ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

LUẬN VĂN TỐT NGHIỆP

PHÁT TRIỂN
HỆ THỐNG BE-PUM

Giáo Viên Hướng Dẫn:
PGS.TS. QUẢN THÀNH THƠ
ThS. NGUYỄN MINH HẢI
ThS. LÊ ĐÌNH THUẬN

Sinh viên thực hiện:
NGUYỄN XUÂN KHÁNH - MSSV: 51101594
NGUYỄN LÂM HOÀNG YÊN - MSSV: 51104402

Giáo Viên Phản Biện:
TS. BÙI HOÀI THẮNG

TP.Hồ Chí Minh, Ngày 6 tháng 12 năm 2015

LỜI CAM KẾT

Chúng tôi xin cam đoan rằng mọi thông tin và công việc được trình bày trong bài báo cáo này ngoài việc tham khảo các nguồn tài liệu khác có ghi đầy đủ trong phần phụ lục các tài liệu tham khảo, thì đều do chính chúng tôi thực hiện và chưa có phần nội dung nào được sao chép từ các đề tài thực tập tốt nghiệp, luận văn tốt nghiệp của trường này và trường khác. Nếu có bất kì sai phạm hay gian lận nào, chúng tôi xin chịu hoàn toàn trách nhiệm trước Ban Chủ Nhiệm Khoa và Ban Giám Hiệu Nhà Trường.

TP. Hồ Chí Minh, Ngày 6 tháng 12 năm 2015

Nhóm sinh viên thực hiện đề tài

LỜI CẢM ƠN

Đầu tiên em xin gửi lời cảm ơn sâu sắc đến thầy PGS.TS. Quản Thành Thơ, thầy đã có những sự định hướng, động viên, giải đáp và quan tâm to lớn đến em và cho cả mọi người đang cùng dưới sự dẫn dắt của thầy. Thầy luôn theo sát mọi người qua từng ngày, từng tuần thực hiện nghiên cứu và đưa ra những gợi ý để công việc được suôn sẻ cũng như đạt kết quả như mong muốn. Những lời chỉ bảo và răn đe của thầy luôn là động lực để em và mọi người hoàn thành nhiệm vụ. Chính phong thái đó của thầy là lý do để em chọn vào tham gia nghiên cứu trong nhóm của thầy; và càng ngày em càng cảm ơn về sự lựa chọn đúng đắn đó.

Kế đến em xin dành sự biết ơn to lớn dành cho ThS. Nguyễn Minh Hải, với em anh không chỉ là một người hướng dẫn đề tài, mà còn là một người anh gần gũi vì những sự trợ giúp ân cần từ những ngày đầu em tham gia cùng mọi người thực hiện nghiên cứu. Làm việc với anh không phải là một sự cứng nhắc và rập khuôn, bởi anh luôn thân thiện thảo luận cùng mọi người để đưa ra những ý kiến, công việc và thời hạn hợp tình, hợp lý. Anh luôn dành sự quan tâm và tận tình cho từng thành viên dưới sự hướng dẫn của anh; kèm theo đó là những lời động viên, chia sẻ để từ đó em có thêm động lực để hoàn thành được đề tài này.

Em cũng xin cảm ơn ThS. Lê Đình Thuận đã giúp em hoàn thiện bài báo cáo này qua việc theo dõi sát sao mỗi ngày trong thời gian viết bài để hướng dẫn, đóng góp ý kiến và giúp em sửa chữa những sai sót có trong báo cáo.

Em xin dành những sự tri ân chân thành và to lớn đến toàn thể quý thầy cô của khoa Khoa học và Kỹ thuật Máy tính mà em đang theo học. Nhờ những sự tận tình giảng dạy, truyền đạt kiến thức và kinh nghiệm quý báu qua từng bài giảng mà em có thêm được sự vun đắp, vững chắc trong vốn hiểu biết của mình; để từ đó làm nền tảng cho em hoàn thành được đề tài ngày hôm nay.

Em cũng không quên bày tỏ lòng biết ơn đến gia đình, người thân của mình, những người đã luôn quan tâm, động viên và chăm sóc em qua từng ngày để em có được sức khỏe, tinh thần và môi trường phát triển, giúp em có được như ngày hôm nay. Cuối cùng em xin cảm ơn những người bạn đã hỗ trợ, chia sẻ để em hoàn thành tốt được đề tài này.

Với toàn bộ sự biết ơn sâu sắc đó, em xin gửi lời chúc đến mọi người đã giúp em làm được những điều đến ngày hôm nay. Chúc cho mọi quý thầy cô, đặc biệt là những người em đã nêu tên ở trên, của khoa Khoa học và Kỹ thuật Máy tính luôn có được một sức khỏe dồi dào để luôn hạnh phúc trong cuộc sống và có khả năng để cống hiến thêm cho khoa, cho trường và cho thế hệ đi sau.

Trân trọng.

TP. Hồ Chí Minh, 19/12/2015

Nhóm sinh viên thực hiện đề tài

TÓM TẮT LUẬN VĂN

Mục lục

Phụ lục

Danh sách hình ảnh

Danh sách các bảng

Chương 1

Giới Thiệu

1.1 Giới thiệu về BE-PUM

BE-PUM tên đầy đủ là Binary Emulation for Pushdown Model generation, là một công cụ dùng để phân tích động mã nhị phân của một chương trình bất kỳ chạy trên kiến trúc X86 của hệ điều hành Microsoft Windows nền tảng 32-bit. Sau khi phân tích, BE-PUM sẽ sinh ra hợp ngữ – mã assembly và đồ thị luồng điều khiển (control flow graph – CFG) của chương trình đầu vào.

BE-PUM được xây dựng chính trên mã nguồn của JakStab nhưng không hạn hẹp ở việc chỉ phân tích tĩnh, BE-PUM có thể phân tích động và chỉ ra lại mỗi dòng lệnh của mã assembly môi trường làm việc của nó là như thế nào. Việc này sẽ giải quyết được những trường hợp phân tính vào những nhánh không cần thiết – không bao giờ được thực thi hoặc khi chương trình đang cố gắng thay đổi chính nội dung của mình.

Với việc phân tích mã nhị phân đó, BE-PUM đang được phát triển để tập trung vào phân tích những phần mềm bị nghi ngờ để rồi sau đó sẽ phát hiện được những kỹ thuật tấn công, và cuối cùng là xác định xem đây có thực là phần mềm gây hại đến máy tính hay không?!

1.2 Mục tiêu đề tài

Trong phạm vi của đề tài thực tập tốt nghiệp, mục tiêu nhắm tới là phát triển hệ thống xử lý các Windows API cho BE-PUM. Với số lượng các API rất lớn hiện có trong hệ điều hành Windows, hiện tại đề tài đang tập trung vào xử lý các API ở phiên bản Win32 API, do hầu hết các phần mềm độc hại mà BE-PUM hướng tới vẫn đang dùng bộ API này; với sự ưu tiên từng bước xây dựng cho các API được dùng phổ biến trước.

Bên cạnh việc nhận thông tin đầu vào từ vùng nhớ đã được xây dựng của BE-PUM và trả về kết quả sau khi gọi API vào đúng địa chỉ cần thiết, điều quan trọng là phải đảm bảo không gây ngắt quãng cũng như tránh nguy hại hệ thống đang chạy. Và như vậy với những tương tác vật lý từ lời gọi API (bộ lưu trữ máy tính, cơ sở dữ liệu registry...) hay tương tác người dùng (API tạo cửa sổ message box, lệnh cho một thread "ngủ đông" trong một khoảng thời gian,...) cần được kiểm soát để không làm ảnh hưởng tới kết quả thực thi của BE-PUM.

Lưu ý: do nội dung đề tài tập trung vào xử lý cho Win32 API, nên kể từ đây, khi báo cáo nhắc đến Windows API tức là nói đến Win32 API.

1.3 Cấu trúc của báo cáo

Bài báo cáo này bao gồm những đề mục sau đây:

Chương 1

Giới thiệu tổng quan về BE-PUM, yếu tố quyết định để cho ra đề tài này; dẫn nhập về Windows API, thành phần sẽ được áp dụng để phát triển cho BE-PUM; và cuối cùng nêu ra được mục đính chính của đề tài sẽ cần làm gì

.

Chương 2

Đem đến những cái nhìn về những vấn đề đã và đang được lưu tâm khi thực hiện đề tài này; sự phổ biến của Windows API trong những phần mềm độc hại để thấy sự cần

thiết của việc xây dựng một bộ xử lý Windows API cho BE-PUM; những khó khăn khi thực hiện điều đó và giải pháp cho vấn đề.

Chương 3

Trình bày những kiến thức cần thiết cho quá trình thực hiện đề tài; từ những kiến thức phải nắm được về hệ thống BE-PUM do đây là một đề tài làm việc dựa trên đó; và mỗi khi làm việc với một thư viện bất kỳ, đòi hỏi ta phải tìm hiểu cách thức làm việc với thư viện đó và cả những kiến thức cần thiết do bộ thư viện ấy yêu cầu.

Chương 4

Mỗi chương trình bất kỳ đều cần một thiết kế tốt để giúp cho việc xây dựng dễ dàng và quy chuẩn hơn. Mục này sẽ trình bày cách mà bộ xử lý Windows API đã được hiện thực để tương tai sau này có thể dễ dàng sửa chữa, bảo trì và bổ sung thêm vào kiến trúc đó.

Chương 5

Trình bày về kết quả mà bộ xử Windows API đã đạt được với những Windows API đã được hỗ trợ cho hệ thống BE-PUM.

Chương 6

Liệt kê về những tài liệu và nguồn tham khảo có liên quan đến đề tài này.

Chương 2

Phân Tích Vấn Đề

2.1 Những điểm cần nhắc đến trong BE-PUM

2.2 Các câu lệnh hợp ngữ

2.3 Windows API trong những phần mềm độc hại

Để cung cấp sức mạnh và sự tiện lợi cho lập trình viên trong việc viết ứng dụng chạy trên hệ điều hành Windows, các API trong bộ Windows API mở ra nhiều cách thức nhanh chóng và mạnh mẽ cho lập trình viên trong việc tương tác với hệ thống.

Và vấn đề gì cũng có hai mặt của nó, sự hỗ trợ mạnh mẽ đó cũng là con đường đơn giản để các tin tặc áp dụng vào việc xây dựng nên các phương pháp tấn công, cũng như cho ra đời những phần mềm nguy hại (malware), để lại bao hậu quả xấu cho hệ thống máy vi tính trên toàn cầu.

Trong quá trình xây dựng BE-PUM và qua việc phân tích hàng ngàn mẫu malware chạy trên môi trường Windows được phát tán ở khắp nơi trên thế giới, hầu hết những mẫu malware trên đều áp dụng lời gọi Windows API vào cách thức tấn công của chúng. Những phương pháp tấn công phổ biến như SEH hay phương pháp chống phát hiện đều

có sự tồn tại của Windows API trong đó.

Do đó, việc xây dựng một bộ công cụ xử lý những thông tin trả về từ Windows API là rất cần thiết cho việc phát triển hệ thống BE-PUM, một hệ thống tập trung vào phân tích mã nhị phân của malware.

2.4 BE-PUM và Windows API

Mã nguồn của những API trong bộ Windows API được tập đoàn Microsoft giữ kín và không hề công bố. Chỉ có những đặc tả và hướng dẫn sử dụng được Microsoft phổ biến rộng rãi cho lập trình viên. Nghĩa rằng ta chỉ có thể biết được đầu vào của lời gọi và mong muốn đầu ra sẽ như ý, chứ không thể nắm rõ lô-gíc xử lý bên trong của chúng. Điều đó khiến cho việc xử lý đúng đắn một cách tổng quát đối với mọi đầu vào của mỗi API bằng cách viết lại bộ mã xử lý tương ứng của chúng vào trong BE-PUM dường như trở nên không thể.

Hướng tiếp cận hiện tại là tiến hành lấy nội dung bộ nhớ, nội dung các đối số nằm trên stack bên trong BE-PUM và tiến hành gọi thực sự với Windows API, nhận kết quả trả về và nạp lại vào trong BE-PUM để tiếp tục tiến hành phân tích các câu lệnh tiếp theo.

BE-PUM là một dự án được phát triển lên từ nhân của dự án JakStab và được viết hoàn toàn trên ngôn ngữ lập trình Java. Với Windows API thì lại là một câu chuyện hoàn toàn khác, Windows API được phát triển chủ yếu tập trung vào ngôn ngữ lập trình C kèm với các mô tả và cấu trúc dữ liệu được viết trên đó. Thêm lần nữa, việc hiện thực ý tưởng gọi để lấy kết quả Windows API từ trong lòng BE-PUM gặp nhiều khó khăn. Đặc biệt là việc ánh xạ các dữ liệu kiểu cấu trúc giữa hai thành phần trên cũng là một trăn trở.

Vì những lý do trên, cần tìm hiểu một cách thức giải quyết vấn đề nhanh chóng và đơn giản hơn bằng một bộ công cụ nào đó để xử lý rào cản ngôn ngữ giữa Java và C. Thêm vào đó, bộ công cụ này cũng cần có tính linh hoạt và mềm dẻo để cho việc phát

triển về sau được dễ dàng.

2.5 Truy xuất Windows API bên trong BE-PUM thông qua JNA

Vấn đề trên được giải quyết thông qua bộ thư viện Java Native Access (JNA).

Java Native Access là một thư viện được cộng đồng phát triển, nhằm giúp cho các chương trình được viết bằng ngôn ngữ lập trình Java dễ dàng truy cập vào các thư viện native shared mà không cần thông qua Java Native Interface. Thiết kế của JNA cũng cung cấp khả năng này mà không cần bỏ ra nhiều công sức.

Với khả năng ánh xạ dễ dàng giao diện lập trình giữa hai ngôn ngữ Java và C; bao gồm ánh xạ tên hàm, kiểu dữ liệu trả về, kiểu dữ liệu của các thông số đầu vào; từ những kiểu dữ liệu cơ bản đến những kiểu dữ liệu cấu trúc và kể cả con trỏ; đó là những ưu điểm để lựa chọn JNA áp dụng vào trong việc giải quyết yêu cầu của đề tài nêu trên.

Chương 3

Kiến Thức Nền

3.1 Các câu lệnh hợp ngữ

3.2 Windows API

Chương 4

Thiết Kế Và Xây Dựng

4.1 Các câu lệnh hợp ngữ

4.2 Windows API

Chương 5

Kết Quả

Kết quả

Kiến nghị

Chương 6

Hướng Phát Triển Trong Tương Lai

viet noi dung

Tài liệu tham khảo

1. Kenneth H. Rosen, Chapter 3, Discrete Mathematics and Its Applications, 6th edition, McGraw-Hill, 2007.
2. S. Avora and B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2007.
3. Big O notation - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Big_O_notation
4. Computational complexity theory - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Computational_complexity_theory
5. Phân tích thời gian thực hiện giải thuật. http://thanhcuong.wordpress.com/2011/01/14/phn-tch-thời-gian-thực-hiện-giải-thuật/

   Phân tích thiết kế thuật toán và đánh giá độ phức tạp giải thuật. - Trường đại học Sư Phạm Hà Nội.
6. Bài giảng điện tử trường đại học công nghiệp TP.HCM.
7. Bài giảng môn học phân tích và thiết kế thuật toán - Trường đại học Phương Đông. - Hà Nội - 2008.
8. Lý thuyết ngôn ngữ hình thức và Ôtômat. - Trường đại học Khoa Học - Huế - 2004.
9. Bài giảng điện tử chương 7: Vấn đề NP-đầy đủ. - Đại học Bách Khoa TP.HCM.
10. Giáo trình Lý thuyết thuật toán. - Thai Nguyen University of Information and Communiction Technology, 2010.
11. Lý thuyết độ phức tạp tính toán - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/Lý_thuyết_độ_phức_tạp_tính_toán
12. Luận văn của Phạm Thái Sơn - Sinh viên đại học Bách Khoa Hà Nội - 2009
13. P (độ phức tạp) - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/P_(độ_phức_tạp)
14. NP (độ phức tạp) - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/NP_(độ_phức_tạp)
15. NP-đầy đủ - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/NP-đầy_đủ
16. NP-khó - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/NP-khó
17. Bài toán xếp ba lô - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/Bài_toán_xếp_ba_lô
18. Bài toán người bán hàng - Bách khoa toàn thư mở Wikipedia tiếng việt. http://vi.wikipedia.org/wiki/Bài_toán_người_bán_hàng