# PROGRAMA PRINCIPAL
###################
print "Esperando mensaje desde el intermediario: " /
+ IP_INTERMEDIARIO + "\n---"
fin_mensaje = FALSE
# Fijamos el filtro de captura del tcpdump.
filtro_captura = "ip src host " + IP_INTERMEDIARIO
while (not fin_mensaje):
   paquete_recibido = scapy.sniff( /
filter=filtro_captura , count = 1)
   # Accedemos al campo ACK del subpaquete TCP recibido.
   caracter_valor_ascii = /
paquete_recibido[0][scapy.TCP].ack  - 1
   caracter = chr(caracter_valor_ascii)
   if (caracter == '#'):
      fin_mensaje = TRUE
   sys.stdout.write(caracter)
   sys.stdout.flush()
print "\n---\nMensaje recibido."
print "¡Que tenga un buen dia!"
```

CAPÍTULO III

EVALUACIÓN: INTEGRACIÓN DE LA ESTEGANOGRAFÍA AL PROTOCOLO HTTP

III.- INTRODUCCION

El desarrollo de este capítulo se realiza con información de cada uno de los casos que presenta la interacción entre la esteganografía y el protocolo HTTP como pueden ser imágenes, archivos de audio, servicio de correo electrónico, entre otros.

III.1 ESTEGANOGRAFÍA EN EL PROTOCOLO HTTP

Debido a que el protocolo HTTP funciona en un esquema cliente/servidor, esta característica se puede utilizar para que sobre él vayan otros datos que no son parte del protocolo como son documentos, imágenes, archivos de audio o video sin que estos datos afecten la información del protocolo.

Cabe aclarar que el protocolo HTTP funciona normalmente cuando se implementa un sistema que proporcione este paso de información.

Como los datos introducidos pueden ser de diferentes tamaños puede llegar a modificar la información del objeto portador, por ello se deben analizar qué datos son modificables, y cuáles valores son los válidos para esta información, así como cuál es el mejor medio para insertar la información, pues uno de los objetivos de la Esteganografía, es que al realizar el paso de la información, este no sea detectable por ningún dispositivo de la red como un firewall o un IDS, ni por un administrador de red que analice el tráfico colocando un sniffer en un segmento de red por el cual transiten los datos introducidos por lo que es necesario tener presente la información que se presenta a continuación en la tabla 3.1.

Tabla 3.1 Tabla de análisis de datos a enviar mediante esteganografía.

Tipo de información	Características
Documentos	Se puede enviar cualquier tipo de documento en cualquier tipo de objeto portador, ya que al no ser de gran peso no modifica las características del objeto portador.
Imágenes	Al igual que los documentos son fáciles de enviar insertadas en cualquier medio ya que su peso no es muy grande.
Audio	En este caso se debe de buscar un objeto portador de un tamaño mayor al de archivo de audio para que no se sufra ningún tipo de alteración en el objeto portador.
Video	Al igual que nos los archivos de audio, los archivos de video se deben de insertar en objetos cuyo tamaña se mas grande, de lo contrario si se sufrían modificaciones en el objeto portador
Código fuente	Un código fuente de cualquier tipo se puede enviar incrustado en otro código sin ningún tipo de problema, sin importar el tamaño, ya que se insertara en forma de comentario, y estos se sabe no tienen ningún tipo de efecto al momento de compilar y ejecutar.

III.2 Aplicación de la esteganografía en imágenes.

Una de las formas para ocultar la información por medio de la esteganografía es el uso de imágenes, esto se debe realizar sin afectar las propiedades de la imagen.

Para este primer caso se genera una carpeta donde se aloja la imagen que va a servir para ocultar la información y los documentos, los cuales ya se encuentran comprimidos. (Véase la figura 3.1)

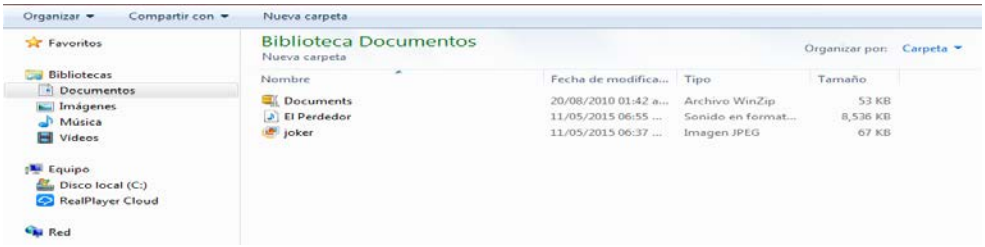

Fig. 3.1 Carpeta donde se encuentra los documentos a ocultar y la imagen portadora.

Después, desde el símbolo de sistemas se ingresa a la carpeta que contiene la imagen y los documentos. (Véase la figura 3.2)

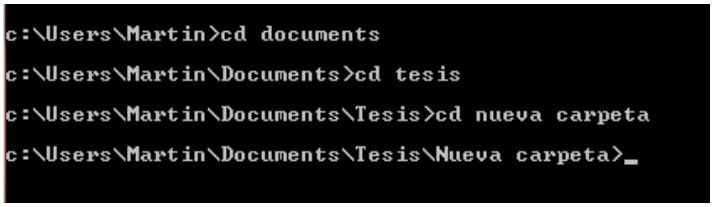

Fig. 3.2 Acceso a la carpeta desde el símbolo del sistema.

Como se puede observar en la figura 3.3 la imagen que se va a utilizar tiene un peso de 67 KB y con el editor hexadecimal se puede observar que la cola de la imagen termina en FF D9 y después ya no hay nada de información por lo que no tiene ningún tipo de información oculta como se muestra en la figura 3.4.

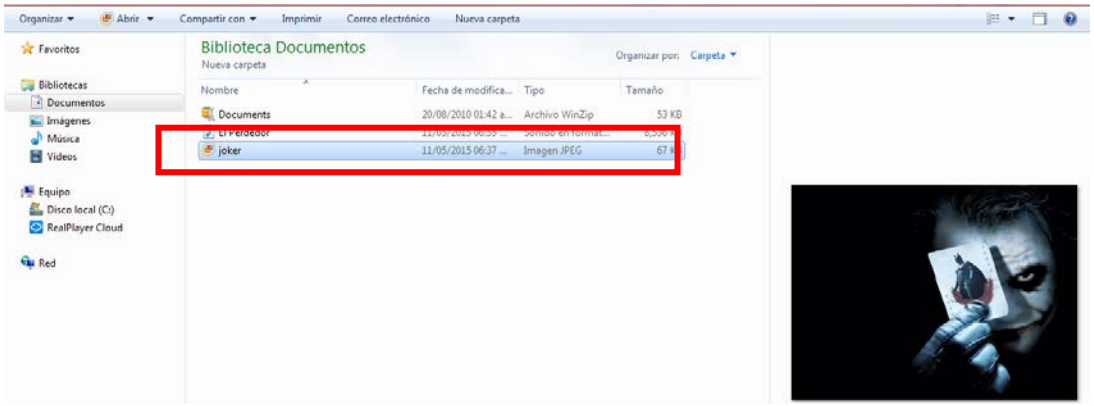

Fig. 3.3 Validación del peso de la imagen portadora

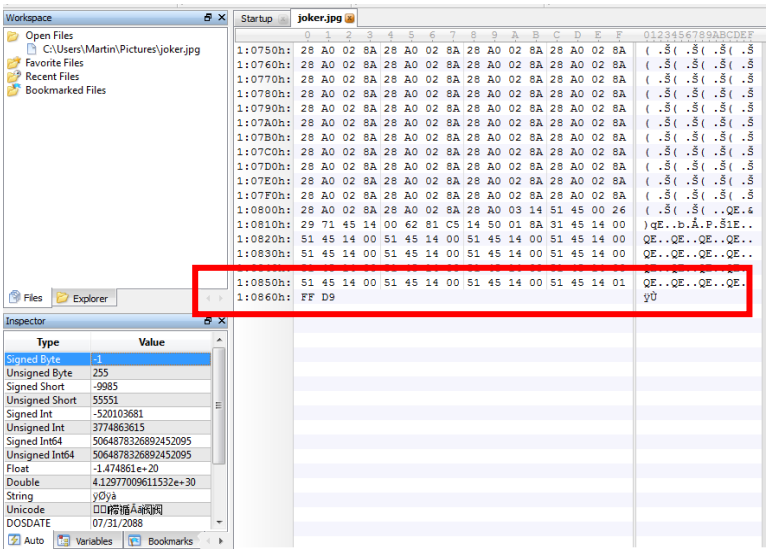

Fig. 3.4 Cola de la imagen mediante el editor hexadecimal.

Regresando a la pantalla del símbolo del sistema, y estando en la carpeta se coloca el comando **copy /b joker.jpg + documents.rar resultado.jpg**. (Véase figura 3.5)

Lo que este comando realizará será una suma binaria entre los valores hexadecimales de la imagen y los documentos comprimidos y alojará los documentos comprimidos dentro de la imagen.

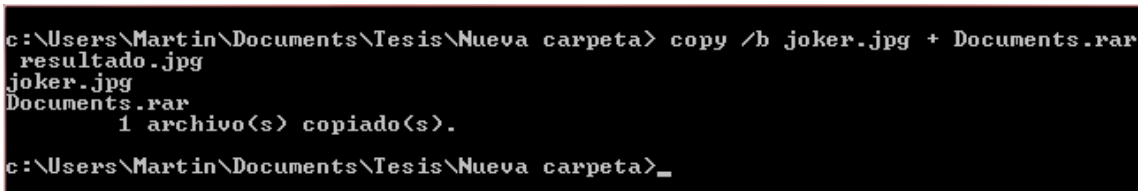

Fig. 3.5 Suma binaria de los documentos.

Como se puede ver en la figura 3.6 se copió un archivo y dentro de la carpeta ya se encuentra la nueva imagen con la información insertada, y si se comparan las imágenes, sin archivo insertado y con archivo insertado como se muestra en la figura 3.7 se puede apreciar que visualmente las imágenes son exactamente iguales, a simple vista podría decirse que la imagen no sufrió ninguna modificación, sin embargo, al revisar la carpeta, se observa que la diferencia está en el peso de la imagen, el cual incrementó de 67 KB a 119 KB.(Ver figura 3.6)

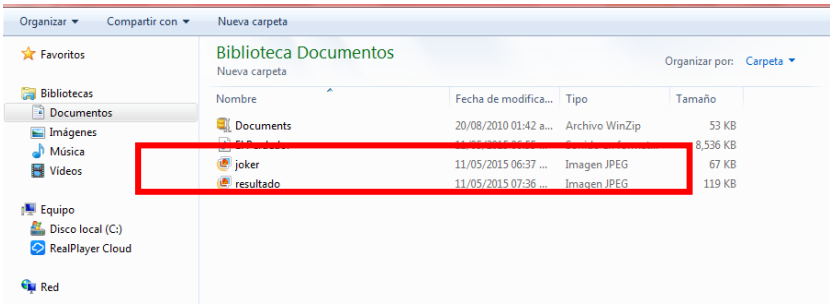

Fig. 3.6 Nueva imagen en la carpeta.

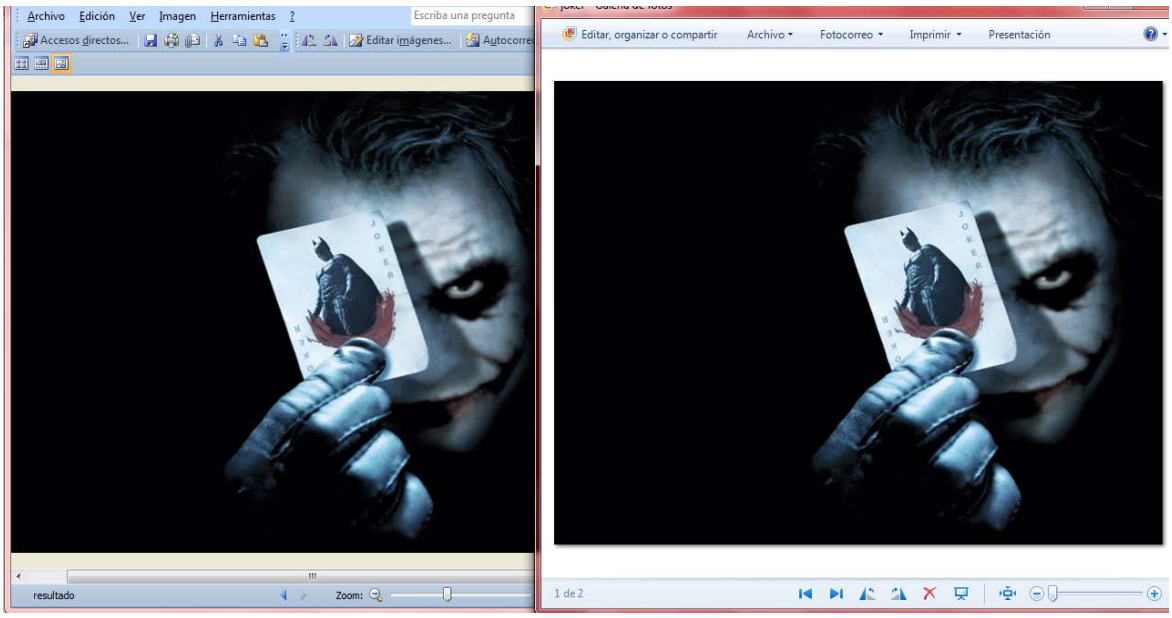

Fig. 3.7 Comparación de las dos imágenes.

Por medio del editor hexadecimal es posible comprobar que la imagen tiene información ya que después de su cola la cual terminaba en FF D9, además de que se podrá observar la palabra .rar en donde comienza la nueva información como se aprecia en la figura 3.8.

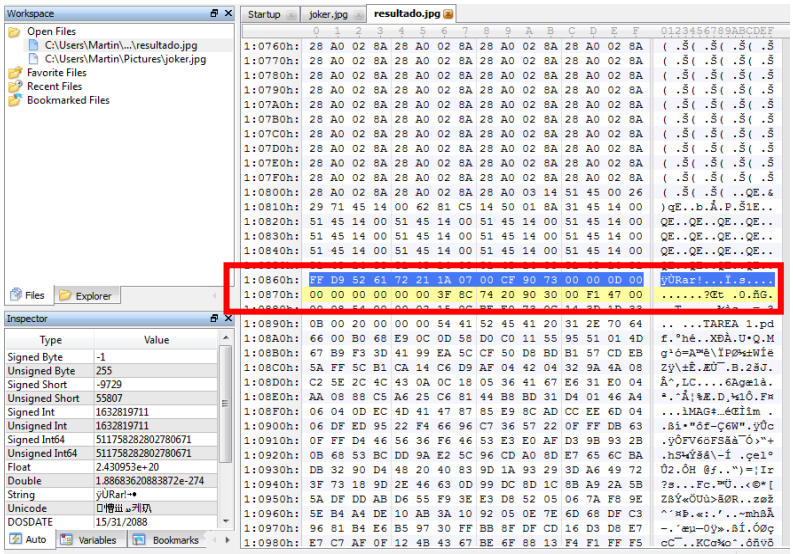

Fig. 3.8 Se observa la información insertada después de la cola de la imagen.

Para terminar se envía la imagen vía correo electrónico a la persona receptora (figuras 3.9 y 3.10), una vez que se recibe con el editor hexadecimal se copia la nueva información y se almacena en un documento nuevo y así se podrá ver la información enviada mediante la imagen, como se muestra en las figuras 3.11, 3.12 y 3.13.

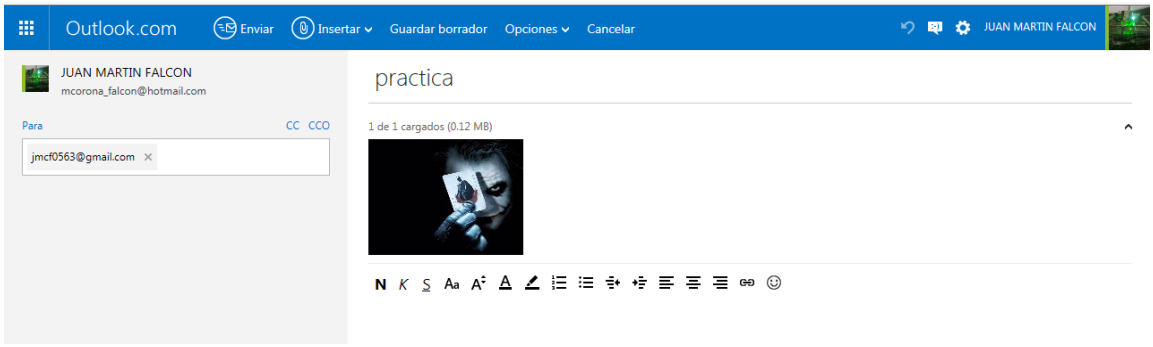

Fig. 3.9 Envío de la imagen portadora mediante servicio de Hotmail.

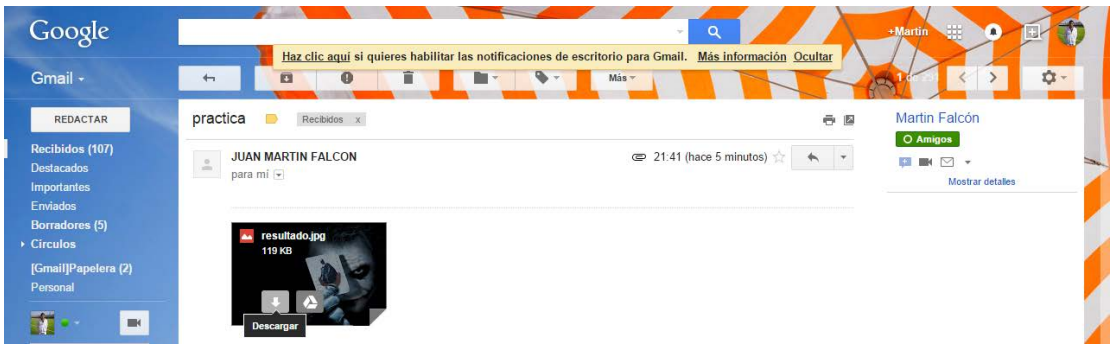

Fig. 3.10 Recepción de la imagen portadora mediante servicio de Gmail.

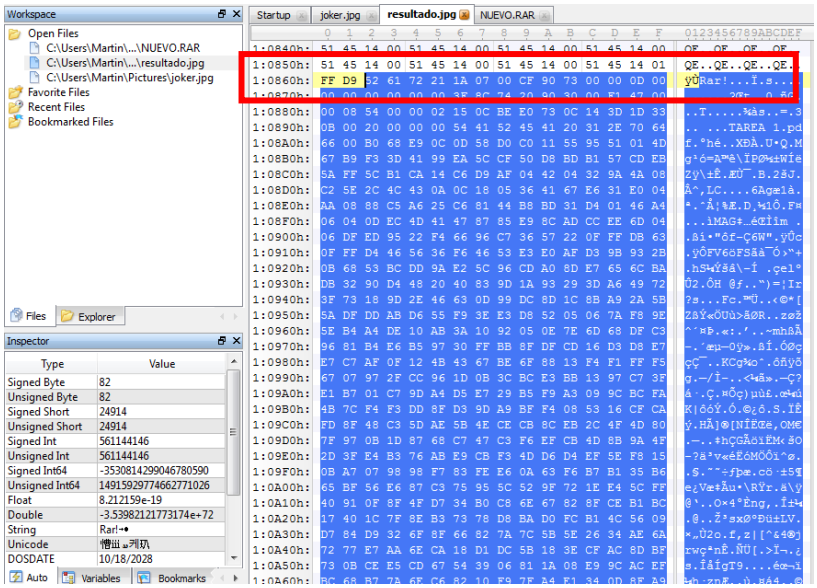

Fig. 3.11 Selección de la información oculta en la imagen.

Fig. 3.12 Se abre un nuevo archivo y se pega la información seleccionada y se guarda como .rar.

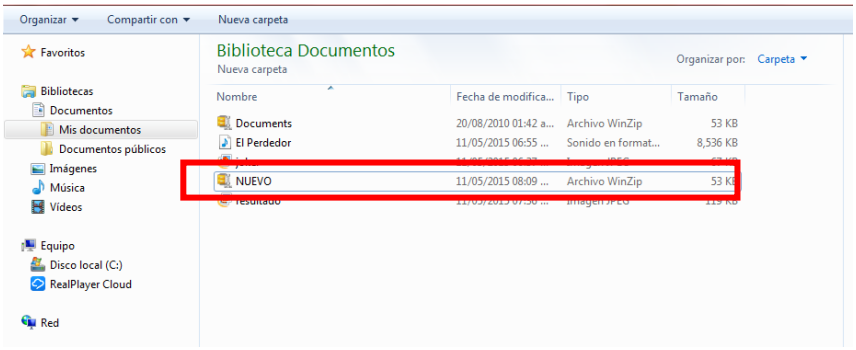

Fig. 3.13 Se observa en la carpeta un nuevo documento .rar el cual es la información que se extrajo de la imagen portadora.

En la prueba anterior se insertó información comprimida dentro de una imagen, ahora se va a insertar una imagen y un archivo de audio dentro de otra imagen.

Se usará una nueva imagen como portadora la cual tiene por nombre “tigre” y cuenta con un peso de 135 Kb como se muestra en la figura 3.14.

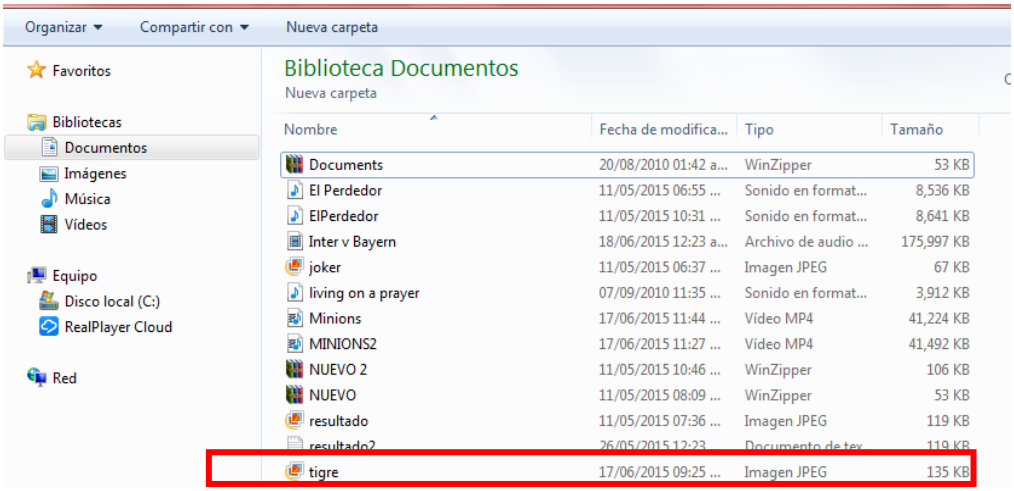

Fig. 3.14 Nueva imagen portadora.

Como se observa en la figura 3.15 la cola del código hexadecimal de la nueva imagen portadora termina en FF D9 y como ya se ha visto es aquí donde se inserta el código de la imagen que se desea ocultar.

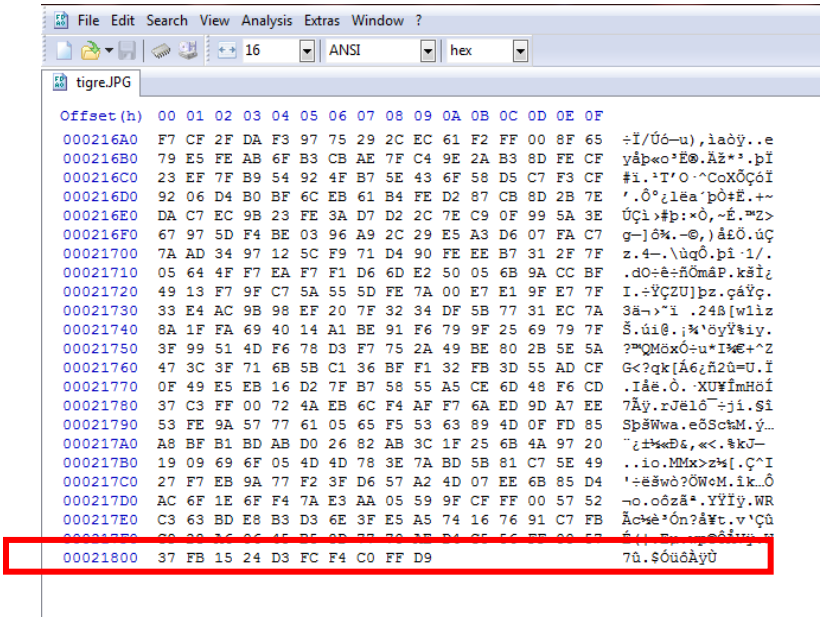

Fig. 3.15 Código hexadecimal de la nueva imagen portadora.

Como se ha realizado anteriormente, se hará la misma suma binaria de los códigos hexadecimales

entre la nueva imagen portadora y la imagen que se desea insertar, en este caso la utilizada en la primera prueba. (Ver Fig. 3.16).

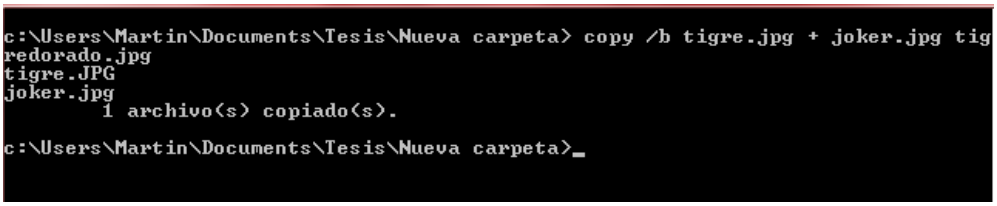

Fig. 3.16 Suma binaria entre ambas imágenes.

Como se observa la suma se realiza correctamente y se genera una nueva imagen con el nombre de "tigredorado" y un peso de 201 Kb, el cual es superior al de la imagen portadora original (135kb), y al revisar el código hexadecimal de la imagen portadora se puede ver como esta insertado el código de la otra imagen. (Ver Figuras 3.17 y 3.18).

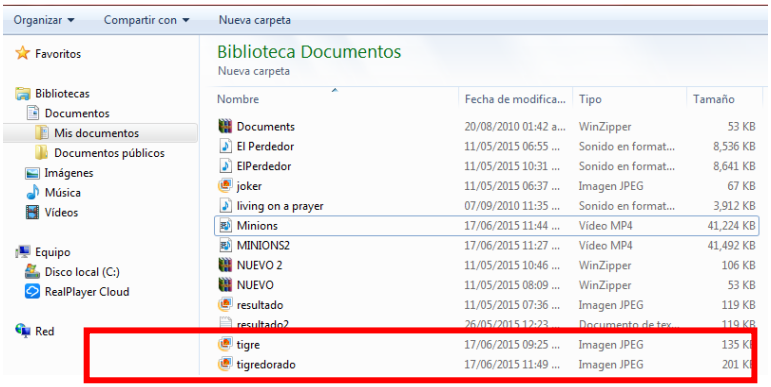

Fig. 3.17 Nueva imagen generada.

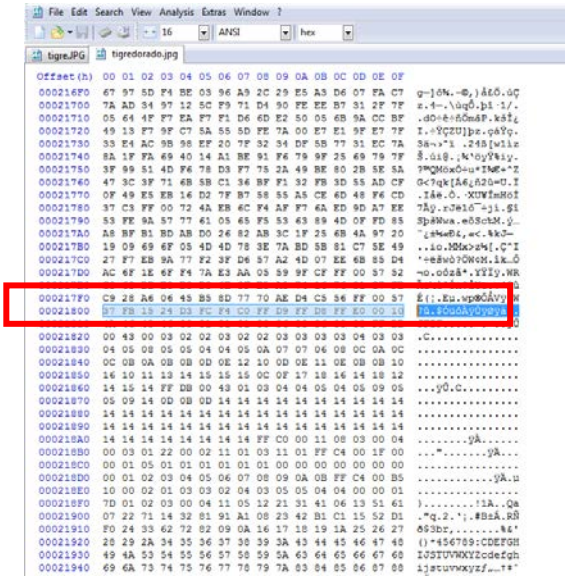

Fig. 3.18 Código hexadecimal incrustado en la imagen portadora.

Al realizar la comparación de las imágenes se puede observar que no sufrió ninguna modificación en

su formato. (Ver figura 3.19).

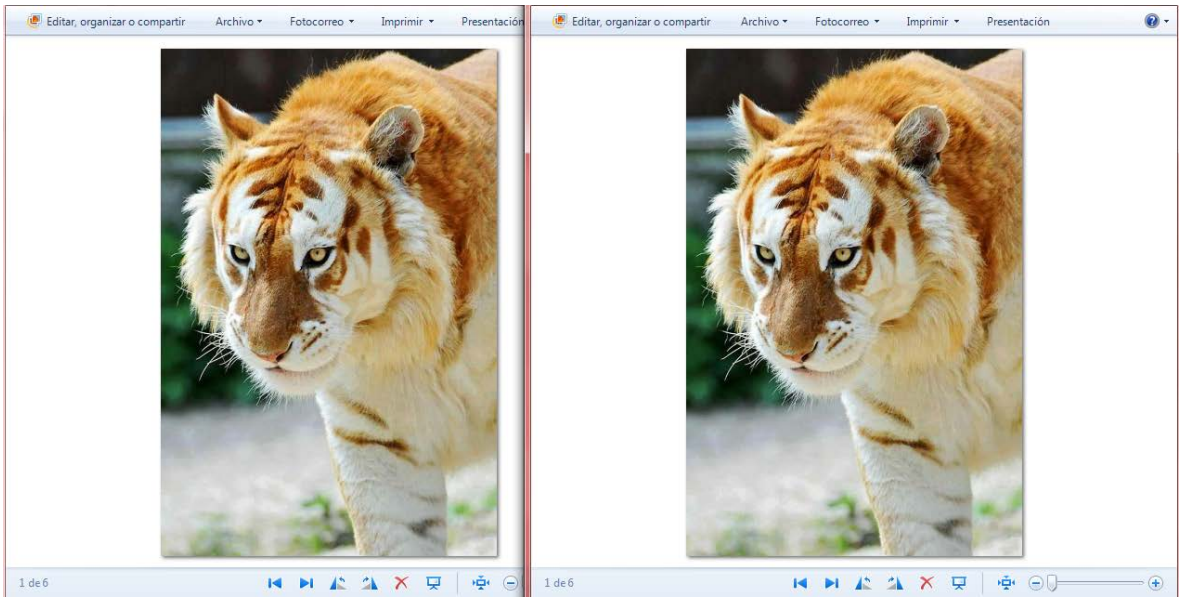

Fig. 3.19 Comparación de la nueva imagen portadora con la imagen insertada y sin insertar.

Para termina con las pruebas de esteganografía en imágenes se intenta insertar un archivo de audio en una imagen mediante el mismo procedimiento de la suma binaria de código hexadecimal. (Ver figura 3.20).

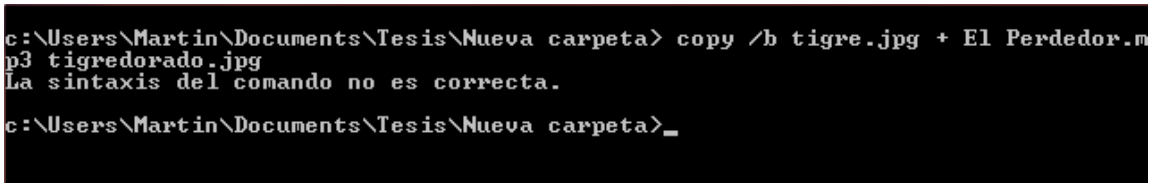

Fig. 3.20 suma binaria de una imagen y un archivo de audio

Como se puede observar en la imagen 3.20 arroja un error de sintaxis en el comando, esto debido a que el tamaño del archivo de audio es mucho mayor al de la imagen y si se realiza esta suma el objeto portador sufrirá modificaciones y con ello el proceso de esteganografía no se cumple.

III.3 Aplicación de la esteganografía en audio.

Como se mencionó anteriormente uno de los métodos de aplicación de la estenografía es el insertar la información en archivos de audio.

Para este caso se va a realizar el miso procedimiento que para insertar información en una imagen. Por medio del símbolo del sistema se ingresa a la carpeta donde se encuentra contenido el archivo de audio por donde se desea enviar la información. En este caso es la misma carpeta que la prueba anterior y se realiza la misma suma binara de los códigos hexadecimales del archivo de audio portador y los documentos ya comprimidos, con el mismo comando. (figura 3.21)

```
c:\Users\Martin\Documents\Tesis\Nueva carpeta> copy /b El Perdedor.mp3 + Documen
ts.rar ElPerdedor.mp3
ElPerdedor.mp3
Documents.rar
        1 archivo(s) copiado(s).

c:\Users\Martin\Documents\Tesis\Nueva carpeta>_
```

Fig. 3.21 Suma binaria de los documentos.

Se puede observar que se generó otro archivo, donde la diferencia es el nombre de El Perdedor.mp3 a ElPerdedor.mp3 y el tamaño de cada uno de 8536 kb a 8641kb como se observa en la figura 3.22.

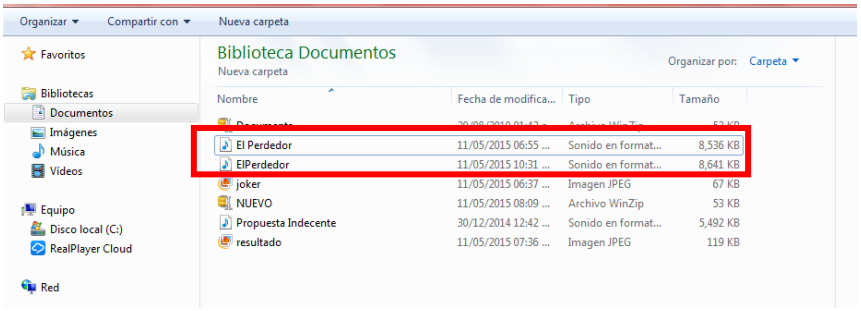

Fig. 3.22 Se observa el nuevo archivo de audio con la información oculta en él.

Si se comparan los archivos con el editor hexadecimal se puede identificar la cola del archivo de audio sin información alguna, (figura 3.23) y al igual que en la imagen, la información se colocó después de la cola del archivo de audio y se identifica con la palabra .rar. (Ver figura 3.24).

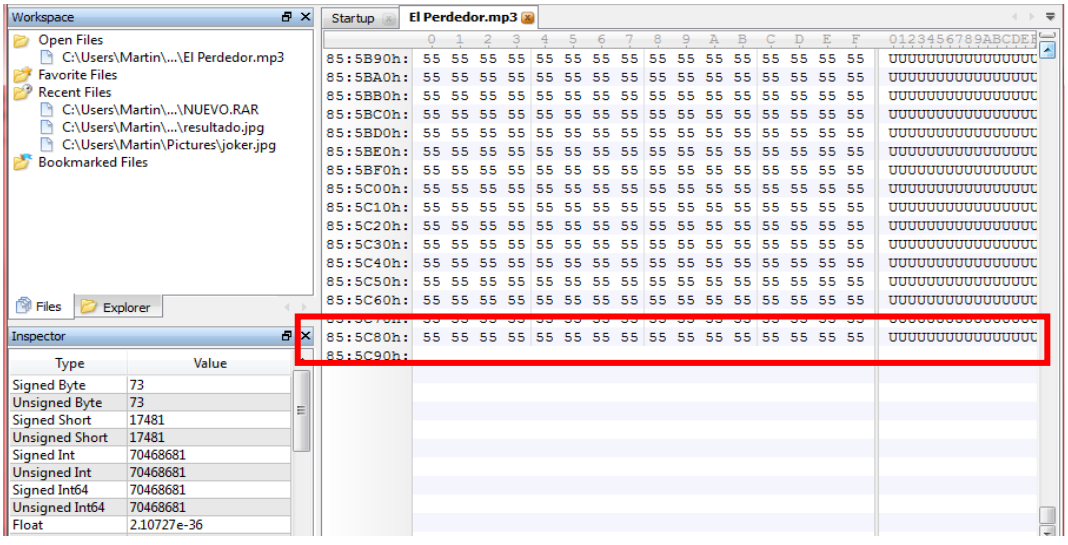

Fig. 3.23 Código hexadecimal de la canción sin información oculta.

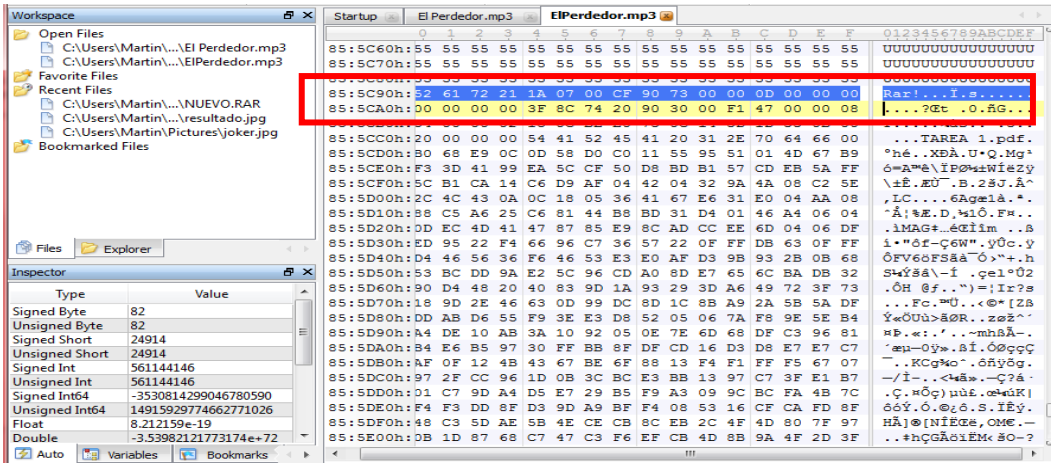

Fig. 3.24 Código hexadecimal de la canción con información oculta.

De igual forma que con la imagen, se envía por correo electrónico a la persona receptora (figuras 3.25 y 3.26) quien al recibirla la abre con el editor hexadecimal y se copia la información en un documento nuevo y así se obtiene la información que se ocultó en el archivo de audio.(Figuras 3.27, 3.28 y 3.29).

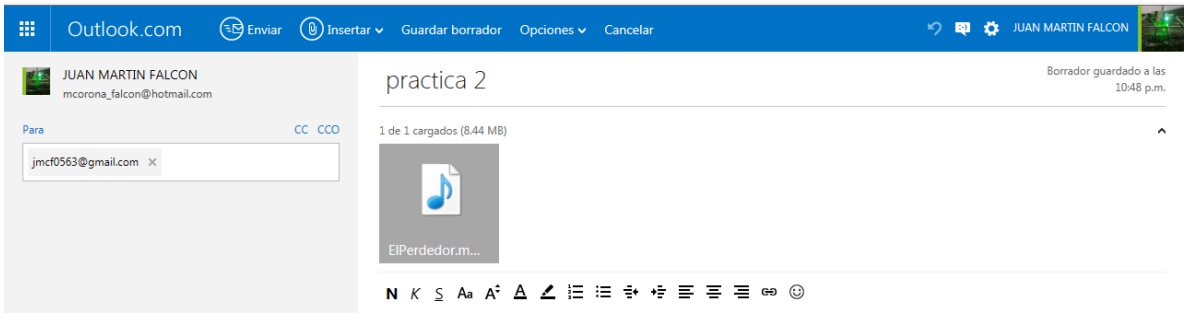

Fig. 3.25 Envío de la canción portadora mediante servicio de Hotmail

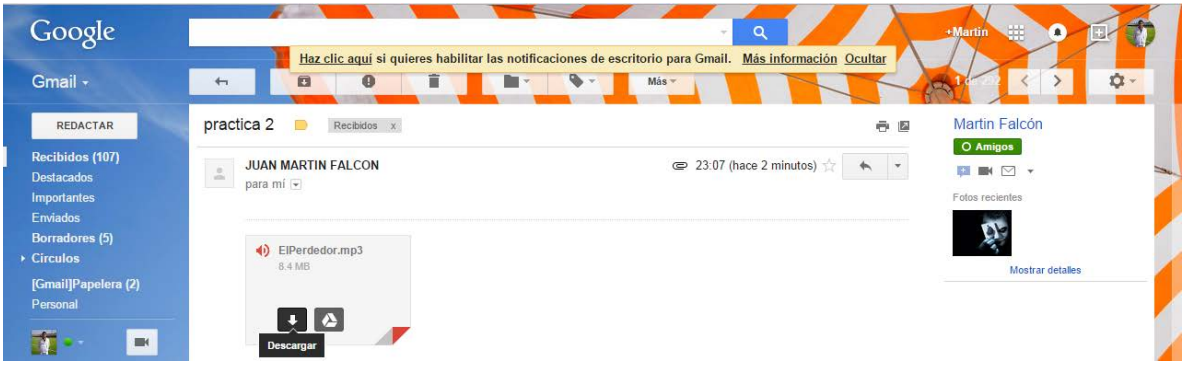

Fig. 3.26 Recepción de la canción portadora mediante servicio de Hotmail

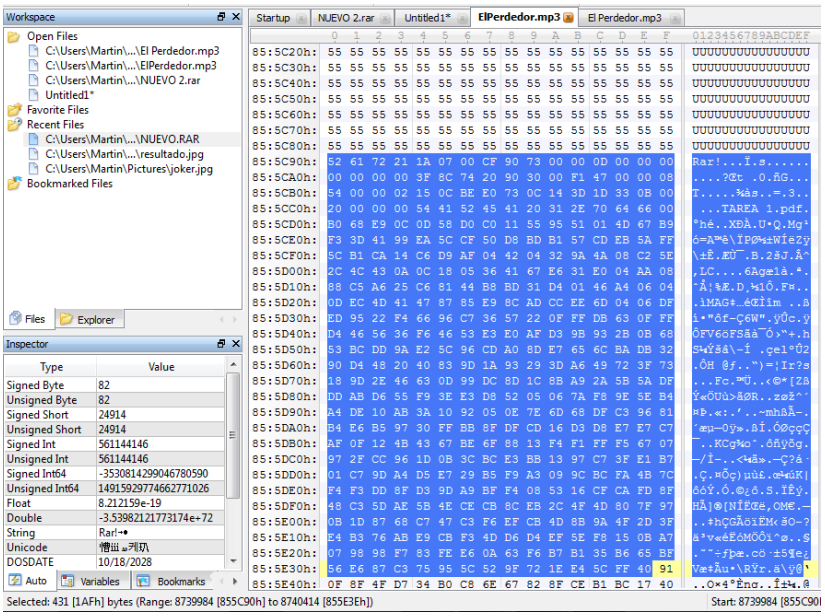

Fig. 3.27 Selección de la información oculta en la imagen.

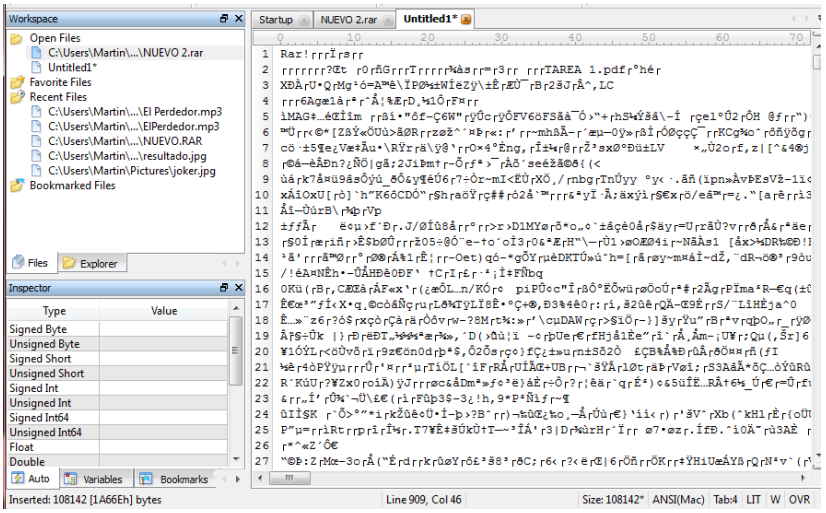

Fig. 3.28 Se abre un nuevo archivo y se pega la información seleccionada y se guarda como .rar.

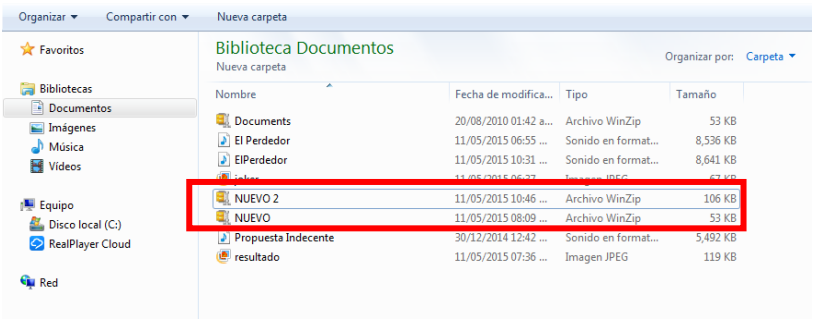

Fig. 3.29 Se observa en la carpeta un nuevo documento .rar el cual es la información que extrajimos de la imagen portadora.

Al igual que con la imagen lo que se busca es ocultar otro tipo de información en el archivo de audio,

la cual será otro archivo de audio y un video.

Como se muestra en las imágenes 2.30 y 2.31 al realizarse la suma binaria se genera el mismo error de sintaxis, por lo cual no es factible ocultar estos dos tipos de archivo debido al peso que tienen.

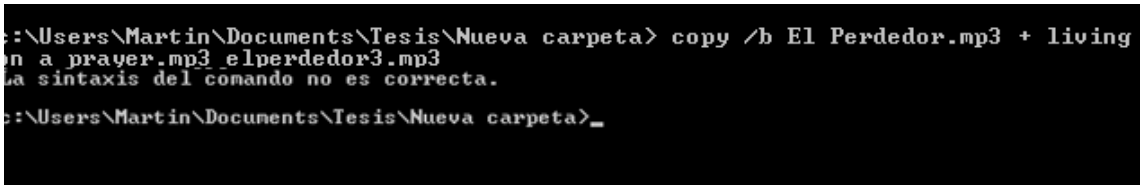

Fig. 3.30 Suma binaria de dos archivos de audio, "El perdedor.mp3 y living on a prayer.mp3".

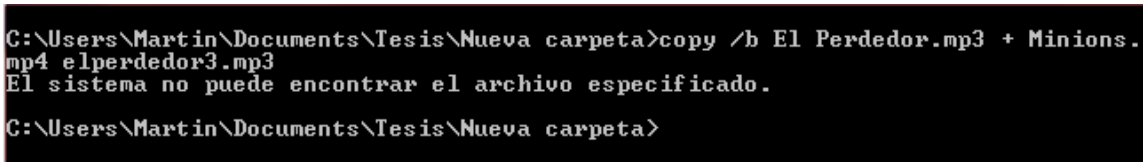

Fig. 3.31 Suma binaria de un archivo de audio y un video, "El perdedor.mp3 Minions.mp4".

III.4 Aplicación de la esteganografía en video.

Otro método de ocultamiento de la esteganografía es por medio de archivos de video, para esta prueba se toma un video con el nombre de "Minions" y un peso de 41,224 Kb. (véase figura 3.32)

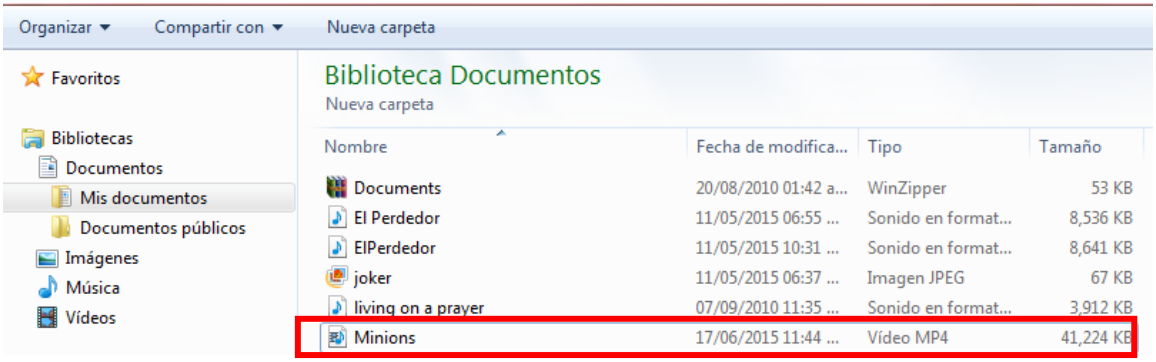

Fig. 3.32 Video portador.

Mediante el editor de código hexadecimal se puede ver que su cola de código termina en .100. (Ver figura 3.33).

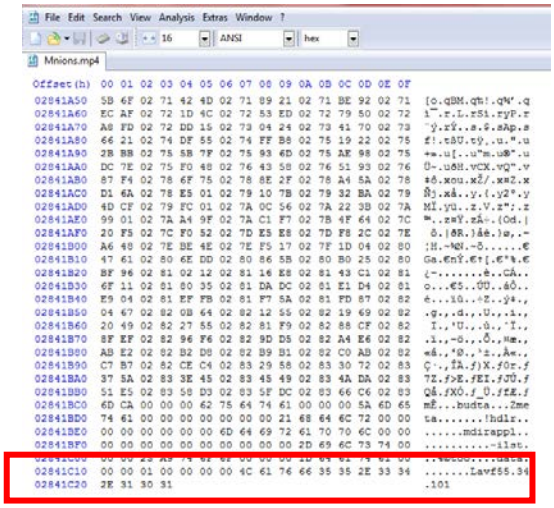

Fig. 3.33 Cola del código hexadecimal del video portador.

Después de realizar la suma binaria entre el archivo de video (Minions.mp4) y la imagen (tigre.jpg) se observa un nuevo archivo de video con un peso de 41,492 Kb. (Ver figuras 3.34 y 3.35).

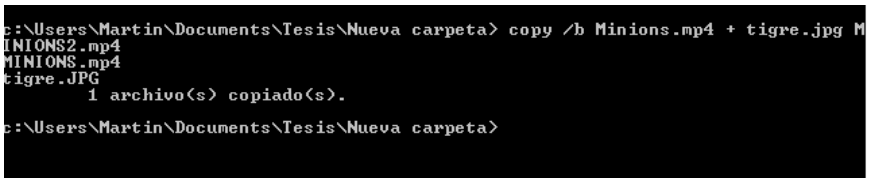

Fig. 3.34 Suma binaria de archivo de video e imagen.

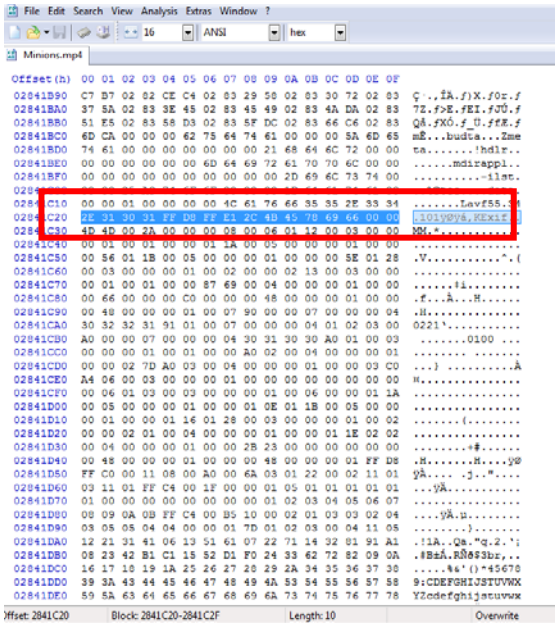

Fig. 3.35 Código hexadecimal con imagen incrustada.

Por último se inserta un archivo de video y un archivo de audio en el video portador mediante el comando de una suma binaria hexadecimal.

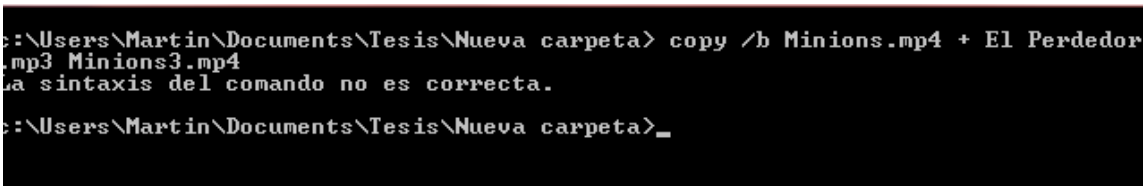

Fig. 3.36 Suma binaria de archivo de video y audio.

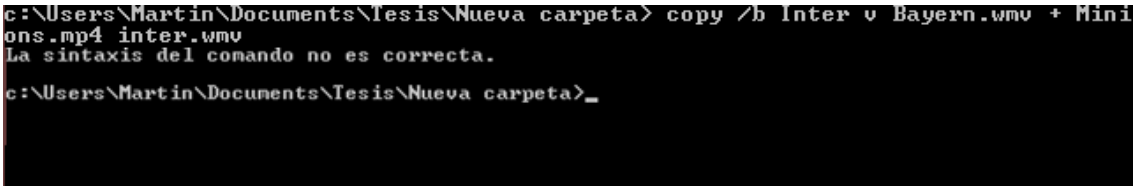

Fig. 3.37 Suma binaria de dos archivos de video.

Como se puede observar en las figuras 3.36 y 3.37 se genera el mismo error de sintaxis debido al peso de los archivos que se desean ocultar ya que como se había mencionado anteriormente modificarían al objeto portador.

III.5 Aplicación de la esteganografía en el protocolo http.

En las pruebas anteriores se utilizó el protocolo http como un medio de envió, el cual fue el servicio de correo electrónico y se observó que ni los objetos portadores ni la información enviada sufrió algún tipo de cambio y llego integra a su destino.

En este caso el medio portador es el propio protocolo, insertando información en el código de una página web. En este caso la página de inicio de Yahoo. (figura 3.38).

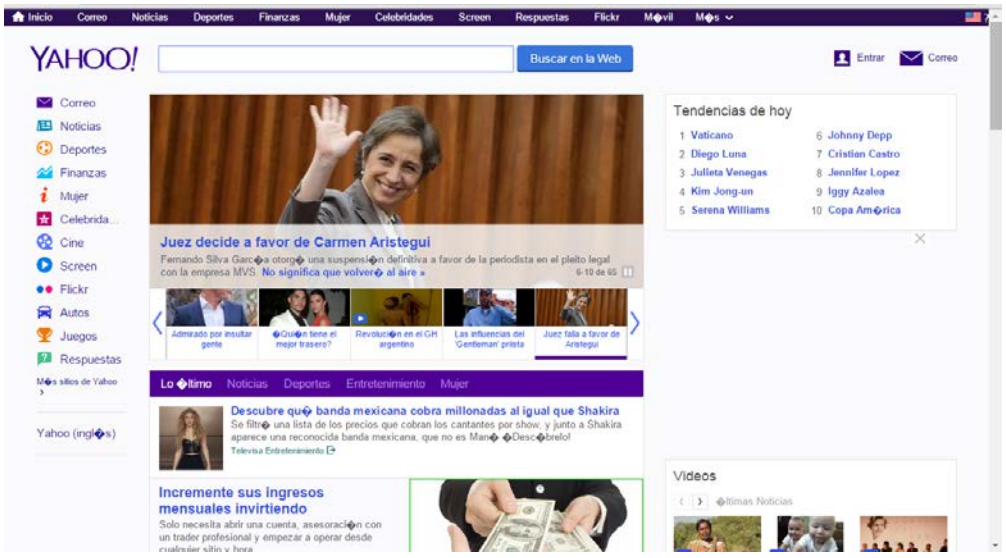

Fig. 3.38 Página principal de Yahoo.

De esta página se va a utilizar su código fuente, que es donde se colocará la información oculta. (véase figura 3.39).

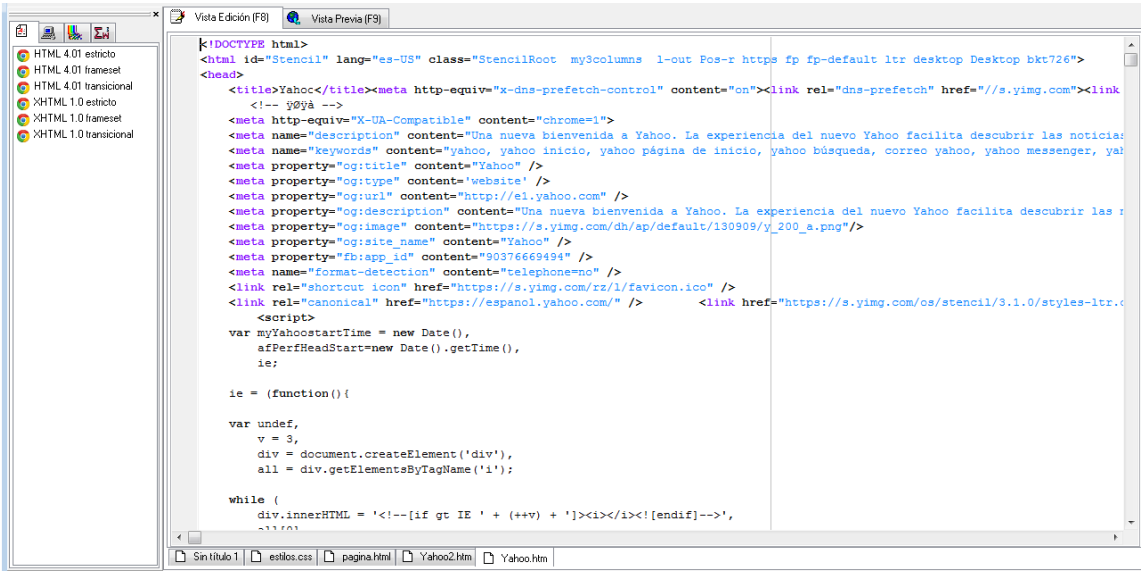

Fig. 3.39 Código fuente de la página de Yahoo.

Se toma el código hexadecimal de la imagen que se usó como objeto portador en la primera prueba y se coloca en el *"head"* del código de la página web. (figura 3.40).

Al final del código hexadecimal se colocó el formato de la información oculta, en este caso están las letras JPG, lo cual hace referencia a que es una imagen la cual esta oculta como se muestra en la figura 3.41.

Fig. 3.40 Código hexadecimal insertado en el código fuente de la página de Yahoo

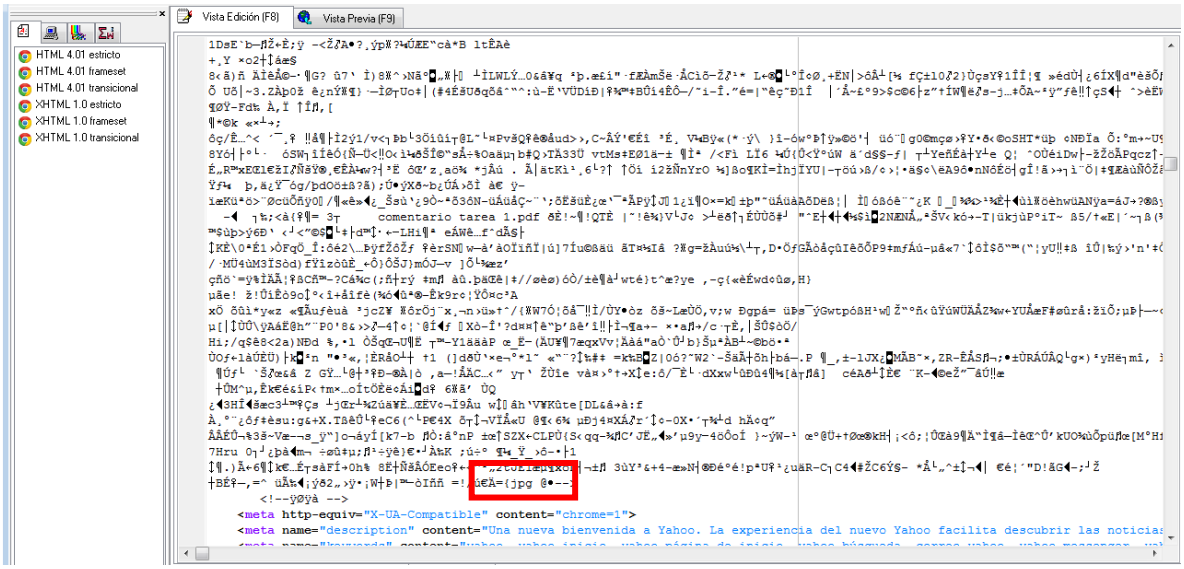

Fig. 3.41 Código hexadecimal insertado en el código fuente de la página de Yahoo y al final se observa el formato de la información oculta, JPG.

Al ejecutar el código fuente de la página se observa que la página se carga de manera correcta y sin ningún tipo de modificación. (figura 3.42).

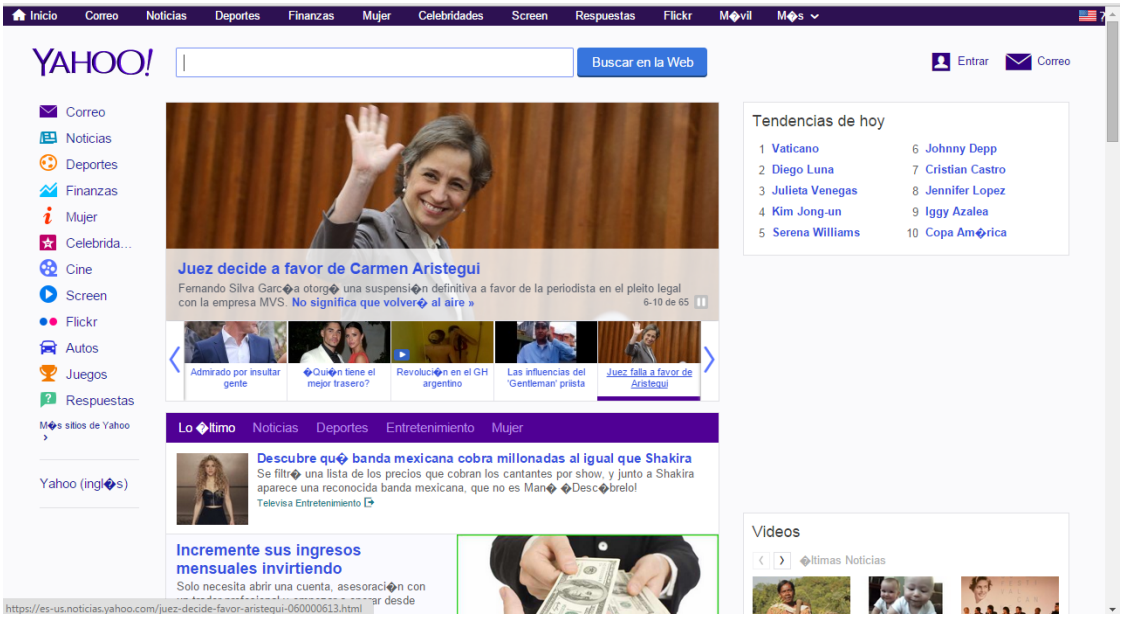

Fig. 3.42 Ejecución de la página web teniendo ya insertada la imagen en su código fuente.

Si se accede al código fuente se puede ver como se encuentra incrustado el código hexadecimal de la imagen oculta en la cabecera del código fuente de la página web. (Ver figuras 3.43, 3.44 y 3.45)

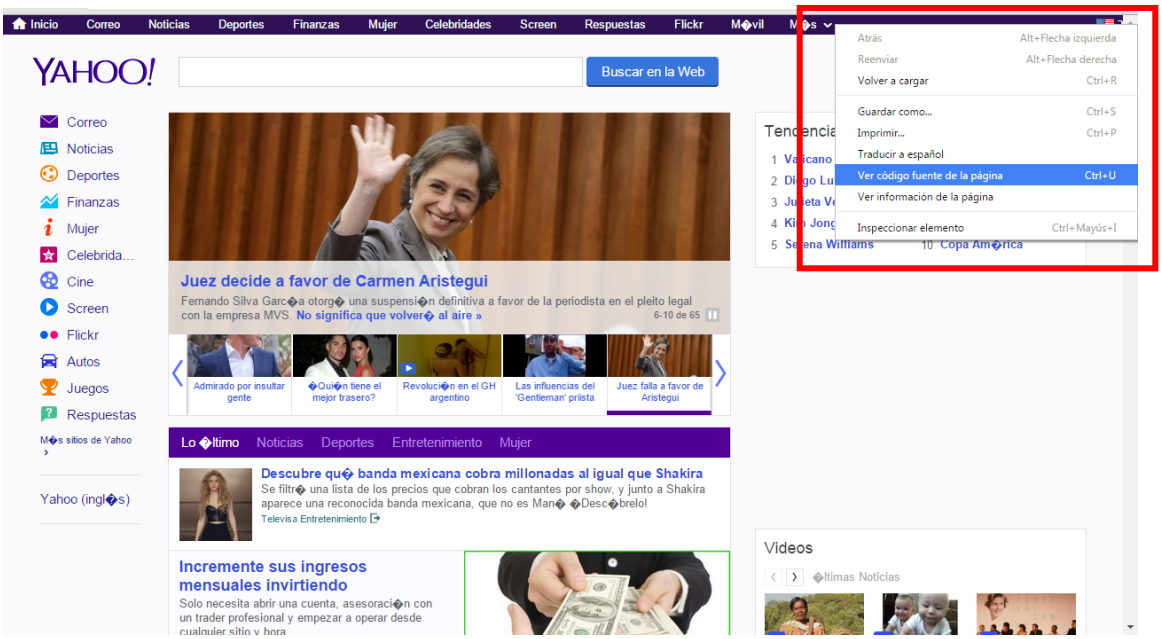

Fig. 3.43 Acceso al código fuente de la página web para validar que se encuentra insertada la imagen en su código.

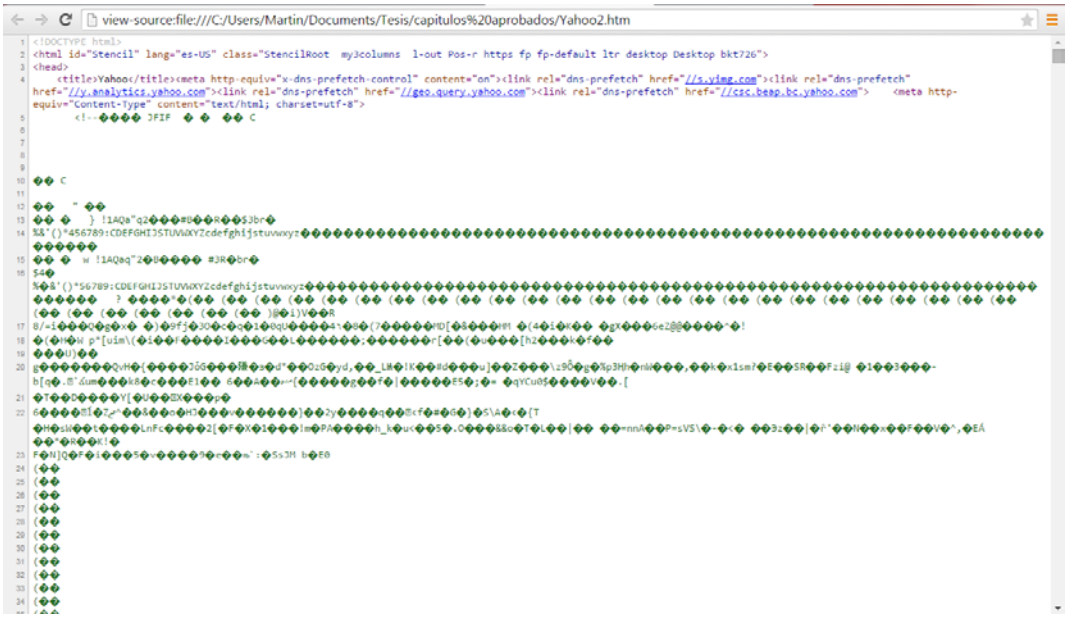

Fig.3.44 Inicio del código hexadecimal de la imagen insertado en el la cabecera del código fuente de la página web

Fig. 3.45 Fin del código hexadecimal de la imagen insertado en el la cabecera del código fuente de la página web

Ahora se toma el código hexadecimal de un archivo de audio (living on a prayer) mediante el editor hexadecimal y se incrusta de igual forma en el código html como se realizó con la imagen para ver si sufre alguna modificación. (ver figuras 3.46 y 3.47).

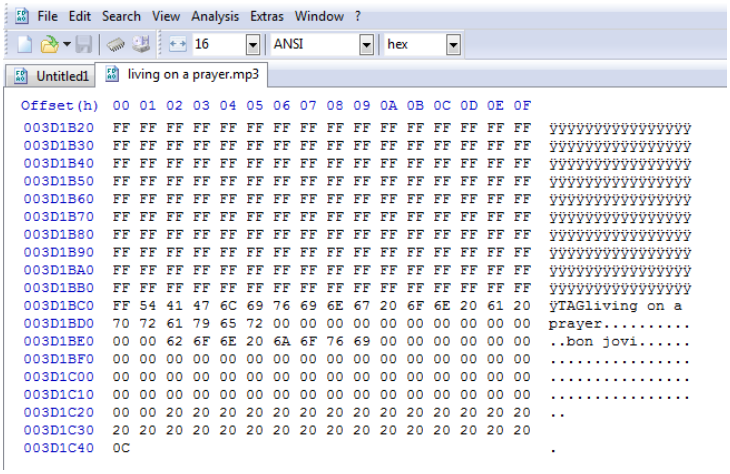

Fig. 3.46 Código hexadecimal del archivo de audio que se va a insertar.

Fig. 3.47 Página principal de Yahoo.

En esta ocasión el código se insertara en el boody del código html, para comparar si hay una diferencia entre hacerle en header o en el boody. (ver figura 3.48).

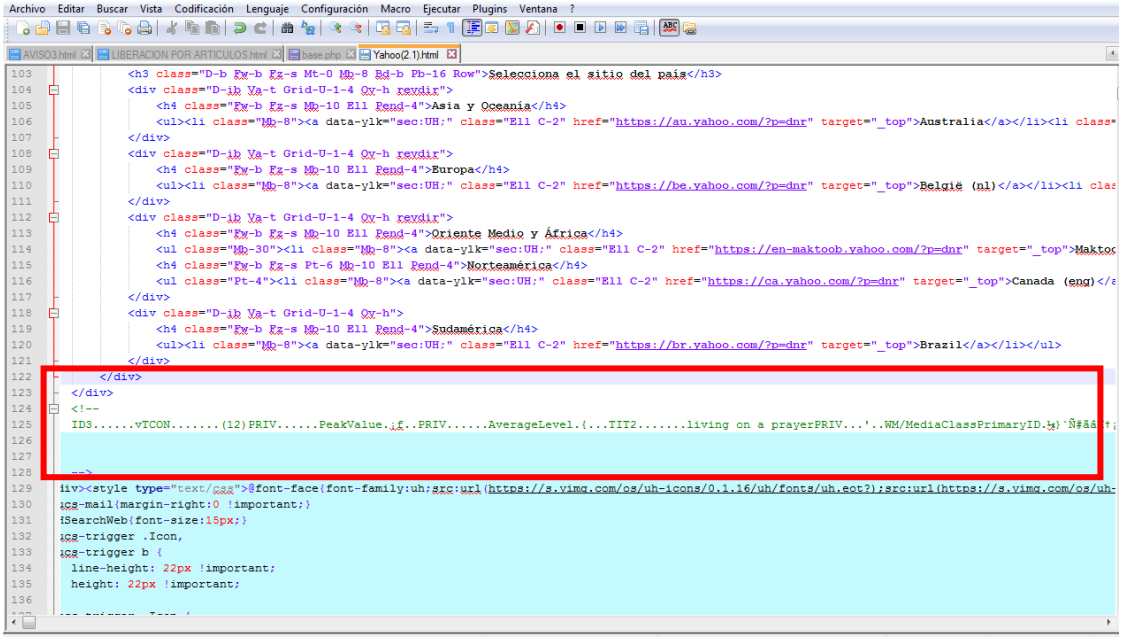

Fig. 3.48 Archivo de audio inserta en el boody del código html.

Después de haber insertado el código hexadecimal del archivo de audio en el código html de la página de Yahoo, se ejecuta y se observa que la página no sufre ningún tipo de modificación y se ejecuta con normalidad. (ver figura 3.49).

Fig. 3.49 Página de Yahoo ejecutad con el código del archivo de audio insertado.

Si se accede al código fuente de la página html se observa el código del archivo de audio insertado como lo muestran las figuras 3.50, 3.51 y 3.52.

Fig. 3.50 Acceso al código fuente de la página de Yahoo.

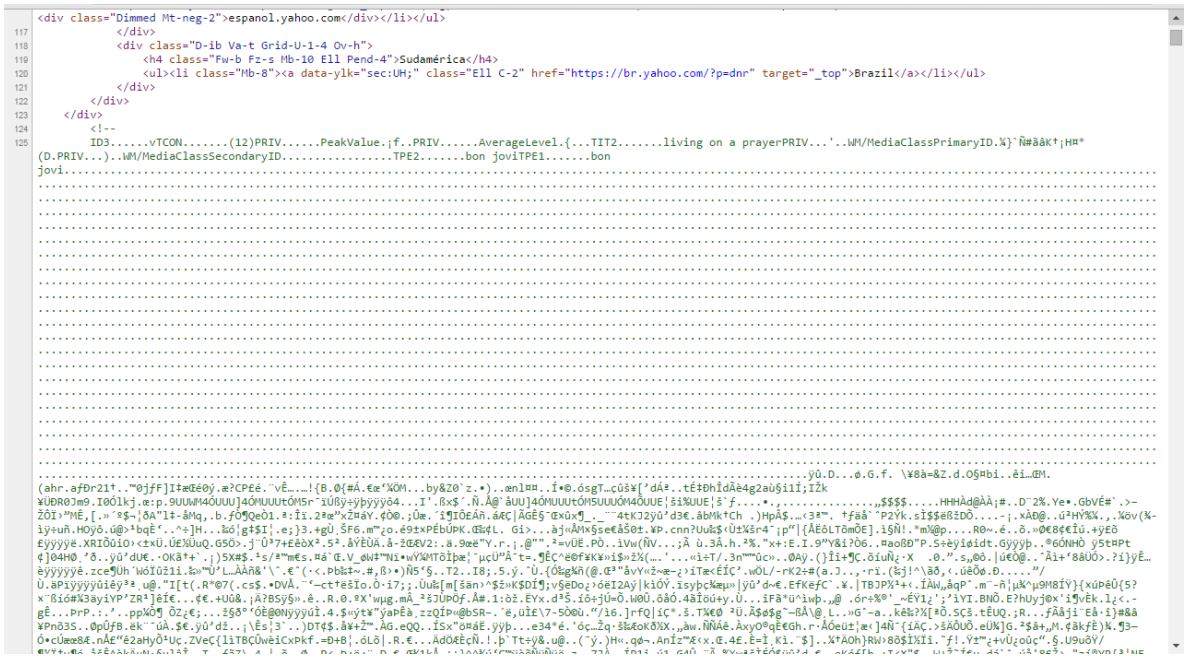

Fig. 3.51 Inicio del código hexadecimal del archivo de audio contenido dentro del código html.

Fig. 3.52 Fin del código hexadecimal del archivo de audio contenido dentro del código html.

Con las pruebas realizadas se comprobó de manera satisfactoria la inserción de una imagen y de un archivo de audio dentro del código html de una página web sin ningún tipo de modificación en ella. A continuación se examina si sucede lo mismo en cualquier otro tipo de código de programación.

Se toma el código fuente de c++ de un programa que calcula la ley de Ohm, en las figuras 3.53 y 3.54 se puede ver el código fuente del programa y el programa en ejecución.

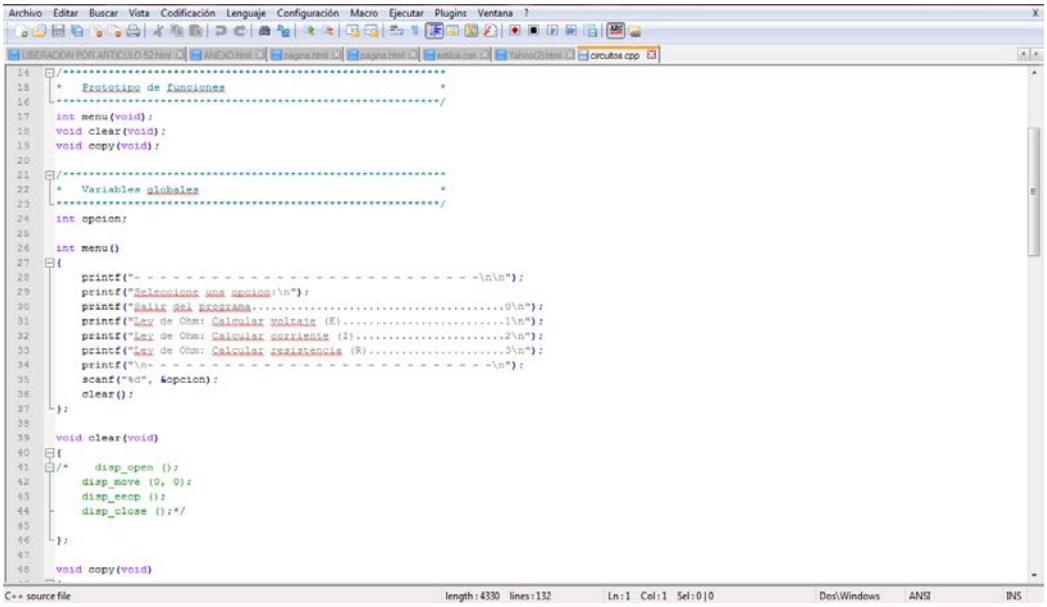

Fig. 3.53 Código fuente en C++ de un programa que calcula la ley de Ohm.

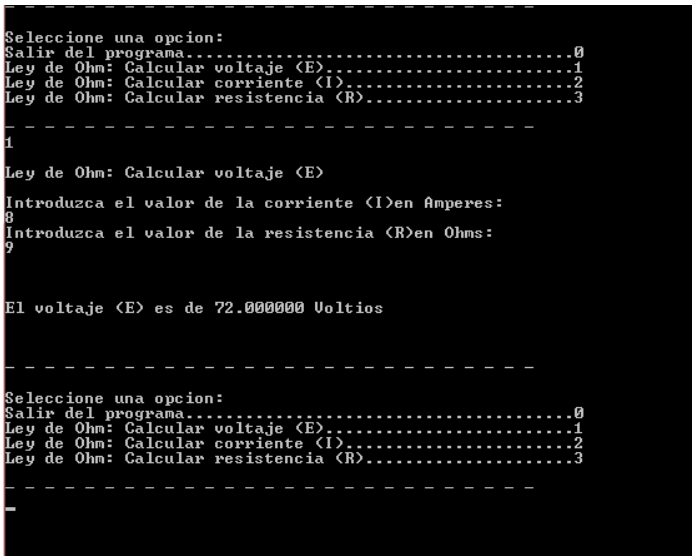

Fig. 3.54 Programa que calcula la ley de Ohm en ejecución.

Se toma el código hexadecimal de la imagen “tigre” y se coloca como un comentario en una sección del código fuente del programa (ver figura 3.55), se observa que se compila sin ningún tipo de error, se ejecuta y el programa función de manera normal (ver figuras 3.56 y 3.57).

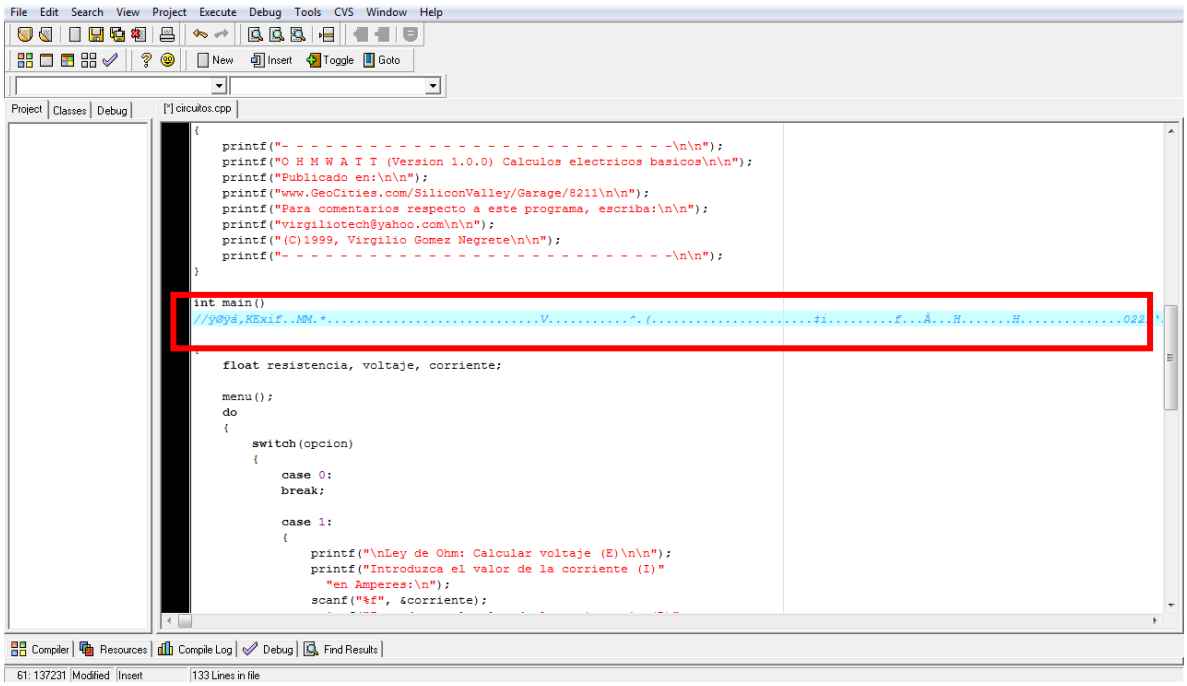

Fig. 3.55 Código hexadecimal de la imagen "tigre" en el código C++

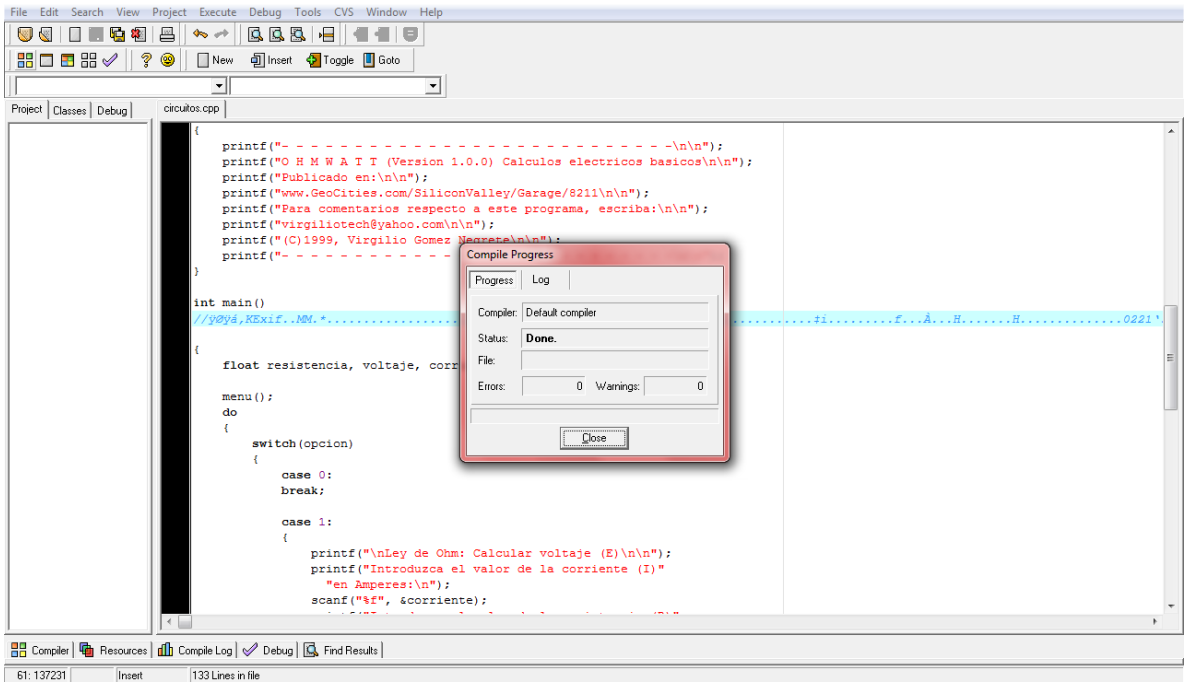

Fig. 3.56 Compilación exitosa del código C++ con el código hexadecimal incristado.

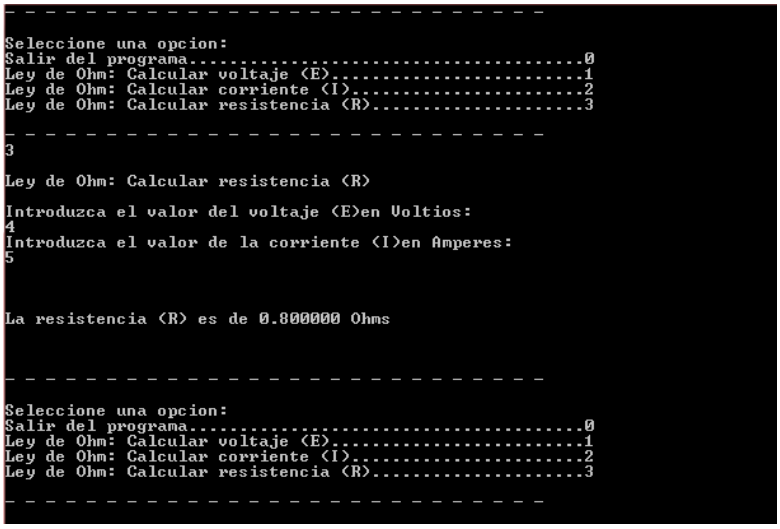

Fig. 3.57 Ejecución normal del programa.

Por último se inserta el codigo hexadecimal de un archivo de audio en el código c++ para comprobar que tampoco sufre nungún tipo de modificación el programa y demostrar que se puede incrustar cualquier tipo de informacion en un código fuente.

Para esta prueba se utiliza el código hexadecimal del archivo de audio “living on a pryer”, se inserta en el códifo c++, se compila y se observa que no presenta ningun error. (ver figura 3.58).

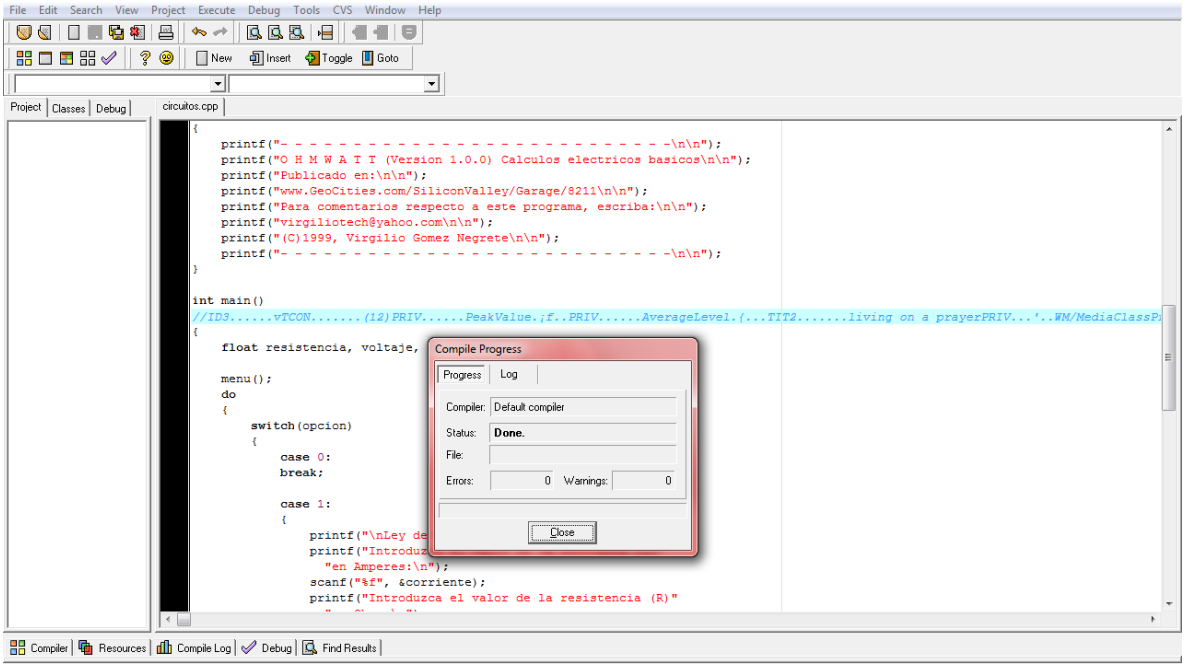

Fig. 3.58 Compilación del código C++ con el código hexadecimal del archivo de audio insertado.

Si se vuelve a ejecutar el programa se observa que corre sin ningún tipo de problema y funciona de manera normal. (Ver figura 3.59).

Fig. 3.59 Ejecución normal del programa.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones y recomendaciones.

Como se pudo observar en las pruebas realizadas, el método de la suma binaria entre el objeto portador y de la información a ocultar resultó satisfactoria, ya que en ningún momento se vieron afectadas las propiedades del objeto portador, ya fuera la imagen, el archivo de audio o video.

Con base en los resultados obtenidos se muestra una tabla comparativa de la efectividad de los medios portadores según el tipo de información que se desea enviar (véase tabla 3.2):

Tabla 3.2 Comparación de los medios portadores
Donde SI corresponde a FACTIBLE y NO se refiere a NO FACTIBLE

	Objeto portador				
Información a enviar	Imagen	Audio	Video	HTML	C++
Documentos	SI	SI	SI	SI	SI
Imagen	SI	SI	SI	SI	SI
Audio	NO	NO	NO	SI	SI
Video	NO	NO	NO	SI	SI

De los datos mostrados en la tabla 3.2 se puede decir que mediante este método de suma binaria de ocultamiento de información se debe tener un objeto portador de mayor tamaño que el objeto que se desea enviar, de lo contrario no se podrá ocultar la información y marcará un error. De manera que en estos casos no se cumple con el objetivo principal de la esteganografía el cual es el no modificar las propiedades del objeto portador.

Se realizó, la misma prueba pero usando como objeto portador un código fuente de lenguaje C++ y al insertar el código hexadecimal dos tipos de información diferente (imagen y audio) este código no sufrió ningún tipo de modificación al momento de su compilación y ejecución, por lo que se puede usar cualquier tipo de código fuente como objeto portador.

A partir de lo anterior se usó como objeto portador el código html de una página web, los objetos portadores no sufrieron ninguna modificación y al momento de viajar por internet, la información se mantuvo íntegra, no hubo pérdidas de la misma y no importó el tamaño de la información a enviar, ni tampoco se modificó la apariencia de dicha página web, por lo que nadie detectaría que hay información oculta en el código html.

En este caso se selecciona el código html como objeto portador ya que es el código fuente más común y más usado para la generación de páginas web y por todo tipo de personas aunque no tengan algún tipo de conocimiento informático usa algún servicio web.

En el caso de ocultar información en un código fuente del cualquier programa no existirá problema de errores ya que la información que se está agregando se hace en forma de comentario y como se sabe un

programa puede tener tantos comentarios como sea necesario.

De esta manera se comprobó la buena adaptación de la esteganografía al protocolo HTTP, lo cual proporciona una opción más para el ocultamiento de la información.

El concepto de esteganografía de red puede abarcar varios aspectos de la Ingeniería en Computación, sin embargo se logró describir ampliamente la interacción que tiene con la seguridad Informática, principalmente con el ocultamiento de la información ya que pasa desapercibida por que usa objetos portadores los cuales son de uso habitual para las personas y si se quiere hacer más robusto se puede combinar con la criptografía, de manera que primero se aplique un algoritmo criptográfico para resguardar la información y después de transporte utilizando un proceso esteganográfico, de manera que si se llegase a detectar el mensaje esteganografiado no le sea posible al interceptor conocer la información.

Con estas pruebas se pudo validar una herramienta más para la seguridad informática y un complemento para la criptografía ya que como se mencionó durante este trabajo pueden trabajar juntas y hacer una herramienta más robusta.

El tema de la esteganografía aún es algo desconocido para muchas personas y en ocasiones puede ser confundido con la criptografía, pero mediante pruebas como estas se puede ir haciendo difusión sobre el tema al grado de que sea más conocido por las personas y poco a poco ser utilizado a la par de la criptografía.

La esteganografía es un tema sencillo el cual con la combinación de más herramientas de seguridad se puede convertir en una herramienta bastante robusta y así combatir con más armas los ataques informáticos que cada día son más sofisticados.

ANEXOS

GLOSRIO DE TERMINOS

1. **ACK.-** Del inglés acknowledgement, en español acuse de recibo o asentimiento, en comunicaciones entre computadores, es un mensaje que el destino de la comunicación envía al origen de ésta para confirmar la recepción de un mensaje. Si el mensaje está protegido por un código detector de errores y el dispositivo de destino posee además capacidad para procesar dicha información, el ACK también puede informar si se ha recibido de forma íntegra y sin cambios.
2. **ASCII.-** Código Estándar Estadounidense para el Intercambio de Información), pronunciado generalmente [áski] o [ásci], es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales. El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión.
3. **BIT.**- Bit es el acrónimo Binary digit (dígito binario). Un bit es un dígito del sistema de numeración binario.
4. **BMP.-** Formato propio del programa, que viene con el sistema operativo Windows. Puede guardar imágenes de 24 bits (16,7 millones de colores), 8 bits (256 colores) y menos.
5. **BODY.- E**tiqueta del código HTML en el se incluirán todas las instrucciones HTML y el texto que forman el documento.
6. **CGI**.- Interfaz de entrada común es una importante tecnología de la World Wide Web (www) que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.
7. **CONNECT.-** Si habilita el método CONNECT, el servidor podrá establecer una sesión de túnel SSL entre un cliente (por ejemplo, Netscape Navigator) y un servidor remoto a través de un servidor proxy. Las sesiones entre el cliente y el proxy y entre el proxy y el servidor remoto serán seguras. El proxy no puede acceder a los datos enviados al cliente. El servidor proxy puede ser un servidor base o un servidor seguro. Get -- Si habilita el método de tipo GET (generic envelope type), el servidor devolverá los datos identificados por el URL. Si éste se refiere a un programa ejecutable, el servidor devolverá la salida del programa.
8. **COPYRIGHT.-** El derecho de autor es un conjunto de normas jurídicas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores (los derechos de autor), por el solo hecho de la creación de una obra literaria, artística, musical, científica o didáctica, esté publicada o inédita.
9. **C++.-** Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.
10. **DCT.**- Expresa una secuencia finita de puntos de datos en términos de una suma de coseno funciones que oscilan a diferentes frecuencias.
11. **DELETE.-** Si habilita el método delete, el servidor suprimirá el objeto identificado por el URL. Una vez se haya suprimido el objeto, el URL no será válido. Dado que delete normalmente permite que los clientes supriman información del servidor, debe utilizar configuraciones de protección para definir quién puede utilizar este método y qué archivos pueden suprimirse.

12. **DNS.-** Sistema de nombres de dominio es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada.
13. **ECHO REPLY**.- Es un mensaje generado como respuesta a un mensaje Echo Request (petición de Eco).
14. **ECHO REQUEST.-** Es un mensaje de control que se envía a un host con la expectativa de recibir de él un Echo Reply (Respuesta eco). Esto es conocido como Ping y es una utilidad del protocolo ICMP, subprotocolo de IP. Todo host debe responder a un Echo Request con un Echo Reply que contenga exactamente los mismos datos que el primero.
15. **FTP.-** Protocolo de Transferencia de Archivos en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.
16. **GIF.-** Es un formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.
17. **HEAD.-** Si habilita el método HEAD, el servidor devolverá la cabecera de documento HTTP sin el cuerpo del documento.
18. **HEADER.-** Etiqueta del código HTML contiene, normalmente, una declaración directa de clases, subrutinas, variables, u otros identificadores.
19. **HTML.**- Lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.
20. **HTTP.-** Es el protocolo usado en cada transacción de la World Wide Web.
21. **ICMP.-** Es el sub protocolo de control y notificación de errores del Protocolo de Internet . Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.
22. **IP.**- Dirección IP el número que identifica a cada dispositivo dentro de una red con protocolo IP.
23. **JAVASCRIPT.-** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico.
24. **JPEG.-** Del inglés Joint Photographic Experts Group, Grupo Conjunto de Expertos en Fotografía es el nombre de un comité de expertos que creó un estándar de compresión y codificación de archivos de imágenes fijas.
25. **LSB.-** En computación , el bit menos significativo, es el bit de la posición en un binario de número entero que indica el valor de unidades, es decir, determinar si el número es par o impar.
26. **MESSENGER.-** Programa de mensajería instantánea creado por Microsoft Windows en 1999 y descontinuado en el 2007debido al reemplazo por Windows Live Messenger y ahora Skype.
27. **MP3.**- Formato de compresión de audio digital patentado que usa un algoritmo con pérdida para conseguir un menor tamaño de archivo. Es un formato de audio común usado para música tanto en ordenadores como en reproductores de audio portátil.
28. **MULTIMEDIA.-** El término multimedia se utiliza para referirse a cualquier objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales) para presentar o comunicar información.
29. **NCOVERT.-** Permite ocultar transferencias de archivos de la red a través de Internet. Mediante el uso de la falsificación de paquetes, NCovert esconde la transferencia de archivos por encubrimiento en los datos aparentemente inofensivos. Las funciones avanzadas le permiten ocultar su verdadera dirección IP, y con una planificación cuidadosa puede ocultar verdadera dirección IP de destino también.
30. **NETWORKING**.- Es una colección de ordenadores y otro hardware interconectados por canales de comunicación que permiten el intercambio de recursos y la información.

31. **OPTIONS.-** Si habilita el método OPTIONS, la petición devolverá información referente a las opciones de comunicaciones en la cadena de respuesta identificada por el URL. Este método permite al cliente determinar cuáles son las opciones y los requisitos asociados con un objeto, o bien cuáles son las capacidades de un servidor. No es necesaria ninguna acción sobre el objeto ni su recuperación.
32. **OSI.-** El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), (en inglés open system interconnection) es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización(ISO) en el año 1984. Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.
33. **PASSWORD.-** Es una palabra secreta o cadena de caracteres que se utiliza para el usuario de autenticación para probar la identidad, o para aprobación de acceso para obtener acceso a un recurso.
34. **PING.-** Es una utilidad de diagnóstico en redes de computadoras que comprueba el estado de la comunicación con el host local con uno o varios equipos remotos de una red TCP/IP por medio del envío de paquetes ICMP de solicitud y de respuesta.
35. **PIXEL.-** Un píxel o pixel, plural píxeles (acrónimo del inglés picture element, "elemento de imagen") es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.
36. **POP.-** Protocolo de Oficina de Correo o "Protocolo de Oficina Postal" en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto. Es un protocolo de nivel de aplicación en el Modelo OSI.
37. **POST.-** Seleccione POST para indicar que la entrada del programa CGI se pasará a éste en la corriente de entrada estándar.
38. **PUT.-** La petición contiene datos y un URL. El servidor almacena el recurso identificado en el URL. Si el recurso ya existe, PUT lo sustituye. Si el recurso no existe, PUT lo crea. Dado que PUT normalmente permite que los clientes añadan o sustituyan información en el servidor, debe utilizar configuraciones de protección para definir quién puede utilizar este método y para qué archivos.
39. **RFC.-** Es un documento cuyo contenido es una propuesta oficial para un nuevo protocolo de la red Internet (originalmente de ARPANET), que se explica con todo detalle para que en caso de ser aceptado pueda ser implementado sin ambigüedades.
40. **RGB.-** En Inglés Red, Green, Blue, en Español rojo verde azul es la composición del color en términos de la intensidad de los colores primarios de la luz puede referirse a: el modelo de color RGB, el tratamiento de la señal de vídeo RGB y el uso de RGB en HTML y otros lenguajes de programación.
41. **RST.-** Es un código usado para describir la calidad de las transmisiones de radio, especialmente en reportes de recepción escritos por oyentes de onda corta. Cada letra del código representa un factor específico de la señal, y cada factor tiene diferentes escalas.
42. **SCAPY.-** Aplicación para la manipulación de paquetes interactiva. Es capaz de forjar o decodificar los paquetes de un gran número de protocolos, enviarlos en el cable, la captura de ellos, satisfacer las peticiones y respuestas, y mucho más.
43. **SMTP.-** Protocolo para la transferencia simple de correo electrónico, es un protocolo de la capa de aplicación. Protocolo de red basado en texto, utilizados para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.
44. **SMTP.-** Protocolo para la transferencia simple de correo electrónico, es un protocolo de la capa de aplicación.
45. **SOAP.-** Es un protocolo de especificaciones para el intercambio de información estructurada en la implementación de Servicios Web en redes informáticas.

46. **SOCKS.-** Es un protocolo de Internet que permite a las aplicaciones Cliente-servidor usar de manera transparente los servicios de un firewall de red.
47. **SYN.-** Es un bit de control dentro del segmento TCP, que se utiliza para sincronizar los números de secuencia iniciales ISN de una conexión en el procedimiento de establecimiento de tres fases.
48. **TCPDUMP.-** Es un herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red.
49. **TCP/IP.-** Es un modelo de descripción de protocolos de red desarrollado en la década de los 70 por Vinton Cerf y Robert E. Kahn.
50. **TELNET.-** Telecommunication Network es el nombre de un protocolo de red a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella.
51. **TUNEL**.- Se conoce como túnel al efecto de la utilización de ciertos protocolos de red que encapsulan a otro protocolo. Así, el protocolo A es encapsulado dentro del protocolo B, de forma que el primero considera al segundo como si estuviera en el nivel de enlace de datos. La técnica de tunelizar se suele utilizar para trasportar un protocolo determinado a través de una red que, en condiciones normales, no lo aceptaría. Otro uso de la tunelización de protocolos es la creación de diversos tipos de redes privadas virtuales.
52. **TRACE.-** Si habilita el método TRACE, el servidor se hará eco del mensaje de petición enviado por el cliente. Este método permite al cliente ver qué es lo que se recibe al otro extremo de la cadena de petición. Entonces, el cliente puede utilizar los datos para comprobación o para información de diagnóstico. El tipo de contenido de la respuesta.
53. **UNIX.-** Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969, por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.
54. **URI.-** En español identificador uniforme de recursos es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema. Los URI pueden ser localizadores uniformes de recursos (URL), Uniform Resource Name (URN), o ambos.
55. **URL.-** Un localizador de recursos uniforme, más comúnmente denominado URL, es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, etcétera.
56. **WAV**.- Formato de audio digital normalmente sin compresión de datos desarrollado y propiedad de Microsoft y de IBM que se utiliza para almacenar sonidos en el PC, admite archivos mono y estéreo a diversas resoluciones y velocidades de muestreo, su extensión es wav.
57. **WEB.-** El término Web comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario1 y la colaboración en la World Wide Web.
58. **WHATSAPP.-** Es un software privativo multiplataforma de mensajería Instantánea para teléfonos inteligentes.
59. **WWW.-** En informática, la World Wide Web o Red informática mundial1 es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.
60. **XML-RPC.-** Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.
61. **XOR**.- En criptografía, el cifrado XOR es, como su nombre indica, un algoritmo de cifrado basado en el operador binario XOR.

FUENTES DE INFORMACIÓN

1. Pablo Andres Deymonnaz, (2012), "*Análisis de vulnerabilidades esteganográficas en protocolos de comunicación IP y HTTP"*, Tesis de Ingeniería en Informática, Universidad de Buenos Aires, Argentina.
2. Francisco José Súarez Alonso, (2010/2011), *"Área de Arquitectura y Tecnología de Computadoras"*, Universidad de Oviedo.
3. Ribagorda Garnacho, Juan M. Estévez-Tapiador, Julio César Hernández Castro, (2007),*" Descubriendo el reverso de internet: Web mining, mensajes aparentes y secretos ocultos*".
4. Álvaro Navarro Clemente, (2005), *"Esteganografía"*, Universidad Rey Juan Carlos, Móstoles, España.
5. *"Esteganografía desde cero"*, (2009), XiONex, http://www.xionexsystems.blogspot.com.
6. Observatorio de la Seguridad de la Información, *"Esteganografía, el arte de ocultar información"*, Instituto Nacional de las Tecnologías de la Comunicación.
7. Jesús Díaz Vico, (2010), "*Esteganografía y Estegoanálisis: Ocultacion de datos en sistemas de audio"* Tesis de Master en Tecnologías de la Información, Universidad Politécnica de Madrid, España.
8. Lukasz Grzegorz Maciak, Micheal Alexis Ponniah, Renu Sharma, "*MP3 Stegonography, Applying Stenography to Music Captioning*".
9. University of Tennessee, (2002) *"Network Working Group".*
10. David Rodríguez Hernández, (2007), "*Redes y comunicaciones"*, Zamora.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

PROCEDIMIENTO DE INTEGRACIÓN DE LA ESTEGANOGRAFÍA AL PROTOCOLO HTTP

TESIS PROFESIONAL

PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

JUAN MARTÍN CORONA FALCÓN

DIRECTORA DE TESIS:

M.C. María Jaquelina López Barrientos

CD. UNIVERSITARIA, MÉXICO, 2015