Tạp chí Tin học và Điều khiển học, T. 13, S. 3 (1997) (123–132)

CÁC GIẢI PHÁP CHO PHẦN MỀM CHỐNG VIRUS THÔNG MINH

NGUYỄN QUANG THỦY[1], TRƯƠNG MINH NHẬT QUANG[2]

Abstract. In this paper we shall propose some solutions for an in telligent anti-virus system which are investigated also in the system D2. Principles of such sulotions will be detailed. The implementation is beling carried out, showing promising results.

I. ĐẶT VẤN ĐỀ

Trong thời đại ngày nay, làn sóng công nghệ thông tin nhanh chóng lan tỏa khắp mọi lĩnh vực, đời sống xã hội. Càng ngày nhu cầu giao tiếp thông tin giữa người dùng, giữa các trạm làm việc trong mạng... trở nên phổ biến. Nhờ các mạng máy tính, đặc biệt là mạng diện rộng Internet, người dùng có thể truy nhập thông tin, trao đổi chương trình, liên lạc... trên phạm vi toàn thế giới. Trong bối cảnh đó, sự xâm nhập của virus tin học là điều không thể tránh khỏi. Chúng đã gây ra những vụ mất mát, sai lạc dữ liệu, thậm chí làm sụp đổ cả hệ thống công nghệ thông tin... trên qui mô rộng. Vì thế việc bảo vệ dữ liệu cho các hệ thống tin học trở nên cấp bách hơn bao giờ hết.

Giống như quá trình chữa bệnh cho con người, việc chẩn đoán và trị liệu các tác nhân nhiễm bệnh trên máy tính được thực hiện theo qui trình định sẵn. Có thể tóm tắt qui trình đó như sau:

(1) Phát hiện tác nhân lạ (virus) nhiễm vào hệ thống.

(2) Phân tích cơ chế hoạt động của virus.

(3) Tạo "thuốc" chống lại sự lây nhiễm, phục hồi thông tin nếu có thể.

Hầu như tất cả các chương trình chống virus hiện nay đều áp dụng qui trình này trong việc cập nhật và phát triển phần mềm. Các công đoạn nói trên được thực hiện bởi các chuyên gia về virus tin học. Ngoài công đoạn (3) được thực hiện hoàn toàn bằng tin học, hai công đoạn (1) và (2) - quan trọng nhất, lại thường được thao tác "thủ công" hoặc chỉ sử dụng tin học một phần. Điều này khiến phần mềm chống virus luôn bị động vì phải luôn đối phó với sự ra đời của các virus mới, các hệ thống công nghệ thông tin phải đặt trong tình trạng cảnh giác, thường xuyên cập nhật "thuốc" dò tìm và diệt các virus mới có khả năng thâm

nhập vào hệ thống. Chính vì vậy, người ta vẫn xem việc thiết kế một phần mềm chống virus thông minh là không tưởng.

Để giúp các hệ thống thông tin có thể chủ động phòng ngừa các virus tin học, chúng ta cần tin học hóa hoàn toàn các công đoạn (1) và (2). Nếu thực hiện được điều này, việc thiết kế một hệ phần mềm chống virus thông minh, có khả năng tự phát hiện, phân tích và tự động tạo thuốc chống các virus mới sẽ hoàn toàn khả thi.

II. TỔNG QUAN VỀ VIRUS TIN HỌC

1. Phân loại virus

Thuật ngữ "Virus tin học" được dùng để chỉ các chương trình tin học có khả năng tự lây lan trong các hệ thống công nghệ tin học. Chúng được thiết kế để phục vụ cho một ý đồ nào đó. Ngoại trừ một số rất ít các chương trình được cài bí mật vào các hệ thống tài chính kế toán để thực hiện lấy cắp mật mã, rút tiền bằng thẻ tín dụng của người khác... các virus phổ biến hiện nay chỉ mang tính lây nhiễm và phá hoại đơn thuần. Tùy theo hành vi lây lan, có thể phân virus tin học thành 2 loại chính:

a. *B-virus:* Loại virus tấn công vào các mẫu tin khởi động: Boot-sector đối với đĩa mềm và Mater-Boot đối với đĩa cứng. Khi máy PC khởi động, các mẫu tin này sẽ được tải vào vùng nhớ để chuẩn bị cho việc tải hệ điều hành. Khống chế các mẫu tin này, B-virus sẽ khống chế tất cả các vụ truy xuất đĩa ở mức BIOS. Vì vậy B-virus dễ dàng lây nhiễm trên tất cả các hệ điều hành, kể cả hệ điều hành UNIX nổi tiếng là "bất khả xâm phạm" đối với virus tin học cũng có thể bị B-virus tấn công một cách dễ dàng (!). B-virus chính là tác nhân phá hoại các thiết bị nlưu trữ thông tin về mặt logic.

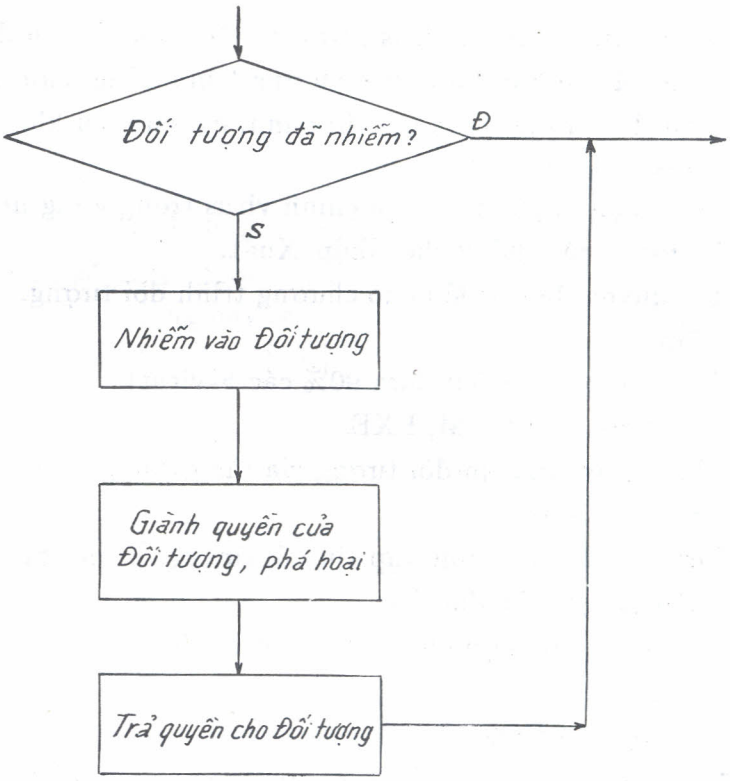
b. F-virus: Loại virus nhiễm vào các tập tin thi hành của hệ điều hành. Đối với DOS, WINDOW 95, đó là tập tin có phần mở rộng là COM và EXE. Tuy F-virus chỉ hoạt động trên một hệ điều hành nhất định nhưng khả năng lây lan của chúng rất mạnh. Chúng là tác nhân chính gây nên các hiệu ứng sai lệch, mất mát dữ liệu cúa hệ thống.

Ngoài ra cũng phải kể đến virus macro sử dụng các công cụ macro của Microsoft Word, Microsoft Exel trên các văn bản, bảng tính... lây nhiễm và phá hoại dữ liệu trên môi trường WINDOW.

2. Cấu trúc của một virus tin học

a. Mô hình tổng quát

Tùy thuộc virus được thiết kế là một B-virus hay F-virus, chúng sẽ có cấu trúc với những đặc thù riêng. Tuy nhiên có thể phác thảo một mô hình tổng quát của chúng như sau:



Thực thể "Đối tượng" có thể là vùng nhớ, mẫu tin khởi động hoặc là một tập tin COM/EXE nào đó. Như vậy về nguyên tắc, khi lây vào đối tượng, *bao giờ virus cũng cất lại thông tin ban đầu của Đối tượng* để trả lại quyền điều khiển cho hệ thống. Tuy nhiên, virus thường mã hóa các thông tin này nhằm gây khó khăn cho việc phân tích cơ chế hoạt động của chúng. Chính việc mã hóa này làm cho các phần mềm chống virus phải luôn đối phó với sự xuất hiện của virus mới. Trong thực tế, *không có một bộ giải mã nào có thể giải được tất cả các bộ mã*, vì thế người ta phải cập nhật các thủ tục giải mã từng virus một vào chương trình chống virus.

b. Các dạng B-virus

Có hai dạng B-virus phổ biến được các hacker khai thác: MB-virus (Master Boot virus) tấn công vào Mater boot đĩa cứng và BS-virus (Boot Sector virus) tấn công vào Boot sector khởi động của hệ điều hành. Cách phát hiện và diệt chúng hoàn toàn như nhau.

c. Các dạng F-virus

Khác với B-virus, F-virus thường phong phú và đa dạng hơn. Có thể chiếm qua một số dạng phổ biến:

(1) Append File Virus:

+ Ghép chương trình virus (progvi) vào cuối tập tin đối tượng.
+ Thay các lệnh ở điểm vào chương trình bằng lệnh nhảy đến progvi.

Khi tập tin đối tượng được gọi thi hành, quyền điều khiển sẽ được trao đổi cho virus, chúng sẽ thực hiện:

+ Nhận dạng sự tồn tại của chính virus trong vùng nhớ.
+ Thường trú, khống chế Nhập/Xuất.
+ Trả quyền điều khiển cho chương trình đối tượng.

* Đặc điểm:

+ Rất phổ biến (chiếm hơn 90% các F-virus).
+ Lây được trên COM, EXE.
+ Kích thước tập tin đối tượng gia tăng (bằng kích thước virus).

(2) Insert File Virus:

+ Dời toàn bộ nội dung tập tin đối tượng về cuối file.
+ Chèn progvi vào đầu file.

Các bước kế tiếp giống như dạng Append File.

* Đặc điểm:

+ Chỉ lây trên COM.
+ Kích thước tập tin đối tượng gia tăng (bằng kích thước virus).

(3) Overwrite File Virus:

+ Tìm buffers đủ lớn trên file.
+ Ghi một phần/toàn bộ progvi vào buffers của file.
+ Ghi thêm phần progvi còn lại cuối file.
+ Thay các lệnh ở điểm vào chương trình bằng lệnh nhảy đến progvi.

* Đặc điểm:

+ Có thể tự đối sang dạng khác nếu không tìm thấy buffers.
+ Kích thước không gia tăng hoặc gia tăng không đáng kể.
+ Khó khôi phục tập tin nhiễm, không bảo đảm chính xác.

(4) Device Virus:

+ Khống chế cấu trúc thư mục trên các entry đặc tả file EXE, COM.
+ Thay địa chỉ cluster file bằng địa chỉ trỏ đến cluster chứa progvi.

Khi file được gọi thi hành, cluster chứa progvi được hệ điều hành tải vào vùng nhớ, virus sẽ:

+ Kiểm tra sự tồn tại của chính nó trong bộ nhớ.
+ Khống chế các khối điều khiển thiết bị (Block device) của hệ thống.
+ Tính toán lại địa chỉ ban đầu của file.
+ Tự nạp file và thi hành.

* Đặc điểm:
  + Cấu trúc đĩa sẽ tê liệt nếu virus chưa thường trú.
  + Lây lan mạnh trên đĩa vật lý.
  + Không thể dùng các trình tiện ích đĩa trên hệ thống nhiễm.
  + Kích thước tập tin không gia tăng.

Trong thực tế, một số virus được thiết kế bằng cách pha trộn các kỹ thuật khác nhau, hoặc biến đổi từ dạng này sang dạng khác. Đôi khi để tăng tầm hoạt động, chúng còn có khả năng lưỡng tính (vừa F-virus, vừa B-virus hoặc ngược lại).

III. NGUYÊN TẮC HOẠT ĐỘNG CỦA MỘT PHẦN MỀM DIỆT VIRUS THÔNG MINH

1. Vùng nhớ

Để diệt virus, chương trình phải vô hiệu hóa các thủ tục khống chế Nhập/Xuất của virus nếu chúng đang thường trú. Nếu không, các thao tác truy nhập của chương trình (đĩa/file) đều bị virus kiểm soát, dễ dẫn đến tình trạng bùng nổ lây nhiễm virus trên hệ thống. Việc vô hiệu hóa virus thường trú là một bộ phận quan trọng và không thể thiếu của chương trình, nó quyết định tính an toàn và hiệu quả của phần mềm. Tiếc thay tiến trình này thường bị coi nhẹ và bỏ qua. Thật ra, chương trình chỉ cần *chiếm lại các ngắt* (Interrupt) của hệ thống đã bị virus chiếm trước đó, hoặc *đổi lại các địa chỉ ngắt, bộ điều khiển thiết bị...*, trở về giá trị ban đầu là virus sẽ không còn khả năng lây / phá hoại.

Thông thường virus chỉ chiếm một số ngắt quan trọng như ngắt 13 h (truy xuất đĩa ở mức BIOS), ngắt 21h (truy xuất tập tin của DOS), ngắt 8h, 1 Ch (đồng hồ máy).... Vấn đề là làm sao tính toán được cá địa chỉ ngắt đó, vì mỗi virus lại lưu địa chỉ ngắt tại những nơi khác nhau? Hơn nữa, tùy theo BIOS của máy, version của hệ thống điều hành... mà các địa chỉ ngắt này lại khác nhau.

Như trên đã nói, hầu như có rất ít phần mềm chống virus, kể cả các phần mềm của nước ngoài, lấy được giá trị chuẩn của các ngắt. Một số phần mềm cúa nước ngoài, lấy được giá trị chuẩn của các ngắt. Một số phần mềm không chạy được khi vùng nhớ đang nhiễm virus, thậm chí đó là những virus mà phần mềm đã biết.

Phần mềm chống virus D2 (Detect and Destroy Viruses) được phát riển trong nước, là phần mềm có nhiều cố gắng trong việc phát biện và vô hiệu hóa các virus thường trú trong vùng nhớ, kể cả cac virus *đã biết* hoặc *chưa biết*. Ngoài khả năng dự đoán sự tồn tại của virus lạ trong vùng nhớ, D2 còn lấy được địa chỉ của các ngắt nguyên thủy. Khi chương trình thi hành, tất cả tác vụ Nhập / Xuất đều được gọi qua địa chỉ này, tránh được tình trạng lây nhiễm hàng loạt virus vào các

đĩa / tập tin sạch. Nếu phát hiện bộ nhớ nhiễm một loại virus lạ, D2 sẽ chiếm lại toàn bộ các gia trị ngắt nguyên thủy. Nhờ đó, virus này sẽ không còn khả năng lây nhiễm.

Vì thế, khả năng phát hiện sự có mặt virus lạ trong vùng nhớ của chương trình sẽ quyết định tính hiệu quả của việc diệt trừ các virus chưa biết trong vùng nhớ. Qua thực tế sử dụng, khả năng phát hiện các virus lạ trong vùng nhớ của D2 khá tin cậy, hầu hết các virus mới đều bị D2 vô hiệu hóa, hạn chế đến mức thấp nhất khả năng "bám trở lại" vào file vừa được kiểm tra của virus.

2. Trên đĩa

Các thao tác trên đĩa chỉ tập trung vào các mẫu tin khởi động, bao gồm Boot sector trên đĩa mềm. master boot và Boot sector trên đĩa cứng. Ngoài tiến trình dò mã các B-virus đã biết, chương trình phải có khả năng dự đoán khả năng xuất hiện một B-virus mới và phục hồi mẫu tin khởi động khi cần thiết.

Ngoài một số rất ít các mẩu tin khởi động chứa đoạn mã thực hiện những nhiệm vụ đặc biệt (chứa tham số, kích hoạt các trình điều khiển thiết bị...), phần lớn các mẫu tin khởi động là do hệ điều hành qui định, có cùng kích thước và dạng mã thi hành. Vì vậy việc dò tìm một virus mới trên các mẫu tin khởi động phổ biến là điều có thể thực hiện được. Tuy nhiên cần chú ý các mẫu tin đặc biệt đã nói. Không có một nguyên tắc cụ thể nào cho việc bố trí các mã lệnh nào đó, cũng như không thể cấm hệ thống sử dụng các đĩa đặc biệt.

Có thể tham khảo phương án diệt B-virus lạ của phần mềm D2:

+ Đảm bảo Master boot/Boot sector của hệ thống là sạch.
+ Cất lại các mẫu tin này dưới dạng tập tin dữ liệu.
+ Mỗi lần chương trình thực hiện:
  - Dùng Int 21h chuẩn để đọc tập tin chứa các mẫu tin khởi động.
  - Dùng Int 13h chuẩn để đọc các mẫu tin khởi động hiện thời.
  - Tiến hành so sánh, nếu khác nhau, phát cảnh báo nguy cơ xuất hiện B-virus mới, yêu cầu người dùng xác nhận việc phục hồi các mẫu tin này.

Với biện pháp này, D2 đã diệt thành công gần như toàn bộ các *B-virus* lạ. Công đoạn quan trọng nhất vẫn là quá trình phân tích tính hợp lệ của mẫu tin khởi động hiện tại. Để hạn chế rủi ro, D2 cung cấp thêm khả năng phục hồi mẫu tin khởi động từ dạng chuẩn, chính là các mẫu tin khởi động mặc định của hệ điều hành.

3. Trên tập tin

Việc phát hiện virus mới trên tập tin là một thử thách lớn đối với tất cả các phần mềm chống virus hiện nay. Hành vi của F-virus tùy thuộc vào ý đồ của tác giả tạo ra nó. Mặt khác, mã lệnh của virus lại không khác mã lệnh của chính tập

tin nó lây nhiễm. Vì vậy nếu chỉ đơn thuần phân tích mã lệnh của tập tin, khó có thể dự đoán về khả năng tồn tại một F-virus mới trên file.

Để tiếp cận quá trình nhận dạng F-virus trên tập tin, chúng ta thử điểm lại các cố gắng đã được các phần mềm chống virus sử dụng:

a. Tạo tập tin tham khảo

+ Đảm bảo tập tin đối tượng là sạch.

+ Ghi nhận các thông tin tối thiểu về tập tin đối tượng (Tên, kích thước, ngày giờ, đoạn mã điểm vào lệnh...) vào tập tin tham khảo trên thư mục chứa file.

+ Mỗi khi duyệt cây thư mục, chương trình sẽ đối chiếu thông tin trên tập tham khảo với các tập tin trên như mục đang duyệt. Nếu sai, chương trình sẽ phục hồi các thông tin cũ từ tập tham khảo.

- Ưu điểm:

1. Dễ thực hiện.
2. Đạt hiệu quả tốt đối với các F-virus đơn giản.

- Nhược điểm:

1. Người dùng "dị ứng" vì sự xuất hiện các tập tin tham khảo trên các thư mục.
2. Không hiệu quả nếu virus có khả năng gây nhiễu các chức năng lấy kích thước tập tin của hệ điều hành.
3. Phá hỏng file nếu F-virus thuộc các loại:

-Insert

- Overwrite
- Mã hóa

4. Không phát hiện được virus mới file, dễ ghi nhận thông tin của file nhiễm virus vào tập tin tham khảo.
5. Không tin cậy trên những hệ thống có độ biến động chương trình/dữ liệu cao.

b. Tạo thư viện Backup

+ Tạo tập tin thư viện chứa các chương trình sạch.

+ Chép lại các chương trình bị nhiễm virus mới từ thư viện đã tạo.

- Ưu điểm:

1. Dễ thực hiện.
2. Các chương trình được khôi phục nguyên vẹn, kể cả các trường hợp file bị hư đơn thuần.
3. Chống được tất cả các F-virus mới.

- Nhược điểm:

1. Chiếm dụng không gian đĩa.

2. Không phát hiện được virus mới trên file, dễ ghi lầm file nhiễm virus vào thư viện backup.

3. Không tin cậy trên những hệ thống có độ biến động chương trình / dữ liệu cao.

c. Tạo "vắc-xin"

+ Đảm bảo tập tin là sách.

+ Đính vào cuối tập tin một đoạn mã.

+ Chuyển điểm vào lệnh đến mã vừa đính vào.

Khi tập tin được gọi thi hành, đoạn mã này có nhiệm vụ kiểm tra kích thước tập tin. Nếu kích thước hợp lệ, trả quyền điều khiển cho chương trình, Ngược lại, phát cảnh báo, phục hồi nội dung, kích thước cũ của tập tin nếu người dùng chấp nhận.

- Ưu điểm:

1. Dễ thực hiện.

2. Đạt hiệu quả tốt đối với các F-virus đơn giản.

3. Khắc phục nhược điểm 3 của phương pháp Tạo mẫu tập tin tham khảo

- Nhược điểm:

1. Tăng kích thước file, gây cảm giác hoang mang cho người dùng.

2. Không hiệu quả nếu virus có khả năng gây nhiễm các chức năng lấy kích thước tập tin của hệ điều hành.

3. Không phát hiện được virus mới trên file, vì thế dễ vô tình "nhốt" virus vào file khiến việc phục hồi gặp nhiều khó khăn.

d. Tạo thư viện virus do người dùng định nghĩa

+ Cho phép người dùng mở tập tin thư viện riêng, chứa các mẫu virus mới do người dùng định nghĩa (tạm gọi làUserVirus).

+ Chương trình đối chiếu file đang kiểm tra với các mẫu UserVirus trong thư viện. Nếu phát hiện, chương trình sẽ xóa file chứa các UserVirus.

- Ưu điểm:

1. Có thể nhận dạng các F-virus mới.

2. Linh hoạt, có khả năng mở cho người dùng tham gia vào việc bảo vệ hệ thống.

Nhược điểm:

1. Chỉ phát huy tác dụng đối với người dùng am hiểu biết về virus tin học, có khả năng chỉ định chính xác đoạn mã virus. Ngược lại, sẽ gây nên

một số hiệu ứng ngược nguy hiểm: file bị xóa lầm, đoạn mã bị mất tác dụng...

2. Giảm tốc độ thực hiện. Chương trình cồng kềnh vì phải chứa các công cụ hỗ trợ thư viện UserVirus (tìm kiếm, chèn, xóa...).

3. Mất tác dụng đối với các F-virus có sử dụng kỹ thuật mã hóa.

Các kỹ thuật nêu trên đều có những ưu/nhược điểm riêng. Nhưng nhược điểm chung dễ thấy đó là thiếu các giải thuật tự phân tích, đánh giá tính trong sạch của một tập tin bất kỳ. Nếu có thêm các mô-tơ suy diễn đủ mạnh thì hệ thống sẽ tỏ ra tin cậy và thông minh hơn.

IV. GIẢI PHÁP ĐỀ NGHỊ

Như trên đã dẫn chứng, việc thiết kế một bộ giải mã nhằm cố gắng tìm ra chìa khóa áp dụng cho tất cả các F-virus mới là điều không thể thực hiện được. Vì vậy, vấn đề chỉ còn giới hạn trong phạm vi hẹp: nhận dạng khả năng tồn tại một F-virus mới trên một file bất kỳ để quyết định có nên lưu trữ lại các thông tin về file dành cho các lần chạy sau, hay là xóa hẳn file này. Việc xóa file không phải là một giải pháp xấu như nhiều ý kiến phê phán, vì các file COM và EXE chỉ có giá trị trên một hệ cụ thể và thường được lưu lại trên các đĩa dự phòng của người dùng. Nếu không, người dùng cũng có thể cài đặt lại phần mềm từ các bộ đĩa gốc khi các file thi hành nhiễm virus đã bị xóa bởi một phần mềm chống virus nào đó.

Việc thiết kế một mô-tơ suy diễn có khả năng nhận dạng sự tồn tại của F-virus mới trên tập tin chủ yếu dựa vào hành vi lây nhiễm trên tập thi hành của virus. Để tăng tính hiệu quả, chương trình có thể trang bị thêm một số công cụ cho phép lần vết (trace) theo các chỉ thị của file, kết hợp với một số thủ thuật bẫy (trap) và gỡ rối (debug).

Nếu kết hợp quá trình phân tích này với các kỹ thuật truyền thống đã nói, chúng ta sẽ thiết kế được hệ mong muốn. Có thể chọn *Tạo thư viện Backup* vì phương pháp này tỏ ra hiệu quả nhất. Việc sử dụng mô-tơ suy diễn nhận dạng F-virus mới sẽ khắc phục nhược điểm (2) và (3). Nhược điểm (1) có thể được khắc phục bằng cách kết hợp giải pháp nén với kỹ thuật tạo tập chỉ mục (index) nhằm tăng tốc độ thi hành chương trình.

V. KẾT LUẬN

Việc thiết kế phần mềm chống virus thông minh là một đề tài gây nhiều tranh luận. Nó chứng minh cho ý tưởng thay thế các hoạt động của con người bằng các chương trình máy tính, hay nói cách khác, đó chính là một phần khoa học Trí tuệ nhân tạo.

Trong bài viết này, chúng tôi đã sử dụng một số kỹ thuật của Hệ Chuyên gia, kết hợp với việc sử dụng các kinh nghiệm được tin học hóa nhằm đưa ra một mô hình phần mềm mong muốn. Đây chỉ là một phương án, chưa thể đặc trưng cho toàn bộ công việc của các chuyên gia trong thế giới thực. Tuy hệ có thể đáp ứng được yêu cầu của bài toán: *phát hiện và diệt trừ các virus mới* nhưng mức độ "thông minh" của hệ cũng còn giới hạn. Nếu điều kiện cho phép, việc mở rộng các mô-tơ suy diễn tiến dần đến hoạt động của các chuyên gia trong thế giới thực, sẽ trở thành hiện thực.

TÀI LIỆU THAM KHẢO

1. Trương Minh Nhật Quang, *Các nguyên tắc phòng chống virus sinh học.* Tin học và Đời sống, số 5-6 (1994).
2. Trương Minh Nhật Quang, *DOS 5.0 vô hiệu hóa virus Dir 2/FAT.* Tin học và Đời sống, số 7-8 (1992).
3. Trương Minh Nhật Quang, *Hãy cảnh giác virus Wai-Chan 94.* PC-WORLD Việt Nam, tháng 10 - 1996.
4. Nguyễn Thanh Thủy, *Hệ chuyên gia, các kỹ thuật xây dựng cơ sở tri thức.* Hội nghị khoa học Viện Công nghệ thông tin, 1996.
5. Trương Minh Nhật Quang, *Hỏi đáp về virus tin học.* Tin học và Đời sống, số 9-10/1994.
6. Trương Nhật Minh Quang, *Sự tấn công lén lút của loại virus mới trên mạn gmáy tính.* John Dehaven (Byte 5/1993), dịch, Tin học và Đời sống, số 9-10 (1994).
7. Richard B. Levin, *The computer Virus HandBook*, Osborne/McGraw-Hill, 1990.
8. Dave Williams, *The Programmer's Technical Referance: MS-DOS.* IBM PC & Compatibles. Tech Publications Pte Ltd-Sigma Press, England, 1993.
9. Ngô Anh Vũ, *Virus tin học, huyền thoại và thực tế.* NXB Tp. Hồ Chí Minh, 1991.

(1) Đại học Bách khoa Hà Nội.
(2) Viện Tin học tiếng Pháp.

Nhận bài ngày 2-5-1997