ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN HẠNH PHÚC

MỘT SỐ PHƯƠNG PHÁP ĐẢM BẢO CHẤT LƯỢNG HỆ THỐNG PHẦN MỀM TRÊN NỀN WEB
(Tóm tắt)

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội - 2023

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN HẠNH PHÚC

MỘT SỐ PHƯƠNG PHÁP ĐẢM BẢO CHẤT LƯỢNG HỆ THỐNG PHẦN MỀM TRÊN NỀN WEB
(Tóm tắt)

Chuyên ngành: Kỹ thuật phần mềm
Mã số: 9480103.01

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PGS.TS. Trương Ninh Thuận

Hà Nội - 2023

Chương 1

GIỚI THIỆU

1.1. Đặt vấn đề

Các hệ thống phần mềm nền Web hiện đang được sử dụng sâu rộng trong nhiều lĩnh vực của đời sống xã hội. Mọi lĩnh vực quan trọng như quân sự, kinh tế, y tế, giáo dục, v.v. cũng đều sử dụng phần mềm trong các công việc. Vì vậy, chất lượng của phần mềm nền Web có ảnh hưởng trực tiếp hoặc gián tiếp đến người dùng hệ thống.

Bên cạnh các ưu điểm, các ứng dụng web cũng có những nhược điểm như: Do ứng dụng web chạy trên môi trường internet, người dùng phải có kết nối internet để truy cập. Tốc độ của web cũng thường ít được tối ưu tốt và có thể chậm hơn so với ứng dụng truyền thống, đặc biệt khi có quá nhiều người truy cập cùng lúc. Bảo mật dữ liệu cũng là một vấn đề đáng lo ngại. Do dữ liệu lưu trữ trên server nên hoàn toàn có thể bị truy cập trái phép. Ngoài ra, ứng dụng web phải tương thích với nhiều nền tảng và trình duyệt khác nhau, làm cho quá trình phát triển và kiểm thử phức tạp hơn. Các thiết bị hoặc trình duyệt đời cũ hoặc ít phổ biến cũng có thể gây khó khăn trong việc sử dụng ứng dụng web.

Để đảm bảo chất lượng của các phần mềm trên nền web, luận án tập trung đến các tính chất sẵn sàng, bảo mật, toàn vẹn của hệ thống. Luận án **Một số phương pháp đảm bảo chất lượng hệ thống phần mềm trên nền Web"** đề ra các mục tiêu nghiên cứu: *i. Nghiên cứu các vấn đề liên quan đến tính an ninh (tính bảo mật, tính toàn vẹn, v.v) của ứng dụng web. Từ đó, đề xuất, xây dựng các phương pháp nhằm hỗ trợ quá trình kiểm thử để phát hiện, cảnh báo một số dạng tấn công, lỗ hổng phổ biến trong ứng dụng web*; *ii. Phát triển các công cụ để hỗ trợ quá trình kiểm thử tự động.*

1.2. Nội dung nghiên cứu

Các nội dung chính của luận án bao gồm:

- Nghiên cứu về các dạng tấn công vào hệ thống phần mềm trên nền web.

- Đề xuất phương pháp phân tích nhật ký truy cập của một hệ thống web: Tìm hiểu cách thức thu thập và phân tích dữ liệu nhật ký truy cập của hệ thống web, nhằm phát hiện ra các dấu hiệu bất thường của người dùng có thể dẫn đến tấn công DoS, DDoS. Giúp hệ thống phát hiện sớm các hành vi đáng ngờ và đưa ra cảnh báo kịp thời cho những người quản trị hệ thống để họ có thể đưa ra các biện pháp phòng ngừa và giảm thiểu rủi ro.

- Nghiên cứu về kiến trúc hướng sự kiện: Tìm hiểu kiến trúc hướng sự kiện (EDA) và các phương pháp để kiểm chứng tiến trình thực thi các sự kiện trong kiến trúc này. Tập trung vào việc đảm bảo tính toàn vẹn và chính xác của các sự kiện được sinh ra trong hệ thống thời gian thực.

- Đề xuất các thuật toán để sinh và tối ưu đường kiểm thử lỗ hổng XSS trên ứng dụng web: Sử dụng phương pháp học tăng cường Q-learning để tối ưu hóa quá trình sinh đường kiểm thử lỗ hổng XSS trên ứng dụng web. Các thuật toán được đề xuất giúp nâng cao chất lượng của phần mềm trên nền web và đảm bảo an toàn thông tin cho người dùng.

1.3. Đóng góp của luận án

Sau quá trình giải quyết bài toán với mục tiêu, đối tượng và phương pháp nghiên cứu đã đề ra, luận án có các đóng góp chính sau đây:

1. *Đề xuất phương pháp phân tích nhật ký truy cập của một hệ thống web theo theo thời gian thực nhằm phát hiện các dấu hiệu bất thường của người dùng có thể dẫn đến tấn công DoS, DDoS.*

2. *Đề xuất một phương pháp kiểm chứng tiến trình xuất hiện của các sự kiện trong kiến trúc hướng sự kiện (EDA) trong thời gian thực thi.* Phương pháp được giới thiệu là một cách tiếp cận để kiểm chứng tiến trình thực thi các sự kiện trong EDA.

3. *Đề xuất phương pháp nhằm giải quyết vấn đề tối ưu hóa các đường kiểm thử trong kiểm thử lỗ hổng XSS trên các ứng dụng web.* Phương pháp này sẽ cải thiện chất lượng và hiệu quả của quá trình kiểm thử lỗ hổng XSS bằng cách tối ưu hóa số lượng và chất lượng các đường kiểm thử được sinh ra. Kết quả đạt được của phương pháp này sẽ hỗ trợ tốt cho việc sinh các ca kiểm thử và cải thiện khả năng phát hiện các lỗ hổng XSS trên các ứng

dụng web, giúp đảm bảo an toàn cho người dùng và bảo vệ thông tin cá nhân.

1.4. Cấu trúc luận án

Luận án *"Một số phương pháp đảm bảo chất lượng hệ thống phần mềm trên nền Web"* bao gồm 6 chương. Trong đó, Chương 1 *Giới thiệu* trình bày về lý do thực hiện đề tài, đối tượng, phạm vi, nội dung nghiên cứu, các đóng góp và cấu trúc của luận án. Các chương tiếp theo của luận án được có nội dung lần lượt như sau:

Chương 2 *Kiến thức cơ sở* trình bày về các kiến thức nền tảng được sử dụng trong các chương tiếp theo. Chương 3 *Phân tích nhật ký truy cập theo thời gian thực để phòng ngừa tấn công DDoD* đề xuất phương pháp phân tích nhật ký truy cập của một hệ thống web nhằm phát hiện các dấu hiệu bất thường của người dùng có thể dẫn đến tấn công DDoS.

Chương 4 *Kiểm chứng tiến trình xuất hiện của các sự kiện tại thời điểm thực thi* giới thiệu một phương pháp để kiểm chứng tiến trình xuất hiện của các sự kiện trong một kiến trúc hướng sự kiện (EDA) thời điểm thực thi. Chương 5 *Kiểm thử fuzz các ứng dụng web*. Giới thiệu về bài toán nghiên cứu, phương pháp đề xuất và các bước thực hiện của nó.

Cuối cùng là Chương 6 *Kết luận và hướng phát triển*. Chương này tiến hành phân tích về các ưu, nhược điểm của từng phương pháp đã đề xuất và so sánh với một số phương pháp nghiên cứu liên quan. Từ đó, luận án thảo luận về các hướng nghiên cứu tiếp theo trong tương lai.

Chương 2

KIẾN THỨC CƠ SỞ

Trong chương này, luận án sẽ trình bày về những kiến thức cơ sở được sử dụng trong các chương tiếp theo.

2.1. Kiểm thử phần mềm

2.1.1. Khái niệm

Kiểm thử phần mềm (testing) là quá trình thử nghiệm phần mềm bằng các test case để đảm bảo tính đúng đắn và chất lượng của phần mềm. Quá trình này liên quan đến các khái niệm như lỗi, sai sót, thất bại và sự cố. Mục đích của kiểm thử là hai điều: tìm và báo cáo lỗi để sửa chữa, hoặc chứng minh rằng phần mềm đang hoạt động chính xác theo các yêu cầu đã đề ra.

Để đảm bảo chất lượng phần mềm trên nền web, có nhiều phương pháp kiểm thử ứng dụng web được đề xuất.

2.1.2. Kiểm thử thâm nhập

Kiểm thử thâm nhập là một loại kiểm thử bảo mật để khám phá các lỗ hổng, mối đe dọa, rủi ro trong các ứng dụng phần mềm, mạng hoặc ứng dụng web mà kẻ tấn công có thể khai thác. Mục đích của kiểm thử thâm nhập là để tìm ra tất cả các lỗ hổng bảo mật trong hệ thống đang được kiểm thử.

2.1.3. Kiểm thử fuzz

Fuzz testing hoặc fuzzing là một phương pháp kiểm tra phần mềm tự động đưa các đầu vào không hợp lệ, không đúng định dạng hoặc không mong muốn vào hệ thống để phát hiện các lỗi và lỗ hổng phần mềm. Một công cụ làm mờ đưa các đầu vào này vào hệ thống và sau đó theo dõi các trường hợp ngoại lệ như sự cố hoặc rò rỉ thông tin. Nói một cách đơn giản hơn, fuzzing đưa các đầu vào không mong muốn vào một hệ thống và theo dõi để xem liệu hệ thống có bất kỳ phản ứng tiêu cực nào đối với các đầu vào cho thấy lỗ hổng hoặc vấn đề về bảo mật, hiệu suất hoặc chất lượng hay không.

2.2. An ninh phần mềm

Phần mềm đang được sử dụng rộng rãi trong nhiều lĩnh vực của đời sống xã hội, văn hóa, v.v. Do đó, chất lượng của phần mềm có ảnh hưởng trực tiếp hoặc/và gián tiếp đến người sử dụng. Bên cạnh những lợi ích mà phần mềm mang lại, thì cũng xuất hiện nhiều vấn đề vi phạm truy cập tài nguyên làm ảnh hưởng đến chất lượng phần mềm và người dùng trong hệ thống. Một số sự cố an ninh phổ biến hay được nhắc đến là thất thoát hoặc làm sai lệch các thông tin riêng tư, quan trọng do hệ thống quản lý.

Một số tính chất quan trọng khác của phần mềm liên quan đến tính chất an ninh là tính tin cậy (*dependability*), tính đúng đắn (*correctness*), tính dự đoán (*predictability*), độ tin cậy (*reliability*) và tính an toàn (*safety*). Những tính chất này không ảnh hưởng trực tiếp, nhưng sự liên quan được mô tả và hiểu theo cách chúng ảnh hưởng đến những tính chất an ninh cốt lõi. Những tính chất này bị ảnh hưởng bởi kích thước, độ phức tạp, khả năng lần vết của phần mềm. Nhiều hoạt động của kỹ thuật an ninh phần mềm tập trung vào việc xử lý các tính chất hướng đến các tính chất an ninh cốt lõi.

2.3. Một số loại tấn mạng phổ biến

2.3.1. Tấn công vào lỗ hổng XSS

Cross Site Scripting (XSS) [**?**] là một trong những lỗ hổng phổ biến và dễ bị tấn công nhất mà tất cả những người thử nghiệm có kinh nghiệm đều biết. XSS là một lỗ hổng liên quan đến lập trình xảy ra khi dữ liệu đầu vào của người dùng không được làm sạch đúng cách. Kẻ tấn công khai thác lỗ hổng này để đưa mã tập lệnh chưa được lọc vào ứng dụng web, dẫn đến việc chiếm đoạt tài khoản, đánh cắp section hoặc cookie và định tuyến lại trang web của kẻ tấn công khi trình phân tích cú pháp xử lý tập lệnh.

2.3.2. Tấn công từ chối dịch vụ

DDoS (Distributed Denial-of-Service) là kỹ thuật tấn công từ chối dịch vụ phân tán mà ở đó kẻ xấu sẽ điều khiển mạng BotNet để tấn công vào một mục tiêu máy chủ. Gây ra tình trạng từ chối dịch vụ của máy chủ victim.

Một cuộc tấn công DDOS với quy mô lớn, tấn công vào những dịch vụ đang chạy của hệ thống máy tính nạn nhân hoặc nhằm vào tài nguyên mạng bằng

cách khởi tạo cuộc tấn công trung gian từ các hệ thống máy tính BotNet (là các máy chủ bị nhiễm trojan, virus hoặc bị chiếm quyền điều khiển) trên không gian Internet.

2.4. Kiến trúc hướng sự kiện

Kiến trúc hướng sự kiện (EDA - Event Driven Architecture) là một mô hình thiết kế phần mềm cho phép các tổ chức phát hiện và xử lý các sự kiện hoặc các tình huống quan trọng (ví dụ như giao dịch, truy cập trang web, giỏ hàng bị bỏ qua, v.v.) theo thời gian thực hoặc gần thời gian thực. Đây là một phương pháp thay thế kiến trúc truyền thống "request/response" ("yêu cầu/phản hồi"), trong đó các dịch vụ phải đợi phản hồi trước khi thực hiện tác vụ tiếp theo. Kiến trúc hướng sự kiện điều khiển luồng làm việc bằng cách sử dụng các sự kiện và được thiết kế để phản hồi hoặc thực hiện các hành động phản hồi sự kiện. Trong một hệ thống hướng sự kiện, các thành phần của nó hợp tác bằng cách gửi và nhận các sự kiện. Người gửi gửi một sự kiện đến một trình điều phối. Trình điều phối sự kiện có trách nhiệm phân phối sự kiện đến tất cả các thành phần đã tuyên bố quan tâm để nhận nó. Do đó, trình điều phối sự kiện cho phép tách rời giữa các nguồn và người nhận sự kiện. Một hệ thống hướng sự kiện thông thường có ba phong cách xử lý sự kiện chung: đơn giản, luồng và phức tạp. Nó có thể bao gồm một số phần chính như xử lý sự kiện, công cụ sự kiện, nguồn và đích, siêu dữ liệu sự kiện.

Nội dung của phần này sẽ được đề cập lại trong chương 4 của luận án này.

2.5. Học tăng cường

Học tăng cường là một lĩnh vực liên quan đến việc dạy một tác nhân (agent) thực hiện một nhiệm vụ bằng cách tương tác với môi trường thông qua hành động và nhận phần thưởng. Trong luận án này, tác nhân là một trình điều khiển kiểm thử. Đây là một chương trình có khả năng thực thi nhằm mục đích chạy hàm đang cần kiểm thử với một dữ liệu kiểm thử cho trước. Môi trường mà tác nhân tương tác là ứng dụng đang được kiểm thử (Application Under Test-AUT).

Luận án sử dụng phương pháp học tăng cường để tạo đường kiểm thử. Các bước cụ thể của quá trình sinh đường kiểm thử sẽ được trình bày trong các phần sau của luận án.

Chương 3

PHÂN TÍCH NHẬT KÝ TRUY CẬP THEO THỜI GIAN THỰC ĐỂ PHÒNG NGỪA TẤN CÔNG DDOS

Trong chương này, luận án đề xuất phương pháp phân tích nhật ký truy cập theo thời gian thực của hệ thống web nhằm phát hiện các dấu hiệu bất thường có thể dẫn đến tấn công DDoS. Cụ thể là phát hiện và cảnh báo các địa chỉ IP có số lượng truy cập cao vượt ngưỡng và những IP có lượng truy cập cao bất thường vào hệ thống. Thêm vào đó, phương pháp được đề xuất cũng đã được tiến hành thực nghiệm với hệ thống web *Shopbase* bằng công nghệ *ApacheSpark* và *Kubernetes* và luận án cũng giới thiệu tính hiệu quả tích cực của phương pháp trong thực tế.

3.1. Giới thiệu

DoS *(Denial-of-Service)* và DDoS *(Distributed Denial-of-Service)* là những kiểu tấn công mạng phổ biến nhằm mục đích khiến máy chủ tràn ngập nhiều gói giao thức điều khiển truyền dẫn/giao thức gói dữ liệu người dùng (TCP/UDP) hơn mức nó có thể xử lý. Điều này có thể dẫn đến hỏng dữ liệu và tài nguyên có thể bị định hướng sai hoặc thậm chí cạn kiệt tài nguyên đến mức làm tê liệt hệ thống. Sự khác biệt chính giữa các cuộc tấn công DoS và DDoS là cuộc tấn công DDoS sử dụng nhiều kết nối internet để đặt mạng máy tính của nạn nhân ngoại tuyến trong khi cuộc tấn công DoS sử dụng một kết nối duy nhất. Hiện nay, việc ngăn chặn và cảnh báo các cuộc tấn công DoS, DDoS đặt ra rất nhiều thách thức trong lĩnh vực bảo mật phần mềm.

Đã có nhiều nghiên cứu về vấn đề tấn công DDoS vào các ứng dụng web. Các kỹ thuật thường được sử dụng trong các cuộc tấn công từ chối dịch vụ là học máy (dựa trên phân loại hoặc học sâu) hoặc trí tuệ nhân tạo (AI). Đối với bài toán dự đoán tấn công mạng, một số kỹ thuật học máy đã được nghiên cứu và ứng dụng vào thực tế, trong đó có nhiều kỹ thuật được sử dụng với cây quyết định. Hiện tại chưa có phương pháp nào có thể ngăn chặn hoàn toàn tấn công

DoS/DDoS. Vì vậy, các vấn đề nghiên cứu trong an ninh mạng thường quan tâm đến việc phát hiện để ngăn chặn các cuộc tấn công DDoS, từ đó giảm nguy cơ trở thành nạn nhân và giảm thiểu tác động của một cuộc tấn công DDoS. Một số phương pháp được đề cập trong nghiên cứu là phân tích luồng thông tin để phát hiện các dấu hiệu tấn công, giám sát lưu lượng để phát hiện sớm các cuộc tấn công, tạo danh sách quản lý truy cập để chặn địa chỉ IP của kẻ tấn công, nâng cao năng lực xử lý của hệ thống để tiếp nhận thêm lưu lượng truy cập...

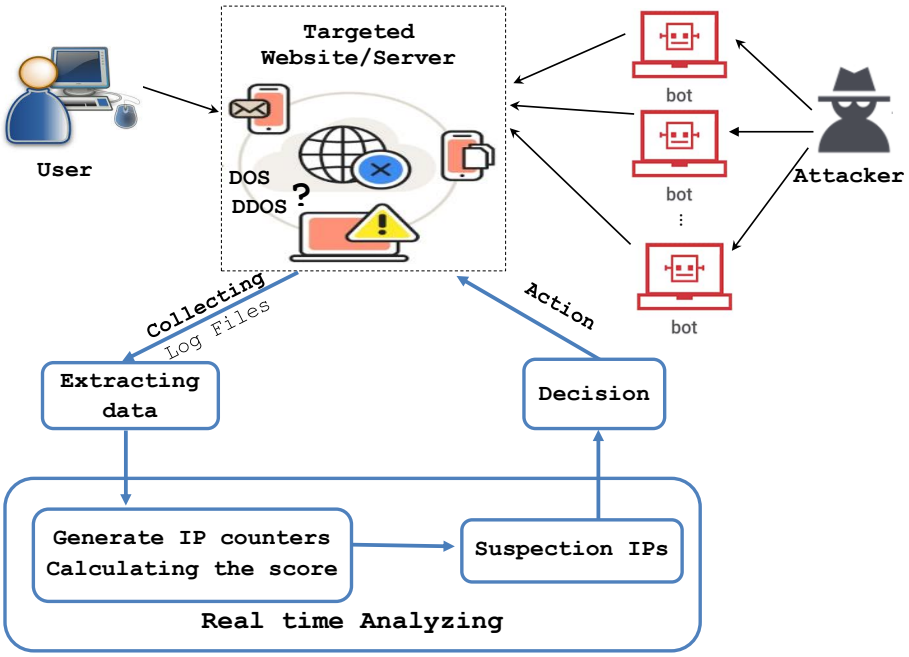
Với các vấn đề đã nêu ở trên, chương này của luận án sẽ đề xuất phương pháp phân tích nhật ký truy cập của hệ thống web theo thời gian thực để phát hiện sớm những điểm bất thường của người dùng có thể ảnh hưởng xấu đến hệ thống trong khi vận hành làm dẫn đến các cuộc tấn công DDoS. Những đóng góp của chương này bao gồm:

- *phát hiện và cảnh báo các địa chỉ IP có số lượng yêu cầu gửi đến máy chủ vượt ngưỡng cho phép trong các cuộc tấn công DoS;*
- *tạo công thức chấm điểm cho các địa chỉ IP và tìm ra những IP có nguy cao tạo ra các cuộc tấn công DDoS dựa trên một số tiêu chí;*
- *xây dựng phần mềm thử nghiệm để minh họa tính khả thi và hiệu quả của phương pháp đề xuất.*

3.2. Phương pháp phân tích nhật ký truy cập theo thời gian thực

Nhật ký truy cập của người dùng vào các ứng dụng web đóng một vai trò quan trọng trong việc phát hiện các cuộc tấn công mạng. Bởi vì, khi tham gia các hoạt động trên internet, một số thông tin của người dùng sẽ được đính kèm theo các yêu cầu của người dùng đến hệ thống máy chủ cung cấp dịch vụ. Khi đó, các thông tin này sẽ được ghi vào bản ghi dữ liệu nhật ký truy cập của hệ thống máy chủ mà người dùng yêu cầu.

Phương pháp đề xuất của chương này nhằm mục đích phát hiện và cảnh báo các nguy cơ tấn công mạng theo thời gian thực như trong Hình. 3.1, đặc biệt là đối với hai bài toán DoS và DDoS. Tổng quan về quy trình giải quyết vấn đề bao gồm ba giai đoạn cơ bản:*(1) thu thập nhật ký truy cập của người dùng; (2) phân tích nhật ký; và (3) thông báo cho người quản trị (admin).*



Hình 3.1: Tổng quan tiến trình xử lý giải quyết bài toán.

3.2.1. Bài toán DoS

Liên quan đến bài toán DoS, mục tiêu của phương pháp là phát hiện, cảnh báo và ngừng cung cấp dịch vụ cho những địa chỉ IP có số lượng yêu cầu vào hệ thống cao vượt ngưỡng quy định. Ý tưởng chính để giải quyết vấn đề này là tạo bộ đếm IP truy cập theo khoảng thời gian đặt trước (cứ sau hai giây), sau đó tổng hợp và lọc ra các IP vượt quá ngưỡng và đưa chúng vào BlackList) để tạm dừng dịch vụ các IP này trong một khoảng thời gian nhất định (trong ba giây). Sau thời gian đó các IP này sẽ hoạt động trở lại bình thường. Việc này nhằm giảm tạm thời lưu lượng truy cập vào hệ thống để hệ thống đáp ứng được yêu cầu của các IP thông thường khác giúp hệ thống hoạt động tốt hơn. Các tác vụ cần thực hiện và quy trình giải quyết vấn đề DoS được mô tả như trong Thuật toán 3.1.

3.2.2. Bài toán DDoS

Do kẻ tấn công có thể phân phối số lượng yêu cầu tới hệ thống từ nhiều thiết bị trong một cuộc tấn công nên DDoS cho nên để phát hiện tấn công đòi hỏi cơ chế xử lý phức tạp hơn nhiều so với vấn đề DoS. Do đó, thay vì đếm số lượng yêu cầu từ một địa chỉ IP, phương pháp đề xuất sẽ xem xét địa chỉ IP trên nhiều khía cạnh khác nhau (*e.g.*, thời gian phản hồi, nội dung lớn,...) làm cơ sở để đưa

Thuật toán 3.1 Giải quyết bài toán DoS.

Input : Luồng dữ liệu về nhật ký truy cập của người dùng.
Output: Chỉ ra các địa chỉ IP có số lượng yêu cầu cao bất thường trong khoảng thời gian đặt trước.
Data : *threshold* - giá trị của ngưỡng để vượt qua IP và BlackList.

```
1 Procedure solveDosProblem(s)
2 begin
3     for each IP do
4         aggregateRequests(IP); //Tổng hợp tổng số yêu cầu theo từng IP trong
           khoảng thời gian đặt trước.
5         if aggregateRequests(IP) > threshold then
6             BlackList ← IP;
7             indicated(IP);
```

ra cảnh báo tấn công DDoS. Quá trình giải quyết ngăn chặn tấn công DDoS được mô tả trong Thuật toán 3.2 theo các giai đoạn sau:

1. Trích xuất thông tin quan trọng cho từng IP (tiêu chí được xem xét);
2. Tính điểm cho từng tiêu chí và suy ra điểm của từng địa chỉ IP;
3. So sánh điểm của từng IP với ngưỡng và đưa ra quyết định tương ứng.

Thuật toán 3.2 Giải quyết bài toán DDoS.

Input : Luồng dữ liệu về nhật ký truy cập của người dùng.
Output: Thông báo tới người dùng
Data : *thresholds - weights*

```
1 Procedure solveDDosProblem(s)
2 begin
3     for each IP do
4         scoreCalculate(IP); // Tính điểm cho từng tiêu chí của địa chỉ IP.
          decision(IP);
```

3.3. Tóm tắt chương

Trong bối cảnh tấn công mạng đang đặt ra những thách thức rất lớn đối với an ninh phần mềm, trong chương này, luận án đã đề xuất phương pháp phát

hiện và cảnh báo tấn công mạng, cụ thể là các bài toán DOS và DDoS. Về phương diện lý thuyết, phương pháp đề xuất này đã trình bày về giải pháp tìm kiếm các địa chỉ IP bất thường có khả năng cao dẫn tới tấn công từ chối dịch vụ. Về mặt thực nghiệm, phương pháp đề xuất đã triển khai và xây dựng hệ thống *Shopbase* áp dụng các công nghệ *Apache Spark* and *Kubernetes*. Đóng góp chính của phương pháp chính là kết quả của quá trình phân tích được thực hiện trong thời gian thực. Kết quả thực nghiệm cho thấy, các ứng dụng web được triển khai thực tế trên một hệ thống web lớn với hàng triệu người dùng vẫn đang mang lại hiệu quả và đã thực sự vận hành ổn định với các mục tiêu phi chức năng của nó.

Chương 4

KIỂM CHỨNG TIẾN TRÌNH CỦA CÁC SỰ KIỆN TẠI THỜI ĐIỂM THỰC THI

Chương này giới thiệu một phương pháp để kiểm chứng tiến trình các sự kiện trong một kiến trúc hướng sự kiện (EDA) tại thời điểm thực thi.

4.1. Giới thiệu

Kiểm chứng thời gian thực thi là phương pháp phân tích và thực thi hệ thống máy tính bằng cách trích xuất thông tin từ một hệ thống đang chạy và sử dụng thông tin đó để phát hiện và phản ứng với các hành vi phù hợp hoặc vi phạm các thuộc tính nhất định. Các thuộc tính nhất định như datarace và deadlock freedom thường được mong muốn bởi tất cả các hệ thống và có thể được thực hiện tốt nhất theo thuật toán. Các thuộc tính khác có thể được biểu diễn dưới dạng thông số kỹ thuật hình thức để dễ dàng nắm bắt.

4.2. Phương pháp đề xuất

Luận án này đề xuất một phương pháp để kiểm tra việc thực thi các sự kiện trong hệ thống để đảm bảo rằng chúng đáp ứng đủ các yêu cầu được quy định. Trong một hệ thống dựa trên sự kiện, các sự kiện được sử dụng để đồng bộ hóa các thành phần khác nhau của hệ thống và đảm bảo rằng chúng hoạt động cùng nhau để đạt được mục tiêu chung. Do đó, đảm bảo rằng các sự kiện được thực thi đúng cách là vô cùng quan trọng.

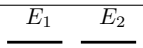
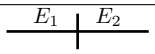
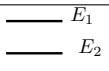
Bước đầu tiên của phương pháp là thu thập các tệp nhật ký về thời điểm bắt đầu và kết thúc của các sự kiện khi hệ thống thực thi. Các tệp nhật ký này được thu thập bởi Kafka, một hệ thống xử lý dữ liệu phân tán. Các tệp nhật ký này cung cấp thông tin chi tiết về thời gian bắt đầu và kết thúc của các sự kiện trong hệ thống. Bảng 4.2 mô phỏng dữ liệu trong các tệp này.

Sau đó, phương pháp sử dụng các thuật toán để kiểm chứng tiến trình thực thi của các sự kiện trong EDA có đáp ứng được các đặc tả hay không. Các thuật

toán này sẽ kiểm chứng các mối quan hệ giữa các sự kiện và đảm bảo rằng chúng đáp ứng đủ các đặc tả được quy định. Chũng cũng phân tích các tệp nhật ký và kiểm tra xem các sự kiện có thực thi đúng theo thứ tự và trong khoảng thời gian được quy định hay không. Các thuật toán này được trình bày chi tiết ở phần sau của luận án.

Cuối cùng, sử dụng một công cụ hỗ trợ để thực hiện việc kiểm chứng các hệ thống dựa trên sự kiện một cách tự động. Như đã đề vập ở trên, phương pháp của chúng tôi chỉ kiểm chứng được một số đặc tả như trong Bảng 4.1.

Bảng 4.1: Allen's thirteen atomic interval temporal relations to represent the temporal relations between two events X and Y

Relations	Symbol	Inverse	Pictorial Meaning
E_1 before E_2	b	bi	
E_1 meet E_2	m	mi	
E_1 overlaps E_2	o	oi	
E_1 starts E_2	s	si	
E_1 during E_2	d	di	
E_1 finishes E_2	f	fi	
E_1 equal E_2	eq	eq	

Giả sử rằng, các sự kiện trong hệ thống có thể được ghi lại dưới dạng Bảng 4.2, trong đó chúng ta chỉ nhận được tên sự kiện, thời gian bắt đầu và thời gian kết thúc của các sự kiện.

Bảng 4.2: Simulation data of events in the system

Event name	Start Time	End Time
E_1	t1	t2
E_2	t1	t2
E_3	t1	t2
E_4	t1	t2
E_3	t1	t2
E_2	...	...

4.3. Các thuật toán kiểm chứng

4.3.1. Sự kiện $E1$ xảy ra trước sự kiện $E2$ ($E1$ before $E2$)

Nếu thời điểm đang kiểm chứng là thời điểm mà $E1$ đang diễn ra thì chúng ta sẽ không thể kiểm chứng được mối quan hệ này vì chưa thể có dữ liệu về tương lai. Do đó không thể xác định được thời điểm nào sự kiện $E1$ sẽ kết thúc và sự kiện $E2$ xảy ra lúc nào. Nhưng khi kiểm chứng trong khi $E2$ đã xảy ra thì hoàn toàn có thể được bởi vì lúc này đã biết được sự kiên $E1$ đã xảy ra hay chưa và đã kết thúc từ lúc nào.

Thuật toán 4.1 $E1$ before $E2$.

Input : Mảng dữ liệu thời gian A, hai sự kiện $E1$ và $E2$.

Output: Giá trị Boolean để kiểm tra ràng buộc có thỏa mãn hay không

```
1  Initialize the R result variable to true;
2  for i = length(A) − 1 to i = 0 do
3      if event E2 has not finished then
4          Ignore the current occurrence data of event E2;
5          else
6              for each time event E1 occurs (A[e1]), find the previous occurrence of
                event E1 and check if event E2 occurred during this interval;
7              if the occurence of event E2 is found (A[e2]) then
8                  if the end time of A[e2] ≥ the start time of A[e1] then
9                      R = false;
10                     Return R;
11                     else
12                         R = false;
13                         Return R;
14 if there exist only one of the event data E1 or E2 then
15     R = false;
16 Return R
```

4.3.2. Sự kiện $E1$ xảy ra sau sự kiện $E2$ ($E1$ after $E2$)

Ngược lại với ràng buộc trên, ràng buộc này có thể thực hiện kiểm chứng được nếu chúng ta xét tại một thời điểm mà $E1$ đang diễn ra. Lúc này đã có thể biết được sự xuất hiện của sự kiện $E2$ hay chưa.

Thuật toán 4.2 $E1$ after $E2$.

Input : Mảng dữ liệu thời gian A, hai sự kiện $E1$ và $E2$.

Output: Giá trị Boolean để kiểm tra ràng buộc có thỏa mãn hay không

```
1  Initialize the R result variable to true;
2  for i = 0 to length(A) − 1 do
3      if event E1 has not finished then
4          Ignore the current occurrence data of event E1;
5          else
6              for each time event E1 occurs (A[e1]), find if events E2 occurs afterwards
                before event E1 occurs next time;
7              if the occurence of event E2 is found (A[e2]) then
8                  if the end time of A[e1] ≥ the start time of A[e2] then
9                      R = false;
10                     Return R;
11                 else
12                     R = false;
13                     Return R;
14 if there exist only one of the event data E1 or E2 then
15     R = false;
16 Return R
```

Chương 5

KIỂM THỬ FUZZ CÁC ỨNG DỤNG WEB

5.1. Giới thiệu

Tính bảo mật của các hệ thống web luôn là mối quan tâm lớn đối với các nhà phát triển. Ở giai đoạn triển khai, để ngăn chặn các cuộc tấn công XSS, các lập trình viên có thể sử dụng các tính năng bảo mật trong các framework. Tuy nhiên, trên thực tế việc lập trình luôn tiềm ẩn những sai sót. Do đó, việc cung cấp các giải pháp đảm bảo chất lượng cho các ứng dụng dựa trên web ngày càng trở nên quan trọng.

5.2. Phương pháp kiểm thử tự động

Trong phần này, trước tiên luận án giới thiệu tổng quan về phương pháp để thực hiện kiểm thử tự động ứng dụng web *jFAT*. Hình **??** mô tả kiến trúc của khung làm việc đề xuất. Phương pháp kiểm thử đề xuất được thực hiện dựa trên Q-Learning, mô-đun và POM (Page Object Model) cho phép sinh các đường dẫn kiểm thử, ca kiểm thử, phát triển và duy trì các trường hợp kiểm thử dễ dàng hơn. Nó bao gồm 3 giai đoạn chính: *(1) Sinh các đường dẫn kiểm thử cho ứng dụng web*; *(2) Sinh các ca kiểm thử*; *(3) Tiến hành kiểm thử.*

5.2.1. Sinh đường dẫn kiểm thử

Trong mục này, luận án sẽ trình bày về các bước được thực hiện trong giai đoạn đầu tiên để tự động tạo các đường dẫn kiểm thử có khả năng chứa các tấn công XSS trong các ứng dụng web.

1. Sinh đồ thị của ứng dụng web từ địa chỉ của trang web (URL).
2. Từ đồ thị $\mathcal{G}$ và ma trận phần thưởng $\mathcal{R}$, chúng ta thêm trọng số vào $\mathcal{G}$ để xây dựng đồ thị trọng số $\mathcal{G}$'.
3. Xây dựng ma trận $\mathcal{Q}$ để biểu diễn kết quả về những gì tác nhân đã học được thông qua kinh nghiệm và được xây dựng từ đồ thị có trọng số $\mathcal{G}$'và ma trận $\mathcal{R}$.

4. Tạo đường thử nghiệm để kiểm thử thâm nhập.

5.2.1.1. Tạo đồ thị

Trong đề xuất này, ứng dụng web được kiểm thử là môi trường và công cụ thử nghiệm thâm nhập là tác nhân. Công cụ kiểm thử thâm nhập áp dụng các tương tác với ứng dụng để tìm chính sách tạo điều kiện phát hiện các lỗ hổng hệ thống. Quá trình khám phá này tạo ra các chuỗi sự kiện có thể dùng làm cơ sở thử nghiệm.

Dựa trên ý tưởng tìm kiếm theo chiều sâu, phương pháp đề xuất này phân tích một ứng dụng web và biểu diễn dưới dạng biểu đồ trạng thái $\mathcal{G}$ như sau.

Định nghĩa 5.1 *(Đồ thị chuyển của ứng dụng web) Đồ thị $\mathcal{G} = \langle \mathbb{S}, \mathbb{V} \rangle$ là đồ thị có hướng. $\mathbb{S} = \{s_0, s_1, ..., s_n\}$, mỗi nút s_i trong biểu đồ là một trang con của ứng dụng web đang được thử nghiệm. Cạnh từ nút s_i đến nút s_j nếu trang s_i có liên kết đến trang hoặc được chuyển hướng đến trang s_j.*

5.2.1.2. Thiết lập trọng số cho đồ thị

Ở bước này, luận án thiết lập trọng số cho đồ thị $\mathcal{G}$ và xây dựng ma trận phần thưởng R. Trên một trang web, các thành phần có thể bị tổn thương bao gồm các trường đăng nhập, trường tìm kiếm, trường nhận xét và bất kỳ mục nhập dữ liệu nào khác, v.v. Trong phương pháp này, luận án đặt các thành phần này vào một tập hợp VS được gọi là tập các phần dễ bị tấn công XSS. Trong các nghiên cứu trước đây, giá trị của mỗi phần tử trong ma trận R thường là một số nguyên (>0, 0, hoặc -1). Tuy nhiên, khi tạo ma trận Q ở giai đoạn tiếp theo thì bài toán truy vết rất phức tạp. Một trong các đóng góp của phương pháp này chính là sự cải tiến trong việc xây dựng R. Cụ thể là, mỗi phần tử trong ma trận R sẽ được bổ sung thêm thông tin của trạng thái tiếp theo. Điều này làm giảm thời gian truy vết khi tìm đường dẫn và tối ưu hóa thời gian trong quá trình tạo đường dẫn kiểm thử.

Thuật toán 5.2 chịu trách nhiệm thiết lập các trọng số cho đồ thị $\mathcal{G}$ và xây dựng ma trận phần thưởng R.

5.2.1.3. Xây dựng ma trận Q

Thuật toán Q-Learning có thể tìm ra cách tối ưu để đạt được một trạng thái nhất định của môi trường. Do đó, thuật toán này có thể được sử dụng để hướng

Thuật toán 5.1 Phân tích cấu trúc ứng dụng web.

Input : s_0 - URL of the webapp.
d - limited depth.
Output: $\mathcal{G} =< \mathcal{S}, \mathcal{V} >$ - state graph.
Data : s' - the next state of s after performing a.
T - the set of termination states.

```
1  Function createStateGraph(s_0, d)
2  begin
3      URLs ← {s_0};
4      S ← ∅;
5      depth(s_0) = 0;
6      while true do
7          if URLs = ∅ then
8              exit;
9          s = getFirst(URLs);
10         getA_s(s);
11         S ← S ∪ {s};
12         if getAs(s) = ∅ then
13             T ← T ∪ {s}; exit;
14         if depth(s) ⩽ d then
15             get s's sub urls s';
16             URLs ← URLs ∪ {s'};
17             depth(s') ← depth(s) + 1;
18             V ← V ∪ {(s, s')};
19 return G;
```

dẫn thăm dò các trạng thái tồn tại các lỗ hổng bảo mật.

Thuật toán 5.3 trình bày các bước để xây dựng ma trận phần thưởng tích lũy Q từ $\mathcal{G}'$ và ma trận phần thưởng R. Làm thế nào để tác nhân biết nên chọn hành động nào để đạt được phần thưởng lớn nhất? Điều này được thực hiện bằng cách sử dụng một giá trị gọi là hàm Q-value. Hàm Q-value cho phép tác nhân tạo thử nghiệm "nhìn trước" khi đưa ra lựa chọn về sự kiện sẽ chọn trong một trạng thái cụ thể.

Thuật toán 5.2 Đặt trọng số cho ma trận $\mathcal{G}$ và xây dựng ma trận R.

Input : $\mathcal{G} =< \mathcal{S}, \mathcal{V} >$ - state graph of webapp.
Output: $\mathcal{G}$'- weight state graph.
R - reward matrix.
Data : s' - the next state of s after performing a.
VS - the vulnerable parts of the XSS attack.

```
1  Procedure setWeightForGraph(𝒢)
2  begin
3      initation(VS);
4      foreach node s ∈ 𝒮 do
5          foreach a ∈ A(s) do
6              if a ∈ VS then
7                  R(s, a) = (1, s');
8                  weight(s, s') = 1;
9              else
10                 R(s, a) = (0, s');
11                 weight(s, s') = 0;
12         for all a ∉ A(s) do
13             R(s, a) = (−1, ∅);
14 return 𝒢', R;
```

5.2.1.4. Tạo đường dẫn kiểm thử

Trong giai đoạn này, dựa trên ma trận Q có sẵn, phương pháp đề xuất sẽ tìm thấy các đường dẫn hữu ích có khả năng chứa lỗ hổng XSS. Bắt đầu từ trạng thái ban đầu, dựa trên ma trận Q, thuật toán tìm các hành động có giá trị phần thưởng cao nhất được ghi trong ma trận Q cho trạng thái hiện tại. Chuỗi các (trạng thái, hành động) được tìm thấy là đường dẫn thử nghiệm cần thiết để kiểm tra các lỗ hổng XSS.

5.2.2. Sinh ca kiểm thử

Từ kết quả các đường dẫn kiểm thử thu được của giai đoạn trên, trong giai đoạn tiếp theo này, luận án sẽ tiếp tực thực hiện công việc sinh ca kiểm thử để thực hiện tiến trình trong khung kiểm thử đề xuất ở Hình **??**. Quá trình sinh ca kiểm thử được chỉ ra như Thuật toán 5.5.

Thuật toán 5.3 Xây dựng ma trận phần thưởng Q.

Input : $\mathcal{G}$'- weight state graph.
R - reward matrix.
Output: Q - the cumulative reward matrix.
Data : *episode* - a path from the initial state to the end state.
T - the set of termination states.
γ - the discount factor.
s, s_0 - the sates.

```
1 Function builtQmatrix(G', R) begin
2     Set the γ parameter; Initialize matrix Q to zero;
3     foreach episode do
4         Select s0 as initial state;
5         while s ∉ T do
6             Select action a ∈ A(s);
7             Q(s, a) = R(s, a) + γ * Max[Q(nextstate, allactions)];
8             s = next state;
9 return Q;
```

5.2.3. Tiến hành kiểm thử

Sau khi xây dựng được tập các ca kiểm thử, phương pháp đề xuất sẽ tiến hành kiểm thử tự động theo các bước: *(i) Tạo POM từ các ứng dụng web*; *(ii) Xây dựng kịch bản test*; *(iii) Thực thi kịch bản kiểm thử*; và *(iv) Ghi nhật ký kết quả kiểm tra.*

5.3. Tóm tắt chương

Luận án giới thiệu phương pháp sử dụng Q-learning để tạo các đường dẫn kiểm thử phát hiện lỗ hổng bảo mật XSS của webapp. Trong quá trình thử nghiệm, công cụ đã tiến hành một số thử nghiệm với nhiều hệ thống website và kết quả cho thấy phương pháp này có ưu thế về mặt thời gian do ma trận phần thưởng được cải thiện.

Các Kết quả nghiên cứu của chương này đã được công bố tại Hội nghị *NAFOSTED Conference on Information and Computer Science (NICS 2022)* và Hội nghị *NAFOSTED Conference on Information and Computer Science (NICS 2022).*

Thuật toán 5.4 Sinh đường kiểm thử

Input : Q - the cumulative reward matrix.
Output: TPs - test paths.
Data : T - the set of termination states.

```
1  Function testPath(Q)
   begin
2      TP = ∅;
3      TPs = ∅;
4      int(curentState) = s_0;
5      while true do
6          selectedEvent=maxValue(Q(curentState, allEvent));
7          TP ← TP ∪ curentState;
8          if currentState ∈ T then
9              break;
10         else
11             curentState ← R[curentState, selectedEvent].nextState;
12     TPs ← TPs ∪ TP;
13 return TPs;
```

Thuật toán 5.5 Sinh ca kiểm thử.

Input : TPs - các đường kiểm thử.
Output: TCs - các ca kiểm thử.
Data : S - Tập các nút.

```
1  Function testCases(TPs)
   begin
2      TCs = ∅;
3      S = ∅;
4      foreach TP ∈ TPs do
5          S ← S ∪ getNode(TP);
6      foreach s ∈ S do
7          TCs ← TCs ∪ generateTC(s);
8  return TCs;
```

Chương 6

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Sau một thời gian nghiên cứu và giải quyết bài toán, luận án đã có một số kết quả như sau:

(i) *Đề xuất phương pháp phân tích nhật ký truy cập của một hệ thống web nhằm phát hiện các dấu hiệu bất thường của người dùng có thể dẫn đến tấn công DDoS.* Nghiên cứu này đề xuất một phương pháp với các đóng góp sau: *i) phát hiện và cảnh báo các địa chỉ IP có số lượng yêu cầu gửi đến máy chủ vượt ngưỡng cho phép trong các cuộc tấn công DoS; ii) tạo công thức chấm điểm cho các địa chỉ IP và tìm ra những IP có nguy cao tạo ra các cuộc tấn công DDoS dựa trên một số tiêu chí; iii) xây dựng phần mềm thử nghiệm để minh họa tính khả thi và hiệu quả của phương pháp đề xuất.*. Nghiên cứu này góp phần cải thiện, nâng cáo tính sẵn sàng của phần mềm nền web.

(ii) *Đề xuất một phương pháp kiểm chứng tiến trình thực thi của các sự kiện trong kiến trúc hướng sự kiện (EDA) trong thời gian thực thi.*

(iii) *Kiểm thử fuzz ứng dụng web.* Đề xuất một khung kiểm thử tự động cho các ứng dụng web bằng Java, tích hợp với Selenium và TestNG, cung cấp các tính năng để thực hiện kiểm thử tự động. Đồng thời, đề xuất một phương pháp sử dụng Q-learning để tự động tạo ra các đường dẫn kiểm thử cho kiểm thử thâm nhập trên các ứng dụng web. Các đường dẫn kiểm thử được tạo ra nhanh chóng bằng cách cải thiện ma trận phần thưởng để nhanh chóng theo dõi trạng thái tiếp theo, tiết kiệm thời gian tạo đường dẫn thử nghiệm. Ngoài ra, một công cụ cũng được phát triển để tạo ra các đường dẫn thử nghiệm có khả năng chứa các cuộc tấn công XSS trên các ứng dụng web và công cụ để thực hiện kiểm thử tự động theo khung làm việc đã đề xuất. Nghiên cứu này góp phần cải thiện, nâng cao tính bảo mật, toàn vẹn của phần mềm nền web.

6.2. Hướng phát triển

Luận án này đã đề xuất một số phương pháp để nhằm phát hiện các hoạt động bất thường của người dùng, mà có thể là dấu hiệu của các cuộc tấn công, hay cảnh báo các lỗ hổng trong hệ thống phần mềm trên nền web. Đồng thời luận án đã đề xuất phương pháp phân tích Event log của các hệ thống EDA để để kiểm chứng tiến trình thực thi của các sự kiện trong một kiến trúc hướng sự kiện (EDA) tại thời điểm thực thi. Tuy nhiên, một số thách thức vẫn tồn tại trong việc triển khai các phương pháp này trong thực tế. Ví dụ, việc phát hiện hoạt động bất thường của người dùng phụ thuộc vào việc phân tích và đánh giá dữ liệu nhật ký truy cập, vì vậy độ chính xác của phương pháp sẽ bị ảnh hưởng bởi chất lượng và độ tin cậy của dữ liệu đó. Ngoài ra, việc triển khai các phương pháp này trong các hệ thống lớn và phức tạp có thể gặp khó khăn do các thách thức về hiệu suất và quản lý dữ liệu. Do đó, việc tiếp tục nghiên cứu và phát triển các phương pháp mới để giải quyết những thách thức này sẽ là cần thiết để đảm bảo an toàn và bảo mật cho các hệ thống phần mềm trên nền web.

(i) Nâng cao hiệu suất của phương pháp phân tích dữ liệu: Phương pháp hiện tại có thể được cải tiến để tăng tốc độ phân tích dữ liệu và đưa ra cảnh báo nhanh hơn để giảm thiểu thiệt hại của các cuộc tấn công.

(ii) Xây dựng một hệ thống giám sát tự động: Các nghiên cứu tiếp theo có thể tập trung vào xây dựng một hệ thống giám sát tự động có khả năng tự động phát hiện các hành vi bất thường và đưa ra các hành động phù hợp để bảo vệ hệ thống.

(iii) Áp dụng phương pháp vào các môi trường thực tế: Nghiên cứu này cần được áp dụng để giải quyết các vấn đề thực tế trong các hệ thống web. Các nghiên cứu tiếp theo có thể tập trung vào phát triển các giải pháp cho các vấn đề như tăng cường tính khả dụng của hệ thống, giảm thiểu số lượng cảnh báo sai và cải thiện khả năng thích ứng của hệ thống.