(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0048698**

(51)[2020.01] **G06F 21/00**

(13) **B**

(21) 1-2020-06834

(22) 26/11/2020

(45) 25/07/2025 448

(43) 25/10/2021 403A

(73) 1. Trường Đại học Công nghệ - Đại học Quốc Gia Hà Nội (VN)
E3, 144 Xuân Thủy, Phường Dịch Vọng Hậu, Quận Cầu Giấy, Thành phố Hà Nội
2. Nguyễn Ngọc Hoá (VN)
P311-E3, 144 Xuân Thuỷ, phường Dịch Vọng Hậu, quận Cầu Giấy, Hà Nội

(72) Nguyễn Ngọc Hoá (VN); Lê Việt Hà (VN); Phạm Hải Đăng (VN).

(54) PHƯƠNG PHÁP PHÁT HIỆN ĐOẠN MÃ ĐỘC TRONG MÃ NGUỒN ỨNG DỤNG WEB SỬ DỤNG NGÔN NGỮ PHP

(21) 1-2020-06834

(57) Sáng chế đề cập đến phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP, trong đó phương pháp này sử dụng mô hình mạng nơron tích chập (CNN) để phát hiện các đoạn mã độc (WebShell), phương pháp này bao gồm các bước: (i) sử dụng phương pháp đối sánh mẫu để xây dựng một bộ dữ liệu chứa đoạn mã độc Webshell, (ii) chuyển đổi tệp mã nguồn PHP thành một chuỗi các mã lệnh thực thi PHP và (iii) áp dụng phương pháp học sâu với mô hình mạng nơron tích hợp (CNN) dựa trên các bộ dữ liệu đã chuyển sang vectơ chuỗi mã lệnh để xác định tệp mã nguồn có bị nhúng đoạn mã độc hay không.

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp để phát hiện đoạn mã độc (Webshell) trong mã nguồn ứng dụng Web được xây dựng sử dụng ngôn ngữ PHP. Phương pháp đề xuất dựa trên sự kết hợp giữa phương pháp đối sánh mẫu và phương pháp học sâu sử dụng mạng nơron tích chập.

Tình trạng kỹ thuật của sáng chế

Webshell là đoạn mã lệnh thực thi, được nhúng trong ứng dụng Web, cho phép tạo giao diện để có thể truy cập, thi hành các lệnh/thao tác điều khiển máy chủ Web từ xa, giống như sử dụng công cụ shell tại máy chủ đó. Khi bị nhúng vào các máy chủ Web, các đoạn mã WebShell trở thành mã độc và là một trong những công cụ hữu hiệu để tin tặc sử dụng, tấn công hệ thống thông tin liên quan đến máy chủ Web. Các đoạn mã độc Webshell có thể bị nhúng một cách chủ động thông qua những hành vi bất hợp pháp của nhà phát triển ứng dụng Web, cũng có thể bị nhúng thụ động bởi tin tặc dựa vào việc khai thác lỗ hổng bảo mật máy chủ Web. Để đối phó với những hình thức mất an toàn thông tin đó, cần thiết phải có công cụ cho phép phân tích tính toàn bộ mã nguồn ứng dụng Web để phát hiện và loại bỏ những đoạn mã độc Webshell.

Hiện nay, các công trình nghiên cứu và các sáng chế phục vụ phát hiện đoạn mã độc (Webshell) chủ yếu tập trung ứng dụng những mô hình học máy mới, điển hình là các mô hình học sâu. Tài liệu sáng chế CN107516041A, công bố năm 2017, đề xuất phương pháp phát hiện WebShell sử dụng học sâu dựa trên trích xuất các đặc trưng cấu trúc phân cấp của cây cú pháp trừu tượng. Phương pháp này gồm ba bước: mã nguồn chương trình được phân tích hình thái tạo ra luồng đơn vị từ vựng, luồng đơn vị từ vựng này được phân tích và xây dựng cây cú pháp từ vựng, các

thông tin ngữ nghĩa không liên quan sẽ được lọc bỏ; sau đó, sẽ sinh ra các mẫu dựa vào cây cú pháp từ vựng đã được xây dựng ở bước đầu tiên sử dụng môđun AST_RRNN. Cuối cùng, sáng chế sử dụng phương pháp học sâu Recursive_LSTM để phát hiện các WebShell.

Tài liệu sáng chế CN108833409A, công bố năm 2018, cung cấp phương pháp phát hiện WebShell dựa trên học sâu và học bán giám sát. Trước tiên, phương pháp lấy vectơ văn bản của mẫu bằng cách sử dụng phương pháp kiểm thử Chi-square và học sâu. Sau đó sử dụng phân lớp đơn và học sâu để đào tạo và cải thiện hiệu suất phân lớp. Thêm vào đó, phương pháp được đào tạo và kiểm tra bằng cách sử dụng tập dữ liệu công khai. Kết quả phương pháp này cho tỷ lệ phát hiện WebShell khá chính xác.

Tài liệu sáng chế CN109948340B, nhóm tác giả đã đưa ra phương pháp phát hiện Webshell PHP kết hợp mạng nơron tích chập (CNN) và XGboost để nâng cao tốc độ huấn luyện mô hình. Đầu tiên, các tệp tin PHP được phân tích thành mã thực thi bởi công cụ biên dịch PHP. Sau đó, các mã thực thi này được chuyển đổi thành chuỗi chỉ mục có thể được phát hiện bằng học máy sử dụng phương pháp ánh xạ và tiếp tục thực hiện phân lớp trên chuỗi chỉ mục này sử dụng mô hình mạng nơron tích chập. Đối với các tệp PHP không thể phân tích thành mã thực thi, sáng chế đưa ra phương pháp sử dụng phân tích ngữ nghĩa N-Gram để phân tích tệp tin PHP thành một chuỗi cụm từ khóa tương ứng. Sau đó nhóm các cụm từ này lại thành một mẫu tần suất nhóm các từ khóa để có thể huấn luyện và phát hiện mẫu tần suất nhóm từ khóa sử dụng mô hình XGBoost. Từ đó có thể phát hiện ra các mẫu có chứa mã độc Webshell hay không.

Trong số các nghiên cứu, sáng chế nêu trên, hiện chưa có nghiên cứu nào tập trung vào phương pháp làm sạch các bộ dữ liệu phục vụ quá trình huấn luyện mô hình. Chính vì thế, trong sáng chế này tác giả sẽ đề xuất phương pháp phát hiện đoạn mã độc Webshell dựa trên phương pháp học sâu nhưng áp dụng cả phương

pháp đối sánh mẫu trong quá trình xây dựng bộ dữ liệu huấn luyện để nâng cao hiệu suất phát hiện.

Bản chất kỹ thuật của sáng chế

Cách hiệu quả nhất để đảm bảo an toàn thông tin cho các ứng dụng Web là tìm kiếm và loại bỏ các nguy cơ rủi ro từ mã độc (chẳng hạn Webshell), và lỗ hổng bảo mật (chẳng hạn những đoạn mã dẫn đến lỗi SQL Injection) là phân tích tĩnh, dò quét toàn bộ mã nguồn. Trong sáng chế này, tác giả tập trung vào phương pháp phát hiện đoạn mã độc (Webshell) trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP. Trong đó, sáng chế đề xuất phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP, trong đó phương pháp này sử dụng mô hình mạng nơron tích chập (CNN) để phát hiện các đoạn mã độc (WebShell), phương pháp này bao gồm các bước:

(a) thu thập tập dữ liệu các tệp tin mã nguồn PHP bao gồm các tệp tin lành tính để tạo thành bộ dữ liệu sạch (Benign), các tập tin lành tính này được thu thập từ các nhà cung cấp hệ thống quản lý nội dung CMS thông dụng (như Joomla, Wordpress, Drupal, v.v.), và thu thập tập dữ liệu các tệp chứa các đoạn mã độc 101 để tạo thành bộ dữ liệu các đoạn mã độc (bộ dữ liệu Webshell), các tệp chứa các đoạn mã độc này đã được tập hợp bởi cộng đồng mạng (chẳng hạn như, cộng đồng Github (tennc/WebShell, /b374k/b374k, ...));

(b) làm sạch tập dữ liệu tệp mã nguồn PHP chứa đoạn mã độc bằng cách sử dụng phương pháp đối sánh 102 (chẳng hạn như, theo bộ quy tắc Yara) để loại bỏ các tệp tin lành tính 103 bị lẫn trong bộ dữ liệu các đoạn mã độc với mục tiêu chỉ giữ lại các tệp chứa đoạn mã độc nhằm mục đích giảm nhiễu trong quá trình huấn luyện mô hình mạng nơron tích chập, hoặc để bổ sung vào bộ dữ liệu các đoạn mã độc hại 104 nếu các tệp tin được xác định là độc hại;

(c) chuyển hóa mã nguồn mỗi tệp tin PHP thành dưới dạng không gian vectơ bằng cách chuyển hóa sang dạng mã lệnh thực thi (opcodes); quá trình này được thực hiện thông qua công cụ thông dịch ngôn ngữ PHP để biến tệp PHP thành chuỗi mã lệnh thực thi, sau đó ánh xạ chuỗi lệnh đó thành vectơ chuỗi lệnh tương ứng với mã lệnh thực thi.

(d) loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại 205; trong đó việc loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn do các không gian vectơ này là quá nhỏ đối với một đoạn mã độc; trong đó việc loại bỏ các không gian vectơ đã tồn tại trong bộ dữ liệu để đảm bảo mỗi không gian vectơ trong bộ dữ liệu là duy nhất để giảm thiểu kích thước của tập dữ liệu và tăng tốc độ huấn luyện mô hình mạng nơron tích chập;

(e) huấn luyện mô hình mạng nơron tích chập (CNN) với bộ dữ liệu sạch và bộ dữ liệu các đoạn mã độc đã làm sạch; mô hình này được thiết lập với ba lớp tích chập để học đặc trưng, sử dụng phương pháp hồi quy tuyến tính để phân lớp với thuật toán tối ưu Adam; trong đó:

tầng đầu vào 401 là tập dữ liệu các tệp tin sạch thuộc bộ dữ liệu sạch (Benign) và tệp tin mã độc thuộc bộ dữ liệu các đoạn mã độc (bộ dữ liệu Webshell) sau khi đã loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại;

tầng thứ thứ hai 402 là tầng tích chập được cấu thành từ ba lớp tích chập; tiếp đó, các lớp tích chập này sẽ được ghép nối lại thành một lớp tích chập hợp nhất theo cơ chế nối chuỗi (concatenate), mục đích cuối cùng của tầng thứ hai để thực hiện phát hiện ra những đặc trưng hiệu quả nhất được thể hiện bởi ma trận đặc trưng (feature map).

tầng thứ ba 403 là lớp gộp hay còn gọi là lớp tổng hợp (pooling layer) nó làm giảm số tham số mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh quá mức (overfitting);

tầng thứ thứ tư 404 là tầng phân lớp, tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vectơ chứa xác suất của các đối tượng cần được dự đoán;

(f) phân tích tệp PHP với mô hình mạng nơron tích chập đã được huấn luyện để xác định có chứa đoạn mã độc (WebShell) hay không.

Theo một khía cạnh khác của sáng chế, trong bước chuyển hóa mã nguồn mỗi tệp tin PHP thành dưới dạng không gian vectơ, công cụ thông dịch ngôn ngữ PHP là bộ công cụ cung cấp chức năng kết xuất các tập lệnh PHP 201 ("Provides functionality to dump the internal representation of PHP scripts") (hay còn được gọi là bộ công cụ mở rộng VLD) cho phép biên dịch các tệp tin mã nguồn thành dạng mã lệnh thực thi; trong trường hợp biên dịch không thành công 202 có nghĩa là mã nguồn tệp tin PHP có lỗi, khi đó tệp tin này sẽ bị loại bỏ khỏi tập dữ liệu; trong trường hợp biên dịch thành công 203, mã lệnh thực thi sẽ được chuyển đổi thành dạng không gian vectơ 204 tương ứng.

Theo một khía cạnh khác của sáng chế, trong bước tiến hành huấn luyện mô hình mạng nơron tích chập, tầng thứ thứ hai 402 bao gồm 128 bộ lọc (filters) có kích thước lần lượt là 3, 4, 5 sử dụng hàm kích hoạt phi tuyến ReLU $f(x) = \max(0, x)$ trong đó x là giá trị đầu vào để tạo ra thông tin trừu tượng hơn (Abstract/higher-level) cho các lớp tiếp theo.

Theo một khía cạnh khác của sáng chế, trong bước tiến hành huấn luyện mô hình mạng nơron tích chập, tầng thứ ba 403 sử dụng lớp tổng hợp tối đa (global_max_pooling) với kích thước bằng với kích thước của dữ liệu đầu vào và hệ số sụt giảm (dropout) 0,8 nghĩa là trong quá trình huấn luyện mô hình, với mỗi lần thực hiện cập nhật hệ số ta ngẫu nhiên loại bỏ 80% số lượng nút trong lớp; mục đích chính của tầng này là giảm chiều của tầng trước đó, loại bỏ những đặc trưng không còn cần thiết, thay vào đó giữ lại các đặc trưng đã đủ để phân loại đối tượng.

Theo một khía cạnh khác của sáng chế, trong bước tiến hành huấn luyện mô hình mạng nơron tích chập, sử dụng hàm kích hoạt softmax, là một cách ràng buộc

đầu ra của các mạng nơron phải có tổng bằng 1; qua đó, các giá trị đầu ra của hàm softmax có thể được coi như là một phân phối xác suất của các biến đầu ra hay nói cách khác hàm softmax sẽ chuyển đổi giá trị đầu ra của mạng nơron bằng cách chia cho tổng giá trị; hàm softmax được thể hiện trong công thức sau:

$$\text{softmax}(X)_{ij} = \frac{\exp(X_{ij})}{\sum_k \exp(X_{ik})}$$

hàm mất mát cross entropy, sử dụng để so sánh khoảng cách giữa các giá trị đầu ra của softmax và quá trình biến đổi từng giá trị thành các đặc trưng nhị phân (One-hot encoding), trong đó quá trình biến đổi từng giá trị thành các đặc trưng nhị phân (One-hot encoding) là quá trình biến đổi từng giá trị thành các đặc trưng nhị phân chỉ chứa giá trị 1 hoặc 0, mỗi mẫu trong đặc trưng phân loại sẽ được biến đổi thành một vectơ có kích thước m chỉ với một trong các giá trị là 1 (biểu thị nó là active); cross-entropy là một hàm mất mát và giá trị của nó có thể được cực tiểu hoá; điều này giúp cho một mạng nơron đánh giá được xác suất (độ chắc chắn) của phép dự đoán một mẫu dữ liệu tương ứng với một lớp (class); cross entropy là tổng của các xác suất logarit âm; hàm cross entropy được định nghĩa theo công thức sau:

$$-(y\log(p) + (1-y)\log(1-p))$$

đối với mục tiêu làm mịn các bước của gradien xuống để nó có thể hội tụ nhanh hơn, sử dụng thuật toán tối ưu adam (Adaptive Moment Estimation) thông dụng trong các kiến trúc mạng nơron tích chập; các tham số được thiết lập cho bộ tối ưu này gồm learning_rate=0,05; $\beta 1$=0,9; $\beta 2$=0,999 và ε =1e-08.

Mô tả tóm tắt các hình vẽ kèm theo

Hình 1 là hình vẽ dạng sơ đồ minh hoạ bước làm sạch tập dữ liệu tệp mã nguồn PHP chứa đoạn mã độc bằng phương pháp đối sánh;

Hình 2 là hình vẽ dạng sơ đồ minh hoạ bước chuyển hóa mã nguồn mỗi tập tin PHP thành vectơ chuỗi mã lệnh;

Hình 3 là hình vẽ dạng sơ đồ minh hoạ bước loại bỏ các vectơ có giá trị không phù hợp hoặc bị trùng lắp;

Hình 4 là hình vẽ dạng sơ đồ minh hoạ bước huấn luyện mô hình mạng nơron tích chập bằng tập dữ liệu đã được biểu diễn dưới dạng không gian vectơ;

Hình 5 là hình vẽ dạng sơ đồ minh hoạ bước phân tích tệp PHP bằng mô hình mạng nơron tích chập để phát hiện mã độc;

Hình 6 là hình vẽ dạng sơ đồ minh hoạ phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP theo sáng chế.

Mô tả chi tiết sáng chế

Theo một khía cạnh của sáng chế, sáng chế đề xuất phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP, trong đó phương pháp này sử dụng mô hình mạng nơron tích chập (CNN) để phát hiện các đoạn mã độc (WebShell), phương pháp này bao gồm các bước:

(a) thu thập tập dữ liệu các tệp tin mã nguồn PHP bao gồm các tệp tin lành tính để tạo thành bộ dữ liệu sạch (Benign), các tập tin lành tính này được thu thập từ các nhà cung cấp hệ thống quản lý nội dung CMS thông dụng (như Joomla, Wordpress, Drupal, v.v.), và thu thập tập dữ liệu các tệp chứa các đoạn mã độc 101 để tạo thành bộ dữ liệu các đoạn mã độc (bộ dữ liệu Webshell), các tệp chứa các đoạn mã độc này đã được tập hợp bởi cộng đồng mạng (chẳng hạn như, cộng đồng Github (tennc/WebShell, /b374k/b374k, ...));

(b) làm sạch tập dữ liệu tệp mã nguồn PHP chứa đoạn mã độc bằng cách sử dụng phương pháp đối sánh 102 (chẳng hạn như, theo bộ quy tắc Yara) để loại bỏ các tệp tin lành tính 103 bị lẫn trong bộ dữ liệu các đoạn mã độc với mục tiêu chỉ giữ lại các tệp chứa đoạn mã độc nhằm mục đích giảm nhiễu trong quá

trình huấn luyện mô hình mạng nơron tích chập, hoặc để bổ sung vào bộ dữ liệu các đoạn mã độc hại 104 nếu các tệp tin được xác định là độc hại;

(c) chuyển hóa mã nguồn mỗi tệp tin PHP thành dưới dạng không gian vectơ bằng cách chuyển hóa sang dạng mã lệnh thực thi (opcodes); quá trình này được thực hiện thông qua công cụ thông dịch ngôn ngữ PHP để biến tệp PHP thành chuỗi mã lệnh thực thi, sau đó ánh xạ chuỗi lệnh đó thành vectơ chuỗi lệnh tương ứng với mã lệnh thực thi.

(d) loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại 205; trong đó việc loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn do các không gian vectơ này là quá nhỏ đối với một đoạn mã độc; trong đó việc loại bỏ các không gian vectơ đã tồn tại trong bộ dữ liệu để đảm bảo mỗi không gian vectơ trong bộ dữ liệu là duy nhất để giảm thiểu kích thước của tập dữ liệu và tăng tốc độ huấn luyện mô hình mạng nơron tích chập;

(e) huấn luyện mô hình mạng nơron tích chập (CNN) với bộ dữ liệu sạch và bộ dữ liệu các đoạn mã độc đã làm sạch; mô hình này được thiết lập với ba lớp tích chập để học đặc trưng, sử dụng phương pháp hồi quy tuyến tính để phân lớp với thuật toán tối ưu Adam; trong đó:

tầng đầu vào 401 là tập dữ liệu các tệp tin sạch thuộc bộ dữ liệu sạch (Benign) và tệp tin mã độc thuộc bộ dữ liệu các đoạn mã độc (bộ dữ liệu Webshell) sau khi đã loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại;

tầng thứ thứ hai 402 là tầng tích chập được cấu thành từ ba lớp tích chập; tiếp đó, các lớp tích chập này sẽ được ghép nối lại thành một lớp tích chập hợp nhất theo cơ chế nối chuỗi (concatenate), mục đích cuối cùng của tầng thứ hai để thực hiện phát hiện ra những đặc trưng hiệu quả nhất được thể hiện bởi ma trận đặc trưng (feature map).

tầng thứ ba 403 là lớp gộp hay còn gọi là lớp tổng hợp (pooling layer) nó làm giảm số tham số mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh quá mức (overfitting);

tầng thứ thứ tư 404 là tầng phân lớp, tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vectơ chứa xác suất của các đối tượng cần được dự đoán;

(f) phân tích tệp PHP với mô hình mạng nơron tích chập đã được huấn luyện để xác định có chứa đoạn mã độc (WebShell) hay không.

Trong bản mô tả sáng chế này, "phương pháp lai" được hiểu là phương pháp sử dụng kết hợp phương pháp so khớp mẫu thông qua sử dụng bộ quy tắc yara và phương pháp học sâu với mạng nơron tích chập; "mã độc Webshell" được hiểu là một dạng mã độc có nhiều chức năng được sử dụng bởi tin tặc nhằm mục đích điều khiển, truy cập trái phép từ xa đối với máy chủ ứng dụng Web. Webshell thường được viết bằng nhiều loại ngôn ngữ và thường thì chính là ngôn ngữ mà trang Web đó đang sử dụng; "bộ quy tắc Yara" được hiểu là tập hợp các quy tắc, thường được thể hiện dưới dạng các biểu thức chính quy, có chứa kèm các dấu hiệu để mô tả cách thức nhận diện các đoạn mã độc.

Để xây dựng phương pháp này, tác giả sử dụng hai bộ dữ liệu để huấn luyện mô hình học sâu. Bộ dữ liệu chứa các tệp PHP sạch được gọi tắt là "bộ dữ liệu sạch Benign"; trong khi đó, bộ dữ liệu chứa các tệp PHP có chứa đoạn mã độc Webshell được gọi tắt là "bộ dữ liệu Webshell".

Sau đây giải pháp sẽ được mô tả chi tiết có dựa vào các hình vẽ kèm theo.

Ngôn ngữ PHP lập trình ứng dụng Web hiện được sử dụng rất rộng rãi hiện nay (với thị phần đến 79%), đặc biệt đối với các dịch vụ công trong chính phủ điện tử Việt Nam. Chính vì thế, việc đảm bảo an toàn thông tin cho các ứng dụng Web được xây dựng bằng PHP là cấp thiết hiện nay, đặc biệt cần phải đảm bảo an toàn ngay từ khâu phát triển ứng dụng đến triển khai thực tế (theo mô hình DevSecOps).

Trong sáng chế này, tác giả tập trung xây dựng phương pháp lai để phát hiện đoạn mã độc Webshell trong các tệp mã nguồn PHP.

Phương pháp lai đề xuất được xây dựng với ý tưởng áp dụng phương pháp đối sánh mẫu bằng cách áp dụng bộ quy tắc Yara để làm sạch và xây dựng được bộ dữ liệu Webshell và bộ dữ liệu sạch Benign. Từ đó, tác giả xây dựng mô hình biểu diễn tệp mã nguồn PHP dưới dạng một vectơ chuỗi mã lệnh. Sau đó, các bộ dữ liệu sạch Benign và Webshell sẽ được chuyển hết thành tập các vectơ chuỗi mã lệnh phục vụ huấn luyện mô hình học máy. Tác giả sử dụng phương pháp học sâu với mô hình mạng nơron tích chập gồm ba lớp tích chập phục vụ quá trình học đặc trưng và bộ phân lớp nhị phân dựa vào hồi quy. Từ mô hình đã huấn luyện, tác giả xây dựng quy trình để phát hiện đoạn mã độc Webshell trong tệp tin PHP.

Để xây dựng bộ dữ liệu sạch Benign, tác giả đã tiến hành thu thập các tệp PHP sạch từ các nhà phát triển hệ quản trị nội dung, diễn đàn, framwork, v.v., như Laravel, Wordpress, Joomla, phpMyAdmin, phpPgAdmin, phpbb, v.v.. Sau khi loại bỏ và làm sạch, bộ dữ liệu sạch Benign chứa 7400 tệp PHP.

Để có được bộ dữ liệu Webshell tốt, tác giả đã tiến hành thu thập các tệp PHP chứa Webshell từ một số nguồn tổng hợp tin cậy trên Github. Các tệp PHP cũng cần được làm sạch để loại đi những tệp không thực sự chứa đoạn mã độc Webshell. Lý do cần loại bỏ xuất phát từ việc quá trình huấn luyện mô hình học sâu sẽ bị nhiễu nếu có cả tệp sạch trong bộ dữ liệu Webshell. Đây cũng chính là luận điểm tác giả sẽ yêu cầu bảo hộ với việc làm sạch bộ dữ liệu Webshell bằng phương pháp so khớp mẫu theo bộ quy tắc Yara.

Hình 1 minh họa quá trình sử dụng phương pháp đối sánh theo bộ quy tắc Yara để làm sạch bộ dữ liệu Webshell. Tại bước thu thập tập dữ liệu các tệp chứa các đoạn mã độc 101, tác giả tiến hành thu thập từ các nguồn có số sao được đánh giá cao trên nền tảng chia sẻ mã nguồn Github (*https://github.com/*). Ở bước này, tác giả đã thu thập 4171 tệp nghi có chứa Webshell. Đối với các tệp tin PHP độc hại, mặc dù có thể dễ dàng thu thập được từ các nguồn có độ tin cậy cao trên Github, tuy

nhiên, các nguồn tập hợp này không thể tránh khỏi bị lẫn các tệp tin lành tính, làm ảnh hưởng rất nhiều chất lượng của tập dữ liệu. Vì vậy việc làm sạch các tệp tin lành tính trong tập dữ liệu các tệp tin độc hại là vô cùng cần thiết.

Tại bước làm sạch tập dữ liệu tệp mã nguồn PHP chứa đoạn mã độc bằng cách sử dụng phương pháp đối sánh 102, từ bộ dữ liệu thô này, tác giả sẽ tiến hành sử dụng công cụ yara với tập luật đã được tác giả xây dựng từ công trình "GuruWS: A Hybrid Platform for Detecting Malicious Web Shells and Web Application Vulnerabilities" (*Van-Giap Le, Huu-Tung Nguyen, Duy-Phuc Pham, Van-On Phung, Ngoc-Hoa Nguyen, "GuruWS: A Hybrid Platform for Detecting Malicious Web Shells and Web Application Vulnerabilities", Transactions on Computational Collective Intelligence XXXII. Lecture Notes in Computer Science, vol 11370. Springer, Cham 978-3-319-90286-9, 2019 (ISI WoS, Scopus Journal)*). Bộ quy tắc Yara với đặc điểm là phát hiện các đoạn mã độc bằng phương pháp đối sánh mẫu, nên sẽ rất hiệu quả đối với các mẫu đoạn mã độc đã biết. Quá trình làm sạch được thực hiện với tất cả các tệp PHP thô đã thu thập. Với mỗi tệp PHP, nếu bộ công cụ yara so khớp với tập luật mẫu cho kết quả tệp tin lành tính tại 103, tệp tin đó sẽ không được đưa vào bộ dữ liệu Webshell. Nếu kết quả trả về là tệp tin độc hại tại 104, tác giả sẽ đưa tệp tin PHP vào bộ dữ liệu Webshell.

Từ hai bộ dữ liệu sạch Benign và Webshell, tác giả đã sử dụng phương pháp biểu diễn tệp tin PHP sang vectơ chuỗi mã lệnh. Hình 2 mô tả phương pháp đề xuất biểu diễn mã nguồn PHP dưới dạng không gian vectơ bằng cách chuyển hóa sang dạng mã lệnh OpCode (Operation Code). Đối với bộ thông dịch ngôn ngữ PHP, hiện tập mã lệnh này bao gồm 197 lệnh (*https://www.php.net/manual/en/internals2.opcodes.php*). Ở bước 201, PHP với bộ công cụ mở rộng VLD (*https://pecl.php.net/package/vld*) cho phép biên dịch các tệp tin mã nguồn thành dạng mã lệnh OpCode. Ví dụ dưới đây là mã nguồn tệp tin *test.php*:

```
<?php

function test($a,$b) {
   $c = $a + $b;
   return $c;
```

```
}

$i = 1;
$j = 2;

test($i,$j);
```

Thực thi câu lệnh biên dịch mã nguồn tệp tin *test.php* thành OpCode bằng cách gọi đến thư viện VLD theo cú pháp sau:

```
php -dvld.active=1  -dvld.execute=0 test.php
```

Trong trường hợp biên dịch không thành công 202, có nghĩa là mã nguồn tệp tin test.php có lỗi, khi đó tệp tin này sẽ bị loại bỏ khỏi tập dữ liệu. Trong trường hợp, biên dịch thành công 203, mã lệnh OpCode sẽ được chuyển đổi thành dạng không gian vectơ 204 tương ứng, như trong mô tả hình 3, khi đó ta thu được vectơ chuỗi lệnh là:

[101, 0, 101, 38, 101, 38, 101, 102, 66, 66, 60, 103, 62, 104, 63, 63, 101, 1, 38, 101, 62, 101, 62]

Ở bước loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại 205, loại bỏ các tệp tin mà vectơ chuỗi lệnh của nó có giá trị vô nghĩa hoặc bị trùng lắp trong tập dữ liệu. Những vectơ vô nghĩa là những vectơ khi thực hiện lệnh biên dịch từ mã nguồn thành opcode vì một lý do nào đó gặp lỗi, hoặc có độ dài quá bé đối với một đoạn mã độc (hiện được xác định bằng độ vectơ chuỗi lệnh Webshell ngắn nhất trừ đi một), do đó không có giá trị sử dụng để huấn luyện mô hình học sâu. Bên cạnh đó, cũng sẽ có những trường hợp các nhiều vectơ chuỗi lệnh có giá trị trùng nhau khi đó chỉ cần lưu lại một không gian vectơ duy nhất để giảm thiểu kích thước của tập dữ liệu và tăng tốc độ huấn luyện mô hình.

Sau quá trình chuyển đổi sang không gian vectơ chuỗi mã lệnh và loại bỏ đi những tệp không có giá trị, bộ dữ liệu sạch Benign của tác giả thu được 6057 vectơ; bộ dữ liệu Webshell có 1324 vectơ. Đây là các bộ dữ liệu được tác giả sử dụng để huấn luyện mô hình học sâu.

Hình 4 mô tả kiến trúc của mô hình mạng nơron tích chập CNN để dự đoán Webshell trong mã nguồn ứng dụng, trong đó mô hình được hình thành nên bởi bốn tầng.

Đầu tiên là tầng đầu vào 401, đây chính là tập dữ liệu các tệp tin sạch Benign và tệp tin WebShell dưới dạng các vectơ chuỗi lệnh có kích thước tối đa là 80.000. Giá trị này được lựa chọn do trong bộ dữ liệu Webshell thu thập được, chuỗi lệnh dài nhất trong các tệp tin PHP chứa đoạn mã độc Webshell mới chỉ xấp xỉ 80.000 lệnh.

Tầng thứ thứ hai 402 gọi là tầng tích chập được cấu thành từ ba lớp tích chập gồm 128 bộ lọc (filters) có kích thước lần lượt là 3, 4, 5 sử dụng hàm kích hoạt phi tuyến ReLU $f(x) = \max(0, x)$ trong đó x là giá trị đầu vào để tạo ra thông tin trừu tượng hơn (Abstract/higher-level) cho các lớp tiếp theo. Tiếp đó, các lớp tích chập này sẽ được ghép nối lại thành một lớp tích chập hợp nhất theo cơ chế nối chuỗi (*concatenate*), mục đích cuối cùng của tầng thứ hai để thực hiện phát hiện ra những đặc trưng hiệu quả nhất được thể hiện bởi ma trận đặc trưng (feature map).

Tầng thứ ba 403 là lớp gộp hay còn gọi là lớp tổng hợp (pooling layer) nó làm giảm số tham số mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh quá mức (overfitting). Mô hình đề xuất sử dụng lớp tổng hợp tối đa (global_max_pooling) với kích thước bằng với kích thước của dữ liệu đầu vào và hệ số sụt giảm (dropout) 0,8 nghĩa là trong quá trình huấn luyện mô hình, với mỗi lần thực hiện cập nhật hệ số ta ngẫu nhiên loại bỏ 80% số lượng nút trong lớp. Mục đích chính của tầng này là giảm chiều của tầng trước đó, loại bỏ những đặc trưng không còn cần thiết, thay vào đó giữ lại các đặc trưng đã đủ để phân loại đối tượng.

Tầng thứ thứ tư 404 là tầng phân lớp, tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vectơ chứa xác suất của các đối tượng cần được dự đoán. Tại tầng này, mô hình đề xuất sử dụng các tham số:

Hàm kích hoạt softmax, là một cách ràng buộc đầu ra của các mạng nơron phải có tổng bằng 1. Qua đó, các giá trị đầu ra của hàm softmax có thể được coi như

là một phân phối xác suất của các biến đầu ra hay nói cách khác hàm softmax sẽ chuyển đổi giá trị đầu ra của mạng nơron bằng cách chia cho tổng giá trị. Hàm softmax được thể hiện trong công thức sau

$$softmax(X)_{ij} = \frac{\exp(X_{ij})}{\sum_k \exp(X_{ik})}$$

Hàm mất mát cross entropy, sử dụng để so sánh khoảng cách giữa các giá trị đầu ra của softmax và quá trình biến đổi từng giá trị thành các đặc trưng nhị phân (One-hot encoding), trong đó quá trình biến đổi từng giá trị thành các đặc trưng nhị phân (One-hot encoding) là quá trình biến đổi từng giá trị thành các đặc trưng nhị phân chỉ chứa giá trị 1 hoặc 0, mỗi mẫu trong đặc trưng phân loại sẽ được biến đổi thành một vectơ có kích thước m chỉ với một trong các giá trị là 1 (biểu thị nó là active). Cross-entropy là một hàm mất mát và giá trị của nó có thể được cực tiểu hoá. Điều này giúp cho một mạng nơron đánh giá được xác suất (độ chắc chắn) của phép dự đoán một mẫu dữ liệu tương ứng với một lớp (class). Cross entropy là tổng của các xác suất logarit âm. Hàm cross entropy được định nghĩa theo công thức sau:

$$-(ylog(p) + (1-y)\log(1-p))$$

Đối với mục tiêu làm mịn các bước của gradien xuống để nó có thể hội tụ nhanh hơn, tác giả sử dụng thuật toán tối ưu adam (viết tắt của Adaptive Moment Estimation) thông dụng trong các kiến trúc mạng nơron tích chập. Các tham số được thiết lập cho bộ tối ưu này gồm learning_rate=0,05; β_1=0,9; β_2=0,999 và ε =1e-08.

Trong khuôn khổ sáng chế này, bài toán phân loại tệp tin mã nguồn có hai lớp tương ứng với hai số nhận giá trị trong khoảng từ 0 đến 1, lớp phân loại này sẽ chuyển ma trận đặc trưng của lớp trước thành vectơ có hai chiều thể hiện xác suất tương ứng với tệp tin là sạch hoặc là WebShell.

Hình 5 mô tả quy trình dự đoán một tệp tin mã nguồn PHP có phải là tệp tin WebShell hay không bằng cách sử dụng mô hình mạng nơron tích chập đã được sáng kiến đề xuất xây dựng. Tương tự như quy trình xây dựng tập dữ liệu để huấn luyện mô hình, tệp tin mã nguồn sẽ được thông dịch thành chuỗi lệnh sử dụng thư viện VLD 501 nếu quá trình thông dịch không thành công 502, có nghĩa là tệp tin này không có khả năng thực thi được nó sẽ được phân loại là tệp tin lành tính. Trong trường hợp biên dịch thành công thành chuỗi lệnh OpCode 503, mô hình đề xuất sẽ biểu diễn chuỗi lệnh này dưới dạng vectơ chỉ mục, nếu vectơ chỉ mục này không có giá trị hoặc giá trị vô nghĩa, tệp tin sẽ là lành tính 504. Ngược lại, vectơ chỉ mục này sẽ được sử dụng làm đầu vào để mô hình CNN đưa ra dự đoán, nếu giá trị dự đoán agrmax của mô hình bằng 0 thì tệp tin là lành tính, bằng 1 thì tệp tin có chứa đoạn mã độc Webshell.

Hình 6 là hình vẽ dạng sơ đồ minh hoạ phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP theo sáng chế.

Với bộ dữ liệu sạch Benign và Webshell đã làm sạch nêu trên, sau khi huấn luyện mô hình CNN với các tham số nêu trên, phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP được đề xuất trong sáng chế đã đạt tỷ lệ chính xác đến 99,02%; tỷ lệ phát hiện nhầm rất nhỏ, chỉ khoảng 0,85%.

YÊU CẦU BẢO HỘ

1. Phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web sử dụng ngôn ngữ PHP, trong đó phương pháp này sử dụng mô hình mạng nơron tích chập (CNN) để phát hiện các đoạn mã độc (WebShell), phương pháp này bao gồm các bước:

(a) thu thập tập dữ liệu các tệp tin mã nguồn PHP bao gồm các tệp tin lành tính để tạo thành bộ dữ liệu sạch (Benign), các tập tin lành tính này được thu thập từ các nhà cung cấp hệ thống quản lý nội dung CMS thông dụng (như Joomla, Wordpress, Drupal, v.v.), và thu thập tập dữ liệu các tệp chứa các đoạn mã độc (101) để tạo thành bộ dữ liệu các đoạn mã độc (bộ dữ liệu Webshell), các tệp chứa các đoạn mã độc này đã được tập hợp bởi cộng đồng mạng (chẳng hạn như, cộng đồng Github (tennc/WebShell, /b374k/b374k, ...));

(b) làm sạch tập dữ liệu tệp mã nguồn PHP chứa đoạn mã độc bằng cách sử dụng phương pháp đối sánh (102) (chẳng hạn như, theo bộ quy tắc Yara) để loại bỏ các tệp tin lành tính (103) bị lẫn trong bộ dữ liệu các đoạn mã độc với mục tiêu chỉ giữ lại các tệp chứa đoạn mã độc nhằm mục đích giảm nhiễu trong quá trình huấn luyện mô hình mạng nơron tích chập, hoặc để bổ sung vào bộ dữ liệu các đoạn mã độc hại (104) nếu các tệp tin được xác định là độc hại;

(c) chuyển hóa mã nguồn mỗi tệp tin PHP thành dưới dạng không gian vectơ bằng cách chuyển hóa sang dạng mã lệnh thực thi (opcodes); quá trình này được thực hiện thông qua công cụ thông dịch ngôn ngữ PHP để biến tệp PHP thành chuỗi mã lệnh thực thi, sau đó ánh xạ chuỗi lệnh đó thành vectơ chuỗi lệnh tương ứng với mã lệnh thực thi;

(d) loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại (205); trong đó việc loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn do các không gian vectơ này là quá nhỏ đối với một đoạn mã độc; trong đó việc loại bỏ các không gian vectơ đã tồn tại trong bộ dữ liệu để đảm bảo mỗi không gian vectơ trong bộ dữ liệu là duy

nhất để giảm thiểu kích thước của tập dữ liệu và tăng tốc độ huấn luyện mô hình mạng nơron tích chập;

(e) huấn luyện mô hình mạng nơron tích chập (CNN) với bộ dữ liệu sạch và bộ dữ liệu các đoạn mã độc đã làm sạch; mô hình này được thiết lập với ba lớp tích chập để học đặc trưng, sử dụng phương pháp hồi quy tuyến tính để phân lớp với thuật toán tối ưu Adam; trong đó:

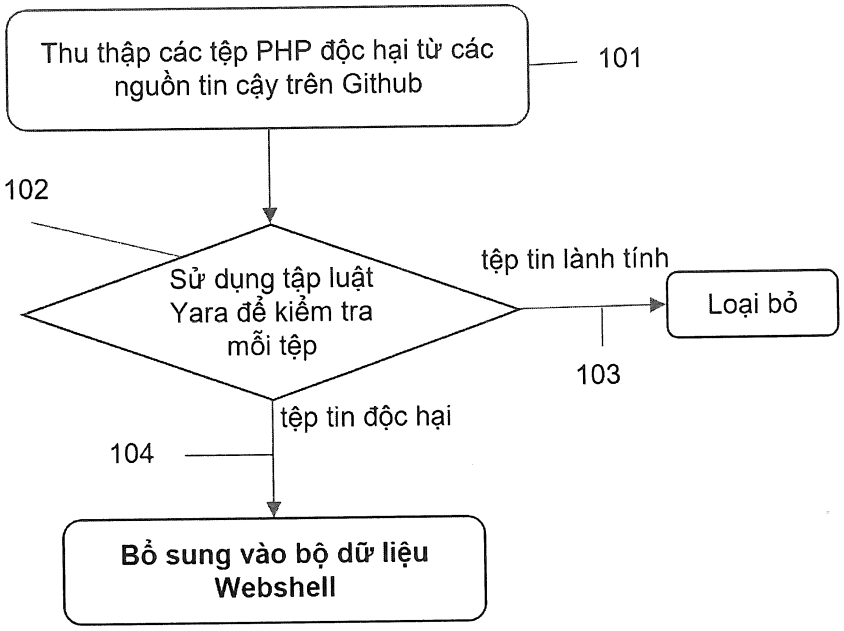
tầng đầu vào (401) là tập dữ liệu các tệp tin sạch thuộc bộ dữ liệu sạch (Benign) và tệp tin mã độc thuộc bộ dữ liệu các đoạn mã độc (bộ dữ liệu Webshell) sau khi đã loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại;

tầng thứ thứ hai (402) là tầng tích chập được cấu thành từ ba lớp tích chập; tiếp đó, các lớp tích chập này sẽ được ghép nối lại thành một lớp tích chập hợp nhất theo cơ chế nối chuỗi (concatenate), mục đích cuối cùng của tầng thứ hai để thực hiện phát hiện ra những đặc trưng hiệu quả nhất được thể hiện bởi ma trận đặc trưng (feature map);

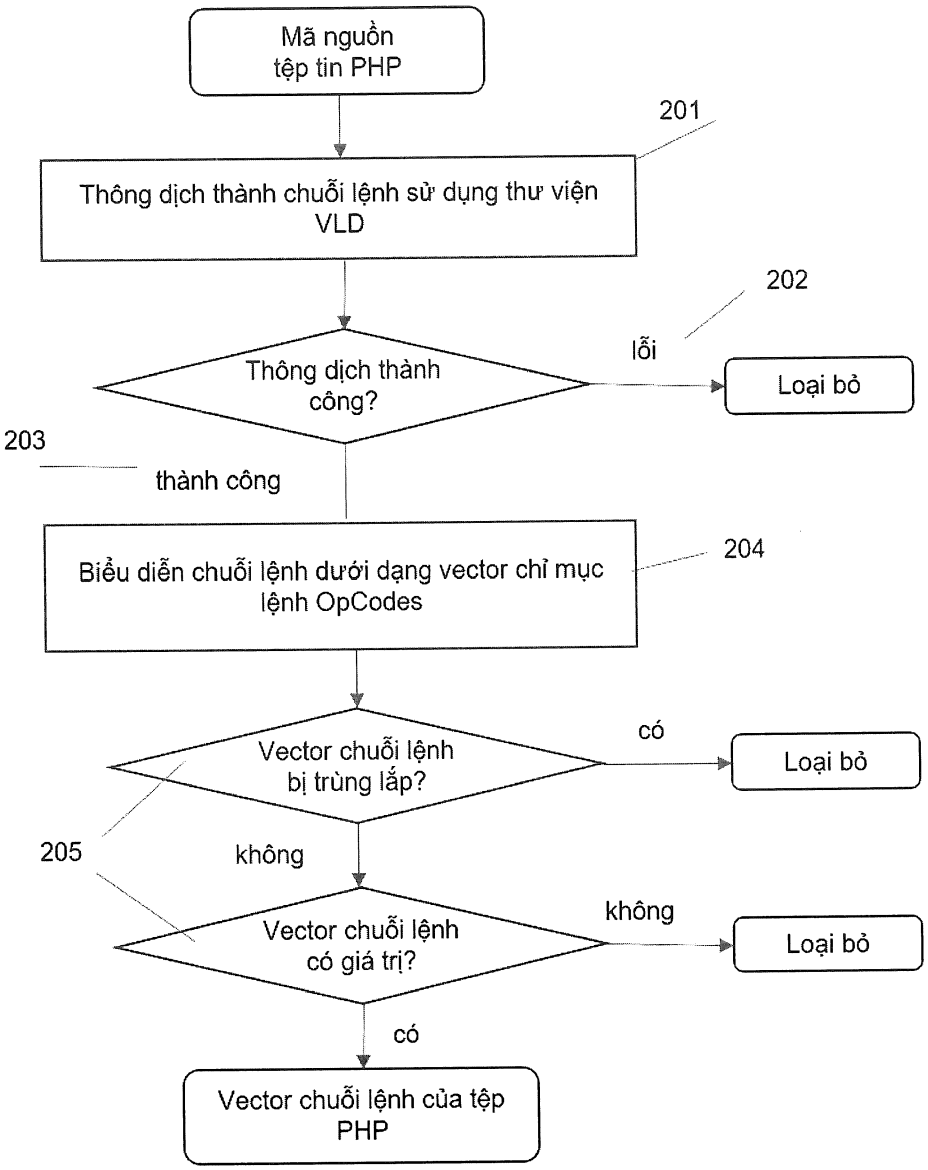
tầng thứ ba (403) là lớp gộp hay còn gọi là lớp tổng hợp (pooling layer) nó làm giảm số tham số mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh quá mức (overfitting);

tầng thứ thứ tư (404) là tầng phân lớp, tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vectơ chứa xác suất của các đối tượng cần được dự đoán;

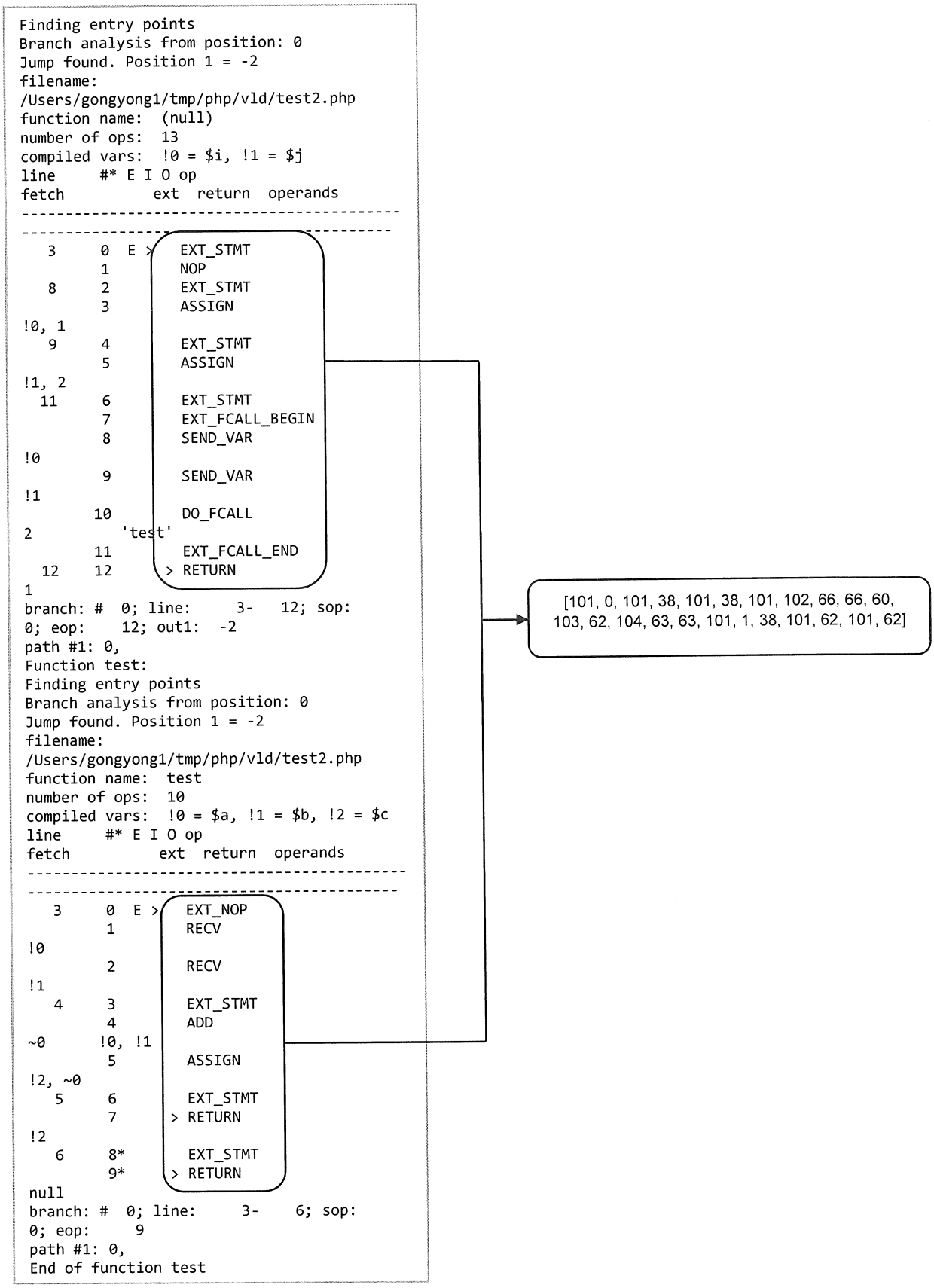
(f) phân tích tệp PHP với mô hình mạng nơron tích chập đã được huấn luyện để xác định có chứa đoạn mã độc (WebShell) hay không.


Thu thập các tệp PHP độc hại từ các nguồn tin cậy trên Github
101
102
Sử dụng tập luật Yara để kiểm tra mỗi tệp
tệp tin lành tính
Loại bỏ
103
tệp tin độc hại
104
Bổ sung vào bộ dữ liệu Webshell

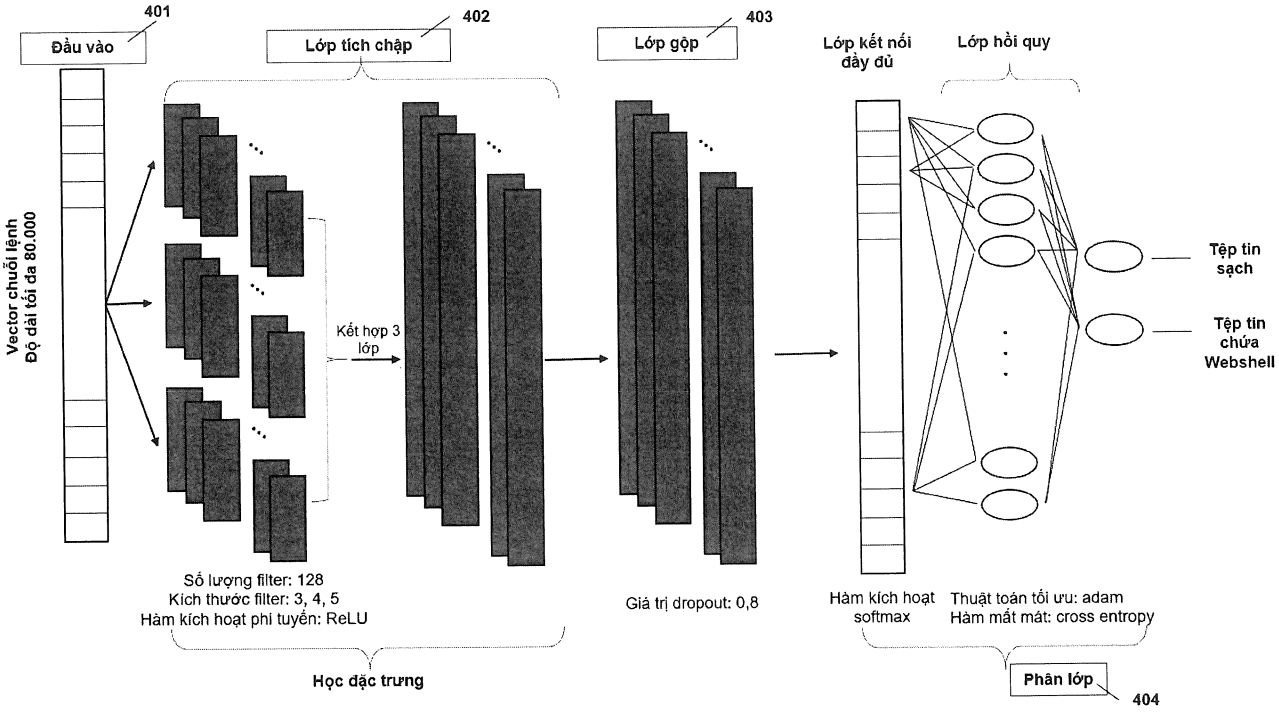
HÌNH 1


Mã nguồn
tệp tin PHP
201
Thông dịch thành chuỗi lệnh sử dụng thư viện VLD
202
lỗi
Thông dịch thành công?
Loại bỏ
203
thành công
Biểu diễn chuỗi lệnh dưới dạng vector chỉ mục lệnh OpCodes
204
có
Vector chuỗi lệnh bị trùng lắp?
Loại bỏ
205
không
không
Vector chuỗi lệnh có giá trị?
Loại bỏ
có
Vector chuỗi lệnh của tệp PHP

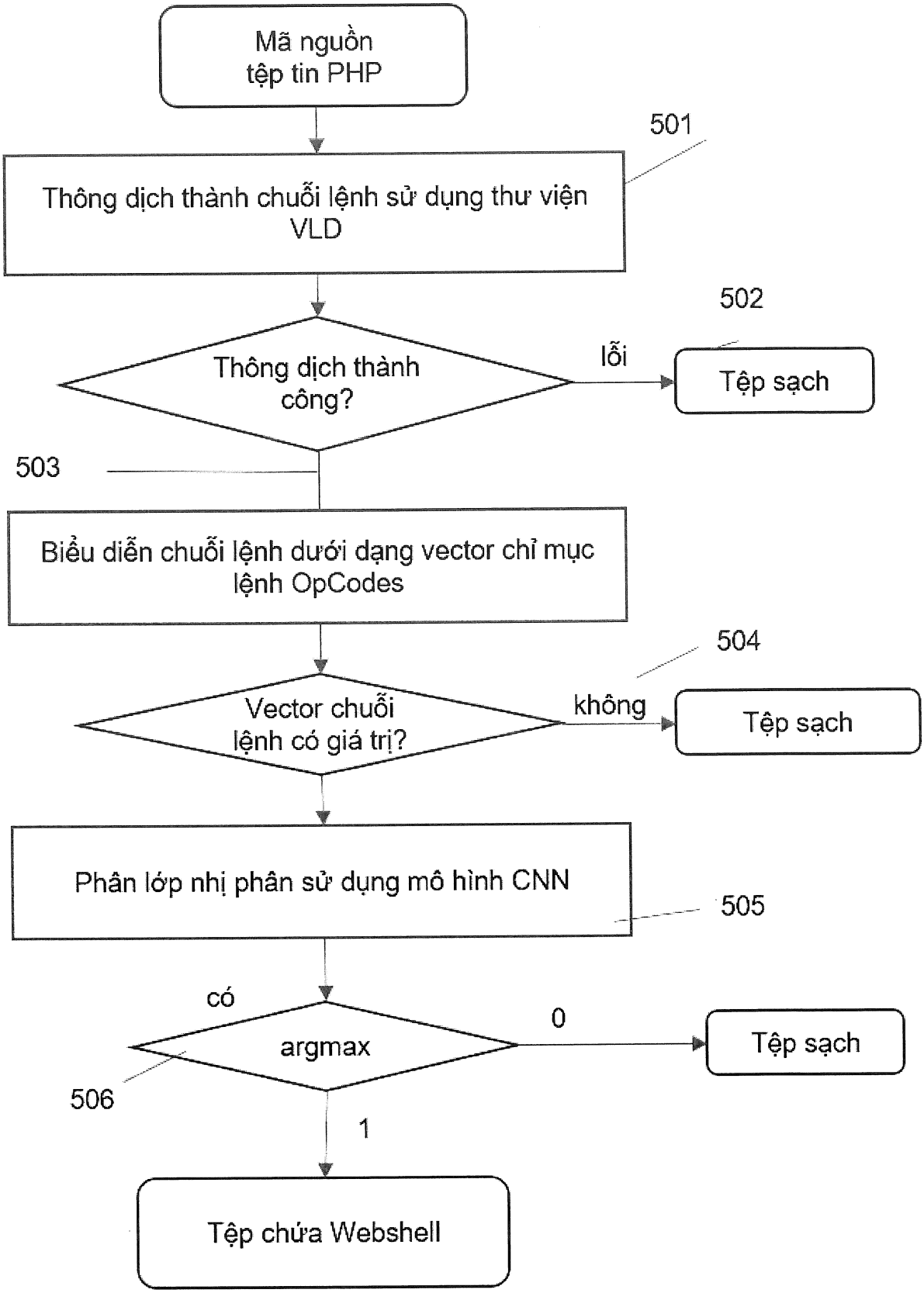
HÌNH 2


Finding entry points
Branch analysis from position: 0
Jump found. Position 1 = -2
filename:
/Users/gongyong1/tmp/php/vld/test2.php
function name: (null)
number of ops: 13
compiled vars: !0 = $i, !1 = $j
line #* E I O op
fetch ext return operands
3 0 E > EXT_STMT
1 NOP
8 2 EXT_STMT
3 ASSIGN
!0, 1
9 4 EXT_STMT
5 ASSIGN
!1, 2
11 6 EXT_STMT
7 EXT_FCALL_BEGIN
8 SEND_VAR
!0
9 SEND_VAR
!1
10 DO_FCALL
2 'test'
11 EXT_FCALL_END
12 12 > RETURN
1
branch: # 0; line: 3- 12; sop:
0; eop: 12; out1: -2
path #1: 0,
Function test:
Finding entry points
Branch analysis from position: 0
Jump found. Position 1 = -2
filename:
/Users/gongyong1/tmp/php/vld/test2.php
function name: test
number of ops: 10
compiled vars: !0 = $a, !1 = $b, !2 = $c
line #* E I O op
fetch ext return operands
3 0 E > EXT_NOP
1 RECV
!0
2 RECV
!1
4 3 EXT_STMT
4 ADD
~0 !0, !1
5 ASSIGN
!2, ~0
5 6 EXT_STMT
7 > RETURN
!2
6 8* EXT_STMT
9* > RETURN
null
branch: # 0; line: 3- 6; sop:
0; eop: 9
path #1: 0,
End of function test
[101, 0, 101, 38, 101, 38, 101, 102, 66, 66, 60, 103, 62, 104, 63, 63, 101, 1, 38, 101, 62, 101, 62]

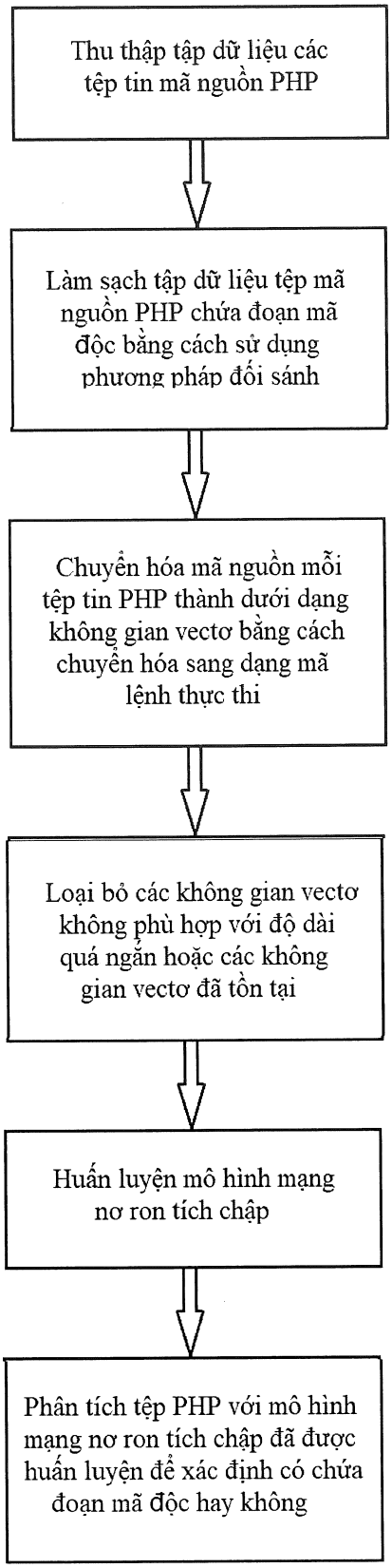
HÌNH 3


401
Đầu vào
402
Lớp tích chập
403
Lớp gộp
Lớp kết nối đầy đủ
Lớp hồi quy
Vector chuỗi lệnh
Độ dài tối đa 80.000
Kết hợp 3 lớp
Tệp tin sạch
Tệp tin chứa Webshell
Số lượng filter: 128
Kích thước filter: 3, 4, 5
Hàm kích hoạt phi tuyến: ReLU
Giá trị dropout: 0,8
Hàm kích hoạt softmax
Thuật toán tối ưu: adam
Hàm mất mát: cross entropy
Học đặc trưng
Phân lớp
404

HÌNH 4


Mã nguồn
tệp tin PHP
501
Thông dịch thành chuỗi lệnh sử dụng thư viện
VLD
502
Thông dịch thành
công?
lỗi
Tệp sạch
503
Biểu diễn chuỗi lệnh dưới dạng vector chỉ mục
lệnh OpCodes
504
Vector chuỗi
lệnh có giá trị?
không
Tệp sạch
Phân lớp nhị phân sử dụng mô hình CNN
505
có
argmax
0
Tệp sạch
506
1
Tệp chứa Webshell

HÌNH 5


Thu thập tập dữ liệu các tệp tin mã nguồn PHP
Làm sạch tập dữ liệu tệp mã nguồn PHP chứa đoạn mã độc bằng cách sử dụng phương pháp đối sánh
Chuyển hóa mã nguồn mỗi tệp tin PHP thành dưới dạng không gian vectơ bằng cách chuyển hóa sang dạng mã lệnh thực thi
Loại bỏ các không gian vectơ không phù hợp với độ dài quá ngắn hoặc các không gian vectơ đã tồn tại
Huấn luyện mô hình mạng nơ ron tích chập
Phân tích tệp PHP với mô hình mạng nơ ron tích chập đã được huấn luyện để xác định có chứa đoạn mã độc hay không

HÌNH 6