Giải pháp đánh giá và quản lý rủi ro an toàn thông tin trong Chính phủ điện tử

Phùng Văn Ổn, Lê Việt Hà, Nguyễn Ngọc Hóa

***Tóm tắt*—Bài báo này trình bày kết quả nghiên cứu xây dựng giải pháp đánh giá và quản lý rủi ro an toàn hệ thống thông tin trong Chính phủ điện tử. Trong bài toán này, chúng tôi tập chung vào xây dựng (i) quy trình đánh giá, quản lý rủi ro an toàn thông tin (ATTT), và (ii) hệ thống phần mềm UET.SRA (Security Risk Assessment System) hỗ trợ đánh giá, quản lý rủi ro theo quy trình đã xây dựng. Việc quản lý và đánh giá rủi ro ATTT được kết hợp theo các tiêu chuẩn trong nước và quốc tế bao gồm các quy trình trong ISO/IEC 27005:2011 và NIST SP 800-39, nhưng được tuỳ biến để phù hợp với thực tiễn của các cơ quan chính phủ. Hệ thống phần mềm UET.SRA đánh giá rủi ro ATTT dựa theo phương pháp kiểm tra các lỗ hổng và sự phơi nhiễm phổ biến (Common Vulnerabilities and Exposures – CVE); việc ước lượng rủi ro định lượng theo Hệ thống chấm điểm lỗ hổng phổ biến (Common Vulnerability Scoring System - CVSS) và Dự án mở về bảo mật ứng dụng web (Open Web Application Security Project - OWASP). Ngoài ra, UET.SRA còn cung cấp chức năng phân tích, phát hiện các lỗ hổng, các đoạn mã độc trong mã nguồn ứng dụng Web sử dụng công nghệ học sâu (deep learning). Kết quả thử nghiệm giải pháp UET.SRA tại Bộ Tài nguyên và Môi trường (TN&MT) bước đầu đã minh chứng được ý nghĩa thực tiễn và cho phép quản lý được các rủi ro ATTT đối với một số hệ thống trọng yếu của Bộ TN&MT.**

***Abstract*—This article presents the results of building a solution to access and manage security risks for the e-Government information system. We focus on building a process and software system UET.SRA to manage and assess security risks. The process was developed using a combination of international and domestic standards including ISO/IEC 27005:2011 and NIST SP 800-39, but customized to match the practice of government agencies. UET.SRA evaluates security risks based on CVEs vulnerability testing; quantitative risk based on CVSS and OWASP standards. In addition, UET.SRA also provides the function of detecting vulnerabilities and webshell in the source code of web applications using deep learning algorithms. The experimental results of UET.SRA at the Ministry of Natural Resources and Environment have initially demonstrated practical effectiveness in managing security risks for a number of critical systems.**

Từ khóa—quản lý rủi ro, đánh giá rủi ro, dò quét lỗ hổng hệ thống, dò quét mã nguồn.

Keywords—security risk management, security risk assessment, vulnerable scan, source code scan.

I. ĐẶT VẤN ĐỀ

Đánh giá, quản lý rủi ro ATTT là một yêu cầu quan trọng trong việc đảm bảo an toàn các hệ thống công nghệ thông tin (CNTT) nói chung và trong Chính phủ điện tử nói riêng. Trên thị trường đã có một số sản phẩm hỗ trợ giải quyết vấn đề trên nhưng những sản phẩm đó chưa thực sự phù hợp cho các cơ quan chính phủ, nơi mà giá trị dữ liệu vượt ra ngoài phạm vi của các hợp đồng dân sự. Điều này đặt ra nhu cầu cấp thiết phải làm chủ công nghệ để phát triển các sản phẩm ATTT, tích hợp hệ thống hỗ trợ đánh giá, quản lý rủi ro ATTT cho các hệ thống CNTT nói chung và Chính phủ điện tử của Việt Nam nói riêng.

Trong bài toán này, yêu cầu đặt ra gồm xây dựng quy trình phục vụ cho việc đánh giá, quản lý rủi ro ATTT; xây dựng giải pháp kỹ thuật xác định các rủi ro an toàn trong các hệ thống CNTT và thiết kế, xây dựng hệ thống phần mềm đảm nhiệm vai trò tương ứng.

Việc xác định các rủi ro ATTT trong nghiên cứu này được giới hạn trong phạm vi:

Bài báo được nhận ngày 14/4/2021. Bài báo được nhận xét bởi phản biện thứ nhất ngày 30/5/2021 và được chấp nhận đăng ngày 27/10/2021. Bài báo được nhận xét bởi phản biện thứ hai ngày 13/6/2021 và được chấp nhận đăng ngày 20/6/2021.

- Xác định các lỗ hổng bảo mật trong các hệ thống CNTT; các lỗ hổng trong mã nguồn ứng dụng Web.
- Xác định các thiết lập chính sách của hệ điều hành Windows (Windows policy) chưa tuân thủ theo những quy định đảm bảo ATTT.
- Xác định các rủi ro do việc chưa cập nhật các bản vá lỗ hổng (patch) của hệ điều hành Windows và một số hệ thống thông tin quan trọng.
- Việc dò quét mã nguồn được giới hạn trong việc nghiên cứu các ứng dụng Web sử dụng hai ngôn ngữ lập trình phổ biến là ASP và PHP.

Bài báo được tổ chức như sau: Phần II tóm lược một số khái niệm, phương pháp và kỹ thuật sử dụng trong quản lý rủi ro ATTT nói chung. Phần III trình bày giải pháp đánh giá, quản lý rủi ro ATTT mà chúng tôi đã xây dựng trong khuôn khổ đề tài cấp nhà nước KC.01.19/16-20; Phần IV tổng hợp kết quả thử nghiệm giải pháp tại Bộ TN&MT. Phần cuối tóm lược kết quả và một số hướng phát triển kế tiếp.

II. Phương pháp, kỹ thuật phục vụ đánh giá, quản lý rủi ro ATTT

Trong mục này, chúng tôi tóm lược những phương pháp, kỹ thuật được sử dụng trong hoạt động đánh giá, quản lý rủi ro ATTT.

A. Phương pháp phát hiện nguy cơ, lỗ hổng bảo mật dựa trên mẫu thử lỗ hổng

Một trong những kỹ thuật hiện được các hãng bảo mật sử dụng để dò quét, xác định lỗ hổng là sử dụng các mẫu dò quét thử lỗ hổng NVT (Network Vulnerability Test) [1]-[2]. Trong kỹ thuật này, mỗi mẫu thử lỗ hổng NVT sẽ được viết bằng ngôn ngữ mô tả kịch bản tấn công NASL (Network Attack Scripting Language). Đây là ngôn ngữ kịch bản được sử dụng bởi các bộ dò quét lỗ hổng như Nessus và OpenVAS. Hiện đã có hàng chục nghìn plugin được viết bằng NASL cho Nessus và OpenVAS [3]-[5].

B. Phương pháp xác định chính sách Windows vi phạm quy định

Để phát hiện các chính sách Windows vi phạm quy định, cần phải dò quét qua mạng hoặc trực tiếp để thu thập thông tin về các chính sách đang được thiết lập trên máy tính. Các chính sách thu thập được so sánh với các yêu cầu tuân thủ, qua đó, đưa ra các báo cáo, cảnh báo về tình trạng tuân thủ chính sách của máy tính [6].

C. Phương pháp xác định bản vá chưa được áp dụng

Dò quét các bản vá chưa được áp dụng có mục đích dò quét các máy tính trong hệ thống mạng để thu thập thông tin về các bản vá của hệ điều hành được phân loại thành bản vá nghiêm trọng và tùy chọn. Các bản vá nghiêm trọng được khuyến nghị cài đặt sớm nhất có thể cho máy chủ để vá các lỗ hổng bảo mật, lỗi liên quan đến hệ điều hành, sau đó mới đến các thiết bị còn lại. Các bản vá tùy chọn là các bản vá liên quan đến nâng cấp tính năng, điều chỉnh hiệu năng hệ điều hành, sửa các lỗi nhỏ không phổ biến. Việc cài đặt các bản vá loại này tùy thuộc vào yêu cầu của tổ chức [7].

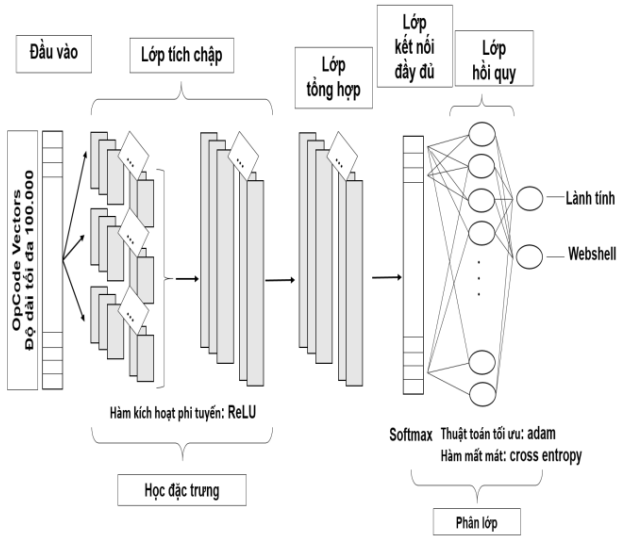
D. Phương pháp xác định lỗ hổng Web

Để phát hiện lỗ hổng ứng dụng Web, chúng tôi đã đánh giá và lựa chọn khung W3af (Web Application Attack and Audit Framework). Hiện W3af được đánh giá là một trong những khung đánh giá lỗ hổng bảo mật ứng dụng Web tốt nhất. Khung đánh giá này cung cấp công cụ khai thác và quét lỗ hổng cho các ứng dụng Web, từ đó cung cấp thông tin về các lỗ hổng bảo mật để sử dụng trong các cuộc thử nghiệm thâm nhập [8].

E. Phương pháp phát hiện đoạn mã độc trong mã nguồn ứng dụng Web

Trong phương pháp này, chúng tôi tập trung vào hai loại ngôn ngữ lập trình được sử dụng phổ biến nhất để phát triển các ứng dụng Web hiện nay là PHP và ASP [9]-[11].

Chúng tôi sử dụng mô hình học sâu với mạng nơ-ron tích chập (Convolutional Neural Network – CNN) được minh họa như Hình 1 dưới đây.



Hình 1. Mô hình mạng CNN ứng dụng trong phát hiện đoạn mã độc.

F. Phương pháp đánh giá rủi ro ATTT tổng thể

Các kết quả dò quét lỗ hổng ATTT có được thông qua việc triệu gọi các công cụ dò quét lỗ hổng bảo mật sẽ được gắn kèm thông tin cụ thể về mức độ nghiêm trọng cũng như nguyên nhân dẫn đến lỗ hổng đó. Các thông tin này cơ bản đã được cộng đồng nghiên cứu tập hợp và tổng hợp thành các bộ dữ liệu về CVE và Danh sách nền tảng hệ thống thông dụng (Common Platform Enumeration – CPE).

Toàn bộ các dữ liệu liên quan đến các thông tin này được thu thập thông qua việc làm giàu dữ liệu về các lỗ hổng bảo mật đã phát hiện từ cộng đồng và những nguồn dữ liệu CVE, CPE, NVT uy tín trên thế giới. Hiện trong khuôn khổ đề tài, các nguồn dữ liệu được chúng tôi khai thác đến từ các tổ chức sau:

- Cơ sở dữ liệu lỗ hổng quốc gia của viện NIST, Mỹ: https://nvd.nist.gov/products.
- Cơ sở dữ liệu lỗ hổng của cộng đồng CVE: https://cve.mitre.org/index.html.
- Cơ sở dữ liệu lỗ hổng từ cộng đồng CVE Detail: https://www.cvedetails.com.

Mức độ rủi ro của hệ thống CNTT sẽ được xác định dựa vào độ rủi ro cao nhất trong danh mục lỗ hổng bảo mật được phát hiện.

Ngoài việc sử dụng các dữ liệu trên, đối với các hệ thống mà không thể dò quét bằng các công cụ tự động, hệ thống cũng cho phép đánh giá định tính để ước lượng rủi ro. Khi đó, trước hết nên xác định tất cả các tiêu chí đo lường rủi ro bảo mật. Đối với hệ thống thông tin của Chính phủ điện tử Việt Nam, các lĩnh vực tác động bao gồm Danh tiếng/Tín nhiệm, Hoạt động sản xuất, An toàn và sức khỏe, Tài chính, Chi phí và các hình phạt pháp lý. Trong bối cảnh của chúng tôi, các lĩnh vực tác động được ưu tiên theo quy mô như trong Bảng 1.

BẢNG 1. ĐỘ ƯU TIÊN ĐÁNH GIÁ MỨC ĐỘ ẢNH HƯỞNG BỞI RỦI RO TRONG CPĐT

Miền ảnh hưởng	Độ ưu tiên
Danh tiếng và lòng tin của người dân	5
Hoạt động, sản xuất	4
An toàn và sức khỏe	3
Tài chính	2
Chi phí và các hình phạt pháp lý	1

Các khu vực tác động này rõ ràng sẽ được thiết lập tùy thuộc vào bối cảnh hiện tại của một bộ hoặc một tỉnh trong Chính phủ điện tử.

Về mặt kỹ thuật, chúng tôi xác định các mối đe dọa sau có thể ảnh hưởng đến hệ thống thông tin:

- Mã độc: Sử dụng mã để thực hiện tiết lộ, điều chỉnh hoặc phá hủy trái phép.
- Thu thập thông tin: Thu thập dữ liệu/thông tin của hệ thống quan trọng.
- Xâm nhập: Truy cập trái phép vào dữ liệu, hệ thống, tài liệu vật lý hoặc cơ sở vật chất.
- Tính khả dụng: Không có sẵn hệ thống, con người, tài liệu vật lý hoặc cơ sở vật chất.
- Bảo mật nội dung thông tin: Điều chỉnh nội dung thông tin trái phép.
- Gian lận: Gian lận là sự lừa dối có chủ ý để đạt được lợi ích không công bằng hoặc bất hợp pháp, hoặc tước bỏ quyền hợp pháp của nạn nhân.

Sau khi xác định được mối đe doạ, chúng ta tiến hành đánh giá định tính mức độ ảnh hưởng của cả năm miền ảnh hưởng, từ thấp đến cao. Bảng 2 cung cấp các tiêu chí đo lường được đề xuất của chúng tôi cho các mỗi miền ảnh hưởng.

BẢNG 2. TIÊU CHÍ ĐO LƯỜNG RỦI RO CỦA MIỀN ẢNH HƯỞNG VỚI MỘT HỆ THỐNG

Miền ảnh hưởng	Thấp	Trung bình	Cao	Nghiêm trọng
Danh tiếng/Tín nhiệm	Thiệt hại tối thiểu cho danh tiếng và niềm tin của các bên liên quan	Thiệt hại đáng kể cho uy tín và sự tin cậy của các bên liên quan	Thiệt hại không thể khắc phục đối với niềm tin của các bên liên quan ít quan trọng hơn	Thiệt hại không thể khắc phục đối với niềm tin của các bên liên quan chính hoặc danh tiếng công khai
Hoạt động, sản xuất	Cần thêm 1 chuyên viên toàn thời gian dưới 1 tháng	Cần thêm 1 chuyên viên toàn thời gian dưới 2 tháng	Cần thêm đến 4 chuyên viên toàn thời gian dưới 6 tháng	Cần huy thêm 4 chuyên viên toàn thời gian trên 6 tháng
An toàn và sức khỏe	Không có nguy cơ đáng kể cho sức khỏe và an toàn	Tối đa 5 ngày suy giảm sức khỏe của chuyên viên hoặc công dân	Hơn 5 ngày suy giảm sức khỏe của chuyên viên hoặc công dân	Suy giảm vĩnh viễn về sức khỏe của chuyên viên hoặc công dân
Tài chính	Mất dưới 50 triệu đồng/năm	Mất từ 50 đến 200 triệu đồng/năm	Mất từ 200 đến 500 triệu đồng/năm	Mất trên 500 triệu đồng/năm
Chi phí và các hình phạt pháp lý	Tiền phạt, chi phí kiện/điều tra không vượt quá 50 triệu đồng	Tiền phạt, chi phí kiện/điều tra từ 50 đến 200 triệu đồng	Tiền phạt, chi phí kiện/điều tra vượt quá 200 triệu đồng	Chi phí rất lớn/Phải tiến hành điều tra pháp lý

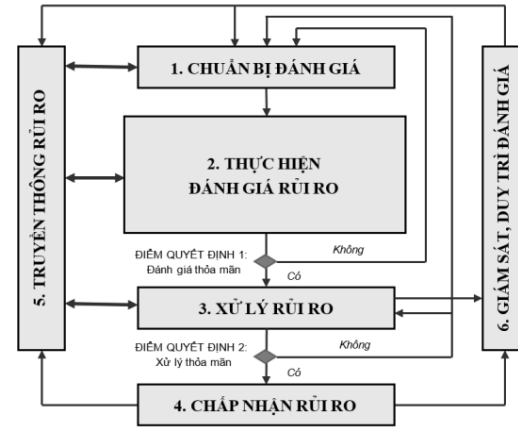
III. GIẢI PHÁP ĐÁNH GIÁ, QUẢN LÝ RỦI RO ATTT

Trong khuôn khổ đề tài cấp nhà nước KC.01.19/16-20, giải pháp đánh giá, quản lý rủi ro ATTT của chúng tôi được xây dựng bao gồm: (i) quy trình đánh giá, quản lý rủi ro ATTT và (ii) hệ thống phần mềm cung cấp các chức năng để các chuyên viên có chức năng trong các cơ quan cấp Bộ có thể thực hiện được các bước trong quy trình đó. Hai thành phần của giải pháp này sẽ được trình bày cụ thể ở các mục sau [12]-[13].

A. Quy trình đánh giá, quản lý rủi ro ATTT trong Chính phủ điện tử

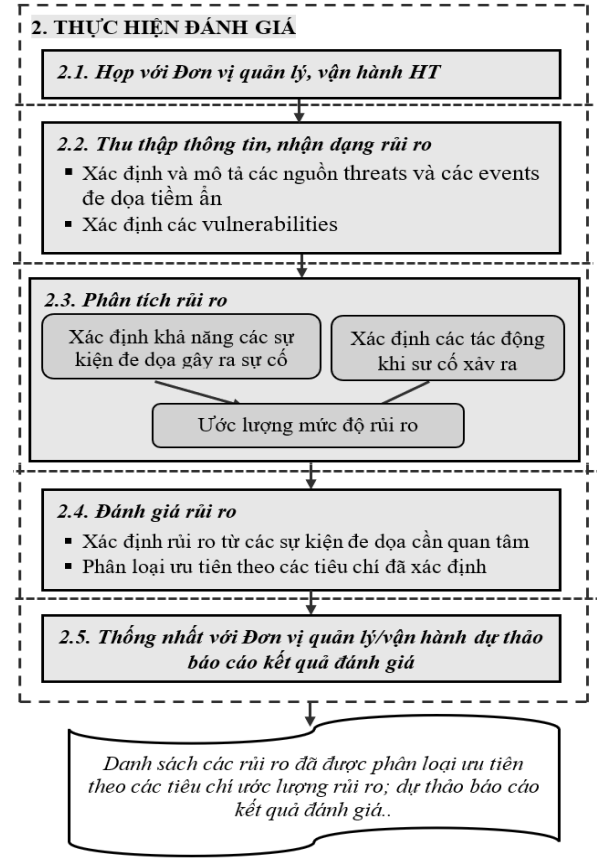
Về quy trình đánh giá, quản lý ATTT, hiện trên thế giới đã có ISO/IEC 27005 và NISP SP800-39. Hướng tiếp cận của nghiên cứu là tùy biến, kết hợp các quy trình trên để đề xuất các quy trình đánh giá, quản lý rủi ro ATTT trong Chính phủ điện tử.

Quy trình chung đánh giá, quản lý rủi ro ATTT theo ISO/IEC 27005 như Hình 2 [14]:



Hình 2. Quy trình chung đánh giá, quản lý ATTT.

Từ quy trình chung nêu trên, kết hợp và tùy biến quy trình của NISP SP800-39 [15]-[16], quy trình đánh giá rủi ro ATTT trong các cơ quan chính phủ được thiết kế như trong Hình 3.



Hình 3. Quy trình đánh giá rủi ro ATTT.

B. Kiến trúc hệ thống hỗ trợ đánh giá, quản lý rủi ro ATTT trong các hệ thống thông tin của Chính phủ điện tử

Hệ thống đánh giá, quản lý rủi ro ATTT, được chúng tôi xây dựng và gọi tắt là UET.SRA bám sát quy trình đánh giá, quản lý rủi ro đã xây dựng ở mục trên, bao gồm các thành phần chính sau:

1. Phần thu thập, xác định rủi ro ATTT

Phần này được giao phó cho các công cụ phát hiện lỗ hổng, thu thập thông tin hiện trạng các bản vá hệ thống thông tin tại các máy tính. Việc thu thập rủi ro ATTT sẽ bao gồm 2 công cụ chính:

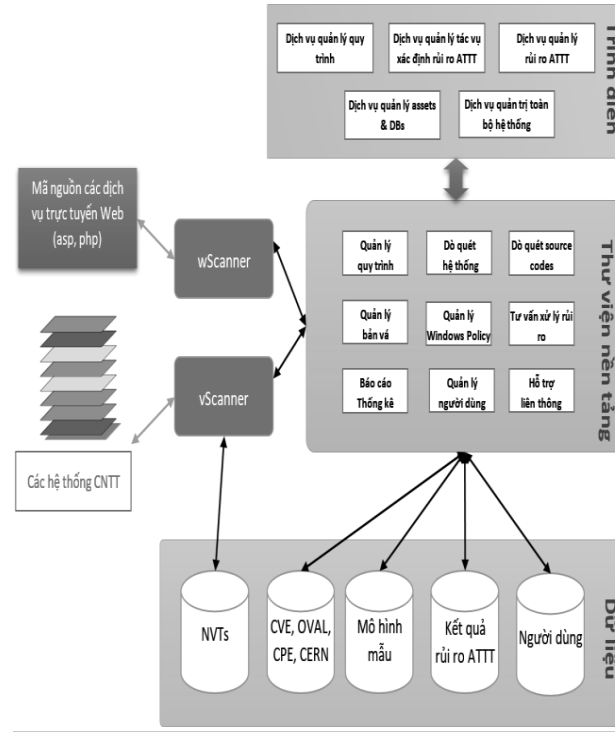
- Công cụ *vScanner* phục vụ dò quét các lỗ hổng ATTT, các chính sách Windows bị vi phạm quy định, các bản vá chưa được thi hành trong phần mềm trên máy tính và các thiết bị mạng được kiểm tra. vScanner đảm nhiệm chức năng đánh giá từ xa đối với mọi hệ thống CNTT nói chung: các máy tính đầu cuối với các hệ thống thông tin (cả hệ điều hành lẫn ứng dụng), các thiết bị mạng. Ngoài ra, vScanner còn hỗ trợ khả năng đánh giá tại chỗ thông qua các chức năng dò quét sâu bên trong hệ thống phần mềm cần đánh giá. Việc này sẽ được thực hiện đối với những máy tính chuyên biệt cần phải đánh giá rủi ro ATTT sâu hơn. Trong bài báo này, hiện chúng tôi chỉ chú trọng đến máy tính sử dụng hệ điều hành Windows và được cấp tài khoản đủ quyền để đánh giá các chính sách Windows và các bản vá lỗ hổng trên máy tính đó.

- Công cụ *wScanner* phục vụ dò quét các lỗ hổng và Webshell trong mã nguồn ứng dụng Web được xây dựng bằng ngôn ngữ lập trình thông dụng PHP hoặc ASP. Được xây dựng kết hợp giữa phương pháp học sâu mạng nơ-ron tích chập CNN với kỹ thuật so khớp mẫu sử dụng tập luật yara rule để tăng cường độ chính xác.

Việc quản lý quá trình thi hành các công cụ xác định rủi ro ATTT nêu trên sẽ được tiến hành thông qua dịch vụ trực tuyến trên nền Web.

2. Phần quản lý rủi ro ATTT

Phần này đảm nhiệm chức năng hỗ trợ phân tích, đánh giá rủi ro, từ đó đề xuất phương án xử lý (thông qua các báo cáo, thống kê phục vụ cho công tác xử lý các lỗ hổng) và giám sát kết quả xử lý rủi ro ATTT sẽ được thực hiện tại hệ thống trung tâm thông qua dịch vụ trực tuyến trên nền Web.



Hình 4. Mô hình kiến trúc tổng thể hệ thống đánh giá, quản lý rủi ro ATTT.

C. Xây dựng hệ thống đánh giá, quản lý rủi ro ATTT - UET.SRA

1. Nền tảng thư viện hệ thống

Nền tảng thư viện hệ thống đánh giá, quản lý rủi ro ATTT có vai trò cung cấp tập các chức năng bên dưới (dạng Back-End) để từ đó các công cụ, dịch vụ trực tuyến ở mức trên (Front-End) có thể khai thác, sử dụng để thi hành các chức năng theo thiết kế.

Trong thư viện này, chúng tôi tiến hành xây dựng các mô đun chức năng chính sau:

- Quản lý quy trình: đây là mô-đun đảm nhiệm các chức năng để quản lý quy trình đánh giá rủi ro ATTT của toàn bộ hệ thống. Các chức năng đánh giá và quản lý rủi ro được thiết kế tuân thủ theo đúng các quy trình đã được đặc tả trong mục A.

- Dò quét hệ thống: đây là mô-đun bao gồm các hàm chức năng để cho phép triệu gọi bộ dò quét lỗ hổng *vScanner* để tiến hành

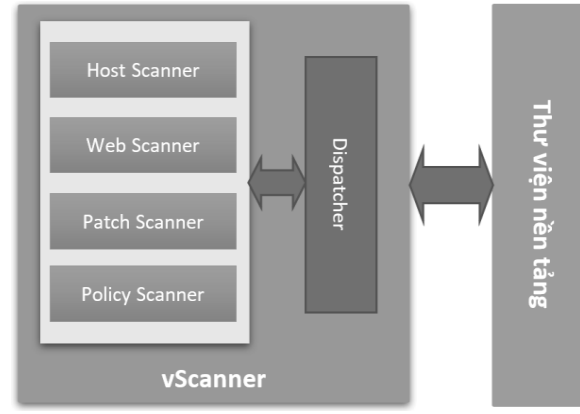
đánh giá hệ thống được định danh thông qua tên hoặc địa chỉ IP.

- Dò quét mã nguồn: là mô đun đảm nhiệm các chức năng triệu gọi công cụ *wScanner* để dò quét các lỗ hổng, các đoạn mã độc trong mã nguồn các ứng dụng Web được xây dựng từ hai ngôn ngữ lập trình là PHP và ASP.NET.
- Quản lý bản vá: là mô-đun chứa các thư viện phục vụ phát hiện các bản vá hệ thống và đưa ra những cảnh báo đối với người quản trị để xử lý phù hợp.
- Quản lý chính sách Windows: là mô đun phục vụ chức năng dò quét, xác định các chính sách trong hệ điều hành Windows vi phạm theo những tiêu chí đã xác lập trong hệ thống (được gọi là bộ chính sách cơ sở - Policy Baseline) và đưa ra những cảnh báo đối với người quản trị.
- Tư vấn, xử lý rủi ro: là mô-đun đảm nhiệm chức năng phân tích các kết quả dò quét, đánh giá hệ thống thông tin và tư vấn, đề xuất phương án để giảm thiểu rủi ro, nâng cao độ ATTT.
- Báo cáo, thống kê: là mô đun tập hợp các hàm chức năng để hỗ trợ hình thành các báo cáo, thống kê theo yêu cầu nghiệp vụ từ thành phần front-end của hệ thống.
- Quản lý người dùng: là mô đun chứa các hàm phục vụ quản lý người dùng theo vai trò và theo quyền được giao để khai thác các chức năng của hệ thống đích.
- Hỗ trợ liên thông: là mô đun chứa tập các hàm chức năng cho phép tiếp nhận các kết quả đánh giá rủi ro từ các hệ thống đánh giá khác cũng như cho phép truy xuất được kết quả đánh giá rủi ro ATTT từ hệ thống sản phẩm đề tài dựa trên những tài khoản có thẩm quyền tương ứng.

2. Công cụ vScanner

Công cụ vScanner đảm nhiệm công việc dò quét các lỗ hổng dẫn đến rủi ro ATTT trong các hệ thống CNTT (bao hàm các máy tính và thiết bị mạng trong hạ tầng mạng, các phần mềm ứng dụng thông dụng tại các máy tính đó; trong tiếng Anh hay dùng thuật ngữ "asset"). Công cụ này sẽ được xây dựng theo cả kỹ thuật kiểm tra hộp đen (đánh giá từ bên ngoài hệ thống) lẫn các phương pháp dò quét sâu (thông qua việc sử dụng tài khoản chuyên biệt để đăng nhập hệ thống, có thể sử dụng ssh, remote login,… để thực hiện việc dò quét trực tiếp bên trong hệ thống). Một số phương pháp học máy thống kê cũng có thể được áp dụng nghiên cứu trong nhiệm vụ này nhằm năng cao độ chính xác của giai đoạn phát hiện lỗ hổng. Đối với các hệ thống ứng dụng Web, công cụ vScanner được tích hợp mô-đun WebScanner thực hiện dò quét chuyên sâu. Ngoài việc phát hiện các lỗ hổng, vScanner còn đảm nhiệm thêm chức năng hỗ trợ phát hiện các Chính sách Windows vi phạm quy định, các ứng dụng/phần mềm chưa thi hành các bản vá lỗ hổng đã được khuyến cáo áp dụng. Dĩ nhiên, các chức năng này phải có tài khoản truy cập vào các máy tính Windows cần phải đánh giá những loại rủi ro này.

Mô hình tổng thể của vScanner được minh họa như Hình 5.



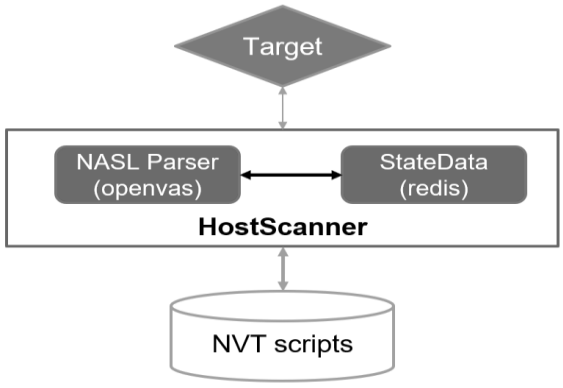
Hình 5. Kiến trúc tổng thể của vScanner.

- HostScanner

Hiện cả Nessus lẫn Greenbone OpenVAS đều cho phép người dùng sử dụng tập plugin chứa tệp các mẫu dò quét lỗ hổng NVT. Theo thống kê, hiện tại Greenbone cung cấp khoảng hơn 65.000 mẫu NVT cho cộng đồng; Nessus cung cấp khoảng hơn 150.000 mẫu thử NVT. Việc sử dụng phiên bản NASL mới cho phép tích hợp được nhiều NVT, giảm số lượng thi hành các bước thử lỗ hổng và đây là lý do số lượng NVT của Greenbone chỉ còn khoảng một nửa của Nessus. Chính vì vậy, trong nghiên cứu này, chúng tôi sử dụng bộ mẫu thử lỗ hổng NVT của Greenbone để

hình thành cơ sở dữ liệu tập mẫu thử NVT phục vụ dò quét lỗ hổng hệ thống [2].

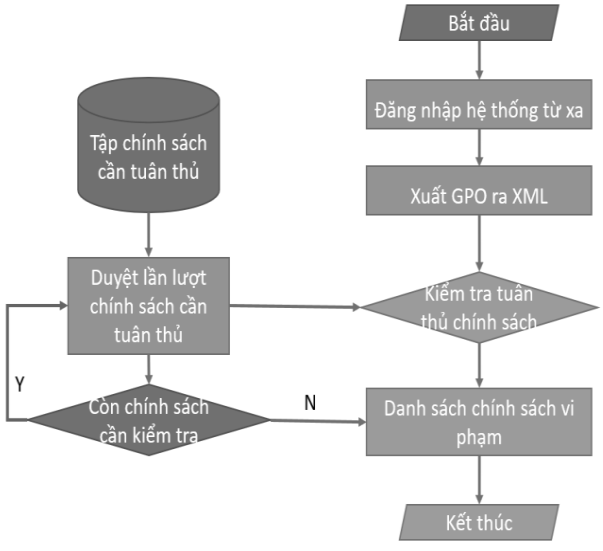
Trong mô hình tại Hình 6, hoạt động của HostScanner sử dụng công cụ Redis để giảm tải thời gian truy vấn CSDL NVT. Với giải pháp này, toàn bộ CSDL NVT sẽ được HostScanner tải về và đưa vào Redis để tăng tốc độ truy xuất. Ngoài ra, Redis cũng được sử dụng để thi hành song song bộ dò quét NASL và lưu trữ tạm thời các kết quả dò quét.



Hình 6. Kiến trúc dò quét lỗ hổng dựa trên CSDL mẫu thử.

- PolicyScanner

Phương pháp dò quét các chính sách Windows bị vi phạm được minh họa như lược đồ giải thuật ở Hình 7.



Hình 7. Giải thuật kiểm tra chính sách Windows bị vi phạm.

Chi tiết đoạn mã chương trình này được trình bày dưới đây:

```
from pypsrp.powershell import
PowerShell, RunspacePool
from pypsrp.wsman import WSMan
wsman = WSMan(in_server, ssl=in_ssl,
auth=in_auth,
encryption=in_encryption,
username=in_username,
password=in_password)
```

tạo kết nối đến máy server in_server qua username và password.

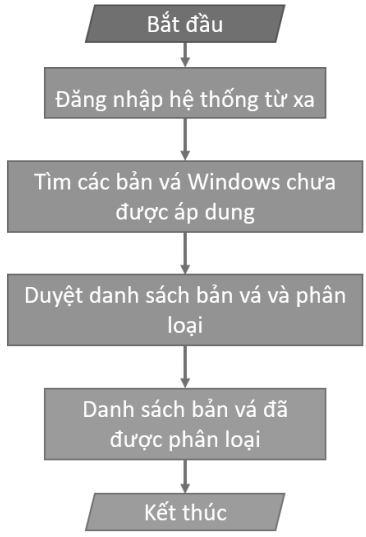
```
with RunspacePool(wsman) as pool:
  ps = PowerShell(pool)
  ps.add_script(my_ps_code)
  # output dạng list do script từ
powershell truyền về
  #
['info,LockoutDuration,30,30,minute'
,
'warning,ResetLockoutCount,30,40,min
ute']
  output = ps.invoke()
  for item in output:
    print(item)
```

Kết quả dò quét các Chính sách Windows vi phạm như Hình 8.

```
optional,MaxRenewAge,7,unknown,unknown
optional,MaximumPasswordAge,180,unknown,unknown
optional,MinimumPasswordAge,0,unknown,unknown
optional,MaxServiceAge,600,unknown,unknown
warning,LockoutBadCount,0,20,minute
optional,MaxClockSkew,5,unknown,unknown
optional,MaxTicketAge,10,unknown,unknown
optional,PasswordHistorySize,24,unknown,unknown
warning,MinimumPasswordLength,5,10,character
optional,PasswordComplexity,0,unknown,unknown
optional,ClearTextPassword,0,unknown,unknown
optional,TicketValidateClient,0,unknown,unknown
```

Hình 8. Kết quả thực nghiệm dò quét các Windows Policy vi phạm.PatchScanner.

Quy trình thực hiện việc dò quét và lấy được danh sách các bản vá chưa được cập nhật trên máy chủ Windows như Hình 9.

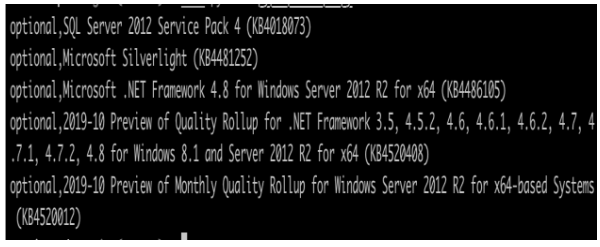


Hình 9. Giải thuật xác định danh mục bản vá chưa được áp dụng.

Việc quản lý bản vá hệ điều hành được xây dựng dựa trên ngôn ngữ lập trình Python, ngôn ngữ kịch bản tương tác máy chủ là Powershell. Để kết nối đến các máy chủ chạy hệ điều hành Windows Server, chúng tôi sử dụng remote Powershell thông qua dịch vụ Winrm và sử dụng ngôn ngữ Powershell để thu thập thông tin.

Tiếp theo, chúng tôi viết một chương trình bằng ngôn ngữ Python để có thể triệu hồi gọi tệp tin PowerShell script từ máy tính Windows. Chương trình này có thể kết nối tới một máy tính Windows thông qua IP, tài khoản và mật khẩu được cấp để có thể gọi tệp tin PowerShell dò quét bản vá chưa được áp dụng.

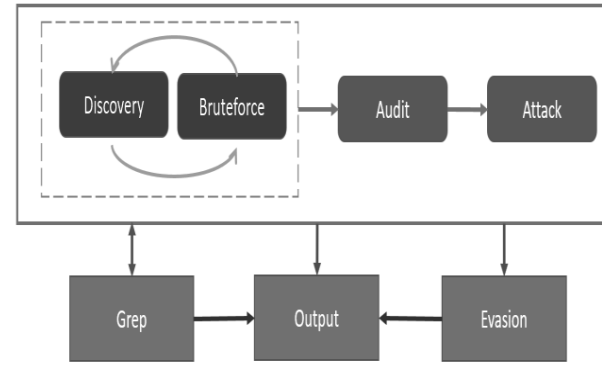
Kết quả dò quét các bản vá Windows chưa được áp dụng:



Hình 10. Kết quả thực nghiệm dò quét các bản vá Windows.

- WebScanner

W3af được chia thành hai phần chính, phần lõi - Core và phần bổ sung - Plugin. Core điều phối quá trình và cung cấp các tính năng được sử dụng bởi các Plugin, có vai trò sẽ tìm ra các lỗ hổng và thử khai thác chúng. Các Plugin được kết nối, chia sẻ thông tin và được phân loại thành Discovery, Audit, Grep, Attack, Output, Mangle, Evasion và Bruteforce.



Hình 11. Kiến trúc công cụ dò quét lỗ hổng ứng dụng Web.

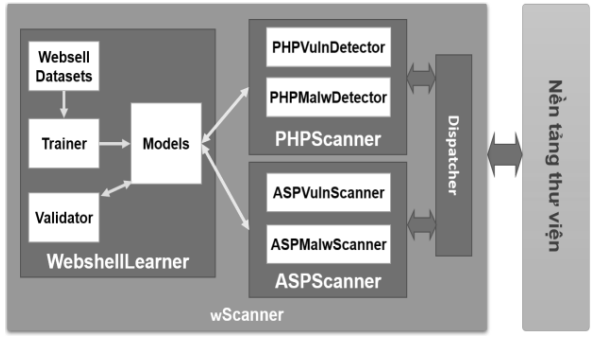
Hiện W3af có thể xác định được hơn 200 lỗ hổng bảo mật ứng dụng Web. Khung này được phát triển bằng Python, dễ sử dụng, mở rộng và tích hợp trong các ứng dụng khác. W3af có một số Plugin có thể giao tiếp với nhau. Ví dụ: Plugin Discovery có thể xác định các URL khác nhau cho ứng dụng và chuyển kết quả của nó đến Plugin Audit rồi có thể sử dụng các URL để tìm kiếm lỗ hổng bảo mật. Sau đó, Plugin Exploit có thể được sử dụng để khai thác bất kỳ lỗ hổng đã xác định nào.

3. Công cụ wScanner

Công cụ *wScanner* đảm nhiệm vai trò dò quét tĩnh mã nguồn các ứng dụng Web để phát hiện các đoạn mã độc hoặc Webshell. Tuy nhiên, khác với *vScanner* dùng để dò quét lỗ hổng các ứng dụng web chung, còn đối với công cụ *wScanner* phụ thuộc rất nhiều vào công nghệ và ngôn ngữ phát triển ứng dụng web. Do đó, công cụ *wScanner* được tập trung vào các loại ứng dụng Web thông dụng nhất hiện nay là ASP và PHP. Đây là nhiệm vụ cũng rất chuyên biệt trong đề tài này với mong muốn của nhóm là có thể xây dựng được hệ thống phần mềm không chỉ dò quét bên ngoài các hệ thống CNTT mà còn phân tích sâu toàn bộ mã nguồn của các ứng dụng Web để từ đó tìm ra những lỗ hổng và các Webshell độc hại đã được đưa vào (có thể do bất cẩn hoặc cố ý) trong mã nguồn. Sử dụng *wScanner* sẽ cho phép giảm thiểu ngay được những nguy cơ từ trong mã nguồn liên quan đến những lỗ hổng cơ bản như

SQL Injection, XSS, XSS flaw (bao gồm Command Injection, Object Injection, File Inclusion, XPATH injection, Arbitrary Eval Code Injection); kiểm tra được các hàm an toàn (secure function) với các điều kiện cụ thể để phát hiện lỗ hổng. Phương pháp học sâu cũng sẽ được chú trọng nghiên cứu trong nhiệm vụ này để có thể áp dụng ngay được công nghệ đánh giá độ tin cậy và rủi ro theo cách tiếp cận thích ứng liên tục (CARTA - Continuous Adaptive Risk and Trust Assessment) đối với các ứng dụng Web phổ biến trong các cơ quan nhà nước hiện nay.

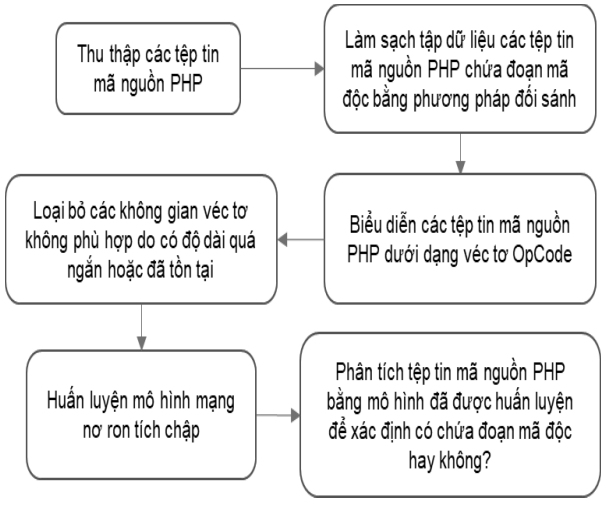
Kiến trúc tổng thể của wScanner được minh họa như Hình 12.



Hình 12. Kiến trúc tổng thể của wScanner.

- Mô hình phát hiện đoạn mã độc trong PHP

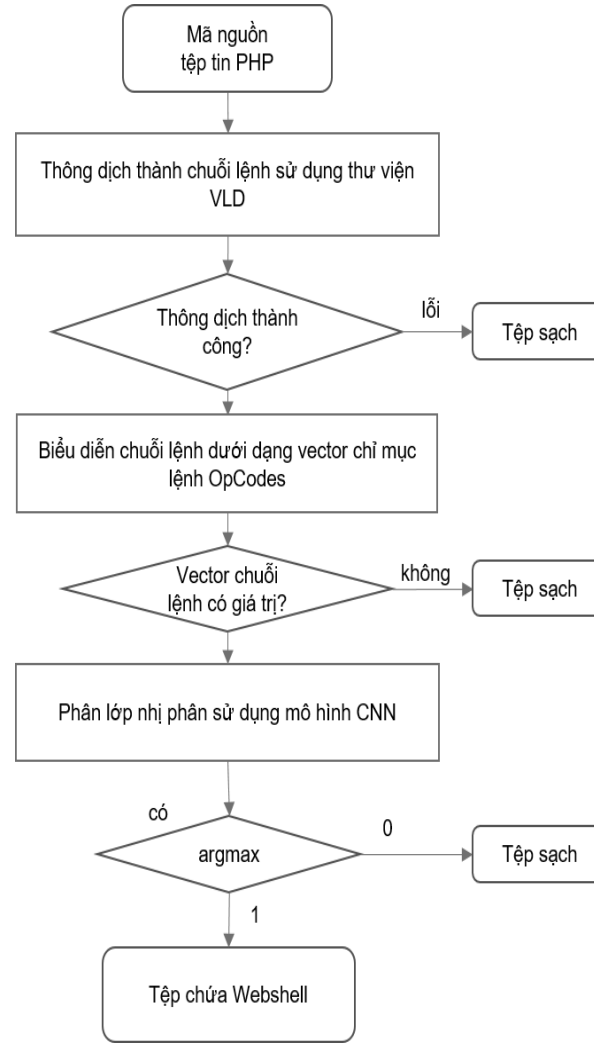
Đối với ứng dụng Web sử dụng PHP, phương pháp để phát hiện đoạn mã độc (Webshell) trong mã nguồn ứng dụng Web được đề xuất dựa trên sự kết hợp giữa phương pháp đối sánh mẫu và phương pháp học sâu sử dụng mạng nơ-ron tích chập, được minh họa như Hình 13 [3]:



Hình 13. Phương pháp phát hiện đoạn mã độc trong ứng dụng Web PHP.

Từ mô hình giải pháp trên, chúng tôi ứng dụng phương pháp học sâu để phân tích mã nguồn ứng dụng Web phát hiện các đoạn mã độc được cài vào, cụ thể là mô hình học sâu mạng nơ-ron tích chập (Convolution Neural Network – CNN) kết hợp với kỹ thuật so khớp mẫu với tập dữ liệu yara rule để tăng cường độ chính xác cho mô hình học sâu CNN để xây dựng công cụ phục vụ phát hiện đoạn mã độc, đặt tên chung là WebshellLearner.

Dựa trên các nguồn dữ liệu đã được một số tổ chức nghiên cứu đã sử dụng như OWASP, NIST, chúng tôi đã lựa chọn và thu thập từ trang GitHub được hơn 4.200 tệp PHP chứa Webshell. Để có được các tệp PHP "sạch" (benign), chúng tôi đã sử dụng các tệp từ các bộ thư viện, CMS nổi tiếng như Laravel,Wordpress, Joomla, phpMyAdmin, phpPgAdmin, phpbb, tổng cộng đã thu thập được 7.400 tệp.



Hình 14. Quy trình phát hiện đoạn mã độc trong ứng dụng Web PHP.

Sau khi loại bỏ và chuẩn hoá các tệp mã nguồn thành các chuỗi opcode, chúng tôi đã thu được tập dữ liệu để huấn luyện và kiểm thử như sau (tỷ lệ ban đầu được chọn là 7:3) như trong Bảng 3.

BẢNG 3. THÔNG TIN VỀ TẬP DỮ LIỆU ĐỂ HUẤN LUYỆN MÔ HÌNH VÀ KIỂM THỬ CHO PHP

	Tập huấn luyện	Tập kiểm thử
Benign	4,875	1,182
Webshell	1,049	275

Kết quả kiểm thử với tập thử nghiệm nêu trên được minh họa ở Bảng 4 [11]:

BẢNG 4. KẾT QUẢ KIỂM THỬ BỘ PHÁT HIỆN ĐOẠN MÃ ĐỘC WEBSHELL PHP

	Accuracy	Precision	Recall	F1-Score	FPR
Benign	99,02	99,66	99,15	99,41	1,60
Webshell	99,02	96,09	98,40	97,23	0,85
Micro Avg	99,02	99,66	99,15	99,41	0,85
Macro Avg	99,02	97,88	98,78	98,32	1,22
Weighted Avg	99,02	99,04	99,02	99,03	1,47

Như vậy, với việc áp dụng mô hình học sâu CNN vào bộ phát hiện mã độc của wScanner, kết quả dò quét đoạn mã độc đạt độ chính xác accuracy trên 99%, tỷ lệ phát hiện nhầm kiểu dương tính giả FPR với Webshell là 0,85%.

So với một số phương pháp phát hiện Webshell khác, kết quả của chúng tôi cũng rất đáng ghi nhận với bảng kết quả như Bảng 5:

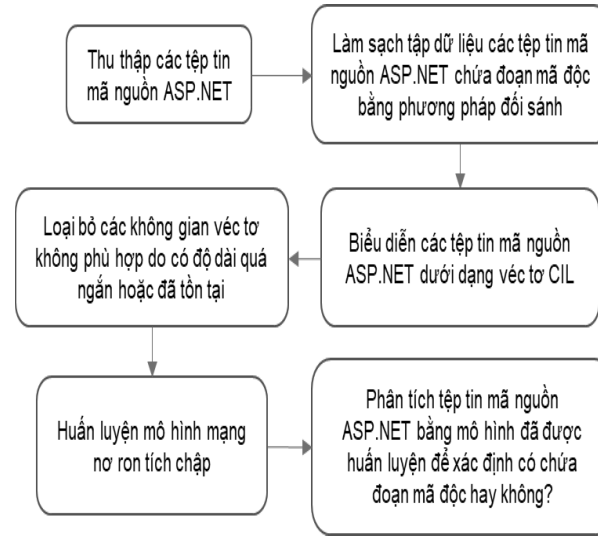
BẢNG 5. SO SÁNH KẾT QUẢ GIỮA WEBSHELLLEARNER VÀ MỘT SỐ PHƯƠNG PHÁP KHÁC ĐỐI VỚI PHP

	Accuracy	F1-Score	FPR
php-malware-finder	94.23	96.46	0.00
Word2Vec+CNN	98.60	98.60	0.92
RF-GBDT	99.16	99.09	0.93
GuruWS	85.56	92.00	0.00
Yara	98.15	98.87	0.17
CNN	97.60	97.88	2.11
WebshellLearner	99.02	99.41	0.85

- Mô hình phát hiện đoạn mã độc trong ASP.NET

Kỹ thuật phát hiện đoạn mã độc trong mã nguồn ứng dụng ASP.NET

Đối với ứng dụng Web sử dụng ASP.NET, mô hình đề xuất của chúng tôi dựa trên ba ý tưởng chính: (i) đề xuất phương pháp biểu diễn mã nguồn tệp tin ASP.NET dưới dạng vector mã lệnh CIL OpCode, (ii) làm sạch bộ dữ liệu huấn luyện mô hình dựa trên kỹ thuật đối sánh mẫu so khớp sử dụng bộ công cụ yara, và (iii) áp dụng mô hình học sâu sử dụng mạng nơ-ron tích chập CNN để huấn luyện mô hình và xác định xem mã nguồn ứng dụng Web có chứa Webshell hay không. Mô hình phát hiện đoạn mã độc Webshell ASP.NET được minh họa như Hình 15.



Hình 15. Quy trình phát hiện đoạn mã độc trong mã nguồn ứng dụng ASP.NET

Chúng tôi cũng sử dụng phương pháp học sâu với mạng nơ-ron tích chập CNN được minh họa như trên. Toàn bộ mô hình huấn luyện và phát hiện đoạn mã độc trong ứng dụng ASP.NET cũng được chúng tôi tích hợp vào công cụ WebshellLearner.

Để thu thập dữ liệu phục vụ huấn luyện mô hình, dựa trên các nguồn dữ liệu đã được một số tổ chức nghiên cứu đã sử dụng như OWASP, NIST,… chúng tôi lựa chọn và thu thập từ trang GitHub được hơn 7.208 tệp mã ASP.NET sạch và 913 mẫu chứa Webshells. Sau khi loại bỏ và chuẩn hoá các tệp mã nguồn thành các chuỗi opcodes, chúng tôi đã thu được tập dữ liệu để

huấn luyện và kiểm thử như Bảng 6 (tỷ lệ ban đầu được chọn là 7:3).

BẢNG 6. THÔNG TIN VỀ TẬP DỮ LIỆU ĐỂ HUẤN LUYỆN MÔ HÌNH VÀ KIỂM THỬ CHO ASP.NET

	Tập huấn luyện	Tập kiểm thử
Benigns	4.547	1.161
Webshells	616	228

Kết quả kiểm thử với tập thử nghiệm nêu trên được minh họa ở Bảng 7.

BẢNG 7. KẾT QUẢ MÔ HÌNH PHÁT HIỆN ĐOẠN MÃ ĐỘC ASP.NET (%)

Metric	Value
Accuracy	98.49
Precision	99.65
F1-score	99.09
Recall	98.54
True Negative Rate	98.25
Negative Predictive Value	92.95
False Discovery Rate	0.35
False Negative Rate	1.46
False Positive Rate	1.75

Như vậy, với việc áp dụng mô hình học sâu CNN vào bộ phát hiện mã độc của wScanner, kết quả dò quét đoạn mã độc ASP.NET đạt độ chính xác accuracy trên 98,5%, tỷ lệ phát hiện nhầm kiểu dương tính giả FPR với Webshell là 1%. Một điểm cũng cần nhấn mạnh hơn ở điểm này là hiện chưa có nhiều bộ phát hiện đoạn mã độc ASP.NET được công bố tỷ lệ phát hiện chính xác. Do đó, phương pháp này của chúng tôi chưa thể tiến hành so sánh đánh giá kết quả này như đối với PHP.

4. Dịch vụ trực tuyến quản lý rủi ro ATTT

Để quản lý rủi ro ATTT, bao gồm cả các chức năng phục vụ phân tích, đánh giá và hỗ trợ xử lý rủi ro ATTT có giám sát kèm theo, chúng tôi xây dựng 05 dịch vụ trực tuyến. Mô tả cụ thể của mỗi dịch vụ sẽ được trình bày cụ thể dưới đây:

- Dịch vụ quản lý quy trình nghiệp vụ đánh giá, quản lý rủi ro ATTT

Đây là dịch vụ hỗ trợ cho các bộ phận quản lý và giám sát quá trình thi hành các chính sách đảm bảo ATTT trong các cơ quan nhà nước. Dịch vụ này cung cấp các chức năng cơ bản cho phép tra cứu, tìm kiếm thông tin mẫu liên quan đến quy trình đánh giá và quản lý rủi ro ATTT nói chung. Ngoài ra, dịch vụ này cũng cung cấp thông tin chi tiết về các quy trình đánh giá, quản lý rủi ro ATTT đã được công bố trong các tiêu chuẩn cả trong nước lẫn quốc tế. Cụ thể, cung cấp quy trình trong tiêu chuẩn TCVN 10295:2014 (ISO/IEC 27005:2011), TCVN ISO/IEC 27001:2009, TCVN 8709-1/2/3:2011 (ISO/IEC 15408-1/2/3:2008); quy trình đánh giá rủi ro ATTT của Tổ chức tiêu chuẩn Mỹ NIST SP 800-30r1, NIST SP 800-39, NIST SP 800-53r5 [14]-[19].

- Dịch vụ quản lý tác vụ xác định rủi ro ATTT trong các hệ thống CNTT

Dịch vụ này đảm nhiệm vai trò triệu gọi các phần mềm dò quét, phát hiện rủi ro ATTT *vScanner* và *wScanner* để tiến hành đánh giá, xác định các nguy cơ, lỗ hổng bảo mật có thể có trong các hệ thống CNTT. Dịch vụ này có tổng hợp các chức năng rất quan trọng trong hệ thống đánh giá, quản lý rủi ro ATTT và tương tác trực tiếp với hai bộ công cụ *vScanner* và wScanner. Từ các hệ thống CNTT đã được xác định đưa vào hệ thống để đánh giá rủi ro, *vScanner*/*wScanner* sẽ được triệu gọi, lưu lại kết quả xác định rủi ro trong CSDL của hệ thống. Dịch vụ này cũng đảm nhiệm thêm chức năng định kỳ thực hiện tác vụ xác định rủi ro theo lịch biểu đã xác lập.

- Dịch vụ quản lý rủi ro ATTT trong các hệ thống CNTT

Dựa trên các kết quả đã tiến hành đánh giá và xác định được các nguy cơ, lỗ hổng có thể dẫn đến rủi ro ATTT ở dịch vụ quản lý tác vụ xác định rủi ro ATTT nêu trên, dịch vụ sẽ có vai trò phân tích, đánh giá toàn bộ các nguy cơ, lỗ hổng đó; từ đó đề xuất phương án xử lý các lỗ hổng, các chính sách Windows vi phạm quy định, và áp dụng thi hành tự động cập nhật các bản vá lỗ hổng tại các máy tính Windows đã được cấp quyền vá lỗ hổng. Ngoài ra, dịch vụ này cũng còn phải cung cấp thêm các chức năng hỗ trợ báo cáo, thống kê kết quả phân tích, đánh giá và phương án xử lý rủi ro ATTT; quản lý và hỗ trợ giám sát kết quả xử lý rủi ro ATTT.

Việc phân tích, đánh giá các rủi ro nói chung sẽ được tiến hành với sự kết hợp cả phân tích định tính và phân tích định lượng. Các phân tích định

lượng sẽ được áp dụng thông qua những kỹ thuật sử dụng các mẫu rủi ro đã được các tổ chức trên thế giới công bố (CVE và CPE), sử dụng các tiêu chuẩn CVSS hoặc OWASP [20-21]. Đối với phân tích định tính, do bài toán xác định rủi ro ATTT cơ bản dựa vào việc xác định lỗ hổng bảo mật có độ đo rủi ro lớn nhất. Hơn nữa, mỗi lỗ hổng bảo mật gắn với ít nhất một thông tin CVE, nên toàn bộ các phương án xử lý lỗ hổng, tránh rủi ro ATTT từ các lỗ hổng đó đều được gắn kèm vào trong CSDL hệ thống. Từ đó, quá trình phân tích định tính xác định mức rủi ro quy về việc xác định giá trị rủi ro cao nhất và ánh xạ sang mức độ rủi ro theo theo các ngưỡng khác nhau.

Ngoài ra, trong dịch vụ này chúng tôi cũng sẽ cung cấp thêm một công cụ để người dùng có thể tự đánh giá mức độ rủi ro ATTT, một cách định lượng, theo tiêu chuẩn CVSS.

- Dịch vụ quản lý các hệ thống CNTT và các CSDL

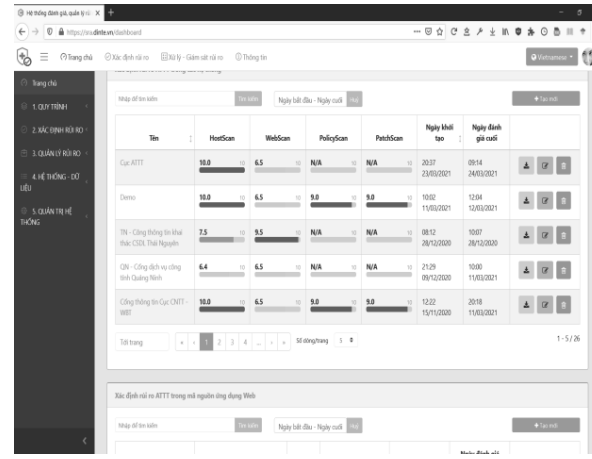
Đây là dịch vụ có vai trò quản lý toàn bộ các tài sản đã được đánh giá trong cơ quan và các bộ CSDL mẫu phục vụ cho cả công tác xác định rủi ro (các lỗ hổng) lẫn công tác đánh giá, hỗ trợ xử lý. Các bộ CSDL mẫu được quản lý bởi dịch vụ này bao gồm: danh mục các ứng dụng CPE, danh mục các lỗ hổng CVE+OPAL, danh mục các lỗ hổng được các trung tâm ứng cứu khẩn cấp xác lập, danh mục các bản vá lỗ hổng ứng dụng/hệ thống, danh mục các quy định liên quan đến chính sách Windows và danh mục các mẫu phục vụ cho quá trình kiểm tra các lỗ hổng đã có.

- Dịch vụ quản trị toàn bộ hệ thống

Dịch vụ trực tuyến thứ năm này đảm nhiệm vai trò quản trị cả người dùng lẫn các cấu hình của hệ thống quản lý rủi ro ATTT. Ngoài những chức năng trên, dịch vụ này còn cung cấp thêm một giao diện dashboard để hiển thị các thông tin quan trọng của toàn bộ các dịch vụ trên và cung cấp chức năng giám sát và quản lý các bản vá tại các máy tính đầu cuối.

IV. Thử nghiệm

Hệ thống đã được thử nghiệm thực tế tại Trung tâm hạ tầng CNTT thuộc Cục CNTT và Dữ liệu tài nguyên môi trường (Bộ TN&MT).



Hình 16. Giao diện hệ thống UET.SRA triển khai tại Bộ TN&MT.

Kịch bản thử nghiệm bao gồm dò quét, đánh giá ATTT Hệ thống CNTT tại Trung tâm Hạ tầng, Cổng thông tin điện tử của Cục CNTT và Dữ liệu TNMT, các dịch vụ công trực tuyến của Bộ TN&MT và mã nguồn một số ứng dụng Web của Cục CNTT và Dữ liệu TNMT.

Các dữ liệu thử nghiên được so sánh với công cụ Nessus và cho kết quả tích cực, được mô tả trong Bảng 8.

Bảng 8. So sánh kết quả thử nghiệm hệ thống với Nessus

Hệ thống	UET.SRA (High-Medium-Low)	Nessus (Critical-High-Medium-Low)	Nhận xét
Cổng dịch vụ công trực tuyến Bộ TNMT	0-0-2	0-0-1-0	SRA phát hiện nhiều lỗ hổng hơn; cùng mức rủi ro
Hệ thống quản lý khoa học công nghệ Bộ TNMT	0-9-0	0-0-0-0	SRA phát hiện nhiều lỗ hổng hơn; cao hơn một mức rủi ro
Cổng thông tin Cục CNTT&DLTNMT	0-0-1	0-0-2-1	Nessus phát hiện nhiều lỗ hổng hơn; cùng mức rủi ro
Máy chủ DB Cổng dịch vụ công trực tuyến	0-1-1	0-0-8-1	Nessus phát hiện nhiều lỗ hổng hơn; cùng mức rủi ro

Kết luận

Trong bài báo này, chúng tôi đã trình bày giải pháp đánh giá, quản lý rủi ro ATTT trong các hệ thống thông tin của Chính phủ điện tử. Giải pháp của chúng tôi dựa trên bộ quy trình đánh giá, quản lý rủi ro an toàn các hệ thống CNTT được kết hợp, tùy biến từ ISO/IEC 27005:2011 và NIST SP 800-39 để phù hợp với thực tiễn Việt Nam; cùng hệ thống hỗ trợ công tác đánh giá, quản lý rủi ro ATTT UET.SRA.

Hệ thống được xây dựng bám sát quy trình đề xuất, đã tích hợp được nhiều công cụ phần mềm để dò quét, xác định rủi ro ATTT dựa theo phương pháp kiểm tra các lỗ hổng bảo mật; (trên hệ thống và mã nguồn ứng dụng Web); dò quét bằng công cụ vScanner/wScanner và hỗ trợ đánh giá tổng thể với các tiêu chuẩn CVSS, OWASP. Một điểm mạnh của UET.SRA nữa là hệ thống này cho phép phân tích, phát hiện các lỗ hổng, các đoạn mã độc trong mã nguồn các ứng dụng Web sử dụng công nghệ học sâu với độ chính xác F10-score rất cao (99,41% đối với PHP và 99,09% đối với ASP.NET), tỷ lệ phát hiện nhầm FPR cũng rất nhỏ (0,85% đối với PHP và 1,75% đối với ASP.NET). Hệ thống đã thử nghiệm thực tế và so sánh kết quả đánh giá cho kết quả khả quan, bám sát được công cụ chuyên dụng hàng đầu hiện nay Nessus.

Trong thời gian tới, chúng tôi sẽ tiếp tục hoàn thiện thêm hệ thống phần mềm, bổ sung thêm những mẫu dò quét lỗ hổng chuyên sâu cho các hệ thống chuyên dụng trong Chính phủ điện tử tại Việt Nam.

LỜI CẢM ƠN

Bài báo này được hỗ trợ từ đề tài nghiên cứu cấp Nhà nước số KC.01.19/16-20.

TÀI LIỆU THAM KHẢO

[1] R. M. Savola and P. Heinonen (2011), "A visualization and modeling tool for security metrics and measurements management," doi:10.1109/ISA.2011.6027518. Information Security for South Africa (ISSA) Conference, pp. 1-8.

[2] Himanshu Kumar (2014), "Learning Nessus for Penetration Testing", Packt Publishing.

[3] Sagar Rahalkar (2018), "Network Vulnerability Assessment: Identify security loopholes in your network's infrastructure", Packt Publishing.

[4] https://www.greenbone.net/en/live-demo/ (Truy cập ngày 10/3/2021).

[5] https://www.tenable.com/plugins/newest (Truy cập ngày 10/3/2021).

[6] Open Vulnerability Assessment Language (OVAL) scans- https://oval.cisecurity.org/ (Truy cập ngày 15/3/2021).

[7] NIST (2013), "NIST SP 800-40r3 Guide to Enterprise Patch Management Technologies". http://dx.doi.org/10.6028/NIST.SP.800-40r3.

[8] Web Application Attack and Audit Framework – w3af- http://w3af.org/. Truy cập ngày 20/3/2021.

[9] Ngoc-Hoa NGUYEN, Viet-Ha LE, Van-On PHUNG, Phuong-Hanh DU (2019): "Toward a Deep Learning Approach for Detecting PHP Webshell". Proceedings of the Tenth International Symposium on Information and Communication Technology 2019 (SoICT 2019), ACM, New York, NY, USA. Pages 514–521. https://doi.org/10.1145/3368926.3369733.

[10] Lv ZH., Yan HB., Mei R. (2019), "Automatic and Accurate Detection of Webshell Based on Convolutional Neural Network". In Yun X. et al. (eds) Cyber Security. CNCERT 2018. Communications in Computer and Information Science, vol 970. Springer, Singapore. pp 73-85. https://doi.org/10.1007/978-981-13-6621-5_6.

[11] Yifan Tian, Jiabao Wang, Zhenji Zhou, and Shengli Zhou (2017). "CNN-Webshell: Malicious Web Shell Detection with Convolutional Neural Network". In Proceedings of the 2017 VI International Conference on Network, Communication and Computing (ICNCC 2017). ACM, New York, NY, USA, pp. 75-79.

[12] Ha LE Viet, On PHUNG Van and Hoa NGUYEN Ngoc (2020): "Information Security Risk Management by a Holistic Approach: a Case Study for Vietnamese e-Government". IJCSNS- International Journal of Computer Science and Network Security. VOL 20 No.6, June 2020. pp. 72-82.

[13] Nguyễn Ngọc Hóa (2021), "Báo cáo tổng hợp kết quả đề tài KC01.19/16-20".

[14] ISO/IEC 27005:2018 Information technology - Security techniques - Information security risk management (third edition). https://www.iso.org/standard/75281.html (Truy cập ngày 22/3/2021).

[15] NIST (2012), "NIST SP 800-30r, Guide for Conducting Risk Assessments". https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final (Truy cập ngày 25/3/2021).

[16] NIST (2011), "NIST SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View". https://csrc.nist.gov/publications/detail/sp/800-39/final (Truy cập ngày 25/3/2021).

[17] ISO/IEC 15408-1:2009 Information technology - Security techniques - Evaluation criteria for IT security. https://www.iso.org/standard/50341.html (Truy cập ngày 22/3/2021).

[18] NIST (2018), "NIST Framework for Improving Critical Infrastructure Cybersecurity (Cybersecurity Framework)". https://www.nist.gov/cyberframework (Truy cập ngày 25/3/2021).

[19] NIST (2020), "NIST SP 800-53r4, Security and Privacy Controls for Federal Information Systemsand Organizations". https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final (Truy cập ngày 25/3/2021).

[20] "Common Vulnerability Scoring System v3.1: Specification Document", https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf (Truy cập ngày 28/3/2021).

[21] "OWASP Risk Rating Methodology", https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology (Truy cập ngày 28/3/2021).

SƠ LƯỢC VỀ TÁC GIẢ

Phùng Văn Ổn

Đơn vị công tác: Viện CNTT Trường Đại học Tài chính - Ngân hàng Hà Nội.

Email: onphungvan@gmail.com

Quá trình đào tạo: Tốt nghiệp đại học ngành Máy tính năm 1980 tại Trường Đại học Tổng hợp Hà Nội; Tốt nghiệp Thạc sỹ năm 1997 và Tiến sỹ năm 2001 chuyên ngành Bảo đảm toán học cho máy tính và hệ thống tính toán tại Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội.

Hướng nghiên cứu hiện nay: Hệ thống thông tin; An toàn thông tin; Chính phủ điện tử; Giao thông thông minh.

Lê Việt Hà

Đơn vị công tác: Trung tâm Tin học Văn phòng Chính phủ.

Email: levietha@chinhphu.vn

Quá trình đào tạo: Tốt nghiệp đại học năm 2003 Thạc sỹ năm 2011 ngành Toán ứng dụng Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội. Nghiên cứu sinh ngành Hệ thống thông tin từ năm 2019 tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Hướng nghiên cứu hiện nay: Hệ thống thông tin; An toàn thông tin.

Nguyễn Ngọc Hóa

Đơn vị công tác: Khoa CNTT Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Email: hoas.nguyen@gmail.com

Quá trình đào tạo: Tốt nghiệp đại học năm 1999 ngành Khoa học máy tính tại Trường Đại học Bách khoa học Hà Nội. Tốt nghiệp Tiến sỹ ngành Khoa học máy tính năm 2005 tại Trường Đại học Joseph Fourier, Cộng hòa Pháp.

Hướng nghiên cứu hiện nay: Hệ thống thông tin; An toàn thông tin; Chính phủ điện tử, Blockchain, BigData.