**Ôn tập**

**Cơ chế hoạt động của mã độc**

**------**

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

*Trả lời:*

* Virus

Là một chương trình phần mềm được gắn vào các file hoặc phần mềm khác, có khả năng tự nhân bản và lây lan sang các máy tính mà không cần sự cho phép của người dùng. Mục đích gây hại, xóa dữ liệu, phá hoại hệ thống hoặc đánh cắp thông tin

* Worm

Cũng giống như virus nhưng khác là *tự động lây lan* qua mạng máy tính và không cần gắn vào file hoặc phần mềm để lây lan. Worm sẽ tìm lỗ hổng bảo mật trong hệ thống đề xâm nhập và lây nhiễm vào máy tính khác. Worm có thể gây ra tổn hại lớn cho mạng bằng cách tiêu tốn tài nguyên hệ thống và băng thông.

* Botnet

Là mạng lưới các máy tính (bot) bị nhiễm virus, worm hoặc malware khác và bị điều khiển từ xa bởi hacker. Những máy tính trong botnet được dùng để thực hiện DDoS, phát tán spam hoặc khai thác tài nguyên của máy tính.

* Ransomware

Là một loại mã độc khóa hoặc mã hóa dữ liệu của nạn nhân và yêu cầu một khoản tiền chuộc để khôi phục quyền truy cập vào dữ liệu hoặc hệ thống. Thường lan truyền bằng mail phising, tệp đính kèm hoặc từ các trang web không an toàn.

* Easter egg

Là một thông điệp hoặc tính năng được ẩn trong trò chơi hoặc các phần mềm. Easter egg có thể mang tính giải trí nhưng bất kỳ thứ gì mà chương trình không cần thì không nên thêm vào.

* Salami attack

Là một kỹ thuật tấn công trong đó kẻ tấn công thực hiện một loạt các hành động nhỏ và hầu như không đáng chú ý, nhưng khi cộng lại có thể gây thiệt hại lớn. Thường được sử dụng trong các cuộc tấn công tài chính, chẳng hạn như ăn cắp một lượng tiền nhỏ từ nhiều tài khoản.

* Backdoor

Là cơ chế bí mật trong phần mềm, OS hoặc thiết bị cho phép user hoặc attacker truy cập vào hệ thống mà không cần các phương thức xác thực. Thường dùng để truy cập và kiểm soát hệ thống từ xa mà không cần sự cho phép. Backdoor thường được tạo ra bởi các nhà phát triển để giải quyết các vấn đề bảo trì nhưng có thể bị kẻ tấn công lợi dụng.

* Rootkit

Là tập hợp các công cụ và chương trình độc hại được sử dụng để ẩn đi sự tồn tại của attacker hoặc các hoạt động xâm nhập. Rootkit thường thay đổi các thành phần hệ thống hoặc ghi đè các chương trình hệ thống để che giấu hoạt động độc hại.

* Logic bomb

Là một đoạn mã độc được kích hoạt khi một điều kiện cụ thể được thỏa mãn..

* Time-bomb

Là một loại logic bomb được cài đặt để kích hoạt vào một thời điểm cụ thể trong tương lai.

* Trojan

Còn được biết đến là trojan horse, là một loại mã độc ngụy trang dưới dạng phần mềm hợp pháp để lừa người dùng cài đặt và thực thi. Một khi được cài đặt, trojan có thể thực hiện nhiều hành động gây hại khác nhau như đánh cắp thông tin, cài đặt thêm mã độc, hoặc tạo backdoor.

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

*Trả lời:*

Trong ngữ cảnh các phần mềm độc hại, packer, cryptor, và protector là ba công cụ phổ biến được sử dụng để ẩn dấu hoặc bảo vệ mã độc khỏi bị phát hiện và phân tích.

* Packer

Packer là một công cụ nén tệp thực thi để giảm kích thước của nó. Nó thường được sử dụng để đóng gói các phần mềm độc hại nhằm làm khó khăn cho việc phân tích tĩnh bằng cách thay đổi cấu trúc tệp.

* Cryptor

Cryptor là một công cụ mã hóa tệp thực thi để bảo vệ nội dung của nó. Crypto được sử dụng để làm cho việc phân tích mã nguồn khó khăn hơn, bằng cách mã hóa nội dung tệp, nhằm tránh bị phát hiện và phân tích bởi các chuyên gia bảo mật và phần mềm diệt virus.

* Protector

Protector là một công cụ bảo vệ toàn diện hơn, thường bao gồm cả tính năng nén (packer) và mã hóa (cryptor), nhưng còn thêm nhiều biện pháp bảo vệ khác như chống gỡ lỗi, chống phân tích ngược và chống giả lập.

|  |  |  |  |
| --- | --- | --- | --- |
| **Đặc điểm** | **Packer** | **Cryptor** | **Protector** |
| Chức năng chính | Nén tệp thực thi | Mã hóa tệp thực thi | Kết hợp nén, mã hóa và các biện pháp bảo vệ khác |
| Phương thức bảo vệ | Giảm kích thước, thay đổi dấu vết | Mã hóa nội dung để làm khó khăn cho việc phân tích | Tích hợp nén, mã hóa và chống gỡ lỗi, chống phân tích ngược |
| Mức độ bảo vệ | Cơ bản, tập trung vào kích thước và dấu vết | Trung bình, tập trung vào làm khó khăn việc phân tích | Toàn diện, kết hợp nhiều biện pháp bảo vệ và chống lại kỹ thuật phân tích. |

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

*Trả lời:*

**Kỹ thuật tiêm tiến trình (process injection)** là một phương pháp được sử dụng trong an ninh mạng, đặc biệt bởi các phần mềm độc hại và các tấn công mạng, để chạy mã độc trong không gian tiến trình của một ứng dụng hợp pháp. Điều này giúp mã độc hoạt động mà không bị phát hiện vì nó ẩn nấp bên trong một tiến trình tin cậy.

Cách hoạt động của kỹ thuật tiêm tiến trình:

1. Mở tiến trình mục tiêu
2. Cấp phát bộ nhớ trong tiến trình mục tiêu
3. Tiêm mã độc vào bộ nhớ đã cấp phát
4. Chạy mã độc trong tiến trình mục tiêu

Các phương pháp phổ biến của kỹ thuật tiêm tiến trình

* DLL Injection: kỹ thuật này tiêm một Dynamic Link Library (DLL) vào không gian bộ nhớ của tiến trình mục tiêu. Sau đó, DLL này sẽ được tiến trình mục tiêu tải và thực thi.
* APC Injection: APC (Asynchronous Procedure Call) Injection sử dụng các APC để tiêm và thực thi mã độc trong một thread của tiến trình mục tiêu.
* Thread Hijacking: kỹ thuật này thay đổi luồng (thread) của tiến trình mục tiêu để thực thi mã độc, sau đó khôi phục luồng để tiến trình tiếp tục hoạt động bình thường.
* Process Hollowing: kỹ thuật này tạo một tiến trình mới, sau đó loại bỏ nội dung của tiến trình này và thay thế bằng mã độc, sau đó tiến hành chạy mã độc trong vỏ của tiến trình hợp pháp.

**Process Hollowing** là một kỹ thuật tấn công mà mã độc sử dụng để chạy mã độc bên trong một tiến trình hợp pháp nhằm tránh bị phát hiện bởi các phần mềm bảo mật và hệ thống phát hiện xâm nhập (IDS). Kỹ thuật này được sử dụng với các mục đích sau:

* Ẩn giấu mã độc: Mã độc ẩn mình bên trong tiến trình hợp pháp để tránh bị phát hiện bởi các phần mềm diệt virus và các hệ thống bảo mật khác.
* Chạy mã với quyền hạn cao: Nếu tiến trình hợp pháp có quyền truy cập cao, mã độc sẽ tận dụng quyền hạn này để thực thi các hoạt động độc hại.
* Tránh giám sát và phân tích: Bằng cách hoạt động trong ngữ cảnh của một tiến trình tin cậy, mã độc làm cho việc giám sát và phân tích hành vi trở nên khó khăn hơn.

Nguyên tắc thực hiện Process Hollowing:

* Tạo tiến trình mới trong trạng thái tạm dừng (Suspended Process)
* Gỡ bỏ mã thực thi của tiến trình hợp pháp
* Tiêm mã độc vào tiến trình
* Thiết lập các điểm nhập (Entry Point) cho mã độc
* Chạy mã độc trong tiến trình hợp pháp

**4. Kỹ thuật song trùng tiến trình (Process Doppelgänging) là gì? Nêu các nguyên lý, cách thức thực hiện trong việc lây nhiễm mã độc trên máy tính.**

*Trả lời:*

**Process Doppelgänging** là một kỹ thuật khai thác cách hệ điều hành Windows xử lý các giao dịch tệp (file transaction) để thực thi mã độc mà không để lại dấu vết trong hệ thống tệp hoặc không bị phát hiện bởi các phần mềm bảo mật.

Nguyên lý thực hiện Process Doppelgänging:

Kỹ thuật Process Doppelgänging tận dụng các giao dịch tệp (file transactions) của hệ điều hành Windows để che giấu mã độc. Quá trình này bao gồm việc tạo một giao dịch tệp sử dụng tính năng NTFS Transactional NTFS (TxF), tạo và ghi đè mã độc vào một tệp tạm thời trong giao dịch này. Sau đó, giao dịch được hoàn tác, khiến mã độc không được ghi vào hệ thống tệp thực tế. Tiến trình từ tệp đã bị thay thế được tạo ra từ bộ nhớ chia sẻ mà không sử dụng các hàm API truyền thống thường bị giám sát chặt chẽ như NtUnmapViewOfSection, VirtualProtectEx, và SetThreadContext.

Kỹ thuật song trùng tiến trình (Process Doppelgänging) được thực hiện qua 4 bước:

1. Giao dịch (Transact)

Tạo một giao dịch TxF (Windows Transactional NTFS) sử dụng một tệp thực thi hợp pháp, sau đó ghi đè tệp này bằng mã độc. Các thay đổi này sẽ bị cô lập và chỉ hiển thị trong ngữ cảnh của giao dịch.

1. Tải (Load)

Tạo một vùng bộ nhớ chia sẻ và tải tệp thực thi độc hại vào đó.

1. Hoàn tác (Rollback)

Hoàn tác các thay đổi đối với tệp thực thi gốc, thực chất là loại bỏ mã độc khỏi hệ thống tệp.

* Kích hoạt (Animate)

Tạo một tiến trình từ vùng bộ nhớ bị nhiễm và bắt đầu thực thi mã độc.

\* Bổ sung:

Windows Transactional NTFS (TxF) được giới thiệu trên Vista như một phương pháp để thực hiện các thao tác tệp an toàn. Để đảm bảo tính toàn vẹn của dữ liệu, TxF chỉ cho phép một handle giao dịch ghi vào tệp tại một thời điểm. Cho đến khi giao dịch ghi này kết thúc, tất cả các handle khác đều bị cô lập khỏi handle đang ghi và chỉ có thể đọc phiên bản đã cam kết của tệp tồn tại vào thời điểm handle được mở. Để tránh hư hỏng, TxF thực hiện hoàn tác tự động nếu hệ thống hoặc ứng dụng gặp sự cố trong quá trình giao dịch ghi.

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

*Trả lời:*

Định nghĩa

* Dropper

Dropper là một loại phần mềm độc hại được thiết kế để cài đặt hoặc "thả" (drop) phần mềm độc hại khác lên hệ thống bị nhiễm (thường là cài đặt rootkit).

Dropper là một phương tiện để kết thúc quá trình lây nhiễm rootkit, chứ nó không phải là mục tiêu cần được thực hiện cuối cùng. Điều này được thực hiện bằng cách cài đặt các chương trình gây hại khác trên hệ thống máy tính mục tiêu. Khi dropper được thực thi, nó sẽ giải nén các đoạn mã độc hại lên đĩa cứng lưu trữ máy tính. Những chương trình độc hại này một khi được cài vào sẽ làm công việc phá hoại máy tính nạn nhân. Trên thực tế, những trình dropper như vậy thường sẽ tự xóa sau khi nó cài đặt đồng phạm.

Một dropper thường nhúng các đoạn mã nhị phân vào trong phần Resource của tập tin thực thi. Để trích xuất, giải nén được các tập tin thực thi trong phần Resource của Dropper, nó sử dụng các lời gọi hàm API như là FindResource(), LoadResource(), LockResource() và SizeOfResource() .

* Downloader

Downloaders đơn giản là tải xuống một phần mềm độc hại khác từ Internet và thực thi nó trên hệ thống cục bộ.

Downloaders thường được đóng gói kèm với một khai thác (exploit). Downloaders thường sử dụng Windows API URLDownloadtoFile(), tải tập tin xuống đĩa cứng máy tính. Sau khi tải xuống, nó sẽ sử dụng các lệnh gọi API CHellExecute(), WinExec() hoặc CreatProcess() để thực thi các thành phần đã tải xuống. Thông thường chúng ta sẽ thấy các trình Downloader được sử dụng như một phần của shellcode.

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Dropper** | **Downloader** |
| **Các thức hoạt động** | Giải nén và cài đặt mã độc hại từ bên trong lên hệ thống mục tiêu. Thường tự xóa sau khi hoàn thành. | Kết nối với máy chủ từ xa để tải xuống phần mềm độc hại và thực thi. Sử dụng API của Windows để thực hiện. |
| **Kích thước** | Thường lớn hơn vì chứa sẵn phần mềm độc hại. | Thường nhỏ gọn và đơn giản hơn. |
| **Độ phức tạp** | Có thể phức tạp hơn do cần mã hóa và giấu kín phần mềm độc hại. | Phức tạp ở chỗ phải đảm bảo kết nối với máy chủ từ xa và vượt qua bảo mật mạng. |
| **Phương thức phán tán** | Qua tập tin đính kèm email, phần mềm giả mạo, hoặc các phương thức khác mà người dùng có thể tải về và mở. | Qua tập tin đính kèm email, liên kết độc hại trên web, hoặc ẩn trong các phần mềm hợp pháp bị xâm nhập. |
| **Cách phòng chống** | Sử dụng phần mềm diệt virus và cập nhật thường xuyên; Hạn chế tải và cài đặt phần mềm từ nguồn không tin cậy; Kiểm tra tính toàn vẹn của phần mềm trước khi cài đặt. | Sử dụng tường lửa để chặn các kết nối không mong muốn; Cập nhật phần mềm diệt virus; Giáo dục người dùng về rủi ro khi nhấp vào liên kết không xác định hoặc tải tệp từ nguồn không rõ ràng. |

**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.**

*Trả lời:*

Rủi ro:

* Macro độc hại

Các macro là các đoạn mã được viết bằng VBA (Visual Basic for Applications) và có thể được nhúng vào các tài liệu Office. Macro có thể được sử dụng để tự động hóa các tác vụ, nhưng cũng có thể bị lợi dụng để thực hiện các hành động độc hại như tải và thực thi mã độc, thay đổi tệp tin, hoặc truy cập thông tin nhạy cảm.

* OLE (Object Linking and Embedding)

OLE cho phép nhúng và liên kết các đối tượng trong tài liệu. Tin tặc có thể sử dụng OLE để nhúng các tệp thực thi độc hại trong tài liệu Office.

* Tệp tin đính kèm trong email

Tài liệu Office đính kèm trong email có thể chứa mã độc. Khi người dùng mở các tệp đính kèm này, mã độc có thể được kích hoạt.

* Khai thác lỗ hổng phần mềm

Các lỗ hổng bảo mật trong ứng dụng Microsoft Office có thể bị khai thác để thực thi mã độc.

Cách tin tặc thực hiện tấn công mã độc thông qua các dạng tài liệu Microsoft Office:

* Chuẩn bị mã độc
* Tạo tài liệu office
* Nhúng mã độc vào tài liệu (sử dụng các cách nêu trên)
* Phân phối tài liệu và dụ dỗ người dùng
* Kích hoạt mã độc

**7. Thuật ngữ Process Injection (tiêm tiến trình) dùng cho mục đích gì? Nêu tên và giải thích nguyên tắc thực hiện của 03 kỹ thuật phổ biến của Process Injection?**

***Trả lời:***

Process Injection (tiêm tiến trình) là một phương pháp chèn mã độc vào tiến trình đang hoạt động trong hệ thống với mục tiêu thực thi mã tùy ý trong không gian địa chỉ của một tiến trình đó. Chạy mã trong ngữ cảnh của một tiến trình khác có thể cho phép truy cập vào bộ nhớ của tiến trình đó, các tài nguyên hệ thống/mạng và có thể là đặc quyền cao hơn. Việc thực thi thông qua tiêm tiến trình cũng có thể né tránh sự phát hiện từ các sản phẩm bảo mật vì việc thực thi được ẩn dưới một tiến trình hợp pháp.

03 kỹ thuật phổ biến của Process Injection:

1. Dynamic-link Library Injection (DLL Injection)

Kẻ tấn công có thể tiêm các thư viện động (DLL) vào các tiến trình để né tránh các biện pháp phòng thủ dựa trên tiến trình cũng như có thể nâng cao đặc quyền. Quá trình này thường bao gồm việc ghi đường dẫn của DLL vào không gian địa chỉ ảo của tiến trình mục tiêu và sau đó tải DLL bằng cách kích hoạt một luồng mới trong tiến trình đó.

1. Code Injection (Tiêm Mã)

Code Injection một kỹ thuật được sử dụng để chèn mã độc trực tiếp vào bộ nhớ của một tiến trình đang hoạt động. Mã độc được thực thi trong không gian địa chỉ của tiến trình này và thường được thiết kế để thay đổi luồng điều khiển của tiến trình hoặc thực hiện các hành vi độc hại.

1. Process Hollowing

Process hollowing thường được thực hiện bằng cách tạo một tiến trình ở trạng thái đang treo (một tiến trình giả) sau đó hủy bỏ ánh xạ/ làm rỗng bộ nhớ của nó và thay thế bằng mã độc.

**8. EPO virus là 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.**

*Trả lời:*

**EPO (Entry Point Obscuring) virus** là một loại mã độc có khả năng che giấu entry point của nó trong một chương trình đã bị lây nhiễm. Thay vì điều khiển và thực hiện các hành động ngay khi chương trình được sử dụng hoặc chạy, virus cho phép chương trình hoạt động bình thường một thời gian trước khi thực sự kích hoạt. Kỹ thuật này giúp virus tránh sự phát hiện sớm từ các công cụ phân tích mã độc và các chương trình diệt virus dựa trên mẫu.

*Đặc điểm:*

* Obfuscation (làm rối): EPO virus thay đổi entry point của file thực thi giúp virus che giấu vị trí chính xác của mã độc và cách thức thực thi của nó.
* Stealth (núp): EPO virus được thiết kế để hoạt động ẩn danh và không gây ra nhiều sự chú ý từ phía người dùng hoặc các hệ thống bảo mật. Virus cố gắng tránh bị phát hiện sớm bằng cách che giấu và chỉ thực thi mã độc sau một thời gian hoạt động bình thường của chương trình.
* Sophistication (tinh vi): EPO virus thường sử dụng các kỹ thuật phức tạp và hiện đại để đảm bảo khả năng lây nhiễm rộng rãi và khó bị phát hiện.
* Anti-analysis techniques (các kỹ thuật chống phân tích): EPO virus thường sử dụng các kỹ thuật như self-modification (tự sửa đổi), mã hóa, hay che giấu để gây khó khăn cho các công cụ phân tích mã độc.

*Mục đích:*

* Làm thay đổi luồng thực thi của các chương trình mục tiêu mà không bị phát hiện. EPO virus thường được thiết kế để tạo ra một cánh cửa sau lưng (backdoor) trong hệ thống đã bị nhiễm.
* Một trong các mục tiêu chính của EPO virus là đánh cắp thông tin nhạy cảm từ hệ thống mục tiêu.
* EPO virus cũng có thể được sử dụng để thiết lập các kênh kết nối từ xa, cho phép kẻ tấn công điều khiển và thao tác từ xa trên hệ thống mục tiêu.

*Các loại EPO virus:*

* TLS-EPO virus: Sử dụng kỹ thuật Thread Local Storage (TLS) để che giấu và lây nhiễm mã độc vào các tiến trình. Virus này thường thực hiện việc đặt mã giải mã ban đầu vào một khu vực nhớ mà các tiến trình sử dụng TLS để lưu trữ dữ liệu cục bộ.
* Append-EPO virus: Virus này nối mã độc của mình vào cuối của tệp thực thi và thay đổi điểm nhập của chương trình để thực thi mã virus đầu tiên khi tệp bị chạy.
* Gap-EPO virus: Sử dụng các khoảng trống trong phần mã của tệp thực thi để viết mã giải mã ban đầu. Sau đó, nó sửa đổi điểm nhập của chương trình để chuyển đổi quyền điều khiển sang mã độc.

*Nguyên tắc TLS-EPO virus:*

TLS-EPO virus là một kỹ thuật phức tạp, nhằm thay đổi luồng điều khiển của một tiến trình bằng cách chiếm giữ và thay đổi dữ liệu trong vùng bộ nhớ TLS. Các bước chính:

1. Xác định vùng bộ nhớ TLS
2. Ghi đè dữ liệu TLS
3. Thay đổi luồng điều khiển

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

*Trả lời:*

Các phương pháp tạo mã độc đột biến (mutating malware) nhằm tăng sự khác biệt của các phiên bản mã độc, từ đó làm khó cho các công cụ phòng chống virus nhận diện và loại bỏ chúng.

1. Oligomorphic virus: tạo ra các biến thể của mã độc bằng cách thay đổi đoạn mã hoặc sử dụng các thuật toán mã hóa virus. Tạo ra một số lượng hạn chế các bản giải mã (decryptor) khác nhau (khoảng vài chục bộ) và chọn ngẫu nhiên một bộ trong số đó khi nhân bản. Kích thước lớn
2. Polymorphic virus (virus đa hình): sinh ra các biến thể mã độc hoàn toàn mới mỗi khi lây nhiễm. Các biến thể này có thể khác nhau về cấu trúc, mã hóa và thứ tự các lệnh, thường đi kèm với việc chèn các lệnh rác (junk instructions) để làm phức tạp thêm quá trình phân tích. Đối với polymorphic viruses, toàn bộ cấu trúc mã độc có thể thay đổi, không chỉ giới hạn ở phần giải mã
3. Metamorphic virus (virus biến chất): sử dụng các kỹ thuật biến chất để biến đổi toàn bộ cấu trúc mã độc mỗi khi nhân bản. Metamorphic viruses không chỉ đơn thuần thay đổi cấu trúc mà còn làm thay đổi nội dung và sắp xếp lại toàn bộ mã độc, tạo ra những biến thể hoàn toàn khác nhau về mặt hình thái so với phiên bản trước đó.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Oligomorphic** | **Polymorphic virus** | **Metamorphic virus** |
| Số lượng biến thể | Vài chục | Hàng triệu | Hàng triệu |
| Mục đích chính | Giới hạn số lượng giải mã để làm khó phát hiện dựa trên mẫu cụ thể | Tạo ra các biến thể mã độc để đa dạng hóa và làm phức tạp việc phân tích | Biến chất toàn bộ cấu trúc và nội dung mã độc để tăng sự khó khăn trong phát hiện |
| Phạm vi biến đổi | Giải mã (decryptor) | Toàn bộ cấu trúc mã độc | Toàn bộ cấu trúc và nội dung mã độc |
| Ví dụ | Whale, Memorial | V2PX, 1260 | MSIL/Gastropod (Microsoft Intermediate Language) |
| Tính chất | Các mã độc chung nhóm chia sẻ đoạn mã và thuật toán thay đổi | Các biến thể khác nhau ở cấu trúc, mã hóa, tên biến,… (vẫn dính tới mã độc trước đó) | Không giữ nguyên đoạn mã cụ thể trong mã gốc |

**10. Trình bày 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.**

*Trả lời:*

Tập tin PDF (Portable Document Format) có cấu trúc phức tạp, được thiết kế để bao gồm các đối tượng và dữ liệu phương tiện như văn bản, hình ảnh, đồ họa vector, font chữ, và các phần tử tương tác như liên kết, form, và mã JavaScript. Cấu trúc của một tập tin PDF bao gồm các thành phần chính sau:

1. Header: Đây là phần đầu tiên của tập tin PDF, chứa thông tin cơ bản về phiên bản PDF và các thông số khác.
2. Body: Nội dung chính của tập tin PDF, chứa các đối tượng như trang, văn bản, hình ảnh, và các phương tiện khác.
3. Cross-Reference Table (XRef): Bảng tham chiếu chéo lưu trữ thông tin về vị trí của các đối tượng trong tập tin. Được sử dụng để tìm kiếm nhanh vị trí của các đối tượng.
4. Trailer: Phần cuối cùng của tập tin, cung cấp thông tin bổ sung về tập tin PDF như ký tự số và kết thúc tập tin.
5. Objects: Các đối tượng PDF, bao gồm các loại như văn bản, hình ảnh, font, mã JavaScript, liên kết, form, vv.

Chiến lược chèn đoạn mã độc vào tập tin PDF và khả năng tấn công trên các loại kỹ thuật này:

* Chèn mã JavaScript: PDF hỗ trợ việc nhúng mã JavaScript, có thể được sử dụng để tạo lỗ hổng bảo mật. Mã JavaScript trong PDF có thể gây ra các hành động không mong muốn như mở các đường dẫn độc hại, thực thi mã độc trên hệ thống, gửi thông tin nhạy cảm ra ngoài, vv.
* Sử dụng các đối tượng có lỗi (vulnerable objects): Các đối tượng như hình ảnh, font chữ, hoặc form trong PDF có thể có lỗ hổng bảo mật. Kẻ tấn công có thể tận dụng các lỗ hổng này để thực thi mã độc khi tài liệu PDF được mở.
* Exploit các ứng dụng đọc PDF: Kẻ tấn công có thể tạo các tập tin PDF chứa các lỗ hổng được biết đến của các ứng dụng đọc PDF như Adobe Acrobat Reader, Foxit Reader, hoặc SumatraPDF. Khi tài liệu này được mở bằng ứng dụng này, lỗ hổng sẽ được khai thác để thực thi mã độc.
* Sử dụng kỹ thuật phân tán (obfuscation): Để tránh phát hiện, mã độc trong tập tin PDF có thể được mã hóa hoặc che giấu dưới dạng các đoạn mã phức tạp hoặc sử dụng kỹ thuật steganography để giấu thông tin.

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

*Trả lời:*

**Environmental keying** sử dụng mật mã học tạo ra các khóa mã hóa/giải mã để ràng buộc việc thực thi hoặc các hành động dựa trên các điều kiện cụ thể của môi trường mà kẻ tấn công cung cấp, những điều kiện này dự kiến sẽ có mặt trên mục tiêu.

Mục đích của Environmental Keying:

* Tránh phát hiện: Mã độc có thể kiểm tra các yếu tố môi trường như tên máy tính, địa chỉ IP, phiên bản hệ điều hành, hoặc sự hiện diện của phần mềm bảo mật. Nếu môi trường không khớp với mục tiêu, mã độc có thể không thực thi hoặc thực hiện các hành động vô hại.
* Tăng độ hiệu quả của tấn công bằng cách chỉ hoạt động trong môi trường mục tiêu, mã độc có thể tối ưu hóa các hành động độc hại và giảm thiểu nguy cơ bị phát hiện sớm.
* Tăng sự độc nhất và bảo mật của khóa bằng cách sử dụng thông tin môi trường, đồng thời cũng tăng tính ngẫu nhiên để đảm bảo an ninh.

Kỹ thuật thực hiện Environmental Keying:

1. Thu thập thông tin môi trường: thu thập các giá trị cụ thể từ môi trường mục tiêu như tên miền, địa chỉ IP, phiên bản phần mềm, tệp tin cụ thể, chia sẻ mạng, thiết bị vật lý, thời gian hệ thống, v.v.
2. Xử lý và trích xuất thông tin: Phân tích và trích xuất các giá trị cần thiết từ thông tin môi trường đã thu thập được.
3. Ánh xạ thông tin vào khóa mã: Sử dụng các giá trị đã trích xuất để tạo khóa mã hóa hoặc giải mã.
4. Sử dụng khóa mã: Áp dụng khóa mã đã tạo để mã hóa hoặc giải mã tải trọng độc hại.

**Environmental Sensitivity** cũng liên quan đến việc mã độc phản ứng với môi trường xung quanh, nhưng kỹ thuật này nhấn mạnh việc thay đổi hành vi dựa trên các thay đổi trong môi trường thay vì chỉ kiểm tra các yếu tố tĩnh khi khởi động.

|  |  |  |
| --- | --- | --- |
|  | **Environmental keying** | **Environmental Sensitivity** |
| Mục đích chính | Tránh phát hiện bằng cách kiểm tra các yếu tố môi trường tĩnh | Thích nghi với môi trường bằng cách phản ứng với các thay đổi trong môi trường |
| Cơ chế hoạt động | Kiểm tra các yếu tố như tên máy tính, địa chỉ IP, phần mềm bảo mật khi khởi động | Theo dõi và phản ứng với các thay đổi trong môi trường sau khi khởi động |
| Thời điểm kiểm tra | Thực hiện kiểm tra khi khởi động | Liên tục theo dõi và phản ứng trong suốt quá trình thực thi |
| Điều kiện thực thi | Chỉ thực thi mã độc khi tên máy tính khớp với danh sách mục tiêu | Thay đổi hành vi khi phát hiện thay đổi trong cấu hình mạng hoặc hoạt động của người dùng |

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

*Trả lời:*

1. Anti-debugging

Một kỹ thuật phổ biến chống phân tích, được mã độc sử dụng để nhận biết khi nó đang bị kiểm soát bởi một trình gỡ lỗi hoặc để làm gián đoạn hoạt động của các trình gỡ lỗi.

* Khi mã độc nhận ra rằng nó đang chạy trong một trình gỡ lỗi, nó có thể thay đổi đường dẫn thực thi mã thông thường của nó hoặc sửa đổi mã để gây ra sự cố.
* Điều này khiến các nhà phân tích gặp khó khăn trong việc hiểu mã độc, đồng thời tăng thêm thời gian và công sức cần thiết để phân tích nó.

1. Windows Debugger Detection

Sử dụng Windows API: Windows API cung cấp một số hàm mà một chương trình có thể sử dụng để xác định xem nó có đang bị gỡ lỗi hay không.

* Một số hàm này được thiết kế đặc biệt để phát hiện trình gỡ lỗi.
* Một số hàm khác được thiết kế cho các mục đích khác nhau nhưng có thể được tái sử dụng để phát hiện trình gỡ lỗi.

|  |
| --- |
| IsDebuggerPresent  BOOL WINAPI IsDebuggerPresent(void);   * Hàm này tìm kiếm trường IsDebugged trong cấu trúc Process Environment Block (PEB). * Nếu tiến trình hiện tại đang chạy trong ngữ cảnh của một trình gỡ lỗi, giá trị trả về sẽ khác 0 và ngược lại.   CheckRemoteDebuggerPresent  BOOL WINAPI CheckRemoteDebuggerPresent(  \_In\_ HANDLE hProcess,  \_Inout\_ PBOOL pbDebuggerPresent);   * hProcess [in]: Một handle tới tiến trình. * pbDebuggerPresent [in, out]: Một con trỏ tới một biến mà hàm sẽ thiết lập thành TRUE nếu tiến trình được chỉ định đang bị gỡ lỗi, hoặc FALSE nếu không. * Nếu hàm thành công, giá trị trả về là khác 0, nếu không, giá trị trả về là 0. * Xác định liệu tiến trình được chỉ định (từ xa) có đang bị gỡ lỗi hay không. |

\* Kiểm tra NTGlobalFlag

* PEB\*\* có một trường gọi là NtGlobalFlag (offset 0x68) mà các chương trình có thể kiểm tra để xác định liệu chúng có đang bị gỡ lỗi hay không.
* Thông thường, khi một tiến trình không bị gỡ lỗi, trường NtGlobalFlag chứa giá trị 0x0.
* Khi tiến trình đang bị gỡ lỗi, trường này thường chứa giá trị 0x70.

1. Xác định hành vi của trình gỡ lỗi

Trình gỡ lỗi có thể được sử dụng để thiết lập các điểm dừng (breakpoints) hoặc để từng bước thực hiện qua một tiến trình nhằm hỗ trợ nhà phân tích phần mềm độc hại trong việc dịch ngược mã (reverse-engineering). Nhiều kỹ thuật chống gỡ lỗi được sử dụng bởi phần mềm độc hại để phát hiện loại hành vi của trình gỡ lỗi này:

* INT Scanning (Mã lệnh (opcode) cho INT 3 là 0xCC)
* Kiểm tra checksum
* Kiểm tra thời gian (timing checks) (The most common timing check method uses the rdtsc instruction (opcode 0x0F31))

1. Anti-Virtual Machine (anti-VM) Techniques

Kỹ thuật này dùng để phát hiện xem mã độc có chạy trong môi trưởng ảo hay không bằng cách kiểm tra biến môi trường, giá trị đặc biệt trong bộ nhớ hoặc các đặc trưng của máy ảo.

* Tìm kiếm các tiến trình như VMwareService.exe, VMwareTray.exe, và VMwareUser.exe.

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

*Trả lời:*

1. Code Obfuscation (Rối mã): Đây là kỹ thuật chính để làm cho mã nguồn hoặc mã thực thi trở nên khó hiểu và khó phân tích bởi con người. Code Obfuscation có thể bao gồm:

- Thay đổi tên biến, tên hàm thành các tên không mô tả.

- Chèn các đoạn mã rác (junk code) như các lệnh không cần thiết hoặc không có tác dụng.

- Thay đổi cấu trúc luồng điều khiển, bao gồm việc chèn điều kiện và vòng lặp phức tạp để làm khó việc theo dõi chương trình.

2. Dynamic Code Generation (Tạo mã động): Thay vì sử dụng mã cố định, chương trình mã độc có thể tạo ra mã thực thi tại runtime. Điều này làm cho việc phân tích tĩnh trở nên không hiệu quả vì mã thực thi chỉ xuất hiện khi chương trình thực thi và có thể thay đổi từ lần chạy này sang lần chạy khác.

3. Encryption and Packing (Mã hóa và đóng gói): Sử dụng mã hóa để bảo vệ phần quan trọng của chương trình hoặc các dữ liệu độc hại. Mã hóa có thể áp dụng cho một phần hay toàn bộ chương trình.

4. Self-modifying Code (Mã tự sửa đổi): Sử dụng mã thay đổi chính nó trong quá trình thực thi để làm khó việc phân tích tĩnh. Việc này gây khó khăn cho các công cụ phân tích vì mã thực thi thay đổi theo thời gian và không giống như mã tĩnh trong quá trình phân tích.

**14. Phân biệt thuật ngữ Ransomware as a Service (RaaS) so với Ransomware? Chúng có những loại chính nào?**

*Trả lời:*

**Ransomware** là một loại phần mềm độc hại (malware) được thiết kế để xâm nhập vào máy tính hoặc hệ thống, mã hóa các tệp dữ liệu quan trọng và yêu cầu người dùng trả tiền chuộc để nhận được khóa giải mã.

**Ransomware as a service (RaaS)** là một mô hình kinh doanh trong ngành tội phạm mạng, trong đó các kẻ tấn công không phải là tác giả chính của mã độc, mà họ có thể thuê hoặc mua các dịch vụ ransomware từ các nhà cung cấp RaaS.

RaaS có thể được coi là một trong những nguyên nhân chính dẫn đến sự bùng nổ nhanh chóng của các cuộc tấn công ransomware, vì nó làm cho việc triển khai ransomware trở nên dễ dàng hơn đối với nhiều nhà đe dọa — kể cả những người có ít kiến thức kỹ thuật — để triển khai ransomware vào các mục tiêu.

RaaS dựa trên mô hình phần mềm dưới dạng dịch vụ (SaaS), trong đó phần mềm có thể được truy cập trực tuyến theo hình thức đăng ký dựa trên thuê bao.

VD: BlackCat, LockBit 3.0, CLoP, Black Basta, Royal, Akira, BianLian

Các loại RaaS chính:

* Fully-featured RaaS: Các dịch vụ RaaS đầy đủ tính năng cung cấp cho khách hàng một gói hoàn chỉnh bao gồm cả phần mềm mã hóa, hỗ trợ kỹ thuật và cả hệ thống quản lý nạn nhân.
* Affiliate RaaS: Mô hình này cho phép người sử dụng (các đối tác liên kết) lựa chọn từ một loạt các dịch vụ và chỉ trả tiền khi có tiền chuộc từ nạn nhân.
* RaaS trên dark web: Các dịch vụ RaaS có thể được tìm thấy và thuê trên các sàn thương mại ngầm trên dark web, nơi giao dịch thường được thực hiện bằng tiền điện tử.