ĐỀ XUẤT MÔ HÌNH GẮN KÈM MÃ THỰC THI VÀO DỮ LIỆU THI HÀNH

Tống Minh Đức*

Tóm tắt: *Bài báo đề xuất mô hình gắn kèm mã thực thi vào dữ liệu thi hành trên môi trường windows, vượt qua cảnh báo của một số phần mềm anti-virus, dựa trên việc nghiên cứu kỹ thuật và mô hình lây nhiễm của virus. Mô hình và kỹ thuật cài đặt đề xuất làm giảm thiểu sự thay đổi đến cơ chế thi hành của dữ liệu gốc. Mô hình được ứng dụng trong trường hợp khi cần cài đặt một số chức năng cho hệ thống để kiểm soát, điều tra, thu thập thông tin từ hệ thống máy tính. Với việc cập nhật mô hình mới, giúp cho việc tăng cường khả năng kiểm soát các dữ liệu thi hành với những loại biến thể mới của virus.*

Từ khóa: Bảo mật máy tính, Mã độc, Virus, Anti-virus, Mô hình lây nhiễm virus.

1. ĐẶT VẤN ĐỀ

Hiện nay, xu hướng tấn công, phát tán phần mềm có mã độc vào các cơ quan, doanh nghiệp là hình thái mới của giới tội phạm mang tính chất quốc tế và đã xuất hiện tại Việt Nam. Bên cạnh các loại mã độc phổ biến, ngày càng xuất hiện nhiều các dạng mã độc mới. Khi xâm nhập vào hệ thống máy tính, các mã độc âm thầm kiểm soát toàn bộ máy tính nạn nhân, mở cổng hậu (backdoor), cho phép tin tặc điều khiển máy tính nạn nhân, và lấy cắp dữ liệu …

Trong khoa học máy tính, mã độc (malware) được định nghĩa là một chương trình được chèn một cách bí mật vào hệ thống với mục đích làm tổn hại đến tính bí mật, tính toàn vẹn hoặc tính sẵn sàng của hệ thống. Thuật ngữ virus máy tính (thường được người sử dụng gọi tắt là virus) là những chương trình hay đoạn mã được thiết kế để tự nhân bản và sao chép chính nó vào các đối tượng lây nhiễm khác (tập tin, ổ đĩa, máy tính, ...) [3].

Những virus mới được biết trong thời gian gần đây đa phần hướng đến việc lấy cắp các thông tin cá nhân nhạy cảm (các mã số thẻ tín dụng, tài khoản, tài liệu mật…) mở cửa sau cho tin tặc đột nhập chiếm quyền điều khiển hoặc thực hiện các hành động khác nhằm có lợi cho người phát tán virus. Chiếm trên 90% số virus đã được phát hiện là nhằm vào hệ thống sử dụng hệ điều hành họ Windows bởi hệ điều hành này được sử dụng nhiều nhất trên thế giới. Do tính thông dụng của Windows nên các tin tặc thường tập trung hướng vào chúng nhiều hơn là các hệ điều hành khác.

Một hình thức tấn công mới đặc biệt nguy hiểm ngày nay là các phần mềm mã độc xâm nhập vào hệ thống thông qua nhiều hình thức khác nhau, nằm ẩn trong hệ thống, chờ cơ hội ra tay thực hiện ý đồ của mình. Để có thể phát hiện được các loại mã độc này phần mềm anti-virus ngoài việc phát hiện, việc cập nhật các mã độc đã biết, phần mềm anti-virus còn phải phân tích, nhận diện các loại mã độc mới để cập nhật vào hệ thống. Với những mã độc mới, tùy vào khả năng phát hiện của các phần mềm anti-virus mà phần mềm cảnh báo cho người sử dụng cho phép cách ly, hoặc cho phép người sử dụng lựa chọn thi hành.

Đã có nhiều nghiên cứu liên quan tới bài toán phát hiện và nhận dạng mã độc [2, 4, 5, 6]. Thường bài toán phân tích được chia làm 2 loại: Phân tích tĩnh dựa trên đoạn mã đặc trưng, và phân tích động dựa trên nhận diện hành vi. Trường hợp phân tích tĩnh tỏ ra khá hiệu quả đối với các mã độc đã được biết trước, các phần mềm anti-virus dựa trên các đoạn mã đặc trưng nhận diện mã độc, ngăn chặn và cách ly hiệu quả. Tuy nhiên, vấn đề gặp khó khăn khi phải đối mặt với những mã độc nằm ẩn mình trong máy tính, chờ cơ hội, thời gian hợp lý ra

tay phá hoại. Các mã độc này là loại khó diệt nhất và nguy hiểm nhất đối với người sử dụng, nó có thể ẩn mình trong hệ thống lâu dài mà người dùng không hề hay biết.

Đối với kỹ thuật phân tích động, dựa trên hành vi của mã độc để phân tích đã tỏ ra khá hiệu quả với những loại mã độc mới hoặc biến thể của nó, dựa vào các hành vi như: mở dữ liệu, ghi kèm vào dữ liệu, tìm dữ liệu, mở cổng để truy cập nối mạng, gửi tài liệu ra ngoài, quyết tìm các dữ liệu trên ổ đĩa. Tuy nhiên, kỹ thuật này phụ thuộc rất lớn vào việc mã độc phải thi hành ngay. Một số mã độc sử dụng kỹ thuật Bypass ASLR (Address space layout randomization) vượt qua cơ chế bảo mật trên các hệ điều hành windows. Để chương trình chiếm quyền hệ thống, toàn quyền sử dụng các API mà hệ thống cung cấp, một phương pháp khá phổ biến được nhiều loại mã độc sử dụng hiện nay đó là khai thác lỗ hổng 0 days, leo thang đặc quyền [7].

Trong thực tế, cũng có những trường hợp khi cần điều tra, giám sát, thu thập thông tin từ hệ thống máy tính quản lý, hoặc cần cài đặt một số chức năng cho hệ thống để kiểm soát hệ thống dữ liệu, phục vụ công tác giám sát, cũng như tăng cường khả năng kiểm soát với những loại mã độc phát triển theo mô hình mới, chẳng hạn trường hợp cài đặt kiểm soát trong: trong các dữ liệu thực thi, cài đặt thêm một đoạn mã để trước khi chạy chương trình, đoạn mã được thi hành trước để phát hiện những thay đổi của dữ liệu chủ. Thường các đoạn mã cài đặt theo các mô hình truyền thống của mã độc dễ bị các hệ thống anti-virus phát hiện, cảnh báo.

Trong bài báo, tác giả tập trung nghiên cứu, phân tích các kỹ thuật đặc trưng của virus và mô hình của virus gắn kèm vào dữ liệu thực thi. Sau khi phân tích các yếu tố dễ làm thay đổi cơ chế dữ liệu thi hành, tác giả đề xuất mô hình gắn kèm mã thực thi vào dữ liệu với sự thay đổi cơ chế thi hành dữ liệu chủ ít nhất, sử dụng mã hóa mã lệnh để che giấu đoạn mã thực thi, nhằm trách sự kiểm soát của các hệ thống anti-virus. Qua nghiên cứu giúp chúng ta làm chủ được hệ thống với các mô hình mới của virus, kết quả nghiên cứu còn được ứng dụng phục vụ công tác quản lý, giám sát, thu thập thông tin điều hành.

2. MÔ HÌNH ĐÍNH KÈM MÃ LỆNH VÀO DỮ LIỆU THI HÀNH

Trong mô hình đính kèm mã lệnh vào dữ liệu thi hành thì yếu tố bắt buộc là đoạn mã đính kèm phải được thi hành trước dữ liệu chủ, vì trong trường hợp dữ liệu chủ được thi hành trước, mã độc sẽ rất khó kiểm soát khi nào thì chương trình chủ trao quyền điều khiển cho mã độc. Chính vì vậy, mã độc sẽ chiếm quyền kiểm soát trước, sau khi thi hành xong các chức năng của mình, mã độc sẽ trao lại quyền điều khiển cho dữ liệu chủ thi hành. Để thực hiện được kỹ thuật này, virus đã phải điều hướng lệnh thi hành xuống đoạn chương trình virus thông qua một số kỹ thuật như: thay đổi đầu vào OEP (Original Entry Point) của chương trình xuống vị trí chứa đoạn mã virus hoặc sử dụng đoạn lệnh nhảy (jump) ngay từ lệnh đầu tiên của chương trình xuống virus. Trong phần này sẽ đi sâu phân tích các mô hình lây nhiễm của virus và ưu nhược điểm.

2.1. Mô hình lây nhiễm của virus

Virus dữ liệu khi tiến hành lây nhiễm phải tuân theo nguyên tắc: quyền điều khiển phải thuộc về virus trước khi virus trả lại quyền điều khiển cho dữ liệu bị lây nhiễm. Tất cả các dữ liệu của dữ liệu phải được bảo toàn sau khi quyền điều khiển thuộc về dữ liệu. Dưới đây trình bày một số mô hình cơ bản lây nhiễm của virus lây vào dữ liệu dựa trên vị trí của virus: mô hình ghi đè, mô hình chèn vào đầu, mô hình đính kèm vào cuối.

2.1.1. Mô hình ghi đè

Mô hình này khá đơn giản khi virus lây nhiễm vào dữ liệu chủ thì virus chỉ việc ghi đè mã lệnh lên dữ liệu chủ, đảm bảo đầu vào của chương trình chủ là đầu vào của virus. Do đó, khi chương trình chủ chạy thì virus chiếm quyền điều khiển và thi hành mã lệnh virus.

Có thể dễ dàng nhận thấy rằng do dữ liệu chủ không còn hoạt động được nên virus dễ bị phát hiện và ngăn chặn. Mặt khác, với virus hoạt động theo phương thức ghi đè, không có khả năng "tháo gỡ" virus để trả lại dữ liệu chủ như ban đầu theo cách hiểu truyền thống. Một trong những virus được thiết kế ghi đè theo mô hình trên đã hoạt động rất hiệu quả là Virus LOVELETTER.

Mô hình này có ưu điểm là khá đơn giản do virus không cần định vị lại trên vùng nhớ. Khi virus chạy không lưu giữ lại phần dữ liệu của chương trình chủ nên sau khi virus lây nhiễm, dữ liệu chủ sẽ không còn hoạt động được. Nhược điểm của mô hình ghi đè mang tính chất phá hoại nhiều hơn và khả năng lây lan của virus sử dụng mô hình này là rất hạn chế. Do dữ liệu chủ bị hỏng nên mô hình này khó phát tán được virus.

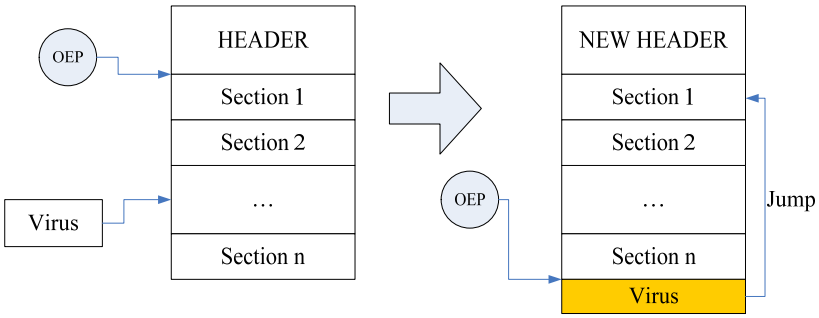
2.1.2. Mô hình ghi chèn vào đầu.

Virus chèn đoạn mã của nó vào đầu dữ liệu bị lây nhiễm và đưa toàn bộ nội dung dữ liệu xuống phía dưới ở ngay sau chương trình virus.

Nhược điểm của mô hình này do đầu vào của dữ liệu bị rời một đoạn bằng độ dài của virus gán vào dữ liệu nên nếu giữ nguyên thì dữ liệu không thi hành được vì bị sai lệch địa chỉ tuyệt đối. Do đó, trước khi virus trả quyền điều khiển cho dữ liệu thi hành thì virus cần phải tải lại dữ liệu đúng như hành ảnh ban đầu của dữ liệu khi chưa bị lây nhiễm, việc thi hành dữ liệu sẽ chậm gây ảnh hưởng đến bộ nhớ. Vì vậy, mô hình này trong thực tế ít được sử dụng.

2.1.3. Mô hình chèn vào cuối dữ liệu

Mô hình lây nhiễm chèn mã virus vào cuối dữ liệu là mô hình được sử dụng ở hầu hết các loại dữ liệu, do tính chất lây nhiễm không làm ảnh hưởng nhiều tới vị trí tải dữ liệu gốc vào bộ nhớ và thi hành dữ liệu gốc, giảm được nhiều thời gian tải lại dữ liệu vào đúng vị trí như mô hình lây nhiễm vào đầu dữ liệu, giảm được thời gian chuyển quyền điều khiển cho dữ liệu, vì vậy, mô hình được sử dụng nhiều hơn so với các mô hình khác. Mô hình này mã virus sẽ được gắn vào cuối dữ liệu chủ sau khi lây nhiễm. Do mã của virus không nằm đúng đầu vào chương trình OEP (Original Entry Point) nên virus sẽ định vị lại dữ liệu bị lây nhiễm bằng cách thay đổi dữ liệu của dữ liệu sao cho quyền điều khiển trỏ vào đúng mã lệnh của virus, và sử dụng kỹ thuật định vị chương trình để tính toán các địa chỉ trong đoạn mã lệnh virus sử dụng.



Hình 1. *Mô hình ghi vào cuối dữ liệu.*

Mô hình lây nhiễm ghi mã virus vào cuối dữ liệu có ưu điểm có thể lây nhiễm trên mọi dữ liệu thi hành. Mặt khác, sự thay đổi dữ liệu trên dữ liệu bị lây nhiễm là không đáng kể và việc chiếm quyền điều khiển của virus là không mấy khó khăn nên mô hình này được áp dụng rất nhiều.

Nhược điểm của mô hình ghi mã virus vào cuối dữ liệu là rất dễ dàng cho người diệt virus trong việc khôi phục dữ liệu và việc phát hiện có mã đính kèm.

Với kỹ thuật phân tích tĩnh, dựa vào dấu hiệu của virus khi lây nhiễm do chuyển đoạn mã thi hành ngay xuống section cuối, một số phần mềm anti-virus hiện nay lợi dụng vào đặc điểm này để phát hiện, đưa ra cảnh báo có sự xuất hiện của mã mới trên dữ liệu thi hành. Phần sau chúng tôi đề xuất mô hình gắn kèm mã thực thi vào dữ liệu thi hành, mô hình cải tiến tránh được việc chuyển ngay đoạn mã thực thi xuống section cuối dữ liệu như các cách lây nhiễm truyền thống của virus, đề xuất kỹ thuật che giấu mã thi hành bằng kỹ thuật mã hóa dữ liệu, có thể qua mặt được một số phần mềm cảnh báo anti-virus.

3. PHÁT TRIỂN MÔ HÌNH GẮN KÈM MÃ VÀO DỮ LIỆU THI HÀNH

3.1. Mô hình đề xuất

Với những phân tích ở trên, chúng tôi đề xuất mô hình gắn kèm mã thi hành vào dữ liệu thi hành được, yêu cầu mô hình đề xuất nhằm đảm bảo 2 yếu tố:

- Đoạn mã thi hành trên dữ liệu phải được nằm trên section đầu tiên (Code –payload).
- Tạo thêm 01 section cuối dữ liệu, phần đoạn mã sẽ được mã hóa (data) và gắn vào section cuối cùng mới tạo.

Ở mô hình này, chương trình virus sẽ sinh đoạn payload "chèn" vào phần trống của section đầu tiên, chuyển con trỏ OEP vào đoạn code mới thêm vào, đồng thời mở thêm 1 section mới chứa dữ liệu data chính là dữ liệu virus đã được mã hóa. Đoạn payload thực hiện giải mã data, nạp lên vùng nhớ thực thi. Việc này chính là thực thi đoạn mã một cách gián tiếp thông qua 1 dữ liệu PE khác, nhưng lại là thực thi trực tiếp qua việc nạp thẳng đoạn chương trình lên vùng nhớ thực thi.

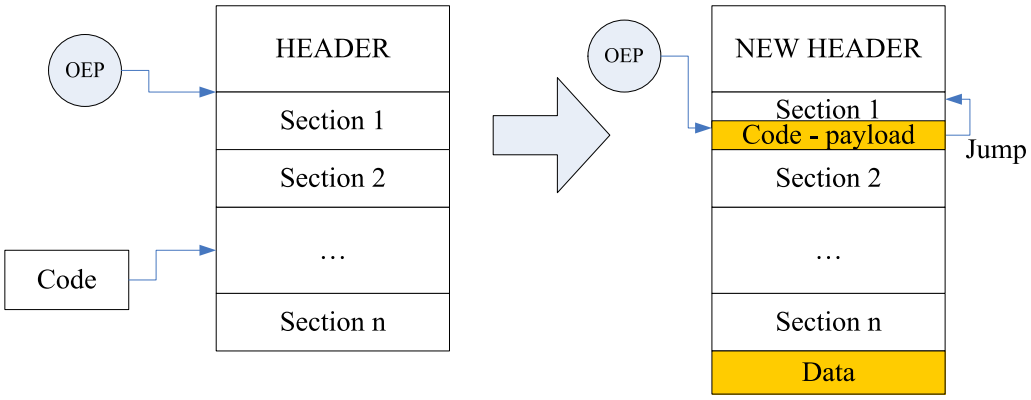
Khi dữ liệu thi hành đoạn mã Code-payload có nhiệm vụ:

- Giải mã đoạn dữ liệu được ghi trong section cuối;
- Tạo dữ liệu mới, copy đoạn dữ liệu trong section cuối vào dữ liệu mới tạo;
- Thực thi dữ liệu vừa tạo;
- Trả lại điều khiển cho dữ liệu chủ.

Dữ liệu vừa tạo có nhiệm vụ:

- Lây nhiễm vào các dữ liệu khác theo mô hình mới đề xuất;
- Thực hiện hoạt động khác (theo mục đích người viết).

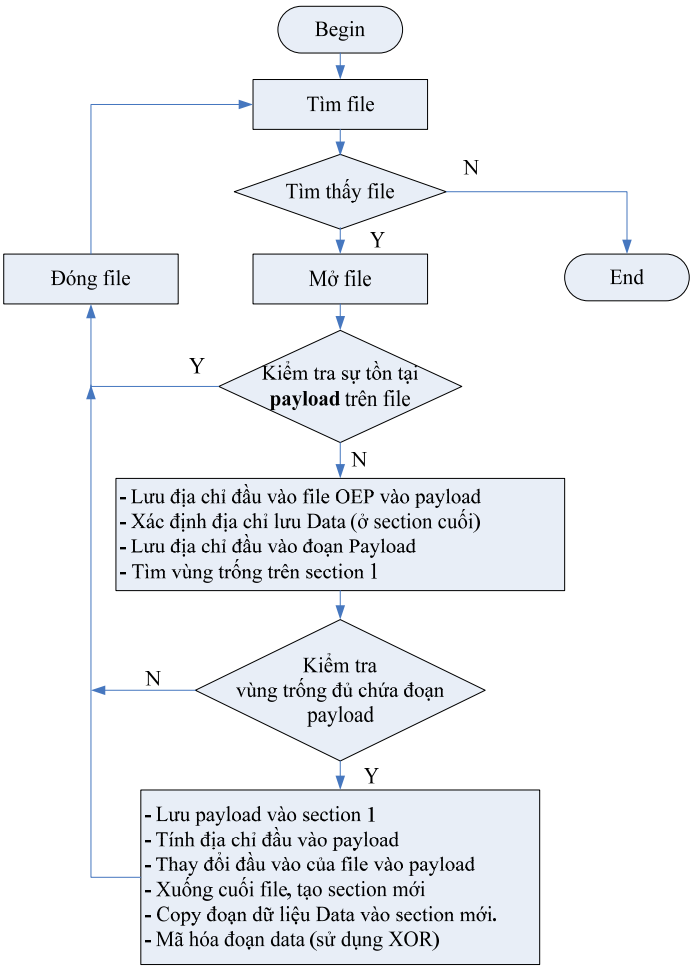
Mô hình cải tiến:



Hình 2. *Mô hình cải tiến ghi vào cuối dữ liệu.*

3.2. Sơ đồ thuật toán

Thuật toán sau mô tả các thao tác của đoạn mã data sau khi được giải mã đưa lên vùng nhớ:



Hình 3. *Sơ đồ thuật toán thi hành dữ liệu khi tải lên vùng nhớ.*

3.3. Kết quả thử nghiệm

Chương trình gắn kèm đoạn mã vào dữ liệu thi hành được viết theo mô hình cải tiến, viết trên ngôn ngữ C++, trình biên dịch VC++ 2012, Platform 64bit.

Kết quả được thử nghiệm trên hệ điều hành windows 64 bit. Tác giả đã lần lượt thử nghiệm với hệ thống được cài đặt được một số phần mềm anti-virus như: Kapersky Antivirus 2016 và Bitdefender antivirus 2016. Ngoài ra, tác giả cho thử nghiệm upload dữ liệu sau khi đính kèm mã viết theo mô hình đề xuất lên trang kiểm tra mã độc online www.virustotal.com cho kiểm tra. Các phần mềm anti-virus hiện nay và hệ thống virus-total đều không phát hiện ra dấu hiệu nghi nghờ và không có cảnh báo.

4. KẾT LUẬN

Trong các hệ thống anti-virus hiện nay, ngoài việc phân tích các loại mã độc và các hành vi của mã độc để phát hiện và loại bỏ các loại mã độc đã biết trước, việc phân tích cơ chế, dự đoán các mã độc đã nằm ẩn trong máy tính từ lâu, chờ cơ hội thuật lợi để ra tay là vấn đề rất cần thiết. Đối phó với những loại mã độc này là thách thức lớn trong lĩnh vực bảo đảm an toàn thông tin. Việc nghiên cứu các kỹ thuật và mô hình lây nhiễm của virus trên dữ liệu thi hành giúp làm chủ được hệ thống trong trường hợp có virus mới tấn công hệ thống. Hơn nữa, việc nghiên cứu mô hình lây nhiễm giúp cho chúng ta có thể cài đặt vào hệ thống phần mềm hữu ích, giúp giám sát sự thay đổi của các dữ liệu, hoặc giám sát

hệ thống. Việc phân tích và tìm hiểu mô hình ít thay đổi nhất về cơ chế thi hành trong dữ liệu cũng là hướng đi mới để các nhà phân tích virus quan tâm.

Trong bài báo mới chỉ đề cập tới mô hình lây nhiễm và thuật toán cài đặt mã thi hành lên dữ liệu đảm bảo dữ liệu thi hành vẫn hoạt động được và có sự thay đổi ít nhất về cơ chế thi hành dữ liệu. Tác giả hướng tới việc đoạn mã nằm ẩn trong dữ liệu để có thể trách được một số cảnh báo của các phần mềm anti-virus hiện nay trong trường hợp kiểm tra trên dữ liệu tĩnh cũng như thi hành trên bộ nhớ mà hệ thống cho phép thi hành dữ liệu. Tuy nhiên, để có thể thi hành được đoạn mã qua mặt được các hệ thống hạn chế quyền truy cập thì cần đòi hỏi phải sử dụng một số kỹ thuật Bypass ASLR ..., loại bỏ được các tiến trình thì hệ thống phải được cài đặt với quyền NTAUTHORIZE/SYSTEM. Vấn đề để thi hành ở mức hệ thống, trong trường hợp hạn chế quyền truy cập chúng tôi sẽ có những nghiên cứu và phân tích tiếp theo.

TÀI LIỆU THAM KHẢO

[1]. Aman Hardikar .M, "*Malware 101 – viruses*", SANS Institute, 2008.

[2]. Dennis Distler, "*Perfoming Behavioral Analysis of Malware*", SANS Training, 2011.

[3]. Karen Kent, Murugiah Souppaya, "*Guide to Malware Incident Prevention and Handling for Desktops and Laptops -Special Publication 800-83*", NIST(National Institute of Standards and Technology) (2013), tr. 2-6.

[4]. Konrad Rieck, Philipp Trinius, Carsten Willems and Thorsten Holz, "*Automatic Analysis of Malware Behavior using Machine Learning*", Journal of Computer Security, Volume 13, Number 4/2011, tr. 639-668.

[5]. Nicolas Falliere, "*Windows Anti-Debug Reference*", (2012)

[6]. Michael Sikorski, Andrew Honig, "*Practical Malware Analysis*". San Francisco, 2012.

[7]. Morton Christiansen, "*Bypassing Malware Defenses*", SANS Institute Reading Room (2010), tr. 17-34.

[8]. Philipp Trinius, Thorsten Holz, Konrad Rieck and Carsten Willems, "*A malware Instruction Set for Behavior-Based Analysis*", Sicherheit Berlin, Germany, 2010, tr. 205.

[9]. Peter Szor, "*The Art of Computer Virus Research and Defense*", 2005.

[10]. http://www.bkav.com.vn/hoi-dap/-/chi_tiet/399683/vu-tan-cong-vietnam-airlines-hacker-da-su-dung-virus, 2016.

ABSTRACT

AN INSERTED CODE INTO EXECUTED DATA MODEL

The paper proposes a model that an executable data is inserted using some codes into the windows environment. This modified data can bypass specific alerts of the anti-virus software. Depending on the virus techniques and models that are empoyed, our work proposes new techniques and model that can minimize the changes to the executed mechanism of the original data. Our model is applied for the purposes of implementing some system functions in order to control, investigate,and collect system information. In addition, the model has the capacity to minimise new viruses in the field of execution.

Keywords: Computer Security, Malware, Virus, Anti-virus, Virus infection model.

Nhận bài ngày 26 tháng 12 năm 2016
Hoàn thiện ngày 06 tháng 02 năm 2017
Chấp nhận đăng ngày 18 tháng 8 năm 2017

Địa chỉ: [1] Học viện Kỹ thuật quân sự.
[*] Email: ductm75@gmail.com.