Phát hiện mã độc IoT botnet dựa trên đồ thị PSI với mô hình Skip-gram

Ngô Quốc Dũng, Lê Văn Hoàng, Nguyễn Huy Trung

***Tóm tắt*— Trong bài báo này, nhóm tác giả đề xuất một phương pháp phát hiện mã độc IoT botnet dựa trên đồ thị PSI (Printable String Information) sử dụng mạng nơ-ron tích chập (Convolutional Neural Network - CNN). Thông qua việc phân tích đặc tính của Botnet trên các thiết bị IoT, phương pháp đề xuất xây dựng đồ thị để thể hiện các mối liên kết giữa các PSI, làm đầu vào cho mô hình mạng nơ-ron CNN phân lớp. Kết quả thực nghiệm trên bộ dữ liệu 10033 tập tin ELF gồm 4002 mẫu mã độc IoT botnet và 6031 tập tin lành tính cho thấy phương pháp đề xuất đạt độ chính xác (accuracy) và độ đo F1 lên tới 98,1%.**

***Abstract*— In this paper, the authors propose a method for detecting IoT botnet malware based on PSI graphs using Convolutional Neural Network (CNN). Through analyzing the characteristics of Botnet on IoT devices, the proposed method construct the graph to show the relations between PSIs, as input for the CNN neural network model. Experimental results on the 10033 data set of ELF files including 4002 IoT botnet malware samples and 6031 benign files show Accuracy and F1-score up to 98.1%.**

***Từ khóa*— IoT botnet; đồ thị Printable String Information (PSI) ; Mạng nơ ron tích chập.**

***Keywords*— IoT botnet; Printable String Information graph; Convolutional Neural Network.**

I. GIỚI THIỆU

Cuộc cách mạng công nghiệp 4.0 hay còn được gọi với những cái tên như Internet vạn vật (Internet of Things) hay công nghiệp Internet (Industrial Internet) làm biến đổi nhanh chóng nền công nghiệp ở mọi quốc gia, diễn ra trên toàn cầu. Với nhiều tên gọi khác nhau nhưng đặc điểm nổi bật nhất của cuộc cách mạng công nghiệp lần thứ 4 đó là việc dịch chuyển các hệ thống máy móc sản xuất truyền thống sang các hệ thống tự động hoá có khả năng tự hành một cách thông minh dựa trên nền tảng của điện tử viễn thông và công nghệ thông tin. Dựa trên cuộc cách mạng công nghiệp 4.0 mà giáo dục, y tế, chính trị, xã hội, kinh tế đã có những thành tựu vượt bậc trong thời gian ngắn. Bên cạnh những tiện ích mà cuộc cách mạng công nghiệp 4.0 mang lại thì an toàn thông tin trên không gian mạng ngày càng trở nên phức tạp, tiềm ẩn nhiều nguy cơ ảnh hưởng trực tiếp tới an ninh quốc gia, tới lợi ích hợp pháp của người dân. Những nguy cơ này ngày càng hiện hữu khi mà các chuỗi cung ứng, nhà máy, người tiêu dùng và các hoạt động liên quan được kết nối với nhau thông qua các thiết bị IoT. Việc đảm bảo an ninh, an toàn thông tin cho các thiết bị IoT đã và đang thu hút nhiều nhà nghiên cứu và các tổ chức. Các nghiên cứu, công trình công bố có thể chia thành hai nhóm chính gồm: phân tích tĩnh và phân tích động.

Phân tích động hay còn được gọi là phân tích hành vi thực hiện việc giám sát toàn bộ thiết bị hoặc các tập tin thực thi trong quá trình hoạt động để phát hiện các hành vi bất thường. Theo hướng tiếp cận này, Celeda và cộng sự [1] giới thiệu phương pháp phát hiện mã độc Chuck Norris Botnet trên các thiết bị mô-đem bị lây nhiễm. Kết quả nghiên cứu cho thấy hầu hết mã độc lây lan thông qua giao thức telnet do các thiết bị sử dụng mật khẩu yếu hoặc mặc định của nhà sản xuất. Tuy nhiên nghiên cứu này chỉ áp dụng được trên kiến trúc MIPS. Để mở rộng phạm vi nghiên cứu trên các kiến trúc vi xử lý khác như ARM, PowerPC… bộ công cụ QEMU ngày càng được sử dụng rộng rãi. Trong [2], Jonas và cộng sự đã xây dựng framework Avatar để phân tích Firmware các thiết bị nhúng bằng cách phối hợp quá trình thực thi của bộ mô phỏng dựa trên QEMU với phần cứng thực tế. Bằng cách tiêm một phần mềm trung gian đặc biệt vào thiết bị nhúng, Avatar thực thi các chỉ thị firmware bên trong bộ mô phỏng trong khi đang truyền các thực thi vào/ra tới thiết bị vật lý. Tuy nhiên, quá trình thực thi mô phỏng chậm hơn nhiều so với quá trình thực thi trên thiết bị thực do việc đồng bộ tín hiệu thông không các kênh UART và JTAG không đảm bảo tốc độ truyền tin. Cùng hướng tiếp cận đó, Yin Minn Pa Pa và cộng sự [3] đã phát triển IoT

Bài báo được nhận ngày 4/10/2018. Bài báo được gửi phản biện thứ nhất vào ngày 14/10/2018 và được chấp nhận đăng vào ngày 5/12/2018. Bài báo được gửi phản biện thứ hai vào ngày 15/10/2018 và được chấp nhận đăng vào ngày 02/12/2018.

honeypot để chặn bắt mã độc IoT dựa trên giao thức telnet; và IoTBOX để phân tích mã độc IoT đa kiến trúc CPU, nhưng chỉ tập trung vào phân tích các hành vi mạng. Cũng dựa trên nền tảng QEMU, Ahmad Darki và cộng sự [4] đã đề xuất RARE – một hệ thống mô phỏng phân tích mã độc và lưu trữ tiểu sử các hành vi của mã độc trên các bộ định tuyến dân dụng (SOHO). Trong đó, RARE sử dụng phân tích tĩnh để cung cấp các thông tin cho quá trình phân tích động từ đó tùy chỉnh môi trường mô phỏng giúp mã độc có thể bộc lộ hết tất cả các hành vi độc hại, kết quả đạt 94% các mẫu mã độc có thể kích hoạt thành công. Tuy nhiên, đặc trưng thu thập qua phân tích tĩnh còn đơn giản (địa chỉ IP và tên miền) và quá trình tương tác giữa Bot và C&C chưa đầy đủ khi chưa thể tùy chỉnh được máy chủ C&C. A.Jacobsson và cộng sự [5] tập trung phát hiện các hành vi bất thường của các thiết bị IoT dân dụng. Chun-Jung Wu và cộng sự [6] đã đề xuất IoTProtect có thể kiểm tra các tiến trình chạy trên thiết bị IoT và dừng những tiến trình không xác định theo một chu kỳ nhất định, IoTProtect có thể triển khai trên các thiết bị thương mại mà không cần chỉnh sửa nhiều firmware. Tuy nhiên, điểm yếu tồn tại của phân tích động là chỉ cho phép phân tích đơn luồng và không thể quan sát tất cả các khả năng thực thi của mã độc [7]. Đồng thời kiến trúc vi xử lý của các thiết bị IoT rất đa dạng (MIPS, ARM, PowerPC…) nên yêu cầu về việc xây dựng môi trường thực thi đảm bảo cho các thiết bị IoT hoạt động để thu thập dữ liệu làm đầu vào cho quá trình phân tích là rất phức tạp.

Phân tích tĩnh [8] hay còn gọi là phân tích dựa trên đặc trưng bao gồm phân tích, phát hiện mã độc và/hoặc lỗ hổng bảo mật trong mã nguồn firmware hoặc các tập tin thực thi mà không phải chạy chúng. Hướng tiếp cận này sử dụng những kỹ thuật như đồ thị luồng điều khiển (CFG – Control Flow Graph), đồ thị luồng dữ liệu (DFG – Data Flow Graph), thực thi biểu tượng (SE – Symbolic Execution) [9] với các đặc trưng thường sử dụng để xác định mã độc như API, Opcode, PSI (Printable String Information), FLF (Function Length Frequency) [10]. Phân tích tĩnh sẽ giúp có một cách nhìn tổng quan các khả năng có thể xảy ra trong tập tin thực thi. Costin và cộng sự [11] đã đề xuất một framework để thu thập, lọc, unpack và phân tích tĩnh firmware quy mô rộng từ đó phát hiện lỗ hổng bảo mật, mã độc. Những nghiên cứu trên chỉ sử dụng các đặc trưng rời rạc mà không đi vào sự tương tác, liên quan giữa các đặc trưng…Trong khi đó, mã độc IoT botnet luôn có quy trình hoạt động khá tương đồng nhau và có sự tương tác với nhau [12], [13]. Chính vì thế trong bài báo này để tăng sự chính xác trong phát hiện mã độc IoT botnet, nhóm tác giả sử dụng đồ thị thể hiện sự liên kết giữa các đặc trưng đó. Tuy nhiên, hạn chế lớn nhất của phương pháp này là không phân tích được các tập tin có độ phức tạp lớn hoặc sử dụng các kỹ thuật gây rối (obfuscation).

Bên cạnh việc sử dụng phân tích tĩnh và phân tích động với học máy, phương pháp học sâu được sử dụng trong phân tích và phát hiện mã độc đem lại kết quả khả quan trong những năm gần đây. Yuan và cộng sự sử dụng hơn 200 đặc trưng từ quá trình phân tích tĩnh động làm đầu vào cho mạng học sâu DBN cho phép đạt được độ chính xác lên tới 96% trong việc phân loại mã độc và tệp tin lành tính [14]. Saxe và Berlin [15] đề xuất mô hình dựa trên mạng nơ-ron truyền thẳng để trích xuất các đặc trưng từ hơn 40,000 tập tin nhị phân ứng dụng Windows, kết quả đạt được độ chính xác 95% với tỷ lệ dương tính giả (false positive rate) là 0,1%. Nghiên cứu của Hamed và cộng sự [16] đã đề xuất giải pháp sử dụng cấu trúc LSTM với RNN (Recurrent Neural Network) trong phát hiện mã độc trên thiết bị IoT dựa trên đặc trưng OpCode trích xuất từ các ứng dụng thực thi nền tảng ARM, độ chính xác đạt 98%. Tuy nhiên các nghiên cứu này mới áp dụng phương pháp học sâu vào phân tích dữ liệu thu thập được từ quá trình hoạt động của hệ thống, mà chưa khai thác những đặc thù của mã độc Botnet, lớp mã độc phổ biến nhất trên các thiết bị IoT.

Trong bài báo này, nhóm tác giả đề xuất sử dụng mạng nơ-ron tích chập (Convolutional Neural Network) để phát hiện mã độc Botnet dựa trên các đặc trưng trích xuất từ đồ thị PSI. Đóng góp chính của bài báo là:

- Đề xuất thuật toán sinh đồ thị PSI từ các tập tin nhị phân của mã độc IoT botnet.
- Đề xuất mạng nơ-ron tích chập trong việc gán nhãn mã độc và tập tin lành tính với độ chính xác, cũng như độ đo F1 lên tới 98%.

Phần còn lại của bài báo được cấu trúc như sau: Mục II giải thích chi tiết giải pháp đề xuất. Mục III sẽ thảo luận triển khai thử nghiệm và tập

dữ liệu được sử dụng. Cuối cùng, Mục IV là trình bày kết quả và định hướng nghiên cứu.

II. PHƯƠNG PHÁP ĐỀ XUẤT

Trong phần này, nhóm tác giả sẽ giới thiệu các bước thực hiện chính trong mô hình tổng quan. Sau đó đi vào trình bày chi tiết các bước sinh đồ thị PSI từ đồ thị CFG. Với kết quả thu được sẽ tiến hành tiền xử lý thông qua mô hình skip-gram để chuyển đổi đồ thì PSI thành các biểu diễn vector. Cuối cùng là áp dụng mô hình mạng CNN để phân lớp tập tin mã độc và lành tính.

A. Tổng quan mô hình đề xuất

Dựa trên những công bố [13, 17, 18], nhóm tác giả thấy rằng các đặc trưng cơ bản của mã độc IoT botnet thường diễn ra theo một quy trình, cụ thể các bước là:

1. Cố gắng kết nối/nhận từ/đến máy chủ C&C ở xa thông qua địa chỉ IP hoặc URL.

2. Cố gắng khai thác các thiết bị IoT bằng cách liên tục dò quét ngẫu nhiên địa chỉ IP và thực hiện tấn công vét cạn thông qua các dịch vụ Telnet, SSH, FTP với một bộ từ điển nhúng sẵn trong tập tin (ví dụ root/root, adim/root, admin/123, …).

3. Cố gắng phân tích kiến trúc phần cứng của thiết bị IoT và tải về các tập tin nhị phân mã độc cần thiết (MIPS, ARM, PowerPC,…) với đoạn mã kịch bản thông qua giao thức wget, TFTP để lây nhiễm trên các thiết bị.

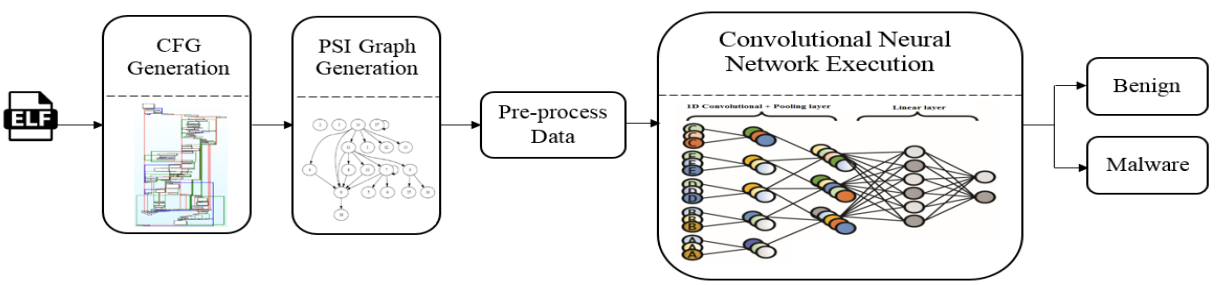
4. Cố gắng tìm kiếm các loại mã độc khác trên thiết bị để hủy hoặc xóa chúng ngay khi lây nhiễm thành công để đảm bảo tài nguyên bởi các thiết bị IoT là những thiết bị có tài nguyên hạn hẹp (ví dụ Mirai tìm và hủy các tiến trình của mã độc .anime và Qbot).

5. Cố gắng chạy trên bộ nhớ của các thiết bị IoT sau đó sẽ tạm dừng hoạt động cho đến khi nhận được lệnh từ kẻ tấn công.

Một điều quan trọng ở đây là mã độc IoT Botnet thường có quy trình thực hiện các bước theo trình tự và hầu hết trong đó yêu cầu các thông tin quan trọng như địa chỉ IP, URL, tên miền…, được gọi là PSI. PSI là một trong những đặc trưng thường được sử dụng trong phân tích tĩnh như [10, 19] để xác định một tập tin ELF là mã độc hay không. Bởi trong nghiên cứu [11] đã cho thấy có rất nhiều hệ điều hành được sử dụng trên các thiết bị IoT như Linux, Windows CE, VXWorks, rtems… nhưng sự phổ biến của các thiết bị IoT dựa trên nền tảng Linux là hơn cả, vì thế trong bài báo này nhóm tác giả sử dụng các tập tin thực thi trên nền tảng Linux là ELF là dữ liệu để thử nghiệm tính đúng đắn của phương pháp đề xuất.

Tuy nhiên những phương pháp đó thường tập trung vào việc kết hợp các đặc trưng, ví dụ như kết hợp tần suất xuất hiện của PSI với FLF (Function Length Frequency), việc kết hợp các đặc trưng giúp cải thiện độ chính xác của bộ phân lớp học máy. Tuy nhiên, những hướng tiếp cận đó không phân tích sự liên kết giữa các PSI, không xem xét đến ngữ cảnh của PSI mặc dù nó biểu diễn chuỗi thông tin mang tính trình tự và lặp lại trong tất cả các mã độc Botnet. Để cải thiện độ chính xác trong phát hiện mã độc dựa trên phân tích PSI, nhóm tác giả đề xuất hướng tiếp cận kết hợp giữa đồ thị PSI và mạng nơ-ron tích chập CNN. Tổng quan phương pháp đề xuất được trình bày ở Hình 1, gồm 4 bước sau:

- Sinh đồ thị luồng điều khiển CFG: sử dụng công cụ IDA pro để trích xuất đồ thị CFG. Bởi IDA (Interactive Disassembler) là công cụ phân tách có khả năng thực hiện dịch ngược và tự động phân tích các ứng dụng nhị phân sử dụng tham chiếu chiếu giữa các vùng mã, ngăn xếp API call và các thông tin khác.
- Sinh đồ thị PSI: nhóm tác giả xây dựng công cụ plugin IDA pro để tự động trích xuất đồ thị PSI từ CFG.



Hình 1. Tổng quan mô hình đề xuất

- Tiền xử lý dữ liệu: mục đích bước này nhằm chuyển đổi tất cả định dạng đồ thị PSI thành dạng danh sách kề phù hợp với bộ phân lớp CNN.
- Bộ phân lớp CNN: ở bước này, nhóm tác giả đề xuất một mạng nơ-ron tích chập có chức năng phân loại tập dữ liệu đầu vào là mã độc hay lành tính.

B. Sinh đồ thị PSI

Trong phạm vi khuôn khổ bài báo, nhóm tác giả đưa ra một số định nghĩa sau:

Định nghĩa 1: Đồ thị CFG là một đồ thị có hướng, G = (V, E) trog đó V là tập các đỉnh $\{v_1, v_2, \ldots, v_n\}$ và E là tập các cạnh có hướng $\{e_1, e_2, \ldots, e_m\}$ với $e_{i,j} = (v_i, v_j)$ là cạnh nối từ đỉnh v_i tới đỉnh v_j. Trong đó, mỗi đỉnh v_i biểu diễn bởi một khối mã lệnh cơ bản (basic block) là chuỗi tuyến tính các chỉ thị chương trình với một điểm đầu vào và duy nhất một điểm đầu ra.

Để giải quyết vấn đề các tập lệnh đa kiến trúc trên các thiết bị IoT như ARM, MIPS, PowerPC, SPARC…, công cụ IDA Pro được nhóm tác giả lựa chọn để sinh CFG. Tuy nhiên, đồ thị CFG thu được luôn có cấu trúc phức tạp và sự liên kết giữa các giá trị dạng chuỗi trong các hàm của tập tin nhị phân đầu vào khó quan sát, đồng thời việc áp dụng các kỹ thuật học sâu cũng mất nhiều thời gian. Chính vì vậy, nhóm tác giả sử dụng đồ thị PSI thay vì sử dụng đồ thị CFG.

Định nghĩa 2: Đồ thị PSI là một đồ thị có hướng G (V, E) mà:

- V là tập các đỉnh được xây dựng bởi các phần tử PSI

- E là tập các cạnh biểu diễn sự liên kết giữa các đỉnh trong đồ thị

Thuật toán *1*: PSI-graph generation (CFG)

1: V = [ ], E = [ ]
2: PSI-graph = (V, E)
3: **For** each $node_i$ in CFG **do**
4: **For** each *psi* in $node_i$ **do**
5: V = V ∪ { $node_i$ }
6: **End for**
7: **For** each $node_j$ connect to $node_i$ **do**
8: **For** each *psi* in $node_j$ **do**
9: E = E ∪ { edge ($node_i$,$node_j$) }
10: **End for**
11: **End for**
12: **End for**
13: **Return** PSI-graph

Đồ thị PSI được xây dựng dựa trên tập đỉnh V và cạnh E, trong đó tập đỉnh V gồm các đỉnh được lựa chọn từ đồ thị luồng điều khiển của tập tin nhị phân ELF. Với mỗi đỉnh $node_i$ trong đồ thị CFG, nếu xuất hiện PSI trong $node_i$ thì sẽ đưa đỉnh $node_i$ vào tập V. Sau đó, trong đồ thị CFG sẽ thực hiện tìm kiếm các đỉnh $node_j$ có liên kết với $node_i$. Cạnh liên kết giữa các đỉnh đó sẽ được đưa vào trong tập E. Thuật toán dùng lại khi không tìm được thêm được đỉnh và cạnh nào thỏa mãn nữa.

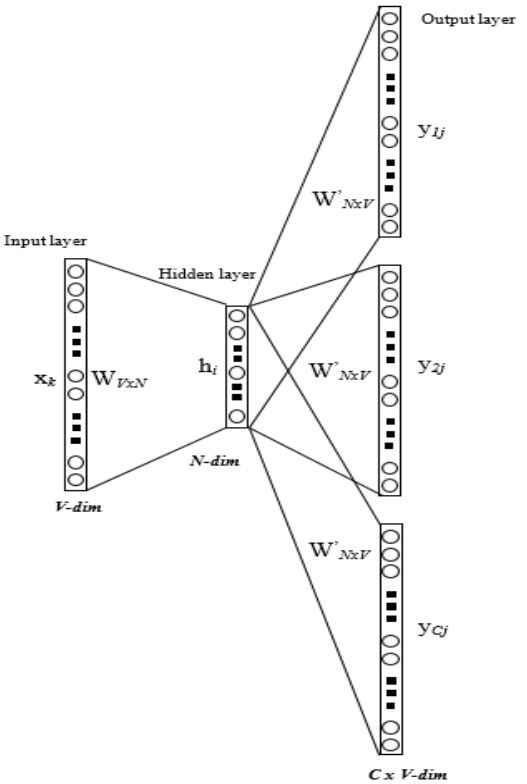
- Sinh đồ thị PSI: PSI là tập các chuỗi có định dạng tường minh và mã hóa. Những chuỗi này phản ánh mục đích của kẻ tấn công và mục tiêu mong muốn bởi chúng thường chứa thông tin quan trọng, ví dụ như "/dev/watchdog; /dev/misc/watchdog" thường xuất hiện trong mã độc Linux.Mirai để nói rằng Botnet đang cố gắng ngăn chặn tiến trình khởi động lại trên thiết bị. Tuy nhiên, hầu hết các chuỗi được trích xuất ra lại bị mã hóa hoặc gây rối. Thuật toán sinh đồ thị PSI được giới thiệu thông qua thuật toán 1.

C. Tiền xử lý và chuẩn hóa dữ liệu

Với dữ liệu là đồ thị PSI thu thập được từ việc phân tích các tệp tin nhị phân nên việc chuyển đổi sang dữ liệu số làm đầu vào cho quá trình huấn luyện với mạng nơ-ron sâu là cần thiết. Các đồ thị PSI là một tập các chuỗi ký tự theo một trật tự nhất định tương ứng với đồ thị thu được. Nhóm tác giả nhận thấy có nhiều điểm tương đồng giữa đồ thị PSI với cấu trúc của một câu văn sử dụng ngôn ngữ tự nhiên. Sự tương đồng này thể hiện qua việc cả hai đều là một tập các chuỗi ký tự và theo một cấu trúc nhất định để mang đến một mục tiêu, ý nghĩa cụ thể. Từ đó, nhóm tác giả sử dụng phương pháp word2vec mà cụ thể là kỹ thuật Skip-gram [20] để chuyển đổi các đồ thị PSI thành các vec tơ số.

Skip-gram là mô hình dự đoán các từ theo từng ngữ cảnh dựa trên các từ mục tiêu phù hợp với đầu vào là các PSI trong các tập tin nhị phân mã độc. Trong bài báo này, nhóm tác giả xây dựng dựa trên ý tưởng xem cả đồ thị như một văn bản và mỗi đồ thị con có gốc xung quanh mỗi đỉnh của đồ thị được xem như các từ xây

dựng lên văn bản và đưa văn bản nhúng vào mạng nơ-ron để học cách biểu diễn toàn bộ đồ thị.



Hình 2. Kiến trúc mô hình skip-gram

Trong Hình 2, đầu vào mô hình là w_i và đầu ra là w_{i-2}, w_{i+1}, w_{i+2} bởi kích thước cửa sổ sử dụng trong bài báo là 2, điều đó do lớp đầu ra phụ thuộc vào kích thước cửa sổ. Đối với cửa sổ kích thước 2 thì sẽ đoán 02 từ bên trái và 02 từ bên phải từ mục tiêu. Do đó mạng sẽ có đầu ra là vector 4 chiều. Kích thước của lớp ẩn tương ứng với V*E trong đó V là kích thước của từ vựng và E là kích thước nhúng.

Công thức tính toán của Skip-gram đưa ra chuỗi các từ w_1, w_2 , ... , w_T với mục đích huấn luyện là tối đa xác xuất logarit trung bình của việc dự đoán các "từ ngữ cảnh" w_{t-C}, … , w_{t+C} xuất hiện gần từ ngữ cảnh w_t được tính như sau:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{|c|\le j \\ j\neq 0}} \log\, p(w_{t+j}|\, w_t) \qquad (1)$$

Trong đó w_t là từ mục tiêu và w_{t+j} là các từ ngữ cảnh trong cửa sổ có kích thước c, $p(w_{t+j}|w_t)$ biểu diễn xác xuất w_{t+j} xuất hiện trong láng giềng của w_t và được tính bởi công thức:

$$p(w_o, w_I) = \frac{\exp\left(v_{w_o}^T v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left(v_{w_o}'^T v_{w_I}\right)} \qquad (2)$$

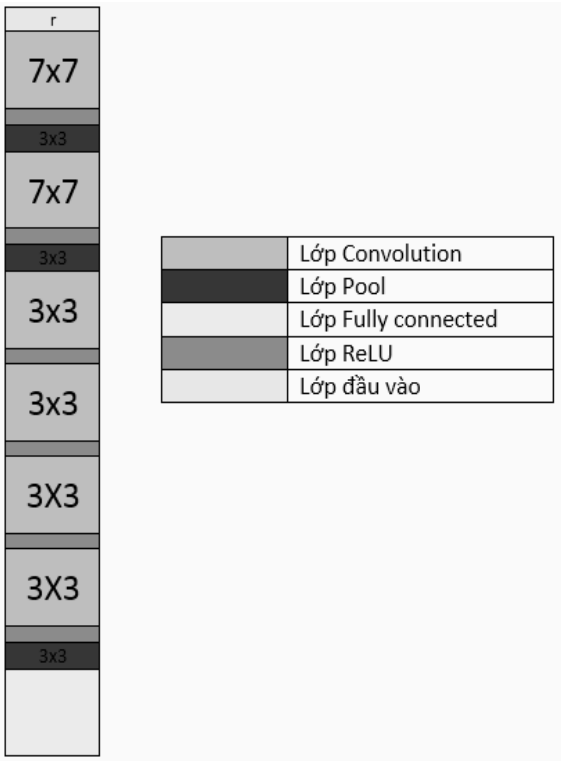
Trong v_{w_o} và v'_{w_o} biểu diễn vector đầu vào và đầu ra của các từ trong từ vựng và W là số lượng từ trong từ vựng.

Bên cạnh đó, mô hình mạng không thể xử lý với đầu vào là các từ hay các PSI vì thế quá trình tiền xử lý tại Hình 1 chính là việc biểu diễn các từ dưới dạng vector. Để thực hiện việc này, nhóm tác giả xây dựng một bộ từ vựng các từ tập huấn luyện (tức là tập các PSI riêng biệt).

D. Kiến trúc mạng nơ-ron

Kiến trúc mạng nơ-ron nhóm tác giả đề xuất dựa trên mạng CNN của [21]. Mô hình mạng gồm 01 lớp đầu vào, 6 lớp ẩn và 01 lớp đầu ra. Trong đó 02 lớp tích chập đầu tiên có kích thước bộ lọc là 7x7 và 4 lớp tích chập còn lại có kích thước bộ lọc là 3x3. Để phân tách các lớp tích chập, ngay sau mỗi lớp tích chập 1D, nhóm tác giả sử dụng hàm ReLU (Rectified Linear Units) thay vì sử dụng hàm tanh hoặc sigmoid vì hàm ReLU có tốc độ xử lý nhanh hơn, có thể giảm độ phức tạp trong tính toán và tránh tình trạng triệt tiêu đạo hàm (vanishing gradien). Ngay sau hàm ReLU của 2 lớp tích chập đầu tiên, nhóm tác giả cũng sử dụng lớp Max Pooling có kích thước 3x3 thay vì các lớp Pooling khác, tức là sẽ thực hiện lấy giá trị lớn nhất trong một phân vùng con hoặc cửa sổ trượt pooling windows, điều này góp phần làm tăng sự phi tuyến bên trong mạng và tạo nên không gian đặc trưng cao cho mỗi đồ thị PSI sẽ tách bạch hơn. Trong phạm vi bài báo này, nhóm tác giả sử dụng hàm mất mát cross-entropy để tối ưu mạng nơ-ron.

Sau khi áp dụng các lớp mạng trên, kết quả thu được là một mảng vector 6 chiều. Để chuyển đổi những vector đó vào một lớp xác xuất thì cần chuyển đổi những vector đó thành một lớp đơn 1 chiều, được gọi là lớp kết nối đầy đủ (fully connected layers). Đầu ra mong muốn sẽ là mã độc hoặc lành tính.



Hình 3. Kiến trúc triển khai mạng Deep Neural Network cho giải pháp đề xuất

III. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Phần này miêu tả cấu hình môi trường và đánh giá kết quả kiểm thử. Để thực nghiệm, nhóm tác giả sử dụng máy tính chip Intel Core i5-850, 3.00 GHz với bộ nhớ RAM 16GB và Nvidia GPU GTX 1070Ti 8GB. Tập dữ liệu phục vụ quá trình huấn luyện gồm 4002 tập tin mã độc thu thập bởi IoTPOT [3] và 6031 tập tin lành tính. Tập dữ liệu mã độc được phân thành 4 nhóm lớn: Linux.Gafgyt.1, Linux.Gafgyt (một biến thể khác của dòng mã độc Linux.Gafgyt), Mirai và Linug.Fgt. Phần còn lại của tập mẫu thuộc về các dòng mã độc tương đối hiếm như Tsunami, Hajime, Light-Aidra [22]. Tập mẫu lành tính được thu thập từ các trang web hoặc trích xuất trực tiếp từ các thiết bị IoT SOHO khác nhau. Trong phạm vi bài báo này, nhóm tác giả chia bộ dữ liệu thực nghiệm thành 2 nhóm: bộ dữ liệu botnet và bộ dữ liệu lành tính để đánh giá hiệu quả của phương pháp đề xuất.

Nhóm tác giả sử dụng Accuracy, Precision, Recall và F1 để đánh giá hiệu quả của phương pháp đề xuất. Chú ý rằng trong phát hiện mã độc thì F1 đôi khi quan trọng hơn Accuracy.

- True Positive (TP): cho biết một tập tin mã độc được định danh chính xác là mã độc.
- True Negative (TN): cho biết một tập tin lành tính được xác định chính xác không phải mã độc.
- False Positive (FP): cho biết một tập tin lành tính bị xác định sai là mã độc .
- False Negative (FN): cho biết tập tin mã độc không được phát hiện và được gán nhãn là lành tính.

Dựa trên các tiêu chí trên, các độ đo sau đây sẽ được sử dụng để xác định tính hiệu quả của hệ thống đã đề xuất.

- Accuracy (ACC): là số lượng mẫu được phát hiện chính xác, chia cho tổng số mẫu mã độc và lành tính.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

- Precision (PR): là tỷ lệ giữa mã độc đã dự đoán và được gán nhãn chính xác là mã độc chia cho tổng số lần gán nhãn chính xác của mẫu mã độc và lành tính.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

- Recall (RC) hoặc tỷ lệ phát hiện là tỷ số giữa mẫu mã độc được dự đoán chính xác với tổng số kết quả của mã độc

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

- F1 score là trọng số trung bình của Precison và Recall

$$\text{F1} = \frac{2 * Recall * Precision}{Recall + Precision} \quad (6)$$

Lưu ý rằng F1 càng gần 1 thì càng tốt.

BẢNG 1. KẾT QUẢ THỬ NGHIỆM VỚI CÁC LỚP TÍCH CHẬP KHÁC NHAU

Số lớp tích chập	Accuracy	Precision	Recall	F1
4	96,7%	96,9%	97,0%	97,1%
5	97,3%	97,7%	97,8%	97,7%
6	98,1%	97,8%	98,5%	98,1%
7	96,6%	97,3%	97,8%	97,5%

So sánh giải pháp đề xuất dựa trên đồ thị PSI với đồ thị luồng điều khiển có thể thấy rằng thời gian huấn luyện tiền xử lý đồ thị CFG có chi phí lớn hơn nhiều so với đồ thị PSI, đồng thời độ đo F1 của PSI cũng lớn hơn so với đồ thị CFG ở mức 98,6%, thông tin cụ thể được cho trong Bảng 2.

BẢNG 2. KẾT QUẢ SO SÁNH GIỮA ĐỒ THỊ PSI VÀ CFG

	Thời gian tiền xử lý graph	**Thời gian training**	**F1-score**
CFG	9 tiếng 30 phút	5 phút	96,4%
PSI Graph *	1 tiếng 25 phút	3 phút	98,6%

IV. KẾT LUẬN

Trong bài báo này, nhóm tác giả đề xuất hướng thu thập đặc trưng của mã độc Botnet trên các thiết bị IoT thông qua việc xây dựng đồ thị PSI. Sau đó, mô hình mạng nơ-ron CNN được sử dụng để cải thiện hiệu quả phân lớp các tập tin mã độc và lành tính. Bằng thực nghiệm, nhóm tác giả đã chứng minh tính hiệu quả của phương pháp đề xuất với độ chính xác (accuracy) và độ đo F1 lên tới 98,1%. Đồng thời, phương pháp tiếp cận theo đồ thị PSI cũng cho kết quả tốt hơn so với đồ thị luồng điều khiển CFG về mặt thời gian. Tuy nhiên, các đặc trưng thu thập để xây dựng đồ thị PSI chủ yếu thông qua phân tích tĩnh và chưa tính đến các khả năng PSI mã hoá. Để cải thiện phương pháp, nhóm tác giả sẽ tiếp tục bổ sung dữ liệu từ nhiều hệ điều hành khác nhau để từ đó nâng cao độ chính xác của phương pháp đề xuất để áp dụng thực tế.

LỜI CẢM ƠN

Nhóm tác giả xin gửi lời cảm ơn đến những góp ý khoa học nghiêm túc, hỗ trợ chuyên môn nhiệt tình của nhóm nghiên cứu MFC500, Học viện An ninh nhân dân. Đồng thời, xin gửi lời chân thành cảm ơn tới nhóm đề tài cấp nhà nước KC01.05 của Học viện Công nghệ Bưu chính viễn thông.

TÀI LIỆU THAM KHẢO

[1]. Pavel Celeda, Radek Krejcí, Jan Vykopal, Martin Drasar, 'Embedded Malware - An Analysis of the Chuck Norris Botnet', presented at the European Conference on Computer Network Defense, Berlin, Germany, 2010.

[2]. Zaddach, Jonas and Bruno, Luca and Francillon, Aurelien and and Balzarotti, Davide, 'AVATAR: A framework to support dynamic security analysis of embedded systems' firmwares', presented at the Proceedings of the Network and Distributed System Security Symposium, France, 2014.

[3]. Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T. and Rossow, C., 'IoTPOT: A Novel Honenypot for Revealing Current IoT Threats', *J. Inf. Process.*, vol. 24, pp. 522–533, May 2016.

[4]. Ahmad Darki, Chun-Yu Chuang, Michalis Faloutsos, Zhiyun Qian, Heng Yin, 'RARE: A Systematic Augmented Router Emulation for Malware Analysis', in *Lecture Notes in Computer Science*, vol. 10771, pp. 60–72, 2018.

[5]. A. Jacobsson, M. Boldt and B. Carlsson, 'A risk analysis of a smart home automation system', *Future Gener. Comput. Syst.*, vol. 56, pp. 719–733, 2016.

[6]. Chun-Jung Wu, Ying Tie, Satoshi Hara, and Kazuki Tamiya, 'IoTProtect: Highly Deployable Whitelist-based Protection for Low-cost Internet-of-Things Devices', *J. Inf. Process.*, vol. 26, pp. 662–672, 2018.

[7]. T. Ronghua, 'An Integrated Malware Detection and Classification System', *MEng Chongqing Univ. BEngChangchun Univ. Sci. Technol.*, vol. Doctor of Philosophy, Aug. 2011.

[8]. Yan Shoshitaishvili, Ruoyu Wang, Christophe Hauser, Christopher Kruegel, Giovanni Vigna, 'Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware', *Yan Shoshitaishvili Ruoyu Wang Christophe Hauser Christopher Kruegel Giovanni Vigna*, pp. 15, 2015.

[9]. D. Davidson, B. Moench, and S. Jha, 'FIE on Firmware, Finding vulnerabilities in embedded systems using symbolic execution', *22nd USENIX Secur. Symp. USENIX*, pp. 16, 2013.

[10]. Rafiqul Islam, Ronghua Tian, Lynn M. Batten, and Steve Versteeg, 'Classification of malware based on integrated static and dynamic features', *J. Netw. Comput. Appl.*, vol. 36, pp. 646–656, 2013.

[11]. A. Costin, J. Zaddach, and A. Francillon, 'A large scale analysis of the security of embedded firmwares', *23rd USENIX Secur. Symp.*, pp. 95–100, 2014.

[12]. Angrishi, Kishore, 'Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets', presented at the arXiv preprint arXiv:1702.03681, 2017.

[13]. Christopher D. McDermott, Farzan Majdani, Andrei V. Petrovski, 'Botnet Detection in the Internet of Things using Deep Learning Approaches', presented at the International joint conference on neural networks 2018, Rio de Janeiro, Brazil.

[14]. Yuan, Z., Lu, Y., Wang, Z., Xue, Y, 'Droid-Sec: deep learning in android malware detection', presented at the ACM SIGCOMM Computer Communication Review, vol. 44, pp. 371–372, 2014.

[15]. Saxe, J., Berlin, K., 'Deep neural network based malware detection using two

dimensional binary program features.', presented at the 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 11–20, 2015.

[16]. Hamed HaddadPajouh, Ali Dehghantanha, Raouf Khayami, Kim-Kwang Raymond Choo, 'A Deep Recurrent Neural Network Based Approach for Internet of Things Malware Threat Hunting', 2018.

[17]. Kishore Angrish, 'Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets', *ArXiv170203681v1 CsNI*, Feb. 2017.

[18]. Michele De Donno, Nicola Dragoni, Alberto Giaretta, Angelo Spognardi, 'Analysis of DDoS-Capable IoT Malwares', in *The Federated Conference on Computer Science and Information Systems*, vol. 11, pp. 807–816, 2017.

[19]. M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and and G. Giacinto, 'Novel feature extraction, selection and fusion for effective malware family classification', presented at the Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, pp. 183–194, 2016.

[20]. Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui and Chen, Yang Liu and Shantanu Jaiswa, 'graph2vec: Learning Distributed Representations of Graphs', presented at the arXiv:1707.05005v1, 2017.

[21]. Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui and Chen, Yang Liu and Shantanu Jaiswa, 'graph2vec: Learning Distributed Representations of Graphs', presented at the arXiv:1707.05005v1, 2017.

[22]. Jiawei Su, Danilo Vasconcellos Vargas, Sanjiva Prasad, Daniele Sgandurra, Yaokai Feng, Kouichi Sakurai, 'Lightweight Classification of IoT Malware based on Image Recognition', *CoRR*, vol. abs/1802.03714, 2018.

[23]. H. HaddadPajouh, A. Dehghantanha, R. Khayami, K.R. Choo, 'A deep Recurrent Neural Network based approach for internet of things malware threat hunting', presented at the Future Generation Computer Systems, 2018.

SƠ LƯỢC VỀ TÁC GIẢ

TS. Ngô Quốc Dũng

Đơn vị công tác: Học viện An ninh nhân dân, Bộ Công an.

Email : quocdung.ngo@gmail.com

Quá trình đào tạo: Nhận bằng Kỹ sư tại Đại học Bách Khoa Nantes năm 2009; Nhận bằng Thạc sĩ tại Đại học Lyon 2 năm 2009; Bảo vệ Tiến sĩ tại Đại học Bách khoa Grenoble, Cộng Hòa Pháp năm 2012.

Hướng nghiên cứu hiện nay: Đảm bảo an toàn, an ninh thông tin trên các thiết bị IoT.

KS. Lê Văn Hoàng

Đơn vị công tác: Công ty AIS.

Email: levanhoang.psa@gmail.com

Quá trình đào tạo: Nhận bằng Kỹ sư Công nghệ và An toàn thông tin, Học viện An ninh nhân dân năm 2017.

Hướng nghiên cứu hiện nay: phân tích phát hiện mã độc trong hệ điều hành Linux và ứng dụng cho thiết bị nhúng.

ThS. Nguyễn Huy Trung

Đơn vị công tác : Học viện An ninh nhân dân, Bộ Công an.

Email: huytrung.nguyen.hvan

@gmail.com

Quá trình đào tạo: Kỹ sư và Thạc sĩ tại Đại học Bách khoa Hà Nội. Hiện là nghiên cứu sinh tại Khoa CNTT – Học viện Khoa học và Công nghệ, Viện Hàn lâm khoa học Việt Nam.

Hướng nghiên cứu hiện nay: phân tích phát hiện mã độc trong các thiết bị IoT và ứng dụng học sâu.