|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Virus | Có tác động của user mới lây | Sửa đổi ,xóa , tắt , làm chậm mt |  |  |
| Worm | Tự nhân bản , lây lan (máy này -> khác) | khai thác các lỗ hổng trong hệ thống | tự động sao chép và lây lan từ máy tính này sang máy tính khác qua mạng |  |
| Botnet | Mạng các mt bị xâm nhập , bị điều khiển | thực hiện các hoạt động độc hại mà không được sự cho phép của chủ sở hữu máy tính. |  |  |
| Ransomware | Mã hóa dự liệu máy tính nạn nhân (trả tiền thì hết :v) | yêu cầu một khoản tiền chuộc để giải mã. Nếu không trả tiền, dữ liệu sẽ bị mất hoặc không thể truy cập. |  |  |
| Easter egg | Đc nhúng vào ct khác (game , crack,…) | được thiết kế để gây thú vị hoặc sự ngạc nhiên cho người dùng |  |  |
| Salami | Tấn công ăn cắp từ nhiều nguồn nhỏ. | lấy đi một lượng nhỏ từ nhiều tài khoản hoặc giao dịch | kết hợp chúng để lợi ích cá nhân, trong khi tổng thể không bị nhận thấy. | vd : làm tròn nhiều số tiền , lấy số tiền từ khoảng đc làm tròn lên so vs số tiền ban đầu => càng lấy càng nhiều |
| Back door | truy cập hệ thống máy tính một cách bí mật và trái phép | tạo ra một cổng vào hệ thống mà không được biết đến bởi người dùng chính hoặc các cơ chế bảo mật thông thường |  |  |
| Rootkit | Khai thac lo hong , chiem quyen root tren he thong => lam dieu xau tac dong len ht | cung cấp quyền truy cập cao nhất (root) cho kẻ tấn công và che giấu các hoạt động độc hại. |  |  |
| Logic bomb | Một phần mềm độc hại đc set điều kiện sẵn để kích hoạt | kích hoạt một cách tự động vào một thời điểm cụ thể trong tương lai. | Nó có thể gây ra sự phá hoại hoặc gián đoạn chức năng của hệ thống. |  |
| Time bomb | Một p mềm độc hại đc set trc tgian kich hoat |  |  |  |
| Trojan | P mem độc hại đc inject vào ct benign | được ẩn trong một phần mềm hoặc tập tin hợp pháp. | Khi người dùng chạy Trojan, nó sẽ thực hiện các hoạt động độc hại mà người dùng không biết |  |

**2. Trình bày sự khác biệt giữa Packer,Decryptor và Protector trong ngữ cảnh malware**

|  |  |  |  |
| --- | --- | --- | --- |
| PACKER | công cụ được sử dụng để nén và mã hóa mã nguồn của phần mềm độc hại. | làm cho phần mềm độc hại trở nên khó nhận biết và phân tích bởi các công cụ phân tích mã độc hại và chống vi-rút. |  |
| DECRYPTOR | giải mã và khôi phục lại mã nguồn ban đầu của phần mềm độc hại đã được nén và mã hóa bằng packer. |  |  |
| PROTECTOR | để bảo vệ phần mềm độc hại khỏi việc phân tích, phát hiện hoặc gỡ bỏ bởi các công cụ an ninh. | sử dụng các kỹ thuật như mã hóa, bao gói, phân tán mã và chống phân tích để làm cho phần mềm độc hại trở nên khó khăn hơn để phát hiện và phân tích. | cơ chế : anti-debugging, anti-virtualization, hoặc anti-sandboxing |

**3. Kỹ thuật tiêm tiến trình là gì ? Process Hollowing là gì ? Nêu cách thực hiện ?**

|  |  |  |
| --- | --- | --- |
| Tiêm tiến trình | tiêm và thực thi mã độc vào một tiến trình khác trong hệ thống máy tính mục tiêu | ẩn và che giấu hoạt động độc hại trong một tiến trình hợp pháp, từ đó tránh được sự phát hiện và phân tích từ phía các công cụ phát hiện mã độc |
| Process Hollowing | tạo ra một tiến trình mới hợp pháp, sau đó ghi đè nội dung của tiến trình mới này bằng mã độc | khiến cho tiến trình mới trông giống hệt tiến trình ban đầu nhưng chứa mã độc bên trong. |

Các bước thực hiện :

1. Chọn một tiến trình hợp pháp đã tồn tại trong hệ thống làm tiến trình mục tiêu để thực hiện tiêm tiến trình.
2. Tạo một tiến trình mới sử dụng các API hệ thống như **CreateProcess** hoặc **NtCreateProcess**..
3. Lấy nội dung bộ nhớ của tiến trình mới đã tạo.
4. Ghi đè nội dung của tiến trình mới với mã độc từ tiến trình mục tiêu bằng cách sử dụng các API như **WriteProcessMemory** hoặc **NtWriteVirtualMemory**.
5. Đặt các thông số và context của tiến trình mới theo yêu cầu.
6. Thực thi tiến trình mới sử dụng các API như **ResumeThread** hoặc **NtResumeThread**

* Thực hiện các thứ mà attacker muốn , có thể gây hại đến hệ thống

**4. Process DopppleGangging ? cách thực hiện ?**

|  |  |  |
| --- | --- | --- |
| Process DopppleGangging | Là 1 kỹ thuật chèn mã độc vào tiến trình , với mục đích thực thi mã độc và không làm cho tiến trình đó bị phát hiện bởi các phần mềm như anti-virus |  |

Các bước thực hiện :

1. Tạo Unfinished transaction với 1 file thực thi benign sử dụng TxF với các API như **CreateFileTransacted**
2. Chèn mã độc vào file đó sử API như **WriteFileTransacted**
3. Lưu lại và lấy địa chỉ vùng nhớ của file sau khi bị thay đổi (file mã độc) **NtCreateSection**
4. Hoàn tác lại các tác vụ làm với file bengin bằng **RollBackTransaction**
5. Tạo 1 tiến trình mới bằng API **CreateProcessMemory** và map vùng nhớ chứa mã độc vào tiến trình này
6. Tiến trình chạy thì mã độc được thực thi

**5.Sự khác biệt giữa dropper và downloader trong ngữ cảnh hoạt động của malware. Nhận xét tác động gây hại và phương pháp phòng chống với vấn đề bảo mật an toàn thông tin**

|  |  |  |  |
| --- | --- | --- | --- |
| DROPPER | được sử dụng để triển khai và giải mã các thành phần khác của malware lên hệ thống mục tiêu. | giúp cho malware tiếp cận và thâm nhập vào hệ thống mục tiêu một cách an toàn và ẩn danh. | tự giải mã, giải nén hoặc tạo ra các tệp tin hoặc thư mục mới chứa các thành phần chính của malware. |
| DOWNLOADER | để tải về và cài đặt các phần mềm độc hại | giúp malware cung cấp các tính năng bổ sung, như botnet, keylogger, ransomware |  |
| Tác động gây hại | góp phần trong hoạt động của malware | gây ra các tác động gây hại trên hệ thống và thông tin cá nhân của người dùng. | đánh cắp thông tin cá nhân, mã hóa dữ liệu và yêu cầu tiền chuộc, hoặc thực hiện các hành động phá hoại khác. |
| Phòng chống | việc duy trì các tường lửa, phần mềm diệt virus và cập nhật hệ thống | kiên nhẫn và cảnh giác khi mở các tệp tin đính kèm không rõ nguồn gốc hoặc truy cập vào các trang web đáng ngờ | Đảm bảo rằng phần mềm bảo mật và hệ điều hành luôn được cập nhật với các bản vá |

**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 word. Trình bày cách tin tặc thực hiện tấn công mã độc thông qua các dạng tài liệu Microsoft word**

|  |  |  |  |
| --- | --- | --- | --- |
| Macro malware | chèn mã độc vào các macro (đoạn mã được sử dụng để thực hiện các tác vụ tự động) trong tài liệu Word. | Khi tài liệu được mở, macro sẽ chạy tự động và thực thi mã độc, gây ra các hành động độc hại như cài đặt phần mềm độc hại hoặc lây lan qua hệ thống. | Tắt các tính năng auto chạy tác vụ của word ,… |
| Exploit: | sử dụng các lỗ hổng trong phần mềm Microsoft Word để chèn mã độc. |  | Cập nhật bản vá |
| Social Engineering: | gửi tài liệu Word độc hại qua email hoặc trang web giả mạo, kèm theo lời mời người nhận mở tài liệu. |  | Xem rõ nguồn gốc người gửi , xem xét file nhận đc có gì đáng nghi không |
| Embedded Objects: | nhúng các đối tượng như hình ảnh, video, âm thanh hoặc tệp tin khác vào tài liệu Word. | Những đối tượng nhúng này có thể chứa mã độc và khi tài liệu được mở, mã độc sẽ được thực thi. | Sử dụng tool quét |
|  |  |  |  |

1. **Thuật ngữ Process Injection dùng cho mục đích gì ? Nêu tên và giải thích nguyên tắc thực hiện của 3 kỹ thuật phổ biến của Process Injection ?**

|  |  |  |
| --- | --- | --- |
| Process inject | Tiêm mã độc vào tiến trình làm cho tiến trình thực thi mã độc mà không bị phát hiện bởi các phần mềm phát hiện mã độc như anti-virus |  |
| DLL injection | Inject các thư viện độc hại vào process làm cho process đó thực thi sai và gây hại đến ht |  |
| Process Hollowing | Tạo bản sao của process benign và chèn mã độc vào bản sao đó và làm sao process đó thực thi | CreateProcess,WriteProcessMemory,  ThreadResume |
| Process Dopple ganging | Tạo 1 unfinished transaction với file thực thi benign , chèn mã độc vào process của nó , tạo section (vùng nhớ) chứa mã độc , hoàn tác lại những gì vừa thực hiện, tạo process mới có vùng nhớ là section mã độc đó , làm cho process đó chạy | CreateFileTransacted  WriteFileTransacted  NtCreateSection  RollBackTransaction  NtCreateThreadEX |
| Code Injection | Chèn mã độc vào ct : .exe,.. |  |

**8. EPO virus là gì ? đặc điểm , mục đích . Có những loại nào ?Trình bày nguyên tắc của trường hợp TLS-EPO virus**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EPO / TLS-EPO virus | Encapsulated PostScript Object  Transport Layer Security | virus này khai thác một lỗ hổng bảo mật trong giao thức TLS để tấn công và kiểm soát hệ thống mục tiêu. |  |  |
| Đặc điểm | mã độc được gửi qua kết nối bảo mật TLS (sử dụng giao thức HTTPS) để tránh bị phát hiện. | Tận dụng các lỗ hổng trong giao thức TLS để thực hiện tấn công xâm nhập | đóng gói dưới dạng Encapsulated PostScript Object (EPS) để truyền tải và thực thi mã độc. | được sử dụng trong các cuộc tấn công mạng có mục tiêu cao cấp và nhắm vào hệ thống quan trọng. |
| Mục tiêu | Kiểm soát từ xa hệ thống mục tiêu | Đánh cắp thông tin nhạy cảm | Lây nhiễm và lây lan |  |
| Nguyên tắc | tận dụng một lỗ hổng bảo mật trong giao thức TLS để thực hiện tấn công | lỗ hổng này có thể là một lỗ hổng phần mềm hoặc các vấn đề liên quan đến quản lý chứng chỉ TLS | tạo ra một chứng chỉ TLS giả mạo hoặc sử dụng một chứng chỉ hết hạn hoặc thuộc sở hữu của người khác | có tính phức tạp và đòi hỏi kiến thức chuyên sâu về bảo mật để thực hiện và phòng chống. |

9.Trình bày mục đích của pp tạo mã độc đột biến dị ,đa ,siêu hình

**10.Trình bày cấu trúc tập tin PDF , các chiến lược chèn mã độc vào tập Tin PDF và khả năng tấn công của các loại kĩ thuật này**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PDF |  |  |  |  |
| Cấu trúc | Header: tt định dạng tập tin ,…  Body: nd tt , văn bản ,h ảnh ,… | Cross-Reference Table : các đối tượng và vi tri của chúng trong Body  Trailer : tt kết thúc tập tin và danh sách các đối tượng |  |  |
| chiến lược chèn mã độc vào tập tin PDF | Exploiting Vulnerabilities : sử dụng các lỗ hổng trong phần mềm đọc PDF để chèn mã độc vào tập tin PDF | Obfuscation :  Che giấu để ẩn mã độc trong tập tin PDF  nén hoặc mã hóa | Social Engineering : tạo ra các tập tin PDF với nội dung mạo danh hoặc cám dỗ người dùng | Embedded object : nhúng mã độc vào các hình ảnh , video trong PDF |
| Khả năng tấn công | Dễ thực hiện hơn cách sử dụng OPE-TLS virus , không đòi hỏi yêu cầu kiến thức quá chuyên sâu |  |  |  |

**11.Enviromental Keying là gì ? Mục đích và các kỹ thuật thực hiện trong các chương trình phần mềm chứa mã độc hại ?**

Trong lĩnh vực malware thì enviromental keying là kỹ thuật trong việc tạo ra khóa (key) hoặc thông tin xác thực (authentication information) dựa trên các đặc điểm hoặc môi trường của hệ thống mục tiêu.

Mục đích của Environmental Keying là tạo ra một khóa duy nhất hoặc thông tin xác thực dựa trên những thông tin đặc thù của môi trường để sử dụng trong các chương trình phần mềm chứa mã độc hại.

Mục đích :

Ví dụ : malware A bị detect và ngăn chặn bởi tool B ,C thì điều kiện thực thi của malware A là không có Tool B,C trên hệ thống thì nó sẽ thực thi và ngoặc lại thì nó sẽ không thực thi . Bởi khi dựa vào điều kiện này thì nó sẽ thực thi trong ngữ cảnh không bị phát hiện và gây hại âm thầm đến hệ thống

Ngoài ra còn nhiều mục đích khác nữa ,…

1. Kiểm soát từ xa (Remote Control):
2. Đánh cắp thông tin (Information Theft)
3. Phá hoại hệ thống (System Disruption)
4. Lây nhiễm và lan truyền (Infection and Propagation)

Kỹ thuật thực hiện trong các phần mềm chứa code độc hại :

1. Code injection

* Chèn mã độc vào ct benign thg là .exe,…
* Thực thi mã độc trước khi thực hiện mục đích của ct chính

1. Obfucasion

* Che dấu mã độc bằng nén,mã hóa , giải mã phần virus body đc mã hóa khi chúng cần thực thi
* Che dấu ,trốn tránh các phần mềm như anti-virus

1. Exploit
2. Social Engineering
3. Backdoor
4. Trojan
5. Phishing

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

1. (Environment keying)

sử dụng các phương pháp để kiểm tra và phát hiện môi trường gỡ rối hoặc các tiến trình phân tích đang chạy, và sau đó thực hiện các hành động để ngăn chặn việc phân tích.

2.Enviromental sensitivity

**13.Chống phân tích tĩnh malware thường sử dụng chiến lược nào trong mã nguồn của nó**

1. Code Obfuscation (Mã hoá)

2. Anti-Debugging Techniques

Nếu phát hiện các môi trường gỡ rối hoặc công cụ phân tích, malware có thể thay đổi hành vi của nó hoặc dừng thực thi để tránh bị phát hiện và phân tích.

4. Dynamic Code Generation

tạo ra mã động trong quá trình thực thi thay vì sử dụng mã nguồn tĩnh.

1. Anti-Disassembly Techniques

trộn lẫn mã, sử dụng các hàm không xác định và các cấu trúc phức tạp để gây khó khăn trong quá trình phân tích tĩnh và disassembly mã