# MỤC LỤC

[MỤC LỤC 1](#_Toc444551962)

[LỜI CAM ĐOAN 4](#_Toc444551963)

[LỜI CẢM ƠN 5](#_Toc444551964)

[DANH MỤC CÁC TỪ VIẾT TẮT 6](#_Toc444551965)

[DANH MỤC HÌNH VẼ 8](#_Toc444551966)

[DANH MỤC CÁC BẢNG 9](#_Toc444551967)

[PHẦN MỞ ĐẦU 10](#_Toc444551968)

[Cơ sở thực tiễn 10](#_Toc444551969)

[Mục tiêu và nhiệm vụ nghiên cứu. 11](#_Toc444551970)

[Bố cục của luận văn 11](#_Toc444551971)

[CÁCH TIẾP CẬN 12](#_Toc444551972)

[Khảo sát các chợ phần mềm Android 12](#_Toc444551973)

[Phát triển chương trình thu thập dữ liệu 12](#_Toc444551974)

[Thu thập dữ liệu và xây dựng cơ chế đánh giá độ an toàn 13](#_Toc444551975)

[Xây dựng hệ thống kênh thông tin cho người dùng 13](#_Toc444551976)

[CHƯƠNG 1. TỔNG QUAN 14](#_Toc444551977)

[I. Hiện trạng an toàn của các ứng dụng Android. 14](#_Toc444551978)

[1. Khái niệm. 14](#_Toc444551979)

[2. Cách thức quản lý. 14](#_Toc444551980)

[3. Các mối nguy cơ. 15](#_Toc444551981)

[4. Giải pháp hiện có 19](#_Toc444551982)

[5. Bài toán. 19](#_Toc444551983)

[II. Cơ sở lý thuyết và công cụ thực hiện. 20](#_Toc444551984)

[1. Nền tảng Android. 20](#_Toc444551985)

[2. Hệ thống permissions. 22](#_Toc444551986)

[3. Tổng quan về điện toán đám mây. 24](#_Toc444551987)

[4. Môi trường Koding. 27](#_Toc444551988)

[5. Thư viện Scrapy. 33](#_Toc444551989)

[6. Thư viện Web Tornado 36](#_Toc444551990)

[CHƯƠNG 2. ĐỀ XUẤT CƠ CHẾ 39](#_Toc444551991)

[I. Mô hình chung. 39](#_Toc444551992)

[II. Mô hình thu thập dữ liệu. 39](#_Toc444551993)

[1. Mô hình 39](#_Toc444551994)

[2. Giải thích 39](#_Toc444551995)

[III. Mô hình phân tích. 39](#_Toc444551996)

[1. Mô hình 39](#_Toc444551997)

[2. Giải thích 39](#_Toc444551998)

[IV. Mô hình tổng hợp điểm. 39](#_Toc444551999)

[1. Mô hình 39](#_Toc444552000)

[2. Phân tích permission. 39](#_Toc444552001)

[3. Cơ chế đề xuất 39](#_Toc444552002)

[4. Description Checking 40](#_Toc444552003)

[5. Permission Rating 40](#_Toc444552004)

[6. Social Rating 42](#_Toc444552005)

[CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ 43](#_Toc444552006)

[I. Kiến trúc ứng dụng. 43](#_Toc444552007)

[1. Phân tích thiết kế. 43](#_Toc444552008)

[2. Biểu đồ. 43](#_Toc444552009)

[II. Cài đặt. 43](#_Toc444552010)

[1. Client 43](#_Toc444552011)

[2. Server 43](#_Toc444552012)

[3. Cài đặt môi trường Koding. 43](#_Toc444552013)

[4. Xây dựng module dịch vụ. 43](#_Toc444552014)

[5. Xây dựng module Crawler. 43](#_Toc444552015)

[6. Xây dựng module đánh giá độ an toàn. 43](#_Toc444552016)

[III. Thực nghiệm 43](#_Toc444552017)

[1. Môi trường 43](#_Toc444552018)

[2. Kịch bản 43](#_Toc444552019)

[3. Đánh giá 43](#_Toc444552020)

[CHƯƠNG 4. KẾT QUẢ VÀ BÀN LUẬN 44](#_Toc444552021)

[I. Kết quả 44](#_Toc444552022)

[II. Bàn luận. 44](#_Toc444552023)

[1. Ưu điểm. 44](#_Toc444552024)

[2. Nhược điểm. 44](#_Toc444552025)

[3. Khó khăn. 44](#_Toc444552026)

[4. Hướng giải quyết. 44](#_Toc444552027)

[KẾT LUẬN 45](#_Toc444552028)

[I. Kết luận 45](#_Toc444552029)

[II. Một số đề xuất 45](#_Toc444552030)

[III. Hướng phát triển của đề tài. 45](#_Toc444552031)

[TÀI LIỆU THAM KHẢO 46](#_Toc444552032)

[PHỤ LỤC 47](#_Toc444552033)

# LỜI CAM ĐOAN

Tác giả luận văn xin cam đoan đây là công trình nghiên cứu của riêng tác giả luận văn đúc kết từ quá trình nghiên cứu từ việc tập hợp các nguồn tài liệu, các kiến thức đã học đến việc tự thu thập các thông tin liên quan và liên hệ thực tế tại đơn vị công tác. Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả luận văn xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Tác giả luận văn xin chịu trách nhiệm về nghiên cứu của mình.

**Học viên thực hiện luận văn**

**Nguyễn Ngọc Tú**

# LỜI CẢM ƠN

Trước tiên, tôi – tác giả luận văn xin được gửi lời cảm ơn đến các Quý Thầy/Cô đã giảng dạy trong chương trình đào tạo thạc sỹ, viện Công Nghệ Thông Tin và Truyền Thông, Đại học Bách Khoa Hà Nội, những người đã tận tình truyền đạt kiến thức trong quá trình tôi học tập. Đây là vốn kiến thức thực sự hữu ích, không chỉ làm nền tảng cho quá trình nghiên cứu khóa luận, mà còn là hành trang giúp tôi vững bước trên con đường hoạt động trong lĩnh vực Công nghệ thông tin nói riêng và trong cuộc sống nói chung.

Với tất cả lòng kính trọng và biết ơn, tôi xin được gửi lời cảm ơn chân thành tới TS. Vũ Thị Hương Giang – người đã định hướng, hướng dẫn cho tôi những ngày đầu thực hiện luận văn. Mặc dù trong quá trình thực hiện luận văn, có những giai đoạn khó khăn không thuận lợi, nhưng những lời động viên, hướng dẫn quý báu của cô đã giúp tôi vượt qua và hoàn thành luận văn này.

Cuối cùng tôi xin gửi lời biết ơn sâu sắc đến gia đình và bạn bè – những người đã luôn bên cạnh giúp đỡ và tạo điều kiện tốt nhất cho tôi trong suốt quá trình học tập cũng như thực hiện luận văn.

Do thời gian có hạn và kinh nghiệm nghiên cứu khoa học chưa nhiều nên luận văn còn nhiều thiếu sót, rất mong nhận được ý kiến góp ý của Quý Thầy/Cô và các bạn học viên.

**Học viên thực hiện luận văn**

**Nguyễn Ngọc Tú**

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải ý nghĩa** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# DANH MỤC HÌNH VẼ

# DANH MỤC CÁC BẢNG

# PHẦN MỞ ĐẦU

## Cơ sở thực tiễn

Hiện nay công nghệ di động phát triển vô cùng mạnh mẽ. Trong đó, những chiếc điện thoại sử dụng hệ điều hành Android trở thành sự lựa chọn phổ biến của nhiều người dùng. Với số lượng người dùng lớn thì đây là một thị trường tiềm năng , và các nhà phát hành cũng như những lập trình viên cá nhân ngày ngày cho ra đời những phần mềm, ứng dụng với tính năng đa dạng, phong phú. Tuy nhiên, khi đối diện với số lượng khổng lồ các ứng dụng có trên các store, người dùng không khỏi bỡ ngỡ cũng như đắn đo về độ an toàn tin cậy của một phần mềm hoặc ứng dụng mà họ quan tâm. Hiện nay không chỉ ở Việt Nam mà trên thế giới xuất hiện những hiện tượng xấu như các phần mềm chạy ngầm nhắn tin SMS tới đầu số tính phí để ăn cắp tiền của người dùng, sử dụng mạng 3G để trừ tiền người dùng sử dụng wap… Vậy làm thế nào để có thể biết được rằng một ứng dụng có khai thác, yêu cầu các quyền can thiệp quá mức so với chức năng mà nó cần?

Điện toán đám mây là một giải pháp toàn diện cung cấp công nghệ thông tin như một dịch vụ. Đó là một giải pháp điện toán dựa trên Internet, ở đó cung cấp tài nguyên phần mềm, phần cứng, dịch vụ… chia sẻ cho người dùng kết nối và sử dụng mỗi khi họ cần. Các máy tính trong các đám mây được cấu hình để làm việc cùng nhau và các ứng dụng khác nhau sử dụng sức mạnh điện toán tập hợp như thể là chúng đang chạy trên một hệ thống duy nhất. Tính linh hoạt của đám mây được thể hiện ở khả năng phân phát tài nguyên theo yêu cầu. Với các dịch vụ sẵn có trên Internet, người sử dụng không phải mua và duy trì hàng trăm, thậm chí hàng nghìn máy tính cũng như phần mềm. Họ chỉ cần tập trung vào kinh doanh lĩnh vực riêng của mình bởi vì nhà cung cấp dịch vụ đám mây đã lo cơ sở hạ tầng và công nghệ thông tin thay họ.

Với những đặc tính kể trên, điện toán đám mây hứa hẹn giải quyết thỏa đáng bài toán đo đánh giá mức độ an toàn của các ứng dụng trên thiết bị di động. Cần nghiên cứu ứng dụng tính năng lưu trữ dữ liệu và xây dựng máy chủ dịch vụ trên đám mây để thiết lập kênh thông tin cho người dùng biết được các ứng dụng đã cài trên máy có yêu cầu hợp lí các quyền của nó hay không - so với các chức năng mà nó cung cấp cũng như so với các ứng dụng cùng loại khác.

## Mục tiêu và nhiệm vụ nghiên cứu.

Trong luận văn này tác giả đề xuất và thử nghiệm ứng dụng công nghệ điện toán đám mây xây dựng một kênh thông tin cung cấp cho người dùng Android các đánh giá về độ an toàn của các ứng dụng, cụ thể:

* Đề xuất cơ chế thu thập các thông tin của các ứng dụng Android: tên ứng dụng, tên package, nhà phát triển, danh sách các permission…
* Đề xuất cơ chế thống kê, đánh giá các thông tin đã thu thập để đưa ra độ an toàn của các ứng dụng.
* Xây dựng hệ thống xử lí công việc thu thập và thống kê thông tin trên đám mây.
* Xây dựng ứng dụng Client trên Android cung cấp giao diện người dùng của kênh thông tin.

## Bố cục của luận văn

Luận văn gồm 5 chương:

* Chương 1: Tổng quan
* Chương 2: Đề xuất cơ chế
* Chương 3: Thực nghiệm và đánh giá
* Chương 4: Kết quả và bàn luận

# CÁCH TIẾP CẬN

Quá trình thực hiện bài luận văn trải qua các bước

## Khảo sát các chợ phần mềm Android

Đầu tiên, tác giả bài luận văn tiến hành xác định những thông tin cần thiết phục vụ cho bài luận văn và khảo sát các chợ phần mềm Android để tìm được chợ phù hợp. Những thông tin cần thiết đã được lựa chọn bao gồm những thông tin cơ bản về phần mềm như: tên, tên package, permission… Ngoài ra còn cần thêm các thông tin gắn liền với từng chợ như số lượng download, rating, miêu tả của phần mềm…Danh sách các chợ phần mềm Android là lớn và đa dạng. Ta có googleplay là chợ phần mềm Android chính thức. Ngoài ra còn có chợ phần mềm chuyên dành cho các phần mềm mã mở F-Droid, hoặc chợ phần mềm của các hang di động như LG, SamSung… Cuối cùng tác giả bài luận văn đã quyết định sẽ chọn googleplay.com là nơi để thu thập dữ liệu.

## Phát triển chương trình thu thập dữ liệu

Ban đầu việc tìm hiểu một môi trường đám mây để triển khai hệ thống được tiến hành. Một vài cái tên được đề xuất như Cloud9, Nitrious, Koding… Cuối cùng Koding đã được chọn bởi tính thân thiện, tiện lợi cũng như khả năng điều khiển quyền root mà máy ảo Koding cung cấp.

Tiếp đến là việc phát triển các chương trình để thu thập dữ liệu. Sau khi phân tích tính mở rộng, môi trường cũng như các thư viện hỗ trợ thì Python đã được chọn là ngôn ngữ sử dụng để phát triển, các thư viện được sử dụng là Scrapy, Web server Tornado. Chương trình quan trọng nhất là Crawler, cào dữ liệu từ trang web googleplay.com. Tuy nhiên có những khó khăn mắc phải như: các permissions trên trang web chỉ hiển thị sau một tương tác với giao diện( cụ thể là ấn vào link ), các permissions này là không nguyên bản, đã được làm thân thiện hóa với người dùng. Vì thế tác giả luận văn đã phải đi tìm hiểu thêm các thư viện hỗ trợ lấy permissions của các ứng dụng android từ googleplay.

## Thu thập dữ liệu và xây dựng cơ chế đánh giá độ an toàn

Tác giả luận văn tìm hiểu nghiên cứu các bài báo khoa học đã được công bố trên thế giới về đề tài permission trên Android. Qua đó đã biết được thêm các phương pháp sử dụng thông tin permissions để đánh giá cũng như dự đoán được các ứng dụng có hại. Tác giả đã tìm hiểu một cách có chọn lọc, lựa chọn các phương pháp được coi là phù hợp với đề tài luận văn để học tập, ứng dụng. Ngoài ra tác giả còn mở rộng, cải tiến bằng việc thêm các tiêu chí đánh giá và đề xuất cơ chế tính độ an toàn.

## Xây dựng hệ thống kênh thông tin cho người dùng

Trong giai đoạn này, tác giả luận văn tiến hành xây dựng hệ thống server sử dụng các chương trình đã phát triển ở trên, đồng thời xây dựng client trên nền tảng Android như là giao diện người dùng. Nhiều cải tiển đã được tiến hành sau khi được giảng viên hướng dẫn góp ý và hướng dẫn.

# CHƯƠNG 1. TỔNG QUAN

## Hiện trạng an toàn của các ứng dụng Android.

### Khái niệm.

Trong những năm gần đây, các thiết bị di động thông minh trên nền tảng Android phát triển một cách bùng nổ. Đi kèm theo đó là số lượng khổng lồ những ứng dụng được phát triển cho người dùng của các thiết bị này. Dự đoán cho biết sẽ có 1 tỉ thiết bị Android được bán ra vào năm 2017, với hơn 50 tỉ lượt tải ứng dụng từ khi Android ra mắt vào năm 2008. Có một lượng người dùng lớn cộng với việc phát triển và phân phối ứng dụng trên chợ ứng dụng Android là dễ dàng khiến nền tảng Android thành đối tượng cho những kẻ xấu. Đã tồn tại những ứng dụng độc hại yêu cầu quá quyền mà chúng nên có và thực hiện các hành vi xấu sau lưng người dùng, như theo dõi dữ liệu cá nhân hoặc gửi tin nhắn mất phí…

Tính an toàn trên Android được xét đến ở đây bao gồm khả năng hoạt động theo đúng như yêu cầu, không lợi dùng quyền để gây tổn hại đến người dùng như việc để rò rỉ hoặc thu thập thông tin một cách không hợp pháp, lợi dụng để thu tiền người dùng hoặc phá hủy dữ liệu…

### Cách thức quản lý.

Nền tảng Android quản lý tính an toàn bằng các chức năng chính sau:

* Bảo mật thông qua Linux Kernel
* Hệ thống cấp quyền permission
* Chữ ký ứng dụng
* Mô hình hộp cát sandbox

Bài luận văn tập trung vào hệ thống cấp quyền permission.

### Các mối nguy cơ.

Hiện tại trên nền tảng Android có tồn tại nhiều mối nguy cơ đe dọa đến độ an toàn. Ở đây tác giả luận văn sẽ trình bày những mối nguy cơ phổ biến nhất. Với một ứng dụng Android độc hại thì có thể chứa nhiều mối nguy cơ khác nhau

* 1. Trojans

Bởi vì cơ chế Sandbox của Android, sẽ được trình bày ở II.1, việc sử dụng virus hoặc worm để tấn công phần lớn là không thể. Do đó những người phát triển phần mềm độc hại hướng tới sử dụng Trojan. Điều đó lí giải cho việc hầu hết các malware trên Android là Trojan.

Những đoạn code độc hại sẽ được gói kèm thành một phần của những ứng dụng tưởng như vô hại. Hoặc thêm vào các ứng dụng đã tồn tại và phân phối lại. Những ứng dụng được sử dụng làm Trojan thường là những ứng dụng trả phí ở chợ ứng dụng chính thức của Google, nhưng sau đó bị chỉnh sửa và thêm mã độc hại vào và phân phối miễn phí trên các chợ ứng dụng của bên thứ ba.

* 1. Spyware

Một trong những loại ứng dụng độc hại phổ biến trên Android là phần mềm gián điệp, được thiết kế để ăn trộm thông tin cá nhân. Phần mềm gián điệp có hai loại: thương mại và độc hại.

Phần mềm gián điệp thương mại là những ứng dụng được cài đặt thủ công lên thiết bị của một người bởi một người khác, với mục đích để do thám người chủ thiết bị. Phần mềm loại này thường được cài vào do chính những nhà sản xuất thiết bị để thu thập thông tin người dùng.

Phần mềm gián điệp độc hại thì lại thu thập thông tin, ăn cắp và truyền thông tin đó đến bên thứ ba. Ví dụ như để biết được các cuộc gọi, các tin nhắn trên thiết bị, có thể phục vụ mục đích như phụ huynh quản lí con em, hoặc chồng quản lí vợ… Nhưng đều vi phạm quyền riêng tư cá nhân.

Một trong những trường hợp nổi tiếng của phần mềm gián điệp thương mại đó là ứng dụng CarrierIQ [TODO], được sử dụng rộng rãi bởi nhiều nhà sản xuất và cung cấp thiết bị di động. CarrierIQ có khả năng biết được tất cả hoạt động được thực hiện trên thiêt bị, bao gồm cả việc tìm kiếm web sử dụng giao thức HTTPS an toàn. Mục đích của CarrierIQ là để thu thập thông tin người dùng và tự đó giúp các nhà cung cấp, sản xuất đưa ra các giải pháp cải thiện mức độ thỏa mãn của người dùng. Nhưng vấn đề phát sinh khi ứng dụng này có quyền quá mạnh và những người dùng bình thường thì không thể xóa bỏ nó. Thêm vào đó, người dùng cũng không thể biết nhà sản xuất và phát hành đã thu thập những thông tin gì.

* 1. Lợi dụng quyền Root

Quyền root ở các thiết bị Android hoạt động tương tự như nền tảng Unix, và có thể so sánh với quyền quản trị admin trên hệ điều hành Window. Một thiết bị Android theo mặc định thì người dùng không có quyền root. Quyền này bị khóa bởi nhà phát hành nhằm ngăn chặn người dùng truy cập vào các phần của hệ thống có thể gây tổn hại, phá hủy hoàn toàn hệ thống, hoặc ngăn việc xóa các ứng dụng được cái đặt bởi các nhà phát hành đó.

Việc mở khóa root trong hầu hết các trường hợp là do các thành viên trong cộng đồng Android tìm hiểu và tiến hành. Mục đích là có thể hoàn toàn quản lí được thiết bị của họ. Nhưng đây cũng là con dao hai lưỡi. Vì những ứng dụng có quyền root cũng có hoàn toàn quyền truy cập đến hệ thống. Nếu quyền root rơi vào tay các ứng dụng độc hại thì chúng có thể hoàn toàn quản lí thiết bị bằng cách xóa quyền của người dùng.

Các ứng dụng Trojan sử dụng quyền root là những ứng dụng độc hại nhất, có thể gây ra mọi loại tàn phá lên thiết bị mà người dùng không hề hay biết. Với quyền root thì chúng có thể thay thế, chỉnh sửa hoặc cài đặt các ứng dụng theo mong muốn. Một ví dụ là Trojan DroidKungFu [TODO], sau khi có được quyền root thì Trojan này sẽ cài đặt một cổng sau trên thiết bị. Cổng sau này được nguy trang khỏi người dùng bởi việc sử dụng tên thân thiện và ẩn đi biểu tượng ứng dụng. Từ đó Trojan sẽ sử dụng cổng sau dể cài đặt các ứng dụng độc hại khác hoặc đơn giản là ăn cắp thông tin cá nhân.

* 1. Botnet

Botnet là một mạng lưới các thiết bị, chủ yếu là máy tính, mà kẻ tấn công dùng cho mục đích cá nhân, thường là ăn trộm thông tin nhạy cảm hoặc là tham giao vào các cuộc tấn công từ chối dịch vụ. Người chủ của các thiết bị này không biết mình bị lây nhiểm ngoài việc thấy thiết bị hoạt động chậm hơn bình thường. Các phiên bản mới của Trojan DroidKungFu có khả năng tạo một mạng Botnet giữa các thiết bị Android bị lây nhiễm.

* 1. Gửi tin nhắn SMS thu phí

Một vài ứng dụng độc hại sẽ hỏi thẳng người dùng cung cấp quyền gửi tin SMS. Khi được chấp nhận vài cài đặt thì các ứng dụng này sẽ gửi tin nhắn đến các đầu số thu phí. Các ứng dụng loại này gặp rất nhiều trong các game được phân phối trên các chợ ứng dụng bên thứ ba.

Một ví dụ điển hinh là Trojan Rufraud [TODO], chúng sẽ lừa người dùng bằng cách giả danh thành các bản miễn phí của các ứng dụng nổi tiếng. Sau khi được cài đặt thì dựa vào vị trí quốc gia của thiết bị mà ứng dụng này sẽ gửi tin nhắn đến đầu số tính tiền tương ứng.

* 1. Drive-by-download

Gần đây, nền tảng Android cũng đã bị tấn công bởi phương thức drive-by-download [TODO]. Phương thức này đánh lừa người dùng bằng cách giả làm một bản cập nhập hệ thống khi người dùng truy cập đến các trang web bị lây nhiễm. Nếu người dùng chấp nhận cài đặt thì sẽ bị nhiễm ứng dụng độc hại.

* 1. Proof-of-concept

Các loại Trojan proof-of-concept là những loại ít nguy hiểm nhất. Những loại tấn công này thường không kèm theo sự phá hủy hoặc lây lan theo diện rộng mà chỉ tập trung vào những thứ nó muốn tại thiết bị. Chúng thường được tạo ra để khoe khoang hoặc để cảnh báo các lỗ hổng bảo mật.

* 1. Trojans phá hủy

Các loại Trojan này nhằm vào việc phá hủy thiết bị bị lây nhiễm, hoặc dữ liệu được lưu trữ trên các thiết bị đó. Chúng cỏ thể làm các tệp tin bị hỏng, xóa sạch dữ liệu…

* 1. Các mối đe dọa khác

Ngoài các mối nguy hiểm chính kể trên thì nền tảng Android còn có thể bị tấn công bởi các mối đe dọa khác như

* Phishing: Phishing là hoạt động giả mạo để lừa lấy thông tin như tài khoản, thẻ giao dịch… Theo nghiên cứu của Felt và Wagner [TODO] thì việc các thiết bị Android có màn hình nhỏ sẽ gây khó khăn cho người dùng ở một mức độ nhận định khi xác định các thông tin, và vì thế dễ bị giả mạo. Đã có những báo cáo cho thấy các ứng dụng giả vờ là các ứng dụng ngân hàng để đánh lừa người dùng cung cấp thông tin đăng nhập.
* Rò rỉ chức năng: Dự án Woodpecker [TODO] báo cáo cho thấy các ứng dụng đang làm rò rỉ quyền truy cấp đến các chức năng của thiết bị, cho phép các ứng dụng khác có quyền sử dụng những chức năng mà chúng vốn không được phép truy cập. Một ví dụ điển hình là ứng dụng Power Control widget [TODO], là một ứng dụng cơ bản của các thiết bị Android. Ứng dụng này làm rò rỉ chức năng GPS, khiến các ứng dụng khác có thể bật tắt GPS thông qua ứng dụng này mà không cần xin quyền. Tuy đây không giống như một vấn đề lớn, nhưng ứng dụng này tồn tại trên các thiết bị Android. Nguy cơ bị làm dụng là lớn hơn nếu như các chức năng bị rò rỉ là nguy hiểm hơn.
* Rò rỉ thông tin: tương tự như rò rỉ chức năng, rò rỉ thông tin sẽ để lộ ra những thông tin nhạy cảm cho các ứng dụng khác trên thiết bị. Điều này có thể bị xảy ra khi các ứng dụng nhạy cảm được lưu trữ ở các khu vực không được bảo vệ, hoặc ứng dụng cung cấp trực tiếp các thông tin cho bất kì lời yêu cầu nào phù hợp. Một ví dụ là công cụ log của HTC được cài đặt trên các thiết bị của họ [TODO]. Công cụ này cung cấp một lượng thông tin lớn cho bất cứ yêu cầu nào sử dụng một request HTTP đơn giản, mà không hề có cơ chế xác thực hoặc phân quyền. Một nguồn rò rỉ thông tin khác là quyền READ\_LOG [TODO]. Quyền này cho phép ứng dụng truy cập vào log của hệ thống, từ đó có thể truy xuất thông tin như GET\_TASKS, DUMP và READ\_HISTORY\_BOOKMARKS.

### Giải pháp hiện có

Biện pháp hiện tại của Android trước những mối nguy cơ từ các ứng dụng là cảnh báo người dùng trước khi cài đặt về yêu cầu các quyền truy cập của ứng dụng đó. Từ đó để người dùng hoàn toàn tự quyết định mà không có thêm sự trợ giúp thông tin nào. Có thể thấy cách tiếp cận này là không hiệu quả. Đa số các ứng dụng Android yêu cầu nhiều quyền permission khác nhau để hoạt động. Khi người dùng mỗi lần cài đặt ứng dụng đều nhìn thấy lời cảnh báo thì hiệu quả của cảnh báo sẽ giảm đi theo thời gian. Và sau đó người dùng hoàn toàn bỏ qua các lời cảnh báo mà không để tâm tới.

Đã có những bài nghiên cứu về tính an toàn của các ứng dụng Android. Ví dụ như những bài nghiên cứu tập trung vào việc phát hiện các ứng dụng độc hại [4], [5]. Tuy nhiên các cách tiếp cận này đòi hỏi việc phân tích mã nguồn ứng dụng hoặc theo dõi các lời gọi lệnh hệ thống trong lúc ứng dụng đang chạy. Vì thế các cách này là khó để triển khai và cài đặt, đồng thời không hiệu quả. Người dùng cũng không muốn sử dụng một ứng dụng thường xuyên chạy và quét hệ thống.

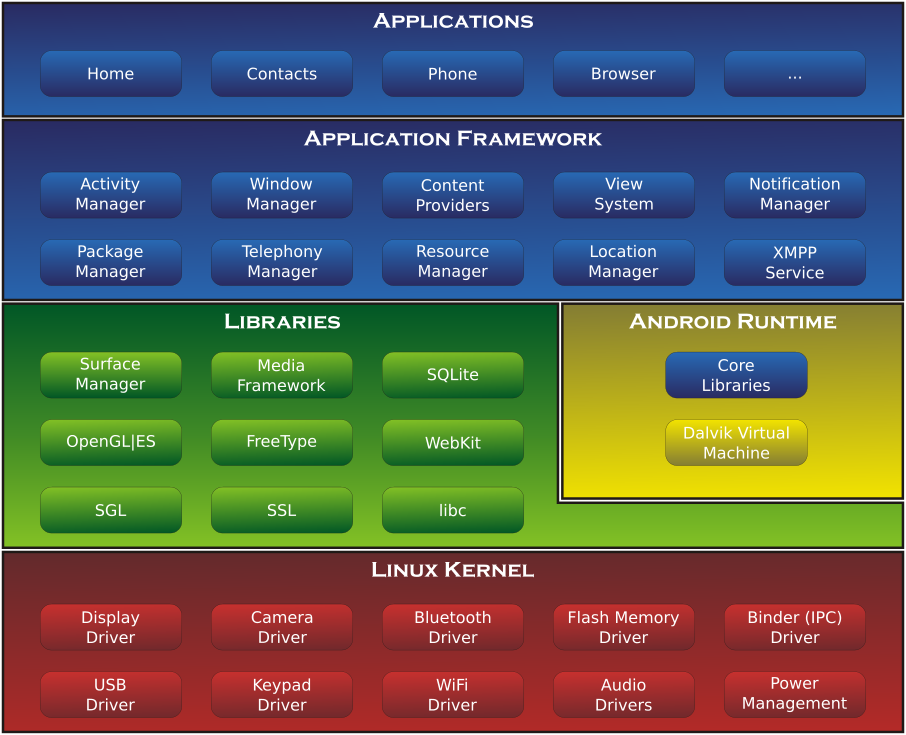
### Bài toán.

Bài toán đặt ra là xây dựng một hệ thống kênh thông tin xét lại tính an toàn của các ứng dụng Android dựa vào hệ thống Permission của từng ứng dụng và các thông tin liên quan đến ứng dụng đó trên chợ ứng dụng. Kênh thông tin sẽ cung cấp cho người dùng biết về độ an toàn của các ứng dụng mà họ quan tâm sau khi được hệ thống đánh giá, đó có thể là các ứng dụng đã cài đặt hoặc các ứng dụng mong muốn cài đặt.

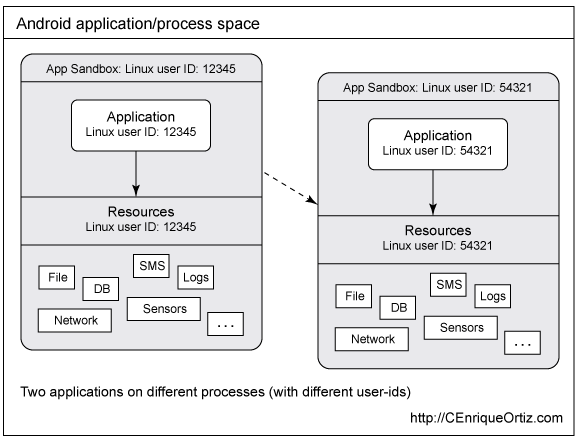
## Cơ sở lý thuyết và công cụ thực hiện.

### Nền tảng Android.

Nền tảng Android được phát triển bởi công ty Google dựa trên Linux2.6, được kết hợp giữa Linux và Java. Nền tảng này sử dụng một kiến trúc đa lớp bao gồm một hệ điều hành đa người dùng dựa vào Linux kernel, các thư viện, các khung quản lí và tập các ứng dụng.



Người dùng nền tảng Android sẽ cài đặt các ứng dụng tải về từ các chợ ứng dụng, như chợ ứng dụng chính thức Google’s play hoặc các chợ ứng dụng bên thứ ba như: SlideMe, F-Droid... Android sử dụng một số các cơ chế để đảm bảo an ninh, trong đó quan trọng nhất gồm cơ chế sandbox và hệ thống permission. Các cơ chế này giúp cho Android quản lí được quyền truy cập của các ứng dụng vào các chức năng chính, tài nguyên của hệ thống.



Cơ chế sandbox tăng cường an ninh giữa các ứng dụng và hệ thống thông qua việc xác định và cô lập các tài nguyên ứng dụng. Mỗi một ứng dụng Android được đặt một định danh độc nhất( UID ) và thực thi ở một luồng riêng biệt. Dưới cơ chế sandbox, các ứng dụng không thể tương tác với nhau và bị giới hạn truy cấp tới các tài nguyên hệ thống. Trong khi hầu hết các ứng dụng Android được phát triển bởi Java, native code cũng được tích hợp vào các ứng dụng. Tất cả các loại ứng dụng bao gồm Java, native và kết hợp đều được chi phối bởi cơ chế sand box giống nhau, và vì thế có mức độ an ninh tương đương.

Một trong những điểm trọng tâm trong cơ chế an ninh của Android là hệ thống permission – quyền truy cập.

### Hệ thống permissions.

Android giới hạn khả năng của các ứng dụng đã cài đặt trên thiết bị bằng cách yêu cầu sự cho phép của người dùng tới quyền truy cập các phần khác nhau của hệ điều hành hoặc các tính năng của thiết bị. Một ứng dụng muốn sử dụng các tài nguyên này thì cần phải yêu cầu các permission tương ứng và được người dùng cho phép trong quá trình cài đặt ứng dụng đó. Hệ thống permission là toàn diện, cung cấp một cơ chế tốt để xác định các tài nguyên mà một ứng dụng cần quyền truy cập khi nó hoạt động trên thiết bị.

Các permission này được lưu trên một tệp tin là manifest.xml và không thể bị thay đổi sau khi ứng dụng đã được cài đặt. Một ngoại lệ cho việc này là khi ứng dụng được cập nhập, khi đó người phát triển có thể yêu cầu thêm các permission mới trong quá trình cài đặt bản cập nhập.

Để yêu cầu các permission thì người phát triển ứng dụng cần khai báo thuộc tính <uses-permission> trong tệp tin manifest.xml. Trong đó trường android:name được dùng để chỉ ra tên của permission:

<uses-permission android:name=”PermissionName” />

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.android.app.myapp" >

<uses-permission android:name="android.permission.RECEIVE\_SMS" />

...

</manifest>

Cơ chế permissions là đồng ý hoặc không sử dụng. Khi một người dùng cài đặt ứng dụng, người dùng đó sẽ phải chấp nhận đồng ý các quyền permissions mà ứng dụng đó yêu cầu hoặc là không cài đặt ứng dụng đó nữa. Giải pháp này giúp cho những nhà phát triển không phải lo lắng về các quyền permissions không được cho phép sẽ gây lỗi ứng dụng hoặc hoạt động bất bình thường. Tuy nhiên nó cũng ảnh hưởng đến các ứng dụng sạch khi mà người dùng từ chối cài đặt do các ứng dụng này yêu cầu các permissions đáng nghi

Permissions được chia ra bốn mức bảo vệ khác nhau :

* Normal: đây là những permission an toàn, thường được coi là không tiềm tàng các mối nguy hiểm. Ví dụ như : VIBRATE, SET\_WALLPAPER. Các permission này được tự động đồng ý và không cần yêu cầu sự cho phép của người dùng.
* Dangerous: đây là những permission có độ rủi ro cao. Các permission này yêu cầu truy cập tới các chức năng có tiềm tàng mối nguy hiểm( ví dụ như : READ\_CONTACTS, SMS… ), từ đó có thể gây ra việc tuồn thông tin cá nhân người dùng, hoặc mất quyền điều khiển thiết bị. Các permission này được hiển thị rõ ràng và yêu cầu sự chấp nhận của người dùng trong quá trình cài đặt ứng dụng.
* Signature: Các permissions mà chỉ các ứng dụng có cùng chứng nhận với các ứng dụng khai báo permission này được sử dụng. Các permissions này được dùng để chia sẻ thông tin giữa các ứng dụng của cùng người phát triển. Nếu chứng nhận là giống nhau thì hệ thống tự động cho quyền mà không cần yêu cầu xác nhận của người dùng.
* SignatureOrSystem: tương tự như signature permissions, để sử dụng các permissions này thì ứng dụng phải là ứng dụng hệ thống hoặc ứng dụng có cùng chứng nhận với các ứng dụng hệ thống. Permission loại này thường được dùng trong các trường hợp đặc biệt khi mà nhiều nhà cung cấp có các ứng dụng hệ thống và muốn chia sẻ chức năng, dữ liệu với nhau.

Người phát triển các ứng dụng cũng có thể tự tạo ra các permission của riêng mình. Các permission này sẽ cho phép các ứng dụng khác quyền truy cập đến các chức năng người phát triển cung cấp qua ứng dụng của mình. Tuy nhiên điều này lại đi ngược với cơ chế sandbox của Android. Các permission như này không được hiển thị khi người dùng cài đặt ứng dụng, nhưng có thể được xác định thông qua việc kiểm tra tệp tin manifest.xml

<permission

android:name="com.testpackage.mypermission" android:label="my\_permission"

android:protectionLevel="dangerous"

android:permission=”android.permission-group.PERSONAL\_INFO” >

</permission>

Đoạn XML trên định nghĩa một permission mới, mypermission thuộc package com.testpackage. Thuộc tính protectionLevel được đặt ở mức Dangerous, do đó permission này sẽ được hiển thị cho người dùng xác nhận khi có yêu cầu. Trong trường hợp này thì permission mới này thuộc loại PERSONAL\_INFO và sẽ được hiển thị cùng với các permission có quyền truy cập thông tin cá nhân khác.

### Tổng quan về điện toán đám mây.

* 1. Khái niệm cơ bản

Điện toán đám mây là một trong những cụm từ được nhắc tới rộng rãi trong thế giới IT. Điện toán đám mây định nghĩa một chức năng được cung cấp bởi các tài nguyên tính toán qua một mạng máy tính. Trong phương pháp tính toán truyền thống, máy tính của người dùng giữ tất cả các phần mềm quan trọng và dữ liệu để thực hiện các công việc tính toán trên các tệp tin đó. Trong điện toán đám mây, máy tính người dùng hầu như không cần thiết phải có gì cả ngoại trừ yêu cầu tối thiểu là hệ điều hành với trình duyệt và kết nối internet để truy cập các tệp tin và chương trình từ các nguồn trực tuyến. Rất nhiều công ty như IBM, Microsoft, HP, Dell, VMware… đang đầu tư rất mạnh vào các nền tảng ảo hóa. Họ đầu tư không chỉ để giúp người dùng truy cập dễ dàng tới các dịch vụ của họ mà còn để tận dụng sức mạnh của họ trên công nghệ đám mây thế hệ mới.

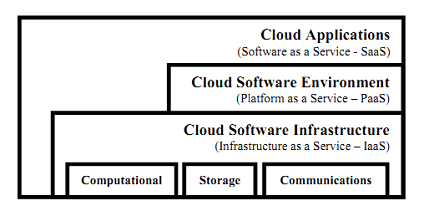
Đến thời điểm hiện tại vẫn chưa có một định nghĩa chính xác nào cho điện toán đám mây, tuy nhiên Ian Foster cung cấp một định nghĩa tổng quát về điện toán đám mây trong bài báo “Cloud Computing and Grid Computing 360-Degree Compared”: “Một mẫu tính toán phân tán quy mô lớn được điều hướng bởi quy mô, trong đó một tập hợp các năng lực tính toán, nền tảng, kho lưu trữ dữ liệu, các dịch vụ được ảo hóa, trừu tượng hóa, có khả năng mở rộng tùy ứng và được cung cấp cho các yêu cầu của khách hàng qua internet”.

Một định nghĩa khác là: “Đám mây là một tập lớn bao gồm các tài nguyên được ảo hóa, dễ dàng sử dụng và dễ dàng truy cập (ví dụ như phần cứng, các nền tảng phát triển hoặc các dịch vụ). Các tài nguyên này có thể được điều chỉnh một cách năng động. Tập hợp các tài nguyên này thường được sử dụng bởi các mô hình trả-khi-dùng.”

* 1. Kiến trúc điện toán đám mây

Kiến trúc điện toán đám mây chủ yếu bao gồm ba lớp: IAAS, PAAS, và SAAS.

* IAAS: Cơ sở hạ tầng như là một dịch vụ. Các nhà cung cấp cho khách hàng sử dụng các tài nguyên phần cứng của họ như kho dữ liệu, máy chủ, thành phần mạng… Việc bảo trì của các tài nguyên phần cứng này được tiến hành bởi các nhà cung cấp dịch vụ đám mây. Thông thường cho trường hợp này, người dùng dịch vụ chỉ phải trả tiền cho những gì mà họ cần, và họ không phải trả tiền sau khi công việc của họ hoàn thành. Tùy vào nhu cầu mà người dùng có thể thay đổi về kích thước, mở rộng tài nguyên họ cần. IAAS được cung cấp cho người dùng bở sự trợ giúp của sự ảo hóa, có hai loại ảo hóa khác nhau:
* Ảo hóa toàn phần: một hệ thống hoặc phần mềm trên một máy có thể thực thi hoàn toàn một hệ thống ảo hóa khác bằng việc giả lập nó
* Ảo hóa bán phần: là một loại mở rộng của ảo hóa toàn phần, nhưng cho phép thực thi nhiều hệ điều hành tại cùng một thời điểm
* SAAS: Phần mềm như là một dịch vụ, các phần mềm được cung cấp cho người dùng thông qua đám mây. Người dùng không cần tiêu tốn tiền quá nhiều để có được giấy phép sử dụng các phần mềm.
* PAAS: Nền tảng như là một dịch vụ, cung cấp môi trường phát triển, xây dựng, kiểm thử và phân phối chương trình phần mềm hoặc dịch vụ qua đám mấy mà không cần phải cài đặt trên máy người dùng.

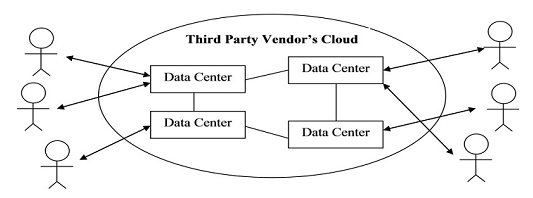


Người dùng điện toán đám mây có thể sử dụng một hoặc nhiều dịch vụ dựa vào yêu cầu của họ. Có rất nhiều cách để sử dụng điện toán đám mây. Ngày nay điện toán đám mây trở thành phổ biến trong tất cả các lĩnh vực kinh doanh, thương mai vì nó giúp làm giảm giá thành khi sử dụng các chương trình phần mềm và phần cứng. Điện toán đám mây giúp giảm việc sử dụng vi xử lý và bộ nhớ. Người dùng có thể yêu cầu không gian bộ nhớ tùy theo nhu cầu một cách nhanh chóng và kinh tế.

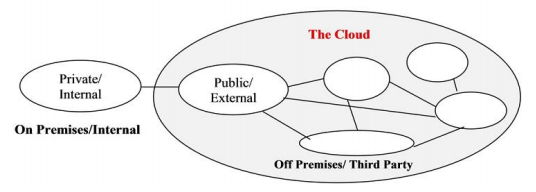
* 1. Các loại đám mây

Điện toán đám mây có bốn loại khác nhau:

* **Đám mây công cộng**: Một thị trường mở cho những yêu cầu về năng lực tính toán cũng như tài nguyên IT. Các nhà cung cấp bên thứ ba cung cấp IAAS, SAAS, PAAS cho người dùng bên ngoài truy cập vào.



* **Đám mây riêng tư**: dành riêng cho các doanh nghiệp, tổ chức với quy mô lớn, sử dụng để bảo trì, cung cấp các dịch vụ trong chính các doanh nghiệp, tổ chức đó.
* **Đám mây lai**: mở rộng của hình thức đám mây riêng tư bằng cách kết nối với các đám mây của các nhà cung cấp bên ngoài để sử dụng tài nguyên, dịch vụ của họ.



* **Đám mây cộng đồng**: Hình thức đám mây riêng tư được chia sẻ giữa các công ty có những yêu cầu giống nhau.

### Môi trường Koding.

Tác giả luận văn chọn Koding là môi trường phát triển phần server cũng như triển khai ứng dụng trên mây.

Koding là một môi trường phát triển trực tuyến dựa trên đám mây cung cấp một máy ảo miễn phí, một IDE ngay trên mây và quyền truy cập root. Koding được phát triển bởi công ty Koding, Inc. Mục đích của Koding là hỗ trợ việc lập trình trực tuyến trên trình duyệt web một các dễ dàng, không cần phải cài đặt các gói công cụ hỗ trợ lập trình. Koding hỗ trợ nhiều ngôn ngữ khác nhau như C, C++, Go, Java, Node.js, Perl, PHP, Python và Ruby.

Trước đây Koding được bến đến với cái tên Kodingen. Năm 2009, Kodingen được cho ra mắt và đi vào hoạt động. Sau đó Kodingen đã tạm dừng trong một thời gian để phát triển thành Koding. Phiên bản đầu tiên của Koding được ra mắt vào ngày 16 tháng một năm 2012 dưới dạng private beta, và đi vào open beta ngày 24 tháng bảy năm 2012.

* 1. Máy ảo

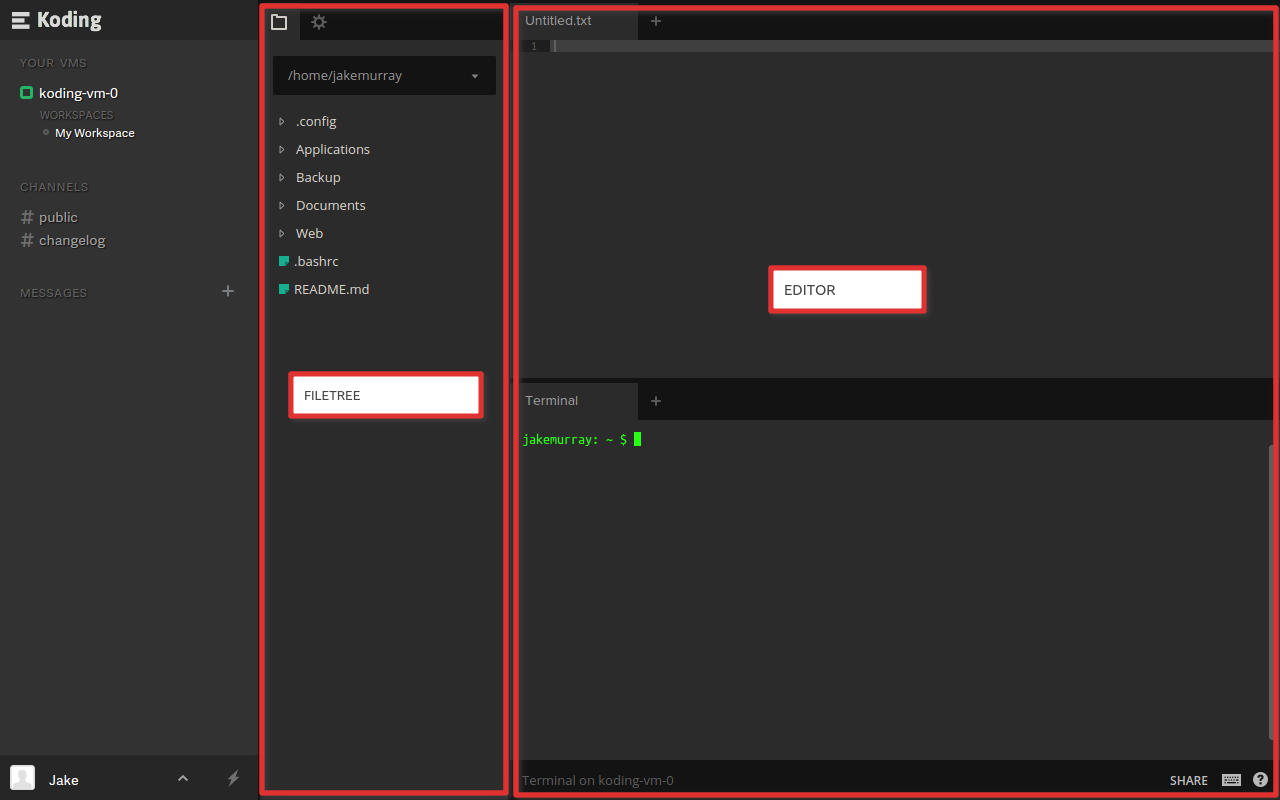
Koding cung cấp cho người dùng những máy ảo để sử dụng. Số lượng và cấu hình các máy ảo phụ thuộc vào loại tài khoản người dùng. Tài khoản miễn phí được cấp sẵn một máy áo vơi cấu hình 1GB Ram, 3GB ổ cứng, CPU 1 core, cài hệ điều hành Ubuntu 14.02

Các máy ảo này có thể được khởi tạo rất nhanh trong vài giây. Khi gặp vấn đề thì người dùng hoàn toàn có thể tái khởi tạo lại máy ảo một cách dễ dàng.

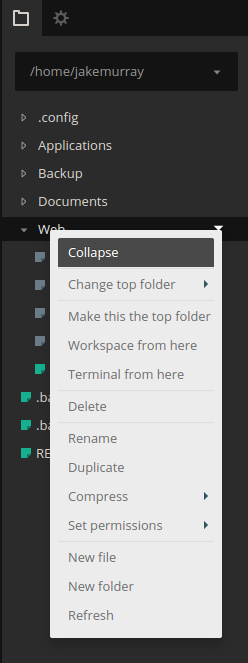
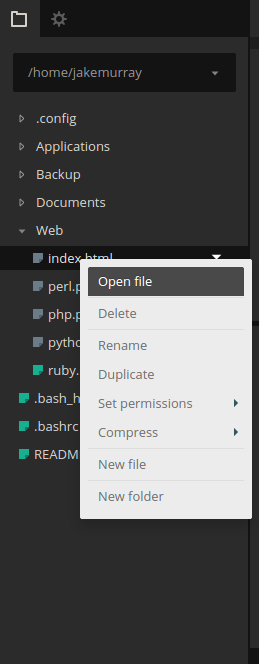
Đặc biệt, trong khi các nhà cung cấp dịch vụ tương ứng khác không cho quyền root thì Koding cho phép người dùng có quyền root trên tất cả các máy chủ của mình. Điều này cho phép người dùng có quyền sử dụng máy mà không bị giới hạn.

* 1. Môi trường phát triển

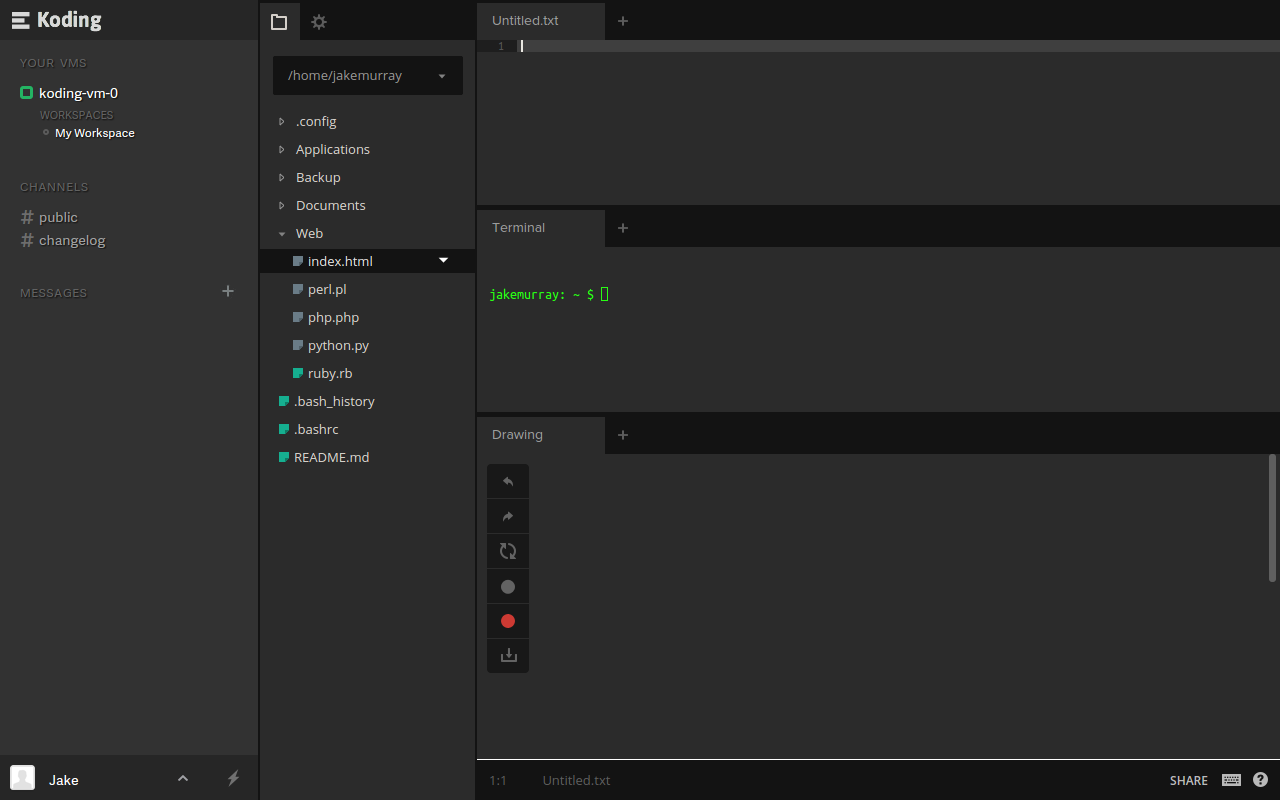
Koding cung cấp một môi trường phát triển tích hợp sẵn trên trình duyệt web, gọi là Koding IDE. Koding IDE bao gồm hai phần chính: cây thư mục và trình soạn thảo.



Koding IDE cho phép người dùng có thể tùy chỉnh giao diện cũng như chức năng theo mong muốn. Trong cây thư mục sẽ hiển thị các tệp tin có trên máy ảo. Người dùng có thể tương tác với các tệp tin, thư mục một cách trực quan thông qua giao diện cây thư mục đó. Ví dụ như: copy, xóa, tạo…

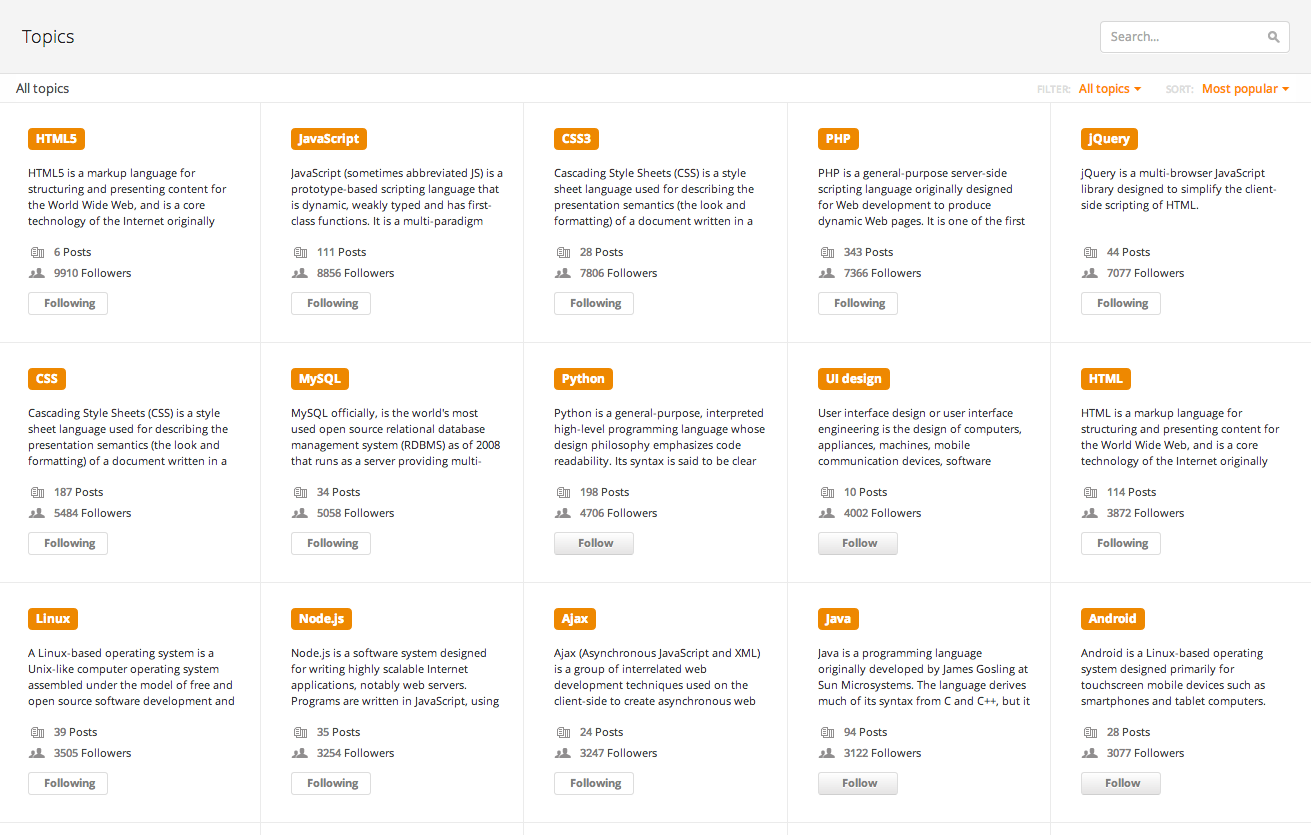


Trình soạn thảo nằm ở bên phải giao diện cây thư mục. Trình soạn thảo của Koding IDE có khả năng chỉnh sửa nhiều kiểu tệp tin khác nhau như .php, .rb, .html. Giao diện trình soạn thảo có thể được phân tách thành các phần nhỏ hơn theo nhu cầu người dùng, có thể để mở một terminal, một giao diện vẽ hoặc một tệp tin.



* 1. Cộng đồng người dùng

Khi đăng ký sử dụng Koding, người dùng đã được thêm vào danh sách nhóm người dùng mặc định “Koding”. Các thông tin mới, các chủ đề, các thông báo sự kiện đều được báo đến người dùng qua nhóm mặc định này. Người dùng cũng có thể chủ động tạo ra các nhóm riêng của mình. Những thành viên trong nhóm có thể biết được thông tin của những thành viên khác, biết được những gì đã được thực hiện, được cập nhập…



Trong cùng một nhóm, người dùng có thể tạo ra các máy ảo chia sẻ để nhiều người có thể truy cập.

Koding đã kết hợp một môi trường phát triển trực tuyến trên đám mây với mạng xã hội người dùng cùng cơ chế quản lí dự án để phát triển thành một dịch vụ mạnh mẽ mà hoàn toàn miễn phí.

* 1. Khuyết điểm

Koding có rất nhiều ưu điểm vượt trội, cung cấp một môi trường phát triển tiện lợi cho người dùng. Nhưng không phải là không có những khuyết điểm:

* Độ trễ: so với việc phát triển tại máy tính để bàn thì Koding chậm hơn do sử dụng kết nối internet. Khi mở một tệp tin thì Koding cần vài giây để hoàn thành.
* Sự phụ thuộc vào internet: Để sử dụng Koding IDE thì cần phải có kết nối internet, do dịch vụ được triển khai trên đám mây.

### Thư viện Scrapy.

Scrapy là một thư viện nguồn mở dùng để thu thập dữ liệu trên web phát triển bằng ngôn ngữ Python. Ngoài ra, qua các bản phát triển, hiện nay thư viện còn có thể được sử dụng để lấy dữ liệu thông qua các API. Scrapy hiện tại được bảo trì bởi công ty Scrapinghub Ltd, một công ty chuyên phát triển và cung cấp dịch vụ thu thập thông tin mạng.

Kiến trúc của Scrapy được miêu tả qua sơ đồ sau:



**Các thành phần**

**Scrapy Engine**

Phần engine đảm nhiệm vai trò điều khiển luồng dữ liệu giữa các thành phần trong hệ thống, kích hoạt các sự kiện khi có các hành động cụ thể xảy ra.

Bộ lập lịch Scheduler

Bộ lập lịch nhận các yêu cầu từ engine và đưa các yêu cầu này vào xếp hàng để xử lí sau.

**Bộ tải Downloader**

Bộ tải đảm nhiệm vai trò tải xuống các trang web, đưa các trang web cho phần engine và từ đó sẽ đưa đến các spider

**Spiders**

Spiders là các lớp được viết bởi người dùng thư viện để tiến hành phân tích các trang web và trích xuất dữ liệu ra thành các item, đồng thời trích xuất ra các đường URL để tiếp tục lần theo và thu thập thông tin. Mỗi một spider có thể đảm nhiệm một tên miền cụ thể hoặc một nhóm các tên miền.

**Item Pipeline**

Item Pipeline có vai trò xử lí các item sau khi đã được trích xuất từ Spider. Nhiệm vụ chủ yếu là:

* Làm sạch dữ liệu HTML.
* Kiểm tra dữ liệu đã thu thập( ví dụ như có chứa những trường cụ thể nào đó hay không).
* Kiểm tra giá trị lặp và loại bỏ.
* Lưu trữ giá trị vào cơ sở dữ liệu.

**Downloader middlewares**

Thành phần này nằm ở giữa Engine và bộ tải, đảm nhiệm vai trò xử lí các yêu cầu từ Engine đến bộ tải và các phản hồi ngược lại từ bộ tải đến Engine. Thành phần này cung cấp một cơ chế tiện lợi để mở rộng các chức năng của Scrapy bằng cách thêm các đoạn code xử lí.

**Spider middlewares**

Tương tự như Downloader middlewares, thành phần này nằm giữa Engine và các Spiders với vai trò xử lí các đầu vào cho spider( các phản hồi ) và xử lí đầu ra của spider( các item, các yêu cầu request ).

**Luồng dữ liệu**

Luồng dữ liệu trong Scrapy được điều khiển bởi sự hoạt động của bộ máy engine:

* Bước 1: Bộ máy engine mở một tên miền, tìm kiếm đến spider xử lí tên miền đó, và yêu cầu spider cung cấp đường dẫn URLs đầu tiên để bắt đầu cào dữ liệu.
* Bước 2: Bộ máy nhận được URLs đầu tiên để cào dữ liệu và lập lịch cho chúng trong bộ lập lịch bằng các gửi các yêu cầu Requests.
* Bước 3: Bộ máy engine yêu cầu bộ lập lịch đưa đường dẫn URL tiếp theo để cào dữ liệu
* Bước 4: Bộ lập lịch trả về đường dẫn URL tiếp theo cho bộ máy engine, từ đó gửi đến bộ tải Downloader thông qua Downloader Middleware( theo hướng request).
* Bước 5: Sau khi trang web cần được cào dữ liệu đã được tải hoàn toàn thì bộ tải Downloader sẽ tạo ra một phản hồi và gửi về cho bộ máy Engine, thông qua Downloader Middleware( theo hướng response ).
* Bước 6: Bộ máy engine nhận được phản hồi từ bộ tải, gửi các phản hồi này vào cho Spider để xử lí, thông qua Spider middleware( hướng đầu vào input ).
* Bước 7: Các spider sẽ xử lí phản hồi, trích xuất thông tin và trả về các item hoặc tạo ra các yêu cầu cào dữ liệu mới theo các đường link được trích xuất đến bộ máy engine.
* Bước 8: Bộ máy engine gửi các item được cào đến Item pipeline để xử lí dữ liệu và gửi các yêu cầu mới được tạo đến bộ lập lịch.
* Bước 9: Tiến trình lặp lại từ bước 2 đến khi không còn yêu cầu từ bộ lập lịch, bộ máy engine tiến hành đóng tên miền.

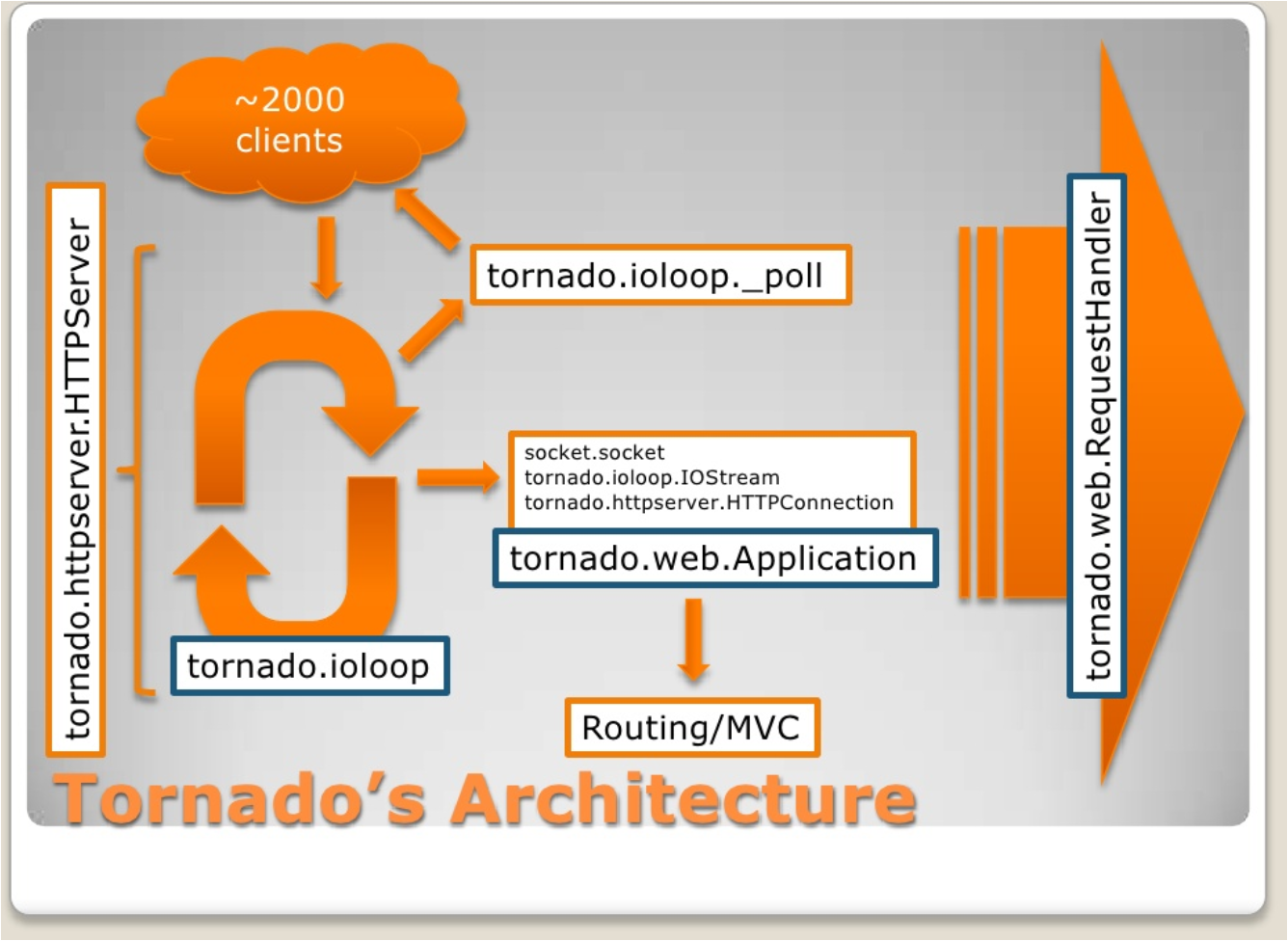
### Thư viện Web Tornado

Tornado là một thư viện web và mạng không đồng bộ được phát triển trên nền Python bởi FriendFeed. Vào năm 2009, khi FaceBook mua lại, Facebook đã open source dự án này. Tornado là một web framework khá đơn giản nhưng có khả năng xử lý các request không đồng bộ, non blocking I/O, Tornado có thể mở rộng lên đến hàng vạn kết nối, và vì thế là thư viện phù hợp cho việc phát triển các chương trình yêu cầu các kết nối lâu dài đến người dùng hoặc phát triển WebSocket

Tornado có thể được chia thành bốn thành phần chính:

* Một thư viện web( bao gồm một RequestHandler để dựa vào đó phát triển ra các ứng dụng web và các lớp hỗ trợ khác nhau ).
* Các lớp cài đặt phía client và phía server sử dụng HTTP( HTTPServer và AsyncHTTPClient ).
* Một thư viện mạng không đồng bộ( IOLoop và IOStream ), sử dụng để phát triển các thành phần HTTP và để cài đặt các giao thức khác.
* Một thư viện Coroutine( tornado.gen ) cho phép xây dựng code không đồng bộ một cách dễ dàng

Chúng ta có hình tổng quan về cách hoạt động của 1 Tornado Server:

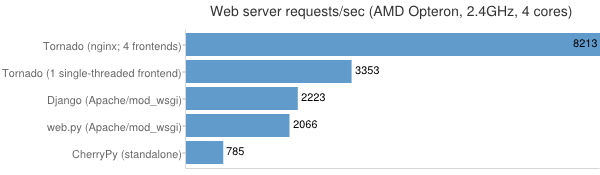


Tuy là một framework đơn giản so với các webframework khác nhưng Tornado vẫn hỗ trợ khác nhiều tính năng, có thể kể đến như:

* Template( với các đoạn code Python nhúng bên trong ).
* Localization(tự động dịch các đoạn text sang ngôn ngữ tương ứng với yêu cầu người dùng)
* XSS, CSRF protection
* Xác thực bằng secure cookie hoặc bên thứ ba(Google, Facebook, Twitter, OAuth…)
* Static file caching

Các web server là threaded server, với mỗi request đến, server sẽ sinh ra một luồng mới để xử lí yêu cầu đó. Tuy nhiên khi số lượng lên đến vài chục nghìn thì việc sinh ra luồng mới sẽ trở nên tốn kém tài nguyên của server. Tornado sử dụng non-blocking IO và là server hướng sự kiện nên có thể xử lý hàng nghìn request. Tuy nhiên để phát huy tối đa khả năng của Tornado thì các thư viện sử dụng cùng với nó cũng phải hoạt động theo cơ chế non-blocking

Ta có kết quả benchmark của Tornado so với các web server khác



# CHƯƠNG 2. ĐỀ XUẤT CƠ CHẾ TỔNG HỢP ĐIỂM AN TOÀN

## Mô hình chung.

E:\Document\Cao hoc\Luan Van\Documents\Báo cáo\DiIn\MoHinhChung.png

Tác giả luận văn đề xuất cơ chế đưa ra điểm an toàn qua ba bước được tiến hành theo trình tự:

* Thu thập dữ liệu
* Phân tích dữ liệu
* Tổng hợp điểm an toàn

## Mô hình thu thập dữ liệu.

### Mô hình

E:\Document\Cao hoc\Luan Van\Documents\Báo cáo\DiIn\MoHinhThuThapDuLieu.png

### Mô tả

Mục đích:

Để có thể đánh giá được độ an toàn của ứng dụng Android, trước hết cần phải thu thập thông tin các ứng dụng trên chợ ứng dụng chính thức của google, xây dựng thành một cơ sở dữ liệu các ứng dụng.

Cách thức:

Trong mô hình thu thập dữ liệu, quan trọng nhất là mô đun Crawler. Mô đun này được phát triển dựa trên thư viện Scrapy.

Mô đun Crawler sẽ tiến hành cào dữ liệu trên trang chủ của chợ ứng dụng chính thức google play. Trong quá trình cào dữ liệu thì trên giao diện trang web không có cung cấp sẵn thông tin về các permission. Vì thế sau khi cào xong dữ liệu và có package\_name của từng ứng dụng, mô đun Crawler sẽ gọi đến thư viện google\_play\_api để request thêm thông tin permission dựa vào package\_name đã có.

Sau khi có đầy đủ thông tin của ứng dụng thì ứng dụng sẽ được lưu lại trong cơ sở dữ liệu

Các thông tin lưu trữ:

* Package name:
* Tên ứng dụng:
* Description:
* User Rating:
* Permissions:
* Editor choice/Top Developer:
* Các thông tin khác: icon url, ảnh chụp màn hình… dùng để hiển thị ở client

## Mô hình phân tích.

### Mô hình

E:\Document\Cao hoc\Luan Van\Documents\Báo cáo\DiIn\Diagram\PhanTich.png

### Phân tích permission.

Như đã đề cập ở chương I phần II.2, hệ thống permission hiện tại bao gồm các permission ở các mức bảo vệ khác nhau: Normal, Dangerous, Signature và Signature-Or-System.

Hai mức bảo vệ cuối là không thể truy cập bởi các ứng dụng có hại. Mức bảo vệ Normal bao gồm các permission tầm thường. Chỉ có lại mức Dangerous là đáng chú ý.

Ở đây tác giả luận văn đề xuất việc chia nhỏ các permission trong mức Dangerous thành các nhóm nhỏ hơn dựa vào các bài báo, nghiên cứu đi trước:

* A - Độ nguy hiểm cao
* B - Độ nguy hiểm trung bình
* C - Độ nguy hiểm thấp
* D - Độ nguy hiểm cực thấp

Trong thực tế cũng cho thấy có những permission là cần thiết cho một loại ứng dụng cụ thể nhưng bị lạm dụng ở các loại ứng dụng khác. Ví dụ điển hình như SEND\_SMS, cho phép gửi tin nhắn SMS. Permission này là bình thường cho các ứng dụng nhắn tin nhưng sẽ là đáng nghi ngờ khi các ứng dụng trò chơi yêu cầu permission này. Vì thế tác giả quyết định chia các ứng dụng thành các nhóm khác nhau, ví dụ như: Trò chơi, Nhắn tin, Giải Trí, Tài chính… Mỗi một nhóm này sẽ có một tập các permission cần thiết để chúng hoạt động được xây dựng dựa trên sự quan sát cũng như tìm hiểu của tác giả. Ngoài ra mỗi một nhóm cũng có danh sách các permission không phù hợp và không nên dùng. Danh sách này được tạo dựa trên đánh giá chung về các quyền mà những ứng dụng thuộc nhóm đó nên yêu cầu. Đồng thời bổ sung vào các permission thường bị sử dụng với mục đích sai trái như trong miêu tả của các bài báo [1], [2], [3].

Bảng các permission phân theo độ nguy hiểm:

Bảng các nhóm ứng dụng và các permission không nên có

## Mô hình tổng hợp điểm.

### Mô hình

E:\Document\Cao hoc\Luan Van\Documents\Báo cáo\DiIn\Diagram\TongHop.png

### Cơ chế đề xuất

Ở chương này, tác giả đề xuất ra một cơ chế kết hợp để đánh giá độ an toàn dựa vào các thông tin về ứng dụng đã thu thập được ở chương trước. Cơ chế này được tính toán bằng cách kết hợp ba tham số đầu vào:

* Kiểm tra miêu tả
* Tính điểm permission
* Tính điểm cộng đồng

Mỗi một tham số đầu vào sẽ được đánh trọng số( 20% cho kiểm tra miêu tả, 70% cho tính điểm permission, 10% cho tính điểm cộng đồng)

Điểm an toàn của ứng dụng cuối cùng sẽ được tính bằng việc cộng tổng các trọng số trên lại với nhau.

Safety Score = Trọng SốMiêu tả + Trọng SốĐiểm Permission + Trọng SốCộng Đồng

### Description Checking

Trên chợ ứng dụng, mỗi ứng dụng đều có một phần miêu tả đặc chưng cho riêng mình. Một mục miêu tả rõ ràng và chi tiết sẽ có khả năng cung cấp cho người dùng thông tin về các quyền mà ứng dụng đó sẽ sử dụng.

Để tính được trọng số miêu tả thì cần thu thập thông tin miêu tả của ứng dụng cũng như các permission mà ứng dụng đó yêu cầu. Sau đó sẽ áp dụng công thức đề xuất sau:

Trọng SốMiêu Tả = ( ) x 0.2 x 10

Trong đó:

* KWF: Số keyword tương ứng với permissions trong manifest được tìm thấy trong miêu tả. Với mỗi một permission sẽ có một tập các keyword tương ứng. Chỉ cần có keyword tồn tại trong nội dung miêu tả ứng dụng thì giá trị KWF sẽ tăng thêm 1 đơn vị
* PC: số lượng permission trong manifest

Như vậy giá trị của Trọng SốMiêu Tả lớn nhất sẽ là 2

### Permission Rating

Trọng SốPermission sẽ được tính bởi công thức:

Trọng SốPermission = (P1 + P2) x 0.7

Trong đó

P1 được tính theo công thức:

P1 =( 1 - ) x 100

Với PA, PB, PC, PD: số lượng các permission thuộc vào các mức A, B, C, D định nghĩa ở trên. VA, VB, VC, VD là điểm số tương ứng của một permission trong các nhóm. Ở đây tác giả đã đơn giản hóa và đặt VA = 4, VB = 3, VC = 2, VD = 1

và P2 được tính theo công thức:

P2 = (1 - ) x 100

Trong đó PC – Số lượng các permission đáng nghi ngờ

P – Tổng permission được yêu cầu

### Social Rating

Tác giả luận văn nhận thấy các yếu tố về cộng đồng cũng cung cấp giá trị trong đánh giá điểm an toàn của ứng dụng. Cụ thể như uy tín, chất lượng của nhà phát triển. Nếu nhà phát triển ứng dụng là Top Developer hoặc ứng dụng được gắn mác Editor Choice thì khi đó, ứng dụng đã được các nhà quản trị chợ ứng dụng đánh giá cao.

Ngoài ra điểm số rating, số lượng download cũng cho thấy được đánh giá của cộng đồng người chơi với ứng dụng đó. Một ứng dụng tốt sẽ có nhiều lượt tải xuống với điểm rating cao. Ngược lại một ứng dụng tồi sẽ nhận được nhiều điểm rating thấp.

Công thức tính trọng số Cộng Đồng như sau:

Trọng SốCộng Đồng =

# CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

## Kiến trúc ứng dụng.

### Sơ đồ kiến trúc.

* 1. *Biểu đồ client server*
  2. *Mô tả hệ thống*
  3. *Yêu cầu chức năng*
  4. *Yêu cầu phi chức năng*

### Phân tích thiết kế.

* 1. *Biểu đồ use case và đặc tả ca sử dụng*
  2. *Biểu đồ lớp*
  3. *Thiết kế cơ sở dữ liệu*

## Cài đặt.

### Client

### Server

### Cài đặt môi trường Koding.

### Xây dựng module dịch vụ.

### Xây dựng module Crawler.

### Xây dựng module đánh giá độ an toàn.

## Thực nghiệm

### Môi trường

Cấu hình server, cấu hình máy client

### Kịch bản

### Đánh giá

# CHƯƠNG 4. KẾT QUẢ VÀ BÀN LUẬN

## Kết quả

## Bàn luận.

### Ưu điểm.

### Nhược điểm.

### Khó khăn.

### Hướng giải quyết.

# KẾT LUẬN

## Kết luận

## Một số đề xuất

## Hướng phát triển của đề tài.

# TÀI LIỆU THAM KHẢO

Tiếng Việt

Tiếng Anh

[1]

[2]

[3]

[4] W. Enck, P.Gilbert, B.-G Chun, L. P. Cox, J. Jung, P.McDaniel, A.N.Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphone.

[5] T. Luo, H. Hao, W. Du, Y. Wang, H. Yin. Attacks on webview in the android system.

[6] <http://code.tutsplus.com/articles/coding-with-koding--net-35285>

[7] http://scrapy.org/

[8] https://viblo.asia/vigov5/posts/73KbvZZOvmWB

# PHỤ LỤC