Phương pháp đánh giá rủi ro cho ứng dụng Web

Hoàng Đăng Hải, Hồ Kim Cường

***Tóm tắt*— Bảo đảm an toàn ứng dụng Web đang là một nhu cầu thực tế cấp thiết, vì đó là nền tảng cho hầu hết các ứng dụng và giao dịch trên mạng hiện nay. Để đảm bảo an toàn cho hệ thống, cần thường xuyên rà quét lỗ hổng bảo mật, xác định nguy cơ tiềm ẩn và mức độ rủi ro để có những biện pháp hạn chế, khắc phục điểm yếu và tăng cường an toàn. Trong quá trình thực hiện đánh giá, có nhiều khó khăn, thách thức về mặt kỹ thuật được đặt ra do sự phức tạp, gia tăng của các loại lỗ hổng và tấn công. Việc xác định mô hình và phương pháp đánh giá rủi ro phù hợp có tầm quan trọng đặc biệt và là chủ đề nghiên cứu của bài báo này. Thông qua việc đánh giá lỗ hổng bảo mật, bài báo đề xuất một mô hình và phương pháp đánh giá định lượng rủi ro trên cơ sở đánh giá mức độ sự cố và tác động của sự cố đối với ứng dụng Web.**

***Abstract*— Ensuring secure Web applications is becoming an urgent practical demand because it is the foundation for most of the applications and transactions on the network today. To meet that demand, it is necessary to regularly scan security vulnerabilities, to identify potential risks and risk levels of systems in order to provide measures to reduce risks, remedy weakness and enhance security. Many difficulties and technical challenges posed by the increased complexity of the vulnerability and attack types. Identifying appropriate models and methods for Web application's risk assessment has a special significance and is the subject of this paper. By evaluating vulnerabilities, this paper proposed a model and methods for quantitative risk assessment based on the assessment of likelihood and impact of incidents for Web applications.**

Từ khóa— đánh giá rủi ro; rủi ro ứng dụng Web; phương pháp đánh giá rủi ro.

I. MỞ ĐẦU

Các lỗ hổng bảo mật và mối đe dọa có tác động trực tiếp đến khả năng xảy ra sự cố, nghĩa là xảy ra rủi ro. Để đánh giá rủi ro, việc đầu tiên cần làm là xác định khả năng xảy ra sự cố đối với từng loại lỗ hổng bảo mật. Tiếp đó là xác định tác động khi xảy ra sự cố đó.

Một yêu cầu đặt ra trong thực tế là làm thế nào để tính toán định lượng được mức độ rủi ro đối với từng loại lỗ hổng bảo mật, cũng như đối với toàn bộ ứng dụng Web. Qua khảo sát trên mạng và một số tài liệu nghiên cứu [1-5] cho thấy, vẫn chưa có một mô hình hoàn chỉnh giúp đánh giá định lượng cho các ứng dụng Web. Điều đó một phần là do thiếu các số liệu có tính định lượng cho việc đánh giá [1]. Công thức tính rủi ro [2,3] vẫn mang tính chất khái quát hóa, khó áp dụng trong thực tế. Trong [4] và [5], các tác giả cũng đã đề xuất cách tính xác suất cho khả năng xảy ra sự cố và tác động của sự cố. Tuy nhiên có thể thấy, việc xác định các giá trị để tính toán vẫn chưa rõ. Mặt khác, cũng khó đưa ra được các mức có tính định lượng phù hợp cho từng mô hình.

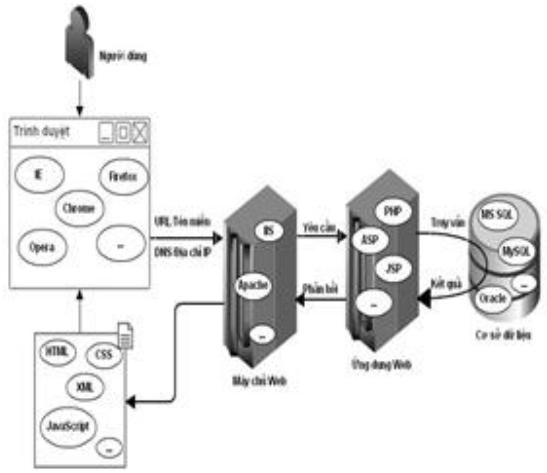
Bài báo này đề xuất một mô hình và phương pháp đánh giá định lượng rủi ro trên cơ sở đánh giá khả năng sự cố và tác động của sự cố đối với ứng dụng Web. Bài báo có bố cục như sau: Sau Mục I, Mục II trình bày về lỗ hổng bảo mật ứng dụng Web làm cơ sở cho đánh giá rủi ro. Mục III trình bày mô hình và phương pháp đánh giá định lượng rủi ro, đưa ra phương pháp tính toán định lượng cho các mức độ. Mục IV là kết quả thử nghiệm đánh giá rủi ro cho một ứng dụng Web. Cuối cùng là Mục Kết luận.

II. LỖ HỔNG BẢO MẬT ỨNG DỤNG WEB

A. Các thành phần ứng dụng Web

Một ứng dụng Web khi triển khai, sẽ có ba lớp cơ bản sau: lớp trình diễn, lớp ứng dụng và lớp cơ sở dữ liệu. Mỗi lớp có thể được thiết lập trên từng máy chủ riêng biệt (hoặc không riêng biệt) gồm máy chủ Web, máy chủ ứng dụng, máy chủ cơ sở dữ liệu (CSDL) [6].

Thông qua trình duyệt, người dùng cuối sẽ kết nối và gửi yêu cầu truy xuất thông tin đến máy chủ Web. Tại đây, máy chủ Web sẽ xử lý thông tin này và gửi yêu cầu đến ứng dụng Web. Tùy theo yêu cầu, ứng dụng Web sẽ truy vấn đến CSDL và nhận kết quả trả về, sau đó sẽ gửi phản hồi về máy chủ Web. Cuối cùng, máy chủ Web sẽ gửi dữ liệu về trình duyệt dưới dạng siêu văn bản và người dùng cuối sẽ nhận được thông tin hiển thị trên trình duyệt [6].



Hình 1. Mô hình hoạt động của ứng dụng Web

Việc phát triển ứng dụng Web có thể dựa trên những bộ khung có sẵn như Joomla, Dotnetnuke, SharePoint, PHP Nuke, CMS portal, Liferay,… hoặc tự xây dựng bằng những ngôn ngữ lập trình như .NET, PHP, JSP, Perl, Python.... Ngoài ra, hệ quản trị CSDL đóng một vai trò hết sức quan trọng trong việc lưu trữ và truy xuất dữ liệu Web một cách linh hoạt. Một số CSDL được sử dụng cho ứng dụng Web phổ biến hiện nay như Oracle, SQL Server, MySQL... [7,8].

Rủi ro của ứng dụng Web xuất phát từ mối đe dọa tấn công và lỗ hổng bảo mật trong các thành phần ứng dụng Web. Như vậy, đánh giá rủi ro ứng dụng Web cần xem xét đến các lỗ hổng bảo mật trong các thành phần ứng dụng Web, nghĩa là trong máy chủ Web, máy chủ ứng dụng, máy chủ CSDL. Ngoài ra, còn cần xem xét đến máy trạm (máy người dùng sử dụng trình duyệt) và kết nối giữa máy trạm và máy chủ Web.

B. Lỗ hổng bảo mật của ứng dụng Web

OWASP (Open Web Application Security Project) [9,10,12] là một dự án mở về bảo mật ứng dụng Web. Dự án này thường xuyên cập nhật danh sách 10 lỗ hổng bảo mật Web phổ biến nhất. Sau đây là 10 lỗ hổng điển hình nhất được công bố năm 2013 (được ký hiệu từ A.1 đến A.10):

A.1. Tiêm mã (Injection): Do sai sót trong nhập liệu, kẻ tấn công có thể lợi dụng sơ hở này để thực hiện các lệnh không mong muốn hay truy cập các dữ liệu bất hợp pháp.

A.2. Lỗi xác thực, quản lý phiên (Broken Authentication and Session Management): Lỗ hổng này cho phép kẻ tấn công lợi dụng để lấy mật khẩu, khóa hay chiếm phiên làm việc, để mạo danh phiên làm việc và danh tính của người dùng.

A.3. Tạo kịch bản chéo trang XSS (Cross Site Scripting): Lỗ hổng bảo mật này do thiếu cơ chế kiểm soát dữ liệu nhập vào. Điều này cho phép kẻ tấn công thực thi các kịch bản độc hại trên trình duyệt Web của nạn nhân.

A.4. Tham chiếu đối tượng trực tiếp không an toàn (Insecure Direct Object References): Nhà phát triển ứng dụng Web đưa ra tham chiếu đến một đối tượng bên trong ứng dụng như là một tập tin, một thư mục hay một khóa của CSDL. Nếu việc kiểm tra quá trình tham chiếu này không an toàn, kẻ tấn công có thể lợi dụng để tham chiếu đến các dữ liệu mà họ không có quyền truy cập.

A.5. Lỗi do cấu hình bảo mật (Security Misconfiguration): Để có mức an toàn cao, cần phải có một cấu hình được xác định là an toàn và cần triển khai cho khung ứng dụng, máy chủ ứng dụng, máy chủ CSDL, nền tảng,… các phương pháp bảo mật cần thiết, thống nhất và liên kết với nhau.

A.6. Lộ dữ liệu nhạy cảm (Sensitive Data Exposure): Các dữ liệu nhạy cảm như thông tin thẻ tín dụng, tài khoản ngân hàng, ID (định danh) để nộp thuế… không được lưu trữ và bảo vệ an toàn, kẻ tấn công có thể lấy cắp hoặc thay đổi những thông tin này. Dữ liệu nhạy cảm cần được lưu trữ, truyền tải, bảo vệ đúng cách, sử dụng mã hóa và sao lưu dữ liệu.

A.7. Thiếu kiểm soát truy nhập mức chức năng (Missing Function Level Access Control): Thiếu các điều khoản trong việc phân quyền quản trị các mức, dẫn đến việc kẻ tấn công có thể lợi dụng và truy ra các điểm yếu trên hệ thống, hoặc lợi dụng để leo thang đặc quyền.

A.8. Giả mạo yêu cầu chéo trang (Cross Site Request Forgery - CSRF): Lợi dụng sơ hở của người dùng, tin tặc có thể đánh lừa người dùng thực hiện các hành động nguy hiểm mà họ không hề hay biết, ví dụ chuyển tiền từ tài khoản người dùng sang tài khoản kẻ tấn công.

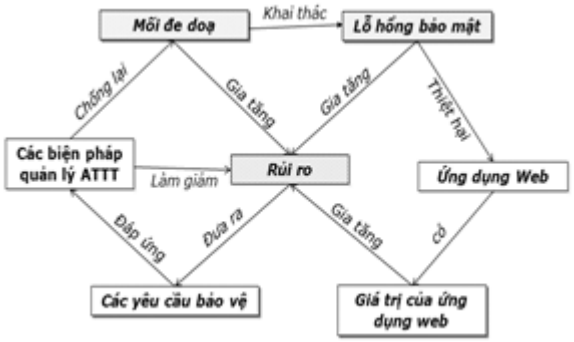
A.9. Sử dụng các thành phần sơ hở đã biết (Using Known Vulnerable Components): Việc sử dụng các thư viện, plugin, module,… có chứa các lỗ hổng đã được công khai, dễ dàng dẫn đến việc bị kẻ tấn công lợi dụng để tấn công vào hệ thống một cách nhanh chóng.

A.10. Chuyển hướng và chuyển tiếp sai (Unvalidated Redirects and Forwards): Việc chuyển hướng không an toàn có thể dẫn người dùng đến một đường dẫn độc hại bên ngoài, và bị kẻ tấn công lợi dụng để chuyển hướng nạn nhân đến một trang đích được chuẩn bị sẵn của kẻ tấn công.

III. MÔ HÌNH VÀ PHƯƠNG PHÁP ĐÁNH GIÁ RỦI RO ỨNG DỤNG WEB

A. Mối quan hệ giữa các thành phần

Kẻ tấn công thường tìm cách khai thác lỗ hổng bảo mật. Nếu hệ thống không được rà quét, loại bỏ các lỗ hổng, hạn chế các đe dọa tấn công, thì khả năng rủi ro đối với hệ thống sẽ tăng lên. Như vậy, lỗ hổng bảo mật, mối đe dọa tấn công và rủi ro là ba yếu tố có mối quan hệ chặt chẽ với nhau (Hình 2) [2].



Hình 2. Mối quan hệ giữa mối đe dọa, lỗ hổng bảo mật và rủi ro

Mức độ rủi ro của một ứng dụng Web cũng tỷ lệ thuận với giá trị tài sản của ứng dụng đó [2,3]. Ví dụ, một trang Web của ngân hàng sẽ tiềm ẩn nhiều rủi ro hơn là những trang Web phổ biến khác.

B. Mô hình đánh giá rủi ro ứng dụng Web

Như trên đã nêu, lỗ hổng bảo mật và mối đe dọa có tác động trực tiếp đến khả năng xảy ra sự cố, nghĩa là xảy ra rủi ro. Để đánh giá rủi ro, việc đầu tiên cần làm là xác định khả năng xảy ra sự cố đối với từng loại lỗ hổng bảo mật. Tiếp đó là xác định tác động khi sự cố đó xảy ra.

- *Định lượng khả năng xảy ra sự cố*

Đối với mỗi loại lỗ hổng khác nhau, khả năng bị khai thác sẽ khác nhau. Mô hình của bài báo này đưa ra mức định lượng cho từng khả năng xảy ra sự cố. Ứng với mỗi mức độ, có thể đánh giá giá trị là 1, 2 hoặc 3 (từ cao đến thấp) như sau:

- *Khả năng khai thác lỗ hổng bảo mật:* lỗ hổng của hệ thống có thể chia ra ba mức độ đánh giá tùy thuộc vào trình độ chuyên môn của kẻ tấn công: 1 - Khó khai thác; 2 - Không quá khó; 3 - Khai thác dễ dàng.

- *Khả năng phát hiện ra lỗ hổng:* lỗ hổng của hệ thống có thể đang tồn tại, nhưng không phải ai cũng có thể phát hiện ra. Có thể chia thành 3 mức: 1 - Rất khó phát hiện; 2 - Khó phát hiện; 3 - Phát hiện được bằng các công cụ thông thường hoặc thủ công.

- *Khả năng lỗ hổng đã được biết đến:* điều này cho biết mức độ phổ biến của lỗ hổng. Có thể đánh giá theo ba mức độ: 1 - Rất nhiều người biết (có thể nằm trong Top 10 của OWASP); 2 - Ít người biết; 3 - Rất hiếm gặp (rất ít người biết).

- *Khả năng phát hiện xâm nhập của hệ thống:* khi xảy ra tấn công, hệ thống có thể phát hiện kịp thời hay không. Có thể đánh giá theo 3 mức: 1 - Hệ thống phát hiện xâm nhập tức thời; 2 - Được ghi nhật ký và kiểm tra nhật ký; 3 - Được ghi nhật ký nhưng không kiểm tra hoặc không ghi nhật ký.

- *Định lượng mức tác động của sự cố*

Để đánh giá được mức tác động của một sự cố đối với thông tin, có thể phân ra thành ba loại là tác động đến: tính bí mật, tính toàn vẹn, tính sẵn sàng của dữ liệu.

- *Tác động đến tính bí mật:* tùy thuộc từng loại dữ liệu, ta có thể xác định mức bí mật để gán giá trị tương ứng. Cụ thể là: 1 - Dữ liệu yêu cầu bí mật; 2 - Dữ liệu hạn chế công bố; 3 - Dữ liệu không yêu cầu bí mật (ví dụ, một thông báo).

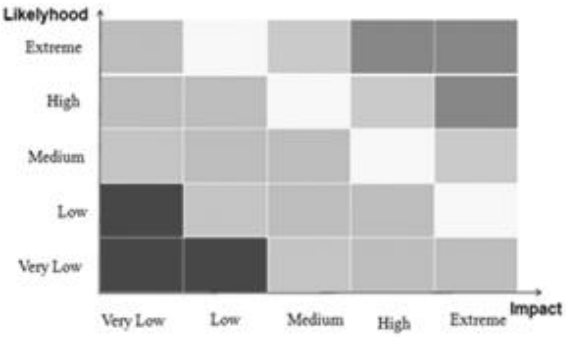
- *Tác động đến tính toàn vẹn:* căn cứ theo mức độ bị hư hại của dữ liệu để gán các giá trị tương ứng. Cụ thể là: 1 - Dữ liệu bị hỏng hoàn toàn (không thể khôi phục); 2 - Dữ liệu bị hỏng, nhưng có thể khôi phục một phần; 3 - Dữ liệu bị hỏng ít.

- *Tác động đến tính sẵn sàng:* Có thể phân chia làm hai loại dịch vụ (hoặc dữ liệu) là dịch vụ cốt yếu và dịch vụ không cốt yếu. Khi đó ta có các giá trị đánh giá như sau: 1 - Tất cả các dịch vụ bị gián đoạn; 2 - Dịch vụ cốt yếu bị gián đoạn; 3 - Dịch vụ không cốt yếu bị gián đoạn.

- *Công thức đánh giá rủi ro*

Đối với mỗi loại lỗ hổng bảo mật, ta có thể căn cứ vào các cách thức đánh giá định lượng nêu trên để xác định mức độ rủi ro tương ứng. Từ đó, ta có thể tính được mức rủi ro của toàn bộ ứng dụng Web (của cả hệ thống).

Như đã nêu, rủi ro đối với mỗi lỗ hổng bảo mật phụ thuộc vào hai yếu tố: khả năng xảy ra sự cố và tác động của sự cố. Hình 3 biểu diễn khả năng xác định rủi ro căn cứ theo hai trục số: trục tung là khả năng xảy ra sự cố và trục hoành là tác động khi xảy ra sự cố với 5 mức, rất thấp (very low), thấp (low), trung bình (medium), cao (high), rất cao (extreme).



Hình 3. Biểu đồ xác định mức rủi ro

Từ đó, ta có công thức tính rủi ro như sau:

$$R = L * I \quad (1)$$

Trong đó:

R = Thước đo mức độ rủi ro (Risk);

L = Thước đo khả năng xảy ra sự cố (Likelyhood);

I = Thước đo mức độ ảnh hưởng, hay tác động (Impact).

Ta sử dụng ký hiệu định lượng cho các giá trị về khả năng xảy ra sự cố và khả năng tác động cho từng lỗ hổng bảo mật như sau:

Khả năng xảy ra sự cố:

Gọi khả năng khai thác lỗ hổng là L_o, khả năng phát hiện ra lỗ hổng là L_d, khả năng lỗ hổng được biết đến là L_a, khả năng phát hiện xâm nhập là (L_n).

Từ đó, ta có thể suy ra công thức tính khả năng xảy ra sự cố đối với lỗ hổng thứ i như sau (là giá trị trung bình của 4 thành phần):

$$L_i = (L_o + L_d + L_a + L_n) / 4 \quad (2)$$

Tác động của sự cố:

Gọi tác động đến tính bí mật là I_c, tác động đến tính toàn vẹn là I_y, tác động đến tính sẵn sàng là I_a.

Từ đó ta có công thức tính mức độ tác động đối với lỗ hổng thứ i như sau:

$$I_i = (I_c + I_y + I_a) / 3 \quad (3)$$

Từ công thức (1), (2), (3) ta có mức độ rủi ro đối với lỗ hổng i là:

$$R_i = (L_i) * (I_i) \quad (4)$$

Hay có thể viết dưới dạng:

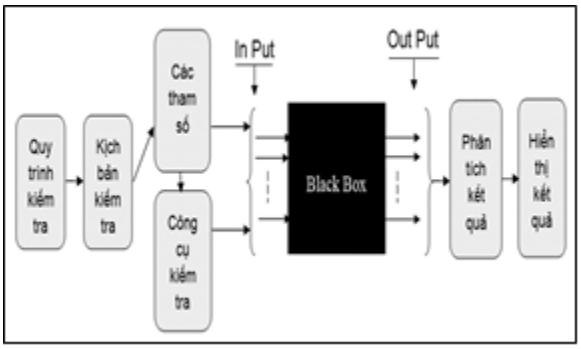
$$\log(R_i) = \log(L_i) + \log(I_i) \quad (5)$$

Tổng mức rủi ro của ứng dụng Web cho n lỗ hổng (theo tỉ lệ phần trăm) được tính như sau:

$$R = \frac{\sum_{i=0}^{R} R_i}{n}\% = \frac{\sum_{i=0}^{R} L_i \times I_i}{n}\%$$

C. Phương pháp đánh giá lỗ hổng ứng dụng Web bằng phương pháp hộp đen

Các phương pháp kiểm tra lỗ hổng bảo mật phổ biến hiện nay là: phương pháp kiểm tra hộp đen, hộp trắng và hộp xám. Mỗi phương pháp kiểm tra trên đều có những ưu và nhược điểm riêng.



Hình 4. Phương pháp kiểm tra hộp đen

Với phương pháp kiểm tra hộp đen, người kiểm tra hoàn toàn đứng trên quan điểm kẻ tấn công. Đây là một yêu cầu rất quan trọng trong quá trình kiểm tra, vì mục tiêu của việc kiểm tra là tìm ra những điểm yếu mà từ đó kẻ tấn công có thể xâm nhập vào hệ thống. Mặt khác, việc chi phí về thời gian cũng như tài chính sẽ nằm trong phạm vi cho phép đối với nhiều tổ chức. Vì vậy, phương pháp kiểm tra hộp đen phù hợp với phần lớn các hệ thống trong thực tế.

Phương pháp này gồm các bước sau:

Quy trình kiểm tra: giúp định hướng rõ ràng trong việc đánh giá, cho biết các công việc cụ thể cần tiến hành là gì, ai là người thực hiện công việc đó và thực hiện nó như thế nào. Việc xây dựng quy trình là bước đầu tiên trước khi tiến hành các công việc tiếp theo của quá trình đánh giá rủi ro cho ứng dụng Web. Quy trình được đánh giá khoa học, đầy đủ, chính xác sẽ giúp cho kết quả của quá trình đánh giá tốt, tiết kiệm được thời gian và chi phí.

Kịch bản kiểm tra: Kịch bản được hiểu như một phác thảo các hành động cần thiết để tiến hành kiểm tra một hoặc nhiều lỗ hổng của hệ thống. Kịch bản được xây dựng trên cơ sở dự đoán những phản ứng của ứng dụng Web khi có tác động từ phía tác nhân.

Các tham số truyền vào: các tham số được lựa chọn có trong kịch bản kiểm tra. Các tham số này có thể được truyền trực tiếp tới giao diện đầu vào (input) từ người đánh giá hoặc có thể thông qua các công cụ rà quét lỗ hổng bảo mật. Ví dụ người đánh giá có thể nhập vào một tham số là đoạn script vào ô nhập liệu để xem phản hồi của một ứng dụng Web:

<script>alert('Lo hong XSS')</script>

Công cụ kiểm tra: là phần mềm được thiết kế các chức năng kiểm tra theo các kịch bản có sẵn. Nó có thể nhận những phản hồi từ phía hệ thống được kiểm tra, sau đó phân tích và đưa ra những cảnh báo về các lỗ hổng đang tồn tại, nó có thể xuất ra báo cáo chi tiết kèm theo các khuyến nghị, phương pháp khắc phục của từng lỗ hổng.

Phân tích và hiển thị kết quả: sau khi nhận được kết quả kiểm tra sẽ tiến hành phân tích, đánh giá, phân loại và tính toán ra mức độ rủi ro theo từng lỗ hổng. Kết quả tính toán rủi ro sẽ được hiển thị dưới dạng các báo cáo, cho thấy được mức chi tiết hoặc tổng quan.

D. Phân tích lỗ hổng bảo mật của các thành phần ứng dụng Web

- *Lỗ hổng bảo mật của hệ điều hành*

Trong các máy chủ Web và máy trạm, hệ điều hành (HĐH) được sử dụng phổ biến là Linux (36,72%), Windows (33,10%) [13]. Theo thống kê hàng năm của CVE Details [13] lỗ hổng bảo mật trên HĐH Windows luôn chiếm số lượng lớn hơn hẳn các HĐH khác, trong số đó có rất nhiều lỗ hổng nghiêm trọng được đánh giá mức 9,3/10 (mức 10 là nghiêm trọng nhất).

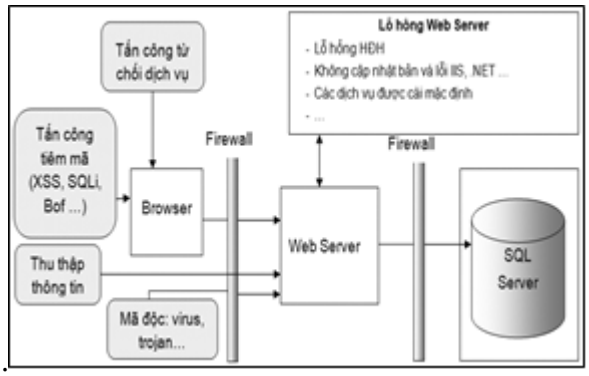
Các loại lỗ hổng bảo mật trên hệ điều hành máy chủ gồm:

- Không cập nhật bản vá lỗi;
- Nhiều dịch vụ không cần thiết, song vẫn được cài đặt;
- Nhiều giao thức không cần thiết, nhưng vẫn bị kích hoạt;
- Quản lý tài khoản không chặt chẽ;
- Thiếu cơ chế bảo vệ các tệp và các thư mục;
- Mở những cổng không cần thiết;
- Thiếu cơ chế ghi nhật ký (log) và cơ chế kiểm toán (audit).

Phía máy trạm cũng tồn tại nhiều lỗ hổng bảo mật tương tự như các máy chủ.

- *Lỗ hổng bảo mật máy chủ Web*

Ngoài những lỗ hổng của HĐH đã nêu trên, máy chủ Web thông dụng như IIS, Apache HTTP, Apache Tomcat,… còn có thể tồn tại các lỗ hổng của chính những dịch vụ được cài đặt trên đó. Ví dụ, máy chủ IIS 7.0 tồn tại lỗ hổng nguy hiểm MS10-040, khai thác lỗi tràn bộ đệm, giúp kẻ tấn công có thể tấn công chiếm quyền điều khiển hệ thống [12,14].



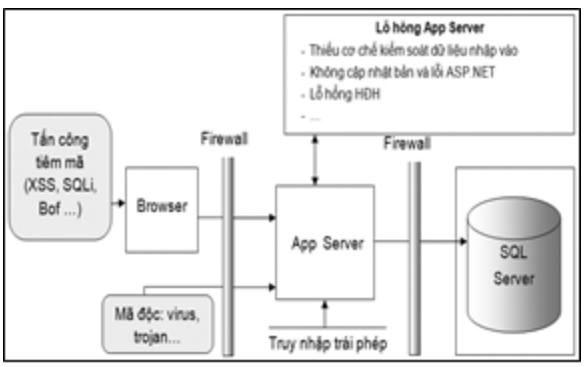
Hình 5. Các lỗ hổng và mối đe dọa đối với máy chủ Web

Máy chủ Web tiếp xúc trực tiếp với môi trường Internet nên thường xuyên phải đối mặt với các nguy cơ như: tấn công từ chối dịch vụ, mã độc, tấn công tiêm mã,… (Hình 5). Để đảm bảo an toàn cho máy chủ, chỉ nên kích hoạt những dịch vụ cần thiết.

- *Lỗ hổng bảo mật máy chủ ứng dụng*

Theo mô hình thiết kế ba lớp (Hình 6), máy chủ này được đặt ở giữa lớp một (máy chủ Web) và lớp ba (máy chủ CSDL).

Trong mô hình này, máy chủ ứng dụng không tiếp xúc trực tiếp với Internet nên có thể hạn chế được một số tấn công trực tiếp bên ngoài, nhưng vẫn phải đối mặt với các nguy cơ khác từ bên trong hoặc từ các lỗ hổng phát sinh trong quá trình phát triển ứng dụng. Ví dụ, SQL injection, CSRF, LFI, XSS....



Hình 6. Lỗ hổng và mối đe dọa đối với máy chủ ứng dụng

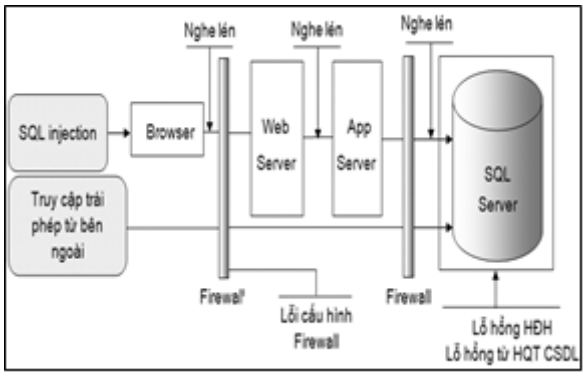
Ngoài ra, máy chủ ứng dụng còn phải đối mặt với các nguy cơ bên trong hệ thống mạng, như truy nhập trái phép, nghe lén trên mạng….

- *Lỗ hổng máy chủ cơ sở dữ liệu*

Máy chủ CSDL được cài đặt các phần mềm như Hệ quản trị CSDL, như: SQL Sever, MySQL, Oracle, Postgres.... Đây là những phần mềm có chức năng lưu trữ và cung cấp dữ liệu theo yêu cầu của các ứng dụng.

Ngoài các lỗ hổng thuộc HĐH, máy chủ CSDL có thể xuất hiện những lỗ hổng xuất phát từ Hệ quản trị CSDL như: mở các cổng mặc định còn lưu các CSDL mẫu trên hệ thống, lưu các dịch vụ mặc định.

Mật khẩu mặc định của hệ thống luôn được kích hoạt trên hệ thống, nếu là mật khẩu yếu sẽ là một lỗ hổng rất lớn.



Hình 7. Lỗ hổng và mối đe dọa đối với máy chủ CSDL

- *Lỗ hổng bảo mật trong kết nối*

Lỗ hổng bảo mật trong quá trình kết nối có thể chia thành hai loại: kết nối giữa hai thiết bị máy chủ, kết nối giữa máy chủ và máy trạm.

Theo Hình 7, máy chủ ứng dụng kết nối với máy chủ Web và máy chủ CSDL. Khi người dùng truy cập ứng dụng Web, họ có thể gửi thông tin xác thực, thông tin tài khoản ngân hàng, hoặc bất kỳ thông tin nào từ trình duyệt đến máy chủ. Các thông tin nhạy cảm được truyền dưới dạng bản tin rõ có thể bị nghe lén. Các giao dịch giữa máy trạm (trình duyệt) với máy chủ Web cũng có nguy cơ bị nghe lén. Kiểu tấn công này còn được gọi là tấn công người đứng giữa (Man-in-the-middle). Để đảm bảo an toàn kênh kết nối giữa các máy chủ, thông tin trên đường truyền cần được mã hoá (có thể sử dụng IPsec). Đối với kênh liên lạc giữa máy trạm với máy chủ cần sử dụng giao thức HTTPS.

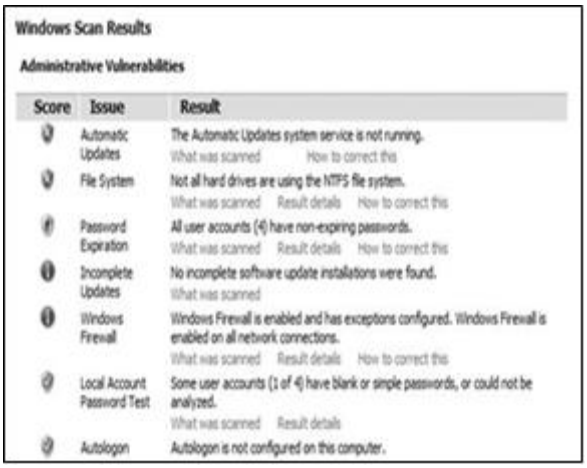
IV. KẾT QUẢ THỬ NGHIỆM ĐÁNH GIÁ RỦI RO ỨNG DỤNG WEB

A. Công cụ sử dụng cho kiểm tra đánh giá lỗ hổng bảo mật ứng dụng Web

Hiện nay, có nhiều khả năng lựa chọn các bộ công cụ phần mềm có sẵn để kiểm tra lỗ hổng bảo mật, từ đó đánh giá rủi ro cho ứng dụng Web. Các bộ công cụ có thể là sản phẩm thương mại hoặc miễn phí. Nhiều công cụ tuy miễn phí song cũng có những tính năng khá mạnh. Trong phần này, chỉ giới thiệu tóm tắt một số công cụ phổ biến thường được sử dụng, được sử dụng trong bài để kiểm thử ứng dụng Web.

- *Công cụ kiểm tra lỗ hổng hệ thống*

Công cụ MBSA: MBSA (Microsoft Baseline Security Analyzer) được phát hành miễn phí bởi Microsoft dành cho HĐH Windows, với dung lượng chỉ 2MB. Giao diện của MBSA là rất đơn giản với hai lựa chọn là quét một máy tính và quét nhiều máy tính. Với công cụ này, ta có thể kiểm tra quyền quản trị trên máy tính, độ an toàn của mật khẩu, trạng thái tường lửa trên máy tính, tình trạng file hệ thống, trạng thái cập nhật bản vá lỗi, kiểm tra lỗ hổng IIS, kiểm tra SQL. Quá trình kiểm tra diễn ra rất nhanh và chính xác. Hình 8 là ví dụ minh họa về kết quả quét hệ thống với MBSA.



Hình 8. Kết quả kiểm tra với công cụ MBSA

Công cụ Nmap: Nmap là một công cụ quét cổng (scan port) rất mạnh. Nó hỗ trợ toàn bộ các phương thức quét cổng. Công cụ này giúp cho việc kiểm tra các cổng trên hệ thống đang được mở. Những cổng mở mà không được sử dụng chính là các điểm yếu của hệ thống cần được khắc phục.

Nmap Scan	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	-sS	YES	YES	NO
TCP connect() Scan	-sT	NO	YES	NO
FIN Scan	-sF	YES	YES	NO
Xmas Tree Scan	-sX	YES	YES	NO
Null Scan	-sN	YES	YES	NO
Ping Scan	-sP	NO	NO	NO
Version Detection	-sV	NO	NO	NO
UDP Scan	-sU	YES	NO	YES
IP Protocol Scan	-sO	YES	NO	NO
ACK Scan	-sA	YES	YES	NO
Window Scan	-sW	YES	YES	NO
RPC Scan	-sR	NO	NO	NO
List Scan	-sL	NO	NO	NO
Idlescan	-sI	YES	YES	NO
FTP Bounce Attack	-b	NO	YES	NO

Hình 9. Kết quả kiểm tra với công cụ Nmap

Công cụ Nessus: Đây là một công cụ có thể chạy trên nhiều HĐH khác nhau, như: Windows, Linux, UNIX. Nó được đánh giá là một trong

những công cụ mạnh trong việc kiểm tra các lỗ hổng của HĐH và dễ dàng sử dụng.

- *Các công cụ kiểm tra lỗ hổng ứng dụng Web*

Hình 10. Kết quả kiểm tra với Acunetix

Công cụ Acunetix: Đây là công cụ thương mại với các khả năng kiểm tra lỗi bảo mật rất mạnh và được xếp thứ 3 trong số các công cụ thương mại. Acunetix hỗ trợ tất cả các phương thức kiểm tra như kiểm tra thủ công hoặc kiểm tra tự động. Công cụ này cho phép quét thông qua máy trạm với mọi ứng dụng Web. Sau khi hoàn tất việc rà quét, nó xuất ra một tập tin báo cáo rõ ràng, đồng thời có kèm theo mô tả chi tiết về lỗ hổng và cách thức khắc phục. Ta có thể lưu lại báo cáo này để phục vụ cho việc phân tích. Mức độ bảo mật của Web được đánh giá từ thấp (low), trung bình (medium), cao (high).

Công cụ AppScan: AppScan là công cụ thương mại rất mạnh dùng để quét các lỗ hổng bảo mật ứng dụng Web. Công cụ này có thể rà quét được những ứng dụng phát triển trên nền tảng công nghệ khác nhau, mạnh hơn Acunetix đối với việc quét các ứng dụng phát triển bằng PHP. Ví dụ, với lỗ hổng Local File Inclusion, nếu quét bằng Acunetix thì độ chính xác là 57,35% , còn nếu quét bằng công cụ AppScan thì xấp xỉ 100%. AppScan là một công cụ dễ sử dụng, sau khi rà quét có thể đưa ra những báo cáo rõ ràng và chi tiết. Công cụ này đứng đầu trong bảng xếp hạng về tính năng trong số các công cụ thương mại.

B. Kết quả đánh giá rủi ro ứng dụng Web

Sau khi lựa chọn công cụ như đã nêu ở mục IV.A, phần này trình bày kết quả triển khai đánh giá cho một ứng dụng Web cụ thể. Trang Web thực tế được dùng cho thử nghiệm đánh giá (tạm dùng tên *http://www.xyz.com*) là một trang Web thương mại. Mọi chức năng an toàn đều cần được bảo vệ ở mức cao, đặc biệt là thông tin về tài khoản của khách hàng, CSDL, tính ổn định của dịch vụ. Những yêu cầu này sẽ làm căn cứ để xác định các trọng số trong tính toán rủi ro.

Ứng dụng Web được triển khai theo mô hình 3 lớp, các máy chủ ứng dụng, máy chủ CSDL được bảo vệ bởi lớp tường lửa bên trong và đưa ra hệ thống phát hiện tấn công.

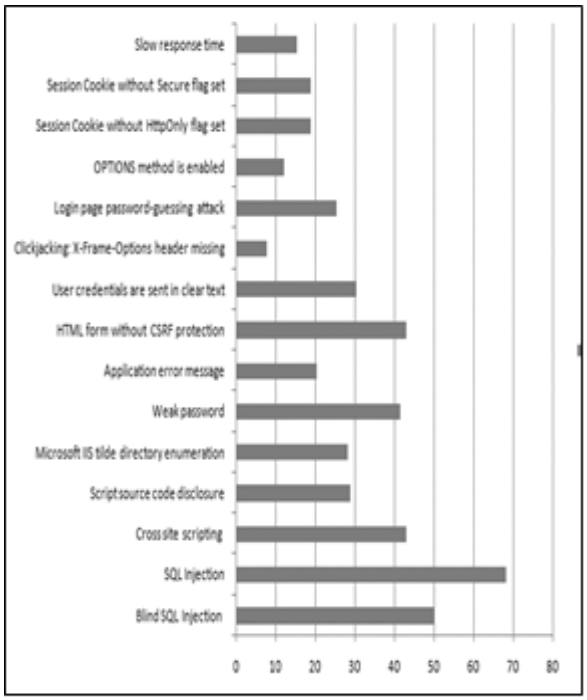
Sau khi rà quét bằng công cụ Acunetix ta được kết quả là danh sách gồm 15 loại lỗ hổng, mỗi loại có thể xuất hiện tại nhiều vị trí của ứng dụng. Kết quả tổng hợp về tác động, khả năng xảy ra sự cố và mức độ rủi ro của từng lỗ hổng bảo mật thu được sau khi phân tích được thể hiện trong Bảng 1 như sau:

BẢNG 1. MỨC RỦI RO CỦA TỪNG LỖ HỔNG BẢO MẬT

STT	Lỗ hổng	Tác động	Khả năng sự cố	Rủi ro (%)
1	Blind SQL Injection	5,57	9	50,13
2	SQL Injection	7,57	9	68,13
3	Cross site scripting	7,14	6	42,84
4	Script source code disclosure	5,43	5,33	28,94
5	Microsoft IIS tilde directory enumeration	5,29	5,33	28,20
6	Weak password	7.29	5,67	41,33
7	Application error message	5,57	3,67	20,44
8	HTML form without CSRF protection (CSRF)	5,86	7,33	42,95
9	User credentials are sent in clear text	7	4,33	30,31
10	Clickjacking: X-Frame-Options header missing	3,86	2	7,72
11	Login page password-guessing attack	5,86	4,33	25,37
12	OPTIONS method is enabled	3,29	3,67	12,07
13	Session Cookie without HttpOnly flag set	4,71	4	18,84
14	Session Cookie without Secure flag set	4,71	4	18,84
15	Slow response time	3,86	4	15,44

Theo công thức trong mục III.B, ta tính được tổng rủi ro cho ứng dụng Web là:

R = (50,13 + 68,13 + 42,84 + 28,94 + 28,2 + 41,33 + 20,44 + 42,95 + 30,31 + 7,72 + 25,37 + 12,07 + 18,84 + 18,84 + 15,44): 15= 30,1%.



Hình 11. Biểu đồ tổng hợp mức rủi ro

Như trong Bảng 1 và Hình 11, ta có một số kết quả cụ thể như sau:

- Mức rủi ro cao nhất thuộc lỗ hổng SQL Injection: 68,13%;
- Mức rủi ro thấp nhất là lỗ hổng Clickjacking: 7,72 %;
- Các lỗ hổng cao mức thứ 2: Blind SQL Injection (50,13%); Cross site scripting (42,84%); CSRF (42,95%); Weak Password (41,33%);
- Các lỗ hổng còn lại ở mức trung bình và thấp;

Như vậy, đối với ứng dụng Web đã nêu, mức rủi ro là 30,1%. Trong đó có những lỗ hổng nghiêm trọng, có mức rủi ro cao, như SQL Injection. Mức rủi ro của ứng dụng này có thể so sánh với tập các kết quả rủi ro của ứng dụng Web khác trong cùng lĩnh vực, hoặc có thể mở rộng hơn. Chuyên gia đánh giá có thể chủ động bổ sung thêm tiêu chí đánh giá để tăng tính chính xác của kết quả. Ta có thể kết luận rằng ứng dụng Web này có mức bảo mật Chưa đạt (trong số các mức Chưa đạt - Đạt - Trung bình - Khá - Tốt - Rất tốt).

V. KẾT LUẬN

Bài báo đã trình bày một mô hình và phương pháp đánh giá định lượng rủi ro trên cơ sở đánh giá khả năng sự cố và tác động của nó đối với ứng dụng Web. Qua phân tích cho thấy, lỗ hổng bảo mật và mối đe dọa có tác động trực tiếp đến khả năng xảy ra sự cố, nghĩa là xảy ra rủi ro. Để đánh giá rủi ro, việc đầu tiên cần làm là xác định khả năng xảy ra sự cố đối với từng loại lỗ hổng bảo mật. Tiếp đó là xác định tác động khi xảy ra sự cố đó. Bên cạnh đó, bài báo cũng đưa ra cách tính giá trị cho các mức khả năng sự cố và mức tác động. Các giá trị này có thể đặt theo ngưỡng và có thể điều chỉnh với trọng số phù hợp. Kết quả thử nghiệm đánh giá rủi ro cho một ứng dụng Web cho thấy mô hình đề xuất và cách tính trong bài báo là khả thi với chi phí chấp nhận được đối với các yêu cầu thực tế.

TÀI LIỆU THAM KHẢO

[1]. H. Joh, Y.K. Malaiya, "Defining and Assessing Quantitative Security Risk Measures Using Vulnerability Lifecycle and CVSS Metrics", Int'l Conf. Security and Management | SAM'11, pp. 10-16, 2011.

[2]. National Institute of Standards and Techno-logy (NIST), "Risk management guide for information technology systems". Special Publication 800-30, 2001.

[3]. L. A. Cox, "Some Limitations of Risk = Threat Vulnerability Consequence for Risk Analysis of Terrorist Attacks, Risk Analysis", 28(6), pp. 1749-1761, 2008.

[4]. H. Joh and Y. K. Malaiya, "A Framework for Software Security Risk Evaluation using the Vulnerability Lifecycle and CVSS Metrics", Proc. International Workshop on Risk and Trust in Extended Enterprises, pp. 430-434, November 2010.

[5]. P. Mell, K. Scarfone, and S. Romanosky, CVSS: "A complete Guide to the Common Vulnerability Scoring System Version 2.0", Forum of Incident Response and Security Teams (FIRST), 2007.

[6]. http://en.wikipedia.org/wiki/Web_application

[7]. Madeyski, "Architectural design of modern Web application", madeyski.einformatyka.pl/download/23.pdf

[8]. D.Nelson, "Next Gen Web Architecture for the Cloud Era", http://www.sei.cmu.edu/library /assets/ pre-sentations/ nelson-saturn2013.pdf .

[9]. Open Web Application Security Project (OWASP) Top 10 2014 - The Ten Most Critical Web Application Security Risks, http://ww w.owasp.org/index.php/

[10]. The Open Web Application Security Project (2013) OWASP_ Testing_ Guide _v4.

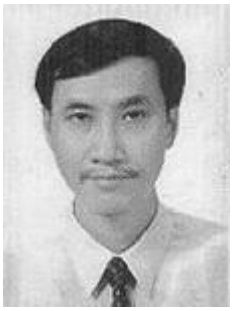
[11]. Microsoft, Improving Web Application Security, http://msdn.micro-soft.com/en-us/library/ff648657.aspx

[12]. https://www.owasp.org/index.php/OWASP Risk Rating Methodology

[13]. http://www.cvedetails.com/index. php

[14]. https://www.owasp.org/index.php/Category: Vulnerability.

SƠ LƯỢC VỀ TÁC GIẢ



PGS.TSKH. Hoàng Đăng Hải

Đơn vị công tác: Học viện Công nghệ Bưu chính viễn thông, Bộ Thông tin và Truyền thông, Hà Nội.

Email: hdhai@ptit.edu.vn

Tốt nghiệp kỹ sư chuyên nghành Tin học và Điều khiển tại Đức năm 1984. Được phong hàm Phó giáo sư năm 1999. Nhận bằng Tiến sỹ khoa học năm 2003.

Hướng nghiên cứu hiện nay: An toàn thông tin, mạng truyền thông, công nghệ mạng thế hệ mới.

ThS. Hồ Kim Cường

Đơn vị công tác: Trung tâm VNCERT, Bộ Thông tin và Truyền thông, Hà Nội.

Email: hkcuong@vncert.vn

Tốt nghiệp chuyên ngành Công nghệ thông tin, Học viện Kỹ thuật Quân sự năm 2009. Nhận bằng Thạc sĩ chuyên ngành Hệ thống thông tin của Học viện Công nghệ Bưu chính viễn thông năm 2015.

Hướng nghiên cứu hiện nay: Bảo mật ứng dụng web, mã độc, chính sách về an toàn thông tin.

"Journal of Science and Technology on Information security"

(Version No. II of the *Information Security Journal*)

Information Security Journal publishes a periodical academic – scientific journal in the field of Information Security with the title "Research of Science and Technology in the field of Information Security". The Publication aims to create a forum for exchange of specialized scientific – technological issues in the field of Information Security, to support the research of science and technology, contributing to connecting research, trainning and applications deployement.

The Journal is published in Vietnamese and English, issued 2 editions / year in the whole territory of Vietnam, to the leaders, managers, scientific – technical staff, teachers, graduate students, the fellows,... such people who are conducting the activitiesin the field of Information Security in the country.

The papers that are published in journal are the scientific research works, applications of new technologies, scientific achievements, new techniques in the field of secrecy and information security, which have not been sent to any magazine to be published or to any conference proceedings and must be of scientific quality, which have been appraised and assessed by the experts in order to be counted points by the State's scientific Councils for the converted scientific works.