Chương 1

GIỚI THIỆU

1.1. Đặt vấn đề

Phầm mềm đang được sử dụng phổ biến trong nhiều lĩnh vực như kinh tế, giáo dục, quân sự, y tế, v.v. Bên cạnh những lợi ích mà phần mềm mang lại thì các vi phạm an ninh cũng xuất hiện ngày càng đa dạng, phức tạp. Các hệ thống phần mềm, đặc biệt là các ứng dụng web có chứa các thông tin quan trọng luôn tiềm ẩn nhiều nguy cơ tấn công an ninh. Có nhiều khía cạnh cần quan tâm đối với một hệ thống phần mềm. Với những phần mềm trọng yếu về an ninh thì việc đảm bảo tính chất an ninh là vấn đề sống còn của hệ thống.

Trong thực tế, mỗi hệ thống phần mềm luôn cần một chính sách an ninh phù hợp để đảm bảo các tính chất an ninh theo các yêu cầu của khách hàng. Tuy nhiên, việc triển khai chính sách an ninh ở các giai đoạn phát triển phần mềm luôn có khả năng chứa các lỗi an ninh tiềm ẩn, đặc biệt là ở giai đoạn lập trình. Nguyên nhân gây lỗi có thể do có sự xung đột khi kết hợp các thành phần hệ thống, lạm dụng việc sử dụng các thư viện bên thứ ba hay người lập trình không tuân thủ chính xác các yêu cầu của hệ thống, v.v. Nếu những lỗi này được phát hiện càng muộn thì chi phí sửa chữa hệ thống, khắc phục hậu quả có thể càng lớn và phức tạp. Chính vì vậy, việc kiểm tra để đảm bảo chính sách an ninh của hệ thống được cài đặt chính xác ở từng giai đoạn phát triển phần mềm giúp phát hiện sớm các vi phạm an ninh, giảm chi phí sửa chữa, đảm bảo tính an ninh và gia tăng chất lượng của phần mềm.

Vì vậy, luận án *"Phương pháp kiểm chứng các tính chất an ninh của phần mềm"* tập trung đề xuất các phương pháp phân tích, biểu diễn và các thuật toán để kiểm tra chính sách truy cập của các ứng dụng web được triển khai theo phương pháp an ninh lập trình, an ninh khai báo.

1.2. Các kết quả chính của luận án

Các kết quả nghiên cứu của luận án góp phần bổ sung và hoàn thiện các phương pháp phân tích, biểu diễn và kiểm tra các chính sách truy cập của các ứng dụng web. Cụ thể, luận án có các đóng góp chính sau đây:

(i) *Đề xuất phương pháp kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình.* Các phương thức khai thác tài nguyên trong ứng dụng web được rút trích thành danh sách quyền truy cập tài nguyên. Sau đó kết hợp với thành phần *Controller* và *View* để xây dựng đồ thị khai thác tài nguyên và ma trận kiểm soát truy cập

theo vai trò. Hai thuật toán được đề xuất để chuyển đổi đồ thị khai thác tài nguyên thành ma trận kiểm soát truy cập theo vai trò và phát hiện những quy tắc truy cập được triển khai không chính xác trong ứng dụng web. Phương pháp đề xuất được xây dựng thành công cụ kiểm chứng và thực nghiệm với hệ thống quản lý hồ sơ y tế.

(ii) *Đề xuất phương pháp kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo.* Chính sách truy cập của ứng dụng web được kiểm tra thông qua việc truy vấn cơ sở dữ liệu và phân tích các tệp cấu hình triển khai chính sách truy cập trong ứng dụng. Một cây phân tích truy cập theo vai trò được đề xuất để biểu diễn các quy tắc cấp quyền theo vai trò và các ràng buộc trong ứng dụng web. Hai thuật toán được đề xuất để phát hiện các phép gán đã triển khai không phù hợp với đặc tả. Phương pháp đề xuất đang được triển khai thành công cụ kiểm chứng để thực nghiệm với hệ thống quản lý hồ sơ y tế.

(iii) *Đề xuất phương pháp kiểm chứng chính sách kiểm soát truy cập theo thuộc tính (ABAC).* Các quy tắc truy cập của ứng dụng web được phân tích, tổng hợp và biểu diễn bằng SpEL. Sự phù hợp của chính sách ABAC được triển khai trong ứng web và đặc tả của nó được thực hiện thông qua việc đưa ra các định nghĩa hình thức và các thuật toán kiểm tra tính bảo mật, tính toàn vẹn và tính sẵn sàng. Một công cụ kiểm chứng đã được phát triển để hỗ trợ quá trình kiểm chứng tự động và thực nghiệm với hệ thống quản lý hồ sơ y tế theo phương pháp đã đề xuất.

1.3. Bố cục luận án

Luận án *"Phương pháp kiểm chứng các tính chất an ninh của phần mềm"* bao gồm 6 chương. Trong đó, Chương 1 *Giới thiệu* trình bày về lý do chọn đề tài, các kết quả đạt được của luận án. Chương 2 trình bày tóm tắt về các kiến thức cơ sở được sử dụng trong luận án. Chương 3 đề xuất phương pháp kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình. Phương pháp kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo được đề xuất trong Chương 4. Nội dung Chương 5 đề xuất phương pháp kiểm chứng chính sách kiểm soát truy cập theo thuộc tính. Cuối cùng, Chương 6 là kết luận và một số hướng nghiên cứu tiếp theo của luận án.

Chương 2

KIẾN THỨC CƠ SỞ

2.1. An ninh phần mềm

An ninh phần mềm đã trở thành một tiêu chí quan trọng trong việc đánh giá các hệ thống phần mềm ngày nay. Để hạn chế các vi phạm an ninh, các nhà phát triển thường phải xây dựng các chính sách an ninh cho các sản phẩm phần mềm của họ nhằm đáp ứng các yêu cầu an ninh của khách hàng. Khi đó, sự phù hợp của chính sách an ninh với các yêu cầu an ninh sẽ được thể hiện thông qua các tính chất an ninh của phần mềm. Theo nghiên cứu của Mead và các đồng tác giả, một số các tính chất an ninh cốt lõi của phần mềm gồm tính bảo mật (*confidentiality*), tính toàn vẹn (*integrity*), tính sẵn sàng (*availability*), tính trách nhiệm (*accountability*) và tính chống chối cãi (*non-repudiation*).

Việc đảm bảo an ninh phần mềm là một nhiệm vụ khó, khó hơn khi các nhà phát triển phải nhận thức và thực hiện đầy đủ các yêu cầu an ninh của tổ chức. Việc đảm bảo vấn đề an ninh phần mềm không chỉ gồm các chính sách an ninh của hệ thống, giải pháp về hạ tầng mà còn có cả nhận thức của người tham gia hệ thống. Với mục tiêu làm giảm số lượng các sai sót càng sớm càng tốt đồng thời giảm thiểu sự nhập nhằng và những điểm yếu khác. Tác giả McGraw đã giới thiệu một số hoạt động đảm bảo an ninh phần mềm như phân tích mã nguồn, phân tích rủi ro, v.v.

2.2. Một số phương pháp mô hình hóa chính sách an ninh của hệ thống

Kiểm soát truy cập cung cấp các cơ chế để kiểm soát và hạn chế các hành động hay phép toán được thực hiện bởi người dùng trên các tài nguyên trong hệ thống. *Kiểm soát truy cập dựa trên vai trò* (RBAC) là phương pháp cấp quyền cho người dùng theo vị trí chức vụ (vai trò) của người dùng trong tổ chức. Với bài toán cấp quyền không chỉ phụ thuộc vào vai trò của người dùng trong hệ thống mà việc cấp quyền cho người dùng còn phải được thỏa mãn một số ràng buộc khác như thời gian thực hiện hay đơn vị công tác của người dùng, lúc đó thì chính sách truy cập của hệ thống được biểu diễn bằng *ngôn ngữ mô hình hóa thống nhất an toàn* (SecureUML). Khi các hệ thống phần mềm lớn và phức tạp hơn, đặc biệt là có sự liên kết giữa nhiều tổ chức thì mô hình kiểm soát truy cập theo thuộc tính (ABAC) sẽ hiệu quả và linh hoạt.

2.3. Một số kiến trúc thiết kế phần mềm trong JavaEE

Model-View-Controller được gọi tắt là MVC là một mẫu thiết kế phần mềm được sử dụng phổ biến khi xây dựng, triển khai các ứng dụng web với nhiều ngôn ngữ khác nhau như Java, C#, Ruby, PHP, v.v. Những ứng dụng được thiết kế theo kiến trúc MVC nhằm tái sử dụng mã và phát triển song song một cách hiệu quả. Mục đích của mẫu thiết kế phần mềm này là đạt được sự phân tách giữa ba thành phần *Model*, *View* và *Controller* của một ứng dụng bất kỳ.

Spring Security là một khung làm việc an toàn hỗ trợ một số cơ chế kiểm soát truy cập để bảo đảm các chính sách truy cập cho các ứng dụng phần mềm doanh nghiệp dựa trên nền tảng JavaEE. Kiến trúc Spring Security bao gồm các hệ thống con có nhiệm vụ xác thực và ủy quyền. Xác thực có trách nhiệm thiết lập tính hợp lệ của thông tin khách hàng và ủy quyền sẽ quyết định quyền truy cập tài nguyên cho người dùng hợp lệ. Spring Security cho phép người lập trình cấu hình chính sách truy cập cả với Java và XML. Do đó, chính sách bảo mật có thể được thực thi theo các quy tắc kiểm soát truy cập dựa trên URL trong môi trường web hoặc ở mức độ của phương thức Java.

2.4. An ninh truy cập trong JavaEE

Các tài nguyên cần được bảo vệ của các hệ thống phần mềm thường được lưu trữ trong các hệ thống cơ sở dữ liệu. An ninh truy cập mã (*code access security*) là một giải pháp quan trọng trong các hệ thống cơ sở dữ liệu. Bởi vì, một số vấn đề có thể xảy ra khi người dùng ở máy khách đăng nhập hệ thống bằng cách sử dụng mã được ủy quyền là mã nguồn phía máy khách có thể không đáng tin cậy hoặc có lỗi, bao gồm mã độc. Với giải pháp an ninh truy cập mã, chính sách an ninh của các ứng dụng JavaEE có thể được triển khai theo hai phương pháp là *An ninh khai báo* và *An ninh lập trình*.

2.5. Phân tích chương trình

Trong quá trình triển khai chính sách an ninh của các hệ thống phần mềm. Việc chứng minh sự phù hợp giữa chính sách an ninh và đặc tả góp phần gia tăng tính tin cậy của hệ thống và đảm bảo chất lượng của phần mềm. Phân tích chương trình là phương pháp phổ biến được sử dụng để kiểm tra chính sách an ninh của các hệ thống phần mềm ở giai đoạn lập trình. Có hai kỹ thuật phân tích chương trình là phân tích tĩnh và phân tích động. Phân tích tĩnh đảm bảo bao quát đầy đủ các nhánh của chương trình, sự phụ thuộc chương trình, hoặc các tệp cấu hình được khai thác. Phân tích tĩnh cung cấp các phương pháp luận khác nhau, bao gồm kiểm chứng mô hình, chứng minh mô hình, để xác định các đường thực thi của một chương trình mà không phải thực hiện nó thực sự. Không giống như

rà soát thủ công, các bộ phân tích mã tĩnh có thể nắm bắt toàn diện và chính xác các mô hình của phần mềm, ví dụ như một biểu diễn trừu tượng của tất cả các đường thực thi.

2.6. Một số phương pháp biểu diễn chương trình

Sau quá trình phân tích mã nguồn, tùy theo mục đích và chức năng của quá trình kiểm chứng, các chương trình phần mềm sẽ được biểu diễn bằng các phương pháp khác nhau. Một số kỹ thuật chính được sử dụng để biểu diễn chương trình phần mềm là cây cú pháp trừu tượng (AST), đồ thị gọi và đồ thị luồng điều khiển. AST thường là kết quả của giai đoạn phân tích cú pháp khi trình biên dịch thực thi, là một đại diện trung gian của chương trình và có tác động mạnh mẽ đến đầu ra cuối cùng của trình biên dịch. Một đồ thị lời gọi biểu diễn mối quan hệ ***gọi*** giữa các chương trình con trong chương trình phần mềm. Đồ thị luồng điều khiển cung cấp "chi tiết" tốt hơn vào cấu trúc của chương trình nói chung và của các chương trình con nói riêng.

Chương 3

KIỂM CHỨNG CHÍNH SÁCH RBAC TRIỂN KHAI THEO PHƯƠNG PHÁP AN NINH LẬP TRÌNH

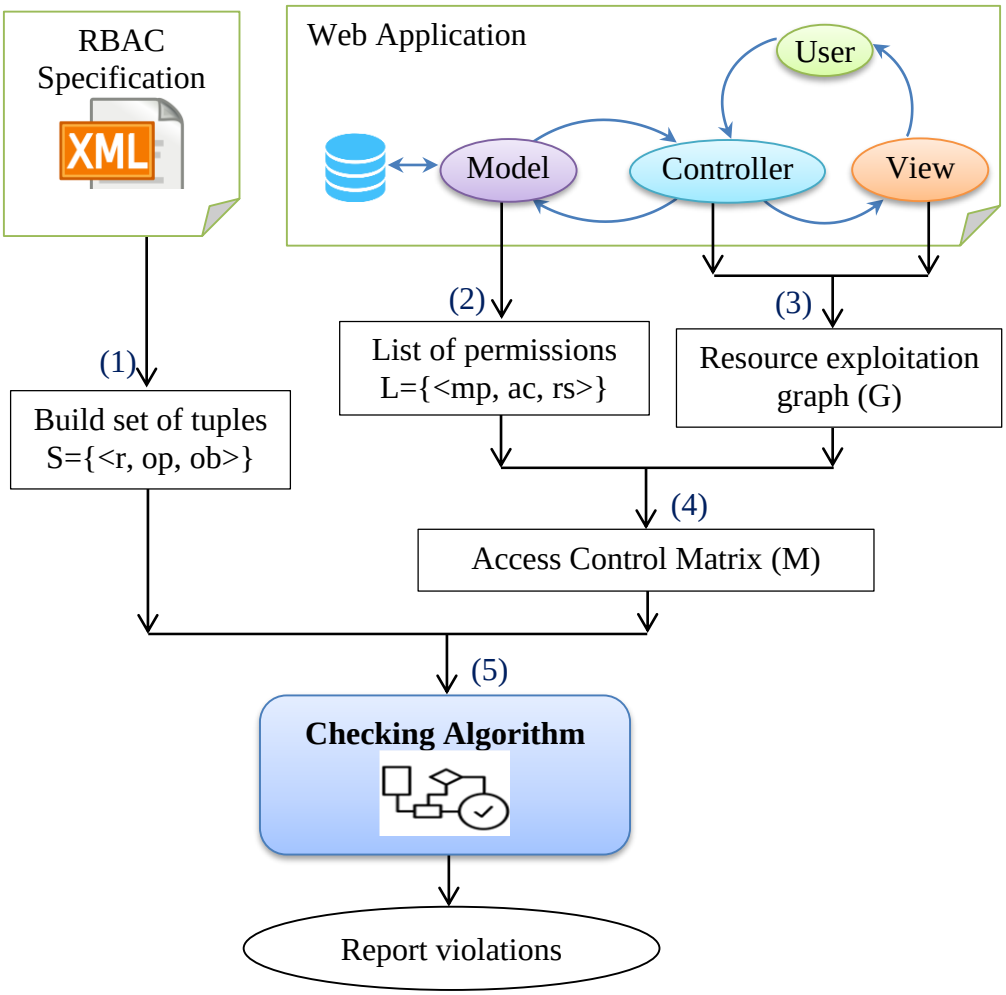
3.1. Giới thiệu

Kiến trúc MVC được sử dụng phổ biến trong việc thiết kế các ứng dụng web ở giai đoạn lập trình. Người lập trình có thể tận dụng cơ chế hoạt động của các thành phần trong kiến trúc MVC để triển khai chính sách RBAC theo phương pháp an ninh lập trình. Tuy nhiên, sự phức tạp của phương pháp triển khai này có thể phát sinh một số vấn đề như gán *thừa* hoặc *thiếu* các quyền cho các vai trò. Điều này dẫn đến việc người dùng sẽ có những chức năng không theo quy định của tổ chức và làm vi phạm các tính chất an ninh của hệ thống phần mềm.

Để kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình, ý tưởng đề xuất gồm: trình bày cấu trúc và các bước xây dựng danh sách các quyền, đồ thị khai thác tài nguyên, và ma trận kiểm soát truy cập theo vai trò của các ứng dụng web; đề xuất thuật toán kiểm tra sự phù hợp của ma trận kiểm soát truy cập theo vai trò và đặc tả chính sách RBAC của hệ thống.

3.2. Phương pháp kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình

Phương pháp kiểm chứng chính sách RBAC của ứng dụng web được thiết kế theo kiến trúc MVC gồm năm bước được mô tả tổng quát trong Hình 3.1. Đầu tiên là việc rút trích chính sách RBAC đã đặc tả của hệ thống thành tập các quy tắc truy cập. Cụ thể, luận án phân tích các phương thức khai thác tài nguyên thành danh sách các quyền. Tiếp đó là xây dựng đồ thị khai thác tài nguyên để biểu diễn sự liên kết, khai thác tài nguyên của các trang web. Một ma trận kiểm soát truy cập theo vai trò được giới thiệu để trực quan hóa chính sách RBAC của hệ thống. Cuối cùng là thuật toán kiểm tra sự phù hợp của ma trận kiểm soát truy cập theo vai trò với chính sách đã đặc tả. Hai kịch bản vi phạm truy cập mà phương pháp đề xuất giải quyết góp phần đảm bảo tính bí mật, tính toàn vẹn và tính sẵn sàng của các ứng dụng web.



Hình 3.1: Quy trình kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình

3.2.1. Xây dựng tập quy tắc truy cập

Nhiệm vụ của bước này là rút trích các thông tin trong chính sách truy cập để xây dựng tập quy tắc truy cập có dạng $\mathbb{S} = \langle r, op, ob \rangle$. Trong đó: r, op, ob lần lượt là thông tin về vai trò, phép toán và đối tượng (tài nguyên của hệ thống). Mỗi quy tắc $s \in \mathbb{S}$ được hiểu theo nghĩa, với vai trò r người dùng được phép thực hiện phép toán op trên đối tượng ob.

3.2.2. Danh sách các quyền

Để xây dựng danh sách các quyền của người dùng, đầu tiên cần xác định danh sách các tài nguyên của hệ thống ($\mathbb{RS}$). Tiếp theo là xác định các lớp trong $Model$ chứa các phương thức tương tác với tài nguyên. Các phương thức khai thác tài nguyên sẽ được phân tích thành các cây cú pháp trừu tượng và được rút trích thành danh sách các quyền ($\mathbb{L}$) của người dùng trong ứng dụng web. Mỗi quyền được mô tả là một bộ có ba thành phần mp, ac và rs. Với $\mathbb{L} = \{\langle mp, ac, rs \rangle\}$, trong đó: mp là khai báo nguyên

mẫu của phương thức khai thác tài nguyên chứa các thông tin như tên lớp chứa phương thức, kiểu trả về của phương thức, tên phương thức và danh sách các kiểu dữ liệu của các tham số của phương thức; $ac \in \mathbb{AC}$ là kiểu phép toán trong tập các phép toán; và $rs \in \mathbb{RS}$ là một tài nguyên trong tập tài nguyên cần bảo vệ.

3.2.3. Đồ thị khai thác tài nguyên

Trong những ứng dụng web được thiết kế theo kiến trúc MVC, thành phần *View* sẽ chứa toàn bộ danh sách các trang (**.jsp*, **.html*, v.v) của ứng dụng. Các trang này thường được thiết kế ở dạng giao diện đồ họa để tương tác với người dùng. Ở tiểu mục này, luận án đề xuất cấu trúc và các bước xây dựng đồ thị khai thác tài nguyên $\mathcal{G}$ của một ứng dụng web để biểu diễn sự liên kết giữa các trang (page) và các khai thác tài nguyên của chúng. Đồ thị $\mathcal{G}$ được xây dựng theo hai bước:

Bước 1 - Xây dựng các đỉnh và các cạnh: Các đỉnh và các cạnh của đồ thị được xây dựng theo quy tắc:

- Mỗi đỉnh của đồ thị là thông tin về một trang (*page*) có trong *View*, tên của đỉnh chính là tên của trang đó (số đỉnh của đồ thị là số trang trong ứng dụng).
- Tồn tại một cạnh đi từ đỉnh A đến đỉnh B ($A \neq B$) trên đồ thị $\mathcal{G}$ nếu từ trang A có đường liên kết đến trang B hoặc được chuyển hướng đến trang B hoặc chứa trang B.

Bước 2 - Đính kèm các phương thức khai thác tài nguyên: Để đính kèm được các phương thức khai thác tài nguyên cho các đỉnh của đồ thị cần tận dụng nguyên tắc thiết kế trong kiến trúc MVC. Mỗi trang trong *View* đều được hệ thống xử lý thông qua *servlet* của nó (tệp **.java* trong *Controller*). Bằng việc sử dụng kỹ thuật phân tích mã nguồn của tệp điều khiển này thành cây AST, sẽ dễ dàng tập hợp được danh sách các lời gọi phương thức của nó. Từ kết quả đó, tiếp tục rút trích lấy các phương thức được gọi để khai thác tài nguyên của hệ thống và đính kèm chúng vào đỉnh tương ứng trên đồ thị.

3.2.4. Ma trận kiểm soát truy cập theo vai trò

Các thông tin được mô tả trong ma trận phản ánh các phép toán trên các tài nguyên của từng vai trò bên trong ứng dụng web. Ma trận $\mathcal{M}$ được xây dựng từ đồ thị khai thác tài nguyên $\mathcal{G}$ theo quy tắc:

- Chiều đầu tiên của ma trận có tên là *ROLE*. Nó chứa tập các vai trò bên trong ứng dụng web $\mathbb{R} = \{R_1, R_2, .., R_n\}$. Thông tin về các vai trò có trong chính sách truy cập được rút trích từ kết quả phân tích tệp *.java* trong *Controller* xử lý trang đăng nhập. Bởi vì, mỗi người dùng muốn sử dụng hệ thống phải tiến hành đăng nhập vào hệ thống thông qua một tài khoản gồm *username*, *password*. Nếu việc

đăng nhập thành công thì người dùng hợp lệ sẽ được chuyển hướng đến trang nghiệp vụ tương ứng với vai trò của họ trong hệ thống.

- Chiều thứ hai của ma trận là $ACTION$, nó chứa tất cả các phép toán cơ bản dùng để khai thác tài nguyên $\mathbb{AC} = \{AC_1, AC_2, .., AC_m\}$. Mỗi AC_i có thể là $Read$, $Create$, $Delete$, hay $Update$.
- Chiều thứ ba là $RESOURCES$ gồm danh sách các tài nguyên cần bảo vệ của hệ thống $\mathbb{RS} = \{RS_1, RS_2, .., RS_p\}$. Danh sách này được xác định khi xây dựng danh sách các quyền của ứng dụng web.

3.2.5. Thuật toán kiểm tra sự phù hợp của ma trận kiểm soát truy cập theo vai trò và chính sách RBAC đã đặc tả

Sau quá trình phân tích và xây dựng được ma trận kiểm soát truy cập theo vai trò của ứng dụng web. Việc kiểm chứng chính sách RBAC triển khai trong ứng dụng web sẽ được thực hiện bởi Thuật toán 3.1.

Thuật toán 3.1 Kiểm tra ma trận kiểm soát truy cập theo vai trò và chính sách RBAC đã đặc tả

Input : $\mathbb{S}$ - chính sách truy cập tài nguyên đã đặc tả.
$\mathcal{M}$ - ma trận kiểm soát truy cập theo vai trò của ứng dụng web.
$\mathbb{R}, \mathbb{AC}, \mathbb{RS}$ - lần lượt là tập các vai trò, phép toán và các tài nguyên có trong ứng dụng web.
Output: Các báo cáo vi phạm (nếu có).
Data : s - là phần tử thuộc tập các quy tắc truy cập đã đặc tả $\mathbb{S}$.
r, ac, rs - lần lượt là phần tử thuộc tập vai trò $\mathbb{R}$, tập phép toán $\mathbb{AC}$ và tập tài nguyên $\mathbb{RS}$.

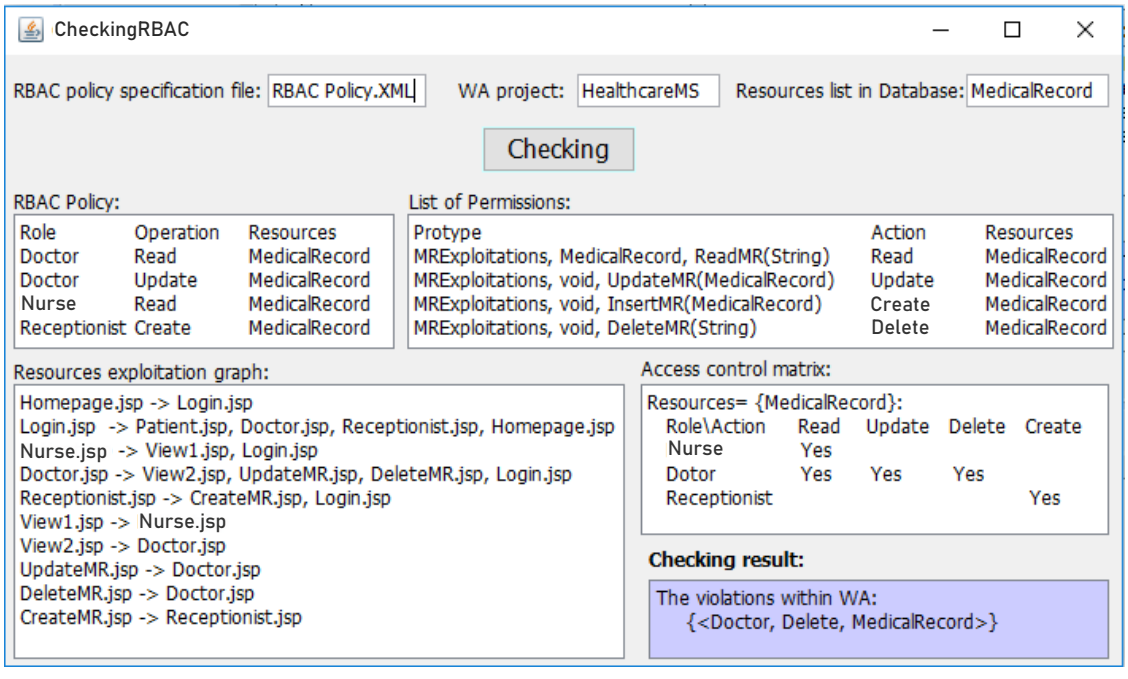
```
1  Procedure CheckingRBAC(𝕊, ℳ, ℝ, 𝔸ℂ, ℝ𝕊)
2  begin
3      foreach r_i ∈ ℝ do
4          foreach ac_j ∈ 𝔸ℂ do
5              foreach rs_k ∈ ℝ𝕊 do
6                  if ℳ[i][j][k] = True then
7                      m ← {⟨r_i, ac_j, rs_k⟩};
8                      if m ∉ 𝕊 then
9                          Write ("Redundancy:", m); (i)
10                     else
11                         𝕊 ← 𝕊 \ {m};
12     if (𝕊 ≠ ∅) then
13         foreach s ∈ 𝕊 do
14             Write ("Lack:", s); (ii)
```

Thuật toán kiểm chứng đề xuất có thể phát hiện được hai trường hợp

triển khai không chính xác quy tắc truy cập trong ứng dụng web so với sự đặc tả: (i) trong ứng dụng web có xuất hiện các quy tắc truy cập không được đặc tả, (*thừa quyền*) và (ii) ứng dụng web triển khai thiếu so các quy tắc đã đặc tả (*thiếu quyền*).

3.3. Công cụ kiểm chứng

Công cụ kiểm chứng được mô tả trong Hình 3.2. Dữ liệu đầu vào của công cụ kiểm chứng chính bao gồm: là tệp tin đặc tả chính sách RBAC của hệ thống (**.xml*), tên của dự án ứng dụng web cần phân tích và danh sách các tài nguyên của hệ thống. Sau khi cung cấp các dữ liệu đầu vào, công cụ sẽ tự động thực hiện lần lượt các bước theo quy trình đã đề xuất và hiển thị các kết quả phân tích của từng bước vào các mục tương ứng. Kết quả kiểm tra của các ứng dụng được hiển thị ở mục *Checking result*. Trong trường hợp chính sách truy cập của ứng dụng web không nhất quán với đặc tả. Công cụ sẽ trả về thông báo vi phạm kèm theo các thông tin liên quan đến quy tắc truy cập là nguyên nhân làm mất tính nhất quán. Điều này sẽ hỗ trợ các nhà lập trình dễ dàng trong việc rà soát mã nguồn để khắc phục nguyên nhân dẫn đến vi phạm truy cập.



Hình 3.2: Giao diện của công cụ kiểm chứng chính sách RBAC

Kết quả nghiên cứu của chương này đã được công bố tại Hội nghị *International Conference on Context-Aware Systems and Applications (ICCASA 2018)*.

Chương 4

KIỂM CHỨNG CHÍNH SÁCH RBAC KẾT HỢP RÀNG BUỘC CẤP QUYỀN TRIỂN KHAI THEO PHƯƠNG PHÁP AN NINH KHAI BÁO

4.1. Giới thiệu

Khi chính sách truy cập của hệ thống có sự tham gia của các ràng buộc cấp quyền thì chính sách truy cập của hệ thống được biểu diễn với mô hình SecureUML. Với phương pháp an ninh khai báo, việc triển khai chính sách RBAC được thực hiện thông qua các chú thích. Việc triển khai theo phương pháp này mang đến sự tin cậy cao, tiết kiệm thời gian, v.v. Tuy nhiên, việc lập trình và ghép nối các thành phần, sử dụng các thư viện bên thứ ba luôn có khả năng chứa đựng các lỗi tiềm ẩn do không hiểu đúng hoặc tự phát sinh lỗi khi có những kết hợp phức tạp. Do đó, sự thực thi các chức năng của các hệ thống phần mềm sau khi cài đặt có thể sẽ không chính xác như các yêu cầu ban đầu của nó.

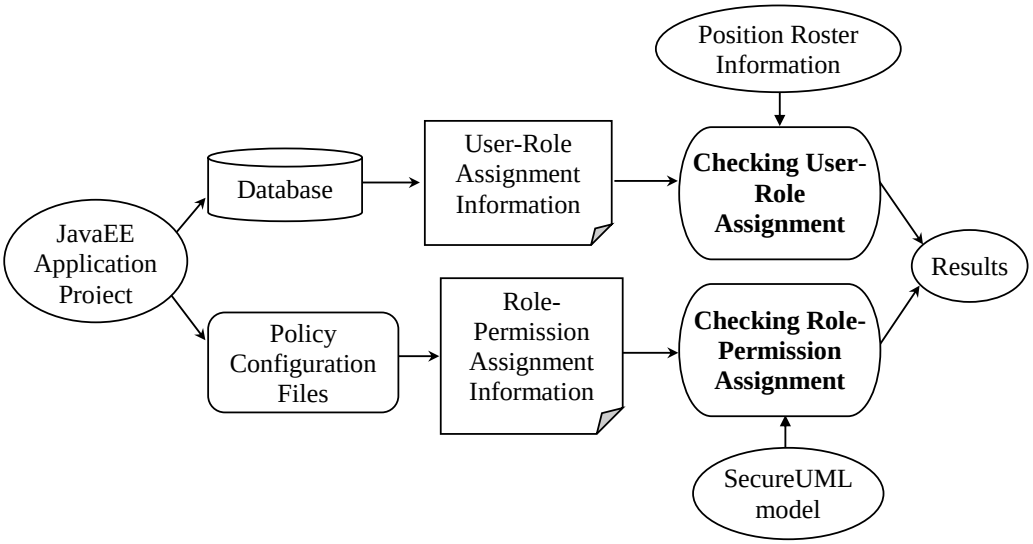
Để kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo, các bước cần thực hiện gồm: đề xuất phương pháp phân tích cơ sở dữ liệu và các tệp cấu hình chính sách truy cập của hệ thống; giới thiệu cấu trúc cây phân tích truy cập theo vai trò để biểu diễn tập quy tắc truy cập của ứng dụng web; đề xuất các thuật toán để phát hiện các phép gán vai trò-người dùng và gán vai trò-quyền được triển khai không chính xác trong ứng dụng web dẫn đến các vi phạm truy cập; triển khai một công cụ để hỗ trợ quá trình kiểm chứng tự động.

4.2. Phương pháp kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo

Tổng quan về quy trình kiểm chứng đề xuất được mô tả trong Hình 4.1 . Phương pháp đề xuất tập trung giải quyết hai vấn đề liên quan đến phép gán vai trò-người dùng và phép gán vai trò-quyền.

4.2.1. Kiểm tra phép gán vai trò - người dùng

Để tiến hành kiểm tra phép gán vai trò - người dùng được triển khai trong ứng dụng web, công việc đầu tiên cần thực hiện là rút trích các thông tin về mối liên hệ giữa vai trò và người dùng trong cơ sở dữ liệu



Hình 4.1: Quy trình kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo

Thuật toán 4.1 Kiểm tra phép gán vai trò - người dùng

Input : URI_Table là bảng thông tin về người dùng và vai trò trong hệ thống ứng dụng.
RR_Table là bảng phân công vai trò của tổ chức.
Output: Các báo cáo vi phạm
Data : *row* là một đối tượng có chứa thông tin về người dùng *(id, fullName, roleName)*.

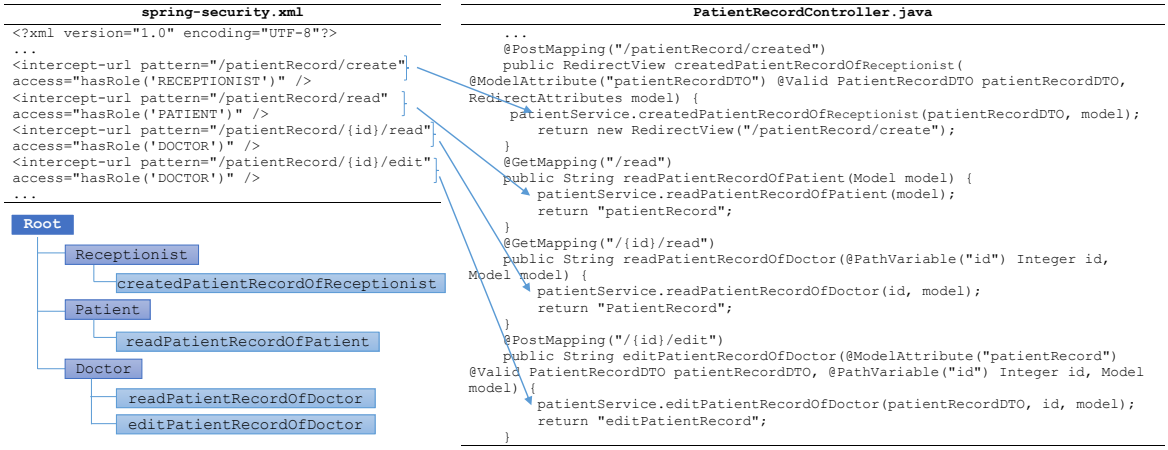
```
1 Procedure URChecking(URI_Table, RR_Table)
2 begin
3     foreach row ∈ URI_Table do
4         if CheckExist(row, RR_Table) == false then
5             Write ("Redundancy:", row);
6     foreach row ∈ RR_Table do
7         if CheckExist(row, URI_Table) == false then
8             Write ("Lack:", row);
```

của ứng dụng web. Công việc này được thực hiện dựa trên lược đồ cơ sở dữ liệu của hệ thống và câu lệnh truy vấn của SQL. Thuật toán 4.1 mô tả các bước để phát hiện sự dư thừa hoặc thiếu phép gán giữa các vai trò và người dùng trong ứng dụng web. Đầu vào của thuật toán này là bảng thông tin về người dùng và vai trò của họ trong ứng dụng và tổ chức. Kết quả đầu ra của thuật toán là các báo cáo vi phạm (nếu có). Báo cáo vi phạm xảy ra khi có sự xuất hiện của một phép gán vai trò - người dùng

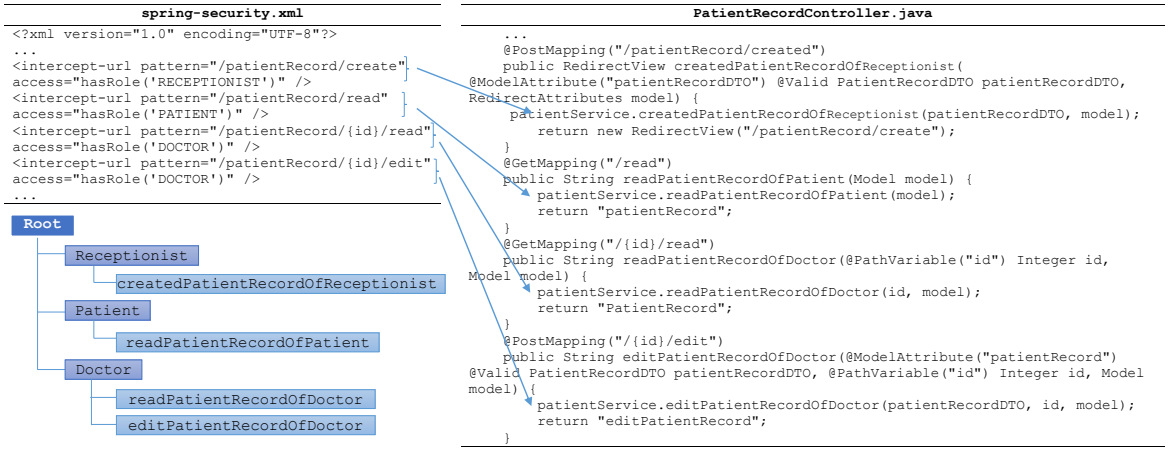
trong cơ sở dữ liệu của ứng dụng web không có trong bảng phân công chức vụ cho người dùng của tổ chức, hoặc một phân công chức vụ cho người dùng của tổ chức không được triển khai trong cơ sở dữ liệu của ứng dụng web.

4.2.2. Kiểm tra phép gán vai trò - quyền

Một cây phập tích truy cập theo vai trò $\mathcal{T}$ được giới thiệu để trực quan hóa chính sách RBAC và các ràng buộc cấp quyền được phân tích từ ứng dụng web. Hình 4.2, 4.3, và 4.4 mô tả tiến trình xây dựng cây phân tích theo vai trò của các ứng dụng web thông qua việc áp dụng với hệ thống quản lý hồ sơ y tế minh họa.

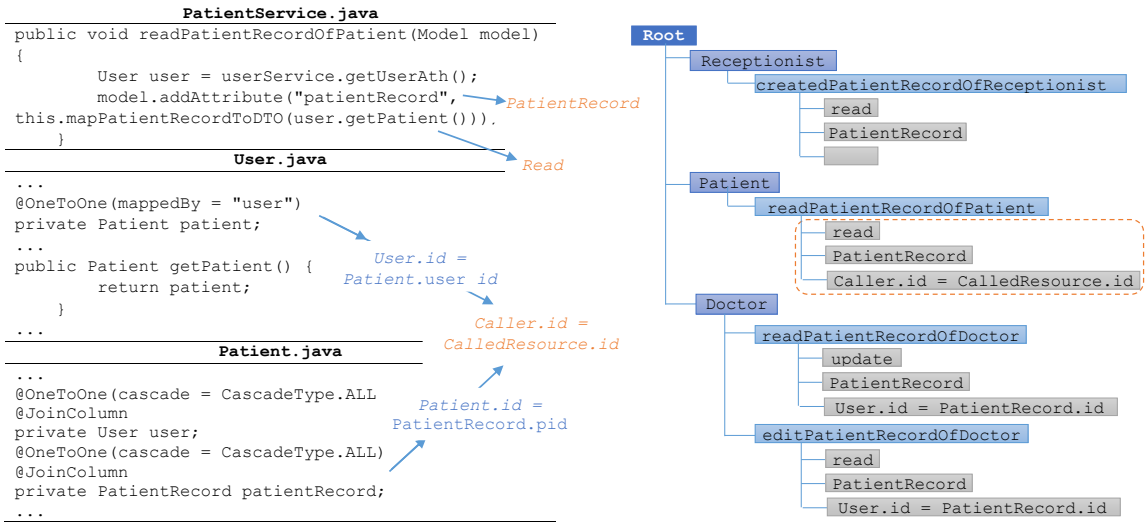


Hình 4.2: Xây dựng các vai trò



Hình 4.3: Xây dựng các vai trò

Sau khi hoàn thành việc xây dựng cây phân tích của ứng dụng web ($\mathcal{T}$), Thuật toán 4.2 sẽ được áp dụng để kiểm tra sự phù hợp của phép gán vai trò - quyền được biểu diễn trên cây $\mathcal{T}$ với chính sách đã đặc tả ($SFile$). Kết quả trả về của thuật toán này là các thông báo vi phạm trong việc



Hình 4.4: Xây dựng các vai trò

triển khai phép gán vai trò - quyền của ứng dụng web (nếu có).

Thuật toán 4.2 Kiểm tra phép gán vai trò - quyền

Input : $SFile$ là tệp tin chứa đặc tả chính sách RBAC và ràng buộc cấp quyền của hệ thống.
$\mathcal{T}$ là cây phân tích truy cập theo vai trò của ứng dụng web.
Output: Các báo cáo vi phạm
Data : $role, prm, act, res, cond$ lần lượt là vai trò, kiểu phép toán, tài nguyên và điều kiện.
ar là một đối tượng chứa các thông tin về một quy tắc truy cập

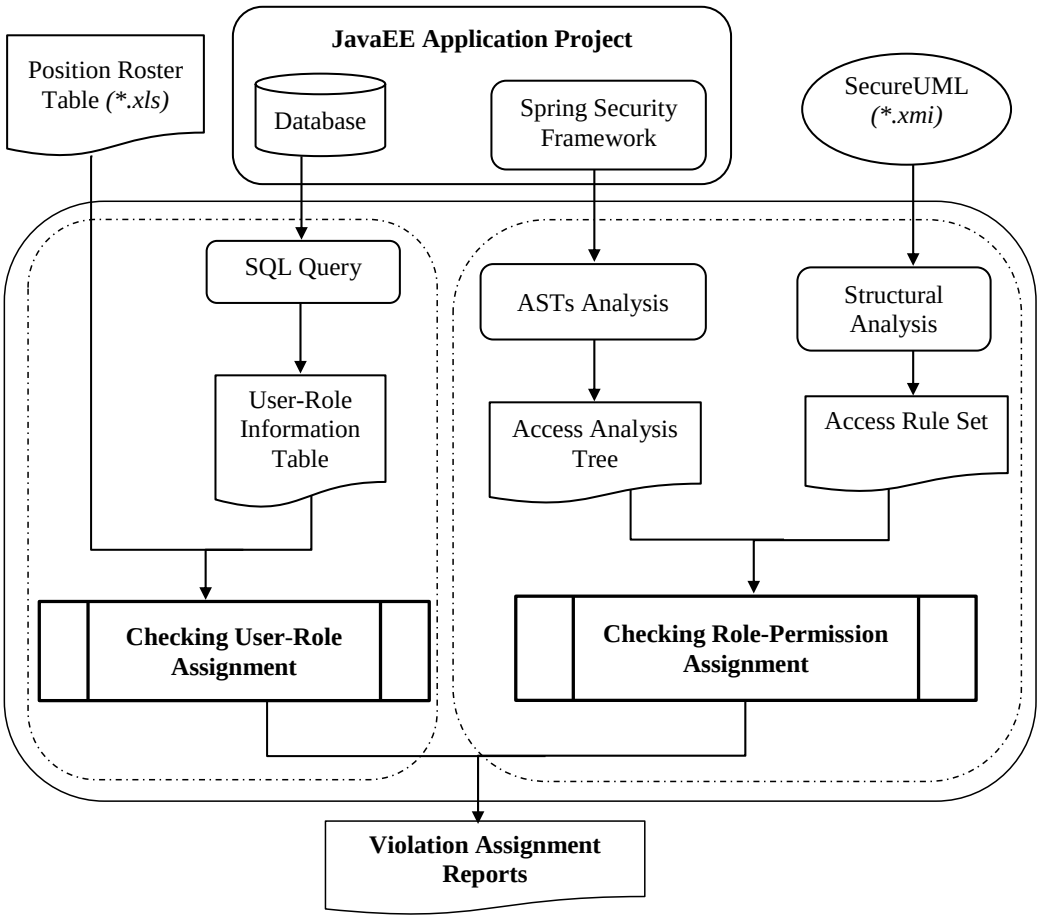
```
1  Procedure UPChecking(SFile, T)
2  begin
3      foreach role ← GetChildren(T, root) do
4          foreach prm ← GetChildren(T, role) do
5              act ←GetChildren(T, prm);
6              res ←GetChildren(T, prm);
7              cond ←GetChildren(T, prm);
8              ar = ⟨role, act, res, cond⟩;
9              if CheckSpecifiation(ar, SFile) == false then
10                 Write ("Redundancy:", ar);
11     foreach ar ← GetElement(SFile) do
12         if CheckTree(ar, T) == false then
13             Write ("Lack:", ar);
```

Trong cả hai thuật toán kiểm tra phép gán vai trò - người dùng và vai trò - quyền, ngoài việc phát hiện các phép gán vi phạm trong việc triển

khai chính sách truy cập của hệ thống dẫn đến không đảm bảo tính bảo mật, tính toàn vẹn và tính sẵn sàng của hệ thống ứng dụng. Thuật toán còn cung cấp các thông tin của các phép gán là nguyên nhân của sự vi phạm để các nhà lập trình có thể lần vết và khắc phục các sai sót.

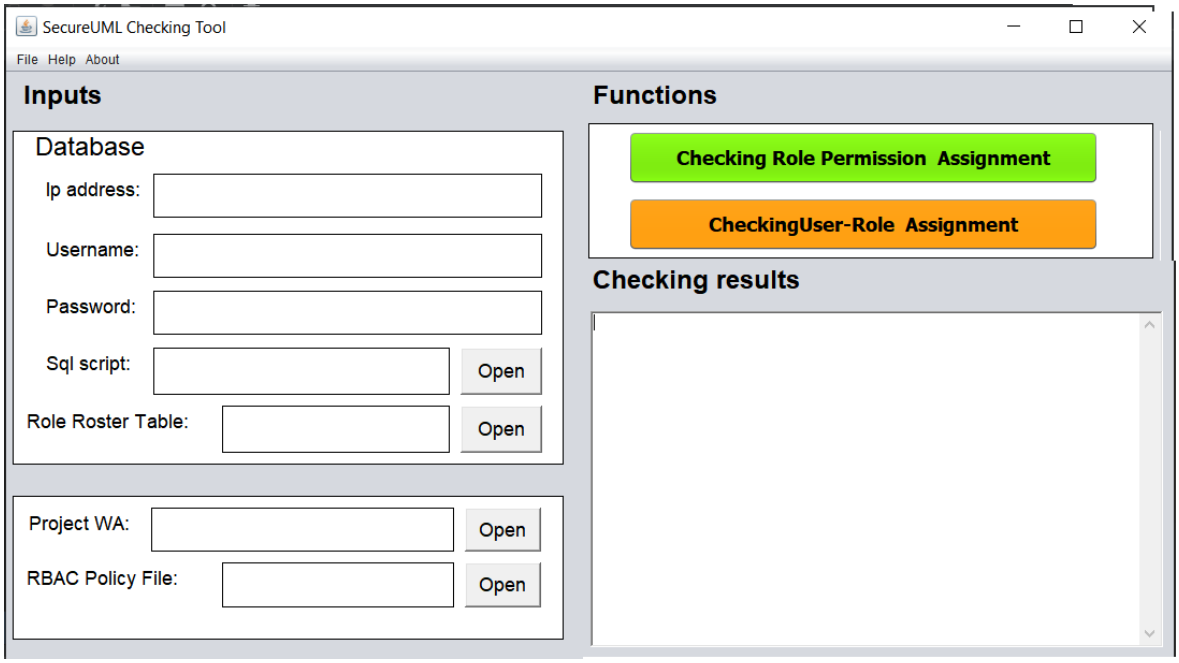
4.3. Triển khai công cụ

Kiến trúc của công cụ hỗ trợ được mô tả trong Hình. 4.5 với hai chức năng chính là kiểm tra phân công vai trò người dùng ($ChekingRolePermissionAssi$) và kiểm tra phân quyền cho phép vai trò ($ChekingUserRoleAssignment$). Hai chức năng này đều có nhiệm vụ phát hiện các phép gán được thực hiện không chính xác trong ứng dụng web. Tuy nhiên, để thực hiện các chức năng này, công cụ phải thực hiện ba chức năng nhỏ *SQL Query*, *AST Analysis* và *Structural Analysis* để tạo đầu vào cho các thuật toán kiểm tra. Các chức năng này được thực hiện tự động phía sau giao diện của công cụ. Cụ thể, hàm *SQL Query* lấy cơ sở dữ liệu của hệ thống đầu vào và trả về bảng thông tin vai trò người dùng. Hàm *AST Analysis* chịu trách



Hình 4.5: Kiến trúc của công cụ kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền

nhiệm phân tích các tệp cấu hình chính sách và trả về cây phân tích truy cập. Hàm *Structural Analysis* chuyển đổi chính sách truy cập hệ thống đã chỉ định thành một tập quy tắc truy cập.



Hình 4.6: Giao diện của công cụ kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền

Phương pháp đề xuất của chúng tôi đang được triển khai thành công cụ kiểm tra với giao diện đồ họa được trình bày trong Hình. 4.6. Để thực thi các chức năng của công cụ hỗ trợ là kiểm tra phân công vai trò - quyền *ChekingRolePermissionAssignment* và kiểm tra phân công vai trò - người dùng *ChekingUserRoleAssignment*, người dùng phải cung cấp thông tin đầu vào liên quan đến tên dự án và tên tệp đặc tả chính sách của ứng dụng web; các thông tin về cơ sở dữ liệu và tệp chứa bảng danh sách chức của tổ chức. Quá trình kiểm tra các phép gán được thực hiện tự động và trả lại các thông báo phép gán không được triển khai chính xác (nếu có). Trong trường hợp công cụ dò tìm ra các phép gán được triển khai không chính xác, công cụ sẽ hiển thị thông tin về chúng trên màn hình kết quả để lập trình viên có thể dễ dàng lần vết và sửa lại.

Các kết quả nghiên cứu của chương này đã được công bố tại Hội nghị ACIIDS 2017 (*Asian Conference on Intelligent Information and Database Systems*), đã gửi và đang chờ kết quả phản biện của Tạp chí IJSEKE (*International Journal of Software Engineering and Knowledge Engineering*).

Chương 5

KIỂM CHỨNG CHÍNH SÁCH KIỂM SOÁT TRUY CẬP THEO THUỘC TÍNH (ABAC)

5.1. Giới thiệu

Việc triển khai chính sách kiểm soát truy cập vào các hệ thống ứng dụng web rất đa dạng và thường có sự kết hợp phức tạp của nhiều khung làm việc. Với các ứng dụng web được phát triển bởi JavaEE, kiến trúc MVC và thư viện bên thứ ba như Spring Security là phương án triển khai được nhiều nhà lập trình lựa chọn. Mặc dù các thư viện đều cung cấp các cơ chế để đảm bảo các tính chất an ninh cho các hệ thống phần mềm, nhưng đôi khi việc kết hợp nhiều cơ chế, lạm dụng các thư viện hoặc sự phức tạp trong lập trình cũng có thể gây xung đột chức năng, lãng phí thời gian và công sức cũng như phát sinh các lỗ hổng bảo mật tiềm ẩn trong các hệ thống phần mềm. Thêm vào đó, các hệ thống triển khai chính sách ABAC thường lớn và có thể rất phức tạp để quản lý. Tính biểu đạt cao của các đặc tả ABAC cũng làm tăng khả năng có khiếm khuyết. Do đó, việc kiểm tra và xác minh để đảm bảo tính chính xác của chính sách ABAC được triển khai trong hệ thống là rất quan trọng.

Các công việc cần tiến hành để kiểm chứng chính sách kiểm soát truy cập theo thuộc tính gồm: phương pháp phân tích, tổng hợp tập quy tắc truy cập của ứng dụng web; định nghĩa hình thức và các thuật toán để kiểm tra sự phù hợp của ứng dụng web và đặc tả thông qua tính bảo mật, tính toàn vẹn và tính sẵn sàng; xây dựng công cụ hỗ trợ quá trình kiểm chứng tự động.

5.1.1. Phân tích chính sách ABAC

Bởi vì khung làm việc của Spring Security chỉ hỗ trợ trực tiếp việc xác thực và phân quyền theo mô hình RBAC cho nên chính sách ABAC sẽ được triển khai trong ứng dụng web theo cả hai phương pháp an ninh lập trình và an ninh khai báo. Khung làm việc để triển khai chính sách ABAC vào các ứng dụng web là rất quan trọng. Bởi vì, nó quy định vị trí cũng như cấu trúc của các mã an ninh triển khai chính sách truy cập của hệ thống. Việc tổng hợp tập quy tắc truy cập của ứng dụng web sẽ được tiến hành từ việc phân tích mã an ninh lập trình và mã an ninh khai báo.

Định nghĩa 5.1 *Tập quy tắc truy cập:*

(i) Tập quy tắc truy cập được phân tích từ mã nguồn của ứng dụng web có cấu trúc $\mathbb{AR} = \{AR_1, AR_2, ..., AR_n\}$.

(ii) Mỗi quy tắc truy cập $AR_i = (target, condition)$ *gồm hai thành phần là các biểu thức logic được viết bằng SpEL. Trong đó: target mô tả khả năng áp dụng của quy tắc và condition mô tả điều kiện thực hiện của target.*

Vì các dự án web được thiết kế từ JavaEE, Trong bước này, thư viện *JDT* sẽ được tích hợp vào môi trường soạn thảo như Eclipse để phân tích các tệp $*.java$ thành các cây cú pháp trừu tượng (AST). Từ đó, tất cả các "*target*" và "*condition*" của các phương thức truy cập tài nguyên trong các ứng dụng được tập hợp. Ở mức phương thức Java, Spring Security cho phép định nghĩa chính sách kiểm soát truy cập theo hai cách bằng cách cấu hình trong tệp *.xml* hoặc dùng các chú thích trong tệp *.java*. Vì vậy, sau khi phân tích tệp này sẽ thu thập được thông tin về các phương thức và vai trò. Từ đó tiếp tục xây dựng các *"target"* và *"condition"* của các quy tắc truy cập. Tổng hợp kết quả sau hai quá trình phân tích, ta thu được tập các quy tắc truy cập của ứng dụng web.

5.1.2. Các thuật toán kiểm tra chính sách ABAC của ứng dụng web

Chính sách ABAC của một ứng dụng web được xem là phù hợp với đặc tả của nó nếu không có sự triển khai dư thừa hoặc thiếu các quy tắc truy cập đã đặc tả. Việc triển khai thừa quy tắc truy cập trong các ứng dụng web có thể dẫn tới việc người dùng có thừa các chức năng thao tác với tài nguyên so với quy định. Khi đó, người dùng có thể xem hoặc thay đổi tài nguyên của ứng dụng. Tuy nhiên, nếu các ứng dụng web triển khai thiếu các quy tắc truy cập thì người dùng không có đủ các chức năng để thực hiện công việc của họ. Nói cách khác, nếu triển khai không chính xác chính sách truy cập sẽ dẫn đến việc không đảm bảo được *Tính bảo mật*, *Tính toàn vẹn* và *Tính sẵn sàng* của hệ thống.

5.1.2.1. Tính bảo mật

Định nghĩa 5.2 *(Tính bảo mật) Cho chính sách ABAC của hệ thống* $\mathbb{P} = \{R_1, R_2, ..., R_n\}$ *và tập các quy tắc truy cập của ứng dụng web* $\mathbb{AR} = \{AR_1, AR_2, ..., AR_m\}$. *Ứng dụng web được gọi là đảm bảo tính bảo mật nếu* $\nexists AR_i \in \mathbb{AR} \wedge op(AR_i) =$*"Read"*: $AR_i \notin \mathbb{P}$.

5.1.2.2. Tính toàn vẹn

Định nghĩa 5.3 *(Tính toàn vẹn) Cho chính sách ABAC của hệ thống* $\mathbb{P} = \{R_1, R_2, ..., R_n\}$ *và tập các quy tắc truy cập của ứng dụng web* $\mathbb{AR} = \{AR_1, AR_2, ..., AR_m\}$. *Ứng dụng web được gọi là đảm bảo tính toàn vẹn nếu* $\nexists AR_i \in \mathbb{AR} \wedge op(AR_i) \in \{$*"Read", "Update","Delete",*$\} : AR_i \notin \mathbb{P}$.

Thuật toán 5.1 Kiểm tra tính bảo mật

Input : $\mathbb{P}$ là chính sách ABAC theo đặc tả.
$\mathbb{AR}$ là tập các quy tắc truy cập của ứng dụng web
Output: Kiết quả kiểm chứng
Data : R, AR lần lượt là các phần tử của tập $\mathbb{P}, \mathbb{AR}$.
i, j là các số nguyên.

```
1  Function isConfidentiality(ℙ, 𝔸ℝ)
2  begin
3      for i = 1 to |𝔸ℝ| do
4          for j = 1 to |ℙ| do
5              if isEquivalent(AR_i, R_j) then
6                  break;
7          if j > |ℙ| then
8              if getAction(AR_i) = "Read" then
9                  Write ("No Confidentiality:", AR_i);
10                 return false;
11     return true;
```

Thuật toán 5.2 Kiểm tra tính toàn vẹn

Input : $\mathbb{P}$ là chính sách ABAC theo đặc tả.
$\mathbb{AR}$ là tập các quy tắc truy cập của ứng dụng web
Output: Kiết quả kiểm chứng
Data : R, AR lần lượt là các phần tử của tập $\mathbb{P}, \mathbb{AR}$.
i, j là các số nguyên.

```
1  Function isIntegrity(ℙ, 𝔸ℝ)
2  begin
3      for i = 1 to |𝔸ℝ| do
4          for j = 1 to |ℙ| do
5              if isEquivalent(AR_i, R_j) then
6                  break;
7          if j > |ℙ| then
8              if getAction(AR_i) ∈ {"Create", "Update", "Delete"} then
9                  Write ("No Integrity:", AR_i);
10                 return false;
11     return true;
```

5.1.2.3. Tính sẵn sàng

Định nghĩa 5.4 *(Tính sẵn sàng)*

Cho chính sách ABAC của hệ thống $\mathbb{P} = \{R_1, R_2, ..., R_n\}$ *và tập các chính sách truy cập của ứng dụng web* $\mathbb{AR} = \{AR_1, AR_2, ..., AR_m\}$*. Ứng dụng web được gọi là đảm bảo tính sẵn sàng nếu* $\forall R_i \in \mathbb{P} : R_i \in \mathbb{AR}$*.*

Thuật toán 5.3 Kiểm tra tính sẵn sàng

Input : $\mathbb{P}$ là chính sách ABAC theo đặc tả.
$\mathbb{AR}$ là tập các quy tắc truy cập của ứng dụng web
Output: Kiết quả kiểm chứng
Data : R, AR lần lượt là các phần tử của tập $\mathbb{P}, \mathbb{AR}$.
i, j là các số nguyên.

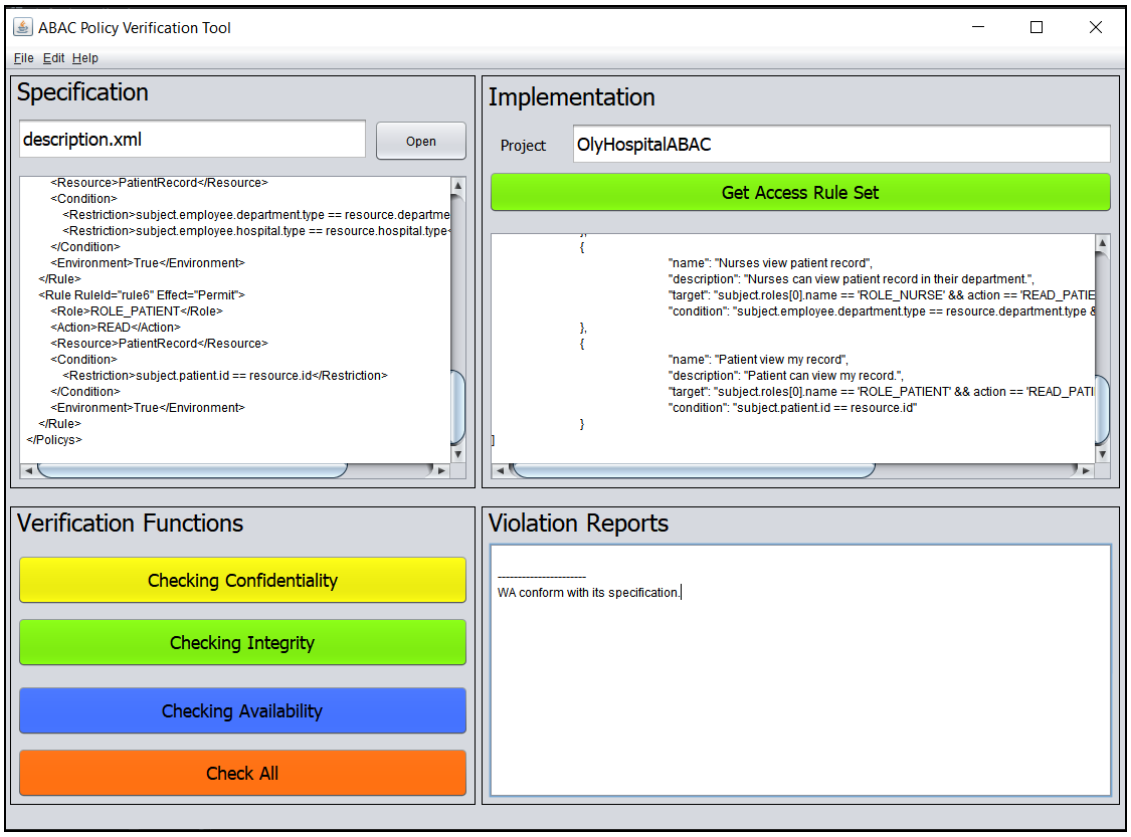
```
1  Function isAvailability(ℙ, 𝔸ℝ)
2  begin
3      for i = 1 to |ℙ| do
4          for j = 1 to |𝔸ℝ| do
5              if isEquivalent(R_i, AR_j) then
6                  break;
7          if j > |𝔸ℝ| then
8              Write ("No Availability:", P_i);
9              return false;
10     return true;
```

5.2. Công cụ kiểm chứng

Phương pháp đề xuất của chương này được triển khai thành công cụ có tên là *APV* để hỗ trợ quá trình kiểm chứng tự động. Giao diện đồ họa của công cụ *APV* được mô tả trong Hình. 5.1. Thông tin đầu vào của công cụ này bao gồm tệp đặc đặc tả chính sách ABAC của hệ thống và mã nguồn dự án triển khai ứng dụng. Chúng được thể hiện lần lượt trong các mục *Specification* và *Implementation*. Sau khi cung cấp dữ liệu đầu vào cho công cụ, chức năng *Get Access Rule Set* sẽ có nhiệm vụ rút trích tập quy tắc truy cập của ứng dụng và biểu diễn chúng bằng SpEL.

Việc kiểm tra sự phù hợp của ứng dụng web với đặc tả được công cụ thể hiện thông qua ba chức năng *Checking Confidentiality*, *Checking Integrity* và *Checking Availability* tương ứng với việc kiểm tra tính bảo mật, tính toàn vẹn và tính sẵn sàng của ứng dụng web. Với mỗi chức năng này, hệ thống sẽ thực hiện dò tìm tự động và báo cáo các quy tắc truy cập. Chức năng *Checking All* thực hiện tổng hợp cả ba chức năng trên để kiểm tra

sự phù hợp của ứng dụng web và đặc tả của nó.



Hình 5.1: Giao diện đồ họa của công cụ *APV*

Các kết quả nghiên cứu của chương này đã được công bố tại Hội nghị NICS 2019 (*NAFOSTED Conference on Information and Computer Science*), đã gửi và đang chờ kết quả phản biện của Tạp chí khoa học VNU (*VNU Journal of Science: Computer Science and Communication Engineering*).

Chương 6

KẾT LUẬN

6.1. Kết luận

Việc xây dựng một chính sách an ninh đủ mạnh cho các hệ thống phần mềm để ngăn chặn các tấn công an ninh, hạn chế rủi ro và đáp ứng các yêu cầu an ninh của khách hàng là rất quan trọng. Tuy nhiên, chính sách an ninh của hệ thống cũng cần phải được triển khai chính xác ở từng giai đoạn phát triển phần mềm. Vì vậy, công việc rà soát chính sách an ninh trong suốt quá trình triển khai có ý nghĩa thực tiễn trong việc đảm bảo chất lượng phần mềm.

Một trong những phương pháp hiệu quả để hạn chế các vi phạm truy cập tài nguyên, đảm bảo chính sách an ninh của các hệ thống phần mềm là kiểm soát truy cập. Tuy nhiên, sự phức tạp trong việc triển khai chính sách an ninh ở các giai đoạn nhất là giai đoạn lập trình làm cho hệ thống có khả năng chứa các lỗ hổng an ninh tiềm ẩn. Vì thế, bài toán kiểm chứng các chính sách truy cập của các ứng dụng web từ mã nguồn có ý nghĩa trong việc đảm bảo các tính chất an ninh cũng như làm tăng tính tin cậy của các hệ thống phần mềm của hệ thống. Sau một thời gian nghiên cứu và giải quyết bài toán, luận án đã có một số đóng góp như sau:

(i) *Đề xuất phương pháp kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình.* Chính sách truy cập theo vai trò trong các ứng dụng web rút trích thành ma trận kiểm soát theo vai trò thông qua việc phân tích các phương thức khai thác tài nguyên, xây dựng danh sách các quyền và đồ thị khai thác tài nguyên. Một thuật toán được đề xuất để kiểm tra sự phù hợp của ma trận kiểm soát truy cập theo vai trò với chính RBAC đã đặc tả. Một công cụ triển khai phương pháp đề xuất để hỗ trợ quá trình kiểm chứng tự động.

(ii) *Đề xuất phương pháp kiểm chứng chính sách RBAC kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo.* Sự phù hợp của chính sách truy cập và các ràng buộc cấp quyền trong các ứng dụng web được kiểm tra thông qua phép gán vai trò - người dùng và phép gán vai trò - quyền. Với phép gán thứ nhất, phương pháp được tiến hành dựa trên việc phân tích cơ sở dữ liệu. Việc kiểm tra phép gán còn lại, các quy tắc truy cập của ứng dụng web được phân tích và biểu diễn thành cây phân tích quy tắc truy cập tài nguyên của ứng dụng web. Cuối cùng, các thuật toán được đề xuất để thực hiện

kiểm tra tính chính xác của các phép gán. Một công cụ đang được phát triển để hỗ trợ quá trình kiểm chứng tự động.

(iii) *Đề xuất phương pháp kiểm chứng chính sách kiểm soát truy cập theo thuộc tính.* Quá trình trình kiếm chứng được thực hiện từ việc phân tích, tổng hợp các quy tắc truy cập của ứng dụng web. Sự phù hợp của chính sách kiểm soát truy cập theo thuộc tính trong các ứng dụng web được kiểm chứng thông qua các định nghĩa hình thức và ba thuật toán kiểm tra tính bảo mật, tính toàn vẹn và tính sẵn sàng của hệ thống. Từ phương pháp đề xuất. Một công cụ kiểm chứng hỗ trợ quá trình kiểm chứng tự động được phát triển từ phương pháp đề xuất.

Các kết quả của luận án đã được công bố trong các hội nghị, đã gửi và đang chờ kết quả phản biện của các tạp chí chuyên ngành trong nước và quốc tế có phản biện.

6.2. Hướng phát triển

Bước đầu, luận án mới tập trung giải quyết bài toán mà chính sách an ninh của hệ thống phần mềm liên quan đến vấn đề kiểm soát truy cập. Luận án đã đề xuất được một số phương pháp để biểu diễn hình thức, phân tích, kiểm tra các chính sách truy cập từ mã nguồn của một số ứng dụng web. Tuy nhiên, quy mô của các hệ thống bài toán nghiên cứu còn mang tính đơn giản chưa bao quát được hết các tình huống vi phạm truy cập trong thực tế. Các phương pháp phân tích đã đề xuất còn phụ thuộc vào kiến trúc thiết kế, thư viện sử dụng, phương pháp triển khai chính sách truy cập cũng như ngôn ngữ lập trình khi xây dựng các ứng dụng web. Vì vậy, đối với mỗi bài toán đã nghiên cứu vẫn còn một số hướng có thể xem xét và phát triển. Một số hướng nghiên cứu tiếp theo của luận án có thể tiến hành là:

(i) Với bài toán *Đề xuất phương pháp kiểm chứng chính sách RBAC triển khai theo phương pháp an ninh lập trình*, nghiên cứu có thể mở rộng theo hướng đề xuất một khung làm việc cho việc xây dựng kiến trúc triển khai chính sách truy cập để quá trình phân tích, xây dựng công cụ được tiến hành tự động với một lớp các bài toán lớn hơn. Trong đề xuất này, luận án cũng chưa giải quyết được các vấn đề liên quan đến thừa kế vai trò và cấp phát động các quyền, vai trò cho người dùng được triển khai trong ứng dụng web.

(ii) Trong bài toán *Đề xuất phương pháp kiểm chứng chính sách kiểm soát truy cập theo vai trò kết hợp ràng buộc cấp quyền triển khai theo phương pháp an ninh khai báo*, các vấn đề liên quan đến thừa kế vai trò và cấp phát động các quyền, vai trò cho người dùng trong ứng dụng web cũng chưa được đề cập và giải quyết.

(iii) Đối với bài toán*Đề xuất phương pháp kiểm chứng chính sách kiểm*

soát truy cập theo thuộc tính, nghiên cứu này có thể mở rộng cho các khung triển khai chính sách ABAC khác.

Bên cạnh đó, các định nghĩa hình thức được đề xuất trong trong luận án còn đơn giản. Tính tối ưu của các thuật toán chưa được đề cập cũng như tính đúng đắn của các phương pháp đề xuất chưa được chứng minh bằng các phương pháp hình thức. Vì vậy, với các nghiên cứu tiếp theo, luận án có thể xem xét việc tối ưu thuật toán và chứng minh tính đúng đắn của các phương pháp đã đề xuất.

Các phương pháp đề xuất trong luận án đều chưa giải quyết các vấn đề liên quan đến truy vết và lưu trữ các hoạt động của người dùng trong các ứng dụng web. Vì thế, các tính chất an ninh mà luận án đã đề cập và giải quyết được mới chỉ dừng lại ở tính bảo mật, tính toàn vẹn và tính sẵn sàng. Do đó, trong các nghiên cứu tiếp theo, luận án có thể tiếp tục mở rộng để triển khai với các tính chất an ninh khác.