ESTEGANOGRAFÍA DE ARCHIVOS DE AUDIO WAV: INSERCIÓN DE EJECUTABLES Y ANÁLISIS DE EVASIÓN ANTIVIRUS

WAV AUDIO FILE STEGANOGRAPHY: EXECUTABLE EMBEDDING AND ANTIVIRUS EVASION ANALYSIS

Jhamil Arturo Zeballos Soruco
Universidad San Francisco Xavier
zeballos.jhamil@usfx.bo

Recibido: 28 Abril 2025 / Revisado: 4 Agosto 2025 / Aceptado: 22 Agosto 2025 / Publicado: 23 Septiembre 2025

Resumen

La esteganografía digital consiste en ocultar información dentro de archivos multimedia con el propósito de que su presencia pase desapercibida. El presente estudio analiza la viabilidad de ocultar archivos ejecutables (.exe, .bat, .dll) en archivos de audio en formato WAV mediante la herramienta Steghide. Se empleó un diseño experimental para evaluar la capacidad de incrustación, la reproducción del archivo modificado y su detección por antivirus convencionales y la plataforma VirusTotal. Los resultados evidencian que es posible incrustar archivos ejecutables sin alterar la calidad perceptible del audio y sin que los sistemas de detección los identifiquen como maliciosos. Este hallazgo subraya un riesgo real en la propagación de malware a través de medios aparentemente inocuos, resaltando la necesidad de reforzar los sistemas de seguridad informática con técnicas de detección más avanzadas.

Palabras clave: Esteganografía, Antivirus, steghide, archivos de audio, archivos ocultos

Introducción

La esteganografía es una técnica utilizada desde la antigüedad entendiendo de que ella data de hace siglos atrás y fue utilizada como una *técnica de ocultar información sensible o privada dentro de algo que parezca que todo es normal* (Zone H, 2024). En el contexto digital, este concepto se ha expandido hacia el uso de imágenes, videos y archivos de audio como portadores o anfitriones de información secreta para ocultar información dentro de un medio que aparenta ser inocuo. A diferencia de la criptografía, cuyo propósito es proteger el contenido frente a la intercepción, la esteganografía busca evitar que la existencia del mensaje sea detectada (Sánchez, 2023).

En los últimos años, la esteganografía digital ha sido utilizada no solo con fines legítimos, sino también en ciberataques. Este fenómeno ha dado lugar al concepto de stegomalware: software malicioso que utiliza técnicas esteganográficas para insertarse en archivos multimedia y evadir mecanismos de seguridad (Badar, 2025). Casos documentados, como Waterbug, han mostrado cómo audios WAV han sido empleados para propagar malware sin ser detectados por antivirus tradicionales (Dittmann, 2024).

Este estudio tiene como objetivo evaluar la viabilidad de ocultar ejecutables en archivos de audio WAV utilizando la herramienta Steghide, analizar los resultados obtenidos y discutir sus implicaciones en el ámbito de la seguridad informática. La relevancia académica de este trabajo radica en exponer una vulnerabilidad frecuentemente ignorada, así como en destacar la necesidad de desarrollar herramientas de detección más sofisticadas.

Estado del arte

La literatura especializada describe múltiples técnicas de esteganografía de audio. El método más común es el de sustitución del bit menos significativo (LSB), que altera de forma imperceptible el último bit de cada muestra para incrustar información (Gupta, 2012). Otros métodos incluyen la codificación de fase, el eco-hiding y transformaciones en el dominio de frecuencia como DWT-SVD, que ofrecen mayor robustez frente a compresión y ruido (Yang, 2024).

Investigaciones recientes han explorado la combinación de esteganografía con criptografía caótica, lo que incrementa la capacidad de ocultación y la resistencia frente a técnicas de detección (Nasr, 2024). Paralelamente, la comunidad forense ha reportado el uso de archivos WAV como vehículos de malware en campañas APT, confirmando el carácter práctico de esta amenaza (Strachanski, 2024). Sin embargo, la detección de estos métodos sigue siendo limitada. Modelos de aprendizaje profundo y redes neuronales multiescalares han mostrado avances en estegoanálisis, aunque aún no alcanzan una tasa de detección suficiente en escenarios reales (Peng, 2025).

Este estado del arte muestra un panorama en el que coexisten técnicas simples y accesibles como Steghide con desarrollos avanzados en ocultación y detección. El presente estudio se centra en la primera categoría para evidenciar que incluso con métodos básicos es posible comprometer la seguridad informática.

Métodos

El diseño de la investigación es experimental-descriptivo, con el propósito de demostrar la viabilidad de ocultar ejecutables en archivos WAV. Se utilizaron los siguientes instrumentos: Steghide v0.5.1 como software esteganográfico, Windows 10/11 como sistema operativo anfitrión y VirusTotal como plataforma de validación antivirus.

El procedimiento incluyó: (1) selección de archivos WAV originales, (2) inserción de ejecutables (.bat, .exe, .dll) en los audios utilizando la herramienta de incrustación steghide, (3) verificación de la reproducción y tamaño resultante, (4) análisis del archivo esteganográfico en antivirus locales y en VirusTotal, y (5) registro de métricas como variación de tamaño, tiempo de procesamiento y detección/no detección de anomalías.

Experimentación y resultados

Los experimentos mostraron que la inserción de archivos ejecutables en audios WAV fue exitosa en múltiples casos, sin alterar la calidad percibida del audio ni generar alertas en antivirus. A continuación, se presentan los resultados acompañados de capturas representativas de cada caso. A continuación, se exponen por casos, las incrustaciones de archivos diferenciados por casos.

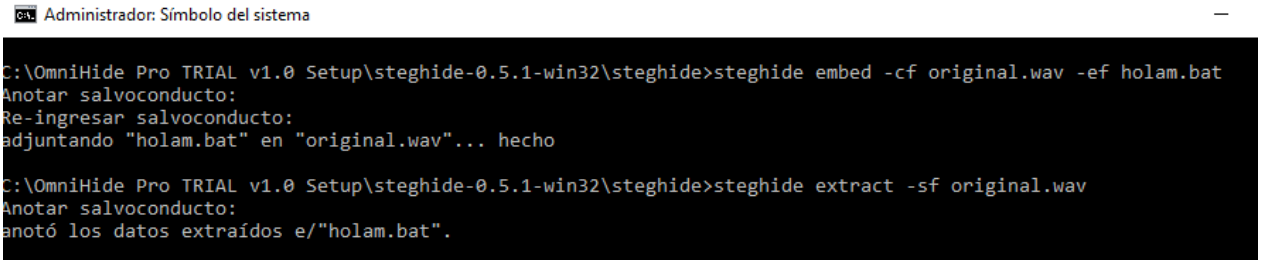

Figura 1: *Captura de Incrustación de archivo .bat y su posterior extracción*

Caso 1: Incrustación archivo bat.

La Figura 1 muestra la ejecución de comandos de Steghide en Windows desde la consola de comandos (cmd), y corresponde a una prueba de incrustación y extracción de un archivo .bat dentro de un archivo de audio .wav. El proceso seguido es el siguiente:

1. Incrustación (primera parte de la ejecución en el cmd): ***steghide embed -cf original.wav -ef holam.bat***

Donde:

- ***-cf original.wav:*** especifica el archivo anfitrión (carrier file), en este caso un audio WAV.
- **-ef holam.bat**: archivo a ocultar dentro del WAV.
- La herramienta solicita la clave (salvoconducto) que sirve como clave de acceso para recuperar posteriormente el archivo (un posible atacante conocerá dicha clave)
- **Resultado:** el archivo holam.bat fue incrustado exitosamente en **original.wav.**

2. Extracción (segunda parte de la ejecución en el cmd): **steghide extract -sf original.wav**

- **-sf original.wav**: indica el archivo donde se ha aplicado esteganografía desde el cual se quiere extraer el contenido oculto.

- El sistema solicita la clave usada en la incrustación.
- **Resultado: Steghide extrajo correctamente el archivo oculto y lo guardó como holam.bat.**

Se evidencia que el procedimiento ejecutado para ocultar y recuperar un archivo ejecutable tipo **(.bat)** en un archivo WAV funciona de manera exitosa con Steghide y se advierten los siguientes resultados iniciales:

- El archivo de audio mantiene una apariencia "normal".
- El archivo oculto se puede recuperar íntegramente con la clave.
- Esto demuestra la factibilidad práctica de la técnica y su potencial uso con fines maliciosos.

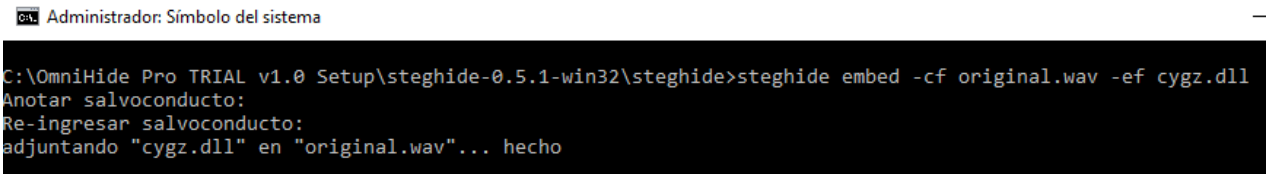

Figura 2: *Captura de Incrustación archivo dll*

Caso 2: Incrustación archivo dll.

La Figura 2 muestra el mismo procedimiento que en la Figura 1, en este caso aplicado a una prueba de incrustación de un archivo .dll dentro de un archivo de audio .wav. con el siguiente proceso:

Incrustación: **steghide embed -cf original.wav -ef cygz.dll**

Donde:

- **-cf original.wav:** archivo anfitrión (un audio en formato WAV).
- **-ef cygz.dll:** archivo a ocultar, en este caso una librería dinámica de Windows (cygz.dll).
- La herramienta solicita la Clave (salvoconducto):
- **Resultado:** El mensaje final indica: adjuntando "cygz.dll" en "original.wav"... hecho. Esto significa que la incrustación del archivo DLL en el WAV se realizó de forma exitosa.

La figura 2 demuestra que, no solo archivos .bat pueden ser ocultados, sino también bibliotecas dinámicas (.dll). Esto amplía la superficie de ataque, ya que una DLL maliciosa podría ser incrustada y distribuida bajo la apariencia de un archivo de audio legítimo.

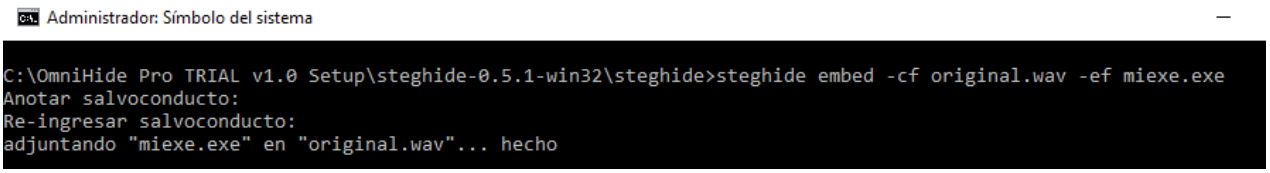

Figura 3: *Captura de Incrustación archivo exe*

Caso 3: Incrustación archivo exe.

Con el mismo procedimiento aplicado en los anteriores casos, la figura 3, muestra la incrustación de un archivo ejecutable .EXE. El proceso es el siguiente:

Incrustación: steghide embed -cf original.wav -ef miexe.exe

Donde:

- **-cf original.wav:** define el archivo portador (archivo de audio en formato WAV).
- **-ef miexe.exe:** indica el archivo a ocultar, en este caso un ejecutable de Windows (miexe.exe, generado desde un compilador).
- La herramienta solicita la Clave (salvoconducto):
- **Resultado:** El mensaje final indica: adjuntando "miex.exe" en "original.wav"... hecho Confirma que el archivo ejecutable fue insertado correctamente en el archivo WAV.

La figura 3 demuestra que Steghide también puede ocultar archivos ejecutables completos (.exe) dentro de un archivo WAV sin alterar su apariencia.

Esto implica un riesgo mayor, ya que los ejecutables pueden ejecutar código malicioso directamente en el sistema al ser recuperados, convirtiendo a un simple archivo de audio en un posible vector de ataque.

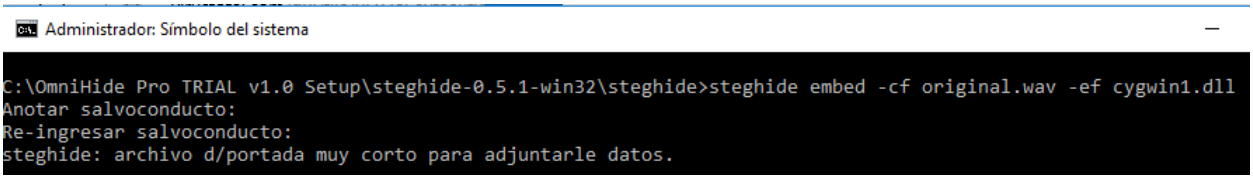

Figura 4: *Captura de Incrustación no exitosa por tamaño*

Caso 4: Incrustación archivo dll fallida.

Para el caso de la Figura 4, la incrustación no es exitosa, debido a que el archivo anfitrión es de 4.18Mb de tamaño, mientras que el dll incrustado de 5.01Mb. En otras pruebas realizadas se ha podido verificar que, para el tamaño del anfitrión, se destina entre 10 y 12% del tamaño del archivo original para poder embeberlo, tamaños mayores no pueden ser incrustados dando resultados fallidos en el experimento. El proceso seguido es el siguiente:

Incrustación: **steghide embed -cf original.wav -ef cygwin1.dll**

Donde:

- **-cf original.wav:** archivo portador (un archivo de audio WAV).
- **-ef cygwin1.dll:** archivo a ocultar, en este caso una librería dinámica de gran tamaño (cygwin1.dll).
- La herramienta solicita la Clave (salvoconducto), pero al continuar el proceso aparece un error: **steghide: archivo d/portada muy corto para adjuntarle datos.**
- **Resultado:** El tamaño del archivo portador (original.wav) no es suficiente para contener el archivo DLL, por lo que la incrustación falló.

La figura 4 demuestra una limitación técnica de Steghide y del método de esteganografía empleado:

- El archivo portador debe tener una capacidad mínima proporcional al tamaño del archivo a ocultar.
- En los experimentos previos se determinó que el límite ronda entre un 10% y 12% del tamaño del archivo anfitrión.
- En este caso, al ser el archivo DLL demasiado grande en relación con el WAV, la incrustación no fue posible.

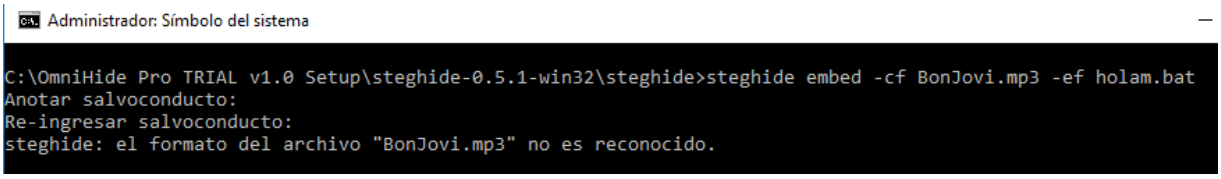

Figura 5: *Captura de Prueba no exitosa con mp3*

Caso 5: Incrustación fallida en archivo mp3.

La figura 5, muestra otro caso de incrustación no exitosa, en este caso se usa un archivo MP3 como anfitrión.El proceso utilizado es el siguiente:

Incrustación: steghide embed -cf BonJovi.mp3 -ef holam.bat

Donde:

- -cf BonJovi.mp3: archivo portador, en este caso un audio comprimido en formato MP3.
- -ef holam.bat: archivo a ocultar, un ejecutable por lotes (holam.bat).
- La herramienta solicita ingresar y reingresar la Clave (salvoconducto), pero la ejecución no continúa con éxito.
- **Resultado:** steghide: el formato del archivo "BonJovi.mp3" no es reconocido.

Esto significa que Steghide no admite el formato MP3 como archivo anfitrión, ya que estos

archivos están comprimidos con pérdida y no conservan el espacio necesario para ocultar datos.

se ha podido validar que dado que estos archivos ya se encuentran comprimidos no guardan espacio para incrustaciones.

Validaciones de tamaño

Por otra parte, se ha realizado la captura de las propiedades del archivo **original.wav** antes de ser sometido a la incrustación del archivo a ocultar dando como resultado la Figura 6:

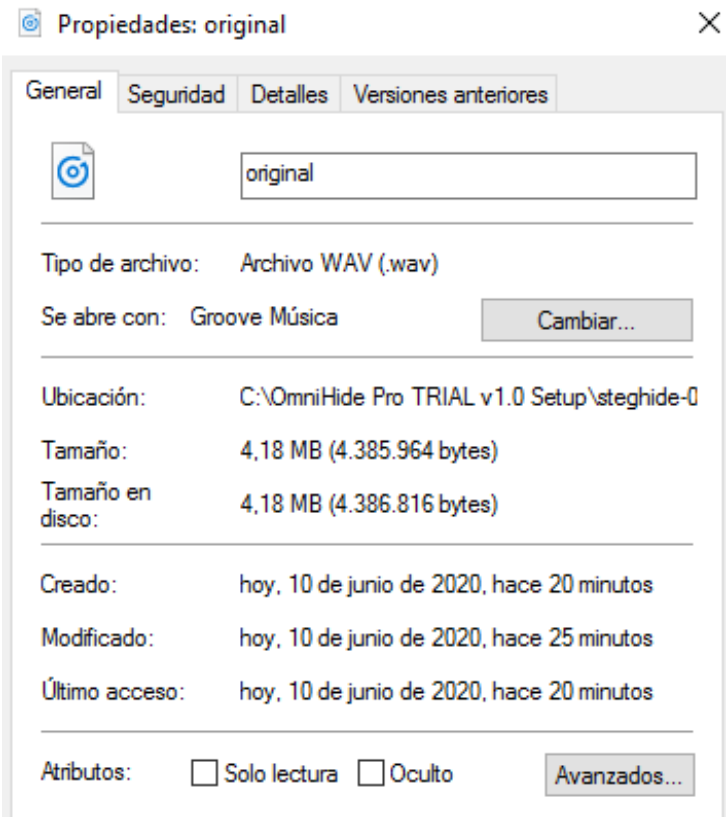

Figura 6: *Propiedades archivo original.wav*

La Figura 6 muestra las propiedades del archivo original.wav antes de realizar la incrustación de archivos ejecutables. Se puede observar su tamaño y metadatos básicos, los cuales sirven como referencia para comparar con el archivo resultante después de la esteganografía.

El análisis experimental indicó que, aun tras insertar archivos .bat, .exe y .dll, el tamaño del archivo WAV no presenta variaciones visibles. Esto refuerza la peligrosidad de la técnica, ya que a simple vista (e incluso mediante verificaciones superficiales) no se detectan cambios, dificultando la identificación del archivo como sospechoso.

Tabla 1: *Métricas básicas de inserción y aplicación antivirus*

Archivo WAV	Tamaño original (MB)	Archivo oculto	Tamaño modificado (MB)	Resultado antivirus
original.wav	4.18	bat (0.2 MB)	4.18	No detectado
original.wav	4.18	exe (0.3 MB)	4.18	No detectado
original.wav	4.18	dll (0.05 MB)	4.18	No detectado
original.wav	4.18	dll (5.01 MB)	—	Incrustación fallida
bonjovi.mp3	4.92	bat (0.2 MB)	—	Incrustación fallida

Análisis en Virustotal

La Figura 8, corresponde al resultado de someter el archivo WAV esteganografiado a la plataforma VirusTotal en el sitio virustotal.com (virustotal, 2025) que integra múltiples motores antivirus (59 en este experimento). El análisis mostró que ninguno de los motores detectó anomalías o malware en el archivo, a pesar de que contenía ejecutables ocultos.

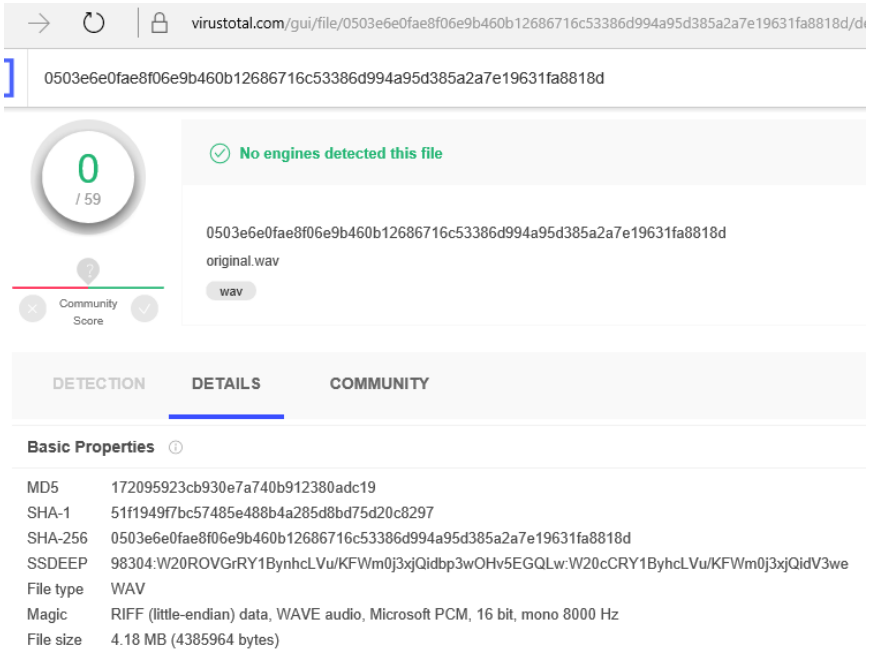

Figura 7: *Análisis archivo modificado*

Este resultado confirma lo planteado por Kaspersky (Kuksov, 2019): la esteganografía es una de las técnicas más difíciles de detectar mediante firmas antivirus tradicionales. El hallazgo enfatiza la necesidad de adoptar técnicas de detección basadas en comportamiento y de fomentar mayor investigación en estegoanálisis aplicado a audio. Llama la atención, que un ejercicio tan simple como la incrustación de archivos de ejecución en entornos Windows como exe, bat o dll puedan ser utilizados tan fácilmente como instrumentos de ataque o de ejecución de instrucciones en los equipos de los usuarios y que estos pasen inadvertidos por los antivirus.

Discusión

Los resultados evidencian que incluso con una herramienta básica como Steghide es posible ocultar ejecutables en archivos WAV y evadir la detección antivirus. Este hallazgo coincide con investigaciones que señalan la dificultad de detectar stegomalware mediante firmas convencionales (Badar, 2025; Strachanski, 2024). Sin embargo, se observaron limitaciones: la capacidad de inserción depende del tamaño del archivo anfitrión y el método no funciona en archivos comprimidos como MP3.

Comparado con técnicas modernas como DWT-SVD o codificación caótica, el enfoque de Steghide es menos robusto, pero su simplicidad representa un riesgo, pues facilita que atacantes con pocos recursos lo utilicen. En este sentido, las implicaciones prácticas son relevantes: archivos de audio, ampliamente compartidos, pueden convertirse en vectores de propagación de malware.

Conclusiones

- Es técnicamente viable ocultar ejecutables en archivos WAV mediante Steghide, manteniendo la reproducción normal y sin ser detectados por los antivirus tradicionales.
- La capacidad de inserción está limitada al tamaño del archivo anfitrión, estimada entre un 10% y 12% de su tamaño total.
- Archivos comprimidos como MP3 no permiten incrustaciones bajo este método.
- Los hallazgos demuestran la necesidad de desarrollar sistemas de detección más sofisticados, basados en análisis de comportamiento o técnicas de inteligencia artificial.
- Se recomienda ampliar las investigaciones hacia formatos comprimidos, plataformas diferentes a Windows y métodos avanzados de estegoanálisis.
- Se recomienda ampliar las investigaciones hacia formas de aprovechamiento de este tipo de vulnerabilidades por parte de los atacantes.

Referencias

Badar, L. T. (2025). A comprehensive survey on stegomalware detection in digital media. Signal Processing.

Dittmann, J. (2024). Forensic trace analysis for MP3-based stego-malware. ACM Transactions on Multimedia Forensics.

Gupta, S. G. (2012). nformation hiding using least significant bit steganography and cryptography. I.J. Modern Education and Computer Science, 27-34, Vol1, Num. 6.

Hetzl, S. (24 de Diciembre de 2003). http://steghide.sourceforge.net/. Recuperado el 10 de Junio de 2025, de http://steghide.sourceforge.net/

Kuksov, I. (4 de Julio de 2019). kaspersky.es. Recuperado el 10 de Junio de 2025, de https://www.kaspersky.es/blog/digital-steganography/18791/

Nasr, M. A.-S.-R.-S.-F.-S. (2024). Robust audio steganography using chaotic maps. Scientific Reports.

Peng, J. (2025). Audio steganalysis using multi-scale feature fusion-based CNN. IET Signal Processing.

Sánchez, A. (26 de 6 de 2023). Proteger mi PC. (Proteger mi PC) Recuperado el 9 de 6 de 2025, de https://protegermipc.net/2018/06/26/introduccion-a-la-esteganografia/

Shailender Gupta, A. G. (2012). Information Hiding Using Least Significant Bit Steganography and Cryptography. I.J.Modern Education and Computer Science, 1(6), 27-34.

Strachanski, F. (2024). A comprehensive pattern-based overview of stegomalware. ACM Computing Surveys.

twoeggz. (29 de Noviembre de 2017). twoeggz. Recuperado el 10 de Junio de 2025, de https://www.twoeggz.com/int/5265059.html

virustotal. (4 de abril de 2025). Recuperado el 10 de Junio de 2025, de virustotal: https://www.virustotal.com/

Yang, G. (2024). An improved phase coding audio steganography algorithm. arXiv preprint.

Zone H. (9 de Junio de 2024). zone-h.org. Obtenido de http://zone-h.org: http://www.infosecwriters.com/text_resources/pdf/Steganography_AMangarae.pdf