|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” | Mẫu 2 |

BÁO CÁO CHUYÊN ĐỀ SỐ 4.2.2

“Phương pháp trích xuất đặc trưng động”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

Hà Nội - 2023

|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” |  |

BÁO CÁO CHUYÊN ĐỀ SỐ 4.2.2

“Phương pháp trích xuất đặc trưng động”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

|  |  |
| --- | --- |
| **Người thực hiện chuyên đề** | **Cơ quan chủ trì** |
| *(Họ tên và chữ ký)* | *(Họ tên và chữ ký)* |

Hà Nội - 2023

MỤC LỤC

[MỤC LỤC 1](#_Toc129697764)

[DANH MỤC HÌNH ẢNH 2](#_Toc129697765)

[PHÂN TÍCH, TRÍCH XUẤT ĐẶC TRƯNG ĐỘNG MÃ ĐỘC ANDROID 3](#_Toc129697766)

[1.1. Các thành phần trong ứng dụng android 3](#_Toc129697767)

[1.2. Phân tích gom nhóm họ mã độc của các tập tin 3](#_Toc129697768)

[1.3. Trích xuất đặc trưng bằng phương pháp động 7](#_Toc129697769)

[1.4. Kết luận 11](#_Toc129697770)

[TÀI LIỆU THAM KHẢO 13](#_Toc129697771)

# DANH MỤC HÌNH ẢNH

[Hình 1. Khởi tạo môt trường cho andropytool 4](#_Toc129697740)

[Hình 2. Truy cập vào thư mục andropytool 5](#_Toc129697741)

[Hình 3. Tiến hành phân tích với cơ sở dữ liệu của totalvirus 5](#_Toc129697742)

[Hình 4. Truy cập vào tập tin av\_class 6](#_Toc129697743)

[Hình 5. Lấy dữ liệu phân tích về 6](#_Toc129697744)

[Hình 6. Họ virus được nhận diện với totalvirus 7](#_Toc129697745)

[Hình 7. Các folder chứa kết quả phân tích động 9](#_Toc129697746)

[Hình 8. Các thông tin lấy được trong phân tích động 9](#_Toc129697747)

[Hình 9. Recvnet trong phân tích động 9](#_Toc129697748)

[Hình 10. Accessedfiles 10](#_Toc129697749)

[Hình 11. Opennet trong phân tích động 10](#_Toc129697750)

[Hình 12. Dataleaks trong phân tích động 10](#_Toc129697751)

# PHÂN TÍCH, TRÍCH XUẤT ĐẶC TRƯNG ĐỘNG MÃ ĐỘC ANDROID

## Các thành phần trong ứng dụng android

Các tập tin trên nền tảng Android có phần mở rộng là APK được phân tích để trích xuất và lựa chọn các đặc trưng. Tập tin có phần mở rộng APK là định dạng sử dụng cho việc phân phối và cài đặt các phần mềm ứng dụng điện thoại có hệ điều hành Android. Bản chất của tập tin apk là một tệp tin nén với cấu trúc như sau:

* META-INIF/: chứa thông tin mô tả từ tập tin jar của Java
* res/: chứa các tài liệu nguồn
* libs/: chứa các thư viện .so từ phần mềm phát triển.
* androidManifest.xml: chứa các tập tin cấu hình về quyền truy cập và dịch vụ.
* classes.dex: chứa mã byte-code.
* resources.asc: chứa mã nguồn nhị phân sau khi biên dịch.

## Phân tích gom nhóm họ mã độc của các tập tin

Andropytool là một công cụ để trích xuất cả hai tính năng tĩnh và động từ các ứng dụng Android [12]. Androidpytool giúp cho các nhà phân tích mã độc trên Android trích xuất được các đặc trưng để có thể mô hình hóa hành vi của các mẫu mã độc, từ đó có thể phân biệt được các loại mã độc khác nhau và mã sạch.

Andropytool tích hợp các công cụ nổi tiếng như AndroGuard, Droidbox, FlowDroid, AVClass, VirusTotal, Strace. Do đó Androidpytool cho phép người phân tích thu được nhiều đặc trưng khác nhau như: lời gọi hàm (API) , quyền được khai báo, quyền yêu cầu sử dụng (permission), nhãn (các họ mã độc) thu được từ các công cụ chống mã độc khác được tích hợp trong Virus Total, source-Sinks, các lời gọi API được gọi trong thời gian thực , các hoạt động truy cập, các hoạt động của người dùng v.v.

Andropytool tích hợp các công cụ phân tích khác nhau và các công cụ xử lý ứng dụng Android, từ đó cung cấp các đặc trưng liên quan tới các hành vi và đặc điểm riêng của từng mã độc.

* **Các bước để phân tích gom nhóm họ mã độc**

**B1: Lọc các tập tin định dạng .APK**. Việc lọc này dùng để kiểm tra xem tập tin APK được đưa vào liệu có phải là ứng dụng Android hợp lệ hay không.

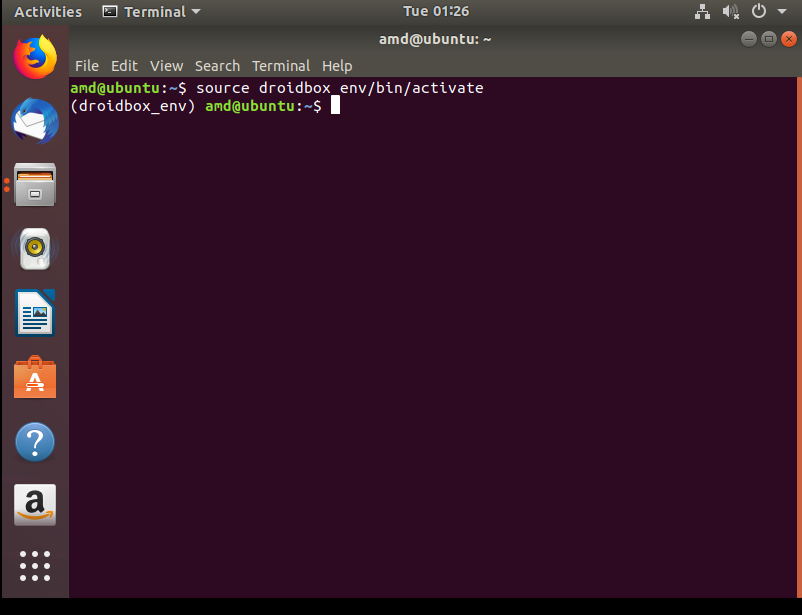
**B2: Phân tích VirusTotal.** Với mỗi tập tin được đưa vào VirusTotal sau khi được phân tích sẽ trả về kết quả từ hơn 60 công cụ phát hiện mã độc (Antivirus) khác nhau.

**B3: Phân đoạn dữ liệu.** Phân loại dựa trên báo cáo VirusTotal được sử dụng để chia bộ mẫu thành hai bộ: mẫu mã độc (có ít nhất một công cụ Antivirus cho kết quả tập tin đưa vào là mã độc) và mẫu mã sạch.

**B4: Thực thi FlowDroid.** Trong bước này, công cụ phân tích Android DroidBox được thực hiện cho từng mẫu.

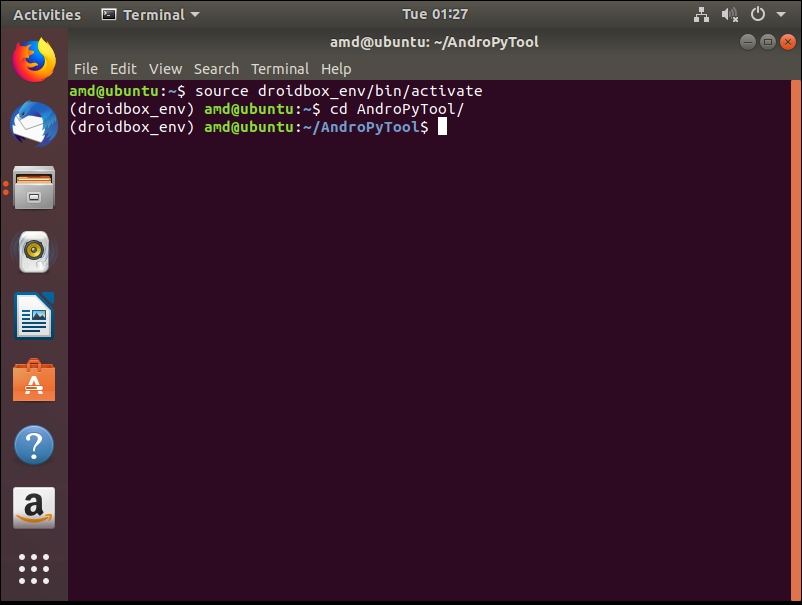
**B5: Trích xuất thông tin.** Tất cả các thông tin liên quan tới tập tin Android sẽ được trích xuất. Các thông tin này có thể được lưu thành 03 định dạng: .CSV, JSON và Mongo DB. Mỗi tập tin định dạng này sẽ lưu thông tin của một tập tin APK.

Dựa vào các bước phân tích như trên, sử dụng ubuntu để chạy thực nghiệm việc gom nhóm họ mã độc bằng TotalVirus được tích hợp trong Androidpytool như sau:



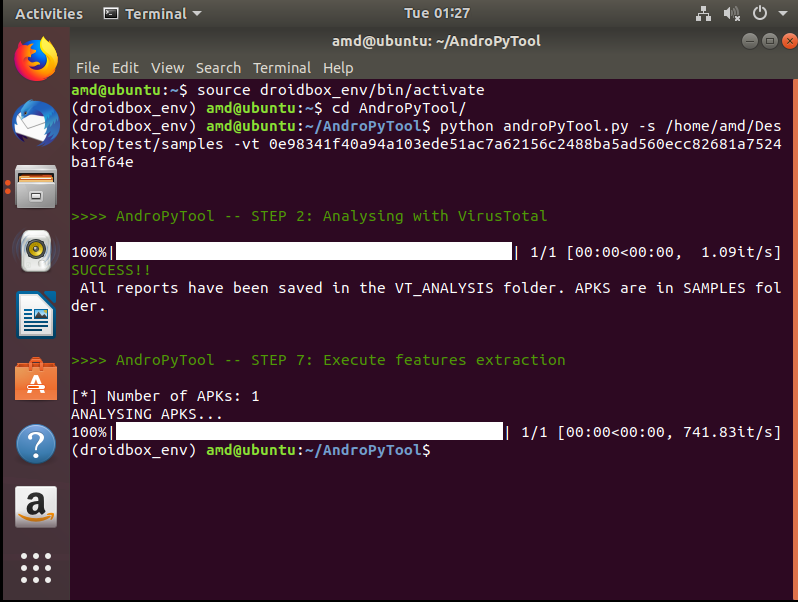
1. Khởi tạo môt trường cho andropytool

Bước khởi tạo này chính là bước kích hoạt môi trường. Để có thể vào được chương trình Andropytool.



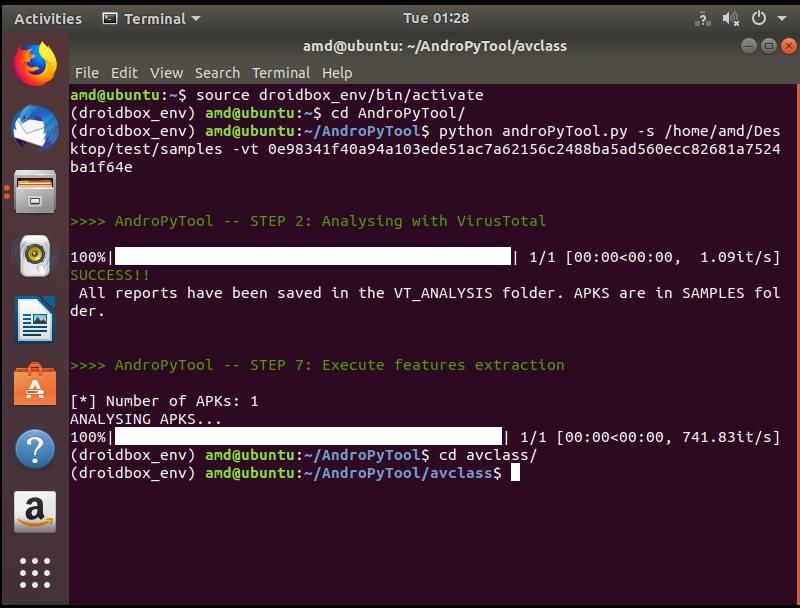
1. Truy cập vào thư mục andropytool

Trong hình 2, sử dụng câu lệnh cd AndroPytool để vào thư mục AndroPytool. Từ đó tiến hành phân tích một tập tin apk như trong hình 3.



1. Tiến hành phân tích với cơ sở dữ liệu của totalvirus

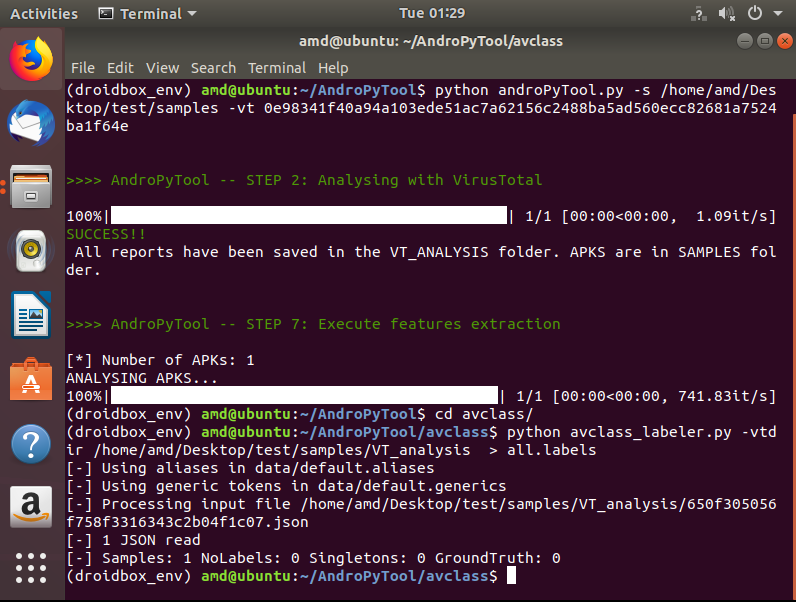
Trong quá trình phân tích tập tin, AndroPytool sẽ phân tích và gán tập tin này vào một nhãn. Giá trị nhãn này được gán dựa vào av\_class được truy cập vào thư mục như hình 4.



1. Truy cập vào tập tin av\_class

Sau khi truy cập vào avclass, bắt đầu chạy lệnh sau như hình 5:

Python avclass\_labeler.py -vtd ir /home/amd/Desktop/test/samples/VT\_analysis > all.labels



1. Lấy dữ liệu phân tích về

Sau khi phân tích xong tập tin, kết quả được lưu vào tập tin all.labels, trong đó gồm 2 thành phần: tên tập tin và nhãn được gán.



1. Họ virus được nhận diện với totalvirus

Quá trình gán nhãn của tập tin này được dùng kết hợp với các đặc trưng của tập tin. Từ đó làm đầu vào trong quá trình huấn luyện mô hình sử dụng học máy (được nêu trong chuyên đề 2) và so sánh với các nhãn được chẩn đoán trong giai đoạn kiểm tra.

Trong mục 1.3 và 1.4 tiếp theo, sẽ mô tả cách thức trích xuất đặc trưng tĩnh và động và nêu ý nghĩa của các đặc trưng đó.

## Trích xuất đặc trưng bằng phương pháp động

Phân tích động là hành động chạy ứng dụng Android (Cài đặt và chạy tập tin .apk), cho phép quan sát các hành vi trong ứng dụng và được ghi ra các file log. Bước này thường được thực hiện sau khi quá trình phân tích tĩnh kết thúc. Sử dụng kỹ thuật này có thể vượt qua được những hạn chế mà quá trình phân tích tĩnh không làm được như: packed, obfuscates,… Tuy nhiên phân tích động cũng có những hạn chế riêng như: cần tạo môi trường ảo để cho ứng dụng được cài đặt và chạy [7], thời gian để phân tích một tập tin lâu hơn phân tích tĩnh.

Để phân tích động thường sử dụng công nghệ Sanbox. Sanbox là công nghệ cho phép chạy các chương trình trong một môi trường an toàn, không sợ bị ảnh hưởng tới môi trường thực tế. Sanbox hội tụ đủ các yếu tố, kể cả kết nối mạng để chương trình được chạy như một môi trường thực tế.

Có rất nhiều Sanbox hỗ trợ phân tích mã độc: Cuckoo Sanbox, Norman Sanbox, GFI Sanbox,… Trong bài tài liệu này, chúng tôi sử dụng Droidbox được tích hợp trong AndroPyTool.

**Droidbox** được phát triển để thực hiện phân tích động trên nền tảng Android. Những thông tin sau khi thực hiện Droibox [8]:

* Hashes cho các package được phân tích
* Đầu vào/đầu ra của mạng dữ liệu
* Thao tác đọc và ghi tập tin
* Services được sử dụng và các classes được tải trong DexClassLoader
* Thông tin bị rò rỉ trong mạng, tệp tin và SMS
* Các quyền được chia sẻ
* Các hoạt động mã hóa được thực hiện bằng Android API
* Danh sách các recievers được phát hành
* Các tin nhắn được gửi và các cuộc gọi điện thoại

Trong Droidbox, máy ảo sử dụng Android nền tảng 4.1.2 với API cấp độ 16 và các thông số phần cứng phù hợp với nền tảng Android.

* **Thực hiện**

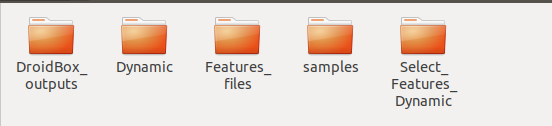
Tương tự với phân tích trích xuất đặc trưng bằng phương pháp tĩnh, phân tích trích xuất đặc trưng bằng phương pháp động cũng được thực hiện trên trình terminal với các bước sau đây:

*Bước 1:* Kích hoạt môi trường để phân tích và trích xuất

*Bước 2:* Di chuyển đến thư mục AndroPyTool

*Bước 3:* Thực hiện câu lệnh để phân tích và trích xuất các đặc trưng với cấu trúc lệnh như sau*: python androPyTool.py -s <PATH\_FOLDER\_APK> -dr*

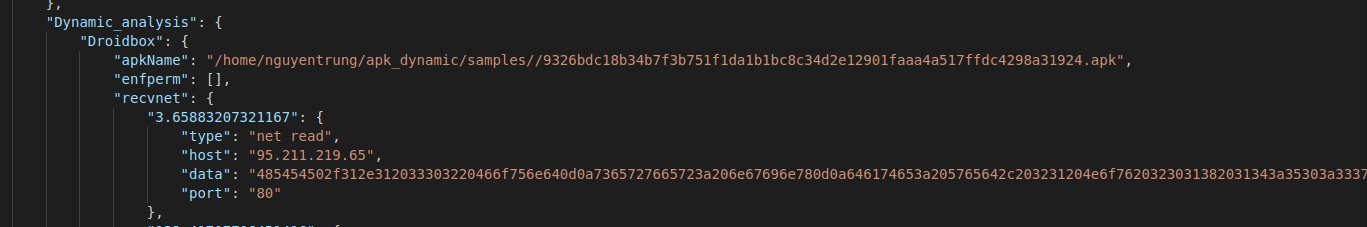
Sau khi kết thúc quá trình cũng có các folder chứa kết quả phân tích động.



1. Các folder chứa kết quả phân tích động

Trong đó:

* Thư mục **DroidBox\_outputs** chức các tập tin .txt ghi logcat và strace trong quá trình .apk chạy trên máy ảo.
* Thư mục **Features\_files** bao gồm các tập tin có phần mở rộng .json chứa các đặc trưng phân tích tĩnh.



1. Các thông tin lấy được trong phân tích động

Cấu trúc của đối tượng “**Dynamic\_Statics**” gồm các thành phần quan trọng sau:

* **Recvnet**

|  |
| --- |
|  |
| 1. Recvnet trong phân tích động |

Trong “**recvnet**” là các giá trị thời gian tính theo giây, trong mỗi giá trị đó là các các hành động như: net read, net write, host và port dữ liệu được đọc và ghi, ngoài ra còn có dữ liệu đã được mã hóa về hệ 16. Dữ liệu ở đây có thể là các dữ liệu người dùng bị đánh cắp. Qua đó, việc phân tích “**recvnet**” là cần thiết vì nó cho thông tin về dữ liệu đến và đi.

* **Accessedfiles**

|  |
| --- |
|  |
| 1. Accessedfiles |

Là các danh sách các đường dẫn tập tin được ứng dụng chấp nhận truy cập. Việc truy cập vào các tập tin này có thể để lưu dữ liệu được đánh cắp từ máy người dùng.

* **Opennet**

|  |
| --- |
|  |
| 1. Opennet trong phân tích động |

Các hoạt động kết nối tới địa chỉ IP và các port để gửi hoặc lấy dữ liệu.

* **Dataleaks**

|  |
| --- |
|  |
| 1. Dataleaks trong phân tích động |

**Dataleaks** cho biết được dữ liệu bị rò rỉ với chi tiết về loại rò rỉ, host và port nguồn, thẻ liên quan, nội dung dữ liệu rò rỉ nhưng đã bị mã hóa. Việc xác định được dataleaks chứng tỏ rõ ràng rằng ứng dụng này đang lấy đi thông tin của người dùng và lén lút gửi về một địa chỉ ID nào đó. Tuy nhiên, không phải mã độc nào cũng bị rò rỉ thông tin, vì vậy sẽ có lúc dataleaks rỗng.

* Thư mục **Dyndamic** chứa kết quả strace dưới dạng .csv và kết quả phân tích động được tách ra từ thư mục Features\_files.
* Thư mục **Samples** chứa các tập tin đã được phân tích động.
* Thư mục **Select\_Features\_Dynamic** chức các đặc trưng được chọn của tập tin .apk đó để làm đặc trưng cho tập tin đó.

Một số đặc trưng phổ biến được sử dụng trong trafic flow được mô tả trong bảng

|  |  |
| --- | --- |
| Feature | Description |
| Flow duration | Duration of the flow in Microsecond |
| flowBytesPerSecond | Number of flow bytes per second |
| Total\_opackets | Total packets in the outgoing direction |
| Total\_ipackets | Total packets in the incoming direction |
| Min\_opktl | Mininum size of packet in outgoing direction |
| Max\_opktl | Maximum size of packet in outgoing direction |
| Mean\_opktl | Mean size of packet in outgoing direction |
| Std\_opktl | Standard deviation size of packets in outgoing direction |
| Min\_ipktl | Minimum size of packet in imcoming direction |
| Max\_ipktl | Maximum size of packet in incoming direction |
| Mean\_ipktl | Mean size of packet in incoming direction |
| Std\_ipktl | Standard deviation size of packets in incoming direction |
| Min\_flowpktl | Minimum length of a flow |
| Max\_flowpktl | Maximum length of a flow |
| Mean\_flowpktl | Mean length of a flow |

## Kết luận

Sử dụng phương pháp động để có thể lấy được các đặc trưng mà chỉ có thực thi ứng dụng mới có thể có như:

* Hiệu suất sử dụng ứng dụng (sử dụng RAM, CPU, v.v.)
* Các traffic
* V.v.

Những đặc trưng này không thể có khi phân tích mã nguồn. Hạn chế được những nhược điểm trong vẫn đề phân tích tĩnh. Tuy nhiên, phân tích động cần thời gian phân tích nhiều hơn so với phương pháp phân tích tĩnh.

# TÀI LIỆU THAM KHẢO

1. Omer Tripp, Marco Pistoia, Patrick Cousot, Radhia Cousot, Salvatore Guarnieri, "An droneda: Accurate and scalable security analysis of web applications, Fundamental approaches to software engineering," in Fundamental approaches to software engineering. 16th international conference, FASE 2013, held as part of the European joint conferences on theory and practice of software, ETAPS 2013, Rome, Italy, Proceedings (pp. 220 - 225), March 2013.
2. H. Alatwi, “Android malware detection using category-based machine learning classiﬁers”, Rochester institute of technology, Masters thesis, June 2016.
3. "DeverloperAndroid," [Online]. Available: <https://developer.android.com/studio/build/application-id>.
4. Pham Thanh Giang, Nguyen Viet Duc, Pham Minh Vi, "Permission Analysis for Android Malware Detection, The Proceedings," in The Proceedings of the 7th VAST-AIST workshop “Research collaboration: review and prespective”, 11/2015.
5. Altyeb Altaher, Omar Mohammed Barukab, "Intelligent Hybrid Approach for Androi Malware Detection based on Permission and API Calls," in International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 8, No. 6, 2017.
6. Justin Del Vecchio, Feng Shen, Kenny M. Yee, Boyu Wang, Steven Y.Ko, Lukasz Ziarek, in String Analysis of Android Applications- AndroDialysis: Analysis of Android Intent Effectiveness in Malware Dectection- Ali Feizollahm, Nor Badrul Anuar, 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE).
7. "Securitybox" [Online]. Available: <https://securitybox.vn/2337/cac-ky-thuat-phan-tich-dong-muc-co-ban-phan-1/>.
8. "GitHub" [Online]. Available: <https://github.com/pjlantz/droidbox>.
9. D.-J. Wu, C.-H. Mao and T.-E. Wei, "Droid mat: Android malware detection through manifest and API calls tracing," in Seventh Asia Joint Conference on Information Security, pp. 62 - 69, 2012.
10. "GitHub" [Online]. Available: <https://github.com/idanr1986/droidmon>.
11. Nguyễn Thị Biên, “Khai phá luật kết hợp trong cơ sở dữ liệu đa phương tiện”, luân văn thạc sĩ ngành công nghệ phần mềm, Đại học công nghệ, 2012.
12. Chenglin Li, Keith Mills, Rui Zhu, Di Niu, Hongwen Zhang, Husam Kinawi, Android Malware Detection based on Factorization Machine, arXiv:1805.11843.
13. Daniel Arp, Michael Spreitzenbarth, Malte Huebner, Hugo Gascon, and Konrad Rieck ["Drebin: Efficient and Explainable Detection of Android Malware in Your Pocket"](https://www.tu-braunschweig.de/Medien-DB/sec/pubs/2014-ndss.pdf), 21th Annual Network and Distributed System Security Symposium (NDSS), February 2014