BÀI 09. PHẦN MỀM MÃ ĐỘC

Bùi Trọng Tùng,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

Nội dung

- Giới thiệu về phần mềm mã độc
- Virus
- Trojan
- Worm
- Phát hiện và giảm thiểu nguy cơ tấn công bằng phần mềm độc hại

1. GIỚI THIỆU CHUNG

Khái niệm

- Phần mềm độc hại(malicious software hoặc malware) là những chương trình máy tính mà khi thực thi sẽ gây tổn hại tới tài nguyên của hệ thống hoặc chiếm đoạt một phần/toàn bộ quyền điều khiển hệ thống
- Phân loại:
  - Virus: tự lây nhiễm vào các file
  - Worm: tự lây nhiễm vào các chương trình thực thi
  - Trojan: chương trình ẩn giấu trong các tệp tin có vẻ vô hại, không có khả năng tự lây nhiễm
  - Sự phân biệt các loại này là không rõ ràng. Trong bài giảng sử dụng thuật ngữ quen thuộc là “virus”

Các hành vi gây hại

- Phá hủy dữ liệu, phần cứng
- Nghe trộm hoạt động của người dùng trên các thiết bị vào ra(Keylogging)
- Đánh cắp thông tin (spyware)
- Mã hóa dữ liệu (ransomware)
- Đánh cắp tài nguyên tính toán (coinminer)
- Tạo cửa hậu (backdoor) để kẻ tấn công xâm nhập và điều khiển
- Che giấu hoạt động(rootkit)
- Thực hiện các hành vi tấn công(botnet)

Các hành vi này có thể được thực hiện ngay hoặc đợi điều kiện nào đó (time bomb, logic bomb)

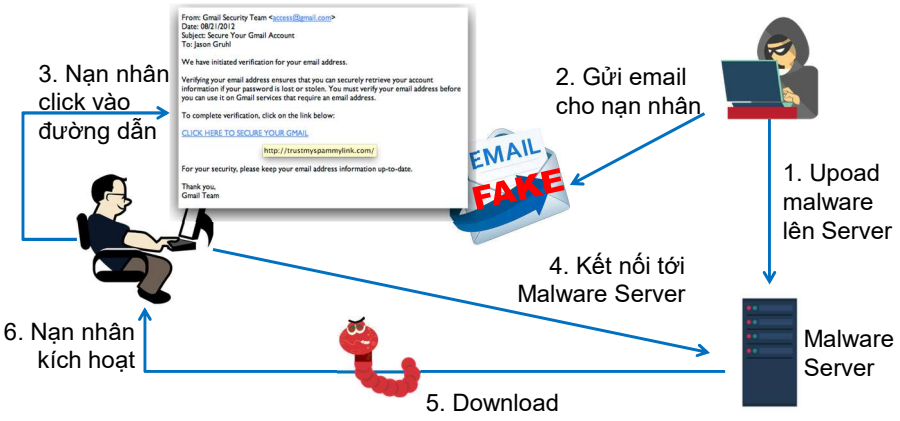
5

Các con đường lây nhiễm

- Email
- Ứng dụng truyền thông điệp (Instant messaging)
- Các thiết bị lưu trữ di động
- Chương trình giả mạo
- Tiện ích chia sẻ file trong mạng LAN
- Phần mềm bẻ khóa bản quyền
- Chương trình chia sẻ file
- Lỗ hổng phần mềm
- …

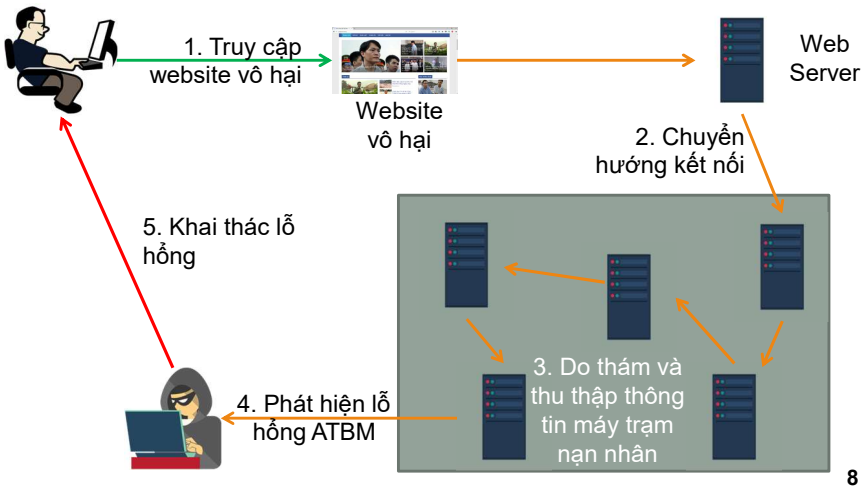
6

Một kịch bản phát tán và lây nhiễm



Kịch bản khác

- Sử dụng các công cụ khai thác lỗ hổng

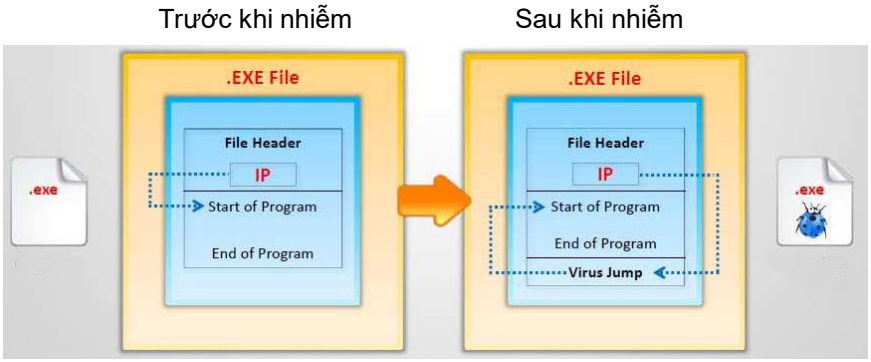


Cách thức hoạt động của virus

- Virus thông thường có 3 đoạn mã:
  - Đoạn mã lây nhiễm: cho phép virus tự sao chép bản thân nó và lây nhiễm từ chương trình này sang chương trình khác
  - Đoạn mã kích hoạt: Là các sự kiện hoặc điều kiện xác định khi nào *hoạt động chính* sẽ được kích hoạt
  - Đoạn mã hoạt động: phần thực hiện các hành động phá hoại của virus
- Virus được mô tả với 2 đặc trưng:
  - Cách thức lây nhiễm
  - Các hành vi phá hoại

9

Cơ chế tiêm nhiễm



- Nguyên tắc cơ bản: Virus thay thế lệnh đầu tiên của file bị nhiễm (.exe) bằng một lệnh JUMP tới đoạn mã thực thi của virus. Kết thúc đoạn mã thực thi của virus là lệnh JUMP khác để nhảy tới lệnh đầu tiên của chương trình ban đầu

10

2. CÁC PHƯƠNG PHÁP PHÁT HIỆN

Cuộc đua giữa phát hiện và lẩn tránh

11

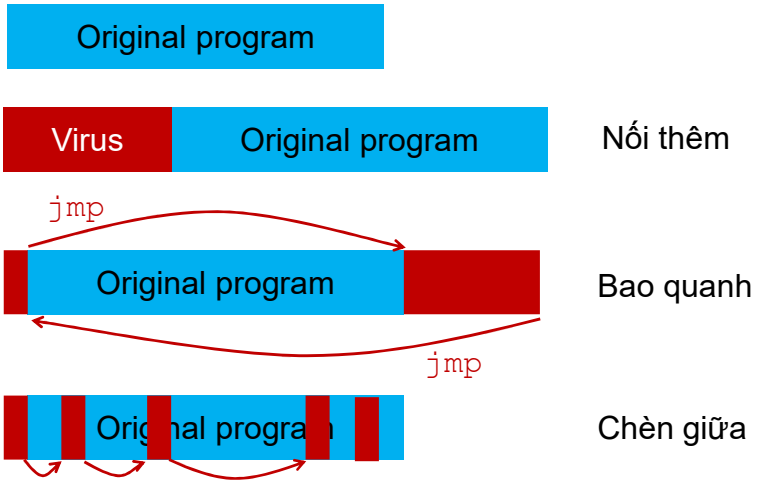
Phát hiện virus

- Phương pháp phổ biến: Phát hiện dựa trên đặc trưng
  - Thu thập các mẫu virus và xây dựng CSDL đặc trưng về các virus. Thông thường là các đoạn mã lây nhiễm ở đầu file
  - Phát hiện: So sánh các byte trên file với những mẫu virus đã có
- Nếu là tin tặc, bạn sẽ làm gì?

12

Cách thức lẩn tránh

- Làm cho đặc trưng trở nên khó tìm kiếm hơn

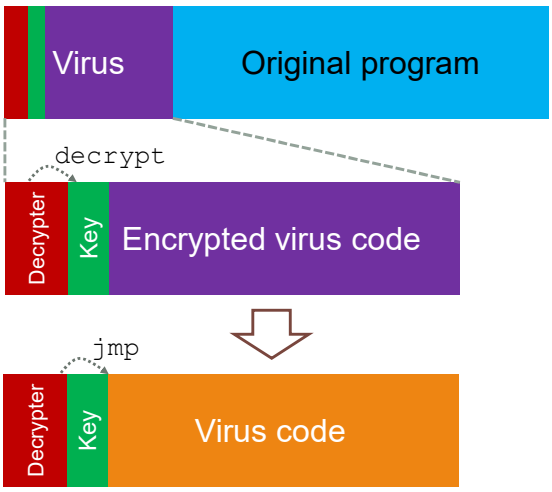


13

13

Polymorphic virus – Virus đa hình

- Thay đổi mã nguồn một cách ngẫu nhiên

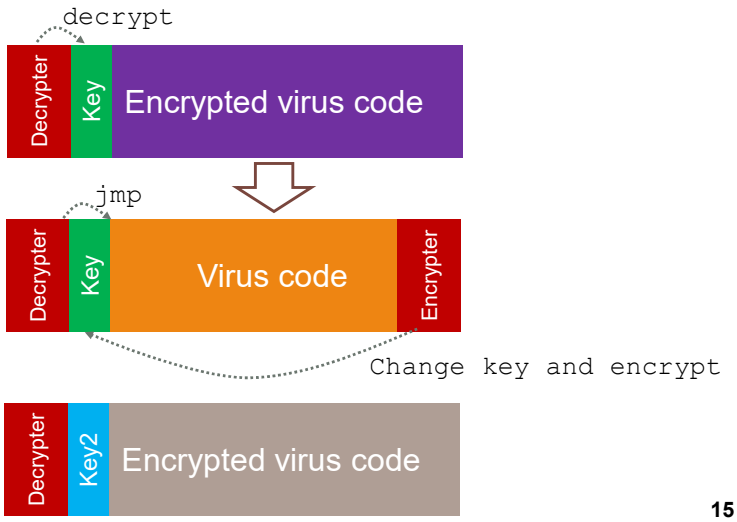


14

14

Polymorphic virus – Lây nhiễm

- Thay đổi khóa và mã hóa lại mã nguồn



15

Polymophic virus – Phát hiện

- Ý tưởng 1: Sử dụng đặc trưng "hẹp" để phát hiện trình giải mã decrypter
  - Số byte mã nguồn cần so sánh ít hơn → dễ phát hiện nhầm
  - Tin tặc có thể nhanh chóng thay đổi trình giải mã
- Ý tưởng 2: Thực thi để phát hiện sự có mặt của đặc trưng trên mã nguồn đã giải mã
  - Vấn đề: Thực thi đến thời điểm nào thì so sánh đặc trưng?
- Làm thế nào để lẩn tránh chương trình phát hiện virus?

16

Metamorphic Virus

T-1000 in Terminator 2

17

Metamorphic Virus

- Virus siêu đa hình: sử dụng đoạn mã đặc biệt (metamorphic code) để tự thay đổi mã nguồn về mặt ngữ nghĩa khi thực thi
  - Không thay đổi ngữ nghĩa ở mức cao hơn (vẫn giữ nguyên các chức năng, tính năng)
- Một số kỹ thuật thực hiện:
  - Tạo ra các đoạn mã dư thừa ngẫu nhiên
  - Thay đổi các thanh ghi
  - Thay đổi trình tự trong biểu thức điều kiện
  - Thay đổi trình tự các câu lệnh xử lý không có ràng buộc với nhau
  - Thay thế các thuật toán

18

Win95/Regswap(1998)

```
5A              pop  edx
BF04000000      mov  edi,0004h
8BF5            mov  esi,ebp
B80C000000      mov  eax,000Ch
81C288000000    add  edx,0088h
8B1A            mov  ebx,[edx]
899C8618110000  mov  [esi+eax*4+00001118],ebx

58              pop  eax
BB04000000      mov  ebx,0004h
8BD5            mov  edx,ebp
BF0C000000      mov  edi,000Ch
81C088000000    add  eax,0088h
8B30            mov  esi,[eax]
89B4BA18110000  mov  [edx+edi*4+00001118],esi
```

19

19

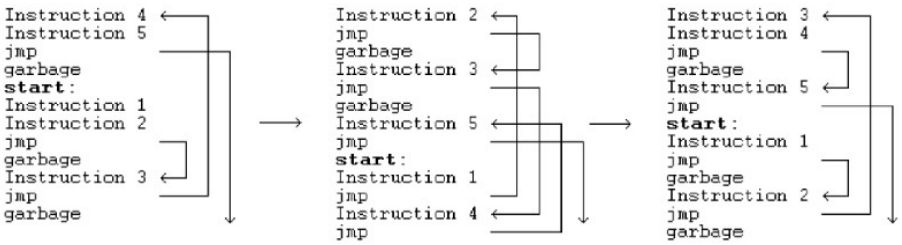
Win32/Evol(2000)

```
a. An early generation:

C7060F000055  mov      dword ptr [esi],5500000Fh
C746048BEC5151 mov     dword ptr [esi+0004],5151EC8Bh

b. And one of its later generations:

BF0F000055    mov      edi,5500000Fh
893E          mov      [esi],edi
5F            pop      edi
52            push     edx
B640          mov      dh,40
BA8BEC5151    mov      edx,5151EC8Bh
53            push     ebx
8BDA          mov      ebx,edx
895E04        mov      [esi+0004],ebx

c. And yet another generation with recalculated ("encrypted") "con-
stant" data.

BB0F000055    mov      ebx,5500000Fh
891E          mov      [esi],ebx
5B            pop      ebx
51            push     ecx
B9CB00C05F    mov      ecx,5FC000CBh
81C1C0EB91F1  add      ecx,F191EBC0h ; ecx=5151EC8Bh
894E04        mov      [esi+0004],ecx
```

20

20

Zperm.A(2000)



ZPerm can directly reorder the instructions in its own code

21

Phát hiện virus siêu đa hình

- Phát hiện dựa trên hành vi (Behavior-based detection)

Phân tích động

- Thực thi mã độc trên môi trường Sandbox và quan sát hoạt động của mã độc
- Ưu điểm: thời gian phân tích nhanh, có thể xác định ngay cách thức hoạt động của virus
- Nhược điểm: yêu cầu môi trường an toàn để phân tích, không xác định được hết tất cả các hành vi

Phân tích tĩnh

- Sử dụng kỹ thuật dịch ngược để phân tích mã thực thi
- Ưu điểm: không cần kích hoạt mã độc, xác định được tất cả các cơ chế hoạt động, hành vi của mã độc
- Hạn chế: phức tạp, đòi hỏi trình độ nhân lực cao hơn, mất nhiều thời gian

22

Quy trình phân tích

Tạo môi trường Sandbox để phân tích

- Bước 1: Tạo các máy ảo(Virtualbox, Hyper-V, …) và các môi trường ảo hóa khác nếu cần(mạng, CSDL…)
- Bước 2: Cài đặt hệ điều hành trên máy ảo
- Bước 3: Tắt hoặc hạn chế hoạt động của cạc mạng trên máy ảo để cách ly với môi trường thực
- Bước 4: Tắt các chức năng chia sẻ file, thư mục
- Bước 5: Chuyển mã độc vào môi trường phân tích

Môi trường phân tích phải cách ly hoàn toàn với môi trường làm việc và được giám sát đầy đủ

Quy trình phân tích

Phân tích tĩnh

- Bước 1: Dịch ngược mã nguồn
- Bước 2: Thu thập thông tin:
  - Giá trị các xâu ký tự: sử dụng công cụ BinText
  - Các kỹ thuật đóng gói, nén, mã hóa của virus và thực hiện các thao tác giải nén, giải mã cần thiết: sử dụng công cụ UPX

Phân tích động

- Bước 3: Thiết lập kết nối mạng(vật lý) cho môi trường phân tích. Lưu ý, giám sát chặt chẽ và không kết nối với mạng tác nghiệp của tổ chức
- Bước 4: Kích hoạt virus và thu thập thông tin tiến trình thực thi của virus, thông tin hệ thống khi virus hoạt động. Sử dụng các công cụ Process Monitor và Process Explorer

Quy trình phân tích

- Bước 5: Ghi nhận các kết nối mạng(logic) mà virus tạo ra. Bắt và phân tích lưu lượng phát sinh trên các kết nối này. Các công cụ có thể sử dụng: Wireshark, tcpdump, NetResistent, TCPView
- Bước 6: Xác định các tệp tin mới, tiến trình mới được tạo ra, sự thay đổi các giá trị registry trên hệ thống (sử dụng RegShot)
- Bước 7: Phân tích mã thực thi trên RAM, sử dụng công cụ OllyDbg, ProcDump

25

Lẩn tránh

- Chống phân tích tĩnh: Tạo ra các đoạn mã phức tạp để che giấu hoạt động thực sự
- Chống phân tích động:
  - Phát hiện môi trường thực thi để thay đổi hành vi
  - Tạo ra các hành động khiến quá trình thực thi kéo dài
- Ứng phó của phần mềm anti-virus:
  - Tìm kiếm và bỏ qua các đoạn mã/hành vi vô nghĩa
  - Mô hình hóa các hành vi chung
- Tiếp tục…

→ Cuộc đua giữa tin tặc và phần mềm AV mà tin tặc thường bước đi trước(Tại sao?)

26

Rootkit/Stealth Virus

- Có khả năng ẩn mình trước các phần mềm phát hiện virus.
- Cơ chế chung: sử dụng kỹ thuật hook để chặn các sự kiện và can thiệp vào quá trình xử lý sự kiện
- User-level rootkit: hook vào hàm thư viện
  - Dễ bị phát hiện
- Kernel-level rootkit: hook vào các hàm thực thi lời gọi hệ thống, hàm xử lý ngắt, driver điều khiển thiết bị, firmware của thiết bị
  - Khó bị phát hiện
- Virtualization-based rootkit: ẩn mình trong môi trường ảo hóa → gần như không thể bị phát hiện

27

Phát hiện và phòng chống rootkit

- Phát hiện dựa trên hành vi
  - Phát hiện các hành vi hook
  - Sự biến đổi của số lượng, tần suất và thứ tự thực hiện các lời gọi hệ thống
- Kiểm tra toàn vẹn tập tin hệ thống
- Phát hiện dựa trên sự sai khác với hệ thống tham chiếu

28

Phòng chống và giảm thiểu

- Tránh mở các file đính kèm từ các email không rõ nguồn gốc
- Sử dụng firewall chặn tất cả các cổng dịch vụ không cần thiết
- Tránh nhận các file từ ứng dụng tin nhắn
- Gia cố hệ thống, tắt các chức năng không cần thiết trên máy tính
- Kiểm soát lưu lượng nội bộ
- Không tải và thực thi các file ứng dụng từ nguồn lạ
- Cập nhật các bản vá bảo mật

29

Phòng chống và giảm thiểu

- Quét, rà soát virus trên các thiết bị nhớ lưu động(USB drive, CD/DVD, thẻ nhớ, thiết bị di động,…) khi kết nối với máy tính
- Phân quyền người dùng
- Sử dụng phần mềm bản quyền. Không dùng các công cụ bẻ khóa, cung cấp mã bản quyền
- Cài đặt phần mềm diệt virus
- Xây dựng chính sách và đào tạo người dùng

30