ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

PHAN THẾ DUY

NGHIÊN CỨU CƠ CHẾ PHÁT HIỆN RÒ RỈ THÔNG TIN BẰNG PHƯƠNG PHÁP PHÂN TÍCH TĨNH TRÊN ỨNG DỤNG ANDROID

ĐỀ CƯƠNG LUẬN VĂN THẠC SĨ

Ngành: **Công Nghệ Thông Tin**

TP. HỒ CHÍ MINH – 2015

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

PHAN THẾ DUY

NGHIÊN CỨU CƠ CHẾ PHÁT HIỆN RÒ RỈ THÔNG TIN BẰNG PHƯƠNG PHÁP PHÂN TÍCH TĨNH TRÊN ỨNG DỤNG ANDROID

ĐỀ CƯƠNG LUẬN VĂN THẠC SĨ

Ngành: **Công Nghệ Thông Tin**
Mã ngành: **60.48.02.01**

Người hướng dẫn khoa học: **TS. Phạm Văn Hậu**

TP. HỒ CHÍ MINH – 2015

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

ĐỀ CƯƠNG ĐỀ TÀI LUẬN VĂN THẠC SĨ

1. Tên đề tài hoặc hướng nghiên cứu:

Tên Tiếng Việt: **Nghiên cứu cơ chế phát hiện rò rỉ thông tin bằng phương pháp phân tích tĩnh trên ứng dụng Android.**

Tên Tiếng Anh: **Static Analysis for detecting sensitive data leakage on Android applications.**

2. Ngành và mã ngành đào tạo: Ngành Công nghệ Thông Tin, mã ngành: 60.48.02.01

3. Thông tin học viên và người hướng dẫn:

Họ tên học viên thực hiện đề tài: Phan Thế Duy

Địa chỉ email: phantheduy.ptd@gmail.com

Điện thoại liên lạc của học viên: 01683456056

Người hướng dẫn: TS. Phạm Văn Hậu

Địa chỉ email: haupv@uit.edu.vn

Điện thoại liên lạc của người hướng dẫn: 0915727282

4. Tổng quan tình hình nghiên cứu trong nước và ngoài nước:

4.1. **Giới thiệu chung:**

Cùng với việc các thiết bị di động như điện thoại thông minh, máy tính bảng nền tảng Android ngày càng trở nên phổ biến rộng khắp trong cuộc sống hàng ngày, mỗi ngày có hàng nghìn ứng dụng mới được các nhà phát triển công bố trên các mô hình kho ứng dụng marketplace, nơi mà người dùng có thể dễ dàng tiếp cận và tải về thiết bị Android của mình một cách tiện lợi các ứng dụng phù hợp với nhu cầu bản thân. Theo AppBrain Stats [2], kho ứng dụng Android phổ biến nhất toàn cầu Google Play đã đạt ngưỡng 1.400.000 ứng dụng vào thời điểm tháng 11, năm 2014. Cũng theo một số lượng thống kê [1], có hơn 50 tỉ lượt tải ứng dụng về thiết bị cài đặt từ Google Play tính đến tháng 7, năm 2013; và có hơn 1 tỷ người dùng thông qua thiết bị Android đã được kích hoạt.

Tuy nhiên, các nhà nghiên cứu đã chỉ ra rằng những ứng dụng Android có thể ẩn chứa những mối nguy hiểm tiềm tàng bằng khả năng được truy cập, tiếp cận vào các dữ liệu thông tin cá nhân quan trọng, nhạy cảm của người dùng, như là vị trí địa lý hiện tại, danh bạ liên hệ, thông tin định danh của thiết bị (IMEI, số điện

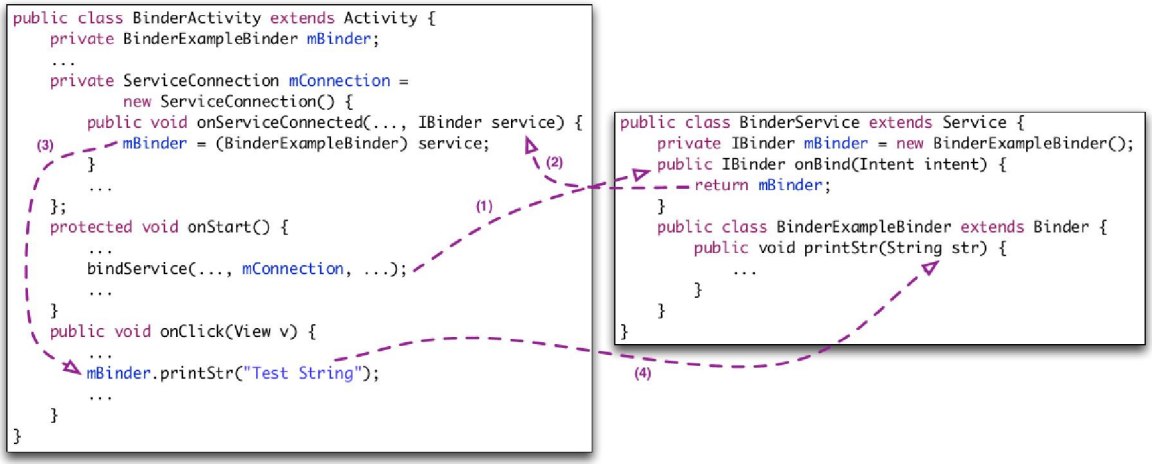
thoại), … và thường xuyên gửi ra bên ngoài những thông tin cá nhân mà không được sự đồng ý từ người dùng [28]. Những ứng dụng mang tính giải trí hay nội dung trò chơi thường được kẻ xấu lợi dụng để thiết lập những lỗ hổng đánh cắp dữ liệu cá nhân trên thiết bị người dùng, nơi mà nhiều ứng dụng như mạng xã hội hay các phần mềm tài chính, ngân hàng có thể thu thập và lưu trữ một khối lượng lớn dữ liệu về thông tin cá nhân người sử dụng. Các lo ngại ngày càng tăng lên trong việc thông tin cá nhân của người dùng bị rò rỉ ra bên ngoài trái phép, vi phạm chính sách về quyền riêng tư, kèm với việc theo dõi hành vi người sử dụng mà không có sự đồng ý. Bằng chứng là, trên kho ứng dụng Google Play phổ biến nhất hiện nay của nền tảng Android có hàng nghìn ứng dụng độc hại truy cập, sử dụng, đánh cắp dữ liệu nhạy cảm của người dùng thiết bị khi cài đặt, sử dụng một ứng dụng mà chính họ không hay biết được, từ số lượng khoảng 11.000 ứng dụng độc hại được ghi nhận năm 2011, tăng lên hơn 42.000 trường hợp ứng dụng malware được phát hiện trên Google Play, theo số liệu được công bố vào tháng 2, năm 2014 của RiskIQ [20] , một công ty chuyên về các giải pháp an ninh trực tuyến của Hoa Kỳ. Các ứng dụng này được xác định là các chương trình Android độc hại gây rò rỉ thông tin cá nhân hay những dữ liệu nhạy cảm của người dùng. Theo dữ liệu được công bố năm 2013 bởi Kaspersky [12], có khoảng 148.427 ứng dụng di động nguy hại khác nhau được chia thành 777 nhóm và 98.05% số đó được ghi nhận thuộc về các ứng dụng Android trên kho ứng dụng Google Play và các bên thứ ba. Bên cạnh đó, một nghiên cứu được báo cáo bởi Securelist [22], [16] chỉ ra rằng gần một nửa các ứng dụng nguy hiểm được phát hiện trên Android là Trojan có khả năng đánh cắp thông tin người sử dụng được lưu trữ trên các thiết bị smartphone.

Nền tảng di động Android phổ biến hiện nay, sử dụng hệ thống quyền hạn (permission system) được cấp phát khi cài đặt ứng dụng để thiết lập giới hạn đặc quyền mà ứng dụng được hoạt động truy cập vào các nguồn dữ liệu trên thiết bị. Tuy nhiên, hệ thống quyền hạn này không đủ hiệu quả để ngăn chặn việc đánh cắp hay rò rỉ thông tin nhạy cảm, hay dữ liệu cá nhân quan trọng từ phía người dùng được lưu trữ trên thiết bị di động.

Giới nghiên cứu an ninh, bảo mật đưa ra 2 phương pháp cơ bản: phân tích tĩnh và phân tích động, để phát hiện những lỗ hổng rò rỉ thông tin cá nhân trên các ứng dụng Android. Phương pháp phân tích động như TaintDroid [8], liên quan đến việc nghiên cứu cơ chế hành vi của ứng dụng bằng cách thực thi nó trên một môi trường cụ thể, thí dụ như xác định các hành vi của chương trình nào đó khi ứng dụng đó được cài đặt và chạy trên thiết bị Android,… Trong khi đó, phương pháp phân tích tĩnh, điển hình như FlowDroid [3] là phương pháp phân tích chương trình ứng dụng dựa trên mã ứng dụng dưới dạng mã nguồn hay mã bytecode mà không cần thực thi mã lệnh của nó. Thực tế chứng minh rằng, không phải bất cứ chương trình ứng dụng nào cũng có thể được dự đoán chính xác các hành vi bất thường của nó bằng cách cài đặt và thử nghiệm trong môi trường phân tích giả lập, một ứng dụng nguy hại có thể nhận biết nó đang bị phân tích bằng phương pháp phân tích động, và che giấu các hành vi nguy hại của mình nên kết quả phân tích sẽ không chính xác. Ngược lại, phương pháp phân tích tĩnh lại có thể đưa ra các kết quả hữu

ích bằng cách ước lượng, có thể xác định nhiều khía cạnh, các kiểu đường đi của dữ liệu nhạy cảm để chỉ ra đâu là lỗ hổng rò rỉ dữ liệu trong ứng dụng.

Một công cụ phát hiện rò rỉ thông tin cá nhân trên ứng dụng Android dựa vào phương pháp phân tích tĩnh có thể sử dụng nhiều cách thức khác nhau. Một trong những kĩ thuật hiện thực phương pháp phân tích tĩnh là phân tích dòng dữ liệu. Trong đó, "taint analysis" là trường hợp đặc biệt của cách phân tích dòng dữ liệu, nó theo vết dữ liệu đã được đánh dấu xuyên suốt các con đường lan truyền dữ liệu của ứng dụng. Trong kỹ thuật này, dữ liệu cá nhân quan trọng, nhạy cảm ở nguồn dữ liệu (source) đã được định nghĩa trước như đọc vị trí địa lý GPS, danh sách liên hệ, các thông tin định danh, xác thực… được đánh dấu với 1 "taint", từ đó theo vết dữ liệu "taint" này lan truyền xa hơn thông qua các đường chạy của chương trình như thế nào. Sự hiện diện của "taint" đã được xác định trước đó tại một nơi chứa (sink) – là nơi có khả năng gửi dữ liệu ra bên ngoài như kết nối mạng, SMS, ghi và lưu file,… đã được chỉ định được dùng để thiết lập một dòng dữ liệu giữa bộ nguồn (source) và bộ chứa (sink) [18]. Nói tóm lại, dòng chảy dữ liệu này được dùng để phát hiện sự rò rỉ dữ liệu, thông tin cá nhân từ bộ nguồn đến bộ chứa.



Hình - Ví dụ về dòng dữ liệu liên kết giữa Activity và Service

Cấu trúc ứng dụng Android bao gồm nhiều thành phần (component) khác nhau, hầu hết những lỗ hổng rò rỉ thông tin cá nhân đơn giản thực hiện hành vi của mình trong phạm vi một thành phần đơn nhất của ứng dụng. Nhưng gần đây, vấn đề thất thoát dữ liệu riêng tư được phát hiện và ghi nhận xảy ra trong phạm vi liên thành phần trong một ứng dụng, hay thậm chí là liên ứng dụng thông qua việc tương tác của các thành phần khác nhau giữa những ứng dụng khác biệt [26]. Phương pháp phân tích riêng rẽ các thành phần trong các ứng dụng, cũng như phân tích từng ứng dụng trong điều kiện tách biệt, không đủ để phát hiện nhiều hơn các mối nguy cơ rò rỉ sự riêng tư của người dùng ra bên ngoài, vì sự rò rỉ có thể gián tiếp xảy ra ở một thành phần khác trong ứng dụng, hoặc do sự thỏa hiệp giữa các

ứng dụng với nhau. Do đó, vấn đề quan trọng được đặt ra hiện nay là phải thực hiện phân tích liên thành phần trong các ứng dụng, hay còn gọi là phân tích đa ứng dụng, kết hợp tăng cường với phân tích đơn ứng dụng, đưa ra kết quả chính xác cao nhất và giảm tỉ lệ phát hiện sai lầm, để cảnh báo những nguy hiểm tiềm tàng trong các ứng dụng malware độc hại.

Các phương pháp phân tích tĩnh dòng dữ liệu bằng kỹ thuật đánh dấu "taint flow analysis" có khả năng phân tích liên thành phần, đa ứng dụng trở nên cần thiết và hứa hẹn nhiều kết quả nổi trội so với các phương pháp khác trong việc xác định dữ liệu cá nhân trên thiết bị Android được lưu trữ và sử dụng trong giới hạn mong muốn như thế nào, đề phòng nguy cơ rò rỉ dữ liệu nhạy cảm từ những ứng dụng tiềm ẩn độc hại nếu được cài đặt vào thiết bị người dùng.

4.2. Các thách thức

- Đầu tiên, khác với những chương trình ứng dụng trên Windows, một ứng dụng Android có thể được cấu thành từ ít nhất bởi một hoặc nhiều hơn từ 4 thành phần cơ bản: Activity, Service, Content provider, Broadcast Receiver. Chúng hoạt động tương tác với nhau trong một ứng dụng, ngoài ra một thành phần trong ứng dụng này có thể giao tiếp hoạt động tương tác với các thành phần của một ứng dụng khác bằng cơ chế ICC – (Inter Component Communication), và do đó các con đường luân chuyển, truyền dữ liệu trong một ứng dụng Android không thể xác định một cách rõ ràng, tường minh. Một ứng dụng Android có thể được xem là an toàn đối với thông tin của người dùng nếu phân tích riêng biệt độc lập; tuy nhiên trong nhiều trường hợp, khi hoạt động cùng với một ứng dụng khác trên cùng thiết bị Android thì nó sẽ trở thành một ứng dụng độc hại, hoặc được khai thác như một công cụ để đánh cắp thông tin nhạy cảm vì các ứng dụng này có khả năng tương tác, thỏa hiệp để thực hiện các hành vi rò rỉ dữ liệu của người dùng.

  Activity: Là thành phần quan trọng của bất kỳ một ứng dụng Android nào. Thuật ngữ Activity chỉ một việc mà người dùng có thể thực hiện trong một ứng dụng Android, gần như mọi activity đều tương tác với người dùng thông qua một giao diện nhất định. Nói tóm lại, nó là một thành phần của ứng dụng cung cấp một giao diện mà người dùng có thể tương tác vào đó để thực hiện một hành động nhất định.

  Service: Một service là một thành phần của ứng dụng, thể hiện mong muốn ứng dụng thực hiện các hành động trong khi không tương tác với người dùng hoặc cung cấp chức năng cho các ứng dụng khác sử dụng. Nói một cách đơn giản, service là các tác vụ (task) chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó.

  Content provider: Là nơi lưu trữ và cung cấp cách truy cập dữ liệu do các ứng dụng tạo nên. Đây là cách duy nhất mà các ứng dụng có thể chia sẻ dữ liệu của nhau.

Broadcast receiver: có chức năng dùng để thu nhận các sự kiện, thông báo, thông điệp mà các ứng dụng hoặc hệ thống phát đi.

- Phương pháp phân tích tĩnh đòi hỏi phải xác định được nơi xuất phát của các nguồn dữ liệu nhạy cảm để đánh dấu theo vết dữ liệu nhằm đưa ra kết quả hữu ích đối với nhà phân tích và người dùng. Nhưng việc xác định đâu là những nguồn thông tin nhạy cảm, cần được theo dõi, bảo vệ trước nguy cơ bị rò rỉ cũng là một thách thức không nhỏ, bởi số lượng các nguồn thông tin của người dùng rất nhiều và không thể dự đoán, hay chỉ ra hết những trường thông tin nào cần được "đánh dấu" taint để theo dõi một cách rõ ràng, chính xác nhất.
- Android là một hệ thống hướng sự kiện, cấu trúc ứng dụng có rất nhiều entry-point nên không thể định hình trước toàn bộ các dòng dữ liệu như trong các cấu trúc chương trình có 1 hàm chính khởi chạy ứng dụng thường được gọi là hàm main(). Các control-flow được điều khiển bởi các sự kiện từ môi trường của một ứng dụng có thể kích hoạt các lời gọi callbacks khác nhau, do đó xác định các control flow khả nghi, các dòng dữ liệu không tường minh trong 4 module cấu thành một ứng dụng Android mà không đưa ra quá nhiều các báo động giả cũng là một thách thức lớn. Hơn thế, không phải tất cả các thành phần của một ứng dụng Android đều hoạt động theo cách thức giống nhau, do đó đòi hỏi làm như thế nào để đưa ra một mô hình phù hợp với việc phân tích dòng dữ liệu nhằm phát hiện cảnh báo rò rỉ dữ liệu trên từng bộ phận cấu thành nói trên.
- Một số ứng dụng Android được đóng gói kèm với các mã native code, trong khi chưa có một công cụ nào có phương pháp phân tích hiệu quả trên dạng ứng dụng này, phải kể đến các công cụ đình đám hiện nay như IccTA [15], AsDroid [9], DidFail [11] cũng chưa có giải pháp.

4.3. Tình hình nghiên cứu

Tình hình nghiên cứu ngoài nước:

Hiện tại, một số nhóm nghiên cứu có đề xuất công bố một số nghiên cứu kèm với công cụ phân tích tĩnh dành cho các ứng dụng Android, hiệu năng, hiệu quả của các công cụ này có khác nhau do cách mô hình hóa phương pháp phân tích tập trung vào các đối tượng thành phần của ứng dụng khác nhau, chẳng hạn một số phương pháp chỉ tập trung vào một đối tượng nhất định như xem xét hệ thống Permission và thành phần Intent – Activity trong trường hợp của PermissionFlow [23]; hay như cách tiếp cận của AsDroid [9] chỉ tập trung phân tích nguồn thông tin dạng text xuất phát từ user interface của ứng dụng.

Ngoài ra, khả năng phân tích trong điều kiện tách biệt, riêng rẽ các thành phần hay đặt chúng trong mối liên hệ ở phạm vi một hay đồng thời nhiều ứng dụng Android cũng mang đến kết quả phù hợp với nhiều nhóm đối tượng khác nhau; như IccTA [15], DidFail [11], Flow Permissions [24] có tính năng phân tích tĩnh đơn và đa ứng dụng nhưng theo những cách tiếp cận khác biệt. Đối với kỹ thuật đánh dấu dòng dữ liệu để phân tích lỗ hổng đánh cắp thông tin, vấn đề xác định source – nơi xuất phát của dữ liệu nhạy cảm, và sink – nơi có khả năng gửi thông tin ra bên

ngoài cũng được nghiên cứu một cách có hệ thống như công cụ SuSi [4], [21] với tính năng tự động phân tích đưa ra các dạng source và sink khác nhau.

CHEX [17] là một công cụ phân tích tĩnh dùng phát hiện các thành phần đánh cắp, rò rỉ thông tin trong ứng dụng Android, bằng cách theo dõi các dòng dữ liệu được đánh dấu giữa nguồn dữ liệu nhạy cảm (source) và các interface có khả năng truy cập, gửi thông tin ra bên ngoài. Tuy nhiên, nó được giới hạn tối đa là chỉ có một đối tượng mang thông tin nhạy cảm (1-object-sensitivity) được phân tích, dẫn đến sự thiếu chính xác trong thực tế.

FlowDroid [3] được công bố kèm các công trình nghiên cứu liên quan [14], [5] được đánh giá như một công cụ phát hiện rò rỉ thông tin hiệu quả dựa vào phân tích tĩnh dòng dữ liệu được đánh dấu trước đó, với tỉ lệ phát hiện sai thấp trong việc phát hiện lỗ hổng mất cắp dữ liệu trong từng thành phần đơn nhất của ứng dụng; tuy nhiên nó không hoạt động trong việc phát hiện rò rỉ với điều kiện giao tiếp liên thành phần (ICC – Inter Component Communication) trong các ứng dụng, và chỉ có khả năng phân tích độc lập một ứng dụng Android.

Epicc [19] là công cụ phân tích tĩnh giải quyết vấn đề ICC, nhưng nó chủ yếu tập trung vào lỗ hổng rò rỉ thông tin thông qua các giao tiếp, kết nối liên thành phần ICC trong ứng dụng đơn nhất. Tính trung bình, để đưa ra một kết quả phân tích cần ít hơn 2 phút cho mỗi ứng dụng trong một nghiên cứu quy mô lớn gồm 1.200 ứng dụng. Do Epicc không kiểm soát các URI trong ứng dụng, dẫn đến không thể tìm các liên kết tương tác liên thành phần ICC cho Content Provider, không phát hiện được nguy cơ rò rỉ thông tin ra bên ngoài, cũng như cho kết quả sai khi 3 thành phần còn lại của ứng dụng Android tương tác với nhau thông qua URI.

Trong khi đó, PCLeaks [16], một công cụ nhận dạng các ứng dụng đánh cắp thông tin người dùng bằng phương pháp phân tích tĩnh, đánh dấu theo dõi dòng dữ liệu, trọng tâm là xem xét các đường rò rỉ ICC trong nội tại một ứng dụng đơn nhất. Nó cũng gặp phải vấn đề khi không hoạt động chính xác trên Content Provider giống như trường hợp của Epicc, do PCLeaks chỉ đề xuất phương pháp phân tích trên Activity, Service và Broadcast Receiver.

LeakMiner [27] phát hiện rò rỉ thông tin trên các ứng dụng Android bằng phương pháp phân tích "taint" tĩnh. Thiết kế của LeakMiner tập trung vào việc xác định các ứng dụng mang nguy cơ rò rỉ thông tin và loại bỏ ngay ứng dụng đó ở bên phía kho market ứng dụng trước khi nó được phân phối tới người dùng. Nó áp dụng một thuật toán xây dựng biểu đồ lời gọi (call graph) để phân tích, do Android là 1 hệ thống hướng sự kiện. LeakMiner đưa ra cách tiếp cận gồm 3 bước: đầu tiên, tập tin cài đặt APK của ứng dụng được chuyển thành bytecote Java để phân tích trực tiếp trên bytecote Java, đồng thời các metadata của ứng dụng được trích xuất từ file manifest của ứng dụng Android; bước tiếp theo, LeakMiner xác định các thông tin nhạy cảm dựa vào các metadata đã được trích xuất; ở bước cuối cùng các thông tin đánh dấu được lan truyền qua các lời gọi callbacks, để xác định các đường dữ liệu dẫn đến rò rỉ thông tin. Tuy nhiên, LeakMiner chỉ có khả năng phân tích đơn ứng dụng và đưa ra kết quả không chính xác trong những trường hợp đường lan truyền dữ liệu dài.

Công cụ Amandroid [25] là một áp dụng của phương pháp phân tích tĩnh để phát hiện rò rỉ thông tin nhạy cảm bằng cách theo vết các luồng điều khiển và dòng dữ liệu ICC liên thành phần giữa nhiều component khác nhau trong cùng một ứng dụng hoặc cơ chế liên ứng dụng - inter-app communication (IAC) giữa các ứng dụng khác nhau, tức là công cụ này có khả năng phân tích đơn và đa ứng dụng. Tuy nhiên, Amandroid bị giới hạn trong khả năng kiểm soát ngoại lệ của chương trình, nếu một chương trình có lỗ hổng rò rỉ thông tin do 1 exception phát sinh thì Amandroid không thể phát hiện được lỗ hổng này. Ngoài ra, Amandroid không thể phát hiện rò rỉ trên Content Provider, một trong bốn thành phần của một ứng dụng Android; nó cũng không hiệu quả đối với một số phương thức ICC phức tạp như bindService() và startActivityForResult().

Nhóm nghiên cứu IccTA [15]: công bố công cụ IccTA sử dụng phương pháp phân tích tĩnh dựa trên phân tích đánh dấu dòng dữ liệu ICC để phát hiện nguy cơ rò rỉ thông tin trên đơn hoặc đa ứng dụng Android. Công cụ này dựa trên sự kết hợp giữa Epicc [19] và FlowDroid [3] được tùy biến lại, cả Epicc và FlowDroid được xây dựng dựa trên framework Soot [13] dùng để phân tích các chương trình Java. Soot sử dụng plugin Dexpler [7] để chuyển đội mã byte code Android Dalvik sang một dạng file tên là Jimple được định nghĩa nội tại trong nó, sau đó xây dựng chính xác biểu đồ các lời gọi (call graphs), phân tích đánh dấu theo dõi các dòng dữ liệu trong từng thành phần của ứng dụng Android để xác định những vị trí rò rỉ dữ liệu. Phiên bản IccTA hiện tại gặp hạn chế trước các lỗ hổng rò rỉ liên quan đến Content Provider do không xử lý các dòng dữ liệu liên quan đến thành phần này. Ngoài ra, công cụ IccTA chỉ có khả năng phát hiện rò rỉ liên ứng dụng IAC giữa 2 chương trình Android.

DidFail [11] hiện thực phương pháp phân tích tĩnh dòng dữ liệu được đánh dấu để phát hiện mối nguy cơ rò rỉ dữ liệu nhạy cảm trong một ứng dụng độc lập hay trong môi trường nhiều ứng dụng Android tương tác với nhau, tức là có khả năng phát hiện nguy cơ từ sự thỏa hiệp giữa các ứng dụng này với nhau bên cạnh tính năng nhận dạng lỗ hổng trong một ứng dụng đơn nhất. DidFail được xây dựng dựa trên 2 công cụ phân tích tĩnh đã có là FlowDroid [3] – dùng để tìm các dòng thông tin dữ liệu trong nội tại một thành phần của ứng dụng, và Epicc [19] – để xác định các mối liên kết giữa các intent. Tuy nhiên, hạn chế của DidFail cũng bao gồm những điểm yếu của Epicc và FlowDroid; ngoài ra, DidFail không xem xét đến khía cạnh quyền hạn permission khi so khớp các intent để tìm ra các đường liên kết thất thoát dữ liệu gây ra một số kết quả thiếu chính xác. Vì ở thời điểm hiện tại, DidFail chỉ tập trung vào việc phân tích trên Activity, nên nó không thể phát hiện được bất cứ lỗ hổng rò rỉ thông tin nào trong các thành phần còn lại là: Service, Broadcast Receiver, Content Provider và ngay cả loại Pending Intent của chương trình Android.

Tình hình nghiên cứu trong nước:

Tại Việt Nam, nhóm nghiên cứu của TS. Phạm Văn Hậu, Nguyễn Lê Thanh Trúc, Đại học Quốc tế, ĐHQG Tp. HCM, có đề tài "Detecting privilege escalation

attacks on android device" (2012) [10], nghiên cứu về phương pháp phân tích tĩnh trong phát hiện loại tấn công dựa vào các đặc quyền được cấp phát cho các ứng dụng Android có ý định đánh cắp dữ liệu cá nhân người dùng trên thiết bị di động chạy nền tảng Android. Tuy nhiên, phương pháp này tập trung vào việc phân tích hệ thống quyền hạn (permission) được cấp phát cho ứng dụng.

Ngoài ra, thị trường các công cụ phòng chống bảo vệ người dùng trước các loại virus, spyware, phần mềm gián điệp cũng được phát triển, bằng chứng là BKAV ra mắt với sản phẩm Bkav Mobile Security, có chức năng giám sát truy cập, quét và cảnh báo các phần mềm có thể truy cập vào các thông tin riêng tư trên máy như danh bạ, tin nhắn, vị trí máy... Đặc biệt là phát hiện và loại bỏ các phần mềm nghe lén trên điện thoại di động. Nhưng công cụ này chỉ có khả năng hoạt động một khi các ứng dụng độc hại được cài vào thiết bị người dùng, chỉ hoạt động đơn nhất trên từng ứng dụng, và dựa trên permission đã được cấp phát cho từng ứng dụng để đánh giá rủi ro bị nghe lén, hay chính xác hơn nó có khả năng gây nhầm lẫn vì không phải ứng dụng nào có permission nhất định nào đó cũng là nguy hại nếu nó không thực hiện hành vi đánh cắp thông tin ra bên ngoài. Tóm lại, công cụ Bkav Mobile Security là 1 phần mềm hỗ trợ người dùng xem những ứng dụng nào có thể làm gì với những nhóm permission mà nó được cấp phát, khó có thể mang lại độ chính xác cao trong dự đoán các mối nguy hại rò rỉ thông tin ra bên ngoài.

5. **Tính khoa học và tính mới của đề tài**:

Ở thời điểm này, một số công cụ phân tích tĩnh đã được phát triển nhắm đến mục đích xác định được các chương trình ứng dụng rò rỉ thông tin trên các thiết bị Android. Một số công cụ phân tích tĩnh như CHEX [17], FlowDroid [3], Epicc [19] chỉ tập trung phân tích đơn ứng dụng, đặt ứng dụng trong tách biệt để thực hiện việc phân tích dòng dữ liệu, dẫn đến không thể phát hiện được những lỗ hổng rò rỉ do sự bắt tay phối hợp giữa nhiều ứng dụng. Bên cạnh đó, một số ít công cụ phân tích tĩnh dòng dữ liệu có thể phân tích đơn và đa ứng dụng đồng thời như Amandroid [25] , IccTA [15] [6], DidFail [11], để xác định các lỗ hổng dựa trên mối liên kết giữa các thành phần khác nhau của ứng dụng riêng biệt - theo cơ chế liên thành phần – ICC, và sự tương tác giữa nhiều ứng dụng theo cơ chế liên ứng dụng – IAC (hay còn gọi là cơ chế ICC không cùng trong một ứng dụng). Những công cụ này cũng đã bước đầu đáp ứng được một phần nào nhu cầu đặt ra như cung cấp tính năng nhận dạng các ứng dụng độc hại có chứa các tác nhân gây nguy hiểm đến tính an toàn bảo mật thông tin người dùng, nhưng vẫn có những sai sót nhất định và gặp phải một số giới hạn trong việc chưa thể thực hiện hết hoàn toàn các phân tích trên 4 thành phần cơ bản của một ứng dụng Android.

Do các công cụ này chỉ mới tập trung vào phân tích các luồng sự kiện, dòng dữ liệu liên quan đến một số thành phần nhất định như các Activity, Intent trong cấu trúc một ứng dụng chạy nền tảng Android, chưa giải quyết được việc nhận dạng trên Pending Intent, Service, Content provider, hay Broadcast Receiver, ví dụ như trường hợp của DidFail.

Chính vì vậy, đề tài này sẽ tập trung vào việc tăng cường độ chính xác trong nhận dạng phát hiện các mối nguy rò rỉ thông tin cá nhân trong các ứng dụng Android dựa trên các mô hình, công cụ phân tích tĩnh đã có ngoài thực tế:

- Đề xuất mô hình cải tiến, kết hợp những ưu điểm của các công cụ nhận dạng các mối nguy cơ rò rỉ thông tin bằng ***phương pháp phân tích tĩnh với kỹ thuật đánh dấu dòng dữ liệu*** trong các ứng dụng Android nhằm tăng cường độ chính xác, giảm thiểu việc xác định nhầm lẫn của các công cụ phân tích tĩnh hiện thời, có khả năng phát hiện rò rỉ thông tin qua ***2 cơ chế liên thành phần ICC và liên ứng dụng IAC*** .
- Tập trung vào việc xây dựng cách thức ***phân tích tĩnh đa ứng dụng - liên thành phần trên 4 module chính trong các ứng dụng Android: Activity, Service, Content provider, Broadcast receiver.***

6. Mục tiêu, đối tượng và phạm vi NC đề tài

Mục tiêu:

- Tìm hiểu, phân tích các công cụ phân tích tĩnh trong việc phát hiện rò rỉ thông tin: tổng hợp, cài đặt, thử nghiệm các công cụ hiện có nhằm nắm bắt được các khả năng hỗ trợ và những điểm yếu của chúng.
- Nghiên cứu cơ chế lan truyền thông tin dạng callbacks, các cơ chế lưu chuyển thông tin dữ liệu giữa các thành phần của ứng dụng Android qua ICC và IAC, tìm hiểu kỹ thuật phân tích dòng dữ liệu được đánh dấu (tainting flow analysis).
- Nghiên cứu mô hình phân tích ngược ứng dụng nền tảng Android để nhận dạng các chương trình malware đánh cắp dữ liệu, đưa ra các đề xuất nhằm cải thiện độ chính xác, giải quyết những hạn chế, nhược điểm của kỹ thuật phân tích tĩnh được sử dụng trong các công cụ nhận dạng lỗ hổng thất thoát dữ liệu state-of-the-art hiện nay.
- Xây dựng bộ nhận dạng có khả năng phát hiện lỗ hổng rò rỉ trên các thành phần của một hay nhiều chương trình Android, dựa trên các công cụ phân tích tĩnh có sẵn hiện nay, nhằm tìm kiếm sự tồn tại của các thông tin nhạy cảm và liệu thông tin nhạy cảm có được sử dụng trong callbacks hay không.

Đối tượng, phạm vi nghiên cứu:

Các chương trình ứng dụng trên Android, các bộ công cụ phân tích tĩnh có thể phân tích phát hiện trên các ứng dụng Android.

7. Nội dung, phương pháp dự định NC

Nội dung 1:

Mục tiêu:

- Khảo sát, tìm hiểu các nguy cơ rò rỉ thông tin trong các ứng dụng Android, các vấn đề trở ngại thách thức trong bài toán phân tích phát hiện rò rỉ thông tin nếu cài đặt ứng dụng độc hại.

Kết quả:

- Tài liệu về mối nguy hiểm rò rỉ thông tin trên các ứng dụng Android, cách thức hoạt động của chúng.
- Bộ dữ liệu ứng dụng chuẩn dùng cho việc đánh giá kết quả bài toán phát hiện ứng dụng rò rỉ thông tin.

Nội dung 2:

Mục tiêu:

- Tìm hiểu kỹ thuật xác định source và sink dùng trong phân tích dòng dữ liệu ở ứng dụng Android.
- Khảo sát, thu thập, tìm hiểu các phương pháp phân tích tĩnh, kỹ thuật phân tích đánh dấu dòng dữ liệu, cũng như các công cụ phân tích tĩnh đã có trong việc giúp nhận dạng phát hiện sự rò rỉ thông tin trong các ứng dụng Android.

Kết quả:

- Các phương pháp phân tích tĩnh dùng trong các ứng dụng phân tích bảo mật giúp nhận dạng mối nguy hiểm rò rỉ thông tin trên các ứng dụng Android.
- Phương pháp xác định source và sink trong các ứng dụng Android.
- Tài liệu tổng hợp đánh giá về tính năng, hiệu quả, hạn chế của các công cụ nhận dạng rò rỉ thông tin bằng phương pháp phân tích tĩnh, tập trung vào kỹ thuật phân tích đánh dấu dòng dữ liệu.
- Mã nguồn của các bộ công cụ phân tích áp dụng các phương pháp state-of-the-art hiện nay đã được công bố.

Nội dung 3:

Mục tiêu:

- Tìm hiểu cơ chế rò rỉ thông tin thông qua phương thức tương tác liên thành phần trong cùng một ứng dụng (ICC) hay giữa nhiều ứng dụng Android với nhau (IAC).
- Nghiên cứu phát triển kỹ thuật nhận dạng một cách chính xác và toàn diện hơn các nguy cơ an ninh tiềm ẩn rò rỉ dữ liệu cá nhân ***trên 4 thành phần cơ bản của một ứng dụng Android*** mà các công cụ đã được công bố chưa giải quyết được ở một số trường hợp.
- Xem xét các hướng tiếp cận và khả năng kết hợp hoặc tối ưu của các công cụ state-of-the-art hiện nay là IccTA và DidFail trong việc phân tích dòng dữ liệu phát hiện các rò rỉ xảy ra dưới dạng tương tác liên thành phần ICC và liên ứng dụng IAC.

Kết quả:

- Phương pháp xác định các liên kết tương tác dòng dữ liệu giữa các thành phần khác nhau của những ứng dụng Android.

- Mô hình phát hiện nguy cơ rò rỉ thông tin bằng ***phương pháp phân tích tĩnh đánh dấu dòng dữ liệu có thể hoạt động trên 4 module căn bản:***
  - ***Activity***
  - ***Service***
  - ***Broadcast Receiver***
  - ***Content Provider***
- Các hướng cải thiện và phương pháp kết hợp tối ưu hóa phương pháp phân tích tĩnh phát hiện rò rỉ dạng ICC và IAC dựa trên DidFail, IccTA cũng như một số công cụ khác được công bố.

Nội dung 4:

Mục tiêu:

- Đề xuất phương pháp cải tiến, tối ưu ***phương pháp phân tích tĩnh dòng dữ liệu bằng cơ chế đánh dấu "tainting"*** có khả năng phân tích lỗ hổng rò rỉ đến từ sự tương tác giữa nhiều ứng dụng ***thông qua cơ chế ICC và IAC*** dựa trên mô hình phân tích 2 giai đoạn của DidFail có sự bổ sung của IccTA**.**
- Tăng cường tính chính xác trong mô hình nhận diện rò rỉ thông tin khi phân tích 4 module của ứng dụng, cải thiện được hiệu năng và các sai sót khi phân tích đa ứng dụng.
- Hiện thực cách tiếp cận phân tích tĩnh dạng đánh dấu dòng dữ liệu theo mô hình được đề xuất.

Kết quả:

- Phương pháp phân tích liên thành phần, liên ứng dụng với tỉ lệ sai sót thấp và tối thiểu hóa tốc độ phân tích trên bộ dữ liệu ứng dụng Android độc hại.
- Công cụ nhận dạng rò rỉ thông tin có thể hoạt động tốt và có độ chính xác cao trong việc phát hiện các ứng dụng khai thác bất hợp pháp, rò rỉ thông tin cá nhân người dùng.
- Đưa ra các nội dung cần cải tiến sau khi hoàn thành việc kiểm thử lại kết quả áp dụng phương pháp phân tích tĩnh dùng để xác định các ứng dụng nguy hại tiềm tàng, mang rủi ro đánh cắp thông tin nhạy cảm trong các ứng dụng Android.

8. Kế hoạch bố trí thời gian NC

Nội dung	Thời gian
- Khảo sát cơ chế phát hiện rò rỉ thông tin trên thiết bị Android bằng phương pháp phân tích tĩnh. - Tìm hiểu các đặc tính, hành vi của các ứng	

dụng Android độc hại gây rò rỉ đã được phát hiện. - Tìm hiểu đánh giá các công cụ phân tích tĩnh đã công bố, các kết quả nghiên cứu hiện tại, tập trung vào kỹ thuật đánh dấu dòng dữ liệu trong phương pháp phân tích tĩnh. - Tập hợp các ứng dụng Android độc hại dùng để kiểm thử kết quả của đề tài.	3.2015 – 4.2015
- Nghiên cứu cách tổ chức, xử lý của DidFail và IccTA, cấu trúc thành phần cũng như source code của công cụ này. - Đề xuất mô hình, cơ chế tăng cường, cải tiến độ chính xác trong bài toán phát hiện rò rỉ thông tin trong các ứng dụng Android trên cơ sở 2 công cụ DidFail và IccTA. - Đưa ra phương pháp phân tích tĩnh phát hiện rò rỉ thông tin thông qua kỹ thuật đánh dấu theo dõi dòng dữ liệu giữa các thành phần cơ bản của một hay nhiều ứng dụng Android.	5.2015 – 7.2015
- Đánh giá kết quả phương pháp phân tích tĩnh đã đề xuất vào bài toán phát hiện rò rỉ thông tin ở các ứng dụng Android. - Tổng hợp, viết luận văn	8.2015

9.Tài liệu tham khảo

[1] Android (operating system).

URL: http://en.wikipedia.org/wiki/Android_(operating_system), 22.02, 2015.

[2] AppBrain. "Number of Android applications". AppBrain Stats.

URL: http://www.appbrain.com/stats/number-of-android-apps , 26.11, 2014.

[3] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau and P. McDaniel, (2014), "FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps", *Proceedings of the 35th annual ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI 2014)*.

[4] Steven Arzt, Siegfried Rasthofer, and Eric Bodden, (2013), SuSi: A Tool for the Fully Automated Classification and Categorization of Android Sources and Sinks. Technical Report TUD-CS-2013-0114, EC SPRIDE, May 2013. 2.2.1.

[5] Steven Arzt, Siegfried Rasthofer, Eric Bodden, (2013), Highly Precise Taint Analysis for Android Applications, Technical Report Nr. TUD- CS- 2013 – 0113.

[6] Alexandre Bartel, (2014), Security Analysis of Permission-Based Systems using Static Analysis: An Application to the Android Stack, Doctoral thesis, University of Luxembourg, Luxembourg.

[7] Alexandre Bartel, Jacques Klein, Yves Le Traon, Martin Monperrus, (2012), "Dexpler: converting Android Dalvik bytecode to Jimple for static analysis with Soot", *Proceeding SOAP '12 Proceedings of the ACM SIGPLAN International Workshop on State of the Art in Java Program analysis*, Beijing, June 14th 2012 (SOAP@PLDI 2012), pp. 27-38.

[8] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol Sheth, (2010). "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones*", OSDI. Vol. 10. 2010, pp. 255–270.*

[9] Huang, Jianjun and Zhang, Xiangyu and Tan, Lin and Wang, Peng and Liang, Bin, (2014), "AsDroid: Detecting Stealthy Behaviors in Android Applications by User Interface and Program Behavior Contradiction", *Proceedings of the 36th International Conference on Software Engineering.*

[10] Pham Van Hau, Nguyen Le Thanh Truc (2012), "Detecting privilege escalation attacks on android device", International University –VNU HCM.

[11] William Klieber, Lori Flynn, Amar Bhosale, Limin Jia, Lujo Bauer, (2014), "Android Taint Flow Analysis for App Sets", *3rd ACM SIGPLAN International Workshop on the State Of the Art in Program Analysis (SOAP 2014).*

[12] Kaspersky Security Bulletin 2013. Malware evolution. URL:

http://media.kaspersky.com/pdf/KSB_2013_EN.pdf . (2014)

[13] Patrick Lam, Eric Bodden, Ondrej Lhoták, and Laurie Hendren, (2011), "The Soot framework for Java program analysis: a retrospective", *Cetus Users and Compiler Infastructure Workshop (CETUS 2011).*

[14] Li Li, Alexandre Bartel, Jacques Klein, Yves Le Trao, Steven Arzt, Siegfried Rasthofer, Eric Bodden, Damien Octeau, Patrick McDanie, (2013), I know what leaked in your pocket: uncovering privacy leaks on Android Apps with Static Taint Analysis, ISBN 978-2-87971-129-4.

[15] Li Li, Alexandre Bartel, Tegawend´e F. Bissyand´e, Jacques Klein, Yves Le Trao, Steven Arzt, Siegfried Rasthofer, Eric Bodden, Damien Octeau, Patrick McDaniel, (2015), "IccTA: Detecting Inter-Component Privacy Leaks in Android Apps", *The 37th International Conference on Software Engineering (ICSE 2015).*

[16] L. Li, A. Bartel, J. Klein and Y. Le Traon, (2014), "Automatically Exploiting Potential Component Leaks in Android Applications", *The 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2014).*

[17] Long Lu, Zhichun Li, Zhenyu Wu, Wenke Lee, Guofei Jiang, (2012), "CHEX: Statically vetting Android apps for component hijacking vulnerabilities," *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, Raleigh, North Carolina, USA: ACM, 2012, pp. 229–240.

[18] Mann, Christopher and Starostin, Artem, (2012), "A Framework for Static Detection of Privacy Leaks in Android Applications", *Proceedings of the 27th Annual ACM Symposium on Applied Computing.*

[19] Damien Octeau, Patrick McDaniel, Somesh Jha, Alexandre Bartel, Eric Bodden, Jacques Klein, and Yves Le Traon, (2013), "Effective inter-component communication mapping in android with Epicc: An essential step towards holistic security analysis", *Proceedings of the 22nd USENIX Security Symposium (USENIX Security 13)*, pp. 543-558, Washington, D.C., 2013.

[20] PCWorld report: Malware-infected Android apps spike in the Google Play store, http://www.pcworld.com/article/2099421/report-malwareinfected-android-apps-spike-in-the-google-play-store.html - 19 February, 2014.

[21] Siegfried Rasthofer, Steven Arzt, and Eric Bodden, (2014), "A Machine-learning Approach for Classifying and Categorizing Android Sources and Sinks", *Proceeding NDSS*, 2014. 4.1.2

[22] Securelist Analysis Report, IT Threat Evolution: Q2. URL: https://www.securelist.com/en/analysis/204792299/IT_Threat_Evolution_Q2_2013, (2014).

[23] Dragos Sbirlea, Michael G. Burke, Salvatore Guarnieri, Marco Pistoia, Vivek Sarka, (2013), Automatic Detection of Inter-application Permission Leaks in Android Applications, Department of Computer Science, Rice University & IBM Watson Research Center.

[24] Shen, Feng and Vishnubhotla, Namita and Todarka, Chirag and Arora, Mohit and Dhandapani, Babu and Lehner, Eric John and Ko, Steven Y. and Ziarek, Lukasz, (2014), "Information Flows As a Permission Mechanism", *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*.

[25] Fengguo Wei, Sankardas Roy, Xinming Ou, Robby, (2014). Amandroid: A Precise and General Inter-component Data Flow Analysis Framework for Security Vetting of Android Apps. Dept. of Computing and Information Sciences, Kansas State University, Manhattan, Kansas 66506.

[26] Lei Wu, Michael Grace, Yajin Zhou, Chiachih Wu, and Xuxian Jiang, (2013), "The impact of vendor customizations on android security", *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security,* pp. 623–634.

[27] Zhemin Yang and Min Yang, (2012), "LeakMiner: Detect Information Leakage on Android with Static Taint Analysis", *Software Engineering (WCSE), 2012 Third World Congress on.*

[28] Yajin Zhou and Xuxian Jiang, (2012), "Dissecting android malware: Characterization and evolution", *Security and Privacy (SP), 2012 IEEE Symposium*, pp. 95–109.

TP. HCM, ngày 13 tháng 4 năm 2015

NGƯỜI HƯỚNG DẪN	**HỌC VIÊN KÝ TÊN**
(*Họ tên và chữ ký*)	(*Họ tên và chữ ký*)
……Phạm Văn Hậu……..	……Phan Thế Duy……..

THUYẾT MINH SỬA ĐỀ CƯƠNG LUẬN VĂN

Tên giáo viên góp ý 1: TS. Lê Trung Quân

Tên giáo viên góp ý 2: TS. Nguyễn Minh Sơn

Ngày bảo vệ đề cương: 04.04.2015

Nội dung trước sửa	Nội dung sau khi sửa
Phần 4.3 - trang 7-8 Tình hình nghiên cứu trong nước: Tại Việt Nam, nhóm nghiên cứu của TS. Phạm Văn Hậu, Nguyễn Lê Thanh Trúc, Đại học Quốc tế, ĐHQG Tp. HCM, có đề tài "Detecting privilege escalation attacks on android device" (2012) [10], nghiên cứu về phương pháp phân tích tĩnh trong phát hiện loại tấn công dựa vào các đặc quyền được cấp phát cho các ứng dụng Android có ý định đánh cắp dữ liệu cá nhân người dùng trên thiết bị di động chạy nền tảng Android. Tuy nhiên, phương pháp này tập trung vào việc phân tích hệ thống quyền hạn (permission) được cấp phát cho ứng dụng.	***Phần 4.3 - trang 7-8*** Tình hình nghiên cứu trong nước: Tại Việt Nam, nhóm nghiên cứu của TS. Phạm Văn Hậu, Nguyễn Lê Thanh Trúc, Đại học Quốc tế, ĐHQG Tp. HCM, có đề tài "Detecting privilege escalation attacks on android device" (2012) [10], nghiên cứu về phương pháp phân tích tĩnh trong phát hiện loại tấn công dựa vào các đặc quyền được cấp phát cho các ứng dụng Android có ý định đánh cắp dữ liệu cá nhân người dùng trên thiết bị di động chạy nền tảng Android. Tuy nhiên, phương pháp này tập trung vào việc phân tích hệ thống quyền hạn (permission) được cấp phát cho ứng dụng. Ngoài ra, thị trường các công cụ phòng chống bảo vệ người dùng trước các loại virus, spyware, phần mềm gián điệp cũng được phát triển, bằng chứng là BKAV ra mắt với sản phẩm Bkav Mobile Security, có chức năng giám sát truy cập, quét và cảnh báo các phần mềm có thể truy cập vào các thông tin riêng tư trên máy như danh bạ, tin nhắn, vị trí máy... Đặc biệt là phát hiện và loại bỏ các phần mềm nghe lén trên điện thoại di động. Nhưng công cụ này chỉ có khả năng hoạt động một khi các ứng dụng độc hại được cài vào thiết bị người dùng, chỉ hoạt động đơn nhất trên từng ứng dụng, và dựa trên permission đã được cấp

	phát cho từng ứng dụng để đánh giá rủi ro bị nghe lén, hay chính xác hơn nó có khả năng gây nhầm lẫn vì không phải ứng dụng nào có permission nhất định nào đó cũng là nguy hại nếu nó không thực hiện hành vi đánh cắp thông tin ra bên ngoài. Tóm lại, công cụ Bkav Mobile Security là 1 phần mềm hỗ trợ người dùng xem những ứng dụng nào có thể làm gì với những nhóm permission mà nó được cấp phát, khó có thể mang lại độ chính xác cao trong dự đoán các mối nguy hại rò rỉ thông tin ra bên ngoài.
Phần 7 trang 11 Nội dung 4: ▪ Mục tiêu: - Hiện thực cách tiếp cận phân tích tĩnh dạng đánh dấu dòng dữ liệu theo mô hình được đề xuất. ▪ Kết quả: - Công cụ nhận dạng rò rỉ thông tin có thể hoạt động tốt và có độ chính xác cao trong việc phát hiện các ứng dụng khai thác bất hợp pháp, rò rỉ	***Phần 7 trang 11*** Nội dung 4: ▪ Mục tiêu: - Đề xuất phương pháp cải tiến, tối ưu phương pháp phân tích tĩnh dòng dữ liệu bằng cơ chế đánh dấu "tainting" có khả năng phân tích lỗ hổng rò rỉ đến từ sự tương tác giữa nhiều ứng dụng thông qua cơ chế ICC và IAC dựa trên mô hình phân tích 2 giai đoạn của DidFail có sự bổ sung của

thông tin cá nhân người dùng. - Đưa ra các nội dung cần cải tiến sau khi hoàn thành việc kiểm thử lại kết quả áp dụng phương pháp phân tích tĩnh dùng để xác định các ứng dụng nguy hại tiềm tàng, mang rủi ro đánh cắp thông tin nhạy cảm trong các ứng dụng Android.	IccTA. - Tăng cường tính chính xác trong mô hình nhận diện rò rỉ thông tin khi phân tích 4 module của ứng dụng, cải thiện được hiệu năng và các sai sót khi phân tích đa ứng dụng. - Hiện thực cách tiếp cận phân tích tĩnh dạng đánh dấu dòng dữ liệu theo mô hình được đề xuất. ▪ Kết quả: - Phương pháp phân tích liên thành phần, liên ứng dụng với tỉ lệ sai sót thấp và tối thiểu hóa tốc độ phân tích trên bộ dữ liệu ứng dụng Android độc hại. - Công cụ nhận dạng rò rỉ thông tin có thể hoạt động tốt và có độ chính xác cao trong việc phát hiện các ứng dụng khai thác bất hợp pháp, rò rỉ thông tin cá nhân người dùng. - Đưa ra các nội dung cần cải tiến sau khi hoàn thành việc kiểm thử lại kết quả áp dụng phương pháp phân tích tĩnh dùng để xác định các ứng dụng nguy hại tiềm tàng, mang rủi ro đánh cắp thông tin nhạy cảm trong các ứng dụng Android.
Phần 8 – trang 12 5.2015 – 7. 2015 - Đề xuất mô hình, cơ chế tăng cường, cải tiến độ chính xác trong bài toán phát hiện rò rỉ thông tin trong các ứng dụng Android bằng phương pháp phân tích tĩnh thông qua việc đánh dấu theo dõi dòng dữ liệu giữa các thành phần của một hay nhiều ứng dụng.	***Phần 8 – trang 12*** 5.2015 – 7. 2015 - Nghiên cứu cách tổ chức, xử lý của DidFail và IccTA, cấu trúc thành phần cũng như source code của công cụ này. - Đề xuất mô hình, cơ chế tăng cường, cải tiến độ chính xác trong bài toán phát hiện rò rỉ thông tin trong các ứng dụng Android trên cơ sở 2 công cụ DidFail và IccTA. - Đưa ra phương pháp phân tích tĩnh phát hiện rò rỉ thông tin thông qua kỹ thuật

	đánh dấu theo dõi dòng dữ liệu giữa các thành phần cơ bản của một hay nhiều ứng dụng Android.
Phần 8 – trang 12 8. 2015 - Viết luận văn	***Phần 8 – trang 12*** 8. 2015 - Đánh giá kết quả phương pháp phân tích tĩnh đã đề xuất vào bài toán phát hiện rò rỉ thông tin ở các ứng dụng Android. - Tổng hợp, viết luận văn