ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN NAM CHUNG

THEO DÕI CÁC NGUỒN DỮ LIỆU NHẠY CẢM TRÊN CÁC THIẾT BỊ DI ĐỘNG CHẠY HỆ ĐIỀU HÀNH ANDROID

Ngành: Công Nghệ Thông Tin
Chuyên ngành: Kỹ Thuật Phần Mềm
Mã số: 60480103

TÓM TẮT LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. Nguyễn Đại Thọ

Hà Nội – 2017

MỞ ĐẦU

Các thiết bị di động hiện nay có rất nhiều các cảm biến và dịch vụ bên trong không gian riêng tư của người dùng. Vậy nên chúng có khả năng giám sát rất nhiều khía cạnh nhạy cảm trong cuộc sống người sử dụng (ví dụ: vị trí, sức khỏe hay giao thiệp). Người dùng thường giao dịch thanh toán trực tuyến, trong nhiều tình huống họ không đánh giá đầy đủ về bản chất cũng như mức độ thông tin sẽ được khai thắc bởi các ứng dụng bên thứ 3. Sự gia tăng nhanh chóng của các thiết bị di động khiến chúng trở thành không thể thiếu với cuộc sống của nhiều người. Thật vậy, các thiết bị cung cấp cho người dùng một loạt các dịch vụ thiết yếu (ví dụ: định vị, liên lạc và kết nối Internet) cùng các chức năng hữu ích (ví dụ: nghe nhạc, nhiếp ảnh, xem truyền hình, mua sắm trực tuyến). Để đáp ứng các dịch vụ này, các thiết bị di động hiện đại đã được trang bị rất nhiều cảm biến, khả năng thu thập thông tin và môi trường xung quanh về người dùng vô cùng phong phú. Người dùng và nhà phát triển cũng đã chấp nhận các công nghệ thu thập thông tin để đổi lấy rất nhiều tính năng công nghệ cao mà chúng mang lại. Trên thực tế, rất nhiều ứng dụng mời chào các dịch vụ hoàn toàn miễn phí với sự đánh đổi ẩn dấu về việc thu thập dữ liệu mà hầu hết sẽ được dùng cho việc quảng cáo.

Có rất nhiều nghiên cứu đã chỉ ra, người dùng thường hành động mà không hiểu mức độ thông tin của họ có thể được trích ra từ các thông tin được thu thập này. Ngay cả khi người dùng ý thức việc thu thập dữ liệu, họ không thể hoàn toàn nhận ra những hàm ý không trực quan về việc chia sẻ dữ liệu của họ. Các nhà nghiên cứu đã chỉ ra các cảm biến trên các thiết bị có thể được dùng bí mật để bắt các phím bấm, chạm trên điện thoại để dò tìm vị trí, ghi âm giọng nói hay các hoạt động thường ngày của người dùng. Phần lớn các thiết bị được chạy trên hệ điều hành Android và iOS. Cùng với sự bùng nổ của thị trường hệ điều hành nguồn mở thì Android đã trở thành hệ điều hành phổ biến nhất hiện nay với kho ứng dụng lên đến hơn 2,5

triệu ứng dụng. Dữ liệu thu thập được có thể được truy xuất bởi rất nhiều bên và thường không có sự cho phép rõ ràng của người dùng. Thế nên việc giám sát các luồng dữ liệu do các ứng dụng trao đổi với bên ngoài là hết sức cần thiết. Để đảm bảo tính bảo mật, toàn vẹn và khả dụng của thông tin mà thiết bị di động truy cập, lưu trữ và xử lý là một thách thức khó khăn. Nhưng hiện nay việc kiểm soát an ninh vẫn chưa theo kịp với những rủi ro gây ra bởi các thiết bị di động. Các nhà phát triển cũng như các doanh nghiệp vẫn đang nỗ lực đưa ra các giải pháp an ninh thông tin trên các thiết bị động và hiện nay có một số hệ nổi bật về cả phần mềm, phần cứng cũng như tích hợp như: Samsung Knox, BlackBerry Balance, AndroidLeaks, SCANDAL, IccTAp, TaintDroid.

- TaintDroid là một hệ thống có khả năng kiểm tra truy cập thông tin nhạy cảm của người dùng ở mức thời gian thực. Nó kiểm tra luồng thông tin trên các thiết bị di động chạy hệ điều hành Android rất hiệu quả. Tuy còn hạn chế khi không kiểm tra được dưới dạng luồng điều khiển, nhưng đây không phải định hướng mức kiến trúc khi ban đầu khi xây dựng. TaintDroid có thể chạy trên các thiết bị chạy hệ điều hành Android từ phiên bản 2.1 trở đi. Nó hỗ trợ kiểm tra dấu vết (taint) thông tin cá nhân nhạy cảm của người dùng bị các ứng dụng truy cập. Mục tiêu chính của hệ thống là rò tìm khi nào dữ liệu nhạy cảm bị gửi đi từ những ứng dụng không tin cậy. TaintDroid hiện tại có thể kiểm tra nhiều loại thông tin như: vị trí, số điện thoại, máy ảnh, số IMEI, lịch sử trình duyệt. Chính vì các ưu điểm nổi bật của nó so với các hệ thống khác về theo dõi truy cập thông tin nhạy cảm mà nó được chọn làm đề tài nghiên cứu và cải tiến trong khuôn khổ luận văn. Tuy hệ thống đã có khả năng kiểm soát và cảnh báo truy cập trái phép các nhóm thông tin kể trên, nhưng chỉ cảnh báo chung ở mức loại thông tin theo taint. Việc này chỉ hỗ trợ người dùng kiểm soát chung nhất việc truy cập thông tin nhạy cảm mà chưa biết được chính xác những thông tin gì trong đó bị truy cập và gửi đi trái phép.

- Hướng cải tiến trong khuôn khổ luận văn là bổ sung cho hệ thống TaintDroid tính năng cảnh báo người dùng khi ứng dụng truy cập thông tin nhạy cảm trong lịch sử trình duyệt. Mục tiêu chính của cải tiến là sẽ thông báo cho người dùng khi có ứng dụng không tin cậy truy cập đến tên đăng nhập, mật khẩu hay mã số thẻ tín dụng. Với nhu cầu và thói quen của người dùng điện thoại thông minh (Smartphone) hiện nay, việc truy cập Internet là thường xuyên và cùng với đó là việc sử dụng các ứng dụng truy Internet như FaceBook, Twitter, Chrome, … để làm việc và giải trí. Như chúng ta thấy, với cơ chế hoạt động của trình duyệt web cũng như các ứng dụng hoạt động trên nền web thì có rất nhiều thông tin được trình duyệt lưu lại trong quá trình sử dụng. Các thông tin đó được lưu lại thành dữ liệu lịch sử của trình duyệt (browser history**)**, trong đó chứa nhiều thông tin nhạy cảm.

- Cải tiến đã hoàn thành với kết quả đạt được cụ thể và rõ ràng như định hướng đề ra. Hệ thống đã có thể kiểm tra được chính xác các taint chứa thông tin nhạy cảm trong lịch sử trình duyệt bị các ứng dụng không tin cậy truy cập. Việc cải tiến cũng không làm ảnh hưởng đến các luồng xử lý cũng như hiệu năng của hệ thống hiện tại. Cụ thể khi thông tin về tên truy cập, mật khẩu hay mã số thẻ tín dụng bị truy cập, hệ thống sẽ hiện thông báo riêng so với các thông báo sẵn có bằng hình thức thay đổi đèn LED và tần suất nhấp nháy. Việc cải tiến được kiểm chứng trên môi trường thật, thiết bị được sử dụng là điện thoại di động Google Nexus 4. Việc xây dựng, cài đặt cũng như chạy thử đều tuân thủ các bước do nhóm phát triển hệ thống TaintDroid đưa ra. Toàn bộ tài liệu luận văn được bố trí với bố cục các chương mục tóm tắt như sau:

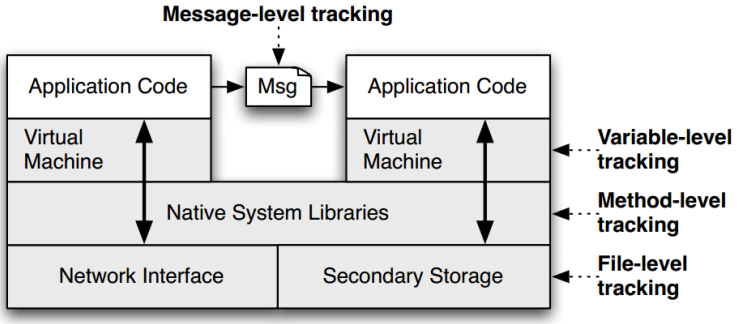
- **Chương 1** - Bảo mật riêng tư trên các thiết bị di động: Chương này giới thiệu các khái niệm, tầm quan trọng cũng như các phương pháp và nguyên lý bảo mật riêng tư. Ngoài ra còn giới thiệu chi tiết về bảo mật cho trình duyệt web và một số hệ thống an ninh cho thiết bị di động tiêu biểu.

- **Chương 2** - Hệ thống TantDroid: Giới thiệu từ tổng quan đến chi tiết các thành phần của TaintDroid cũng như phân tích đánh giá hiệu năng của hệ thống.
- **Chương 3** - Cải tiến theo dõi nguồn dữ liệu nhạy cảm: Chương này miêu tả chi tiết việc cải tiến hệ thống TaintDroid, từ giải pháp chi tiết về kiểm soát truy cập taint lịch sử trình duyệt web đến giải pháp tổng thể để kiểm soát truy cập các loại taint mà hệ thống hiện có.
- **Chương 4** - Kết quả thử nghiệm: Miêu tả chi tiết quá trình thực nghiệm từ môi trường, thiết bị đến việc chạy ứng dụng trên thiết bị thật. Đưa ra các đánh giá về cải tiến theo các tiêu chí cụ thể, hướng thảo luận và định hướng tiếp theo.
- **Kết luận**: Chỉ ra tính ứng dụng của hệ thống cải tiến, các hạn chế còn tồn tại.

Chương 1. Bảo mật riêng tư trên các thiết bị di động

1.1. Bối cảnh chung

Trong xã hội ngày nay, nói đến smartphone là nói đến công nghệ mới. Các smartphone về cơ bản luôn kết nối nhiều dữ liệu cá nhân trong cuộc sống, không chỉ dữ liệu đơn thuần như danh bạ mà nhiều loại dữ liệu kiểu mới như vị trí, sở thích mua sắm trực tuyến. Chúng cũng có khả năng tải và chạy các ứng dụng của bên thứ 3 có kết nối với Internet. Một ví dụ điển hình về ứng dụng hình nền (Wallpaper) gửi thông tin số điện thoại tới nhà phát triển. Khi một ứng dụng chạy thường có thể truy cập bất cứ thông tin nào trên thiết bị không tường minh, ngay cả cách chúng thực hiện việc này. Trong nghiên cứu, nhóm tác giả đã chọn tên Dynamic Taint Analysis, thỉnh thoảng gọi là Taint Tracking. Ý tưởng cơ bản là đánh dấu thẻ taint thông tin nhạy cảm tại nguồn và sau đó theo dấu dữ liệu được chuyển đi qua hệ thống. Trong ngữ cảnh của báo cáo, dữ liệu được đánh dấu truyền qua giao diện mạng của smartphone khi đó có thể biết thông tin có nhạy cảm hay không? Trong quá trình nghiên cứu, nhóm tác giả đã đưa ra mô hình sơ lược hướng tiếp cận đa mức về kiểm tra hiệu năng smartphone như sau:



Hình 1.1. Mô hình kiểm tra hiệu năng điện thoại di động

1.2. Khái niệm bảo mật riêng tư

Chính sách riêng tư là một tuyên bố pháp lý xác định chủ quyền thương mại với dữ liệu cá nhân sẽ được thu thập bởi người sử dụng, bao gồm cả việc dữ liệu được xử lý như thế nào vào ra sao. Từ những năm 1960, Hội đồng châu Âu đã nghiên cứu về sự mở rộng của Internet và tập trung vào sự ảnh hưởng của công nghệ đến các quyền con người. Nghiên cứu về các chính sách để bảo vệ dữ liệu cá nhân. Nó được chúng ta biết đến với tên gọi tiếng Anh "Privacy Policy". Tên gọi này chỉ ra thỏa thuận pháp lý về quyền và sự bảo vệ dữ liệu cá nhân một cách đầy đủ, thỏa thuận này cũng có thể được biết tới dưới các tên sau:

1.3. Tầm quan trọng của bảo mật riêng tư

Chính sách bảo mật là một trong những tài liệu quan trọng nhất của bất kỳ một ứng dụng web hay di động. Nó miêu tả chi tiết các quan điểm và thủ tục trong việc thu thập thông tin từ người sử dụng. Các phần chính của một tài liệu chính sách bảo mật được miêu tả dưới đây.

1.4. Các phương pháp và công cụ đảm bảo tính riêng tư

Chính sách với người dùng ứng dụng đặc biệt quan trọng, tài liệu chính sách bảo mật chỉ ra các phương thức cho việc nhận và thu thập thông tin cá nhân bởi ứng dụng cũng như cách dùng chúng. Dưới đây là các phương pháp và công cụ điển hình trong việc đảm bảo tính riêng tư.

1.5. Các nguyên lý chung đảm bảo tính tiêng tư

Từ những thập niên 1980, có rất nhiều tổ chức hay quốc gia tiến hành xây dựng các nguyên lý cho việc đảm bảo tính riêng tư. Các nước Mỹ, Canada, Australia, Ấn Độ hay liên minh Châu Âu cũng đều đưa ra các nguyên lý chung. Điển hình là tổ chức hợp tác và phát triển kinh tế Châu Âu đã đưa ra 7 nguyên lý sau được cho là khá đầy đủ:

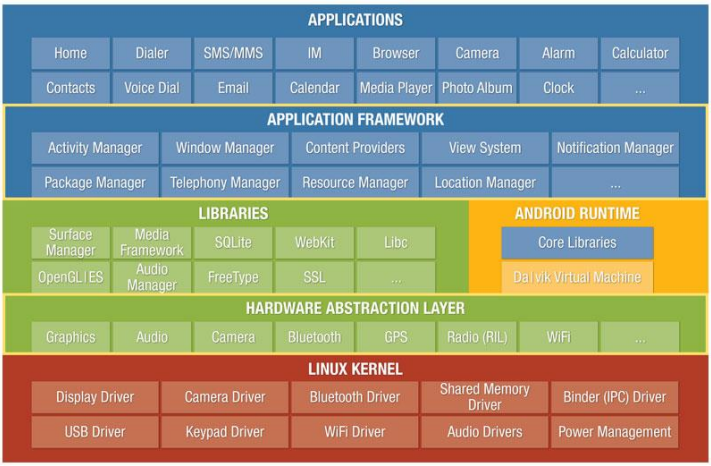
1.6. Bảo mật riêng tư cho trình duyệt web

Trình duyệt web là một ứng dụng phần mềm dùng để nhận, trình bày và duyệt các nguồn thông tin trên Internet. Một tài nguyên thông tin được xác định bởi định danh tài nguyên duy nhất (URI) và có thể là một trang web, hình ảnh, video hay một mẩu dữ liệu. Một trình duyệt web có thể định nghĩa như một ứng dụng phần mềm hay chương trình thiết kế để người dùng có thể truy cập, nhận xem các tài liệu và các tài nguyên khác trên Internet.

Chương 2. Hệ thống TaintDroid

2.1. Giới thiệu Android

Android là một nền tảng di động mã nguồn mở với nhân hệ điều hành Linux, dành cho các thiết bị di động như smartphone và máy tính bảng. Hầu hết các chức năng được phát triển như các ứng dụng chạy bên trên tầng trung gian. Các ứng dụng được viết bằng ngôn ngữ Java hoặc C/C++, chúng được dịch thành các mã tùy biến như định dạng DEX. Mỗi ứng dụng thực thi các thực thể và biên dịch bên trong máy ảo Dalik và dưới đây là miêu tả tổng quát kiến trúc của một hệ điều hành Android.



Hình 2.1. Kiến trúc hệ điều hành Android

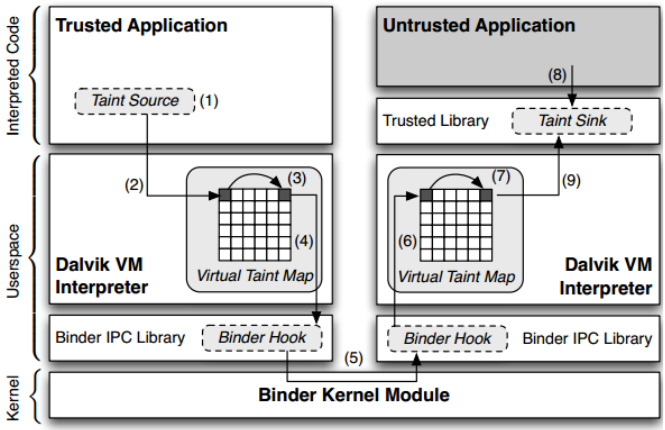
2.2. Giới thiệu TaintDroid

TaintDroid là một hiện thực hóa trong việc theo dõi các dấu vết bên trong hệ điều hành Android. TaintDroid thực hiện việc theo dõi ở mức biến hệ thống bên trong máy ảo biên dịch. Đánh dấu vết được lưu trữ trong một thẻ taint. Khi các ứng dụng thực hiện các phương thức nguyên gốc, các thẻ taint được đưa trở lại. Cuối cùng các thẻ taint được gán cho gói và được tiếp nhận thông qua binder. Hình 2.2 miêu tả kiến trúc của TaintDroid và thông tin được đánh dấu trong

một ứng dụng tin cậy với đủ thông tin. Giao diện đánh dấu gọi một phương thức nguyên gốc thực thi trình biên dịch máy ảo Dalvik. Máy ảo Dalvik phân tán các thẻ taint tương ứng với các quy định về luồng dữ liệu như các ứng dụng tin cậy sử dụng thông tin được đánh dấu.

2.3. Kiến trúc TaintDroid

TaintDroid phiên bản mới nhất được phát triển trên nền hệ điều hành Android 4.3, nên nhìn chung toàn bộ kiến trúc hệ thống tương tự như hệ điều hành Android 4.3.



Hình 2.2. Kiến Trúc TaintDroid Bên Trong Android

2.4. Các chức năng

- Thông báo khi dữ liệu nhạy cảm bị gửi đi: TaintDroid liên tục kiểm tra luồng dữ liệu được gửi đi từ các ứng dụng. Khi dữ liệu nhạy cảm được gửi đi dựa trên việc ánh xạ với bảng taint ảo đã định nghĩa sẵn. Hệ thống sẽ thông báo chi tiết đến người dùng về loại dữ liệu bị gửi đi kèm theo các thông tin như: ứng dụng gửi, thời gian, IP sẽ gửi dữ liệu đến.
- Lưu dấu vết dữ liệu để xác thực quyền truy cập: Khi hệ thống phát hiện taint được gửi đi, dấu vệt của nó sẽ được lưu lại và xác

thực quyệt truy cập thông tin đó. Việc xác định giữa trên quyền của ứng dụng được khai báo từ lúc cài đặt vào hệ điều hành trên thiết bị.

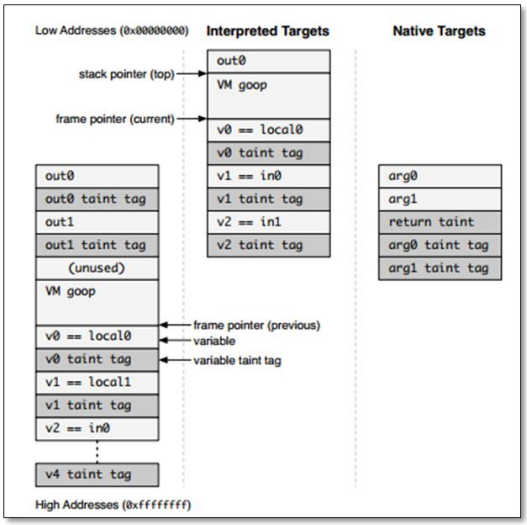
2.5. Nguyên lý hoạt động

- Thông tin được đánh dấu bên trong ứng dụng đã được xác định. Giao diện taint triệu gọi phương thức nguyên gốc từ giao diện trình biên dịch máy ảo Dalvik, lưu trữ các dấu taint bên trong bản đồ taint ảo. Máy ảo Dalvik đưa các thẻ taint tương ứng qua luồng dữ liệu như dành cho các ứng dụng tin cậy sử dụng thông tin taint. Mọi thực thể trình biên dịch sẽ phân tán các thẻ taint dưới dạng mô phỏng.
- Triển khai kiến trúc trên còn có một số thách thức mức hệ thống, bao gồm 1) Lưu trữ các thẻ Taint, 2) Phân tán các mã biên dịch taint, 3) Phân tán các taint mã nguyên gốc, 4) Phân tán taint IPC, 5) Phân tán các taint thiết bị lưu chữ thứ cấp.

2.6. Các thành phần chính

2.6.1. Lưu trữ các thẻ Taint

- Việc lựa chọn cách lưu trữ các thẻ dấu vết sẽ ảnh hưởng đến hiệu năng và bộ nhớ của hệ thống. Các hệ thống kiểm tra dấu vết động thường lưu trữ các thẻ mức từng byte hay word dữ liệu. Các thẻ taint thường được lưu trữ trong bộ nhớ bóng không liền kề và bản đồ dấu vết.
- Dalvik có 5 kiểu biến cần cho lưu trữ dấu vết gồm biến nội bộ phương thức, đối phương thức, trường của lớp tĩnh, trường thực thể lớp và mảng. Mỗi biến đều được lưu trữ dạng vector với độ dài 32bit mã hóa nên cho phép tạo 32 taint khác nhau.



Hình 2.3. Định dạng thay đổi trong Stack

2.6.2. Phân tán các mã biên dịch taint

TaintDroid được kiểm tra theo luồng nên chính xác và không làm ảnh hưởng đến hiệu năng. Nó thực hiện kiểm tra các taint ở mức biến với máy ảo biên dịch Dalvik. Các biến cung cấp ngữ nghĩa có giá trị cho việc phân tán taint, các con trỏ dữ liệu riêng biệt từ các biến vô hướng và chủ yếu kiểm tra các biến kiểu số nguyên, số thư. Tuy nhiên có vài trường hợp khi các tham chiếu đối tượng cần được theo dõi để đảm bảo việc phân tán taint hoạt động chính xác.

2.6.3. Phân tán các taint mã nguyên gốc

Mã nguyên gốc không được theo dõi trong TaintDroid. Lý tưởng là chúng ta có được ngữ nghĩa phân tán như trong bản sao biên dịch. Do đó sẽ cần 2 tiền điều kiện cho việc kiểm tra taint chính xác trong môi trường như Java. TaintDroid có được các tiền điều kiện thông qua việc đo không tự động, hồ sơ phương thức và phụ thuộc vào các yêu cầu theo tình huống.

2.6.4. Phân tán taint IPC

Các thẻ taint phải được phân tán giữa các ứng dụng khi chúng trao đổi dữ liệu. Việc theo dõi ảnh hưởng đến hiệu năng và bộ nhớ hệ

thống. TaintDroid theo dõi taint ở mức thông điệp. Một thẻ taint thông điệp thể hiện bên trên của các đánh dấu taint được gán cho các biến chứa bên trong thông điệp. Chúng ta dùng tính chất của mức thông điệp để tối ưu hiệu năng và lưu trữ trong lúc IPC. Và cũng chọn việc thực hiện mức thông điệp trên việc phân tán taint mức biến vì trong một hệ thống mức biến, một bộ nhận có thể kiểm soát bởi việc mở gói các biến theo cách khác để biết thông tin không cần phân tán taint.

2.6.5. Phân tán các taint thiết bị lưu trữ thứ cấp

Các thẻ taint có thể bị mất khi lưu dữ liệu vào tệp, thiết kế hiện tại lưu mỗi thẻ taint vào một tệp. Để làm điều này chúng ta đã phải mở rộng thuộc tính hỗ trợ bởi hệ thống tệp của máy chủ Android và định dạng thẻ nhớ ngoài theo hệ thống tệp ext2. Như với các mảng và IPC, lưu trữ mỗi thẻ taint mỗi tệp dẫn đến vấn đề giới hạn của các đánh dấu taint cho CSDL thông tin. Cách khác chúng ta có thể kiểm tra các thẻ taint bằng một bộ tinh chỉnh để giảm chi phí của việc thêm bộ nhớ và vấn đề hiệu năng.

2.6.6. Thư viện giao diện taint

Tài nguyên taint được định nghĩa bên trong môi trường ảo hóa phải kết nối các thẻ taint với hệ thống kiểm tra. Chúng ta trừu tượng tài nguyên taint vào một thư viện giao diện taint đơn lẻ. Giao diện thực hiện 2 chức năng: 1) thêm các đánh dấu taint vào các biến; và 2) nhận các đánh dấu taint từ các biến. Thư viện chỉ cung cấp khả năng để thêm, bớt các thẻ taint như chức năng có thể được dùng bởi mã Java không tin cậy để xóa các đánh dấu taint.

Chương 3. Cải tiến theo dõi nguồn dữ liệu nhạy cảm

Trước khi miêu tả chi tiết các giải pháp cải tiến theo dõi dữ liệu nhạy cảm, chúng ta sẽ phân tích chi tiết các thành phần của hệ thống hiện tại. TaintDroid chỉ kiểm tra ở mức luồng dữ liệu, không thể kiểm tra ở mức luồng điều khiển để có thể tối ưu hóa hiệu năng. TaintDroid chỉ có thể kiểm tra dữ liệu nhận được của ứng dụng và những hành động khả nghi. Tuy nhiên các ứng dụng độc hại thực thụ nó sẽ lấy thông tin của người dùng thông qua các luồng điều khiển. Việc kiểm tra luồng điều khiển yêu cầu việc xử lý tĩnh, có nghĩa không thể xử lý các ứng dụng bên thứ 3 khi không có mã nguồn. Điều khiển trực tiếp các luồng có thể kiểm tra được một các linh hoạt khi taint được xác định..

3.1. Giải pháp cải tiến theo dõi truy cập lịch sử trình duyệt

Nhưng với nhu cầu và thói quen của người dùng thiết bị di động hiện nay, việc truy cập Internet là rất cao, cùng với đó là việc sử dụng các ứng dụng truy cập Internet như FaceBook, Twitter, Browser, … để làm việc và giải trí. Như chúng ta đã thấy, với cơ chế hoạt động của trình duyệt website cũng như các ứng dụng hoạt động trên nền website thì có rất nhiều thông tin được trình duyệt lưu lại trong quá trình sử dụng và trong đó cũng chứa nhiều thông tin nhạy cảm.

Vậy chúng ta có thể cải tiến hệ thống để hỗ trợ việc kiểm soát truy cập lịch sử của trình duyệt, nó cũng góp phần cho hệ thống được hoàn thiện hơn trong việc giám sát các ứng dụng truy cập dữ liệu nhạy cảm của người dùng. Trong khuôn khổ của việc nghiên cứu trên nền tảng Taintdroid, mọi nghiên cứu cũng như kiểm soát truy cập sẽ được thực hiện trên trình duyệt Google Chrome vì nó là trình duyệt ngầm định trên hệ điều hành Android.

Như đã trình bày, việc cải tiến là nhằm đưa phân tích chi tiết thông tin lịch sử trình duyệt để TaintDroid có thể theo dõi và thông báo khi có truy cập không an toàn. Nên trước tiên sẽ phải thêm các

logic trong lớp TaintDroid NotifyService của chương trình TaintDroidNotifyController và xử lý taint này và dưới đây là chi tiết từng bước cần thực hiện giải pháp cái tiến:

Bước 1: Lấy thông tin log của hệ thống

Bước 2: Tìm kiếm taint lịch sử trình duyệt

Bước 3: Xử lý taint

Bước 4: Điều chỉnh và gửi thông báo

3.1.1. Lấy thông tin log của hệ thống

Lấy thông tin log của hệ thống, để lấy được thông tin log của hệ thống, cần phải xử lý dữ liệu trên lớp TaintDroidNotifyService, đây là lớp thực thi dịch vụ. Mỗi khi thực thi dịch vụ, các block được xử lý liên tục đến khi dừng dịch vụ. Nó hỗ trợ lấy các khối hàng đợi của hệ thống thông qua giao diện BlockingQueue. Các block này sẽ được chuyển đổi sang các đối tượng log tương tự khái báo của TaintDroid. Mỗi khối trong hàng đợi đều chứa đầy đủ các thông tin như: thời gian, mã số tiến trình, thẻ và thông điệp.

3.1.2. Tìm kiếm taint lịch sử trình duyệt

Tìm kiếm taint lịch sử trình duyệt trên log, khi xử lý từng khối của hàng đợi, hệ thống sẽ trích xuất dữ liệu từ thông điệp để lấy ra các thông tin cần thiết như: địa chỉ IP, kiểu Taint, tên tiến trình, dữ liệu. Cụ thể việc tìm kiếm và định danh taint từ thông điệp là thông qua việc phân tích thông điệp theo mẫu. Nếu tìm được các số của taint thì sẽ so sánh với bảng ánh xạ định nghĩa sẵn. Tại thời điểm trên, chúng ta sẽ tìm các taint tương ứng của browser history.

3.1.3. Xử lý taint

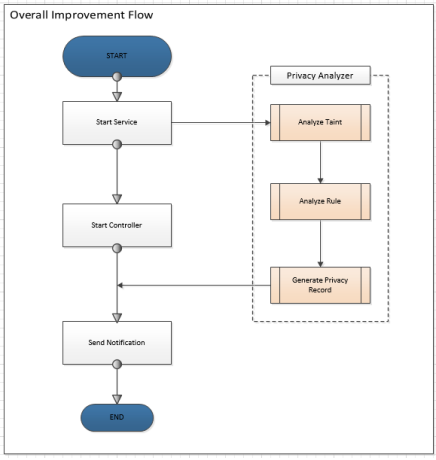
Trong luồng xử lý taint của hệ thống, chúng ta sẽ tiến hành kiểm tra và xử lý taint lịch sử trình duyệt. Dựa vào dữ liệu taint, chúng ta sẽ tiến hành kiểm tra xem ứng dụng có trong danh sách đen truy cập không. Nếu không thuộc danh sách đen sẽ tiến hành kiểm tra xem có vi phạm truy cập thông tin đã định nghĩa. Hình 3.3 dưới đây sẽ miêu tả chi tiết luồng xử lý lịch sự trình duyệt tích hợp trong luồng xử lý của hệ thống.

3.1.4. Điều chỉnh và gửi thông báo

Dựa vào kết quả kiểm tra vi phạm truy cập theo luồng xử lý như hình 3.4 ở trên, nếu vi phạm chúng ta sẽ điều chỉnh thông báo dựa theo mức độ vi phạm. Mỗi thông báo thuộc phần cải tiến sẽ đồng thời hiển thị đèn LED theo cường độ và tần suất xác định trước theo cải tiến. Ngoài ra hệ thống cũng phát âm thanh riêng biệt theo mức độ truy xuất thông tin nhạy cảm của người dùng. Các thông báo trước khi được đưa lên giao diện của hệ thống đều được điều chỉnh phù hợp với mức độ bảo mật bị truy cập như trên.

3.2. Giải pháp cải tiến tổng quát

Trên cơ sở cải tiến kiểm soát truy cập thông tin nhạy cảm của lịch sử trình duyệt ở trên, tôi xin được đề xuất giải pháp cải tiến tổng quát như sau. Chúng ta sẽ xây dựng một bộ phân tích chính sách (Privacy Analyzer), nó sẽ phân tích các taint theo các luật của chính sách (Privacy rule) đầu vào xác định và trả lại kết quả dữ liệu taint có vi phạm chính sách can thiệp hay không. Khi dịch vụ của TaintDroid được chạy, các module đăng ký dịch vụ có thể nhận được các taint. Module có thể sử dụng bộ phân tích chính sách này để kiểm tra vị phạm chính sách của một taint bất kỳ.



Hình 3.4. Luồng Cải Tiến Tổng Quát

3.2.1. Phân tích taint

Khi dịch vụ TaintDroidNotifyService được chạy, các module đăng ký dịch vụ có thể nhận được các taint đang được ứng dụng truy cập và gửi đi theo luồng. Module sẽ chuẩn bị các luật của chính sách, gửi kèm thông tin về taint để xử lý.

Dịch vụ thông báo TaintDroid được mở rộng từ dịch vụ Android, nên nó sẽ lắng nghe tất các dữ liệu trong hàng đợi của hệ điều hành. Dữ liệu trong hàng đợi đều được định nghĩa theo cấu trúc theo lớp hàng đợi của Android. Mỗi khi có khối hàng đợi được truyền đi, TaintDroid sẽ lấy dữ liệu bằng phương thức take() của lớp Queue. Như vậy mỗi hàng đợi block điều được TaintDroid xử lý và khóa theo taint đã định nghĩa để thông báo cho người dùng khi cần thiết. Dưới đây là đoạn mã chương trình minh họa xử lý trên.

3.2.2. Phân tích luật

Khi một module mới được mở rộng từ dịch vụ của TaintDroid, nó sẽ nhận được các hàng đợi của dịch vụ và chuẩn bị luật của riêng mình. Sau đó nó sẽ gọi thư viện của PrivacyAnalyzer để xử lý và kiểm tra vi phạm chính sách. Thông thường các module tao mới để kiểm tra truy cập thông tin nhạy cảm, nên nó sẽ cung cấp các bộ luật tương ứng đến thông tin nhạy cảm. Để chuẩn bị thông tin đầu vào trước khi xử lý, cần phải xác định danh sách các luật để kiểm tra. Và dưới đây là cấu trúc của một luật, chính là một thực thể tương ứng của lớp PrivacyRule.

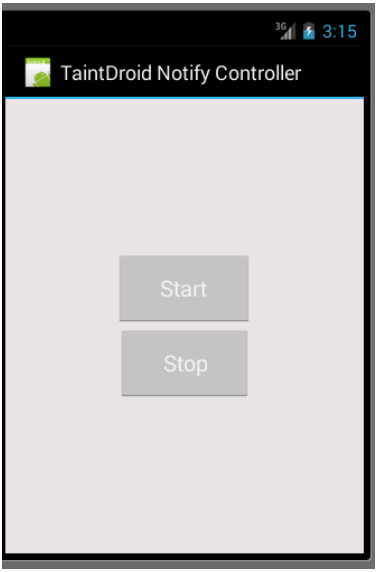
3.2.3. Tạo bản ghi chính sách can thiệp

Khi việc phân tích các luật trên taint đầu vào hoàn thành, bộ phân tích chính sách sẽ chuẩn bị dữ liệu để tạo ra bản ghi chính sách và trả lại cho module gọi. Bản ghi sẽ miêu tả chi tiết tình trạng vi phạm chính sách từ trạng thái đến đặc tả chi tiết về vi phạm. Để tạo tra cấu trúc bản ghi cần tạo ra một lớp mới có tên PrivacyRecord và phải công khai để các module khác có thể truy xuất và dưới đây là hình ảnh minh họa cấu trúc của lớp này.

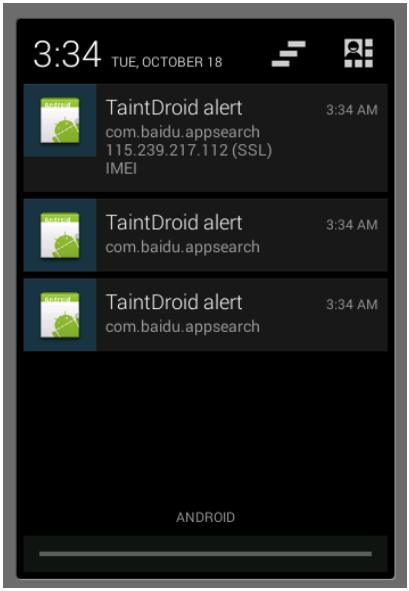
Chương 4. Kết quả thử nghiệm

4.1. Kiểm tra cải tiến

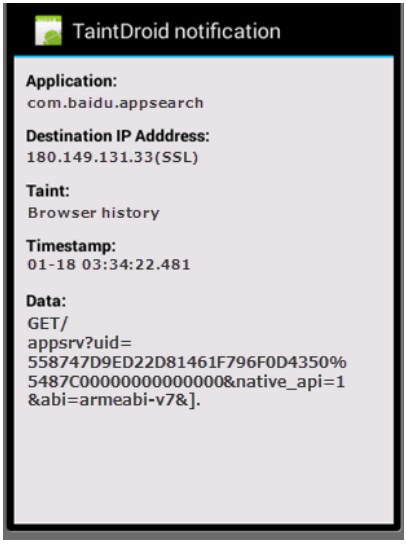
Sau khi cài đặt xong ứng dụng TaintDroid Notify trên điện thoại, ta sẽ chọn ứng dụng từ màn hình chính, chọn nút "Start" để bắt đầu theo dõi và dưới đây là một số hình ảnh minh hoạt hoạt động kiểm tra của cải tiến truy cập lịch sử trình duyệt.



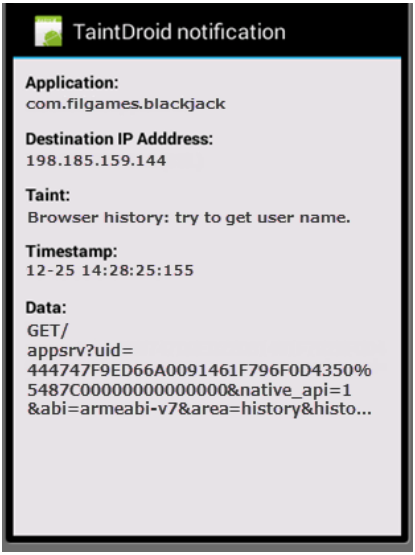
Hình 4.1: Giao diện chương trình thực thi



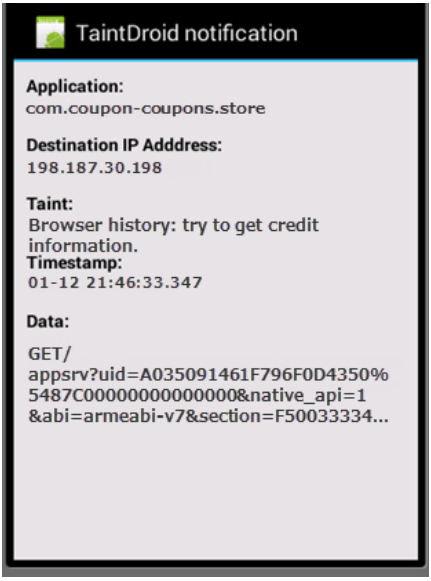
Hình 4.2: Danh sách thông báo



Hình 4.3. Thông báo truy cập lịch sử trình duyệt



Hình 4.4. Thông báo truy xuất tên đăng nhập



Hình 4.5. Thông báo truy xuất mã số thẻ tín dụng

4.2. Đánh giá cải tiến

Hệ thống sau khi cải tiến (TaintDroid') đã hoạt động bình thường và không có bất kỳ thay đổi nào về mặt kiến trúc cũng như luồng xử lý ảnh hưởng đến hệ thống cũ. Và dưới đây là một số tiêu chí đánh giá nổi bật so với hệ điều hành gốc (Android) và hệ thống TaintDroid trước khi cải tiến.

4.2.1. MacrobenchMarks

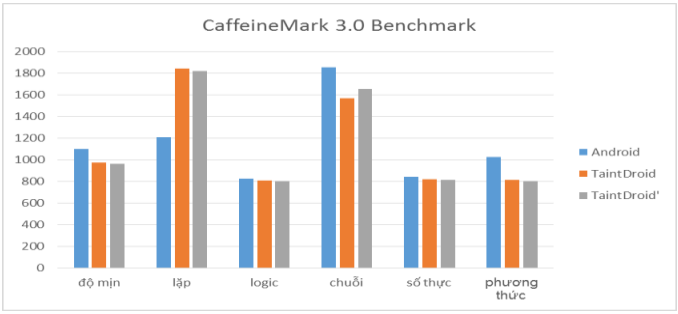
Mỗi thực nghiệm tiến hành đo 50 lần và quan sát khoảng 95% thời lượng. Mỗi trường hợp đều bỏ qua lần chạy đầu để không tính thời gian khởi tạo và kết quả được ghi nhận trong bảng 4.1.

Bảng 4.1: Kết quả MacrobenchMarks (1.000 thông điệp)

	Android	TaintDroid	TaintDroid'
Tải ứng dụng	63 ms	65 ms	68 ms
Tạo danh bạ	348 ms	367 ms	372 ms
Đọc danh bạ	101 ms	119ms	125 ms
Gọi điện	96 ms	106 ms	112 ms
Chụp ảnh	1718 ms	2216 ms	2247 ms

4.2.2. Java MacrobenchMark

Hình 4.6 cho thấy kết quả thời gian thực thi của một chỉ số microbenchmark Java. Một cổng Android của CaffeineMark tiêu chuẩn được sử dụng. CaffeineMark chỉ dùng một số liệu lưu trữ hữu dùng cho các so sánh liên quan.



Hình 4.6. Microbenmark của overhead Java

4.2.3. IPC MacrobenchMark

Chỉ số IPC xem xét chi phí trong khi điều chỉnh các gói. Trong thực nghiệm này các ứng dụng client-service được tạo ra để thực hiện các giao dịch nhanh nhất có thể. Dịch vụ cung cấp các đối tượng tài khoản (tên đăng nhập, số dư) và 2 giao diện setAccount() và getAccount(), thực nghiệm đo được thời gian client yêu cầu thực hiện mỗi giao diện 5.000 lần. Bảng 4.2 tóm tắt kết quả chỉ số IPC.

Bảng 4.2: Kết quả kiểm tra thông lượng IPC (5.000 thông điệp)

	Android	TaintDroid	TaintDroid'
Thời gian (s)	9.24	10.03	10.15
Bộ nhớ (ứng dụng)	21.05 MB	21.76 MB	22.04 MB
Bộ nhớ (dịch vụ)	18.52 MB	19.42 MB	20.72 MB

4.3. Định hướng tiếp theo

Mối quan ngại về an ninh thông tin trên điện thoại di động đang gia tăng. Các bảo vệ ở mức hệ điều hành như Kylin, Saint và Security-by-Contact cung cấp các máy bảo mật cải tiến cho Android và Windows Mobile. Các tiếp cận trên trống lại việc truy cập đến các thông tin nhạy cảm, tuy nhiên khi thông tin được đưa vào ứng dụng thì không thể thêm được bất cứ điều chỉnh nào. Với hệ thống có

những màn hình to hơn, các widget đồ họa có thể hỗ trợ người dùng các cách truy cập trực quan hơn. Các hệ điều hành điều khiển luồng thông tin phân quyền cải tiến như Asbestos và HiStar xử lý gán nhãn và truy xuất điều khiển dựa trên mô hình lưới Denning cho bảo mật luồng thông tin. Flume cung cấp các cải tiến tương tự cho các trừu tượng hệ điều hành.

KẾT LUẬN

Luận văn đã cải tiến TaintDroid thành công như mục tiêu đặt ra, tính năng giám sát truy cập các thông tin nhạy cảm trong lịch sự trình duyệt đã hoạt động. Sau khi cải tiến, hệ thống vẫn hoạt động với các luồng xử lý chính như ban đầu. Hiệu năng của hệ thống sau khi cải tiến cũng duy trì ở mức độ cao. Các tài nguyên khi chạy ứng dụng cải tiến ra tăng không đáng kể và không ảnh hưởng nhiều so với hệ thống ban đầu. Phần cải tiến cũng đã đưa ra giải pháp tổng thể cho việc phân tích chính sách truy cập taint. Tuy cải tiến vẫn còn một số hạn chế, nhưng nó cũng mang tính ứng dụng cao và dưới đây sẽ chỉ ra hai mặt trên.

- Tính ứng dụng: Hiện nay, việc người dùng thường xuyên bị thu thập thông tin cá nhân nói chung và các thông tin trong lịch sử trình duyệt là rất phổ biến. Nhất là khi họ cài đặt những ứng dụng không tin cậy từ bên thứ 3. Việc sử dụng trình duyệt để đăng nhập vào các website cũng như mua sắm trực tuyến và cung cấp thông tin đăng nhập hay thẻ tín dụng đã rất phổ biến. Cái tiến thông báo truy xuất trái phép thông tin trên từ lịch sử trình duyệt rất thiết thực. Người dùng có thể biết khi nào các ứng dụng không tin cậy truy xuất các thông tin trên. Nó sẽ góp phần hỗ trợ người dùng tốt hơn trong việc kiểm soát bảo mật thông tin cá nhân trong thời đại bùng nổ Internet hiện nay.

- Các hạn chế: Lịch sử trình duyệt chỉ là một cải tiến cụ thể và có thể áp dụng với các thành phần khác. Các cải tiến cũng đang bị giới hạn do TaintDroid chỉ kiểm tra theo luồng dữ liệu. Nó không thể kiểm tra ở mức luồng điều khiển để có thể tối ưu hóa hiệu năng. Việc kiểm tra luồng điều khiển yêu cầu việc xử lý tĩnh, có nghĩa không thể áp dụng để xử lý các ứng dụng bên thứ 3 mà không có mã nguồn. Hơn nữa, với sự tinh vi như hiện nay, các ứng dụng độc hại thực thụ có khả năng sẽ mã hóa dữ liệu trước khi gửi đi, nên việc kiểm tra luồng dữ liệu và lọc thông tin tại thời điểm này sẽ bị vô hiệu hóa.