BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ

Nguyễn Ngọc Toàn

NGHIÊN CỨU NÂNG CAO HIỆU QUẢ PHÁT HIỆN MÃ ĐỘC IOT DỰA TRÊN HỌC MÁY SỬ DỤNG CÁC ĐẶC TRƯNG CỦA TẬP TIN THỰC THI

Ngành: An toàn thông tin

Mã số: 9480202

TÓM TẮT LUẬN ÁN TIẾN SĨ

HÀ NỘI - 2024

Công trình được hoàn thành tại Học viện Kỹ thuật mật mã

Người hướng dẫn khoa học: PGS.TS. Lương Thế Dũng – Học viện Kỹ thuật mật mã

Phản biện 1: PGS.TS. Bùi Thu Lâm, Học viện Kỹ thuật mật mã
Phản biện 2: PGS.TS. Thân Quang Khoát, Đại học Bách Khoa Hà Nội
Phản biện 3: PGS.TS. Nguyễn Hữu Quỳnh, Trường Đại học CMC

Luận án được bảo vệ trước Hội đồng đánh giá luận án tiến sĩ cấp Học viện họp tại Học viện kỹ thuật mật mã, vào hồi 8 giờ, ngày 25 tháng 10 năm 2024

CÁN BỘ HƯỚNG DẪN KHOA HỌC **NGHIÊN CỨU SINH**

PGS.TS Lương Thế Dũng **Nguyễn Ngọc Toàn**

XÁC NHẬN CỦA ĐƠN VỊ ĐÀO TẠO

Có thể tìm hiểu luận án tại thư viện:
1. Thư viện Học viện kỹ thuật mật mã
2. Thư viện Quốc gia Việt Nam

MỤC LỤC

MỞ ĐẦU

1. Cơ sở đề xuất nghiên cứu

Từ những khảo sát hiệu quả của ứng dụng học máy trong phát hiện mã độc IoT, luận án được thúc đẩy bởi một số vấn đề nghiên cứu mở sau đây:

Thứ nhất, các mô hình phát hiện mã độc IoT đơn kiến trúc dựa trên đặc trưng của tập tin thực thi sử dụng nhiều loại đặc trưng thu thập từ phân tích tĩnh hoặc phân tích động, cần nhiều thời gian thu thập và xử lý lý đặc trưng. Việc sử dụng mạng học sâu, kết hợp nhiều đặc trưng hoặc thuật toán học máy trong xây dựng mô hình phát hiện IoT có thể nâng cao được độ chính xác cho mô hình nhưng chưa hiệu quả cao về số lượng đặc trưng sử dụng, thời gian thực thi tập tin có chứa mã độc, tốc độ tính toán, thời gian phát hiện và khả năng ứng dụng mô hình trong thực tế cho các ứng dụng phát hiện mã độc IoT theo thời gian thực hoặc các thiết bị IoT hạn chế tài nguyên.

Thứ hai, các mô hình phát hiện mã độc IoT đa kiến trúc dựa trên đặc trưng của tập tin thực thi dựa trên thu thập đặc trưng động hoặc tĩnh không phụ thuộc vào nền tảng kiến trúc, sử dụng các mạng học sâu phức tạp hoặc kết hợp nhiều loại mạng học sâu đã đem lại hiệu quả đối với một số dòng mã độc IoT nhưng chưa hiệu quả đối với các mã độc mới, mã độc tấn công có chủ đích hoặc mã độc zero-day trên thiết bị IoT đa dạng nền tảng. Các mô hình huấn luyện được khó triển khai trên các thiết bị IoT hạn chế tài nguyên.

Thứ ba, các nghiên cứu về mã độc trên thiết bị IoT thời gian qua tập trung nhiều vào mã độc trên thiết bị sử dụng hệ điều hành Android và tập trung nhiều vào mã độc IoT Botnet, chưa nghiên cứu các họ mã độc IoT khác. Tập dữ liệu phục vụ huấn luyện và đánh giá mô hình phát hiện trên các kiến trúc vi xử lý còn hạn chế, đặc biệt số lượng tập mẫu trên các kiến trúc vi xử lý của thiết bị IoT mới hoặc chưa được thu thập trước đây. Nhiều tập dữ liệu đa kiến trúc đã công bố và được thử nghiệm trong các nghiên cứu đối mặt với vấn đề mất cân bằng dữ liệu các lớp mã độc, giữa các nền tảng kiến trúc vi xử lý.

Vì vậy, việc nghiên cứu nâng cao hiệu quả cho các mô hình phát hiện mã độc IoT hoạt động đơn nền tảng kiến trúc và đa nền tảng kiến trúc sử dụng đặc trưng của tập thi thực thi là cấp thiết, có ý nghĩa về mặt khoa học và thực tiễn hiện nay.

2. Mục tiêu nghiên cứu

Từ việc phân tích tính cấp thiết trong nội dung trên, luận án xác định mục tiêu nghiên cứu chính là nghiên cứu phát triển các mô hình phát hiện mã độc IoT dựa trên học máy nhằm nâng cao hiệu quả về mặt số lượng đặc trưng, thời gian thu thập, kích thước, độ chính xác của mô hình trong phát hiện mã độc IoT hoạt động

đơn và đa nền tảng kiến trúc vi xử lý nhằm tăng khả năng tích hợp mô hình trong các giải pháp phát hiện mã độc tự động trong hệ thống IoT tài nguyên hạn chế.

3. Đối tượng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu là các tập tin thực thi ELF trên các thiết bị IoT sử dụng hệ điều hành Linux nhúng được thu thập từ các nguồn uy tín.

3.2. Phạm vi nghiên cứu

Thứ nhất, luận án tập trung mô hình học máy có khả năng phân biệt tập tin thực thi hành hai nhãn lành tính hoặc mã độc.

Thứ hai, luận án chỉ tập trung phát hiện mã độc trên thiết bị IoT tài nguyên hạn chế sử dụng hệ điều hành Linux Kernel 2.6 hoặc 3.2 với các kiến trúc bộ vi xử lý phổ biến gồm Intel, MIPS, ARM, SPARC, PowerPC.

Thứ ba, luận án lựa tiếp cận theo phương pháp phân tích động và phân tích tĩnh tập tin nhằm thu thập các đặc trưng dạng chuỗi phục vụ xây dựng các mô hình phát hiện mã độc IoT dựa trên học máy.

Thứ tư, luận án tiếp cận theo kiến trúc vi xử lý trung tâm để phân biệt mã độc đa nền tảng trên thiết bị IoT gồm mã độc IoT đơn kiến trúc vi xử lý và mã độc IoT đa kiến trúc vi xử lý.

4. Các đóng góp chính của luận án

Các đóng góp chính của luận án này bao gồm:

- *Đóng góp 1*: Luận án phát triển mô hình phát hiện mã độc IoT đơn kiến trúc vi xử lý sử dụng đặc trưng chuỗi lời gọi hệ thống ngắn thu thập từ phân tích động tập tin thực thi nhưng vẫn đảm bảo tiêu chí độ chính xác tương đồng các nghiên cứu liên quan dựa trên đề xuất huấn luyện các mô hình dựa đoán chuỗi phục vụ phân lớp tập tin thực thi sử dụng mạng học sâu LSTM. Mô hình phát hiện mã độc IoT sử dụng một loại đặc trưng dạng chuỗi duy nhất là chuỗi lời gọi hệ thống với độ dài chuỗi cần thu thập là 150 góp phần làm giảm thời gian thu thập đặc trưng trong phát hiện mã độc IoT và là cơ sở để xây dựng các giải pháp phát hiện mã độc IoT tích hợp trong các hệ thống bảo mật đáp ứng thời gian thực.

- *Đóng góp 2:* Luận án phát triển mô hình phát hiện mã độc IoT đơn kiến trúc vi xử lý sử dụng tối thiểu số lượng mã thực thi để thu thập chuỗi mã thực thi từ phân tích tĩnh tập tin thực thi nhưng vẫn đảm bảo tiêu chí độ chính xác tương đồng các nghiên cứu liên quan dựa trên đề xuất sử dụng phương pháp trích rút đặc trưng hiệu quả. Mô hình phát hiện mã độc IoT chỉ sử dụng số lượng 20 mã thực thi hiệu quả nhất trên nền tảng MIPS phục vụ xây dựng giải pháp trích chọn chuỗi mã thực thi đại diện cho tập tin thực thi góp phần làm giảm thời gian thu thập đặc trưng,

giảm tài nguyên sử dụng của mô hình phát hiện, có thể phù hợp với các thuật toán học máy truyền thống đơn giản và là cơ sở để xây dựng các giải pháp phát hiện mã độc IoT tích hợp trong các hệ thống bảo mật IoT tài nguyên hạn chế.

- *Đóng góp 3:* Luận án phát triển mô hình phát hiện mã độc IoT đa kiến trúc hiệu quả về độ chính xác, có khả năng phát hiện mã độc trên kiến trúc vi xử lý mới hoặc ít dữ liệu và tri thức về mã độc dựa trên đề xuất phương pháp chuyển đổi đặc trưng chuỗi mã thực thi chéo kiến trúc vi xử lý khác nhau. Mô hình phát hiện mã độc IoT đa kiến trúc vi xử lý đã đạt hiệu quả về độ chính xác tốt khi tăng cường được tập dữ liệu huấn luyện và chỉ sử dụng các thuật toán học máy truyền thống đơn giản, kích thước, thời gian phát hiện phù hợp khi triển khai trong môi trường IoT tài nguyên hạn chế, có khả năng dự đoán mã độc zero-day đa kiến trúc trên các thiết bị IoT đa dạng.

CHƯƠNG 1: TỔNG QUAN VỀ MÃ ĐỘC IOT VÀ PHÁT HIỆN MÃ ĐỘC IOT DỰA TRÊN HỌC MÁY

1.1. Tổng quan về mã độc IoT

1.1.1. Khái niệm, đặc điểm của thiết bị IoT

Với các khái niệm NCS đã khảo sát, NCS sử dụng khái niệm thiết bị IoT tổng quát trong Định nghĩa 1.1. Một mô hình ứng dụng các thiết bị IoT được minh hoạ qua hình 1.1. Trong đó, phần lớn các thiết bị IoT sử dụng hệ điều hành Linux . Với khả năng kết nối và hoạt động đa nền tảng, thiết bị IoT có những đặc trưng khác biệt với những công nghệ khác.

1.1.2. Khái niệm, đặc điểm của mã độc IoT

Tương tự cách tiếp cận như mã độc trên thiết bị di động và mã độc Linux thì khái niệm mã độc IoT được sử dụng trong luận án được xác định trong Định nghĩa 1.2. Ngoài những đặc trưng chung của mã độc thì mã độc IoT có một số đặc trưng riêng biệt khác. Khái niệm mã độc đơn kiến trúc và mã độc đa kiến trúc được tổng quát trong Định nghĩa 1.3 và Định nghĩa 1.4. Mã độc zero-day đa kiến trúc trên thiết bị IoT được tổng quát trong Định nghĩa 1.5.

1.1.3. Phân loại mã độc IoT

Tương tự đối với mã độc nói chung, mã độc IoT thường được các nghiên cứu phân loại và gán nhãn dựa trên dựa trên hoạt động của mã độc bao gồm các loại chính như Botnet, Worm, Trojan, Spyware, Rootkit, Virus,…

1.1.4. Xu hướng phát triển của mã độc IoT

Trong 10 năm trở lại đây, một số dòng mã độc IoT và các biến thể điển hình được trình bày trong bảng 1.2. Các mã độc trên thiết bị IoT thường được phát triển và tạo ra nhiều biến thể để hoạt động trên một kiến trúc vi xử lý định sẵn

hoặc có khả năng hoạt động trên nhiều kiến trúc vi xử lý khác nhau.

1.2. Tổng quan về phát hiện mã độc IoT dựa trên học máy

1.2.1. Quy trình phân tích phục vụ phát hiện mã độc IoT

Với bài toán phát hiện mã độc thường được các chuyên gia phân tích thực hiện thông qua quá trình phân tích tập tin thực thi gồm 6 bước mô tả trong hình 1.2. Trong đó, 2 bước phân tích động và phân tích tĩnh là rất cần thiết có thể tìm hiểu, đánh giá đầy đủ hành vi, hoạt động của một tập tin trong thực tế nhưng lại đòi hỏi nhiều kiến thức chuyên sâu của người phân tích. Khi số lượng mẫu tập thực thi thu thập ngày càng nhiều và phức tạp, việc thực hiện các bước phân tích sơ lược, phân tích động và phân tích tĩnh để xác định hành vi độc hại của tập tin là không đáp ứng được yêu cầu thực tiễn. Vì vậy, ứng dụng học máy trong xây dựng các mô hình phát hiện mã độc IoT là cần thiết trong việc xác định hành vi độc hại của một tập tin thực thi. Các mô hình phát hiện mã độc IoT dựa trên học máy có thể thay thế các chuyên gia phân tích mã độc để phát hiện và phân loại mã độc trong tập tin thực thi. Việc áp dụng học máy trong xây dựng mô hình phát hiện mã độc IoT không chỉ giúp tự động hóa và tăng tốc quá trình phát hiện mà còn nâng cao hiệu quả và khả năng phát hiện các mẫu mã độc mới và phức tạp.

1.2.2. Mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng của tập tin thực thi

Mô hình phát hiện mã độc IoT dựa trên học máy là một mô hình được xây dựng dựa trên sử dụng các kỹ thuật và phương pháp học máy để phát hiện mã độc từ tập dữ liệu đầu vào. Mô hình là sự kết hợp giữa việc sử dụng các kỹ thuật học máy với các đặc trưng đã thu thập để xây dựng các bộ phân lớp tập tin hoặc dự đoán mã độc. Quá trình xây dựng một mô hình phát hiện mã độc IoT dựa trên học máy thường bao gồm 6 bước. Các bước quan trọng trong quá trình xây dựng mô hình phát hiện mã độc dựa trên học máy cần tập trung giải quyết bao gồm: Trích xuất, lựa chọn được các đặc trưng hiệu quả và lựa chọn mô hình huấn luyện phù hợp để giải quyết các vấn đề khác nhau trong phát hiện mã độc IoT.

1.3. Đánh giá hiệu quả của một số mô hình phát hiện mã độc bị IoT dựa trên học máy sử dụng đặc trưng của tập tin thực thi

1.3.1. Đánh giá hiệu quả của mô hình phát hiện mã độc IoT đơn kiến trúc dựa trên học máy sử dụng đặc trưng của tập tin thực thi

Đối với đặc trưng của tập tin trích xuất từ phân tích động, một số mô hình phát hiện mã độc IoT đơn kiến trúc dựa trên học máy sử dụng đặc trưng động của tập tin thực thi thể hiện như bảng 1.6. Các mô hình này đã đem lại độ chính xác tốt hơn sử dụng các đặc trưng động khác như lưu lượng mạng và tài nguyên máy chủ.

Tuy nhiên, một số nghiên cứu cần thu thập nhiều thông tin hành vi từ phân tích động, quá trình thu thập đặc trưng lời gọi hệ thống cho từng mẫu lớn. Các mô hình học máy, mạng học sâu đã được kết hợp để nâng cao độ chính xác nhưng các nghiên cứu đối mặt với độ phức tạp tính toán lớn khi sử dụng các mạng học sâu phức tạp và cần huấn luyện nhiều mô hình phân lớp khác nhau để phát hiện mã độc. Vì vậy, khi triển khai mô hình phát hiện đã được huấn luyện trong các giải pháp bảo mật trên môi trường IoT tài nguyên hạn chế hoặc cần đảm bảo yêu cầu phát hiện, cảnh báo sớm mã độc cho nhiều tập tin thực thi trong thời gian ngắn sẽ gặp nhiều khó khăn. Mặt khác, một số tập dữ liệu được các nghiên cứu thử nghiệm có sự chênh lệch và hạn chế về số mẫu tập tin lành tính thu thập được.

Đối với đặc trưng của tập tin trích xuất từ phân tích tĩnh, một số mô hình phát hiện mã độc IoT hiệu quả dựa trên học máy sử dụng đặc trưng tĩnh của tập tin thực thi thể hiện như bảng 1.8. Phương pháp phát hiện mã độc dựa trên học máy sử dụng đặc trưng tĩnh cho độ chính xác cao và độ tin cậy tốt khi phát hiện các mã độc IoT có đặc trưng tĩnh tương tự nhau. Tuy nhiên, các nghiên cứu sử dụng phương pháp này đã được chỉ ra các hạn chế trong phát hiện mã độc có sử dụng các kỹ thuật che giấu hoặc các mã độc IoT có tính chất động. Bên cạnh đó, các phương pháp đã đề xuất có độ phức tạp cao trong trích xuất đặc trưng dạng đồ thị và sử dụng mạng học sâu kiến trúc phức tạp, chưa có các đánh giá về mặt tài nguyên sử dụng trong huấn luyện mô hình học máy và mô hình phát hiện sau khi huấn luyện. Một số nghiên cứu thử nghiệm với tập dữ liệu IoT chưa đủ lớn và chỉ tập trung vào nền tảng kiến trúc vi xử lý của thiết bị di động như ARM.

Trong xây dựng mô hình phát hiện mã độc IoT đơn kiến trúc hiệu quả dựa trên đặc trưng của tập tin thu thập từ phân tích động, phân tích tĩnh bên cạnh việc đảm bảo yêu cầu về độ chính xác thì cần xem xét các yêu cầu về thời gian trích xuất đặc trưng, tiền xử lý, kích thước và thời gian phát hiện của mô hình. Việc sử dụng kết hợp đặc trưng của tập tin thực thi trích xuất từ phân tích động và phân tích tĩnh trong xây dựng các mô hình phát hiện mã độc IoT đơn kiến trúc hiệu quả có thể hạn chế được các trường hợp mã độc sử dụng các kỹ thuật lẩn tránh phân tích và nâng cao khả năng phân lớp chính xác tập tin thực thi. Tuy nhiên, việc sử dụng kết hợp các đặc trưng động và đặc trưng tĩnh có thể làm tăng độ phức tạp của mô hình, thời gian trích xuất, thu thập đặc trưng và thời gian phát hiện mã độc của mô hình.

1.3.2. Đánh giá hiệu quả của mô hình phát hiện mã độc IoT đa kiến trúc dựa trên học máy sử dụng đặc trưng của tập tin thực thi

Một số phương pháp xây dựng mô hình phát hiện mã độc IoT đa kiến trúc dựa

trên học máy bao gồm: Dựa trên dữ liệu đa nguồn, dựa trên kỹ thuật trích chọn đặc trưng đa kiến trúc, dựa trên kết hợp nhiều mô hình đơn kiến trúc, dựa trên mô hình phát hiện IoT chéo kiến trúc,… Các phương pháp trên trong phát hiện mã độc IoT đa kiến trúc dựa trên học máy đã chứng minh hiệu quả bước đầu. Tuy nhiên, các phương pháp, mô hình luận án đã khảo sát trong bảng 1.10 chưa cho hiệu quả cao về độ chính xác phù hợp với tài nguyên tính toán và chưa đề cập đến khả năng phát hiện, dự báo mã độc trên kiến trúc vi xử lý mới.

Bên cạnh đó, các nghiên cứu phát hiện mã độc IoT đa kiến trúc đã thu thập và sử dụng tập dữ liệu thử nghiệm mô tả trong bảng 1.11. Các mô hình phát hiện mã độc IoT đa kiến trúc đã khảo sát thời đang gặp khó khăn về tập dữ liệu thử nghiệm và độ chính xác của mô hình phát hiện. Các tập dữ liệu mã độc thử nghiệm chủ yếu được các nhóm tác giả thu thập từ các nguồn Internet như IoTPOT, VirusShare, VirusTotal ,… Số mẫu huấn luyện có các nhãn và trên các kiến trúc vi xử lý khác nhau có sự chênh lệch lớn. Điều này có thể dẫn đến vấn đề mất cân bằng dữ liệu trong huấn luyện mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng của tập tin thực thi. Mặt khác, với sự phát triển đa dạng, nhanh chóng của các loại thiết bị IoT hiện nay về hệ điều hành và nền tảng kiến trúc vi xử lý sử dụng, việc tận dụng tri thức từ các tập dữ liệu đã có trước đây hoặc tạo ra dữ liệu mới từ dữ liệu ban đầu sẽ có thể mang lại nhiều hiệu quả trong phát hiện mã độc IoT đa nền tảng kiến trúc trong tương lai.

1.4. Bài toán nâng cao hiệu quả cho mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng của tập tin thực thi

1.4.1. Bài toán nghiên cứu

Với các đánh giá hiệu quả của một số mô hình phát hiện mã độc IoT dựa trên học máy đã trình bày ở trên, mỗi mô hình phát hiện mã độc IoT dựa trên học máy đã khảo sát đều có những ưu và nhược điểm khác nhau. Để giải quyết 3 hạn chế đã phân tích trong xây dựng mô hình phát hiện mã độc IoT hiệu quả dựa trên học máy và đặc trưng của tập tin thực thi, luận án nghiên cứu bài toán như sau:

Nghiên cứu xây dựng các mô hình phát hiện mã độc trên thiết bị IoT đa dạng kiến trúc vi xử lý dựa trên học máy sử dụng ít thời gian và số lượng đặc trưng biểu diễn dạng chuỗi của tập tin thực thi cần thu thập từ phân tích động và phân tích tĩnh nhằm nâng cao khả năng tích hợp trong hệ thống IoT tài nguyên hạn chế, đảm bảo khả năng áp dụng cho tất cả các tập tin thực thi trong môi trường IoT với độ chính xác tốt và có thể dự báo mã độc IoT zero-day chéo kiến trúc vi xử lý.

1.4.2. Giải quyết bài toán

Để giải quyết bài toán, luận án xây dựng các mô hình phát hiện mã độc IoT

đơn kiến trúc và đa kiến trúc hiệu quả dựa trên học máy sử dụng đặc trưng biểu diễn dạng chuỗi của tập tin thực thi thu tập thông qua các phương pháp phân tích động, phân tích tĩnh. Trong đó, tập trung nâng cao hiệu quả của các mô hình phát hiện dựa trên nghiên cứu các phương pháp thu thập, lựa chọn đặc trưng động và đặc trưng tĩnh, xử lý dữ liệu đa nền tảng kiến trúc vi xử lý và lựa chọn thuật toán học máy phù hợp với môi trường IoT. Thông qua việc khảo sát, đánh giá, phân tích các mô hình phát hiện mã độc IoT trong luận án, việc nâng cao hiệu quả cho mô hình phát hiện mã độc IoT dựa trên học máy được thực hiện dựa trên các nội dung sau đây:

- Thu thập và lựa chọn đặc trưng hiệu quả trong phát hiện mã độc IoT.
- Lựa chọn thuật toán học máy hiệu quả trong phát hiện mã độc IoT.
- Tăng cường dữ liệu trong huấn luyện mô hình phát hiện mã độc IoT.

1.5. Kết luận chương 1

Chương 1 đã trình bày những nội dung cơ bản về thiết bị IoT và mã độc IoT với các xu hướng phát triển của mã độc trên các thiết bị IoT. Đánh giá, phân tích quy trình phân tích và cách thức xây dựng mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng của tập tin thực thi. Phân tích, đánh giá một số mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng của tập tin thực thi hiệu quả. Từ đó, cho thấy việc nghiên cứu các mô hình phát hiện mã độc trên thiết bị IoT hiệu quả dựa trên học máy và đặc trưng của tập tin thực thi còn nhiều hạn chế và cần được cải tiến. Vì vậy, đề tài luận án tập trung vào nghiên cứu nâng cao hiệu quả cho các mô hình phát hiện mã độc trên thiết bị IoT tài nguyên hạn chế dựa trên học máy sử dụng đặc trưng của tập tin thực thi thu thập thông qua phương pháp phân tích động và phân tích tĩnh.

Từ nội dung đã phân tích, luận án xác định các hướng tiếp cận nâng cao hiệu quả của mô hình phát hiện mã độc IoT tập trung vào phương pháp thu thập, lựa chọn đặc trưng, tăng cường tập dữ liệu và sử dụng phương pháp học máy phù hợp, hiệu quả với các đặc trưng của tập tin thực thi thu thập thông qua phân tích động và phân tích tĩnh. Từ đó đề xuất bài toán nghiên cứu của luận án và giải quyết bài toán thông qua các mô hình phát hiện mã độc IoT đơn kiến trúc và đa kiến trúc được trình bày trong các chương tiếp theo của luận án.

CHƯƠNG 2: XÂY DỰNG MÔ HÌNH PHÁT HIỆN MÃ ĐỘC IOT ĐƠN KIẾN TRÚC HIỆU QUẢ SỬ DỤNG ĐẶC TRƯNG ĐỘNG CỦA TẬP TIN THỰC THI

2.1. Mở đầu

Chương 2 sẽ trình bày mô hình phát hiện mã độc IoT đơn kiến trúc hiệu quả

với việc thu thập chuỗi lời gọi hệ thống ngắn thu thập từ phân tích động nhưng vẫn đảm bảo các tiêu chí về độ chính xác của mô hình dựa trên phương pháp Bagging nhưng sử dụng cùng một kiến trúc mạng nơ-ron LSTM để huấn luyện trên hai tập dữ liệu đã được gán nhãn mã độc và lành tính.

Mô hình đề xuất trong chương này giải quyết vấn đề sau đây:

- Cho M = {M_1, M_2, ..., M_n} với M_i là mã độc (i = 1,2,...,m). M-Model là mô hình dự đoán chuỗi lời gọi hệ thống xây dựng từ tập M.

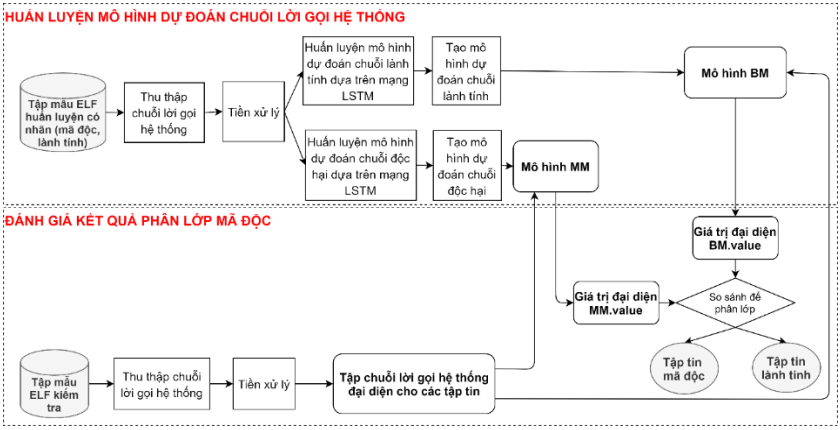
- Cho B = {B_1, B_2, ..., B_k} với B_j là lành tính (j = 1,2,...,n). B-Model là mô hình dự đoán chuỗi lời gọi hệ thống xây dựng từ tập B.

- Cho S_1^L = {s_1, s_2, ..., s_L} là một chuỗi lời gọi hệ thống thu thập từ tập tin có độ dài L, mỗi lời gọi hệ thống s_t (t = 1,2,...,L) trong tập S_1^L sẽ xác định xác suất xuất hiện tương ứng của chuỗi lời gọi hệ thống S_1^t đối với M-Model và B-Model lần lượt là $P_{M-Model}(S_1^t)$ và $P_{B-Mode}(S_1^t)$.

- Cần tìm giá trị k = $arg\,min_k$ $|P_{M-Model}(S_1^t)$ - $P_{B-Model}(S_1^t)|$ và độ chính xác được đảm bảo khi so sánh với các mô hình khác có liên quan.

2.2. Xây dựng mô hình phát hiện mã độc IoT đề xuất dựa trên chuỗi lời gọi hệ thống

Quá trình xây dựng mô hình phát hiện mã độc IoT đơn kiến trúc dựa trên đặc trưng chuỗi lời gọi hệ thống thu thập từ phương pháp phân tích động tập tin thực thi đề xuất gồm 2 giai đoạn chính: Huấn luyện mô hình dự đoán chuỗi lời gọi hệ thống và đánh giá kết quả phân lớp mã độc dựa trên hai mô hình dự đoán chuỗi lời gọi hệ thống đã huấn luyện.



2.2.1. Thu thập chuỗi lời gọi hệ thống

Để có được chuỗi lời gọi hệ thống hoàn chỉnh nhất, các vectơ đặc trưng được xây dựng bởi tất cả các lời gọi hệ thống. F-Sandbox được sử dụng để trích xuất nhật ký lời gọi hệ thống của các tập tin thực thi ELF. Cấu trúc của F-Sandbox gồm 4 thành phần chính được thể hiện trong hình 2.3.

2.2.2. Tiền xử lý dữ liệu

Quá trình chuyển đổi các chuỗi lời gọi hệ thống thành các vector phục vụ quá trình huấn luyện các mô hình dự đoán chuỗi lời gọi hệ thống được mô tả trong thuật toán 2.1. Việc chuyển đổi các chuỗi lời gọi hệ thống được tiến hành trên từng tập dữ liệu Mal-SysCallLog và Beg-SysCallLog.

Thuật toán 2.1. *Tiền xử lý chuỗi lời gọi hệ thống*

Đầu vào*: Một chuỗi lời gọi hệ thống (syscall) với ngưỡng độ dài L*
Đầu ra: *Hai vector sau khi biến đổi chuỗi lời gọi hệ thống*

```
# Khởi tạo danh sách x và y
1: x ← []
2: y ← []
3: For i in [0, min (L, length(syscall))]
        # Tách chuỗi syscall thành các chuỗi con có độ dài từ 0 tới i
4:      Slice_syscall ← substring(systemcall,0,i)
        # Thêm đệm là giá trị "0" thông qua padding vào đầu chuỗi con vừa tách
để đảm bảo độ dài các chuỗi là L
5:      x_padded ← padding (slice_syscall, max_length=L)
        # Mã hoá x_padded và y_padded thành ma trận one-hot encoding với độ
dài bằng kích thước từ điển lời gọi hệ thống trên kiến trúc tương ứng (vocab_size)
6:      x_onehot ← onehot (x_padded, num_class = vocab_size)
7:      y_onehot ← onehot (syscall[i], num_class = vocab_size)
        # Thêm dữ liệu x_onehot, y_onehot vào danh sách x và y
8:      x.append (x_onehot)
9:      y.append (y_onehot)
10: Endfor
11: Return x, y
```

2.2.3. Huấn luyện mô hình dự đoán chuỗi lời gọi hệ thống dựa trên mạng nơ-ron LSTM

Quá trình xây dựng hai mô hình dự đoán chuỗi lời gọi hệ thống MM và BM dựa trên một kiến trúc mạng học sâu LSTM được thể hiện trong đoạn giả mã của thuật toán 2.2.

Thuật toán 2.2. *Xây dựng mô hình dự đoán chuỗi lời gọi hệ thống dựa trên mạng nơ-ron LSTM*

Đầu vào: *Chuỗi tất cả các lời gọi của tập dữ liệu (X)*
Đầu ra*: Mô hình dự đoán chuỗi lời gọi hệ thống (model)*

```
1: Model ← Khởi tạo mô hình dự đoán dựa trên mạng học sâu LSTM
# Khởi tạo hàm mất mát sử dụng cross entropy
```

2: Criterion ← Cross-Entropy Loss
 # Khởi tạo hàm tối ưu hoá đối với các tham số trong mô hình
3: Optimizer () ← Adam (Parameter of Model)
4: **For** $\boldsymbol{s_i}$ **in** X:
 # Đặt lại gradient của các tham số trong thuật toán tối ưu hoá
5: Optimizer zero gradient ()
6: P_{h_i} ← model ($\boldsymbol{s_i}$) = p($s_i|S_1^{i-1}$)
 # Giá trị dự đoán $P_i = max\ (P_{h_i})$
7: Predicted_syscall=($s_i|\boldsymbol{P_i}$,1≤i≤n)
8: Loss ← criterion (predicted_syscall, $\boldsymbol{s_i}$)
 # Lan truyền ngược thông qua hàm Loss Backward
9: Loss Backward ()
 # Cập nhật trọng số cho mô hình thông qua hàm Optimizer
10: Optimizer ()
11: **Endfor**
12: **Return** Model

2.2.4. Xây dựng bộ phân lớp chuỗi lời gọi hệ thống dựa trên các mô hình dự đoán chuỗi đã huấn luyện

Quá trình phân lớp chuỗi lời gọi hệ thống thu thập từ một tập tin thực thi dựa trên 2 mô hình dự đoán chuỗi sử dụng mạng nơ-ron LSTM được trình bày bằng giả mã như Thuật toán 2.3.

Thuật toán 2.3. Phát hiện mã độc IoT dựa trên hai mô hình dự đoán chuỗi lời gọi hệ thống sử dụng mạng nơ-ron LSTM

Đầu vào: *Chuỗi lời gọi hệ thống cần xác định nhãn* S_1^L*; 2 mô hình dự đoán chuỗi lời gọi hệ thống MM và BM.*
Đầu ra*: Nhãn của tập tin sinh ra chuỗi* S_1^L.

 # Tiền xử lý chuỗi lời gọi hệ thống S_1^L
1: Preprocess_syscall (S_1^L)
 # Xác định giá trị đại diện của tập tin dựa trên mô hình MM
2: p(S_1^L) ← $MM(S_1^L)$
3: Representative _value_mal ← exp($\frac{\sum_{i=1}^{N} log\ p_i}{N}$)
 # Xác định giá trị đại diện của tập tin dựa trên mô hình BM
4: p(S_1^L) ← $BM(S_1^L)$
5: Representative _value_beg ← exp($\frac{\sum_{i=1}^{N} log\ p_i}{N}$)
 # So sánh hai giá trị đại diện của tập tin để quyết định nhãn
6: **If** Representative _value_mal > Representative _value_beg **then**
7: $Malware \leftarrow Nhãn\ (S_1^L)$
8: **Else**
9: $Benign \leftarrow Nhãn\ (S_1^L)$
10: **Endif**
11: **Return** $Nhãn\ (S_1^L)$

2.3. Triển khai thử nghiệm

2.3.1. Môi trường thử nghiệm

Các thử nghiệm thu thập các tập dữ liệu chuỗi lời gọi hệ thống được cài đặt và kết nối theo hình 2.9. Thử nghiệm huấn luyện, đánh giá mô hình phát hiện mã độc IoT dựa trên mạng nơ-ron LSTM và chuỗi lời gọi hệ thống dựa trên thư viện Tensorflow và sử dụng GPU trên Google Colaboratory để tăng tốc huấn luyện.

2.3.2. Xây dựng tập dữ liệu thử nghiệm

Thực nghiệm sử dụng tập dữ liệu tập tin thực thi trên môi trường IoT được xây dựng từ một bộ dữ liệu C500-IoT dataset công bố bởi Phú và các cộng sự, chi tiết về tập dữ liệu thể hiện trong bảng 2.2. Ngoài ra, luận án đã tiến hành thu thập bổ sung thêm vào tập dữ liệu thử nghiệm các chương trình lành tính từ các chương trình tiện ích của các nhà cung cấp phần mềm tin cậy và đã được đánh giá bởi công cụ quét mã độc trực tuyến. Tập dữ liệu thử nghiệm trong chương 2 của luận án bao gồm 300 tập tin lành tính và 930 tập tin mã độc. Trong đó có 37 họ mã độc khác khau với nhiều họ mã độc IoT phổ biến.

2.3.3. Xây dựng các tập chuỗi lời gọi hệ thống

Chuỗi lời gọi hệ thống của một tập tin được thu thập qua 2 bước và kết quả thu thập chuỗi lời gọi hệ thống từ các tập tin ELF đã được gán nhãn thể hiện như bảng 2.3. Kết quả các tập chuỗi lời gọi hệ thống theo các ngưỡng độ dài chuỗi được thể hiện trong bảng 2.4.

2.3.4. Kết quả xây dựng mô hình phát hiện mã độc IoT

Luận án tiến hành thử nghiệm mô hình phát hiện mã độc IoT dựa trên cùng một trên kiến trúc mạng nơ-ron LSTM cho hai tập dữ liệu đã được gán nhãn nhưng khác nhau về các tham số như trong bảng 2.5. Luận án đánh giá hiệu suất của mô hình dựa trên độ dài của chuỗi lời gọi hệ thống, số lượng lớp ẩn và số lượng đơn vị ẩn trong mỗi lớp. Từ đó, xác định độ dài chuỗi lời gọi hệ thống, số lớp ẩn và số đơn vị ẩn trong mỗi lớp cho hiệu quả đối với tập dữ liệu thử nghiệm trên kiến trúc MIPS.

- Kết quả đánh giá hiệu suất của mô hình dựa trên độ dài của chuỗi lời gọi hệ thống được thể hiện trong Bảng 2.7.

- Kết quả đánh giá hiệu suất của mô hình dựa trên số lớp ẩn và số lượng nơ-ron trong mỗi lớp được thể hiện trong Bảng 2.8 và Bảng 2.9.

Như vậy, từ các kết quả thử nghiệm, luận án thấy rằng mô hình phát hiện sử dụng mạng LSTM với 4 lớp ẩn, 1000 unit và chuỗi lời gọi hệ thống có ngưỡng độ dài là 150 khi xây dựng mô hình MM và BM sẽ cho kết quả phát hiện mã độc IoT trên kiến trúc MIPS hiệu quả nhất về các độ đo độ chính xác.

2.3.5. So sánh hiệu quả của mô hình đề xuất với các mô hình khác có liên quan

- So sánh hiệu quả mô hình đề xuất với mô hình phát hiện mã độc IoT sử dụng một mạng nơ-ron LSTM duy nhất: Để thực hiện so sánh, mô hình phát hiện mã độc IoT trên kiến trúc MIPS sử dụng đặc trưng chuỗi lời gọi hệ thống được huấn luyện dựa trên duy nhất 1 mạng nơ-ron LSTM như trong các nghiên cứu có liên quan khác được thử nghiệm trên cùng một bộ dữ liệu, cùng thiết bị và môi trường thực nghiệm như mô hình NCS đã sử dụng. Kết quả so sánh được thể hiện trong Bảng 2.10. Việc sử dụng 2 mạng LSTM trong xây dựng mô hình phát hiện đã chứng minh khả năng tìm hiểu và xử lý dữ liệu đặc trưng dạng chuỗi trích xuất từ phân tích động tập tin IoT đa dạng hơn so với việc chỉ áp dụng một mạng LSTM duy nhất.

- So sánh với các mô hình phát hiện mã độc IoT sử dụng học máy và phương trích chọn đặc trưng n-gram: Để thực hiện so sánh, ba mô hình phát hiện sử dụng học máy và phương pháp trích chọn đặc trưng n-gram được sử dụng trong các nghiên cứu phát hiện mã độc IoT như các nghiên cứu khác tiến hành thực nghiệm để đánh giá. Ba mô hình so sánh là 3 mô hình phát hiện mã độc IoT trên kiến trúc MIPS điển hình sử dụng chuỗi lời gọi hệ thống được sử dụng trong luận án của Phú. Các đánh giá, so sánh hiệu quả các mô hình được tiến hành thử nghiệm trên cùng một bộ dữ liệu, cùng thiết bị và môi trường thực nghiệm. Kết quả so sánh được thể hiện trong Bảng 2.11. Kết quả so sánh cho thấy rằng mô hình phát hiện mã độc IoT trên kiến trúc MIPS mà luận án đề xuất hiệu quả hơn về các độ đo độ chính xác và độ dài chuỗi lời gọi hệ thống cần thu thập so với 3 mô hình phát hiện mã độc IoT khác sử dụng học máy và phương pháp n-gram.

- So sánh hiệu quả với các mô hình phát hiện sử dụng chuỗi lời gọi hệ thống ngắn khác: Mô hình đề xuất chỉ cần sử dụng duy nhất một loại đặc trưng thu thập từ quá trình phân tích động tập tin là chuỗi lời gọi hệ thống với độ dài chuỗi cần thu thập ngắn là 150 đã có thể cho độ chính xác tương đồng với một số nghiên cứu khác luận án đã khảo sát. So sánh hiệu quả với các nghiên cứu khác có liên quan được thể hiện trong Bảng 2.12. Kết quả cho thấy mô hình đã khai thác tốt đặc trưng tuần tự của từng tập dữ liệu chuỗi lời gọi hệ thống thu thập từ tập tin mã độc và tập tin lành tính. Do đó, mô hình đề xuất có thể được sử dụng để tích hợp mô hình phát hiện vào các giải pháp phát hiện mã độc cần ít thông tin và thời gian thu thập từ thực thi tập tin nhưng vẫn đảm bảo độ chính xác cao trong các hệ thống bảo mật thực tế.

2.5. Kết luận chương 2

Chương 2 đã đề xuất mô hình phát hiện mã độc IoT dựa trên phương pháp

Bagging sử dụng cùng một kiến trúc mạng nơ-ron LSTM cho từng tập dữ liệu chuỗi lời gọi hệ thống trích xuất thông qua phân tích động đã được gán nhãn. Mô hình đề xuất sử dụng chuỗi lời gọi hệ thống ngắn hơn so với các mô hình phát hiện mã độc IoT khác đã khảo sát nhưng vẫn đạt hiệu quả cao về tiêu chí độ chính xác, ít bị tác động bởi vấn đề mất cân bằng dữ liệu huấn luyện giữa các nhãn trong phát hiện mã độc IoT trên kiến trúc MIPS. Do đó, mô hình có thể được huấn luyện, tạo mô hình phát hiện để sử dụng tích hợp trong các hệ thống phát hiện và cảnh báo mã độc IoT trong thực tế.

Tuy nhiên, trong các thử nghiệm của luận án đã chứng minh phương pháp phân tích động chưa thu thập được toàn bộ đặc trưng lời gọi hệ thống của tất cả các tập tin khi thực thi, một số tập tin độc hại chưa thực thi trong môi trường F-sandbox. Việc sử dụng chuỗi lời gọi hệ thống ngắn đã đem lại hiệu quả trong các thử nghiệm nhưng có thể gặp khó khăn khi phát hiện các mã độc sử dụng các kỹ thuật che giấu hành vi, chưa bộc lộ hành vi ngay sau khi thực thi tập tin. Vì vậy, để xây dựng giải pháp phát hiện mã độc IoT tổng thể theo quy trình phân tích mã độc, luận án sẽ nghiên cứu đề xuất mô hình phát hiện mã độc IoT hiệu quả dựa trên học máy sử dụng đặc trưng thu thập từ phân tích tĩnh tập tin.

Ý tưởng và kết quả thực nghiệm của mô hình đề xuất trong chương này đã được trình bày, công bố trên tạp chí quốc tế [TC3].

CHƯƠNG 3: XÂY DỰNG MÔ HÌNH PHÁT HIỆN MÃ ĐỘC IOT ĐƠN KIẾN TRÚC HIỆU QUẢ SỬ DỤNG ĐẶC TRƯNG TĨNH CỦA TẬP TIN THỰC THI

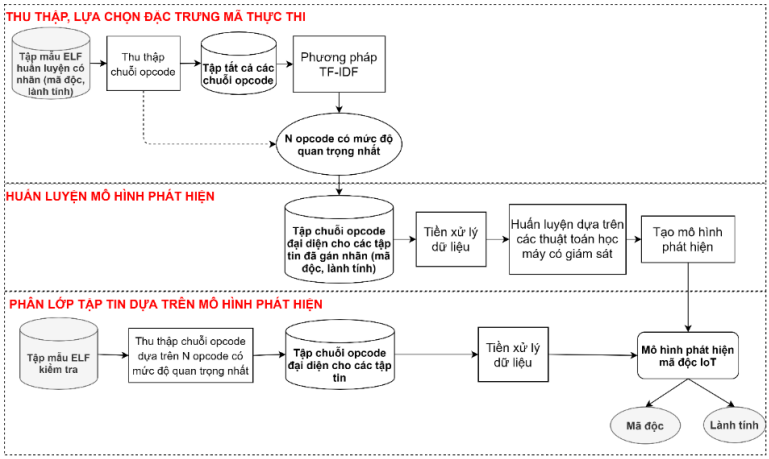
3.1. Mở đầu

Để nâng cao hiệu quả cho mô hình phát hiện mã độc IoT đơn kiến trúc sử dụng đặc trưng tĩnh của tập tin thực thi, chương 3 đề xuất kết hợp phương pháp trích xuất, thu thập đặc trưng chuỗi mã thực thi hiệu quả, phù hợp với các mô hình học máy truyền thống trong xây dựng mô hình phát hiện mã độc IoT trên nền tảng kiến trúc MIPS. Mô hình phát hiện mã độc IoT đơn kiến trúc sử dụng tối thiểu số lượng mã thực thi cần thu thập từ phân tích tĩnh tập tin, sử dụng các thuật toán học máy truyền thống để giảm độ phức tạp tính toán, thời gian tính toán và kích thước mô hình phát hiện sau khi huấn luyện nhưng vẫn đảm bảo độ chính xác cao tương đồng các nghiên cứu khác luận án đã khảo sát.

3.2. Xây dựng mô hình hình phát hiện mã độc IoT dựa trên chuỗi mã thực thi đề xuất

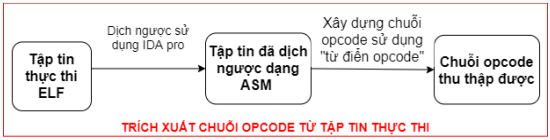
Quá trình xây dựng mô hình phát hiện mã độc IoT dựa trên đặc trưng chuỗi mã thực thi đề xuất gồm 3 giai đoạn chính: Thu thập, lựa chọn đặc trưng; Huấn

luyện mô hình phát hiện; Đánh giá mô hình phát hiện.



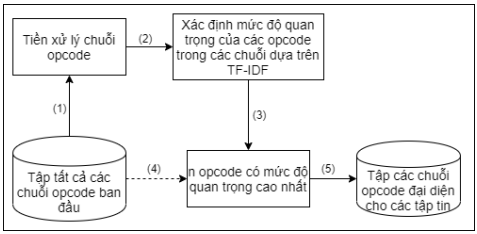
3.2.1. Đề xuất phương pháp trích xuất, lựa chọn đặc trưng mã thực thi

- Đối với quá trình trích xuất chuỗi mã thực thi dựa trên phân tích tĩnh tập tin thực thi: Phương pháp trích xuất một chuỗi mã thực thi từ một tập tin thực thi ELF trên hệ điều hành Linux được thu thập luận án đề xuất sử dụng được thực hiện thông qua các bước chi tiết trong sơ đồ sau:



Chi tiết quá trình thu thập chuỗi mã thực thi được thể hiện trong thuật toán mô tả tại hình 3.6.

- Đề xuất "từ điển opcode" hiệu quả phục vụ xây dựng tập chuỗi mã thực thi đại diện cho các tập tin:: Phương pháp lựa chọn đặc trưng chuỗi mã thực thi đại diện cho các tập tin ELF luận án đề xuất được mô tả trong hình sau:



3.2.2. Huấn luyện mô hình phát hiện mã độc IoT

Các thuật toán học máy có giám sát phổ biến trong phát hiện mã độc được luận án lựa chọn để huấn luyện các mô hình phát hiện mã độc IoT dựa trên đặc trưng chuỗi mã thực thi đã lựa xác định giá trị trong không gian vector. Trong

mô hình phát hiện mã độc IoT đề xuất, chúng tôi lựa chọn thuật toán học máy có giám sát phổ biến trong phát hiện mã độc gồm Support Vector Machines, Random Forest, Naive Bayes để đánh giá hiệu quả.

3.2.3. Phân lớp tập tin dựa trên mô hình phát hiện

Tương tự quá trình huấn luyện mô hình, việc phân lớp các tập tin kiểm tra dựa trên mô hình phát hiện mã độc IoT được tiến hành thông qua thu thập chuỗi mã thực thi dựa trên từ điển opcode đã xây dựng trong giai đoạn thu thập, lựa chọn đặc trưng mã thực thi. Tập các chuỗi mã thực thi đại diện cho các tập tin kiểm tra được tiền xử lý với phương pháp n-gram và phân lớp với các mô hình phát hiện mã độc IoT tương ứng.

3.3. Thử nghiệm và đánh giá hiệu quả của mô hình đề xuất

3.4.1. Môi trường thử nghiệm

Các thử nghiệm được thực hiện trên cùng một máy tính sử dụng hệ điều hành Windows 10 64-bit, chip Intel Core i7-6500U, 2,59 GHz v, RAM 8GB. Để đánh giá hiệu suất, ba mô hình máy học sử dụng phương pháp trích chọn đặc trưng n-gram được tiến hành trên cùng một tập dữ liệu, cùng một hệ thống.

3.4.2. Tập dữ liệu thử nghiệm

Bộ dữ liệu C500-IoT dataset được công bố bởi Phú và các cộng sự như trong nội dung 2.3.2 đã trình bày trong luận án được sử dụng. Để đánh giá các kịch bản thử nghiệm, NCS sử dụng tập dữ liệu mở rộng của tập đã sử dụng trong chương 2 để thử nghiệm hiệu quả của mô hình phát hiện mã độc IoT đơn kiến trúc sử dụng đặc trưng tĩnh của tập tin thực thi gồm 8.904 tập tin ELF trên kiến trúc MIPS (với 4.511 tập tin độc hại và 4.393 tập tin lành tính) trong tập C500-IoT dataset. Tập dữ liệu huấn luyện và kiểm tra được phân chia ngẫu nhiên theo tỉ lệ 70/30 phục vụ đánh giá hiệu quả các mô hình phát hiện.

3.4.3. Kết quả trích xuất và lựa chọn đặc trưng mã thực thi

Kết quả thu thập tập dữ liệu chuỗi mã thực thi thông qua phân tích tĩnh tập tin được thể hiện chi tiết trong Bảng 3.3. Kết quả lựa chọn các mã thực thi trên kiến trúc vi MIPS với mức độ quan trọng hàng đầu (có giá trị TF-IDF cao nhất) được thể hiện trong bảng 3.4.

3.4.4. Kết quả huấn luyện mô hình phát hiện mã độc IoT

Để xây dựng mô hình phát hiện mã độc IoT và mô hình phân loại mã độc IoT trên kiến trúc MIPS sử dụng đặc trưng chuỗi mã thực thi, NCS thử nghiệm mô hình với từng thuật toán học máy đơn lẻ gồm SVM, RF, NB được cài đặt thông qua ngôn ngữ lập trình Python với thư viện Sklearn. Các tham số chính sử dụng trong các thuật toán học máy được mô tả trong Bảng 3.5. Luận án xem xét lựa chọn số lượng opcode có mức độ quan trọng cao nhất theo các kịch bản thử nghiệm gồm 5, 10, 14, 16, 20, 30, 40 kết hợp với ba thuật toán học máy RF,

SVM, NB. Kết quả phát hiện mã độc IoT trên kiến trúc MIPS thể hiện trong Bảng 3.6, Bảng 3.7 và Bảng 3.8. Khi số lượng opcode quan trọng hàng đầu là 20, mô hình RF có độ chính xác cao nhất là 99,8% và F1-Weight là 99,8%. Những kết quả thu được từ có thể được chứng minh bằng thực tế rằng các opcode chưa được chọn lọc chứa nhiều điểm bất thường khác nhau như nhiễu sẽ ảnh hưởng đến hiệu quả của thuật toán máy học trong quá trình huấn luyện mô hình phát hiện mã độc IoT. Việc lựa chọn số lượng nhiều opcode để huấn luyện các mô hình phát hiện cũng làm tăng sự phức tạp tính toán và thời gian huấn luyện của mô hình học máy.

Bên cạnh đó, thời gian phát hiện trung bình các lần thử nghiệm và kích thước của các mô hình phát hiện mã độc IoT dựa trên học máy khi lựa chọn 20 opcode có mức độ quan trọng cao nhất hiện thể hiện trong Bảng 3.9. Từ kết quả thử nghiệm có thể thấy rằng các mô hình áp dụng các phương pháp n-gram với n tăng dần thì thời gian phát hiện và kích thước của mô hình có xu hướng tăng đối với tất cả các thuật toán học máy đã thử nghiệm. So với các nghiên cứu luận án đã khảo sát trong luận án, thời gian và kích thước của mô hình phát hiện khi áp dụng phương pháp 2-gram cho độ chính xác cao, kích thước mô hình và thời gian phát hiện phù hợp với môi trường IoT. Vì vậy, mô hình phát hiện có khả năng tích hợp trong các giải pháp phát hiện mã độc trong môi trường thiết bị IoT tài nguyên hạn chế.

3.3.5. So sánh hiệu quả của mô hình phát hiện mã độc IoT luận án đề xuất với các mô hình khác có liên quan

- Thực nghiệm so sánh hiệu quả với 3 mô hình phát hiện mã độc IoT dựa trên chuỗi mã thực thi trên cùng một tập dữ liệu như của NCS thu thập, cùng một môi trường thiết bị: Kết quả so sánh được thể hiện trong Bảng 3.10 cho thấy rằng mô hình chúng tôi đề xuất đã cho kết quả phát hiện mã độc IoT có độ chính xác tốt nhất. Kết quả độ đo đã vượt trội hơn nhiều so với phương pháp của Ding đã đề xuất. Bên cạnh đó, với việc sử dụng số lượng opcode ít hơn thì việc trích xuất và phát hiện mã độc IoT nhanh hơn các mô hình còn lại do hai mô hình khác gặp hạn chế về thời gian và tài nguyên khi xử lý các tập tin có kích thước lớn.

- Đánh giá khả năng phát hiện đối các tập tin không thu thập được đặc trưng chuỗi lời gọi hệ thống từ phân tích động:

Từ thực nghiệm đã chứng minh mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng thu thập từ phân tích tĩnh tập tin đã khắc phục hạn chế và hỗ trợ tốt đối với trường hợp mô hình phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng thu thập từ phân tích động đề xuất trong chương 2 của luận án trong một số trường hợp. Đây là cơ sở để xây dựng các giải pháp phát hiện mã

độc IoT dựa trên học máy toàn diện trong thực tế. Các mô hình phát hiện mã độc IoT đơn kiến trúc dựa trên đặc trưng thu thập từ phân tích động và phân tích tĩnh có thể hỗ trợ, khắc phục các hạn chế của nhau, giúp giải pháp có thể phát hiện đối với tất cả các trường hợp phân tích một tập tin thực thi.

- So sánh hiệu quả với các mô hình phát hiện sử dụng ít đặc trưng mã thực thi khác: So sánh hiệu quả với các nghiên cứu khác có liên quan được thể hiện trong bBng 3.11. Kết quả cho thấy mô hình sử dụng số lượng mã thực thi trong từ điển ít nhất cần thu thập từ tập tin nhưng độ chính xác và thời gian phát hiện vẫn phù hợp trong môi trường IoT khi so sánh với các mô hình khác đã công bố.

3.5. Kết luận chương 3

Trong chương này, đặc trưng chuỗi mã thực thi của các tập tin trên thiết bị IoT đã được sử dụng và cho thấy hiệu quả tốt trong xây dựng các mô hình phát hiện mã độc IoT dựa trên học máy. Luận án đã đề xuất phương pháp thu thập và lựa chọn đặc trưng chuỗi mã thực thi của tập tin thực thi trên môi trường IoT phục vụ xây dựng mô hình phát hiện mã độc IoT đơn kiến trúc hiệu quả. Các thử nghiệm để đánh giá hiệu quả của phương pháp và đưa ra được "từ điển opcode" hiệu quả cho mô hình phát hiện mã độc IoT trên kiến trúc MIPS với độ chính xác cao nhất là 99,8%. Bên cạnh đó, với việc áp dụng các thuật toán học máy truyền thống giúp giảm độ phức tạp tính toán và thời gian huấn luyện và sử dụng mô hình trong môi trường IoT hạn chế tài nguyên. Mặt khác, các thực nghiệm đã chứng minh mô hình phát hiện mã độc IoT đã đề xuất dựa trên phân tích tĩnh có thể giải quyết hiệu quả đối với trường hợp các tập tin không thể thu thập thông tin từ phân tích động đã trình bày trong chương 2 của luận án. Với sự phát triển của các mã độc IoT, đây là cơ sở để xây dựng các giải pháp toàn diện trong phát hiện mã độc IoT đơn kiến trúc dựa trên đặc trưng của tập tin thực thi trong thực tế.

Tuy nhiên, các mô hình đã đề xuất trong chương 2 và chương 3 đã có khả năng phát hiện hiệu quả mã độc IoT đơn kiến trúc, tuy nhiên với sự phát triển của mã độc trên các thiết bị IoT đa dạng, việc phát hiện các mã độc trên kiến trúc mới và các biến thể mã độc từ các kiến trúc phổ biến trước đây là rất cần thiết. Vì vậy, để hướng đến giải quyết bài toán tổng thể của luận án, mô hình phát hiện mã độc IoT dựa trên học máy được nghiên cứu mở rộng để giải quyết các vấn đề phát hiện mã độc IoT đa kiến trúc và dự đoán mã độc zero-day hoạt động đa kiến trúc trên thiết bị IoT trong nội dung tiếp theo của luận án.

Ý tưởng tiếp cận và các kết quả nghiên cứu trình bày trong chương này đã được NCS công bố tại hội thảo và đăng tạp chí [TC2] và [TC4].

CHƯƠNG 4: XÂY DỰNG MÔ HÌNH PHÁT HIỆN MÃ ĐỘC IOT ĐA KIẾN TRÚC HIỆU QUẢ SỬ DỤNG ĐẶC TRƯNG CHUYỂN ĐỔI CHÉO KIẾN TRÚC VI XỬ LÝ

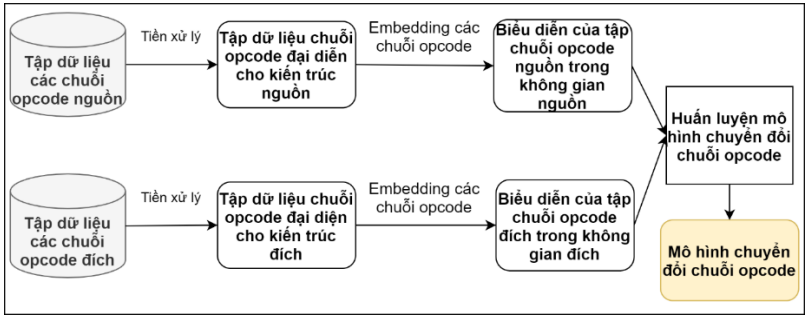
4.1. Mở đầu

Để nâng cao hiệu quả trong phát hiện mã độc IoT đa kiến trúc dựa trên xây dựng mô hình phát hiện IoT chéo kiến trúc vi xử lý, chương 4 luận án đề xuất phương pháp chuyển đổi đặc trưng chuỗi mã thực thi chéo kiến trúc vi xử lý phục vụ xây dựng mô hình phát hiện mã độc IoT đa kiến trúc dựa trên các thuật toán học máy truyền thống nhằm tăng cường tập dữ liệu đa kiến trúc vi xử lý, tăng độ chính xác và khả năng phát hiện, dự báo mã độc zero-day trên thiết bị IoT sử dụng kiến trúc vi xử lý đa dạng. Mô hình đề xuất có thể thay thế việc cần kết hợp nhiều mô hình phát hiện mã độc IoT đơn kiến trúc để phát hiện mã độc cho một tập tin thực thi và giải quyết vấn đề hạn chế dữ liệu huấn luyện trong môi trường IoT đa dạng kiến trúc vi xử lý.

4.2. Đề xuất mô hình chuyển đổi đặc trưng của tập tin thực thi

Với hướng tiếp cận xây dựng mô hình phát hiện mã độc IoT chéo kiến trúc vi xử lý, qua các khảo sát trong luận án, chúng tôi đề xuất sử dụng đặc trưng dạng chuỗi trích xuất từ phân tích tĩnh để thực hiện xây dựng mô hình phát hiện mã độc IoT đa kiến trúc. Phương pháp chuyển đổi chéo kiến trúc vi xử lý đặc trưng chuỗi mã lệnh thực thi thu thập từ phân tích tĩnh tập tin được đề xuất để tăng cường tập dữ liệu đa kiến trúc phục vụ xây dựng mô hình phát hiện mã độc IoT. Dựa vào cách tiếp cận đặc trưng mã thực thi thu thập từ phân tích tĩnh tập tin là các biểu diễn dạng chuỗi, việc xử lý dữ liệu chuỗi mã thực thi chéo kiến trúc để xác định sự tương đồng sẽ được thực hiện thông qua việc chuyển đổi đặc trưng chuỗi mã thực thi chéo kiến trúc. Việc chuyển đổi đặc trưng chuỗi mã thực thi chéo nền tảng kiến trúc giúp hình thức các tập dữ liệu đa kiến trúc vi xử lý, tạo ra mô hình có khả năng phát hiện mã độc cho đặc trưng đầu vào thuộc nhiều nền tảng kiến trúc khác nhau.

Phép biến đổi chuỗi mã thực thi trong luận án được định nghĩa 4.2 và quá trình xây dựng mô hình chuyển đổi chuỗi mã thực thi được mô tả như hình sau:



4.2.1. Biểu diễn mã thực thi trong không gian nhúng

Quá trình bắt đầu với việc tiền xử lý tập dữ liệu các chuỗi mã thực thi thu thập được sau khi trích xuất từ các tập tin trên kiến trúc vi xử lý nguồn và đích. Các chuỗi mã thực thi của mỗi kiến trúc vi xử lý được ghép lại để trở thành các tập dữ liệu mã thực thi đại diện cho các kiến trúc vi xử lý. Việc này giống như tạo ra một "văn bản" trong ngôn ngữ tự nhiên từ các câu là các chuỗi mã thực thi. Sau đó, tập dữ liệu chuỗi mã thực thi đại diện cho mỗi kiến trúc được nhúng trong không gian để tạo ra biểu diễn của tập chuỗi mã thực thi trong không gian tương ứng với mỗi kiến trúc vi xử lý.

Lựa chọn giữa các phương pháp này thường phụ thuộc vào yêu cầu cụ thể của bài toán cũng như tính chất của dữ liệu. Skip-gram thích hợp cho các tác vụ đòi hỏi biểu diễn chi tiết cho các từ hiếm, trong khi CBOW có thể nhanh chóng học các từ phổ biến. FastText có thể hữu ích khi cần xử lý các từ không có trong từ điển thông thường bằng cách sử dụng thông tin từ n-gram. Vì vậy, luận án sử dụng giải pháp FastText để thực hiện nhúng các chuỗi mã thực thi thành vectơ trong không gian. Quá trình hoạt động của skipgram trong FastText thường bao gồm các bước: (1) Chuẩn bị dữ liệu, (2) Xây dựng cặp mã thực thi – ngữ cảnh, (3) Tạo tập dữ liệu huấn luyện, (4) Huấn luyện mô hình, (5) Tạo vector biểu diễn mã thực thi.

4.2.2. Huấn luyện các mô hình chuyển đổi chuỗi mã thực thi

Sau khi đã có biểu diễn của các tập chuỗi mã thực thi trong không gian vector nhúng, việc xây dựng mô hình chuyển đổi tốt nhất được thực hiện thông qua ba bước gồm: Xây dựng bảng ánh xạ mã thực thi, mô hình hoá chuỗi mã thực thi và chuyển đổi ngược lặp đi lặp lại.

Cụ thể quá trình huấn luyện mô hình chuyển đổi chuỗi mã thực thi giữa hai kiến trúc vi xử lý được trình bày giả mã trong thuật toán 4.1 dưới đây:

Thuật toán 4.1: Huấn luyện mô hình chuyển đổi chuỗi mã thực thi

Đầu vào: *Hai tập dữ liệu chuỗi mã thực thi của hai kiến trúc trong không gian nhúng* (O_n, O_d).

Đầu ra: *Mô hình chuyển đổi chuỗi mã thực thi giữa kiến trúc nguồn và kiến trúc đích.*

 # Xây dựng bảng ánh xạ mã thực thi

1: Opcode _ translation_table ← $\boldsymbol{f(O_n, O_d | min_{distance}(WO_n, O_d))}$

 # mô hình hoá chuỗi mã thực thi

2: Opcode_sequences_ Model ← Statistical Language Models by Moses

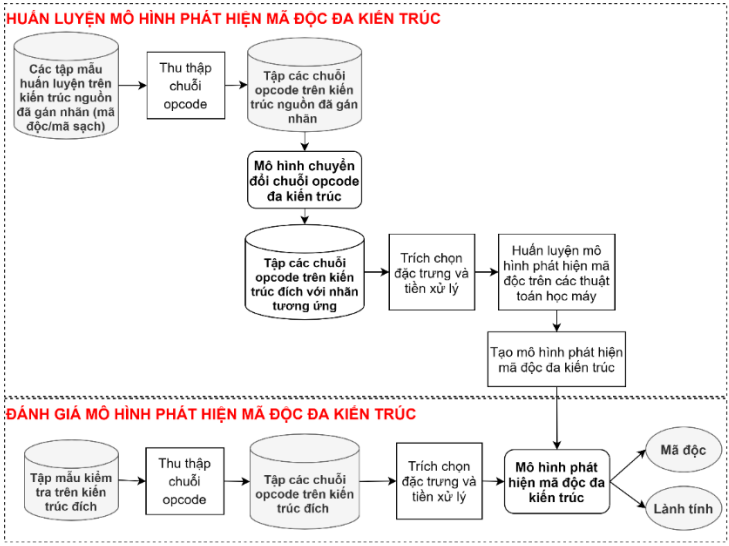
 # Xác định các giá trị hạt giống $P^0_{n \to d}$ *dựa trên bảng ánh xạ mã thực thi và mô hình hoá chuỗi mã thực thi*

3: $\boldsymbol{P^0_{n\to d}}$ ← Opcode _ translation_table & Opcode_sequences_ Model
Sử dụng giá trị hạt giống để chuyển đổi chuỗi mã thực thi nguồn sang đích
4: $\boldsymbol{Y^0_d \leftarrow P^0_{n\to d}}$
 # Quá trình chuyển đổi ngược lặp lại N lần
5: **For** i=1 **to** N **do**
 # Huấn luyện mô hình chuyển đổi mã thực thi đích sang mã thực thi nguồn
6: Train model $\boldsymbol{P^i_{d\to n} \leftarrow Y^{i-1}_d}$
 # Sử dụng mô hình P để chuyển đổi chuỗi mã thực thi đích
7: $\boldsymbol{Y^i_n \leftarrow P^i_{d\to n}}$
 # Huấn luyện mô hình chuyển đổi mã thực thi nguồn sang đích
8: Train model $\boldsymbol{P^i_{n\to d} \leftarrow Y^i_n}$
 # Sử dụng giá trị P để chuyển đổi chuỗi mã thực thi nguồn
9: $\boldsymbol{Y^i_d \leftarrow P^i_{n\to d}}$
10: **Endfor**
11: **Return** $\boldsymbol{Y^i_d}$

Công việc xác định và đánh giá hiệu quả của quá trình chuyển đổi đặc trưng chuỗi mã thực thi chéo kiến trúc trên các thiết bị còn hạn chế do NCS chưa khảo sát được một công bố về bảng ánh xạ hoặc chuyển đổi chuỗi mã thực thi từ kiến trúc vi xử lý các thiết bị IoT khác nhau. Để có cơ sở đánh giá, dựa trên các khảo sát kỹ thuật tạo mã độc IoT hoạt động đa kiến trúc vi xử lý, nghiên cứu sinh tiến hành thu thập các mã nguồn mã độc IoT đã công bố từ đó thực hiện kỹ thuật cross-compiler mã nguồn để thu thập các tập tin thực thi trên các kiến trúc vi xử lý của thiết bị IoT. Các tập tin thực thi trên các kiến trúc đó được chúng tôi thực hiện dịch ngược và phân tích tĩnh thủ công nhằm xác định các vùng chuỗi mã thực thi tương ứng phù hợp trên hai kiến trúc vi xử lý phục vụ điều chỉnh các tham số trong quá trình huấn luyện mô hình chuyển đổi chuỗi mã thực thi chéo kiến trúc. Mặt khác, chúng tôi thực hiện so sánh và đánh giá kết quả chuỗi mã thực thi thu được sau khi chuyển đổi với chuỗi mã thực thi thu thập từ nền tảng ban đầu của cùng một loại mã độc IoT để hiệu chỉnh các tham số trong quá trình xây dựng mô hình chuyển đổi chuỗi mã thực thi chéo kiến trúc.

4.3. Xây dựng mô hình phát hiện mã độc IoT đa kiến trúc dựa trên mô hình chuyển đổi đặc trưng chéo kiến trúc vi xử lý đề xuất

Quá trình xây dựng mô hình phát hiện mã độc IoT đa kiến trúc dựa trên phương pháp chuyển đổi đặc trưng chuỗi opcode luận án đề xuất gồm 2 giai đoạn: Huấn luyện mô hình phát hiện mã độc đa kiến trúc và đánh giá mô hình phát hiện. Chi tiết được thể hiện trong hình sau:



4.3.1. Huấn luyện mô hình phát hiện mã độc IoT đa kiến trúc dựa trên mô hình chuyển đổi chuỗi mã thực thi

Giai đoạn thực hiện các công việc sau đây:

- Thu thập chuỗi mã thực thi.
- Tăng cường tập dữ liệu chuỗi mã thực đa kiến trúc thông qua mô hình chuyển đổi chuỗi mã thực thi chéo kiến trúc vi xử lý.
- Trích chọn đặc trưng và tiền xử lý.
- Huấn luyện mô hình phát hiện mã độc trên các thuật toán học máy.

4.3.2. Đánh giá mô hình phát hiện mã độc IoT đề xuất

Đối với quá trình đánh giá mô hình đề xuất, luận án đánh giá thông qua phương pháp thực nghiệm và tiến hành đánh giá theo các kịch bản đánh giá trong bảng 4.3.

4.4. Thử nghiệm và đánh giá kết quả mô hình đề xuất

4.4.1. Môi trường thử nghiệm

Các thử nghiệm của luận án được thực hiện trên cùng một máy tính. Các thử nghiệm được tiến hành 10 lần tại các mốc thời gian khác nhau. Kết quả đánh giá là trung bình cộng của các giá trị thu được.

4.4.2. Tập dữ liệu thử nghiệm

Tập dữ liệu sử dụng để thử nghiệm là các mẫu được lựa chọn trong C500-IoT dataset. Bên cạnh đó, hiện nay để tạo ra các mã độc có khả năng hoạt động trên nhiều kiến trúc vi xử lý, luận án đã thu thập các mã nguồn chương trình mã độc công bố trên các nguồn khác nhau để tiến hành phân tích hành vi mã độc trong mã nguồn và biên dịch tập tin thực thi trên nhiều kiến trúc vi xử lý khác nhau

phục vụ các kịch bản thử nghiệm. Chi tiết thông tin về tập dữ liệu đa kiến trúc phục vụ thử nghiệm được mô tả trong bảng 4.4.

4.4.3. Kết quả xây dựng kịch bản thử nghiệm

- Kết quả thu thập chuỗi mã thực thi của các tập dữ liệu thử nghiệm thể hiện trong bảng 4.5.

- Kết quả tập dữ liệu chuỗi mã thực thi thu thập được phân theo ngưỡng độ dài tối thiểu thể hiện chi tiết trong bảng 4.6.

- Kết quả chuyển đổi đặc trưng chuỗi mã thực thi chéo kiến trúc thể hiện như bảng 4.7, hình 4.11 và hình 4.12.

- Kết quả xây dựng tập dữ liệu chuỗi mã thực thi phục vụ đánh giá mô hình theo các kịch bản thể hiện chi tiết trong bảng 4.8.

4.4.4. Kết quả đánh giá các mô hình phát hiện mã độc IoT đa kiến trúc đề xuất

Với kết quả đã đạt được trong chương 3 của luận án, để xây dựng mô hình phát hiện hiệu quả trên các kịch bản thử nghiệm, các thuật toán học máy gồm SVM, RF, NB tiếp tục được chúng tôi sử dụng kết hợp với phương pháp n-gram.

Trong quá trình thực nghiệm các kịch bản thử nghiệm, NCS đã tiến hành thực nghiệm nhiều lần để lựa chọn các siêu tham số phù hợp đối với mỗi thuật toán học máy. Quá trình huấn luyện và kiểm tra khả năng phát hiện của các mô hình, NCS đồng thời tiến hành tinh chỉnh các tham số chính của các thuật toán học máy để tìm ra mô hình phù hợp nhất. Tham số chính được sử dụng trong các thuật toán học máy trong các thực nghiệm sử dụng được mô tả trong bảng 4.9.

Kết quả thử nghiệm các kịch bản thể hiện trong bảng 4.10 và bảng 4.11. Trong trường hợp huấn luyện trên nền tảng kiến trúc Intel để xây dựng mô hình phát hiện trên nền tảng kiến trúc MIPS cho độ chính xác tốt nhất đạt accuracy là 98,8% và đạt 99,4% trong trường hợp ngược lại huấn luyện trên kiến trúc MIPS để phát hiện trên kiến trúc Intel khi sử dụng thuật toán Random Forest. Tuy nhiên, mô hình phát hiện sử dụng thuật toán Naive Bayes cho kích thước mô hình và thời gian phát hiện tốt nhất, phù hợp cao khi xây dựng trong các giải pháp bảo mật trên môi trường IoT tài nguyên hạn chế mà vẫn có thể đảm bảo độ chính xác phát hiện trên 95%. Vì vậy, khi huấn luyện và triển khai ứng dụng các mô hình phát hiện mã độc đa kiến trúc trong môi trường IoT thực tế cần lựa chọn các thuật toán và phương pháp trích chọn đặc trưng n-gram phù hợp để đảm bảo về mặt thời gian và kích thước của mô hình phát hiện cài đặt trên các thiết bị IoT.

Bên cạnh đó, thời gian phát hiện của các mô hình học máy khi sử dụng các thuật toán SVM, NB, RF là khác nhau. Về mặt tổng thể khi so sánh, đánh giá với các nghiên cứu đã khảo sát trong nội dung 1.3 của luận án này, các mô hình phát hiện

mã độc IoT đa kiến trúc trên đảm bảo các độ đo độ chính xác, kích thước mô hình và thời gian phát hiện đáp ứng các yêu cầu ứng dụng triển khai trong các giải pháp bảo mật trên môi trường IoT.

Với mục tiêu của luận án và từ kết quả phát hiện mã độc IoT, mô hình phát hiện huấn luyện chéo tập dữ liệu sử dụng thuật toán NB và 2-gram được lựa chọn để thử nghiệm phát hiện mã độc đa kiến trúc và chứng minh khả năng dự đoán mã độc zero-day trên các kiến trúc vi xử lý khác nhau. Kết quả phát hiện của mô hình của mô hình khi sử dụng tập dữ liệu trên nền tảng kiến trúc Intel để huấn luyện 3 mô hình phát hiện mã độc trên các nền tảng kiến trúc Intel, MIPS và PowerPC thể hiện trong bảng 4.12. Kết quả cho thấy rằng phương pháp chuyển đổi đặc trưng chuỗi mã thực thi có thể mở ra hướng tiếp cận mới trong phát hiện mã độc zero-day hoạt động đa kiến trúc vi xủ lý trên các kiến trúc vi xử lý mới và ít tri thức mã độc trước đó trong tương lai.

4.5.5. So sánh hiệu quả với các mô hình khác có liên quan

Để đánh giá hiệu quả của mô hình đề xuất trong phát hiện mã độc IoT đa kiến trúc dựa trên phương pháp phát hiện chéo kiến trúc vi xử lý, luận án đã tiến hành thực nghiệm và so sánh với một số phương pháp xây dựng mô hình phát hiện mã độc IoT đa nền tảng kiến trúc dựa trên cách tiếp cận huấn luyện và phát hiện chéo kiến trúc. Kết quả so sánh được thể hiện trong bảng 4.13. Từ kết quả trong bảng 4.13 chỉ ra rằng phương pháp đề xuất đã giải quyết được hạn chế của các mô hình hiệu quả khác trong phát hiện mã độc đa kiến trúc dựa trên phát hiện chéo khi có sự khác biệt về đặc trưng mã thực thi trên mỗi kiến trúc vi xử lý. Bên cạnh việc nâng cao độ chính xác trong phát hiện, mô hình đề xuất còn mở ra khả năng ứng dụng trong xây dựng các giải pháp phát hiện mã độc zero-day, mã độc hoạt động trên các kiến trúc vi xử lý mới được phát triển trong thời gian tới.

Tuy nhiên, trong quá trình thực nghiệm xây dựng mô hình chuyển đổi tập đặc trưng chuỗi mã thực thi giữa các kiến trúc vi xử lý, một số chuỗi mã thực thi của kiến trúc nguồn có độ dài nhỏ, không đa dạng tập mã thực xuất hiện trong chuỗi đã dẫn đến mô hình chuyển đổi chuỗi mã thực thi giữa hai kiến trúc chưa thể chuyển đổi thành công chuỗi mã thực thi do việc biểu diễn và xác định sự tương quan đặc trưng chuỗi mã thực thi chéo kiến trúc vi xử lý còn hạn chế.

4.5. Kết luận chương 4

Trong chương này, luận án đã đề xuất phương pháp chuyển đổi chuỗi mã thực thi chéo kiến trúc vi xử lý để phục vụ xây dựng mô hình phát hiện mã độc IoT đa kiến trúc hiệu quả dựa trên học máy và đặc trưng của tập tin thực thi. Kết quả thực nghiệm đã thể hiện khả năng phát hiện mã độc đa kiến trúc với độ chính xác

tốt, kích thước và thời gian phát hiện của mô hình phù hợp khi triển khai trong môi trường IoT tài nguyên hạn chế. Hơn nữa, phương pháp chuyển đổi chuỗi mã thực thi chéo kiến trúc luận án đề xuất đã chứng minh hiệu quả trong việc phát hiện mã độc hoạt động trên các kiến trúc vi xử lý có sự khác biệt lớn khi so sánh với các mô hình khác có cùng cách tiếp cận. Phương pháp chuyển đổi đặc trưng chuỗi mã thực thi cho phép xây dựng các mô hình dự báo mã độc zero-day trên các kiến trúc vi xử lý khác nhau thông qua xây dựng các tập dữ liệu chuỗi mã thực thi mã độc trên các kiến trúc vi xử lý mới từ các mã độc đã biết. Tuy nhiên, việc xác định mức độ tương đồng giữa các chuỗi opcode của 2 kiến trúc có khác biệt lớn còn hạn chế. Trong tương lai, việc tiếp tục nghiên cứu các phương pháp trích xuất đặc trưng opcode và chuyển đổi đặc trưng opcode chéo nền tảng khác là cần thiết.

Ý tưởng và kết quả thực nghiệm của mô hình đề xuất trong chương này đã được trình bày, công bố trên các Tạp chí khoa học quốc tế [TC1].

KẾT LUẬN VÀ KIẾN NGHỊ

Nội dung của luận án đã tập trung nghiên cứu các phương pháp phát hiện mã độc IoT dựa trên học máy sử dụng đặc trưng biểu diễn dạng chuỗi thu thập từ phân tích động và phân tích tĩnh tập tin thực thi. Từ đó, luận án nâng cao độ chính xác, giảm số lượng và thời gian trích xuất đặc trưng của tập tin, giảm thời gian phát hiện, nâng cao khả năng phát hiện, dự báo mã độc hoạt động đa kiến trúc, mã độc zero-day của các mô hình phát hiện mã độc IoT dựa trên học máy.

Các mô hình phát hiện mã độc trên thiết bị IoT đề xuất trong luận án có tính thực tiễn với việc triển khai ứng dụng một phần trong thực tế (ĐT1, ĐT2, ĐT3). Tuy nhiên, với sự phát triển của các thiết bị và công nghệ IoT, mã độc trên các thiết bị IoT sẽ tiếp tục được xây dựng, cải tiến và lây lan sâu rộng trong thời gian tới. Mặc dù luận án đã đạt được các kết quả nghiên cứu quan trọng về lý luận và thực tiễn, nhưng vẫn còn một số vấn đề cần nghiên cứu và cải tiến sau đây:

Thứ nhất, thử nghiệm và điều chỉnh tham số các thuật toán học máy của các mô hình đề xuất trong luận án với các tập dữ liệu mới, các tập dữ liệu uy tín khác.

Thứ hai, nghiên cứu kết hợp cả hai đặc trưng biểu diễn dạng chuỗi thu thập từ phân tích động và phân tích tĩnh để xây dựng một mô hình phát hiện mã độc IoT hiệu quả tối ưu nhất dựa trên phương pháp phân tích lai.

Thứ ba, cải tiến về môi trường sandbox và các phương pháp trích xuất chuỗi opcode để phục vụ xây dựng các mô hình phát hiện mã độc IoT hiệu quả hơn.

Thứ tư, các tiêu chí khác về tính hiệu quả của mô hình như độ tin cậy của đầu vào, mức độ tài nguyên sử dụng cũng cần được tiếp tục nghiên cứu, cải tiến.

DANH MỤC CÔNG TRÌNH CÔNG BỐ

[TC1]: Luong The Dung, Nguyen Ngoc Toan, Tran Nghi Phu, CAIMP: Cross-architecture IoT malware detection and prediction based on static feature, The Computer Journal, 2024;bxae042, https://doi.org/10.1093/comjnl/bxae042 (SCIE index, Q2).

[TC2]: Nguyen Ngoc Toan, Luong The Dung, Dang Quang Thang, Static Feature Selection for IoT Malware Detection, Journal of Science and Technology on Information Security, Special Issue CS (15) 2022, https://doi.org/10.54654/isj.v1i15.844, ISSN 2615 -9570.

[TC3]: Toan Nguyen Ngoc, Dung Luong The, Phu Tran Nghi, A novel approach to detect IoT malware by System calls and Long Short-term Memory model, Journal of Theoretical and Applied Information Technology, 31st August 2021 -- Vol. 99. No. 16 – 2021 (SCOPUS,Q4), ISSN: 1992-8645.

[TC4]: Tran Nghi Phu, Nguyen Dai Tho, Le Huy Hoang, Nguyen Ngoc Toan, Nguyen Ngoc Binh, An Efficient Algorithm to Extract Control Flow-Based Features for IoT Malware Detection, The Computer Journal, Volume 64, Issue 4, April 2020, Pages 599–609, https://doi.org/10.1093/comjnl/bxaa087. (SCIE index, Q2).

Một số công trình tham khảo khác đã công bố:

[HT1]: Ngoc Toan Nguyen, Xuan Tuan Le; The Dung Luong, An Ensemble Method for Sentiment Classification of Long Vietnamese Documents, 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 2022, pp. 428-433, doi: 10.1109/RIVF55975.2022.10013802. (SCOPUS index), ISSN: 2162-786X.

Bên cạnh đó, một phần kết quả nghiên cứu được ứng dụng trong các đề tài nghiên cứu khoa học NCS là đồng tác giả sau đây:

[ĐT1]: Đề tài khoa học và công nghệ cấp Bộ "Nghiên cứu xây dựng hệ thống phân tích, phát hiện mã độc và lỗ hổng bảo mật trong phần sụn (firrmware) của một số thiết bị mạng", đã nghiệm thu năm 2022.

[ĐT2]: Đề tài khoa học và công nghệ cấp Bộ "Nghiên cứu xây dựng hệ thống diễn tập phòng thủ và tấn công mạng phục vụ công tác đào tạo, huấn luyện đảm bảo an toàn thông tin mạng", đã nghiệm thu năm 2022.

[ĐT3]: Đề tài khoa học và công nghệ cấp cơ sở "Nghiên cứu phát triển công cụ giám sát máy chủ tài Học viện An ninh nhân dân", đã nghiệm thu năm 2022**.**