Câu hỏi ôn tập

Cơ chế hoạt động của mã độc

1. Phân biệt virus, sâu máy tính (worm), botnet, ransomware, castern egg, salami attack, backdoor, rootkit, logic bomb, time-bomb, trojan.

   Virus: Virus là một loại mã độc có khả năng tự nhân bản và lây nhiễm vào các file thực thi hoặc phần mềm khác. Virus thường gắn kết vào các file và cần sự tương tác của người dùng để lây lan. Khi một file chứa virus được chạy, virus sẽ sao chép chính nó và lây nhiễm sang các file khác trong hệ thống.

   Sâu máy tính (Worm): Sâu máy tính là một loại mã độc tự nhân bản và tự lan truyền qua mạng mà không cần sự tương tác của người dùng. Worm tìm kiếm các máy tính và hệ thống mạng đã bị lỗ hổng bảo mật để xâm nhập và lây nhiễm. Một khi sâu máy tính đã lây nhiễm vào một hệ thống, nó có thể tiếp tục lan truyền và gây hại đến các máy tính khác.

   Botnet: Botnet là một mạng các máy tính bị kiểm soát từ xa bởi kẻ tấn công thông qua sử dụng mã độc. Các máy tính trong botnet thường bị lây nhiễm và trở thành "robot" (bot) để thực hiện các hoạt động độc hại, chẳng hạn như gửi thư rác, tấn công mạng hoặc phục vụ cho mục đích tấn công từ chối dịch vụ (DDoS).

   Ransomware: Ransomware là một loại mã độc được sử dụng để tấn công và mã hóa dữ liệu trên máy tính của người dùng. Khi dữ liệu bị mã hóa, kẻ tấn công yêu cầu một khoản tiền chuộc để giải mã dữ liệu. Nếu không trả tiền, dữ liệu có thể bị mất hoặc không thể khôi phục.

   Easter egg: Easter egg là một phần của phần mềm hoặc trò chơi bí mật và ẩn chứa những tính năng, trò đùa hoặc thông điệp được nhúng bên trong. Easter egg thường không gây hại và thường được tạo ra như một sự bất ngờ hay vui nhộn cho người dùng.

   Salami attack: Salami attack (còn gọi là "salami slicing") là một phương pháp tấn công mà kẻ tấn công lấy đi một lượng nhỏ, không đáng kể từ nhiều tài khoản hoặc giao dịch để tạo ra lợi ích lớn cho mình. Ví dụ, một chương trình máy tính có thể lấy một khoản tiền rất nhỏ từ mỗi giao dịch và gửi vào tài khoản của kẻ tấn công.

Backdoor: Backdoor là một cửa sau không được biết đến trong phần mềm hoặc hệ thống để tránh cơ chế kiểm soát và cho phép truy cập trái phép. Backdoor thường được tạo ra bởi nhà phát triển hoặc kẻ tấn công để thực hiện việc kiểm soát từ xa hoặc truy cập trái phép vào hệ thống.

Rootkit: Rootkit là một loại mã độc mà kẻ tấn công sử dụng để che giấu sự hiện diện của nó trên hệ thống. Rootkit thường can thiệp vào hệ điều hành và che giấu các hoạt động độc hại, hỗ trợ kẻ tấn công duy trì quyền kiểm soát và truy cập trái phép vào hệ thống.

Logic bomb và Time bomb: Logic bomb và Time bomb đều là những phần mềm độc hại được thiết lập để thực hiện một hành động cụ thể vào thời gian hoặc điều kiện nhất định. Logic bomb thường được kích hoạt bởi một sự kiện cụ thể xảy ra trong hệ thống, trong khi Time bomb được thiết lập để kích hoạt vào một thời điểm nhất định.

Trojan: Trojan (hoặc còn gọi là "Trojan horse") là một loại mã độc được giấu dưới dạng phần mềm hoặc tập tin khác và thường được người dùng tải xuống từ các nguồn không đáng tin cậy. Một khi Trojan được chạy, nó có thể tạo ra lỗ hổng bảo mật, thu thập thông tin cá nhân hoặc cho phép kẻ tấn công từ xa kiểm soát và truy cập trái phép vào hệ thống.

Malicious Software	Description	Replication & How	Propagation & How	Payload
Virus	Malware that infects and replicates by attaching itself to files or programs	Replicates by modifying host files or programs	Spreads through file sharing, email attachments, etc.	Can corrupt or delete files, steal data, or disrupt system functionality
Worm	Self-replicating malware that spreads across networks and systems without human action	Replicates by exploiting vulnerabilities or using social engineering techniques	Spreads through network connections, email, or removable media	Consumes network resources, slows down systems, and can carry other malicious payloads
Botnet	A network of compromised computers controlled by an attacker for malicious activities	No replication, infects individual computers	Spreads by infecting new computers and recruiting them to the botnet	Can be used for distributed denial of service (DDoS) attacks, spamming, or data theft
Ransomware	Encrypts user's files and demands a ransom payment in exchange for decryption	Does not replicate, often delivered via email or exploit kits	Spreads through infected email attachments, malicious downloads, or compromised websites	Prevents access to files until a ransom is paid, causing financial loss or data compromise
Easter Egg	Hidden feature or message intentionally placed by developers in software or applications	N/A	N/A	Generally harmless, serving as a hidden surprise or inside joke

Salami Attack	Fraudulent method of stealing tiny amounts from multiple financial transactions	No replication, involves manipulating financial transactions	N/A	Aggregates small amounts of money from numerous transactions for illicit gains
Backdoor	Secretly created entry point in a system or software, allowing unauthorized access	No replication, manually installed by an attacker	N/A	Enables remote control, data theft, or other unauthorized activities
Rootkit	Stealthy software that hides its presence and grants unauthorized control over a system	No replication, typically installed through an exploit	N/A	Can enable remote access, modify system logs, or install additional malware
Logic Bomb	Malware that lies dormant until a specific condition is met, triggering malicious actions	No replication, embedded in a system or software	N/A	Can delete files, corrupt data, or disrupt system operations when the trigger is activated
Time Bomb	Similar to a logic bomb, but activated at a specific time and date	No replication, embedded in a system or software	N/A	Can cause system crashes, data loss, or other malicious actions at a predetermined time
Trojan	Malware disguised as legitimate software, deceiving users into executing it	No replication, typically delivered via email or downloads	N/A	Can provide unauthorized access, steal data, or install other malware silently

Phân loại:

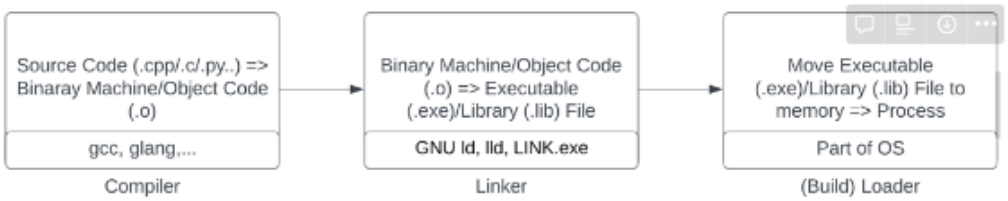
Loại mã độc	Hành vi	Đặc điểm	Cách lây lan	Ảnh hưởng
Mã độc Trojan	Xâm nhập và kiểm soát từ xa	Giả mạo thành phần hữu ích	Email, trang web độc hại, phần mềm giả mạo	Đánh cắp thông tin, tạo cửa sau để tấn công
Mã độc Ransomware	Mã hóa dữ liệu và yêu cầu tiền chuộc	Yêu cầu tiền chuộc	Email độc hại, tải xuống từ trang web độc hại	Mất dữ liệu, mất tiền, ảnh hưởng kinh doanh
Mã độc Spyware	Thu thập thông tin người dùng	Ghi lại hoạt động người dùng	Email độc hại, trang web độc hại, phần mềm giả mạo	Đánh cắp thông tin cá nhân, ảnh hưởng đến quyền riêng tư
Mã độc Adware	Hiển thị quảng cáo không mong muốn	Hiển thị quảng cáo không mong muốn	Phần mềm miễn phí, tải xuống từ trang web độc hại	Quảng cáo phiền hà, giảm hiệu suất hệ thống
Mã độc Botnet	Kiểm soát hệ thống từ xa	Tạo mạng lưới zombie kiểm soát từ xa	Email độc hại, trang web độc hại, phần mềm giả mạo	Tấn công từ chối dịch vụ (DDoS), lây lan mạng lưới độc hại
Mã độc Keylogger	Ghi lại các phím được nhấn trên bàn phím	Ghi lại hoạt động người dùng	Email độc hại, trang web độc hại, phần mềm giả mạo	Đánh cắp thông tin đăng nhập, mật khẩu
Mã độc Rootkit	Ẩn đi các hoạt động độc hại	Ẩn mình khỏi phân tích và phát hiện mã độc	Phần mềm giả mạo, tải xuống từ trang web độc hại	Kiểm soát hệ thống từ xa, khó phát hiện và loại bỏ
Mã độc Phishing	Lừa đảo	Mô phỏng trang web, email giả mạo	Email độc hại, trang web độc hại, tin nhắn xã hội	Đánh cắp thông tin cá nhân, tài khoản
Mã độc Virus	Tự sao chép và lây nhiễm hệ thống	Tự sao chép và gắn kết vào các tệp tin khác	Email độc hại, USB, tải xuống từ trang web độc hại	Phá hoại dữ liệu, ảnh hưởng đến hiệu suất hệ thống
Mã độc Worm	Tự lây lan qua mạng	Tự sao chép và lây nhiễm qua các hệ thống khác	Lỗ hổng mạng, email độc hại, USB, trang web độc hại	Lây nhiễm và tốn tài nguyên mạng, ảnh hưởng đến hiệu suất hệ thống
Mã độc Logic Bomb	Kích hoạt theo sự kiện hoặc điều kiện cụ thể	Được kích hoạt theo sự kiện hoặc điều kiện cụ thể	Tải xuống từ trang web độc hại, email độc hại, phần mềm giả mạo	Gây hại hoặc xóa dữ liệu khi điều kiện được đáp ứng
Mã độc Fileless	Không tạo ra tệp tin mới trên hệ thống	Sử dụng các kỹ thuật tiêm mã vào bộ nhớ	Email độc hại, trang web độc hại, phần mềm giả mạo	Khó phát hiện, khó loại bỏ, ảnh hưởng đến hiệu suất hệ thống

Virus:

▼ Features:

Là đoạn mã độc hại được chèn vào các vật chứa là các file thực thi (PE file)

▼ File Injection:



Common Formats:

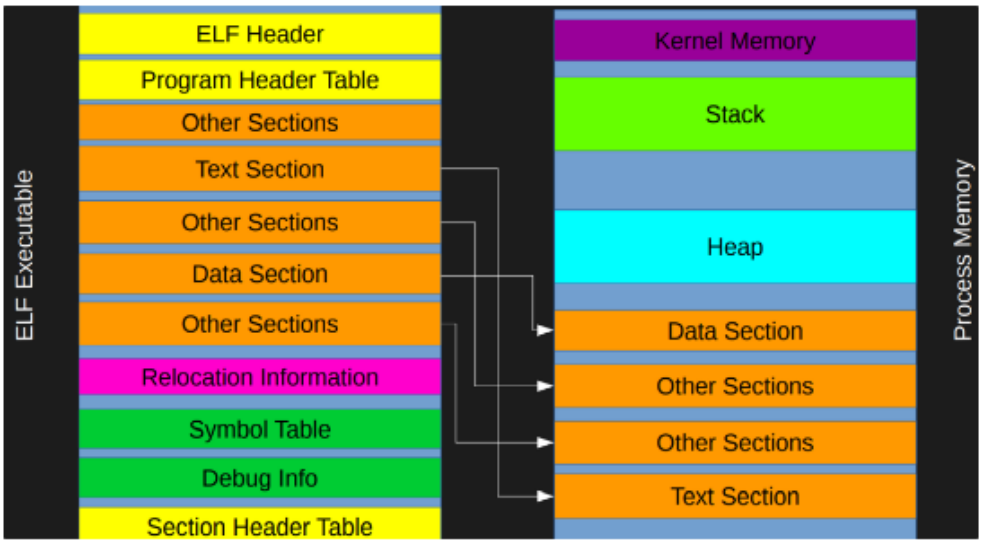
▼ **ELF (Executable and Linkable Format):**

Defines format for:

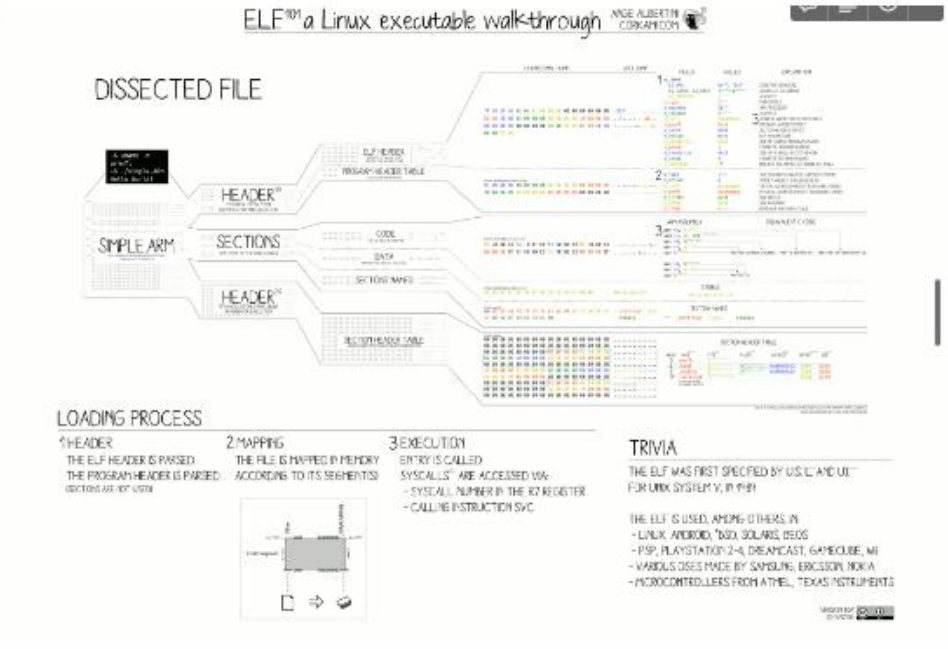
– Executables
– Object files
– Dynamic libraries (shared libraries)
– Core dumps

Architecture to Memory:

Some sections will be directly copied to memory:
– Example: .text, .data, .init, .dynamic
– The location (memory addresses) of these sections are defined in the ELF (if not PIC/PIE and ASLR)
Some sections will not be copied to memory
– Example: symbol table, debug info



Executable walk through:



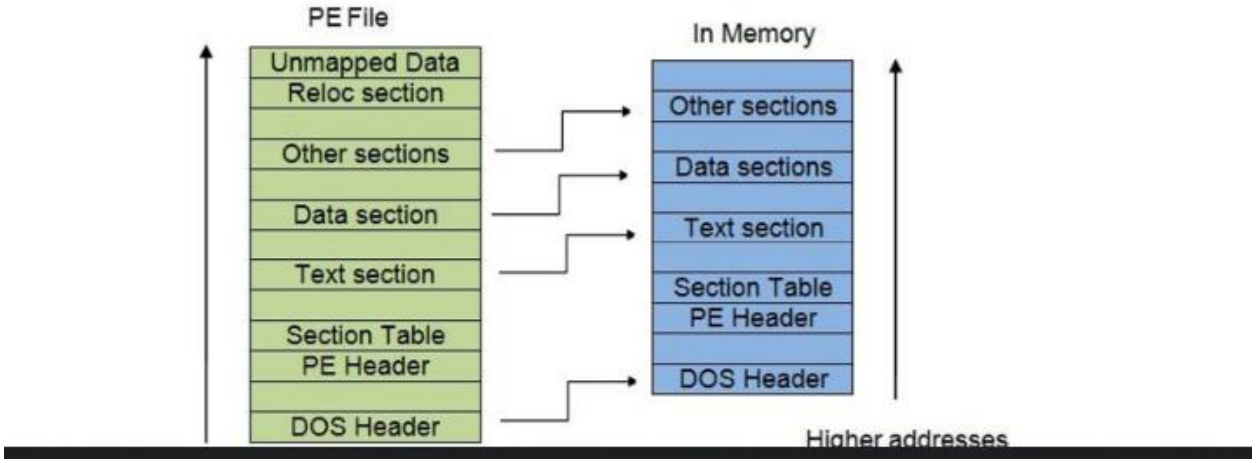
▼ PE (Portable Executable):

PE32 for 32-bit code; PE32+ for 64-bit code.

Architecture to Memory:

Common sections are .text (for code), .data (read/write data), .rdata (read-only data, .reloc (relocation data used to build IATs)

There are empty spaces in executable files
– The beginning of ELF files
– Empty spaces between functions
– Empty spaces between sections
– Nops in functions
– Some linkers make executable file align to page boundaries
⇒ **Dead spaces are perfect locations to hide viruses**



DOS header (Disk Operater System):

Entry point of a program invoked DOS command.

For most PE32 executables, the DOS header contains a tiny executable that prints: "This application must be run from Windows", then exits

File Injection Technical:

Goals:

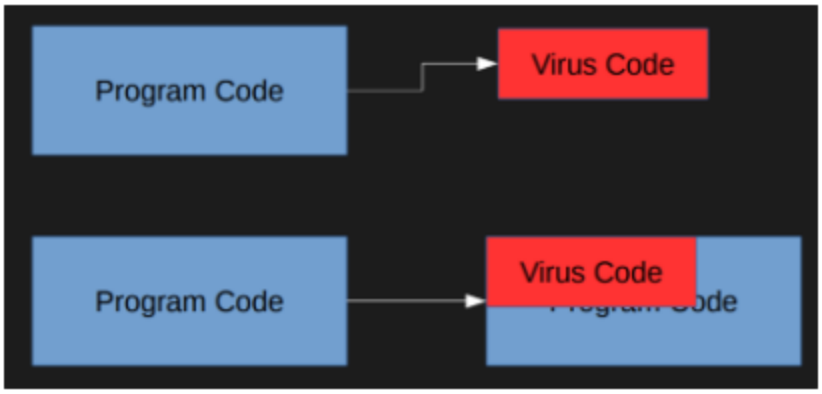
- Execute được code tấn công.
- Ẩn giấu: Giữ nguyên kích thước file, chức năng file. Giả vờ là 1 đoạn code bình thường để tránh các phần mềm anti-virus.

- Phân loại virus:

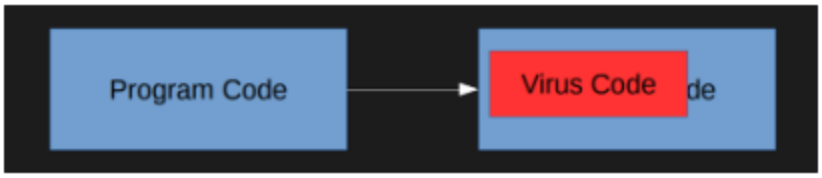
	Beginning	Random	Appending	Cavity
Where payload in file?	Beginning of executable file (preserve/overwriting current code)	Random place in executable file (Virus could be executed or not; executable file could run or crash)	In empty space larger than the virus code.	Spaces within a file that is filled with zeroes or ASCII blanks. (Cavity)
Method(s)	- Replace *.exe file with virus *.exe (Worst stealthy) - Overwrite only the beginning of a *.exe that is larger than the virus *.exe	- Overwrite a random place of a *.exe to the virus.	Copies the application code into a temporary file, then calls system() or a similar function to execute the contents of that file	- Replace 1 cavity big enough for a whole virus by the virus. - Distributing Virus into multiple small cavities. Loading virus into memory by virus loader code at the head of the virus, connected by jump instructions (a fractionated cavity virus) ⇒ Reach the start of the virus with a jump, or modify the PE entry point
Strength	Easy to implement	Easy to implement	Executing the original application successfully Not spending too long in virus code.	Note change size of file ⇒ More steathiness
Weakness	Destructiveness reduces the stealthiness of virus	Destructiveness reduces the stealthiness of virus; Could crash the program/not run virus.	The file size has changed ⇒ Reduces the stealthiness. The jump, or tricky jump, is easily spotted by anti-virus. Must pass original command-line arguments!	Jump, or modified PE entry point, detectable by anti-virus software; Disinfection can be difficult
Famous Representation	Love Letter	Russian Omud (8888)	Vienna and Suicide	

Examples pictures:

- Beginning:



- Random



- List

- Types:
- Detection:
- Anti-anti-Virus:

2. Trình bây sự khác biệt giữa packer, cryptor và protector trong ngữ cảnh sử dụng của các phần mềm độc hại.

   Trong ngữ cảnh sử dụng của các phần mềm độc hại, packer, cryptor và protector là các công cụ hoặc phương pháp được sử dụng để che giấu và bảo vệ mã độc. Dưới đây là sự khác biệt giữa chúng:

   Packer: Packer là một công cụ được sử dụng để nén và mã hóa mã độc, nhằm che giấu nội dung của nó và tránh bị phát hiện bởi các phần mềm chống vi-rút. Packer thường tạo ra một phiên bản nén của mã độc, và khi chương trình bị chạy, nó sẽ giải nén và khôi phục mã gốc để thực hiện các hoạt động độc hại. Mục đích chính của packer là làm cho mã độc trở nên khó phát hiện và phân tích.

   Cryptor: Cryptor là một công cụ được sử dụng để mã hóa mã độc, nhằm che giấu nội dung của nó và làm cho nó khó có thể đọc hoặc phân tích. Cryptor thường sử dụng các thuật toán mã hóa mạnh để biến đổi mã độc thành một dạng khác, chỉ có thể được giải mã bởi một khóa mật khẩu hoặc quy trình giải mã đặc biệt. Mục đích của cryptor là làm cho mã độc trở nên khó phát hiện, phân tích và phá vỡ mã hóa.

   Protector: Protector là một công cụ hoặc phương pháp được sử dụng để bảo vệ mã độc khỏi việc phân tích, thay đổi hoặc xâm nhập. Protector thường áp dụng các kỹ thuật như mã hóa, xáo trộn mã, kiểm tra tính toàn vẹn và sử dụng các cơ chế phòng thủ để ngăn chặn việc phát hiện và phân tích mã độc. Mục đích của protector là tăng cường độ bảo mật và khó khăn cho các nỗ lực phá vỡ hay tấn công vào mã độc.

   Tuy các công cụ và phương pháp này có thể được sử dụng bởi các phần mềm độc hại để che giấu và bảo vệ nội dung của chúng, nhưng cũng có thể được sử dụng trong các hoạt động bảo mật hợp lệ để bảo vệ phần mềm khỏi việc phân tích và tấn công không mong muốn.

Methods	Define	Purpose	Purpose in Malware
Packer (Trình đóng gói)	Runtime Packer (Excutable Compression) - Tự động giải nén tập tin con đã được đóng gói khi thực thi tập tin mẹ.	Làm cho kích thước tệp nhỏ hơn	Gây khó khăn cho các kĩ thuật dịch ngược. Ứng dụng trong compressing virus
Cryptor (Trình mã hóa)	Bao gồm: Obfuscation - Làm rối mã, Encryption - Mã hóa,...	Anti-debug Bảo vệ dữ liệu	Gây khó khăn cho các kĩ thuật dịch ngược và phân tích. Tăng tính ẩn giấu của mã độc.
Protector (Trình bảo vệ)	Kết hợp 2 kỹ thuật Packing+ Encrypting ⇒ Bảo vệ các đoạn code or Code virtualization: 1 tập instructions có thể thay đổi tùy môi trường	Ngăn chặn giả mạo (Tampering) Ngăn chặn dịch ngược (Anti-disassembly)	Bảo vệ payload ⇒ Anti-Reverse

3. Kỹ thuật tiêm tiến trình là gì? Kỹ thuật tấn công mã độc thông qua tiền trình ma (Process Hollowing) được mã độc dùng trong mục đích gì? nêu nguyên tắc thực hiện?

   Kỹ thuật tiêm tiến trình (Process Injection) là một phương pháp được sử dụng trong lĩnh vực phần mềm độc hại để tiêm mã độc vào một quy trình (tiến trình) đang chạy trên hệ thống mà không bị phát hiện. Kỹ thuật này cho phép mã độc chạy trong ngữ cảnh của quy trình được tiêm, tận dụng quyền hạn và tài nguyên của quy trình đó để thực hiện các hoạt động độc hại.

   Một trong các kỹ thuật tiêm tiến trình phổ biến là "Process Hollowing" hay còn gọi là "Process Replacement". Kỹ thuật này được mã độc sử dụng nhằm che giấu sự tồn tại của chính nó và thực hiện các hoạt động độc hại trong quy trình hợp lệ khác.

   Nguyên tắc thực hiện kỹ thuật Process Hollowing bao gồm các bước sau:

1 Chọn một quy trình hợp lệ: Mã độc chọn một quy trình hợp lệ đã được tạo ra hoặc sẵn có trên hệ thống để tiêm vào.

2 Tạo một bản sao trống của quy trình: Mã độc tạo ra một bản sao của quy trình mục tiêu mà không chứa nội dung của quy trình gốc, gọi là quy trình hollowed (quy trình bị rỗng).

3 Tiêm mã độc vào quy trình hollowed: Mã độc tiêm mã độc của nó vào quy trình hollowed, thay thế toàn bộ nội dung của quy trình này.

4 Khởi động lại quy trình hollowed: Mã độc khởi động lại quy trình hollowed bằng cách chạy mã độc mới đã được tiêm vào. Điều này cho phép mã độc chạy trong ngữ cảnh của quy trình hợp lệ mà không gây ngờ ngại hay bị phát hiện.

Kỹ thuật Process Hollowing cho phép mã độc che giấu mình bên trong một quy trình hợp lệ, làm cho việc phát hiện và phân tích mã độc trở nên khó khăn hơn. Nó cũng cho phép mã độc tận dụng các quyền hạn và tài nguyên của quy trình hợp lệ để thực hiện các hoạt động độc hại, chẳng hạn như thu thập thông tin, ghi log, gửi dữ liệu đến kẻ tấn công, hoặc thực hiện các hành động không mong muốn khác trên hệ thống.

4. kỹ thuật song trung tiến trình (Process Doppelganging) là gì? nêu các nguyên lý, cách thức thực hiện trong việc lây nhiễm mã độc trên máy tính.

   Kỹ thuật song trung tiến trình (Process Doppelganging) là một phương pháp tiêm mã độc vào hệ thống mà không để lại bất kỳ dấu vết hay phần mềm độc hại nào trên máy tính. Phương pháp này được sử dụng để đánh lừa các công cụ phân tích và các giải pháp bảo mật, gây khó khăn trong việc phát hiện và ngăn chặn mã độc.

   Nguyên lý của kỹ thuật Process Doppelganging là tận dụng khả năng của Transactional NTFS (TxF) để tạo ra một tiến trình giả mạo có cùng đặc điểm và hành vi như một tiến trình hợp lệ. Quá trình thực hiện bao gồm các bước sau:

1 Tạo một thư mục giao dịch: Mã độc tạo một thư mục giao dịch mới trên hệ thống tệp NTFS. Thư mục này được sử dụng như một không gian lưu trữ tạm thời để tiêm mã độc.

2 Tạo một tiến trình giả mạo: Mã độc tạo một tiến trình giả mạo mới bằng cách tạo một bản sao của một tiến trình hợp lệ bất kỳ. Quá trình này sử dụng các tài nguyên và thông tin từ tiến trình hợp lệ, nhưng không tạo ra bất kỳ tiến trình thực tế nào trên hệ thống.

3 Tiêm mã độc vào tiến trình giả mạo: Mã độc tiêm mã độc của mình vào tiến trình giả mạo thông qua việc thao tác với thư mục giao dịch. Quá trình này không để lại bất kỳ dấu vết hoặc tệp tin độc hại trên hệ thống tệp NTFS.

4 Khôi phục tiến trình giả mạo: Mã độc sử dụng Transactional NTFS để khôi phục tiến trình giả mạo, làm cho nó trở thành một tiến trình hợp lệ. Quá trình này gây khó khăn cho các công cụ phân tích và giải pháp bảo mật trong việc phát hiện hoặc ngăn chặn mã độc.

Kỹ thuật Process Doppelganging cho phép mã độc lây nhiễm vào máy tính mà không để lại bất kỳ dấu vết nào và che giấu mình trong một tiến trình hợp lệ. Điều này gây khó khăn cho việc phát hiện và phân tích mã độc, tạo ra một môi trường an toàn cho mã độc thực hiện các hoạt động độc hại như thu thập thông tin, truy cập hệ thống, hoặc tấn công khác.

5. Nêu khái niệm và trình bày sự khác biệt giữa dropper và dowrloader trong ngữ cảnh hoạt động của các chương trình độc hại. Nhận xét về tác động gây hai cũa hai loại này và phương pháp phòng chống đối với vấn đề an toàn thông tin

   Dropper và downloader là hai thuật ngữ được sử dụng trong ngữ cảnh hoạt động của các chương trình độc hại, nhưng có chức năng và tác động khác nhau. Dưới đây là khái niệm và sự khác biệt giữa dropper và downloader:

   1. Dropper:

   - Dropper (cũng được gọi là payload dropper hoặc malware dropper) là một thành phần của chương trình độc hại, chịu trách nhiệm cho việc gửi và triển khai payload (mã độc) lên hệ thống mục tiêu.

- Dropper thường được thiết kế để giấu payload và đảm bảo việc triển khai mã độc được thực hiện một cách ẩn danh và không bị phát hiện.
- Dropper có thể có nhiều hình thức thực hiện, bao gồm email độc hại, tải xuống từ các trang web độc hại, hoặc sử dụng các lỗ hổng bảo mật trong hệ điều hành hoặc các ứng dụng để tự động triển khai payload.

2. Downloader:

- Downloader là một loại chương trình độc hại được thiết kế để tải xuống và cài đặt các phần mềm độc hại (payload) từ máy chủ từ xa.
- Chức năng chính của downloader là tải về và triển khai mã độc từ nguồn từ xa vào hệ thống mục tiêu, thường thông qua kết nối mạng hoặc Internet.
- Downloader thường được sử dụng để cung cấp các phiên bản mới của phần mềm độc hại, bổ sung tính năng hoặc cập nhật payload hiện có, hoặc để tải xuống các module bổ sung để thực hiện các hoạt động độc hại khác.

Tác động của dropper và downloader:

- Dropper: Dropper chịu trách nhiệm cho việc triển khai mã độc lên hệ thống, tạo điều kiện cho các hoạt động độc hại sau này. Nó có thể mở cửa sau cho các chương trình độc hại khác, gây hại cho hệ thống bằng cách lây nhiễm, tiêm mã độc vào quá trình chạy hoặc thực hiện các hoạt động xâm nhập.
- Downloader: Downloader có tác động trực tiếp đến quá trình tải xuống và cài đặt mã độc lên hệ thống. Nó cho phép tấn công từ xa cung cấp mã độc mới hoặc các module bổ sung, mở rộng khả năng tấn công và kiểm soát của kẻ tấn công trên hệ thống.

Phòng chống và đối phó với vấn đề an toàn thông tin:

- Đối với dropper và downloader, các biện pháp phòng chống bao gồm:
    - Cập nhật và áp dụng bản vá bảo mật cho hệ điều hành và ứng dụng, nhằm giảm khả năng khai thác các lỗ hổng bảo mật.
    - Sử dụng phần mềm diệt malware và tường lửa mạnh mẽ để phát hiện và chặn các mối đe dọa tiềm ẩn.

- Cẩn trọng khi tải xuống và mở các tệp tin từ nguồn không đáng tin cậy, tránh nhấp vào liên kết không rõ nguồn gốc.
- Hạn chế quyền truy cập đối với các tệp tin và thư mục quan trọng trên hệ thống.
- Giáo dục người dùng về các nguy cơ và cách thức phòng chống tấn công từ phần mềm độc hại.

Lưu ý rằng việc duy trì các biện pháp bảo mật toàn diện và cập nhật là quan trọng để giảm thiểu nguy cơ từ dropper, downloader và các loại chương trình độc hại khác.

6. Trong bối cảnh của các chương trình độc hại, nêu các rủi ro về bảo mật và quyền riêng tư đối với các tài liệu Microsoft Office. Trình bày cách tin tặc thực hiện tấn công mã độc thông qua các dạng tài liệu Microsoft Office.

- Rủi ro về bảo mật và quyền riêng tư đối với các tài liệu Microsoft Office:
  - Tấn công vào tính năng File Sharing gây lộ/mất dữ liệu trong hệ thống.
  - Đánh cắp dữ liệu, thông tin xác thực bằng cách fishing email.
  - Dẫn tới các cuộc tấn công Ransomware ⇒ Tống tiền, mất dữ liệu...
  - Chỉnh sửa file từ Writable sang Read-only hoặc ngược lại
- Phương pháp *tấn công mã độc thông qua các dạng tài liệu Microsoft Office:*
  - VBA Macro Malware: Một cách phổ biến để tấn công mã độc trong các tài liệu Office là sử dụng macro malware. Tin tặc thường nhúng mã macro độc hại vào tài liệu Office, chẳng hạn như file Word hay Excel. Khi người dùng mở tài liệu và cho phép chạy macro, mã độc sẽ được kích hoạt, và nó có thể thực hiện các hành động độc hại như cài đặt phần mềm độc hại, thu thập thông tin cá nhân, hoặc mở cửa sau lưng (backdoor) để cho phép truy cập từ xa.
  - Giả mạo file (file exec → document OR file có macro nhưng bị sửa để không thấy có macro)

- Lợi dụng tính năng AutoRun (Tự động chạy các tệp đính kèm hoặc liên kết khi tài liệu được mở) của MS để tự động thực thi malicious file/link/embedded objects được nhúng vào tài liệu.

7. Thuật ngữ Process Injection (tiêm tiền trình) đùng cho mục đích gì? Nêu tên và giải thích nguyên của Process Injection?

   Thuật ngữ "Process Injection" (tiêm tiến trình) được sử dụng trong lĩnh vực an ninh mạng và phần mềm độc hại để chỉ một kỹ thuật mà trong đó mã độc tiêm vào một quá trình (tiến trình) đang chạy trên hệ thống máy tính mà không cần tạo ra một tiến trình riêng biệt. Mục đích chính của Process Injection là che giấu mã độc và lây nhiễm vào quá trình hợp lệ để thực hiện các hoạt động độc hại mà không bị phát hiện.

   - Có một số phương pháp tiêm tiến trình phổ biến được sử dụng trong Process Injection, bao gồm:
   - DLL Injection (Tiêm DLL): Mã độc tiêm các thư viện DLL (Dynamic Link Library) vào quá trình hợp lệ. Khi thực thi, quá trình hợp lệ sẽ tải và chạy DLL độc hại, cho phép kẻ tấn công kiểm soát và thực hiện các hoạt động độc hại trong quá trình đó.
   - Code Injection (Tiêm mã): Mã độc tiêm mã độc trực tiếp vào vùng bộ nhớ của quá trình hợp lệ. Mã độc này sau đó được thực thi như một phần của quá trình đó, cho phép kẻ tấn công kiểm soát và thực hiện các hoạt động độc hại.

   Process Hollowing (Tiêm rỗng tiến trình): Mã độc tạo ra một quá trình giả mạo và sau đó thay thế nội dung của quá trình giả mạo bằng mã độc. Điều này cho phép mã độc chạy trong bối cảnh của một quá trình hợp lệ, ẩn đi và gây khó khăn trong việc phát hiện và phân tích.

   Nguyên tắc chung của Process Injection là sử dụng các kỹ thuật và lỗ hổng trong hệ điều hành hoặc ứng dụng để tiêm mã độc vào quá trình hợp lệ. Bằng cách này, mã độc có thể che giấu trong quá trình hợp lệ và thực hiện các hoạt động độc hại mà không gây nghi ngờ hoặc bị phát hiện. Kỹ thuật Process Injection đòi hỏi kiến thức chuyên sâu về cấu trúc nội bộ của quá trình và các cơ chế hoạt động của hệ điều hành, và nó được sử dụng bởi các kẻ tấn công để tấn công hệ thống, ăn cắp thông tin, hoặc thực hiện các hoạt động độc hại khác trên máy tính nạn nhân.

8. EPO viruslà gì? Đặc điểm, mục đích của EPO virus. Nó bao gồm những loại nào? Trình bày chi tiết nguyên tắc của trường hợp TLS-EPO virus.

   EPO virus là một loại mã độc được phát triển để tấn công hệ thống máy tính bằng cách tận dụng các lỗ hổng trong quy trình chứng thực và mã hóa thông tin của giao thức TLS (Transport Layer Security). EPO viết tắt của "Encrypted Payload Overwrite" (Ghi đè Dữ liệu Mã hóa), chỉ ra cách thức hoạt động của loại virus này.

   Đặc điểm và mục đích của EPO virus:

   - EPO virus tập trung vào việc xâm nhập và tấn công giao thức bảo mật TLS, mà thường được sử dụng để bảo vệ sự an toàn và bí mật của dữ liệu khi truyền qua mạng.
   - Mục đích chính của EPO virus là đánh lừa và xâm nhập vào quá trình mã hóa và giải mã của giao thức TLS để chèn và ghi đè dữ liệu độc hại vào trong giao thức đã được mã hóa, mà không làm thay đổi kích thước hay tính toàn vẹn của gói tin đã mã hóa.

   Có một số loại EPO virus phổ biến, bao gồm:

   1. TLS-EPO virus: Đây là một dạng EPO virus tấn công trực tiếp vào giao thức TLS. Nó khai thác các lỗ hổng trong quá trình chứng thực và mã hóa TLS để chèn dữ liệu độc hại vào trong giao thức mã hóa TLS mà không làm thay đổi cấu trúc của gói tin.

   Nguyên tắc của trường hợp TLS-EPO virus:

   1. Xác định gói tin TLS: EPO virus phải xác định gói tin TLS cần tấn công để chèn dữ liệu độc hại vào.
   2. Phân tích cấu trúc gói tin TLS: EPO virus phân tích cấu trúc của gói tin TLS, bao gồm tiêu đề và dữ liệu đã được mã hóa.
   3. Xác định vị trí và kích thước: EPO virus xác định vị trí và kích thước của dữ liệu cần ghi đè trong gói tin TLS, đảm bảo rằng ghi đè không làm thay đổi kích thước hay tính toàn vẹn của gói tin.
   4. Tiến hành ghi đè: EPO virus chèn và ghi đè dữ liệu độc hại vào vị trí đã xác định trong gói tin TLS mà không làm thay đổi kích thước hay tính toàn vẹn của gói tin.

5. Đóng gói và chuyển tiếp: EPO virus đóng gói lại gói tin TLS đã bị tấn công và tiếp tục chuyển tiếp đến đích mà không làm phát hiện hoặc gây ngờ vực.

Việc sử dụng trường hợp TLS-EPO virus cho phép kẻ tấn công chèn và truyền dữ liệu độc hại qua các kết nối bảo mật TLS mà không làm gián đoạn sự truyền tải và làm phát hiện bởi các biện pháp bảo mật thông thường. Điều này tạo ra một mối đe dọa nghiêm trọng đối với tính bảo mật của giao thức TLS và đòi hỏi các biện pháp phòng chống và cải tiến liên quan đến việc xử lý gói tin TLS để ngăn chặn tấn công EPO virus.

9. Trình bày mục đích của các phương pháp tạo mã độc đột biến, cho biết sự khác nhan giữa các chiến lược tạo biển thể mã độc?

Các phương pháp tạo mã độc đột biến nhằm mục đích thay đổi cấu trúc, hình thức và tính chất của mã độc để tránh phát hiện và phòng ngừa từ các phần mềm diệt malware và các biện pháp bảo mật khác. Mục đích chính của các phương pháp này là gia tăng khả năng xâm nhập và tồn tại của mã độc trong hệ thống.

Các chiến lược tạo biển thể mã độc khác nhau có sự khác biệt về cách thức thực hiện và phạm vi thay đổi mã độc. Dưới đây là một số sự khác nhau chính giữa các chiến lược tạo biển thể mã độc:

- Mã hóa mã độc (Code Encryption): Phương pháp này sử dụng mã hóa để mã hóa toàn bộ hoặc một phần của mã độc. Mã độc sẽ được giải mã và thực thi trong quá trình chạy, giúp che giấu hình thức và tính chất của mã độc. Mục đích làm như vậy là để tránh phát hiện dựa trên chữ ký hoặc nhận dạng chuỗi mã độc.
- Tạo mã ngẫu nhiên (Random Code Generation): Phương pháp này tạo ra các phiên bản mã độc ngẫu nhiên mỗi lần mã độc được triển khai. Các biến thể mã độc này sẽ có cấu trúc và hình thức khác nhau, làm khó cho các công cụ phân tích và phần mềm diệt malware nhận dạng và phòng ngừa.
- Tạo mô-đun (Module Generation): Phương pháp này tạo ra các mô-đun mã độc khác nhau, và trong quá trình tấn công, mã độc sẽ tải và kết hợp các mô-đun này thành một hình thức hoàn chỉnh để thực

thi. Sự thay đổi trong cấu trúc và chức năng của các mô-đun khiến việc phân tích và phát hiện trở nên khó khăn hơn.

- Tạo mã độc đa dạng (Polymorphic Code Generation): Phương pháp này tạo ra các phiên bản mã độc khác nhau từ cùng một mã nguồn gốc. Sự đa dạng này được đạt được thông qua việc thay đổi cấu trúc, luồng điều khiển, và các thành phần khác của mã độc, trong khi vẫn duy trì chức năng và mục tiêu của nó. Điều này làm tăng khả năng tồn tại và tránh phát hiện từ các công cụ phân tích tự động.

Mục đích chung của các phương pháp tạo biển thể mã độc là che giấu mã độc, làm khó cho các công cụ phân tích và phần mềm diệt malware phát hiện và phòng ngừa. Bằng cách thay đổi mã độc, tạo biển thể và đa dạng hóa, mã độc có thể tránh được các biện pháp phòng ngừa thông thường và tồn tại trong hệ thống máy tính một cách lâu dài.

10. Trình báy cấu trúc tập tin PDF, các chiến lược chèn các đoạn mã độc hại vào tập tin PDF và khả năng tấn công trên các loại kỹ thuật này

Cấu trúc tập tin PDF (Portable Document Format) là một định dạng tập tin được sử dụng phổ biến cho việc chia sẻ tài liệu điện tử. Tập tin PDF bao gồm các phần chính sau:

1. Header: Phần đầu của tập tin PDF chứa thông tin về phiên bản của định dạng PDF và các thông số khác như mã hóa, nén, font chữ sử dụng, và thuộc tính tài liệu.
2. Body: Phần chính của tập tin PDF chứa các đối tượng và nội dung của tài liệu. Các đối tượng bao gồm các trang, hình ảnh, văn bản, liên kết, và các tài nguyên khác.
3. Cross-Reference Table: Bảng tham chiếu chéo (Cross-Reference Table) chứa thông tin về vị trí của các đối tượng trong tập tin PDF. Nó giúp quá trình đọc và tìm kiếm nhanh chóng trong tập tin PDF.
4. Trailer: Phần cuối cùng của tập tin PDF chứa các thông tin khác như kích thước của tài liệu, danh sách các đối tượng, và khối mã xác thực.

Các chiến lược chèn đoạn mã độc vào tập tin PDF nhằm tấn công người dùng khi mở tập tin này. Dưới đây là một số chiến lược chèn mã độc vào tập tin PDF:

1. Sử dụng kỹ thuật JavaScript: Tập tin PDF có thể chứa mã JavaScript để thực hiện các hành động độc hại khi tài liệu được mở. Mã JavaScript có thể được chèn vào các sự kiện, liên kết, hoặc nút trong tập tin PDF để khai thác các lỗ hổng trong trình đọc PDF và tấn công hệ thống.
2. Sử dụng kỹ thuật mã hóa: Mã độc có thể được mã hóa hoặc nén để tránh phát hiện bằng các công cụ phân tích tự động. Mã độc sẽ được giải mã hoặc giải nén trong quá trình thực thi tập tin PDF.
3. Sử dụng kỹ thuật hình ảnh động: Một số tập tin PDF có thể chứa các hình ảnh động, ví dụ như GIF hoặc Flash, để tấn công người dùng khi tài liệu được xem. Hình ảnh động có thể chứa đoạn mã độc hoặc khai thác các lỗ hổng trong trình đọc.
4. Sử dụng kỹ thuật tệp đính kèm: Tập tin PDF có thể chứa các tệp đính kèm, ví dụ như tệp tin thực thi (executable), để tấn công khi tài liệu được mở. Khi người dùng mở tệp đính kèm, đoạn mã độc sẽ được thực thi trên hệ thống.
5. Nhúng URL: Các tệp PDF độc hại có thể chứa các URL được nhúng trỏ đến các trang web độc hại hoặc các tệp có thể tải xuống. Khi người dùng nhấp vào một liên kết như vậy trong PDF, nó có thể dẫn đến việc tải xuống và thực thi phần mềm độc hại trên hệ thống của họ.
6. Nhúng phần mềm độc hại: Có thể nhúng phần mềm độc hại vào trong file PDF, chẳng hạn như các tệp hoặc tập lệnh thực thi. Khi tệp PDF được mở, phần mềm độc hại nhúng sẽ được thực thi (AutoRun), dẫn đến lây nhiễm hoặc xâm phạm hệ thống.

Các kỹ thuật tấn công trên các loại kỹ thuật này nhằm lợi dụng các lỗ hổng trong trình đọc PDF hoặc các ứng dụng liên quan để thực thi mã độc và tấn công hệ thống. Điều này có thể dẫn đến việc xâm nhập hệ thống, lây lan mã độc, đánh cắp thông tin, hoặc gây hại khác cho người dùng và hệ thống.

- *Khả năng tấn công trên các loại kỹ thuật này:*

- Các tập tin PDF có thể bị khai thác để thực hiện các cuộc tấn công từ xa, như lợi dụng các lỗ hổng trong trình đọc PDF để thực thi mã độc hại.
- Mã độc hại có thể tiếp cận và kiểm soát các tài liệu PDF, gây nguy hiểm cho hệ thống và thông tin cá nhân của người dùng.
- Các hành động độc hại trong tập tin PDF có thể tạo ra các tác động phụ như gửi thông tin nhạy cảm đến các bên thứ ba hoặc thực hiện các hành động độc hại trên hệ thống người dùng.

Để phòng chống tấn công từ mã độc trong tập tin PDF, cần tuân thủ các biện pháp an ninh thông tin như:

- Cập nhật các phần mềm liên quan đến trình đọc PDF và ứng dụng hỗ trợ để bảo đảm rằng các lỗ hổng đã được vá.
- Sử dụng phần mềm diệt malware và tường lửa để ngăn chặn và phát hiện các tập tin PDF độc hại.
- Kiểm tra và xác thực nguồn gốc của tập tin PDF trước khi mở.
- Hạn chế việc mở các tập tin PDF không tin cậy hoặc gửi từ nguồn không rõ.
- Thực hiện việc giáo dục và tạo nhận thức để người dùng nhận biết và tránh mở các tập tin PDF độc hại.

11. Kỹ thuật Environmental Keying là gì? Trình bày mục đích và các kỹ thuật thực hiện trong các chương trình phần mềm chứa mã độc hại?

Kỹ thuật Environmental Keying là một phương pháp trong lĩnh vực tấn công mạng, trong đó mã độc sử dụng thông tin về môi trường hoạt động của hệ thống (environment) để thực hiện các hành động độc hại. Thông tin môi trường bao gồm các thông số, cấu hình và trạng thái của hệ điều hành, phần cứng, phần mềm và môi trường mạng.

Mục đích chính của kỹ thuật Environmental Keying trong các chương trình phần mềm chứa mã độc hại là tăng tính linh hoạt và khả năng tấn công, đồng thời giảm khả năng phát hiện và phòng ngừa từ các biện pháp bảo mật.

Dưới đây là một số kỹ thuật thực hiện kỹ thuật Environmental Keying trong các chương trình phần mềm chứa mã độc hại:

- Phân tích môi trường: Mã độc sẽ tiến hành thu thập thông tin về môi trường hệ thống, bao gồm hệ điều hành, phiên bản phần mềm, cấu hình mạng, địa chỉ IP, v.v. Thông tin này được sử dụng để xác định các điều kiện và cấu hình hệ thống để điều chỉnh hành vi của mã độc.
- Giải mã và chạy mã độc: Mã độc sẽ sử dụng thông tin môi trường đã thu thập để giải mã các phần của mã độc hoặc thực thi các chức năng cụ thể. Việc giải mã hoặc thực thi sẽ được thực hiện dựa trên sự tương thích và điều kiện của môi trường hệ thống.
- Điều khiển từ xa: Mã độc có thể tạo kết nối từ xa đến môi trường hệ thống thông qua các giao thức mạng như TCP/IP. Điều này cho phép kẻ tấn công tương tác với mã độc, cập nhật và điều khiển nó từ xa để thực hiện các hành động độc hại.
- Tự động thích ứng: Mã độc có thể sử dụng thông tin môi trường để tự động thích ứng và điều chỉnh hành vi của nó. Ví dụ, nếu một phần mềm diệt malware cụ thể được phát hiện trên hệ thống, mã độc có thể thay đổi cách hoạt động của mình để tránh phát hiện từ phần mềm này.

Kỹ thuật Environmental Keying tạo ra sự linh hoạt và khó phát hiện đối với mã độc, khiến nó trở nên khó chống lại và phòng ngừa. Để đối phó với kỹ thuật này, cần thực hiện các biện pháp bảo mật như cập nhật và vá lỗ hổng hệ thống, sử dụng phần mềm diệt malware hiện đại, theo dõi và giám sát hoạt động của hệ thống một cách cẩn thận để phát hiện các hoạt động đáng ngờ.

12. Trình bày các kỹ thuật chống phân tích động trong các chương trình độc hại

Các kỹ thuật chống phân tích động (dynamic analysis evasion techniques) được sử dụng trong các chương trình độc hại nhằm gài bẫy hoặc tránh phát hiện và phân tích bởi các công cụ và môi trường phân tích động. Dưới đây là một số kỹ thuật phổ biến trong việc chống lại phân tích động của các chương trình độc hại:

- Phát hiện môi trường: Mã độc có thể kiểm tra môi trường thực thi của mình để xác định liệu nó đang chạy trong một môi trường phân tích hay một môi trường thực tế. Mã độc có thể kiểm tra các thông số như địa chỉ IP, cấu hình hệ thống, các tiến trình đang chạy, sự hiện diện của các công cụ phân tích động, v.v. Dựa vào kết quả này, mã độc có thể thay đổi hành vi hoặc ẩn tránh phân tích.
- Kiểm tra môi trường giả lập: Mã độc có thể kiểm tra các đặc điểm của môi trường giả lập, chẳng hạn như một máy ảo hay môi trường giả lập hành vi. Mã độc có thể phát hiện các biểu hiện của môi trường giả lập và tương tác khác biệt, và từ đó thay đổi hành vi của mình để tránh phân tích.
- Kiểm tra hoạt động mạng: Mã độc có thể kiểm tra hoạt động mạng của hệ thống để xác định liệu nó đang chạy trong một môi trường giám sát hay một mạng thực tế. Mã độc có thể kiểm tra các thông tin như địa chỉ IP, cổng mạng, lưu lượng mạng, v.v. Dựa vào kết quả này, mã độc có thể thay đổi hoạt động mạng của mình để tránh phát hiện và phân tích.
- Chờ đợi thời gian: Mã độc có thể chờ đợi một khoảng thời gian ngắn trước khi thực hiện các hành động độc hại. Điều này có thể làm cho mã độc tránh được các công cụ phân tích động vì các công cụ này thường thực hiện quá trình phân tích trong một thời gian ngắn.
- Tương tác người dùng giả: Mã độc có thể tạo ra các cửa sổ giả mạo hoặc yêu cầu tương tác người dùng để làm cho việc phân tích trở nên khó khăn. Điều này có thể bao gồm việc hiển thị thông báo lừa đảo, yêu cầu nhập liệu hoặc xác nhận, v.v.

Các kỹ thuật chống phân tích động nhằm gây khó khăn cho quá trình phân tích và phát hiện mã độc. Tuy nhiên, các công cụ phân tích động và kỹ thuật phòng ngừa mã độc tiến bộ cũng ngày càng cải thiện để đối phó với các kỹ thuật này.

13. Để chống phân tích tĩnh, chương trình mã độc sử đụng những chiến lược nào trong mã nguồn của nó?

Để chống phân tích tĩnh (static analysis), chương trình mã độc sử dụng một số chiến lược nhằm làm khó khăn hoặc ngăn cản quá trình phân tích mã nguồn của nó. Dưới đây là một số chiến lược phổ biến được sử dụng trong mã độc để chống phân tích tĩnh:

- Mã hóa: Chương trình mã độc có thể sử dụng các kỹ thuật mã hóa để ẩn đi mã nguồn của nó. Điều này làm cho mã độc trở nên khó đọc và hiểu, gây khó khăn cho quá trình phân tích.
- Nén: Mã độc có thể được nén bằng các thuật toán nén để giảm kích thước và khó khăn trong việc phân tích. Trước khi thực thi, chương trình sẽ được giải nén để khôi phục mã nguồn ban đầu.
- Gắn kết động (Dynamic linking): Chương trình mã độc có thể tận dụng tính năng gắn kết động để chờ đợi và tải các thư viện hoặc mã nguồn bổ sung từ các nguồn bên ngoài. Điều này làm cho việc phân tích tĩnh trở nên khó khăn hơn, vì mã độc có thể không hoàn chỉnh hoặc không chứa đủ thông tin để phân tích một cách đầy đủ.
- Anti-Debugging: Mã độc có thể chứa các kỹ thuật anti-debugging để phát hiện và ngăn chặn quá trình phân tích từ các công cụ debug. Điều này có thể bao gồm việc kiểm tra sự hiện diện của các công cụ debug, thay đổi luồng thực thi để tránh bị theo dõi, hoặc gây lỗi hoặc chết chương trình khi phát hiện sự can thiệp của công cụ debug.
- Obfuscation: Mã độc có thể sử dụng kỹ thuật obfuscation để thay đổi cấu trúc và logic của mã nguồn một cách phức tạp. Điều này làm cho mã độc khó hiểu và khó phân tích, gây khó khăn cho việc tìm hiểu cách hoạt động của nó.

Những chiến lược này nhằm làm khó khăn và ngăn cản quá trình phân tích tĩnh của chương trình mã độc. Tuy nhiên, các công cụ phân tích tĩnh và kỹ thuật phân tích tiến bộ cũng ngày càng cải thiện để vượt qua những chiến lược này và phát hiện mã độc một cách hiệu quả.